Kyaw Za Yan naing (Student ID: 10238699



[Programming Fundamental]

[CA3 Assignment]

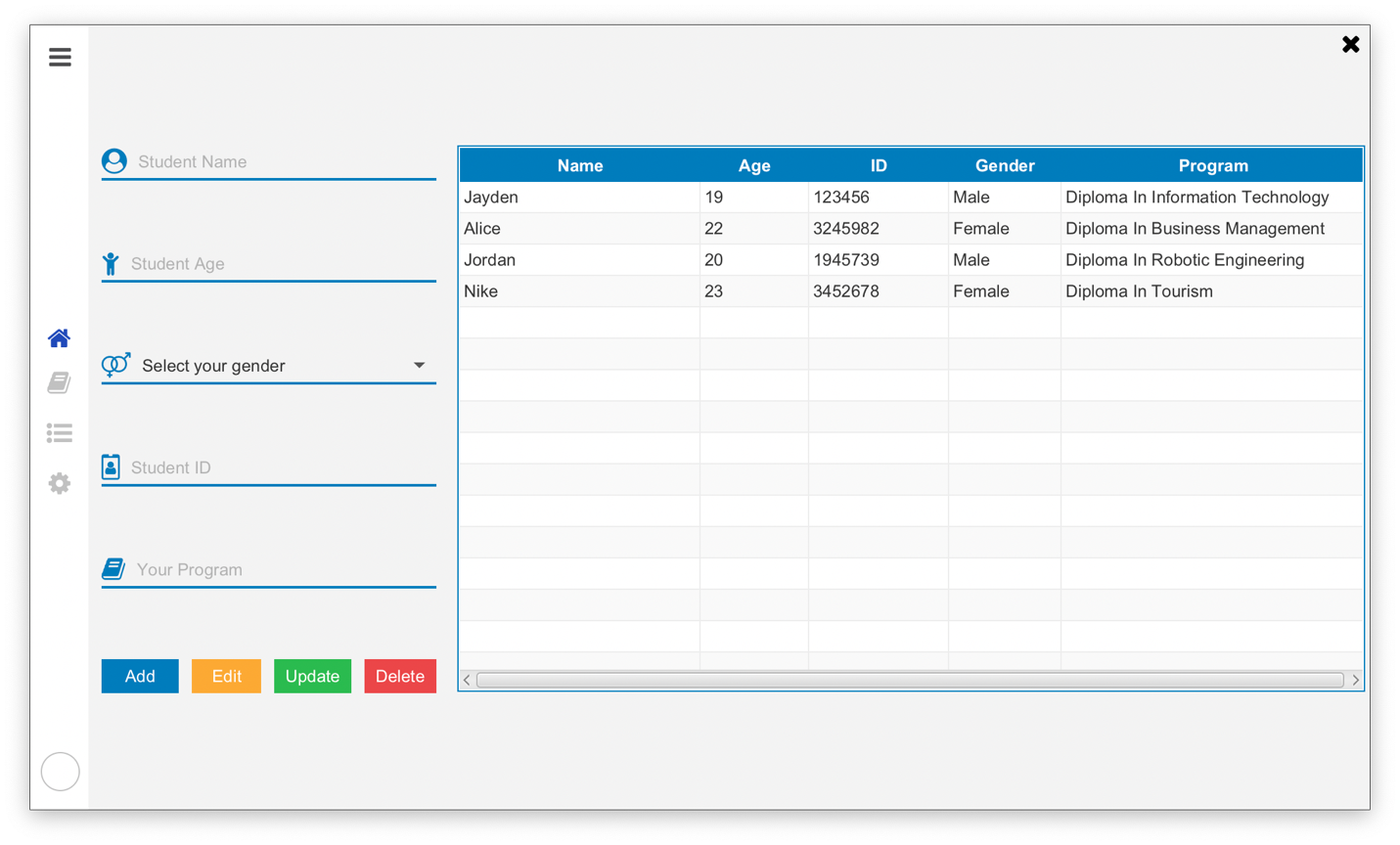
Documentation For Java Assignment

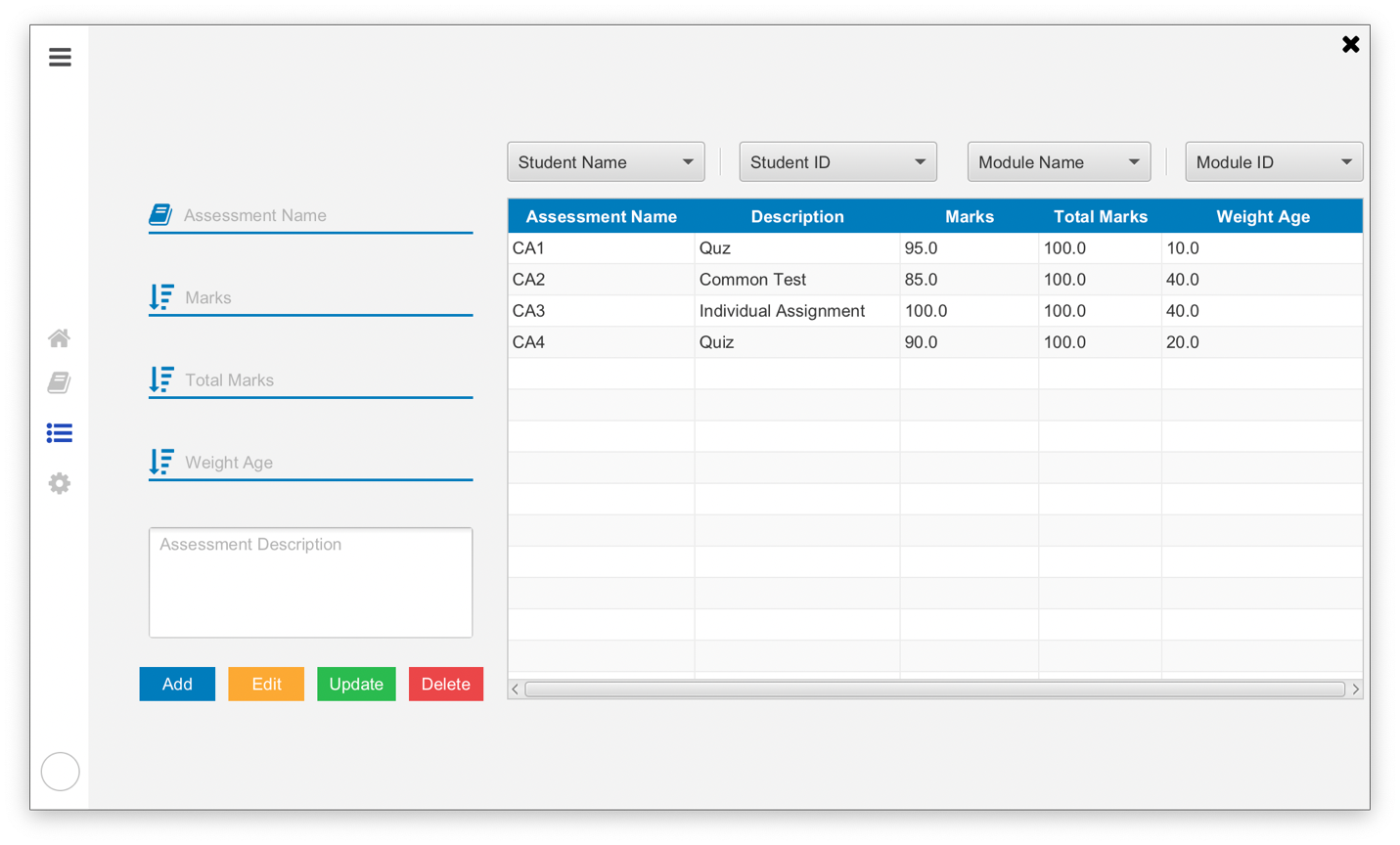
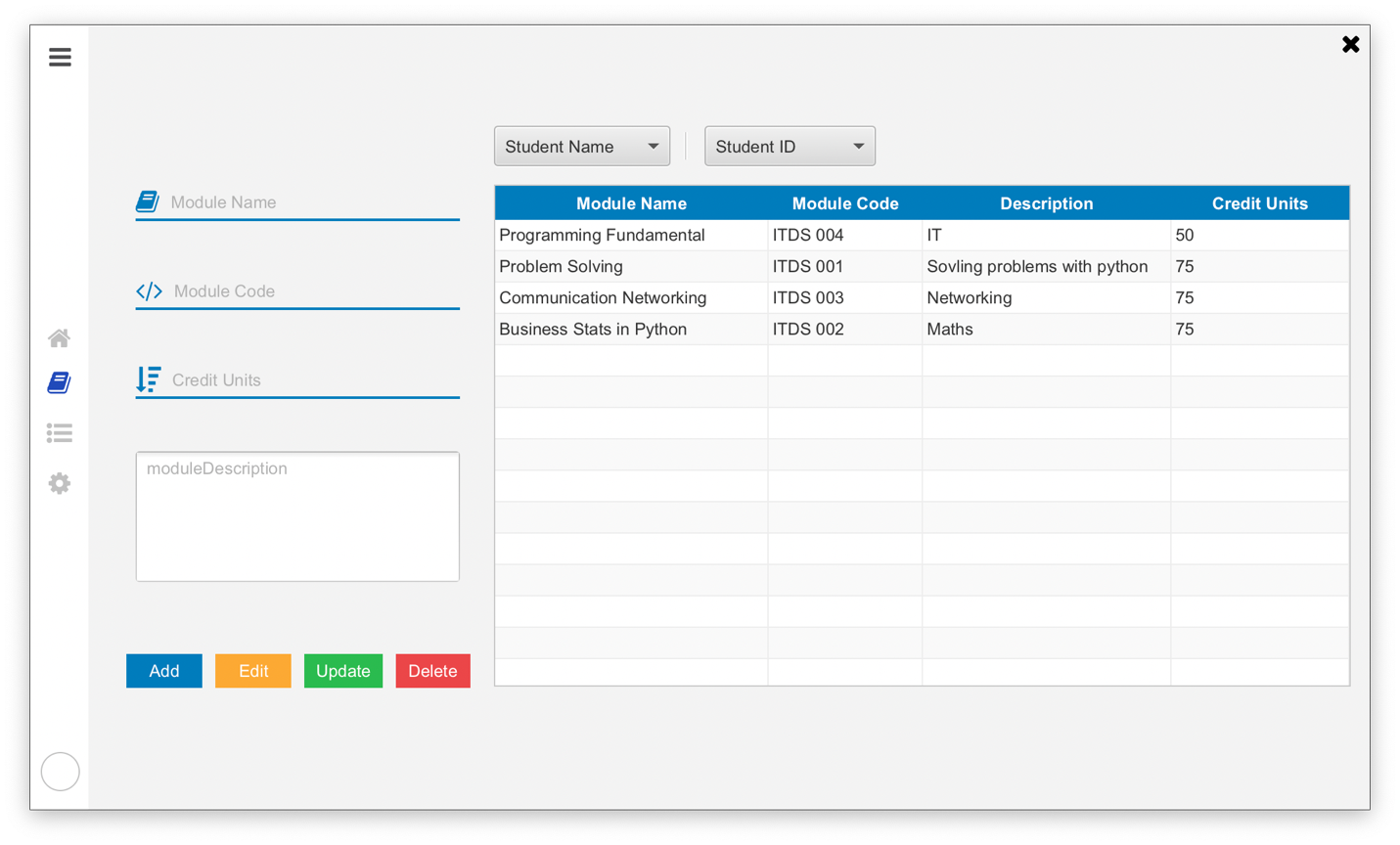
The purpose of this assignment is simple however hard to implement due to its complex data structure. We must store all the student information such as student name, student ID, what modules they are taking, modules specifications and assessment specifications based on the module they take. Its approach is more of a grade tracking program than a management program. And this application is clearly not intended for students it is for teachers and supervisors to keep tracking the students’ grades.

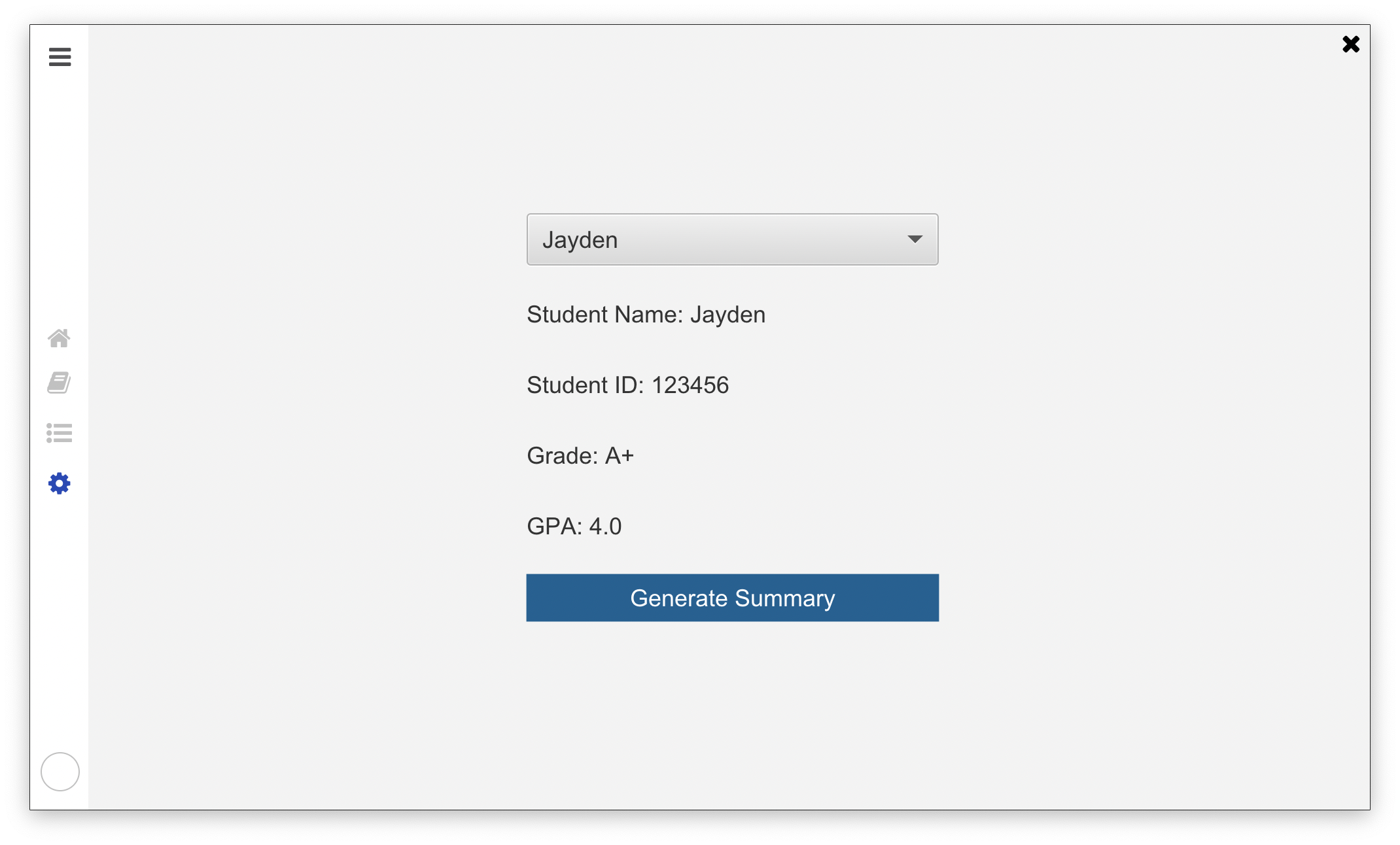
I had to face so many hardships and challenges in order to create this java application. Since java is compiling language and it’s known for object-based programming language and its faster than of the interpreted languages and it includes a high security infrastructure. On the other hand, it is hard to implement in a real-life application for me. Even though it has highly performed data structure, the program it’s self wont store the data forever. Since, I decided to use the Graphical User Interface for the application, I needed a database in order to store data permanently and runs the application faster. I didn’t have enough time to implements a database for this program. I just had to use an ArrayList to store data temporarily. In GUI application, I added some menu navigation to go through the process of CRUD which stands for create, read, update and delete. Every time a scene or a page changed the application is reloaded and the temporary data got deleted. So, I added dummy data to test the application. I created four initial students with modules they are taking and assessments they are taking. This a weakness of the application and If I had more time, I would have created a database with MySQL or SQLite data structure to boost up the performance of the application and reduce the data loss and create more reliable for storing data. I used java version 18.0.1- and third-party package called java which version is 18 along with some packages such as font-awesome which is a library for adding icons in the application and bootstrap for styling the components which saves a lot of time and effort. The routing is simple yet effective. There are four different pages. First page is for creating students, second page is for adding module to the created students, third page is for adding assessments to those created modules and the last but not the least the summary page where we can check the for the summary. In the last page we can check for student’s overall grade and overall GPA score of each student by selecting students from the combo box and render the respective data on the labels. In other three pages, I implement a concept called create, read, update and delete CRUD for students, modules and assessments. You can add data by entering necessary information on the text field and clicking the add button which will add data to the table and to the ArryList. We can edit the added data on the table by selecting the data and clicking the edit data which will render the data on the text fields again we can change whatever we pleased and update them by clicking update button. The same way goes for the deleting by selecting the data and clicking the delete button. I also added error handling methods to handle the user errors and show them what they can do to fix errors by alert errors and warnings.

Implementing the necessary methods to calculate, add data and render them on the page is not the hard part. The hard part is looping through the ArrayList so we can add data in the right place. It is three level ArrayList the first level is the student data, and the second level is the module data which is inside the student ArrayList and the last one is assessment data inside the module ArrayList. So, finding out the exact place to add necessary data is nearly impossible to get on the right track since I had to put them on a data table for data visualization. As a mentioned before, If I had a little more time, I could build an actual database to store, edit and delete data permanently.

Here are screenshots of the applications.







The last page is kind of messy because I couldn’t put the data on the table as I did on the last three pages. This is one of the things I couldn’t solve. The circle thing on the bottom of the menu bar is for the profile picture of the user. For future expectations, I could make a login page for authentication process. For conclusion, the application won’t be ready for real world market because it can’t store data permanently. Once it has that functionality and some changes in UI design, this application will be ready to sell on the open market and I believed this could handle necessary functionalities and could run for long term process.