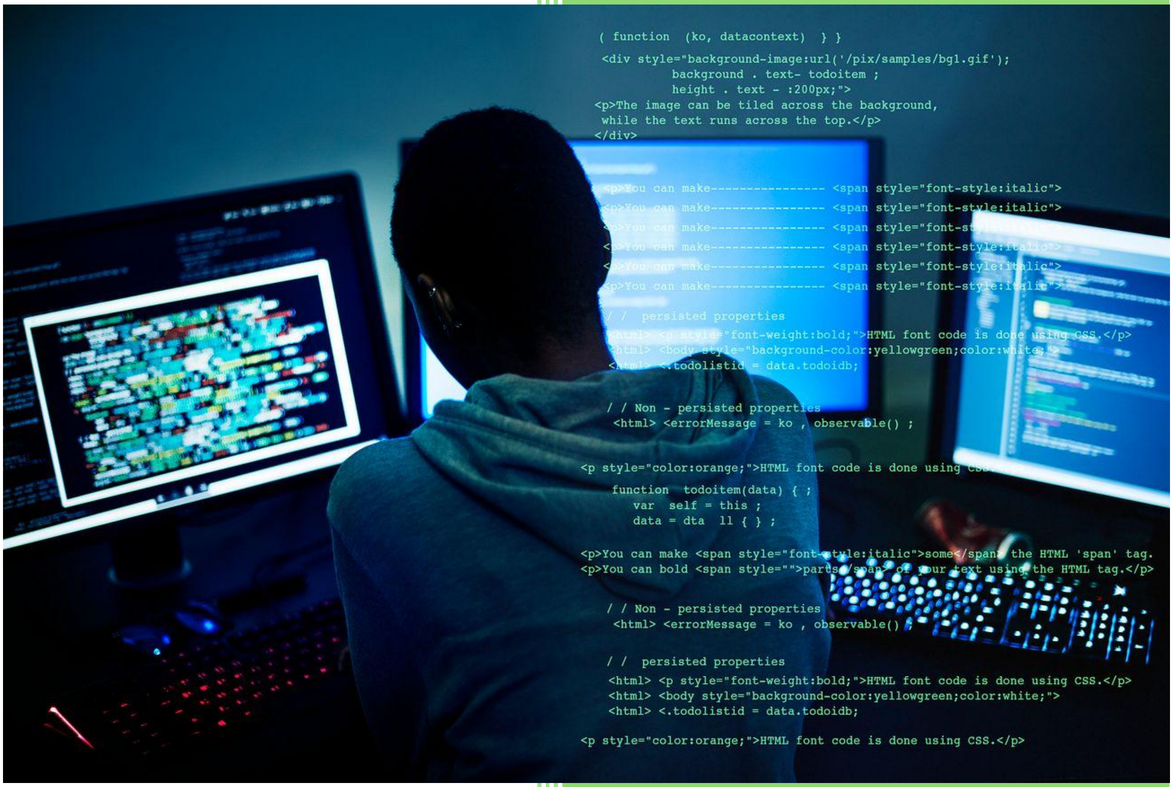
2025

DIS2 Prog6212 Part 1 GR01



Lesego Ayanda Mathe

St10440650

Table of Contents

Project Planning

1. Project Overview

In this overview project, I’m going to talk and plan about monthly system claim and this will cover planning, designing and implementation for the prototype of the system, which will help lecturers to submit and track their claims, and also assist programme coordinators and program managers to approve or decline claims.

2. User and Roles
    - Lecturers will claim submissions and monitor the status.
    - Programme Coordinator will oversee the report (whether to approve or reject).
    - Academic Management will receive claims from programme Coordinator and finalize the claims (whether to approve or reject).
3. Objectives
    - Easy access
    - User-friendly seamless platform
    - Data security and Data integration
    - Ensure accuracy and transparency

System Workflow

1. Lecturer registration and login
    - Lecturers register and logs into the system.
    - Submit claims → track claims
2. Claims will be marked as pending with submission time and date
3. Coordinator reviews
    - If approved, the claim status → Approved
    - If rejects, or requests changes the claim status → rejected
4. Academic manager review the claim status from the Coordinator
    - If finalized the claim status →Finalized
    - If rejected the claim status → RejectedByAM
5. Lecturer receives the updates
    - Lecturer gets notified of the decision
    - If rejected, lecturer → edit the claims and resubmit
    - If approved, the claim →moves to the finance or admin team (if applicable).
6. Track all claims with time and date and user History

Information Collection

1. Pages
    - Registration page
    - Login page
    - Submit claim page by uploading the docs.
    - Track the claim page
    - Coordinator review page
    - Academic Manager finalization page
    - Lecturer report page
2. Functional fields
    - User Info
    - Claim details
    - Documents
    - Status along with date and time

Architecture and Design Choice

1. Architecture (MVC)
    - Models (Data(users))
    - Views (User interface, (Html,CSS,JS))
    - Controllers (Business logic and claim workflow)
2. Frond-End
    - Html, CSS, JavaScript for responsiveness and accessibility
3. Back-end
    - C# for logic and validations
    - MySQL for relational database
4. Design Considerations
    - Separation of concerns (MVC)
    - Cross-device compatibility (desktop + mobile)
    - User-friendly forms and tables
    - Status color-coding (Pending, Approved, Rejected, Finalized)

Database Design

1. Users Table
    - user_id, name, email, role, password_hash
2. Courses Table
    - course_id, course_name, programme_id
3. Programmes Table

- programme_id, programme_name, coordinator_id

4. Claims Table
   - Claims id, lecturer_id, module, sessions, hours, rate, total_amount, status, submitted_at, submitted_on, approved_at, approved_on, finalized_at, finalized_on,document_paths
5. Notifications Table
   - notification_id, user_id, message, sent_date, read_status and sent_time

Relationships

One user can submit many claims

Each claim is linked to one course

Each course belongs to one programme

Each programme has one user

Each claim can have multiple documents

Each user can receive multiple notifications

Security & Validation

1. Password Security
   - Store salted hash instead of plain password
2. Authentication & Authorization
   - Lecturers will create/read own claims
   - Coordinators will read pending claims, approve/reject
   - AM will finalize coordinator-approved claims
3. Input Validation
   - Unique, valid email
   - Numeric validation for sessions, hours, rates
   - File type and size restrictions for documents (PDF, PNG, JPG)
4. Auditing
   - Track who changed status and when and what time

Detailed Design & Workflow Implementation

1. Model
   - Users Model → store user info

- Claims Model → store claim info & workflow

2. Claim Submission
   - Lecturer fills form → system calculates total_amount automatically
   - Attach documents → submit → status = Pending
3. Claim Approval
   - Coordinator will review and updates status
   - AM reviews and updates status
4. Claim Tracking:
   - Lecturer can see all their submitted claims with status
   - Optional reporting by month, module, or lecturer

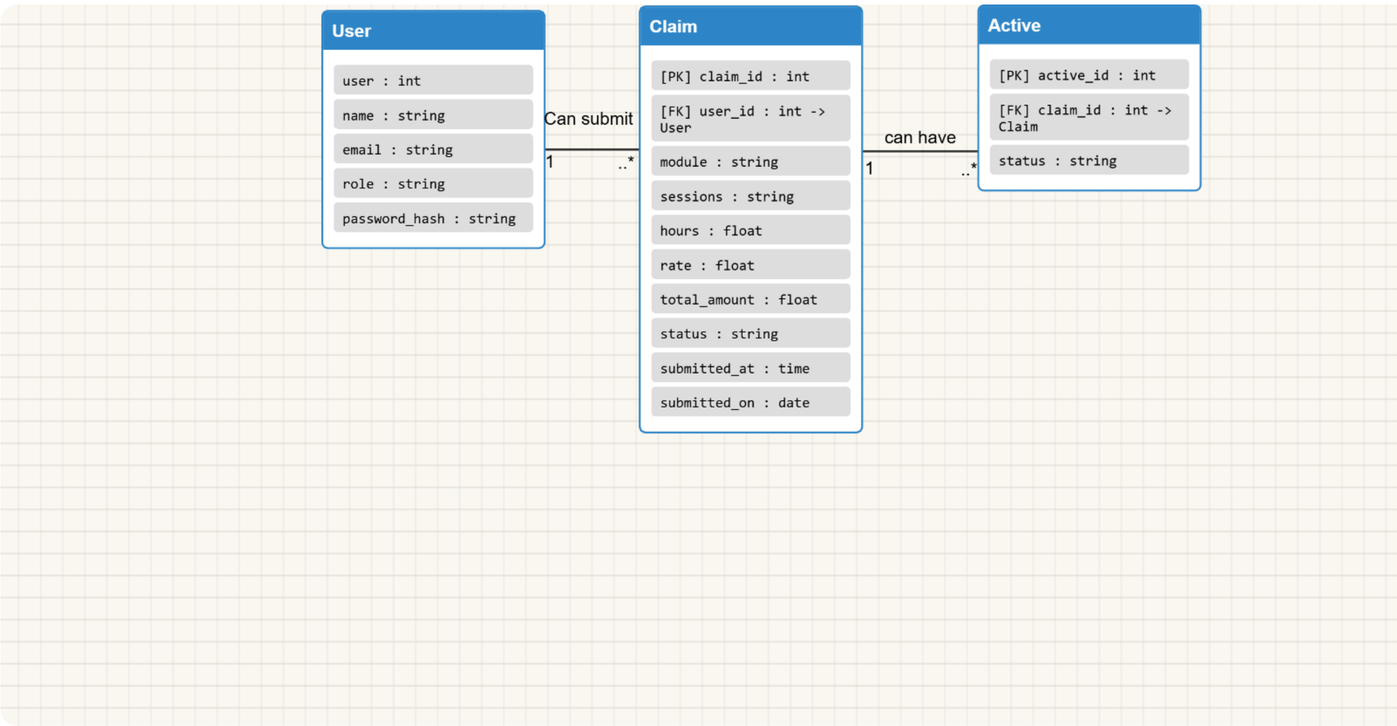
UI / UX Design Choices

- The design of the system is guided by principles of usability, accessibility, and consistency to ensure that users can interact with it easily. The layout is organized with a sidebar for navigation and a main content area for displaying forms and tables. The sidebar provides direct access to important features such as the Dashboard, Submit Claim, and Track Claim, which reduces the number of steps users need to perform. Separating navigation from the content area also creates a clear structure, making the interface easy to follow.
- The choice of colours supports both clarity and professionalism. A white background has been selected to create a clean look and improve readability, while dark blue is used as the main accent colour for buttons, headers, and highlights. This combination gives the system a formal and trustworthy appearance, while also helping users to focus on important elements. In addition, colour is used to communicate system feedback through status badges, ensuring that information is not only readable but also visually clear.
- Status indicators are an important part of the design. They are represented by colour-coded badges that show the stage of each claim. Pending claims are displayed in orange to signal that they are still in progress, approved claims are in green to show success, and rejected claims are in red to draw attention to issues that require action. These colours follow widely recognized meanings, which helps users quickly understand claim statuses without reading long text descriptions.
- The system also follows responsive design practices to make it accessible across different devices. On smaller screens, such as mobile phones, the sidebar will collapse into a compact menu to maximize space. Forms will adjust to a vertical format so that users can complete them without scrolling side to side, and tables will allow horizontal scrolling when necessary. These adjustments improve usability and ensure that the system is functional on

desktops, tablets, and mobile devices. In addition, accessibility principles such as clear fonts, high contrast, and simple navigation are considered to support all users, including those with visual or physical limitations.

UML Class diagram

1.UML DIAGRAM



Project plan

1.Tasks		**Days**	**Dates**
Phase 1: Planning & Requirements	T1. Define requirements (users, roles, workflow, pages, database design). T2. Create system architecture (MVC choice, front-end/back-end structure). T3. Prepare UI/UX wireframes (sidebar	6	26 July 2025 to 01 Aug 2025

	layout, forms, status indicators).		
Phase 2: Database & Backend Setup	T4. Set up database schema (Users, Claims, Programmes, Notifications). T5. Implement Models in C# (User, Claim, Notification models). T6. Implement authentication (registration/login with hashed passwords).	6	02 Aug 2025 to 7Aug 2025
Phase 3: Core Features Development	T7. Lecturer functions (submit claim, upload docs, track claim). T8. Coordinator functions (review pending claims, approve/reject). T9. Academic Manager functions (finalize/reject claims). T10. Notifications system (inform users about status updates).	6	08 Aug 2025 to 14 Aug 2025
Phase 4: UI/UX Front-End Development	T11. Build registration & login pages. T12. Build claim submission form (with validations, auto-calculation). T13. Build claim tracking & status page (with color-coded badges). T14. Build coordinator & manager dashboards (approve/reject workflows)	6	15 Aug 2025 to 21 Aug 2025
Phase 5: Testing & Refinement	T15. Unit testing (form validation, login, claim workflow). T16. Integration testing (end-to-end workflow from lecturer → AM).	6	22 Aug 2025 to 28 Aug 2025

	T17. Bug fixes & adjustments.		
Phase 6: Prototype Delivery	T18. Deploy prototype on local server/test environment. T19. Prepare documentation (user guide + system workflow). T20. Final presentation/demo of prototype.	6	29 Aug 2025 to 04 Sept 2025
2. Dependencies			
T2 depends on T1 (can't design architecture before requirements).	T3 depends on T1 (UI wireframes need requirements).		
T4 depends on T2 (DB schema follows architecture).	T5–T6 depend on T4 (backend needs database).		
T7–T10 depend on T5–T6 (features need models + authentication).			
	T11–T14 depend on T3 & T7–T10 (UI built once workflows exist).		
T15–T16 depend on T7–T14 (testing after core features).	T17 depends on T15–T16 (fixes after testing).		
T18–T20 depend on all earlier tasks.			
3.Timeline (6 Weeks Plan)			

Week 1(Tasks)	T1 (Requirements), T2 (Architecture), T3 (Wireframes)		
Week 2(tasks)	T4 (Database), T5 (Models), T6 (Authentication)		
Week 3(tasks)	T7 (Lecturer features), T8 (Coordinator features)		
Week 4(tasks)	T9 (Academic Manager), T10 (Notifications), start T11–T12 (UI pages)		
Week 5(tasks)	T13–T14 (UI dashboards + status tracking), T15 (Unit testing)		
Week 6(tasks)	T16 (Integration testing), T17 (Bug fixes), T18 (Deployment), T19 (Docs), T20 (Demo)		
4.Days	6		
5.Duration	2025-07-26 to 2025-09-05		