



School of Computer Science & Information Technology

CSIT-22-S4-24 - Automatic Project Assignment

Project Requirement Document

Group No.: FYP-22-S4-32

Name	UOW ID	Email
Matthew Chua Yu Jie	7060750	myjc930@uowmail.edu.au
Tan Kwan Quan	7058925	kqt960@uowmail.edu.au
Tyrell Lim Yi Jin	7060579	yjtl910@uowmail.edu.au
Yang YiJie	7222075	yy690@uowmail.edu.au
Yeo Jun Ming	7430905	jmy234@uowmail.edu.au
Zheng Yinli	6959593	yz957@uowmail.edu.au

Supervisor: Mr Tian Sion Hui
Assessor: Dr Loo Poh Kok

Document Control

Title: Project Requirement Document

Document Name: APA-PRD-01

Owner	Current Version	Last Change on		Approved by
		Date	Time	
Matthew Chua Yu Jie	1.4	09/11/2022	1800	

Distribution List

Name	Title/Role	Where (location where the document is stored)
Matthew Chua Yu Jie	Project Manager, Frontend Programmer, Backend Programmer	Google Drive
Tan Kwan Quan	Backend Lead, Backend Programmer	Google Drive
Tyrell Lim Yi Jin	Backend Programmer, Documenter and Editor, Developer and Tester	Google Drive
Yang Yijie	Backend Programmer, Scrum Master	Google Drive
Yeo Jun Ming	Frontend Programmer, Test Case Lead, Documenter and Editor	Google Drive
Zheng Yinli	Frontend Lead, Frontend Programmer	Google Drive

Record of Revision

Revision Date	Description	Section Affected	Changes Made by	Version after Revision
11/10/2022	Draft of the Proposal	All	Everyone	1.0
19/10/2022	Completion of Proposal Draft with Research	All	Everyone	1.1
22/10/2022	Addition of comments provided by Supervisor	All	Everyone	1.2
30/10/2022	Adding of Use Case Descriptions and Use Case Diagrams	Project Proposal	Everyone	1.3
09/11/2022	Finalization of Project Requirement	All	Everyone	1.4

--	--	--	--	--

Table of Content

Document Control	2
Table of Content	3
1. Executive Summary	5
2. Project Background	6
2.1 Project Objective	6
2.2 Problem Description	6
2.3 Technical Research on Technologies used	8
3. Project Proposal	10
3.1 Project Vision	10
3.2 Project Goals	10
3.3 Proposed Solutions to Identified Problems	10
4. Our Product	12
4.1 Introduction	12
4.2 Objectives	12
4.3 Target Audience	12
4.4 Scope	12
4.4.1 Problems of Limitation	13
4.5 Product Diagram	14
4.6 Use Case Diagrams	15
4.7 Use Case Description	18
5. Product Features	71
6. Functional model and Description	75
6.1 Hardware Interface	76
6.2 Software Interface	76
7. Non-Functional Requirement	76
8. Project Backlog	77
9. Project Plan	79
9.1 Development Method	79
9.2 Development Tools	81

10. Risk Management Matrix	82
11. System Design	84
12. Change Management Plan	86
12.1 Roles and Responsibilities in Change Review	86
12.2 Rules and procedures to carry out request	86
13. Proposed Team Organization	88
13.1 Organization Diagram	88
13.2 Roles and Responsibilities	89
14. Project Plan	91
14.1 Project Milestones & Deliverables	91
14.2 Project Schedule	93
15. References	95
16. Appendix	95
16.1 Appendix A : Change Request Form	95
16.2 Appendix B: MongoDB resources	95
16.3 Appendix C: Express.js resources	95
16.4 Appendix D: React resources	96
16.5 Appendix E: Node.js resources	96

1. Executive Summary

Most IT Companies adopt a standard protocol of manually organizing a team of employees and assigning them an IT Project based on availability or what the Lead Project Developer has decided upon.

This leads to issues with the current process, mainly it being prone to human error . Additionally, not all employees may be adequately competent for the chosen IT Project.

By implementing our proposed plan, we will help to omit any human error during the project assignment phase and improve the quality of employees assigned to projects based on their skills and competency.

Our application will be based on the idea of solving the problem with manual allocation and will also function as a tool for collaboration within the corporate hierarchy.

While there may be tools in the market with similar functionality , our application will be tailored to be as mainstream as possible to be readily implemented in any IT settings/environment.

The proposal will outline the scope of our application , how we differ from other tools and the benefits our application will bring about in resolving the stated issues above.

2.Project Background

2.1 Project Objective

The goal is to have a single system that handles the processes related to the management of the automatic project assignments. These include:

1. A single system to keep the details of all employees, project details, project assignment details and statistics of all projects assigned based on the criteria for every round of project assignment.
2. An all in one interface with methods to add/modify/delete single/multiple employee user accounts and projects.
3. Allowing employees to indicate their preference on the projects that they wish to take up based on availability and the employee's skills and competency, where their first choice is always prioritised if possible, followed by their second, then third and so on and so forth if applicable.
4. Automatic creation of IT project groups based on the choices made by the employees.

2.2 Problem Description

Manually allocating employees to various projects involves the process of maintaining and handling numerous documentation pertaining to involved employees and projects . These documentation are stored using the traditional way of filing which may be misplaced or lost.

Information relating to these documents are also entered through Excel which is also prone to human error due to the volume of information that is managed manually.

Furthermore , employee's preferences are also manually documented and submitted which contributes to yet another chunk of data needed to be processed manually.

Making the most optimal allocation whilst considering each employee's preferences and project requirements is not only a tall order but time-consuming and laborious.

This results in an issue where there is a need to process a large amount of data without any compromise in time and manpower , hence the need for automation and digitization.

#	Problem	Description	Impact
1	Human error when manually maintaining records using a traditional filing system.	Mistakes can happen when the project admin/manager maintains records of the employees and IT project listings throughout the year as well as group assignments for each project.	<ul style="list-style-type: none"> Files are stored in an on premise system without backup. Typo or wrong listing can result in wrong allocation of personnels for IT Projects. Potential lost of past records
2	Manually allocating projects to employees, regardless of competency.	Every employee has different strengths and weaknesses, and assigning projects without taking such a criteria into consideration will result in a loss in work efficiency and work quality.	<ul style="list-style-type: none"> Requires more effort to maintain the system and more time during the allocation process. Chance of low efficiency of employees assigned to projects due to lack of competency.
3	Manual collection of employee's details	Collection of information through e-documents or physical document	<ul style="list-style-type: none"> Physical documents can be lost or misplaced E-documents collected via email, may pose privacy concerns in the event of email phishing or impersonation.
4	Traditional system of manual allocation	The system can only be managed by people who are familiar with the spreadsheet structure and there is no systematic maintenance of record.	<ul style="list-style-type: none"> The management process is not streamlined and requires time and effort to teach a new person in-charge. Requires more effort in maintaining the system as well as the allocation process.

To solve the above problems, it is proposed that all this data is handled in a simple to use application where the project administrator/manager only needs to upload project listings and their details and requirements. Employees will get to update their details and indicate their project preferences via the same application. All groupings will be done automatically using the web application, requiring much less manual effort and reducing human error.

2.3 Technical Research on Technologies used

One of the main components to be considered before developing a web page is to select the language to be used. Developing a functional website requires proficiency in programming languages such as HTML, CSS, JavaScript, Java, Databases, and Internet/Web knowledge/skills for general development. Programming languages introduce an interactive experience to the webpage through elements like visuals and audio bytes. These languages also allow the input of algorithms, scripts, and applications, contributing to a more whole and complex webpage.

Technology	Description	Application to Automatic Project Assignment
MongoDB Details: Appendix A	<p>MongoDB is an open-source, cross-platform, and distributed document-based database designed for ease of application development and scaling.</p> <p>MongoDB is not a Relational Database Management System (RDBMS). It's called a "NoSQL" database. It is opposite to SQL based databases where it does not normalize data under schemas and tables where every table has a fixed structure. Instead, it stores data in the collections as JSON based documents and does not enforce schemas. It does not have tables, rows, and columns as other SQL (RDBMS) databases.</p>	<ul style="list-style-type: none"> ● Replace the traditional way of storing data of employees and project listings they have taken in separate excel files or through pen and paper ● Allow users of the system to maintain and view all table data without the use of multiple worksheets. ● Faster and more efficient handling of information as the database is centralized ● A backup of the data can be easily made on a cloud server if needed
Express.js Details: Appendix B	<p>Express.js, or simply Express, is a back end web application framework for building RESTful APIs with Node.js, released as free and open-source software under the MIT Licence.</p>	<ul style="list-style-type: none"> ● Scripting language used for the development of backend web application frameworks that is to be layered on top of Node.js ● Replaces the functions that are used on an excel worksheet.

	It is designed for building web applications and APIs. It has been called the de facto standard server framework for Node.js.	
React Details: Appendix C	<p>React is an open source, JavaScript library for building user interfaces in web, mobile, and desktop applications. It was developed and released by Facebook back in 2013.</p> <p>Its simplicity and flexibility in building components make it one of the most popular tools for front-end development.</p>	<ul style="list-style-type: none"> ● React offers reusable components as the components are independent, reusable bits of code. This means that React will allow us to create the required interactive elements at a much faster pace.
Node.js Details: Appendix D	<p>Node.js is an open source, cross-platform runtime environment for developing server-side and networking applications. Node.js applications are written in JavaScript, and can be run within the Node.js runtime on OS X, Microsoft Windows, and Linux.</p> <p>Node.js also provides a rich library of various JavaScript modules which simplifies the development of web applications using Node.js to a great extent.</p>	<ul style="list-style-type: none"> ● Main scripting language for the development of backend functions that run on servers and is used as the framework foundation for Express.js. ● Integrates very well with React at the front end and with MongoDB for databases

3. Project Proposal

3.1 Project Vision

Our application strives to streamline the process of allocating projects to employees through automation. Fulfilling requirements such as, but not limited to, project count per candidate, candidate count per project, project skill set requirements, and more

3.2 Project Goals

- To provide a seamless and user-friendly experience to users.
- To reduce the time and effort needed to perform the assignment process.
- To reduce the effort needed to manage the data involved.
- To ensure users preferences and project requirements are met
- To ensure the most optimal allocation

3.3 Proposed Solutions to Identified Problems

#	Problem	Solution	Impact
1	Human error when manually maintaining records using a traditional filing system.	All information/records will be stored using MongoDB with the option of manual input of file imports Databases contains features which will streamline and support the management process	<ul style="list-style-type: none"> ● Improves transparency and consistency relating to data management ● Lower risk of data mishandling or lost data.
2	Manually allocating projects to employees, regardless of competency.	Developing features that will collect user preferences and be able to set project requirements . These data will then be used to consider the most optimal allocation outcome	<ul style="list-style-type: none"> ● Lower risk of human error ● Employees preferences will be met ● Projects will have higher success rate due to the correct allocation
3	Manual collection of employee's details	All information will be entered and collected within the system. Centralizing data by integrating all collected information into a shareable platform.	<ul style="list-style-type: none"> ● Time spent on menial tasks like manual documentation can be cut down ● No risk of lost documents ● Faster and more efficient handling of information as the data

			<p>is centralized</p> <ul style="list-style-type: none"> • Allow users of the system to maintain and view all table data without the use of multiple worksheets.
4	Traditional system of manual allocation	<p>All records will be stored and maintained using databases</p> <p>Design will be tailored to be intuitive and user-friendly.</p> <p>In-built algorithms will automate the allocation process and decide on the most optimal assignment.</p>	<ul style="list-style-type: none"> • Data will be centralized , allowing information to be readily available • System will be standardized and can be easily maintained without much technical knowledge. • The most optimal allocation will always be selected , reducing the risk of human error

4. Our Product

4.1 Introduction

Our product introduces an alternative through our algorithms to automate assigning projects to users based on their preferences and project specifications.

With our application, there would be no need for physical documentation of any sort ; it will be all done within our system, where all the information is readily stored and available.

Our application also introduces a more streamlined process as compared to the traditional way of manual allocation , which not only reduces the production of human error but also time spent.

4.2 Objectives

The goal is to have a single system that handles the processes related to the management of the automatic project assignments. These include:

1. A single system to keep the details of all employees, project details, project assignment details and statistics of all projects assigned based on the criteria for every round of project assignment.
2. An all in one interface with methods to add/modify/delete single/multiple employee user accounts and projects.
3. Allowing employees to indicate their preference on the projects that they wish to take up based on availability and the employee's skills and competency, where their first choice is always prioritised if possible, followed by their second, then third and so on and so forth if applicable.
4. Automatic creation of IT project groups based on the choices made by the employees.

4.3 Target Audience

- IT Companies
- Project Managers
- Project Coordinators
- Project Engineers

4.4 Scope

The team will center our attention on building a solution for our customers to improve the efficiency of the project assignment phase, and to increase the odds of completing and implementing the assigned projects successfully. In addition, we will also be looking to provide a platform for our users to share and receive project materials and other relevant information to and from the relevant parties. The team will

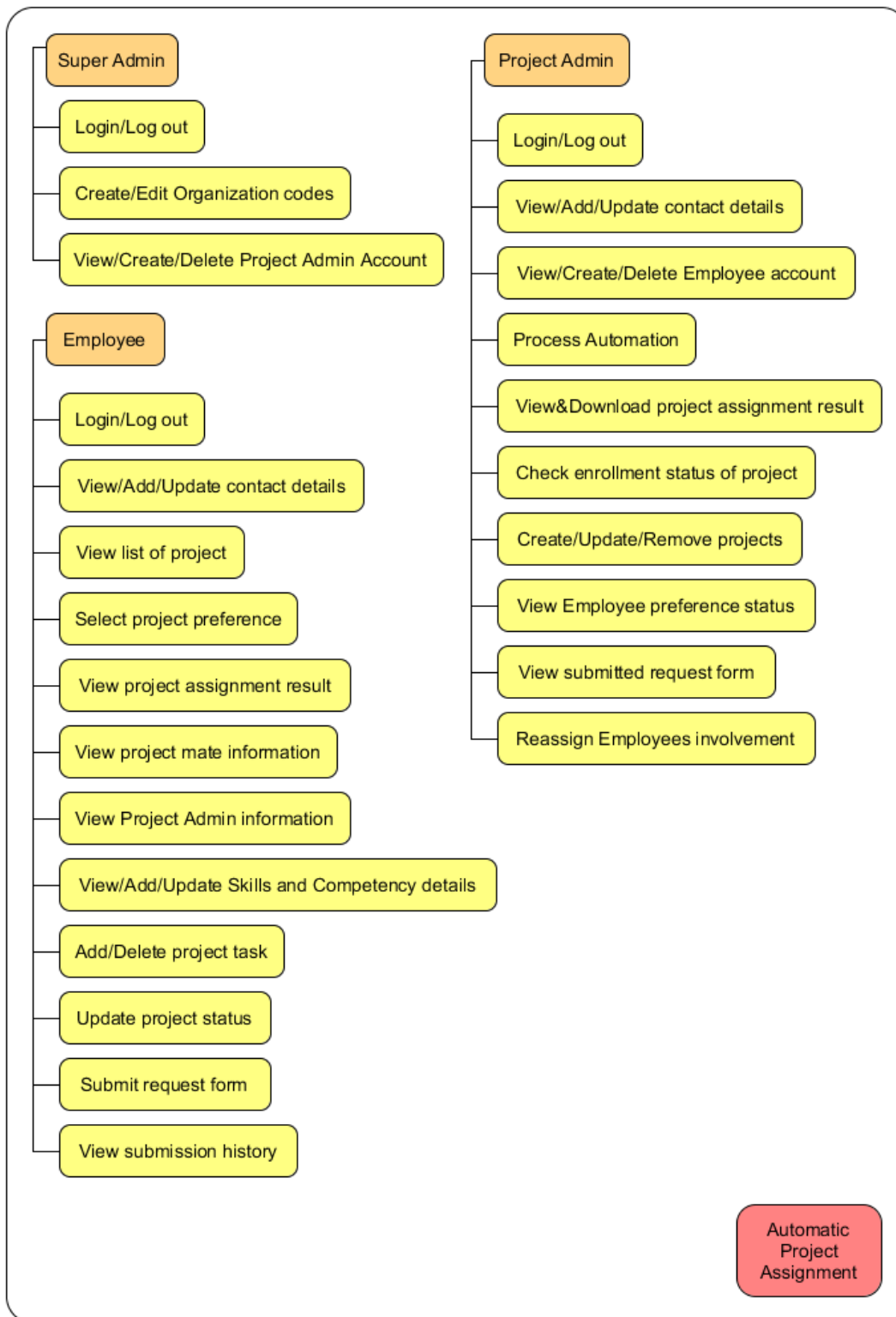
also be looking into other needs of our target users to provide them with a solution that includes other relevant components and functionalities that supplement our core service.

At this stage of the project, the team will focus on building our solution as a web application because we feel that our users will primarily be employing our product's services when they are not on the move. Our users will require an active internet connection to engage with our product.

4.4.1 Problems of Limitation

In the event that the skills of users are not diverse enough, our solution may not work as intended. The automatic project assignment system may resort to a level of randomness due to mismatching of skills between the projects and the users. Hence, we will be exploring different mechanisms and controls to bring down the level of randomness, if not eliminate it completely, when such a situation arises.

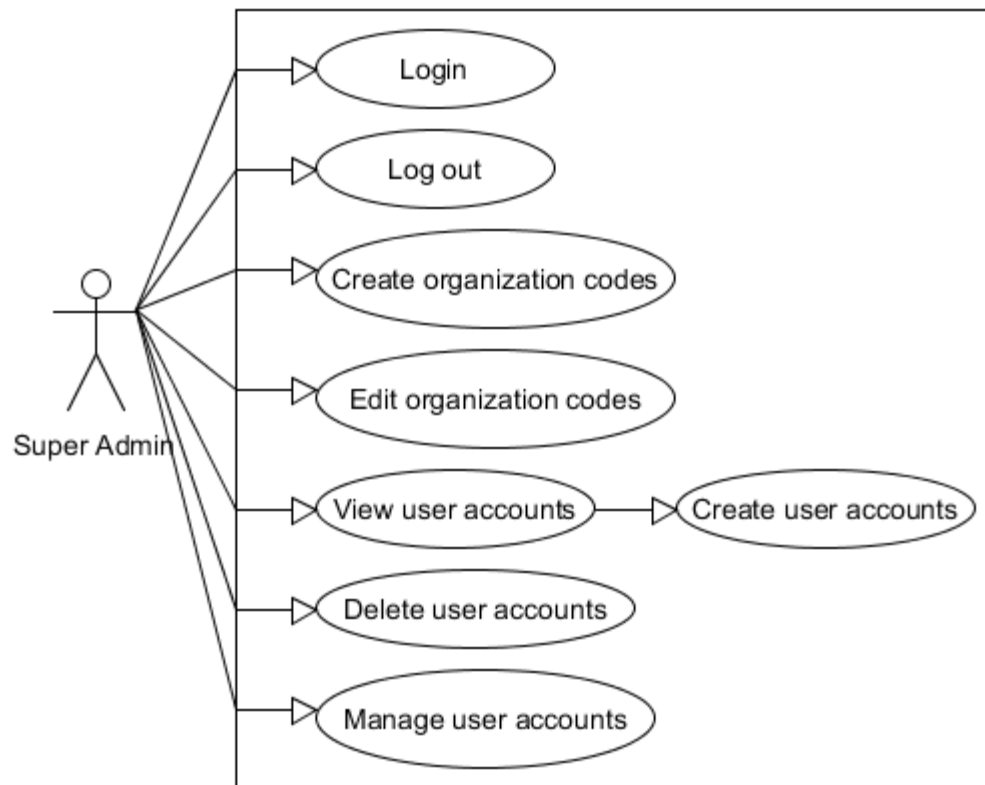
4.5 Product Diagram



4.6 Use Case Diagrams







4.7 Use Case Description

Use case: Login as Project Admin	Taiga ID: #3
Stakeholders and goals: Project Admin - Login	
Description: The Project Admin login to the system	
Actors: Project Admin	
Pre-condition: User is Admin and has not logged into the system	
Post-condition: User is logged into the system as Admin	
Trigger: The Project Admin wants to login to the system	
Normal flow: <ol style="list-style-type: none"> 1. The user visits the website 2. The user enters their Username and Password 3. The user clicks on the login button 4. The system authenticates the information provided 5. The system prompts the user that the login attempt was successful 6. The system brings the user back to the homepage 	
Sub-flows: None	
Alternative/Exceptional flows: <p>5.a The system fails to authenticate the information provided, rejects the login attempt then prompts the user that the login attempt failed</p>	

Use case: Log out as Project Admin	Taiga ID: #4
Stakeholders and goals: Project Admin - Log out	
Description: The Project Admin logs out of the system	
Actors: Project Admin	
Pre-condition: User is Admin and has logged into the system as Admin	
Post-condition: User is logged out the system	
Trigger: The Project Admin wants to log out of the system	
Normal flow: <ol style="list-style-type: none"> 1. The user clicks on the log out button 2. The system verifies the log out request 3. The system prompts the user that the log out attempt is successful 4. The system brings the user back to the homepage 	
Sub-flows: None	
Alternative/Exceptional flows: None	

Use case: Add contact details as Project Admin	Taiga ID: #5
Stakeholders and goals: Project Admin - Add contact details	
Description: The Project Admin adds their contact information into the system	
Actors: Project Admin	
Pre-condition: User is logged in as Admin user	
Post-condition: Admin user added their contact information	
Trigger: The Project Admin wants to add their contact information into the system	
<p>Normal flow:</p> <ol style="list-style-type: none"> 1. The user clicks on their profile icon 2. The system brings the user to their profile page 3. The user clicks on edit profile details 4. The system brings the user to the edit profile page 5. The user adds in their contact information and clicks the save button 6. The system verifies the request from the user 7. The system saves the information and prompts the user that the save attempt was successful 8. The system brings the user back to their profile page 	
Sub-flows: None	
<p>Alternative/Exceptional flows:</p> <p>6.a The system fails to verify the request from the user, information was not saved and the system prompts the user that the save attempt was a failure</p>	

Use case: View account information as Project Admin	Taiga ID: #6
Stakeholders and goals: Project Admin - View account information	
Description: The Project Admin views their account information	
Actors: Project Admin	
Pre-condition: User is Admin and has logged into the system as Admin.	
Post-condition: User views their account information	
Trigger: The Project Admin wants to view their contact information	
Normal flow: <ol style="list-style-type: none"> 1. The user searches for the account's information by typing their email on the search bar and clicking search. 2. The system brings the user to their profile page. 3. The user views their account information 	
Sub-flows: None	
Alternative/Exceptional flows: None	

Use case: Update account information as Project Admin	Taiga ID: #7
Stakeholders and goals: Project Admin - Update account information	
Description: The Project Admin updates their account information	
Actors: Project Admin	
Pre-condition: User is Admin and has logged into the system as Admin.	
Post-condition: User updated their account information	
Trigger: The Project Admin wants to update their contact information	
<p>Normal flow:</p> <ol style="list-style-type: none"> 1. The user searches for the account's information by typing their email on the search bar and clicking search. 2. The system brings the user to their profile page 3. The user clicks on edit profile details 4. The system brings the user to the edit profile page 5. The user updates their information and clicks on the save button 6. The system verifies the request from the user 7. The system saves the information and prompts the user that the save attempt was successful 8. The system brings the user back to their profile page 	
Sub-flows: None	
<p>Alternative/Exceptional flows:</p> <p>6.a The system fails to verify the request from the user, information was not saved and the system prompts the user that the save attempt was a failure</p>	

Use case: Create new projects as Project Admin	Taiga ID: #8
Stakeholders and goals: Project Admin - Create new projects	
Description: The Project Admin create new projects	
Actors: Project Admin	
Pre-condition: User is Admin and is logged into the system	
Post-condition: User created new projects	
Trigger: The Project Admin wants to create new projects	
<p>Normal flow:</p> <ol style="list-style-type: none"> 1. The user clicks on Project Listing 2. The system bring the user to the Project Listing page 3. The user clicks on the add new project button 4. The system bring the user to add project page 5. The user enters the new project informations & project details clicks on the add button 6. The system verifies the request from the user 7. The system saves the information and prompts the user that the save attempt was successful 8. The system brings the user back to the Project Listing page 	
Sub-flows: None	
<p>Alternative/Exceptional flows:</p> <p>6.a The system fails to verify the request from the user, information was not saved and the system prompts the user that the save attempt was a failure</p>	

Use case: Update project details as Project Admin	Taiga ID: #10
Stakeholders and goals: Project Admin - Update project details	
Description: The Project Admin updates project details	
Actors: Project Admin	
Pre-condition: User is logged in as Admin and the chosen project exists	
Post-condition: User updates project details	
Trigger: The Project Admin wants to update project details	
<p>Normal flow:</p> <ol style="list-style-type: none"> 1. The user clicks on Project Listing 2. The system bring the user to the Project Listing page 3. The user clicks on the edit button of the selected project 4. The system brings the user to the edit project page 5. The user updates the project detail information and clicks the save button 6. The system verifies the request from the user 7. The system saves the information and prompts the user that the save attempt was successful 8. The system brings the user back to the Project Listing page 	
Sub-flows: None	
<p>Alternative/Exceptional flows:</p> <p>6.a The system fails to verify the request from the user, information was not saved and the system prompts the user that the save attempt was a failure</p>	

Use case: Setting project threshold as Project Admin	Taiga ID: #11
Stakeholders and goals: Project Admin - set project threshold	
Description: The Project Admin sets project threshold	
Actors: Project Admin	
Pre-condition: User is logged in as Admin	
Post-condition: User sets a threshold for the process	
Trigger: The Project Admin wants to set a threshold for the process	
Normal flow: <ol style="list-style-type: none"> 1. The user click on the automatic assign button 2. The system brings the user to the project group generation page 3. The user enters the relevant threshold information 	
Sub-flows: None	
Alternative/Exceptional flows: None	

Use case: Setting number of employee required as Project Admin	Taiga ID: #12
Stakeholders and goals: Project Admin - set number of employee required for the project	
Description: The Project Admin sets number of employee required for the project	
Actors: Project Admin	
Pre-condition: User is logged in as Admin and the chosen project exists	
Post-condition: User sets the number of employee required for the chosen project	
Trigger: The Project Admin wants to set a number of employee for the chosen project	
<p>Normal flow:</p> <ol style="list-style-type: none"> 1. The user clicks on Project Listing 2. The system bring the user to the Project Listing page 3. The user clicks on the edit button of the selected project 4. The system brings the user to the edit project page 5. The user enters a number under the number of employee required section and clicks the save button 6. The system verifies the request from the user 7. The system saves the information and prompts the user that the save attempt was successful 8. The system brings the user back to the Project Listing page 	
Sub-flows: None	
<p>Alternative/Exceptional flows:</p> <p>6.a The system fails to verify the request from the user, information was not saved and the system prompts the user that the save attempt was a failure</p>	

Use case: Check enrollment status of project as Project Admin	Taiga ID: #13
Stakeholders and goals: Project Admin - Check enrollment status of project	
Description: The Project Admin checks the enrollment status project	
Actors: Project Admin	
Pre-condition: User is logged in as Admin and the chosen project exists	
Post-condition: User views the enrollment status of project	
Trigger: The Project Admin wants to view the enrollment status of project	
Normal flow: <ol style="list-style-type: none"> 1. The user clicks on Project Listing 2. The system bring the user to the Project Listing page 3. The user views the enrollment status of the projects 	
Sub-flows: None	
Alternative/Exceptional flows: None	

Use case: Remove project as Project Admin	Taiga ID: #14
Stakeholders and goals: Project Admin - Remove project	
Description: The Project Admin removes project	
Actors: Project Admin	
Pre-condition: User is logged in as Admin and the chosen project exists	
Post-condition: User removes the chosen project	
Trigger: The Project Admin wants to remove the chosen project	
<p>Normal flow:</p> <ol style="list-style-type: none"> 1. The user clicks on Project Listing 2. The system bring the user to the Project Listing page 3. The user clicks on the edit button of the selected project 4. The system brings the user to the edit project page 5. The user clicks on the delete button 6. The system verifies the request from the user 7. The system prompts the user for additional confirmation 8. The user click on the confirm delete button 9. The system verifies the request from the user 10. The system deletes the project and prompts the user that the delete process is successful 11. The system brings the user back the the Project Listing page 	
Sub-flows: None	
<p>Alternative/Exceptional flows:</p> <p>6.a The system fails to verify the request from the user, prompts the user that an error has occurred and brings the user back to the Project Listing page</p> <p>8.a The user clicks on the no button, the system remains at the edit project page and awaits user input</p>	

Use case: View employee information as Project Admin	Taiga ID: #15
Stakeholders and goals: Project Admin - View employee information	
Description: The Project Admin views the information of employees	
Actors: Project Admin	
Pre-condition: User is logged in as Admin and the chosen employee exists	
Post-condition: User views the information of the chosen employee	
Trigger: The Project Admin wants to view information of the chosen employee	
Normal flow: <ol style="list-style-type: none"> 1. The user clicks on Employee Listing 2. The system brings the user to the Employee Listing page 3. The user clicks on the more details button of the chosen employee 4. The system bring the user to the profile page of the chosen employee 5. The user views the information available 	
Sub-flows: None	
Alternative/Exceptional flows: None	

Use case: Process the automation process as Project Admin	Taiga ID: #16
Stakeholders and goals: Project Admin - Process the automation process	
Description: The Project Admin starts the automatic assignment process	
Actors: Project Admin	
Pre-condition: User is logged in as Admin	
Post-condition: The automatic assignment process starts	
Trigger: The Project Admin wants to start the automatic assignment process	
<p>Normal flow:</p> <ol style="list-style-type: none"> 1. The user click on the automatic assign button 2. The system brings the user to the project group generation page 3. The user enters the relevant threshold information 4. The user clicks on the start button 5. The system prompts the user for additional confirmation 6. The user clicks on the yes auto assign button 7. The system verifies the request from the user 8. The system runs the automatic project assignment process 9. The system prompts the user that the process is successful 10. The system brings the user to the result page 	
Sub-flows: None	
<p>Alternative/Exceptional flows:</p> <p>6.a The user clicks on the no button, the system remains at the homepage and awaits user input</p> <p>7.a The system fails to verify the request from the user, prompts the user that an error has occurred and brings the user back to the homepage</p> <p>8.a The system fails to run the automatic project assignment process, prompts the user about the failure and brings the user back to the homepage</p>	

Use case: View assignment result as Project Admin	Taiga ID: #17
Stakeholders and goals: Project Admin - View assignment result	
Description: The Project Admin wants to view the result of the automatic assignment process	
Actors: Project Admin	
Pre-condition: User is logged in as Admin and the automated assignment process is successful	
Post-condition: User views the result of the automated assignment process	
Trigger: The Project Admin runs the automated assignment process	
Normal flow: <ol style="list-style-type: none"> 1. The system brings the user to the result page 2. The user views the result of the automated assignment process 	
Sub-flows: None	
Alternative/Exceptional flows: None	

Use case: View preference result as Project Admin	Taiga ID: #18
Stakeholders and goals: Project Admin - View preference result	
Description: The Project Admin wants to view how many people enrolled to their preference result	
Actors: Project Admin	
Pre-condition: User is logged in as Admin and the automated assignment process is successful	
Post-condition: User views the preference result	
Trigger: The Project Admin runs the automated assignment process	
Normal flow: <ol style="list-style-type: none"> 1. The system brings the user to the result page 2. The user views the preference result 	
Sub-flows: None	
Alternative/Exceptional flows: None	

Use case: View total number of unassigned project as Project Admin	Taiga ID: #19
Stakeholders and goals: Project Admin - View total number of unassigned project	
Description: The Project Admin wants to view total number of unassigned project	
Actors: Project Admin	
Pre-condition: User is logged in as Admin and the automated assignment process is successful	
Post-condition: User views the total number of unassigned project	
Trigger: The Project Admin runs the automated assignment process	
Normal flow: <ol style="list-style-type: none"> 1. The system brings the user to the result page 2. The user views total number of unassigned project 	
Sub-flows: None	
Alternative/Exceptional flows: None	

Use case: Generate a rate of accuracy report as Project Admin	Taiga ID: #20
Stakeholders and goals: Project Admin - Generate a rate of accuracy report	
Description: The Project Admin generates a rate of accuracy report and view it	
Actors: Project Admin	
Pre-condition: User is logged in as Admin and the automated assignment process is successful	
Post-condition: User generates a rate of accuracy report	
Trigger: The Project Admin wants to generate a rate of accuracy report	
Normal flow: <ol style="list-style-type: none"> 1. The system brings the user to the result page 2. The user clicks on the generate rate of accuracy report button 3. The system verifies the request from the user 4. The system generates a rate of accuracy report and shows the user 	
Sub-flows: None	
Alternative/Exceptional flows: <ol style="list-style-type: none"> 3.a The system fails to generate a rate of accuracy report and prompt the user that an error has occurred. The system remains at the result page, awaits user input 	

Use case: Login as Employee	Taiga ID: #21
Stakeholders and goals: Employee - Login	
Description: The Employee login to the system	
Actors: Employee	
Pre-condition: User is Employee and has not logged into the system	
Post-condition: User is logged into the system as Employee	
Trigger: The Employee wants to login to the system	
Normal flow: <ol style="list-style-type: none"> 1. The user visits the website 2. The user enters their Username and Password 3. The user clicks on the login button 4. The system authenticates the information provided 5. The system prompts the user that the login attempt was successful 6. The system brings the user back to the homepage 	
Sub-flows: None	
Alternative/Exceptional flows: <ol style="list-style-type: none"> 5.a The system fails to authenticate the information provided, rejects the login attempt then prompts the user that the login attempt failed 	

Use case: Log out as Employee	Taiga ID: #22
Stakeholders and goals: Employee - Log out	
Description: The Employee logs out of the system	
Actors: Employee	
Pre-condition: User is Employee and has logged into the system as Employee.	
Post-condition: User is logged out of the system.	
Trigger: The Employee wants to log out of the system	
Normal flow: <ol style="list-style-type: none"> 1. The user clicks on the log out button 2. The system verifies the log out request 3. The system prompts the user that the log out attempt is successful 4. The system brings the user back to the homepage 	
Sub-flows: None	
Alternative/Exceptional flows: None	

Use case: Add contact details as Employee	Taiga ID: #23
Stakeholders and goals: Employee - Add contact details	
Description: The Employee adds their contact information into the system	
Actors: Employee	
Pre-condition: User is logged in as Employee	
Post-condition: Employee added their contact information	
Trigger: The Employee wants to add their contact information into the system	
<p>Normal flow:</p> <ol style="list-style-type: none"> 1. The user clicks on their profile icon 2. The system brings the user to their profile page 3. The user clicks on edit profile details 4. The system brings the user to the edit profile page 5. The user adds in their contact information and clicks the save button 6. The system verifies the request from the user 7. The system saves the information and prompts the user that the save attempt was successful 8. The system brings the user back to their profile page 	
Sub-flows: None	
<p>Alternative/Exceptional flows:</p> <p>6.a The system fails to verify the request from the user, information was not saved and the system prompts the user that the save attempt was a failure</p>	

Use case: View account information as Employee	Taiga ID: #24
Stakeholders and goals: Employee - View account information	
Description: The Employee views their account information	
Actors: Employee	
Pre-condition: User is Employee and has logged into the system as Employee	
Post-condition: User views their account information	
Trigger: The Employee wants to view their contact information	
Normal flow: <ol style="list-style-type: none"> 1. The user clicks on their profile icon 2. The system brings the user to their profile page 3. The user views their account information 	
Sub-flows: None	
Alternative/Exceptional flows: None	

Use case: Update account information as Employee	Taiga ID: #25
Stakeholders and goals: Employee - Update account information	
Description: The Employee updates their account information	
Actors: Employee	
Pre-condition: User is Employee and has logged into the system as Employee	
Post-condition: User updated their account information	
Trigger: The Employee wants to update their contact information	
<p>Normal flow:</p> <ol style="list-style-type: none"> 1. The user clicks on their profile icon 2. The system brings the user to their profile page 3. The user clicks on edit profile details 4. The system brings the user to the edit profile page 5. The user updates their information and clicks on the save button 6. The system verifies the request from the user 7. The system saves the information and prompts the user that the save attempt was successful 8. The system brings the user back to their profile page 	
Sub-flows: None	
<p>Alternative/Exceptional flows:</p> <p>6.a The system fails to verify the request from the user, information was not saved and the system prompts the user that the save attempt was a failure</p>	

Use case: Add skills and competency details as Employee	Taiga ID: #26
Stakeholders and goals: Employee - Add skills and competency details	
Description: The Employee add their skills and competency details into the system	
Actors: Employee	
Pre-condition: User is Employee and has logged into the system as Employee	
Post-condition: User added their skills and competency details into the system	
Trigger: The Employee wants to add their skills and competency details	
<p>Normal flow:</p> <ol style="list-style-type: none"> 1. The user clicks on their profile icon 2. The system brings the user to their profile page 3. The user clicks on edit profile details 4. The system brings the user to the edit profile page 5. The user adds their skills and competency details and clicks on the save button 6. The system verifies the request from the user 7. The system saves the information and prompts the user that the save attempt was successful 8. The system brings the user back to their profile page 	
Sub-flows: None	
<p>Alternative/Exceptional flows:</p> <p>6.a The system fails to verify the request from the user, information was not saved and the system prompts the user that the save attempt was a failure</p>	

Use case: View skills and competency details as Employee	Taiga ID: #27
Stakeholders and goals: Employee - View skills and competency details	
Description: The Employee views their skills and competency details	
Actors: Employee	
Pre-condition: User is Employee and has logged into the system as Employee	
Post-condition: User views their skills and competency details	
Trigger: The Employee wants to view their skills and competency details	
Normal flow: <ol style="list-style-type: none"> 1. The user clicks on their profile icon 2. The system brings the user to their profile page 3. The user views their skills and competency details 	
Sub-flows: None	
Alternative/Exceptional flows: None	

Use case: Update skills and competency details as Employee	Taiga ID: #28
Stakeholders and goals: Employee - Update skills and competency details	
Description: The Employee updates their skills and competency details	
Actors: Employee	
Pre-condition: User is Employee and has logged into the system as Employee	
Post-condition: User updated their skills and competency details	
Trigger: The Employee wants to update their skills and competency details	
<p>Normal flow:</p> <ol style="list-style-type: none"> 1. The user clicks on their profile icon 2. The system brings the user to their profile page 3. The user clicks on edit profile details 4. The system brings the user to the edit profile page 5. The user updates their skills and competency details and clicks on the save button 6. The system verifies the request from the user 7. The system saves the information and prompts the user that the save attempt was successful 8. The system brings the user back to their profile page 	
Sub-flows: None	
<p>Alternative/Exceptional flows:</p> <p>6.a The system fails to verify the request from the user, information was not saved and the system prompts the user that the save attempt was a failure</p>	

Use case: View project listing as Employee	Taiga ID: #29
Stakeholders and goals: Employee - View project listing	
Description: The Employee views project listings	
Actors: Employee	
Pre-condition: User is logged in as Employee and project exists	
Post-condition: User views the project listings	
Trigger: The Employee wants to view project listings	
Normal flow: <ol style="list-style-type: none"> 1. The user clicks on Project Listing 2. The system bring the user to the Project Listing page 3. The user views the list of available project 	
Sub-flows: None	
Alternative/Exceptional flows: None	

Use case: View project details as Employee	Taiga ID: #30
Stakeholders and goals: Employee - View project details	
Description: The Employee views project details	
Actors: Employee	
Pre-condition: User is logged in as Employee and project details exists	
Post-condition: User views the project details	
Trigger: The Employee wants to view project details	
Normal flow: <ol style="list-style-type: none"> 1. The user clicks on Project Listing 2. The system bring the user to the Project Listing page 3. The user views the project details 	
Sub-flows: None	
Alternative/Exceptional flows: None	

Use case: Input project preference as Employee	Taiga ID: #31
Stakeholders and goals: Employee - Input project preference	
Description: The Employee inputs their project preference	
Actors: Employee	
Pre-condition: User is logged in as Employee and project exists	
Post-condition: User inputted their project preference	
Trigger: The Employee wants to input their project preference	
Normal flow: <ol style="list-style-type: none"> 1. The user clicks on Project Listing 2. The system bring the user to the Project Listing page 3. The user clicks on submit preference 4. The system bring the user to the project preference page 5. The user selects their first, second, and third preference 	
Sub-flows: None	
Alternative/Exceptional flows: None	

Use case: Save and submit project preference as Employee	Taiga ID: #32
Stakeholders and goals: Employee - Save and submit project preference	
Description: The Employee saves and submit their project preference	
Actors: Employee	
Pre-condition: User is logged in as Employee and project preference has been inputted	
Post-condition: User saves and submit their project preference	
Trigger: The Employee wants to saves and submit their project preference	
<p>Normal flow:</p> <ol style="list-style-type: none"> 1. The user clicks the save & submit button 2. The system prompts the user for additional confirmation 3. The user clicks on the confirm submit button 4. The system verifies the request from the user 5. The system prompts the user that the submission was successful and their preferences has been submitted, then brings the user back to the homepage 	
Sub-flows: None	
<p>Alternative/Exceptional flows:</p> <p>3.a The user clicks on the cancel button, the system remains at the project preference page and awaits user input</p> <p>4.a The system fails to verify the users request and prompts the user that the submission was a failure, the system remains at the project preference page and awaits user input</p>	

Use case: View assignment result as Employee	Taiga ID: #33
Stakeholders and goals: Employee - View assignment result	
Description: The Employee wants to view the result of the automatic assignment process	
Actors: Employee	
Pre-condition: User is logged in as Employee and the automated assignment process is successful	
Post-condition: User views the result of the automated assignment process	
Trigger: The Employee wants to view the result of the automated assignment process	
Normal flow: <ol style="list-style-type: none"> 1. The user clicks on My projects 2. The system brings the user to their project page 3. The user views the result of the assigned project 	
Sub-flows: None	
Alternative/Exceptional flows: None	

Use case: View skills and competency details of project mates as Employee	Taiga ID: #34
Stakeholders and goals: Employee - View skills and competency details of project mates	
Description: The Employee wants to view the skills and competency details of their project mates	
Actors: Employee	
Pre-condition: User is logged in as Employee and the automated assignment process is successful	
Post-condition: User views the skills and competency details of their project mates	
Trigger: The Employee wants to view the skills and competency details of their project mates	
Normal flow: <ol style="list-style-type: none"> 1. The user clicks on My projects 2. The system brings the user to their project page 3. The user clicks on the chosen project mates icon 4. The system brings the user to the chosen project mates profile page 5. The user views the chosen project mates skills and competency details 	
Sub-flows: None	
Alternative/Exceptional flows: None	

Use case: View contact details of project mates as Employee	Taiga ID: #36
Stakeholders and goals: Employee - View contact details of project mates	
Description: The Employee wants to view the contact details of their project mates	
Actors: Employee	
Pre-condition: User is logged in as Employee and the automated assignment process is successful	
Post-condition: User views the contact details of their project mates	
Trigger: The Employee wants to view the contact details of their project mates	
Normal flow: <ol style="list-style-type: none"> 1. The user clicks on My projects 2. The system brings the user to their project page 3. The user clicks on the chosen project mates icon 4. The system brings the user to the chosen project mates profile page 5. The user views the chosen project mates contact details 	
Sub-flows: None	
Alternative/Exceptional flows: None	

Use case: View contact details of project admin as Employee	Taiga ID: #37
Stakeholders and goals: Employee - View contact details of project admin	
Description: The Employee wants to view the contact details of their project admin	
Actors: Employee	
Pre-condition: User is logged in as Employee and the automated assignment process is successful	
Post-condition: User views the contact details of their project admin	
Trigger: The Employee wants to view the contact details of their project admin	
Normal flow: <ol style="list-style-type: none"> 1. The user clicks on My projects 2. The system brings the user to their project page 3. The user clicks on the project admin icon 4. The system brings the user to the chosen project admin profile page 5. The user views the chosen project admin contact details 	
Sub-flows: None	
Alternative/Exceptional flows: None	

Use case: Login as Super Admin	Taiga ID: #38
Stakeholders and goals: Super Admin - Login	
Description: The Super Admin login to the system	
Actors: Super Admin	
Pre-condition: User is Super Admin and has not logged into the system	
Post-condition: User is logged into the system as Super Admin	
Trigger: The Super Admin wants to login to the system	
Normal flow: <ol style="list-style-type: none"> 1. The user visits the website 2. The user enters their Username and Password 3. The user clicks on the login button 4. The system authenticates the information provided 5. The system prompts the user that the login attempt was successful 6. The system brings the user back to the homepage 	
Sub-flows: None	
Alternative/Exceptional flows: <ol style="list-style-type: none"> 5.a The system fails to authenticate the information provided, rejects the login attempt then prompts the user that the login attempt failed 	

Use case: Log out as Super Admin	Taiga ID: #39
Stakeholders and goals: Super Admin - Log out	
Description: The Super Admin logs out of the system	
Actors: Super Admin	
Pre-condition: User is Super Admin and has logged into the system as Super Admin	
Post-condition: User is logged out the system	
Trigger: The Super Admin wants to log out of the system	
Normal flow: <ol style="list-style-type: none"> 1. The user clicks on the log out button 2. The system verifies the log out request 3. The system prompts the user that the log out attempt is successful 4. The system brings the user back to the homepage 	
Sub-flows: None	
Alternative/Exceptional flows: None	

Use case: Create Organization codes as Super Admin	Taiga ID: #40
Stakeholders and goals: Super Admin - Create Organization codes	
Description: The Super Admin creates Organization codes	
Actors: Super Admin	
Pre-condition: User is logged in as Super Admin	
Post-condition: User has created the Organization code	
Trigger: The Super Admin wants to create Organization codes	
<p>Normal flow:</p> <ol style="list-style-type: none"> 1. The user clicks on Organization Listing 2. The system brings the user to the Organization Listing page 3. The user clicks on add new Organization code 4. The system brings the user to add organization code page 5. The user inputs the information required and clicks of the save button 6. The system verifies the request from the user 7. The system prompts the user that the creation was successful and brings the user back to the Organization Listing page 	
Sub-flows: None	
<p>Alternative/Exceptional flows:</p> <p>6.a The system fails to verify the request from the user and prompts the user that an error has occurred, the system remains at the add organization code page and awaits user input</p>	

Use case: Edit Organization details as Super Admin	Taiga ID: #41
Stakeholders and goals: Super Admin - Edit Organization details	
Description: The Super Admin edits Organization details	
Actors: Super Admin	
Pre-condition: User is logged in as Super Admin and the chosen Organization exists	
Post-condition: User has edited the Organization details	
Trigger: The Super Admin wants to edit Organization details	
<p>Normal flow:</p> <ol style="list-style-type: none"> 1. The user clicks on Organization Listing 2. The system brings the user the the Organization Listing page 3. The user clicks on the edit button of the chosen Organization 4. The system bring the user to edit organization page 5. The user inputs the information required and clicks of the save button 6. The system verifies the request from the user 7. The system prompts the user that the creation was successful and brings the user back to the Organisation Listing page 	
Sub-flows: None	
<p>Alternative/Exceptional flows:</p> <p>6.a The system fails to verify the request from the user and prompts the user that an error has occurred, the system remains at the edit organization page and awaits user input</p>	

Use case: Manage users in an Organization as Super Admin	Taiga ID: #42
Stakeholders and goals: Super Admin - Manage users in an Organization	
Description: The Super Admin manages users in an Organization	
Actors: Super Admin	
Pre-condition: User is logged in as Super Admin and the chosen Organization exists	
Post-condition: User has managed the users in an Organization	
Trigger: The Super Admin wants to manage users in an Organization	
<p>Normal flow:</p> <ol style="list-style-type: none"> 1. The user clicks on Organisation Listing 2. The system brings the user to the Organisation Listing page 3. The user clicks on the manage users button of the chosen Organization 4. The system brings the user to the manage user page 5. The user manages the user of the chosen Organization and clicks the save button 6. The system verifies the request from the user 7. The system prompts the user that the information was saved successfully and brings the user back to the Organization Listing page 	
Sub-flows: None	
<p>Alternative/Exceptional flows:</p> <p>6.a The system fails to verify the request from the user and prompts the user that an error has occurred, the system remains at the manage user page and awaits user input</p>	

Use case: Create user account as Super Admin	Taiga ID: #43
Stakeholders and goals: Super Admin - Create user account	
Description: The Super Admin creates user account for chosen Organization	
Actors: Super Admin	
Pre-condition: User is logged in as Super Admin and the chosen Organization exists	
Post-condition: User has created the user account	
Trigger: The Super Admin wants to create a user account	
<p>Normal flow:</p> <ol style="list-style-type: none"> 1. The user clicks on create account at the side of the page 2. The system brings the user to the Organisation Listing page 3. The user clicks on view user account for the chosen Organization 4. The system bring the user to view organization user page 5. The user clicks on the add user button 6. The system brings the user to the add organization project admin page 7. The user inputs the information required and clicks of the save button 8. The system verifies the request from the user 9. The system prompts the user that the creation was successful and brings the user back to the Organisation Listing page 	
Sub-flows: None	
<p>Alternative/Exceptional flows:</p> <p>8.a The system fails to verify the request from the user and prompts the user that an error has occurred, the system remains at the add organization user page and awaits user input</p>	

Use case: View user account as Super Admin	Taiga ID: #44
Stakeholders and goals: Super Admin - View user account	
Description: The Super Admin view user account for chosen Organization	
Actors: Super Admin	
Pre-condition: User is logged in as Super Admin and the chosen Organization exists	
Post-condition: User has viewed the user account for the chosen Organization	
Trigger: The Super Admin wants to view the user account for the chosen Organization	
Normal flow: <ol style="list-style-type: none"> 1. The user clicks on Organisation Listing 2. The system brings the user the the Organisation Listing page 3. The user clicks on view user account for the chosen Organization 4. The system bring the user to view organization user page 5. The user views the user accounts for the chosen Organization 	
Sub-flows: None	
Alternative/Exceptional flows: None	

Use case: Delete user account as Super Admin	Taiga ID: #115
Stakeholders and goals: Super Admin - Delete user account	
Description: The Super Admin deletes user account for chosen Organization	
Actors: Super Admin	
Pre-condition: User is logged in as Super Admin and the chosen user account exists	
Post-condition: User has deleted the user account for the chosen Organization	
Trigger: The Super Admin wants to delete user account for the chosen Organization	
Normal flow: <ol style="list-style-type: none"> 1. The user clicks on the delete button for the chosen user accounts 2. The system verifies the request from the user 3. The system prompts the user the deletion was successful and brings the user back to the view organization user page 	
Sub-flows: None	
Alternative/Exceptional flows: <ol style="list-style-type: none"> 2.a The system fails to verify the request from the user and prompts the user that the deletion was a failure, the system remains the the view organization user page and awaits user input 	

Use case: Create Employee account as Project Admin	Taiga ID: #116
Stakeholders and goals: Project Admin - Create Employee account	
Description: The Project Admin creates Employee account	
Actors: Project Admin	
Pre-condition: User is logged in as Project Admin	
Post-condition: User has created the Employee account	
Trigger: The Project Admin wants to create a Employee account	
<p>Normal flow:</p> <ol style="list-style-type: none"> 1. The user clicks on create Employee account at the side of the page 2. The system brings the user to the add Employee page 3. The user inputs the information required and clicks of the save button 4. The system verifies the request from the user 5. The system prompts the user that the creation was successful and brings the user back to the previous page 	
Sub-flows: None	
<p>Alternative/Exceptional flows:</p> <p>4.a The system fails to verify the request from the user and prompts the user that an error has occurred, the system remains at the add Employee page and awaits user input</p>	

Use case: Delete Employee account as Project Admin	Taiga ID: #216
Stakeholders and goals: Project Admin - Delete Employee account	
Description: The Project Admin deletes Employee account	
Actors: Project Admin	
Pre-condition: User is logged in as Project Admin and the chosen Employee account exists	
Post-condition: User has deleted the Employee account	
Trigger: The Project Admin wants to delete Employee account	
Normal flow: <ol style="list-style-type: none"> 1. The user clicks on the delete button for the chosen Employee accounts 2. The system verifies the request from the user 3. The system prompts the user the deletion was successful and brings the user back to the view Employee page 	
Sub-flows: None	
Alternative/Exceptional flows: <ol style="list-style-type: none"> 2.a The system fails to verify the request from the user and prompts the user that the deletion was a failure, the system remains in the view Employee page and awaits user input 	

Use case: View employee preference status as Project Admin	Taiga ID: #217
Stakeholders and goals: Project Admin - View employee preference status	
Description: The Project Admin views employee preference status	
Actors: Project Admin	
Pre-condition: User is logged in as Project Admin	
Post-condition: User has viewed employee preference status	
Trigger: The Project Admin wants to view employee preference status	
Normal flow: <ol style="list-style-type: none"> 1. The user click on view Employee 2. The system brings the user to the view Employee page 3. The user views the status of Employee project preference submission status 	
Sub-flows: None	
Alternative/Exceptional flows: None	

Use case: Download result of project assignment as Project Admin	Taiga ID: #218
Stakeholders and goals: Project Admin - Download result of project assignment	
Description: The Project Admin downloads the result of project assignment	
Actors: Project Admin	
Pre-condition: User is logged in as Admin and the automated assignment process is successful	
Post-condition: User has downloaded the result of project assignment	
Trigger: The Project Admin wants to download the result of project assignment	
Normal flow: <ol style="list-style-type: none"> 1. The system brings the user to the result page 2. The user clicks on the download result button 3. The system verifies the request from the user 4. The system generates a PDF file for the user to view offline 	
Sub-flows: None	
Alternative/Exceptional flows: <ol style="list-style-type: none"> 3.a The system fails to verify the request from the user and prompts the user that an error has occurred, the system remains the the result page and awaits user input 	

Use case: View submitted request form as Project Admin	Taiga ID: #220
Stakeholders and goals: Project Admin - View submitted request form	
Description: The Project Admin views submitted request form	
Actors: Project Admin	
Pre-condition: User is logged in as Project Admin	
Post-condition: User has viewed submitted request form	
Trigger: The Project Admin wants to view submitted request form	
Normal flow: <ol style="list-style-type: none"> 1. The user clicks on requests 2. The system brings the user to the view request page 3. The user views the submitted request 	
Sub-flows: None	
Alternative/Exceptional flows: None	

Use case: Reassign employees involvement as Project Admin	Taiga ID: #221
Stakeholders and goals: Project Admin - Reassign employees involvement	
Description: The Project Admin reassigns employees involvement	
Actors: Project Admin	
Pre-condition: User is logged in as Project Admin and the chosen Employee account exists	
Post-condition: User has reassigned employees involvement	
Trigger: The Project Admin wants to reassign employees involvement	
<p>Normal flow:</p> <ol style="list-style-type: none"> 1. The user searches for the account's information by typing their email on the search bar and clicking search. 2. The system brings the user to their profile page. 3. The user reassigns the chosen Employee involved project 4. The user clicks on the save button 5. The system verifies the request from the user 6. The system prompts the user that the changes was saved and brings the user back to the user profile page 	
Sub-flows: None	
<p>Alternative/Exceptional flows:</p> <p>2.a The system fails to verify the request from the user and prompts the user that the save was a failure, the system remains in the user profile page and awaits user input</p>	

Use case: Update project status as Employee	Taiga ID: #222
Stakeholders and goals: Employee - Update project status	
Description: The Employee updates project status	
Actors: Employee	
Pre-condition: User is logged in as Employee and the chosen project exists	
Post-condition: User has updated the project status	
Trigger: The Employee wants to update project status	
Normal flow: <ol style="list-style-type: none"> 1. The user clicks on my projects 2. The system brings the user to the my project page 3. The user selects the desired project 4. The system brings the user to the respective project page 5. The user updates the project status 6. The system verifies the changes made 	
Sub-flows: None	
Alternative/Exceptional flows: 5.a The system fails to verify the request from the user and prompts the user that an error has occurred, the system remains in the respective project page and awaits user input	

Use case: Add task for project as Employee	Taiga ID: #223
Stakeholders and goals: Employee - Add task for project	
Description: The Employee adds task for the chosen project	
Actors: Employee	
Pre-condition: User is logged in as Employee and the chosen project is selected	
Post-condition: User has added task for the chosen project	
Trigger: The Employee wants to add task for the chosen project	
<p>Normal flow:</p> <ol style="list-style-type: none"> 1. The user clicks on the add task button 2. The user enters the information needed and clicks on the save button 3. The system verifies the request from the user 4. The system saves the changes made and prompts the user that the task creation is successful and remains in the respective project page 	
Sub-flows: None	
<p>Alternative/Exceptional flows:</p> <p>4.a The system fails to verify the request from the user and prompts the user that the task was not added, the system remains in the respective project page and awaits user input</p>	

Use case: Delete project task as Employee	Taiga ID: #224
Stakeholders and goals: Employee - Delete project task	
Description: The Employee deletes project task	
Actors: Employee	
Pre-condition: User is logged in as Employee and the task exists	
Post-condition: User has deleted the task	
Trigger: The Employee wants to delete the project task	
<p>Normal flow:</p> <ol style="list-style-type: none"> 1. The user clicks on the delete button besides the desired task 2. The system prompts the user for additional confirmation 3. The user clicks on the yes button 4. The system verifies the request from the user 5. The system deletes the task and prompts the user that the deletion was successful and remains in the respective project page 	
Sub-flows: None	
<p>Alternative/Exceptional flows:</p> <ol style="list-style-type: none"> 3. The user clicks on the no button and nothing happens 4.a The system fails to verify the request from the user and prompts the user that the deletion was a failure, the system remains in the respective project page and awaits user input 	

Use case: Change and reset password as Employee	Taiga ID: #225
Stakeholders and goals: Employee - Change and reset password	
Description: The Employee changes and resets their password	
Actors: Employee	
Pre-condition: User is logged in as Employee	
Post-condition: User has changed and reseted their password	
Trigger: The Employee wants to change and reset password	
<p>Normal flow:</p> <ol style="list-style-type: none"> 1. The user clicks on their profile icon 2. The system brings the user to their profile page 3. The user clicks on edit profile details 4. The system brings the user to the edit profile page 5. The user changes their password and clicks the save button 6. The system verifies the request from the user 7. The system saves the changes made and prompts the user that the save was successful and brings the user back to their profile page 	
Sub-flows: None	
<p>Alternative/Exceptional flows:</p> <p>6.a The system fails to verify the request from the user and prompts the user that the changes was not saved, the system remains in the edit profile page and awaits user input</p>	

Use case: View submission history as Employee	Taiga ID: #226
Stakeholders and goals: Employee - View submission history	
Description: The Employee views submission history	
Actors: Employee	
Pre-condition: User is logged in as Employee	
Post-condition: User has viewed submission history	
Trigger: The Employee wants to view submission history	
Normal flow: <ol style="list-style-type: none"> 1. The user clicks on requests 2. The system brings the user to the request page 3. The user click on request history 4. The system brings the user to the request history page 5. The user views their request submission history 	
Sub-flows: None	
Alternative/Exceptional flows: None	

Use case: Submit a request form as Employee	Taiga ID: #227
Stakeholders and goals: Employee - Submit a request form	
Description: The Employee submits a request form	
Actors: Employee	
Pre-condition: User is logged in as Employee	
Post-condition: User has submitted a request form	
Trigger: The Employee wants to submit a request form	
<p>Normal flow:</p> <ol style="list-style-type: none"> 1. The user clicks on requests 2. The system brings the user to the request page 3. The user clicks on the add request button 4. The system brings the user to the add request page 5. The user enters the relevant informations and clicks the save button 6. The system verifies the request from the user 7. The system saves the request from the user and prompts the user that the request was successfully submitted 8. The system brings the user back to the request page 	
Sub-flows: None	
<p>Alternative/Exceptional flows:</p> <p>6.a The system fails to verify the request from the user and prompts the user that the request was not submitted, the system remains in the add request page and awaits user input</p>	

5. Product Features

The application will provide a wide range of features for all users.

The following table provides an overview of highlighted core and enhanced features that will be implemented throughout the development process.

Role	Features	Description
Employee	Login/Logout (Basic)	Employees will be able to safely access in and out of their accounts.
	Change and reset password	Employees will be able to safely and easily change/reset their account password . Security code will be sent to email linked to the account in the event of a reset.
	Navigate project list (Core)	Employees will be able to navigate through the full project list with their respective description. Project list will only contain projects based on their competency and skills.
	View Project history (Additional)	Employees will be able to look up their past project history
	Select and save Project Preference (Core)	Employees will be able to select their top choices and save these choices before submission. This allows employees to deliberate over their selection before submission.
	View User Profiles (Core)	Employees will be able to easily access other employees profiles to find contact information/details to allow them to contact certain employees etc..

	Manage and edit profile details (Basic)	Employees will be able to manage and edit their profile details according to their preferences.
	Download offline copy of project list (Additional)	Employees will be able to download/print an offline copy for them to view outside the system.
	Request additional team members (Additional)	Employees will be able to request for additional team members.
	View Assignment Results	Employees will be able to see their assigned projects.
	Update Project Status	Employees will be able to update project status to show project progress.
	Add and Delete Project Tasks	Employees will be able to add/delete tasks relating to ongoing projects.
Project Admin	Login/Logout (Basic)	Admin will be able to safely access in and out of their accounts.
	Change and reset password (Basic)	Admin will be able to safely and easily change/reset their account password . Security code will be sent to email linked to the account in the event of a reset.
	Navigate project list (Core)	Admin will be able to navigate through the full project list with their respective description.
	View/Create/Remove/Update project and descriptions	Admin will be able to create/remove/update/view new and existing projects alongside their descriptions.
	Set project requirements (Core)	Admin will be able to set project requirements before the automation process.

		Requirements like employee's competency/enrollment limit will be set using this feature.
	Add & Remove Employees	Admin will be able to add and remove new employees using this feature
	Set and update project deadline (Core)	Admin will be able to set and update deadlines for the project , as the project timeline may change.
	Set Assignment Criteria	Admin will be to set assignment criteria before the automation process , (e.g project threshold)
	View employee profile and assignment status	Admin will be able to view selected employee profiles and their current assignment status.
	Initiating project allocation (Core)	Admin will be able to set project allocation based on various settings like project threshold . After selecting the intended prefixes , the admin can choose to start the automating process.
	Access request submission form (Additional)	Admin will be able to access and read submitted forms made by employees .
	Re-assign employees involved in	If needed , Admin can add or

	ongoing projects (Additional)	remove employees from ongoing projects
	View and update profile details	Admin will be able to view and update their profile details
	Download Offline Copy of Assignment results and overview statistics	Admin will be able to download an offline copy of the assignment result and overview statistic after the automation process.
	View Assignment Results	Admin will be able to view assignment results after the automation process
	View Assignment Overview Statistics	Admin will be able to view the overview statistic after the automation process
Super Admin	Login/Logout (Basic)	S.Admin will be able to safely access in and out of their accounts.
	Change and reset password (Basic)	S.Admin will be able to safely and easily change/reset their account password . Security code will be sent to email linked to the account in the event of a reset.
	Create Organisation Code	S.Admin will be able to create an organization code so that employees/P.Admin will be able to access the system based on their organization.
	Edit and update Organization details	S.Admin will be able to update details regarding the organization (e.g Company name)
	Create/Delete/View/Update User accounts (Core)	User Admin will be able to manage all user accounts and delete or remove if needed.

6. Functional model and Description

Below is a list of complete features in our system roadmap:

Role	Requirements
Employee	<ul style="list-style-type: none"> - Login & Log out - Change & Reset Password - Manage and Update Profile details - View Account Profile - Add Profile Details - View Project Details - Navigate Project list. - Select Project Preferences - View Project History - Save and Submit Project Selections - View Teammate and Project admin profile - View Assignment Results - Download Offline Copy of project list - Request for additional team members - Update Project Status - Add and Delete Tasks
Project Admin	<ul style="list-style-type: none"> - Login & Log out - Change & reset password - Add and Remove Employee Account - Navigate project list. - View and Update Profile Details - View and Update all User list - View and Update project list and descriptions - Set Assignment Criteria - Set Project Requirements - Create & Remove project list and descriptions - Check Project and Submission status - Initiate automatic project assignment - View employee Assignment status - View Assignment Results - Download Offline Copy of Assignment Results - View Assignment Overview Statistics - Download Offline Copy of Assignment Overview Statistics - View request submission form - Reassign employees into ongoing project

Super Admin	<ul style="list-style-type: none"> - Login & Log out - Change & reset password - Create organization code - Edit and update organization details - Manage User accounts - Create/Delete/View/Update user accounts
-------------	---

6.1 Hardware Interface

The product will utilize the MongoDB Atlas which is a document-oriented cloud server database to handle all backend functionalities.

6.2 Software Interface

The Live Server will run on Windows 10/11 settings. The system will be operated using a web application interface where users will be able to access the system with any computer terminal running on a chrome web browser.

7. Non-Functional Requirement

Performance	<ul style="list-style-type: none"> ⊙ Responsiveness of the website ⊙ Responsiveness of the features
Security	<ul style="list-style-type: none"> ⊙ User information is protected by their email as login ID and their password ⊙ Important data will always be encrypted ⊙ Advanced feature is protected behind permission given to different users, such as deletion and creation of accounts ⊙ Only users with accounts will be allowed access into the system
Reliability	<ul style="list-style-type: none"> ⊙ Ability to perform said task under the expected time in an controlled situation ⊙ Little to no down-time ⊙ Readily accessible and available
Usability	<ul style="list-style-type: none"> ⊙ Website is user-friendly and intuitive ⊙ Users will only be able to access functions relating to their roles ⊙ Upon account creation , users will be able to access the application
Maintainability	<ul style="list-style-type: none"> ⊙ System logs will log and update every user actions ⊙ Data stored in server is backed up

Verifiability	<ul style="list-style-type: none"> ⊙ Dummy data will be used to test the system and application features ⊙ Dummy data will be used during the various stages of development ⊙ Developers and testers will have identical development environment configurations
Confidentiality	<ul style="list-style-type: none"> ⊙ Users will not be able to view the details of other users , only administrators are allowed to ⊙ Important details like password and user information will be encrypted before storing into database

8. Project Backlog

As a	I want to be able to	So that	Score	Sprint	ID
Project Admin	Login	I can use the system's feature	5	1	#3
Project Admin	Log out	I can prevent unauthorized access	5	1	#4
Project Admin	Add contact details	Other personnels can contact me	5	2	#5
Project Admin	View account information	I can see the personal details in the system	5	2	#6
Project Admin	Update contact details	I can keep my contact details updated	5	2	#7
Project Admin	Create new project	Employees will be able to see them	5	-	#8
Project Admin	Update project details	I can make necessary changes	5	-	#10
Project Admin	Set project threshold	I can decide the project per employee ratio	5	-	#11
Project Admin	Set employee limit per project	I can limit the no. of employees enrolled	5	-	#12
Project Admin	Check project enrollment	I can make sure no project is unassigned	5	-	#13
Project Admin	Remove project	Unavailable projects can be deleted	5	-	#14
Project Admin	View employee information	I know the involvement of each employee	5	-	#15
Project Admin	Start automation process	I can give the employee their assignment	5	-	#16
Project Admin	View assignment result	I know who each employee's assignment	5	-	#17
Project Admin	View no. of people enrolled	I know the quality of the system's algorithm	5	-	#18

Project Admin	View unassigned projects	I know the effectiveness of the algorithm	5	-	#19
Employee	Login	I can use the system's feature	5	1	#21
Employee	Log out	I can prevent unauthorized access	5	1	#22
Employee	Add contact details	Other personnels can contact me	5	2	#23
Employee	View account information	I can see the personal details in the system	5	2	#24
Employee	Update contact details	I can keep my contact details updated	5	2	#25
Employee	Add skills and competency	I can provide the relevant information	5	2	#26
Employee	View skills and competency	I can see my relevant information	5	-	#27
Employee	Update skills and competency	I can keep the information updated	5	2	#28
Employee	See a list of available projects	I know what project are available	5	-	#29
Employee	View project details	I can obtain more project information	5	-	#30
Employee	Input project preference	I can be assigned to a preferred project	5	-	#31
Employee	Save and submit preference	My selection is recorded in the system	5	-	#32
Employee	See result of allocation	I know what are my allocated project(s)	5	-	#33
Employee	View project mate information	I know what to expect from them	5	-	#34
Employee	View project mate contact	I know how to contact them	5	-	#35
Employee	View project admin contact	I know how to contact him/her	5	-	#36
Employee	Download project listing	I can view it at my convenience	5	-	#37
Super Admin	Login	I can use the system's feature	5	-	#38
Super Admin	Log out	I can prevent unauthorized access	5	-	#39
Super Admin	Create organization code	I can allocate users to their organization	5	-	#40
Super Admin	Edit organization details	I can keep the information updated	5	-	#41
Super Admin	Manage user in organization	I can ensure all user is allocated correctly	5	-	#42
Super Admin	Create user account	Organization user can access our system	5	2	#43
Super Admin	View user in organization	I can see who is in which organization	5	2	#44
Super Admin	Delete user account	I can remove invalid users	5	2	#115

Project Admin	Create Employee account	There is no unauthorized Employee account	5	-	#116
Project Admin	Delete Employee account	All unused Employee account are removed	5	-	#216
Project Admin	View Employee preference status	I can remind them to do so	5	-	#217
Project Admin	Download assignment result	I can view it at my convenience	5	-	#218
Project Admin	View submitted request form	I can approve or reject the request	5	-	#220
Project Admin	Reassign Employee involvement	I can better divide the workload	5	-	#221
Employee	Update project status	I can keep track of the progress	5	-	#222
Employee	Add tasks required for project	I can set progress to follow	5	-	#223
Employee	Delete task for project	I can remove task that are no longer needed	5	-	#224
Employee	Change and reset my password	I can keep my information secured	5	-	#225
Employee	See submission history	I can keep track of my submission progress	5	-	#226
Employee	Submit a request from	I can request for more team members	5	-	#227

9. Project Plan

9.1 Development Method

The methodology we will adopt is the Agile and SCRUM development model, which consists of 5 phases. As for our project management tool, we are using Taiga, an agile project management platform.

Phase 1 - Requirements Gathering & Planning

At this stage, we will be analyzing the current systems and methods used by academic institutions for project allocation, and identifying shortcomings that can be resolved. To help us better understand the functional requirements of our project from the users' perspective, we will develop user stories to identify the functionalities that our system should have. Followed by ranking these requirements and planning the scope of our proposed system. After which, we will decide on the programming languages to use, and the project's feasibility. Accordingly, we will delegate tasks among the group members, and plan weekly meetings to discuss and update each other on the progress of the tasks. The minutes of our meetings will also be done progressively. We will be using a Gantt chart to keep track of our project deadlines and milestones.

Deliverables: Project proposal, user stories, and system requirements specification ready for Assessment 1.

Phase 2 - Designing

From the system requirements we have identified, we will transform the requirements into a structured use case diagram and descriptions to summarize the features of the users' interactions with our system. A class diagram will be developed to map out the blueprint of our system and display the relationships and dependencies of the system properties. We should have our use cases and class diagrams ready for Assessment 1 on 12 November 2022.

At the same time, we will also be working on our system prototype, which consists of the bare basic features. We should aim to accomplish this by the end of sprint 4, for Assessment 2 on 17 December 2022.

Deliverables: UML use case diagram, use case descriptions, Class Diagrams, system prototype, system prototype PowerPoint slides for Assessment 2.

Phase 3 - Developing

Using the proposed programming languages, each of us will be assigned tasks to develop the system. Around the midpoint of this phase when the basic functionalities of the system are done, we will incorporate the testing stage to identify the system flaws that have to be fixed to minimise the risks of any failures.

Deliverables: A working system that is still undergoing testing

Phase 4 - Testing

This stage requires us to test the functionalities of our system during the development stage to identify any errors and features we have to fix. As well as making sure system bugs are not overlooked. We will also be acquiring feedback and suggestions from our project supervisor to make improvements. Once our system is fully developed, we will start on our test plan, test cases and results for the technical manual document in preparation for our final submission.

Deliverables: A functional system, test plan, test cases and results.

Phase 5 - Implementation/Deployment of system

At this stage, we should have a fully functional system and we will be finishing up our test cases for our technical manual. All our meeting minutes, system reports, and relevant documentation will be compiled into our final project documentation for submission. A marketing video that advertises the features of our final product will also be made.

Deliverables: Final technical manual, final project documentation, fully functional product, marketing video

9.2 Development Tools

For our web-based application development process , we will be using **ReactJS** for our front-end development and for our back-end development we will be using **ExpressJS** in tandem with **NodeJS**. With **ReactJS** reusability and DOM feature , we will be able to break down and test React components independently. Its reusable components and modular structure allows us to quickly scale up our front-end development with effortless maintenance .

ExpressJS is a flexible **NodeJS** web application framework that provides a robust set of features and enhances the functionality of **NodeJS**. With **NodeJS** executing requests quickly with the use of event loops and the ease of coding using **ExpressJS**.

We will be able to handle a large number of client requests and reduce coding time , saving time for both the users and the company.

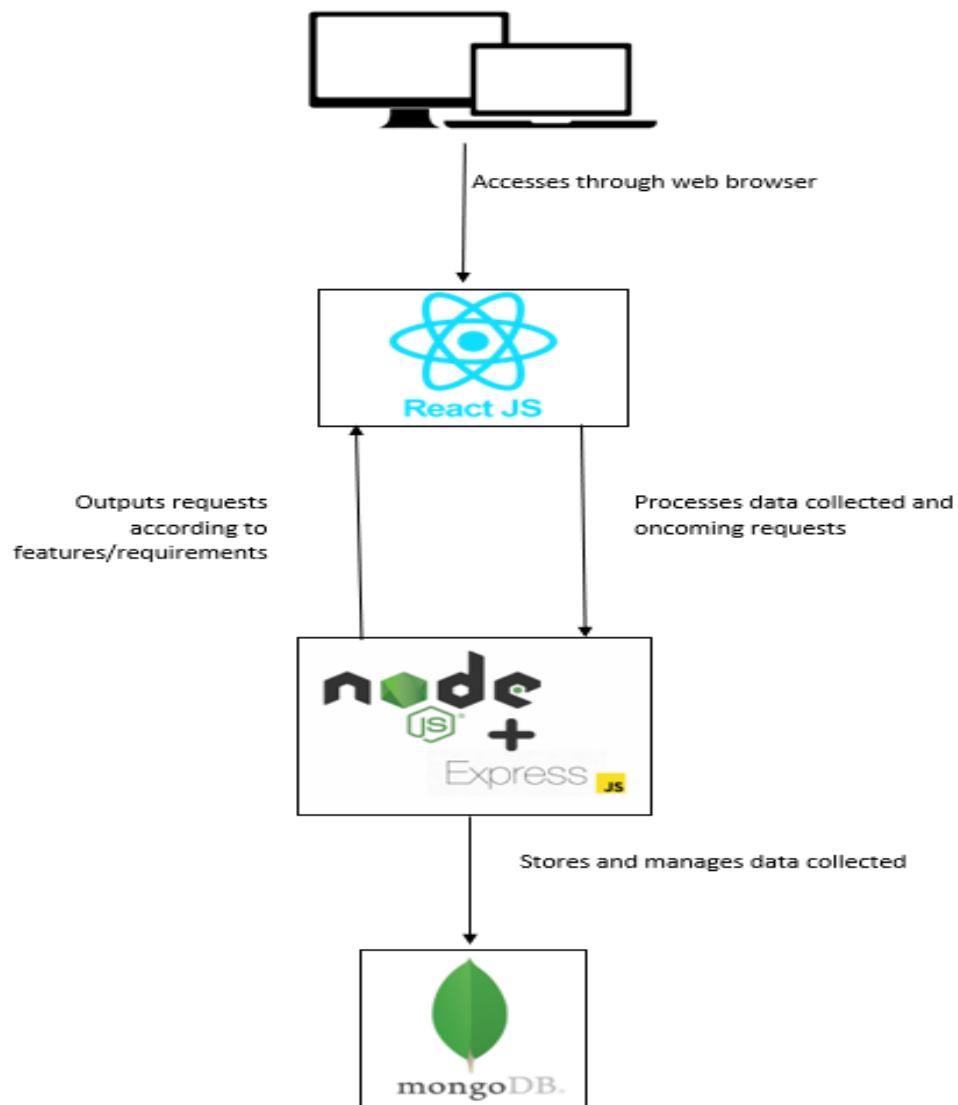
For the storage and management of our data , we will be using **MongoDB** as our centralized database. **MongoDB** is not a Relational Database Management System like mySQL but a document-based database . Given its flexible schema approach in storing data , it will work perfectly with our agile development model.

10. Risk Management Matrix

Risk Description	Probability	Impact	Mitigation
Change in Technology	Improbable	Tolerable	<p>Team should be in-touch with the latest updates involving software used.</p> <p>At least one team member should be tasked in maintaining the software/servers to its latest version.</p>
Failure to meet deadlines	Possible	Intolerable	<p>Team is to update project progress on Taiga regularly.</p> <p>Team members should play to their strengths and weaknesses and occupy positions that will ensure a smoother workflow.</p> <p>Project leads should update their team's progress regularly so that the project manager can monitor deadlines and make the necessary changes.</p>
Inability to meet software requirements and implement core features	Possible	Intolerable	<p>Through prototyping , our team will be able plan for and make changes according to the outcome.</p> <p>Product owner and team should be communicating closely to ensure that the team understands the final outcome of the software.</p>
Inability to implement additional features	Possible	Tolerable	Software with core features should be completed earlier so that the team will have more time to work on the additional features.
Unforeseen software bugs and errors	Probable	Acceptable	Frequent testing during the development stages and independent testing at the end of each sprint should be implemented.
Unplanned absence/withdrawal of a team member	Improbable	Undesirable	<p>Team should forecast dates which they are unable to commit to and inform the team in advance should they withdraw from the course.</p> <p>Workload will be evenly re-distributed</p>

			among the team if needed.
Poor team communication	Possible	Undesirable	<p>Appropriate communication channels should be set up (e.g Discord , Telegram).</p> <p>Regular team meetings should be held in online and offline settings to foster team bonding and allow for team updates.</p>
Zero/Minimal contribution from team member	Improbable	Undesirable	<p>Team should be communicating regularly and hold each other accountable.</p> <p>The Project Manager should intervene if needed.</p>
Changes in software requirements	Probable	Acceptable	Buffer time should allow us to reprioritize our workload and account for shorter/longer sprint intervals to accommodate these changes .
Scheduling risks	Possible	Tolerable	<p>Team should review and update the project timeline regularly .</p> <p>Schedule buffer should be incorporated into the project timeline.</p>

11. System Design



Users will be able to access our system through any browser using a web-connected device.

Our system will be built upon the three-tiered architecture , which consists of three computing tiers ; the presentation tier , the application tier and the data tier. Given that each tier runs on its own infrastructure , each tier can be developed concurrently and updated or scaled without affecting the other layers.

For our presentation tier , the webpage's intuitive and user-friendly UI will be developed using React. Providing users with an interactive and seamless user experience , React will be supporting all the incoming requests when users interact with our features and collect information gathered from these users.

For our application tier ,

Express.js will be used for the development of backend web application frameworks that are to be layered on top of Node.js and will replace the functions used on an excel worksheet .

Our main scripting language for the development of backend functions will be Node.js which will process the information collected and ensure all the inner workings of our user-side of the webpage is up and running.

For our data tier,

MongoDB will be our supporting database which will store and manage all the necessary information to be processed by the application tier.

12. Change Management Plan

As our project development progresses , user stories may be introduced or changed . When such a change occurs , there is a need for review to ensure that the proposed change conforms with our project requirements.

There are a few aspects to consider when implementing a new change

- Requester & Role
- Category (User Stories, additional features ...)
- Priority and Reason for change
- Impact of change and time and effort required

12.1 Roles and Responsibilities in Change Review

The table below summarizes the team's responsibilities and involvement in the review of the change proposal

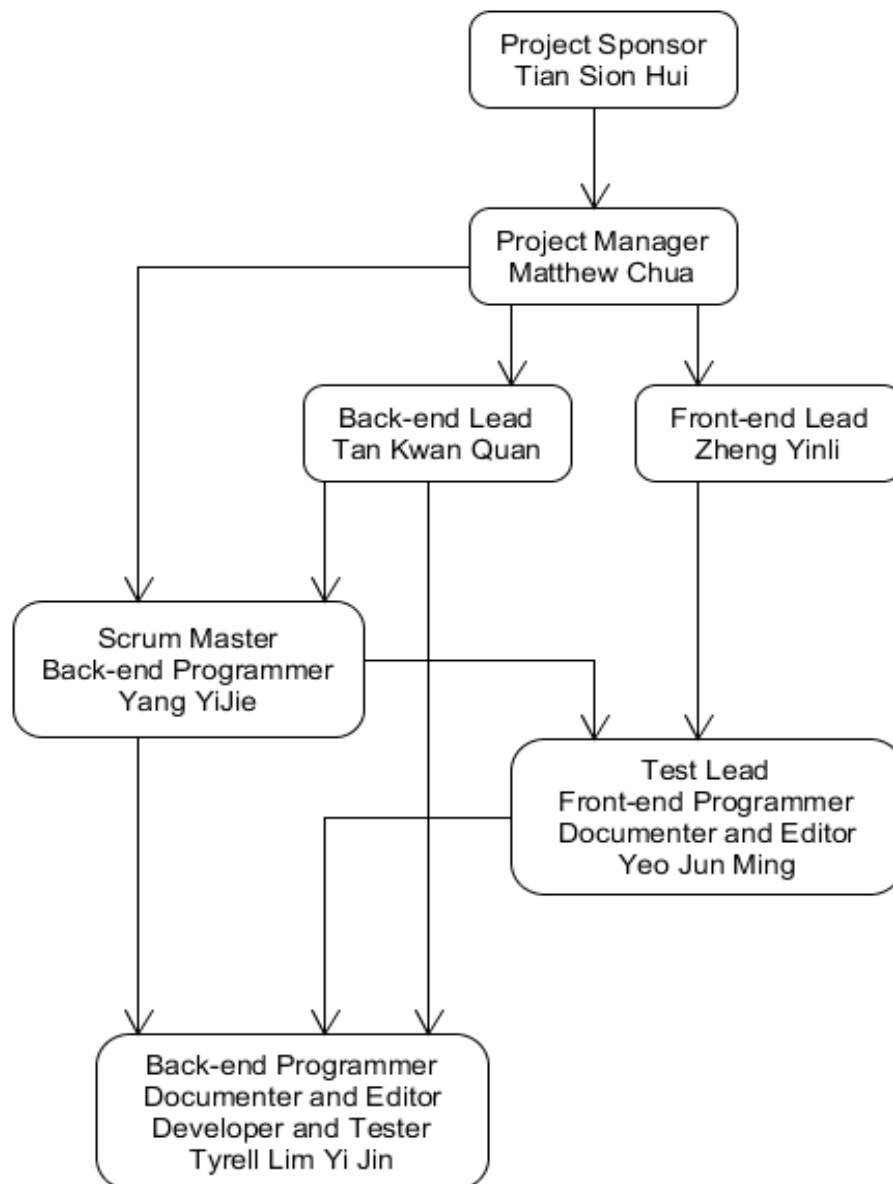
Roles	Responsibilities
Project Manager	<ul style="list-style-type: none"> ● Review Change request ● Reassigns project schedule and resources to carry out change
Project Team	<ul style="list-style-type: none"> ● Submit change requests ● Implement changes
Scrum Master and Leads	<ul style="list-style-type: none"> ● Review change request ● Implement and integrate changes into project ● Records changes onto backlog

12.2 Rules and procedures to carry out request

1. Who can submit a change request?
 - a. Supervisor , Assessor or clients.
2. Change control board
 - a. Supervisor
 - b. Project leads
 - c. Client
3. Procedure
 - a. Submission of change request form
 - b. Change is reviewed by control board
 - c. Decision made to approve/reject request
 - d. Project team to manage and implement the change
 - e. Change is recorded onto Taiga backlog.

13. Proposed Team Organization

13.1 Organization Diagram



13.2 Roles and Responsibilities

Roles	Responsibilities
Project Sponsor/Supervisor	<ul style="list-style-type: none"> - Provides necessary resources and support towards the team - Informs team of potential changes - Provides feedback and ensures team meets the project requirements
Team Leader/Project Manager	<ul style="list-style-type: none"> - Managing Team Members - Delegating Roles and Responsibilities - Amend and allocate/reallocate project schedule and resources - Ensures deadlines are met
Backend Lead	<ul style="list-style-type: none"> - Coordinates with other backend programmers and oversees team's deadline - Develops features in relation to user stories regarding backend - Compile backend's team progression and tasks for Project Manager
Frontend Lead	<ul style="list-style-type: none"> - Coordinates with other frontend programmers and oversees team's deadline - Develops features in relation to user stories regarding frontend - Compile frontend's team progression and tasks for Project Manager
Backend Programmer	<ul style="list-style-type: none"> - Develops features in relation to user stories regarding backend
Frontend Programmer	<ul style="list-style-type: none"> - Develops features in relation to user stories regarding frontend
Test Lead	<ul style="list-style-type: none"> - Coordinates test plans - Schedules all product tests - Oversees and delegates tests to other tester
Developer and Tester	<ul style="list-style-type: none"> - Participates in the development of features relating to prototype/product - Participates in the testing of features relating to prototype/product

Scrum Master	<ul style="list-style-type: none">- Oversees documentation progress- Creates and maintains product backlog- Assigns tasks to streamline team's progression- Ensures documentation fits requirements and meet deadlines
Documenter and Editor	<ul style="list-style-type: none">- Documents essential diagrams and information regarding project- Edits and proofreads other documenters' work if necessary.

14. Project Plan

14.1 Project Milestones & Deliverables

Milestones	Deliverables	Date to be completed
Setting up of communication channels and Figma/Taiga	-	10 October 2022
Setting up of product website	-	31 October 2022
Technical Research on project topic	-	31 October 2022
Project Requirements Documentation	Project Specification	11 November 2022
	Project Proposal	
	Use Case Diagram	
	Use Case Description	
Project Progress Report	Activity Diagram	10 December 2022
	Class Diagram	
	Sequence Diagram	
	Wireframe	
	Progress Report	
System Prototype with basic functionalities	Website Prototype	10 December 2022
Technical Design Manual (Initial)	Technical Design Manual Document	10 December 2022
User Manual (Initial)	User Manual Document	10 December 2022
Submission of System Prototype with Project Progress Report to Moodle.	Website Prototype	17 December 2022
	Project Progress Report	
Project Progress Presentation (Prototype Demo)	Website Prototype	17 December 2022
	Prototype Presentation Slides (pptx.)	
End of Phase 1		

Implementation and testing of phase 2	Test Case Document	23 January 2023
	Project Product	
Completion of Product Development, User Manual, Test Case Document	Final Product	18 February 2023
	User Manual	
	Test Case Document	
Submission of Final Product and Documentation to Moodle	Final Documentation	18 February 2023
	Final Product	
Submission of FYP Presentation Slides and Video of Product Presentation to Moodle	Final Presentation Slides (pptx.)	25 February 2023
	Product Video	

14.2 Project Schedule

Our schedule for the development of this project will be split into two phases ;

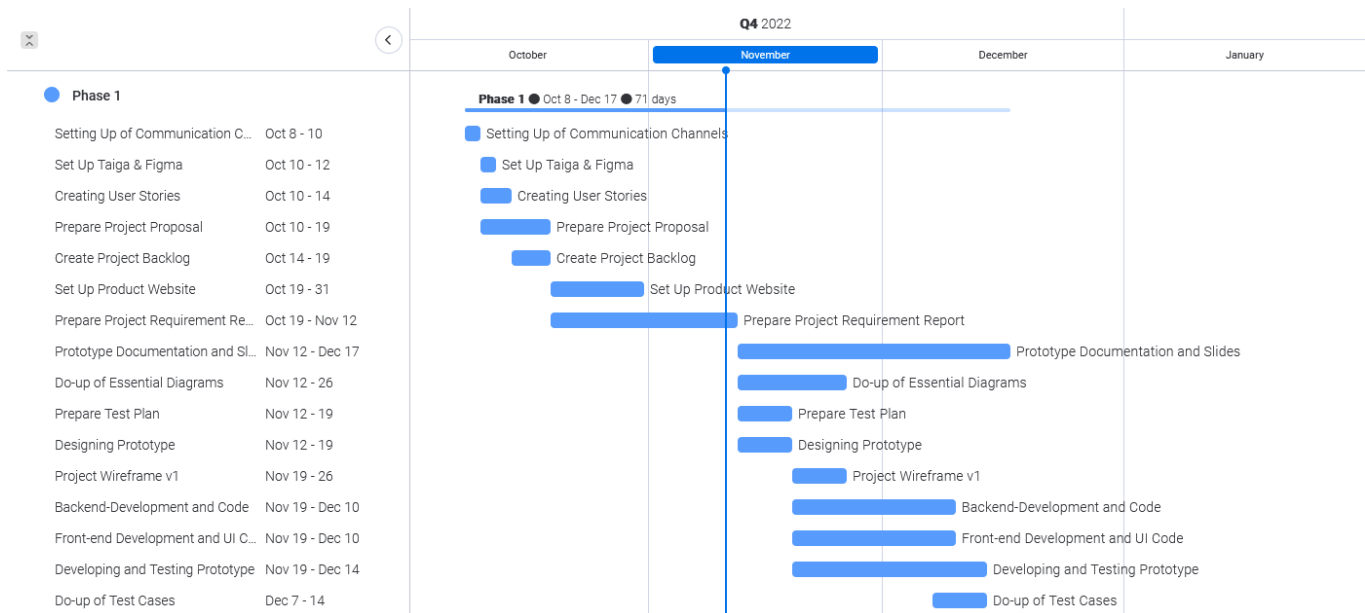
The research and ideation phase will last from 8th Oct 2022 to 17th Dec 2022.

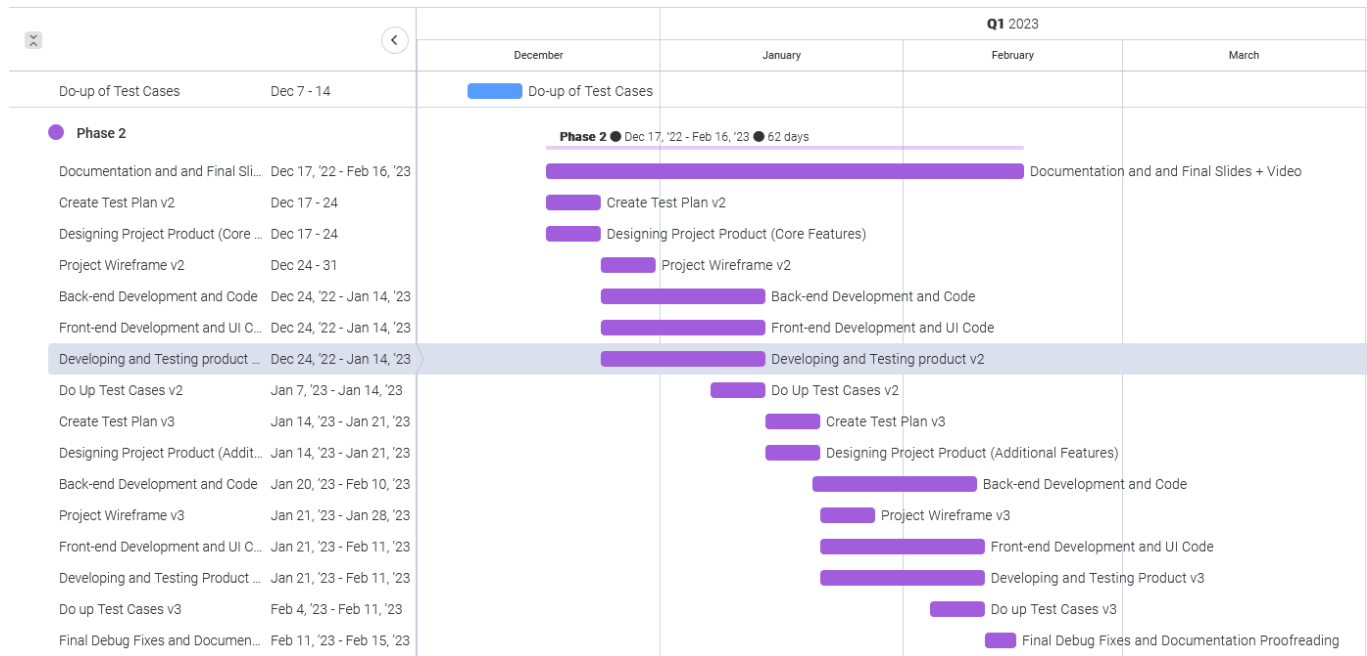
Focus for this phase of our project would be towards theorising and researching before developing a working prototype with some working core features.

The design and development phase will last from 17th Dec 2022 to 25th Feb 2023.

For the second phase , we will mainly be focusing on developing and designing our final product with frequent testing and delivering additional features alongside the core features.

Research and Ideation Phase



Design and Development Phase

15. References

Business Bliss Consultants FZE. (November 2018). Student Project Allocation for Monitoring Duplication.

Retrieved October 14, 2022 from

<https://ukdiss.com/examples/student-project-allocation-monitoring-duplication.php?vref=1>

Scrum phases and Processes. SCRUMstudy. (n.d.). Retrieved October 15, 2022, from

<https://www.scrumstudy.com/whyscrum/scrum-phases-and-processes>

Information Literacy Introduction Program

<http://www.library.uow.edu.au/helptraining/workshops/ilip/>

Intellectual Property Policy

<http://www.uow.edu.au/research/researchmanagement/1998IP.html>

SITACS Style Guide for Footnotes, Documentation, Essay and Report Writing

<http://www.sitacs.uow.edu.au/info/current/styleguide.pdf>

Project Background Sample Clauses

<https://www.lawinsider.com/clause/project-background>

16. Appendix

16.1 Appendix A : Change Request Form

Change Proposal	
Name	
Role	
Description of Change	
Reason for Change	
Priority (High ,Medium ,Low)	
Impact on Deliverables	
Estimated Duration	

16.2 Appendix B: MongoDB resources

<https://www.tutorialsteacher.com/mongodb/what-is-mongodb>

16.3 Appendix C: Express.js resources

<https://en.wikipedia.org/wiki/Express.js>

16.4 Appendix D: React resources

<https://www.freecodecamp.org/news/why-use-react-for-web-development/>

<https://reactjs.org/blog/2013/06/05/why-react.html>

16.5 Appendix E: Node.js resources

https://www.tutorialspoint.com/nodejs/nodejs_introduction.htm