VISVESVARAYA TECHNOLOGICAL UNIVERSITY

BELGAUM-590014

A DBMS Mini-Project Report

On

"STUDENT ATTENDANCE MANAGEMENT SYSTEM"

A Mini-project report submitted in partial fulfillment of the requirements for the award of the degree of ***Bachelor of Engineering in Computer Science and Engineering*** *of Visvesvaraya Technological University, Belgaum.*

Submitted by:
SURYA GANGARAJ K (1DT15CS115)
AND
RAHUL M (1DT15CS091)

Under the Guidance of:
Mr. RAGHU.M.T

(Asst. Prof. Dept of CSE)

Department of Computer Science and Engineering
DAYANADA SAGAR ACADEMY OF TECHNOLOGY AND MANAGAEMENT
Kanakpura Road,Udayapura, Bangalore
2017-2018

DAYANADA SAGAR ACADEMY OF TECHNOLOGY AND MANAGAEMENT,

Kanakpura Road,Udayapura, Bangalore

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

CERTIFICATE

This is to certify that the Mini-Project on Database Management System (DBMS) entitled **STUDENT ATTENDANCE MANAGEMENT SYSTEM** has been successfully carried out by **SURYA GANGARAJ K(1DT15CS115) and RAHUL M(1DT15CS091)** a bona fide students of **Dayananda sagar academy of technology and management** in partial fulfillment of the requirements for the award of degree in **Bachelor of Engineering in Computer Science and Engineering** of **Visvesvaraya Technological University, Belgaum** during academic year 2017-2018. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the report deposited in the departmental library. The mini project report has been approved as it satisfies the academic requirements in respect of project work for the said degree.

GUIDES:

Mr. RAGHU.M.T
(Asst. Prof. Dept of CSE)

Mr. Manjunath R.D
(Asst. Prof. Dept of CSE)

Dr. C. NANDINI
(Vice Principal & HOD, Dept. of CSE)

Examiners: **Signature with Date**

1:

2:

ACKNOWLEDGEMENT

It gives us immense pleasure to present before you our project titled **STUDENT ATTENDANCE MANAGEMENT SYSTEM USING JFRAMES.** The joy and satisfaction that accompany the successful completion of any task would be incomplete without the mention of those who made it possible. We are glad to express our gratitude towards our prestigious institution **DAYANANDA SAGAR ACADEMY OF TECHNOLOGY AND MANAGEMENT** for providing us with utmost knowledge, encouragement and the maximum facilities in undertaking this project.

We wish to express a sincere thanks to our respected principal **Dr. B. R. Lakshmikantha** for all their support.

We express our deepest gratitude and special thanks to **Dr.C.Nandini, Prof & H.O.D, Dept. Of Computer Science Engineering**, for all her guidance and encouragement.

We sincerely acknowledge the guidance and constant encouragement of our mini- project guides, **Assistant Prof. Mr.Raghu. M. T** and **Assistant Prof. Mr.Manjunath D.R**

RAHUL M(1DT15CS091)
AND
SURYA GANGARAJ K (1DT15CS115)

ABSTRACT

Our project Student Attendance Management system is designed to automate and simplify the work done by the operations team of a College Staff. It includes maintenance of Staff details,Attendance details, Subject details. Our software has the facility to add new record, update existing record and delete an existing record and also view the details of each of these. It includes the facility to know the statistics of Staff, Subjects and Attendance .The Student Attendance Management System can be entered using a username and password. It is accessible either by an administrator ,staff or student . Administrator has access to all the functions of the adding student as well as staff . Student can only view his attendance list and his profile .Our Project interface is very user-friendly and simplifies the manual operation of a Student Attendance management.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1. Background

Considering the volumes of data that needs to be tracked in a Student Attendance, it would be very difficult to manage the accuracy and quality of data manually. It would be almost impossible to get the details required in case of manual maintenance of data. The Student Attendance Management System simplifies the manual work and allows smooth administration of the operations of a Student Attendance.

2. Problem Definition

This project is aimed to reduce the manual work involved in data maintenance in a Student Attendance and automates the Student Attendance Management System. This project is developed mainly to simplify the manual work and allows smooth administration of the operations of a Student Attendance. The purpose of the project is to computerize the administrative operations of a Student Attendance and to develop software which is user friendly, simple, fast, and cost – effective. It deals with the collection of Staff, Subject, Attendance and Student information etc. Traditionally, it was done manually. The main function of the system is to enter and store Staff, Subject and Attendance details and retrieve these details as and when required, and also to manipulate these details meaningfully.

3. Motivation

Manual System: The system is very time consuming and lazy. This system is more prone to errors and sometimes the approaches to various problems are unstructured.

Technical System: With the invent of latest technology, we should update our systems which are very fast, accurate, user-friendly and reliable.

4. Objective

Main goal of this project is to simplify the manual operation of a Student Attendance with the following advantages:

1. Faster System
2. Accuracy
3. Reliability
4. Cost Effective
5. User Friendly
6. Immediate access to the data and statistics

5. Scope of the project

The project provides a very simple application which simplifies the manual work done by the operations team of Student Attendance. This application saves the data of staff, Subjects and Attendance in the database. Allows users to enter the details, update / delete the existing details. Our project allows users to view the data stored in the database and to see the statistics.

CHAPTER 2

REQUIREMENTS

The requirements can be broken down into 2 major categories namely hardware and software requirements. The former specifies the minimal hardware facilities expected in a system in which the project has to be run. The latter specifies the essential software needed to build and run the project.

2.1 Hardware Requirements

The Hardware requirements are very minimal and the program can be run on most of the machines.

- Processor - Intel 486/Pentium processor or better
- Processor Speed - 500 MHz or above
- Hard Disk - 20GB(approx.)
- RAM - 64MB or above
- Storage Space - Approx. 4MB

2.2 Software Requirements

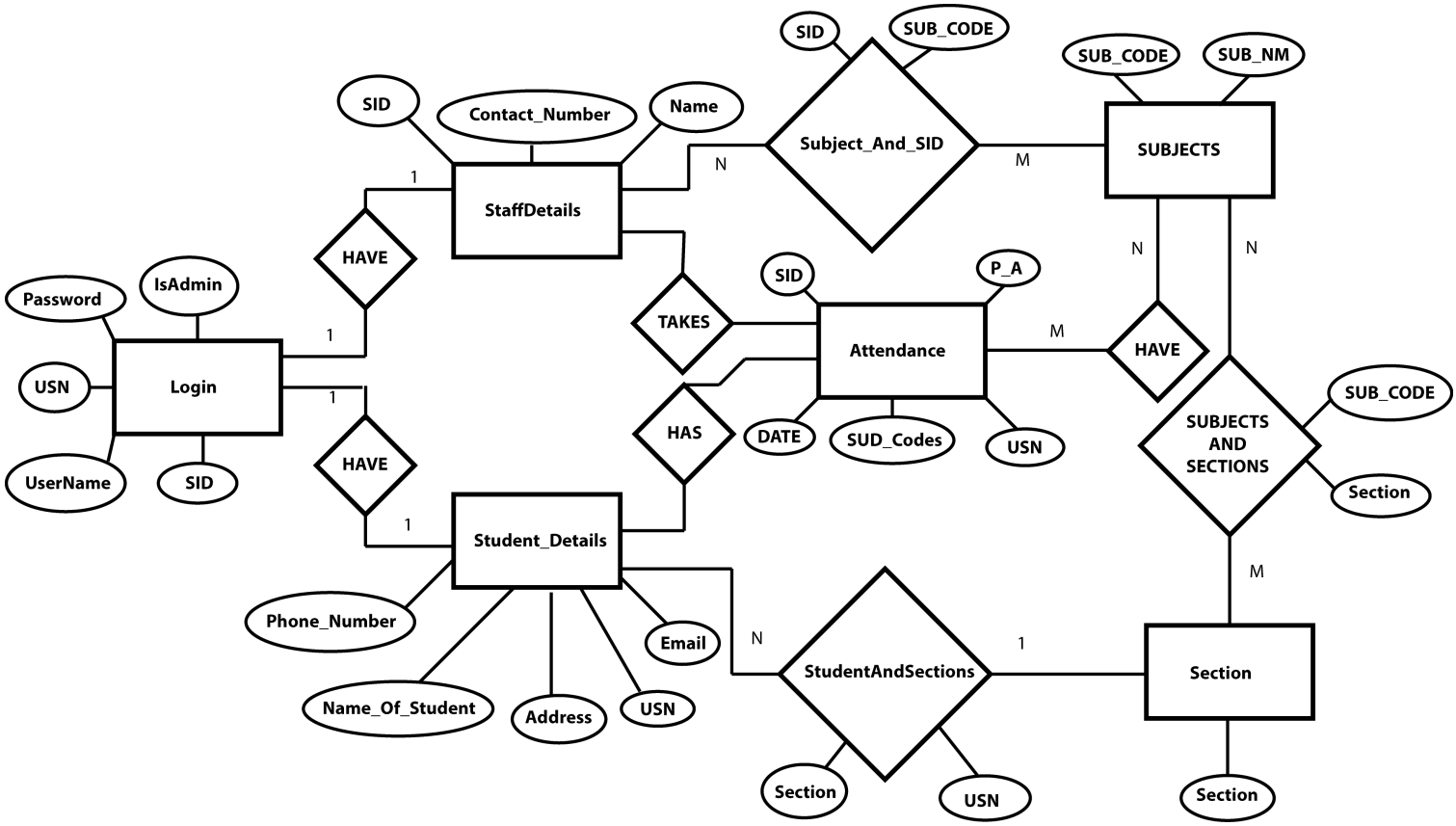
- Technology Implemented : MySQL Server
- Language Used : J2EE
- Database : My SQL
- User Interface Design : JFrames
- Web Browser : Google Chrome, Mozilla, IE8
- Software : NetBeans, MySQL Server

CHAPTER 3

DESIGN

1. Database Design

1.1. E-R Diagram



1.2. Database Schema

Database: Mini_project

Table: Attendance

Column	Type	Primary key	Not Null	UQ	AI
SUB_Code	VARCHAR(20)	NO	NO		
DATE	DATETIME	NO	NO		
P_A	TINYINT(4)	NO	NO		YES
A.NO	INT(11)	YES	YES		
USN	VARCHAR(20)	YES	YES		
SID	VARCHAR(20)	YES	YES		

Table: Login

Column	Type	Primary key	Not Null	UQ	AI
Username	VARCHAR(20)	YES	NO		
Password	VARCHAR(45)		NO		
IsAdmin	TINYINT(4)				
USN	VARCHAR(45)				
SID	VARCHAR(45)				

Table: Staff Details

Column	Type	Primary key	Not Null	UQ	AI
SID	VARCHAR(20)	YES	YES		
Name	VARCHAR(20)		YES		
Contact_Number	VARCHAR(20)				

Table: StudentAndSections

Column	Type	Primary key	Not Null	UQ	AI
USN	VARCHAR(20)	YES	YES		
SECTION	VARCHAR(45)		YES		

Table: Student_Details

Column	Type	Primary key	Not Null	UQ	AI
SL_NO	NUMBER				
USN	VARCHAR(45)	YES	YES		
Name_Of_Student	VARCHAR(45)		YES		
Permanent_Address	VARCHAR(100)				
Phone_Number	NUMBER				
Email	VARCHAR(20)				

Table: SUBCODE_SUB_NM

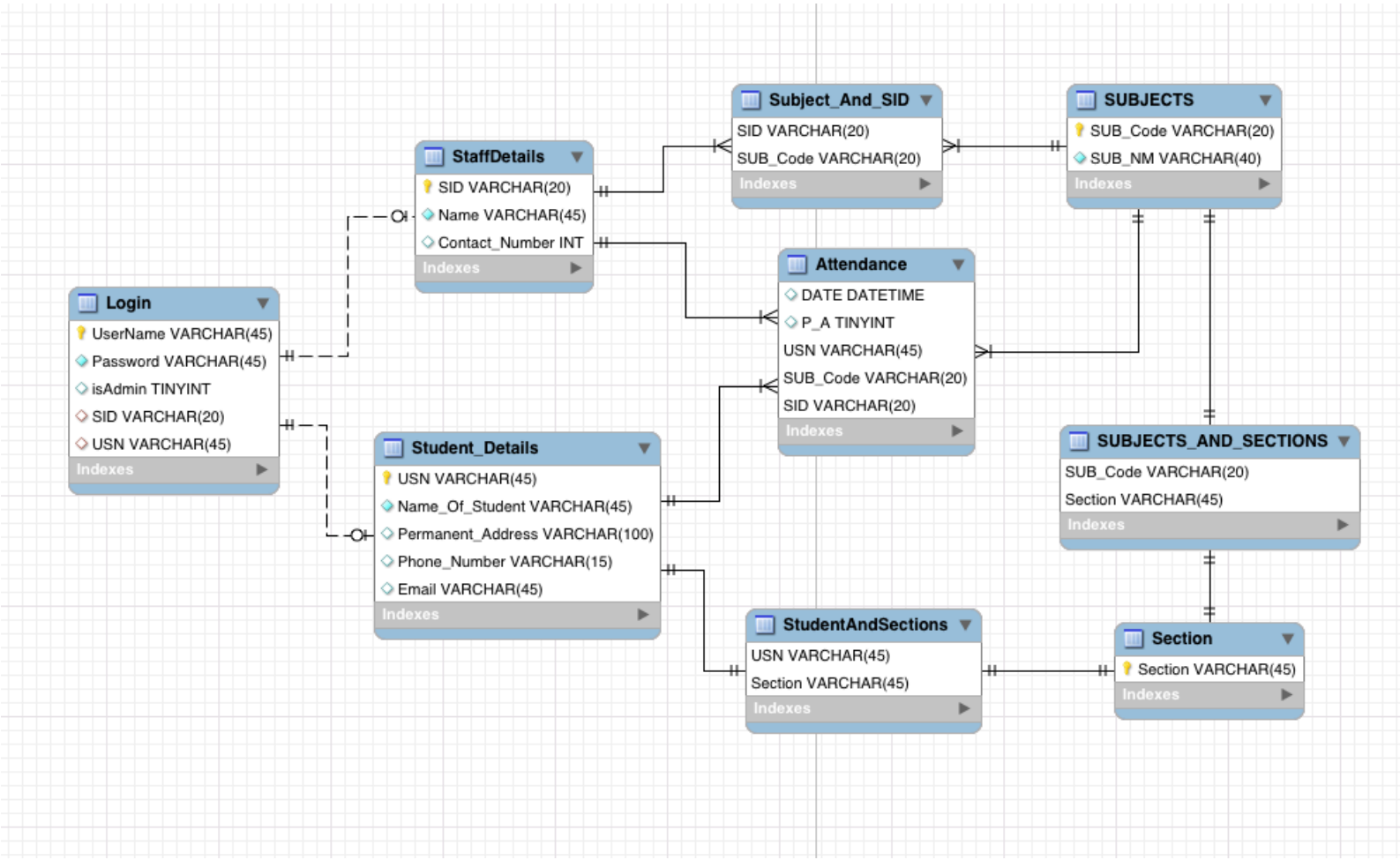
Column	Type	Primary key	Not Null	UQ	AI
SUBCODE	VARCHAR(45)	YES	YES		
SUB_NM	VARCHAR(45)		YES		

Table: Subject_Details

Column	Type	Primary key	Not Null	UQ	AI
SUB_Code	VARCHAR(40)	YES	YES		
SID	VARCHAR(40)	YES	YES		

1.3. Relational Schema

Database: Mini_project



2. Database Normalization

2.1. First Normal Form

All the Relations are designed in such a way that it has no repeating groups. Hence all tables are in 1st Normal Form.

2.2. Second Normal Form

A relation is said to be in second normal form if it is already in first normal form and it has no partial dependency. All the tables in the database are designed in such a way that there is no partial dependency. Hence all tables are in 2nd Normal Form.

2.3. Third Normal Form

A relation is said to be in third normal form if it is already in 1st and 2nd Normal Form and has no transitive dependency. All the tables in the database are designed in such a way that there is no transitive dependency. Hence all tables are in 3rd Normal Norm.

For Example:

- Every attribute of StaffDetails depends on primary key so it is in 3rd normal form
- In SUBJECTS table, SUB_NM depends upon the SUB_Code, hence another table Subject_And_SID with SID and SUB_CODE is created to remove the transitive functional dependency.
- In StudentAndSection table, Section depends upon the USN, hence another table Student_Details with Section is created to remove the transitive functional dependency.

3.3 User Interface

The Student Attendance Management System can be entered using a username and password. ‘admin’ can access all functionalities of the system. ‘staff’ can access only attendance details and can only add, modify and delete the attendance details and see the student list as shown in Figure 3.1.

3.3.1 Login Module

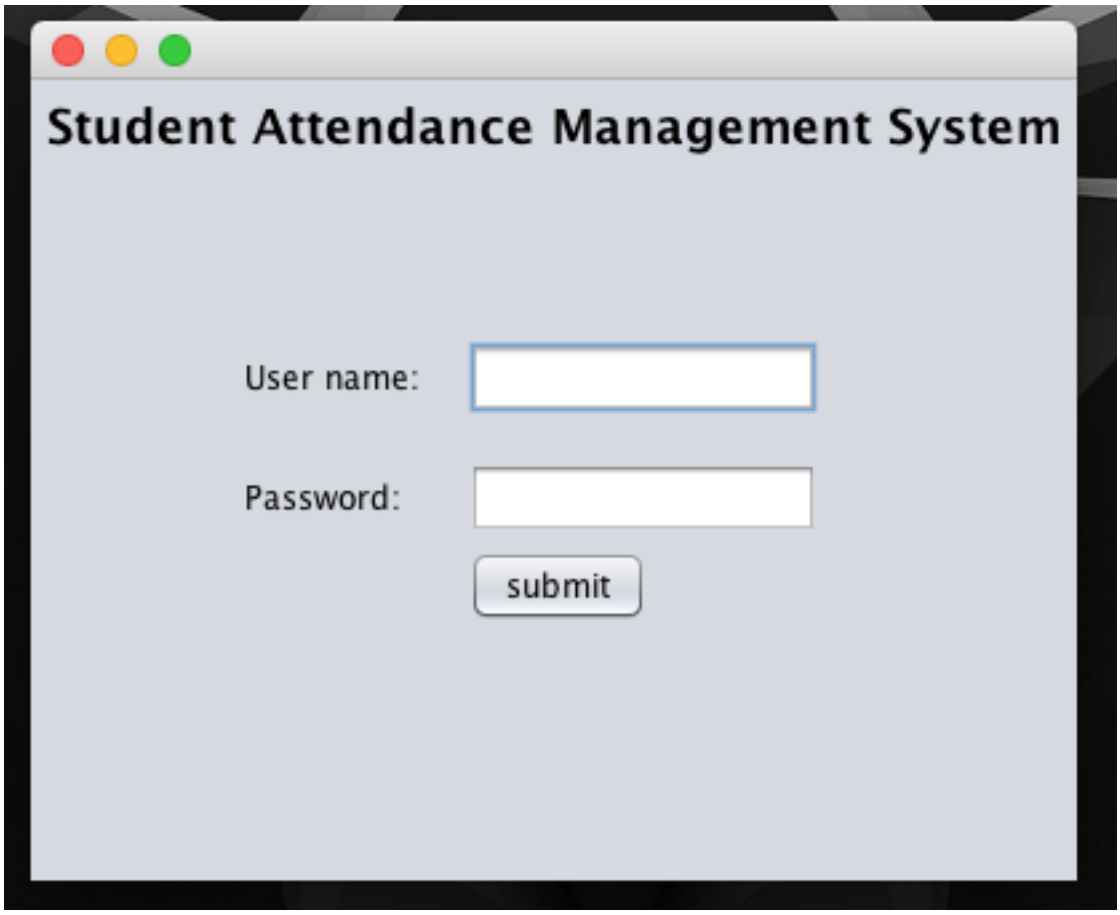


Figure 3.1 Login Module

3.3.2 Staff Module:

Add Teacher:

Figure 3.2 Staff Module

3.3.3 Student Module:

Add Student:

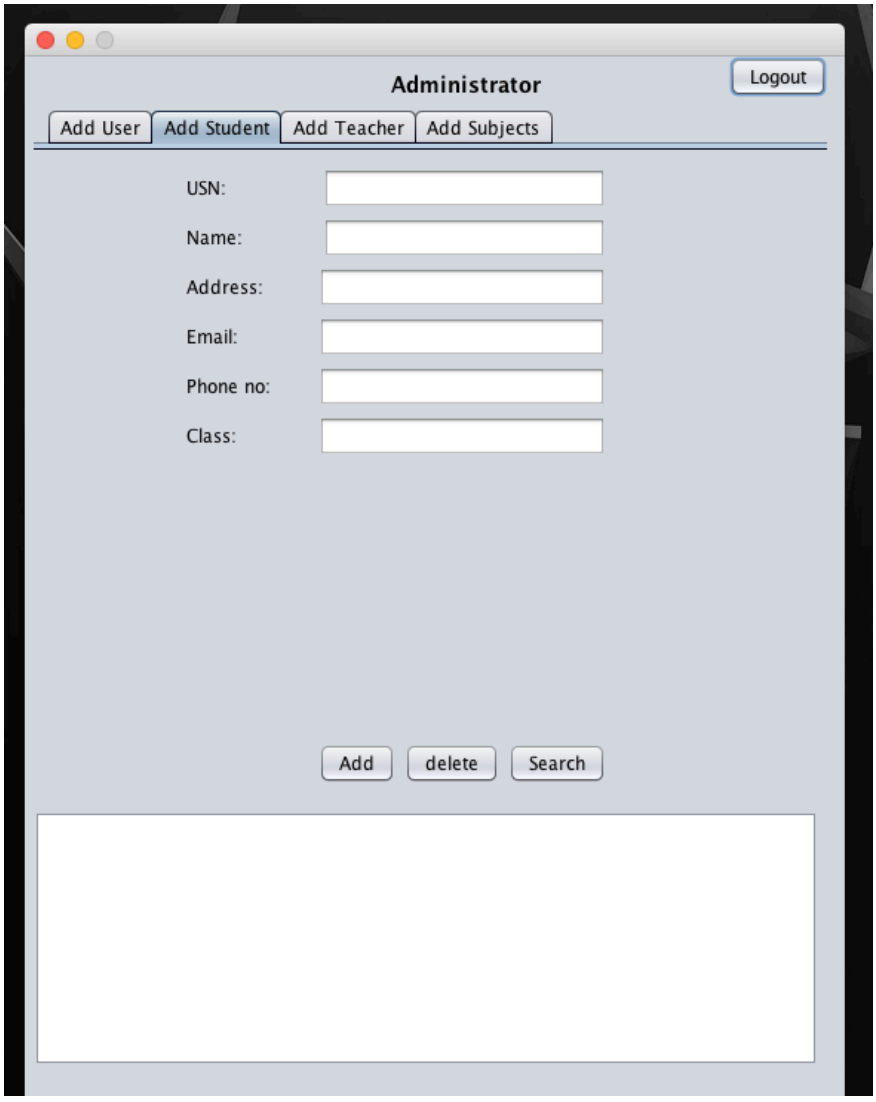


Figure 3.3 Student Module

3.3.4 User Module:

Add User:

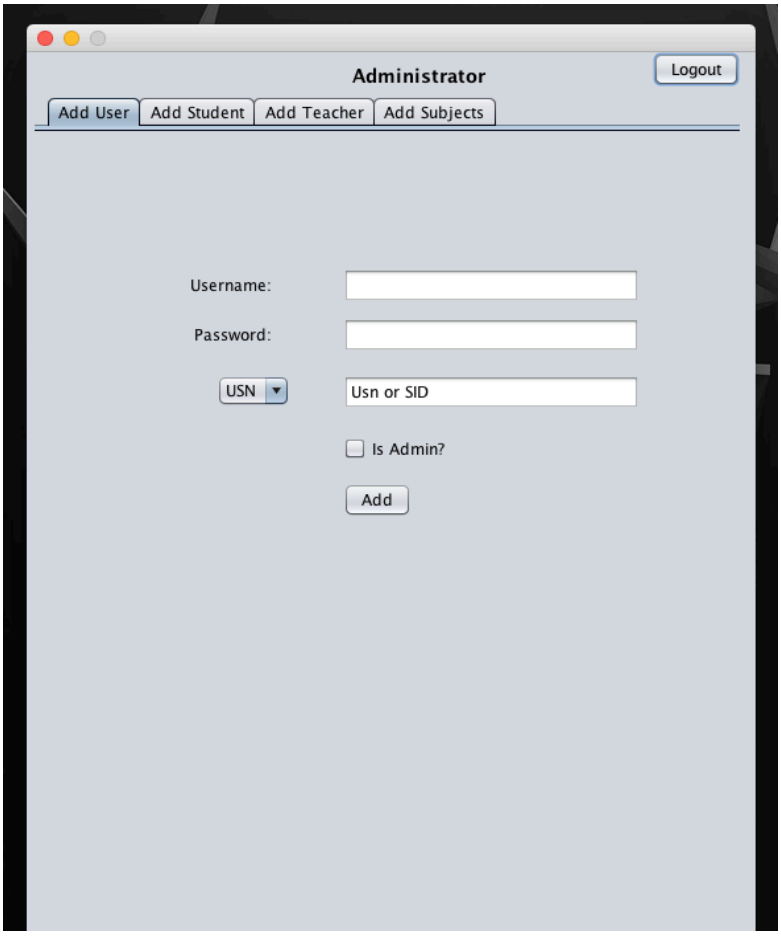


Figure 3.4 User Module

3.3.5 Subjects Module:

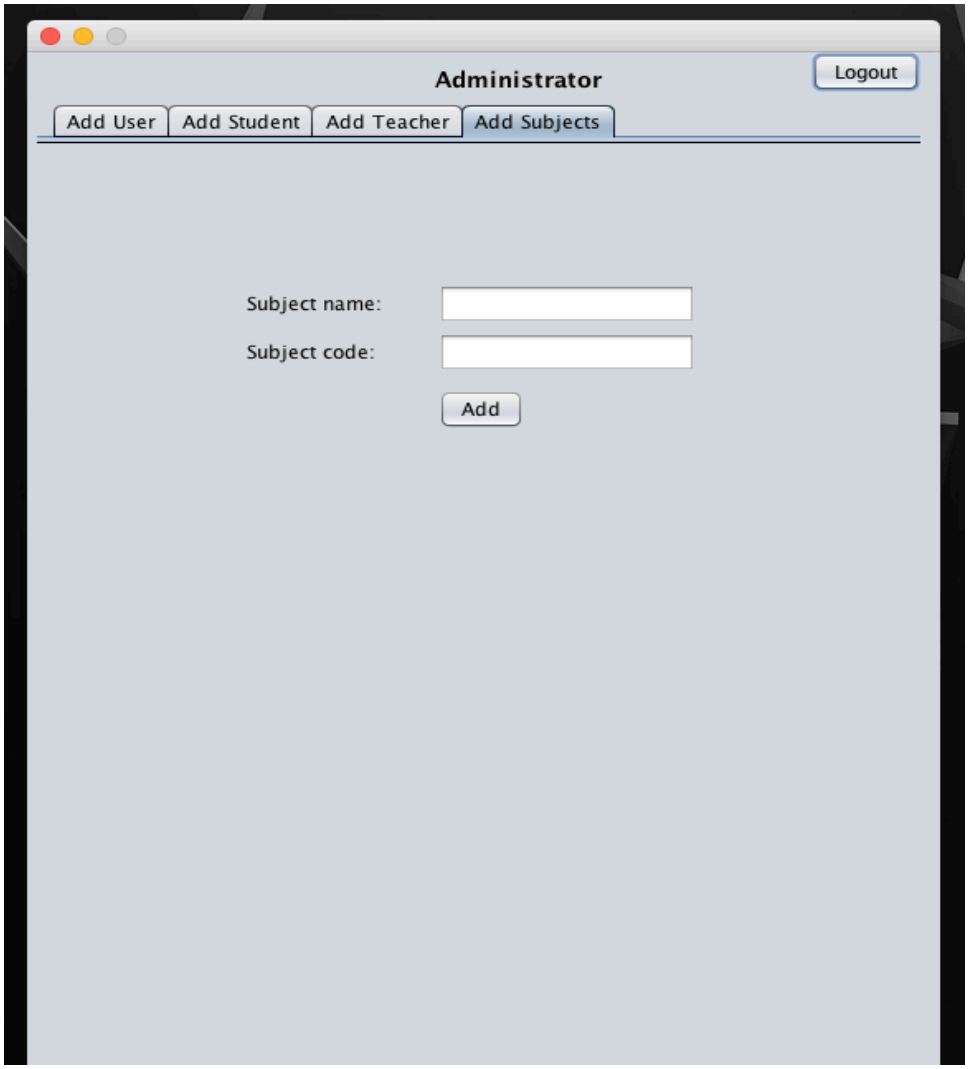


Figure 3.5 Subject Module

3.3.6 Attendance Details:

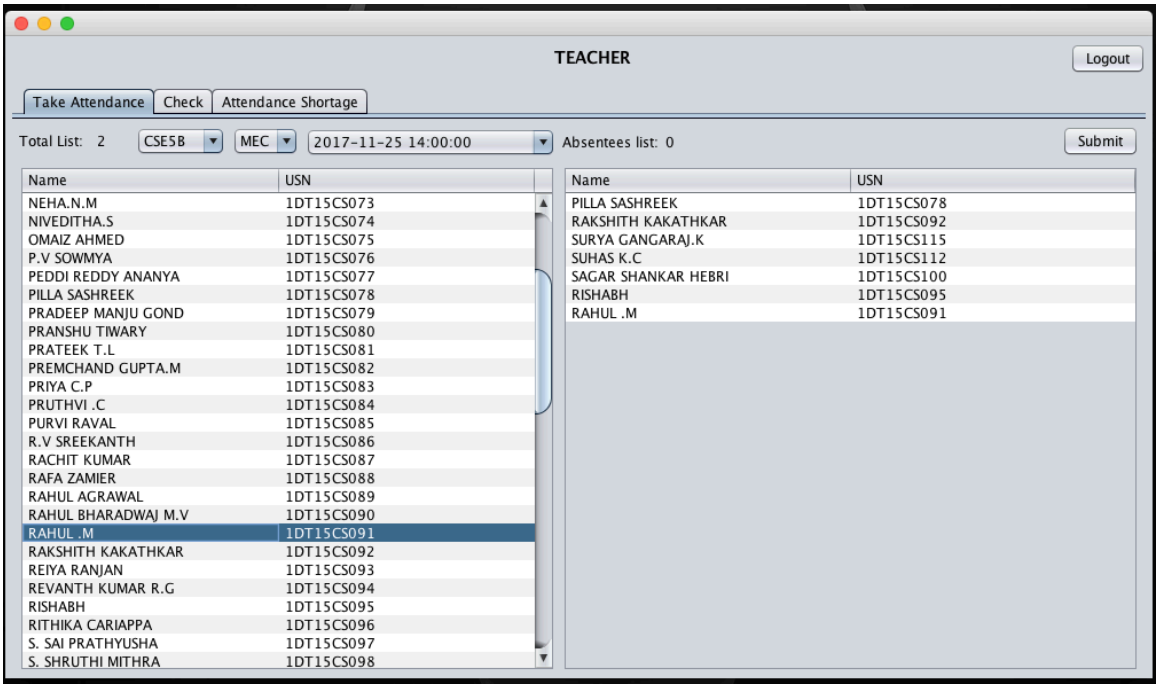


Figure 3.6 Attendance Module

CHAPTER 4

IMPLEMENTATION

1. Login Module

Process Name	**:** Login
Process Number	: 1.1
Input	: User Name : Password
Output	: Status Message
Error Condition	: User Name Already Exists : All Fields are Required

4.2 Staff Module

Process Name	**: Add Teacher**
Process Number	: 2.1
Input	: Name : SID : ContactNumber
Output	: Status Message
Error Condition	: SID Already Exists : SID cannot be Empty

4.3 Student Module

Process Name	**: Add new Student**
Process Number	: 3.1
Input	: USN : Name : Address : Email :Phone no. :Class
Output	: Status Message
Error Condition	: USN Already Exists : SQL error for Class foreign key constraint : All fields are Required

Process Name	**: Search Student**
Process Number	: 3.2
Input	: USN
Output	: Status Message
Error Condition	: No Records Found : SQL error for Class foreign key constraint : All fields are Required

Process Name	**: Delete Student**
Process Number	: 3.3
Input	: USN
Output	: Status Message
Error Condition	: No Records Found

4. User Module

Process Name	**: Add New USER**
Process Number	: 4.1
Input	: UserName : Password : Usn or SID
Output	: Status Message
Error Condition	: User Name Already Exists : All fields are Required

5. Subject Module

Process Name	**: Add Subject name**
Process Number	: 5.1
Input	: Subject name : Subject code
Output	: Status Message
Error Condition	: Subject code Already Exists : All fields are Required

6. Attendance Module

Process Name	**: Take Attendance**
Process Number	: 6.1
Input	: Section : Subject :TimeStamp
Output	: Status Message
Error Condition	: No Students : All fields are Required

Process Name	**: Check Attendance**
Process Number	: 6.2
Input	: TimeStamp :Subject code
Output	: Status Message
Error Condition	: No Students found : All fields are Required

Process Name	**: Attendance Shortage**
Process Number	: 6.3
Input	: Timestamp :Subject code
Output	: Status Message
Error Condition	: No Students Found : SQL error for SUB_CODE foreign key constraint

CHAPTER 5

SOURCE CODE

ADMIN FUNCTION:

```
private void submitbtnActionPerformed(java.awt.event.ActionEvent evt)
      conn = MySqlConnect.ConnectDB();
      String Sql = "select * from Student_Details where USN=?";
      String usnGot = usn2.getText().toUpperCase();
      try {
         pst = conn.prepareStatement(Sql);
         pst.setString(1, usnGot);
         rs = pst.executeQuery();
         if (rs.next()) {
            String c1 = "" + rs.getString(2) + "\n";
            c1 += "" + rs.getString(3) + "\n";
            c1 += "" + rs.getString(4) + "\n";
            c1 += "" + rs.getString(5) + "\n";
            c1 += "" + rs.getString(6) + "\n";
            c1 += "" + rs.getString(7) + "\n";
            output.setText(c1);
         } else {
            JOptionPane.showMessageDialog(null, "USN not found");
         }
      } catch (Exception e) {
         JOptionPane.showMessageDialog(null, e);
      }
private void submitbtnActionPerformed(java.awt.event.ActionEvent evt) {
     conn = MySqlConnect.ConnectDB();
      String Sql = "select * from Student_Details where USN=?";
      String usnGot = usn2.getText().toUpperCase();
      try {
         pst = conn.prepareStatement(Sql);
         pst.setString(1, usnGot);
```

```
            rs = pst.executeQuery();
            if (rs.next()) {
                String c1 = "" + rs.getString(2) + "\n";
                c1 += "" + rs.getString(3) + "\n";
                c1 += "" + rs.getString(4) + "\n";
                c1 += "" + rs.getString(5) + "\n";
                c1 += "" + rs.getString(6) + "\n";
                c1 += "" + rs.getString(7) + "\n";
                output.setText(c1);
            } else {
                JOptionPane.showMessageDialog(null, "USN not found");
            }
        } catch (Exception e) {
            JOptionPane.showMessageDialog(null, e);
        }
    }
    private void stdAddBtnActionPerformed(java.awt.event.ActionEvent evt) {//GEN-
FIRST:event_stdAddBtnActionPerformed
//Class_st
        try {
            /// Student Add btn
            conn = MySqlConnect.ConnectDB();
            String usn = usn2.getText().toUpperCase();
            String name = name2.getText();
            String add = add2.getText();
            String email = email2.getText();
            String phone = phone2.getText();
            String className = Class_st.getText();
            className = className.toUpperCase();
            if (add.isEmpty()) {
                add = "Empty Address";
            }
            if (email.isEmpty()) {
                email = "Empty Email Address";
```

```
        }
        if (phone.isEmpty()) {
            phone = "Empty Phone Number";
        }
        if (className.isEmpty()) {
            status3.setText("Class name is empty");
        }
        if (!className.isEmpty()) {

            String Sql = "INSERT INTO Student_Details ( USN, Name_Of_Student,"
                    + " Permanent_Address, Phone_Number, Email,CLASS) VALUES
( ?, ?, ?, ?, ?,?)";
            PreparedStatement preparedStmt = conn.prepareStatement(Sql);
            preparedStmt.setString(1, usn);
            preparedStmt.setString(2, name);
            preparedStmt.setString(3, add);
            preparedStmt.setString(4, phone);
            preparedStmt.setString(5, email);
            preparedStmt.setString(6, className);

            // execute the java preparedstatement
            preparedStmt.executeUpdate();
            status3.setText("Student Added");
        }
        ///
    } catch (SQLException ex) {
        Logger.getLogger(AdminView.class.getName()).log(Level.SEVERE, null, ex);
    }

  }
```

LOGIN FUNCTION:

```
private void submitBtnActionPerformed(java.awt.event.ActionEvent evt) {
    conn = MySqlConnect.ConnectDB();
```

```
String usr = usernameId.getText();
String pasd = passId.getText();
String Sql = "select * from Login where UserName=? and Password=?";
try {
   pst = conn.prepareStatement(Sql);
   pst.setString(1, usr);
   pst.setString(2, pasd);
   rs = pst.executeQuery();
   if (rs.next()) {
      String ad = rs.getString(3);
      String usn = rs.getString(4);
      String sid = rs.getString(5);

      if (ad.equals("0") && (sid != null) && (usn == null)) {
         this.SID=sid;
         (new TeacherFrame()).setVisible(true);

         this.setVisible(false);
      }
      if (ad.equals("0") && (usn != null) && (sid == null)) {
         this.USN=usn;
         JOptionPane.showMessageDialog(null, "students");
      }
      if (ad.equals("1")) {
         (new AdminView(true)).setVisible(true);
         this.setVisible(false);
      }
   } else {
      statusId.setText("incorrect password or username");
   }
} catch (Exception e) {
}
```

TEACHER FUNCTION:

```
public TeacherFrame() {
        initComponents();
        DefaultTableModel model = (DefaultTableModel) jTable1.getModel();
        Vector comboBoxItems = new Vector();
         String Sql1 = "select SUB_NM from Subject_Details where SID='" + SID + "';";
        comboBoxItems = new Vector();
        conn = MySqlConnect.ConnectDB();
        try {
           Statement st = conn.createStatement();
           rs = st.executeQuery(Sql1);
           while (rs.next()) {
               comboBoxItems.add(rs.getString(1));
           }
        } catch (Exception e) {

        }
```

BIBLIOGRAPHY

BOOK REFERENCES:

- J2EE The Complete Reference And JAVA The Complete Reference
- *Ramakrishnan-Database Management Systems*

WEBSITE REFERENCES:

JAVA Learning:

- https://www.codecademy.com/
- https://www.w3schools.com/

MYSQL Learning:

- https://www.w3schools.com/
- https://www.Mysql.com/