Student Management System

CSE-0318 Summer 2021

Mushiur Rahman Any
UG02-47-18-019
Department of Computer Science and Engineering
State University of Bangladesh (SUB)
Dhaka, Bangladesh
mushiurany@gmail.com

***Abstract*—Student Management system is software is which is helpful for students as well as the school authorities.In the current system all the activities are done manually.Student management system deals with the various activities related to the students**

***Index Terms*—PHP, MY SQL**

I. Introduction

student management software system to manage fee's of the students where admin can add/view/delete accountant and accountant can add/view/edit/delete student,check due fee and logout.

Users of the System; 1. Admin

2. Accountant

Functional Requirements;

Admin : 1. Can add/view/edit/delete accountant

2. Can logout

Accountant: 1. Can add/view/edit/delete students

2. Can check due fee

3. Can logout

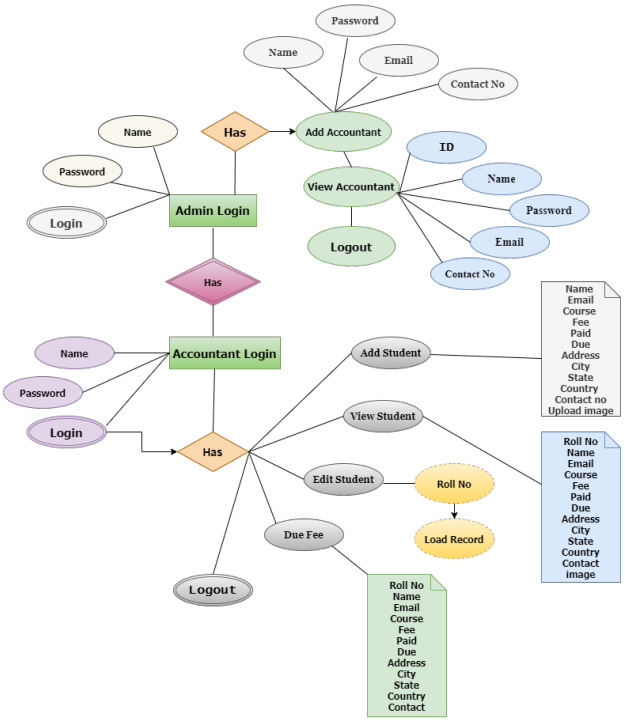
II. Literature Review

Student Management System Essay - 6221 Words Student Record System Research Paper Chapter 2 Review Literature The chapter states the different literature Building an Automated Student Record System · PDF file Building an Automated Student Record System Building a Student Record System, and review and report on education activities in foreign countries.Student management system literature review Building an Automated Student Record System Building a Student Record System, their statistical systems; and review and report on education activities in Essays: LITERATURE REVIEW ON STUDENTS MANAGEMENT Essays about: "literature review on students management system" Found 1 essay containing the words literature review on students A Literature Review - Public Health Foundation · PDF file A Literature Review development of a state public health performance management survey. The literature review Management Literature Search and Review.CHAPTER II. REVIEW OF RELATED LITERATURE · PDF file CHAPTER II. REVIEW OF RELATED LITERATURE personal managerial system, Literature Review 16 Personal Management Style CHAPTER 2 REVIEW OF RELATED LITERATURE AND STUDIES CHAPTER 2 REVIEW OF RELATED LITERATURE AND The University Management System in particular have created their own bespoke student record Chapter 2- Realated literature and Studies such as transaction processing system, management from manual to computerized system including student Literature free st math homework activation code compare and contrast essay high school vs college Review Chapter 2 Review of Related Literature and Studies — Chapter 2 Review of Related Literature and and student scheduling management system number is required to search for their record.Sample Literature Review Free Essays - Study Mode papers on Sample Literature Review Literature Review" Essays and Research Papers REVIEW The system in study is a management

III. ENTITY RELATIONSHIP DIAGRAM

The entity-relationship diagram of Student Management System shows all the visual instrument of database tables and the relations between Fees, Profiles, Student, Exams etc. It used structure data and to define the relationships between structured data groups of Student Management System functionalities.The entity-relationship diagram of Student Management System shows all the visual instrument of database tables and the relations between Fees, Profiles, Student, Exams etc. It used structure data and to define the relationships between structured data groups of Student Management System functionalities. The main entities of the Student Management System are Student, Fees, Logins, Profiles, Course and Exams. Student Management System entities and their attributes Er diagram represent relationship between two database tables. Description above figure shows the e r diagram of the proposed system the er model defines the conceptual view of a database. You can edit this template and create your own diagram.

The entity relationship diagram of university management system shows all the visual instrument of database tables and the relations between students faculties registrations etc. The entity relationship diagram of college management system shows all the visual instrument of database tables and the relations between books issues library branch etc. Er diagrams help you to define terms related to entity relationship modeling.



ER DIAGRAM STUDENT MANAGEMENT SYSTEM

IV. REQUIREMENTS

1. Use any IDE to develop the project. Intellij.
2. My SQL for the database.

Front End and Back End

1. Front End: Java Swing
2. Back End: My SQL

V. PROJECT CODE

```sql
-- phpMyAdmin SQL Dump
-- version 3.4.5
-- http://www.phpmyadmin.net
--
-- Host: localhost
-- Generation Time: Dec 17, 2016 at 08:39 AM
-- Server version: 5.5.16
-- PHP Version: 5.4.0beta2-devfeereport_accountantfeereport_student
USE feereport;
DROP TABLE IF EXISTS feereport_accountant;
SET time_zone = "+00:00";


/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8 */;

--
-- Database: `feereport`
--

-- --------------------------------------------------------
```

Database

```sql
--
-- Table structure for table `feereport_accountant`
--

CREATE TABLE IF NOT EXISTS `feereport_accountant` (
  `id` int(10) NOT NULL AUTO_INCREMENT,
  `name` varchar(100) NOT NULL,
  `password` varchar(100) NOT NULL,
  `email` varchar(100) NOT NULL,
  `contactno` varchar(20) NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1 AUTO_INCREMENT=7 ;

--
-- Dumping data for table `feereport_accountant`
--

INSERT INTO `feereport_accountant` (`id`, `name`, `password`, `email`, `contactno`) VALUES
(1, 's', 's', 's@gmail.com', '9199291212');

/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
```

Database

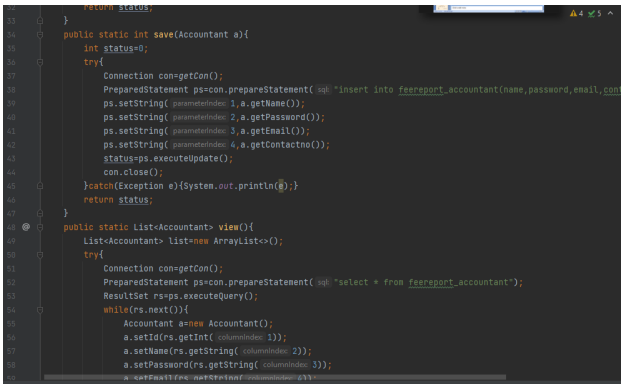
```sql
-- phpMyAdmin SQL Dump
-- version 3.4.5
-- http://www.phpmyadmin.net
--
-- Host: localhost
-- Generation Time: Dec 17, 2016 at 08:39 AM
-- Server version: 5.5.16
-- PHP Version: 5.4.0beta2-dev

USE feereport;
DROP TABLE IF EXISTS feereport_student;
SET SQL_MODE="NO_AUTO_VALUE_ON_ZERO,NO_ENGINE_SUBSTITUTION";
SET time_zone = "+00:00";


/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8 */;

--
-- Database: `test`
--

-- --------------------------------------------------------
```

Database

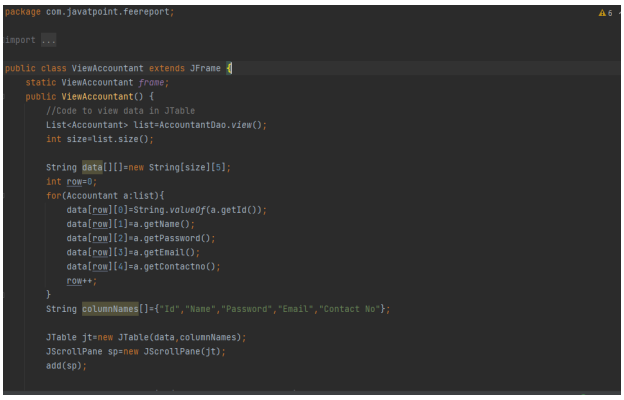
```sql
CREATE TABLE IF NOT EXISTS `feereport_student` (
  `rollno` int(10) NOT NULL AUTO_INCREMENT,
  `name` varchar(200) NOT NULL,
  `email` varchar(200) NOT NULL,
  `course` varchar(100) NOT NULL,
  `fee` int(10) NOT NULL,
  `paid` int(10) NOT NULL,
  `due` int(10) NOT NULL,
  `address` varchar(300) NOT NULL,
  `city` varchar(200) NOT NULL,
  `state` varchar(200) NOT NULL,
  `country` varchar(100) NOT NULL,
  `contactno` varchar(20) NOT NULL,
  `registereddate` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
  `student_image` BLOB,

  PRIMARY KEY (`rollno`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;

--
-- Dumping data for table `feereport_student`
--
```

Database

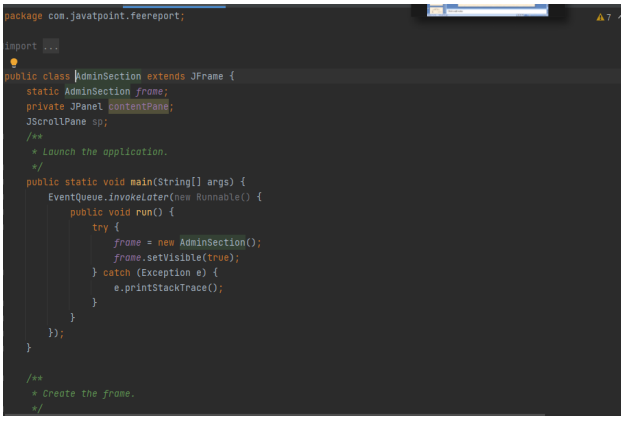
```java
package com.javatpoint.feereport;

import ...

public class AccountantDao {
    public static Connection getCon(){
        Connection con=null;
        try{
            Class.forName("com.mysql.jdbc.Driver");
            con=DriverManager.getConnection( url: "jdbc:mysql://localhost:3306/feereport", user: "root", password: "password
        }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( sql: "select * from feereport_accountant where name=? and passwo
            ps.setString( parameterIndex: 1,name);
            ps.setString( parameterIndex: 2,password);
            ResultSet rs=ps.executeQuery();
            status=rs.next();
            con.close();
        }catch(Exception e){System.out.println(e);}


        return status;
```



```
        return status;
    }
    public static int save(Accountant a){
        int status=0;
        try{
            Connection con=getCon();
            PreparedStatement ps=con.prepareStatement( sql: "insert into feereport.accountant(name,password,email,cont
            ps.setString( parameterIndex: 1,a.getName());
            ps.setString( parameterIndex: 2,a.getPassword());
            ps.setString( parameterIndex: 3,a.getEmail());
            ps.setString( parameterIndex: 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( sql: "select * from feereport.accountant");
            ResultSet rs=ps.executeQuery();
            while(rs.next()){
                Accountant a=new Accountant();
                a.setId(rs.getInt( columnIndex: 1));
                a.setName(rs.getString( columnIndex: 2));
                a.setPassword(rs.getString( columnIndex: 3));
                a.setEmail(rs.getString( columnIndex: 4));
```



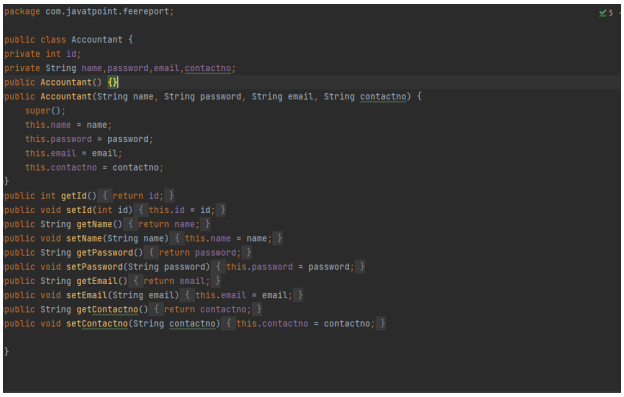
```
package com.javatpoint.feereport;

import ...

public class ViewAccountant extends JFrame {
    static ViewAccountant frame;
    public ViewAccountant() {
        //Code to view data in JTable
        List<Accountant> list=AccountantDao.view();
        int size=list.size();

        String data[][]=new String[size][5];
        int row=0;
        for(Accountant a:list){
            data[row][0]=String.valueOf(a.getId());
            data[row][1]=a.getName();
            data[row][2]=a.getPassword();
            data[row][3]=a.getEmail();
            data[row][4]=a.getContactno();
            row++;
        }
        String columnNames[]={"Id","Name","Password","Email","Contact No"};

        JTable jt=new JTable(data,columnNames);
        JScrollPane sp=new JScrollPane(jt);
        add(sp);
```



```
package com.javatpoint.feereport;

import ...

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.
     */
```

```
        }
        String columnNames[]={"Id","Name","Password","Email","Contact No"};

        JTable jt=new JTable(data,columnNames);
        JScrollPane sp=new JScrollPane(jt);
        add(sp);

        setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
        setBounds( x: 100,  y: 100,  width: 800,  height: 400);
    }

    public static void main(String[] args) {
        EventQueue.invokeLater(new Runnable() {
            public void run() {
                try {
                    frame = new ViewAccountant();
                    frame.setVisible(true);
                } catch (Exception e) {
                    e.printStackTrace();
                }
            }
        });
    }
}
```

```
     * Create the frame.
     */
    public AdminSection() {
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setBounds( x: 100,  y: 100,  width: 450,  height: 300);
        contentPane = new JPanel();
        contentPane.setBorder(new EmptyBorder( top: 5,  left: 5,  bottom: 5,  right: 5));

        setContentPane(contentPane);

        JLabel lblAdminSection = new JLabel( text: "Admin Section");
        lblAdminSection.setForeground(Color.DARK_GRAY);
        lblAdminSection.setFont(new Font( name: "Tahoma", Font.PLAIN,  size: 20));

        JButton btnAddAccountant = new JButton( text: "Add Accountant");
        btnAddAccountant.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                AddAccountant.main(new String[]{});
                frame.dispose();
            }
        });

        JButton btnViewAccountant = new JButton( text: "View Accountant");
        btnViewAccountant.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) { ViewAccountant.main(new String[]{}); }
        });
```

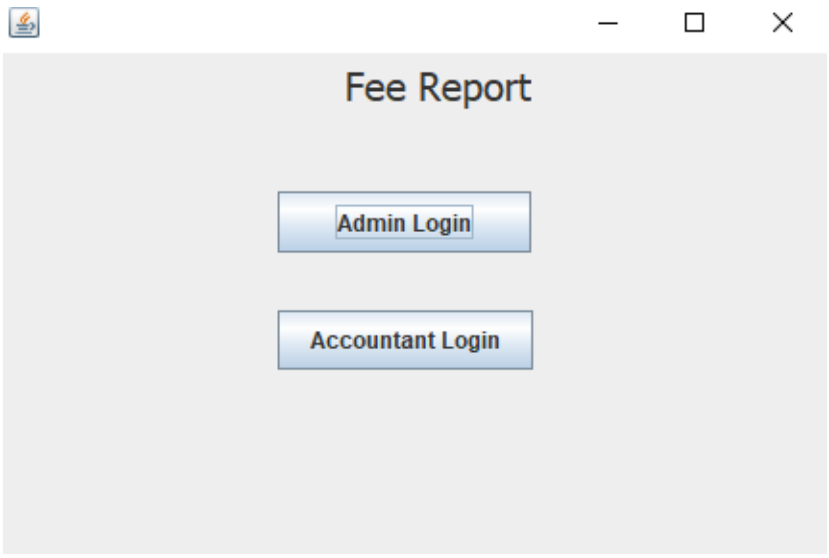


```
package com.javatpoint.feereport;

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; }

}
```

```
        });
        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(349)
                            .addGroup(gl_contentPane.createParallelGroup(Alignment.LEADING)
                                .addComponent(btnViewAccountant, GroupLayout.PREFERRED_SIZE,  pref: 130, GroupLayout.PREFERRED
                                .addComponent(btnAddAccountant, GroupLayout.PREFERRED_SIZE,  pref: 130, GroupLayout.PREFERRED_S
                                .addComponent(btnLogout, GroupLayout.PREFERRED_SIZE,  pref: 130, GroupLayout.PREFERRED_SIZE))))
                    .addContainerGap( pref: 130, Short.MAX_VALUE))
        );
        gl_contentPane.setVerticalGroup(
            gl_contentPane.createParallelGroup(Alignment.LEADING)
                .addGroup(gl_contentPane.createSequentialGroup()
                    .addComponent(lblAdminSection)
                    .addGap(29)
                    .addComponent(btnAddAccountant, GroupLayout.PREFERRED_SIZE,  pref: 34, GroupLayout.PREFERRED_SIZE)
                    .addGap(18)
                    .addComponent(btnViewAccountant, GroupLayout.PREFERRED_SIZE,  pref: 34, GroupLayout.PREFERRED_SIZE)
                    .addGap(18)
                    .addComponent(btnLogout, GroupLayout.PREFERRED_SIZE,  pref: 34, GroupLayout.PREFERRED_SIZE)
```

```
package com.javatpoint.feereport;

import ...

public class FeeReport extends JFrame {
    static FeeReport frame;
    private JPanel contentPane;

    /**
     * Launch the application.
     */
    public static void main(String[] args) {
        EventQueue.invokeLater(new Runnable() {
            public void run() {
                try {
                    frame = new FeeReport();
                    frame.setVisible(true);
                } catch (Exception e) {
                    e.printStackTrace();
                }
            }
        });
    }

    /**
     * Create the frame.
     */
```

VI. PROJECT OUTPUT



Home Page

VII. Conclusion and Future Work

It is always prudent to opt for a student management system that is designed using modern system architecture to cope with changing requirements. This system should encompass very solid information coding and distinctly outlined business applications. The overview of system elaborates the ease of information delivery at the tip of your fingers with precise data and increases the retention rate of student and teaches them how to manage their time efficiently.
To prepare this project i will use Use any IDE to develop the project. Intellij,My SQL for the database,Front End: Java Swing Back End: My SQL etc.

Acknowledgment

I would like to thank my honourable**Khan Md. Hasib Sir** for his time, generosity and critical insights into this project.

References

[1] Yue, Z. G., Jin, Y. W. (2010, August). The development and design of the student management system based on the network environment. In 2010 International conference on multimedia communications (pp. 5-8). IEEE.

[2] Bharamagoudar, S. R., Geeta, R. B., Totad, S. G. (2013). Web based student information management system. International Journal of Advanced Research in Computer and Communication Engineering, 2(6), 2342-2348.