**KHUMO TWALA**

**ST10382111**

**PROG212 PART 1**

**VARSITY COLLEGE BACHELOR OF SCIENCE IN COMPUTER & INFORMATION SCIENCE IN APPLICATION DEVELOPMENT**

**-------------------------------------------------------------------------------------------------------------------**

**DOCUMENTATION**

**Introduction**

The CMCS, Contact Monthly Claim System is a web solution project that uses SQLite as its backend developed using ASP.NET Core MVC. As per specification, the main goal of this web solution is to make the submitting process and the approval of monthly claims for independent contractors, lecturers to be easier. This makes sure that there is overall accountability for each stage while promoting transparency, because the system will put an end to ineffectiveness discovered in manual claim submissions. This structured workflow will give lecturer log hours; coordinators will authenticate verifying details and the academic managers will give the final approval. The parameters of this part 1 focus on project planning and creating visual representation demonstrating the layout, workflow and system design.

**Design Choices**

A role based structure is used for the design of the Contact Monthly Claim System. Lecturers can create new claims through access to a dashboard that also permits them to view claim history. Each group of users can directly interact with the system. Furthermore, being present is a dashboard set apart for Programme coordinators and Academic managers which allows them to approve workflows. Because of this structure, clarity is ensured, and the chances of errors is drastically reduced if not illuminated.

For the layout design, a menu for navigation is promoting convenience in availability and consistency. To reduce user effort, all buttons as well as forms are rationally grouped with also an addition of a simple modest colour scheme. Furthermore, green is used when claims are approved, red is used when claims are rejected, the background is white with navy blue headers and menu. With this visual presence, cognitive load is decreased, and the colour scheme makes sure that updates are visible to any user that is using the web.

SQLite was the preferable database that was chosen because it satisfies the demands of prototyping at the same time leaving room for scalability in the future. On top of that, the entire database is one file which makes it a simple database to use, also it is has coherent integration with application that are .NET based (Dodds, 2024). These are the agile tools used to manage project management: Jira and ClickUp. With Jira, Kanban Board was used for tracking tasks, ClickUp as well was used for Gantt Charts and the timeline. With these tools, best software development practices are achieved because of a schedule drafted in detail and an iterative workflow (Atlassian, 2025).

**Assumptions:**

Approvals are made in honest intentions by the programme coordinators and academic managers, as the Lecturers are in good faith expected to enter hours honestly.

**Constraints:**

All uploaded files have to be in either .pdf, .docx, or .xlsx formats whereby all claims are subject to one single month.

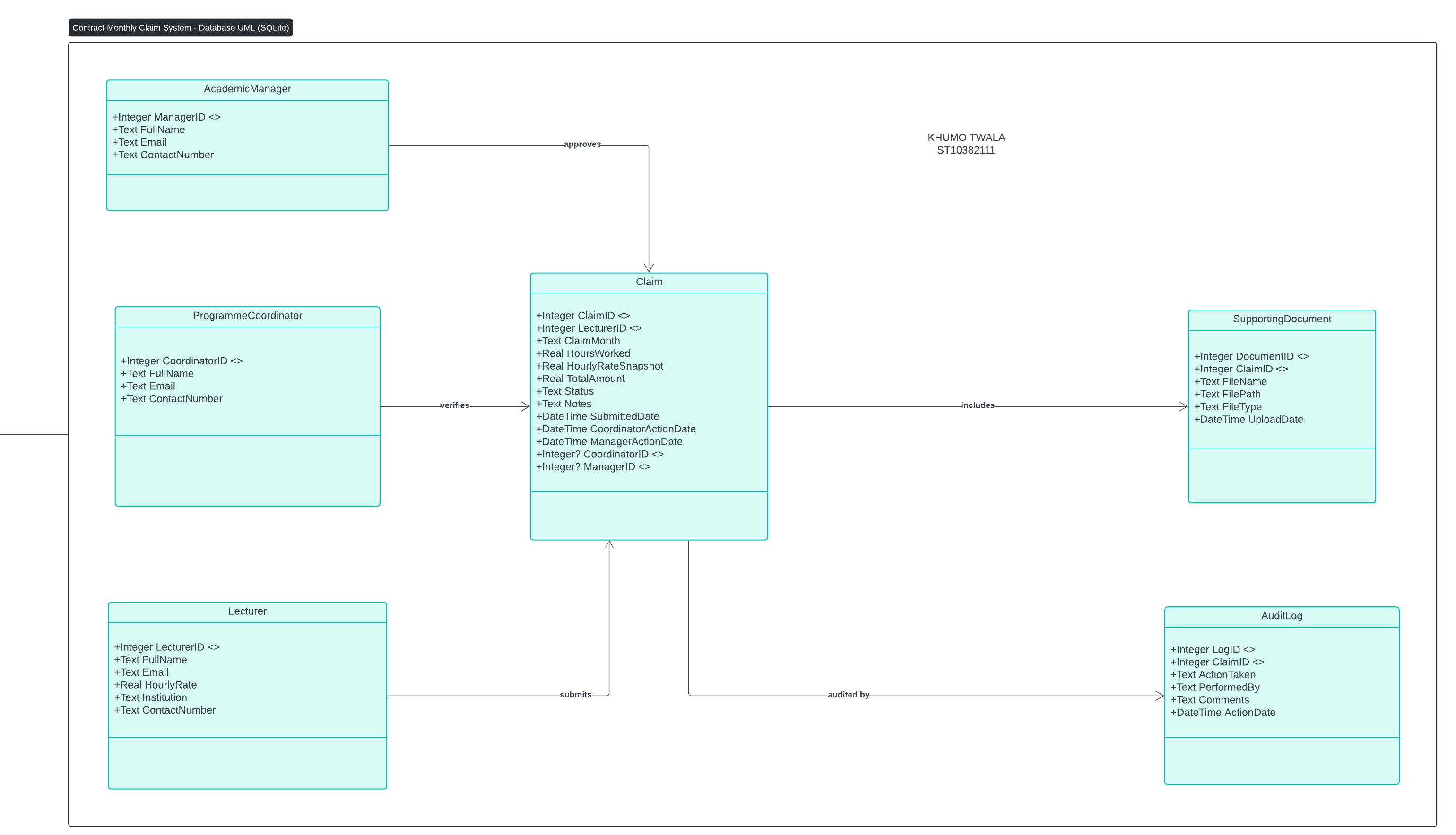
**UML Class Diagram**

The Five key entities are as follows:

Lecturer, Claim, ProgrammeCoordinator, AcademicManager and SupportingDocuments.

The relationship between entities makes sure that a Lecturer can submit many claims and with each claim, there being a document that is supporting the claim.

For approval, each claim firstly goes through the programme coordinator then secondly to the academic manager. The diagram aligns with INTEGER, TEXT, DATENTIME and REAL, promoting integrity of data and compatibility with SQLite. Also giving an auditable design structure having verification in multiple stages.

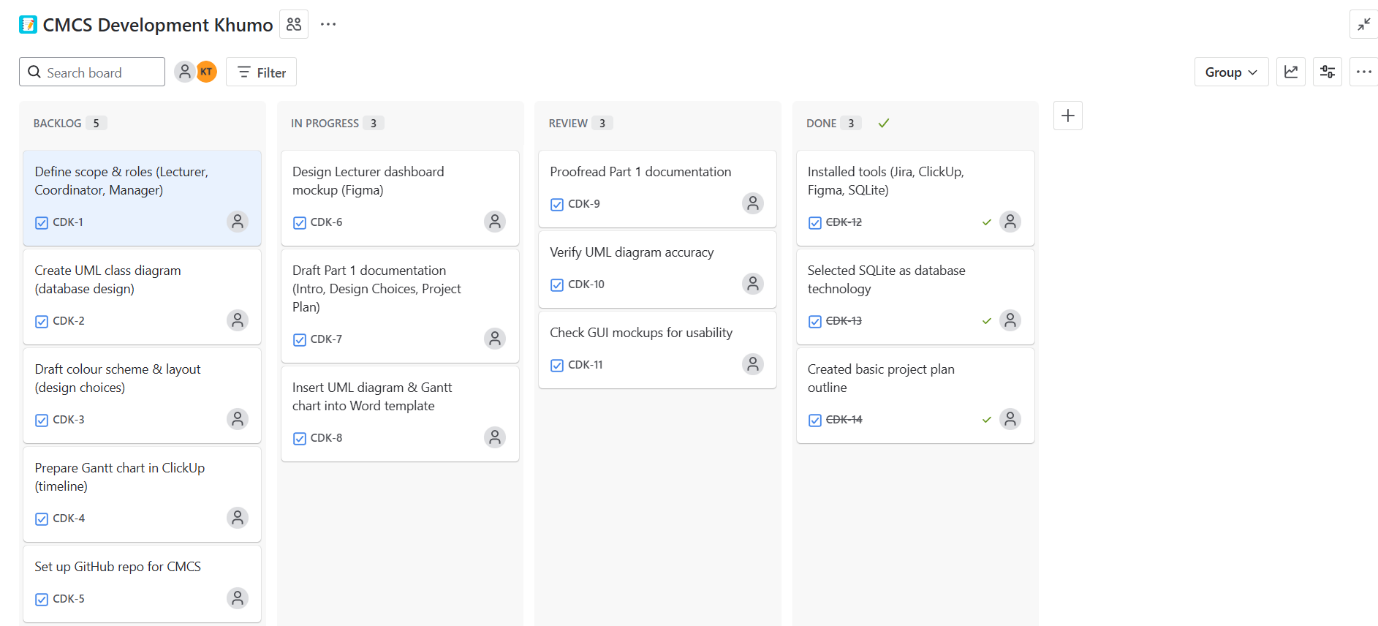
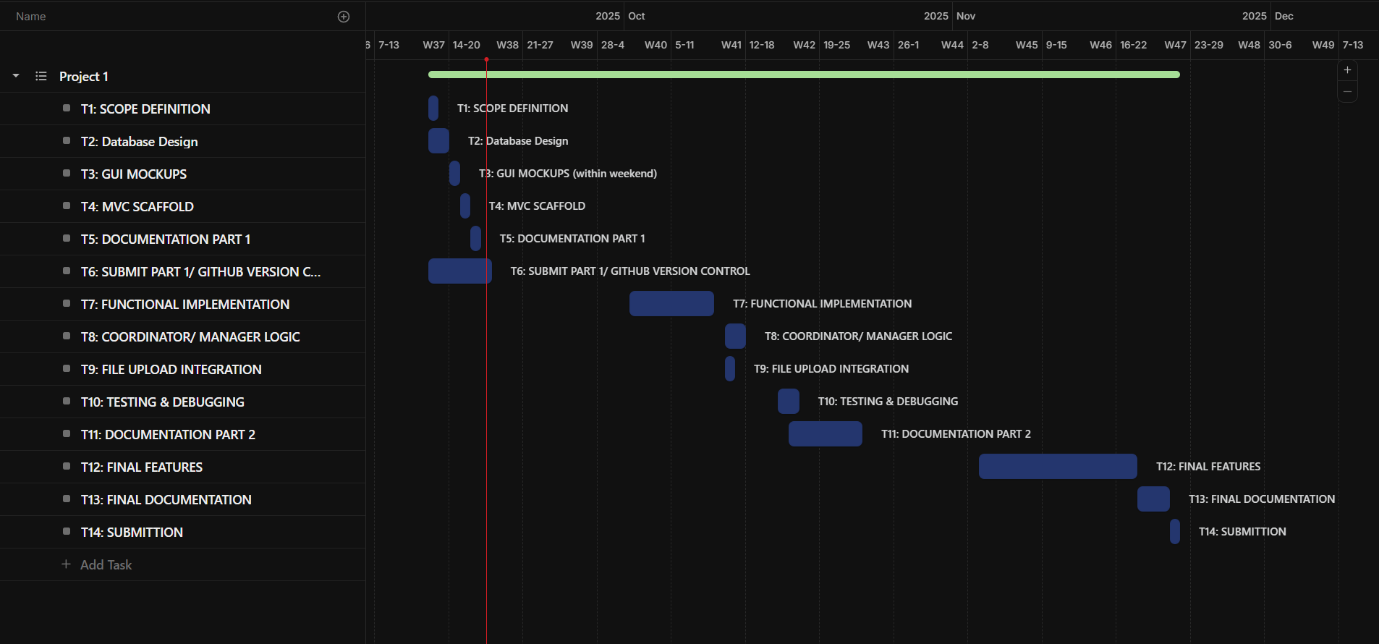
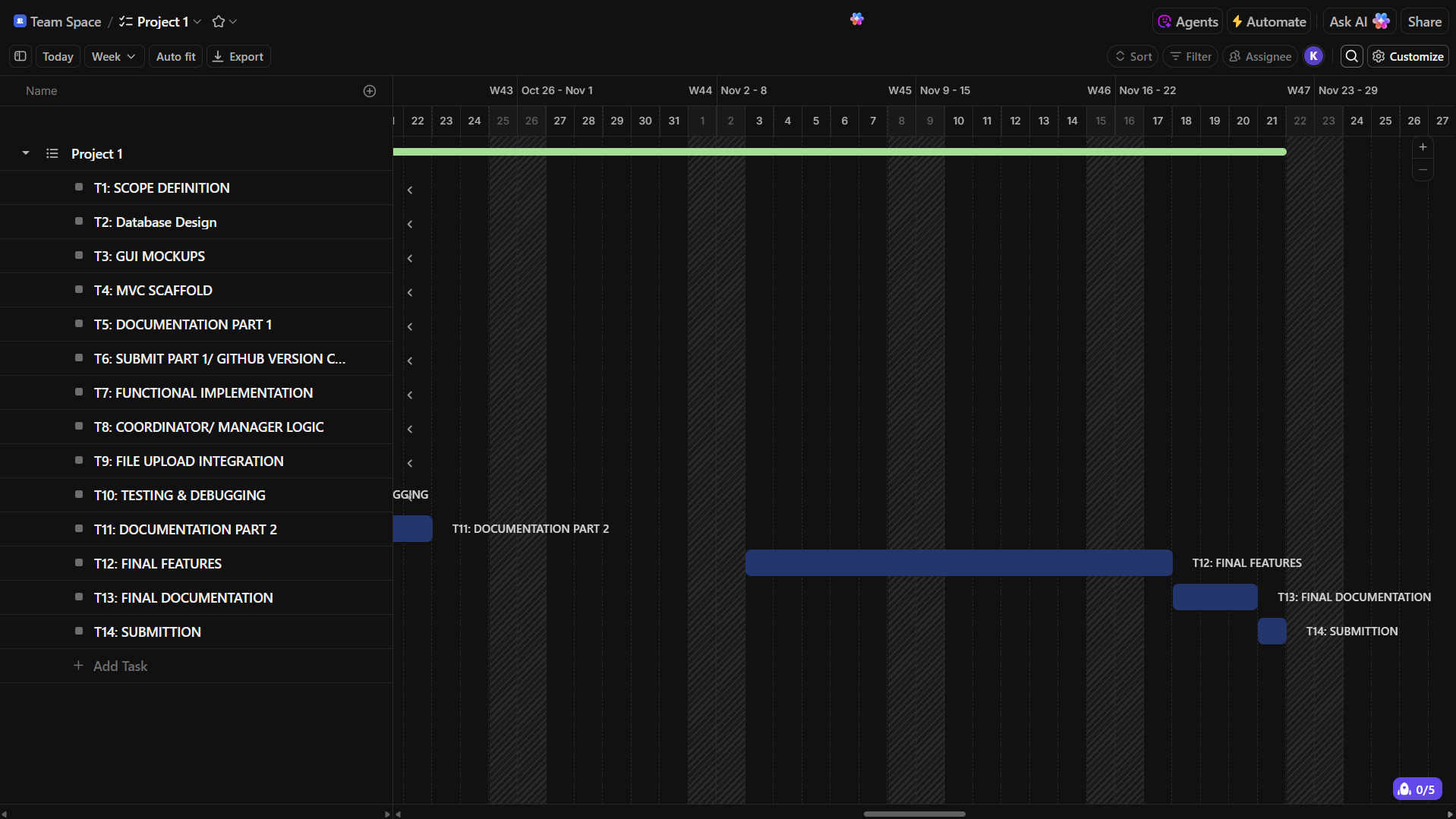
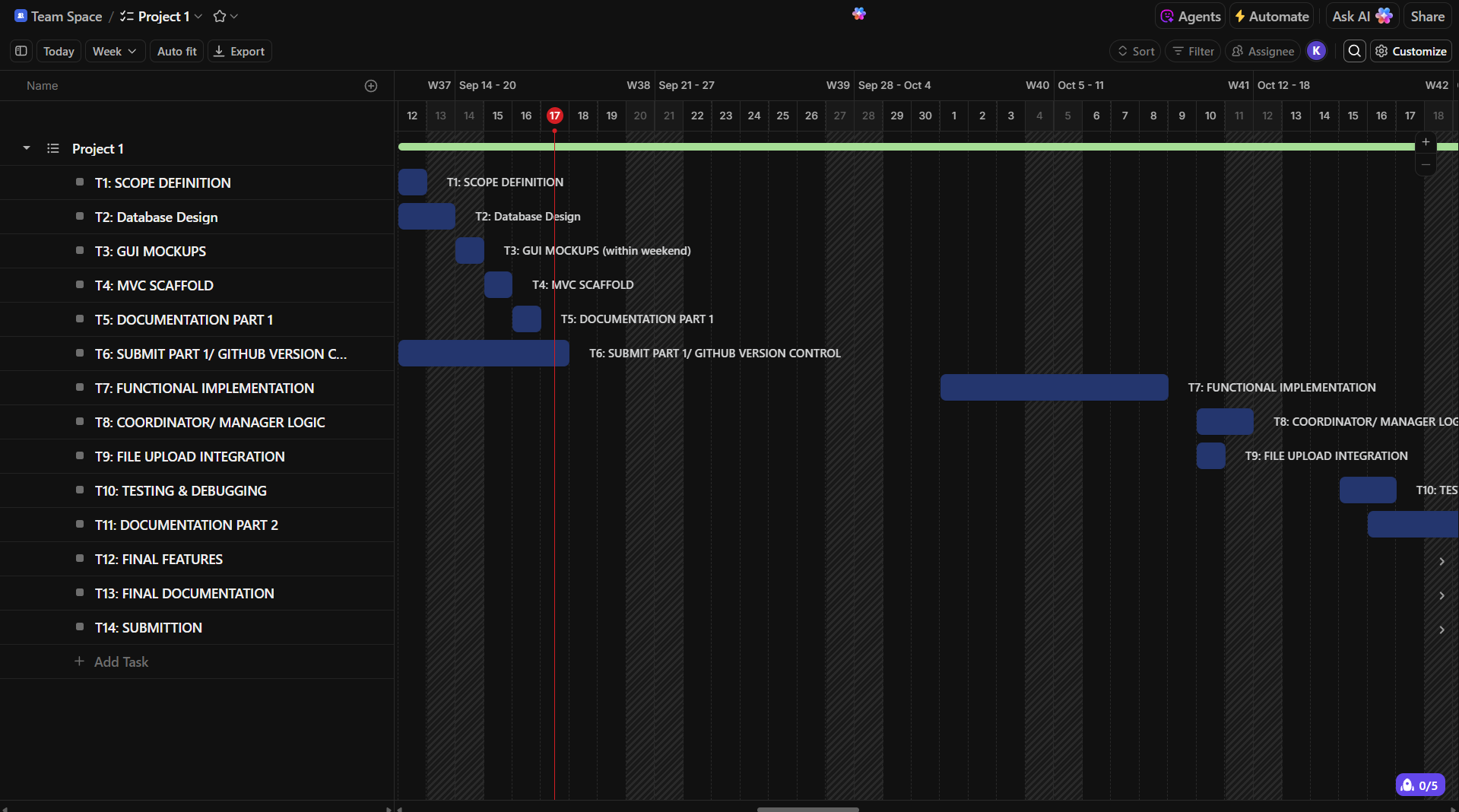


**Project Plan**

A hybrid approach was used in planning the project while also using Kanban timeline for scheduling and management. When using Jira there were different tasks used for each phase.

Using Backlog, these were the tasks used to keep track of the progress using Kanban Board, with “IN PROGRESS”, “REVIEW” and “COMPLETE” also used as visualised cards.

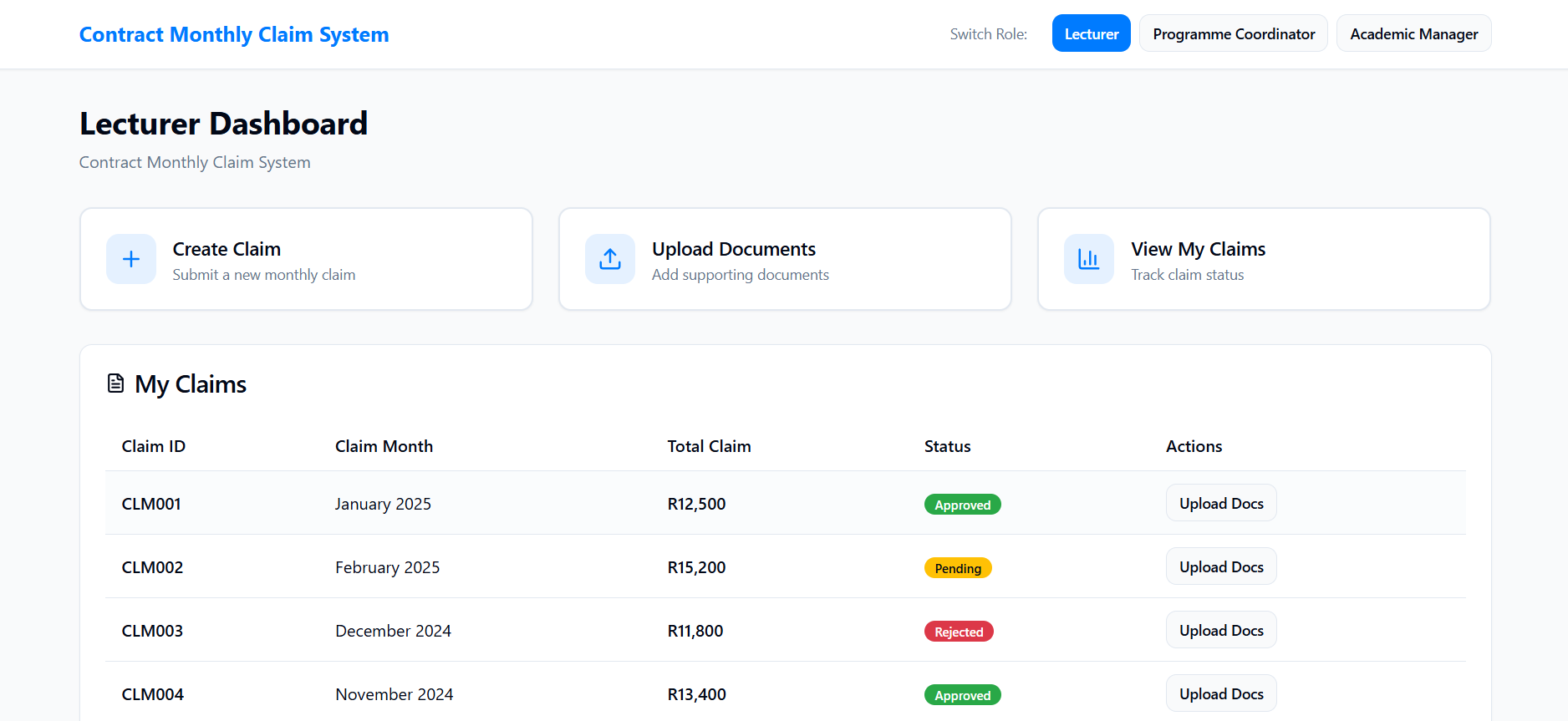
To give light to dependencies between tasks as well as dates that are milestones, Grantt Charts were created through ClickUp, while also making sure that the structure of the project remained intact at the same time having dates that are not assigned to any task to make the project be moveable for changes.

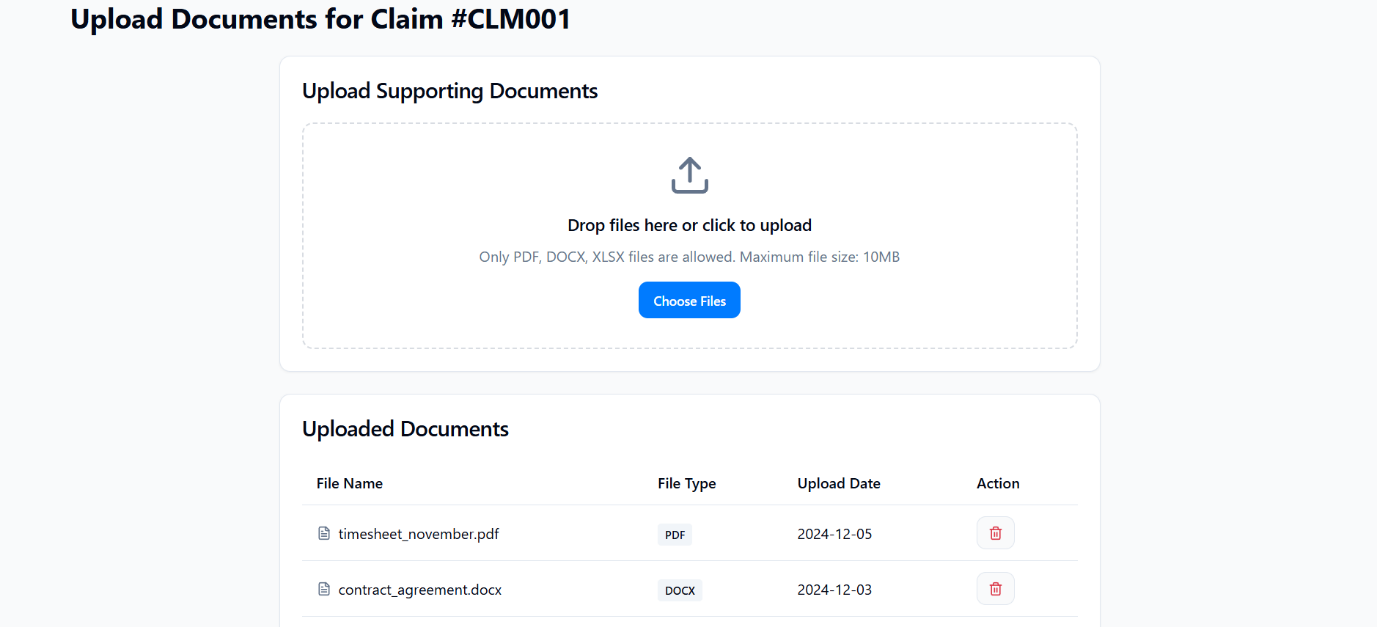


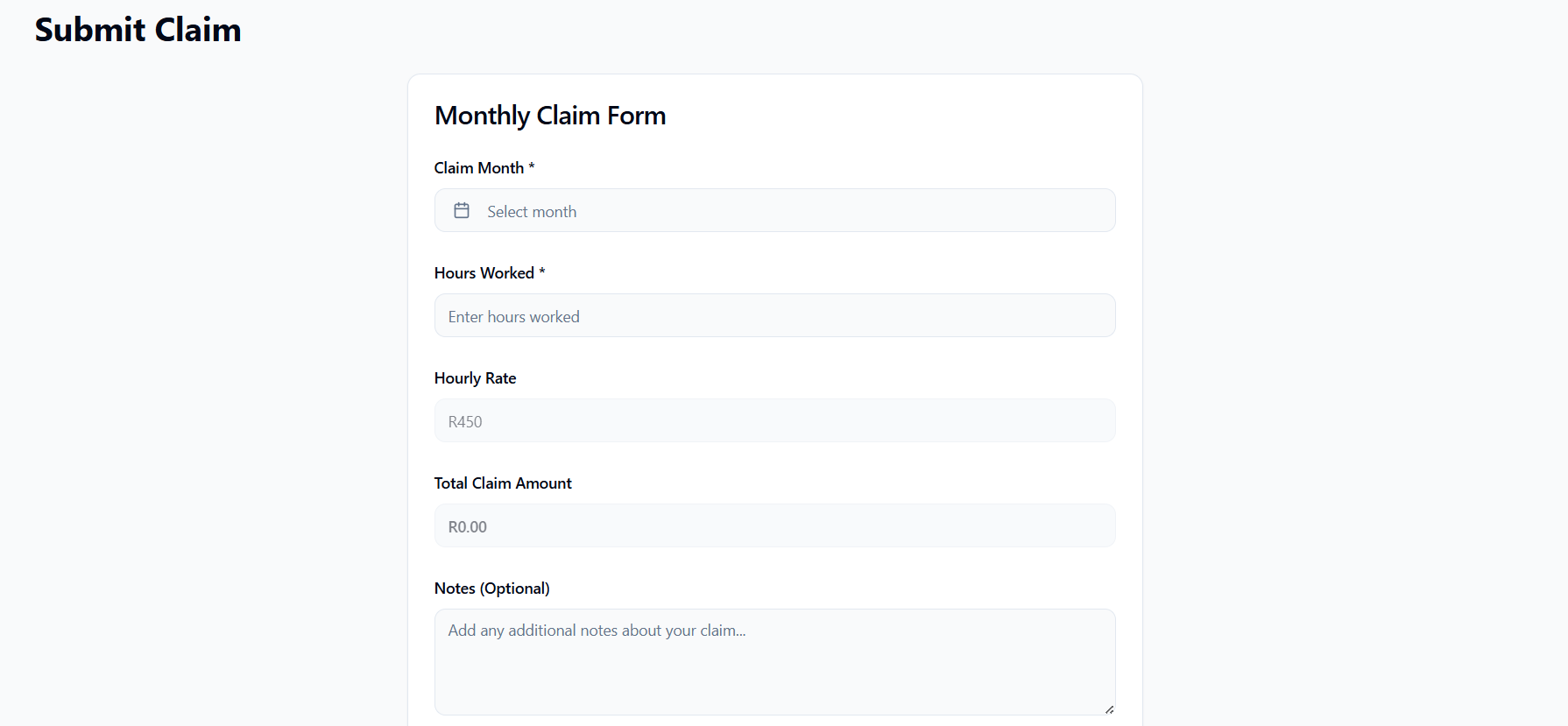
**GUI/UI Layout**

The web app interface was designed using FIGMA. The dashboard gives a speedy access for the Lecturer to claim functions, while the Programme Coordinator view show a list of the pending claims made by the Lecturer for verification, and the Manager has a view which shows all the claims that need the final approval coming from the Administrator. All the buttons, the forms, as well as the indicator that shows the claim status, are put in a position that allows the user to frequently use them, also making them consistent throughout.

LECTURER DASHBOARD

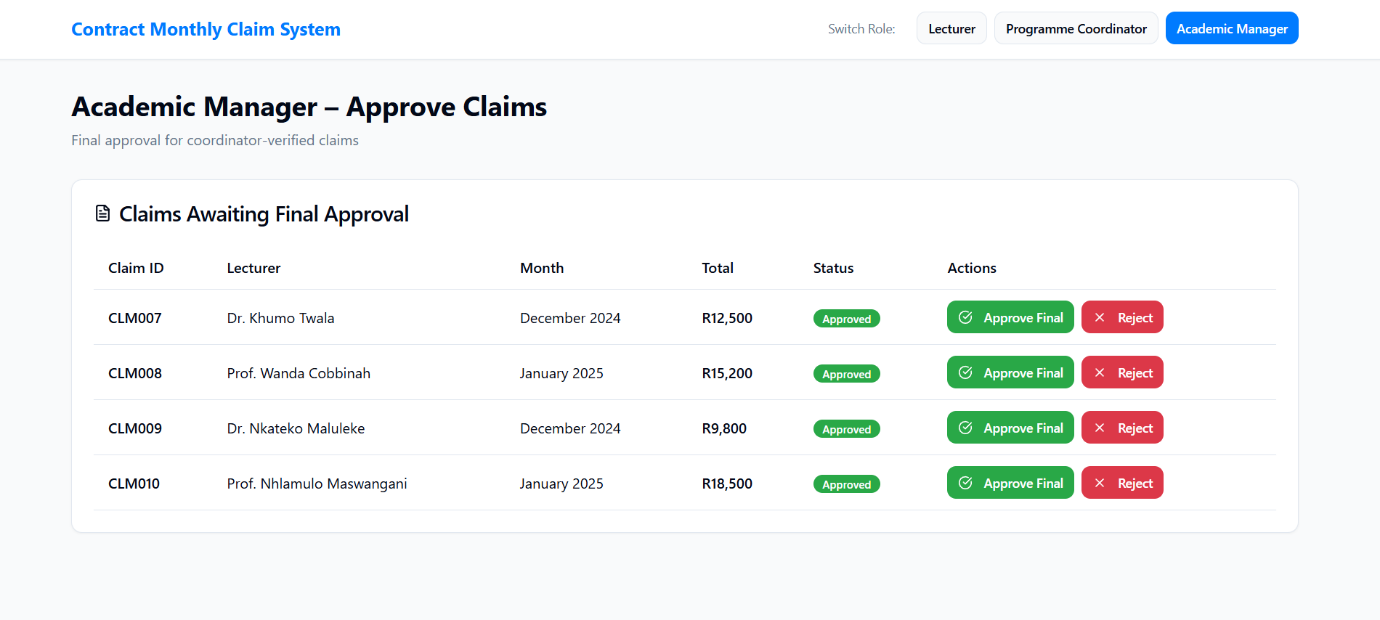






A screenshot of a computer

AI-generated content may be incorrect.



# References

Atlassian. (2025, September 15). *Build world-class software with Jira*. Retrieved from atlassian.com: https://www.atlassian.com/jira/solutions/agile-project-management

Dodds, K. C. (2024, March 27). *Why you should probably be using SQLite*. Retrieved from Epicweb,dev: https://www.epicweb.dev/why-you-should-probably-be-using-sqlite

Engheim, E. (2017, December 10). *Understanding Visual Layout in GUI Design*. Retrieved from medium.com: https://erik-engheim.medium.com/understanding-visual-layout-in-gui-design-2fa91792edef

Lucidchart. (2025, September 17). *UML DIAGRAM*. Retrieved from lucidchart.com: https://lucid.app/lucidchart/c7526882-9b96-4e96-bf1a-197ae3317dd4/edit?loaddialog=TemplatePicker&beaconFlowId=F0DF84B8DC6B66E5&from\_internal=true#?templateid=NULL