

Programming 2B – Part 1

Name: Rahil Ramdass

Student Number: ST10434494

Module Code: PROG6212

Due Date: 17 September 2025

## Contents

1. Introduction .....	3
2. Design Choices .....	3
3. Database Structure .....	3
4. GUI Layout.....	3
5. Assumptions and Constraints .....	4
6. Conclusion .....	4
Reference List.....	5

## 1. Introduction

The purpose of this report is to explain the planning and design of a prototype for the Claims Management and Control System (CMCS). The assignment required the creation of a non-functional prototype that demonstrates key database entities, a basic graphical user interface (GUI), and supporting documentation including a UML diagram and project plan. This document reflects on the design choices, the database structure, the GUI layout, and the assumptions and constraints considered during the prototype development process.

## 2. Design Choices

The system was designed around the core entity Claim, which represents a lecturer's monthly submission of hours worked and associated remuneration. Each claim is linked to a User entity that includes the role (Lecturer, Coordinator, or Manager). A separate ClaimLine entity was introduced to capture day-by-day activities, ensuring transparency and granularity in how claims are calculated. This avoids potential disputes and simplifies auditing, aligning with best practices for data normalisation (Connolly & Begg, 2015).

The SupportingDocument entity was included so that lecturers can attach files such as contracts, proof of hours, or timesheets. Storing documents separately maintains a clean database design and prevents duplication of metadata (Elmasri & Navathe, 2016). The Approval entity records decisions by coordinators and managers. By storing approver details, dates, and comments, the model ensures accountability and an auditable decision trail, which is crucial for institutional governance (Date, 2004).

Overall, the design choices reflect a balance between simplicity for prototyping and sufficient detail to meet the functional requirements of a real-world system.

## 3. Database Structure

The database structure is built around relationships that show how information flows between users, claims, and decisions. A one-to-many relationship exists between a User and their Claims, and similarly between Claim and its ClaimLines, SupportingDocuments, and Approvals. Foreign keys enforce referential integrity and ensure that claims cannot exist without being linked to a valid lecturer (Rob & Coronel, 2019).

Key constraints were introduced:

- HoursWorked and HourlyRate must be greater than zero.
- Status must be one of {Pending, Approved, Rejected}.

This structure supports validation, improves data quality, and aligns with database design principles recommended in higher education contexts (Coronel & Morris, 2016).

## 4. GUI Layout

The prototype GUI consists of three non-functional role-specific pages:

- Lecturer Page: Displays a list of claims with their status, and disabled buttons for *New Claim* and *Upload Supporting Documents*.
- Coordinator Page: Shows pending claims as cards with lecturer details, totals, and disabled *Approve/Reject* actions.

- Manager Page: Provides an overview of all claims using tabbed navigation (Pending, Approved, Rejected), with a static list of claims under each tab.

Bootstrap was applied to ensure the interface is responsive and visually consistent. The focus was on usability and clarity rather than functionality, in line with the assignment requirements (Nielsen, 1994).

## 5. Assumptions and Constraints

Several assumptions were made during the design:

- Authentication and user management are out of scope for Part 1 and will be addressed in later phases.
- Only standard file types (PDF, JPG, PNG) are assumed for supporting documents.
- The prototype is static; all buttons and links are disabled intentionally to emphasise layout rather than functionality.
- Monetary values are displayed in South African Rand (ZAR) with two decimal places.

Constraints include limited time and the requirement to deliver a clear but non-functional demonstration, which influenced the choice to keep interactions disabled.

## 6. Conclusion

This report outlined the rationale behind the UML design, the underlying database structure, the GUI layout, and key assumptions made in the CMCS prototype. The approach ensured that the prototype not only met the minimum requirements but also demonstrated good design principles, positioning the project strongly for later functional development.

## Reference List

- Connolly, T. & Begg, C., 2015. *Database Systems: A Practical Approach to Design, Implementation, and Management*. 6th ed. Harlow: Pearson.
- Coronel, C. & Morris, S., 2016. *Database Systems: Design, Implementation, and Management*. 12th ed. Boston: Cengage Learning.
- Date, C.J., 2004. *An Introduction to Database Systems*. 8th ed. Boston: Addison-Wesley.
- Elmasri, R. & Navathe, S., 2016. *Fundamentals of Database Systems*. 7th ed. Boston: Pearson.
- Nielsen, J., 1994. *Usability Engineering*. San Francisco: Morgan Kaufmann.
- Rob, P. & Coronel, C., 2019. *Database Principles: Fundamentals of Design, Implementation, and Management*. 13th ed. Boston: Cengage Learning.