SCS 2104 Programming 03

Take Home Assignment – 2018

Name: W.E.M.S.S. Ekanayake
Admission Number: 2016/CS/045
Index Number: 16000455

Implementation Details with Relevant User Interfaces

Database Connection

Database connection is handles by dbConnection.java using “JDBC.Driver”.
Database name: ***nsbm***

Login Activity

The program start with the Login interface, where 3 user accesses; **Administrator, Student Manager, Exam Manager** (disabled) are provided.
The Login activity is handles using Login.java; where the users are verified by referring to **User** table in database;

Administrator Login

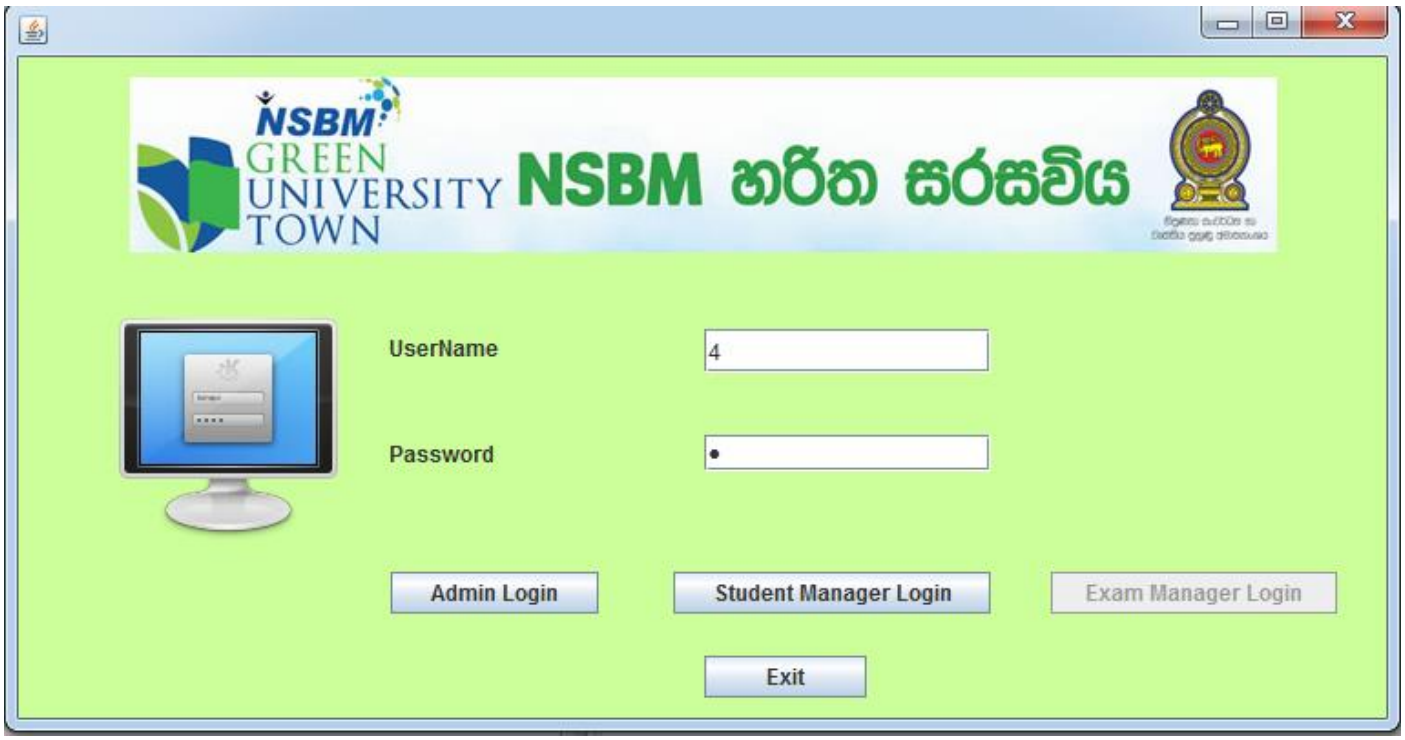
- Get records from database where the username and password is provided data and isAdmin flag is 1.
- Get the count of records; if there is only one record, login successful and show the Administrator Dashboard. Else there are no records or more than one record, an error message is shown to try again.

Student Manager Login

- Same as the Admin login procedure, here the Student Manager Dashboard is shown on successful login.

Exam Manager Login

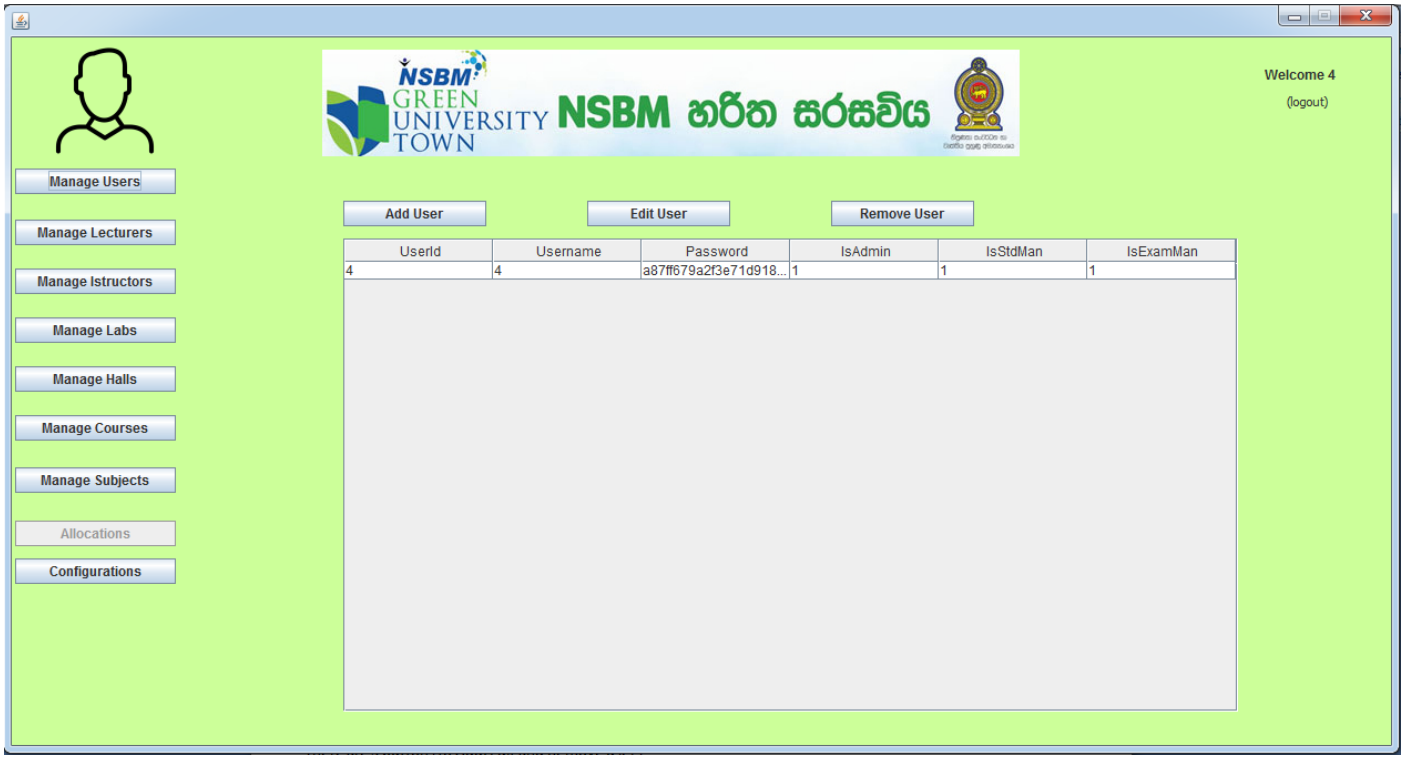
- This option is disabled



Note: Username: 4 and Password: 4 for both Admin and Student Manager Logins

Administrator Dashboard

Administrator dashboard (AdminDash.java) loads with Manage User interface, there are 9 functionalities for administrator like *Manage Users, Lecturers, Instructors, Labs, Halls, Courses, Subjects, Allocation (includes allocation of lecturers, instructors, halls, labs for subjects*; but disabled) and finally *Configuration*.



When the relevant buttons are clicked, JPanels are loaded to middle_panel of AdminDash

Manage Users

Associates: JPanel_ManageUser.java, AddEditUser.java, User.java, table: user

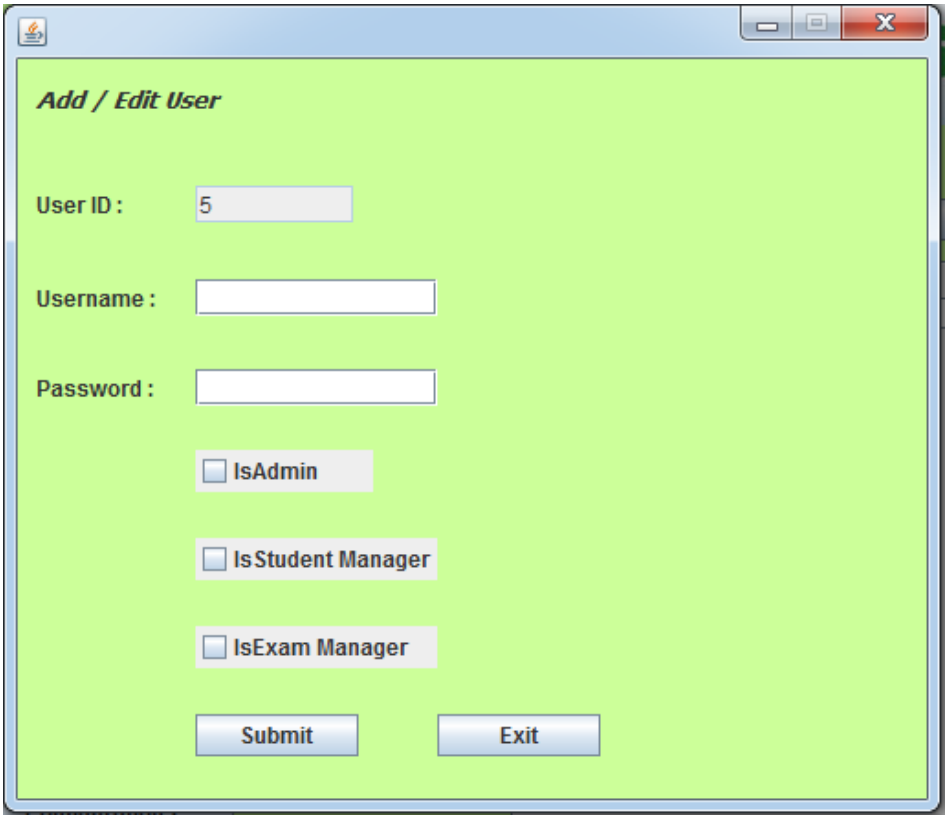
The JPanel_ManageUser.java is the interface to manage Users, consist of a JTable to retrieve user information from database.

JTable is filled using the ResultSet obtained by retriveUsers() method in User Class

There are 3 buttons to Add, Edit and Remove users

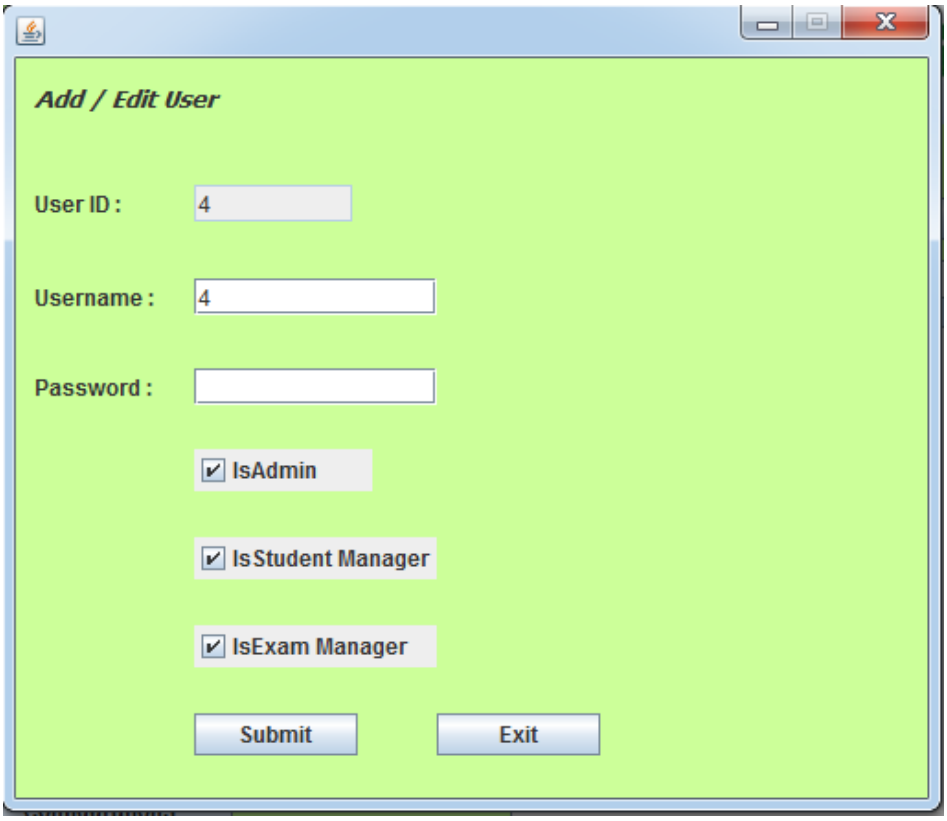
Add:

- Disable AdminDash Frame and opens the JFrame AddEditUser
- Next User ID is automatically filled; others should filled by administrator
- On “Submit” a new User object is created by adding filled details, via addData() method.
- This user information are added to DB using saveNewData() method
- All the field of AddEditUser are Reset
- Administrator can add more users or Click on “Exit”
- AdminDash is re-enabled, AddEditUser disappears and JTable is update on “Exit”



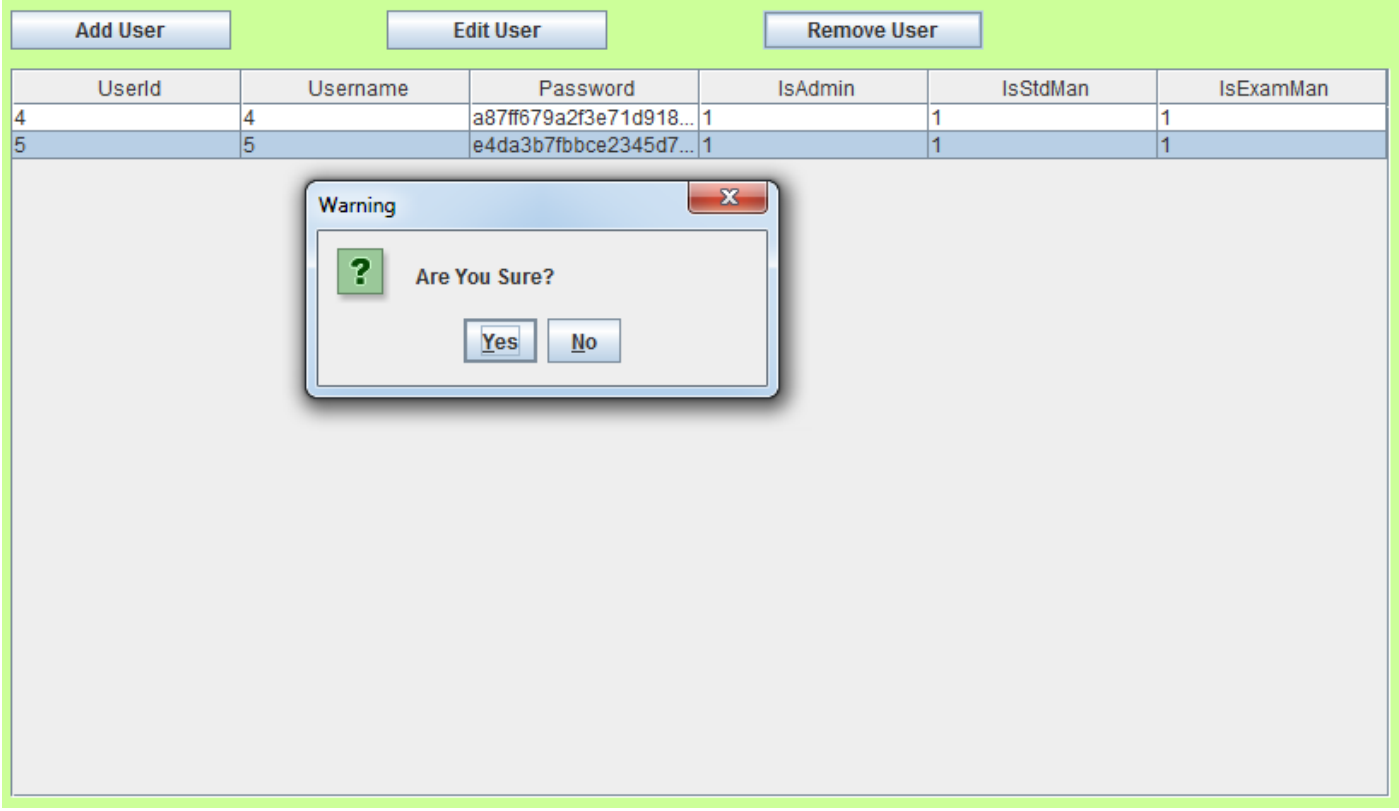
Edit:

- A new user object is created *tempuser*. (A record has to be selected)
- Set the attributes of *tempuser* using changeData() method by information in JTable.
- Assigning *tempuser* for a static reference “tempuser” in SharedData.java
- Disable AdminDash Frame and opens the JFrame AddEditUser
- The textboxes and checkboxes are filled using the data in SharedData. tempuser
- Administrator can edit the user information
- On “Submit”; if the password is changed, the DB is updated using saveData() method else using saveDataNoPass() method.
- AdminDash is re-enabled, AddEditUser disappears and JTable is updated.



Remove:

- A record is selected from the JTable
- Ask for confirmation when “Remove User” is clicked on.
- If confirm, remove the relevant user using removeUser() method of User object



Similar procedure is associated with other Manage Option (Lecturers, Instructors, Labs, Halls, Courses, Subjects) interacting with relevant tables, interfaces and classes/objects.

Allocation: Option is disabled

Configurations

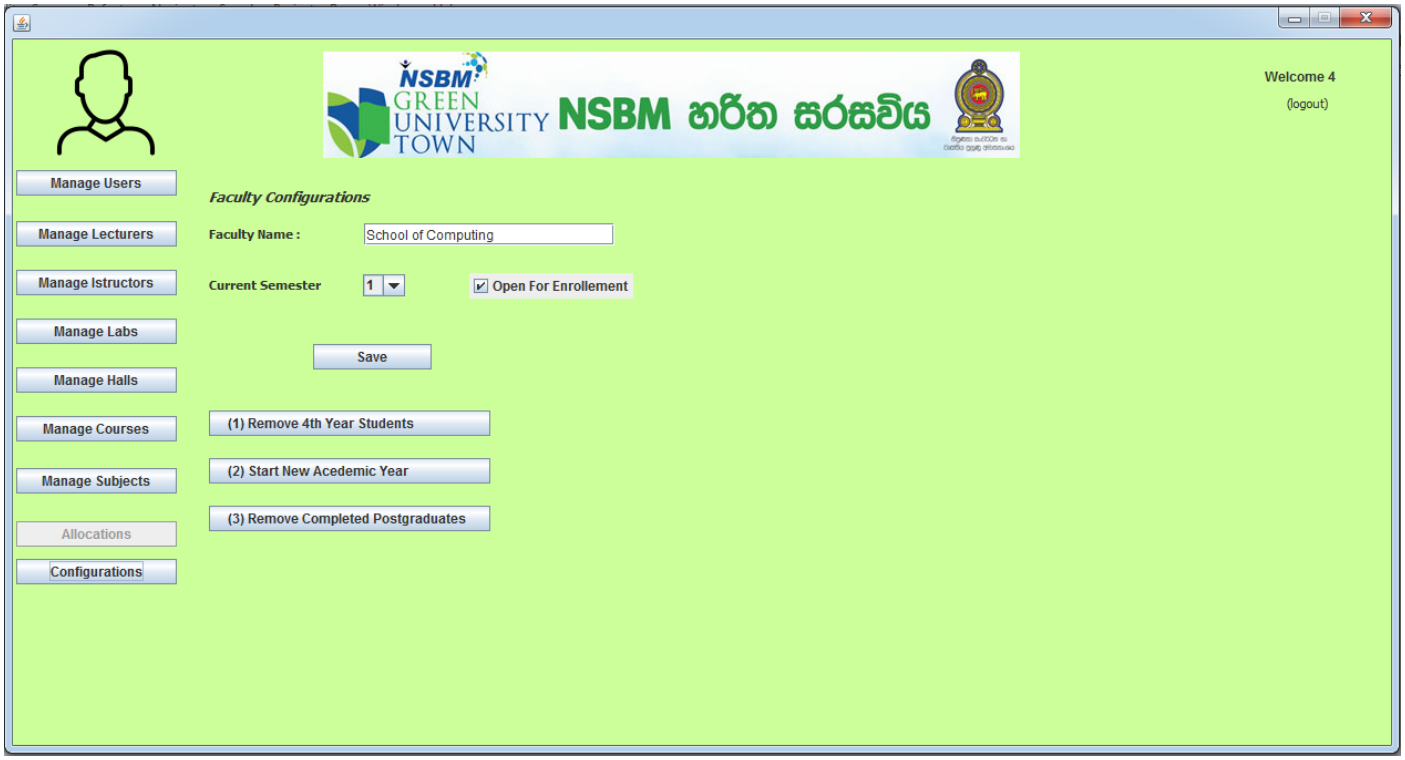
Associates: JPanel_Configuration.java, table: faculty

Initially; Faculty name, Current Semester and Open for enrollment is set using information is database.

When press “Save”; DB is updated using a query in JPanel_Configuration.java and User will be logged out (Login screen appears)

Process of starting a new Academic Year (***should follow orderly***)

1. Remove the 4th Year students
   All the student records in database with CurrentYear=4 will be remove
2. Start new Acedemic Year
   Here all the 3rd year undergraduate student records will be removed, so it is advice to assign relevant 3rd year Students to relevant 4th year degree programmes before performing this action.
   CurrentYear of 2nd year students will set to 3
   CurrentYear of 1st year students will set to 2
3. Remove Competed Postgraduates
   CurrentYear of Competed Postgraduates is set to 3 due to step 2; So Postgraduates with CurrentYear=3 are removed from the database



SharedData.java

The important use of this class is to have **static references** to User, Student, Course, Instructor etc. objects, so that those object created at one JFrame can be accessed using another JFrame.

PrgNo; It is the number given for a program. One program number per one faculty, so that, if necessary all faculties can co-exist in a single database. Initially Faculty-Id (FacId) is set for a relevant program No. in database (Faculty table) by the one who implements the system

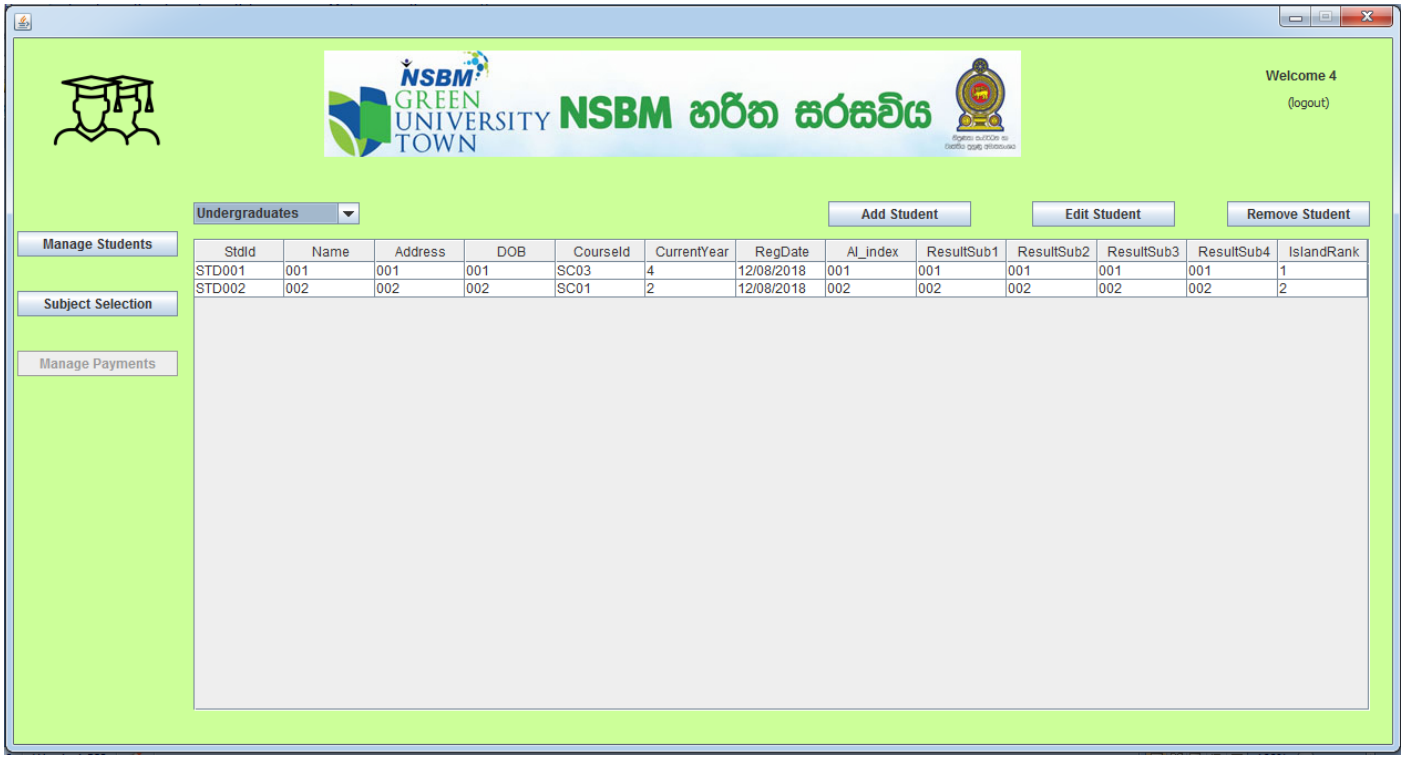
There is a String LoggedUser variable which stores the Username of Current-Logged user so that it could be displayed at the top-right corner ("Welcome *Username*")

The method called **md5** is used to hash the passwords when stored to User table and to compare the entered password with the contents of DB when performing login action.

getFacId(): Used to obtain the faculty ID relevant to a program number.
getDate(): Used to return the current system date in `dd/mm/yyyy` format.
getCurrentSem(): Used to obtain the current semester, 1 or 2.
getCurrentSemIsOpen(): return the availability of current semester.

Student Manager Dashboard

Student Manager Dashboard (StdManDash.java) loads with Manage Students interface, there are 3 functionalities for Student Manager - Manage Students, Subject Selection for Students and Manage Payments (disabled)



When the relevant buttons are clicked, JPanels are loaded to middle_panel of StdManDash

Manage Students

Associates: JPanel_ManageStudent.java, AddEditUnderGradStudent.java, AddEditPostGradStudent.java, Student.java, Undergrad.java, Postgrad.java, tables: student, std_undergrad, std_postgrad

The JPanel_ManageStudent.java is the interface to manage Students, consist of a JTable to retrieve student information from database.
Initially JTable is filled using the ResultSet obtained by retriveStudents() method in Undergrad Class

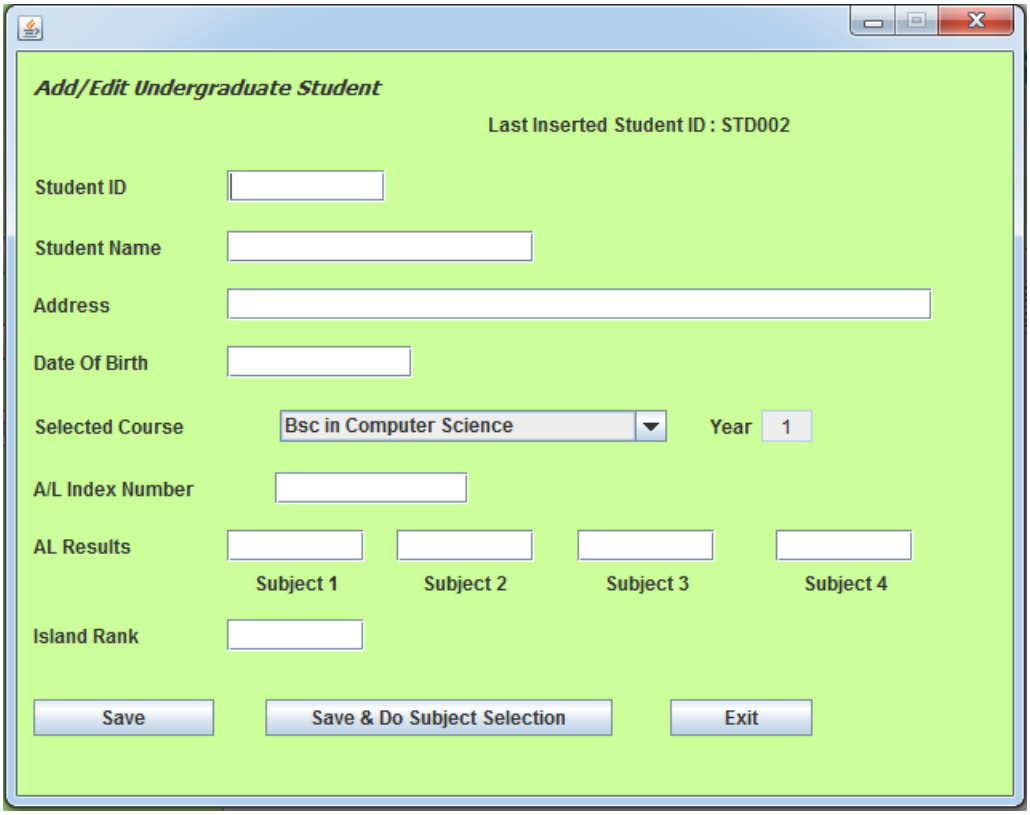
There are 3 buttons to Add, Edit and Remove Students and a comboBox to select either undergraduates or postgraduates.

Manage Students: Undergraduates

When the undergraduates in comboBox is selected, the JTable is filled using the ResultSet obtained by retriveStudents() method in Undergrad Class.

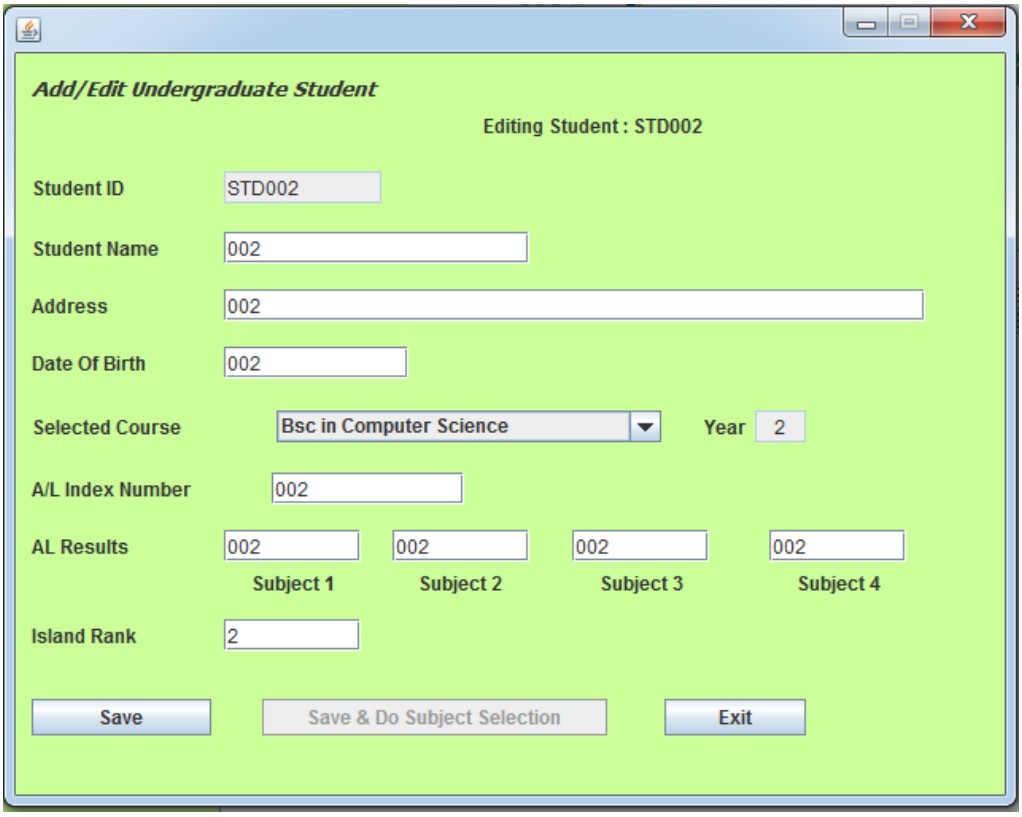
Add Student:

- Disable StdManDash Frame and opens the JFrame AddEditUnderGradStudent
- Last inserted Student ID is displayed using Undergrad.getLastUnderGradStdId() ; The Student Manager (StdMan) is instructed to fill the Student ID field and other fields accordingly.
- On “Save” a new Undergrad object is created by adding filled details, via addData() & UaddData(() methods.
- This user information are added to DB using saveNewData() method
- All the field of AddEditUnderGradStudent are Reset
- StdMan can add more Undergraduates or Click on “Exit”
- StdMan Dash is re-enabled, AddEditUnderGradStudent disappears and JTable is update on “Exit



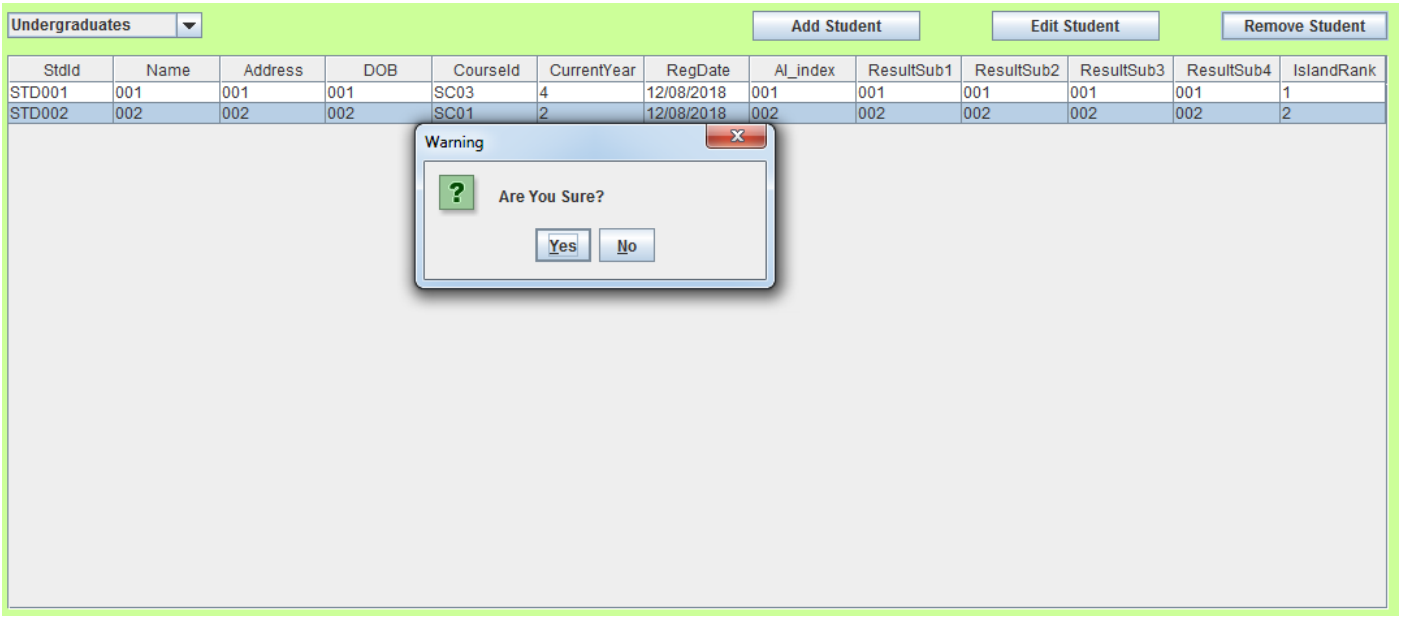
Edit Student:

- A new Undergrad object is created *tempStudent*. (A record has to be selected)
- Set the attributes of *tempStudent* using changeData() and UchangeData() method by information in JTable.
- Assigning *tempStudent* for a static reference "tempUndergrad" in SharedData.java
- Disable StdManDash Frame and opens the JFrame AddEditUnderGradStudent
- The textboxes and comboboxes are filled using the data in SharedData. tempUndergrad
- Student Manager can edit the undergraduate information
- On "Save", the DB is updated using saveData() method.
- StdManDash is re-enabled, AddEditUnderGradStudent disappears and JTable is updated.



Remove:

- A record is selected from the JTable
- Ask for confirmation when “Remove Student” is clicked on.
- If confirm, remove the relevant student using removeStudent() method of Student object



Manage Students: Postgraduates

When the postgraduates in comboBox is selected, the JTable is filled using the ResultSet obtained by retriveStudents() method in Postgrad Class.

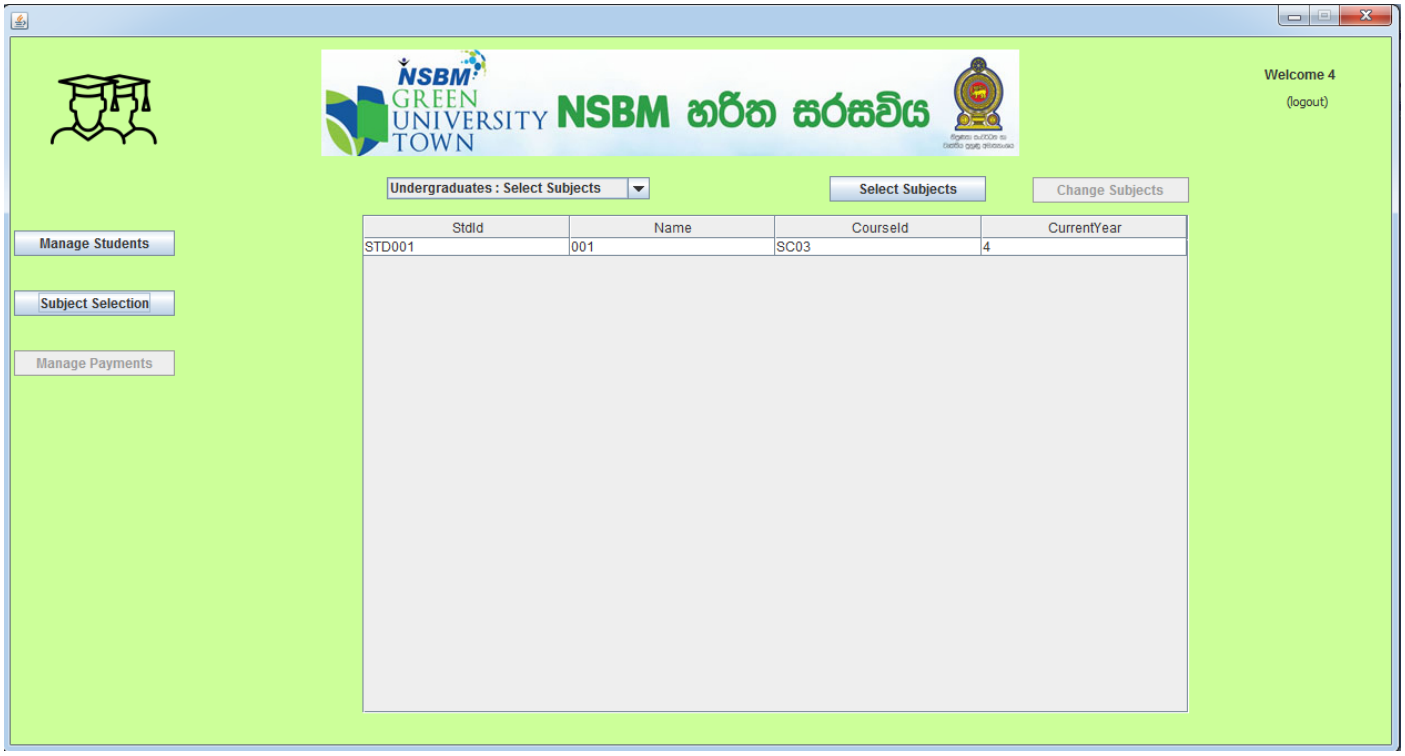
Add Student, Edit Student, Remove Student for a Postgraduate works similarly as with an Undergraduate.

Subject Selection

Associates: SelectStdSub.java, JPanel_ SelectStdSub.java, SelectChangeUnderGradSubject.java, SelectChangePostGradSubject.java, table: Student, std_undergrad, std_postgrad, std_sub, faculty

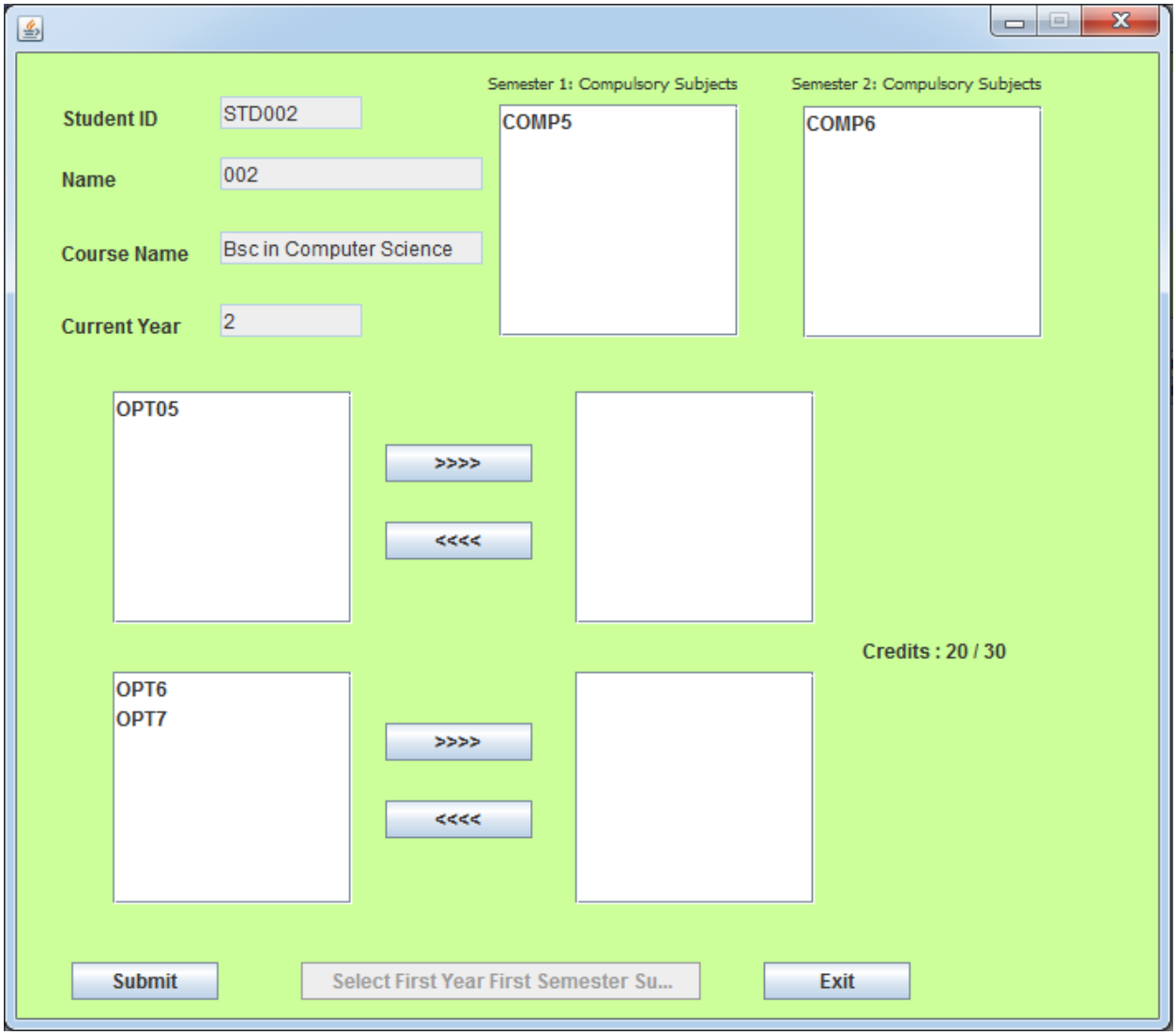
ComboBox => Undergraduate: Select Subjects

The JTable in JPanel_ SelectStdSub.java is filled with the ResultSet obtained by retriveUnderGradSelectSub() method in SelectStdSub Class



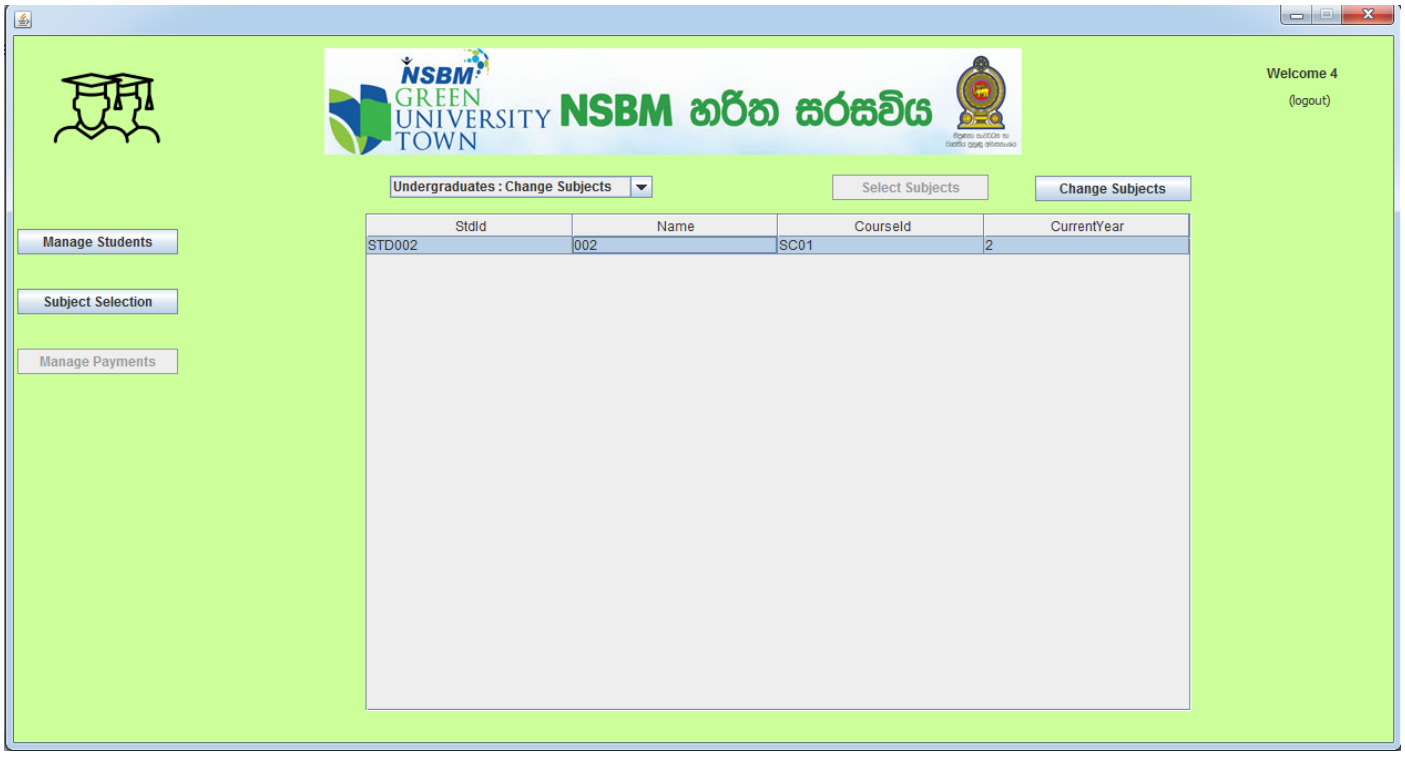
Select Subjects:

- A new SelectStdSub object is created (A record in JTable has to be selected)
- Compulsory Subject list and Optional Subject list of the “object” is set using setMyCompulsorySubjects() and setMyOptionalSubjects() methods of the object.
- The object is assigned to reference SharedData.tempStdSubNew.
- StdManDash is disabled and opens the JFrame SelectChangeUnderGradSubject
- The textboxes, JLists in SelectChangeUnderGradSubject are filled based on the data in SharedData.tempStdSubNew and the configurations in faculty table.
- Student Manager selects optional subjects.
- On “Submit” Selected optional subjects are set using setMyOptionalSubjects() method in tempStdSubNew object and DB is updated using saveNewData() method.
- StdManDash is re-enabled, SelectChangeUnderGradSubject disappears and JTable is updated



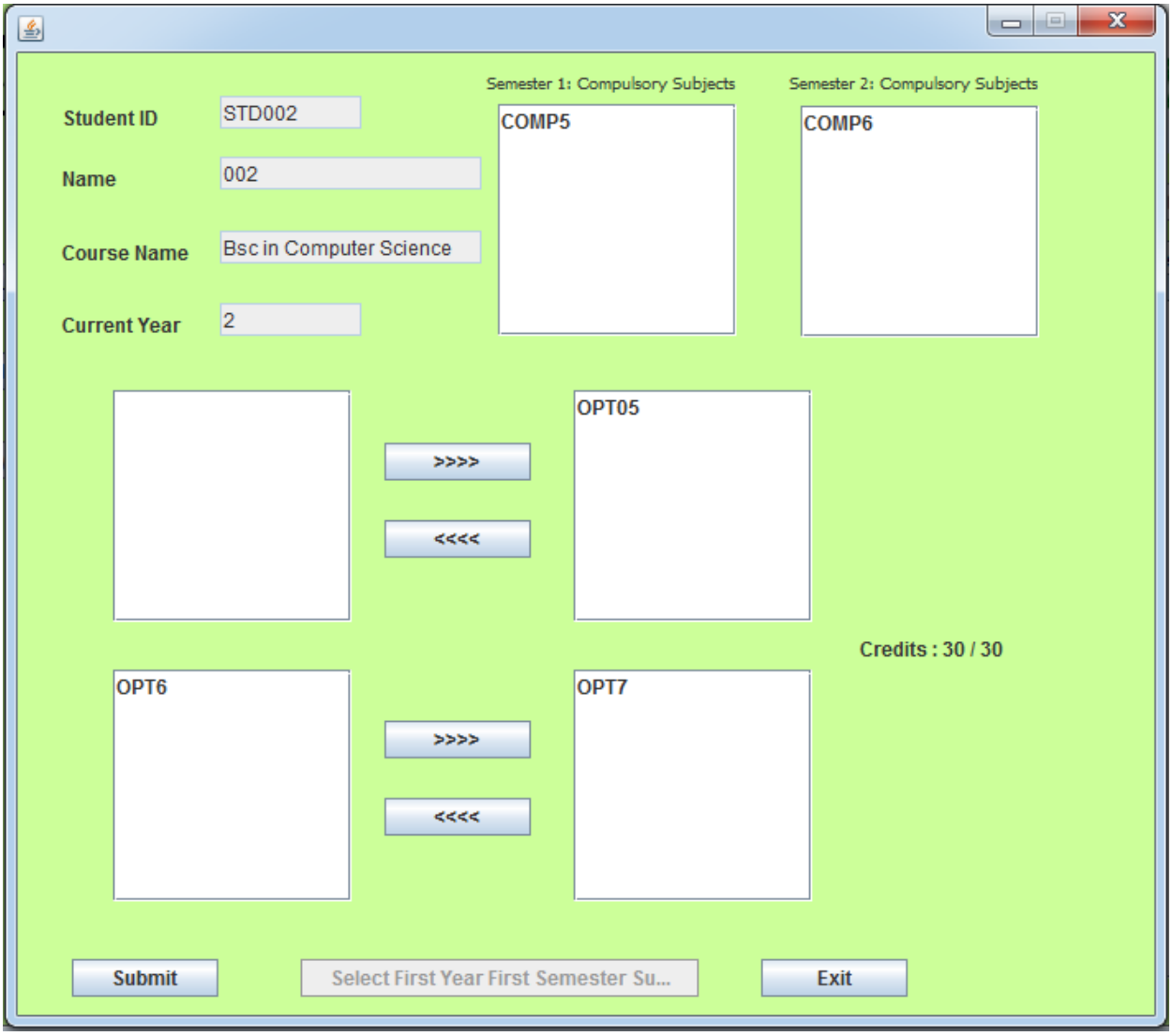
ComboBox => Undergraduate: Change Subjects

The JTable in JPanel_ SelectStdSub.java is filled with the ResultSet obtained by retriveUnderGradChangeSub() method in SelectStdSub Class



Change Subjects:

- A new SelectStdSub object is created (A record in JTable has to be selected)
- Compulsory subject list and Optional Subject list of the "object" is set using setMyCompulsorySubjects() and setMyOptionalSubjects() methods of the object.
- The object is assigned to reference SharedData.tempStdSubChange.
- StdManDash is disabled and opens the JFrame SelectChangeUnderGradSubject
- The textboxes, JList in SelectChangeUnderGradSubject are filled based on the data in SharedData.tempStdSubChange and the configurations in faculty table.
- Already selected optional subjects are filled using getMyOptionalSubjects() method.
- Remaining optional subjects are filled.
- Student Manager changes the selection of optional subjects.
- On "Submit" Selected optional subjects are set using setMyOptionalSubjects() method in tempStdSubChange object and DB is updated using saveData() method.
- StdManDash is re-enabled, SelectChangeUnderGradSubject disappears and JTable is updated



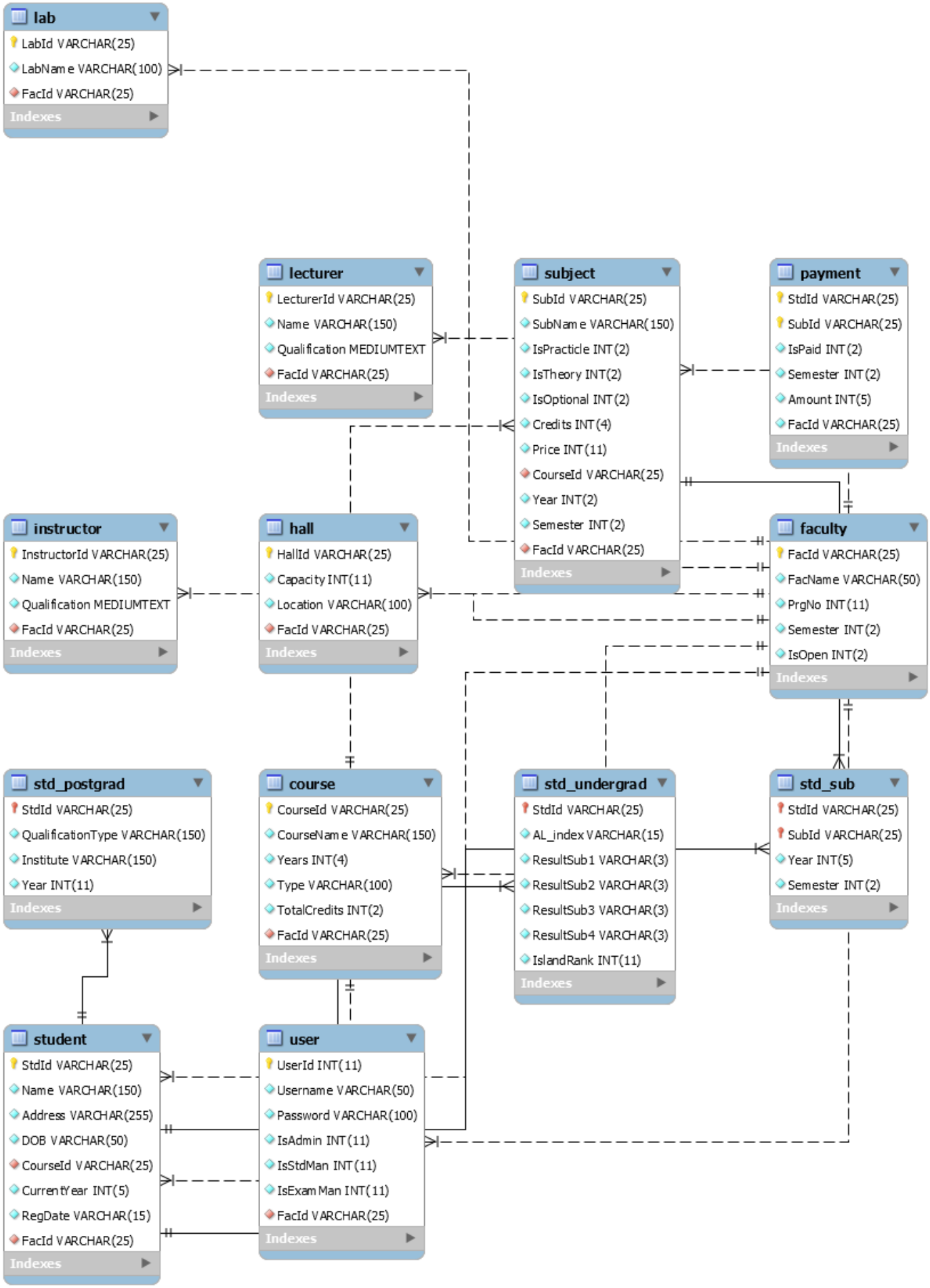
ComboBox => Postgraduate: Select Subjects

The process is similar to that of Undergraduates

ComboBox => Postgraduate: Change Subjects

The process is similar to that of Undergraduates

ER Diagram


lab
LabId VARCHAR(25)
LabName VARCHAR(100)
FacId VARCHAR(25)
Indexes
lecturer
LecturerId VARCHAR(25)
Name VARCHAR(150)
Qualification MEDIUMTEXT
FacId VARCHAR(25)
Indexes
subject
SubId VARCHAR(25)
SubName VARCHAR(150)
IsPracticle INT(2)
IsTheory INT(2)
IsOptional INT(2)
Credits INT(4)
Price INT(11)
CourseId VARCHAR(25)
Year INT(2)
Semester INT(2)
FacId VARCHAR(25)
Indexes
payment
StdId VARCHAR(25)
SubId VARCHAR(25)
IsPaid INT(2)
Semester INT(2)
Amount INT(5)
FacId VARCHAR(25)
Indexes
instructor
InstructorId VARCHAR(25)
Name VARCHAR(150)
Qualification MEDIUMTEXT
FacId VARCHAR(25)
Indexes
hall
HallId VARCHAR(25)
Capacity INT(11)
Location VARCHAR(100)
FacId VARCHAR(25)
Indexes
faculty
FacId VARCHAR(25)
FacName VARCHAR(50)
PrgNo INT(11)
Semester INT(2)
IsOpen INT(2)
Indexes
std_postgrad
StdId VARCHAR(25)
QualificationType VARCHAR(150)
Institute VARCHAR(150)
Year INT(11)
Indexes
course
CourseId VARCHAR(25)
CourseName VARCHAR(150)
Years INT(4)
Type VARCHAR(100)
TotalCredits INT(2)
FacId VARCHAR(25)
Indexes
std_undergrad
StdId VARCHAR(25)
AL_index VARCHAR(15)
ResultSub1 VARCHAR(3)
ResultSub2 VARCHAR(3)
ResultSub3 VARCHAR(3)
ResultSub4 VARCHAR(3)
IslandRank INT(11)
Indexes
std_sub
StdId VARCHAR(25)
SubId VARCHAR(25)
Year INT(5)
Semester INT(2)
Indexes
student
StdId VARCHAR(25)
Name VARCHAR(150)
Address VARCHAR(255)
DOB VARCHAR(50)
CourseId VARCHAR(25)
CurrentYear INT(5)
RegDate VARCHAR(15)
FacId VARCHAR(25)
Indexes
user
UserId INT(11)
Username VARCHAR(50)
Password VARCHAR(100)
IsAdmin INT(11)
IsStdMan INT(11)
IsExamMan INT(11)
FacId VARCHAR(25)
Indexes