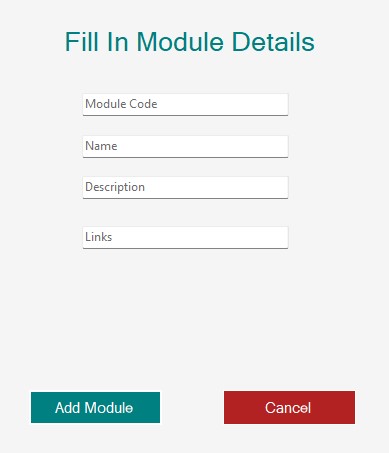
Jaydon Erasmus

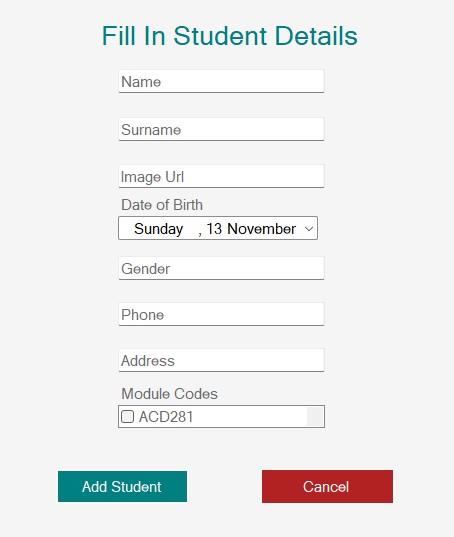
Ozias Mulenda

Matthew Damstra

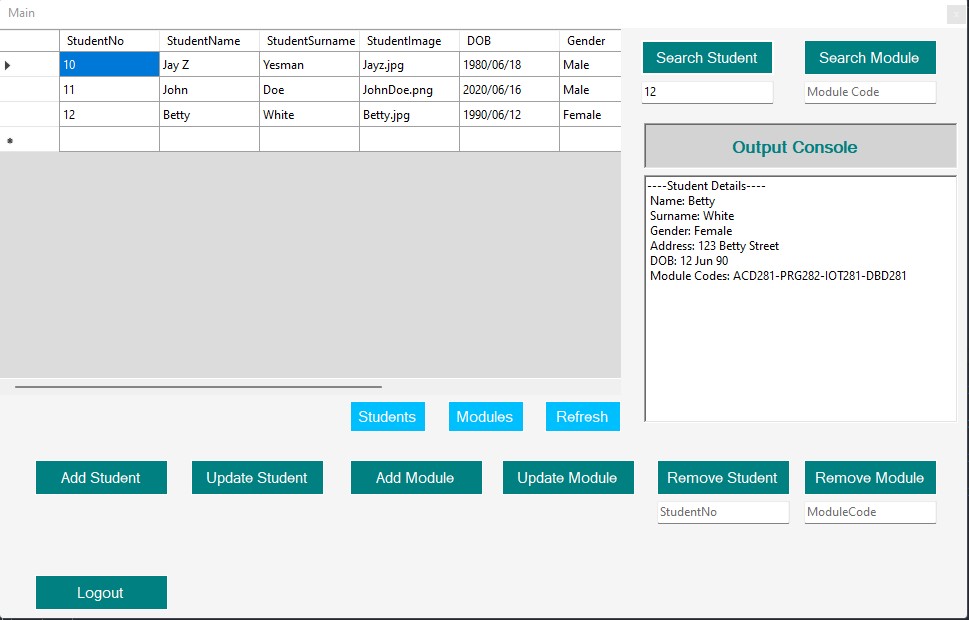
PRG 272 Project



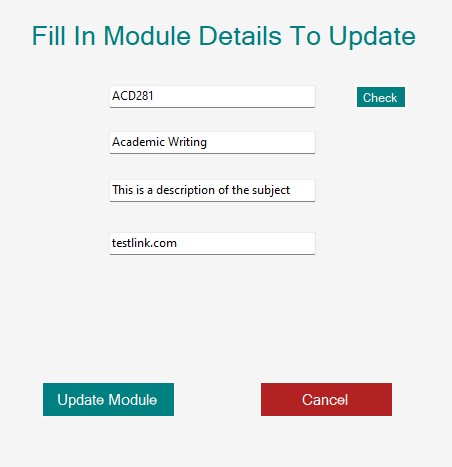
(fig 1. Add module Form)



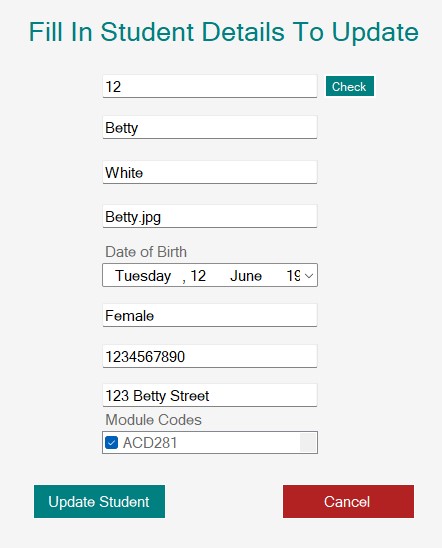
(fig 2. Add Student Form)



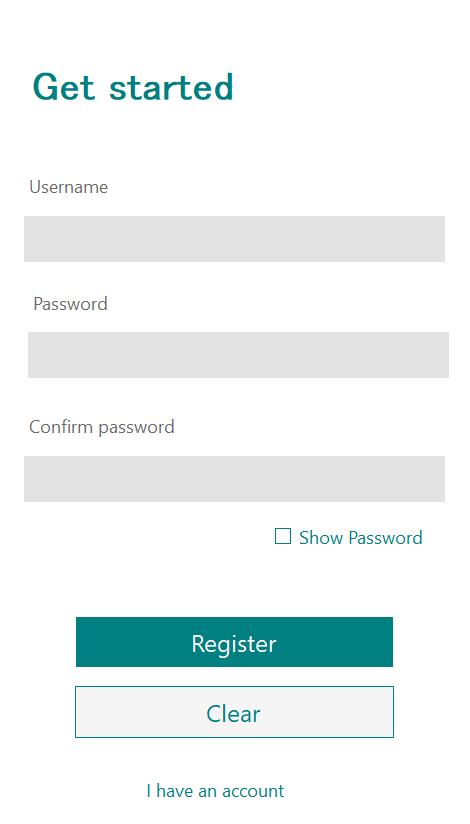
(fig 3. Main Form)



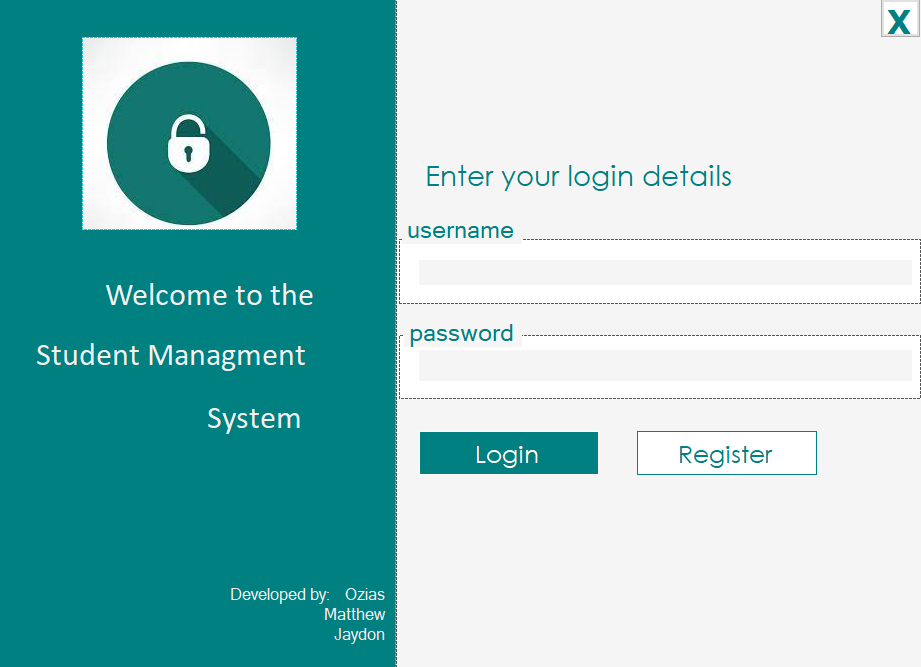
(fig 4. Update Module Form)



(fig 5. Update Student Form)



(Fig 6. Registration form)



(Fig 7. Login Form)

Upon opening the application a form will be created that prompts the user to input a username and corresponding password. The application will then create a data stream to read a text file (Login.txt) which stores the login credentials of each user on one line in the format of: Username, Password.

The stream reader iterates through each line of the text file looking to find a Username that matches the one that had been inputted. If it goes through the entire file without finding a matching username then a message is displayed saying “That user does not exist”. If the username is found however the inputted password does not correspond to the password associated with the given user in the text file then a message is displayed “The password is incorrect” and a counter is incremented that after a certain number of tries the application is locked and the user must wait before attempting to login again.

If a user successfully logs in they will be taken the main form shown in fig 3. From this form the user has the ability to see the details of all existing students and perform operations to; add new students, search the information of a specific student, update a specific students information, search for the information regarding a specific module, update the information of a specific module, delete a module and delete a student.

The creation of a new student will lead to the form shown in fig 2 where the user will enter the Student name, surname, date of birth, gender, phone number, address and select all relevant modules from a drop down menu. Upon hitting the “Add Student” button a new object will be created of class Student which will store the provided information. Once this object has been created a method is called that opens a connection to the database and executes a SQL command to insert the newly created students information into the student table of the database. The textboxes on the form are then cleared and the user can choose to add another student or cancel and return to the main form.

The search student function is called form the main form by entering a students student ID into the text box and clicking the “Search Student” button. This button triggers an event that iterates through the entire student table until it reaches the searched for student then displays all of that students information in the output console.

The search student function is called form the main form by entering a Module code into the text box and clicking the “Search Module” button. This button triggers an event that iterates through the entire Module table until it reaches the requested module then displays all of that modules information in the output console.

Updating a students information prompts the form shown in fig 5. The user must enter the student ID of the students whose information they wish to update and click check. Upon clicking check an event will trigger which calls a method that will iterate through each entry in the student table to find the student with that ID if one exists. If such a student is found the information of that student is used to fill in the textboxes on the form that the user can then edit to how they would like and then click the “Update Student” button which will use the information stored now stored in the textboxes to overwrite the students information in the database using an SQL UPDATE command.

Updating a module works the same way as updating a student by calling the form shown in figure 4 prompting the user to enter a module code and then allowing the user to make any changes they want

Deleting a student works by entering a students Student ID into a text box on the Main form and clicking the delete student button which iterates through the student table until it finds the student and then runs a SQL DELETE command.

Deleting a module works in the same way.