UNIVERSITY OF WOLLONGONG AUSTRALIA

Singapore Institute of Management

School of Computer Science & Information Technology

CSIT-22-S4-24 - Automatic Project Assignment

Technical Manual 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: Technical Manual Document
Document Name: APA-TM-01

Owner	Current Version	Last Change on		Approved by
		Date	Time	
Matthew Chua Yu Jie	1.1	17/02/2023	2000	

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

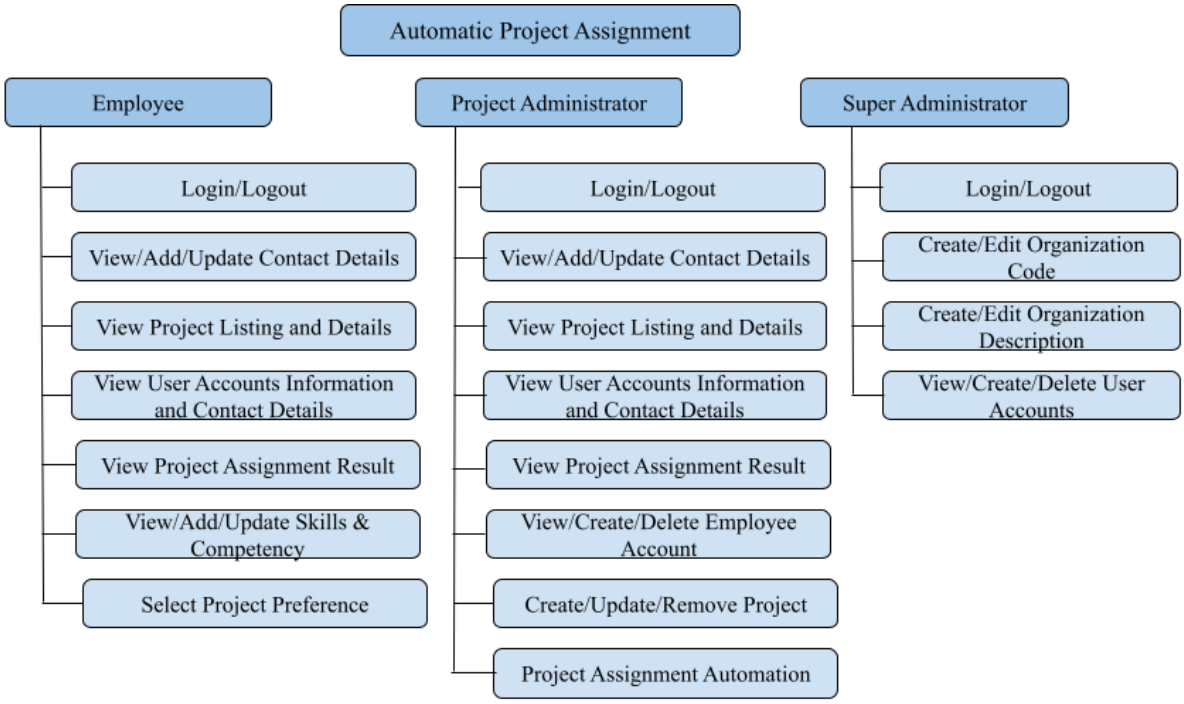
Record of Revision

Revision Date	Description	Section Affected	Changes Made by	Version after Revision
11/10/2022	Draft of the Technical Manual Document	All	Jun Ming	1.0
17/02/2023	Completion of the Technical Manual Document	All	Everyone	1.1

Table of Content

1. Introduction

This document provides an overview of the Technical Design and Implementation for the automatic project assignment application.



2. Purpose

Our objective is to have a single system that handles the processes related to the management of the automatic project assignments. These includes:

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 prioritized 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.

3. Product Features

3.1 Functional Features

Account Type	Functions
Employee	Login/Logout Change & Reset Password Navigate Project List Select and Save Project Preferences View User Profile Manage and Edit Profile Details View Assignment Results Add/View/Edit Skills & Competencies
Project Admin	Login/Logout Change & Reset Password Navigate Project List View/Create/Remove/Update project and descriptions Set project requirements Add & Remove Employees Add/Delete/View Skills Set Assignment Criteria View employee profile Initiating project allocation View and update profile details Create Project Assignment View Assignment Overview Statistics Search for User Accounts
Super Admin	Login/Logout Change and reset password Create Organisation Edit and update Organisation details Manage User Accounts Create/Delete/View/Update User accounts Search for User Accounts

3.2 Non-Functional Requirements

Performance	◎ Responsiveness of the website ◎ Responsiveness of the features ◎ Any interface between user and system shall have a response time not

	exceeding ten seconds ◎ System must accommodate a stipulated amount of simultaneous users
Security	◎ 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	◎ Ability to perform said task under the expected time in an controlled situation ◎ Little to no down-time ◎ Readily accessible and available ◎ System should be rolled back to previous version should an update/error or system failure occur to restore previously saved data
Usability	◎ 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	◎ System logs will log and update every user actions ◎ Data stored in server is backed up
Verifiability	◎ 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	◎ 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 ◎ Employees will not be able to view any details/information pertaining to other employees
Integrity	◎ All data should be validated before being stored within the database ◎ Any changes within the system will be logged

4. Operating Environment

4.1 Hardware Requirements

Windows Desktop/Laptop (Operating System)	Windows 11 - 32-bit/64-bit versions of Home, Pro, and Enterprise Windows 10 - 32-bit/64-bit versions of Home, Pro, and Enterprise
Browser	Microsoft Edge Google Chrome Firefox Opera
iMac/Macbook	macOS X 10 (or above)
Browser	Safari Microsoft Edge Google Chrome Firefox Opera

4.2 Technology/Software Requirements

Technology/Software	**Description**	**Application to Automatic Project Assignment**
MongoDB Details: Appendix A	MongoDB is an open-source, cross-platform, and distributed document-based database designed for ease of application development and scaling. 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	● 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

	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 like other SQL (RDBMS) databases.	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	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. It is designed for building web applications and APIs. It has been called the de facto standard server framework for Node.js.	● Scripting language used for the development of backend web application frameworks that are to be layered on top of Node.js ● Replaces the functions that are used on an excel worksheet.
React Details: Appendix C	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. Its simplicity and flexibility in building components make it one of the most popular tools for front-end development.	● 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	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. Node.js also provides a rich library of various JavaScript modules which simplifies the	● 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 database

	development of web applications using Node.js to a great extent.	

5. System Design

5.1 User Interface Design

5.1.1 Overview

This section contains the wireframe designs done for the User Interface of Automatic Project Assignment. The wireframe designs shown in this section are subjected to review and may vary from the final GUI of the end product.

5.1.2 Login Page

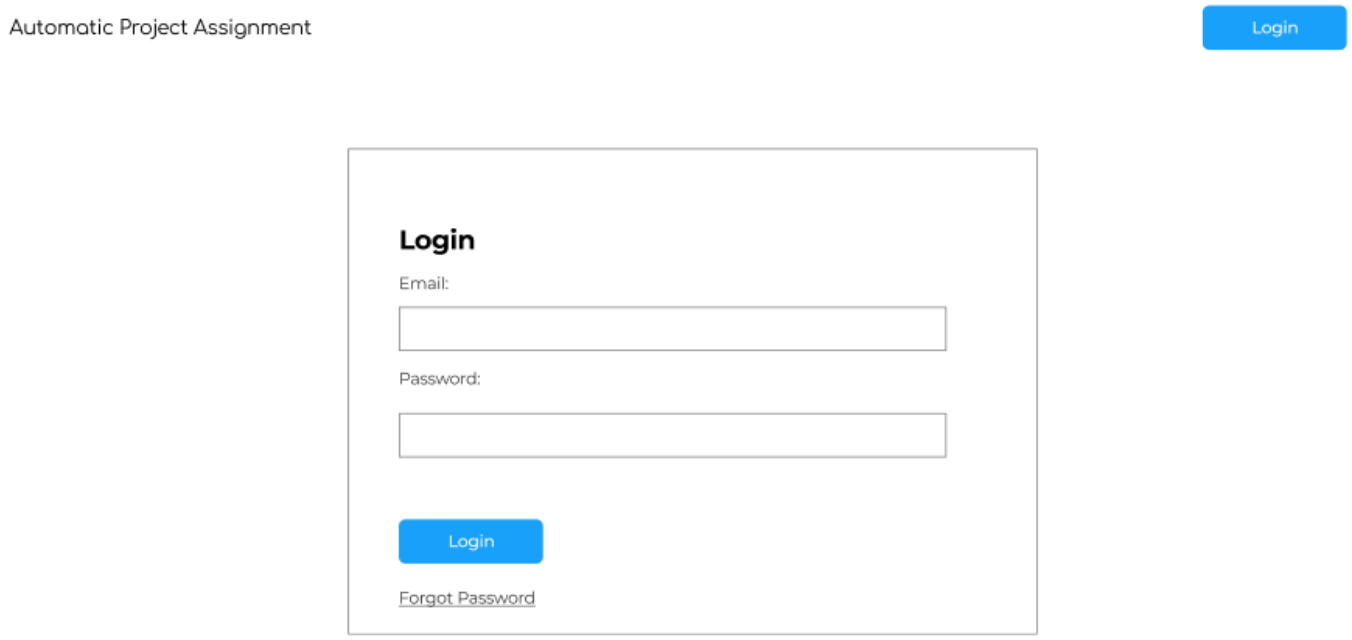


Figure 1, User Login Page

In Figure 1, all users (Super Admin, Project Admin, Employees) will be required to enter their email and password to login into the website. If the user entered the wrong login details, an error message will prompt saying “Invalid Login Credentials”. In the event a user forgets their password, the user can click “Forgot Password” to reset their password which will lead to the next figure below.

5.1.3 Forgot Password Page

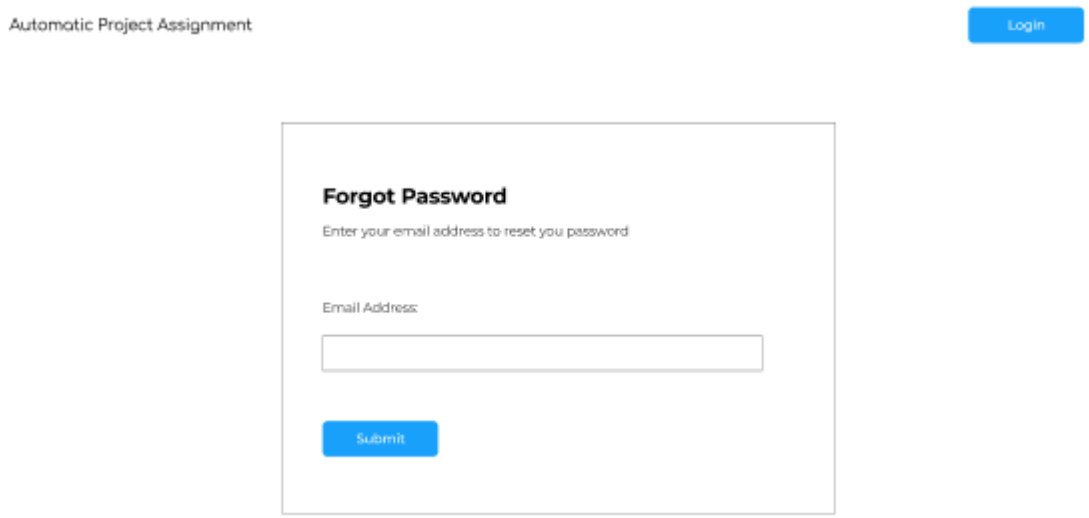


Figure 2, Forgot Password

In Figure 2, if the user clicks on "Forgot Password" in Figure 1, the system will redirect them to this "Forgot Password" page, where they must enter their email address to reset their password. The user will then click on the "Submit" button after entering the email address of their account. If the email address exists in the database, an email with the reset token will be sent to the user's email. In order to successfully reset the password, the user will enter the token, enter the new password, and confirm the password, as shown in the following figure.

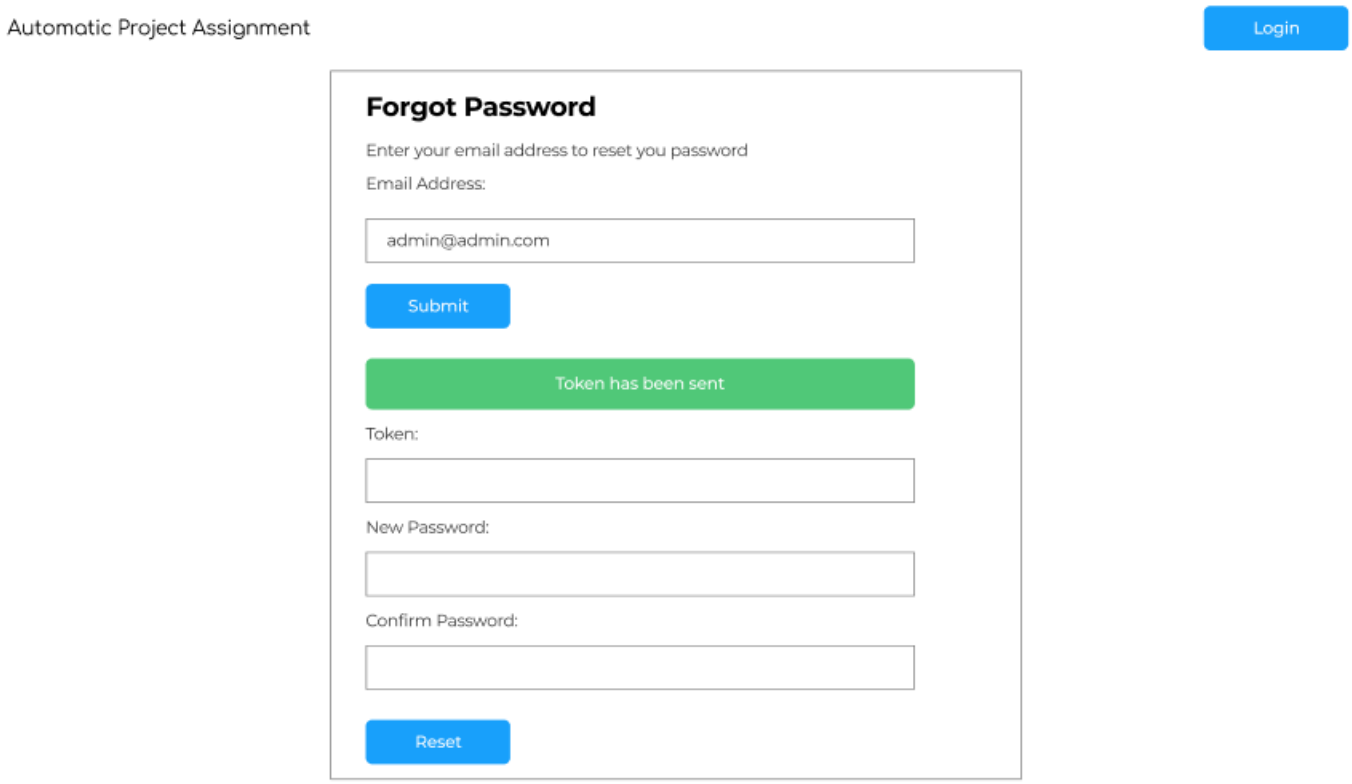


Figure 3, Resetting Password

Project Admin UI Designs

5.1.4 Project Admin Main/Project Listings Page

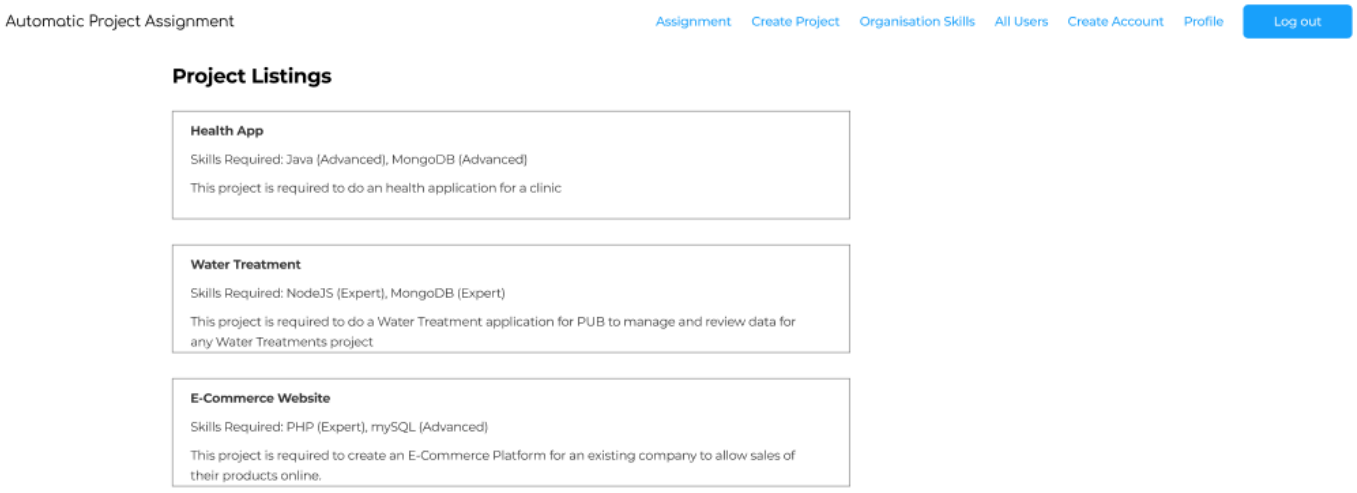


Figure 4, Project Admin Main Page

In Figure 4, this will be the main landing page for the Project Admin. This page displays the project listings

available as well as the details of the organisation the Project Admin belongs to. The Project Admin will be able to log out from this main page as well.

5.1.5 Project Admin Assignment Page

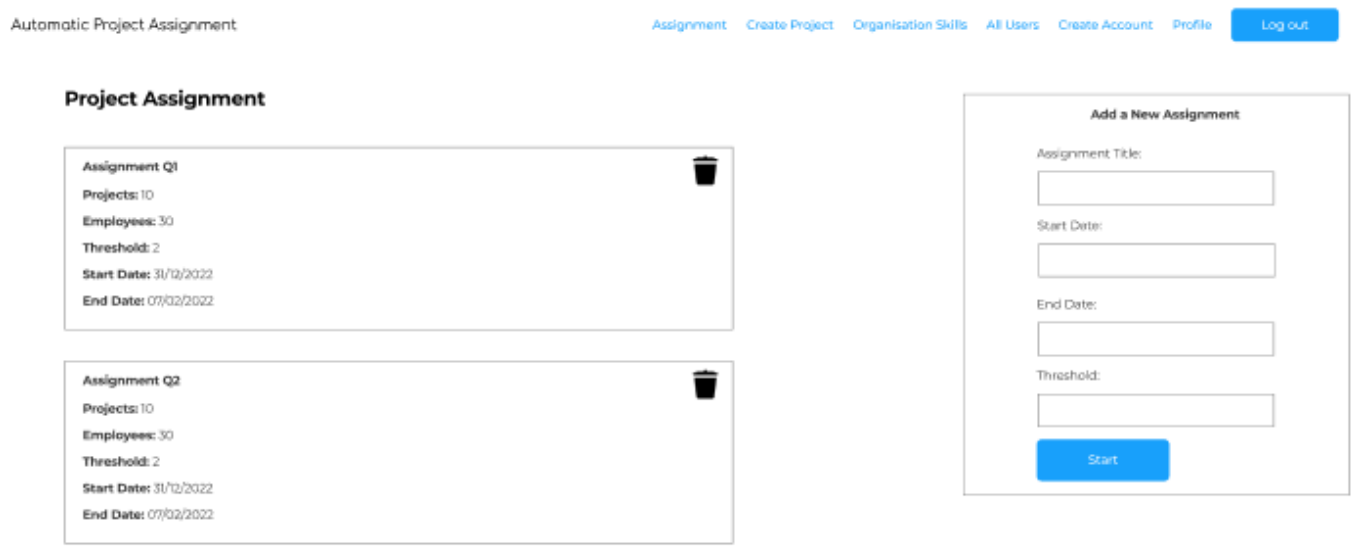


Figure 5, Project Admin Assignment Page

In Figure 5, this page is for the Project Admin to view project automation assignments and also create new project automation assignments. By filling in the details on the right side field and clicking on the “Add Assignment” button, a newly created assignment will be shown in the listing. The Project Admin will have to click on "Assignment" from the listing to add the employees involved and the projects to process the automation allocation of the projects.

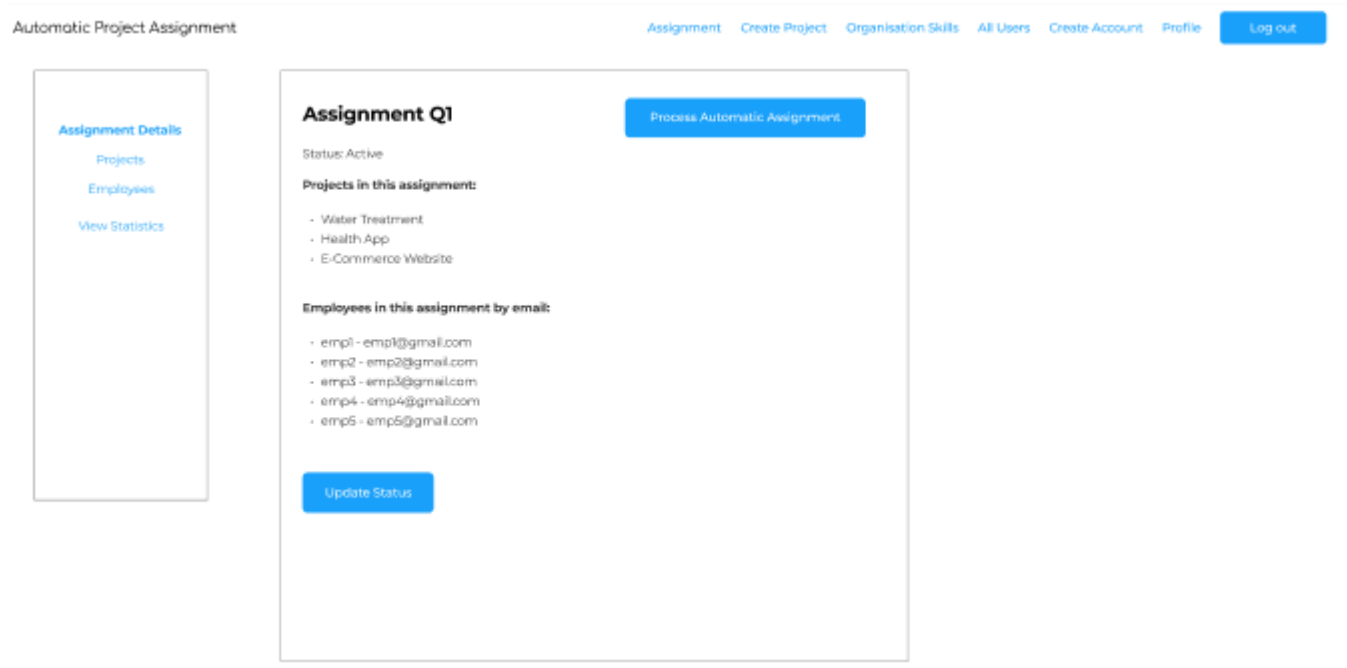


Figure 6, Assignment View Page

As shown in Figure 6, this page will allow the Project Admin to edit the details of this automation assignment process by selecting the projects that are involved as well as the employees. They will be able to add more Projects by selecting “Project” on the sidebar and Employees by selecting “Employee” from the sidebar. Once all the details have been confirmed, Project Admin simply click on the “Process Automatic Assignment” button to allow the website to auto-allocate projects to the employees based on their preference selection and skill competency level. Project Admin will be able to view statistical results of the automation assignment by clicking on the “View Statistics” button.

5.1.6 Project Admin Create New Project Listing

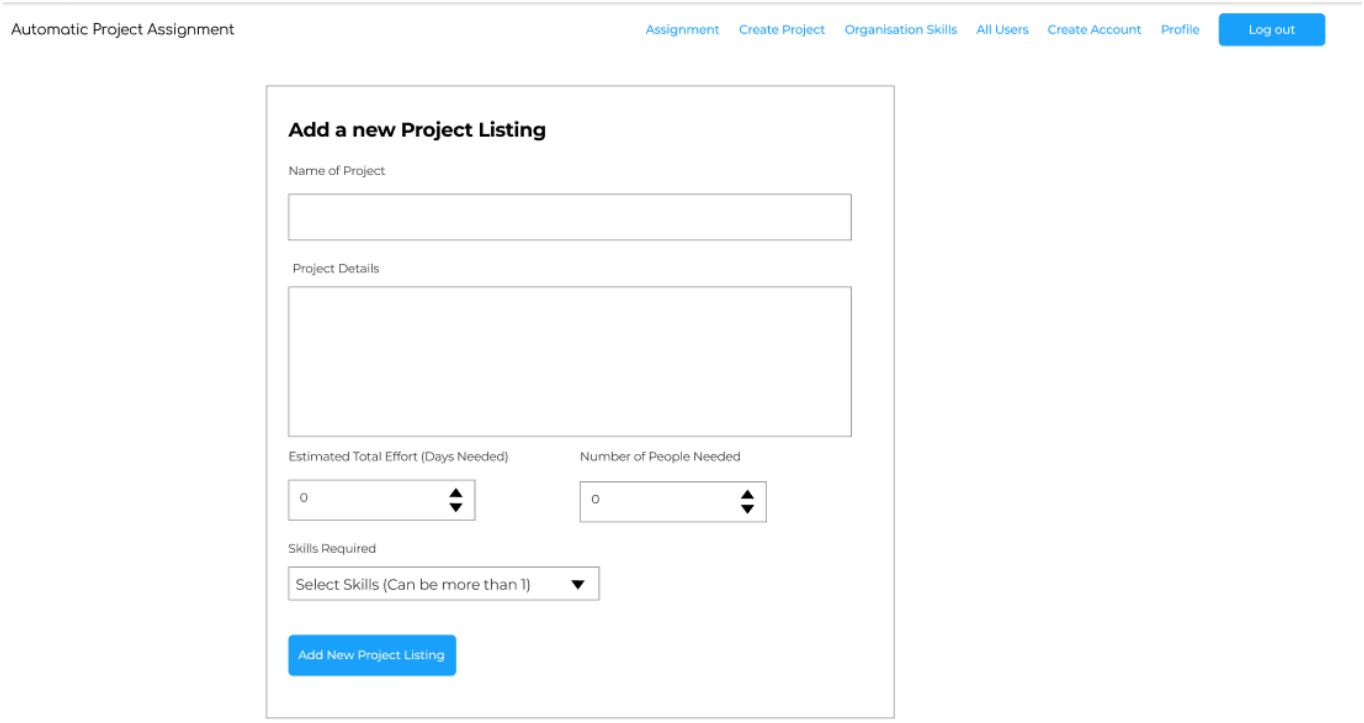


Figure 7, Create Project Listing Page

In Figure 7, this page allows the project admin to create a new project listing in the system for their organisation. Once the fields are all filled up, clicking on the “Add New Project Listing”, the system will create the project listing and redirect back to the Project Listings page.

5.1.7 Project Admin View Project Listing Page

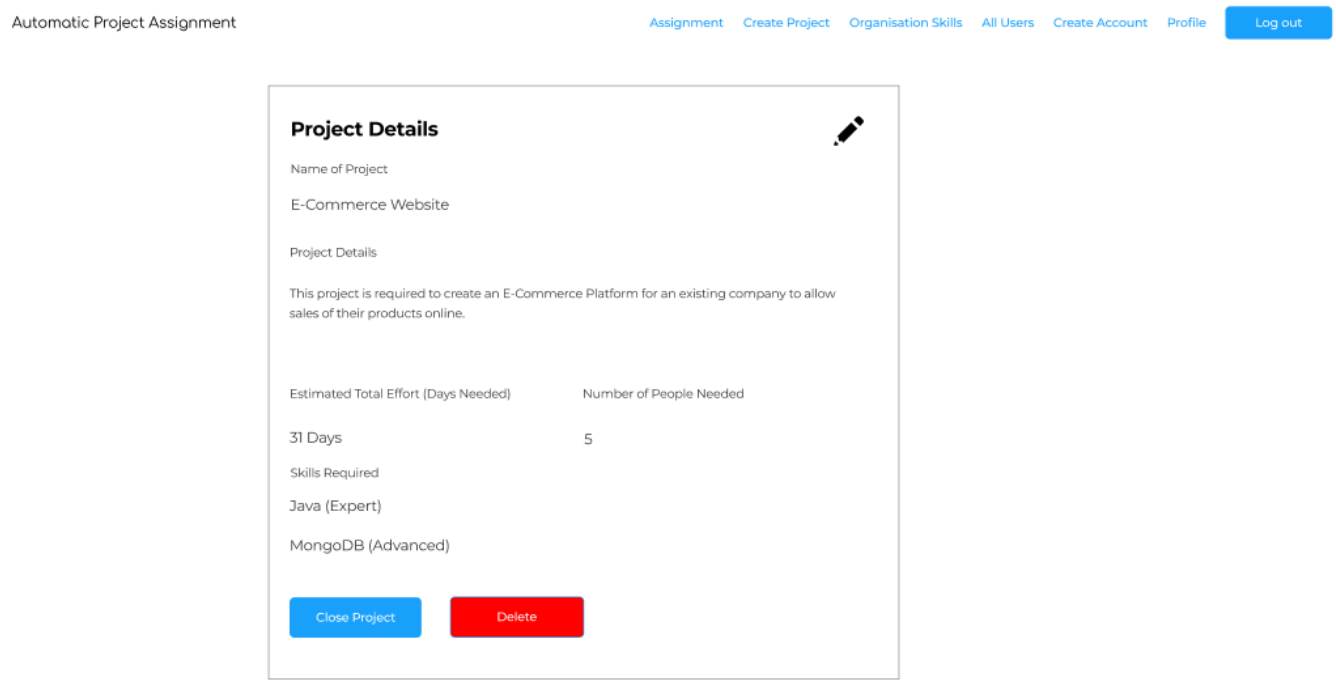


Figure 8, View Project Listing Page

In Figure 8, this page allows the project admin to view a project's details, which allows them to edit, close, or delete it. By closing the project, it would not be allowed for selection for the automation assignment process. To edit the project listing, Project Admin will need to click on the "Pencil" icon to start editing the project details. Click on the "Delete" button to delete the project listing.

5.1.8 Project Admin Organisation Skills Page

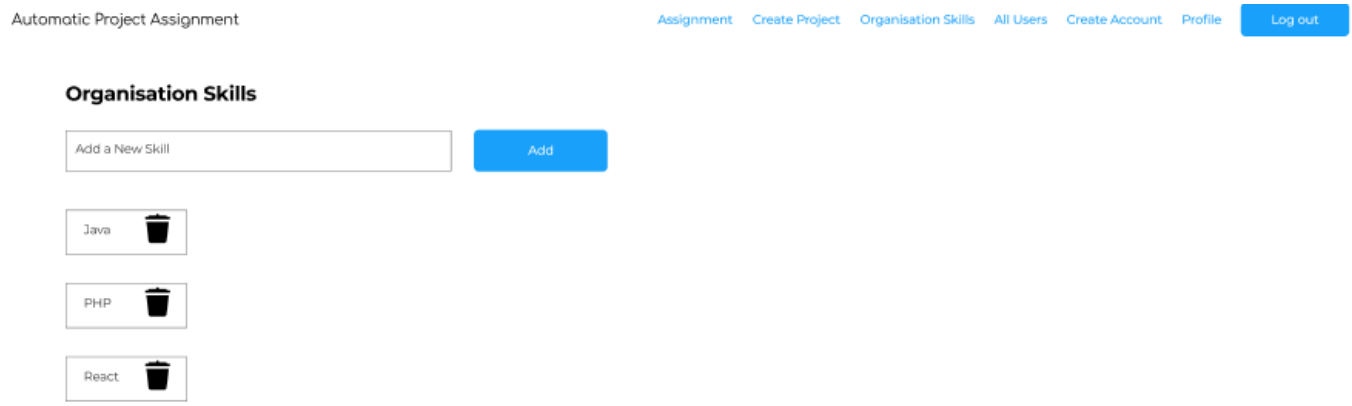


Figure 9, Organisation Skills Page

In Figure 9, this page allows Project Admin to manage the Organisation skills their organisation have which will allow employees to input their skill competency levels of such skills. Project Admin will have to input the Skill name and click on the “Add” button to add a new skill. For skills that they would like to remove, they will have to click the “Dustbin” icon beside the skill they would like to remove.

5.1.9 Project Admin All Users Page

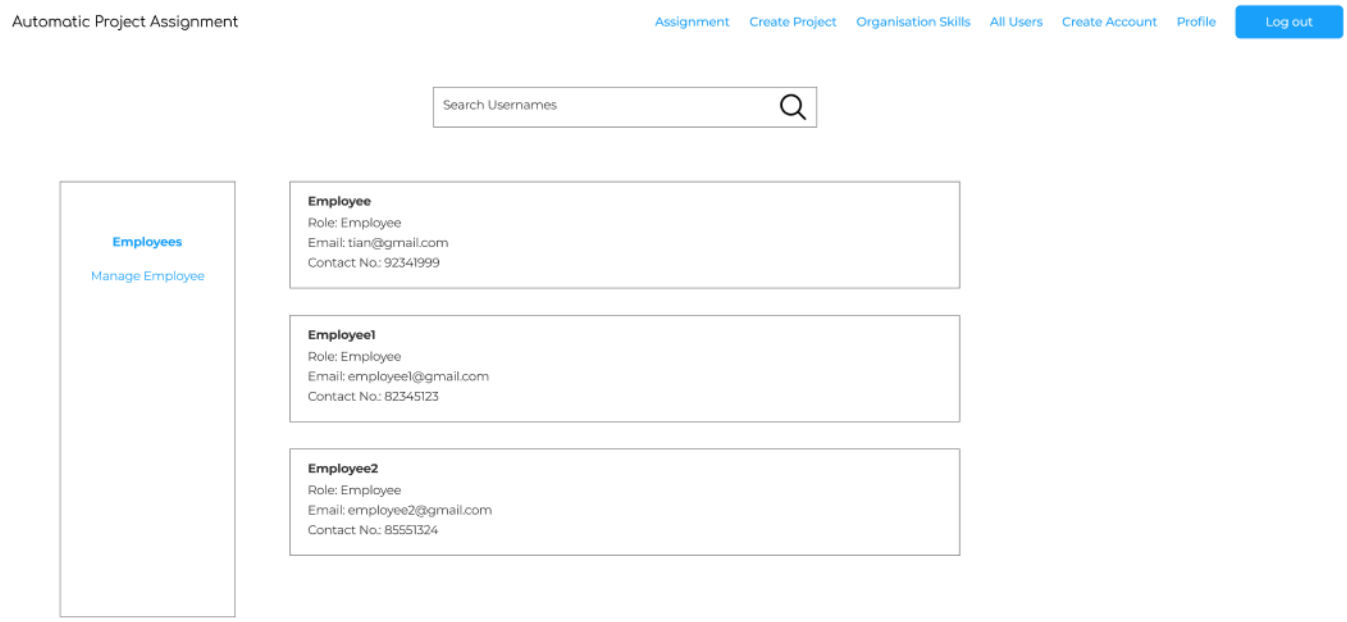


Figure 10, All Users Page

In Figure 10, this page allows Project Admins to view the list of all employees in their organisation. They will be able to search for the user immediately upon entering the details in the search bar. They would also be able to see the user’s details upon clicking on them. By clicking on “Manage Users” at the side navigation bar, it will show the project admin the view based on Figure 12 below.

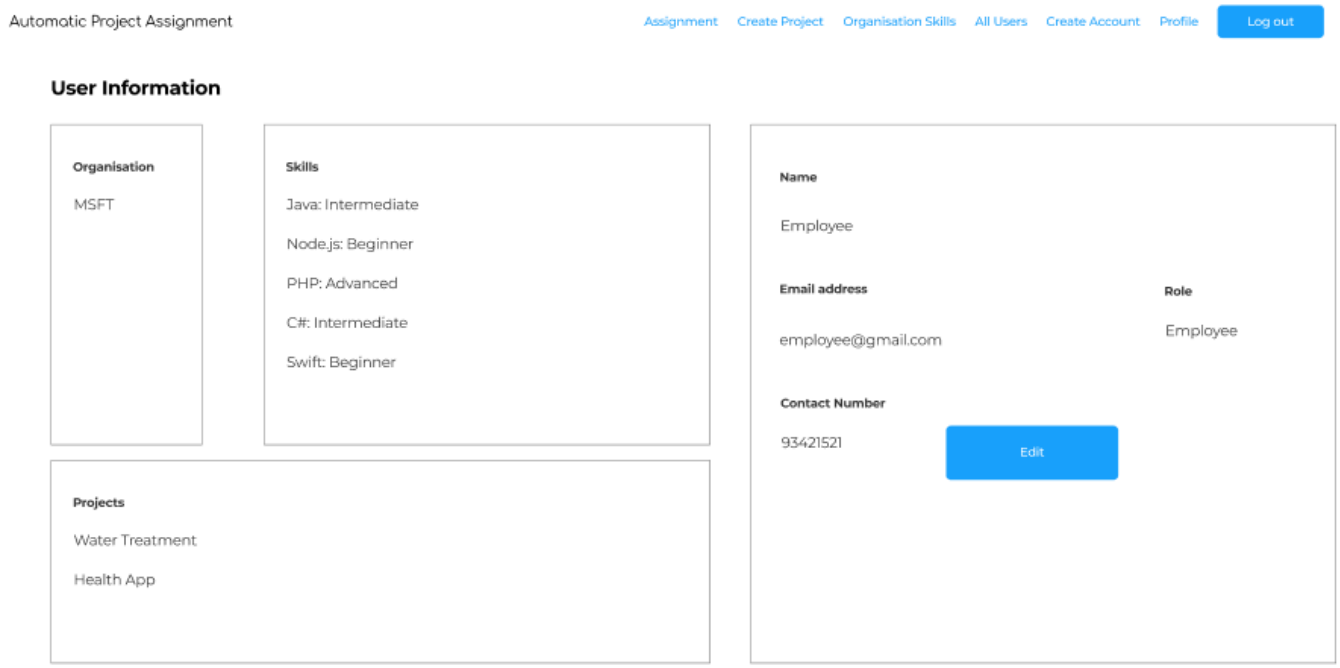


Figure 11, User Information Page

In Figure 11, the project admin will be able to see the selected user information after clicking on the desired user from Figure 10.

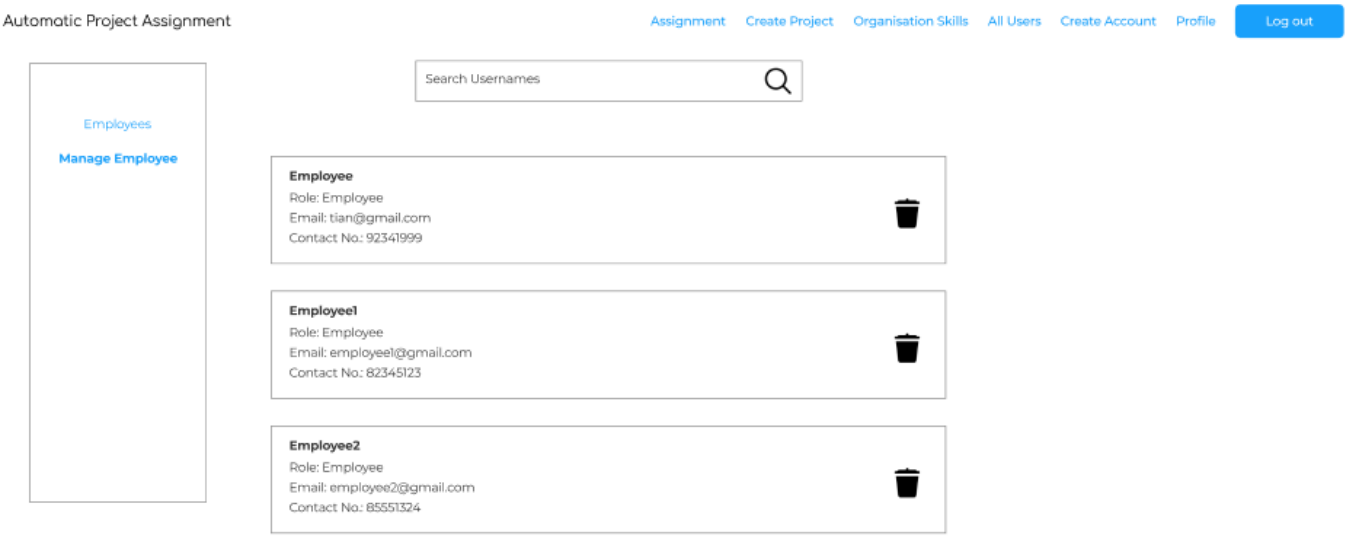


Figure 12, Manage Users Page

By clicking on the “Dustbin” icon, the project admin will be able to delete the desired user from the database.

5.1.10 Project Admin Create Account Page

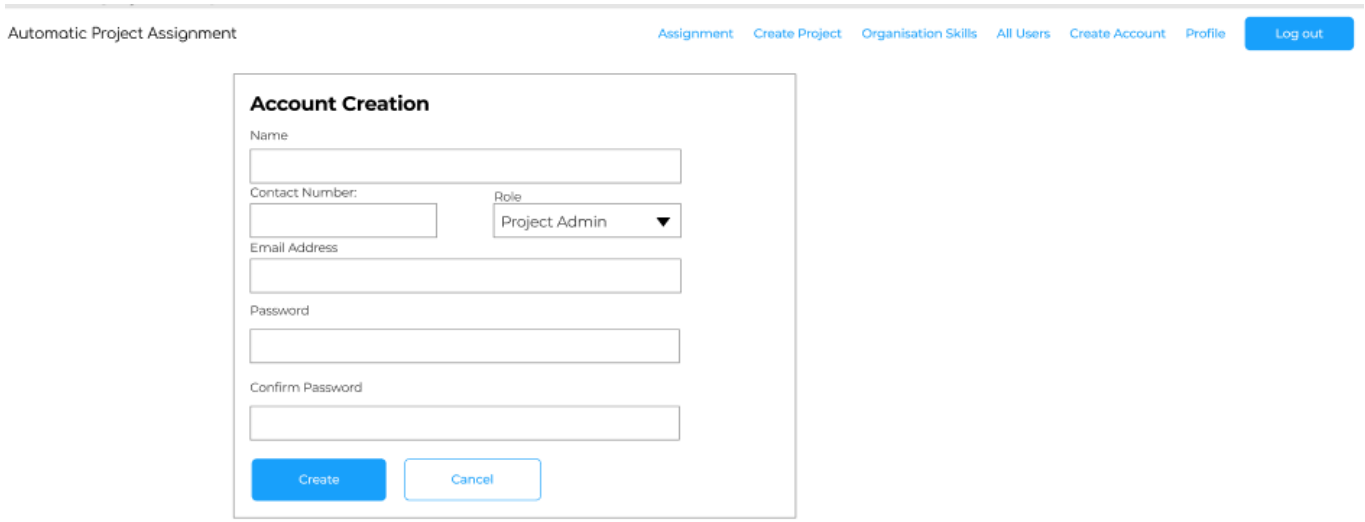


Figure 13, Create Account Page

In Figure 12, this page allows the Project Admin to create a new account for their organisation. They will be required to fill in the fields and select the role of the new account. By clicking the “Create” button, the new account will be created once it passes the validation of any existing accounts and matching passwords.

5.1.11 Project Admin Profile Page

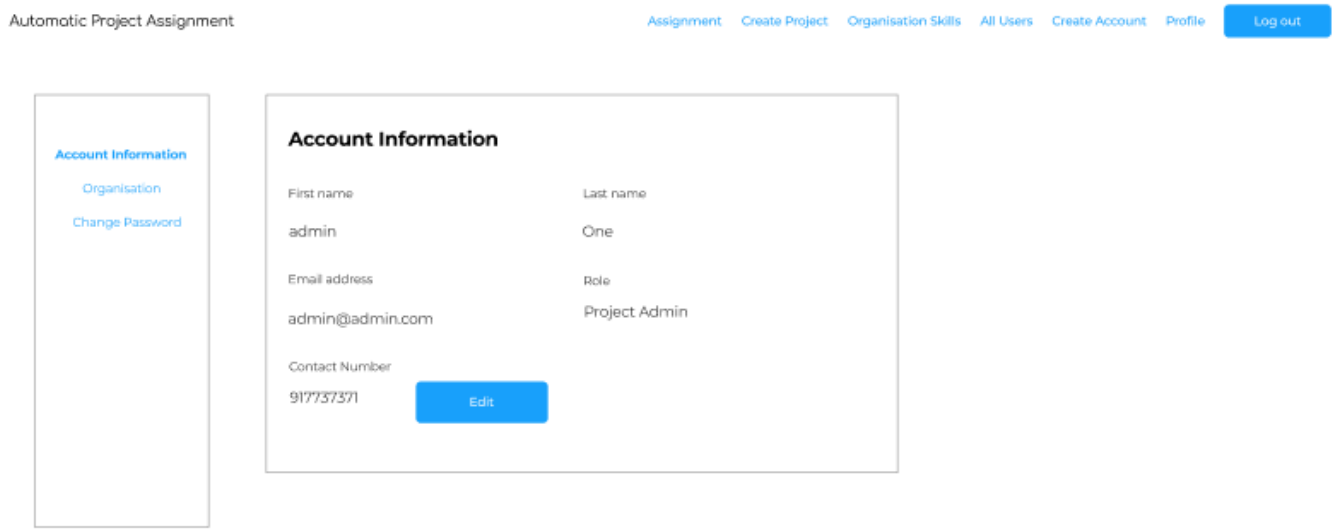


Figure 14, Profile Page

In Figure 14, in this page, the project admin will be able to manage his/her details. They will be able to edit their

contact number by clicking on the “Edit” button. If the project admin would like to change his/her password, they will have to click on “Change Password” from the side navigation bar.

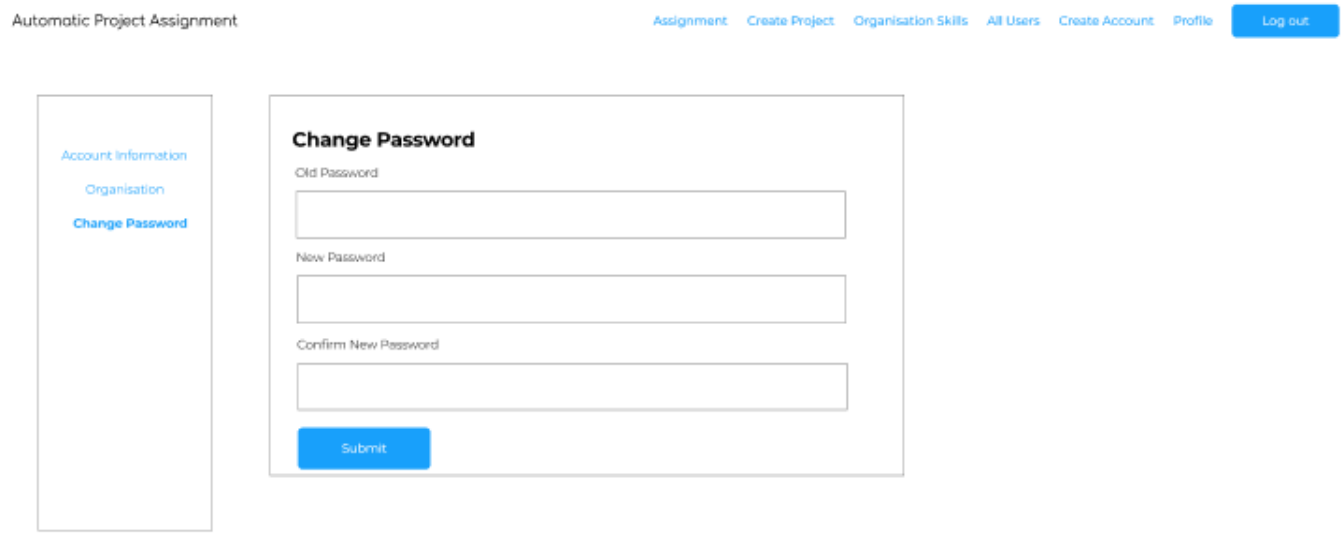


Figure 15, Change Password Page

In Figure 15, the project admin will be able to change their password by clicking on the “Submit” button after filling in the fields.

Employee UI Designs

5.1.12 Employee Main/Project Listings Page

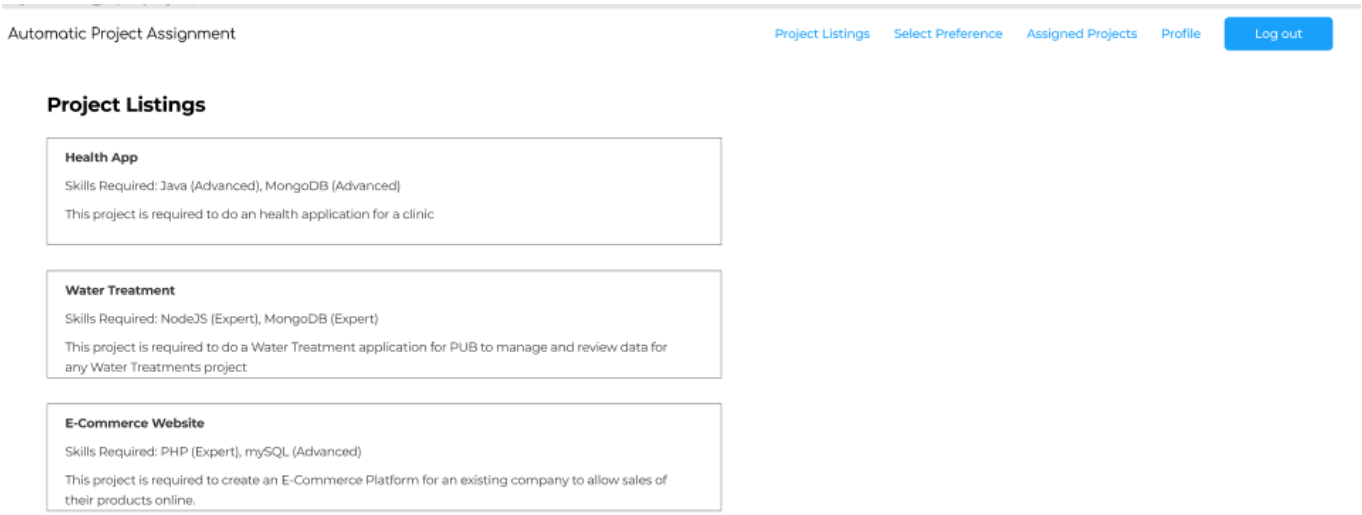


Figure 16, Employee Main/Project Listing Page

In Figure 16, on this page, employees will be able to see the project listings that are available for them to indicate their choices for the assignment period. The project listing is also the main page that the employee will see once

they are logged in. They will be able to log out from this page.

5.1.13 Employee View Project Details Page

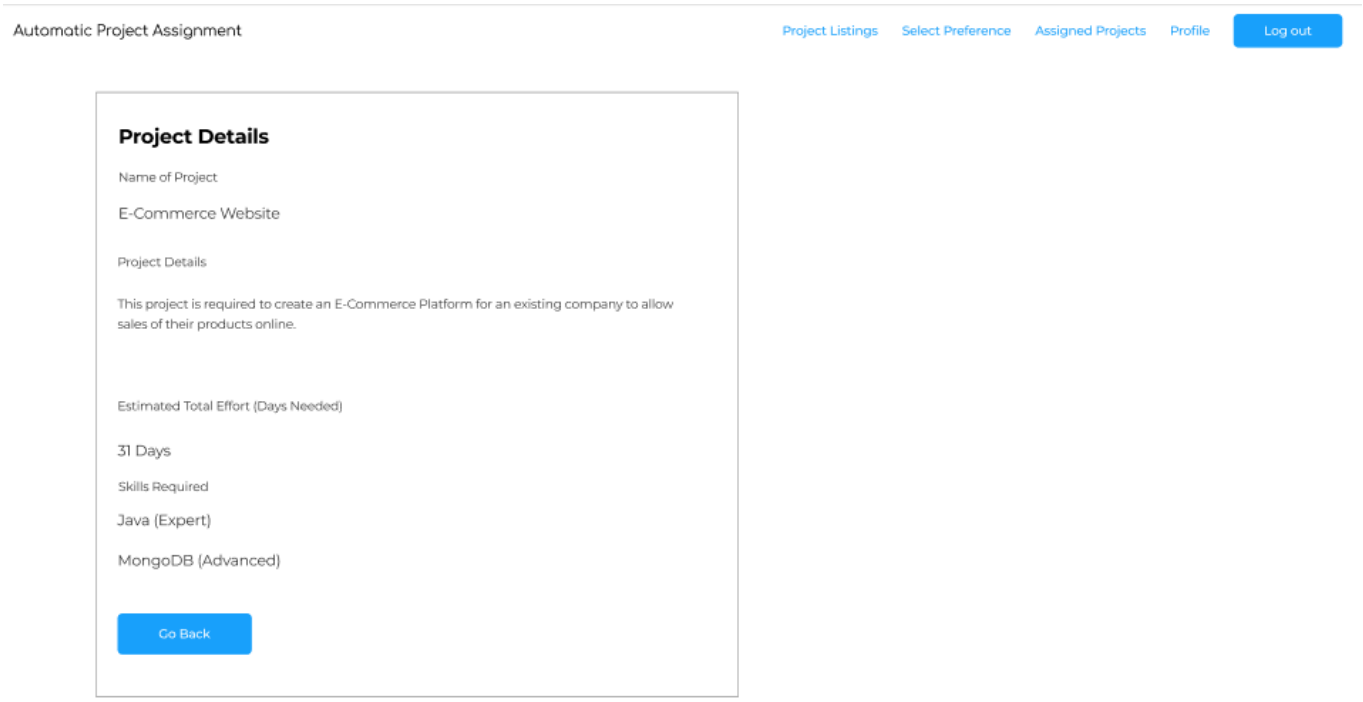


Figure 17, Employee View Project Details Page

In Figure 17, this page will allow the employee to view the details of the project. They will be able to return to the project listing page by clicking on the “Go Back” button.

5.1.14 Employee Select Preference Page

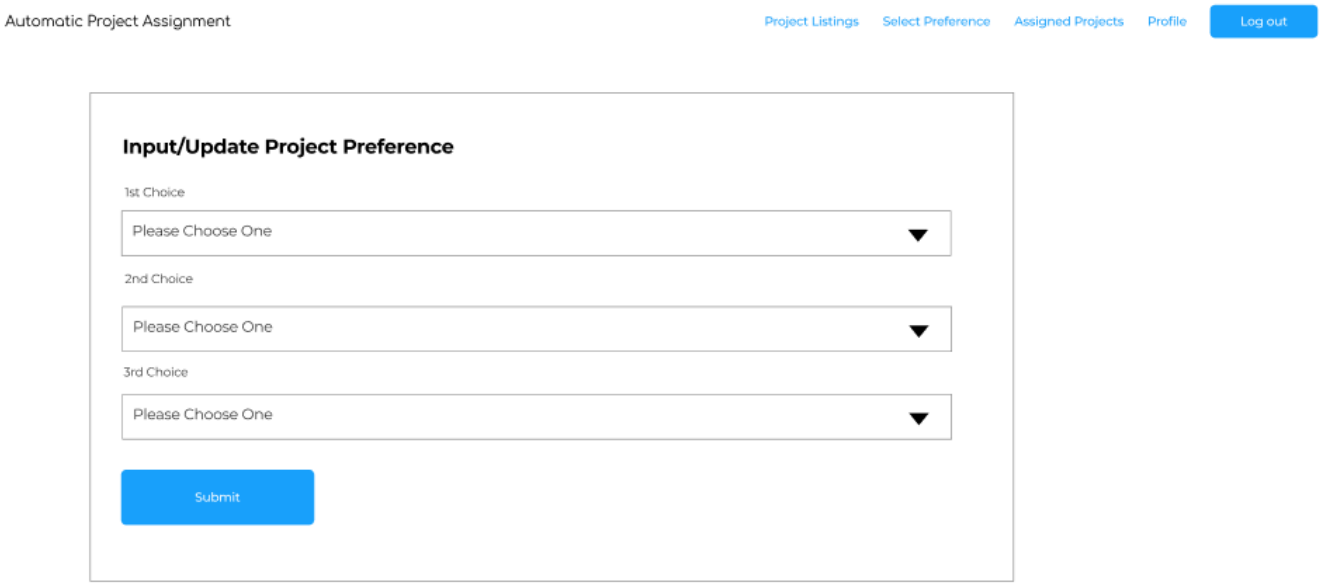


Figure 18, Employee Select Preference Page

In Figure 18, this page allows employees to choose or update their Project preference for the current round of project allocation they are allocated in. After selecting from the dropdown list for their preferred projects, they have to click on the “Submit” button to update their selection.

5.1.15 Employee Assigned Projects Page

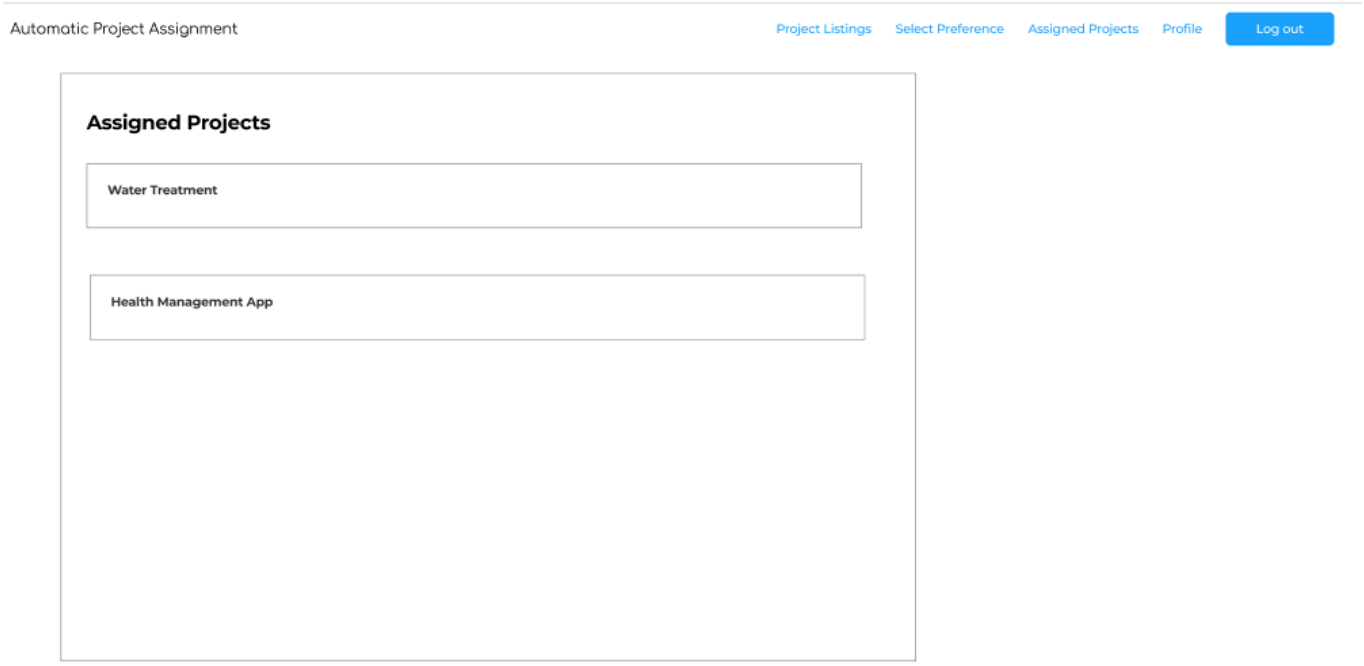


Figure 19, Project Assigned Page

In Figure 19, this page allows Employees to view the projects they are assigned to after the automation process.

5.1.16 Employee Personal Profile Page

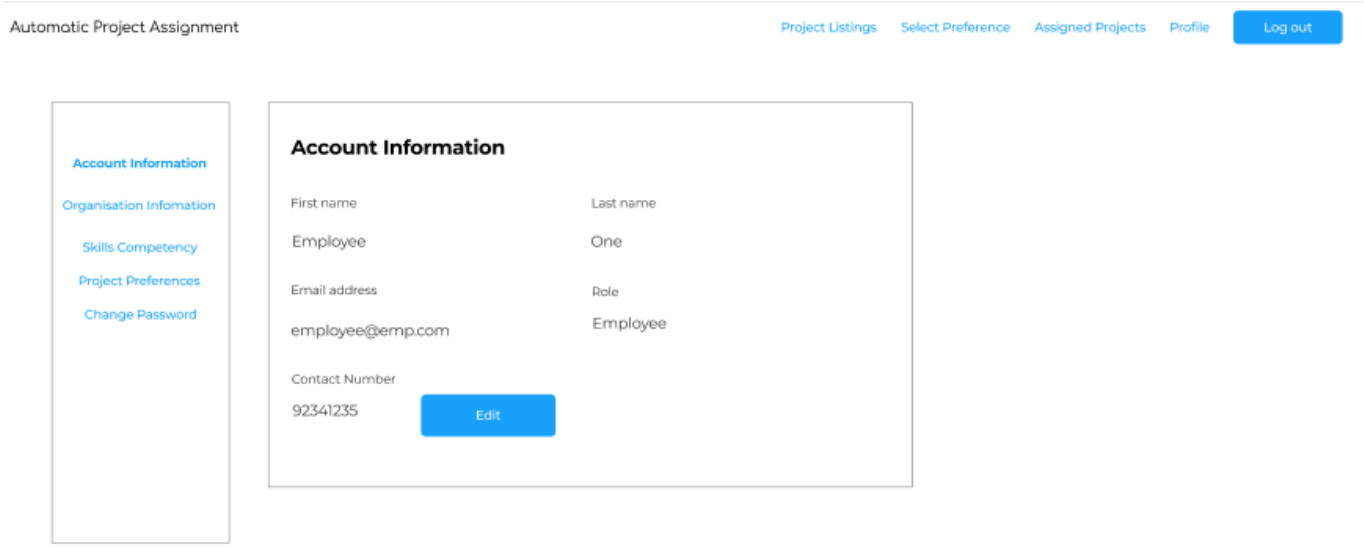


Figure 20, Employee Personal Profile Page

In Figure 20, this page allows employees to view their account information as well as their skills competency, and project preferences as well as change their password. By clicking on the “Edit” button, the Employee will be able to edit their contact information.

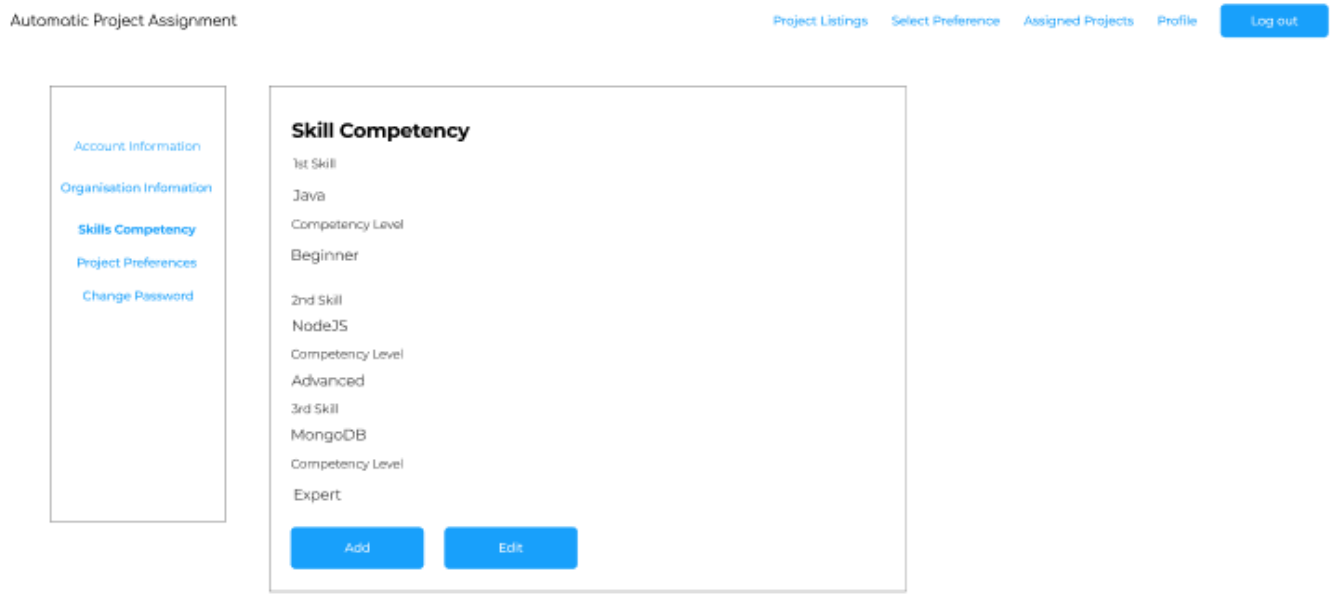


Figure 21, Skill Competency Page

In Figure 21, for this page, the employee will be able to add or edit the skill competency level they have. They will be allowed to choose the set of skills added by their organisation’s project admin and input their competency level for such skills.

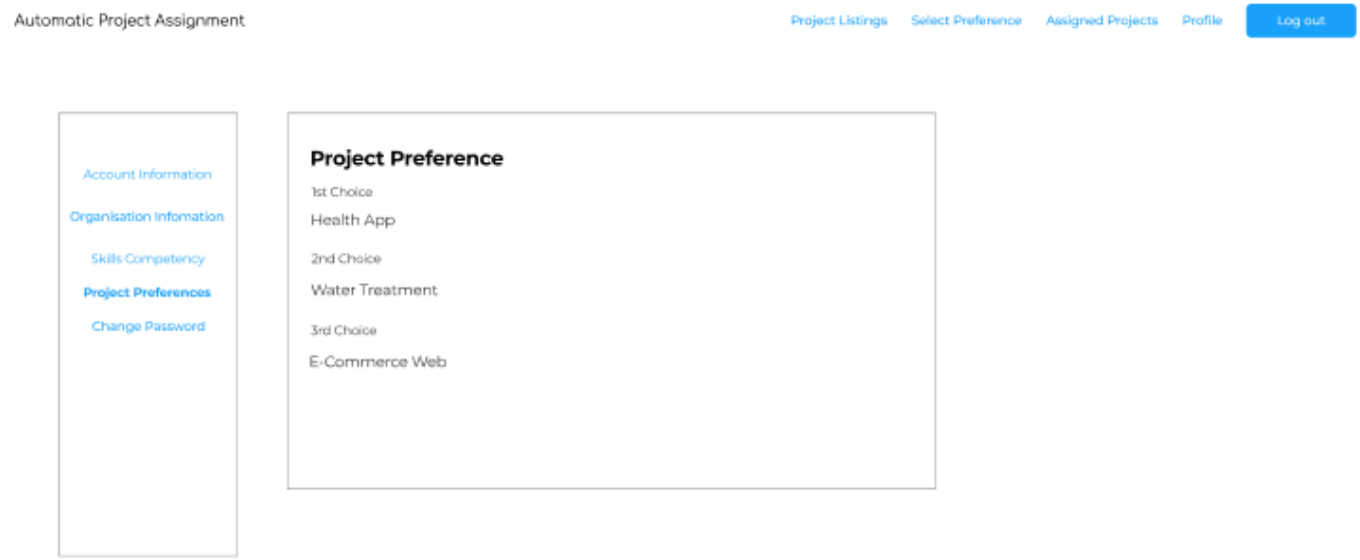


Figure 22, Project Preference Page

In Figure 22, the employee will be able to see the project preference they have indicated. To update/add their project preference, they can select the “Select Preference” button on the top navigation bar to do so.

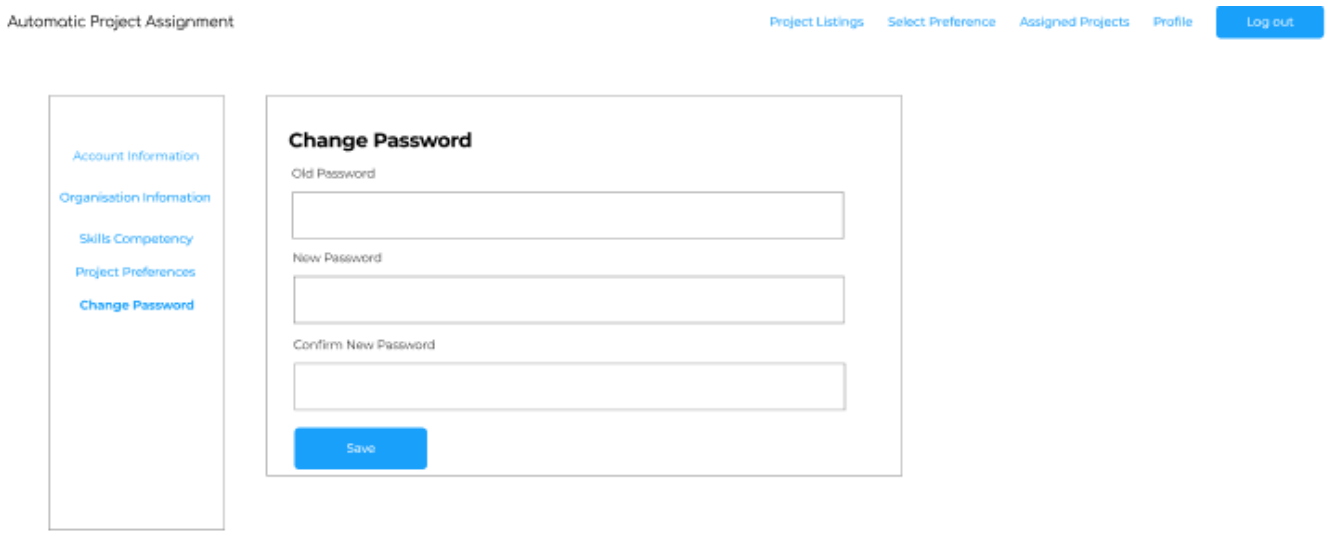


Figure 23, Change Password Page

In Figure 23, the employee will be able to change their password by clicking on the “Submit” button after filling in the fields.

Super Admin UI Designs

5.1.17 Super Admin Main/Organisation Listing Page

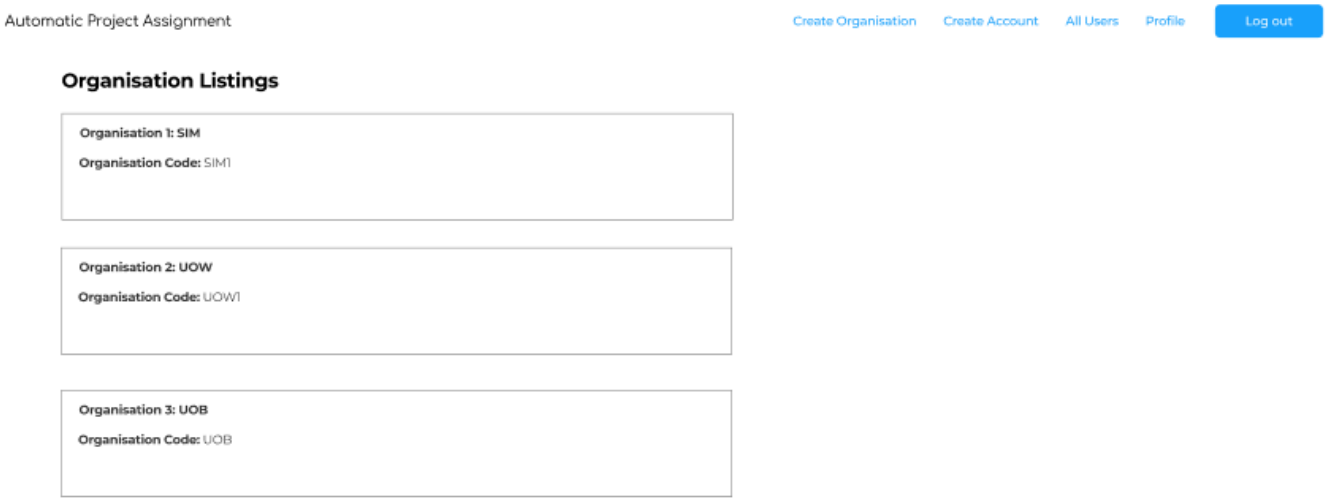


Figure 24, Super Admin Main/Organisation Page

In Figure 24, on this page, Super Admins will be able to see the organisation listings of all the organisations created on this website. The organisation listing is also the main page of what the Super Admin will first see once they are logged in. They will be able to logout from this page.

5.1.18 Super Admin Organisation Details Page

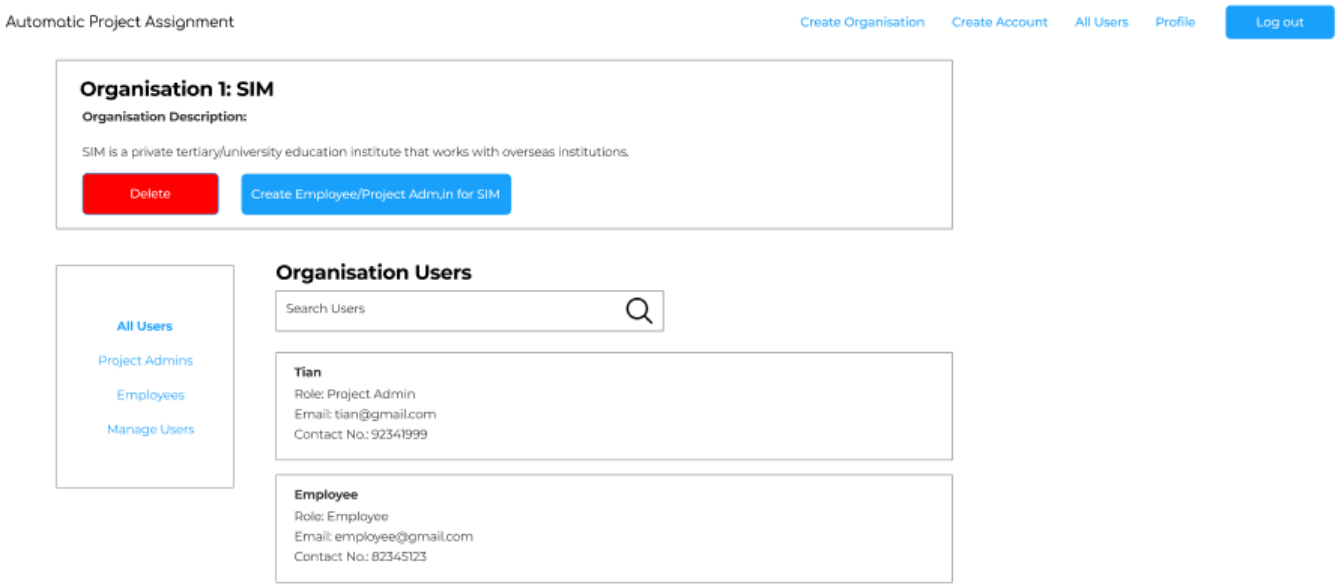


Figure 25, Organisation Details Page

In Figure 25, in this page, Super Admins will be able to see the users belonging to the selected organisation from Figure 24. The Super Admin will be able to filter by the user's role by clicking on "Project Admins" or "Employees" on the side navigation bar. The users can be searched using the search bar as well. By clicking on "Create Employee/Project Admin for SIM", it will redirect the Super Admin to the account creation page shown in figure 26 below. Super Admin can also see the details of a selected account and it will be shown in the figure 31 below.

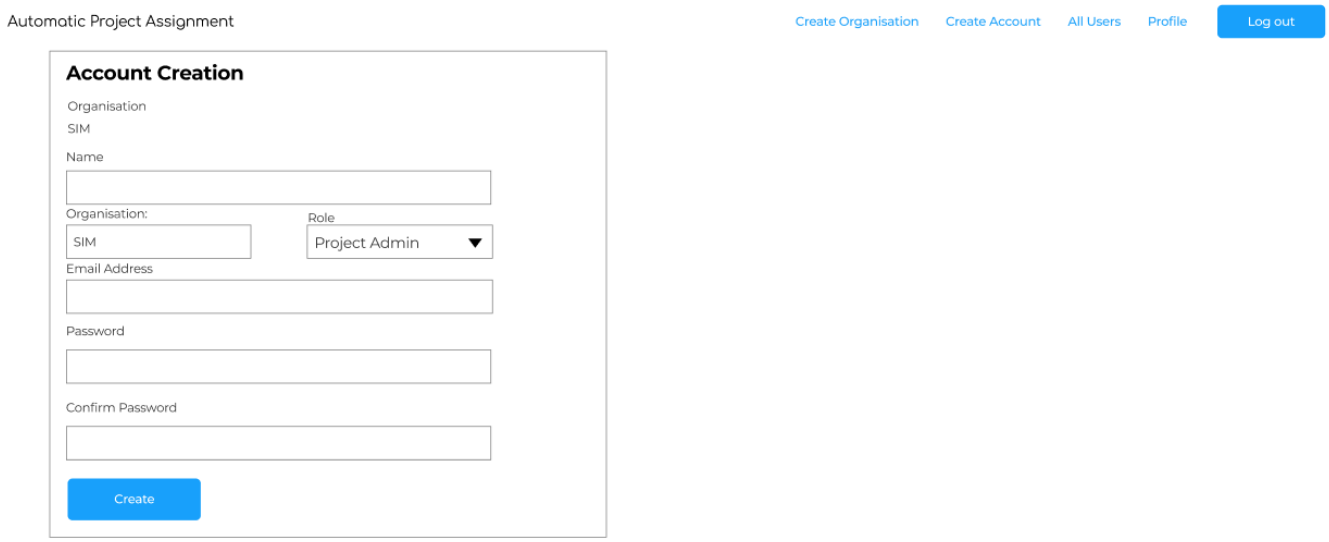


Figure 26, Account Creation Page via Organisation Details

In Figure 26, in this page, the Super Admin will be required to key in the details of the new user account in order to create a new account for the organisation. The Organisation field will be auto filled if the Super Admin was

redirected from Figure 25. By clicking on the “Create” button, the new user account will be created, taking in all the information keyed in the fields.

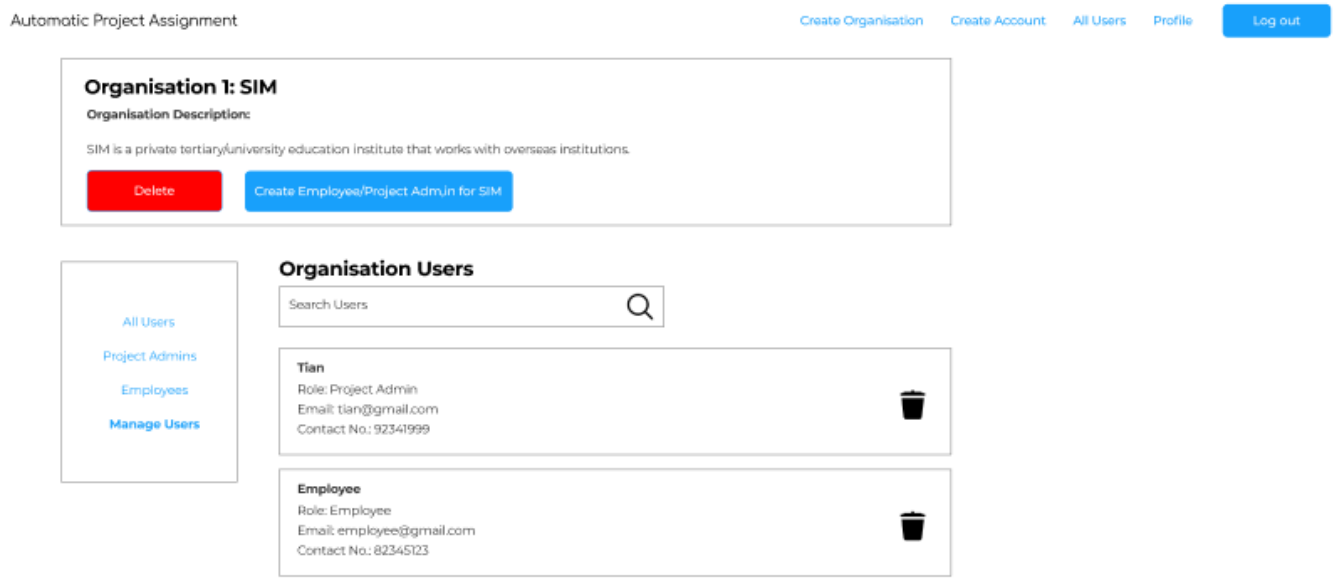


Figure 27, Manage Accounts via Organisation Details

In Figure 27, on this page, the Super Admin will be able to delete the users by clicking on the “Manage Users” from the side navigation bar. By clicking on the “Dustbin” button of the account that Super Admin wish to remove, it will remove the selected account from the database.

5.1.19 Super Admin Create New Organisation Listing Page

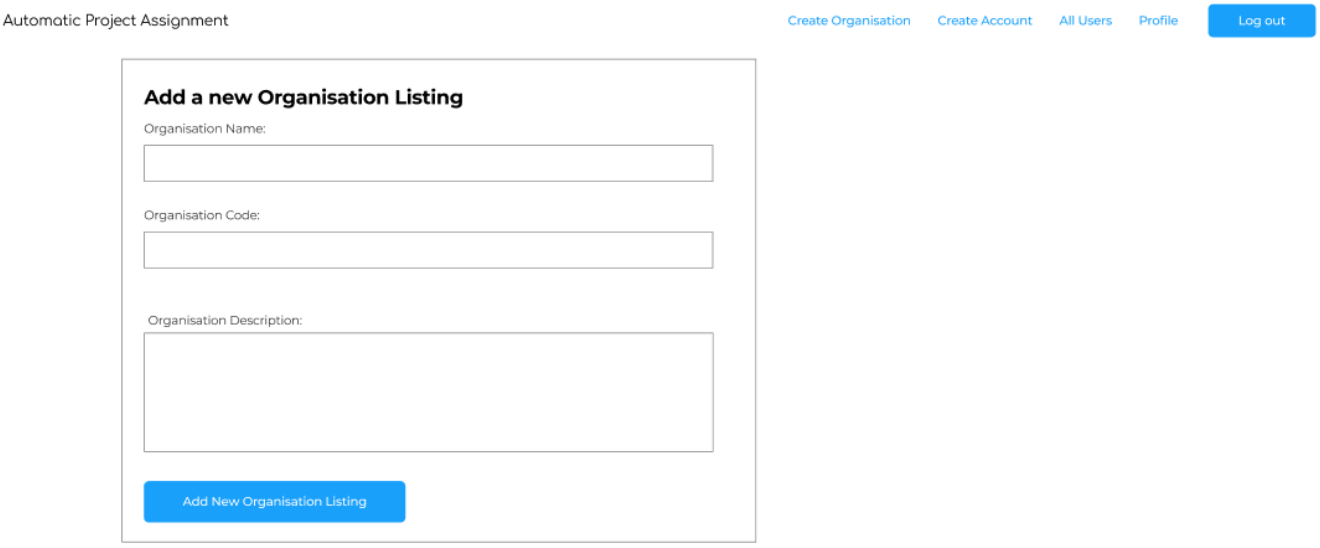


Figure 28, Create New Organisation Listing Page

In Figure 28, the Super Admin will be able to create a new organisation listing in the database. The super admin will be required to fill in all the relevant datas in order to create a new organisation listing. Clicking on “Add New Organisation Listing” button, a new organisation listing will be created with the information entered by the Super

Admin.

5.1.20 Super Admin Create Account Page

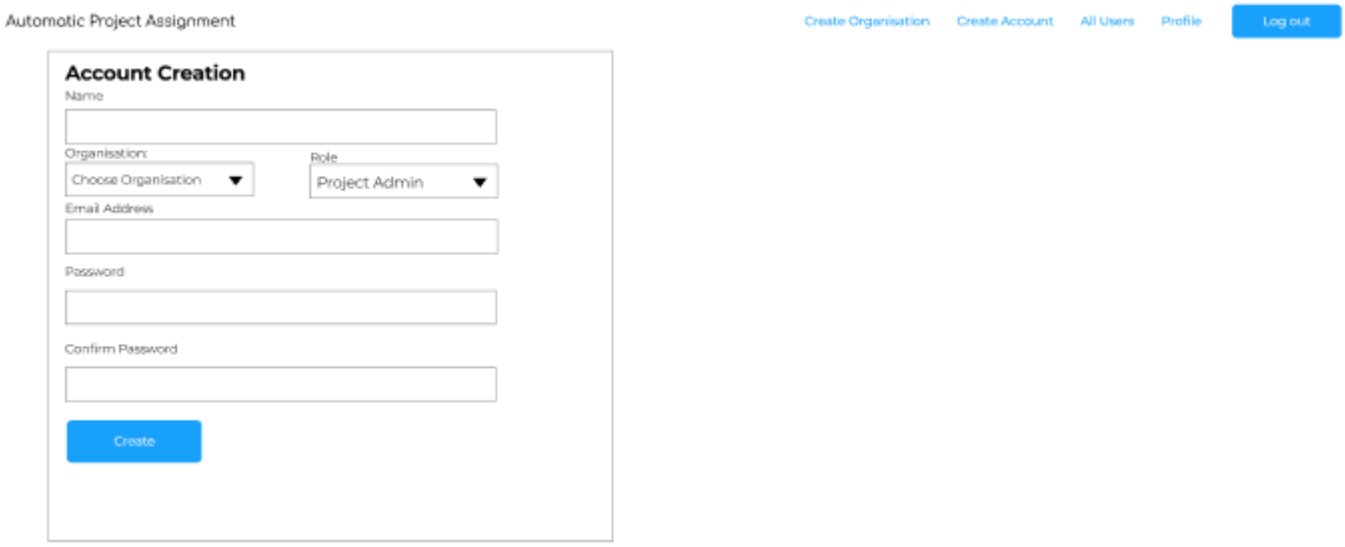


Figure 29, Create Account Page

As shown in Figure 29, this page is relatively similar to Figure 26, however for the Organisation field, the Super Admin will have to select the organisation that the new Account belongs to. For the creation of Super Admin user role accounts, the organisation field would be undefined as Super Admins do not belong to any organisation. After filling in all the details of the new user account, Super Admin will have to click on the “Create” button and the account will be created with the information entered in the field.

5.1.21 Super Admin View All Users Page

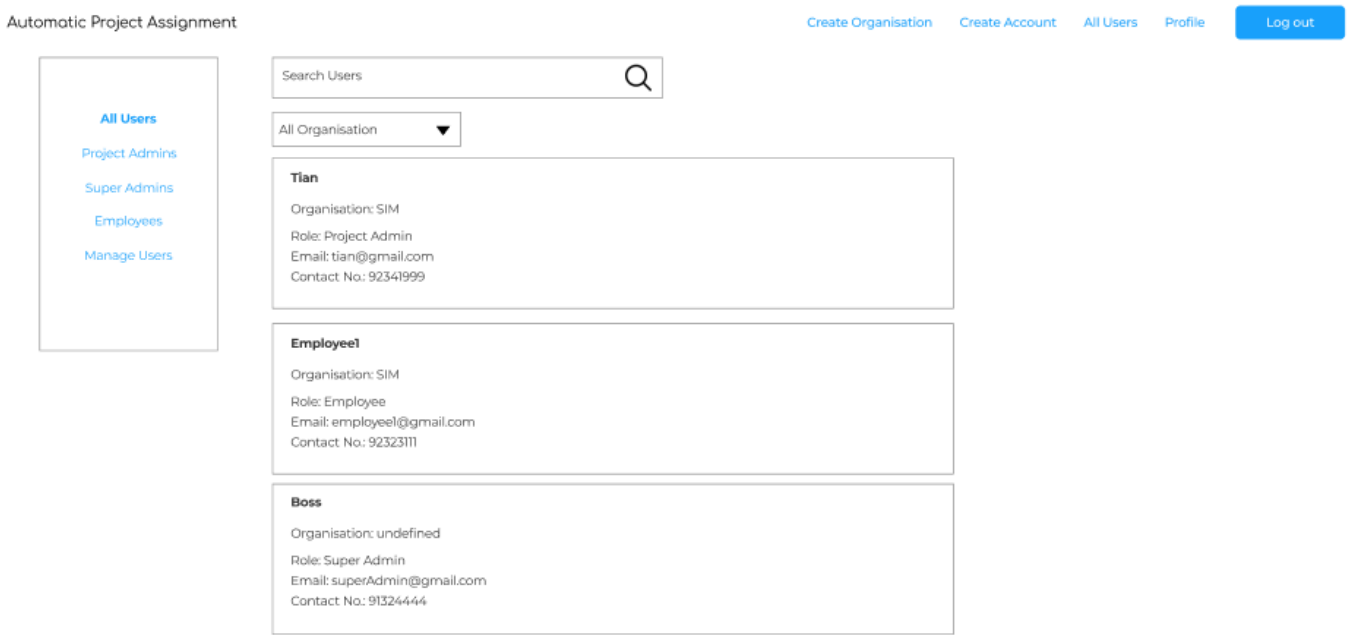


Figure 30, View All Users Page

In Figure 30, for this page, Super Admins will be able to view all the users using the website. Super Admin will be able to filter users by their roles as well as their organisation. The user can also be filtered using the search bar by entering their details.

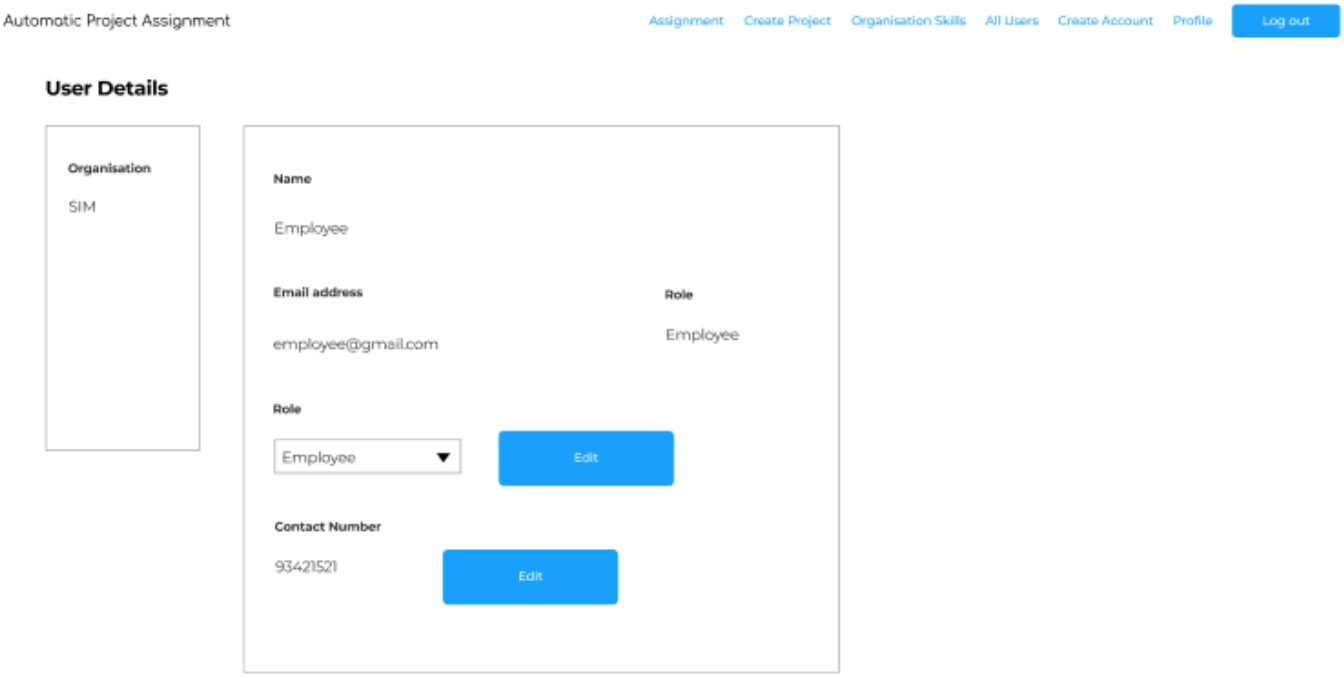


Figure 31, User Details via Organisation Details

In Figure 31, on this page, it will display the User Details of the selected user. Super Admin will be able to edit

the role by clicking on the “Edit” button beside the role. Likewise, for the contact info, the super admin will have to click on the “Edit” button beside the contact number to edit the contact information.

5.1.22 Super Admin User Profile Page

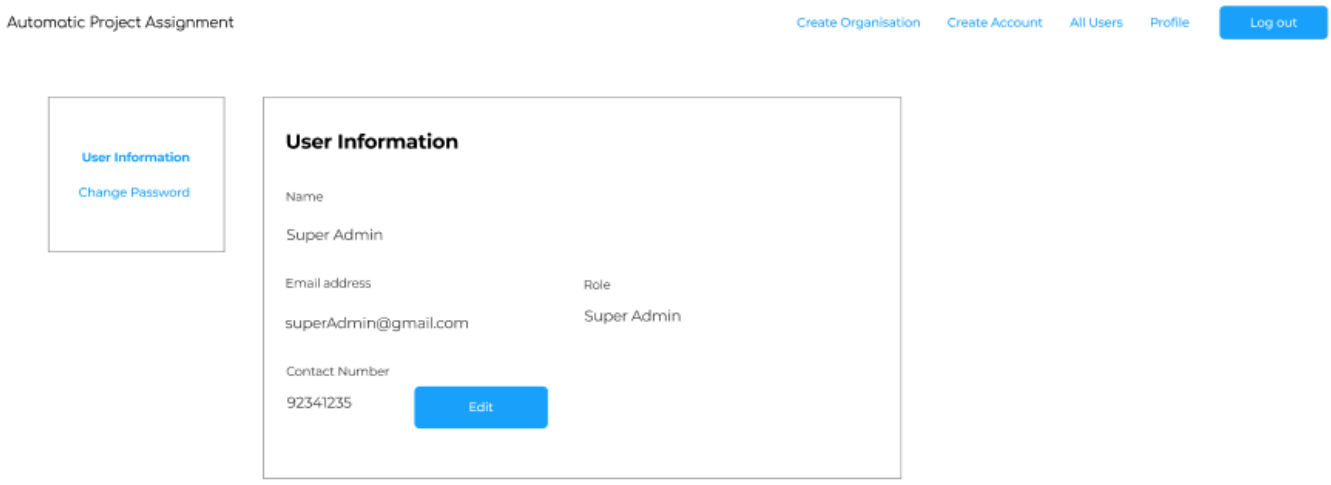


Figure 32, User Information Page

In Figure 32, on this page, Super Admin will be able to view their user information as well as edit their contact number. They can edit their contact number by clicking on the “Edit” button. They will be able to change their password as well by clicking on the “Change Password” from the side navigation bar shown in figure 33 below.

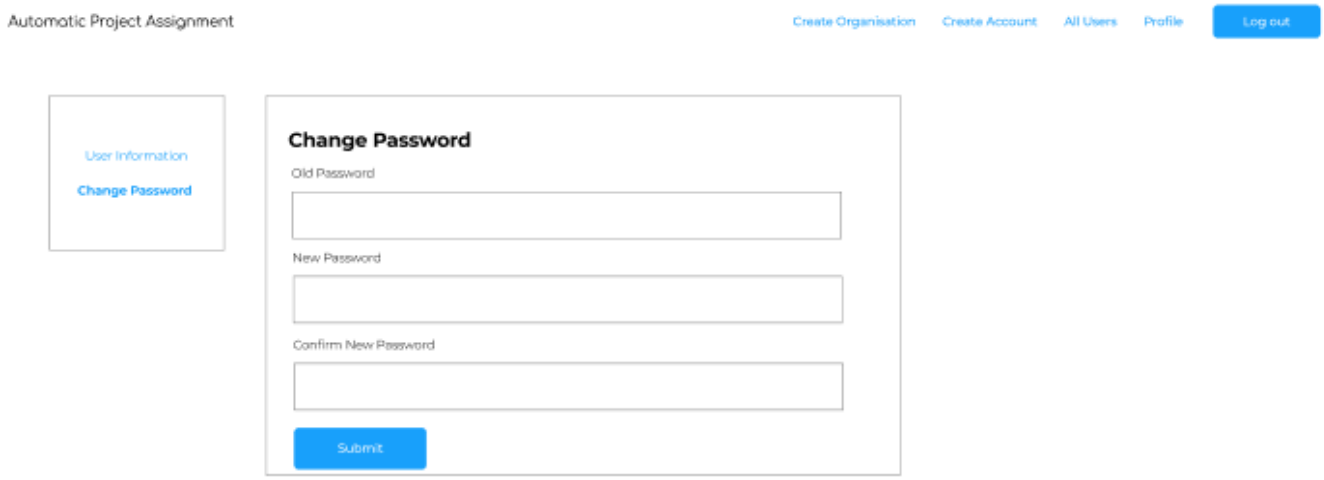


Figure 33, Change Password Page

In Figure 33, Super Admin will be able to change their password by clicking on the “Submit” button after filling in the fields.

5.2 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 log in to the system	
Normal flow: 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 brings the user to the homepage	
Sub-flows: None	
Alternative/Exceptional flows: 4.a The system prompts the user for invalid login credentials	

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 has logged out the system	
Trigger: The Project Admin wants to log out of the system	
Normal flow: 1. The user clicks on the log-out button 2. The system verifies the log-out request 3. 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	
Normal flow: 1. The user clicks on their profile 2. The system brings the user to their profile page 3. The user clicks on the edit button 4. The user adds in their contact information and clicks the submit button 5. The system verifies the request from the user 6. The system saves the information and prompts the user that the contact info is updated	
Sub-flows: None	
Alternative/Exceptional flows: 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.	

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: 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 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	
Normal flow: 1. The user clicks on their profile icon 2. The system brings the user to their profile page 3. The user clicks on the edit button 4. The user updates their information and clicks on the submit button 5. The system verifies the request from the user 6. The system saves the information and prompts the user that the contact info is updated	
Sub-flows: None	
Alternative/Exceptional flows: 5.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	

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	
Normal flow: 1. The user clicks on Create Project 2. The system brings the user to the create project page 3. The user enters the relevant information 4. The user adds in the relevant skills 5. The user clicks on the add new project listing button 6. The system verifies the request from the user 7. The system adds the new project and brings the user to the home page	
Sub-flows: None	
Alternative/Exceptional flows: 6.a The system fails to verify the request from the user, information was not saved and the system prompts the user to fill in all fields	

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	
Normal flow: 1. The user clicks on the name of the desired project 2. The system brings the user to the relevant project page 3. The user clicks on the edit icon 4. The user updates the project detail information and clicks the save button 5. The system verifies the request from the user 6. The system saves the information and prompts the user that the save attempt was successful 7. The system brings the user back to the relevant project page	
Sub-flows: None	
Alternative/Exceptional flows: 5.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	

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 project	
Trigger: The Project Admin wants to set a threshold for the project	
Normal flow: 1. The user click on Assignment 2. The system brings the user to the Assignment 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 employees required for the project	
Description: The Project Admin sets the number of employees 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 employees required for the chosen project	
Trigger: The Project Admin wants to set a number of employee for the chosen project	
Normal flow: 1. The user clicks on the name of the desired project 2. The system brings the user to the relevant project page 3. The user clicks on the edit icon 4. The user enters a number under the number of employees required section and clicks the edit project listing button 5. The system verifies the request from the user 6. The system saves the information and prompts the user that the save attempt was successful 7. The system brings the user back to the relevant project page	
Sub-flows: None	
Alternative/Exceptional flows: 5.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	

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	
Normal flow: 1. The user clicks on the name of the desired project 2. The system brings the user to the relevant project page 3. The user clicks on the delete button 4. The system verifies the users' requests 5. The project is removed and the system brings the user back to the home page	
Sub-flows: None	
Alternative/Exceptional flows: 4.a The system fails to verify the request from the user, prompts the user that an error has occurred	

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: 1. The user clicks on All Users 2. The system brings the user to all users page 3. The user clicks on the account name of the desired user 4. The system brings the user to the profile page of the chosen user 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	
Normal flow: 1. The user click on Assignment 2. The system brings the user to the Assignment page 3. The user clicks on the desired assignment 4. The system brings the user to the relevant assignment page 5. The user clicks on the process automatic assignment button 6. The system verifies the request from the user 7. The system prompts the user that the automatic assignment has been processed	
Sub-flows: None	
Alternative/Exceptional flows: 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 assignment page	

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: 1. The user clicks on the view statistics link on the side panel 2. The user views the result of the automated assignment process	
Sub-flows: None	
Alternative/Exceptional flows: None	

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 log into the system	
Normal flow: 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 brings the user to the homepage	
Sub-flows: None	
Alternative/Exceptional flows: 4.a The system prompts the user for invalid login credentials	

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: The user is logged out of the system.	
Trigger: The Employee wants to log out of the system	
Normal flow: 1. The user clicks on the log-out button 2. The system verifies the log-out request 3. The system brings the user back to the login page	
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	
Normal flow: 1. The user clicks on their profile 2. The system brings the user to their profile page 3. The user clicks on the edit button 4. The user adds in their contact information and clicks the submit button 5. The system verifies the request from the user 6. The system saves the information and prompts the user that the contact info updated	
Sub-flows: None	
Alternative/Exceptional flows: 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	

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: 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	
Normal flow: 1. The user clicks on their profile icon 2. The system brings the user to their profile page 3. The user clicks on the edit button 4. The user updates their information and clicks on the submit button 5. The system verifies the request from the user 6. The system saves the information and prompts the user that the contact info is updated	
Sub-flows: None	
Alternative/Exceptional flows: 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	

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	
Normal flow: 1. The user clicks on their profile icon 2. The system brings the user to their profile page 3. The user clicks on Skills from the side panel 4. The system brings the user to their skills page 5. The user clicks on the edit skills button 6. The user adds their skills and competency details and clicks on the submit button 7. The system verifies the request from the user and saves the information	
Sub-flows: None	
Alternative/Exceptional flows: 7.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	

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: 1. The user clicks on their profile icon 2. The system brings the user to their profile page 3. The user clicks on Skills in the side panel 4. The system brings the user to the skills section 5. 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	
Normal flow: 1. The user clicks on their profile icon 2. The system brings the user to their profile page 3. The user clicks on Skills from the side panel 4. The system brings the user to their skills page 5. The user clicks on the edit skills button 6. The user updates their skills and competency details and clicks on the submit button 7. The system verifies the request from the user and saves the information	
Sub-flows: None	
Alternative/Exceptional flows: 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	

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 exist	
Post-condition: User views the project listings	
Trigger: The Employee wants to view project listings	
Normal flow: 1. The user clicks on Project Listing 2. The system brings 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: 1. The user clicks on Project Listing 2. The system brings the user to the Project Listing page 3. The user clicks on the desired project link 4. The system brings the user to the corresponding project page 5. 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 exist	
Post-condition: User inputted their project preference	
Trigger: The Employee wants to input their project preference	
Normal flow: 1. The user clicks on select preference 2. The system brings the user to the select preference page 3. The user selects their first, second, and third preference 4. The user clicks on the submit selection button 5. The system verifies the users' requests and saves the users' selection 6. The system brings the user back to the project listing page	
Sub-flows: None	
Alternative/Exceptional flows: 5.a The system fails to verify the request from the user and prompts the user to ensure all fields are selected and that no projects are duplicated	

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: 1. The user clicks on the Assigned Projects 2. The system brings the user to their Assigned Projects page 3. The user views the result of the project assignment. They should be able to see the projects allocated to them.	
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 log into the system	
Normal flow: 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 brings the user to the homepage	
Sub-flows: None	
Alternative/Exceptional flows: 4.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 has logged out the system	
Trigger: The Super Admin wants to log out of the system	
Normal flow: 1. The user clicks on the log-out button 2. The system verifies the log-out request 3. The system brings the user back to the homepage	
Sub-flows: None	
Alternative/Exceptional flows: None	

Use case: Create Organisation codes as Super Admin	Taiga ID: #40
Stakeholders and goals: Super Admin - Create Organisation codes	
Description: The Super Admin creates Organisation codes	
Actors: Super Admin	
Pre-condition: User is logged in as Super Admin	
Post-condition: User has created the Organisation code	
Trigger: The Super Admin wants to create Organisation codes	
Normal flow: 1. The user clicks on Create Organisation 2. The system brings the user to the Create Organisation page 3. The user inputs the information required and clicks on the Add New Organisation Listing button 4. The system verifies the request from the user 5. The system creates the new organisation and brings the user back to the home page	
Sub-flows: None	
Alternative/Exceptional flows: 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 organisation code page and awaits user input	

Use case: Manage users in an Organisation as Super Admin	Taiga ID: #42
Stakeholders and goals: Super Admin - Manage users in an Organisation	
Description: The Super Admin manages users in an Organisation	
Actors: Super Admin	
Pre-condition: User is logged in as Super Admin and the chosen Organisation exists	
Post-condition: User has managed the users in an Organisation	
Trigger: The Super Admin wants to manage users in an Organisation	
Normal flow: 1. The user clicks the desired organisation name 2. The system brings the user to the corresponding organisation page 3. The user clicks on Manage Users	
Sub-flows: None	
Alternative/Exceptional flows: None	

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 Organisation	
Actors: Super Admin	
Pre-condition: User is logged in as Super Admin and the chosen Organisation exists	
Post-condition: User has created the user account	
Trigger: The Super Admin wants to create a user account	
Normal flow: 1. The user clicks on Create Account 2. The system brings the user to the sign up page 3. The user enters the relevant information and clicks on the sign up button 4. The system verifies the users' request 5. The prompts the user that the creation was successful	
Sub-flows: None	
Alternative/Exceptional flows: 8.a The system fails to verify the request from the user and prompts the user that an error has occurred	

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 Organisation	
Actors: Super Admin	
Pre-condition: User is logged in as Super Admin and the chosen Organisation exists	
Post-condition: User has viewed the user account for the chosen Organisation	
Trigger: The Super Admin wants to view the user account for the chosen Organisation	
Normal flow: 1. The user clicks on All users 2. The system brings the user to the all users page	
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 Organisation	
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 Organisation	
Trigger: The Super Admin wants to delete user account for the chosen Organisation	
Normal flow: 1. The user clicks on All users 2. The system brings the user to the all users page 3. The user click on the Manage Users on the side panel 4. The user clicks on the delete icon for the desired user 5. The system prompts a confirmation from the user 6. The user clicks on the ok button 7. The system verifies the request from the user 8. The system deletes the selected user.	
Sub-flows: None	
Alternative/Exceptional flows: 6.a The user clicks on the cancel button and nothing happens 7.a The system fails to verify the request from the user and prompts the user that the deletion was a failure	

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	
Normal flow: 1. The user clicks on Create Account 2. The system brings the user to the sign up page 3. The user inputs the information required and clicks sign up button 4. The system verifies the request from the user 5. The system prompts the user that the creation was successful	
Sub-flows: None	
Alternative/Exceptional flows: 4.a The system fails to verify the request from the user and prompts the user that an error has occurred	

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: 1. The user clicks on All users 2. The system brings the user to the all users page 3. The user click on the Manage Employees on the side panel 4. The user clicks on the delete icon for the desired user 5. The system prompts a confirmation from the user 6. The user clicks on the ok button 7. The system verifies the request from the user 8. The system deletes the selected user.	
Sub-flows: None	
Alternative/Exceptional flows: 6.a The user clicks on the cancel button and nothing happens 7.a The system fails to verify the request from the user and prompts the user that the deletion was a failure	

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 reset their password	
Trigger: The Employee wants to change and reset password	
Normal flow: 1. The user clicks on their profile icon 2. The system brings the user to their profile page 3. The user clicks on change password 4. The system brings the user to the change password page 5. The user enters their current password and the new password 6. The user clicks the submit button 7. The system verifies the request from the user 8. The system saves the changes made and prompts the user that the save was successful	
Sub-flows: None	
Alternative/Exceptional flows: 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 change password page and awaits user input	

Use case: Edit user role as Super Admin	Taiga ID: #291
Stakeholders and goals: Super Admin- edit user role	
Description: The Super Admin edit the role of selected user	
Actors: Super Admin	
Pre-condition: User is logged in as Super Admin and is at the all users page	
Post-condition: Role of the selected user has been edited	
Trigger: The super admin want to edit role of selected user	
Normal flow: 1. The user click on the account name of the desired user 2. The system brings the user to the corresponding user page 3. The user clicks on the Edit button under the Role section 4. The user selects the desired role for the corresponding account 5. The user clicks on the submit button 6. The system verifies the request from the user 7. The system prompts that the role is updated successfully	
Sub-flows: None	
Alternative/Exceptional flows: 6.a The system fails to verify the request from the user and prompts the user that an error has occured	

Use case: Edit user account contact info as Super Admin	Taiga ID: #292
Stakeholders and goals: Super Admin - edit user account contact info	
Description: The Super Admin edit the contact info of the selected user	
Actors: Super Admin	
Pre-condition: User is logged in as Super Admin and is at the view user page	
Post-condition: The contact info of the selected user has been updated	
Trigger: The Super Admin wants to edit contact info of desired user	
Normal flow: 1. The user clicks on the account name of the desired user 2. The system brings the user to the corresponding user page 3. The user clicks on the edit button under the contact info section 4. The user enter the relevant information 5. The user clicks on the save button 6. The system verifies the request from the user 7. The system saves the information entered	
Sub-flows: None	
Alternative/Exceptional flows: 6.a The system fails to verify the request from the user and prompts the user that the request was not saved	

Use case: Edit Employee account contact info as Project Admin	Taiga ID: #293
Stakeholders and goals: Project Admin - edit Employee contact info	
Description: The Project Admin edit the contact info of the selected Employee	
Actors: Project Admin	
Pre-condition: User is logged in as Project Admin and is at the all users page	
Post-condition: The contact info of the selected Employee has been updated	
Trigger: The Project Admin wants to edit contact info of desired Employee	
Normal flow: 1. The user entered the account name of the desired user 2. The user clicks on the account name of the desired user 3. The system brings the user to the corresponding user page 4. The user clicks on the edit button under the contact info section 5. The user enter the relevant information 6. The user clicks on the save button 7. The system verifies the request from the user 8. The system saves the information entered	
Sub-flows: None	
Alternative/Exceptional flows: 6.a The system fails to verify the request from the user and prompts the user that the request was not saved	

Use case: Add skills to organisation list as Project Admin	Taiga ID: #375
Stakeholders and goals: Project Admin - Add skill to organisation list	
Description: The Project Admin add skills to the organisation list	
Actors: Project Admin	
Pre-condition: User is logged in as Project Admin and adds skills to organisation list	
Post-condition: The skill has been added to the organisation list	
Trigger: The Project Admin wants add skill to the organisation list	
Normal flow: 1. The user clicks on Organisation Skills 2. The system brings the user to the organisation skills page 3. The user enters new skills information and clicks the add button 4. The system verifies the request from the user 5. The system adds the new skill to the organisation skill list	
Sub-flows: None	
Alternative/Exceptional flows: 5.a The system fails to verify the information entered and prompts the user that an error has occured	

Use case: Remove skills from the organisation list as Project Admin	Taiga ID: #376
Stakeholders and goals: Project Admin - remove skill from organisation list	
Description: The Project Admin remove skills from the organisation list	
Actors: Project Admin	
Pre-condition: User is logged in as Project Admin, removes skills from organisation list	
Post-condition: The skill has been removed from the organisation list	
Trigger: The Project Admin wants to remove skill from the organisation list	
Normal flow: 1. The user clicks on Organisation Skills 2. The system brings the user to the Organisation Skills page 3. The user clicks on the delete icon beside the desired selection 4. The system prompts a confirmation from the user 5. The user clicks on the ok button 6. The system verifies the request from the user 7. The system deletes the selected skill	
Sub-flows: None	
Alternative/Exceptional flows: 5.a The user clicks on the cancel button and nothing happens 6.a The system fails to verify the request from the user and prompts that an error has occurred.	

Use case: Creating assignments as Project Admin	Taiga ID: #602
Stakeholders and goals: Project Admin - creating assignments	
Description: The Project Admin creates an assignment	
Actors: Project Admin	
Pre-condition: User is logged in as Project Admin	
Post-condition: The assignment has been created	
Trigger: The Project Admin wants to create an assignment	
Normal flow: 1. The user clicks on Assignment 2. The system brings the user to the Assignment page 3. The user enters the relevant information under the Add a New Assignment section 4. The user clicks on the Add Assignment button 5. The system verifies the request from the user 6. The system adds the new assignment to the project assignment page	
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	

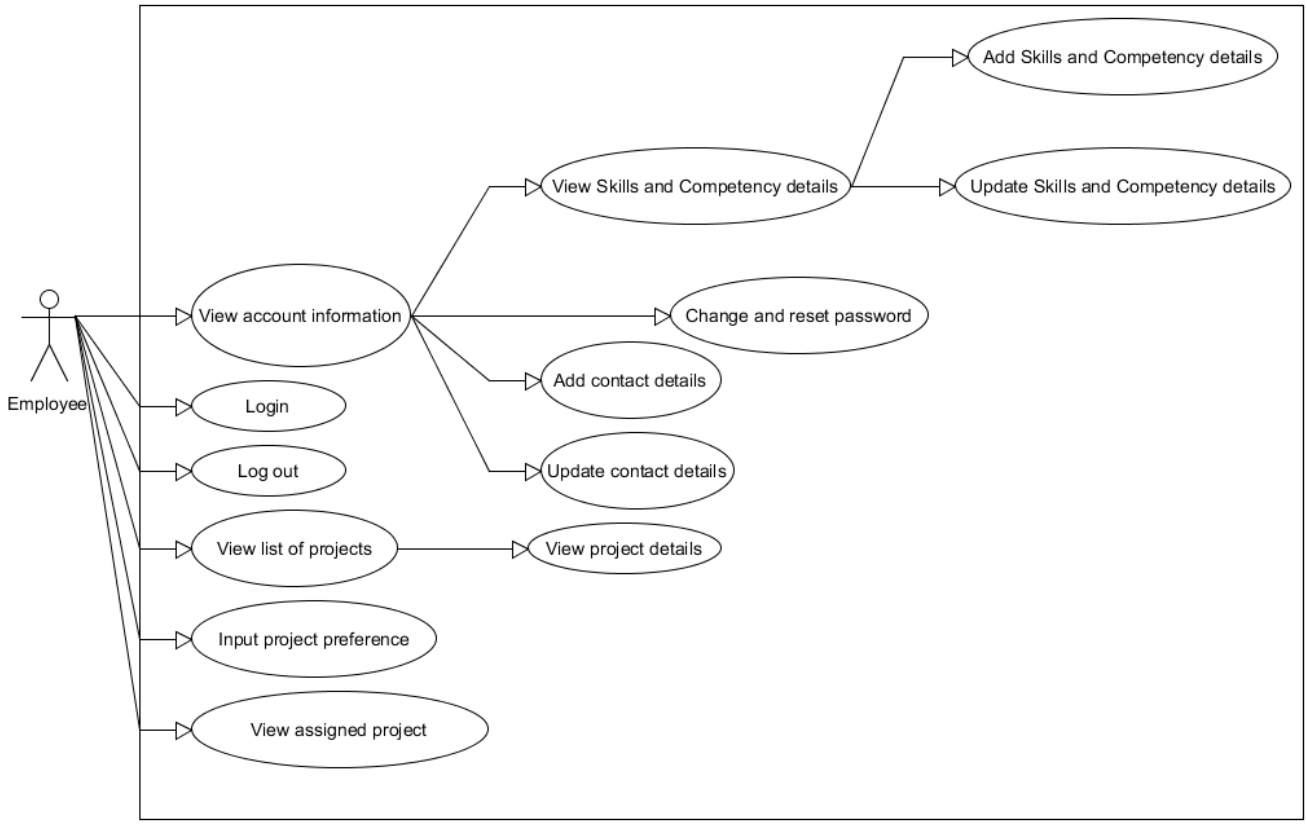
Use case: Adding project into the assignments as Project Admin	Taiga ID: #603
Stakeholders and goals: Project Admin - add the project into assignment	
Description: The Project Admin adds projects into the assignment	
Actors: Project Admin	
Pre-condition: User is logged in as Project Admin and assignment is created	
Post-condition: The project has been added	
Trigger: The Project Admin wants to add the project into the assignment	
Normal flow: 1. The user clicks on Assignment 2. The system brings the user to the Assignment page 3. The user clicks on the name of the desired assignment 4. The system brings the user to the corresponding assignment page 5. The user clicks on projects on the side panel 6. The user clicks on the Edit projects button 7. The user selects the desired projects and clicks on the submit button 8. The system verifies the request from the user 9. The system adds the project into the assignment	
Sub-flows: None	
Alternative/Exceptional flows: 8.a The system fails to verify the request from the user and prompts the user that an error has occurred	

Use case: Removing project into the assignments as Project Admin	Taiga ID: #604
Stakeholders and goals: Project Admin - remove the project from assignment	
Description: The Project Admin removes projects from the assignment	
Actors: Project Admin	
Pre-condition: User is logged in as Project Admin and projects are in the assignment	
Post-condition: The project has been removed	
Trigger: The Project Admin wants to remove a project from the assignment	
Normal flow: 1. The user clicks on Assignment 2. The system brings the user to the Assignment page 3. The user clicks on the name of the desired assignment 4. The system brings the user to the corresponding assignment page 5. The user clicks on projects on the side panel 6. The user clicks on the Edit projects button 7. The user clicks on the delete icon beside the desired project selection 8. The system verifies the request from the user 9. The system removes the project into the assignment	
Sub-flows: None	
Alternative/Exceptional flows: 8.a The system fails to verify the request from the user and prompts the user that an error has occurred	

Use case: Adding employees into the assignments as Project Admin	Taiga ID: #605
Stakeholders and goals: Project Admin - add employees into assignment	
Description: The Project Admin adds employees into the assignment	
Actors: Project Admin	
Pre-condition: User is logged in as Project Admin and assignment is created	
Post-condition: The employee has been added	
Trigger: The Project Admin wants to add employees into the assignment	
Normal flow: 1. The user clicks on Assignment 2. The system brings the user to the Assignment page 3. The user clicks on the name of the desired assignment 4. The system brings the user to the corresponding assignment page 5. The user clicks on Employees on the side panel 6. The user clicks on the Edit employees button 7. The user selects the desired employee and clicks on the submit button 8. The system verifies the request from the user 9. The system adds the employees into the assignment	
Sub-flows: None	
Alternative/Exceptional flows: 8.a The system fails to verify the request from the user and prompts the user that an error has occurred	

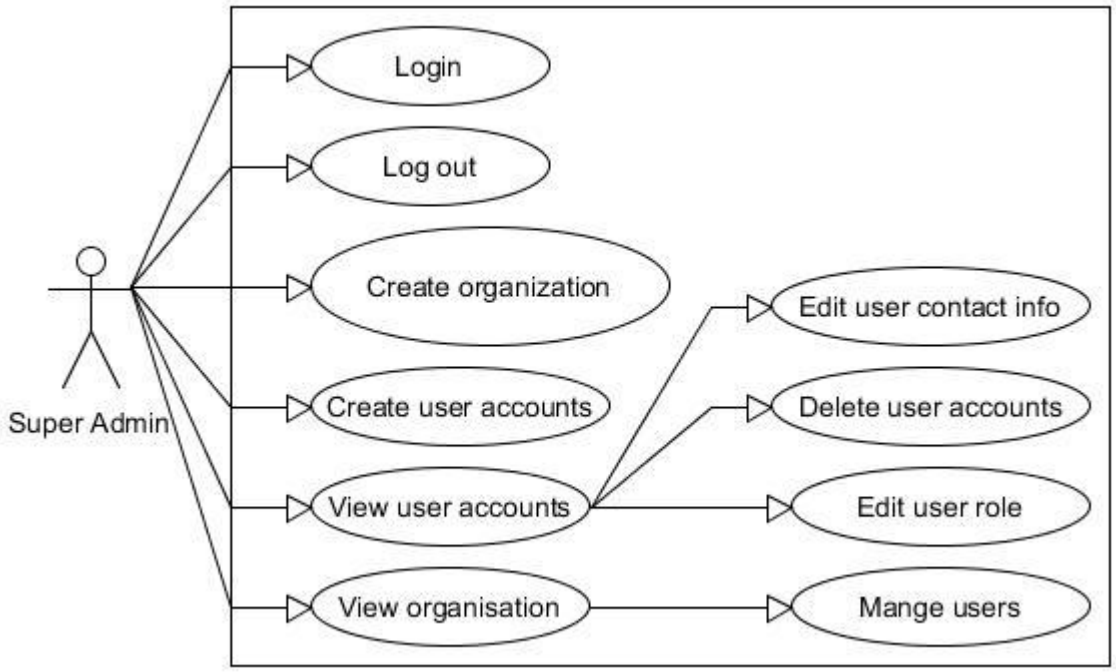
5.3 Use Case Diagram

Employee



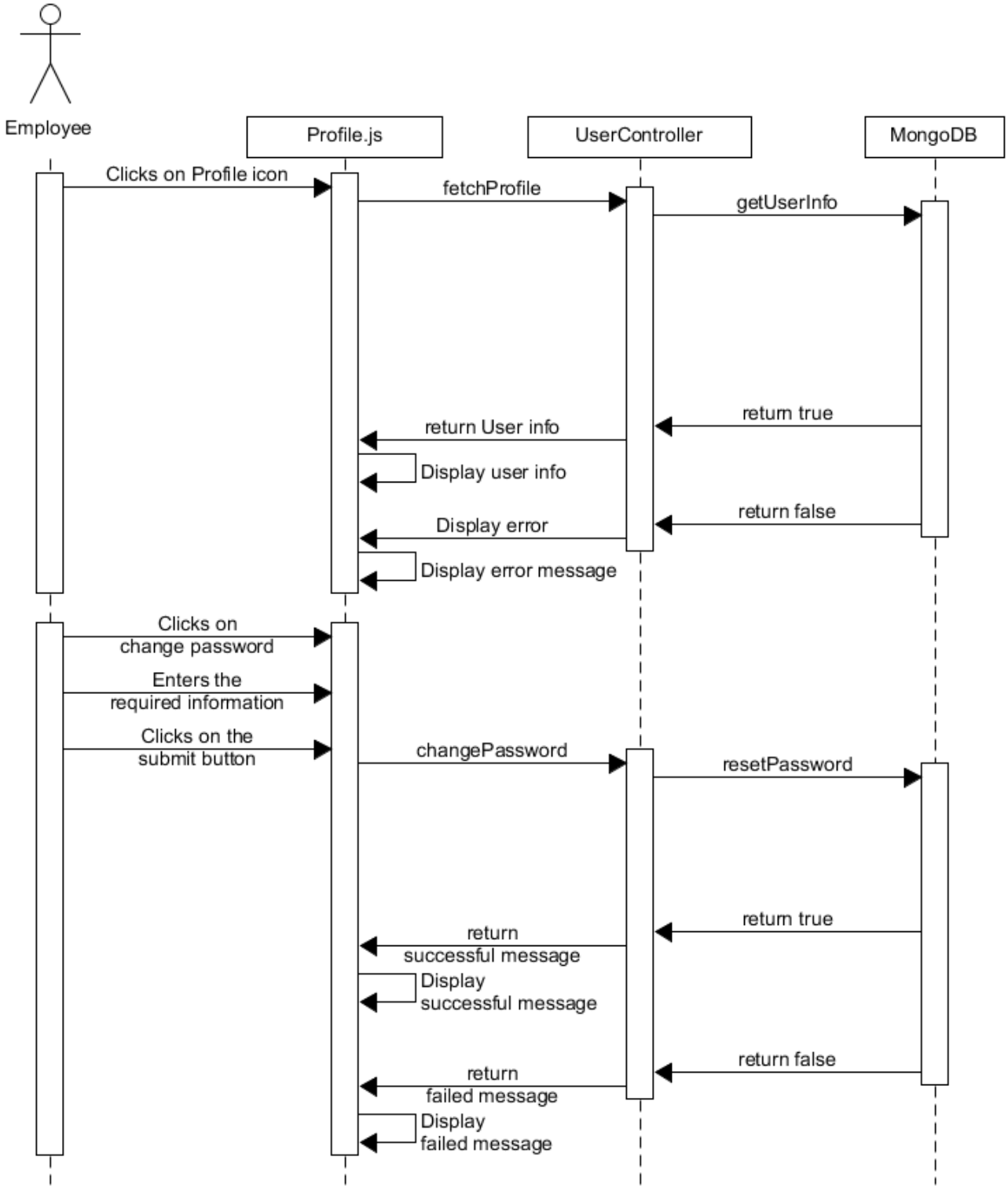
Project Admin

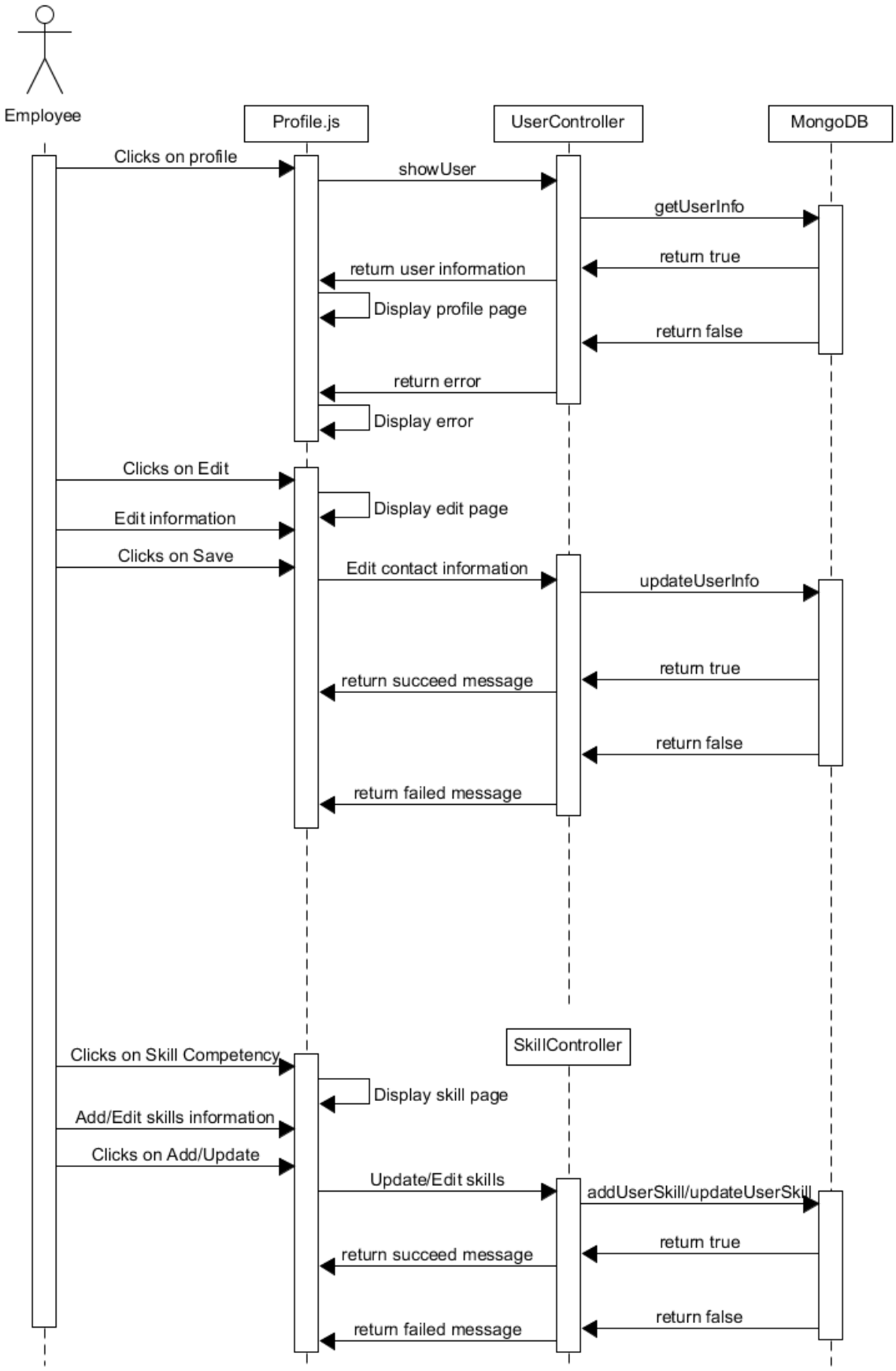
Super Admin

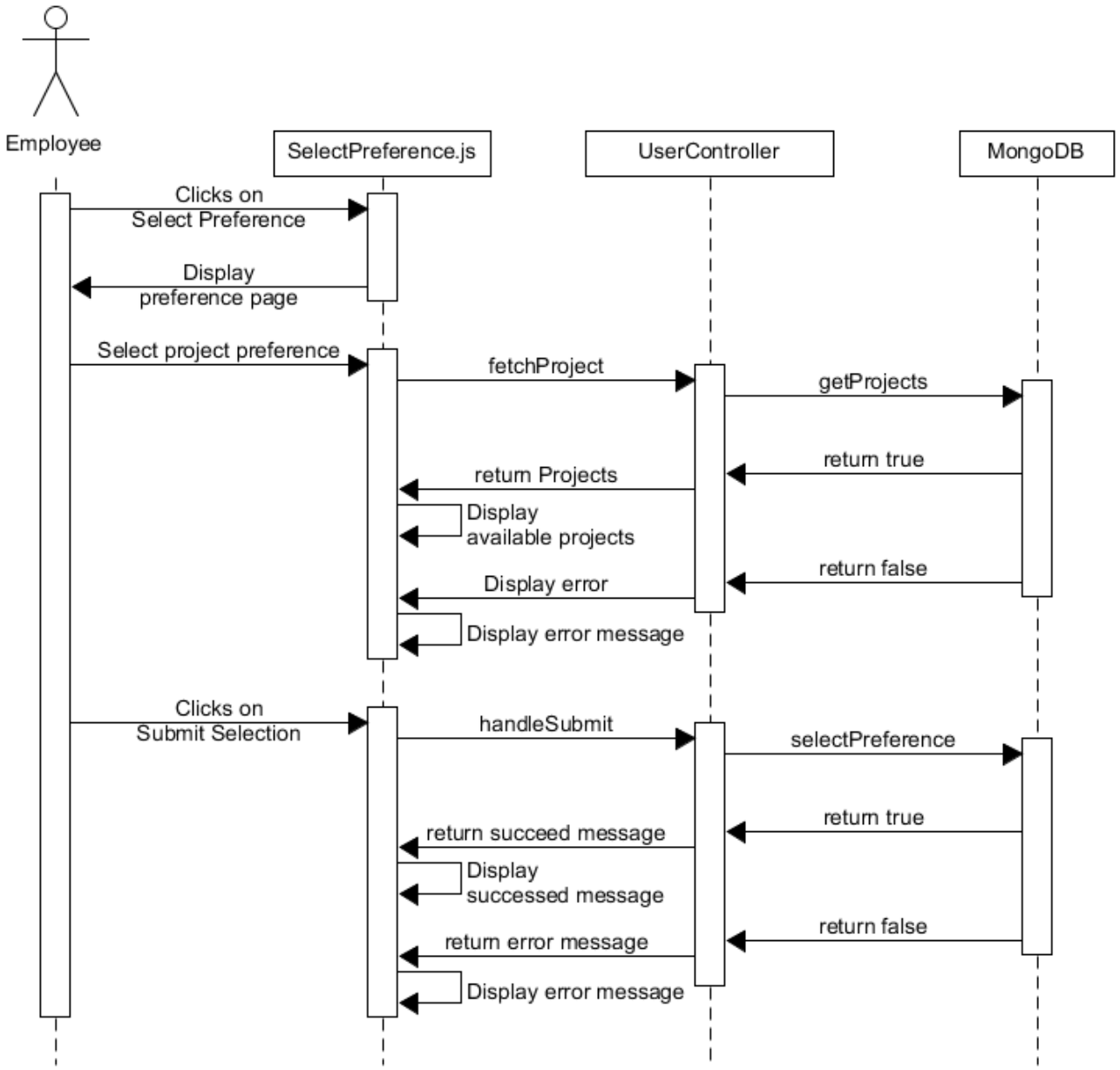

Login
Log out
Create organization
Create user accounts
View user accounts
View organisation
Edit user contact info
Delete user accounts
Edit user role
Mange users
Super Admin

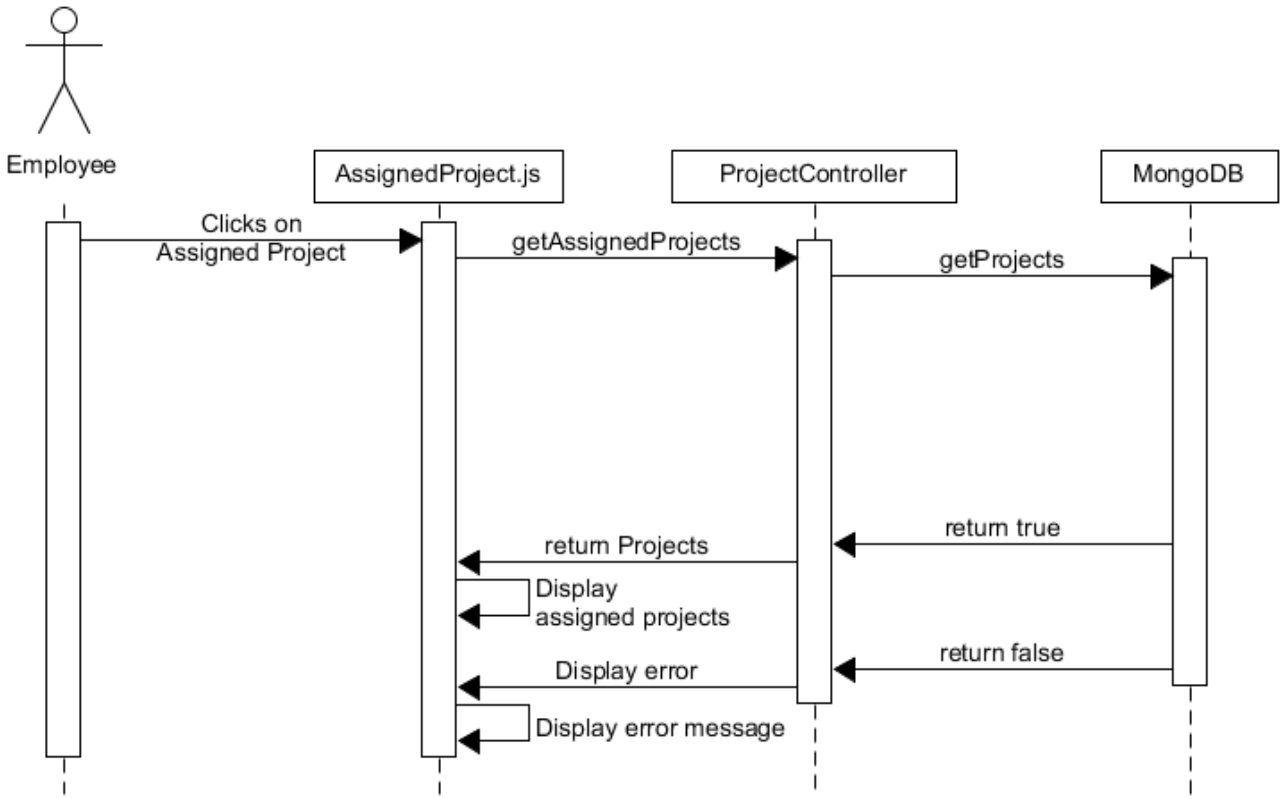
5.4 Sequence Diagram

Employee

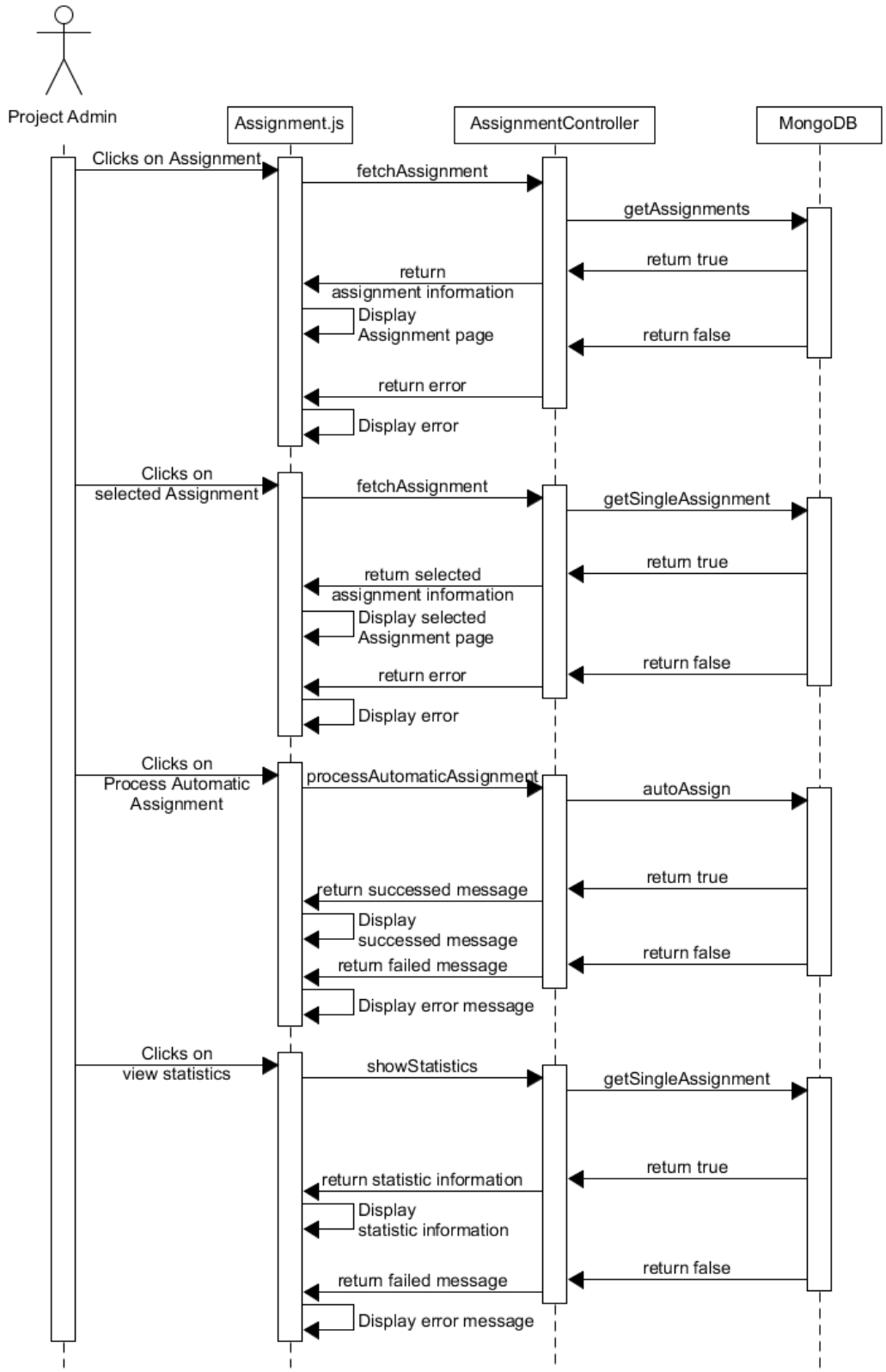


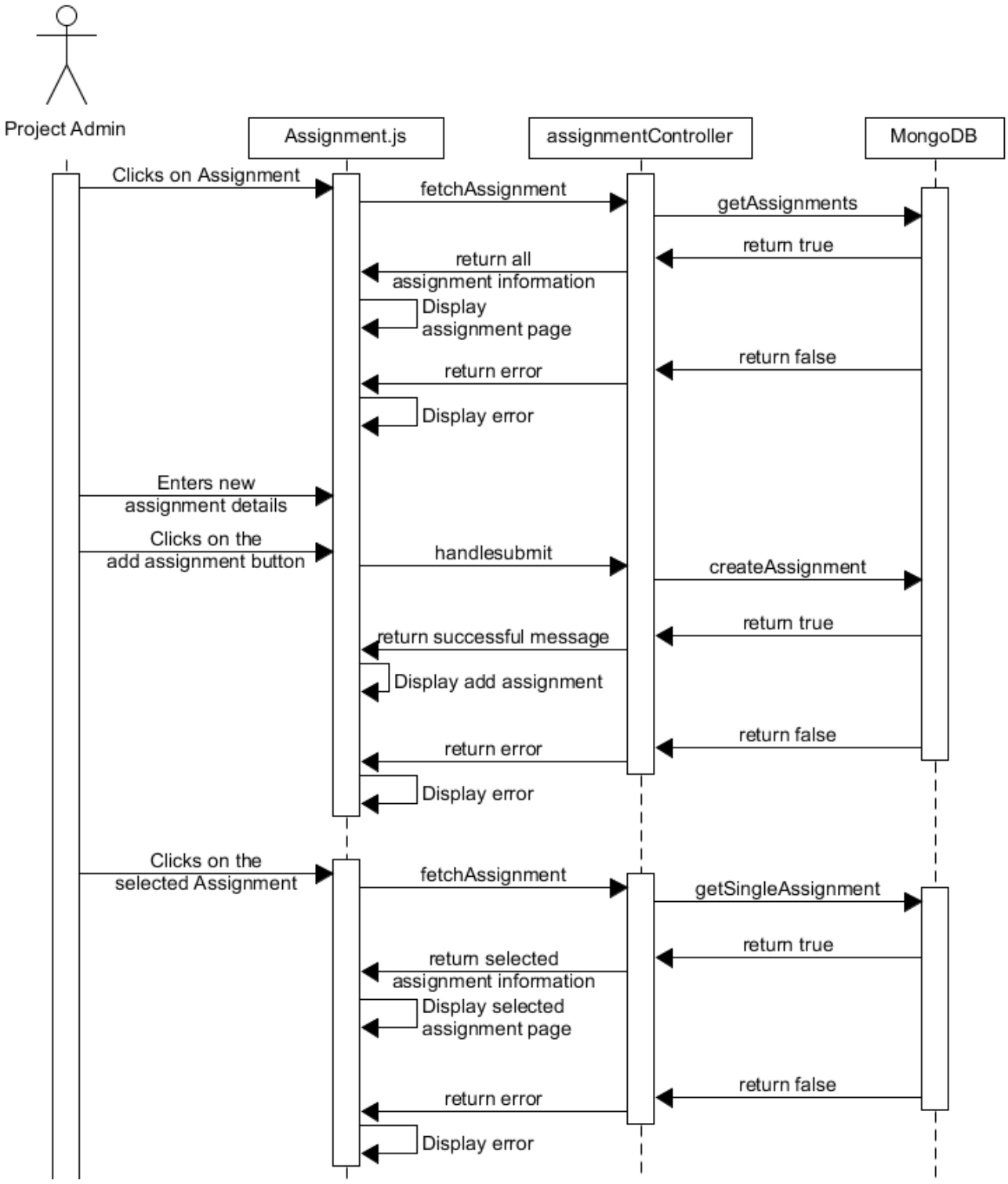

Employee
Profile.js
UserController
MongoDB
Clicks on profile
showUser
getUserInfo
return true
return user information
Display profile page
return false
return error
Display error
Clicks on Edit
Display edit page
Edit information
Clicks on Save
Edit contact information
updateUserInfo
return true
return succeed message
return false
return failed message
SkillController
Clicks on Skill Competency
Display skill page
Add/Edit skills information
Clicks on Add/Update
Update/Edit skills
addUserSkill/updateUserSkill
return true
return succeed message
return false
return failed message

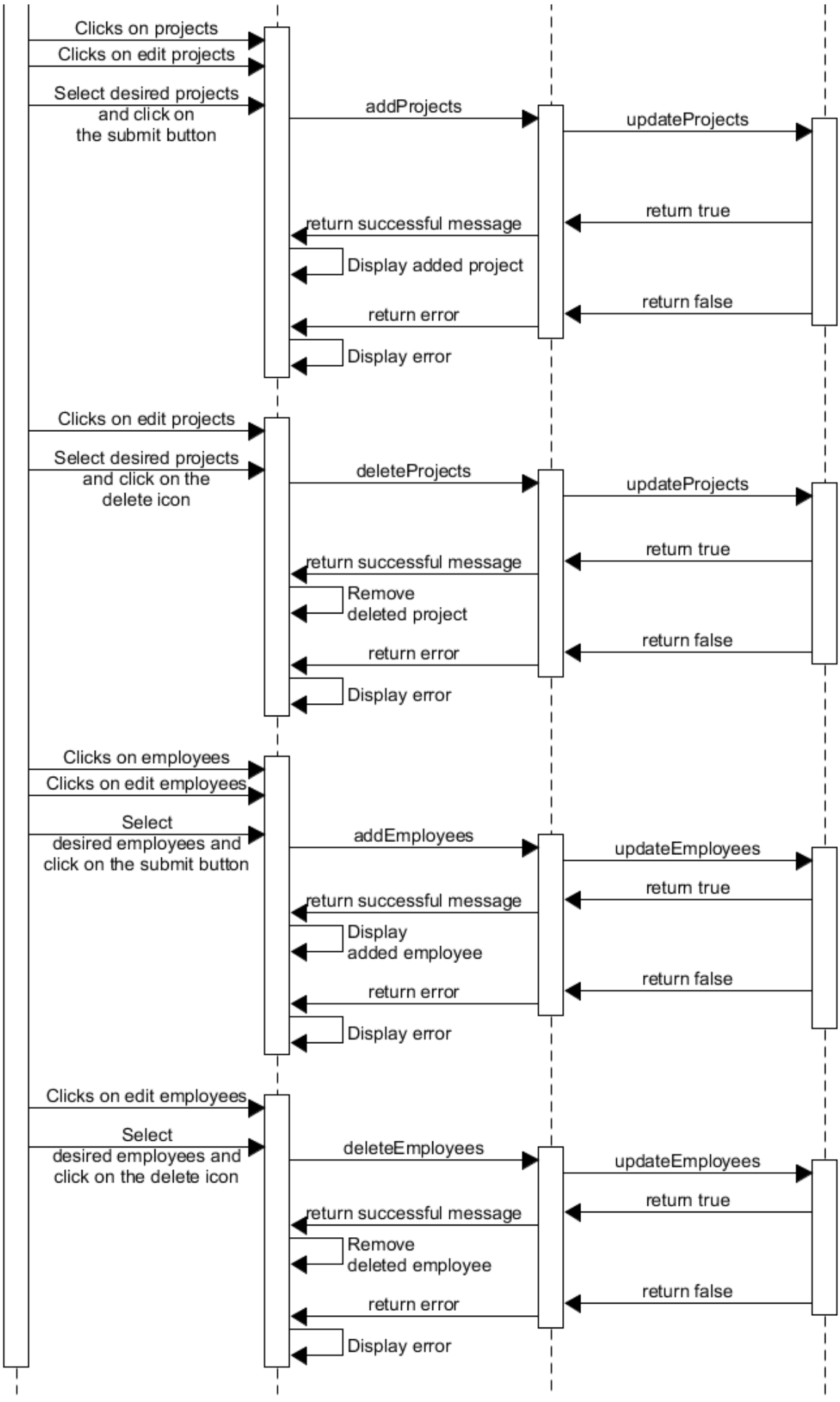

Employee
SelectPreference.js
UserController
MongoDB
Clicks on
Select Preference
Display
preference page
Select project preference
fetchProject
getProjects
return true
return Projects
Display
available projects
return false
Display error
Display error message
Clicks on
Submit Selection
handleSubmit
selectPreference
return true
return succeed message
Display
successed message
return false
return error message
Display error message

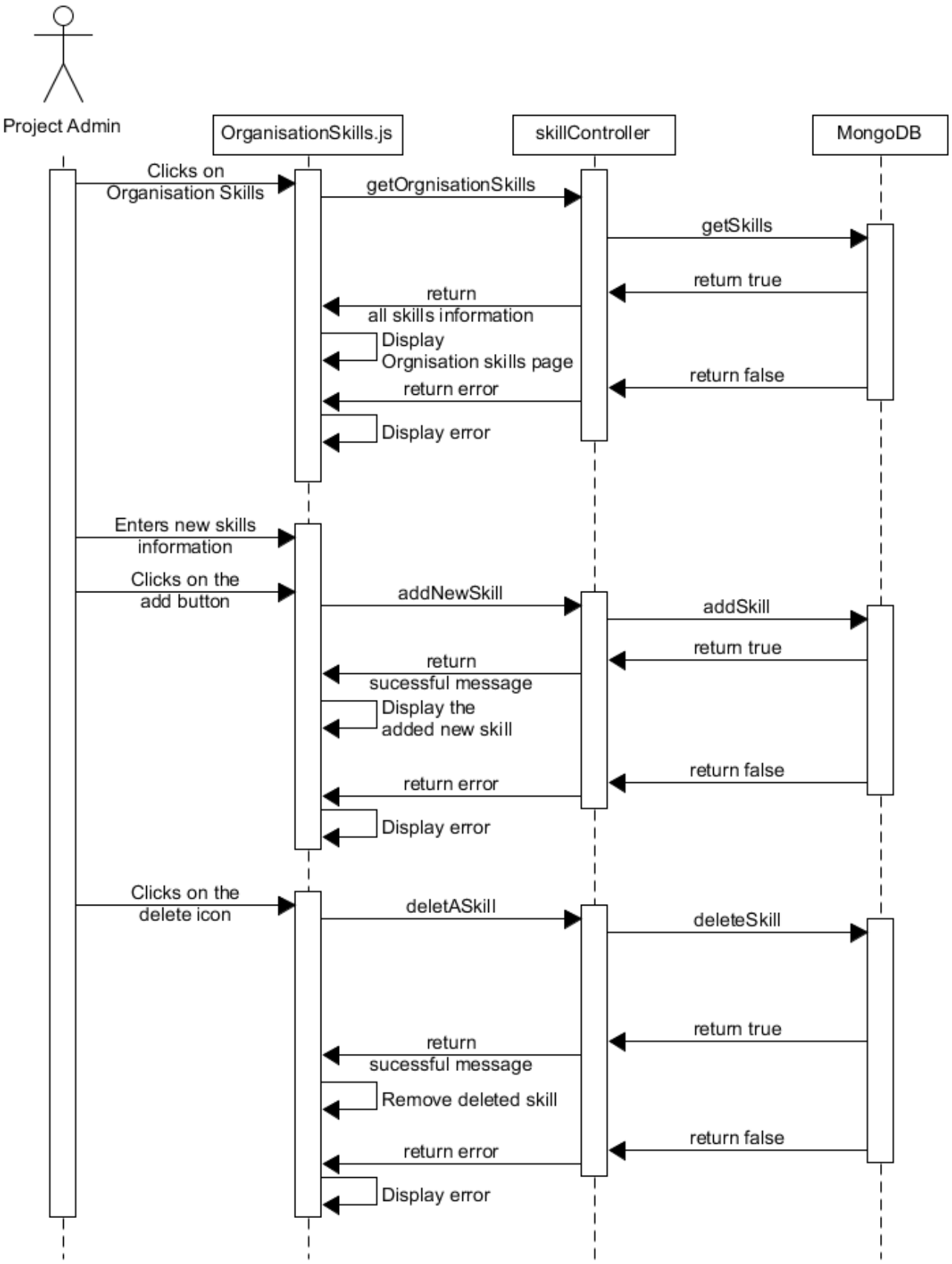

Employee
AssignedProject.js
ProjectController
MongoDB
Clicks on
Assigned Project
getAssignedProjects
getProjects
return true
return Projects
Display
assigned projects
return false
Display error
Display error message

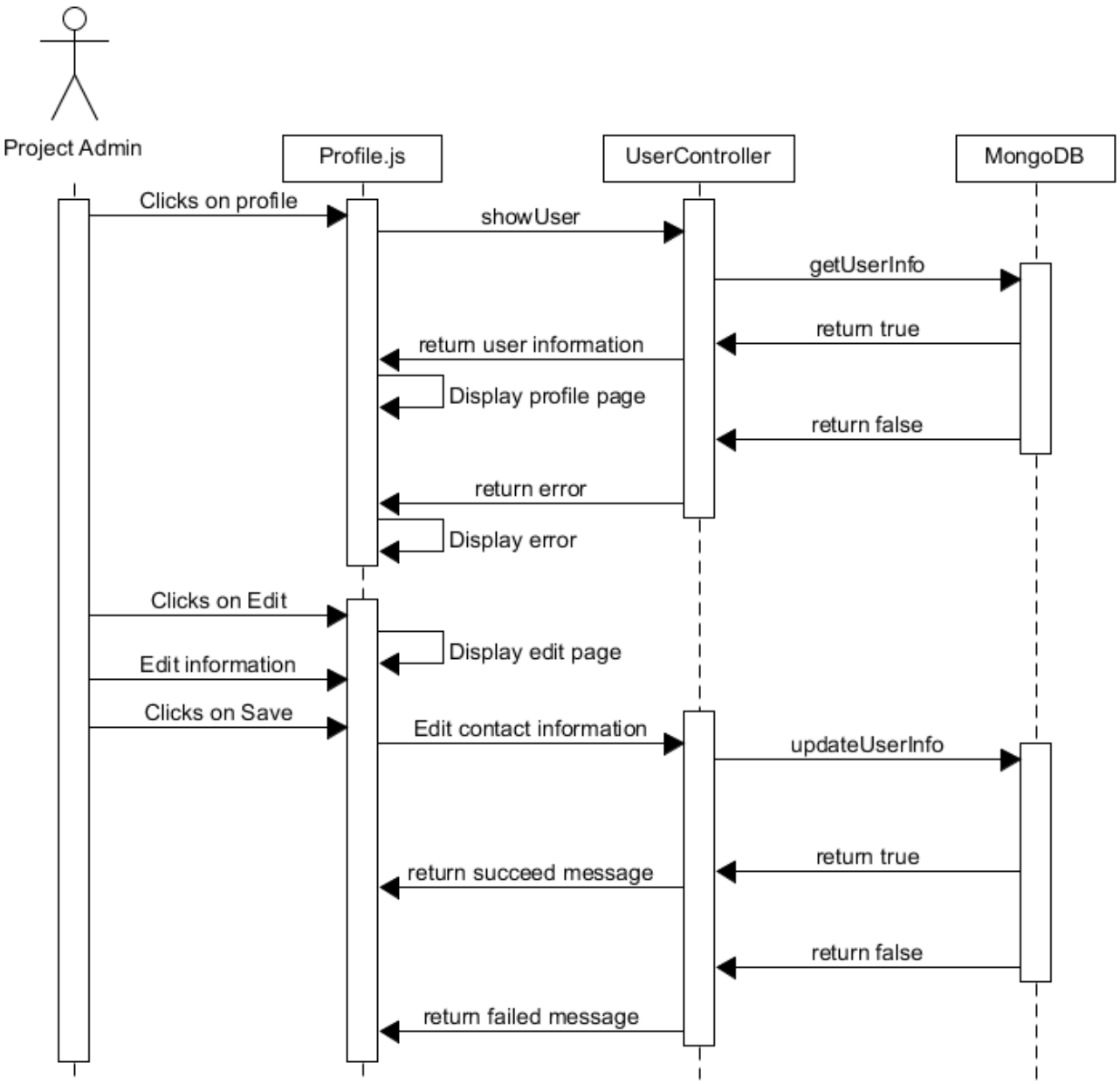
Project Admin

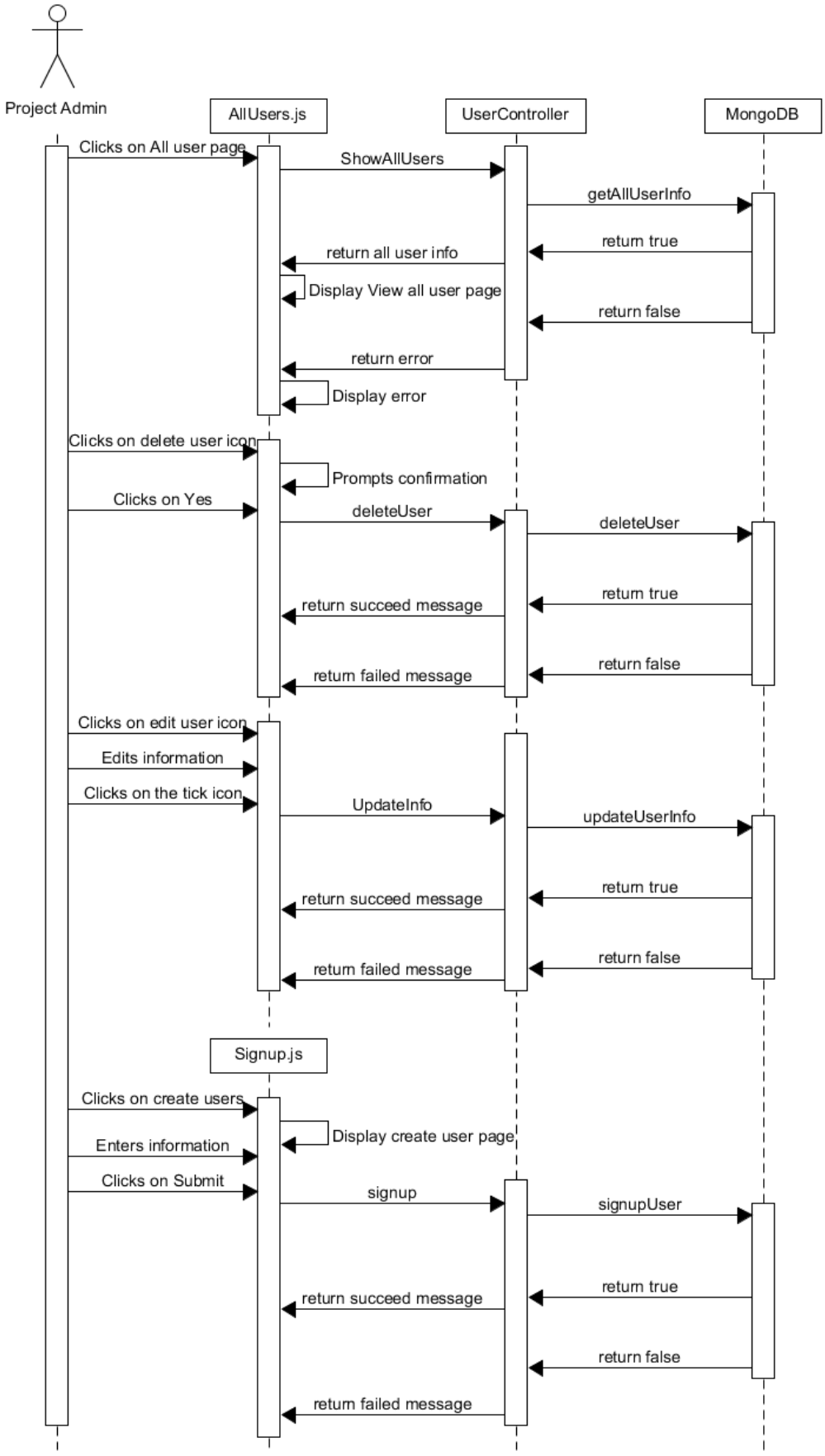


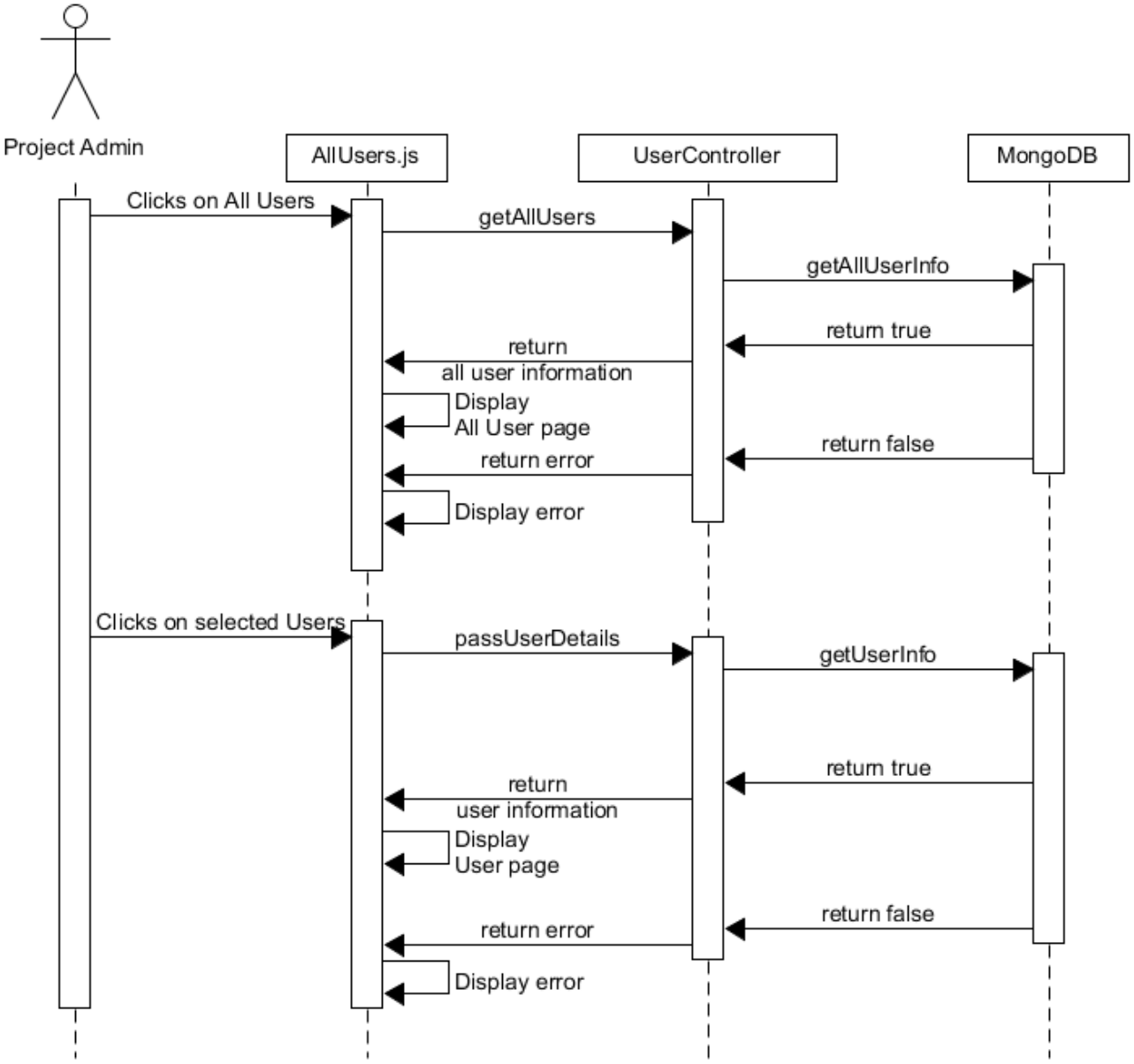

Project Admin
Assignment.js
assignmentController
MongoDB
Clicks on Assignment
fetchAssignment
getAssignments
return true
return all
assignment information
Display
assignment page
return false
return error
Display error
Enters new
assignment details
Clicks on the
add assignment button
handlesubmit
createAssignment
return true
return successful message
Display add assignment
return false
return error
Display error
Clicks on the
selected Assignment
fetchAssignment
getSingleAssignment
return true
return selected
assignment information
Display selected
assignment page
return false
return error
Display error


Clicks on projects
Clicks on edit projects
Select desired projects and click on the submit button
addProjects
updateProjects
return true
return successful message
Display added project
return false
return error
Display error
Clicks on edit projects
Select desired projects and click on the delete icon
deleteProjects
updateProjects
return true
return successful message
Remove deleted project
return false
return error
Display error
Clicks on employees
Clicks on edit employees
Select desired employees and click on the submit button
addEmployees
updateEmployees
return true
return successful message
Display added employee
return false
return error
Display error
Clicks on edit employees
Select desired employees and click on the delete icon
deleteEmployees
updateEmployees
return true
return successful message
Remove deleted employee
return false
return error
Display error

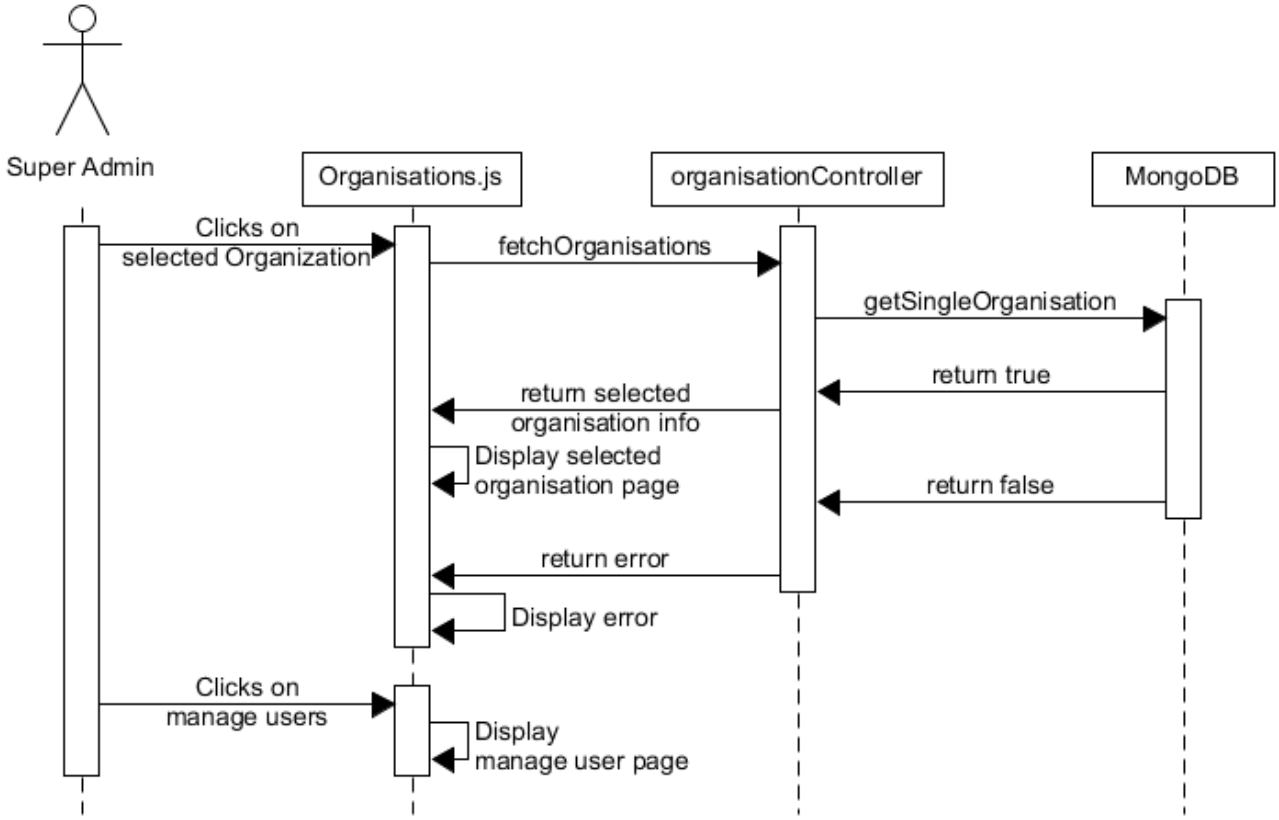

Project Admin
OrganisationSkills.js
skillController
MongoDB
Clicks on
Organisation Skills
getOrgnisationSkills
getSkills
return true
return
all skills information
Display
Orgnisation skills page
return false
return error
Display error
Enters new skills
information
Clicks on the
add button
addNewSkill
addSkill
return true
return
sucessful message
Display the
added new skill
return false
return error
Display error
Clicks on the
delete icon
deletASkill
deleteSkill
return true
return
sucessful message
Remove deleted skill
return false
return error
Display error

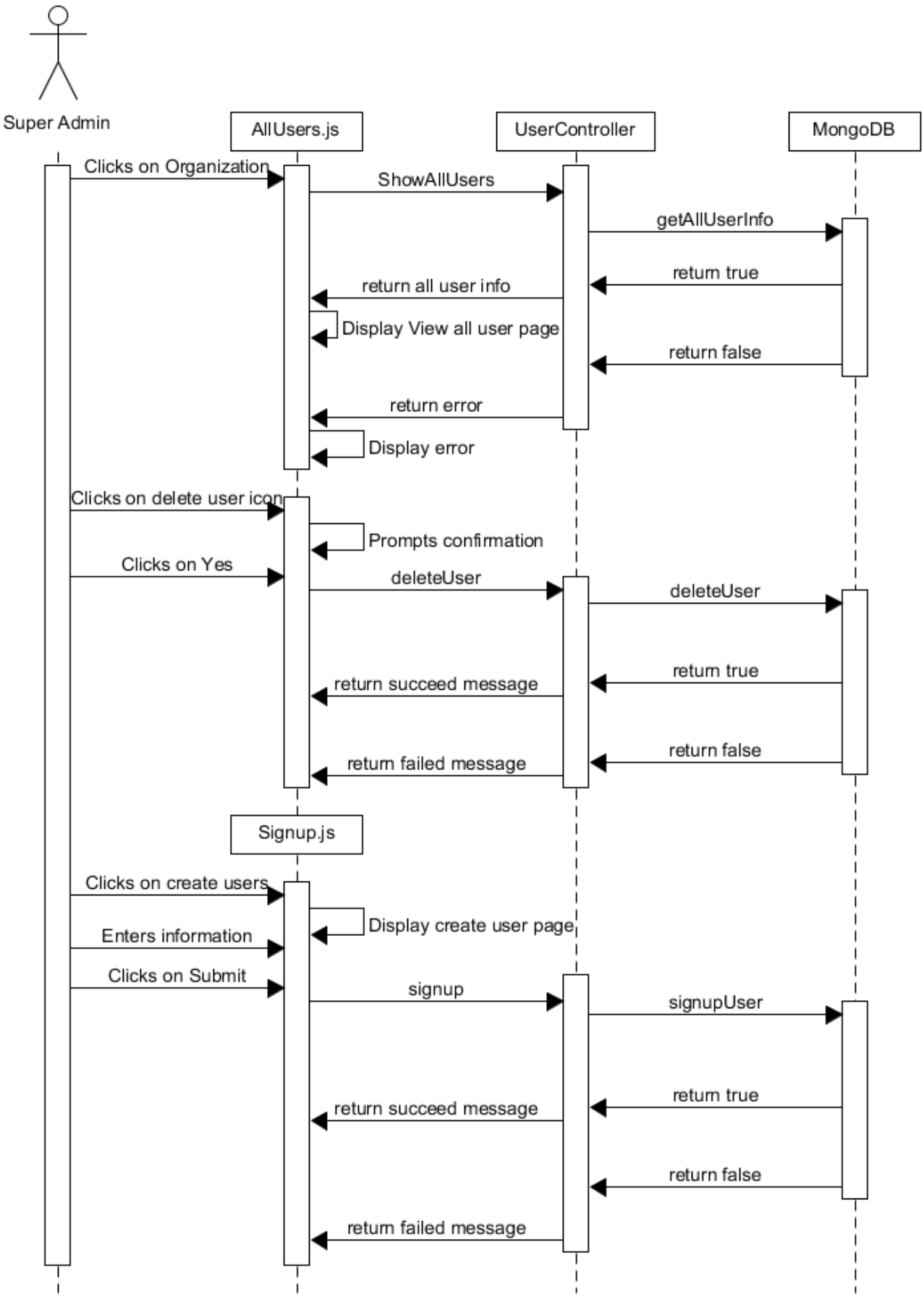

Project Admin
Profile.js
UserController
MongoDB
Clicks on profile
showUser
getUserInfo
return true
return user information
Display profile page
return false
return error
Display error
Clicks on Edit
Display edit page
Edit information
Clicks on Save
Edit contact information
updateUserInfo
return true
return succeed message
return false
return failed message

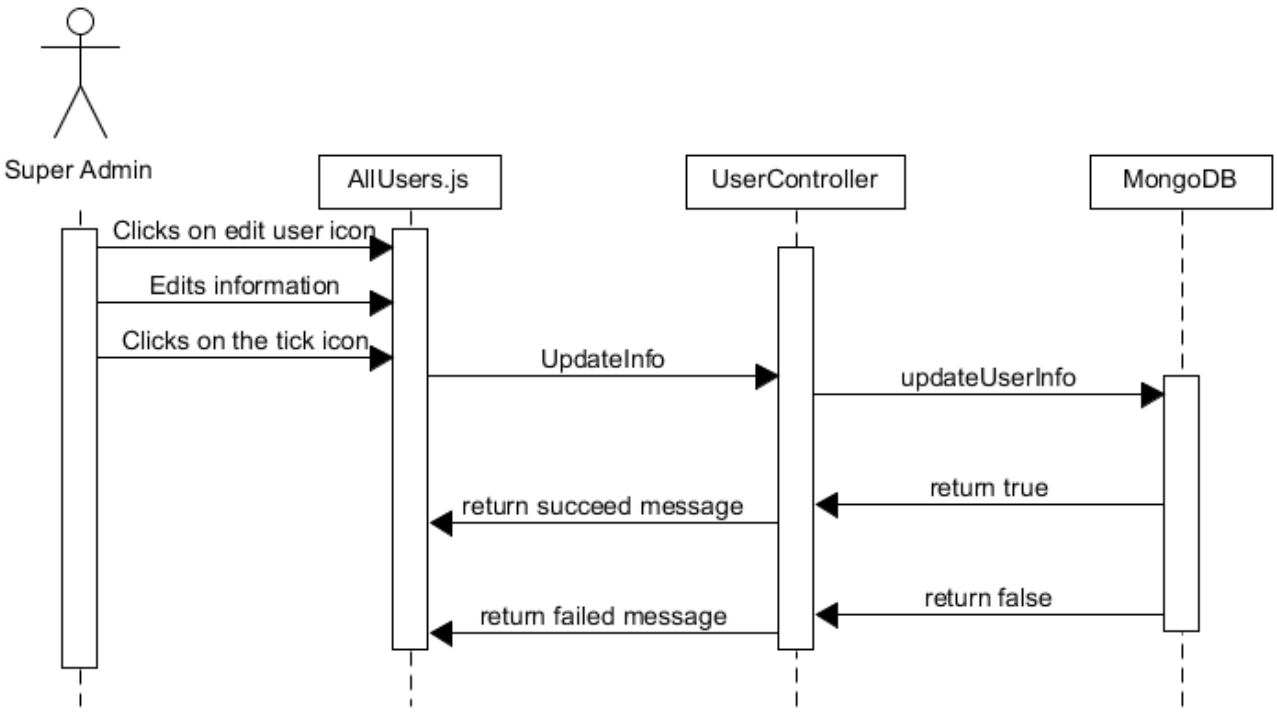

Project Admin
AllUsers.js
UserController
MongoDB
Clicks on All user page
ShowAllUsers
getAllUserInfo
return true
return all user info
Display View all user page
return false
return error
Display error
Clicks on delete user icon
Prompts confirmation
Clicks on Yes
deleteUser
deleteUser
return true
return succeed message
return false
return failed message
Clicks on edit user icon
Edits information
Clicks on the tick icon
UpdateInfo
updateUserInfo
return true
return succeed message
return false
return failed message
Signup.js
Clicks on create users
Display create user page
Enters information
Clicks on Submit
signup
signupUser
return true
return succeed message
return false
return failed message

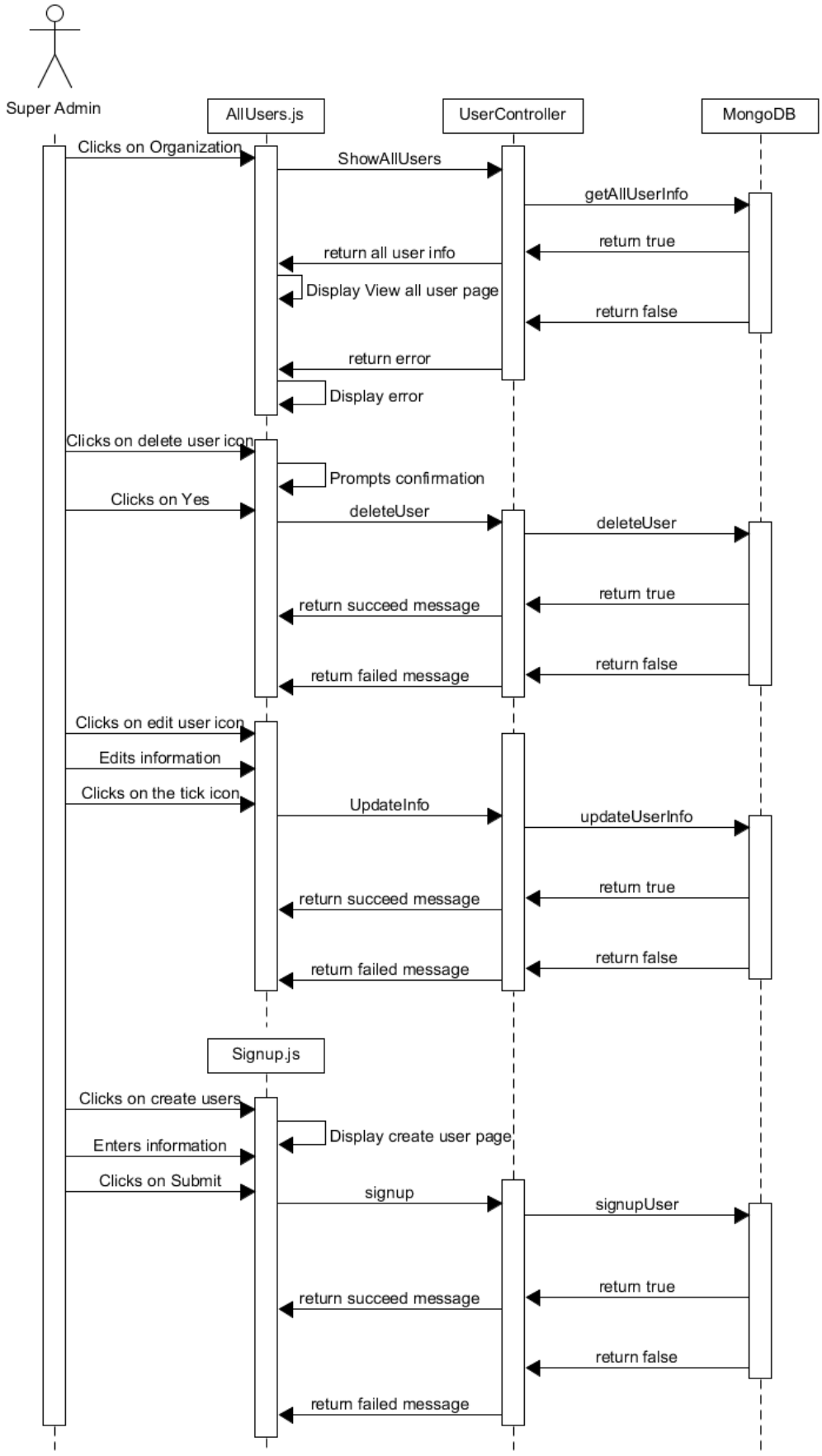

Project Admin
AllUsers.js
UserController
MongoDB
Clicks on All Users
getAllUsers
getAllUserInfo
return true
return
all user information
Display
All User page
return false
return error
Display error
Clicks on selected Users
passUserDetails
getUserInfo
return true
return
user information
Display
User page
return false
return error
Display error

Super Admin


Super Admin
Organisations.js
organisationController
MongoDB
Clicks on selected Organization
fetchOrganisations
getSingleOrganisation
return true
return selected organisation info
Display selected organisation page
return false
return error
Display error
Clicks on manage users
Display manage user page

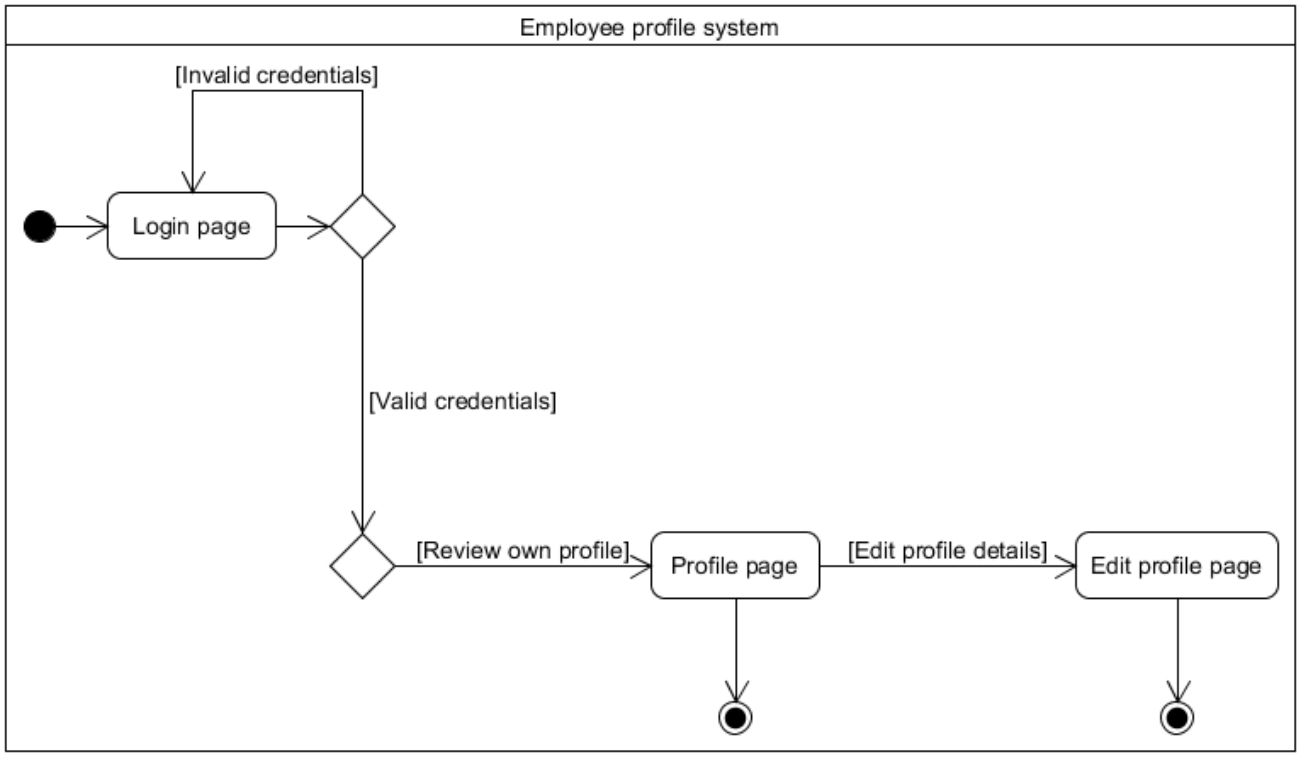

Super Admin
AllUsers.js
UserController
MongoDB
Clicks on Organization
ShowAllUsers
getAllUserInfo
return true
return all user info
Display View all user page
return false
return error
Display error
Clicks on delete user icon
Prompts confirmation
Clicks on Yes
deleteUser
deleteUser
return true
return succeed message
return false
return failed message
Signup.js
Clicks on create users
Display create user page
Enters information
Clicks on Submit
signup
signupUser
return true
return succeed message
return false
return failed message

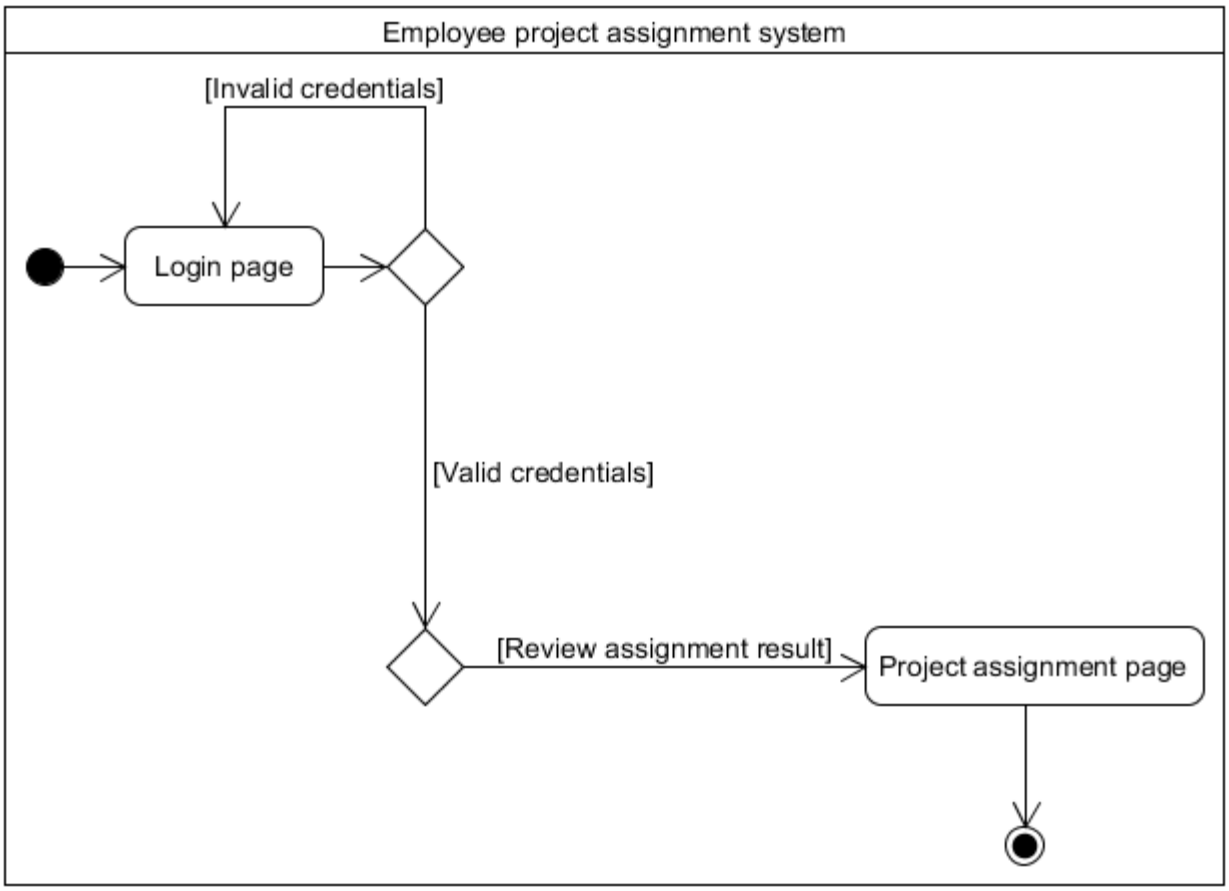

Super Admin
AllUsers.js
UserController
MongoDB
Clicks on edit user icon
Edits information
Clicks on the tick icon
UpdateInfo
updateUserInfo
return true
return succeed message
return false
return failed message

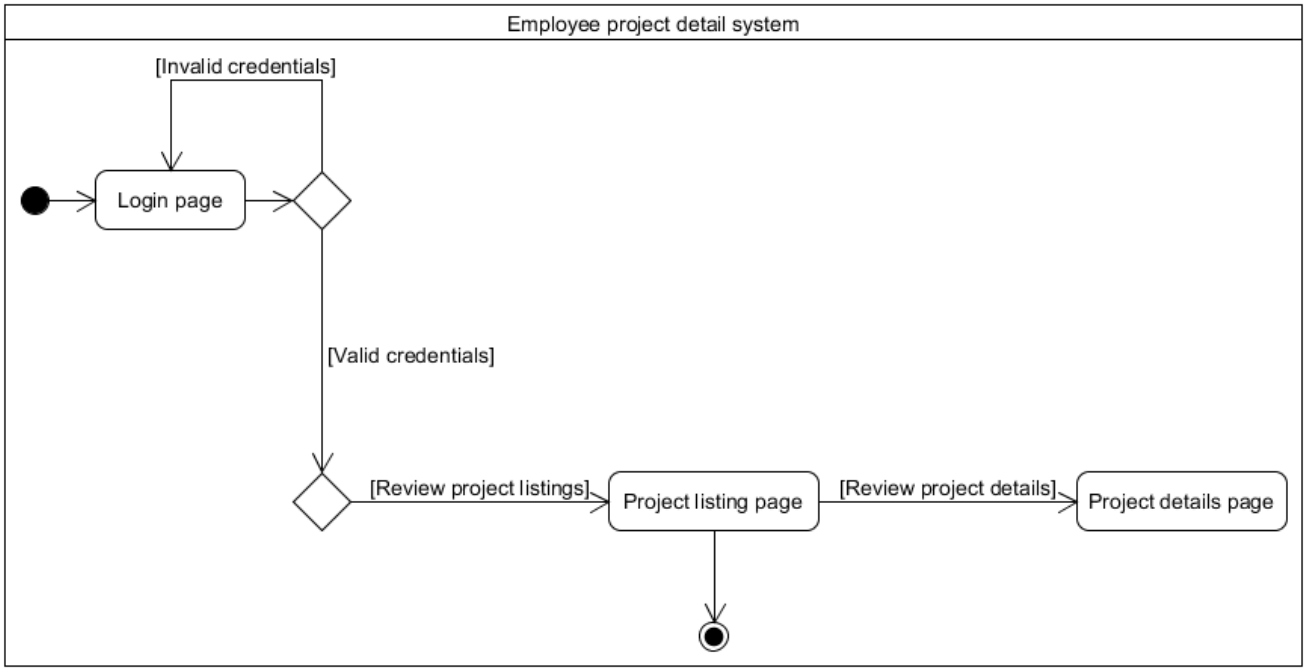

Super Admin
AllUsers.js
UserController
MongoDB
Clicks on Organization
ShowAllUsers
getAllUserInfo
return true
return all user info
Display View all user page
return false
return error
Display error
Clicks on delete user icon
Prompts confirmation
Clicks on Yes
deleteUser
deleteUser
return true
return succeed message
return false
return failed message
Clicks on edit user icon
Edits information
Clicks on the tick icon
UpdateInfo
updateUserInfo
return true
return succeed message
return false
return failed message
Signup.js
Clicks on create users
Display create user page
Enters information
Clicks on Submit
signup
signupUser
return true
return succeed message
return false
return failed message

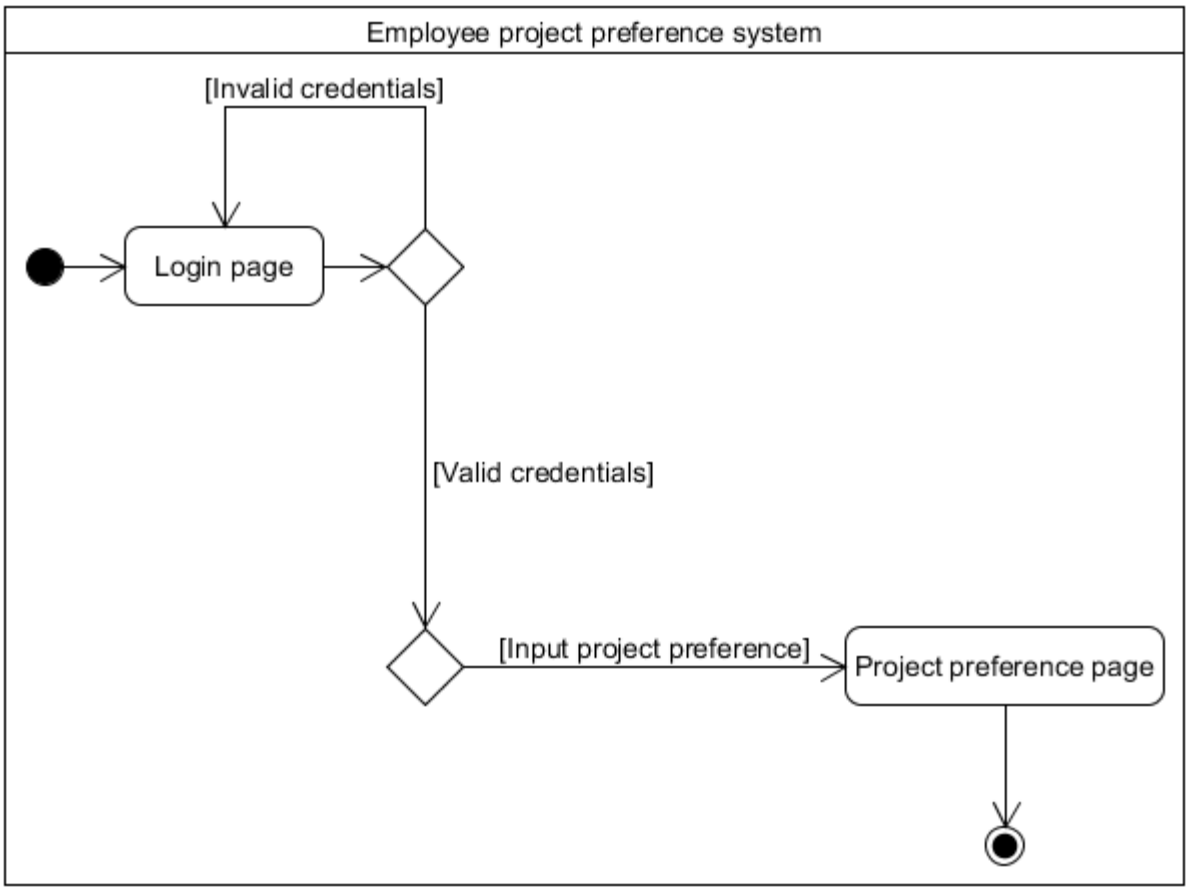
5.5 State Diagram

Employee

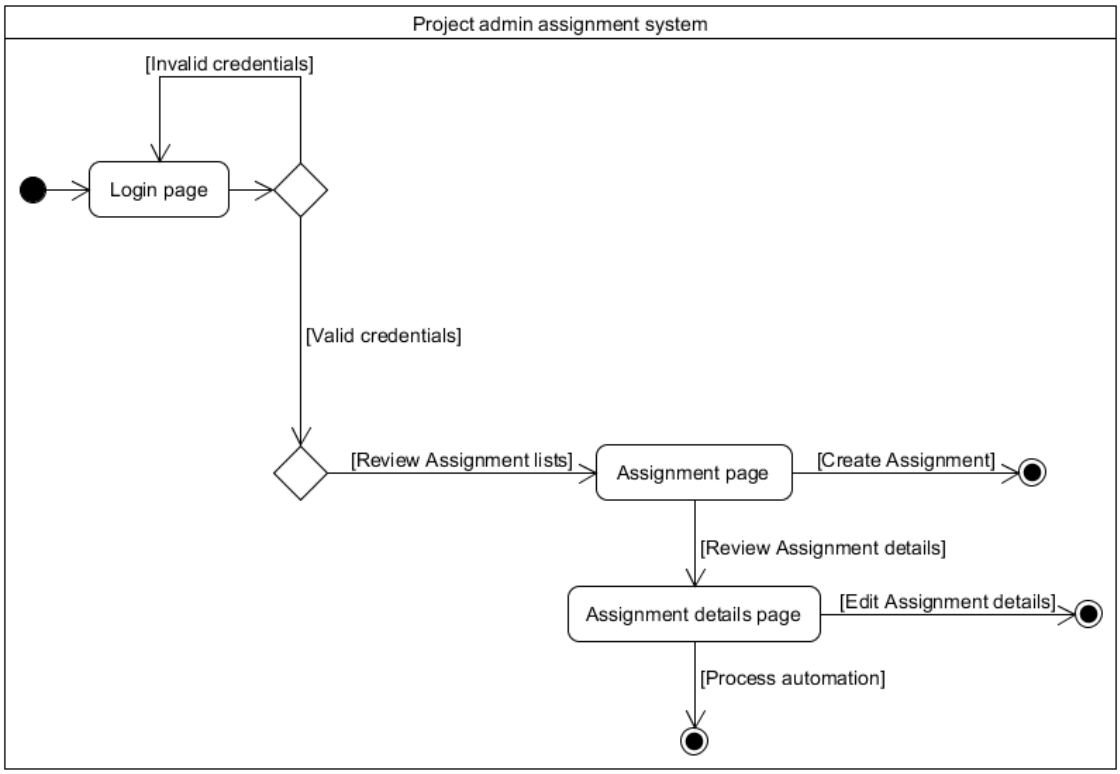


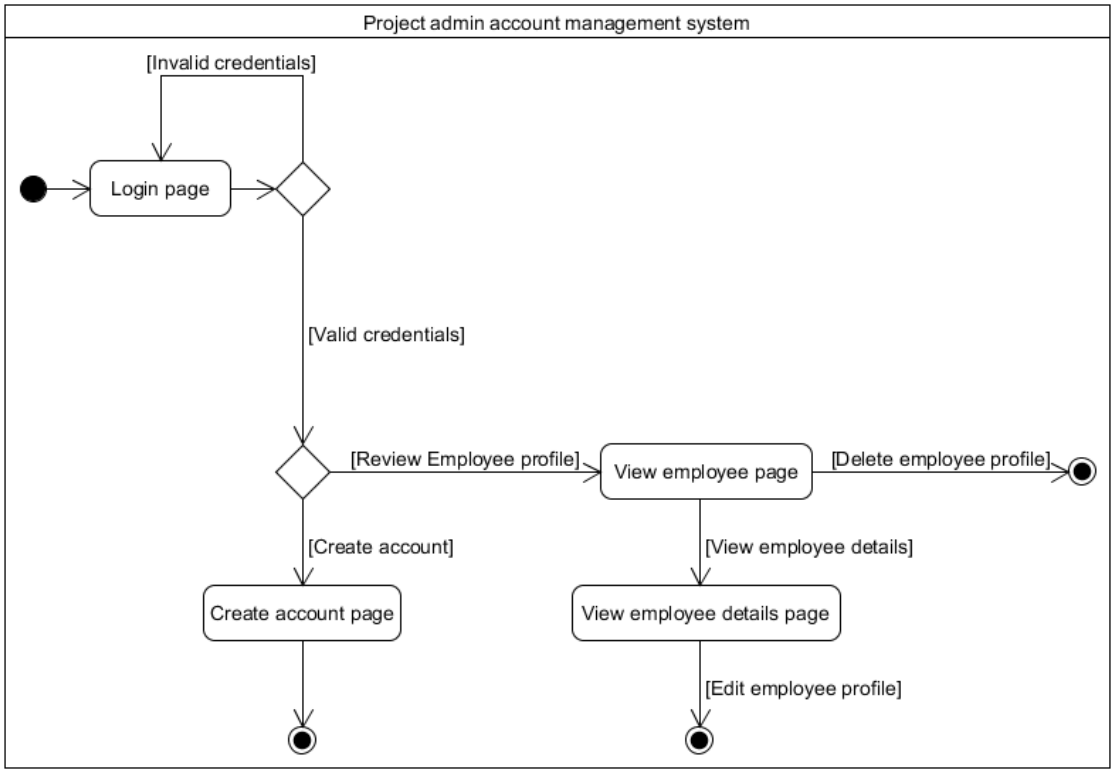

Employee project assignment system
[Invalid credentials]
Login page
[Valid credentials]
[Review assignment result]
Project assignment page

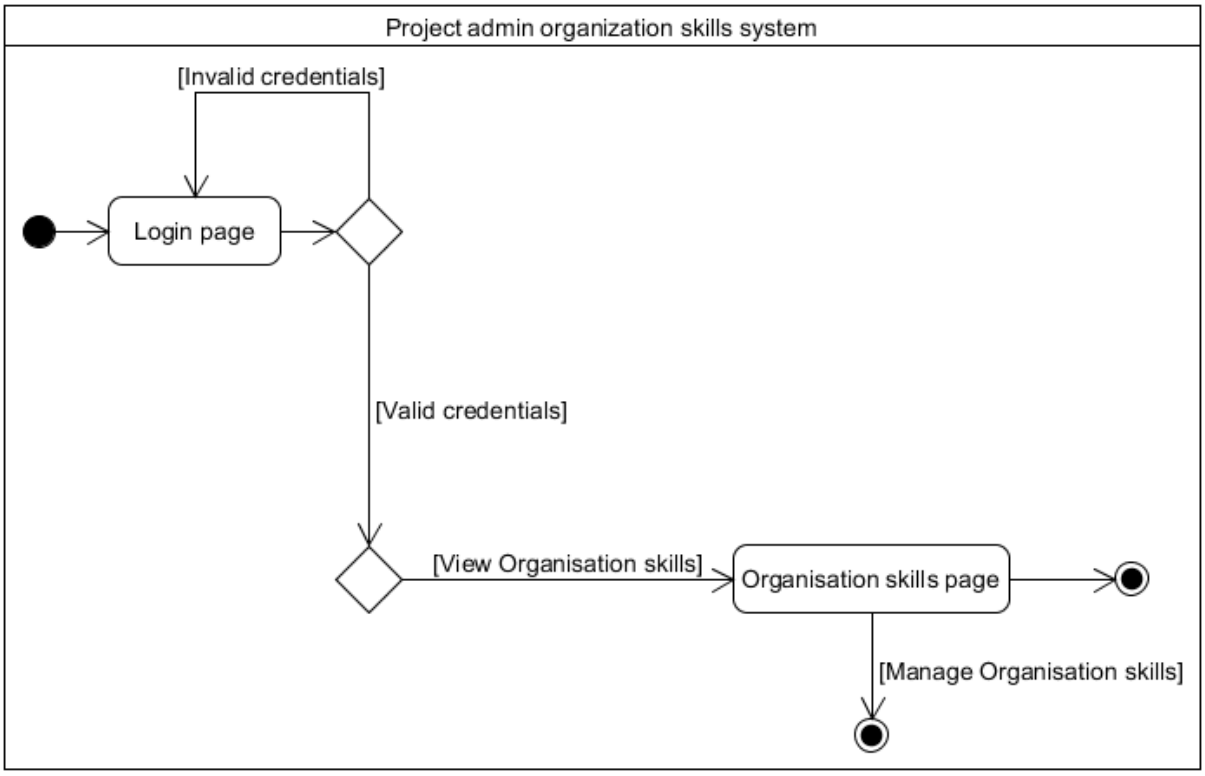

Employee project detail system
[Invalid credentials]
Login page
[Valid credentials]
[Review project listings]
Project listing page
[Review project details]
Project details page

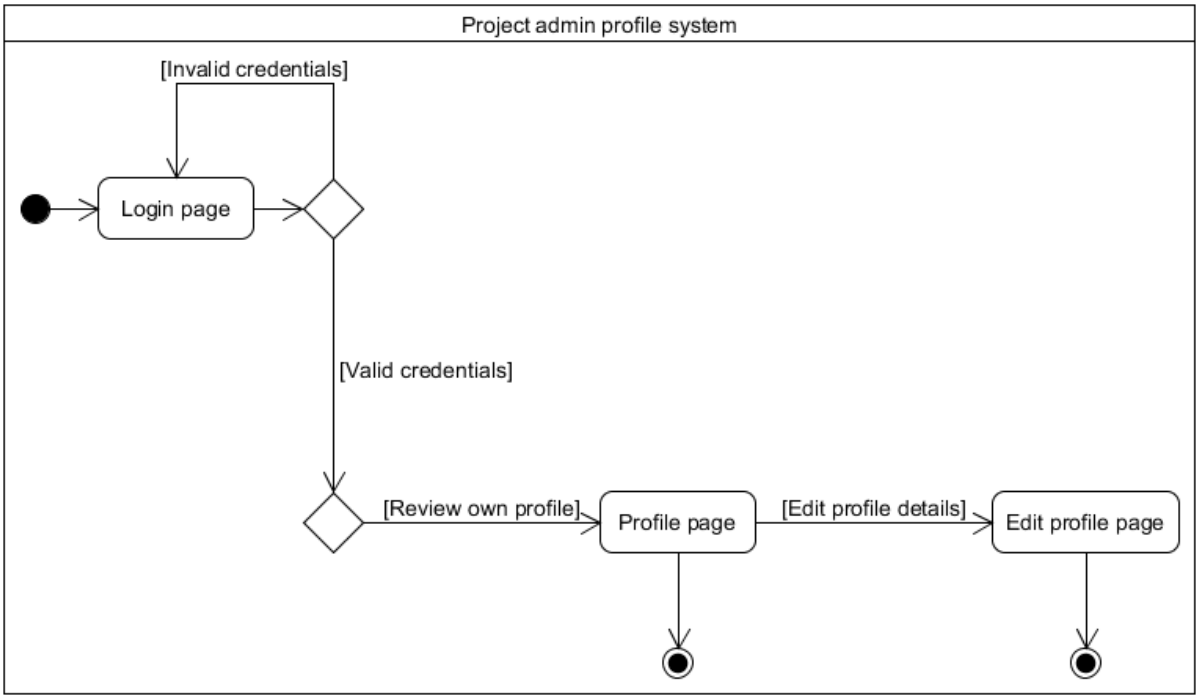

Employee project preference system
[Invalid credentials]
Login page
[Valid credentials]
[Input project preference]
Project preference page

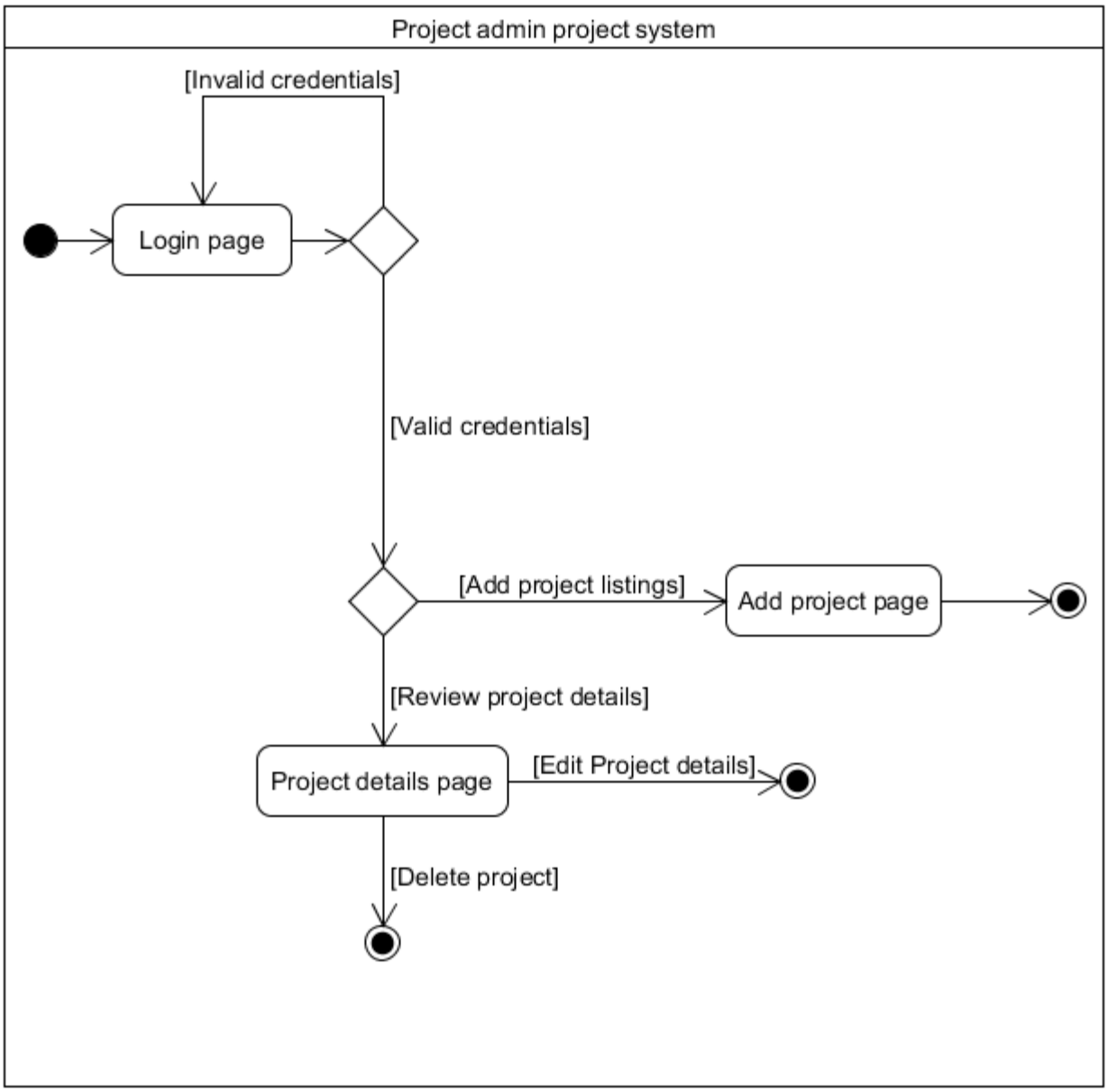
Project Admin



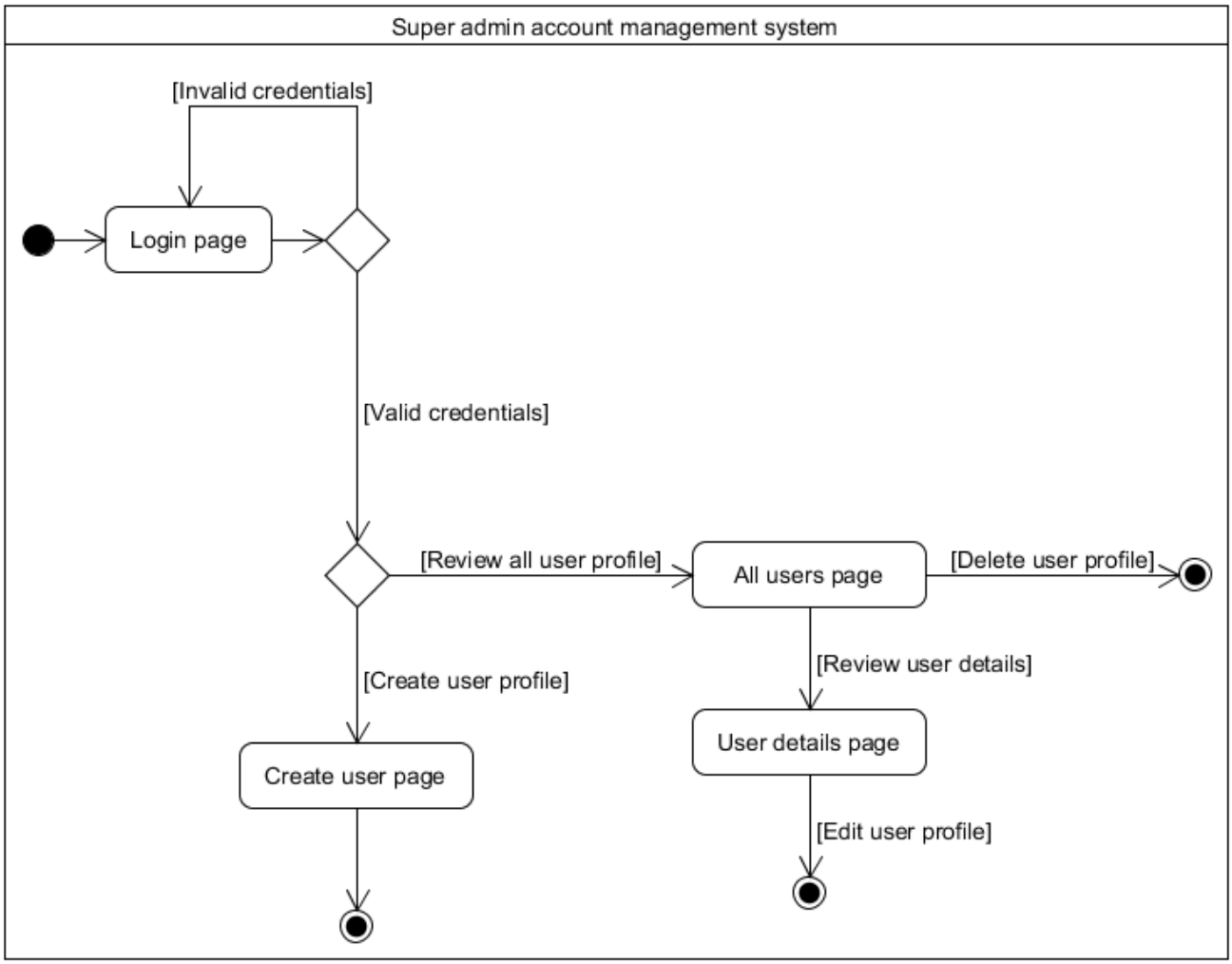


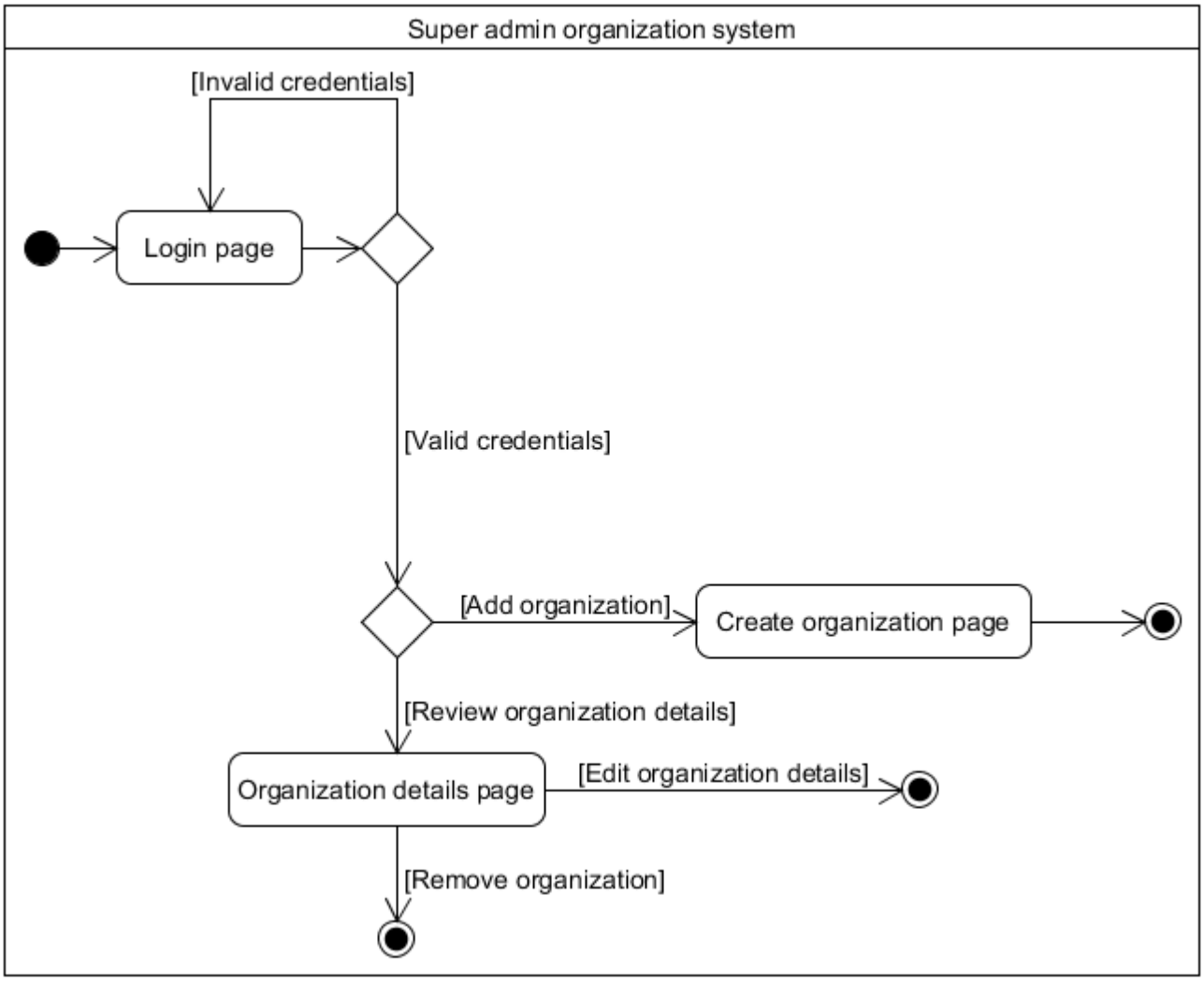

Project admin organization skills system
[Invalid credentials]
Login page
[Valid credentials]
[View Organisation skills]
Organisation skills page
[Manage Organisation skills]

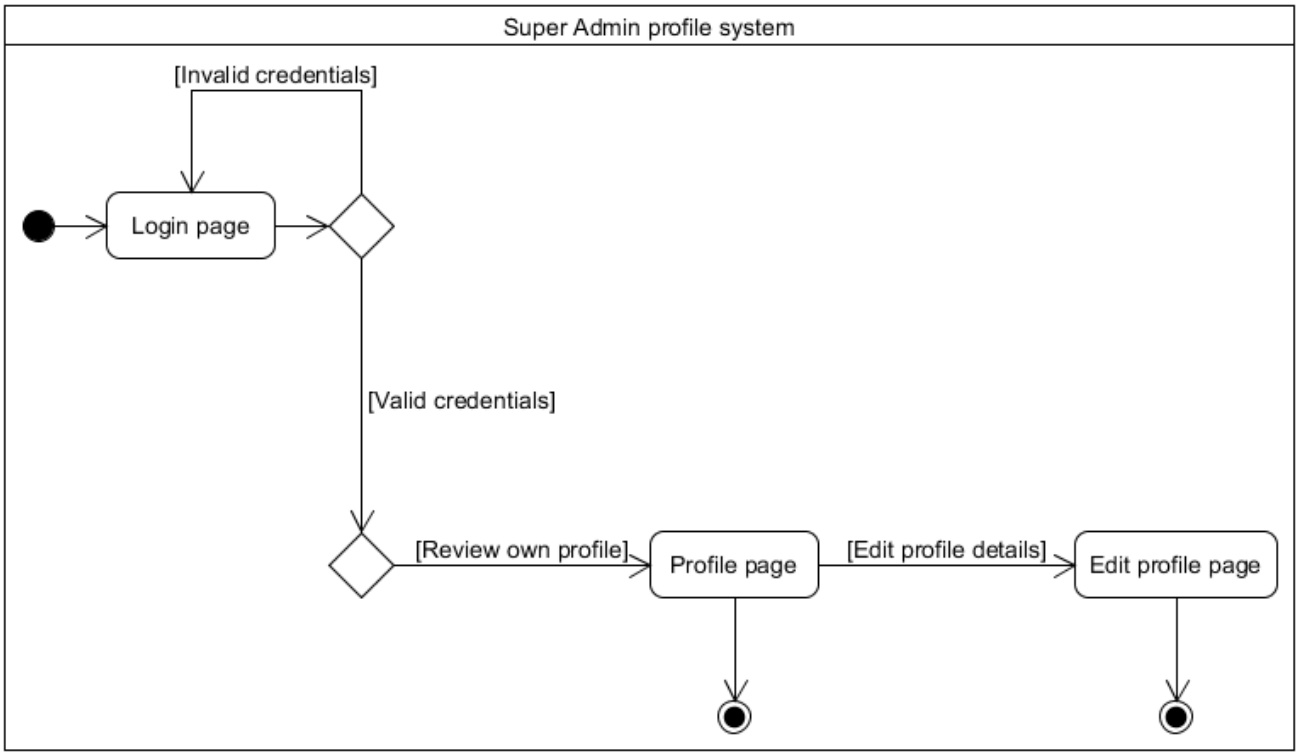

Project admin profile system
[Invalid credentials]
Login page
[Valid credentials]
[Review own profile]
Profile page
[Edit profile details]
Edit profile page


Project admin project system
[Invalid credentials]
Login page
[Valid credentials]
[Add project listings]
Add project page
[Review project details]
Project details page
[Edit Project details]
[Delete project]

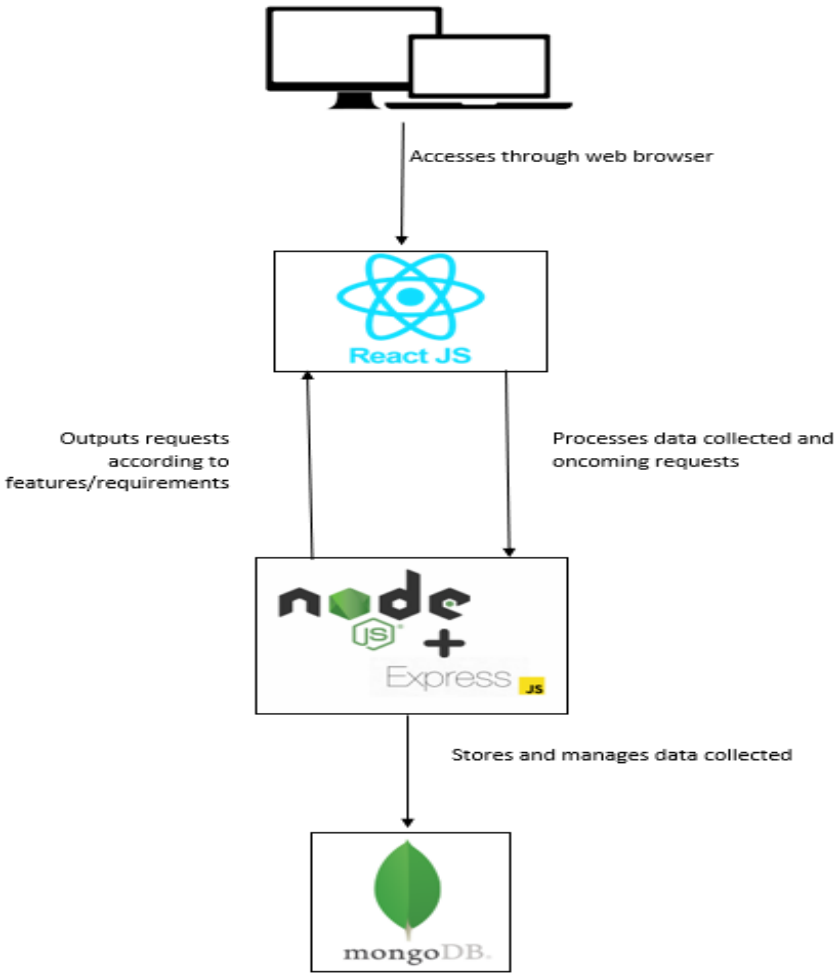
Super Admin


Super admin account management system
[Invalid credentials]
Login page
[Valid credentials]
[Review all user profile]
All users page
[Delete user profile]
[Create user profile]
[Review user details]
Create user page
User details page
[Edit user profile]


Super admin organization system
[Invalid credentials]
Login page
[Valid credentials]
[Add organization]
Create organization page
[Review organization details]
Organization details page
[Edit organization details]
[Remove organization]


Super Admin profile system
[Invalid credentials]
Login page
[Valid credentials]
[Review own profile]
Profile page
[Edit profile details]
Edit profile page

6. Architectural 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.

Technology Architecture

System Overview	
Hardware Interface	The product will utilize the MongoDB Atlas which is a document-oriented cloud server database to handle all backend functionalities.
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 Javascript-enabled web browser.
Connectivity Requirements	To access the web application, users are required to have an internet connection via Wi-Fi/Ethernet etc.

6. Database Design

Entities

Below are the entities that will be used for our database.

- Organisations
- Users
- Skills
- Projects

Relationships

Below are each entity interacting with one another.

- - Organisations → Users = 1:1..*
- - Organisations → Projects = 1:1..*
- - Organisations → Assignments = 1:1..*
- - Users → Projects = 1:1..*
- - Users → Skills = 1:1..*
- - Users → Assignments = 1..*:1
- - Users → Organisations = 1..*:1
- - Assignments → Organisations 1..*:1
- - Assignments → Projects = 1:1..*
- - Assignments → Users = 1:1..*
- - Projects → Skills = 1*:1..*
- - Projects → Users = 1:1..*
- - Projects → Organisations = 1..*:1
- - Projects → Assignments = 1..*:1
- - Skills → Users = 1..*:1

Attributes

Organization	
_id	ID
name	String
code	String

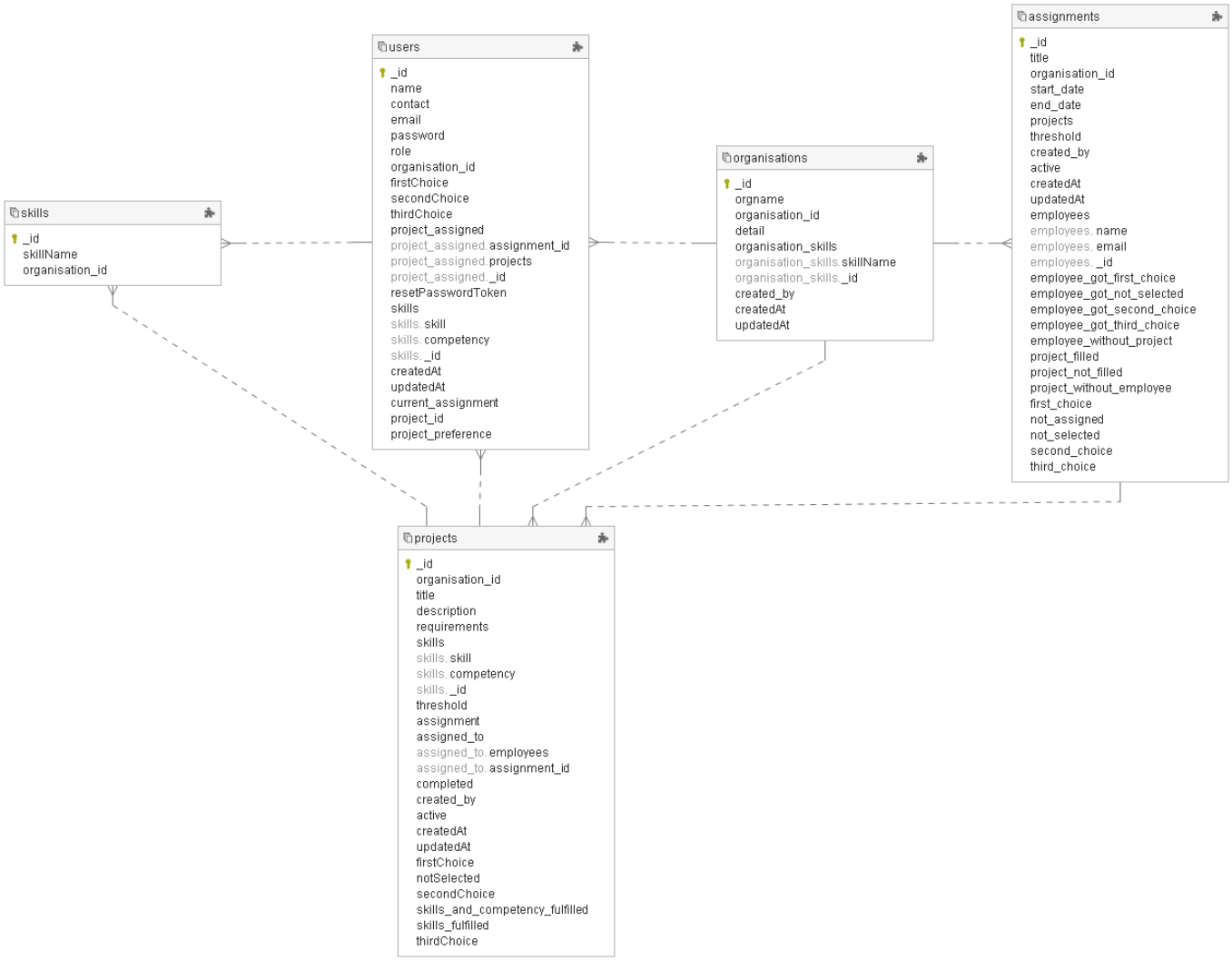
Projects	
_id	ID
organisation_id	ID
title	String
description	String
requirements	String
threshold	Int32
created_by	String
skills	Document[]
skills.skill	String
skills.competency	String
skills._id	ID
assignment	String
assigned_to	String
firstChoice	Int32
secondChoice	Int32
thirdChoice	Int32
notSelected	Int32
skills_and_competency_fulfilled	Int32
completed	Boolean
createdAt	DateTime
updatedAt	DateTime
active	Boolean

Skills	
_id	ID
skillName	String
organisation_id	String

Users	
_id	ID
organisation_id	ID
email	String
name	String
contact	Int32
password	String (Hashed)
role	String
project_preference	String[]
project_assigned	String[]
skills	String[]/Document[]
skills.skill	String
skills.competency	String
skills._id	ID
current_assignment	String
firstChoice	String
secondChoice	String
thirdChoice	String
createdAt	DateTime
updatedAt	DateTime

Assignments	
_id	ID
title	String
organisation_id	String
start_date	Date
end_date	Date
projects	String[]
threshold	Int32
employees	Object[]
employee_got_first_choice	Int32
employee_got_second_choice	Int32
employee_got_third_choice	Int32
employee_without_project	Int32
project_filled	Int32
project_not_filled	Int32
project_without_employee	Int32
created_by	String
active	Boolean
createdAt	Date
updatedAt	Date

Design


assignments
_id
title
organisation_id
start_date
end_date
projects
threshold
created_by
active
createdAt
updatedAt
employees
employees. name
employees. email
employees. _id
employee_got_first_choice
employee_got_not_selected
employee_got_second_choice
employee_got_third_choice
employee_without_project
project_filled
project_not_filled
project_without_employee
first_choice
not_assigned
not_selected
second_choice
third_choice
users
_id
name
contact
email
password
role
organisation_id
firstChoice
secondChoice
thirdChoice
project_assigned
project_assigned. assignment_id
project_assigned. projects
project_assigned. _id
resetPasswordToken
skills
skills. skill
skills. competency
skills. _id
createdAt
updatedAt
current_assignment
project_id
project_preference
organisations
_id
orgname
organisation_id
detail
organisation_skills
organisation_skills. skillName
organisation_skills. _id
created_by
createdAt
updatedAt
skills
_id
skillName
organisation_id
projects
_id
organisation_id
title
description
requirements
skills
skills. skill
skills. competency
skills. _id
threshold
assignment
assigned_to
assigned_to. employees
assigned_to. assignment_id
completed
created_by
active
createdAt
updatedAt
firstChoice
notSelected
secondChoice
skills_and_competency_fulfilled
skills_fulfilled
thirdChoice

7. Appendixes

Appendix A: MongoDB resources

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

Appendix B: Express.js resources

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

Appendix C: React resources

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

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

Appendix D: Node.js resources
https://www.tutorialspoint.com/nodejs/nodejs_introduction.htm