

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**. 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

Chapter #	Chapter Name	Page #
1	INTRODUCTION	1
1.1	Background	1
1.2	Problem Definition	1
1.3	Motivation	1
1.4	Objective	2
1.5	Scope of the project	2
2	REQUIREMENTS	3
2.1	Hardware Requirements	3
2.2	Software Requirements	3
3	DESIGN	4
3.1	Database Design	4
3.1.1	E-R Diagram	4
3.1.2	Database Schema	5
3.1.3	Relational Schema	7
3.2	Database Normalization	8
3.2.1	First Normal Form	8
3.2.2	Second Normal Form	8
3.2.3	Third Normal Form	8
3.3	User Interface	9
3.3.1	Login Module	9
3.3.2	Staff Module	10
3.3.3	Tourist Module	11
3.3.4	Animal Module	13
3.3.5	Species Module	14
3.3.6	Data Module	16
4	IMPLEMENTATION	18
4.1	Staff Module	18
4.2	Tourist Module	19
4.3	Animal Module	20
4.4	Species Module	21
4.5	Data Module	22
5	SOURCE CODE	23
6	CONCLUSION AND FUTURE WORK	24
6.1	Advantages	24

BIBLIOGRAPHY**LIST OF FIGURES**

SL #	FIGURE #	TOPIC	PAGE #
1	Figure 3.1	Login Module	9
2	Figure 3.2	Add New Staff	10
3	Figure 3.3	Update Staff Details	10
4	Figure 3.4	Delete Staff Details	11
5	Figure 3.5	Add New Tourist	11
6	Figure 3.6	Update Tourist Details	12
7	Figure 3.7	Delete Tourist Details	12
8	Figure 3.8	Add New Animal	13
9	Figure 3.9	Update Animal Details	13
10	Figure 3.10	Delete Animal Details	14
11	Figure 3.11	Add New Species	14
12	Figure 3.12	Update Species Details	15
13	Figure 3.13	Delete Species Details	15
14	Figure 3.14	Staff Data	16
15	Figure 3.15	Tourist Data	16
16	Figure 3.16	Animal Data	17
17	Figure 3.17	Species Data	17

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

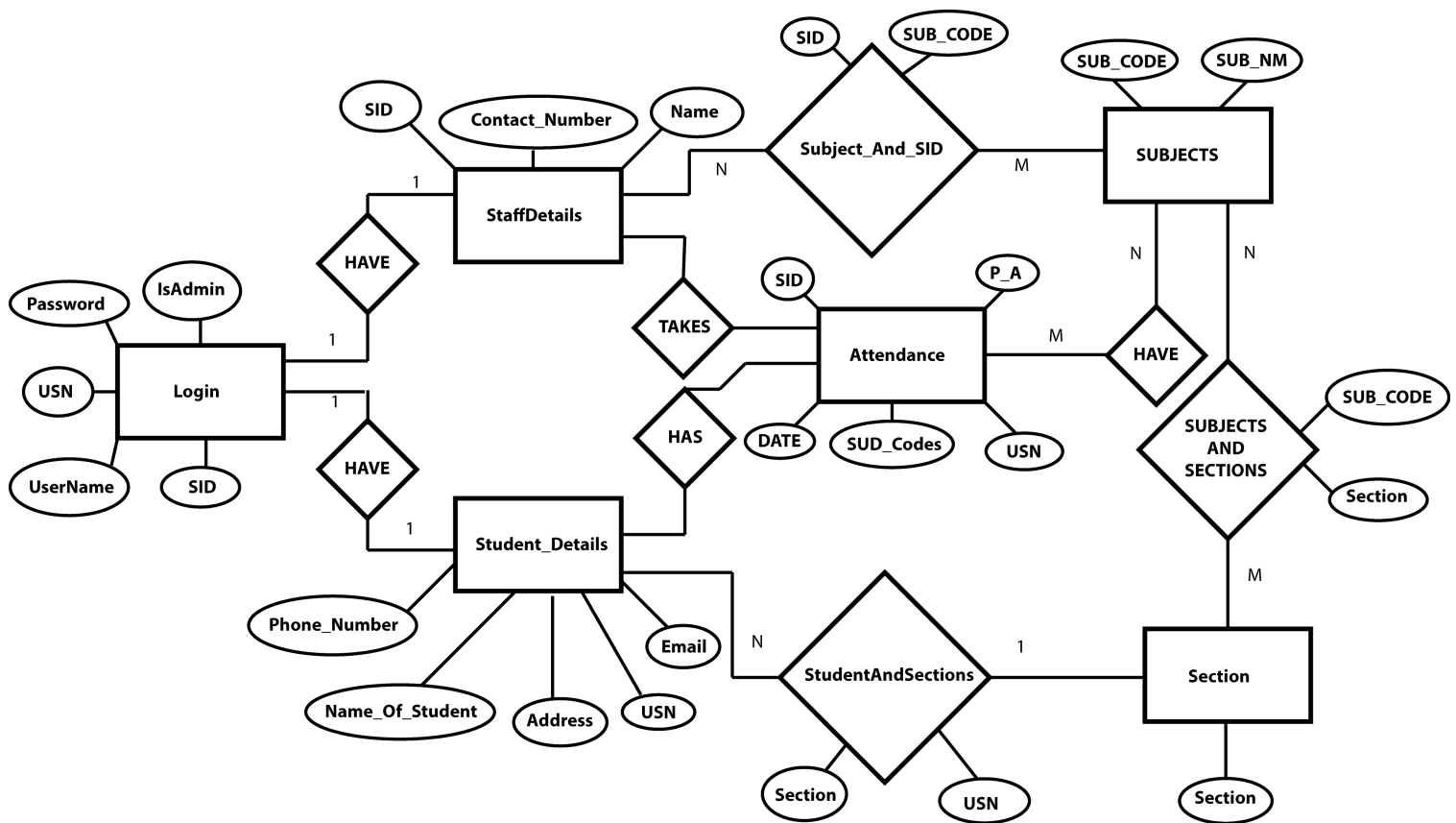
- Technology Implemented : MySQL Server
- Language Used : J2EE
- Database : My SQL
- User Interface Design : JFrames
- Web Browser : Google Chrome, Mozilla, IE8
- Software : NetBeans, MySQL Server

CHAPTER 3

DESIGN

1. Database Design

1. E-R Diagram



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

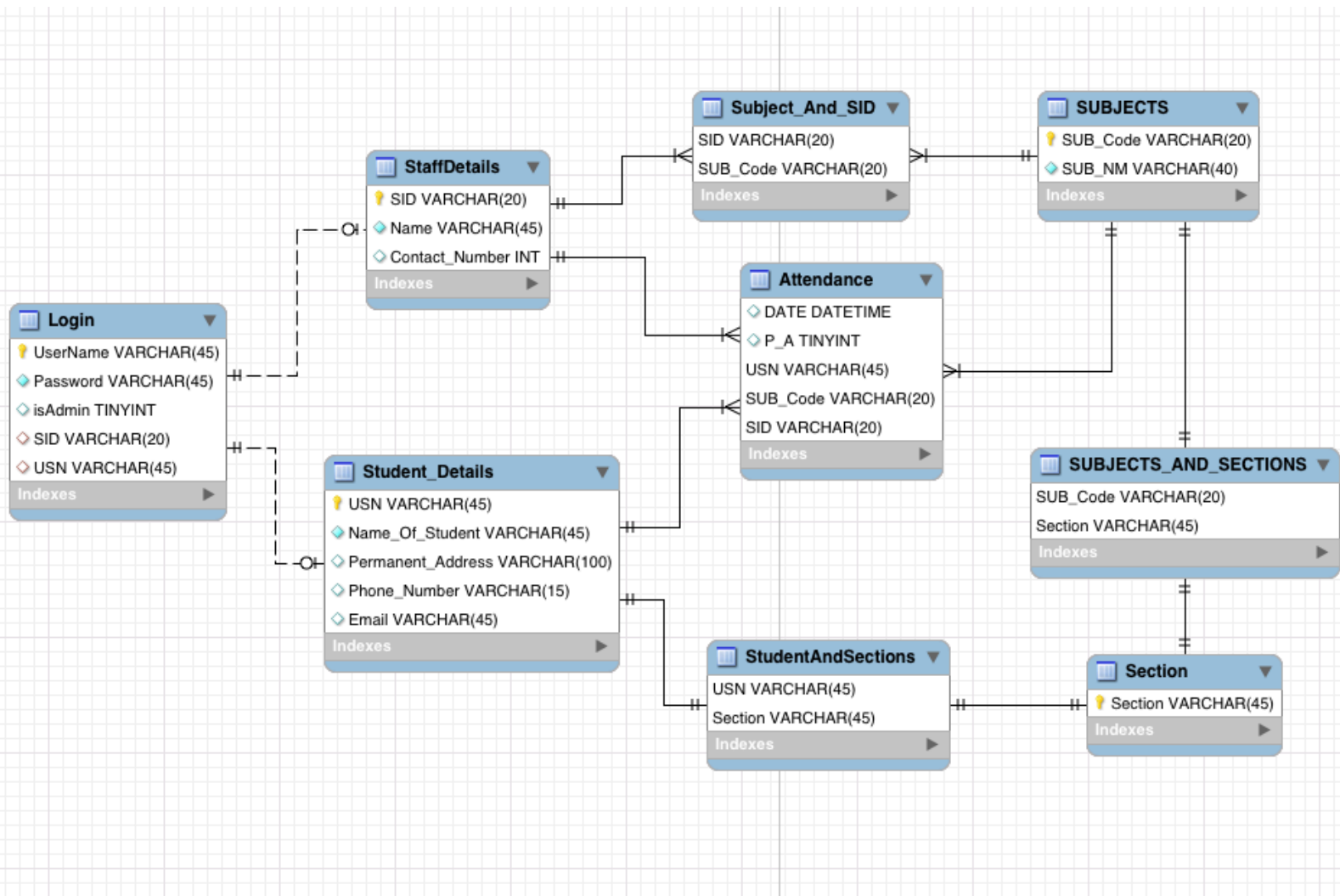
Column	Type	Primary key	Not Null	UQ	AI
SUBCODE	VARCHAR(45)	YES	YES		
SUB_NM	VARCHAR(45)		YES		

Table: Subject_Details

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

3. Relational Schema

Database: Mini_project



2. Database Normalization

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. 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.

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. 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.

1. Login Module

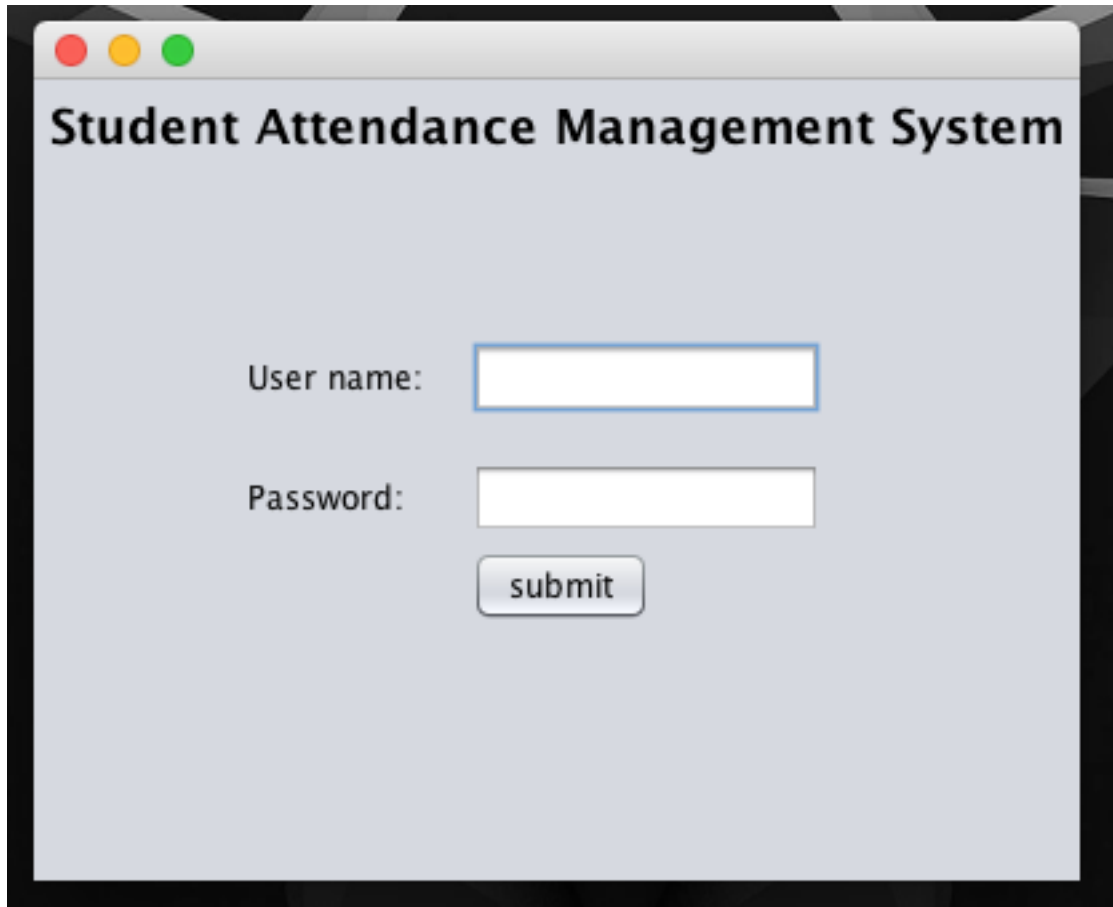
The image shows a screenshot of a web application window titled "Student Attendance Management System". The window has a light gray background and a dark gray border. At the top, there are three colored circles (red, yellow, green) representing window control buttons. Below the title, there are two input fields: "User name:" and "Password:". The "User name:" field is a white rectangle with a blue border. The "Password:" field is a white rectangle with a gray border. Below the "Password:" field is a "submit" button, which is a rounded rectangle with a gray border and the word "submit" in black text.

Figure 3.1 Login Module

Staff Module:**Add New Staff:**

Administrator [Logout](#)

[Add User](#) [Add Student](#) [Add Teacher](#) [Add Subjects](#)

NAME:

SID:

ContactNumber:

Available Subjects:

SUBCODE	SUB NAME

Subjects Selected:

SUBCODE	SUB NAME

[Submit](#)

Student Module:

The screenshot displays the 'Administrator' window of the Wildlife Sanctuary Management System. The window has a standard macOS-style title bar with red, yellow, and green window control buttons. The title 'Administrator' is centered at the top, and a 'Logout' button is located in the top right corner. Below the title bar, there are four tabs: 'Add User', 'Add Student', 'Add Teacher', and 'Add Subjects'. The 'Add Student' tab is the active one. The main area of the window contains a form with six input fields, each preceded by a label: 'USN:', 'Name:', 'Address:', 'Email:', 'Phone no:', and 'Class:'. At the bottom of the form area, there are three buttons: 'Add', 'delete', and 'Search'. Below these buttons is a large, empty rectangular box, likely intended for displaying a list of students or other data.

User Module:

The screenshot displays a web application window titled "Administrator". At the top right is a "Logout" button. Below the title bar is a navigation menu with four buttons: "Add User", "Add Student", "Add Teacher", and "Add Subjects". The main content area contains a form for adding a new user. It includes a "Username:" label followed by a text input field, a "Password:" label followed by a text input field, a dropdown menu currently showing "USN" with a downward arrow, and a text input field labeled "Usn or SID". Below these is a checkbox labeled "Is Admin?". At the bottom of the form is an "Add" button.

Subjects Module:

The screenshot shows a web application window titled "Administrator" with a "Logout" button in the top right corner. Below the title bar, there is a navigation menu with four buttons: "Add User", "Add Student", "Add Teacher", and "Add Subjects". The "Add Subjects" button is currently selected. The main content area of the window contains two input fields: "Subject name:" and "Subject code:". Below these fields is an "Add" button.

Administrator [Logout](#)

[Add User](#) [Add Student](#) [Add Teacher](#) [Add Subjects](#)

Subject name:

Subject code:

[Add](#)

Attendance Details:

TEACHER Logout

Take Attendance Check Attendance Shortage

Total List: 2 CSE5B MEC 2017-11-25 14:00:00 Absentees list: 0 Submit

Name	USN	Name	USN
NEHA.N.M	1DT15CS073	PILLA SASHREEK	1DT15CS078
NIVEDITHA.S	1DT15CS074	RAKSHITH KAKATHKAR	1DT15CS092
OMAIZ AHMED	1DT15CS075	SURYA GANGARAJ.K	1DT15CS115
P.V SOWMYA	1DT15CS076	SUHAS K.C	1DT15CS112
PEDDI REDDY ANANYA	1DT15CS077	SAGAR SHANKAR HEBRI	1DT15CS100
PILLA SASHREEK	1DT15CS078	RISHABH	1DT15CS095
PRADEEP MANJU GOND	1DT15CS079	RAHUL .M	1DT15CS091
PRANSHU TIWARY	1DT15CS080		
PRATEEK T.L	1DT15CS081		
PREMCHAND GUPTA.M	1DT15CS082		
PRIYA C.P	1DT15CS083		
PRUTHVI .C	1DT15CS084		
PURVI RAVAL	1DT15CS085		
R.V SREEKANTH	1DT15CS086		
RACHIT KUMAR	1DT15CS087		
RAFA ZAMIER	1DT15CS088		
RAHUL AGRAWAL	1DT15CS089		
RAHUL BHARADWAJ M.V	1DT15CS090		
RAHUL .M	1DT15CS091		
RAKSHITH KAKATHKAR	1DT15CS092		
REIYA RANJAN	1DT15CS093		
REVANTH KUMAR R.G	1DT15CS094		
RISHABH	1DT15CS095		
RITHIKA CARIAPPA	1DT15CS096		
S. SAI PRATHYUSHA	1DT15CS097		
S. SHRUTHI MITHRA	1DT15CS098		

CHAPTER 4

IMPLEMENTATION

1. Staff Module

Process Name	: Add New Staff
Process Number	: 1.1
Input	: Staff ID : Staff Name : Occupation : Salary
Output	: Status Message
Error Condition	: Staff ID Already Exists : Salary should be numeric : All Fields are Required

Process Name	: Update Staff Details
Process Number	: 1.2
Input	: Staff ID
Output	: Status Message
Error Condition	: No Records Found : Salary should be numeric : All Fields are Required

Process Name	: Delete Staff
Process Number	: 1.3
Input	: Staff ID
Output	: Status Message
Error Condition	: SQL error for Staff_ID with Guide_ID Foreign Key constraint

2. Tourist Module

Process Name	: Add New Tourist
Process Number	: 2.1
Input	: Tourist ID : Date Visited : Guide ID : Visit Fare
Output	: Status Message
Error Condition	: Tourist ID Already Exists : Date Visited can't be greater than today's date : SQL error for Guide_ID foreign key constraint : Visit Fare should be numeric : All Fields are Required

Process Name	: Update Tourist Details
Process Number	: 2.2
Input	: Tourist ID
Output	: Status Message
Error Condition	: No Records Found : Date Visited can't be greater than today's date : SQL error for Guide_ID foreign key constraint : Visit Fare should be numeric : All Fields are Required

Process Name	: Delete Tourist
Process Number	: 2.3

Input	: Tourist ID
Output	: Status Message
Error Condition	: No Records Found : SQL error for Guide_ID Foreign Key constraint

3. Animal Module

Process Name	: Add New Animal
Process Number	: 3.1
Input	: Animal ID : Animal Type : Species ID : Feed
Output	: Status Message
Error Condition	: Animal ID Already Exists : SQL error for Species_ID foreign key constraint : SQL error for Feed foreign key constraint : All fields are Required

Process Name	: Update Animal Details
Process Number	: 3.2
Input	: Animal ID
Output	: Status Message
Error Condition	: No Records Found : SQL error for Species_ID foreign key constraint : SQL error for Feed foreign key constraint : All fields are Required

Process Name	: Delete Animal
Process Number	: 3.3
Input	: Animal ID
Output	: Status Message

Error Condition	: No Records Found
-----------------	--------------------

4. Species Module

Process Name	: Add New Species
Process Number	: 4.1
Input	: Species ID : Biological Name : Family
Output	: Status Message
Error Condition	: Species ID Already Exists : All fields are Required

Process Name	: Update Species Details
Process Number	: 4.2
Input	: Species ID
Output	: Status Message
Error Condition	: No Records Found : All fields are Required

Process Name	: Delete Species
Process Number	: 4.3
Input	: Species ID
Output	: Status Message
Error Condition	: No Records Found : SQL error for Species_ID foreign key constraint

Process Name	: Staff Data
Process Number	: 8.1
Input	: None
Output	: Staff Details in tabular format
Error Condition	: None
Process Name	: Tourist Data
Process Number	: 8.2
Input	: None
Output	: Tourist Details in tabular format
Error Condition	: None
Process Name	: Animal Data
Process Number	: 8.3
Input	: None
Output	: Animal Details in tabular format
Error Condition	: None
Process Name	: Species Data
Process Number	: 8.4
Input	: None
Output	: Species Details in tabular format
Error Condition	: None
Process Name	: Feed Data
Process Number	: 8.5
Input	: None
Output	: Feed Details in tabular format
Error Condition	: None
Process Name	: Guide Data
Process Number	: 8.6

Input	: None
Output	: Guide Details in tabular format
Error Condition	: None

CHAPTER 5**SOURCE CODE****ADMIN FUNCTION:**

```
private void submitbtnActionPerformed(java.awt.event.ActionEvent evt)
```

```
    conn = MySqlConnection.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 = MySqlConnection.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 = MySqlConnection.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 = MySqlConnection.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 = MySqlConnection.ConnectDB();
    try {
        Statement st = conn.createStatement();
        rs = st.executeQuery(Sql1);
        while (rs.next()) {
            comboBoxItems.add(rs.getString(1));
        }
    } catch (Exception e) {

    }
}
```

CHAPTER 6

CONCLUSION AND FUTURE WORK

The Wildlife Sanctuary Management System is a great improvement over the manual system which uses lots of manual work and paper. The computerization of the system speeds up the process. This system was thoroughly checked and tested with dummy data and found to be very reliable.

1. Advantages

- The Wildlife Sanctuary Management System is fast, efficient and reliable.
- Avoids data redundancy and inconsistency
- Web-based
- Number of personnel required is considerably less
- Provides more security and integrity to data

2. Future Enhancements

The Wildlife Sanctuary Management System can be enhanced by including more functionality like entering Animal caretaker details, Track the Animal Health, maintain Tourists Feedback, Reports, Receipt Generation to Tourists, Billing etc.

BIBLIOGRAPHY

BOOK REFERENCES:

- Learn to Code HTML and CSS: Develop and Style Websites (Web Design Courses) 1st, Kindle Edition by Shay Howe
- *PHP 6 and MySQL 5* - Larry Ullman

WEBSITE REFERENCES:

HTML Learning:

- <https://www.codecademy.com/>
- <https://dash.generalassemb.ly/>
- <https://www.w3schools.com/>

PHP Learning:

- <http://www.tutorialspoint.com/php/>
- <https://killerphp.com>
- <https://www.w3schools.com/>