Software Requirements Specification

for

Web-based Student Project Management Systems

Prepared by Shabb03

Student Number: ********

08/12/2022

Table of Contents

Revision History

Name	Date	Reason For Changes	Version
Shabb03	08/12/22	New SRS Draft	V0
Shabb03	20/02/23	Updated SRS Draft to comply with system	V1

1. Introduction

1.1 Purpose

This document is designed to give a whole overview of the Web-based Student Project Management Systems, the system requirements and analysis of the system. All system functions and updates will be documented here for reference as proof of building the system from start to finish.

1.2 Document Conventions

This section will be updated as new terms or conventions are seen throughout the design process and documentation of the proposed system.

Project Webpage: Web-based Student Project Management Systems.

Figures will be referenced in *italics.*

Terms will be in **bold.**

Reference to code in text will use the `Courier New` font.

1.3 Intended Audience and Reading Suggestions

The intended audience are DCU students, project supervisors and the CA326 module demonstration panel. It is best to read this document from top to bottom to understand the whole system at its best.

Section 2 provides an overview of the system for all readers of the document to understand. This includes the purpose of the product, different types of users who will use the product and the functions each type of user will have access to use, the problems that may arise during the development of the product and additional help for the users when using the product.

Section 3 describes how the product will interact with the user through the interface, software, hardware and communication procedures used in the interaction.

Section 4 lists and describes each system function in detail on what it will do and how it will be implemented into the system as well as the priority of each function concerning how it would affect the system if the feature does not work as intended.

Section 5 provides detailed list of non-functional requirements, these are related to the performance, safety, security, software quality and business-related regulations of the product for the users.

Section 6 provides detailed list of additional non-functional requirements not specified in section 5.

Appendix A is a glossary which provides a listed definition of the terms used in this document including acronyms and abbreviations.

Appendix B contains any relevant figures or models used and referenced to in this document.

Appendix C contains a numbered list of to-be-determined references that remain in the SRS so they can be tracked to closure.

1.4 Product Scope

The system to be developed is a student project management system with a web-based front end. The objective of the system is to allow students, project supervisors to organize their project, schedules, reports and deadlines all in one place.

The system allows DCU students to write a project proposal which can be saved as a draft then submit and wait for an approval, request a meeting with their supervisor and view a timetable that shows the available times for a project demonstration. Project supervisors can accept or reject a project proposal, add available time slots for project demonstration, create a project marking form as well as view status reports of the tasks the supervised students have done. The admin can view all of the registered students and professors, assign students without a project supervisor to one and create a demonstration timetable with one click

The benefits include less time consumption from sending emails, looking up deadlines and writing a report from scratch using another tool as well as reduce memory load from remembering meeting dates, saving report drafts in different folders and saving deadlines to calendars. This also reduces memory load for project supervisors to group all projects being supervised into one page and see which group they received an update from. For the admin the benefits are viewing a list of all the different types of users, assign students to a supervisor easily and create a demonstration timetable instantly

1.5 References

[https://dev.to/theme_selection/best-css-frameworks-in-2020-1jjh]
[Title: Best CSS Frameworks in 2022]
[Author: NA]
[Year: 15 October, 2022]

[https://www.techradar.com/best/browser]
[Title: The best web browsers for 2022]
[Author: Daryl Baxter, Carrie Marshal]
[Year: 01 December 2022]

[https://www.techtarget.com/searchapparchitecture/definition/user-interface-UI]
[Title: User Interface (UI)]
[Author: Fred Churchville]
[Year: NA]

[https://www.scaledagileframework.com/nonfunctional-requirements/#:~:text=Nonfunctional%20Requirements%20(NFRs)%20define%20system,system%20across%20the%20different%20backlogs.]
[Title: Nonfunctional Requirements]
[Author: NA] [Year: NA]

2. Overall Description

2.1 Product Perspective

This system is a new self-contained product designed for DCU students, professors and the admin for modules like CA326 and CA400. The aim of both modules is for students to design large-scale software or systems by applying the skills they have learned in previous years. The **Project Webpage** will be used as a timetable to view deadlines and available meeting or project demonstration dates as well as incorporate tools for writing forms, accepting project proposals, and viewing status reports of a list of all the projects undertaken and their status. The **Project Webpage** will send emails via Gmail to those with DCU accounts and add shortcuts to DCU's Gitlab for quick access. The purpose of this project is to save time and stress from both students and professors by organizing everything into one system. The **Project Webpage** will of course be as user-friendly as possible and easy to use with low memory load.

2.2 Product Functions

Below is a list of the main functions incorporated within the **Project Webpage**.

For students:

- Sign Up
- Add Details
- Edit Details
- Login
- Create Project Proposal
- Save proposal draft
- Edit Project Proposal
- Submit Project Proposal
- Schedule Meeting
- View Demonstration Scheduling Timetable
- Log out

For professors:

- Sign Up
- Add Details
- Login
- Approve Project
- Decline Project
- Schedule Meeting
- Add Available Time Slots for Demonstration
- View Projects
- View Individual Project Status
- Mark Project
- Log Out

For administrators:

- View All Users
- Confirm User Is a Professor
- View a List of All Students
- View a List of All Professors
- View a List of All Added Timetable Slots
- Create a Timetable for Demonstrating Projects

Diagram: Refer to *figure1*, the Use Case Model in Appendix B.

2.3 User Classes and Characteristics

The 3 main groups of users will be DCU students in 3rd or 4th year doing a large-scale project, DCU professors who will supervise the projects and the Administrators. The system will operate on desktop and requires Wi-Fi as well as a DCU email to use.

STUDENTS:
The students who will use the **Project Webpage** will be 20+ and will have basic knowledge of how to use a computer as well as finding the features of a webpage. Some users may have never used a computer before applying to DCU but will most likely have learned how to use one within the first 2 years and all modules related to large-scale projects are 3 or 4 years into the course.

PROFESSORS:
The professors will be 30+ and will have a basic to advanced knowledge with computers as well as the features of a webpage as they have used loop or their own custom site over the years. The professor's account will be validated by the admin to ensure they are a professor in DCU

ADMINISTRATORS:
The administrators of the **Project Webpage** will have absolute ownership and control of the system as well as high level permissions. The administrators will have advanced computer knowledge and the skill to update or modify the system as required.

2.4 Operating Environment

The **Project Webpage** will operate on desktop **OS** systems only, it will not be mobile friendly. It will be run on modern up to date browsers such as Chrome or Safari and will most likely not work on older browsers such as Internet Explorer. A good Wi-Fi connection will also be required.

To use the **Project Webpage** all users must be logged in with their DCU accounts, once students have signed up with their DCU accounts, they can then log in to the system and access all the timetables and tools the system has to provide.

Professors will be registered by the administrator to ensure no student signs in as a professor, the professors will be given a unique password which they must use to sign in along with their DCU email. From there they can view project proposals and supervise the undertaken projects and teams.

Desktop **OS**: Windows / MAC

2.5 Design and Implementation Constraints

Memory Constraint:
Due to the limit of the database, there is a limit on the scale to which the system can be tested, the **Project Webpage** will only be able to allow a certain number of users to sign in as it must store proposals, drafts, timetables, and lots of other data. This would pose problems later as more users sign up.

Time Constraint:
The deadline for the system will be 24th February which only allows for 55 days on the system as well as working on other modules. I specifically have chosen 3 modules instead of 1 for semester 2, which will significantly reduce the time available to complete the **Project Webpage**.

With little time to develop the system, there is also little time to test towards the end of the deadline which allows for lots of undiscovered bugs

Group Constraint:
Due to there being an odd number of students, I must work on this project alone which may reduce the available time significantly but also increases workload that would have been split between the two group members.

Design Constraint:
Will the **Project Webpage** be **UX** and **UI** friendly for all intended users, will there be a good balance of both accessibility and efficiency or must one be prioritized over the other completely.

User Requirements Constraint:
Will the system fulfil the requirements by the deadline? will the users be satisfied with the system and continue to use it or will it not save time and stress but simply be another alternative.

2.6 User Documentation

A user manual will be provided on how to use the system. It will include a step-by-step user instruction guide on how to use the system's major components along with images to be as user friendly as possible

A video on the walk-through of the system showing the major features in action will also be provided.

2.7 Assumptions and Dependencies

- The projects are in year 3 or year 4 so students in their 3rd or 4th year will be using this system
- Both students and professors have basic knowledge of using computers
- This system will be used only by DCU's faculty of computing students and professors
- The layout of the webpage depends on **Bulma's CSS** framework
- The Front-end depends on **Node**.
- The Back-end depends on **Django**.

3. External Interface Requirements

3.1 User Interfaces

The user interface for the system shall be compatible to any modern up-to-date browser such as Chrome, Safari, or Firefox by which user can access the system. The user interface shall be implemented using a front-end **JS** framework, **CSS** framework and **HTML**

The Web-Design shall use the 8 Golden Rules of Interface Design to ensure the UI design meets user satisfaction at its best and make the UI design easy to use with extremely low memory load.

The **Project Webpage** shall use the same colours as loop.dcu.ie to replicate the experience of using the educational loop page in another webpage. The colours associated with Loop are black, white and shades of blue.

A strong button contrast shall be implemented to show a call to action as well as improve readability for the button to be clear enough for the user to see. A help icon will link to the user manual on how to use the system. Error messages will be displayed on the webpage for any incorrect URL's or attempts to access a page when the user is not logged in.

The user will use a mousepad/mouse and keyboard to interact with the system through navigation and text input.

Refer to *figure2_1->2_5*, the mockup UIs of the system in Appendix B.

3.2 Hardware Interfaces

The system shall be available on modern up-to-date browsers that support **HTML**, **CSS** and **JS** on any desktop.

A keyboard and mouse shall be required to navigate through the system and use the provided functions.

3.3 Software Interfaces

The system shall have a web-based front-end interface for interaction between the user and the system through **JS**.

The system shall store the user's login details into the **SQLITE3** database through the back end and generate an encryption key.

The system shall communicate with the **SQLITE3** database through the back end to look at and receive data to be shown on the client side in JSON format.

The system shall send data from the client side to the back end in JSON format and store the data in the **SQLITE3** database.

The system shall be given access to the user’s email account to create and send an email.

Refer to *figure3*, the architecture diagram in Appendix B.

3.4 Communications Interfaces

The system shall use HTTPS for communication between the user and the system. client roles and server roles must be separated by a uniform interface. All client requests will be sent to the database via **API**s in JSON format and will receive a response in JSON format.

The system shall encrypt the user’s login key for communication security as to not allow any other user to view or steal the user’s login credentials.

The system shall communicate externally via DCU’s email.

4. System Features

4.1 Sign Up for a new account

4.1.1 Description and Priority

This function is the first step a new user uses to become a member of the system, if the user is new and a member of DCU, they will be given an account and added to the system.

Priority: High

A user must sign up first to use the system. if the user cannot sign up, they will not have an account and cannot access the system.

4.1.2 Stimulus/Response Sequences

The user will click on Sign Up and be directed to the sign-up page where the user will be required to fill in the mandatory details, if a username with the provided one already exists, the user will be alerted, the user's details will be stored into the database and the user will be provided with an account to access the system.

4.1.3 Functional Requirements

REQ-1: The system shall allow new users to sign up for an account.
REQ-2: The system shall allow user's to input their details and create an account.
REQ-3: The system shall store the user's provided details into the database for logging in.

4.2 Enter Student Details

4.1.1 Description and Priority

This function is the next step a new student must do to become a member of the system, if the user is a new student, and a member of DCU, they will be given access to the functions and pages available to students.

Priority: High

A user must enter their student details to use the system. if the user cannot create a student account, they will not have access to any functions for students

4.1.2 Stimulus/Response Sequences

The user will be redirected after signing up and will be required to fill in a form asking the student for their name, DCU email, student number, course and year. They will then be redirected to the student homepage.

4.1.3 Functional Requirements

REQ-1: The system shall allow new users to create a student account
REQ-2: The system shall allow user's to input their details and create an account.
REQ-3: The system shall store the user's provided details into the database for further requirements

4.3 Enter Professor Details

4.1.1 Description and Priority

This function is the next step a new professor must do to become a member of the system, if the user is a new professor, and a member of DCU, they will be given access to the functions and pages available to professors.

Priority: High

A user must enter their professor details to use the system. if the user cannot create a professor account, they will not have access to any functions for professors.

4.1.2 Stimulus/Response Sequences

The user will be redirected after signing up and will be required to fill in a form asking the professor for their name and DCU email.

4.1.3 Functional Requirements

REQ-1: The system shall allow new users to create a professor account.
REQ-2: The system shall allow users .to input their details and create an account.
REQ-3: The system shall store the user's provided details into the database for further requirements.

4.4 Log in to the System

4.1.1 Description and Priority

All users must log in to ensure they are members of the system, then proceed to use the tools and functions provided for each group to complete their necessary tasks.

Priority: High

If the user cannot log in to the system, they cannot access the tools and functions necessary to complete their tasks and as a result the entire system is deemed to be broken as there is no other method of accessing it. Unauthorized users must not be allowed access to the system.

4.1.2 Stimulus/Response Sequences

The user will put in their username and password then click log in, the system will authenticate the user with the provided details to check if they are a member and what permissions they have and allow access once identified the user is a member of the system.

4.1.3 Functional Requirements

REQ-1: The system shall allow users to input their username and password then press log in.
REQ-2: The system shall identify the user to ensure they are registered in the database.
REQ-3: The system shall authenticate the user to determine what level of access they have
REQ-4: The system shall not allow unregistered users to log in and access the system.
REQ-5: When the user is logged in, the system shall set the user's logged in access to true.

4.5 Edit Student Details

4.1.1 Description and Priority

The student user can edit their course and year details on the system if they have changed their course or have gone up an academic year.

Priority: Low

If the student cannot update their details, it will not cause any problem to the system, the student can simply inform their project supervisor of the error and tell them the correct details

4.1.2 Stimulus/Response Sequences

The user will click on Edit details and be redirected to a page with a form that displays the user's details and allows the user to change their course and year.

4.1.3 Functional Requirements

REQ-1: The system shall allow DCU students to edit their course or year details.
REQ-2: When the student submits their new details, the system shall update the database with the new details.

4.6 Edit Project Proposal Draft

4.1.1 Description and Priority

A project proposal will be automatically created for the student to work on, each time the project proposal is saved, or the proposal was rejected, student's will have their data saved onto the proposal form to continue working on each time

Priority: Medium

The student must be able to come back and update their draft whenever they wish otherwise it will remain as a still copy and cannot be updated.

4.1.2 Stimulus/Response Sequences

The user will open up the proposal form page and will be directed to the project proposal draft page where they can continue to input or modify their already added details.

4.1.3 Functional Requirements

REQ-1: The system shall allow DCU students to edit their created project proposal.
REQ-2: The system shall allow DCU students to input data into the draft.
REQ-3: The system shall fetch the stored input data and display it onto the proposal form.

4.7 Save Project Proposal Draft

4.1.1 Description and Priority

Once a new project proposal has been created, students will then have the option of saving the draft and its contents.

Priority: Medium

After spending a long time, it will be frustrating for the user to finish the project proposal and not be able to save it or save it but does not show the saved version the next time the user comes back to the project proposal draft.

4.1.2 Stimulus/Response Sequences

The user will click on save draft, the system will store the user's input into the database and then notify the user that the draft has been saved.

4.1.3 Functional Requirements

REQ-1: The system shall allow DCU students to save their project proposal.
REQ-2: The system shall store the input data from the project proposal draft into the database.

4.8 Submit Project Proposal

4.1.1 Description and Priority

Once the student has saved their draft, they will have the option to submit their project proposal to their chosen DCU professor.

Priority: Medium

If the user is unable to send the project proposal, the page can be downloaded as a PDF and forwarded by email to the chosen professors but will require a vast amount of time to do. It can pose to be a problem if the professor does not receive it and it not notified of the proposal.

4.1.2 Stimulus/Response Sequences

The user will click on send draft and choose who to send the proposal to, the system will update the professor's database with the students' proposals.

4.1.3 Functional Requirements

REQ-1: The system shall allow DCU students to submit project proposals.
REQ-2: The system shall store the student's project proposals in the professor's database.
REQ-3: The system shall send an email to notify the professor of the sent proposal.
REQ-4: The system shall update the professor's interface to view and accept or decline the project proposal.

4.9 Submit Functional Specification

4.1.1 Description and Priority

Students can submit their functional specification file and may edit their submission as many times as they wish.

Priority: Low

If the user is unable to send their file or cannot update the file they uploaded, they can simply email their project supervisor with their functional specification.

4.1.2 Stimulus/Response Sequences

The user will click on submit functional specification and choose a file from their library to submit, once submitted, the page will reload to show the student their submitted file and may update their submission.

4.1.3 Functional Requirements

REQ-1: The system shall allow DCU students to submit a file.
REQ-2: The system shall store the student's submitted file in the database.
REQ-3: The system shall allow the student to update their file submission as many times as they wish.

4.10 Submit Documentation

4.1.1 Description and Priority

Students can submit their documentation file and may edit their submission as many times as they wish.

Priority: Low

If the user is unable to send their file or cannot update the file they uploaded, they can simply email their project supervisor with their documentation.

4.1.2 Stimulus/Response Sequences

The user will click on submit documentation and choose a file from their library to submit, once submitted, the page will reload to show the student their submitted file and may update their submission

4.1.3 Functional Requirements

REQ-1: The system shall allow DCU students to submit a file.
REQ-2: The system shall store the student's submitted file in the database.
REQ-3: The system shall allow the student to update their file submission as many times as they wish.

4.11 View Project Proposal

4.1.1 Description and Priority

The professor can choose to view the full project proposal sent by the student

Priority: Low

If the professor cannot view it for any reason, they can ask the student via email to send the project proposal by email but will create some hassle and waste time.

4.1.2 Stimulus/Response Sequences

The professor will click View Full Proposal and be directed to another webpage to view the full proposal

4.1.3 Functional Requirements

REQ-1: The system shall allow DCU professors to click on the button to view the project proposal.
REQ-2: The system shall redirect the professor to another webpage to view the full project proposal.

4.12 Accept Project Proposal

4.1.1 Description and Priority

The professor will have the ability to accept and supervise the student's project. The student will be notified via an automated email.

Priority: High

The student must know that their project has been accepted by a supervisor and must be in the professor's database for the list of students. If this function does not work, the entire system cannot work properly so it is vital all aspects of this function work correctly and on time.

4.1.2 Stimulus/Response Sequences

The professors clicks on Accept Project, the student will be updated into the professor's database, an automated email will be sent to the student to notify of the acceptance.

4.1.3 Functional Requirements

REQ-1: The system shall provide the option for professors to accept a project proposal

REQ-2: The system shall add the student to the professor's database as the list of students under supervision.

REQ-3: The system shall send the student an automated email to notify of the acceptance.

4.13 Reject Project Proposal

4.1.1 Description and Priority

The professor will have the ability to reject the student's project. The student will be notified via email with feedback.

Priority: Low

A proposal is considered rejected unless stated otherwise, if the function does not work, the professor can simply write an email to the student with feedback on their proposal.

4.1.2 Stimulus/Response Sequences

The professors clicks on Reject Project, an email with written feedback will be sent to the student.

4.1.3 Functional Requirements

REQ-1: The system shall provide the option for professors to reject a project proposal

REQ-2: The system shall add a text input box for the professor to input feedback on the student's project proposal
REQ-3: The system shall send the student an email with the written feedback to notify of the rejected proposal.

4.14 Schedule Meeting

4.1.1 Description and Priority

This function allows both students or professors to schedule a supervision meeting, the professor has the option to do so in case of emergency or urgent matter to discuss

Priority: Medium

Both students and professors must be notified of the meeting request as soon as possible or it could cause problems if neither is notified, the meeting can be scheduled via email but will create a lot of hassle.

4.1.2 Stimulus/Response Sequences

The student or professor will click on Schedule Meeting and input the time range and date they wish to have a meeting on.

4.1.3 Functional Requirements

REQ-1: The system shall allow students and professors to schedule for a meeting
REQ-2: The system shall allow the user to input text and click Send Meeting Time.

4.15 Accept Meeting

4.1.1 Description and Priority

The student or professor will be notified of the requested meeting time and be given the option to accept the meeting time and send a project supervision form via email.

Priority: Low

If the student or professor cannot accept the requested the meeting time, they can accept via email.

4.1.2 Stimulus/Response Sequences

The user will click on accept meeting and a project supervision form will be sent to the student via email.

4.1.3 Functional Requirements

REQ-1: The system shall allow the user to accept the meeting
REQ-2: The system shall update the system timetable of the meeting time
REQ-3: The system shall send the student a project supervision form via email.

4.16 Reject Meeting

4.1.1 Description and Priority

The student or professor will be notified of the requested meeting time and be given the option to reject the meeting time.

Priority: Low

If the student or professor cannot reject the requested the meeting time, they can suggest another time via email.

4.1.2 Stimulus/Response Sequences

The user will click on reject meeting and the sender will be notified.

4.1.3 Functional Requirements

REQ-1: The system shall allow the user to reject the meeting.
REQ-2: The system shall notify the sender of the rejected time.

4.17 View Meeting History

4.1.1 Description and Priority

This function allows both students or professors to view their meeting history whether past, upcoming, accepted, rejected, or pending approval.

Priority: Medium

Both students and professors may require viewing their meeting history to view their rejected meeting and as proof of meeting with their supervisor/student.

4.1.2 Stimulus/Response Sequences

The student or professor will click on View Meeting History and view a table with all their meeting history data.

4.1.3 Functional Requirements

REQ-1: The system shall allow students and professors to view their meeting history.
REQ-2: The system shall redirect the user to a page with a table of their meeting history data.

4.18 Add Timetable Slot

4.1.1 Description and Priority

The professor can add a timetable slot for when the are available to supervise student project demonstrations.

Priority: Medium

If a professor cannot add a timetable slot, they must write down their availability or send an email each time which will increase the workload and time wasted.

4.1.2 Stimulus/Response Sequences

The user will click on add timetable and be directed to a page with a form that takes the date and time as input for when the professor is available at that time.

4.1.3 Functional Requirements

REQ-1: The system shall allow professors to add a timetable slot.
REQ-2: When the user submits their available time, the system shall store the user's input into the database for creating the demonstration timetable

4.19 Remove Timetable Slot

4.1.1 Description and Priority

The professor can remove an added timetable slot if they have made a mistake with the added time or are not available anymore during that time.

Priority: High

If a professor cannot remove an added timetable slot, they must tell the admin before the timetable is generated but failure to do so results in the timetable being generated with the wrong times and causes a lot of stress for changing the timetable.

4.1.2 Stimulus/Response Sequences

The user will view a table of their added timetable slots and click remove on any one timetable slot to remove that time from the systems database.

4.1.3 Functional Requirements

REQ-1: The system shall allow professors to remove an added timetable slot.
REQ-2: When the user removed their available time, the system shall remove the requested time from the database.

4.20 View Projects Status Report

4.1.1 Description and Priority

This function is for professors to the view the status of all the projects they have undertaken, including a project search bar to find a specific project. It also shows the dates for the project demonstration for a student.

Priority: High

The professor must be able to view all the projects to know which projects they are supervising and their status. If the function does not work, the professors will have to email every student they are supervising if they remember and ask for the current details of their project which they may provide false information to.

4.1.2 Stimulus/Response Sequences

The professor will click on View Status Report and will be redirected to a new page full of project listings with details

4.1.3 Functional Requirements

REQ-1: The system shall allow DCU professors to view the status report of all projects.

REQ-2: The system shall display the status and details of all projects in list order.

4.21 View Project Status of individual project

4.1.1 Description and Priority

This function is for professors to the view the full details of a specific project from the list of projects including the full title and description of the project as well as the project results report.

Priority: High

The professor must be able to view the full details and status of each individual project otherwise there is no way of documenting its status apart from writing it in a separate document or emailing the student in which neither is satisfactory.

4.1.2 Stimulus/Response Sequences

The professor will click on View Project Status and will be open up the page with the full details and status of that individual project.

4.1.3 Functional Requirements

REQ-1: The system shall allow DCU professors to view the status of an individual project.

REQ-2: The system shall display the full details and status of the individual project.

4.22 Mark Project

4.1.1 Description and Priority

The professor will input the title and feedback per comment box and add more boxes for various aspects of the project, then click Mark Project to send the project marking form to the student via email

Priority: Low

This is similar to writing an email to the student but in a more organized manner, if the function does not work, the professor can simply send the project marking form via email.

4.1.2 Stimulus/Response Sequences

The professor will click on Mark Project and will be redirected to a new page, the professor will input text into the title area and input text into the feedback area, the professor can choose to add extra boxes to provide feedback on different areas of the project.

4.1.3 Functional Requirements

REQ-1: The system shall allow DCU professors to Mark a student project.
REQ-2: The system shall allow the user to input text into the comment box.
REQ-3: The system shall display the add option to add another comment box.
REQ-4: When the user clicks the add option, another comment box shall be added.

4.23 View Demonstration Timetable for Professor

4.1.1 Description and Priority

Once the demonstration timetable has been created, the professor can go to the demonstration timetable page and view their time slots, location and the student they are doing the demonstration with

Priority: Medium

The professor must be able to view their demonstration timetable or else they will not know when and where to attend and must ask the admin for the whole timetable and filter for their name.

4.1.2 Stimulus/Response Sequences

The user will redirect to the timetable page and view a table of time slots with the student’s name, date, time, and location of the said demonstration.

4.1.3 Functional Requirements

REQ-1: The system shall allow the professor to view their demonstration timetable
REQ-2: The system shall redirect the professor to a page with a table of their demonstration timetable data.

4.24 Assign Deadline

4.1.1 Description and Priority

This function will allow the admin to set the deadlines for submitting a project proposal, functional specification and the project.

Priority: High

If the admin cannot set the deadlines, the system cannot stop the students from submitting past the final date and the professors will be unable to tell when each document was submitted.

4.1.2 Stimulus/Response Sequences

The administrator will go to the deadline page with a form and input 3 separate dates for the project proposal deadline, the functional specification deadline and the project deadline, then submit the details and store it into the database.

4.1.3 Functional Requirements

REQ-1: The system shall allow the admin to set the deadline for submitting project proposals, functional specification, and the project.
REQ-2: The system shall allow the admin to input 3 separate dates.
REQ-3: The system shall store the admins set dates into the database.

4.25 Accept New Professor

4.1.1 Description and Priority

This function will allow the admin to accept a newly registered professor into the system to allow them professor privileges if they recognize the user’s name and email.

Priority: High

If the admin cannot accept any new professors into the system, there will not be any professors with access to use the system and as such the system will not be usable in any way.

4.1.2 Stimulus/Response Sequences

The administrator will view a table with the list of all new registered professors and click on accept beside the user’s details to allow them entry into the system as a professor.

4.1.3 Functional Requirements

REQ-1: The system shall allow the admin to view a list of new registered professor.
REQ-2: The system shall allow the admin to choose and accept a new professor into the system.
REQ-3: The system shall grant the chosen professor with access to all functions and pages for professors.

4.26 Reject New Professor

4.1.1 Description and Priority

This function will allow the admin to reject a newly registered professor entry into the system if they do not recognize the user's name and email.

Priority: Low

If the admin cannot reject any new professors into the system, there will not be any changes as that user will not have any access to any provided functions for either students, professors or the administrator.

4.1.2 Stimulus/Response Sequences

The administrator will view a table with the list of all new registered professors and click on reject beside the user's details to deny them entry into the system.

4.1.3 Functional Requirements

REQ-1: The system shall allow the admin to view a list of new registered professor.
REQ-2: The system shall allow the admin to choose and reject a new professor entry into the system.
REQ-3: The system shall remove the chosen user from the list of new registered professors and remove their professor account details.

4.27 Assign Student to Professor

4.1.1 Description and Priority

This function will allow the admin to assign students to a chosen project supervisor if the students do not have a project supervisor by the given deadline.

Priority: High

If the students cannot be assigned to a project supervisor, they cannot use interact with their supervisor through the system and thus cannot use most of the system functions.

4.1.2 Stimulus/Response Sequences

The administrator will view 2 tables with a list of all students without a project supervisor and a list of all professors, the admin will choose a minimum of one student and only one professor to assign the student(s) to the chosen professor as a supervisor.

4.1.3 Functional Requirements

REQ-1: The system shall allow the admin to view a list of students without a supervisor and all professors.
REQ-2: The system shall allow the admin to choose at least one student and only one professor.
REQ-3: The system shall assign each chosen student to the chosen professor as a supervisor and store it into the database.

4.28 View Professors Added Time Slots

4.1.1 Description and Priority

This function will allow the admin to view each professors added timetable slot to determine if a professor has not enough time slots or some time slots may require change.

Priority: Medium

If the admin cannot view any professors added time slots, they will have to ask each individual professor to send their added timeslots by email and it will be messy

4.1.2 Stimulus/Response Sequences

The administrator will view the chosen professors name and a table with a list of all added timetable slots by that professor which includes the date and time.

4.1.3 Functional Requirements

REQ-1: The system shall allow the admin to view any chosen professor’s added timetable slots.

4.29 Add Venue Location

4.1.1 Description and Priority

This function will allow the admin to add a venue and the number of places available in that venue in one hour for students to demonstrate their project at a location.

Priority: Medium

If the admin cannot add any venues, the timetable will be generated without any set locations which will require the admin to manually do.

4.1.2 Stimulus/Response Sequences

The administrator will go to the add venue page with a form to input a location and a number for the number of available places in that location per hour.

4.1.3 Functional Requirements

REQ-1: The system shall allow the admin to input a new venue location and a number.
REQ-2: The system shall stores the admin's input into the database

4.30 Remove Venue Location

4.1.1 Description and Priority

This function will allow the admin to remove an added venue.

Priority: Medium

If the admin cannot remove an added venue, it will be used when creating the timetable and as a result will add locations that may not be available.

4.1.2 Stimulus/Response Sequences

The administrator will view a table of added venue locations and click remove on any one to remove that location from the systems database.

4.1.3 Functional Requirements

REQ-1: The system shall allow the admin to remove an added venue location.
REQ-2: When the admin removes the location, the system shall remove the requested venue from the database.

4.31 Edit Venue Location

4.1.1 Description and Priority

This function will allow the admin to edit an added venue and its number of available places.

Priority: Medium

If the admin cannot edit an added venue, it will be used when creating the timetable and as a result will add locations or use a number of places that may not be available.

4.1.2 Stimulus/Response Sequences

The administrator will view a table of added venue locations, change the venue location or number of places and click edit on any one to save the changes into the systems database.

4.1.3 Functional Requirements

REQ-1: The system shall allow the admin to edit an added venue location.
REQ-3: The system shall allow the admin to change the value of the venue or its number.
REQ-2: When the admin submits the edit, the system shall update the requested venue in the database.

4.32 Create Demonstration Timetable

4.1.1 Description and Priority

This function will allow the admin to create a demonstration timetable.

Priority: High

This function is necessary to immensely reduce the amount of stress and workload of creating a timetable with hundreds of students by simply inputting a date and a number then clicking a button to have it automatically generated.

4.1.2 Stimulus/Response Sequences

The administrator will go to the scheduler page with a form to take a starting date for the demonstration timetable and the number of required assessors per demonstration as its input then submit and automatically generate the timetable.

4.1.3 Functional Requirements

REQ-1: The system shall allow the admin to access the scheduler page.
REQ-2: The system shall allow the admin to input a date and a number.
REQ-3: The system shall allow the admin to submit the input details and automatically create a timetable for project demonstration or return an error with the limited amount of available time slots

4.33 Log Out of the System

4.1.1 Description and Priority

The user that is logged in can log out of the system.

Priority: Low

If the user cannot log out for any reason, they simply need to close the browser or restart their desktop to be automatically logged out.

4.1.2 Stimulus/Response Sequences

The user will click log out and the system will log the user out of the system, the user will then have to log in again to use the system.

4.1.3 Functional Requirements

REQ-1: The system shall allow the user to log out.
REQ-2: When the user is logged out, the system shall set the user's logged in access to false.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

The system shall load the browser within 10 seconds.
The system shall store data into the database within 5 seconds of user data input.
The system shall update the interface within 5 seconds of user interaction

5.2 Safety Requirements

The system shall display a white background to reduce eye strain for users.
The system shall encrypt the user’s password to ensure user privacy and security

5.3 Security Requirements

The system shall only allow the REST **API**’s to communicate with the database.
When a new user signs up, the system shall store the user’s encrypted details in the database.
When a new user signs up, the system shall generate a login key for the user to login securely.
When a user logs in, the system shall authenticate the user using the user’s provided details and login key.
If the login information is incorrect, the system shall not allow the user to login.

5.4 Software Quality Attributes

The system shall be easy to learn for new users.
The system shall be accessible on desktop OS such as MAC or Windows.
The system shall be accessible to the user 24 hours a day unless specified otherwise for update or maintenance.

5.5 Business Rules

The system shall only be available to members of DCU.

The system shall only allow the administrator to add professors accounts and update or modify the system's database.

The system shall not allow students to sign up as professors.

The system shall allow students to only use tools or functions provided for students.

6. Other Requirements

Portability:

The system shall be accessible and logged in from any desktop supporting browsers.

Logical Database Model

Refer to *figure4, figure5, figure6, figure7, and figure8* logical database models in Appendix B.

Preliminary Schedule

Refer to *figure8,* Gantt Chart in Appendix B.

Appendix A: Glossary

HTML: HyperText Markup Language, this is the language used to display elements onto a web browser.

CSS: Cascading Style Sheets, this is the language used to present the HTML elements in an aesthetic manner onto a web browser.

JS: JavaScript: this is the language used to make the web browser interactive.

MYSQL: An open-source relational database management system.

API: Application Programming Interface: A method of communication between 2 or more computers.

UI: User Interface: the interaction between the user and the computer.

UX: User Experience: the user's experience with the interaction with the computer.

OS: Operating System: System that manages the computers hardware and software resources to provide services for its user

Django: The Python library used to design the backend of the Web-based Student Project Management Systems.

Node: The JS Framework used to design the frontend of the Web-based Student Project Management Systems.

Bulma: The CSS Framework used for the design of the front-end to make it more aesthetically pleasing.

Appendix B: Analysis Models

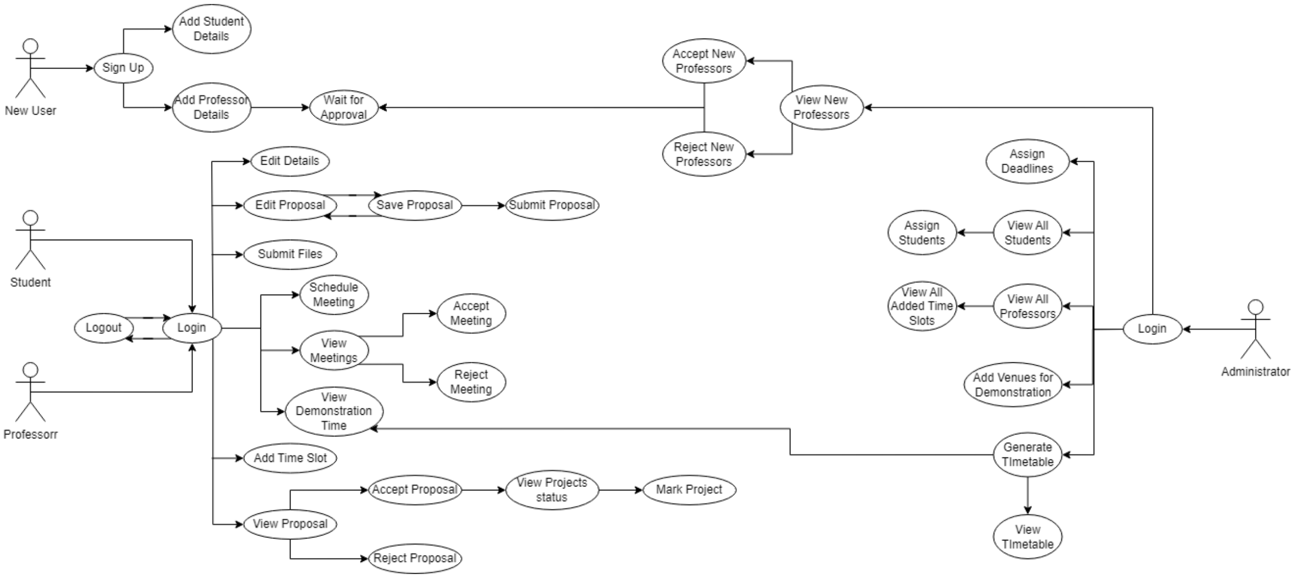


figure1: Use Case Model

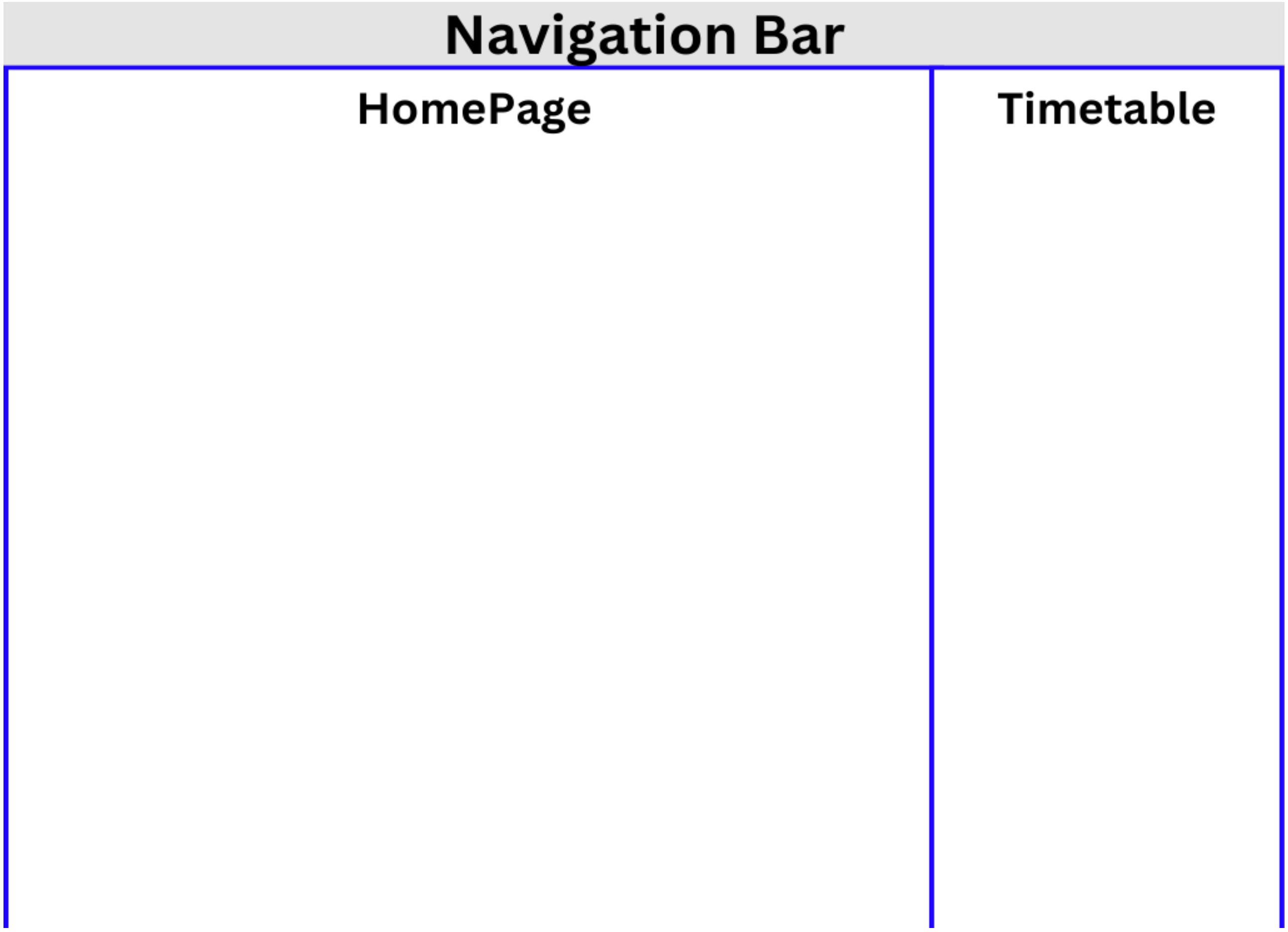


figure2_1: Homepage Mockup UI

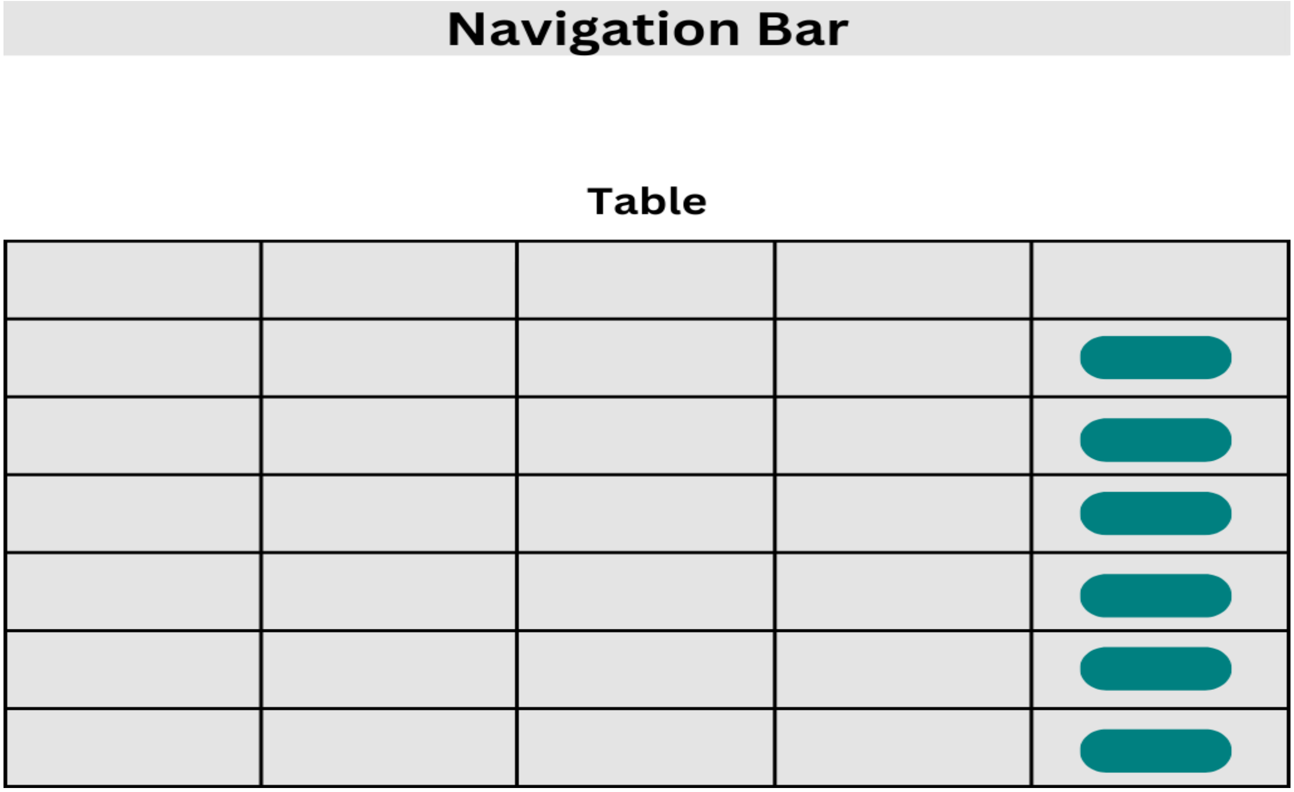


figure2_2: Table Mockup UI

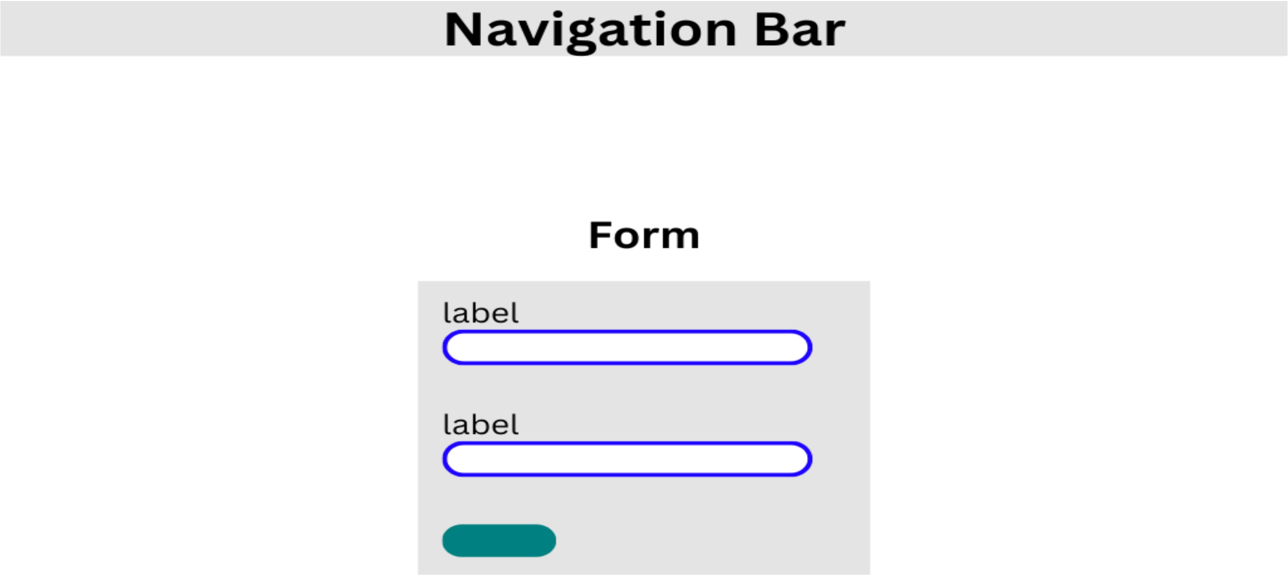


figure2_3: Form Mockup UI

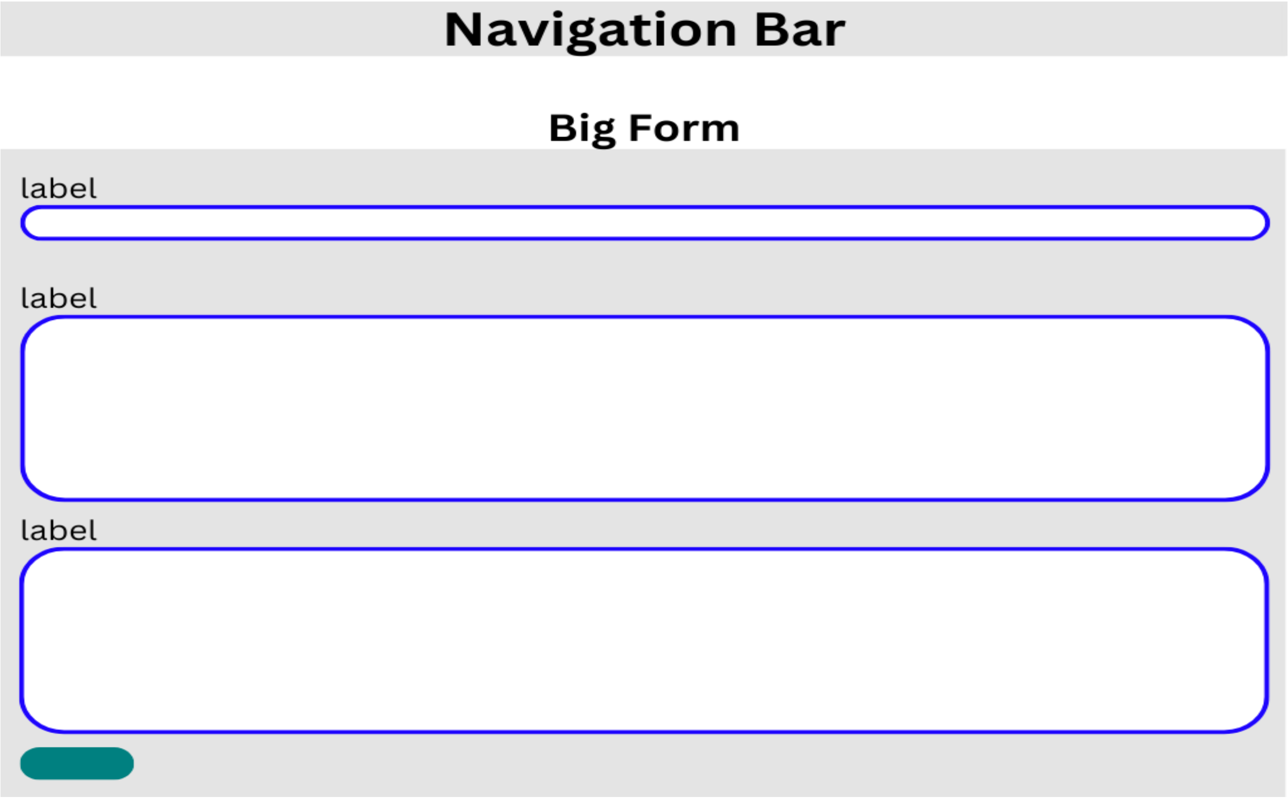


figure2_4: Big Form Mockup UI

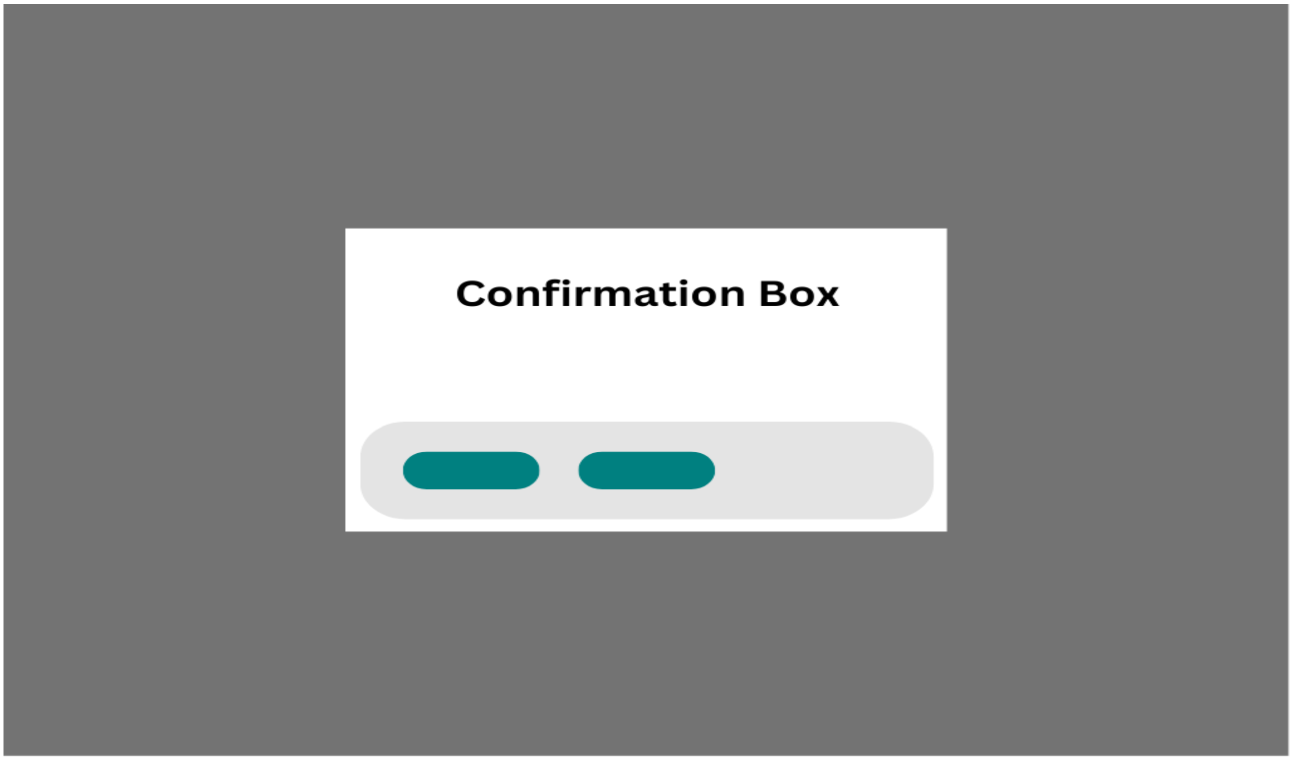


figure2_5: Confirmation Box Mockup UI

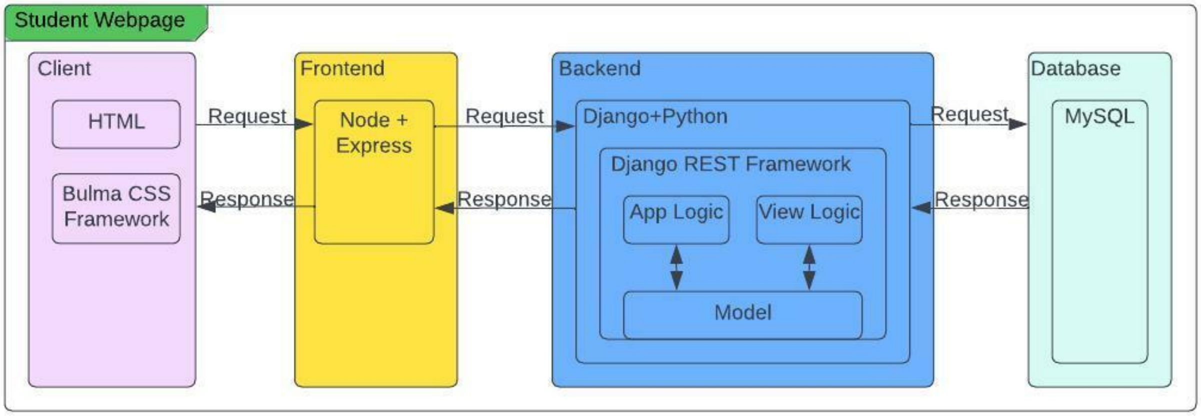


figure3: Architecture Diagram

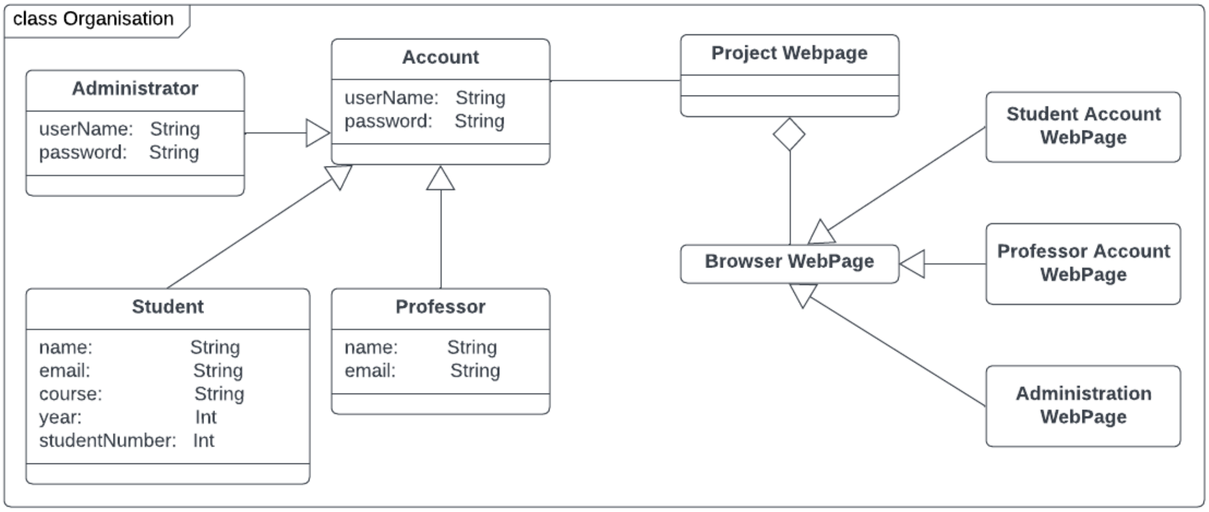


figure4: Logical Database Model

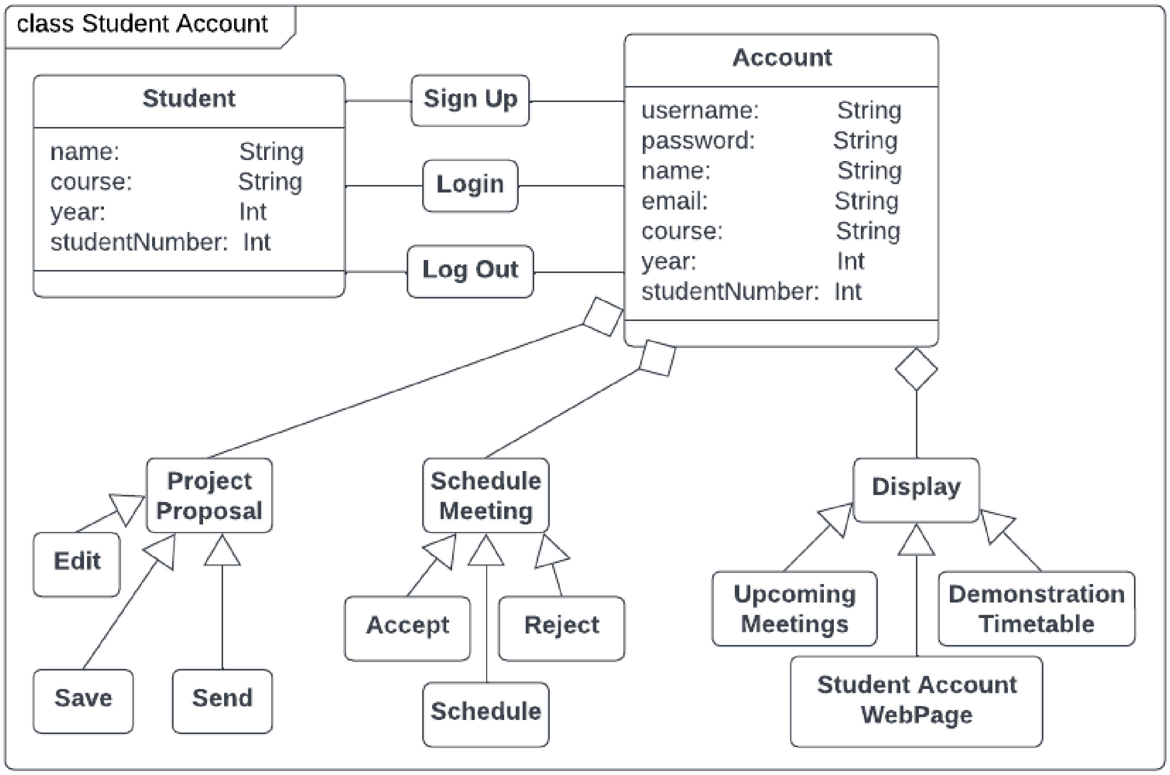


figure5: Logical Database Model

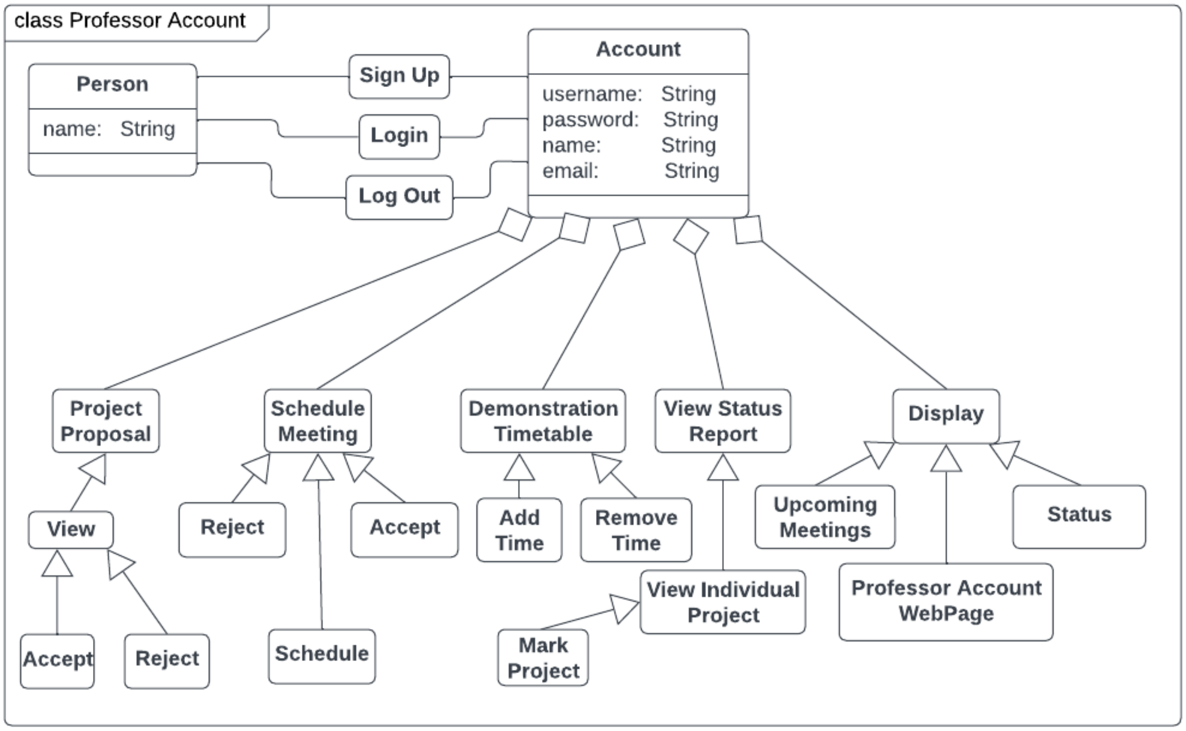


figure6: Logical Database Model

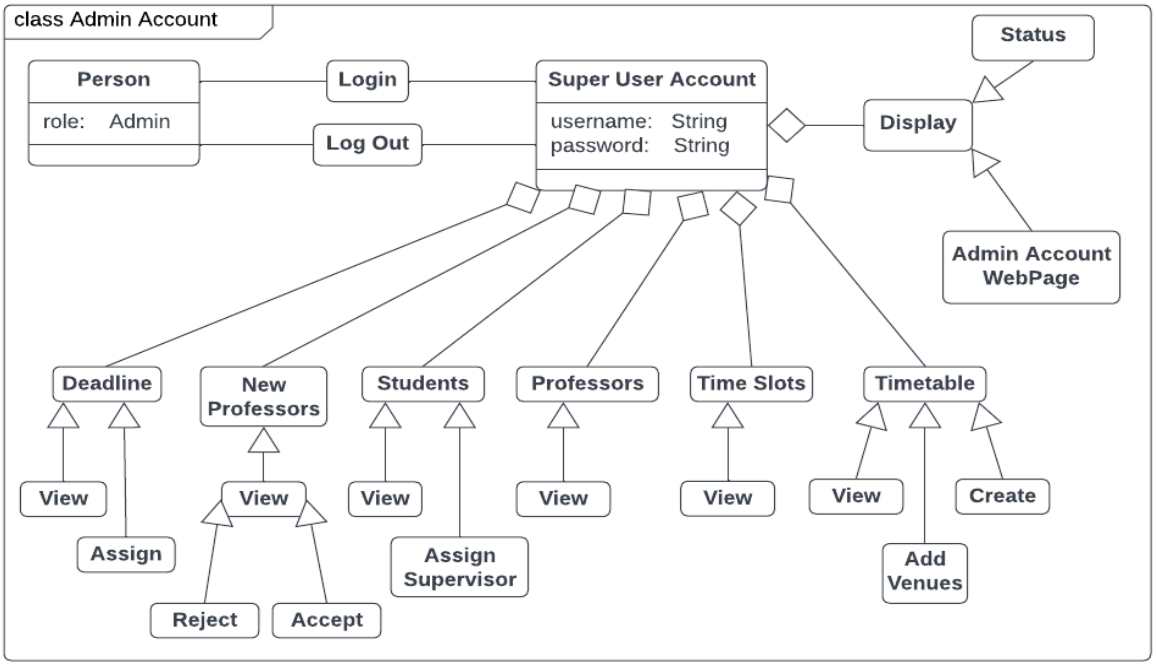
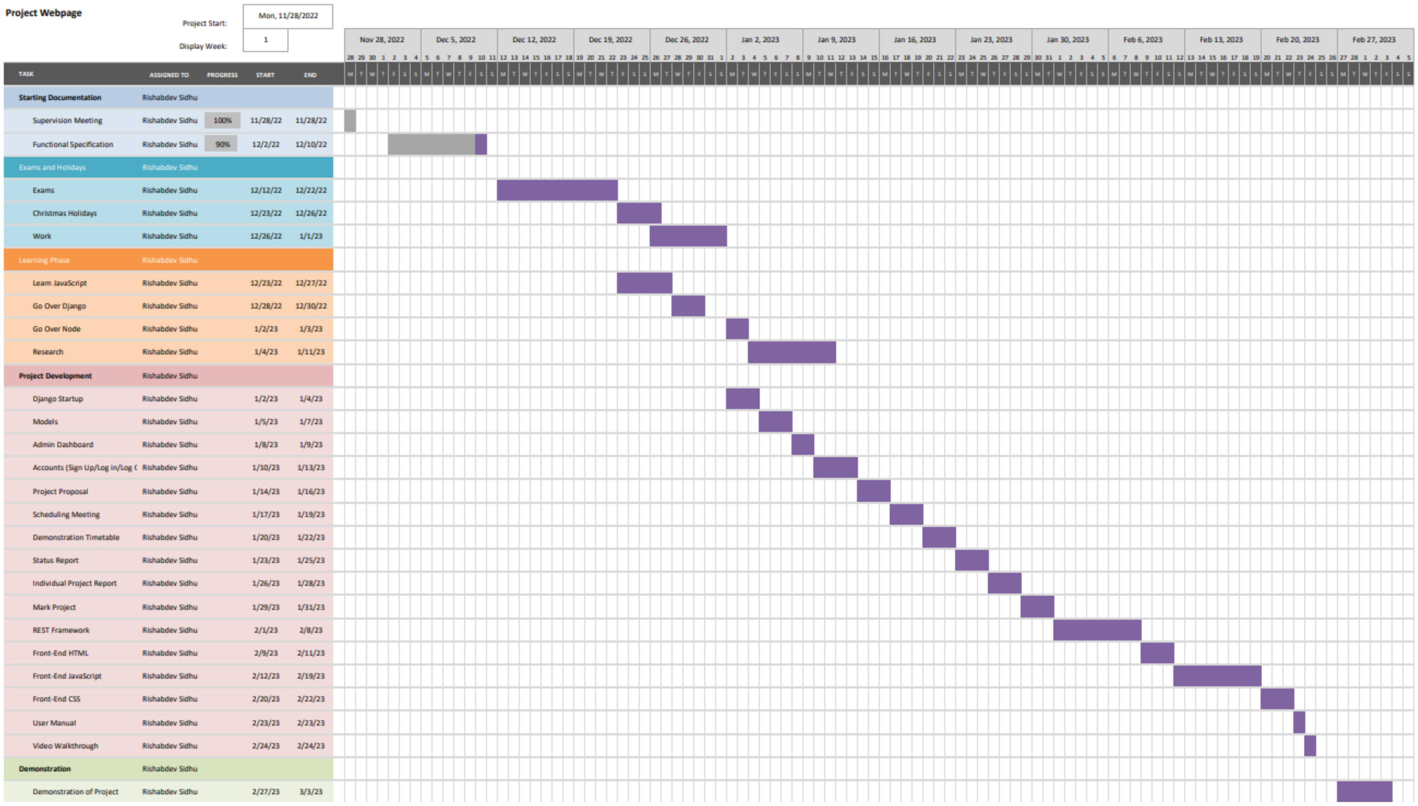


figure7: Logical Database Model



Project Webpage

Project Start: Mon, 11/28/2022

Display Week: 1

TASK	ASSIGNED TO	PROGRESS	START	END
Starting Documentation	Rishabdev Sidhu			
Supervision Meeting	Rishabdev Sidhu	100%	11/28/22	11/28/22
Functional Specification	Rishabdev Sidhu	90%	12/2/22	12/10/22
Exams and Holidays	Rishabdev Sidhu			
Exams	Rishabdev Sidhu		12/12/22	12/22/22
Christmas Holidays	Rishabdev Sidhu		12/23/22	12/26/22
Work	Rishabdev Sidhu		12/26/22	1/1/23
Learning Phase	Rishabdev Sidhu			
Leam JavaScript	Rishabdev Sidhu		12/23/22	12/27/22
Go Over Django	Rishabdev Sidhu		12/28/22	12/30/22
Go Over Node	Rishabdev Sidhu		1/2/23	1/3/23
Research	Rishabdev Sidhu		1/4/23	1/11/23
Project Development	Rishabdev Sidhu			
Django Startup	Rishabdev Sidhu		1/2/23	1/4/23
Models	Rishabdev Sidhu		1/5/23	1/7/23
Admin Dashboard	Rishabdev Sidhu		1/8/23	1/9/23
Accounts (Sign Up/Log in/Log (	Rishabdev Sidhu		1/10/23	1/13/23
Project Proposal	Rishabdev Sidhu		1/14/23	1/16/23
Scheduling Meeting	Rishabdev Sidhu		1/17/23	1/19/23
Demonstration Timetable	Rishabdev Sidhu		1/20/23	1/22/23
Status Report	Rishabdev Sidhu		1/23/23	1/25/23
Individual Project Report	Rishabdev Sidhu		1/26/23	1/28/23
Mark Project	Rishabdev Sidhu		1/29/23	1/31/23
REST Framework	Rishabdev Sidhu		2/1/23	2/8/23
Front-End HTML	Rishabdev Sidhu		2/9/23	2/11/23
Front-End JavaScript	Rishabdev Sidhu		2/12/23	2/19/23
Front-End CSS	Rishabdev Sidhu		2/20/23	2/22/23
User Manual	Rishabdev Sidhu		2/23/23	2/23/23
Video Walkthrough	Rishabdev Sidhu		2/24/23	2/24/23
Demonstration	Rishabdev Sidhu			
Demonstration of Project	Rishabdev Sidhu		2/27/23	3/3/23

Figure8: Gantt Chart

Appendix C: To Be Determined List

Section 3.1 User Interface
The **UI** mockup design is only a template of the actual design which will consistently change and adapt to the 9 guidelines so there is no certain full mockup design but will be updated as the web-based aspect of the system is being developed.

Section 4 System Features
As the system is being developed, some system features may be changed to build the system, and some may be added to the system as a result. Any new features will be documented and added to a later version of this document.

Section 6 Other Requirements
As the system is being developed, other requirements that have been missed in this document may be found and added to a later version of the document.

V1 Changes:
Figure 1, Use Case Model has been updated to display the actors and their use cases in a more readable way and up-to-date with the system's current features

Figure 2, Mockup UI has been updated to the system's current UI Design with the system's most used templates including the homepage, tables, forms and confirmation box

Figures 4, 5, 6 and 7 Logical Database Models updated to the system's current logical functions

Section 4, Added the following:

- 4.2 Enter Student Details
- 4.3 Enter Professor Details
- 4.9 Submit Functional Specification
- 4.10 Submit Documentation
- 4.17 View Meeting History
- 4.18 Add Timetable Slot
- 4.19 Remove Timetable Slot
- 4.24 Assign Deadline
- 4.25 Accept New Professor
- 4.26 Reject New Professor
- 4.27 Assign Student to Professor
- 4.28 View Professors Added Time Slots
- 4.29 Add Venue Location
- 4.30 Remove Venue Location
- 4.31 Edit Venue Location
- 4.32 Create Demonstration Timetable