**A**

**PROJECT REPORT**

**ON**

**PROJECT “*Fee Management System*”**

Submitted in partial fulfillment of the requirements

For the award of degree of

**Bachelor of Technology**

**In**

**Computer Science and Engineering**

**Submitted by: Submitted to :**

Aryan Chaturvedi(1906490100007)

Ankur Yadav(1906490100005)

Deepak Singh(1906490100012)

Vikas Rajbhar(1906490100039)



**M.G. INSTITUTE OF MANAGEMENT AND TECHNOLOGY**

**BANTHRA, LUCKNOW -226401**

**Affiliated To**

**DR. A.P.J ABDUL KALAM TECHNICAL UNIVERSITY, LUCKNOW**

**DECLARATION**

We, hereby declare that the work, which is being represented in the report “**Fee Report Management**” in partial fulfillment of the requirement of the award of Bachelor of Technology in **Computer Science and Engineering** and submitted to the Department of Computer Science and engineering ,M.G Institute of management and technology, Lucknow.

The contents of this report, in full or in the parts, have not been submitted to any other Institute or University for the award of any degree and are free from any plagiarism.

Aryan Chaturvedi

Ankur Yadav

Deepak Singh

Vikas Rajbhar

(B.tech CSE)

**ACKNOWLEDGEMENT**

During the course of project “Fee Report Management”, we received guidance and encouragement from many. This Acknowledgement is a small tribute to all of them. We have taken efforts in this project. However, it would not have been possible without the kind support and help of many individuals. We would like to thank all of them. My Sincere thanks to Respected Director Sir and Head of Department, Mr. Deepesh Kumar and the members of faculty for providing me wonderful opportunities and giving me a chance to have exposure to working conditions in the company. We feel honoured to work under her guidance. This project would not have been possible without her guidance.

**DEPARTMENT OF COMPUTER SCIENCE&ENGINEERING**

**CERTIFICATION**

This is to certify that the project titled “Fee Management System” is the bona fide work carried out by Aryan Chaturvedi, Ankur Yadav, Deepak Singh, Vikas Rajbhar students of B.Tech Computer Science and Engineering of M.G Institute of management and technology affiliated to AKTU,Uttar Pradesh in partial fulfillment of the requirements of Bachelor of Technology in CSE, under the supervision of Mr. Deepesh Singh , HOD computer Science and Engineering.

Deepesh Singh

(Head of Department)

The <Fees Management System= is a desktop application with specialized

majorities in the field of Fees management. Allow to admin to includes the

information of Courses and Fees details. Allow customers to check, modify the

information, and print data, fand verify the information.

The <Fees Management System= is a desktop application with specialized

majorities in the field of Fees management. Allow to admin to includes the

information of Courses and Fees details. Allow customers to check, modify the

information, and print data, fand verify the information.

**ABSTRACT**

The “Fees Management System” is a desktop application with specialized majorities in the field of Fees management .Allow the admin to includes the information of course fee and fee details of students.

It is a management system aiming to give safe storing of fee details and courses. The most useful and reliable functions which cannot be find in other systems. The project is built using Eclipse IDE and MySQL workbench.

**INDEX**

|  |  |  |
| --- | --- | --- |
| **S.no** | **Contents** | **Page no.** |
| i | DECLARATION | 2 |
| ii | ACKNOWLEDGEMENT | 3 |
| iii | CERTIFICATION | 4 |
| iv | ABSTRACT | 5 |
| v | INTRODUCTION | 7-9 |
| vi | SYSTEM ANALYSIS | 10-14 |
| vii | DEVELOPMENT ENVIRONMENT | 15-18 |
| viii | FUTURE ENHANCEMENT | 19 |
| ix | CODING | 20-46 |
| x | IMPLEMENATION AND TESTING | 47-53 |
| xi | CONCLUSION | 54 |
| xii | BIBILIOGRAPHY | 55 |

**INTRODUCTION TO PROJECT**

The main Function of this system is to develop fee management system by using Java language. This system mainly reduces the work task and it is easy to maintain the records for a long time than normal hand written records. The user can check his record details by just entering his name no need to search all the record. With the help of this system fee calculations can be done very easily by this system. So the maintenance and management of fee became very easy. This System also allows to insert any number of records and also stores it in the database that is connected by using JDBC concept.

Classes used in the project are as follows:-

* Accountant
* Accountant Dao
* Accountant login
* Admin Login
* Add Student
* Add Accountant
* Student
* View Student
* View accountant
* Fee report
* Due fees
* Edit Student

**PROBLEM – SPECIFICATION**

The aim of problem statement was to design a model:-

* The model must be user-friendly
* It should not allow viewing others data
* The data privileges should be easily viewed
* By all this features the admin can handle data easily

**SCOPE OF THE PROJECT**

• To reduce paperwork

• To make storage of information more efficient and secure.

• To have a friendly interface.

• To operate it easily and with minimum experience.

• To save time and energy of the admin

**SYSTEM ANALYSIS**

**I. FEASIBILITY STUDY**

An initial investigation in a proposal that determines whether an alternative system is feasible. A proposal summarizing the thinking of the analyst is presented to the user for review. When approved, the proposal initiates feasibility study that describes and evaluates candidate systems and provides for the selection of best system that meets system performance requirements. To do a feasibility study, we need to consider the economic, technical factors in system development. First a project team is formed. The team develops system flowcharts that identify the characteristics of candidate systems, evaluate the performance of each system, weigh system performance and cost data and select the best candidate system for the job. The study culminates in a final report to the management.

**II. INTRODUCTION**

1. Describe and identify characteristics of fee report system.

2. Determine and evaluate performance and cost effectiveness of each Student system.

3. Weigh system performance and cost data.

**III.SUMMARY AND CONSTRAINTS**

1. A feasibility study is conducted to select the best system that meets performance requirements. This entails an identification description, an evaluation of candidate systems, and the selection of the best system for the job.

2. A statement of constraints, the identification of specific system objectives and a description of outputs define a system’s required performance. The analyst is then ready to evaluate the feasibility of candidate systems to produce these outputs.

3. Three key considerations are involved in feasibility analysis: economic, technical and behavioural.

4. There are few steps in feasibility study.

**a. STATEMENT OF CONSTRAINTS:** - Constraints are factors that limit the solution of a problem. Some constraints are identified during the initial investigation.

**b. IDENTIFICATION OF SPECIFIC SYSTEM OBJECTIVES:** - Once the constraints are spelled out, the analyst proceeds to identify the system’s specific performance objectives. They are derived from the general objectives specified in the project directive at the end of the initial investigation. The steps are to state the system’s benefits and then translate them into measurable objectives.

**IV.TYPES OF FEASIBLE STUDY**

**Legal Feasibility:** - Determines whether the proposed system conflicts with legal requirements, e.g. a data processing system must comply with the local Data Protection Acts.

**Operational Feasibility:** -Operational feasibility is a measure of how well a proposed system solves the problems, and takes advantage of the opportunities identified during scope definition and how it satisfies the requirements identified in the requirements analysis phase of system development. The operational feasibility assessment focuses on the degree to which the proposed development projects fits in with the existing business environment and objectives with regard to development schedule, delivery date, corporate culture, and existing business processes. To ensure success, desired operational outcomes must be imparted during design and development. These include such design dependent parameters such as reliability, maintainability, supportability, usability, predictability, disposability, sustainability, affordability and others. These parameters are required to be considered at the early stages of design if desired operational 16 behaviours are to be realized. A system design and development requires appropriate and timely application of engineering and management efforts to meet the previously mentioned parameters. A system may serve its intended purpose most effectively when its technical and operating characteristics are engineered into the design. Therefore operational feasibility is a critical aspect of systems engineering that needs to be an integral part of the early design phases.

**Economic Feasibility:** -The purpose of the economic feasibility assessment is to determine the positive economic benefits to the organization that the proposed system will provide. It includes quantification and identification of all the benefits expected. This assessment typically involves a cost/ benefits analysis.

**Technical Feasibility:** -The technical feasibility assessment is focused on gaining an understanding of the present technical resources of the organization and their applicability to the expected needs of the proposed system. It is an evaluation of the hardware and software and how it meets the need of the proposed system.

**V.HARDWARE AND SOFTWARE REQUIREMENTS**

**HARDWARE REQUIREMENTS:**

**• PROCESSOR**: Pentium IV processor or Greater

**• RAM**: 128 Mega Byte (MB) or Greater

**• HARDDISK**: 1.2 Giga Byte (GB) or Greater

**•** Keyboard & Mouse

**SOFTWARE REQUIREMENTS:**

**• Operating System**: Windows 2000/ XP /7/8

**• IDE:** Eclipse IDE

**• Database:** MySQL (MySQL Workbench)

**DEVELOPMENT ENVIRONMENT**

**I. INTRODUCTION TO JAVA**

• Java is an object oriented programming language originally developed by Sun Microsystems and released in 1995.

• Java was originally developed by James Gosling at Sun Microsystems (which has since merge into Oracle Corporation).

• Java code that runs on one platform does not need to be recompiled to run on another platform, it’s called “write once, run anywhere” (WORA).

• Java programs are platform independent which means they can be run on any operating system with any type of processor as long as the Java interpreter is available on that system.

**USES OF JAVA:**

Earlier, java was only used to design and program small computing devices but later adopted as one of the platform independent programming language and now according to Sun, 3 billion devices run java. Java is one of the most important programming language in today’s IT industries.

• **JSP** – Java is used to create web applications like PHP and ASP, JSP (Java Server Pages) used with normal HTML tags, which helps to create dynamic web pages.

• **Applets** – This is another type of Java program that used within a web page to add many new features to a web browser.

• **J2EE** – The software Java 2 Enterprise Edition are used by various companies to transfer data based on XML structured documents between one another.

• **JavaBeans** – This is something like Visual Basic, a reusable software component that can be easily assemble to create some new and advanced application.

• **Mobile** – Besides the above technology, Java is also used in mobile devices, many kind of games and services built in Java. Today, all leading mobile service provider like Nokia, Siemens, Vodafone are using Java technology.

**FACTS ABOUT JAVA** :

• **Object** **Oriented** – In java everything is an Object. Java can be easily expanded since it is based on the Object model.

• **Platform** **independent** – C and C++ are platform dependency languages hence the application programs written in one Operating system cannot run in any other Operating system, but in platform independence language like Java application programs written in one Operating system can able to run on any Operating system.

• **Simple** – Java is designed to be easy to learn. If you understand the basic concept of OOP java would be easy to master.

• **Secure** – With Java’s secure feature it enables to develop virus-free, tamper-free systems. Authentication techniques are based on public key encryption.

• **Robust** – Java makes an effort to eliminate error prone situations by emphasizing mainly on compile time error checking and runtime checking.

• **Multithreaded** – With Java’s multithreaded feature it is possible to write programs that can do many tasks simultaneously. This design feature allows developers to construct smoothly running interactive applications.

• **Interpreted** – Java byte code is translated on the fly to native machine instructions and is not stored anywhere. The development process is more rapid and analytical since the linking is an incremental and light weight process.

• **Distributed** – Java is designed for the distributed environment of the internet.

• **Dynamic** – Java is considered to be more dynamic than C or C++ since it is designed to adapt to an evolving environment. Java programs can carry an extensive amount of runtime information that can be used to verify and resolve accesses to objects on runtime.

**Popular Java Editors**: To write your java programs you will need a text editor. There are even more sophisticated IDE available in the market. But for now, you can consider one of the following:

• **IntelliJ Idea**– is a Java IDE that is open source and free which can be downloaded from

• **Eclipse** – is also a java IDE developed by the eclipse open source community.

**INTRODUCTION TO Eclipse IDE**

* Eclipse is an integrated development environment used in computer programming. It contains a base workspace and an extensible plug-in system for customizing the environment. It is the second-most-popular IDE for Java development, and, until 2016, was the most popular.



* The Eclipse Web Tools Platform (WTP) project is an extension of the Eclipse platform with tools for developing Web and Java EE applications. It includes source and graphical editors for a variety of languages, wizards and built-in applications to simplify development, and tools and APIs to support deploying, running, and testing apps.

**MySQL CONNECTOR**

In the MySQLI documentation, the term connector refers to a piece of

software that allows application to connect to the MySQLI database server. MySQLI

provides connectors for a variety of languages, including PHP.

In Java application needs to communicate with a database server will need to write Java code to perform such activities as connecting to the database server,

querying the database and other database related functions.

Software is required to provide the API that Java application will use and also handle

the communication between application and the database server, possible using other

intermediate libraries where necessary. This software is known generically as a

connector, as it allows application to connector a database server.

**SYSTEM-REQUIREMENTS**

|  |  |  |
| --- | --- | --- |
| Requirement | Minimum | Recommended |
|  |  |  |
| RAM | 2 GB of free RAM | 8 GB of total system RAM |
| CPU | Any modern CPU | Multi-core CPU. Eclipse IDE supports multithreading for different operations and processes making it faster the more CPU cores it can use. |
| Disk space | 2.5 GB and another 1 GB for caches | SSD drive with at least 5 GB of free space |
| Monitor resolution | 1024×768 | 1920×1080 |
| Operating system | Officially released 64-bit versions of the following:   * Microsoft Windows 8 or later * macOS 10.14 or later * Any Linux distribution that supports Gnome, KDE, or Unity DE.   Pre-release versions are not supported. | Latest 64-bit version of Windows, macOS, or Linux (for example, Debian, Ubuntu, or RHEL) |

**JVM languages﻿**

Use Eclipse IDE develop applications in the following languages that can be compiled into the JVM byte code, namely:

* [Java](https://www.oracle.com/java/technologies/javase-downloads.html)
* [Kotlin](https://kotlinlang.org/)
* [Scala](https://www.scala-lang.org/)
* [Groovy](https://groovy-lang.org/)

**FUTURE ENHANCEMENT**

Educational institution contains a lot of student enrolled in different fields of education .It is very hard to keep the records of all these students in the hand-written formatted documents.

To solve this problem fee report system should be enhanced in future-

* Many other useful modules can be added.
* A class to calculate the fees of different student.
* Based on the future security issues, Security can be improved using emerging technology.

**CODING**

**ACCOUNTANT()-**

package codeclause1;

public class Accountant {

private int id;

private String name,password,email,contactno;

public Accountant() {}

public Accountant(String name, String password, String email, String contactno) {

super();

this.name = name;

this.password = password;

this.email = email;

this.contactno = contactno;

}

public int getId() {

return id;

}

public void setId(int id) {

this.id = id;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public String getPassword() {

return password;

}

public void setPassword(String password) {

this.password = password;

}

public String getEmail() {

return email;

}

public void setEmail(String email) {

this.email = email;

}

public String getContactno() {

return contactno;

}

public void setContactno(String contactno) {

this.contactno = contactno;}

**Accountant Dao()-**

package codeclause1;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.util.ArrayList;

import java.util.List;

public class AccountantDao {

public static Connection getCon(){

Connection con=null;

try{

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

con=DriverManager.getConnection("jdbc:mysql://localhost:3306/test","root","Aryanch@12");

}catch(Exception e){System.out.println(e);}

return con;

}

public static boolean validate(String name,String password){

boolean status=false;

try{

Connection con=getCon();

PreparedStatement ps=con.prepareStatement("select \* from feereport\_accountant where name=? and password=?");

ps.setString(1,name);

ps.setString(2,password);

ResultSet rs=ps.executeQuery();

status=rs.next();

con.close();

}catch(Exception e){System.out.println(e);}

return status;

}

public static int save(Accountant a){

int status=0;

try{

Connection con=getCon();

PreparedStatement ps=con.prepareStatement("insert into feereport\_accountant(name,password,email,contactno) values(?,?,?,?)");

ps.setString(1,a.getName());

ps.setString(2,a.getPassword());

ps.setString(3,a.getEmail());

ps.setString(4,a.getContactno());

status=ps.executeUpdate();

con.close();

}catch(Exception e){System.out.println(e);}

return status;

}

public static List<Accountant> view(){

List<Accountant> list=new ArrayList<>();

try{

Connection con=getCon();

PreparedStatement ps=con.prepareStatement("select \* from feereport\_accountant");

ResultSet rs=ps.executeQuery();

while(rs.next()){

Accountant a=new Accountant();

a.setId(rs.getInt(1));

a.setName(rs.getString(2));

a.setPassword(rs.getString(3));

a.setEmail(rs.getString(4));

a.setContactno(rs.getString(5));

list.add(a);

}

con.close();

}catch(Exception e){System.out.println(e);}

return list;

}

}

**Accountant.Login()-**

package codeclause1;

import java.awt.BorderLayout;

import java.awt.EventQueue;

import javax.swing.JFrame;

import javax.swing.JPanel;

import javax.swing.border.EmptyBorder;

import javax.swing.GroupLayout;

import javax.swing.GroupLayout.Alignment;

import javax.swing.JLabel;

import javax.swing.JOptionPane;

import java.awt.Color;

import java.awt.Font;

import java.awt.Image;

import java.awt.Toolkit;

import javax.swing.JTextField;

import javax.swing.JPasswordField;

import javax.swing.JButton;

import javax.swing.LayoutStyle.ComponentPlacement;

import java.awt.event.ActionListener;

import java.awt.event.ActionEvent;

public class AccountantLogin extends JFrame {

static AccountantLogin frame;

private JPanel contentPane;

private JTextField textField;

private JPasswordField passwordField;

/\*\*

\* Launch the application.

\*/

public static void main(String[] args) {

EventQueue.invokeLater(new Runnable() {

public void run() {

try {

frame = new AccountantLogin();

frame.setVisible(true);

} catch (Exception e) {

e.printStackTrace();

}

}

});

}

/\*\*

\* Create the frame.

\*/

public AccountantLogin() {

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setBounds(500, 100, 450, 300);

// Image img = Toolkit.getDefaultToolkit().getImage();

contentPane = new JPanel();

contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));

setContentPane(contentPane);

JLabel lblAccountantLogin = new JLabel("Accountant Login");

lblAccountantLogin.setFont(new Font("Tahoma", Font.PLAIN, 20));

lblAccountantLogin.setForeground(Color.DARK\_GRAY);

JLabel lblName = new JLabel("Name:");

textField = new JTextField();

textField.setColumns(10);

JLabel lblPassword = new JLabel("Password:");

passwordField = new JPasswordField();

JButton btnLogin = new JButton("Login");

btnLogin.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent arg0) {

String name=textField.getText();

String password=String.valueOf(passwordField.getPassword());

boolean status=AccountantDao.validate(name, password);

if(status){

AccountantSection.main(new String[]{});

frame.dispose();

}else{

JOptionPane.showMessageDialog(AccountantLogin.this,"Sorry, username or password error!","Login error!",JOptionPane.ERROR\_MESSAGE);

}

}

});

JButton btnBack = new JButton("back");

GroupLayout gl\_contentPane = new GroupLayout(contentPane);

gl\_contentPane.setHorizontalGroup(

gl\_contentPane.createParallelGroup(Alignment.LEADING)

.addGroup(gl\_contentPane.createSequentialGroup()

.addGroup(gl\_contentPane.createParallelGroup(Alignment.LEADING)

.addGroup(gl\_contentPane.createSequentialGroup()

.addGap(28)

.addGroup(gl\_contentPane.createParallelGroup(Alignment.LEADING)

.addComponent(lblPassword)

.addGroup(gl\_contentPane.createSequentialGroup()

.addComponent(lblName)

.addGroup(gl\_contentPane.createParallelGroup(Alignment.LEADING)

.addGroup(gl\_contentPane.createSequentialGroup()

.addGap(76)

.addComponent(lblAccountantLogin))

.addGroup(gl\_contentPane.createSequentialGroup()

.addGap(54)

.addGroup(gl\_contentPane.createParallelGroup(Alignment.LEADING, false)

.addComponent(passwordField)

.addComponent(textField, GroupLayout.DEFAULT\_SIZE, 186, Short.MAX\_VALUE)))))))

.addGroup(gl\_contentPane.createSequentialGroup()

.addGap(158)

.addComponent(btnLogin, GroupLayout.PREFERRED\_SIZE, 81, GroupLayout.PREFERRED\_SIZE)

.addGap(52)

.addComponent(btnBack)))

.addContainerGap(78, Short.MAX\_VALUE))

);

gl\_contentPane.setVerticalGroup(

gl\_contentPane.createParallelGroup(Alignment.LEADING)

.addGroup(gl\_contentPane.createSequentialGroup()

.addContainerGap()

.addComponent(lblAccountantLogin)

.addGap(18)

.addGroup(gl\_contentPane.createParallelGroup(Alignment.BASELINE)

.addComponent(lblName)

.addComponent(textField, GroupLayout.PREFERRED\_SIZE, GroupLayout.DEFAULT\_SIZE, GroupLayout.PREFERRED\_SIZE))

.addGap(18)

.addGroup(gl\_contentPane.createParallelGroup(Alignment.LEADING)

.addComponent(lblPassword)

.addComponent(passwordField, GroupLayout.PREFERRED\_SIZE, GroupLayout.DEFAULT\_SIZE, GroupLayout.PREFERRED\_SIZE))

.addPreferredGap(ComponentPlacement.UNRELATED)

.addGroup(gl\_contentPane.createParallelGroup(Alignment.LEADING)

.addComponent(btnLogin, GroupLayout.PREFERRED\_SIZE, 32, GroupLayout.PREFERRED\_SIZE)

.addComponent(btnBack))

.addContainerGap(96, Short.MAX\_VALUE))

);

contentPane.setLayout(gl\_contentPane);

}

}

**AccountantSection()-**

package codeclause1;

import java.awt.BorderLayout;

import java.awt.EventQueue;

import javax.swing.JFrame;

import javax.swing.JPanel;

import javax.swing.border.EmptyBorder;

import javax.swing.GroupLayout;

import javax.swing.GroupLayout.Alignment;

import javax.swing.JButton;

import javax.swing.JLabel;

import java.awt.Color;

import java.awt.Font;

import javax.swing.LayoutStyle.ComponentPlacement;

import java.awt.event.ActionListener;

import java.awt.event.ActionEvent;

public class AccountantSection extends JFrame {

static AccountantSection frame;

private JPanel contentPane;

/\*\*

\* Launch the application.

\*/

public static void main(String[] args) {

EventQueue.invokeLater(new Runnable() {

public void run() {

try {

frame = new AccountantSection();

frame.setVisible(true);

} catch (Exception e) {

e.printStackTrace();

}

}

});

}

/\*\*

\* Create the frame.

\*/

public AccountantSection() {

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setBounds(500, 100, 450, 423);

contentPane = new JPanel();

contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));

setContentPane(contentPane);

JButton btnNewButton = new JButton("Add Student");

btnNewButton.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

AddStudent.main(new String[]{});

frame.disable();

}

});

JLabel lblAccountantSection = new JLabel("Accountant Section");

lblAccountantSection.setFont(new Font("Tahoma", Font.PLAIN, 20));

lblAccountantSection.setForeground(Color.DARK\_GRAY);

JButton btnViewStudent = new JButton("View Student");

btnViewStudent.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

ViewStudent.main(new String[]{});

}

});

JButton btnEditStudent = new JButton("Edit Student");

btnEditStudent.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent arg0) {

EditStudent.main(new String[]{});

frame.dispose();

}

});

JButton btnDueFee = new JButton("Due Fee");

btnDueFee.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

DueFee.main(new String[]{});

}

});

JButton btnLogout = new JButton("Logout");

btnLogout.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent arg0) {

FeeReport.main(new String[]{});

frame.dispose();

}

});

GroupLayout gl\_contentPane = new GroupLayout(contentPane);

gl\_contentPane.setHorizontalGroup(

gl\_contentPane.createParallelGroup(Alignment.LEADING)

.addGroup(gl\_contentPane.createSequentialGroup()

.addGroup(gl\_contentPane.createParallelGroup(Alignment.LEADING)

.addGroup(gl\_contentPane.createSequentialGroup()

.addGap(136)

.addComponent(lblAccountantSection))

.addGroup(gl\_contentPane.createSequentialGroup()

.addGap(52)

.addGroup(gl\_contentPane.createParallelGroup(Alignment.LEADING, false)

.addGroup(gl\_contentPane.createSequentialGroup()

.addComponent(btnEditStudent, GroupLayout.PREFERRED\_SIZE, 133, GroupLayout.PREFERRED\_SIZE)

.addPreferredGap(ComponentPlacement.RELATED, GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(btnDueFee, GroupLayout.PREFERRED\_SIZE, 133, GroupLayout.PREFERRED\_SIZE))

.addGroup(gl\_contentPane.createSequentialGroup()

.addComponent(btnNewButton, GroupLayout.PREFERRED\_SIZE, 133, GroupLayout.PREFERRED\_SIZE)

.addGap(53)

.addComponent(btnViewStudent, GroupLayout.PREFERRED\_SIZE, 133, GroupLayout.PREFERRED\_SIZE))))

.addGroup(gl\_contentPane.createSequentialGroup()

.addGap(144)

.addComponent(btnLogout, GroupLayout.PREFERRED\_SIZE, 133, GroupLayout.PREFERRED\_SIZE)))

.addContainerGap(53, Short.MAX\_VALUE))

);

gl\_contentPane.setVerticalGroup(

gl\_contentPane.createParallelGroup(Alignment.LEADING)

.addGroup(gl\_contentPane.createSequentialGroup()

.addGap(7)

.addComponent(lblAccountantSection)

.addGap(25)

.addGroup(gl\_contentPane.createParallelGroup(Alignment.BASELINE)

.addComponent(btnNewButton, GroupLayout.PREFERRED\_SIZE, 36, GroupLayout.PREFERRED\_SIZE)

.addComponent(btnViewStudent, GroupLayout.PREFERRED\_SIZE, 36, GroupLayout.PREFERRED\_SIZE))

.addGap(35)

.addGroup(gl\_contentPane.createParallelGroup(Alignment.BASELINE)

.addComponent(btnEditStudent, GroupLayout.PREFERRED\_SIZE, 36, GroupLayout.PREFERRED\_SIZE)

.addComponent(btnDueFee, GroupLayout.PREFERRED\_SIZE, 36, GroupLayout.PREFERRED\_SIZE))

.addGap(36)

.addComponent(btnLogout, GroupLayout.PREFERRED\_SIZE, 36, GroupLayout.PREFERRED\_SIZE)

.addContainerGap(138, Short.MAX\_VALUE))

);

contentPane.setLayout(gl\_contentPane);

}

**}**

**Admin ()-**

package codeclause1;

import java.awt.BorderLayout;

import java.awt.EventQueue;

import javax.swing.JFrame;

import javax.swing.JPanel;

import javax.swing.border.EmptyBorder;

import javax.swing.GroupLayout;

import javax.swing.GroupLayout.Alignment;

import javax.swing.JLabel;

import javax.swing.JOptionPane;

import java.awt.Font;

import java.awt.Color;

import javax.swing.JTextField;

import javax.swing.JPasswordField;

import javax.swing.JButton;

import javax.swing.LayoutStyle.ComponentPlacement;

import java.awt.event.ActionListener;

import java.awt.event.ActionEvent;

public class AdminLogin extends JFrame {

static AdminLogin frame;

private JPanel contentPane;

private JTextField textField;

private JPasswordField passwordField;

/\*\*

\* Launch the application.

\*/

public static void main(String[] args) {

EventQueue.invokeLater(new Runnable() {

public void run() {

try {

frame = new AdminLogin();

frame.setVisible(true);

} catch (Exception e) {

e.printStackTrace();

}

}

});

}

/\*\*

\* Create the frame.

\*/

public AdminLogin() {

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setBounds(500, 100, 450, 300);

contentPane = new JPanel();

contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));

setContentPane(contentPane);

JLabel lblAdminLogin = new JLabel("Admin Login");

lblAdminLogin.setForeground(Color.DARK\_GRAY);

lblAdminLogin.setFont(new Font("Tahoma", Font.PLAIN, 20));

JLabel lblName = new JLabel("Name:");

JLabel lblPassword = new JLabel("Password:");

textField = new JTextField();

textField.setColumns(10);

passwordField = new JPasswordField();

JButton btnLogin = new JButton("login");

btnLogin.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

String name=textField.getText();

char ch[]=passwordField.getPassword();

String password=String.valueOf(ch);

if(name.equals("admin")&&password.equals("admin123")){

String s[]={};

AdminSection.main(s);

frame.dispose();

}else{

JOptionPane.showMessageDialog(AdminLogin.this,"Sorry, username or password error!");

textField.setText("");passwordField.setText("");

}

}

});

GroupLayout gl\_contentPane = new GroupLayout(contentPane);

gl\_contentPane.setHorizontalGroup(

gl\_contentPane.createParallelGroup(Alignment.LEADING)

.addGroup(gl\_contentPane.createSequentialGroup()

.addGroup(gl\_contentPane.createParallelGroup(Alignment.LEADING)

.addGroup(gl\_contentPane.createSequentialGroup()

.addGap(180)

.addComponent(lblAdminLogin))

.addGroup(gl\_contentPane.createSequentialGroup()

.addGap(25)

.addGroup(gl\_contentPane.createParallelGroup(Alignment.LEADING)

.addComponent(lblName)

.addComponent(lblPassword))

.addGap(58)

.addGroup(gl\_contentPane.createParallelGroup(Alignment.TRAILING, false)

.addComponent(passwordField)

.addComponent(textField, GroupLayout.DEFAULT\_SIZE, 180, Short.MAX\_VALUE)))

.addGroup(gl\_contentPane.createSequentialGroup()

.addGap(177)

.addComponent(btnLogin, GroupLayout.PREFERRED\_SIZE, 86, GroupLayout.PREFERRED\_SIZE)))

.addContainerGap(111, Short.MAX\_VALUE))

);

gl\_contentPane.setVerticalGroup(

gl\_contentPane.createParallelGroup(Alignment.LEADING)

.addGroup(gl\_contentPane.createSequentialGroup()

.addComponent(lblAdminLogin)

.addGap(29)

.addGroup(gl\_contentPane.createParallelGroup(Alignment.BASELINE)

.addComponent(lblName)

.addComponent(textField, GroupLayout.PREFERRED\_SIZE, GroupLayout.DEFAULT\_SIZE, GroupLayout.PREFERRED\_SIZE))

.addGap(27)

.addGroup(gl\_contentPane.createParallelGroup(Alignment.BASELINE)

.addComponent(passwordField, GroupLayout.PREFERRED\_SIZE, GroupLayout.DEFAULT\_SIZE, GroupLayout.PREFERRED\_SIZE)

.addComponent(lblPassword))

.addGap(36)

.addComponent(btnLogin, GroupLayout.PREFERRED\_SIZE, 33, GroupLayout.PREFERRED\_SIZE)

.addContainerGap(51, Short.MAX\_VALUE))

);

contentPane.setLayout(gl\_contentPane);

}

}

**AdminSection()-**

package codeclause1;

import java.awt.BorderLayout;

import java.awt.EventQueue;

import javax.swing.JFrame;

import javax.swing.JPanel;

import javax.swing.JScrollPane;

import javax.swing.JTable;

import javax.swing.border.EmptyBorder;

import javax.swing.GroupLayout;

import javax.swing.GroupLayout.Alignment;

import javax.swing.JLabel;

import java.awt.Font;

import java.awt.Color;

import javax.swing.JButton;

import java.awt.event.ActionListener;

import java.util.List;

import java.awt.event.ActionEvent;

public class AdminSection extends JFrame {

static AdminSection frame;

private JPanel contentPane;

JScrollPane sp;

/\*\*

\* Launch the application.

\*/

public static void main(String[] args) {

EventQueue.invokeLater(new Runnable() {

public void run() {

try {

frame = new AdminSection();

frame.setVisible(true);

} catch (Exception e) {

e.printStackTrace();

}

}

});

}

/\*\*

\* Create the frame.

\*/

public AdminSection() {

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setBounds(500, 100, 450, 300);

contentPane = new JPanel();

contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));

setContentPane(contentPane);

JLabel lblAdminSection = new JLabel("Admin Section");

lblAdminSection.setForeground(Color.DARK\_GRAY);

lblAdminSection.setFont(new Font("Tahoma", Font.PLAIN, 20));

JButton btnAddAccountant = new JButton("Add Accountant");

btnAddAccountant.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

AddAccountant.main(new String[]{});

frame.dispose();

}

});

JButton btnViewAccountant = new JButton("View Accountant");

btnViewAccountant.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

ViewAccountant.main(new String[]{});

}

});

JButton btnLogout = new JButton("Logout");

btnLogout.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

FeeReport.main(new String[]{});

frame.dispose();

}

});

GroupLayout gl\_contentPane = new GroupLayout(contentPane);

gl\_contentPane.setHorizontalGroup(

gl\_contentPane.createParallelGroup(Alignment.LEADING)

.addGroup(gl\_contentPane.createSequentialGroup()

.addGroup(gl\_contentPane.createParallelGroup(Alignment.LEADING)

.addGroup(gl\_contentPane.createSequentialGroup()

.addGap(161)

.addComponent(lblAdminSection))

.addGroup(gl\_contentPane.createSequentialGroup()

.addGap(149)

.addGroup(gl\_contentPane.createParallelGroup(Alignment.LEADING)

.addComponent(btnViewAccountant, GroupLayout.PREFERRED\_SIZE, 130, GroupLayout.PREFERRED\_SIZE)

.addComponent(btnAddAccountant, GroupLayout.PREFERRED\_SIZE, 130, GroupLayout.PREFERRED\_SIZE)

.addComponent(btnLogout, GroupLayout.PREFERRED\_SIZE, 130, GroupLayout.PREFERRED\_SIZE))))

.addContainerGap(136, Short.MAX\_VALUE))

);

gl\_contentPane.setVerticalGroup(

gl\_contentPane.createParallelGroup(Alignment.LEADING)

.addGroup(gl\_contentPane.createSequentialGroup()

.addComponent(lblAdminSection)

.addGap(29)

.addComponent(btnAddAccountant, GroupLayout.PREFERRED\_SIZE, 34, GroupLayout.PREFERRED\_SIZE)

.addGap(18)

.addComponent(btnViewAccountant, GroupLayout.PREFERRED\_SIZE, 34, GroupLayout.PREFERRED\_SIZE)

.addGap(18)

.addComponent(btnLogout, GroupLayout.PREFERRED\_SIZE, 34, GroupLayout.PREFERRED\_SIZE)

.addContainerGap(59, Short.MAX\_VALUE))

);

contentPane.setLayout(gl\_contentPane);

}

}

**Student()-**

package codeclause1;

public class Student {

private int rollno;

private String name,email,course;

private int fee,paid,due;

private String address,city,state,country,contactno;

public Student() {}

public Student(String name, String email, String course, int fee, int paid, int due, String address, String city,

String state, String country, String contactno) {

super();

this.name = name;

this.email = email;

this.course = course;

this.fee = fee;

this.paid = paid;

this.due = due;

this.address = address;

this.city = city;

this.state = state;

this.country = country;

this.contactno = contactno;

}

public Student(int rollno, String name, String email, String course, int fee, int paid, int due, String address,

String city, String state, String country, String contactno) {

super();

this.rollno = rollno;

this.name = name;

this.email = email;

this.course = course;

this.fee = fee;

this.paid = paid;

this.due = due;

this.address = address;

this.city = city;

this.state = state;

this.country = country;

this.contactno = contactno;

}

public int getRollno() {

return rollno;

}

public void setRollno(int rollno) {

this.rollno = rollno;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public String getEmail() {

return email;

}

public void setEmail(String email) {

this.email = email;

}

public String getCourse() {

return course;

}

public void setCourse(String course) {

this.course = course;

}

public int getFee() {

return fee;

}

public void setFee(int fee) {

this.fee = fee;

}

public int getPaid() {

return paid;

}

public void setPaid(int paid) {

this.paid = paid;

}

public int getDue() {

return due;

}

public void setDue(int due) {

this.due = due;

}

public String getAddress() {

return address;

}

public void setAddress(String address) {

this.address = address;

}

public String getCity() {

return city;

}

public void setCity(String city) {

this.city = city;

}

public String getState() {

return state;

}

public void setState(String state) {

this.state = state;

}

public String getCountry() {

return country;

}

public void setCountry(String country) {

this.country = country;

}

public String getContactno() {

return contactno;

}

public void setContactno(String contactno) {

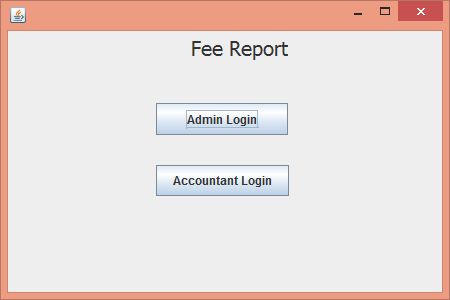
this.contactno = contactno;

}

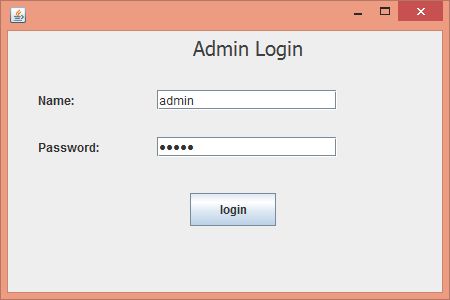
}

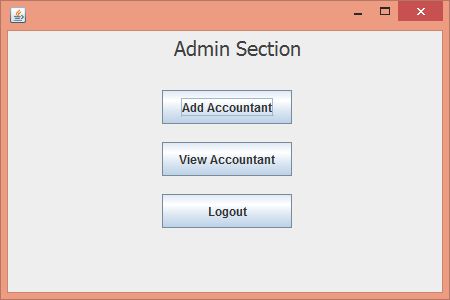
**IMPLEMENTATION AND TESTING**

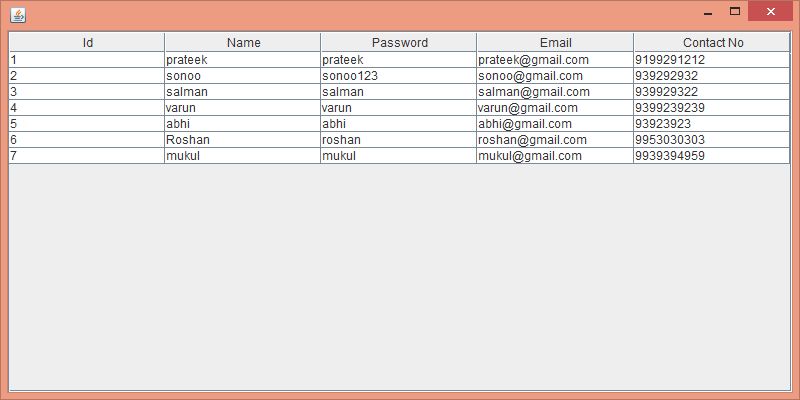
1. Run the Fee Report project.



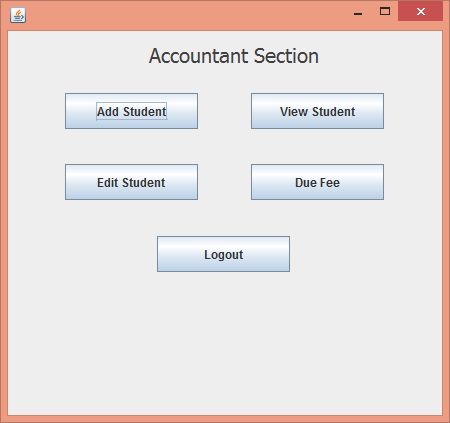
1. By Clicking on the Admin Login or Accountant Login , enter the user name and password.



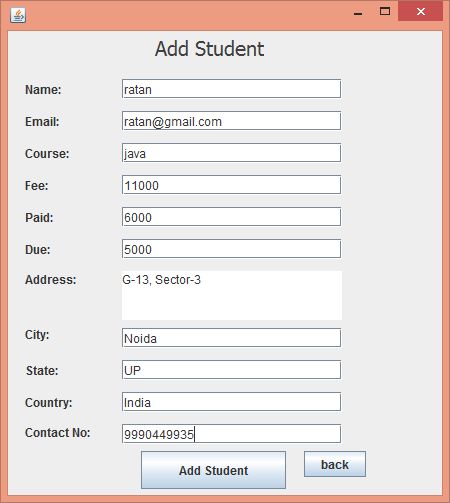
1. Click on Add Accountant.
2. After entering details of new Accountant, click on view Accountant.



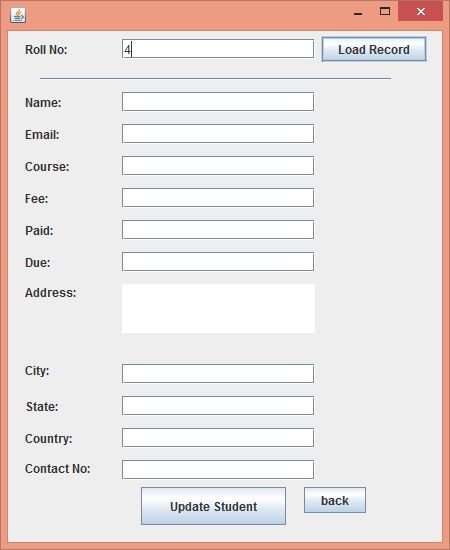
1. Now , Accountant login looks like the –



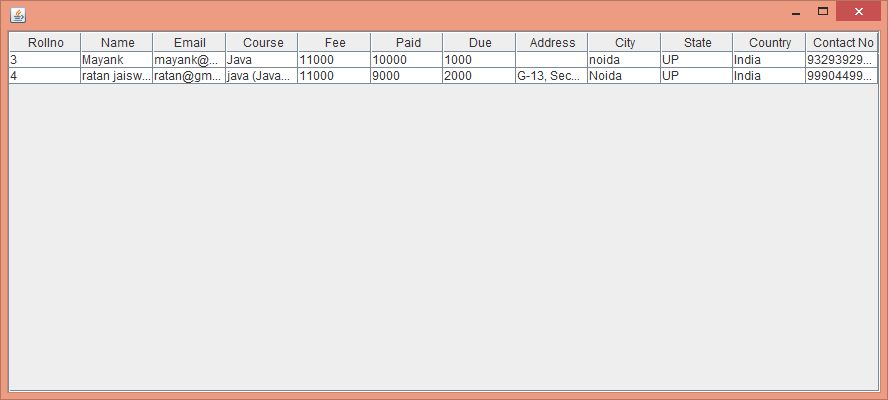
1. Click on Add Student



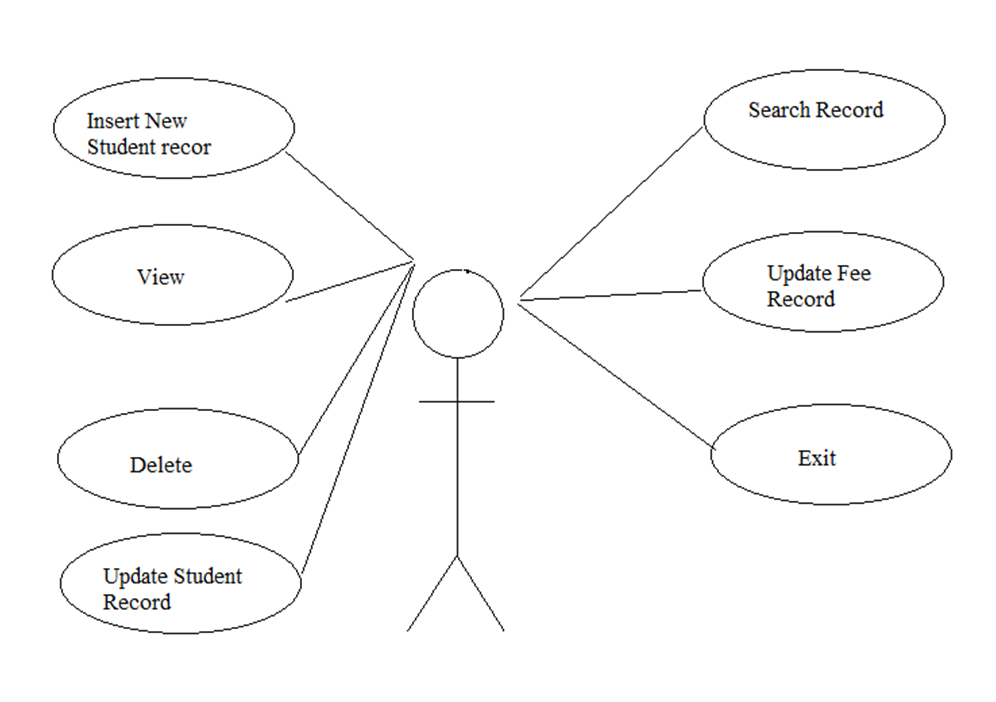
1. We can also edit the information of the existing student in the database.



1. We can check the Due amount by clicking on the due fee.



**FIGURE**



**TESTING**

**TESTING: -** Testing is the process of exercising software with the intent of finding errors and ultimately correcting them. The following testing techniques have been used to make this project free of errors.

**1. Content Review:**

The whole content of the project has been reviewed thoroughly to uncover typographical errors, grammatical error and ambiguous sentences.

**2. Navigation Errors:**

Different users were allowed to navigate through the project to uncover the navigation errors. The views of the user regarding the navigation flexibility and user friendliness were taken into account and implemented in the project.

**3. Unit Testing:**

Focuses on individual software units, groups of related units.

• Unit – smallest testable piece of software.

• A unit can be compiled /assembled / linked/loaded; and put under a test harness.

• Unit testing done to show that the unit does not satisfy the application and /or its implemented software does not match the intended designed structure.

**4. Integration Testing:**

Focuses on combining units to evaluate the interaction among them

• Integration is the process of aggregating components to create larger components.

• Integration testing done to show that even though components were individually satisfactory, the combination is incorrect and inconsistent.

**5. System testing:**

Focuses on a complete integrated system to evaluate compliance with specified requirements (test characteristics that are only present when entire system is run).

• A system is a big component.

• System testing is aimed at revealing bugs that cannot be attributed to a component as such, to inconsistencies between components or planned interactions between components.

**6. Alpha Testing:**

Alpha testing is simulated or actual operational testing by potential users/customers or an independent test team at the developers' site. Alpha testing is often employed for off-the-shelf software as a form of internal acceptance testing, before the software goes to beta testing.

**7. Beta Testing:**

Beta testing comes after alpha testing and can be considered a form of external user acceptance testing. Versions of the software, known as beta versions, are released to a limited audience outside of the programming team. The software is released to groups of people so that further testing can ensure the product has few faults or bugs. Sometimes, beta versions are made available to the open public to increase the feedback field to a maximal number of future users.

**CONCLUSION**

* The project of “Fee Management System” is designed in order to reduce the maintaining bulk of records of all student fees details of who study in an Educational Institution.
* Inserting, retrieving and updating the fees details of a student are easy when it is compared to the manual register and storing. Maintaining the project is also easy which can is easily understandable. Maintaining the database is manageable

**BIBILIOGRAPHY**

**1. www.javatpoint.com/java**

**2.** [**www.google.co.in**](http://www.google.co.in)

**3.** ·[**https://www.geeksforgeeks.org/**](https://www.geeksforgeeks.org/)

**4. www.wikiepedia.org**