

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.

Developing a Courseware Database for The AudioGraph

A Thesis presented in partial fulfilment of
the requirements for the degree of

Master
in
Computer Science

At Massey University, Palmerston North,
New Zealand.

Jun Pan

2000

Dedication

To memory of my auntie, Jin Yuan Pan

To my eldest sister, Feng Lan Pan

To my lovely daughter, Shu Ke

Acknowledgments

I would like to take this opportunity to thank the people who have helped to make my thesis a reality.

First, I would like to thank my supervisor, Chris Jesshope, for providing valuable guidance and suggestions along the way. I would also like to thank Elizabeth Kemp, who was very careful of checking my thesis and giving suggestions.

Second, I would like to give my thanks to Zhenzi Zhang, my colleague and my best friend, for her ability to endure my demands for improvement. Without her continued help, my thesis wouldn't be handed in on time. Thanks to my friend, Margaret Rollinson for checking my English grammar and for making this a smooth and understanding process. Thanks to my colleagues, Simon Zhang, Robin Luo and Yongqiu Liu. They gave me valuable feedback for my coding test.

Finally I give my thanks to my lovely daughter, Shu for her support and understanding during thesis writing. Giving up mum time has not been easy for her. Shu, keep working on your computer and someday you can go beyond your mum.

Contents

Abstract

List of Figures

Definitions

Chapter 1 Introduction.....	V
1.1 The Description of the Project	V
1.1.1 <i>Background of the project</i>	
1.1.2 <i>Purpose of the project</i>	
1.1.3 <i>Components of the project</i>	
1.2 An Overview of the Project's Phases	3
1.2.1 <i>Requirements specification and analysis</i>	
1.2.2 <i>Design schemata of the system</i>	
1.2.3 <i>Implementation of the system</i>	
1.3 Layout of the Thesis.....	11
1.4 Summary	12
Chapter 2 Background	13
2.1 AudioGraph System.....	14
2.1.1 <i>Principle for AudioGraph</i>	
2.1.2 <i>Studio for AudioGraph production</i>	
2.1.3 <i>AudioGraph presentation across the Internet</i>	
2.1.4 <i>A façade for developing AudioGraph application</i>	
2.2 Mini SQL Database System.....	15
2.2.1 <i>Database system and relational database model</i>	
2.2.2 <i>The mini SQL system</i>	
2.3 Applications in Java	18
2.3.1 <i>Features of Java</i>	
2.3.2 <i>Java is the best for developing Internet-based applications</i>	
2.4 JDBC and mSQL-JDBC Driver.....	20
2.4.1 <i>The structure of JDBC</i>	
2.4.2 <i>Using JDBC for our database access</i>	
2.4.3 <i>mSQL-JDBC Driver</i>	
2.5 Working Environment with the Project	25
2.5.1 <i>Java Developer's Kit (JDK) 1.1.8</i>	
2.5.2 <i>Collections</i>	
2.5.3 <i>Java Swing</i>	

2.5.4	<i>MRJ on Macintosh</i>	
2.6	Summary	26
Chapter 3	System Schema	27
3.1	Concepts of the ER and UML Modeling	27
3.1.1	<i>Entity-Relationship modeling</i>	
3.1.2	<i>OO methodology and modeling with UML</i>	
3.2	Requirements Analysis	30
3.2.1	<i>Database requirement analysis</i>	
3.2.2	<i>User-Interface application analysis</i>	
3.3	System Schema Design.....	33
3.3.1	<i>Database design</i>	
3.3.2	<i>User-Interface application design</i>	
3.4	Physical Schema Mapping.....	35
3.4.1	<i>Physical database mapping</i>	
3.4.2	<i>Physical application mapping</i>	
3.5	Summary	38
Chapter 4	Implementation	40
4.1	Methods Employed on the Server.....	40
4.1.1	<i>Installing mSQL 2.0 in the server system</i>	
4.1.2	<i>Using the standard programs and utilities</i>	
4.1.3	<i>Turning the Entity Relational Model into a relational Database</i>	
4.2	Methods Employed on the Client	44
4.2.1	<i>The Database package implementation</i>	
4.2.2	<i>The utility package implementation</i>	
4.2.3	<i>The interface package implementation</i>	
4.3	Tools Involved in the Project.....	50
4.3.1	<i>Create and save projects</i>	
4.3.2	<i>Add source files and library class</i>	
4.3.3	<i>Using target settings panel</i>	
4.3.4	<i>Java Virtual Machine</i>	
4.4	Problems Explored and Solutions.....	52
4.4.1	<i>Client/Server development</i>	
4.4.2	<i>Graphical User Interface components</i>	
4.4.3	<i>Runtime.exec() invoking</i>	
4.5	Summary	56
Chapter 5	Issues of Cross-platform Compatibility	57
5.1	Writing Multi-platform Java Code.....	57

5.1.1	<i>Architecture of Java's support for Platform Independence</i>	
5.2	Problem Evaluation.....	59
5.3	Solutions	60
5.4	The Politics of Platform Independence.....	62
5.4.1	<i>Java Foundation Classes</i>	
5.4.2	<i>Other tools to support Java</i>	
5.4.3	<i>Write Java code on the Macintosh first</i>	
5.5	Summary	64
Chapter 6	The Results of the System.....	67
6.1	Presentations of the System	67
6.1.1	<i>Login Authoring interface</i>	
6.1.2	<i>Courses and Staff Lists interface</i>	
6.1.3	<i>Courses Content interface</i>	
6.1.4	<i>A Lecture Note Presentation interface</i>	
6.2	Java Performance Issues	74
6.2.1	<i>Runtime Performance</i>	
6.2.2	<i>Program Performance</i>	
6.3	Work Still Required	78
6.3.1	<i>FTP technique and FTP in Java</i>	
6.3.2	<i>Password Encryption</i>	
6.3.3	<i>Search by relevant keywords</i>	
6.4	Summary	80
Chapter 7	Conclusion	81
7.1	Outline of the System.....	81
7.2	Achievement of the System	81
7.3	Issues of the System.....	82
7.4	Trends and Future of the System	83
Bibliography	84
Appendices	88
Part A	Server Database Files.....	89
Part B	Client Application Java Classes.....	108

Abstract

The goal of this project is to investigate and prototype a database driven server for the editing and delivery of multimedia courseware. This project required the analysis, design, and construction of a client/server based, distributed educational system. The components of the project are a relational database server with a particular database schema that can be downloaded or distributed with an existing project and the AudioGraph. The AudioGraph is an application using a multi-media tool to publish university lectures, tutorials or training material on the Web. The front-end interface is a Java application that lets the lecturers or students interact with the database. This system can be used to keep track of various stages of courseware development and web publishing. The overall aim was a flexible and adaptive system with the current lecture development and environment maintained. The system may be distributed on Windows NT, Unix and Macintosh platforms and so is portable and extendible and is platform-independent. The background and technology employed in the project is introduced. Each stage of the project process is explained in terms of the development lifecycle of the system. A limitation imposed by multi-platform compatibility is discussed and the achievement is presented by screenshots. Through the report, the structure of the file, run time environment, inter-process communication, user interface, and server access are explained.

List of Figures

Chapter 1 Introduction

Figure 1.1	Conceptual view of the system -----	3
Figure 1.2	Architecture of the system -----	6
Figure 1.3	Search by 'course-plan' schema -----	9

Chapter 2 Background

Figure 2.1	The classes and interfaces of the JDBC API package -----	22
Figure 2.2	The architecture of the mSQL JDBC driver -----	24

Chapter 3 System Schema

Figure 3.1	The conceptual Entity-Relationship model -----	31
Figure 3.2	Use case diagram for the courseware system -----	33
Figure 3.3	The Enhanced Entity-Relationship model -----	34
Figure 3.4	The DDL for the tables of Courses, Lec159703 and Slides -----	37
Figure 3.5	The database classes in the design model -----	39
Figure 3.6	The interface classes in the design model -----	39

Chapter 4 Implementation

Figure 4.1	The slide schema for the database system -----	42
Figure 4.2	The data of the slide table for the database system -----	43
Figure 4.3	The dumping data of the Slide_Keyword to the database -----	44
Table 4.1	The Core Java Package -----	46
Figure 4.4	CardLayout -----	47
Figure 4.5	The location of Java Virtual Machine in a system -----	52
Figure 4.6	The diagram of the client/server system -----	53
Figure 4.7	The screen shot of the Authentication Window -----	54

Chapter 5 Issues of Cross-platform Compatibility

Figure 5.1	The piece of Java code on Windows NT -----	65
Figure 5.2	The piece of Java code on Macintosh -----	66

Chapter 6 The Results of the System

Figure 6.1	The screen shot of the Login Authoring interface -----	68
Figure 6.2	The screen shot of the Connect Authoring interface -----	68
Figure 6.3	The Course List frame for the user -----	69
Figure 6.4	The Lists of Courses for the staff member 'Paul Lyons' -----	70
Figure 6.5	The Lists of Courses for Computer Science -----	71
Figure 6.6	The Course Content interface of lecture 159201 -----	72
Figure 6.7	The Course Content with the lecture notes of the user -----	72
Figure 6.8	The interface of All Lecture Notes -----	73
Figure 6.9	The interface for browser choice -----	73
Figure 6.10	The screen shot of a Lecture Note Presentation -----	74

Definitions

<i>AudioGraph</i>	A multimedia authoring tool for Web publishing, designed by researchers at Massey University and distributed by NZEdSoft.
<i>API</i>	Application Programming Interface: the group of system tools that allows an application to access the system.
<i>AWT</i>	Abstract Window Toolkit: the Java API that enables programmers to develop Java applications with GUI components.
<i>CASE tool</i>	Computer Aided Software Engineering tool: this refers to any computer-based tool for software planning, development, and evolution.
<i>CORBA</i>	Common Object Request Broker Architecture: the OMG (see OMG definition) platform-independent technique for programs running on different machines to communicate with each other.
<i>COM</i>	Component Object Model: Microsoft's interface specification for hooks into Java for inter-program communication.
<i>DBA</i>	Database Administrator is responsible for the physical realization of the database system.
<i>DBMS</i>	Database Management System: this enables users to create, define and maintain the database and provides access controls to this database.
<i>DCOM</i>	Distributed Component Object Model is a set of Microsoft concepts and program interfaces in which a client program object can request services from server program objects on other computers in a network.
<i>DCE</i>	Distributed Computing Environment: an industry-standard software technology for setting up and managing computing and data exchange in a system of distributed computers.
<i>DDL</i>	Data Definition Language: a language can be used to define the database.
<i>DML</i>	Data Manipulation Language: this enables users to insert, update, delete and retrieve data from the database.
<i>ANSI</i>	The American National Standards Institute, which is the primary organization for fostering the development of technology standards in the United States.
<i>ER</i>	Entity-Relationship is a methodology to model a relational database.
<i>FTP</i>	File Transfer Protocol: this is basically a tool used for transferring and handling files across computers anywhere on the Internet.
<i>GUI</i>	Graphical User Interface to a computer.
<i>HCI</i>	Human-Computer Interaction is the study of how people interact with computers and to what extent computers are or are not developed for successful interaction with human beings.
<i>IDE</i>	The Integrated Development Environment is a collaborative strategic infrastructure embodying data standards that support business processes across a number of geographically dispersed and heterogeneous organizations.

<i>JDBC</i>	Java Database Connectivity is an application program interface specification for connecting programs written in Java to the data in a popular database.
<i>JDK</i>	Java Development Kit: a development environment for writing applets and applications that conform to the Java platform.
<i>JFC</i>	Java Foundation Classes: this extends the original AWT and is portable and compatible with all AWT-based applications.
<i>JIT</i>	Just-in-Time compiler, which improves the performance of Java run time and is used to implement a Java Virtual Machine.
<i>JNMI</i>	Java Native Method Interface
<i>JRE</i>	Java Runtime Environment.
<i>JVM</i>	Java Virtual Machine: this provides an application with a guaranteed run time environment.
<i>MIME</i>	Multi-Purpose Internet Mail Extensions are an extension of the original Internet e-mail protocol that let people use the protocol to exchange different kinds of data files on the Internet.
<i>MRJ</i>	Macintosh Runtime for Java refers to the Java Virtual Machine and associated class libraries for the Macintosh operating system.
<i>mSQL</i>	Mini SQL is a lightweight database engine. It offers a subset of SQL as its query interface.
<i>mSQL-JDBC</i>	This is a JDBC driver of the third kind; A pure-Java JDBC driver for mSQL was created and is being maintained by George Reese from The Center for Imaginary Environments.
<i>MySQL</i>	This is a true multi-user, multi-threaded SQL database server.
<i>ODBC</i>	This stands for Microsoft's Open Database Connectivity; A standard or open application programming interface for accessing a database.
<i>OO Language</i>	Object-Oriented Language: this is used to programming.
<i>RDBMS</i>	Relational Database Management System.
<i>RFC</i>	Request for Comments, in which the FTP commands are described.
<i>RMI</i>	Remote Method Invocation: enables the programmer to create distributed Java-to-Java applications, in which the methods of remote Java objects can be invoked from other Java virtual machines, possibly on different hosts.
<i>SDK</i>	Software Development Kit: a programming package that enables a programmer to develop applications for a specific platform.
<i>SDLC</i>	System Development Life Cycle.
<i>SQL</i>	Standard Query Language is a simple, high-level language used for specifying transactions on a database.
<i>UML</i>	This is a OO language and stands for Unified Modeling Language, which was designed by Grady Booch, James Rumbaugh, and Ivar Jacobson, known as the three amigos.
<i>URL</i>	Uniform Resource Locator; URLs are the mysterious text strings.