PORTFOLIO OF EVIDENCE

Programming 2B [PROG6212]

HP

VARSITY COLLEGE  Group 3 Lecturer: Matthew Prinsloo

# Table of Contents

[Introduction 2](#_Toc209011686)

[Design Choices 2](#_Toc209011687)

[UML Diagram 3](#_Toc209011688)

[Project Plan 4](#_Toc209011689)

[GUI UI Design](#_Toc209011690) 5

[UI Designs Breakdown](#_Toc209011691) 6

[Conclusion 6](#_Toc209011692)

[Reference List 6](#_Toc209011693)

[AI Usage and Disclaimer 7](#_Toc209011694)

# Introduction

The Contract Monthly Claim System (CMCS) is a pioneering prototype web application designed to revolutionize the management of monthly claims for independent contractor lecturers. Its primary purpose is to enhance operational efficiency by providing a centralized, user-friendly platform that simplifies the submission, verification, and approval processes. This system targets three key user groups: lecturers who submit claims, programme coordinators who review them, and academic managers who provide final approvals. Part 1 of the project focuses exclusively on the prototype phase. This involves developing a project plan, designing a Unified Modelling Language (UML) class diagram for the database, and creating a non-functional graphical user interface (GUI) using ASP.NET Core MVC. No functionality is implemented at this stage, aligning with the prototype's exploratory nature, setting the foundation for future development in Parts 2 and 3. According to Lonc (2025), non-functional GUI prototypes like this one enable early validation and iteration, mitigating risks by identifying usability issues before coding begins, which is crucial for cost savings in web app development.

# Design Choices

This document is meticulously structured into sections: introduction, design choices, UML diagram, project plan, GUI, conclusion, references, and AI usage disclaimer. This organization ensures a logical progression, guiding the reader from context to technical details and reflections. The overall layout features a fixed navigation bar at the top, a dynamic hero section on the home page, and card-based dashboards for lecturer and manager interfaces, fostering an intuitive and accessible user experience. According to Khanh (2025), fixed navbars with icon integration, such as those used here for home, lecturer, and manager links, reduce cognitive load by providing consistent visibility and seamless transitions, improving usability in web apps like CMCS.

The structure leverages a responsive grid system powered by Bootstrap 5, ensuring the interface adapts flawlessly across desktops, tablets, and mobile devices. This choice supports diverse user environments, enhancing accessibility. The colour scheme is a deliberate blend of professionalism and engagement: a gradient navbar transitioning from #2c3e50 (deep slate) to #34495e (charcoal grey) exudes authority and reliability, while the hero section’s gradient from #2980b9 (bright blue) to #6dd5fa (light cyan) to #ffffff (white) creates a vibrant yet clean welcome. Hover effects in #f1c40f (yellow) highlight interactivity. According to Metral (2024), blue conveys trust and calmness, ideal for professional apps to foster reliability, while yellow adds warmth and energy to encourage interaction, making gradients a modern way to enhance visual appeal without overwhelming users. These selections align with WCAG accessibility standards for contrast, ensuring inclusivity.

Key assumptions include the expectation that lecturers will input accurate claim details, deferring validation to Part 2 to maintain prototype simplicity, and a constraint limiting file uploads to .pdf, .docx, and .xlsx formats due to compatibility with common office tools. The UML diagram, crafted in draw.io, models the database structure, while the project plan, customized in Jira, uses a Gantt chart to track tasks, dependencies, and timelines, adjusted for today’s deadline.

# UML Diagram

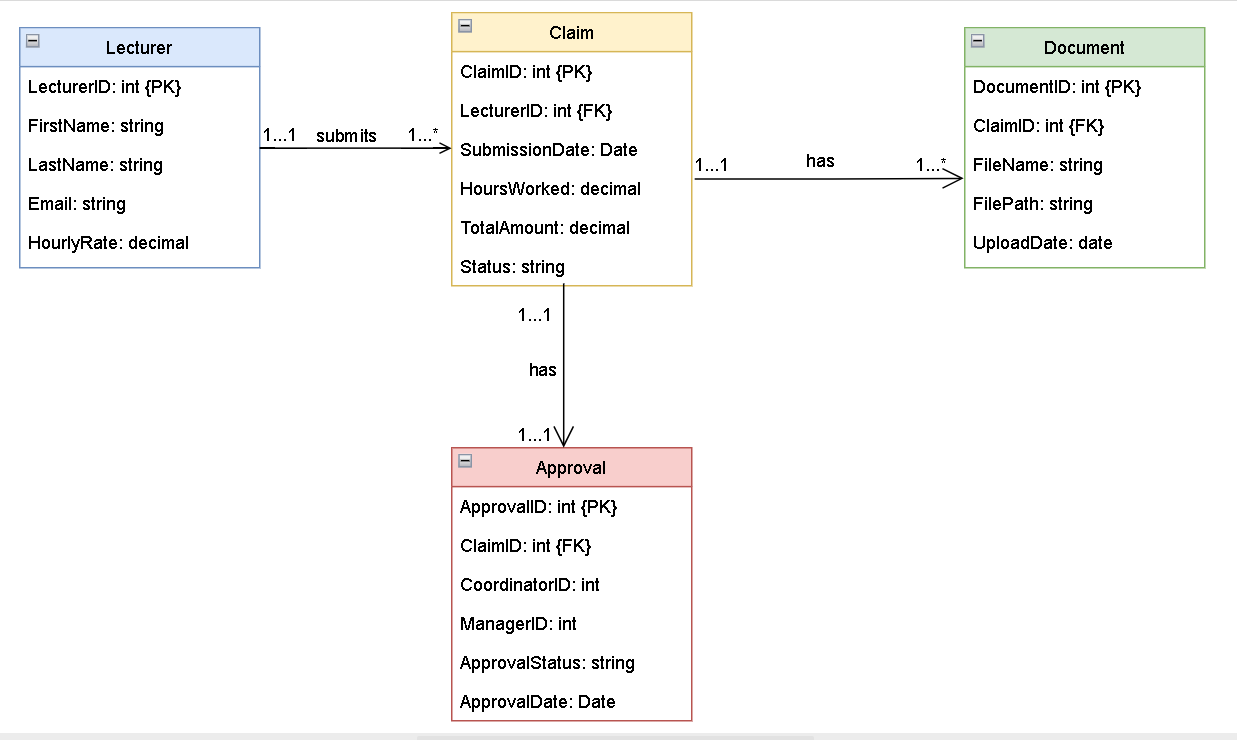


Figure : UML Diagram

The UML class diagram accurately represents the CMCS database requirements, as shown in the provided model. It includes core classes: Lecturer (attributes: lecturerId (PK), firstName, lastName, email, hourlyRate), Claim (attributes: claimId (PK), lecturerId (FK), submissionDate, hoursWorked, totalAmount, status), Document (attributes: documentId (PK), claimId (FK), fileName, filePath, uploadDate), and Approval (attributes: approvalId (PK), claimId (FK), coordinatorId, managerId, approvalStatus, approvalDate). All attributes use camelCase for consistency. Relationships feature one-to-many from Lecturer to Claim (1..1 submits 1..), one-to-many from Claim to Document (1..1 has 1..), and one-to-one from Claim to Approval (1..1 has 1..1), with solid lines and multiplicity notations (e.g., 1, 1..\*) for clarity. Symbols include PK/FK labels and rectangles for classes. According to Visual Paradigm (n.d.), this structure ensures completeness in database modeling by mapping entities to relational schemas via foreign keys, supporting claim management workflows like submission and approval tracking in prototypes. The diagram's accuracy validates system requirements without gaps.

# Project Plan

Figure : Project plan

# GUI UI Design

Figure : CMCS project Manager Dashboard Design

Figure : CMCS project Lecturer Dashboard design

Figure : CMCS project landing page design

## UI Designs Breakdown

The GUI, prototyped in Visual Studio with ASP.NET Core MVC, comprises three pages. The home page features a hero section with a gradient background, a bold headline ("Streamline Your Monthly Claims"), and dual call-to-action buttons (lecturer start, admin login), highlighting quick submissions, uploads, and tracking. According to Bootstrap (2018), the responsive grid and components like cards enable rapid prototyping, with a mobile-first approach ensuring adaptability for web apps like CMCS. The lecturer dashboard, split into a claim form (hours, rate, notes) and a right column for document upload (mock file) and status tracker (progress bar at 50%, "Pending Coordinator"), uses cards for organization. The manager dashboard offers a searchable table of claims (e.g., CMCS-001, John Doe, $500) with action buttons (approve, reject, view), enhanced by pagination. All elements are non-functional, styled with Bootstrap for a user-friendly, aesthetic layout. Lonc (2025) emphasizes that such prototypes facilitate early user testing, reducing costs by validating designs before full implementation.

# Conclusion

The CMCS prototype successfully establishes a robust framework with a well-designed GUI, UML model, and project plan. The aesthetic enhancements and structured approach lay a solid groundwork for functional implementation in subsequent parts, promising an efficient claim management system(xAI,2025).

**GitHub link:** [**https**://github.com/VCCT-PROG6212-2025-G3/ST10307204\_CMCS\_PROG6212\_POE](https://github.com/VCCT-PROG6212-2025-G3/ST10307204_CMCS_PROG6212_POE)

# Reference List

xAI. (2025) Grok (Version 1.0) [Large language model]. Available at: <https://x.ai/grok> [Accessed: 17 September 2025].

Bootstrap. (2018) Bootstrap CSS Documentation, Available at: <https://getbootstrap.com/docs/3.4/css/> (Accessed: 16 September 2025).

Khanh, L. (2025) Navigation UX: Pattern Types and Tips to Enhance User Experience, Available at: <https://userpilot.com/blog/navigation-ux/> (Accessed: 16 September 2025).

Lonc, J. (2025) What is Prototyping in Software Development (and Why It's Important), Available at: <https://sparkbusinessworks.com/blog/prototyping-in-software-development/> (Accessed: 16 September 2025).

Metral, K. (2024) The Power of Color Psychology in Web Design, Available at: <https://cosmicostudios.medium.com/the-power-of-color-psychology-in-web-design-b798e956797a> (Accessed: 15 September 2025).

Visual Paradigm. (n.d.) UML Practical Guide - All you need to know about UML modeling, Available at: <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/uml-practical-guide/> (Accessed: 15 September 2025).

# 

# AI Usage and Disclaimer

* **Section(s) within the assessment in which generative AI was used**: (GUI UI Design)
* **Name of AI tool(s) used**: ChatGPT
* **Purpose/intention behind use**: Enhancing the design of the GUI UI by exploring and integrating cutting-edge tools such as Bootstrap 5 and Font Awesome to further expand my knowledge of modern web design techniques and best practices. This approach aimed to refine my skills in responsive design and aesthetic optimization, aligning with the project’s technical requirements while providing a valuable learning experience.
* **Date(s) in which generative AI was used**: September 16, 2025
* **A link to the actual generative AI chat, or screenshots of the chat**: [ChatGPT](https://chatgpt.com/c/68c9e0ae-4fb0-8333-8272-5d1f42751e29)

**Reference for ChatGPT**

OpenAI. (2024) *ChatGPT* (Version 3.5) [Large language model]. Available at: <https://chat.openai.com/> [Accessed: 16 September 2025].