A software system for student grading using the Internet

Report submitted in partial fulfillment of the requirement for the Degree of B.Sc.

ELECTRICAL AND ELECTRONIC ENGINEERING

Under the Supervision of Dr. NAGLA OSMAN MOHAMED ABU ALNUR

BY Mustafa Ali Mohammed Ahmed Taha

TO DEPARTMENT OF ELECTRICAL AND ELECTRONIC ENGINEERING

UNIVERSITY OF KHARTOUM
June, 2006
Dedication:

To my parents
Who inspired me with Love and respect towards all people…

To my teachers
Who gave me the keys of knowledge …

To my friends
Who always stand by my side …
Acknowledgement:

Thanks to Electrical and Electronic Engineering staff.
Thanks to Java, MySQL and NetBeans vendors.
Thanks to those who helped to print this project.
Special thanks to our supervisor Dr. Nagla Osman Abo Alnoor.
Abstract:

The main purpose of this project is to develop a software system that enables staff of Electrical and Electronic Engineering to create and process the grading remotely and divides the efforts of process grading to all members, each member do a small task and this leads to decrease the percentage of mistakes and increase the speed of process grading. MYSQL is used as Data Base Management System to create the data base that stores data, and the Java Applets used to interact with user by providing GUI through the web browser.
تجريد:

يهدف المشروع إلى تطوير نظام برمجيات يمكن إضافة قسم الهندسة الكهربائية والالكترونية من إنشاء ومعالجة النتائج عبر الإنترنت، و يقوم بتقسيم مجهود تكوين ومعالجة النتيجة بين كل الأعضاء وذلك بتخصيص مهمة صغيرة لكل عضو مما يقلل نسبة حدوث الأخطاء، و يزيد من سرعة إنشاء النتيجة.

استخدمت MySQL كواحدة من نظام إدارة قواعد البيانات لإنشاء قاعدة البيانات و حفظ البيانات، أيضًا استخدمت تقنية Java لتوفير واجهة المستخدم الرسومية لتعرض داخل (JSP) Java Server Page المتصفح عند استدعاء صفحة.
# Contents

<table>
<thead>
<tr>
<th>Chapter</th>
<th>Title</th>
<th>Pages</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Introduction</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>Overview of Data Base Management System (DBMS)</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>2.1 Data Base Management System properties</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>2.2 The Purpose of an RDBMS</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>2.3 MySQL Database Management System</td>
<td>5</td>
</tr>
<tr>
<td>3</td>
<td>Overview of Java and Java Data Base Connectivity (JDBC)</td>
<td>7</td>
</tr>
<tr>
<td></td>
<td>3.1 What Java Is?</td>
<td>7</td>
</tr>
<tr>
<td></td>
<td>3.2 Where Did Java Come From?</td>
<td>7</td>
</tr>
<tr>
<td></td>
<td>3.3 Java Programming Language</td>
<td>7</td>
</tr>
<tr>
<td></td>
<td>3.4 Java Platform</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>3.5 Why Java's a Better Programming Language?</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>3.6 Java Applets</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>3.7 Java Data Base Connectivity</td>
<td>12</td>
</tr>
<tr>
<td></td>
<td>3.7.1 Overview of JDBC</td>
<td>12</td>
</tr>
<tr>
<td></td>
<td>3.7.2 The JDBC API</td>
<td>12</td>
</tr>
<tr>
<td></td>
<td>3.7.3 The JDBC Architecture</td>
<td>14</td>
</tr>
</tbody>
</table>
Chapter 4: Data base design:

4.1 Data base users 15
4.2 Tables design 15
4.3 Tables properties 17

Chapter 5: Java software implementation

5.1 Description of project files 19
5.1.1 grading.java 19
5.1.2 connect.java 19
5.1.3 login.java 20
5.1.4 admin.java 20
5.1.5 registrar. java 21
5.1.6 teachers. Java 22
5.1.7 developer. Java 22
5.1.8 Index.JSP 22

Chapter 6: Results

6.1 Installation and deployment 23
6.2 Running and testing 23

Chapter 7: Conclusion and recommendation:

7.1 Conclusion 36

7.2: Recommendation 37

References 38
## List of figures:

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1</td>
<td>How an applet is displayed by a Web browser.</td>
<td>10</td>
</tr>
<tr>
<td>3.2</td>
<td>JDBC drivers</td>
<td>13</td>
</tr>
<tr>
<td>3.3</td>
<td>The JDBC architecture</td>
<td>14</td>
</tr>
<tr>
<td>4.1</td>
<td>Entity relationship diagram.</td>
<td>16</td>
</tr>
<tr>
<td>6.1</td>
<td>MySQL screen</td>
<td>25</td>
</tr>
<tr>
<td>6.2</td>
<td>NetBeans server starting</td>
<td>26</td>
</tr>
<tr>
<td>6.3</td>
<td>Java virtual machine not installed in client</td>
<td>27</td>
</tr>
<tr>
<td>6.4</td>
<td>Login screen</td>
<td>28</td>
</tr>
<tr>
<td>6.5</td>
<td>Administrator's teachers panel</td>
<td>29</td>
</tr>
<tr>
<td>6.6</td>
<td>Administrator's subjects panel</td>
<td>30</td>
</tr>
<tr>
<td>6.7</td>
<td>Administrator's classes panel</td>
<td>31</td>
</tr>
<tr>
<td>6.8</td>
<td>Administrator's grading panel</td>
<td>32</td>
</tr>
<tr>
<td>6.9</td>
<td>registrars panel</td>
<td>33</td>
</tr>
<tr>
<td>6.10</td>
<td>Teachers panel</td>
<td>34</td>
</tr>
<tr>
<td>6.11</td>
<td>Developers panel</td>
<td>35</td>
</tr>
</tbody>
</table>
Chapter 1: Introduction

Grading is one of the important issues in education, used to be done in one computer by one person or a small group, it used to be a hard and ambiguous for them, by the introduction of the internet in the last few years it became necessary to develop a software system that makes use of the internet.

The Internet is the worldwide, publicly accessible network of interconnected computer networks that transmit data by packet switching using the standard Internet Protocol (IP). It consists of millions of smaller domestic, academic, business, and government networks, which together carry various information and services, such as electronic mail, online chat, and the interlinked Web pages and other documents of the World Wide Web.

The internet became important global resource to all of the people in the world, because of:

- Geographic Distribution
- Internet has high Speeds
- Universal Internet Access
- Internet Growth Rates

To develop the grading software system there should be a data base to store data and a Graphical User Interface to help users to interact with the system, and the system connected to the internet.

The project consists of the following chapters:

Chapter 2: Overview of Data Base Management System (DBMS)

This chapter gives an overview of DBMS's and data bases which are structured collections of data used to add, access, and process data stored in a computer database.

A database management system such as MySQL Server is needed, the main purpose of Database Management System is to organize data so that it can be retrieved and manipulated by an application or a user.
Chapter 3: Overview of Java and Java Data Base Connectivity (JDBC)

This chapter gives an overview of the programming language used to develop the system, Java. Java is a simple, object-oriented, distributed, interpreted, robust, secure, architecture-neutral, portable, high-performance, multithreaded and dynamic language[1].

The Java platform is a software-only platform that runs on top of other hardware-based platforms, Java technology has many platforms; each platform is based on a Java VM that has been ported to the target hardware environment

Chapter 4: Data base design:

This chapter shows the design of tables needed to develop the required data base for the project.

Chapter 5: Java software implementation:

This chapter shows the main files used in the project and there related tasks.

Chapter 6: Results:

This chapter shows the Graphical User Interface of the system as it is running successfully.

Chapter 7: Conclusion and recommendation:

This chapter illustrates general steps for development of the system, and recommendations for developers to extend the system.
Chapter 2: Overview of Data Base Management System (DBMS):

2.1 Data Base Management System properties:

Databases have been in use since the earliest days of electronic computing, but the vast majority of these were custom programs written to access custom databases. Unlike modern systems which can be applied to widely different databases and needs, these systems were tightly linked to the database in order to gain speed at the expense of flexibility.

The main purpose of Database Management System is to organize data so that it can be retrieved and manipulated by an application or a user. Since data takes many forms and its volume may range from very small to extremely large, creating a consistent method for managing that data is a daunting task.

A good Database Management System (DBMS) provides the following functionality:

Data Definition: A DBMS must define a structure for stored data, and provide a means for a user to define and organize their own data within that structure.

Data Retrieval: A DBMS must provide a tool set that allows a user to retrieve data stored in the database. This tool set can take many forms, such as programming APIs and simple query tools.

Data manipulation: Database users must be able to add new data, change existing data, or delete existing data.

Access control: The database administrator should be able to define data access by individual or group. A single DBMS can manage data that can be seen or manipulated by only one individual while also providing other data that can be viewed by the entire user community. It should detect and prevent any unauthorized access.

Data sharing: More than one user should be able to use the database at the same time without fear of overwriting each other's data changes.

Data integrity: A DBMS should provide mechanisms for maintaining data integrity through system failures and inconsistent, or incomplete, updates.
To achieve this functionality, a DBMS defines a data structure and access techniques collectively called the "data model". The DBMS's data model really defines how well it is suited to a particular application.

**The four major data models are:**

- **File Management:**

  All data is stored in individual files. Before true database management became a science, such file management was an early attempt at implementing databases.

- **Hierarchical database:**

  Organizes data in a tree structure, sometimes called a Bill-Of-Materials structure. A single defined parent record and any number of child records explicitly links together all information in the database.

- **Network database:**

  Extends the Hierarchical model by defining multiple parents. The relationships between specific records of one-to-one, one-to-many, or many-to-many are explicitly defined in the database definition. The network data model allows us to define, for example, a book database where an author can have many books and a book could have more than one author.

- **Relational database:**

  The relational data model eliminates the explicit definition of parent/child relationships and organizes data into rows and columns in distinct tables. While relationships between tables can be explicitly defined in the relational data model, the tools used to access and manipulate the data can also define relationships dynamically.

**2.2 The Purpose of an RDBMS:**

An RDBMS exists for the purpose of providing a reliable permanent storage mechanism with very strict properties collected in the ACID test, these properties are:

- **Atomicity**

  Results of a transaction's execution are either all committed or all rolled back. All changes take effect, or none do.
• **Consistency**
  The database is transformed from one valid state to another valid state. A transaction is legal only if it obeys user-defined integrity constraints. Illegal transactions aren't allowed and, if an integrity constraint can't be satisfied the transaction is rolled back.

• **Isolation**
  The results of a transaction are invisible to other transactions until the transaction is complete.

• **Durability**
  Once committed (completed), the results of a transaction are permanent and survive future system and media failures.

### 2.3 MySQL Database Management System:

A database is a structured collection of data. To add, access, and process data stored in a computer database, a database management system such as MySQL Server is needed. Since computers are very good at handling large amounts of data, database management systems play a central role in computing, as standalone utilities or as parts of other applications.

MySQL, the most popular Open Source SQL database management system, is developed, distributed, and supported by MySQL AB. MySQL AB is a commercial company, founded by the MySQL developers, that builds its business by providing services around the MySQL.

#### 2.3.1 MySQL is a relational database management system.

A relational database stores data in separate tables rather than putting all the data in one big storeroom. This adds speed and flexibility. The SQL part of "MySQL" stands for "Structured Query Language." SQL is the most common standardized language used to access databases and is defined by the ANSI/ISO SQL Standard. The SQL standard has been evolving since 1986 and several versions exist.
2.3.2 MySQL software is Open Source.

Open Source means that it is possible for anyone to use and modify the software. Anybody can download the MySQL software from the Internet and use it without paying anything.

2.3.3 The MySQL Database Server is very fast, reliable, and easy to use.

MySQL Server was originally developed to handle large databases much faster than existing solutions and has been successfully used in highly demanding production environments for several years. Although under constant development, MySQL Server today offers a rich and useful set of functions. Its connectivity, speed, and security make MySQL Server highly suited for accessing databases on the Internet.

2.3.4 MySQL Server works in client/server or embedded systems.

The MySQL Database Software is a client/server system that consists of a multi-threaded SQL server that supports different back ends, several different client programs and libraries, administrative tools, and a wide range of application programming interfaces (APIs). Also MySQL Server is an embedded multi-threaded library that can be linked into application to get a smaller, faster, easier-to-manage product.
Chapter 3: Overview of Java and Java Data Base Connectivity (JDBC)

3.1 What Java Is?

Java is a high-level, third generation programming language, like C, Fortran, Smalltalk, Perl, and many others. Java can be used to write computer applications that manipulate numbers, process words, play games, store data or do any of the thousands of other things computer software can do.

Compared to other programming languages, Java is most similar to C. However although Java shares much of C's syntax, it is not C. Knowing how to program in C or C++ will certainly help to learn Java more quickly; but there is no need to know C to learn Java.

3.2 Where Did Java Come From?

A few Sun old-timers, most specifically including James Gosling, needed a programming language for a new set of consumer-electronics devices they were building. James Gosling developed the language, originally called Oak, for this purpose. Upon finding out that the name Oak was taken, a late-night meeting at a local cafe determined the new name: "Java."

The Java programming language was developed as a small-footprint object-oriented programming language loosely based on C++ that would be processor-independent, The Public announcement of Java done in 1995. After going a while down the path of consumer-electronics devices, they realized that they had something particularly cool in the Java language, and focused on it as a language for network computing. Sun formed the JavaSoft group, which in a little over three years has grown to over six hundred people working on Java-related technologies.

3.3 Java Programming Language:

The Java programming language enables writing powerful, enterprise programs that run in the browser, from the desktop, on a server, or on a consumer device. Java programs are run on -- interpreted by -- another program called the Java Virtual Machine (Java VM). Rather than running directly on the native operating system, the program is interpreted by the Java VM for the native operating system.
3.4 Java Platform:

The Java platform is a software-only platform that runs on top of other hardware-based platforms. Because hardware-based platforms vary in their storage, memory, network connection, and computing power capabilities, specialized Java platforms are available to address applications development for and deployment to those different environments. Java technology has many platforms; each platform is based on a Java VM that has been ported to the target hardware environment. The java different platforms are: -

Java 2 Platform, Standard Edition (J2SE), provides an environment for Core Java and Desktop Java applications development.

Java 2 Platform, Enterprise Edition (J2EE), defines the standard for developing component-based multi tier enterprise applications.

Java 2 Platform, Micro Edition (J2ME), is a set of technologies and specifications targeted at consumer and embedded devices, such as mobile phones, personal digital assistants (PDA’s), printers.

Java Card technology, adapts the Java platform to enable smart cards and other intelligent devices with limited memory and processing capabilities to benefit from many of the advantages of Java technology.

3.5 Why Java's a Better Programming Language?

The Java language shares many superficial similarities with C, C++, and Objective C. For instance, loops have identical syntax in all four languages, However, Java is not based on any of these languages, nor have efforts been made to make it compatible with them.

Java is sometimes referred to as C++++--. James Gosling invented Java because C++ proved inadequate for certain tasks. Since Java's designers were not burdened with compatibility with existing languages, they were able to learn from the experience and mistakes of previous object-oriented languages. They added a few things C++ doesn't have like garbage collection and multithreading (the ++) and they threw away C++ features that had proven to be better in theory than in practice like multiple inheritance and operator overloading (the --). A few advanced features like closures and parameterized types that the Java team liked were nonetheless left out of the language due to time constraints. There's still argument over whether the right choices were made.
3.5.1 Java is Simple

Java was designed to make it much easier to write bug free code. Shipping C code has, on average, one bug per 55 lines of code. The most important part of helping programmers write bug-free code is keeping the language simple and easy.

3.5.2 Java is Object-Oriented

In object-oriented programs data is represented by objects. Objects have two sections, fields (instance variables) and methods. Fields tell us what an object is. Methods tell us what an object does.

Object oriented programming Provides abstraction, encapsulation, inheritance, polymorphism and any code that we write in Java is inside a class.

3.5.3 Java is Platform Independent

Java was designed to not only be cross-platform in source form like C, but also in compiled binary form. Since this is frankly impossible across processor architectures, Java is compiled to an intermediate form called byte-code.

3.5.4 Java is Safe

Java was designed from the ground up to allow for secure execution of code across a network, even when the source of that code was untrusted and possibly malicious. And it provides strong type-checking on user’s computer.

3.5.6 Java is High Performance

Comparable in speed to other compiler based programs. Java is faster than other interpreter based languages as it is compiled too.

3.5.7 Java is Multi-Threaded

Java is inherently multi-threaded. A single Java program can have many different processes executing independently and continuously.
3.5.8 Java is dynamic

Maintaining different versions of an application is easy. Possible to make interconnections between modules at run-time. Avoids problems caused by change of code used by our program.

3.5.9 Java is distributed

Can built distributed applications that use resources from any other networked computer.

3.5.10 Internationalization

Internationalization is the ability to translate programs to other languages.

3.5.11 Java forces a programmer to perform error processing.

It is mandatory in Java to handle possible run-time errors. It has a mechanism that is called "Exceptions" and if a method is called, that may give a run-time error, this program will not be executed until these possible errors have taken care of.

3.6 Java Applets:

Applets are Java programs that are integrated in Web pages. When a Web page containing an applet is displayed by a Web browser, the applet is loaded and executed. The applet's output is displayed within a subset of the browser's display area. The Applet class is a subclass of the Panel class, and applets are implemented as a panel within a Web document.

![Browser Window][1]

*Figure 3.1: How an applet is displayed by a Web browser.*
Advantages of an applet:-

- Needs no explicit installation on local machine.
- Loads and runs itself automatically in a Java-enabled browser.
- Get parameters specified by the user in the <APPLET> tag.
- Applets can usually make network connections to the host they came from.
- Applets running within a Web browser can easily cause HTML documents to be displayed.
- Applets can invoke public methods of other applets on the same page.

With these capabilities / advantages applets is a choice for the client tier.

What Applets Can't Do?

Every browser implements security policies to keep applets from compromising system security. Current browsers impose the following restrictions on any applet that is loaded over the network:

- An applet cannot load libraries or define native methods.
- It cannot ordinarily read or write files on the host that's executing it.
- It cannot make network connections except to the host that it came from.
- It cannot start any program on the host that's executing it.
- It cannot read certain system properties.
- Windows that an applet brings up look different than windows that an application brings up.

Each browser has a Security Manager object that implements its security policies. When a Security Manager detects a violation, it throws a Security Exception.
3.7 Java Data Base Connectivity:

3.7.1 Overview of JDBC:

JDBC are programming interfaces allowing external access to SQL database manipulation and update commands. They allow the integration of SQL calls into a general programming environment by providing library routines that interface with the database. In particular, Java based JDBC has a rich collection of routines which make such an interface extremely simple and intuitive.

A normal Java program is written. Somewhere in the program, interaction with to a database is needed. Using standard library routines, a connection is opened to the database and then use JDBC to send SQL code to the database, and process the results that are returned. When this is done, the connection is closed.

3.7.2 The JDBC API

Sun has created four classifications of JDBC drivers based on their architectures. Each JDBC driver classification represents a trade-off between performance and flexibility.

Type 1

Type 1 drivers use a bridging technology to access a database. The JDBC-ODBC bridge that comes with JDK 1.2 is the most common example of this kind of driver.

Type 2

Type 2 drivers are native API drivers. "Native API" means that the driver contains Java code that calls native C or C++ methods provided by the database vendor.
Type 3
Type 3 drivers provide a client with a pure Java implementation of the JDBC API in which the driver uses a network protocol to talk to middleware on the server. This middleware, in turn, performs the actual database access. The middleware may or may not use JDBC for its database access.

Type 4
Using network protocols built into the database engine, Type 4 drivers talk directly to the database using Java sockets. This is a pure Java solution. Because these network protocols are almost never documented, most Type 4 drivers come from the database vendors.

Figure 3.2 shows the different JDBC driver types.
3.7.3 The JDBC Architecture

JDBC is a set of interfaces implemented by different vendors. The Figure 3.3 shows how database access works from an application's perspective. Essentially, the application simply makes method calls to the JDBC interfaces. Under the hood, the implementation being used by that application performs the actual database calls.

![Figure 3.3 The JDBC architecture](image)

Figure 3.3 The JDBC architecture
Chapter 4: Data base design:

4.1 Data base users

In the grading system there should be four types of users, each type with specific privileges and tasks, which are administrators, registrars, teachers, and developers.

4.1.1 Administrator

The administrator has the following tasks:

- Add, delete, search and update and maintain privilege for other users (teachers, registrars, developers).
- Add, delete, search and update subjects to the system.
- Assign subjects to each class.
- Process and show the grading for the specific class.

4.1.2 Registrar

The registrar's task is to add, delete, search and update students in each class.

4.1.3 Teacher

When a teacher log to the system he finds his own subject, chooses one of them, finds all students who learn that subject, and then teacher can assign marks to both paper and course work.

4.1.4 Developer

Developer can handle problems happened with the data base.

4.2 Tables design:

To meet these requirements, grading data base is designed where data will be stored in tables that have specific properties of fields and relationships; relations between tables should be determined for a perfect design of a relational data base.

Tables needed in the design of the student grading Data Base should reflect relations between teachers, students, subjects and classes.
Tables are organized as follows:

4.2.1 Teachers table:

A table that contains data about each teacher, it consists of four fields represent teacher id, name, password and privilege.
4.2.2 Subjects tables:
A table that contains data about each subject, it consists of three fields represent subject’s code, name and teacher. And a table for each subject (e.g. EC1101, EC1102…etc) that consists of three fields represent student’s index, paper mark and course work mark.

4.2.3 Classes tables:
A table for each class (e.g. First general, second general…etc) with one field that contains all class subjects.

4.2.4 Students table:
A table that contains data about each student, it consists of three fields represent student index, name, and class.

4.3 Tables properties:
The properties of fields within tables, schemas of tables and relations between tables are shown below:

4.3.1 Teachers:

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Null</th>
<th>Unique</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>VARCHAR (6)</td>
<td>No</td>
<td>yes</td>
</tr>
<tr>
<td>name</td>
<td>VARCHAR (50)</td>
<td>No</td>
<td>yes</td>
</tr>
<tr>
<td>tprivilage</td>
<td>VARCHAR (20)</td>
<td>No</td>
<td>No</td>
</tr>
</tbody>
</table>

CREATE TABLE teachers (id VARCHAR (6) NOT NULL UNIQUE, name VARCHAR (50) NOT NULL UNIQUE, password VARCHAR (20), tprivilage VARCHAR (20) NOT NULL);

4.3.2 Subjects:

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Null</th>
<th>Unique</th>
</tr>
</thead>
<tbody>
<tr>
<td>scode</td>
<td>VARCHAR (6)</td>
<td>No</td>
<td>yes</td>
</tr>
<tr>
<td>sname</td>
<td>VARCHAR (50)</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>steacher</td>
<td>VARCHAR (50)</td>
<td>No</td>
<td>No</td>
</tr>
</tbody>
</table>

CREATE TABLE subjects (scode VARCHAR (6) NOT NULL, sname VARCHAR (50) NOT NULL, steacher VARCHAR (50) NOT NULL, UNIQUE(scode));
4.3.3 Subject (e.g. EC1101, EC1102…etc):

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Null</th>
<th>Unique</th>
</tr>
</thead>
<tbody>
<tr>
<td>sindex</td>
<td>VARCHAR (6)</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>paper</td>
<td>UNSIGNED INT (2)</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>cw</td>
<td>UNSIGNED INT (2)</td>
<td>Yes</td>
<td>No</td>
</tr>
</tbody>
</table>

CREATE TABLE EC1101 (sindex VARCHAR (6), paper INT (2) UNSIGNED, cw INT (2) UNSIGNED, UNIQUE(sindex)) ;

4.3.4 Students :

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Null</th>
<th>Unique</th>
</tr>
</thead>
<tbody>
<tr>
<td>sindex</td>
<td>VARCHAR (6)</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>sname</td>
<td>VARCHAR (50)</td>
<td>yes</td>
<td>No</td>
</tr>
<tr>
<td>sclass</td>
<td>VARCHAR (20)</td>
<td>yes</td>
<td>No</td>
</tr>
</tbody>
</table>

CREATE TABLE students (sindex VARCHAR (6), sname VARCHAR (50), sclass VARCHAR (20), UNIQUE(sindex)) ;

4.3.5 Class (e.g. First general, second general…etc):

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Null</th>
<th>Unique</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class_subject</td>
<td>VARCHAR (50)</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

CREATE TABLE first_general (Class_subject VARCHAR (50), UNIQUE(Class_subject)) ;
Chapter 5: Java software implementation:

5.1 Description of project files:

A after the design of the data base, Java is used to provide a Graphical User Interface (GUI) and to provide some operations on the data and to be used with HTML to be loaded on the internet. There are seven main classes, each located on its own file, and another file (JSP file) contains HTML tag , and each class contains a number of data members and methods and performs specific task .

These files are

- grading.java
- connect.java
- login.java
- admin.java
- registrar.java
- teacher.java
- developer.java
- index.jsp

5.1.1 grading.java

consists of the following methods :-

- grading()
  
  It is the constructor of the grading class, it initiate the Graphical User Interface and the connection to the data base.

- init()
  
  It is inhertly the method that java applets start execution with and initiate look and feel.

5.1.2 connect.java

It loads the JDBC driver and establish the connection with data base.
5.1.3 login.java

consists of the following methods :-

- login()
  It is the constructor of the login class, it initiate the Graphical User Interface and the connection to the data base.

- do_login()
  Takes user name and password as the parameters and verify user name and password and get user's privileges and initiate the appropriate class.

5.1.4 admin.java

consists of the following methods :-

- admin()
  It is the constructor of the admin class, it initiate the Graphical User Interface and the connection to the data base.

When loading the GUI of the admin class, user can choose one of four panels that contain the major tasks of the administrator, these panels and their methods are :

- teachersPanel
  has the methods :
  
  - add_teacher()
    A method used to add teachers to the data base taking teacher ID, name, password and privilege as parameters.
  
  - delete_teacher()
    A method used to delete teachers from the data base taking teacher ID as parameter.
  
  - update_teacher()
    A method used to update teachers information to the data base taking teacher ID, name, password and privilege as parameters.
  
  - search()
    A method used to delete teachers from the data base taking teacher ID as parameter, and returns teacher's name, password and privilege.
  
  - report()
    A method used to display all staff data stored in the data base.
- subjectsPanel
  has the methods:
    - add_subject()
    - delete_subject()
    - update_subject()
    - ssearch()
    - sreport()
  As the names of these methods suggest, they are used to add, delete, update, search and give the report of subjects.

- subjectsPanel
  has the methods:
    - add_Csubject() , delete_Csubject()
      Adds subjects to a specific class
    - creport()
      shows subjects assigned to each class.

- gradingPanel
  has the method:
    - process_grading()
      when this method used, taking the class name as a parameter of it, it process the grading for this class by showing the indices and associated paper and course work marks of each student of that class.

5.1.5 registrar.java
  consists of the following methods:
    - registrar()
      It is the constructor of the registrar class, it initiate the Graphical User Interface and the connection to the data base.
    - add_student()
    - delete_student()
    - update_student()
    - search()
    - show_students()
  As the names of these methods suggest, they are used to add, delete, update, search and give the report of students.
5.1.6 teachers. java

Consists of the following methods:-

- **teachers()**
  It is the constructor of the teachers class, it initiate the Graphical User Interface and the connection to the data base.

- **show_students()**
  Shows all students in the class.

- **search()**
  It is used to show the student's marks.

- **insert_grade()**
  It is used to insert paper and course work marks for a student and then goes to the next index automatically.

- **cancel()**
  It goes to the next index without modifying the selected index.

5.1.7 developer. java

Consists of the following methods:-

- **developer()**
  It is the constructor of the teachers class, it initiate the Graphical User Interface and the connection to the data base.

- **execute_query()**
  It receives an SQL query and then executes it.

5.1.8 Index.JSP

It is a file that contain a tag for calling grading.class file to start the applet in the client's browser.
Chapter 6: Results:

6.1 Installation and deployment:

- **MY-SQL Server 4.1** is installed into the system's port 3306, and grading data base was created with mentioned tables (Figure 6.1).

- Java **NetBeans 4.1** is installed, and is used in:
  1. Collecting project files.
  2. Running **sun java System Application server 8.1** on systems port 4848 (Figure 6.2).

- Server is configured with an IP address, and port number 8080.

6.2 Running and testing:

In running the project, the following panels are displayed:-

If the client's machines don’t have Java Virtual Machine (JVM), the following screen will be displayed (Figure 6.3), and if JVM is installed the following panel is displayed:

6.2.1 Login panel:

Used for inputting user name and password, showing error massages when introducing wrong or empty user names or passwords; and it calls one of four panels depending on the user privilege (Figure 6.4).

6.2.2 Administrator panel:

Contains four panels for the administrator, they are:

- The first panel is teachers panel it used for input other user data (add, delete, modify, search) (Figure 6.5).

- The next panel is subjects panel it used by administrator to (add, delete, modify, search) subjects (Figure 6.6).

- The next panel is classes panel it used by administrator to (add, delete, modify) class subjects (Figure 6.7).

- The last panel is grading panel it used by administrator to process the grading (Figure 6.8).
6.2.3 Register:

Inputting student’s data (index, name, class,) as shown in (Figure 6.9).

6.2.4 Teacher:

Inputting student’s marks after selecting subject and index as shown in (Figure 6.10).

6.2.5 Developer:

Writing SQL quires that can be executed as shown in (Figure 6.11).
Chapter 7: Conclusion and recommendation:

7.1 Conclusion

- The software design model used in the project was the evolutionary model, starting with a very simple implementation of the project, and extend it into bigger versions until the final implementation of the system obtained. With this model implementation was not hard, but changes were very frequent in the design of whole system.

- The JDBC driver is needed to establish the connection between Java applets and data base, but java applets can not do anything inside client desk, so there were problems with Java security manager when trying to load JDBC driver. As a result, connection could not be established, until NetBeans IDE4.1 program that includes sun Java system Application Server 8.1 is installed, and project files (including applets and MYSQL JDBC driver) are collected, and JSP pages loads the driver and applets successfully.

- Loading time of Java applet is slow, because all applet’s byte code is loaded into client, but after loading an Applet, it works as if it was an ordinary application in the user’s machine, also Applets have problem with loading images when trying to display them in the login screen.

- The source code of the project and the data base files are attached in the project’s soft copy
7.2: Recommendation

- Classical client/server programming model (2-tier System) was used in the design; But when using multi-tier design which is consist of client, business logic and data base, client can get only applet that contains GUI, and in server side there is a business logic implementation, and a data base in a data base server, this approach is better in performance and security issues, but it is complex compared to the Client/server approach.

- In the future, an additional method that processes student's result to be pasted on the board can be added to the interface without affecting database. Also another method that modifies all marks in fixed rates can be added.

- Also additional methods that enable students to display their marks can be added.
References:

Books:


Websites:

www.itworld.com

http://www.mysql.com/


http://en.wikipedia.org/wiki/MySQL

http://www.infoweblinks.com/content/dbms.