**Software Engineering Task 2 –**

**Object Oriented Programming**

**Task Definition –**

The main objective/goal of this task is to construct a GUI (Graphical User Interface) application which can take in passwords and assess its security and provide feedback to improve the password’s security.

**Algorithm Design (Pseudocode) –**

The Pseudocode is a universal form of visualizing code and is useful for planning out how the code should be formatted. Note this is only an outline and the final product can change format.

score = 0

feedback = ""

IF password contains a digit:

score = score + 1

feedback = feedback + "Password contains a digit. "

ELSE:

score = score - 1

feedback = feedback + "Password does not contain a digit. "

IF password contains a letter:

score = score + 1

feedback = feedback + "Password contains a letter. "

IF password contains a lowercase letter:

score = score + 2

feedback = feedback + "Password contains a lowercase letter. "

ELSE:

score = score - 2

feedback = feedback + "Password does not contain a lowercase letter. "

IF password contains an uppercase letter:

score = score + 2

feedback = feedback + "Password contains an uppercase letter. "

ELSE:

score = score - 2

feedback = feedback + "Password does not contain an uppercase letter. "

ELSE:

score = score - 1

feedback = feedback + "Password does not contain a letter. "

PRINT feedback

PRINT (score / 10) \* 100

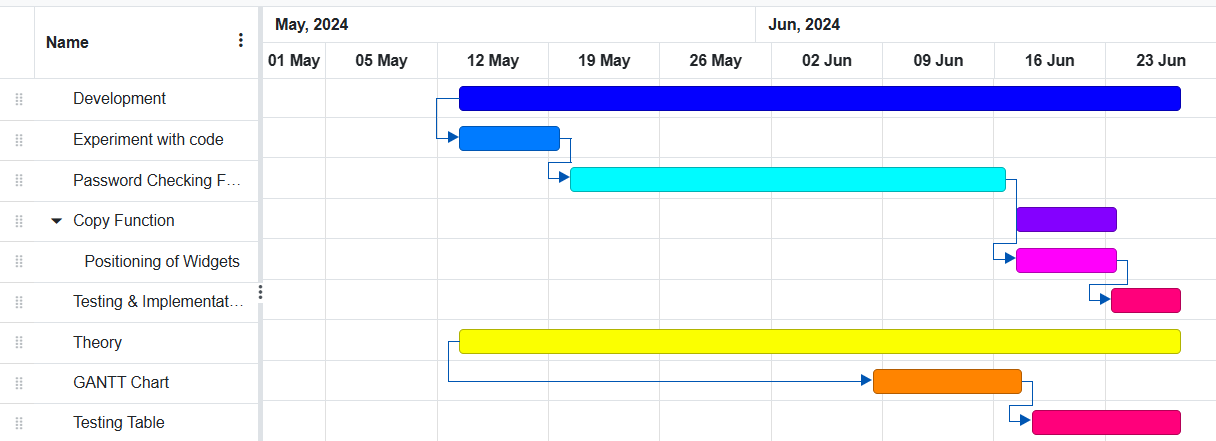
**Storyboard –**

A screenshot of a computer screen

Description automatically generatedThe storyboard aims to highlight a rough draft of what the project aims to achieve and create. Furthermore, it takes into account of the features of GooeyPie and Python.

**GANTT Chart –**

The GANTT Chart is a useful tool in time management, and allows to space out work, and moreover, GANTT Charts can be easily adjusted to suit an ever-changing schedule



**Implementation –**

The code for this application will be stored on a GitHub Repo/Repository. This is considerably an efficient way to look back and to store any progress made. And can be used to recover any lost code

A pixel art of a lock

Description automatically generated

**Locksmith**

*GitHub Repo*: <https://github.com/imamchris/Locksmith-Password-Checker-.git>

**Testing –**

To test for any bugs within the code, a testing table may be used to encapsulate and record any errors found. This may involve passing through different data types or testing the game logic (like if statements or other functions) are functioning as intended.



By using this table any bugs/errors in the code can be dealt with more easily and systematically. After the testing was conducted the following was fixed in the proceeding update…

A screenshot of a computer

Description automatically generated

This shows how by testing and putting results of the tests within a table you are able to root out any errors more easily.

**Evaluation –**

The project overall was a success, and compared to the last project I did (Assessment Task 1), I felt more confident in what I was doing and did not rely so heavily on ChatGPT and only used it to assist me if I had an error in my code which I could not figure out how to solve or I was unsure on how to code a specific feature or function I wanted to perform in my code (e.g. calculating the score/rating for the password), and I was also asking a few more questions on the possibilities with the code and what I could do with it.

This project also had very few roadblocks, and I was successful in making three releases (1 Pre-Release, the rest official Releases).

However, I think the most difficult part of the task was the Theory. I tried to start the theory first, however, inevitably I ended up doing a lot of tinkering and experimenting with the coding and doing a bit of the theory at the same time.

For the theory, I first did the task definition, then I quickly and very easily did the storyboard, but when I got up to the GANTT Chart I found it was the most difficult to work with, I had very little idea how the site worked and had to experiment on my own and it became very tedious process of planning out my schedule and moving around the boxes, and took a while to do and to master, another further challenge was the pseudocode, since I hadn’t done it in a long time I had forgotten how to do it and had to resort to ChatGPT, the internet, and my class notes to help guide me in writing my pseudocode. However, the rest of the theory was simple and easy. The Implementation did not take long along with the testing of the code.

Overall, I think this project was a success and helped in better understanding of what I should do and acts steppingstone to help me further improve my work on these assessments and for any future projects.