**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** 1](#_Toc40302999)

[**a.** **My Project** 2](#_Toc40303000)

[**b.** **Tasks** 2](#_Toc40303001)

[**2.** **ENTITY-RELATIONSHIP MODELING** 2](#_Toc40303002)

[**a.** **Entity and attribute** 3](#_Toc40303003)

[**b.** **Explaination** 4](#_Toc40303004)

[**3.** **RELATIONAL MODEL** 5](#_Toc40303005)

[**4.** **DATABASE** 5](#_Toc40303006)

[**5.** **GUI** 9](#_Toc40303007)

[**REFERENCES** 9](#_Toc40303008)

1. **INTRODUCTION**
   1. **My Project**

* We have implemented a project on the topic School Management System
  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

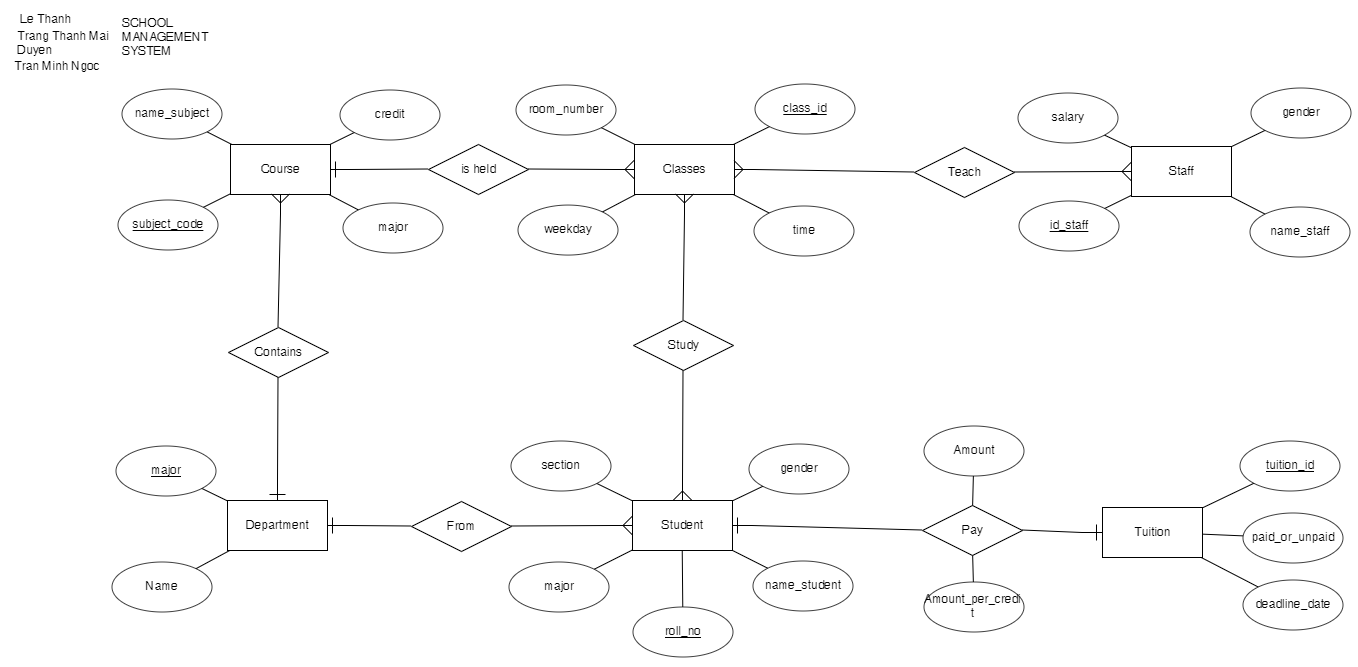
• Next, we base on ERD and draw Relational Model

• 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** |
| Create table | X |  |  |
| Insert data | X |  | X |
| Draw ERD |  |  | X |
| Draw Relational Model |  |  | X |
| Write query retrieving data | X |  |  |
| Find the reference | X | X | X |
| Design GUI |  | X |  |
| Code the GUI function |  | X |  |
| Write proposal | X |  |  |
| Write report |  |  | X |
| Check grammar/edit report | X |  |  |
| Fix bug | X | X |  |

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. **Explanation**
   * 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)
   * 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)

* + Function of 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

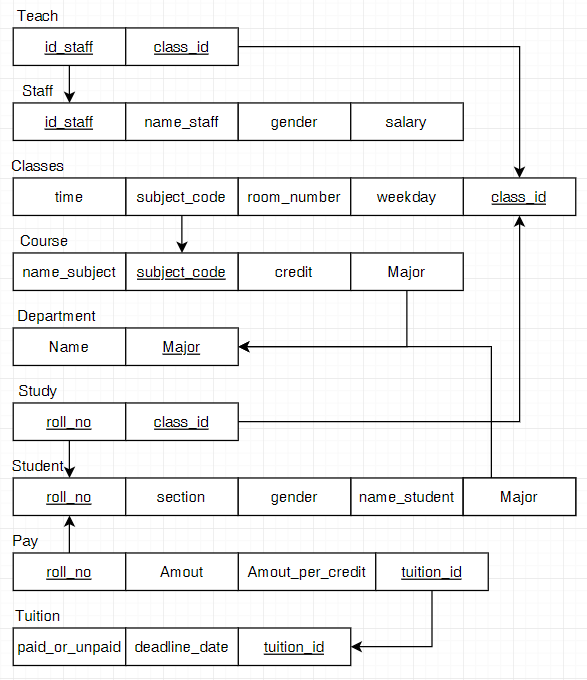
• Allow admin to handle and edit data in 8 tables (except Account and Department)

• 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. **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(PK)

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

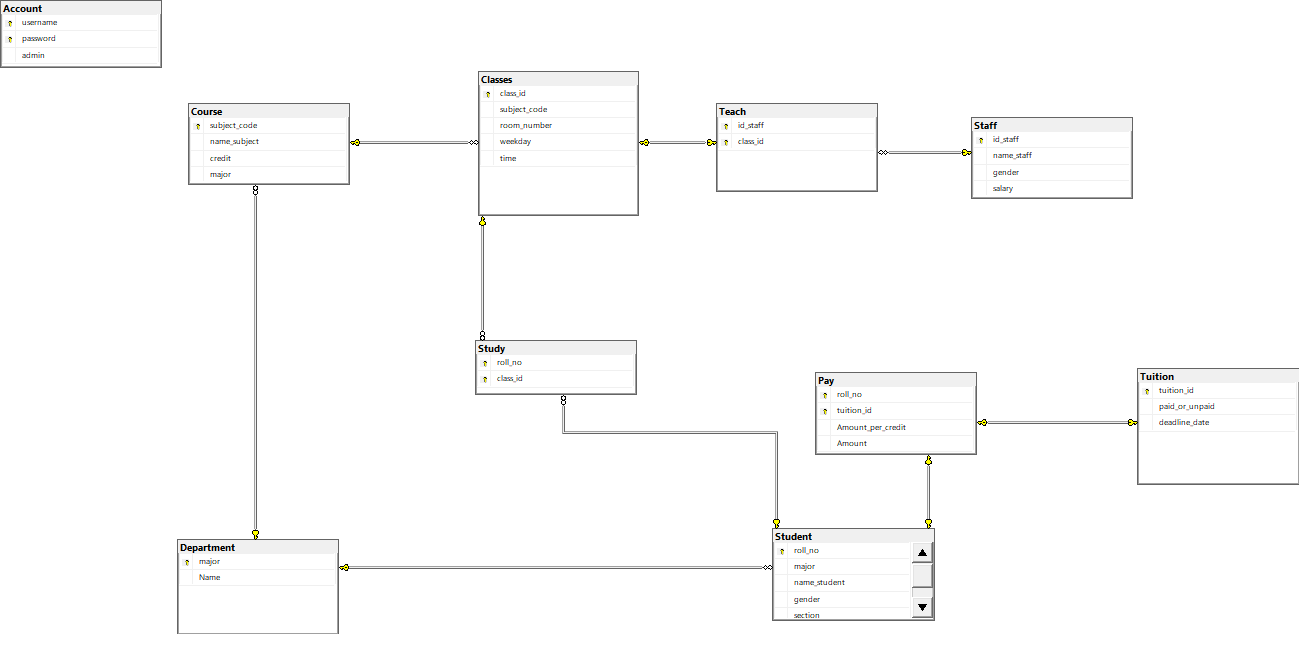
**•** Teach([id\_staff(FK),class\_id(FK)])

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

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

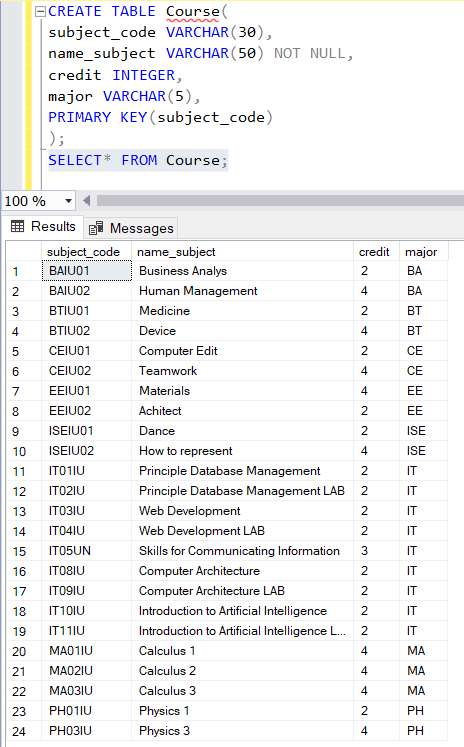
**•** Tuition(fee, paid\_or\_unpaid, tuition\_id)

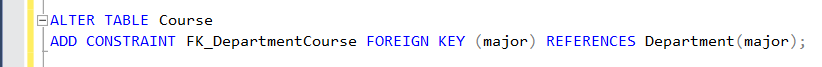
**•** Department(major, Name)

1. **DATABASE DIAGRAM**
2. **DATABASE**
3. **Table**

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

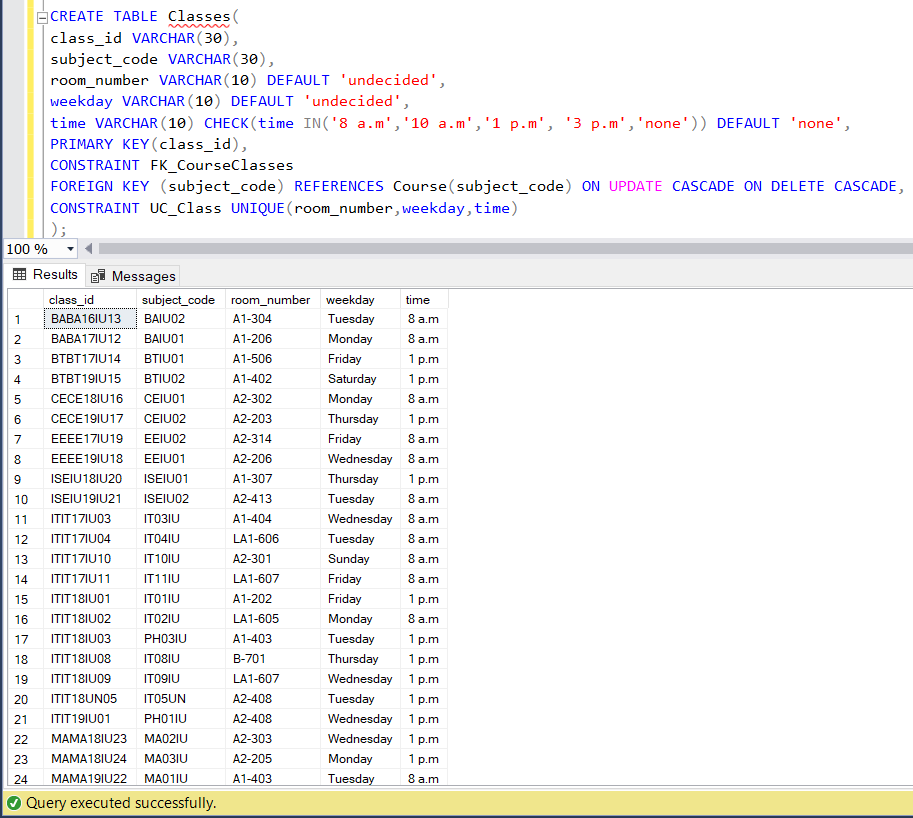
**• Course**





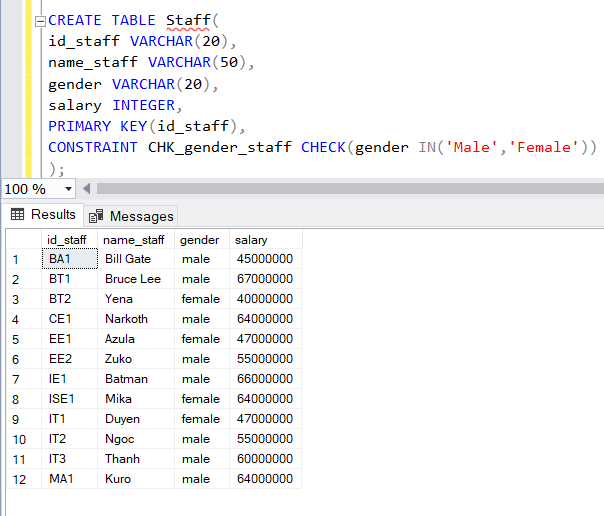
* Subject\_code and name\_subject represented for the id and the name of course in type varchar with maximum size, credit is in integer type
* Set the primary key for subject\_code (cannot be duplicated), and it can be used as foreign key in another table that has relationship with this table
* Attribute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

**• Classes**



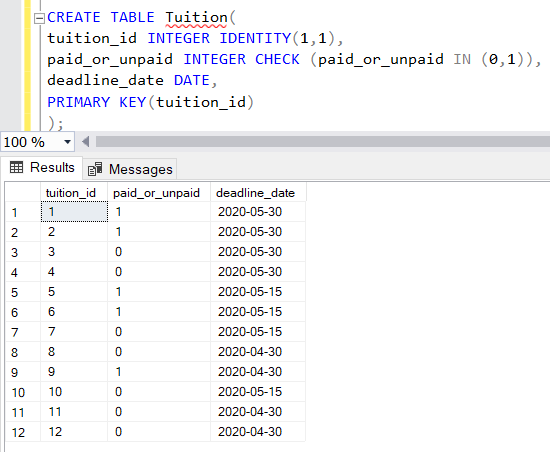
* In this table, the subject\_code has the foreign key from subject\_code of table “Course”
* Room\_number and weekday are type varchar and set if the user skip input this attribute, it will automatically display “undecided”
* In attribute times, it is static in 4 particular times (8a.m, 10a.m,1p.m,3p.m) 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**



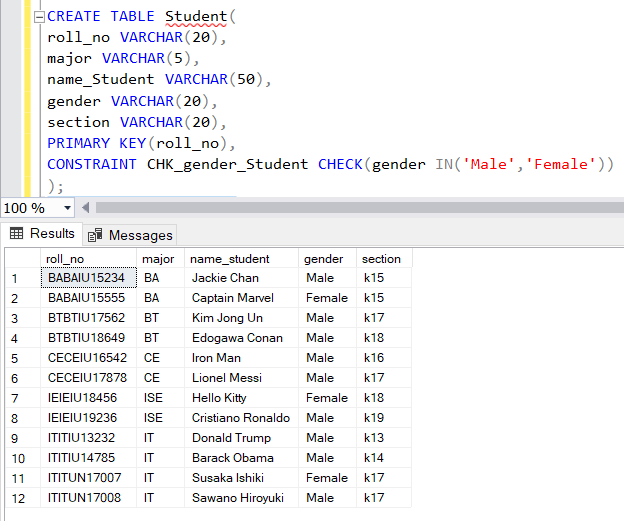
* Id\_staff, name\_staff and gender are in type varchar, and attribute salary in is integer
* Set the primary key for id\_staff and other tables can use it as foreign key if it has a relationship with “Staff”
* And the gender must be “Male” or “Female”

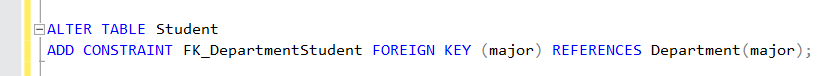
• **Tuition**



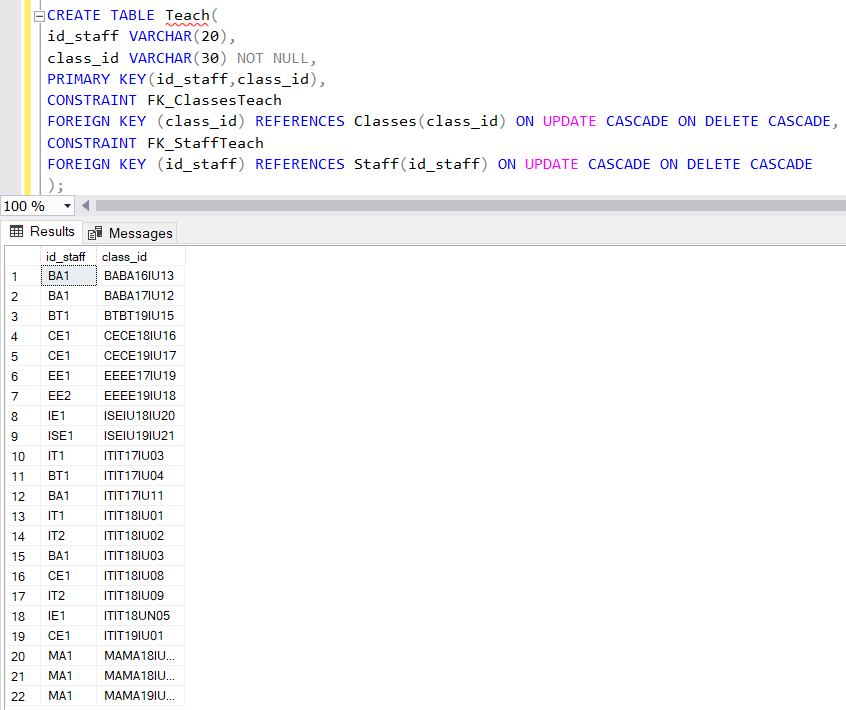
* Tuition\_id is declared in type varchar,both fee and paid\_or\_unpaid in integer and set if it is paid,it will display “1”;otherwise, unpaid “0”
* Set the primary key for tuition\_id and it can be used as foreign key in another table which has a relationship with “Tuition”

• **Student**



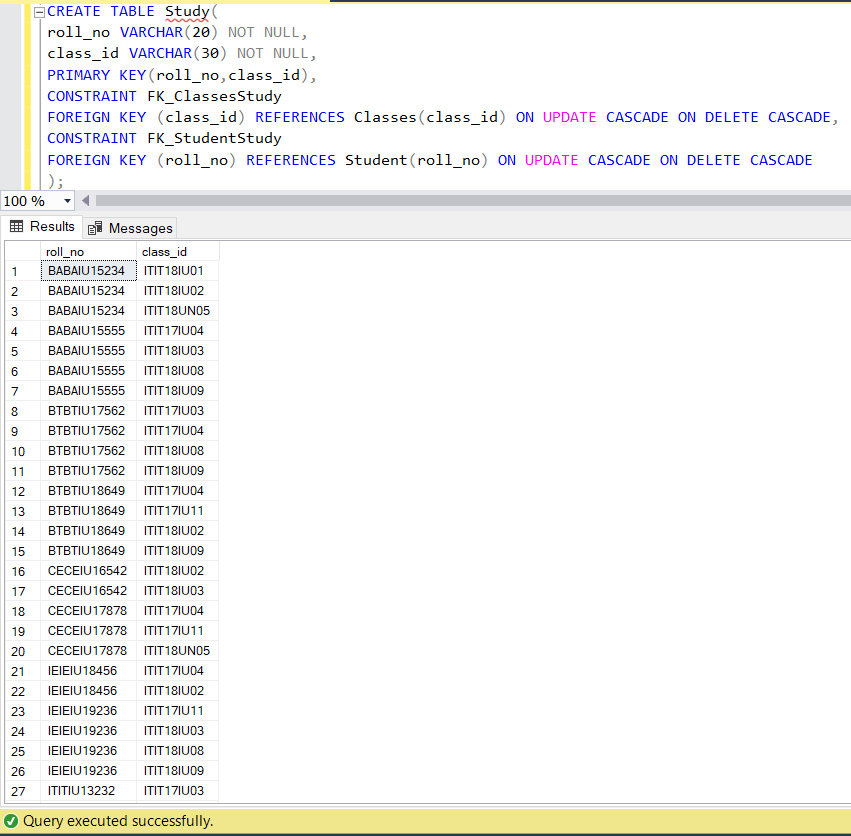


* Roll\_no, section, name\_student, gender are varchar datatypes with different lengths
* Set the primary key for roll\_no and it can be used as foreign key in another table which has a relationship with “Student”
* And we set gender must input as “Male” or Female”
* Attribute tuition\_id and major are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



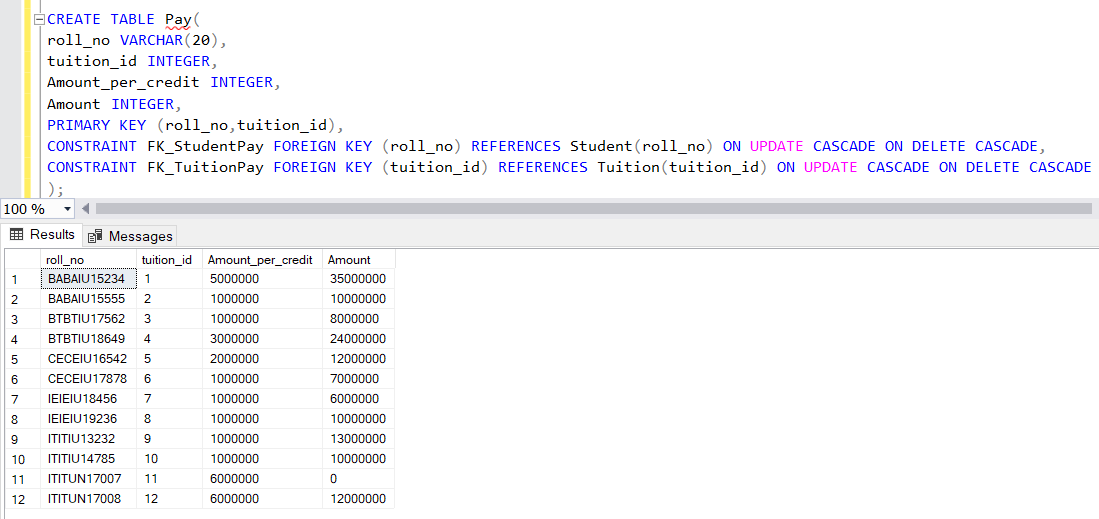
* Attribute class\_id is foreign key from class\_id of table “Classes”
* Attribute id\_staff is foreign key from id\_staff of table “Staff”

**• Study**



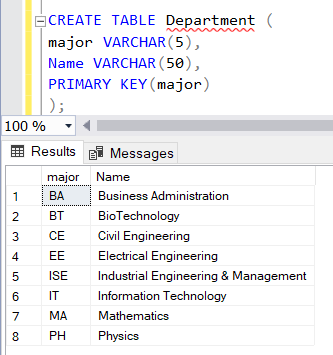
* Attribute roll\_no is subset of roll\_no from table “Student”
* Attribute class\_id is subset of class\_id from table “Classes”

• **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**

****

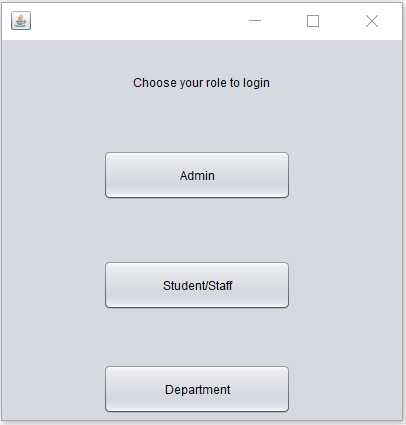
* Attribute major, Name are in type varchar
* Set the primary key for major and it can be used as foreign key in other tables which is relation to “Department”

1. **Explaination**

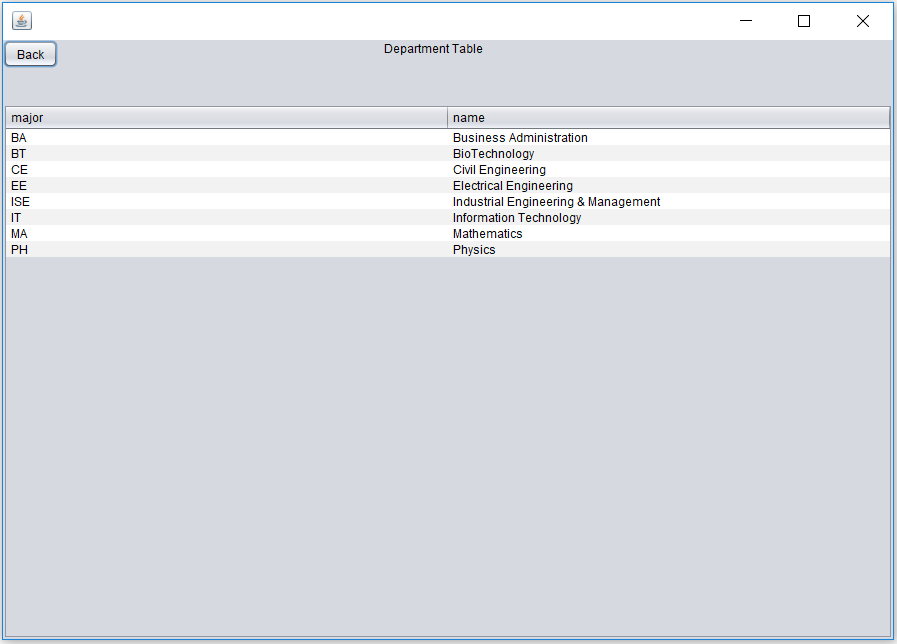
* In the system, a Course has 4 attributes: subject\_code, name\_subject, credit, major (FK).
* A course can be held in many classes, which has 5 attributes class\_id, subject\_code ,room\_number, weekday, time and those cannot be duplicated by using primary key for class\_id, and UNIQUE (room\_number,weekday,time).
* A student is studying in classes (a student can attend to many classes, a class can be attended by many students), and the Student has 5 main attributes: roll\_no, section,major(FK) ,name\_student, gender.
* Staff has 4 basic attributes: id\_staff,name\_staff,gender and salary, he/she teaches classes (a staff can teach many classes, a class can be taught by many teachers)
* Tuition has 3 basic attributes (each student has 1 tuition): tuition\_id, paid\_or\_unpaid , deadline\_date. The deadline\_date, and paid\_or\_unpaid, which shows the payment status of 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 support the function: the student cannot see his/her timetable, if he/she had not paid his/her tuition fee before the deadline date.
* Department has 2 attributes ( each department can has many students and Courses): major , Name( full name of major)

1. **GUI**
2. **Login**

* 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)

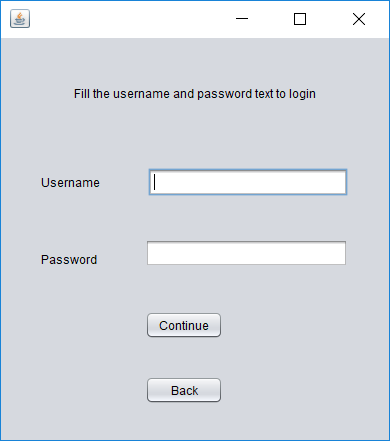


* We start with button “ Department”. It is not required to log in and it shows all majors that the school has

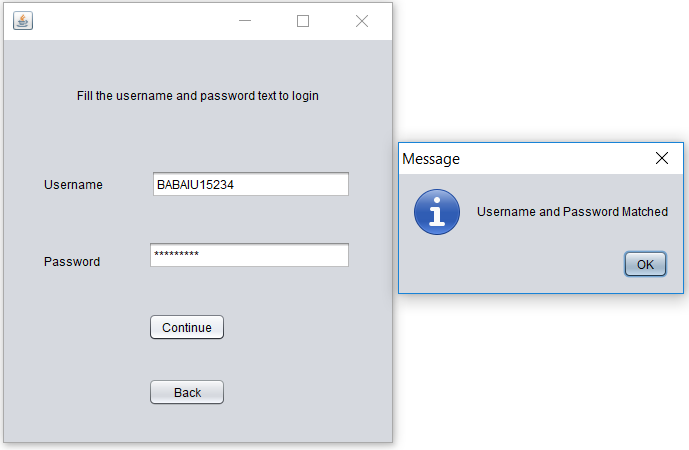
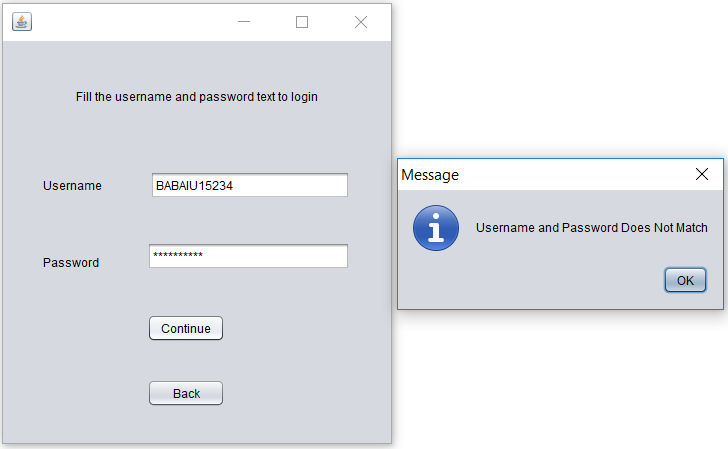


1. **Student/Staff**

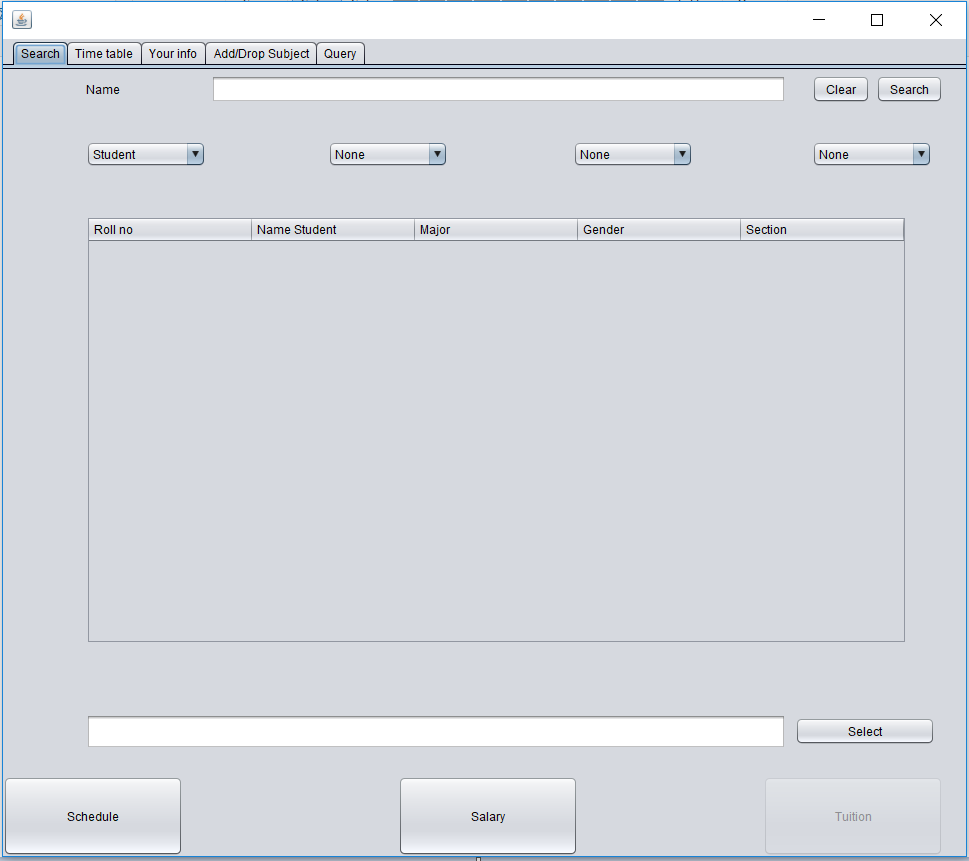
* Then when user access to Student/Staff, it will display



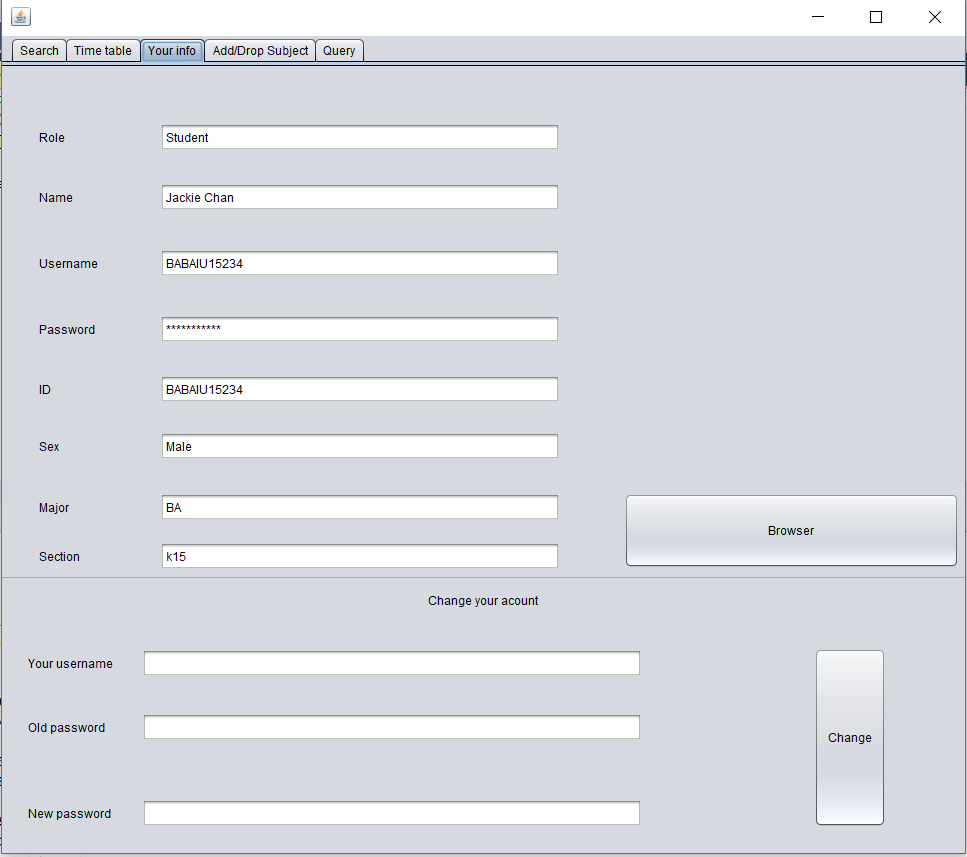
* Use the accounts that are stored in database (table “Account”) to login into the system. There are 2 possible cases can be happened: login successfully or fail



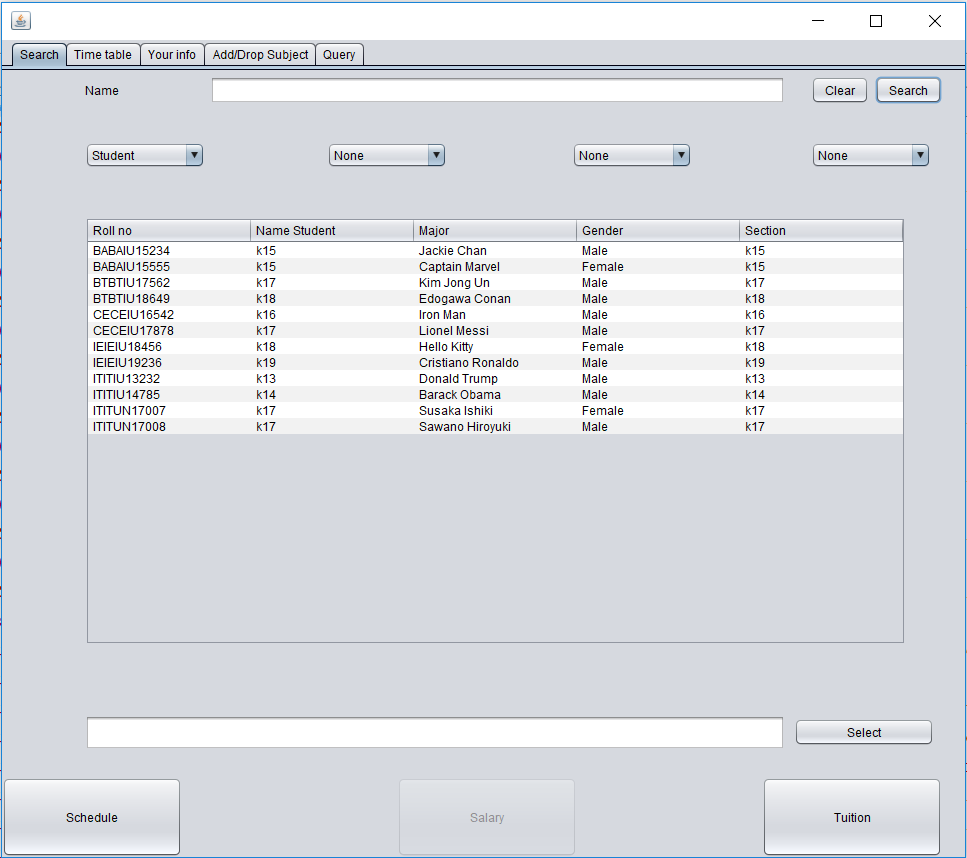
* When the user access successfully, this table will display



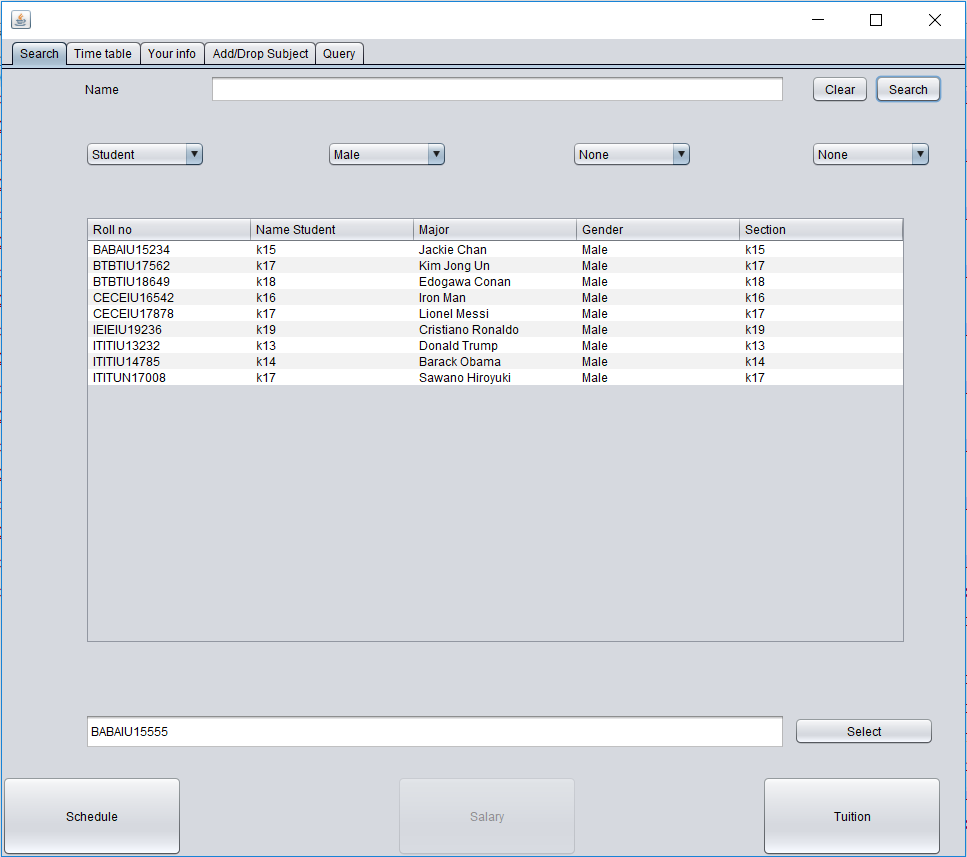
* First, choose the function in this table in order to use
* This interface has 5 tabs: Search, Time table, Your info, Add/Drop Subject, Query
* Access “Your info” tab to see the info of student that logged in



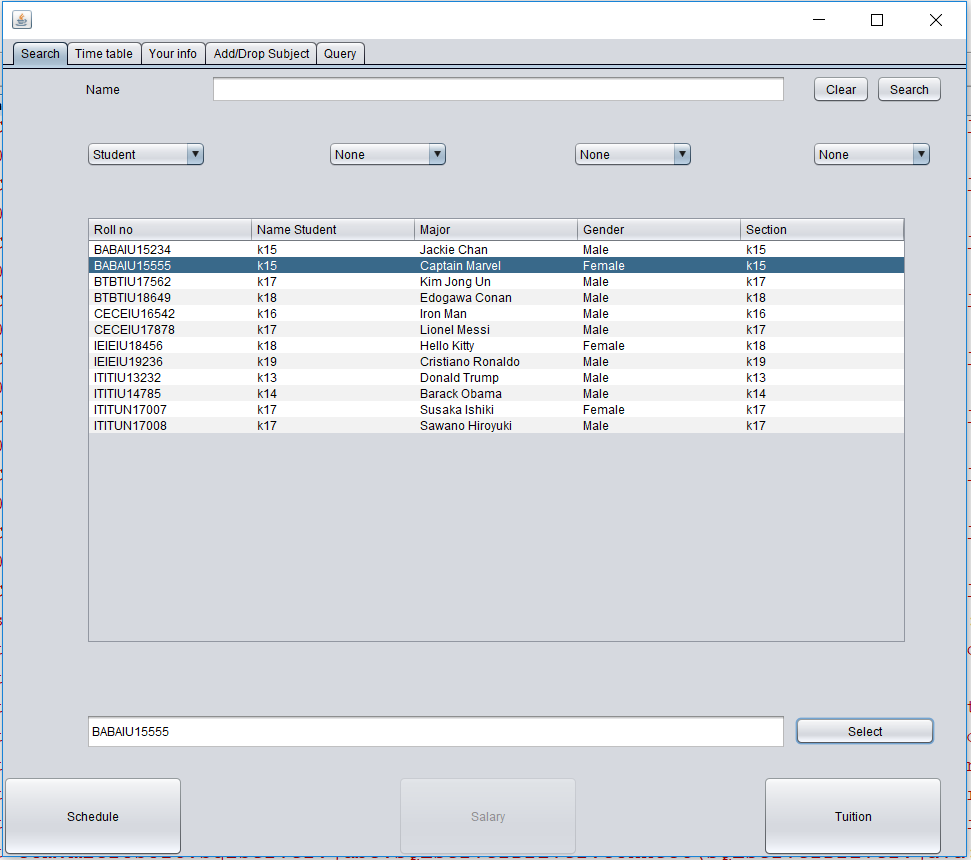
* Input the keyword into the textField name, then click search button. It will search for the data. If the textField is empty, it will automatically display all students in table.

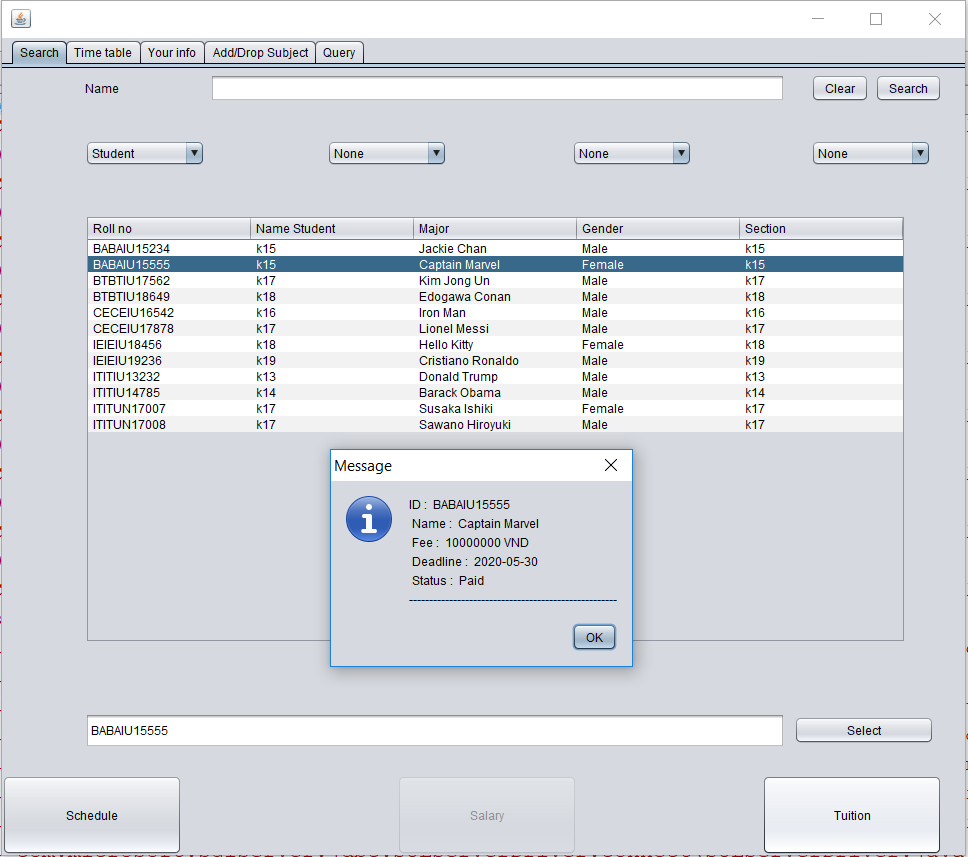


* 4 checkboxes below the name textfield represent for 4 attributes of a student that the user want to search ( system can know at least 1 and max 4 to search, example the user want to search all student that are male)

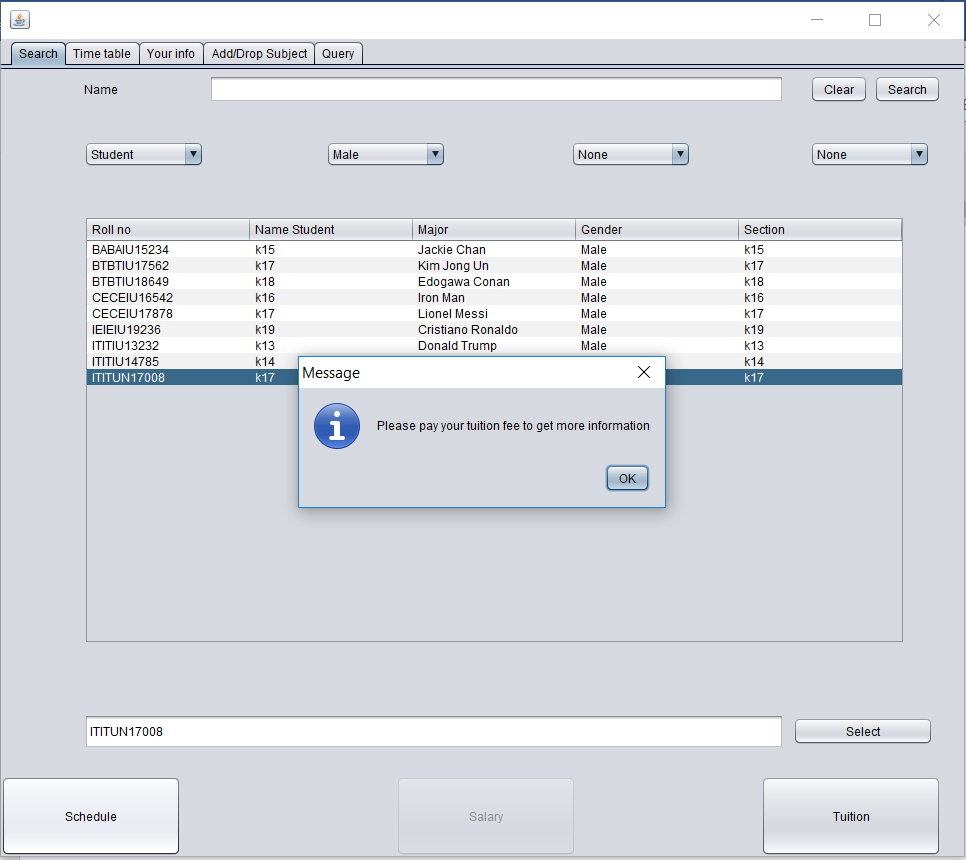


* Select button will help the system to lock to student that the user want to see this student’s schedule or tuition fee
* After clicking select button then if user want to se the tuition of his/her, then click the “Tuition” button

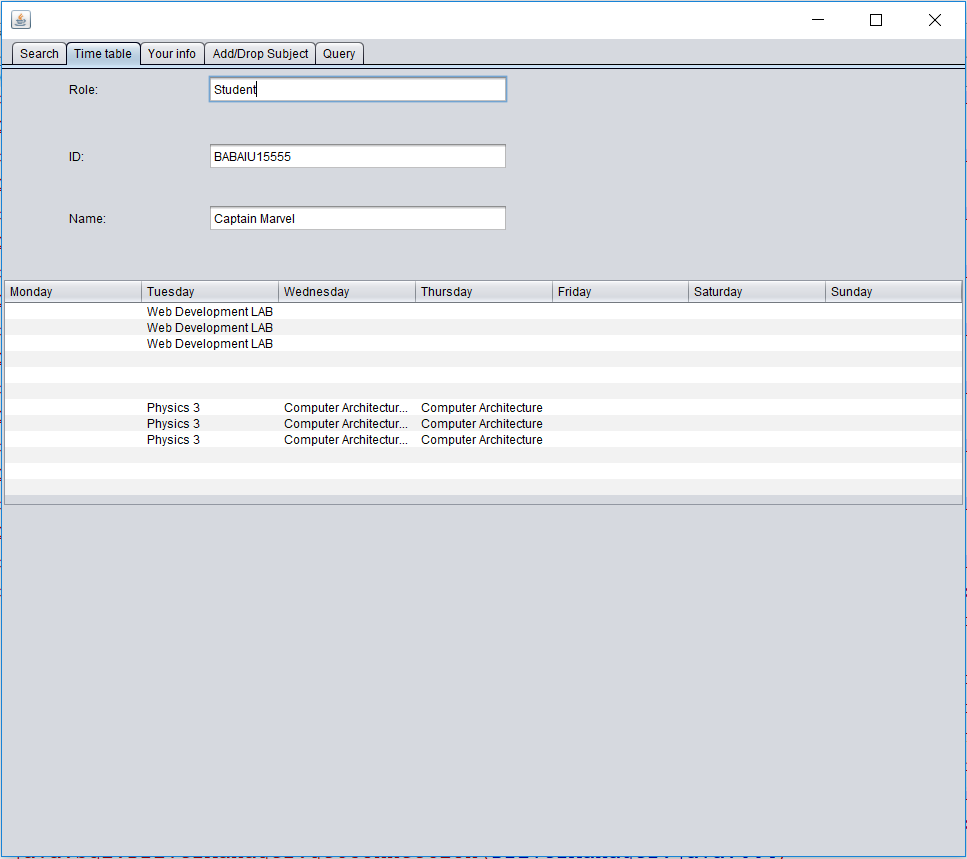




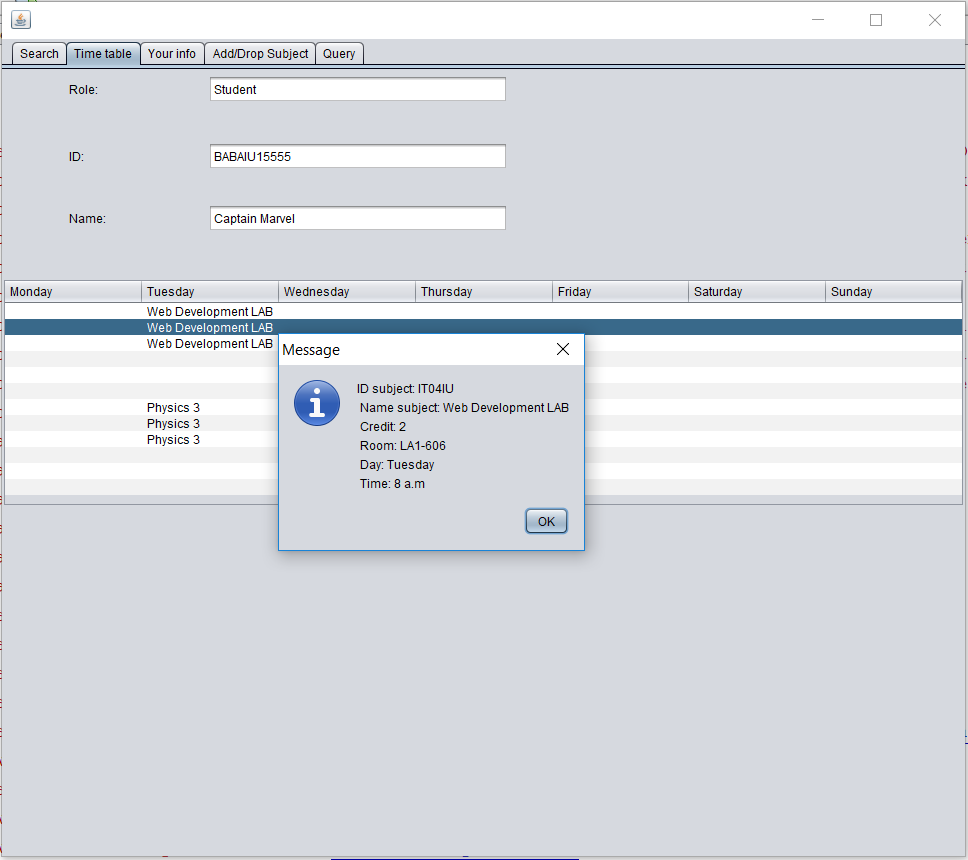
* Then if user want to see the schedule of this student, user must check if his/her has already paid or not, if not the system don’t allow user to see the schedule



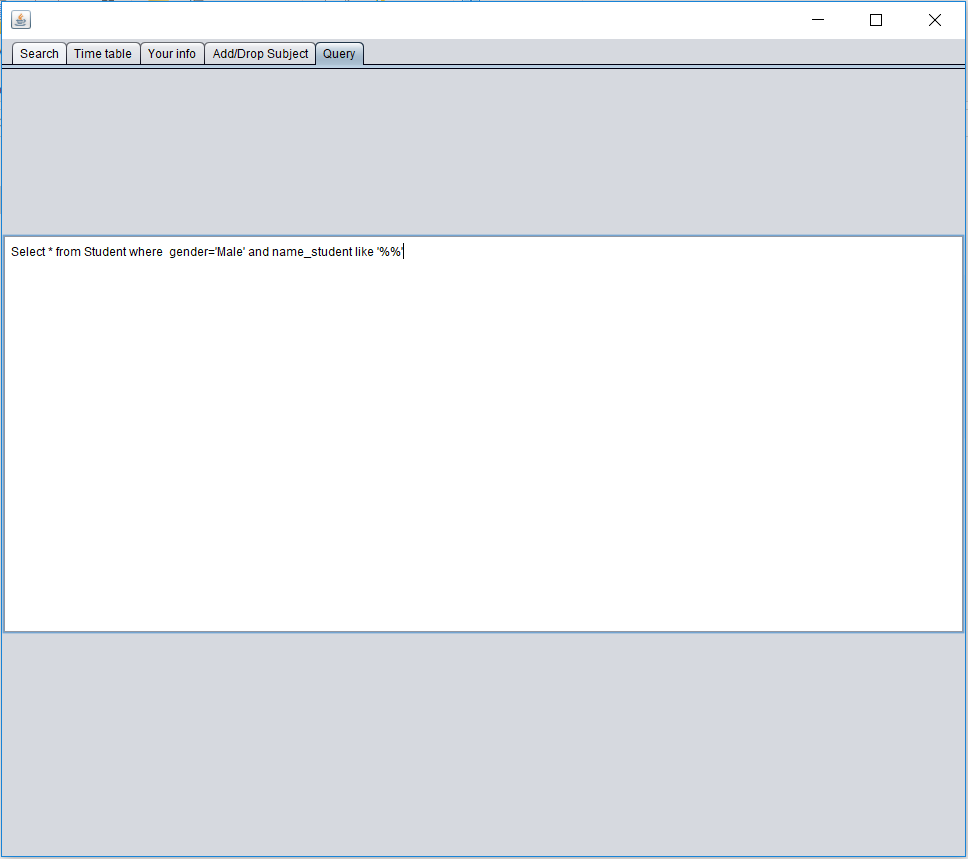
* And if he/she paid we click the schedule button and go to the next “Time table” tab, it will display his/her information and schedule



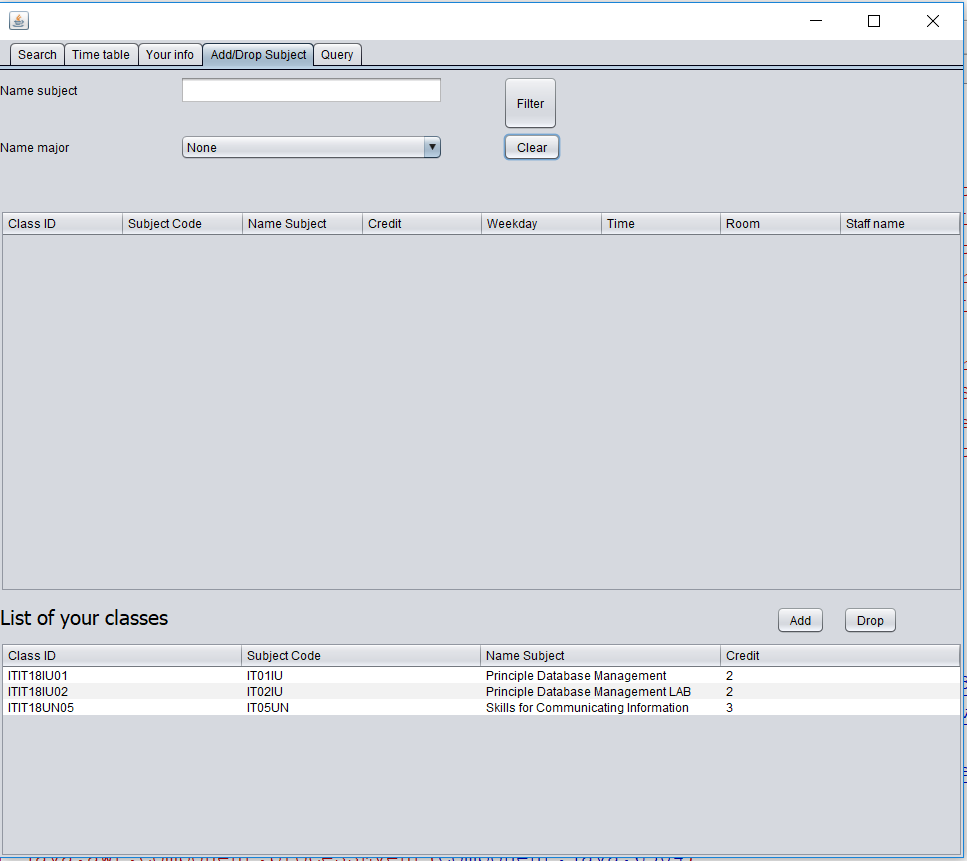
* In case that student used his/her account to log in and paid fee, just click the button “Schedule” then move to the “Time table” tab to see the schdule
* The user also can know more information about the course by clicking in the course



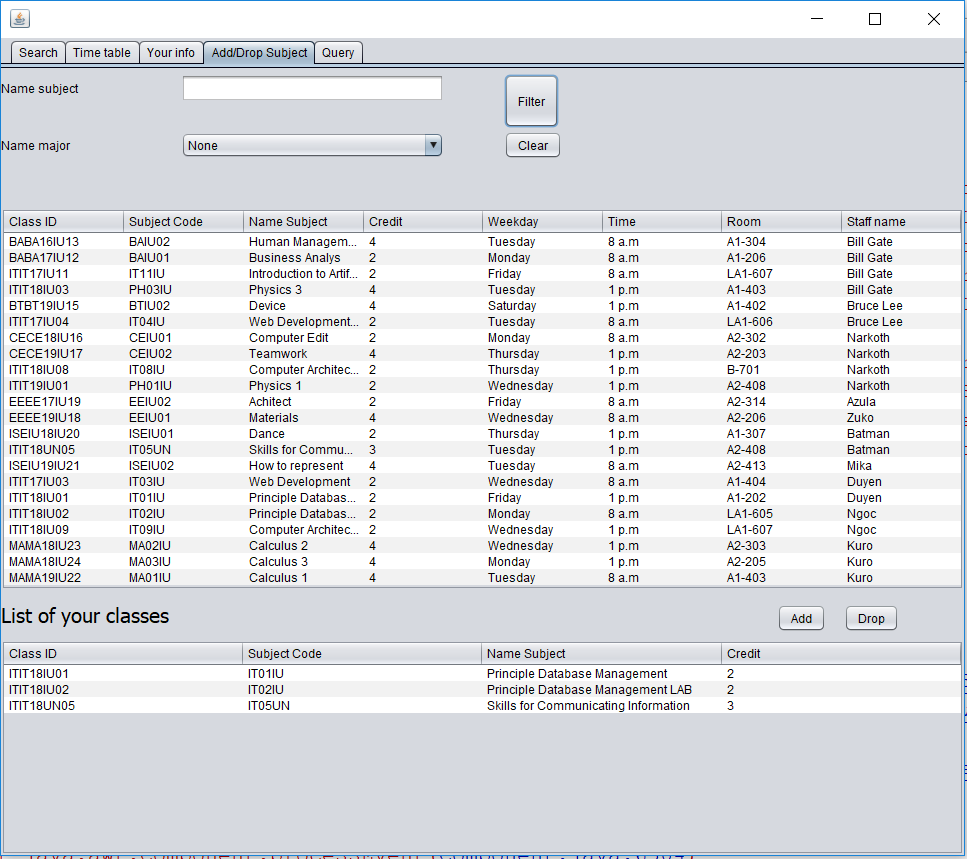
* Tab “query” display the code for all action user do, such as click button, each button has its own code so this tab will print it into to screen to give us more comfortable to handle
* Example when user choose attribute male so that he/she wants use search button to search all the student that is male



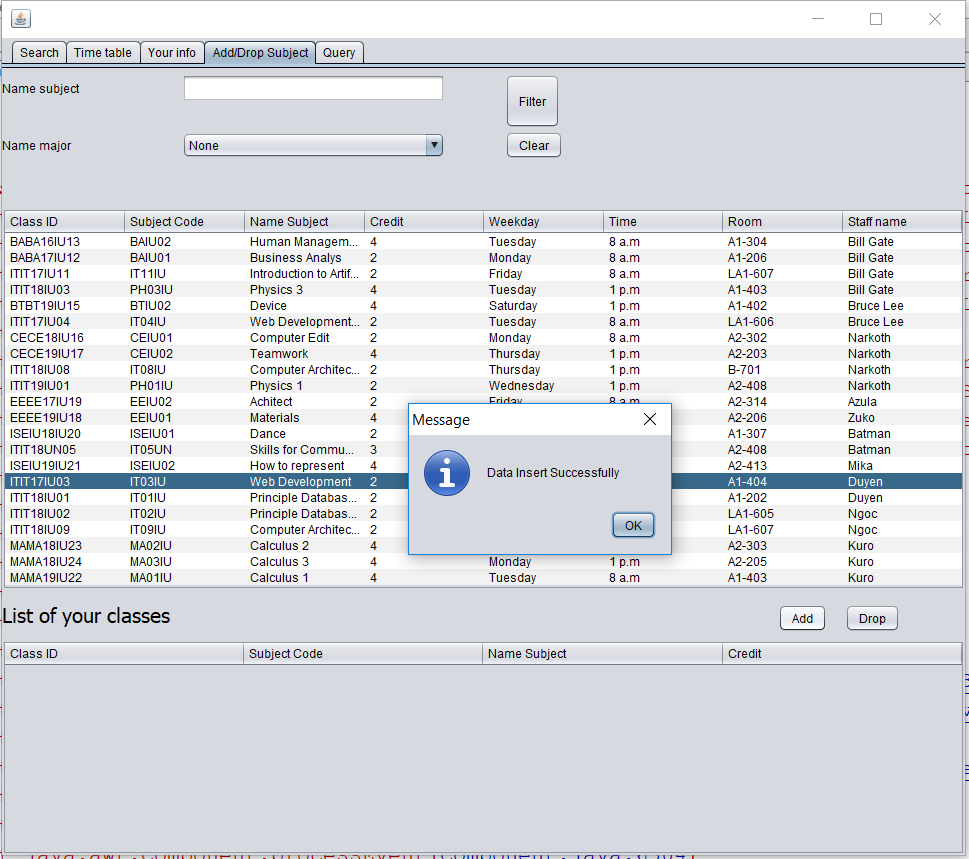
* In tab “Add/Drop Subject” system allow students to add and drop their classes

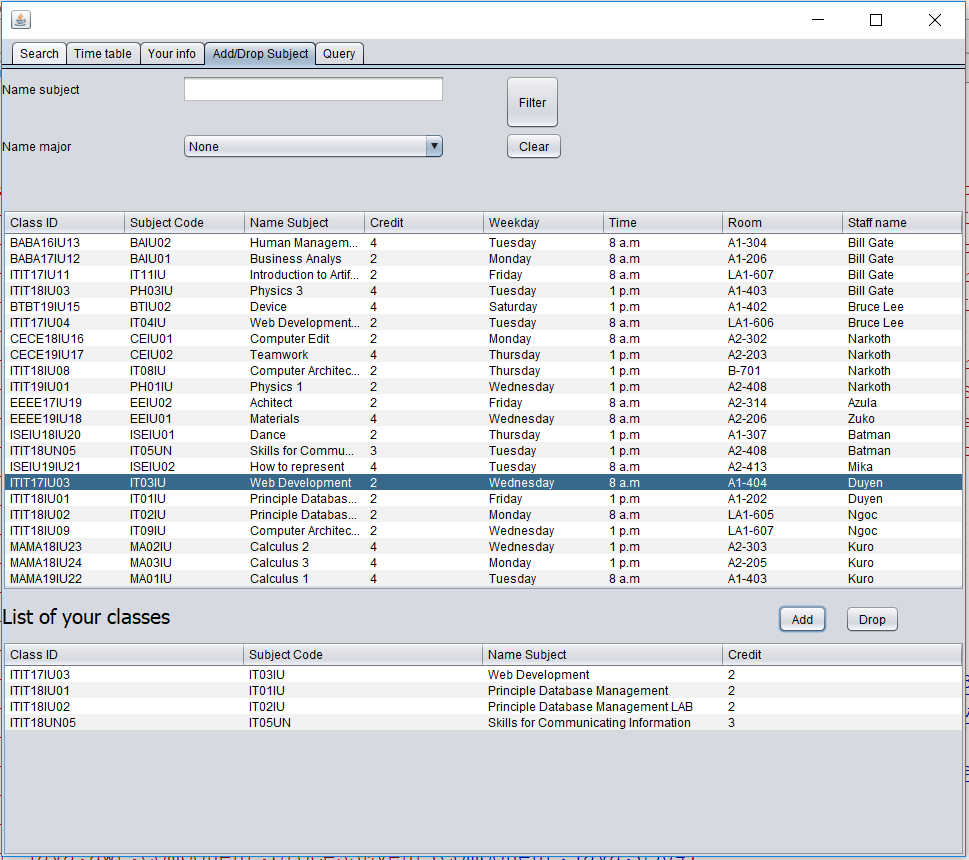


* Input the keyword into the textField Name subject or choose Name major, then click “Filter” button. It will search for the data. If the textField is empty, it will automatically display all classes in table.

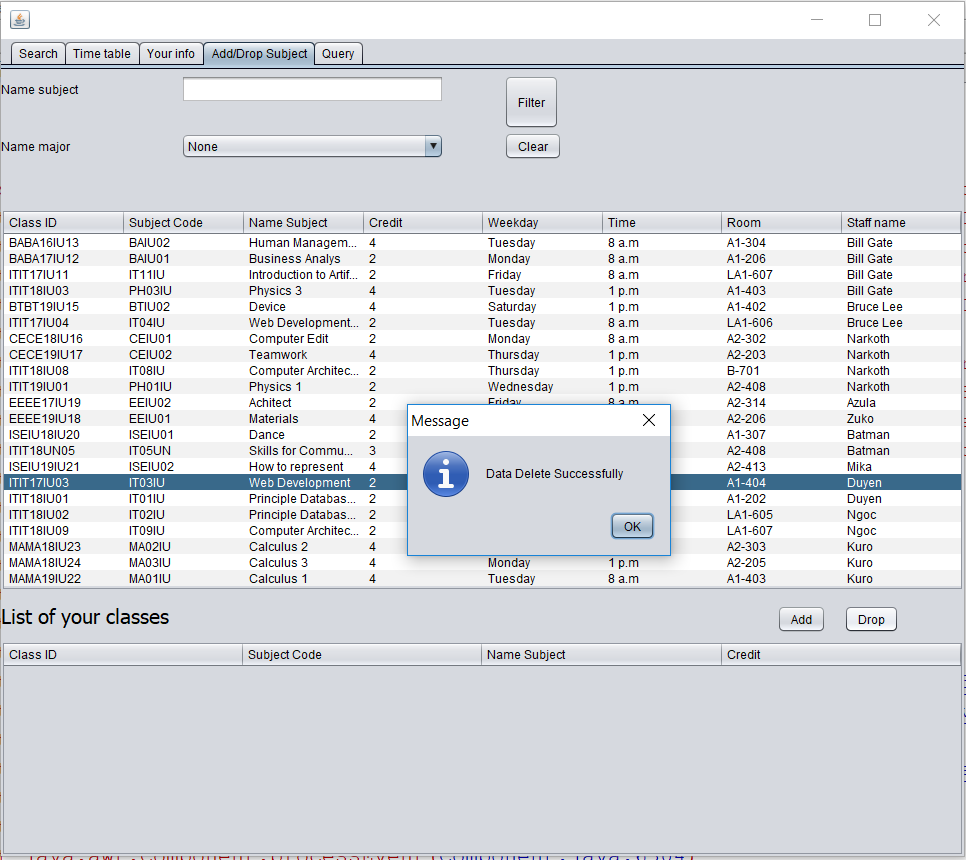


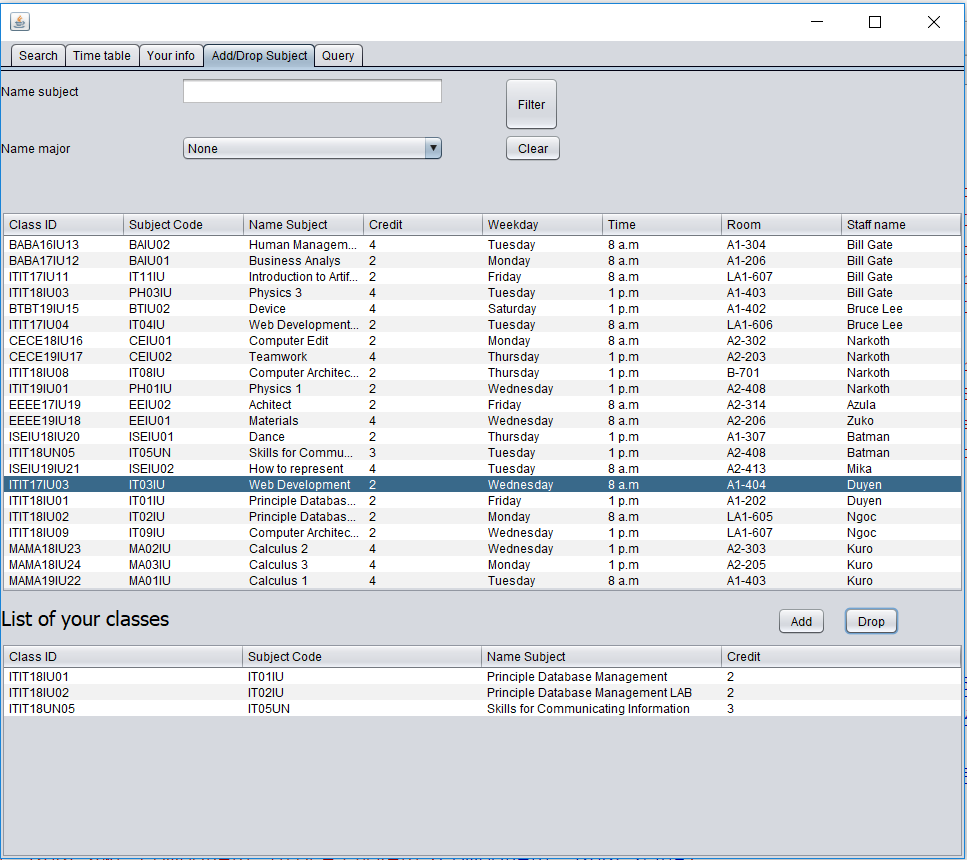
* Add





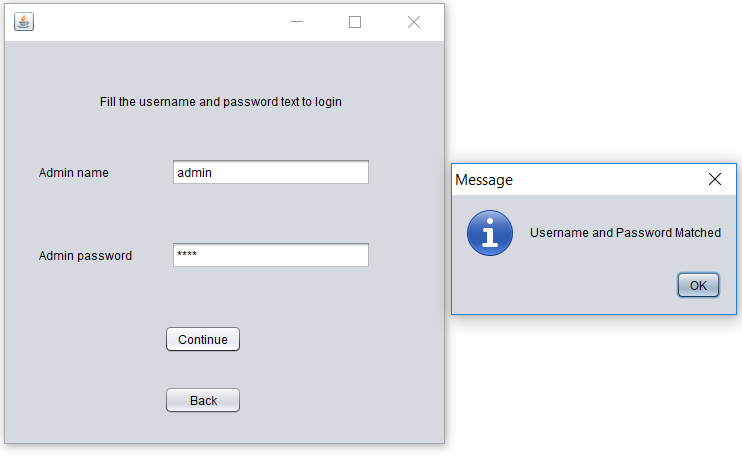
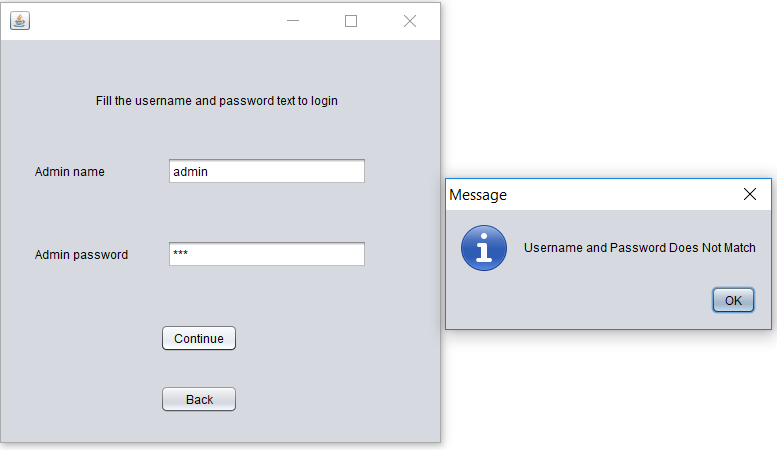
* Drop



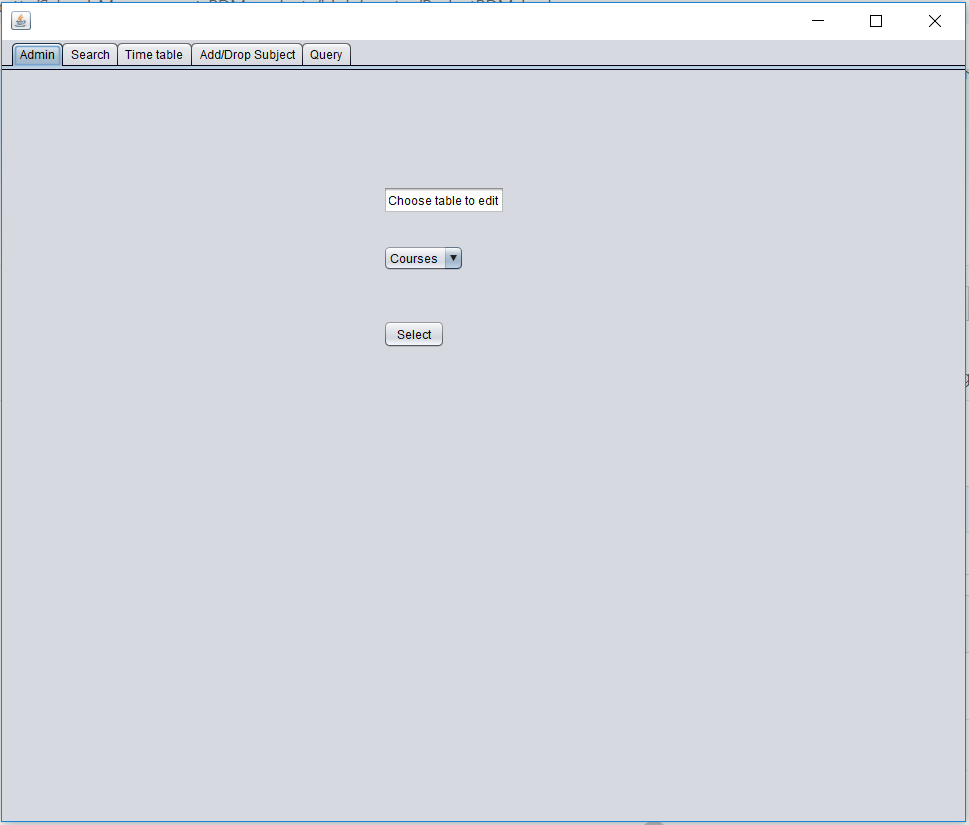


1. **Admin**

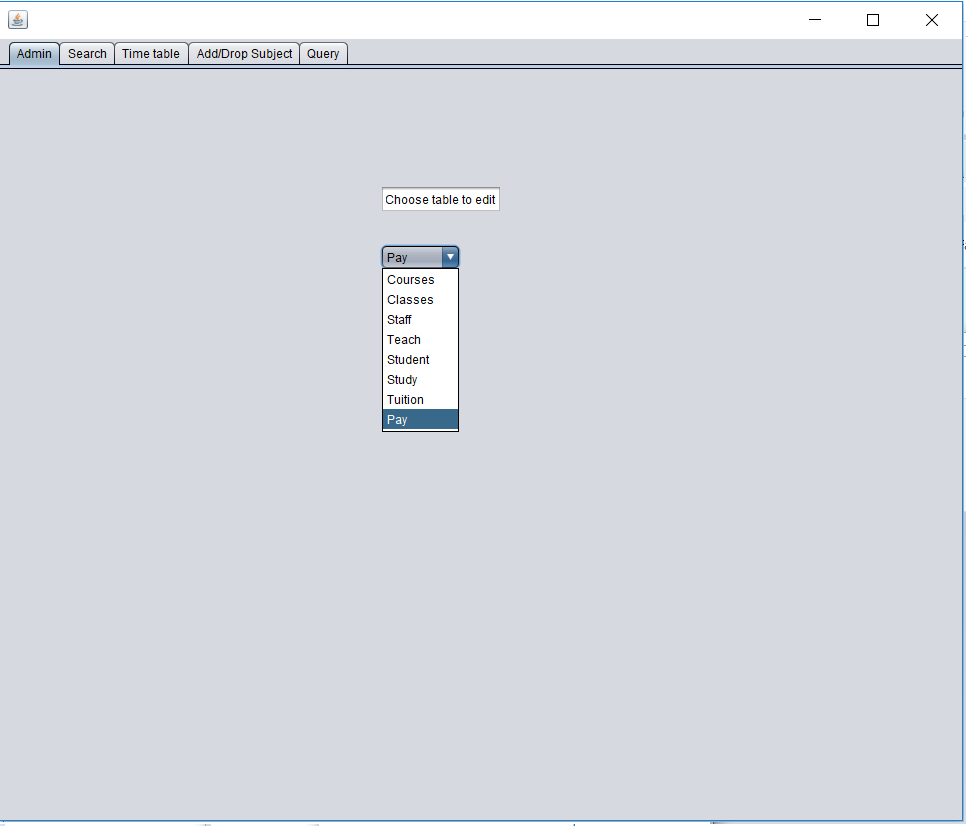
* When user access in system by login with role Admin
* Use the accounts that are stored in database (table “Account”) to login into the system. There are 2 possible cases can be happened: login successfully or fail



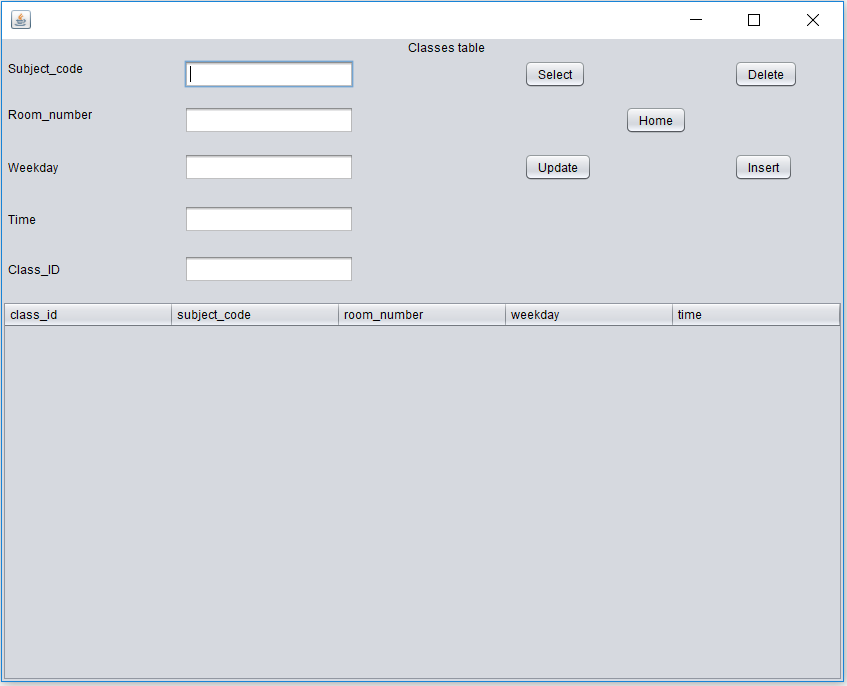
* This interface has 5 tabs : Admin, Search, Time table, Add/Drop Subject Query
* In Tab “Admin” user is allowed to edit and handle all table and data in database by accsessing to each table



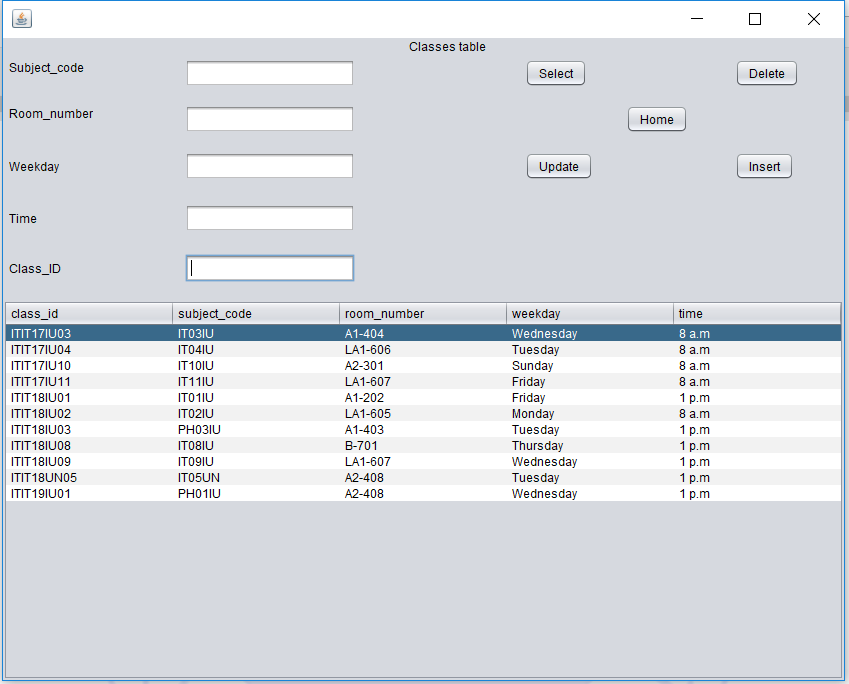
* And then choose which table he/she wants to access



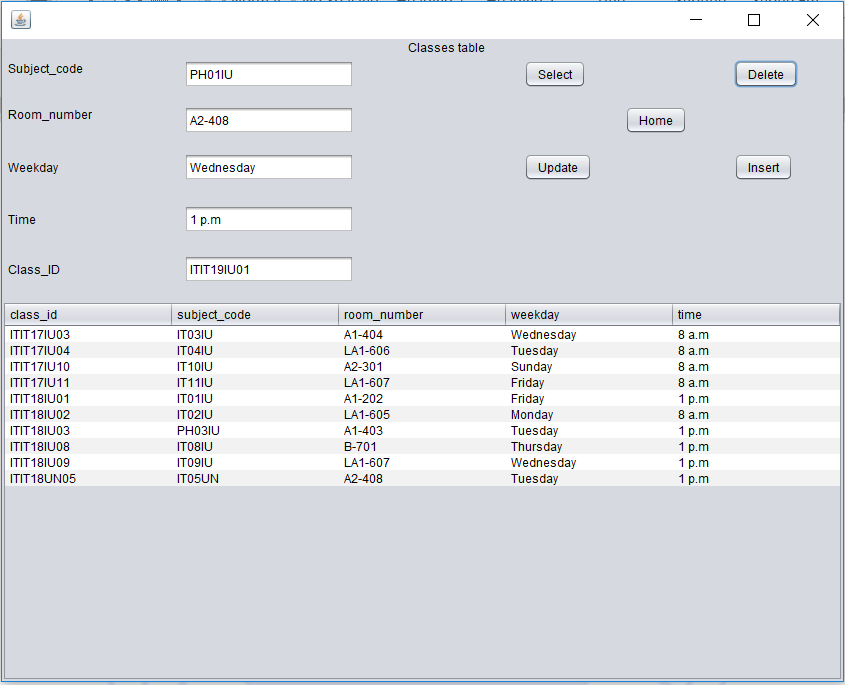
* And then click the select button to access to to jframe contains tables in database
* System has 8 tables: Courses, Classes, Staff, Teach, Student, Study, Tuition, Pay
* First starting with table Classes



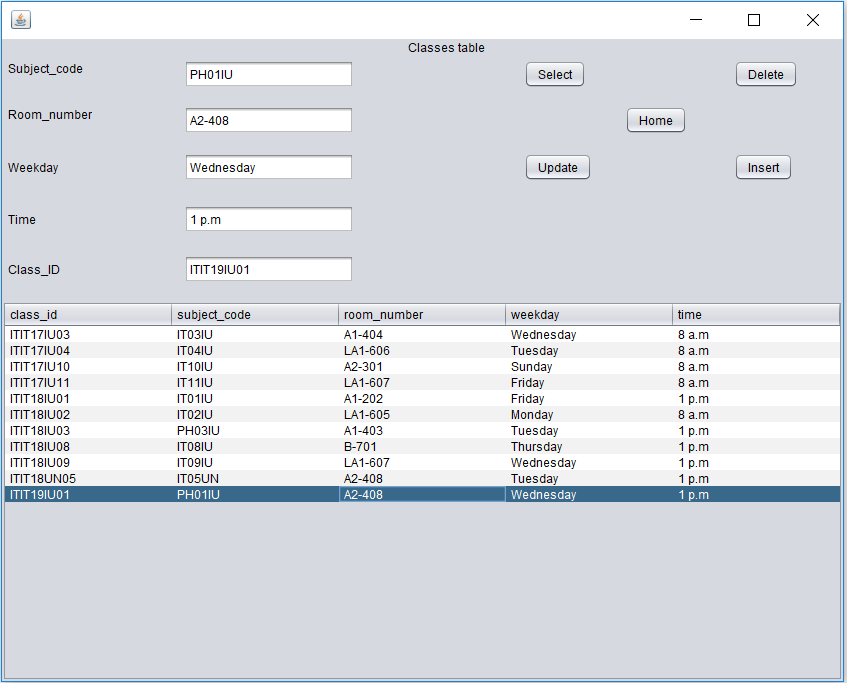
* There are 5 buttons here: Home(return), Select, Insert, Update, Delete
* When user click “Select” it will display all the classes that in database



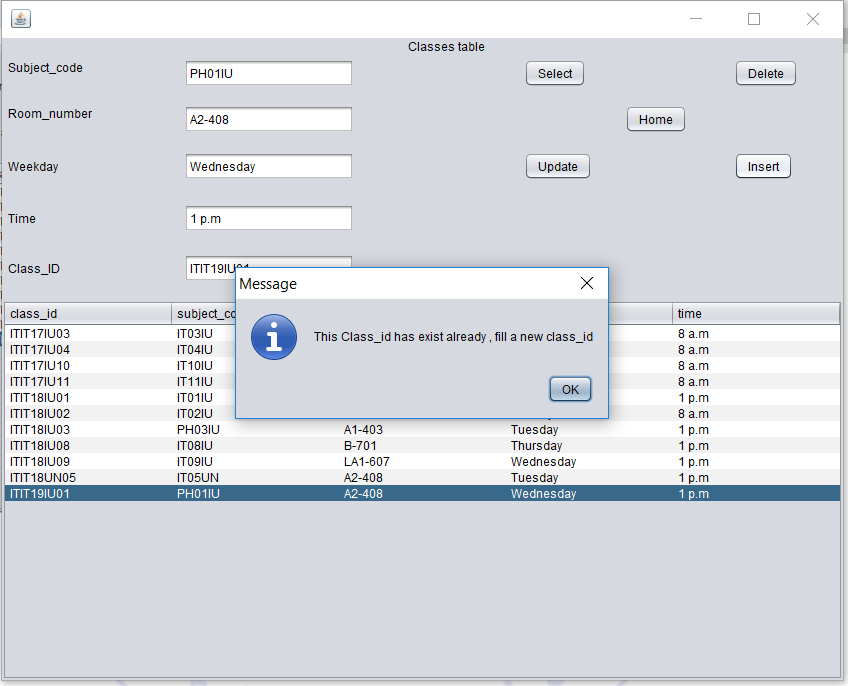
* Then use mouse to choose class and click the button “Delete” it will delete this class



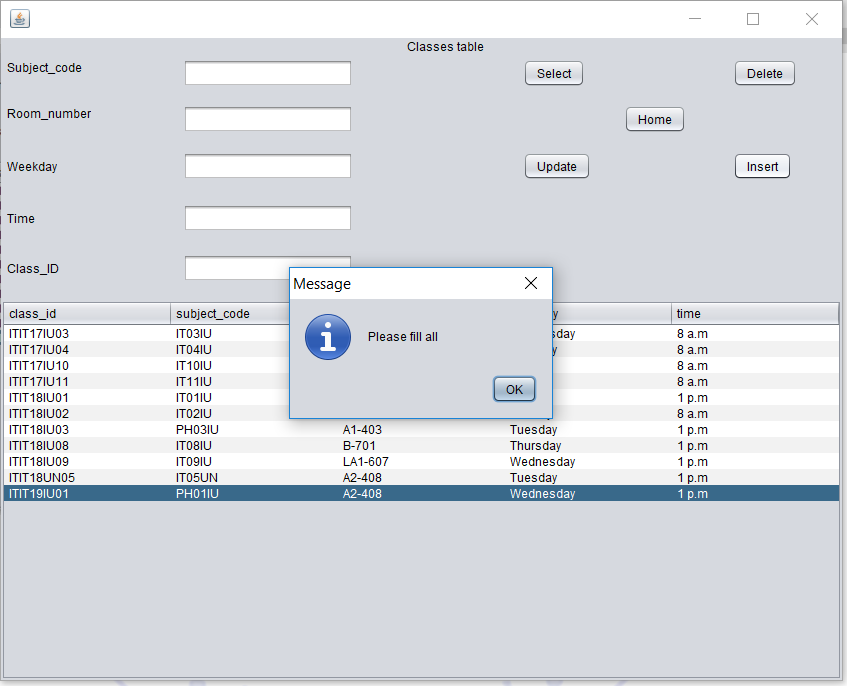
* I just delete the class with class\_id = ITIT19IU01 then I fill the information of this class in textfield and click the “Insert” button to insert it to database again



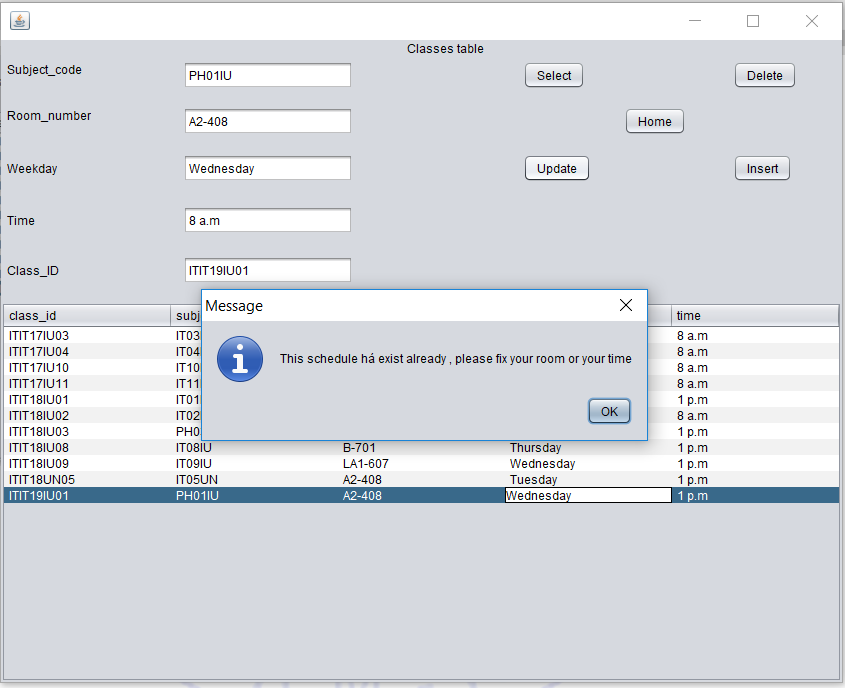
* If the class is already exist it will display



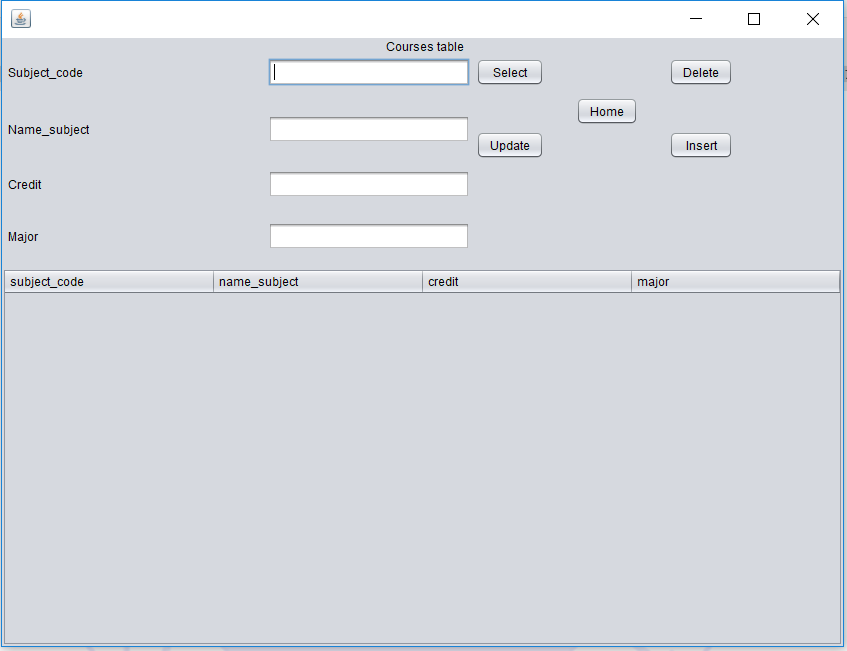
* Or if we don’t fill any thing in text field and click the “Insert” button the system will display



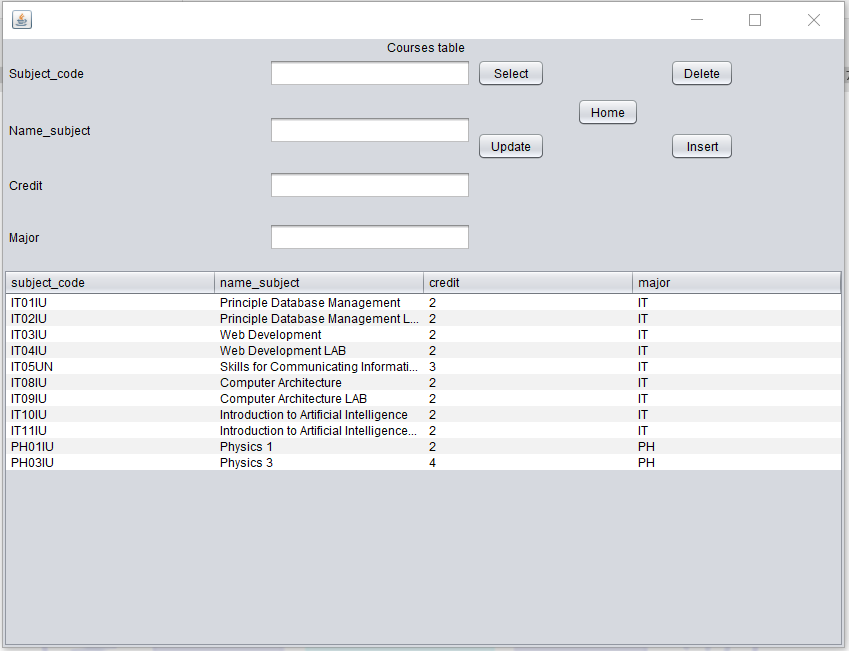
* Next, the “Update” button is easy to use, just use the mouse click to the class user want to choose and edit the information in textfield and click button
* And in the case user update the information and the room time and weekday is complicated with another class the system will display



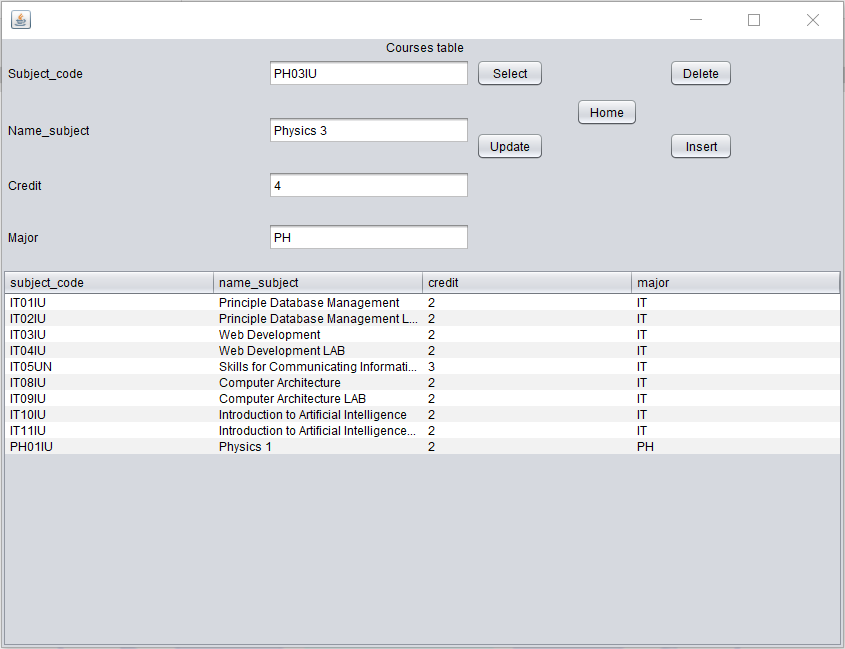
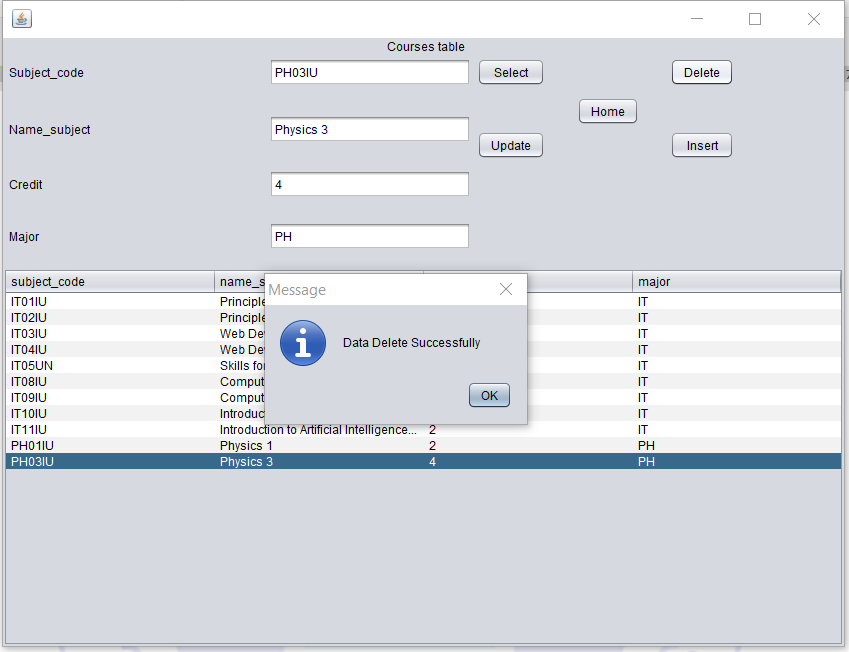
* Then move to the the next jframe, Courses



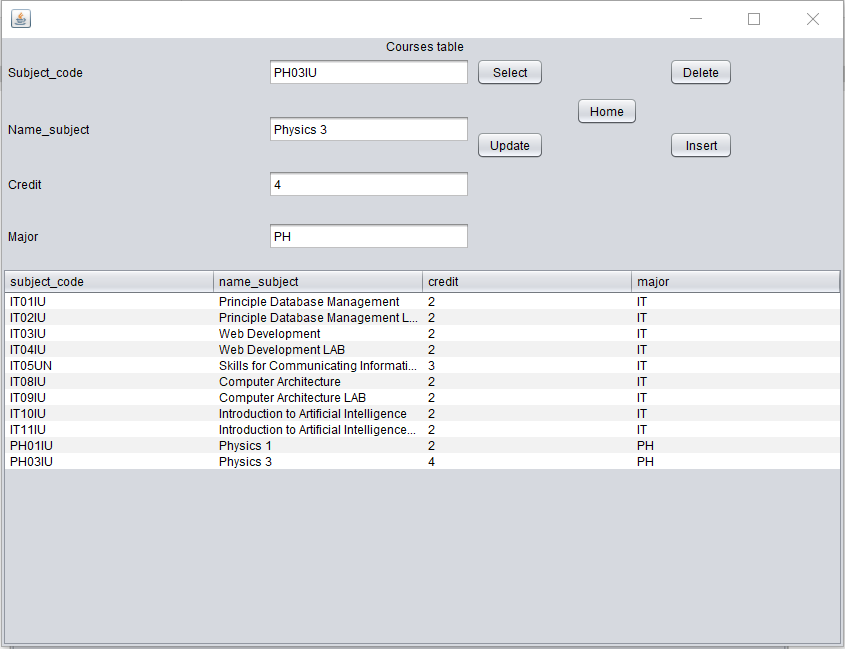
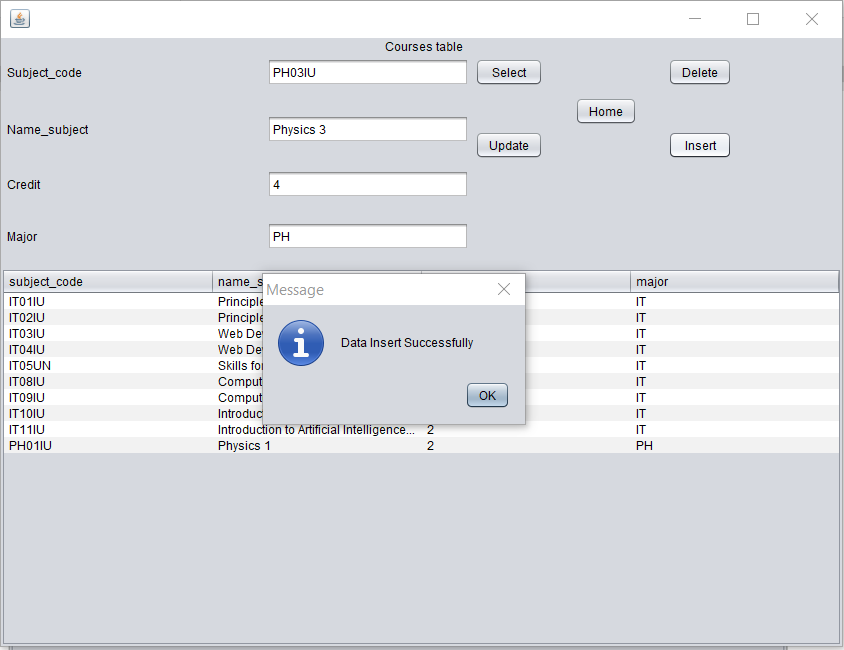
* There are 5 buttons similarly to Classes: Home( return home), Select, Delete, Insert, Update
* Click the Select button it will show all the courses



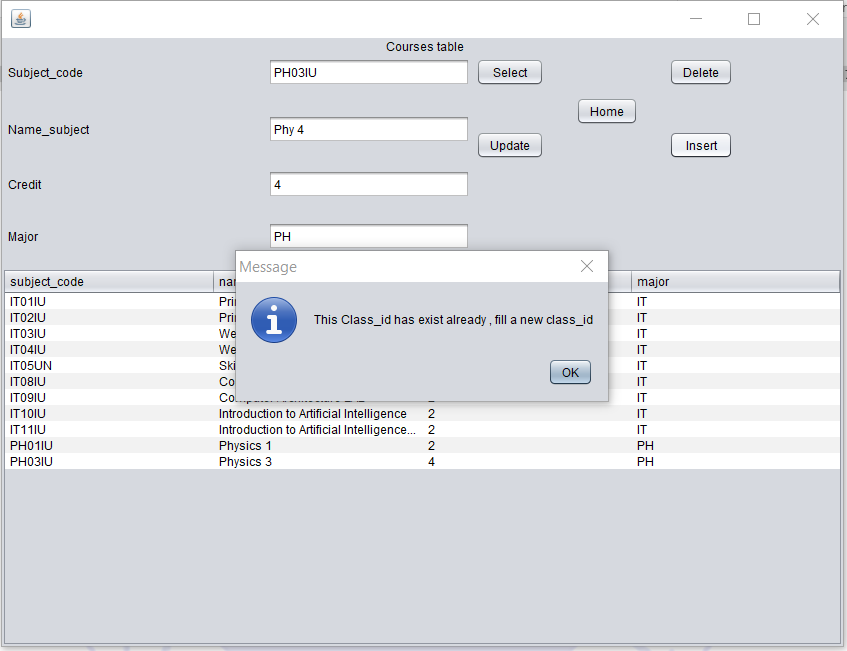
* And when user choose course and delete



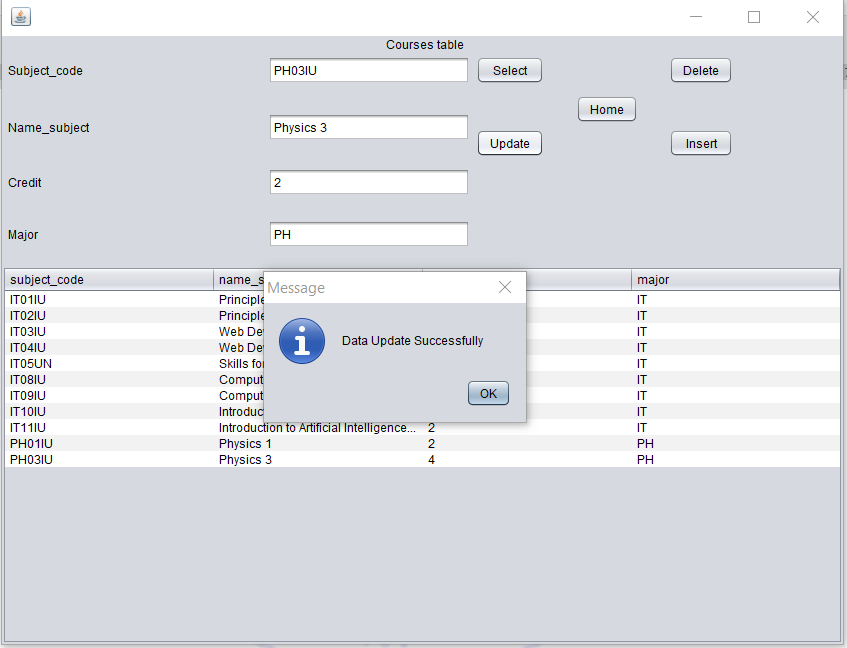
* And user insert the course he/she deletes it before

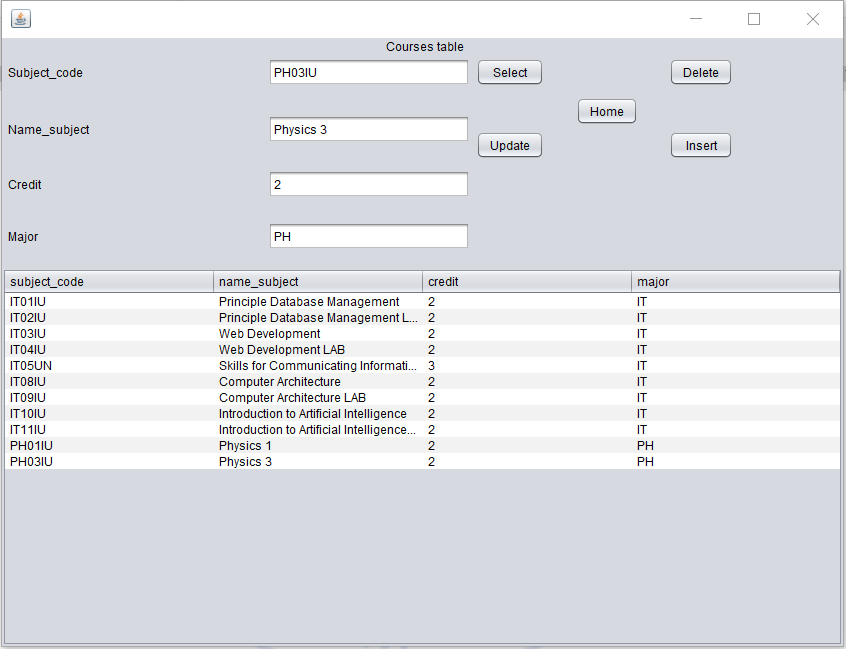


* In the case user insert the course that is not allowed to insert ,it will print out error

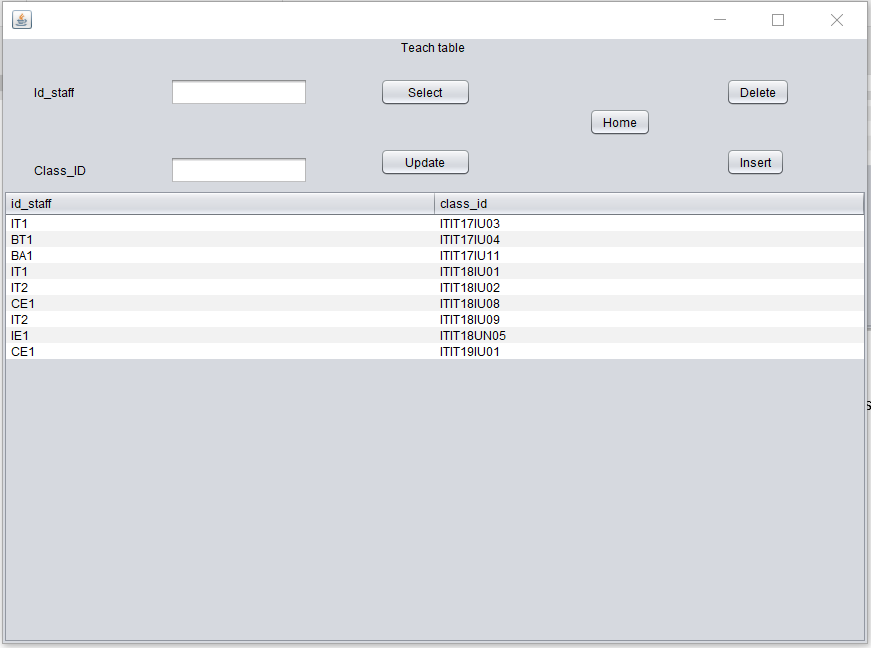
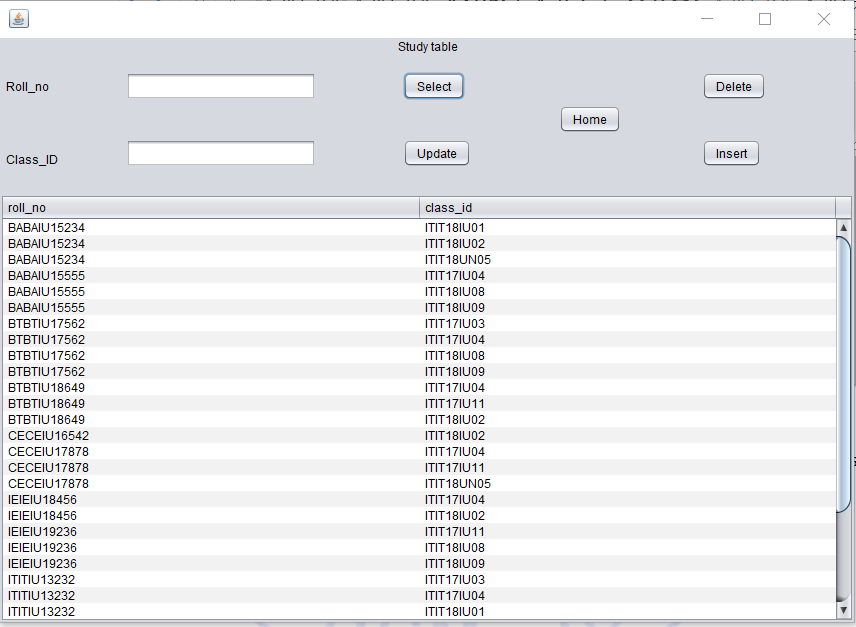
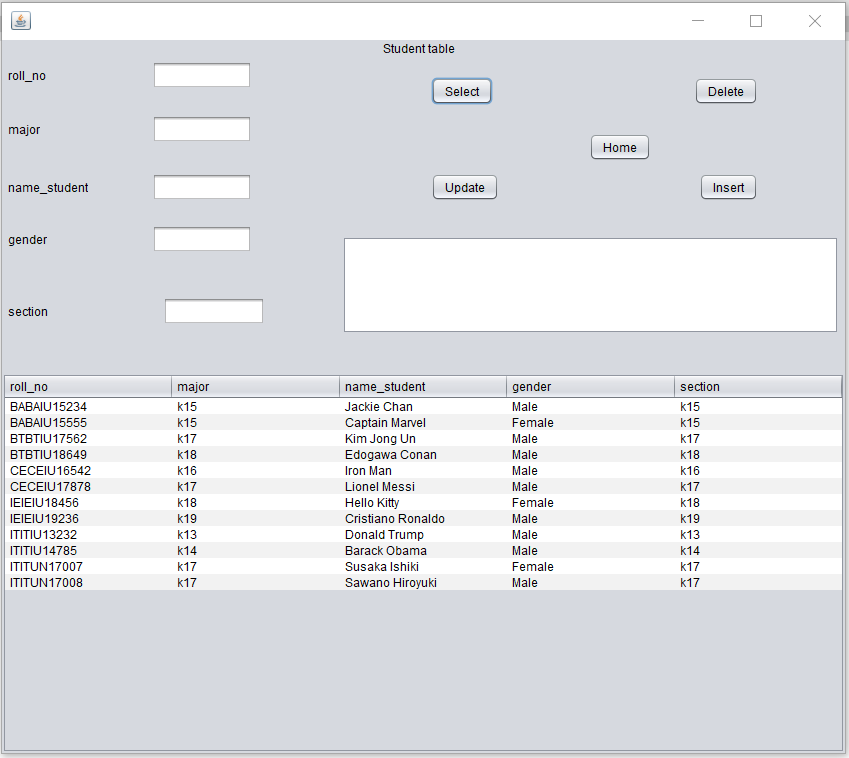
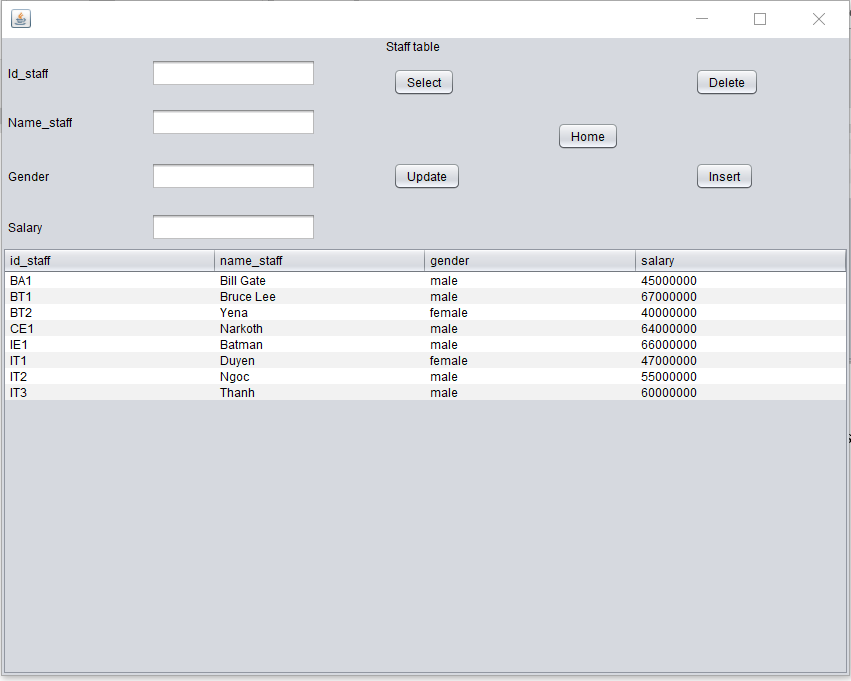


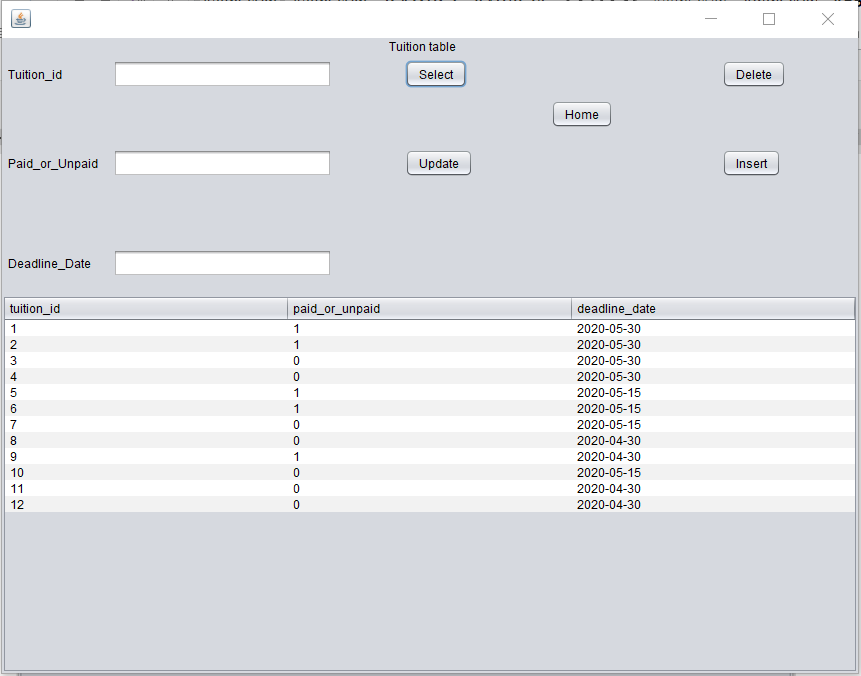
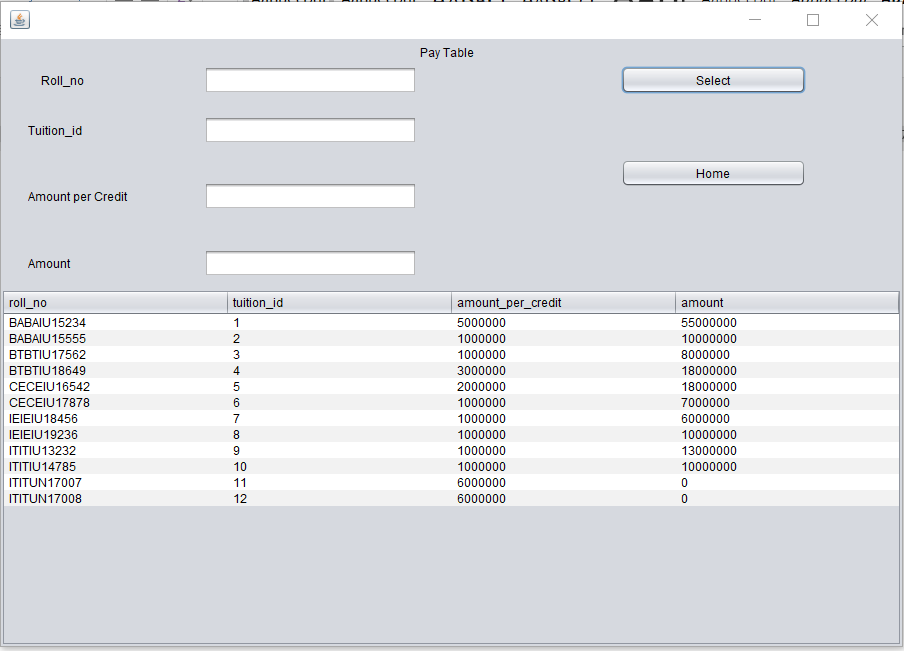
* Update button is so simple, just choose course and edit the information in text field then click update
* Example I will change the credit for Physics 3 from 4 to 2



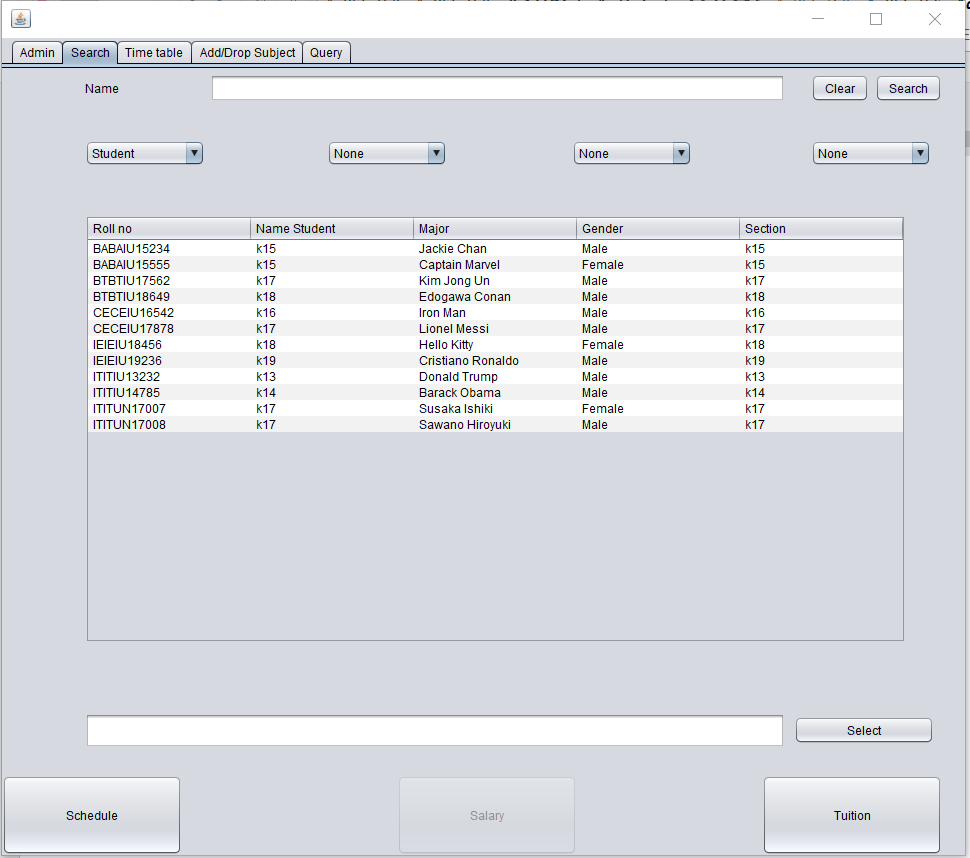


* Then move to the another jframe: Staff, Teach, Student, Study, Tuition, Pay
* These table are similar to and have same function, button with Course and Classes so I don’t show it again

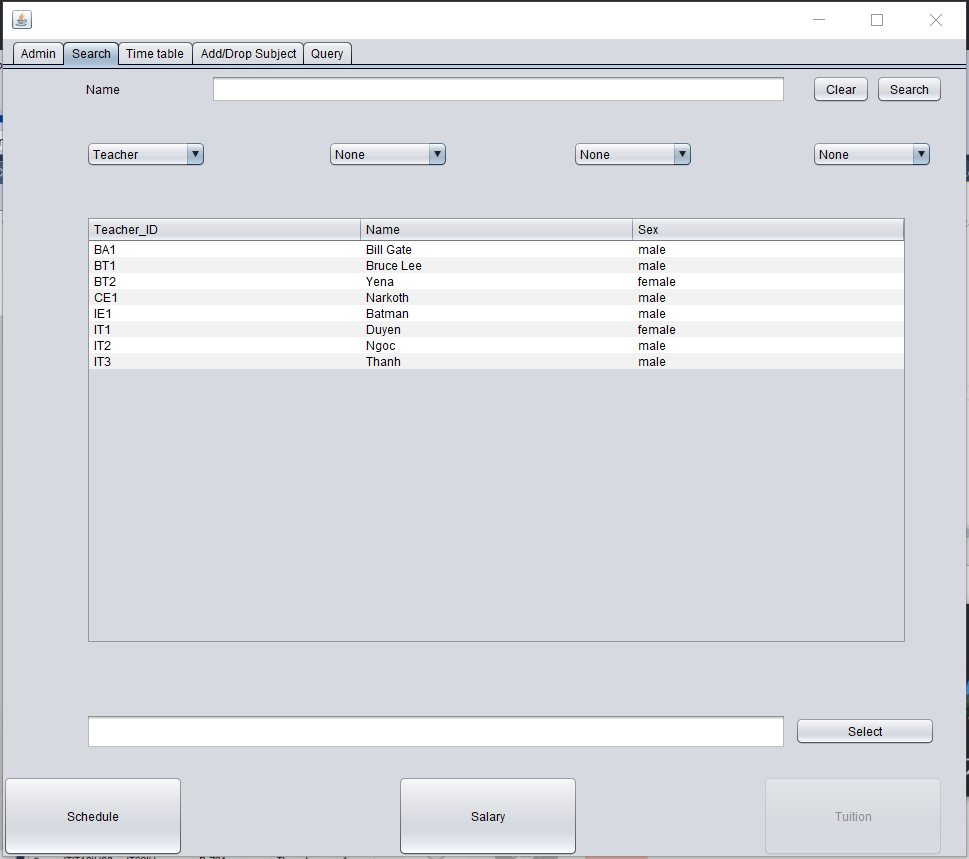




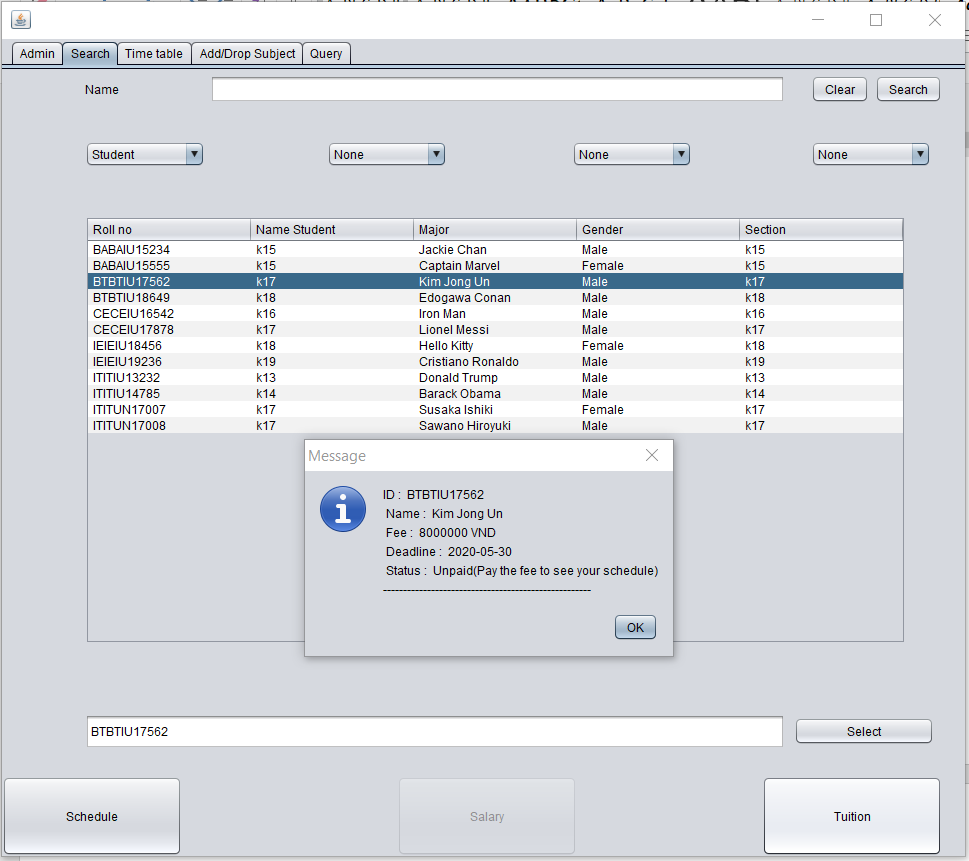
* Return to home and move to the next tab “Search”
* Then user can click the “Search” button to see the list of student and teacher(can choose Student/Staff with checkbox)



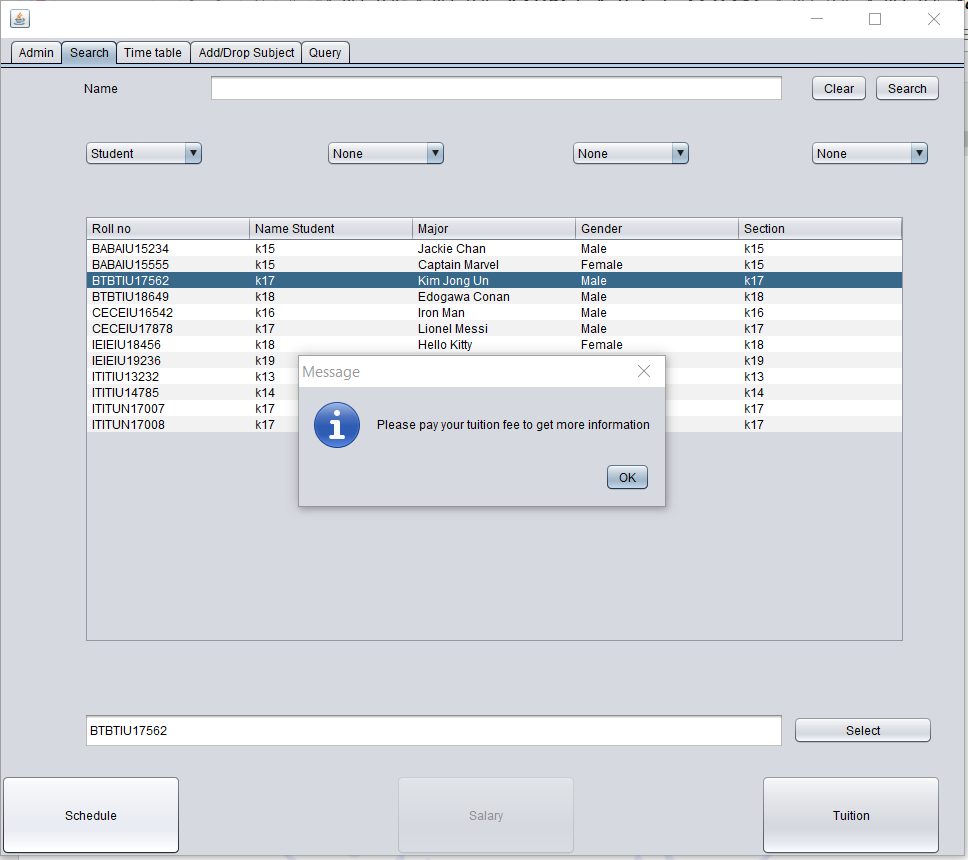
* As you can see, when we show the student the button salary below is disable, Schedule and Tuition is enable to click because student just has tuition fee



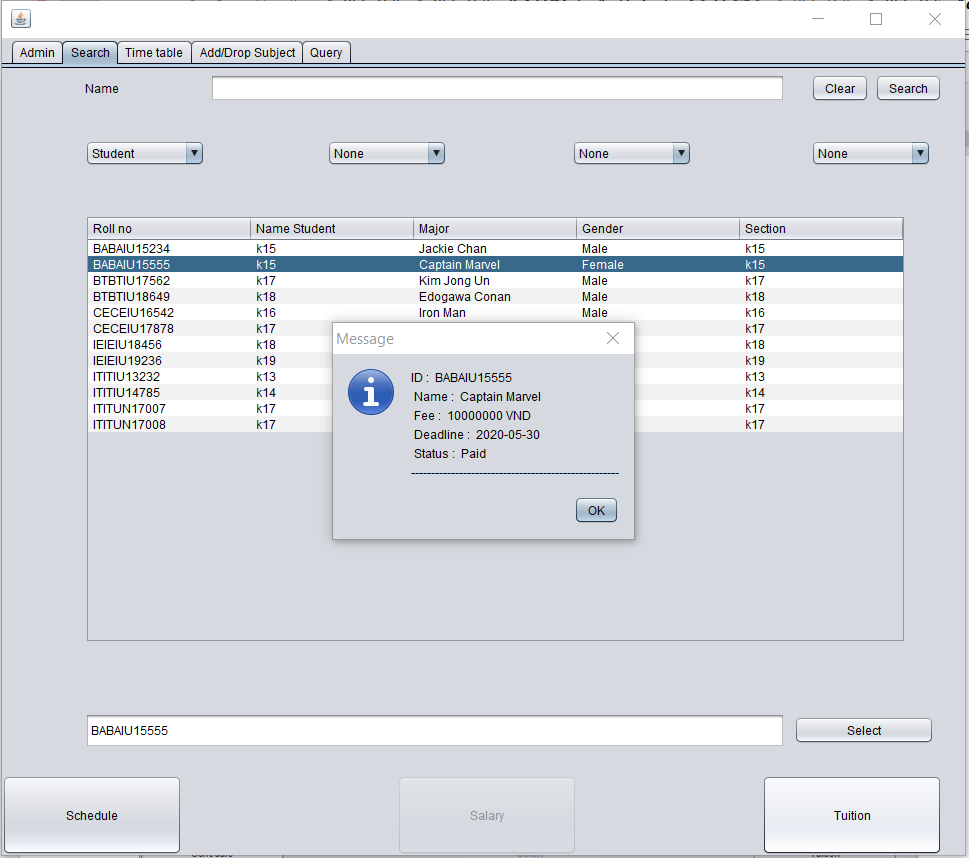
* As you can see, when we show the student the button Tuition below is disable, Schedule and Salary is enable to click because staff just has salary
* We will use this way to see student’s tuition fee and schedule
* First use the mouse click to the student we want to see
* Then click the “select” button to lock he/she
* Then click the “Tuition” button to see tuition fee of his/ her

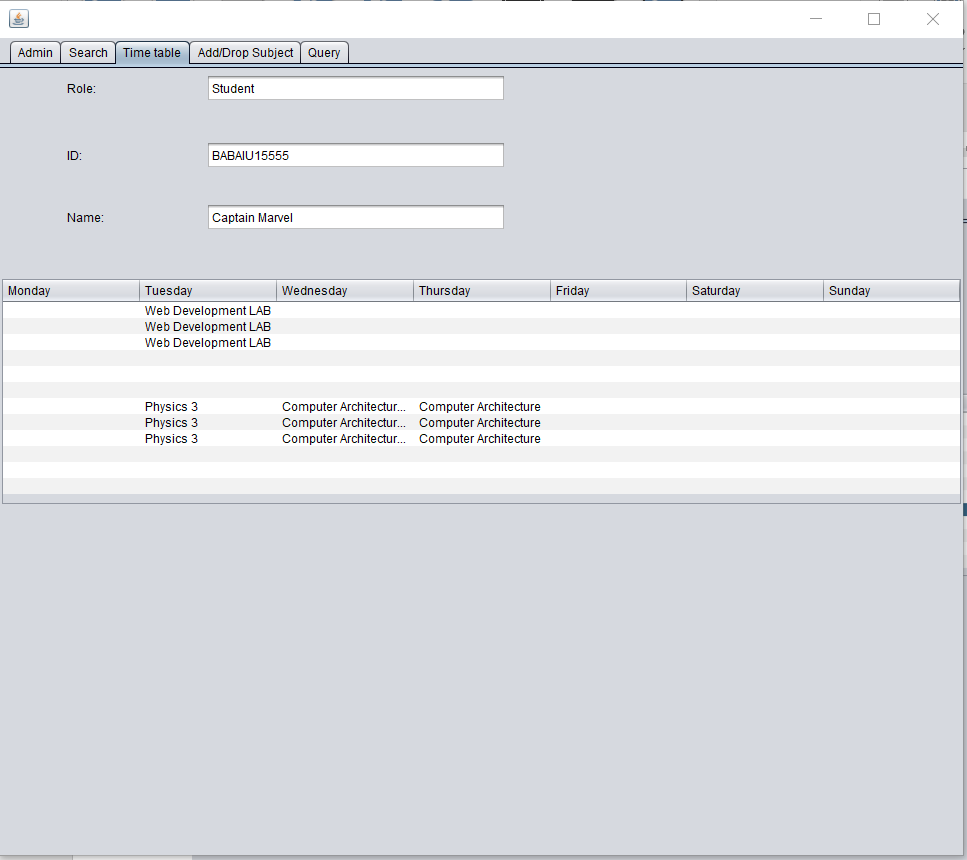


* Then want to see Schedule, we must sure that he/she has already paid, if not it will show annoucement

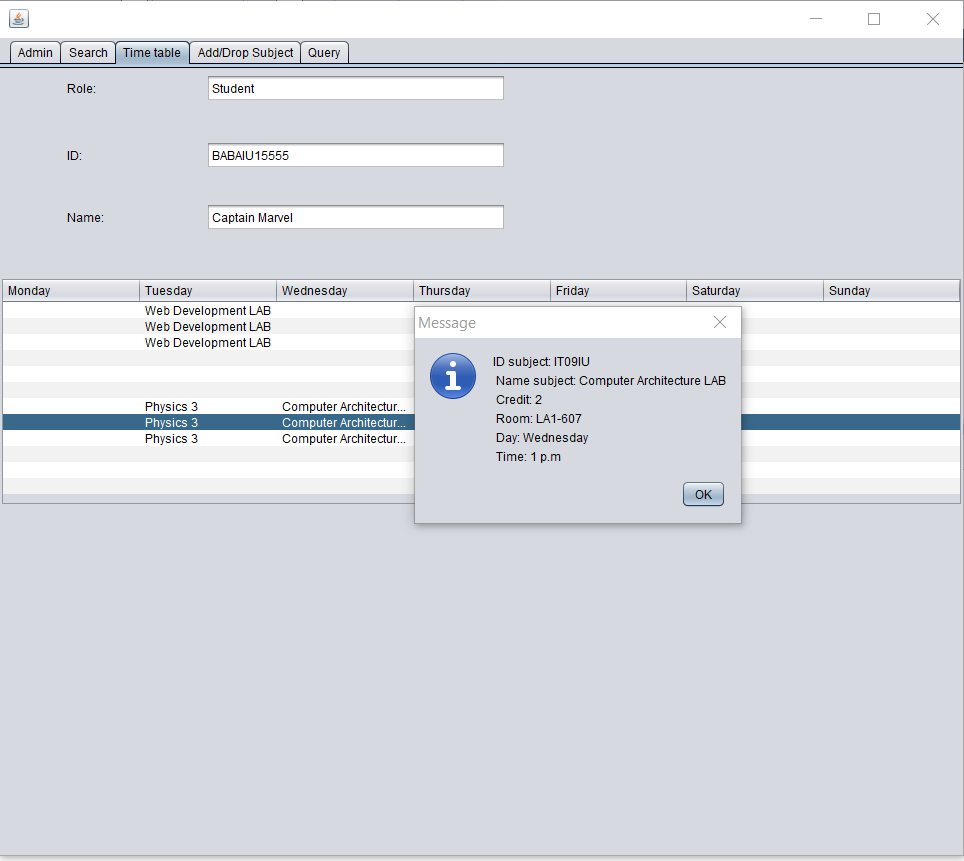


* If he/she paid, we click the “Schedule” button and move to tab “Time table” to see schedule

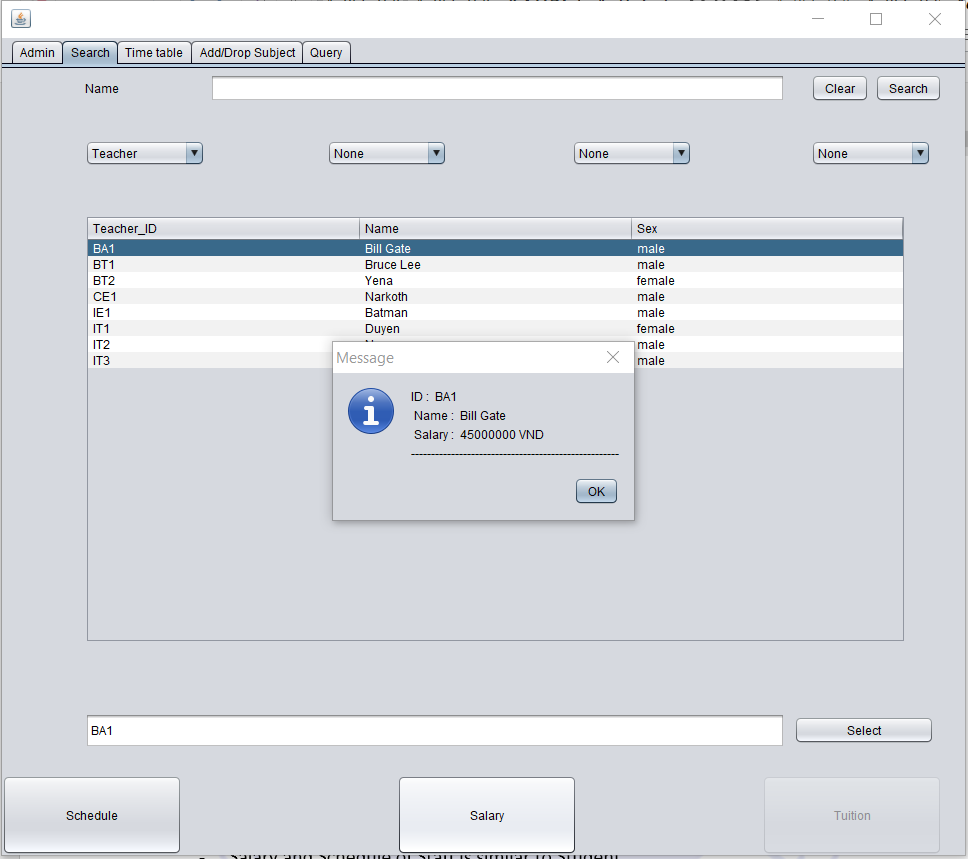


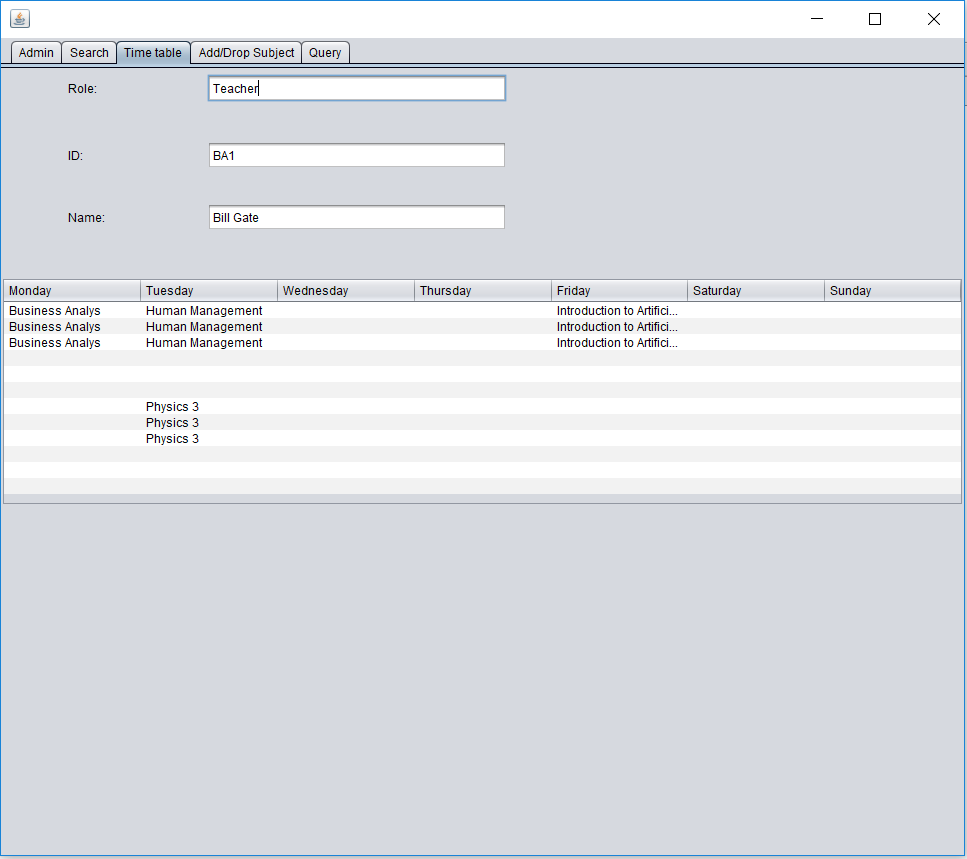


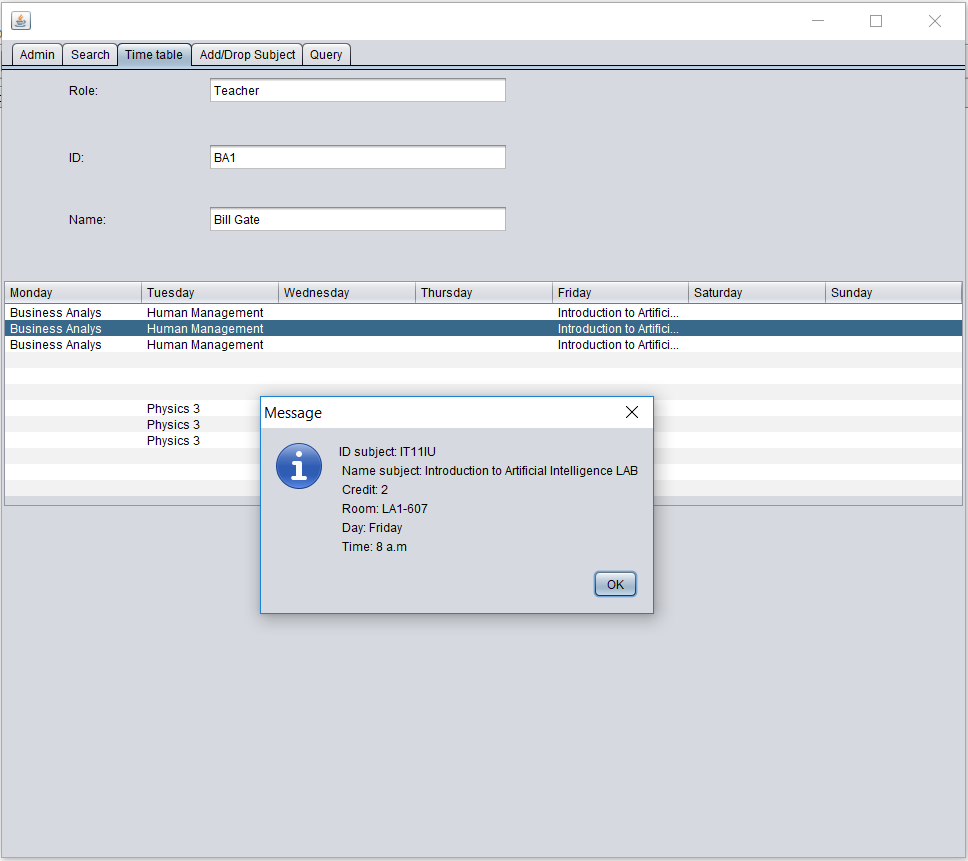
* Especially, user can click to the course to see this information



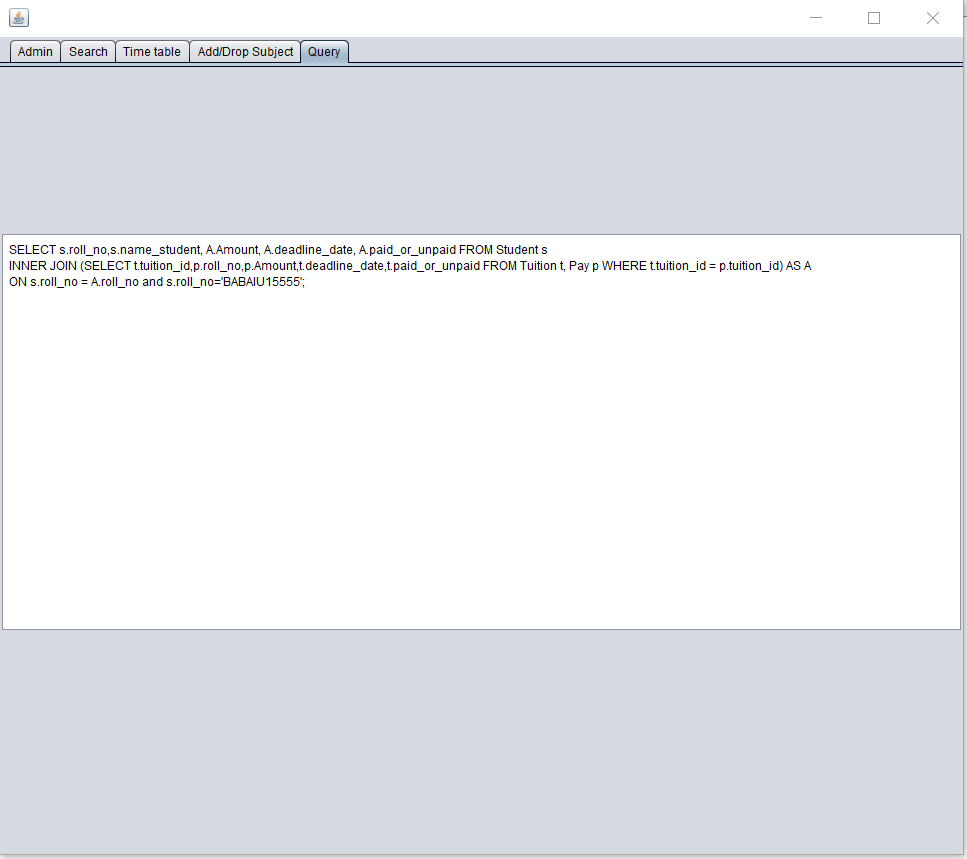
* Salary and Schedule of Staff is similar to Student



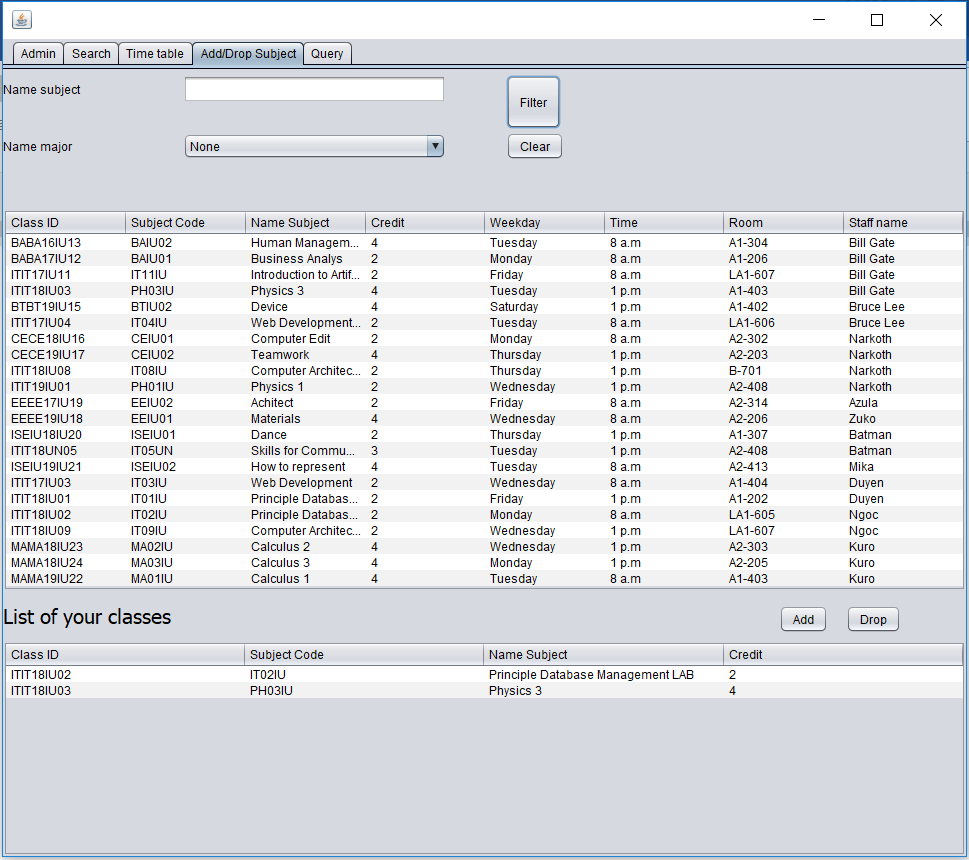




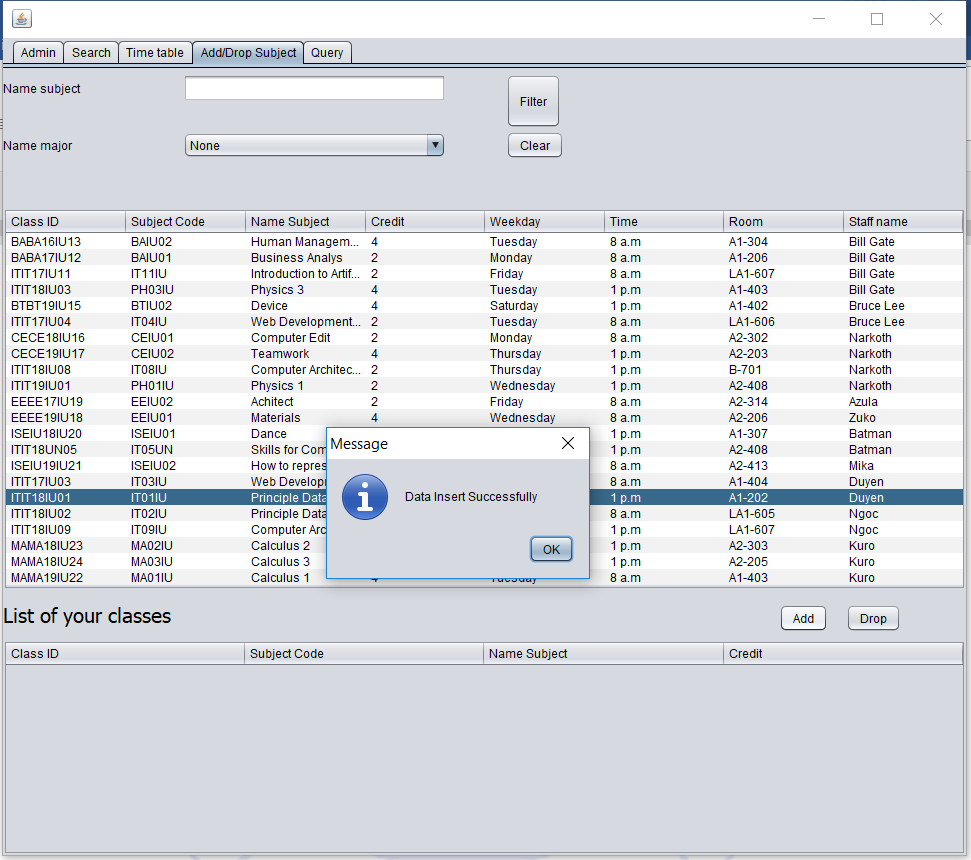
* Tab query is display the code for each button when we click and the system run this code
* Example when click the “Tuition” button to see the tuition fee of student

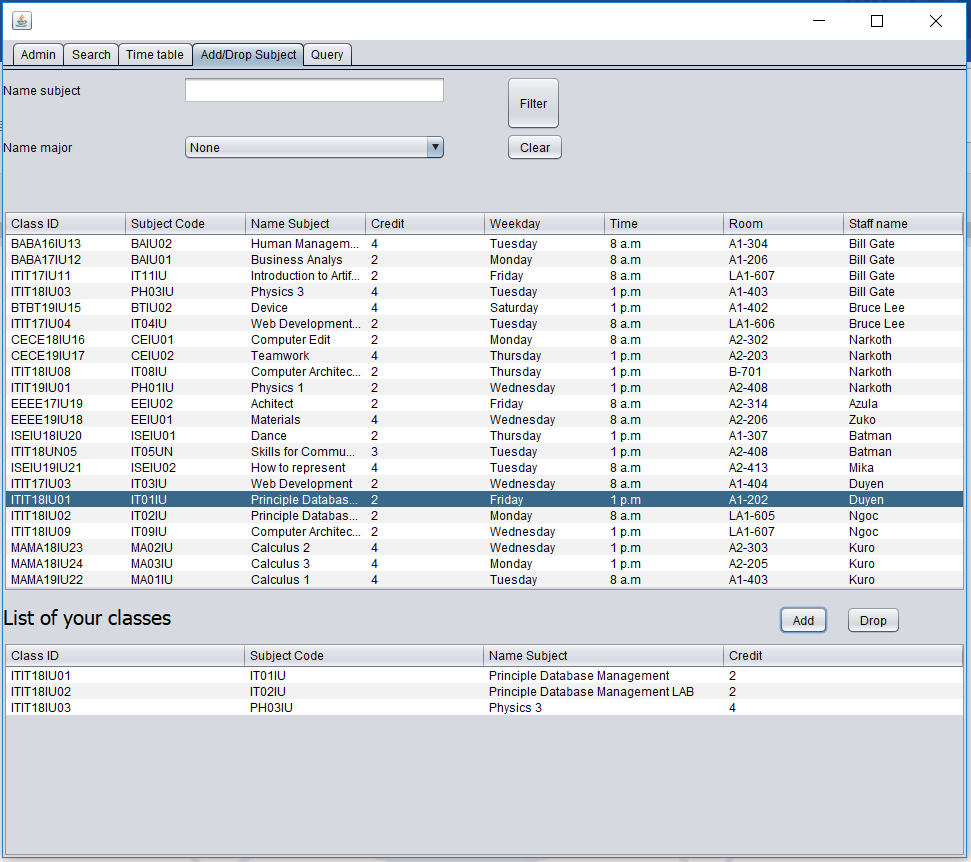


* In tab “Add/Drop Subject, the system allow admin to add or drop subject of student/staff
* Input the keyword into the textField Name subject or choose Name major, then click “Filter” button. It will search for the data. If the textField is empty, it will automatically display all classes in table.

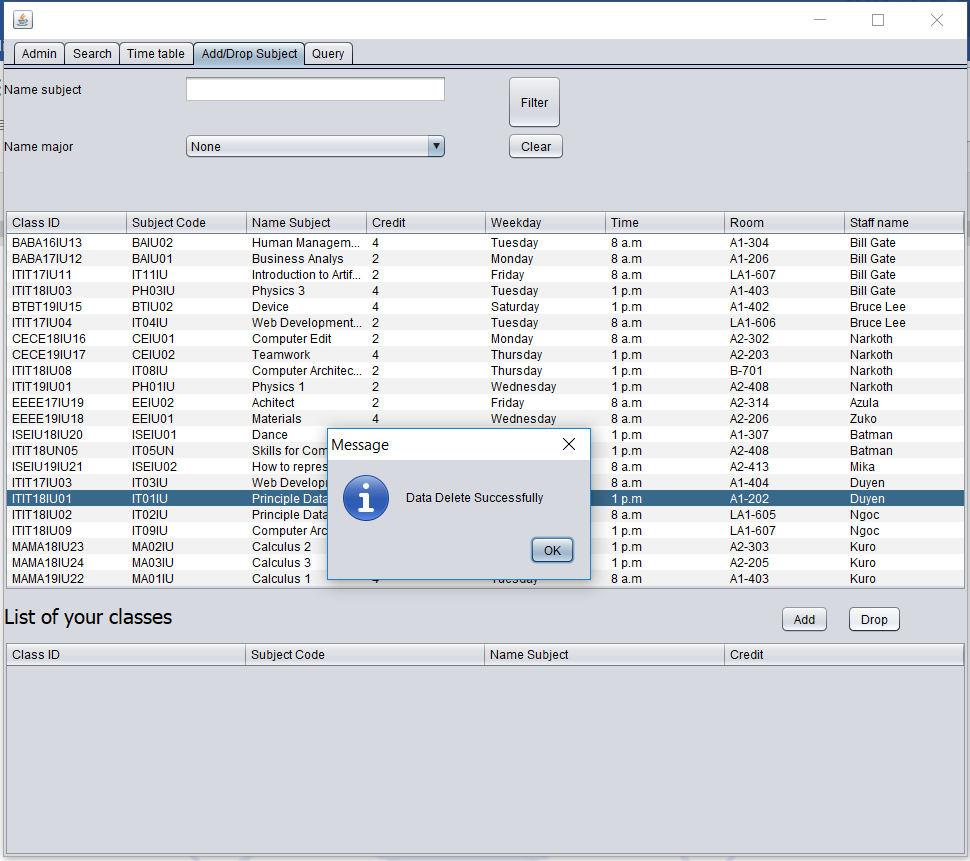


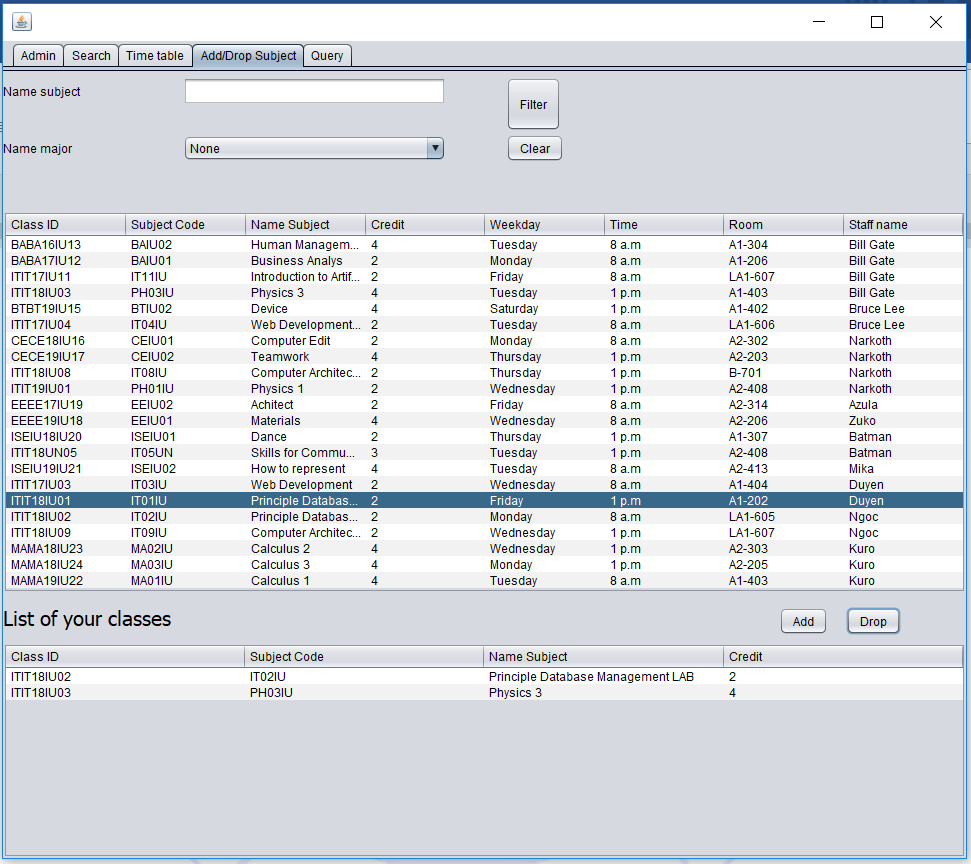
* Then admin is allowed to add and drop if the class is not be duplicated
* Add





* Drop





**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>
* Github of our project:
  + - <https://github.com/neonpotts/School-Management-PDM-project->