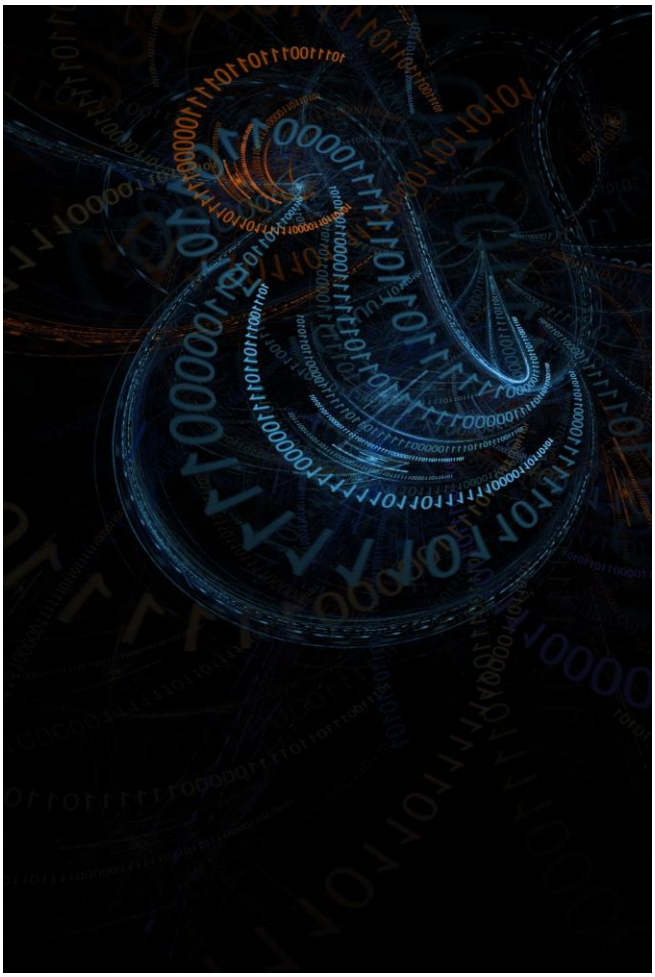


PROG2B

POE 1

Edin Jamie Shand
Diploma in Software Development (DISD2)

Content Page

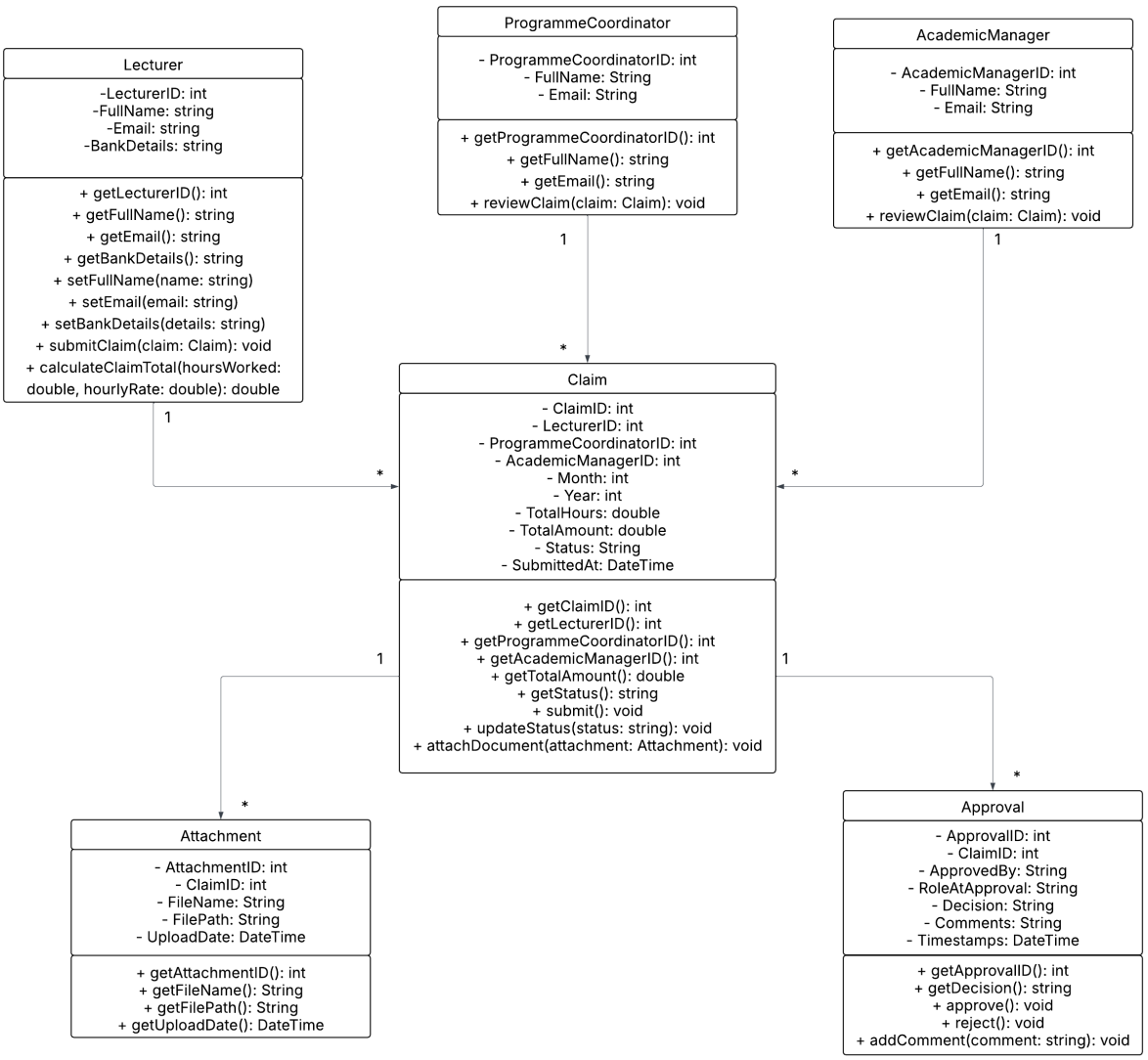
Documentation

- Design Choices:
  - Platform:
    - → I decided to use Visual Studio 2022 to create ASP.NET Core MVC as it is browser accessible and easy to deploy. WPF could also be used for a desktop version.
  - Architecture:
    - → To keep things structured and manageable, I employed a layered design (Presentation → Services → Domain → Data)
- Roles:
  - Lecturer:
    - → Submits claims, uploads documents and monitors claim status.
  - Programme Coordinator (PC):
    - → Reviews claims and can approve or return the claims.
  - Academic Manager:
    - → Gives the final approval or rejection.
- Workflow:
  - Claims go from the lecturer to the programme coordinator to the academic manager with clear status updates.
- File Handling:
  - Attachments are stored in a folder with their information saved in the database.
- Database Structure:
  - Lecturer:
    - → LecturerID, FullName, Email, BankDetails
  - Claim:
    - → ClaimID, LecturerID, ProgrammeCoordinatorID, AcademicManagerID, Month, Year, TotalHours, TotalAmount, Status, SubmittedAt
  - Programme Coordinator:
    - → ProgrammeCoordinatorID, FullName, Email
  - Academic Manager:
    - → AcademicManagerID, FullNAme, Email
  - Attachment:
    - → AttachmentID, ClaimID, FileName, FilePath, UploadDate
  - Approval:
    - → ApprovalID, ClaimID, ApprovedBy (ProgrammeCoordinator or AcademicManager), RoleAtApproval (ProgrammeCoordinator or

AcademicManager), Decision(Approved/Rejected/Pending), Comments, Timestamps

- Relationships:
  - Lecturer 1 → *Claims
  - ProgrammeCoordinator 1 → * Claims
  - AcademicManager 1 → * Claims
  - Claim 1 → * Attachments
  - Claim 1 → * Approvals
- GUI Layout:
  - Login Page:
    - → Users log in based on their role
  - Lecturer Dashboard:
    - → View and submit claims, upload documents and track status
  - Claim Submission Form:
    - → Add multiple claim lines, upload files, reviews totals
  - Programme Coordinator Dashboard:
    - → Approve or returns claims with comments
  - Academic Manager:
    - → Final approval screen and track claim statuses
- Assumptions and Constraints
  - One claim per Lecturer per Month/Year
  - Maximum attachment size: 10MB
  - Calculations are based on HourlyRate
  - Roles musts follow the workflow order

UML Class Diagram for Databases



Project Plan

- Tasks:
  1. Define the project scope and requirements.
  2. Design a UML class diagram and draft a database schema.
  3. Create wireframes and user flows for the GUI.
  4. Develop a static GUI prototype (layout only, non-functional).
  5. Compile the documentation and submit the deliverables.

- Dependencies:
  - Task 2 depends on task 1
  - Task 3 depends on task 1 and task 2
  - Task 4 depends on task 3
  - Task 5 depends on task 1, task 2, task 3 and task 4.

- Timeline:
  - 28 Jul – 3 Aug →Define the project scope and requirements.
  - 4 – 17 Aug → Design a UML class diagram and draft a database schema.
  - 18 – 24 Aug → Create wireframes and user flows for the GUI.
  - 25 Aug – 7 Sep → Develop a static GUI prototype (layout only, non-functional).
  - 8 – 9 Sep → Compile the documentation and submit the deliverables.

GUI/UI

- Main Screens:
  - Login Page:
    - → Username and password fields
    - → Login button
  - Lecturer Dashboard:
    - → Sections: My Claims, Submit New Claim, Track Status
    - → Buttons: Submit Claim, View Claim
  - Claim Submission Form:
    - → Fields: Month, Year, Claim Lines (ActivityType, Hours, LineAmount)
    - → File Upload: Drag and drop or browse
    - → Buttons: Add Line, Remove Line, Submit
  - Programme Coordinator Dashboard:
    - → List of claims for approval
    - → Actions: Approve and return with comments
  - Academic Manager Dashboard
    - → View claims and comments
    - → Buttons: approve or reject claims
    - → Track overall claim statuses
- Design Principles:
  - Clear labels and buttons
  - Easy navigation between screens
  - Status indicators for claims
  - Totals always visible for financial overview