**FINAL REPORT**

TOPIC: SCHOOL MANAGEMENT SYSTEM

**MEMBERS:**

|  |  |  |
| --- | --- | --- |
| **Name** | **Student ID** | **Role** |
| Lê Thanh | ITITIU17070 | Member |
| Trần Minh Ngọc | ITITIU17068 | Member |
| Trang Thanh Mai Duyên | ITITUN17007 | Leader |

TABLE OF CONTENTS

[1. INTRODUCTION 2](#_Toc41149680)

[**a. Project 2**](#_Toc41149681)

[**b. Tasks 2**](#_Toc41149682)

[2. ENTITY-RELATIONSHIP MODELING 3](#_Toc41149683)

[**a. Entity and attribute 4**](#_Toc41149684)

[**b. Schema of entities 5**](#_Toc41149685)

[3. RELATIONAL MODEL 7](#_Toc41149686)

[4. DATABASE DIAGRAM 8](#_Toc41149687)

[5. DATABASE 8](#_Toc41149688)

[6. GUI 17](#_Toc41149689)

[**a. Student/Staff login 19**](#_Toc41149690)

[**b. Admin 37**](#_Toc41149691)

[7. Conclusions 78](#_Toc41149692)

[REFERENCES 79](#_Toc41149693)

1. INTRODUCTION

This project is created for organizing the educating system as well as supporting the students on their studying process.

* 1. **Project**

Topic: School Management System (topic 20)

* Github: <https://github.com/neonpotts/School-Management-PDM-project->
  1. **Tasks**
* We met and discussed a few things and decided to divide the project into the following tasks:

• First, we discussed together and came up with ideas about ERD

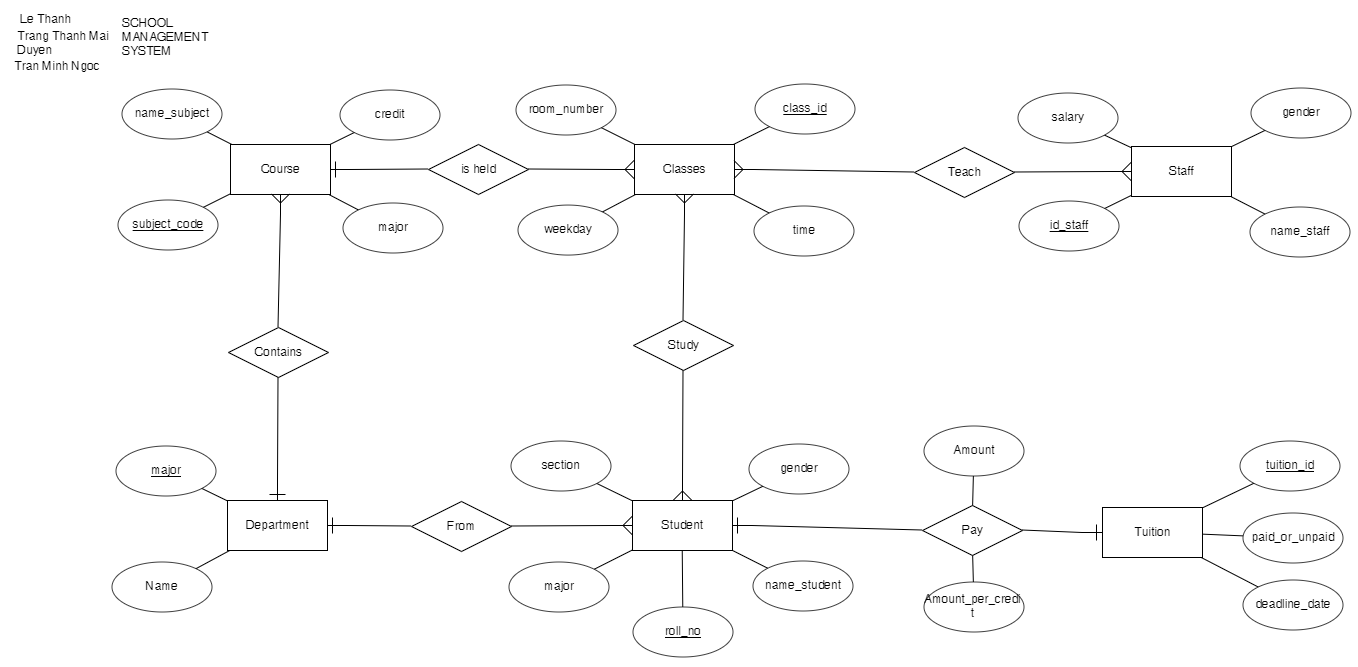
• Next, we base on ERD and draw Relational Model

• Then create the table and insert the information into the tables in the database

• Finally, we designed the interface to retrieve data from the database and display it in the most user-friendly way

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Duyên** | **Ngọc** | **Thanh** | **Complete(%)** |
| Create table | X |  |  | 100% (Duyen) |
| Insert data | X |  | X | 60% (Duyen) - 40% (Thanh) |
| Draw ERD |  |  | X | 100% (Thah) |
| Draw Relational Model |  |  | X | 100% (Thanh) |
| Write query retrieving data | X |  |  | 100% (Duyen) |
| Design GUI |  | X |  | 100% (Ngoc) |
| Code the GUI function |  | X |  | 100% (Ngoc) |
| Write proposal | X |  |  | 100% (Duyen) |
| Write report | x |  | X | 20% (Duyen) – 80% (Thanh) |
| Check grammar/edit report | X |  |  | 100% (Duyen) |
| Fix bug | X | X |  | 10% (Duyen) – 90% (Ngoc) |

1. **ENTITY-RELATIONSHIP MODELING**

****

1. **Entity and attribute**

* **Course:**

• credit

• subject\_code(PK)

• name\_subject

• major

* **Classes:**

• weekday

• room\_number

• time

• class\_id (PK)

* **Student:**

• roll\_no (PK)

• section

• gender

• major

• name\_student

* **Staff:**

• id\_staff (PK)

• name\_staff

• gender

• salary

* **Tuition:**

• tuition\_id (PK)

• paid\_or\_unpaid

• deadline\_date

* **Department:**

• major (PK)

• Name

1. **Schema of entities**
   * Entity course is represented for the subject so it has attributes: subject\_code(code of subject), name\_subject (name of subject),major and credit
   * Entity Classes is represented for the classes of each course (1 course can have many classes in different time or room in the schedule, and it cannot be duplicated) so it has attribute: weekday (day of week), room\_number (room number), time
   * Entity Student is represented for student and it has attributes: roll\_no(id of student), section, gender(male or female), name\_student (name of student), major
   * Entity Staff is represented for teacher and it have attribute: id\_staff (id of staff), name\_staff (name of staff), gender (male or female), salary (salary of staff)
   * Entity Tuition is represented for the tuition fee that student must pay for school to and it has attributes: paid\_or\_unpaid(the status of the tuition fee to distinguish that student had paid and not), deadline\_date(the deadline date for student to pay the fee), tuition\_id ( id for the payment)
   * Entity Department has 2 attributes: major and Name (full name of major)
   * Function of the system:

• Get information of staffs, students

• Get the information about the courses

• Get the information about the tuition of students and salary of staffs

• Get the schedule of students

• Get the schedule of teachers

• Get the major and number of major that system has

• Allow admin to handle and edit data in 8 tables (except Account and Department)

• Allow Student and Staff can add and drop courses

• Allow Student and Staff can change the account password

• Allow Student and Staff can update and delete SOME data which they have permission to change

1. **Explanation:**

In relationship, we have those relation between 2 entities:

• “Locate” between Course and Classes (1 to many)

• “Study” between Classes and Student (many to many)

• “Teach” between Classes and Staff (many to many)

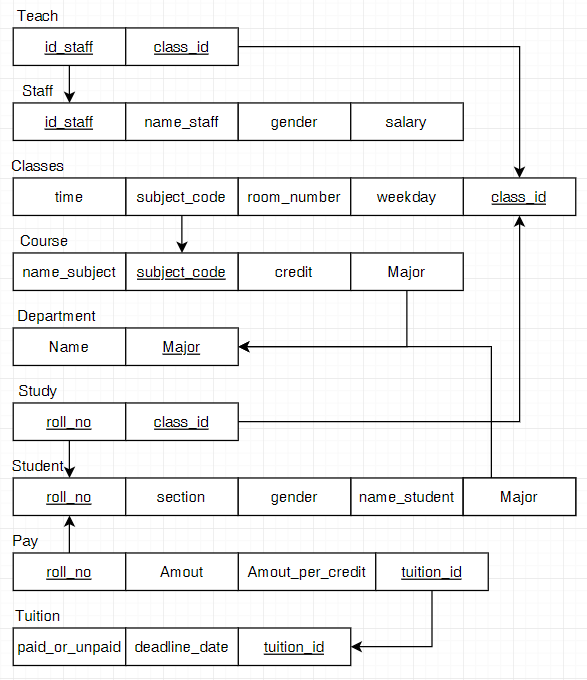
• “Pay” between Tuition and Student (1 to 1)

• “Contains” between Department and Course (1 to many)

• “From” between Department and Student (1 to many)

In the system, a course can be held in many classes, and those classes cannot be duplicated by using primary key for class\_id, and UNIQUE (room\_number,weekday,time). Next, a student is studying in classes and it has 5 main attributes ,and he/she can attend to many classes and a class can be attended by many students. Also, the students’ major and the courses’ major has relation references from the major of Department, which has major and name inside. Obviously, each student has his/her own major and a major can have many students in. Besides that, Staff has 4 basic attributes: id\_staff,name\_staff,gender and salary. In particularly, staff in our system is represented for teachers, he/she is assigned to teach classes which are assigned by admin. And, a staff can teach many classes, but a class can be taught by many teachers. Furthermore, tuition has 3 basic attributes: tuition\_id, paid\_or\_unpaid, deadline\_date. The deadline\_date, and paid\_or\_unpaid, which shows the payment status of a student (use Boolean in Java, CHECK constraint in SQL), are for displaying the student data who had paid or unpaid the tuition fee up to date. This will be helpful on the function that the student cannot see his/her timetable, if he/she had not paid his/her tuition fee before the deadline date. Moreover, the student and tuition are connected with each other (each student has 1 tuition) by Pay that has not only 2 foreign keys from Tuition(tuition\_id) and Student(roll\_no), but also Amount\_per\_credit, price per credit, and Amount, the student’s tuition fee.

1. **RELATIONAL MODEL**

****

* + Our goal when creating more tables (Study, Teach, Pay) but not to change the amount of information compared to ERD is to make the data more visible and easier to control

**•** Course(subject\_codename\_subject, credit, major(FK))

**•** Classes(time, subject\_code(FK), room\_number, weekday, class\_id)

**•** Staff(id\_staff,name\_staff, gender, salary)

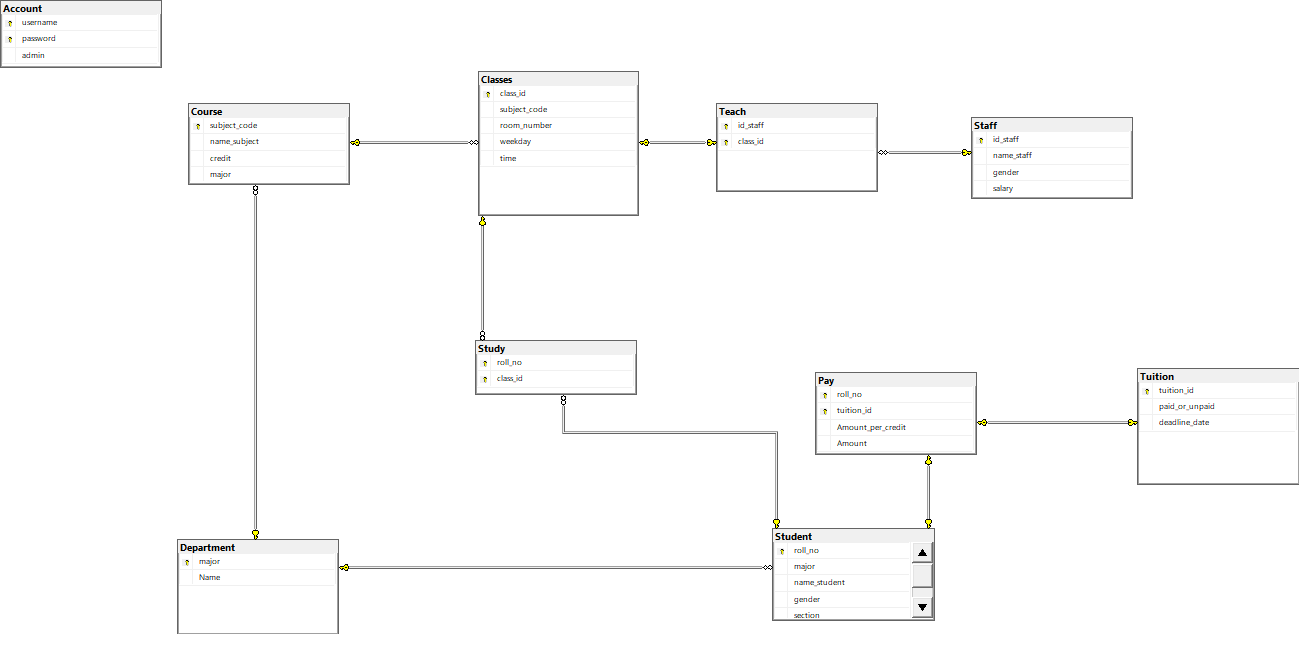
**•** Teach([id\_staff(FK),class\_id(FK)])

**•** Study([roll\_no(FK),class\_id(FK)])

**•** Student(roll\_no, section, gender, name\_student, major(FK))

**•** Tuition(fee, paid\_or\_unpaid, tuition\_id)

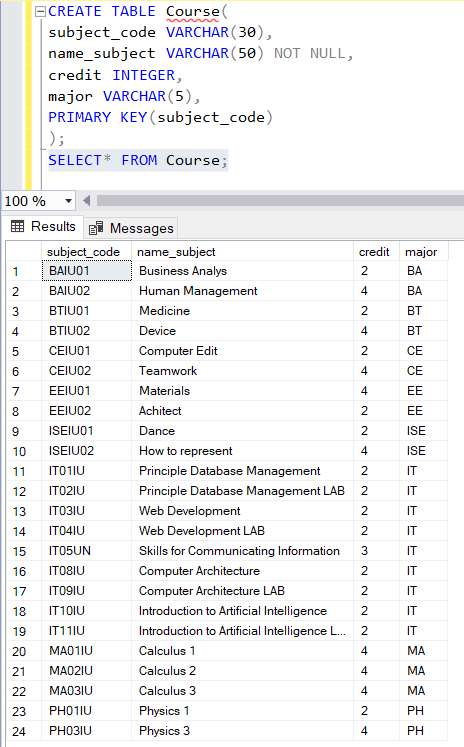
**•** Department(major, Name)

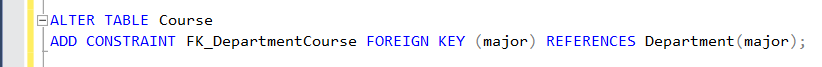
1. **DATABASE DIAGRAM**
2. **DATABASE**

**Table**

* Create new database name “School\_Management” in SQL Server
* Create tables and insert data:

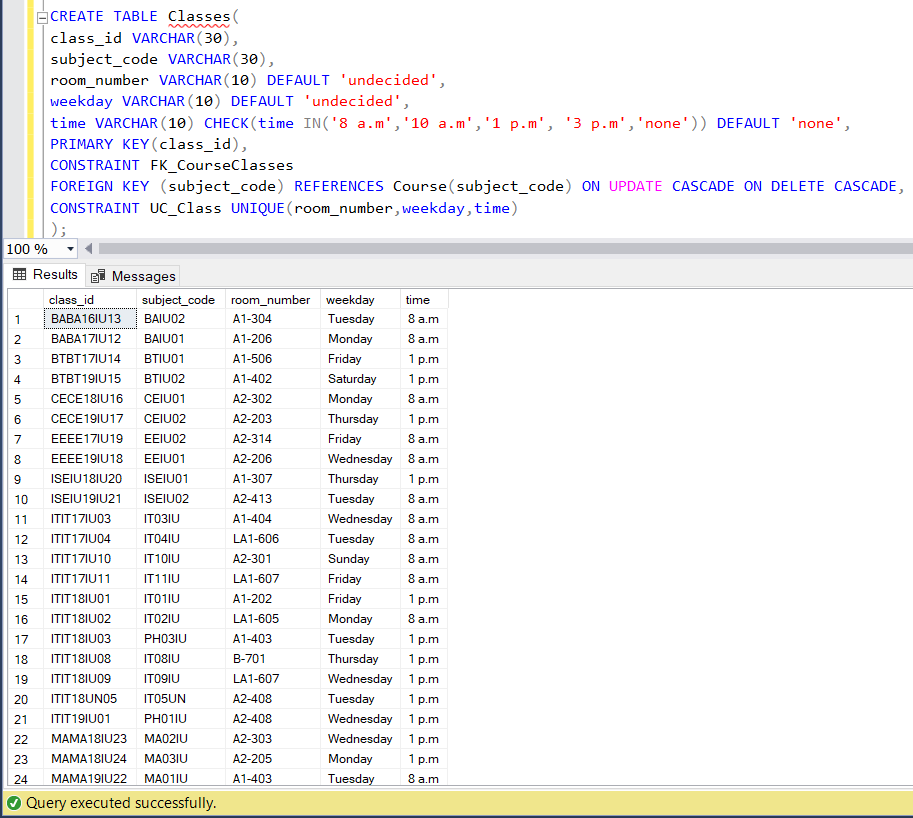
**• Course**





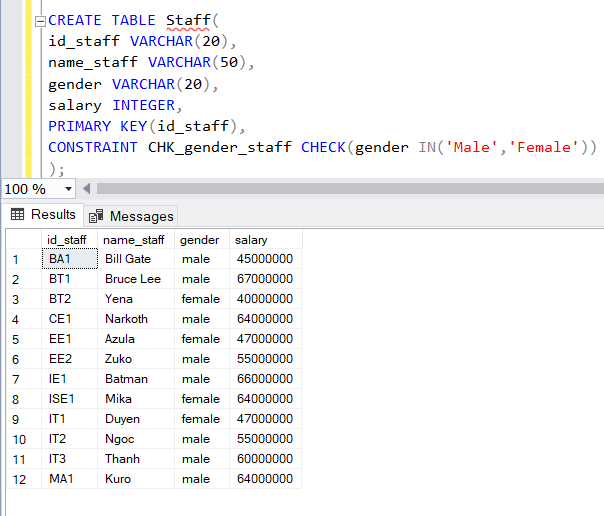
* Subject\_code and name\_subject represented for the id and the name of course in type varchar with maximum size, credit is in integer type
* Set the primary key for subject\_code (cannot be duplicated), and it can be used as foreign key in another table that has relationship with this table
* Attribute major is foreign key that get from table “Department”

**• Classes**



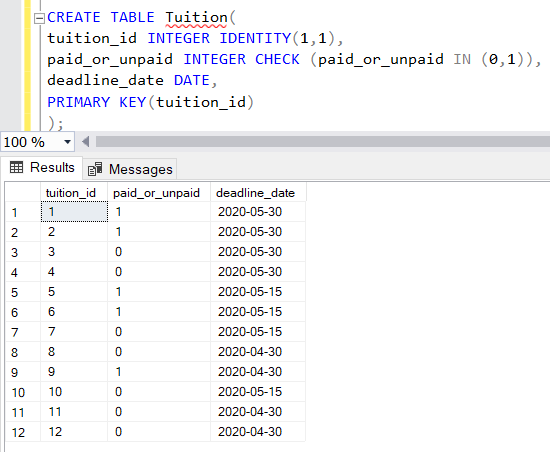
* In this table, the subject\_code has the foreign key from subject\_code of table “Course”
* Room\_number and weekday are type varchar and set if the user skip input this attribute, it will automatically display “undecided”
* In attribute times, it is static in 4 particular times (8a.m, 10a.m,1p.m,3p.m) and if we don’t enter the value in it, it will auto display “none”
* Set attribute weekday and room\_number are unique which make the class cannot be duplicated in the same day and room
* Set the primary key for attribute class\_id and another table can use it as foreign key if it has relationship with this table

•  **Staff**



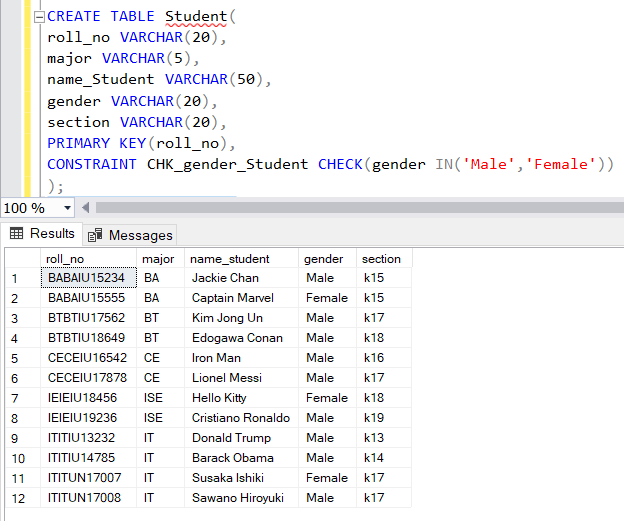
* Id\_staff, name\_staff and gender are in type varchar, and attribute salary in is integer
* Set the primary key for id\_staff and other tables can use it as foreign key if it has a relationship with “Staff”
* And the gender must be “Male” or “Female”

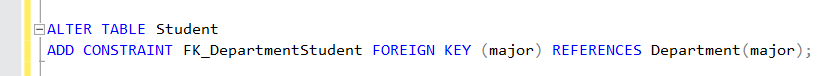
• **Tuition**



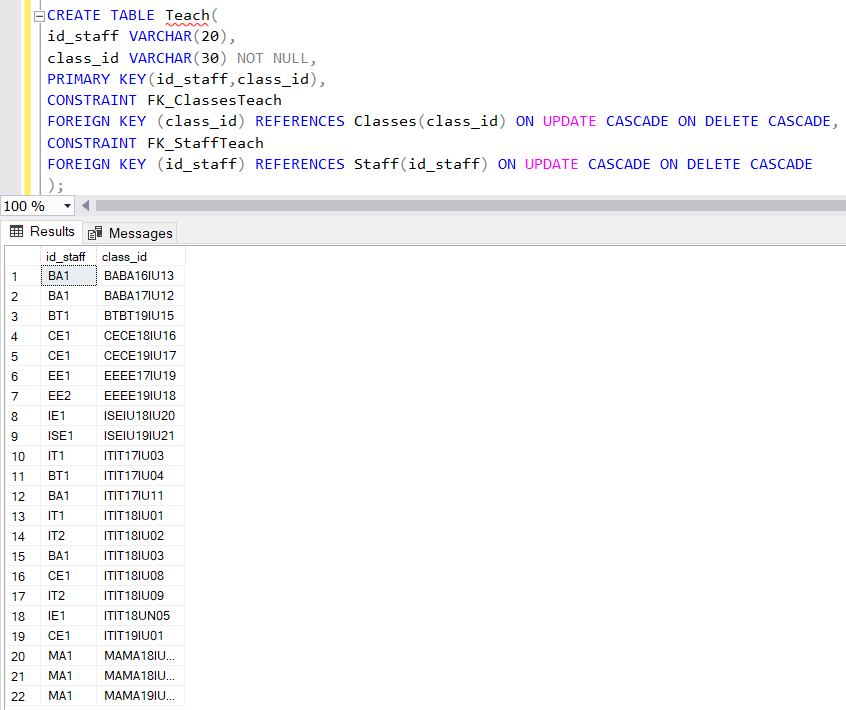
* Tuition\_id is declared in type varchar,both fee and paid\_or\_unpaid in integer and set if it is paid,it will display “1”;otherwise, unpaid “0”
* Set the primary key for tuition\_id and it can be used as foreign key in another table which has a relationship with “Tuition”

• **Student**



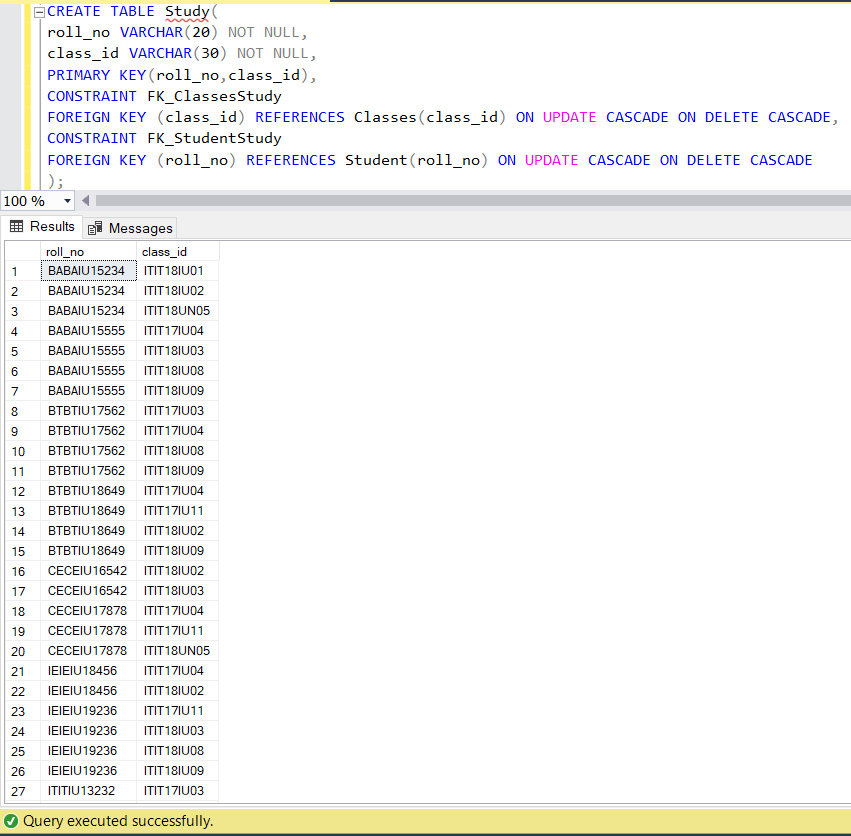


* Roll\_no, section, name\_student, gender are varchar datatypes with different lengths
* Set the primary key for roll\_no and it can be used as foreign key in another table which has a relationship with “Student”
* And we set gender must input as “Male” or Female”
* Attribute tuition\_id and major are foreign key references from tuititon\_id of table “Tuition” and major of “Department”, respectively
* • **Teach**



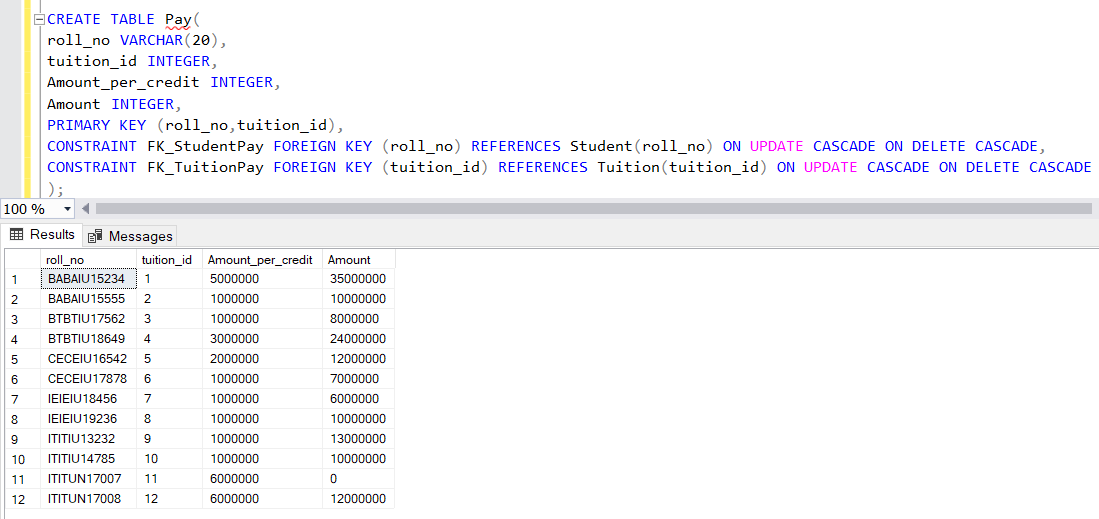
* Attribute class\_id is foreign key from class\_id of table “Classes”
* Attribute id\_staff is foreign key from id\_staff of table “Staff”

**• Study**



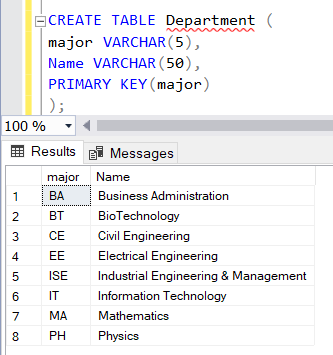
* Attribute roll\_no is subset of roll\_no from table “Student”
* Attribute class\_id is subset of class\_id from table “Classes”

• **Pay**

****

* Attribute roll\_no is subset of roll\_no from table “Student”
* Attribute tuition\_id is subset of tuition\_id from table from table “Tuition”
* We declared Amount\_per\_credit, which is the price of one credit, and Amount, tuition fee, in type integer
* roll\_no,tuition\_id is the Primary key

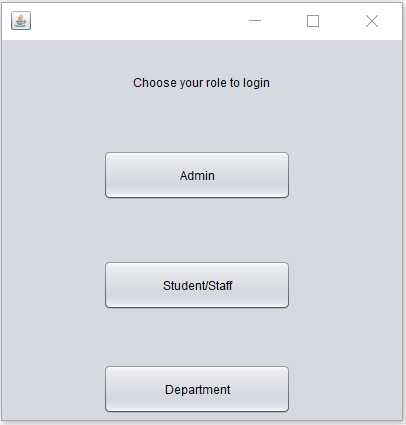
• **Department**

****

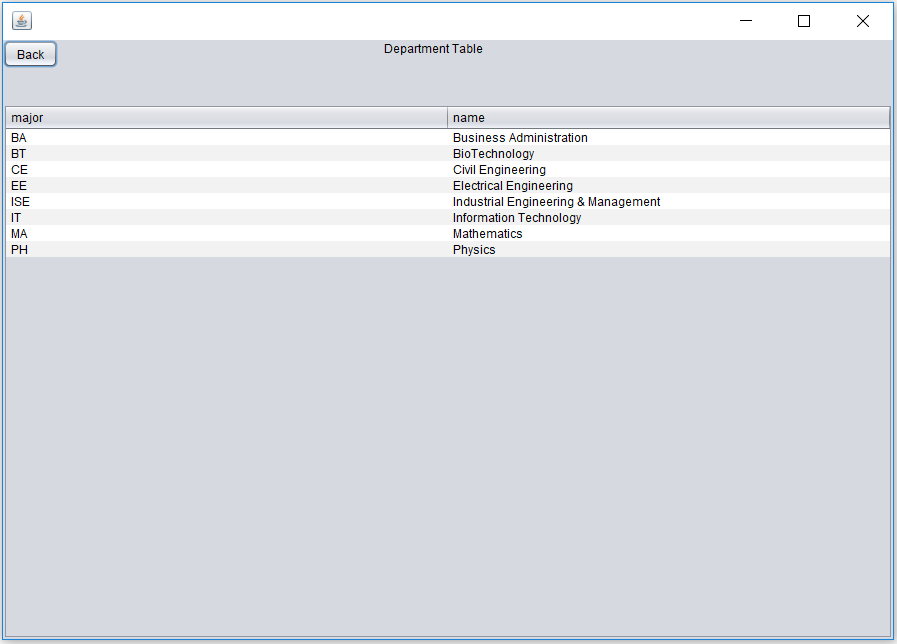
* Attribute major, Name are in type varchar
* Set the primary key for major and it can be used as foreign key in other tables which is relation to “Department”

1. **GUI**

* First, the user can access to the system, they will be required to log in with 2 roles (student/teacher or admin), (run file ChooseRole.java)

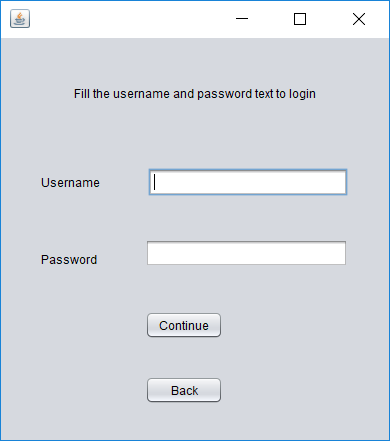


* Starting with button “Department”. It is not required to log in and it shows all the majors that the school has

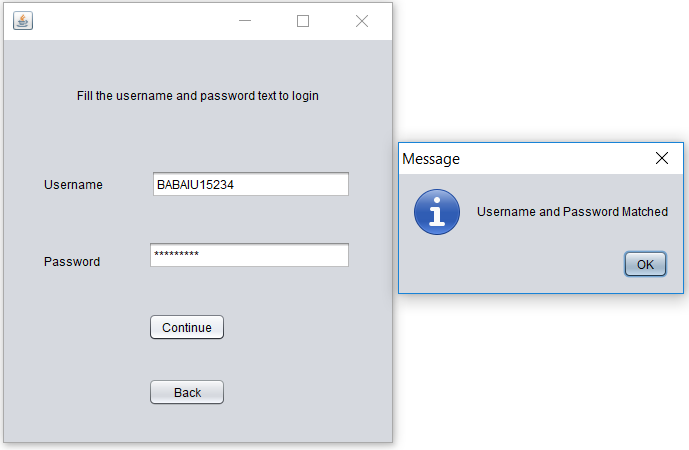
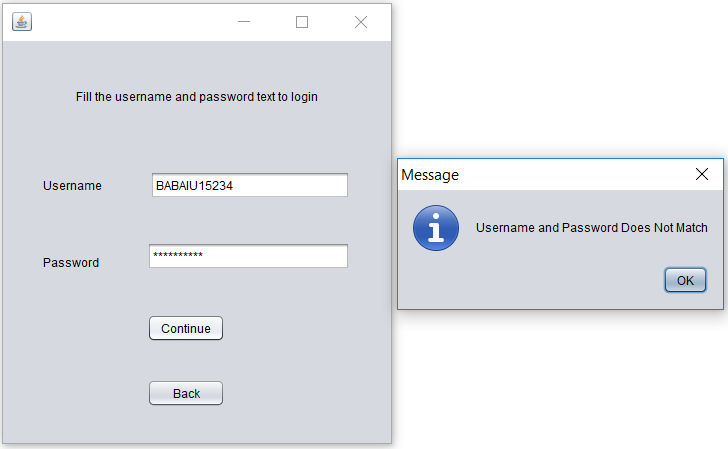


1. **Student/Staff login**

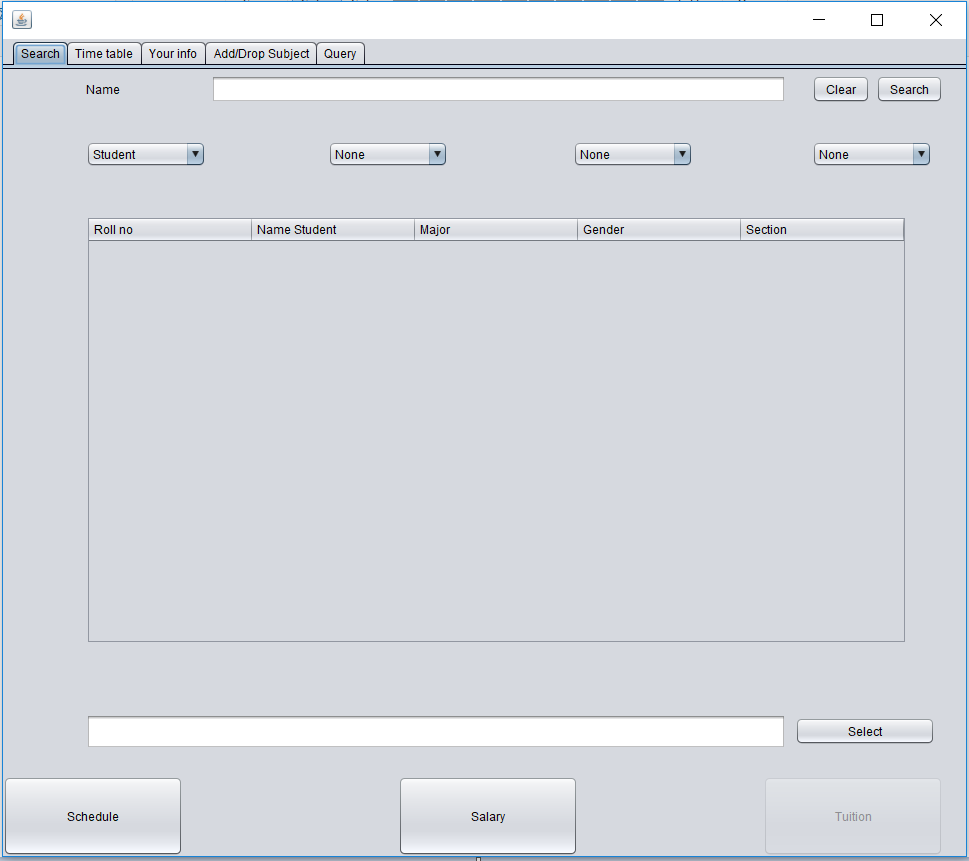
* Next, clicking the Student/Staff button, it will display the login form, where the user has to input username and password to access into the system



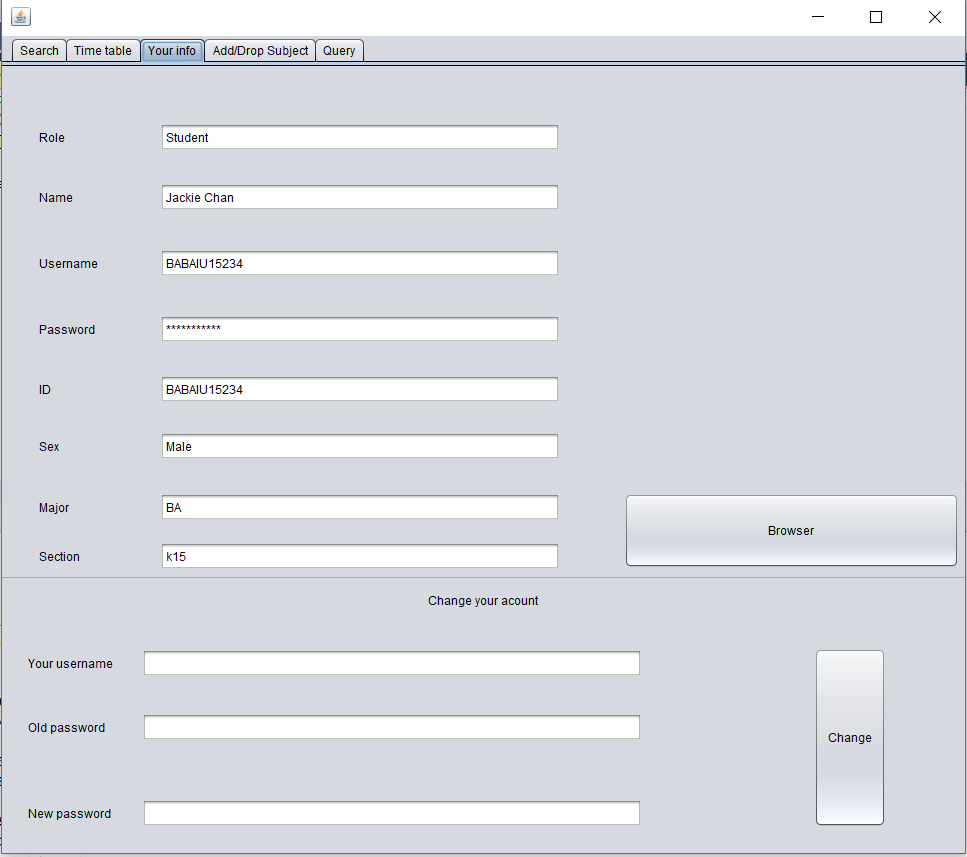
* Use the accounts that are stored in database (table “Account”) in order to login into the system. There are 2 possible cases that can be happened: login successfully and login failed



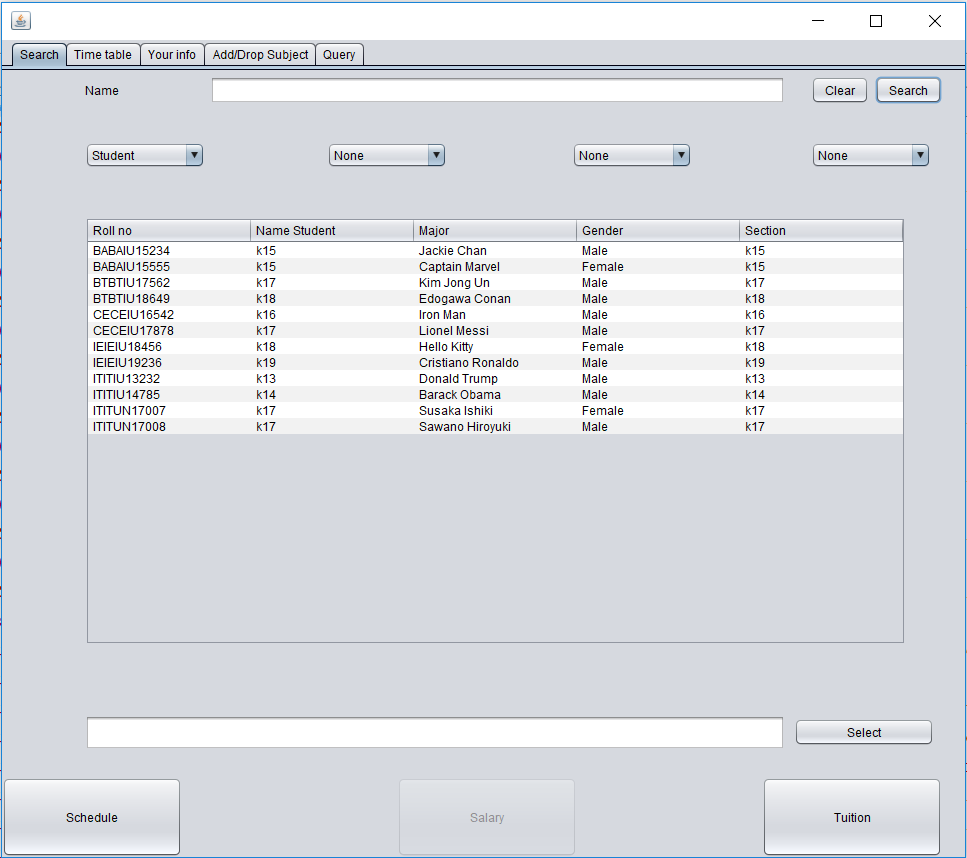
* When the user access successfully, this table will display. In here, the interface has 5 tabs: Search, Time table, Your info, Add/Drop Subject, Query



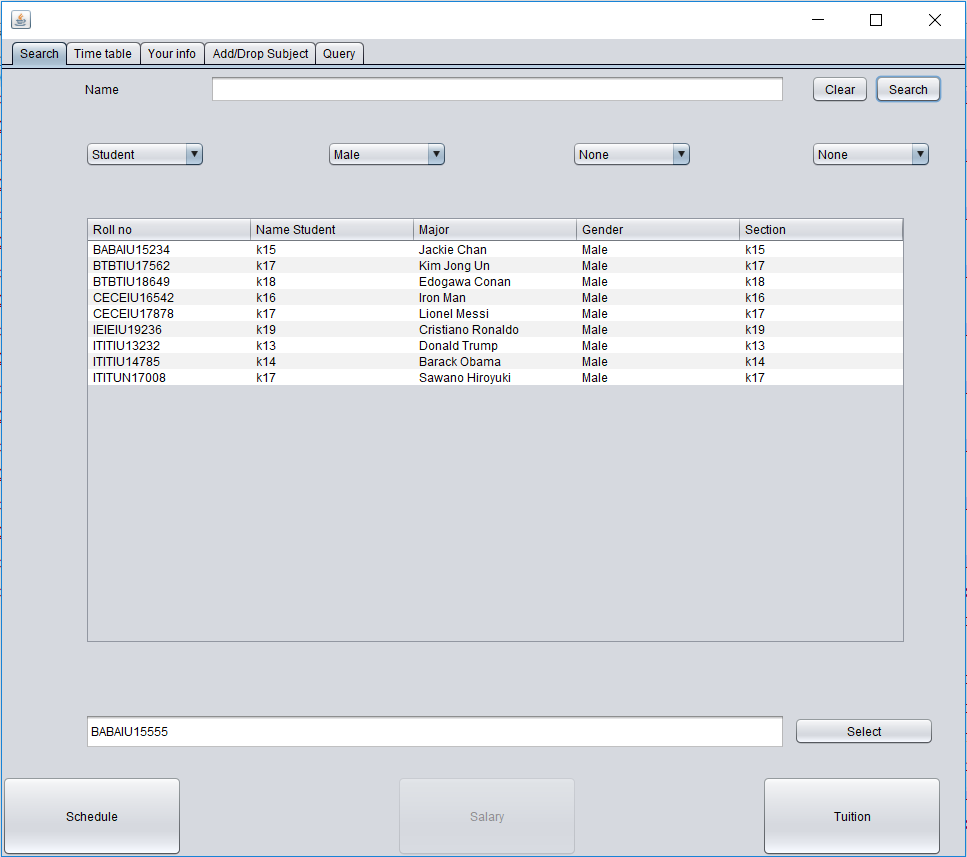
* Access “Your info” tab to see the information of student who just logged in



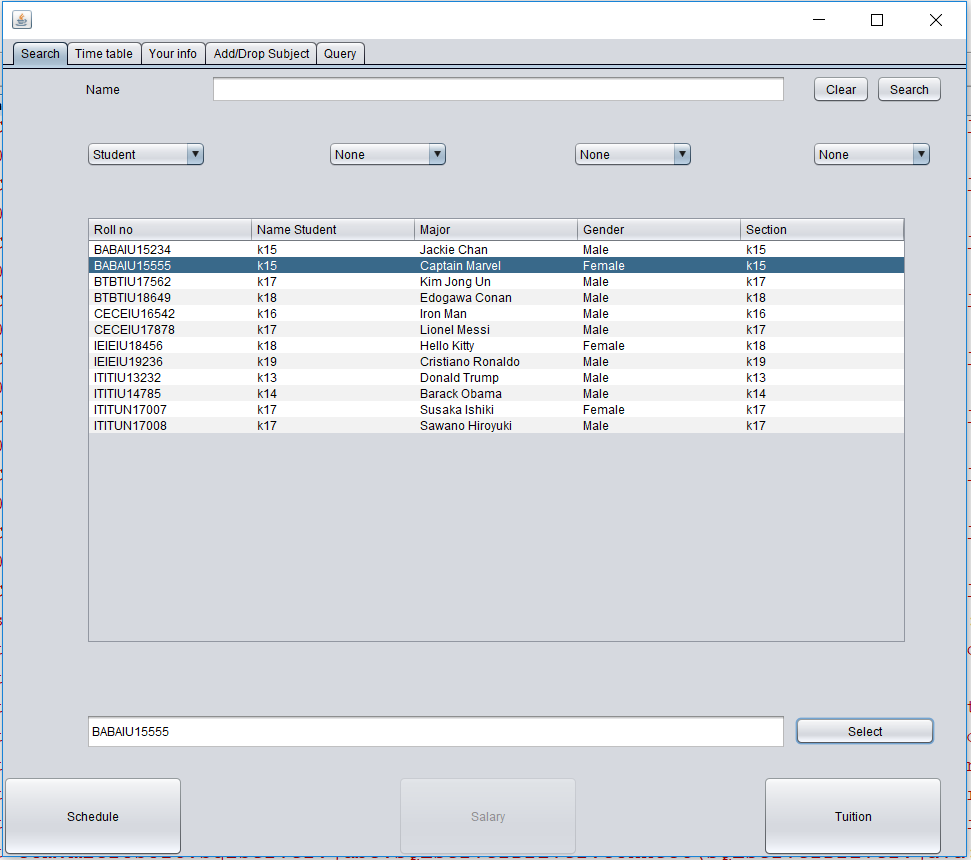
* Moving back to the Search type, there are the textField where to search of filter the data we want to display in the table below. When the user want to search for students’ information of others, he/she can input the keyword of the name into the textField , then click search button. It will search for the data with the keyword name he/she just input. If the textField is empty, it will automatically display all students in table.

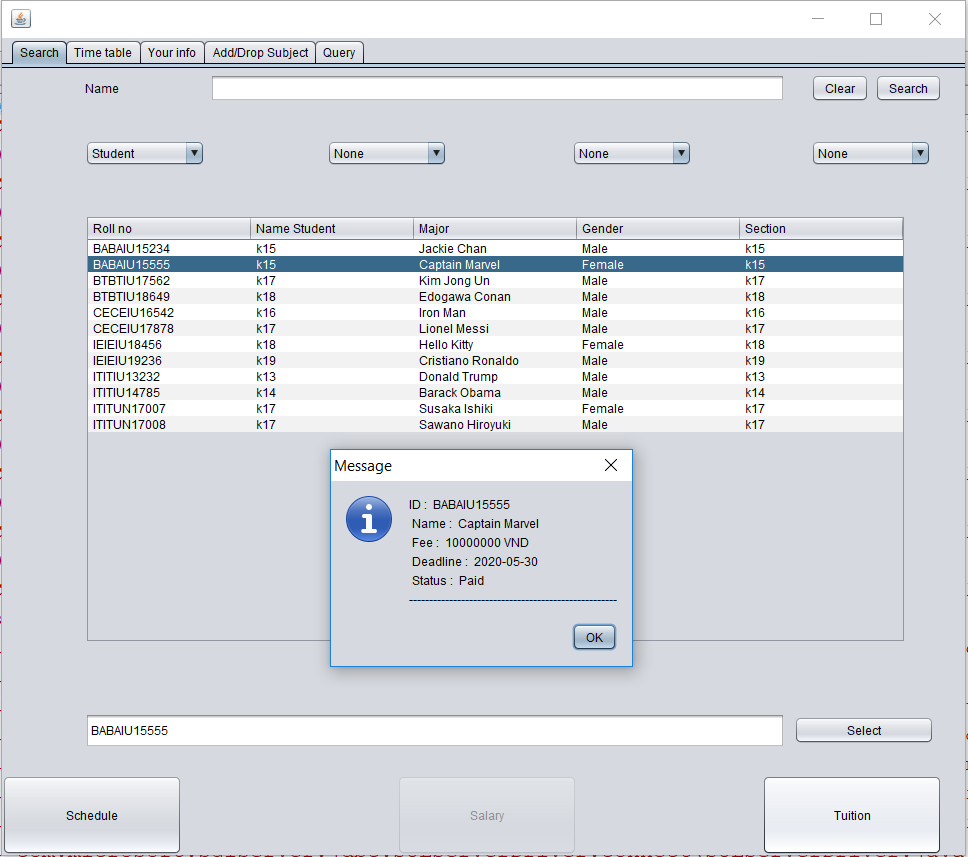


* 4 checkboxes below the name textfield represent for 4 attributes of a student that the user want to search (the system can know at least 1 and max 4 to search, example the user want to search all student that are male)

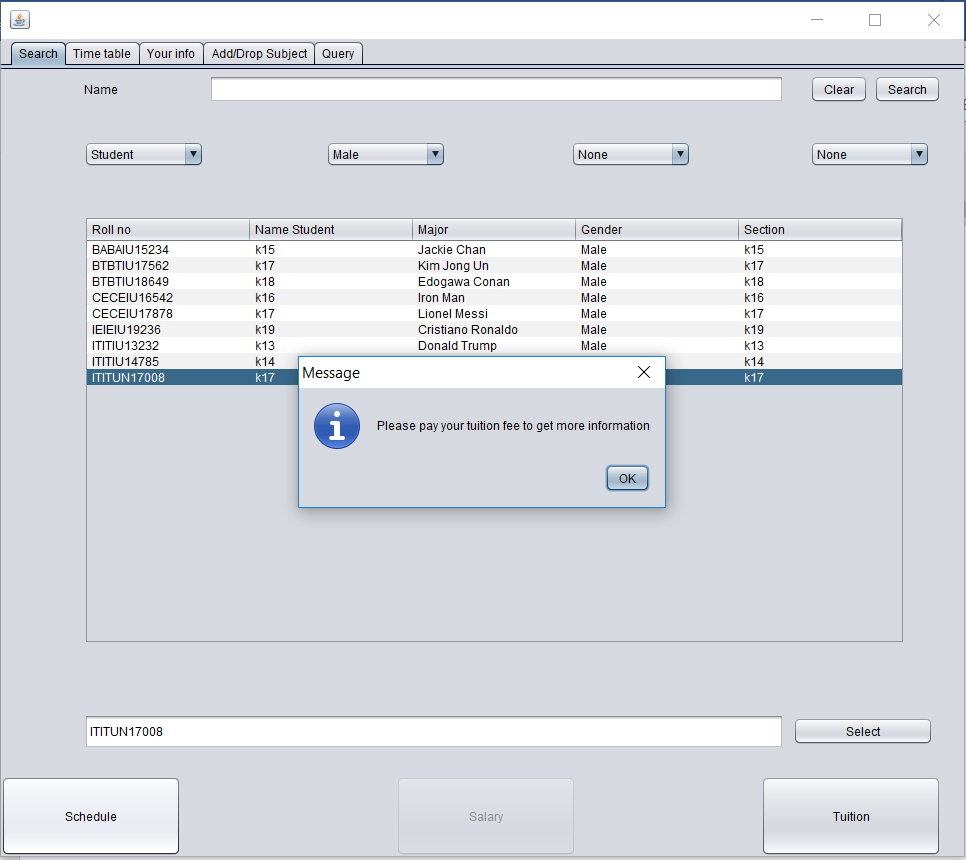


* Select button will help the system to lock the student that the user wants to see this student’s schedule (if paid) or tuition fee
* After clicking select button, if the user wants to see the tuition of his/her, he/she just click the “Tuition” button

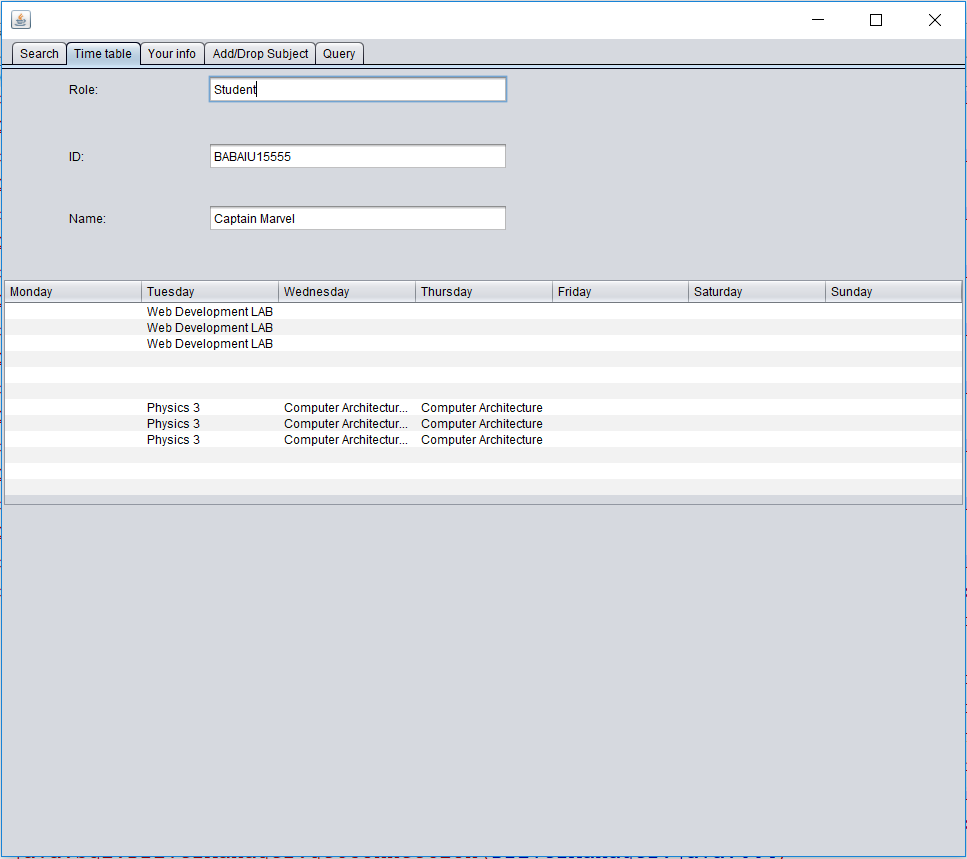




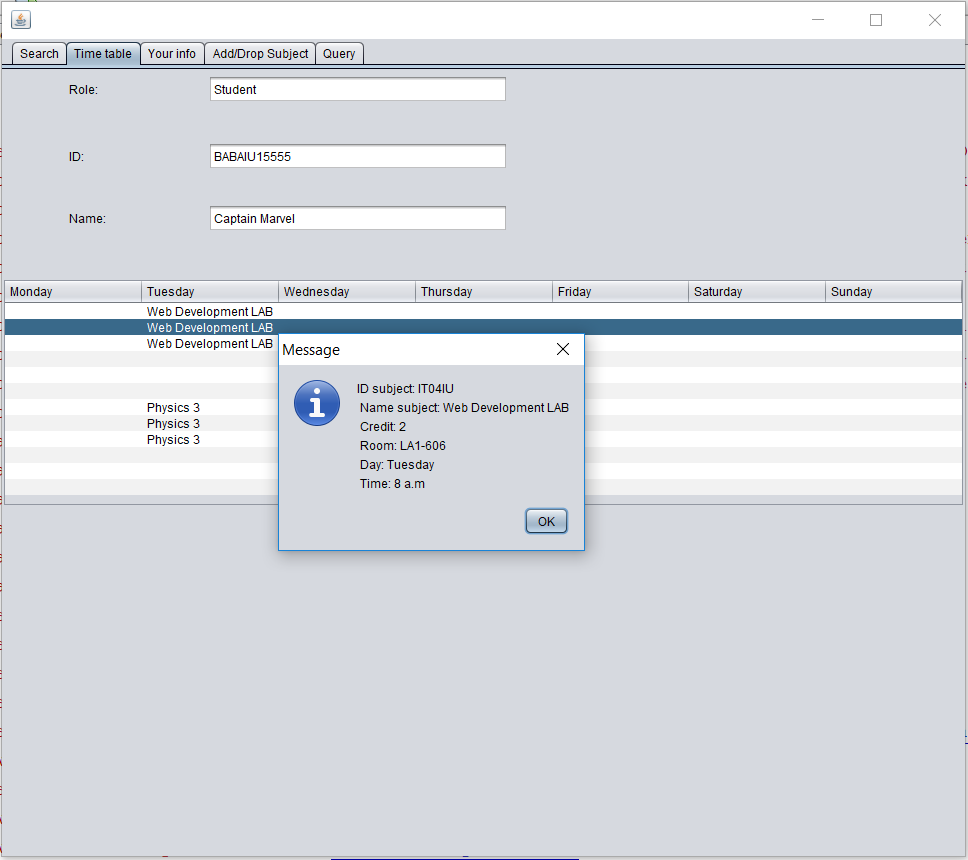
* If user wants to see the schedule of this student, user must check if his/her has already paid or not, if not the system don’t allow user to see the schedule



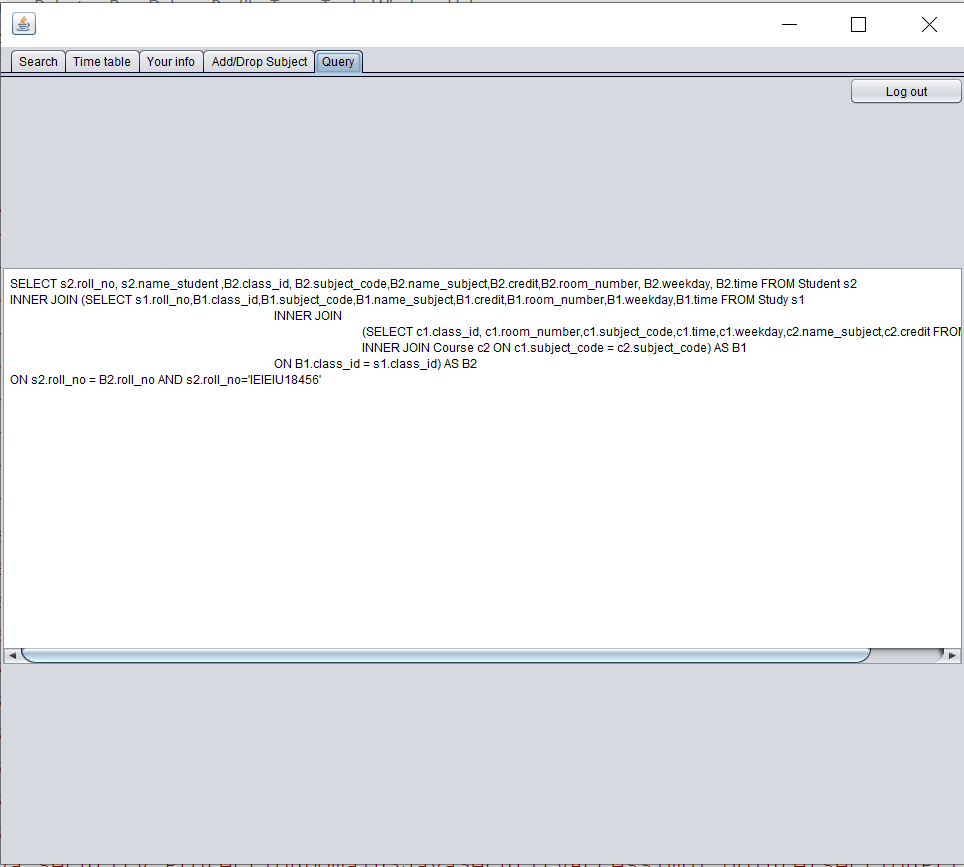
* And if he/she paid we click the schedule button. And going to the next “Time table” tab, it will display his/her information and schedule



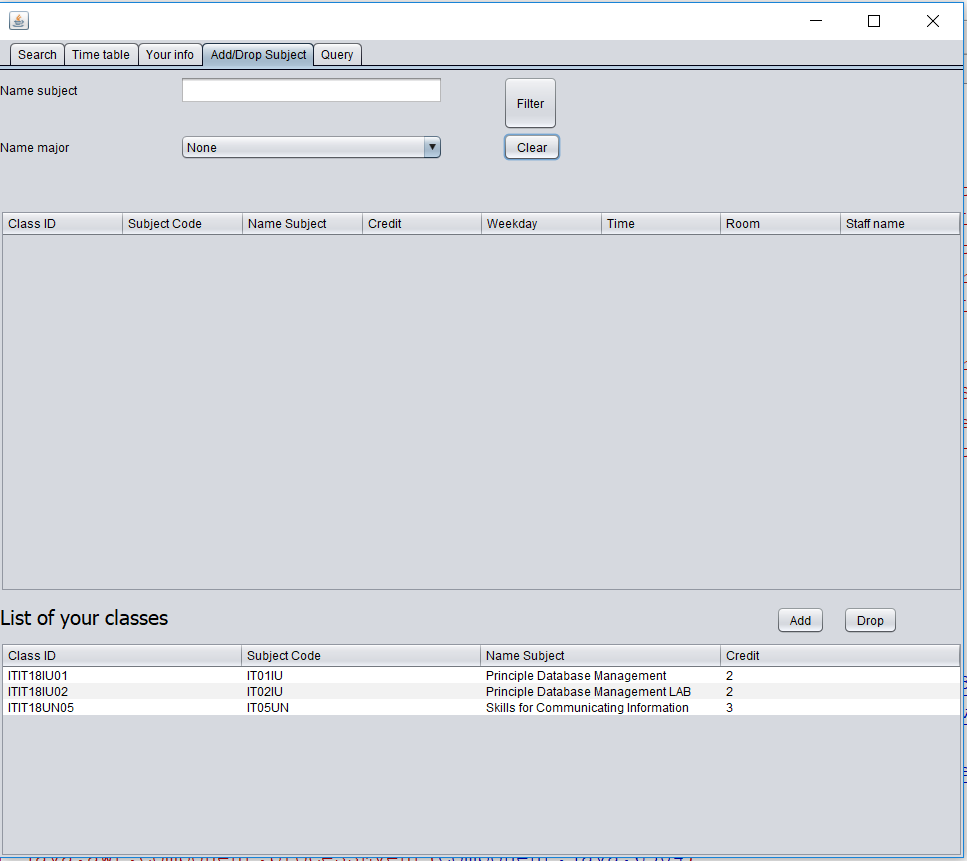
* In case that student used his/her account to log in and paid fee, just click the button “Schedule” then move to the “Time table” tab to see the schedule
* The user also can know more information about the course by double click in the course inside the schedule



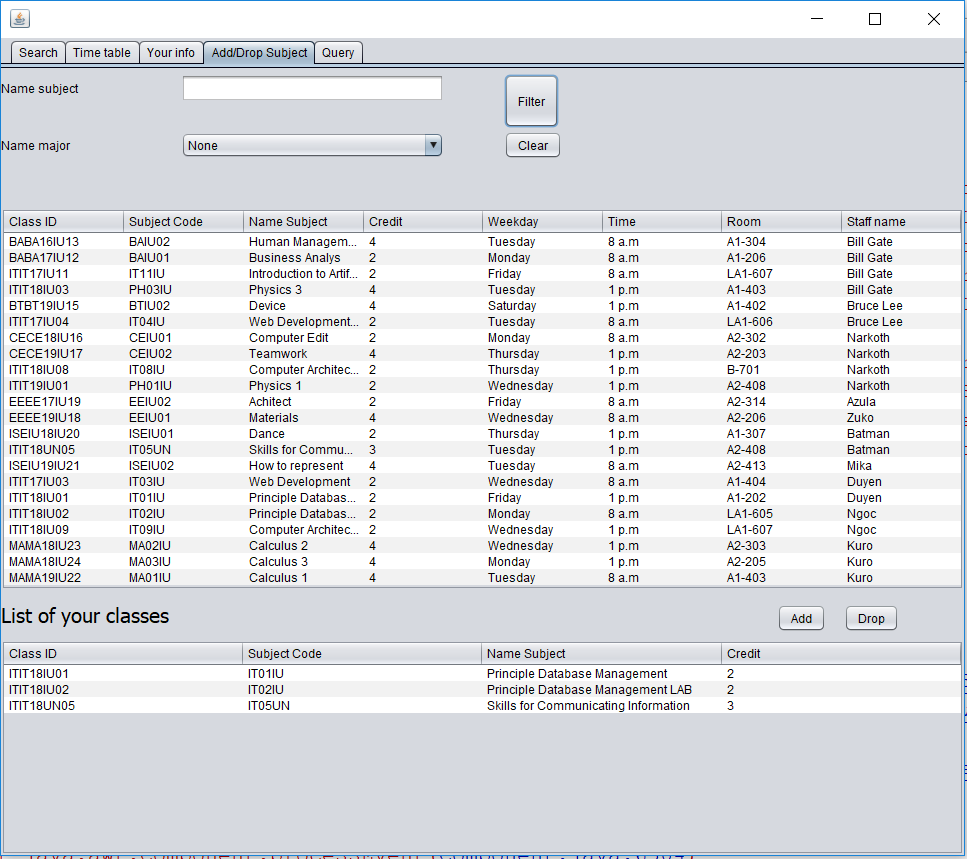
* Tab “query” is for displaying the query for all the actions that the user did, such as clicking buttons, each button has its own query so this tab will print it out into to screen to give us more comfortable to handle
* Example, when user choose ‘male’, this means he/she wants use search button to search all the student that is male



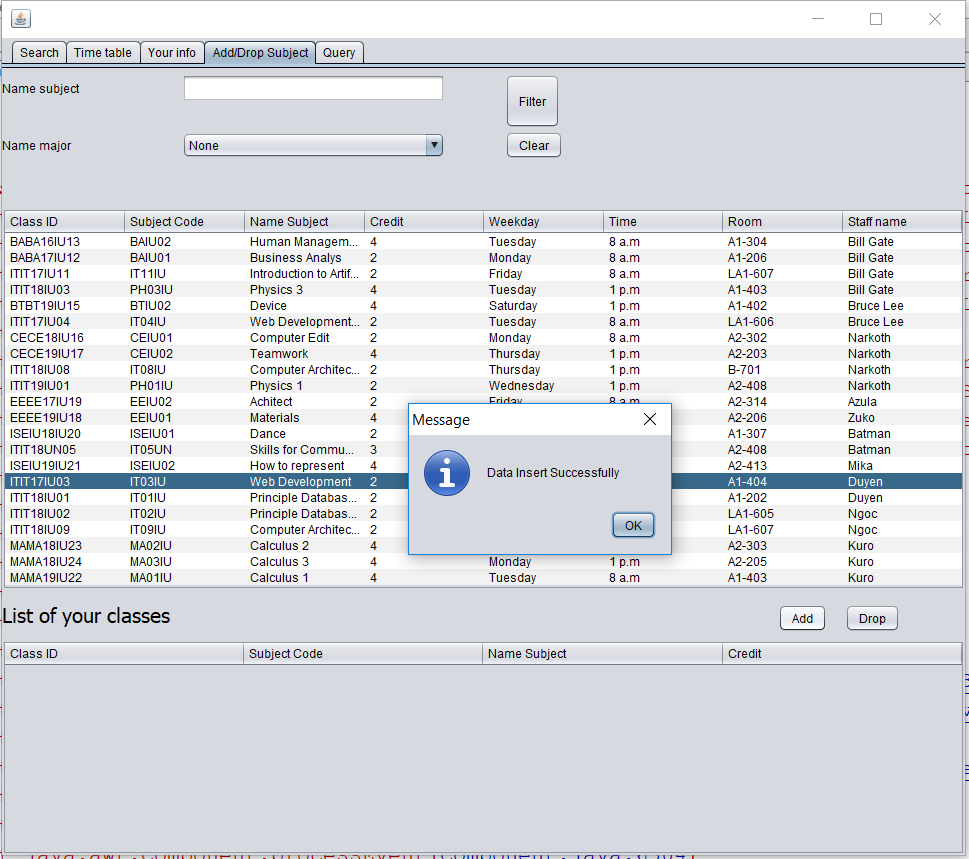
* In tab “Add/Drop Subject” allow students to add and drop their classes

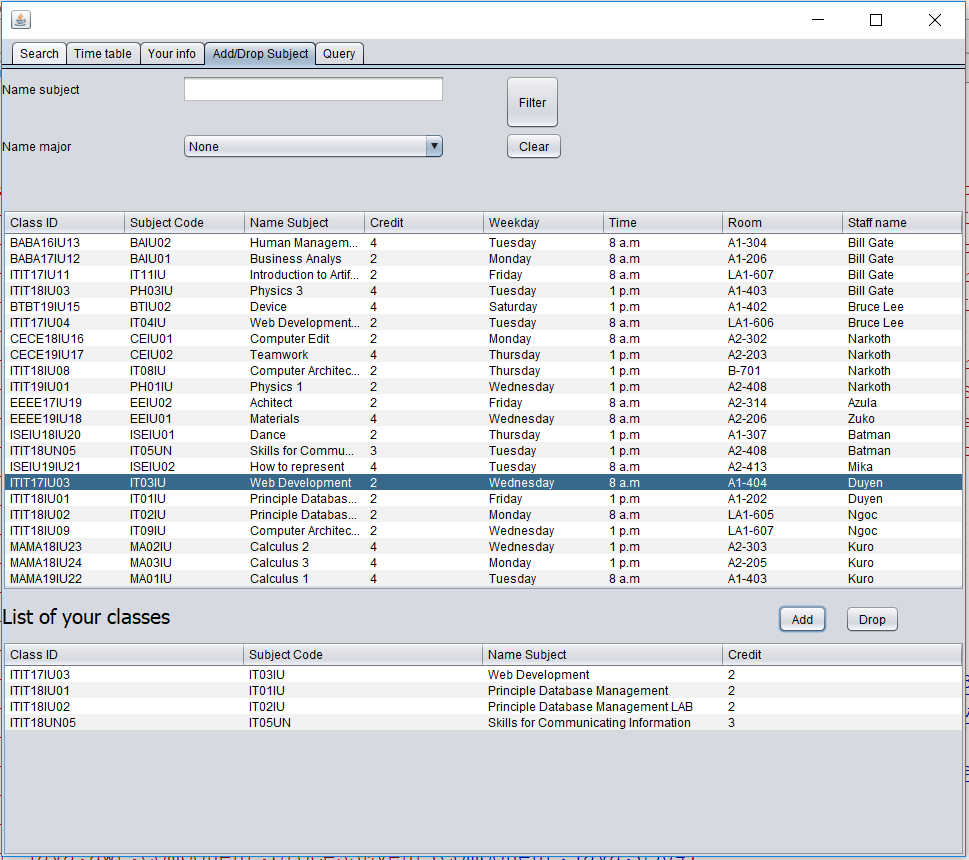


* Input the keyword into the textField Name subject or choose Name major, then click “Filter” button. It will search for the data. If the textField id empty, it will automatically display all the classes, which already inserted in the SQL, in table.

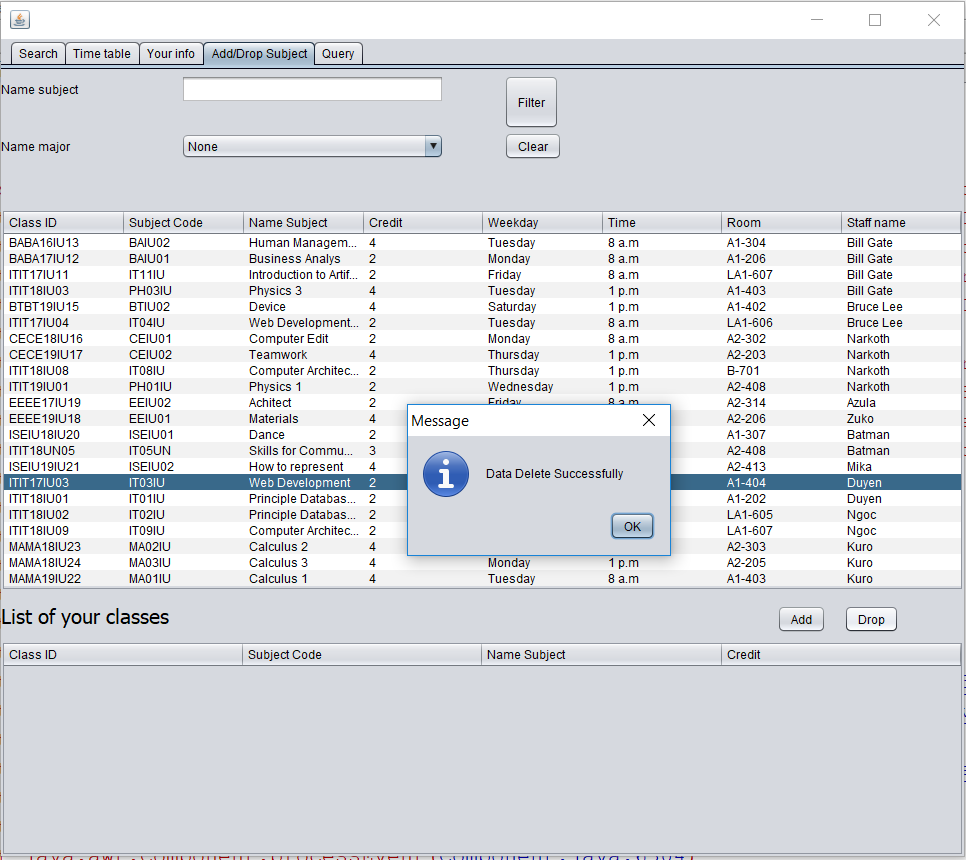


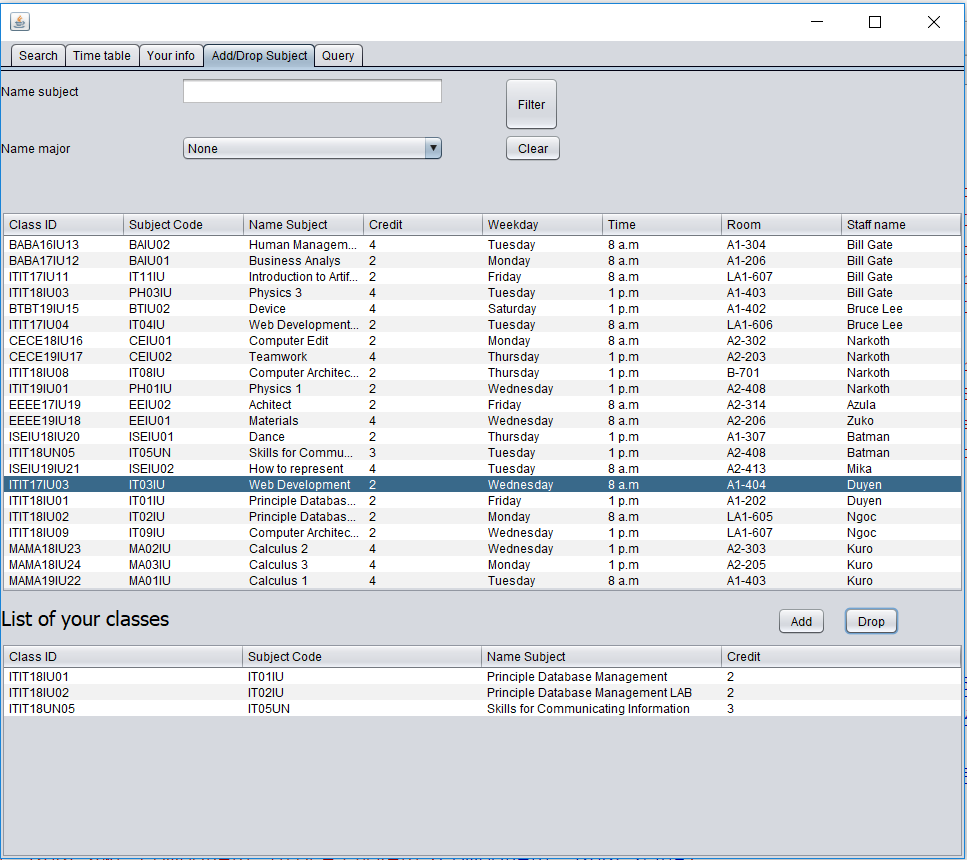
* When adding a class, it will auto update the tuition fee in the Amount of Pay





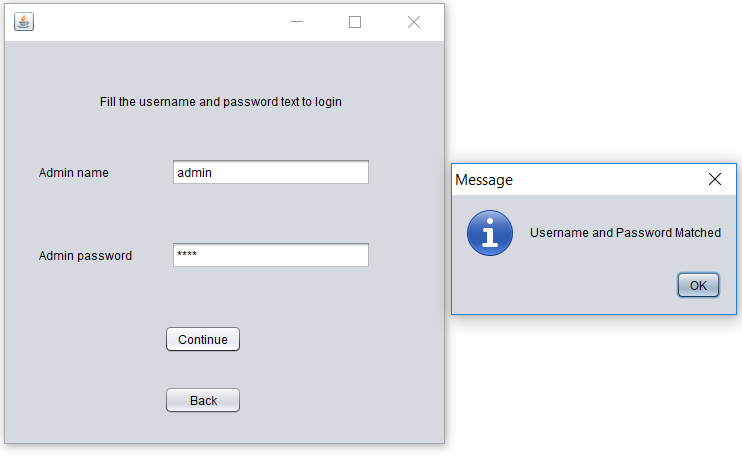
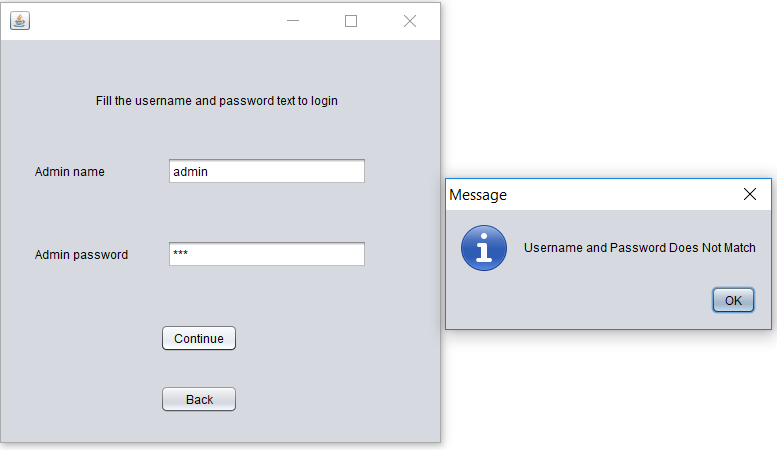
* When dropping classes, it will also automatically update the tuition fee in the Amount of Pay



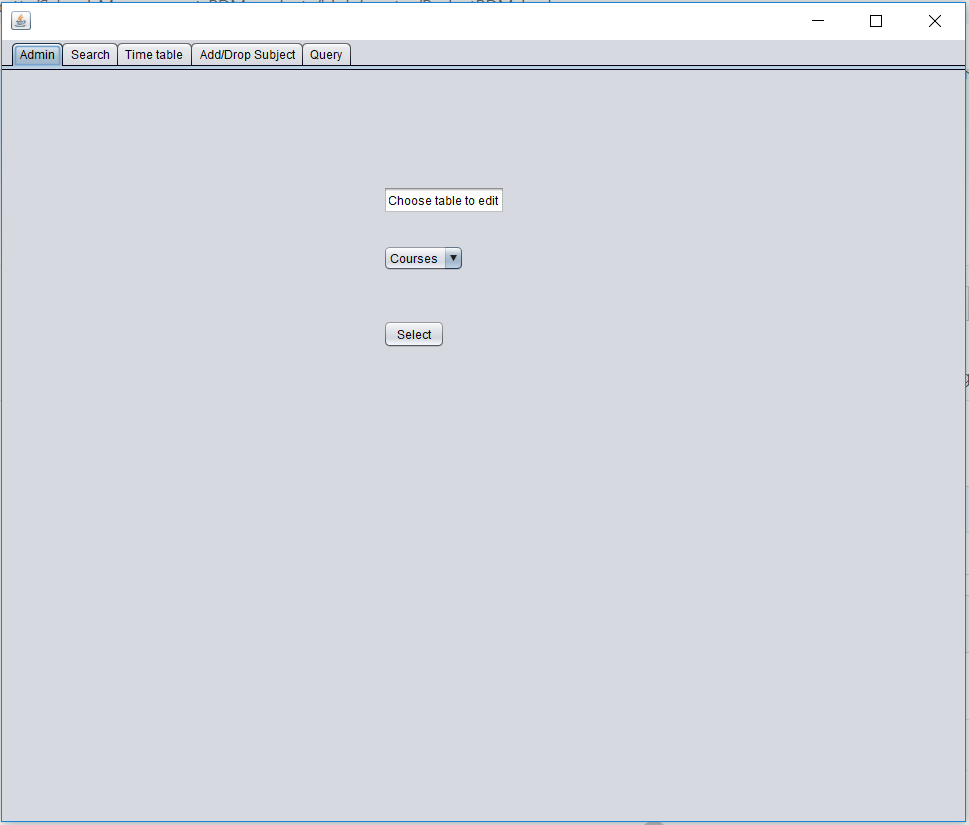


1. **Admin**

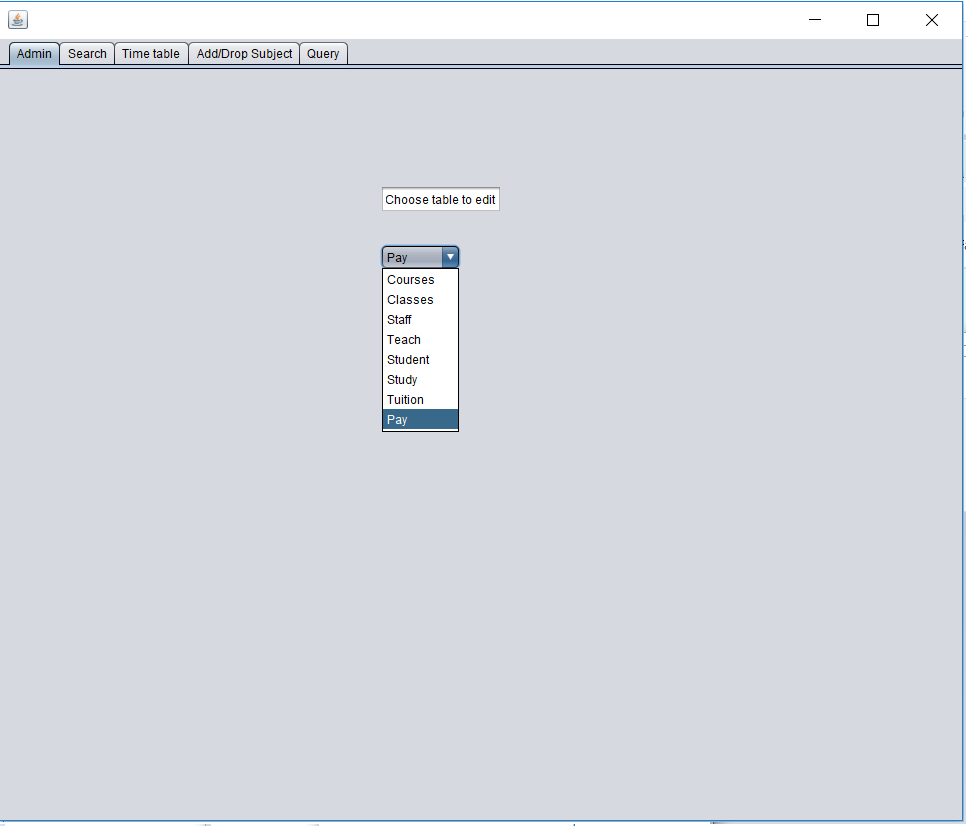
* When the user accesses the system by login with the role Admin
* Use the accounts that are stored in database (table “Account”) to login into the system. There are 2 possible cases can be happened: login successfully or fail



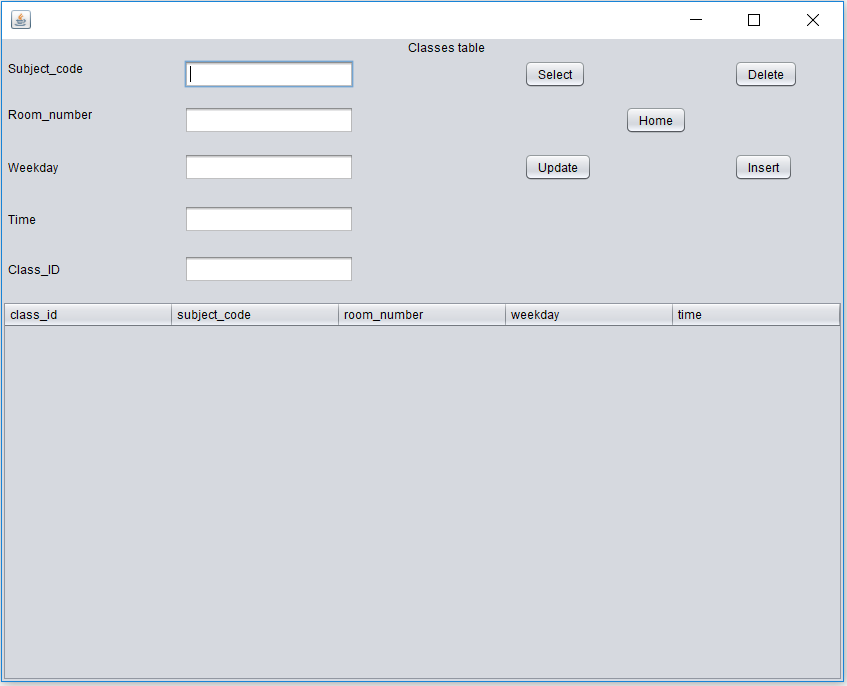
* This interface has 5 tabs: Admin, Search, Time table, Add/Drop Subject Query
* In Tab “Admin” user is allowed to edit and handle all tables and data in database by entry to each table



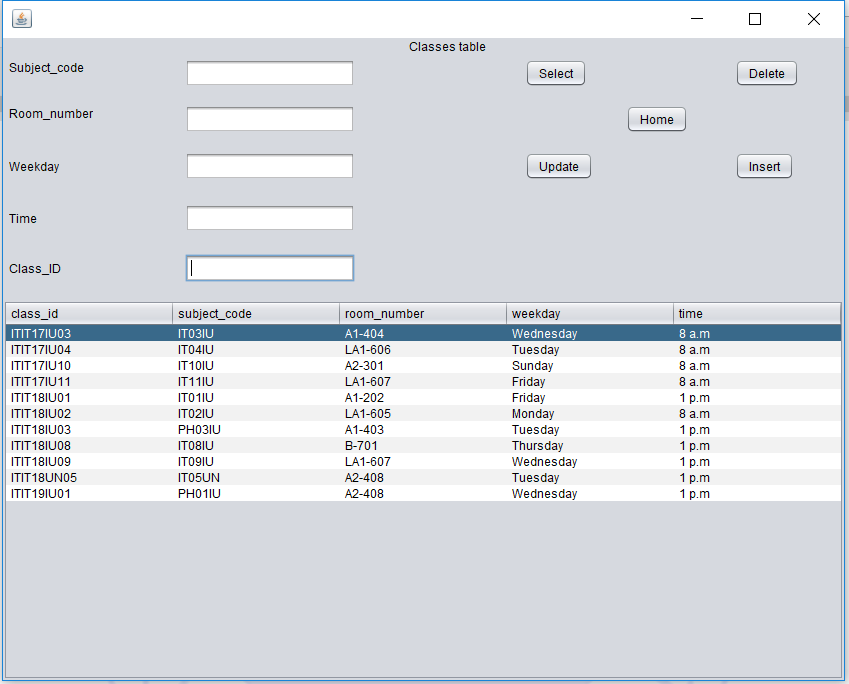
* Choose which table he/she wants to entry



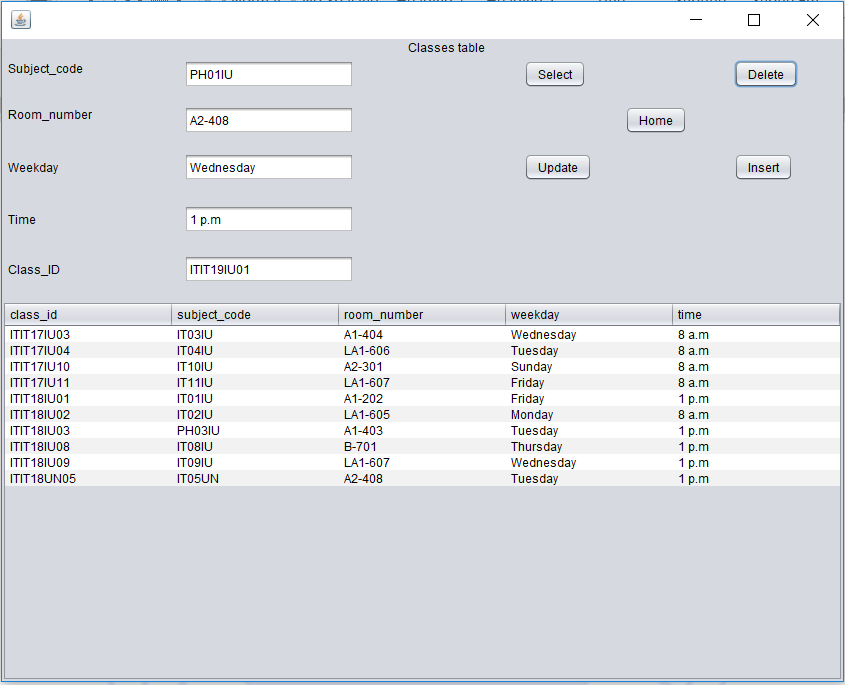
* And then click the select button to go into the jframe that contains table in database
* The system has 8 tables: Courses, Classes, Staff, Teach, Student, Study, Tuition, Pay rely on the SQL that are already created
* First starting with the table Classes



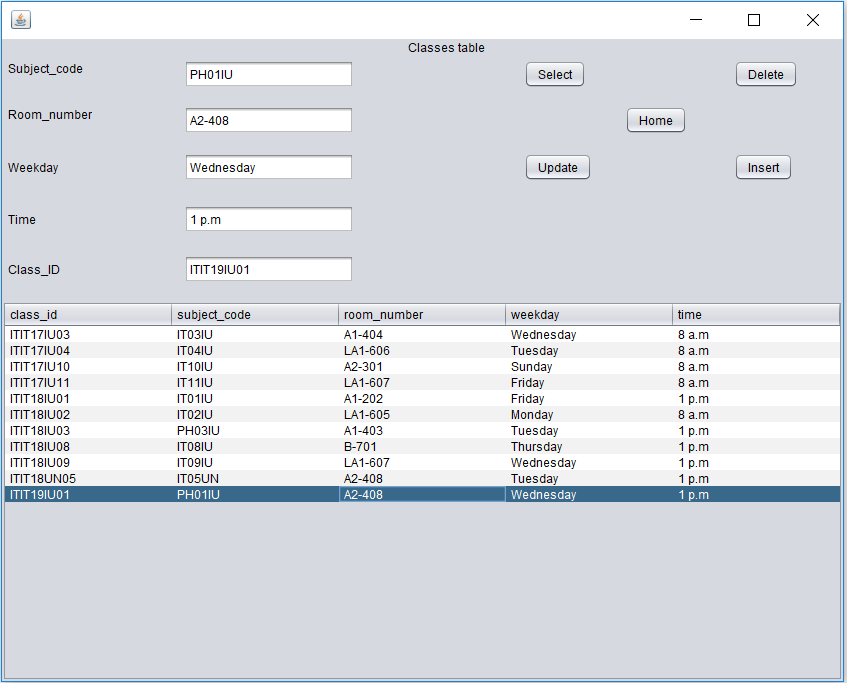
* There are 5 buttons here: Home(return to previous), Select, Insert, Update, Delete
* When user click “Select” it will display all database of the classes



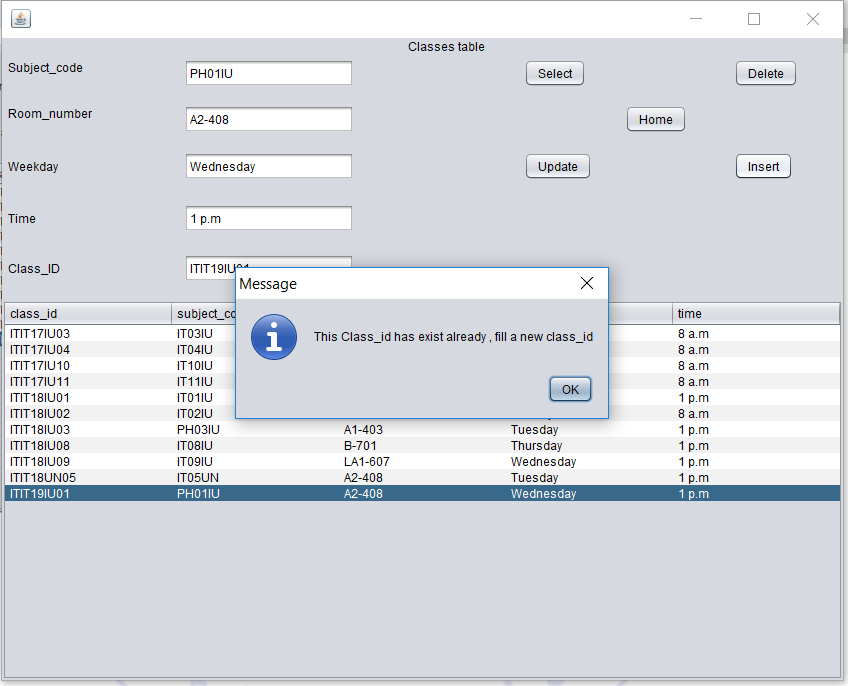
* Click the button “Delete” it will delete this class data



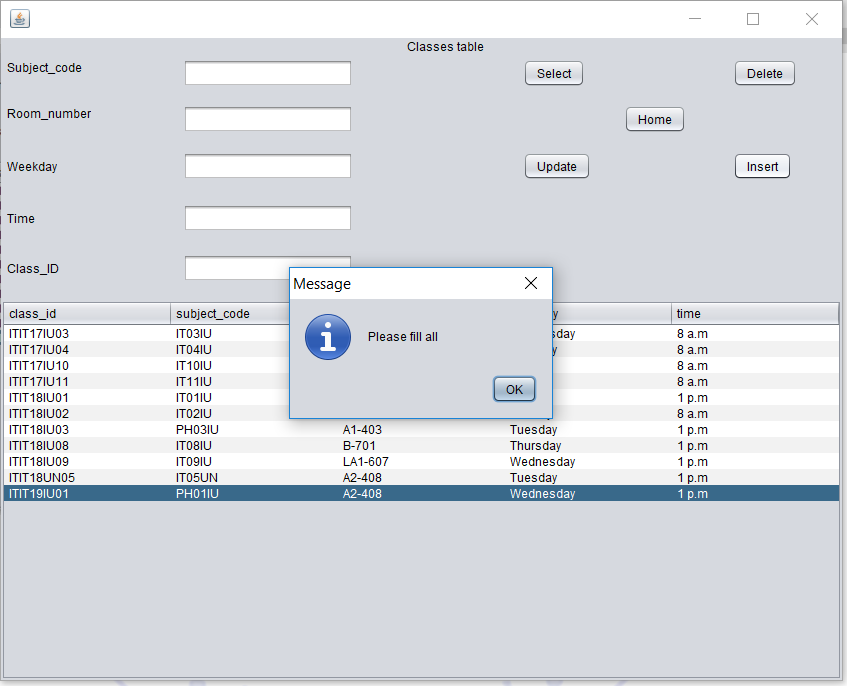
* Delete the class with class\_id = ITIT19IU01 then fill the information of this class in textfield and click the “Insert” button to insert it to database again



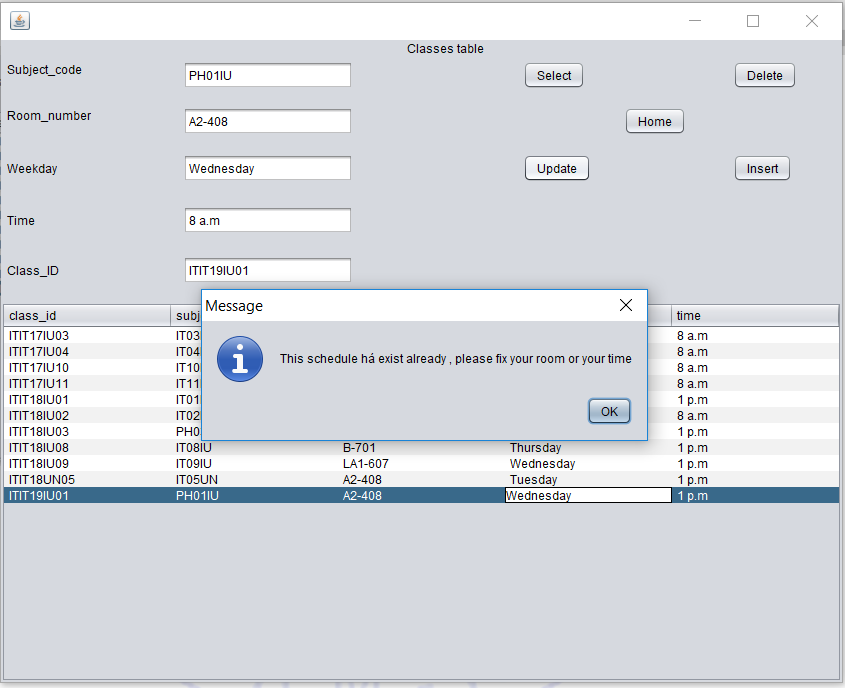
* If the class is already existed, it will display the error message “This class\_id has exist already, fill a new class\_id”



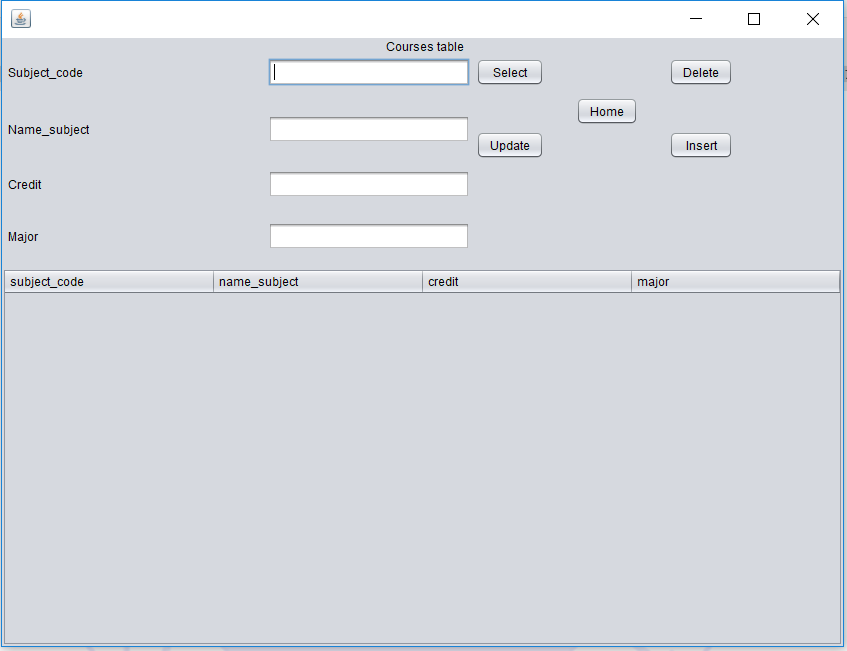
* Or if the user not fill anything in text field but click the “Insert” button the system will display error message “Please fill al”



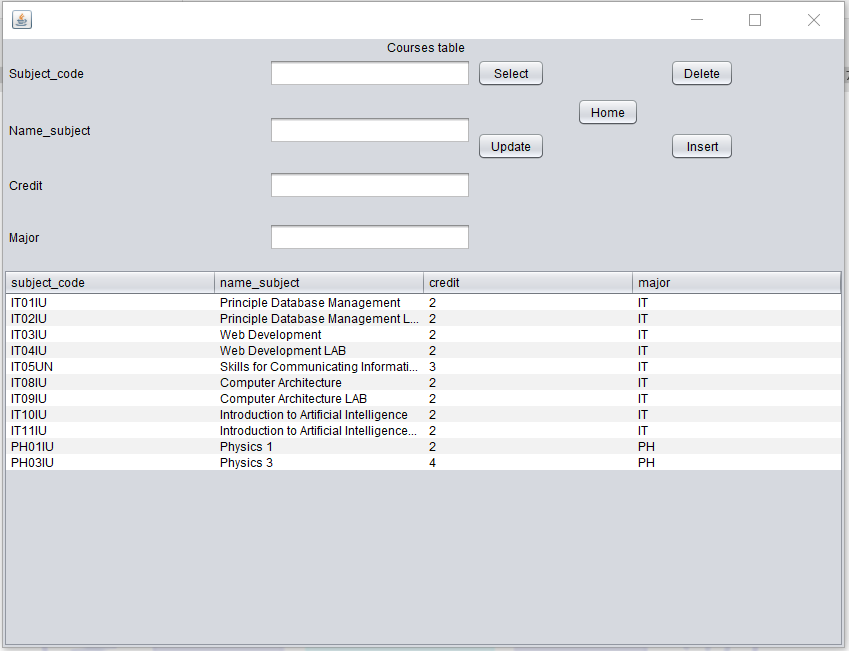
* Next, the “Update” button is easy to use, just click to the class user that the user wants to choose and edit the information in textfield and click button
* In the case, user wants to update the information and the room time and weekday is duplicated with another class time or room the system will display the error message “This schedule has exist already, please fix your room or your time”



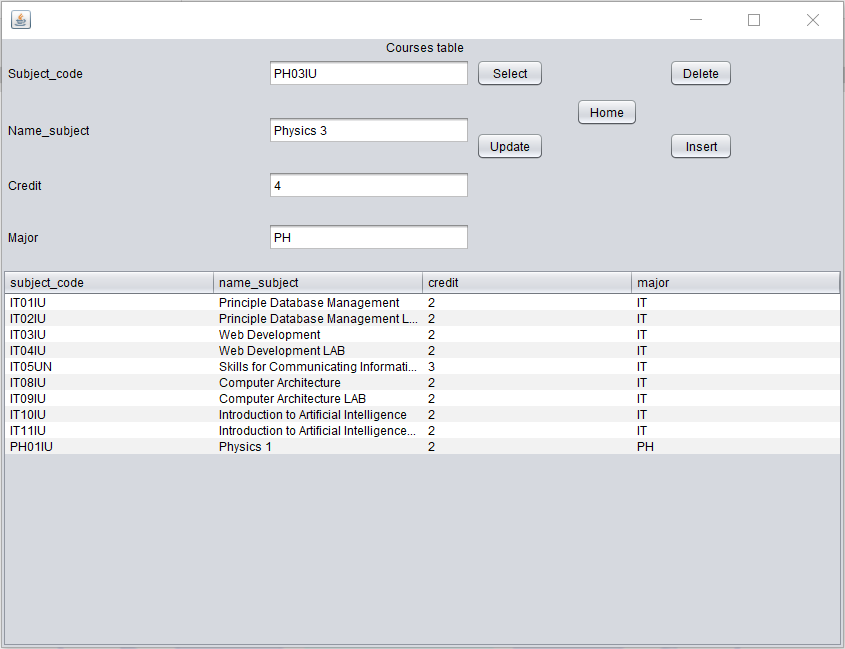
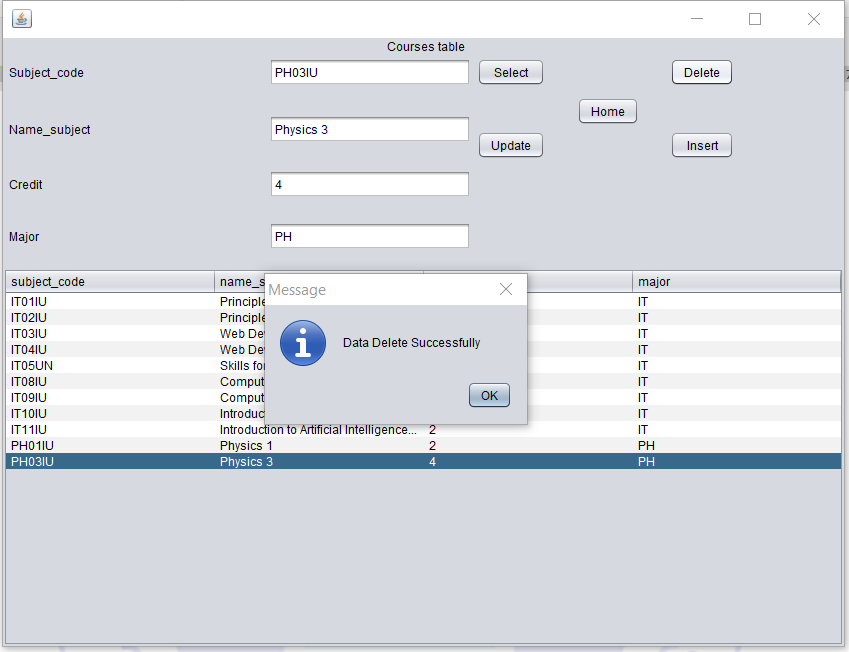
* Then move to the the the Jframe Courses



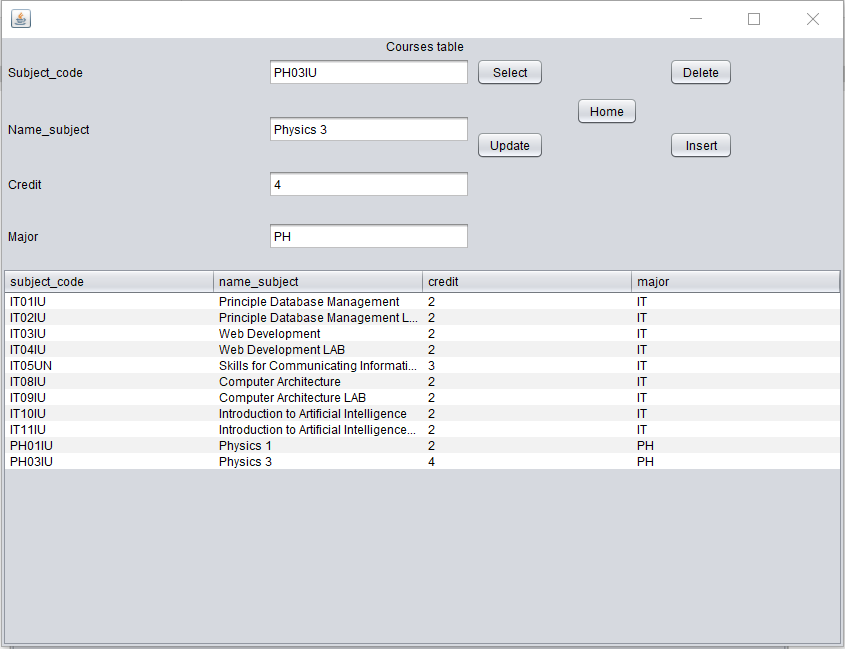
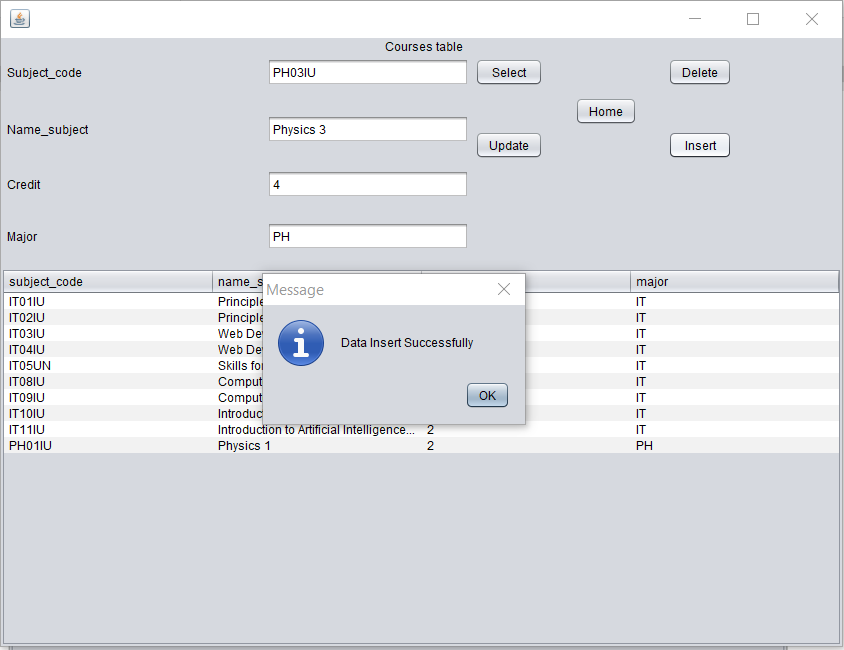
* There are 5 buttons, similarly to Classes: Home(return home), Select, Delete, Insert, Update
* Click the Select button it will show all the courses



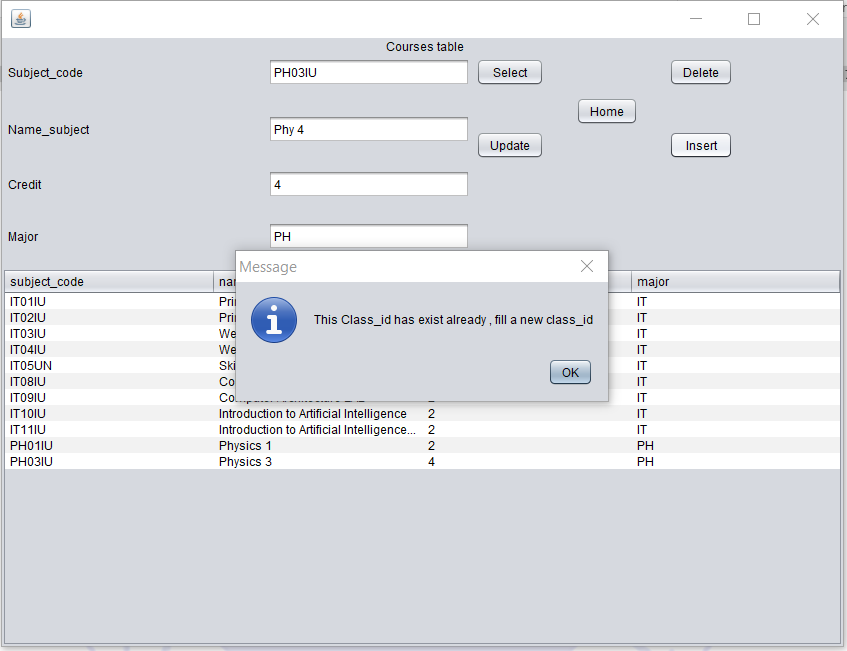
* When the user choose course to delete



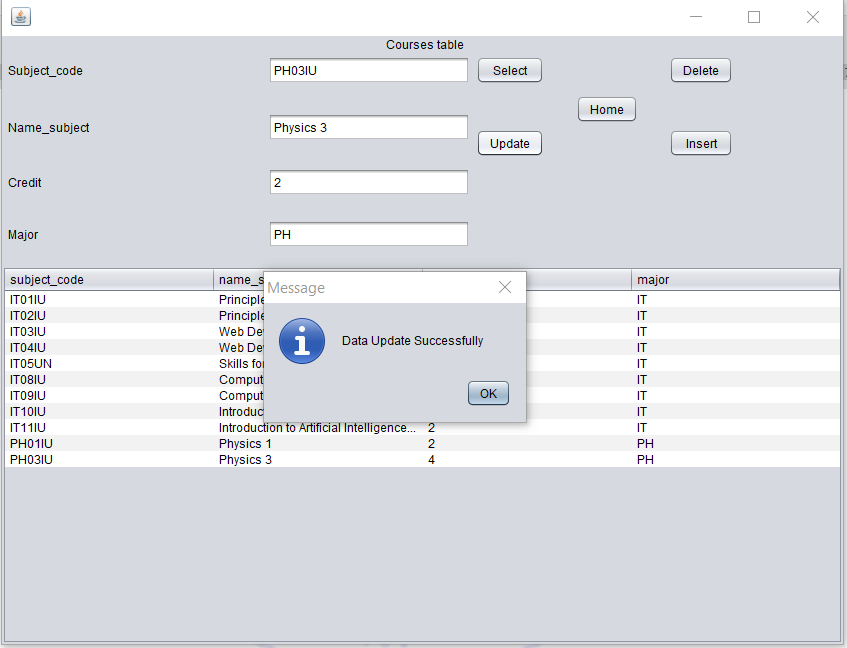
* User inserts the course that he/she deleted it before

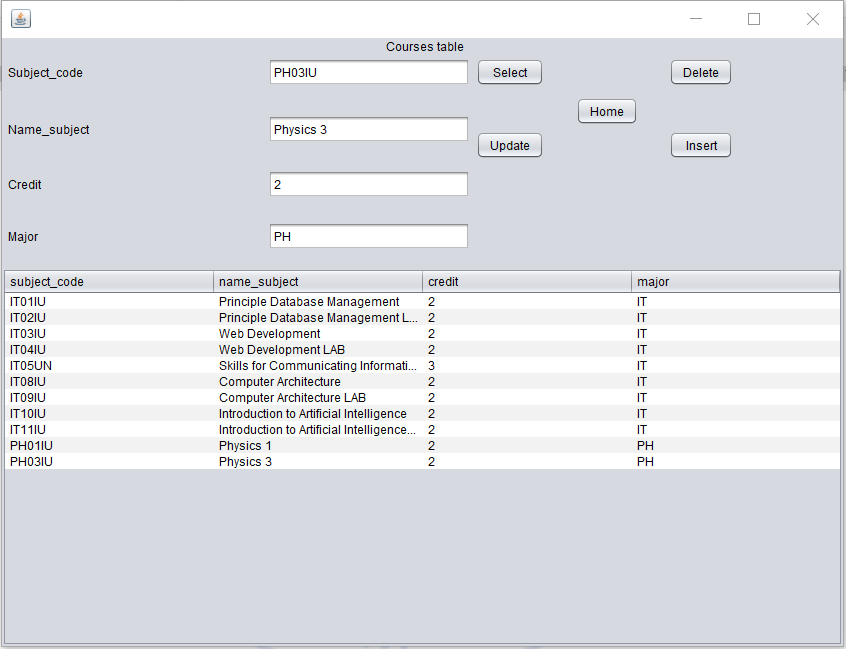


* In case the user inserts the course that already in the data table, it will print out error “This class\_id has exists already, fill a new clas\_id”

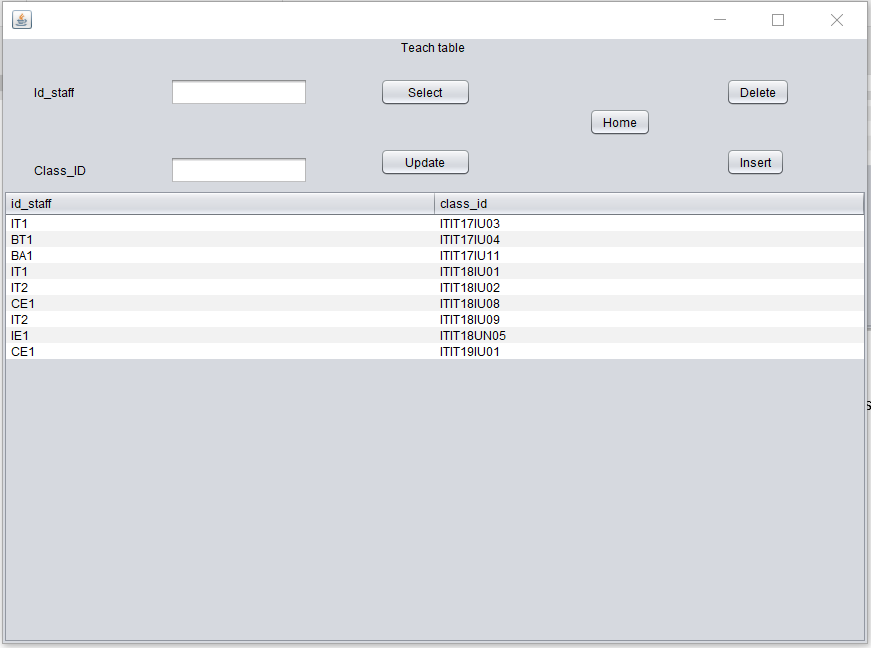
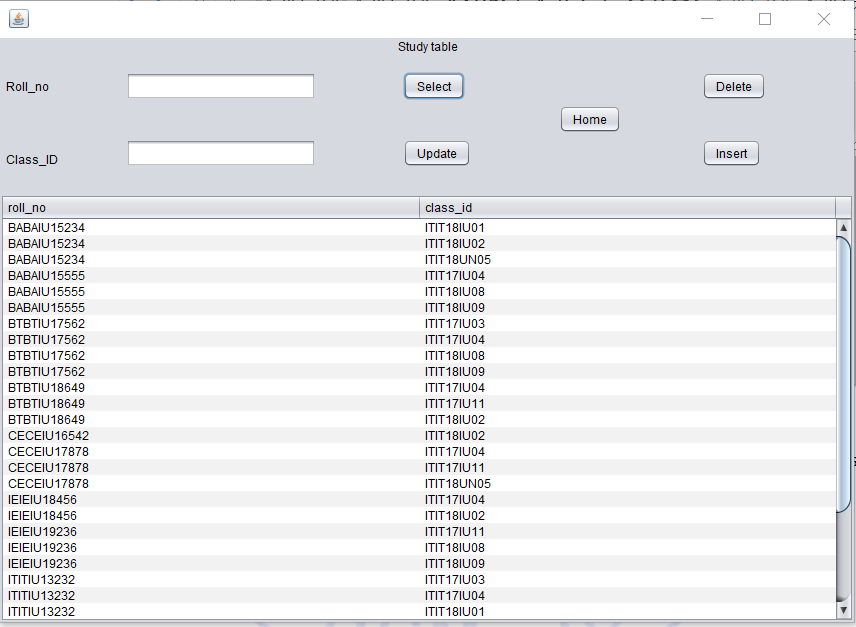
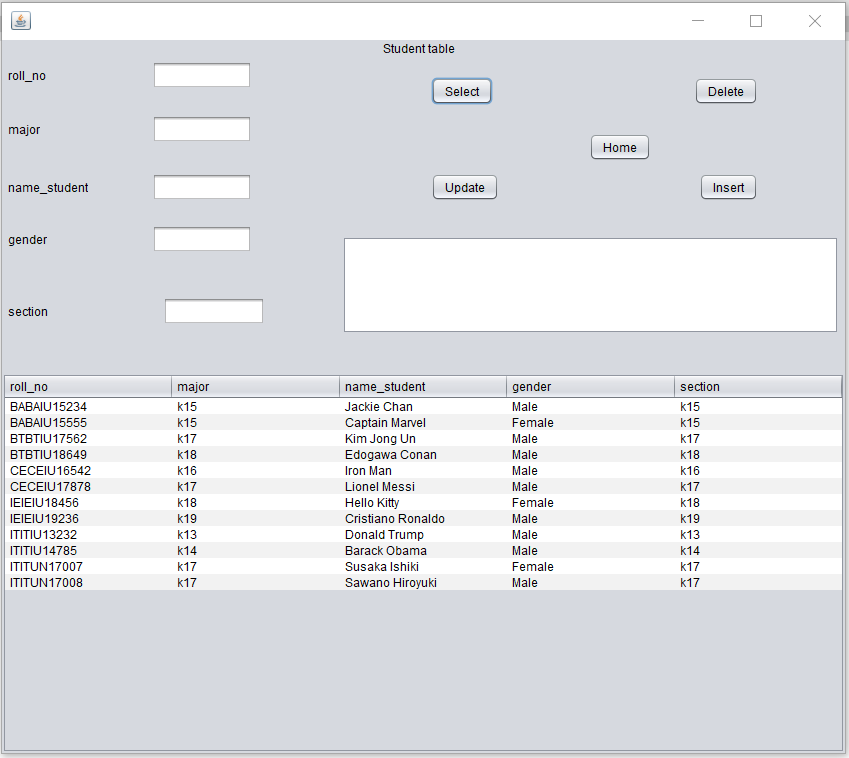
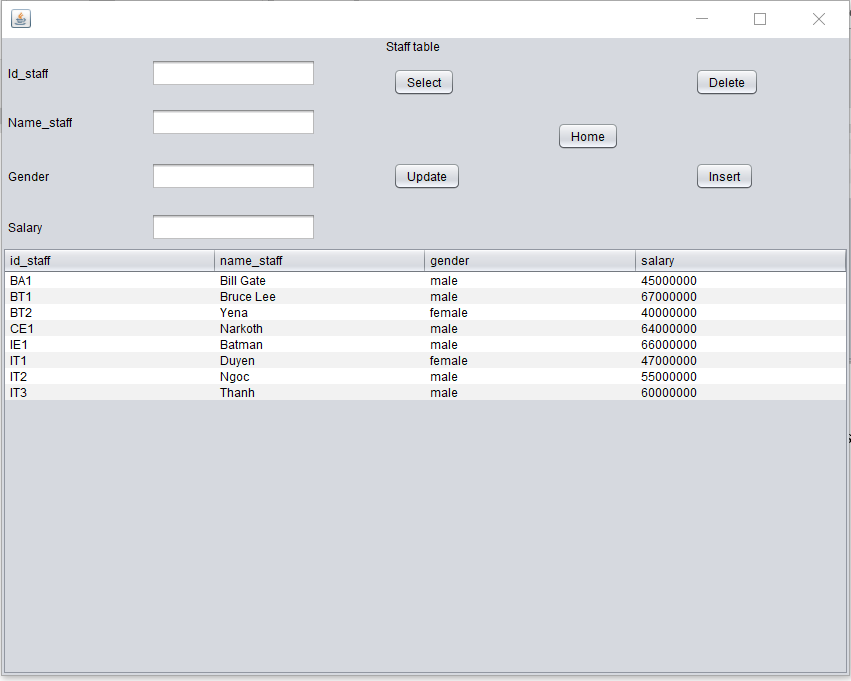


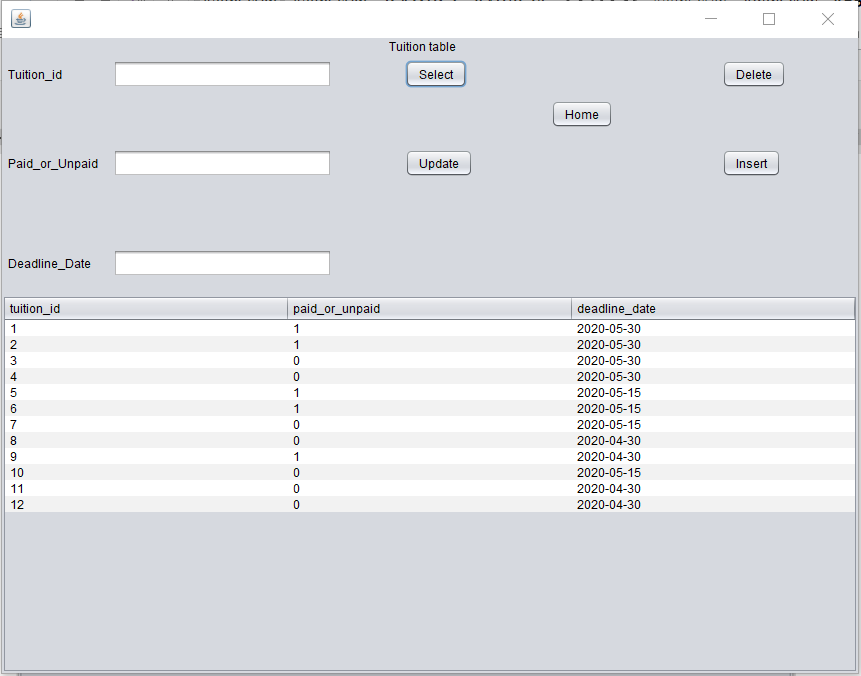
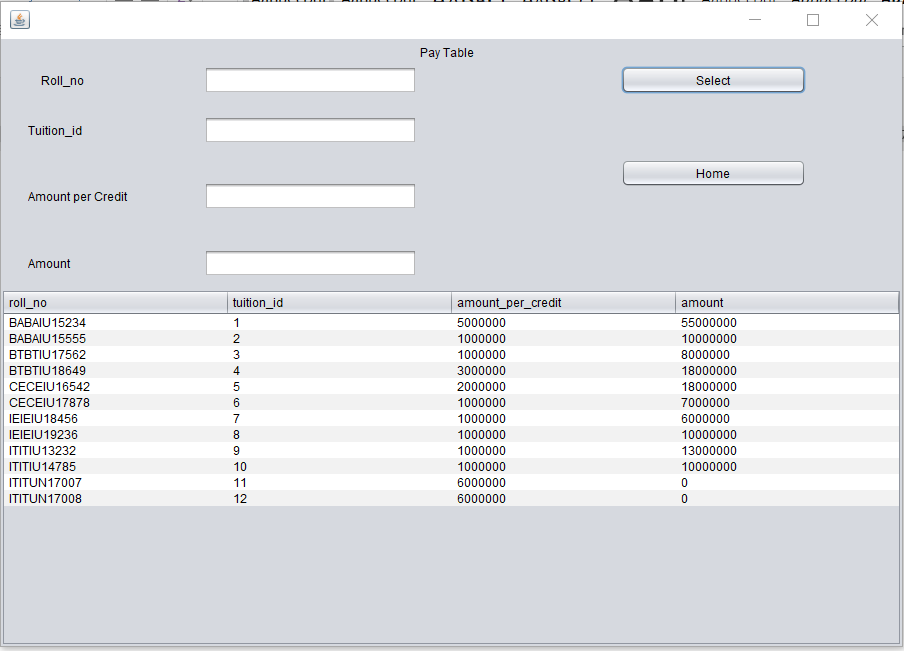
* Update button is also simple, just choose course and edit the information in text field then click update
* For example, change the credit for Physics 3 from 4 to 2





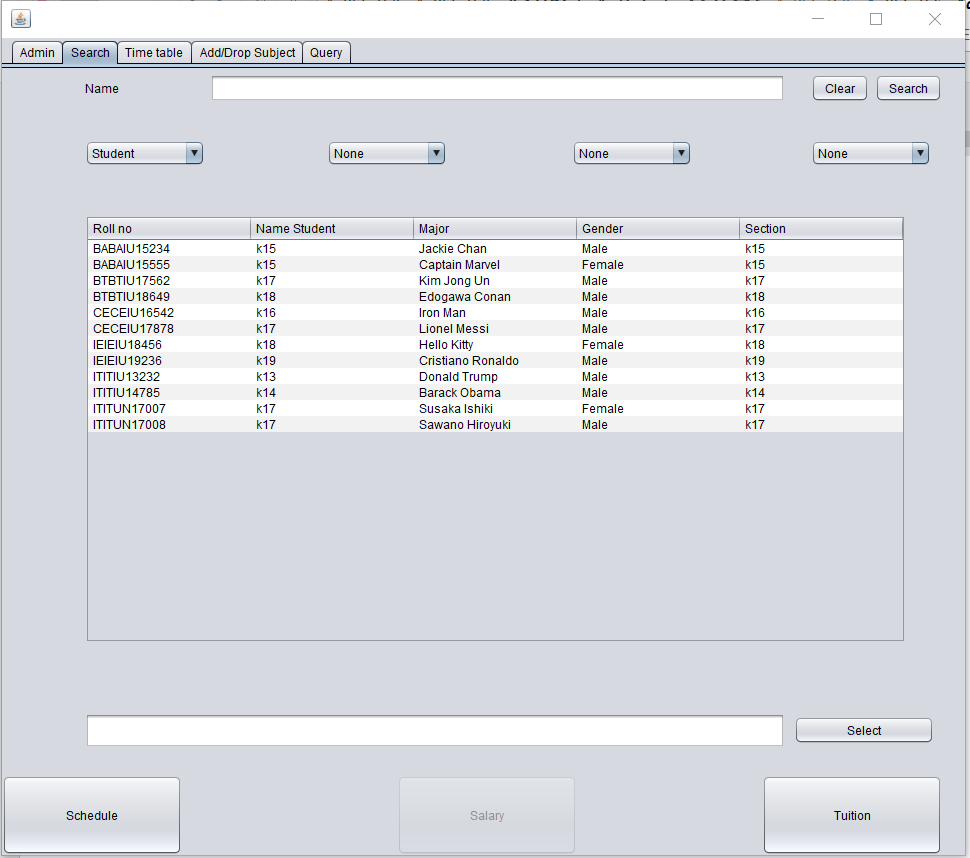
* Then move to the other Jframes: Staff, Teach, Student, Study, Tuition, Pay
* These table are similarly, they have the same function and buttons with Course and Classes



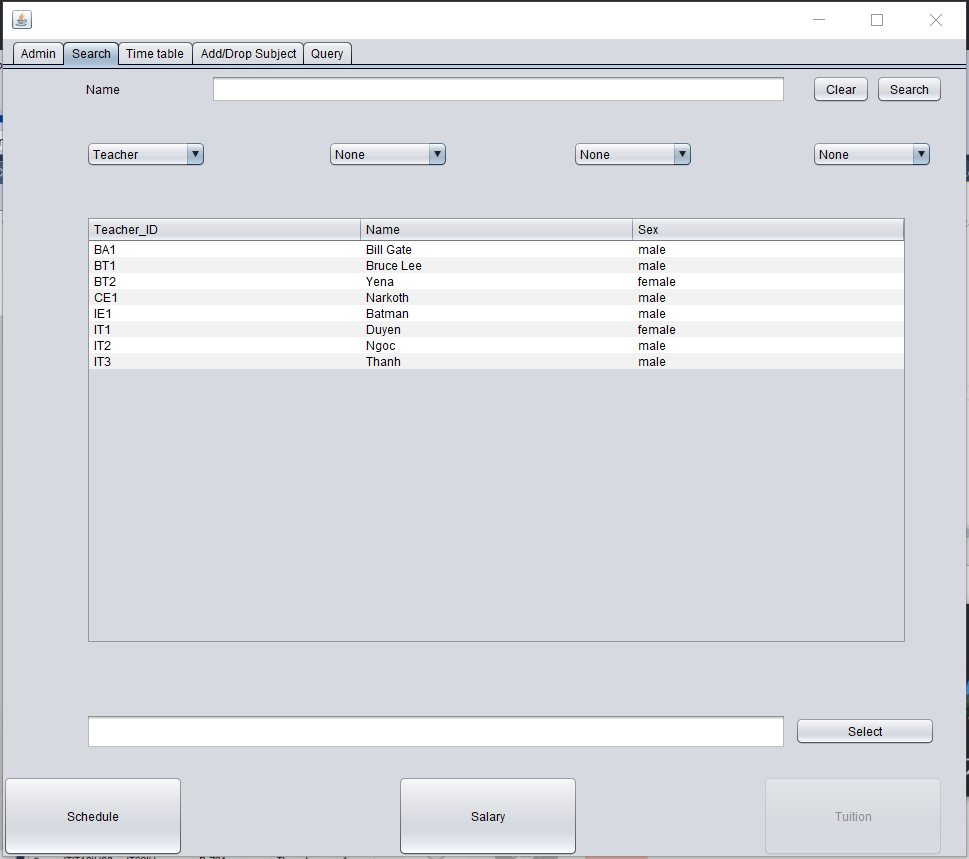


Return to home and move to the next tab “Search”

Click the “Search” button to see the list of student and teacher(choose Student/Staff with checkbox)



In the student role the salary button below is disable, tuition is enabled to click because student just has tuition fee



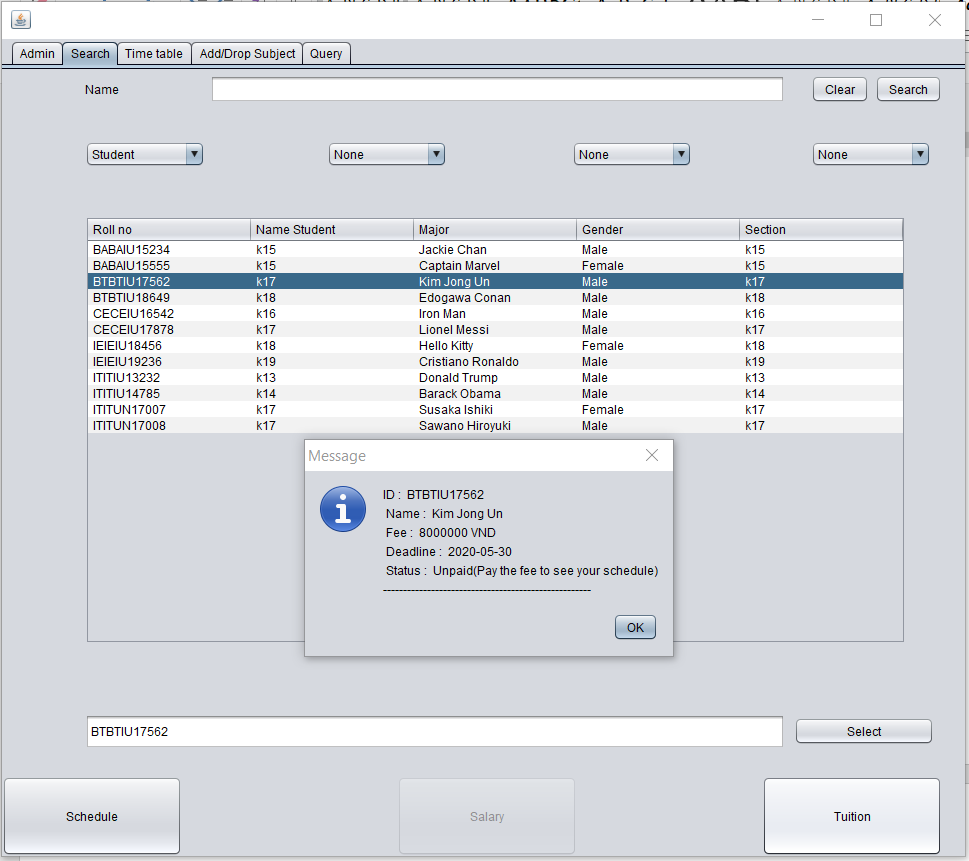
In contrast, when the user in the staff role, the button Tuition below is disable, Salary button is enabled to click because staff only has salary

To see student’s tuition fee and schedule:

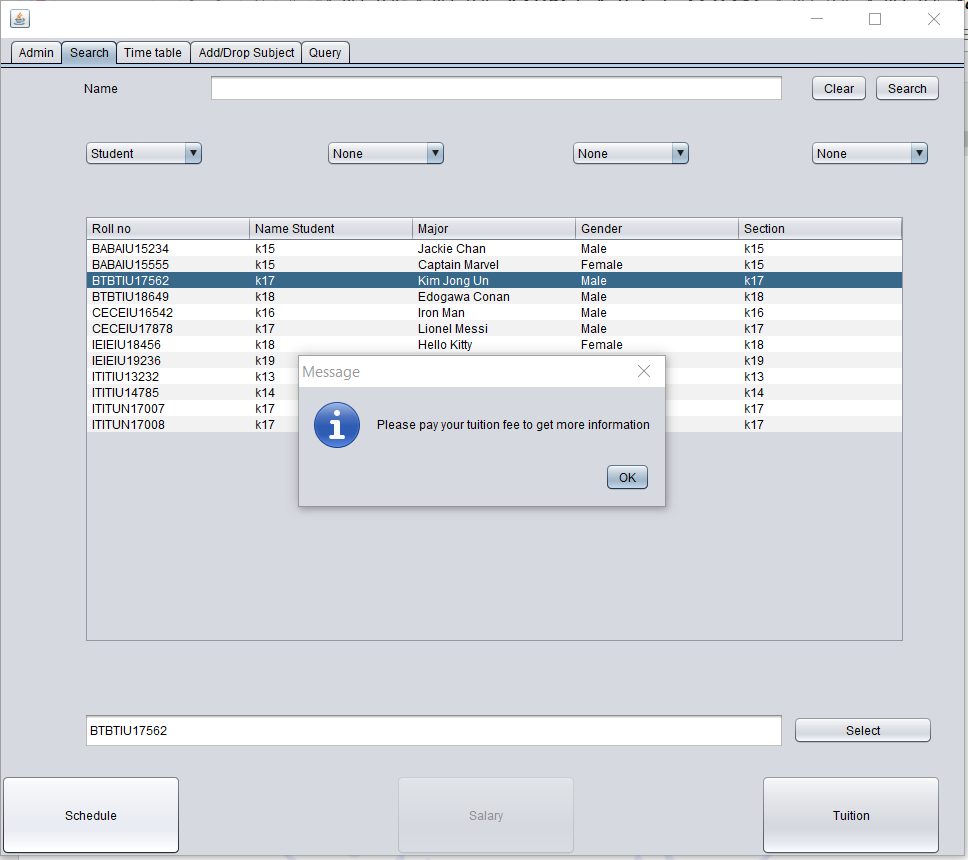
First use the mouse click to the student which we want to see

Then click the “select” button to lock he/she

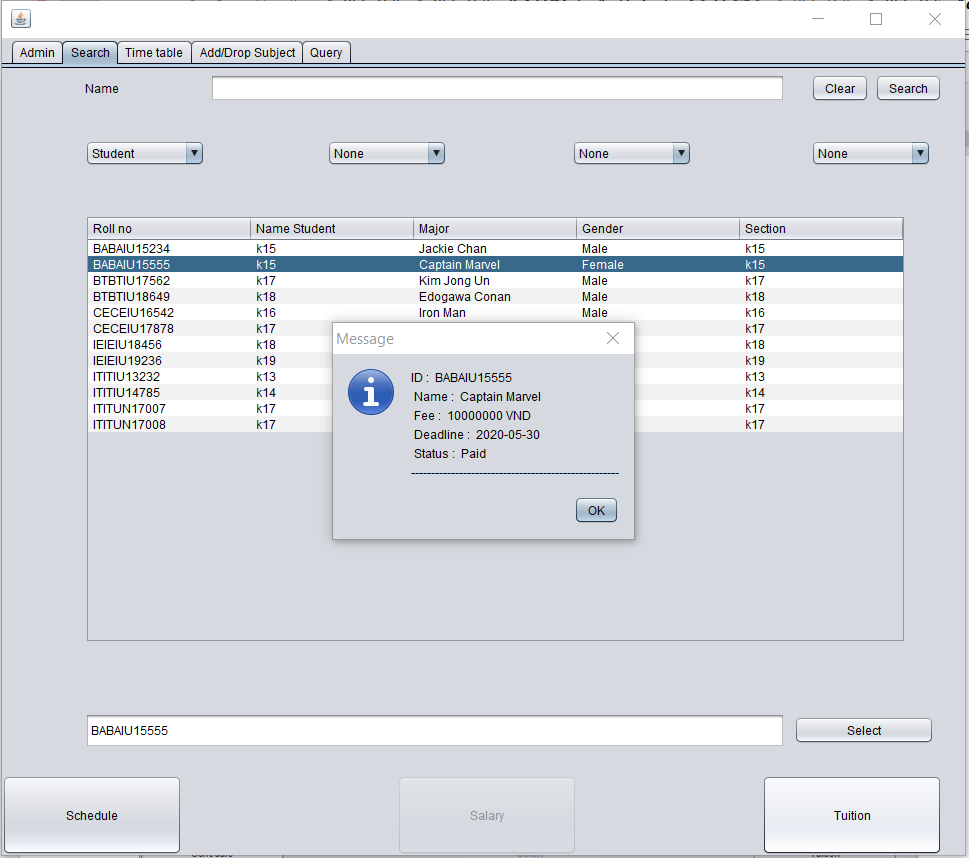
Then click the “Tuition” button to see tuition fee of his/ her

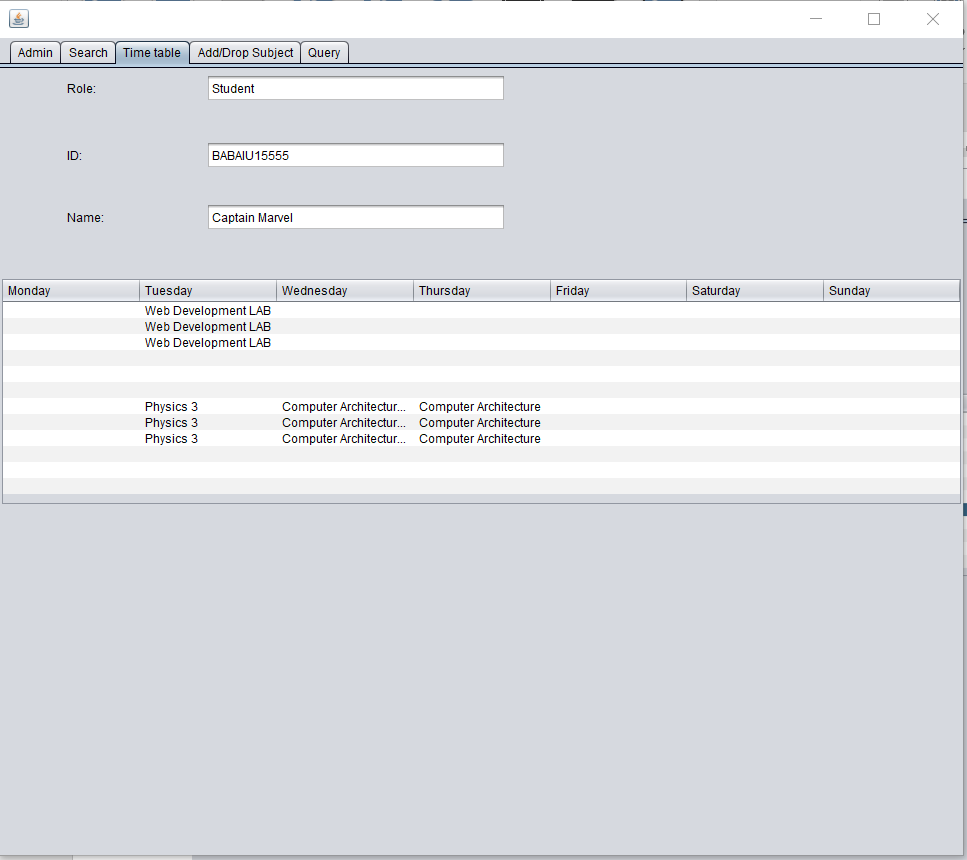


When we want to see Schedule, make sure that the student which we selected has already paid the tuition fee, if not it will show the message “Please pay your tuition fee to get more information”

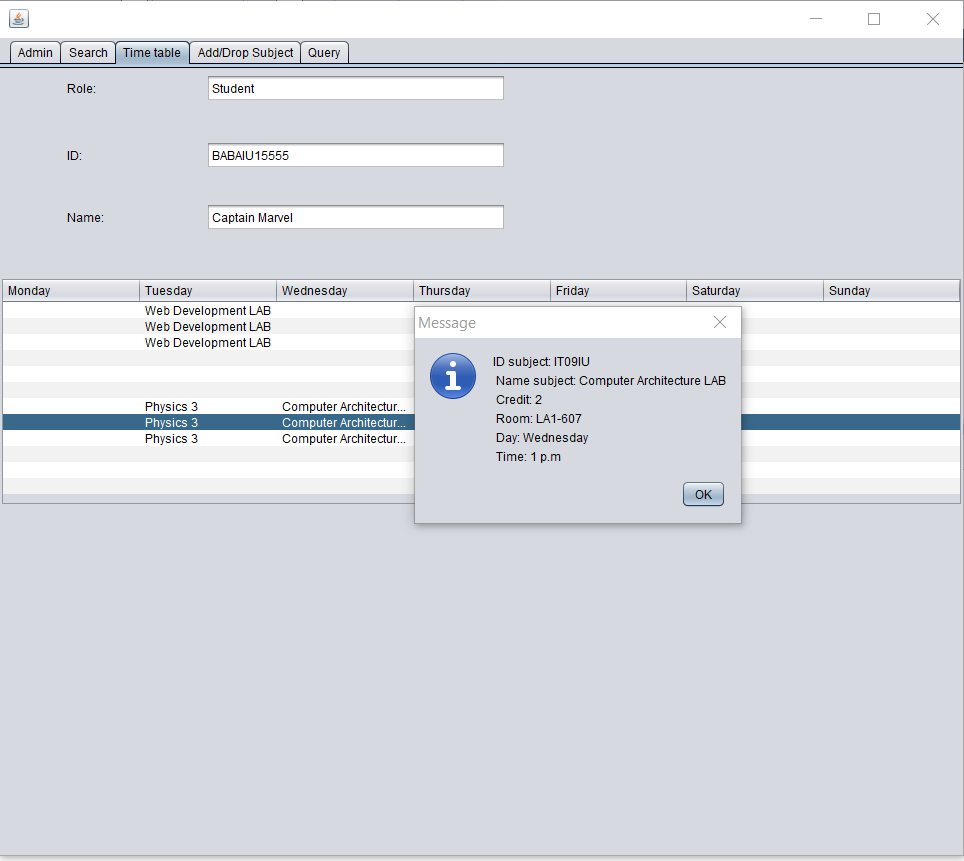


Otherwise, if the student already paid the fee, the user can see by clicking the “Schedule” button and move to the tab “Time table” to see that student’s schedule

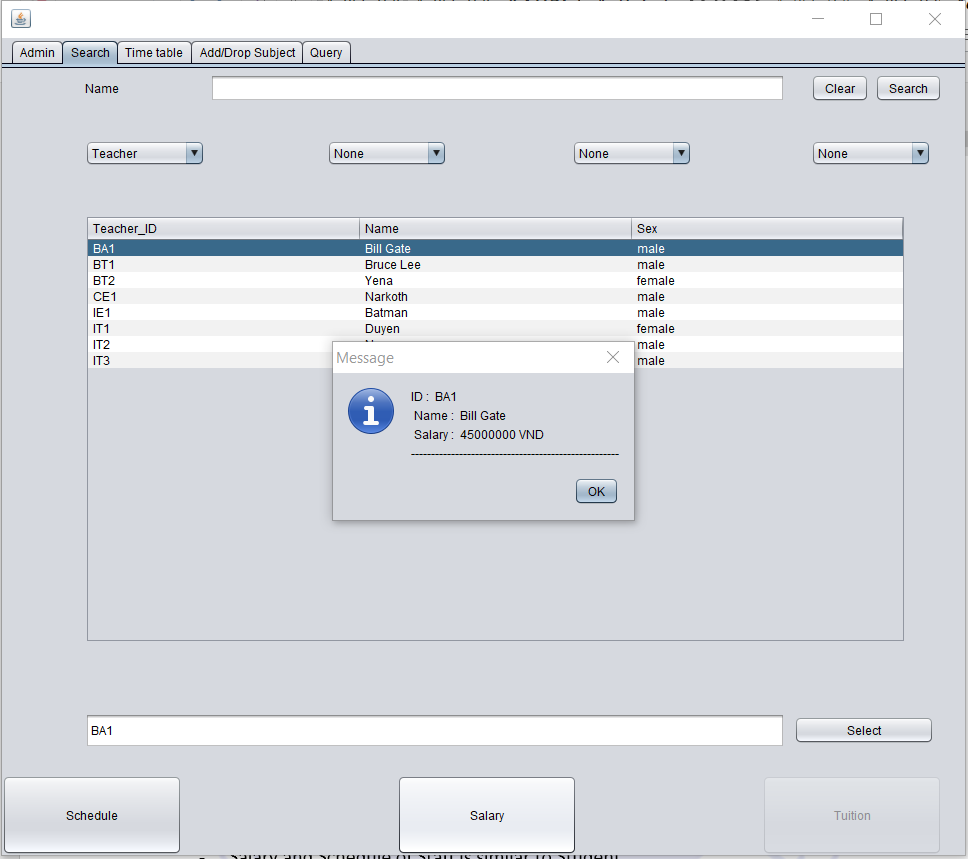


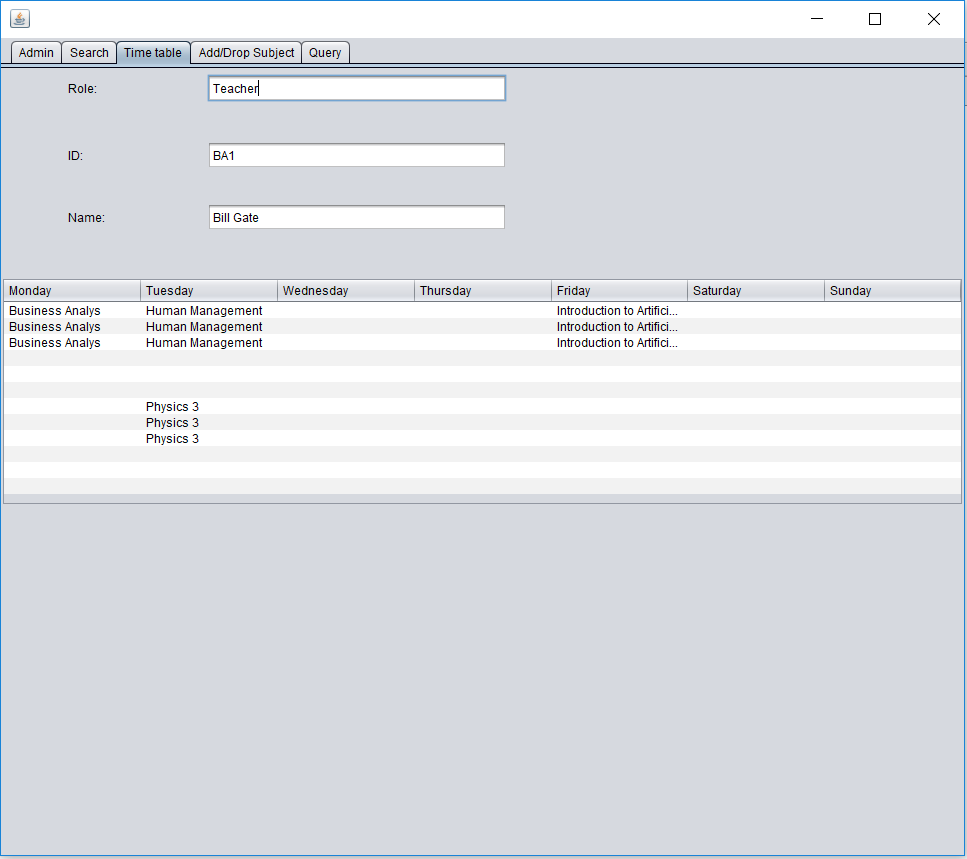


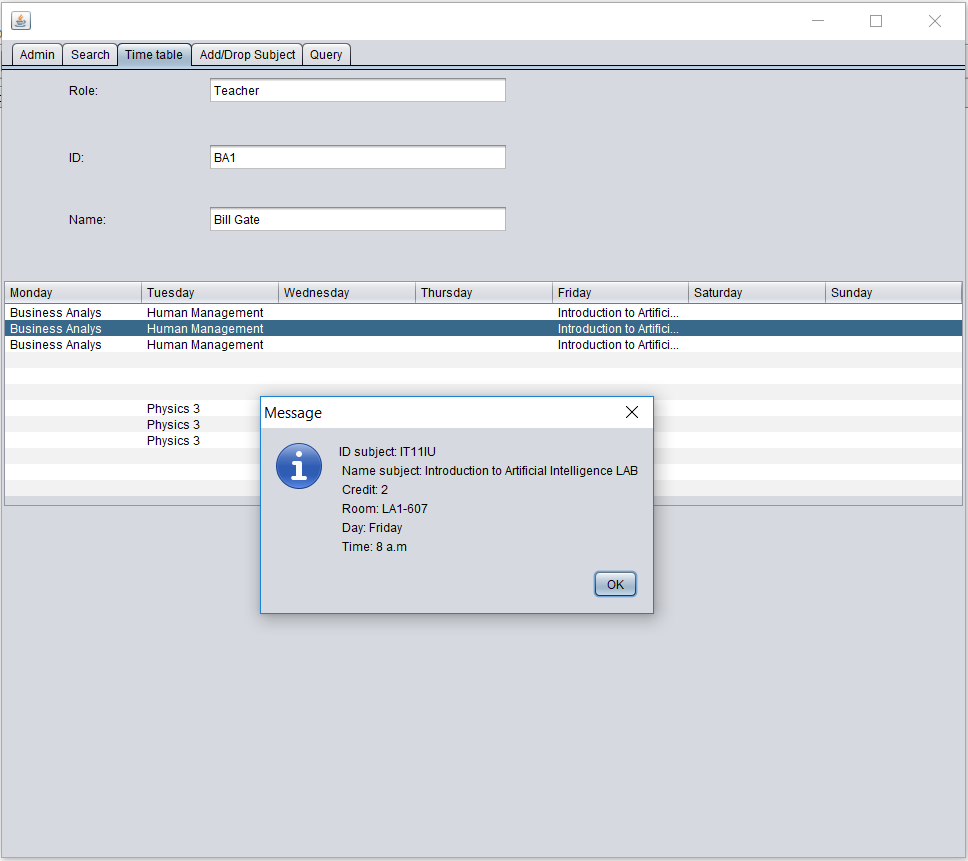
Especially, user can click to the course to see this information



Salary and Schedule of Staff is similar to Student

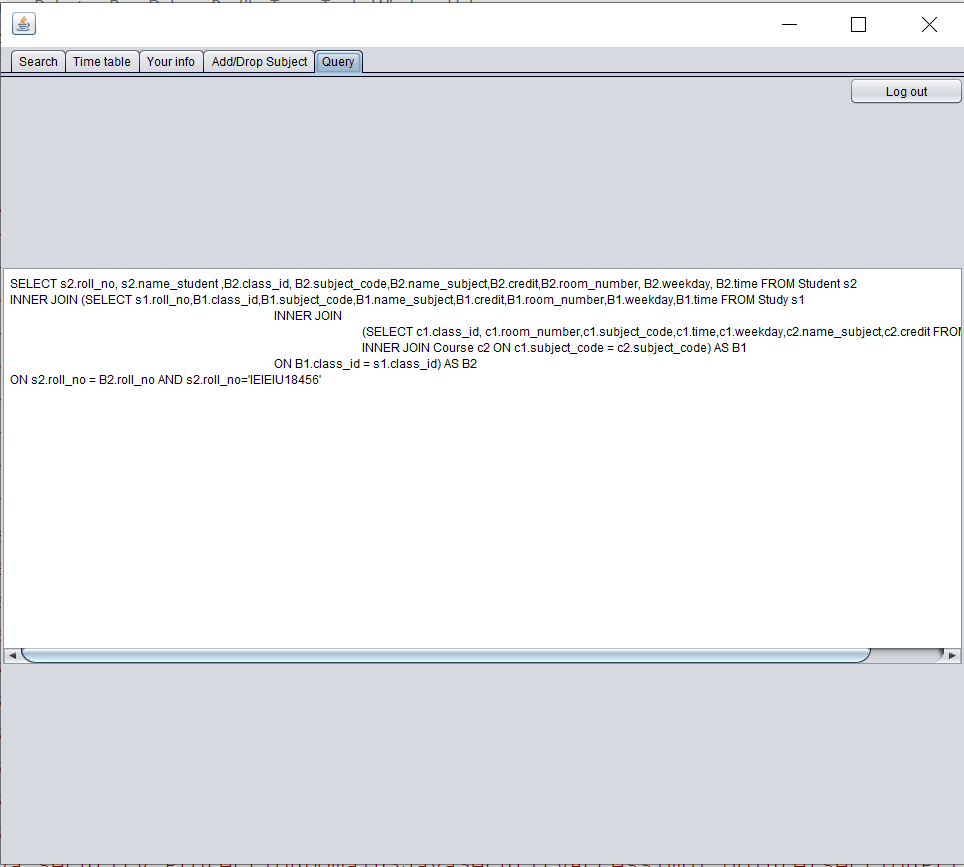






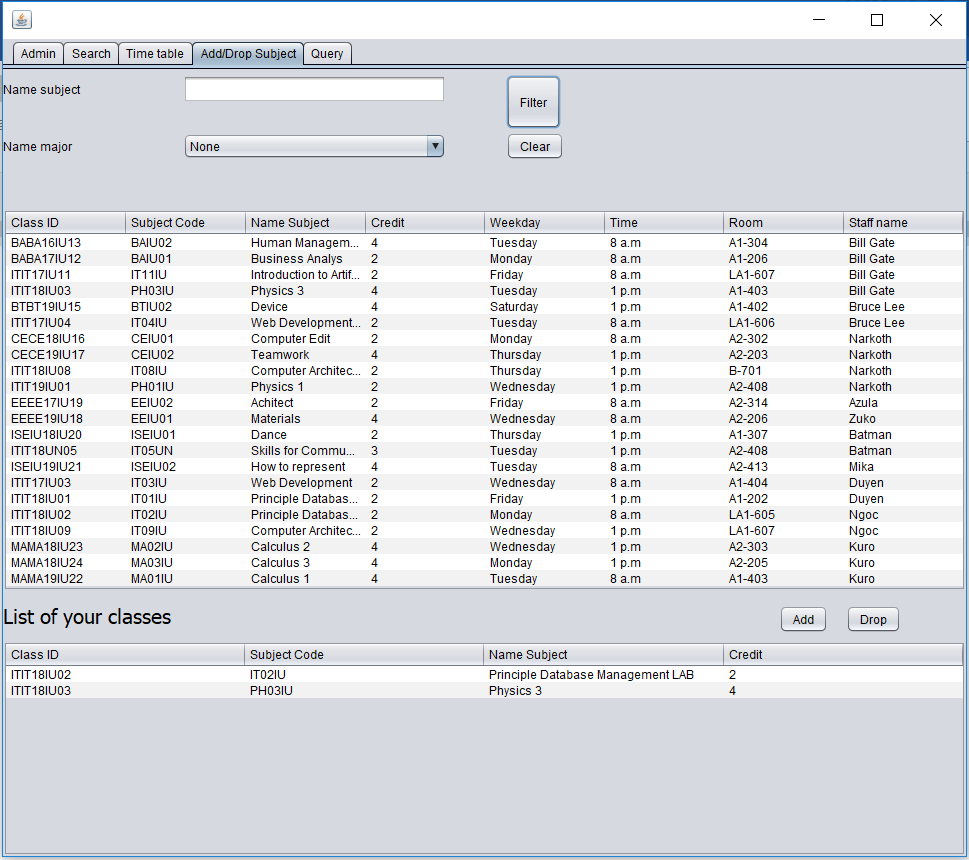
Tab query is for displaying the query for each button that the user click and the system run this query

For example, click the “Tuition” button to see the tuition fee of student



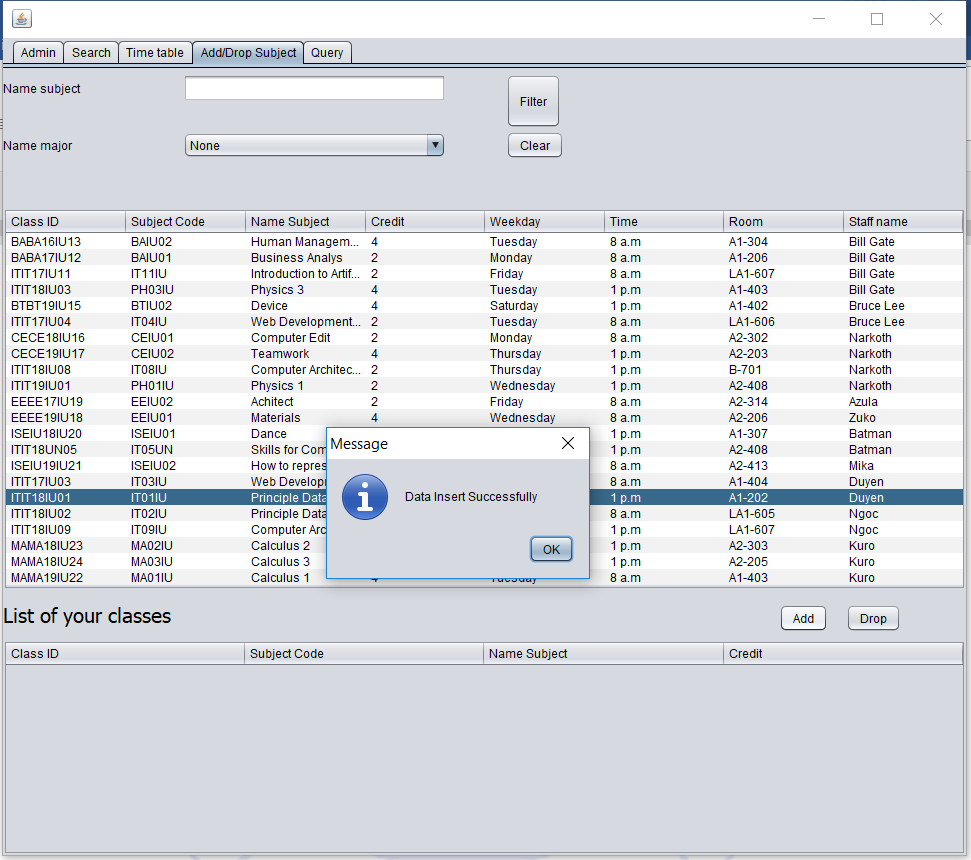
In tab “Add/Drop Subject, the system allow admin to add or drop subject of student/staff

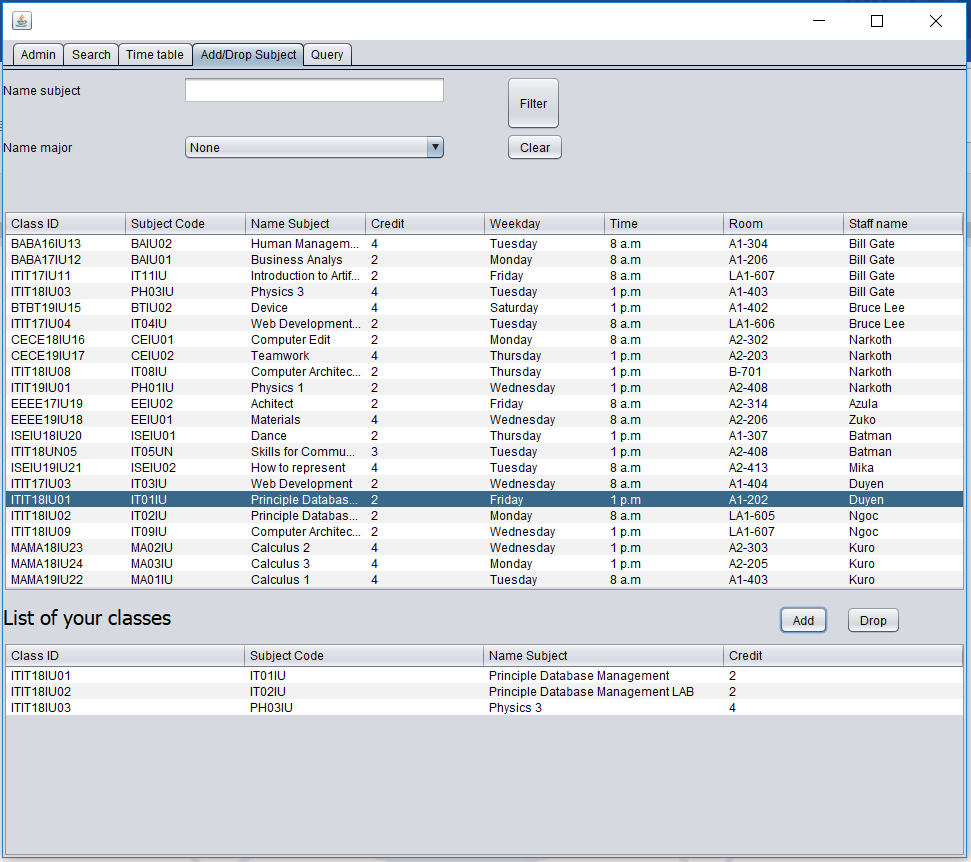
Input the keyword into the textField Name subject or choose Name major, then click “Filter” button. It will search for the data. If the textField is emptied, it will automatically display all classes in table.



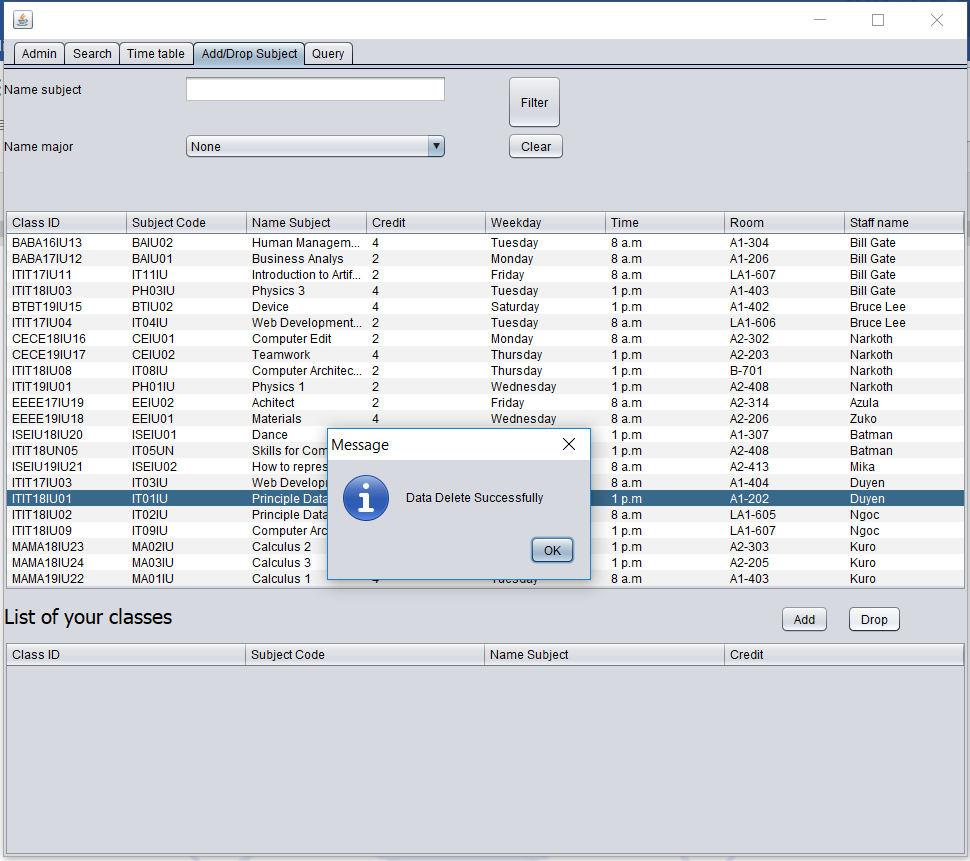
Admin is allowed to add and drop if the class schedule is not be duplicated

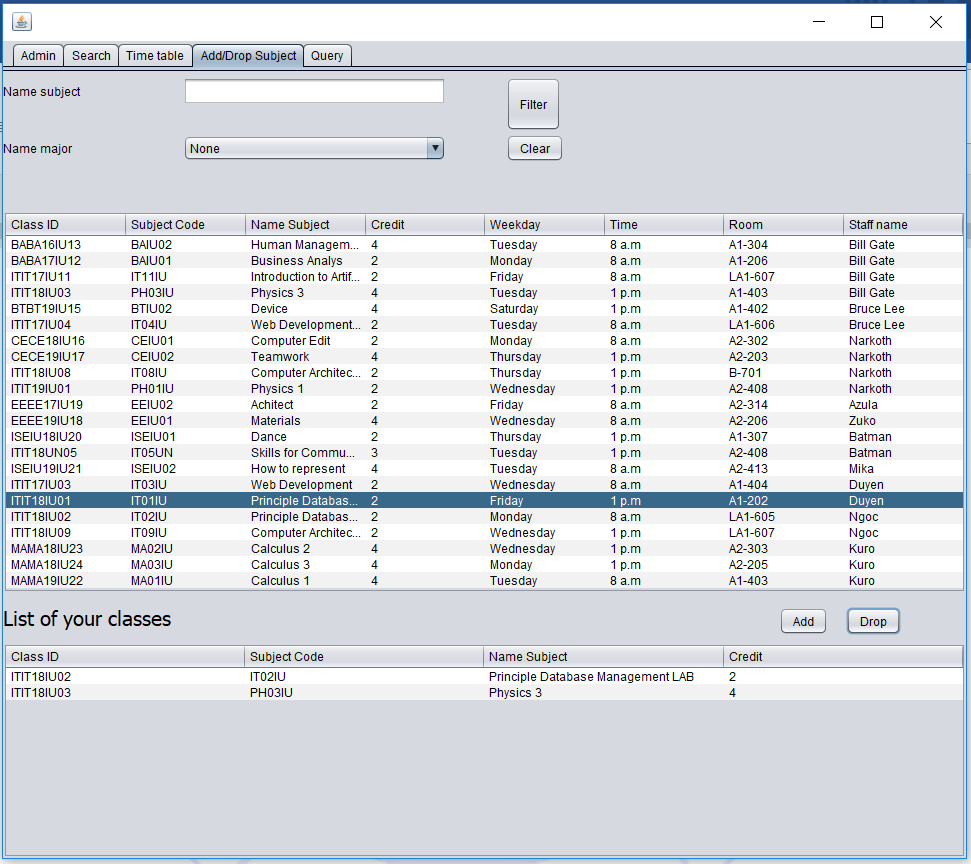
Add





* Drop





1. Conclusions

This project provides a useful tool for school or university in order to organize their students and staffs. In particularly, they can insert, delete and update their students’ and staffs’ information. The students and teachers are finding more comfortable and convenient by using the system during their studying or teaching process. Accordingly, students can be easy to register or unregister in courses, and check their school timetables and information around the study programs through the system. Also, staff has a tool to keep track with their working schedule.

**REFERENCES**

* SQL:
  + - <https://www.w3schools.com/sql/>
* GUI (java swing):
  + - <https://www.javatpoint.com/java-joptionpane>
    - <https://www.youtube.com/watch?v=QKsfHqu4Pps>
    - <https://www.youtube.com/watch?v=WIMojkwMTa0>