**VISVESVARAYA TECHNOLOGICAL UNIVERSITY**

**“Jnanasangama”, Belagavi-590018, Karnataka**

****

**BANGALORE INSTITUTE OF TECHNOLOGY**

**K.R. Road, V.V.Puram, Bengaluru-560004**



**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

**DATABASE MANAGEMENT SYSTEM MINI PROJECT**

**18CSL58**

**“UNIVERSITY MANAGEMENT SYSTEM”**

**Submitted By**

**Sanjeev Kumar Mendegar**

**1BI18CS134**

**for the academic year 2020-21**

**Department of Computer Science & Engineering**

**Bangalore Institute of Technology**

**K.R. Road, V.V.Puram, Bengaluru-560 004**

**VISVESVARAYA TECHNOLOGICAL UNIVERSITY**

**“Jnanasangama”, Belagavi-590018, Karnataka**

**BANGALORE INSTITUTE OF TECHNOLOGY**

**K.R. Road, V.V.Puram, Bengaluru-560 004**

****

**Department of Computer Science & Engineering**

***Certificate***

This is to certify that the implementation of **DBMS MINI PROJECT** entitled   
**“University Management System”** has been successfully completed by **USN:1BI18CS134 NAME:SANJEEV KUMAR MENDEGAR**

of V semester B.E. for the partial fulfillment of the requirements for the Bachelor's degree in **Computer Science & Engineerin**g of the **Visvesvaraya Technological University** during the academic year **2020-2021.**

**Lab Incharges :**

**Dr Savitha S. K. Dr.Asha T**

Assistant Professor Professor and Head

Dept. of CS&E Department of CS&E

Bangalore Institute of Technology BIT

Bengaluru Bengaluru

**Prof.Kanchan A.P.**

Assistant Professor

Dept. of CS&E

Bangalore Institute of Technology

Bengaluru

Examiners: 1) 2)

**ACKNOWLEDGEMENT**

The knowledge & satisfaction that accompany the successful completion of any task would be incomplete without mention of people who made it possible, whose guidance and encouragement crowned my effort with success. I would like to thank all and acknowledge the help I have received to carry out this Mini Project.

I would like to convey my thanks to Head of Department **Dr. ASHA T.** for being kind enough to provide the necessary support to carry out the mini project.

I am most humbled to mention the enthusiastic influence provided by the lab in-charges **Dr Savitha S. K.** and **Prof. Kanchan A. P.,** on the project for their ideas, time to time suggestions for being a constant guide and co-operation showed during the venture and making this project a great success.

I would also take this opportunity to thank my friends and family for their constant support and help. I'm very much pleasured to express my sincere gratitude to the friendly co-operation showed by all the staff members of Computer Science Department, BIT

SANJEEV KUMAR MENDEGAR

1BI18CS134

**Table of Contents**

|  |  |  |  |
| --- | --- | --- | --- |
| **1.** |  | **Introduction** |  |
| **1.1** | Introduction | 2 |
| **1.2** | Problem Statement | 2 |
| **2.** |  | **Back end Design** |  |
| **2.1** | Conceptual Database Design | 4 |
| **2.2** | Logical Database Design | 5 |
| **2.3** | Normalization | 6-10 |
| **3** |  | **Front End Design** |  |
| **3.1** | Screen Layout Design For Pages, Forms | 12-13 |
| **3.2** | Connectivity to the Database from Front End | 13-15 |
| **4** |  | **Major Modules** | 16-17 |
| **5** |  | **Implementation** |  |
| **5.1** | MySQL(Database Code) | 19-21 |
| **5.2** | Java(Front End Code) | 22-35 |
| **6** |  | **Snapshots** |  |
| **6.1** | Snapshots | 37-42 |
| **6.2** | Testing | 43 |
| **7** |  | **Applications** |  |
| **7.1** | Applications | 45 |
| **8** |  | **Conclusion** |  |
| **8.1** | Conclusion | 47 |

**List of Figures**

|  |  |  |  |
| --- | --- | --- | --- |
| **2** | **2.1.1** | ER-Diagram - University Management System | 4 |
| **2.2.1** | Relational Mapping - University Management System | 5 |
| **6** | **6.1.1** | Login | 37 |
| **6.1.2** | Add Student | 37 |
| **6.1.3** | Student Details UI | 38 |
| **6.1.4** | Student Attendance UI | 38 |
| **6.1.5** | Examinations UI | 39 |
| **6.1.6** | Faculty UI | 39 |
| **6.1.7** | Faculty Details UI | 40 |
| **6.1.8** | Update Faculty UI | 40 |
| **6.1.9** | Update Student UI | 41 |
| **6.1.10** | Fee UI | 41 |
| **6.1.11** | Fee Structure UI | 42 |
| **6.1.12** | About us UI | 42 |

**CHAPTER-1**

**INTRODUCTION**

**INTRODUCTION**

**1.1 Introduction**

University Management System is an online platform developed for employee to maintain the data of university in an easy and efficient way. The purpose of the software project is to develop the Management Information System (MIS) to automate the record keeping of student entry, teacher entry and fee entry with a view to enhance the decision making of the functionaries.

A MIS mainly consists of a computerized database, a collection of inter-related tables for a particular subject or purpose, capable to produce different reports relevant to the user. An application program is tied with the database for easy access and interface to the database. Using Application program or front-end, we can store, retrieve and manage all information in proper way. This software, being simple in design and working, does not require much of training to users, and can be used as a powerful tool for automating a University Management System.

## 1.2 PROBLEM STATEMENT

**UNIVERSITY MANAGEMENT SYSTEM** is a Desktop Application project here we can manage the details of Student, Faculty, Attendance, Examination, Fee from this project. This system is used to store data over a centralized server which consists of database where information cannot be accessed by a third party. The main objective to develop University Management System Project is to overcome the manual errors and make a computerized system.

**CHAPTER-2**

**BACK END DESIGN**

**BACK END DESIGN**

**2.1 Conceptual database design**

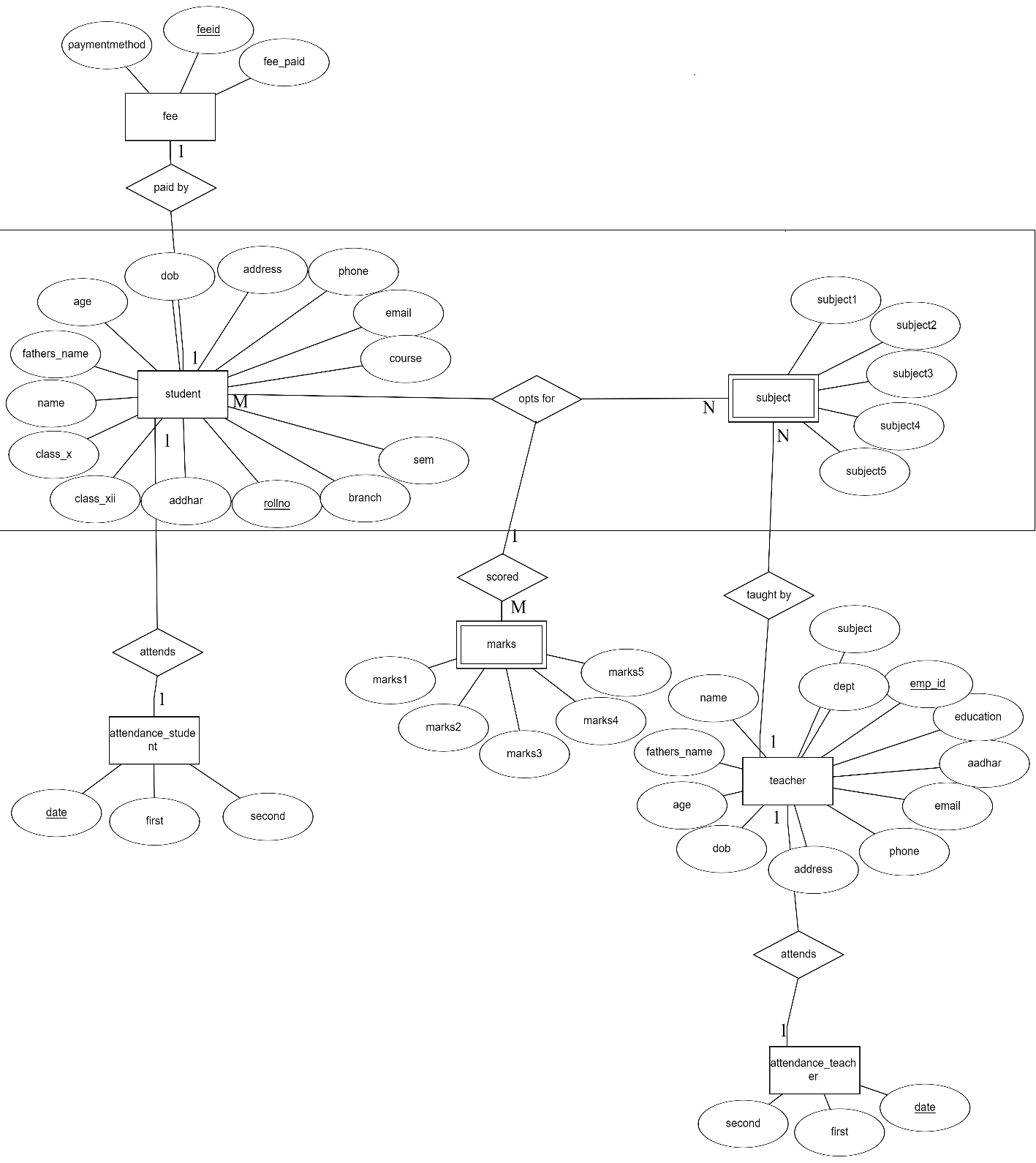


Figure 2.1.1 ER-Diagram – University Management System

## 2.2 Logical database design

STUDENT(Name,Fathers\_name,Age,DOB,Address,Phone,Email,Class\_X,Class\_XII,Aaddhar,RollNo,Course,Branch,Sem)

FACULTY(Name,Fathers\_name,Age,DOB,Address,Phone,Email,Aaddhar,Education,Emp\_id,Dept)

FEE (RollNo, Name, Course, Branch, Semester, Fee\_paid, Fee\_id, PaymentMethod)

TEACHER\_ATTENDANCE (Emp\_id, Date, First, Second)

STUDENT\_ATTENDANCE (Emp\_id, Date, First, Second)

SUBJECT (RollNo, Subject1Subject2, Subject3,Subject4,Subject5)

MARKS ( RollNo , Marks1, Marks2, Marks3, Marks4, Marks5)

## 

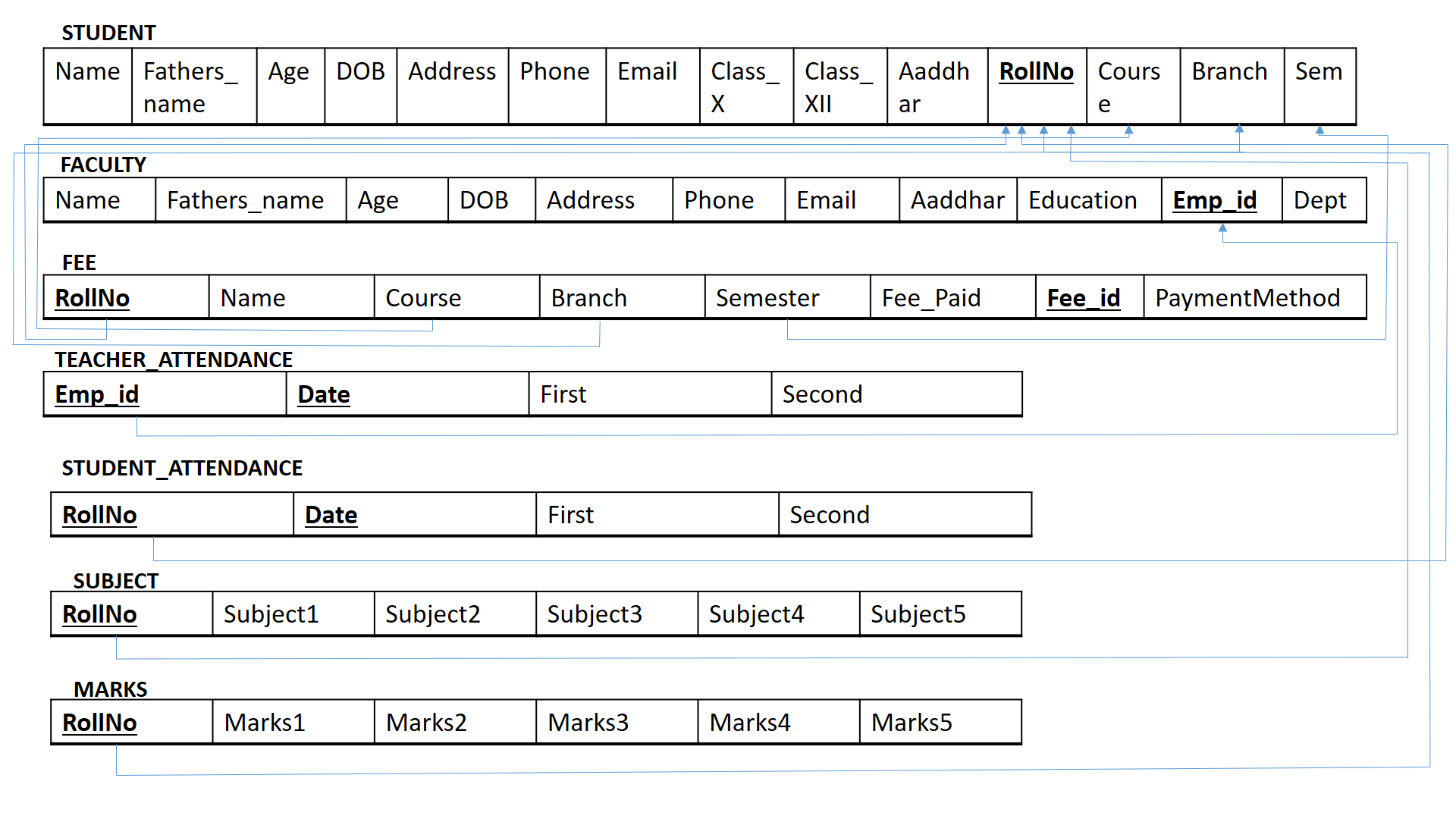


Figure 2.2.1 Relational Mapping - University Management System

**2.3 Normalization**

Database Normalization is a technique of organizing the data in the database. Normalization is a systematic approach of decomposing tables to eliminate data redundancy and undesirable characteristics like Insertion, Update and Deletion Anomalies. It is a multi- step process that puts data into tabular form by removing duplicated data from the relation tables.

Normalization is used for mainly two purposes,

* Eliminating redundant(useless) data.
* Ensuring data dependencies make sense i.e. data id logically stored.

**First Normal Form (1NF) :**

As per First Normal Form

* There are no duplicated rows in the table.
* Each attribute is single valued or atomic.

**Second Normal Form (2NF) :**

As per Second Normal Form, a table is in 2NF if every non prime attribute is not partially dependent on any key of the table.

**Third Normal Form (3NF) :**

Third Normal Form applies that every non-prime attribute of table must be dependent on primary key, or we can say that, there should not be the case that a non-prime attribute is determined by another non-prime attribute. So this transitive functional dependency should be removed from the table and also the table must be in the Second Normal Form.

**NORMALIZATION OF STUDENT:**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Name | Fathers\_name | Age | Dob | Address | Phone | Email | Class\_x | Class\_xii | Aadhar | Rollno | Course | Branch | sem |

First Normal Form

In STUDENT, all the attributes are atomic and there cannot be duplicate rows.

Hence, it is in 1NF.

Second Normal Form

This is already in 2NF since every non key attribute is fully dependent on primary key.

Third Normal Form

Since there is no transitive functional dependency, therefore table is already in 3NF.

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Name | Fathers\_name | Age | Dob | Address | Phone | Email | Aadhar | Education | Department | Emp\_id | Subject |

**NORMALIZATION OF TEACHER:**

First Normal Form

In TEACHER, all the attributes are atomic and there cannot be duplicate rows. Hence, it is in 1NF.

Second Normal Form

This is already in 2NF since every non key attribute is fully dependent on primary key.

Third Normal Form

Since there is no transitive functional dependency, therefore table is already in 3NF.

**NORMALIZATION OF FEE:**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| rollno | Name | Course | Branch | Semester | Fee\_paid | Feeid | paymentmethod |

First Normal Form

In FEE, all the attributes are atomic and there cannot be duplicate rows. Hence, it is in 1NF.

Second Normal Form

This is already in 2NF since every non key attribute is fully dependent on primary key.

Third Normal Form

Since there is no transitive functional dependency, therefore table is already in 3NF.

**NORMALIZATION OF STUDENT ATTENDANCE:**

|  |  |  |  |
| --- | --- | --- | --- |
| rollno | Date | First | second |

First Normal Form

In STUDENT\_ATTENDANCE, all the attributes are atomic and there cannot be duplicate rows. Hence, it is in 1NF.

Second Normal Form

This is already in 2NF since every non key attribute is fully dependent on primary key.

Third Normal Form

Since there is no transitive functional dependency, therefore table is already in 3NF.

**NORMALIZATION OF TEACHER ATTENDANCE:**

|  |  |  |  |
| --- | --- | --- | --- |
| rollno | Date | First | second |

First Normal Form

In TEACHER\_ATTENDANCE, all the attributes are atomic and there cannot be duplicate rows. Hence, it is in 1NF.

Second Normal Form

This is already in 2NF since every non key attribute is fully dependent on primary key.

Third Normal Form

Since there is no transitive functional dependency, therefore table is already in 3NF.

**NORMALIZATION OF SUBJECT:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Rollno | Subject1 | Subject2 | Subject3 | Subject4 | Subject5 |

First Normal Form

In SUBJECT, all the attributes are atomic and there cannot be duplicate rows. Hence, it is in 1NF.

Second Normal Form

This is already in 2NF since every non key attribute is fully dependent on primary key.

Third Normal Form

Since there is no transitive functional dependency, therefore table is already in 3NF.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Rollno | Marks1 | Marks2 | Marks3 | Marks4 | Marks5 |

**NORMALIZATION OF MARKS:**

First Normal Form

In MARKS, all the attributes are atomic and there cannot be duplicate rows. Hence, it is in 1NF.

Second Normal Form

This is already in 2NF since every non key attribute is fully dependent on primary key.

Third Normal Form

Since there is no transitive functional dependency, therefore table is already in 3NF

**CHAPTER-3**

**FRONT END DESIGN**

**FRONT END DESIGN**

**3.1 Screen Layout Design for Pages, Forms**

Java swing has been used for the design for the screen layout. Java swing is a Java library used to build Rich Internet Applications. The applications written using this library can run consistently across multiple platforms. The applications developed using Java swing can run on various devices such as Desktop Computers, Mobile Phones, TVs, Tablets, etc. To develop GUI Applications using Java programming language, the programmers rely on libraries such as Advanced Windowing Tool kit and Swing.

Following are some of the important features of Java swing :-

**Platform Independent:** It is platform independent, the swing components that are used to build the program are not platform specific. It can be used at any platform and anywhere.

**Lightweight:** Swing components are lightweight which helps in creating the UI lighter. Swings component allows it to plug into the operating system user interface framework that includes the mappings for screens or device and other user interactions like key press and mouse movements.

**Plugging:**It has a powerful component that can be extended to provide the support for the user interface that helps in good look and feel to the application. It refers to the highly modular-based architecture that allows it to plug into other customized implementations and framework for user interfaces. Its components are imported through a package called java.swing.

**Manageable:** It is easy to manage and configure. Its mechanism and composition pattern allows changing the settings at run time as well. The uniform changes can be provided to the user interface without doing any changes to application code.

**MVC:** They mainly follows the concept of MVC that is Model View Controller. With the help of this, we can do the changes in one component without impacting or touching other components. It is known as loosely coupled architecture as well.

**Customizable:** Swing controls can be easily customized. It can be changed and the visual appearance of the swing component application is independent of its internal representation.

**3.2 Connectivity to the Database from Front End**

The process of connecting the Application and the Database is divided into five routines.

These include:

* Loading and registration of JDBC driver
* Connecting to the DBMS
* Creating and executing a statement Data processing returned by the DBMS
* Terminating the connection with the DBMS

The Steps are as follows:

**3.2.1 Loading the JDBC Driver :**

The JDBC driver must be loaded before the J2EE component can connect to the DBMS. The **Class.forName()** method is used to load the JDBC driver. The developer must write a routine that loads the Pure Java/Thin driver called **java.sql.Driver**. The driver is loaded by calling the **Class.forName()** method and passing it in the name of the driver, as shown in the following code:

**Class.forName("com.mysql.cj.jdbc.Driver")**

**3.2.2 Connect to the DBMS :**

Once the driver is loaded, the J2EE component must connect to the DBMS using the **DriverManager.getConnection()** method. The **java.sql.DriverManager** class is the highest class in the java.sql hierarchy and is responsible for managing driver information. The DriverManager.getConnection() method passes the URL as a String object that contains the driver name and the name of the database is being accessed by

the J2EE component. The **DriverManager.getConnection()** method returns a Connection interface that is used throughout the process to reference the database. The **java.sql.Connection** interface in another member of the java.sql package that manages communication between the driver and the J2EE component. It is the java.sql.Connection interface that sends statements to the DBMS for processing.

String url="jdbc:mysql://localhost:3306/ums"; String user=”root”;

String password=””;

public Connection con; Statement st;

try {

con=DriverManager.getConnection(url,user,password);

}

**3.2.3 Create and Execute a SQL Statement :**

The next step, after the JDBC driver is loaded and connection is successfully made with a particular database managed by the DBMS is to send a SQL query to the DBMS for processing. A SQL query consist of a series of SQL commands that direct the DBMS to do something such as to return rows of data to the J2EE component.

The **Connect.createStatement()** method is used to create a **Statement** object. The Statement object is then used to execute a query and return a **ResultSet** object that contains the response from the DBMS which is usually one or more rows of information requested by the J2EE component.

Typically, the query is assigned to a **String** Object, which is passed to the Statement object’s **executeQuery()** method. Once the ResultSet is received from the DBMS, the **close()** method is called to terminate the statement. The code as follows:

Statement st;

ResultSet rs;

try {

String sql=”select \* from student”;

st=con.creataeStatement();

rs=st.executeQuery(sql);

con.close();

}

**3.2.4 Process Data Returned by the DBMS:**

The **java.sql.ResultSet** object is assigned the results received from the DBMS after the query is processed. The java.sql.ResultSet object consists of method used to interact with data that is returned by the DBMS to the J2EE component. The resultSet returned

by the DBMS is already assigned to the ResultSet object called Results. The first time that next() method of the ResultSet is called, the ResultSet pointer is positioned at first row in the ResulSet and return a Boolean value that if false indicates that no rows are present in the ResultSet.

**3.2.5 Terminating the connection with the DBMS :**

The connection to the DBMS is terminated using the **close()** method of connection object once the Java component is finished accessing the DBMS. The close() method throws an exception if a problem is encountered when disengaging the DBMS. Although closing the database connection automatically close the ResultSet, it is better to close the ResultSet explicitly before closing the connection. **con.close();**

# **CHAPTER-4**

**MAJOR MODULES**

**MAJOR MODULES**

The major modules of University Management system are:

The University Management system is developed using specified front end and back end tools. This system contains the following operations:

* Login
* Student UI
* Teacher UI
* Attendance UI
* Examination UI
* Fee Details

## 1. Admin Login:

* It facilitates a login for those who have authorization.
* The admin needs to use his/her login-id and password to enter into to the page where they can have their choice stored at a particular place.

**2. Student UI:**

* This module consists of details of all student who got admission in the University.
* Admin can add or remove or modify an student from the database.

**3. Teacher UI:**

* This module consists of details of all teacher in the University.
* Admin can add or remove or modify an teacher from the database

**4. Attendance UI:**

* It facilitates to mark the attendance of the student or teacher.
* It facilitates to download the attendance of the student or teacher.

**5. Examination UI:**

* It facilitates to enter the marks and subjects of the student.
* This module consists of all examinations records of the Student.

**6. Fee details:**

* It holds the information of Payment Status of student

# **CHAPTER-5**

**IMPLEMENTATION**

**IMPLEMENTATION**

5.**1 DATABASE CODE**

**TABLES CREATION**

CREATE database ums;

use ums;

CREATE TABLE student(

name varchar(20) DEFAULT NULL,

fathers\_name varchar(20) DEFAULT NULL,

age varchar(5) DEFAULT NULL,

dob varchar(20) DEFAULT NULL,

address varchar(30) DEFAULT NULL,

phone varchar(15) DEFAULT NULL,

email varchar(25) DEFAULT NULL,

class\_x varchar(10) DEFAULT NULL,

class\_xii varchar(10) DEFAULT NULL,

aadhar varchar(15) DEFAULT NULL,

rollno varchar(15) NOT NULL,

course varchar(10) DEFAULT NULL,

branch varchar(20) DEFAULT NULL,

sem varchar(5) DEFAULT NULL,

PRIMARY KEY(rollno));

CREATE TABLE teacher(

name varchar(20) DEFAULT NULL,

fathers\_name varchar(20) DEFAULT NULL,

age varchar(5) DEFAULT NULL,

dob varchar(20) DEFAULT NULL,

address varchar(30) DEFAULT NULL,

phone varchar(15) DEFAULT NULL,

email varchar(25) DEFAULT NULL,

aadhar varchar(15) DEFAULT NULL,

Education varchar(10) DEFAULT NULL,

emp\_id varchar(15) NOT NULL,

dept varchar(20) DEFAULT NULL,

subject varchar(20) ,

PRIMARY KEY (emp\_id));

CREATE TABLE fee(

rollno varchar(20) NOT NULL,

name varchar(25) DEFAULT NULL,

course varchar(10) DEFAULT NULL,

branch varchar(20) DEFAULT NULL,

Semester varchar(10) DEFAULT NULL,

fee\_paid varchar(15) DEFAULT NULL,

feeid varchar(10) NOT NULL,

PRIMARY KEY (rollno,feeid),

FOREIGN KEY (rollno) REFERENCES student (rollno),

FOREIGN KEY (name) REFERENCES student (name),

FOREIGN KEY (course) REFERENCES student (course),

FOREIGN KEY (branch) REFERENCES student (branch),

FOREIGN KEY (semester) REFERENCES student (semester));

CREATE TABLE attendance\_teacher(

emp\_id varchar(20) NOT NULL,

Date varchar(40) NOT NULL,

first varchar(10) DEFAULT NULL,

second varchar(10) DEFAULT NULL,

PRIMARY KEY (emp\_id,Date),

FOREIGN KEY (emp\_id) references teacher (emp\_id) ON DELETE CASCADE ON UPDATE CASCADE);

CREATE TABLE attendance\_student(

rollno varchar(20) NOT NULL,

Date varchar(30)NOT NULL,

first varchar(10) DEFAULT NULL,

second varchar(10) DEFAULT NULL,

PRIMARY KEY (rollno,Date),

FOREIGN KEY (rollno) REFERENCES student (rollno) ON DELETE CASCADE ON UPDATE CASCADE);

CREATE TABLE subject(

rollno varchar(15) NOT NULL,

subject1 varchar(20) DEFAULT NULL,

subject2 varchar(20) DEFAULT NULL,

subject3 varchar(20) DEFAULT NULL,

subject4 varchar(20) DEFAULT NULL,

subject5 varchar(20) DEFAULT NULL,

PRIMARY KEY (rollno),

FOREIGN KEY (rollno) REFERENCES student(rollno) ON DELETE CASCADE ON UPDATE CASCADE);

CREATE TABLE marks(

rollno varchar(15) NOT NULL,

marks1 varchar(20) DEFAULT NULL,

marks2 varchar(20) DEFAULT NULL,

marks3 varchar(20) DEFAULT NULL,

marks4 varchar(20) DEFAULT NULL,

marks5 varchar(20) DEFAULT NULL,

PRIMARY KEY (rollno),

FOREIGN KEY (rollno) REFERENCES student(rollno) ON DELETE CASCADE ON UPDATE CASCADE);

**STORED PROCEDURE:**

1. With the stored procedure we are going to determine details of the student.

CREATE PROCEDURE TEST()

SELECT \* FROM STUDENT;

2.With the stored procedure we are going to determine details of the teacher.

CREATE PROCEDURE TEST1()

SELECT \* FROM TEACHER;

## TRIGGERS

1. This trigger generates an exception if the Rollno of student was already taken.

CREATE TRIGGER TRIG BEFORE INSERT ON STUDENT FOR EACH ROW

BEGIN

IF(SELECT EXISTS(SELECT ROLLNO FROM STUDENT WHERE ROLLNO=NEW.ROLLNO))

THEN

SIGNAL SQLSTATE '45000' SET MESSAGE\_TEXT='ROLLNO ALREADY EXIST';

END IF;

END

2. This trigger generates an exception if the emp\_id of teacher was already taken.

CREATE TRIGGER TRIG1 BEFORE INSERT ON teacher FOR EACH ROW

BEGIN

IF(SELECT EXISTS(SELECT emp\_id FROM teacher WHERE emp\_id=NEW.emp\_id))

THEN

SIGNAL SQLSTATE '45000' SET MESSAGE\_TEXT='Employee Already Exist';

END IF;

END

5.2 JAVA CODE

CONNECTION TO DATABASE

package university.management.system;

import java.sql.\*;

public class conn{

Connection c;

Statement s;

public conn(){

try{

Class.forName("com.mysql.jdbc.Driver");

c =DriverManager.getConnection("jdbc:mysql:///ums","root","");

s =c.createStatement();

}catch(Exception e){

System.out.println(e);

}

}

}

MAIN PAGE

package university.management.system;

import java.awt.\*;

import java.awt.event.\*;

import javax.swing.\*;

public class Project extends JFrame implements ActionListener{

Project(){

super("University Management System");

setSize(1920,1030);

ImageIcon ic = new ImageIcon(ClassLoader.getSystemResource("university/management/system/icons/third.jpg"));

Image i3 = ic.getImage().getScaledInstance(1900, 950,Image.SCALE\_DEFAULT);

ImageIcon icc3 = new ImageIcon(i3);

JLabel l1 = new JLabel(icc3);

add(l1);

JMenuBar mb = new JMenuBar();

JMenu master = new JMenu("Master");

JMenuItem m1 = new JMenuItem("New Faculty");

JMenuItem m2 = new JMenuItem("New Student Admission");

master.setForeground(Color.BLUE);

m1.setFont(new Font("monospaced",Font.BOLD,16));

ImageIcon icon1 = new ImageIcon(ClassLoader.getSystemResource("university/management/system/icons/icon1.png"));

Image image1 = icon1.getImage().getScaledInstance(25, 25,Image.SCALE\_DEFAULT);

m1.setIcon(new ImageIcon(image1));

m1.setMnemonic('A');

m1.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK\_D, ActionEvent.CTRL\_MASK));

m1.setBackground(Color.WHITE);

m2.setFont(new Font("monospaced",Font.BOLD,16));

ImageIcon icon2 = new ImageIcon(ClassLoader.getSystemResource("university/management/system/icons/icon2.png"));

Image image2 = icon2.getImage().getScaledInstance(20, 20,Image.SCALE\_DEFAULT);

m2.setIcon(new ImageIcon(image2));

m2.setMnemonic('B');

m2.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK\_M, ActionEvent.CTRL\_MASK));

m2.setBackground(Color.WHITE);

m1.addActionListener(this);

m2.addActionListener(this);

JMenu user = new JMenu("Details");

JMenuItem u1 = new JMenuItem("Student Details");

JMenuItem u2 = new JMenuItem("Teacher Details");

user.setForeground(Color.RED);

u1.setFont(new Font("monospaced",Font.BOLD,16));

ImageIcon icon4 = new ImageIcon(ClassLoader.getSystemResource("university/management/system/icons/icon3.png"));

Image image4 = icon4.getImage().getScaledInstance(25, 25,Image.SCALE\_DEFAULT);

u1.setIcon(new ImageIcon(image4));

u1.setMnemonic('C');

u1.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK\_P, ActionEvent.CTRL\_MASK));

u1.setBackground(Color.WHITE);

u2.setFont(new Font("monospaced",Font.BOLD,16));

ImageIcon icon5 = new ImageIcon(ClassLoader.getSystemResource("university/management/system/icons/icon4.jpg"));

Image image5 = icon5.getImage().getScaledInstance(25, 25,Image.SCALE\_DEFAULT);

u2.setIcon(new ImageIcon(image5));

u2.setMnemonic('D');

u2.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK\_B, ActionEvent.CTRL\_MASK));

u2.setBackground(Color.WHITE);

u1.addActionListener(this);

u2.addActionListener(this);

JMenu attendance = new JMenu("Attendance");

JMenuItem a1 = new JMenuItem("Student Attendance");

JMenuItem a2 = new JMenuItem("Teacher Attendance");

attendance.setForeground(Color.BLUE);

a1.setFont(new Font("monospaced",Font.BOLD,16));

ImageIcon icon23 = new ImageIcon(ClassLoader.getSystemResource("university/management/system/icons/icon14.jpg"));

Image image24 = icon23.getImage().getScaledInstance(25, 25,Image.SCALE\_DEFAULT);

a1.setIcon(new ImageIcon(image24));

a1.setMnemonic('M');

a1.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK\_P, ActionEvent.CTRL\_MASK));

a1.setBackground(Color.WHITE);

attendance.add(a1);

a2.setFont(new Font("monospaced",Font.BOLD,16));

ImageIcon icon25 = new ImageIcon(ClassLoader.getSystemResource("university/management/system/icons/icon15.png"));

Image image26 = icon25.getImage().getScaledInstance(25, 25,Image.SCALE\_DEFAULT);

a2.setIcon(new ImageIcon(image26));

a2.setMnemonic('N');

a2.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK\_B, ActionEvent.CTRL\_MASK));

a2.setBackground(Color.WHITE);

attendance.add(a2);

a1.addActionListener(this);

a2.addActionListener(this);

JMenu attendance\_detail = new JMenu("Attendance Detail");

JMenuItem b1 = new JMenuItem("Student Attendance Detail");

JMenuItem b2 = new JMenuItem("Teacher Attendance Detail");

attendance\_detail.setForeground(Color.RED);

b1.setFont(new Font("monospaced",Font.BOLD,16));

ImageIcon icon27 = new ImageIcon(ClassLoader.getSystemResource("university/management/system/icons/icon15.png"));

Image image28 = icon27.getImage().getScaledInstance(25, 25,Image.SCALE\_DEFAULT);

b1.setIcon(new ImageIcon(image28));

b1.setMnemonic('O');

b1.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK\_P, ActionEvent.CTRL\_MASK));

b1.setBackground(Color.WHITE);

attendance\_detail.add(b1);

b2.setFont(new Font("monospaced",Font.BOLD,16));

ImageIcon icon29 = new ImageIcon(ClassLoader.getSystemResource("university/management/system/icons/icon14.jpg"));

Image image30 = icon29.getImage().getScaledInstance(25, 25,Image.SCALE\_DEFAULT);

b2.setIcon(new ImageIcon(image30));

b2.setMnemonic('P');

b2.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK\_B, ActionEvent.CTRL\_MASK)

b2.setBackground(Color.WHITE);

attendance\_detail.add(b2);

b1.addActionListener(this);

b2.addActionListener(this);

JMenu exam = new JMenu("Examination");

JMenuItem c1 = new JMenuItem("Examination Details");

JMenuItem c2 = new JMenuItem("Enter Marks");

exam.setForeground(Color.BLUE);

c1.setFont(new Font("monospaced",Font.BOLD,16));

ImageIcon icon30 = new ImageIcon(ClassLoader.getSystemResource("university/management/system/icons/icon16.png"));

Image image31 = icon30.getImage().getScaledInstance(25, 25,Image.SCALE\_DEFAULT);

c1.setIcon(new ImageIcon(image31));

c1.setMnemonic('Q');

c1.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK\_P, ActionEvent.CTRL\_MASK));

c1.setBackground(Color.WHITE);

exam.add(c1);

c2.setFont(new Font("monospaced",Font.BOLD,16));

ImageIcon icon32 = new ImageIcon(ClassLoader.getSystemResource("university/management/system/icons/icon17.png"));

Image image33 = icon32.getImage().getScaledInstance(25, 25,Image.SCALE\_DEFAULT);

c2.setIcon(new ImageIcon(image33));

c2.setMnemonic('R');

c2.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK\_B, ActionEvent.CTRL\_MASK));

c2.setBackground(Color.WHITE);

exam.add(c2);

c1.addActionListener(this);

c2.addActionListener(this);

JMenu report = new JMenu("Update Details");

JMenuItem r1 = new JMenuItem("Update Students");

JMenuItem r2 = new JMenuItem("Update Teachers");

report.setForeground(Color.RED);

r1.setFont(new Font("monospaced",Font.BOLD,16));

ImageIcon icon7 = new ImageIcon(ClassLoader.getSystemResource("university/management/system/icons/icon5.png"));

Image image7 = icon7.getImage().getScaledInstance(25, 25,Image.SCALE\_DEFAULT);

r1.setIcon(new ImageIcon(image7));

r1.setMnemonic('E');

r1.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK\_R, ActionEvent.CTRL\_MASK));

r1.setBackground(Color.WHITE);

r1.addActionListener(this);

r2.setFont(new Font("monospaced",Font.BOLD,16));

ImageIcon iconn = new ImageIcon(ClassLoader.getSystemResource("university/management/system/icons/icon6.png"));

Image imagee = iconn.getImage().getScaledInstance(25, 25,Image.SCALE\_DEFAULT);

r2.setIcon(new ImageIcon(imagee));

r2.setMnemonic('F');

r2.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK\_R, ActionEvent.CTRL\_MASK));

r2.setBackground(Color.WHITE);

r2.addActionListener(this);

JMenu fee = new JMenu("Fee Details");

JMenuItem s1 = new JMenuItem("Fee Structure");

JMenuItem s2 = new JMenuItem("Student Fee Form");

fee.setForeground(Color.BLUE);

s1.setFont(new Font("monospaced",Font.BOLD,16));

ImageIcon icon14 = new ImageIcon(ClassLoader.getSystemResource("university/management/system/icons/icon7.png"));

Image image15 = icon14.getImage().getScaledInstance(20, 20,Image.SCALE\_DEFAULT);

s1.setIcon(new ImageIcon(image15));

s1.setMnemonic('G');

s1.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK\_R, ActionEvent.CTRL\_MASK));

s1.setBackground(Color.WHITE);

s1.addActionListener(this);

s2.setFont(new Font("monospaced",Font.BOLD,16));

ImageIcon icon16 = new ImageIcon(ClassLoader.getSystemResource("university/management/system/icons/icon8.png"));

Image image17 = icon16.getImage().getScaledInstance(20, 20,Image.SCALE\_DEFAULT);

s2.setIcon(new ImageIcon(image17));

s2.setMnemonic('H');

s2.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK\_R, ActionEvent.CTRL\_MASK));

s2.setBackground(Color.WHITE);

s2.addActionListener(this);

JMenu utility = new JMenu("Utility");

JMenuItem ut1 = new JMenuItem("Notepad");

JMenuItem ut2 = new JMenuItem("Calculator");

JMenuItem ut3 = new JMenuItem("Web Browser");

utility.setForeground(Color.RED);

ut1.setFont(new Font("monospaced",Font.BOLD,16));

ImageIcon icon18 = new ImageIcon(ClassLoader.getSystemResource("university/management/system/icons/icon9.png"));

Image image19 = icon18.getImage().getScaledInstance(25, 25,Image.SCALE\_DEFAULT);

ut1.setIcon(new ImageIcon(image19));

ut1.setMnemonic('I');

ut1.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK\_C, ActionEvent.CTRL\_MASK));

ut1.setBackground(Color.WHITE);

ut2.setFont(new Font("monospaced",Font.BOLD,16));

ImageIcon icon20 = new ImageIcon(ClassLoader.getSystemResource("university/management/system/icons/icon10.png"));

Image image21 = icon20.getImage().getScaledInstance(25, 25,Image.SCALE\_DEFAULT);

ut2.setIcon(new ImageIcon(image21));

ut2.setMnemonic('J');

ut2.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK\_X, ActionEvent.CTRL\_MASK));

ut2.setBackground(Color.WHITE);

ut3.setFont(new Font("monospaced",Font.BOLD,16));

ImageIcon icon10 = new ImageIcon(ClassLoader.getSystemResource("university/management/system/icons/icon11.png"));

Image image10 = icon10.getImage().getScaledInstance(25, 25,Image.SCALE\_DEFAULT);

ut3.setIcon(new ImageIcon(image10));

ut3.setMnemonic('K');

ut3.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK\_W, ActionEvent.CTRL\_MASK));

ut3.setBackground(Color.WHITE);

ut1.addActionListener(this);

ut2.addActionListener(this);

ut3.addActionListener(this);

JMenu about = new JMenu("About");

JMenuItem aboutus = new JMenuItem("About Us");

about.setForeground(Color.BLUE);

aboutus.setFont(new Font("monospaced",Font.BOLD,16));

ImageIcon icon21 = new ImageIcon(ClassLoader.getSystemResource("university/management/system/icons/icon13.jpg"));

Image image22 = icon21.getImage().getScaledInstance(25, 25,Image.SCALE\_DEFAULT);

aboutus.setIcon(new ImageIcon(image22));

aboutus.setMnemonic('L');

aboutus.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK\_C, ActionEvent.CTRL\_MASK));

aboutus.setBackground(Color.WHITE);

about.add(aboutus);

aboutus.addActionListener(this);

JMenu exit = new JMenu("Exit");

JMenuItem ex = new JMenuItem("Log Out");

exit.setForeground(Color.RED);

ex.setFont(new Font("monospaced",Font.BOLD,16));

ImageIcon icon11 = new ImageIcon(ClassLoader.getSystemResource("university/management/system/icons/icon12.png"));

Image image11 = icon11.getImage().getScaledInstance(25, 25,Image.SCALE\_DEFAULT);

ex.setIcon(new ImageIcon(image11));

ex.setMnemonic('Z');

ex.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK\_Z, ActionEvent.CTRL\_MASK));

ex.setBackground(Color.WHITE);

ex.addActionListener(this);

master.add(m1);

master.add(m2);

user.add(u1);

user.add(u2);

report.add(r1);

report.add(r2);

fee.add(s1);

fee.add(s2);

utility.add(ut1);

utility.add(ut2);

utility.add(ut3);

exit.add(ex);

mb.add(master);

mb.add(user);

mb.add(attendance);

mb.add(attendance\_detail);

mb.add(exam);

mb.add(report);

mb.add(fee);

mb.add(utility);

mb.add(about);

mb.add(exit);

setJMenuBar(mb);

setFont(new Font("Senserif",Font.BOLD,16));

setLayout(new FlowLayout());

setVisible(false);

}

public void actionPerformed(ActionEvent ae){

String msg = ae.getActionCommand();

if(msg.equals("New Student Admission")){

new AddStudent().f.setVisible(true);

}else if(msg.equals("New Faculty")){

new AddTeacher().f.setVisible(true);

}else if(msg.equals("Student Details")){

new StudentDetails().setVisible(true);

}else if(msg.equals("Teacher Details")){

new TeacherDetails().setVisible(true);

}

else if(msg.equals("Update Students")){

new UpdateStudent().f.setVisible(true);

}

else if(msg.equals("Update Teachers")){

new UpdateTeacher().f.setVisible(true);

}

else if(msg.equals("Fee Structure")){

new FeeStructure().setVisible(true);

}

else if(msg.equals("Student Fee Form")){

new StudentFeeForm().setVisible(true);

}

else if(msg.equals("Notepad")){

try{

Runtime.getRuntime().exec("notepad.exe");

}catch(Exception e){ }

}else if(msg.equals("Calculator")){

try{

Runtime.getRuntime().exec("calc.exe");

}catch(Exception e){ }

}else if(msg.equals("Web Browser")){

try{

Runtime.getRuntime().exec("C:\\Program Files\\Google\\Chrome\\Application\\chrome.exe");

}catch(Exception e){ }

}else if(msg.equals("Log Out")){

System.exit(0);

}else if(msg.equals("About Us")){

new AboutUs().setVisible(true);

}else if(msg.equals("Student Attendance")){

new StudentAttendance().setVisible(true);

}else if(msg.equals("Teacher Attendance")){

new TeacherAttendance().setVisible(true);

}else if(msg.equals("Student Attendance Detail")){

new StudentAttendanceDetail().setVisible(true);

}else if(msg.equals("Teacher Attendance Detail")){

new TeacherAttendanceDetail().setVisible(true);

}else if(msg.equals("Examination Details")){

new ExaminationDetails().setVisible(true);

}else if(msg.equals("Enter Marks")){

new EnterMarks().setVisible(true);

}

}

public static void main(String[] args){

new Project().setVisible(true);

}

}

LOGIN PAGE

package university.management.system;

import java.awt.\*;

import javax.swing.\*;

import java.awt.event.\*;

import static java.lang.System.exit;

import java.sql.\*;

public class Login extends JFrame implements ActionListener{

JFrame f;

JLabel l1,l2;

JTextField t1;

JPasswordField t2;

JButton b1,b2;

Login(){

super("Login");

setBackground(Color.white);

setLayout(null);

l1 = new JLabel("Username");

l1.setBounds(40,20,100,30);

add(l1);

l2 = new JLabel("Password");

l2.setBounds(40,70,100,30);

add(l2);

t1=new JTextField();

t1.setBounds(150,20,150,30);

add(t1);

t2=new JPasswordField();

t2.setBounds(150,70,150,30);

add(t2);

ImageIcon i1 = new ImageIcon(ClassLoader.getSystemResource("university/management/system/icons/second.jpg"));

Image i2 = i1.getImage().getScaledInstance(200,200,Image.SCALE\_DEFAULT);

ImageIcon i3 = new ImageIcon(i2);

JLabel l3 = new JLabel(i3);

l3.setBounds(350,20,150,150);

add(l3);

b1 = new JButton("Login");

b1.setBounds(40,140,120,30);

b1.setFont(new Font("serif",Font.BOLD,15));

b1.addActionListener(this);

b1.setBackground(Color.BLACK);

b1.setForeground(Color.WHITE);

add(b1);

b2=new JButton("Cancel");

b2.setBounds(180,140,120,30);

b2.setFont(new Font("serif",Font.BOLD,15));

b2.setBackground(Color.BLACK);

b2.setForeground(Color.WHITE);

add(b2);

b2.addActionListener(this);

getContentPane().setBackground(Color.WHITE);

setVisible(true);

setSize(600,300);

setLocation(500,300);

}

public void actionPerformed(ActionEvent ae){

try{

conn c1 = new conn();

String u = t1.getText();

String v = t2.getText();

String q = "select \* from login where username='"+u+"' and password='"+v+"'";

ResultSet rs = c1.s.executeQuery(q);

/\* String q =c1.prepareCall("{call new\_procedure("+u+","+v+")}");

ResultSet rs = c1.s.executeQuery(q);\*/

if(rs.next()){

new Project().setVisible(true);

setVisible(false);

}else{

JOptionPane.showMessageDialog(null, "Username or password dosenot match");

setVisible(false);

}

}catch(Exception e){

e.printStackTrace();

}

}

public static void main(String[] arg){

Login l = new Login();

}

}

ADD STUDENT PAGE

package university.management.system;

import java.awt.\*;

import java.awt.event.\*;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.util.Random;

import javax.swing.\*;

class AddStudent implements ActionListener{

JFrame f;

JLabel id,id1,id2,id3,id4,id5,id6,id7,id8,id9,id10,id11,id12,id15,id16,id17,lab,lab1,lab2,lab3;

JTextField t,t1,t2,t3,t4,t5,t6,t7,t8,t9,t10,t11,t12,t13;

JButton b,b1,b2,b3;

JComboBox c1,c2,c3;

Random ran = new Random();

long first4 = (ran.nextLong() % 9000L) + 1000L;

long first = Math.abs(first4);

public AddStudent(){

f = new JFrame("Add Student");

f.setBackground(Color.white);

f.setLayout(null);

id15=new JLabel();

id15.setBounds(0,0,900,700);

id15.setLayout(null);

ImageIcon img = new ImageIcon(ClassLoader.getSystemResource("university/management/system/icons/fourth.jpg"))

Image i3 = img.getImage().getScaledInstance(1220, 700,Image.SCALE\_DEFAULT);

ImageIcon icc3 = new ImageIcon(i3);

id15.setIcon(icc3);

id8 = new JLabel("New Student Details");

id8.setBounds(320,30,500,50);

id8.setFont(new Font("serif",Font.ITALIC,25));

id8.setForeground(Color.black);

id15.add(id8);

f.add(id15);

id1 = new JLabel("Name");

id1.setBounds(50,150,100,30);

id1.setFont(new Font("serif",Font.BOLD,20));

id15.add(id1);

t1=new JTextField();

t1.setBounds(200,150,150,30);

id15.add(t1);

id2 = new JLabel("Father's Name");

id2.setBounds(400,150,200,30);

id2.setFont(new Font("serif",Font.BOLD,20));

id15.add(id2);

t2=new JTextField();

t2.setBounds(600,150,150,30);

id15.add(t2);

id3= new JLabel("Age");

id3.setBounds(50,200,100,30);

id3.setFont(new Font("serif",Font.BOLD,20));

id15.add(id3);

t3=new JTextField();

t3.setBounds(200,200,150,30);

id15.add(t3);

id4= new JLabel("DOB (dd/mm/yyyy)");

id4.setBounds(400,200,200,30);

id4.setFont(new Font("serif",Font.BOLD,20));

id15.add(id4);

t4=new JTextField();

t4.setBounds(600,200,150,30);

id15.add(t4);

id5= new JLabel("Address");

id5.setBounds(50,250,100,30);

id5.setFont(new Font("serif",Font.BOLD,20));

id15.add(id5);

t5=new JTextField();

t5.setBounds(200,250,150,30);

id15.add(t5);

id6= new JLabel("Phone");

id6.setBounds(400,250,100,30);

id6.setFont(new Font("serif",Font.BOLD,20));

id15.add(id6);

t6=new JTextField();

t6.setBounds(600,250,150,30);

id15.add(t6);

id7= new JLabel("Email Id");

id7.setBounds(50,300,100,30);

id7.setFont(new Font("serif",Font.BOLD,20));

id15.add(id7);

t7=new JTextField();

t7.setBounds(200,300,150,30);

id15.add(t7);

id9= new JLabel("Class X(%)");

id9.setBounds(400,300,130,30);

id9.setFont(new Font("serif",Font.BOLD,20));

id15.add(id9);

t8=new JTextField();

t8.setBounds(600,300,150,30);

id15.add(t8);

id10= new JLabel("Class XII(%)");

id10.setBounds(50,350,130,30);

id10.setFont(new Font("serif",Font.BOLD,20));

id15.add(id10);

t9=new JTextField();

t9.setBounds(200,350,150,30);

id15.add(t9);

id11= new JLabel("Aadhar No");

id11.setBounds(400,350,100,30);

id11.setFont(new Font("serif",Font.BOLD,20));

id15.add(id11);

t10=new JTextField();

t10.setBounds(600,350,150,30);

id15.add(t10);

id12= new JLabel("Roll No");

id12.setBounds(50,400,150,30);

id12.setFont(new Font("serif",Font.BOLD,20));

id15.add(id12);

t11=new JTextField();

t11.setBounds(200,400,150,30);

t11.setText("1533"+first);

id15.add(t11);

lab=new JLabel("Course");

lab.setBounds(400,400,150,30);

lab.setFont(new Font("serif",Font.BOLD,20));

id15.add(lab);

String course[] = {"B.Tech","BBA","BCA","Bsc","Msc","MBA","MCA","BA","BCom"};

c1 = new JComboBox(course);

c1.setBackground(Color.WHITE);

c1.setBounds(600,400,150,30);

id15.add(c1);

lab2=new JLabel("Branch");

lab2.setBounds(50,450,150,30);

lab2.setFont(new Font("serif",Font.BOLD,20));

id15.add(lab2);

String branch[] = {"","Computer Science","Electronics","Electrical","Mechanical","Civil"};

c2 = new JComboBox(branch);

c2.setBackground(Color.WHITE);

c2.setBounds(200,450,150,30);

id15.add(c2);

lab3=new JLabel("Semester");

lab3.setBounds(400,450,150,30);

lab3.setFont(new Font("serif",Font.BOLD,20));

id15.add(lab3);

String semester[] = {"","1","2","3","4","5","6","7","8"};

c3 = new JComboBox(semester);

c3.setBackground(Color.WHITE);

c3.setBounds(600,450,150,30);

id15.add(c3);

b = new JButton("Submit");

b.setBackground(Color.BLACK);

b.setForeground(Color.WHITE);

b.setBounds(250,550,150,40);

id15.add(b);

b1=new JButton("Cancel");

b1.setBackground(Color.BLACK);

b1.setForeground(Color.WHITE);

b1.setBounds(450,550,150,40);

id15.add(b1);

b.addActionListener(this);

b1.addActionListener(this);

f.setVisible(true);

f.setSize(900,700);

f.setLocation(400,150);

}

public void actionPerformed(ActionEvent ae){

String a = t1.getText();

String bb = t2.getText();

String c = t3.getText();

String d = t4.getText();

String e = t5.getText();

String ff = t6.getText();

String g = t7.getText();

String h = t8.getText();

String i = t9.getText();

String j = t10.getText();

String k = t11.getText();

String l = (String)c1.getSelectedItem();

String m = (String)c2.getSelectedItem();

String n = (String)c3.getSelectedItem();

if(ae.getSource() == b){

try{

conn cc = new conn();

String q = "insert into student values('"+a+"','"+bb+"','"+c+"','"+d+"','"+e+"','"+ff+"','"+g+"','"+h+"','"+i+"','"+j+"','"+k+"','"+l+"','"+m+"','"+n+"')";

cc.s.executeUpdate(q);

JOptionPane.showMessageDialog(null,"Student Details Inserted Successfully");

f.setVisible(false);

}/\*catch(Exception ee){

System.out.println("The error is:"+ee);

JOptionPane.showMessageDialog(null,"Roll No Already Exist");\*/

catch(SQLException sqlex)

{

JOptionPane.showMessageDialog(null,sqlex.getMessage());

}

}else if(ae.getSource() == b1){

f.setVisible(false);

new Project().setVisible(true);

}

}

public static void main(String[ ] arg){

new AddStudent().f.setVisible(true);

}

# **CHAPTER-6**

**SNAPSHOTS**

**SNAPSHOTS**

**6.1 SCREENSHOTS**

1. **LOGIN**

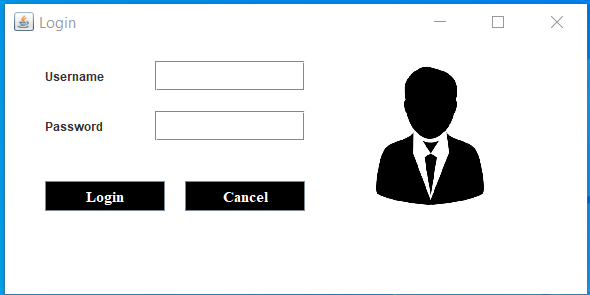


Figure 6.1.1 Login

1. **ADD STUDENT**

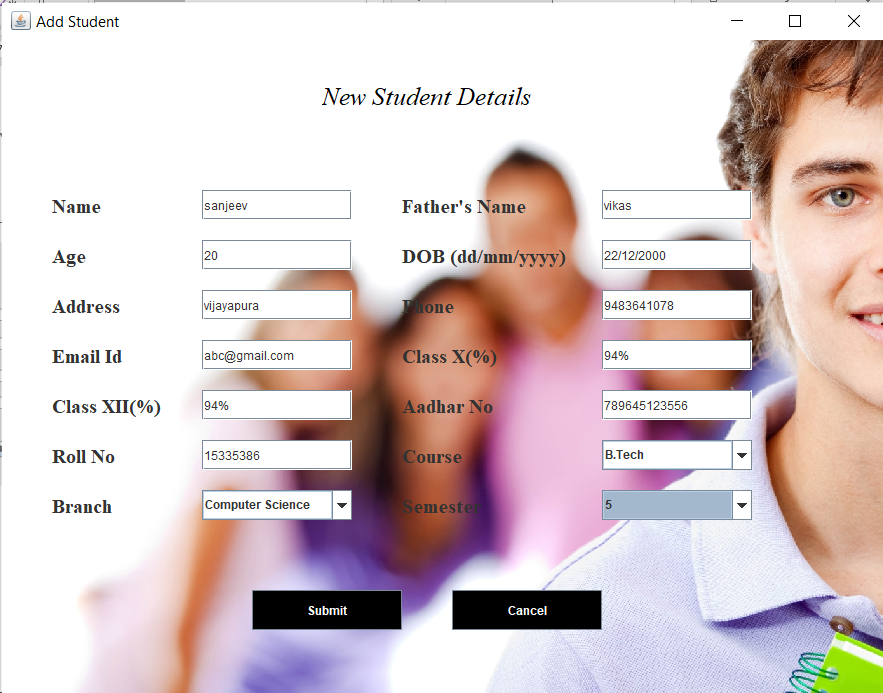
****

Figure 6.1.2 Add student

1. **STUDENT DETAILS UI**

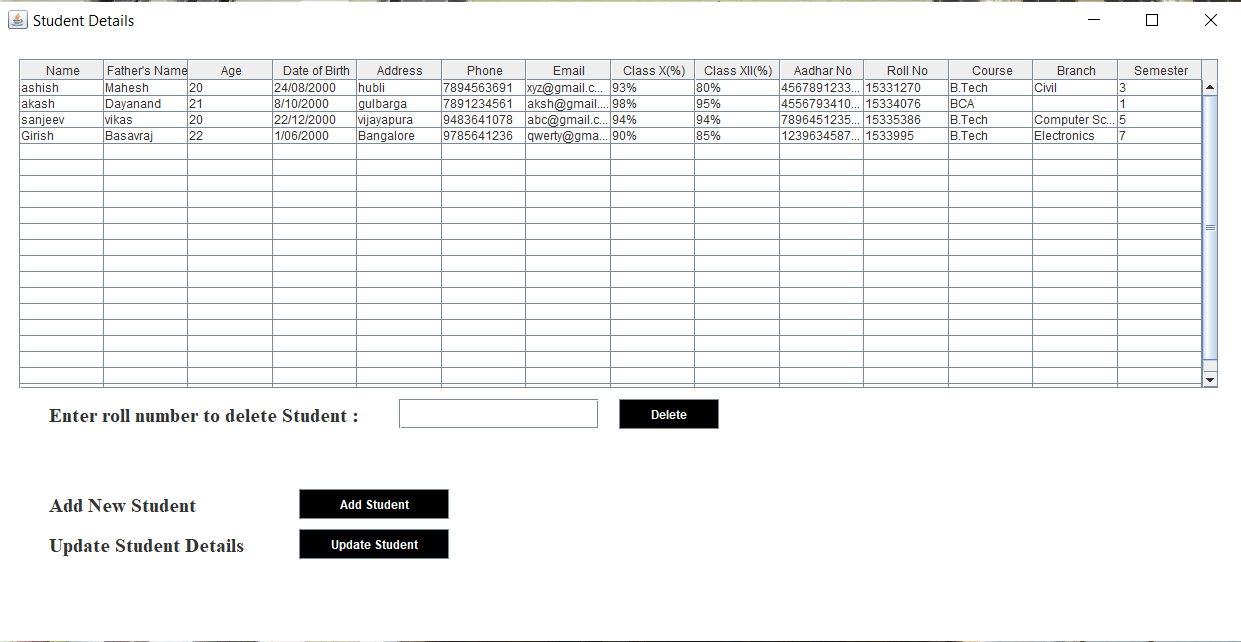
****

Figure 6.1.3 Student details

1. **STUDENT ATTENDANCE UI**

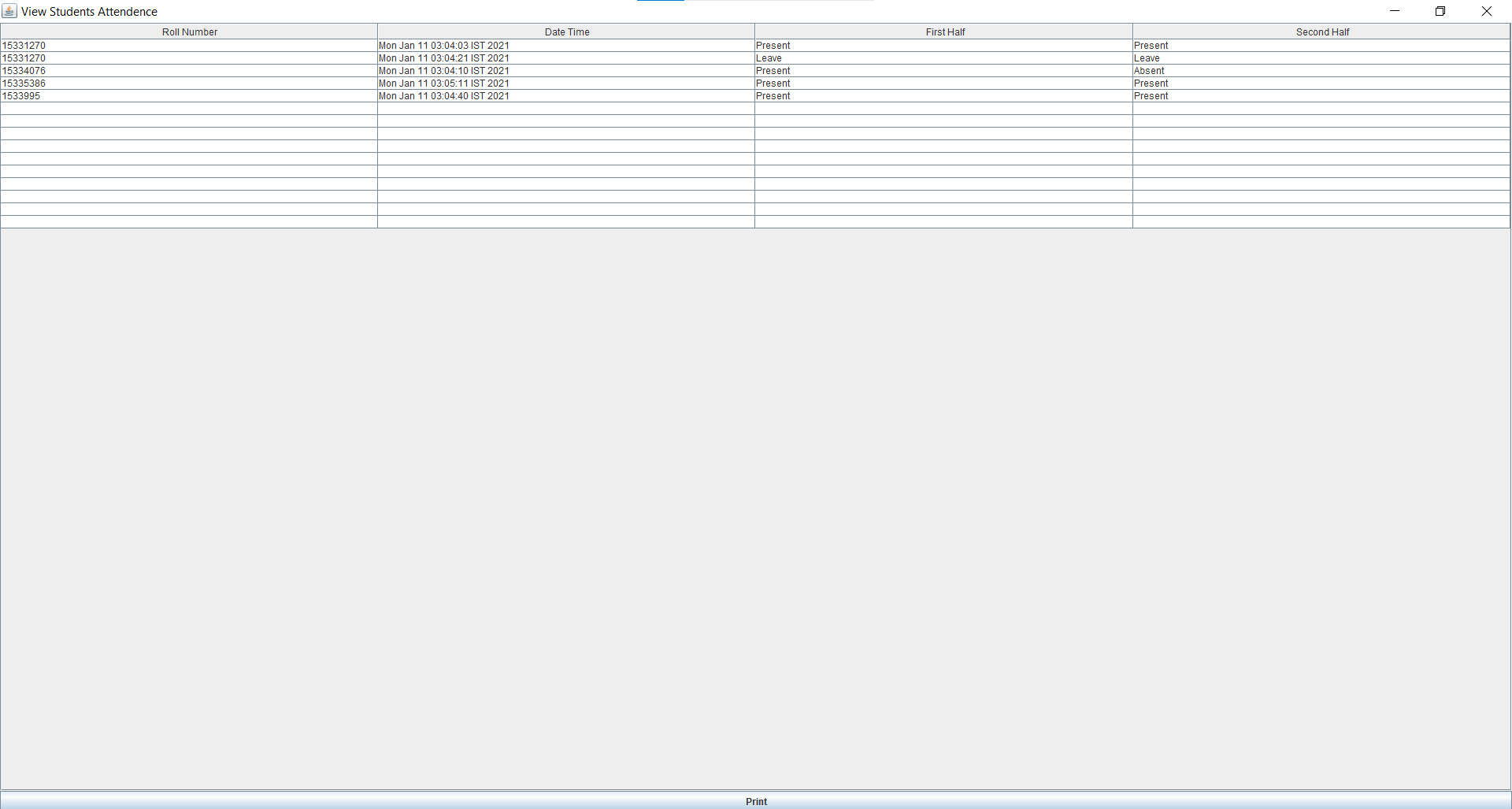
****

Figure 6.1.4 Student attendance

1. **EXAMINATION UI**

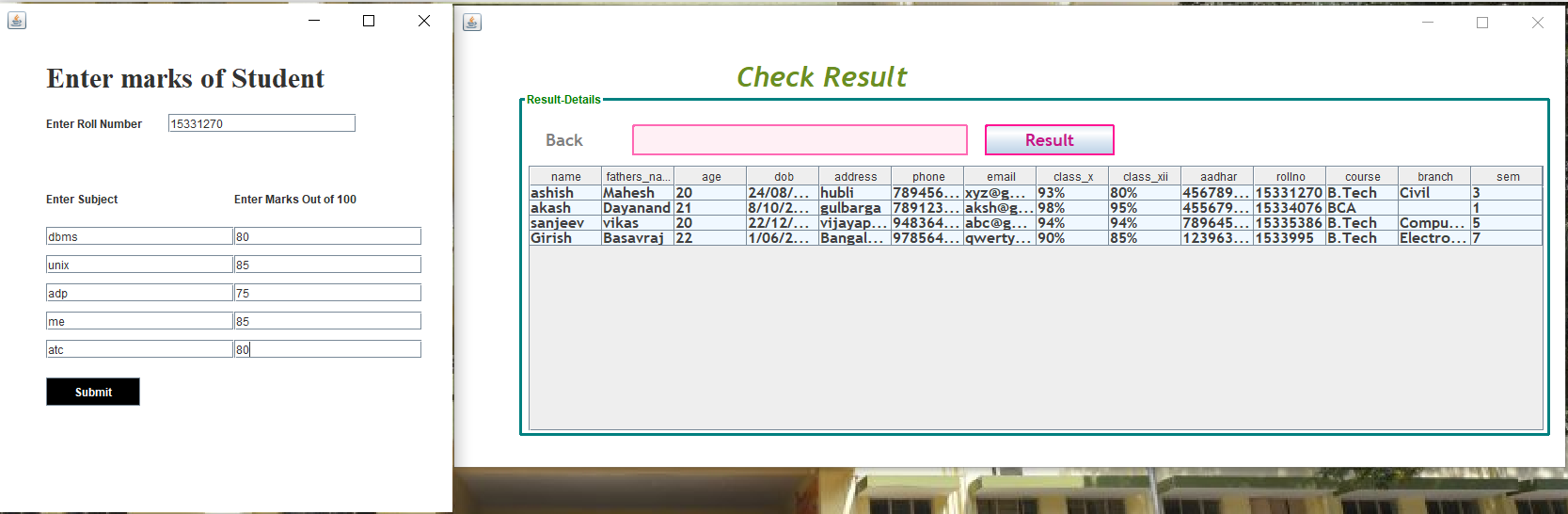
****

Figure 6.1.5 Examination

1. **FACULTY UI**

****

Figure 6.1.6 Faculty

1. **FACULTY DETAILS UI**

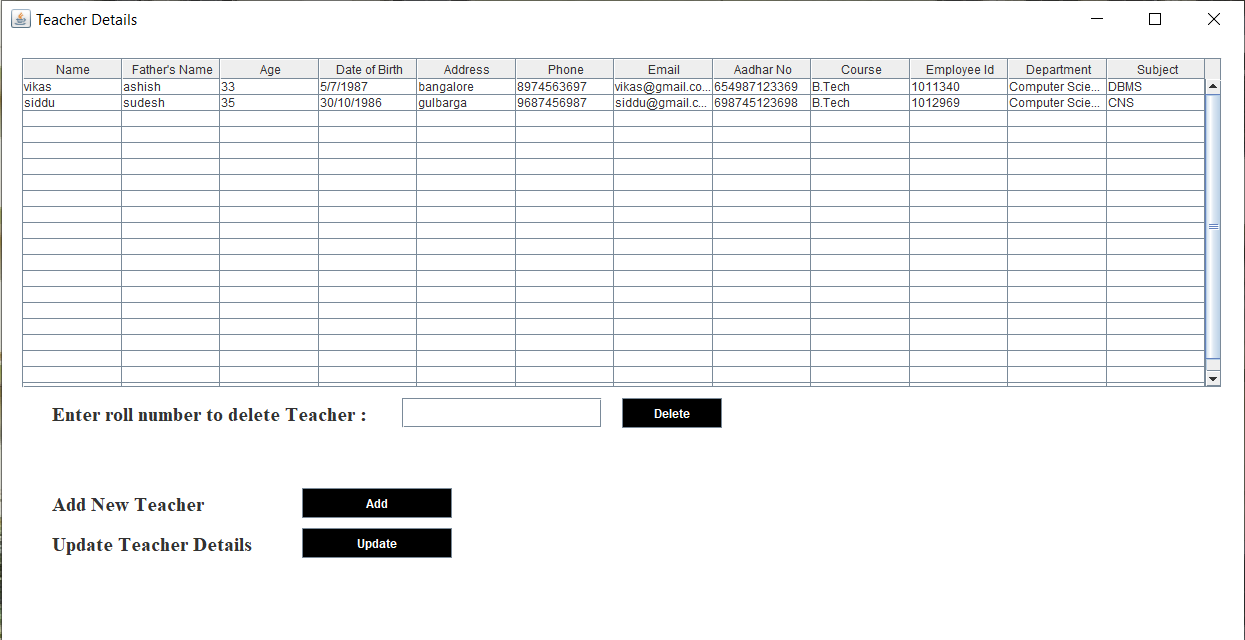
****

Figure 6.1.7 Faculty Details

1. **UPDATE FACULTY UI**

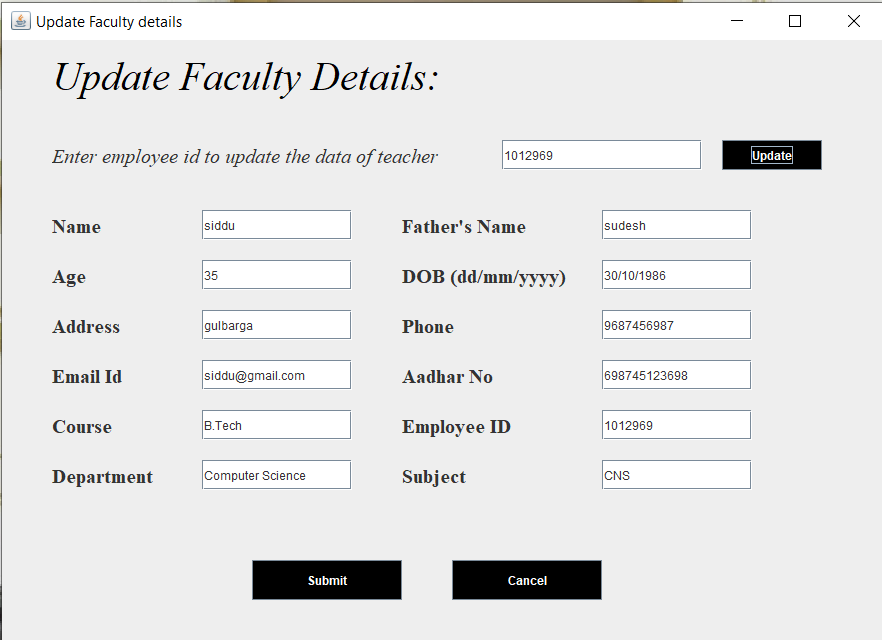
****

Figure 6.1.8Update faculty

1. **UPDATE STUDENT UI**

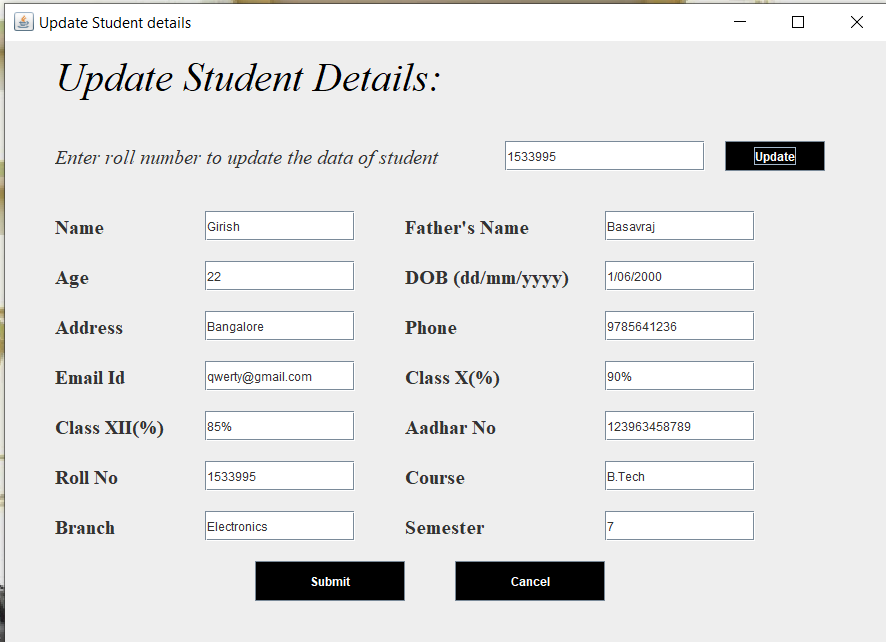
****

Figure 6.1.9 Update student

1. **FEE UI**

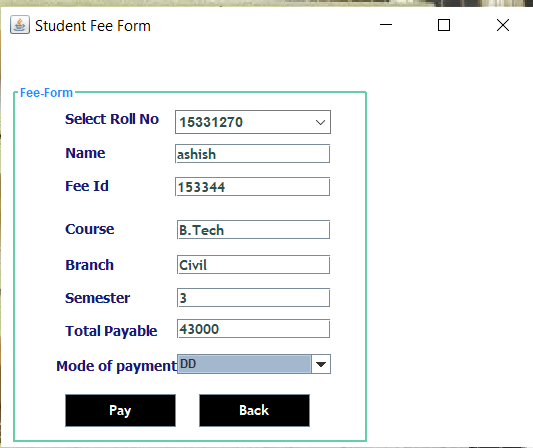
****

Figure 6.1.10 Fee

1. **FEE STRUCTURE UI**

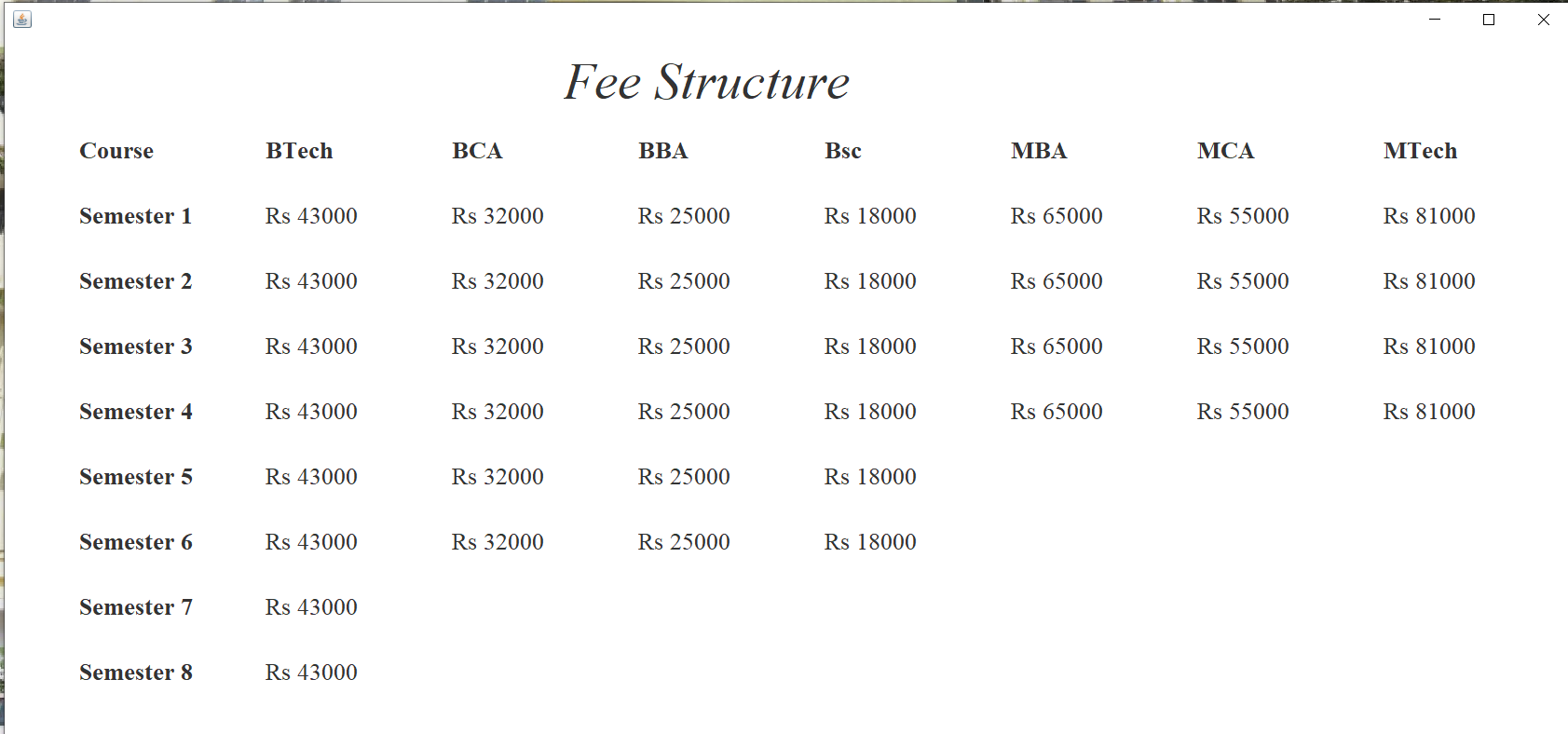
****

Figure 6.1.11 fee structure

1. **ABOUT US UI**

****

Figure 6.1.12 About us

**6.2 TEST CASES**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test id** | **Test name** | **Input** | **Output** | **Results** |
| 1 | Login | Correct password and username | Login | Success |
| 2 | Denial | In-Correct password and username | Access Denied | Failure |
| 3 | Add Student | No primary key conflict | Student Added | Success |
| 4 | Error | Primary key Conflict | Error adding data to database | Failure |
| 5 | Add Teacher | No primary keyconflict | Teacher Added | Success |
| 6 | Error | Primary keyConflict | Error adding data to database | Failure |
| 7 | Delete student | RollNo | Student deleted | Success |
| 8 | Delete Faculty | Empid | Faculty deleted | Success |
| 9 | Logout | If logged in | Logged out | Success |

**CHAPTER-7**

**APPLICATIONS**

**APPLICATIONS**

**7.1 APPLICATIONS**

* This Desktop application can be used by educational institutions like colleges and schools to maintain the data of students and faculty.

# **CHAPTER-8**

**CONCLUSION**

**CONCLUSION**

**8.1 CONCLUSION**

* Thus, an integrated database system has been created to keep record of students and the faculty could track the development of the students and make sure everyone is one the right track with their tests and attendance .
* This project has been created using MySQL as the back end tool using WampServer and Java with swing as the front end tool using Netbeans and JForm as the environments.
* It was wonderful to get an opportunity to work on this project and I learnt a lot from it.
* I learnt how real-world software engineering projects are made and the steps involved in it. Through this I came to know how professionals create software.
* I enjoyed each and every bit of work in order to complete this project successfully.