**1. Documentation**

The Contract Monthly Claim System was designed with the main objective of to simplify how lecturers submit their monthly claims, how documents are attached and how approvals are handled by programme coordinators and managers. The database structure is based on the relationships between users, lecturers, claims, supporting documents, and approvals.

The reasoning behind my design choices was to keep the process transparent while also making it easy for lecturers to create and track claims. Claims often involve multiple documents and need formal approval, so I structured the database around these entities. The User table is the foundation since both lecturers and approvers fall under users, differentiated by their assigned roles. The Lecturer entity extends this further, keeping contract-related details such as employee numbers and hourly rates.

Ambler, S.W. (2005)

The Claim table captures all claim submissions and is linked to lecturers. Supporting evidence like timesheets or proof documents are kept in the Document entity, while the Approval entity makes sure of a proper review process. The design ensures that every claim has an audit trail from submission to approval.

Some of the assumptions I considered are:

* All lecturers have a working and valid user account before they can submit claims.
* Hourly rates are set and kept in the system.
* Approvers can only review claims within their assigned role and permission level.

These decisions ensure that the system can scale and provide accurate, reliable information to both staff and management.

**2. UML Class Diagram**

The UML class diagram illustrates how the classes are structured and connected. Each class shows the key attributes and operations. For example, the *Claim* class includes methods like SubmitClaim(), CalculateAmount(), and AddDocument(). The relationships between classes also reflect the business process, a lecturer can create multiple claims, each claim can have multiple documents and each claim can go through one or more approval stages. Ambler, S.W. (2005)

**3. Project Plan**

The development of the prototype will be divided into straight forward tasks (not everything excel sheet made below):

1. **Requirement Analysis** – Confirm the business rules, lecturer workflows, and approval processes.
2. **Database Design** – Build the schema based on the UML class diagram.
3. **WPF Front-End Prototype** – Create non-functional UI screens where lecturers can “submit” a claim, upload documents, and track the approval status.
4. **Approval Screens** – Build mock pages for programme coordinators and managers to review claims.
5. **Testing & Refinement** – Check if the layout is user-friendly and easy to navigate.

Sommerville, I. (2015)

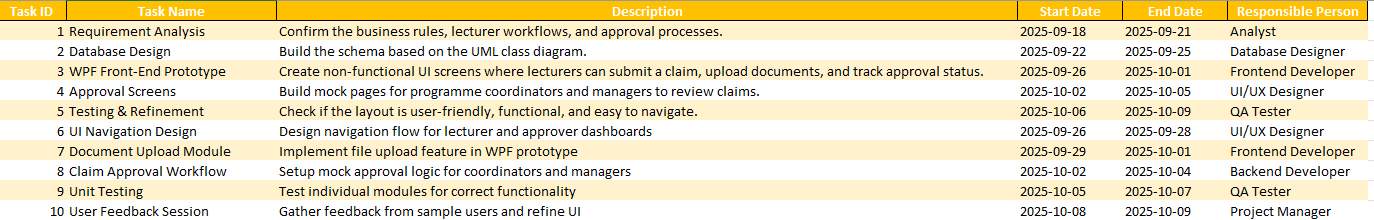
**4. GUI / UI Design**

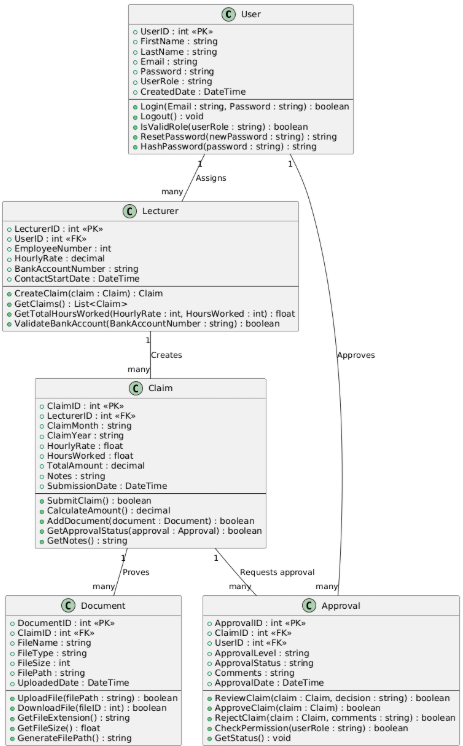
The WPF interface will be designed to be straight forward and intuitive. When you log in, the system will display role-specific options. Lecturers will see different options to create claims, attach documents and check claim status. Approvers will see a dashboard showing pending claims with the option to approve or reject.

Buttons and navigation will be kept small to lower confusion. A lecturer will be able to click “Submit Claim” to open a structured form and a document upload area will be included. Approvers will be able to view the claim and details alongside supporting documents before deciding.

The focus of the GUI at this stage is mainly on layout and navigation flow. The actual functionality, such as saving data or validating input, will be implemented in later.

remade the project plan to look better and more understandable and added more information

‌



Created with plantUML

Plantuml.com. (2025). *PlantUML Web Server*. [online] Available at: <https://www.plantuml.com/plantuml/uml/SyfFKj2rKt3CoKnELR1Io4ZDoSa700002#google_vignette> [Accessed 16 Sep. 2025].

Stack Overflow (2009) *Open a WPF Window from another window* Available at: <https://stackoverflow.com/questions/553817/open-a-wpf-window-from-another-window> (Accessed: 17 September 2025).

Stack Overflow (2012) *How to handle MouseDoubleClick event in WPF ListView?* Available at: <https://stackoverflow.com/questions/10901541/how-to-handle-mousedoubleclick-event-in-wpf-listview> (Accessed: 17 September 2025).

Sommerville, I. (2015) *Software Engineering*. 10th edn. Harlow: Pearson Education. Available at: <https://www.pearson.com/en-gb/subject-catalog/p/software-engineering/P200000003003/9781292096131> (Accessed: 17 September 2025).

Stack Overflow (2015) *How to center WPF window on screen?* Available at: <https://stackoverflow.com/questions/385263/how-to-center-wpf-window-on-screen> (Accessed: 17 September 2025).

Stack Overflow (2018) *How to style WPF TabControl?* Available at: <https://stackoverflow.com/questions/12117916/how-to-style-wpf-tabcontrol> (Accessed: 17 September 2025).

PlantUML (2025) *PlantUML Official Documentation*. Available at: [https://plantuml.com/](https://plantuml.com/?utm_source=chatgpt.com) (Accessed: 17 September 2025).

PlantUML (2025) *PlantUML Language Reference Guide (PDF)*. Available at: [https://pdf.plantuml.net/PlantUML\_Language\_Reference\_Guide\_en.pdf](https://pdf.plantuml.net/PlantUML_Language_Reference_Guide_en.pdf?utm_source=chatgpt.com) (Accessed: 14 September 2025).

Crashedmind (2025) *The Hitchhiker’s Guide to PlantUML*. Available at: [https://crashedmind.github.io/PlantUMLHitchhikersGuide/](https://crashedmind.github.io/PlantUMLHitchhikersGuide/?utm_source=chatgpt.com) (Accessed: 14 September 2025).

Real World PlantUML (2025) *PlantUML Diagram Examples*. Available at: [https://real-world-plantuml.com/](https://real-world-plantuml.com/?utm_source=chatgpt.com) (Accessed: 17 September 2025).

Ambler, S.W. (2005) *The Elements of UML 2.0 Style*. Cambridge: Cambridge University Press. Available at: <https://www.cambridge.org/core/books/elements-of-uml-20-style/> (Accessed: 16 September 2025).

Sharp, J. (2015) *Microsoft Visual C# Step by Step*. 9th edn. Microsoft Press. Available at: <https://www.microsoftpressstore.com/store/microsoft-visual-c-sharp-step-by-step-9780735698239> (Accessed: 16 September 2025).