

Copyright is owned by the Author of the thesis. Permission is given for a copy to be downloaded by an individual for the purpose of research and private study only. The thesis may not be reproduced elsewhere without the permission of the Author.

Class AttriSet

```
package lozpt1;

import java.awt.*;
import javax.swing.text.*;
import javax.swing.*;

public class AttriSet {
    static AttriSet aSet= new AttriSet();
    static ImageIcon c1= new ImageIcon();
    static ImageIcon c2= new ImageIcon();
    static SimpleAttributeSet[] mainTextAs = new
        SimpleAttributeSet[10];
    static SimpleAttributeSet[] rBotTextAs = new
        SimpleAttributeSet[10];
    static SimpleAttributeSet[] rTopTextAs = new
        SimpleAttributeSet[10];

    static SimpleAttributeSet mainHead = new SimpleAttributeSet( );
    static SimpleAttributeSet heading0 = new SimpleAttributeSet( );
    static SimpleAttributeSet heading1 = new SimpleAttributeSet( );
    static SimpleAttributeSet heading2 = new SimpleAttributeSet( );
    static SimpleAttributeSet heading3 = new SimpleAttributeSet( );
    static SimpleAttributeSet heading4 = new SimpleAttributeSet( );
    static SimpleAttributeSet para1 = new SimpleAttributeSet( );
    static SimpleAttributeSet para2 = new SimpleAttributeSet( );
    static SimpleAttributeSet para3 = new SimpleAttributeSet( );
    static SimpleAttributeSet para4 = new SimpleAttributeSet( );
    static SimpleAttributeSet topNote = new SimpleAttributeSet( );
    static SimpleAttributeSet midNote = new SimpleAttributeSet( );
    static SimpleAttributeSet botNote = new SimpleAttributeSet( );
    static SimpleAttributeSet imagel1 = new SimpleAttributeSet( );
    static SimpleAttributeSet image2 = new SimpleAttributeSet( );
    static SimpleAttributeSet highL1 = new SimpleAttributeSet( );
    static SimpleAttributeSet highL2 = new SimpleAttributeSet( );
    public AttriSet() {
    }
    public static void setStyle(int tbox, int plan){
        switch (plan){
            case 1: switch (tbox) {
                case 1: planMainT1(); break;
                case 2: planTopT1(); break;
                case 3: planBotT1(); break;
            };
            break;
            case 2: switch (tbox) { //this is temporary work
                case 1: planMainT2(); break;
                case 2: planTopT2(); break;
                case 3: planBotT2(); break;
            };; break;
        }
    }

    public static void setStyle(int tbox, int plan, ImageIcon c){
        c1=c;
        setStyle(tbox,plan);
    }

    public static void setStyle(int tbox, int plan, ImageIcon c11,
    ImageIcon c22){
        c1=c11;c2=c22;
    }
}
```

```

        setStyle(tbox,plan);
    }

    public static void setPCmainT(){
        mainTextAs[0]=botNote;//heading
        mainTextAs[1]=para1;mainTextAs[2]=para1;
        mainTextAs[3]=para1;mainTextAs[4]=para1;mainTextAs[5]=para1;
        mainTextAs[6]=heading2; mainTextAs[7]=para2;
        mainTextAs[8]=heading2;mainTextAs[9]=para2;
        planMainT1();
    }

    public static void setHL(){
        StyleConstants.setForeground(highL1, Color.black);
        StyleConstants.setFontFamily(highL1, "Serif");
        StyleConstants.setFontSize(highL1, 30);
        StyleConstants.setBold(highL1, true);
        StyleConstants.setBackground(highL1, Color.yellow);
        StyleConstants.setForeground(highL2, Color.yellow);
        StyleConstants.setFontFamily(highL2, "Serif");
        StyleConstants.setFontSize(highL2, 30);
        StyleConstants.setBold(highL2, true);
        StyleConstants.setBackground(highL2, Color.black);
    }

    public static void setNote(ImageIcon c11){
        StyleConstants.setAlignment(image1,StyleConstants.ALIGN_CENTER);
        //StyleConstants.setIcon(image1,c11);
        StyleConstants.setForeground(topNote, Color.darkGray);
        StyleConstants.setFontFamily(topNote, "Impact");
        StyleConstants.setFontSize(topNote, 35);
        StyleConstants.setBold(topNote, true);
        StyleConstants.setForeground(midNote, Color.darkGray);
        StyleConstants.setFontFamily(midNote, "Arial");
        StyleConstants.setFontSize(midNote, 30);
        StyleConstants.setBold(midNote, false);
        StyleConstants.setForeground(botNote, Color.black);
        StyleConstants.setFontFamily(botNote, "Serif");
        StyleConstants.setFontSize(botNote, 30);
        StyleConstants.setBold(botNote, true);
    }

    private static void planMainT1(){
        StyleConstants.setForeground(botNote, Color.darkGray);
        StyleConstants.setFontFamily(botNote, "Serif");
        StyleConstants.setFontSize(botNote, 25);
        StyleConstants.setBold(botNote, true);
        StyleConstants.setForeground(mainHead, Color.darkGray);
        StyleConstants.setFontFamily(mainHead, "Impact");
        StyleConstants.setFontSize(mainHead, 30);
        StyleConstants.setBold(mainHead, true);
        StyleConstants.setForeground(heading0, Color.black);
        StyleConstants.setFontFamily(heading0, "Serif");
        StyleConstants.setFontSize(heading0, 30);
        StyleConstants.setBold(heading0, true);
        StyleConstants.setForeground(heading1, Color.darkGray);
        StyleConstants.setFontFamily(heading1, "Helvetica");
        StyleConstants.setFontSize(heading1, 25);
    }

```

```

StyleConstants.setBold(heading1, true);
StyleConstants.setForeground(heading2, Color.black);
StyleConstants.setFontFamily(heading2, "Serif");
StyleConstants.setFontSize(heading2, 25);
StyleConstants.setBold(heading2, true);
StyleConstants.setForeground(heading3, Color.darkGray);
StyleConstants.setFontFamily(heading3, "Arial");
StyleConstants.setFontSize(heading3, 20);
StyleConstants.setBold(heading3, true);
StyleConstants.setForeground(heading4, Color.darkGray);
StyleConstants.setFontFamily(heading4, "Impact");
StyleConstants.setFontSize(heading4, 20);
StyleConstants.setBold(heading4, true);
StyleConstants.setItalic(heading4, true);
StyleConstants.setForeground(para1, Color.black);
StyleConstants.setFontFamily(para1, "Arial");
StyleConstants.setFontSize(para1, 20);
StyleConstants.setForeground(para2, Color.darkGray);
StyleConstants.setFontFamily(para2, "Sherif");
StyleConstants.setFontSize(para2, 20);
StyleConstants.setItalic(para2, true);
StyleConstants.setForeground(para3, Color.gray);
StyleConstants.setFontFamily(para3, "Impact");
StyleConstants.setAlignment(para3, StyleConstants.ALIGN_CENTER);
StyleConstants.setLeftIndent(para3, 5);
StyleConstants.setFontSize(para3, 20);
StyleConstants.setItalic(para3, true);
StyleConstants.setForeground(para4, Color.black);
StyleConstants.setFontFamily(para4, "Arial");
StyleConstants.setFontSize(para4, 20);
StyleConstants.setItalic(para4, true);
StyleConstants.setLeftIndent(para4, 10);
StyleConstants.setAlignment(image1, StyleConstants.ALIGN_CENTER);
StyleConstants.setIcon(image1, c1);
StyleConstants.setAlignment(image2, StyleConstants.ALIGN_CENTER);
StyleConstants.setIcon(image2, c2);
}

private static void planMainT2(){
StyleConstants.setForeground(mainHead, Color.magenta);
StyleConstants.setFontFamily(mainHead, "Impact");
StyleConstants.setFontSize(mainHead, 35);
StyleConstants.setBold(mainHead, true);
StyleConstants.setForeground(heading0, Color.blue);
StyleConstants.setFontFamily(heading0, "Arial");
StyleConstants.setFontSize(heading0, 30);
StyleConstants.setBold(heading0, true);
StyleConstants.setForeground(heading1, Color.red);
StyleConstants.setFontFamily(heading1, "Impact");
StyleConstants.setFontSize(heading1, 30);
StyleConstants.setBold(heading1, true);
StyleConstants.setForeground(heading2, Color.blue);
StyleConstants.setFontFamily(heading2, "Serif");
StyleConstants.setFontSize(heading2, 25);
StyleConstants.setBold(heading2, true);
StyleConstants.setForeground(heading3, Color.blue);
StyleConstants.setFontFamily(heading3, "Arial");
StyleConstants.setFontSize(heading3, 20);
StyleConstants.setBold(heading3, true);
StyleConstants.setForeground(heading4, Color.blue);
StyleConstants.setFontFamily(heading4, "Impact");
}

```

```

StyleConstants.setFontSize(heading4, 20);
StyleConstants.setBold(heading4, true);
StyleConstants.setItalic(heading4, true);
StyleConstants.setForeground(para1, Color.darkGray);
StyleConstants.setFontFamily(para1, "Arial");
StyleConstants.setFontSize(para1, 20);
StyleConstants.setForeground(para2, Color.blue);
StyleConstants.setFontFamily(para2, "Sherif");
StyleConstants.setFontSize(para2, 20);
StyleConstants.setItalic(para2, true);
StyleConstants.setForeground(para3, Color.cyan);
StyleConstants.setFontFamily(para3, "Impact");
StyleConstants.setFontSize(para3, 20);
StyleConstants.setItalic(para3, true);
StyleConstants.setForeground(para4, Color.cyan);
StyleConstants.setFontFamily(para4, "Arial");
StyleConstants.setFontSize(para4, 20);
StyleConstants.setItalic(para4, true);
StyleConstants.setLeftIndent(para4, 10);
}

private static void planTopT1(){
    StyleConstants.setForeground(heading1, Color.red);
    StyleConstants.setFontFamily(heading1, "Impact");
    StyleConstants.setFontSize(heading1, 20);
    StyleConstants.setBold(heading1, true);
    StyleConstants.setForeground(heading2, Color.black);
    StyleConstants.setFontFamily(heading2, "Serif");
    StyleConstants.setFontSize(heading2, 15);
    StyleConstants.setBold(heading2, true);
    StyleConstants.setForeground(heading3, Color.darkGray);
    StyleConstants.setFontFamily(heading3, "Serif");
    StyleConstants.setFontSize(heading3, 10);
    StyleConstants.setBold(heading3, true);
    StyleConstants.setForeground(heading4, Color.black);
    StyleConstants.setFontFamily(heading4, "Arial");
    StyleConstants.setFontSize(heading4, 10);
    StyleConstants.setBold(heading4, true);
    StyleConstants.setItalic(heading4, true);
    StyleConstants.setForeground(para1, Color.darkGray);
    StyleConstants.setFontFamily(para1, "Arial");
    StyleConstants.setFontSize(para1, 10);
    StyleConstants.setForeground(para2, Color.black);
    StyleConstants.setFontFamily(para2, "Sherif");
    StyleConstants.setFontSize(para2, 10);
    StyleConstants.setItalic(para2, true);
    StyleConstants.setForeground(para3, Color.black);
    StyleConstants.setFontFamily(para3, "Arial");
    StyleConstants.setFontSize(para3, 10);
    StyleConstants.setItalic(para3, true);
    StyleConstants.setForeground(para4, Color.black);
    StyleConstants.setFontFamily(para4, "Arial");
    StyleConstants.setFontSize(para4, 10);
    StyleConstants.setItalic(para4, true);
    StyleConstants.setLeftIndent(para4, 10);
}

private static void planBotT1(){
    StyleConstants.setForeground(heading1, Color.darkGray);
    StyleConstants.setFontFamily(heading1, "Sherif");
    StyleConstants.setFontSize(heading1, 20);

```

```

        StyleConstants.setBold(heading1, true);
        StyleConstants.setForeground(heading2, Color.darkGray);
        StyleConstants.setFontFamily(heading2, "Serif");
        StyleConstants.setFontSize(heading2, 15);
        StyleConstants.setBold(heading2, true);
        StyleConstants.setForeground(heading3, Color.black);
        StyleConstants.setFontFamily(heading3, "Serif");
        StyleConstants.setFontSize(heading3, 10);
        StyleConstants.setBold(heading3, true);
        StyleConstants.setForeground(heading4, Color.darkGray);
        StyleConstants.setFontFamily(heading4, "Arial");
        StyleConstants.setFontSize(heading4, 10);
        StyleConstants.setBold(heading4, true);
        StyleConstants.setItalic(heading4, true);
        StyleConstants.setForeground(para1, Color.black);
        StyleConstants.setFontFamily(para1, "Arial");
        StyleConstants.setFontSize(para1, 10);
        StyleConstants.setForeground(para2, Color.darkGray);
        StyleConstants.setFontFamily(para2, "Sherif");
        StyleConstants.setFontSize(para2, 10);
        StyleConstants.setItalic(para2, true);
        StyleConstants.setForeground(para3, Color.darkGray);
        StyleConstants.setFontFamily(para3, "Arial");
        StyleConstants.setFontSize(para3, 10);
        StyleConstants.setItalic(para3, true);
        StyleConstants.setForeground(para4, Color.darkGray);
        StyleConstants.setFontFamily(para4, "Arial");
        StyleConstants.setFontSize(para4, 10);
        StyleConstants.setItalic(para4, true);
        StyleConstants.setLeftIndent(para4, 10);
    }

private static void planTopT2(){
    StyleConstants.setForeground(heading1, Color.darkGray);
    StyleConstants.setFontFamily(heading1, "Tahoma");
    StyleConstants.setFontSize(heading1, 25);
    StyleConstants.setBold(heading1, true);
    StyleConstants.setForeground(heading2, Color.black);
    StyleConstants.setFontFamily(heading2, "Serif");
    StyleConstants.setFontSize(heading2, 10);
    StyleConstants.setBold(heading2, true);
    StyleConstants.setForeground(heading3, Color.darkGray);
    StyleConstants.setFontFamily(heading3, "Serif");
    StyleConstants.setFontSize(heading3, 8);
    StyleConstants.setBold(heading3, true);
    StyleConstants.setForeground(heading4, Color.black);
    StyleConstants.setFontFamily(heading4, "Arial");
    StyleConstants.setFontSize(heading4, 8);
    StyleConstants.setBold(heading4, true);
    StyleConstants.setItalic(heading4, true);
    StyleConstants.setForeground(para1, Color.black);
    StyleConstants.setFontFamily(para1, "Helvetica");
    StyleConstants.setFontSize(para1, 20);
    StyleConstants.setForeground(para2, Color.darkGray);
    StyleConstants.setFontFamily(para2, "Sherif");
    StyleConstants.setFontSize(para2, 19);
    StyleConstants.setItalic(para2, true);
    StyleConstants.setForeground(para3, Color.darkGray);
    StyleConstants.setFontFamily(para3, "Arial");
    StyleConstants.setFontSize(para3, 8);
    StyleConstants.setItalic(para3, true);
}

```

```

        StyleConstants.setForeground(para4, Color.black);
        StyleConstants.setFontFamily(para4, "Arial");
        StyleConstants.setFontSize(para4, 8);
        StyleConstants.setItalic(para4, true);
        StyleConstants.setLeftIndent(para4, 10);
    }

private static void planBotT2(){
    StyleConstants.setForeground(heading1, Color.darkGray);
    StyleConstants.setFontFamily(heading1, "Sherif");
    StyleConstants.setFontSize(heading1, 25);
    StyleConstants.setBold(heading1, true);
    StyleConstants.setForeground(heading2, Color.black);
    StyleConstants.setFontFamily(heading2, "Serif");
    StyleConstants.setFontSize(heading2, 10);
    StyleConstants.setBold(heading2, true);
    StyleConstants.setForeground(heading3, Color.darkGray);
    StyleConstants.setFontFamily(heading3, "Serif");
    StyleConstants.setFontSize(heading3, 8);
    StyleConstants.setBold(heading3, true);
    StyleConstants.setForeground(heading4, Color.darkGray);
    StyleConstants.setFontFamily(heading4, "Arial");
    StyleConstants.setFontSize(heading4, 8);
    StyleConstants.setBold(heading4, true);
    StyleConstants.setItalic(heading4, true);
    StyleConstants.setForeground(para1, Color.darkGray);
    StyleConstants.setFontFamily(para1, "Arial");
    StyleConstants.setFontSize(para1, 18);
    StyleConstants.setForeground(para2, Color.black);
    StyleConstants.setFontFamily(para2, "Sherif");
    StyleConstants.setFontSize(para2, 17);
    StyleConstants.setItalic(para2, true);
    StyleConstants.setForeground(para3, Color.black);
    StyleConstants.setFontFamily(para3, "Arial");
    StyleConstants.setFontSize(para3, 8);
    StyleConstants.setItalic(para3, true);
    StyleConstants.setForeground(para4, Color.darkGray);
    StyleConstants.setFontFamily(para4, "Arial");
    StyleConstants.setFontSize(para4, 8);
    StyleConstants.setItalic(para4, true);
    StyleConstants.setLeftIndent(para4, 10);
}
}
}

```

Class BasicConcept

```

package lozpt1;

/* <p>Company: </p>
 * @author unascribed
 * @version 1.0
 */

public {
    private short basCptID;
    private short cptType;
    private String heading;
}

```

```

private String what;
private String why;
private String how;
private short minTime;
private short maxTime;
private short basicMSID;
private MentalState basicMS=null;

public BasicConcept(short bcID, short cType, String h, String wt,
    String wy, String hw, short miT, short mxT, short mS) {
    basCptID= bcID; cptType=cType; heading = h; what=wt;
    why=wy; how=hw; minTime=miT; maxTime=mxT;
    basicMSID=mS;
}
public short getBCID(){
    return basCptID;
}

public String getWhat(){
    return what;
}
public String getWhy(){
    return why;
}
public String getHow(){
    return how;
}
public String getHead(){
    return heading;
}
public short getType(){
    return cptType;
}
public short getMinTime(){
    return minTime;
}
public short getMaxTime(){
    return maxTime;
}

public MentalState getBasicMS(){
    if (basicMS ==null)
        basicMS=MentalStateDB.msDB().findaMentalState(basicMSID);
    return basicMS;
}
}

```

Class BasicConceptDB

```

package lozpt1;

import com.borland.dx.dataset.*;
import com.borland.dx.sql.dataset.*;
import java.lang.*;

public class BasicConceptDB extends LozDB{
    private static BasicConceptDB bcDB = new BasicConceptDB();

    public BasicConceptDB(){
    }
}

```



```

public static BasicConceptDB bcDB(){
    return bcDB;
}
public BasicConcept findaBasicConcept(short bcID){
    short type=1;
    BasicConcept bc = (BasicConcept)Cache.find(new Short(bcID),type);
    if (bc==null){
        bc=findINdb(bcID);
        Cache.put(new Short(bcID), bc,type);
    }
    return bc;
}

public BasicConcept findINdb(short bcID){
    QueryDataSet getBasQ = new QueryDataSet();
    QueryDescriptor qD = new QueryDescriptor(LOZdB, "SELECT
        \"BasicConcept\".*\"+ \" FROM \"BasicConcept\" WHERE
        \"BasicConcept\".\"bcID\" = \" + bcID , null, false, Load.ALL);
    getBasQ.setQuery(qD);
    getBasQ.executeQuery();
    LOZdB.closeConnection();
    if (getBasQ.isEmpty())
        return null;
        BasicConcept bc= new BasicConcept( bcID, getBasQ.getShort(2),
            getBasQ.getString(3), getBasQ.getString(4),
            getBasQ.getString(5), getBasQ.getString(6),
            getBasQ.getShort(7), getBasQ.getShort(8),
            getBasQ.getShort(9));
    getBasQ.closeStatement();
    return bc;
}
}

```

Class Cache

```

package lozpt1;

import java.util.*;

public {
    private static Map concepts= new TreeMap();
    private static Map mcqs= new TreeMap();
    private static Map mcqAnws= new TreeMap();
    private static Map misconcepts= new TreeMap();
    private static Map mentalStates= new TreeMap();
    private static Map lms= new TreeMap();
    private static Map learner1= new TreeMap();
    private static Map learner2= new TreeMap();
    public Cache() {
    }
    public static Map findmap(short type ){
        Map map=null;
        switch (type) {
            case 1: map= concepts; break;
            case 2: map= misconcepts; break;
            case 3: map= mcqs; break;
            case 4: map= mcqAnws; break;
            case 5: map= mentalStates; break;
            case 6: map= lms; break;
        }
    }
}

```

```

        case 7: map= learner1;break;
        case 8: map= learner2;break;
    }
    return map;
}
public static Object find(Object key, short type ){
    Map map=findmap(type);
    return map.get(key);
}
public static void put(Object key, Object value, short type ){
    Map map=findmap(type);
    map.put(key,value);
}
}
}

```

Class CurrentSession

```

package lozpt1;

public class CurrentSession extends Session{
    private long curActSTime;
    private long curLCSTime; // there is Session.StartTime also
    private MainConcept curMC=null;
    private LearnerMC curLMC=null;
    private LearnerMCQ curLMCQ=null;
    private LearnConcept curLC=null;
    private MentalState curMS=null;
    private LearnerMS curLMS=null;
    private LearnerMS[] preLMS={null,null};
    private boolean firstAttempt=true;
    private MCQ curMCQ=null;
    private short currentAction=0;
    // showing preConcept-100;MC-200;LC1-300;LC1Q1-310;LC1Q1FB1-311
    private short currentState=0;
    private short aOa=0;
    private String shortSummary="";
    private short curScuffL=0;
    public CurrentSession(short sesID,short lID,long sT, long d, short
        lsID, String S, MainConcept mc) {
        super(sesID,lID,sT,d,lsID,S); curMC=mc;
    }
    public void setCurAoA(short a){
        aOa=a;
    }
    public short getCurAoA(){
        return aOa;
    }
    public short getCurAction(){
        return currentAction;
    }
    public void setCurAction(short a){
        currentAction=a;
    }
    public short getCurState(){
        return currentState;
    }
    public void setCurState(short s){
        currentState=s;
    }
    /*public void plusCurState(short s){

```

```

        currentState += s;
    }*/
    public short getShoSumm(){
        return currentState;
    }
    public void setShoSumm(String s){
        shortSummary=s;
    }
    public void setCurActSTime(long t){
        curActSTime=t;
    }
    public long getCurActSTime(){
        return curActSTime;
    }
    public void setCurLC(LearnConcept L){
        curLC=L;
    }

    public void setCurLCSTime(long t){
        curLCSTime=t;
    }
    public long getCurLCSTime(){
        return curLCSTime;
    }
    public MainConcept getCurMC(){
        return curMC;
    }
    public LearnerMC getCurLMC(){
        return curLMC;
    }
    public LearnConcept getCurLC(){
        return curLC;
    }
    public MentalState getCurMS(){
        return curMS;
    }
    public LearnerMS getPreLMS(int i){
        return preLMS[i];
    }

    public void setPreLMS(LearnerMS lms,int i){
        preLMS[i]=lms;
    }
    public LearnerMS getCurLMS(){
        return curLMS;
    }
    public MCQ getCurMCQ(){
        return curMCQ;
    }
    public LearnerMCQ getCurLMCQ(){
        return curLMCQ;
    }
    public void setCurMS(MentalState MS){
        curMS=MS;
    }
    public void setCurLMS(LearnerMS LMS){
        curLMS=LMS;
    }
    public void setCurLMC(LearnerMC LMC){
        curLMC=LMC;
    }
}

```

```

public void setCurMCQ(MCQ Mcq){
    curMCQ=Mcq;
}
public void setCurLMCQ(LearnerMCQ LMcq){
    curLMCQ=LMcq;
}
public void setCurScuffL(short i){
    curScuffL=i;
}
public short getCurScuffL(){
    return curScuffL;
}
public void setFirstAttempt (boolean b){
    firstAttempt=b;
}
public boolean firstAttempt(){
    return(firstAttempt);
}
}

```

Class Example

```

package lozpt1;

import javax.swing.*;

public class Example {
    private String ExampleText;
    private ImageIcon ExampleImg;
    public Example(String et, String eft){
        ExampleText = et;
        String path = LozGlobal.lozG().path();
        ExampleImg=new ImageIcon(path+eft);
    }
    public String getExampleText(){
        return ExampleText;
    }
    public ImageIcon getExampleImg(){
        return ExampleImg;
    }
}

```

Class ExploreConcept

```

package lozpt1;

public class ExploreConcept extends BasicConcept{
    private String exp1;
    private String exp2;
    public ExploreConcept(short bcID, short cType, String h, String wt,
        String wy, String hw, short miT, short mxT, short mS, String e1,
        String e2) {
        super(bcID,cType,h,wt,wy,hw,miT,mxT,mS);
        exp1 = e1; exp2 = e2;
    }
    public String exp(int i){
        if (i==0)
            return exp1;
    }
}

```

```

    else
        return exp2;
    }
}

```

Class ExploreConceptDB

```

package lozpt1;

import com.borland.dx.dataset.*;
import com.borland.dx.sql.dataset.*;
import java.sql.*;
import java.lang.*;

public class ExploreConceptDB extends LozDB{
    private static ExploreConceptDB ecDB = new ExploreConceptDB();
    public ExploreConceptDB(){
    }
    public static ExploreConceptDB ecDB(){
        return ecDB;
    }
    public ExploreConcept findaExploreConcept(short ecID){
        short type=1;
        ExploreConcept ec = (ExploreConcept)Cache.find(new
            Short(ecID),type);
        if (ec==null){
            ec=findINdb(ecID);
            Cache.put(new Short(ecID), ec,type);
        }
        return ec;
    }
    public ExploreConcept findINdb(short ecID){
        QueryDataSet getBasQ = new QueryDataSet();
        QueryDescriptor qD = new QueryDescriptor(LOZdB, "SELECT
            \"BasicConcept\".*\"+ \" FROM \"BasicConcept\" WHERE
            \"BasicConcept\".\"bcID\" = \" + ecID , null, false, Load.ALL);
        getBasQ.setQuery(qD);
        getBasQ.executeQuery();
        if (getBasQ.isEmpty())
            return null;
        QueryDataSet getExpQ = new QueryDataSet();
        QueryDescriptor qD1 = new QueryDescriptor(LOZdB, "SELECT
            \"ExploreConcept\".*\"+ \" FROM \"ExploreConcept\" WHERE
            \"ExploreConcept\".\"exID\" = \" + ecID , null, false, Load.ALL);
        getExpQ.setQuery(qD1);
        getExpQ.executeQuery();
        LOZdB.closeConnection();
        if (getExpQ.isEmpty())
            return null;
        ExploreConcept ec= new ExploreConcept(ecID, getBasQ.getShort(2),
            getBasQ.getString(3),getBasQ.getString(4),getBasQ.getString(5),
            getBasQ.getString(6), getBasQ.getShort(7), getBasQ.getShort(8),
            getBasQ.getShort(9), getExpQ.getString(2),
            getExpQ.getString(3));
        getBasQ.closeStatement();
        getExpQ.closeStatement();
        return ec;
    }
}

```

Class FBCtrl

```
package lozpt1;

import java.util.*;
import javax.swing.JOptionPane;
import java.awt.*;

public class FBCtrl {
    private static Mentor M;
    private boolean packFrame = false;
    private static FBUI fbUI=null;
    public FBCtrl(Mentor m ) {
        M=m;
        fbUI = new FBUI(this,"Feedback Page");
        fbUI.setSize(new Dimension(800,700));
        Util.U.setCentre(fbUI);
        try {
            if (packFrame)
                fbUI.pack();
            else
                fbUI.validate();
        }
        catch(Exception e) {
            e.printStackTrace();
        }
    }

    public FBUI fbUI(){
        return fbUI;
    }
    public void setMentor( Mentor m){
        M=m;
    }
    public Mentor getMentor( ){
        return M;
    }

    public void setFBUI(String title, int x, int y){
        fbUI.setSize(new Dimension(x,y));
        Util.U.setCentre(fbUI);
        fbUI.validate();
        fbUI.setVisible(true);
    }
    public void OKpressed(){
        fbUI.dispose();
        M.fbOKpressed();
    }
    public void skipPressed(){
        fbUI.dispose();
        M.fbSKIPpressed();
    }
    public void processFB1(short index, short sA){
        long sT= System.currentTimeMillis(); //start time in mlSec
        M.getCsession().setCurActSTime(sT);
        //M.getCsession().plusCurState((short)10);
        M.mLCtrl().mcqDlog().dispose();
        switch (index) {
            case 1: pasW1a(sA);break;
        }
    }
}
```

```

        case 2: pasW2a(sA);break;
        case 3: pasW3(sA);break;
        case 4: pasW4(sA);break;
        case 5: pasW5(sA);break;
        case 11: pasC1(sA);break;
        case 12: pasC2(sA);break;
        case 13: pasC3(sA);break;
        case 14: pasC4(sA);break;
        case 15: pasC5(sA);break;
    }
}
public void processFB2W(short index, short sA){
    long sT= System.currentTimeMillis(); //start time in mSec
    M.getCsession().setCurActSTime(sT);
    switch (index) {
        case 1: pasW1W(sA);break;
        case 2: pasW2W(sA);break;
    }
}
public void processFB2C(short index, short sA){
    long sT= System.currentTimeMillis(); //start time in mSec
    M.getCsession().setCurActSTime(sT);
    M.mLCtrl().mcqDlog().setVisible(false);
    switch (index) {
        case 1: pasW1C(sA);break;
        case 2: pasW2C(sA);break;
    }
}

public void pasW1a(short sA){
    JOptionPane.showMessageDialog(M.mLCtrl().mLUI()," You selected
    answer "+ String.valueOf(sA)+ ". \n" + " It is not correct, try
    the SAME Question Again.", "Feedback - Try Again.",
        JOptionPane.ERROR_MESSAGE);
    M.fbOKpressed(); //pretend the above dialogbox as FBUI.
    // with fbOKpressed goes to m.next
}

public void pasW2a(short sA){
    String t=" You selected answer "+ String.valueOf(sA)+ ". It is
    not correct \n"; String[] c={"Because :"+
    M.getCsession().getCurMCQ().answer(sA).whyWorC()+ "\n"};
    String b=" Next ==> try the SAME Question Again.";
    fbUI.include(t,c,0,b);
    setFBUI("Feedback - Try Again.",800,300);
    // with fbOKpressed goes to m.next
}

public void pasW1C(short sA){ //like pasC1
    M.mLCtrl().mcqDlog().setVisible(false);
    JOptionPane.showMessageDialog(M.mLCtrl().mLUI(),"Weldone,
    "+M.getLearner().userID()+ " Answer; "+ String.valueOf(sA)+ " is
    correct. \n Try next question.", "Feedback- You Did
    It.",JOptionPane.OK_OPTION);
    M.fbOKpressed(); ////pretend the above dialogbox as FBUI. Later
    goes to m.next
}

public void pasW2C(short sA){ //like pasC2
    M.mLCtrl().mcqDlog().setVisible(false);

```

```

M.getCsession().setCurAction((short)20);
String t="Weldone. "+" You selected answer; "+
String.valueOf(sA)+ ". It is correct. \n"; String[] c={" "+
M.getCsession().getCurMCQ().answer(sA).whyWorC()+ "\n"};
String b=" Next ==> SAME LEVEL MCQ.";
fbUI.include(t,c,0,b);
setFBUI("Feedback - Weldone.",800,300);
// with fbOKpressed goes to m.next
}

public void pasW1W(short sA){ //like pasW3
String t=" You are not correct again. \n";
String[] c = {"Because :"+ M.getCsession().getCurMCQ(). answer(sA).
whyWorC() + "\n", M.getCsession().getCurMCQ(). whyWorCfull()," "};
if (M.getLearner().knowZ())
c[2]=M.getCsession().getCurMCQ().zFeed();
String b=" Next ==> ANOTHER MCQ, but at SAME LEVEL MCQ .";
fbUI.include(t,c,2,b);
//change pasIndex
setFBUI("Feedback - Wrong Again",800,700);
// with fbOKpressed goes to m.next
}

public void pasW2W(short sA){// like pasW5
M.mLCtrl().mcqDlog().setVisible(false);
String t= " Sorry, you are not correct again \n";
String[] c = { "Because :"+ .getCsession().getCurMCQ().answer(sA).
whyWorC()+ "\n", M.getCsession().getCurMCQ().
whyWorCfull()+"\n", M.getCsession().getCurMCQ(). feedl()," "};
if (M.getLearner().knowZ())
c[3]=M.getCsession().getCurMCQ().zFeed();
String b=" Next ==> ANOTHER MCQ, but at SAME LEVEL MCQ ";
fbUI.include(t,c,3,b);
//change pasIndex
setFBUI("Feedback - Wrong Again",800,700);
// with fbOKpressed goes to m.next
}

public void pasW3(short sA){
M.mLCtrl().mcqDlog().setVisible(false);
String t=" Your answer is not correct\n";
String[] c = { "Because :"+
M.getCsession().getCurMCQ().answer(sA). whyWorC()+
"\n",M.getCsession().getCurMCQ(). whyWorCfull() ," "};
if (M.getLearner().knowZ())
c[2]=M.getCsession().getCurMCQ().zFeed();
String b=" Next ==> ANOTHER MCQ, but at SAME LEVEL MCQ ";
fbUI.include(t,c,2,b);
setFBUI("Feedback ",800,700);
// with fbOKpressed goes to m.next
}

public void pasW4(short sA){
M.mLCtrl().mcqDlog().setVisible(false);
String t=" Your answer is not correct\n";
String[] c = { "Because :"+ M.getCsession().getCurMCQ().answer(sA).
whyWorC()+ "\n", M.getCsession().getCurMCQ(). whyWorCfull(),
M.getCsession().getCurMCQ().feedl()," "};
if (M.getLearner().knowZ())
c[3]=M.getCsession().getCurMCQ().zFeed();
String b=" Next ==> ANOTHER, but SAME LEVEL MCQ ";
fbUI.include(t,c,3,b);
}

```



```

setFBUI("Feedback ",800,700);
// with fbOKpressed goes to m.next
}
public void pasW5(short sA){
M.mLCtrl().mcqDlog().setVisible(false);
String t=" Your answer is not correct\n";
String[] c = { "Because :"+ M.getCsession().getCurMCQ().answer(sA).
  whyWorC()+ "\n", M.getCsession().getCurMCQ(). whyWorCfull(),
  M.getCsession().getCurMCQ().feedl()," "};
if (M.getLearner().knowZ())
  c[3]=M.getCsession().getCurMCQ().zFeed();
String b=" Next ==> LOWER LEVEL MCQ ";
fbUI.include(t,c,3,b);
setFBUI("Feedback ",800,700);
// with fbOKpressed goes to m.next
}

public void pasC1(short sA){

  JOptionPane.showMessageDialog(M.mLCtrl().mLUI(),"Weldone,M.getLearner().userID()+ " Answer "+ String.valueOf(sA)+ " is correct. \n Try next question ", "Feedback- You Did It",JOptionPane.OK_OPTION);
  M.fbOKpressed();
  // pretend dialog as FBUI- with fbOKpressed goes to m.next
}

public void pasC2(short sA){
  String t="Weldone "+" You selected answer; "+ String.valueOf(sA)+ ". It is correct \n";
  String[] c={M.getCsession().getCurMCQ().answer(sA).whyWorC()+ "\n"};
  String b=" Next ==> NEXT LEVEL MCQ ";
  fbUI.include(t,c,0,b);
  setFBUI("Feedback - Weldone",800,300);
  // with fbOKpressed goes to m.next
}

public void pasC3(short sA){
  String t="Weldone "+" You selected answer; "+ String.valueOf(sA)+ ". It is correct \n";
  String[] c = { M.getCsession().getCurMCQ().answer(sA).whyWorC()+ "\n", M.getCsession().getCurMCQ().whyWorCfull()," "};
  if (M.getLearner().knowZ())
    c[2]=M.getCsession().getCurMCQ().zFeed();
  String b=" Next ==> NEXT LEVEL MCQ ";
  fbUI.include(t,c,2,b);
  setFBUI("Feedback - Weldone",800,700);
  // with fbOKpressed goes to m.next
}

public void pasC4(short sA){
  String t="Weldone "+" You selected answer; "+ String.valueOf(sA)+ ". It is correct \n";
  String[] c = { M.getCsession().getCurMCQ().answer(sA).whyWorC()+ "\n",M.getCsession().getCurMCQ().whyWorCfull(), M.getCsession().getCurMCQ().feedl()," "};
  if (M.getLearner().knowZ())
    c[3]=M.getCsession().getCurMCQ().zFeed();
  String b=" Next ==> NEXT LEVEL MCQ ";
  fbUI.include(t,c,3,b);
  setFBUI("Feedback - Weldone.",800,700);
  // with fbOKpressed goes to m.next
}

```

```

    }
    public void pasC5(short sA){
    String t="Weldone "+" You selected answer; "+ String.valueOf(sA)+ ".
    It is correct \n";
    String[] c = { M.getCsession().getCurMCQ().answer(sA).whyWorC()+
    "\n", M.getCsession().getCurMCQ().whyWorCfull(),
    M.getCsession().getCurMCQ().feedl(), " "};
    if (M.getLearner().knowZ())
    c[3]=M.getCsession().getCurMCQ().zFeed();
    String b=" Next ==> ANOTHER, but SAME LEVEL MCQ ";
    fbUI.include(t,c,3,b);
    setFBUI("Feedback - Weldone",800,700);
    // with fbOKpressed goes to m.next
    }

    public void refine(){
    String u=M.getCsession().getCurMCQ().jCodeUrl();
    new FileView(u);
    }
}

```

Class FBUI

```

package lozpt1;

import java.awt.*;
import javax.swing.*;
import com.borland.jbcl.layout.*;
import java.awt.event.*;

public class FBUI extends JFrame {
    private FBCtrl fbCtrl;
    private JPanel panell = new JPanel();
    private BorderLayout borderLayout1 = new BorderLayout();
    private JPanel jPanel1 = new JPanel();
    private JPanel jPanel2 = new JPanel();
    private JPanel jPanel3 = new JPanel();
    private JTextArea jTextArea1 = new JTextArea();
    private BorderLayout borderLayout2 = new BorderLayout();
    private JTextArea jTextArea2 = new JTextArea();
    private JPanel jPanel4 = new JPanel();
    private JScrollPane jScrollPane1 = new JScrollPane();
    private JButton OK = new JButton();
    private JButton skip = new JButton();
    private BorderLayout borderLayout3 = new BorderLayout();
    private BorderLayout borderLayout4 = new BorderLayout();
    private FlowLayout flowLayout1 = new FlowLayout();
    private JButton Refine = new JButton();
    private JTextPane jTextPanel = new JTextPane();
    private BorderLayout borderLayout5 = new BorderLayout();
    private JButton LearnM = new JButton();

    public FBUI(FBCtrl f, String title) {
        this.setTitle(title);
        fbCtrl=f;
        try {

```

```

        jbInit();
        pack();
    }
    catch(Exception ex) {
        ex.printStackTrace();
    }
}
private void jbInit() throws Exception {
    panell1.setLayout(borderLayout1);
    jPanell1.setLayout(borderLayout3);
    jPanel3.setLayout(borderLayout2);
    jPanel3.setBorder(BorderFactory.createLoweredBevelBorder());
    jTextArea2.setBackground(new Color(203, 255, 197));
    jTextArea2.setFont(new java.awt.Font("Serif", 0, 25));
    jTextArea2.setForeground(Color.red);
    jTextArea2.setBorder(BorderFactory.createLoweredBevelBorder());
    jTextArea2.setCaretColor(Color.red);
    jPanel2.setLayout(borderLayout4);
    OK.setBackground(Color.orange);
    OK.setFont(new java.awt.Font("Dialog", 1, 12));
    OK.setActionCommand("OK");
    OK.setText("OK");
    OK.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(ActionEvent e) {
            OK_actionPerformed(e);
        }
    });
    skip.setBackground(Color.orange);
    skip.setEnabled(false);
    skip.setActionCommand("skip");
    skip.setText("SKIP");
    skip.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(ActionEvent e) {
            skip_actionPerformed(e);
        }
    });
    jPanel4.setLayout(flowLayout1);
    jPanell1.setBorder(BorderFactory.createRaisedBevelBorder());
    jPanel4.setBackground(new Color(236, 190, 216));
    jPanel4.setBorder(BorderFactory.createRaisedBevelBorder());
    jTextAreal1.setBackground(new Color(158, 177, 197));
    jTextAreal1.setFont(new java.awt.Font("SansSerif", 0, 20));
    jTextAreal1.setForeground(Color.red);
    jTextAreal1.setCaretColor(Color.red);
    Refine.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(ActionEvent e) {
            Refine_actionPerformed(e);
        }
    });
    Refine.setText("JAVA CODE");
    Refine.setActionCommand("skip");
    Refine.setBackground(Color.orange);
    Refine.setFont(new java.awt.Font("Dialog", 1, 12));
    jTextPanel1.setBackground(new Color(255, 249, 175));
    this.getContentPane().setLayout(borderLayout5);
    LearnM.setFont(new java.awt.Font("Dialog", 1, 12));
    LearnM.setBackground(Color.orange);
    LearnM.setActionCommand("skip");
    LearnM.setText("ViewMyRecs");
    LearnM.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(ActionEvent e) {

```

```

        LearnM_actionPerformed(e);
    }
});
this.setDefaultCloseOperation(DO_NOTHING_ON_CLOSE);
getContentPane().add(panell1, BorderLayout.CENTER);
panell1.add(jPanell1, BorderLayout.NORTH);
jPanell1.add(jTextArea1, BorderLayout.CENTER);
panell1.add(jPanel2, BorderLayout.SOUTH);
jPanel2.add(jTextArea2, BorderLayout.CENTER);
jPanel4.add(LearnM, null);
jPanel4.add(Refine, null);
jPanel4.add(skip, null);
jPanel4.add(OK, null);
panell1.add(jPanel3, BorderLayout.CENTER);
jPanel3.add(jScrollPane1, BorderLayout.CENTER);
jScrollPane1.getViewport().add(jTextPanel1, null);
jPanel3.add(jPanel4, BorderLayout.SOUTH);
}

void OK_actionPerformed(ActionEvent e) {
    fbCtrl.OKpressed();
}

void skip_actionPerformed(ActionEvent e) {
    fbCtrl.skipPressed();
}

public void cleanAll(){
    jTextPanel1.setText("");
    jTextArea1.setText("");
    jTextArea2.setText("");
}

public void include(String t, String[] c,int i,String b){
    cleanAll();
    AttributeSet.aSet.setStyle(1,1);
    jTextPanel1.setText("");
    Util.U.appendToTextPane(jTextArea1,t,AttributeSet.aSet.heading3);
    includeC(c,i);
    Util.U.appendToTextPane(jTextArea2,b,AttributeSet.aSet.heading2);
    jTextPanel1.setCaretPosition(1);
}
public void includeC(String[] c, int i){
    Util.U.appendToTextPane(jTextPanel1,c[0],AttributeSet.aSet.para1);
    if (i>0)
Util.U.appendToTextPane(jTextPanel1,"\n"+c[1],AttributeSet.aSet.para2);
    if (i>1)
Util.U.appendToTextPane(jTextPanel1,"\n"+c[2],AttributeSet.aSet.para3);
    if (i>2)
Util.U.appendToTextPane(jTextPanel1,"\n"+c[3],AttributeSet.aSet.para2);
    if (i>3)
Util.U.appendToTextPane(jTextPanel1,"\n"+c[4],AttributeSet.aSet.para2);
}
void Refine_actionPerformed(ActionEvent e) {
    fbCtrl.refine();
}

void LearnM_actionPerformed(ActionEvent e) {
    fbCtrl.getMentor().pls();
}

```

```
}
```

Class FileView

```
package lozpt1;

import java.awt.*;
import java.awt.event.*;
import java.net.*;
import javax.swing.*;

public extends JFrame {
    protected JEditorPane jEditorPane;
    protected BorderLayout BorderLayout1 = new BorderLayout();

    public FileView(String urlString) {
        setTitle("Java Code Browser");
        setDefaultCloseOperation(DISPOSE_ON_CLOSE);
        Container content = getContentPane( );
        content.setLayout(new BorderLayout( ));
        jEditorPane = new JEditorPane( );
        jEditorPane.setEditable(false);
        content.add(new JScrollPane(jEditorPane), BorderLayout.CENTER);
        setSize(500, 600);
        Util.U.setCentre(this);
        setVisible(true);
        openURL(urlString);
        addWindowListener(new WindowAdapter( ) {
            public void windowClosing(WindowEvent e) { }
        });
    }

    protected void openURL(String urlString) {
        try {
            URL url = new URL(urlString);
            jEditorPane.setPage(url);
        }
        catch (Exception e) {
        }
    }
}
```

Class FuzzyMem3L

```
package lozpt1;

public class FuzzyMem3L {
    private short[] level=new short[3];
    private short crisp=0;
    public FuzzyMem3L(short[] l) { //fuzzy set ==> crisp
        level=l;
        crisp=defuzzify();
    }
    public FuzzyMem3L() {
        for (int i=0; i<3; i++)
            level[i]=0;
    }
    public short[] level(){
```

```

    return level;
}
public short level(short i){
    return level[i];
}
public short crisp(){
    return crisp;
}
public short order(){
    short order=0;
    if ((crisp >= 0)&&(crisp <= 20))
        order=1;
    else
        if ((crisp > 20)&&(crisp <= 30))
            order=2;
        else
            if ((crisp >30)&&(crisp <= 70))
                order=3;
            else
                if ((crisp >70)&&(crisp <= 80))
                    order=4;
                else
                    if ((crisp >80)&&(crisp <= 100))
                        order=5;
    return order;
}

public static FuzzyMem3L fuzzify(short x){
    FuzzyMem3L f=new FuzzyMem3L();
    f.crisp=x;
    if ((x>=0) && (x<21))
        f.level[0]=100;
    else
        if (x<41){
            f.level[0]= (short)((40-x)*5);
            f.level[1]=(short) (100-f.level[0]);
        }else
            if (x<61)
                f.level[1]=100;
            else
                if (x<80){
                    f.level[1]=(short)((80-x)*5);
                    f.level[2]= (short)(100-f.level[1]);
                }else
                    if (x<101)
                        f.level[2]=100;
    return f;
}

public short defuzzify(){ //need to use centroid with mirroring
    short i=(short)(level[0]*0.1+ level[1]*0.5+level[2]*0.9);
    return i; // note that inverse not true;
}

public short defuzzMax(){ //
    short max=0;
    if (level[1] >= level[0])
        max=1;
    if (level[2]>=level[max])
        max=2;
    return max; // note that inverse not true;
}

```

```

    }

    public String des(int type){ //type=1;sms/2;D
        String d="";
        switch (this.order()){
            case 1: switch (type){
                    case 1: d= "Very Weak";break;
                    case 2: d= "Very Easy";break;
                }
                break;
            case 2: switch (type){
                    case 1: d= "Weak";break;
                    case 2: d= "Easy";break;
                }
                break;
            case 3: switch (type){
                    case 1: d= "Moderate Level";break;
                    case 2: d= "Medium Level";break;
                }
                break;
            case 4: switch (type){
                    case 1: d= "Strong";break;
                    case 2: d= "Hard";break;
                }
                break;
            case 5: switch (type){
                    case 1: d= "Very Strong";break;
                    case 2: d= "Very Hard";break;
                }
                break;
        }
        return d;
    }
}

```

Class FuzzyMem5L

```
package lozpt1;
```

```

public class FuzzyMem5L {
    private short crisp=0;
    private short[] level=new short[5];
    public FuzzyMem5L(short[] l) {
        level=l;
    }
    //public static FuzzyMem5L fuzzify(short x){} - not necessary now
    //public short defuzzify(){

    public short defuzzMax(){
        short maxI=0;
        for (short j=1;j<5;j++)
            if (level[j]>level[maxI])
                maxI =j;
        return maxI;
    }

    public short[] level(){
        return level;
    }
}

```

```

    public short level(short i){
        return level[i];
    }
}

```

Class FuzzyMX3and3to3

```
package lozpt1;
```

```

public class FuzzyMX3and3to3 {
    //public static FuzzyMXpas fMXpas= new FuzzyMXpas();
    private static short[][] LLA = {{1,1,2},{1,2,2},{2,3,3}};
        //GLA*T ==> LLA
    private static short[][] L = {{1,1,2},{1,2,2},{2,3,3}}; //LLA*T ==>
L
    private static short[][] SMS = {{1,1,2},{1,2,3},{2,3,3}};
        //Dynamic SMS. SMS*L ==> SMS
    private static short[][] P2 = {{3,1,1},{3,3,1},{3,3,3}};
        // SMS*D ==> P2; 3-correct
    private static short[][] P3 = {{3,2,1},{3,2,2},{3,3,2}};
        //SMS*D ==> P3

    public FuzzyMX3and3to3() { }

    public static FuzzyMem3L applyRuleLLA(FuzzyMem3L C1, FuzzyMem3L C2){
        return applyRule(C1,C2,LLA);
    }
    public static FuzzyMem3L applyRuleL(FuzzyMem3L C1, FuzzyMem3L C2){
        return applyRule(C1,C2,L);
    }
    public static FuzzyMem3L applyRuleSMS(FuzzyMem3L C1, FuzzyMem3L C2){
        return applyRule(C1,C2,SMS);
    }
    public static FuzzyMem3L applyRuleP3(FuzzyMem3L C1, FuzzyMem3L C2){
        return applyRule(C1,C2,P3);
    }
    public static FuzzyMem3L applyRuleP2(FuzzyMem3L C1, FuzzyMem3L C2){
        return applyRule(C1,C2,P2);
    }
    private static FuzzyMem3L applyRule(FuzzyMem3L C1, FuzzyMem3L C2,
        short[][] mx){
        short[] l = new short[3];
        short minP;
        for (short i=0;i<3;i++)
            l[i]=0;
        for (short i=0; i<3;i++){
            for (short j=0; j<3;j++){
                minP= (short) Util.U.min(C1.level(i),C2.level(j));
                if (minP > l[mx[i][j]-1])
                    l[mx[i][j]-1]= minP; //get maximum
            }
        }
        return new FuzzyMem3L(l);
    }
}

```

Class FuzzyMXpas

```
package lozpt1;
```



```

public class FuzzyMXpas {
    //public static FuzzyMXpas fMXpas= new FuzzyMXpas();
    //can be considered as 3*3
    private static short[][] mxW = {{3,4,5},{2,3,4},{1,2,3}};
    private static short[][] mxC = {{3,4,5},{2,3,4},{1,2,3}};
    public FuzzyMXpas() { }
    public static FuzzyMem5L applyRule(FuzzyMem3L SMS, FuzzyMem3L D,
boolean correct){
        short[] l = new short[5];
        short minP;
        for (short i=0;i<5;i++)
            l[i]=0;
        short[][] mx=mxW;
        if (correct) mx=mxC;
        for (short i=0; i<3;i++)
            for (short j=0; j<3;j++){
                minP= (short) Util.U.min(SMS.level(i),D.level(j));
                if (minP > l[mx[i][j]-1])
                    l[mx[i][j]-1]= minP;
            }
        return new FuzzyMem5L(l);
    }
}

```

Class GlobalLA

```

package lozpt1;

public {
    private short gla;
    private FuzzyMem3L glaFZ=null;
    private short dynT=90; //dynamic change probability for true to
true
    private short dynF=10;
    private float affirm= (float)2.7;
        //mental state affirms local learn ability
    private float deny=(float)0.29;
    public GlobalLA(short l) {
        //crisp ==> fuzzy set; but crisp in glaFZ may not be equal to gla
        gla=l;
        glaFZ=FuzzyMem3L.fuzzify(l);
    }
    public GlobalLA(){
    }

    public short get(){
        return gla;
    }
    public FuzzyMem3L getFZ(){
        return glaFZ;
    }
    public void set(short g){
        gla=g;
        glaFZ=FuzzyMem3L.fuzzify(g);
    }

    public void predictGLA (){
        gla=(short) (dynT*gla+dynF*(1-gla)/100);
    }
}

```

```

    }

    public void updateGLA(boolean t){
        float ohNOe;
        if (t)
            ohNOe= ((float)100/gla)*affirm;
        else
            ohNOe= ((float)100/gla)*deny;
        gla=(short)((ohNOe*100)/(1+ohNOe));
    }
}

```

Class LastLearnRecord

```

package lozpt1;

public {
    private short leaRecID; //same as leaID, can omit later
    private short leaID;
    private short lastSessID;
    private String lastSummary; //this is short summary of lastSession
    private short lastMCID;
    private MainConcept mc=null;
    private short lastTopID;
    private short lastState=0; // 0- no idea
    private short lastAction=0; // 0- no idea; start afresh
    private short completed;
        //0- no, repeat; 1-not sure, prompt;2 -yes, next action

    public LastLearnRecord(short lRecID, short lID, short lsID, short
        lCptID, short ltopID, short lSt, short lAct, short c, String s) {
        leaRecID=leaID; lastState=lSt; leaID=lID; completed=c;
        lastSummary=s; lastAction=lAct; lastSessID=lsID;lastMCID=
        lCptID; lastTopID=ltopID;
    }
    public MainConcept getLastMC(){
        if (mc==null)
            mc=MainConceptDB.mcDB().findaMainConcept(lastMCID);
        return mc;
    }
    public Topic getLastTop(){
        return TopicDB.toDB().findaTopic(lastTopID);
    }
    public short getLState(){
        return lastState;
    }
    public short getLAction(){
        return lastAction;
    }
    public String getLSummary(){
        return lastSummary;
    }
    public Session getLSession(){
        return SessionDB.ssDB().findaSession(lastSessID);
    }
    public short getLSessID(){
        return lastSessID;
    }
}

```

```

public String nextLesson(){
    String lesson=" ";
    if (lastState==0)
        lesson=getLastMC().getHead();
    return lesson;
}
}

```

Class LastLearnRecordDB

```

package lozpt1;

import com.borland.dx.dataset.*;
import com.borland.dx.sql.dataset.*;
import java.sql.*;

public class LastLearnRecordDB extends LozDB{
    private static LastLearnRecordDB lrDB = new LastLearnRecordDB();

    public static LastLearnRecordDB lrDB(){
        return lrDB;
    }
    public LastLearnRecord findLastLearnRecord(short llRID){
        if (llRID==0)
            return null;
        QueryDataSet getLeaQ = new QueryDataSet();
        QueryDescriptor qD = new QueryDescriptor(LOZdB, "SELECT
        \"LearnRecords\".*"+ " FROM\"LearnRecords\" WHERE
        \"LearnRecords\".\"lastLeaRdID\"=\"" + llRID, null, false,
        Load.ALL);
        getLeaQ.setQuery(qD);
        getLeaQ.executeQuery();
        LOZdB.closeConnection();
        if (getLeaQ.isEmpty())
            return null;
        LastLearnRecord lr=new LastLearnRecord(llRID, getLeaQ.getShort(2),
        getLeaQ.getShort(3), getLeaQ.getShort(4), getLeaQ.getShort(5),
        getLeaQ.getShort(6),getLeaQ.getShort(7),getLeaQ.getShort(8),getL
        eaQ.getString(9));
        getLeaQ.closeStatement();
        return lr;
    }
}

```

Class LeaInsUI

```

package lozpt1;

import javax.swing.*;
import java.awt.*;
import com.borland.jbcl.layout.*;
import java.awt.event.*;

public class LeaInsUI extends JFrame{

```

```

private Mentor m;
private JPanel contentPane;
private JPanel jPanel2 = new JPanel();
private JPanel jPanel3 = new JPanel();
private JButton jButton5 = new JButton();
private JButton OK = new JButton();
private FlowLayout flowLayout1 = new FlowLayout();
private JLabel jLabel11 = new JLabel();
private JPanel jPanel11 = new JPanel();
private JLabel jLabel2 = new JLabel();
private XYLayout xYLayout1 = new XYLayout();
private JLabel jLabel3 = new JLabel();
private JLabel jLabel4 = new JLabel();
private JTextField gla0 = new JTextField();
private JTextField lla0 = new JTextField();
private JTextField sms0 = new JTextField();
private JTextField glaN = new JTextField();
private JTextField llaN = new JTextField();
private JTextField smsN = new JTextField();
private JLabel jLabel5 = new JLabel();
private JLabel jLabel6 = new JLabel();
private JLabel jLabel7 = new JLabel();
private JButton jButton6 = new JButton();
private JSlider glaS = new JSlider();
private JSlider llaS = new JSlider();
private JSlider smsS = new JSlider();
private JLabel jLabel8 = new JLabel();
private JTextField sms01 = new JTextField();
private JLabel jLabel9 = new JLabel();
private JLabel jLabel10 = new JLabel();
private JTextField sms02 = new JTextField();
private JTextField smsN1 = new JTextField();
private JSlider pasS = new JSlider();
private BorderLayout borderLayout1 = new BorderLayout();
private XYLayout xYLayout2 = new XYLayout();
private JLabel jLabel111 = new JLabel();
private JPanel jPanel4 = new JPanel();
private JPanel jPanel9 = new JPanel();
private JPanel jPanel10 = new JPanel();
private JLabel jLabel112 = new JLabel();
private JPanel jPanel5 = new JPanel();
private JLabel jLabel113 = new JLabel();
private JPanel jPanel111 = new JPanel();
private JPanel jPanel112 = new JPanel();
private JPanel jPanel6 = new JPanel();
private JLabel jLabel114 = new JLabel();
private JPanel jPanel113 = new JPanel();
private JPanel jPanel114 = new JPanel();
private JPanel jPanel115 = new JPanel();
private JPanel jPanel116 = new JPanel();
private JPanel jPanel7 = new JPanel();
private JLabel jLabel115 = new JLabel();
private JPanel jPanel8 = new JPanel();
private JLabel jLabel116 = new JLabel();
private JPanel jPanel117 = new JPanel();
private JPanel jPanel118 = new JPanel();
private JLabel jLabel117 = new JLabel();
private JLabel jLabel118 = new JLabel();

public LeaInsUI(Mentor m ) {
    this.m=m;

```

```

    try {
        jbInit();
    }
    catch(Exception e) {
        e.printStackTrace();
    }
}

private void jbInit() throws Exception {
    contentPane = (JPanel) this.getContentPane();
    setTitle("LOZ- VIEW & CHANGE THE LEARNER MODEL");
    contentPane.setLayout(borderLayout1);
    jPanel3.setLayout(xYLayout2);
    jPanel2.setLayout(flowLayout1);
    jButton5.setFont(new java.awt.Font("Monospaced", 1, 14));
    jButton5.setForeground(SystemColor.textHighlight);
    jButton5.setText("CANCEL");
    jButton5.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(ActionEvent e) {
            jButton5_actionPerformed(e);
        }
    });
    OK.setFont(new java.awt.Font("Monospaced", 1, 14));
    OK.setForeground(SystemColor.textHighlight);
    OK.setText("OK");
    OK.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(ActionEvent e) {
            OK_actionPerformed(e);
        }
    });
    jPanel3.setBackground(new Color(118, 249, 216));
    jPanel3.setMinimumSize(new Dimension(400, 30));
    jPanel3.setPreferredSize(new Dimension(400, 50));
    jPanel1.setBackground(new Color(236, 255, 216));
    jPanel1.setBorder(BorderFactory.createRaisedBevelBorder());
    jPanel1.setLayout(xYLayout1);
    jLabel2.setFont(new java.awt.Font("Serif", 1, 12));
    jLabel2.setText("Global Learn Ability");
    jLabel3.setFont(new java.awt.Font("Serif", 1, 12));
    jLabel3.setText("Local Learn Ability");
    jLabel4.setFont(new java.awt.Font("Serif", 1, 12));
    jLabel4.setText("Strength -SMS");
    jLabel5.setText("Current");
    jLabel5.setFont(new java.awt.Font("Serif", 1, 10));
    jLabel5.setForeground(Color.magenta);
    jLabel6.setFont(new java.awt.Font("Serif", 1, 10));
    jLabel6.setForeground(Color.magenta);
    jLabel6.setText("New");
    jPanel2.setBackground(new Color(46, 233, 216));
    sms0.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(ActionEvent e) {
            sms0_actionPerformed(e);
        }
    });
    jLabel1.setFont(new java.awt.Font("Dialog", 0, 20));
    jLabel1.setForeground(Color.red);
    jLabel7.setFont(new java.awt.Font("Serif", 1, 12));
    jLabel7.setForeground(new Color(255, 0, 158));
    jLabel7.setText("YOU MAY CHANGE THE VALUES :");
    jButton6.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(ActionEvent e) {

```

```

        jButton6_actionPerformed(e);
    }
});
jButton6.setText("HELP");
jButton6.setForeground(SystemColor.textHighlight);
jButton6.setFont(new java.awt.Font("Monospaced", 1, 14));
jLabel8.setText("Difficulty - MCQ");
jLabel8.setFont(new java.awt.Font("Serif", 1, 12));
sms01.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(ActionEvent e) {
        sms01_actionPerformed(e);
    }
});
jLabel9.setText("<= Cannot be changed.");
jLabel9.setForeground(Color.magenta);
jLabel9.setFont(new java.awt.Font("Serif", 1, 15));
jLabel10.setFont(new java.awt.Font("Serif", 1, 12));
jLabel10.setText("Strength - PAS");
contentPane.setMinimumSize(new Dimension(598, 400));
contentPane.setPreferredSize(new Dimension(598, 400));
sms02.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(ActionEvent e) {
        sms02_actionPerformed(e);
    }
});
jLabel11.setText("Select required PAS below, and SMS will be
    modified accordingly.");
jLabel11.setForeground(new Color(255, 0, 158));
jLabel11.setFont(new java.awt.Font("Serif", 1, 12));
sms01.setText("55");
sms02.setText("2");
glaS.setExtent(1);
llaS.setExtent(1);
smsS.setExtent(1);
pasS.setExtent(1);
pasS.setMaximum(5);
pasS.setMinimum(1);
jPanel4.setBackground(Color.pink);

jPanel10.setBackground(UIManager.getColor("RadioButtonMenuItem.selecti
onBackground"));
jPanel9.setBackground(Color.green);
jLabel12.setFont(new java.awt.Font("Dialog", 0, 10));
jLabel12.setText("high=100");
jPanel5.setBackground(Color.pink);
jLabel13.setText("high=45");
jLabel13.setFont(new java.awt.Font("Dialog", 0, 10));
jPanel11.setBackground(Color.green);

jPanel12.setBackground(UIManager.getColor("RadioButtonMenuItem.selecti
onBackground"));
jPanel6.setBackground(Color.pink);
jLabel14.setText("high=75");
jLabel14.setFont(new java.awt.Font("Dialog", 0, 10));
jPanel13.setBackground(Color.green);

jPanel14.setBackground(UIManager.getColor("RadioButtonMenuItem.selecti
onBackground"));

jPanel15.setBackground(UIManager.getColor("RadioButtonMenuItem.selecti
onBackground"));

```

```

jPanel16.setBackground(Color.green);
jPanel7.setBackground(Color.pink);
jLabel15.setFont(new java.awt.Font("Dialog", 0, 10));
jLabel15.setText("medium=25");
jPanel8.setBackground(Color.pink);
jLabel16.setText("high=100");
jLabel16.setFont(new java.awt.Font("Dialog", 0, 10));
jPanel17.setBackground(Color.green);

jPanel18.setBackground(UIManager.getColor("RadioButtonMenuItem.selecti
onBackground"));
jLabel17.setText("medium=100");
jLabel17.setFont(new java.awt.Font("Dialog", 0, 10));
jLabel18.setFont(new java.awt.Font("Dialog", 0, 10));
jLabel18.setText("medium=55");
smsN.setText("75");
jPanel2.add(jButton6, null);
jPanel2.add(jButton5, null);
jPanel2.add(OK, null);
contentPane.add(jPanel1, BorderLayout.CENTER);
jPanel1.add(gla0, new XYConstraints(227, 24, 24, 25));
jPanel1.add(jLabel2, new XYConstraints(11, 20, 122, 28));
jPanel1.add(jLabel3, new XYConstraints(11, 54, 122, 28));
jPanel1.add(lla0, new XYConstraints(227, 51, 24, 25));
jPanel1.add(llaS, new XYConstraints(249, 59, 110, 16));
jPanel1.add(jLabel10, new XYConstraints(11, 178, 122, 28));
jPanel1.add(smsO2, new XYConstraints(227, 174, 24, 25));
jPanel1.add(pasS, new XYConstraints(249, 182, 110, 16));
jPanel1.add(jLabel8, new XYConstraints(11, 144, 122, 28));
jPanel1.add(jLabel4, new XYConstraints(11, 112, 122, 28));
jPanel1.add(smsS, new XYConstraints(249, 115, 110, 16));
jPanel1.add(smsO, new XYConstraints(227, 111, 24, 25));
jPanel1.add(smsO1, new XYConstraints(227, 144, 24, 25));
jPanel1.add(glaS, new XYConstraints(249, 30, 110, 16));
jPanel1.add(glaN, new XYConstraints(359, 30, 26, 24));
jPanel1.add(llaN, new XYConstraints(359, 57, 26, 24));
jPanel1.add(smsN, new XYConstraints(359, 114, 26, 24));
jPanel1.add(smsN1, new XYConstraints(359, 179, 26, 24));
jPanel1.add(jLabel6, new XYConstraints(358, 0, 37, 28));
jPanel1.add(jLabel5, new XYConstraints(220, 0, -1, 28));
jPanel1.add(jLabel9, new XYConstraints(256, 140, 184, 28));
jPanel1.add(jPanel13, new XYConstraints(393, 129, 26, 7));
jPanel1.add(jPanel14, new XYConstraints(418, 123, 26, 13));
jPanel1.add(jPanel6, new XYConstraints(444, 107, 26, 29));
jPanel1.add(jLabel12, new XYConstraints(166, 0, 46, 18));
jPanel1.add(jLabel16, new XYConstraints(162, 41, 46, 18));
jPanel1.add(jPanel4, new XYConstraints(177, 16, 26, 27));
jPanel1.add(jPanel10, new XYConstraints(153, 37, 26, 6));
jPanel1.add(jPanel9, new XYConstraints(132, 36, 26, 7));
jPanel1.add(jLabel17, new XYConstraints(149, 132, 71, 18));
jPanel1.add(jPanel15, new XYConstraints(161, 149, 26, 31));
jPanel1.add(jPanel7, new XYConstraints(187, 174, 26, 6));
jPanel1.add(jPanel5, new XYConstraints(178, 115, 26, 20));
jPanel1.add(jPanel12, new XYConstraints(152, 113, 26, 22));
jPanel1.add(jPanel11, new XYConstraints(128, 128, 26, 7));
jPanel1.add(jPanel16, new XYConstraints(134, 172, 26, 7));
jPanel1.add(jLabel13, new XYConstraints(178, 98, 46, 18));
jPanel1.add(jPanel8, new XYConstraints(177, 56, 26, 27));
jPanel1.add(jPanel18, new XYConstraints(155, 77, 26, 6));
jPanel1.add(jPanel17, new XYConstraints(129, 76, 26, 7));
jPanel1.add(jLabel11, new XYConstraints(9, 80, 369, 24));

```

```

jPanel1.add(jLabel18, new XYConstraints(107, 97, 71, 18));
jPanel1.add(jLabel15, new XYConstraints(384, 106, 59, 18));
jPanel1.add(jLabel14, new XYConstraints(429, 89, 46, 18));
contentPane.add(jPanel3, BorderLayout.NORTH);
jPanel3.add(jLabel11, new XYConstraints(0, 0, 501, 31));
jPanel3.add(jLabel7, new XYConstraints(221, 28, 201, 24));
//jPanel3.add(jLabel2, new XYConstraints(244, 2, 124, 42));
//jPanel3.add(jLabel3, new XYConstraints(385, 2, 411,
42));
contentPane.add(jPanel2, BorderLayout.SOUTH);
int n;
pasS.setValue(2);
try {
    n=m.getLearner().gla().get();
    glaO.setText(Integer.toString(n));
    glaS.setValue(n);
}
catch (java.lang.NullPointerException e) {
    glaO.setText(Integer.toString(0));
}
try {
    n=m.getCsession().getCurLMC().getLLA().get();
    llaO.setText(Integer.toString(n));
    llaS.setValue(n);
}
catch (java.lang.NullPointerException e) {
    llaO.setText(Integer.toString(0));
}
try {
    n=m.getCsession().getCurLMS().getSMS().get();
    smsO.setText(Integer.toString(n));
    smsS.setValue(n);
}
catch (java.lang.NullPointerException e) {
    smsO.setText(Integer.toString(0));
}
}

void OK_actionPerformed(ActionEvent e) {
    short s=0;
    if ((glaN.getText().length()>0)&&
        (llaN.getText().length()>0) &&
        (smsN.getText().length()>0)) {
        try {
            s = (short)Integer.parseInt(glaN.getText());
        }
        catch (NumberFormatException e1) {
        }
        if (s>0)
            m.getLearner().gla().set(s);
        s = (short)Integer.parseInt(llaN.getText());
        if (s>0)
            m.getCsession().getCurLMC().getLLA().setLLA(s);
        s = (short)Integer.parseInt(smsN.getText());
        if (s>0)
            m.getCsession().getCurLMS().getSMS().set(s);
        }
    this.dispose();
}

void jButton5_actionPerformed(ActionEvent e) {

```



```

    this.dispose();
}

public void presentWelcome(String w1){
    jLabel1.setText(w1);
}

void sms0_actionPerformed(ActionEvent e) {

}

void jButton6_actionPerformed(ActionEvent e) {

}

void sms01_actionPerformed(ActionEvent e) {

}

void sms02_actionPerformed(ActionEvent e) {

}

}
}
}

```

Class LearnConcept

```

package lozpt1;

public class LearnConcept extends BasicConcept{
    private static short maxMS =2;
    private Notation notation;
    private Example example;
    private short[] msIDs = new short[maxMS]; //redundant
    private MentalState[] ms = new MentalState[maxMS+1];
    private short ooUmlCptID;
    private short addiCptID;
    private short zCptID;
    public LearnConcept(short lcID, short lcTy, String h, String wt,
        String wy, String hw, short miT, short mxT, short mS, String nt,
        String nft, String et, String eft, short oo, short ad, short z,
        short[] msid) {
        super(lcID,lcTy, h,wt,wy,hw,miT,mxT,mS);
        notation= new Notation(nt,nft);example= new Example(et, eft);
        for (short i=0; i<maxMS; i++ ) {
            msIDs[i] = msid[i]; ms[i]=null;
        }
        ooUmlCptID=oo; addiCptID=ad; zCptID=z;
    }
    public Notation getNotation(){
        return notation;
    }
    public Example getExample(){
        return example;
    }
    public static short maxMS(){
        return maxMS;
    }
    public MentalState getMS(short i){
        if (ms[i-1]==null)
            if (i==1)
                ms[i-1]=this.getBasicMS();
            else
                ms[i-1]= MentalStateDB.msDB().findaMentalState(msIDs[i-2]);
    }
}

```

```

    return ms[i-1];
}
public BasicConcept getOOUmlCpt(){
    return BasicConceptDB.bcDB().findaBasicConcept(ooUmlCptID);
}
public BasicConcept getAddiCpt(){
    return BasicConceptDB.bcDB().findaBasicConcept(addiCptID);
}
public BasicConcept getZCpt(){
    return BasicConceptDB.bcDB().findaBasicConcept(zCptID);
}
public String getLessSum(){
    String s = "The current learning concept is << "+
    this.getHead() + " >>\n";
    s+=" ==> There are " + Integer.toString(maxMS+1)+ " Mental
    States associated with this concept. \n";
    for (int i=0; i<maxMS+1; i++) {
        s+= "    Stage (" +Integer.toString(i+1)+") : ## " +
        this.getMS((short)(i+1)).describe() + " ## \n";
    }
    return s;
}
}
}

```

Class LearnConceptDB

```

package lozpt1;

import com.borland.dx.dataset.*;
import com.borland.dx.sql.dataset.*;
import java.sql.*;

public class LearnConceptDB extends LozDB{
    private static LearnConceptDB lcDB = new LearnConceptDB();
    public LearnConceptDB() {
    }
    public static LearnConceptDB lcDB(){
        return lcDB;
    }

    public LearnConcept findaLearnConcept(short lcID){
        short type=1;
        LearnConcept lc = (LearnConcept)Cache.find(new Short(lcID),type);
        if (lc==null){
            lc=findINdb(lcID);
            Cache.put(new Short(lcID), lc,type);
        }
        return lc;
    }

    public LearnConcept findINdb(short lcID){
        QueryDataSet getBasQ = new QueryDataSet();
        QueryDescriptor qD = new QueryDescriptor(LOZdB, "SELECT
        \"BasicConcept\".*"+      " FROM \"BasicConcept\" WHERE
        \"BasicConcept\".\"bcID\" = " + lcID , null, false, Load.ALL);
        getBasQ.setQuery(qD);
        getBasQ.executeQuery();
        if (getBasQ.isEmpty())
            return null;
        QueryDataSet getLeaQ = new QueryDataSet();
    }
}

```

```

    QueryDescriptor qD1 = new QueryDescriptor(LOZdB, "SELECT
        \"LearnConcept\".*\"+ \" FROM \"LearnConcept\" WHERE
        \"LearnConcept\".\"lcID\" = \" + lcID , null, false, Load.ALL);
    getLeaQ.setQuery(qD1);
    getLeaQ.executeQuery();
    LOZdB.closeConnection();
    if (getLeaQ.isEmpty())
        return null;
    short maxMS=LearnConcept.maxMS();
    short mSIDs[] = new short[maxMS];
    for (short i=0; i<maxMS ;i++){
        mSIDs[i]= getLeaQ.getShort(9+i);
    }
    LearnConcept lc= new
    LearnConcept(lcID,getBasQ.getShort(2),getBasQ.getString(3),
        getBasQ.getString(4), getBasQ.getString(5),getBasQ.getString(6),
        getBasQ.getShort(7), getBasQ.getShort(8), etBasQ.getShort(9),
        getLeaQ.getString(2),getLeaQ.getString(3),getLeaQ.getString(4),
        getLeaQ.getString(5),getLeaQ.getShort(6),getLeaQ.getShort(7),
        getLeaQ.getShort(8), mSIDs);
    getBasQ.closeStatement();
    getLeaQ.closeStatement();
    return lc;
}
}

```

Class Learner

```
package lozpt1;
```

```

public class Learner {
    private static short lastLeaID;
    private short lastSessID;
    private short leaID;
    private String userID;
    private String passWD;
    private LastLearnRecord lastLRd=null;
    private LearnerOtherRecord leaOthRd=null;
    private GlobalLA gla=new GlobalLA();
    private boolean knowZ;
    private LearnerPreference lp=null;
    public Learner(short lID, String uID, String pW, short ltleRID,
        short leRID, short la, boolean k,short lpID, short lsID) {
        leaID=lID; userID=uID; passWD=pW; gla.set(la); knowZ=k;
        lastSessID=lsID;
        lastLRd = LastLearnRecordDB.lrDB().findaLastLearnRecord(ltleRID);
        lp=LearnerPreferenceDB.lpDB().findaLearnerPreference(lpID);
        leaOthRd=LearnerOtherRecordDB.lOrDB().findaLearnerOtherRecord(leRID);
    }

    public Learner(short lID, String uID, String pW, short la,boolean k,
        short z, short m, short o, short u) { //creating new learner
        leaID=lID; userID=uID; passWD=pW; gla.set(la); knowZ=k;
        lastSessID=0;
        lastLRd = null; lp=new LearnerPreference(lID,true,true,k);
        leaOthRd= new LearnerOtherRecord(lID,lID,z,m,o,u);
    }
    public boolean pwOK(char[] pw){

```

```

    String pws= new String(pw);
    return (passWD.equals(pws));
}
public short leaID(){
    return leaID;
}
public String userID(){
    return userID;
}
public GlobalLA gla(){
    return gla;
}
public boolean knowZ(){
    return knowZ;
}
public LastLearnRecord getLastLearnRecord(){
    return lastLRd;
}
public LearnerPreference getLastLP(){
    return lp;
}
public void setLastSessID(short sid){
    lastSessID=sid;
}
public short getNextSessID(){
    return ++lastSessID;
}
public static void setLastLeaID(short sid){
    lastLeaID=sid;
}
public static short getNextLeaID(){
    return ++lastLeaID;
}
public LearnerOtherRecord getLearnerOtherRecord(){
    return leaOthRd;
}
}
}

```

Class LearnerBC

```

package lozpt1;

public class LearnerBC {
    private Learner l;
    private BasicConcept bc;
    private short[] bcSs= new short[1]; //assume only one
    SessionBC/LC/MC/EC as a sub session
    public LearnerBC(short lid, short bcid) {
        l=LearnerDB.lDB().findaLearner(lid);
        bc=BasicConceptDB.bcDB().findaBasicConcept(bcid);
        bcSs[0]=0;
    }
    public LearnerBC(Learner l, BasicConcept bc){
        this.l=l;this.bc=bc;
        bcSs[0]=0;
    }
    public LearnerBC() {
        bcSs[0]=0;
    }
}

```

```

public void setL(Learner l){
    this.l=l;
}
public void setBC(BasicConcept b){
    bc=b;
}
}

```

Class LearnerDB

```

package lozpt1;

import com.borland.dx.dataset.*;
import com.borland.dx.sql.dataset.*;
import java.sql.*;

public extends LozDB{
    private static LearnerDB lDB = new LearnerDB();
    public LearnerDB() {
        try {
            jbInit();
        }
        catch(Exception e) {
            e.printStackTrace();
        }
    }
    private void jbInit() throws Exception {
        LOzdB.setConnection(connD);
    }
    public static LearnerDB lDB(){
        return lDB;
    }

    public Learner findaLearnerUID(String uID){
        short type=7;
        Learner l = (Learner)Cache.find(uID,type);
        if (l==null){
            l=findINdb(uID);
            Cache.put(new String(uID), l,type);
        }
        return l;
    }

    public Learner findINdb(String uID){ // uID is a String
        QueryDataSet getLeaQ = new QueryDataSet();
        QueryDescriptor qD = new QueryDescriptor(LOzdB, "SELECT
        \"LEARNER\".*" + " FROM LEARNER WHERE LEARNER.\"userID\"= '" +
uID
        + "'", null, false, Load.ALL);
        getLeaQ.setQuery(qD);
        getLeaQ.executeQuery();
        LOzdB.closeConnection();
        if (getLeaQ.isEmpty())
            return null;
        Learner l= new
Learner(getLeaQ.getShort(1),uID,getLeaQ.getString(3),
getLeaQ.getShort(4), getLeaQ.getShort(5),getLeaQ.getShort(6),
getLeaQ.getBoolean(7),getLeaQ.getShort(8), getLeaQ.getShort(9));

```

```

        getLeaQ.closeStatement();
        return l;
    }

    public Learner findALearner(short lID){
        QueryDataSet getLeaQ = new QueryDataSet();
        QueryDescriptor qD = new QueryDescriptor(LOZdB, "SELECT \"LEARNER\
        *.*\" + \" FROM LEARNER WHERE LEARNER.\"leaid\"=\"" + lID, null,
            false, Load.ALL);
        getLeaQ.setQuery(qD);
        getLeaQ.executeQuery();
        LOZdB.closeConnection();
        if (getLeaQ.isEmpty())
            return null;
        getLeaQ.closeStatement();
        return new Learner(lID,getLeaQ.getString(2), getLeaQ.getString(3),
            getLeaQ.getShort(4), getLeaQ.getShort(5), getLeaQ.getShort(6),
            getLeaQ.getBoolean(7),getLeaQ.getShort(8),getLeaQ.getShort(9));
    }

    /*
    public Learner putALearner(Learner newL){
        QueryDataSet getLeaQ = new QueryDataSet();
        QueryDescriptor qD = new QueryDescriptor(LOZdB, "INSERT INTO
            \"LEARNER\"\" +
            \"({ VALUES (\" NewL.lid, \"+)\" + lID, null, false, Load.ALL);
        getLeaQ.setQuery(qD);
        getLeaQ.executeQuery();
        LOZdB.closeConnection();
        putALearnerP;
        putALearnerOthRd;
    }*/
}

```

Class LearnerLC

```

package lozpt1;

public class LearnerLC extends LearnerBC{
    private short[] LearnerMCQs=new short[1]; //assume only one now
    public LearnerLC(short lid, short bcid) {
        super(lid,bcid);
    }
    public void setLeaMCQ(short leaMcq, int i){
        LearnerMCQs[i]=leaMcq;
    }
}

```

Class LearnerMC

```

package lozpt1;

public extends LearnerBC{

```

```

private LocalLA lla=null;
private short[] LearnerLCs;
public LearnerMC(short lid, short bcid) {
    super(lid,bcid);
}
public LearnerMC(short lla) {
    this.lla= new LocalLA(lla);
}
public LearnerMC() {
}

public static LearnerMC findORcreateLMC(Learner L, MainConcept mc){
    LearnerMC lmc= new
LearnerMCDB().lmcDB().findaLearnerMCDB(L.leaID(),
    mc.getBCID());
    lmc.setL(L);lmc.setBC(mc);
    //if new learner, lla will be null o/w some value from DB.
    return lmc;
}

public void setLeaLC(short leaLc, int i){
    LearnerLCs[i]=leaLc;
}
public LocalLA getLLA(){
    return lla;
}
public void setLLA(LocalLA la){
    lla=la;
}
}

```

Class LearnerMCDB

```

package lozpt1;

import com.borland.dx.dataset.*;
import com.borland.dx.sql.dataset.*;
import java.sql.*;

public class LearnerMCDB extends LozDB{
    private static LearnerMCDB lmcDB = new LearnerMCDB();
    public LearnerMCDB() {
        try {
            jbInit();
        }
        catch(Exception e) {
            e.printStackTrace();
        }
    }
    private void jbInit() throws Exception {
        LOZdB.setConnection(connD);
    }
    public static LearnerMCDB lmcDB(){
        return lmcDB;
    }

    public LearnerMC findaLearnerMCDB(short lID,short mcID){
        LearnerMC lmc;

```

```

QueryDataSet getLeaQ = new QueryDataSet();
QueryDescriptor qD = new QueryDescriptor(LOZdB, "SELECT
  \"LearnerMC\".*" + " FROM \"LearnerMC\" WHERE
  \"LearnerMC\".\"leaID\"=\"" + lID + " AND \"LearnerMC\".\"mcID\"
="
  + mcID, null, false, Load.ALL);
getLeaQ.setQuery(qD);
getLeaQ.executeQuery();
LOZdB.closeConnection();
if (getLeaQ.isEmpty())
  lmc=new LearnerMC(); // different from others;
else
  lmc=new LearnerMC(getLeaQ.getShort(3));
getLeaQ.closeStatement();
return lmc;
}
}

```

Class LearnerMCQ

```

package lozpt1;

public class LearnerMCQ {
  private Learner L;
  private MCQ mcq;
  private StrengthPA pas=null;
  private short[] mcqSs; //assume only one SessionMCQ as a sub session
  in a session.

  public LearnerMCQ(short p) {
    pas= new StrengthPA(p);
  }

  public LearnerMCQ() {
  }

  public static LearnerMCQ findORcreateLMCQ(Learner L, MCQ mcq){
    LearnerMCQ lmq= new
LearnerMCQDB().lmqDB().findaLearnerMCQDB(L.leaID(),mcq.mcqID());
    lmq.setL(L);lmq.setMCQ(mcq);
    return lmq;
  }
  public StrengthPA getPAS(){
    return pas;
  }
  public void setPAS(StrengthPA p){
    pas=p;
  }
  public void setL(Learner L){
    this.L=L;
  }
  public void setMCQ(MCQ m){
    mcq=m;
  }
}

```

Class LearnerMCQDB

```

package lozpt1;

```



```

import com.borland.dx.dataset.*;
import com.borland.dx.sql.dataset.*;
import java.sql.*;

public class LearnerMCQDB extends LOzDB{
    private static LearnerMCQDB lmqDB = new LearnerMCQDB();
    public LearnerMCQDB() {
        try {
            jbInit();
        }
        catch(Exception e) {
            e.printStackTrace();
        }
    }
    private void jbInit() throws Exception {
        LOzDB.setConnection(connD);
    }
    public static LearnerMCQDB lmqDB(){
        return lmqDB;
    }

    public LearnerMCQ findaLearnerMCQDB(short lID,short mcqID){
        LearnerMCQ lmq;
        QueryDataSet getLeaQ = new QueryDataSet();
        QueryDescriptor qD = new QueryDescriptor(LOzDB, "SELECT
\"LearnerMCQ\".*" +
        " FROM \"LearnerMCQ\" WHERE \"LearnerMCQ\".\"leaID\"=" + lID +
        " AND \"LearnerMCQ\".\"mcqID\" =" + mcqID, null, false,
Load.ALL);
        getLeaQ.setQuery(qD);
        getLeaQ.executeQuery();
        LOzDB.closeConnection();
        if (getLeaQ.isEmpty())
            lmq=new LearnerMCQ(); // different from others;
        else
            lmq=new LearnerMCQ(getLeaQ.getShort(3));
        getLeaQ.closeStatement();
        return lmq;
    }
}

```

Class LearnerMS

```

package lozpt1;

public class LearnerMS {
    private Learner L;
    private MentalState ms;
    private short lastMCQno=0; //important to keep last tried mcq
related to ms
    private StrengthMS sms= null;
    private static String summ="";

    public LearnerMS(short m, short l) {
        sms= new StrengthMS(m);
        lastMCQno=1;
        summ="\n MS visited before- SMS already exist, old value is ==>
"+
        Integer.toString(m);
    }
}

```

```

    }

    public LearnerMS() {
    }

    // for not first MS of MC; includes first MS of not first LC, and
    not first MS of first LC
    public static LearnerMS findORcreateLMS(Learner L, MentalState ms,
    LearnerMS lLMS, short scuffL){
        LearnerMS lms= new
    LearnerMSDB().lmsDB().findaLearnerMSDB(L.leaID(),ms.msID());
        lms.L=L;lms.ms=ms;lms.lastMCQno=(short)0;
        short smsV= findSMSv(lLMS, scuffL);
        if (lms.sms==null){ // if MS not visited before,
            //summ+="\n MS not visited before- new SMS has been calculated
using SMS of previous MS";
            lms.sms= new StrengthMS(smsV);
        } else if (lms.sms.get()<smsV){
            lms.sms= new StrengthMS(smsV);
            //summ+="\n MS visited before- but new SMS, calculated
using SMS of previous MS is bigger";
        }
        return lms;
    }

    public static LearnerMS findORcreateFreshLMS(Learner L, MentalState
    ms, LearnerMS lLMS){
        LearnerMS lms= new
    LearnerMSDB().lmsDB().findaLearnerMSDB(L.leaID(),ms.msID());
        lms.L=L;lms.ms=ms;lms.lastMCQno=(short)0;
        short smsV= findSMSv(lLMS);
        if (lms.sms==null){ // if MS not visited before,
            //summ+="\n MS not visited before- new SMS has been calculated
using SMS of previous MS (of previous LC)";
            lms.sms= new StrengthMS(smsV);
        } else if (lms.sms.get()<smsV){
            lms.sms= new StrengthMS(smsV);
            //summ+="\n MS visited before- but new SMS, calculated
using SMS of previous MS (of previous LC) is bigger";
        }
        return lms;
    }

    public static LearnerMS findORcreateLMS(Learner L, MentalState ms){
    // for first MS of MC (first LC)
        LearnerMS lms= new
    LearnerMSDB().lmsDB().findaLearnerMSDB(L.leaID(),ms.msID());
        lms.L=L;lms.ms=ms;lms.lastMCQno=(short)0;
        if (lms.sms==null){ //MS not visited before
            lms.sms= new StrengthMS((short)50);
            summ+="\n MS not visited before- new SMS is assigned,
                SMS <== 50";
        }else if (lms.sms.get()<50) {
            lms.sms= new StrengthMS((short) 50);
            summ+="\n MS visited before, but less than 50)-
therefore, new SMS is assigned, SMS <== 50";
        } // else leave the previous sms
        return lms;
    }
}

```

```

public StrengthMS getSMS(){
    return sms;
}
public void setSMS(StrengthMS m){
    sms=m;
}
public void setL(Learner L){
    this.L=L;
}

public void setLmcqNO(short i){
    lastMCQno=i;
}
public short getLmcqNO(){
    return lastMCQno;
}
public String summ(){
    return summ;
}

public static short findSMSv(LearnerMS lLMS, short scuffL){
    //inter LC
    short m=50;
    short m1=lLMS.getSMS().get();
    //if (ms.dependMS(0).msID() == lastLMS.ms.msID()) {
    // not first MS
    if (m1>m){
        m = (short)(m +(m1-m)*0.60); //scuffL not used now; cater both
        // new and intermediate MSs
        summ="\n SMS of previous MS (" + Integer.toString(m1)+ ") is
        greater than 50; new SMS <-- "+
        Integer.toString(m)+ " is assigned at the scaffolding
        level ==> "+ Integer.toString(scuffL+1);
    }
    else{
        m = (short)(m1 +(m-m1)*0.60); //scuffL not used now; cater
        both new and intermediate MSs
        summ="\n SMS of previous MS (" + Integer.toString(m1)+ ") is
        less than (or equal to)50; new SMS <-- "+
        Integer.toString(m)+ " is assigned at the scaffolding
        level ==> "+ Integer.toString(scuffL+1);
    }
    return m;
}

public static short findSMSv(LearnerMS lLMS){ // inter MC
    short m=50;
    short m1=lLMS.getSMS().get();
    //if (ms.dependMS(0).msID() == lastLMS.ms.msID()) {
    // not first MS
    if (m1>m){
        m = (short)(m +(m1-m)*0.60); //scuffL not used now; cater
both new and intermediate MSs
        summ="\n SMS of previous MS (" + Integer.toString(m1)+ ") is
greater than 50; new SMS <-- "+
        Integer.toString(m)+ " is assigned";
    }
    else{
        m = (short)(m1 +(m-m1)*0.60); //scuffL not used now; cater
both new and intermediate MSs

```

```

        summ="\n SMS of previous MS (" + Integer.toString(m1) + ") is
less than (or equal to)50; new SMS <-- "+
        Integer.toString(m) + " is assigned";
    }
    return m;
}
}
/*
public void initSMS(){ // will it be used?
    short m=50; //sms for first MS
    if (ms.dependMS(0) !=null) { // not first MS
        short m1= (short)(ms.dependMS(0).getSMS(L));
        if (m1>m)
            m= (short)(50 + ((m1-50)/3)); //
    }
    sms= new StrengthMS(m);
}
*/

public void setMS(MentalState MS){
    ms=MS;
}
public MentalState getMS(){
    return ms;
}

public void updateSMS(short dL, boolean c){
    sms.updateSMS(dL,c);
}
}
}

```

Class LearnerMSDB

```

package lozpt1;

import com.borland.dx.dataset.*;
import com.borland.dx.sql.dataset.*;
import java.sql.*;

public class LearnerMSDB extends LozDB{
    private static LearnerMSDB lmsDB = new LearnerMSDB();
    public LearnerMSDB() {
        try {
            jbInit();
        }
        catch(Exception e) {
            e.printStackTrace();
        }
    }
    private void jbInit() throws Exception {
        LOZdB.setConnection(connD);
    }
    public static LearnerMSDB lmsDB(){
        return lmsDB;
    }

    public LearnerMS findaLearnerMSDB(short lID,short msID){
        short type=6;
        LearnerMS lms = (LearnerMS)Cache.find(new Short(msID),type);
        if (lms==null){
            lms=findINdb(lID,msID);
        }
    }
}

```

```

        Cache.put(new Short(msID), lms,type);
    }
    return lms;
}

public LearnerMS findINdb(short lID,short msID){
    LearnerMS lms;
    QueryDataSet getLeaQ = new QueryDataSet();
    QueryDescriptor qD = new QueryDescriptor(LOZdB, "SELECT
        \"LearnerMS\".*\" + \" FROM \"LearnerMS\" WHERE
        \"LearnerMS\".\"leaID\"=\" + lID + \" AND \"LearnerMS\".\"msID\"
        =\" + msID, null, false, Load.ALL);
    getLeaQ.setQuery(qD);
    getLeaQ.executeQuery();
    LOZdB.closeConnection();
    if (getLeaQ.isEmpty())
        lms=new LearnerMS(); // different from others;
    else
        lms=new LearnerMS(getLeaQ.getShort(3),getLeaQ.getShort(4));
    getLeaQ.closeStatement();
    return lms;
}
}
}

```

Class LearnerOtherRecord

```

package lozpt1;

public class LearnerOtherRecord {
    private short leaORecID; //same as leaID, can omit later
    private short leaID;
    private short zLevel;
    private short mathsLevel;
    private short ooLevel;
    private short umlLevel;
    private short sessionIDs[];
    private short leTopicIDs[];
    short leMCIDs[];
    short leLCIDs[];
    short leQIDs[];
    short leFBIDs[];
    short leMSIDs[];
    long regDate;
    String regLog="";

    public LearnerOtherRecord(short leaOReID, short leID, short z, short
m,
                                short o, short u) {
        leaORecID=leaOReID; leaID=leID; zLevel=z; mathsLevel=m; ooLevel=o;
umlLevel=u;
    }
    public short getZlevel(){
        return zLevel;
    }
    public short initialGLA(){ //initial Global Learn Ability
        return (short) ((mathsLevel+ooLevel+umlLevel+zLevel)/3);
    }
    public void addRegLog(String m){
        regLog=regLog+m;
    }
}

```

```
}  
}
```

Class LearnerOtherRecordDB

```
package lozpt1;  
  
import com.borland.dx.dataset.*;  
import com.borland.dx.sql.dataset.*;  
import java.sql.*;  
  
public class LearnerOtherRecordDB extends LozDB{  
    private static LearnerOtherRecordDB loRDB = new  
LearnerOtherRecordDB();  
    public LearnerOtherRecordDB() {  
    }  
    public static LearnerOtherRecordDB loRDB(){  
        return loRDB;  
    }  
    public LearnerOtherRecord findaLearnerOtherRecord(short loRID){  
        QueryDataSet getLeaQ = new QueryDataSet();  
        QueryDescriptor qD = new QueryDescriptor(LOZdB, "SELECT  
\"LearnerOtherRecord\".*"+  
            " FROM\"LearnerOtherRecord\" WHERE  
\"LearnerOtherRecord\".\"leaORecID\"=\"" + loRID  
            , null, false, Load.ALL);  
        getLeaQ.setQuery(qD);  
        getLeaQ.executeQuery();  
        LOZdB.closeConnection();  
        if (getLeaQ.isEmpty())  
            return null;  
        LearnerOtherRecord lr=new LearnerOtherRecord(loRID,  
getLeaQ.getShort(2),  
            getLeaQ.getShort(3), getLeaQ.getShort(4),  
getLeaQ.getShort(5),  
            getLeaQ.getShort(6));  
        getLeaQ.closeStatement();  
        return lr;  
    }  
}
```

Class LearnerPreference

```
package lozpt1;  
  
public class LearnerPreference {  
    private short lpID;  
    private boolean summaryON=true;  
    private boolean summOptON=true;  
    private boolean zhhelpON=false;  
    public LearnerPreference(short lPID, boolean s, boolean so, boolean  
z) {  
        lpID=lPID; summaryON=s; summOptON=so; zhhelpON=z;  
    }  
    public boolean summaryON(){  
        return summaryON;  
    }  
    public boolean summOptON(){
```

```

    return summOptON;
}
public void toggleSummary(){
    summaryON = !(summaryON);
}
public void toggleSummOptON(){
    summOptON = !(summOptON);
}
public boolean zhelpON(){
    return zhelpON;
}
}

```

Class LearnerPreferenceDB

```

package lozpt1;

import com.borland.dx.dataset.*;
import com.borland.dx.sql.dataset.*;
import java.sql.*;

public class LearnerPreferenceDB extends LozDB{
    private static LearnerPreferenceDB lpDB = new LearnerPreferenceDB();
    public LearnerPreferenceDB() {
    }
    public static LearnerPreferenceDB lpDB(){
        return lpDB;
    }
    public LearnerPreference findaLearnerPreference(short lpID){
        QueryDataSet getLeaQ = new QueryDataSet();
        QueryDescriptor qD = new QueryDescriptor(LOZdB, "SELECT
\"LearnerPreference\".*"+
        " FROM\"LearnerPreference\" WHERE
\"LearnerPreference\".\"leaID\"=" + lpID
        , null, false, Load.ALL);
        getLeaQ.setQuery(qD);
        getLeaQ.executeQuery();
        LOZdB.closeConnection();
        if (getLeaQ.isEmpty())
            return null;
        LearnerPreference lp=new
LearnerPreference(lpID,getLeaQ.getBoolean(2),
        getLeaQ.getBoolean(3),getLeaQ.getBoolean(4));
        getLeaQ.closeStatement();
        return lp;
    }
}

```

Class LearnerRecordDB

```

package lozpt1;

import com.borland.dx.dataset.*;
import com.borland.dx.sql.dataset.*;
import java.sql.*;

public class LearnerRecordDB extends LozDB{
    private static LearnerRecordDB lrDB = new LearnerRecordDB();

```

```

public LearnerRecordDB() {
}
public static LearnerRecordDB lrDB(){
    return lrDB;
}
public LearnRecord findaLearnRecord(Short lID){
    QueryDataSet getLeaQ = new QueryDataSet();
    QueryDescriptor qD = new QueryDescriptor(LOZdB, "SELECT
\"LearnRecords\".*"+
        " FROM\"LearnRecords\" WHERE \"LearnRecords\".\"leaID\"=\"" +
lID.shortValue()
        , null, false, Load.ALL);
    getLeaQ.setQuery(qD);
    getLeaQ.executeQuery();
    LOZdB.closeConnection();
    if (getLeaQ.isEmpty())
        return null;
    LearnRecord lr=new LearnRecord(new Short(getLeaQ.getShort(1)), new
Short(getLeaQ.getShort(2)),
        getLeaQ.getShort(3), new
Short(getLeaQ.getShort(4)));
    getLeaQ.closeStatement();
    return lr;
}
}

```

Class LearnerSessLC

```

package lozpt1;

public class LearnerSessLC {
    private short leaID;
    private short lcID;
    private short sessID;
    private long duration;
    private short[] leaQUIDs;
    private short leaAbi;
    public LearnerSessLC() {
    }
}

```

Class LearnRecord

```

package lozpt1;

public class LearnRecord {
    Short leaRecID;
    short lastLeaCptType=3; // not in constructor
    Short lastLeaCptID;
    short pointer;
    Short leaID;

    public LearnRecord(Short lRecID, Short lCptID, short po, Short lID)
    {
        leaRecID=leaID; lastLeaCptID=lCptID; pointer=po; leaID=lID;
    }
    public Short getLastLeaCpt(){

```



```

        return lastLeaCptID;
    }
    public short getPointer(){
        return pointer;
    }
}

```

Class LeaSummUI

```

package lozpt1;

import javax.swing.*;
import java.awt.*;
import com.borland.jbcl.layout.*;
import java.awt.event.*;

public class LeaSummUI extends JFrame{
    private Mentor m;
    private JPanel contentPane;
    private JPanel jPanel2 = new JPanel();
    private JScrollPane jScrollPane1 = new JScrollPane();
    private JPanel jPanel3 = new JPanel();
    private JTextPane jTextPane1 = new JTextPane();
    private BorderLayout BorderLayout1 = new BorderLayout();
    private JButton jButton5 = new JButton();
    private JButton OK = new JButton();
    private FlowLayout flowLayout1 = new FlowLayout();
    private JLabel jLabel1 = new JLabel();
    private BorderLayout BorderLayout2 = new BorderLayout();
    private JButton jButton6 = new JButton();

    public LeaSummUI(Mentor m ) {
        try {
            jbInit();
        }
        catch(Exception e) {
            e.printStackTrace();
        }
        this.m=m;
    }

    private void jbInit() throws Exception {
        contentPane = (JPanel) this.getContentPane();
        setTitle("LOZ- LEARNER INTERACTION SUMMARY");
        contentPane.setLayout(borderLayout1);
        jPanel3.setLayout(borderLayout2);
        jPanel2.setLayout(flowLayout1);
        jButton5.setBackground(new Color(236, 255, 216));
        jButton5.setFont(new java.awt.Font("SansSerif", 1, 15));
        jButton5.setForeground(Color.red);
        jButton5.setText("Inspect LM");
        jButton5.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(ActionEvent e) {
                jButton5_actionPerformed(e);
            }
        });
    }
}

```

```

OK.setBackground(new Color(236, 255, 216));
OK.setFont(new java.awt.Font("SansSerif", 1, 15));
OK.setForeground(Color.red);
OK.setText("OK");
OK.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(ActionEvent e) {
        OK_actionPerformed(e);
    }
});
jPanel3.setBackground(new Color(236, 217, 216));
jPanel3.setMinimumSize(new Dimension(400, 30));
jPanel3.setPreferredSize(new Dimension(400, 50));
jLabel1.setFont(new java.awt.Font("Serif", 1, 25));
jLabel1.setForeground(new Color(144, 131, 79));
jPanel2.setBackground(new Color(236, 217, 216));
jButton6.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(ActionEvent e) {
        jButton6_actionPerformed(e);
    }
});
jButton6.setText("Ins.. Lesson");
jButton6.setForeground(Color.red);
jButton6.setFont(new java.awt.Font("SansSerif", 1, 15));
jButton6.setBackground(new Color(236, 255, 216));
contentPane.add(jScrollPane1, BorderLayout.CENTER);
jScrollPane1.getViewport().add(jTextPanel, null);
jPanel2.add(jButton6, null);
jPanel2.add(jButton5, null);
jPanel2.add(OK, null);
contentPane.add(jPanel3, BorderLayout.NORTH);
jPanel3.add(jLabel1, BorderLayout.WEST);
//jPanel3.add(jLabel2, new XYConstraints(244, 2, 124, 42));
//jPanel3.add(jLabel3, new XYConstraints(385, 2, 411, 42));
contentPane.add(jPanel2, BorderLayout.SOUTH);
}

public void presentWelcome(String w1){
    jLabel1.setText(w1);
}

public void pls(String s){
    jTextPanel.setText("");
    AttributeSet.aSet.setStyle(1,1);
    Util.U.appendToTextPane(jTextPanel,"Learner's Interaction
Summary \n",AttributeSet.aSet.heading2);
    Util.U.appendToTextPane(jTextPanel, " << LOZ assumes that you
already understand the semantics"+
        " of the measures Global Learn Ability (GLA), Local Learn
Ability (LLA), Strength of Mental State (SMS),"+
        "and Strength of Pedagogical Action (PAS or PAS-Index). More
information about these measures may be"+
        " found in Learner Model Help >> \n*****\n",
AttributeSet.aSet.para2);
    Util.U.appendToTextPane(jTextPanel,s+"\n",AttributeSet.aSet.para1);
    Util.U.appendToTextPane(jTextPanel," ==> You may change your
Learner Model; press <Inspect LM> \n",
AttributeSet.aSet.para2);
}

void OK_actionPerformed(ActionEvent e) {
    this.dispose();
}

```

```

    }

    void jButton5_actionPerformed(ActionEvent e) {
        m.inspectLM();
    }
    void jButton6_actionPerformed(ActionEvent e) {
        LessShowUI l=new LessShowUI(m);
        l.setSize(900,350);
        l.validate();
        l.setVisible(true);
        Util.U.setCentre(l);
    }
}

```

Class LessShowUI

```

package lozpt1;

import javax.swing.*.*;
import java.awt.*.*;
import com.borland.jbcl.layout.*;
import java.awt.event.*;

public class LessShowUI extends JFrame{
    private Mentor m;
    private JPanel contentPane;
    private JPanel jPanel2 = new JPanel();
    private JScrollPane jScrollPane1 = new JScrollPane();
    private JTextPane jTextPane1 = new JTextPane();
    private BorderLayout borderLayout1 = new BorderLayout();
    private JPanel jPanel3 = new JPanel();
    private JLabel jLabel1 = new JLabel();
    private XYLayout xYLayout1 = new XYLayout();
    private JLabel jLabel2 = new JLabel();
    private JTextField jTextField1 = new JTextField();
    private JTextField jTextField2 = new JTextField();
    private JTextField jTextField3 = new JTextField();
    private JTextField jTextField4 = new JTextField();
    private JLabel jLabel3 = new JLabel();
    private JLabel jLabel4 = new JLabel();
    private BorderLayout borderLayout2 = new BorderLayout();
    private JPanel jPanel1 = new JPanel();
    private JButton jButton1 = new JButton();
    private JButton jButton2 = new JButton();
    private JLabel jLabel5 = new JLabel();
    private JLabel jLabel6 = new JLabel();

    public LessShowUI(Mentor m ) {
        try {
            jbInit();
        }
        catch(Exception e) {
            e.printStackTrace();
        }
        this.m=m;
        pls();
    }
}

```

```

private void jbInit() throws Exception {
    contentPane = (JPanel) this.getContentPane();
    setTitle("LOZ- LESSON & SCAFFOLDING STAGES");
    contentPane.setLayout(borderLayout1);
    jPanel2.setLayout(borderLayout2);
    jPanel2.setBackground(new Color(236, 217, 216));
    jPanel3.setLayout(xYLayout1);
    jPanel3.setBackground(new Color(236, 217, 216));
    jPanel3.setBorder(BorderFactory.createRaisedBevelBorder());
    jPanel3.setMinimumSize(new Dimension(400, 30));
    jPanel3.setPreferredSize(new Dimension(400, 50));
    jLabel1.setFont(new java.awt.Font("Serif", 1, 15));
    jLabel1.setForeground(new Color(144, 0, 66));
    jLabel1.setText("Start Stage");
    jLabel2.setText("Stage Steps");
    jLabel2.setForeground(new Color(144, 0, 0));
    jLabel2.setFont(new java.awt.Font("Serif", 1, 15));
    jTextField1.setFont(new java.awt.Font("SansSerif", 1, 15));
    jTextField1.setText("1");
    jTextField3.setFont(new java.awt.Font("SansSerif", 1, 15));
    jTextField3.setText("1");
    jLabel3.setText("New");
    jLabel3.setForeground(Color.red);
    jLabel3.setFont(new java.awt.Font("Serif", 1, 15));
    jLabel4.setFont(new java.awt.Font("Serif", 1, 15));
    jLabel4.setForeground(Color.red);
    jLabel4.setText("Old");
    cancel.setBackground(new Color(236, 184, 190));
    cancel.setFont(new java.awt.Font("SansSerif", 1, 15));
    cancel.setForeground(new Color(0, 33, 66));
    cancel.setText("Cancel");
    cancel.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(ActionEvent e) {
            cancel_actionPerformed(e);
        }
    });
    OK.setBackground(new Color(236, 184, 190));
    OK.setFont(new java.awt.Font("SansSerif", 1, 15));
    OK.setForeground(new Color(0, 33, 66));
    OK.setText("OK");
    OK.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(ActionEvent e) {
            OK_actionPerformed(e);
        }
    });
    jPanel1.setBackground(new Color(236, 217, 216));
    jLabel5.setText("Old");
    jLabel5.setForeground(Color.red);
    jLabel5.setFont(new java.awt.Font("Serif", 1, 15));
    jLabel6.setText("New");
    jLabel6.setForeground(Color.red);
    jLabel6.setFont(new java.awt.Font("Serif", 1, 15));
    contentPane.add(jScrollPane1, BorderLayout.CENTER);
    jScrollPane1.getViewport().add(jTextPanel, null);
    jPanel3.add(jTextField1, new XYConstraints(140, 2, 45, 24));
    jPanel3.add(jTextField2, new XYConstraints(140, 19, 45, 24));
    jPanel3.add(jTextField3, new XYConstraints(346, 0, 45, 24));
    jPanel3.add(jTextField4, new XYConstraints(347, 21, 45, 24));
    jPanel3.add(jLabel4, new XYConstraints(310, 1, 32, 24));
    jPanel3.add(jLabel5, new XYConstraints(102, 0, 32, 24));
    jPanel3.add(jLabel3, new XYConstraints(103, 21, 32, 24));
}

```

```

jPanel3.add(jLabel1, new XYConstraints(217, 10, 85, 24));
jPanel3.add(jLabel2, new XYConstraints(9, 9, 85, 24));
jPanel3.add(jLabel6, new XYConstraints(306, 21, 32, 24));
jPanel2.add(jPanel1, BorderLayout.SOUTH);
jPanel1.add(cancel, null);
jPanel1.add(OK, null);
jPanel2.add(jPanel3, BorderLayout.CENTER);
//jPanel3.add(jLabel2, new XYConstraints(244, 2, 124, 42));
//jPanel3.add(jLabel3, new XYConstraints(385, 2, 411, 42));
contentPane.add(jPanel2, BorderLayout.SOUTH);

}

public void presentWelcome(String w1){
    jLabel1.setText(w1);
}

public void pls(){
    jTextPanel.setText("");
    AttributeSet.aSet.setStyle(1,1);
    Util.U.appendToTextPane(jTextPanel," Next Lesson is ==> <<
VISIBILITY LIST - basics >>" , AttributeSet.aSet.heading4);
    Util.U.appendToTextPane(jTextPanel, "\n There are 3 Learning
Stages associated with this lesson: \n" +
        "    Stage (1) : Able to specify Visibility List, if UML
accessibility spec is fully given. \n " +
        "    Stage (2) : Able to specify Visibility List, if UML
spec is partially given (without accessibility). \n" +
        "    Stage (3) : Able to specify Visibility List, for a
given textual specification; \n",AttributeSet.aSet.para1);
    Util.U.appendToTextPane(jTextPanel," \n==> You may change the
Start Stage and/or the Steps : Give new values below",
        AttributeSet.aSet.para2);
}

void OK_actionPerformed(ActionEvent e) {
    this.dispose();
}

void cancel_actionPerformed(ActionEvent e) {
    this.dispose();
}
}

```

Class LocalLA

```

package lozpt1;

public class LocalLA {
    private short lla=0;
    private FuzzyMem3L llaFZ=null;
    private String fuzz="";
    private String deFuzz="";
    private String upD="";
    private String pred="";
    private short dynT=90; //dynamic change probability for true to true
    private short dynF=10;
    private float affirm= (float)2.7; //mental state affirms local learn
ability
    private float deny=(float)0.29;
}

```

```

public LocalLA(short la){ //crisp to set
    lla=la;
    fuzz=fuzz+ "\n@@ The current % scale for LLA is "
+Integer.toString(lla) + "\n";
    llaFZ=FuzzyMem3L.fuzzify(la); //crisp may not equal to lla now
    /*
    fuzz=fuzz+ "\n@@ The resultant FUZZY values for LLA are \n"+
        "==> The fuzzy value, LLA- HIGH = " +
Integer.toString(llaFZ.level((short)0)) + "\n"+
        "==> The fuzzy value, time- MEDIUM = " +
Integer.toString(llaFZ.level((short)1)) + "\n"+
        "==> The fuzzy value, time- LOW = " +
Integer.toString(llaFZ.level((short)2)) + "\n";
    */
}

public LocalLA(GlobalLA gla){ //GLA to LLA (LLA will be considered
GLA)
    lla=gla.get();
    fuzz=fuzz+ "\n@@ The current % scale for LLA is "
+Integer.toString(lla) + "\n";
    llaFZ=gla.getFZ();
    /*
    fuzz=fuzz+ "\n@@ The resultant FUZZY values for LLA are \n"+
        "==> The fuzzy value, LLA- HIGH = " +
Integer.toString(llaFZ.level((short)2)) + "\n"+
        "==> The fuzzy value, LLA- MEDIUM = " +
Integer.toString(llaFZ.level((short)1)) + "\n"+
        "==> The fuzzy value, LLA- LOW = " +
Integer.toString(llaFZ.level((short)0)) + "\n";
    */
}

public LocalLA(GlobalLA gla, short time){ //set to crisp
    FuzzyMem3L timeFZ= FuzzyMem3L.fuzzify(time); //time in 0-100
    fuzz=fuzz+ "\n\n@@ The current % scale for TIME is "
+Integer.toString(time) + "\n"+
        "==> The fuzzy value, time- MUCH = " +
Integer.toString(timeFZ.level((short)2)) + "\n"+
        "==> The fuzzy value, time- REASONABLE = " +
Integer.toString(timeFZ.level((short)1)) + "\n"+
        "==> The fuzzy value, time- A LITTLE = " +
Integer.toString(timeFZ.level((short)0)) + "\n";
    FuzzyMem3L glaFZ= FuzzyMem3L.fuzzify(gla.get());
    fuzz=fuzz+ "\n\n@@ The current % scale for GLA is "
+Integer.toString(gla.get()) + "\n"+
        "==> The fuzzy value, GLA- HIGH = " +
Integer.toString(glaFZ.level((short)2)) + "\n"+
        "==> The fuzzy value, GLA- MEDIUM = " +
Integer.toString(glaFZ.level((short)1)) + "\n"+
        "==>@@ The fuzzy value, GLA - LOW = " +
Integer.toString(glaFZ.level((short)0)) + "\n";
    llaFZ= FuzzyMX3and3to3.applyRuleLLA(glaFZ,timeFZ);
    fuzz=fuzz+"\n\n!!!! Rules in GLA*Time ==> LLA is applied - see
help in LM for detail !!!!!\n";
    fuzz=fuzz+ "\n@@ The resultant values for LLA are \n"+
        "==> The fuzzy value, LLA- HIGH = " +
Integer.toString(glaFZ.level((short)2)) + "\n"+
        "==> The fuzzy value, time- MEDIUM = " +
Integer.toString(glaFZ.level((short)1)) + "\n"+

```

```

        "=> The fuzzy value, time- LOW = " +
Integer.toString(glaFZ.level((short)0)) + "\n";
        lla=llaFZ.crisp(); //crisp is equal to lla now
        fuzz=fuzz+ "@@ The resultant (% scale) for LLA now is "
+Integer.toString(lla) + "\n";
    }

    public short get(){
        return lla;
    }
    public FuzzyMem3L getFZ(){
        return llaFZ;
    }
    public String fuzz(){
        return fuzz;
    }
    public String deFuzz(){
        return deFuzz;
    }
    public String upD(){
        return upD;
    }
    public String pred(){
        return pred;
    }

    public void setLLA(short la){
        lla=la;
        llaFZ=FuzzyMem3L.fuzzify(la); //crisp may not equal to lla now
    }

    public void predictLLA (){
        pred=pred+ " \n$$$ The current (% scale) for LLA is " +
Integer.toString(lla) + "\n";
        lla=(short) ((dynT*lla+dynF*(100-lla))/100);
        pred=pred+ " \n$$$ Now, after DYNAMIC predictions, the (% scale)
for LLA is " +
            Integer.toString(lla) + "\n";
    }

    public void updateLLA(boolean t){
        float ohNOe= ((float)lla/(100-lla));
        upD=upD+ " \n$$$ The current (% scale) for LLA is " +
Integer.toString(lla) + "\n";
        String b = t ? "true" : "false";
        upD=upD+ " \n $$$ You are Strong in last Mental State - This
statement is " + b + "!";
        b = !t ? "denied by " + Float.toString(deny) : "affirmed by " +
Float.toString(affirm);
        upD=upD+ " \n $$$ Now, the (% scale) for LLA is " +
            Integer.toString(lla) + " " + b + "\n";
        if (t)
            ohNOe *= affirm;
        else
            ohNOe *= deny;
        lla=(short)((ohNOe*100)/(1+ohNOe));
        upD=upD+ " \n\n$$$ After updating, the resultant (% scale) for LLA
is " + Integer.toString(lla) ;
    }
}
}

```

Class LogInCtrl

```
package lozpt1;

import java.util.*;
import javax.swing.*;
import java.awt.*;

public class LogInCtrl {
    private boolean packFrame = false;
    private LogInUI logInUI;
    private Mentor TheMentor;
    private Learner aLearner;

    public LogInCtrl() {
        try {
            logInUI = new LogInUI(this);
            if (packFrame) {
                logInUI.pack();
            }
            else {
                logInUI.validate();
            }
            Dimension screenSize =
                Toolkit.getDefaultToolkit().getScreenSize();
            Dimension frameSize = logInUI.getSize();
            if (frameSize.height > screenSize.height) {
                frameSize.height = screenSize.height;
            }
            if (frameSize.width > screenSize.width) {
                frameSize.width = screenSize.width;
            }
            logInUI.setLocation((screenSize.width - frameSize.width) / 2,
                (screenSize.height - frameSize.height) / 2);
            logInUI.setVisible(true);
        }
        catch(Exception e) {
            e.printStackTrace();
        }
    }

    public void newLearner(){
        logInUI.dispose();
        NewLearnerCtrl nlCtrl= new NewLearnerCtrl();
    }

    public void OKpressed(){
        aLearner = LearnerDB.lDB().findaLearnerUID(logInUI.userID());
        if (aLearner==null)
            JOptionPane.showMessageDialog(null, "Invalid User Name. Try
                again.", "Error Message", JOptionPane.ERROR_MESSAGE);
        else
            if (aLearner.pwOK(logInUI.passWD())){
                logInUI.dispose();
                Mentor theMentor = new Mentor(aLearner, "Records Loaded
                    Successfully");
                theMentor.startLessonOldL();
            }
            else
    }
    }
}
```



```

        JOptionPane.showMessageDialog(null, "Invalid password. Try
        again.", "Error Message", JOptionPane.ERROR_MESSAGE);
    }

    public void DemoPressed(){
        JFrame wd = new WelcomeDemoUI();
        wd.pack();
        Util.U.setCentre(wd);
        wd.setVisible(true);
    }
}

```

Class LogInUI

```

package lozpt1;

import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import com.borland.jbcl.layout.*;
import java.io.*;
import java.applet.*;
import java.net.URL;

public class LogInUI extends JFrame {
    private String userID;
    private char[] passWD;
    private LogInCtrl logInCtrl;
    private ActionEvent lEvent;
    private JPanel contentPane;
    private XYLayout xYLayout1 = new XYLayout();
    private JPanel jPanel2 = new JPanel();
    private JPanel jPanel3 = new JPanel();
    private JButton NewLearner = new JButton();
    private XYLayout xYLayout2 = new XYLayout();
    private JButton Demo = new JButton();
    private JButton About = new JButton();
    private JButton Quit = new JButton();
    private XYLayout xYLayout3 = new XYLayout();
    private JLabel returnLearner = new JLabel();
    private JTextField userId = new JTextField();
    private JLabel jLabel2 = new JLabel();
    private JLabel jLabel3 = new JLabel();
    private JButton OK = new JButton();
    private JPasswordField pwText = new JPasswordField(20);
    private JLabel jLabel4 = new JLabel();
    private JLabel jLabel1 = new JLabel();
    private AudioClip wcClip;

    //Construct the frame
    public LogInUI(LogInCtrl liC) {
        enableEvents(AWTEvent.WINDOW_EVENT_MASK);
        try {
            logInCtrl=liC;
            jbInit();
        }
        catch(Exception e) {
            e.printStackTrace();
        }
    }
}

```

```

    }
}
//Component initialization
private void jbInit() throws Exception {
    contentPane = (JPanel) this.getContentPane();
    contentPane.setLayout(xYLayout1);
    this.getContentPane().setBackground(new Color(236, 175, 177));
    this.setSize(new Dimension(554, 377));
    this.setTitle("LOZ- LogIn Screen");
    jPanel3.setBackground(new Color(59, 177, 197));
    jPanel3.setBorder(BorderFactory.createRaisedBevelBorder());
    jPanel3.setLayout(xYLayout3);
    jPanel2.setBackground(new Color(59, 171, 190));
    jPanel2.setBorder(BorderFactory.createRaisedBevelBorder());
    jPanel2.setLayout(xYLayout2);
    NewLearner.setBackground(new Color(66, 171, 197));
    NewLearner.setFont(new java.awt.Font("Dialog", 1, 12));
    NewLearner.setForeground(new Color(49, 255, 197));
    NewLearner.setText("New Learner");
    NewLearner.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(ActionEvent e) {
            NewLearner_actionPerformed(e);
        }
    });
    Demo.setBackground(new Color(66, 171, 197));
    Demo.setEnabled(false);
    Demo.setFont(new java.awt.Font("Dialog", 1, 12));
    Demo.setForeground(new Color(49, 255, 197));
    Demo.setText("Demo");
    Demo.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(ActionEvent e) {
            Demo_actionPerformed(e);
        }
    });
    About.setBackground(new Color(66, 171, 197));
    About.setFont(new java.awt.Font("Dialog", 1, 12));
    About.setForeground(new Color(49, 255, 197));
    About.setText("About");
    About.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(ActionEvent e) {
            About_actionPerformed(e);
        }
    });
    Quit.setBackground(new Color(66, 171, 197));
    Quit.setFont(new java.awt.Font("Dialog", 1, 12));
    Quit.setForeground(new Color(49, 255, 197));
    Quit.setText("Quit");
    Quit.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(ActionEvent e) {
            Quit_actionPerformed(e);
        }
    });
    returnLearner.setFont(new java.awt.Font("Dialog", 0, 20));
    returnLearner.setForeground(new Color(46, 255, 118));
    returnLearner.setText("Returning Learner");
    userId.setText("Give your Learner ID");
    jLabel2.setFont(new java.awt.Font("Dialog", 3, 14));
    jLabel2.setForeground(new Color(49, 255, 197));
    jLabel2.setText("Learner ID");
    jLabel3.setFont(new java.awt.Font("Dialog", 3, 14));
    jLabel3.setForeground(new Color(49, 255, 197));
}

```

```

jLabel3.setText("Password");
OK.setBackground(new Color(66, 171, 197));
OK.setFont(new java.awt.Font("Dialog", 1, 12));
OK.setForeground(new Color(49, 255, 197));
OK.setText("OK");
OK.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(ActionEvent e) {
        OK_actionPerformed(e);
    }
});
contentPane.setBackground(new Color(66, 171, 197));
contentPane.setForeground(new Color(66, 171, 97));
pwText.setText("test");
jLabel4.setText("A System for Learning Object-Z");
jLabel4.setForeground(new Color(0, 255, 85));
jLabel4.setFont(new java.awt.Font("Dialog", 0, 18));
jLabel1.setText("WELCOME to LOZ");
jLabel1.setForeground(new Color(53, 255, 79));
jLabel1.setFont(new java.awt.Font("Serif", 1, 20));
contentPane.add(jPanel3, new XYConstraints(13, 91, 360, 236));
contentPane.add(jPanel2, new XYConstraints(399, 93, 135, 232));
jPanel2.add(NewLearner, new XYConstraints(15, 18, -1, -1));
jPanel2.add(Quit, new XYConstraints(15, 183, 107, -1));
jPanel2.add(Demo, new XYConstraints(15, 69, 107, -1));
jPanel2.add>About, new XYConstraints(15, 127, 107, -1));
contentPane.add(jLabel1, new XYConstraints(169, 9, 188, 40));
contentPane.add(jLabel4, new XYConstraints(132, 49, 289, 31));
jPanel3.add(returnLearner, new XYConstraints(16, 11, 221, 33));
jPanel3.add(jLabel2, new XYConstraints(269, 53, 75, 19));
jPanel3.add(userId, new XYConstraints(21, 73, 318, 35));
jPanel3.add(jLabel3, new XYConstraints(269, 116, 74, -1));
jPanel3.add(OK, new XYConstraints(253, 190, 84, 34));
jPanel3.add(pwText, new XYConstraints(21, 143, 313, 38));
String u="file:///"+LozGlobal.lozG().path()+"//wc.wav";
URL url = new URL(u);
wcClip= Applet.newAudioClip(url);
wcClip.loop();
}
//Overridden so we can exit when window is closed
protected void processWindowEvent(WindowEvent e) {
    super.processWindowEvent(e);
    if (e.getID() == WindowEvent.WINDOW_CLOSING) {
        System.exit(0);
    }
}
void Password_actionPerformed(ActionEvent e) {
    passWD= pwText.getPassword();
    wcClip.stop();
}
void NewLearner_actionPerformed(ActionEvent e) {
    logInCtrl.newLearner();
    wcClip.stop();
}
void OK_actionPerformed(ActionEvent e) {
    userID= userId.getText();
    passWD= pwText.getPassword();
    logInCtrl.OKpressed();
    wcClip.stop();
}
}

```

```

String userID(){
    return userID;
}

char[] passWD(){
    return passWD;
}

void Quit_actionPerformed(ActionEvent e) {
    wcClip.stop();
    this.dispose();
}

void About_actionPerformed(ActionEvent e) {
    JOptionPane.showMessageDialog(null, "A Learning System for LOZ -
    More Later", "About LOZ", JOptionPane.PLAIN_MESSAGE);
}

void Demo_actionPerformed(ActionEvent e) {
    logInCtrl.DemoPressed();
}
}

```

Class LozDB

```

package lozpt1;

import com.borland.dx.sql.dataset.*;
import java.sql.*;

public class LozDB {
    protected Database LOZdB = new Database();
    protected ConnectionDescriptor connD = new
    ConnectionDescriptor("jdbc:borland:dslocal:"
        + "C:\\Mohan\\LOZpt1\\LOZdB\\LOZdB.jds",
            "sm", "tm", false,
        "com.borland.datastore.jdbc.DataStoreDriver");

    public LozDB() {
        try {
            jbInit();
        }
        catch(Exception e) {
            e.printStackTrace();
        }
    }
    private void jbInit() throws Exception {
        LOZdB.setConnection(connD);
    }
}

```

Class LozGlobal

```

package lozpt1;

public class LozGlobal {
    private static LozGlobal lG = new LozGlobal();
    public short firstMCid=10;
}

```

```

private static String path = "C:\\mohan\\lozpt1\\src\\";
public LozGlobal() { }
public static LozGlobal lozG(){
    return lG;
}
public void putL(){
    DataBase.dB.putL();
}
public void initLOZ(){
    // must read from a file
    short llid=102;
    Learner.setLastLeaID(llid);
    //DataBase.dB.putL();
}
public String path(){
    return path;
}
}

```

Class LOZmain

```

package lozpt1;

import javax.swing.*;
import java.awt.*;

public class LOZmain {
    //Construct the application
    public LOZmain() {
        LozGlobal.lozG().initLOZ();
        LogInCtrl logInCtrl = new LogInCtrl();
    }

    //Main method
    public static void main(String[] args) {
        try {
            UIManager.setLookAndFeel(UIManager.
                getSystemLookAndFeelClassName());
        }
        catch(Exception e) {
            e.printStackTrace();
        }
        new LOZmain();
    }
}

```

Class MainConcept

```

package lozpt1;

public class MainConcept extends LearnConcept{
    private static short maxLC =3;
    private static short maxEC =3;
    private Topic topic;
    private static short topicID;
    private short preCondID;
    private short[] lcIDs = new short[maxLC];
    private short[] ecIDs = new short[maxEC]; //assume only One now
}

```

```

public MainConcept(short mcID, short mcTy, String h, String wt,
String wy, String hw, short miT, short mxT,short mS, String nt,
String nft, String et, String eft, short oo,short ad, short z,
short[] mSIDs, short tID, short pcID, short[] lCIDs, short[]
ecIDs ){

super(mcID,mcTy,h,wt,wy,hw,miT,mxT,mS,nt,nft,et,eft,oo,ad,z,mSIDs);
preCondID=pcID; topicID=tID;
for (short i=0; i<maxLC; i++ )
lcIDs[i] = lCIDs[i];
for (short i=0; i<maxEC; i++ )
ecIDs[i] = eCIDs[i];
}
public LearnConcept getPreCondition(){
return LearnConceptDB.lcDB().findaLearnConcept(preCondID);
}
public LearnConcept getLC(short i){
return LearnConceptDB.lcDB().findaLearnConcept(lcIDs[i-1]);
}
public ExploreConcept getEC(short i){
return ExploreConceptDB.ecDB().findaExploreConcept(ecIDs[i-1]);
}
public Topic getTopic(){
if (topic==null)
topic= TopicDB.toDB().findaTopic(topicID);
return topic;
}
public static short getMaxLC(){
return maxLC;
}
public static short getMaxEC(){
return maxEC;
}
public MainConcept nextMC(){
return topic.nextMC(this.getBCID());
}
}
}

```

Class MainConceptDB

```

package lozpt1;

import com.borland.dx.dataset.*;
import com.borland.dx.sql.dataset.*;
import java.sql.*;
import java.lang.Number;

public class MainConceptDB extends LozDB{
private static MainConceptDB mCDB = new MainConceptDB();
public MainConceptDB() {
}
public static MainConceptDB mcDB(){
return mCDB;
}

public MainConcept findaMainConcept(short mcID){
short type=1;
MainConcept mc = (MainConcept)Cache.find(new Short(mcID),type);
}
}

```

```

    if (mc==null){
        mc=findINdb(mcID);
        Cache.put(new Short(mcID), mc,type);
    }
    return mc;
}

public MainConcept findINdb(short mcID){
    QueryDataSet getBasQ = new QueryDataSet();
    QueryDescriptor qD = new QueryDescriptor(LOZdB, "SELECT
        \"BasicConcept\".*"+
        " FROM \"BasicConcept\" WHERE \"BasicConcept\".\"bcID\" = " +
        mcID, null, false, Load.ALL);
    getBasQ.setQuery(qD);
    getBasQ.executeQuery();
    if (getBasQ.isEmpty())
        return null;
    QueryDataSet getLeaQ = new QueryDataSet();
    QueryDescriptor qD1 = new QueryDescriptor(LOZdB, "SELECT
        \"LearnConcept\".*"+
        " FROM \"LearnConcept\" WHERE \"LearnConcept\".\"lcID\" = " +
        mcID, null, false, Load.ALL);
    getLeaQ.setQuery(qD1);
    getLeaQ.executeQuery();
    if (getLeaQ.isEmpty())
        return null;
    QueryDataSet getMaiQ = new QueryDataSet();
    QueryDescriptor qD2 = new QueryDescriptor(LOZdB, "SELECT
        \"MainConcept\".*"+
        " FROM \"MainConcept\" WHERE"+
        "\"MainConcept\".\"mcID\" = " +mcID , null, false, Load.ALL);
    getMaiQ.setQuery(qD2);
    getMaiQ.executeQuery();
    LOZdB.closeConnection();
    if (getMaiQ.isEmpty())
        return null;
    short maxMS=LearnConcept.maxMS();
    short mSIDs[] = new short[maxMS];
    for (short i=0; i<maxMS ;i++){
        mSIDs[i]= getLeaQ.getShort(9+i);
    }
    short maxLC=MainConcept.getMaxLC();
    short lcIDs[] = new short[maxLC];
    for (short i=0; i<maxLC ;i++){
        lcIDs[i]= getMaiQ.getShort(4+i);
    }
    short maxEC=MainConcept.getMaxEC();
    short ecIDs[] = new short[maxEC];
    for (short i=0; i<maxEC ;i++){
        ecIDs[i]= getMaiQ.getShort(4+maxLC+i);
    }
    MainConcept mc = new MainConcept(mcID,getBasQ.getShort(2),getBasQ.
        getString(3),getBasQ.getString(4),getBasQ.getString(5),getBasQ.
        getString(6),getBasQ.getShort(7), getBasQ.getShort(8), getBasQ.
        getShort(9), getLeaQ.getString(2), getLeaQ.getString(3),
        getLeaQ.
        getString(4),getLeaQ.getString(5), getLeaQ.getShort(6),getLeaQ.
        getShort(7),getLeaQ.getShort(8),
        mSIDs, getMaiQ.getShort(2),getMaiQ.getShort(3),lcIDs,ecIDs);
    getBasQ.closeStatement();
    getLeaQ.closeStatement();
}

```

```

        getLeaQ.closeStatement();
        return mc;
    }
}

```

Class MainContent

```

package lozpt1;

import java.awt.*;
import javax.swing.*;
import com.borland.jbcl.layout.*;
import java.awt.event.*;
import javax.swing.tree.*;

public class MainContent extends JFrame {
    private XYLayout xYLayout1 = new XYLayout();
    private JPanel MainTextP1 = new JPanel();
    private XYLayout xYLayout11 = new XYLayout();
    private JPanel MTextP2 = new JPanel();
    private JPanel jPanel1 = new JPanel();
    private JPanel jPanel2 = new JPanel();
    private JLabel jLabel1 = new JLabel();
    private GridLayout gridLayout1 = new GridLayout();
    private BorderLayout borderLayout1 = new BorderLayout();
    private JScrollPane jScrollPane1 = new JScrollPane();
    private JTextPane jTextPanel = new JTextPane();
    private JTree jTree1 = new JTree();
    private JButton jButton1 = new JButton();
    private XYLayout xYLayout2 = new XYLayout();
    private JButton jButton2 = new JButton();

    public MainContent() {
        try {
            jbInit();
        }
        catch(Exception e) {
            e.printStackTrace();
        }
    }

    private void jbInit() throws Exception {
        MTextP2.setBorder(BorderFactory.createLoweredBevelBorder());
        MTextP2.setLayout(borderLayout1);

        this.getContentPane().setLayout(xYLayout1);
        MainTextP1.setBorder(BorderFactory.createRaisedBevelBorder());
        MainTextP1.setLayout(xYLayout11);
        xYLayout1.setWidth(514);
        xYLayout1.setHeight(532);
        this.getContentPane().setBackground(Color.pink);
        this.setDefaultCloseOperation(DO_NOTHING_ON_CLOSE);
        this.setTitle("LOZ- We Learn Object-Z");
        jPanel1.setBorder(BorderFactory.createLoweredBevelBorder());
        jPanel1.setLayout(xYLayout2);
        jPanel2.setBackground(Color.pink);
        jPanel2.setBorder(BorderFactory.createLoweredBevelBorder());
        jPanel2.setLayout(gridLayout1);
        jLabel1.setBackground(Color.lightGray);
        jLabel1.setFont(new java.awt.Font("Monospaced", 1, 18));
        jLabel1.setForeground(Color.red);
        jLabel1.setHorizontalAlignment(SwingConstants.CENTER);
    }
}

```



```

        jLabel1.setText("Main Contents - What you can learn in LOZ");
        jButton1.setFont(new java.awt.Font("SansSerif", 0, 12));
        jButton1.setForeground(Color.red);
        jButton1.setText("Quit");
        jButton2.setFont(new java.awt.Font("SansSerif", 0, 12));
        jButton2.setForeground(Color.red);
        jButton2.setText("Proceed");
        MainTextP1.add(jPanell1, new XYConstraints(6, 488, 493,
31));
        jPanell1.add(jButton1, new XYConstraints(45, 1, 85, -1));
        jPanell1.add(jButton2, new XYConstraints(350, 0, -1, -1));
        MainTextP1.add(MTextP2, new XYConstraints(5, 39, 485, 441));
        MTextP2.add(jScrollPane1, BorderLayout.CENTER);
        MainTextP1.add(jPanel2, new XYConstraints(6, 4, 489, 34));
        jPanel2.add(jLabel1, null);
        jScrollPane1.getViewport().add(jTree1, null);
        this.getContentPane().add(MainTextP1,new
XYConstraints(5,1,510,530));
        MutableTreeNode root = new DefaultMutableTreeNode("Learn Object-Z
: LOZ");
        MutableTreeNode les1 = new DefaultMutableTreeNode("Lesson-1:
Formal Specification- Overview");
        MutableTreeNode les2 = new DefaultMutableTreeNode("Lesson-2:
Object-Z Specification- Overview");
        MutableTreeNode les21 = new DefaultMutableTreeNode("Lesson-2.1:
Class Schema- Overview");
        MutableTreeNode les211 = new DefaultMutableTreeNode("Lesson-2.1.1:
State Schema- Overview");
        MutableTreeNode les212 = new DefaultMutableTreeNode("Lesson-2.1.2:
Operation Schema- Overview");
        MutableTreeNode les3 = new DefaultMutableTreeNode("Lesson-3: Class
Schema- More");
        root.insert(les1,0);
        root.insert(les2,1);
        root.insert(les3,2);
        les2.insert(les21,0);
        les21.insert(les211,0);
        les21.insert(les212,1);
        TreeModel model= new DefaultTreeModel(root);

        jTree1.setModel(model);
    }

    void jButton6_actionPerformed(ActionEvent e) {

    }
}

class MainContent_jButton6_actionAdapter implements
java.awt.event.ActionListener {
    private MainContent adaptee;

    MainContent_jButton6_actionAdapter(MainContent adaptee) {
        this.adaptee = adaptee;
    }
    public void actionPerformed(ActionEvent e) {
        adaptee(jButton6_actionPerformed(e));
    }
}

```

Class MainContentCtrl

```
package lozpt1;

import javax.swing.JOptionPane;

public class MainContentCtrl {
    private Mentor M;
    private MainContentUI mcUI;
    public MainContentCtrl(String lesson, Mentor M) {
        this.M=M;
        mcUI= new MainContentUI(lesson,this);
        mcUI.setSize(700,800);
        mcUI.pack();
        Util.U.setCentre(mcUI);
        mcUI.setVisible(true);
    }
    public void selected(short mcID){
        /*int n= JOptionPane.showConfirmDialog(mcUI, "You are selecting a
different lesson. \n "+
        "LOZ may not be able to offer much help", "You are the BOSS
now!", JOptionPane.YES_NO_OPTION);
        if (n== JOptionPane.YES_OPTION){ */
            mcUI.setVisible(false);
            M.startSelLesson(mcID); // works only for 200 (VL) and 1200
(SS?)
        // }
    }
    public void quitPressed(){
        int n= JOptionPane.showConfirmDialog(mcUI, "Leaving LOZ, Good
Bye!",
        "Closing LOZ", JOptionPane.YES_NO_OPTION);
        if (n== JOptionPane.YES_OPTION)
            mcUI.setVisible(false);
    }

    public void proceed(){
        mcUI.setVisible(false);
        M.startLesson(); //lesson ID is already there in Learner records
    }
    public MainContentUI mcUI(){
        return mcUI;
    }
}
```

Class MainContentUI

```
package lozpt1;

import java.awt.*;
import javax.swing.*;
import com.borland.jbcl.layout.*;
import java.awt.event.*;
import javax.swing.tree.*;
import javax.swing.JEditorPane;
import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.JScrollPane;
```

```

import javax.swing.JSplitPane;
import javax.swing.UIManager;

import javax.swing.JTree;
import javax.swing.tree.DefaultMutableTreeNode;
import javax.swing.tree.TreeSelectionModel;
import javax.swing.event.TreeSelectionEvent;
import javax.swing.event.TreeSelectionListener;

import java.net.URL;
import java.io.IOException;
import java.awt.Dimension;
import java.awt.GridLayout;
import javax.swing.border.*;

public class MainContentUI extends JDialog{
    private MainContentCtrl mcCtrl;
    private XYLayout xYLayout1 = new XYLayout();
    private BorderLayout borderLayout1 = new BorderLayout();
    private JTree jTree1 = new JTree();
    private BorderLayout borderLayout2 = new BorderLayout();
    private GridLayout gridLayout1 = new GridLayout();
    private JLabel jLabel1 = new JLabel();
    private JPanel jPanel2 = new JPanel();
    private JPanel jPanel1 = new JPanel();
    private JScrollPane mainContentScrollPane = new JScrollPane(jTree1);
    private JLabel message = new JLabel();
    private JLabel lessonLabel = new JLabel();
    private JPanel jPanel3 = new JPanel();
    private JButton jButton1 = new JButton();
    private JButton jButton2 = new JButton();
    private JLabel message2 = new JLabel();
    private Border border1;
    private BorderLayout borderLayout3 = new BorderLayout();
    private XYLayout xYLayout1 = new XYLayout();
    private JButton Cancel = new JButton();

    public MainContentUI(String lesson, MainContentCtrl mc) {
        try {
            jbInit();
        }
        catch(Exception e) {
            e.printStackTrace();
        }
        mcCtrl=mc;
        lessonLabel.setText(" "+lesson);
    }
    private void jbInit() throws Exception {
        border1 = BorderFactory.createBevelBorder(BevelBorder.LOWERED,
            Color.white,Color.white,new Color(115, 114, 105),new Color(165,
            163, 151));
        this.getContentPane().setLayout(borderLayout2);
        this.getContentPane().setBackground(Color.pink);
        this.setDefaultCloseOperation(DO_NOTHING_ON_CLOSE);
        this.setTitle("LOZ- We Learn Object-Z");
        jLabel1.setBackground(Color.lightGray);
        jLabel1.setFont(new java.awt.Font("Serif", 1, 20));
        jLabel1.setForeground(Color.red);
        jLabel1.setHorizontalAlignment(SwingConstants.CENTER);
        jLabel1.setText("Main Contents - What you can learn in LOZ");
    }
}

```

```

jPanel2.setBackground(Color.pink);
jPanel2.setBorder(BorderFactory.createLoweredBevelBorder());
jPanel2.setLayout(gridLayout1);
jPanel1.setBackground(new Color(210, 223, 177));
jPanel1.setBorder(BorderFactory.createLoweredBevelBorder());
jPanel1.setLayout(borderLayout3);
message.setFont(new java.awt.Font("Serif", 1, 14));
message.setForeground(Color.red);
message.setText("      THE NEXT LESSON IS ==>");
lessonLabel.setFont(new java.awt.Font("Serif", 1, 14));
lessonLabel.setBorder(BorderFactory.createEtchedBorder());
lessonLabel.setText("  Visibility List");
jButton1.setEnabled(false);
jButton1.setFont(new java.awt.Font("SansSerif", 1, 12));
jButton1.setForeground(Color.red);
jButton1.setText("Quit");
jButton1.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(ActionEvent e) {
        jButton1_actionPerformed(e);
    }
});
jButton2.setEnabled(false);
jButton2.setFont(new java.awt.Font("SansSerif", 1, 12));
jButton2.setForeground(Color.red);
jButton2.setText("Proceed");
jButton2.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(ActionEvent e) {
        jButton2_actionPerformed(e);
    }
});
jPanel3.setBackground(new Color(236, 223, 216));
jPanel3.setBorder(border1);
jPanel3.setLayout(xYLayout1);
message2.setText("      To continue, press <PROCEED> ,
    o/w select " + "from tree!");
message2.setFont(new java.awt.Font("SansSerif", 1, 12));
message2.setBorder(BorderFactory.createRaisedBevelBorder());
Cancel.setEnabled(false);
Cancel.setFont(new java.awt.Font("SansSerif", 1, 12));
Cancel.setForeground(Color.red);
Cancel.setText("Cancel");
Cancel.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(ActionEvent e) {
        Cancel_actionPerformed(e);
    }
});
this.getContentPane().add(jPanel2, BorderLayout.NORTH);
jPanel2.add(jLabel1, null);
this.getContentPane().add(jPanel1, BorderLayout.SOUTH);
this.getContentPane().add(mainContentScrollPane, BorderLayout.
    CENTER);

jPanel1.add(jPanel3, BorderLayout.CENTER);
jPanel3.add(jButton1, new XYConstraints(10, 1, -1, -1));
jPanel3.add(jButton2, new XYConstraints(420, 3, -1, -1));
jPanel3.add(Cancel, new XYConstraints(75, 2, 83, -1));
jPanel3.add(lessonLabel, new XYConstraints(172, 4, 234, -1));
jPanel1.add(message, BorderLayout.NORTH);
jPanel1.add(message2, BorderLayout.SOUTH);
//mainContentScrollPane.getViewPort().add(jTree1, null);
MutableTreeNode root = new DefaultMutableTreeNode("Learn Object-Z
: LOZ");

```

```

        addNodes(root);
        TreeModel model= new DefaultTreeModel(root);
        jTree1.setModel(model);
        jTree1.addTreeSelectionListener(new TreeSelectionListener(){
            public void valueChanged(TreeSelectionEvent tse){
                treeSelected(tse);
            }
        });
    }

    void jButton6_actionPerformed(ActionEvent e) {

    }

    void jButton1_actionPerformed(ActionEvent e) {
        mcCtrl.quitPressed();
    }

    void treeSelected(TreeSelectionEvent tse){
        Cancel.setEnabled(true);
        TreePath tp= tse.getNewLeadSelectionPath();
        DefaultMutableTreeNode n = (DefaultMutableTreeNode)
tp.getLastPathComponent();
        Lessons L = (Lessons)n.getUserObject();
        if (L.mcID >0)
            mcCtrl.selected(L.mcID);
    }

    void jButton2_actionPerformed(ActionEvent e) {
        mcCtrl.proceed();
    }

    private void addNodes( MutableTreeNode root){
        MutableTreeNode les1 = new DefaultMutableTreeNode
            (new Lessons("Lesson-1: Formal Specification-
Overview", (short)0));
        MutableTreeNode les2 = new DefaultMutableTreeNode
            (new Lessons("Lesson-2: Object-Z Specification-
Overview", (short)0));
        MutableTreeNode les21 = new DefaultMutableTreeNode
            (new Lessons("Lesson-2.1: Class Schema-
Overview", (short)0));
        MutableTreeNode les211 = new DefaultMutableTreeNode
            (new Lessons("Lesson-2.1.1: Visibility
List", (short)200));
        MutableTreeNode les212 = new DefaultMutableTreeNode
            (new Lessons("Lesson-2.1.2: Schema-
Overview", (short)0));
        MutableTreeNode les213 = new DefaultMutableTreeNode
            (new Lessons("Lesson-2.1.3: State Schema-
Overview", (short)1200));
        MutableTreeNode les3 = new DefaultMutableTreeNode
            (new Lessons("Lesson-3: Class Schema- More", (short)0));
        root.insert(les1,0);
        root.insert(les2,1);
        root.insert(les3,2);
        les2.insert(les21,0);
        les21.insert(les211,0);
        les21.insert(les212,1);
        les21.insert(les213,2);
    }

```

```

class MainContent_jButton6_actionAdapter implements
java.awt.event.ActionListener {
    private MainContentUI adaptee;

    MainContent_jButton6_actionAdapter(MainContentUI adaptee) {
        this.adaptee = adaptee;
    }
    public void actionPerformed(ActionEvent e) {
        adaptee.jButton6_actionPerformed(e);
    }
}
private class Lessons{
    private String title;
    private short mcID;
    public Lessons(String t, short id){
        title=t;mcID=id;
    }
    public String toString(){
        return title;
    }
}

void Cancel_actionPerformed(ActionEvent e) {
    this.setVisible(false);
}
}

```

Class MainFrame

```

package lozpt1;

import java.awt.*;
import javax.swing.*;
import com.borland.jbcl.layout.*;
import java.awt.event.*;

public class MainFrame extends JFrame {
    private XYLayout xYLayout1 = new XYLayout();
    private JPanel mainPane = new JPanel();
    private XYLayout xYLayout2 = new XYLayout();
    private JPanel mainPaneIn = new JPanel();
    private JPanel statusBar = new JPanel();
    private JMenuBar jMenuBar1 = new JMenuBar();
    private JMenu jMenu1 = new JMenu();
    private JButton classDia = new JButton();
    private JButton useCaseDia = new JButton();
    private JButton useCaseDes = new JButton();
    private JButton jButton20 = new JButton();
    private JPanel CSuml = new JPanel();
    private JPanel CaseStudy = new JPanel();
    private JPanel CSDemo = new JPanel();
    private JButton jButton19 = new JButton();
    private XYLayout xYLayout7 = new XYLayout();
    private JLabel jLabel3 = new JLabel();
    private XYLayout xYLayout6 = new XYLayout();
    private JLabel jLabel1 = new JLabel();
    private XYLayout xYLayout8 = new XYLayout();
    private JButton useCaseReaon24 = new JButton();
    private BorderLayout borderLayout1 = new BorderLayout();
    private JPanel toolBar = new JPanel();
}

```

```

private JButton New = new JButton();
private JPanel toolBar3 = new JPanel();
private JButton NoteBook = new JButton();
private GridLayout gridLayout4 = new GridLayout();
private GridLayout gridLayout2 = new GridLayout();
private JButton eMail = new JButton();
private JPanel toolBar1 = new JPanel();
private GridLayout gridLayout5 = new GridLayout();
private JButton LearnerModel = new JButton();
private JButton Select = new JButton();
private GridLayout gridLayout3 = new GridLayout();
private JButton Demo = new JButton();
private JPanel toolBar2 = new JPanel();
private JButton HelpGeneral = new JButton();
private JPanel Additional = new JPanel();
private BorderLayout borderLayout3 = new BorderLayout();
private JLabel staus = new JLabel();
private JMenu jMenu2 = new JMenu();
private JMenu jMenu3 = new JMenu();
private JMenu jMenu4 = new JMenu();
private JMenu jMenu5 = new JMenu();
private JMenuItem jMenuItem1 = new JMenuItem();
private JMenuItem jMenuItem2 = new JMenuItem();
private JMenuItem jMenuItem3 = new JMenuItem();
private JMenuItem jMenuItem4 = new JMenuItem();
private JMenuItem jMenuItem5 = new JMenuItem();
private JMenuItem jMenuItem6 = new JMenuItem();
private JMenuItem jMenuItem7 = new JMenuItem();
private JMenuItem jMenuItem8 = new JMenuItem();
private JMenuItem jMenuItem9 = new JMenuItem();
private JMenuItem jMenuItem10 = new JMenuItem();
private JMenuItem jMenuItem11 = new JMenuItem();
private JMenuItem jMenuItem12 = new JMenuItem();
private JMenuItem jMenuItem13 = new JMenuItem();
private JMenuItem jMenuItem14 = new JMenuItem();
private JMenuItem jMenuItem15 = new JMenuItem();
private JMenuItem jMenuItem16 = new JMenuItem();
private JMenuItem jMenuItem17 = new JMenuItem();
private JMenuItem jMenuItem18 = new JMenuItem();
private JMenu jMenu6 = new JMenu();
private JMenuItem jMenuItem19 = new JMenuItem();
private JPanel jPanel11 = new JPanel();
private XYLayout xYLayout3 = new XYLayout();
private JButton Z = new JButton();
private JButton Replay = new JButton();
private GridLayout gridLayout6 = new GridLayout();
private JTabbedPane pre = new JTabbedPane();
private JTextPane preText = new JTextPane();
private JTextPane presentText = new JTextPane();
private JTextPane postText = new JTextPane();
private JTextPane mainText = new JTextPane();
ImageIcon soundI;
private JPanel NaviPanel = new JPanel();
private JButton Search = new JButton();
private JButton previousLesson = new JButton();
private JButton NextLesson = new JButton();
private JPanel naviBar = new JPanel();
private JButton HelpSubject = new JButton();
private JButton previous = new JButton();
private JButton MainContent = new JButton();
private JButton GoTo = new JButton();

```

```

private GridLayout gridLayout1 = new GridLayout();
private JButton ObjectZ = new JButton();
private JButton Next = new JButton();
private XYLayout xYLayout4 = new XYLayout();
private JButton jButton1 = new JButton();
private JButton jButton2 = new JButton();

public MainFrame() {
    try {
        jbInit();
    }
    catch(Exception e) {
        e.printStackTrace();
    }
}

private void jbInit() throws Exception {
    //soundI = new
ImageIcon(MainFrame.class.getResource("icons/sound.gif"));
xYLayout1.setWidth(896);
xYLayout1.setHeight(510);
this.getContentPane().setBackground(Color.lightGray);
this.setDefaultCloseOperation(DO_NOTHING_ON_CLOSE);
this.setJMenuBar(jMenuBar1);
this.setTitle("LOZ- We will Learn Object-Z");
this.getContentPane().setLayout(xYLayout1);
mainPane.setLayout(xYLayout2);
mainPane.setBorder(BorderFactory.createRaisedBevelBorder());
mainPaneIn.setBorder(BorderFactory.createLoweredBevelBorder());
mainPaneIn.setLayout(borderLayout1);
statusBar.setBorder(BorderFactory.createLoweredBevelBorder());
statusBar.setLayout(xYLayout3);
jMenu1.setText("NoteBook");
classDia.setText("Class Diagram");
useCaseDia.setText("Usecase Diagram");
useCaseDes.setText("Description");
jButton20.setText("Short");
CSuml.setLayout(xYLayout7);
CSuml.setBorder(BorderFactory.createLoweredBevelBorder());
CaseStudy.setBorder(BorderFactory.createRaisedBevelBorder());
CaseStudy.setLayout(xYLayout8);
CSDemo.setBorder(BorderFactory.createEtchedBorder());
CSDemo.setLayout(xYLayout6);
jButton19.setText("Full");
jLabel3.setFont(new java.awt.Font("Dialog", 1, 14));
jLabel3.setText("Demo");
jLabel1.setBackground(Color.red);
jLabel1.setFont(new java.awt.Font("Dialog", 1, 14));
jLabel1.setText("Case Study");
useCaseReaon24.setText("Realizations");
toolBar.setLayout(gridLayout2);
New.setBorder(BorderFactory.createLoweredBevelBorder());
New.setText("New");
New.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(ActionEvent e) {
        New_actionPerformed(e);
    }
});
toolBar3.setBorder(BorderFactory.createEtchedBorder());
toolBar3.setLayout(gridLayout4);
NoteBook.setBorder(BorderFactory.createLoweredBevelBorder());
NoteBook.setText("NoteBook");

```



```

NoteBook.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(ActionEvent e) {
        Notebook_actionPerformed(e);
    }
});
eMail.setText("eMail");
toolBar1.setBorder(BorderFactory.createEtchedBorder());
toolBar1.setLayout(gridLayout5);
LearnerModel.setText("LrModel");
Select.setBorder(BorderFactory.createLoweredBevelBorder());
Select.setText("Select");
Demo.setText("Demo");
toolBar2.setBorder(BorderFactory.createLoweredBevelBorder());
toolBar2.setLayout(gridLayout3);
HelpGeneral.setText("GenHelp");
Additional.setLayout(borderLayout3);
staus.setText("status");
jMenu2.setText("Lesson");
jMenu3.setText("Learner");
jMenu4.setText("Help");
jMenu5.setText("New");
jMenuItem1.setText("Book");
jMenuItem2.setText("Lesson");
jMenuItem3.setText("Page");
jMenuItem4.setText("Open");
jMenuItem5.setText("ReOpen");
jMenuItem6.setText("Close");
jMenuItem7.setText("Contents");
jMenuItem8.setText("BookMark");
jMenuItem9.setText("GoTO");
jMenuItem10.setText("Search");
jMenuItem11.setText("ViewModel");
jMenuItem12.setText("Edit");
jMenuItem13.setText("Select");
jMenuItem14.setText("Select");
jMenuItem15.setText("PassWord");
jMenuItem16.setText("Demo");
jMenuItem17.setText("Lessons");
jMenuItem18.setText("ObjectZ");
jMenu6.setText("Exit");
jMenuItem19.setText("Exit");
jPanell1.setBorder(BorderFactory.createLoweredBevelBorder());
jPanell1.setLayout(gridLayout6);
Z.setText("Z");
Replay.setIcon(soundI);
Replay.setText("Replay");
pre.setTabPlacement(JTabbedPane.BOTTOM);
preText.setText("Conditioning");
presentText.setText("Additional Info");
postText.setText("Exploration");
NaviPanel.setLayout(xYLayout4);
NaviPanel.setBorder(BorderFactory.createRaisedBevelBorder());
Search.setBorder(BorderFactory.createLoweredBevelBorder());
Search.setText("Search");
previousLesson.setFont(new java.awt.Font("Dialog", 1, 25));

previousLesson.setBorder(BorderFactory.createLoweredBevelBorder());
previousLesson.setText("<<");
NextLesson.setFont(new java.awt.Font("Dialog", 1, 25));
NextLesson.setBorder(BorderFactory.createLoweredBevelBorder());
NextLesson.setText(">>");

```

```

naviBar.setBorder(BorderFactory.createRaisedBevelBorder());
naviBar.setLayout(gridLayout1);
HelpSubject.setBorder(BorderFactory.createLoweredBevelBorder());
HelpSubject.setText("SubHelp");
HelpSubject.addActionListener(new java.awt.event.ActionListener()
{
    public void actionPerformed(ActionEvent e) {
        HelpSubject_actionPerformed(e);
    }
});
previous.setFont(new java.awt.Font("Dialog", 1, 25));
previous.setBorder(BorderFactory.createLoweredBevelBorder());
previous.setText("<");
MainContent.setBorder(BorderFactory.createLoweredBevelBorder());
MainContent.setText("Contents");
GoTo.setBorder(BorderFactory.createLoweredBevelBorder());
GoTo.setText("GoTo");
ObjectZ.setBorder(BorderFactory.createLoweredBevelBorder());
ObjectZ.setText("ObjectZ");
Next.setFont(new java.awt.Font("Dialog", 1, 25));
Next.setBorder(BorderFactory.createLoweredBevelBorder());
Next.setText(">");
jButton1.setText("jButton1");
jButton2.setText("jButton2");
this.getContentPane().add(mainPane, new XYConstraints(0, 30, 584,
483));
mainPane.add(statusBar, new XYConstraints(4, 442, -1, 28));
mainPane.add(mainPaneIn, new XYConstraints(7, 5, 565, 397));
mainPaneIn.add(mainText, BorderLayout.CENTER);
mainPane.add(NaviPanel, new XYConstraints(5, 409, -1, 31));
naviBar.add(Search, null);
naviBar.add(GoTo, null);
naviBar.add(ObjectZ, null);
naviBar.add(HelpSubject, null);
naviBar.add(MainContent, null);
NaviPanel.add(previous, new XYConstraints(38, 0, 38, 27));
NaviPanel.add(NextLesson, new XYConstraints(490, 1, 38, 27));
NaviPanel.add(previousLesson, new XYConstraints(0, 0, -1, 27));
NaviPanel.add(Next, new XYConstraints(530, 0, 35, 27));
NaviPanel.add(naviBar, new XYConstraints(78, 0, 414, 29));
CSDemo.add(jButton20, new XYConstraints(207, 9, 66, 31));
CSDemo.add(jButton19, new XYConstraints(115, 11, 72, 29));
CSDemo.add(jLabel11, new XYConstraints(19, 7, 100, -1));
CSDemo.add(jLabel3, new XYConstraints(39, 22, 52, -1));
CaseStudy.add(CSum1, new XYConstraints(4, 61, 297, 144));
this.getContentPane().add(toolBar, new XYConstraints(-7, 0, 589, -
1));
CSum1.add(useCaseDia, new XYConstraints(6, 9, -1, -1));
CSum1.add(useCaseReaon24, new XYConstraints(9, 50, 133, -1));
CSum1.add(useCaseDes, new XYConstraints(154, 9, 125, -1));
CSum1.add(classDia, new XYConstraints(8, 91, 130, -1));
CSum1.add(jButton1, new XYConstraints(151, 50, 133, -1));
CSum1.add(jButton2, new XYConstraints(155, 91, 125, -1));
CaseStudy.add(CSDemo, new XYConstraints(3, 6, 302, 53));
jMenuBar1.add(jMenu1);
jMenuBar1.add(jMenu2);
jMenuBar1.add(jMenu3);
jMenuBar1.add(jMenu4);
jMenuBar1.add(jMenu6);
toolBar.add(toolBar1, null);
toolBar1.add(email, null);

```

```

        toolBar1.add(LearnerModel, null);
        toolBar.add(toolBar2, null);
        toolBar2.add(HelpGeneral, null);
        toolBar2.add(Demo, null);
        toolBar.add(toolBar3, null);
        toolBar3.add(NoteBook, null);
        toolBar3.add(Select, null);
        toolBar3.add(New, null);
        this.getContentPane().add(Additional, new XYConstraints(585, 210,
312, 301));
        jMenu1.add(jMenu5);
        jMenu1.add(jMenuItem4);
        jMenu1.add(jMenuItem5);
        jMenu1.add(jMenuItem6);
        jMenu1.add(jMenuItem19);
        jMenu5.add(jMenuItem1);
        jMenu5.add(jMenuItem2);
        jMenu5.add(jMenuItem3);
        jMenu2.add(jMenuItem7);
        jMenu2.add(jMenuItem8);
        jMenu2.add(jMenuItem9);
        jMenu2.add(jMenuItem10);
        jMenu2.add(jMenuItem13);
        jMenu3.add(jMenuItem11);
        jMenu3.add(jMenuItem12);
        jMenu3.add(jMenuItem14);
        jMenu3.add(jMenuItem15);
        jMenu4.add(jMenuItem16);
        jMenu4.add(jMenuItem17);
        jMenu4.add(jMenuItem18);
        statusBar.add(jPanel1, new XYConstraints(438, 0, 123, 24));
        jPanel1.add(Replay, null);
        jPanel1.add(Z, null);
        statusBar.add(staus, new XYConstraints(4, 0, 432, 22));
        Additional.add(pre, BorderLayout.CENTER);
        pre.add(preText, "Conditioning");
        pre.add(presentText, "Additional Info");
        pre.add(postText, "Exploration");
        this.getContentPane().add(CaseStudy, new XYConstraints(582, 1,
314, 211));
    }

    void NoteBook_actionPerformed(ActionEvent e) {

    }

    void New_actionPerformed(ActionEvent e) {

    }
    void HelpSubject_actionPerformed(ActionEvent e) {

    }
}

```

Class MainLessonCtrl2

```

}
package lozpt1;

import java.util.*;
import javax.swing.UIManager;

```

```

import java.awt.*;
import javax.swing.*;

public class MainLessonCtrl2 {
    private boolean packFrame = false;
    private boolean zState=false;
    private MainLessonUI2 mLUI;
    private Expand exp;
    private mcqDlog mcl;
    public Mentor mentor;
    private int currentPlan =1;
    public MainLessonCtrl2(Mentor M) { //one main constructor
        try {
            mentor = M;
            mLUI = new MainLessonUI2(this);
            mcl= new mcqDlog(mLUI,"",false,mentor);
            //Validate frames that have preset sizes
            //Pack frames that have useful preferred size info, e.g. from
                their layout
            if (packFrame) {
                mLUI.pack();
            }
            else {
                mLUI.validate();
            }
            //Center the window
            Util.U.setCentre(mLUI);
            mLUI.setVisible(true);
        }
        catch(Exception e) {
            e.printStackTrace();
        }
    }
    public void pls(){
        mentor.pls();
    }
    public MainLessonUI2 mLUI(){
        return mLUI;
    }
    public mcqDlog mcqDlog(){
        return mcl;
    }
    public void presentPC(LearnConcept pCpt){
        mLUI.status().setText(" A REVISION LESSON is being
loaded.....");
        mLUI.presentPC(pCpt);
    }
    public void presentLC(LearnConcept lCpt){
        mLUI.status().setText(" A LESSON is being loaded.....");
        mLUI.presentLC(lCpt);
    }

    public void presentEC(ExploreConcept exCpt){
        mLUI.status().setText(" An Exploration Topic is being
loaded.....");
        mLUI.presentEC(exCpt);
    }

    public void mcq1(MCQ mcq, String ms){
        mLUI.status().setText(" An MCQ is being loaded.....");
        mLUI.presentMCQfirst(mcq,ms);
    }
}

```

```

    }
    public void showMCQDlog(){
        Util.U.setRBot(mcl);
        mcl.setVisible(true);
        mcl.refresh();
    }
    public void nextPressed(){
        mentor.next();
    }

    public void expand(){
        exp=new Expand(mentor.getBC(),this);
        mLUI.disableExpand();
    }

    public void expandOK(){
        exp.setVisible(false);
        mLUI.enableExpand();
    }

    public void disableNext(){
        mLUI.disableNext();
    }

    public void enableNext(){
        mLUI.enableNext();
    }
}
/*public void previous(){
    int size=oldMainTexts.size();
    if (naviIndex < size){
        naviIndex++;
        Document D = (Document) oldMainTexts.get(size-naviIndex);
        mLUI.updateMainText(D);
    }
}
*/

/*
public void doEx(MCQ m, short plan){
    mLUI.presentEx(m, plan);
}
*/
//}

```

Class MainLessonUI2

```

package lozpt1;

import java.awt.*;
import javax.swing.*;
import javax.sound.sampled.*;
import com.borland.jbcl.layout.*;
import java.awt.event.*;
import javax.swing.text.*;
import javax.swing.border.*;
import java.applet.*;
import java.net.URL;

```

```

import java.io.IOException;
import java.io.File;
import javax.swing.SwingUtilities;
import java.lang.reflect.*;

public class MainLessonUI2 extends JFrame {
    private MainLessonCtrl2 mlCtrl;
    private LearnConcept lcPt;
    private BasicConcept topCpt;
    private BasicConcept botCpt;
    private BasicConcept zCpt;
    private SourceDataLine line=null;
    private boolean pcState=false;
    private boolean lcState=false;
    private boolean mcqState=false;
    private boolean replay=true;
    private String urlString="http://www-ist.massey.ac.nz/smohan";
    private JPanel mainPane = new JPanel();
    private JPanel mainPaneIn = new JPanel();
    private JMenuBar jMenuBar1 = new JMenuBar();
    private JMenu jMenu1 = new JMenu();
    private JPanel toolBar = new JPanel();
    private JButton NoteBook = new JButton();
    private JButton MainContents = new JButton();
    private JPanel toolBar1 = new JPanel();
    private JButton Demo = new JButton();
    private JButton HelpGeneral = new JButton();
    private JMenu jMenu2 = new JMenu();
    private JMenu jMenu3 = new JMenu();
    private JMenu jMenu4 = new JMenu();
    private JMenu jMenu5 = new JMenu();
    private JMenuItem jMenuItem1 = new JMenuItem();
    private JMenuItem jMenuItem2 = new JMenuItem();
    private JMenuItem jMenuItem3 = new JMenuItem();
    private JMenuItem jMenuItem4 = new JMenuItem();
    private JMenuItem jMenuItem5 = new JMenuItem();
    private JMenuItem jMenuItem6 = new JMenuItem();
    private JMenuItem jMenuItem7 = new JMenuItem();
    private JMenuItem jMenuItem8 = new JMenuItem();
    private JMenuItem jMenuItem9 = new JMenuItem();
    private JMenuItem jMenuItem10 = new JMenuItem();
    private JMenuItem jMenuItem11 = new JMenuItem();
    private JMenuItem jMenuItem12 = new JMenuItem();
    private JMenuItem jMenuItem13 = new JMenuItem();
    private JMenuItem jMenuItem14 = new JMenuItem();
    private JMenuItem jMenuItem15 = new JMenuItem();
    private JMenuItem jMenuItem16 = new JMenuItem();
    private JMenuItem jMenuItem17 = new JMenuItem();
    private JMenuItem jMenuItem18 = new JMenuItem();
    private JMenu jMenu6 = new JMenu();
    private JMenuItem jMenuItem19 = new JMenuItem();
    private JButton Search = new JButton();
    private JButton previousLesson = new JButton();
    private JPanel naviBar = new JPanel();
    private JPanel NaviPanel = new JPanel();
    private JButton MainContent = new JButton();
    private JButton HelpSubject = new JButton();
    private JButton previous = new JButton();
    private JButton GoTo = new JButton();
    private GridLayout gridLayout1 = new GridLayout();
    private JButton ObjectZ = new JButton();

```

```

private Border border1;
private XYLayout xYLayout4 = new XYLayout();
private JScrollPane jScrollPane1 = new JScrollPane();
private JTextPane mainText = new JTextPane();
private JButton LearnerM = new JButton();
private GridLayout gridLayout2 = new GridLayout();
private SourceDataLine currSDL;
private SourceDataLine zSDL;
private SourceDataLine xSDL;
private AudioClip currClip;
//private AudioClip zClip;
//private AudioClip xClip;
private JButton quit = new JButton();
private BorderLayout borderLayout2 = new BorderLayout();
private BorderLayout borderLayout4 = new BorderLayout();
private BorderLayout borderLayout5 = new BorderLayout();
private JLabel status = new JLabel();
private JPanel statusBar = new JPanel();
private BorderLayout borderLayout3 = new BorderLayout();
private BorderLayout borderLayout1 = new BorderLayout();
private JButton next = new JButton();
private JButton NextLesson = new JButton();
private JToggleButton Replay = new JToggleButton();
private JButton Expand = new JButton();

public MainLessonUI2(MainLessonCtrl2 mLCtrl) {
    try {
        mLCtrl = mLCtrl;
        jbInit();
    }
    catch(Exception e) {
        e.printStackTrace();
    }
}

private void jbInit() throws Exception {
    //ImageIcon soundI = new
ImageIcon(MainFrame.class.getResource("icons/sound.gif"));
    border1 =
BorderFactory.createBevelBorder(BevelBorder.RAISED,Color.white,Color.w
hite,new Color(115, 114, 105),new Color(165, 163, 151));
    this.setSize(new Dimension(800, 709));
    this.getContentPane().setBackground(Color.lightGray);
    this.setDefaultCloseOperation(DO_NOTHING_ON_CLOSE);
    this.setJMenuBar(jMenuBar1);
    this.setTitle("LOZ- We shall Learn Object-Z");
    this.getContentPane().setLayout(borderLayout1);
    mainPane.setLayout(borderLayout2);
    mainPane.setBorder(BorderFactory.createRaisedBevelBorder());
    mainPaneIn.setBorder(BorderFactory.createLoweredBevelBorder());
    mainPaneIn.setLayout(borderLayout5);
    jMenu1.setEnabled(false);
    jMenu1.setText("NoteBook");
    toolBar.setLayout(borderLayout4);
    NoteBook.setBackground(new Color(255, 210, 144));
    NoteBook.setEnabled(false);
    NoteBook.setBorder(BorderFactory.createRaisedBevelBorder());
    NoteBook.setText("NoteBook");
    NoteBook.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(ActionEvent e) {
            NoteBook_actionPerformed(e);
        }
    }
}

```

```

    });
    MainContents.setBackground(new Color(255, 210, 66));
    MainContents.setEnabled(false);
    MainContents.setFont(new java.awt.Font("SansSerif", 1, 12));
    MainContents.setForeground(Color.darkGray);
    MainContents.setText("MainContents");
    MainContents.addActionListener(new java.awt.event.ActionListener()
{
    public void actionPerformed(ActionEvent e) {
        MainContents_actionPerformed(e);
    }
});
    toolBar1.setBorder(BorderFactory.createEtchedBorder());
    toolBar1.setLayout(gridLayout2);
    Demo.setBackground(new Color(255, 210, 144));
    Demo.setEnabled(false);
    Demo.setText("Demo");
    HelpGeneral.setBackground(new Color(255, 210, 144));
    HelpGeneral.setEnabled(false);
    HelpGeneral.setBorder(BorderFactory.createRaisedBevelBorder());
    HelpGeneral.setText("GenHelp");
    jMenu2.setEnabled(false);
    jMenu2.setText("Lesson");
    jMenu3.setEnabled(false);
    jMenu3.setText("Learner");
    jMenu4.setEnabled(false);
    jMenu4.setText("Help");
    jMenu5.setEnabled(false);
    jMenu5.setText("New");
    jMenuItem1.setText("Book");
    jMenuItem2.setText("Lesson");
    jMenuItem3.setText("Page");
    jMenuItem4.setText("Open");
    jMenuItem5.setText("ReOpen");
    jMenuItem6.setText("Close");
    jMenuItem7.setText("Contents");
    jMenuItem8.setText("BookMark");
    jMenuItem9.setText("GoTO");
    jMenuItem10.setText("Search");
    jMenuItem11.setText("ViewModel");
    jMenuItem12.setText("Edit");
    jMenuItem13.setText("Select");
    jMenuItem14.setText("Select");
    jMenuItem15.setText("PassWord");
    jMenuItem16.setText("Demo");
    jMenuItem17.setText("Lessons");
    jMenuItem18.setText("ObjectZ");
    jMenuItem6.setText("Exit");
    jMenuItem19.setText("Exit");
    //Replay.setIcon(soundI);
    this.validate();
    Search.setBackground(new Color(171, 233, 131));
    Search.setEnabled(false);
    Search.setForeground(Color.darkGray);
    Search.setBorder(BorderFactory.createLoweredBevelBorder());
    Search.setText("Search");
    previousLesson.setBackground(new Color(171, 233, 131));
    previousLesson.setEnabled(false);
    previousLesson.setFont(new java.awt.Font("Dialog", 1, 25));

previousLesson.setBorder(BorderFactory.createLoweredBevelBorder());

```



```

previousLesson.setText("<<");
naviBar.setBorder(BorderFactory.createRaisedBevelBorder());
naviBar.setLayout(gridLayout1);
NaviPanel.setLayout(xYLayout4);
NaviPanel.setBackground(new Color(236, 233, 144));
NaviPanel.setBorder(BorderFactory.createRaisedBevelBorder());
MainContent.setBackground(new Color(171, 233, 131));
MainContent.setEnabled(false);
MainContent.setForeground(Color.darkGray);
MainContent.setBorder(BorderFactory.createLoweredBevelBorder());
MainContent.setText("Contents");
HelpSubject.setBackground(new Color(171, 233, 131));
HelpSubject.setEnabled(false);
HelpSubject.setForeground(Color.darkGray);
HelpSubject.setBorder(BorderFactory.createLoweredBevelBorder());
HelpSubject.setText("SubHelp");
HelpSubject.addActionListener(new java.awt.event.ActionListener()
{
    public void actionPerformed(ActionEvent e) {
        HelpSubject_actionPerformed(e);
    }
});
previous.setBackground(new Color(171, 233, 131));
previous.setEnabled(false);
previous.setFont(new java.awt.Font("Dialog", 1, 25));
previous.setBorder(BorderFactory.createLoweredBevelBorder());
previous.setText("<");
previous.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(ActionEvent e) {
        previous_actionPerformed(e);
    }
});
GoTo.setBackground(new Color(171, 233, 131));
GoTo.setEnabled(false);
GoTo.setForeground(Color.darkGray);
GoTo.setBorder(BorderFactory.createLoweredBevelBorder());
GoTo.setText("GoTo");
ObjectZ.setBackground(new Color(171, 233, 131));
ObjectZ.setEnabled(false);
ObjectZ.setForeground(Color.darkGray);
ObjectZ.setBorder(BorderFactory.createLoweredBevelBorder());
ObjectZ.setText("ObjectZ");
mainText.setBackground(new Color(255, 210, 175));
mainText.setFont(new java.awt.Font("SansSerif", 0, 30));
mainText.setForeground(Color.orange);
mainText.setCaretColor(Color.red);
mainText.setEditable(false);
LearnerM.setBackground(new Color(255, 210, 66));
LearnerM.setFont(new java.awt.Font("SansSerif", 1, 12));
LearnerM.setForeground(Color.darkGray);
LearnerM.setBorder(BorderFactory.createRaisedBevelBorder());
LearnerM.setText("ViewMyRecs");
LearnerM.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(ActionEvent e) {
        LearnerM_actionPerformed(e);
    }
});
quit.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(ActionEvent e) {
        quit_actionPerformed(e);
    }
}

```

```

});
quit.setText("QUIT");
quit.setForeground(Color.darkGray);
quit.setFont(new java.awt.Font("Dialog", 1, 12));
quit.setBackground(new Color(255, 197, 177));
toolBar.setBorder(BorderFactory.createRaisedBevelBorder());
status.setBackground(new Color(236, 184, 144));
status.setFont(new java.awt.Font("Serif", 1, 14));
status.setForeground(Color.blue);
status.setBorder(BorderFactory.createLoweredBevelBorder());
statusBar.setBorder(BorderFactory.createLoweredBevelBorder());
statusBar.setLayout(borderLayout3);
next.setBackground(new Color(171, 203, 46));
next.setFont(new java.awt.Font("Dialog", 1, 25));
next.setBorder(BorderFactory.createLoweredBevelBorder());
next.setText(">");
next.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(ActionEvent e) {
        next_actionPerformed(e);
    }
});
NextLesson.setBackground(new Color(171, 233, 131));
NextLesson.setEnabled(false);
NextLesson.setFont(new java.awt.Font("Dialog", 1, 25));
NextLesson.setBorder(BorderFactory.createLoweredBevelBorder());
NextLesson.setText(">>");
jMenuBar1.setEnabled(false);
Replay.setBackground(new Color(236, 255, 223));
Replay.setFont(new java.awt.Font("Dialog", 1, 12));
Replay.setForeground(Color.darkGray);
Replay.setText("STOPplay");
Replay.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(ActionEvent e) {
        Replay_actionPerformed(e);
    }
});
Expand.setBackground(new Color(255, 210, 66));
Expand.setEnabled(false);
Expand.setFont(new java.awt.Font("SansSerif", 1, 12));
Expand.setBorder(BorderFactory.createRaisedBevelBorder());
Expand.setText("Expand");
Expand.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(ActionEvent e) {
        Expand_actionPerformed(e);
    }
});
naviBar.add(Search, null);
naviBar.add(GoTo, null);
naviBar.add(ObjectZ, null);
naviBar.add(HelpSubject, null);
naviBar.add(MainContent, null);
NaviPanel.add(next, new XYConstraints(454, 3, 40, 29));
NaviPanel.add(NextLesson, new XYConstraints(501, 3, 34, 29));
NaviPanel.add(previous, new XYConstraints(130, 0, 30, 30));
NaviPanel.add(naviBar, new XYConstraints(165, 0, -1, 33));
this.getContentPane().add(statusBar, BorderLayout.SOUTH);
statusBar.add(status, BorderLayout.CENTER);
NaviPanel.add(quit, new XYConstraints(2, 2, -1, -1));
NaviPanel.add(previousLesson, new XYConstraints(90, 0, -1, 34));
NaviPanel.add(Replay, new XYConstraints(581, 6, 94, -1));
mainPaneIn.add(jScrollPane1, BorderLayout.CENTER);

```

```

mainPaneIn.add(NaviPanel, BorderLayout.SOUTH);
jScrollPane1.getViewport().add(mainText, null);
this.getContentPane().add(toolBar, BorderLayout.NORTH);
jMenuBar1.add(jMenu1);
jMenuBar1.add(jMenu2);
jMenuBar1.add(jMenu3);
jMenuBar1.add(jMenu4);
jMenuBar1.add(jMenu6);
jMenu6.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(ActionEvent e) {
        jMenu6_actionPerformed(e);
    }
});
toolBar.add(toolBar1, BorderLayout.CENTER);
toolBar1.add(Expand, null);
toolBar1.add(LearnerM, null);
toolBar1.add(MainContents, null);
toolBar1.add(Demo, null);
toolBar1.add(NoteBook, null);
toolBar1.add(HelpGeneral, null);
//toolBar1.add(Expand, null);
this.getContentPane().add(mainPane, BorderLayout.CENTER);
mainPane.add(mainPaneIn, BorderLayout.CENTER);
jMenu1.add(jMenu5);
jMenu1.add(jMenuItem4);
jMenu1.add(jMenuItem5);
jMenu1.add(jMenuItem6);
jMenu1.add(jMenuItem19);
jMenu5.add(jMenuItem1);
jMenu5.add(jMenuItem2);
jMenu5.add(jMenuItem3);
jMenu2.add(jMenuItem7);
jMenu2.add(jMenuItem8);
jMenu2.add(jMenuItem9);
jMenu2.add(jMenuItem10);
jMenu2.add(jMenuItem13);
jMenu3.add(jMenuItem11);
jMenu3.add(jMenuItem12);
jMenu3.add(jMenuItem14);
jMenu3.add(jMenuItem15);
jMenu4.add(jMenuItem16);
jMenu4.add(jMenuItem17);
jMenu4.add(jMenuItem18);
}

void NoteBook_actionPerformed(ActionEvent e) {

}

void Expand_actionPerformed(ActionEvent e) {
    mlCtrl.expand();
}

public void HelpSubject_actionPerformed(ActionEvent e) {
}

public void next_actionPerformed(ActionEvent e){
    replay=false;
    Replay.setText("Replay");
    if (currClip!=null) {currClip.stop(); currClip=null;}
    mlCtrl.nextPressed();
}

```

```

    }

    public void previous_actionPerformed(ActionEvent e) {
        //mlCtrl.previous();
    }

    public void updateMainText(Document D){
        mainText.setText("");
        mainText.setDocument(D);
    }

    public void LearnerM_actionPerformed(ActionEvent e) {
        replay=false;
        Replay.setText("Replay");
        if (currClip!=null) {currClip.stop(); currClip=null;}
        mlCtrl.pls();
    }

    private void addBCtoTArea(JTextPane TP, BasicConcept bCpt, int
tbox,int plan){
        AttributeSet.aSet.setStyle(tbox,plan);
        presentBC1(TP,bCpt,tbox);
    }
    private void addLCtoTArea(JTextPane TP,int tbox, int plan, ImageIcon
img1,
                                ImageIcon img2){
        AttributeSet.aSet.setStyle(tbox,plan,img1,img2);
        presentLC1(TP);
    }

    private void clearAll(){
        mainText.setText("");
        if (currClip!=null) {currClip.stop(); currClip=null;}
        // if (zClip!=null) {zClip.stop();zClip=null;}
        // if (xClip!=null) {xClip.stop();xClip=null;}
    }

    public void presentBCinMain(BasicConcept bCpt){
        addBCtoTArea(mainText,bCpt,1,1);
    }

    public void presentLCinMain(){
        ImageIcon n=lCpt.getNotation().getNotaImg();
        ImageIcon e=lCpt.getExample().getExampleImg();
        addLCtoTArea(mainText,1,1,n,e);
        final String bID=Integer.toString(lCpt.getBCID());
        currClip=Util.U.setClip("head"+bID+".wav"); //make sure this is not
available if blink
        //zClip=Util.U.setClip("z"+bID+".wav");
        //xClip=Util.U.setClip("exp"+bID+".wav");
        replay=true;
        Replay.setText("STOPplay");
        if (currClip!=null)
            currClip.play();
    }

    public void presentBC1(JTextPane TP, BasicConcept bCpt, int tbox){
        //mainText.setFont(new Font("Serif",Font.PLAIN,24));
        //TP.setCaretPosition(carPos1[1][tbox-1]);
    }

```

```

        Util.U.appendToTextPane(TP,
bCpt.getHead()+"\n",AttriSet.aSet.mainTextAs[0]);
        //carPos1[1][tbox-1]=TP.getCaretPosition();
        Util.U.appendToTextPane(TP," ###
"+bCpt.getWhat()+"\n",AttriSet.aSet.mainTextAs[1]);
        //carPos1[2][tbox-1]=TP.getCaretPosition();
        Util.U.appendToTextPane(TP," ###
"+bCpt.getWhy()+"\n",AttriSet.aSet.mainTextAs[2]);
        //carPos1[3][tbox-1]=TP.getCaretPosition();
        String howT=bCpt.getHow();
        if (!howT.equals("omit"))
            Util.U.appendToTextPane(TP,howT,
AttriSet.aSet.mainTextAs[3]);
        //carPos1[4][tbox-1]=TP.getCaretPosition();
    }
    public void presentLC1(JTextPane TP){ // from addLCtoTArea
        presentBC1(TP,lCpt,1); //tbox-1=0
        //TP.setCaretPosition(carPos[4][0]);
        String notationT=lCpt.getNotation().getNotaText();
        if (!notationT.equals("omit")) {
            Util.U.appendToTextPane(TP," \n Notation : >> ",
AttriSet.aSet.mainTextAs[6]);
            //carPos[5][0]=mainText.getCaretPosition();
            Util.U.appendToTextPane(TP, notationT + "\n",
AttriSet.aSet.mainTextAs[7]);
            //ImageIcon n=lCpt.getNotation().getNotaImg();
            //carPos[5][0]=mainText.getCaretPosition();
            Util.U.appendImgToTextPane(TP,AttriSet.aSet.image1); //should
insert notation;
            //TP.insertIcon(n);
            //carPos[6][0]=0; carPos[7][0]=mainText.getCaretPosition();
        }
        String ex=lCpt.getExample().getExampleText();
        if (!ex.equals("omit")) {
            Util.U.appendToTextPane(TP," \n Example : >> ",
AttriSet.aSet.mainTextAs[8]);
            //carPos[8][0]=mainText.getCaretPosition();
            Util.U.appendToTextPane(TP,ex +
"\n",AttriSet.aSet.mainTextAs[9]);
            //carPos[8][0]=mainText.getCaretPosition();
            Util.U.appendImgToTextPane(TP,AttriSet.aSet.image2); //should
insert notation;
            //TP.insertIcon(lCpt.getExample().getExampleImg());
        }
        //carPos[9][0]=0;mainText.getCaretPosition();
        TP.setCaretPosition(1);
    }

    public void presentPC(final LearnConcept lCpt){
        this.lCpt=lCpt;
        topCpt= lCpt.getOOUmlCpt();
        botCpt= lCpt.getAddiCpt();
        clearAll();
        status.setText(" A UML concept related to VISIBILITY LIST is being
presented; press > to move to next lesson");
        AttriSet.aSet.setPCmainT();
        //carPos[0][0]=mainText.getCaretPosition();
        Util.U.appendToTextPane(mainText," Before learning VISIBILITY LIST,
we shall refresh the related UML concept. \n",
            AttriSet.aSet.mainTextAs[0]);
        //carPos[1][0]=mainText.getCaretPosition();

```

```

    // mcqState=false; pcState=true; lcState=false;
    presentLCinMain();
}

/*
public void presentPC(final LearnConcept lCpt){
    this.lCpt=lCpt;
    status.setText(" The pre condition lesson is being presented. Press
> to move to next lesson");
    try {
        URL url = new URL("http://www-
ist.massey.ac.nz/smohan/LOZdata/preVL.htm");
        mainText.setPage(url);
    } catch (Exception e){
        System.out.print(e.getMessage());
    }
}

public void Replay_actionPerformed(ActionEvent e) {
    if (currClip==null)
        JOptionPane.showMessageDialog(this, "Nothing Available
Now", "Empty", JOptionPane.OK_OPTION);
    else if (replay){
        replay=false;
        Replay.setText("Replay");
        if (currClip!=null) currClip.stop();
    }else{
        replay=true;
        Replay.setText("STOPplay");
        if (currClip!=null) currClip.play();
        //if (zClip!=null) zClip.stop();
        //if (xClip!=null) xClip.stop();
    }
}

public void presentLC(final LearnConcept lc){ //from Mentor/Ctrl
    lCpt=lc;
    topCpt= lCpt.getOOumlCpt();
    botCpt= lCpt.getAddiCpt();
    zCpt=lCpt.getZCpt();
    clearAll();
    status.setText(" A sub lesson on VISIBILITY LIST is being
presented; press > to move to next lesson");
    mcqState=false; pcState=false; lcState=true;
    presentLCinMain();
}

public void presentEC(ExploreConcept exCpt){
    clearAll();
    status.setText(" Some topics that you may discuss with your peers
are being presented; press > to move to next lesson");
    presentBCinMain(exCpt);
}
public void disableNext(){
    next.setEnabled(false);
}
public void enableNext(){
    next.setEnabled(true);
}
public void disableExpand(){
    Expand.setEnabled(false);
}

```

```

    }
    public void enableExpand(){
        Expand.setEnabled(true);
    }

    public void presentMCQfirst(MCQ mcq, String ms){
        mcqState=true; pcState=false; lcState=false;
        clearAll();
        AttributeSet.aSet.setNote(mcq.getImage()); //image not usefull, aSet
        refined later
        Util.U.appendToTextPane(mainText,"Answer the following MCQ (at
        Level "+
                                Integer.toString(mcq.d3Level()+ " ). \n" ,
                                AttributeSet.aSet.botNote);
        presentMCQ(mcq,ms);
    }

    public void presentMCQ(MCQ mcq, String ms){
        AttributeSet.aSet.setStyle(1,1);
        Util.U.appendToTextPane(mainText," QUESTION \n",AttributeSet.aSet.
            heading0); Util.U.appendToTextPane(mainText, mcq.question()
            +"\n",AttributeSet.aSet.paral);
        if (mcq.getImage()!=null){
            SimpleAttributeSet imagel = new SimpleAttributeSet( );
            StyleConstants.setAlignment(imagel,StyleConstants.ALIGN_CENTER);
            StyleConstants.setIcon(imagel,mcq.getImage());
            Util.U.appendImgToTextPane(mainText,imagel); //should insert
            image; not from aSet
        }
        Util.U.appendToTextPane(mainText,"\n ANSWERS \n",AttributeSet.aSet.
            heading0);
        for (int i=1; i<5;i++)
            Util.U.appendToTextPane(mainText,"[ Answer "+String.valueOf(i)+
            ] "+mcq.answer(i).answer()+"\n", AttributeSet.aSet.paral);
        Util.U.appendToTextPane(mainText,"\n ==> Select the most suitable
            answer in the Dialog Box? ", AttributeSet.aSet.para2);
        mainText.setCaretPosition(1);
        status.setText(" An MCQ on VISIBILITY LIST is being presented.
            Select suitable answer.");
        next.disable();
        mlCtrl.showMCQDlog();
    }
    public void showZbox(){
    }
    public void disFor_actionPerformed(ActionEvent e) {
        new WebView(urlString);
    }
    public JLabel status(){
        return status;
    }
    public void jMenuItem6_actionPerformed(ActionEvent e) {
        int n= JOptionPane.showConfirmDialog(this, "Your records are being
        written, Wait....",
                                "Closing LOZ",
                                JOptionPane.YES_NO_OPTION);
        if (n== JOptionPane.NO_OPTION)
            this.dispose();
    };
    public void quit_actionPerformed(ActionEvent e) {
        int n= JOptionPane.showConfirmDialog(this, "Your records are being
        written, Wait....", "Closing LOZ", JOptionPane.YES_NO_OPTION);
    }

```

```

        if (n== JOptionPane.NO_OPTION){
            int n1= JOptionPane.showConfirmDialog(this, "Do you really want
to quit? (if you restart, give different user name)","Really Closing
LOZ", JOptionPane.YES_NO_OPTION);
            if (n1== JOptionPane.YES_OPTION){
                if (currClip!=null)
                    currClip.stop();
                this.dispose();
            }
        }
    }
}
public void MainContents_actionPerformed(ActionEvent e) {
    mlCtrl.mentor.mcCtrl().mcUI().setVisible(true);
}
}
}

```

Class MCQ

```

package lozpt1;
import javax.swing.*;

public class MCQ {
    private short mcqID;
    private short level; //0-4 for MS1, 3-7 for MS2, 6-9 for MS3 ;
1*10+5
    private short scufLevel=0; //1-3 depend on different scuffolding
levels-redundant (dependMS)
    private String question;
    private String rTop;
    private String rBot;
    private String imgFileName;
    private static short maxAns=4;
    private McqAnswer[] answers = new McqAnswer[maxAns];
    private short correct;
    private short mSid;
    private String feed1;
    private String jCodeFile; //javaCode
    private String zFeed;
    private BasicConcept helpCpt=null;
    //answerIDs need not to be given
    public MCQ(short mId, short l, String q, String rT, String rB,
        String img, short cor, short msid, String f1, String f2, String
        zF, short hcID) {
        mcqID=mId; question=q; rTop=rT; rBot=rB; level=l; correct=cor;
        imgFileName=img; feed1=f1;
        jCodeFile=f2; zFeed=zF; mSid =msid;
        helpCpt=BasicConceptDB.bcDB().findaBasicConcept(hcID);
        for (short i=0; i<maxAns; i++)
            answers[i]= McqAnswerDB.mcqADB().
                findaMcqAnswer(mId,(short)(i+1));
        /* Remove 'correct' column from mcqAnswer
        for (short i=0; i<maxAns; i++)
            answers[i].setCorrect((i==cor));
        */
    }
    public MentalState ms(){
        return MentalStateDB.msDB().findaMentalState(mSid);
    }
    public String question(){

```



```

    return question;
}
public ImageIcon getImage(){
    String path = LozGlobal.lozG().path();
    if (imgFileName.equals("omit"))
        return null;
    else
        return new ImageIcon(path+imgFileName);
}
public short mcqID(){
    return mcqID;
}
public String rTop(){
    return rTop;
}
public String rBot(){
    return rBot;
}
public String feed1(){
    return feed1;
}
public String jCodeUrl(){
    String u="file:///"+LozGlobal.lozG().path()+jCodeFile;
    return u;
}
public String zFeed(){
    return zFeed;
}
public BasicConcept getHelpCpt(){
    return helpCpt;
}
public McqAnswer answer(int i){
    return answers[i-1];
}
public short correct(){
    return correct;
}
public short level(){
    return level;
}
public short d3Level(){ //level 1- is low; level-3 is high
    short d3l;
    if (level<64)
        d3l=1;
    else
        if (level < 74)
            d3l=2;
        else
            d3l=3;
    return d3l;
}
public String whyWorCfull(){
    String M= " Why an answer is correct and others wrong?" +
        "\n ----- \n\n";
    M=M+" Answer "+Integer.toString(correct)+" is correct. \n" +
        this.answers[correct-1].whyWorC() +
        "\n===== \n";
    for (int i=1;i<5;i++)
        if (i!=correct)
            M=M+ "Answer "+Integer.toString(i)+ " is Wrong. Because \n"+
this.answers[i-1].whyWorC()+

```

```

        "\n===== \n";
    return M;
}
}

```

Class McqAnswer

```

package lozpt1;

public class McqAnswer {
    private short mcqID;
    private short aID;
    private String answer;
    private boolean correct; // for correct answer misconception is null
    private String whyWorC;
    private String zFeed;
    private short misCid=0; // for correct answer misconception is null
    // The c will be removed from constructor
    public McqAnswer(short mID,short aid, String ans, boolean c,
        String wWR, String zF, short mcID) {
        mcqID=mID; aID=aid; answer=ans; correct=c;
        whyWorC=wWR; zFeed=zF; misCid=mcID;
    }
    public MisConception misC(){
        return MisConceptDB.mcDB().findaMisConcept(misCid);
    }
    public String answer(){
        return answer;
    }
    public String whyWorC(){
        return whyWorC;
    }
    public String zFeed(){
        return zFeed();
    }

    public void setCorrect(boolean c){
        correct=c;
    }
}

```

Class McqAnswerDB

```

package lozpt1;

import com.borland.dx.dataset.*;
import com.borland.dx.sql.dataset.*;
import java.sql.*;
import java.lang.*;

public class McqAnswerDB extends LozDB{
    private static McqAnswerDB mcqADB = new McqAnswerDB();
    public McqAnswerDB() {
        try {

```

```

        jbInit();
    }
    catch(Exception e) {
        e.printStackTrace();
    }
}
private void jbInit() throws Exception {
    LOZdB.setConnection(connD);
}
public static McqAnswerDB mcqADB(){
    return mcqADB;
}

public McqAnswer findaMcqAnswer(short mcqAID, short aID){
    /*short type=3;
    Integer key = new Integer (mcqAID + aID);
    McqAnswer mcqA = (McqAnswer)Cache.find(key,type);
    if (mcqA==null){
        mcqA=findINdb(mcqAID,aID);
        Cache.put(key, mcqA,type);
    }*/
    return findINdb(mcqAID,aID); // mcqA; - is it error in Cache-
map?
}
//the correct column will be removed
public McqAnswer findINdb(short mcqAID, short aID){
    QueryDataSet getLeaQ = new QueryDataSet();
    QueryDescriptor qD = new QueryDescriptor(LOZdB, "SELECT
    \"McqAnswer\".*" + " FROM \"McqAnswer\" WHERE
        \"McqAnswer\".\"mcqID\" = " + mcqAID + " AND \"
    \"McqAnswer\".\"aID\" = " + aID , null, false, Load.ALL);
    getLeaQ.setQuery(qD);
    getLeaQ.executeQuery();
    LOZdB.closeConnection();
    if (getLeaQ.isEmpty())
        return null;
    McqAnswer mcqA= new McqAnswer(mcqAID,aID,getLeaQ.getString(3),
        getLeaQ.getBoolean(4), getLeaQ.getString(5),
        getLeaQ.getString(6), getLeaQ.getShort(7));
    getLeaQ.closeStatement();
    return mcqA;
}
}
}

```

Class McqDB

```

package lozpt1;

import com.borland.dx.dataset.*;
import com.borland.dx.sql.dataset.*;
import java.sql.*;

public extends LozDB{
    private static McqDB mcqDB = new McqDB();
    public McqDB() {
        try {
            jbInit();
        }
        catch(Exception e) {
            e.printStackTrace();
        }
    }
}

```

```

    }
}
private void jbInit() throws Exception {
    LOZdB.setConnection(connD);
}
public static McqDB mcqDB(){
    return mcqDB;
}

public MCQ findaMCQ(short mcqID){
    /*short type=3;
    MCQ mcq = null; //(MCQ)Cache.find(new Short(mcqID),type);
    if (mcq==null){
        mcq=findINdb(mcqID);
        //Cache.put(new Short(mcqID), mcq,type);
    }
    */
    return findINdb(mcqID); //should be mcq, error in Cache-map?
}

public MCQ findINdb(short mcqID){
    QueryDataSet getLeaQ = new QueryDataSet();
    QueryDescriptor qD = new QueryDescriptor(LOZdB, "SELECT \"MCQ\
    *.*\" + \" FROM \"MCQ\" WHERE \"MCQ\".\"mcqID\" = \" + mcqID , null,
    false, Load.ALL);
    getLeaQ.setQuery(qD);
    getLeaQ.executeQuery();
    LOZdB.closeConnection();
    if (getLeaQ.isEmpty())
        return null;
    MCQ mcq= new MCQ(mcqID,getLeaQ.getShort(2),getLeaQ.getString(3),
        getLeaQ.getString(4),getLeaQ.getString(5), getLeaQ.getString(6),
        getLeaQ.getShort(7),getLeaQ.getShort(8), getLeaQ.getString(9),
        getLeaQ.getString(10),getLeaQ.getString(11),getLeaQ.getShort(12)
    );
    getLeaQ.closeStatement();
    return mcq;
}
}
}

```

Class mcqDlog2

```

package lozpt1;

import java.awt.*;
import javax.swing.*;
import com.borland.jbcl.layout.*;
import java.awt.event.*;

public class mcqDlog2 extends JDialog {
    private Mentor theM;
    private JPanel panell = new JPanel();
    private BorderLayout borderLayout1 = new BorderLayout();
    private JPanel jPanel1 = new JPanel();
    private JPanel jPanel2 = new JPanel();
    private JPanel jPanel3 = new JPanel();
    private JLabel jLabel1 = new JLabel();
    private JLabel jLabel2 = new JLabel();
    private XYLayout xYLayout2 = new XYLayout();
}

```

```

private ButtonGroup bg= new ButtonGroup();
private JRadioButton jRadioButton1 = new JRadioButton();
private JRadioButton jRadioButton2 = new JRadioButton();
private JRadioButton jRadioButton3 = new JRadioButton();
private JRadioButton jRadioButton4 = new JRadioButton();
private JButton OK = new JButton();
private FlowLayout flowLayout1 = new FlowLayout();
private JButton Cancel = new JButton();
private BorderLayout borderLayout2 = new BorderLayout();
private JRadioButton jRadioButton5 = new JRadioButton();
private JRadioButton jRadioButton6 = new JRadioButton();
private JRadioButton jRadioButton7 = new JRadioButton();
private JRadioButton jRadioButton8 = new JRadioButton();
private JLabel jLabel3 = new JLabel();
private JLabel jLabel4 = new JLabel();
private JLabel jLabel5 = new JLabel();
private JLabel jLabel6 = new JLabel();
private JLabel jLabel7 = new JLabel();
private JSlider jSlider1 = new JSlider();
private JSlider jSlider2 = new JSlider();
private JSlider jSlider3 = new JSlider();
private JSlider jSlider4 = new JSlider();
private JSlider jSlider5 = new JSlider();
private JSlider jSlider6 = new JSlider();
private JSlider jSlider7 = new JSlider();
private JSlider jSlider8 = new JSlider();
private JLabel jLabel8 = new JLabel();
private JLabel jLabel9 = new JLabel();
private JLabel jLabel10 = new JLabel();
private JLabel jLabel11 = new JLabel();
private JLabel jLabel12 = new JLabel();
private JLabel jLabel13 = new JLabel();
private JLabel jLabel14 = new JLabel();
private JRadioButton jRadioButton9 = new JRadioButton();
private JRadioButton jRadioButton10 = new JRadioButton();
private JRadioButton jRadioButton11 = new JRadioButton();
private JRadioButton jRadioButton12 = new JRadioButton();
private JRadioButton jRadioButton13 = new JRadioButton();

public mcqDlog2(Frame frame, String title, boolean modal, Mentor m) {
    super(frame, title, modal); theM=m;
    try {
        jbInit();
        pack();
    }
    catch(Exception ex) {
        ex.printStackTrace();
    }
}

public mcqDlog2(Mentor m) {
    this(null, "", false, m);
}

private void jbInit() throws Exception {
    panell1.setLayout(borderLayout1);
    this.setTitle("MCQ Answer Dialog");
    jLabel1.setFont(new java.awt.Font("Serif", 3, 30));
    jLabel1.setForeground(Color.red);
    jLabel1.setText("Express Your Belief.");
    jPanel3.setLayout(flowLayout1);
    jPanel3.setBackground(new Color(236, 197, 144));
    jPanel3.setBorder(BorderFactory.createEtchedBorder());
}

```

```

jPanel1.setBackground(new Color(255, 200, 164));
jPanel1.setBorder(BorderFactory.createEtchedBorder());
jPanel1.setLayout(borderLayout2);
jPanel2.setLayout(xYLayout2);
jPanel2.setBackground(new Color(203, 190, 175));
jPanel2.setBorder(BorderFactory.createRaisedBevelBorder());
jRadioButton1.setBackground(new Color(203, 190, 175));
jRadioButton1.setFont(new java.awt.Font("SansSerif", 1, 20));
jRadioButton1.setForeground(new Color(0, 98, 158));
jRadioButton1.setActionCommand("A1");
jRadioButton2.setBackground(new Color(203, 190, 175));
jRadioButton2.setFont(new java.awt.Font("SansSerif", 1, 20));
jRadioButton2.setForeground(new Color(0, 98, 158));
jRadioButton2.setActionCommand("A2");
jRadioButton2.addActionListener(new
java.awt.event.ActionListener() {
    public void actionPerformed(ActionEvent e) {
        jRadioButton2_actionPerformed(e);
    }
});
jRadioButton3.setBackground(new Color(203, 190, 175));
jRadioButton3.setFont(new java.awt.Font("SansSerif", 1, 20));
jRadioButton3.setForeground(new Color(0, 98, 158));
jRadioButton3.setActionCommand("A3");
jRadioButton3.addActionListener(new
java.awt.event.ActionListener() {
    public void actionPerformed(ActionEvent e) {
        jRadioButton3_actionPerformed(e);
    }
});
jRadioButton4.setBackground(new Color(203, 190, 175));
jRadioButton4.setFont(new java.awt.Font("SansSerif", 1, 20));
jRadioButton4.setForeground(new Color(0, 98, 158));
jRadioButton4.setActionCommand("A4");
jRadioButton3.setActionCommand("A3");
OK.setBackground(new Color(255, 175, 125));
OK.setFont(new java.awt.Font("Dialog", 1, 12));
OK.setActionCommand("OK");
OK.setText("OK");
OK.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(ActionEvent e) {
        OK_actionPerformed(e);
    }
});
Cancel.setBackground(new Color(255, 175, 125));
Cancel.setFont(new java.awt.Font("Dialog", 1, 12));
Cancel.setActionCommand("Cancel");
Cancel.setText("Cancel");
Cancel.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(ActionEvent e) {
        Cancel_actionPerformed(e);
    }
});
jLabel2.setBackground(Color.orange);
jRadioButton5.addActionListener(new
java.awt.event.ActionListener() {
    public void actionPerformed(ActionEvent e) {
        jRadioButton5_actionPerformed(e);
    }
});
jRadioButton5.setActionCommand("A1");

```

```

jRadioButton5.setForeground(new Color(0, 98, 158));
jRadioButton5.setFont(new java.awt.Font("SansSerif", 1, 20));
jRadioButton5.setBackground(new Color(203, 190, 175));
jRadioButton6.setActionCommand("A3");
jRadioButton6.addActionListener(new
java.awt.event.ActionListener() {
    public void actionPerformed(ActionEvent e) {
        jRadioButton6_actionPerformed(e);
    }
});
jRadioButton6.setActionCommand("A3");
jRadioButton6.setForeground(new Color(0, 98, 158));
jRadioButton6.setFont(new java.awt.Font("SansSerif", 1, 20));
jRadioButton6.setBackground(new Color(203, 190, 175));
jRadioButton7.addActionListener(new
java.awt.event.ActionListener() {
    public void actionPerformed(ActionEvent e) {
        jRadioButton7_actionPerformed(e);
    }
});
jRadioButton7.setActionCommand("A2");
jRadioButton7.setForeground(new Color(0, 98, 158));
jRadioButton7.setFont(new java.awt.Font("SansSerif", 1, 20));
jRadioButton7.setBackground(new Color(203, 190, 175));
jRadioButton8.setActionCommand("A4");
jRadioButton8.setForeground(new Color(0, 98, 158));
jRadioButton8.setFont(new java.awt.Font("SansSerif", 1, 20));
jRadioButton8.setBackground(new Color(203, 190, 175));
jLabel3.setFont(new java.awt.Font("SansSerif", 0, 20));
jLabel3.setText("Answers");
jLabel4.setFont(new java.awt.Font("Dialog", 0, 20));
jLabel4.setText("[ 1 ]");
jLabel5.setText("[ 2 ]");
jLabel5.setFont(new java.awt.Font("Dialog", 0, 20));
jLabel6.setText("[ 3 ]");
jLabel6.setFont(new java.awt.Font("Dialog", 0, 20));
jLabel7.setText("[ 4 ]");
jLabel7.setFont(new java.awt.Font("Dialog", 0, 20));
jLabel8.setText("Wrong");
jLabel9.setText("Correct");
jLabel10.setText("Not Sure");
jLabel11.setText("100%");
jLabel12.setText("100%");
jLabel13.setText("0%");
jLabel14.setText("0%");
jRadioButton9.setActionCommand("A4");
jRadioButton9.setForeground(new Color(0, 98, 158));
jRadioButton9.setFont(new java.awt.Font("SansSerif", 1, 20));
jRadioButton9.setBackground(new Color(203, 190, 175));
jRadioButton10.setActionCommand("A3");
jRadioButton10.addActionListener(new
java.awt.event.ActionListener() {
    public void actionPerformed(ActionEvent e) {
        jRadioButton10_actionPerformed(e);
    }
});
jRadioButton10.setActionCommand("A3");
jRadioButton10.setForeground(new Color(0, 98, 158));
jRadioButton10.setFont(new java.awt.Font("SansSerif", 1, 20));
jRadioButton10.setBackground(new Color(203, 190, 175));

```

```

jRadioButton11.addActionListener(new
java.awt.event.ActionListener() {
    public void actionPerformed(ActionEvent e) {
        jRadioButton11_actionPerformed(e);
    }
});
jRadioButton11.setActionCommand("A2");
jRadioButton11.setForeground(new Color(0, 98, 158));
jRadioButton11.setFont(new java.awt.Font("SansSerif", 1, 20));
jRadioButton11.setBackground(new Color(203, 190, 175));
jRadioButton12.setActionCommand("A1");
jRadioButton12.setForeground(new Color(0, 98, 158));
jRadioButton12.setFont(new java.awt.Font("SansSerif", 1, 20));
jRadioButton12.setBackground(new Color(203, 190, 175));
jRadioButton13.setBackground(new Color(203, 190, 175));
jRadioButton13.setFont(new java.awt.Font("SansSerif", 1, 20));
jRadioButton13.setForeground(new Color(0, 98, 158));
jRadioButton13.setActionCommand("A3");
jRadioButton13.addActionListener(new
java.awt.event.ActionListener() {
    public void actionPerformed(ActionEvent e) {
        jRadioButton13_actionPerformed(e);
    }
});
jRadioButton13.setActionCommand("A3");
getContentPane().add(panell1);
panell1.add(jPanell1, BorderLayout.SOUTH);
jPanell1.add(Cancel, BorderLayout.WEST);
jPanell1.add(OK, BorderLayout.EAST);
jPanell1.add(jLabel2, BorderLayout.CENTER);
panell1.add(jPanel2, BorderLayout.CENTER);
jPanel2.add(jRadioButton5, new XYConstraints(92, 32, 22, 32));
jPanel2.add(jRadioButton7, new XYConstraints(92, 81, -1, -1));
jPanel2.add(jRadioButton6, new XYConstraints(92, 119, -1, -1));
jPanel2.add(jRadioButton8, new XYConstraints(92, 157, -1, -1));
jPanel2.add(jLabel4, new XYConstraints(15, 39, 50, 24));
jPanel2.add(jLabel5, new XYConstraints(15, 78, 50, 24));
jPanel2.add(jLabel6, new XYConstraints(15, 117, 50, 24));
jPanel2.add(jLabel7, new XYConstraints(15, 156, 50, 24));
jPanel2.add(jSlider1, new XYConstraints(118, 41, 82, 21));
jPanel2.add(jSlider2, new XYConstraints(221, 41, 82, 21));
jPanel2.add(jSlider4, new XYConstraints(118, 81, 82, 21));
jPanel2.add(jSlider3, new XYConstraints(220, 81, 83, 21));
jPanel2.add(jSlider5, new XYConstraints(218, 121, 85, 21));
jPanel2.add(jSlider6, new XYConstraints(118, 121, 83, 21));
jPanel2.add(jSlider7, new XYConstraints(219, 159, 83, 21));
jPanel2.add(jSlider8, new XYConstraints(117, 159, 83, 21));
jPanel2.add(jRadioButton3, new XYConstraints(319, 122, -1, -1));
jPanel2.add(jRadioButton1, new XYConstraints(319, 33, 25, 32));
jPanel2.add(jRadioButton2, new XYConstraints(319, 83, -1, -1));
jPanel2.add(jRadioButton4, new XYConstraints(319, 161, -1, -1));
jPanel2.add(jLabel10, new XYConstraints(193, 5, 65, 35));
jPanel2.add(jLabel11, new XYConstraints(110, 171, 39, 35));
jPanel2.add(jLabel12, new XYConstraints(277, 173, 39, 35));
jPanel2.add(jLabel14, new XYConstraints(214, 171, 25, 35));
jPanel2.add(jLabel13, new XYConstraints(195, 170, 24, 35));
jPanel2.add(jLabel3, new XYConstraints(7, 15, 106, 18));
jPanel2.add(jLabel8, new XYConstraints(98, 8, 45, 35));
jPanel2.add(jLabel9, new XYConstraints(291, 8, 45, 35));
jPanel2.add(jRadioButton9, new XYConstraints(359, 191, -1, -1));
jPanel2.add(jRadioButton10, new XYConstraints(204, 117, -1, -1));

```



```

        jPanel2.add(jRadioButton13, new XYConstraints(204, 157, -1, -1));
        jPanel2.add(jRadioButton11, new XYConstraints(204, 79, -1, -1));
        jPanel2.add(jRadioButton12, new XYConstraints(204, 34, 25, 32));
        panel1.add(jPanel3, BorderLayout.NORTH);
        jPanel3.add(jLabel1, null);
        bg.add(jRadioButton1); bg.add(jRadioButton2);
bg.add(jRadioButton3);
        bg.add(jRadioButton4);

    }
    void jRadioButton2_actionPerformed(ActionEvent e){
    };

    void jRadioButton3_actionPerformed(ActionEvent e){
        };

    void OK_actionPerformed(ActionEvent e) {
        ButtonModel rB= bg.getSelection();
        if (rB!=null){
            char c=rB.getActionCommand().charAt(1);
            rB.setSelected(false);
            short selAnswer= (short)(c-48);
            theM.processAnswer(selAnswer);
        }
    }

    void Cancel_actionPerformed(ActionEvent e) {
        this.setModal(false);this.setVisible(false);
    }

    void refresh(){
        jRadioButton1.setSelected(false);
        jRadioButton2.setSelected(false);
        jRadioButton3.setSelected(false);
        jRadioButton4.setSelected(false);
    }

    void jRadioButton5_actionPerformed(ActionEvent e) {
    }
    void jRadioButton6_actionPerformed(ActionEvent e) {
    }
    void jRadioButton7_actionPerformed(ActionEvent e) {
    }
    void jRadioButton10_actionPerformed(ActionEvent e) {
    }
    void jRadioButton11_actionPerformed(ActionEvent e) {
    }
    void jRadioButton13_actionPerformed(ActionEvent e) {
    }
}

```

Class MentalState

```
package lozpt1;

public class MentalState {
    private short msID;
    private String mentalState; //description
    private static short maxQ=2;
    private short[] mcqIDs=new short[maxQ]; //with decreasing diffLevel
    private static short maxDependMS=1; //only preceeding one
    private short[] dependMSids= new short[maxDependMS]; //weight
    omitted; only for scaffolding
    private short msAbility=0;
    private short bCptID;

    public MentalState(short msid, String des, short bcID, short[]
    mcqIDs,
        short[] dMS, short msA) {
        msID=msid; mentalState=des; bCptID=bcID; msAbility = msA;
        //if ((msID==200) || (msID==210) || (msID==220))
        //maxQ=2;
        for (short i=0; i<maxQ; i++)
            this.mcqIDs[i] = mcqIDs[i];
        for (short i=0; i<maxDependMS; i++)
            dependMSids[i] = dMS[i];
    }
    public BasicConcept bCpt(){
        return BasicConceptDB.bcDB().findaBasicConcept(bCptID);
    }
    public short msID() { return msID;}
    public static short maxQ() { return maxQ;}
    public static short maxDepMS() { return maxDependMS;}
    public String describe() {return mentalState;};
    public MCQ mcq(int i){
        if (i>maxQ)
            return null;
        else
            return McqDB.mcqDB().findaMCQ(mcqIDs[i-1]);
    }
    public MentalState dependMS(int i){ //only i=0
        return MentalStateDB.msDB().findaMentalState(dependMSids[i]);
    }
    public short getSMS(Learner L){
        LearnerMS lms= new
LearnerMSDB().lmsDB().findaLearnerMSDB(L.leaID(),this.msID());
        return lms.getSMS().get();
    }
}
package lozpt1;

import com.borland.dx.dataset.*;
import com.borland.dx.sql.dataset.*;
import java.sql.*;
import java.lang.*;
public class MentalStateDB extends LozDB{
    private static MentalStateDB msDB = new MentalStateDB();
    public MentalStateDB() {
    }
    public static MentalStateDB msDB(){
        return msDB;
    }
}
```

```

    }

    public MentalState findaMentalState(short msID){
        short type=5;
        MentalState ms = (MentalState)Cache.find(new
Short(msID),type);
        if (ms==null){
            ms=findINdb(msID);
            Cache.put(new Short(msID), ms,type);
        }
        return ms;
    }

    public MentalState findINdb(short msID){
        QueryDataSet getLeaQ = new QueryDataSet();
        QueryDescriptor qD = new QueryDescriptor(LOZdB, "SELECT
        \"MentalState\".* \" + \"FROM \"MentalState\" WHERE
        \"MentalState\".\"msID\" = \" + msID,
            null, false, Load.ALL);
        getLeaQ.setQuery(qD);
        getLeaQ.executeQuery();
        LOZdB.closeConnection();
        if (getLeaQ.isEmpty())
            return null;
        short maxQ=MentalState.maxQ();
        short[] mcqIDs= new short[maxQ];
        for (short i=0; i<maxQ; i++)
            mcqIDs[i]= getLeaQ.getShort(4+i);
        short maxDepMS=MentalState.maxDepMS();
        short[] depMSs= new short[maxDepMS];
        for (short i=0; i<maxDepMS; i++)
            depMSs[i]= getLeaQ.getShort(4+maxQ+i);
        MentalState ms= new MentalState(msID,getLeaQ.getString(2),
            getLeaQ.getShort(3), mcqIDs,depMSs,getLeaQ.
            getShort(4+maxQ+maxDepMS));
        getLeaQ.closeStatement();
        return ms;
    }
}

```

Class Mentor

```

package lozpt1;

import java.util.*;
import java.sql.Time;
import javax.swing.*;
import java.awt.*;
import java.io.*;

public class Mentor {
    private Learner aLearner;
    private short threshHold =1; //minutes, should be in Global
    private int maxFBtime=3; //minutes, should be with FB
    private int minFBtime=2; //minutes, should be with FB
    private int globI=0; //
    private byte[] ans= new byte[30]; // simul
    private boolean newL=false;
    private LearnerPreference lp;
}

```

```

private LastLearnRecord aLRec=null;
private CurrentSession cSession=null;
private MainConcept mc=null;
private LearnConcept lc=null;
private MainLessonCtrl2 mLCtrl=null;
private MainContentCtrl mcCtrl=null;
private FBCtrl fbCtrl;
private boolean summShown=false;
private final String firstLesson="Formal Specification- Overview";
private String leaSumm="";
private String lessSumm="";
private LeaSummUI leaSummUI=null;
private LeaInsUI leaInspUI=null;
private String currSummFileName;
public Mentor(Learner L, String leaSumm) {
    aLearner=L;
    lp=aLearner.getLastLP();
    aLRec=aLearner.getLastLearnRecord();
    //currSummFileName="C:\\Mohan\\Xsumm"+L.userID();
    startLessonNewL(leaSumm);
}

public MainLessonCtrl2 mLCtrl(){
    return mLCtrl;
}

public MainContentCtrl mcCtrl(){
    return mcCtrl;
}

public BasicConcept getBC(){
    return (BasicConcept)lc;
}
public CurrentSession getCSession(){
    return cSession;
}
public Learner getLearner(){
    return aLearner;
}
public boolean newL(){
    return newL;
}
public void changeLPsumm(boolean s, boolean o){
    if (s) lp.toggleSummary();
    if (o) lp.toggleSummOptON();
}

// public void nextTopic(Learner L, LearnConcept NextT, MainFrame1
MainF){
    // NextT.checkValid(L);
    //NextT.presentConcept(MainF);
    //}
    // Save current summary to a text file; handle not yet having a
filename; report to statusBar.

private void saveFile(String s, String fName) {
    try
    {
        File file = new File (fName); //open the file
        FileWriter out = new FileWriter(file);
        out.write(s);
    }
}

```

```

        out.close();
    }
    catch (IOException e) {
        System.out.print("Error Saving");
    }
}

private void saveSummFile() {
    // currSummFileName1+=".txt";
    // saveFile(leaSumm,currSummFileName2);
}

public void pls(){
    if (leaSummUI==null){
        leaSummUI=new LeaSummUI(this);
        leaSummUI.setSize(1000,600);
        Util.U.setCentre(leaSummUI);
        leaSummUI.validate();
        leaSummUI.presentWelcome("Hello "+aLearner.userID());
    }
    leaSummUI.setVisible(true);
    //leaSumm+= "\n Summary is being reviewed \n";
    //saveSummFile();
    leaSummUI.pls(leaSumm);
}

public void inspectLM(){
    leaInspUI=new LeaInspUI(this);
    leaInspUI.setSize(500,400);
    Util.U.setCentre(leaInspUI);
    leaInspUI.validate();
    leaInspUI.presentWelcome("The current detail kept in LOZ about " +
        aLearner.userID());
    leaInspUI.setVisible(true);
}

public void startLessonNewL(String lSum){
    leaSumm+=lSum;
    newL=true; //new learner
    //showWD();
    /* simul
    showMC();//Delete this line, if to go to WelcomeDemo
    */
    startSelLesson((short) 200); // simul ;to get VL
}

public void startLessonOldL(){
    showMC();
    // if (lp.summaryON())
        //showShortSummary(); //reactivate these lines if summary
wanted
};

public void showMC(){ //WDctrl call this
    if (newL)
        mcCtrl= new MainContentCtrl(firstLesson,this);
    else
        mcCtrl= new MainContentCtrl(aLRec.nextLesson(),this);
}

```

```

public void showWD(){
    WelcomeDemoCtrl wcCtrl=new WelcomeDemoCtrl(this);
}

private void showShortSummary(){
    String summ= aLRec.getLSummary();
    String well= " WELCOME " + aLearner.userID();
    boolean b=lp.summOptON();
    SummaryCtrl sCtrl = new SummaryCtrl(this,b);
    sCtrl.setWelcome(well);
    sCtrl.setSummary(summ);
    leaSumm=leaSumm+ "\n==> A short summary was shown \n";
    summShown=true;
}

public void showDetailSummary(SummaryCtrl sCtrl){
    Session aLSess= aLRec.getLSession();
    String summ=aLSess.summary();
    String well= " WELCOME " + aLearner.userID();
    Date sD= new Date(aLSess.startTime());
    String dLine= "\n Your Last Session was on      ==> "+
        sD.toString().substring(0,19);
    String tLine = "You have used the system for ==> "+
        aLSess.duration()+ " Minuites \n";
    sCtrl.setSizeUI(800,600);
    sCtrl.setWelcome(well);
    sCtrl.setSummary(dLine,tLine,summ);
    leaSumm=leaSumm+ " ==> You requested a detailed summary also,
        and it was shown \n";
}

public void startLesson(){ // comes from main content page- from
    proceed
    mLCtrl= new MainLessonCtrl2(this);
    if (newL)
        startFirstLesson(); //will not work now
    else
        startOldLesson();
}

public void startFirstLesson(){
    leaSumm=leaSumm+" ==> As recommended by LOZ, you started from the
        first lesson " + firstLesson+ "\n";
    mc=MainConceptDB.mcDB().findaMainConcept(LozGlobal.lozG().firstM
        Cid);
    presentMC();
}

public void startOldLesson(){
    leaSumm=leaSumm+" ==> As recommended by LOZ, you started the
        next lesson " + aLRec.nextLesson()+"\n";
    if (aLRec.getLAction() == 0) // start afresh
        if (aLRec.getLState()==0){ //MC completed, start with next MC
            mc=aLRec.getLastMC(); // new MC is already there
            if (mc != null) // otherwise start with new Topic
                presentMC(); // important
        }
}
}

```

```

public void startSelLesson(short mcID){ //comes from MC, works only
for 200(VL) & 1200(SS?)
/*leaSumm=leaSumm+" ==> CAUTION: You selected a lesson in the
middle, which is not in the order "+ " recommended by LOZ \n
You were already warned about this: LOZ may not be able to
give "+ "propper help \n"; */
mLCtrl= new MainLessonCtrl2(this);
mc=MainConceptDB.mcDB().findaMainConcept(mcID);
presentMC();
}

public void presentMC(){
initCsession(mc);
//cSession.setCurAction((short)10);//show MC
presentPC(); // important
}

public void initCsession(MainConcept mc){
long sT= System.currentTimeMillis(); //start time in mSec
Date stD=new Date(sT);
Time stT=new Time(sT);
//leaSumm+=" ==> You started " + mc.getHead() + " lesson on "+
stD.toString().substring(0,10)+ // " at "+ stD.toString().
substring(11,20)+"\n";
cSession= new CurrentSession(aLearner.getNextSessID(),
aLearner.leaID(),sT, //Session.StartTime
(long)0, (newL)? (short)0 : aLRec.getLSessID(),null,mc);
LearnerMC lmc= LearnerMC.findORcreateLMC(aLearner,mc);
//important
cSession.setCurLMC(lmc);
cSession.setCurActSTime(sT);
cSession.setCurLCSTime(sT);
cSession.setCurState((short)1); // MC state
}

public void warnPC(){
String mainHeading = mc.getHead();
String preHeading = mc.getPreCondition().getHead();
String msg="Next Lesson is ==> " +mainHeading + "\n Before that we
shall just see \n" + "What is " + preHeading;
int n=JOptionPane.showConfirmDialog(mLCtrl.mLUI(), msg , "Pre
Lesson",JOptionPane.OK_CANCEL_OPTION);
if (n==JOptionPane.CANCEL_OPTION){
leaSumm=leaSumm+ " ==> You were shown the following message ##
\n\n" +msg + " \n\n## and you had skipped refreshing
your past knowledge \n";
cSession.setCurAction((short)15); // precondition skipped
next(); //should select next lesson
}
leaSumm=leaSumm+ " ==> Before learning "+ mainHeading + " You
prefer to referesh your past knowledge "+ "and you started to learn
" + preHeading + "\n";
}

public void presentPC(){
cSession.setCurState((short)5); // precondition state
LearnConcept pc = mc.getPreCondition();
// warnPC(); may lead to Action 15
cSession.setCurAction((short)10);//show precondition

```

```

    mLCtrl.presentPC(pc); // important
}

public boolean ms1State(){ //first MS1 state
short state = cSession.getCurState();
return ((state+80)%100==0);
}

public boolean ms2State(){ //first MS1 state
short state = cSession.getCurState();
return ((state+50)%100==0);
}

public boolean ms3State(){ //first MS1 state
short state = cSession.getCurState();
return ((state+20)%100==0);
}

public boolean msState(){ // other MS states, MS1 and MS2
short state = cSession.getCurState();
return
(((state+80)%100==0)||((state+50)%100==0)||((state+20)%100==0));
}

public boolean pcState(){
return (cSession.getCurState()==5); // preC state
}

public boolean lc1State(){ // if LC1 (part of MC)
short state = cSession.getCurState();
return (state==10); //LC1 state
}

public boolean lcState(){ // all LC states including LC1
short action = cSession.getCurAction();
short state = cSession.getCurState();
return ((state+90)%100==0); //LC1-3 states
}

public boolean explState(){
short state = cSession.getCurState();
return (state==400);
}

//fbOKpressed also calls this; o/w that may call fromMS() directly

public void next(){ //preC OR LC1, LC2 or LC3 OR EC finished
//if (!mcqON()) //should be in individual cases
leaSumm+= "\n =====>>>>> NEXT pressed (or evoked
from FeedBackOK) \n";
//saveSummFile(); //simul
if (pcState())
fromPC();//preC finished, LLA initialized from GLA; process first
LC
else if(lcState())
fromLC(); // other LC finished, process first MS
else if (msState())
fromMS(); // ??last LC of an MC finished
else if (explState()){
JOptionPane.showMessageDialog(mLCtrl.mLUI(),
"You have successfully completed the LESSON : "+

```



```

        "\n Try some more MCQs ", "WELDONE, " +aLearner.userID(),
        JOptionPane.OK_OPTION); //nextMC();
        mLCtrl.mLUI().dispose();
        leaSumm+= "You have successfully completed the LESSON : "
            + mc.getHead()+ "\n THANK YOU VERY MUCH "; // simul
        //saveSummFile();//simul
    }
}

public void fromPC() {
    short time=getLessonTime(); // now just gives maximum time for
all;
    short cAction = cSession.getCurAction();
    short cState = cSession.getCurState();
    //if ((cAction==10)|| (cAction==15)) { // a pre concept presented or
skipped
        LocalLA lla=cSession.getCurLMC().getLLA();
        if (lla!=null)
            leaSumm+= " ==> You already have a value for (Local
            Learning Ability) LLA \n "+ lla.fuzz();
        //IMPORTANT, this will check old LLA 7 do GLA*T ==>
            LLA :disabled; now only LLA <-- GLA
        initLLA(time); // time not used now; sets LMC.llla in cSession
        cSession.setCurAction((short)10); //LC1 is being presented
        cSession.setCurState((short)10); // a main lesson, MC-LC1 state
        cSession.setCurActSTime(System.currentTimeMillis());
        cSession.setCurLCSTime(System.currentTimeMillis());
        lc = (LearnConcept) mc;
        //LearnerLC llc=LearnerLC.findORcreateLLC(aLearner,lc); rather
assume from first lc
        //cSession.setCurLLC(llc);
        lessSumm=lc.getLessSum(); //this is lessSumm not leaSumm
        mLCtrl.presentLC(lc);
    }
}

public void initLLA(short time){
    GlobalLA gla=aLearner.gla();
    LocalLA lla= new LocalLA(gla); // just transfer GLA to LLA
    cSession.getCurLMC().setLLA(lla); //old estimate of LLA
simply ignored hereafter
    leaSumm=leaSumm+ lla.fuzz();
    /*
    if ((time>threshHold) && (cSession.getCurAction()==10)){
        lla=new LocalLA(gla, time); //IMPORTANT, this will do GLA*T
==> LLA
        leaSumm=leaSumm+ " ==> You had spent "+ time + "(% scale
            on learning Pre-Lesson and \n" +
            " ==> Your Global Learn Ability was <<" + gla.get()+
            " (%scale) >>\n"+
            " ==> From this, your current Local Learn Ability is
            calculated. LLA = " + lla.get()+ " (%scale)\n"+
            "***** More detailed explanation is given
            below \n"+ lla.fuzz());
    } else {
        lla= new LocalLA(gla.get()); //if skip LLA <-- GLA
        leaSumm=leaSumm+ " ==> + You skipped OR had spent only a
            little time on learning Pre-Lesson \n" +
            " ==> Therefore, your Local Learn Ability will be
            initially your current GLA \n"+
            " ==> That is, your current LLA = " + gla.get()+ " \n";
    }
}

```

```

    }
    /* /*
    LocalLLA oldLla=cSession.getCurLMC().getLLA(); //just discard
the lla, use gla as lla
    if (oldLla!=null){
        leaSumm=leaSumm+ "==> Your old LLA was "+
        Integer.toString(oldLla.get())+ " \n";
        if (oldLla.get()>lla.get())
            leaSumm=leaSumm+ "==> And it was better than the current
            estimate \n" +Integer.toString(lla.get());
    }
    */
}

public void fromLC() {
    short time=getLessonTime(); // now just gives maximum time for all;
    // other LC states, initSMS, goto respective MS1
    setNextMS((short)1); //state will be updated; also sets new LMS (so
new SMS)
    //if (!doMCQ(80)) next();
    LearnerMS lms=cSession.getCurLMS();
    // due to lesson (due to feed back learning -->in fbOKpressed)
    //disable time related predicting LLA*T==>L , but include dynamic;
& SMS*L (<--LLA)==> SMS
    lms.getSMS().predictSMS(cSession.getCurLMC().getLLA(),time,
threshHold); //predict SMS- IMPORTANT
    leaSumm=leaSumm+lms.getSMS().pred();
    presentMCQ(1); //prediction will be handled there
}

public void setNextMS(short msI){ //msI=1,2,3 !s=10-->20, 20 --> 50,
50 --> 80
    MentalState ms= lc.getMS(msI);
    cSession.setCurMS(ms);
    boolean fini=setNewLMS(msI); // create new LMS (so, new SMS)
    short state = cSession.getCurState();
    cSession.setCurScuffL((short)(msI));
    if (fini)
        if (msI==1)
            cSession.setCurState((short)(state+10)); // move to MS1 state
        else
            cSession.setCurState((short)(state+30)); // move to MS2 state
}

public boolean setNewLMS(short msI){
    if (msI==1)
        setLCfreshLMS(); //covers MC fresh (SMS <--50) & others SMS <--
        depends pre LC's last SMS
    else
        setLCotherLMS(msI); //SMS <-- depends pre LC's last SMS
    return true;
}

public void setLowMS(){
    short state = cSession.getCurState(); state -=30;
    cSession.setCurState((short)(state)); // move to low MS state
    MentalState ms= lc.getMS((short)((state==20)? 1:2));
    cSession.setCurMS(ms);
}

public void setLCfreshLMS(){ //msI=1 only; for MS1 only,

```

```

for all LCs
    MentalState ms=cSession.getCurMS();
    //leaSumm+= "\n The current Learning Stage is <<"+ ms.describe()+"
>>\n";
    //leaSumm+=" A new Estimate for Your Learning Ability (SMS)to be
initialized" +
    // " OR old SMS will be obtained";
    LearnerMS lastLMS=cSession.getCurLMS();
    cSession.setPreLMS(null,0);cSession.setPreLMS(null,1); // preLMSs
are all null
    //SMS will be initialized to 50 (or, if visited before may vary)
    LearnerMS lms;
    if (lclState())
        lms= LearnerMS.findORcreateLMS(aLearner,ms);
    else
        lms= LearnerMS.findORcreateFreshLMS(aLearner,ms,lastLMS);
    leaSumm+=lms.summ();
    leaSumm+=lms.getSMS().fuzz(); //old-keep;new-??;depend-equate (only
preceding)
    cSession.setCurLMS(lms);
}

public void setLCotherLMS(short msI){ //msI>1
    MentalState ms=cSession.getCurMS();
    LearnerMS lastLMS=cSession.getCurLMS();
    //leaSumm+= "\n The current Learning Stage is << "+ ms.describe()+
" >> \n";
    //leaSumm+=" A new Estimate for Your Learning Ability (SMS for the
new MS) to be initialized"+
    // "OR old SMS will be obtained";
    cSession.setPreLMS(lastLMS,msI-2); //always msI>1
    /*if (msI>3){//need to do ctsly
        if (cSession.getPreLMS(msI-1).getSMS().get() <
            lastLMS.getSMS().get()){
            cSession.getPreLMS(msI-1).getSMS().set(lastLMS.getSMS().
                get());
        }
        if (cSession.getPreLMS(msI-2).getSMS().get() <
            lastLMS.getSMS().get()){
            cSession.getPreLMS(msI-2).getSMS().set(lastLMS.getSMS().
                get());
        }
    }
    }else if (msI>2) {//need to do ctsly
        if (cSession.getPreLMS(msI-2).getSMS().get() <
            lastLMS.getSMS().get()){
            cSession.getPreLMS(msI-2).getSMS().set(lastLMS.
                getSMS().get());
        }
    }
    */
    LearnerMS lms=
LearnerMS.findORcreateLMS(aLearner,ms,lastLMS,cSession.getCurScuffL())
;
    leaSumm+=lms.summ();
    leaSumm+=lms.getSMS().fuzz(); //old-keep;new-??;depend-equate (only
preceding)
    cSession.setCurLMS(lms);
}

public boolean doMCQ(int s){//score s changes, initially 80, later
90
    boolean doo =true;
    if (cSession.getCurLMS().getSMS().get()>s){ //it may ask when
Lesson is presented, but learner may revise

```

```

        String s1="\n\n You performed well in this lesson. Do you still
want to try more MCQs? \n";
        leaSumm+= "=> As you already got good score, LOZ suggests that
you might not need to do MCQ at"
        +" this level, and you were shown this message \n" +s;
        int n=JOptionPane.showConfirmDialog(mLCtrl.mLUI(), s1, "Good
Work "+aLearner.userID(),
                                JOptionPane.YES_NO_OPTION);
        if (n==JOptionPane.NO_OPTION){
            cSession.setCurAction((short)15); // first MCQ skipped
            leaSumm+= "\n\n=> You decided not to do MCQs at this level";
            doo=false;
        }
    }
    return doo;
}

public void presentMCQ(int i){
    cSession.setCurActSTime(System.currentTimeMillis());
    MentalState ms = cSession.getCurMS();
    LearnerMS lms=cSession.getCurLMS();
    cSession.setCurMCQ(ms.mcq(i));
    lms.setLmcqNO((short)i); // set MCQ number
    cSession.setCurAction((short)10); // MCQ is being presented
    LearnerMCQ lmq=
LearnerMCQ.findORcreateLMCQ(aLearner,ms.mcq(i)); //PAS initialized
    cSession.setCurLMCQ(lmq);
    int index=0;
    /*if (lmq.getPAS()!=null) index=lmq.getPAS().index();
if (index>0){
    String msg= lmq.getPAS().fuzz()+" \n You did this MCQ already.
\n Do you want to do it again?";
    int n=JOptionPane.showConfirmDialog(mLCtrl.mLUI(),msg , "You
Did IT!",
JOptionPane.OK_CANCEL_OPTION);
    if (n==JOptionPane.CANCEL_OPTION){
        leaSumm=leaSumm+ " ==> You were informed that you did this MCQ
already## \n\n" +
            " \n\n## and you opted to skip it \n";
        cSession.setCurAction((short)15); // precondition skipped
        next(); //should select next lesson
    }
} else{
*/
    MCQ mcq=ms.mcq(i);
    StrengthMS sms=lms.getSMS();
    predictPerf(mcq,sms);
    mLCtrl.disableNext();
    mLCtrl.mcq1(mcq,ms.describe()); //show MCQ
    //simulProcessAnswer();
}

// A procedure for simulation
private void simulProcessAnswer(){
    MCQ mcq =cSession.getCurMCQ();
    LearnerMS lms=cSession.getCurLMS();
    LearnerMCQ lmq=cSession.getCurLMCQ();
    short state = cSession.getCurState();
    cSession.setCurActSTime(System.currentTimeMillis());
    short sA= ans[globI++];

```



```

private void predictPerf(MCQ mcq, StrengthMS sms){
    leaSumm+="\n^^^^^^^^^PERFORMANCE PREDICTION^^^^^^^^^^^^\n";
    //leaSumm+="\n==> You will be given a question of difficulty
level " + Integer.toString(mcq.level())+"\n";
    StrengthPerF sPerf3= new StrengthPerF(mcq.level(),sms,3);
    StrengthPerF sPerf2= new StrengthPerF(mcq.level(),sms,2);
    //leaSumm+="==> How will you perform? \n";
    leaSumm+= "\n LOZ's prediction (3 levels)
is:"+sPerf3.describe3();
    leaSumm+= "\n LOZ's prediction (2 levels)
is:"+sPerf2.describe2();
}

private boolean lastMSofMC(){
    short last= (short)(cSession.getCurMC().getMaxLC()*100+100-
20);//assumes 3 MS
    return ( cSession.getCurState()==last);
}

private boolean lastMSofLC(){
    return (((cSession.getCurState()+20)%100) == 0);
}

public void fromMS(){
    //short time=getFBTime();
    LearnerMS lms=cSession.getCurLMS();
    /*Check- repeating in fbOKpressed?
if (time>threshHold) // due to feed back learning

lms.getSMS().predictSMS(cSession.getCurLMC().getLLA(),time,threshHold)
; //IMPORTANT prediction
    leaSumm=leaSumm+lms.getSMS().pred();
    */
    cSession.setFirstAttempt(true);
    short action= cSession.getCurAction();
    cSession.setCurAction((short)20); // MCQ being presented
    int currSeq=0;
    if (action==15) //if previous MCQ skipped
        currSeq=3;
    else
        currSeq= cSession.getCurLMC().getPAS().currSeq(); //get the
next level, low-1/same-2/next-3
    leaSumm=leaSumm+ "\n The index for the next MCQ's difficulty
level is "+Integer.toString(currSeq)+
        " (low-1/same-2/next-3) \n";
    if (currSeq==3) // goto next MS
        if (lastMSofMC()){// in LC2-MS3, no next MS, no next LC, go to
explo topic
            dynUpdateLLA();
            presentExplore();
        } else if (lastMSofLC())// LC1,MS3 state;80/180/except last80;
no next ms , go to next lc
            presentLC2();
        else
            processMSnextCQ3(); // not last MS, goto next MS
    else { //currSeq==2 & 1
        int i=lms.getLmcqNO()+1; // to get the next mcq in same
level
        /*
        if (i<cSession.getCurMS().maxQ()+1){

```

```

        //lms.setLmcqNO((short)i);
        presentMCQ(i);
    } else {
        leaSumm += " Same MS level, but low difficulty level MCQ is
not available; "+
                "trying next MS level MCQ";
        processMSnextCQ3();
        /*if (currSeq==1) //move to low level MS
processMSnextCQ1();
*/
        int msID = lms.getMS().msID();
        if ((msID==200)|| (msID==210)|| (msID==220))
            if (i==2)
                presentMCQ(2);
            else //i>2, i=1 will not come here
                if (lastMSofLC())// msID=220
                    presentLC2();
                else
                    processMSnextCQ3(); //msID=200,210
            else //i>1; no second MCQs
                if (lastMSofLC())// msID=224,228,232
                    presentLC2();
                else
                    processMSnextCQ3();
//msID=204,214,208,218,212,222
        }
    }

    public void presentLC2(){
        short state = cSession.getCurState();
        cSession.setCurActSTime(System.currentTimeMillis());
        cSession.setCurLCSTime(System.currentTimeMillis());
        lc=mc.getLC((short)((state+20)/100)); // state 80 or 180
        cSession.setCurAction((short)10); //LC1 is being presented
        cSession.setCurState((short)(30+state)); // a main lesson,
MC-LC1 state
        cSession.setCurScuffL((short)1);
        cSession.setCurLC(lc); // diable below 2 lines if simul
        mLCtrl.enableNext();
        mLCtrl.presentLC(lc);
        //next(); //simul
    }

    public void processMSnextCQ3(){// move to next MS; last MS will not
come here
        short msI=1;
        int cState =cSession.getCurState();
        if ((cState+80)%100==0) msI=2;
        else if ((cState+50)%100==0) msI=3; //(last MS will not come)
        setNextMS((short)msI); // MS-1 --> MS-2 ; includes new LMS (so
new SMS)
        //if (!doMCQ(89))
        //next();
        // else {cSession.getCurLMS().setLmcqNO((short)1);
        //if (!doMCQ(89))
        //next();
        presentMCQ(1);
    }

    public void processMSnextCQ1(){// move to LOW MS
        short cAction = cSession.getCurAction();

```

```

        short cState = cSession.getCurState();
        if (!(cState+80)%100==0){ // in MS1, no low MS, remain in same
ms, don't go to low L
            setLowMS(); //state will be updated
            //setNewLMS(); //ms will be updated
            cSession.getCurLMS().setLmcqNO((short)1);
            presentMCQ(1);
        }else {
            LearnerMS lms=cSession.getCurLMS();
            leaSumm += " Lower level Mental State not available (in
MS1); "+
                    "trying same MS level, but low difficulty level
MCQ";

            int i=lms.getLmcqNO()+1;
            if (i<cSession.getCurMS().maxQ()+1){
                cSession.getCurLMS().setLmcqNO((short)i);
                //if (!doMCQ(89))
                //next();
                presentMCQ(lms.getLmcqNO()+1);
            } else {
                leaSumm += " Same MS level, but low difficulty level
MCQ also not available; "+
                    "trying next MS level MCQ";
                processMSnextCQ3();
            }
        }
    }

private boolean mcqON(){
    boolean m=mLCtrl.mcqDlog().isVisible();
    if (m)
        JOptionPane.showMessageDialog(mLCtrl.mLUI(), "Close MCQ Dialog
box",
            "CAUTION: LOZ may not help you further, "
+aLearner.userID(), JOptionPane.OK_OPTION);
    return m;
}

public void dynUpdateLLA(){
    long fT= System.currentTimeMillis(); //start time in mSec
    Date ftD=new Date(fT);
    Time ftT=new Time(fT);
    leaSumm=leaSumm+" ==> You finished this Lesson on "+
ftD.toString().substring(1,10)+
        " at "+ ftT.toString()+ "\n";
    int t= threshHold+1;// (int) ((fT-cSession.getCurLCSTime())/6000);
//in minutes
    leaSumm=leaSumm+ "=="> You spent " + Integer.toString(t) + "
minutes in this lesson \n" ;
    short time=0;
    if (t<threshHold)
        leaSumm+="You had spent only little time on previous lesson !\n
"+
            "LOZ assume only reasonable learning activity, \n, Please
advise otherwise <use LearnerModel> ";
    else {
        time= 90;//(short) (t*100/mc.getMinTime()); //time scaled 0-
100
        leaSumm+=" \n Local Learning Ability - LLA is to be predicted;
LLA0ld --> LLAnew";
    }
}

```



```

    Time ftT=new Time(fT);
    int t= (int) ((fT-cSession.getCurActSTime())/60000); //in minutes
    leaSumm=leaSumm+" \n ==> You finished this sub-session lesson on "+
ftD.toString().substring(0,10)+
        " at "+ ftT.toString()+"\n";

    leaSumm=leaSumm+ " ==> You spent " + Integer.toString(t) + " minutes
in this lesson \n" +
        " ==> The estimated maximum learning time for beginners for
this lesson is "+lc.getMaxTime()+
        " minutes \n";
    if (t>=lc.getMaxTime()){
        String msg="\n\nYou had spent much time on previous lesson \n "+
            "LOZ assume maximum learning activity (that is 100% in
percentage scale). \n "+
                " Please advise otherwise <use
LearnerModel> ";
        JOptionPane.showMessageDialog(mLCtrl.mLUI(), msg, "Long Time?",
JOptionPane.OK_OPTION);
        leaSumm=leaSumm+ " ==>\n***** You were shown the following
message \n *****" +msg + "\n";
        t=lc.getMinTime(); // this is 100% time, minTime is just an
average time
    }
    short time=0;
    if (t<threshHold){ //must be < minTime
        leaSumm=leaSumm+ "\n==> The estimated minimum learning time for
experts for this lesson is "+
            threshHold+ " minutes \n";
        String msg="\nYou had spent only little time on previous lesson
\n "+
            "LOZ assume reasonable learning activity, \n, Please advise
otherwise <use LearnerModel> ";
        JOptionPane.showMessageDialog(mLCtrl.mLUI(), msg, "Short Time?",
JOptionPane.OK_OPTION);
    }
    else
        time= (short) (t*100/mc.getMinTime()); //time scaled 0-100
    return time;
    */
}

private short getFBTime(){
    return (short) 90;
    /*
    long fT= System.currentTimeMillis(); //start time in mSec
    Date ftD=new Date(fT);
    Time ftT=new Time(fT);
    leaSumm=leaSumm+"\n\n ==> You finished this feed back lesson on "+
ftD.toString().substring(0,10)+
        "at "+ ftT.toString()+"\n";
    int t= (int) ((fT-cSession.getCurActSTime())/60000); //in minutes
    leaSumm=leaSumm+ "\n==> You spent " + Integer.toString(t) + "
minutes in this FB \n" ;
    if (t>= maxFBtime) {
        String msg="\n\nYou had spent much time on previous lesson !\n "+
            "LOZ assume maximum learning activity (that is 100% in
percentage scale). \n "+
                " Please advise otherwise <use
LearnerModel> ";

```

```

        JOptionPane.showMessageDialog(mLCtrl.mLUI(), msg, "Long Time?",
JOptionPane.OK_OPTION);
        leaSumm=leaSumm+ "=>\n***** You were shown the following
message \n *****" +msg + "\n";
        t=minFBtime; // this is 100% time, minTime is just an average
time
    }
    short time=0;
    if (t<threshHold){
        leaSumm=leaSumm+ "\n\n=> The estimated minimum learning time for
experts for this FB is "+
            threshHold+ " minutes \n";
    }
    else
        time = (short) (t*100/minFBtime); //time scaled 0-100
    return time;
*/
}
}
}

```

Class MisConceptDB

```
package lozpt1;
```

```

import com.borland.dx.dataset.*;
import com.borland.dx.sql.dataset.*;
import java.sql.*;

public class MisConceptDB extends LozDB{
    private static MisConceptDB mcDB = new MisConceptDB();
    public MisConceptDB() {
    }
    public static MisConceptDB mcDB(){
        return mcDB;
    }
    public MisConception findaMisConcept(short mcID){
        short type=2;
        MisConception mc = (MisConception)Cache.find(new
Short(mcID),type);
        if (mc==null){
            mc=findINdb(mcID);
            Cache.put(new Short(mcID), mc,type);
        }
        return mc;
    }

    public MisConception findINdb(short mcID){
        QueryDataSet getLeaQ = new QueryDataSet();
        QueryDescriptor qD = new QueryDescriptor(LOZdB, "SELECT
        \"MisConcept\".* "+ "FROM \"MisConcept\" WHERE
        \"MisConcept\".\"misConID\" = " + mcID ,
            null, false, Load.ALL);
        getLeaQ.setQuery(qD);
        getLeaQ.executeQuery();
        LOZdB.closeConnection();
        if (getLeaQ.isEmpty())
            return null;
        MisConception mc= new MisConception(mcID,getLeaQ.getShort(2));
    }
}

```

```

        getLeaQ.closeStatement();
        return mc;
    }
}

```

Class MisConception

```

package lozpt1;

public class MisConception {
    private short misConID;
    private BasicConcept concept;
    public MisConception(short mcID, short bcID) {
        misConID=mcID;
        concept = BasicConceptDB.bcDB().findaBasicConcept(bcID);
    }
}

```

Class NewLearnerCtrl

```

package lozpt1;

import javax.swing.JOptionPane;

public class NewLearnerCtrl {
    private NewLearnerUI nlUI;
    public NewLearnerCtrl() {
        nlUI=new NewLearnerUI(this);
        nlUI.setSize(500,700);
        nlUI.pack();
        Util.U.setCentre(nlUI);
        nlUI.setVisible(true);
    }

    public void OKpressed(String un, String pw, short m, short oo, short
    uml, short z){
        nlUI.dispose();
        z = (short)(z*10+5); m= (short)(m*10+5); oo= (short)(oo*10+5);
        uml= (short)(uml*10+5);
        short nLid=Learner.getNextLeaID();
        short la= (short)((m*5+oo*6+uml*7+z*8)/26);
        String leaSumm = " ==> You have selected the following scales for
        your knowledge levels in OO concepts,"+
            "UML Notation , Mathematics, and Z Notation" +
            Integer.toString(oo)+ " " +Integer.toString(uml) + " "+
            Integer.toString(m)+ " "+Integer.toString(z) + "\n"+ " ==>
        Your initial Global Learn Ability (GLA) is " +
            Integer.toString(la);
        Learner newL =new Learner(nLid,un,pw,la,(z>5),z,m,oo,uml);
        //create lp,lor also
        //LearnerDB.lDB().putALearner(newL); //write lp, lor also
        new Mentor(newL,leaSumm);
    }
    public void closePressed(){
        int n= JOptionPane.showConfirmDialog(nlUI, "Leaving LOZ?, Good
        Bye!",

```

```

        "Closing LOZ", JOptionPane.YES_NO_OPTION);
    if (n== JOptionPane.YES_OPTION)
        nlUI.dispose();
    }
}

```

Class NewLearnerUI

```

package lozpt1;

import java.awt.*;
import javax.swing.*;
import com.borland.jbcl.layout.*;
import java.awt.event.*;

public class NewLearnerUI extends JFrame {
    private NewLearnerCtrl nlCtrl=null;
    private BorderLayout borderLayout1 = new BorderLayout();
    private JPanel jPanel1 = new JPanel();
    private JPanel jPanel2 = new JPanel();
    private JPanel jPanel3 = new JPanel();
    private JLabel jLabel1 = new JLabel();
    private XYLayout xYLayout2 = new XYLayout();
    private JLabel jLabel2 = new JLabel();
    private JTextField unText = new JTextField();
    private JLabel jLabel3 = new JLabel();
    private JLabel jLabel5 = new JLabel();
    private JLabel jLabel6 = new JLabel();
    private XYLayout xYLayout3 = new XYLayout();
    private JLabel jLabel7 = new JLabel();
    private JLabel jLabel8 = new JLabel();
    private JTextField dmText = new JTextField();
    private JLabel jLabel9 = new JLabel();
    private JLabel jLabel11 = new JLabel();
    private JTextField ooText = new JTextField();
    private JTextField umText = new JTextField();
    private JTextField znText = new JTextField();
    private JButton vButton1 = new JButton();
    private JButton vButton2 = new JButton();
    private JButton vButton3 = new JButton();
    private JButton vButton4 = new JButton();
    private JButton OK = new JButton();
    private JButton cancel = new JButton();
    private FlowLayout flowLayout1 = new FlowLayout();
    private JPasswordField jPW = new JPasswordField();

    public NewLearnerUI(NewLearnerCtrl nc) {
        try {
            jbInit();
        }
        catch(Exception e) {
            e.printStackTrace();
        }
        nlCtrl=nc;
    }

    private void jbInit() throws Exception {
        this.getContentPane().setLayout(borderLayout1);
        this.setTitle("New Learner - Enter Detail");
        jPanel1.setLayout(flowLayout1);
    }
}

```

```

jLabel1.setFont(new java.awt.Font("Serif", 1, 18));
jLabel1.setText("Give a USER NAME");
jPanel3.setLayout(xYLayout2);
jPanel3.setBackground(new Color(236, 158, 216));
jPanel3.setBorder(BorderFactory.createRaisedBevelBorder());
jLabel2.setFont(new java.awt.Font("Serif", 1, 18));
jLabel2.setText("Give a PASS WORD");
unText.setFont(new java.awt.Font("SansSerif", 0, 24));
jPanel2.setBackground(new Color(236, 210, 216));
jPanel2.setFont(new java.awt.Font("Dialog", 0, 16));
jPanel2.setBorder(BorderFactory.createEtchedBorder());
jPanel2.setLayout(xYLayout3);
jLabel3.setFont(new java.awt.Font("Serif", 1, 18));
jLabel3.setForeground(new Color(255, 72, 0));
jLabel3.setText("Please Give an Estimation for Your Knowledge");
jLabel5.setText("Use Scales 0-10, from Nothing (0) to
Excellence(10)");
jLabel5.setForeground(new Color(255, 0, 236));
jLabel5.setFont(new java.awt.Font("Serif", 1, 18));
jLabel6.setText("Discrete Mathematics :");
jLabel6.setForeground(new Color(98, 164, 85));
jLabel6.setFont(new java.awt.Font("Serif", 1, 18));
jLabel7.setFont(new java.awt.Font("Serif", 1, 18));
jLabel7.setForeground(new Color(98, 164, 85));
jLabel7.setText("Object Orientation :");
jLabel8.setFont(new java.awt.Font("Serif", 1, 18));
jLabel8.setForeground(new Color(98, 164, 85));
jLabel8.setText("UML Notation :");
dmText.setFont(new java.awt.Font("Dialog", 0, 16));
jLabel9.setFont(new java.awt.Font("Serif", 1, 18));
jLabel9.setForeground(new Color(98, 164, 85));
jLabel9.setText("Z Notation");
jLabel11.setFont(new java.awt.Font("Serif", 1, 20));
jLabel11.setForeground(new Color(98, 26, 85));
jLabel11.setText("Register Yourself!");
ooText.setFont(new java.awt.Font("Dialog", 0, 16));
umText.setFont(new java.awt.Font("Dialog", 0, 16));
znText.setFont(new java.awt.Font("Dialog", 0, 16));
vButton1.setBackground(Color.orange);
vButton1.setEnabled(false);
vButton1.setFont(new java.awt.Font("Dialog", 1, 14));
vButton1.setMnemonic('0');
vButton1.setText("doTest");
vButton2.setEnabled(false);
vButton2.setText("doTest");
vButton2.setMnemonic('0');
vButton2.setFont(new java.awt.Font("Dialog", 1, 14));
vButton2.setBackground(Color.orange);
vButton3.setEnabled(false);
vButton3.setText("doTest");
vButton3.setMnemonic('0');
vButton3.setFont(new java.awt.Font("Dialog", 1, 14));
vButton3.setBackground(Color.orange);
vButton4.setEnabled(false);
vButton4.setText("doTest");
vButton4.setMnemonic('0');
vButton4.setFont(new java.awt.Font("Dialog", 1, 14));
vButton4.setBackground(Color.orange);
OK.setBackground(new Color(236, 164, 216));
OK.setText("OK");
OK.addActionListener(new java.awt.event.ActionListener() {

```

```

        public void actionPerformed(ActionEvent e) {
            OK_actionPerformed(e);
        }
    });
cancel.setBackground(new Color(236, 164, 216));
cancel.setText("Cancel");
cancel.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(ActionEvent e) {
        cancel_actionPerformed(e);
    }
});
jPanel1.setBackground(Color.pink);
jPanel1.setBorder(BorderFactory.createRaisedBevelBorder());
jPW.setFont(new java.awt.Font("Dialog", 0, 16));
this.getContentPane().add(jPanel2, BorderLayout.CENTER);
this.getContentPane().add(jPanel3, BorderLayout.NORTH);
jPanel3.add(unText, new XYConstraints(257, 41, 193, 33));
jPanel3.add(jLabel11, new XYConstraints(197, 0, -1, 37));
jPanel3.add(jPW, new XYConstraints(257, 83, 193, 33));
jPanel3.add(jLabel2, new XYConstraints(44, 87, 242, 37));
jPanel3.add(jLabel1, new XYConstraints(43, 45, 243, 35));
this.getContentPane().add(jPanel1, BorderLayout.SOUTH);
jPanel2.add(dmText, new XYConstraints(260, 79, 37, 40));
jPanel2.add(jLabel6, new XYConstraints(51, 85, -1, -1));
jPanel2.add(ooText, new XYConstraints(260, 130, 37, 40));
jPanel2.add(umText, new XYConstraints(260, 180, 37, 40));
jPanel2.add(znText, new XYConstraints(260, 231, 37, 40));
jPanel2.add(jLabel7, new XYConstraints(51, 137, -1, -1));
jPanel2.add(jLabel9, new XYConstraints(51, 242, -1, -1));
jPanel2.add(jLabel8, new XYConstraints(51, 190, -1, -1));
jPanel2.add(vButton1, new XYConstraints(337, 85, 101, 26));
jPanel2.add(vButton2, new XYConstraints(337, 137, 101, 26));
jPanel2.add(vButton4, new XYConstraints(337, 240, 101, 26));
jPanel2.add(vButton3, new XYConstraints(337, 188, 101, 26));
jPanel2.add(jLabel3, new XYConstraints(96, 25, -1, -1));
jPanel2.add(jLabel5, new XYConstraints(89, 286, -1, -1));
jPanel1.add(cancel, null);
jPanel1.add(OK, null);
}

void cancel_actionPerformed(ActionEvent e) {
    nlCtrl.closePressed();
}

void OK_actionPerformed(ActionEvent e) {
    String message="";
    String un=unText.getText();
    if (un.length()== 0)
        message=message + " \n select a User Name!";
    String pw= new String(jPW.getPassword());
    if (pw.length()== 0)
        message=message + " \n select a Pass Word!"; //
    String dm=dmText.getText();    short dmI=0;
    if (dm.length()==0)
        message=message + " \n select a Range (0-9) for Mathematics
Knowledge!";
    else
        dmI= (short) Integer.parseInt(dm.trim());
    String oo=ooText.getText();    short ooI=0;
    if (oo.length()==0)
        message=message + " \n select a Range (0-9) for OO Knowledge!";
}

```



```

else
    ooI= (short)Integer.parseInt(oo.trim());
String um=umText.getText();    short umI=0;
if (um.length()==0)
    message=message + " \n select a Range (0-9) for UML Knowledge!";
else
    umI= (short)Integer.parseInt(um.trim());
String zn=znText.getText();    short znI=0;
if (zn.length()==0)
    message=message + " \n select a Range (0-9) for Z Knowledge!";
else
    znI= (short)Integer.parseInt(zn.trim());
if (message.length()==0)
    nlCtrl.OKpressed(un,pw,dmI,ooI,umI,znI);
else
    JOptionPane.showMessageDialog(null, message, "Not
Completed!",JOptionPane.ERROR_MESSAGE);
}
}

```

Class Notation

```

package lozpt1;
import javax.swing.*;
import java.lang.Object;

public class Notation {
    private String notaText;
    private ImageIcon notaImg;
    public Notation(String nt, String nft){
        notaText = nt;
        String path = "c:\\mohan\\lozpt1\\src\\";
        notaImg=new ImageIcon(path+nft);
    }
    public String getNotaText(){
        return notaText;
    }
    public ImageIcon getNotaImg(){
        return notaImg;
    }
}
package lozpt1;

import java.util.*;

public class Session {
    private static short lastSubSessID=0;
    private static final short maxHis=3;
    private short sessID;
    private short leaID;
    private long startTime;
    private long duration; // in minutes; session kept only if any
subsession exist
    private String summary;// accumulated history in string- a detailed
summary
    private ArrayList subSessIDs=new ArrayList(10); //some action may
not have subsession
    private short lastSessID;

```

```

    public Session(short sesID,short leID, long sT, long d, short lsID,
String S) {
        sessID=sesID; leaID=leID; startTime=sT; duration=d; summary =S;
lastSessID=lsID;
    }
    public ArrayList getSubSessIDs(){
        return subSessIDs;
    }
    public long startTime(){
        return startTime;
    }
    public long duration(){
        return duration;
    }
    public String summary(){
        return summary;
    }
    public Session getLSession(){
        return SessionDB.ssDB().findaSession(lastSessID);
    }
    public static short nextSubSessID(){
        return ++lastSubSessID;
    }
    public static void setLastSubSessID(short s){
        lastSubSessID=s;
    }
}

```

Class SessionBC

```

package lozpt1;

public class SessionBC extends SubSession{
    private short bcID;
    public SessionBC(short sID, short ssID, short ty,long sT, long d,
short td, short a, short s,
        String summ, short bID){
        super(sID,ssID,ty,sT,d,td,a,s,summ); bcID=bID;
    }
}

```

Class SessionDB

```

package lozpt1;

import com.borland.dx.dataset.*;
import com.borland.dx.sql.dataset.*;
import java.sql.*;

public class SessionDB extends LozDB{
    private static SessionDB ssDB = new SessionDB();

    public static SessionDB ssDB(){
        return ssDB;
    }
    public Session findaSession(short ssID){ //must also check leaID
        QueryDataSet getLeaQ = new QueryDataSet();
    }
}

```

```

        QueryDescriptor qD = new QueryDescriptor(LOZdB, "SELECT
\"Session\".*"+
        " FROM\"Session\" WHERE \"Session\".\"ssID\"=" + ssID,
        null, false, Load.ALL);
        getLeaQ.setQuery(qD);
        getLeaQ.executeQuery();
        LOZdB.closeConnection();
        if (getLeaQ.isEmpty())
            return null;
        Session lr=new Session(ssID, getLeaQ.getShort(2),
getLeaQ.getLong(3),
        getLeaQ.getLong(4), getLeaQ.getShort(5),
getLeaQ.getString(6));
        getLeaQ.closeStatement();
        return lr;
    }
}

```

Class SessionMCQ

```

package lozpt1;

public class SessionMCQ extends SubSession{
    short mcqID;
    short pasIndex=0;
    public SessionMCQ(short sID, short ssID, short ty, long sT, long d,
long td,short a,
        short s, String summ, short mID) {
        super(sID,ssID,ty,sT,d,td,a,s,summ); mcqID = mID;
    }
}

```

Class StrengthMS

```

package lozpt1;

public class StrengthMS {
    short sms = 50; //initial
    FuzzyMem3L smsFZ= new FuzzyMem3L(new short[]{50,50,0});
    private static float[] affirm= {3,4,7}; //low to high
    private static float[] denies=
    {(float)0.13,(float)0.25,(float)0.33};//low to high
    private String fuzz="";
    private String deFuzz="";
    private String upD="";
    private String pred="";
    public StrengthMS() {
    }
    public StrengthMS(short m) {
        sms=m;
        smsFZ=FuzzyMem3L.fuzzify(m);
        fuzz=fuzz+ "\n@@ The current % scale for SMS is "
        +Integer.toString(sms) + "\n";
        /*
        fuzz=fuzz+ "\n@@ The resultant values for SMS are \n"+
        "=> The fuzzy value, SMS- STRONG = " +
        Integer.toString(smsFZ.level((short)2)) + "\n"+
        "=> The fuzzy value, SMS- MEDIUM = " +

```

```

Integer.toString(smsFZ.level((short)1)) + "\n"+
    "==> The fuzzy value, SMS- LOW = " +
        Integer.toString(smsFZ.level((short)0)) + "\n";
    */
}
public short get(){
    return sms;
}
public FuzzyMem3L getFZ(){
    return smsFZ;
}
public void set(short m){
    sms=m;
    smsFZ=FuzzyMem3L.fuzzify(m);
}

private FuzzyMem3L predictLLAXTtoL(LocalLA la, short T){
    FuzzyMem3L timeFZ=FuzzyMem3L.fuzzify(T);
    pred=pred+ "\n\n@@ The current % scale for Time is "
        +Integer.toString(T) + "\n"+
        "==> The fuzzy value, time- MUCH = " +
        Integer.toString(timeFZ.level((short)2)) + "\n"+
        "==> The fuzzy value, time- REASONABLE = " +
        Integer.toString(timeFZ.level((short)1)) + "\n"+
        "==> The fuzzy value, time- A LITTLE = " +
        Integer.toString(timeFZ.level((short)0)) + "\n";
    pred=pred+ "\n@@ The current % scale for LLA is "
        +Integer.toString(la.get()) + "\n"+
        "==> The fuzzy value, LLA- HIGH = " +
        Integer.toString(la.getFZ().level((short)2)) + "\n"+
        "==> The fuzzy value, LLA- MEDIUM = " +
        Integer.toString(la.getFZ().level((short)1)) + "\n"+
        "==> The fuzzy value, LLA- LOW = " +
        Integer.toString(la.getFZ().level((short)0)) + "\n";
    pred=pred+"\n !!!!! Rules in LLA*Time ==> L is applied - see help
        in LM for detail !!!!!\n";
    return FuzzyMX3and3to3.applyRuleL(la.getFZ(),timeFZ);
}

public void predictSMS (LocalLA la, short T, short threshold){
    FuzzyMem3L L;
    /*
    Disable time related predictions; just transfer LLA to L

    if (T >threshold)
        L=predictLLAXTtoL(la,T); // LLA X T ==> L
    else {
        L= FuzzyMem3L.fuzzify(la.get()); // L <-- LLA
        pred=pred+"\n !!Spent Time on Lesson < Threshold : --> L is
just initialized by LLA !!!!!\n";
    }
    */

    //But, Dynamic prediction also disabled; Both LLA*T ==> L & SMS*L
    ==> SMS disabled
    //pred=pred+"\n Time related calculation disabled: L is just
    initialized by LLA\n";
    L = la.getFZ(); // just transfer LLA to L
    /*simul
    pred=pred+ "\n@@ The resultant values for L are \n"+

```

```

        "=> The fuzzy value, L- THOROUGHLY = " +
Integer.toString(L.level((short)2)) + "\n"+
        "=> The fuzzy value, L- MODERATELY = " + I
neger.toString(L.level((short)1)) + "\n"+
        "=> The fuzzy value, L- SCARECELY = " +
Integer.toString(L.level((short)0)) + "\n";
pred=pred+ "\n@@ The current % scale for SMS is "
+Integer.toString(sms) + "\n"+
        "=> The fuzzy value, SMS- STRONG = " +
Integer.toString(smsFZ.level((short)2)) + "\n"+
        "=> The fuzzy value, SMS- MEDIUM = " +
Integer.toString(smsFZ.level((short)1)) + "\n"+
        "=>The fuzzy value, SMS- WEAK = " +
Integer.toString(smsFZ.level((short)0)) + "\n";
*/
    smsFZ=FuzzyMX3and3to3.applyRuleSMS(smsFZ,L); // Dynamic
Prediction SMS*L ==> SMS
pred="\n Dynamic Fuzzy Rules in SMSold * L ==> SMSnew is applied -
see help for detail \n";//simul
/*
pred=pred+ "\n@@ The resultant FUZZY values for SMS are \n"+
        "=> The fuzzy value, SMS- STRONG = " +
Integer.toString(smsFZ.level((short)2)) + "\n"+
        "=> The fuzzy value, SMS- MEDIUM = " +
Integer.toString(smsFZ.level((short)1)) + "\n"+
        "=> The fuzzy value, SMS- WEAK = " +
Integer.toString(smsFZ.level((short)0)) + "\n";
*/
    sms=smsFZ.crisp();
pred=pred+ "\n@@ The resultant (% scale) for SMS now is "
+Integer.toString(sms) + "\n";
}

public void updateSMS(short d,boolean c){
    float ohNOe= ((float)sms/(100-sms));
    upD= "\n The current (% scale) for SMS is " +
Integer.toString(sms) + "\n";
    if (c){
        upD+= " You answered correctly in an MCQ of difficulty level
"+ Integer.toString(d) + "====> This AFFIRMS your SMS by " +
Float.toString(affirm[d-1]);
        ohNOe *= affirm[d-1];
    }else {
        upD+= " You answered incorrectly in an MCQ of difficulty level
"+Integer.toString(d)+ "====> This DENIES your SMS by " +
Float.toString(denies[d-1]);
        ohNOe *= denies[d-1];
    }
    sms=(short)((ohNOe*100)/(1+ohNOe));
    smsFZ=FuzzyMem3L.fuzzify(sms);
    upD=upD+ " \n The resultant (% scale) for System's Estimate
about Your Learning Ability- SMS is " + Integer.toString(sms) +
"\n";
}

public String fuzz(){
    return fuzz;
}
public String deFuzz(){
    return deFuzz;
}
}

```

```

public String upD(){
    return upD;
}
public String pred(){
    return pred;
}
}

```

Class StrengthPA

```

package lozpt1;

public class StrengthPA {
    private short pas=0; //normal defuzz
    private short pasIndex=0; //defuzz using max, with 10 added for
correct;1-5;11-15
    private FuzzyMem5L pasFZ=null;
    private String fuzz="";
    private String pred="";
    static private String expl="";
    private short currSeq=0;

    public StrengthPA() {
    }

    public StrengthPA(short pI) {
        pasIndex=pI;
        fuzz= " You already answered this MCQ - Your answer was ";
        String c= pI >10?"CORRECT":"WRONG";
        fuzz+= c+ "\n The Feed back level given to you was an Index
(called PAS)==>"+Integer.toString(pI);
    }

    public void setPasFZ(FuzzyMem5L pZ, boolean c) {
        pasFZ=pZ;
        pasIndex= (short) (pasFZ.defuzzMax()+1);
        if (c)
            pasIndex+=10;
    }

    public String fuzz(){
        return fuzz;
    }
    public FuzzyMem5L pasFZ(){
        return pasFZ;
    }
    public short index(){
        return pasIndex;
    }
    public short currSeq(){ //curriculum sequencing
        currSeq=3;
        switch (pasIndex) {
            case 5: currSeq=1; break;
            case 2:
            case 3:
            case 4:
            case 15: currSeq=2;break;
        }
        return currSeq;
    }
}

```

```

}

public static StrengthPA selectPAS(short mcqD, StrengthMS
sms,boolean c){
    FuzzyMem3L mcqFZ= FuzzyMem3L.fuzzify(mcqD);
    FuzzyMem3L smsFZ= sms.getFZ();
    FuzzyMem5L pasFZ= FuzzyMXpas.applyRule(smsFZ,mcqFZ,c);
    StrengthPA pas= new StrengthPA();
    pas.pasFZ=pasFZ;
    pas.expl= " The system assumes that you are << " + smsFZ.des(1) +
        " >> in the current mental state \n";
    pas.expl+= " You have answered an MCQ, which is, in general, << "
        + mcqFZ.des(2) + " >> \n";
    pas.expl+= c ? " Your answer is << CORRECT >>" : "Unfortunately,
        your answer is << NOT CORRECT >>\n";
    pas.pasIndex= (short) (pasFZ.defuzzMax()+1);
    if (c)
        pas.pasIndex+=10;
    /*
    pas.pred += "\n The current % scale for System's estimate about
        your learning ability - SMS is " +Integer.toString(sms.get()) +
        "\n" +
        " ==> The fuzzy values are ;\n SMS- STRONG = " +
        Integer.toString(smsFZ.level((short)2)) +
        ": SMS- MEDIUM = " +
        Integer.toString(smsFZ.level((short)1)) +
        ": SMS- WEAK = " + Integer.toString(smsFZ.level((short)0))
    + "\n";
    pas.pred += "\n The % scale for MCQ Difficulty is "
    +Integer.toString(mcqD) + "\n" +
        " ==> The fuzzy values are ;\n D - HIGH = " +
        Integer.toString(mcqFZ.level((short)2)) +
        ": D - MODERATE = " +
        Integer.toString(mcqFZ.level((short)1)) +
        ": D - LOW = " +
        Integer.toString(mcqFZ.level((short)0)) + "\n";
    pas.pred += "\n The Resultant values for Required Level of Teaching
    Action (PAS) are \n" +
        " PAS- TOO HIGH = " + I
        nteger.toString(pasFZ.level((short)4)) +
        " PAS- HIGH = " + Integer.toString(pasFZ.level((short)3))
    +
        " PAS - MODERATE = " +
        Integer.toString(pasFZ.level((short)2)) + "\n" +
        " PAS - LOW = " + Integer.toString(pasFZ.level((short)1))
    +
        " PAS - TOO LOW = " +
        Integer.toString(pasFZ.level((short)0)) + "\n";
    */
    pas.pred = "\n\n The Resultant Required Level of Teaching Action
    - PAS INDEX is >>> " +
        Integer.toString(pas.index())+"\n\n"; //simul
    pas.expl+=pas.explain();
    pas.expl+= "\n\n^^^^^^^^^^^^^^^^^^^^ More Technical
    Explanation^^^^^^^^^^^^^^^^^^^^";
    pas.expl += pas.pred; // to get a detailed explanation
    return pas; // correct --> PAS10-PAS15 else PAS0-PAS5
}

public String pred(){
    return pred;
}

```

```

public String expl(){
    return expl;
}
public String sExp(){
    String s=expl;
    switch (pasIndex) {
        case 11 : s="\n^^ LOZ assumed that the question was very
easy for you, "+
            " and, therefore, you were simply given the"+
            "next level question to answer next."; break;
        case 12 : s="\n^^ LOZ assumed that the question was easy
for you, "+
            " and, therefore, you were just told why the "+
            "answer is correct and, given the next level question
to answer next.";break;
        case 13 : s="\n^^ LOZ assumed that the question was not
hard for you, "+
            " and, Therefore, you were just told why the answer was
correct "+
            " and, why the other answers were wrong , and given the
next level question to answer next.";break;
        case 14 : s="\n^^ LOZ assumed that the question was hard
for you, "+
            " but, you answered it correctly. Therefore, LOZ
thought, in case you might need to know why the answer was
correct "+
            " and, why the other answers were wrong , and some
detailed explanation on the topic "+
            " However, you were given the next level question to
answer next.";break;
        case 15 : s="\n^^ LOZ assumed that the question was very
hard for you, "+
            " but, you answered it correctly. Therefore, LOZ
thought, in case you might need to know why the answer was
correct "+
            " and, why the other answers were wrong , and some
detailed explanation on the topic "+
            " and to strengthen your knowledge, you were encouraged
to answer another same level question.";break;
        case 1 :s="\n^^ LOZ assumed that the question was easy for
you, "+
            " Therefore, LOZ assumed it was a careless mistake"+
            " and gave you one more chance to answer the same
question";break;
        case 2 :s="\n^^ LOZ assumed that the question was not hard
for you, "+
            " Therefore, LOZ assumed it was a careless mistake."+
            " However, LOZ told you why your answer was wrong "+
            " and gave you one more chance to answer the same
question";break;
        case 3 :s="\n^^ LOZ assumed that the question was not easy
for you, "+
            " but, still, you answered it wrongly. Therefore, LOZ told
you why your answer was wrong "+
            " and, why the other answers were wrong , and some
detailed explanation on the topic "+
            " and also to strengthen your knowledge, you were
encouraged to answer another same level question.";break;
        case 4 :s="\n^^ LOZ assumed that the question was hard for
you, "+

```



```

        " and, in fact, you answered it wrongly. Therefore, LOZ
assumed you need a detailed feed back. "+
        " Therefore, you were told why your answer was wrong "+
        " and, why the other answers were wrong , and some
detailed explanation on the topic "+
        " however, you were encouraged to answer another same
level question.";break;
        case 5 :s="\n^^ LOZ assumed that the question was very hard
for you, "+
        " and, in fact, you answered it wrongly. Therefore, LOZ
assumed you need a more detailed feed back. "+
        " Therefore, you were told why your answer was wrong "+
        " and, why the other answers were wrong , and some
detailed explanation on the topic "+
        " however, to strengthen your knowledge, you were
encouraged to try a lower level question now.";break;
    }
return s;
}

public String explain(){
String s=expl;
switch (pasIndex) {
    case 11 : s="\n^^ LOZ assumed that the question was
easy for you, "+
        " and, infact you answered it correctly. Therefore,
you were simply given the"+
        "next level question to answer next."; break;
    case 12 : s="\n^^ LOZ assumed that the question was not
hard for you, "+
        " and, infact you answered it correctly. Therefore,
you were just told why the "+
        "answer is correct and, given the next level question
to answer next.";break;
    case 13 : s="\n^^ LOZ assumed that the question was
neither hard nor easy for you, "+
        " and, infact you answered it correctly. Therefore,
you were just told why the answer was correct "+
        " and, why the other answers were wrong , and given
the next level question to answer next.";break;
    case 14 : s="\n^^ LOZ assumed that the question was not
easy for you, "+
        " but, you answered it correctly. Therefore, LOZ
thought, in case you might need to know why the answer was
correct "+
        " and, why the other answers were wrong , and some
detailed explanation on the topic "+
        " However, you were given the next level question to
answer next.";break;
    case 15 : s="\n^^ LOZ assumed that the question was
hard for you, "+
        " but, you answered it correctly. Therefore, LOZ
thought, in case you might need to know why the answer was
correct "+
        " and, why the other answers were wrong , and some
detailed explanation on the topic "+
        " and to strengthen your knowledge, you were
encouraged to answer another same level question.";break;
    case 1 :s="\n^^ LOZ assumed that the question was easy
for you, "+

```

```

        " but, unfortunately, you answered it wrongly.
Therefore, LOZ assumed it was a careless mistake"+
        " and gave you one more chance to answer the same
question";break;
        case 2 :s="\n^^ LOZ assumed that the question was not
hard for you, "+
        " but, unfortunately, you answered it wrongly.
Therefore, LOZ assume it was a careless mistake."+
        " However, LOZ told you why your answer was wrong "+
        " and gave you one more chance to answer the same
question";break;
        case 3 :s="\n^^ LOZ assumed that the question was
neither hard nor easy for you, "+
        " but, still, you answered it wrongly. Therefore, LOZ
told you why your answer was wrong "+
        " and, why the other answers were wrong , and some
detailed explanation on the topic "+
        " and also to strengthen your knowledge, you were
encouraged to answer another same level question.";break;
        case 4 :s="\n^^ LOZ assumed that the question was not
easy for you, "+
        " and, in fact, you answered it wrongly. Therefore,
LOZ assumed you need a detailed feed back. "+
        " Therefore, you were told why your answer was wrong
"+
        " and, why the other answers were wrong , and some
detailed explanation on the topic "+
        " however, you were encouraged to answer another same
level question.";break;
        case 5 :s="\n^^ LOZ assumed that the question was hard
for you, "+
        " and, in fact, you answered it wrongly. Therefore, LOZ
assumed you need a more detailed feed back. "+
        " Therefore, you were told why your answer was wrong "+
        " and, why the other answers were wrong , and some
detailed explanation on the topic "+
        " however, to strengthen your knowledge, you were
encouraged to try a lower level question now.";break;
    }
    return s;
}

}

```

Class StrengthPerF

```

package lozpt1;

public class StrengthPerF {
    private short perf=0; //normal defuzz
    private String[] des = {"Wrong","Medium","Correct"};
    private FuzzyMem3L perfFZ=null;

    public StrengthPerF(short mcqD, StrengthMS sms, int p) {
        FuzzyMem3L mcqFZ= FuzzyMem3L.fuzzify(mcqD);
        FuzzyMem3L smsFZ= sms.getFZ();
    }
}

```

```

        if (p==3)
            perfFZ= FuzzyMX3and3to3.applyRuleP3(smsFZ,mcqFZ);
        else
            perfFZ= FuzzyMX3and3to3.applyRuleP2(smsFZ,mcqFZ);
    }

    public FuzzyMem3L perfFZ(){
        return perfFZ;
    }

    public String describe3( ){
        String s= "\n"+des[0] +" ==>
"+Integer.toString(perfFZ.level((short)0))+"\n"+
        des[1]+ " ==>
"+Integer.toString(perfFZ.level((short)1))+"\n"+
        des[2]+ " ==>
"+Integer.toString(perfFZ.level((short)2))+"\n";
        s+= "Prediction for Performance is ==> "+
des[perfFZ.defuzzMax()]+\n";
        return s;
    }
    public String describe2( ){
        String s= "\n"+des[0] +" ==>
"+Integer.toString(perfFZ.level((short)0))+"\n"+
        des[2]+ " ==>
"+Integer.toString(perfFZ.level((short)2))+"\n";
        s+= "Prediction for Performance is ==> "+
des[perfFZ.defuzzMax()]+\n";
        return s;
    }
}

```

Class SubSession

```

package lozpt1;

public { // not inherited from Session, but composed leaID?
    private short subSesID;
    private short sessID;
    private short type; //1-preL,2-LC,3-MC(LC),4-
    private long startTime;
    private long duration; //if threshold >duration, subsession is kept
    private long totalDuration;
    private short action=0;
    private short state=0;
    private String summary;

    public SubSession(short sID, short ssID, short ty, long sT,long d,
        long td,short a, short s, String summ) {
        subSesID=ssID;sessID=ssID;type=ty;startTime=sT;duration=d;totalD
uration=td;action=a;state=s;
        summary=summ;
    }

    public long getDuration(){
        return duration;
    }
}

```

```

public void setDuration(long d){
    duration=d;
}

public long getStTime(short t){
    return startTime;
}

}

```

Class SummaryCtrl

```

package lozpt1;

import java.util.*;
import javax.swing.*;
import java.awt.*;

public class SummaryCtrl {
    private boolean packFrame = false;
    private SummaryUI summaryUI;
    private Mentor theMentor;

    public SummaryCtrl(Mentor M, boolean so) {
        try {
            summaryUI = new SummaryUI(this,true);
            if (packFrame) {
                summaryUI.pack();
            }
            else {
                summaryUI.validate();
            }
            Util.U.setCentre(summaryUI);
            summaryUI.setVisible(true);
            theMentor= M;
        }
        catch(Exception e) {
            e.printStackTrace();
        }
    }

    public SummaryUI summaryUI(){
        return summaryUI;
    }

    public void OKpressed(boolean summON, boolean optON ){
        if (summON)
            theMentor.changeLPsumm(summON,optON);
        summaryUI.dispose();
    }

    public void detailPressed( ){
        theMentor.showDetailSummary(this);
    }

    public void inspectLM( ){
        theMentor.inspectLM();
    }

    public void setSizeUI(int x, int y){
        summaryUI.setSize(new Dimension(x,y));
        summaryUI.validate();
        Util.U.setCentre(summaryUI);
    }
}

```

```

}
public void setWelcome(String w1){
    summaryUI.presentWelcome(w1);
}
public void setSummary(String d, String t, String s){
    summaryUI.presentSummary(d,t,s);
}
public void setSummary(String s){
    summaryUI.presentSummary(s);
}
}
}

```

Class SummaryUI

```

package lozpt1;

import javax.swing.*;
import java.awt.*;
import com.borland.jbcl.layout.*;
import java.awt.event.*;

public extends JFrame{
    private SummaryCtrl summCtrl;
    private JPanel contentPane;
    private JPanel jPanel2 = new JPanel();
    private JScrollPane jScrollPane1 = new JScrollPane();
    private JPanel jPanel3 = new JPanel();
    private JTextPane jTextPanel = new JTextPane();
    private BorderLayout borderLayout1 = new BorderLayout();
    private JRadioButton jRadioButtonNoSumm = new JRadioButton();
    private JRadioButton jRadioButtonNoOpt = new JRadioButton();
    private ButtonGroup bg1= new ButtonGroup();
    private JPanel jPanel4 = new JPanel();
    private JButton jButton4 = new JButton();
    private JButton jButton2 = new JButton();
    private JButton OK = new JButton();
    private JButton jButton1 = new JButton();
    private XYLayout xYLayout1 = new XYLayout();
    private JLabel jLabel1 = new JLabel();
    private XYLayout xYLayout2 = new XYLayout();
    private JButton detail = new JButton();
    private JButton jButton5 = new JButton();
    private XYLayout xYLayout3 = new XYLayout();
    private JButton jButton6 = new JButton();
    private JToolBar jToolBar1 = new JToolBar();
    private JLabel jLabel2 = new JLabel();
    private JLabel jLabel3 = new JLabel();

    public SummaryUI(SummaryCtrl sCl, boolean so) {
        try {
            jbInit();
            if (so) {
                // jRadioButtonNoOpt.disable();
                //jRadioButtonNoSumm.disable();
                //jPanel2.disable();
            }
        }
    }
}

```

```

        catch(Exception e) {
            e.printStackTrace();
        }
        summCtrl=sCl;
    }
    public SummaryUI( ) {
        try {
            jbInit();
        }
        catch(Exception e) {
            e.printStackTrace();
        }
        this.setTitle("LOZ- LEARNER INTERACTION SUMMARY");
        this.setSize(new Dimension(800,800));
        this.pack();
        this.validate();
        Util.U.setCentre(this);
        this.setVisible(true);
    }
    private void jbInit() throws Exception {
        contentPane = (JPanel) this.getContentPane();
        setTitle("LOZ- LAST SESSION SUMMARY");
        this.setSize(new Dimension(800,500));
        contentPane.setLayout(borderLayout1);
        jPanel3.setLayout(xYLayout2);
        jButtonNoSumm.setText("DO NOT show summary in future");
        jButtonNoSumm.addActionListener(new java.awt.event.
            ActionListener() {
                public void actionPerformed(ActionEvent e) {
                    jButtonNoSumm_actionPerformed(e);
                }
            });
        jButtonNoOpt.setText("DO NOT show this option in future");
        jPanel4.setLayout(xYLayout3);
        jPanel4.setBorder(BorderFactory.createLoweredBevelBorder());
        jButton4.setText("GOTO..");
        jButton2.setToolTipText("");
        jButton2.setText("<<");
        jButton2.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(ActionEvent e) {
                jButton2_actionPerformed(e);
            }
        });
        OK.setText("OK");
        OK.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(ActionEvent e) {
                OK_actionPerformed(e);
            }
        });
        jButton1.setEnabled(false);
        jButton1.setToolTipText("");
        jButton1.setText(">>");
        jButton1.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(ActionEvent e) {
                jButton1_actionPerformed(e);
            }
        });
        jPanel2.setLayout(xYLayout1);
        detail.setText("Show Detail");
        detail.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(ActionEvent e) {

```

```

        detail_actionPerformed(e);
    }
});
jButton5.setText("Inspect LM");
jButton5.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(ActionEvent e) {
        jButton5_actionPerformed(e);
    }
});
jButton6.setText("Help");
jLabel1.setFont(new java.awt.Font("SansSerif", 1, 24));
jLabel1.setForeground(new Color(177, 72, 79));
jLabel2.setFont(new java.awt.Font("Serif", 2, 14));
jLabel2.setForeground(Color.red);
jLabel3.setFont(new java.awt.Font("Dialog", 1, 11));
jLabel3.setForeground(Color.pink);
contentPane.add(jScrollPane, BorderLayout.CENTER);
jScrollPane.getViewPort().add(jTextPanel, null);
jPanel4.add(detail, new XYConstraints(4, 5, -1, -1));
jPanel4.add(jButton5, new XYConstraints(110, 4, -1, -1));
jPanel4.add(OK, new XYConstraints(714, 2, -1, -1));
jPanel4.add(jButton6, new XYConstraints(572, 1, -1, -1));
jPanel4.add(jToolBar1, new XYConstraints(267, 1, -1, 30));
jToolBar1.add(jButton2, null);
jToolBar1.add(jButton4, null);
jToolBar1.add(jButton1, null);
jPanel2.add(jRadioButtonNoOpt,new XYConstraints(543, 52, -1, -1));
jPanel2.add(jRadioButtonNoSumm, new XYConstraints(40, 49,-1, -1));
jPanel2.add(jPanel4, new XYConstraints(10, 8, 775, -1));
contentPane.add(jPanel3, BorderLayout.NORTH);
jPanel3.add(jLabel1, new XYConstraints(29, 2, 757, 42));
//jPanel3.add(jLabel2, new XYConstraints(244, 2, 124, 42));
//jPanel3.add(jLabel3, new XYConstraints(385, 2, 411, 42));
contentPane.add(jPanel2, BorderLayout.SOUTH);
bg1.add(jRadioButtonNoSumm);bg1.add(jRadioButtonNoOpt);

}
void jRadioButtonNoSumm_actionPerformed(ActionEvent e) {

}
void jButton2_actionPerformed(ActionEvent e) {

}
void OK_actionPerformed(ActionEvent e) { summCtrl.OKpressed
    (jRadioButtonNoSumm.isEnabled(),jRadioButtonNoOpt.isEnabled());
}
void jButton1_actionPerformed(ActionEvent e) {
}
public void presentWelcome(String w1){
    jLabel1.setText(w1);//    jLabel1.setText(w2);
jLabel1.setText(w3);
}
public void presentSummary(String d,String t,String s){
    AttributeSet.aSet.setStyle(1,1);
    jTextPanel.setText("");
    Util.U.appendToTextPane(jTextPanel,"A DETAILED SUMMARY
        \n",AttributeSet.aSet.heading2);
    Util.U.appendToTextPane(jTextPanel,d+"\n",AttributeSet.aSet.heading3);
    Util.U.appendToTextPane(jTextPanel,t+"\n",AttributeSet.aSet.heading3);
    Util.U.appendToTextPane(jTextPanel,s+"\n",AttributeSet.aSet.paral);
}

```

```

        Util.U.appendToTextPane(jTextPanel," ==> You may inspect your
Learner Model; press <Inspect LM> \n",
                                AttriSet.aSet.para2);
    }
    public void presentSummary(String s){
        AttriSet.aSet.setStyle(1,1);
        Util.U.appendToTextPane(jTextPanel,"A SHORT SUMMARY
\n",AttriSet.aSet.heading2);
        Util.U.appendToTextPane(jTextPanel,s+"\n",AttriSet.aSet.para1);
        Util.U.appendToTextPane(jTextPanel," ==> You may get detail
summary; press <Detail> \n",
                                AttriSet.aSet.para2);
        Util.U.appendToTextPane(jTextPanel," ==> You may inspect your
Learner Model; press <Inspect LM> \n",
                                AttriSet.aSet.para2);

    }
    public void presentLearnerSummary(String s){
        AttriSet.aSet.setStyle(1,1);
        Util.U.appendToTextPane(jTextPanel,"Learner's Interaction
Summary \n",AttriSet.aSet.heading1);
        Util.U.appendToTextPane(jTextPanel,s+"\n",AttriSet.aSet.para1);
        Util.U.appendToTextPane(jTextPanel," ==> You may change your
Learner Model; press <Inspect LM> \n",
                                AttriSet.aSet.para2);
    }

    void detail_actionPerformed(ActionEvent e) {
        summCtrl.detailPressed();
        detail.disable();
    }

    void jButton5_actionPerformed(ActionEvent e) {
        summCtrl.inspectLM();
    }
}

```

Class summaUI

```

package lozpt1;

import java.awt.*;
import javax.swing.*;

public class summaUI extends JFrame {
    private JPanel jPanel1 = new JPanel();
    private JPanel jPanel2 = new JPanel();
    private JScrollPane jScrollPane1 = new JScrollPane();
    private JTextPane jTextPanel = new JTextPane();

    public summaUI() {
        try {
            jbInit();
        }
        catch(Exception e) {
            e.printStackTrace();
        }
    }
    private void jbInit() throws Exception {

```



```

        jTextPanel.setText("jTextPanel");
        this.getContentPane().add(jPanell1, BorderLayout.SOUTH);
        this.getContentPane().add(jPanel2, BorderLayout.NORTH);
        this.getContentPane().add(jScrollPane1, BorderLayout.CENTER);
        jScrollPane1.getViewport().add(jTextPanel, null);
    }
}
package lozpt1;

import javax.swing.*;
import java.awt.*;
import com.borland.jbcl.layout.*;
import java.awt.event.*;

public class summUI {
    private JPanel jPanell1 = new JPanel();
    private JPanel jPanel2 = new JPanel();
    private JScrollPane jScrollPane1 = new JScrollPane();
    private JPanel jPanel3 = new JPanel();
    private JTextPane jTextPanel = new JTextPane();
    private BorderLayout BorderLayout1 = new BorderLayout();
    private JRadioButton jRadioButtonNoSumm = new JRadioButton();
    private JRadioButton jRadioButtonNoOpt = new JRadioButton();
    private JPanel jPanel4 = new JPanel();
    private JButton jButton4 = new JButton();
    private JButton jButton2 = new JButton();
    private JButton OK = new JButton();
    private JButton jButton1 = new JButton();
    private XYLayout xYLayout3 = new XYLayout();
    private XYLayout xYLayout1 = new XYLayout();
    private JLabel jLabel1 = new JLabel();
    private XYLayout xYLayout2 = new XYLayout();

    public summUI() {
        try {
            jbInit();
        }
        catch(Exception e) {
            e.printStackTrace();
        }
    }
    private void jbInit() throws Exception {
        jPanell1.setLayout(BorderLayout1);
        jPanel3.setLayout(xYLayout2);
        jTextPanel.setText("jTextPanel");
        jRadioButtonNoSumm.setText("DO NOT show summary in future");
        jRadioButtonNoSumm.addActionListener(new
java.awt.event.ActionListener() {
            public void actionPerformed(ActionEvent e) {
                jRadioButtonNoSumm_actionPerformed(e);
            }
        });
        jRadioButtonNoOpt.setText("DO NOT show this option in future");
        jPanel4.setLayout(xYLayout3);
        jPanel4.setBorder(BorderFactory.createLoweredBevelBorder());
        jButton4.setText("GOTO..");
        jButton2.setToolTipText("");
        jButton2.setText("<<");
        jButton2.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(ActionEvent e) {

```

```

        jButton2_actionPerformed(e);
    }
});
OK.setText("OK");
OK.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(ActionEvent e) {
        OK_actionPerformed(e);
    }
});
jButton1.setEnabled(false);
jButton1.setToolTipText("");
jButton1.setText(">>");
jButton1.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(ActionEvent e) {
        jButton1_actionPerformed(e);
    }
});
jPanel2.setLayout(xYLayout1);
jLabel1.setText("jLabel1");
jPanel1.add(jScrollPane, BorderLayout.CENTER);
jScrollPane.getViewport().add(jTextPanel, null);
jPanel1.add(jPanel2, BorderLayout.SOUTH);
jPanel4.add(jButton2, new XYConstraints(7, 0, 69, -1));
jPanel4.add(jButton4, new XYConstraints(80, 1, -1, -1));
jPanel4.add(jButton1, new XYConstraints(159, 1, 61, -1));
jPanel4.add(OK, new XYConstraints(307, 0, 58, -1));
jPanel2.add(jRadioButtonNoSumm,new XYConstraints(97, 45, -1, -1));
jPanel2.add(jRadioButtonNoOpt new XYConstraints(94, 75, -1, -1));
jPanel2.add(jPanel4, new XYConstraints(10, 8, 381, -1));
jPanel1.add(jPanel3, BorderLayout.NORTH);
jPanel3.add(jLabel1, new XYConstraints(3, 5, 389, 35));
}
void jRadioButtonNoSumm_actionPerformed(ActionEvent e) {

}
void jButton2_actionPerformed(ActionEvent e) {

}
void OK_actionPerformed(ActionEvent e) {

}
void jButton1_actionPerformed(ActionEvent e) {

}
}
}

```

Class Topic

```

package lozpt1;

public class Topic {
    private short topicID;
    private String title;
    private String description;
    private static short maxMC=3;
    private short mCs[]; //MainConcept[]
    public Topic(short tID, String t, String d, short mcs1, short mcs2,
short mcs3) { //need short mCs[]
        topicID=tID; title=t;description=d;
    }
}

```

```

        mCs[0] =mcs1; mCs[1]=mcs2; mCs[2]=mcs3;
        //for (short i=0; i<maxMC; i++ )
            //mCs[i] = MainConceptDB.mcDB().findaMainConcept(mcs[i]);
    }
    public static short getMaxMC(){
        return maxMC;
    }
    public MainConcept nextMC(short id){
        for (short i=0; i<maxMC; i++ )
            if (mCs[i] == id)
                if (i<maxMC-1)
                    return MainConceptDB.mcDB().findaMainConcept(mCs[i+1]);
        return null;
    }
}

```

Class TopicDB

```

package lozpt1;

import com.borland.dx.dataset.*;
import com.borland.dx.sql.dataset.*;
import java.sql.*;

public class TopicDB extends LozDB{
    private static TopicDB toDB = new TopicDB();
    public TopicDB() {
    }
    public static TopicDB toDB(){
        return toDB;
    }
    public Topic findaTopic(short toID){
        QueryDataSet getTopQ = new QueryDataSet();
        QueryDescriptor qD = new QueryDescriptor(LOZdB, "SELECT
\"Topics\".*"+
        " FROM\"Topics\" WHERE \"Topics\".\"tID\"=" + toID
        , null, false, Load.ALL);
        getTopQ.setQuery(qD);
        getTopQ.executeQuery();
        LOZdB.closeConnection();
        if (getTopQ.isEmpty())
            return null;
        //short[] mCs;
        //for (short i=0; i<Topic.getMaxMC(); i++)
            //mCs[i]= getTopQ.getShort(1+i);
        Topic topic=new
Topic(toID,getTopQ.getString(2),getTopQ.getString(3),
        getTopQ.getShort(5),getTopQ.getShort(6),
getTopQ.getShort(7));
        getTopQ.closeStatement();
        return topic;
    }
}

```

class Util

```
package lozpt1;

import java.awt.*;
import javax.swing.*;
import com.borland.jbcl.layout.*;
import java.awt.event.*;
import javax.swing.text.*;
import javax.swing.border.*;
import java.util.*;
import java.applet.*;
import java.net.URL;

public class Util {
    public static Util U =new Util();
    public Util() {

        public static void appendToTextPane(JTextComponent TP, String s,
SimpleAttributeSet attributes) {
            Document d = TP.getDocument( );
            ArrayList a= splitLines(s);
            for (short i=0; i< a.size(); i++){
                String line= a.get(i).toString();
                if (i<a.size()-1) line+="\n"; //avoid last return
                try { d.insertString(d.getLength( ), line, attributes); }
                    catch (BadLocationException ble) {}
            }
        }

        public static void appendImgToTextPane(JTextComponent TP,
SimpleAttributeSet attributes) {
            Document d = TP.getDocument( );
            try { d.insertString(d.getLength( ), " ",
attributes); }
                catch (BadLocationException ble) {}
            }

        public static void setCentre(Window JF){
            Dimension screenSize =
Toolkit.getDefaultToolkit().getScreenSize();
            Dimension frameSize = JF.getSize();
            if (frameSize.height > screenSize.height) {
                frameSize.height = screenSize.height;
            }
            if (frameSize.width > screenSize.width) {
                frameSize.width = screenSize.width;
            }
            JF.setLocation((screenSize.width - frameSize.width) / 2,
                (screenSize.height - frameSize.height) / 2);
        }

        public static void setRBot(Window JF){
            Dimension screenSize =
Toolkit.getDefaultToolkit().getScreenSize();
            Dimension frameSize = JF.getSize();
            if (frameSize.height > screenSize.height) {
                frameSize.height = screenSize.height;
            }
        }
    }
}
```

```

        if (frameSize.width > screenSize.width) {
            frameSize.width = screenSize.width;
        }
        JF.setLocation((screenSize.width - frameSize.width-50),
            (screenSize.height - frameSize.height-50));
    }

    public static void setRTop(Window JF){
        Dimension screenSize =
Toolkit.getDefaultToolkit().getScreenSize();
        Dimension frameSize = JF.getSize();
        if (frameSize.height > screenSize.height) {
            frameSize.height = screenSize.height;
        }
        if (frameSize.width > screenSize.width) {
            frameSize.width = screenSize.width;
        }
        JF.setLocation((screenSize.width - frameSize.width-20),
            (screenSize.height-20));
    }

    public static int min (short x, short y){
        return (x<y)? x:y;
    }

    public static int max (short x, short y){
        return (x>y)? x:y;
    }

    public static ArrayList splitLines(String s){
        ArrayList a = new ArrayList();
        int i= s.indexOf("\n");
        while (i>0) { // if newLine exist
            a.add(s.substring(0,i));
            s=s.substring(i+2,s.length());
            i= s.indexOf("\n");
        }
        a.add(s); //last one may be "\n"
        return a;
    }

    public static AudioClip setClip(String file){
        String u="file:///"+LozGlobal.lozG().path();
        u+=file;
        AudioClip clip=null;
        try {
            URL url=new URL(u);
            clip= Applet.newAudioClip(url);
        }
        catch(Exception e) {
            e.printStackTrace();
        }
        return clip;
    }
}

```

Class WebView

```
package lozpt1;
```

```

import java.awt.*;
import java.net.*;
import javax.swing.*;
import javax.swing.event.*;

public class WebView extends FileView {

    public WebView(String urlString) {
        super(urlString);
        setTitle (" Discussion Forum - Web View");
        jEditorPane.addHyperlinkListener(new LinkActivator( ));
    }

    class LinkActivator implements HyperlinkListener {
        public void hyperlinkUpdate(HyperlinkEvent he) {
            HyperlinkEvent.EventType type = he.getEventType( );
            if (type == HyperlinkEvent.EventType.ENTERED)
                jEditorPane.setCursor(
                    Cursor.getPredefinedCursor(Cursor.HAND_CURSOR));
            else if (type == HyperlinkEvent.EventType.EXITED)
                jEditorPane.setCursor(Cursor.getDefaultCursor( ));
            else if (type == HyperlinkEvent.EventType.ACTIVATED)
                openURL(he.getURL().toExternalForm( ));
        }
    }
}

```

```

package lozpt1;

import java.awt.*;
import javax.swing.*;
import com.borland.jbcl.layout.*;
import java.awt.event.*;

public class WelcomeDemo extends JFrame {
    private JPanel jPanel1 = new JPanel();
    private JLabel jLabel1 = new JLabel();
    private XYLayout xYLayout1 = new XYLayout();
    private JLabel jLabel2 = new JLabel();
    private JLabel jLabel3 = new JLabel();
    private JLabel jLabel4 = new JLabel();
    private JLabel jLabel5 = new JLabel();
    private JLabel jLabel6 = new JLabel();
    private JLabel jLabel7 = new JLabel();
    private JButton jButton1 = new JButton();
    private JButton jButton2 = new JButton();
    private JButton jButton3 = new JButton();
    private JButton jButton4 = new JButton();
    private JButton jButton5 = new JButton();
    private JButton jButton6 = new JButton();
    private BorderLayout borderLayout1 = new BorderLayout();

    public WelcomeDemo() {
        try {
            jbInit();
        }
    }
}

```

```

        catch(Exception e) {
            e.printStackTrace();
        }
    }
}
private void jbInit() throws Exception {
    jPanel1.setBackground(Color.pink);
    jPanel1.setBorder(BorderFactory.createLoweredBevelBorder());
    jPanel1.setLayout(xYLayout1);
    jLabel1.setBackground(Color.pink);
    jLabel1.setFont(new java.awt.Font("SansSerif", 1, 30));
    jLabel1.setForeground(new Color(171, 0, 0));
    jLabel1.setHorizontalAlignment(SwingConstants.CENTER);
    jLabel1.setText("Welcome to Demo page");
    jLabel2.setText("go through the FULL DEMO...");
    jLabel2.setHorizontalAlignment(SwingConstants.CENTER);
    jLabel2.setForeground(Color.red);
    jLabel2.setFont(new java.awt.Font("SansSerif", 1, 20));
    jLabel2.setBackground(Color.pink);
    jLabel3.setBackground(Color.pink);
    jLabel3.setFont(new java.awt.Font("SansSerif", 1, 20));
    jLabel3.setForeground(new Color(255, 0, 197));
    jLabel3.setHorizontalAlignment(SwingConstants.CENTER);
    jLabel3.setText("We strongly recommend you to");
    jLabel4.setBackground(Color.pink);
    jLabel4.setFont(new java.awt.Font("SansSerif", 1, 20));
    jLabel4.setForeground(new Color(255, 98, 0));
    jLabel4.setHorizontalAlignment(SwingConstants.LEFT);
    jLabel4.setText("It has three parts !");
    jLabel5.setBackground(Color.pink);
    jLabel5.setFont(new java.awt.Font("SansSerif", 1, 20));
    jLabel5.setForeground(Color.red);
    jLabel5.setHorizontalAlignment(SwingConstants.LEFT);
    jLabel5.setText("* Learning issues in LOZ...");
    jLabel6.setBackground(Color.pink);
    jLabel6.setFont(new java.awt.Font("SansSerif", 1, 20));
    jLabel6.setForeground(Color.red);
    jLabel6.setHorizontalAlignment(SwingConstants.LEFT);
    jLabel6.setText("* Working with NoteBook...");
    jLabel7.setText("* Harnessing Learner Model...");
    jLabel7.setHorizontalAlignment(SwingConstants.LEFT);
    jLabel7.setForeground(Color.red);
    jLabel7.setFont(new java.awt.Font("SansSerif", 1, 20));
    jLabel7.setBackground(Color.pink);
    jButton1.setBackground(new Color(236, 164, 216));
    jButton1.setFont(new java.awt.Font("DialogInput", 1, 18));
    jButton1.setForeground(new Color(0, 118, 0));
    jButton1.setBorder(BorderFactory.createRaisedBevelBorder());
    jButton1.setText("GO");
    jButton2.setBackground(new Color(236, 164, 216));
    jButton2.setFont(new java.awt.Font("DialogInput", 1, 18));
    jButton2.setForeground(new Color(0, 118, 0));
    jButton2.setBorder(BorderFactory.createRaisedBevelBorder());
    jButton2.setText("GO");
    jButton3.setBackground(new Color(236, 164, 216));
    jButton3.setFont(new java.awt.Font("DialogInput", 1, 18));
    jButton3.setForeground(new Color(0, 118, 0));
    jButton3.setBorder(BorderFactory.createRaisedBevelBorder());
    jButton3.setText("GO");
    jButton4.setBackground(new Color(236, 164, 216));
    jButton4.setFont(new java.awt.Font("DialogInput", 1, 18));
    jButton4.setForeground(new Color(0, 118, 0));
}

```

```

jButton4.setBorder(BorderFactory.createRaisedBevelBorder());
jButton4.setText("GO");
jButton5.setBackground(new Color(255, 197, 175));
jButton5.setFont(new java.awt.Font("Dialog", 0, 18));
jButton5.setForeground(Color.red);
jButton5.setText("GOTO main contents...");
jButton5.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(ActionEvent e) {
        jButton5_actionPerformed(e);
    }
});
jButton6.setText("CLOSE");
jButton6.setForeground(Color.red);
jButton6.setFont(new java.awt.Font("Dialog", 0, 18));
jButton6.setBackground(new Color(255, 197, 175));
this.getContentPane().setLayout(borderLayout1);
this.setTitle("Welcome to Demo!");
jPanell1.add(jLabel3, new XYConstraints(27, 73, 312, 56));
jPanell1.add(jLabel5, new XYConstraints(103, 216, 269, 56));
jPanell1.add(jLabel2, new XYConstraints(120, 115, 294, 56));
jPanell1.add(jLabel7, new XYConstraints(103, 336, 305, 58));
jPanell1.add(jLabel1, new XYConstraints(23, 10, 489, 56));
jPanell1.add(jLabel6, new XYConstraints(103, 276, 280, 56));
jPanell1.add(jButton2, new XYConstraints(453, 128, -1, -1));
jPanell1.add(jButton1, new XYConstraints(452, 236, -1, -1));
jPanell1.add(jButton3, new XYConstraints(452, 292, -1, -1));
jPanell1.add(jButton4, new XYConstraints(452, 347, -1, -1));
jPanell1.add(jButton5, new XYConstraints(272, 418, 217, 32));
jPanell1.add(jButton6, new XYConstraints(67, 418, 196, 32));
jPanell1.add(jLabel4, new XYConstraints(39, 163, 303, 56));
this.getContentPane().add(jPanell1, BorderLayout.CENTER);
}

void jButton5_actionPerformed(ActionEvent e) {
    JFrame MC = new MainContent();
    MC.pack();
    MC.setVisible(true);
    this.hide();
}
}
package lozpt1;

import javax.swing.JOptionPane;

public class WelcomeDemoCtrl {
    private WelcomeDemoUI wcUI;
    private Mentor M;
    public WelcomeDemoCtrl(Mentor M) {
        this.M=M;
        wcUI=new WelcomeDemoUI(this);
        wcUI.setSize(585,585);
        wcUI.validate();
        Util.U.setCentre(wcUI);
        wcUI.setVisible(true);
    }
    public void closePressed(){
        int n= JOptionPane.showConfirmDialog(wcUI, "Leaving LOZ, Good
Bye!",
                                "Closing LOZ", JOptionPane.YES_NO_OPTION);
        if (n== JOptionPane.YES_OPTION)
            wcUI.dispose();
    }
}

```



```

    }

    public void mcPressed(){
        wcUI.dispose();
        M.showMC();
    }
}

```

Class WelcomeDemoUI

```

package lozpt1;

import java.awt.*;
import javax.swing.*;
import com.borland.jbcl.layout.*;
import java.awt.event.*;
import javax.swing.border.*;

public class WelcomeDemoUI extends JFrame {
    private WelcomeDemoCtrl wdCtrl=null;
    private JPanel jPanel1 = new JPanel();
    private JButton mainContent = new JButton();
    private JButton close = new JButton();
    private BorderLayout borderLayout1 = new BorderLayout();
    private JPanel jPanel3 = new JPanel();
    private JLabel jLabel1 = new JLabel();
    private JPanel jPanel2 = new JPanel();
    private JPanel jPanel4 = new JPanel();
    private JPanel jPanel5 = new JPanel();
    private JLabel jLabel4 = new JLabel();
    private JButton jButton2 = new JButton();
    private JLabel jLabel3 = new JLabel();
    private JLabel jLabel2 = new JLabel();
    private XYLayout xYLayout2 = new XYLayout();
    private JButton jButton4 = new JButton();
    private JButton jButton3 = new JButton();
    private JLabel jLabel7 = new JLabel();
    private JLabel jLabel6 = new JLabel();
    private JButton jButton1 = new JButton();
    private JLabel jLabel5 = new JLabel();
    private XYLayout xYLayout3 = new XYLayout();
    private BorderLayout borderLayout2 = new BorderLayout();
    private Border border1;

    public WelcomeDemoUI(WelcomeDemoCtrl wCtrl) {
        try {
            jbInit();
        }
        catch(Exception e) {
            e.printStackTrace();
        }
        wdCtrl=wCtrl;
    }

    public WelcomeDemoUI() {
        try {
            jbInit();
        }
        catch(Exception e) {

```

```

        e.printStackTrace();
    }
    wdCtrl=null;
    mainContent.setEnabled(false);
}
private void jbInit() throws Exception {
    border1 =
BorderFactory.createBevelBorder(BevelBorder.LOWERED,Color.white,Color.
white,new Color(115, 89, 105),new Color(165, 128, 151));
    jPanell1.setBackground(Color.pink);
    jPanell1.setBorder(BorderFactory.createLoweredBevelBorder());
    jPanell1.setLayout(borderLayout2);
    mainContent.setBackground(new Color(255, 197, 175));
    mainContent.setFont(new java.awt.Font("Dialog", 0, 18));
    mainContent.setForeground(Color.red);
    mainContent.setText("GOTO main contents...");
    mainContent.addActionListener(new java.awt.event.ActionListener()
{
    public void actionPerformed(ActionEvent e) {
        mainContent_actionPerformed(e);
    }
});
close.setText("CLOSE");
close.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(ActionEvent e) {
        close_actionPerformed(e);
    }
});
close.setForeground(Color.red);
close.setFont(new java.awt.Font("Dialog", 0, 18));
close.setBackground(new Color(255, 197, 175));
this.getContentPane().setLayout(borderLayout1);
this.setTitle("Welcome to Demo!");
jLabel1.setBackground(Color.pink);
jLabel1.setFont(new java.awt.Font("SansSerif", 1, 30));
jLabel1.setForeground(new Color(171, 0, 0));
jLabel1.setHorizontalAlignment(SwingConstants.CENTER);
jLabel1.setText("Welcome to Demo page");
jPanel3.setBackground(new Color(236, 158, 216));
jPanel2.setBackground(new Color(236, 233, 171));
jPanel4.setBackground(new Color(236, 184, 216));
jPanel4.setLayout(xYLayout2);
jPanel5.setLayout(xYLayout3);
jPanel5.setBackground(new Color(236, 184, 216));
jLabel4.setBackground(Color.pink);
jLabel4.setFont(new java.awt.Font("SansSerif", 1, 20));
jLabel4.setForeground(new Color(85, 66, 105));
jLabel4.setHorizontalAlignment(SwingConstants.LEFT);
jLabel4.setText("It has three parts !");
jButton2.setBackground(new Color(236, 164, 216));
jButton2.setFont(new java.awt.Font("DialogInput", 1, 18));
jButton2.setForeground(new Color(0, 118, 0));
jButton2.setBorder(BorderFactory.createRaisedBevelBorder());
jButton2.setText("GO");
jButton2.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(ActionEvent e) {
        jButton2_actionPerformed(e);
    }
});
jLabel3.setBackground(Color.pink);
jLabel3.setFont(new java.awt.Font("SansSerif", 1, 20));

```

```

jLabel3.setForeground(new Color(255, 0, 197));
jLabel3.setHorizontalAlignment(SwingConstants.CENTER);
jLabel3.setText("We strongly recommend you to");
jLabel2.setText("go through the FULL DEMO...");
jLabel2.setHorizontalAlignment(SwingConstants.CENTER);
jLabel2.setForeground(Color.red);
jLabel2.setFont(new java.awt.Font("SansSerif", 1, 20));
jLabel2.setBackground(Color.pink);
jButton4.setBackground(new Color(236, 164, 216));
jButton4.setFont(new java.awt.Font("DialogInput", 1, 18));
jButton4.setForeground(new Color(0, 118, 0));
jButton4.setBorder(BorderFactory.createRaisedBevelBorder());
jButton4.setText("GO");
jButton3.setBackground(new Color(236, 164, 216));
jButton3.setFont(new java.awt.Font("DialogInput", 1, 18));
jButton3.setForeground(new Color(0, 118, 0));
jButton3.setBorder(BorderFactory.createRaisedBevelBorder());
jButton3.setText("GO");
jLabel7.setText("* Harnessing Learner Model...");
jLabel7.setHorizontalAlignment(SwingConstants.LEFT);
jLabel7.setForeground(Color.red);
jLabel7.setFont(new java.awt.Font("SansSerif", 1, 20));
jLabel7.setBackground(Color.pink);
jLabel6.setBackground(Color.pink);
jLabel6.setFont(new java.awt.Font("SansSerif", 1, 20));
jLabel6.setForeground(Color.red);
jLabel6.setHorizontalAlignment(SwingConstants.LEFT);
jLabel6.setText("* Working with NoteBook...");
jButton1.setBackground(new Color(236, 164, 216));
jButton1.setFont(new java.awt.Font("DialogInput", 1, 18));
jButton1.setForeground(new Color(0, 118, 0));
jButton1.setBorder(BorderFactory.createRaisedBevelBorder());
jButton1.setText("GO");
jLabel5.setBackground(Color.pink);
jLabel5.setFont(new java.awt.Font("SansSerif", 1, 20));
jLabel5.setForeground(Color.red);
jLabel5.setHorizontalAlignment(SwingConstants.LEFT);
jLabel5.setText("* Learning issues in LOZ...");
jPanel1.add(jPanel4, BorderLayout.NORTH);
jPanel4.add(jLabel3, new XYConstraints(31, 57, -1, -1));
jPanel4.add(jButton2, new XYConstraints(418, 86, 61, 31));
jPanel4.add(jLabel2, new XYConstraints(84, 51, -1, 122));
jPanel5.add(jLabel6, new XYConstraints(61, 174, 310, -1));
jPanel5.add(jLabel7, new XYConstraints(61, 119, 310, -1));
jPanel5.add(jLabel5, new XYConstraints(61, 63, 314, -1));
jPanel5.add(jLabel4, new XYConstraints(204, 5, -1, -1));
jPanel5.add(jButton1, new XYConstraints(417, 62, 58, 29));
jPanel5.add(jButton4, new XYConstraints(417, 117, 58, 29));
jPanel5.add(jButton3, new XYConstraints(417, 169, 58, -1));
jPanel1.add(jPanel5, BorderLayout.CENTER);
this.getContentPane().add(jPanel3, BorderLayout.NORTH);
jPanel3.add(jLabel1, null);
this.getContentPane().add(jPanel2, BorderLayout.SOUTH);
jPanel2.add(close, null);
jPanel2.add(mainContent, null);
this.getContentPane().add(jPanel1, BorderLayout.CENTER);
}

void mainContent_actionPerformed(ActionEvent e) {
    wdCtrl.mcPressed();
}

```

```
void close_actionPerformed(ActionEvent e) {
    if (wdCtrl==null)
        this.dispose();
    else
        wdCtrl.closePressed();
}
void jButton2_actionPerformed(ActionEvent e) {
}
}
```