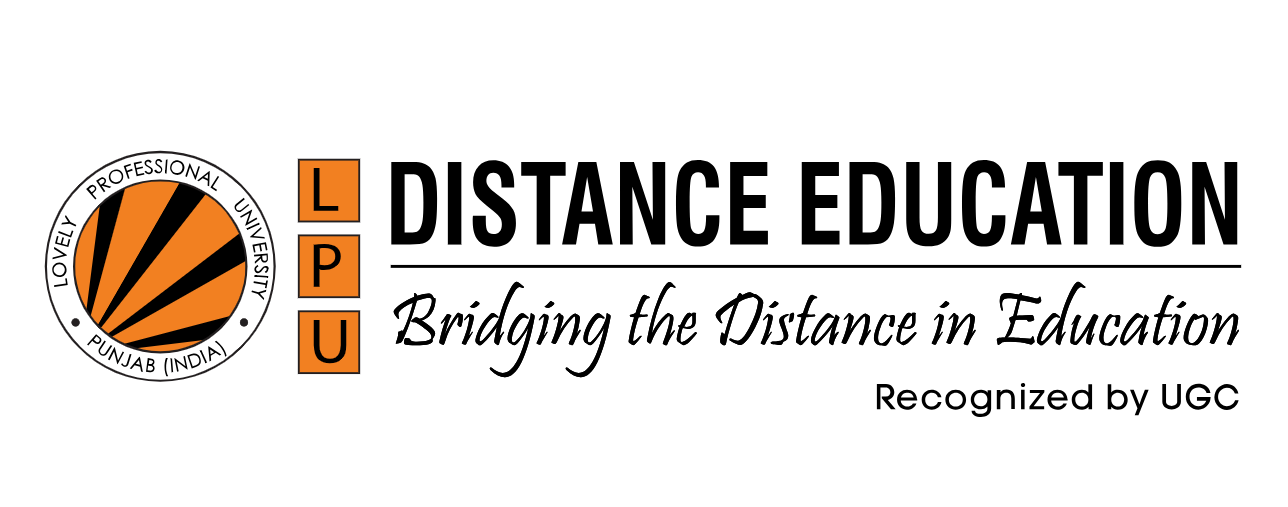
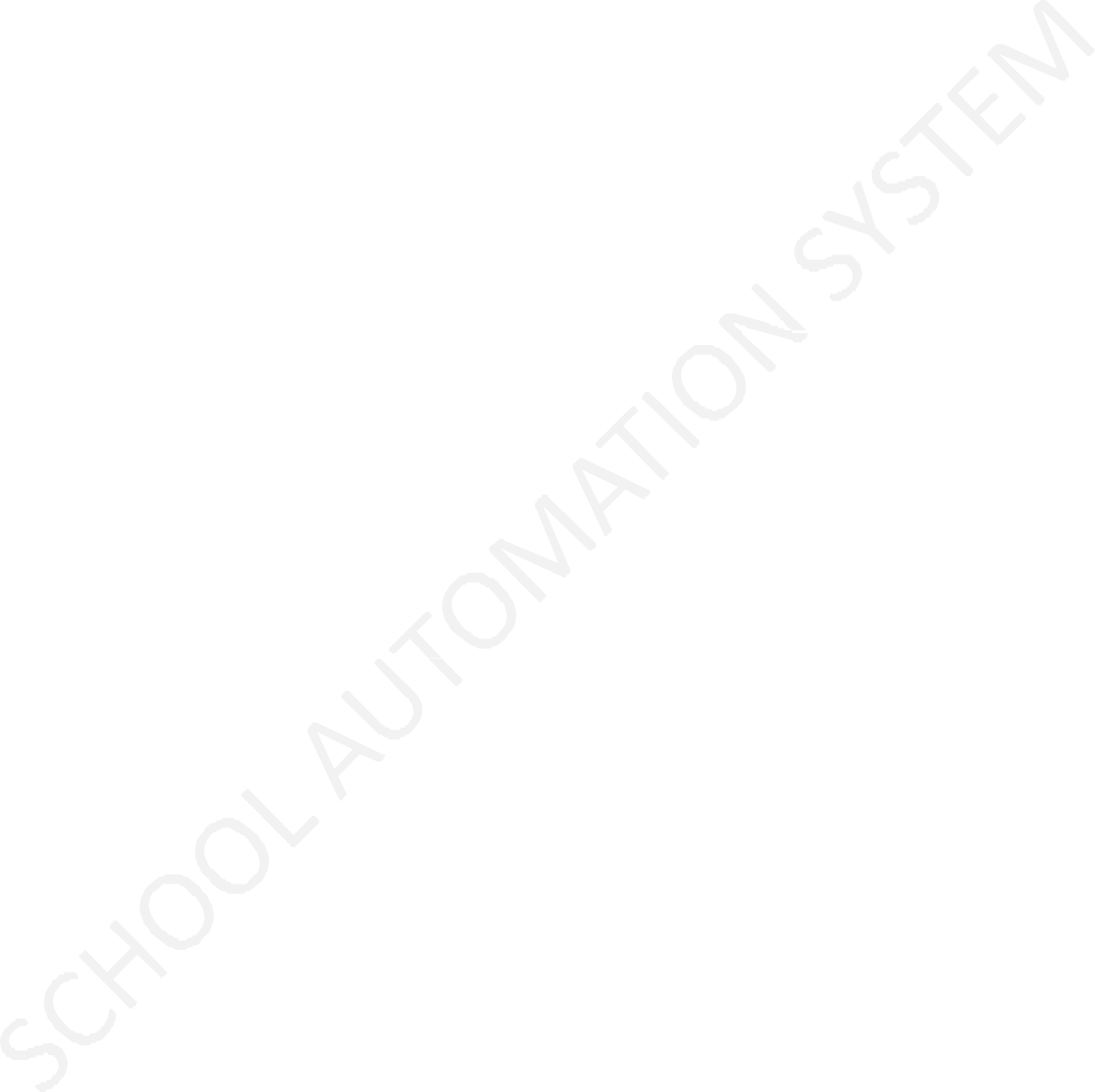
****

**Project Report**

on

School Automation System

SUBMITTED TO

# LOVELY PROFESSIONAL UNIVERSITY

IN PARTIAL FULFILLMENT OF

THE REQUIREMENTS FOR THE AWARD OF THE DEGREE OF

**MASTERS OF COMPUTER APPLICATION ( MCA )**

Submitted By:

Raghav Vehgal

Programme: 1624-S

Batch:

Under the Guidance of

Name of Project Guide: Sumit Jaiswal

Designation: Associate Professor

**LOVELY FACULTY OF DISTANCE EDUCATION**

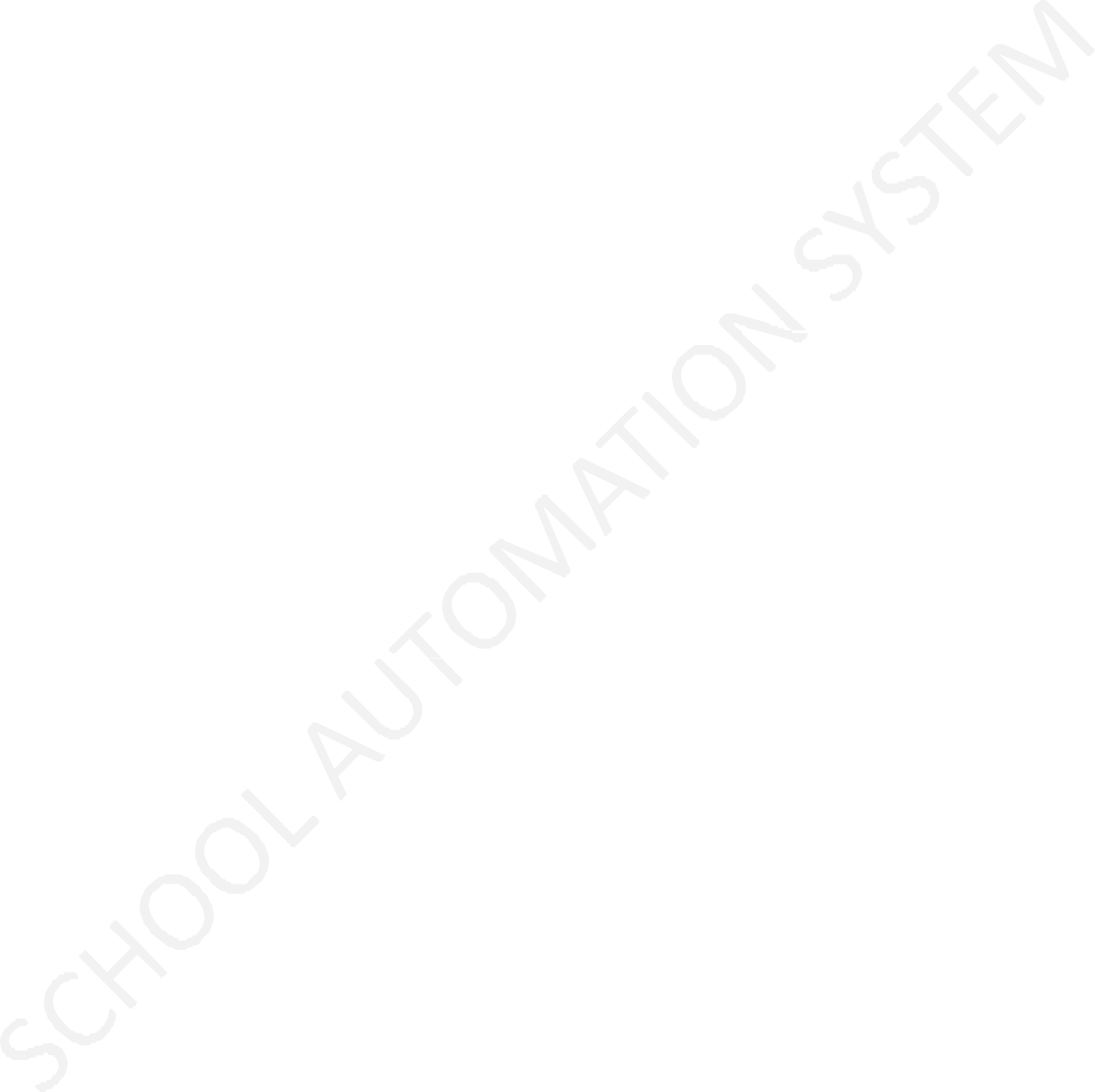
**LOVELY PROFESSIONAL UNIVERSITY, PHAGWARA**

**2024**

**Project Report**

on

School Automation System

SUBMITTED TO

# LOVELY PROFESSIONAL UNIVERSITY

IN PARTIAL FULFILLMENT OF

THE REQUIREMENTS FOR THE AWARD OF THE DEGREE OF

**MASTERS OF COMPUTER APPLICATION ( MCA )**

Submitted By:

Raghav Vehgal

Programme: 1624-S

Batch:

Under the Guidance of

Name of Project Guide: Sumit Jaiswal

Designation: Associate Professor

**LOVELY FACULTY OF DISTANCE EDUCATION**

**LOVELY PROFESSIONAL UNIVERSITY, PHAGWARA**

**2024**

ii

**Lovely Professional University, Phagwara**

**Lovely Faculty of Distance Education**

**DECLARATION**

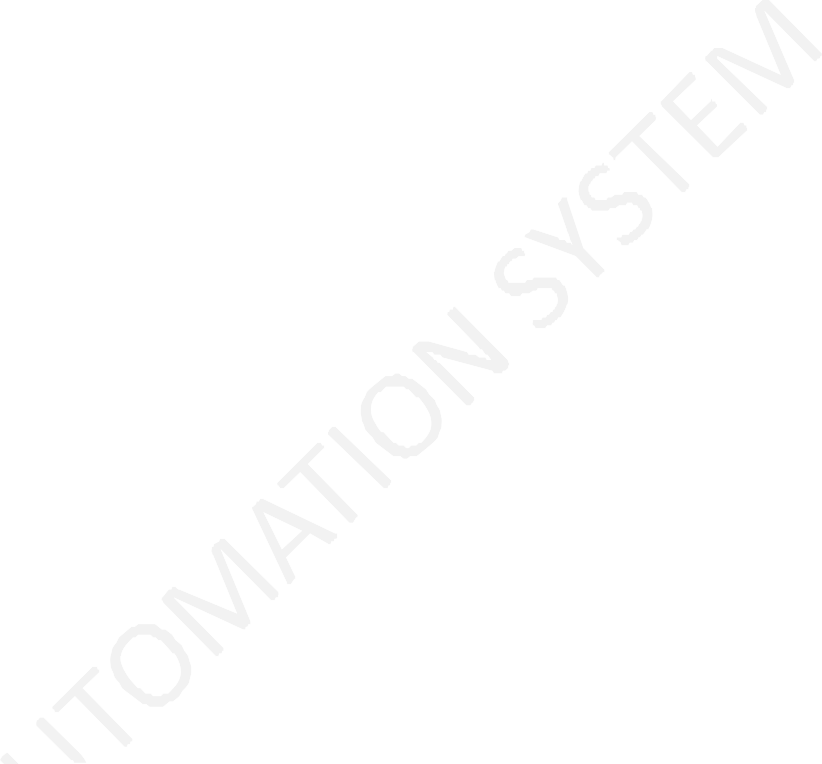
This to certify that the project report titled “**School Automation System**“ submitted by **Raghav Vehgal** , **“ Registration No. 22106070322** “in partial fulfillment for the award of “ Name of Program ” **from Lovely Professional University, Phagwara** is a record of original work done by me. This project has not been submitted to any other University or Institution for the award of any Degree/Diploma/certificate.

**Signature:**

**Place:**

**Date:**

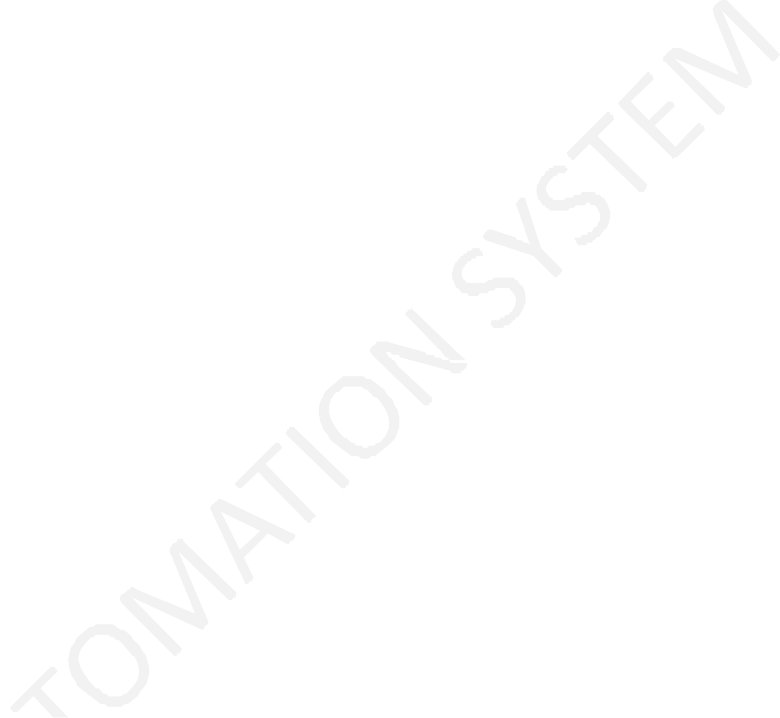
**iii**

**Table of Content**

|  |  |  |
| --- | --- | --- |
| **Sr.No** | **Chapter Name** | **Page No.** |
| a) | Organization Overview |  |
| b) | Profile of the Problem |  |
| c) | Problem Analysis  Product Definition  Feasibility Analysis  Project Plan |  |
| d) | Software Requirement Analysis  Introduction  General Description  Specific Requirements |  |
| e) | Design  System Design  Design Notation  Detailed Design  Flow Charts  Pseudo Code |  |
| f) | Testing  Testing |  |
| g) | User Manual |  |
| h) | Software Used  Software used |  |
| i) | Conclusion and Limitation  Conclusion  Limitation |  |
| j) | Bibliography |  |
|  |  |  |

iv

1. **ORGANIZATION OVERVIEW**



School Automation Systems plays an essential role in the current educational system. School authorities all over the world are engaged in a lot of day-to- day administrative and academic activities to manage and provide a better academic experience to students effectively. However, maintaining and keeping track of school administrative activities is not an easy process. It requires hard work and often it is time- consuming.

To better perform the school administrative activities of educational institute , educational institutes utilize School Automation software now days. Such applications often offer many features that help to enhance the performance of schools with minimum efforts. School Automation software does it by avoiding the manual paper works and automation of many academic and administrative activities.

We will talk about the existing system of School Automation System in later chapters, there limitation problems and what will we do to solve problems or overcome limitation of existing system.

Here the advance thing we are going to do in this project is creating e-mail transfer system. We will create some e-mail which will send to bulk of students automatically etc. After successful completion of registration of a student we will send him thanks for joining mail with some other details, and will do some other stuff.

Main focus is to create a software, which is user friendly and can able to perform all basic activity of the institute

1

# PROFILE OF THE PROBLEM

## 

*“Research or Investigation is the best way to know something --- More you investigate more you understand the things---”*

Before start working on my Project, I investigate the existing system Problems in a School(which is situated in my local area). That System perform all the administrative like registration of student and staff , Keep record of its fee payment , Fee structure of various courses in institute. These all things are performed in paper pan way.

Now I will convert there paper pen manual work to computerized work. I will make my system with these features of registration , fee payment etc. and will add some other features.

You no nothing is perfect in this world specially in computer world. Every system can be updated and we will add some other things init .

So, in my investigation I found some limitation of system .

Some limitation like-:

* **They store student record in big registers, which make them work hard to see details of a particular student.**
* **No use of technology to communicate with students or staff.**

#### Student has to come to college for keep himself up-to-date with notice board. Which is not possible in these COVID-19 Days.

here are the some limitation in the existing system. As I am working on it, so it is my responsibility to over come these limitation as much as possible, so that it will give an positive effect to my Project.

2

## PROBLEM ANALYSIS

In previous section(B) we took an eye on some limitation of Existing System. Limitation is always come with problem. In this section we will discuss about the Problems arrived in the system and will discuss the solution in next section.

* + - **Paper Pen Work:**
      * In Existing System they do paper pen work.
      * They do registration of student in there big registers. And it become a challenging task for them to keep safe and secure these registers from any kind of damage or any kind of human mistake.
      * Loss of the records can create a trouble for a institute

#### New Notice Issue:

* + - * In this COVID period many of the institute is close and people are trying to avoid social gathering. Think Institute has issued a new notice regarding, pending payments due date or regarding exam and classes time table. So it is difficult for student to come to a particular location and see the notice. So it is responsibility of institute to tell the student/staff about the new notice, by using some kind of technology, so that all the members always be up-to-date with the Institute.

## Product Definition-:.

School Automation Systems is user to keep the Student record in the school database and maintain the information about the students.

Here the advance thing we are going to do in this project is creating e-mail transfer system. We will create some e-mail which will send to bulk of students automatically etc. After successful completion of registration of a student we will send him thanks for joining mail with some other details, and will do some other stuff.

Main focus is to create a software, which is user friendly and can able to perform all basic activity of the institute

## Feasibility Analysis:

**Overview-:**

Feasibility studies are important to business development. They can allow a business to address where and how it will operate. It also help to tell the project we are working on will help the business or not. Resources used while using the project is affordable by a user or not. It is important if some one investing money on something, it must be worth it.

Here in this section we will divide feasibility into three types-:

* **Economic Feasibility-** The objective of the economic feasibility I to develop a financial model of the business venture.
* **Behavioral Feasibility-** This assessment involves undertaking a study to analyze and determine whether—and how well– the organization needs can be met by completing the project.

3

* **Technical Feasibility-:** Here we will talk about how we will provide service to the customer.

#### Economic Feasibility-:

As we have discussed some hardware and software requirement to run School Automation application. Things are not cheap to buy they are not that expensive that some one cant afford that, it is good Investment for a organization/institute. This kind of system make work faster. In short we are using money to save time. Hardware and software needed for this system are important tools to run this application.

Things to buy-: A complete desktop system with 4GB of RAM and 1TB of Rom , window 7 or above O.S. it does not cost you more then 1 Lakh rupee.

#### Behavioral Feasibility-:

This project help Institute to make administrative work easy and quick, Think about tradition way of storing record in paper then searching for particular record from bulk of record.

This task is quite hard, but computer can do searching and sorting things very quickly. This application fulfill all the small needs of institute. And it is not so expensive.

#### Technical Feasibility-:

I did not need any special kind of technology to transfer my software from one computer to another. This process can simply be done with using pendrive, or with direct transfer of program from my computer to the target computer using some wire etc.

And I will set the arrangement of this program on other computer so it will start working on other computer and my client can simply use it.

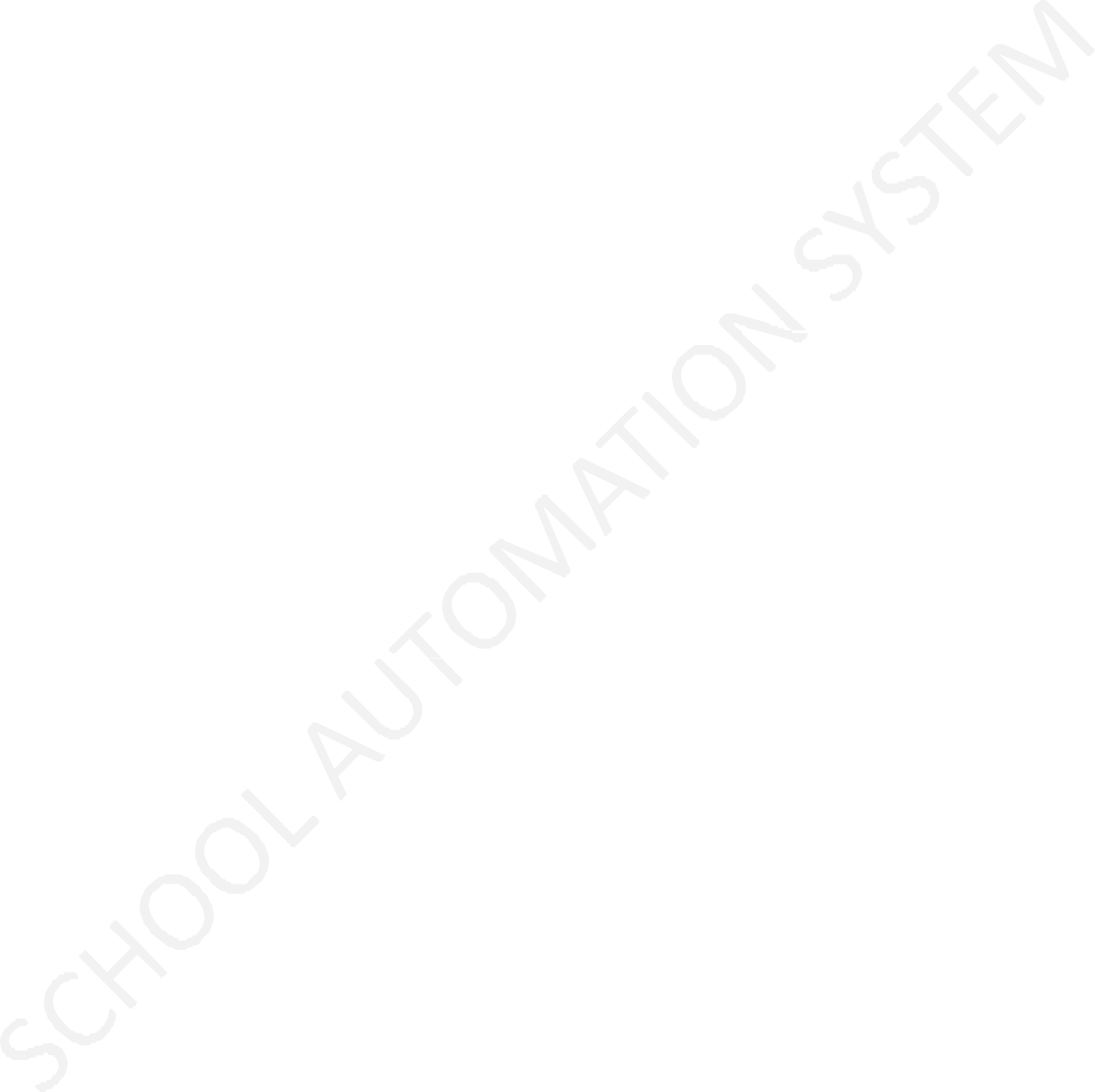
## Project Plan:

## Built the system which help in educational institute to maintain the data records easily with atleast paper work

## 4

1. **SOFTWARE REQUIREMENT ANALYSIS**

#### Introduction-:

In this section we will discuss, what special kind of Hardware or Software required to run my application. I will try to explain all the minimum requirement of the resources we need to perfectly use SAS System. Without talking to much lets concentrate on system requirements.

#### Specific Requirements:

#### Hardware Requirements-:

* **Complete Desktop-setup:** Need a complete system of desktop, by complete system I means **monitor, keyboard, C.P.U, mouse.** These are the most basic requirements of every desktop.
* **Random-access Memory(RAM):** Minimum required **4GB** of RAM to run program in a system without any kind of lag, But as we no **1GB** of space is almost covered by Operating System installed on system.
* **Processor:** Intel core i5 processor or higher processors
* **Good Source Internet Connection:** As we are sending mail so we need a Internet connection.
* **Secondary Memory:** Secondary Memory for storing records of students and staff. Its all depend how much large amount data you are going to store. I personally prefer **1TB** of System space. This much space help to store bulk of records.
* **Camera or Picture Scanner:** To click or scan Pictures of student and staff to create ID cards etc.

#### Software Requirement-:

* **Windows Operating System:** Microsoft Windows operating system, The main reason is Windows is one of the most used operating system in the world. Because he is user friendly and easy to use/understand. Window 7 or more will be preferable.
* **MySql:** For database purpose.

5

**SOFTWARE REQUIREMENT AND SPECIFICATION**

## Software Requirements Specifications-:.

In School Automation System first thing you will see is login page where the staff (teacher,clerk,admin) can login using there password and id.

After the successful login-

Clerk can do task like registration of new student. After every successful registration system will generate welcome mail to new registered person on there mail id and all details of course he opt, or collection of fees and fine, can watch records of student who did not clear the dues, and able to send them mail regarding there pending fees.

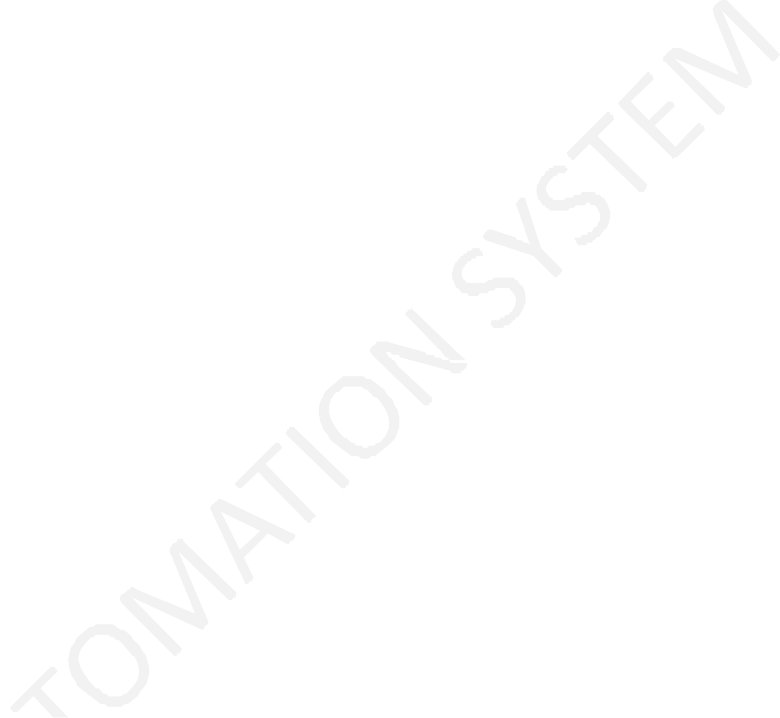
If a new Teacher is joined the school the system will generate him/her a id and a password automatically with the help of this id and password that teacher can login in the system with this password and user id and he can see the record of class he is allocated

Teacher can do task like add and update marks of students in every semester in every subject. Teacher are only able to maintain the marks record of those courses to which he/she is teaching. Teacher can generate Marks-sheet of every student. Teacher can send mail to every student individually or in a bulk to all class students. Writing of mail content can be done threw this system no need to go and open Gmail separately. Sender will be the email id of teacher, he need to enter password he can save it in his system as a file rather then in a server.

Admin is the one who controls this application, like giving access to staff by generating there user account in this application. Admin can add or update new course added in school like- Addition of new subject In a course. He is able to see the all recent activity done by clerk like new admission fee payment,

6

## Output Requirements:



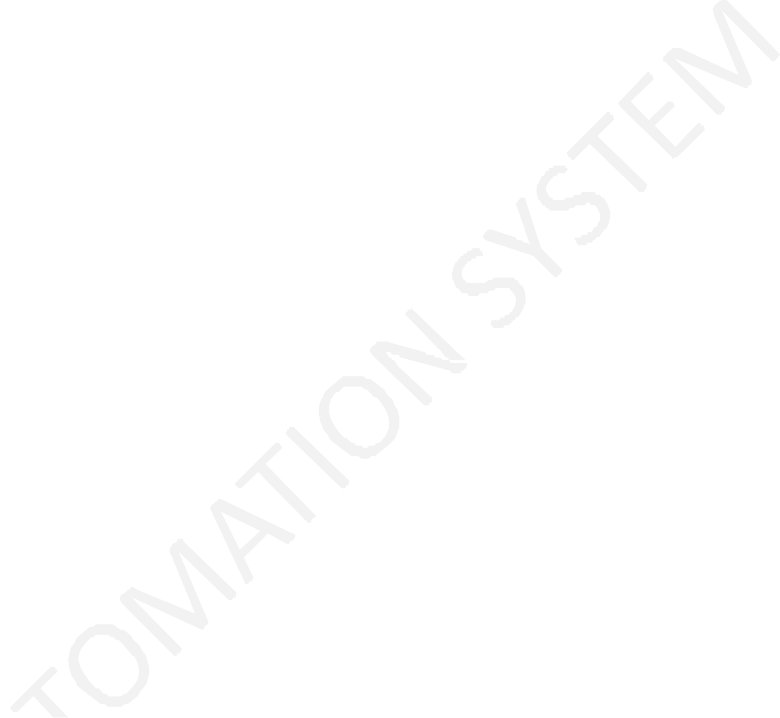
**Home Panel (Teacher)**

**Login Panel**

|  |  |
| --- | --- |
| Report Name | Login Panel |
| Objective | Staff of school will login using there Id and password to use the application. |
| Input | Person want to use application must have a user Id and a Password |
| Validation | Before successful login there is a validation check is done by system.  Where system will check the user wants to enter must fill the user-name field and password field correct, and must select the type of staff (teacher, admin, clerk)  Validation check will held when user click on login button. |

|  |  |
| --- | --- |
| Report Name | Home Panel |
| Objective | Here Teacher see the list of courses/subject he is teaching in school. |
| Input | Teacher click on particular subject and can see the student name roll  -no and marks |
| Process | List of student will visible after clicking on subject button and can update the list can see marks is different format like lower then higher then and can print the list. |

7



**Fee Panel**

**Admission Form**

**Home Panel (Clerk)**

|  |  |
| --- | --- |
| Report Name | Home Panel |
| Objective | Here Clerk see buttons like new admission, fees, salary etc. |

|  |  |
| --- | --- |
| Report Name | Admission |
| Objective | Admit a new Student in the school. |
| Process | Enter all the basic and personal details, Which is used by the school for persons records. |
| Input | Personal information like Name , addhar no, contact no and address , Image, family info, previous school info, paying fees, course he is going to join etc. |

|  |  |
| --- | --- |
| Report Name | Fee Panel |
| Objective | Maintain fee records of all students. |
| Process | List of all student whose fees is pending |
| Input | Take student rollno, and see its fee details. Take fee from student and marked as cleared or not. |

8

**Home Panel (Admin)**

|  |  |
| --- | --- |
| Report Name | Admin’s Home Panel |
| Objective | Add a new user to the application, can be a new Teacher or new Clerk |
| Input | Take staff id and generate a user for application. |

**Functional Requirements:**

Functional Requirement is a description of the services that the software must offer. It describes a software system or its component. Project “School Automation System” will be divided into the various modules.

* + - **Login Panel:**
      * User come and select there type (Admin,Teacher,Clerk) enter there Id and password to login in the system.

#### oRegistration Panel:

* + - * Clerk will do student registration and allot him class. Class Roll number is provided by the system.
      * Clerk can do Teacher registration
      * Can maintain fee records of the students.

#### Marks Panel:

* + - * Teacher can maintain marks of its class students. Marks depend on terms.

#### Fee Panel:

* + - * Here Clerk will maintain fee records of students.

#### oActivity Panel:

* + - * Here admin can see all the report and activity done by clerk.

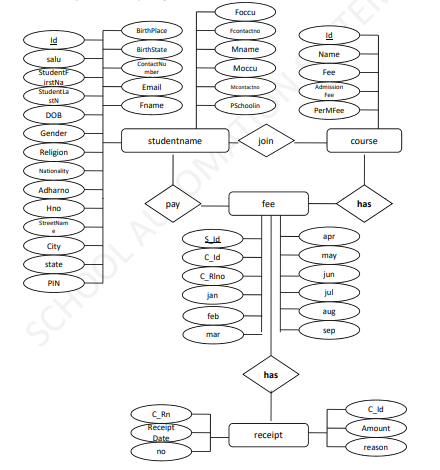
#### oCourse Panel:

* + - * Admin can view classes in the school, can add new class with subjects, admission fee, per-month fee.

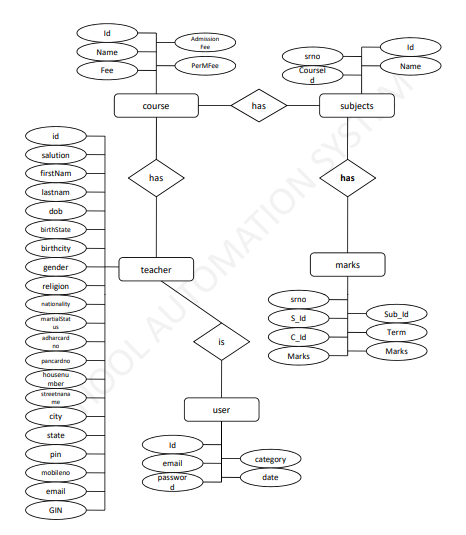
9

# DESIGN

# System Design



10



11

#### Studentname:

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Description |
|  |  |  |
| ID | INT(11) | ID give unique identification to each Student. |
| Salu | VARCHAR(4) | *Students Salutation* |
| StudentFirstName | VARCHAR(10) | Here we store first name of student |
| StudentLastName | VARCHAR(10) | Here we store Last Name of Student |
| Gender | VARCHAR(7) | It tells student is a girl or a boy |
| DOB | DATE | To know the birth date of student. |
| Religion | VARCHAR(15) | To know Religion of student |
| Nationality | VARCHAR(15) | To know Nationality of student |
| Adharno | VARCHAR(15) | *Government proof* |
| Hno | VARCHAR(15) | *Address* |
| StreetName | VARCHAR(45) | *Address* |
| City | VARCHAR(15) | *Address* |
| PIN | VARCHAR(11) | *Address* |
| State | VARCHAR(15) | *Address* |
| ContactNumber | VARCHAR(12) | Persons telephone or mobile number |
| Email | VARCHAR(40) | Persons mail id, for sending fee details , school notice, etc. to student |
| CourseId | INT(11) | To know in which course student has taken  FOREIGN\_KEY with course table |
| Image | VARCHAR(50) | Picture of student, to identify him. |
| Register\_Date | DATE | Date when student register in school |

**PRIMARY\_KEY -: ID, FOREIGN\_KEY -:Course\_Id**

12

#### Courses:

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Description |
| Id | NT(11) | Unique Id for each course |
| Name | VARCHAR(4) | *Course Name* |
| Fee | VARCHAR(10) | Course Fee |
| AdmissionFee | VARCHAR(10) | Course Admission Fee |
| PerMFee | VARCHAR(7) | It tells student is a girl or a boy |

**Primary Key: ID**

* + 1. **Marks:**

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Description |
| S\_Id | INT(11) | For particular student Id |
| C\_Id | INT(11) | *Course student has taken* |
| Sub\_Id | INT(11) | *Subject between the course* |
| Term | INT(11) | For which Term |
| Marks | INT(11) | *Marks of student* |
| C\_RollNo | INT(11) | *Class RollNo* |

**PRIMARY\_KEY -: Id FOREIGN\_KEY-: S\_ID, C\_Id, Sub\_Id**

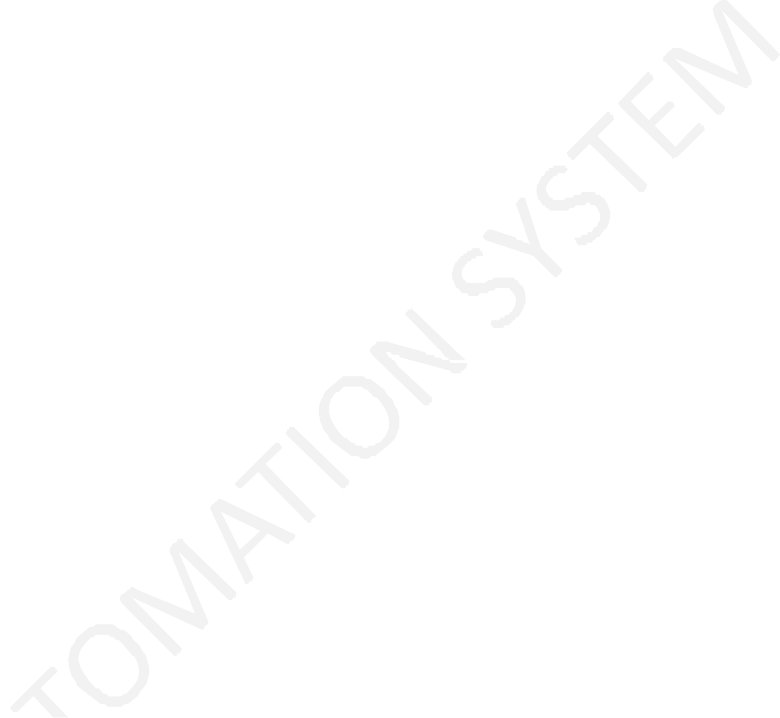
* + 1. **Receipt:**

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Description |
| C\_Id | INT(11) | *Course student has taken* |
| C\_Rn | INT(11) | *Class Rollno of student* |
| ReceiptDate | DATE | Date when he pay bills |
| Amount | INT(11) | *How much amount he pays* |
| Reason | VARCHAR(45 | *Reason of paying amount* |
| No | INT(11) | *Unique Numbers* |

**PRIMARY\_KEY -: No FOREIGN\_KEY-: C\_Id**

13

#### Fee:



**FOREIGN\_KEY-: S\_ID, C\_Id,**

**6) Subjects:**

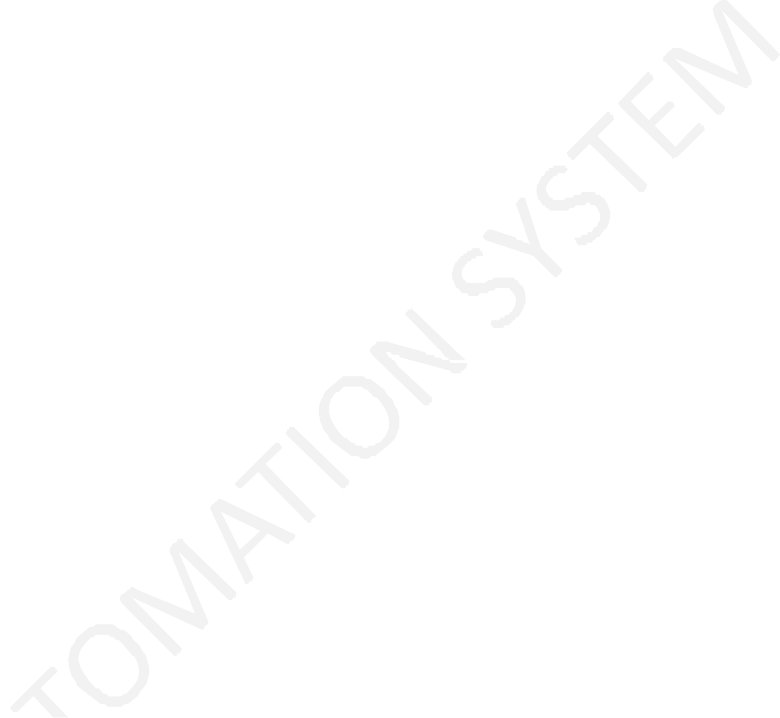
|  |  |  |
| --- | --- | --- |
| Name | Data Type | Description |
|  |  |  |
| S\_Id | INT(11) | Student Id |
| C\_Id | INT(11) | *Course Id* |
| C\_Rlno | INT(11) | *Class Roll number* |
| Jan | INT(11) | Fee detail of January month |
| Feb | INT(11) | Fee detail of February month |
| Mar | INT(11) | Fee detail of March month |
| Apr | INT(11) | Fee detail of April month |
| May | INT(11) | Fee detail of May month |
| Jun | INT(11) | Fee detail of June month |
| Jul | INT(11) | Fee detail of July month |
| Aug | INT(11) | Fee detail of August month |
| Sep | INT(11) | Fee detail of September month |
| Oct | INT(11) | Fee detail of October month |
| Nov | INT(11) | Fee detail of November month |
| Decm | INT(11) | Fee detail of December month |
| Year1 | INT(11) | *Session Start Year* |
| year2 | INT(11) | Session End Year |

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Description |
| CousresId | INT(11) | To represent course |
| Id | VARCHAR(4) | *Unique Id for every subject in a course* |
| Name | VARCHAR(10) | Name of subject |

**PRIMARY\_KEY -: Id FOREIGN\_KEY-: CourseId**

14

#### Teacher:



|  |  |  |
| --- | --- | --- |
| Name | Data Type | Description |
|  |  |  |
| ID | INT(11) | ID give unique identification number to each Teachers. |
| Subjectpref1 | VARCHAR(7) | Teacher Preference to a subject |
| Subjectpref2 | VARCHAR(7) | Teacher second Preference to a subject |
| Salutation | VARCHAR(4) | *Teacher Salutation* |
| FirstName | VARCHAR(15) | Initial Name of Teacher |
| LastName | VARCHAR(15) | Last Name of Teacher |
| Gender | VARCHAR(7) | *Teacher Gender* |
| DOB | DATE | To know the birth date of Teacher |
| Religion | VARCHAR(15) | To know Religion of Teacher |
| Nationality | VARCHAR(15) | To know Nationality of Teacher |
| Adharno | VARCHAR(15) | *Government proof* |
| Hno | VARCHAR(15) | *Address* |
| StreetName | VARCHAR(45) | *Address* |
| City | VARCHAR(15) | *Address* |
| PIN | VARCHAR(11) | *Address* |
| State | VARCHAR(15) | *Address* |
| ContactNumber | VARCHAR(12) | Persons telephone or mobile number |
| Email | VARCHAR(40) | *Teacher Mail Id to send Id and password to access this application.* |
| HghestQualification | VARCHAR(15) | Highest Qualification he/she did. |
| Image | VARCHAR(50) | Picture of student, to identify him. |
| Bed | VARCHAR(3) | *Is teacher did B.Ed or Not* |
| PancardNo | VARCHAR(15) | *Take Pan Card Number* |

**PRIMARY\_KEY -: ID**

15

#### User:

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Description |
| Id | VARCHAR(10) | User Id |
| Email | VARCHAR(4) | *Email of application user* |
| Password | VARCHAR(10) | Password of application user |
| Category | VARCHAR(2) | Type of User |

**FOREIGN\_KEY-: Id, Email**

#### TeacherAllocation:

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Description |
| Cid | INT(11) | To represent course |
| Tid | INT(11) | *To represent teacher* |

**FOREIGN\_KEY-: Cid,Tid**

16

## Data Flow Diagram-:

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. It shows how data enters and leaves the system and where data is stored. The objective of DFD is to show the scope and boundaries of a system as a whole. It may be used as a starting point for redesigning a system. The DFD is also called as a data flow graph or bubble chart.

Standard symbols for DFDs are given below:

|  |  |  |
| --- | --- | --- |
| Symbol | Name | Function |
|  | Data Flow | Used to Connect Symbol of each other |
|  | Process | Perform Some transformation of input data to yield output data |
|  | Source | Source of System input |
|  | Data Store | A repository of data |

## Data Flow Diagram Level 0:

Admin

Teacher

School

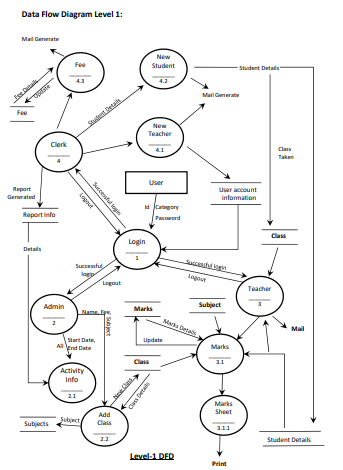
Automation System

Database

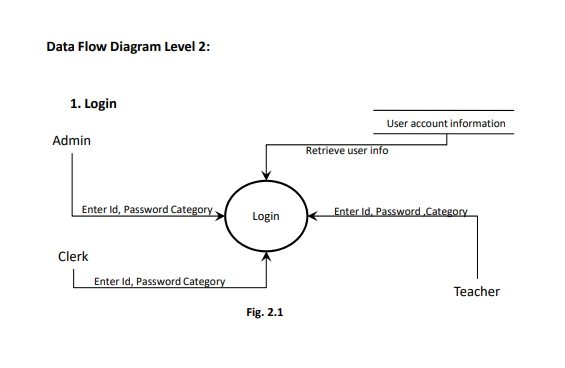
Clerk

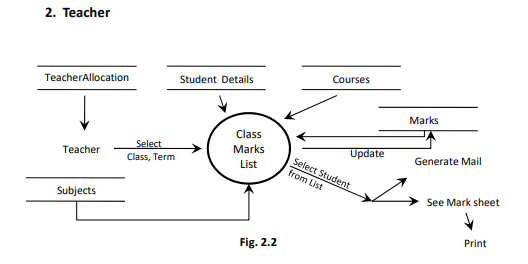
**Level-0 DFD**

17



18





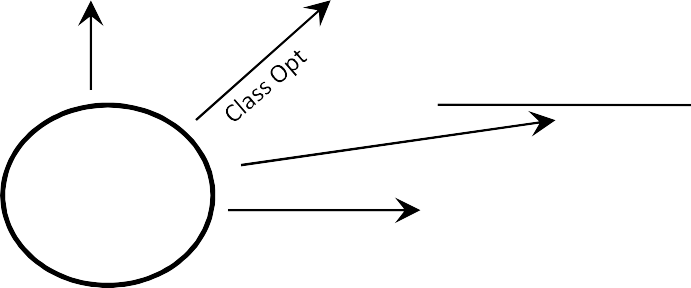
19

## 2. Clerk

###### user

Marks Courses

###### Clerk



Student Details

Add

Student

Mail Generated

Details

user

###### Teache

Records

Clerk

Details

Add Teacher

###### User Id, password generated

user

###### Student Details

Mail Generated

###### fee

Clerk

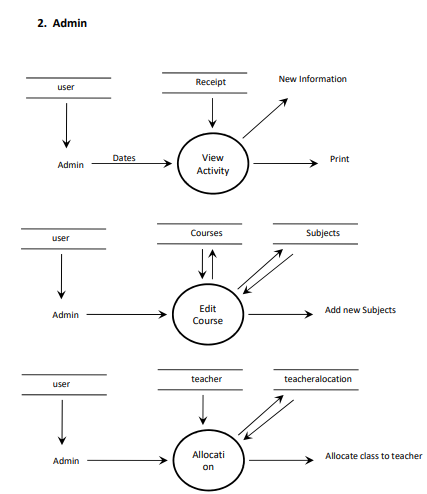
Class,Rollno

Fee

###### Mail Generated

Record

20



21

## Pseudo Code

## Class School Automation System

## package schoolautomationsystem;

## public class SchoolAutomationSystem {

## public static void main(String[] args) {

## 

## new MainFrame();

## }

## 

## }

## Class Login package schoolautomationsystem;

## import java.awt.Color;

## import java.awt.event.FocusEvent;

## import java.awt.event.FocusListener;

## import java.awt.event.KeyEvent;

## import java.awt.event.KeyListener;

## import java.awt.event.MouseEvent;

## import java.awt.event.MouseListener;

## import java.io.File;

## import java.io.IOException;

## import java.sql.SQLException;

## import java.util.logging.Level;

## import java.util.logging.Logger;

## import javax.imageio.ImageIO;

## import javax.swing.ImageIcon;

## import javax.swing.JButton;

## import javax.swing.JComboBox;

## import javax.swing.JPanel;

## import javax.swing.JLabel;

## import javax.swing.JOptionPane;

## import javax.swing.JTextField;

## import javax.swing.border.LineBorder;

## public class Login extends JPanel {

## 

## 

## JPanel imgP=new JPanel();

## JPanel bodyP=new JPanel();

## 22

## JLabel headingL=new JLabel("<html> <font style=\" font-size:30; font-family:Arial;\">Login");

## JTextField loginTF=new JTextField();

## JPanel loginDLine=new JPanel();

## JTextField passwordTF=new JTextField();

## JComboBox chooseCB= new JComboBox();

## JPanel passwordDLine=new JPanel();

## JButton submitB=new JButton("<html> <pre><font style=\" font-size:12; font-family:arial;\">Login");

## JPanel borderP=new JPanel();

## 

## MyConnection con=new MyConnection();

## 

## String email="",pass="",cat="";

## int Tid=0;

## 

## public Login() {

## 

## }

## 

## void design(MainFrame f){

## 

## setLayout(null);

## //frame setting

## f.setBounds(f.x\*20, f.y\*20,f.x\*70 , f.y\*70);

## f.setTitle("School Automation System/Login");

## 

## //panel setting

## setBackground(Color.WHITE);

## setBounds(0, 0, f.getWidth(), f.getHeight());

## setVisible(true);

## 

## int x=f.getWidth()/100,y=f.getHeight()/100;

## 

## System.out.println(x\*4+" "+f.getHeight());

## 

## 

## JLabel imgL=new JLabel("<html><font style=\" font-size:105; color:#ffffff; \">👨‍");

## imgL.setBounds(x\*11,y\*11,x\*15,y\*25);

## //imgL.setOpaque(true);

## 

## //img

## JLabel heading1=new JLabel("<html><font style=\" font-size:25; color:#ffffff; font-family:Century Gothic;\">Welcome to the");

## 23

## heading1.setBounds(x\*12,y\*42,x\*30,y\*8);

## //heading1.setOpaque(true);

## 

## JLabel heading2=new JLabel("<html><font style=\" font-size:25; color:#ffffff; font-family:Century Gothic;\">School Automation");

## heading2.setBounds(x\*8,y\*50,x\*32,y\*8);

## //heading2.setOpaque(true);

## 

## JLabel heading3=new JLabel("<html><font style=\" font-size:25; color:#ffffff; font-family:Century Gothic;\">System");

## heading3.setBounds(x\*24,y\*58,x\*15,y\*8);

## //heading3.setOpaque(true);

## 

## JLabel heading4=new JLabel("<html><font style=\" font-size:10; color:#ffffff; font-family:Century Gothic;\">Developed By");

## heading4.setBounds(x\*25,y\*97,x\*8,y\*4);

## 

## JLabel heading5=new JLabel("<html><font style=\" font-size:10; color:#ffffff; font-family:Century Gothic;\">Raghav Vehgal");

## heading5.setBounds(x\*24,y\*101,x\*10,y\*4);

## 

## imgP.setBounds(0, 0, x\*40, f.getHeight());

## imgP.setLayout(null);

## imgP.setBackground(new Color(41,28,185));

## imgP.setVisible(true);

## imgP.setOpaque(true);

## 

## imgP.add(heading1);

## imgP.add(heading2);

## imgP.add(heading3);

## imgP.add(heading4);

## imgP.add(heading5);

## imgP.add(imgL);

## //bodyP design

## bodyP.setBounds(x\*40, 0, x\*60, f.getHeight());

## bodyP.setBackground(Color.white);

## bodyP.setLayout(null);

## bodyP.setVisible(true);

## 

## x=bodyP.getWidth()/100;y=bodyP.getHeight()/100;

## 

## borderP.setBounds(0, 0, bodyP.getWidth(), 10);

## borderP.setBackground(Color.blue);

## 

## headingL.setBounds(x\*10, y\*10, x\*40, y\*10);

## headingL.setForeground(Color.blue);

## 24

## chooseCB.setBounds(x\*20, y\*30,x\*60,y\*8);

## chooseCB.setBackground(Color.white);

## chooseCB.setForeground(Color.blue);

## chooseCB.setBorder(new LineBorder(Color.blue,2, true));

## chooseCB.addItem("Select Category");

## chooseCB.addItem("Admin");

## chooseCB.addItem("Teacher");

## chooseCB.addItem("Clerk");

## 

## JLabel loginTFicon=new JLabel("<html> <font style=\" font-size:20;color:#4169E1\">👨‍");

## loginTFicon.setBounds(x\*14,y\*42,x\*6,y\*9);

## 

## loginTF.setBounds(x\*20,y\*42,x\*60,y\*8);

## loginTF.setBorder(null);

## loginTF.setText("user name");

## loginTF.setForeground(Color.gray);

## loginTF.addFocusListener(new FocusListener() {

## @Override

## public void focusGained(FocusEvent e) {

## loginTF.selectAll();

## }

## @Override

## public void focusLost(FocusEvent e) {

## 

## }

## });

## 

## loginDLine.setBounds(x\*20, y\*51, x\*60, 2);

## loginDLine.setBackground(Color.blue);

## 

## JLabel passTFicon=new JLabel("<html> <font style=\" font-size:20;color:#4169E1\">🔓");

## passTFicon.setBounds(x\*14,y\*64,x\*6,y\*9);

## 

## passwordTF.setBounds(x\*20,y\*64,x\*60,y\*8);

## passwordTF.setBorder(null);

## passwordTF.setText("password");

## passwordTF.setForeground(Color.gray);

## passwordTF.addFocusListener(new FocusListener() {

## @Override

## public void focusGained(FocusEvent e) {

## passwordTF.selectAll();

## }

## 25

## @Override

## public void focusLost(FocusEvent e) {

## 

## }

## });

## 

## passwordDLine.setBounds(x\*20, y\*72, x\*60, 2);

## passwordDLine.setBackground(Color.blue);

## 

## bodyP.add(borderP);

## bodyP.add(headingL);

## bodyP.add(loginTFicon);

## bodyP.add(loginTF);

## bodyP.add(loginDLine);

## bodyP.add(passTFicon);

## bodyP.add(passwordTF);

## bodyP.add(passwordDLine);

## bodyP.add(submitB);

## bodyP.add(chooseCB);

## 

## 

## submitB.setBounds(x\*20, y\*80,x\*20 , y\*10);

## submitB.setForeground(Color.blue);

## submitB.setBorder(new LineBorder(Color.blue, 2, true));

## submitB.setBackground(Color.white);

## submitB.setOpaque(true);

## submitB.addMouseListener(new MouseListener() {

## @Override

## public void mouseClicked(MouseEvent e) {

## 

## fun(f);

## }

## @Override

## public void mousePressed(MouseEvent e) {

## 

## }

## @Override

## public void mouseReleased(MouseEvent e) {

## 

## }

## 26

## @Override

## public void mouseEntered(MouseEvent e) {

## submitB.setBackground(Color.cyan);

## }

## @Override

## public void mouseExited(MouseEvent e) {

## submitB.setBackground(Color.white);

## }

## });

## String pass="no";

## 

## 

## submitB.addKeyListener(new KeyListener() {

## @Override

## public void keyTyped(KeyEvent e) {

## 

## }

## @Override

## public void keyPressed(KeyEvent e) {

## if(e.getKeyCode()==KeyEvent.VK\_ENTER){

## fun(f);

## }

## }

## @Override

## public void keyReleased(KeyEvent e) {

## 

## }

## });

## 

## add(imgP);

## add(bodyP);

## 

## }

## 

## void fun(MainFrame f){

## 

## switch (chooseCB.getSelectedIndex()) {

## case 1:

## pass="";

## cat="";

## 

## 27

## try {

## con.pre\_s=con.conn.prepareStatement("select Id,password,Category from user where Id=? or email=?");

## con.pre\_s.setString(1, loginTF.getText().trim());

## con.pre\_s.setString(2, loginTF.getText().trim());

## con.r\_set=con.pre\_s.executeQuery();

## 

## while(con.r\_set.next()){

## Tid=Integer.parseInt(con.r\_set.getString(1));

## pass=con.r\_set.getString(2);

## cat=con.r\_set.getString(3);

## }

## } catch (SQLException ex) {

## JOptionPane.showMessageDialog(null, ex, "Login case 2", 0);

## }

## System.out.println("Tid pass cat= "+Tid+" "+pass+" "+cat);

## if(pass.equals(passwordTF.getText().trim()) && cat.equals("a")){

## setVisible(false);

## f.login.setVisible(false);

## if(f.adminV==0){

## f.adminP.removeAll();

## f.adminP.design(f);

## }else{

## f.adminP.setVisible(true);

## f.setBounds(0, 0,f.width, f.height-(f.height/100));

## }

## 

## }else{

## JOptionPane.showMessageDialog(null, "Email password or selected Category no match in database", "Mis Match", 1);

## }

## break;

## 

## case 2:

## pass="";

## cat="";

## 

## try {

## con.pre\_s=con.conn.prepareStatement("select Id,password,Category from user where Id=? or email=?");

## con.pre\_s.setString(1, loginTF.getText().trim());

## con.pre\_s.setString(2, loginTF.getText().trim());

## con.r\_set=con.pre\_s.executeQuery();

## 

## 

## 28

## while(con.r\_set.next()){

## Tid=Integer.parseInt(con.r\_set.getString(1));

## pass=con.r\_set.getString(2);

## cat=con.r\_set.getString(3);

## }

## } catch (SQLException ex) {

## JOptionPane.showMessageDialog(null, ex, "Login case 2", 0);

## }

## System.out.println("Tid pass cat= "+Tid+" "+pass+" "+cat);

## if(pass.equals(passwordTF.getText().trim()) && cat.equals("t")){

## 

## setVisible(false);

## f.teacherP.removeAll();

## f.teacherP.intial(Tid);

## f.teacherP.design(f);

## f.adminP.setVisible(false);

## }else{

## JOptionPane.showMessageDialog(null, "Email password or selected Category no match in database", "Mis Match", 1);

## }

## 

## break;

## 

## case 3:

## pass="";

## cat="";

## 

## try {

## con.pre\_s=con.conn.prepareStatement("select Id,password,Category from user where Id=? or email=?");

## con.pre\_s.setString(1, loginTF.getText().trim());

## con.pre\_s.setString(2, loginTF.getText().trim());

## con.r\_set=con.pre\_s.executeQuery();

## 

## while(con.r\_set.next()){

## Tid=Integer.parseInt(con.r\_set.getString(1));

## pass=con.r\_set.getString(2);

## cat=con.r\_set.getString(3);

## }

## } catch (SQLException ex) {

## JOptionPane.showMessageDialog(null, ex, "Login case 2", 0);

## }

## System.out.println("Tid pass cat= "+Tid+" "+pass+" "+cat);

## 

## 29

## Class Home:

## package schoolautomationsystem;

## import java.awt.Color;

## import java.awt.event.MouseEvent;

## import java.awt.event.MouseListener;

## import javax.swing.JLabel;

## import javax.swing.JPanel;

## import javax.swing.border.LineBorder;

## import javax.swing.border.SoftBevelBorder;

## public class Home extends JPanel{

## JPanel sideButtonP=new JPanel();

## 

## JPanel containerP=new JPanel();

## 

## 

## JLabel studentB=new JLabel("<html> <pre><font style=\" font-size:20; font-family:arial;\"> 🎓 Student");

## JPanel selectorSP=new JPanel();

## 

## JLabel teacherB=new JLabel("<html> <pre><font style=\" font-size:20; font-family:arial;\"> 👨‍ Teacher");

## JPanel selectorTP=new JPanel();

## 

## JLabel expensesB=new JLabel("<html> <pre><font style=\" font-size:20; font-family:arial;\"> 💲 Expenses");

## JPanel selectorEP=new JPanel();

## 

## JLabel settingsB=new JLabel("<html> <pre><font style=\" font-size:20; font-family:arial;\"> ⚙ Settings");

## JPanel selectorSeP=new JPanel();

## 

## JLabel confiB=new JLabel("<html> <pre><font style=\" font-size:20; font-family:arial;\"> 🛠 Configuration");

## JPanel selectorCP=new JPanel();

## 

## JPanel upperLineP=new JPanel();

## 

## JLabel introL=new JLabel("Intro");

## //Student studentP=new Student();

## Configuration configurationP=new Configuration();

## 

## 

## 

## 30

## public Home() {

## 

## add(upperLineP);

## add(sideButtonP);

## add(containerP);

## 

## }

## 

## void design(int x,int y,int width, int height,MainFrame f){

## 

## f.setTitle("School Automation System/Home");;

## 

## upperLineP.setBounds(x\*20, 0, width, 10);

## upperLineP.setBackground(new Color(30,19,34));

## 

## designSideButtonP(x,y,width,height,f);

## designContainerP(x,y,width,height,f);

## 

## f.student.hid();

## //introP.setVisible(true);

## setLayout(null);

## setBounds(0, 0, width, height);

## setVisible(true);

## System.out.println("inside Home");

## 

## }

## 

## void designSideButtonP(int x, int y, int width, int height,MainFrame f){

## 

## int w=x\*20;

## 

## sideButtonP.setBounds(0, 0, x\*20, height);

## sideButtonP.setLayout(null);

## sideButtonP.setBackground(new Color(30,19,34));

## sideButtonP.setBorder(new SoftBevelBorder(0));

## 

## sideButtonP.add(studentB);

## sideButtonP.add(teacherB);

## sideButtonP.add(expensesB);

## sideButtonP.add(settingsB);

## sideButtonP.add(confiB);

## 

## studentB.setBounds(0,y\*10 ,w , y\*10);

## studentB.setOpaque(true);

## 

## 31

## studentB.setBackground(Color.GRAY);

## studentB.setForeground(Color.white);

## studentB.setBorder(new LineBorder(Color.white, 1, true));

## studentB.setLayout(null);

## studentB.add(selectorSP);

## 

## selectorSP.setBackground(Color.RED);

## selectorSP.setBounds(2,0,10,studentB.getHeight());

## selectorSP.setVisible(false);

## 

## studentB.addMouseListener(new MouseListener() {

## @Override

## public void mouseClicked(MouseEvent e) {

## //studentP.design(x, y, width, height, f);

## f.student.sho();

## 

## configurationP.setVisible(false);

## 

## }

## @Override

## public void mousePressed(MouseEvent e) {

## selectorCP.setVisible(false);

## selectorEP.setVisible(false);

## selectorSP.setVisible(true);

## selectorSeP.setVisible(false);

## selectorTP.setVisible(false);

## }

## @Override

## public void mouseReleased(MouseEvent e) {

## 

## }

## @Override

## public void mouseEntered(MouseEvent e) {

## studentB.setBackground(new Color(105,105,105));

## }

## @Override

## public void mouseExited(MouseEvent e) {

## studentB.setBackground(new Color(30,19,34));

## }

## });

## 

## teacherB.setBounds(0,y\*22 ,w , y\*10);

## teacherB.setBorder(new LineBorder(Color.white, 1, true));

## teacherB.setBackground(new Color(30,19,34));

## 

## 32

## teacherB.setForeground(Color.white);

## teacherB.setOpaque(true);

## teacherB.setLayout(null);

## teacherB.add(selectorTP);

## 

## selectorTP.setBackground(Color.RED);

## selectorTP.setBounds(2,0,10,studentB.getHeight());

## selectorTP.setVisible(false);

## 

## teacherB.addMouseListener(new MouseListener() {

## @Override

## public void mouseClicked(MouseEvent e) {

## 

## }

## @Override

## public void mousePressed(MouseEvent e) {

## selectorCP.setVisible(false);

## selectorEP.setVisible(false);

## selectorSP.setVisible(false);

## selectorSeP.setVisible(false);

## selectorTP.setVisible(true);

## }

## @Override

## public void mouseReleased(MouseEvent e) {

## 

## }

## @Override

## public void mouseEntered(MouseEvent e) {

## teacherB.setBackground(new Color(105,105,105));

## }

## @Override

## public void mouseExited(MouseEvent e) {

## teacherB.setBackground(new Color(30,19,34));

## }

## });

## 

## expensesB.setBounds(0,y\*34 ,w , y\*10);

## expensesB.setBorder(new LineBorder(Color.white, 1, true));

## expensesB.setBackground(new Color(30,19,34));

## expensesB.setForeground(Color.white);

## expensesB.setOpaque(true);

## expensesB.setLayout(null);

## expensesB.add(selectorEP);

## 

## 

## 33

## TESTING

## Testing

Testing is the major quality control measure used during software development. Its basic function is to detect errors in the software. During requirement analysis and design, the output is a document that is usually textual and no executable. After the coding phase, computer programs are available that can be executed for testing purposes. This implies that testing not only has to uncover errors introduced during coding, but also errors introduced during the previous phases. Thus, the goal of testing is to uncover requirement, design, and coding errors in the programs.

Different types of testing are performed to detect errors. These are :

Unit Testing: In unit testing, a module is tested separately and is often performed by the coder himself simultaneously along with the coding of the module. The purpose is to exercise the different parts of the module code to detect coding errors.

Integration Testing: During Integration of modules, Integration testing is performed to detect design errors. By focusing on testing the interconnection Between Modules.

System Testing: In this, the system is tested against the System requirements to see if all the requirements are met and if the system performs as specified by the requirements.

Acceptance Testing: It is performed to demonstrate to the client, on the real life data of the client, the operation of the system.

## Test Plan

Testing is an extremely critical and time-consuming activity. It requires proper planning of the overall testing process. Testing process starts with a test plan. This plan identifies all the testing related activities that must be performed and specifies he schedule, allocates the resources, and specified guidelines for testing. The test plan specifies conditions that should be tested, different units to be tested, and the manner in which the modules will be integrated together.

34

**STEPS FOLLOWES IN TESTING OVERALL SOFTWARE:-**

* First of all units testing was performed. In this I tested each and every java file of the system was tested so that there are no syntax and logical errors.
* Integration testing was performed next by combining tested modules or java file into subsystems.

Establish connection with server and database, try to find changes done by any user is updating to other modules or not in some modules update occurs when we restart the application I work on it to solve it and I did it.

* When the development of whole system was completed, we performed overall system testing to discover and remaining errors. As some other Exception does not raise I run this module for continues 6 hours and did all the activity which is my system can do and I hit with some run time errors. Which occurs because some modules are creating its child replica every time we access I also modified so this kind of things does not happen in future.

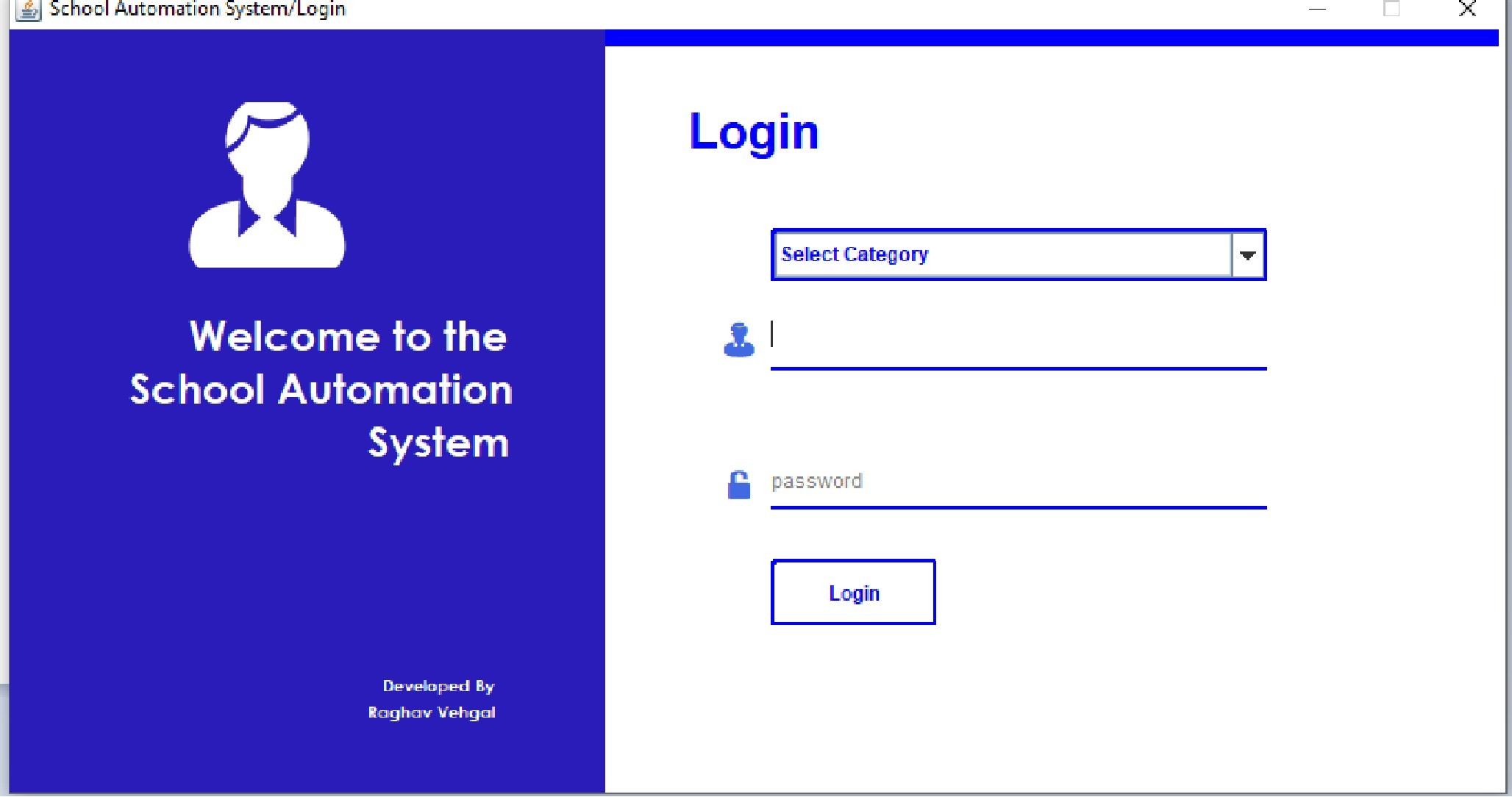
Also encountered with Some unexpected exceptions which occurs due to my coding mistakes but I also resolve that problem, due to this exception system stops automatically so I added a exception handler to resolve this issue.

35

# User Manual

## Login

Select Category Admin ,



Teacher Or Clerk

Enter User Id, Or Registered Email

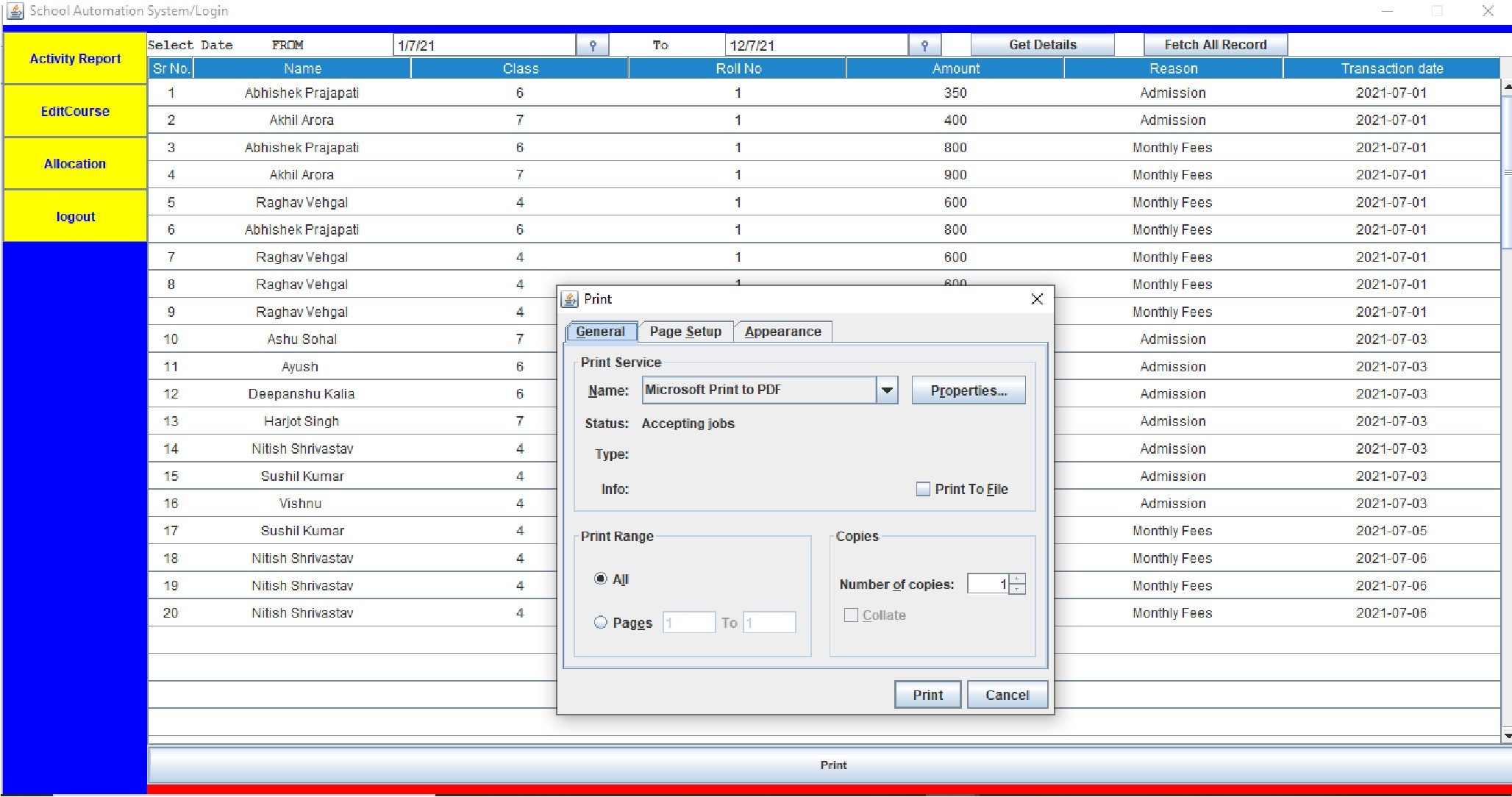
Click or press Enter To login

Enter Password

## Admin

#### .1Report

Active button



Date

To date

Fetch data according

Fetch all data

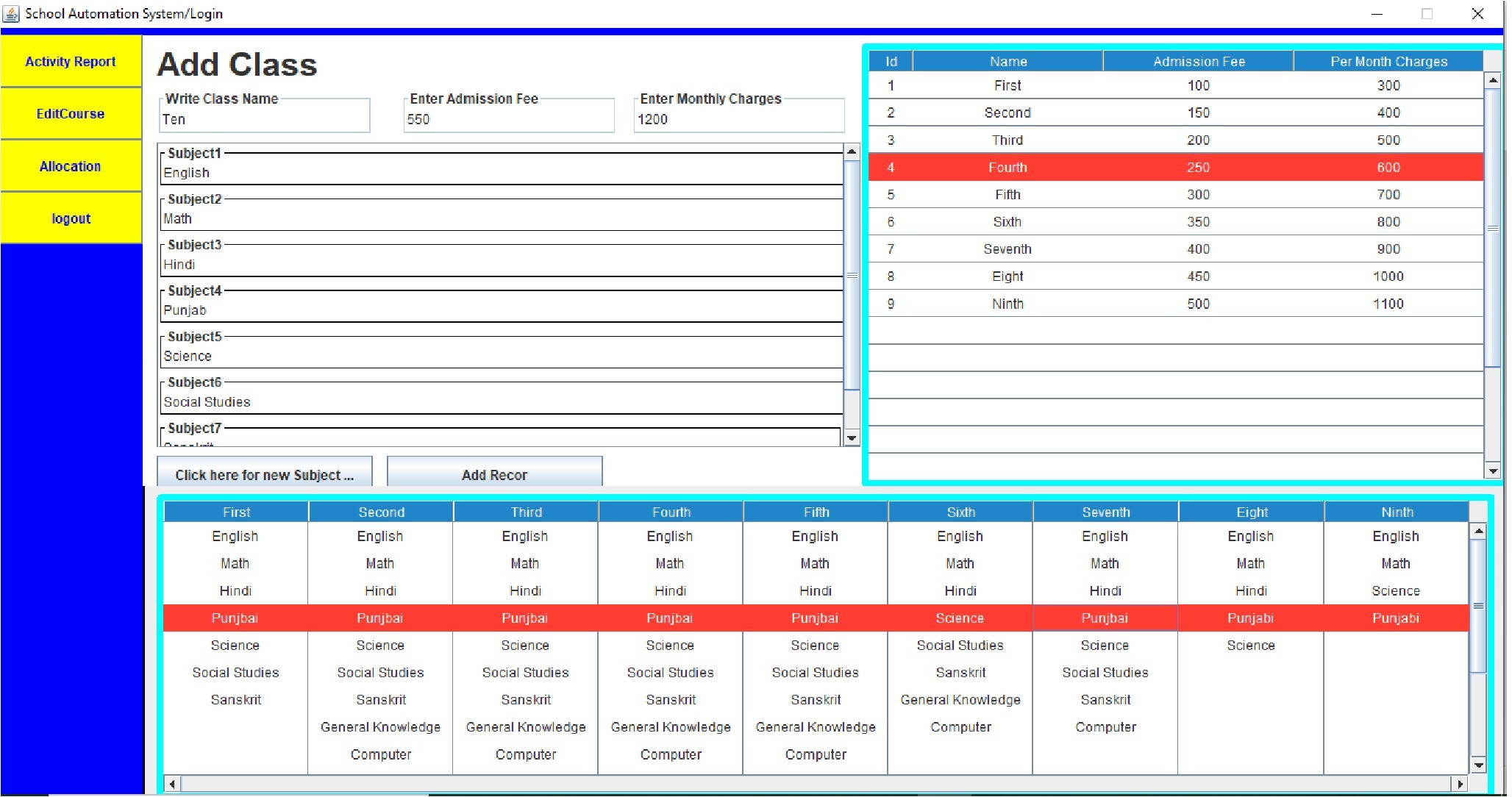
## Add Class

Active button

Class Info

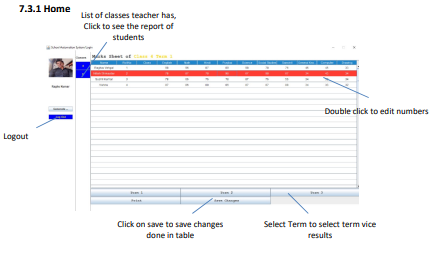
36

Insert new Subject



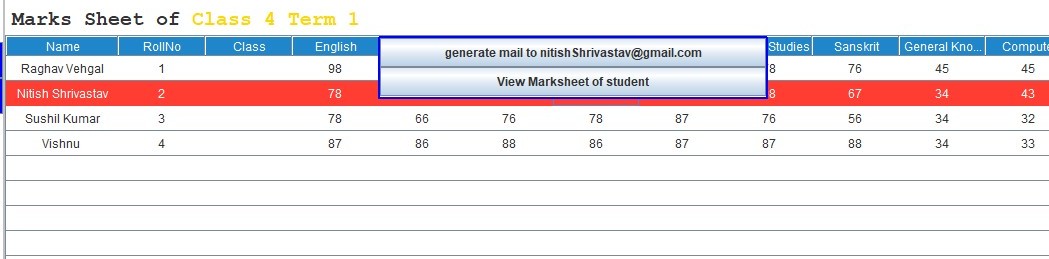
In subject list Class Subjects Info

## Teacher

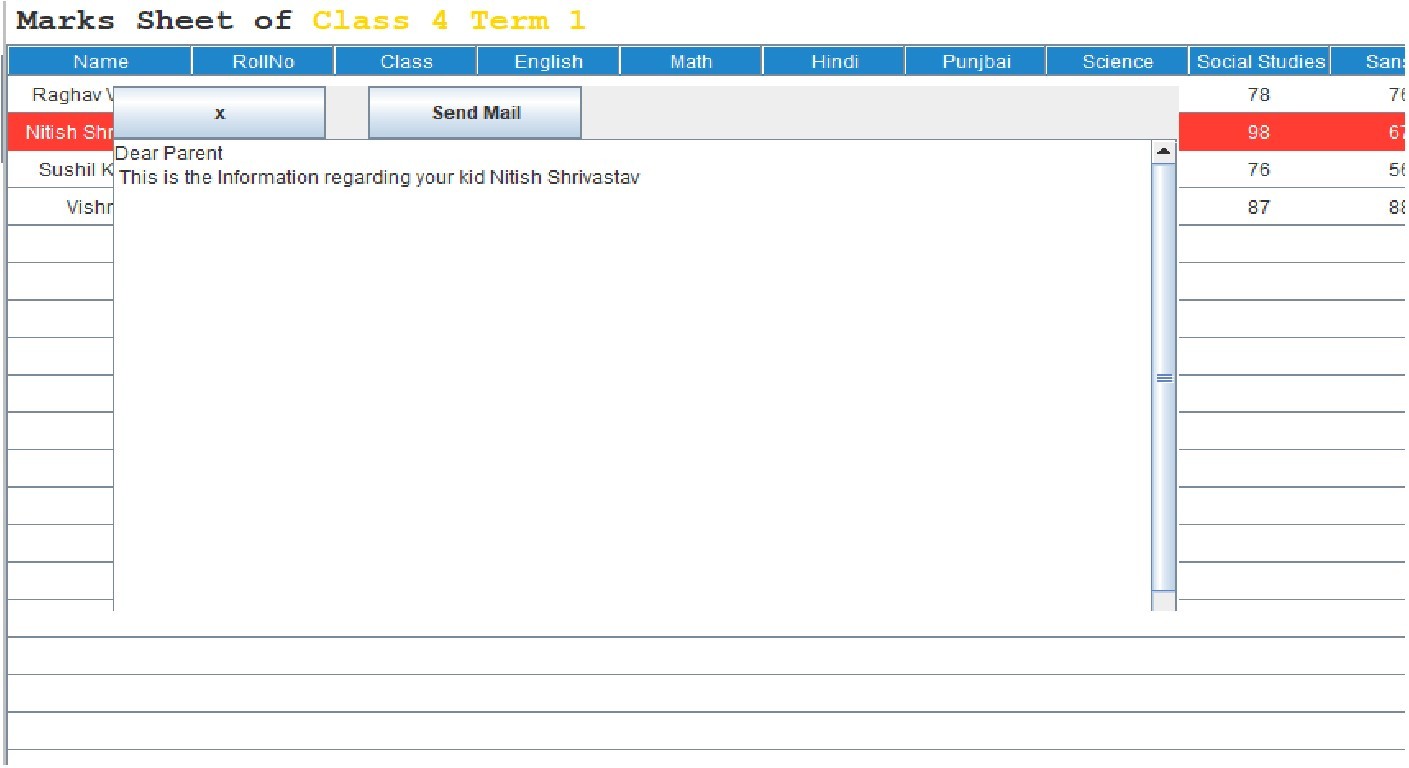


37

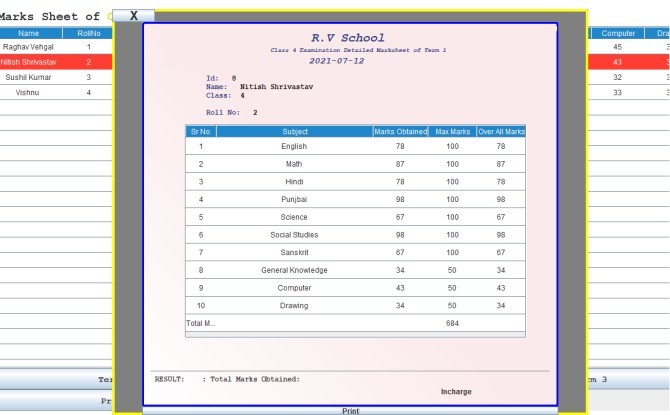
#### Option Pane



Right click on the row and upper box will appear With two option

**Mail Generator Pane:** After clicking on first option it will be the output

**Marks sheet Pane:** After clicking on second option this will be the output

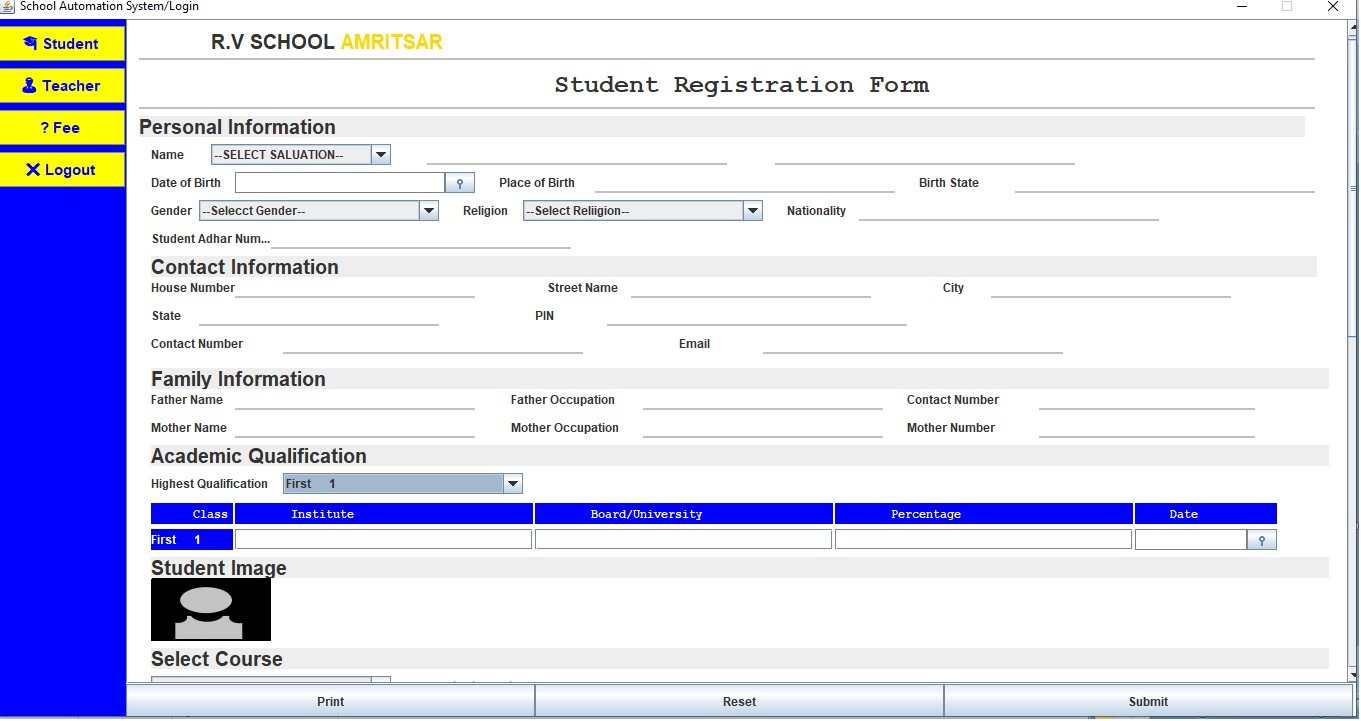


38

## Clerk

#### Student Form

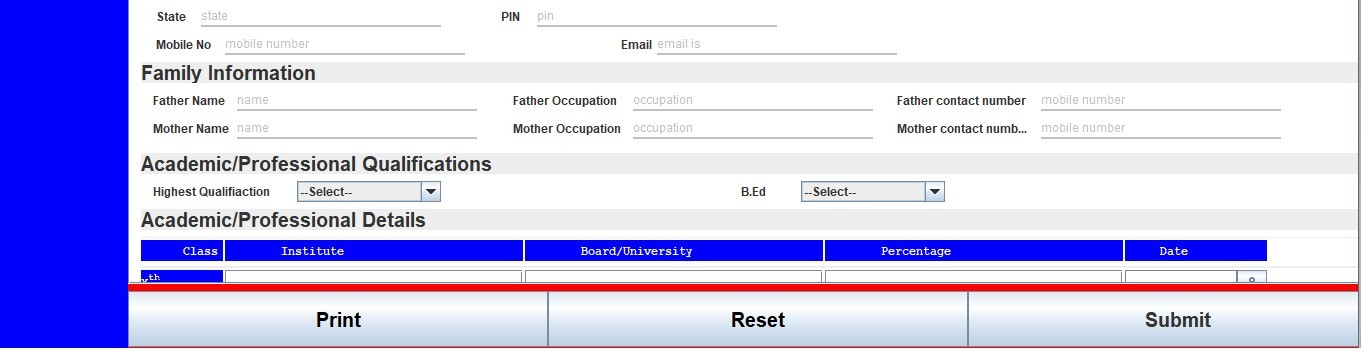
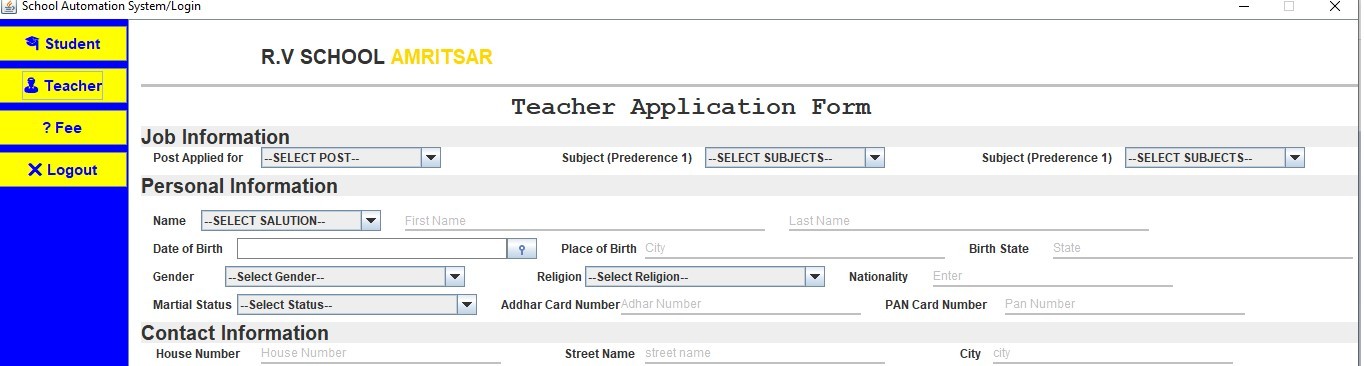
Active Option



#### Teacher Form

Click on submit student information will be stored in database and system will assign student id and class rollno to student

Active Option

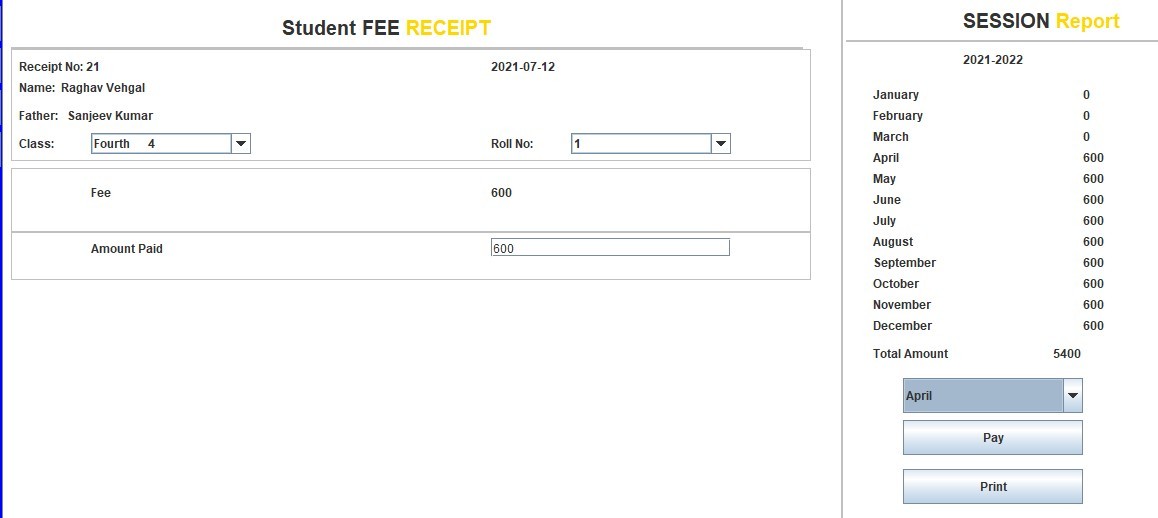


Click on submit, to store data and generate Id , password

#### Fee

Student Info

Select Class Choose roll no



Info Panel

39

# SOFTWARE USED

## NetBeans IDE:-

IDE stands for Integrated Development Environment. This software used to develop and maintain project done in JAVA Programming language. This System has its own output shell, This IDE is smart Enough to convert normal java text file to class file and store in particular folder.

This IDE is best for JAVA Developers. He can build the project like if your project is completed by using IDE build functions we can create jar file of the project and can make it one click run application.

NetBeans has inbuilt Java Development Kit (JDK) and Java Runtime Environment (JRE) which help to run and execute the java and class file. He will create packages automatically in desired location which store all the java files.

With its functionality we can add other jar or library files in over project.

JAR files are nothing just a executable code written in java.

## External Jar File-:

To make my system perform some other task like deal with server or database we need some jar file to built a bridge between two different software. Like I want to established a connection between my java project and MySql database to make this thing possible Java uses a external jar file software mysql\_connector.jar file.

* + - **mysql-connector-java-8.0.15.jar-** This jar file is used to make connection between java application and MySql workbench. All the queries are exchanged and manage by this jar file. All the Mysql answers and responses is first managed by this jar file and then send to the java project.
    - **activation.jar and mail.jar-** This two jar file works with each other to established a connection with any Email threw Internet. With the help of these jar files we can generate and mail to any person who has email account.
    - **swingx.jar-** This jar files is used to create a datepicker module which I used in this project to let the user to choose date from calendar. Swingx.jar is extended version of swing in javax.

40

## MySql:-

MySql is a database software. Here we can create and manage database or data tables. MySql comes with many facilities user can create and modify table without using qurery. To Install server with MySql is your choice. MySql come with things like-

**MySql Shell-** This is just like command prompt of Mysql, where we write multiple commands and queries to view and modify tables, create table, triggers etc.

**MySql Workbench-** This if fully automated system it helps to create and modify tables without any coding. All things are done by clicking and selecting with mouse. And coding part is done by the software automatically depending on your changes

41

# CONCLUSION AND LIMITATION

## Conclusion:-

Here I come to end of my project. I did all the possible things and make the system helping hand for schools management to perform basics activity of school. My School Automation do all the things which is expected to be done. My application can generate E-Mails, store and manage student record, able to generate mark-sheet, can update marks of student. The System is able to show all terms exam marks.

No need to look for other application to write content of email or to open any other application , mail generation thing can be done by the system, User can write the mail before sending it, Some mails are auto generated by the developer.

For every new teacher or clerk joining always generate a Id and Password and send to there respected mail Id automatically.

Every Teacher is only able to maintain and modify the class data to which he is allocated. If teacher is allocated to 4th class he can see only the report of 4th class and can send mail to only 4th class student. He/she can send n number of mails to all class with just one click of button.

Admin is able to see all daily transaction , and can add new class with there subjects and fees etc.

At the End I just want to say this system is user friendly, and can perform all the tasks which I mentioned already in upper paragraph.

## Limitation:-

As you know nothing is perfect in this world, my system has some

limitation.

In Fee system only single month fee can be entered. System Look and feel cannot be modified by the Admin.

If the System file does not found it will raise run time error and it cannot be handle by any user.

42

# BIBILIGRAPHY

|  |  |  |
| --- | --- | --- |
| S.no | Name | Link |
| 1 | W3School | https://[www.w3.org/](http://www.w3.org/) |
| 2 | Tutorials Point | https://[www.tutorialspoint.com/index.h](http://www.tutorialspoint.com/index.h) tm |
| 3 | JavaTPoint | https://[www.javatpoint.com/java-](http://www.javatpoint.com/java-) tutorial |
| 4 | GeeksforGeeks | https://[www.geeksforgeeks.org](http://www.geeksforgeeks.org/) |

43