
Software Requirements Specification

for

**Web-based Student Project Management
Systems**

Prepared by Shabb03

Student Number: *****

08/12/2022

Table of Contents

Table of Contents	ii
Revision History	iii
1. Introduction	1
1.1 Purpose.....	1
1.2 Document Conventions.....	1
1.3 Intended Audience and Reading Suggestions	1
1.4 Product Scope	2
1.5 References.....	2
2. Overall Description	3
2.1 Product Perspective.....	3
2.2 Product Functions	3
2.3 User Classes and Characteristics	4
2.4 Operating Environment.....	4
2.5 Design and Implementation Constraints	4
2.6 User Documentation	5
2.7 Assumptions and Dependencies	5
3. External Interface Requirements	6
3.1 User Interfaces	6
3.2 Hardware Interfaces	6
3.3 Software Interfaces	6
3.4 Communications Interfaces	7
4. System Features	8
4.1 Sign Up for a new account.....	8
4.2 Log in to the system.....	8
4.3 Create new project proposal.....	9
4.4 Save project proposal draft	9
4.5 Edit project proposal draft	10
4.6 Send project proposal.....	10
4.7 View project proposal.....	11
4.8 Accept project.....	11
4.9 Reject project.....	12
4.10 Schedule meeting.....	12
4.11 Accept meeting	13
4.12 Reject meeting	13
4.13 View demonstration timetable	14
4.14 Edit demonstration timetable	14
4.15 View status report.....	15
4.16 View project status of individual project	15
4.17 Mark project.....	16
4.18 Log out of the system.....	16
4.19 Add Account (Administrator).....	17
5. Other Nonfunctional Requirements.....	18
5.1 Performance Requirements	18
5.2 Safety Requirements	18
5.3 Security Requirements	18
5.4 Software Quality Attributes	18
5.5 Business Rules	19
6. Other Requirements & Preliminary Schedule.....	19
Appendix A: Glossary.....	20
Appendix B: Analysis Models.....	21
Appendix C: To Be Determined List.....	25

Revision History

Name	Date	Reason For Changes	Version
Shabb03	08/12/22	New SRS Draft	V0

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 coordinators, 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 this document to understand it. This includes the purpose of the product, the different types of users who will use the product and the functions each type of user will have access to use, where the product can be used, 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 great 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 a 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 a detailed list of any 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 coordinators and 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 create a project marking form as well as view status reports of the tasks the supervised students have done.

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.

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.](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 both DCU students and professors 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
- Login
- Create Project Proposal
- Save proposal draft
- Edit Project Proposal
- Schedule Meeting
- View Scheduling Timetable
- Log out

For professors:

- Login
- Approve Project
- Decline Project
- Schedule Meeting
- Mark Project
- View Projects
- View Individual Project Status
- Log out

For administrators:

- Add accounts

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 account 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 provided to them by the administrators and need only log in.

ADMINISTRATORS:

The administrators of the **Project Webpage** will have absolute ownership of the system as well as controls and permissions to add and remove accounts. The administrators will have advanced computer knowledge and the skill to update or modify the system as required. The administrators will add the DCU professors account and provide them with a password.

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 have to 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 9 design guideline principles 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*, the mockup UI 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 **MYSQL** database through the back end and generate an encryption key.

The system shall communicate with the **MYSQL** 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 **MYSQL** 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 **APIs** 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: Medium

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 but the administrator can provide an account if necessary.

4.1.2 Stimulus/Response Sequences

The user will click on New User and be directed to the sign up page where the user will be required to fill in the mandatory details and only user's with DCU accounts shall be accepted as members. If the email provided contains "@mail.dcu.ie", 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 DCU students 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.
- REQ-4: The system shall not allow user's with an email not of DCU's to sign up.

4.2 Log in to the system

4.1.1 Description and Priority

Both Students and professors must log in to ensure they are members of DCU, then proceed to use the tools and functions provided for each group to complete their project.

Priority: High

If the user cannot log in to the system, they cannot access the tools and functions necessary to complete their project 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 mandatory details and click log in, the system will authenticate the user with the provided user and allow access once identified the user is a member of the system.

4.1.3 Functional Requirements

- REQ-1: The system shall allow user's to input their details and press log in.
- REQ-2: The system shall authenticate the user to ensure they are registered in the database.
- REQ-3: The system shall not allow unregistered user's to log in and access the system.
- REQ-4: When the user is logged in, the system shall set the user's logged in access to true.

4.3 Create new Project Proposal

4.1.1 Description and Priority

The student user can create a new project proposal draft in which they enter the mandatory details about their project in a neat and organized manner.

Priority: Low

If a new project proposal cannot be created, there are many other tools such as notepad, Microsoft Word and Google Docs to use but it requires more time to organize and send.

4.1.2 Stimulus/Response Sequences

The user will click on Create New Project Proposal and will be directed to a new page where they input details.

4.1.3 Functional Requirements

- REQ-1: The system shall allow DCU students to create a new project proposal.
- REQ-2: The system shall allow DCU students to input data into the draft.
- REQ-3: When a new project proposal is made, the system shall set the proposal variable to true.

4.4 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.

4.5 Edit Project Proposal Draft

4.1.1 Description and Priority

Once a new project proposal has been created, students will then have the option of leaving the **Project Webpage** and come back to edit their draft with its saved contents.

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 click on Edit Project Proposal and will be directed to the project proposal draft page where they can continue to input or modify 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.

4.6 Send Project Proposal

4.1.1 Description and Priority

Once the student has saved their draft, they will have the option to send their project proposal to their chosen DCU professor(s).

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 professors database with the students' proposals.

4.1.3 Functional Requirements

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

4.7 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.

4.8 Approve Project

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 professors 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.

4.9 Reject Project

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.

4.10 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.

4.11 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.

4.12 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

4.13 View Demonstration Timetable

4.1.1 Description and Priority

The student can view the demonstration timetable to check the available times for demonstrating their project. The student can also view the locations available to do a demonstration of the project.

Priority: Medium

If the demonstration timetable cannot be viewed, the student must email the professors to ask for available times and having a lot of emails with different times can be a hassle to go through.

4.1.2 Stimulus/Response Sequences

The user will click on View Demonstration Timetable and be redirected to a new page, the user can select a box time frame to expand and show the available professors as well as the available locations during that time

4.1.3 Functional Requirements

REQ-1: The system shall allow DCU students view the demonstration timetable

REQ-2: When the user clicks on a time frame, the system shall expand the time frame, view all the professors and locations available.

4.14 Edit Demonstration Timetable

4.1.1 Description and Priority

The professor can add the times they are available for viewing a student project's demonstration.

Priority: High

4.1.2 Stimulus/Response Sequences

The professor will click on edit demonstration and choose to add available times or remove added times.

4.1.3 Functional Requirements

- REQ-1: The system shall allow DCU professors to edit the demonstration timetable.
- REQ-2: The system shall allow professors to add their time availability.
- REQ-3: If the professor has added a time availability, the system shall allow professors to remove that time availability.

4.15 View Status Report

4.1.1 Description and Priority

This function is for professors to view the current 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 current 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.

4.16 View Project Status of individual project

4.1.1 Description and Priority

This function is for professors to 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.

4.17 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.

4.18 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.

4.19 Add Account (Administration)

4.1.1 Description and Priority

This function is only available for the administrators of the system, the administrators will add the professors accounts into the database and provide a password for them to ensure students will not have access to the same system as the professors.

Priority: High

For security reasons, professors cannot sign up and their account must be added into the database by the administrator otherwise there will be no professors on the system.

4.1.2 Stimulus/Response Sequences

The administrator will click on Add Professor and enter a suitable password for the account, the account will be stored into the database as a professor account.

4.1.3 Functional Requirements

REQ-1: The system shall allow the administrator to add an account.

REQ-2: The system shall allow the administrator to input the account name and password.

REQ-3: The system shall store the provided account details into the database for logging in.

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 not display the user's login details to prevent unauthorized login from another user.

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 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*, and *figure7*, 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

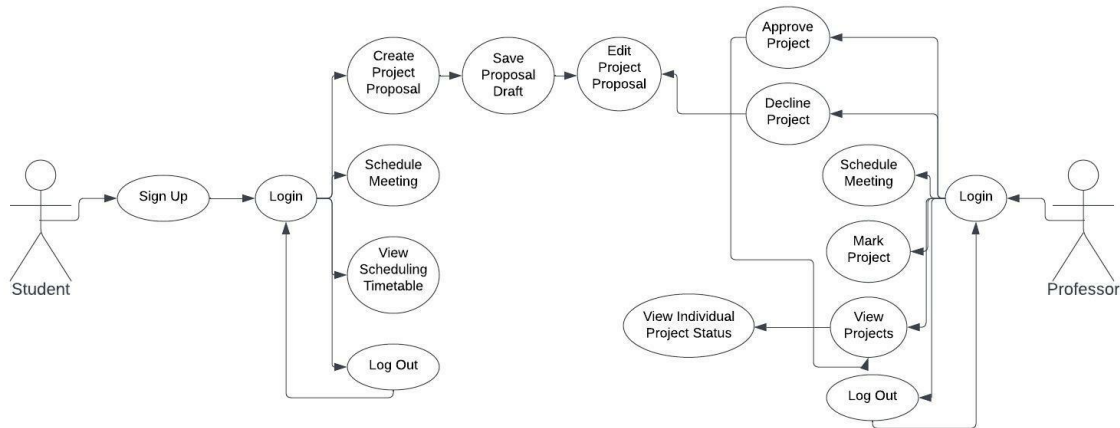


figure1: Use Case Model

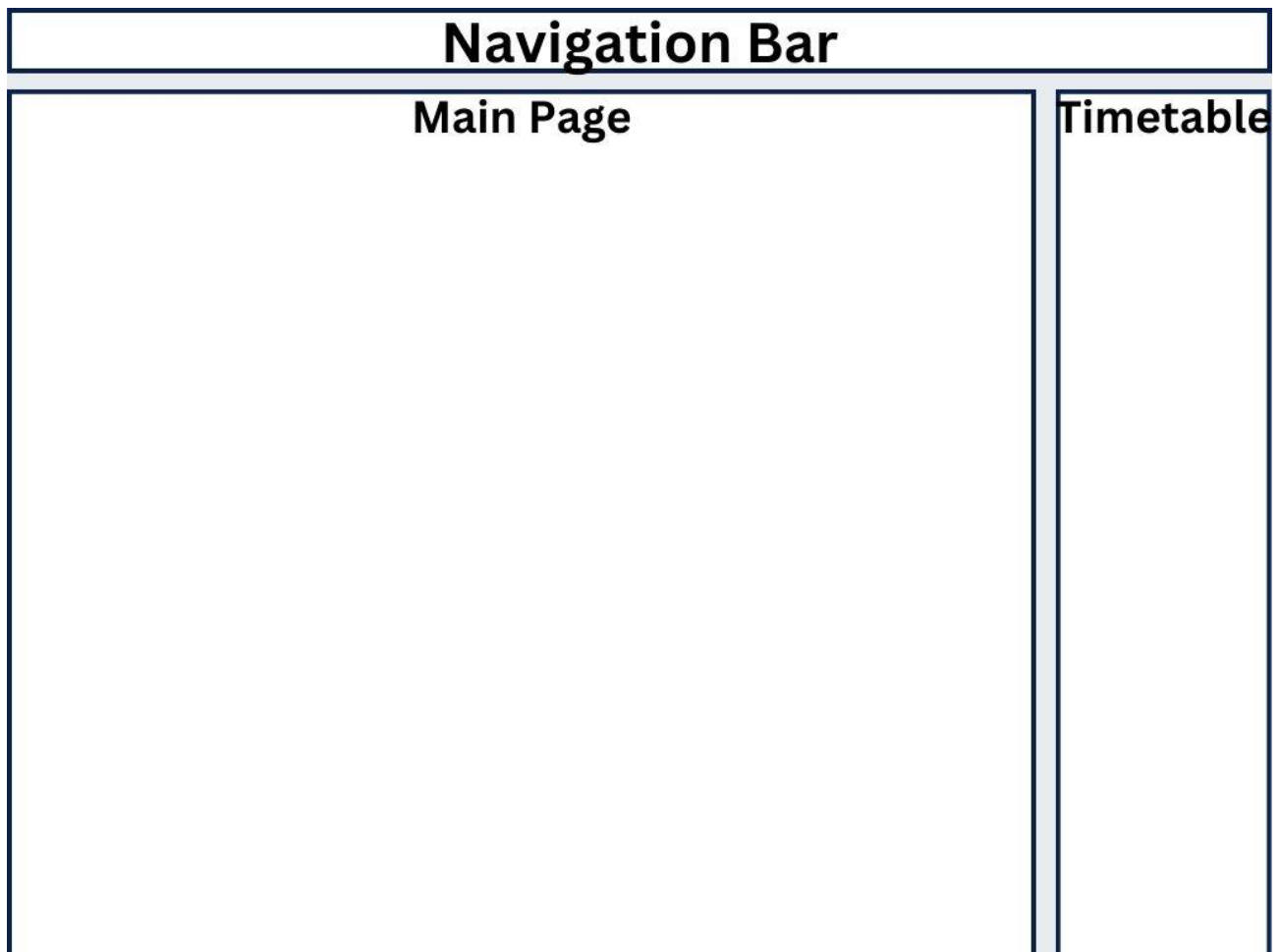


figure2: Mockup UI

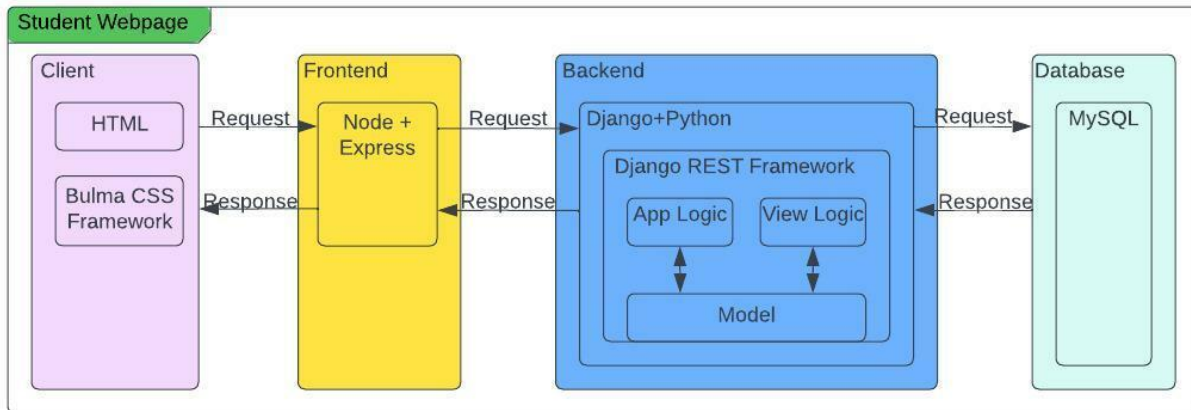


figure3: Architecture Diagram

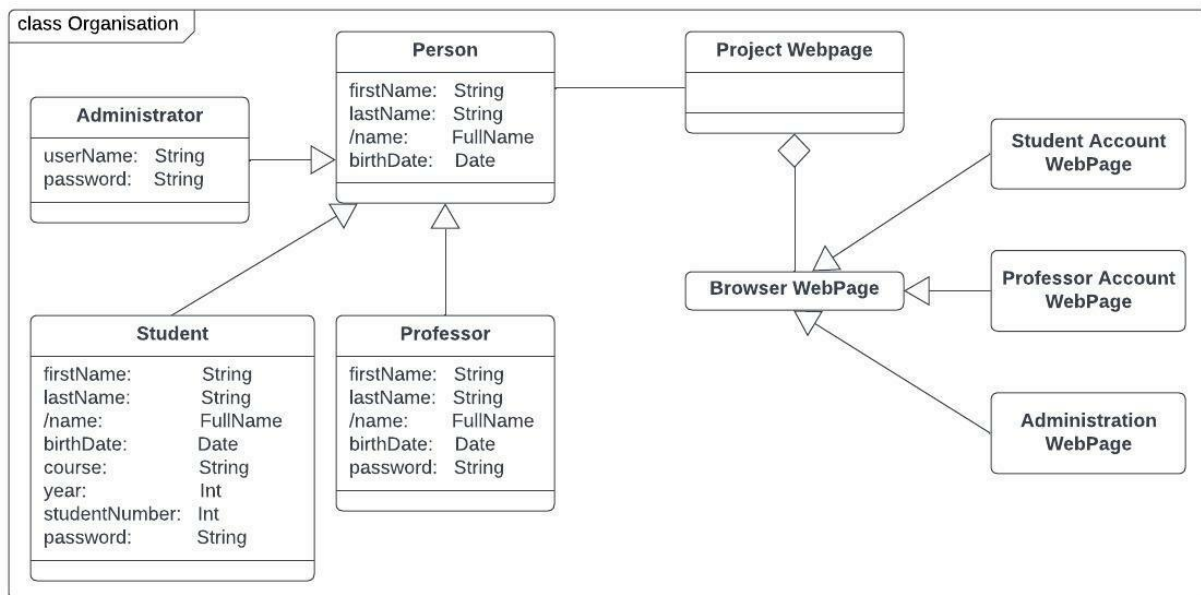


figure4: Logical Database Model

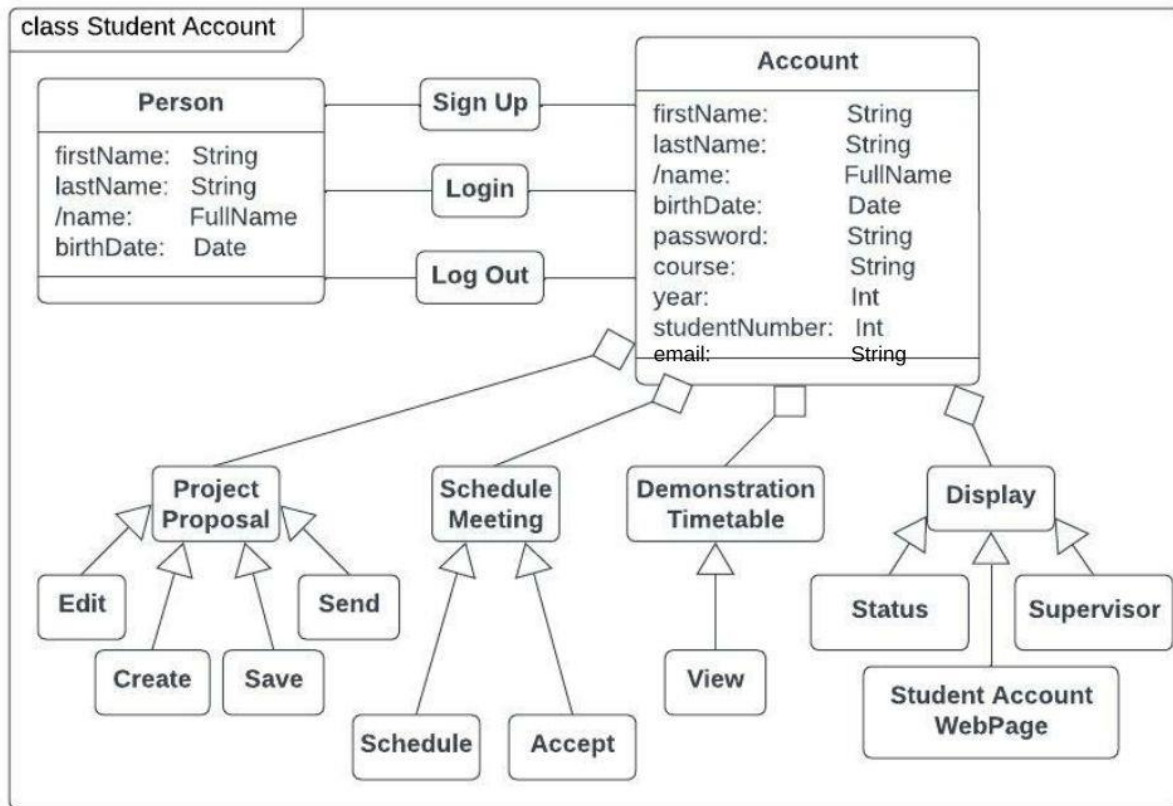


figure5: Logical Database Model

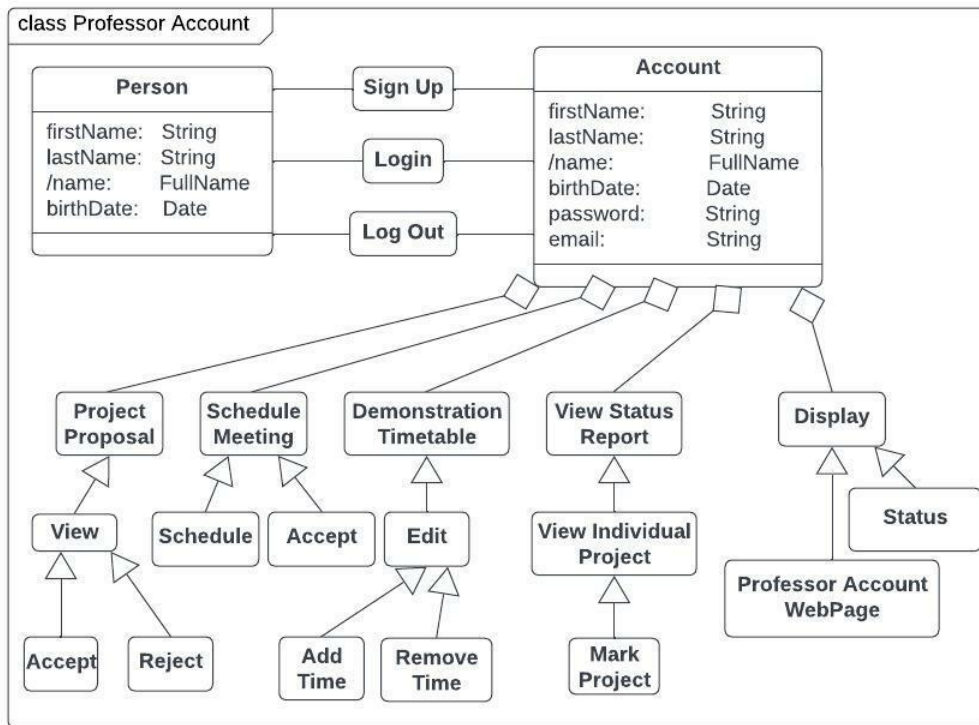


figure6: Logical Database Model

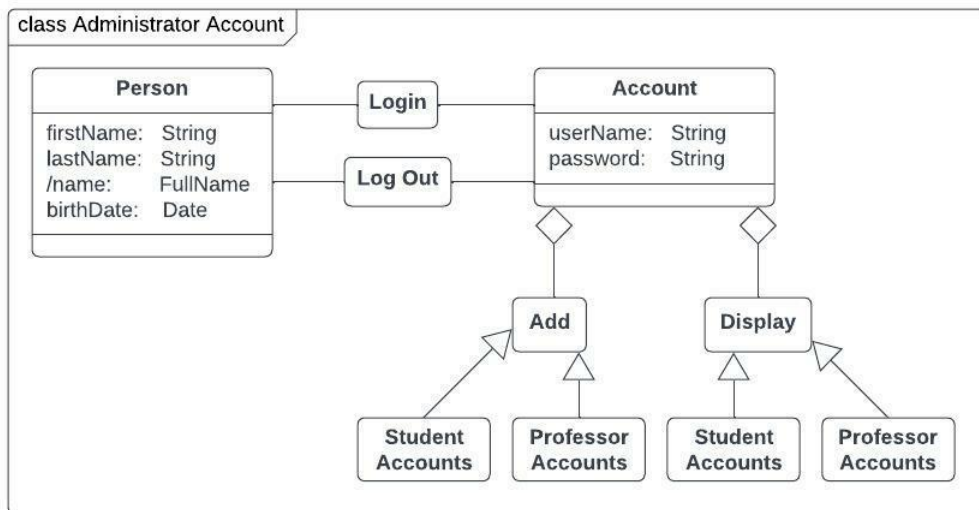


figure7: Logical Database Model

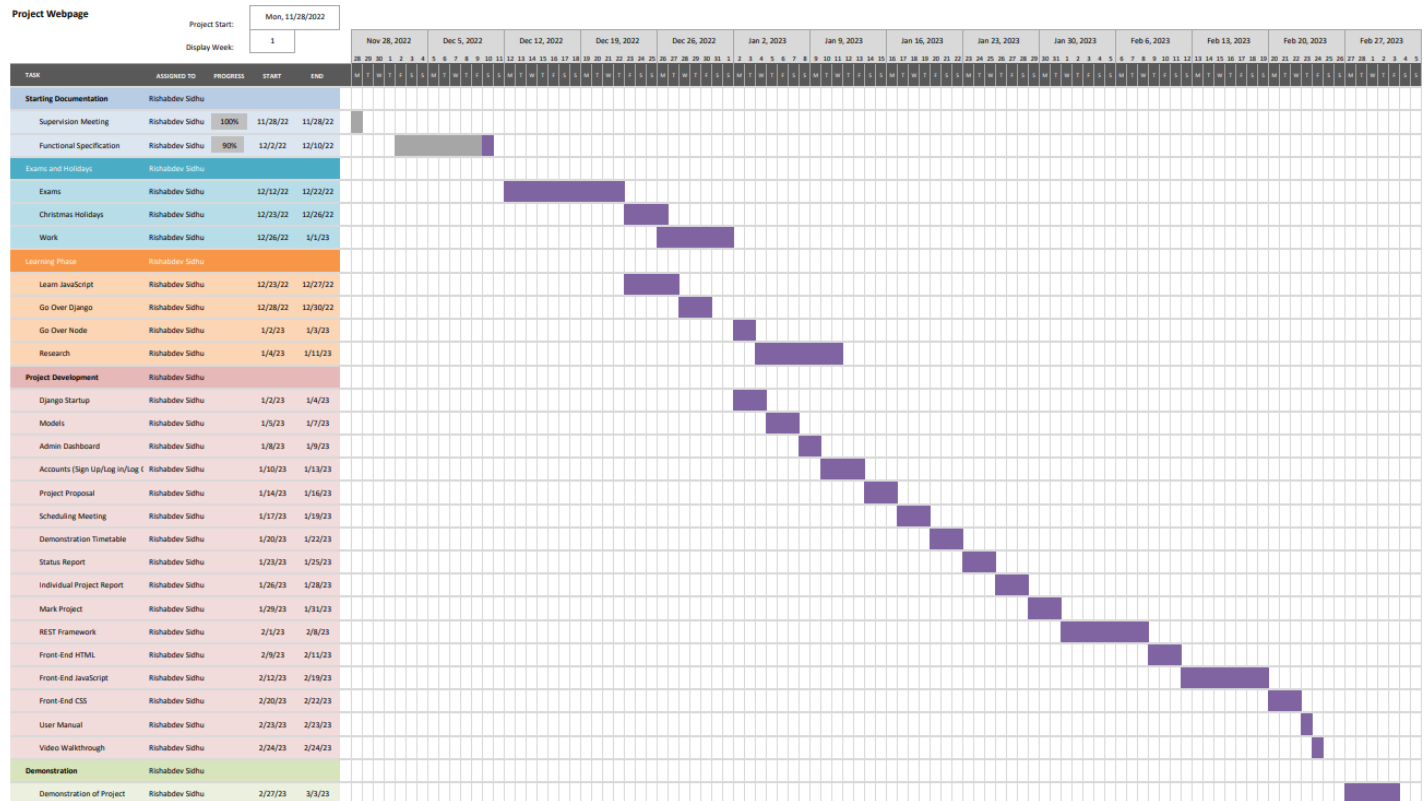


figure8: Gantt Chart

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.