

Project Presentation
for
School Management System

Submitted by –

Ankit Raj
159802231
BCA 6th Semester
Session 2015-2018

SCHOOL OF COMPUTER AND INFORMATION SCIENCE



INDEX

- 1. Project Proposal Performa**
- 2. Original Project Proposal Approval Letter**
- 3. Guides Biodata**
- 4. Certificate of genuinely**
- 5. Project Report**
- 6. Installer Guide**

Certificate

This is to certify that the project entitled, "**Aeon School Management System**" submitted by "**Ankit Raj**" whose "**Enrollment no is 159802231**", in partial fulfillment of the requirements for the award of "**Degree of Bachelor of Computer Applications**" at the "**Indira Gandhi National Open University, Regional center Ranchi**" is an original & authentic work carried out by Student under my (**Mr. Kaushal Kumar Singh**) supervision and guidance.

To the best of my knowledge, the matter embodied in the project is genuine work done by Student and it has not been submitted whether to this or any other University or Institute for the award of any Degree or Diploma.

Signature of Student:

Date:

Address:

Signature of guide:

Date:

Address:

Preface

In modern days it's becoming more and more difficult for schools to keep track of their data and activities related to their staffs and student, in such Scenario my SMS Project entitled “Aeon School Management System” will handle Most of the activities of the school. This SMS software has most of the facilities that a modern school requires to computerize its day-to-day jobs. It provides facilities to keep the records of student, teaching and non-teaching staff with all their required details. It has facilities to generate various types of reports, which are required by the management during normal business operations to operate the business effectively.

As the Software is developed under the modern UX guidelines so system will be very user-friendly and easy to use, I hope this system will be admired by the end users.

Acknowledgement

I would like to express my sincere gratitude and regards to my guide **Mr. Kaushal Kumar Singh** for his constant inspiration, supervision and invaluable guidance during the Project development. I would also like to thank our **Regional Director of IGNOU, Ranchi Regional Center and IGNOU** for giving me such an opportunity.

At last I would also like to extend my sincere gratitude to all our faculty members and developers at Stackoverflow.com and Lynda.com for giving their valuable suggestions.

With regards

Ankit RAJ (159802231)

BCA 6th Semester

Signature of Student

Project Report
on
School Management System

Submitted by –

Name	Ankit Raj
Enrollment No.	159802231
Course	BCA
Semester	6 th Semester
Session	2015-2018
Project Entitled as	AeonSMS

Under the Guidance of –
Mr. Kaushal Kumar

SCHOOL OF COMPUTER AND INFORMATION SCIENCE



SCHOOL MANAGEMENT SYSTEM

TABLE CONTENT

A. System Resource Specification

1. Problem Statement.....	3
1.1. Problem abstraction.....	3
1.2. Current System.....	3
1.3. Function requirements.....	4
1.4. The Proposed System	5
1.5. Boundaries of System.....	6
1.6. Hardware and Software requirements	6
1.7. System UI Design.....	7
2. Customer Requirements Specification.....	8
2.1. User of system.....	8
2.1.1. <i>Administrator</i>	8
2.1.2. <i>Staff Manager</i>	9
2.1.3. <i>Student Manager</i>	9
2.2. System functions:	10
2.2.1. Manage Admin.....	10
2.2.2. Manage Staff Manager.....	13
2.2.3. Manage Student Manager	18
3. Architecture & Design of Project	25
3.1. Data Flow Diagram	25
3.1.1. DFD level 0.....	25
3.1.2. DFD Level 1	26
3.1.3. DFD Level 2	29
3.2. Flow chart.....	47
3.2.1. Login	47
3.2.2. Add Records.....	48
3.2.3. Edit Records.....	49
3.2.4. Remove Records	50
3.2.5. Search and View Records	51
3.3. Entity Relationship Diagram	52
3.4. Database structure	54
3.4.1. Table Admin	54
3.4.2. Table User	54
3.4.3. Table Authentication.....	55
3.4.4. Table Staff.....	56

3.4.5.	Department.....	56
3.4.7.	Promotion.....	57
3.4.8.	Relatives staff.....	58
3.4.9.	Family Staff	58
3.4.10.	Student.....	59
3.4.11.	Admission.....	60
3.4.12.	Academic.....	60
3.4.13.	Batch.....	60
3.4.14.	Subject.....	61
3.4.15.	Mark	61
3.4.16.	Extra curricular.....	62
3.4.17.	Extra_curricular_has_Student	62
3.4.18.	Remark	63
3.4.19.	Feedback.....	63
3.4.20.	Relatives student	64
3.5.	Screen design/Document Workflow	65
1.	Splash Screen	65
2.	Database Setting	66
3.	Login	68
4.	Account information.....	73
5.	User Management Screen	75
6.	Staff Management Screen	76
7.	Department Management Screen.....	80
8.	Position Management Screen	83
9.	Student Management Screen.....	86
10.	Subject Management Screen	90
11.	Batch Management Screen	91
12.	Mark Management Screen	92
13.	Extra-Curricular Management Screen	93
14.	Export	94
15.	Logout	96
4.	Task Sheet.....	97
5.	System Development	98
5.1.	Source code	98
5.1.1.	Codes for Controls.....	98
5.1.2.	Codes for Database Integration	175
5.1.3.	Codes for Student Management Module	178
	• For Batch Management Module	178
6.	BIBLIOGRAPHY	299
7.	Future Perspectives	299

1. Problem Statement

1.1. Problem abstraction

Mr. Anonymous is heading the administration team of A Hypothetical high school. His team maintains the entire details of staff as well as students. This information is stored manually in various files. The report preparation becomes extremely time consuming as the information needs to be gathered from these files. Hence it has decided to computerize the management of this data which will manage the entire school data more effectively. Also this will retrieve data more quickly than that of the manual data files and save time. A “School Management System” needs to be developed which will help the school to manage all their activities.

1.2. Current System

Nowadays, Student and staff management system of A Hypothetical high school is performed through Files. Managing through files make a trouble in data inputting, out report. Simultaneously, it's waste time so much and lack of accuracy so it's too easy to lost the data.

The damages of Management system:

- Time Wasting
- Difficult to use
- Not exactly high
- Insecure
- Difficulty in backup, data migration when necessary.

1.3. *Function requirements*

The system includes 2 main functions are Manage student and manage staff.

The function of Manage student has some problems:

- Managing student, updating student's personal information's, family's information's, grade's information's, course, considering about remarking of teacher to student.
- Managing Batch
- Managing Subject
- Managing Mark
- Managing Extra-Curricular activities of school for student

The function of Manage staff has some problems:

- Managing Staff, updating personal information's, promotions, feedbacks of student forwards to the staff
- Managing Department
- Managing Position

1.4. The Proposed System

As above said, managing through files is very limitative. Automation is necessary insistence of Management. Management will become to easier, more exact and safer using the proposed System named AeonSMS (School Management System)

Benefit of System:

- Faster. Quick in updating information, searching information, out report. Then, save human resources.
- Easier. Using friendly and easy to use applications will not require knowledge too much at user. Contrary to this, using files requires users to understand Office's knowledge.
- More exact. Inputting from software and take out of database will helps the system processing more exact when compared with taking information out of files
- Saver. With managing through File, everyone can access to edit the information. And with this application, logging in by Username and Password is compulsory, guarantee for the best of security.
- Easy to back up and move. When necessary, the backup, move only simple is back up the database. It will more simple and safer when compared to you have to find the data

1.5. *Boundaries of System*

We built the system based on theory. In fact, there will be a few changes to the operating system more efficient.

1.6. *Hardware and Software requirements*

Used During Development

Hardware

- Processor Intel i5 7th Generation
- 8 Gigabytes of RAM or better
- Windows 10 Education

Software

- NetBeans IDE
- Icon8 for decent materialized Icons
- MySQL Server

After Development During Runtime

Hardware

- A minimum computer system that will help you access all the tools in the courses is a Pentium quad core processor or better
- 2 Gigabytes of RAM or better
- Windows 7 (or higher)

Software

- Java Virtual Machine
- MySQL Server

1.7. System UI Design

I have developed this School Management System Software (Named as AeonSMS) Using modern UX guidelines, Materialized color and design had been used to improve the UI and user interaction

2. Customer Requirements Specification

2.1. *User of system*

2.1.1. *Administrator*

- They are Administrator of System
- They can view, add, edit, reactive, active, remove a account of User
- Change password, email in their account
- They can manage Staff
- They manage Position
- They manage Department
- Manage below view/add/edit/reactive/active/remove object
- They can export list Staff, Department, Position to file
- Change password, email in their account
- They manage Student
- They manage Extra-curricular
- They manage Batch
- They manage Subject
- They manage Mark of Student
- They can export list Student, Mark, Batch, Subject, Extracurricular to file
- Change password, email in their account

2.1.2. Staff Manager

- They are Staff Manager of System
- They manage Staff
- They manage Position
- They manage Department
- Manage below view/add/edit/reactive/active/remove object
- They can export list Staff, Department, Position to file
- Change password, email in their account

2.1.3. Student Manager

- They are Student Manager of System
- They manage Student
- They manage Extra-curricular
- They manage Batch
- They manage Subject
- They manage Mark of Student
- They can export list Student, Mark, Batch, Subject, Extracurricular to file
- Change password, email in their account

2.2. System functions:

2.2.1. Manage Admin

- Login

Description	Function allows admin to login system
Input	Enter username, password, role
Process	Access to the database, check valid
Output	If login is successful go to GUI form, otherwise login is fail and show the message: “Username or Password is incorrect !” or “Connect fail !” if can’t connect to database

- Logout

Description	Click on the Logout button in menu, a logout confirms appear Click on “OK” button if they want to logout Click on “Cancel” button if player don’t want logout
Input	Logout action
Process	Escape from individual workspace
Output	Return Login screen

- View Account information

Description	Function allows users to view some information of account
Input	Click View info in menu My Account
Process	Access to the database. Retrieve data
Output	Display information: Username, Email, Action history

- Change Password

Description	Function allows users to change their password
Input	- Enter old password - Enter new password
Process	- Check valid old password in database - Edit new password to this user
Output	If OK show notice successful, otherwise notice unsuccessful

- Manager User Account

Description	Function allows Admin to search User, add new User , remove User already exist, change info about User already exist, reactive/active User
Input	<ul style="list-style-type: none"> - Search: enter Username, Name, Role - Add: Enter the necessary information (Username, Password, Name, Email, Role) - Remove: Choose ID want to Remove - Edit: Enter the information to change (Password, Name, Email, Role). - Deactivate/Active
Process	<ul style="list-style-type: none"> - Search: search database for Username, Name, Role - Add: add data to database - Remove: Remove data from database - Edit: Edit new data to database - Deactivate/Active: Change User status in database
Output	If OK show notice successful otherwise notice unsuccessful

2.2.2. Manage Staff Manager

- Login

Description	Function allows user to login system
Input	Enter username, password, role
Process	Access to the database, check valid
Output	If login is successful go to GUI form, otherwise login is fail and show the message: “Username or Password is incorrect !” or “Connect fail !” if can’t connect to database

- Logout

Description	Click on the Logout button in menu, a logout confirms appear Click on “OK” button if they want to logout Click on “Cancel” button if player don’t want logout
Input	Logout action
Process	Escape from individual workspace
Output	Return Login screen

- View Account Information

Description	Function allows users to view some information of account
Input	Click View info in menu My Account
Process	Access to the database. Retrieve data
Output	Display information: Username, Email, Action history

- Change Password

Description	Function allows users to change their password
Input	- Enter old password - Enter new password
Process	- Check valid old password in database - Edit new password to this user
Output	If OK show notice successful, otherwise notice unsuccessful

- Manage Staff

Description	Function allows Staff Manager to search Staff, add new Staff , Remove Staff already exist, change info about Staff already exist, deactivate/active Staff
Input	<ul style="list-style-type: none"> - Search: Enter Staff's Name, Position, Department, Gender, Birthday - Add: Enter the necessary information (Name, Phone, Gender, Email, Address, Birthday, Promotion Detail, Family Detail, Status) - Remove: Choose No want to Remove - Edit: Enter the information to change (Name, Phone, Gender, Email, Address, Birthday, Promotion Detail, Family Detail, Status) - Deactivate/Active: Choose Staff No want to deactivate/Active
Process	<ul style="list-style-type: none"> - Search: search database for Name, Position, Department, Gender, Birthday - Add: add data to database - Remove: Remove data from database - Edit: Edit new data to database - Deactivate/Active: Change Staff status in database
Output	If OK show notice successful otherwise notice unsuccessful

- Manage Department

Description	Function allows Staff Manager to search Staff, add new Staff , Remove Staff already exist, change info about Staff already exist, deactivate/active Staff
Input	<ul style="list-style-type: none"> - Search: Enter Staff's Name, Position, Department, Gender, Birthday - Add: Enter the necessary information (Name, Phone, Gender, Email, Address, Birthday, Promotion Detail, Family Detail, Status) - Remove: Choose No want to Remove - Edit: Enter the information to change (Name, Phone, Gender, Email, Address, Birthday, Promotion Detail, Family Detail, Status) - Deactivate/Active: Choose Staff No want to deactivate/Active
Process	<ul style="list-style-type: none"> - Search: search database for Name, Position, Department, Gender, Birthday - Add: add data to database - Remove: Remove data from database - Edit: Edit new data to database - Deactivate/Active: Change Department status in database
Output	If OK show notice successful otherwise notice unsuccessful

- Manage Position

Description	Function allows Position Manager to search Position, add new Position, Remove Position already exist, change info about Position already exist, deactivate/active Position
Input	<ul style="list-style-type: none"> - Search: Enter Position's Name, Department, - Add: Enter the necessary information (Name, Department) - Remove: Choose Position want to Remove - Edit: Enter the information to change (Name, Department) - Deactivate/Active: Choose Position want to Deactivate/Active
Process	<ul style="list-style-type: none"> - Search: search database for Name, Department - Add: add data to database - Remove: Remove data from database - Edit: Edit new data to database - Deactivate/Active: Change Position status in database
Output	If OK show notice successful otherwise notice unsuccessful

2.2.3. Manage Student Manager

- Login

Description	Function allows users to login system
Input	Enter username & password
Process	Access to the database, check valid
Output	If login is successful go to GUI form, otherwise login is fail and show the message: “Username or Password is incorrect”

- Logout

Description	Click on the Logout button, a logout confirms appear Click on “OK” button if player wants to logout Click on “Cancel” button if player don’t want logout
Input	Logout action
Process	Escape from individual workspace
Output	Return presentation screen

- Change Password

Description	Function allows users to change their password
Input	- Enter old password - Enter new password
Process	- Check valid old password in database - Edit new password to this user
Output	If OK show notice successful, otherwise notice unsuccessful

- Manage Student

Description	Function allows Student Manager to search Student, add new Student, Remove Student already exist, change info about Student already exist, deactivate/active Student
Input	<ul style="list-style-type: none"> - Search: Enter Student's Name, Register No, No, Batch - Add: Enter the necessary information (Name, Phone, Gender, Email, Address, Birthday, Family Detail, Admission Detail,) - Remove: Choose Student want to Remove - Edit: Enter the information to change (Name, Phone, Gender, Email, Address, Birthday, Family Detail, Admission Detail,) - Deactivate/Active: Choose Student No want to Deactivate/Active
Process	<ul style="list-style-type: none"> - Search: search database for Name, Register No, No, Batch - Add: add data to database - Remove: Remove data from database - Edit: Edit new data to database - deactivate/Active: Change Student status in database
Output	If OK show notice successful otherwise notice unsuccessful

- Manage Extra Curricular

Description	Function allows Student Manager to search Extra Curricular, add new Extra Curricular, Remove Extra Curricular already exist, change info about Extra Curricular already exist, deactivate/active Extra Curricular
Input	<ul style="list-style-type: none"> - Search: Enter Extra Curricula's Name, Start Date, End Date - Add: Enter the necessary information (Name, Start Date, End Date) - Remove: Choose No want to Remove - Edit: Enter the information to change (Name, Start Date, End Date) - deactivate/Active: Choose Extra Curricular want to deactivate/Active
Process	<ul style="list-style-type: none"> - Search: search database for Name, Start Date, End Date - Add: add data to database - Remove: Remove data from database - Edit: Edit new data to database - Deactivate/Active: Change Extra-Curricular status in database
Output	If OK show notice successful otherwise notice unsuccessful

- Manage Batch

Description	Function allows Student Manager to search Batch, add new Batch, Remove Batch already exist, change info about Batch already exist, deactivate/active Batch
Input	<ul style="list-style-type: none"> - Search: Enter Batch's Name - Add: Enter the necessary information (Name, Created date) - Remove: Choose Batch want to Remove - Edit: Enter the information to change (Name, Created) - Deactivate/Active: Choose Batch want to Deactivate/Active
Process	<ul style="list-style-type: none"> - Search: search database for Name, Position, Department, Gender, Birthday - Add: add data to database - Remove: Remove data from database - Edit: Edit new data to database - Deactivate/Active: Change Batch status in database
Output	If OK show notice successful otherwise notice unsuccessful

- Manage Subject

Description	Function allows Student Manager to search Subject, add new Subject, Remove Subject already exist, change info about Subject already exist, deactivate/active Subject
Input	<ul style="list-style-type: none"> - Search: Enter Subject's Name - Add: Enter Subject's Name - Remove: Choose Subject want to Remove - Edit: Enter Subject's Name - Deactivate/Active: Choose Subject want to Deactivate/Active
Process	<ul style="list-style-type: none"> - Search: search database for Name - Add: add data to database - Remove: Remove data from database - Edit: Edit new data to database - Deactivate/Active: Change Subject status in database
Output	If OK show notice successful otherwise notice unsuccessful

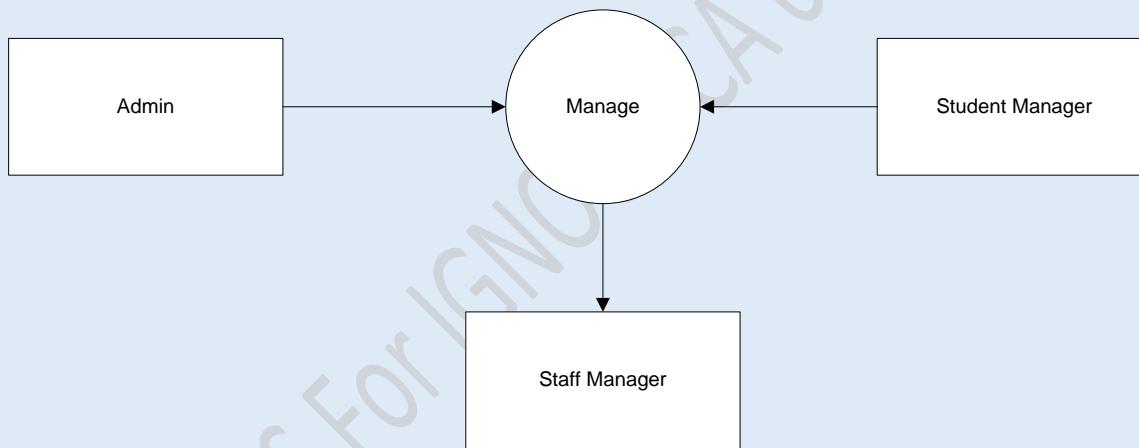
- Manage Mark

Description	Function allows Student Manager to search Mark, add new Mark, Remove Mark already exist, change info about Mark already exist,
Input	<ul style="list-style-type: none"> - Search: Enter Student's No, Student's Name, Subject's name, Created Date - Add: Enter the necessary information (Student's No, Student's Name, Subject's name, Created Date) - Remove: Choose Mark want to Remove - Edit: Enter the information to change (Student's No, Student's Name, Subject's name, Created Date)
Process	<ul style="list-style-type: none"> - Search: search database for Name, Position, Department, Gender, Birthday - Add: add data to database - Remove: Remove data from database - Edit: Edit new data to database
Output	If OK show notice successful otherwise notice unsuccessful

3. Architecture & Design of Project

3.1. Data Flow Diagram

3.1.1. DFD level 0

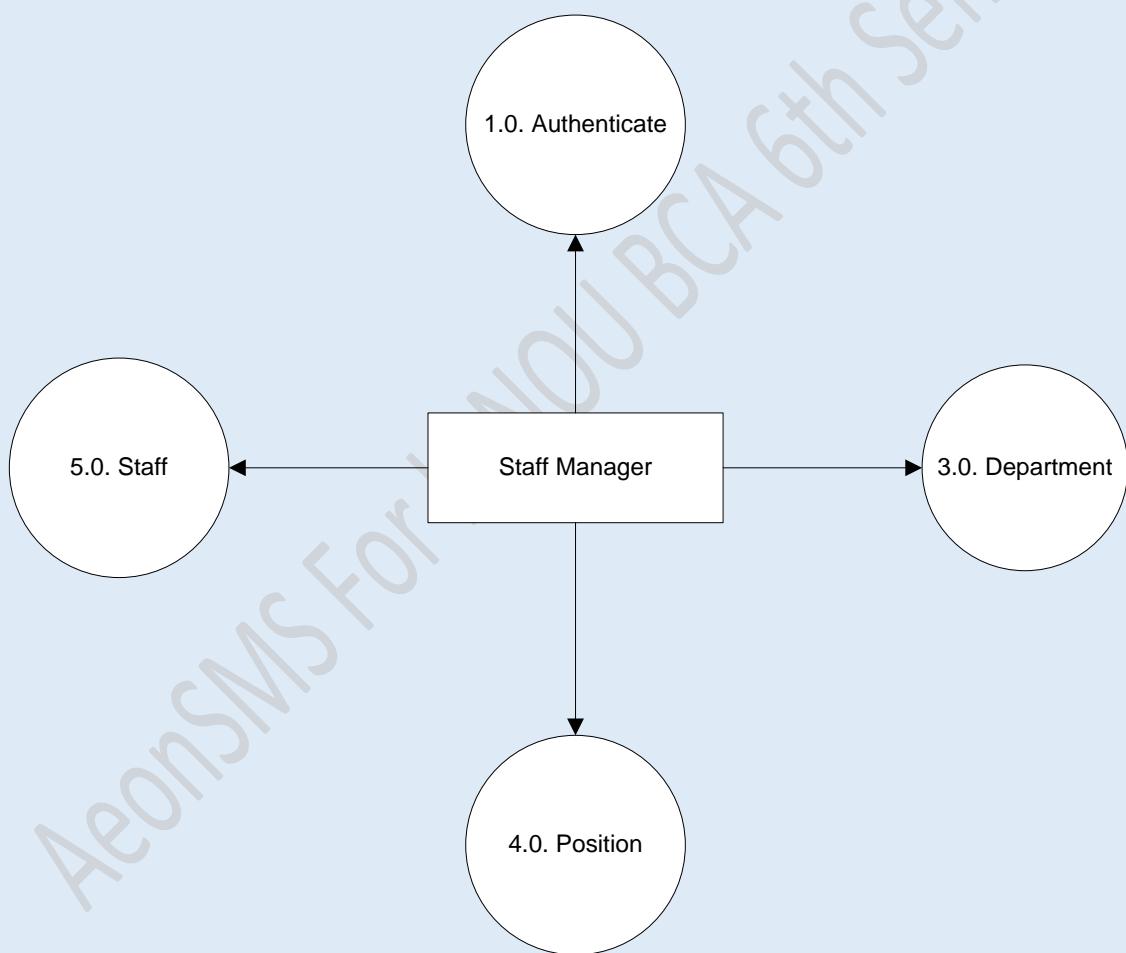


3.1.2. DFD Level 1

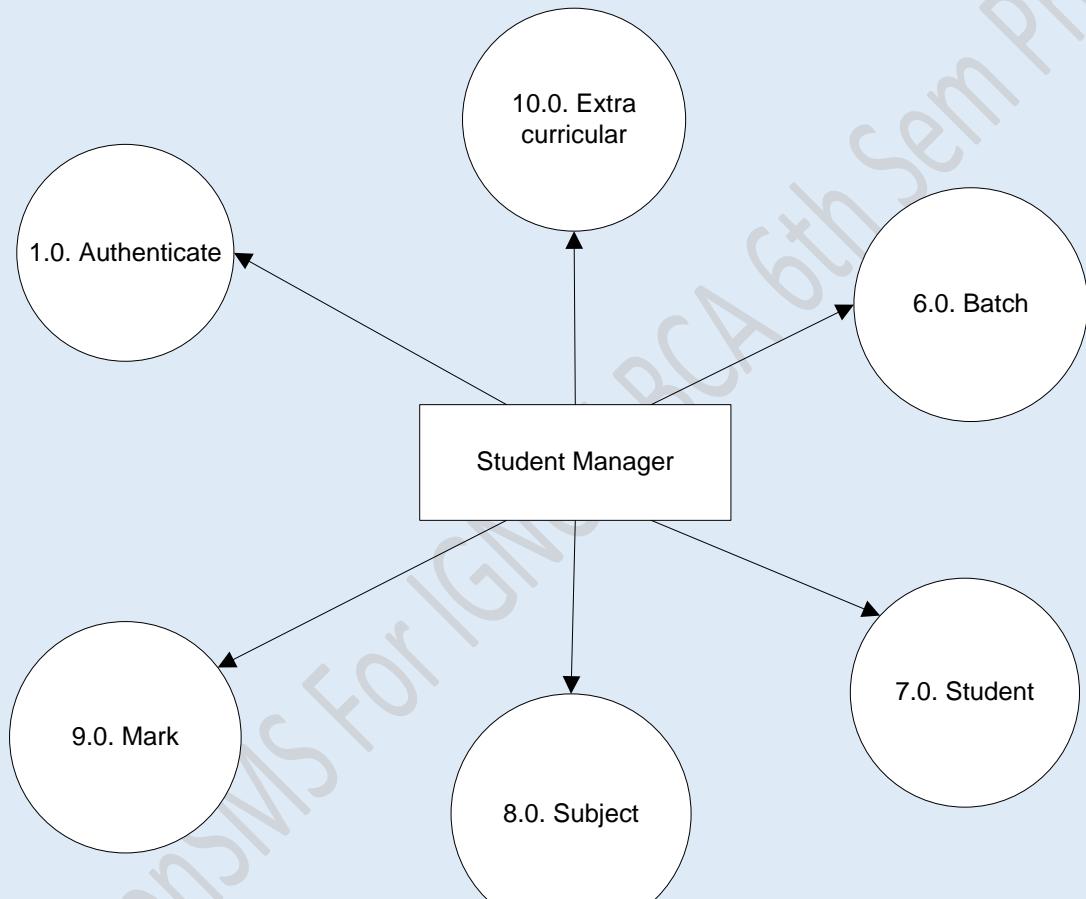
Level 1.1. Admin



Level 1.2. Staff Manager

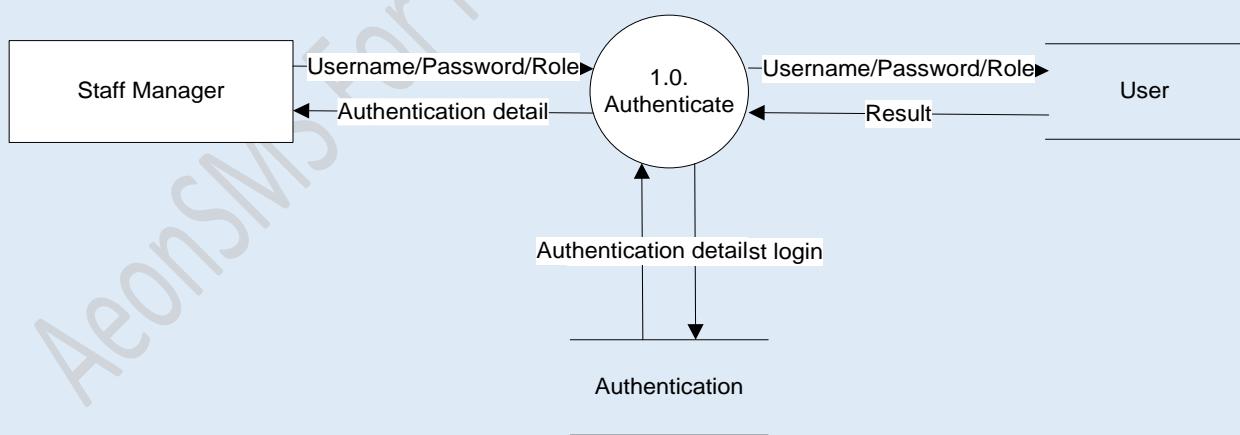
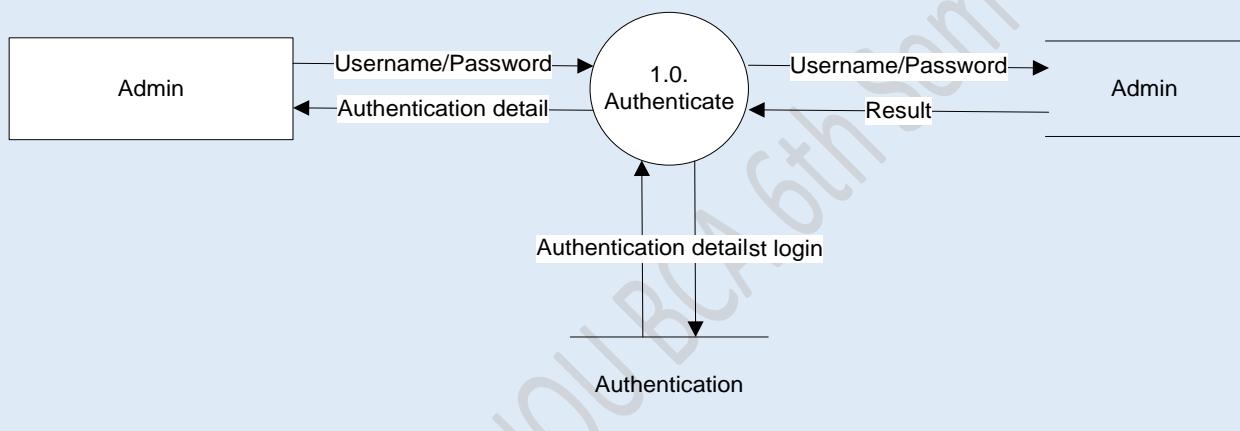


Level 1.3. Student Manager

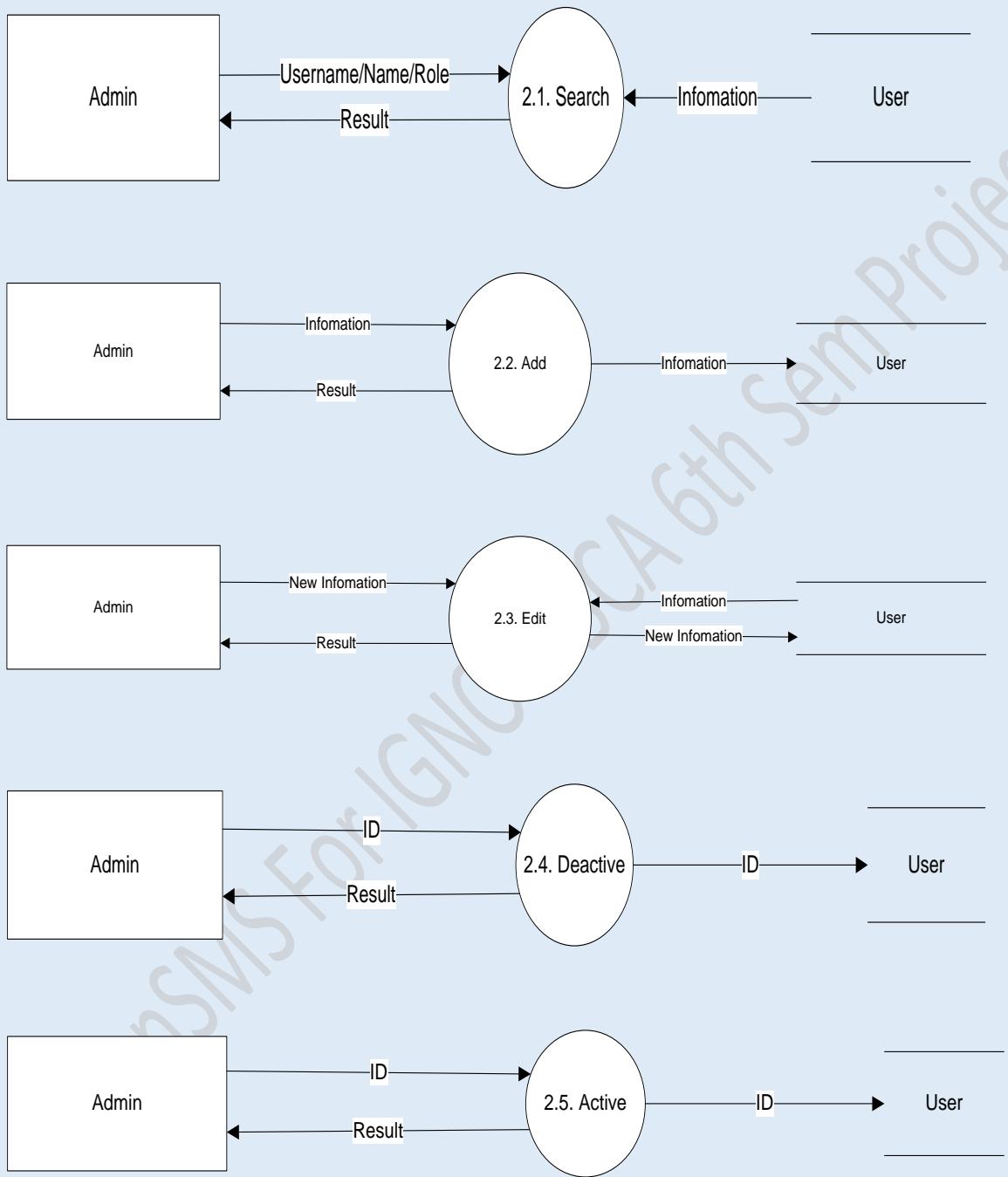


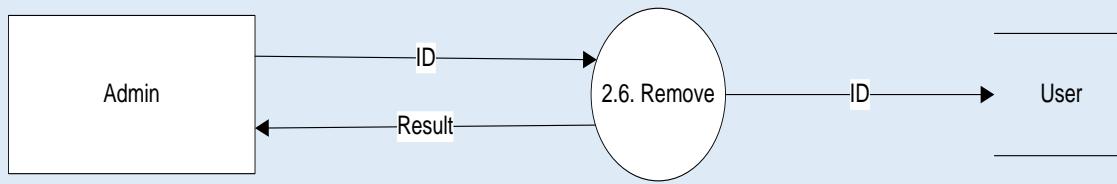
3.1.3. DFD Level 2

Level 2.1. Authentication

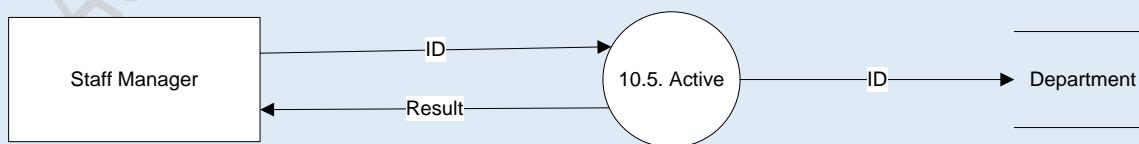
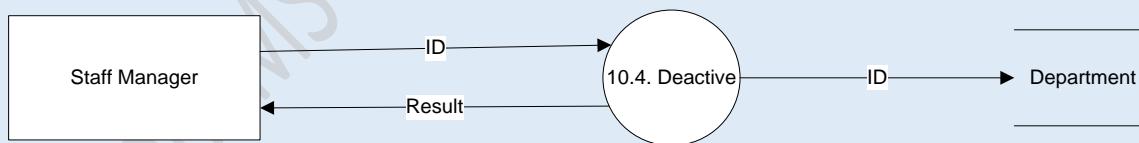
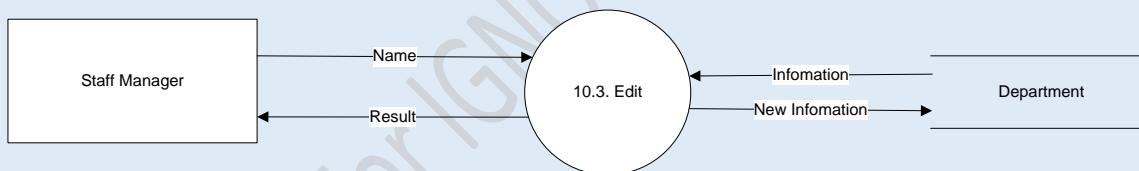
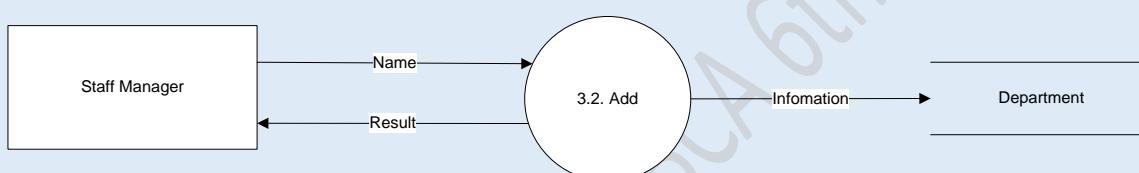
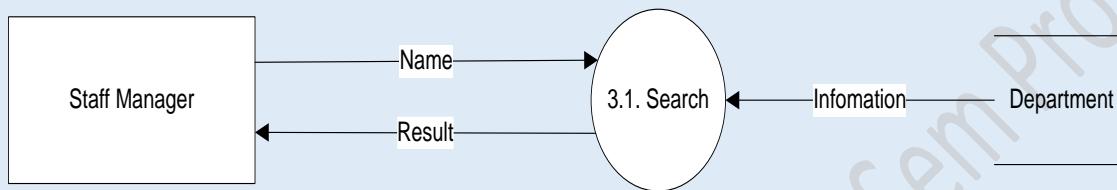


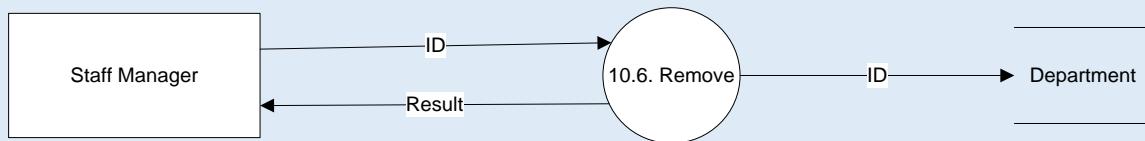
Level 2.2. Manage User Account



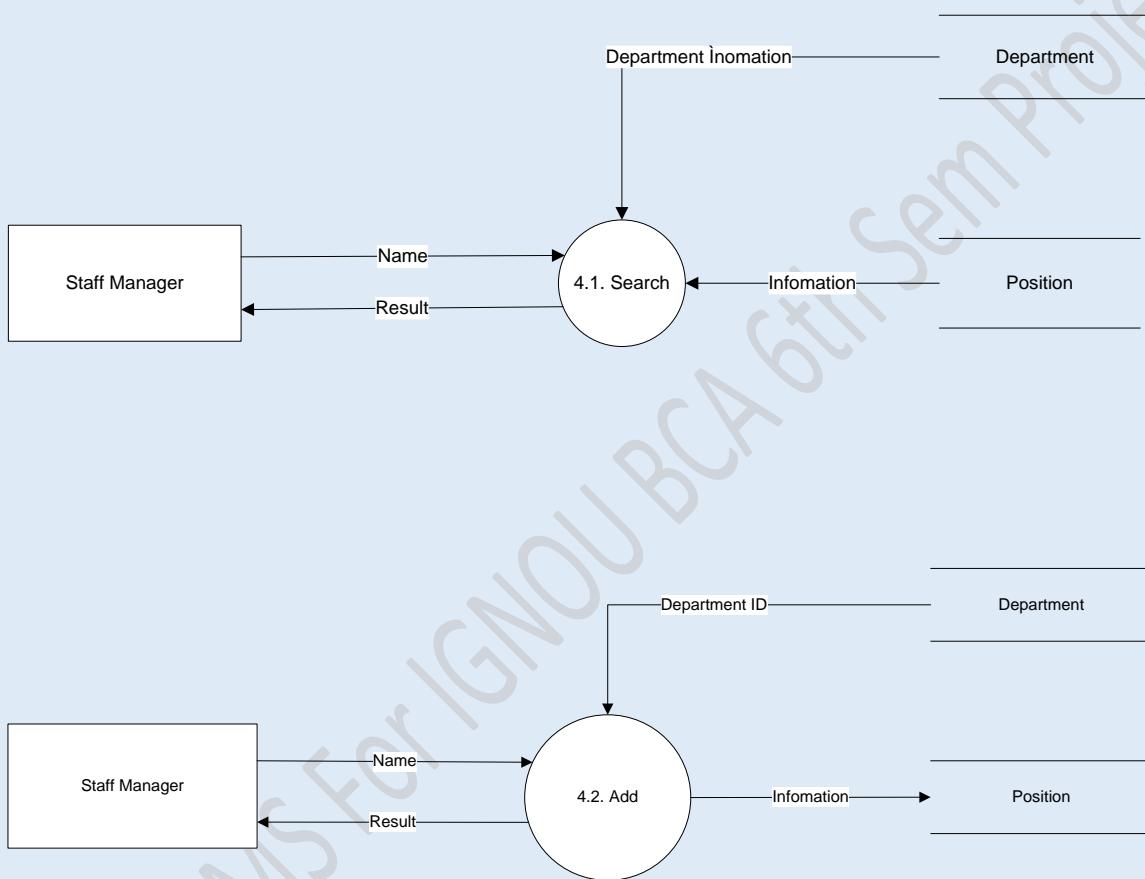


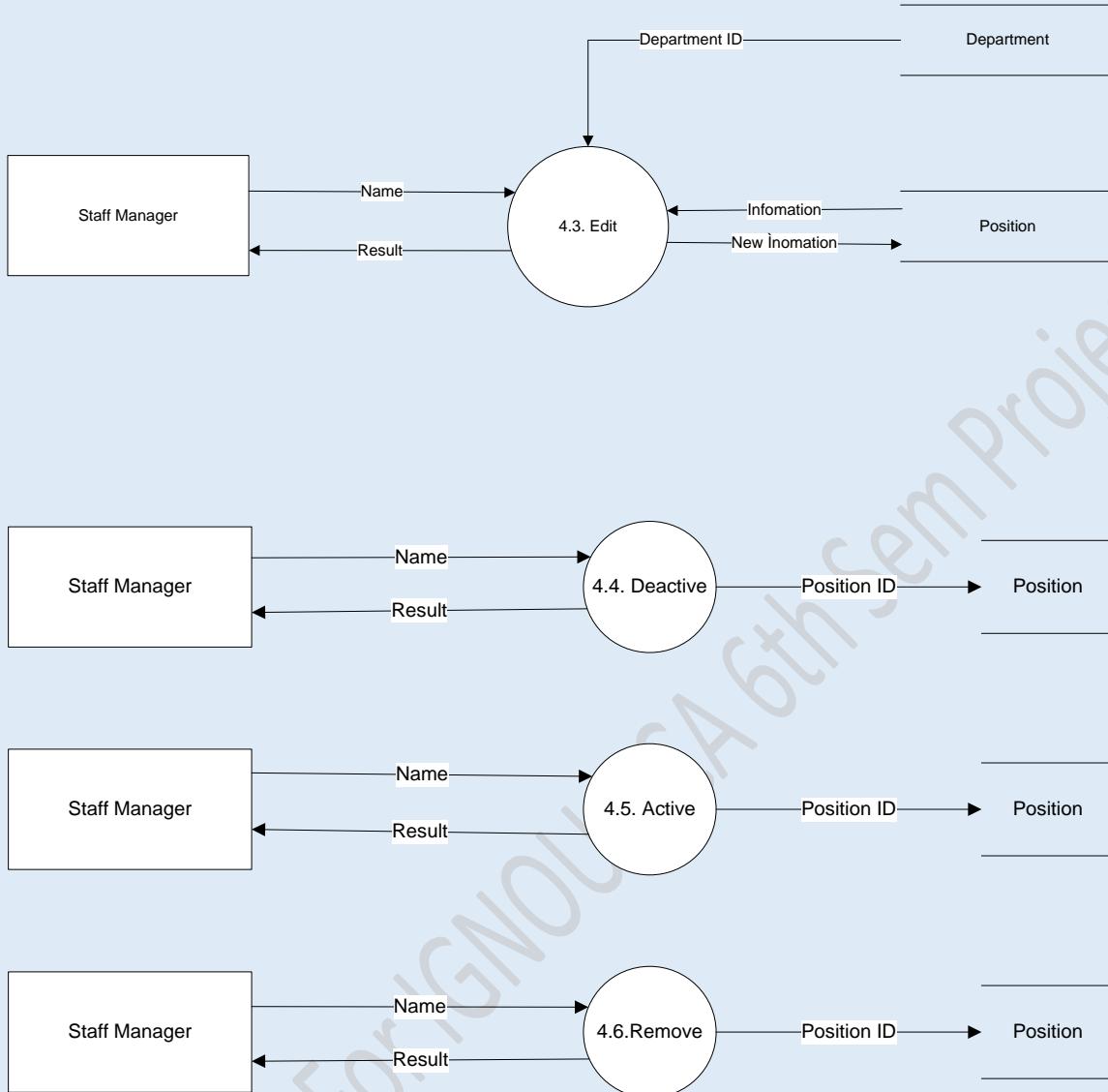
Level 2.3. Manage Department



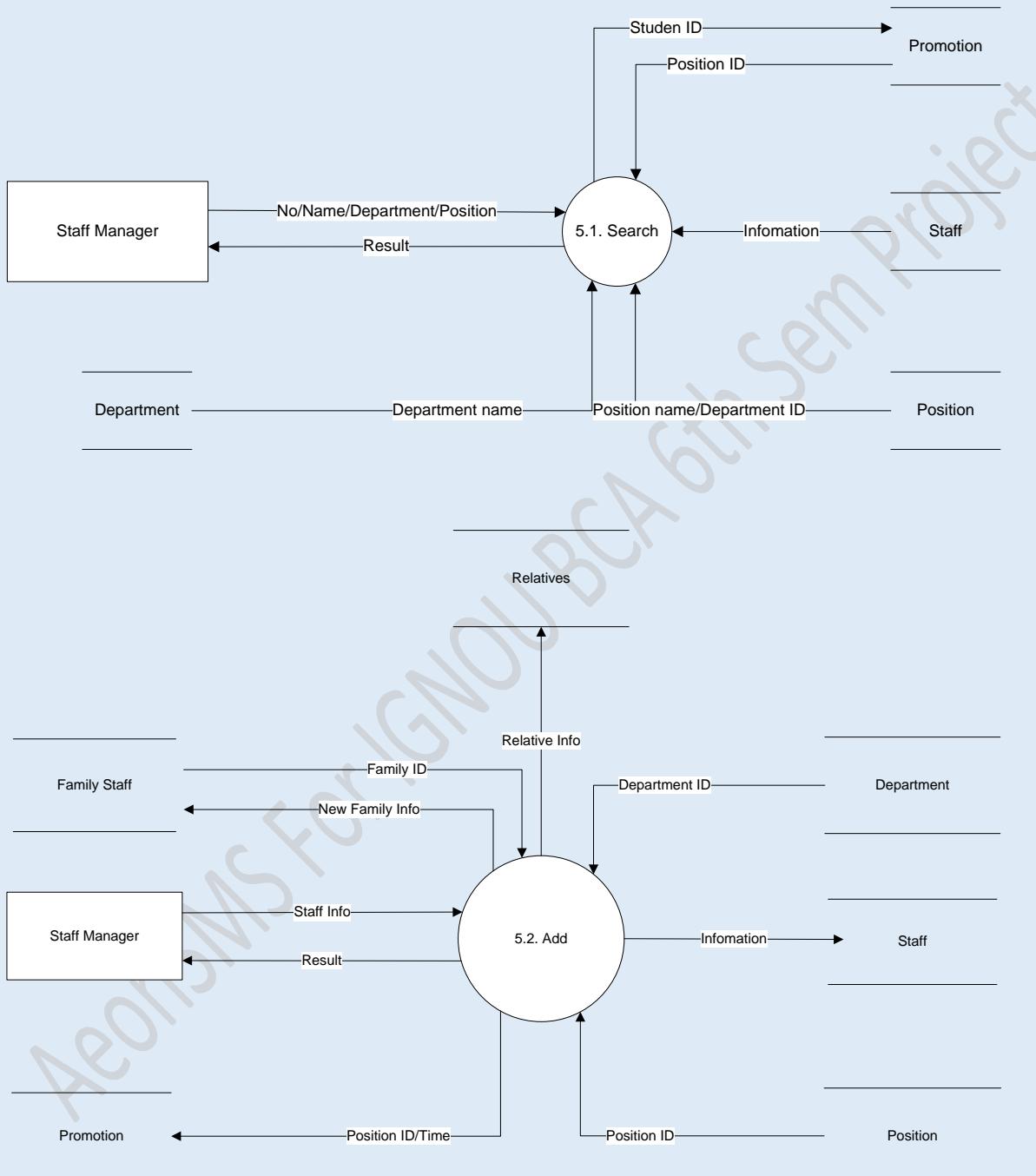


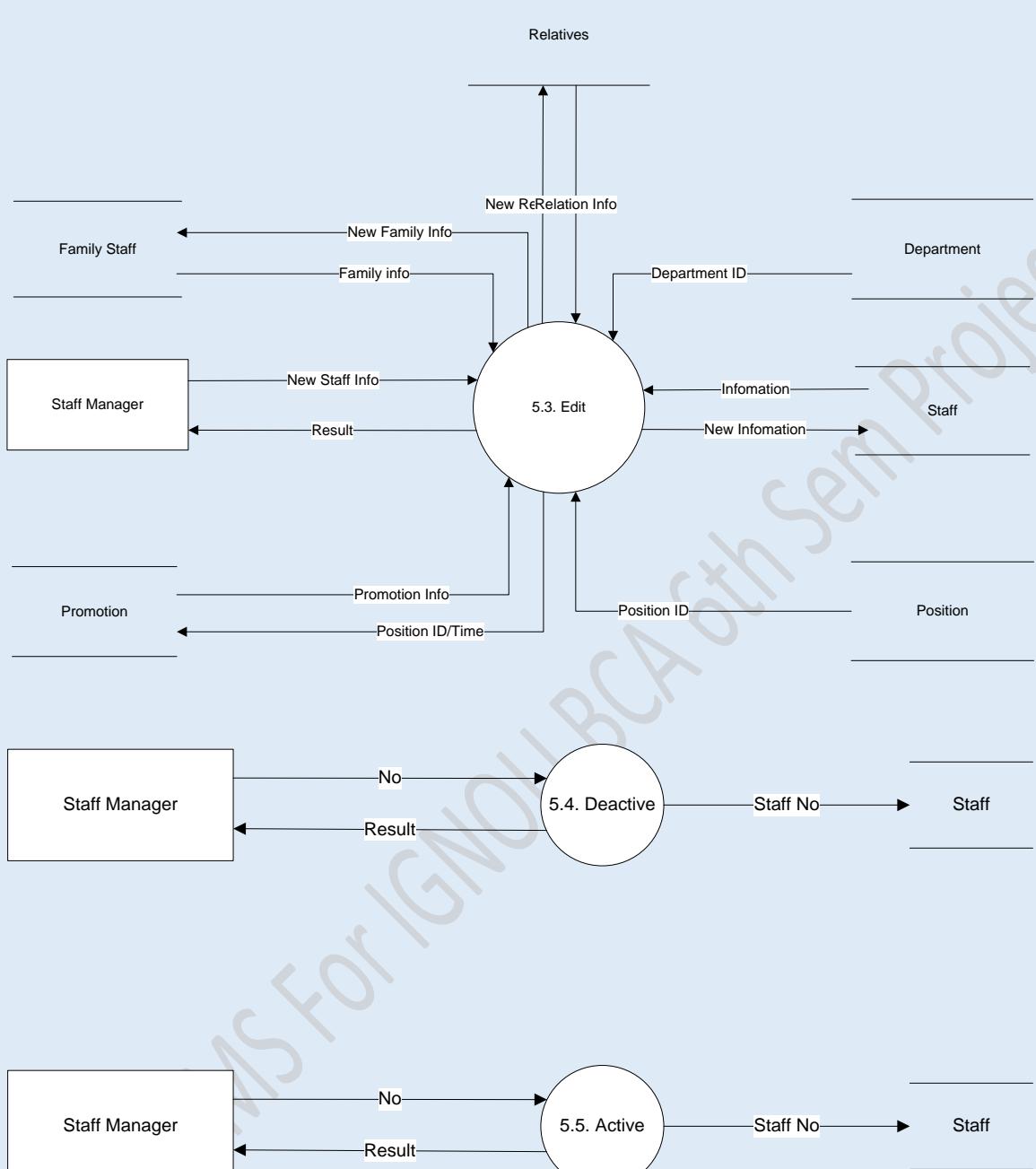
Level 2.4. Manage Position

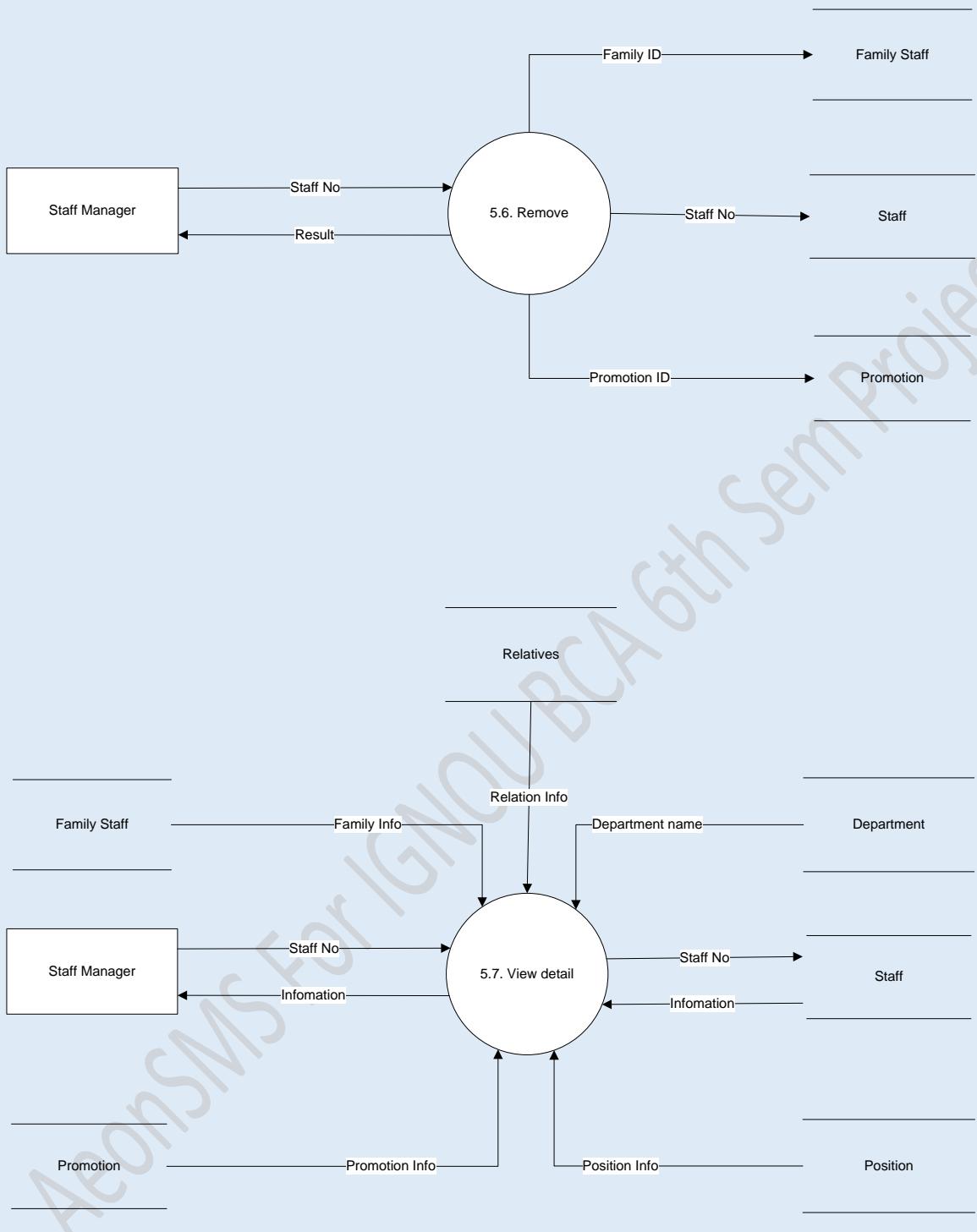




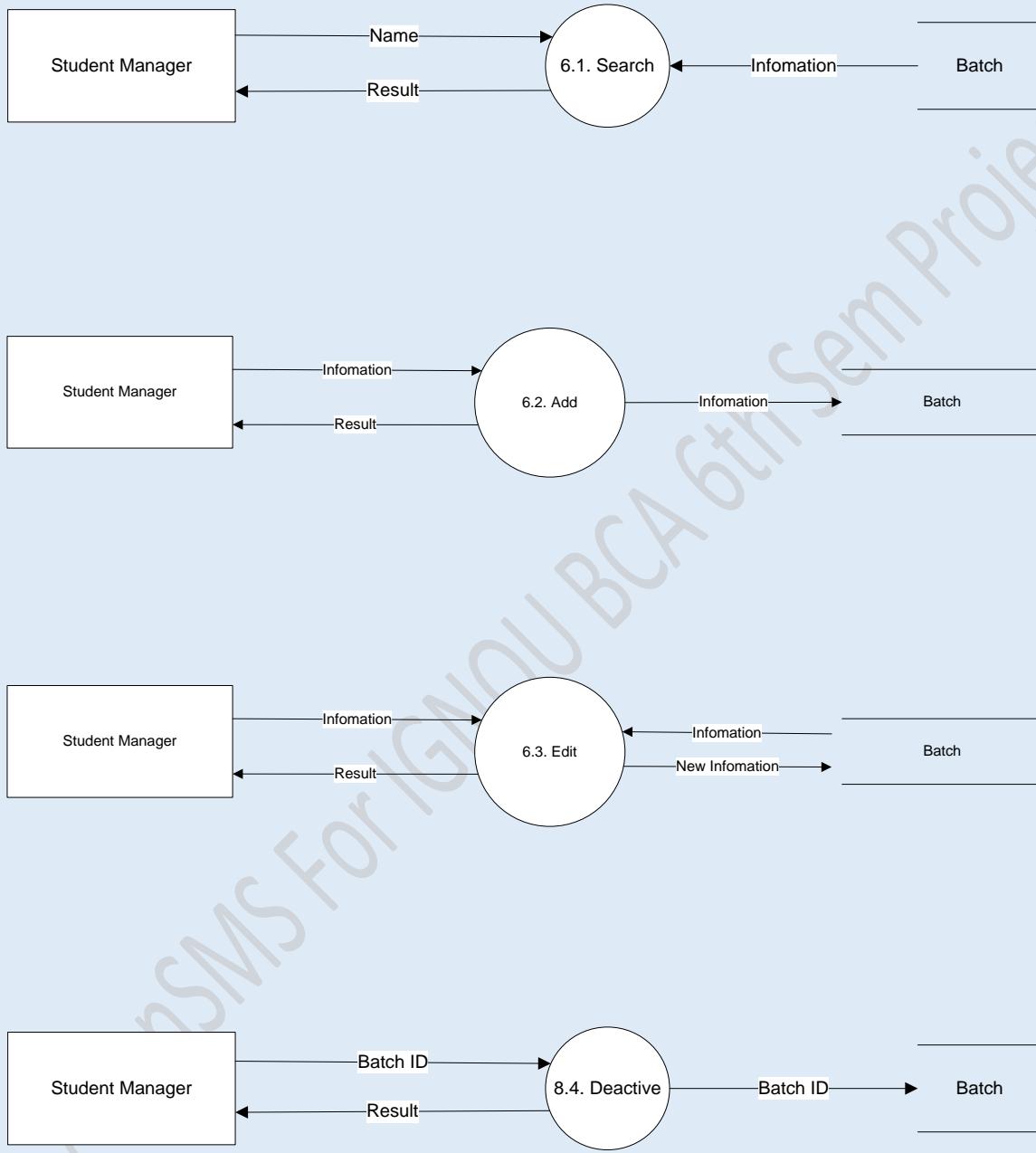
Level 2.5. Manage Staff

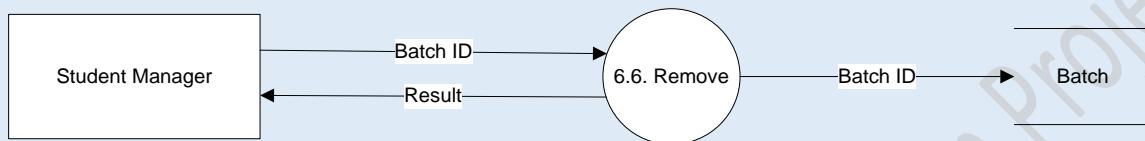
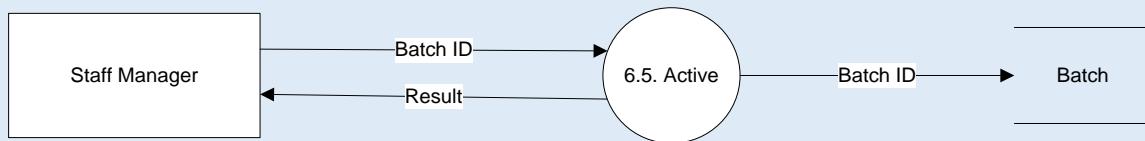




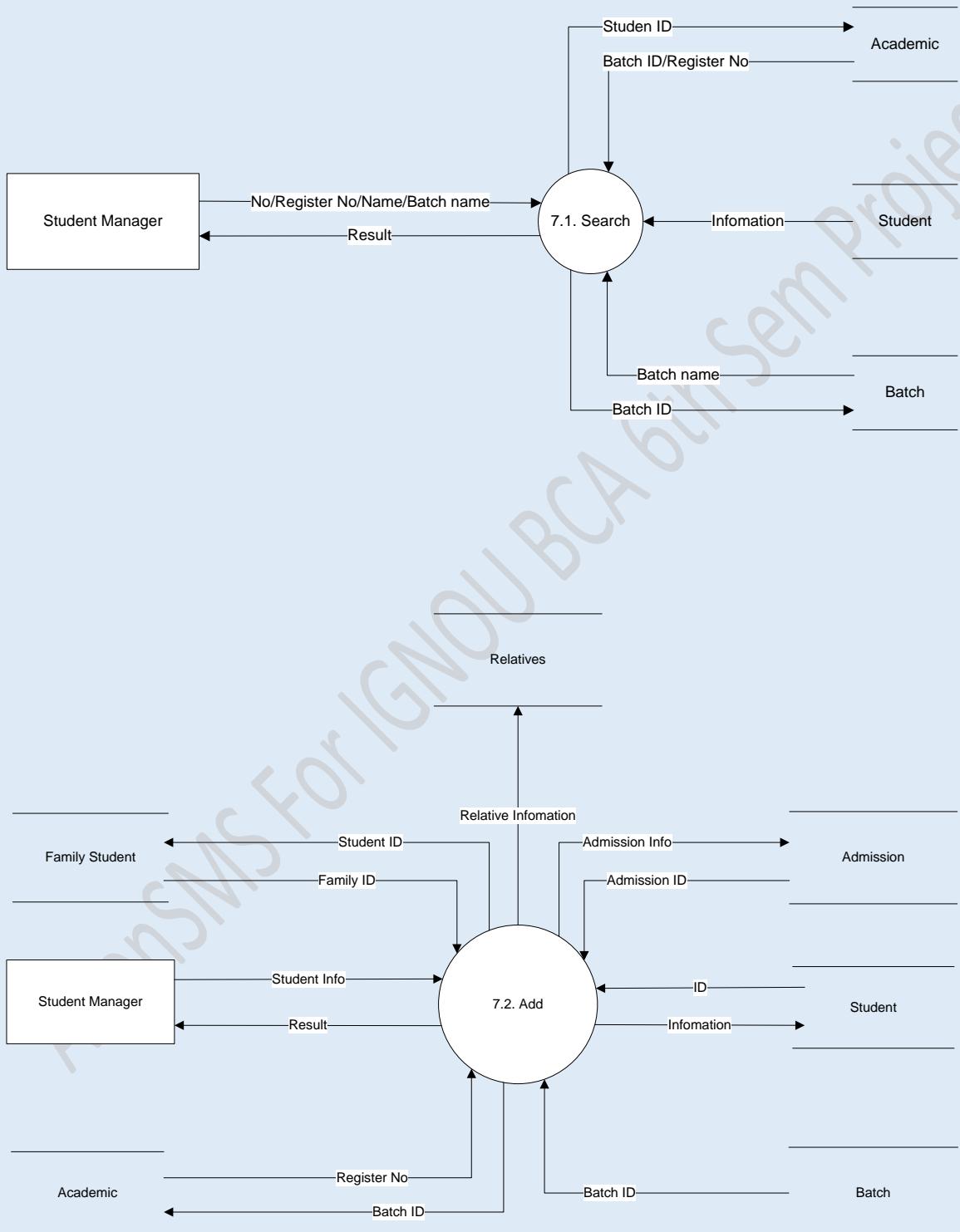


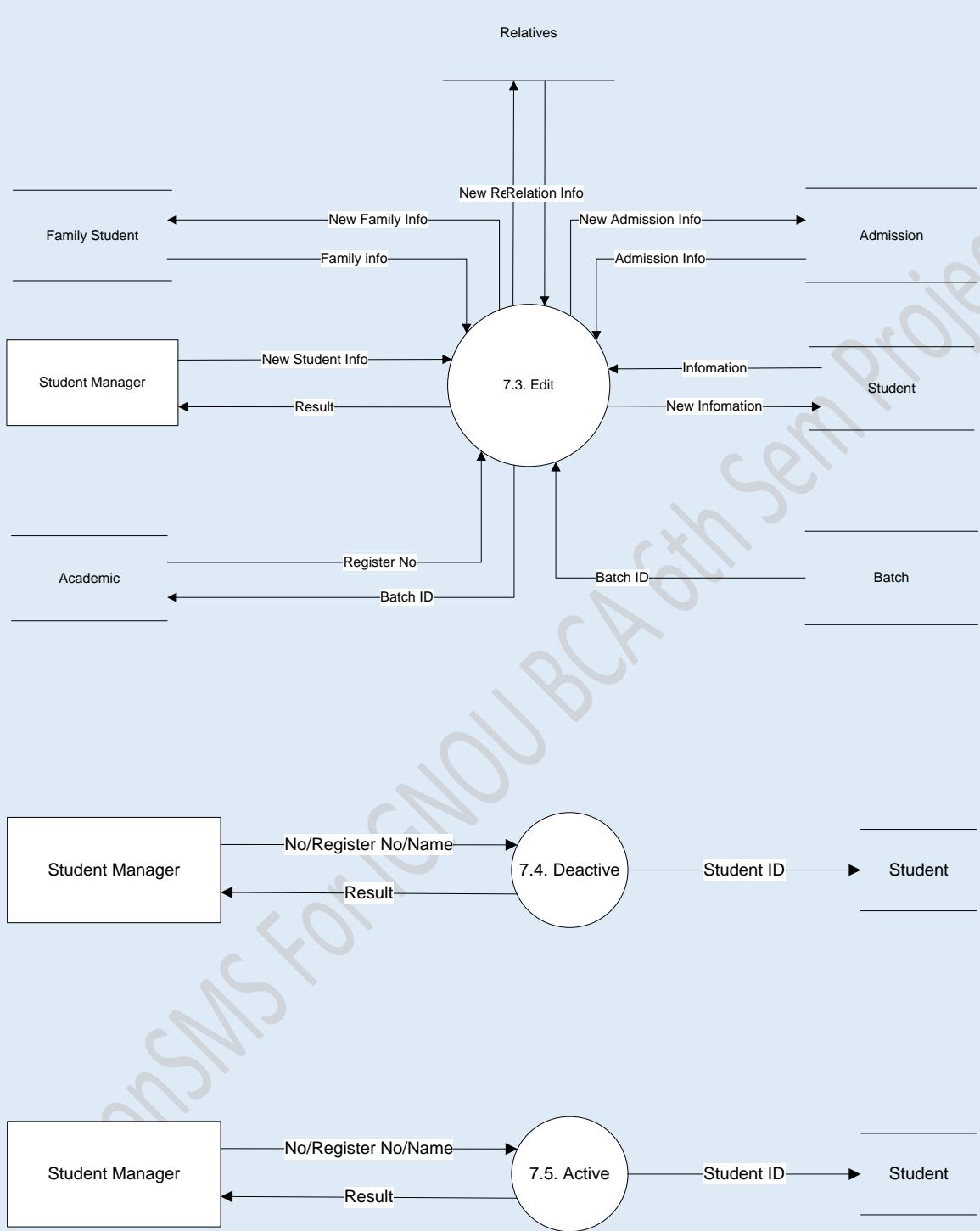
Level 2.6. Manage Batch

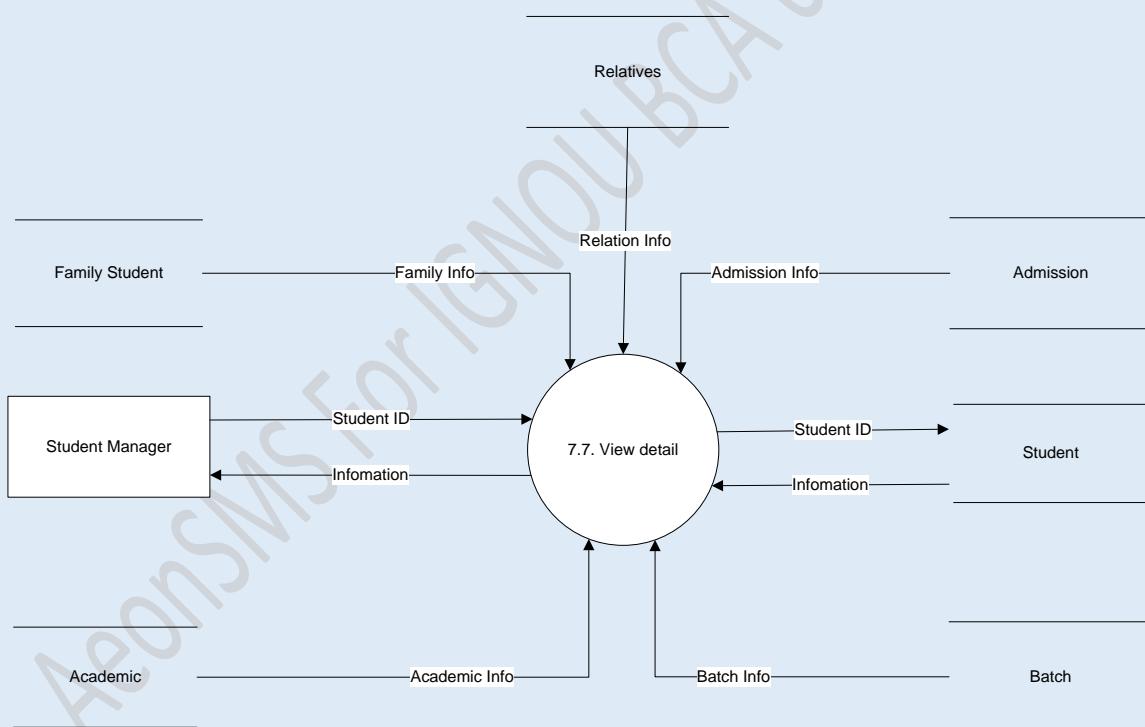
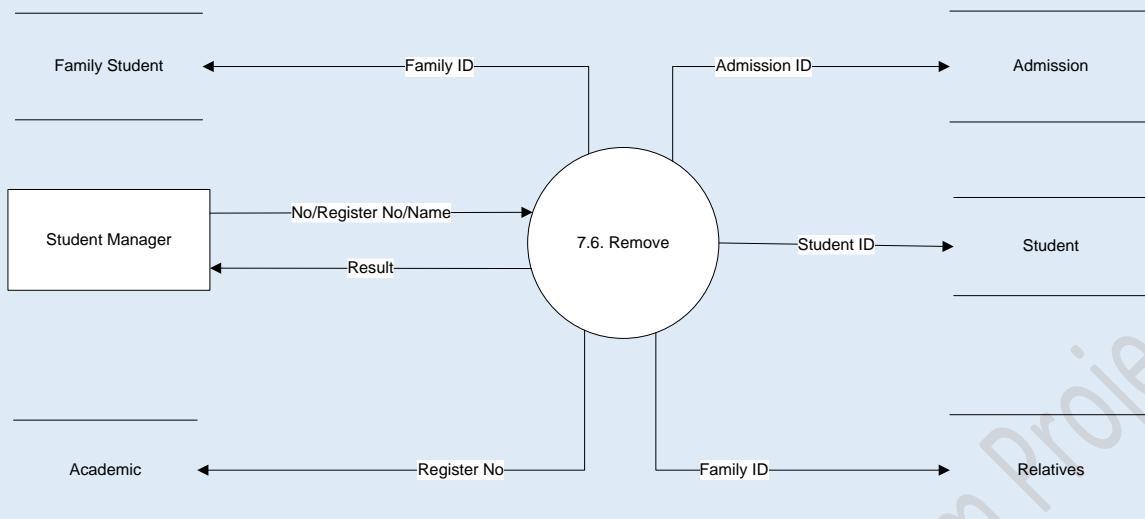




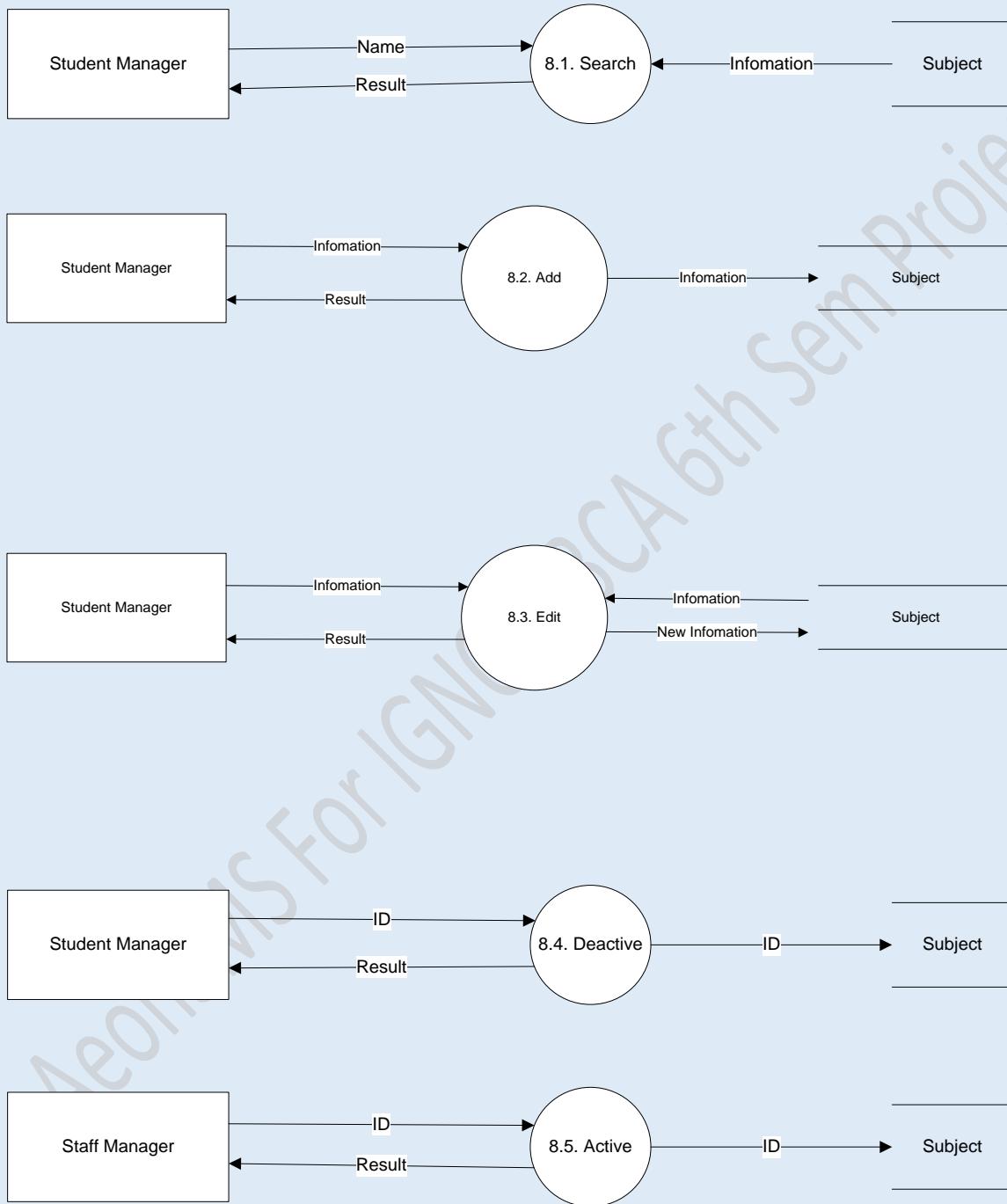
Level 2.7. Manage Student

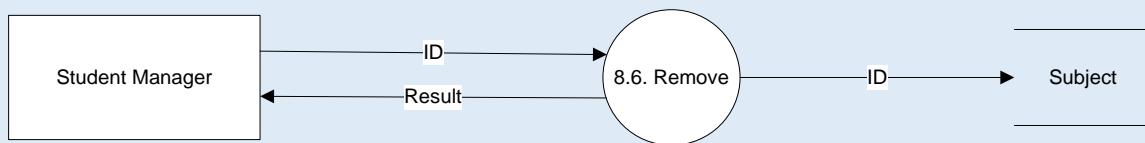




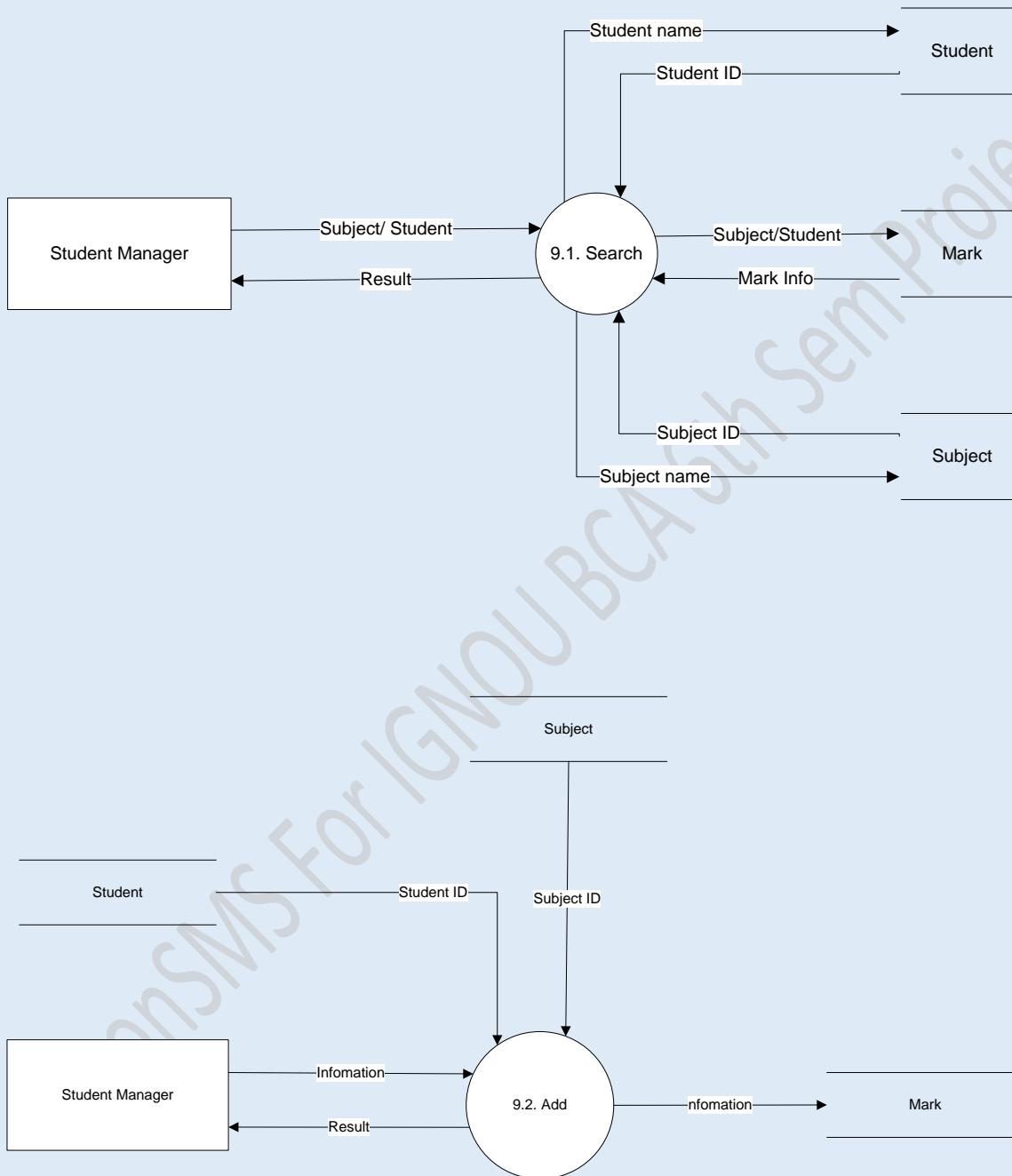


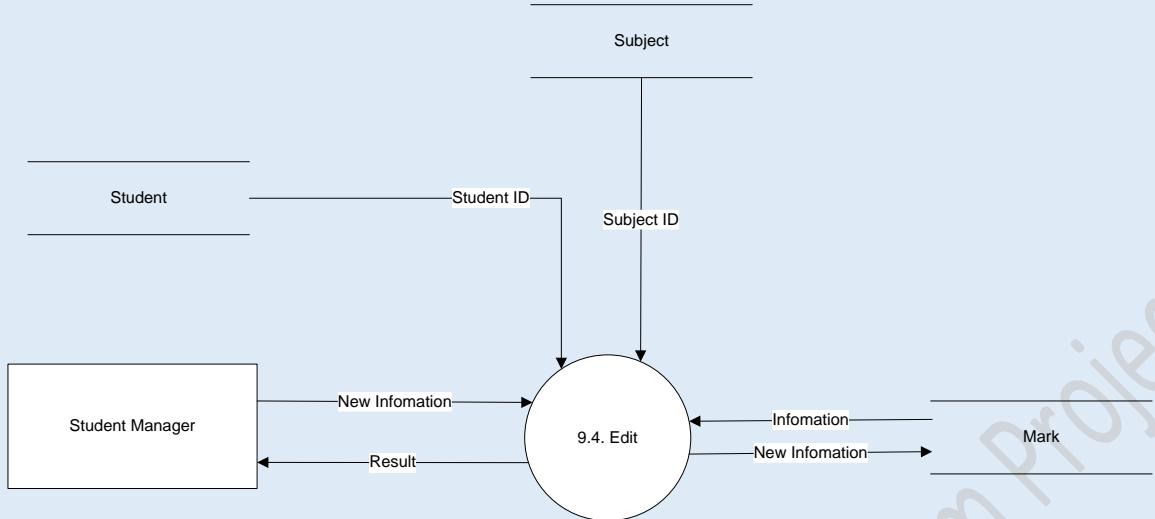
Level 2.8. Manage Subject



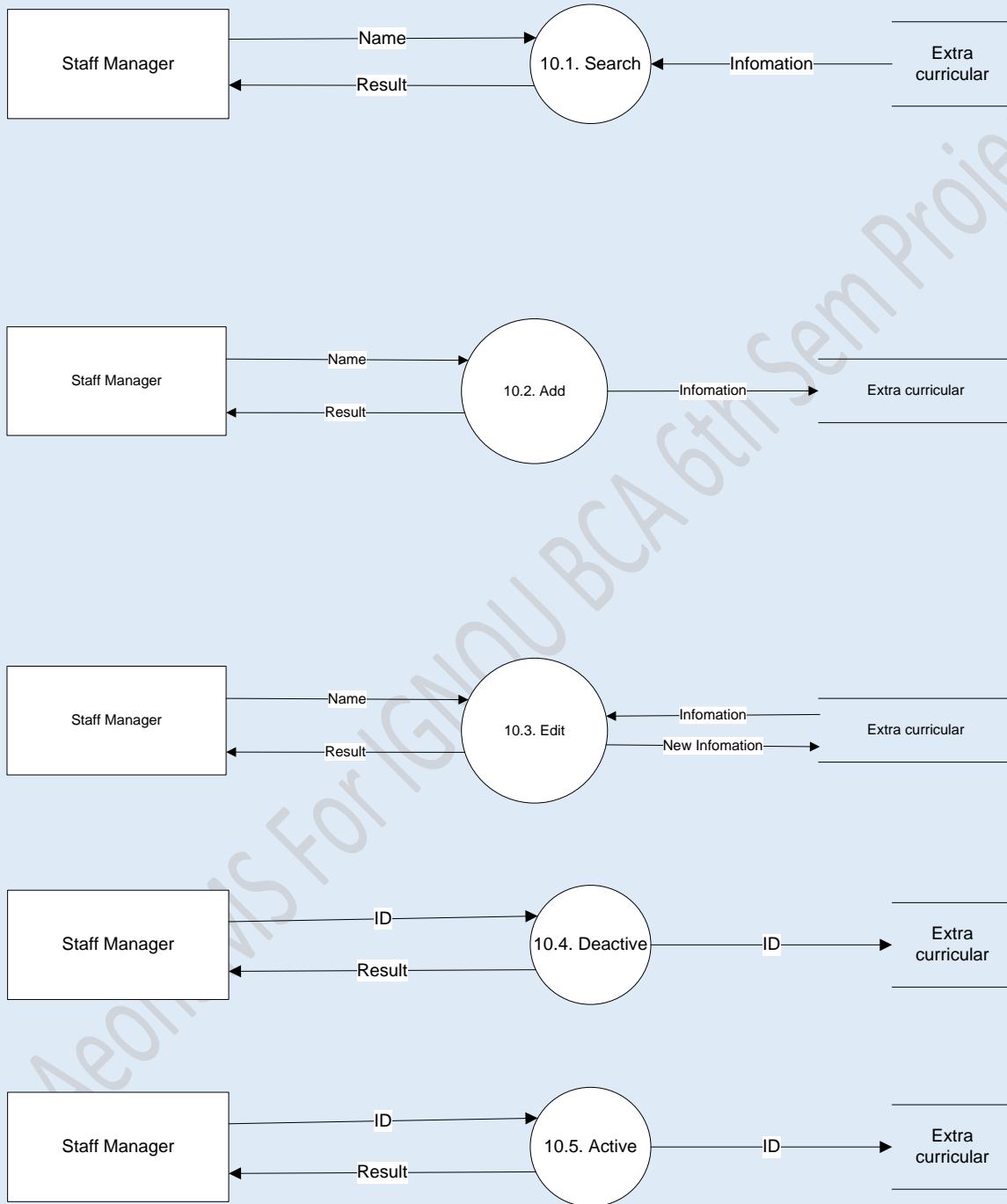


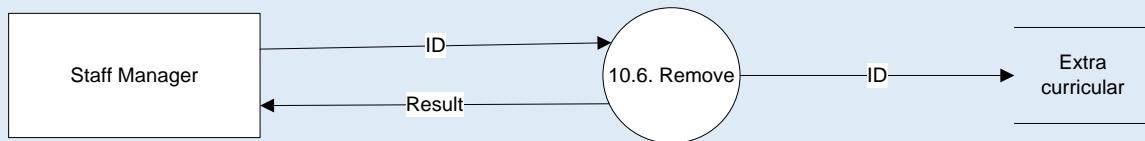
Level 2.9. Manage Mark





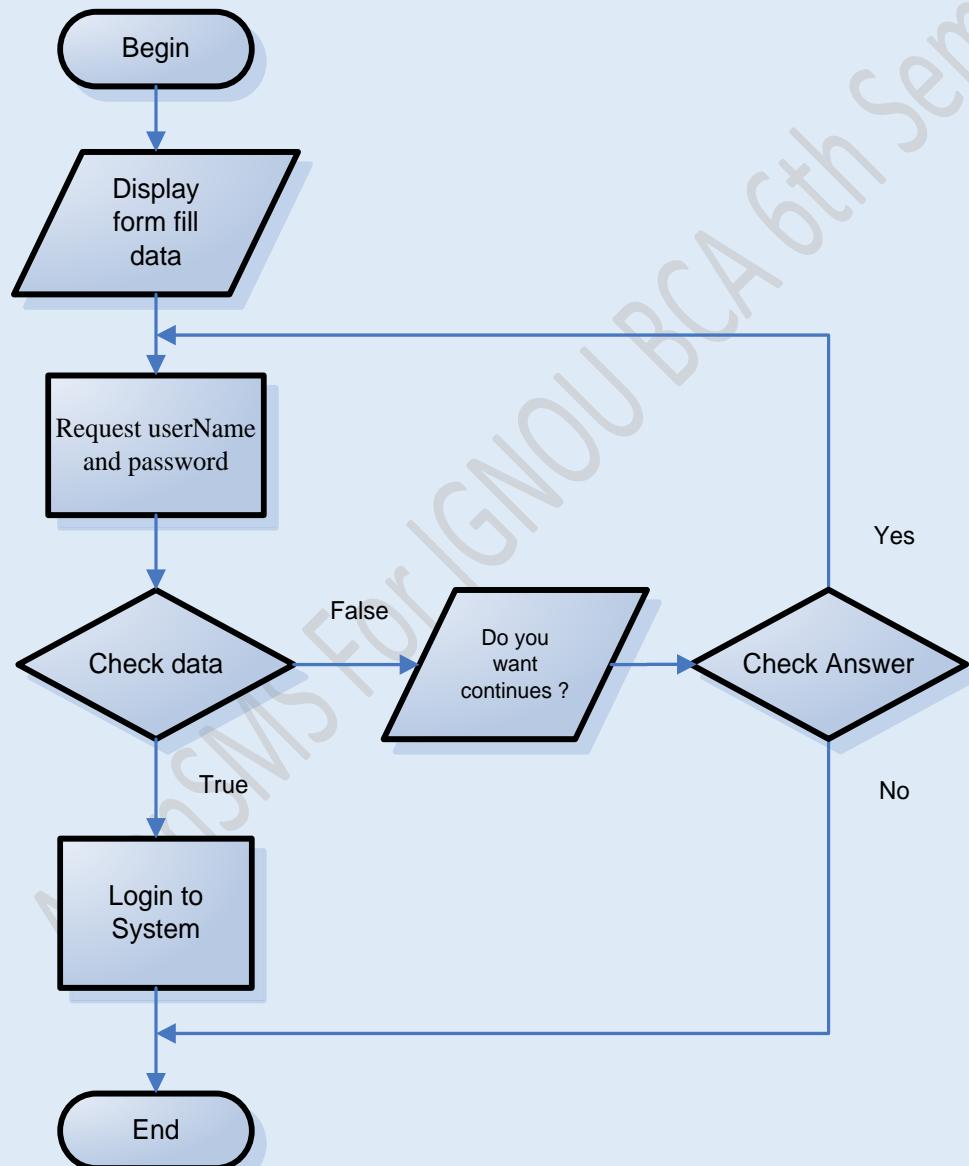
Level 2.10. Manage Extra Curricular



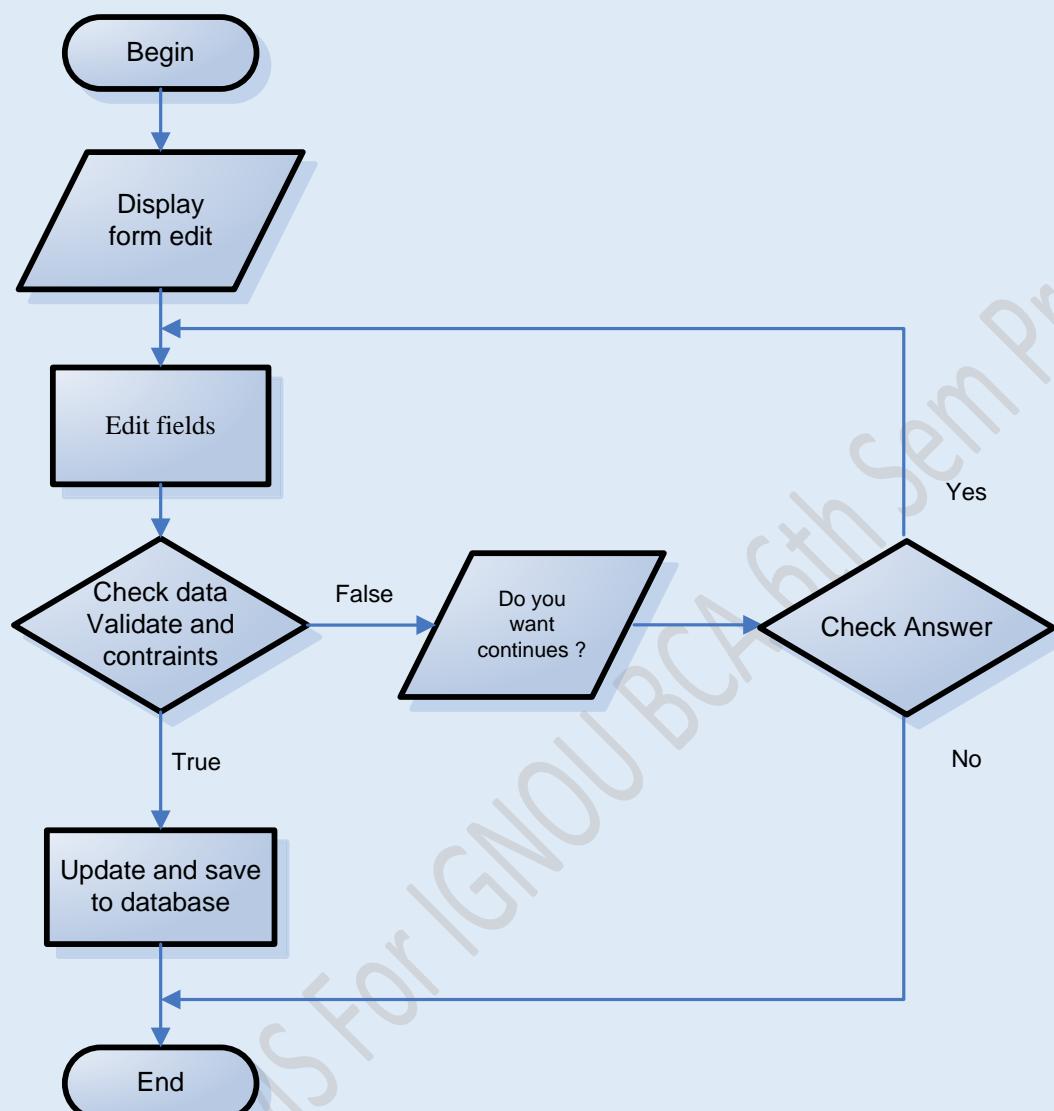


3.2. Flow chart

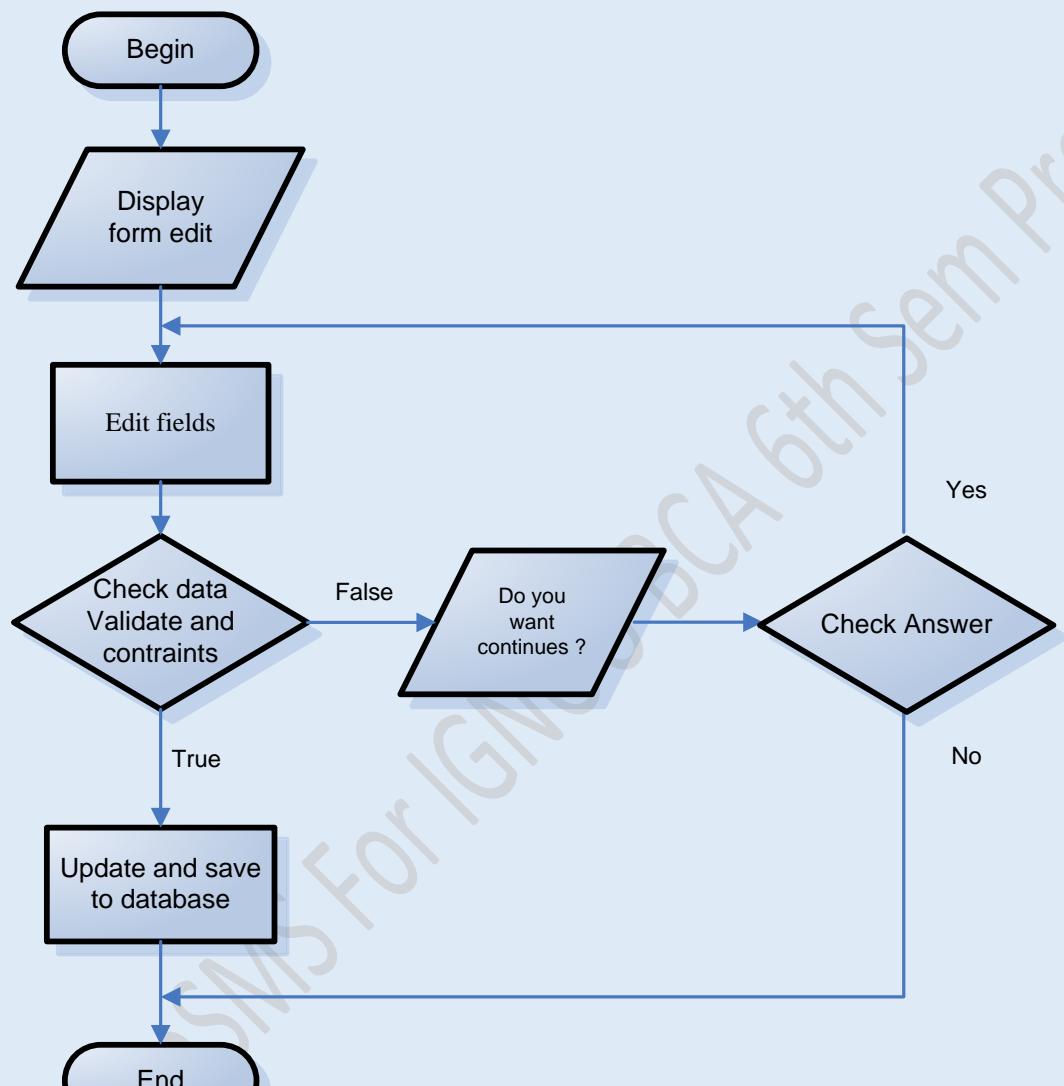
3.2.1. Login



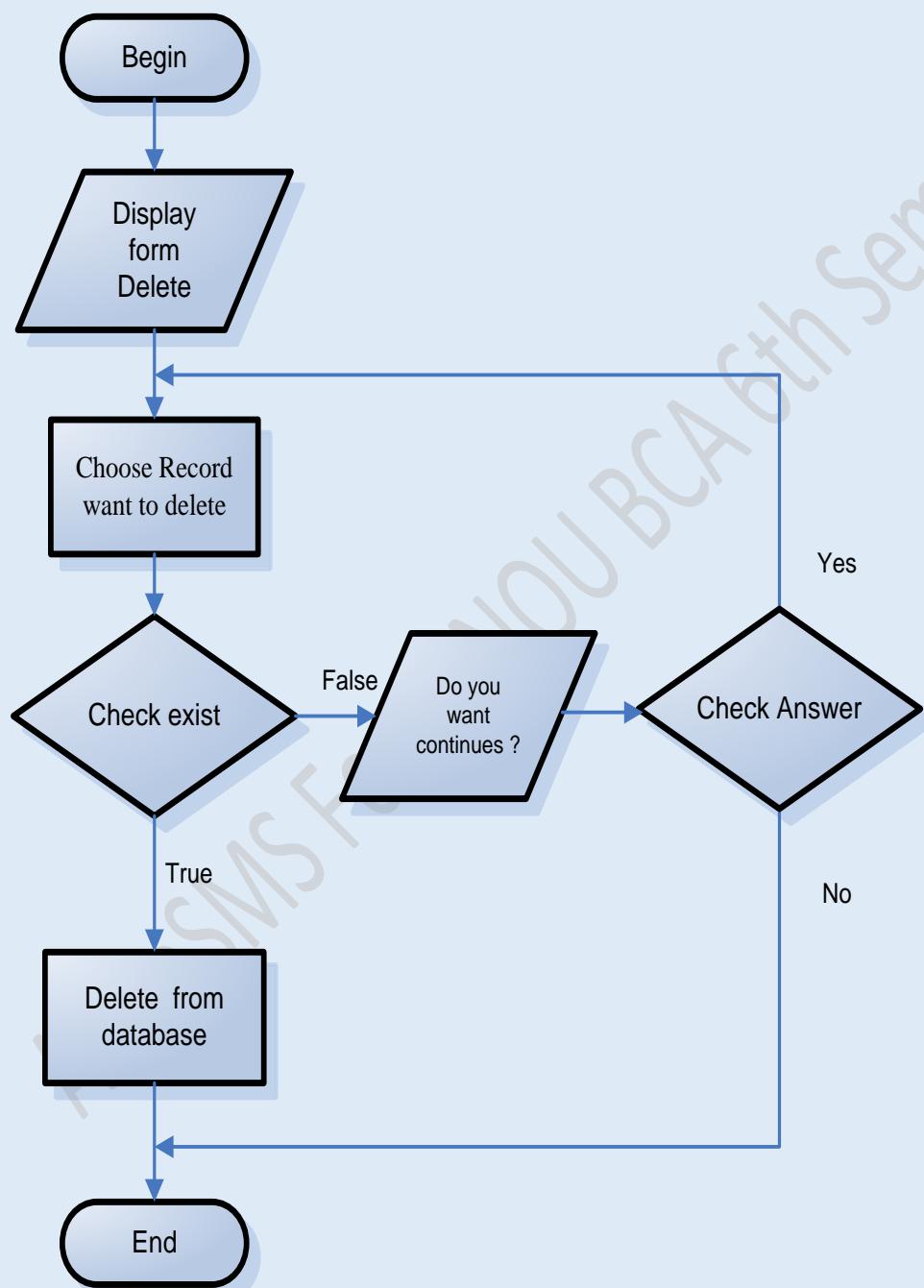
3.2.2. Add Records



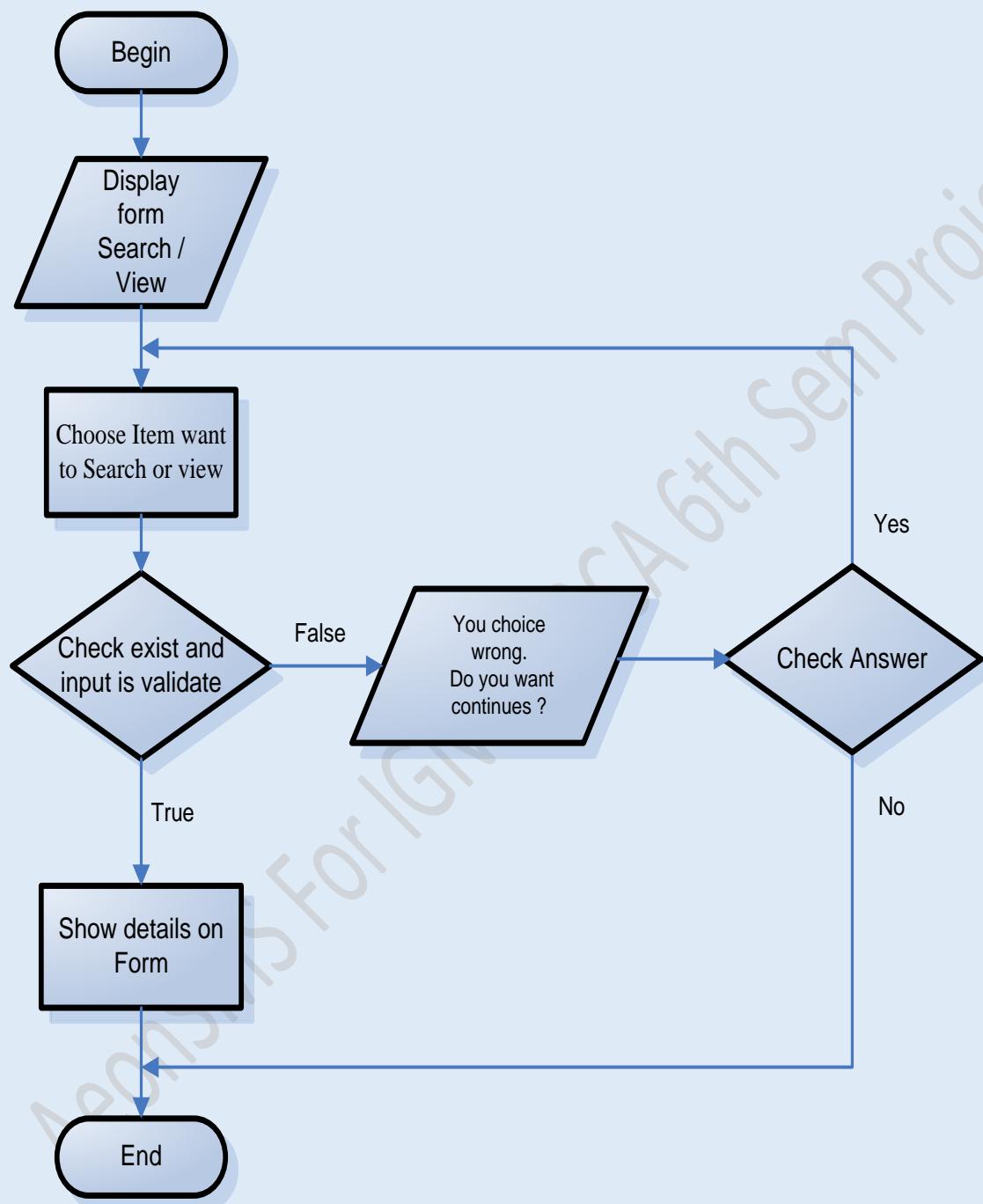
3.2.3. Edit Records



3.2.4. Remove Records



3.2.5. Search and View Records



3.3. Entity Relationship Diagram

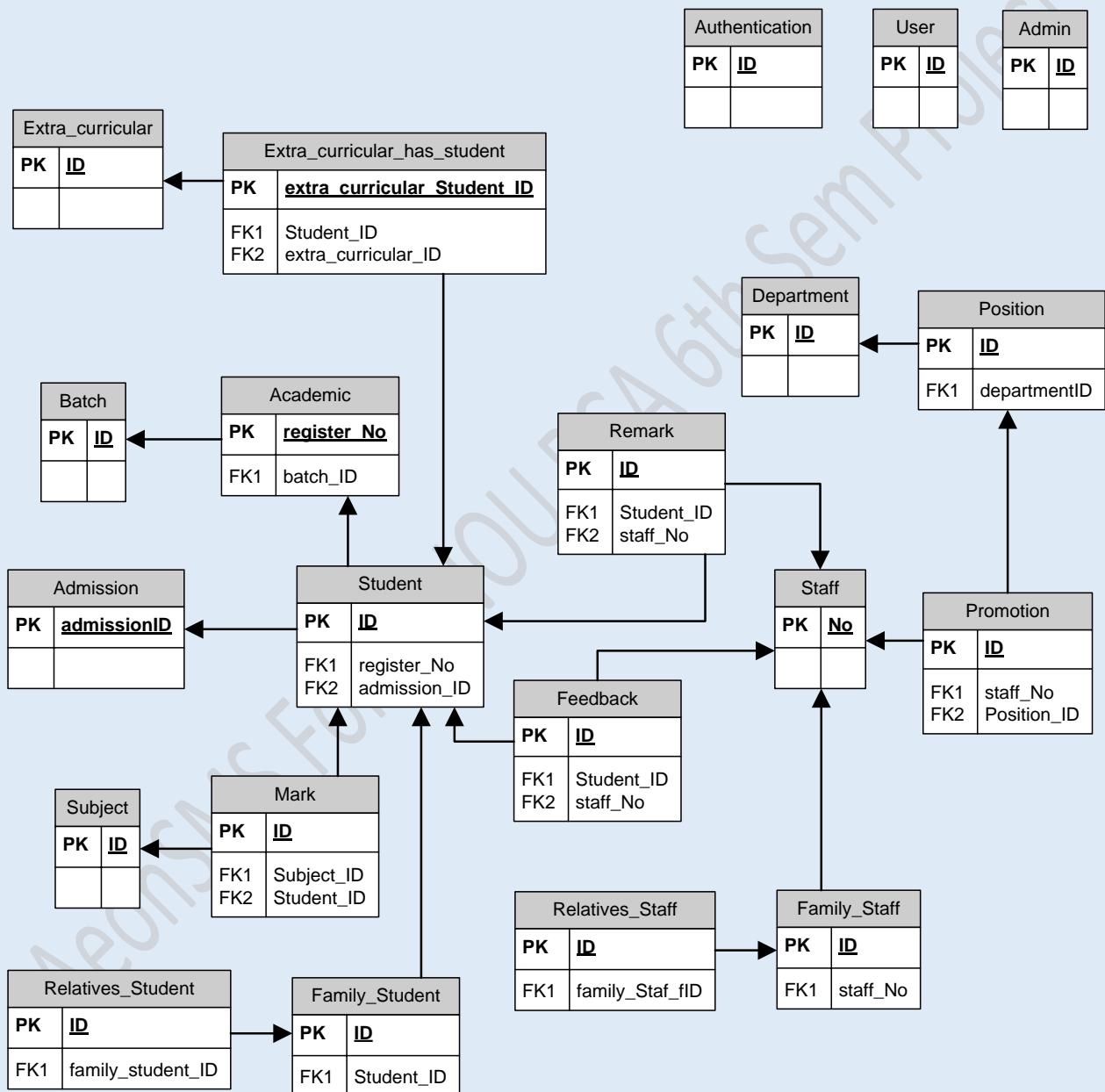
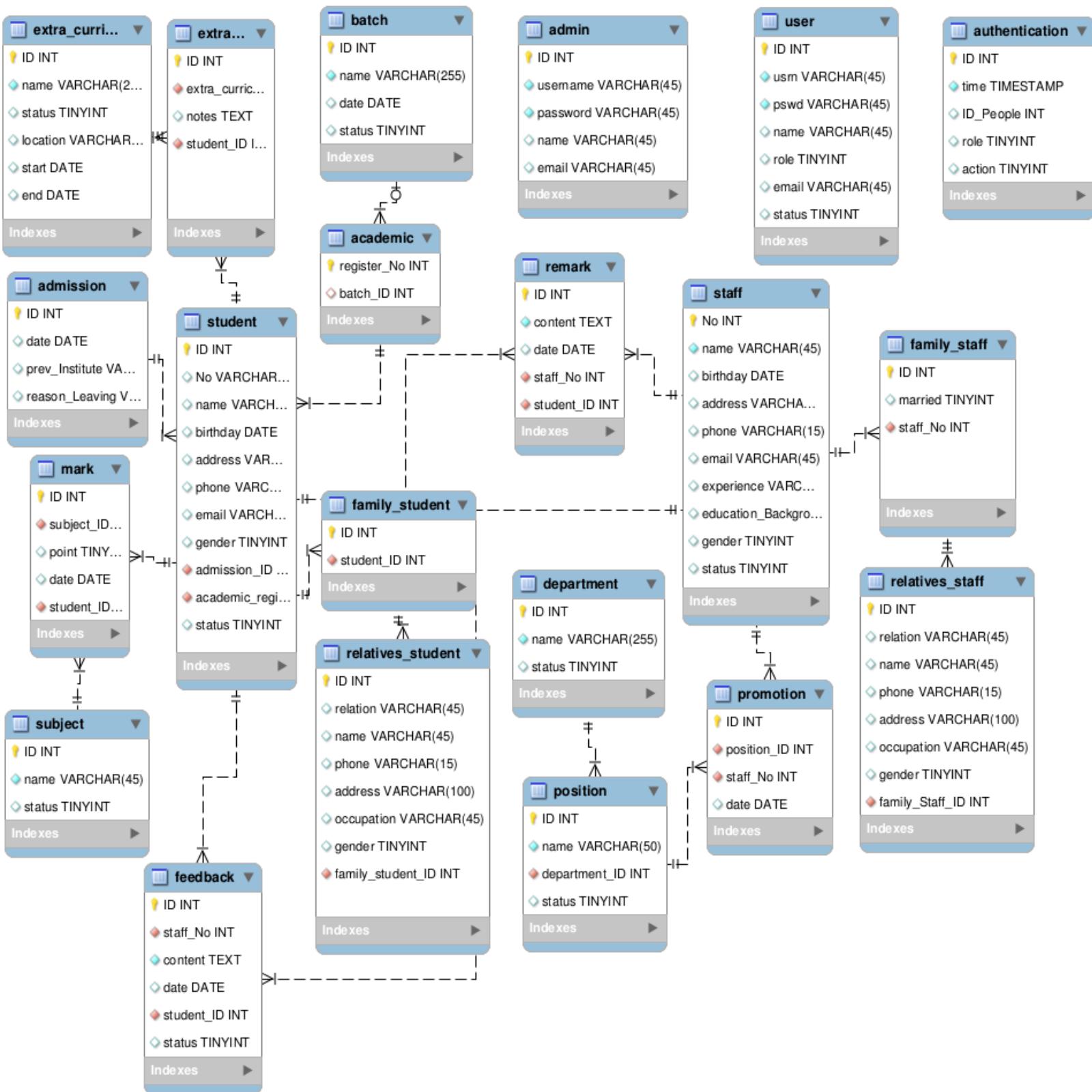


Table Diagram



3.4. Database structure

3.4.1. Table Admin

Field Name	Type	Null	Key	Description
ID	Int	No	PK	Stores Admin ID
Username	Varchar(50)	No		Stores Username of Admin
Password	Varchar(50)	No		Stores Password of Admin
Name	Varchar(50)	Yes		Stores Name of Admin
Email	Varchar(50)	Yes		Stores Email of Admin

3.4.2. Table User

Field Name	Type	Null	Key	Description
ID	Int	No	PK	Stores User ID
Usrn	Varchar(50)	No		Stores Username of User
Pswd	Varchar(50)	No		Stores Password of User
Name	Varchar(50)	Yes		Stores Name of User
Role	Tinyint	No		Stores Role of User. Value 1 for Staff Manager, and 0 for Student Manager
Email	Varchar(50)	Yes		Stores Email of User
Status	Tinyint	No		Stores Status of User

3.4.3. Table Authentication

Field Name	Type	Null	Key	Description
ID	Int	No	PK	Stores Authentication ID
Time	Timestamp	No		Stores Login/Logout Time
ID_People	Int	No		Stores ID of User/Admin login or logout
Role	Tinyint	No		Stores Role of User/Admin. Value 0 for Admin and value 1 for User
type	Tinyint	No		Stores Action of User/Admin. Value 0 for Login and value 1 for Logout

3.4.4. Table Staff

Field Name	Type	Null	Key	Description
No	Int	No	PK	Stores Employee No of Staff
Name	Varchar(50)	Yes		Stores Name of Staff
Birthday	Date	Yes		Stores Birthday of Staff
Address	Text	Yes		Stores Address of Staff
Phone	Varchar(20)	Yes		Stores Phone of Staff
Email	Varchar(50)	Yes		Stores Email of Staff
Experience	Text	Yes		Stores Experience of Staff
Education	Text	Yes		Stores Education background of Staff
Gender	Tinyint	No		Stores Gender of Staff. Value 0 for Female, value 1 for Male and -1 for Other
Status	Tinyint	No		Stores Status of Staff. Value 1 for Active and 0 for Deactive

3.4.5. Department

Field Name	Type	Null	Key	Description
ID	Int	No	PK	Stores Department ID
Name	Varchar(50)	No		Stores Name of Department
Status	Tinyint	No		Stores Status of Department. Value 1 for Active and 0 for Deactive

3.4.6. Table Position

Field Name	Type	Null	Key	Description
ID	Int	No	PK	Stores Position ID
Name	Varchar(50)	No		Stores Name of Position
Department_ID	Int	No	FK	Stores Department ID
Status	Tinyint	No		Stores Status of Position. Value 1 for Active and 0 for Deactive

3.4.7. Promotion

Field Name	Type	Null	Key	Description
ID	Int	No	PK	Stores Promotion ID
Position_ID	Int	No	FK	Stores Position ID
Staff_No	Int	No	FK	Stores Staff No
Date	Date	Yes		Stores Date of Promotion

3.4.8. Relatives staff

Field Name	Type	Null	Key	Description
ID	Int	No	PK	Stores Relatives ID
Relation	Varchar(50)	Yes		Stores Relation of Relative
Name	Varchar(50)	Yes		Stores Name of Relative
Phone	Varchar(20)	Yes		Stores Phone of Relative
Address	Text	Yes		Stores Address of Relative
Occupation	Varchar(50)	Yes		Stores Occupation of Relative
Gender	Tinyint	No		Stores Gender of Relative
Family_staff_ID	Int	No	FK	Stores Family Staff ID

3.4.9. Family Staff

Field Name	Type	Null	Key	Description
ID	Int	No		Stores Family Staff ID
Married	Tinyint	No		Married of Staff. Value 1 for Married and value 0 for Single
Staff_ID	Int	No	FK	Stores Staff ID

3.4.10. Student

Field Name	Type	Null	Key	Description
ID	Int	No	PK	Stores Student ID
Register_No	Int	No	FK	Stores Register No of Student
Admission	Int	No	FK	Stores Admission ID
No	Varchar(10)	No		Stores Student No
Name	Varchar(50)	No		Stores Name of Student
Birthday	Date	Yes		Stores Birthda of Student
Address	Text	Yes		Stores Address of Student
Phone	Varchar(20)	Yes		Stores Phone of Student
Email	Varchar(50)	Yes		Stores Email of Student
Gender	Tinyint	No		Stores Gender of Student. Value 0 for Female, value 1 for Male and -1 for Other
Status	Tinyint	No		Stores Status of Student. Value 1 for Active and 0 for Deactive

3.4.11. Admission

Field Name	Type	Null	Key	Description
ID	Int	No	PK	Stores Admission ID
Date	Date	Yes		Stores admit Date
Prev_Institute	Varchar(50)	Yes		Stores Previous Institute
Reason_Leaving	Text	Yes		Stores Reason Leaving

3.4.12. Academic

Field Name	Type	Null	Key	Description
Register_No	Int	No	PK	Stores Register No
Batch_ID	Int	No	FK	Stores Batch ID

3.4.13. Batch

Field Name	Type	Null	Key	Description
ID	Int	No	PK	Stores Batch ID
Name	Varchar(50)	No		Stores Name of Batch
Date	Date	Yes		Stores Date create Batch
Status	Tinyint	No		Stores Status of Batch. Value 1 for Active and 0 for Deactive

3.4.14. Subject

Field Name	Type	Null	Key	Description
ID	Int	No	PK	Stores Subject ID
Name	Varchar(50)	No		Stores Name of Subject
Status	Tinyint	No		Stores Status of Position. Value 1 for Active and 0 for Deactive

3.4.15. Mark

Field Name	Type	Null	Key	Description
<u>ID</u>	Int	No	PK	Stores Mark ID
Student_ID	Int	No	FK	Stores Student ID
Subject_ID	Int	No	FK	Stores Subject ID
Point	Tinyint	Yes		Stores Point of Mark
Date	Date	Yes		Stores Date

3.4.16. Extra curricular

Field Name	Type	Null	Key	Description
ID	Int	No	PK	Stores Extra curricular ID
Name	Varchar(50)	No		Stores Name of Extra curricular
Status	Tinyint	No		Stores Status of Extra curricula. Value 1 for Active and 0 for Deactive
Location	Varchar(50)	Yes		Store Location of Extra curricular
Start	Date	Yes		Store Start Date of Extra curricular
End	Date	Yes		Store End Date of Extra curricular

3.4.17. Extra_curricular_has_Student

Field Name	Type	Null	Key	Description
ID	Int	No	PK	Stores ExCur _Student ID
Extra_curricular_ID	Int	No	FK	Stores Extra Curricular ID
Student_ID	Int	No	FK	Stores Student ID
Notes	Text	Yes		Notes

3.4.18. Remark

Field Name	Type	Null	Key	Description
ID	Int	No	PK	Stores Remark ID
Staff_ID	Int	No	FK	Stores Staff ID
Student_ID	Int	No	FK	Stores Student_ID
Content	Text	Yes		Stores Context of Remark
Date	Date	Yes		Stores Creating date Remark

3.4.19. Feedback

Field Name	Type	Null	Key	Description
<u>ID</u>	Int	No	PK	Stores Feedback ID
Staff_ID	Int	No	FK	Stores Staff ID
Student_ID	Int	No	FK	Stores Student_ID
Content	Text	Yes		Stores Context of Feedback
Date	Date	Yes		Stores Creating date Feedback
Status	Tinyint	No		Stores Status of Feedback. Value 1 processed and value 0 for unprocessed

3.4.20. Relatives student

Field Name	Type	Null	Key	Description
ID	Int	No	PK	Stores Relatives ID
Relation	Varchar(50)	Yes		Stores Relation of Relative
Name	Varchar(50)	Yes		Stores Name of Relative
Phone	Varchar(20)	Yes		Stores Phone of Relative
Address	Text	Yes		Stores Address of Relative
Occupation	Varchar(50)	Yes		Stores Occupation of Relative
Gender	Tinyint	No		Stores Gender of Relative
Family_Student_ID	Int	No	FK	Stores Family Student ID

3.4.21. Family Student

Field Name	Type	Null	Key	Description
ID	Int	No	PK	Stores Family Student ID
Student_ID	Int	No	FK	Stores Student ID

Document WorkFlow
for
School Management System

Submitted by –

Ankit Raj
159802231
BCA 6th Semester
Session 2015-2018

SCHOOL OF COMPUTER AND INFORMATION SCIENCE



3.5. Screen design/Document Workflow

1. Splash Screen

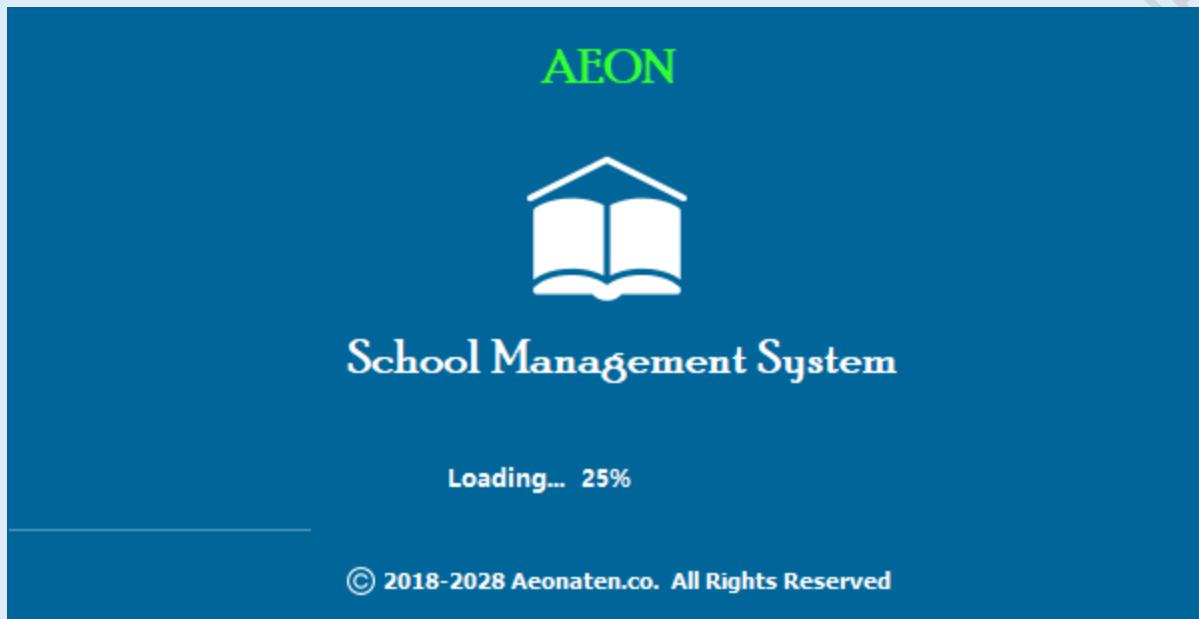


Figure 1.0: Startup Splash Screen

Process and Importance of Splash Screen in AeonSMS.

The splash screen or launcher image is here to give the user immediate feedback that application has been started and is loading, the splash image is there to improve the UX by simulating faster loading time

2. Database Setting

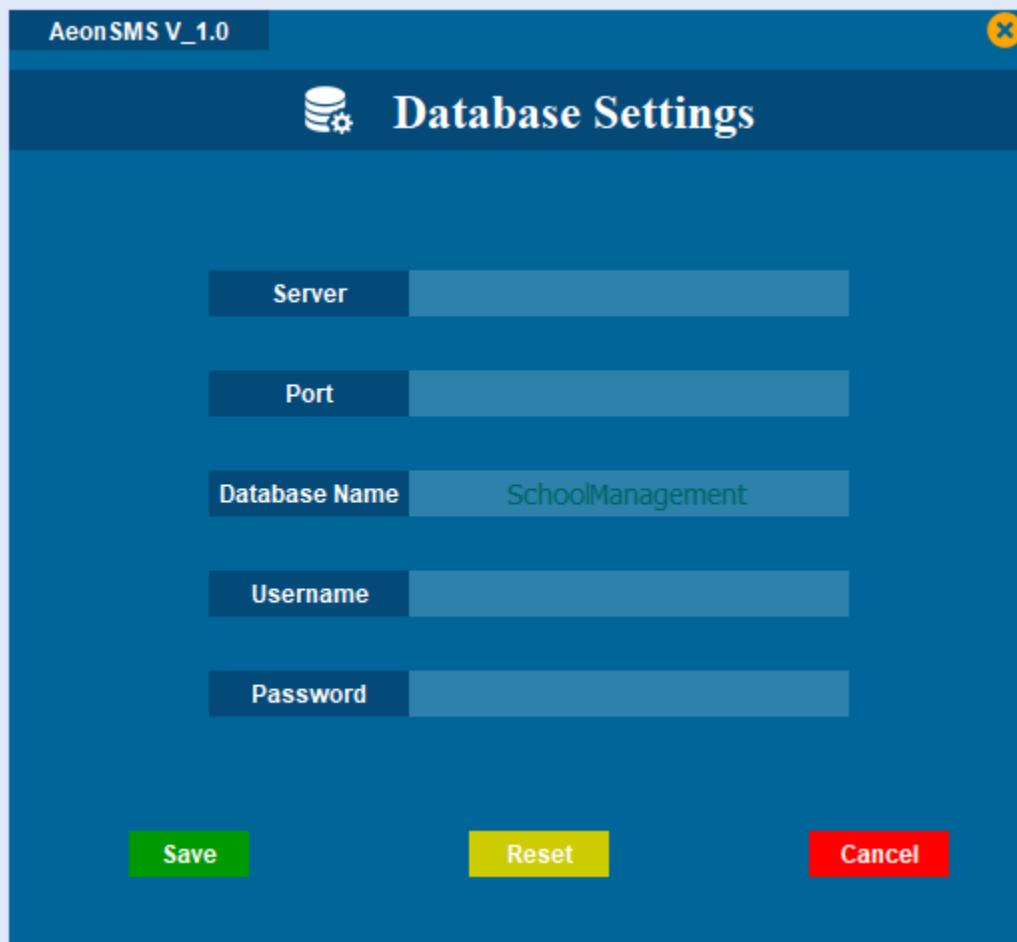


Figure 2.0: Database Setup

Process And Importance of Database Setting Form of AeonSMS

Database setting is provided for the portability to the different system, as project is environment dependent (being a Database restricted system, dependent on MySQL) and as different systems may be using standard database setup or some may also use customized port number and server address for the MySQL Installation .

To overcome incompatibility and conflict between system, I have created this Database Setup form.

Fields

1. Server field – user need to enter server name for the MySQL connection.
2. Port field – User need to enter the port number for the installation of MySQL.
3. Database Name – it is predefined for the application and database in the MySQL Database System should have same be in existence, if not please import query.sql file provided in the project folder.
4. Username – name of the connection to under which above concerned database exists
5. Password – Password for the same connection under which the above concerned Db Exists.

3. Login

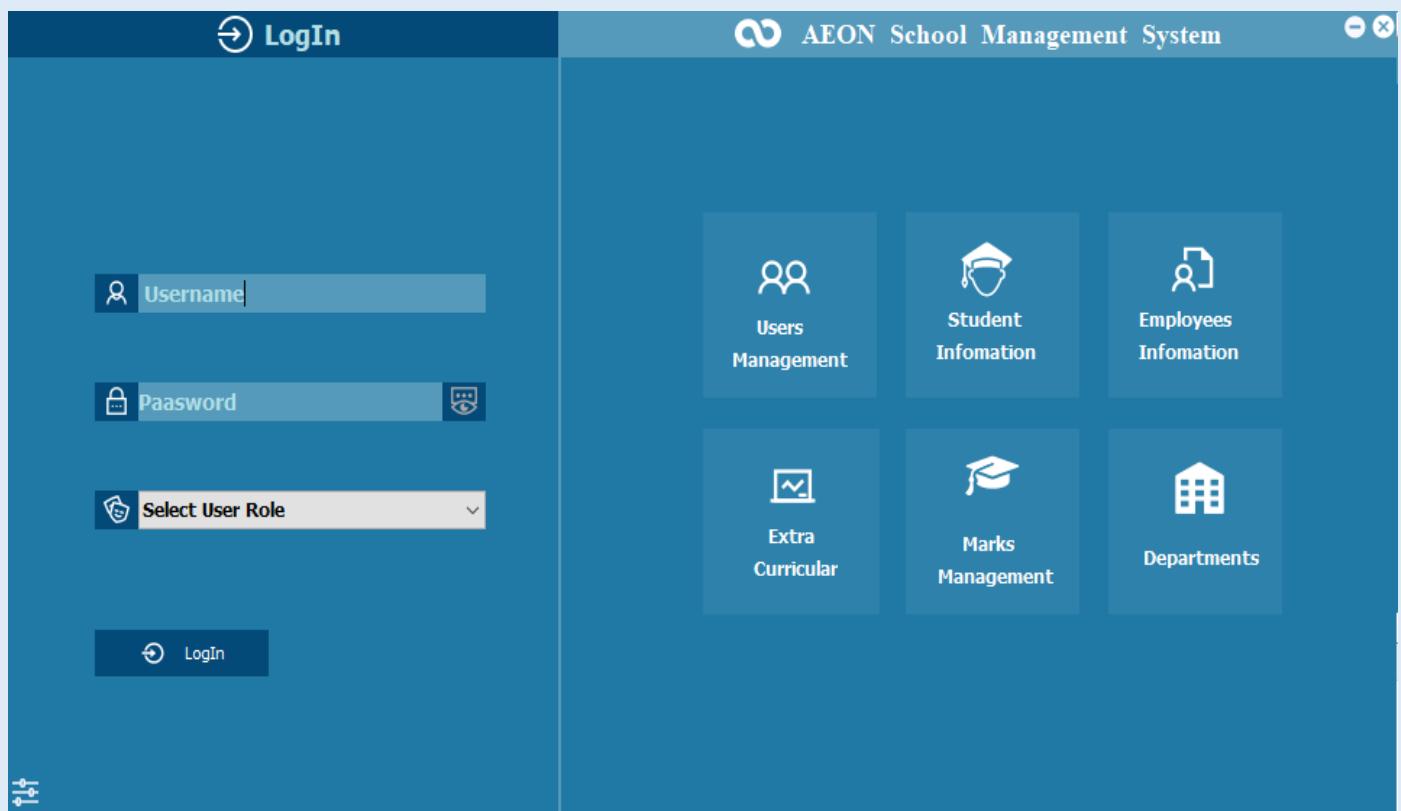


Figure 3.0. Login Form

Process and importance of Login form in the AeonMS

Login form act as the gatekeeper to the School management system preventing the unauthorized access to the application and Schools sensitive database.

Fields

1. Username – Username of the user who is intending to access the system and database of school management system.
2. Password – corresponding password for the user.

3. User Role – there are primarily two type of user who can access the access
The school management system

A. Administrator.

B. Manager

Administrator ::

This user is the system root user having access to all the basic features along with advance administrative features which includes adding new users, deactivating a existing user, removing a existing user, updating the information of existing user and keeping the history of all existing user.

Manager User ::

Manager is a normal user who can access the info's of student and employee of the school.

To maintain and specialized the complexity of the system there is two type of Manager User in the system

- a. Student Manager- this user automates all the student related tasks such as addmission, Course,Class,Subject and marks related tasks.
- b. Employee Manager – This type of the user can automate all the employee related tasks Of the concerned school.

To maintain the Resources consistently AeonSMS Application provides different home page to the user depending on their user role.

3.1 Administrator Home

The screenshot shows the AeonSMS V_1.0 Administrative Manager interface. At the top, there is a header bar with the title "AeonSMS V_1.0 Administrative Manager", the user name "admin", and navigation links for "Change Password", "info", "Logout", and a close button.

The main content area is divided into three main sections: "Users Administration", "Student Management", and "Employee Management".

- Users Administration:**
 - Users Management (Icon: Two people)
 - Deactivated Users (Icon: Two people with a crossed-out symbol)
 - Database Setup (Icon: Database with gears)
- Student Management:**
 - Student Infomation (Icon: Student cap)
 - Suspended Student (Icon: Student cap with a crossed-out symbol)
 - Marks Management (Icon: Graduation cap)
 - Extra Curricular (Icon: Book with a checkmark)
 - Batch Management (Icon: Three people)
 - Activate Batch (Icon: Three people with a checkmark)
 - Course Management (Icon: Books)
 - Activate Course (Icon: Books with a checkmark)
- Employee Management:**
 - Employees Infomation (Icon: Person)
 - Job Position (Icon: Building)
 - Suspended Employee (Icon: Person with a crossed-out symbol)
 - Departments (Icon: Building)
 - Suspended Department (Icon: Building with a crossed-out symbol)

Administrator home provides home page for the administrative users
Have some advance options for administration of student,Employee
And Users.

1.2 Student Manager Home

AeonSMS V_1.0 Student Manager anakin Change Password info Logout

Student Manager

Available Employee Related Options

Student Infomation	Suspended Student	Class Management	Activate Class	Course Management
Activate Course	Curricular Management	Marks Management		

Welcome to Student Manager of Aeon School Management System

Using Student Manager of AeonSMS, you can see list of all enrolled student in your Institution as wel you can control addmission,updation of marks,slc of students and addition of class,batch and courses to your institute.

As AeonSMS is in its initial stage several functions like attendance management,id card generation system,examination management system,schedule management system will be added in the future releases.

Thank You for Evaluation Aeonsms
~Ankit RAJ(Developer)

1.3 Employee Manager Home

AeonSMS V.1.0 Employee Manager

ankitjha Change Password info Logout

Employee Manager

Available Employee Related Options

- Employees Infomation
- Suspended Employee
- Job Position
- Departments
- Suspended Department

Upcoming Employee Related Options

- Employees Attendance
- Employee Salary
- Employee Job Profile
- Generate ID card
- Leave History

Welcome to Employee Manager of Aeon School Management System

In Employee Manager of AeonSMS, you can see list of all working employee in your Institution in addition to employee details you can also add,delete and inactivate new departments and departmental posts.

As AeonSMS is developed to fulfill my graduation project assigment so it is limited in funtionalitity but, I am willing to add new features like employee payroll,employee attendance, leave,employee id card generation system,employee job profile,employee performance register and web integration in furthur releases.

Thank you for Evaluating AeonSMS

~Ankit RAJ(Developer)

4. Account information

The screenshot shows the 'Account Infomation' page of the AeonSMS V_1.0 Administrative Manager. The left sidebar contains links for Home, Users Management, Deactivate Users, and Database Settings. The top right features a user profile icon, the name 'admin', and navigation links for Change Password, info, Logout, and a help icon.

The main content area has a title 'User Details' with a person icon. It displays two rows of user information:

Username	admin
Email	ankit1110kr@gmail.com
Name	
Role	Admin

Below this is a section titled 'Action History' with a history icon. It lists a series of log entries:

Time	Action
2018-04-12 21:22:22.0	Login
2018-04-10 22:47:15.0	Logout
2018-04-10 22:47:05.0	Login
2018-04-10 22:09:09.0	Login
2018-04-10 22:07:56.0	Login
2018-04-10 22:06:56.0	Login
2018-04-10 22:05:50.0	Login
2018-04-10 22:04:57.0	Login
2018-04-10 22:00:36.0	Login
2018-04-10 21:58:57.0	Login
2018-04-10 21:58:04.0	Login
2018-04-10 21:57:12.0	Login
2018-04-10 21:55:39.0	Login

At the bottom are 'Save' and 'Reset' buttons.

Figure 2.1. Account Information

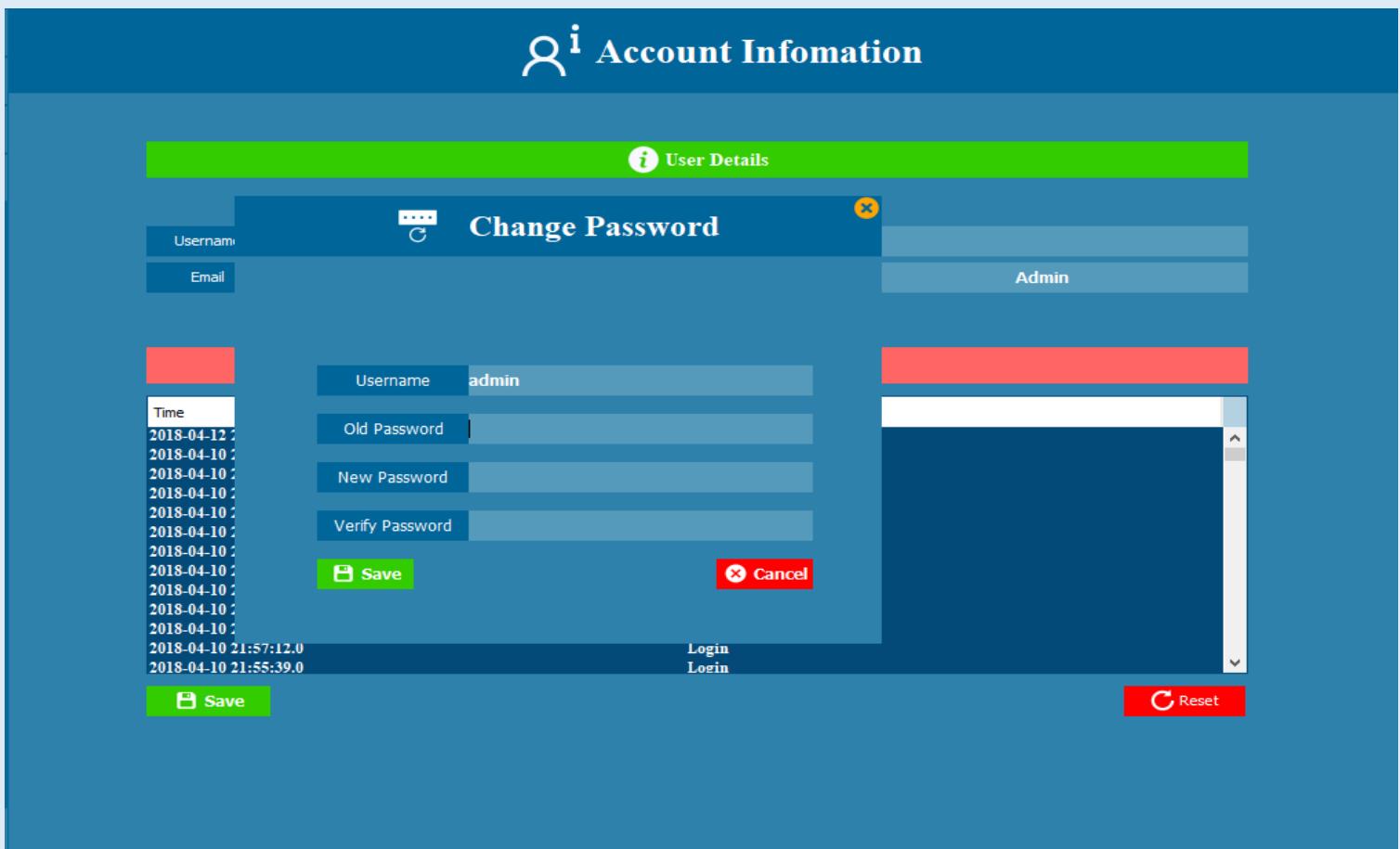


Figure 2.2. Change Password Screen

5. User Management Screen

The screenshot shows the 'Users Management' screen of the AeonSMS administrative interface. The top navigation bar includes links for Home, Users Management, Deactivate Users, Database Settings, Change Password, info, Logout, and a close button. The main content area features a search bar with fields for Username, Name, and Role (set to All), and buttons for Search and Reset. Below the search bar is a table titled 'All Users List' with columns for ID, Username, Password, Name, Role, and Email. The table contains five rows of user data. At the bottom of the page are three buttons: 'Add' (green), 'Edit' (pink), and 'Deactivate' (red).

ID	Username	Password	Name	Role	Email
1	ankit	password	ankit	Unknown	
2	ankitjha	jha	ankit	Staff manager	
3	anakin	ankit	anakin	Student manager	
4	1	1	0	Staff manager	
5	ank	11	ankit	Student manager	

Figure 3.1. User Management Screen

6. Staff Management Screen

The screenshot shows the 'Employee Manager' section of the AeonSMS administrative interface. The left sidebar contains links for Home, Employee Info, Inactive Employee, Department Info, Activate Departments, and Job Position. The main area is titled 'Employee Manager' and displays a table of employee data. The table has columns for No, Name, Phone, Email, Position, Department, and Gender. There are search and filter options at the top, and action buttons (Add, Edit, Deactivate, Export, Detail) at the bottom.

No	Name	Phone	Email	Position	Department	Gender
3	Ankit	988834566				Male
4	Ajit Kumar	9633323232	ajit011@gmail.com			Male
5	anfs					Male
6	new					Male

Figure 4.1. Employee Management Screen

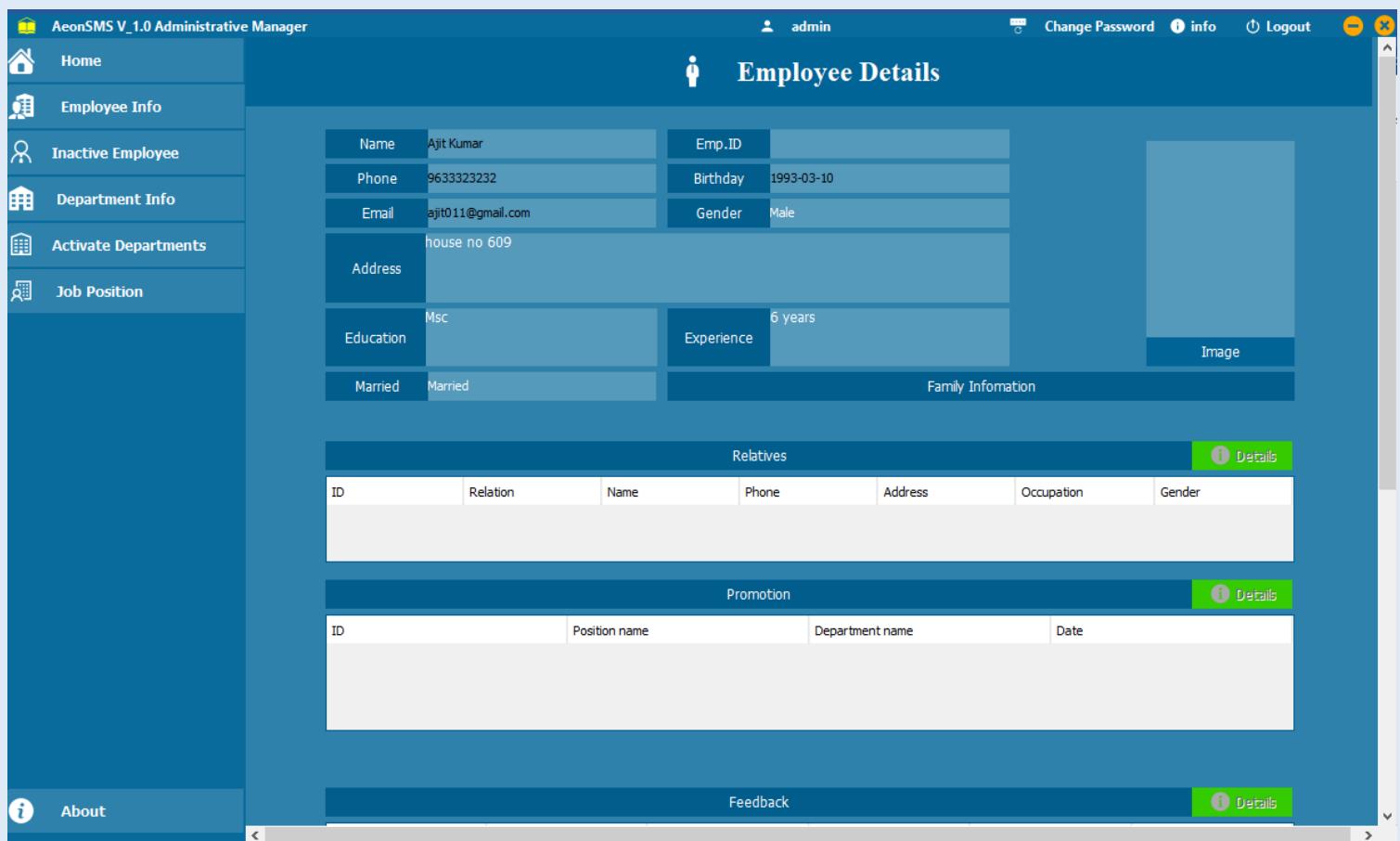


Figure 4.2. Employee detail Screen

AeonSMS For IGNOU

AeonSMS V_1.0 Administrative Manager admin Change Password info Logout X

Add New Employee

Back

Name	Emp.ID
Birthday	D.O.J
Gender	Married
Email	Phone
Education	Experience
Address	

photo

Family Information

ID	Relation	Name	Phone	Address	Occupation	Gender

Relatives

Promotion

ID	Position name	Department name	Time

About

Figure 4.3. Add Employee Screen

AeonSMS V_1.0 Administrative Manager admin Change Password info Logout X

Home Employee Info Inactive Employee Department Info Activate Departments Job Position

Back Update Employee Details

Name: new | Emp.ID: photo

Birthday: D.O.J: Maried: Married

Gender: Male | Email: Phone: Experience:

Education: Address: Browse

Family Information Relatives + Edit Delete

ID	Relation	Name	Phone	Address	Occupation	Gender

Promotion + Edit Delete

ID	Position name	Department name	Time

About

Figure 4.4. Edit Staff Screen

7. Department Management Screen

AeonSMS V_1.0 Administrative Manager

admin

Change Password info Logout

Departments

Name

ID	Title	Room No	H.O.D
3	Teacher		
4	Head		
5	Maths		
6	Science		
7	Sport		
9	admin		
10	admins		
11	2nd		
12	newad		
13	3rd		

About

Figure 5.1. Department Management Screen

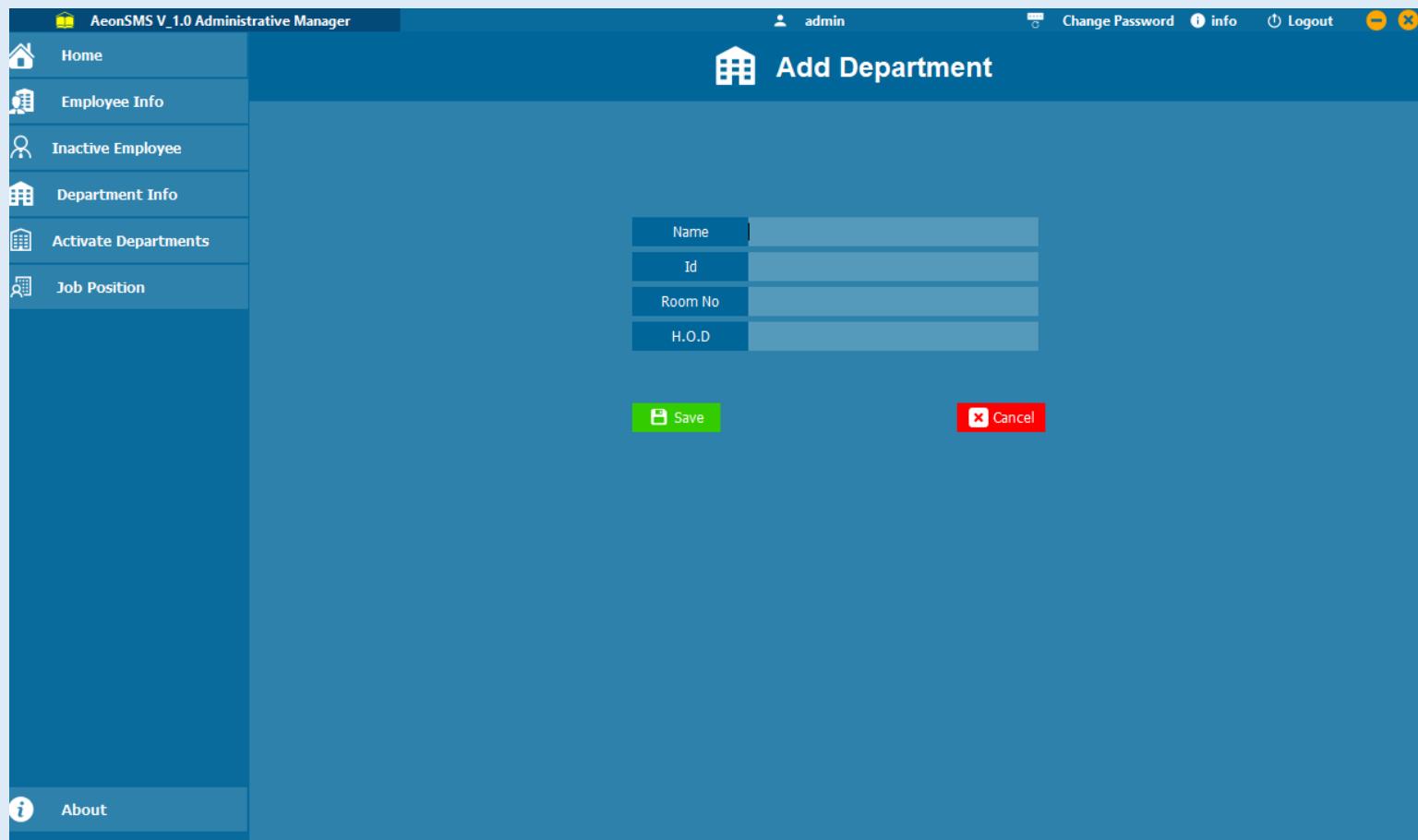


Figure 5.2. Add Department Screen

The screenshot shows the AeonSMS V_1.0 Administrative Manager interface. The left sidebar contains links for Home, Employee Info, Inactive Employee, Department Info, Activate Departments, Job Position, and About. The main content area is titled "Edit Department" and displays a form with the following data:

Name	Admission Department
Id	ADD012
Room No	XI-34
H.O.D	Mr. Anakin Sam

At the bottom are three buttons: "Save" (green), "Reset" (blue), and "Cancel" (red).

Figure 5.3. Edit Department Screen

8. Position Management Screen

The screenshot shows the 'Employee Post List' screen of the AeonSMS V_1.0 Administrative Manager. The top navigation bar includes links for 'Change Password', 'info', 'Logout', and a close button. On the left, a vertical sidebar menu lists 'Home', 'Employee Info', 'Inactive Employee', 'Department Info', 'Activate Departments', 'Job Position' (which is selected and highlighted in blue), and 'About'. The main content area is titled 'Employee Post List' and contains a search bar with fields for 'Name' and 'Department' (set to 'Choose') and a 'Search' button. Below the search bar is a table with columns 'ID', 'Name', and 'Department'. The data in the table is as follows:

ID	Name	Department
3	Math Teacher	Teacher
4	Instructor	Maths
5	Student Manager	Head
6	Administrator	Head
7	new	Sport

At the bottom of the screen are four buttons: '+ Add' (green), 'Edit' (pink), 'Deactivate' (red), and 'Export' (blue).

Figure 6.1. Position Management Screen

The screenshot shows the AeonSMS V_1.0 Administrative Manager interface. The top navigation bar includes links for 'Change Password', 'info', 'Logout', and a close button. On the left, a sidebar menu lists 'Home', 'Employee Info', 'Inactive Employee', 'Department Info', 'Activate Departments', 'Job Position' (which is selected), and 'About'. The main content area is titled 'Add Position' and contains three input fields: 'ID' (09X12), 'Job Title' (Maths Teacher), and 'Department' (a dropdown menu set to 'Choose'). At the bottom are 'Save' and 'Cancel' buttons.

Figure 6.2. Add New Job Position Screen

AeonSMS V.1.0 Administrative Manager

admin

Change Password info Logout

Edit Position

ID: 09X12

Job Title: Bus Driver

Department: Choose

Save Reset Cancel

Home Employee Info Inactive Employee Department Info Activate Departments Job Position About

The screenshot shows the 'Edit Position' page of the AeonSMS administrative manager. The left sidebar contains links for Home, Employee Info, Inactive Employee, Department Info, Activate Departments, Job Position, and About. The main area has tabs for Change Password, info, and Logout. The 'Edit Position' tab is active. It shows fields for ID (09X12), Job Title (Bus Driver), and Department (Choose). At the bottom are Save, Reset, and Cancel buttons.

Figure 6.3. Edit Job Position Screen

9. Student Management Screen

Student List

ID	Register No	No	Name	Batch	Phone	Email	Gender
3	3 1	ank		9934233556	ark010@gmail....	Male	
4	4 4	anakin	new	9934233556		Male	
5	5 1	Aeon	new	9934233556	anakin@aeon.com	Male	
6	6 4	ajit	Class X	6539390003	ANKRIV@gmail....	Male	

Action Buttons:

- Detail
- Add
- Edit
- Deactivate
- Export

Figure 7.2. Student Management Screen

Student Details

Back

Register No	3	Roll No	1
Name	ank	Gender	Male
Date of Birth	1993-12-22	Batch	new
Email	ank010@gmail.com	Phone	9934233556
Previous School	Self	Time	2018-03-14
Reason leaving	No reason		
Address	Qno 506, Line tank toad Jharkhand Ranchi 834001		

Extra Curricular

ID	Name	Location	Start	End	Notes
3	Science	0	2018-03-22		

Remark

ID	Staff No	Staff name	Content	Time

Figure 7.2. Student detail Screen

AeonSMS V_1.0 Administrative Manager admin Change Password info Logout

Update Student Detail

Back

Register No	6
D.O.B	Mar 2, 2018
Class	Class X
Roll No	4
Phone	6539390003
Last School	DAV
Reason leaving	

Name	ajit
Gender	Male
Batch	
Email	ANKIRV@gmail.com
Time	Mar 7, 2018

Address: house 6

Extra Curricular

ID	Name	Location	Start	End	Notes

Mark

Figure 7.3. Edit Student Screen

The screenshot shows the 'Add New Student' page of the AeonSMS administrative manager. The left sidebar contains links for Home, Student Info, Suspended Student, Extra Curricular, Activate Extra Curricular, Class List, Batch, Activate Batch, Course, Activate Course, Marks Management, and About. The main area has a title 'Add New Student' with a back button. It includes fields for Register No (6), Name, Gender (Male), D.O.B., Class (Choose), Batch, Email, Phone, Time, Last School, Reason leaving, and Address. Below this is an 'Extra Curricular' section with columns for ID, Name, Location, Start, End, and Notes, along with add, edit, and delete buttons. At the bottom is a 'Mark' button.

Figure 7.4. Edit Student Screen

10. Subject Management Screen

AeonSMS V_1.0 Administrative Manager

admin

Change Password info Logout

Subjects

ID	Subject Title	Teacher	Class	Room No
1	Science			
2	Maths			
3	Polity			

Add Edit Deactivate Export

Figure 8.1. Subject Management Screen

11. Batch Management Screen

AeonSMS V_1.0 Administrative Manager

admin

Change Password | info | Logout

Batch Manager

ID	Name	Date
1	new	2018-03-20
2	Class X	2018-03-23

Add | Edit | Deactivate | Export

Figure 9.1. Batch Management Screen

12. Mark Management Screen

 AeonSMS V_1.0 Administrative Manager admin Change Password info Logout - X

Figure 10.1. Mark Management Screen

13.Extra-Curricular Management Screen

The screenshot shows the 'Extra Curricular' management screen within the AeonSMS V_1.0 Administrative Manager. The left sidebar contains navigation links for Home, Student Info, Suspended Student, Extra Curricular, Activate Extra Curricular, Class List, Batch, Activate Batch, Course, Activate Course, Marks Management, and About. The main area has a title 'Extra Curricular' with a search icon. Below it are two sets of search fields: one for 'Name' and 'Start Month' with a 'Search' button, and another for 'ID' and 'End Month' with a 'Reset' button. A table displays a single record: ID 1, Name asad, Location scvsx, Start 2018-03-16, and End 2018-03-16. At the bottom are four buttons: 'Add' (green), 'Edit' (pink), 'Deactivate' (red), and 'Export' (blue).

ID	Name	Location	Start	End
1	asad	scvsx	2018-03-16	2018-03-16

Figure 11.1. Extra-Curricular Management Screen

14. Export

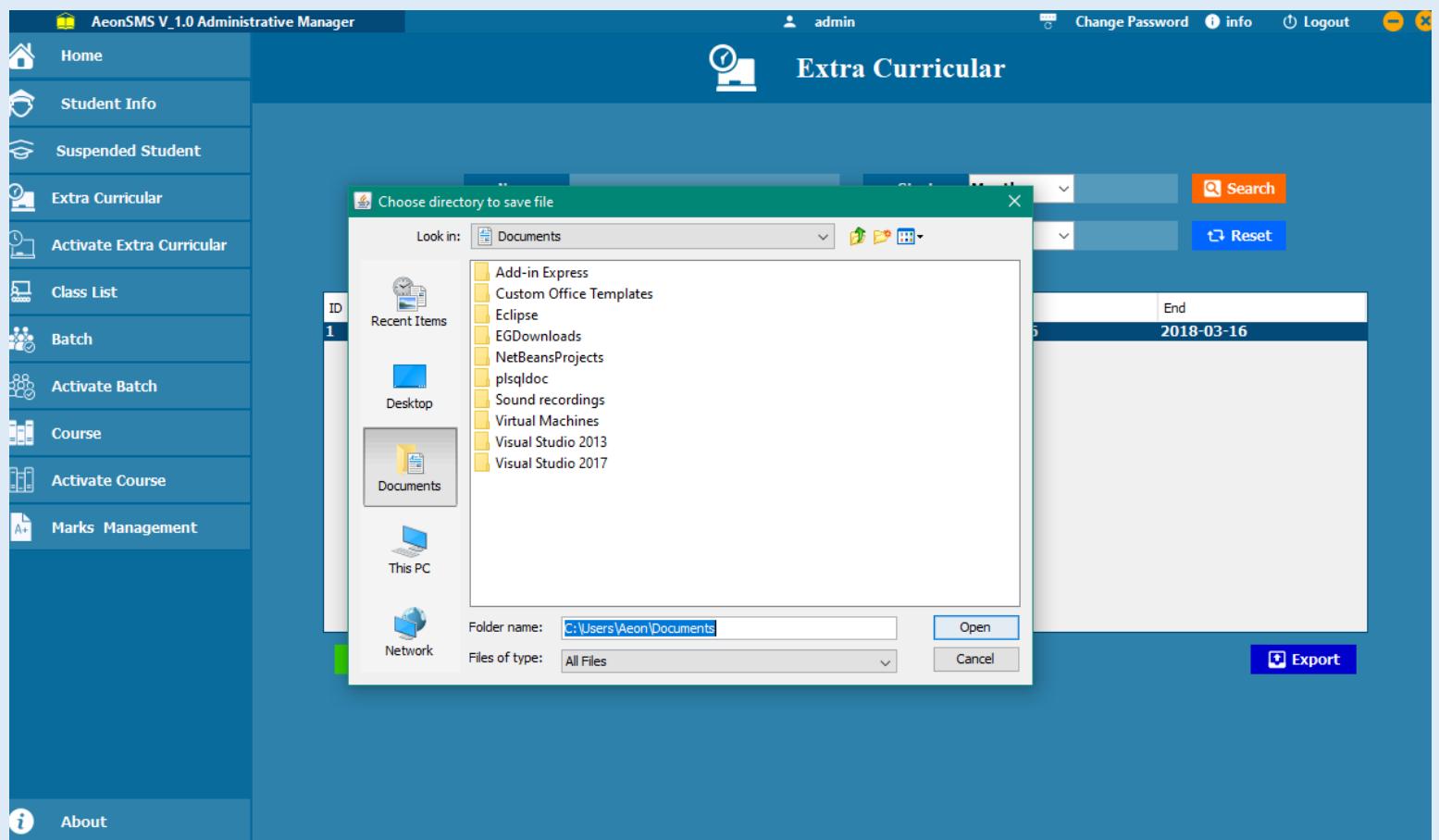


Figure 12.1. Export Screen

	A	B	C	D	E	F	G	H
1								
2								
3								
4								
5								
6								
7								
8								
9								
10	ID	Name	Location	Start	End			
11	2	sdfsdfsdfsdfsdf	sdfsdfsdf	2013-10-03	2017-10-18			
12	3	sdsdfsdf	dfqdfqdfg	2013-10-03	2013-12-20			
13	4	sdsdq	dfqdfqdfg	2013-10-08	2013-10-09			
14	5	sdqfdq	dfqdfqdfg	2013-10-01	2013-10-14			
15	6	dfqdfqdfq	dfqdfqdfg	2013-10-03	2013-10-02			
16	7	fsdfsd	tsdfsdfsdfsd	2013-10-15	2013-10-16			
17	8	dqdf	fqfqf	2013-10-03	2013-10-11			
18	9	sdfsdfsdfsdf	dfsdfsdf	2012-10-04	2012-10-17			
19	10	sdfsdf	sdfsdfsdfs	2013-10-04	2013-10-17			
20								
21								
22								
23								
24								
25								
26								
27								
28								
29								
30								
31								

Figure 12.2. Excel File after Export

AeonSMS For IGNOU

15.Logout

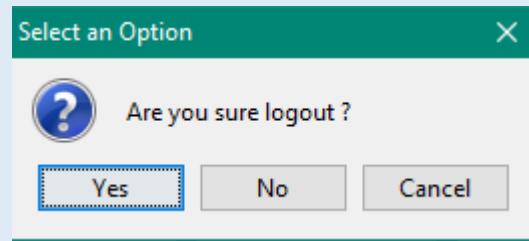


Figure 13.1. Logout Confirm Screen

4. Task Sheet

Project Ref. No.:		Project Title:	Activity Plan Prepared By:	Date of Preparation of Activity Plan:			
Sr. No.	Task			Actual Start Date	Actual Days	Team Member Names	Status
1	GUI Design	School Management			7	All member	Completed
2	Database Design	School Management			2	All member	Completed
3	ERD,DFD	School Management			2	All member	Completed
4	Coding	School Management			20	All member	Completed
5	Error Checking	School Management			5	All member	Completed

*Source codes
for
School Management System*

Submitted by –

Ankit Raj
159802231
BCA 6th Semester
Session 2015-2018

SCHOOL OF COMPUTER AND INFORMATION SCIENCE



5. System Development

5.1. Source code

5.1.1. Codes for Controls

```
// for admin related db activities
package Controls;

/**
 *
 * @author Aeon(Ankit RudrAshokJha)
 */
public class Admin extends Manager implements UserMng{
    @Override
    public int addUser(String[] data) {
        String SQL="INSERT user (usrn, pswd, name, role, email) VALUES ( "
                + ""+data[0]+", "
                + ""+data[1]+", "
                + ""+data[2]+", "
                +data[3]+",
                + ""+data[4]+")";
        System.out.println(SQL);
        return executeUpdate(SQL);
    }

    @Override
    public int editUser(String[] data) {
        String SQL="UPDATE user SET "
                + "usrn='"+data[1]+", "
                + "pswd='"+data[2]+", "
                + "name='"+data[3]+", "
                + "role='"+data[4]+", "
                + "email='"+data[5]+"' "
                + "WHERE ID="+data[0];
        System.out.println(SQL);
        return executeUpdate(SQL);
    }

    @Override
    public int deactiveUser(int ID) {
        String SQL="UPDATE user SET status=0 WHERE ID="+ID;
        return executeUpdate(SQL);
    }
}
```

```
}

// for user authentication

/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */
package Controls;

import DataSett.ConnectDB;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.sql.Timestamp;
import java.util.logging.Level;
import java.util.logging.Logger;

/**
 *
 * @author Aeon(Ankit RudrAshokJha)
 */
public class Authentication {

    private int ID;
    private int role;
    private String name;

    public static int CONNECT_FAIL=-1;
    public static int NOT_AUTH=0;
    public static int AUTH=1;
    public static int ADMIN=1;
    public static int USER=0;
    public static int LOGIN=1;
    public static int LOGOUT=0;
    public static int SUCCESS=1;
    public static int FAIL=0;
    public static int HEADER_ADMIN=1;
    public static int STAFF_MANAGER=2;
    public static int STUDENT_MANAGER=3;

    public int getID() {
```

```
        return ID;
    }

    public String getName() {
        return name;
    }

    public int getRole() {
        return role;
    }

    public boolean check(){
        if(ID > 0){
            return true;
        }
        return false;
    }

    public Authentication(int ID, int role) {
        this.ID = ID;
        this.role = role;
    }

    public Authentication() {
    }

    public int login(String usrn, String pswd, int role){
        Authentication authentication=null;
        Connection connect=ConnectDB.getConnect();
        if(connect==null){
            return CONNECT_FAIL;
        }
        PreparedStatement state=null;
        ResultSet res=null;
        String SQL="";
        if(role==USER){
            SQL="SELECT ID, name FROM user WHERE usrn=? and pswd=?";
        } else if(role==ADMIN){
            SQL="SELECT ID, name FROM admin WHERE username=? and password=?";
        }
        try {
            state=connect.prepareStatement(SQL);
            state.setString(1, usrn);
            state.setString(2, pswd);
            res=state.executeQuery();
            if(res.next()){
                int id_people=res.getInt(1);
```

```

if(role==USER){
    this.role=getRoleManager(id_people);
    addAuthTable(id_people, this.role, LOGIN);
    this.ID=id_people;

}else if(role==ADMIN){
    addAuthTable(id_people, HEADER_ADMIN, LOGIN);
    this.ID=id_people;
    this.role=HEADER_ADMIN;
}
this.name=res.getString(2);
return AUTH;
}
} catch (SQLException ex) {
    Logger.getLogger(Authentication.class.getName()).log(Level.SEVERE, null, ex);
}finally {
    try {
        connect.close();
    } catch (SQLException ex) {
        Logger.getLogger(StaffManager.class.getName()).log(Level.SEVERE, null, ex);
    }
}
return NOT_AUTH;
}
private int getRoleManager(int ID){
    Connection connect=ConnectDB.getConnect();
    Statement state=null;
    ResultSet res=null;
    try {
        String SQL="SELECT role FROM user where ID="+ID;
        state=connect.createStatement();
        res=state.executeQuery(SQL);
        if(res.next()){
            int role=res.getInt(1);
            return role;
        }
    } catch (SQLException ex) {
        Logger.getLogger(Authentication.class.getName()).log(Level.SEVERE, null, ex);
    }finally {
        try {
            connect.close();
        } catch (SQLException ex) {
            Logger.getLogger(StaffManager.class.getName()).log(Level.SEVERE, null, ex);
        }
    }
    return FAIL;
}

```

```

    }
    private int addAuthTable(int ID,int role, int action){
        Connection connect=ConnectDB.getConnect();
        if(connect==null){
            return CONNECT_FAIL;
        }
        PreparedStatement state=null;
        String SQL="INSERT authentication(ID_People, role, action) values(?, ?, ?)";
        int res=0;
        try {
            state=connect.prepareStatement(SQL, ResultSet.TYPE_SCROLL_SENSITIVE,
ResultSet.CONCUR_UPDATABLE);
            state.setInt(1, ID);
            state.setInt(2, role);
            state.setInt(3, action);
            res=state.executeUpdate();
            if(res==1){
                return SUCCESS;
            }
        } catch (SQLException ex) {
            Logger.getLogger(Authentication.class.getName()).log(Level.SEVERE, null, ex);
        }finally {
            try {
                connect.close();
            } catch (SQLException ex) {
                Logger.getLogger(StaffManager.class.getName()).log(Level.SEVERE, null, ex);
            }
        }
        return FAIL;
    }
    public Timestamp getLastLoin(){
        Timestamp lastLogin=null;
        Connection connect=null;
        PreparedStatement state=null;
        ResultSet res=null;
        if(!check()){
            return null;
        }
        connect=ConnectDB.getConnect();
        if(connect==null){
            return null;
        }
        String SQL="select time from authentication where ID_People=? and action=0 and role=?
ORDER BY time DESC limit 1";
        try {
            state=connect.prepareStatement(SQL);

```

```

state.setInt(1, this.ID);
state.setInt(2, this.role);
res=state.executeQuery();
if(res.next()){
    lastLogin=res.getTimestamp(1);
}
} catch (SQLException ex) {
    Logger.getLogger(Authentication.class.getName()).log(Level.SEVERE, null, ex);
}finally {
    try {
        connect.close();
    } catch (SQLException ex) {
        Logger.getLogger(StaffManager.class.getName()).log(Level.SEVERE, null, ex);
    }
}
return lastLogin;
}
public int logout(){
    if(this.ID <1){
        return FAIL;
    }
    int res=addAuthTable(this.ID, this.role, LOGOUT);
    this.ID=0;
    this.role=-1;
    this.name=null;
    return res;
}
}

// for batch entity interface
package Controls;

/**
 *
 * @author Aeon(Ankit RudrAshokJha)
 */
public interface BatchMng {
    public int addBatch(String[] data);
    public int editBatch(String[] data);
    public String removeBatch(int ID);
    public int deactiveBatch(int ID);
}

```

```
// for department entity interface
package Controls;
public interface DepartmentMng {
    public int deactivateDepartment(int ID);
    public int editDepartment(String[] data);
    public int addDepartment(String[] data);
    public String removeDepartment(int ID);
}

// for extracurricular entity interface

package Controls;
public interface ExtraCurricularMng {
    public int addExtraCurricular(String[] data);
    public int removeExtraCurricular(int ID);
    public int deactivateExtraCurricular(int ID);
    public int editExtraCurricular(String[] data);
}

// for department entity interface

/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */
package Controls;

import DataSett.ConnectDB;
import java.awt.Component;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.List;
import java.util.Vector;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JComboBox;
import javax.swing.JOptionPane;
import javax.swing.JTable;
```

```
import javax.swing.table.DefaultTableModel;

/**
 *
 * @author Aeon(Ankit RudrAshokJha)
 */
public abstract class Manager {

    public static final int SUCCESS = 1;
    public static final int FAIL = 0;
    public static final int CONNECT_FAIL = -1;
    public static final int RECORD_IN_PAGE = 10;
    public static final int MALE = 1;
    public static final int FEMALE = 0;
    public static final int OTHER_GENDER = -1;
    public static final int MARRIED = 1;
    public static final int SINGLE = 0;
    public static final String STUDENT_MODEL[] = {"ID", "Register No", "No", "Name",
        "Batch", "Phone", "Email", "Gender"};
    public static final String STAFF_MODEL[] = {"No", "Name", "Phone", "Email", "Position",
        "Department", "Gender"};
    public static final String DEPARTMENT_MODEL[] = {"ID", "Name"};
    public static final String POSITION_MODEL[] = {"ID", "Name", "Department"};
    public static final String BATCH_MODEL[] = {"ID", "Name", "Date"};
    public static final String MARK_MODEL[] = {"ID", "Subject", "Student No", "Student",
        "Point", "Time"};
    public static final String EXTRA_CURRICULAR_MODEL[] = {"ID", "Name", "Location",
        "Start", "End"};
    public static final String SUBJECT_MODEL[] = DEPARTMENT_MODEL;
    public static final String USER_MODEL[] = {"ID", "Username", "Password", "Name", "Role",
        "Email"};
    public static final String STAFF_CHOOSER_MODEL[] = {"No", "Name", "Position",
        "Department"};
    public static final int USER = 0;
    public static final int STAFF = 1;
    public static final int STUDENT = 2;
    public static final int BATCH = 3;
    public static final int SUBJECT = 4;
    public static final int POSITION = 5;
    public static final int DEPARTMENT = 6;
    public static final int EXTRA_CURRICULAR = 7;
    public static final int MARK=8;
    public static final int DEPARTA = 9;
```

```
public int executeManySQL(List<String> listSQL) {
    Connection connect = null;
    connect = ConnectDB.getConnect();
    if (connect == null) {
        return CONNECT_FAIL;
    }
    Statement state = null;
    try {
        connect.setAutoCommit(false);
        state = connect.createStatement(ResultSet.TYPE_SCROLL_SENSITIVE,
ResultSet.CONCUR_UPDATABLE);
        for (String SQL : listSQL) {
            state.addBatch(SQL);
            System.out.println(SQL);
        }
        state.executeBatch();
        connect.commit();
    } catch (SQLException ex) {
        Logger.getLogger(StaffManager.class.getName()).log(Level.SEVERE, null, ex);
        try {
            connect.rollback();
            return FAIL;
        } catch (SQLException ex1) {
            Logger.getLogger(StaffManager.class.getName()).log(Level.SEVERE, null, ex1);
        } finally {
            try {
                connect.close();
            } catch (SQLException ex1) {
                Logger.getLogger(StaffManager.class.getName()).log(Level.SEVERE, null, ex1);
            }
        }
    }
    return SUCCESS;
}

public int executeUpdate(String SQL) {
    System.out.println(SQL);
    Connection connect = null;
    connect = ConnectDB.getConnect();
    int res = 0;
    if (connect == null) {
        return CONNECT_FAIL;
    }
    Statement state = null;
    try {
```

```

        res = connect.createStatement(ResultSet.TYPE_SCROLL_SENSITIVE,
ResultSet.CONCUR_UPDATABLE).executeUpdate(SQL);
        if (res != 1) {
            return FAIL;
        }
    } catch (SQLException ex) {
        Logger.getLogger(StaffManager.class.getName()).log(Level.SEVERE, null, ex);
        return FAIL;
    } finally {
        try {
            connect.close();
        } catch (SQLException ex) {
            Logger.getLogger(Manager.class.getName()).log(Level.SEVERE, null, ex);
        }
    }
    return SUCCESS;
}

public int executeUpdate(String SQL, String data[]) {
    Connection connect = null;
    connect = ConnectDB.getConnect();
    int res = 0;
    if (connect == null) {
        return CONNECT_FAIL;
    }
    PreparedStatement state = null;
    try {
        state=connect.prepareStatement(SQL);
        for(int i=0; i<data.length; i++){
            state.setString(i+1, data[i]);
        }
        res=state.executeUpdate();
        if (res != 1) {
            return FAIL;
        }
    } catch (SQLException ex) {
        Logger.getLogger(StaffManager.class.getName()).log(Level.SEVERE, null, ex);
        return FAIL;
    } finally {
        try {
            connect.close();
        } catch (SQLException ex) {
            Logger.getLogger(Manager.class.getName()).log(Level.SEVERE, null, ex);
        }
    }
    return SUCCESS;
}

```

```
public Date parseDate(String value) {
    if (!Validate.checkDate(value)) {
        return null;
    }
    Date date = null;
    SimpleDateFormat format = new SimpleDateFormat("yyyy-MM-dd");
    try {
        date = format.parse(value);
    } catch (ParseException | NullPointerException ex) {
        Logger.getLogger(Manager.class.getName()).log(Level.SEVERE, null, ex);
        return null;
    }
    return date;
}

public String parseRole(int role) {
    if (role == 1) {
        return "Admin";
    } else if (role == 2) {
        return "Staff manager";
    } else if (role == 3) {
        return "Student manager";
    } else {
        return "Unknown";
    }
}

public String formatDate(Date date) {
    if (date == null) {
        return "";
    }
    SimpleDateFormat format = new SimpleDateFormat("yyyy-MM-dd");
    String str = format.format(date);
    if (str == null || str.isEmpty()) {
        return "";
    }
    return str;
}

public String parseGender(int gender) {
    if (gender == MALE) {
        return "Male";
    } else if (gender == FEMALE) {
        return "Female";
    }
    return "Other";
```

```
}

public int formatGender(String gender) {
    if (gender.equalsIgnoreCase("Male")) {
        return 1;
    } else if (gender.equalsIgnoreCase("Female")) {
        return 0;
    }
    return -1;
}

public String parseMarried(int married) {
    if (married == MARRIED) {
        return "Married";
    } else if (married == SINGLE) {
        return "Single";
    }
    return "Single";
}

public int formatMarried(String married) {
    if (married.equalsIgnoreCase("Married")) {
        return 1;
    } else if (married.equalsIgnoreCase("Single")) {
        return 0;
    }
    return -1;
}

public int parseType(String value) {
    if (value.equalsIgnoreCase("Staff manager")) {
        return Authentication.STAFF_MANAGER;
    }
    if (value.equalsIgnoreCase("Student manager")) {
        return Authentication.STUDENT_MANAGER;
    }
    return -1;
}

public void loadCombobox(String SQL, JComboBox combobox) {
    combobox.removeAllItems();
    combobox.addItem("Choose");
    Connection connect = null;
    connect = ConnectDB.getConnect();
    if (connect == null) {
        return;
    }
}
```

```
        }
        ResultSet res = null;

        try {
            res = connect.createStatement().executeQuery(SQL);
            while (res.next()) {
                combobox.addItem(res.getString(1));
            }
        } catch (SQLException ex) {
            Logger.getLogger(Manager.class.getName()).log(Level.SEVERE, null, ex);
        } finally {
            try {
                connect.close();
            } catch (SQLException ex) {
                Logger.getLogger(Manager.class.getName()).log(Level.SEVERE, null, ex);
            }
        }
    }

    public int getIDentity(String SQL) {
        Connection connect = null;
        connect = ConnectDB.getConnect();
        if (connect == null) {
            return -1;
        }
        ResultSet res = null;
        try {
            res = connect.createStatement().executeQuery(SQL);
            if (res.next()) {
                return res.getInt(1);
            }
        } catch (SQLException ex) {
            Logger.getLogger(Manager.class.getName()).log(Level.SEVERE, null, ex);
        } finally {
            try {
                connect.close();
            } catch (SQLException ex) {
                Logger.getLogger(Manager.class.getName()).log(Level.SEVERE, null, ex);
            }
        }
        return -1;
    }

    public void loadTable(String SQL, JTable tbl, String[] ColumnName) {
        DefaultTableModel model = (DefaultTableModel)tbl.getModel();
        while (model.getRowCount() > 0) {
```

```

        model.removeRow(0);
    }
    Connection connect = null;
    connect = ConnectDB.getConnect();
    if (connect == null) {
        return;
    }
    ResultSet res = null;
    try {
        res = connect.createStatement().executeQuery(SQL);
        while (res.next()) {
            Vector row = new Vector();
            for (int i = 0; i < ColumnName.length; i++) {
                String str = res.getString(i + 1);
                if (ColumnName[i].equalsIgnoreCase("Gender")) {
                    str = parseGender(Integer.parseInt(str));
                } else if (ColumnName[i].equalsIgnoreCase("Role")) {
                    str = parseRole(Integer.parseInt(str));
                }
                row.add(str);
            }
            model.addRow(row);
        }
    } catch (SQLException ex) {
        Logger.getLogger(Manager.class.getName()).log(Level.SEVERE, null, ex);
    }
}

public void loadTable(PreparedStatement state, JTable tbl, String[] ColumnName) {
    DefaultTableModel model = (DefaultTableModel)tbl.getModel();
    while (model.getRowCount() > 0) {
        model.removeRow(0);
    }
    Connection connect = null;
    connect = ConnectDB.getConnect();
    if (connect == null) {
        return;
    }
    ResultSet res = null;
    try {
        res = state.executeQuery();
        while (res.next()) {
            Vector row = new Vector();
            for (int i = 0; i < ColumnName.length; i++) {
                String str = res.getString(i + 1);

```

```

        if (ColumnName[i].equalsIgnoreCase("Gender")) {
            str = parseGender(Integer.parseInt(str));
        } else if (ColumnName[i].equalsIgnoreCase("Role")) {
            str = parseRole(Integer.parseInt(str));
        }
        row.add(str);
    }
    model.addRow(row);
}

} catch (SQLException ex) {
    Logger.getLogger(Manager.class.getName()).log(Level.SEVERE, null, ex);
}
}

public void loadTable(Object[][] data, JTable tbl) {
    DefaultTableModel model = (DefaultTableModel)tbl.getModel();
    while (model.getRowCount() > 0) {
        model.removeRow(0);
    }
    for (Object[] object : data) {
        model.addRow(object);
    }
}

public int confirmRemove(Component parentComponent, String message) {
    return JOptionPane.showConfirmDialog(parentComponent, message);
}

public int getSelectID(JTable tbl) {
    int rowSelect = tbl.getSelectedRow();
    int ID = Integer.parseInt(tbl.getValueAt(rowSelect, 0).toString());
    return ID;
}

public Vector getSelectRow(JTable tbl) {
    int rowSelect = tbl.getSelectedRow();
    if (rowSelect == -1) {
        return null;
    }
    Vector row = new Vector();
    for (int i = 0; i < tbl.getColumnCount(); i++) {
        if (tbl.getValueAt(rowSelect, i) == null) {
            row.add("");
        } else {
            row.add(tbl.getValueAt(rowSelect, i));
        }
    }
}

```

```
        }
    }
    return row;
}

public void refreshTable(JTable tbl) {
    DefaultTableModel model = (DefaultTableModel)tbl.getModel();
    while (model.getRowCount() > 0) {
        model.removeRow(0);
    }
}

public void removeSelectedRowTable(JTable tbl) {
    DefaultTableModel model = (DefaultTableModel)tbl.getModel();
    model.removeRow(tbl.getSelectedRow());
}

public boolean checkUnique(String nameColumnn, String nameTable, String name) {
    Connection connect = null;
    connect = ConnectDB.getConnect();
    if (connect == null) {
        return false;
    }
    String SQL = "SELECT " + nameColumnn + " FROM " + nameTable;
    ResultSet res = null;
    try {
        res = connect.createStatement().executeQuery(SQL);
        while (res.next()) {
            if (res.getString(1).equalsIgnoreCase(name)) {
                return false;
            }
        }
    } catch (SQLException ex) {
        Logger.getLogger(Manager.class.getName()).log(Level.SEVERE, null, ex);
    }
    return true;
}

public boolean checkUnique(String SQL) {
    Connection connect = null;
    connect = ConnectDB.getConnect();
    if (connect == null) {
        return false;
    }
    ResultSet res = null;
    try {
```

```
res = connect.createStatement().executeQuery(SQL);
if (res.next()) {
    return false;
}
} catch (SQLException ex) {
    Logger.getLogger(Manager.class.getName()).log(Level.SEVERE, null, ex);
    return false;
}

return true;
}

public void showMessage(int res, Component parent) {
    if (res == CONNECT_FAIL) {
        JOptionPane.showMessageDialog(parent, "Connect fail ! Please try again !");
        return;
    }
    if (res == FAIL) {
        JOptionPane.showMessageDialog(parent, "Error ! Please try again !");
        return;
    }
    if (res == SUCCESS) {
        JOptionPane.showMessageDialog(parent, "Success !");
        return;
    }
}

public void showMessage(String res, Component parent) {
    JOptionPane.showMessageDialog(parent, res);
}

public String getNameObject(int ID, int object) {
    String SQL = "";
    switch (object) {
        case USER:
            SQL += "SELECT name FROM user WHERE ID=" + ID;
            break;
        case BATCH:
            SQL += "SELECT name FROM batch WHERE ID=" + ID;
            break;
        case DEPARTMENT:
            SQL += "SELECT name FROM department WHERE ID=" + ID;
            break;
        case POSITION:
            SQL += "SELECT name FROM position WHERE ID=" + ID;
            break;
    }
}
```

```
case STUDENT:  
    SQL += "SELECT name FROM student WHERE ID=" + ID;  
    break;  
case SUBJECT:  
    SQL += "SELECT name FROM subject WHERE ID=" + ID;  
    break;  
case STAFF:  
    SQL += "SELECT name FROM staff WHERE No=" + ID;  
    break;  
case EXTRA_CURRICULAR:  
    SQL += "SELECT name FROM extra_curricular WHERE ID=" + ID;  
    break;  
case DEPARTA:  
    SQL += "SELECT name FROM department WHERE ID=" + ID;  
    break;  
default:  
    SQL = "";  
    break;  
}  
if (SQL.isEmpty()) {  
    return "";  
}  
Connection connect = null;  
connect = ConnectDB.getConnect();  
ResultSet res = null;  
try {  
    res = connect.createStatement().executeQuery(SQL);  
    if (res.next()) {  
        return res.getString(1);  
    }  
} catch (SQLException ex) {  
    Logger.getLogger(Manager.class.getName()).log(Level.SEVERE, null, ex);  
    return "";  
} finally {  
    try {  
        connect.close();  
    } catch (SQLException ex) {  
        Logger.getLogger(Manager.class.getName()).log(Level.SEVERE, null, ex);  
        return "";  
    }  
}  
return "";  
}  
}
```

```

// for manager entity interface
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */
package Controls;

import DataSett.ConnectDB;
import java.awt.Component;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.List;
import java.util.Vector;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JComboBox;
import javax.swing.JOptionPane;
import javax.swing.JTable;
import javax.swing.table.DefaultTableModel;

/**
 *
 * @author Aeon(Ankit RudrAshokJha)
 */
public abstract class Manager {

    public static final int SUCCESS = 1;
    public static final int FAIL = 0;
    public static final int CONNECT_FAIL = -1;
    public static final int RECORD_IN_PAGE = 10;
    public static final int MALE = 1;
    public static final int FEMALE = 0;
    public static final int OTHER_GENDER = -1;
    public static final int MARRIED = 1;
    public static final int SINGLE = 0;
    public static final String STUDENT_MODEL[] = {"ID", "Register No", "No", "Name",
"Batch", "Phone", "Email", "Gender"};
    public static final String STAFF_MODEL[] = {"No", "Name", "Phone", "Email", "Position",
"Department", "Gender"};
    public static final String DEPARTMENT_MODEL[] = {"ID", "Name"};
}

```

```

public static final String POSITION_MODEL[] = {"ID", "Name", "Department"};
public static final String BATCH_MODEL[] = {"ID", "Name", "Date"};
public static final String MARK_MODEL[] = {"ID", "Subject", "Student No", "Student",
"Point", "Time"};
public static final String EXTRA_CURRICULAR_MODEL[] = {"ID", "Name", "Location",
"Start", "End"};
public static final String SUBJECT_MODEL[] = DEPARTMENT_MODEL;
public static final String USER_MODEL[] = {"ID", "Username", "Password", "Name", "Role",
"Email"};
public static final String STAFF_CHOOSER_MODEL[] = {"No", "Name", "Position",
"Department"};
public static final int USER = 0;
public static final int STAFF = 1;
public static final int STUDENT = 2;
public static final int BATCH = 3;
public static final int SUBJECT = 4;
public static final int POSITION = 5;
public static final int DEPARTMENT = 6;
public static final int EXTRA_CURRICULAR = 7;
public static final int MARK=8;
public static final int DEPARTA = 9;

```

```

public int executeManySQL(List<String> listSQL) {
    Connection connect = null;
    connect = ConnectDB.getConnect();
    if (connect == null) {
        return CONNECT_FAIL;
    }
    Statement state = null;
    try {
        connect.setAutoCommit(false);
        state = connect.createStatement(ResultSet.TYPE_SCROLL_SENSITIVE,
ResultSet.CONCUR_UPDATABLE);
        for (String SQL : listSQL) {
            state.addBatch(SQL);
            System.out.println(SQL);
        }
        state.executeBatch();
        connect.commit();
    } catch (SQLException ex) {
        Logger.getLogger(StaffManager.class.getName()).log(Level.SEVERE, null, ex);
        try {
            connect.rollback();
            return FAIL;

```

```

        } catch (SQLException ex1) {
            Logger.getLogger(StaffManager.class.getName()).log(Level.SEVERE, null, ex1);
        } finally {
            try {
                connect.close();
            } catch (SQLException ex1) {
                Logger.getLogger(StaffManager.class.getName()).log(Level.SEVERE, null, ex1);
            }
        }
    }

    return SUCCESS;
}

public int executeUpdate(String SQL) {
    System.out.println(SQL);
    Connection connect = null;
    connect = ConnectDB.getConnect();
    int res = 0;
    if (connect == null) {
        return CONNECT_FAIL;
    }
    Statement state = null;
    try {
        res = connect.createStatement	ResultSet.TYPE_SCROLL_SENSITIVE,
ResultSet.CONCUR_UPDATABLE).executeUpdate(SQL);
        if (res != 1) {
            return FAIL;
        }
    } catch (SQLException ex) {
        Logger.getLogger(StaffManager.class.getName()).log(Level.SEVERE, null, ex);
        return FAIL;
    } finally {
        try {
            connect.close();
        } catch (SQLException ex) {
            Logger.getLogger(Manager.class.getName()).log(Level.SEVERE, null, ex);
        }
    }
    return SUCCESS;
}

// public int executeUpdate(String SQL, String data[]) {
//     Connection connect = null;
//     connect = ConnectDB.getConnect();
//     int res = 0;

```

```

//      if (connect == null) {
//          return CONNECT_FAIL;
//      }
//      PreparedStatement state = null;
//      try {
//          state=connect.prepareStatement(SQL);
//          for(int i=0; i<data.length; i++){
//              state.setString(i+1, data[i]);
//          }
//          res=state.executeUpdate();
//          if (res != 1) {
//              return FAIL;
//          }
//      } catch (SQLException ex) {
//          Logger.getLogger(StaffManager.class.getName()).log(Level.SEVERE, null, ex);
//          return FAIL;
//      } finally {
//          try {
//              connect.close();
//          } catch (SQLException ex) {
//              Logger.getLogger(Manager.class.getName()).log(Level.SEVERE, null, ex);
//          }
//      }
//      return SUCCESS;
//  }

public Date parseDate(String value) {
    if (!Validate.checkDate(value)) {
        return null;
    }
    Date date = null;
    SimpleDateFormat format = new SimpleDateFormat("yyyy-MM-dd");
    try {
        date = format.parse(value);
    } catch (ParseException | NullPointerException ex) {
        Logger.getLogger(Manager.class.getName()).log(Level.SEVERE, null, ex);
        return null;
    }
    return date;
}

public String parseRole(int role) {
    if (role == 1) {
        return "Admin";
    } else if (role == 2) {
        return "Staff manager";
    } else if (role == 3) {

```

```
        return "Student manager";
    } else {
        return "Unknown";
    }
}

public String formatDate(Date date) {
    if (date == null) {
        return "";
    }
    SimpleDateFormat format = new SimpleDateFormat("yyyy-MM-dd");
    String str = format.format(date);
    if (str == null || str.isEmpty()) {
        return "";
    }
    return str;
}

public String parseGender(int gender) {
    if (gender == MALE) {
        return "Male";
    } else if (gender == FEMALE) {
        return "Female";
    }
    return "Other";
}

public int formatGender(String gender) {
    if (gender.equalsIgnoreCase("Male")) {
        return 1;
    } else if (gender.equalsIgnoreCase("Female")) {
        return 0;
    }
    return -1;
}

public String parseMarried(int married) {
    if (married == MARRIED) {
        return "Married";
    } else if (married == SINGLE) {
        return "Single";
    }
    return "Single";
}

public int formatMarried(String married) {
```

```
if (married.equalsIgnoreCase("Married")) {
    return 1;
} else if (married.equalsIgnoreCase("Single")) {
    return 0;
}
return -1;
}

public int parseType(String value) {
    if (value.equalsIgnoreCase("Staff manager")) {
        return Authentication.STAFF_MANAGER;
    }
    if (value.equalsIgnoreCase("Student manager")) {
        return Authentication.STUDENT_MANAGER;
    }
    return -1;
}

public void loadCombobox(String SQL, JComboBox combobox) {
    combobox.removeAllItems();
    combobox.addItem("Choose");
    Connection connect = null;
    connect = ConnectDB.getConnect();
    if (connect == null) {
        return;
    }
    ResultSet res = null;

    try {
        res = connect.createStatement().executeQuery(SQL);
        while (res.next()) {
            combobox.addItem(res.getString(1));
        }
    } catch (SQLException ex) {
        Logger.getLogger(Manager.class.getName()).log(Level.SEVERE, null, ex);
    } finally {
        try {
            connect.close();
        } catch (SQLException ex) {
            Logger.getLogger(Manager.class.getName()).log(Level.SEVERE, null, ex);
        }
    }
}

public int getIDEentity(String SQL) {
    Connection connect = null;
```

```

connect = ConnectDB.getConnect();
if (connect == null) {
    return -1;
}
ResultSet res = null;
try {
    res = connect.createStatement().executeQuery(SQL);
    if (res.next()) {
        return res.getInt(1);
    }
} catch (SQLException ex) {
    Logger.getLogger(Manager.class.getName()).log(Level.SEVERE, null, ex);
} finally {
    try {
        connect.close();
    } catch (SQLException ex) {
        Logger.getLogger(Manager.class.getName()).log(Level.SEVERE, null, ex);
    }
}
return -1;
}

public void loadTable(String SQL, JTable tbl, String[] ColumnName) {
    DefaultTableModel model = (DefaultTableModel)tbl.getModel();
    while (model.getRowCount() > 0) {
        model.removeRow(0);
    }
    Connection connect = null;
    connect = ConnectDB.getConnect();
    if (connect == null) {
        return;
    }
    ResultSet res = null;
    try {
        res = connect.createStatement().executeQuery(SQL);
        while (res.next()) {
            Vector row = new Vector();
            for (int i = 0; i < ColumnName.length; i++) {
                String str = res.getString(i + 1);
                if (ColumnName[i].equalsIgnoreCase("Gender")) {
                    str = parseGender(Integer.parseInt(str));
                } else if (ColumnName[i].equalsIgnoreCase("Role")) {
                    str = parseRole(Integer.parseInt(str));
                }
                row.add(str);
            }
        }
    }
}

```

```

        model.addRow(row);
    }

} catch (SQLException ex) {
    Logger.getLogger(Manager.class.getName()).log(Level.SEVERE, null, ex);
}
}

public void loadTable(PreparedStatement state, JTable tbl, String[] ColumnName) {
    DefaultTableModel model = (DefaultTableModel)tbl.getModel();
    while (model.getRowCount() > 0) {
        model.removeRow(0);
    }
    Connection connect = null;
    connect = ConnectDB.getConnect();
    if (connect == null) {
        return;
    }
    ResultSet res = null;
    try {
        res = state.executeQuery();
        while (res.next()) {
            Vector row = new Vector();
            for (int i = 0; i < ColumnName.length; i++) {
                String str = res.getString(i + 1);
                if (ColumnName[i].equalsIgnoreCase("Gender")) {
                    str = parseGender(Integer.parseInt(str));
                } else if (ColumnName[i].equalsIgnoreCase("Role")) {
                    str = parseRole(Integer.parseInt(str));
                }
                row.add(str);
            }
            model.addRow(row);
        }
    } catch (SQLException ex) {
        Logger.getLogger(Manager.class.getName()).log(Level.SEVERE, null, ex);
    }
}

public void loadTable(Object[][] data, JTable tbl) {
    DefaultTableModel model = (DefaultTableModel)tbl.getModel();
    while (model.getRowCount() > 0) {
        model.removeRow(0);
    }
    for (Object[] object : data) {

```

```

        model.addRow(object);
    }
}

public int confirmRemove(Component parentComponent, String message) {
    return JOptionPane.showConfirmDialog(parentComponent, message);
}

public int getSelectID(JTable tbl) {
    int rowSelect = tbl.getSelectedRow();
    int ID = Integer.parseInt(tbl.getValueAt(rowSelect, 0).toString());
    return ID;
}

public Vector getSelectRow(JTable tbl) {
    int rowSelect = tbl.getSelectedRow();
    if (rowSelect == -1) {
        return null;
    }
    Vector row = new Vector();
    for (int i = 0; i < tbl.getColumnCount(); i++) {
        if (tbl.getValueAt(rowSelect, i) == null) {
            row.add("");
        } else {
            row.add(tbl.getValueAt(rowSelect, i));
        }
    }
    return row;
}

public void refreshTable(JTable tbl) {
    DefaultTableModel model = (DefaultTableModel) tbl.getModel();
    while (model.getRowCount() > 0) {
        model.removeRow(0);
    }
}

public void removeSelectedRowTable(JTable tbl) {
    DefaultTableModel model = (DefaultTableModel) tbl.getModel();
    model.removeRow(tbl.getSelectedRow());
}

public boolean checkUnique(String nameColummn, String nameTable, String name) {
    Connection connect = null;
    connect = ConnectDB.getConnect();
    if (connect == null) {

```

```
        return false;
    }
    String SQL = "SELECT " + nameColumn + " FROM " + nameTable;
    ResultSet res = null;
    try {
        res = connect.createStatement().executeQuery(SQL);
        while (res.next()) {
            if (res.getString(1).equalsIgnoreCase(name)) {
                return false;
            }
        }
    } catch (SQLException ex) {
        Logger.getLogger(Manager.class.getName()).log(Level.SEVERE, null, ex);
    }
    return true;
}

public boolean checkUnique(String SQL) {
    Connection connect = null;
    connect = ConnectDB.getConnect();
    if (connect == null) {
        return false;
    }
    ResultSet res = null;
    try {
        res = connect.createStatement().executeQuery(SQL);
        if (res.next()) {
            return false;
        }
    } catch (SQLException ex) {
        Logger.getLogger(Manager.class.getName()).log(Level.SEVERE, null, ex);
        return false;
    }

    return true;
}

public void showMessage(int res, Component parent) {
    if (res == CONNECT_FAIL) {
        JOptionPane.showMessageDialog(parent, "Connect fail ! Please try again !");
        return;
    }
    if (res == FAIL) {
        JOptionPane.showMessageDialog(parent, "Error ! Please try again !");
        return;
    }
}
```

```
//      if (res == SUCCESS) {
//          JOptionPane.showMessageDialog(parent, "Success !");
//          return;
//      }
}

public void showMessage(String res, Component parent) {
    JOptionPane.showMessageDialog(parent, res);
}

public String getNameObject(int ID, int object) {
    String SQL = "";
    switch (object) {
        case USER:
            SQL += "SELECT name FROM user WHERE ID=" + ID;
            break;
        case BATCH:
            SQL += "SELECT name FROM batch WHERE ID=" + ID;
            break;
        case DEPARTMENT:
            SQL += "SELECT name FROM department WHERE ID=" + ID;
            break;
        case POSITION:
            SQL += "SELECT name FROM position WHERE ID=" + ID;
            break;
        case STUDENT:
            SQL += "SELECT name FROM student WHERE ID=" + ID;
            break;
        case SUBJECT:
            SQL += "SELECT name FROM subject WHERE ID=" + ID;
            break;
        case STAFF:
            SQL += "SELECT name FROM staff WHERE No=" + ID;
            break;
        case EXTRA_CURRICULAR:
            SQL += "SELECT name FROM extra_curricular WHERE ID=" + ID;
            break;
        case DEPARTA:
            SQL += "SELECT name FROM department WHERE ID=" + ID;
            break;
        default:
            SQL = "";
            break;
    }
    if (SQL.isEmpty()) {
        return "";
    }
}
```

```

    }
    Connection connect = null;
    connect = ConnectDB.getConnect();
    ResultSet res = null;
    try {
        res = connect.createStatement().executeQuery(SQL);
        if (res.next()) {
            return res.getString(1);
        }
    } catch (SQLException ex) {
        Logger.getLogger(Manager.class.getName()).log(Level.SEVERE, null, ex);
        return "";
    } finally {
        try {
            connect.close();
        } catch (SQLException ex) {
            Logger.getLogger(Manager.class.getName()).log(Level.SEVERE, null, ex);
            return "";
        }
    }
    return "";
}
}

```

- [for job position entity interface](#)

```

package Controls;

/**
 *
 * @author Aeon(Ankit RudrAshokJha)
 */
public interface PositionMng {
    public int addPosition(String[] data);
    public int deactivatePosition(int ID);
    public int editPosition(String[] data);
    public String removePosition(int ID);
}

```

- for Staff entity interface

```
package Controls;

import TableModel.Staff;

/**
 *
 * @author Aeon(Ankit RudrAshokJha)
 */
public interface StaffMng {
    public String searchStaff(int no, String name);
    public int addStaff(Staff staff);
    public int editStaff(Staff staff, Staff oldStaff);
    public int removeStaff(int ID);
    public int deactivateStaff(int ID);
    public Staff getDetailStaff(int No);
    public String getListStaff();
}
```

- for Student entity interface

```
package Controls;

import TableModel.Student;

/**
 *
 * @author Aeon(Ankit RudrAshokJha)
 */
public interface StudentMng {
    public String searchStudent(int no, String name);
    public int addStudent(Student student);
    public int editStudent(Student student, Student oldStudent);
    public int removeStudent(int ID);
    public int deactivateStudent(int ID);
    public Student getStudentDetail(int No);
    public String getStudentList(int page);
}
```

- for Subject entity interface

```
package Controls;

public interface SubjectMng {
    public int addSubject(String[] data);
    public String removeSubject(int ID);
    public int deactivateSubject(int ID);
```

```

        public int editSubject(String[] data);
    }

    • for General Manager class
package Controls;

public class GeneralManager extends Manager{

}

    • for Class Staff Manager Controller
package Controls;

import DataSett.ConnectDB;
import java.sql.Connection;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.ArrayList;
import java.util.List;
import java.util.logging.Level;
import java.util.logging.Logger;
import TableModel.Department;
import TableModel.FamilyStaff;
import TableModel.Feedback;
import TableModel.Position;
import TableModel.Promotion;
import TableModel.Relative;
import TableModel.Staff;
import TableModel.Student;

/**
 *
 * @author Aeon(Ankit RudrAshokJha)
 */
public class StaffManager extends Manager implements StaffMng, DepartmentMng,
PositionMng{

    @Override
    public String searchStaff(int No, String name) {

        String SQL = "SELECT * FROM " +
                    "(SELECT staff.No, staff.Name, staff.phone, staff.email, position.name as
Position, department.name as Department, staff.gender " +
                    "FROM staff " +
                    "INNER JOIN " +
                    "promotion ON staff.No=promotion.staff_No " +
                    "INNER JOIN " +

```

```

        "position ON promotion.position_ID=position.ID " +
        "INNER JOIN " +
            "department ON position.department_ID=department.ID " +
        "ORDER BY promotion.date DESC) AS sub "+

        "WHERE 2>1 ";
    if (name != null && !name.isEmpty()) {
        SQL += " AND name like '%" + name + "%'";
    }
    if (No !=0){
        SQL += " AND No like '" + No + "%'";
    }
    SQL+="GROUP BY No ";
    return SQL;
}

@Override
public int addStaff(Staff staff) {
    String SQL;
    List<String> listSQL=new ArrayList();
    if(staff.getBirthday() ==null){
        SQL = "INSERT staff(`name`, birthday, address, phone, email, experience,
education_Background, gender) VALUES("
        + "" + staff.getName() + ", "
        + "null , "
        + "" + staff.getAddress() + ", "
        + "" + staff.getPhone() + ", "
        + "" + staff.getEmail() + ", "
        + "" + staff.getExperience() + ", "
        + "" + staff.getEducation() + ", "
        + staff.getGender()
        + ");";
    }else{
        SQL = "INSERT staff(`name`, birthday, address, phone, email, experience,
education_Background, gender) VALUES("
        + "" + staff.getName() + ", "
        + "" + formatDate(staff.getBirthday()) + ", "
        + "" + staff.getAddress() + ", "
        + "" + staff.getPhone() + ", "
        + "" + staff.getEmail() + ", "
        + "" + staff.getExperience() + ", "
        + "" + staff.getEducation() + ", "
        + staff.getGender()
        + ");";
    }
    listSQL.add(SQL);
}

```

```

if (staff.getFamilyStaff() != null) {
    SQL = "SET @last_staff_id:=LAST_INSERT_ID();";
    listSQL.add(SQL);

    SQL = "INSERT family_staff(married, staff_No) values(" +
staff.getFamilyStaff().getMarried() + ", @last_staff_id);";
    listSQL.add(SQL);

    SQL = "SET @last_family_staff_id:=LAST_INSERT_ID();";
    listSQL.add(SQL);

    if (staff.getFamilyStaff().getRelatives() != null) {
        for (Relative relatives : staff.getFamilyStaff().getRelatives()) {
            SQL = "INSERT relatives_staff(relation, name, phone, address, occupation, gender,
family_Staff_ID) values("
                + """ + relatives.getRelation() + """,
                + """ + relatives.getName() + """,
                + """ + relatives.getPhone() + """,
                + """ + relatives.getAddress() + """,
                + """ + relatives.getOccupation() + """",
                + relatives.getGender() + """,
                + "@last_family_staff_id"
                + ");";
            listSQL.add(SQL);
        }
    }
}

if (staff.getPromotions() != null && staff.getPromotions().size() > 0) {
    for (Promotion promotion : staff.getPromotions()) {
        SQL = "INSERT promotion(position_ID, staff_No, date) VALUES("
            + promotion.getPosition().getID() + ", "
            + "@last_staff_id ,"
            + """ + formatDate(promotion.getTime()) + """
            + ")";
        listSQL.add(SQL);
    }
}

//insert feedback
return executeManySQL(listSQL);
}

@Override
public int editStaff(Staff staff, Staff oldStaff) {
    String SQL;
    List<String> listSQL=new ArrayList();

```

```

SQL = "SET @staff_No=" + staff.getNo();
listSQL.add(SQL);
SQL = "UPDATE staff SET "
+ "name=" + staff.getName() + ", "
+ "birthday= " + formatDate(staff.getBirthday()) + ", "
+ "address= " + staff.getAddress() + ", "
+ "phone= " + staff.getPhone() + ", "
+ "email= " + staff.getEmail() + ", "
+ "experience= " + staff.getExperience() + ", "
+ "education_Background=" + staff.getEducation() + ", "
+ "gender=" + staff.getGender()
+ ";";
listSQL.add(SQL);
//Family Infomation
if (staff.getFamilyStaff() != null) {
    if (staff.getFamilyStaff().getID() == 0) {
        SQL = "INSERT family_staff(married, staff_No) values(" +
staff.getFamilyStaff().getMarried() + ", @staff_No);";
    } else {
        SQL = "UPDATE family_staff SET married=" + staff.getFamilyStaff().getMarried() +
" WHERE staff_No=@staff_No;";
    }
    listSQL.add(SQL);
    System.out.println(SQL);

    if (staff.getFamilyStaff().getID() == 0) {
        SQL = "SET @family_staff_id=LAST_INSERT_ID();";
    } else {
        SQL = "SET @family_staff_id:=(SELECT ID FROM family_staff WHERE
staff_No=@staff_No);";
    }
    listSQL.add(SQL);

    if (staff.getFamilyStaff().getRelatives() != null) {
        for (Relative relatives : staff.getFamilyStaff().getRelatives()) {
            if (relatives.getID() == 0) {
                SQL = "INSERT relatives_staff(relation, name, phone, address, occupation,
gender, family_Staff_ID) values("
+ "" + relatives.getRelation() + ", "
+ "" + relatives.getName() + ", "
+ "" + relatives.getPhone() + ", "
+ "" + relatives.getAddress() + ", "
+ "" + relatives.getOccupation() + ", "
+ relatives.getGender() + ", "
+ "@family_staff_id"
+ ");";
            }
        }
    }
}

```

```

} else {
    SQL = "UPDATE relatives_staff SET "
        + "relation=" + relatives.getRelation() + "", "
        + "name=" + relatives.getName() + "", "
        + "phone=" + relatives.getPhone() + "", "
        + "address=" + relatives.getAddress() + "", "
        + "occupation=" + relatives.getOccupation() + "", "
        + "gender=" + relatives.getGender()
        + " WHERE ID=" + relatives.getID()
        + ";";
}
listSQL.add(SQL);
for(Relative re:oldStaff.getFamilyStaff().getRelatives()){
    boolean flag=false;
    for(Relative relative:staff.getFamilyStaff().getRelatives()){
        if(re.getID()==relative.getID()){
            flag=true;
            break;
        }
    }
    if(!flag){
        SQL="DELETE FROM relatives_staff WHERE ID="+re.getID();
        listSQL.add(SQL);
    }
}
}

}

}

//Promotion Infomation
if(staff.getPromotions() !=null && staff.getPromotions().size() >0){
    List<Promotion> promotions=staff.getPromotions();
    for(Promotion p:promotions{
        if(p.getID()==0){
            if(p.getTime()==null){
                SQL="INSERT promotion(position_ID, staff_No, date) VALUES ("+
                    + p.getPosition().getID()+" , "
                    + "@staff_No, "
                    + "null"
                    + ");";
            }else{
                SQL="INSERT promotion(position_ID, staff_No, date) VALUES ("+
                    + p.getPosition().getID()+" , "
                    + "@staff_No, "
                    + """+formatDate(p.getTime())+"""
                    + ");";
            }
        }
    }
}

```

```

        }
    }else{
        SQL="UPDATE promotion SET "
        + "position_ID="+p.getPosition().getID()+" , "
        + "staff_No=@staff_No"+", "
        + "date='"+formatDate(p.getTime())+"' "
        + " WHERE ID="+p.getID();
    }
    listSQL.add(SQL);
}
for(Promotion p:oldStaff.getPromotions()){
    boolean flag=false;
    for(Promotion pro:staff.getPromotions()){
        if(p.getID()==pro.getID()){
            flag=true;
            break;
        }
    }
    if(!flag){
        SQL="DELETE FROM promotion WHERE ID="+p.getID();
        listSQL.add(SQL);
    }
}
//Feedback Infomation
// if(staff.getFeedbacks() !=null && staff.getFeedbacks().size()>0){
//     List<Feedback> feedbacks=staff.getFeedbacks();
//     for(Feedback f:feedbacks){
//         if(f.getID()==0){
//             SQL="INSERT Feedback(staff_No, content, time, student_ID)";
//         }else{
//             SQL="UPDATE WHERE ID="+f.getID();
//         }
//     }
// }
return executeManySQL(listSQL);
}

@Override
public int removeStaff(int ID) {
    String SQL;
    List<String> listSQL=new ArrayList();

    SQL = "SET @staff_No:=" + ID + ";";
    listSQL.add(SQL);
}

```

```

SQL = "SET @family_Staff:=(SELECT ID FROM family_Staff WHERE
staff_No=@staff_No);";
listSQL.add(SQL);

//Delete relatives_staff
SQL = "DELETE FROM relatives_staff WHERE family_Staff_ID=@family_Staff;";
listSQL.add(SQL);

//Delete family_Staff
SQL = "DELETE FROM family_Staff WHERE ID=@family_Staff;";
listSQL.add(SQL);

//Delete promotion_has_staff
SQL = "DELETE FROM promotion_has_staff WHERE staff_No=@staff_No;";
listSQL.add(SQL);

//Delete feedback
SQL = "DELETE FROM feedback WHERE staff_No=@staff_No;";
listSQL.add(SQL);

//Delete staff
SQL = "DELETE FROM staff WHERE No=@staff_No;";
listSQL.add(SQL);

return executeManySQL(listSQL);
}

```

```

@Override
public Staff getDetailStaff(int No) {
    Staff staff = new Staff(No);
    Connection connect = null;
    connect = ConnectDB.getConnect();
    if (connect == null) {
        return null;
    }
    Statement state = null;
    ResultSet res = null;
    String SQL;
    // Load Person detail, education and experience
    SQL = "SELECT No, name, birthday,address, phone, email,
education_Background,experience, gender FROM staff WHERE No=" + No;
    try {
        state = connect.createStatement();
        res = state.executeQuery(SQL);
        int count = 0;

```

```

        if (res.next()) {
            staff = new Staff(res.getInt(++count), res.getString(++count),
parseDate(res.getString(++count)), res.getString(++count), res.getString(++count),
res.getString(++count), res.getString(++count), res.getString(++count), res.getInt(++count));
        } else {
            return null;
        }
        // Load Family
        SQL = "SELECT ID, married FROM family_staff WHERE staff_No=" + No;
        res = state.executeQuery(SQL);
        FamilyStaff familyStaff = null;
        if (res.next()) {
            familyStaff = new FamilyStaff(res.getInt(1), res.getInt(2));
        }
        if (familyStaff != null) {
            SQL = "SELECT ID, relation, name, phone, address, occupation, gender FROM
relatives_staff WHERE family_Staff_ID=" + familyStaff.getID();
            res = state.executeQuery(SQL);
            while (res.next()) {
                count = 0;
                Relative relatives = new Relative(res.getInt(++count), res.getString(++count),
res.getString(++count), res.getString(++count), res.getString(++count), res.getString(++count),
res.getInt(++count));
                familyStaff.addRelative(relatives);
            }
            staff.setFamilyStaff(familyStaff);
        }
        // Load Promotion
        SQL = "SELECT promotion.ID, promotion.position_ID, position.name,
position.department_ID, department.name, promotion.date "
        + "FROM promotion inner join position ON promotion.position_ID=position.ID "
        + "INNER JOIN department "
        + "ON department.ID=position.ID "
        + "WHERE promotion.staff_No=" + No;
        res = state.executeQuery(SQL);
        while (res.next()) {
            Department department=new Department(res.getInt(4), res.getString(5));
            Position position=new Position(res.getInt(2), res.getString(3), department);
            Promotion promotion=new Promotion(res.getInt(1), position,
parseDate(res.getString(6)));
            staff.addPromotion(promotion);
        }
        //Load Feedback
        SQL="SELECT feedback.ID, student.ID, student.academic_register_No, student.name,
feedback.content, feedback.date "
        + "FROM feedback inner join student ON feedback.student_ID=student.ID "

```

```

        + "WHERE feedback.staff_No="+No;
        System.out.println(SQL);
        res=state.executeQuery(SQL);
        List<Feedback> feedbacks=new ArrayList<>();
        staff.setFeedbacks(feedbacks);
        while(res.next()){
            Feedback f=new Feedback(res.getInt(1), new Student(res.getInt(2), res.getInt(3),
            res.getString(4)), res.getString(5), parseDate(res.getString(6)));
            staff.addFeedback(f);
        }
    }

} catch (SQLException ex) {
    Logger.getLogger(StaffManager.class.getName()).log(Level.SEVERE, null, ex);
    return null;
} finally {
    try {
        connect.close();
    } catch (SQLException ex) {
        Logger.getLogger(StaffManager.class.getName()).log(Level.SEVERE, null, ex);
    }
}
return staff;
}

@Override
public String getListStaff(){
    String SQL = "SELECT * FROM "+
        "(SELECT staff.No, staff.Name, staff.phone, staff.email, position.name as
Position, department.name as Department, staff.gender " +
        "FROM staff " +
        "LEFT JOIN " +
        "promotion ON staff.No=promotion.staff_No " +
        "LEFT JOIN " +
        "position ON promotion.position_ID=position.ID " +
        "LEFT JOIN " +
        "department ON position.department_ID=department.ID "+

        "WHERE staff.status=1 " +
        "ORDER BY promotion.date DESC) AS sub "+
        "GROUP BY No ";
    System.out.println(SQL);
    return SQL;
}

@Override
public int deactivateStaff(int ID) {

```

```
String SQL="UPDATE staff SET status=0 WHERE No="+ID;
    return executeUpdate(SQL);
}
@Override
public int deactivateDepartment(int ID) {
    String SQL="UPDATE department SET status=0 WHERE ID="+ID;
    int res=executeUpdate(SQL);
    return res;
}

@Override
public int editDepartment(String[] data) {
    String SQL="UPDATE department SET "
        + "name='"+data[1]+"'"
        + "WHERE ID='"+data[0];
    int res=executeUpdate(SQL);
    return res;
}

@Override
public int addDepartment(String[] data) {
    String SQL="INSERT department(name) VALUES('"+data[0]+"')";
    int res=executeUpdate(SQL);
    return res;
}

@Override
public int addPosition(String[] data) {
    List<String> listSQL=new ArrayList<>();
    listSQL.add("SET @idDepartment:=(SELECT ID FROM department where
name='"+data[1]+");");
    listSQL.add("INSERT position (name, department_ID) VALUES ('"+data[0]+"',
@idDepartment);");
    return executeManySQL(listSQL);
}

@Override
public int deactivatePosition(int ID) {
    String SQL="UPDATE position SET status=0 WHERE ID="+ID;
    return executeUpdate(SQL);
}

@Override
public int editPosition(String[] data) {
    List<String> listSQL=new ArrayList<>();
```

```

        listSQL.add("SET @idDepartment:=(SELECT ID FROM department where
name='"+data[2]+");");
        listSQL.add("UPDATE position SET name='"+data[1]+"' , department_ID=@idDepartment
WHERE ID="+data[0]+";");
        System.out.println(listSQL);
        return executeManySQL(listSQL);
    }

@Override
public String removePosition(int ID) {
    String SQL="";
    SQL="SELECT 1 FROM promotion WHERE position_ID="+ID;
    if(getIDEentity(SQL)==-1){
        SQL="DELETE FROM position WHERE ID=" + ID;
    }else{
        return "Have some staff have this position. You must remove promotion in staff info";
    }
    int res=executeUpdate(SQL);
    if(res==CONNECT_FAIL){
        return "Connect fail ! Please try again !";
    }
    if(res==FAIL){
        return "Have some error !";
    }
    if(res==SUCCESS){
        return "Success !";
    }
    return "Unknown !";
}

@Override
public String removeDepartment(int ID) {
    String SQL="";
    SQL="SELECT 1 FROM position WHERE department_ID="+ID;
    System.out.println(SQL);
    if(getIDEentity(SQL)==-1){
        SQL="DELETE FROM department WHERE ID=" + ID;
    }else{
        return "Have some positon have this department. You must remove or change position";
    }

    int res=executeUpdate(SQL);
    if(res==CONNECT_FAIL){
        return "Connect fail ! Please try again !";
    }
}

```

```

        if(res==FAIL){
            return "Have some error !";
        }
        if(res==SUCCESS){
            return "Success !";
        }
        return "Unknown !";
    }

}

• for Student Manager controller class
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */
package Controls;

import DataSett.ConnectDB;
import java.sql.Connection;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.ArrayList;
import java.util.List;
import java.util.logging.Level;
import java.util.logging.Logger;
import TableModel.Academic;
import TableModel.Admission;
import TableModel.Batch;
import TableModel.FamilyStudent;
import TableModel.Mark;
import TableModel.Relative;
import TableModel.Remark;
import TableModel.Staff;
import TableModel.Student;
import TableModel.Subject;

/**
 *
 * @author Aeon(Ankit RudrAshokJha)
 */
public class StudentManager extends Manager implements StudentMng, BatchMng,
ExtraCurricularMng, SubjectMng {

    @Override
    public String searchStudent(int No, String name) {

```

```

String SQL = "SELECT student.ID, student.academic_register_No, student.No,
student.name,batch.name AS Batch, student.phone, student.email, student.gender "
+ "from student INNER JOIN (academic inner join batch on
academic.batch_ID=batch.ID) ON academic.register_No=student.academic_register_No WHERE
2>1 ";
if (name != null && !name.isEmpty()) {
    SQL += " AND student.name like '%" + name + "%'";
}
if (No != 0) {
    SQL += " AND academic_register_No like '" + No + "%'";
}
return SQL;
}

@Override
public int addStudent(Student student) {
    String SQL;
    List<String> listSQL = new ArrayList();
    //Insert admission
    Admission admission = student.getAdmission();
    if(admission==null){
        admission=new Admission();
    }
    if (admission != null) {
        if(admission.getTime()==null){
            SQL = "INSERT admission (date, prev_Institute, reason_Leaving) VALUES("
                + "null , "
                + "" + admission.getPrev_Institute() + ", "
                + "" + admission.getReason_Leaving() + """
                + ");";
        }else{
            SQL = "INSERT admission (date, prev_Institute, reason_Leaving) VALUES("
                + "" + formatDate(admission.getTime()) + ", "
                + "" + admission.getPrev_Institute() + ", "
                + "" + admission.getReason_Leaving() + """
                + ");";
        }
    } else {
        SQL = "INSERT admission () VALUES();";
    }
    listSQL.add(SQL);
    //Get ID admission
    SQL = "SET @ID_ADMISSION:=LAST_INSERT_ID();";
    listSQL.add(SQL);
    //Insert academic
}

```

```

if (student.getAcademic() != null) {
    Batch b = student.getAcademic().getBatch();
    if (b != null) {
        if (b.getID() < 1) {
            SQL = "SET @ID_BATCH:=(SELECT ID FROM batch WHERE name='" +
b.getName() + "' LIMIT 0,1);";
        } else {
            SQL = "SET @ID_BATCH:=" + b.getID();
        }
        listSQL.add(SQL);
        SQL = "INSERT academic (batch_ID) VALUES (@ID_BATCH);";
        listSQL.add(SQL);
        //Get ID academic
        SQL = "SET @ID_ACADEMIC:=LAST_INSERT_ID();";
        listSQL.add(SQL);
    }
}
//Insert student
if(student.getBirthday()==null){
    SQL = "INSERT student (No, name, birthday, address, phone, email, gender,
admission_ID, academic_register_No) VALUES("
        + "" + student.getNo() + ", "
        + "" + student.getName() + ", "
        + "null , "
        + "" + student.getAddress() + ", "
        + "" + student.getPhone() + ", "
        + "" + student.getEmail() + ", "
        + student.getGender() + ", "
        + "@ID_ADMISSION, "
        + "@ID_ACADEMIC"
        + ");";
}else{
    SQL = "INSERT student (No, name, birthday, address, phone, email, gender,
admission_ID, academic_register_No) VALUES("
        + "" + student.getNo() + ", "
        + "" + student.getName() + ", "
        + "" + formatDate(student.getBirthday()) + ", "
        + "" + student.getAddress() + ", "
        + "" + student.getPhone() + ", "
        + "" + student.getEmail() + ", "
        + student.getGender() + ", "
        + "@ID_ADMISSION, "
        + "@ID_ACADEMIC"
        + ");";
}
listSQL.add(SQL);

```

```

//Get ID student
SQL = "SET @ID_STUDENT:=LAST_INSERT_ID();";
listSQL.add(SQL);
//Insert family_student
SQL = "INSERT family_student (student_ID) VALUES (@ID_STUDENT);";
listSQL.add(SQL);
//Get ID family_student
SQL = "SET @ID_FAMILY:=LAST_INSERT_ID();";
listSQL.add(SQL);
//Insert relatives_student
if (student.getFamilyStudent() != null) {
    List<Relative> relatives = student.getFamilyStudent().getListRelative();
    if (relatives != null && relatives.size() > 0) {
        for (Relative r : relatives) {
            SQL = "INSERT relatives_student (relation, name, phone, address, occupation,
            gender, family_student_ID)"
            + "VALUES("
            + "" + r.getRelation() + ", "
            + "" + r.getName() + ","
            + "" + r.getPhone() + ","
            + "" + r.getAddress() + ","
            + "" + r.getOccupation() + ","
            + r.getGender() + ","
            + "@ID_FAMILY"
            + ");";
            listSQL.add(SQL);
        }
    }
}
//Insert mark
//Get ID subject
List<Mark> marks = student.getMarks();
if (marks != null && marks.size() > 0) {
    for (Mark m : marks) {
        if (m.getSubject() != null) {
            if (m.getSubject().getID() == 0) {
                SQL = "SET @ID SUBJECT:=(SELECT ID FROM subject WHERE name='"
                + m.getSubject().getName() + "' LIMIT 0,1);";
            } else {
                SQL = "SET @ID SUBJECT:=" + m.getSubject().getID();
            }
            listSQL.add(SQL);
        }
        //Insert
        if(m.getTime()==null){
            SQL = "INSERT mark (subject_ID, point, date, student_ID) VALUES("
            + "@ID SUBJECT, "

```

```

        + m.getPoint() + ", "
        + " null , "
        + "@ID_STUDENT"
        + ");";
    }else{
        SQL = "INSERT mark (subject_ID, point, date, student_ID) VALUES("
        + "@ID SUBJECT, "
        + m.getPoint() + ","
        + "" + formatDate(m.getTime()) + ", "
        + "@ID_STUDENT"
        + ");";
    }
    listSQL.add(SQL);
}
}
}

//Insert remark
List<Remark> remarks = student.getRemarks();
if (remarks != null && remarks.size() > 0) {
    //Insert
    for (Remark r : remarks) {
        if (r.getStaff() == null || r.getStaff().getNo() <= 0) {
            continue;
        }
        if(r.getTime()==null){
            SQL = "INSERT remark (content, date, staff_No, student_ID) VALUES("
            + "" + r.getContent() + ", "
            + "null , "
            + r.getStaff().getNo() + ", "
            + "@ID_STUDENT"
            + ");";
        }else{
            SQL = "INSERT remark (content, date, staff_No, student_ID) VALUES("
            + "" + r.getContent() + ", "
            + "" + formatDate(r.getTime()) + ", "
            + r.getStaff().getNo() + ", "
            + "@ID_STUDENT"
            + ");";
        }
        listSQL.add(SQL);
    }
}
return executeManySQL(listSQL);
}

@Override

```

```

public int editStudent(Student student, Student oldStudent) {
    String SQL;
    List<String> listSQL = new ArrayList();
    SQL = "SET @ID_STUDENT:=" + oldStudent.getID() + ";";
    listSQL.add(SQL);
    //Update admission
    Admission admission = student.getAdmission();
    if (admission != null) {
        if(admission.getTime()==null){
            SQL = "UPDATE admission SET "
                + "prev_Institute=" + admission.getPrev_Institute() + ", "
                + "reason_Leaving=" + admission.getReason_Leaving() + ", "
                + "date=null "
                + "WHERE ID=" + admission.getID() + ";";
        }else{
            SQL = "UPDATE admission SET "
                + "prev_Institute=" + admission.getPrev_Institute() + ", "
                + "reason_Leaving=" + admission.getReason_Leaving() + ", "
                + "date=" + formatDate(admission.getTime()) + " "
                + "WHERE ID=" + admission.getID() + ";";
        }
        listSQL.add(SQL);
    }
    //Update batch
    Academic academic = student.getAcademic();
    if (academic != null) {
        Batch b = academic.getBatch();
        if (b != null) {
            if (b.getID() <= 0) {
                SQL = "SET @ID_BATCH:=(SELECT ID FROM batch WHERE name=" +
b.getName() + ");";
                listSQL.add(SQL);
                SQL = "UPDATE academic SET batch_ID=@ID_BATCH WHERE register_No=" +
academic.getRegisterNo() + ";";
                listSQL.add(SQL);
            }
        }
    }
    //Update student
    if(student.getBirthday()==null){
        SQL = "UPDATE student SET "
            + "No=" + student.getNo() + ", "
            + "name=" + student.getName() + ", "
            + "birthday= null, "
            + "phone=" + student.getPhone() + ", "
            + "address=" + student.getAddress() + ", "
    }
}

```

```

+ "email=" + student.getEmail() + "", "
+ "gender=" + student.getGender() + " "
+ "WHERE ID=@ID_STUDENT;";

}else{
SQL = "UPDATE student SET "
+ "No=" + student.getNo() + "", "
+ "name=" + student.getName() + "", "
+ "birthday=" + formatDate(student.getBirthday()) + "", "
+ "phone=" + student.getPhone() + "", "
+ "address=" + student.getAddress() + "", "
+ "email=" + student.getEmail() + "", "
+ "gender=" + student.getGender() + " "
+ "WHERE ID=@ID_STUDENT;";

}
listSQL.add(SQL);
//Update Family
if (student.getFamilyStudent() != null) {
    if (student.getFamilyStudent().getID() == 0) {
        SQL = "INSERT family_student(student_ID) values(@ID_STUDENT);";
    }
    listSQL.add(SQL);

    if (student.getFamilyStudent().getID() <= 0) {
        SQL = "SET @ID_FAMILY=LAST_INSERT_ID();";
    } else {
        SQL = "SET @ID_FAMILY:=" + student.getFamilyStudent().getID() + ";";
    }
    listSQL.add(SQL);

    if (student.getFamilyStudent().getListRelative() != null) {
        for (Relative relatives : student.getFamilyStudent().getListRelative()) {
            if (relatives.getID() == 0) {
                SQL = "INSERT relatives_student(relation, name, phone, address, occupation,
gender, family_student_ID) VALUES("
+ "" + relatives.getRelation() + "", "
+ "" + relatives.getName() + "", "
+ "" + relatives.getPhone() + "", "
+ "" + relatives.getAddress() + "", "
+ "" + relatives.getOccupation() + "", "
+ relatives.getGender() + ", "
+ "@ID_FAMILY"
+ ");";
            } else {
                SQL = "UPDATE relatives_student SET "
+ "relation=" + relatives.getRelation() + "", "
+ "name=" + relatives.getName() + "", "

```

```

        + "phone=" + relatives.getPhone() + "", "
        + "address=" + relatives.getAddress() + "", "
        + "occupation=" + relatives.getOccupation() + "", "
        + "gender=" + relatives.getGender()
        + " WHERE ID=" + relatives.getID()
        + ";";
    }
listSQL.add(SQL);
for (Relative re : oldStudent.getFamilyStudent().getListRelative()) {
    boolean flag = false;
    for (Relative relative : student.getFamilyStudent().getListRelative()) {
        if (re.getID() == relative.getID()) {
            flag = true;
            break;
        }
    }
    if (!flag) {
        SQL = "DELETE FROM relatives_student WHERE ID=" + re.getID() + ";";
        listSQL.add(SQL);
    }
}

}
//Update mark
List<Mark> marks = student.getMarks();
if (marks != null) {
    for (Mark m : marks) {
        if (m.getSubject().getID() <= 0) {
            SQL = "SET @ID SUBJECT:=(SELECT ID FROM subject WHERE name=" +
m.getSubject().getName() + ")";
        } else {
            SQL = "SET @ID SUBJECT:=" + m.getSubject().getID();
        }
        listSQL.add(SQL);
        if (m.getID() <= 0) {
            if (m.getTime() == null) {
                SQL = "INSERT mark (subject_ID, point, date, student_ID) VALUES(" +
                "@ID SUBJECT, "
                + m.getPoint() + ", "
                + "null , "
                + "@ID SUBJECT"
                + ")";
            } else {
                SQL = "INSERT mark (subject_ID, point, date, student_ID) VALUES(" +

```

```

        + "@ID SUBJECT, "
        + m.getPoint() + ", "
        + """" + formatDate(m.getTime()) + "", "
        + "@ID SUBJECT"
        + ")");
    }
} else {
    if(m.getTime()==null){
        SQL="UPDATE mark SET "
        + "subject_ID=@ID SUBJECT, "
        + "point="+m.getPoint()+" , "
        + "date=null "
        + "WHERE ID="+m.getID()+";";
    }else{
        SQL="UPDATE mark SET "
        + "subject_ID=@ID SUBJECT, "
        + "point="+m.getPoint()+" , "
        + "date='"+formatDate(m.getTime())+"' "
        + "WHERE ID="+m.getID()+";";
    }
}
listSQL.add(SQL);
}
if(oldStudent.getMarks() !=null && oldStudent.getMarks().size() >0){
    for(Mark m:oldStudent.getMarks()){
        System.out.println(m.getID());
        boolean flag=false;
        for(Mark mark:marks){
            System.out.println(mark.getID());
            if(m.getID()==mark.getID()){
                flag=true;
                break;
            }
        }
        if(!flag){
            SQL="DELETE FROM mark WHERE ID="+m.getID();
            listSQL.add(SQL);
        }
    }
}
//Update remark
List<Remark> remarks = student.getRemarks();
if (remarks != null) {
    for (Remark r : remarks) {
        if(r.getStaff()==null || r.getStaff().getNo()==0){

```

```

        continue;
    }
    if (r.getID() > 0) {
        if(r.getTime()==null){
            SQL="UPDATE remark SET "
            + "student_ID=@ID_STUDENT, "
            + "staff_No="+r.getStaff().getNo()+", "
            + "content='"+r.getContent()+"', "
            + "date=null "
            + "WHERE ID="+r.getID()+";";
        }else{
            SQL="UPDATE remark SET "
            + "student_ID=@ID_STUDENT, "
            + "staff_No="+r.getStaff().getNo()+", "
            + "content='"+r.getContent()+"', "
            + "date='"+formatDate(r.getTime())+"'"
            + "WHERE ID="+r.getID()+";";
        }
    }else{
        if(r.getTime()==null){
            SQL = "INSERT remark (student_ID, staff_No, content, date) VALUES("
            + "@ID_STUDENT, "
            + r.getStaff().getNo()+ ", "
            + """+r.getContent()+"', "
            + "null "
            + ")";
        }else{
            SQL = "INSERT remark (student_ID, staff_No, content, date) VALUES("
            + "@ID_STUDENT, "
            + r.getStaff().getNo()+ ", "
            + """+r.getContent()+"', "
            + """" + formatDate(r.getTime()) + """
            + ")";
        }
    }
    listSQL.add(SQL);
}
if(oldStudent.getRemarks()!=null && oldStudent.getRemarks().size() >0){
for(Remark r:oldStudent.getRemarks()){
    System.out.println(r.getID());
    boolean flag=false;
    for(Remark re:remarks){
        System.out.println(re.getID());
        if(r.getID()==re.getID()){
            flag=true;
            break;
        }
    }
}
}

```

```

        }
    }
    if(!flag){
        SQL="DELETE FROM remark WHERE ID="+r.getID();
        listSQL.add(SQL);
    }
}
}

return executeManySQL(listSQL);
}

@Override
public int removeStudent(int ID) {
    return SUCCESS;
}
@Override
public int deactiveStudent(int ID) {
    String SQL="UPDATE student SET status =0 WHERE ID="+ID;
    return executeUpdate(SQL);
}

@Override
public Student getStudentDetail(int ID) {
    Student student = null;
    int count = 0;
    String SQL;
    Connection connect = null;
    connect = ConnectDB.getConnect();
    if (connect == null) {
        return null;
    }
    SQL = "SELECT ID, No, academic_register_No, name,birthday, address, phone, email,
gender, admission_ID "
        + "FROM student WHERE ID=" + ID;

    ResultSet res = null;
    try {
        res = connect.createStatement().executeQuery(SQL);
        if (res.next()) {
            student = new Student(res.getInt(++count), res.getString(++count), res.getInt(++count),
res.getString(++count), parseDate(res.getString(++count)), res.getString(++count),
res.getString(++count), res.getString(++count), res.getInt(++count), res.getInt(++count));
        }
        //Load Admission infomation
    }
}

```

```

if (student.getAdmission() != null) {
    SQL = "SELECT prev_Institute, reason_Leaving, date FROM admission WHERE ID=" +
    student.getAdmission().getID();
    res = connect.createStatement().executeQuery(SQL);
    if (res.next()) {
        student.getAdmission().setPrev_Institute(res.getString(1));
        student.getAdmission().setReason_Leaving(res.getString(2));
        student.getAdmission().setTime(parseDate(res.getString(3)));
    }
}
//Load Academic infomation
if (student.getAcademic() != null) {
    SQL = "SELECT batch.ID, batch.name, batch.date FROM batch, academic WHERE
batch.ID=academic.batch_ID AND academic.register_No=" +
    student.getAcademic().getRegisterNo();
    res = connect.createStatement().executeQuery(SQL);
    if (res.next()) {
        student.getAcademic().setBatch(new Batch(res.getInt(1), res.getString(2),
parseDate(res.getString(3))));
    }
}
//Load remarks
SQL = "SELECT remark.ID, remark.content, remark.date, remark.staff_No, staff.name
FROM remark, staff WHERE staff.No=remark.staff_No AND student_ID=" + student.getID();
res = connect.createStatement().executeQuery(SQL);
List<Remark> remarks = new ArrayList<>();
while (res.next()) {
    Remark remark = new Remark(res.getInt(1), res.getString(2),
parseDate(res.getString(3)), new Staff(res.getInt(4), res.getString(5)));
    remarks.add(remark);
}
if (remarks.size() > 0) {
    student.setRemarks(remarks);
}
//Load Marks
SQL = "SELECT subject.ID, subject.name, mark.ID, mark.point, mark.date FROM mark,
subject WHERE subject.ID=subject_ID AND mark.student_ID=" + student.getID();
res = connect.createStatement().executeQuery(SQL);
while (res.next()) {
    Subject subject = new Subject(res.getInt(1), res.getString(2));
    Mark mark = new Mark(res.getInt(3), res.getInt(4), parseDate(res.getString(5)),
subject);
    student.addMark(mark);
}
//Load Extra Curricular
//Load Family

```

```

SQL = "SELECT ID FROM family_student WHERE student_ID=" + student.getID();
res = connect.createStatement().executeQuery(SQL);
FamilyStudent familyStudent = null;
if (res.next()) {
    familyStudent = new FamilyStudent(res.getInt(1));
}
if (familyStudent != null) {
    SQL = "SELECT ID, relation, name, phone, address, occupation, gender FROM
relatives_student WHERE family_student_ID=" + familyStudent.getID();
    res = connect.createStatement().executeQuery(SQL);
    while (res.next()) {
        count = 0;
        Relative relatives = new Relative(res.getInt(++count), res.getString(++count),
res.getString(++count), res.getString(++count), res.getString(++count), res.getString(++count),
res.getInt(++count));
        familyStudent.addRelative(relatives);
    }
    student.setFamilyStudent(familyStudent);
}
} catch (SQLException ex) {
    Logger.getLogger(StudentManager.class.getName()).log(Level.SEVERE, null, ex);
    return null;
} finally {
    try {
        connect.close();
    } catch (SQLException ex) {
        Logger.getLogger(StudentManager.class.getName()).log(Level.SEVERE, null, ex);
    }
}
return student;
}

@Override
public String getStudentList(int page) {
    String SQL = "SELECT student.ID, student.academic_register_No, student.No,
student.name,batch.name AS Batch, student.phone, student.email, student.gender "
    + "FROM student LEFT JOIN (academic LEFT JOIN batch ON
academic.batch_ID=batch.ID) ON academic.register_No=student.academic_register_No "
    + " WHERE student.status=1";
    return SQL;
}

@Override
public int addBatch(String[] data) {

```

```

String SQL="";
if(data[1].isEmpty()){
    SQL= "INSERT batch(name, date) values("
        + "" + data[0] + ", "
        + "null"
        + ") ";
} else{
    SQL = "INSERT batch(name, date) values("
        + "" + data[0] + ", "
        + ""+data[1]+ ""
        + ") ";
}

return executeUpdate(SQL);
}

@Override
public int editBatch(String[] data) {
    String SQL ="";
    if(data[2].isEmpty()){
        SQL="UPDATE batch SET "
            + "name=" + data[1] + ", "
            + "date=null "
            + "WHERE ID=" + data[0];
    } else{
        SQL="UPDATE batch SET "
            + "name=" + data[1] + ", "
            + "date=" + data[2] + " "
            + "WHERE ID=" + data[0];
    }

    return executeUpdate(SQL);
}

@Override
public String removeBatch(int ID) {
    String SQL;
    SQL="SELECT 1 FROM academic WHERE batch_ID=" + ID;
    if(getIDentity(SQL)==-1){
        SQL= "DELETE FROM batch WHERE ID=" + ID;
    } else{
        return "Have some student in batch ! You must remove or change students !";
    }
    int res=executeUpdate(SQL);
    if(res==CONNECT_FAIL){
        return "Connect fail ! Please try again !";
    }
}

```

```
        }
        if(res==FAIL){
            return "Have some error !";
        }
        if(res==SUCCESS){
            return "Success !";
        }
        return "Unknown !";
    }

@Override
public int addExtraCurricular(String[] data) {
    String SQL = "INSERT extra_curricular(name) values("
        + "" + data[0] + ""
        + ") ";

    return executeUpdate(SQL);
}

@Override
public int removeExtraCurricular(int ID) {
    String SQL = "DELETE FROM extra_curricular WHERE ID=" + ID;
    return executeUpdate(SQL);
}

@Override
public int editExtraCurricular(String[] data) {
    String SQL = "UPDATE extra_curricular SET "
        + "name=" + data[1] + " "
        + "WHERE ID=" + data[0];

    return executeUpdate(SQL);
}

@Override
public int addSubject(String[] data) {
    String SQL = "INSERT subject(name) values("
        + "" + data[0] + ""
        + ") ";

    return executeUpdate(SQL);
}

@Override
public String removeSubject(int ID) {
    String SQL ="";
```

```
SQL="SELECT 1 FROM mark WHERE subject_ID="+ID;
if(getIDEentity(SQL)==-1){
    SQL="DELETE FROM subject WHERE ID=" + ID;
}else{
    return "Have some student have this subject. You must remove subject in student info";
}
int res=executeUpdate(SQL);
if(res==CONNECT_FAIL){
    return "Connect fail ! Please try again !";
}
if(res==FAIL){
    return "Have some error !";
}
if(res==SUCCESS){
    return "Success !";
}
return "Unknown !";
}

@Override
public int editSubject(String[] data) {
    String SQL = "UPDATE subject SET "
        + "name='" + data[1] + "'"
        + "WHERE ID=" + data[0];
    return executeUpdate(SQL);
}

public Mark getMarkDetail(int ID) {
    Mark mark = new Mark(ID);

    return mark;
}

@Override
public int deactiveBatch(int ID) {
    String SQL="UPDATE batch SET status=0 WHERE ID="+ID;
    return executeUpdate(SQL);
}

@Override
public int deactiveExtraCurricular(int ID) {
    String SQL="UPDATE extra_curricular SET status=0 WHERE ID="+ID;
    return executeUpdate(SQL);
}

@Override
```

```

public int deactivateSubject(int ID) {
    String SQL="UPDATE subject SET status=0 WHERE ID="+ID;
    return executeUpdate(SQL);
}
}

```

- **For User Validations and verification**

```

/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */
package Controls;

import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JComboBox;

/***
 *
 * @author Aeon(Ankit RudrAshokJha)
 */
public class Validate {
    public static boolean isEmpty(String value){
        if(value==null || value.isEmpty()){
            return true;
        }
        return false;
    }
    public static boolean checkDate(String str){
        if(isEmpty(str)){
            return false;
        }
        Date date = null;
        SimpleDateFormat format = new SimpleDateFormat("yyyy-MM-dd");
        try {
            date = format.parse(str);
        } catch (ParseException ex) {
            Logger.getLogger(Validate.class.getName()).log(Level.SEVERE, null, ex);
            return false;
        }
        return true;
    }
}

```

```
public static boolean isNumber(String str){  
    if(isEmpty(str)){  
        return false;  
    }  
    int res=0;  
    try{  
        res=Integer.parseInt(str);  
    }catch(NumberFormatException ex){  
        return false;  
    }  
    return true;  
}  
public static boolean checkEmail(String str){  
    return true;  
}  
public static boolean checkPoint(String str){  
    if(!isNumber(str)){  
        return false;  
    }  
    int point=0;  
    try{  
        point=Integer.parseInt(str);  
    }catch(NumberFormatException ex){  
        return false;  
    }  
    if(point <0 || point >10){  
        return false;  
    }  
    return true;  
}  
public static boolean choosedComboBox(JComboBox combo){  
    if(combo.getSelectedIndex()<=0){  
        return false;  
    }else{  
        return true;  
    }  
}  
public static boolean checkPassword(String pswd1, String pswd2){  
    if(!pswd1.equalsIgnoreCase(pswd2)){  
        return false;  
    }  
    return true;  
}  
}
```

- For User Listing Page

```
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */
package AdminMan;

import Controls.Admin;
import Controls.Manager;
import Controls.Validate;
import java.util.Vector;
import javax.swing.JOptionPane;
import javax.swing.plaf.basic.BasicInternalFrameUI;
import MainHome.UIInterface;

/**
 *
 * @author Aeon(Ankit RudrAshokJha)
 */
public class ListUser extends javax.swing.JInternalFrame implements UIInterface{
    Admin admin=new Admin();
    private int act;

    public ListUser(int w, int h){
        this.admin=admin;
        initComponents();
        initUI();
        setSize(w, h);
    }
    @Override
    public void initUI(){
        enableButton(false);
        ((BasicInternalFrameUI)this.getUI()).setNorthPane(null);
        editAddPanel.setVisible(false);
        String SQL="SELECT ID, usrn, pswd, name, role, email FROM user";
        admin.loadTable(SQL, tblUser, Manager.USER_MODEL);
    }
    @Override
    public void enableEditAddPanel(boolean b) {
        if(b==false){
            tblUser.setEnabled(true);
            txtName.setText(null);
            editAddPanel.setVisible(false);
        }
    }
}
```

```
        bannerPanel.setVisible(true);
    }else{
        if(act==EDIT){
            labelAction.setText("Edit User");
            loadEdit();
        }else if(act==ADD){
            labelAction.setText("Add User");
            loadEdit();
        }
        tblUser.setEnabled(false);
        editAddPanel.setVisible(true);
        btnReset.setVisible(true);

        bannerPanel.setVisible(false);
    }
}

@Override
public void loadEdit() {
    if(act==EDIT){
        Vector row=admin.getSelectRow(tblUser);
        txtUsername.setText(row.get(1).toString());
        txtPassword.setText(row.get(2).toString());
        txtName.setText(row.get(3).toString());
        comboType.setSelectedItem(row.get(4).toString());
        txtEmail.setText(row.get(5).toString());
    }else if(act==ADD){
        txtUsername.setText("");
        txtPassword.setText("");
        txtName.setText("");
        comboType.setSelectedIndex(0);
        txtEmail.setText("");
    }
    txtUsername.setRequestFocusEnabled(true);
}

public void add() {
    String username=txtUsername.getText();
    String password=txtPassword.getText();
    String role=comboType.getSelectedItem().toString();
    String name=txtName.getText();
    String email=txtEmail.getText();
```

```

//Check Unique
if(!admin.checkUnique("usrn", "user", username)){
    JOptionPane.showMessageDialog(this, "Username must be unique !");
    return;
}
int roleUser=admin.parseType(role);
if(roleUser ==-1){
    JOptionPane.showMessageDialog(this, "Role incorrect !");
    return;
}
String data[]={username, password, name, roleUser+"", email};
int res=admin.addUser(data);
if(res==Manager.SUCCESS){
    initUI();
    enableEditAddPanel(false);
}
admin.showMessage(res, this);
}

public void save(){
    if(!validateForm()){
        return;
    }
    if(act==ADD){
        add();
        return;
    }
    if(act==EDIT){
        edit();
        return;
    }
}

public void edit() {
    int ID=admin.getSelectID(tblUser);
    String username=txtUsername.getText();
    String password=txtPassword.getText();
    String role=comboType.getSelectedItem().toString();
    String name=txtName.getText();
    String email=txtEmail.getText();
    int roleUser=admin.parseType(role);

    String data[]={admin.getSelectID(tblUser)+"", username, password, name,
    roleUser+"", email};
    int res=admin.editUser(data);
}

```

```

admin.showMessage(res, this);
if(res==Manager.SUCCESS){
    initUI();
}
enableEditAddPanel(false);
}

@Override
public void deactivate() {
    int res=admin.deactiveUser(admin.getSelectID(tblUser));
    admin.showMessage(res, this);
    if(res==Manager.SUCCESS){
        initUI();
    }
}

@Override
public void search(){
    String SQL="SELECT ID, usrn, pswd, name, role, email FROM user WHERE 2>1 ";
    if(!txtNameSearch.getText().isEmpty()){
        SQL+=" AND name like '%"+txtNameSearch.getText()+"%'";
    }
    if(!txtUsernameSearch.getText().isEmpty()){
        SQL+=" AND usrn like '%"+txtUsernameSearch.getText()+"%'";
    }
    if(comboTypeSearch.getSelectedIndex() !=0){
        if(comboTypeSearch.getSelectedItem().equals("Staff manager")){
            SQL+=" AND role=2";
        }else if(comboTypeSearch.getSelectedItem().equals("Student manager")){
            SQL+=" AND role=3";
        }
    }
    admin.loadTable(SQL, tblUser, Manager.USER_MODEL);
    enableButton(false);
}
/***
 * 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">
private void initComponents() {

    bannerPanel = new javax.swing.JPanel();

```

```
jLabel2 = new javax.swing.JLabel();
jLabel3 = new javax.swing.JLabel();
txtUsernameSearch = new javax.swing.JTextField();
jLabel4 = new javax.swing.JLabel();
txtNameSearch = new javax.swing.JTextField();
jLabel5 = new javax.swing.JLabel();
comboTypeSearch = new javax.swing.JComboBox();
jLabel12 = new javax.swing.JLabel();
jLabel13 = new javax.swing.JLabel();
btnAdd = new javax.swing.JLabel();
btnEdit = new javax.swing.JLabel();
btnDeactivate = new javax.swing.JLabel();
jScrollPane1 = new javax.swing.JScrollPane();
tblUser = new javax.swing.JTable();
jLabel10 = new javax.swing.JLabel();
editAddPanel = new javax.swing.JPanel();
labelAction = new javax.swing.JLabel();
jLabel1 = new javax.swing.JLabel();
jLabel6 = new javax.swing.JLabel();
txtUsername = new javax.swing.JTextField();
txtPassword = new javax.swing.JTextField();
jLabel7 = new javax.swing.JLabel();
txtName = new javax.swing.JTextField();
jLabel8 = new javax.swing.JLabel();
comboType = new javax.swing.JComboBox();
jLabel9 = new javax.swing.JLabel();
txtEmail = new javax.swing.JTextField();
txtEmail1 = new javax.swing.JTextField();
jLabel11 = new javax.swing.JLabel();
jLabel14 = new javax.swing.JLabel();
jLabel15 = new javax.swing.JLabel();
jLabel16 = new javax.swing.JLabel();
comboType1 = new javax.swing.JComboBox();
txtEmail3 = new javax.swing.JTextField();
txtEmail4 = new javax.swing.JTextField();
jLabel17 = new javax.swing.JLabel();
bthCancle = new javax.swing.JLabel();
btnSave = new javax.swing.JLabel();
btnReset = new javax.swing.JLabel();
dPic = new com.toedter.calendar.JDateChooser();

setBackground(new java.awt.Color(179, 209, 224));
setBorder(null);
setPreferredSize(new java.awt.Dimension(1024, 700));
getContentPane().setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());
```

```
bannerPanel.setBackground(new java.awt.Color(45, 129, 171));
bannerPanel.setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

jLabel2.setBackground(new java.awt.Color(1, 100, 151));
jLabel2.setFont(new java.awt.Font("Times New Roman", 1, 24)); // NOI18N
jLabel2.setForeground(new java.awt.Color(255, 255, 255));
jLabel2.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
jLabel2.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/Images/icons8_User_Account_38px.png"))); // NOI18N
jLabel2.setText("Users Management");
jLabel2.setOpaque(true);
bannerPanel.add(jLabel2, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 0, 1040,
60));

jLabel3.setBackground(new java.awt.Color(0, 102, 153));
jLabel3.setFont(new java.awt.Font("Segoe UI", 1, 12)); // NOI18N
jLabel3.setForeground(new java.awt.Color(255, 255, 255));
jLabel3.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
jLabel3.setText("Username");
jLabel3.setOpaque(true);
bannerPanel.add(jLabel3, new org.netbeans.lib.awtextra.AbsoluteConstraints(240, 110, 90,
25));

txtUsernameSearch.setBackground(new java.awt.Color(85, 153, 187));
txtUsernameSearch.setFont(new java.awt.Font("Segoe UI", 1, 12)); // NOI18N
txtUsernameSearch.setForeground(new java.awt.Color(255, 255, 255));
txtUsernameSearch.setBorder(null);
txtUsernameSearch.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        searchAction(evt);
    }
});
bannerPanel.add(txtUsernameSearch, new
org.netbeans.lib.awtextra.AbsoluteConstraints(330, 110, 220, 25));

jLabel4.setBackground(new java.awt.Color(0, 102, 153));
jLabel4.setFont(new java.awt.Font("Segoe UI", 1, 12)); // NOI18N
jLabel4.setForeground(new java.awt.Color(255, 255, 255));
jLabel4.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
jLabel4.setText("Name");
jLabel4.setOpaque(true);
bannerPanel.add(jLabel4, new org.netbeans.lib.awtextra.AbsoluteConstraints(570, 110, 90,
25));

txtNameSearch.setBackground(new java.awt.Color(85, 153, 187));
```

```
txtNameSearch.setFont(new java.awt.Font("Segoe UI", 1, 12)); // NOI18N
txtNameSearch.setForeground(new java.awt.Color(255, 255, 255));
txtNameSearch.setBorder(null);
txtNameSearch.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        searchAction(evt);
    }
});
bannerPanel.add(txtNameSearch, new org.netbeans.lib.awtextra.AbsoluteConstraints(660, 110, 180, 25));

jLabel5.setBackground(new java.awt.Color(0, 102, 153));
jLabel5.setFont(new java.awt.Font("Segoe UI", 1, 12)); // NOI18N
jLabel5.setForeground(new java.awt.Color(255, 255, 255));
jLabel5.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
jLabel5.setText("Role");
jLabel5.setOpaque(true);
bannerPanel.add(jLabel5, new org.netbeans.lib.awtextra.AbsoluteConstraints(240, 150, 90, 25));

comboTypeSearch.setBackground(new java.awt.Color(85, 153, 187));
comboTypeSearch.setEditable(true);
comboTypeSearch.setModel(new javax.swing.DefaultComboBoxModel(new String[] {
"All", "Staff manager", "Student manager" }));
comboTypeSearch.setBorder(null);
bannerPanel.add(comboTypeSearch, new
org.netbeans.lib.awtextra.AbsoluteConstraints(330, 150, 220, 25));

jLabel12.setBackground(new java.awt.Color(0, 102, 255));
jLabel12.setFont(new java.awt.Font("Segoe UI", 1, 12)); // NOI18N
jLabel12.setForeground(new java.awt.Color(255, 255, 255));
jLabel12.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
jLabel12.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/Images/icons8_Refresh_17px.png")));
// NOI18N
jLabel12.setText("Reset");
jLabel12.setOpaque(true);
jLabel12.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        jLabel12MouseClicked(evt);
    }
});
bannerPanel.add(jLabel12, new org.netbeans.lib.awtextra.AbsoluteConstraints(750, 150, 88, 25));

jLabel13.setBackground(new java.awt.Color(255, 102, 0));
```

```
jLabel13.setFont(new java.awt.Font("Segoe UI", 1, 12)); // NOI18N
jLabel13.setForeground(new java.awt.Color(255, 255, 255));
jLabel13.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
jLabel13.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/Images/icons8_Google_Web_Search_17px.png
")));
// NOI18N
jLabel13.setText("Search");
jLabel13.setOpaque(true);
jLabel13.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        jLabel13MouseClicked(evt);
    }
});
bannerPanel.add(jLabel13, new org.netbeans.lib.awtextra.AbsoluteConstraints(570, 150, 90,
25));

btnAdd.setBackground(new java.awt.Color(51, 204, 0));
btnAdd.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N
btnAdd.setForeground(new java.awt.Color(255, 255, 255));
btnAdd.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
btnAdd.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/Images/icons8_Add_New_17px.png")));
// NOI18N
btnAdd.setText("Add");
btnAdd.setOpaque(true);
btnAdd.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        btnAddMouseClicked(evt);
    }
});
bannerPanel.add(btnAdd, new org.netbeans.lib.awtextra.AbsoluteConstraints(90, 490, 88,
26));

btnEdit.setBackground(new java.awt.Color(204, 0, 204));
btnEdit.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N
btnEdit.setForeground(new java.awt.Color(255, 255, 255));
btnEdit.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
btnEdit.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/Images/icons8_Edit_Row_17px.png")));
// NOI18N
btnEdit.setText("Edit");
btnEdit.setOpaque(true);
btnEdit.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        btnEditMouseClicked(evt);
    }
});
```



```

        {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 [] {
        "ID", "Username", "Password", "Name", "Role", "Email"
    }
) {
    boolean[] canEdit = new boolean [] {
        false, true, true, false, true, true
    };

    public boolean isCellEditable(int rowIndex, int columnIndex) {
        return canEdit [columnIndex];
    }
});
tblUser.setRowHeight(18);
tblUser.setSelectionBackground(new java.awt.Color(0, 153, 51));
tblUser.setShowHorizontalLines(false);
tblUser.setShowVerticalLines(false);
tblUser.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        tblUserMouseClicked(evt);
    }
});
jScrollPane1.setViewportView(tblUser);

bannerPanel.add(jScrollPane1, new org.netbeans.lib.awtextra.AbsoluteConstraints(90, 250,
830, 210));

jLabel10.setBackground(new java.awt.Color(255, 101, 101));
jLabel10.setFont(new java.awt.Font("Times New Roman", 1, 24)); // NOI18N
jLabel10.setForeground(new java.awt.Color(255, 255, 255));
jLabel10.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
jLabel10.setText("All Users List ");
jLabel10.setOpaque(true);
bannerPanel.add(jLabel10, new org.netbeans.lib.awtextra.AbsoluteConstraints(90, 210, 830,
40));

getContentPane().add(bannerPanel, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 0,
1014, 700));

editAddPanel.setBackground(new java.awt.Color(45, 129, 171));
editAddPanel.setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

```

```
labelAction.setBackground(new java.awt.Color(0, 102, 153));
labelAction.setFont(new java.awt.Font("Ubuntu", 1, 24)); // NOI18N
labelAction.setForeground(new java.awt.Color(255, 255, 255));
labelAction.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
labelAction.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/Images/icons8_User_Account_38px.png"))); // NOI18N
labelAction.setText("Add New User");
labelAction.setOpaque(true);
editAddPanel.add(labelAction, new org.netbeans.lib.awtextra.AbsoluteConstraints(-3, 0,
1040, 60));

jLabel1.setBackground(new java.awt.Color(0, 102, 153));
jLabel1.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N
jLabel1.setForeground(new java.awt.Color(255, 255, 255));
jLabel1.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
jLabel1.setText("Username");
jLabel1.setOpaque(true);
editAddPanel.add(jLabel1, new org.netbeans.lib.awtextra.AbsoluteConstraints(190, 110, 90,
25));

jLabel6.setBackground(new java.awt.Color(0, 102, 153));
jLabel6.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N
jLabel6.setForeground(new java.awt.Color(255, 255, 255));
jLabel6.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
jLabel6.setText("Password");
jLabel6.setOpaque(true);
editAddPanel.add(jLabel6, new org.netbeans.lib.awtextra.AbsoluteConstraints(190, 140, 90,
25));

txtUsername.setBackground(new java.awt.Color(85, 153, 187));
txtUsername.setForeground(new java.awt.Color(255, 255, 255));
txtUsername.setBorder(null);
editAddPanel.add(txtUsername, new org.netbeans.lib.awtextra.AbsoluteConstraints(280,
110, 250, 25));

txtPassword.setBackground(new java.awt.Color(85, 153, 187));
txtPassword.setForeground(new java.awt.Color(255, 255, 255));
txtPassword.setBorder(null);
editAddPanel.add(txtPassword, new org.netbeans.lib.awtextra.AbsoluteConstraints(280, 140,
250, 25));

jLabel7.setBackground(new java.awt.Color(0, 102, 153));
jLabel7.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N
jLabel7.setForeground(new java.awt.Color(255, 255, 255));
jLabel7.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
```

```
jLabel7.setText("Name");
jLabel7.setOpaque(true);
editAddPanel.add(jLabel7, new org.netbeans.lib.awtextra.AbsoluteConstraints(190, 170, 90,
25));

txtName.setBackground(new java.awt.Color(85, 153, 187));
txtName.setForeground(new java.awt.Color(255, 255, 255));
txtName.setBorder(null);
editAddPanel.add(txtName, new org.netbeans.lib.awtextra.AbsoluteConstraints(280, 170,
250, 25));

jLabel8.setBackground(new java.awt.Color(0, 102, 153));
jLabel8.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N
jLabel8.setForeground(new java.awt.Color(255, 255, 255));
jLabel8.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
jLabel8.setText("Role");
jLabel8.setOpaque(true);
editAddPanel.add(jLabel8, new org.netbeans.lib.awtextra.AbsoluteConstraints(570, 110, 90,
25));

comboType.setBackground(new java.awt.Color(85, 153, 187));
comboType.setModel(new javax.swing.DefaultComboBoxModel(new String[] { "Choose",
"Admin", "Staff manager", "Student manager" }));
comboType.setBorder(null);
editAddPanel.add(comboType, new org.netbeans.lib.awtextra.AbsoluteConstraints(660, 110,
250, 25));

jLabel9.setBackground(new java.awt.Color(0, 102, 153));
jLabel9.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N
jLabel9.setForeground(new java.awt.Color(255, 255, 255));
jLabel9.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
jLabel9.setText("Email");
jLabel9.setOpaque(true);
editAddPanel.add(jLabel9, new org.netbeans.lib.awtextra.AbsoluteConstraints(570, 140, 90,
25));

txtEmail.setBackground(new java.awt.Color(85, 153, 187));
txtEmail.setForeground(new java.awt.Color(255, 255, 255));
txtEmail.setBorder(null);
editAddPanel.add(txtEmail, new org.netbeans.lib.awtextra.AbsoluteConstraints(660, 140,
250, 25));

txtEmail1.setBackground(new java.awt.Color(85, 153, 187));
txtEmail1.setForeground(new java.awt.Color(255, 255, 255));
txtEmail1.setBorder(null);
```

```
editAddPanel.add(txtEmail1, new org.netbeans.lib.awtextra.AbsoluteConstraints(660, 200, 250, 25));

jLabel11.setBackground(new java.awt.Color(0, 102, 153));
jLabel11.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N
jLabel11.setForeground(new java.awt.Color(255, 255, 255));
jLabel11.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
jLabel11.setText("Emp.ID");
jLabel11.setOpaque(true);
editAddPanel.add(jLabel11, new org.netbeans.lib.awtextra.AbsoluteConstraints(190, 230, 90, 25));

jLabel14.setBackground(new java.awt.Color(0, 102, 153));
jLabel14.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N
jLabel14.setForeground(new java.awt.Color(255, 255, 255));
jLabel14.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
jLabel14.setText("Security Ques");
jLabel14.setOpaque(true);
editAddPanel.add(jLabel14, new org.netbeans.lib.awtextra.AbsoluteConstraints(570, 170, 90, 25));

jLabel15.setBackground(new java.awt.Color(0, 102, 153));
jLabel15.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N
jLabel15.setForeground(new java.awt.Color(255, 255, 255));
jLabel15.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
jLabel15.setText("Ans");
jLabel15.setOpaque(true);
editAddPanel.add(jLabel15, new org.netbeans.lib.awtextra.AbsoluteConstraints(570, 200, 90, 25));

jLabel16.setBackground(new java.awt.Color(0, 102, 153));
jLabel16.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N
jLabel16.setForeground(new java.awt.Color(255, 255, 255));
jLabel16.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
jLabel16.setText("D.O.B");
jLabel16.setOpaque(true);
editAddPanel.add(jLabel16, new org.netbeans.lib.awtextra.AbsoluteConstraints(190, 200, 90, 25));

comboType1.setBackground(new java.awt.Color(85, 153, 187));
comboType1.setModel(new javax.swing.DefaultComboBoxModel(new String[] { "What is Your 1st Pets Name ?", "What is Your the name of your first school ?", "What is your favorite Book ?", "What is your eternal desire ?", "What is the name of your best friend ? ", " " }));
comboType1.setSelectedItem(null);
comboType1.setBorder(null);
```

```
editAddPanel.add(comboType1, new org.netbeans.lib.awtextra.AbsoluteConstraints(660, 170, 250, 25));

txtEmail3.setBackground(new java.awt.Color(85, 153, 187));
txtEmail3.setForeground(new java.awt.Color(255, 255, 255));
txtEmail3.setBorder(null);
editAddPanel.add(txtEmail3, new org.netbeans.lib.awtextra.AbsoluteConstraints(280, 230, 250, 25));

txtEmail4.setBackground(new java.awt.Color(85, 153, 187));
txtEmail4.setForeground(new java.awt.Color(255, 255, 255));
txtEmail4.setBorder(null);
editAddPanel.add(txtEmail4, new org.netbeans.lib.awtextra.AbsoluteConstraints(660, 230, 250, 25));

jLabel17.setBackground(new java.awt.Color(0, 102, 153));
jLabel17.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N
jLabel17.setForeground(new java.awt.Color(255, 255, 255));
jLabel17.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
jLabel17.setText("Mob.");
jLabel17.setOpaque(true);
editAddPanel.add(jLabel17, new org.netbeans.lib.awtextra.AbsoluteConstraints(570, 230, 90, 25));

bthCancle.setBackground(new java.awt.Color(255, 0, 0));
bthCancle.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N
bthCancle.setForeground(new java.awt.Color(255, 255, 255));
bthCancle.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
bthCancle.setIcon(new javax.swing.ImageIcon(getClass().getResource("/Images/icons8_Close_Window_17px.png"))); // NOI18N
bthCancle.setText("Cancel");
bthCancle.setOpaque(true);
bthCancle.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        bthCancleMouseClicked(evt);
    }
});
editAddPanel.add(bthCancle, new org.netbeans.lib.awtextra.AbsoluteConstraints(820, 270, 90, 25));

btnSave.setBackground(new java.awt.Color(51, 204, 0));
btnSave.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N
btnSave.setForeground(new java.awt.Color(255, 255, 255));
btnSave.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
```

```
        btnSave.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/Images/icons8_Save_17px.png"))); // NOI18N
        btnSave.setText("Save");
        btnSave.setOpaque(true);
        btnSave.addMouseListener(new java.awt.event.MouseAdapter() {
            public void mouseClicked(java.awt.event.MouseEvent evt) {
                btnSaveMouseClicked(evt);
            }
        });
        editAddPanel.add(btnSave, new org.netbeans.lib.awtextra.AbsoluteConstraints(500, 270, 90,
25));

        btnReset.setBackground(new java.awt.Color(0, 102, 255));
        btnReset.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N
        btnReset.setForeground(new java.awt.Color(255, 255, 255));
        btnReset.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
        btnReset.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/Images/icons8_Refresh_17px.png"))); // NOI18N
        btnReset.setText("Reset");
        btnReset.setOpaque(true);
        btnReset.addMouseListener(new java.awt.event.MouseAdapter() {
            public void mouseClicked(java.awt.event.MouseEvent evt) {
                btnResetMouseClicked(evt);
            }
        });
        editAddPanel.add(btnReset, new org.netbeans.lib.awtextra.AbsoluteConstraints(190, 270,
90, 25));
        editAddPanel.add(dPic, new org.netbeans.lib.awtextra.AbsoluteConstraints(280, 200, 250,
25));

        getContentPane().add(editAddPanel, new org.netbeans.lib.awtextra.AbsoluteConstraints(0,
0, 1014, 700));

        pack();
}// </editor-fold>

private void searchAction(java.awt.event.ActionEvent evt) {
    search();
}

private void tblUserMouseClicked(java.awt.event.MouseEvent evt) {
    enableButton(true);
}

private void jLabel13MouseClicked(java.awt.event.MouseEvent evt) {
```

```
search(); // TODO add your handling code here:  
}  
  
private void jLabel12MouseClicked(java.awt.event.MouseEvent evt) {  
    txtNameSearch.setText("");  
    txtUsernameSearch.setText("");  
    comboTypeSearch.setSelectedIndex(0); // TODO add your handling code here:  
}  
  
private void btnAddMouseClicked(java.awt.event.MouseEvent evt) {  
    act=ADD;  
    enableEditAddPanel(true);  
    btnReset.setVisible(false); // TODO add your handling code here:  
}  
  
private void btnEditMouseClicked(java.awt.event.MouseEvent evt) {  
    if(admin.getSelectedRow(tblUser) !=null){  
        act=EDIT;  
        enableEditAddPanel(true);  
    } // TODO add your handling code here:  
}  
  
private void btnDeactivateMouseClicked(java.awt.event.MouseEvent evt) {  
    if(admin.confirmRemove(this, "Are you sure ?")==0){  
        deactivate();  
    } // TODO add your handling code here:  
}  
  
private void btnSaveMouseClicked(java.awt.event.MouseEvent evt) {  
    save(); // TODO add your handling code here:  
}  
  
private void btnResetMouseClicked(java.awt.event.MouseEvent evt) {  
    // TODO add your handling code here:  
}  
  
private void bthCancleMouseClicked(java.awt.event.MouseEvent evt) {  
    enableEditAddPanel(false); // TODO add your handling code here:  
}  
  
// Variables declaration - do not modify  
private javax.swing.JPanel bannerPanel;  
private javax.swing.JLabel bthCancle;  
private javax.swing.JLabel btnAdd;  
private javax.swing.JLabel btnDeactivate;  
private javax.swing.JLabel btnEdit;
```

```
private javax.swing.JLabel btnReset;
private javax.swing.JLabel btnSave;
private javax.swing.JComboBox comboType;
private javax.swing.JComboBox comboType1;
private javax.swing.JComboBox comboBoxSearch;
private com.toedter.calendar.JDateChooser dPic;
private javax.swing.JPanel editAddPanel;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel10;
private javax.swing.JLabel jLabel11;
private javax.swing.JLabel jLabel12;
private javax.swing.JLabel jLabel13;
private javax.swing.JLabel jLabel14;
private javax.swing.JLabel jLabel15;
private javax.swing.JLabel jLabel16;
private javax.swing.JLabel jLabel17;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel4;
private javax.swing.JLabel jLabel5;
private javax.swing.JLabel jLabel6;
private javax.swing.JLabel jLabel7;
private javax.swing.JLabel jLabel8;
private javax.swing.JLabel jLabel9;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JLabel labelAction;
private javax.swing.JTable tblUser;
private javax.swing.JTextField txtEmail;
private javax.swing.JTextField txtEmail1;
private javax.swing.JTextField txtEmail3;
private javax.swing.JTextField txtEmail4;
private javax.swing.JTextField txtName;
private javax.swing.JTextField txtNameSearch;
private javax.swing.JTextField txtPassword;
private javax.swing.JTextField txtUsername;
private javax.swing.JTextField txtUsernameSearch;
// End of variables declaration

@Override
public void enableButton(boolean b) {
    btnEdit.setEnabled(b);
    btnDeactivate.setEnabled(b);
}

@Override
public boolean validateForm() {
```

```

String username=txtUsername.getText();
String password=txtPassword.getText();
String role=comboType.getSelectedItem().toString();
String name=txtName.getText();
String email=txtEmail.getText();
int roleUser=admin.parseType(role);

if(Validate.isEmpty(username)){
    JOptionPane.showMessageDialog(this, "Username is required !");
    return false;
}
if(Validate.isEmpty(password)){
    JOptionPane.showMessageDialog(this, "Password is required !");
    return false;
}
if(Validate.isEmpty(name)){
    JOptionPane.showMessageDialog(this, "Name is required !");
    return false;
}

if(!Validate.choosedComboBox(comboType)){
    JOptionPane.showMessageDialog(this, "Role is required !");
    return false;
}

if(roleUser ==-1){
    JOptionPane.showMessageDialog(this, "Type User incorrect !");
    return false;
}
return true;
}

}

```

5.1.2. Codes for Database Integration

- **Database Connection Class**

```

/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */
package DataSett;

import java.sql.Connection;

```

```

import java.sql.DriverManager;
import java.sql.SQLException;
import java.util.logging.Level;
import java.util.logging.Logger;

/**
 *
 * @author Aeon(Ankit RudrAshokJha)
 */
public class ConnectDB {
    public static Connection getConnect(){
        SettingsDB sdb= SettingsDB.readSettings();
        if(sdb==null){
            return null;
        }
        Connection connect=null;
        try {
            Class.forName("com.mysql.jdbc.Driver");

connect=DriverManager.getConnection("jdbc:mysql://" +sdb.getServer() + ":" +sdb.getPort() + "/Sch
oolManagement", sdb.getUsrn(), sdb.getPswd());
        } catch (ClassNotFoundException | SQLException ex) {
            Logger.getLogger(ConnectDB.class.getName()).log(Level.SEVERE, null, ex);
        }

        return connect;
    }
}

```

- **For Database Setting**

```

/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */
package DataSett;

import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.PrintWriter;
import java.util.Calendar;
import java.util.Properties;
import java.util.logging.Level;
import java.util.logging.Logger;

```

```

/**
 *
 * @author Aeon(Ankit RudrAshokJha)
 */
public class SettingsDB {
    private String server;
    private int port;
    private String usrn;
    private String pswd;
    public static SettingsDB readSettings(){
        SettingsDB sdb=new SettingsDB();
        Properties p = new Properties();
        try {
            FileInputStream fin = new FileInputStream("Config.cfg");
            p.load(fin);
        } catch (Exception e) {
            return null;
        }
        sdb.setServer(p.getProperty("Server"));
        sdb.setPort(Integer.parseInt(p.getProperty("Port")));
        sdb.setUsrn(p.getProperty("Username"));
        sdb.setPswd(p.getProperty("Password"));
        return sdb;
    }
    public static void writeSettings(String server, int port, String usrn, String pswd){
        Calendar ca = Calendar.getInstance();
        PrintWriter pw = null;
        try {
            pw = new PrintWriter("Config.cfg");
            pw.println("#" + ca.getTime().toString());
            pw.println("Server=" + server.trim());
            pw.println("Port=" + port);
            pw.println("Username=" + usrn.trim());
            pw.println("Password=" + pswd);
            pw.flush();
        } catch (FileNotFoundException ex) {
            Logger.getLogger(SettingsDB.class.getName()).log(Level.SEVERE, null, ex);
        } finally {
            pw.close();
        }
    }
    public String getServer() {
        return server;
    }
}

```

```

public int getPort() {
    return port;
}

public String getUsrn() {
    return usrn;
}

public String getPswd() {
    return pswd;
}

public void setServer(String server) {
    this.server = server;
}

public void setPort(int port) {
    this.port = port;
}

public void setUsrn(String usrn) {
    this.usrn = usrn;
}

public void setPswd(String pswd) {
    this.pswd = pswd;
}

}

```

5.1.3. Codes for Student Management Module

- **For Batch Management Module**

```

/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */
package StudentMan;

import FileExport.ExportFile;
import Controls.Manager;
import Controls.StudentManager;
import Controls.Validate;
import MainObjects.BatchEntity;

```

```
import javax.swing.JOptionPane;
import javax.swing.plaf.basic.BasicInternalFrameUI;
import TableModel.Batch;
import MainHome.UIInterface;

/**
 *
 * @author Aeon(Ankit RudrAshokJha)
 */
public class ListBatch extends javax.swing.JInternalFrame implements UIInterface{
    StudentManager studentManager=new StudentManager();
    private BatchEntity batchEntity=new BatchEntity();
    private Batch batch;
    private int act;
    /**
     * Creates new form ListBatch
     */
    public ListBatch() {
        initComponents();
    }
    public ListBatch(int w, int h) {
        initComponents();
        initUI();
        setSize(w, h);
    }
    @Override
    public void initUI(){
        enableButton(false);
        ((BasicInternalFrameUI)this.getUI()).setNorthPane(null);

        enableEditAddPanel(false);
        batchEntity.loadList(tblBatch);
    }
    @Override
    public void enableEditAddPanel(boolean b){
        if(b==false){
            tblBatch.setEnabled(true);
            txtName.setText("");
            dPic.setDate(null);

            editAddPanel.setVisible(false);

            bannerPanel.setVisible(true);
        }else{
            if(act==EDIT){
                labelAction.setText("Edit Batch");
            }
        }
    }
}
```

```

        loadEdit();
    }else if(act==ADD){
        labelAction.setText("Add Batch");
    }
    tblBatch.setEnabled(false);
    btnReset.setVisible(true);

    bannerPanel.setVisible(false);
    editAddPanel.setVisible(true);
}
}

public void save(){
    if(!validateForm()){
        return;
    }
    if(batch==null)
        batch=new Batch();
    batch.setName(txtName.getText());
    batch.setTime(dPic.getDate());
    batchEntity.setBatch(batch);
    int res=batchEntity.save();
    if(res==Manager.SUCCESS){
        initUI();
    }
    batchEntity.showMessage(res, this);
}
@Override
public void loadEdit(){
    batch=batchEntity.findById(batchEntity.getSelectID(tblBatch));
    txtName.setText(batch.getName());
    dPic.setDate(batch.getTime());
    txtName.setRequestFocusEnabled(true);
}

// public void add(){
//     String name=txtName.getText();
//     if(!studentManager.checkUnique("name", "batch", name)){
//         JOptionPane.showMessageDialog(this, "Name already exist !");
//         return;
//     }
//     String time=txtDate.getText();
//     String data[]=new String[]{ name, time };
//     Batch b=new Batch(name, batchEntity.parseDate(time));

```

```

//      batchEntity.setBatch(b);
//      int res=batchEntity.save();
//      if(res==Manager.SUCCESS){
//          initUI();
//          enableEditAddPanel(false);
//      }
//      studentManager.showMessage(res, this);
//  }

//  public void edit(){
//      String name=txtName.getText();
//      System.out.println(tblBatch.getValueAt(tblBatch.getSelectedRow(), 1));
//      if(!name.equalsIgnoreCase((String)tblBatch.getValueAt(tblBatch.getSelectedRow(), 1))){
//          if(!studentManager.checkUnique("name", "batch", name)){
//              JOptionPane.showMessageDialog(this, "Name already exist !");
//              return;
//          }
//      }
//      String time=txtDate.getText();
//
//      String data[]={studentManager.getSelectID(tblBatch)+"", name, time};
//      int res=studentManager.editBatch(data);
//      studentManager.showMessage(res, this);
//      initUI();
//      enableEditAddPanel(false);
//  }

@Override
public void deactivate(){
    int res=studentManager.deactiveBatch(studentManager.getSelectID(tblBatch));
    studentManager.showMessage(res, this);
    if(res==Manager.SUCCESS){
        initUI();
    }
}

@Override
public void search(){
    String SQL="SELECT ID, name, date FROM batch WHERE status=1 ";
    if(!txtNameSearch.getText().isEmpty()){
        SQL+=" AND name like '%"+txtNameSearch.getText()+"%'";
    }
    studentManager.loadTable(SQL, tblBatch, Manager.BATCH_MODEL);
    enableButton(false);
}

/**
 * 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">
private void initComponents() {

    editAddPanel = new javax.swing.JPanel();
    labelAction = new javax.swing.JLabel();
    jLabel1 = new javax.swing.JLabel();
    btnReset = new javax.swing.JLabel();
    txtName = new javax.swing.JTextField();
    dPic = new com.toedter.calendar.JDateChooser();
    jLabel7 = new javax.swing.JLabel();
    btnCancel = new javax.swing.JLabel();
    btnSave = new javax.swing.JLabel();
    jLabel2 = new javax.swing.JLabel();
    txtName1 = new javax.swing.JTextField();
    txtName2 = new javax.swing.JTextField();
    jLabel3 = new javax.swing.JLabel();
    bannerPanel = new javax.swing.JPanel();
    btnEdit = new javax.swing.JLabel();
    txtNameSearch = new javax.swing.JTextField();
    jLabel6 = new javax.swing.JLabel();
    jScrollPane1 = new javax.swing.JScrollPane();
    tblBatch = new javax.swing.JTable();
    btnSearch = new javax.swing.JLabel();
    btnAdd = new javax.swing.JLabel();
    btnExport = new javax.swing.JLabel();
    btnDeactive = new javax.swing.JLabel();
    jLabel5 = new javax.swing.JLabel();

    setBackground(new java.awt.Color(224, 224, 255));
    setBorder(null);
    setFrameIcon(null);
    setPreferredSize(new java.awt.Dimension(1024, 700));
    getContentPane().setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

    editAddPanel.setBackground(new java.awt.Color(45, 129, 171));
    editAddPanel.setForeground(new java.awt.Color(255, 255, 255));
    editAddPanel.setMinimumSize(new java.awt.Dimension(1020, 700));
    editAddPanel.setPreferredSize(new java.awt.Dimension(1020, 700));
    editAddPanel.setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

    labelAction.setBackground(new java.awt.Color(1, 100, 151));
    labelAction.setFont(new java.awt.Font("Times New Roman", 1, 24)); // NOI18N
```

```

labelAction.setForeground(new java.awt.Color(255, 255, 255));
labelAction.setHorizontalTextPosition(javax.swing.SwingConstants.CENTER);
labelAction.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/Images/icons8_Batch_Assign_38px.png"))); // NOI18N
labelAction.setText("Add New Batch");
labelAction.setIconTextGap(14);
labelAction.setOpaque(true);
editAddPanel.add(labelAction, new org.netbeans.lib.awtextra.AbsoluteConstraints(-3, 0, 1030, 60));

jLabel1.setBackground(new java.awt.Color(0, 102, 153));
jLabel1.setFont(new java.awt.Font("Tahoma", 1, 12)); // NOI18N
jLabel1.setForeground(new java.awt.Color(255, 255, 255));
jLabel1.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
jLabel1.setText("Name");
jLabel1.setOpaque(true);
editAddPanel.add(jLabel1, new org.netbeans.lib.awtextra.AbsoluteConstraints(190, 120, 88, 26));

btnReset.setBackground(new java.awt.Color(0, 102, 255));
btnReset.setFont(new java.awt.Font("Tahoma", 1, 12)); // NOI18N
btnReset.setForeground(new java.awt.Color(255, 255, 255));
btnReset.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
btnReset.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/Images/icons8_Refresh_17px.png"))); // NOI18N
btnReset.setText("Reset");
btnReset.setOpaque(true);
btnReset.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        btnResetMouseClicked(evt);
    }
});
editAddPanel.add(btnReset, new org.netbeans.lib.awtextra.AbsoluteConstraints(460, 410, 80, 26));

txtName.setBackground(new java.awt.Color(85, 153, 187));
txtName.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N
txtName.setBorder(null);
editAddPanel.add(txtName, new org.netbeans.lib.awtextra.AbsoluteConstraints(280, 120, 212, 26));

dPic.setBackground(new java.awt.Color(255, 255, 255));
editAddPanel.add(dPic, new org.netbeans.lib.awtextra.AbsoluteConstraints(610, 120, 210, 26));

```

```
jLabel7.setBackground(new java.awt.Color(0, 102, 153));
jLabel7.setFont(new java.awt.Font("Tahoma", 1, 12)); // NOI18N
jLabel7.setForeground(new java.awt.Color(255, 255, 255));
jLabel7.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
jLabel7.setText("Date");
jLabel7.setOpaque(true);
editAddPanel.add(jLabel7, new org.netbeans.lib.awtextra.AbsoluteConstraints(520, 120, 90,
26));

btnCancel.setBackground(new java.awt.Color(255, 0, 0));
btnCancel.setFont(new java.awt.Font("Tahoma", 1, 12)); // NOI18N
btnCancel.setForeground(new java.awt.Color(255, 255, 255));
btnCancel.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
btnCancel.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/Images/icons8_Close_Window_17px.png"))); // NOI18N
btnCancel.setText("Cancel");
btnCancel.setOpaque(true);
btnCancel.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        btnCancelMouseClicked(evt);
    }
});
editAddPanel.add(btnCancel, new org.netbeans.lib.awtextra.AbsoluteConstraints(690, 410,
80, 26));

btnSave.setBackground(new java.awt.Color(52, 203, 0));
btnSave.setFont(new java.awt.Font("Tahoma", 1, 12)); // NOI18N
btnSave.setForeground(new java.awt.Color(255, 255, 255));
btnSave.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
btnSave.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/Images/icons8_Save_17px.png"))); // NOI18N
btnSave.setText("Save");
btnSave.setOpaque(true);
btnSave.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        btnSaveMouseClicked(evt);
    }
});
editAddPanel.add(btnSave, new org.netbeans.lib.awtextra.AbsoluteConstraints(200, 410, 80,
26));

jLabel2.setBackground(new java.awt.Color(0, 102, 153));
jLabel2.setFont(new java.awt.Font("Tahoma", 1, 12)); // NOI18N
jLabel2.setForeground(new java.awt.Color(255, 255, 255));
```

```
jLabel2.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
jLabel2.setText("Class");
jLabel2.setOpaque(true);
editAddPanel.add(jLabel2, new org.netbeans.lib.awtextra.AbsoluteConstraints(190, 170, 90,
26));

txtName1.setBackground(new java.awt.Color(85, 153, 187));
txtName1.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N
txtName1.setBorder(null);
editAddPanel.add(txtName1, new org.netbeans.lib.awtextra.AbsoluteConstraints(280, 170,
210, 26));

txtName2.setBackground(new java.awt.Color(85, 153, 187));
txtName2.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N
txtName2.setBorder(null);
editAddPanel.add(txtName2, new org.netbeans.lib.awtextra.AbsoluteConstraints(610, 170,
210, 26));

jLabel3.setBackground(new java.awt.Color(0, 102, 153));
jLabel3.setFont(new java.awt.Font("Tahoma", 1, 12)); // NOI18N
jLabel3.setForeground(new java.awt.Color(255, 255, 255));
jLabel3.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
jLabel3.setText("Course");
jLabel3.setOpaque(true);
editAddPanel.add(jLabel3, new org.netbeans.lib.awtextra.AbsoluteConstraints(520, 170, 90,
26));

getContentPane().add(editAddPanel, new org.netbeans.lib.awtextra.AbsoluteConstraints(0,
0, 1020, 700));

bannerPanel.setBackground(new java.awt.Color(45, 129, 171));
bannerPanel.setMinimumSize(new java.awt.Dimension(1020, 700));
bannerPanel.setPreferredSize(new java.awt.Dimension(1020, 700));
bannerPanel.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        bannerPanelMouseClicked(evt);
    }
});
bannerPanel.setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

btnEdit.setBackground(new java.awt.Color(204, 0, 204));
btnEdit.setFont(new java.awt.Font("SansSerif", 1, 12)); // NOI18N
btnEdit.setForeground(new java.awt.Color(255, 255, 255));
btnEdit.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
```

```

        btnEdit.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/Images/icons8_Edit_Row_17px.png"))); // NOI18N
        btnEdit.setText("Edit");
        btnEdit.setOpaque(true);
        btnEdit.addMouseListener(new java.awt.event.MouseAdapter() {
            public void mouseClicked(java.awt.event.MouseEvent evt) {
                btnEditMouseClicked(evt);
            }
        });
        bannerPanel.add(btnEdit, new org.netbeans.lib.awtextra.AbsoluteConstraints(340, 510, 90,
26));

txtNameSearch.setBackground(new java.awt.Color(85, 153, 187));
txtNameSearch.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N
txtNameSearch.setForeground(new java.awt.Color(255, 255, 255));
txtNameSearch.setBorder(null);
txtNameSearch.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        searchAction(evt);
    }
});
bannerPanel.add(txtNameSearch, new org.netbeans.lib.awtextra.AbsoluteConstraints(340,
100, 320, 26));

jLabel6.setBackground(new java.awt.Color(0, 102, 153));
jLabel6.setFont(new java.awt.Font("Times New Roman", 1, 24)); // NOI18N
jLabel6.setForeground(new java.awt.Color(255, 255, 255));
jLabel6.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
jLabel6.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/Images/icons8_Batch_Assign_38px.png"))); // NOI18N
jLabel6.setText("Batch Manager");
jLabel6.setIconTextGap(20);
jLabel6.setOpaque(true);
bannerPanel.add(jLabel6, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 0, 1026,
60));

jScrollPane1.setBackground(new java.awt.Color(224, 224, 255));
jScrollPane1.setBorder(javax.swing.BorderFactory.createLineBorder(new java.awt.Color(0,
102, 153)));
jScrollPane1.setOpaque(false);

tblBatch.setBackground(new java.awt.Color(4, 74, 121));
tblBatch.setForeground(new java.awt.Color(255, 255, 255));
tblBatch.setModel(new javax.swing.table.DefaultTableModel(

```

```
new Object [][] {
    {null, null, null},
    {null, null, null}
},
new String [] {
    "ID", "Name", "Date"
}
) {
    boolean[] canEdit = new boolean [] {
        false, false, false
    };
}

public boolean isCellEditable(int rowIndex, int columnIndex) {
    return canEdit [columnIndex];
}
});

tblBatch.setSelectionBackground(new java.awt.Color(0, 153, 51));
tblBatch.setShowHorizontalLines(false);
tblBatch.setShowVerticalLines(false);
tblBatch.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        tblBatchMouseClicked(evt);
    }
});
jScrollPane1.setViewportView(tblBatch);

bannerPanel.add(jScrollPane1, new org.netbeans.lib.awtextra.AbsoluteConstraints(140, 150,
700, 340));
```

```
btnSearch.setBackground(new java.awt.Color(255, 102, 0));
btnSearch.setFont(new java.awt.Font("SansSerif", 1, 14)); // NOI18N
btnSearch.setForeground(new java.awt.Color(255, 255, 255));
btnSearch.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
btnSearch.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/Images/icons8_Google_Web_Search_17px.png
")));
// NOI18N
btnSearch.setText("Search");
btnSearch.setOpaque(true);
btnSearch.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        btnSearchMouseClicked(evt);
    }
});
bannerPanel.add(btnSearch, new org.netbeans.lib.awtextra.AbsoluteConstraints(660, 100,
90, 26));

btnAdd.setBackground(new java.awt.Color(51, 204, 0));
btnAdd.setFont(new java.awt.Font("SansSerif", 1, 12)); // NOI18N
btnAdd.setForeground(new java.awt.Color(255, 255, 255));
btnAdd.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
btnAdd.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/Images/icons8_Add_New_17px.png")));
// NOI18N
btnAdd.setText("Add");
btnAdd.setOpaque(true);
btnAdd.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        btnAddMouseClicked(evt);
    }
});
bannerPanel.add(btnAdd, new org.netbeans.lib.awtextra.AbsoluteConstraints(140, 510, 90,
26));

btnExport.setBackground(new java.awt.Color(0, 0, 204));
btnExport.setFont(new java.awt.Font("SansSerif", 1, 12)); // NOI18N
btnExport.setForeground(new java.awt.Color(255, 255, 255));
btnExport.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
btnExport.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/Images/icons8_Outbox_17px.png")));
// NOI18N
btnExport.setText("Export");
btnExport.setOpaque(true);
btnExport.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
```

```
        btnExportMouseClicked(evt);
    }
});
bannerPanel.add(btnExport, new org.netbeans.lib.awtextra.AbsoluteConstraints(750, 510,
90, 26));

btnDeactive.setBackground(new java.awt.Color(255, 0, 0));
btnDeactive.setFont(new java.awt.Font("SansSerif", 1, 12)); // NOI18N
btnDeactive.setForeground(new java.awt.Color(255, 255, 255));
btnDeactive.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
btnDeactive.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/Images/icons8_Radio_Active_17px.png"))); // NOI18N
btnDeactive.setText("Deactivate");
btnDeactive.setOpaque(true);
btnDeactive.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        btnDeactiveMouseClicked(evt);
    }
});
bannerPanel.add(btnDeactive, new org.netbeans.lib.awtextra.AbsoluteConstraints(530, 510,
100, 26));

jLabel5.setBackground(new java.awt.Color(0, 102, 153));
jLabel5.setFont(new java.awt.Font("SansSerif", 1, 14)); // NOI18N
jLabel5.setForeground(new java.awt.Color(255, 255, 255));
jLabel5.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
jLabel5.setText("Name");
jLabel5.setOpaque(true);
bannerPanel.add(jLabel5, new org.netbeans.lib.awtextra.AbsoluteConstraints(250, 100, 90,
26));

getContentPane().add(bannerPanel, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 0,
1020, 700));

pack();
}// </editor-fold>

private void searchAction(java.awt.event.ActionEvent evt) {

}

private void tblBatchMouseClicked(java.awt.event.MouseEvent evt) {
    enableButton(true);
}
```

```
private void btnExportMouseClicked(java.awt.event.MouseEvent evt) {  
    export(); // TODO add your handling code here:  
}  
  
private void btnDeactiveMouseClicked(java.awt.event.MouseEvent evt) {  
    if(studentManager.confirmRemove(this, "Are you sure ?")==0){  
        deactivate();  
    } // TODO add your handling code here:  
}  
  
private void bannerPanelMouseClicked(java.awt.event.MouseEvent evt) {  
    // TODO add your handling code here:  
}  
  
private void btnEditMouseClicked(java.awt.event.MouseEvent evt) {  
    if(studentManager.getSelectRow(tblBatch) !=null){  
        act=EDIT;  
        enableEditAddPanel(true);  
    } // TODO add your handling code here:  
}  
  
private void btnAddMouseClicked(java.awt.event.MouseEvent evt) {  
    act=ADD;  
    enableEditAddPanel(true);  
    btnReset.setVisible(false); // TODO add your handling code here:  
}  
  
private void btnSaveMouseClicked(java.awt.event.MouseEvent evt) {  
    save(); // TODO add your handling code here:  
}  
  
private void btnResetMouseClicked(java.awt.event.MouseEvent evt) {  
    loadEdit(); // TODO add your handling code here:  
}  
  
private void btnCancelMouseClicked(java.awt.event.MouseEvent evt) {  
    enableEditAddPanel(false); // TODO add your handling code here:  
}  
  
private void btnSearchMouseClicked(java.awt.event.MouseEvent evt) {  
    search(); // TODO add your handling code here:  
}  
  
// Variables declaration - do not modify  
private javax.swing.JPanel bannerPanel;  
private javax.swing.JLabel btnAdd;
```

```
private javax.swing.JLabel btnCancel;
private javax.swing.JLabel btnDeactive;
private javax.swing.JLabel btnEdit;
private javax.swing.JLabel btnExport;
private javax.swing.JLabel btnReset;
private javax.swing.JLabel btnSave;
private javax.swing.JLabel btnSearch;
private com.toedter.calendar.JDateChooser dPic;
private javax.swing.JPanel editAddPanel;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel5;
private javax.swing.JLabel jLabel6;
private javax.swing.JLabel jLabel7;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JLabel labelAction;
private javax.swing.JTable tblBatch;
private javax.swing.JTextField txtName;
private javax.swing.JTextField txtName1;
private javax.swing.JTextField txtName2;
private javax.swing.JTextField txtNameSearch;
// End of variables declaration

@Override
public void enableButton(boolean b) {
    btnEdit.setEnabled(b);
    btnDeactive.setEnabled(b);
}

@Override
public boolean validateForm() {
    String name=txtName.getText();
    if(Validate.isEmpty(name)){
        JOptionPane.showMessageDialog(this, "Name is required !");
        return false;
    }
    return true;
}

private void export() {
    ExportFile export=new ExportFile(tblBatch, ExportFile.BATCH);
    export.toExcel();
}
```

- **For Extracurricular Management Module**

```
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */
package StudentMan;

import FileExport.ExportFile;
import Controls.Manager;
import Controls.Validate;
import MainHome.LoginForm;
import MainObjects.ExtraCurricularEntity;
import java.util.Date;
import javax.swing.JOptionPane;
import javax.swing.plaf.basic.BasicInternalFrameUI;
import TableModel.ExtraCurricular;
import MainHome.UIInterface;

/**
 *
 * @author Aeon(Ankit RudrAshokJha)
 */
public class ListExtraCurricular extends javax.swing.JInternalFrame implements UIInterface {
    private ExtraCurricularEntity entity;
    private ExtraCurricular extraCurricular;
    private int actionType;
    /**
     * Creates new form ListPosition
     */
    public ListExtraCurricular(int w, int h) {
        initComponents();
        try {
            for (javax.swing.UIManager.LookAndFeelInfo info :
                javax.swing.UIManager.getInstalledLookAndFeels()) {
                if ("Windows".equals(info.getName())) {
                    javax.swing.UIManager.setLookAndFeel(info.getClassName());
                    break;
                }
            }
        } catch (ClassNotFoundException ex) {
            java.util.logging.Logger.getLogger(LoginForm.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
        } catch (InstantiationException ex) {
```

```

java.util.logging.Logger.getLogger(LoginForm.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
} catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(LoginForm.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
} catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(LoginForm.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
}
this.entity=new ExtraCurricularEntity();
initUI();
setSize(w, h);

setSize(w, h);
}

@Override
public void initUI() {
    ((BasicInternalFrameUI) this.getUI()).setNorthPane(null);
    enableButton(false);
    editAddPanel.setVisible(false);
    entity.loadList(tblExtraCurricular);
}

@Override
public void search() {
    String name=txtNameSearch.getText();
    if( !Validate.isEmpty(txtYearStartSearch.getText()) &&
!Validate.isNumber(txtYearStartSearch.getText())){
        JOptionPane.showMessageDialog(this, "Year is incorrect !");
        return;
    }
    if( !Validate.isEmpty(txtYearEndSearch.getText()) &&
!Validate.isNumber(txtYearEndSearch.getText())){
        JOptionPane.showMessageDialog(this, "Year is incorrect !");
        return;
    }
    int monthStart = 0;
    int monthEnd = 0;
    int yearStart = 0;
    int yearEnd = 0;
    try {
        if (comboMonthStartSearch.getSelectedIndex() != 0) {

```

```

        monthStart=Integer.parseInt(comboMonthStartSearch.getSelectedItem().toString());
    }
    if (comboMonthEndSearch.getSelectedIndex() != 0) {
        monthEnd=Integer.parseInt(comboMonthEndSearch.getSelectedItem().toString());
    }
    if(!txtYearStartSearch.getText().isEmpty()){
        yearStart=Integer.parseInt(txtYearStartSearch.getText());
    }
    if(!txtYearEndSearch.getText().isEmpty()){
        yearEnd=Integer.parseInt(txtYearStartSearch.getText());
    }
    entity.search(name, monthStart, yearStart, monthEnd, yearEnd, tblExtraCurricular);
} catch (NullPointerException | NumberFormatException ex) {}
enableButton(false);
}

@Override
public void enableEditAddPanel(boolean b) {
    if (b == false) {
        tblExtraCurricular.setEnabled(true);
        txtName.setText(null);

        editAddPanel.setVisible(false);
        bannerPanel.setVisible(true);

    } else {
        if (actionType == EDIT) {
            labelAction.setText("Edit Extra Curricular");
            loadEdit();
        } else if (actionType == ADD) {
            labelAction.setText("Add Extra Curricular");
        }
        tblExtraCurricular.setEnabled(false);
        btnReset.setVisible(true);

        editAddPanel.setVisible(true);
        bannerPanel.setVisible(false);
    }
}

@Override
public void save() {
    if (!validateForm()) {
        return;
    }
}

```

```

if(extraCurricular==null){
    extraCurricular=new ExtraCurricular();
}
String name = txtName.getText();
String location=txtLocation.getText();
Date start=null;
start=dPicStart.getDate();
Date end=null;
end=dPicEnd.getDate();

extraCurricular.setName(name);
extraCurricular.setLocation(location);
extraCurricular.setStart(start);
extraCurricular.setEnd(end);
entity.setExtraCurricular(extraCurricular);
int res=0;
res=entity.save();
if (res == Manager.SUCCESS) {
    initUI();
    enableEditAddPanel(false);
}
entity.showMessage(res, this);

}

@Override
public void loadEdit() {
    if(actionType==ADD){
        txtName.setText("");
        txtLocation.setText("");
        dPicStart.setDate(null);
        dPicEnd.setDate(null);
    }else{
        extraCurricular=entity.findById(entity.getSelectID(tblExtraCurricular));
        if(extraCurricular!=null){
            txtName.setText(extraCurricular.getName());
            txtLocation.setText(extraCurricular.getLocation());
            dPicStart.setDate(extraCurricular.getStart());
            dPicEnd.setDate(extraCurricular.getEnd());
        }
    }
}

@Override
public void deactivate() {
    int res = 0;
}

```

```
res=entity.deactive(entity.getSelectID(tblExtraCurricular));
entity.showMessage(res, this);
if (res == Manager.SUCCESS) {
    initUI();
}
/***
 * 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">
private void initComponents() {

    bannerPanel = new javax.swing.JPanel();
    jLabel4 = new javax.swing.JLabel();
    jLabel3 = new javax.swing.JLabel();
    txtNameSearch = new javax.swing.JTextField();
    comboMonthStartSearch = new javax.swing.JComboBox();
    txtYearStartSearch = new javax.swing.JTextField();
    comboMonthEndSearch = new javax.swing.JComboBox();
    txtYearEndSearch = new javax.swing.JTextField();
    jLabel7 = new javax.swing.JLabel();
    jLabel8 = new javax.swing.JLabel();
    jScrollPane1 = new javax.swing.JScrollPane();
    tblExtraCurricular = new javax.swing.JTable();
    btnResetSearch = new javax.swing.JLabel();
    btnSearch = new javax.swing.JLabel();
    txtNameSearch1 = new javax.swing.JTextField();
    jLabel11 = new javax.swing.JLabel();
    btnAdd = new javax.swing.JLabel();
    btnEdit = new javax.swing.JLabel();
    btnDeactive = new javax.swing.JLabel();
    btnExport = new javax.swing.JLabel();
    editAddPanel = new javax.swing.JPanel();
    labelAction = new javax.swing.JLabel();
    jLabel11 = new javax.swing.JLabel();
    txtName = new javax.swing.JTextField();
    jLabel2 = new javax.swing.JLabel();
    txtLocation = new javax.swing.JTextField();
    jLabel5 = new javax.swing.JLabel();
    jLabel6 = new javax.swing.JLabel();
    dPicStart = new com.toedter.calendar.JDateChooser();
    dPicEnd = new com.toedter.calendar.JDateChooser();
```

```
btnReset = new javax.swing.JButton();
btnSave = new javax.swing.JButton();
btnCancel = new javax.swing.JButton();
txtName1 = new javax.swing.JTextField();
jLabel13 = new javax.swing.JLabel();
jLabel14 = new javax.swing.JLabel();
txtName2 = new javax.swing.JTextField();

setBackground(java.awt.Color.white);
setBorder(null);
setPreferredSize(new java.awt.Dimension(1024, 500));
getContentPane().setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

bannerPanel.setBackground(new java.awt.Color(45, 129, 171));
bannerPanel.setMinimumSize(new java.awt.Dimension(1020, 700));
bannerPanel.setPreferredSize(new java.awt.Dimension(1020, 700));
bannerPanel.setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

jLabel4.setBackground(new java.awt.Color(0, 102, 153));
jLabel4.setFont(new java.awt.Font("Times New Roman", 1, 24)); // NOI18N
jLabel4.setForeground(new java.awt.Color(255, 255, 255));
jLabel4.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
jLabel4.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/Images/icons8_Curriculum_40px.png"))); // NOI18N
jLabel4.setText(" Extra Curricular");
jLabel4.setOpaque(true);
bannerPanel.add(jLabel4, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 0, 1030,
60));

jLabel3.setBackground(new java.awt.Color(0, 102, 153));
jLabel3.setFont(new java.awt.Font("Tahoma", 1, 12)); // NOI18N
jLabel3.setForeground(new java.awt.Color(255, 255, 255));
jLabel3.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
jLabel3.setText("Name");
jLabel3.setOpaque(true);
bannerPanel.add(jLabel3, new org.netbeans.lib.awtextra.AbsoluteConstraints(180, 120, 90,
25));

txtNameSearch.setBackground(new java.awt.Color(85, 153, 187));
txtNameSearch.setFont(new java.awt.Font("Tahoma", 1, 11)); // NOI18N
txtNameSearch.setBorder(null);
txtNameSearch.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        txtNameSearchsearchAction(evt);
    }
})
```

```
});  
bannerPanel.add(txtNameSearch, new org.netbeans.lib.awtextra.AbsoluteConstraints(270,  
120, 230, 25));  
  
comboMonthStartSearch.setBackground(new java.awt.Color(85, 153, 187));  
comboMonthStartSearch.setEditable(true);  
comboMonthStartSearch.setFont(new java.awt.Font("Tahoma", 1, 12)); // NOI18N  
comboMonthStartSearch.setForeground(new java.awt.Color(51, 51, 51));  
comboMonthStartSearch.setModel(new javax.swing.DefaultComboBoxModel(new String[]  
{ "Month", "1", "2", "3", "4", "5", "6", "7", "8", "9", "10", "11", "12" }));  
comboMonthStartSearch.setAutoscrolls(true);  
comboMonthStartSearch.setBorder(null);  
bannerPanel.add(comboMonthStartSearch, new  
org.netbeans.lib.awtextra.AbsoluteConstraints(610, 120, 90, 25));  
  
txtYearStartSearch.setBackground(new java.awt.Color(85, 153, 187));  
txtYearStartSearch.setFont(new java.awt.Font("Tahoma", 1, 11)); // NOI18N  
txtYearStartSearch.setBorder(null);  
bannerPanel.add(txtYearStartSearch, new  
org.netbeans.lib.awtextra.AbsoluteConstraints(700, 120, 88, 25));  
  
comboMonthEndSearch.setBackground(new java.awt.Color(85, 153, 187));  
comboMonthEndSearch.setEditable(true);  
comboMonthEndSearch.setFont(new java.awt.Font("Tahoma", 1, 12)); // NOI18N  
comboMonthEndSearch.setForeground(new java.awt.Color(51, 51, 51));  
comboMonthEndSearch.setModel(new javax.swing.DefaultComboBoxModel(new String[] {  
"Month", "1", "2", "3", "4", "5", "6", "7", "8", "9", "10", "11", "12" }));  
comboMonthEndSearch.setAutoscrolls(true);  
comboMonthEndSearch.setBorder(null);  
comboMonthEndSearch.setLightWeightPopupEnabled(false);  
bannerPanel.add(comboMonthEndSearch, new  
org.netbeans.lib.awtextra.AbsoluteConstraints(610, 160, 90, 25));  
  
txtYearEndSearch.setBackground(new java.awt.Color(85, 153, 187));  
txtYearEndSearch.setFont(new java.awt.Font("Tahoma", 1, 11)); // NOI18N  
txtYearEndSearch.setBorder(null);  
bannerPanel.add(txtYearEndSearch, new org.netbeans.lib.awtextra.AbsoluteConstraints(700,  
160, 88, 25));  
  
jLabel7.setBackground(new java.awt.Color(0, 102, 153));  
jLabel7.setFont(new java.awt.Font("Tahoma", 1, 12)); // NOI18N  
jLabel7.setForeground(new java.awt.Color(255, 255, 255));  
jLabel7.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);  
jLabel7.setText("Start");  
jLabel7.setOpaque(true);
```



```

        "ID", "Name", "Location", "Start", "End"
    }
) {
    boolean[] canEdit = new boolean [] {
        false, false, true, true, true
    };

    public boolean isCellEditable(int rowIndex, int columnIndex) {
        return canEdit [columnIndex];
    }
});
tblExtraCurricular.setGridColor(new java.awt.Color(51, 153, 0));
tblExtraCurricular.setShowHorizontalLines(false);
tblExtraCurricular.setShowVerticalLines(false);
tblExtraCurricular.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        tblExtraCurricularMouseClicked(evt);
    }
});
jScrollPane1.setViewportView(tblExtraCurricular);

bannerPanel.add(jScrollPane1, new org.netbeans.lib.awtextra.AbsoluteConstraints(60, 220, 890, 290));

btnResetSearch.setBackground(new java.awt.Color(0, 102, 255));
btnResetSearch.setFont(new java.awt.Font("Tahoma", 1, 12)); // NOI18N
btnResetSearch.setForeground(new java.awt.Color(255, 255, 255));
btnResetSearch.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
btnResetSearch.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/Images/icons8_Refresh_17px.png"))); // NOI18N
btnResetSearch.setText("Reset");
btnResetSearch.setOpaque(true);
btnResetSearch.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        btnResetSearchMouseClicked(evt);
    }
});
bannerPanel.add(btnResetSearch, new org.netbeans.lib.awtextra.AbsoluteConstraints(800, 160, 80, 25));

btnSearch.setBackground(new java.awt.Color(255, 102, 0));
btnSearch.setFont(new java.awt.Font("Tahoma", 1, 12)); // NOI18N
btnSearch.setForeground(new java.awt.Color(255, 255, 255));
btnSearch.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);

```

```

        btnSearch.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/Images/icons8_Google_Web_Search_17px.png
")));
// NOI18N
        btnSearch.setText("Search");
        btnSearch.setOpaque(true);
        btnSearch.addMouseListener(new java.awt.event.MouseAdapter() {
            public void mouseClicked(java.awt.event.MouseEvent evt) {
                btnSearchMouseClicked(evt);
            }
        });
        bannerPanel.add(btnSearch, new org.netbeans.lib.awtextra.AbsoluteConstraints(80, 120,
80, 25));

txtNameSearch1.setBackground(new java.awt.Color(85, 153, 187));
txtNameSearch1.setFont(new java.awt.Font("Tahoma", 1, 11)); // NOI18N
txtNameSearch1.setBorder(null);
txtNameSearch1.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        txtNameSearch1searchAction(evt);
    }
});
bannerPanel.add(txtNameSearch1, new org.netbeans.lib.awtextra.AbsoluteConstraints(270,
160, 230, 25));

jLabel11.setBackground(new java.awt.Color(0, 102, 153));
jLabel11.setFont(new java.awt.Font("Tahoma", 1, 12)); // NOI18N
jLabel11.setForeground(new java.awt.Color(255, 255, 255));
jLabel11.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
jLabel11.setText("ID");
jLabel11.setOpaque(true);
bannerPanel.add(jLabel11, new org.netbeans.lib.awtextra.AbsoluteConstraints(180, 160, 90,
25));

btnAdd.setBackground(new java.awt.Color(51, 204, 0));
btnAdd.setFont(new java.awt.Font("Tahoma", 1, 12)); // NOI18N
btnAdd.setForeground(new java.awt.Color(255, 255, 255));
btnAdd.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
btnAdd.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/Images/icons8_Add_New_17px.png"))); // NOI18N
        btnAdd.setText("Add");
        btnAdd.setOpaque(true);
        btnAdd.addMouseListener(new java.awt.event.MouseAdapter() {
            public void mouseClicked(java.awt.event.MouseEvent evt) {
                btnAddMouseClicked(evt);
            }
        });

```

```
});  
bannerPanel.add(btnAdd, new org.netbeans.lib.awtextra.AbsoluteConstraints(70, 520, 90,  
25));  
  
btnEdit.setBackground(new java.awt.Color(204, 0, 204));  
btnEdit.setFont(new java.awt.Font("Tahoma", 1, 12)); // NOI18N  
btnEdit.setForeground(new java.awt.Color(255, 255, 255));  
btnEdit.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);  
btnEdit.setIcon(new  
javax.swing.ImageIcon(getClass().getResource("/Images/icons8_Edit_Row_17px.png"))); //  
NOI18N  
btnEdit.setText("Edit");  
btnEdit.setOpaque(true);  
btnEdit.addMouseListener(new java.awt.event.MouseAdapter() {  
    public void mouseClicked(java.awt.event.MouseEvent evt) {  
        btnEditMouseClicked(evt);  
    }  
});  
bannerPanel.add(btnEdit, new org.netbeans.lib.awtextra.AbsoluteConstraints(310, 520, 90,  
25));  
  
btnDeactive.setBackground(new java.awt.Color(255, 0, 0));  
btnDeactive.setFont(new java.awt.Font("Tahoma", 1, 12)); // NOI18N  
btnDeactive.setForeground(new java.awt.Color(255, 255, 255));  
btnDeactive.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);  
btnDeactive.setIcon(new  
javax.swing.ImageIcon(getClass().getResource("/Images/icons8_Radio_Active_17px.png"))); //  
NOI18N  
btnDeactive.setText("Deactivate");  
btnDeactive.setOpaque(true);  
btnDeactive.addMouseListener(new java.awt.event.MouseAdapter() {  
    public void mouseClicked(java.awt.event.MouseEvent evt) {  
        btnDeactiveMouseClicked(evt);  
    }  
});  
bannerPanel.add(btnDeactive, new org.netbeans.lib.awtextra.AbsoluteConstraints(570, 520,  
90, 25));  
  
btnExport.setBackground(new java.awt.Color(0, 0, 204));  
btnExport.setFont(new java.awt.Font("Tahoma", 1, 12)); // NOI18N  
btnExport.setForeground(new java.awt.Color(255, 255, 255));  
btnExport.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);  
btnExport.setIcon(new  
javax.swing.ImageIcon(getClass().getResource("/Images/icons8_Outbox_17px.png"))); //  
NOI18N  
btnExport.setText("Export");
```

```
btnExport.setOpaque(true);
btnExport.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        btnExportMouseClicked(evt);
    }
});
bannerPanel.add(btnExport, new org.netbeans.lib.awtextra.AbsoluteConstraints(850, 520, 90, 25));

getContentPane().add(bannerPanel, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 0, 1020, 700));

editAddPanel.setBackground(new java.awt.Color(45, 129, 171));
editAddPanel.setMinimumSize(new java.awt.Dimension(1020, 700));
editAddPanel.setPreferredSize(new java.awt.Dimension(1020, 700));
editAddPanel.setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

labelAction.setBackground(new java.awt.Color(0, 102, 153));
labelAction.setFont(new java.awt.Font("Times New Roman", 1, 24)); // NOI18N
labelAction.setForeground(new java.awt.Color(255, 255, 255));
labelAction.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
labelAction.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/Images/icons8_Curriculum_40px.png"))); // NOI18N
labelAction.setText(" Add New Extra Curricular");
labelAction.setOpaque(true);
editAddPanel.add(labelAction, new org.netbeans.lib.awtextra.AbsoluteConstraints(-3, 0, 1030, 60));

jLabel1.setBackground(new java.awt.Color(1, 100, 151));
jLabel1.setFont(new java.awt.Font("Times New Roman", 1, 12)); // NOI18N
jLabel1.setForeground(new java.awt.Color(255, 255, 255));
jLabel1.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
jLabel1.setText("Name");
jLabel1.setOpaque(true);
editAddPanel.add(jLabel1, new org.netbeans.lib.awtextra.AbsoluteConstraints(220, 140, 90, 25));

txtName.setBackground(new java.awt.Color(85, 153, 187));
txtName.setForeground(new java.awt.Color(255, 255, 255));
txtName.setBorder(null);
editAddPanel.add(txtName, new org.netbeans.lib.awtextra.AbsoluteConstraints(310, 140, 180, 25));

jLabel2.setBackground(new java.awt.Color(1, 100, 151));
jLabel2.setFont(new java.awt.Font("Times New Roman", 1, 12)); // NOI18N
```

```
jLabel2.setForeground(new java.awt.Color(255, 255, 255));
jLabel2.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
jLabel2.setText("Location");
jLabel2.setOpaque(true);
editAddPanel.add(jLabel2, new org.netbeans.lib.awtextra.AbsoluteConstraints(220, 180, 90,
25));

txtLocation.setBackground(new java.awt.Color(85, 153, 187));
txtLocation.setForeground(new java.awt.Color(255, 255, 255));
txtLocation.setBorder(null);
editAddPanel.add(txtLocation, new org.netbeans.lib.awtextra.AbsoluteConstraints(310, 180,
180, 25));

jLabel5.setBackground(new java.awt.Color(1, 100, 151));
jLabel5.setFont(new java.awt.Font("Times New Roman", 1, 12)); // NOI18N
jLabel5.setForeground(new java.awt.Color(255, 255, 255));
jLabel5.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
jLabel5.setText("Start");
jLabel5.setOpaque(true);
editAddPanel.add(jLabel5, new org.netbeans.lib.awtextra.AbsoluteConstraints(500, 140, 88,
25));

jLabel6.setBackground(new java.awt.Color(1, 100, 151));
jLabel6.setFont(new java.awt.Font("Times New Roman", 1, 12)); // NOI18N
jLabel6.setForeground(new java.awt.Color(255, 255, 255));
jLabel6.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
jLabel6.setText("End");
jLabel6.setOpaque(true);
editAddPanel.add(jLabel6, new org.netbeans.lib.awtextra.AbsoluteConstraints(500, 180, 88,
25));

dPicStart.setBackground(new java.awt.Color(85, 153, 187));
dPicStart.setForeground(new java.awt.Color(102, 102, 102));
editAddPanel.add(dPicStart, new org.netbeans.lib.awtextra.AbsoluteConstraints(590, 140,
180, 25));

dPicEnd.setBackground(new java.awt.Color(85, 153, 187));
dPicEnd.setForeground(new java.awt.Color(102, 102, 102));
editAddPanel.add(dPicEnd, new org.netbeans.lib.awtextra.AbsoluteConstraints(590, 180,
180, 25));

btnReset.setBackground(new java.awt.Color(0, 102, 255));
btnReset.setFont(new java.awt.Font("Times New Roman", 1, 12)); // NOI18N
btnReset.setForeground(new java.awt.Color(255, 255, 255));
btnReset.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
```

```
btnReset.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/Images/icons8_Refresh_17px.png"))); // NOI18N
btnReset.setText("Reset");
btnReset.setOpaque(true);
btnReset.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        btnResetMouseClicked(evt);
    }
});
editAddPanel.add(btnReset, new org.netbeans.lib.awtextra.AbsoluteConstraints(460, 350, 88, 25));

btnSave.setBackground(new java.awt.Color(52, 203, 0));
btnSave.setFont(new java.awt.Font("Times New Roman", 1, 12)); // NOI18N
btnSave.setForeground(new java.awt.Color(255, 255, 255));
btnSave.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
btnSave.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/Images/icons8_Save_18px.png"))); // NOI18N
btnSave.setText("Save");
btnSave.setOpaque(true);
btnSave.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        btnSaveMouseClicked(evt);
    }
});
editAddPanel.add(btnSave, new org.netbeans.lib.awtextra.AbsoluteConstraints(220, 350, 88, 25));

btnCancel.setBackground(new java.awt.Color(255, 0, 0));
btnCancel.setFont(new java.awt.Font("Times New Roman", 1, 12)); // NOI18N
btnCancel.setForeground(new java.awt.Color(255, 255, 255));
btnCancel.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
btnCancel.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/Images/icons8_Close_Window_17px.png"))); // NOI18N
btnCancel.setText("Cancel");
btnCancel.setOpaque(true);
btnCancel.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        btnCancelMouseClicked(evt);
    }
});
editAddPanel.add(btnCancel, new org.netbeans.lib.awtextra.AbsoluteConstraints(680, 350, 88, 25));
```

```
txtName1.setBackground(new java.awt.Color(85, 153, 187));
txtName1.setForeground(new java.awt.Color(255, 255, 255));
txtName1.setBorder(null);
editAddPanel.add(txtName1, new org.netbeans.lib.awtextra.AbsoluteConstraints(310, 220,
180, 25));

jLabel13.setBackground(new java.awt.Color(1, 100, 151));
jLabel13.setFont(new java.awt.Font("Times New Roman", 1, 12)); // NOI18N
jLabel13.setForeground(new java.awt.Color(255, 255, 255));
jLabel13.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
jLabel13.setText("ID");
jLabel13.setOpaque(true);
editAddPanel.add(jLabel13, new org.netbeans.lib.awtextra.AbsoluteConstraints(220, 220,
90, 25));

jLabel14.setBackground(new java.awt.Color(1, 100, 151));
jLabel14.setFont(new java.awt.Font("Times New Roman", 1, 12)); // NOI18N
jLabel14.setForeground(new java.awt.Color(255, 255, 255));
jLabel14.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
jLabel14.setText("Points");
jLabel14.setOpaque(true);
editAddPanel.add(jLabel14, new org.netbeans.lib.awtextra.AbsoluteConstraints(500, 220,
90, 25));

txtName2.setBackground(new java.awt.Color(85, 153, 187));
txtName2.setForeground(new java.awt.Color(255, 255, 255));
txtName2.setBorder(null);
editAddPanel.add(txtName2, new org.netbeans.lib.awtextra.AbsoluteConstraints(590, 220,
180, 25));

getContentPane().add(editAddPanel, new org.netbeans.lib.awtextra.AbsoluteConstraints(0,
0, 1020, 700));

pack();
}// </editor-fold>

private void txtNameSearchsearchAction(java.awt.event.ActionEvent evt) {
    search();
}

private void tblExtraCurricularMouseClicked(java.awt.event.MouseEvent evt) {
    enableButton(true);
}

private void txtNameSearch1searchAction(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}
```

```
}

private void btnSearchMouseClicked(java.awt.event.MouseEvent evt) {
    search(); // TODO add your handling code here:
}

private void btnResetSearchMouseClicked(java.awt.event.MouseEvent evt) {
    txtNameSearch.setText("");
    comboMonthStartSearch.setSelectedIndex(0);
    comboMonthEndSearch.setSelectedIndex(0);
    txtYearEndSearch.setText("");
    txtYearStartSearch.setText(""); // TODO add your handling code here:
}

private void btnAddMouseClicked(java.awt.event.MouseEvent evt) {
actionType=ADD;
    enableEditAddPanel(true);
    loadEdit();
    btnReset.setVisible(false); // TODO add your handling code here:
}

private void btnEditMouseClicked(java.awt.event.MouseEvent evt) {
if (entity.getSelectID(tblExtraCurricular) != 0) {
    enableEditAddPanel(true);
    actionType=EDIT;
} // TODO add your handling code here:
}

private void btnDeactiveMouseClicked(java.awt.event.MouseEvent evt) {
if (entity.confirmRemove(this, "Are you sure ?") == 0) {
    deactivate();
} // TODO add your handling code here:
}

private void btnExportMouseClicked(java.awt.event.MouseEvent evt) {
export(); // TODO add your handling code here:
}

private void btnSaveMouseClicked(java.awt.event.MouseEvent evt) {
save(); // TODO add your handling code here:
}

private void btnResetMouseClicked(java.awt.event.MouseEvent evt) {
loadEdit(); // TODO add your handling code here:
}
```

```
private void btnCancelMouseClicked(java.awt.event.MouseEvent evt) {  
    enableEditAddPanel(false); // TODO add your handling code here:  
}  
  
// Variables declaration - do not modify  
private javax.swing.JPanel bannerPanel;  
private javax.swing.JLabel btnAdd;  
private javax.swing.JLabel btnCancel;  
private javax.swing.JLabel btnDeactive;  
private javax.swing.JLabel btnEdit;  
private javax.swing.JLabel btnExport;  
private javax.swing.JLabel btnReset;  
private javax.swing.JLabel btnResetSearch;  
private javax.swing.JLabel btnSave;  
private javax.swing.JLabel btnSearch;  
private javax.swing.JComboBox comboMonthEndSearch;  
private javax.swing.JComboBox comboMonthStartSearch;  
private com.toedter.calendar.JDateChooser dPicEnd;  
private com.toedter.calendar.JDateChooser dPicStart;  
private javax.swing.JPanel editAddPanel;  
private javax.swing.JLabel jLabel1;  
private javax.swing.JLabel jLabel11;  
private javax.swing.JLabel jLabel13;  
private javax.swing.JLabel jLabel14;  
private javax.swing.JLabel jLabel2;  
private javax.swing.JLabel jLabel3;  
private javax.swing.JLabel jLabel4;  
private javax.swing.JLabel jLabel5;  
private javax.swing.JLabel jLabel6;  
private javax.swing.JLabel jLabel7;  
private javax.swing.JLabel jLabel8;  
private javax.swing.JScrollPane jScrollPane1;  
private javax.swing.JLabel labelAction;  
private javax.swing.JTable tblExtraCurricular;  
private javax.swing.JTextField txtLocation;  
private javax.swing.JTextField txtName;  
private javax.swing.JTextField txtName1;  
private javax.swing.JTextField txtName2;  
private javax.swing.JTextField txtNameSearch;  
private javax.swing.JTextField txtNameSearch1;  
private javax.swing.JTextField txtYearEndSearch;  
private javax.swing.JTextField txtYearStartSearch;  
// End of variables declaration  
  
@Override  
public void enableButton(boolean b) {
```

```

        btnEdit.setEnabled(b);
        btnDeactive.setEnabled(b);
    }

    @Override
    public boolean validateForm() {
        String name = txtName.getText();
        if (Validate.isEmpty(name)) {
            JOptionPane.showMessageDialog(this, "Name is required !");
            return false;
        }
        return true;
    }

    private void export() {
        ExportFile export =new ExportFile(tblExtraCurricular,
        ExportFile.EXTRA_CURRICULAR);
        export.toExcel();
    }
}

```

- **For Marks Management Module of AeonSMS**

```

/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */
package StudentMan;

import Controls.Manager;
import FileExport.ExportFile;
import MainHome.UIInterface;
import static MainHome.UIInterface.ADD;
import static MainHome.UIInterface.EDIT;
import MainObjects.MarkEntity;
import MainObjects.StudentEntity;
import MainObjects.SubjectEntity;
import TableModel.Mark;
import TableModel.Subject;
import java.util.Date;
import java.util.List;
import javax.swing.JOptionPane;
import javax.swing.plaf.basic.BasicInternalFrameUI;

```

```
/***
 *
 * @author Aeon(Ankit RudrAshokJha)
 */
public class ListMark extends javax.swing.JInternalFrame implements UIInterface {

    private int act;
    private MarkEntity entity;
    private SubjectEntity subjectEntity;
    private StudentEntity studentEntity;
    private Mark mark;

    /**
     * Creates new form SearchMark
     */
    public ListMark(int w, int h) {
        initComponents();
        this.entity = new MarkEntity();
        this.subjectEntity=new SubjectEntity();
        this.studentEntity=new StudentEntity();
        initUI();
        setSize(w, h);
    }

    @Override
    public void initUI() {
        ((BasicInternalFrameUI) this.getUI()).setNorthPane(null);
        enableEditAddPanel(false);
        entity.loadList(tblMark);
        List<String> subjectList=subjectEntity.getNameList();
        comboSubject.removeAllItems();
        comboSubject.addItem("Choose");
        for(String s:subjectList){
            comboSubject.addItem(s);
        }

        enableButton(false);
    }
    private void export(){

        ExportFile export=new ExportFile(tblMark, ExportFile.MARK);
        export.toExcel();

    }

    private void studentChooser() {
```

```
StudentChooser studentChooser = new StudentChooser(null, true);
studentChooser.setVisible(true);
if (studentChooser.getChooser() > 0) {
    if(mark==null){
        mark=new Mark();
    }
    mark.setStudent(new StudentEntity().findByID(studentChooser.getChooser()));
    labelStudentNo.setText(mark.getStudent().getNo() + "");
}
/***
 * 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">
private void initComponents() {

    bannerPanel = new javax.swing.JPanel();
    txtYearSearch = new javax.swing.JTextField();
    comboSearchSubject = new javax.swing.JComboBox();
    comboMonthSearch = new javax.swing.JComboBox();
    txtSearchStudentName = new javax.swing.JTextField();
    jLabel6 = new javax.swing.JLabel();
    txtSearchStudentNo = new javax.swing.JTextField();
    jLabel44 = new javax.swing.JLabel();
    jLabel47 = new javax.swing.JLabel();
    jLabel14 = new javax.swing.JLabel();
    jLabel15 = new javax.swing.JLabel();
    jScrollPane1 = new javax.swing.JScrollPane();
    tblMark = new javax.swing.JTable();
    btnEdit = new javax.swing.JLabel();
    btnAdd = new javax.swing.JLabel();
    btnExport = new javax.swing.JLabel();
    btnRemove = new javax.swing.JLabel();
    jLabel16 = new javax.swing.JLabel();
    btnSearchReset = new javax.swing.JLabel();
    editAddPanel = new javax.swing.JPanel();
    labelAction = new javax.swing.JLabel();
    jLabel1 = new javax.swing.JLabel();
    labelStudentNo = new javax.swing.JLabel();
    jLabel4 = new javax.swing.JLabel();
    txtPoint = new javax.swing.JTextField();
    jLabel5 = new javax.swing.JLabel();
```

```
dPicTime = new com.toedter.calendar.JDateChooser();
jLabel2 = new javax.swing.JLabel();
comboSubject = new javax.swing.JComboBox();
btnReset = new javax.swing.JButton();
btnSave = new javax.swing.JButton();
btnStudentChooser = new javax.swing.JButton();
btnCancel = new javax.swing.JButton();
labelStudentNo1 = new javax.swing.JLabel();
jLabel3 = new javax.swing.JLabel();
labelStudentNo2 = new javax.swing.JLabel();
jLabel7 = new javax.swing.JLabel();

setBorder(null);
setIcon(null);
setPreferredSize(new java.awt.Dimension(1024, 700));
getContentPane().setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

bannerPanel.setBackground(new java.awt.Color(45, 129, 171));
bannerPanel.setMinimumSize(new java.awt.Dimension(1020, 700));
bannerPanel.setPreferredSize(new java.awt.Dimension(1020, 700));
bannerPanel.setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

txtYearSearch.setBackground(new java.awt.Color(85, 153, 187));
txtYearSearch.setFont(new java.awt.Font("Tahoma", 1, 12)); // NOI18N
txtYearSearch.setForeground(new java.awt.Color(255, 255, 255));
txtYearSearch.setBorder(null);
bannerPanel.add(txtYearSearch, new org.netbeans.lib.awtextra.AbsoluteConstraints(630, 140, 80, 24));

comboSearchSubject.setBackground(new java.awt.Color(85, 153, 187));
comboSearchSubject.setEditable(true);
comboSearchSubject.setFont(new java.awt.Font("Tahoma", 1, 12)); // NOI18N
comboSearchSubject.setForeground(new java.awt.Color(255, 255, 255));
comboSearchSubject.setBorder(null);
bannerPanel.add(comboSearchSubject, new org.netbeans.lib.awtextra.AbsoluteConstraints(250, 140, 170, 24));

comboMonthSearch.setBackground(new java.awt.Color(85, 153, 187));
comboMonthSearch.setEditable(true);
comboMonthSearch.setFont(new java.awt.Font("Tahoma", 1, 12)); // NOI18N
comboMonthSearch.setForeground(new java.awt.Color(255, 255, 255));
comboMonthSearch.setModel(new javax.swing.DefaultComboBoxModel(new String[] {"Month", "1", "2", "3", "4", "5", "6", "7", "8", "9", "10", "11", "12"}));
comboMonthSearch.setBorder(null);
bannerPanel.add(comboMonthSearch, new org.netbeans.lib.awtextra.AbsoluteConstraints(550, 140, 80, 24));
```

```
txtSearchStudentName.setBackground(new java.awt.Color(85, 153, 187));
txtSearchStudentName.setFont(new java.awt.Font("Tahoma", 1, 12)); // NOI18N
txtSearchStudentName.setForeground(new java.awt.Color(255, 255, 255));
txtSearchStudentName.setBorder(null);
bannerPanel.add(txtSearchStudentName, new
org.netbeans.lib.awtextra.AbsoluteConstraints(250, 100, 166, 24));

jLabel6.setBackground(new java.awt.Color(1, 100, 151));
jLabel6.setFont(new java.awt.Font("Tahoma", 1, 12)); // NOI18N
jLabel6.setForeground(new java.awt.Color(255, 255, 255));
jLabel6.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
jLabel6.setText("Date");
jLabel6.setOpaque(true);
bannerPanel.add(jLabel6, new org.netbeans.lib.awtextra.AbsoluteConstraints(480, 140, 70,
24));

txtSearchStudentNo.setBackground(new java.awt.Color(85, 153, 187));
txtSearchStudentNo.setFont(new java.awt.Font("Tahoma", 1, 12)); // NOI18N
txtSearchStudentNo.setForeground(new java.awt.Color(255, 255, 255));
txtSearchStudentNo.setBorder(null);
bannerPanel.add(txtSearchStudentNo, new
org.netbeans.lib.awtextra.AbsoluteConstraints(550, 100, 160, 24));

jLabel44.setBackground(new java.awt.Color(1, 100, 151));
jLabel44.setFont(new java.awt.Font("Tahoma", 1, 12)); // NOI18N
jLabel44.setForeground(new java.awt.Color(255, 255, 255));
jLabel44.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
jLabel44.setText("Student name");
jLabel44.setOpaque(true);
bannerPanel.add(jLabel44, new org.netbeans.lib.awtextra.AbsoluteConstraints(140, 100,
112, 24));

jLabel47.setBackground(new java.awt.Color(1, 100, 151));
jLabel47.setFont(new java.awt.Font("Tahoma", 1, 12)); // NOI18N
jLabel47.setForeground(new java.awt.Color(255, 255, 255));
jLabel47.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
jLabel47.setText("Subject");
jLabel47.setOpaque(true);
bannerPanel.add(jLabel47, new org.netbeans.lib.awtextra.AbsoluteConstraints(140, 140,
110, 24));

jLabel14.setBackground(new java.awt.Color(0, 102, 153));
jLabel14.setFont(new java.awt.Font("Times New Roman", 1, 24)); // NOI18N
jLabel14.setForeground(new java.awt.Color(255, 255, 255));
jLabel14.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
```

```

jLabel14.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/Images/icons8_Exam_36px.png"))); // NOI18N
jLabel14.setText("Mark");
jLabel14.setIconTextGap(24);
jLabel14.setOpaque(true);
bannerPanel.add(jLabel14, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 0, 1020,
60));

jLabel15.setBackground(new java.awt.Color(1, 100, 151));
jLabel15.setFont(new java.awt.Font("Tahoma", 1, 12)); // NOI18N
jLabel15.setForeground(new java.awt.Color(255, 255, 255));
jLabel15.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
jLabel15.setText("Roll no");
jLabel15.setOpaque(true);
bannerPanel.add(jLabel15, new org.netbeans.lib.awtextra.AbsoluteConstraints(480, 100, 70,
24));

jScrollPane1.setBorder(javax.swing.BorderFactory.createLineBorder(new java.awt.Color(4,
74, 121)));

tblMark.setBackground(new java.awt.Color(4, 74, 121));
tblMark.setFont(new java.awt.Font("Tahoma", 1, 11)); // NOI18N
tblMark.setForeground(new java.awt.Color(255, 255, 255));
tblMark.setModel(new javax.swing.table.DefaultTableModel(
new Object [][] {
{null, null, null, null, null, null}
},
new String [] {
"ID", "Subject", "Student No", "Student", "Point", "Time"
}
) {
Class[] types = new Class [] {
java.lang.String.class, java.lang.String.class, java.lang.Object.class,
java.lang.String.class, java.lang.String.class, java.lang.String.class
};
boolean[] canEdit = new boolean [] {
false, false, true, false, false, true
};

public Class getColumnClass(int columnIndex) {
    return types [columnIndex];
}

public boolean isCellEditable(int rowIndex, int columnIndex) {
    return canEdit [columnIndex];
}

```

```
});  
tblMark.addMouseListener(new java.awt.event.MouseAdapter() {  
    public void mouseClicked(java.awt.event.MouseEvent evt) {  
        tblMarkMouseClicked(evt);  
    }  
});  
jScrollPane1.setViewportView(tblMark);  
  
bannerPanel.add(jScrollPane1, new org.netbeans.lib.awtextra.AbsoluteConstraints(140, 220, 730, 250));  
  
btnEdit.setBackground(new java.awt.Color(204, 0, 204));  
btnEdit.setFont(new java.awt.Font("Tahoma", 1, 12)); // NOI18N  
btnEdit.setForeground(new java.awt.Color(255, 255, 255));  
btnEdit.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);  
btnEdit.setIcon(new  
javax.swing.ImageIcon(getClass().getResource("/Images/icons8_Edit_Row_17px.png"))); //  
NOI18N  
btnEdit.setText("Edit");  
btnEdit.setOpaque(true);  
btnEdit.addMouseListener(new java.awt.event.MouseAdapter() {  
    public void mouseClicked(java.awt.event.MouseEvent evt) {  
        btnEditMouseClicked(evt);  
    }  
});  
bannerPanel.add(btnEdit, new org.netbeans.lib.awtextra.AbsoluteConstraints(360, 490, 90, 26));  
  
btnAdd.setBackground(new java.awt.Color(51, 204, 0));  
btnAdd.setFont(new java.awt.Font("Tahoma", 1, 12)); // NOI18N  
btnAdd.setForeground(new java.awt.Color(255, 255, 255));  
btnAdd.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);  
btnAdd.setIcon(new  
javax.swing.ImageIcon(getClass().getResource("/Images/icons8_Add_New_17px.png"))); //  
NOI18N  
btnAdd.setText("Add");  
btnAdd.setOpaque(true);  
btnAdd.addMouseListener(new java.awt.event.MouseAdapter() {  
    public void mouseClicked(java.awt.event.MouseEvent evt) {  
        btnAddMouseClicked(evt);  
    }  
});  
bannerPanel.add(btnAdd, new org.netbeans.lib.awtextra.AbsoluteConstraints(140, 490, 90, 26));  
  
btnExport.setBackground(new java.awt.Color(0, 0, 204));
```

```
btnExport.setFont(new java.awt.Font("Tahoma", 1, 12)); // NOI18N
btnExport.setForeground(new java.awt.Color(255, 255, 255));
btnExport.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
btnExport.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/Images/icons8_Outbox_17px.png"))); // NOI18N
btnExport.setText("Export");
btnExport.setOpaque(true);
btnExport.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        btnExportMouseClicked(evt);
    }
});
bannerPanel.add(btnExport, new org.netbeans.lib.awtextra.AbsoluteConstraints(780, 490, 90, 26));

btnRemove.setBackground(new java.awt.Color(255, 0, 0));
btnRemove.setFont(new java.awt.Font("Tahoma", 1, 12)); // NOI18N
btnRemove.setForeground(new java.awt.Color(255, 255, 255));
btnRemove.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
btnRemove.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/Images/icons8_Trash_17px.png"))); // NOI18N
btnRemove.setText("Remove");
btnRemove.setOpaque(true);
btnRemove.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        btnRemoveMouseClicked(evt);
    }
});
bannerPanel.add(btnRemove, new org.netbeans.lib.awtextra.AbsoluteConstraints(572, 490, 90, 26));

jLabel16.setBackground(new java.awt.Color(255, 102, 0));
jLabel16.setFont(new java.awt.Font("Tahoma", 1, 12)); // NOI18N
jLabel16.setForeground(new java.awt.Color(255, 255, 255));
jLabel16.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
jLabel16.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/Images/icons8_Google_Web_Search_17px.png
"))); // NOI18N
jLabel16.setText("Search");
jLabel16.setOpaque(true);
jLabel16.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        jLabel16MouseClicked(evt);
    }
});

```

```
bannerPanel.add(jLabel16, new org.netbeans.lib.awtextra.AbsoluteConstraints(780, 100, 90, 24));  
  
btnSearchReset.setBackground(new java.awt.Color(0, 102, 255));  
btnSearchReset.setFont(new java.awt.Font("Tahoma", 1, 12)); // NOI18N  
btnSearchReset.setForeground(new java.awt.Color(255, 255, 255));  
btnSearchReset.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);  
btnSearchReset.setIcon(new javax.swing.ImageIcon(getClass().getResource("/Images/icons8_Refresh_17px.png"))); // NOI18N  
btnSearchReset.setText("Reset");  
btnSearchReset.setOpaque(true);  
btnSearchReset.addMouseListener(new java.awt.event.MouseAdapter() {  
    public void mouseClicked(java.awt.event.MouseEvent evt) {  
        btnSearchResetMouseClicked(evt);  
    }  
});  
bannerPanel.add(btnSearchReset, new org.netbeans.lib.awtextra.AbsoluteConstraints(780, 140, 90, 24));  
  
getContentPane().add(bannerPanel, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 0, 1020, 700));  
  
editAddPanel.setBackground(new java.awt.Color(45, 129, 171));  
editAddPanel.setMinimumSize(new java.awt.Dimension(1020, 700));  
editAddPanel.setPreferredSize(new java.awt.Dimension(1020, 700));  
editAddPanel.setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());  
  
labelAction.setBackground(new java.awt.Color(0, 102, 153));  
labelAction.setFont(new java.awt.Font("Times New Roman", 1, 24)); // NOI18N  
labelAction.setForeground(new java.awt.Color(255, 255, 255));  
labelAction.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);  
labelAction.setText("Update Students Result");  
labelAction.setOpaque(true);  
editAddPanel.add(labelAction, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 0, 1020, 60));  
  
jLabel1.setBackground(new java.awt.Color(4, 74, 121));  
jLabel1.setFont(new java.awt.Font("Tahoma", 1, 12)); // NOI18N  
jLabel1.setForeground(new java.awt.Color(255, 255, 255));  
jLabel1.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);  
jLabel1.setText("Student No");  
jLabel1.setOpaque(true);  
editAddPanel.add(jLabel1, new org.netbeans.lib.awtextra.AbsoluteConstraints(210, 120, 112, 25));
```

```
labelStudentNo.setBackground(new java.awt.Color(85, 153, 187));
labelStudentNo.setFont(new java.awt.Font("Tahoma", 1, 11)); // NOI18N
labelStudentNo.setForeground(new java.awt.Color(255, 255, 255));
labelStudentNo.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
labelStudentNo.setText(" Unknown");
labelStudentNo.setHorizontalTextPosition(javax.swing.SwingConstants.CENTER);
labelStudentNo.setOpaque(true);
editAddPanel.add(labelStudentNo, new org.netbeans.lib.awtextra.AbsoluteConstraints(320,
120, 170, 25));

jLabel4.setBackground(new java.awt.Color(4, 74, 121));
jLabel4.setFont(new java.awt.Font("Tahoma", 1, 12)); // NOI18N
jLabel4.setForeground(new java.awt.Color(255, 255, 255));
jLabel4.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
jLabel4.setText("Marks");
jLabel4.setOpaque(true);
editAddPanel.add(jLabel4, new org.netbeans.lib.awtextra.AbsoluteConstraints(550, 120,
112, 25));

txtPoint.setBackground(new java.awt.Color(85, 153, 187));
txtPoint.setFont(new java.awt.Font("Tahoma", 1, 11)); // NOI18N
txtPoint.setForeground(new java.awt.Color(255, 255, 255));
txtPoint.setBorder(null);
editAddPanel.add(txtPoint, new org.netbeans.lib.awtextra.AbsoluteConstraints(660, 120,
130, 25));

jLabel5.setBackground(new java.awt.Color(4, 74, 121));
jLabel5.setFont(new java.awt.Font("Tahoma", 1, 12)); // NOI18N
jLabel5.setForeground(new java.awt.Color(255, 255, 255));
jLabel5.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
jLabel5.setText("Date");
jLabel5.setOpaque(true);
editAddPanel.add(jLabel5, new org.netbeans.lib.awtextra.AbsoluteConstraints(550, 200,
112, 25));

dPicTime.setOpaque(false);
editAddPanel.add(dPicTime, new org.netbeans.lib.awtextra.AbsoluteConstraints(660, 200,
130, 25));

jLabel2.setBackground(new java.awt.Color(4, 74, 121));
jLabel2.setFont(new java.awt.Font("Tahoma", 1, 12)); // NOI18N
jLabel2.setForeground(new java.awt.Color(255, 255, 255));
jLabel2.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
jLabel2.setText("Subject");
jLabel2.setOpaque(true);
```

```
editAddPanel.add(jLabel2, new org.netbeans.lib.awtextra.AbsoluteConstraints(210, 200, 112, 25));

        comboSubject.setBackground(new java.awt.Color(85, 153, 187));
        comboSubject.setEditable(true);
        comboSubject.setBorder(null);
        editAddPanel.add(comboSubject, new org.netbeans.lib.awtextra.AbsoluteConstraints(320, 200, 210, 25));

        btnReset.setBackground(new java.awt.Color(0, 102, 255));
        btnReset.setFont(new java.awt.Font("Tahoma", 1, 12)); // NOI18N
        btnReset.setForeground(new java.awt.Color(255, 255, 255));
        btnReset.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
        btnReset.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/Images/icons8_Refresh_17px.png"))); // NOI18N
        btnReset.setText("Reset");
        btnReset.setOpaque(true);
        editAddPanel.add(btnReset, new org.netbeans.lib.awtextra.AbsoluteConstraints(470, 330, 80, 25));

        btnSave.setBackground(new java.awt.Color(51, 204, 0));
        btnSave.setFont(new java.awt.Font("Tahoma", 1, 12)); // NOI18N
        btnSave.setForeground(new java.awt.Color(255, 255, 255));
        btnSave.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
        btnSave.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/Images/icons8_Save_17px.png"))); // NOI18N
        btnSave.setText("Save");
        btnSave.setOpaque(true);
        btnSave.addMouseListener(new java.awt.event.MouseAdapter() {
            public void mouseClicked(java.awt.event.MouseEvent evt) {
                btnSaveMouseClicked(evt);
            }
        });
        editAddPanel.add(btnSave, new org.netbeans.lib.awtextra.AbsoluteConstraints(210, 330, 80, 25));

        btnStudentChooser.setBackground(new java.awt.Color(4, 74, 121));
        btnStudentChooser.setFont(new java.awt.Font("Tahoma", 1, 12)); // NOI18N
        btnStudentChooser.setForeground(new java.awt.Color(255, 255, 255));
        btnStudentChooser.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
        btnStudentChooser.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/Images/icons8_Browse_Folder_18px.png"))); // NOI18N
        btnStudentChooser.setOpaque(true);
        btnStudentChooser.addMouseListener(new java.awt.event.MouseAdapter() {
```

```

        public void mouseClicked(java.awt.event.MouseEvent evt) {
            btnStudentChooserMouseClicked(evt);
        }
    });
    editAddPanel.add(btnStudentChooser, new
org.netbeans.lib.awtextra.AbsoluteConstraints(490, 120, 40, 25));

    btnCancel.setBackground(new java.awt.Color(255, 0, 0));
    btnCancel.setFont(new java.awt.Font("Tahoma", 1, 12)); // NOI18N
    btnCancel.setForeground(new java.awt.Color(255, 255, 255));
    btnCancel.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
    btnCancel.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/Images/icons8_Close_Window_17px.png"))); // NOI18N
    btnCancel.setText("Cancel");
    btnCancel.setOpaque(true);
    btnCancel.addMouseListener(new java.awt.event.MouseAdapter() {
        public void mouseClicked(java.awt.event.MouseEvent evt) {
            btnCancelMouseClicked(evt);
        }
    });
    editAddPanel.add(btnCancel, new org.netbeans.lib.awtextra.AbsoluteConstraints(710, 330,
80, 25));

    labelStudentNo1.setBackground(new java.awt.Color(85, 153, 187));
    labelStudentNo1.setFont(new java.awt.Font("Tahoma", 1, 11)); // NOI18N
    labelStudentNo1.setForeground(new java.awt.Color(255, 255, 255));
    labelStudentNo1.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
    labelStudentNo1.setText(" Unknown");
    labelStudentNo1.setHorizontalTextPosition(javax.swing.SwingConstants.CENTER);
    labelStudentNo1.setOpaque(true);
    editAddPanel.add(labelStudentNo1, new org.netbeans.lib.awtextra.AbsoluteConstraints(320,
160, 210, 25));

    jLabel3.setBackground(new java.awt.Color(4, 74, 121));
    jLabel3.setFont(new java.awt.Font("Tahoma", 1, 12)); // NOI18N
    jLabel3.setForeground(new java.awt.Color(255, 255, 255));
    jLabel3.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
    jLabel3.setText("Class");
    jLabel3.setOpaque(true);
    editAddPanel.add(jLabel3, new org.netbeans.lib.awtextra.AbsoluteConstraints(210, 160,
112, 25));

    labelStudentNo2.setBackground(new java.awt.Color(85, 153, 187));
    labelStudentNo2.setFont(new java.awt.Font("Tahoma", 1, 11)); // NOI18N
    labelStudentNo2.setForeground(new java.awt.Color(255, 255, 255));

```

```
labelStudentNo2.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
labelStudentNo2.setText(" Unknown");
labelStudentNo2.setHorizontalTextPosition(javax.swing.SwingConstants.CENTER);
labelStudentNo2.setOpaque(true);
editAddPanel.add(labelStudentNo2, new org.netbeans.lib.awtextra.AbsoluteConstraints(660,
160, 130, 25));

jLabel7.setBackground(new java.awt.Color(4, 74, 121));
jLabel7.setFont(new java.awt.Font("Tahoma", 1, 12)); // NOI18N
jLabel7.setForeground(new java.awt.Color(255, 255, 255));
jLabel7.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
jLabel7.setText("Roll No");
jLabel7.setOpaque(true);
editAddPanel.add(jLabel7, new org.netbeans.lib.awtextra.AbsoluteConstraints(550, 160,
110, 25));

getContentPane().add(editAddPanel, new org.netbeans.lib.awtextra.AbsoluteConstraints(0,
0, 1020, 700));

pack();
}// </editor-fold>

private void tblMarkMouseClicked(java.awt.event.MouseEvent evt) {
    enableButton(true);
}

private void jLabel16MouseClicked(java.awt.event.MouseEvent evt) {
search(); // TODO add your handling code here:
}

private void btnExportMouseClicked(java.awt.event.MouseEvent evt) {
export(); // TODO add your handling code here:
}

private void btnRemoveMouseClicked(java.awt.event.MouseEvent evt) {
int confirm = entity.confirmRemove(this, "Are you Sure");
if (confirm == 0) {
    int res=entity.remove(entity.getSelectID(tblMark));
    if(res==Manager.SUCCESS){
        initUI();
    }
    entity.showMessage(res, this);
}
// TODO add your handling code here:
}
```

```
private void btnEditMouseClicked(java.awt.event.MouseEvent evt) {  
    act = EDIT;  
    enableEditAddPanel(true);  
    btnReset.setVisible(false); // TODO add your handling code here:  
}  
  
private void btnAddMouseClicked(java.awt.event.MouseEvent evt) {  
    act = ADD;  
    enableEditAddPanel(true);  
    btnReset.setVisible(false); // TODO add your handling code here:  
}  
  
private void btnSearchResetMouseClicked(java.awt.event.MouseEvent evt) {  
    initUI(); // TODO add your handling code here:  
}  
  
private void btnStudentChooserMouseClicked(java.awt.event.MouseEvent evt) {  
    studentChooser(); // TODO add your handling code here:  
}  
  
private void btnSaveMouseClicked(java.awt.event.MouseEvent evt) {  
    save(); // TODO add your handling code here:  
}  
  
private void btnCancelMouseClicked(java.awt.event.MouseEvent evt) {  
    enableEditAddPanel(false); // TODO add your handling code here:  
}  
  
// Variables declaration - do not modify  
private javax.swing.JPanel bannerPanel;  
private javax.swing.JLabel btnAdd;  
private javax.swing.JLabel btnCancel;  
private javax.swing.JLabel btnEdit;  
private javax.swing.JLabel btnExport;  
private javax.swing.JLabel btnRemove;  
private javax.swing.JLabel btnReset;  
private javax.swing.JLabel btnSave;  
private javax.swing.JLabel btnSearchReset;  
private javax.swing.JLabel btnStudentChooser;  
private javax.swing.JComboBox comboMonthSearch;  
private javax.swing.JComboBox comboSearchSubject;  
private javax.swing.JComboBox comboSubject;  
private com.toedter.calendar.JDateChooser dPicTime;  
private javax.swing.JPanel editAddPanel;  
private javax.swing.JLabel jLabel1;  
private javax.swing.JLabel jLabel14;
```

```
private javax.swing.JLabel jLabel15;
private javax.swing.JLabel jLabel16;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel4;
private javax.swing.JLabel jLabel44;
private javax.swing.JLabel jLabel47;
private javax.swing.JLabel jLabel5;
private javax.swing.JLabel jLabel6;
private javax.swing.JLabel jLabel7;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JLabel labelAction;
private javax.swing.JLabel labelStudentNo;
private javax.swing.JLabel labelStudentNo1;
private javax.swing.JLabel labelStudentNo2;
private javax.swing.JTable tblMark;
private javax.swing.JTextField txtPoint;
private javax.swing.JTextField txtSearchStudentName;
private javax.swing.JTextField txtSearchStudentNo;
private javax.swing.JTextField txtYearSearch;
// End of variables declaration

@Override
public void enableEditAddPanel(boolean b) {
    if (b == false) {
        tblMark.setEnabled(true);
        labelStudentNo.setText("");
        txtPoint.setText("");
        dPicTime.setDate(null);

        editAddPanel.setVisible(false);

        bannerPanel.setVisible(true);
    } else {
        if (act == EDIT) {
            labelAction.setText("Edit Mark");
            loadEdit();
        } else if (act == ADD) {
            labelAction.setText("Add Mark");
        }
        tblMark.setEnabled(false);
        editAddPanel.setVisible(true);
        btnReset.setVisible(true);

        bannerPanel.setVisible(false);
    }
}
```

```
        }  
    }  
  
    @Override  
    public void enableButton(boolean b) {  
        btnEdit.setEnabled(b);  
        btnRemove.setEnabled(b);  
    }  
  
    @Override  
    public void loadEdit() {  
        mark = entity.findById(entity.getSelectID(tblMark));  
        labelStudentNo.setText(mark.getStudent().getNo());  
        comboSubject.setSelectedItem(mark.getSubject().getName());  
        txtPoint.setText(mark.getPoint() + "");  
        dPicTime.setDate(mark.getTime());  
    }  
  
    @Override  
    public void search() {  
        throw new UnsupportedOperationException("Not supported yet."); //To change body of  
generated methods, choose Tools | Templates.  
    }  
  
    @Override  
    public void save(){  
        if(!validateForm()){  
            return;  
        }  
        if(mark==null){  
            mark=new Mark();  
        }  
        Subject sub=subjectEntity.findByName(comboSubject.getSelectedItem().toString());  
        int point=0;  
        Date date=null;  
        try{  
            point=Integer.parseInt(txtPoint.getText());  
        }catch(NumberFormatException ex){}  
        date=dPicTime.getDate();  
        mark.setSubject(sub);  
        mark.setPoint(point);  
        mark.setTime(date);  
        entity.setMark(mark);  
        int res=entity.save();  
        initUI();  
    }  
}
```

```

@Override
public void deactivate() {

}

@Override
public boolean validateForm() {
    if(mark==null || mark.getStudent()==null || mark.getStudent().getID() <1){
        JOptionPane.showMessageDialog(this, "You must be choose Student !");
        return false;
    }
    if(comboSubject.getSelectedIndex()==0){
        JOptionPane.showMessageDialog(this, "You must be choose subject !");
        return false;
    }
    return true;
}
}

```

- **For Student Management Module of AeonSMS**

```

/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */
package StudentMan;

import FileExport.ExportFile;
import Controls.Manager;
import Controls.StudentManager;
import Controls.Validate;
import MainObjects.ExtraCurricularEntity;
import MainObjects.StudentEntity;
import java.util.List;
import java.util.Vector;
import javax.swing.JOptionPane;
import javax.swing.JPanel;
import javax.swing.JTable;
import javax.swing.plaf.basic.BasicInternalFrameUI;
import javax.swing.table.DefaultTableModel;
import TableModel.Admission;
import TableModel.Batch;
import TableModel.ExtraCurricular;
import TableModel.FamilyStudent;

```

```
import TableModel.Student;
import TableModel.Mark;
import TableModel.Relative;
import TableModel.Remark;
import MainHome.UIInterface;
import Employee.RelationDetail;
import Employee.RelationUpdate;

/**
 *
 * @author Aeon(Ankit RudrAshokJha)
 */
public class ListStudent extends javax.swing.JInternalFrame implements UIInterface{
    StudentManager studentManager=new StudentManager();
    private ExtraCurricularEntity extraCurricularEntity;
    private StudentEntity entity;
    private int act;
    private int currentPage=1;
    private int totalPage=3;
    Student currenStudent;
    Student newStudent;
    /**
     * Creates new form ListStudent
     */
    public ListStudent() {
        initComponents();
    }
    public ListStudent(int w, int h ) {
        this.extraCurricularEntity=new ExtraCurricularEntity();
        this.entity=new StudentEntity();
        initComponents();
        initUI();
        setSize(w, h+30);
    }
    public void initUI(){
        enableButton(false);
        showPanel("List Student", listPanel,detailPanel,updatePanel);
        ((BasicInternalFrameUI)this.getUI()).setNorthPane(null);
        studentManager.loadCombobox("SELECT name FROM batch", comboBatchSearch);
        studentManager.loadTable(studentManager.getStudentList(currentPage), tblStudent,
        Manager.STUDENT_MODEL);
    }
    public void showPanel(String name, JPanel panel,JPanel panel1,JPanel panel2) {
        panel.setVisible(true);
        panel1.setVisible(false);
        panel2.setVisible(false);
    }
}
```

```
}

@Override
public void enableEditAddPanel(boolean b) {
    throw new UnsupportedOperationException("Not supported yet."); //To change body of
generated methods, choose Tools | Templates.
}

public void add() {
    int res = studentManager.addStudent(newStudent);
    studentManager.showMessage(res, this);
}

public void edit() {
    int res = studentManager.editStudent(newStudent, currenStudent);
    studentManager.showMessage(res, this);
}

@Override
public void deactive() {
    int res=studentManager.deactiveStudent(studentManager.getSelectID(tblStudent));
    entity.showMessage(res, this);
    if(res==Manager.SUCCESS){
        initUI();
    }
}
private void reset(){
    newStudent = currenStudent;
    if (act == EDIT) {
        studentManager.refreshTable(tblMark);
        studentManager.refreshTable(tblRelation);
        studentManager.refreshTable(tblRemark);
        loadEdit();
    } else if (act == ADD) {
        loadAdd();
    }
}
@Override
public void save(){
    newStudent.setName(txtName.getText());
    newStudent.setAddress(txtAddress.getText());
    newStudent.setEmail(txtEmail.getText());
    newStudent.setGender(studentManager.formatGender((String)
comboGender.getSelectedItem()));
    newStudent.setPhone(txtPhone.getText());
```

```

newStudent.setNo(txtNo.getText());
newStudent.setAdmission(new Admission( admiDPic.getDate(),
txtPrevInstituteEdit.getText(), txtReasonLeavingEdit.getText()));
newStudent.getAcademic().setBatch(new Batch((String)comboBatch.getSelectedItem()));

newStudent.setGender(studentManager.formatGender((String)comboGender.getSelectedItem()));
newStudent.setBirthday(dPic.getDate());
if (newStudent.getFamilyStudent() == null) {
    newStudent.setFamilyStudent(new FamilyStudent());
}
if (act == EDIT) {
    edit();
} else if (act == ADD) {
    add();
}

}
initUI();

return;
}

@Override
public void search(){
    if(!Validate.isEmpty(txtRegisterNoSearch.getText()) &&
!Validate.isNumber(txtRegisterNoSearch.getText())){
        JOptionPane.showMessageDialog(this, "Register No must be number");
        return;
    }
    String SQL = "SELECT student.ID, student.academic_register_No, student.No,
student.name,batch.name AS Batch, student.phone, student.email, student.gender "
        + "from student LEFT JOIN (academic LEFT JOIN batch on
academic.batch_ID=batch.ID) ON academic.register_No=student.academic_register_No WHERE
student.status=1 ";
    if(!txtRegisterNoSearch.getText().isEmpty()){
        SQL+=" AND
student.academic_register_No="+Integer.parseInt(txtRegisterNoSearch.getText());
    }
    if(!txtNoSearch.getText().isEmpty()){
        SQL+=" AND student.No='"+txtNoSearch.getText()+"'";
    }
    if(!txtNameSearch.getText().isEmpty()){
        SQL+=" AND student.name='"+txtNameSearch.getText()+"'";
    }
    if(comboBatchSearch.getSelectedIndex() !=0){
        SQL+=" AND batch.name='"+(String)comboBatchSearch.getSelectedItem()+"'";
    }
}

```

```

studentManager.loadTable(SQL, tblStudent, Manager.STUDENT_MODEL);
enableButton(false);
}
@Override
public void loadEdit(){
    lebUp.setText("Update Student Detail");
    btnDeleteMark.setEnabled(false);
    btnEditMark.setEnabled(false);
    btnDeleteRelation.setEnabled(false);
    btnEditRelation.setEnabled(false);
    btnDeleteRemark.setEnabled(false);
    btnEditRemark.setEnabled(false);

    loadMarkTbl(tblMark);
    loadRelationTbl(tblRelation);
    loadRemarkTbl(tblRemark);
    loadExtraTbl(tblExtraCurricular);
    studentManager.loadCombobox("SELECT name FROM batch", comboBatch);

    txtRegisterNo.setText(currenStudent.getRegisterNo()+"");
    txtAddress.setText(currenStudent.getAddress());
    dPic.setDate(currenStudent.getBirthday());
    txtEmail.setText(currenStudent.getEmail());
    txtPhone.setText(currenStudent.getPhone());
    txtName.setText(currenStudent.getName());
    txtNo.setText(currenStudent.getNo());
    if(currenStudent.getAdmission() !=null){
        txtPrevInstituteEdit.setText(currenStudent.getAdmission().getPrev_Institute());
        txtReasonLeavingEdit.setText(currenStudent.getAdmission().getReason_Leaving());
        admiDPic.setDate(currenStudent.getAdmission().getTime());
    }
    comboGender.setSelectedItem(studentManager.parseGender(currenStudent.getGender()));

    if(currenStudent.getAcademic() !=null && currenStudent.getAcademic().getBatch() !=null)
        comboBatch.setSelectedItem(currenStudent.getAcademic().getBatch().getName());
}

private void loadAdd(){
    lebUp.setText("Add New Student");
    btnDeleteMark.setEnabled(false);
    btnEditMark.setEnabled(false);
    btnDeleteRelation.setEnabled(false);
    btnEditRelation.setEnabled(false);
    btnDeleteRemark.setEnabled(false);
    btnEditRemark.setEnabled(false);
}

```

```
studentManager.refreshTable(tblMark);
studentManager.refreshTable(tblRelation);
studentManager.refreshTable(tblRemark);
studentManager.loadCombobox("SELECT name FROM batch", comboBatch);

txtAddress.setText("");
dPic.setDate(null);
txtEmail.setText("");
txtName.setText("");
txtNo.setText("");
txtPrevInstituteEdit.setText("");
txtReasonLeavingEdit.setText("");
admiDPic.setDate(null);
}

private void loadDetail(){

if(currenStudent==null)
    return;
btnRelationDetail.setEnabled(false);
btnExtraDetail.setEnabled(false);

loadMarkTbl(tblExtraCurricularDetail);
loadRelationTbl(tblRelationDetail);
loadRemarkTbl(tblRemarkDetail);
loadExtraTbl(tblExtraCurricularDetail);

labelAddress.setText(currenStudent.getAddress());
if(currenStudent.getAdmission() !=null){

labelAdmissionTime.setText(studentManager.formatDate(currenStudent.getAdmission().getTime()));
labelPrevInstitute.setText(currenStudent.getAdmission().getPrev_Institute());
labelReasonLeaving.setText(currenStudent.getAdmission().getReason_Leaving());
}

if(currenStudent.getAcademic() !=null){
    labelRegisterNo.setText(currenStudent.getAcademic().getRegisterNo()+"");
    if(currenStudent.getAcademic().getBatch() !=null){
        labelBatch.setText(currenStudent.getAcademic().getBatch().getName());
    }
}
labelBirthday.setText(studentManager.formatDate(currenStudent.getBirthday()));
labelEmail1.setText(currenStudent.getEmail());
labelGender.setText(studentManager.parseGender(currenStudent.getGender()));
labelName.setText(currenStudent.getName());
labelNo.setText(currenStudent.getNo());
labelPhone.setText(currenStudent.getPhone());
```

```

    }
    private void loadMarkTbl(JTable tbl){
        List<Mark> marks=currenStudent.getMarks();
        if(marks==null || marks.size() <1){
            return;
        }
        int count=0;
        String[][] data=new String[marks.size()][4];
        for(Mark m:marks){
            data[count++]=new String[]{
                m.getID()+"",
                m.getSubject().getName(),
                m.getPoint()+"",
                studentManager.formatDate(m.getTime())
            };
        }
        studentManager.loadTable(data, tbl);
    }
    private void loadRelationTbl(JTable tbl){
        int count = 0;
        if (currenStudent.getFamilyStudent()== null ||
currenStudent.getFamilyStudent().getListRelative()== null) {
            return;
        }
        List<Relative> listreRelatives = currenStudent.getFamilyStudent().getListRelative();
        String[][] data = new String[listreRelatives.size()][7];
        for (Relative r : listreRelatives) {
            data[count++] = new String[]{
                r.getID() + "",
                r.getRelation(),
                r.getName(),
                r.getPhone(),
                r.getAddress(),
                r.getOccupation(),
                studentManager.parseGender(r.getGender())
            };
        }
        studentManager.loadTable(data, tbl);
    }
    private void loadRemarkTbl(JTable tbl){
        int count = 0;
        if (currenStudent.getRemarks()== null || currenStudent.getRemarks().size()<1) {
            return;
        }
        List<Remark> remarks = currenStudent.getRemarks();
    }
}

```

```

String[][] data = new String[remarks.size()][4];
for (Remark r : remarks) {
    data[count++] = new String[]{{
        r.getID()+"",
        r.getStaff().getNo()+"",
        r.getStaff().getName(),
        r.getContent(),
        studentManager.formatDate(r.getTime())
    }};
}
studentManager.loadTable(data, tbl);
}

private void loadRemarkInfo(int mode){
    if (act == DETAIL) {
        new RemarkDetail(null, true,
currenStudent.getRemarks().get(tblRemarkDetail.getSelectedRow()),
studentManager).setVisible(true);
        return;
    }
    RemarkUpdate remarkUpdate = null;
    if (mode == ADD) {
        remarkUpdate = new RemarkUpdate(null, true, null, studentManager);
        remarkUpdate.setVisible(true);
    } else if (mode == EDIT) {
        remarkUpdate = new RemarkUpdate(null, true,
studentManager.getSelectRow(tblRemark), studentManager);
        remarkUpdate.setVisible(true);
        if (remarkUpdate.getRemark() != null) {
            removeSelectedRemark();
        }
    }
    if (remarkUpdate.getRemark() != null) {
        addRowTableRemark(remarkUpdate.getRemark());
        newStudent.addRemark(remarkUpdate.getRemark());
    }
}
private void removeSelectedRemark(){
    Vector row = studentManager.getSelectRow(tblRemark);
    newStudent.removeRemark(Integer.parseInt(row.get(1).toString()),
entity.parseDate(row.get(4).toString()));
    studentManager.removeSelectedRowTable(tblRemark);
}
private void addRowTableRemark(Remark r){
    Vector row = new Vector();
    row.add(r.getID());
    row.add(r.getStaff().getNo());
}

```

```

        row.add(r.getStaff().getName());
        row.add(r.getContent());
        row.add(studentManager.formatDate(r.getTime()));
        DefaultTableModel model = (DefaultTableModel) tblRemark.getModel();
        model.addRow(row);
    }

private void loadExtraTbl(JTable tbl){
    int count = 0;
    if (currenStudent.getExtraCurriculars() == null ||
currenStudent.getExtraCurriculars().size() < 1) {
        return;
    }
    List<ExtraCurricular> extraCurriculars = currenStudent.getExtraCurriculars();
    String[][] data = new String[extraCurriculars.size()][4];
    for (ExtraCurricular extra : extraCurriculars) {
        data[count++] = new String[]{
            extra.getID()+"",
            extra.getName(),
            extra.getLocation(),
            extraCurricularEntity.formatDate(extra.getStart()),
            extraCurricularEntity.formatDate(extra.getEnd()),
            extra.getNotes()
        };
    }
    studentManager.loadTable(data, tbl);
}

private void loadExtraInfo(int mode){
    if (act == DETAIL) {
        new ExtraCurricularDetail(null, true,
currenStudent.getExtraCurriculars().get(tblExtraCurricularDetail.getSelectedRow())).setVisible(true);
        return;
    }
    ExtraCurricularUpdate extraCurricularUpdate = null;
    if (mode == ADD) {
        extraCurricularUpdate = new ExtraCurricularUpdate(null, true, null);
        extraCurricularUpdate.setVisible(true);
    } else if (mode == EDIT) {
        extraCurricularUpdate = new ExtraCurricularUpdate(null, true,
entity.getSelectRow(tblExtraCurricular));
        extraCurricularUpdate.setVisible(true);
        if (extraCurricularUpdate.getExtraCurricular() != null) {
            removeSelectedRemark();
        }
    }
}

```

```

        if (extraCurricularUpdate.getExtraCurricular() != null) {
            addRowTableExtra(extraCurricularUpdate.getExtraCurricular());
            newStudent.addExtraCurricular(extraCurricularUpdate.getExtraCurricular());
        }
    }
    private void removeSelectedExtra(){
        Vector row = studentManager.getSelectRow(tblExtraCurricular);
        newStudent.removeExtraCurricular(row.get(0).toString(), row.get(1).toString(),
entity.parseDate(row.get(2).toString()), entity.parseDate(row.get(3).toString()));
        studentManager.removeSelectedRowTable(tblExtraCurricular);
    }
    private void addRowTableExtra(ExtraCurricular extra){
        Vector row = new Vector();
        row.add(extra.getID());
        row.add(extra.getName());
        row.add(extra.getLocation());
        row.add(entity.formatDate(extra.getStart()));
        row.add(entity.formatDate(extra.getEnd()));
        row.add(extra.getNotes());
        DefaultTableModel model = (DefaultTableModel)tblExtraCurricular.getModel();
        model.addRow(row);
    }
    private void loadRelationInfo(int mode){
        if (act == DETAIL) {
            new RelationDetail(null, true,
currenStudent.getFamilyStudent().getListRelative().get(tblRelationDetail.getSelectedRow())).setVisible(true);
            return;
        }
        RelationUpdate relationUpdate = null;
        if (mode == ADD) {
            relationUpdate = new RelationUpdate(null, true, studentManager, null);
            relationUpdate.setVisible(true);
        } else if (mode == EDIT) {
            relationUpdate = new RelationUpdate(null, true, studentManager,
studentManager.getSelectRow(tblRelation));
            relationUpdate.setVisible(true);
            if (relationUpdate.getRelaRelatives() != null) {
                removeSelectedRelation();
            }
        }
        if (relationUpdate.getRelaRelatives() != null) {
            addRowTableRelation(relationUpdate.getRelaRelatives());
            newStudent.getFamilyStudent().addRelative(relationUpdate.getRelaRelatives());
        }
    }
}

```

```

private void addRowTableRelation(Relative r) {
    Vector row = new Vector();
    row.add(r.getID());
    row.add(r.getRelation());
    row.add(r.getName());
    row.add(r.getPhone());
    row.add(r.getAddress());
    row.add(r.getOccupation());
    row.add(studentManager.parseGender(r.getGender()));
    DefaultTableModel model = (DefaultTableModel) tblRelation.getModel();
    model.addRow(row);
}

private void removeSelectedRelation() {
    Vector row = studentManager.getSelectRow(tblRelation);
    newStudent.getFamilyStudent().removeRelative((String) row.get(1), (String) row.get(2));
    studentManager.removeSelectedRowTable(tblRelation);
}

private void loadMarkinfo(int mode){

    MarkUpdate markUpdate=null;
    if(mode==EDIT){
        markUpdate=new MarkUpdate(null, true, studentManager.getSelectRow(tblMark),
studentManager);
        markUpdate.setVisible(true);
        if (markUpdate.getMark()!= null) {
            removeSelectedMark();
        }
    }else if(mode==ADD){
        markUpdate=new MarkUpdate(null, true, null, studentManager);
        markUpdate.setVisible(true);
    }
    if(markUpdate.getMark() !=null){
        addRowTableMark(markUpdate.getMark());
        newStudent.addMark(markUpdate.getMark());
    }
}

private void removeSelectedMark(){
    Vector row = studentManager.getSelectRow(tblMark);
    newStudent.removeMark((String)row.get(1),
studentManager.parseDouble(row.get(3).toString()));
    studentManager.removeSelectedRowTable(tblMark);
}

private void addRowTableMark(Mark mark){
    Vector row=new Vector();
    row.add(mark.getID());

```

```
row.add(mark.getSubject().getName());
row.add(mark.getPoint());
row.add(studentManager.formatDate(mark.getTime())));
DefaultTableModel model = (DefaultTableModel) tblMark.getModel();
model.addRow(row);
}
private void enableButton(Boolean b){ }
/**
 * 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">
private void initComponents() {

    updatePanel = new javax.swing.JPanel();
    jScrollPane4 = new javax.swing.JScrollPane();
    can = new javax.swing.JPanel();
    txtNo = new javax.swing.JTextField();
    comboBatch = new javax.swing.JComboBox();
    jLabel40 = new javax.swing.JLabel();
    jLabel42 = new javax.swing.JLabel();
    txtName = new javax.swing.JTextField();
    jLabel43 = new javax.swing.JLabel();
    jLabel51 = new javax.swing.JLabel();
    txtEmail = new javax.swing.JTextField();
    jLabel52 = new javax.swing.JLabel();
    jScrollPane6 = new javax.swing.JScrollPane();
    txtAddress = new javax.swing.JTextArea();
    jLabel53 = new javax.swing.JLabel();
    jLabel54 = new javax.swing.JLabel();
    jLabel55 = new javax.swing.JLabel();
    jScrollPane7 = new javax.swing.JScrollPane();
    tblMark = new javax.swing.JTable();
    comboGender = new javax.swing.JComboBox();
    lebUp = new javax.swing.JLabel();
    btnCancel = new javax.swing.JLabel();
    jScrollPane21 = new javax.swing.JScrollPane();
    tblRelation = new javax.swing.JTable();
    jLabel60 = new javax.swing.JLabel();
    jLabel61 = new javax.swing.JLabel();
    jLabel62 = new javax.swing.JLabel();
    txtPrevInstituteEdit = new javax.swing.JTextField();
    txtReasonLeavingEdit = new javax.swing.JTextField();
    jLabel63 = new javax.swing.JLabel();
```

```
jScrollPane22 = new javax.swing.JScrollPane();
tblRemark = new javax.swing.JTable();
jLabel3 = new javax.swing.JLabel();
txtPhone = new javax.swing.JTextField();
jLabel67 = new javax.swing.JLabel();
jScrollPane23 = new javax.swing.JScrollPane();
tblExtraCurricular = new javax.swing.JTable();
jLabel44 = new javax.swing.JLabel();
txtRegisterNo = new javax.swing.JTextField();
dPic = new com.toedter.calendar.JDateChooser();
admiDPic = new com.toedter.calendar.JDateChooser();
comboBatch1 = new javax.swing.JComboBox();
jLabel45 = new javax.swing.JLabel();
jLabel46 = new javax.swing.JLabel();
jLabel47 = new javax.swing.JLabel();
btnAddRelation = new javax.swing.JLabel();
btnEditRelation = new javax.swing.JLabel();
btnDeleteRelation = new javax.swing.JLabel();
btnEditMark = new javax.swing.JLabel();
btnDeleteMark = new javax.swing.JLabel();
btnAddMark = new javax.swing.JLabel();
btnDeleteRemark = new javax.swing.JLabel();
btnEditRemark = new javax.swing.JLabel();
btnAddRemark = new javax.swing.JLabel();
btnDeleteExtra = new javax.swing.JLabel();
btnEditExtra = new javax.swing.JLabel();
btnAddExtra = new javax.swing.JLabel();
jLabel58 = new javax.swing.JLabel();
btnSave = new javax.swing.JLabel();
btnReset = new javax.swing.JLabel();
jLabel49 = new javax.swing.JLabel();
listPanel = new javax.swing.JPanel();
jScrollPane1 = new javax.swing.JScrollPane();
tblStudent = new javax.swing.JTable();
jLabel19 = new javax.swing.JLabel();
jLabel64 = new javax.swing.JLabel();
jLabel65 = new javax.swing.JLabel();
jLabel66 = new javax.swing.JLabel();
txtRegisterNoSearch = new javax.swing.JTextField();
txtNoSearch = new javax.swing.JTextField();
txtNameSearch = new javax.swing.JTextField();
comboBatchSearch = new javax.swing.JComboBox();
jLabel5 = new javax.swing.JLabel();
jLabel68 = new javax.swing.JLabel();
jLabel69 = new javax.swing.JLabel();
jLabel70 = new javax.swing.JLabel();
```

```
btnEdit1 = new javax.swing.JLabel();
btnExport = new javax.swing.JLabel();
btnDetail = new javax.swing.JLabel();
btnDeactive = new javax.swing.JLabel();
detailPanel = new javax.swing.JPanel();
jScrollPane3 = new javax.swing.JScrollPane();
jPanel1 = new javax.swing.JPanel();
jLabel14 = new javax.swing.JLabel();
jLabel4 = new javax.swing.JLabel();
jLabel16 = new javax.swing.JLabel();
jLabel39 = new javax.swing.JLabel();
labelReasonLeaving = new javax.swing.JLabel();
jLabel20 = new javax.swing.JLabel();
labelNo = new javax.swing.JLabel();
labelBirthday = new javax.swing.JLabel();
labelPhone = new javax.swing.JLabel();
jLabel11 = new javax.swing.JLabel();
jLabel25 = new javax.swing.JLabel();
labelRegisterNo = new javax.swing.JLabel();
jScrollPane10 = new javax.swing.JScrollPane();
tblMarkDetail3 = new javax.swing.JTable();
jLabel8 = new javax.swing.JLabel();
jLabel1 = new javax.swing.JLabel();
jLabel6 = new javax.swing.JLabel();
jScrollPane18 = new javax.swing.JScrollPane();
tblRemarkDetail = new javax.swing.JTable();
jLabel2 = new javax.swing.JLabel();
labelName = new javax.swing.JLabel();
labelAdmissionTime = new javax.swing.JLabel();
labelBatch = new javax.swing.JLabel();
jLabel23 = new javax.swing.JLabel();
jScrollPane2 = new javax.swing.JScrollPane();
labelAddress = new javax.swing.JTextPane();
labelPrevInstitute = new javax.swing.JLabel();
jScrollPane17 = new javax.swing.JScrollPane();
tblRelationDetail = new javax.swing.JTable();
jScrollPane9 = new javax.swing.JScrollPane();
tblExtraCurricularDetail = new javax.swing.JTable();
jLabel10 = new javax.swing.JLabel();
labelGender = new javax.swing.JLabel();
jLabel30 = new javax.swing.JLabel();
btnDeactive1 = new javax.swing.JLabel();
jLabel28 = new javax.swing.JLabel();
labelEmail1 = new javax.swing.JLabel();
btnExtraDetail = new javax.swing.JLabel();
jLabel29 = new javax.swing.JLabel();
```

```
btnRemarkDetail = new javax.swing.JLabel();
btnRelationDetail = new javax.swing.JLabel();
jLabel48 = new javax.swing.JLabel();
btnViewList = new javax.swing.JLabel();
btnAdd = new javax.swing.JLabel();
btnEdit = new javax.swing.JLabel();
jLabel50 = new javax.swing.JLabel();

setBackground(new java.awt.Color(224, 224, 255));
setBorder(null);
setFrameIcon(null);
setPreferredSize(new java.awt.Dimension(0, 0));
getContentPane().setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

updatePanel.setMinimumSize(new java.awt.Dimension(1020, 700));
updatePanel.setPreferredSize(new java.awt.Dimension(1020, 700));

jScrollPane4.setBorder(null);

can.setBackground(new java.awt.Color(45, 129, 171));
can.setMinimumSize(new java.awt.Dimension(980, 1195));
can.setPreferredSize(new java.awt.Dimension(980, 1195));
can.setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

txtNo.setBackground(new java.awt.Color(85, 153, 187));
txtNo.setBorder(null);
can.add(txtNo, new org.netbeans.lib.awtextra.AbsoluteConstraints(240, 290, 180, 25));

can.add(comboBatch, new org.netbeans.lib.awtextra.AbsoluteConstraints(240, 260, 180, 25));

jLabel40.setBackground(new java.awt.Color(0, 102, 153));
jLabel40.setForeground(new java.awt.Color(255, 255, 255));
jLabel40.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
jLabel40.setText("Roll No");
jLabel40.setOpaque(true);
can.add(jLabel40, new org.netbeans.lib.awtextra.AbsoluteConstraints(150, 290, 90, 25));

jLabel42.setBackground(new java.awt.Color(0, 102, 153));
jLabel42.setForeground(new java.awt.Color(255, 255, 255));
jLabel42.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
jLabel42.setText("Class");
jLabel42.setOpaque(true);
can.add(jLabel42, new org.netbeans.lib.awtextra.AbsoluteConstraints(150, 260, 90, 25));

txtName.setBackground(new java.awt.Color(85, 153, 187));
```

```
txtName.setBorder(null);
can.add(txtName, new org.netbeans.lib.awtextra.AbsoluteConstraints(530, 200, 200, 25));

jLabel43.setBackground(new java.awt.Color(0, 102, 153));
jLabel43.setForeground(new java.awt.Color(255, 255, 255));
jLabel43.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
jLabel43.setText("Name");
jLabel43.setOpaque(true);
can.add(jLabel43, new org.netbeans.lib.awtextra.AbsoluteConstraints(440, 200, 88, 25));

jLabel51.setBackground(new java.awt.Color(0, 102, 153));
jLabel51.setForeground(new java.awt.Color(255, 255, 255));
jLabel51.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
jLabel51.setText("Email");
jLabel51.setOpaque(true);
can.add(jLabel51, new org.netbeans.lib.awtextra.AbsoluteConstraints(440, 290, 88, 25));

txtEmail.setBackground(new java.awt.Color(85, 153, 187));
txtEmail.setBorder(null);
can.add(txtEmail, new org.netbeans.lib.awtextra.AbsoluteConstraints(530, 290, 200, 25));

jLabel52.setBackground(new java.awt.Color(0, 102, 153));
jLabel52.setForeground(new java.awt.Color(255, 255, 255));
jLabel52.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
jLabel52.setText("Phone");
jLabel52.setOpaque(true);
can.add(jLabel52, new org.netbeans.lib.awtextra.AbsoluteConstraints(150, 320, 90, 25));

jScrollPane6.setBorder(null);
jScrollPane6.setForeground(new java.awt.Color(255, 255, 255));

txtAddress.setBackground(new java.awt.Color(85, 153, 187));
txtAddress.setColumns(20);
txtAddress.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N
txtAddress.setForeground(new java.awt.Color(255, 255, 255));
txtAddress.setRows(5);
txtAddress.setToolTipText("");
txtAddress.setBorder(null);
jScrollPane6.setViewportView(txtAddress);

can.add(jScrollPane6, new org.netbeans.lib.awtextra.AbsoluteConstraints(240, 420, 670,
90));

jLabel53.setBackground(new java.awt.Color(0, 102, 153));
jLabel53.setForeground(new java.awt.Color(255, 255, 255));
jLabel53.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
```

```
jLabel53.setText("D.O.B");
jLabel53.setOpaque(true);
can.add(jLabel53, new org.netbeans.lib.awtextra.AbsoluteConstraints(150, 230, 90, 25));

jLabel54.setBackground(new java.awt.Color(0, 102, 153));
jLabel54.setForeground(new java.awt.Color(255, 255, 255));
jLabel54.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
jLabel54.setText("Gender");
jLabel54.setOpaque(true);
can.add(jLabel54, new org.netbeans.lib.awtextra.AbsoluteConstraints(440, 230, 88, 25));

jLabel55.setBackground(new java.awt.Color(0, 102, 153));
jLabel55.setForeground(new java.awt.Color(255, 255, 255));
jLabel55.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
jLabel55.setText("Mark");
jLabel55.setOpaque(true);
can.add(jLabel55, new org.netbeans.lib.awtextra.AbsoluteConstraints(150, 680, 88, 25));

jScrollPane7.setBackground(new java.awt.Color(4, 74, 121));
jScrollPane7.setBorder(javax.swing.BorderFactory.createLineBorder(new java.awt.Color(4, 74, 121)));
jScrollPane7.setForeground(new java.awt.Color(255, 255, 255));

tblMark.setBackground(new java.awt.Color(4, 74, 121));
tblMark.setForeground(new java.awt.Color(255, 255, 255));
tblMark.setModel(new javax.swing.table.DefaultTableModel(
    new Object [][] {
        new String [] {
            "ID", "Subject", "Mark", "Time"
        }
    },
    new String [] {
        "ID", "Subject", "Mark", "Time"
    }
));
tblMark.setSelectionBackground(new java.awt.Color(0, 153, 102));
tblMark.setShowHorizontalLines(false);
tblMark.setShowVerticalLines(false);
tblMark.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        tblMarkMouseClicked(evt);
    }
});
tblMark.setRowHeight(25);

jScrollPane7.setViewportView(tblMark);

can.add(jScrollPane7, new org.netbeans.lib.awtextra.AbsoluteConstraints(150, 710, 760, 100));
```

```
comboGender.setBackground(new java.awt.Color(85, 153, 187));
comboGender.setModel(new javax.swing.DefaultComboBoxModel(new String[] { "Male",
"Female", "Other" }));
comboGender.setBorder(null);
can.add(comboGender, new org.netbeans.lib.awtextra.AbsoluteConstraints(530, 230, 200,
25));

lebUp.setBackground(new java.awt.Color(0, 102, 153));
lebUp.setFont(new java.awt.Font("Times New Roman", 1, 24)); // NOI18N
lebUp.setForeground(new java.awt.Color(255, 255, 255));
lebUp.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
lebUp.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/Images/icons8_Student_Male_38px.png"))); // NOI18N
lebUp.setText("Update Student Details");
lebUp.setIconTextGap(20);
lebUp.setOpaque(true);
can.add(lebUp, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 0, 1030, 60));

btnCancel.setBackground(new java.awt.Color(255, 0, 0));
btnCancel.setForeground(new java.awt.Color(255, 255, 255));
btnCancel.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
btnCancel.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/Images/icons8_Close_Window_17px.png"))); // NOI18N
btnCancel.setText("Cancel");
btnCancel.setOpaque(true);
btnCancel.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        btnCancelMouseClicked(evt);
    }
});
can.add(btnCancel, new org.netbeans.lib.awtextra.AbsoluteConstraints(820, 1140, 88, 25));

jScrollPane21.setBackground(new java.awt.Color(4, 74, 121));
jScrollPane21.setBorder(javax.swing.BorderFactory.createLineBorder(new
java.awt.Color(4, 74, 121)));
jScrollPane21.setForeground(new java.awt.Color(255, 255, 255));

tblRelation.setBackground(new java.awt.Color(4, 74, 121));
tblRelation.setForeground(new java.awt.Color(255, 255, 255));
tblRelation.setModel(new javax.swing.table.DefaultTableModel(
new Object [][] {
},
new String [] {
}
,
```

```
        "ID", "Relation", "Name", "Phone", "Address", "Occupation", "Gender"
    }
));
tblRelation.setSelectionBackground(new java.awt.Color(0, 153, 102));
tblRelation.setShowHorizontalLines(false);
tblRelation.setShowVerticalLines(false);
tblRelation.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        tblRelationMouseClicked(evt);
    }
});
jScrollPane21.setViewportView(tblRelation);

can.add(jScrollPane21, new org.netbeans.lib.awtextra.AbsoluteConstraints(150, 1020, 760,
100));

jLabel60.setBackground(new java.awt.Color(0, 102, 153));
jLabel60.setForeground(new java.awt.Color(255, 255, 255));
jLabel60.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
jLabel60.setText("Last School");
jLabel60.setOpaque(true);
can.add(jLabel60, new org.netbeans.lib.awtextra.AbsoluteConstraints(150, 350, 90, 25));

jLabel61.setBackground(new java.awt.Color(0, 102, 153));
jLabel61.setForeground(new java.awt.Color(255, 255, 255));
jLabel61.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
jLabel61.setText("Reason leaving");
jLabel61.setOpaque(true);
can.add(jLabel61, new org.netbeans.lib.awtextra.AbsoluteConstraints(150, 380, 90, 25));

jLabel62.setBackground(new java.awt.Color(0, 102, 153));
jLabel62.setForeground(new java.awt.Color(255, 255, 255));
jLabel62.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
jLabel62.setText("Time");
jLabel62.setOpaque(true);
can.add(jLabel62, new org.netbeans.lib.awtextra.AbsoluteConstraints(440, 320, 88, 25));

txtPrevInstituteEdit.setBackground(new java.awt.Color(85, 153, 187));
txtPrevInstituteEdit.setBorder(null);
can.add(txtPrevInstituteEdit, new org.netbeans.lib.awtextra.AbsoluteConstraints(240, 350,
495, 25));

txtReasonLeavingEdit.setBackground(new java.awt.Color(85, 153, 187));
txtReasonLeavingEdit.setBorder(null);
can.add(txtReasonLeavingEdit, new org.netbeans.lib.awtextra.AbsoluteConstraints(240, 380,
495, 25));
```

```
jLabel63.setBackground(new java.awt.Color(0, 102, 153));
jLabel63.setForeground(new java.awt.Color(255, 255, 255));
jLabel63.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
jLabel63.setText("Remark");
jLabel63.setOpaque(true);
can.add(jLabel63, new org.netbeans.lib.awtextra.AbsoluteConstraints(150, 830, 88, 25));

jScrollPane22.setBackground(new java.awt.Color(4, 74, 121));
jScrollPane22.setBorder(javax.swing.BorderFactory.createLineBorder(new
java.awt.Color(4, 74, 121)));
jScrollPane22.setForeground(new java.awt.Color(255, 255, 255));

tblRemark.setBackground(new java.awt.Color(4, 74, 121));
tblRemark.setForeground(new java.awt.Color(255, 255, 255));
tblRemark.setModel(new javax.swing.table.DefaultTableModel(
new Object [][] {
},
new String [] {
    "ID", "Staff No", "Staff name", "Content", "Time"
})
);
tblRemark.setSelectionBackground(new java.awt.Color(0, 153, 102));
tblRemark.setShowHorizontalLines(false);
tblRemark.setShowVerticalLines(false);
tblRemark.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        tblRemarkMouseClicked(evt);
    }
});
jScrollPane22.setViewportView(tblRemark);

can.add(jScrollPane22, new org.netbeans.lib.awtextra.AbsoluteConstraints(150, 860, 760,
100));

jLabel3.setBackground(new java.awt.Color(0, 102, 153));
jLabel3.setForeground(new java.awt.Color(255, 255, 255));
jLabel3.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
jLabel3.setText("Address");
jLabel3.setOpaque(true);
can.add(jLabel3, new org.netbeans.lib.awtextra.AbsoluteConstraints(150, 420, 90, 90));

txtPhone.setBackground(new java.awt.Color(85, 153, 187));
txtPhone.setBorder(null);
can.add(txtPhone, new org.netbeans.lib.awtextra.AbsoluteConstraints(240, 320, 180, 25));
```

```
jLabel67.setBackground(new java.awt.Color(0, 102, 153));
jLabel67.setForeground(new java.awt.Color(255, 255, 255));
jLabel67.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
jLabel67.setText("Extra Curricular");
jLabel67.setOpaque(true);
can.add(jLabel67, new org.netbeans.lib.awtextra.AbsoluteConstraints(150, 560, 88, 25));

jScrollPane23.setBackground(new java.awt.Color(4, 74, 121));
jScrollPane23.setBorder(javax.swing.BorderFactory.createLineBorder(new
java.awt.Color(4, 74, 121)));
jScrollPane23.setForeground(new java.awt.Color(255, 255, 255));

tblExtraCurricular.setBackground(new java.awt.Color(4, 74, 121));
tblExtraCurricular.setForeground(new java.awt.Color(255, 255, 255));
tblExtraCurricular.setModel(new javax.swing.table.DefaultTableModel(
new Object [][] {
},
new String [] {
    "ID", "Name", "Location", "Start", "End", "Notes"
})
);
tblExtraCurricular.setSelectionBackground(new java.awt.Color(0, 153, 102));
tblExtraCurricular.setShowHorizontalLines(false);
tblExtraCurricular.setShowVerticalLines(false);
tblExtraCurricular.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        tblExtraCurricularMouseClicked(evt);
    }
});
jScrollPane23.setViewportView(tblExtraCurricular);

can.add(jScrollPane23, new org.netbeans.lib.awtextra.AbsoluteConstraints(150, 590, 760,
75));

jLabel44.setBackground(new java.awt.Color(0, 102, 153));
jLabel44.setForeground(new java.awt.Color(255, 255, 255));
jLabel44.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
jLabel44.setText("Register No");
jLabel44.setOpaque(true);
can.add(jLabel44, new org.netbeans.lib.awtextra.AbsoluteConstraints(150, 200, 90, 25));

txtRegisterNo.setEditable(false);
txtRegisterNo.setBackground(new java.awt.Color(85, 153, 187));
txtRegisterNo.setBorder(null);
```

```
can.add(txtRegisterNo, new org.netbeans.lib.awtextra.AbsoluteConstraints(240, 200, 180, 25));
can.add(dPic, new org.netbeans.lib.awtextra.AbsoluteConstraints(240, 230, 180, 25));
can.add(admiDPic, new org.netbeans.lib.awtextra.AbsoluteConstraints(530, 320, 200, 25));

comboBatch1.setBackground(new java.awt.Color(85, 153, 187));
comboBatch1.setBorder(null);
can.add(comboBatch1, new org.netbeans.lib.awtextra.AbsoluteConstraints(530, 260, 200, 25));

jLabel45.setBackground(new java.awt.Color(0, 102, 153));
jLabel45.setForeground(new java.awt.Color(255, 255, 255));
jLabel45.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
jLabel45.setText("Batch");
jLabel45.setOpaque(true);
can.add(jLabel45, new org.netbeans.lib.awtextra.AbsoluteConstraints(440, 260, 88, 25));

jLabel46.setBackground(new java.awt.Color(85, 153, 187));
jLabel46.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
jLabel46.setOpaque(true);
can.add(jLabel46, new org.netbeans.lib.awtextra.AbsoluteConstraints(770, 200, 140, 180));

jLabel47.setBackground(new java.awt.Color(0, 102, 153));
jLabel47.setForeground(new java.awt.Color(255, 255, 255));
jLabel47.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
jLabel47.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/Images/icons8_Browse_Folder_18px.png")));
// NOI18N
jLabel47.setText("Browse");
jLabel47.setOpaque(true);
can.add(jLabel47, new org.netbeans.lib.awtextra.AbsoluteConstraints(770, 380, 140, 25));

btnAddRelation.setBackground(new java.awt.Color(224, 224, 255));
btnAddRelation.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
btnAddRelation.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/Images/icons8_Plus_Math_20px_1.png")));
// NOI18N
btnAddRelation.setBorder(javax.swing.BorderFactory.createLineBorder(new
java.awt.Color(0, 0, 0)));
btnAddRelation.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        btnAddRelationMouseClicked(evt);
    }
});
can.add(btnAddRelation, new org.netbeans.lib.awtextra.AbsoluteConstraints(800, 990, 30, 25));
```

```
btnEditRelation.setBackground(new java.awt.Color(0, 68, 102));
btnEditRelation.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
btnEditRelation.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/Images/icons8_Edit_20px.png"))); // NOI18N
btnEditRelation.setBorder(javax.swing.BorderFactory.createLineBorder(new
java.awt.Color(0, 0, 0)));
btnEditRelation.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        btnEditRelationMouseClicked(evt);
    }
});
can.add(btnEditRelation, new org.netbeans.lib.awtextra.AbsoluteConstraints(840, 990, 30,
25));

btnDeleteRelation.setBackground(new java.awt.Color(0, 68, 102));
btnDeleteRelation.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
btnDeleteRelation.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/Images/icons8_Trash_20px.png"))); // NOI18N
btnDeleteRelation.setBorder(javax.swing.BorderFactory.createLineBorder(new
java.awt.Color(0, 0, 0)));
btnDeleteRelation.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        btnDeleteRelationMouseClicked(evt);
    }
});
can.add(btnDeleteRelation, new org.netbeans.lib.awtextra.AbsoluteConstraints(880, 990, 30,
25));

btnEditMark.setBackground(new java.awt.Color(0, 68, 102));
btnEditMark.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
btnEditMark.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/Images/icons8_Edit_20px.png"))); // NOI18N
btnEditMark.setBorder(javax.swing.BorderFactory.createLineBorder(new java.awt.Color(0,
0, 0)));
btnEditMark.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        btnEditMarkMouseClicked(evt);
    }
});
can.add(btnEditMark, new org.netbeans.lib.awtextra.AbsoluteConstraints(840, 680, 30, 25));

btnDeleteMark.setBackground(new java.awt.Color(0, 68, 102));
btnDeleteMark.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
btnDeleteMark.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/Images/icons8_Trash_20px.png"))); // NOI18N
```

```
btnDeleteMark.setBorder(javax.swing.BorderFactory.createLineBorder(new
java.awt.Color(0, 0, 0)));
btnDeleteMark.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        btnDeleteMarkMouseClicked(evt);
    }
});
can.add(btnDeleteMark, new org.netbeans.lib.awtextra.AbsoluteConstraints(880, 680, 30,
25));

btnAddMark.setBackground(new java.awt.Color(224, 224, 255));
btnAddMark.setHorizontalTextPosition(javax.swing.SwingConstants.CENTER);
btnAddMark.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/Images/icons8_Plus_Math_20px_1.png"))); // NOI18N
btnAddMark.setBorder(javax.swing.BorderFactory.createLineBorder(new java.awt.Color(0,
0, 0)));
btnAddMark.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        btnAddMarkMouseClicked(evt);
    }
});
can.add(btnAddMark, new org.netbeans.lib.awtextra.AbsoluteConstraints(800, 680, 30, 25));

btnDeleteRemark.setBackground(new java.awt.Color(0, 68, 102));
btnDeleteRemark.setHorizontalTextPosition(javax.swing.SwingConstants.CENTER);
btnDeleteRemark.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/Images/icons8_Trash_20px.png"))); // NOI18N
btnDeleteRemark.setBorder(javax.swing.BorderFactory.createLineBorder(new
java.awt.Color(0, 0, 0)));
btnDeleteRemark.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        btnDeleteRemarkMouseClicked(evt);
    }
});
can.add(btnDeleteRemark, new org.netbeans.lib.awtextra.AbsoluteConstraints(880, 830, 30,
25));

btnEditRemark.setBackground(new java.awt.Color(0, 68, 102));
btnEditRemark.setHorizontalTextPosition(javax.swing.SwingConstants.CENTER);
btnEditRemark.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/Images/icons8_Edit_20px.png"))); // NOI18N
btnEditRemark.setBorder(javax.swing.BorderFactory.createLineBorder(new
java.awt.Color(0, 0, 0)));
btnEditRemark.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
```

```
        btnEditRemarkMouseClicked(evt);
    }
});
can.add(btnEditRemark, new org.netbeans.lib.awtextra.AbsoluteConstraints(840, 830, 30,
25));

btnAddRemark.setBackground(new java.awt.Color(224, 224, 255));
btnAddRemark.setHorizontalTextPosition(javax.swing.SwingConstants.CENTER);
btnAddRemark.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/Images/icons8_Plus_Math_20px_1.png"))); // NOI18N
btnAddRemark.setBorder(javax.swing.BorderFactory.createLineBorder(new
java.awt.Color(0, 0, 0)));
btnAddRemark.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        btnAddRemarkMouseClicked(evt);
    }
});
can.add(btnAddRemark, new org.netbeans.lib.awtextra.AbsoluteConstraints(800, 830, 30,
25));

btnDeleteExtra.setBackground(new java.awt.Color(0, 68, 102));
btnDeleteExtra.setHorizontalTextPosition(javax.swing.SwingConstants.CENTER);
btnDeleteExtra.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/Images/icons8_Trash_20px.png"))); // NOI18N
btnDeleteExtra.setBorder(javax.swing.BorderFactory.createLineBorder(new
java.awt.Color(0, 0, 0)));
btnDeleteExtra.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        btnDeleteExtraMouseClicked(evt);
    }
});
can.add(btnDeleteExtra, new org.netbeans.lib.awtextra.AbsoluteConstraints(880, 560, 30,
25));

btnEditExtra.setBackground(new java.awt.Color(0, 68, 102));
btnEditExtra.setHorizontalTextPosition(javax.swing.SwingConstants.CENTER);
btnEditExtra.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/Images/icons8_Edit_20px.png"))); // NOI18N
btnEditExtra.setBorder(javax.swing.BorderFactory.createLineBorder(new java.awt.Color(0,
0, 0)));
btnEditExtra.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        btnEditExtraMouseClicked(evt);
    }
});
}
```

```
can.add(btnEditExtra, new org.netbeans.lib.awtextra.AbsoluteConstraints(840, 560, 30, 25));

btnAddExtra.setBackground(new java.awt.Color(224, 224, 255));
btnAddExtra.setHorizontalTextPosition(javax.swing.SwingConstants.CENTER);
btnAddExtra.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/Images/icons8_Plus_Math_20px_1.png"))); // NOI18N
btnAddExtra.setBorder(javax.swing.BorderFactory.createLineBorder(new java.awt.Color(0,
0, 0)));
btnAddExtra.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        btnAddExtraMouseClicked(evt);
    }
});
can.add(btnAddExtra, new org.netbeans.lib.awtextra.AbsoluteConstraints(800, 560, 30, 25));

jLabel58.setBackground(new java.awt.Color(0, 102, 153));
jLabel58.setForeground(new java.awt.Color(255, 255, 255));
jLabel58.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
jLabel58.setText("Family ");
jLabel58.setOpaque(true);
can.add(jLabel58, new org.netbeans.lib.awtextra.AbsoluteConstraints(150, 990, 88, 25));

btnSave.setBackground(new java.awt.Color(51, 204, 0));
btnSave.setForeground(new java.awt.Color(255, 255, 255));
btnSave.setHorizontalTextPosition(javax.swing.SwingConstants.CENTER);
btnSave.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/Images/icons8_Save_18px.png"))); // NOI18N
btnSave.setText("Save");
btnSave.setOpaque(true);
btnSave.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        btnSaveMouseClicked(evt);
    }
});
can.add(btnSave, new org.netbeans.lib.awtextra.AbsoluteConstraints(150, 1140, 88, 25));

btnReset.setBackground(new java.awt.Color(0, 102, 255));
btnReset.setForeground(new java.awt.Color(255, 255, 255));
btnReset.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
btnReset.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/Images/icons8_Refresh_17px.png"))); // NOI18N
btnReset.setText("Reset");
btnReset.setOpaque(true);
btnReset.addMouseListener(new java.awt.event.MouseAdapter() {
```

```
public void mouseClicked(java.awt.event.MouseEvent evt) {
    btnResetMouseClicked(evt);
}
});
can.add(btnReset, new org.netbeans.lib.awtextra.AbsoluteConstraints(510, 1140, 88, 25));

jLabel49.setBackground(new java.awt.Color(0, 102, 0));
jLabel49.setFont(new java.awt.Font("Tahoma", 1, 12)); // NOI18N
jLabel49.setForeground(new java.awt.Color(255, 255, 255));
jLabel49.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
jLabel49.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/Images/icons8_Undo_30px.png"))); // NOI18N
jLabel49.setText("Back");
jLabel49.setIconTextGap(9);
jLabel49.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        jLabel49MouseClicked(evt);
    }
});
can.add(jLabel49, new org.netbeans.lib.awtextra.AbsoluteConstraints(20, 70, 90, 30));

jScrollPane4.setViewportView(can);

javax.swing.GroupLayout updatePanelLayout = new
javax.swing.GroupLayout(updatePanel);
updatePanel.setLayout(updatePanelLayout);
updatePanelLayout.setHorizontalGroup(
    updatePanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(updatePanelLayout.createSequentialGroup()
            .addComponent(jScrollPane4, javax.swing.GroupLayout.DEFAULT_SIZE, 1010,
Short.MAX_VALUE)
            .addGap(10, 10, 10)
        )
);
updatePanelLayout.setVerticalGroup(
    updatePanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(updatePanelLayout.createSequentialGroup()
            .addComponent(jScrollPane4, javax.swing.GroupLayout.DEFAULT_SIZE, 700, Short.MAX_VALUE)
            .addGap(10, 10, 10)
        )
);

getContentPane().add(updatePanel, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 0,
1020, 700));

listPanel.setBackground(new java.awt.Color(45, 129, 171));
listPanel.setMinimumSize(new java.awt.Dimension(1020, 700));
listPanel.setPreferredSize(new java.awt.Dimension(1020, 700));
listPanel.setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());
```

```
jScrollPane1.setBackground(new java.awt.Color(224, 224, 255));
jScrollPane1.setBorder(javax.swing.BorderFactory.createLineBorder(new java.awt.Color(0,
102, 153)));

tblStudent.setBackground(new java.awt.Color(4, 74, 121));
tblStudent.setForeground(new java.awt.Color(255, 255, 255));
tblStudent.setModel(new javax.swing.table.DefaultTableModel(
new Object [][] {
    {null, null, null, null, null, null, null, null},
    {null, null, null, null, null, null, null, null}
},
new String [] {
    "ID", "Register No", "No", "Name", "Batch", "Phone", "Email", "Gender"
}
) {
    Class[] types = new Class [] {
        java.lang.Integer.class, java.lang.Integer.class, java.lang.String.class,
        java.lang.String.class, java.lang.String.class, java.lang.String.class, java.lang.String.class,
        java.lang.String.class
    };
    boolean[] canEdit = new boolean [] {
        true, true, false, false, true, false, false, false
    };

    public Class getColumnClass(int columnIndex) {
        return types [columnIndex];
    }
}
```

```
public boolean isCellEditable(int rowIndex, int columnIndex) {  
    return canEdit [columnIndex];  
}  
});  
tblStudent.setShowHorizontalLines(false);  
tblStudent.setShowVerticalLines(false);  
tblStudent.addMouseListener(new java.awt.event.MouseAdapter() {  
    public void mouseClicked(java.awt.event.MouseEvent evt) {  
        tblStudentMouseClicked(evt);  
    }  
});  
jScrollPane1.setViewportView(tblStudent);  
  
listPanel.add(jScrollPane1, new org.netbeans.lib.awtextra.AbsoluteConstraints(160, 190, 733, 285));  
  
jLabel19.setBackground(new java.awt.Color(0, 102, 153));  
jLabel19.setFont(new java.awt.Font("Tahoma", 1, 12)); // NOI18N  
jLabel19.setForeground(new java.awt.Color(255, 255, 255));  
jLabel19.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);  
jLabel19.setText("Register no");  
jLabel19.setOpaque(true);  
listPanel.add(jLabel19, new org.netbeans.lib.awtextra.AbsoluteConstraints(160, 120, 90, 25));  
  
jLabel64.setBackground(new java.awt.Color(0, 102, 153));  
jLabel64.setFont(new java.awt.Font("Tahoma", 1, 12)); // NOI18N  
jLabel64.setForeground(new java.awt.Color(255, 255, 255));  
jLabel64.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);  
jLabel64.setText("Roll No");  
jLabel64.setOpaque(true);  
listPanel.add(jLabel64, new org.netbeans.lib.awtextra.AbsoluteConstraints(160, 150, 90, 25));  
  
jLabel65.setBackground(new java.awt.Color(0, 102, 153));  
jLabel65.setFont(new java.awt.Font("Tahoma", 1, 12)); // NOI18N  
jLabel65.setForeground(new java.awt.Color(255, 255, 255));  
jLabel65.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);  
jLabel65.setText("Name");  
jLabel65.setOpaque(true);  
listPanel.add(jLabel65, new org.netbeans.lib.awtextra.AbsoluteConstraints(500, 120, 90, 25));  
  
jLabel66.setBackground(new java.awt.Color(255, 102, 0));  
jLabel66.setFont(new java.awt.Font("Tahoma", 1, 12)); // NOI18N  
jLabel66.setForeground(new java.awt.Color(255, 255, 255));
```

```
jLabel66.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
jLabel66.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/Images/icons8_Google_Web_Search_17px.png
")));
// NOI18N
jLabel66.setText("Search");
jLabel66.setOpaque(true);
jLabel66.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        jLabel66MouseClicked(evt);
    }
});
listPanel.add(jLabel66, new org.netbeans.lib.awtextra.AbsoluteConstraints(800, 120, 90,
25));

txtRegisterNoSearch.setBackground(new java.awt.Color(85, 153, 187));
txtRegisterNoSearch.setFont(new java.awt.Font("Tahoma", 1, 12)); // NOI18N
txtRegisterNoSearch.setBorder(null);
listPanel.add(txtRegisterNoSearch, new org.netbeans.lib.awtextra.AbsoluteConstraints(250,
120, 200, 25));

txtNoSearch.setBackground(new java.awt.Color(85, 153, 187));
txtNoSearch.setFont(new java.awt.Font("Tahoma", 1, 12)); // NOI18N
txtNoSearch.setBorder(null);
listPanel.add(txtNoSearch, new org.netbeans.lib.awtextra.AbsoluteConstraints(250, 150,
200, 25));

txtNameSearch.setBackground(new java.awt.Color(85, 153, 187));
txtNameSearch.setFont(new java.awt.Font("Tahoma", 1, 12)); // NOI18N
txtNameSearch.setBorder(null);
listPanel.add(txtNameSearch, new org.netbeans.lib.awtextra.AbsoluteConstraints(590, 120,
170, 25));

comboBatchSearch.setBackground(new java.awt.Color(85, 153, 187));
comboBatchSearch.setEditable(true);
comboBatchSearch.setFont(new java.awt.Font("Tahoma", 1, 12)); // NOI18N
comboBatchSearch.setBorder(null);
listPanel.add(comboBatchSearch, new org.netbeans.lib.awtextra.AbsoluteConstraints(590,
150, 170, 25));

jLabel5.setBackground(new java.awt.Color(1, 100, 151));
jLabel5.setFont(new java.awt.Font("Times New Roman", 1, 24)); // NOI18N
jLabel5.setForeground(new java.awt.Color(255, 255, 255));
jLabel5.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
jLabel5.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/Images/icons8_Student_Male_38px.png"))); // NOI18N
```

```
jLabel5.setText("Student List");
jLabel5.setIconTextGap(12);
jLabel5.setOpaque(true);
listPanel.add(jLabel5, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 0, 1050, 70));

jLabel68.setBackground(new java.awt.Color(0, 102, 153));
jLabel68.setFont(new java.awt.Font("Tahoma", 1, 12)); // NOI18N
jLabel68.setForeground(new java.awt.Color(255, 255, 255));
jLabel68.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
jLabel68.setText("Batch");
jLabel68.setOpaque(true);
listPanel.add(jLabel68, new org.netbeans.lib.awtextra.AbsoluteConstraints(500, 150, 90,
25));

jLabel69.setBackground(new java.awt.Color(0, 102, 255));
jLabel69.setFont(new java.awt.Font("Tahoma", 1, 12)); // NOI18N
jLabel69.setForeground(new java.awt.Color(255, 255, 255));
jLabel69.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
jLabel69.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/Images/icons8_Refresh_17px.png"))); // NOI18N
jLabel69.setText("Reset");
jLabel69.setOpaque(true);
jLabel69.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        jLabel69MouseClicked(evt);
    }
});
listPanel.add(jLabel69, new org.netbeans.lib.awtextra.AbsoluteConstraints(800, 150, 90,
25));

jLabel70.setBackground(new java.awt.Color(52, 203, 0));
jLabel70.setFont(new java.awt.Font("Tahoma", 1, 12)); // NOI18N
jLabel70.setForeground(new java.awt.Color(255, 255, 255));
jLabel70.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
jLabel70.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/Images/icons8_Add_New_17px.png"))); // NOI18N
jLabel70.setText("Add");
jLabel70.setOpaque(true);
jLabel70.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        jLabel70MouseClicked(evt);
    }
});
}
```

```
listPanel.add(jLabel70, new org.netbeans.lib.awtextra.AbsoluteConstraints(320, 530, 90, 25));  
  
btnEdit1.setBackground(new java.awt.Color(204, 0, 204));  
btnEdit1.setFont(new java.awt.Font("Tahoma", 1, 12)); // NOI18N  
btnEdit1.setForeground(new java.awt.Color(255, 255, 255));  
btnEdit1.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);  
btnEdit1.setIcon(new javax.swing.ImageIcon(getClass().getResource("/Images/icons8_Edit_Row_17px.png"))); // NOI18N  
btnEdit1.setText("Edit");  
btnEdit1.setOpaque(true);  
btnEdit1.addMouseListener(new java.awt.event.MouseAdapter() {  
    public void mouseClicked(java.awt.event.MouseEvent evt) {  
        btnEdit1MouseClicked(evt);  
    }  
});  
listPanel.add(btnEdit1, new org.netbeans.lib.awtextra.AbsoluteConstraints(480, 530, 90, 25));  
  
btnExport.setBackground(new java.awt.Color(0, 0, 204));  
btnExport.setFont(new java.awt.Font("Tahoma", 1, 12)); // NOI18N  
btnExport.setForeground(new java.awt.Color(255, 255, 255));  
btnExport.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);  
btnExport.setIcon(new javax.swing.ImageIcon(getClass().getResource("/Images/icons8_Outbox_17px.png"))); // NOI18N  
btnExport.setText("Export");  
btnExport.setOpaque(true);  
btnExport.addMouseListener(new java.awt.event.MouseAdapter() {  
    public void mouseClicked(java.awt.event.MouseEvent evt) {  
        btnExportMouseClicked(evt);  
    }  
});  
listPanel.add(btnExport, new org.netbeans.lib.awtextra.AbsoluteConstraints(810, 530, 90, 25));  
  
btnDetail.setBackground(new java.awt.Color(255, 0, 102));  
btnDetail.setFont(new java.awt.Font("Tahoma", 1, 12)); // NOI18N  
btnDetail.setForeground(new java.awt.Color(255, 255, 255));  
btnDetail.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);  
btnDetail.setIcon(new javax.swing.ImageIcon(getClass().getResource("/Images/icons8_Info_18px_1.png"))); // NOI18N  
btnDetail.setText("Detail");  
btnDetail.setOpaque(true);  
btnDetail.addMouseListener(new java.awt.event.MouseAdapter() {
```

```
public void mouseClicked(java.awt.event.MouseEvent evt) {
    btnDetailMouseClicked(evt);
}
});
listPanel.add(btnDetail, new org.netbeans.lib.awtextra.AbsoluteConstraints(160, 530, 90,
25));

btnDeactive.setBackground(new java.awt.Color(255, 0, 0));
btnDeactive.setFont(new java.awt.Font("Tahoma", 1, 12)); // NOI18N
btnDeactive.setForeground(new java.awt.Color(255, 255, 255));
btnDeactive.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
btnDeactive.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/Images/icons8_Radio_Active_17px.png"))); // NOI18N
btnDeactive.setText("Deactivate");
btnDeactive.setOpaque(true);
btnDeactive.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        btnDeactiveMouseClicked(evt);
    }
});
listPanel.add(btnDeactive, new org.netbeans.lib.awtextra.AbsoluteConstraints(650, 530, 100,
25));

getContentPane().add(listPanel, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 0,
1020, 700));

detailPanel.setMinimumSize(new java.awt.Dimension(1020, 700));
detailPanel.setPreferredSize(new java.awt.Dimension(1020, 700));
detailPanel.setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

jScrollPane3.setBorder(null);

jPanel1.setBackground(new java.awt.Color(45, 129, 171));
jPanel1.setForeground(new java.awt.Color(4, 74, 121));
jPanel1.setMinimumSize(new java.awt.Dimension(980, 1145));
jPanel1.setPreferredSize(new java.awt.Dimension(980, 1145));
jPanel1.setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

jLabel14.setBackground(new java.awt.Color(0, 102, 153));
jLabel14.setFont(new java.awt.Font("Tahoma", 1, 12)); // NOI18N
jLabel14.setForeground(new java.awt.Color(255, 255, 255));
jLabel14.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
jLabel14.setText("Name");
jLabel14.setOpaque(true);
```

```
jPanel1.add(jLabel14, new org.netbeans.lib.awtextra.AbsoluteConstraints(160, 170, 120, 26));  
  
jLabel4.setBackground(new java.awt.Color(0, 102, 153));  
jLabel4.setFont(new java.awt.Font("Tahoma", 1, 12)); // NOI18N  
jLabel4.setForeground(new java.awt.Color(255, 255, 255));  
jLabel4.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);  
jLabel4.setText("Phone");  
jLabel4.setOpaque(true);  
jPanel1.add(jLabel4, new org.netbeans.lib.awtextra.AbsoluteConstraints(540, 230, 120, 26));  
  
jLabel16.setBackground(new java.awt.Color(0, 102, 153));  
jLabel16.setFont(new java.awt.Font("Tahoma", 1, 12)); // NOI18N  
jLabel16.setForeground(new java.awt.Color(255, 255, 255));  
jLabel16.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);  
jLabel16.setText("Gender");  
jLabel16.setOpaque(true);  
jPanel1.add(jLabel16, new org.netbeans.lib.awtextra.AbsoluteConstraints(540, 170, 120, 26));  
  
jLabel39.setBackground(new java.awt.Color(0, 102, 153));  
jLabel39.setFont(new java.awt.Font("Tahoma", 1, 12)); // NOI18N  
jLabel39.setForeground(new java.awt.Color(255, 255, 255));  
jLabel39.setHorizontalAlignment(javax.swing.SwingConstants.LEFT);  
jLabel39.setText("Extra Curricular");  
jLabel39.setOpaque(true);  
jPanel1.add(jLabel39, new org.netbeans.lib.awtextra.AbsoluteConstraints(160, 440, 680, 26));  
  
labelReasonLeaving.setBackground(new java.awt.Color(85, 153, 187));  
labelReasonLeaving.setFont(new java.awt.Font("Tahoma", 1, 12)); // NOI18N  
labelReasonLeaving.setForeground(new java.awt.Color(255, 255, 255));  
labelReasonLeaving.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);  
labelReasonLeaving.setText("Reason leaving");  
labelReasonLeaving.setOpaque(true);  
jPanel1.add(labelReasonLeaving, new org.netbeans.lib.awtextra.AbsoluteConstraints(280, 290, 660, 50));  
  
jLabel20.setBackground(new java.awt.Color(0, 102, 153));  
jLabel20.setFont(new java.awt.Font("Tahoma", 1, 12)); // NOI18N  
jLabel20.setForeground(new java.awt.Color(255, 255, 255));  
jLabel20.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);  
jLabel20.setText("Roll No");  
jLabel20.setOpaque(true);  
jPanel1.add(jLabel20, new org.netbeans.lib.awtextra.AbsoluteConstraints(540, 140, 120, 26));
```

```
labelNo.setBackground(new java.awt.Color(85, 153, 187));
labelNo.setFont(new java.awt.Font("Tahoma", 1, 12)); // NOI18N
labelNo.setForeground(new java.awt.Color(255, 255, 255));
labelNo.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
labelNo.setText("No");
labelNo.setOpaque(true);
jPanel1.add(labelNo, new org.netbeans.lib.awtextra.AbsoluteConstraints(660, 140, 280,
26));

labelBirthday.setBackground(new java.awt.Color(85, 153, 187));
labelBirthday.setFont(new java.awt.Font("Tahoma", 1, 12)); // NOI18N
labelBirthday.setForeground(new java.awt.Color(255, 255, 255));
labelBirthday.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
labelBirthday.setText("Birthday");
labelBirthday.setOpaque(true);
jPanel1.add(labelBirthday, new org.netbeans.lib.awtextra.AbsoluteConstraints(280, 200,
240, 26));

labelPhone.setBackground(new java.awt.Color(85, 153, 187));
labelPhone.setFont(new java.awt.Font("Tahoma", 1, 12)); // NOI18N
labelPhone.setForeground(new java.awt.Color(255, 255, 255));
labelPhone.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
labelPhone.setText("Phone");
labelPhone.setOpaque(true);
jPanel1.add(labelPhone, new org.netbeans.lib.awtextra.AbsoluteConstraints(660, 230, 280,
26));

jLabel11.setBackground(new java.awt.Color(0, 102, 153));
jLabel11.setFont(new java.awt.Font("Tahoma", 1, 12)); // NOI18N
jLabel11.setForeground(new java.awt.Color(255, 255, 255));
jLabel11.setText("Family Infomation");
jLabel11.setOpaque(true);
jPanel1.add(jLabel11, new org.netbeans.lib.awtextra.AbsoluteConstraints(160, 770, 690,
26));

jLabel25.setBackground(new java.awt.Color(0, 102, 153));
jLabel25.setFont(new java.awt.Font("Tahoma", 1, 12)); // NOI18N
jLabel25.setForeground(new java.awt.Color(255, 255, 255));
jLabel25.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
jLabel25.setText("Reason leaving");
jLabel25.setOpaque(true);
jPanel1.add(jLabel25, new org.netbeans.lib.awtextra.AbsoluteConstraints(160, 290, 120,
50));

labelRegisterNo.setBackground(new java.awt.Color(85, 153, 187));
```

```
labelRegisterNo.setFont(new java.awt.Font("Tahoma", 1, 12)); // NOI18N
labelRegisterNo.setForeground(new java.awt.Color(255, 255, 255));
labelRegisterNo.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
labelRegisterNo.setText("Register No");
labelRegisterNo.setOpaque(true);
jPanel1.add(labelRegisterNo, new org.netbeans.lib.awtextra.AbsoluteConstraints(280, 140,
240, 26));

jScrollPane10.setBorder(javax.swing.BorderFactory.createLineBorder(new
java.awt.Color(4, 74, 121)));

tblMarkDetail3.setBackground(new java.awt.Color(4, 74, 121));
tblMarkDetail3.setForeground(new java.awt.Color(255, 255, 255));
tblMarkDetail3.setModel(new javax.swing.table.DefaultTableModel(
new Object [][] {
    },
    new String [] {
        "ID", "Subject", "Mark", "Time"
    }
));
tblMarkDetail3.setPreferredSize(new java.awt.Dimension(245, 120));
tblMarkDetail3.setSelectionBackground(new java.awt.Color(0, 204, 51));
tblMarkDetail3.setShowHorizontalLines(false);
tblMarkDetail3.setShowVerticalLines(false);
jScrollPane10.setViewportView(tblMarkDetail3);

jPanel1.add(jScrollPane10, new org.netbeans.lib.awtextra.AbsoluteConstraints(160, 960,
780, 140));

jLabel8.setBackground(new java.awt.Color(0, 102, 153));
jLabel8.setFont(new java.awt.Font("Tahoma", 1, 12)); // NOI18N
jLabel8.setForeground(new java.awt.Color(255, 255, 255));
jLabel8.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
jLabel8.setText("Date of Birth");
jLabel8.setOpaque(true);
jPanel1.add(jLabel8, new org.netbeans.lib.awtextra.AbsoluteConstraints(160, 200, 120, 26));

jLabel11.setBackground(new java.awt.Color(0, 102, 153));
jLabel11.setFont(new java.awt.Font("Times New Roman", 1, 24)); // NOI18N
jLabel11.setForeground(new java.awt.Color(255, 255, 255));
jLabel11.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
jLabel11.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/Images/icons8_Student_Male_38px.png"))); // NOI18N
jLabel11.setText("Student Details");
```

```

jLabel1.setIconTextGap(20);
jLabel1.setOpaque(true);
jPanel1.add(jLabel1, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 0, 1090, 60));

jLabel6.setBackground(new java.awt.Color(0, 102, 153));
jLabel6.setFont(new java.awt.Font("Tahoma", 1, 12)); // NOI18N
jLabel6.setForeground(new java.awt.Color(255, 255, 255));
jLabel6.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
jLabel6.setText("Email");
jLabel6.setOpaque(true);
jPanel1.add(jLabel6, new org.netbeans.lib.awtextra.AbsoluteConstraints(160, 230, 120, 26));

jScrollPane18.setBorder(javax.swing.BorderFactory.createLineBorder(new
java.awt.Color(4, 74, 121)));

tblRemarkDetail.setBackground(new java.awt.Color(4, 74, 121));
tblRemarkDetail.setForeground(new java.awt.Color(255, 255, 255));
tblRemarkDetail.setModel(new javax.swing.table.DefaultTableModel(
new Object [][] {
},
new String [] {
    "ID", "Staff No", "Staff name", "Content", "Time"
}
));
tblRemarkDetail.setPreferredSize(new java.awt.Dimension(245, 120));
tblRemarkDetail.setSelectionBackground(new java.awt.Color(0, 204, 51));
tblRemarkDetail.setShowHorizontalLines(false);
tblRemarkDetail.setShowVerticalLines(false);
tblRemarkDetail.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        tblRemarkDetailMouseClicked(evt);
    }
});
jScrollPane18.setViewportView(tblRemarkDetail);

jPanel1.add(jScrollPane18, new org.netbeans.lib.awtextra.AbsoluteConstraints(160, 630,
780, 120));

jLabel2.setBackground(new java.awt.Color(0, 102, 153));
jLabel2.setFont(new java.awt.Font("Tahoma", 1, 12)); // NOI18N
jLabel2.setForeground(new java.awt.Color(255, 255, 255));
jLabel2.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
jLabel2.setText("Register No");
jLabel2.setOpaque(true);
jPanel1.add(jLabel2, new org.netbeans.lib.awtextra.AbsoluteConstraints(160, 140, 120, 26));

```

```
labelName.setBackground(new java.awt.Color(85, 153, 187));
labelName.setFont(new java.awt.Font("Tahoma", 1, 12)); // NOI18N
labelName.setForeground(new java.awt.Color(255, 255, 255));
labelName.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
labelName.setText("Name");
labelName.setOpaque(true);
jPanel1.add(labelName, new org.netbeans.lib.awtextra.AbsoluteConstraints(280, 170, 240,
26));

labelAdmissionTime.setBackground(new java.awt.Color(85, 153, 187));
labelAdmissionTime.setFont(new java.awt.Font("Tahoma", 1, 12)); // NOI18N
labelAdmissionTime.setForeground(new java.awt.Color(255, 255, 255));
labelAdmissionTime.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
labelAdmissionTime.setText("Time");
labelAdmissionTime.setOpaque(true);
jPanel1.add(labelAdmissionTime, new org.netbeans.lib.awtextra.AbsoluteConstraints(660,
260, 280, 26));

labelBatch.setBackground(new java.awt.Color(85, 153, 187));
labelBatch.setFont(new java.awt.Font("Tahoma", 1, 12)); // NOI18N
labelBatch.setForeground(new java.awt.Color(255, 255, 255));
labelBatch.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
labelBatch.setText("Batch");
labelBatch.setOpaque(true);
jPanel1.add(labelBatch, new org.netbeans.lib.awtextra.AbsoluteConstraints(660, 200, 280,
26));

jLabel23.setBackground(new java.awt.Color(0, 102, 153));
jLabel23.setFont(new java.awt.Font("Tahoma", 1, 12)); // NOI18N
jLabel23.setForeground(new java.awt.Color(255, 255, 255));
jLabel23.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
jLabel23.setText("Batch");
jLabel23.setOpaque(true);
jPanel1.add(jLabel23, new org.netbeans.lib.awtextra.AbsoluteConstraints(540, 200, 120,
26));

jScrollPane2.setBorder(null);
jScrollPane2.setForeground(new java.awt.Color(255, 255, 255));

labelAddress.setEditable(false);
labelAddress.setBackground(new java.awt.Color(85, 153, 187));
labelAddress.setBorder(null);
labelAddress.setFont(new java.awt.Font("Tahoma", 1, 12)); // NOI18N
jScrollPane2.setViewportView(labelAddress);
```

```
jPanel1.add(jScrollPane2, new org.netbeans.lib.awtextra.AbsoluteConstraints(280, 350, 660, 80));  
  
labelPrevInstitute.setBackground(new java.awt.Color(85, 153, 187));  
labelPrevInstitute.setFont(new java.awt.Font("Tahoma", 1, 12)); // NOI18N  
labelPrevInstitute.setForeground(new java.awt.Color(255, 255, 255));  
labelPrevInstitute.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);  
labelPrevInstitute.setText("Previous institute");  
labelPrevInstitute.setOpaque(true);  
jPanel1.add(labelPrevInstitute, new org.netbeans.lib.awtextra.AbsoluteConstraints(280, 260, 240, 26));  
  
jScrollPane17.setBorder(javax.swing.BorderFactory.createLineBorder(new java.awt.Color(4, 74, 121)));  
  
tblRelationDetail.setBackground(new java.awt.Color(4, 74, 121));  
tblRelationDetail.setForeground(new java.awt.Color(255, 255, 255));  
tblRelationDetail.setModel(new javax.swing.table.DefaultTableModel(  
    new Object [][] {  
  
        },  
        new String [] {  
            "ID", "Relation", "Name", "Phone", "Address", "Occupation", "Gender"  
        }  
    ));  
tblRelationDetail.setPreferredSize(new java.awt.Dimension(245, 120));  
tblRelationDetail.setSelectionBackground(new java.awt.Color(0, 204, 51));  
tblRelationDetail.setShowHorizontalLines(false);  
tblRelationDetail.setShowVerticalLines(false);  
tblRelationDetail.addMouseListener(new java.awt.event.MouseAdapter() {  
    public void mouseClicked(java.awt.event.MouseEvent evt) {  
        tblRelationDetailMouseClicked(evt);  
    }  
});  
jScrollPane17.setViewportView(tblRelationDetail);  
  
jPanel1.add(jScrollPane17, new org.netbeans.lib.awtextra.AbsoluteConstraints(140, 800, 780, 110));  
  
jScrollPane9.setBackground(new java.awt.Color(0, 102, 153));  
jScrollPane9.setBorder(javax.swing.BorderFactory.createLineBorder(new java.awt.Color(4, 74, 121)));  
  
tblExtraCurricularDetail.setBackground(new java.awt.Color(4, 74, 121));  
tblExtraCurricularDetail.setForeground(new java.awt.Color(255, 255, 255));  
tblExtraCurricularDetail.setModel(new javax.swing.table.DefaultTableModel(
```

```
new Object [][] {  
    },  
    new String [] {  
        "ID", "Name", "Location", "Start", "End", "Notes"  
    }  
});  
tblExtraCurricularDetail.setPreferredSize(new java.awt.Dimension(245, 120));  
tblExtraCurricularDetail.setSelectionBackground(new java.awt.Color(0, 204, 51));  
tblExtraCurricularDetail.setShowHorizontalLines(false);  
tblExtraCurricularDetail.setShowVerticalLines(false);  
jScrollPane9.setViewportView(tblExtraCurricularDetail);  
  
jPanel1.add(jScrollPane9, new org.netbeans.lib.awtextra.AbsoluteConstraints(142, 118, 800, 120));  
  
jLabel10.setBackground(new java.awt.Color(0, 102, 153));  
jLabel10.setFont(new java.awt.Font("Tahoma", 1, 12)); // NOI18N  
jLabel10.setForeground(new java.awt.Color(255, 255, 255));  
jLabel10.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);  
jLabel10.setText("Address");  
jLabel10.setOpaque(true);  
jPanel1.add(jLabel10, new org.netbeans.lib.awtextra.AbsoluteConstraints(142, 142, 800, 120));  
  
labelGender.setBackground(new java.awt.Color(85, 153, 187));  
labelGender.setFont(new java.awt.Font("Tahoma", 1, 12)); // NOI18N  
labelGender.setForeground(new java.awt.Color(255, 255, 255));  
labelGender.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);  
labelGender.setText("Gender");  
labelGender.setOpaque(true);  
jPanel1.add(labelGender, new org.netbeans.lib.awtextra.AbsoluteConstraints(142, 142, 800, 120));  
  
jLabel30.setBackground(new java.awt.Color(0, 102, 153));  
jLabel30.setFont(new java.awt.Font("Tahoma", 1, 12)); // NOI18N  
jLabel30.setForeground(new java.awt.Color(255, 255, 255));  
jLabel30.setText("Remark");  
jLabel30.setOpaque(true);  
jPanel1.add(jLabel30, new org.netbeans.lib.awtextra.AbsoluteConstraints(142, 142, 800, 120));  
  
btnDeactive1.setBackground(new java.awt.Color(255, 0, 0));  
btnDeactive1.setFont(new java.awt.Font("Tahoma", 1, 12)); // NOI18N  
btnDeactive1.setForeground(new java.awt.Color(255, 255, 255));  
btnDeactive1.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
```

```
btnDeactive1.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/Images/icons8_Radio_Active_17px.png"))); // NOI18N
btnDeactive1.setText("Suspend");
btnDeactive1.setOpaque(true);
btnDeactive1.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        btnDeactive1MouseClicked(evt);
    }
});
jPanel1.add(btnDeactive1, new org.netbeans.lib.awtextra.AbsoluteConstraints(847, 1120, 90, 25));

jLabel28.setBackground(new java.awt.Color(0, 102, 153));
jLabel28.setFont(new java.awt.Font("Tahoma", 1, 12)); // NOI18N
jLabel28.setForeground(new java.awt.Color(255, 255, 255));
jLabel28.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
jLabel28.setText("Previous School");
jLabel28.setOpaque(true);
jPanel1.add(jLabel28, new org.netbeans.lib.awtextra.AbsoluteConstraints(160, 260, 120, 26));

labelEmail1.setBackground(new java.awt.Color(85, 153, 187));
labelEmail1.setFont(new java.awt.Font("Tahoma", 1, 12)); // NOI18N
labelEmail1.setForeground(new java.awt.Color(255, 255, 255));
labelEmail1.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
labelEmail1.setText("Email");
labelEmail1.setOpaque(true);
jPanel1.add(labelEmail1, new org.netbeans.lib.awtextra.AbsoluteConstraints(280, 230, 240, 26));

btnExtraDetail.setBackground(new java.awt.Color(52, 203, 0));
btnExtraDetail.setFont(new java.awt.Font("Tahoma", 1, 12)); // NOI18N
btnExtraDetail.setForeground(new java.awt.Color(255, 255, 255));
btnExtraDetail.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
btnExtraDetail.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/Images/icons8_Info_18px_1.png"))); // NOI18N
btnExtraDetail.setText("Detail");
btnExtraDetail.setOpaque(true);
btnExtraDetail.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        btnExtraDetailMouseClicked(evt);
    }
});
jPanel1.add(btnExtraDetail, new org.netbeans.lib.awtextra.AbsoluteConstraints(840, 440, 100, 26));
```

```
jLabel29.setBackground(new java.awt.Color(0, 102, 153));
jLabel29.setFont(new java.awt.Font("Tahoma", 1, 12)); // NOI18N
jLabel29.setForeground(new java.awt.Color(255, 255, 255));
jLabel29.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
jLabel29.setText("Time");
jLabel29.setOpaque(true);
jPanel1.add(jLabel29, new org.netbeans.lib.awtextra.AbsoluteConstraints(540, 260, 120,
26));

btnRemarkDetail.setBackground(new java.awt.Color(52, 203, 0));
btnRemarkDetail.setFont(new java.awt.Font("Tahoma", 1, 12)); // NOI18N
btnRemarkDetail.setForeground(new java.awt.Color(255, 255, 255));
btnRemarkDetail.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
btnRemarkDetail.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/Images/icons8_Info_18px_1.png"))); // NOI18N
btnRemarkDetail.setText("Detail");
btnRemarkDetail.setOpaque(true);
btnRemarkDetail.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        btnRemarkDetailMouseClicked(evt);
    }
});
jPanel1.add(btnRemarkDetail, new org.netbeans.lib.awtextra.AbsoluteConstraints(840, 600,
100, 26));

btnRelationDetail.setBackground(new java.awt.Color(52, 203, 0));
btnRelationDetail.setFont(new java.awt.Font("Tahoma", 1, 12)); // NOI18N
btnRelationDetail.setForeground(new java.awt.Color(255, 255, 255));
btnRelationDetail.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
btnRelationDetail.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/Images/icons8_Info_18px_1.png"))); // NOI18N
btnRelationDetail.setText("Detail");
btnRelationDetail.setOpaque(true);
btnRelationDetail.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        btnRelationDetailMouseClicked(evt);
    }
});
jPanel1.add(btnRelationDetail, new org.netbeans.lib.awtextra.AbsoluteConstraints(850, 770,
90, 26));

jLabel48.setBackground(new java.awt.Color(0, 102, 153));
jLabel48.setFont(new java.awt.Font("Tahoma", 1, 12)); // NOI18N
jLabel48.setForeground(new java.awt.Color(255, 255, 255));
jLabel48.setText("Mark");
```

```
jLabel48.setOpaque(true);
jPanel1.add(jLabel48, new org.netbeans.lib.awtextra.AbsoluteConstraints(160, 920, 780,
30));

btnViewList.setBackground(new java.awt.Color(52, 203, 0));
btnViewList.setFont(new java.awt.Font("Tahoma", 1, 12)); // NOI18N
btnViewList.setForeground(new java.awt.Color(255, 255, 255));
btnViewList.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
btnViewList.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/Images/icons8_Arrow_Pointing_Left_18px_1.p
ng"))); // NOI18N
btnViewList.setText("Back");
btnViewList.setOpaque(true);
btnViewList.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        btnViewListMouseClicked(evt);
    }
});
jPanel1.add(btnViewList, new org.netbeans.lib.awtextra.AbsoluteConstraints(160, 1120, 77,
25));

btnAdd.setBackground(new java.awt.Color(43, 175, 0));
btnAdd.setFont(new java.awt.Font("Tahoma", 1, 12)); // NOI18N
btnAdd.setForeground(new java.awt.Color(255, 255, 255));
btnAdd.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
btnAdd.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/Images/icons8_Add_New_17px.png"))); // NOI18N
btnAdd.setText("Add");
btnAdd.setOpaque(true);
btnAdd.addMouseMotionListener(new java.awt.event.MouseMotionAdapter() {
    public void mouseDragged(java.awt.event.MouseEvent evt) {
        btnAddMouseDragged(evt);
    }
});
btnAdd.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        btnAddMouseClicked(evt);
    }
});
jPanel1.add(btnAdd, new org.netbeans.lib.awtextra.AbsoluteConstraints(360, 1120, 77, 25));

btnEdit.setBackground(new java.awt.Color(204, 0, 204));
btnEdit.setFont(new java.awt.Font("Tahoma", 1, 12)); // NOI18N
btnEdit.setForeground(new java.awt.Color(255, 255, 255));
btnEdit.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
```

```
        btnEdit.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/Images/icons8_Edit_Row_17px.png"))); // NOI18N
        btnEdit.setText("Edit");
        btnEdit.setOpaque(true);
        btnEdit.addMouseListener(new java.awt.event.MouseAdapter() {
            public void mouseClicked(java.awt.event.MouseEvent evt) {
                btnEditMouseClicked(evt);
            }
        });
        jPanel1.add(btnEdit, new org.netbeans.lib.awtextra.AbsoluteConstraints(620, 1120, 77, 25));

        jLabel50.setBackground(new java.awt.Color(0, 102, 0));
        jLabel50.setFont(new java.awt.Font("Tahoma", 1, 12)); // NOI18N
        jLabel50.setForeground(new java.awt.Color(255, 255, 255));
        jLabel50.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
        jLabel50.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/Images/icons8_Undo_30px.png"))); // NOI18N
        jLabel50.setText("Back");
        jLabel50.setIconTextGap(9);
        jLabel50.addMouseListener(new java.awt.event.MouseAdapter() {
            public void mouseClicked(java.awt.event.MouseEvent evt) {
                jLabel50MouseClicked(evt);
            }
        });
        jPanel1.add(jLabel50, new org.netbeans.lib.awtextra.AbsoluteConstraints(20, 70, 90, 30));

        jScrollPane3.setViewportView(jPanel1);

        detailPanel.add(jScrollPane3, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 0, 1020, 700));

        getContentPane().add(detailPanel, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 0, 1020, 700));

    pack();
}// </editor-fold>

private void tblRelationDetailMouseClicked(java.awt.event.MouseEvent evt) {
    btnRelationDetail.setEnabled(true);
}

private void tblRemarkDetailMouseClicked(java.awt.event.MouseEvent evt) {
    btnExtraDetail.setEnabled(true);
}
```

```

private void tblExtraCurricularMouseClicked(java.awt.event.MouseEvent evt) {
    btnDeleteExtra.setEnabled(true);
    btnEditExtra.setEnabled(true);
}

private void tblRemarkMouseClicked(java.awt.event.MouseEvent evt) {
    btnDeleteRemark.setEnabled(true);
    btnEditRemark.setEnabled(true);
}

private void tblRelationMouseClicked(java.awt.event.MouseEvent evt) {
    btnDeleteRelation.setEnabled(true);
    btnEditRelation.setEnabled(true);
}

private void tblMarkMouseClicked(java.awt.event.MouseEvent evt) {
    btnAddMark.setEnabled(true);
    btnDeleteMark.setEnabled(true);
    btnEditMark.setEnabled(true);
}

private void jLabel69MouseClicked(java.awt.event.MouseEvent evt) {
    txtRegisterNoSearch.setText("");
    txtNoSearch.setText("");
    txtNameSearch.setText("");
    comboBatchSearch.setSelectedIndex(0); // TODO add your handling code here:
}

private void jLabel66MouseClicked(java.awt.event.MouseEvent evt) {
    search(); // TODO add your handling code here:
}

private void tblStudentMouseClicked(java.awt.event.MouseEvent evt) {
    currenStudent=studentManager.getStudentDetail(studentManager.getSelectID(tblStudent));
    enableButton(true);
    loadDetail();
    loadEdit();
}

private void btnDetailMouseClicked(java.awt.event.MouseEvent evt) {
act=DETAIL;
    currenStudent=studentManager.getStudentDetail(studentManager.getSelectID(tblStudent));
    showPanel("Detail Student", detailPanel,listPanel,can);
    loadDetail(); // TODO add your handling code here:
}

```

```
private void jLabel70MouseClicked(java.awt.event.MouseEvent evt) {  
act=ADD;  
    newStudent=new Student();  
    showPanel("Add Student",updatePanel,listPanel,detailPanel);  
    loadAdd();      // TODO add your handling code here:  
}  
  
private void btnEdit1MouseClicked(java.awt.event.MouseEvent evt) {  
act=EDIT;  
    currenStudent=studentManager.getStudentDetail(studentManager.getSelectID(tblStudent));  
    newStudent=currenStudent;  
    showPanel("Edit Student",updatePanel,listPanel,detailPanel);  
    loadEdit();      // TODO add your handling code here:  
}  
  
private void btnDeactiveMouseClicked(java.awt.event.MouseEvent evt) {  
deactive();      // TODO add your handling code here:  
}  
  
private void btnExportMouseClicked(java.awt.event.MouseEvent evt) {  
export();      // TODO add your handling code here:  
}  
  
private void btnAddRelationMouseClicked(java.awt.event.MouseEvent evt) {  
    loadRelationInfo(ADD); // TODO add your handling code here:  
}  
  
private void btnEditRelationMouseClicked(java.awt.event.MouseEvent evt) {  
    loadRelationInfo(EDIT); // TODO add your handling code here:  
}  
  
private void btnDeleteRelationMouseClicked(java.awt.event.MouseEvent evt) {  
    removeSelectedRelation(); // TODO add your handling code here:  
}  
  
private void btnEditMarkMouseClicked(java.awt.event.MouseEvent evt) {  
    // TODO add your handling code here:  
}  
  
private void btnDeleteMarkMouseClicked(java.awt.event.MouseEvent evt) {  
removeSelectedMark();      // TODO add your handling code here:  
}  
  
private void btnAddMarkMouseClicked(java.awt.event.MouseEvent evt) {  
loadMarkinfo(ADD);      // TODO add your handling code here:
```

```
}

private void btnDeleteRemarkMouseClicked(java.awt.event.MouseEvent evt) {
removeSelectedRemark();      // TODO add your handling code here:
}

private void btnEditRemarkMouseClicked(java.awt.event.MouseEvent evt) {
loadRemarkInfo(EDIT);      // TODO add your handling code here:
}

private void btnAddRemarkMouseClicked(java.awt.event.MouseEvent evt) {
loadRemarkInfo(ADD);      // TODO add your handling code here:
}

private void btnDeleteExtraMouseClicked(java.awt.event.MouseEvent evt) {
removeSelectedExtra();      // TODO add your handling code here:
}

private void btnEditExtraMouseClicked(java.awt.event.MouseEvent evt) {
loadExtraInfo(EDIT);      // TODO add your handling code here:
}

private void btnAddExtraMouseClicked(java.awt.event.MouseEvent evt) {
loadExtraInfo(ADD);      // TODO add your handling code here:
}

private void btnCancelMouseClicked(java.awt.event.MouseEvent evt) {
    newStudent=null;
    showPanel("List Student", listPanel,updatePanel,detailPanel); // TODO add your handling
code here:
}

private void btnResetMouseClicked(java.awt.event.MouseEvent evt) {
    reset(); // TODO add your handling code here:
}

private void btnSaveMouseClicked(java.awt.event.MouseEvent evt) {
    save(); // TODO add your handling code here:
}

private void btnExtraDetailMouseClicked(java.awt.event.MouseEvent evt) {
    loadExtraInfo(DETAIL); // TODO add your handling code here:
}

private void btnRemarkDetailMouseClicked(java.awt.event.MouseEvent evt) {
loadRemarkInfo(DETAIL); // TODO add your handling code here:
```

```
}

private void btnRelationDetailMouseClicked(java.awt.event.MouseEvent evt) {
loadRelationInfo(DETAIL); // TODO add your handling code here:
}

private void btnDeactive1MouseClicked(java.awt.event.MouseEvent evt) {
deactive(); // TODO add your handling code here:
}

private void btnEditMouseClicked(java.awt.event.MouseEvent evt) {
act=EDIT;
currenStudent=studentManager.getStudentDetail(studentManager.getSelectID(tblStudent));
newStudent=currenStudent;
showPanel("Edit Student",updatePanel,listPanel,detailPanel);
loadEdit(); // TODO add your handling code here:
}

private void btnAddMouseDragged(java.awt.event.MouseEvent evt) {
// TODO add your handling code here:
}

private void btnAddMouseClicked(java.awt.event.MouseEvent evt) {
act=ADD;
newStudent=new Student();
showPanel("Add Student",updatePanel,listPanel,detailPanel);
loadAdd(); // TODO add your handling code here:
}

private void btnViewListMouseClicked(java.awt.event.MouseEvent evt) {
showPanel("List Student", listPanel,updatePanel,detailPanel); // TODO add your handling
code here:
}

private void jLabel49MouseClicked(java.awt.event.MouseEvent evt) {
newStudent=null;
showPanel("List Student", listPanel,updatePanel,detailPanel); // TODO add your
handling code here:
}

private void jLabel50MouseClicked(java.awt.event.MouseEvent evt) {
newStudent=null;
showPanel("List Student", listPanel,updatePanel,detailPanel); // TODO add your handling
code here:
}
```

```
// Variables declaration - do not modify
private com.toedter.calendar.JDateChooser admiDPic;
private javax.swing.JLabel btnAdd;
private javax.swing.JLabel btnAddExtra;
private javax.swing.JLabel btnAddMark;
private javax.swing.JLabel btnAddRelation;
private javax.swing.JLabel btnAddRemark;
private javax.swing.JLabel btnCancel;
private javax.swing.JLabel btnDeactive;
private javax.swing.JLabel btnDeactive1;
private javax.swing.JLabel btnDeleteExtra;
private javax.swing.JLabel btnDeleteMark;
private javax.swing.JLabel btnDeleteRelation;
private javax.swing.JLabel btnDeleteRemark;
private javax.swing.JLabel btnDetail;
private javax.swing.JLabel btnEdit;
private javax.swing.JLabel btnEdit1;
private javax.swing.JLabel btnEditExtra;
private javax.swing.JLabel btnEditMark;
private javax.swing.JLabel btnEditRelation;
private javax.swing.JLabel btnEditRemark;
private javax.swing.JLabel btnExport;
private javax.swing.JLabel btnExtraDetail;
private javax.swing.JLabel btnRelationDetail;
private javax.swing.JLabel btnRemarkDetail;
private javax.swing.JLabel btnReset;
private javax.swing.JLabel btnSave;
private javax.swing.JLabel btnViewList;
private javax.swing.JPanel can;
private javax.swing.JComboBox comboBatch;
private javax.swing.JComboBox comboBatch1;
private javax.swing.JComboBox comboBatchSearch;
private javax.swing.JComboBox comboGender;
private com.toedter.calendar.JDateChooser dPic;
private javax.swing.JPanel detailPanel;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel10;
private javax.swing.JLabel jLabel11;
private javax.swing.JLabel jLabel14;
private javax.swing.JLabel jLabel16;
private javax.swing.JLabel jLabel19;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel20;
private javax.swing.JLabel jLabel23;
private javax.swing.JLabel jLabel25;
private javax.swing.JLabel jLabel28;
```

```
private javax.swing.JLabel jLabel29;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel30;
private javax.swing.JLabel jLabel39;
private javax.swing.JLabel jLabel4;
private javax.swing.JLabel jLabel40;
private javax.swing.JLabel jLabel42;
private javax.swing.JLabel jLabel43;
private javax.swing.JLabel jLabel44;
private javax.swing.JLabel jLabel45;
private javax.swing.JLabel jLabel46;
private javax.swing.JLabel jLabel47;
private javax.swing.JLabel jLabel48;
private javax.swing.JLabel jLabel49;
private javax.swing.JLabel jLabel5;
private javax.swing.JLabel jLabel50;
private javax.swing.JLabel jLabel51;
private javax.swing.JLabel jLabel52;
private javax.swing.JLabel jLabel53;
private javax.swing.JLabel jLabel54;
private javax.swing.JLabel jLabel55;
private javax.swing.JLabel jLabel58;
private javax.swing.JLabel jLabel6;
private javax.swing.JLabel jLabel60;
private javax.swing.JLabel jLabel61;
private javax.swing.JLabel jLabel62;
private javax.swing.JLabel jLabel63;
private javax.swing.JLabel jLabel64;
private javax.swing.JLabel jLabel65;
private javax.swing.JLabel jLabel66;
private javax.swing.JLabel jLabel67;
private javax.swing.JLabel jLabel68;
private javax.swing.JLabel jLabel69;
private javax.swing.JLabel jLabel70;
private javax.swing.JLabel jLabel8;
private javax.swing.JPanel jPanel1;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JScrollPane jScrollPane10;
private javax.swing.JScrollPane jScrollPane17;
private javax.swing.JScrollPane jScrollPane18;
private javax.swing.JScrollPane jScrollPane2;
private javax.swing.JScrollPane jScrollPane21;
private javax.swing.JScrollPane jScrollPane22;
private javax.swing.JScrollPane jScrollPane23;
private javax.swing.JScrollPane jScrollPane3;
private javax.swing.JScrollPane jScrollPane4;
```

```
private javax.swing.JScrollPane jScrollPane6;
private javax.swing.JScrollPane jScrollPane7;
private javax.swing.JScrollPane jScrollPane9;
private javax.swing.JTextPane labelAddress;
private javax.swing.JLabel labelAdmissionTime;
private javax.swing.JLabel labelBatch;
private javax.swing.JLabel labelBirthday;
private javax.swing.JLabel labelEmail1;
private javax.swing.JLabel labelGender;
private javax.swing.JLabel labelName;
private javax.swing.JLabel labelNo;
private javax.swing.JLabel labelPhone;
private javax.swing.JLabel labelPrevInstitute;
private javax.swing.JLabel labelReasonLeaving;
private javax.swing.JLabel labelRegisterNo;
private javax.swing.JLabel lebUp;
private javax.swing.JPanel listPanel;
private javax.swing.JTable tblExtraCurricular;
private javax.swing.JTable tblExtraCurricularDetail;
private javax.swing.JTable tblMark;
private javax.swing.JTable tblMarkDetail3;
private javax.swing.JTable tblRelation;
private javax.swing.JTable tblRelationDetail;
private javax.swing.JTable tblRemark;
private javax.swing.JTable tblRemarkDetail;
private javax.swing.JTable tblStudent;
private javax.swing.JTextArea txtAddress;
private javax.swing.JTextField txtEmail;
private javax.swing.JTextField txtName;
private javax.swing.JTextField txtNameSearch;
private javax.swing.JTextField txtNo;
private javax.swing.JTextField txtNoSearch;
private javax.swing.JTextField txtPhone;
private javax.swing.JTextField txtPrevInstituteEdit;
private javax.swing.JTextField txtReasonLeavingEdit;
private javax.swing.JTextField txtRegisterNo;
private javax.swing.JTextField txtRegisterNoSearch;
private javax.swing.JPanel updatePanel;
// End of variables declaration
```

```
@Override
public void enableButton(boolean b) {
    btnDetail.setEnabled(b);
    btnEdit.setEnabled(b);
    btnDeactive1.setEnabled(b);
}
```

```

@Override
public boolean validateForm() {
    throw new UnsupportedOperationException("Not supported yet."); //To change body of
generated methods, choose Tools | Templates.
}

private void export() {
    ExportFile export=new ExportFile(tblStudent, ExportFile.STUDENT);
    export.toExcel();
}

}

```

- **For Subject Management Module of AeonSMS**

```

/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */
package StudentMan;

import FileExport.ExportFile;
import Controls.Manager;
import Controls.StudentManager;
import java.util.Vector;
import javax.swing.JOptionPane;
import javax.swing.plaf.basic.BasicInternalFrameUI;
import MainHome.UIInterface;

/**
 *
 * @author Aeon(Ankit RudrAshokJha)
 */
public class ListSubject extends javax.swing.JInternalFrame implements UIInterface{
    StudentManager studentManager=new StudentManager();
    private int act;
    /**
     * Creates new form ListPosition
     */
    public ListSubject() {
        initComponents();
        initUI();
    }
}

```

```

public ListSubject(int w, int h) {
    initComponents();
    initUI();
    setSize(w, h);
}
@Override
public void initUI(){
    enableButton(false);
    ((BasicInternalFrameUI)this.getUI()).setNorthPane(null);
    editAddPanel.setVisible(false);
    String SQL="SELECT ID, name FROM subject WHERE status=1 ";
    studentManager.loadTable(SQL, tblSubject, Manager.SUBJECT_MODEL);
}
@Override
public void enableEditAddPanel(boolean b) {
    if(b==false){
        tblSubject.setEnabled(true);
        txtName.setText(null);

        editAddPanel.setVisible(false);

        bannerPanel.setVisible(true);
    }else{
        if(act==EDIT){
            labelAction.setText("Edit Subject");
            loadEdit();
        }else if(act==ADD){
            labelAction.setText("Add Subject");
        }
        tblSubject.setEnabled(false);
        editAddPanel.setVisible(true);
        btnReset.setVisible(true);

        bannerPanel.setVisible(false);
    }
}

public void save(){
    if(!validateForm()){
        return;
    }
    if(act==ADD){
        add();
        return;
    }
}

```

```

if(act==EDIT){

    edit();
}
}

@Override
public void loadEdit() {
    Vector row=studentManager.getSelectRow(tblSubject);
    txtName.setText(row.get(1).toString());
    txtName.setRequestFocusEnabled(true);
}

public void add() {
    String name=txtName.getText();

    if(!studentManager.checkUnique("name", "department", name)){
        JOptionPane.showMessageDialog(this, "Name already exist !");
        return;
    }

    String data[]=new String[]{ name};
    int res=studentManager.addSubject(data);
    if(res==Manager.SUCCESS){
        initUI();
        enableEditAddPanel(false);
    }else if(res==Manager.CONNECT_FAIL){
        JOptionPane.showMessageDialog(this, "Connect fail !");
    }else{
        JOptionPane.showMessageDialog(this, "Error !");
    }
    studentManager.showMessage(res, this);
}

public void edit() {
    String name=txtName.getText();
    if(!name.equalsIgnoreCase((String)tblSubject.getValueAt(tblSubject.getSelectedRow(),
1))){
        if(!studentManager.checkUnique("name", "department", name)){
            JOptionPane.showMessageDialog(this, "Name already exist !");
            return;
        }
    }
    String data[]=new String[]{ studentManager.getSelectID(tblSubject)+"", name};
    int res=studentManager.editSubject(data);
    studentManager.showMessage(res, this);
}

```

```

        initUI();
        enableEditAddPanel(false);
    }

    @Override
    public void deactivate() {
        int res=studentManager.deactiveSubject(studentManager.getSelectID(tblSubject));
        studentManager.showMessage(res, this);
        if(res==Manager.SUCCESS){
            initUI();
        }
    }

    @Override
    public void search(){
        String SQL="SELECT ID, name FROM subject WHERE status=1 ";
        if(!txtNameSearch.getText().isEmpty()){
            SQL+=" AND name like '%" +txtNameSearch.getText()+"%'";
        }
        studentManager.loadTable(SQL, tblSubject, Manager.SUBJECT_MODEL);
        enableButton(false);
    }
    /**
     * 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">
    private void initComponents() {

        editAddPanel = new javax.swing.JPanel();
        labelAction = new javax.swing.JLabel();
        jLabel2 = new javax.swing.JLabel();
        txtName = new javax.swing.JTextField();
        btnSave = new javax.swing.JLabel();
        btnCancel = new javax.swing.JLabel();
        btnReset = new javax.swing.JLabel();
        jLabel5 = new javax.swing.JLabel();
        txtName1 = new javax.swing.JTextField();
        jLabel6 = new javax.swing.JLabel();
        txtName2 = new javax.swing.JTextField();
        jLabel7 = new javax.swing.JLabel();
        txtName3 = new javax.swing.JTextField();
        txtName4 = new javax.swing.JTextField();
    }

```

```
jLabel8 = new javax.swing.JLabel();
bannerPanel = new javax.swing.JPanel();
txtNameSearch = new javax.swing.JTextField();
jLabel4 = new javax.swing.JLabel();
jScrollPane1 = new javax.swing.JScrollPane();
tblSubject = new javax.swing.JTable();
btnAdd = new javax.swing.JLabel();
btnEdit = new javax.swing.JLabel();
btnDeactive = new javax.swing.JLabel();
btnExport = new javax.swing.JLabel();
btnSearch = new javax.swing.JLabel();
jLabel18 = new javax.swing.JLabel();

setBorder(null);
setFrameIcon(null);
setPreferredSize(new java.awt.Dimension(1024, 700));
getContentPane().setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

editAddPanel.setBackground(new java.awt.Color(45, 129, 171));
editAddPanel.setMinimumSize(new java.awt.Dimension(1020, 700));
editAddPanel.setPreferredSize(new java.awt.Dimension(1020, 700));
editAddPanel.setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

labelAction.setBackground(new java.awt.Color(1, 100, 151));
labelAction.setFont(new java.awt.Font("Ubuntu", 1, 24)); // NOI18N
labelAction.setForeground(new java.awt.Color(255, 255, 255));
labelAction.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
labelAction.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/Images/icons8_Course_30px.png"))); // NOI18N
labelAction.setText("Add New Subject");
labelAction.setIconTextGap(14);
labelAction.setOpaque(true);
editAddPanel.add(labelAction, new org.netbeans.lib.awtextra.AbsoluteConstraints(-2, 3,
1040, 60));

jLabel2.setBackground(new java.awt.Color(0, 102, 153));
jLabel2.setForeground(new java.awt.Color(255, 255, 255));
jLabel2.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
jLabel2.setText("Title");
jLabel2.setOpaque(true);
editAddPanel.add(jLabel2, new org.netbeans.lib.awtextra.AbsoluteConstraints(240, 130, 90,
25));

txtName.setBackground(new java.awt.Color(85, 153, 187));
txtName.setFont(new java.awt.Font("Tahoma", 1, 11)); // NOI18N
```

```
txtName.setForeground(new java.awt.Color(255, 255, 255));
txtName.setHorizontalAlignment(javax.swing.JTextField.CENTER);
txtName.setBorder(null);
editAddPanel.add(txtName, new org.netbeans.lib.awtextra.AbsoluteConstraints(330, 130,
210, 25));

btnSave.setBackground(new java.awt.Color(51, 204, 0));
btnSave.setFont(new java.awt.Font("Tahoma", 1, 12)); // NOI18N
btnSave.setForeground(new java.awt.Color(255, 255, 255));
btnSave.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
btnSave.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/Images/icons8_Save_18px.png"))); // NOI18N
btnSave.setText("Save");
btnSave.setOpaque(true);
btnSave.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        btnSaveMouseClicked(evt);
    }
});
editAddPanel.add(btnSave, new org.netbeans.lib.awtextra.AbsoluteConstraints(510, 260, 90,
24));

btnCancel.setBackground(new java.awt.Color(255, 0, 0));
btnCancel.setFont(new java.awt.Font("Tahoma", 1, 12)); // NOI18N
btnCancel.setForeground(new java.awt.Color(255, 255, 255));
btnCancel.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
btnCancel.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/Images/icons8_Cancel_18px_3.png"))); // NOI18N
btnCancel.setText("Cancel");
btnCancel.setOpaque(true);
btnCancel.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        btnCancelMouseClicked(evt);
    }
});
editAddPanel.add(btnCancel, new org.netbeans.lib.awtextra.AbsoluteConstraints(790, 260,
90, 24));

btnReset.setBackground(new java.awt.Color(0, 102, 255));
btnReset.setFont(new java.awt.Font("Tahoma", 1, 12)); // NOI18N
btnReset.setForeground(new java.awt.Color(255, 255, 255));
btnReset.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
btnReset.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/Images/icons8_Reset_18px_1.png"))); // NOI18N
```

```
btnReset.setText("Reset");
btnReset.setOpaque(true);
btnReset.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        btnResetMouseClicked(evt);
    }
});
editAddPanel.add(btnReset, new org.netbeans.lib.awtextra.AbsoluteConstraints(240, 260,
90, 24));

jLabel5.setBackground(new java.awt.Color(0, 102, 153));
jLabel5.setForeground(new java.awt.Color(255, 255, 255));
jLabel5.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
jLabel5.setText("ID");
jLabel5.setOpaque(true);
editAddPanel.add(jLabel5, new org.netbeans.lib.awtextra.AbsoluteConstraints(240, 160, 90,
25));

txtName1.setBackground(new java.awt.Color(85, 153, 187));
txtName1.setFont(new java.awt.Font("Tahoma", 1, 11)); // NOI18N
txtName1.setForeground(new java.awt.Color(255, 255, 255));
txtName1.setHorizontalAlignment(javax.swing.JTextField.CENTER);
txtName1.setBorder(null);
editAddPanel.add(txtName1, new org.netbeans.lib.awtextra.AbsoluteConstraints(330, 160,
210, 25));

jLabel6.setBackground(new java.awt.Color(0, 102, 153));
jLabel6.setForeground(new java.awt.Color(255, 255, 255));
jLabel6.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
jLabel6.setText("Class");
jLabel6.setOpaque(true);
editAddPanel.add(jLabel6, new org.netbeans.lib.awtextra.AbsoluteConstraints(560, 130, 90,
25));

txtName2.setBackground(new java.awt.Color(85, 153, 187));
txtName2.setFont(new java.awt.Font("Tahoma", 1, 11)); // NOI18N
txtName2.setForeground(new java.awt.Color(255, 255, 255));
txtName2.setHorizontalAlignment(javax.swing.JTextField.CENTER);
txtName2.setBorder(null);
editAddPanel.add(txtName2, new org.netbeans.lib.awtextra.AbsoluteConstraints(650, 130,
240, 25));

jLabel7.setBackground(new java.awt.Color(0, 102, 153));
jLabel7.setForeground(new java.awt.Color(255, 255, 255));
jLabel7.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
jLabel7.setText("Teacher");
```

```
jLabel7.setOpaque(true);
editAddPanel.add(jLabel7, new org.netbeans.lib.awtextra.AbsoluteConstraints(560, 160, 90,
25));

txtName3.setBackground(new java.awt.Color(85, 153, 187));
txtName3.setFont(new java.awt.Font("Tahoma", 1, 11)); // NOI18N
txtName3.setForeground(new java.awt.Color(255, 255, 255));
txtName3.setHorizontalAlignment(javax.swing.JTextField.CENTER);
txtName3.setBorder(null);
editAddPanel.add(txtName3, new org.netbeans.lib.awtextra.AbsoluteConstraints(650, 160,
240, 25));

txtName4.setBackground(new java.awt.Color(85, 153, 187));
txtName4.setFont(new java.awt.Font("Tahoma", 1, 11)); // NOI18N
txtName4.setForeground(new java.awt.Color(255, 255, 255));
txtName4.setHorizontalAlignment(javax.swing.JTextField.CENTER);
txtName4.setBorder(null);
editAddPanel.add(txtName4, new org.netbeans.lib.awtextra.AbsoluteConstraints(330, 190,
210, 25));

jLabel8.setBackground(new java.awt.Color(0, 102, 153));
jLabel8.setForeground(new java.awt.Color(255, 255, 255));
jLabel8.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
jLabel8.setText("Room No");
jLabel8.setOpaque(true);
editAddPanel.add(jLabel8, new org.netbeans.lib.awtextra.AbsoluteConstraints(240, 190, 90,
25));

getContentPane().add(editAddPanel, new org.netbeans.lib.awtextra.AbsoluteConstraints(0,
0, 1020, 700));

bannerPanel.setBackground(new java.awt.Color(45, 129, 171));
bannerPanel.setMinimumSize(new java.awt.Dimension(1020, 700));
bannerPanel.setPreferredSize(new java.awt.Dimension(1020, 700));
bannerPanel.setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

txtNameSearch.setBackground(new java.awt.Color(85, 153, 187));
txtNameSearch.setForeground(new java.awt.Color(255, 255, 255));
txtNameSearch.setBorder(null);
txtNameSearch.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        searchAction(evt);
    }
});
bannerPanel.add(txtNameSearch, new org.netbeans.lib.awtextra.AbsoluteConstraints(370,
130, 280, 25));
```

```
jLabel4.setBackground(new java.awt.Color(1, 100, 151));
jLabel4.setFont(new java.awt.Font("Ubuntu", 1, 24)); // NOI18N
jLabel4.setForeground(new java.awt.Color(255, 255, 255));
jLabel4.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
jLabel4.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/Images/icons8_Course_30px.png"))); //
NOI18N
jLabel4.setText("Subjects");
jLabel4.setIconTextGap(14);
jLabel4.setOpaque(true);
bannerPanel.add(jLabel4, new org.netbeans.lib.awtextra.AbsoluteConstraints(-3, 0, 1040,
60));

jScrollPane1.setBorder(javax.swing.BorderFactory.createLineBorder(new java.awt.Color(4,
74, 121)));

tblSubject.setBackground(new java.awt.Color(4, 74, 121));
tblSubject.setForeground(new java.awt.Color(255, 255, 255));
tblSubject.setModel(new javax.swing.table.DefaultTableModel(
new Object [][] {
    {null, null, null, null, null},
    {null, null, null, null, null}
},
new String [] {
    "ID", "Subject Title", "Teacher", "Class", "Room No"
}
) {
    boolean[] canEdit = new boolean [] {
        false, false, true, true, true
    };
    public boolean isCellEditable(int rowIndex, int columnIndex) {
        return canEdit [columnIndex];
    }
});
tblSubject.setSelectionBackground(new java.awt.Color(0, 204, 102));
tblSubject.setShowHorizontalLines(false);
tblSubject.setShowVerticalLines(false);
tblSubject.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        tblSubjectMouseClicked(evt);
    }
});
jScrollPane1.setViewportView(tblSubject);
```

```
        bannerPanel.add(jScrollPane1, new org.netbeans.lib.awtextra.AbsoluteConstraints(160, 210, 740, 290));  
  
        btnAdd.setBackground(new java.awt.Color(51, 204, 0));  
        btnAdd.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N  
        btnAdd.setForeground(new java.awt.Color(255, 255, 255));  
        btnAdd.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);  
        btnAdd.setIcon(new javax.swing.ImageIcon(getClass().getResource("/Images/icons8_Add_New_17px.png"))); // NOI18N  
        btnAdd.setText("Add");  
        btnAdd.setOpaque(true);  
        btnAdd.addMouseListener(new java.awt.event.MouseAdapter() {  
            public void mouseClicked(java.awt.event.MouseEvent evt) {  
                btnAddMouseClicked(evt);  
            }  
        });  
        bannerPanel.add(btnAdd, new org.netbeans.lib.awtextra.AbsoluteConstraints(160, 520, 88, 25));  
  
        btnEdit.setBackground(new java.awt.Color(204, 0, 204));  
        btnEdit.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N  
        btnEdit.setForeground(new java.awt.Color(255, 255, 255));  
        btnEdit.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);  
        btnEdit.setIcon(new javax.swing.ImageIcon(getClass().getResource("/Images/icons8_Edit_Row_17px.png"))); // NOI18N  
        btnEdit.setText("Edit");  
        btnEdit.setOpaque(true);  
        btnEdit.addMouseListener(new java.awt.event.MouseAdapter() {  
            public void mouseClicked(java.awt.event.MouseEvent evt) {  
                btnEditMouseClicked(evt);  
            }  
        });  
        bannerPanel.add(btnEdit, new org.netbeans.lib.awtextra.AbsoluteConstraints(350, 520, 88, 25));  
  
        btnDeactive.setBackground(new java.awt.Color(255, 0, 0));  
        btnDeactive.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N  
        btnDeactive.setForeground(new java.awt.Color(255, 255, 255));  
        btnDeactive.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);  
        btnDeactive.setIcon(new javax.swing.ImageIcon(getClass().getResource("/Images/icons8_Radio_Active_17px.png"))); // NOI18N  
        btnDeactive.setText("Deactivate");  
        btnDeactive.setOpaque(true);
```

```
btnDeactive.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        btnDeactiveMouseClicked(evt);
    }
});
bannerPanel.add(btnDeactive, new org.netbeans.lib.awtextra.AbsoluteConstraints(580, 520, 100, 25));

btnExport.setBackground(new java.awt.Color(0, 0, 204));
btnExport.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N
btnExport.setForeground(new java.awt.Color(255, 255, 255));
btnExport.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
btnExport.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/Images/icons8_Outbox_17px.png"))); // NOI18N
btnExport.setText("Export");
btnExport.setOpaque(true);
btnExport.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        btnExportMouseClicked(evt);
    }
});
bannerPanel.add(btnExport, new org.netbeans.lib.awtextra.AbsoluteConstraints(810, 520, 88, 25));

btnSearch.setBackground(new java.awt.Color(255, 102, 0));
btnSearch.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N
btnSearch.setForeground(new java.awt.Color(255, 255, 255));
btnSearch.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
btnSearch.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/Images/icons8_Google_Web_Search_17px.png
"))); // NOI18N
btnSearch.setText("Search");
btnSearch.setOpaque(true);
btnSearch.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        btnSearchMouseClicked(evt);
    }
});
bannerPanel.add(btnSearch, new org.netbeans.lib.awtextra.AbsoluteConstraints(650, 130, 88, 25));

jLabel18.setBackground(new java.awt.Color(0, 102, 153));
jLabel18.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N
jLabel18.setForeground(new java.awt.Color(255, 255, 255));
jLabel18.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
```

```
jLabel18.setText("Name");
jLabel18.setOpaque(true);
bannerPanel.add(jLabel18, new org.netbeans.lib.awtextra.AbsoluteConstraints(290, 130, 88, 25));

        getContentPane().add(bannerPanel, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 0, 1020, 700));

        pack();
}// </editor-fold>

private void searchAction(java.awt.event.ActionEvent evt) {

}

private void tblSubjectMouseClicked(java.awt.event.MouseEvent evt) {
    enableButton(true);
}

private void btnSaveMouseClicked(java.awt.event.MouseEvent evt) {
    save();      // TODO add your handling code here:
}

private void btnCancelMouseClicked(java.awt.event.MouseEvent evt) {
    enableEditAddPanel(false);      // TODO add your handling code here:
}

private void btnResetMouseClicked(java.awt.event.MouseEvent evt) {
    loadEdit();      // TODO add your handling code here:
}

private void btnAddMouseClicked(java.awt.event.MouseEvent evt) {
act=ADD;
    enableEditAddPanel(true);
    btnReset.setVisible(false);      // TODO add your handling code here:
}

private void btnEditMouseClicked(java.awt.event.MouseEvent evt) {
if(studentManager.getSelectRow(tblSubject) !=null){
    act=EDIT;
    enableEditAddPanel(true);
}      // TODO add your handling code here:
}

private void btnDeactiveMouseClicked(java.awt.event.MouseEvent evt) {
if(studentManager.confirmRemove(this, "Are you sure ?")==0){
```

```
        deactivate();
    }      // TODO add your handling code here:
}

private void btnExportMouseClicked(java.awt.event.MouseEvent evt) {
export();      // TODO add your handling code here:
}

private void btnSearchMouseClicked(java.awt.event.MouseEvent evt) {
    search();      // TODO add your handling code here:
}

// Variables declaration - do not modify
private javax.swing.JPanel bannerPanel;
private javax.swing.JLabel btnAdd;
private javax.swing.JLabel btnCancel;
private javax.swing.JLabel btnDeactive;
private javax.swing.JLabel btnEdit;
private javax.swing.JLabel btnExport;
private javax.swing.JLabel btnReset;
private javax.swing.JLabel btnSave;
private javax.swing.JLabel btnSearch;
private javax.swing.JPanel editAddPanel;
private javax.swing.JLabel jLabel18;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel4;
private javax.swing.JLabel jLabel5;
private javax.swing.JLabel jLabel6;
private javax.swing.JLabel jLabel7;
private javax.swing.JLabel jLabel8;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JLabel labelAction;
private javax.swing.JTable tblSubject;
private javax.swing.JTextField txtName;
private javax.swing.JTextField txtName1;
private javax.swing.JTextField txtName2;
private javax.swing.JTextField txtName3;
private javax.swing.JTextField txtName4;
private javax.swing.JTextField txtNameSearch;
// End of variables declaration

@Override
public void enableButton(boolean b) {
    btnEdit.setEnabled(b);
    btnDeactive.setEnabled(b);
}
```

```

@Override
public boolean validateForm() {
    String name=txtName.getText();
    if(name==null || name.isEmpty()){
        JOptionPane.showMessageDialog(this, "Name is required !");
        return false;
    }
    return true;
}

private void export() {
    ExportFile export =new ExportFile(tblSubject, ExportFile.SUBJECT);
    export.toExcel();
}
}

```

- **For Marks Updation Class**

```

/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */
package StudentMan;

import Controls.StudentManager;
import java.util.Vector;
import TableModel.Mark;
import TableModel.Remark;
import TableModel.Subject;

/**
 *
 * @author Aeon(Ankit RudrAshokJha)
 */
public class MarkUpdate extends javax.swing.JDialog {
    private Mark mark;
    private StudentManager studentManager;
    /**
     * Creates new form StudentMoreInfoDialog
     */
    public MarkUpdate(java.awt.Frame parent, boolean modal, Vector data, StudentManager
studentManager) {
        super(parent, modal);
        if(data !=null)

```

```

        this.mark=new Mark(Integer.parseInt(data.get(0).toString()),
Integer.parseInt(data.get(2).toString()), studentManager.parseDate(data.get(3).toString()), new
Subject(data.get(1).toString()));
        this.studentManager=studentManager;
        initComponents();
        setLocationRelativeTo(this);
        initUI();
    }
private void initUI(){
    studentManager.loadCombobox("SELECT name FROM subject", comboSubject);
    if(mark==null){
        return;
    }
    comboSubject.setSelectedItem(mark.getSubject().getName());
    txtPoint.setText(mark.getPoint()+"");
    dPic.setDate(mark.getTime());
}
private void save(){
    if(mark==null){
        mark=new Mark();
    }
    mark.setSubject(new Subject((String)comboSubject.getSelectedItem()));
    mark.setPoint(Integer.parseInt(txtPoint.getText()));
    mark.setTime(dPic.getDate());
    dispose();
}
public Mark getMark(){
    return mark;
}

/**
 * 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">
private void initComponents() {

    jPanel1 = new javax.swing.JPanel();
    jLabel2 = new javax.swing.JLabel();
    jLabel4 = new javax.swing.JLabel();
    jLabel1 = new javax.swing.JLabel();
    comboSubject = new javax.swing.JComboBox();
    txtPoint = new javax.swing.JTextField();
    dPic = new com.toedter.calendar.JDateChooser();
}

```

```
labelTitle = new javax.swing.JLabel();
jLabel9 = new javax.swing.JLabel();
jLabel8 = new javax.swing.JLabel();
jLabel6 = new javax.swing.JLabel();

setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE_ON_CLOSE);

jPanel1.setBackground(new java.awt.Color(45, 129, 171));
jPanel1.setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

jLabel2.setBackground(new java.awt.Color(0, 102, 153));
jLabel2.setForeground(new java.awt.Color(255, 255, 255));
jLabel2.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
jLabel2.setText("Mark");
jLabel2.setOpaque(true);
jPanel1.add(jLabel2, new org.netbeans.lib.awtextra.AbsoluteConstraints(70, 110, 90, 25));

jLabel4.setBackground(new java.awt.Color(0, 102, 153));
jLabel4.setForeground(new java.awt.Color(255, 255, 255));
jLabel4.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
jLabel4.setText("Time");
jLabel4.setOpaque(true);
jPanel1.add(jLabel4, new org.netbeans.lib.awtextra.AbsoluteConstraints(70, 140, 88, 25));

jLabel1.setBackground(new java.awt.Color(0, 102, 153));
jLabel1.setForeground(new java.awt.Color(255, 255, 255));
jLabel1.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
jLabel1.setText("Subject");
jLabel1.setOpaque(true);
jPanel1.add(jLabel1, new org.netbeans.lib.awtextra.AbsoluteConstraints(70, 80, 90, 25));

comboSubject.setBackground(new java.awt.Color(85, 153, 187));
comboSubject.setEditable(true);
comboSubject.setBorder(null);
jPanel1.add(comboSubject, new org.netbeans.lib.awtextra.AbsoluteConstraints(160, 80, 240, 25));

txtPoint.setBackground(new java.awt.Color(85, 153, 187));
txtPoint.setForeground(new java.awt.Color(255, 255, 255));
txtPoint.setHorizontalAlignment(javax.swing.JTextField.CENTER);
txtPoint.setBorder(null);
jPanel1.add(txtPoint, new org.netbeans.lib.awtextra.AbsoluteConstraints(160, 110, 240, 25));

jPanel1.add(dPic, new org.netbeans.lib.awtextra.AbsoluteConstraints(160, 140, 240, 25));

labelTitle.setBackground(new java.awt.Color(0, 102, 153));
```

```
labelTitle.setFont(new java.awt.Font("Ubuntu", 1, 24)); // NOI18N
labelTitle.setForeground(new java.awt.Color(255, 255, 255));
labelTitle.setHorizontalTextPosition(javax.swing.SwingConstants.CENTER);
labelTitle.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/Images/icons8_Exam_24px.png"))); // NOI18N
labelTitle.setText("Add Mark");
labelTitle.setIconTextGap(14);
labelTitle.setOpaque(true);
jPanel1.add(labelTitle, new org.netbeans.lib.awtextra.AbsoluteConstraints(1, 0, 480, 60));

jLabel9.setBackground(new java.awt.Color(0, 153, 255));
jLabel9.setFont(new java.awt.Font("Tahoma", 1, 12)); // NOI18N
jLabel9.setForeground(new java.awt.Color(255, 255, 255));
jLabel9.setHorizontalTextPosition(javax.swing.SwingConstants.CENTER);
jLabel9.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/Images/icons8_Reset_18px_1.png"))); // NOI18N
jLabel9.setText("Reset");
jLabel9.setOpaque(true);
jLabel9.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        jLabel9MouseClicked(evt);
    }
});
jPanel1.add(jLabel9, new org.netbeans.lib.awtextra.AbsoluteConstraints(70, 200, 90, 24));

jLabel8.setBackground(new java.awt.Color(51, 204, 0));
jLabel8.setFont(new java.awt.Font("Tahoma", 1, 12)); // NOI18N
jLabel8.setForeground(new java.awt.Color(255, 255, 255));
jLabel8.setHorizontalTextPosition(javax.swing.SwingConstants.CENTER);
jLabel8.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/Images/icons8_Save_18px.png"))); // NOI18N
jLabel8.setText("Save");
jLabel8.setOpaque(true);
jLabel8.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        jLabel8MouseClicked(evt);
    }
});
jPanel1.add(jLabel8, new org.netbeans.lib.awtextra.AbsoluteConstraints(180, 200, 90, 24));

jLabel6.setBackground(new java.awt.Color(255, 0, 0));
jLabel6.setFont(new java.awt.Font("Tahoma", 1, 12)); // NOI18N
jLabel6.setForeground(new java.awt.Color(255, 255, 255));
jLabel6.setHorizontalTextPosition(javax.swing.SwingConstants.CENTER);
```

```

        jLabel6.setIcon(new
        javax.swing.ImageIcon(getClass().getResource("/Images/icons8_Cancel_18px_3.png"))); // NOI18N
        jLabel6.setText("Cancel");
        jLabel6.setOpaque(true);
        jLabel6.addMouseListener(new java.awt.event.MouseAdapter() {
            public void mouseClicked(java.awt.event.MouseEvent evt) {
                jLabel6MouseClicked(evt);
            }
        });
        jPanel1.add(jLabel6, new org.netbeans.lib.awtextra.AbsoluteConstraints(310, 200, 90, 24));

        javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
        getContentPane().setLayout(layout);
        layout.setHorizontalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addContainerGap()
                .addComponent(jPanel1, javax.swing.GroupLayout.PREFERRED_SIZE, 478,
                javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGap(0, 0, Short.MAX_VALUE))
        );
        layout.setVerticalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addContainerGap()
                .addComponent(jPanel1, javax.swing.GroupLayout.PREFERRED_SIZE, 272,
                javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGap(0, 0, Short.MAX_VALUE))
        );

        pack();
    }// </editor-fold>

    private void jLabel9MouseClicked(java.awt.event.MouseEvent evt) {
        initUI(); // TODO add your handling code here:
    }

    private void jLabel8MouseClicked(java.awt.event.MouseEvent evt) {
        save(); // TODO add your handling code here:
    }

    private void jLabel6MouseClicked(java.awt.event.MouseEvent evt) {

        dispose(); // TODO add your handling code here:
    }

    /**
     * @param args the command line arguments
     */

```

```

// Variables declaration - do not modify
private javax.swing.JComboBox comboSubject;
private com.toedter.calendar.JDateChooser dPic;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel4;
private javax.swing.JLabel jLabel6;
private javax.swing.JLabel jLabel8;
private javax.swing.JLabel jLabel9;
private javax.swing.JPanel jPanel1;
private javax.swing.JLabel labelTitle;
private javax.swing.JTextField txtPoint;
// End of variables declaration
}

```

- **For Remarks Updation**

```

/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */
package StudentMan;

import Controls.StudentManager;
import TableModel.Promotion;
import TableModel.Remark;

/**
 *
 * @author Aeon(Ankit RudrAshokJha)
 */
public class RemarkDetail extends javax.swing.JDialog {
    private Remark remark;
    private StudentManager studentManager;
    /**
     * Creates new form StudentMoreInfoDialog
     */
    public RemarkDetail(java.awt.Frame parent, boolean modal, Remark remark, StudentManager studentManager) {
        super(parent, modal);
        this.remark=remark;
        this.studentManager=studentManager;
        initComponents();
        setLocationRelativeTo(this);
        initUI();
    }
    private void initUI(){

```

```
labelStaffNo.setText(remark.getStaff().getNo()+"");
labelStaffName.setText(remark.getStaff().getName());
txtContent.setText(remark.getContent());
labelTime.setText(studentManager.formatDate(remark.getTime()));

}

<太后半身像
 /**
 * 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">
private void initComponents() {

    jPanel1 = new javax.swing.JPanel();
    jLabel2 = new javax.swing.JLabel();
    jLabel4 = new javax.swing.JLabel();
    jLabel5 = new javax.swing.JLabel();
    labelStaffName = new javax.swing.JLabel();
    labelTime = new javax.swing.JLabel();
    jScrollPane1 = new javax.swing.JScrollPane();
    txtContent = new javax.swing.JTextPane();
    jLabel1 = new javax.swing.JLabel();
    labelStaffNo = new javax.swing.JLabel();
    jLabel11 = new javax.swing.JLabel();
    jLabel6 = new javax.swing.JLabel();

    setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE_ON_CLOSE);

    jPanel1.setBackground(new java.awt.Color(45, 129, 171));
    jPanel1.setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

    jLabel2.setBackground(new java.awt.Color(0, 102, 153));
    jLabel2.setFont(new java.awt.Font("Tahoma", 1, 12)); // NOI18N
    jLabel2.setForeground(new java.awt.Color(255, 255, 255));
    jLabel2.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
    jLabel2.setText("Staff Name");
    jLabel2.setOpaque(true);
    jPanel1.add(jLabel2, new org.netbeans.lib.awtextra.AbsoluteConstraints(70, 120, 90, 25));

    jLabel4.setBackground(new java.awt.Color(0, 102, 153));
    jLabel4.setFont(new java.awt.Font("Tahoma", 1, 12)); // NOI18N
    jLabel4.setForeground(new java.awt.Color(255, 255, 255));
    jLabel4.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
    jPanel1.add(jLabel4, new org.netbeans.lib.awtextra.AbsoluteConstraints(70, 120, 90, 25));

    setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE_ON_CLOSE);

    jLabel1.setText("Remark Details");

    jLabel11.setText("Staff No");

    jLabel6.setText("Time");

    txtContent.setContentType("text/html");
    txtContent.setEditable(false);

    javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
    getContentPane().setLayout(layout);
    layout.setHorizontalGroup(
        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addContainerGap()
                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                    .addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addGroup(layout.createSequentialGroup()
                        .addComponent(jLabel1)
                        .addGap(18, 18, 18)
                        .addComponent(jLabel11)
                        .addGap(18, 18, 18)
                        .addComponent(jLabel6)
                        .addGap(18, 18, 18)
                        .addComponent(txtContent, javax.swing.GroupLayout.PREFERRED_SIZE, 350, javax.swing.GroupLayout.PREFERRED_SIZE))
                    .addGroup(layout.createSequentialGroup()
                        .addComponent(jLabel4)
                        .addGap(18, 18, 18)
                        .addComponent(jLabel2)
                        .addGap(18, 18, 18)
                        .addComponent(labelStaffName)
                        .addGap(18, 18, 18)
                        .addComponent(labelTime)
                        .addGap(18, 18, 18)
                        .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 350, javax.swing.GroupLayout.PREFERRED_SIZE)))
                .addContainerGap())
    );
    layout.setVerticalGroup(
        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addContainerGap()
                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                    .addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addGroup(layout.createSequentialGroup()
                        .addComponent(jLabel1)
                        .addGap(18, 18, 18)
                        .addComponent(jLabel11)
                        .addGap(18, 18, 18)
                        .addComponent(jLabel6)
                        .addGap(18, 18, 18)
                        .addComponent(txtContent, javax.swing.GroupLayout.PREFERRED_SIZE, 350, javax.swing.GroupLayout.PREFERRED_SIZE))
                    .addGroup(layout.createSequentialGroup()
                        .addComponent(jLabel4)
                        .addGap(18, 18, 18)
                        .addComponent(jLabel2)
                        .addGap(18, 18, 18)
                        .addComponent(labelStaffName)
                        .addGap(18, 18, 18)
                        .addComponent(labelTime)
                        .addGap(18, 18, 18)
                        .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 350, javax.swing.GroupLayout.PREFERRED_SIZE)))
                .addContainerGap())
    );
}
}
```

```
jLabel4.setText("Content");
jLabel4.setOpaque(true);
jPanel1.add(jLabel4, new org.netbeans.lib.awtextra.AbsoluteConstraints(70, 150, 90, 90));

jLabel5.setBackground(new java.awt.Color(255, 0, 0));
jLabel5.setFont(new java.awt.Font("Tahoma", 1, 12)); // NOI18N
jLabel5.setForeground(new java.awt.Color(255, 255, 255));
jLabel5.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
jLabel5.setText("Close");
jLabel5.setOpaque(true);
jLabel5.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        jLabel5MouseClicked(evt);
    }
});
jPanel1.add(jLabel5, new org.netbeans.lib.awtextra.AbsoluteConstraints(460, 250, 90, 25));

labelStaffName.setBackground(new java.awt.Color(85, 153, 187));
labelStaffName.setFont(new java.awt.Font("Tahoma", 1, 12)); // NOI18N
labelStaffName.setForeground(new java.awt.Color(255, 255, 255));
labelStaffName.setText("Staff Name");
labelStaffName.setOpaque(true);
jPanel1.add(labelStaffName, new org.netbeans.lib.awtextra.AbsoluteConstraints(160, 120, 390, 25));

labelTime.setBackground(new java.awt.Color(85, 153, 187));
labelTime.setFont(new java.awt.Font("Tahoma", 1, 12)); // NOI18N
labelTime.setForeground(new java.awt.Color(255, 255, 255));
labelTime.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
labelTime.setText("Time");
labelTime.setOpaque(true);
jPanel1.add(labelTime, new org.netbeans.lib.awtextra.AbsoluteConstraints(160, 250, 280, 25));

jScrollPane1.setBackground(new java.awt.Color(85, 153, 187));
jScrollPane1.setBorder(null);

txtContent.setEditable(false);
txtContent.setBackground(new java.awt.Color(85, 153, 187));
txtContent.setBorder(null);
txtContent.setFont(new java.awt.Font("Tahoma", 1, 12)); // NOI18N
txtContent.setForeground(new java.awt.Color(255, 255, 255));
jScrollPane1.setViewportView(txtContent);

jPanel1.add(jScrollPane1, new org.netbeans.lib.awtextra.AbsoluteConstraints(160, 150, 390, 90));
```

```
jLabel1.setBackground(new java.awt.Color(0, 102, 153));
jLabel1.setFont(new java.awt.Font("Tahoma", 1, 12)); // NOI18N
jLabel1.setForeground(new java.awt.Color(255, 255, 255));
jLabel1.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
jLabel1.setText("Staff No");
jLabel1.setOpaque(true);
jPanel1.add(jLabel1, new org.netbeans.lib.awtextra.AbsoluteConstraints(70, 90, 90, 25));

labelStaffNo.setBackground(new java.awt.Color(85, 153, 187));
labelStaffNo.setFont(new java.awt.Font("Tahoma", 1, 12)); // NOI18N
labelStaffNo.setForeground(new java.awt.Color(255, 255, 255));
labelStaffNo.setText("Staff No");
labelStaffNo.setOpaque(true);
jPanel1.add(labelStaffNo, new org.netbeans.lib.awtextra.AbsoluteConstraints(160, 90, 390, 25));

jLabel11.setBackground(new java.awt.Color(0, 102, 153));
jLabel11.setFont(new java.awt.Font("Ubuntu", 1, 24)); // NOI18N
jLabel11.setForeground(new java.awt.Color(255, 255, 255));
jLabel11.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
jLabel11.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/Images/icons8_Comments_30px.png"))); // NOI18N
jLabel11.setText("Remark Detail");
jLabel11.setToolTipText("");
jLabel11.setIconTextGap(14);
jLabel11.setOpaque(true);
jPanel1.add(jLabel11, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 0, 616, 50));

jLabel6.setBackground(new java.awt.Color(0, 102, 153));
jLabel6.setFont(new java.awt.Font("Tahoma", 1, 12)); // NOI18N
jLabel6.setForeground(new java.awt.Color(255, 255, 255));
jLabel6.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
jLabel6.setText("Time");
jLabel6.setOpaque(true);
jPanel1.add(jLabel6, new org.netbeans.lib.awtextra.AbsoluteConstraints(70, 250, 90, 25));

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addGap(0, 0, Short.MAX_VALUE)
            .addComponent(jPanel1, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(0, 0, Short.MAX_VALUE))
        .addGroup(layout.createSequentialGroup()
            .addGap(0, 0, Short.MAX_VALUE)
            .addComponent(jLabel6, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(0, 0, Short.MAX_VALUE))
        .addGroup(layout.createSequentialGroup()
            .addGap(0, 0, Short.MAX_VALUE)
            .addComponent(jLabel11, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(0, 0, Short.MAX_VALUE))
        .addGroup(layout.createSequentialGroup()
            .addGap(0, 0, Short.MAX_VALUE)
            .addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(0, 0, Short.MAX_VALUE))
    )
);
```

```
 );
layout.setVerticalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT_SIZE, 331,
Short.MAX_VALUE)
);

    pack();
}// </editor-fold>

private void jLabel5MouseClicked(java.awt.event.MouseEvent evt) {
dispose();      // TODO add your handling code here:
}

<**
 * @param args the command line arguments
 */
// Variables declaration - do not modify
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel11;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel4;
private javax.swing.JLabel jLabel5;
private javax.swing.JLabel jLabel6;
private javax.swing.JPanel jPanel1;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JLabel labelStaffName;
private javax.swing.JLabel labelStaffNo;
private javax.swing.JLabel labelTime;
private javax.swing.JTextPane txtContent;
// End of variables declaration
}
```

6. BIBLIOGRAPHY

- Herbert Scheldt, **Java Complete Reference**, Seventh Edition, Tata McGraw Hill Edition.
- Ian Somerville, **Software Engineering**, Third Edition, Pearson Education
- Ali Bahrami, **Object-Oriented System Development**, Third Edition, Tata McGraw Hill Edition.
- Elmarsi and Navathe, **Fundamentals of Database System** (Third Edition), Addison Wesley.
- **MySQL Complete Reference 5.9**
- <https://stackoverflow.com/>
- <http://www.lynda.com/>
- <https://learnui.design/>

7. Future Perspectives

- In further future development I may will be able to provide distributed database integration and Network work integration to this standalone software.
- Further development for improved performance of this Software will also be in its way. We may able to port this software or various options of database applications like Oracle, Microsoft sql server etc.
- In future we will be able to provide webbased massaging for the various updates about the students and staffs.
- Schedule management, Event Management, Bus Management and SMS based Updation system will we seen in further releases