Group Project

Course: Web-Based Systems
Number: SENG 513
Semester: Winter 2025
Due Dates: Feb 21
Instructor: Mahmoud Alfadel
Version: 1.0.0

Objective

This project aims to provide students with a comprehensive understanding of the principles and techniques involved in web-based system development, coupled with hands-on experience in building a mid-size web-based application within a collaborative software development team. The goal is to develop the necessary skills and knowledge to contribute to a web-based application successfully.

Project Requirements

Groups can select any web-based application they wish, provided it meets the following technical requirements:

Client-side

- Implements HTML, CSS, and JavaScript technologies.
- Must be accessible through a single-page interface.
- Must be fully compatible with Chrome.
- Must serve as a GUI frontend for the server component.

Mobile Support

- Must be fully responsive on desktop and mobile devices.
- Must provide a reasonable amount of functionality on a mobile device.
- Must demonstrate your strategy of either mobile-friendly or mobile-first design.

Server-side

- Majority of business logic must be performed on the server side.
- Communication between the server and client sides must be handled securely and effectively.

Multiple users & roles

- Needs to support five simultaneous users.
- Must implement user authentication.
- Must support at least **three** user roles (e.g., admins, guest, etc.).
- Must provide different functionality based on the user role.

Persistence

- Must demonstrate persistence capabilities, including server restarts, user disconnections/logout, etc.

Deployment

- Must be deployable in a packaged form using Docker.

Sample Projects

- Some of the previous project's demos will be available on D2L.

Technologies to Consider

Your project should include at least one technology you have not used but would like to learn about. You can use almost any technology, E.g., technologies such as Angular, React, Vue.js, Express, Three.js, Tone.js, Bootstrap, RxJs, Webpack, MongoDB, MariaDB, Google Cloud, or serverless technologies such as AWS Lambdas or Firebase. Consider exploring technologies that make it easier for you to add features and demo your apps.

Proposal [60 Marks]

Your team must submit a single, well-structured PDF document with precise language to Dropbox on D2L. The document should be less than 10 pages (even 2-3 pages is ideal) and include the sections below.

5-point and 10-point rubrics (see appendix) will be used for each section listed below unless otherwise indicated.

Team Members [3]:

List all members of your team [1], their student IDs [1], and the project title [1].

The Problem [10]:

Write two paragraphs stating the overall purpose of your proposed web-based application. In your first paragraph, you should provide some background and describe a problem. In your second paragraph, you should propose how to solve this problem.

The target audience for this part is a non-technical person, such as an investor deciding whether to give you money to develop your project.

User Requirements [5]:

Describe primary and secondary user groups/roles and their needs. For example, you may have players, administrators and hosts for an online game. Create and describe a personas for each role. Limit this to 2- 3 sentences. Give examples of 2-3 typical tasks/use cases that each persona would want to perform using your application and how they would accomplish them. This should be described at a very high level.

Functional and Implementation Requirements [10]:

This section aims to give some technical details about your proposed application. Feel free to use tables/lists in this section.

Identify and prioritize features you plan to implement in your project. Start by listing all the potential features your application could have. Then, categorize them based on their importance and value to the user experience. For example, consider a feature like 'settings' in your app, which allows users to update their personal information, such as email address or password. Evaluate how critical this feature is to the app's functionality and user experience. Is it essential for the initial release, or can it be developed later?

List all technologies you plan to use and what you want to use them for. Please note that what you write in this section is not binding. You can change technologies, priorities or even features as you start working on the project. For this part, we expect you to do some preliminary investigation into what technologies might work well with your project. For example, tools like Continuous Integration/Continuous Deployment (CI/CD) platforms have been instrumental in streamlining development processes and enhancing collaboration and utilities such as Ngrok provide support for local testing (i.e., broadcasting your project to a mobile phone).

This section will help us evaluate the scope of your project. We will be evaluating the ambition and feasibility of your project here, as well as the clarity and priority of the features and implementation requirements.

Mobile strategy [5]:

Describe and justify your mobile strategy, determining whether a mobile-first or desktop-first approach best suits different user groups and functionalities. For instance, if your app is centred on student registration, adopting a mobile-first design could enhance accessibility and user experience for students, ensuring the interface is intuitive and streamlined on smaller screens. Following this, you could then expand the design responsively for desktop use. Conversely, a desktop-first strategy may be more appropriate for administrators who require extensive functionalities and a more complex interface. In this scenario, it is crucial to deliberate on which features are essential for the mobile version, how they can be adapted to fit a smaller screen, or if certain functionalities will be exclusive to the desktop version. Your strategy should be well-rounded, ensuring both sets of users have a seamless experience while acknowledging the inherent trade-offs and adjustments required when transitioning between mobile and desktop versions.

Project Proposal Slide/PDF [10]:

Create a single-page presentation with images to showcase your project. Describe its concept, unique features, and functionality. Keep it clear and concise for the best engagement. It should have a reasonable representation of your app's intended appearance. You can think of this as a slide that would be used to get potential investors interested in looking at your work. Evaluation of this will consider both the slide's appearance and the content.

Project Timeline [5]:

Create a project timeline that you intend to follow. This timeline should serve as a roadmap for your team, outlining what tasks need to be completed and their respective due dates. When constructing your timeline, consider all the components and phases of your project. Start by breaking the project into manageable tasks and then assign realistic deadlines for each. For instance, early tasks might include completing UI Mock-ups. As you progress, include more detailed tasks, such as developing the base functionality of your application, followed by more advanced features. Be sure to include everything that needs to be done for the final submission. This timeline should help you keep your team on track and manage workload effectively throughout the duration of the project.

Definition of Roles and Responsibilities [10]:

Create a list of roles and responsibilities for the project and assign team members. When assigning these roles, consider the strengths and interests of your team members to maximize efficiency and engagement.

It is crucial that at least two people share each role. This approach facilitates collaboration and ensures continuity and support in case one team member is unavailable. For example, you might have roles like 'Database Management', which include database creation, maintenance, and optimization. In this case, assign at least two members to take charge of these tasks. Given the relatively small size of your groups, team members will likely need to take on multiple roles. This is a great opportunity to develop a range of skills and gain a comprehensive understanding of different aspects of the project and work with different members of your team.

A partial example of this follows on the next page.

Role	Responsibilities	Members
Database Design	Define database structure Normalization	Entire team
Database Implementation	SQL scripts Test data population Stored procedures Coordinate/Communicate	Darryl & Darlene
GUI Mockups	Gathers ideas from the entire team Lo-fi designs Iterates based on team feedback	Cathy & Carson

A partial example of the roles and responsibilities table.

SUBMISSION [2]:

Pick one member of your group who will submit two PDF files to D2L, one for the proposal and the other for the single-page slide:

- the proposal PDF should be named seng513-proposal-group-<N>.pdf (e.g., seng513-proposal-group-7.pdf)
- the slide PDF should be named seng513-slide-group-<N>.pdf (e.g., seng513-slide-group-7.pdf)