

Contract Monthly Claim System (CMCS) Prototype Development: Project Plan

Project Overview

The CMCS prototype development project focuses on building a web-based application to streamline the monthly claims submission and approval process for independent contractor lecturers. This project plan outlines the tasks, timelines, dependencies, and milestones necessary to complete the prototype development.

Project Objectives

- Design a non-functional prototype for the CMCS using .NET Core MVC or Windows Presentation Forms (WPF).
- Develop a UML class diagram for the database that outlines relationships between key entities.
- Create a user-friendly GUI for lecturers to submit claims, and approvers to process them.
- Provide consistent project documentation with regular version control using GitHub.

Project Phases and Tasks

Requirements Analysis(1 week)

- Identify user requirements (2 days)
 - Conduct research to gather requirements.
- Define system functionality (1 day)
 - Translate user requirements into specific system features and functions.
- Review requirements with stakeholders (2 days)
 - Validate requirements and refine based on feedback from lecture.

Database Design (1 week)

- Define data entities and relationships (2 days)
 - Outline the necessary database entities like Lecturer, Claim, Approver, and Document.
- Create UML class diagram for the database (3 days)
 - Design the database schema using UML, specifying entity relationships and attributes.
- Review design with the team (1 day)
 - Validate the UML diagram with stakeholders and the development team.

GUI Design (1.5 weeks)

- Define user flows and wireframes (3 days)
 - Design basic wireframes and navigation flow for key user interfaces, such as the claim submission form and approval dashboard.
- Create prototype layouts in .NET Core MVC or WPF (4 days)
 - Design front-end components using .NET Core MVC or WPF with a focus on layout, navigation, and user experience.
- Review with lecturer (1 day)
 - Present the GUI prototype to lecturer for feedback and make necessary adjustments.

Prototype Development (2 weeks)

- Implement initial page layouts and views (5 days)
 - Develop front-end pages such as the dashboard, submission form, and approval interface, following the design.
- Integrate database context (5 days)
 - Create and configure the database context in .NET Core using Entity Framework.
- Add basic interactivity (4 days)
 - Implement simple navigation and interaction between the different prototype pages (no functional backend).

Testing & Refinement (1 week)

- Conduct internal testing for GUI design (3 days)
 - Test the prototype's UI across different browsers and screen sizes.
- Refine based on feedback (2 days)
 - Implement improvements based on testing feedback and make refinements to the layout and design.

Documentation (3 days)

- Write design and implementation documentation (2 days)
 - Draft documentation detailing design choices, UML diagrams, GUI design, and system assumptions.
- Review and finalize documentation (1 day)
 - Ensure documentation is clear, concise, and aligns with submission guidelines.

Submission & Version Control (Ongoing)

- Commit and push code regularly to GitHub (at least 5 times throughout development)
 - Use GitHub for version control, ensuring all code and documentation is consistently updated with clear commit messages.
- Final push before submission (1 day)
 - Push the final version of the prototype and documentation to GitHub before the project deadline.

Dependencies

- Database Design: depends on the completion of Requirements Analysis.
- GUI Design: depends on the completion of Database Design.
- Prototype Development: depends on GUI Design.
- Testing and Refinement: follows Prototype Development.
- Documentation: can begin during GUI Design and continue in parallel.

Timeline & Gantt Chart

The project spans over **6 weeks (about 1 and a half months)** with tasks overlapping to optimize productivity. For a visual representation of the timeline and task dependencies, refer to the Gantt chart.

Task	Start Date	End Date	Duration	Dependencies
Requirements Analysis	2023-10-01	2023-10-08	1 week	-
Database Design	2023-10-09	2023-10-16	1 week	Requirements Analysis
GUI Design	2023-10-17	2023-10-31	1.5 weeks	Database Design
Prototype Development	2023-11-01	2023-11-14	2 weeks	GUI Design
Testing & Refinement	2023-11-15	2023-11-21	1 week	Prototype Development
Documentation	2023-10-17	2023-11-22	3 weeks	GUI Design, Prototype Development
Submission & Version Control	Ongoing	Ongoing	Ongoing	Throughout project

Risks and Assumptions

- Assumptions:

- All users (Lecturers and Approvers) have reliable internet access.
- The system will handle a moderate number of concurrent users (up to 1000).
- Document uploads will be limited to common file formats (PDF, DOC, JPEG).
- The claims approval process follows a defined workflow.
- Risks:
 - Scope Creep: Additional feature requests may extend the development timeline.
 - Time Constraints: Delays in requirement gathering or design phases may affect subsequent phases.
 - Technical Challenges: Integration of the database context or front-end components may encounter unexpected issues.

Conclusion

This project plan outlines a clear roadmap for developing the CMCS prototype. By adhering to this plan, we ensure that the development process remains on track, with timely feedback loops, robust documentation, and consistent progress toward delivering a high-quality prototype.