University of Washington Tacoma
School of Engineering and Technology
24-25 Graduate Certificate in Software Development Engineering program (GC-SDE)

TCSS 506 A Sp 25: Practical Full Stack Development

Software Requirements and Specifications(SRS)

for

TaskFlow: A Project Management Platform

Prepared by

Preston Harms

Richard Le

Jannine G. D. MacGormain

May 03, 2025

Table of Contents

Revision History

Name	Date	Reason For Changes	Version
Team TaskFlow	05/03/2025	Initial Draft	Version 1

1. Introduction

1.1 Purpose

This Software Requirements and Specifications (SRS) outlines the requirements for the **TaskFlow** web application, developed as part of the **TCSS 506** Web Development Course Group Project. **TaskFlow** will aim to be a comprehensive project management platform designed for student teams to coordinate group projects, track assignments and improve collaboration. It is intended for project group members responsible for implementing and verifying the functionality of the web application. The implementation will adhere to object-oriented programming principles in Python. The project aims to collaboratively plan, design, and deploy a web application while applying the MVC (Model-View-Controller) design pattern and other design methodologies to enhance the project's modularity. The application will utilize MongoDB for database management, React Typescript for frontend development, the Flask Python web application framework for development, and will be deployed on Amazon Web Services (AWS). It will incorporate external APIs for data integration, and the application will be containerized using Docker for efficient deployment. Version control will be implemented using Git and GitHub for collaborative coding and task management, while project planning will be organized through YouTrack Project Management. This comprehensive approach is essential for the successful completion of the **TCSS 506** course.

1.2 Document Conventions

This document adheres to standard Software Requirements and Specifications (SRS) conventions, including:

- Headings: Used to organize content clearly and hierarchically.
- Bullet Points: Employed for lists to enhance readability.
- Tables: Used to present information in an organized manner.
- Figures: Illustrate concepts or designs visually.
- Terminology: Specific terminology is consistently applied for clarity and precision.

Additionally, priorities for requirements are indicated in the functional requirements sections to help stakeholders understand the importance and urgency of each requirement.

1.3 Intended Audience and Reading Suggestions

This document is intended for developers, project managers, testers, and documentation writers. It is suggested that readers begin with the Introduction, followed by the Overall Description, and then delve into the System Features, External Interface Requirements, Other Nonfunctional Requirements, and Other Requirements as presented in Figure 1.

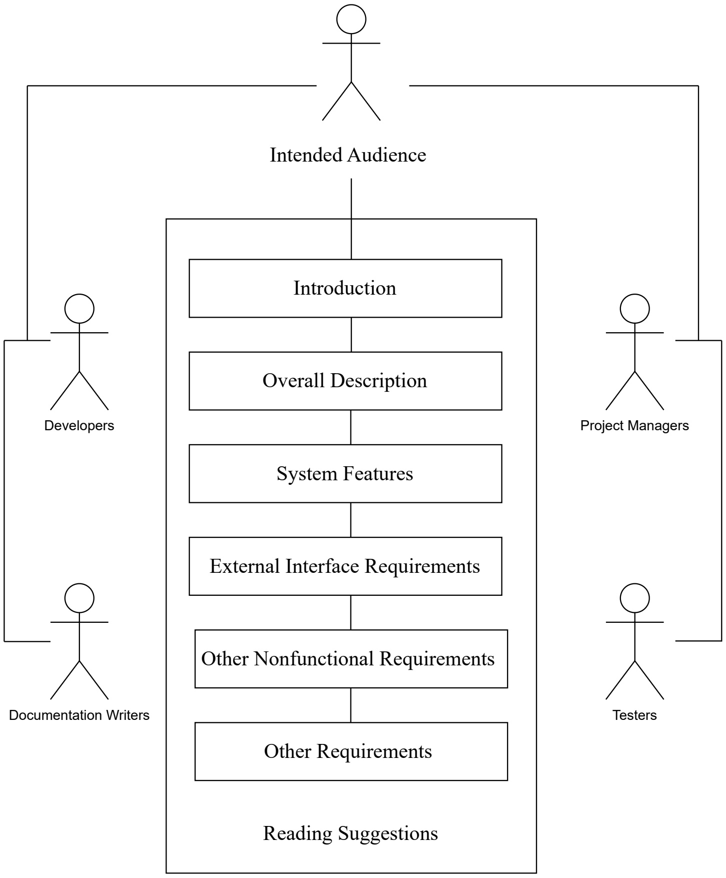


Figure 1 - Intended Audience and Reading Suggestions

This sequence will help readers gain a comprehensive understanding of the project and its requirements.

1.4 Project Scope

The **TaskFlow** web application project aims to define and expand upon the technical requirements and planning documentation necessary for its development.

The project will focus on the following key features, as illustrated in Table 1:

Login/Registration Page	Users should be able to create an account, log in, and log out of the application. This will be done via Google OAuth API.
Database Integration	The application will store and retrieve data relevant to its functionality, such as user profiles, project details, tasks, and other relevant information. It will employ MongoDB for data management.
External API Integration	The application must fetch and utilize data from external APIs to enhance user experience, such as accessing GitHub's API for repository information.
Web Application Structure	The application will feature a clear landing page that introduces users to its functionalities, along with at least one additional content page that provides meaningful interaction or data to users.
Deployment	The application must be deployed on an AWS EC2 instance, utilizing Docker containers for efficient and scalable deployment.

Table 1.1 – Key Features (Continued)

Project Planning and Management	This includes: - The analysis and design phase of the chosen project. - Clearly defined roles and responsibilities for each group member. - A project timeline that outlines key milestones and deadlines. - An overview of the technologies chosen for the project and the rationale behind their selection.

Table 1.2 – Key Features

1.5 References

- Homework-3 Outline of project prepared by Professor Ling-Hong Hung

- TCSS 506 Modules prepared by Professor Ling-Hong Hung

- AWS Documentation: https://docs.aws.amazon.com

- Dockerdocs Manual: https://docs.docker.com/manuals/

- Ubuntu Linux Manual Pages

- Group discussions via Discord and collaborative task progress tracking through the GitHub repository and YouTrack project management.

- [SRS Templates] TCSS 504 Software Requirements and Specifications (SRS) Assignment prepared by Professor Tom Capaul.

- Git and GitHub Documentation.

- YouTrack Documentation.

- Getting Started with Python by Fabrizio Romano et al.

- https://app.diagrams.net

2. Overall Description

2.1 Product Perspective

The TaskFlow will be a standalone web application developed as a course project, featuring an external API for student project tracking activities and implemented using object-oriented programming principles.

2.2 Product Features

The product will include the following key features:

- Login/Registration Page
- Database Integration
- External API Integration
- Web Application Structure
- Deployment
- Project Planning and Management

2.3 User Classes and Characteristics

Regular Users: Individuals who utilize the TaskFlow application to create, manage, and participate in collaborative projects. They can create projects, assign and track tasks, set deadlines, and monitor progress. Users will have access to GitHub integration features allowing them to link repositories to projects, track commits, and synchronize issues with tasks.

Project Managers: Users with elevated permissions within specific projects who can invite team members, assign roles, create project timelines, and generate progress reports. They oversee task distribution and monitor completion status to ensure projects stay on schedule.

System Administrators: Users tasked with managing the TaskFlow application's data and configurations. Their responsibilities include overseeing user accounts, maintaining the integrity of the application's database, monitoring system performance, and ensuring proper functionality of all features including authentication and API integrations.

2.4 Operating Environment

OE-1: The software will run on Windows, macOS, and Linux operating systems with a minimum requirement of 4 GB RAM.

OE-2: The software will require a modern CPU capable of supporting the application’s processing needs.

OE-3: A graphics card capable of rendering basic 2D graphics is necessary for optimal user interface display.

OE-4: The software will require an active internet connection to load the web application and access external APIs and resources.

2.5 Design and Implementation Constraints

CO-1: Compliance with object-oriented design principles to ensure modularity, reusability, and maintainability of the codebase.

CO-2: The application will utilize the MVC (Model-View-Controller) design pattern to separate concerns, facilitating better organization of code and enhancing scalability.

CO-3: Version control will be implemented using Git and GitHub, enabling collaborative coding, tracking changes, and managing different versions of the application.

CO-4: The application will be developed using the Flask web framework, which supports rapid development and allows for easy integration of various components.

CO-5: External APIs will be integrated to enhance the application's functionality, enabling features such as real-time data retrieval (e.g., GitHub's API for repository information and other related data) to provide users with valuable insights.

CO-6: The application must be deployed on an AWS EC2 instance, utilizing Docker containers for efficient and scalable deployment.

2.6 User Documentation

UD-1: Comprehensive user documentation will be provided, including installation instructions, app tutorials, and troubleshooting guides.

2.7 Assumptions and Dependencies

AS-1: Users have basic knowledge of computers and internet operations.

DE-1: The project relies on third-party libraries for graphics.

3. System Features

3.1 TaskFlow Project Management Features

3.1.1 Description

The app will track project progress, task completion, and team collaboration, allowing users to create and manage projects, assign tasks to team members, and monitor deadlines through an intuitive interface.

3.1.2 TaskFlow Dashboard

- **Personalize User Account**: Users can customize their profile with name, avatar, and preferences.
- **Project Management**: Users can create projects, define milestones, and establish timelines.
- **Task Tracking**: Users can create, assign, and monitor tasks with status updates, priority levels, and deadlines.

3.1.3 Web Application Initialization

- **Landing Page**: A structured landing page that introduces the **TaskFlow** application with clear value proposition and features overview.
- **Content Pages**: Multiple interactive pages including dashboard, project view, task management, and team collaboration spaces.

3.1.4 User Interface Components

- **Statistics Display**: Visual representations of project progress, task completion rates, and team contributions.
- **Login/Registration**: Authentication via Google OAuth and traditional email/password options.
- **External API Integration**: GitHub repository connection for tracking code changes, issue synchronization, and commit history.

3.1.5 Stimulus/Response Sequences

Users will interact through text input and button clicks, with the system responding by updating the UI and application state.

3.1.6 Functional Requirements

REQ-1: Users shall navigate between application pages.

REQ-2: The UI shall reflect users' current status and activities.

3.2 Create, View, Modify, and Delete

This feature allows administrators to manage available pages and dashboards, including adding or modifying content.

4. External Interface Requirements

4.1 User Interfaces

The app will feature a user-friendly interface enabling navigation through menus and application screens:

UI-1: Prompt for the user's preferred name at startup.

UI-2: Display of structured landing page that introduces the TaskFlow application with clear value proposition, features overview and multiple interactive pages.

UI-3: Functionality for users to create projects, define milestones, and establish timelines.

4.2 Hardware Interfaces

The application will utilize standard hardware interfaces:

HI-1: Monitor (Laptop or desktop)

HI-2: Keyboard

HI-3: Mouse

4.3 Software Interfaces

SI-1: Python 3.x or higher (preferably the latest stable version) is required.

SI-2: MongoDB for data management is required.

SI-3: Uses Flask web application framework.

4.4 Communications Interfaces

CI-1: The web application will not require external communications in its initial version. However, it will support the integration of multiple local pages to enhance content delivery in future versions, allowing for a more comprehensive user experience.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

PE-1: The app should load within 5 seconds and maintain a minimum frame rate of 30 FPS during interactions.

5.2 Safety Requirements

User data will be securely stored, and the application will be designed to prevent unexpected crashes.

5.3 Security Requirements

SE-1: User data will be encrypted, adhering to standard security practices to prevent unauthorized access.

5.4 Software Quality Attributes

The app will focus on usability, reliability, and maintainability to ensure a smooth user experience.

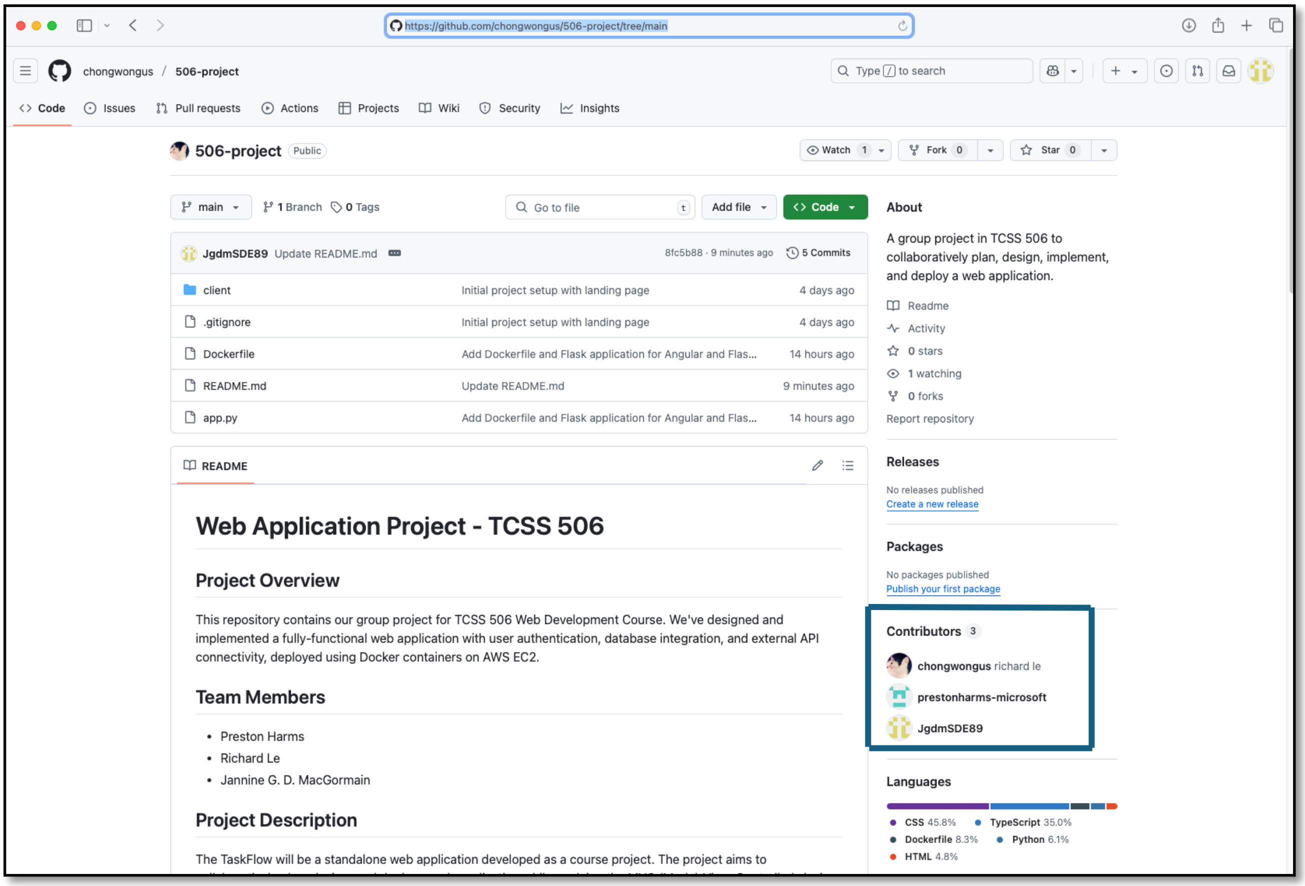
6. Other Requirements

No additional requirements have been identified at this time.

7. Roles and Responsibilities

Student Name	**Main Roles**	**Other Responsibilities**
Preston Harms	AWS Project Manager,…	…
Richard Le	Git and GitHub Version Control Project Manager,..	…
Jannine G. D. MacGormain	YouTrack Project Manager, Documentation Writer,..	…

Table 2 – Roles and Responsibilities



JgdmSDE89 Update README.md 8fc5b88 · 9 minutes ago 5 Commits

client	Initial project setup with landing page	4 days ago
.gitignore	Initial project setup with landing page	4 days ago
Dockerfile	Add Dockerfile and Flask application for Angular and Flas…	14 hours ago
README.md	Update README.md	9 minutes ago
app.py	Add Dockerfile and Flask application for Angular and Flas…	14 hours ago

Web Application Project - TCSS 506

Project Overview

This repository contains our group project for TCSS 506 Web Development Course. We've designed and implemented a fully-functional web application with user authentication, database integration, and external API connectivity, deployed using Docker containers on AWS EC2.

Team Members

- Preston Harms
- Richard Le
- Jannine G. D. MacGormain

Project Description

The TaskFlow will be a standalone web application developed as a course project. The project aims to

About

A group project in TCSS 506 to collaboratively plan, design, implement, and deploy a web application.

Releases

No releases published
Create a new release

Packages

No packages published
Publish your first package

Contributors 3

chongwongus richard le
prestonharms-microsoft
JgdmSDE89

Languages

- CSS 45.8%
- TypeScript 35.0%
- Dockerfile 8.3%
- Python 6.1%
- HTML 4.8%

Figure 2 – Roles and Responsibilities

8. Project Approval Meeting

Project Chosen Topic	**Date**	**Approved by**
TaskFlow		Professor Ling-Hong Hung
		Professor Charlie LeWarne

Table 3 – Project Approval Meeting

Appendix A: Glossary

API	Application Programming Interface
App Administrators	Users responsible for managing app data and configurations.
Developers	Members of Team TaskFlow who are responsible for designing, coding, and implementing software web applications.
Documentation Writers	Developers responsible for creating and maintaining documentation related to software products, including user manuals, technical guides, and other materials that help the intended audience understand the features and functionalities of the software.
MVC	Model-View-Controller design pattern.
Project Managers	Professionals (often referred to as Professors) who oversee projects from inception to completion, ensuring that goals are met within the desired timelines.
Stakeholders	The developers, project managers, testers, and documentation writers who oversee the development of the project

Table 4.1 – Appendix A: Glossary (Continued)

Testers	Quality assurance professionals and developers who evaluate software applications to identify defects and ensure that products meet specified requirements.
Users	The app users and app administrators of the software web application.

Table 4.2 – Appendix A: Glossary

Appendix B: Analysis Models

This section will include UML diagrams representing the class hierarchy, relationships, and interactions within the application.

Figure 3 – UML Class Diagram

Appendix C: Issues List

Information Needed: Stakeholders need to specify the desired performance metrics for the application to ensure alignment with expectations.

Ongoing Development:

As the project is currently in the code spike phase (proof of concepts), the following issues have been identified:

Tables: Specification of data structures for user interactions needs to be defined to facilitate effective data management.

Models: The definition of character and environment models requires finalization to establish the foundational elements of the application.

Figures: Visual representations of user interfaces, app pages, and dashboards are needed to gather stakeholder feedback and ensure usability.

Pending Decision: There is an outstanding decision regarding which additional third-party libraries will be utilized for graphics and other functionalities.

TBD: Decisions regarding the implementation of multiple page content are still pending, which will impact future development.