****

**FINAL PROJECT**

****

Programming Proficiency

Professor Robin GAO

Members:

Aldir Jose Borelli Junior - 19664314

Suman Satasiya - 19663148

**Student Enrolment System**

**Summary**Student enrollment system was developed to the Eastern Sydney University (fictional) so the Students and new Students can enroll in the units that are being offered in the current semester. This system covers the entire module such as how the student can enroll, what are the unit offered, what are the time schedule for each subject, unit information, and student profile, etc. This system was built in Java language and used MYSQL database to store the users’ information, and the system interface was based in Java Swing.

**Introduction**

The project is integrated with the main University’s system, however, only the module Enrollment will be developed. The system uses the Object-Oriented paradigm and has two users named Student and Administrator, each type of user has different permissions.

There are 5 main different interfaces in the system: Login, Home, Profile, Enrolment, Units and Board edition. After logging in, the users access the home interface that presents important notifications divided in 2 boards, student board and upcoming events in the University. These boards show information such as semester fee payment dates, exam registration, change in the exam time table, workshops or festivals that will be held, etc. Students can only see the news but the administrators can update the boards, inserting, updating or deleting new posts.  
The menu on the left side shows different option for both admins and students. Students can enroll in units in the Enrolment interface, units they already enrolled are also presented and they can see a brief information of the units being offered. There is also a Unit interface that presents more information of the Units, one at time, but only the admins can update them. Users can access the Profile interface, in case the user is a student, it presents their information and they can update password, in case the user is an admin, they can check and update students details and also insert a new student into the enrolment system.  
It is important to clarify again that this system does not cover the entire University management system. However, it is integrated with the main system.

**System Analysis**

The development of this Java application is done based on the requirement analysis. The requirement phase involves the gathering of necessities and demands of each category of user – student and admin, then prioritizing these requirements like software system availability. Here some specific subsystem functional requirements are demonstrated:

**Administrator Requirement**

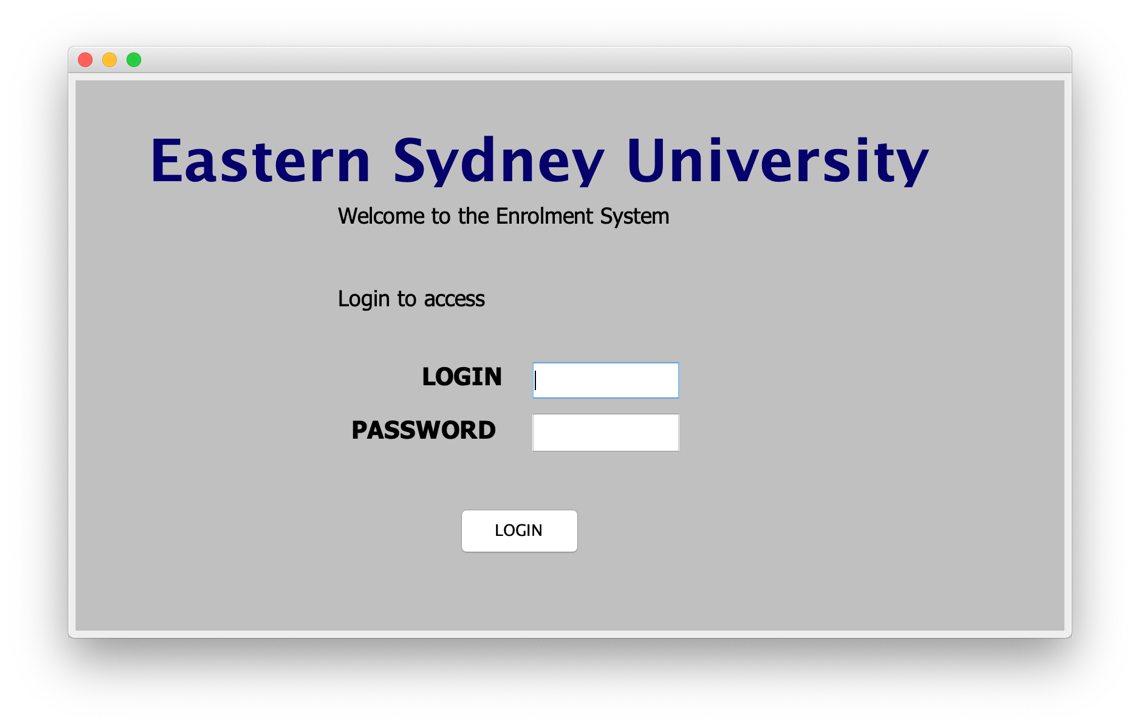
* Manage Students
* Manage Units
* Manage students and news boards

**Student Requirements**

* Check personal data
* Update password and email
* View all up-to-date notifications or activities in the University.
* Enroll in subjects for the current semester
* Access unit information.

Based on this user requirement, the system was developed and screenshots of the system and their functionality are explained below:

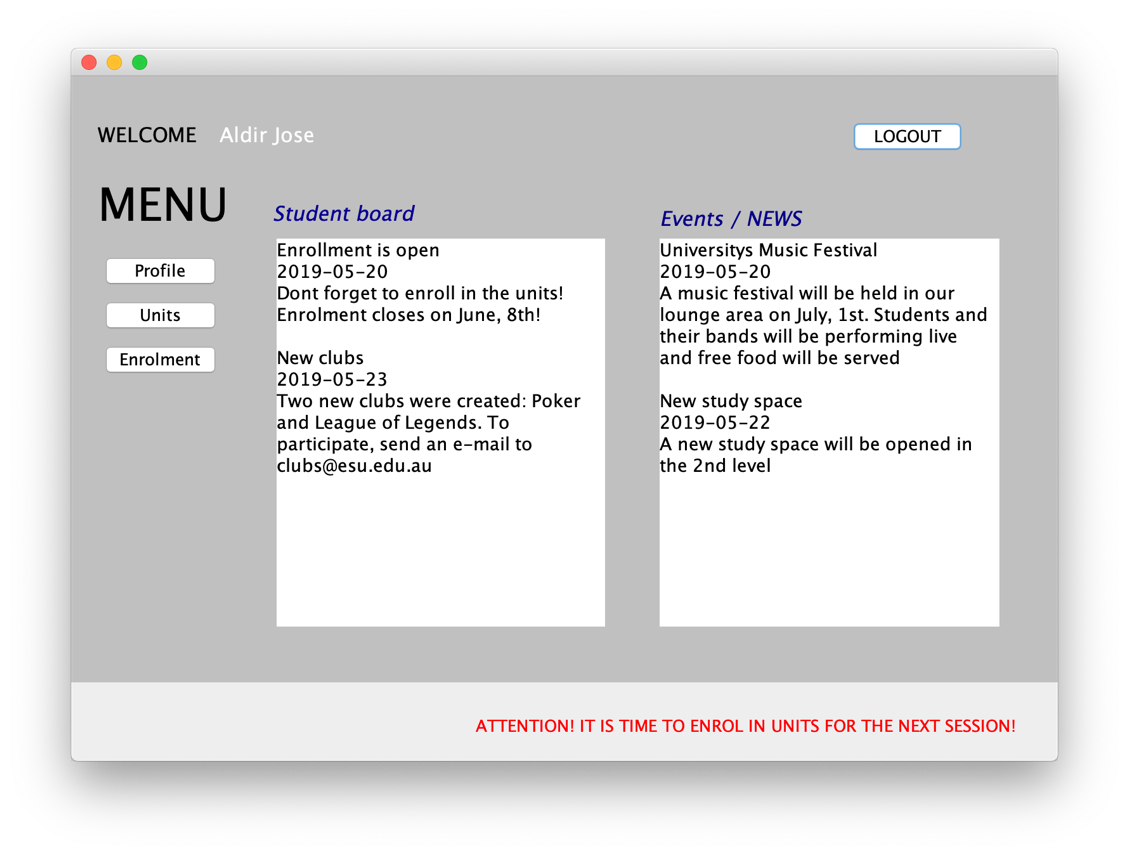
**Login Page**



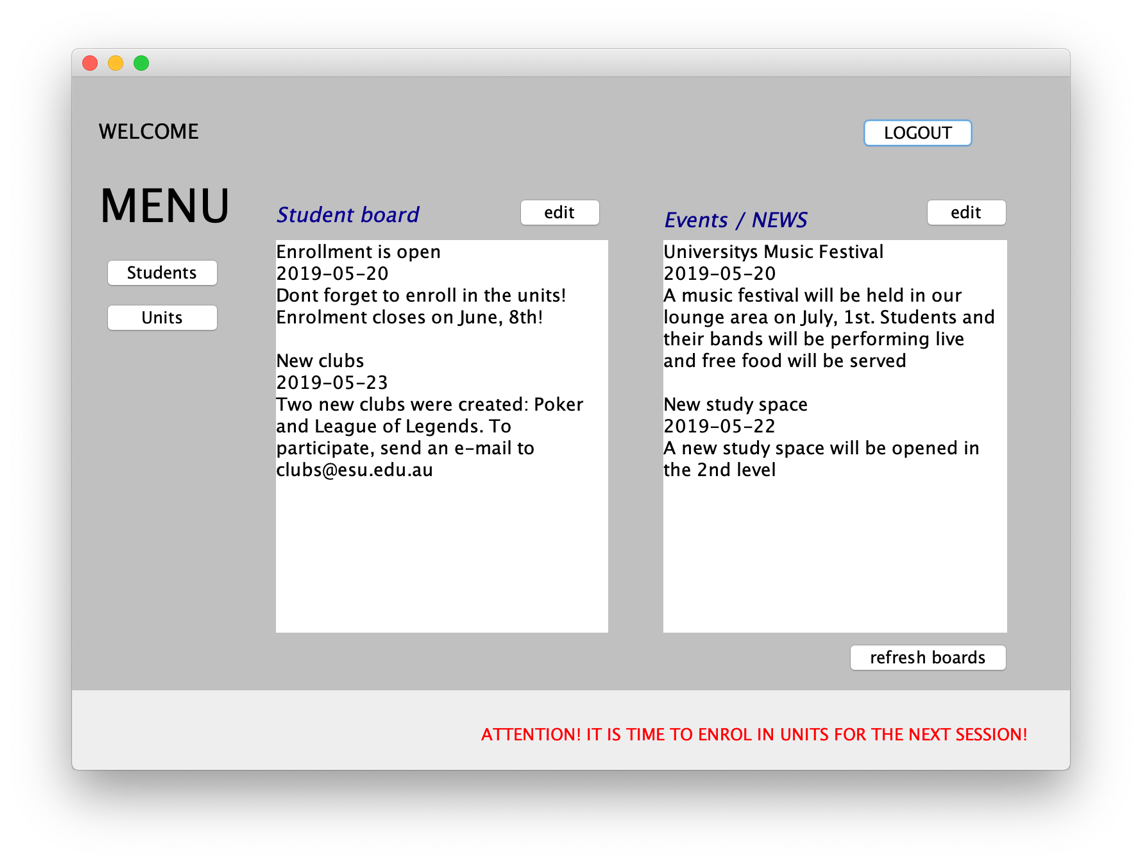
This is the initial page of the system. The user needs to enter the username and password to login into the system and access the system functionalities. As mentioned above there are two types of users: Admins and Students, and the next interface presented is based on the username entered by the user.

**Home Page** (same interface, 2 different views)

**Students:**

****

**Admins:**

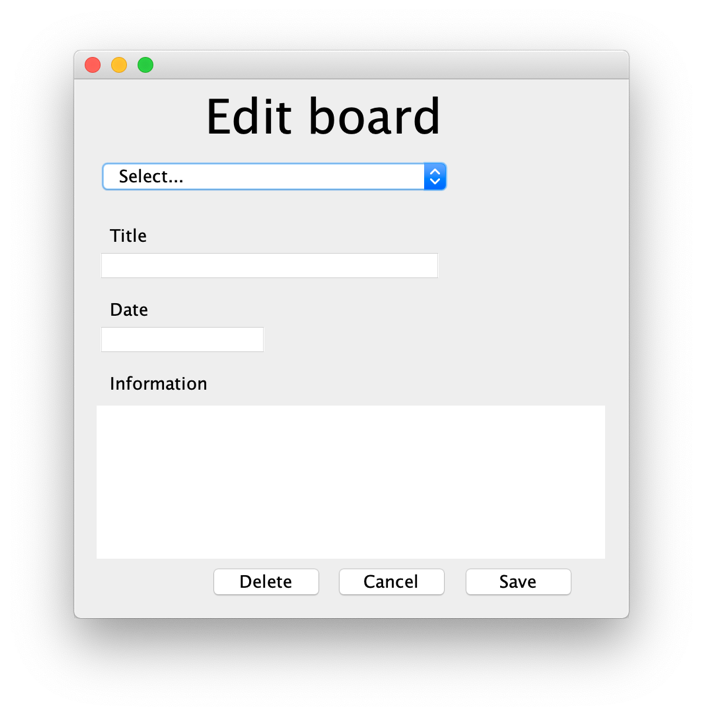
****

This module is accessible by both users. There interface is adaptable for each user, offering only the appropriate buttons for each profile.

Students are welcome by their names on the top left area, the menu is presented on the left side with the options: Profile, Units and Enrolment, and they can also see the student board and news board information. At the bottom, they see important notification and at the top right, the Logout button is present so that the user can securely log out from the system.

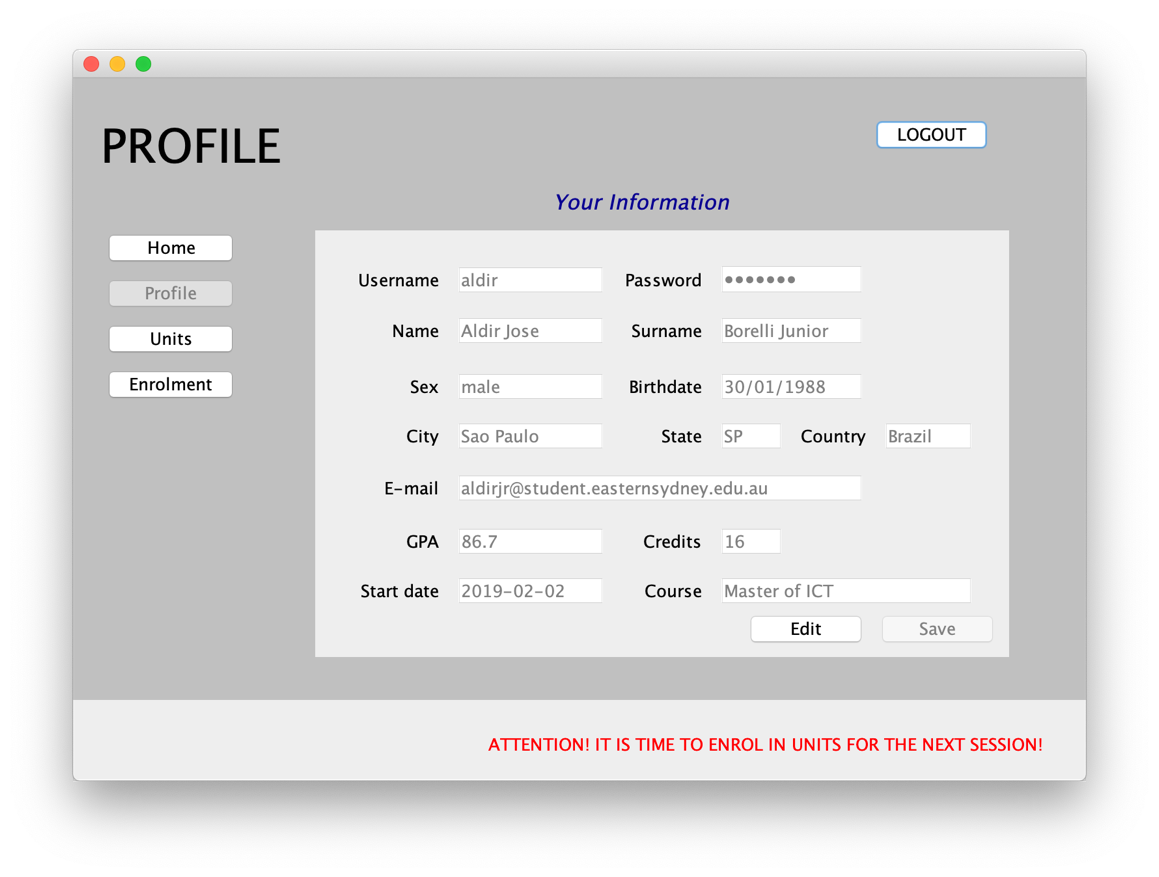
On the other hand, Admins have only 2 menu options: students and units. There are also two options to update student board and news board, by clicking in the respective ‘edit’ button, a new interface will be open. The ‘refresh boards’ button should be used to update the boards that were edited.

**Edit Board**

****

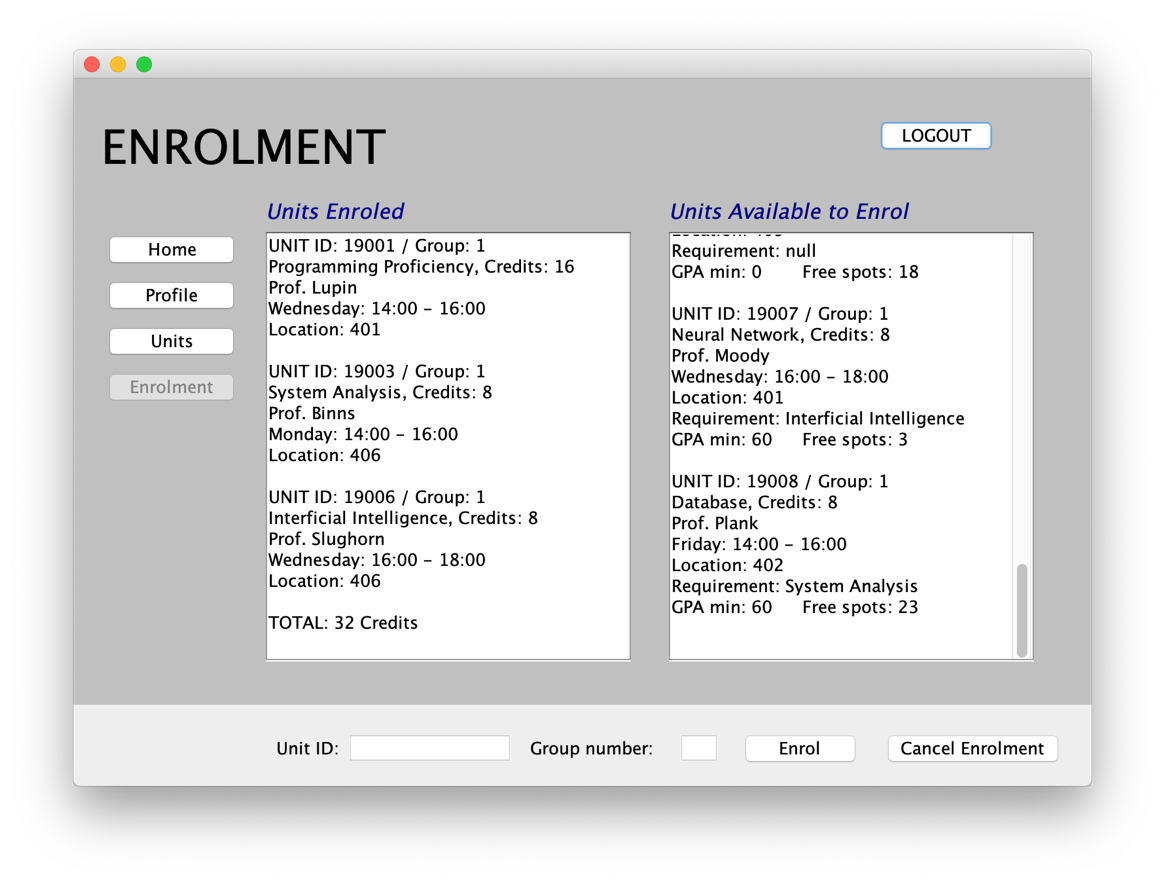
In this screen, the admins can edit the boards information. They can create new announcements and also update the ones already made.

**Profile**

****

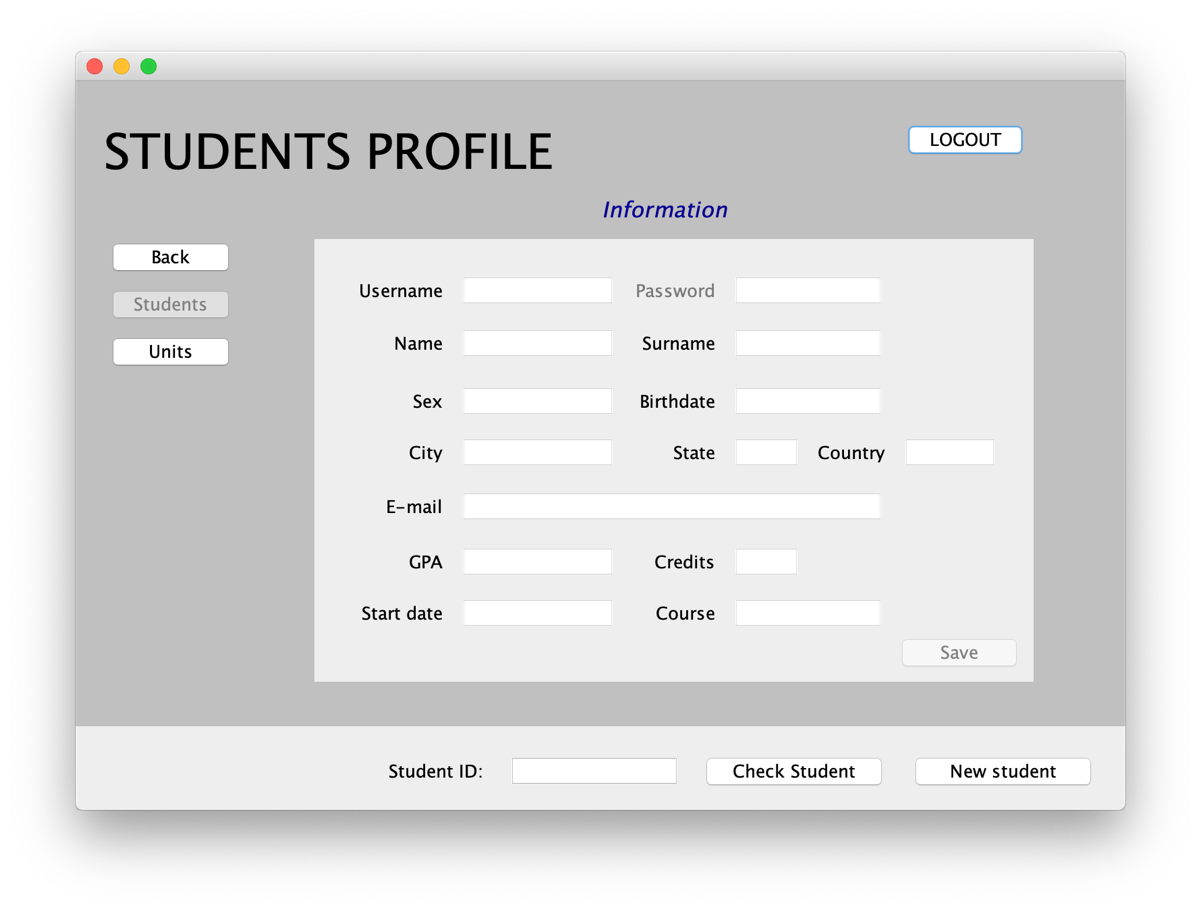
In the profile module, user can see their personal information and able to update email and password. Some information also includes their current GPA and credits obtained till the present moment.

**Enrolment**



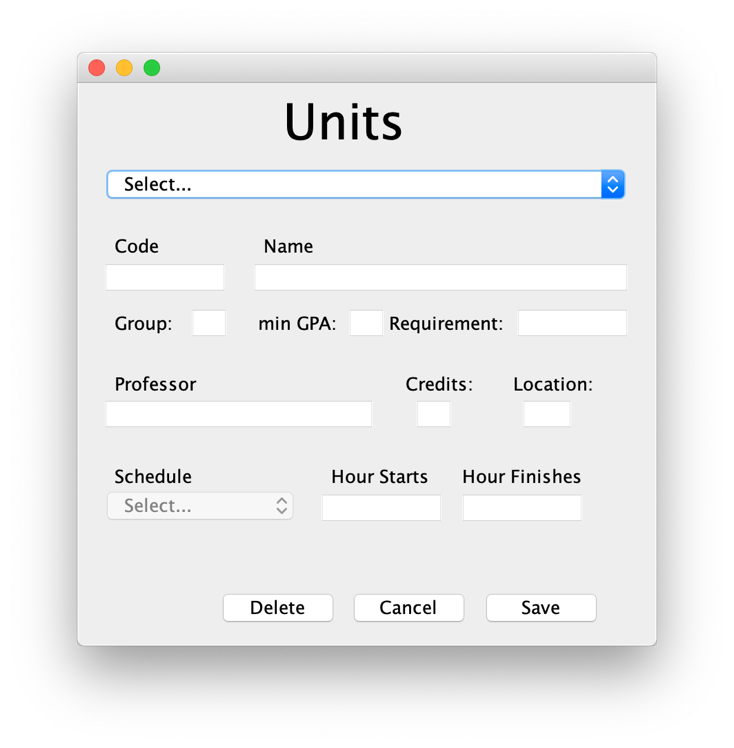
This is the most important module of the system, as it is the main reason the system was created: to allow Students to enroll in the offered subjects. Student can see available units on the right board and enrolled units on the left board. To enroll in a unit, or cancel an enrolment previously made, they have to use the bottom part of the interface, inserting the Unit id and also the group number chosen.

**Student Module**



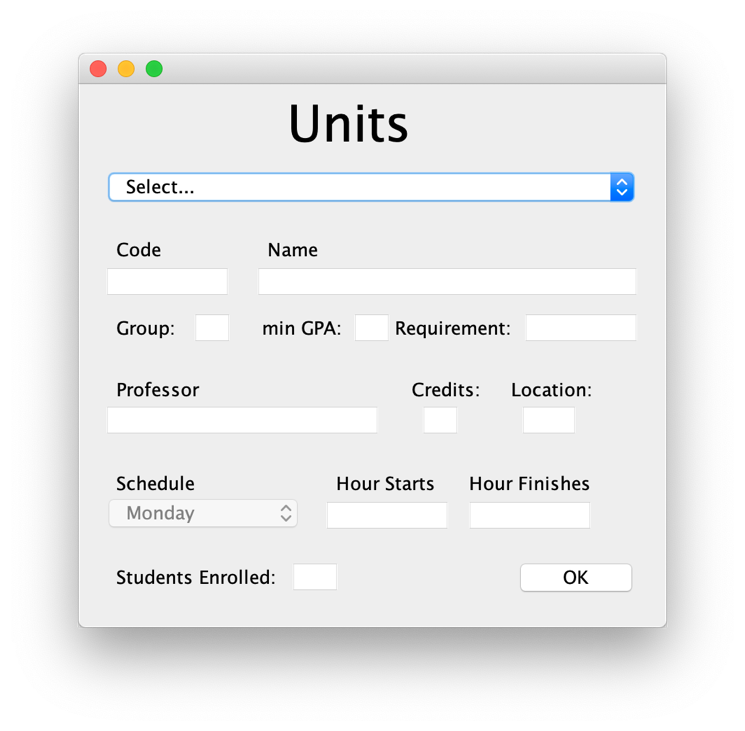
In this module, the administrator can edit Information for existing student or can register a new student. At the bottom, to check a student information, the admin needs to insert the Student ID and press ‘Check Student’ button, and to add a new student to the system, they need to press the ‘New student’ button. Both buttons enable the Information fields – if it is an existing user the form will be filled with the respective information.

**Edit Unit Module**

****

In the Unit module, units can be managed – created, updated and deleted. The combo box menu contains all the Units that are being offered in the current semester and also contains a ‘New’ option that will allow the admin to create a new Unit.

**Check Unit Module**

****

In the Unit module, Students can check the Units separately, the difference of this module when compared to the Admins one, is that students cannot update the units’ information and they can check how many students are already enrolled in the unit.

**System Overview**

This is a system user interface in which the user can easily interact with the system and do the desired operation. It has menu with clear names, therefore, the intended user can easily interact with the system and perform the operations within a few clicks. It will not cover the entire university module but as per scope, it is a small integrated part of the system.

Student enrollment system was developed with a Java programming language which supports object-oriented programming language, used in this project. Object-Oriented Programming (OOP) refers to a type of computer programming in which programmers define not only the data type of a data structure but also the types of operations (functions) that can be applied to the data structure. In this way, the data structure becomes an object that includes both data and functions. One of the principal advantages of object-oriented programming techniques over procedural programming techniques is that they enable programmers to create modules that do not need to be changed when a new type of object is added. A programmer can simply create a new object that inherits many of its features from existing objects. This makes object-oriented programs easier to modify. There are many concepts of object-oriented programming which we used in our system, these concepts are described below.

* **Class**

A class is a collection of method and variables. In our System for every screen we build class and each class has several methods and variables. These methods connect to the database to get or update information, store information in the variables, assign variables to the JFrame and also define the components of this user Interface.

* **Objects**

An object is a software bundle of related state and behavior. Software objects are often used to model the real-world objects that you find in everyday life. In our system, the user is defined as an Object in the Login interface and is passed as argument to all the other classes. The different classes use the user object to retrieve information such as username and in order to present a new interface to the user, the current class would also create an object of this new interface.

* **Package**

A package is a namespace for organizing classes and interfaces in a logical manner. Placing your code into packages makes large software projects easier to manage. Our system is built under one package, so it is easy to access all classes within the same package and to integrate with another system.

* **Inheritance**

Inheritance provides a powerful and natural mechanism for organizing and structuring your software. In Java programming language superclass functionality can be derived into a subclass, therefore inheritance allows the code reusability. In our system, most classes extend the JFrame class, allowing the interface to be managed appropriately.

* **Exception handling**

In java exception handling done through the try-catch block that paradigm. If an exception occurs in try block then the control of execution is passed to the corresponding catch block. A single try block can have multiple catch blocks associated with it, the generic exception handler catch block is at the last.

Our system was designed using exception handling to monitor all the database connections and also the transformation of variables to different types.

Java is a rich programming language set and there are many strong technological aspects. In the group opinion, our student enrollment system covers many strong technical aspects. Firstly, it uses inheritance so that this provides code reusability. Second is exception handling any error will be caught in the catch block, so the user can interact with the system quickly, easily and securely without errors. Thirdly, we use a database with JDBC technology so that we can easily manage user data and perform operations easily. Lastly, the encapsulation of the code, allowing to hide the inner classes and only give access to the desired codes and also make the system more flexible, easier to change and maintain. Moreover, JFrame usage provides a simple visual graphical interface so that user can access the view the information n very simple way.

**Conclusion**

In conclusion, the main purpose of student enrollment system is to develop a system which provides the functionality to university student to enroll in different units with different time schedule as per their choice. Students can see basic information which is related to their course and department also they can do analyses the provided units. This system uses best object-oriented programming concepts that provide code reusability, error handling, and simple graphical user interface and connection to an external database to store, update and retrieve information.

Overall it was an amazing experience to develop the system as we could practice what we have being learning in the Unit and understand how it can be applied to the real world. Programming proficiency unit provides the best knowledge about object-oriented programming, how we can create the new object without affecting the existing object and classes. We can use many methods with the same data and uses the feature of code reusability with super class and subclass hierarchy. Java is a secure and reliable language and it is useful to develop a major system using object-oriented programming language.