**PROJECT REPORT ON**

**CONNECT2CLASS**

**(Classes Management System)**

**UNDER THE GUIDANCE OF**

**TEACHER NAME**

**SUBMITTED BY**

**Seguro JC**

**SEAT NO. Rollno**

**UNIVERSITY OF MUMBAI**

**T.Y.B.Sc (COMPUTER SCIENCE)**

**ACADEMIC YEAR: 2020-2021**

**COLLEGE NAME**

**ADDRESS**

**COLLEGE NAME**

**CERTIFICATE**

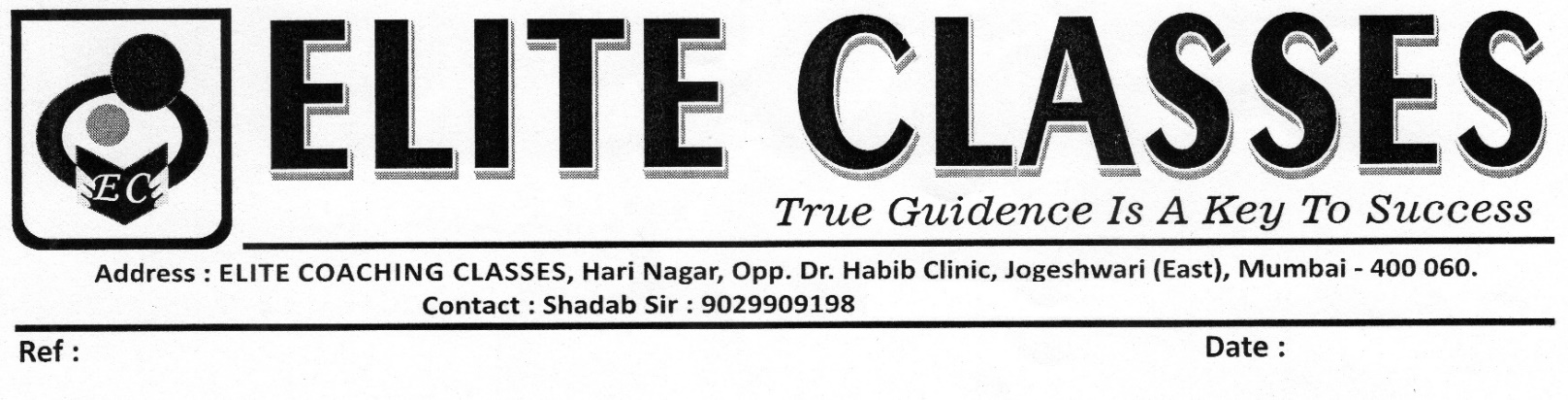
This is to certify that **Mr. Seguro JC**, Seat no. **TYCS 12** **of T. Y. B. Sc. Computer Science** has satisfactorily completed the practical course in **Project Implementation** as prescribed by the University of Mumbai during the academic year **2020 – 2021.**

Signature Signature

**Staff in charge**  **Computer Science Coordinators**

Signature

**Examiner** **College Stamp**

****

**Ref: Date : 1 / 11 / 2020**

**TO WHOMSOEVER IT MAY CONCERN**

This is to confirm that Mr. Seguro JC student of College has developed a software “Connect2Class” for our classes as per requirement given.

Our classes has no objection and permits into carry out the above mention project work at our classes permits.

Prof. NAME

**ACKNOWLEDGEMENT**

I would like to express my sincere gratitude towards the Computer Science Department of College.

After months of hard work, finally I am very happy to present my Sem 5 Project. The Project making was full of new experiences and learning and difficult one too. Though a difficult job it was made simpler by the timely guidance received, which helped me greatly in the completion of my project. But it wouldn’t be right to do so without thanking to those who have helped me in converting our thought into reality. So I would like to take full advantage of this opportunity to thank each and every person who has helped me throughout the completion of our project.

I am obliged to my parents & family members who always support me greatly and encouraged me in each and every step. I give my special thanks and sincere gratitude towards the Principal NAME and Head of Department (Computer Science) Prof. Mrs. NAME

I owe my sincere thanks to our Project guide Ms. NAME for her constant support and encouragement without which the successful completion of this project would have been impossible. They have been instrumental for making me concentrate and focus my effort in this project. And last but not the least the entire college staff specially NAME

I would like to give thanks to my friends for helping me in my project. Finally I would like to thank each and every individual who was directly or indirectly contributing for this project.

**-Thank you.**

**Index**

|  |  |  |
| --- | --- | --- |
| **Sr. No.** | **Topics** | **Page No.** |
| **1.** | **Preliminary Investigation** |  |
|  | * 1. Organizational Overview/Introduction |  |
| * 1. Description of System |  |
| * 1. Limitations |  |
| * 1. Proposed system and Number of Modules |  |
| * 1. Requirement Specification: software/hardware/data requirement |  |
| * 1. Gantt Chart |  |
|  |  |
| **2.** | **System Analysis** |  |
|  | * 1. Fact Finding Techniques (Questionnaire, Sample Reports, Forms...) |  |
| * 1. Event Table |  |
| * 1. Use Case Diagram |  |
| * 1. ERD |  |
| * 1. Class diagram |  |
| * 1. Object Diagram |  |
| * 1. Activity Diagram |  |
| * 1. Sequence diagram |  |
|  |  |
| **3.** | **System Design** |  |
|  | * 1. Component Diagram |  |
| * 1. Package Diagram |  |
| * 1. Deployment Diagram |  |
|  |  |
| **4.** | **System Coding** |  |
|  | * 1. Menu Tree |  |
| * 1. List of tables with attributes and constraints |  |
| * 1. Program Description [ Programs /Classes and their responsibilities in brief ] with Naming Conventions. |  |
| 4.5 Validations |  |
| 4.6 Test Cases, Test Data and Test Results |  |
| 4.7 Screen Layouts & Report Layouts |  |
|  |  |
| **5.** | **Conclusion & Future Scope** |  |
|  |  |  |
| **6.** | **References, Books, web links, research articles, etc.** |  |

**COLLEGE NAME**

**DEPARTMENT OF COMPUTER SCIENCE**

***Project Implementation***

**PHASE COMPLETION CHART**

**CLASS : T.Y.B.Sc. COMPUTER SCIENCE**

**ROLL NO. : ROLLNO**

**STUDENT NAME : SEGURO JC**

**TITLE OF THE PROJECT: CONNECT2CLASS**

|  |  |  |  |
| --- | --- | --- | --- |
| Phase Title | Expected  Date of  Completion | Actual Time of Completion with Guide’s Signature | Remarks |
| I. Preliminary investigation |  |  |  |
| 1.1. Organizational Overview/Introduction  1.2. Description of System  1.3. Limitations  1.4. Proposed system and Number of Modules  1.5. Requirement Specification: software/hardware/data requirement  1.6. Gantt Chart | 24/7/20  24/7/20  24/7/20  24/7/20  24/7/20  6/10/20 |  |  |
| II. System Analysis |  |  |  |
| 2.1. Fact Finding Techniques (Questionnaire, Sample Reports, Forms...)  2.2. Event Table  2.3. Use Case Diagram  2.4. ERD  2.5. Class diagram  2.6. Object Diagram  2.7. Activity Diagram  2.8. Sequence diagram | 28/7/20  1/9/20  7/8/20  4/8/20  18/8/20  18/8/20  18/8/20 |  |  |
| III. System Design |  |  |  |
| 3.1 Component Diagram  3.2 Package Diagram  3.3 Deployment Diagram | 25/8/20  1/9/20  1/9/20 |  |  |
| IV. System Coding |  |  |  |
| 4.1 Menu Tree  4.2 List of tables with attributes and constraints  4.3 Program Description [ Programs /Classes and their responsibilities in brief ] with Naming Conventions.  4.5 Validations  4.6 Test Cases, Test Data and Test Results  4.7 Screen Layouts & Report Layouts | 25/8/20  11/8/20  22/9/20  29/9/20  29/9/20  6/10/20 |  |  |
| V. Project Report Submission | 3/12/20 |  |  |

Note:

Mrs. NAME

Co-ordinator

Computer Science

**LIST OF TABLES:**

|  |  |
| --- | --- |
| **Sr.no** | **Table name** |
| **1** | **Use case description table** |
| **2** | **Event table** |
| **3** | **Validation table** |
| **4** | **Test case table** |

**LIST OF FIGURES:**

|  |  |
| --- | --- |
| **Sr.no** | **Figure name** |
| **1** | **E- R diagram** |
| **2** | **Class diagram** |
| **3** | **Object diagram** |
| **4** | **Use case diagram** |
| **5** | **Activity diagram** |
| **6** | **Sequence diagram** |
| **7** | **Deployment diagram** |
| **8** | **Component diagram** |
| **9** | **Menu tree** |

**1.**

**PRELIMINARY INVESTIGATION**

**1. Preliminary Investigation**

* 1. **Organization Overview:**
* The Organization was established in the year 2003 by Owner Name.
* The Classes is situated in Address
* The name of the classes is Elite Classes.
* The main objective of the Classes Management System is to manage the details of students, teachers, subjects, class and fees etc.

**1.2 Description Of System :**

* The present system consists of a register for managing details of students, teachers, subjects, class and fees etc.
* All records i.e. student, teacher, fees detail etc. are kept manually in various registers and this data are retrieved through records.
* The receipt is made on paper.
* There is no computerized system for various procedures
* The calculation are manually, not computerized

**1.3 Limitation Of System :**

* It is very difficult work to keep all records like student details, teacher details, subject Information manually.
* The present system is very unreliable and hence needs to be computerized.
* In current system a lot of paperwork is involved.
* Also, manual work is always prone to error and errors in such work can have pretty bad consequences.
* To make a monthly balance sheet one has to refer many registers. In case of loss of registers, all the data are lost.
* The present system is very unreliable and hence needs to be compute.

**1.4 Proposed System and its Advantage:**

* It provides flexibility to the user to input the data on a local machine its very easily done by DB to manage the large amount of file. It should also identify the user and admin according to the prescribed level of security. In this system the data will be view by using table and interactive GUI.

**Advantage**

* It is easily to handle lots of data and managing the students, teacher and marks records, apart from registers, books its costs a lot, when there is large amount of data.
* The out sourcing to the local machine also helps in reducing the maintenance.
* It reduces the chance of losing data. Not cheating the owner.

**Number of modules in the proposed system**

* **Login & Register module:** Using this module user enters username and password and the system check whether it is valid. If it is valid user can login, otherwise invalid username and password message is displayed. If the user not have username and password than user also create his/her user account.
* **Home (Dashboard):** Using this module user can easily interact with the system and user other module easily, basically it’s a grid menu for viewing all the modules.
* **Student:** In this module all thestudent details will shown ,filter search is also available and different function available like add student, update student and delete student.
* **Teacher:** In this module all the faculty details will shown and different function is available like add faculty, update faculty and delete faculty.
* **Subject:** In this module which standard has how many subject, total chapter and marks of subject is shown by selecting standard and different function available like Add subject, Update subject, Delete subject.
* **Report:** This module will show all the record of students, teacher, subject, fees and invoice.
* **Fees:** This module is used to generate fee receipt and shows fees amount as per the standard.

**1.5 Feasibility Study:**

The project feasibility is concern with expected benefits. An important outcome of the preliminary investigation is the destination that the proposed system is feasible.

**The types of feasibility study of this system are :**

**a. Technical Feasibility:** This project is technically feasible, software wise it shall be made by using a platform independent IDE which is freely available software and publicly available. We will provide IDE and MySQL to client for managing the desktop applications.

**b. Operational Feasibility:** This project is operationally feasible, easy to handle with basic input, the user interface is easily interact by user. The Operation of adding, updating, deleting is easy to handle. The database connectivity is provided by Developer, and admin also provide information to make changes easily.

**c. Economic Feasibility:** This project requires a storage of 50GB for first 100 records. The Hardware Requirement to run this software is minimum i3 3gen+ processor series, In AMD A4 series+, Ryzen3+ series and minimum ram is 4gb to run smoothly. The running software is freely available. This s necessarily required to run the software. The software design and developed cost is not to be taken and its freely available currently.

**1.6 Stakeholder’s :**

Stakeholder’s are the people who have an interest in the successful implementation of system. We categorize them into following groups .

**1) Technical staff :**

It includes those users who must ensures that the system operators

In computing environment of organization. This particular Application for

which the software is designed is a small scale organization.

**2) User :**

User of those who actually use the system on frequent basis and those are the end users of the system. These are the people who need Current information from the system and uses the system for their purpose.

**1.7 Technologies used :**

**Hardware Requirements**

* + Intel i3 processor
  + Minimum of 2 GB RAM
  + 50 GB Disk Space
  + Windows 7 or higher version

**Software Requirements**

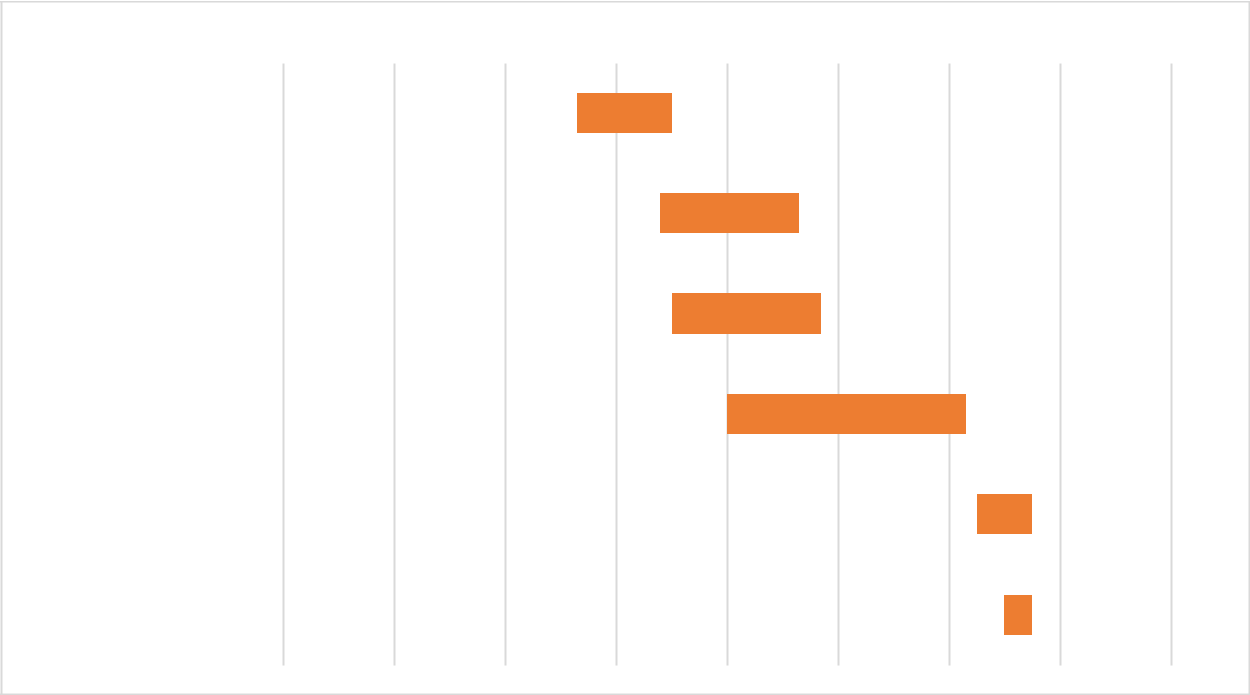
* Front End

Netbeans 8.2 and JDK 8u271

* Back End

MySQL 8.0.2 Command Client and MySQL Workbench 8.0 CE

**1.8 Gantt Chart:**



15-Jul-20 5-Aug-20 25-Aug-20 15-Sep-20 5-Oct-20 25-Oct-20 5-Nov-20 25-Nov-20 30-Nov-20

Preliminary investigation

Analysis

Design

Coding

Testing

Deployment

**2.**

**SYSTEM ANALYSIS**

**2. System Analysis**

**2.1 Facts Finding Techniques:**

**Questionnaire:**

1. Why you choose a custom design software and What’s your requirement?
2. What kind of data you want to manage, and what kind of operation work you want?
3. What is the amount of data you want to manage or basically what's the total number of students, teacher record you want to manage?
4. What kinds of function you need to manage?
5. What kind of Environment you like to use Web Application, Desktop Application or Android Application?
6. Do you need some user like admin /normal user to manipulate the management system?
7. Do you need some extra features?
8. Do you have computer machine, you require this \_\_\_ amount of storage space to run this software?

**2.2 EVENT TABLE:**

**TRIGGER:**

Trigger an occurrence that tells the system that has occurred, either the arrival of data needing or of a point in time

**SOURCE:**

An external agent or actor that supplies data to the system

**ACTIVITY:**

Behavior that the system performs when an event occurs

**RESPONSE:**

An output produced by the system that goes to a destination.

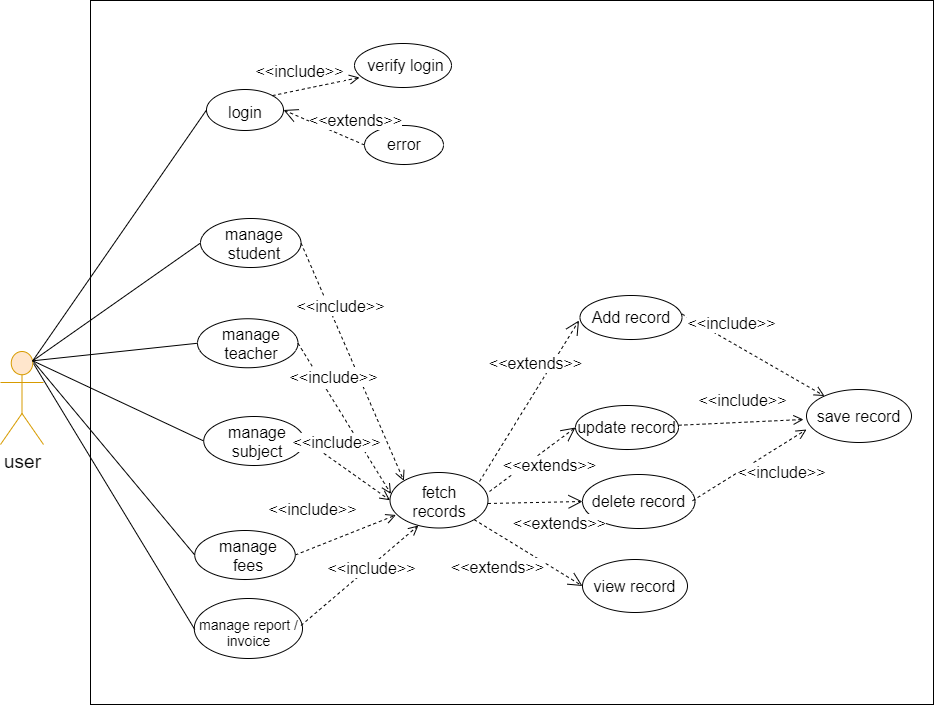
**DESTINATION:**

An external agent or actor that receives data from the system

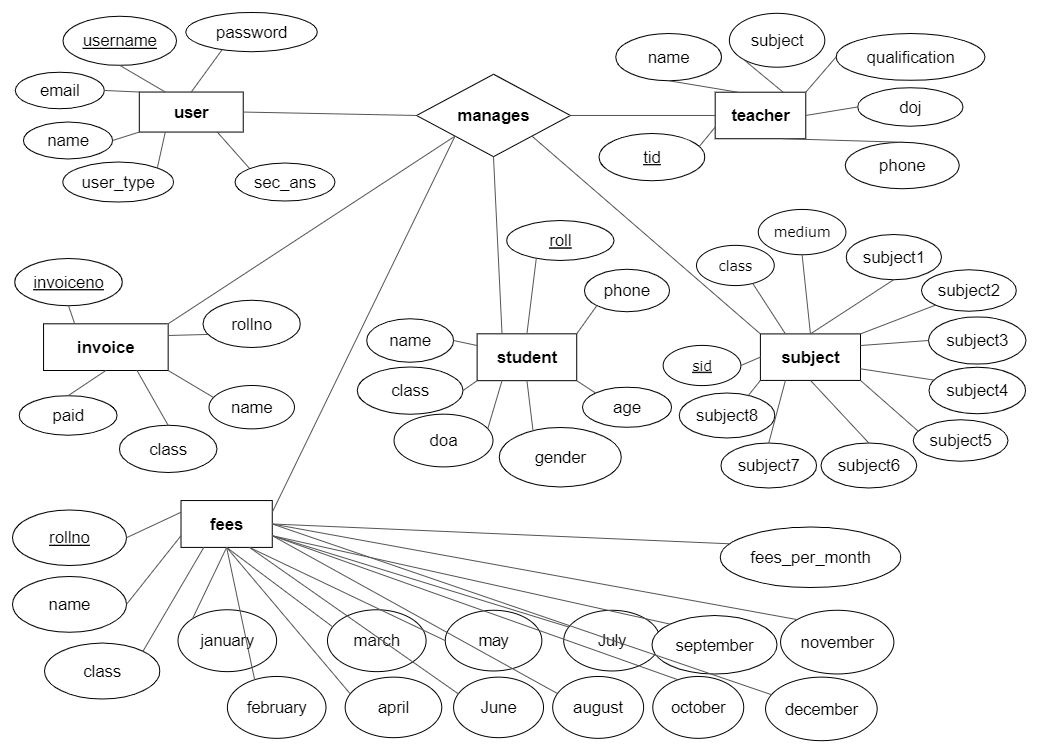
**Table: 1. Login/Signup Use Case**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | |  | | | **Use Case Name** |  | | Log-in/Sign-in |
| **Summary/Description** | Use case depicting how an user would login |
| **Actors** | admin |
| **Pre-Conditions** | Log-in ID must be present in user table for user |
| **Descriptions** | User will have to login after they land on dashboard From here they can perform their respective actions like Add, Update, Delete,View (Student, Teacher, Subject, Fees, Invoice, Report) |
| **Exception** | No ID present in the Login table, which will give them an error |
| **Post-Condition** | User successfully Login. |

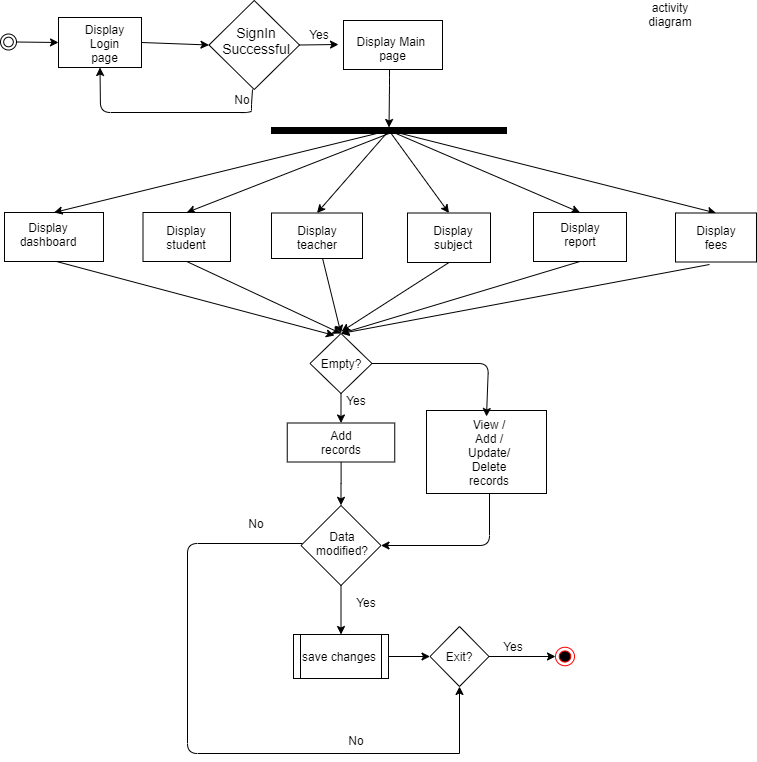
**2.3 Use Case Diagram:**



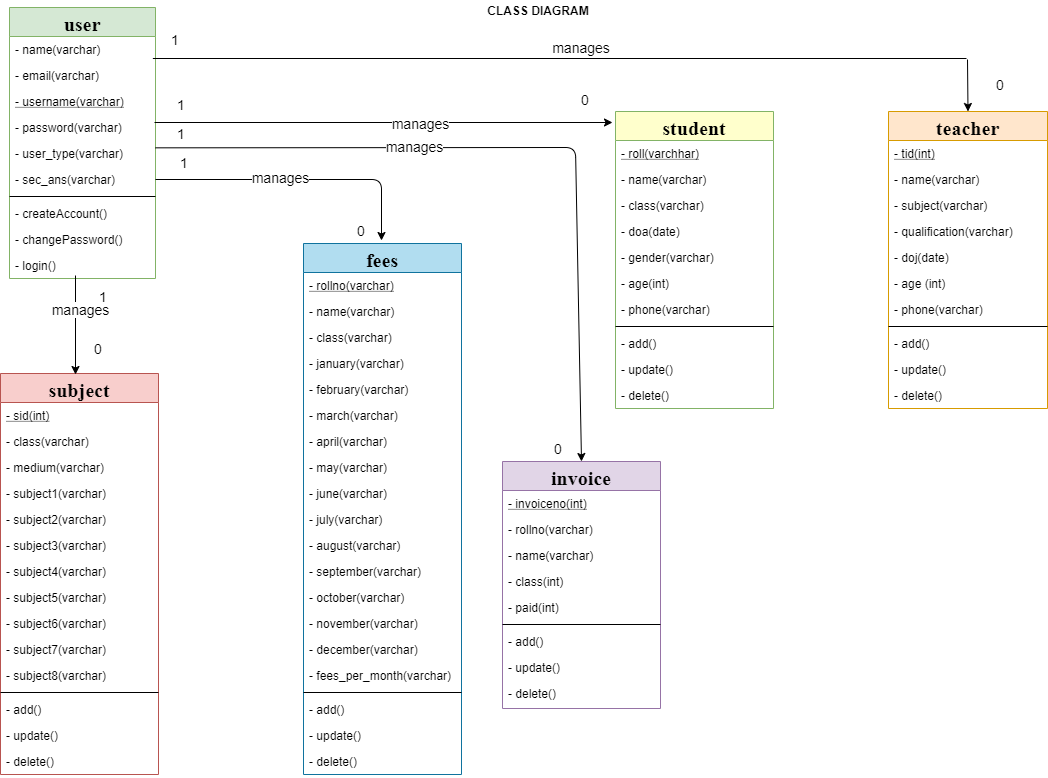
**2.4 ER Diagram:**



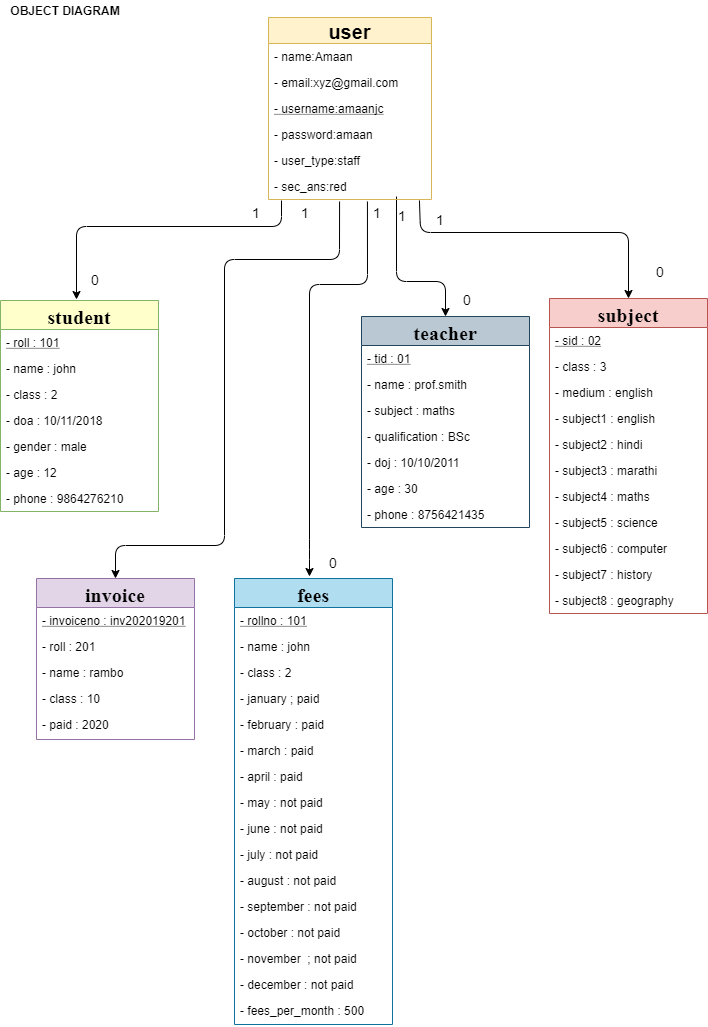
**2.5 Activity Diagram:**



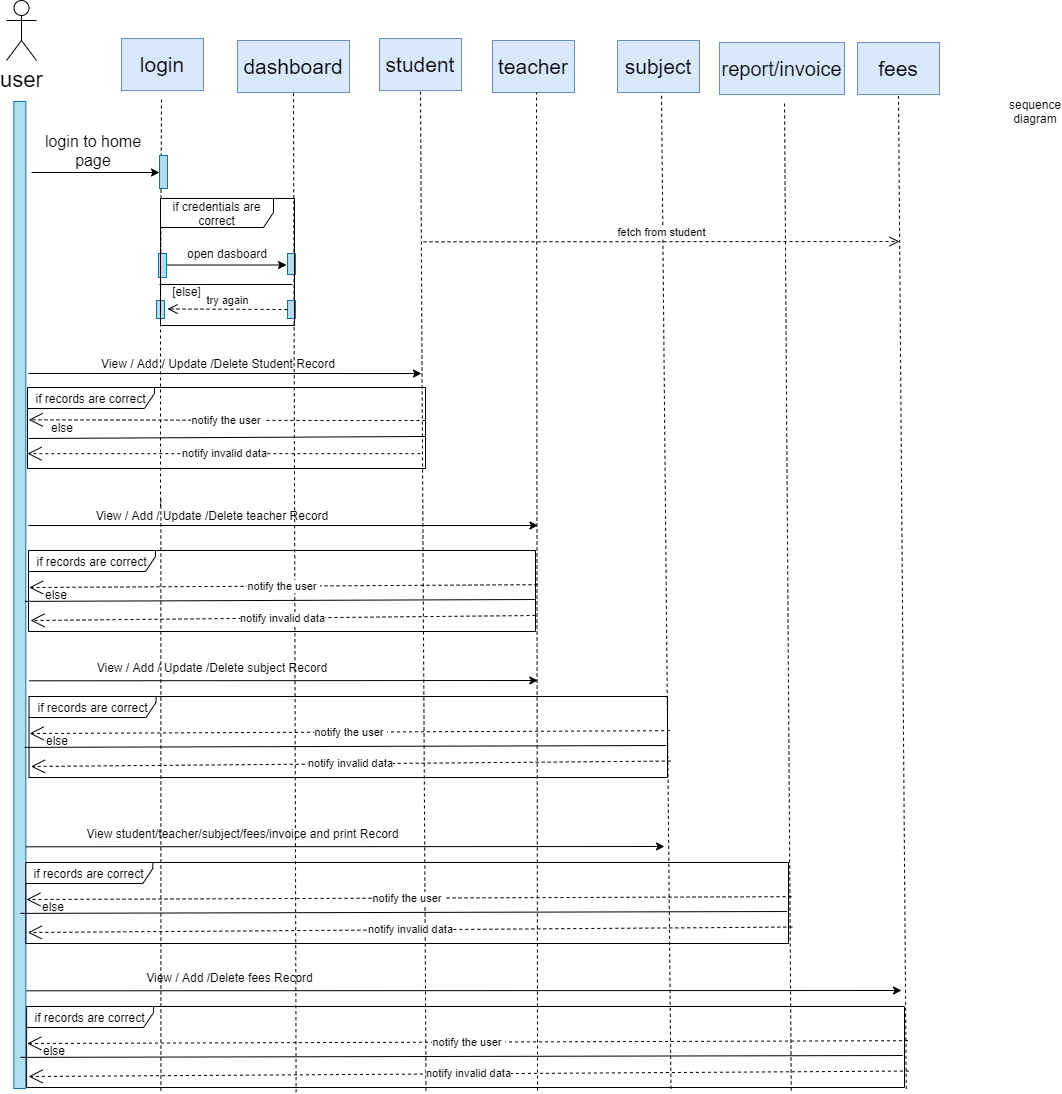
**2.6 Class Diagram:**



**2.7 Object Diagram:**



**2.8 Sequence Diagram:**

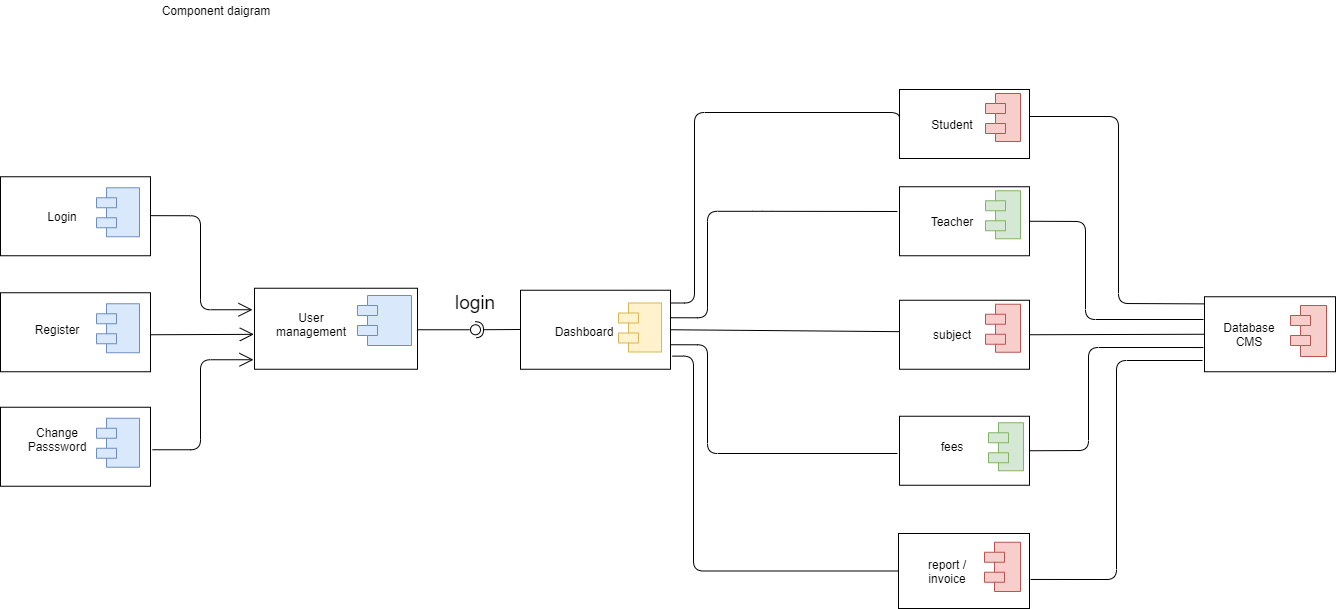


**3 .**

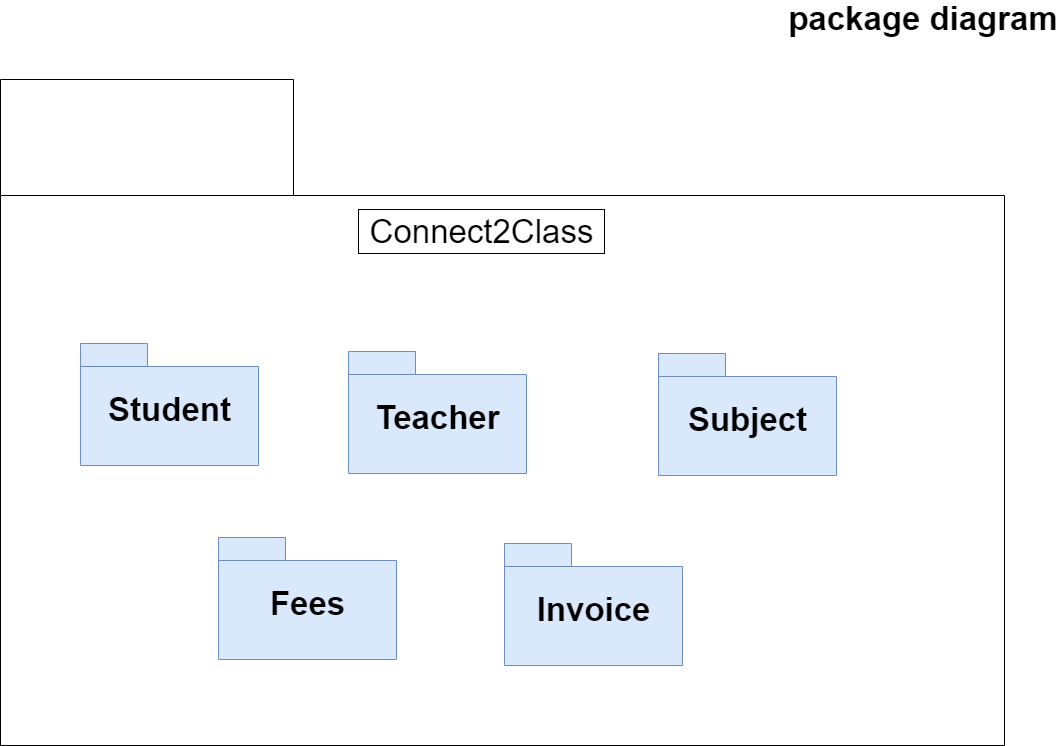
**SYSTEM DESIGN**

**3. System Design**

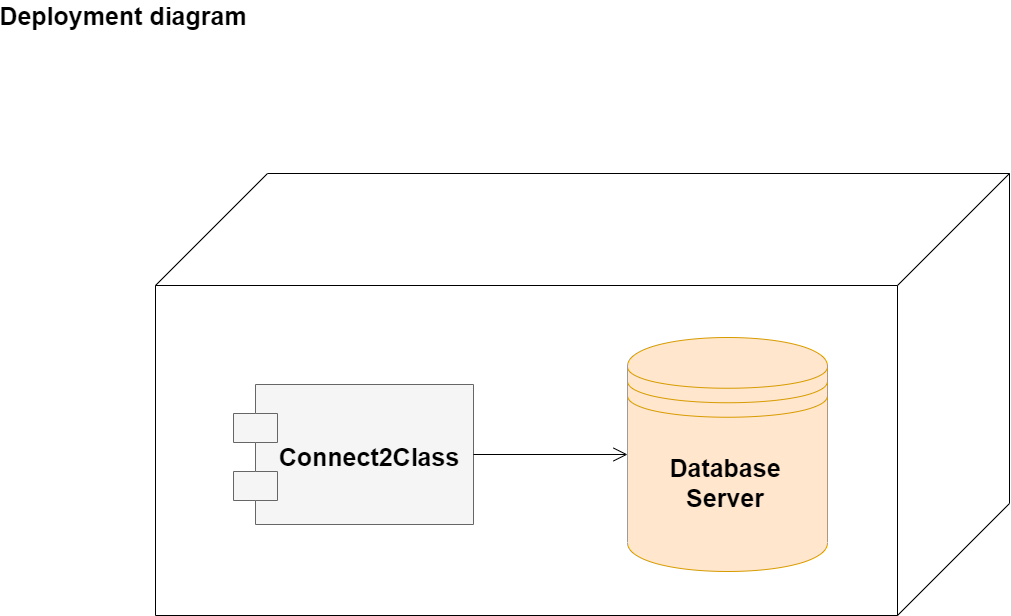
**3.1 Component Diagram:**



**3.2 Package Diagram:**



**3.3 Deployment Diagram:**

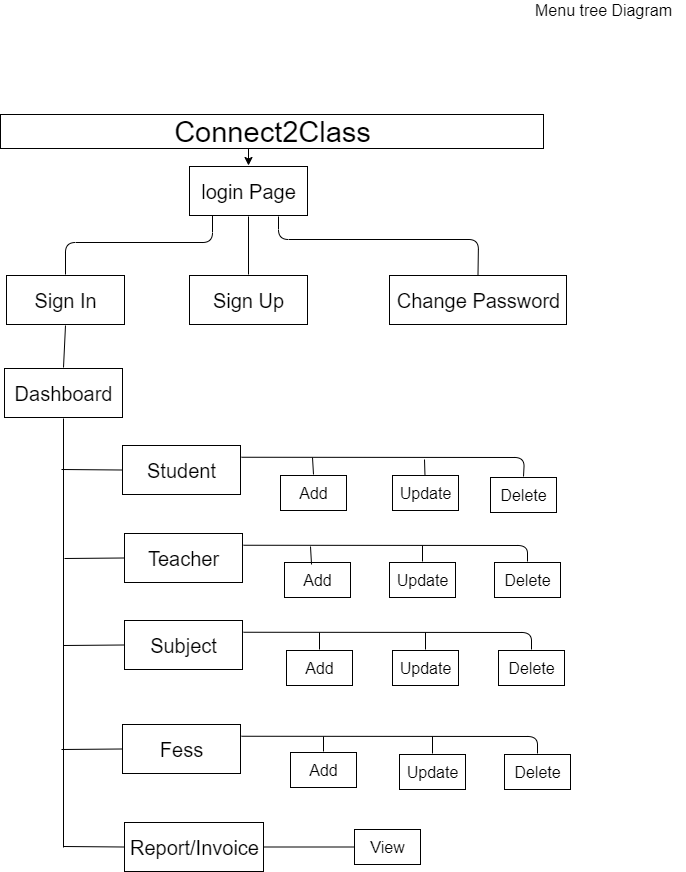


**4.**

**SYSTEM CODING**

**4. System Coding**

**4.1 Menu Tree:**



**4.2 List of tables with attributes and constraints:**

**1.Table Name : User**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Sr.No.** | **Attribute Name** | **Attribute Type** | **Sample Data** | **Data type allowed** | **Description** |
| 1 | Name | varchar(30) | Amaan | Varchar, String, char | Name of user |
| 2 | email | Varchar(50) | xyz@gmail.com | Varchar, String, char | Email of user |
| 3 | username | varchar(30) | amaanjc | Varchar, String, char | Name of user |
| 4 | password | Varchar(30) | ajc1234 | Varchar, String, char | password |
| 5 | user\_type | varchar(30) | admin/staff | Varchar, String, char | User type |
| 6 | sec\_ans | varchar(30) | red | Varchar, String, char | Security answer |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Sr.No.** | **Attribute Name** | **Attribute Type** | **Sample Data** | **Data type allowed** | **Description** |
| 1 | roll | Varchar(30) | 101 | Integer, Number | Rollno of student |
| 2 | name | varchar(30) | Amaan | Varchar, String, char | Name of student |
| 3 | class | varchar(5) | 5 | Varchar, String, char | Class of student |
| 4 | doa | Date | 2019-07-31 | datetime | Date of admission |
| 5 | gender | varchar(10) | male/female | Varchar, String, char | Gender |
| 6 | age | integer | 12 | Integer, Number | Age of student |
| 7 | phone | integer | 9867801364 | Integer, Number | Phone no. |

**2. Table Name : Student**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Sr.No.** | **Attribute Name** | **Attribute Type** | **Sample Data** | **Data type allowed** | **Description** |
| 1 | tid | Int | 1 | Integer | Autogenerated id |
| 2 | name | varchar(30) | Albert | Varchar, String, char | Name of teacher |
| 3 | subject | varchar(100) | Maths,Science | Varchar, String, char | subject teacher |
| 4 | qualification | Varchar(100) | B.com | Varchar, String, char | Qualification of teacher |
| 5 | doj | Date | 2018-07-31 | datetime | Date of joining |
| 6 | phone | integer | 9867801364 | Integer, Number | Phone no. |

**3. Table Name : Teacher**

**4. Table Name : Subject**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sr. No** | **Attribute Name** | **Attribute Type** | **Sample Data** | **Data type allowed** |
| 1 | sid | integer | 1 | Integer, Number |
| 2 | class | Varchar(20) | 1/2/3/4 | Varchar, String, char |
| 3 | medium | Varchar(20) | English/hindi | Varchar, String, char |
| 2 | Subject1 | Varchar(20) | English | Varchar, String, char |
| 3 | Subject2 | Varchar(20) | Hindi | Varchar, String, char |
| 4 | Subject3 | Varchar(20) | Marathi | Varchar, String, char |
| 5 | Subject4 | Varchar(20) | Science | Varchar, String, char |
| 6 | Subject5 | Varchar(20) | Maths | Varchar, String, char |
| 7 | Subject6 | Varchar(20) | History | Varchar, String, char |
| 8 | Subject7 | Varchar(20) | ICT | Varchar, String, char |
| 9 | Subject8 | Varchar(20) | Geography | Varchar, String, char |

**5.Table Name : Invoice table**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Sr.No.** | **Attribute Name** | **Attribute Type** | **Sample Data** | **Data type allowed** | **Description** |
| 1 | invoiceno | integer | 101 | Integer, Number | Invoice no. |
| 2 | rollno | varchar(30) | Amaan | Varchar, String, char | Rollno of student |
| 3 | name | varchar(5) | 5 | Varchar, String, char | Name of student |
| 4 | class | Varchar(10) | 2 | Varchar, String, char | Class of student |
| 5 | paid | integer | 9867801364 | Integer, Number | Phone no. |

**6. Table Name : Fees**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Sr.No.** | **Attribute Name** | **Attribute Type** | **Sample Data** | **Data type allowed** | **Description** |
| 1 | rollno | integer | 101 | Integer, Number | Rollno of student |
| 2 | name | varchar(30) | Amaan | Varchar, String, char | Name of student |
| 3 | class | varchar(5) | 5 | Varchar, String, char | Class of student |
| 4 | January | varchar(10) | Paid/not paid | Varchar, String, char | Student paid their fess or not |
| 5 | february | varchar(10) | Paid/not paid | Varchar, String, char | Student paid their fess or not |
| 6 | march | varchar(10) | Paid/not paid | Varchar, String, char | Student paid their fess or not |
| 7 | april | varchar(10) | Paid/not paid | Varchar, String, char | Student paid their fess or not |
| 8 | may | varchar(10) | Paid/not paid | Varchar, String, char | Student paid their fess or not |
| 9 | june | varchar(10) | Paid/not paid | Varchar, String, char | Student paid their fess or not |
| 10 | july | varchar(10) | Paid/not paid | Varchar, String, char | Student paid their fess or not |
| 11 | august | varchar(10) | Paid/not paid | Varchar, String, char | Student paid their fess or not |
| 12 | september | varchar(10) | Paid/not paid | Varchar, String, char | Student paid their fess or not |
| 13 | october | varchar(10) | Paid/not paid | Varchar, String, char | Student paid their fess or not |
| 14 | november | varchar(10) | Paid/not paid | Varchar, String, char | Student paid their fess or not |
| 15 | december | varchar(10) | Paid/not paid | Varchar, String, char | Student paid their fess or not |
| 16 | fees\_per\_month | integer | 2000 | Integer, Number | Fees amount |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **4.3 Validation**  **1. User** | |  |  |  |
|  |  |  |  |  |
|  | **Field** | **Valid values** | **Invalid values** |  |
|  |  |  |  |  |
|  | Name | Amaan , Seguro | Jammy@1299, 12surya |  |
|  | Email | amn@gmail.com | samko@sh12k, sury2.com |  |
| **2. Student** | |  |  |  |
|  |  |  |  |  |
|  | **Field** | **Valid values** | **Invalid values** |  |
|  |  |  |  |  |
|  | Name | Amaan , Seguro | Jammy@1299, 12surya |  |
|  | Age | 14, 10 | Twelve,$12 |  |
|  | Phone | 9458418426 | Mobile123 |  |
| **3. Teacher** | |  |  |  |
|  |  |  |  |  |
|  | **Field** | **Valid values** | **Invalid values** |  |
|  |  |  |  |  |
|  | Name | Amaan , Seguro | eleven |  |
|  | Age | 18, 20 | $3120 |  |
|  | Phone | 9458418426 | #sam99 |  |

**4.4 Test Cases Module**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **Module** | **Why?** | **Expected** | **Observed** | **Remarks** | |
|  |  |  | **Output** | **Output** |  | |
|  |  |  |  |  |  | |
|  | Login | To check | Only valid login | Only valid login | Pass | |
|  |  | access granted | Credential should | Credential was |  |  |
|  |  | only to | be accepted. | accepted. Invalid |  |  |
|  |  | admin | Invalid should be | was rejected. |  |  |
|  |  |  | rejected. |  |  |  |
|  | Student | For managing | Add/update/delete | student details was | Pass | |
|  |  | Student details | Student successfull | Added ,updated |  |  |
|  |  |  |  | and deleted. |  |  |
|  | Teacher | For managing | Add/update/delete | teacher details was | Pass | |
|  |  | Teacher details | Teacher successfull | Added ,updated |  |  |
|  |  |  |  | and deleted. |  |  |
|  | Subject | For managing | Add/update/delete | subject details was | Pass | |
|  |  | subject details | subject successfull | Added ,updated |  |  |
|  |  |  |  | and deleted. |  |  |
|  | Fees | For managing | Add/update/delete | fees details was | Pass | |
|  |  | fees details | fees successfull | Added ,updated |  |  |
|  |  |  | marked correctly | and deleted. |  |  |
|  | Geneate Receipt | For generating | The receipt must | Student fees was | Pass | |
|  |  | Receipt | Generated | Paid successfully |  |  |
|  |  |  | accurately |  |  |  |
|  | Report | For printing | The table must be | Report table is | Pass | |
|  |  | Report table | printed in pdf format | viewable |  |  |

**4.5 Program Description:**

**Login Code :**

String username = txtuser\_in.getText();

String sql ="select \* from user where username =? and password =?";

String password = String.valueOf(txtpass\_in.getPassword());

if (username.equals("") || password.equals("")) {

JOptionPane.showMessageDialog(rootPane, "Some Fields Are Empty", "Error", 1);

} else {

try {

Connection con = conjdbc.getConnection();

PreparedStatement smt = con.prepareStatement(sql);

smt.setString(1, username);

smt.setString(2, password);

ResultSet st = smt.executeQuery();

if (st.next()) {

JOptionPane.showMessageDialog(null, "You are Successfully logged in");

signup\_pane.setVisible(false);

JFrame frame = new home();

frame.setVisible(true);

dispose();

} else {

JOptionPane.showMessageDialog(null, "Invalid Login Credentials");

}

con.close();

st.close();

} catch (SQLException | ClassNotFoundException ex) {

Logger.getLogger(login.class.getName()).log(Level.SEVERE, null, ex);

}

}

**Add Student Code :**

try {

String sql\_sgender = null;

String roll = txt\_sroll.getText();

String name = txt\_sname.getText();

if (!roll.isEmpty() && !name.isEmpty() && !txt\_sage.getText().isEmpty() && !txt\_sphone.getText().isEmpty()) {

try {

Connection con = conjdbc.getConnection();

PreparedStatement ps = (PreparedStatement) con.prepareStatement("insert into student values(?,?,?,?,?,?,?)");

ps.setString(1, roll);

ps.setString(2, name);

ps.setString(3, cb\_sclass.getSelectedItem().toString());

SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd");

String sql\_sdoa = sdf.format(jdate\_sdoa.getDate());

ps.setString(4, sql\_sdoa);

if (rb\_smale.isSelected()) {

sql\_sgender = "Male";

} else if (rb\_sfemale.isSelected()) {

sql\_sgender = "Female";

}

ps.setString(5, sql\_sgender);

int sql\_sage = Integer.parseInt(txt\_sage.getText());

ps.setInt(6, sql\_sage);

ps.setString(7, txt\_sphone.getText());

int result = ps.executeUpdate();

if (result != 0) {

System.out.println("Student added successfully");

int result2 = JOptionPane.showConfirmDialog(null, "Do You also want to add Student fees?");

if (JOptionPane.YES\_OPTION == result2) {

addstudent\_fees();

}

ps.close();

con.close();

showstudent\_table();

student\_clear();

}

} catch (SQLException ex) {

Logger.getLogger(home.class.getName()).log(Level.SEVERE, null, ex);

} catch (ClassNotFoundException ex) {

Logger.getLogger(home.class.getName()).log(Level.SEVERE, null, ex);

}

} else {

JOptionPane.showMessageDialog(null, "Some Fields are Empty");

}

} catch (Exception ex) {

JOptionPane.showMessageDialog(null, "Please try Again" + ex);

}

**Update Student Code :**

try {

makePanelVisible(addup\_student);

int row = student\_table.getSelectedRow();

String sroll = (student\_table.getModel().getValueAt(row, 0)).toString();

String sql = "select \* from student where roll=" + sroll;

try {

Connection con = conjdbc.getConnection();

PreparedStatement ps = (PreparedStatement) con.prepareStatement(sql);

ResultSet rs = ps.executeQuery();

while (rs.next()) {

txt\_sroll.setText(rs.getString("roll"));

txt\_sname.setText(rs.getString("name"));

String sclass;

sclass = (rs.getString("class"));

cb\_sclass.setSelectedItem(sclass);

String sdoa = (rs.getString("doa"));

Date doa = new SimpleDateFormat("yyyy-MM-dd").parse(sdoa);

jdate\_sdoa.setDate(doa);

String gender = (rs.getString("gender"));

if (gender.equals("Male")) {

rb\_smale.setSelected(true);

} else if (gender.equals("Female")) {

rb\_sfemale.setSelected(true);

}

//JOptionPane.showMessageDialog(null,gender);

txt\_sage.setText(rs.getString("age"));

txt\_sphone.setText(rs.getString("phone"));

}

rs.close();

ps.close();

con.close();

} catch (ClassNotFoundException | SQLException e) {

JOptionPane.showMessageDialog(null, e);

} catch (ParseException ex) {

Logger.getLogger(home.class.getName()).log(Level.SEVERE, null, ex);

}

} catch (Exception e) {

JOptionPane.showMessageDialog(null, "You need to select table row to update student data");

addup\_student.setVisible(false);

student.setVisible(true);

System.out.println(e);

}

try {

String sroll = txt\_sroll.getText();

String sname = txt\_sname.getText();

SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd");

String sql\_sdoa = sdf.format(jdate\_sdoa.getDate());

String sgender = null;

int sage = Integer.parseInt(txt\_sage.getText());

String sphone = txt\_sphone.getText();

String sql = "update student set roll=?,name=?,class=?,doa=?,gender=?,age=?,phone=? where roll=" + sroll;

if (!sroll.isEmpty() && !txt\_sname.getText().isEmpty() && !txt\_sage.getText().isEmpty() && !txt\_sphone.getText().isEmpty()) {

try {

Connection con = conjdbc.getConnection();

PreparedStatement ps = (PreparedStatement) con.prepareStatement(sql);

ps.setString(1, sroll);

ps.setString(2, sname);

ps.setString(3, cb\_sclass.getSelectedItem().toString());

ps.setString(4, sql\_sdoa);

if (rb\_smale.isSelected()) {

sgender = "Male";

} else if (rb\_sfemale.isSelected()) {

sgender = "Female";

}

ps.setString(5, sgender);

ps.setInt(6, sage);

ps.setString(7, sphone);

int rs = ps.executeUpdate();

if (rs != 0) {

JOptionPane.showMessageDialog(null, "Student Data Updated successfully");

int result2 = JOptionPane.showConfirmDialog(null, "Do You also want to add Student fees?");

if (JOptionPane.YES\_OPTION == result2) {

String sroll2 = txt\_sroll.getText();

String sql2 = "update fees set rollno=?,name=?,class=? where rollno=" + sroll2;

String sname2 = txt\_sname.getText();

try {

Connection con2 = conjdbc.getConnection();

PreparedStatement ps2 = (PreparedStatement) con2.prepareStatement(sql2);

ps2.setString(1, sroll2);

ps2.setString(2, sname2);

ps2.setString(3, cb\_sclass.getSelectedItem().toString());

int rs2 = ps2.executeUpdate();

if (rs2 != 0) {

ps2.close();

con2.close();

showfees\_table();

} else {

addstudent\_fees();

}

} catch (Exception e) {

JOptionPane.showMessageDialog(null, e);

}

}

ps.close();

con.close();

showstudent\_table();

student\_clear();

} else {

JOptionPane.showMessageDialog(null, "Updation Unsuccessfull");

}

} catch (HeadlessException | ClassNotFoundException | SQLException e) {

JOptionPane.showMessageDialog(null, e);

}

} else {

JOptionPane.showMessageDialog(null, "Try Again");

}

} catch (Exception ex) {

JOptionPane.showMessageDialog(null, "Please try Again" + ex);

}

**Delete Student Code :**

int result = JOptionPane.showConfirmDialog(null, "Do You want to Delete Student?");

if (JOptionPane.YES\_OPTION == result) {

int result2 = JOptionPane.showConfirmDialog(null, "Do You also want to Delete Student fees?");

try {

int row = student\_table.getSelectedRow();

String cell = student\_table.getModel().getValueAt(row, 0).toString();

String sql = "delete from student where roll=" + cell;

try {

Connection con = conjdbc.getConnection();

PreparedStatement ps = con.prepareStatement(sql);

ps.execute();

JOptionPane.showMessageDialog(null, "Student Data Deleted Successfully");

con.close();

ps.close();

showstudent\_table();

if (JOptionPane.YES\_OPTION == result2) {

String sql1 = "delete from fees where rollno=" + cell;

try {

Connection con1 = conjdbc.getConnection();

PreparedStatement ps1 = con1.prepareStatement(sql1);

ps1.execute();

JOptionPane.showMessageDialog(null, "Student Fees Data Deleted Successfully");

con.close();

ps.close();

showfees\_table();

} catch (Exception ex) {

JOptionPane.showMessageDialog(null, ex);

}

}

} catch (HeadlessException | ClassNotFoundException | SQLException e) {

JOptionPane.showMessageDialog(null, e);

JOptionPane.showMessageDialog(null, "Deletion Unsuccessfull");

}

student\_clear();

} catch (Exception e) {

JOptionPane.showMessageDialog(null, "You need to select table row to delete student data");

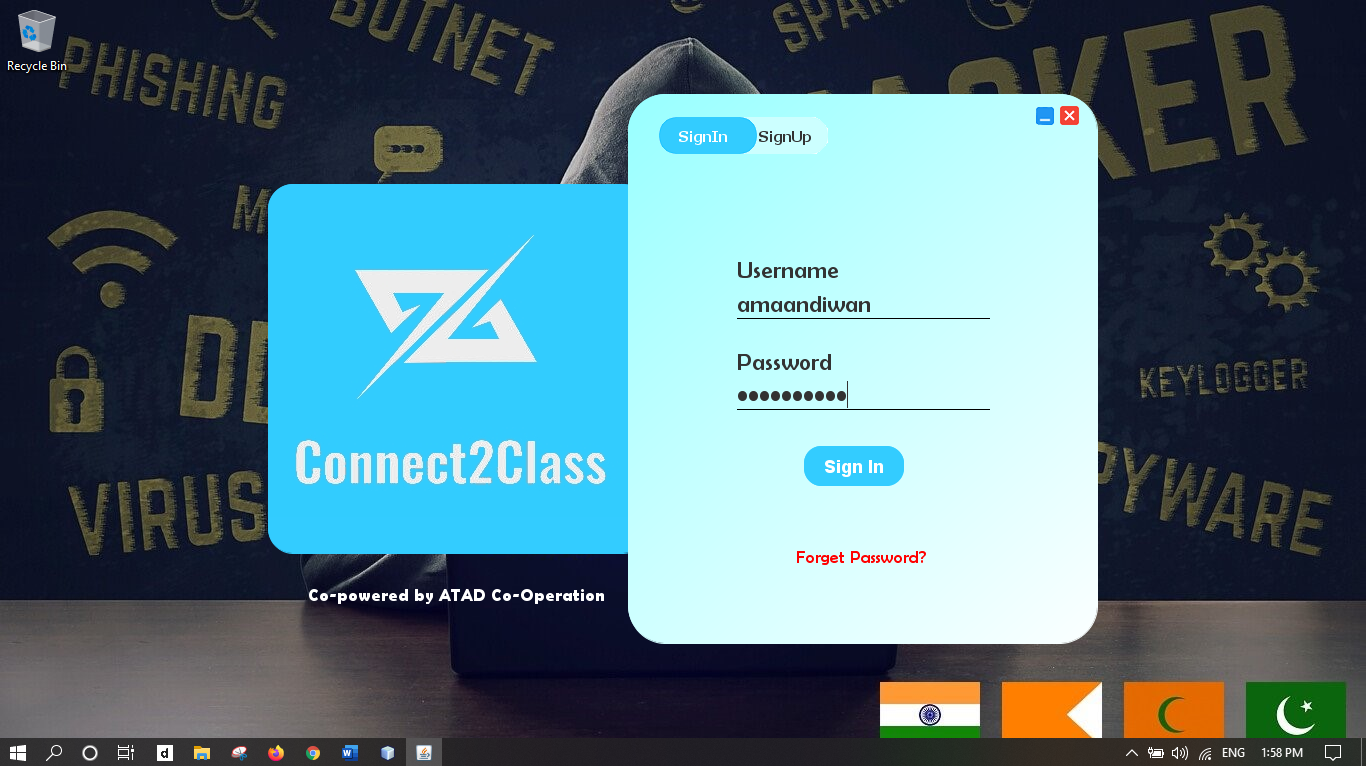
System.out.println(e);

}

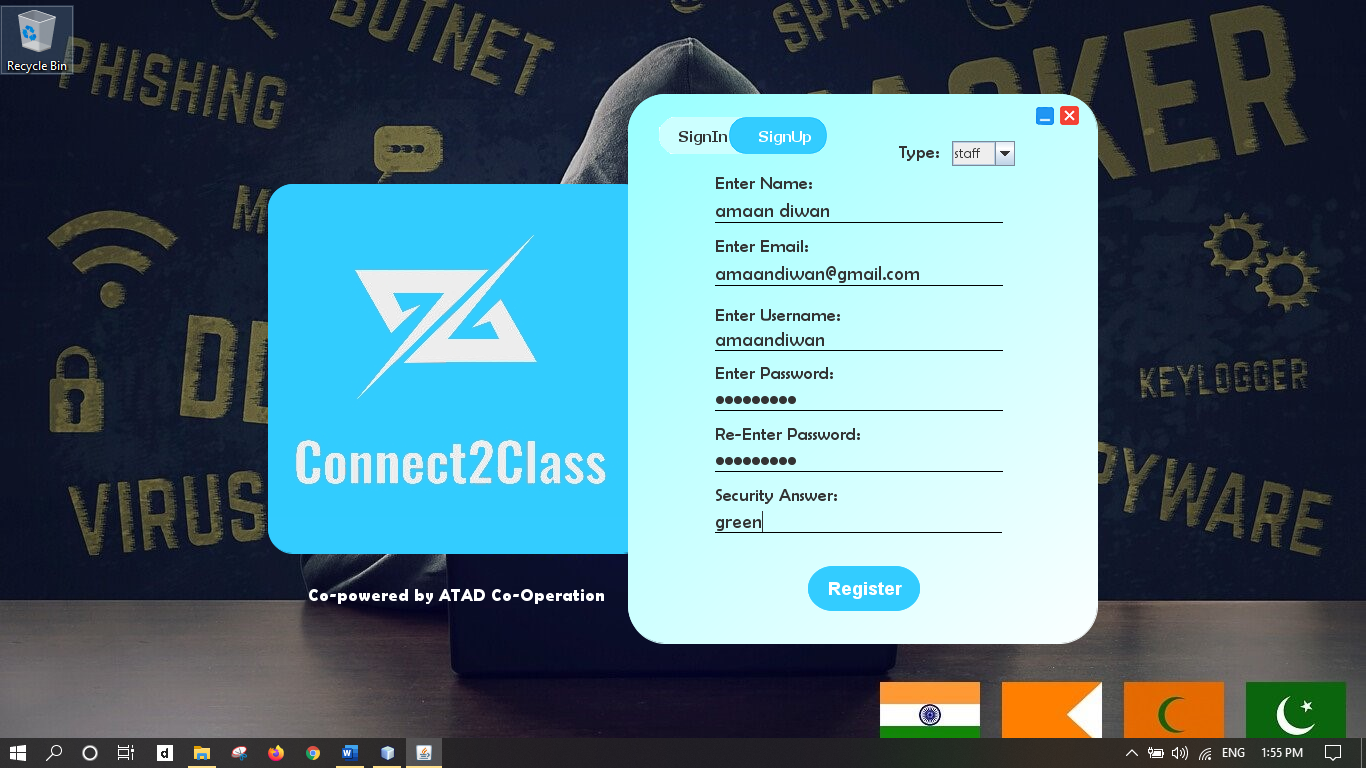
}

**4.6 Screen Layouts & Report Layouts**

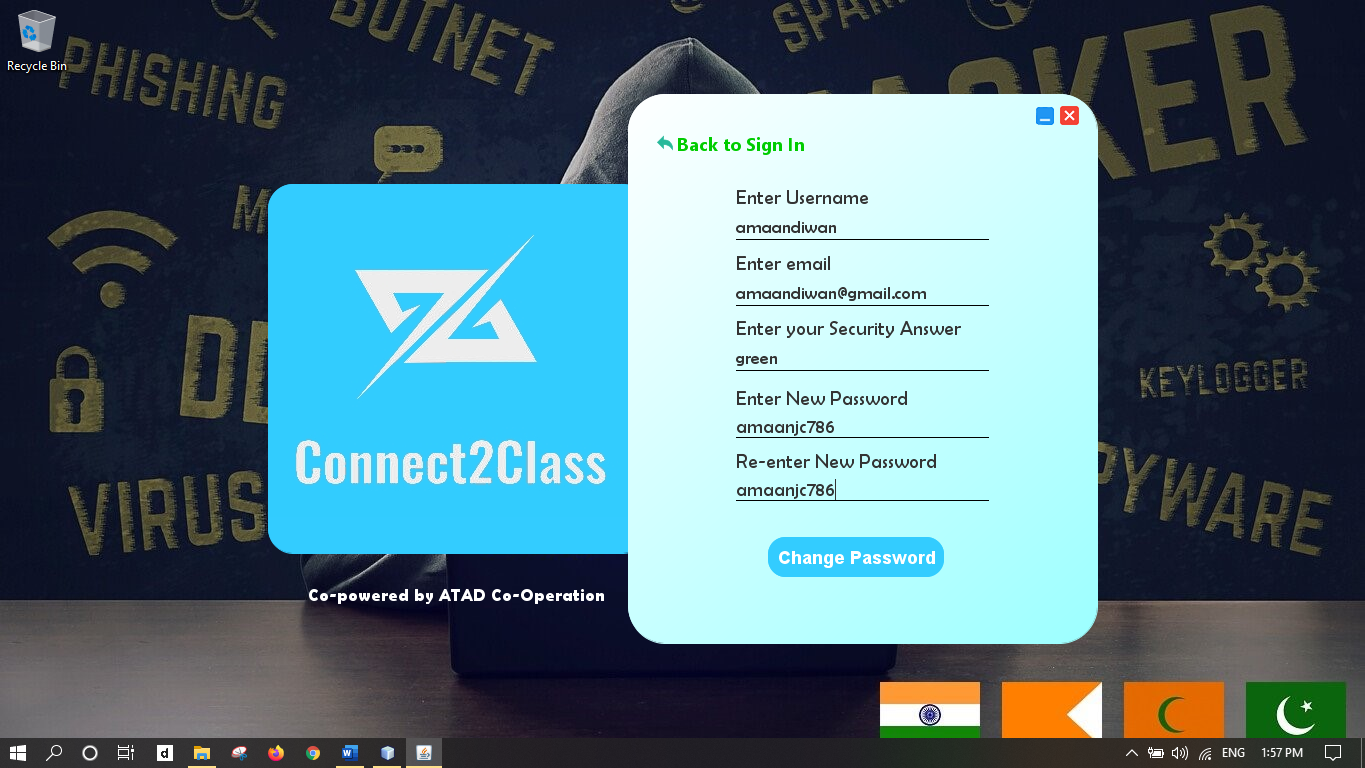
**Sign In Page**



**Sign Up Page**



**Forget Password Page**





Generated Receipt with invoice number

ATTACH RECEIPT SCREENSHOT HERE

1. **Future Enhancement**

* If there is any rectification or enhancement proposed by the user, the software will be modified and the desired module will be implemented into the current system
* If there is any change in the business process , the current system will be made to make provisions for such changes under the administrator power
* The current system is desktop application based , website of the system can made in future.

1. **Conclusion**

* A desktop application for mobile shop was created using java & MySql . The desktop application can be used in the module shop for keeping records of product details & calculate total sales done.
* Also it can be used to keep record of employee attendance and their salary.
* By using this desktop application, a lot of burden of keeping records would be lesser as we can have everything saved onto our databases.

1. **References**
2. [www.youtube.com](http://www.youtube.com/) (keeptoo, SynTech,etc)
3. https://github.com/akash-1618 (@Akash Pawar)
4. <https://www.instagram.com/tancredshaikh90/> (@Tanveer Shaikh)
5. <https://www.instagram.com/zeeshanboi/> (@Zeeshan Mansoori)