

PROG6212 - Part 1

ST10305921 - Zoe Heyneke



GitHub link:

<https://github.com/Zoe-Heyneke/Part-1---Prototype.git>

**Design Choice Explanation:**

The design choices were focused on creating an effective and easy interaction for users to navigate through the application.

Each View is designed separately to avoid cluttering the screen. Headings are designed in a large font and underlined to assure users what page they are on. Buttons are enlarged and designed with colour to create a visibly appealing association such as using the colour green to show approval of a claim and red for rejection of a claim. Subheadings are provided and field descriptions are added to ensure users know what details to insert.

Overall, the design delivers a positive user experience ensuring that the user will enjoy using the application.

**Structure of Database:**

The structure of the database is divided into all the tables. Tables are created according to the entities. Keys are included to ensure relationship is present between tables.

Tables:

* Users
* Lecturer
* Programme Coordinator and Academic Manager
* Claims
* Courses
* Documents

**Layout of GUI:**

The layout of the GUI is designed in such a way that it is innovative and unique to how the scenario was given. Currently there are three Views created in the layout, moving forward to the other parts of the POE, additional views will be created accordingly. Firstly, the Home Page is created to allow separation of roles ensuring security of claim management. Here users can select their role (Lecturer or PC and AM) and will only be directed to their corresponding View.

When Lecture Role is selected, the user is directed to Lecturer View where they provide information to Submit a Claim: Lecturer’s name and surname, course and group number, hours worked, hourly rate, upload documents and inserting additional notes. Each section mentioned is laid out horizontally ensuring that the user completes all fields under the supporting headings.

For the Programme Coordinator and Academic Manager View, users are presented with each Claim in a neat and organized structure, where their duties are to Approve or Reject Claims.

The layout of the GUI also includes supporting headings to ensure a user-friendly experience where users will know what information is needed. Buttons are also visibly shown to guide the user where to click to perform specific actions. Consideration of the layout is taken by creating equal spacing and not making the GUI too cluttered.

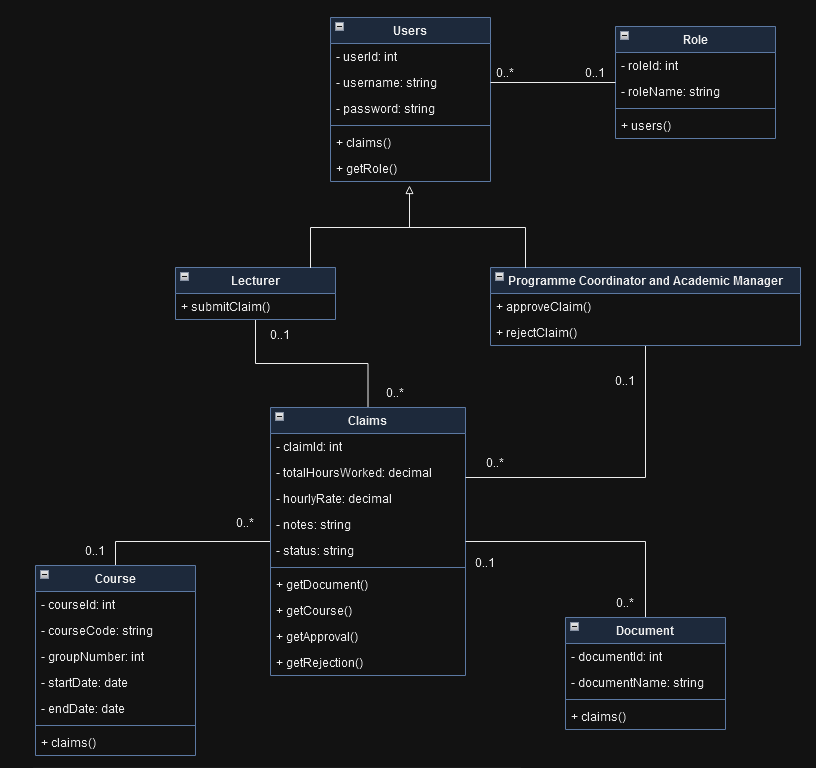
**Assumptions or Constraints:**

It is assumed that user separation is very important as claims can only be submitted by Lecturers and claims can only be approved or rejected by Programme Coordinator and Academic Manager. A user can be a Lecturer or a PC and AM. Therefore, a user sign up and log in feature is created to separate these duties.

Project constraints must ensure that only Lecturers can submit claims by also validating user input, additionally support any file type (pdf, docx, xlsx) and only Programme Coordinator and Academic Manager can approve or reject claims.

**UML Class Diagram:**

The following UML Class Diagram represent all the classes along with their attributes, methods and relationships.

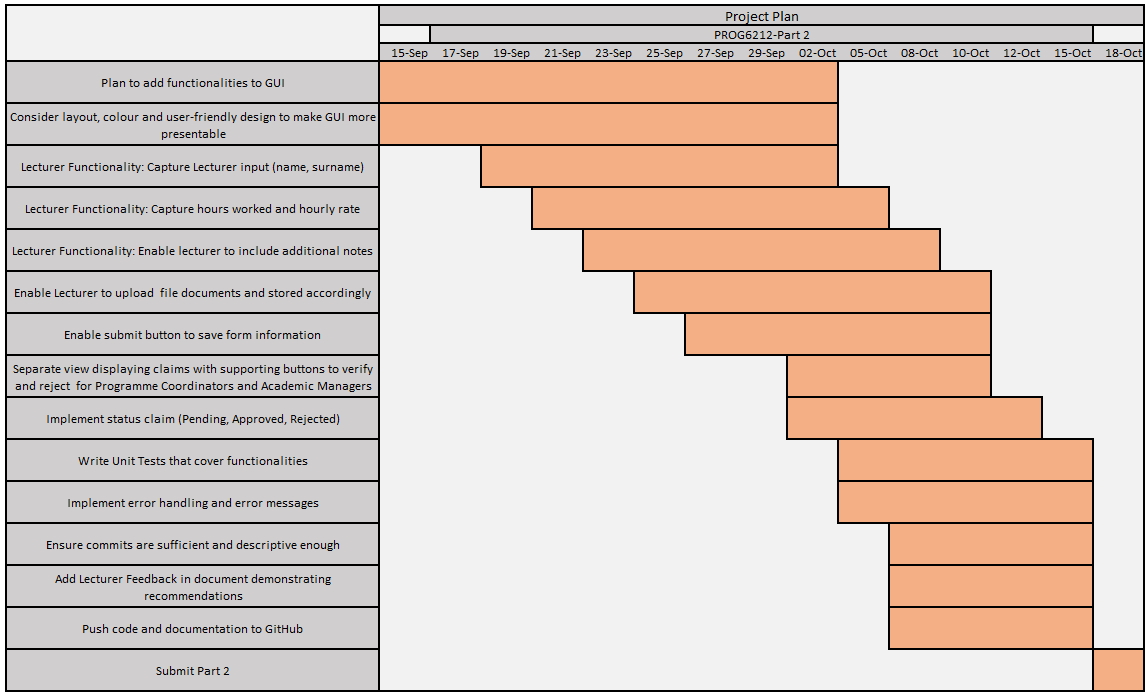


**Project Plan:**

Graphical representation formatted in a Gantt Chart outlining all tasks from Part 1, Part 2, and POE.

A screenshot of a computer

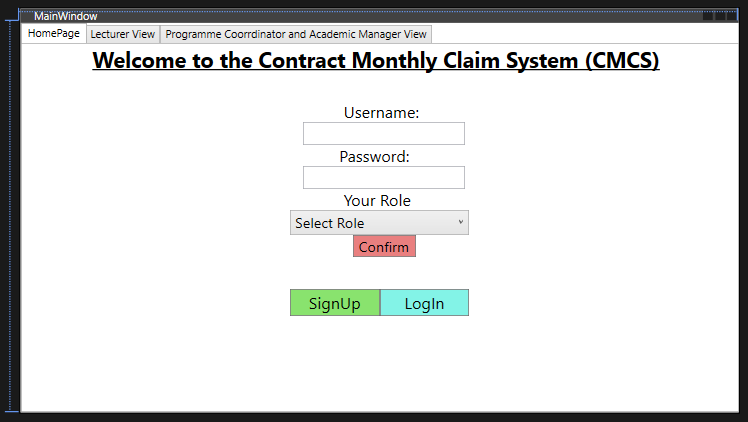
Description automatically generated



A screenshot of a computer

Description automatically generated

**GUI Design:**



A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

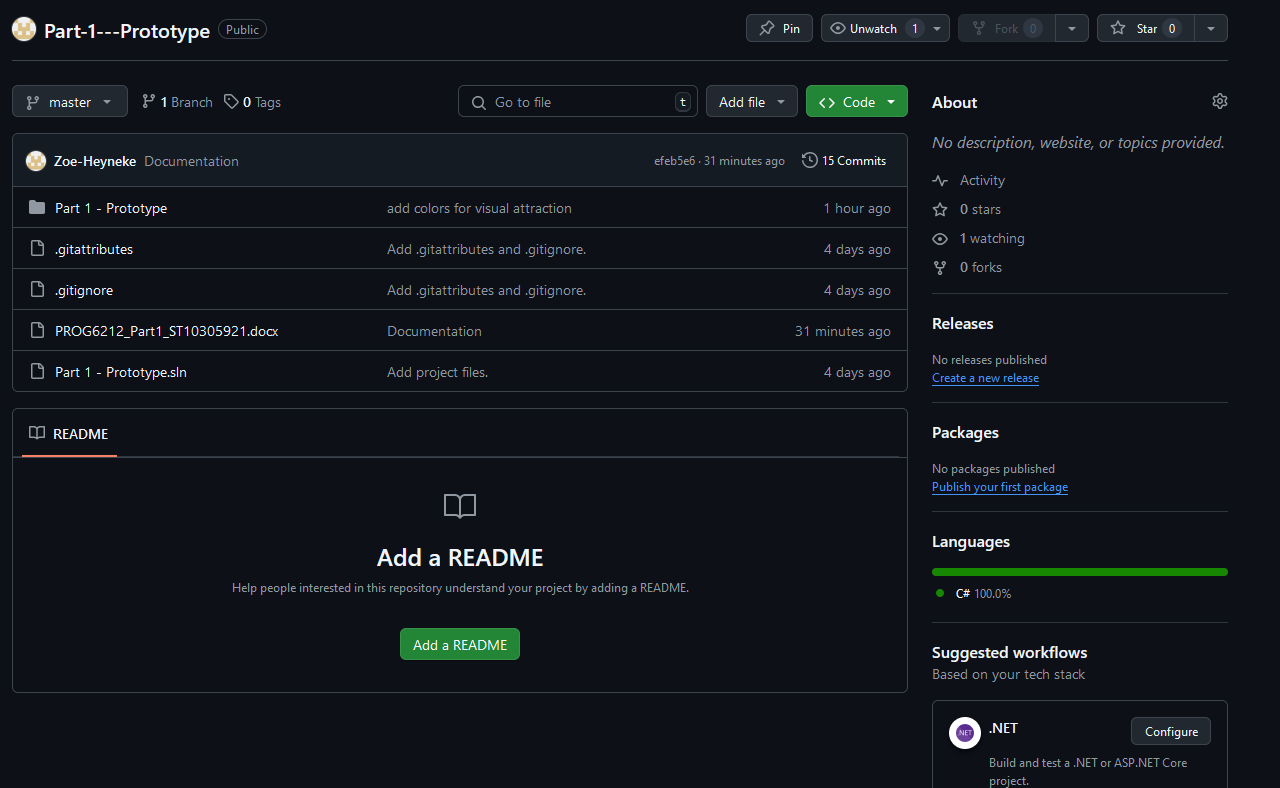
**Total Words: 498**

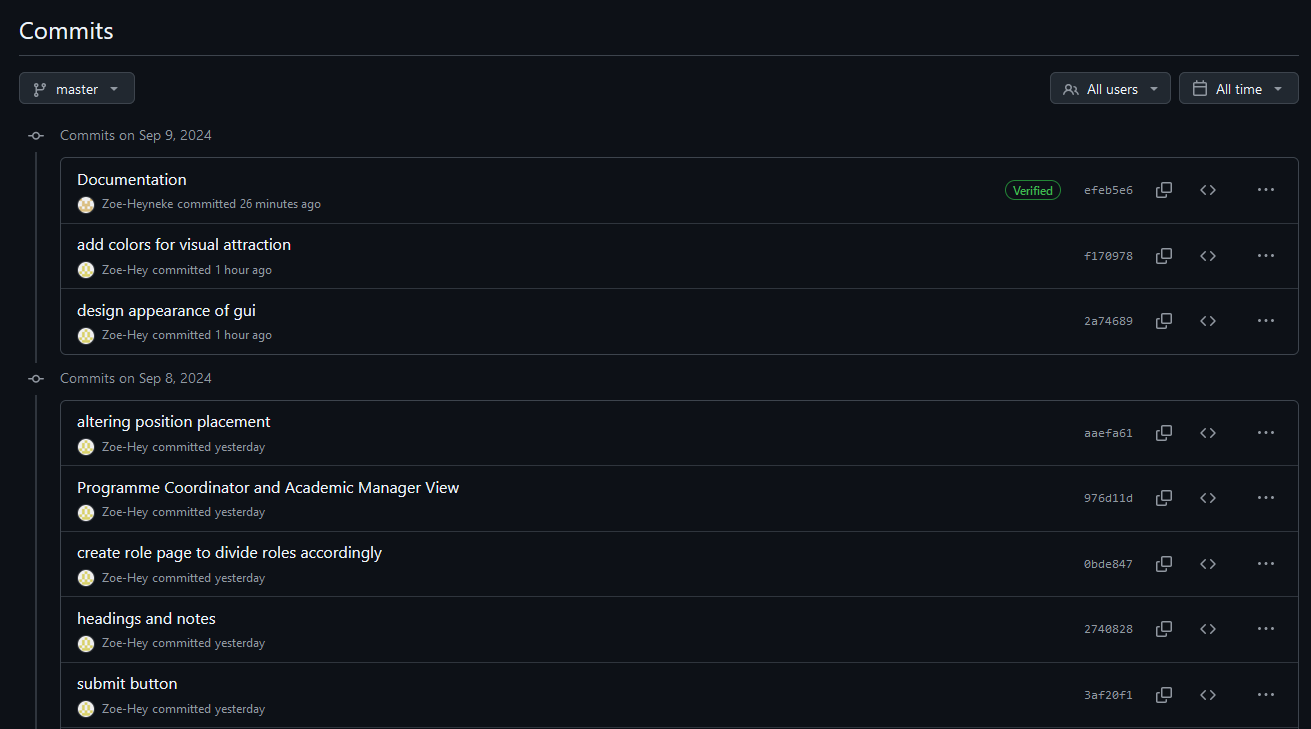
**GitHub Repository Link:**

<https://github.com/Zoe-Heyneke/Part-1---Prototype.git>

Below shows GitHub commits containing pushed code and updated documentation.

(15 Commits)





A black background with white lines

Description automatically generated