**SCHOOL MANAGMENT SYSTEM**

**A Project Report Submitted**

**in Partial Fulfilment of the Requirements**

**for the Degree of**

**MASTER OF COMPUTER APPLICATION**

by

**KRISHNA KUMAR**

**(Enrollment no. 1802914005)**

**Under the Supervision of**

**Dr. SANGEETA ARORA**

**ASSOCIATE PROFESSOR**

**KIET Group of Institutions, Ghaziabad**

**to the**

**Department of Computer Applications**

**Dr. A.P.J. ABDUL KALAM TECHNICAL UNIVERSITY**

**LUCKNOW**

**(Formerly Uttar Pradesh Technical University, Lucknow)**

**(August 2021)**

**DECLARATION**

I hereby declare that the work presented in this report entitled “SCHOOL MANAGMENT SYSTEM”, was carried out by me. I have not submitted the matter embodied in this report for the award of any other degree or diploma of any other University or Institute.

I have given due credit to the original authors/sources for all the words, ideas, diagrams, graphics, computer programs, experiments, results, that are not my original contribution. I have used quotation marks to identify verbatim sentences and given credit to the original authors/sources.

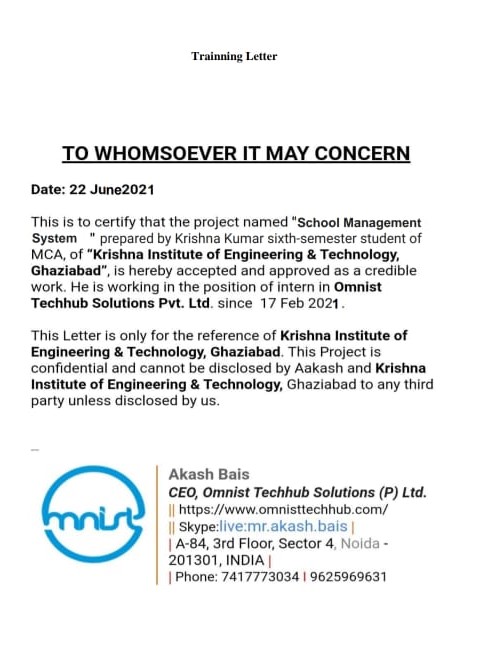
I affirm that no portion of my work is plagiarized, and the experiments and results reported in the report are not manipulated. In the event of a complaint of plagiarism and the manipulation of the experiments and results, I shall be fully responsible and answerable.

Name : Krishna Kumar

Roll. No. : 1802914005

Branch : Master of Computer Applications

**(Candidate Signature)**

****

**CERTIFICATE**

Certified that **Krishna Kumar** (**1802914005**) has carried out the project work presented in this report entitled “**School Managment System**” for the award of **Master of Computer Application** from Dr. A.P.J. Abdul Kalam Technical University, Lucknow under my supervision. The report embodies result of original work, and studies are carried out by the student himself and the contents of the report do not form the basis for the award of any other degree to the candidate or to anybody else from this or any other University.

**Dr. Sangeeta Arora External Examiner**

Associate Professor

Dept. of Computer Applications

KIET Group of Institutions, Ghaziabad

**Dr. Ajay Kumar Srivastava**

Professor & Head

Department of Computer Applications

KIET Group of Institutions, Ghaziabad

Date: 07/08/2021

# ACKNOWLEDGEMENT

I take this occasion to thank God, almighty for blessing us with his grace and taking our

Endeavor to a successful culmination. I extend my sincere and heartfelt thanks to our

Esteemed guide, **Dr.** **SANGEETA** **ARORA**, for providing me with the right guidance and

Advice at the crucial junctures and for showing me the right way. I extend my sincere thanks

to our respected **Head** **of** **the** **department** **Dr.** **AJAY** **KUMAR** **SHRIVASTAVA,** for

Allowing us to use the facilities available. I would like to thank the other faculty members

also, at this occasion. Last but not the least, I would like to thank my friends and family for

the support and encouragement they have given me during the course of our work.

**KRISHNA** **KUMAR**

**(1802914005)**

# SCHOOL MANAGEMENT SYSTEM

# KRISHNA KUMAR

# ABSTRACT

This project work automates school management system. In the system two applications are developed, Windows based (thick client) and Web based (thin client).

The windows application takes most of the activities such as offline student registering, transcript and report card generation and producing the timetable.

The web application facilitates attendance recording by the homeroom teachers, to view status of students by their parents and to view reports by kebele and kifle-ketema education bureau officials. Our solution of the timetable is very simple. In the high school considered for the project there are ten subjects for both grade nine and grade ten. Loads are assigned to each subject teacher and a code is given for each teacher-subject combination.

A simple search technique has been used during allocation of each teacher-subject code to a time slot. A database has been used to enforce constraints and to store data.

The prototype has been tested with data from radient Secondary School. It has been observed that the system successfully registers students, facilitates attendance recording by the home room teachers and generates various reports such as report card, transcript and a feasible timetable satisfying the constraints (requirements).

# TABLE OF CONTENTS

PAGE NO. .

Declaration ii

Training Letter iii

Certificate iv

Acknowledgement v

Abstract vi

Table of contents vii

List of table ix

List of figure x

PAGENO.

**CHAPTER 1:**  **INTRODUCTION 1-3**

1. Introduction 1

1.1 Overview 1

1.2 Study of Systems 2

**CHAPTER 2: LITERATURE REVIEW 4-5**

2.1 Observed product 5

2.2 Manual timetabling 5

2.3 Drawbacks of the reviewed system 5

**CHAPTER 3: SYSTEM ANALYSIS 6-7**

3.1 Functional requirements 7

3.2 Non-functional rrequirements 7

3.3 Analysis model 7

3.3.1 Use case diagram 8

**CHAPTER 4:** **SYSTEMDESIGN 8-21**

4.1 Records 9

4.2. Use case description 9

4.3 Alternative flow of events 10

4.4. Flow Chart 11

4.1.1 ER- Diagram 13

4.1.2 Complete diagram 14

4.1.3 Data flow diagram 15

4.1.4 Data store 15

1. 0-level DFD 15
2. 1-level DFD 16

4.1.1.1 Decomposition 17

4.1.1.2 Module Decomposition 18

**CHAPTER 5: CODING 22-53**

5.1 Login 23

5.2 Main Menu 30

5.3 Student Record 36

5.4 Teacher Record 43

5.5 Employee Record 50

**CHAPTER 6: RESULT 54-62**

6.1 Login page 54

6.2 Main menu 56

**CHAPTER 7. Conclusion** **and** **Recommendation** **63-64**

7.1 Conclusion 63

7.2 Recommendation 63

**REFERENCES** 64

# LIST OF TABLE

**DATABASE TABLE**

Table no: 6.1 58

Table no: 6.2 60

Table no: 6.3 62

# LIST OF FIGURE

3.1.1 Use Case Diagram 8

4.1. Flow Chart 11

**ER** **DIAGRAM**

4.2. Login 12

4.3. User details 12

4.4. Complete diagram 14

**DATA** **FLOW DIAGRAM**

4.5. 0-level DFD 15

4.6. 1-level DFD 16

4.7. Decompostion Description 17

4.8. Data Decompostion 20

# CHAPTER 1

# INTRODUCTION

Education system forms the backbone of every nation. And hence it is important to provide a strong educational foundation to the young generation to ensure the development of open-minded global citizens securing the future for everyone. Advanced technology available today can play a crucial role in streamlining education-related processes to promote solidarity among students, teachers, parents and the school staff

Education is central to development. It is one of the most powerful instruments for reducing poverty and inequality and lays a foundation for sustained economic growth. With this aim currently our government has given special emphasis to the educational sector and school improvement activities such as continuous professional development for teachers, training and upgrading teachers and capacitating schools with manpower and materials are among the major actions which have been taken in both primary and secondary schools. In order to facilitate and simplify these actions one of the major tool is to have automated school management system.

School Management System(SMS) consists of tasks such as registering students, attendance record keeping to control absentees, producing report cards, producing official transcript, preparing timetable and producing different reports for teachers, parents, officials from kebele or kefle ketema education bureaus and other stakeholders.

Automation is the utilization of technology to replace human with a machine that can perform more quickly and more continuously [2]. By automating SMS documents that took up many large storage rooms can be stored on few disks. Transcript images can be annotated. It reduces the time to retrieve old transcripts from hours to seconds. However, the school system in the government schools of Addis Ababa is not automated and the record officers generate transcripts and reports manually and the school administrators use their experienced knowledge of miss and hit approaches to prepare timetables.

## 1.1 Overview:

Secondary education follows eight years of primary education and is for children aged 14 and above. At the beginning of each academic year which starts in September (Ethiopian New Year), the students get registered and assigned rooms. Each class (section) of students is assigned to a fixed room. Home room teachers are assigned to each class of students. There are two semesters per year. The first semester final examination is usually administered during January, the second semester final examination is administered during the end of June and consequently the results of each class of students is collected, organized, ranked by the corresponding home room teacher and reported to each student. The homeroom teacher also records attendance of each student on each school day which is later organized by the attendance officer. A student who has been absent for more than twenty days is not allowed to take a semester final examination and will be forced to withdraw. Transcripts are generated by the record officer. A student may request transcript when he/she wants to transfer to other school or when he/she has completed/graduated from the school and needs to join higher education or for some other purpose. Officials from kebele and kifle-ketema education bureaus want to get statistical reports like number of registered students at the beginning of every year, number of drop outs, and number of passes/failures for each subject at the end of each semester as well as number of passes/failures at a grade level to help them participate in decision making.

## 1.2 STUDY OF THE SYSTEM

## 1. MODULES

The system after careful analysis has been identified to be presented with the following modules and roles.

The modules involved are:

* + Administrator
  + User

## The Time table Problem

School timetabling is a major administrative activity in any school. A number of subjectstaught by the corresponding teachers are allocated into a number of available classrooms and anumber of timeslots, subject to constraints.

The tasks that are considered in constructing thetimetable are:

* Assigning periods to classes. There is a need to spread out lessons across the teaching cycle
* As much as possible, e.g. to avoid having 3 lessons on the same day.
* Some classes need 'double periods' (preferably 2 consecutive periods).This happens
* currently for Mathematics and English since each of the subjects have 6 lessons per week (for five days) and therefore on one of the days these subjects should have two lessons for each class of students.

Some combinations of assignments lead to acceptable timetables, others do not. Suchrestrictions follow from conditions imposed by classes (rooms), students or teachers. Wedistinguish two types of conditions: conditions that must be met (requirements) and conditionsthat should be fulfilled as well as possible (desires) [5]. A class of students has a fixed roomthroughout the academic year and therefore we use class and room alternatively. Here a roomis mentioned only when there are students in it. The time tabling problem is said to be feasibleif and only if it satisfies the following constraints (requirements).

## Details Adds

The shopping cart project contains different kind of products. The products can be classified

into different categories by name. Admin can add new products into the existing system with

all its details including an image.

## Delete Details

Administrator can delete the products based on the stock of that particular product.

## Search Details

Admin will have a list view of all the existing products. He can also search for a particular product by name.

## MANAGE USER

## View Users

The admin will have a list view of all the users registered in the system.

Admin can view all the details of each user in the list except password.

## Add Users

Admin has privileges to add a user directly by providing the details.

## Delete &Block Users

Administrator has a right to delete or block a user. The default status of a new userregistered is set as blocked. The admin must accept the new user by unblocking him.

## Time Slot Assignment

The school timetabling is a weekly scheduling for all the classes of a school, avoidingteachers meeting two classes at the same time, and vice versa. This means that an event maybe placed in the timetable only in such a way that it does not violate constraints.

shows the concepts of timetable construction at schools.Lessons in a subject are taught by teacher to a corresponding class of students and the timetabling problem is a problem ofallocating resources, i.e. assigning to teachers and class of students, time slots and lessons.

A time slot is a period and a lesson is an event associating a teacher, a subject and a classof students with in a time slot. Lesson Teacher Room Time Slot Subject Conceptof timetabling construction at schools Timetabling is based on a rectangular time grid thatdivides the planning period into disjoint time intervals of equal duration (42 minutes in thiscase) which are called time slots or simply periods.shows a typical time grid withone day (Monday) for all possible classrooms (rib) and seven time slots a dayThe lectures are activities and teachers and classes are resources. These resources are not available at certain time periods. A lecture can be given in period t only if the correspondingclass and teacher are available in t. All lectures have equal processing length, i.e. thelength of the period and have to be scheduled without preemption. To automate the school activities some literature reviews have been done.

The lectures are activities and teachers and classes are resources. These resources are not available at certain time periods. A lecture can be given in period t only if the corresponding class and teacher are available in t [1]. All lectures have equal processing length, i.e. the length of the period and have to be scheduled without preemption. To automate the school activities some literature reviews have been done.

The lectures are activities and teachers and classes are resources. These resources are not available at certain time periods. A lecture can be given in period t only if the corresponding class and teacher are available in t [1]. All lectures have equal processing length, i.e. thelength of the period and have to be scheduled without preemption. To automate the school activities some literature reviews have been done

# CHAPTER 2

# Literature Review

Automated SMS plays a great role in simplifying the job of employees at the school andsatisfying the need of customers and stakeholders of the school.Even though no documentation is found in Ethiopia to be reviewed, products have been observed at some schools to help understand the problem of managing schools and handling school data. This chapter reviews these products

## 2.1 Observed Products.

In the year 2003 City Government of Addis Ababa Education Bureau (CGAAEB)was very much interested to have automated school management system to getuniform and quick access to the students’ data for administrative purpose onpromoting the students’ achievement and related issues. The bureau has selectedWundrad Preparatory School for pilot test. At the time the school principals togetherwith officials from CGAAEB signed a contractual agreement with some softwaredeveloper company. The developers installed their first version of the product whichcan register a student offline and generate official transcript with some level ofdifficulty. As the system is not fully automated, it does not support management of

attendance, does not support generating report cards and other important functionssuch as generating school timetable and a web based report for parents. Due to thelack of follow up by the government officials at CGAAEB, the company was unableto complete the project. The school currently is unable to use the partially developedsystem because of lack of trained person and lack of hardware and softwaremaintenance.

## 2.2 Manual Timetabling

Manual timetables are prepared by dedicated teachers. In manual timetabling, it is commonto proceed in an iterative fashion where each iteration selects and schedules a lesson [3].Scheduling a lesson requires to choose a classroom (fixed for each section of students) and atime slot such that the commitment to the choice will not violate any constraint. In schooltimetabling, we are required to schedule a given set of meetings such that the resultingtimetables are feasible and acceptable to all people involved. Humans are able to prepare thetimetable using some hit/miss approach. So it is possible to automate the timetable based on asimulation of the human way of solving the problem. Such techniques, that we call directheuristics, were based on a successive augmentation. That is, a partial timetable is extended,lecture by lecture, until all lectures have been scheduled. The underlying idea of allapproaches is to schedule the most constrained lecture first. Usually some responsibleteachers are assigned to schedule subjects and teachers. The number of teachers available pereach subject is predefined and the load that each teacher has is calculated. With these data thetimetable constructor assigns each teacher-subject association to the appropriate classes withthe available time slots. The manual solution of the timetabling problem usually requiresmany person-days of work. In addition, the solution obtained may be unsatisfactory. The lessons should be fairly distributed satisfying the identified constraints.

## 2.3 Drawbacks of the Reviewed Systems

The reviews described have the following problems:

• Generate official transcript with some level of difficulty,

• Do not sufficiently produce the required reports to allow parents to view status of their children and reports for officials of kebele and kifle-ketema to help them participate in decision making,• Do not generate timetable for the schools

• Do not facilitate attendance record keeping by the homeroom teachers

This project work tries to fill the gap by automating the various activities at schools. It tries

to satisfy customers need and simplify the works of administrators, record officer and

teachers. With an automated school management system parents can easily interact with

the school community to follow up their children’s achievement and play their role in the

school development processes.

# Chapter 3

# System Analysis

In this chapter the functional and non-functional requirements of the system are described and modeled using UML models.

## 3.1 Functional Requirements

The functional requirements of the system are:

• register a student,

• record attendance of students,

• generate various reports,

• generate timetable

## 3.2 Non Functional Requirements

Security requirements are important factors in this system as classified data will be stored in thedatabase. User validation will be done during login to insure that the user is valid and that the useronly has access to his or her permission data. General users will only have access through the user interface

The system will have consistent interface formats and button sets for all forms in the application, will have a form based interface for all data entry and viewing formats, and will generate reports that are formatted in a table and that should look like the existing manual report formats for user friendliness.

The system will be easily maintained by the developer or other authorized trained person and it shall respond as fast as possible in generating report and producing the timetable.

## 3.3 Analysis Model

To produce a model of the system which is correct, complete and consistent we need toconstruct the analysis model which focuses on structuring and formalizing therequirements of the system. Analysis model contains three models: functional, object and

dynamic models. The functional model can be described by use case diagrams. Classdiagrams describe the object model. Dynamic model can also be described in terms ofsequence, state chart and activity diagrams. For the purpose of this project we have

described the analysis model in terms of the functional model and dynamic models usinguse case and sequence diagrams.

Register Student

Teacher Update Record

Delete Record

**Figure:3.1 Use** **Case** **Diagram**

# CHAPTER 4

# System Design

## 4.1 Records

## Name: Record Officer

## Description: A Record Officer is a person who registers a student, input, update student data and produce transcript and report card.

## Name: Homeroom Teacher

## Description: A Homeroom Teacher is a teacher assigned by the school director to each class of students to follow the students closely. Hershel has the responsibility of recording attendance

of students and submitting.

**Name:** Parent

**Description:** A Parent is a person who is registered as parent of the student and responsible to follow the student in close contact with the school. Hershel can view the status of the studentsuch as attendance and result/performance of the student online.

**Name:** Admin

**Description**: Admin is a person who is responsible to produce the timetable for each teacher and classroom by providing the necessary parameters.

**4.2 Use** **Case** **Description**

Register Student

**Record Officerscription:** To register someone as a student of the school

**Precondition:** **A** student has to be eligible (has to be from the pre-specified junior schools that the school will accept)

**Flow** **of** **Event:**

* Student wants to be registered as a student of the school
* Record officer verifies that the student is eligible
* Registration form will be given to the student
* The student completes the registration form that contains student’s full name, address, parent name, emergency person names and addresses and other detail information.
* Record Officer of the school checks whether the contents of the registration form is properly completed
* Record Officer fills and submits the form to the system
* System registers
* Use case ends.

**4.3 Alternative** **Flow** **of** **Events**

**Alternative** **flow** **A:** User is not a home room teacher of the class A3. User can’t record attendance for the required class of students A4.

**Name:**

GenerateReportCard

Record Officer

**Description:** To produce a report card for students per semester

**Precondition:** A student must have complete grade marks in all subjects of the semester

**Flow** **of** **Events:**

1. The record officer logs in and selects the class/section to which the student belongs

2. The record officer searches the student from the class/section based on the search criteria defined

3. The system processes the report card

4. System displays and print the result

5. Use case ends

**Alternative** **Flow** **of** **Events**

**Alternative** **flow** **A:**

The user logged in is not the record officer

A1. User can not generate report card

A2. Use case ends

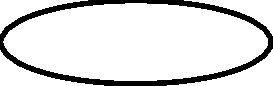
**Alternative** **flow** **B:**

The student is incomplete at least in one subject

B1. The system can not generate the report card.

B2. Use case ends

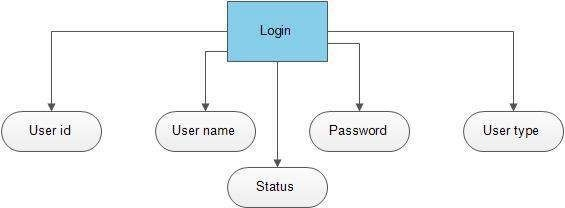
**4.4 Flow chart** **–**



|  |
| --- |
|  |
| **L** **OGI** **N**  **Add** **ViewI** **Category**  **AddI** **Manage**  **Manage** **Make**  **Manage** **Payme** **nt** |
|  |

**Administrato** **User** **r**

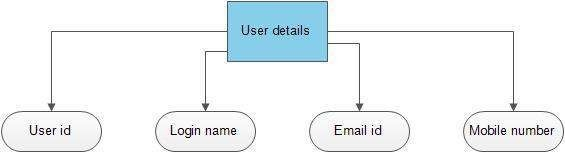
**Fig4.1.flowChart**

**4.1.1 ER** **Diagram–**

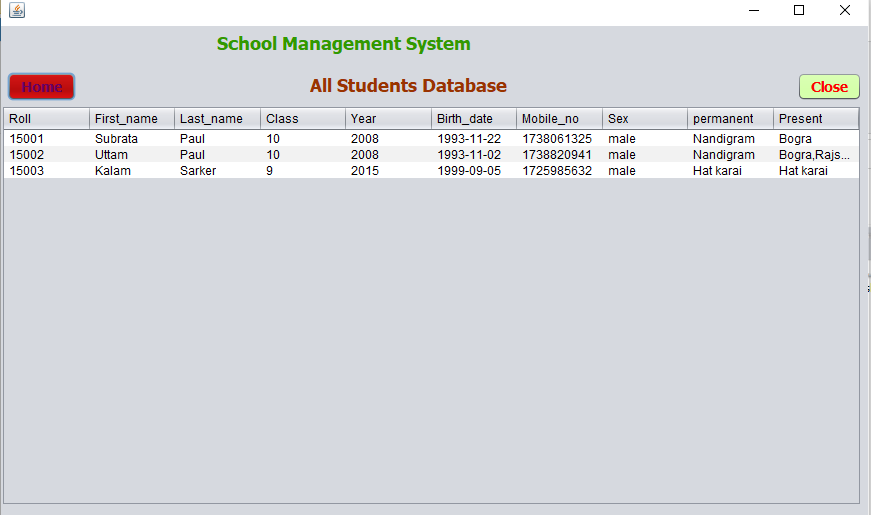
**LOGIN-**

**Fig** 4.**2:Login**

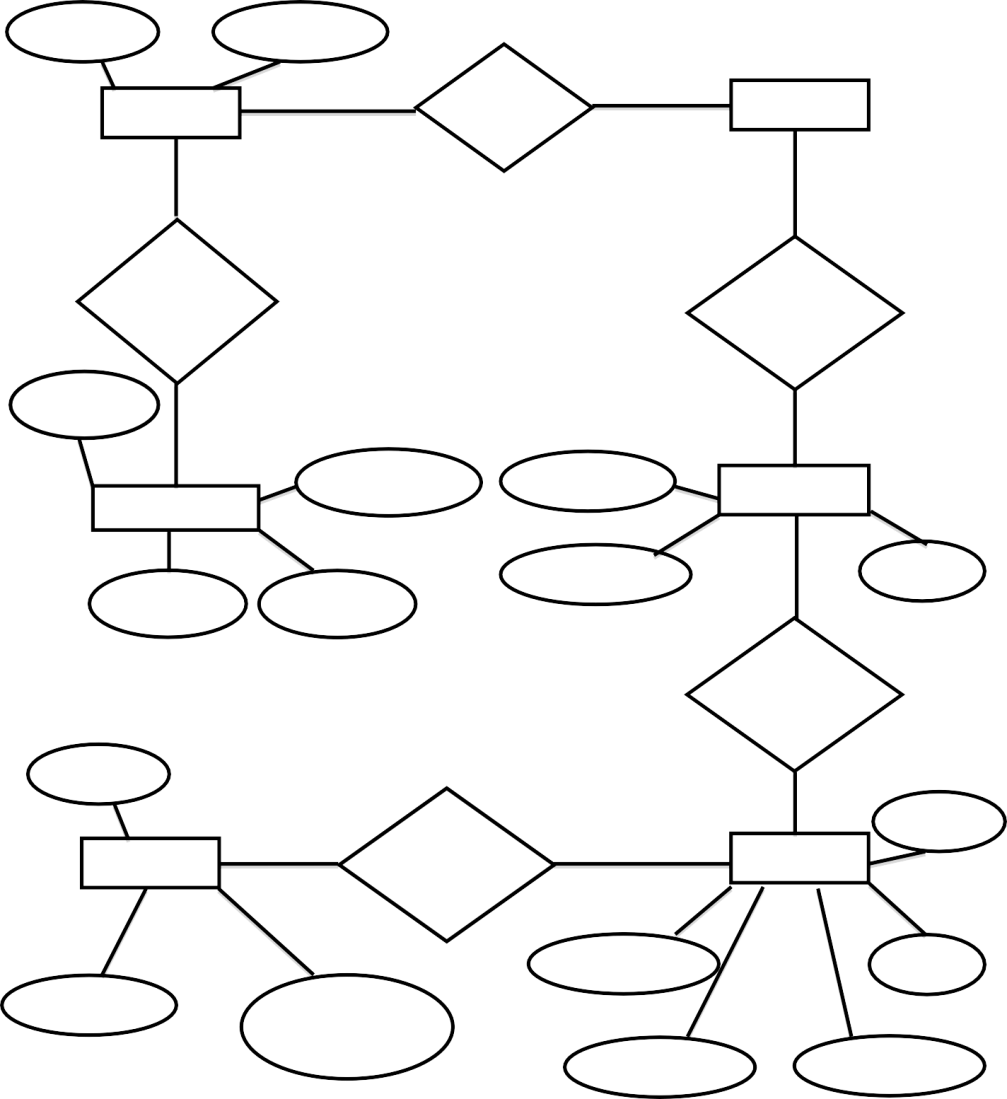
**USER** **DETAILS**



## Fig 4.3: User details

**STUDENT** **DETAILS:**

**4.1.2 COMPLETE** **DIAGRAM**



Em Password ail

**Lo** Access **Details** **gin**

New user Has Category

Address

Register

Em ail

Contactno.

Name

Pname

Description

Category

Feature

Has Category

Feels

**Deta** Payment **ils**

Pname

**Structu** **re**

Amount Student

Name

Pimage price

pid fees

## Fig 4.4: Complete diagram

**4.1.3 DATA** **FLOW** **DIAGRAM**

Sequence diagrams show the interaction between participating objects in a given use case. They are helpful to identify the missing objects that are not identified in the analysis object model. To see the interaction between objects, the following describe the sequence diagram of each identified use cases.

**PROCESS**

A process shows a transformation or manipulation of data flow within the system. The symbol used is an oval shape.

**DATAFLOW**

The data flow shows the flow of information from a source to its destination. Data flow is

represented by a line, with arrowheads showing the direction of flow. Information always

flows to or from a process and may be written, verbal or electronic. Each data flow may be

referenced by the processes or data stores at its head and tail, or by a description of its

content.

**4.1.4 DATASTORE**

A data store is a holding place for information within the system: It is represented by an open ended narrow rectangle. Data stores may be long-term files such as sales ledgers, or may be short-term accumulations: for example batches of documents that are waiting to be processed.

Each data store should be given a reference followed by an arbitrary number.

## 0-level DFD

Online Add

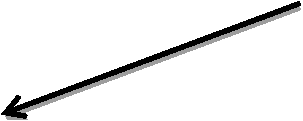
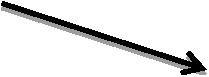
User login website

Admin

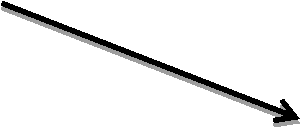
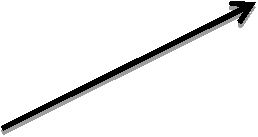
confirmation conformation

## Fig 4.5: 0 level DFD

# 1-level DFD



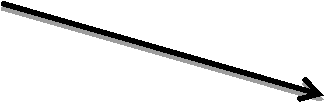
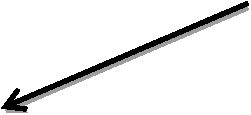
Login Admin



Add newitem

Login

Signup Or



Add Details

Storeand retrieve

storeitem

|  |  |
| --- | --- |
|  | User |
|  |
|  |

search details

Give

confirmation

Search details

Retrieve

item

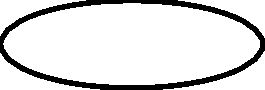
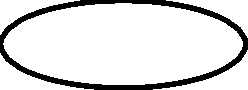
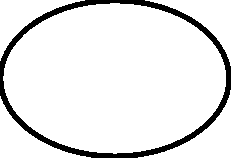
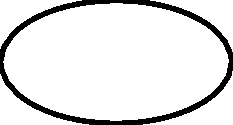
Sequen storeitem ce

Fees

**Fig4. 6: 1 Level DFD**

**4.1.1.1 Decomposition** **description**

**v**



Online Submittig

login

Home Fees Contact

Register

Forgot

password

Feedback

Query

Creditcard Sum mer

Debitcard y

Add

Edit

details

Edit

passwod

Student New

Category update

Type of OS

Android IO S

Windows

**Fig 4.7: Decompostion description**

**4.1.1.2** **Module** **decomposition**

**1.** **Login** **Module** **Description**

We have a module name Login which consist of Username and password if the access is

granted if the user registered for that then the access is allowed.

**2.** **Home** **Module** **Description**

There is one module where the product are shown by there category and the type of the

product is shown by them and the diffence is done through the types of operating system such

as android, windows,os and the details of the product with the help of the amount and the and

see the product images are available as per the original product that is available on that no

fake images are available.

**3.** **Fees** **Module** **Description**

the payment done the category of the payment are Debit card,credit card and cash on delivery method There is another Module named payment.The Payment is done by the category of.

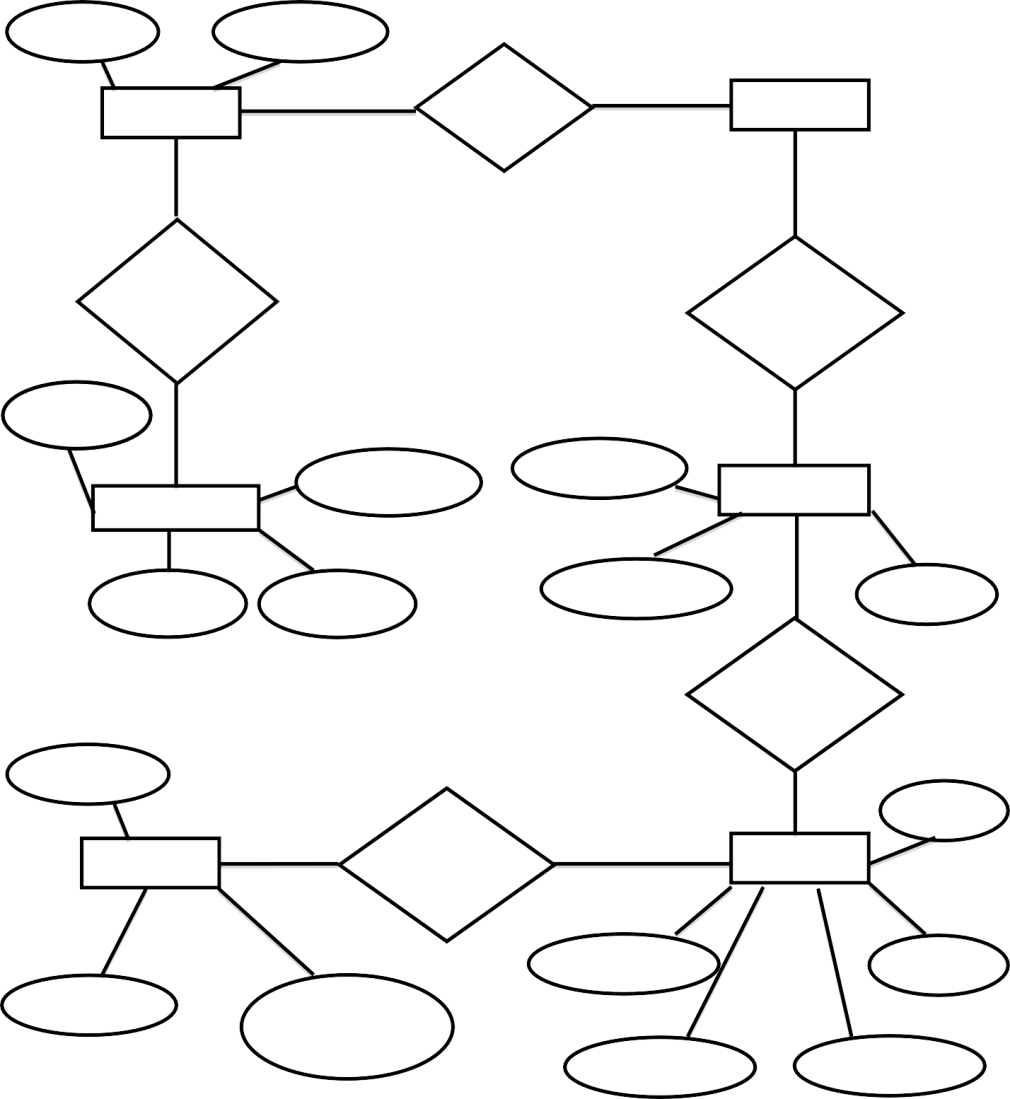
**4.** **Contact** **Module** **Description**

We have a module name Contact which consist of contact details of our site if there is any inconvenient caused and for for any query you submit your query online.

**Concurrent** **Process** **Decompostion**

|  |  |  |  |
| --- | --- | --- | --- |
| **Modules** | **Class** | **Variable** | **Function** |
| Register | Prepared statement Connection Resultset | S S 1 S 2 S 3 S 4 Ps P s1 | getParameter() setstring() next() equals() executequery () |
| Login | Session Statement Drivermana ger Exception Button | Name Passwo rd Conn Flag | getattribute() setAttribute() post() con.close() getstring() setstring() createstatement() |
| Home | Docu ment String  Preparedsatem ent Request | Search Sql  Un ame Value | Equals() Alert() Executeupdate() |
| Fees | Connect ion Driver Class String | R s P s  Con n X Pid Un ame | Getconnection() Post() Forname() Length() Settimeout() Close() tostring () |

**DATA DECOMPOSITION**



Em Password ail

**Lo** Access **Student** **gin**

New user Has Category

Address

Register

Em ail

Contactno.

Name

Pname

Description

**Categor** **y**

Feature

Has Category

Summ

ery Pname

**Fees** **Strict** Fees **Structu** **red**

Amount Student

Name

Pimage Fees

pid Details

**Fig 4.8: Data decompostion**

**LOGIN:-**This module is for for giving the access for the authenticated user. Which have already registered for that.

**REGISTER:-**This is for the registration of new user who did not did there registration.

**HOME:** This module is for showing the products that are available and can check the products according to the category desire.

**FEES:-** This module is for selecting the payment mode that is desired for that there are various payment mode available according to that user desire or wish can do that payment.

**CONTACT:-** This module for that where the user can contact with and solve and discuss their queries in which they are facing the problem.

**Dependency description**

**U\_login**

**Verify()**

**Communicational** **Cohesion**

**d:Register**

**Insertuser()**

**U-profile**

**Add()**

**Update()** **Delete** **()**

|  |
| --- |
| **D:search** |
| **Productname** |
| **searchDB()** |

**Sequential** **cohesion**

**Ask\_query**

**Sendquery()** **Viewquery()**

**D\_giveanswer**

**Sendanswer()** **Viewanswer()**

|  |
| --- |
| **U\_home** |
| **Buy** **Pr\_name** **New** **Price** **Product** **Category** |
| **Add\_newitem()** |

**Interface description**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Module | Login | Home | Payment | Contact |
| Login | X | Control Coupling | Stamp Coupling | Content Coupling |
| Home | Data Coupling | X | Content Coupling | Data coupling |
| Fees | Content coupling | Control coupling | X | Common Coupling |
| Contact | Common coupling | Stamp Coupling | Common Coupling | X |

# CHAPTER 5

# Coding

**5.1 Login**

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package GUI;

import java.awt.HeadlessException;

import java.awt.event.KeyEvent;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import javax.swing.JOptionPane;

/\*\*

\*

\* @author Krishna Kumar

\*/

public class Login extends javax.swing.JFrame {

Connection con=null;

ResultSet rs=null;

PreparedStatement pst=null;

/\*\*

\* Creates new form Login

\*/

public Login() {

initComponents();

setIcon();

}

/\*\*

\* This method is called from within the constructor to initialize the form.

\* WARNING: Do NOT modify this code. The content of this method is always

\* regenerated by the Form Editor.

\*/

@SuppressWarnings("unchecked")

// <editor-fold defaultstate="collapsed" desc="Generated Code">//GEN-BEGIN:initComponents

private void initComponents() {

java.awt.GridBagConstraints gridBagConstraints;

welcomeDialog = new javax.swing.JDialog();

confirmDialog = new javax.swing.JDialog();

jPanel1 = new javax.swing.JPanel();

userNameLabel = new javax.swing.JLabel();

passwordLabel = new javax.swing.JLabel();

userNameField = new javax.swing.JTextField();

passwordField = new javax.swing.JPasswordField();

loginButton = new javax.swing.JButton();

cancelButton = new javax.swing.JButton();

forgetPasswordLabel = new javax.swing.JLabel();

jLabel1 = new javax.swing.JLabel();

javax.swing.GroupLayout welcomeDialogLayout = new javax.swing.GroupLayout(welcomeDialog.getContentPane());

welcomeDialog.getContentPane().setLayout(welcomeDialogLayout);

welcomeDialogLayout.setHorizontalGroup(

welcomeDialogLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGap(0, 400, Short.MAX\_VALUE)

);

welcomeDialogLayout.setVerticalGroup(

welcomeDialogLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGap(0, 300, Short.MAX\_VALUE)

);

javax.swing.GroupLayout confirmDialogLayout = new javax.swing.GroupLayout(confirmDialog.getContentPane());

confirmDialog.getContentPane().setLayout(confirmDialogLayout);

confirmDialogLayout.setHorizontalGroup(

confirmDialogLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGap(0, 400, Short.MAX\_VALUE)

);

confirmDialogLayout.setVerticalGroup(

confirmDialogLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGap(0, 300, Short.MAX\_VALUE)

);

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT\_ON\_CLOSE);

setTitle("Login");

setResizable(false);

jPanel1.setBackground(new java.awt.Color(0, 0, 0));

jPanel1.setLayout(new java.awt.GridBagLayout());

userNameLabel.setBackground(new java.awt.Color(0, 0, 0));

userNameLabel.setFont(new java.awt.Font("Dialog", 1, 12)); // NOI18N

userNameLabel.setForeground(new java.awt.Color(255, 255, 255));

userNameLabel.setText("Username:");

gridBagConstraints = new java.awt.GridBagConstraints();

gridBagConstraints.gridx = 0;

gridBagConstraints.gridy = 0;

gridBagConstraints.ipadx = 38;

gridBagConstraints.ipady = 14;

gridBagConstraints.anchor = java.awt.GridBagConstraints.NORTHWEST;

gridBagConstraints.insets = new java.awt.Insets(81, 73, 0, 0);

jPanel1.add(userNameLabel, gridBagConstraints);

passwordLabel.setBackground(new java.awt.Color(0, 0, 0));

passwordLabel.setFont(new java.awt.Font("Dialog", 1, 12)); // NOI18N

passwordLabel.setForeground(new java.awt.Color(255, 255, 255));

passwordLabel.setText("Password:");

gridBagConstraints = new java.awt.GridBagConstraints();

gridBagConstraints.gridx = 0;

gridBagConstraints.gridy = 1;

gridBagConstraints.ipadx = 39;

gridBagConstraints.ipady = 14;

gridBagConstraints.anchor = java.awt.GridBagConstraints.NORTHWEST;

gridBagConstraints.insets = new java.awt.Insets(43, 73, 0, 0);

jPanel1.add(passwordLabel, gridBagConstraints);

userNameField.setBackground(new java.awt.Color(255, 255, 255));

userNameField.setFont(new java.awt.Font("Dialog", 1, 12)); // NOI18N

userNameField.setForeground(new java.awt.Color(0, 0, 0));

userNameField.setToolTipText("Enter Username Here");

gridBagConstraints = new java.awt.GridBagConstraints();

gridBagConstraints.gridx = 1;

gridBagConstraints.gridy = 0;

gridBagConstraints.gridwidth = 3;

gridBagConstraints.ipadx = 141;

gridBagConstraints.ipady = 10;

gridBagConstraints.anchor = java.awt.GridBagConstraints.NORTHWEST;

gridBagConstraints.insets = new java.awt.Insets(81, 12, 0, 0);

jPanel1.add(userNameField, gridBagConstraints);

passwordField.setBackground(new java.awt.Color(255, 255, 255));

passwordField.setFont(new java.awt.Font("Dialog", 1, 12)); // NOI18N

passwordField.setForeground(new java.awt.Color(0, 0, 0));

passwordField.setToolTipText("Enter Password Here");

passwordField.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

passwordFieldActionPerformed(evt);

}

});

passwordField.addKeyListener(new java.awt.event.KeyAdapter() {

public void keyPressed(java.awt.event.KeyEvent evt) {

passwordFieldKeyPressed(evt);

}

});

gridBagConstraints = new java.awt.GridBagConstraints();

gridBagConstraints.gridx = 1;

gridBagConstraints.gridy = 1;

gridBagConstraints.gridwidth = 3;

gridBagConstraints.ipadx = 141;

gridBagConstraints.ipady = 10;

gridBagConstraints.anchor = java.awt.GridBagConstraints.NORTHWEST;

gridBagConstraints.insets = new java.awt.Insets(43, 12, 0, 0);

jPanel1.add(passwordField, gridBagConstraints);

loginButton.setBackground(new java.awt.Color(255, 255, 255));

loginButton.setFont(new java.awt.Font("Dialog", 1, 12)); // NOI18N

loginButton.setForeground(new java.awt.Color(0, 0, 0));

loginButton.setText("Login");

loginButton.setToolTipText("");

loginButton.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

loginButtonActionPerformed(evt);

}

});

gridBagConstraints = new java.awt.GridBagConstraints();

gridBagConstraints.gridx = 0;

gridBagConstraints.gridy = 2;

gridBagConstraints.gridwidth = 2;

gridBagConstraints.ipadx = 8;

gridBagConstraints.anchor = java.awt.GridBagConstraints.NORTHWEST;

gridBagConstraints.insets = new java.awt.Insets(30, 165, 0, 0);

jPanel1.add(loginButton, gridBagConstraints);

cancelButton.setBackground(new java.awt.Color(255, 255, 255));

cancelButton.setFont(new java.awt.Font("Dialog", 1, 12)); // NOI18N

cancelButton.setForeground(new java.awt.Color(0, 0, 0));

cancelButton.setText("Cancel");

cancelButton.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

cancelButtonActionPerformed(evt);

}

});

gridBagConstraints = new java.awt.GridBagConstraints();

gridBagConstraints.gridx = 3;

gridBagConstraints.gridy = 2;

gridBagConstraints.gridwidth = 4;

gridBagConstraints.anchor = java.awt.GridBagConstraints.NORTHWEST;

gridBagConstraints.insets = new java.awt.Insets(30, 37, 0, 0);

jPanel1.add(cancelButton, gridBagConstraints);

forgetPasswordLabel.setBackground(new java.awt.Color(0, 0, 0));

forgetPasswordLabel.setFont(new java.awt.Font("Dialog", 1, 12)); // NOI18N

forgetPasswordLabel.setForeground(new java.awt.Color(255, 255, 255));

gridBagConstraints = new java.awt.GridBagConstraints();

gridBagConstraints.gridx = 0;

gridBagConstraints.gridy = 3;

gridBagConstraints.gridwidth = 8;

gridBagConstraints.ipadx = 229;

gridBagConstraints.ipady = 40;

gridBagConstraints.anchor = java.awt.GridBagConstraints.NORTHWEST;

gridBagConstraints.insets = new java.awt.Insets(18, 140, 0, 0);

jPanel1.add(forgetPasswordLabel, gridBagConstraints);

jLabel1.setIcon(new javax.swing.ImageIcon(getClass().getResource("/GUI/images/bk3.jpg"))); // NOI18N

gridBagConstraints = new java.awt.GridBagConstraints();

gridBagConstraints.gridx = 0;

gridBagConstraints.gridy = 0;

gridBagConstraints.gridwidth = 9;

gridBagConstraints.gridheight = 5;

gridBagConstraints.anchor = java.awt.GridBagConstraints.NORTHWEST;

jPanel1.add(jLabel1, gridBagConstraints);

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());

getContentPane().setLayout(layout);

layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

);

layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

);

pack();

}// </editor-fold>//GEN-END:initComponents

private void cancelButtonActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_cancelButtonActionPerformed

System.exit(0);

}//GEN-LAST:event\_cancelButtonActionPerformed

private void loginButtonActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_loginButtonActionPerformed

con=Connect.ConnectDB();

String sql= "select \* from Users where Username= '" + userNameField.getText() + "' and User\_Password ='" + passwordField.getText() + "'";

try

{

pst=con.prepareStatement(sql);

rs= pst.executeQuery();

//(""==usernamField.getText())

if("".equals(userNameField.getText()) && "".equals(passwordField.getText())){

JOptionPane.showMessageDialog(null,"Please Enter Username and Password!");

}

else if (rs.next()){

this.hide();

MainMenu menu=new MainMenu();

menu.setVisible(true);

}

else{

JOptionPane.showMessageDialog(null, "Login Failed..Try again !","Access denied",JOptionPane.ERROR\_MESSAGE);

userNameField.setText("");

passwordField.setText("");

userNameField.requestFocus();

}

}

catch(SQLException | HeadlessException e)

{

JOptionPane.showMessageDialog(null, e);

}

}//GEN-LAST:event\_loginButtonActionPerformed

private void passwordFieldActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_passwordFieldActionPerformed

// TODO add your handling code here:

}//GEN-LAST:event\_passwordFieldActionPerformed

private void passwordFieldKeyPressed(java.awt.event.KeyEvent evt) {//GEN-FIRST:event\_passwordFieldKeyPressed

if(evt.getKeyCode()==KeyEvent.VK\_ENTER){

con=Connect.ConnectDB();

String sql= "select \* from Users where Username= '" + userNameField.getText() + "' and User\_Password ='" + passwordField.getText() + "'";

try

{

pst=con.prepareStatement(sql);

rs= pst.executeQuery();

if("".equals(userNameField.getText()) && "".equals(passwordField.getText())){

JOptionPane.showMessageDialog(null,"Please Enter Username and Password!");

}

else if (rs.next()){

this.hide();

MainMenu menu=new MainMenu();

menu.setVisible(true);

}

else{

JOptionPane.showMessageDialog(null, "Login Failed..Try again !","Access denied",JOptionPane.ERROR\_MESSAGE);

userNameField.setText("");

passwordField.setText("");

userNameField.requestFocus();

}

}

catch(SQLException | HeadlessException e)

{

JOptionPane.showMessageDialog(null, e);

}

}

}//GEN-LAST:event\_passwordFieldKeyPressed

/\*\*

\* @param args the command line arguments

\*/

public static void main(String args[]) {

/\* Set the Nimbus look and feel \*/

//<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">

/\* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.

\* For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html

\*/

try {

for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {

if ("Nimbus".equals(info.getName())) {

javax.swing.UIManager.setLookAndFeel(info.getClassName());

break;

}

}

} catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(Login.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(Login.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(Login.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(Login.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

}

//</editor-fold>

//</editor-fold>

//</editor-fold>

//</editor-fold>

/\* Create and display the form \*/

java.awt.EventQueue.invokeLater(new Runnable() {

public void run() {

new Login().setVisible(true);

}

});

}

// Variables declaration - do not modify//GEN-BEGIN:variables

private javax.swing.JButton cancelButton;

private javax.swing.JDialog confirmDialog;

private javax.swing.JLabel forgetPasswordLabel;

private javax.swing.JLabel jLabel1;

private javax.swing.JPanel jPanel1;

private javax.swing.JButton loginButton;

private javax.swing.JPasswordField passwordField;

private javax.swing.JLabel passwordLabel;

private javax.swing.JTextField userNameField;

private javax.swing.JLabel userNameLabel;

private javax.swing.JDialog welcomeDialog;

// End of variables declaration//GEN-END:variables

private void setIcon() {

//setIconImage(Toolkit.getDefaultToolkit().getImage(getClass().getResource("logout.png")));

}

}

**5.2 Main Menu**

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package GUI;

/\*\*

\*

\* @author M Azhar Durrani

\*/

public class MainMenu extends javax.swing.JFrame {

/\*\*

\* Creates new form MainMenu1

\*/

public MainMenu() {

initComponents();

}

/\*\*

\* This method is called from within the constructor to initialize the form.

\* WARNING: Do NOT modify this code. The content of this method is always

\* regenerated by the Form Editor.

\*/

@SuppressWarnings("unchecked")

// <editor-fold defaultstate="collapsed" desc="Generated Code">//GEN-BEGIN:initComponents

private void initComponents() {

jPanel1 = new javax.swing.JPanel();

addStudentButton = new javax.swing.JButton();

studentListButton = new javax.swing.JButton();

addStaffButton = new javax.swing.JButton();

addTeacherButton = new javax.swing.JButton();

teacherListButton = new javax.swing.JButton();

staffListButton = new javax.swing.JButton();

logoutButton = new javax.swing.JButton();

addUserButton = new javax.swing.JButton();

jLabel1 = new javax.swing.JLabel();

jLabel2 = new javax.swing.JLabel();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT\_ON\_CLOSE);

setTitle("Main Menu");

setMinimumSize(new java.awt.Dimension(740, 450));

setResizable(false);

jPanel1.setLayout(null);

addStudentButton.setBackground(new java.awt.Color(255, 255, 255));

addStudentButton.setFont(new java.awt.Font("Dialog", 1, 12)); // NOI18N

addStudentButton.setForeground(new java.awt.Color(0, 0, 0));

addStudentButton.setIcon(new javax.swing.ImageIcon(getClass().getResource("/GUI/images/Add.png"))); // NOI18N

addStudentButton.setText("Admission Form");

addStudentButton.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

addStudentButtonActionPerformed(evt);

}

});

jPanel1.add(addStudentButton);

addStudentButton.setBounds(30, 127, 180, 60);

studentListButton.setBackground(new java.awt.Color(255, 255, 255));

studentListButton.setFont(new java.awt.Font("Dialog", 1, 12)); // NOI18N

studentListButton.setForeground(new java.awt.Color(0, 0, 0));

studentListButton.setIcon(new javax.swing.ImageIcon(getClass().getResource("/GUI/images/Search.png"))); // NOI18N

studentListButton.setText("Student Data");

studentListButton.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

studentListButtonActionPerformed(evt);

}

});

jPanel1.add(studentListButton);

studentListButton.setBounds(30, 259, 180, 60);

addStaffButton.setBackground(new java.awt.Color(255, 255, 255));

addStaffButton.setFont(new java.awt.Font("Dialog", 1, 12)); // NOI18N

addStaffButton.setForeground(new java.awt.Color(0, 0, 0));

addStaffButton.setIcon(new javax.swing.ImageIcon(getClass().getResource("/GUI/images/Add.png"))); // NOI18N

addStaffButton.setText("Staff Form");

addStaffButton.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

addStaffButtonActionPerformed(evt);

}

});

jPanel1.add(addStaffButton);

addStaffButton.setBounds(531, 127, 180, 60);

addTeacherButton.setBackground(new java.awt.Color(255, 255, 255));

addTeacherButton.setFont(new java.awt.Font("Dialog", 1, 12)); // NOI18N

addTeacherButton.setForeground(new java.awt.Color(0, 0, 0));

addTeacherButton.setIcon(new javax.swing.ImageIcon(getClass().getResource("/GUI/images/Add.png"))); // NOI18N

addTeacherButton.setText("Teacher Form");

addTeacherButton.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

addTeacherButtonActionPerformed(evt);

}

});

jPanel1.add(addTeacherButton);

addTeacherButton.setBounds(280, 127, 180, 60);

teacherListButton.setBackground(new java.awt.Color(255, 255, 255));

teacherListButton.setFont(new java.awt.Font("Dialog", 1, 12)); // NOI18N

teacherListButton.setForeground(new java.awt.Color(0, 0, 0));

teacherListButton.setIcon(new javax.swing.ImageIcon(getClass().getResource("/GUI/images/Search.png"))); // NOI18N

teacherListButton.setText("Teacher Data");

teacherListButton.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

teacherListButtonActionPerformed(evt);

}

});

jPanel1.add(teacherListButton);

teacherListButton.setBounds(280, 259, 180, 60);

staffListButton.setBackground(new java.awt.Color(255, 255, 255));

staffListButton.setFont(new java.awt.Font("Dialog", 1, 12)); // NOI18N

staffListButton.setForeground(new java.awt.Color(0, 0, 0));

staffListButton.setIcon(new javax.swing.ImageIcon(getClass().getResource("/GUI/images/Search.png"))); // NOI18N

staffListButton.setText("Staff Data");

staffListButton.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

staffListButtonActionPerformed(evt);

}

});

jPanel1.add(staffListButton);

staffListButton.setBounds(531, 259, 180, 60);

logoutButton.setBackground(new java.awt.Color(255, 255, 255));

logoutButton.setFont(new java.awt.Font("Dialog", 1, 12)); // NOI18N

logoutButton.setForeground(new java.awt.Color(0, 0, 0));

logoutButton.setIcon(new javax.swing.ImageIcon(getClass().getResource("/GUI/images/logout.png"))); // NOI18N

logoutButton.setText("Logout");

logoutButton.setToolTipText("click here to logout");

logoutButton.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

logoutButtonActionPerformed(evt);

}

});

jPanel1.add(logoutButton);

logoutButton.setBounds(614, 367, 97, 30);

addUserButton.setBackground(new java.awt.Color(255, 255, 255));

addUserButton.setFont(new java.awt.Font("Dialog", 1, 12)); // NOI18N

addUserButton.setForeground(new java.awt.Color(0, 0, 0));

addUserButton.setText("+Add User");

addUserButton.setToolTipText("click here to add new user");

addUserButton.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

addUserButtonActionPerformed(evt);

}

});

jPanel1.add(addUserButton);

addUserButton.setBounds(30, 369, 93, 26);

jLabel1.setFont(new java.awt.Font("Dialog", 1, 36)); // NOI18N

jLabel1.setForeground(new java.awt.Color(255, 102, 102));

jLabel1.setText(" School Management System");

jLabel1.setBorder(javax.swing.BorderFactory.createLineBorder(new java.awt.Color(102, 255, 102), 3));

jPanel1.add(jLabel1);

jLabel1.setBounds(12, 12, 717, 50);

jLabel2.setIcon(new javax.swing.ImageIcon(getClass().getResource("/GUI/images/school.jpg"))); // NOI18N

jLabel2.setText("jLabel2");

jPanel1.add(jLabel2);

jLabel2.setBounds(0, 0, 740, 450);

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());

getContentPane().setLayout(layout);

layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT\_SIZE, 740, Short.MAX\_VALUE)

);

layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT\_SIZE, 450, Short.MAX\_VALUE)

);

pack();

}// </editor-fold>//GEN-END:initComponents

private void addStudentButtonActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_addStudentButtonActionPerformed

StudentForm student=new StudentForm();

this.hide();

student.setVisible(true);

}//GEN-LAST:event\_addStudentButtonActionPerformed

private void addUserButtonActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_addUserButtonActionPerformed

UserRegisteration user=new UserRegisteration();

this.hide();

user.setVisible(true);

}//GEN-LAST:event\_addUserButtonActionPerformed

private void addTeacherButtonActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_addTeacherButtonActionPerformed

TeacherForm teacher=new TeacherForm();

this.hide();

teacher.setVisible(true);

}//GEN-LAST:event\_addTeacherButtonActionPerformed

private void addStaffButtonActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_addStaffButtonActionPerformed

StaffForm staff=new StaffForm();

this.hide();

staff.setVisible(true);

}//GEN-LAST:event\_addStaffButtonActionPerformed

private void studentListButtonActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_studentListButtonActionPerformed

StudentRecord record=new StudentRecord();

this.hide();

record.setVisible(true);

}//GEN-LAST:event\_studentListButtonActionPerformed

private void teacherListButtonActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_teacherListButtonActionPerformed

TeacherRecord record=new TeacherRecord();

this.hide();

record.setVisible(true);

}//GEN-LAST:event\_teacherListButtonActionPerformed

private void staffListButtonActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_staffListButtonActionPerformed

StaffRecord record=new StaffRecord();

this.hide();

record.setVisible(true);

}//GEN-LAST:event\_staffListButtonActionPerformed

private void logoutButtonActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_logoutButtonActionPerformed

Login login=new Login();

this.setVisible(false);

login.setVisible(true);

}//GEN-LAST:event\_logoutButtonActionPerformed

/\*\*

\* @param args the command line arguments

\*/

public static void main(String args[]) {

/\* Set the Nimbus look and feel \*/

//<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">

/\* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.

\* For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html

\*/

try {

for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {

if ("Nimbus".equals(info.getName())) {

javax.swing.UIManager.setLookAndFeel(info.getClassName());

break;

}

}

} catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(MainMenu.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(MainMenu.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(MainMenu.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(MainMenu.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

}

//</editor-fold>

//</editor-fold>

/\* Create and display the form \*/

java.awt.EventQueue.invokeLater(new Runnable() {

public void run() {

new MainMenu().setVisible(true);

}

});

}

// Variables declaration - do not modify//GEN-BEGIN:variables

private javax.swing.JButton addStaffButton;

private javax.swing.JButton addStudentButton;

private javax.swing.JButton addTeacherButton;

private javax.swing.JButton addUserButton;

private javax.swing.JLabel jLabel1;

private javax.swing.JLabel jLabel2;

private javax.swing.JPanel jPanel1;

private javax.swing.JButton logoutButton;

private javax.swing.JButton staffListButton;

private javax.swing.JButton studentListButton;

private javax.swing.JButton teacherListButton;

// End of variables declaration//GEN-END:variables

}

**5.3 Student record-**

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package GUI;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import javax.swing.JOptionPane;

import net.proteanit.sql.DbUtils;

/\*\*

\*

\* @author M Azhar Durrani

\*/

public class StudentRecord extends javax.swing.JFrame {

Connection con=null;

ResultSet rs=null;

PreparedStatement pst=null;

/\*\*

\* Creates new form StudentRecord

\*/

public StudentRecord() {

initComponents();

con=Connect.ConnectDB();

Get\_Data();

setLocationRelativeTo(null);

}

private void Get\_Data(){

String sql = "select StudentID as [Student ID], FirstName"+" +"+"LastName as [Full Name], Class as [Class] from StudentRecord";

try{

pst = con.prepareStatement(sql);

rs = pst.executeQuery();

dataTable.setModel(DbUtils.resultSetToTableModel(rs));

}

catch(Exception ex){

JOptionPane.showMessageDialog(null, ex);

}

}

/\*\*

\* This method is called from within the constructor to initialize the form.

\* WARNING: Do NOT modify this code. The content of this method is always

\* regenerated by the Form Editor.

\*/

@SuppressWarnings("unchecked")

// <editor-fold defaultstate="collapsed" desc="Generated Code">//GEN-BEGIN:initComponents

private void initComponents() {

jScrollPane1 = new javax.swing.JScrollPane();

dataTable = new javax.swing.JTable();

jPanel1 = new javax.swing.JPanel();

teacherRecord = new javax.swing.JButton();

staffRecord = new javax.swing.JButton();

teacherForm = new javax.swing.JButton();

staffForm = new javax.swing.JButton();

jButton11 = new javax.swing.JButton();

backButton = new javax.swing.JButton();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT\_ON\_CLOSE);

setTitle("Student Record");

dataTable.setFont(new java.awt.Font("Dialog", 1, 12)); // NOI18N

dataTable.setModel(new javax.swing.table.DefaultTableModel(

new Object [][] {

{null, null, null, null, null, null, null, null, null, null, null, null, null, null},

new String [] {

"Title 1", "Title 2", "Title 3", "Title 4", "Title 5", "Title 6", "Title 7", "Title 8", "Title 9", "Title 10", "Title 11", "Title 12", "Title 13", "Title 14"

}

));

dataTable.addMouseListener(new java.awt.event.MouseAdapter() {

public void mouseClicked(java.awt.event.MouseEvent evt) {

dataTableMouseClicked(evt);

}

});

jScrollPane1.setViewportView(dataTable);

jPanel1.setBackground(new java.awt.Color(102, 102, 102));

teacherRecord.setText("Teacher Record");

teacherRecord.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

teacherRecordActionPerformed(evt);

}

});

staffRecord.setText("Staff Record");

staffRecord.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

staffRecordActionPerformed(evt);

}

});

teacherForm.setText("Teacher Form");

teacherForm.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

teacherFormActionPerformed(evt);

}

});

staffForm.setText("Staff Form");

staffForm.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

staffFormActionPerformed(evt);

}

});

jButton11.setText("Logout");

jButton11.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jButton11ActionPerformed(evt);

}

});

backButton.setText("Main Menu");

backButton.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

backButtonActionPerformed(evt);

}

});

javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);

jPanel1.setLayout(jPanel1Layout);

jPanel1Layout.setHorizontalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel1Layout.createSequentialGroup()

.addGap(96, 96, 96)

.addComponent(backButton)

.addGap(18, 18, 18)

.addComponent(teacherRecord)

.addGap(18, 18, 18)

.addComponent(staffRecord)

.addGap(18, 18, 18)

.addComponent(teacherForm)

.addGap(18, 18, 18)

.addComponent(staffForm)

.addGap(18, 18, 18)

.addComponent(jButton11)

.addContainerGap(javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))

);

jPanel1Layout.linkSize(javax.swing.SwingConstants.HORIZONTAL, new java.awt.Component[] {backButton, jButton11, staffForm, staffRecord, teacherForm, teacherRecord});

jPanel1Layout.setVerticalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, jPanel1Layout.createSequentialGroup()

.addContainerGap(24, Short.MAX\_VALUE)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(teacherForm)

.addComponent(staffForm)

.addComponent(jButton11))

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(teacherRecord)

.addComponent(staffRecord)

.addComponent(backButton)))

.addContainerGap())

);

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());

getContentPane().setLayout(layout);

layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jScrollPane1, javax.swing.GroupLayout.DEFAULT\_SIZE, 1050, Short.MAX\_VALUE)

.addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

);

layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addComponent(jPanel1, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(18, 18, 18)

.addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED\_SIZE, 308, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addContainerGap(64, Short.MAX\_VALUE))

);

pack();

}// </editor-fold>//GEN-END:initComponents

private void dataTableMouseClicked(java.awt.event.MouseEvent evt) {//GEN-FIRST:event\_dataTableMouseClicked

try{

con = Connect.ConnectDB();

int row = dataTable.getSelectedRow();

String tableClick = dataTable.getModel().getValueAt(row, 0).toString();

String sql = "select \* from StudentRecord where StudentID= "+tableClick;

pst = con.prepareStatement(sql);

rs = pst.executeQuery();

if(rs.next()){

this.hide();

StudentForm student=new StudentForm();

student.setVisible(true);

String add1 = rs.getString("StudentID");

student.studentIdField.setText(add1);

String add2 = rs.getString("FirstName");

student.firstNameField.setText(add2);

String add3 = rs.getString("LastName");

student.lastNameField.setText(add3);

String add4 = rs.getString("DateOfBirth");

student.dobField.setText(add4);

String add5 = rs.getString("FatherName");

student.fatherNameField.setText(add5);

String add6 = rs.getString("FatherCNIC");

student.cnicField.setText(add6);

String add7 = rs.getString("FatherPhone");

student.phoneField.setText(add7);

String add8 = rs.getString("FatherMobile");

student.mobileField.setText(add8);

String add9 = rs.getString("Address");

student.addressField.setText(add9);

String add10 = rs.getString("DOA");

student.doaField.setText(add10);

String add11 = rs.getString("LCA");

student.lastClassAttendedField.setText(add11);

String add12 = rs.getString("PSN");

student.lastSchoolAttendedField.setText(add12);

String add13 = rs.getString("Gender");

student.cmbGender.setSelectedItem(add13);

String add14 = rs.getString("Class");

student.admissionClass.setText(add14);

student.updateButton.setEnabled(true);

student.deleteButton.setEnabled(true);

student.saveButton.setEnabled(true);

//student.studentIdField.setEnabled(true);

}

}

catch(Exception ex){

JOptionPane.showMessageDialog(null, ex);

}

}//GEN-LAST:event\_dataTableMouseClicked

private void teacherRecordActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_teacherRecordActionPerformed

TeacherRecord record=new TeacherRecord();

this.hide();

record.setVisible(true);

}//GEN-LAST:event\_teacherRecordActionPerformed

private void staffRecordActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_staffRecordActionPerformed

StaffRecord record=new StaffRecord();

this.hide();

record.setVisible(true);

}//GEN-LAST:event\_staffRecordActionPerformed

private void teacherFormActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_teacherFormActionPerformed

TeacherForm form=new TeacherForm();

this.hide();

form.setVisible(true);

}//GEN-LAST:event\_teacherFormActionPerformed

private void staffFormActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_staffFormActionPerformed

StaffForm form=new StaffForm();

this.hide();

form.setVisible(true);

}//GEN-LAST:event\_staffFormActionPerformed

private void jButton11ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_jButton11ActionPerformed

Login login=new Login();

this.hide();

login.setVisible(true);

}//GEN-LAST:event\_jButton11ActionPerformed

private void backButtonActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_backButtonActionPerformed

/\*StudentForm menu=new StudentForm();

this.hide();

menu.setVisible(true);\*/

MainMenu m=new MainMenu();

m.setVisible(true);

this.hide();

}//GEN-LAST:event\_backButtonActionPerformed

private void formWindowClosing(java.awt.event.WindowEvent evt) {

this.hide();

StudentForm form = new StudentForm();

form.setVisible(true);

}

/\*\*

\* @param args the command line arguments

\*/

public static void main(String args[]) {

/\* Set the Nimbus look and feel \*/

//<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">

/\* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.

\* For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html

\*/

try {

for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {

if ("Nimbus".equals(info.getName())) {

javax.swing.UIManager.setLookAndFeel(info.getClassName());

break;

}

}

} catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(StudentRecord.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(StudentRecord.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(StudentRecord.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(StudentRecord.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

}

//</editor-fold>

/\* Create and display the form \*/

java.awt.EventQueue.invokeLater(new Runnable() {

public void run() {

new StudentRecord().setVisible(true);

}

});

}

// Variables declaration - do not modify//GEN-BEGIN:variables

private javax.swing.JButton backButton;

private javax.swing.JTable dataTable;

private javax.swing.JButton jButton11;

private javax.swing.JPanel jPanel1;

private javax.swing.JScrollPane jScrollPane1;

private javax.swing.JButton staffForm;

private javax.swing.JButton staffRecord;

private javax.swing.JButton teacherForm;

private javax.swing.JButton teacherRecord;

// End of variables declaration//GEN-END:variables

}

**5.4 Teacher Record-**

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package GUI;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import javax.swing.JOptionPane;

import net.proteanit.sql.DbUtils;

/\*\*

\*

\* @author M Azhar Durrani

\*/

public class TeacherRecord extends javax.swing.JFrame {

Connection con=null;

ResultSet rs=null;

PreparedStatement pst=null;

/\*\*

\* Creates new form TeacherRecord

\*/

public TeacherRecord() {

initComponents();

con=Connect.ConnectDB();

Get\_Data();

setLocationRelativeTo(null);

}

private void Get\_Data(){

String sql = "select TeacherID as [Teacher ID], FirstName as [First Name], LastName as [Last Name], FatherName as [Father Name], EmailAddress as [Email Address], ContactNo as [Contact Number], CNIC as [CNIC], Salary as [Salary], Designation as [Designation], AcademicQualification as [Academic Qualification], Address as [Address], Gender as [Gender] from TeacherRecord";

try{

pst = con.prepareStatement(sql);

rs = pst.executeQuery();

dataTableTeacher.setModel(DbUtils.resultSetToTableModel(rs));

}

catch(Exception ex){

JOptionPane.showMessageDialog(null, ex);

}

}

/\*\*

\* This method is called from within the constructor to initialize the form.

\* WARNING: Do NOT modify this code. The content of this method is always

\* regenerated by the Form Editor.

\*/

@SuppressWarnings("unchecked")

// <editor-fold defaultstate="collapsed" desc="Generated Code">//GEN-BEGIN:initComponents

private void initComponents() {

jScrollPane1 = new javax.swing.JScrollPane();

dataTableTeacher = new javax.swing.JTable();

jPanel1 = new javax.swing.JPanel();

backButton = new javax.swing.JButton();

staffRecord = new javax.swing.JButton();

teacherForm = new javax.swing.JButton();

staffForm = new javax.swing.JButton();

studentRecord = new javax.swing.JButton();

jButton11 = new javax.swing.JButton();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT\_ON\_CLOSE);

setTitle("Teacher Record");

dataTableTeacher.setFont(new java.awt.Font("Dialog", 1, 12)); // NOI18N

dataTableTeacher.setModel(new javax.swing.table.DefaultTableModel(

new Object [][] {

{null, null, null, null, null, null, null, null, null, null, null, null},

{null, null, null, null, null, null, null, null, null, null, null, null},

new String [] {

"Title 1", "Title 2", "Title 3", "Title 4", "Title 5", "Title 6", "Title 7", "Title 8", "Title 9", "Title 10", "Title 11", "Title 12"

}

));

dataTableTeacher.addMouseListener(new java.awt.event.MouseAdapter() {

public void mouseClicked(java.awt.event.MouseEvent evt) {

dataTableTeacherMouseClicked(evt);

}

});

jScrollPane1.setViewportView(dataTableTeacher);

backButton.setText("Main Menu");

backButton.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

backButtonActionPerformed(evt);

}

});

staffRecord.setText("Staff Record");

staffRecord.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

staffRecordActionPerformed(evt);

}

});

teacherForm.setText("Teacher Form");

teacherForm.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

teacherFormActionPerformed(evt);

}

});

staffForm.setText("Staff Form");

staffForm.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

staffFormActionPerformed(evt);

}

});

studentRecord.setText("Student Record");

studentRecord.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

studentRecordActionPerformed(evt);

}

});

jButton11.setText("Logout");

jButton11.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jButton11ActionPerformed(evt);

}

});

javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);

jPanel1.setLayout(jPanel1Layout);

jPanel1Layout.setHorizontalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, jPanel1Layout.createSequentialGroup()

.addContainerGap(javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(backButton)

.addGap(18, 18, 18)

.addComponent(staffRecord)

.addGap(18, 18, 18)

.addComponent(teacherForm)

.addGap(18, 18, 18)

.addComponent(staffForm)

.addGap(18, 18, 18)

.addComponent(studentRecord)

.addGap(18, 18, 18)

.addComponent(jButton11)

.addGap(41, 41, 41))

);

jPanel1Layout.linkSize(javax.swing.SwingConstants.HORIZONTAL, new java.awt.Component[] {backButton, jButton11, staffForm, staffRecord, studentRecord, teacherForm});

jPanel1Layout.setVerticalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, jPanel1Layout.createSequentialGroup()

.addContainerGap(javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(teacherForm)

.addComponent(staffForm)

.addComponent(studentRecord)

.addComponent(jButton11))

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(staffRecord)

.addComponent(backButton)))

.addContainerGap())

);

jPanel1Layout.linkSize(javax.swing.SwingConstants.VERTICAL, new java.awt.Component[] {backButton, jButton11, staffForm, staffRecord, studentRecord, teacherForm});

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());

getContentPane().setLayout(layout);

layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jScrollPane1, javax.swing.GroupLayout.DEFAULT\_SIZE, 1400, Short.MAX\_VALUE)

.addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

);

layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addComponent(jPanel1, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED\_SIZE, 215, javax.swing.GroupLayout.PREFERRED\_SIZE)

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

);

pack();

}// </editor-fold>//GEN-END:initComponents

private void dataTableTeacherMouseClicked(java.awt.event.MouseEvent evt) {//GEN-FIRST:event\_dataTableTeacherMouseClicked

try{

con = Connect.ConnectDB();

int row = dataTableTeacher.getSelectedRow();

String tableClick = dataTableTeacher.getModel().getValueAt(row, 0).toString();

String sql = "select \* from TeacherRecord where TeacherID='"+tableClick+"'";

pst = con.prepareStatement(sql);

rs = pst.executeQuery();

if(rs.next()){

this.hide();

TeacherForm teacher=new TeacherForm();

teacher.setVisible(true);

String add1 = rs.getString("TeacherID");

teacher.teacherIdField.setText(add1);

String add2 = rs.getString("FirstName");

teacher.firstNameField.setText(add2);

String add3 = rs.getString("LastName");

teacher.lastNameField.setText(add3);

String add4 = rs.getString("FatherName");

teacher.fatherNameField.setText(add4);

String add5 = rs.getString("EmailAddress");

teacher.emailAddress.setText(add5);

String add6 = rs.getString("ContactNo");

teacher.mobileField.setText(add6);

String add7 = rs.getString("CNIC");

teacher.cnicField.setText(add7);

String add8 = rs.getString("Salary");

teacher.salaryField.setText(add8);

String add9 = rs.getString("Designation");

teacher.designationField.setText(add9);

String add10 = rs.getString("AcademicQualification");

teacher.aqField.setText(add10);

String add11 = rs.getString("Address");

teacher.addressField.setText(add11);

String add12 = rs.getString("Gender");

teacher.cmbGender.setSelectedItem(add12);

String add14 = rs.getString("Experience");

teacher.experienceField.setText(add14);

String add13 = rs.getString("DateOfJoinning");

teacher.joinningDateField.setText(add13);

teacher.updateButton.setEnabled(true);

teacher.deleteButton.setEnabled(true);

teacher.saveButton.setEnabled(false);

}

}

catch(Exception ex){

JOptionPane.showMessageDialog(null, ex);

}

}//GEN-LAST:event\_dataTableTeacherMouseClicked

private void backButtonActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_backButtonActionPerformed

MainMenu m =new MainMenu();

m.setVisible(true);

this.hide();

}//GEN-LAST:event\_backButtonActionPerformed

private void staffRecordActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_staffRecordActionPerformed

StaffRecord record=new StaffRecord();

this.hide();

record.setVisible(true);

}//GEN-LAST:event\_staffRecordActionPerformed

private void teacherFormActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_teacherFormActionPerformed

TeacherForm form=new TeacherForm();

this.hide();

form.setVisible(true);

}//GEN-LAST:event\_teacherFormActionPerformed

private void staffFormActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_staffFormActionPerformed

StaffForm form=new StaffForm();

this.hide();

form.setVisible(true);

}//GEN-LAST:event\_staffFormActionPerformed

private void studentRecordActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_studentRecordActionPerformed

StudentRecord record = new StudentRecord();

this.hide();

record.setVisible(true);

}//GEN-LAST:event\_studentRecordActionPerformed

private void jButton11ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_jButton11ActionPerformed

Login login=new Login();

this.hide();

login.setVisible(true);

}//GEN-LAST:event\_jButton11ActionPerformed

/\*\*

\* @param args the command line arguments

\*/

public static void main(String args[]) {

/\* Set the Nimbus look and feel \*/

//<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">

/\* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.

\* For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html

\*/

try {

for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {

if ("Nimbus".equals(info.getName())) {

javax.swing.UIManager.setLookAndFeel(info.getClassName());

break;

}

}

} catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(TeacherRecord.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(TeacherRecord.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(TeacherRecord.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(TeacherRecord.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

}

//</editor-fold>

/\* Create and display the form \*/

java.awt.EventQueue.invokeLater(new Runnable() {

public void run() {

new TeacherRecord().setVisible(true);

}

});

}

// Variables declaration - do not modify//GEN-BEGIN:variables

private javax.swing.JButton backButton;

private javax.swing.JTable dataTableTeacher;

private javax.swing.JButton jButton11;

private javax.swing.JPanel jPanel1;

private javax.swing.JScrollPane jScrollPane1;

private javax.swing.JButton staffForm;

private javax.swing.JButton staffRecord;

private javax.swing.JButton studentRecord;

private javax.swing.JButton teacherForm;

// End of variables declaration//GEN-END:variables

}

**5.5 Employee Record-**

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package GUI;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import javax.swing.JOptionPane;

import net.proteanit.sql.DbUtils;

/\*\*

\*

\* @author M Azhar Durrani

\*/

public class StaffRecord extends javax.swing.JFrame {

Connection con=null;

ResultSet rs=null;

PreparedStatement pst=null;

/\*\*

\* Creates new form StaffRecord

\*/

public StaffRecord() {

initComponents();

con=Connect.ConnectDB();

Get\_Data();

setLocationRelativeTo(null);

}

private void Get\_Data(){

String sql = "select StaffID as [StaffID], FirstName+LastName as [Full Name], EmailAddress as [Email Address], Designation as [Designation],DateOfJoinning as [Date Of Joinning], Experience as [Experience] from StaffRecord";

try{

pst=con.prepareStatement(sql);

rs=pst.executeQuery();

dataTable.setModel(DbUtils.resultSetToTableModel(rs));

}

catch(Exception ex){

JOptionPane.showMessageDialog(null, ex);

}

}

/\*\*

\* This method is called from within the constructor to initialize the form.

\* WARNING: Do NOT modify this code. The content of this method is always

\* regenerated by the Form Editor.

\*/

@SuppressWarnings("unchecked")

// <editor-fold defaultstate="collapsed" desc="Generated Code">//GEN-BEGIN:initComponents

private void initComponents() {

jPanel1 = new javax.swing.JPanel();

jButton11 = new javax.swing.JButton();

studentRecord = new javax.swing.JButton();

staffForm = new javax.swing.JButton();

teacherForm = new javax.swing.JButton();

teacherRecord = new javax.swing.JButton();

backButton = new javax.swing.JButton();

jScrollPane1 = new javax.swing.JScrollPane();

dataTable = new javax.swing.JTable();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT\_ON\_CLOSE);

setTitle("Staff Record");

jButton11.setFont(new java.awt.Font("Dialog", 1, 12)); // NOI18N

jButton11.setText("Logout");

jButton11.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jButton11ActionPerformed(evt);

}

});

studentRecord.setFont(new java.awt.Font("Dialog", 1, 12)); // NOI18N

studentRecord.setText("Student Record");

studentRecord.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

studentRecordActionPerformed(evt);

}

});

staffForm.setFont(new java.awt.Font("Dialog", 1, 12)); // NOI18N

staffForm.setText("Staff Form");

staffForm.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

staffFormActionPerformed(evt);

}

});

teacherForm.setFont(new java.awt.Font("Dialog", 1, 12)); // NOI18N

teacherForm.setText("Teacher Form");

teacherForm.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

teacherFormActionPerformed(evt);

}

});

teacherRecord.setFont(new java.awt.Font("Dialog", 1, 12)); // NOI18N

teacherRecord.setText("Teacher Record");

teacherRecord.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

teacherRecordActionPerformed(evt);

}

});

backButton.setFont(new java.awt.Font("Dialog", 1, 12)); // NOI18N

backButton.setText("Main Menu");

backButton.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

backButtonActionPerformed(evt);

}

});

dataTable.setModel(new javax.swing.table.DefaultTableModel(

new Object [][] {

{null, null, null, null, null, null, null, null, null, null, null, null},

{null, null, null, null, null, null, null, null, null, null, null, null},

{null, null, null, null, null, null, null, null, null, null, null, null},

new String [] {

"Title 1", "Title 2", "Title 3", "Title 4", "Title 5", "Title 6", "Title 7", "Title 8", "Title 9", "Title 10", "Title 11", "Title 12"

};

dataTable.addMouseListener(new java.awt.event.MouseAdapter() {

public void mouseClicked(java.awt.event.MouseEvent evt) {

dataTableMouseClicked(evt);

}

jScrollPane1.setViewportView(dataTable);

javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);

jPanel1.setLayout(jPanel1Layout);

jPanel1Layout.setHorizontalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel1Layout.createSequentialGroup()

.addGap(55, 55, 55)

.addComponent(backButton)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addComponent(teacherRecord)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addComponent(teacherForm)

.addGap(18, 18, 18)

.addComponent(staffForm)

.addGap(18, 18, 18)

.addComponent(studentRecord)

.addGap(18, 18, 18)

.addComponent(jButton11)

.addContainerGap(javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))

.addComponent(jScrollPane1, javax.swing.GroupLayout.Alignment.TRAILING)

);

jPanel1Layout.linkSize(javax.swing.SwingConstants.HORIZONTAL, new java.awt.Component[] {backButton, jButton11, staffForm, studentRecord, teacherForm, teacherRecord});

jPanel1Layout.setVerticalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, jPanel1Layout.createSequentialGroup()

.addContainerGap(24, Short.MAX\_VALUE)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(teacherForm)

.addComponent(staffForm)

.addComponent(studentRecord)

.addComponent(jButton11)

.addComponent(teacherRecord)

.addComponent(backButton))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED\_SIZE, 321, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(6, 6, 6))

);

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());

getContentPane().setLayout(layout);

layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

);

layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup()

.addComponent(jPanel1, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addContainerGap(39, Short.MAX\_VALUE))

);

pack();

}// </editor-fold>//GEN-END:initComponents

private void dataTableMouseClicked(java.awt.event.MouseEvent evt) {//GEN-FIRST:event\_dataTableMouseClicked

try{

con = Connect.ConnectDB();

int row = dataTable.getSelectedRow();

String table\_Clicked = dataTable.getModel().getValueAt(row, 0).toString();

String sql = "select \* from StaffRecord where StaffID='"+table\_Clicked+"'";

pst = con.prepareStatement(sql);

rs = pst.executeQuery();

if(rs.next()){

this.hide();

StaffForm staff=new StaffForm();

staff.setVisible(true);

String add = rs.getString("StaffID");

staff.staffIdField.setText(add);

String add1 = rs.getString("FirstName");

staff.firstNameField.setText(add1);

String add2 = rs.getString("LastName");

staff.lastNameField.setText(add2);

String add3 = rs.getString("FatherName");

staff.fatherNameField.setText(add3);

String add4 = rs.getString("EmailAddress");

staff.emailAddressField.setText(add4);

String add5 = rs.getString("ContactNo");

staff.mobileField.setText(add5);

String add6 = rs.getString("CNIC");

staff.cnicField.setText(add6);

String add7 = rs.getString("Salary");

staff.salaryField.setText(add7);

String add8 = rs.getString("Designation");

staff.designationField.setText(add8);

String add9 = rs.getString("AcademicQualification");

staff.aqField.setText(add9);

String add10 = rs.getString("Address");

staff.addressField.setText(add10);

String add11 = rs.getString("Gender");

staff.cmbGender.setSelectedItem(add11);

String add12 = rs.getString("Experience");

staff.experienceField.setText(add12);

String add13 = rs.getString("DateOfBirth");

staff.dobField.setText(add13);

String add14 = rs.getString("DateOfJoinning");

staff.dojField.setText(add14);

}

}

catch(Exception ex){

JOptionPane.showMessageDialog(null, ex);

}

}//GEN-LAST:event\_dataTableMouseClicked

private void jButton11ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_jButton11ActionPerformed

Login login=new Login();

this.hide();

login.setVisible(true);

}//GEN-LAST:event\_jButton11ActionPerformed

private void studentRecordActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_studentRecordActionPerformed

StudentRecord record = new StudentRecord();

this.hide();

record.setVisible(true);

}//GEN-LAST:event\_studentRecordActionPerformed

private void staffFormActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_staffFormActionPerformed

StaffForm form=new StaffForm();

this.hide();

form.setVisible(true);

}//GEN-LAST:event\_staffFormActionPerformed

private void teacherFormActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_teacherFormActionPerformed

TeacherForm form=new TeacherForm();

this.hide();

form.setVisible(true);

}//GEN-LAST:event\_teacherFormActionPerformed

private void teacherRecordActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_teacherRecordActionPerformed

TeacherRecord record=new TeacherRecord();

this.hide();

record.setVisible(true);

}//GEN-LAST:event\_teacherRecordActionPerformed

private void backButtonActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_backButtonActionPerformed

MainMenu m=new MainMenu();

this.hide();

m.setVisible(true);

}//GEN-LAST:event\_backButtonActionPerformed

/\*\*

\* @param args the command line arguments

\*/

public static void main(String args[]) {

/\* Set the Nimbus look and feel \*/

//<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">

/\* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.

\* For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html

\*/

try {

for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {

if ("Nimbus".equals(info.getName())) {

javax.swing.UIManager.setLookAndFeel(info.getClassName());

break;

}

} catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(StaffRecord.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(StaffRecord.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(StaffRecord.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(StaffRecord.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

}

//</editor-fold>

/\* Create and display the form \*/

java.awt.EventQueue.invokeLater(new Runnable() {

public void run() {

new StaffRecord().setVisible(true);

}

});

}

// Variables declaration - do not modify//GEN-BEGIN:variables

private javax.swing.JButton backButton;

private javax.swing.JTable dataTable;

private javax.swing.JButton jButton11;

private javax.swing.JPanel jPanel1;

private javax.swing.JScrollPane jScrollPane1;

private javax.swing.JButton staffForm;

private javax.swing.JButton studentRecord;

private javax.swing.JButton teacherForm;

private javax.swing.JButton teacherRecord;

// End of variables declaration//GEN-END:variables

}

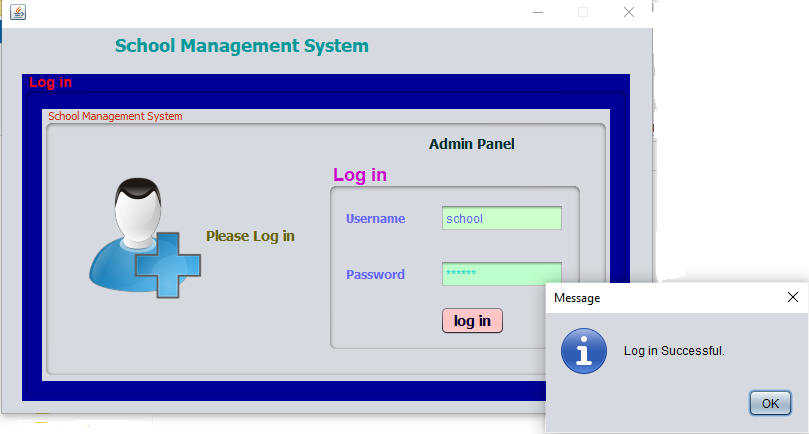
# CHAPTER 6

# Result

**6.1 LOGIN** **PAGE** **:**

**Figure 4.9 Login**

## LOGGED IN

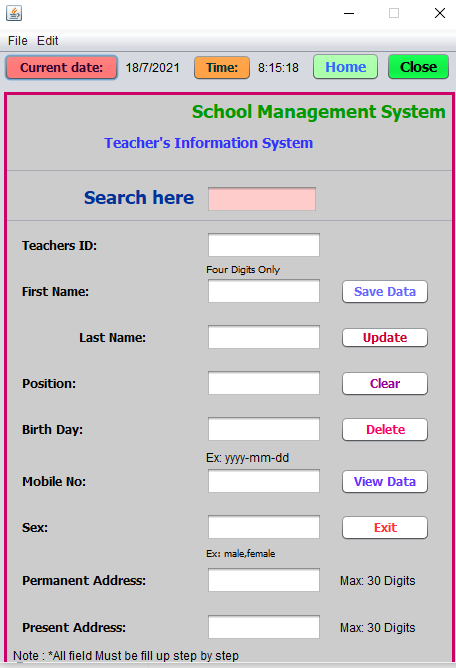


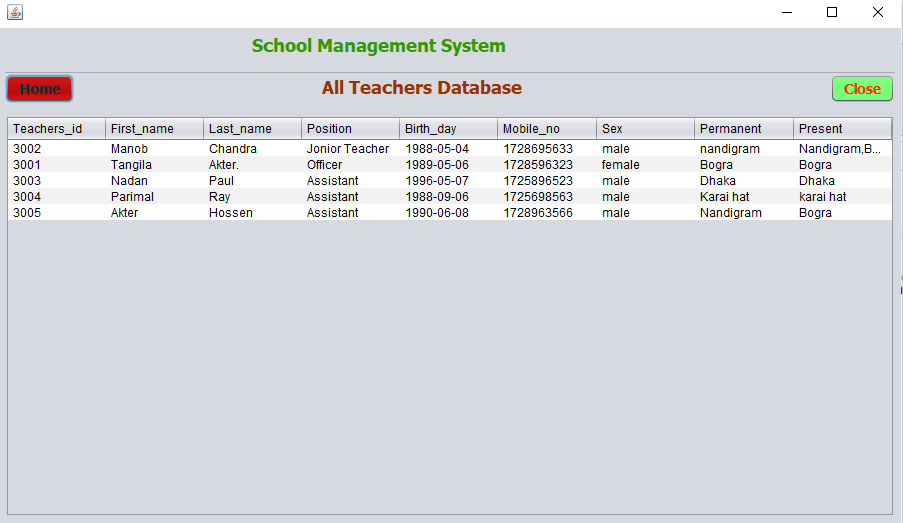
**Figure 5.1 Loggedin**

## 6.2 MAIN MENU

**Figure 5.2 Main Menu**

## TEACHER INFORMATION SYSTEM

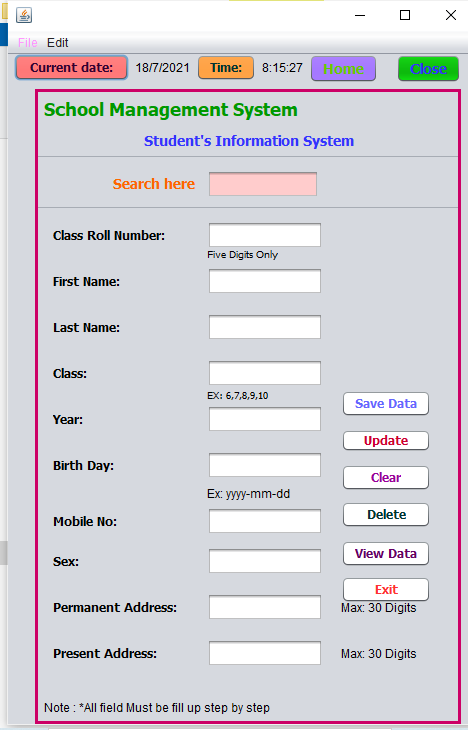




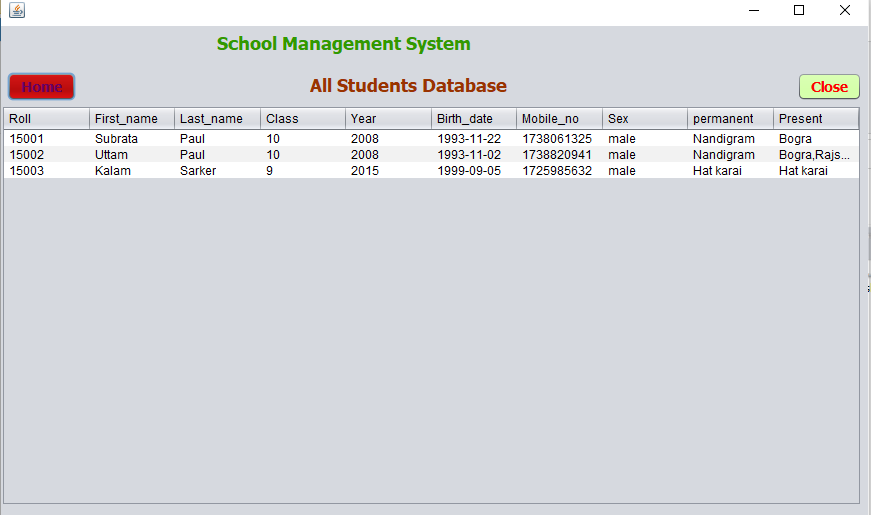
## ALL TEACHER DATABASE TABLE

**Table .6.1 Database**

## STUDENT INFORMATION SYSTEM



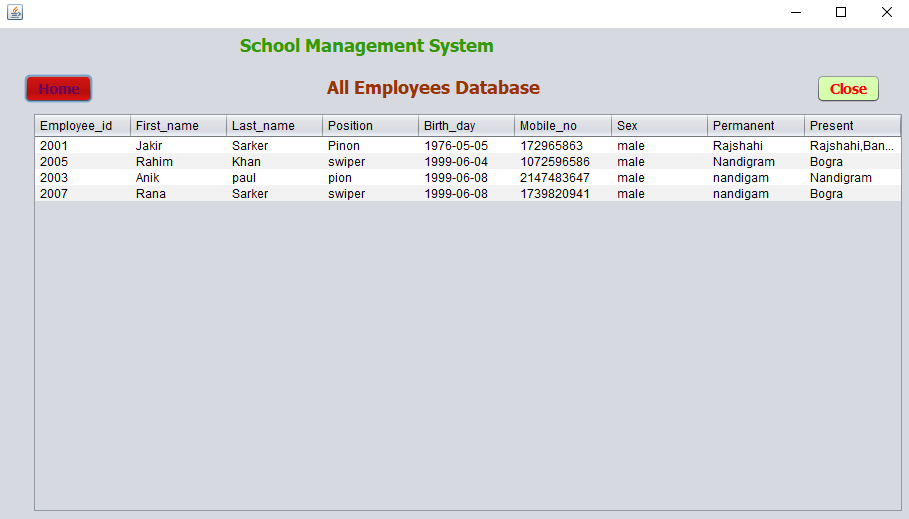
## ALL STUDENT DATABASE TABLE



**TABLE. 6.2 Database**

## EMPLOYEE’S INFORMATION SYSTEM





## ALL EMPLOYEES DATABASE TABLE

**TABLE NO. 6.3 Database**

# CHAPTER 7

# 7.1 Conclusion

In this project, we developed an automated school management system that facilitates the various activities taking place at schools. The system developed in the project consists of windows and web applications. These are two different applications on the same database.

The windows application takes most of the activities such as offline student registering, transcript and report card generation and producing the timetable. The web application facilitates attendance recording by the homeroom teachers and to view reports, to view status of students by students, teachers and parents.

Our solution of the timetabling problem is very simple. Data structures are used to implement the timetable designed. The scheduler selects a subject-teacher from the database, retrieves all the classes assigned to the teacher, calculates the load of the teacher which cannot be greater than the maximum load and selects one of the days randomly based on the number of lessons of the subject, searches a free appropriate time slot and assigns the slot to the lesson.

The scheduler repeats the process until the load of the teacher becomes zero and all the teachers in the database are visited. Finally the result generated is stored in a database

The prototype has been tested with data from Kobe Tsebin Secondary School. It has been shown that the system effectively registers students along with parental information, easily retrieves information about a student and generates the required reports such as transcript, report card and timetable.

In addition to generating a feasible master timetable it produces a timetable for each teacher.

Further more it has been shown that the web application of the system helps attendance recording by the homeroom teacher and parents can view the status of their children using the Internet or Intranet of the school.

# 7.2 Recommendations:

To enhance the efficiency of the system, in the following we have listed some recommendations and future works.

As education is central to development there should be a good facility to make stakeholders participate in school improvement programs and decision making. Parents and Education Bureaus from Kebele and Kifle-ketema are among the stake holders.

To facilitate easy information access to such bodies the web application could be further enhanced by incorporating additional reports required by Kebele and Kifle-ketema Education Bureaus. Such facilities will increase participants in decision making at educational activities and students achievement.

We also believe that timetables should be flexible. In real world situations there are preferences. A restriction of the sort that every teacher should have some specific free periods or some part of days off requires an efficient search technique. Efficiency of the timetable could be further enhanced by improving the search technique so that such constraints as preferences could be taken into consideration.

**References:**

1. M. Backes, S. Bugiel, E. Derr, P. McDaniel, D. Octeau and S. Weisgerber, "On demystifying the android application framework: Re-visiting android permission specification analysis", 25th {USENIX} Security Symposium ({USENIX) Security 16).
2. O. Cinar, "Android Platform" in Android Quick APIs Reference, Berkeley, CA:Apress, pp. 1-14, 2015.
3. S.Holla and M. M. Katti, "Android based mobile application development and its security", International Journal of Computer Trends and Technology, vol. 3, no. 3, pp. 486-490, 2012.
4. M. Latif, Y. Lakhrissi, E. H. Nfaoui and N. Es-Sbai, "Cross platform approach for mobile application development: A survey", 2016 International Conference on Information Technology for Organizations Development (IT40D), pp. 1-5, March 2016.
5. P. R. De Andrade, A. B. Albuquerque, O. F. Frota, R. V. Silveira and F. A. da Silva, Cross platform app: a comparative study, 2015
6. M. Latif, Y. Lakhrissi, E. H. Nfaoui and N. Es-Sbai, "Review of mobile cross platform and research orientations", 2017 International Conference on Wireless Technologies Embedded and Intelligent Systems (WITS), pp. 1-4, April 2017.
7. T. Bläsing, L. Batyuk, A. D. Schmidt, S. A. Camtepe and S. Albayrak, "An android application sandbox system for suspicious software detection", 2010 5th International Conference on Malicious and Unwanted Software, pp. 55-62, October 2010.
8. J. Philip and M. Raju, "A Formal Overview of Application Sandbox in Android and iOS with the Need to Secure Sandbox Against Increasing Number of Malware Attack", Indian Journal of Computer Science, vol. 4, no. 3, pp. 32-40, 2019.
9. M. Spreitzenbarth, T. Schreck, F. Echtler, D. Arp and J. Hoffmann, "Mobile-Sandbox: combining static and dynamic analysis with machine-learning techniques", International Journal of Information Security 14, no. 2, pp. 141-153, 2015.
10. X. Wang, K. Sun, Y. Wang and J. Jing, "DeepDroid: Dynamically Enforcing Enterprise Policy on Android Devices", Ndss, February 2015.
11. P29119-4-DISMay2013 - IEEE Draft International Standard for Software and Systems Engineering--Software Testing--Part 4: Test Techniques.
12. 15026-2-2011 - IEEE Standard--Adoption of ISO/IEC 15026-2:2011 Systems and Software Engineering--Systems and Software Assurance--Part 2: Assurance Case.
13. 730-2014 - IEEE Standard for Software Quality Assurance Processes.
14. 24765-2017 - ISO/IEC/IEEE International Standard - Systems and software engineering—Vocabulary.
15. 828-2012 - IEEE Standard for Configuration Management in Systems and Software Engineering.