BSIT-S25-002

03-135221-013 Hassan Mubasher

03-135221-010 Muhammad Faris Hamza

03-135221-023 Muhammad Ilyas

Collab Design: A Web App for Collaborative UML and SRS Automation

In partial fulfilment of the requirements for the degree of
Bachelor of Science in Computer Science

Supervisor: Hoor Fatima Yousaf

Department of Computer Sciences
Bahria University, Lahore Campus

January 2026

C e r t i f i c a t e

We accept the work contained in the report titled

Collab Design: A Web App for Collaborative UML and FRS Automation

Written by

Hassan Mubasher

Muhammad Faris Hamza

Muhammad Ilyas

As a confirmation to the required standard for the partial fulfilment of the degree of Bachelor of Science in Computer Science.

Approved by:

Supervisor: **Hoor Fatima Yousaf**

_____ (Signature)

January 05, 2026

DECLARATION

We hereby declare that this project report is based on our original work except for citations and quotations which have been duly acknowledged. We also declare that it has not been previously and concurrently submitted for any other degree or award at Bahria University or other institutions.

Enrolment	Name	Signature
03-135221-013	HASSAN MUBASHER	
03-135221-010	MUHAMMAD FARIS HAMZA	
03-135221-023	MUHAMMAD ILYAS	

Date : January 05, 2026

Specially dedicated to

My beloved grandmother, mother and father

(**Hassan Mubasher**)

My beloved grandmother, mother and father

(**Muhammad Faris Hamza**)

My beloved grandmother, mother and father

(**Muhammad Ilyas**)

ACKNOWLEDGEMENTS

We would like to thank everyone who had contributed to the successful completion of this project. We would like to express our gratitude to our research supervisor, Miss Hoor Fatima Yousaf for her invaluable advice, guidance and her enormous patience throughout the development of the research.

In addition, we would also like to express our gratitude to our loving parent and friends who had helped and given us encouragement.

Hassan Mubasher

Muhammad Faris Hamza

Muhammad Ilyas

Project Title

Collab Design: A Web App for Collaborative UML and FRS Automation

Abstract

With the increasing trend of using agile methodologies and parallel workflow, there is a need for a platform that facilitates rapid creation of software requirements and diagrams. Our platform allows users to efficiently create, edit, and manage system requirements and diagrams in participation with their team. It generates automatic Diagrams through AI. To generate diagrams and SRS document, user will require to input requirements in different fields of an input form. With community features such as discussion forums, users can coordinate, share ideas, and provide feedback on each other's work. For discussion and collaboration on projects, user will require to create project and add team-members by inviting them via email. Role-based access control ensures integrity and security.

TABLE OF CONTENTS

LIST OF TABLES

List of Figures

CHAPTER 1

INRTODUCTION

1.1 Background

It is quite common for several small companies to lack software requirement management and diagramming tools that align with modern agile methodologies. Our aim with Collab Design is to provide a multi-tool platform and become the go-to software for small companies or academic needs related to diagrams and requirement management.

With the increasing use of agile methodologies in software development, clear and concise requirements for stakeholders are essential. Our platform, Collab Design, can generate use case diagrams from the project requirements. Version control allow teams to work efficiently and with confidence.

1.2 Aims and Objectives

The aim and objective of our web application is to provide software requirement management and diagramming tools with team coordination. It involves developing:

- Requirement management tool that includes creating, editing and management of requirements document.
- Diagramming tool to create UML diagrams (use case, sequence, activity, collaboration, ERD and WBS) in a drag and drop fashion.
- Cooperative work enabling both annotation and editing.
- Role based access to the projects for parallel workflow and security.
- Version control system that manages versions of requirements and diagrams.
- Discussion forum to build community and discuss workflows, best practices and find solutions to queries.

1.3 Problem Statement

In today's fast paced software development and widespread implementation of agile methodologies, small companies, teams and academics lack comprehensive tools to manage software requirements and diagramming. Existing solutions are not compatible for parallel workflow, expensive and time- consuming.

Our product "Collab Design" addresses these problems using team coordination on the same project for developing diagrams and software requirements. It involves a version control system providing team members the ability to collaborate confidently.

1.4 Project Scope:

The scope of the project is to build a collaborative diagramming and requirement management platform. Users can interact with the interface to develop, modify, and manage diagrams and requirements; thus facilitating seamless coordination across teams. The system features version control and role-based access control to simplify workflow. It includes automation of use case diagram from project requirements. Team members can annotate the work of other team members. It also incorporates UML diagram creation using predefined set of diagrams and shapes. Moreover, the scope includes building a community forum where user can share insights and raise questions to other users.

The scope of this project does not include the development of custom code generators or integration with specific project management or requirements management tools. The focus is on providing a standalone UML diagramming platform with requirement management, and data security features.

CHAPTER 2

2 LITERATURE REVIEW (and/or SRS)

2.1 Existing system study

There are several tools namely UMLetino, draw.io, Visio, eraser.io, and IBM DOORS that offer features and functionalities ranging from theme customization to role-based authentication. We plan to bring together select features from multiple design and project coordination tools into one unified platform. A brief introduction and features of related software programs are given below:

2.1.1 UMLetino

It is a lightweight tool specializing in the creation of UML diagrams. It offers very basic, robust and comprehensive design functionalities along with file management and tracking, but doesn't provide any teamwork features, requirement management and discussion forums. [1]

2.1.2 Draw.io

Draw.io is an integrated online diagramming application. You can import/export files to various formats, the tool supports themes, provide various diagrams; however, it does not provide any solution for requirement management and team coordination. [3]

2.1.3 Visio

Microsoft Visio is a very diagram-heavy application that has full control over themes, diagrams, customization, and file management. OAuth support for secure access is available, however; it does not provide secure sharing, requirement management and team management. [5]

2.1.4 Eraser.io

Eraser.io provides a text editor, requirement management and diagramming dashboard. It provides essential diagramming and editing tools. However, it lacks customization,

variation in diagrams, team management, importing files and requirement traceability. [2]

2.1.5 IBM DOORS

IBM DOORS is a requirements management tool, which includes facilities like role-based authentication and offers a dashboard. On the downside, it cannot produce visual models directly from requirements because it does not support diagramming capabilities. [4]

2.1.6 Tools Comparison

This table compares specific features present in select tools; as is clearly evident Collab Design is the only tool containing all of these specific features.

Table 1 Tools Comparison

Feature	UMLetino	Draw.io	Visio	Eraser	DOORS	Collab Design
Dashboard	-	-	-	-	√	√
Themes	-	√	√	√	√	√
Discussion Forum	-	-	-	-	-	√
Ability to create use case diagram from requirements	-	-	-	-	-	√
Role-based authentication	-	-	-	-	√	√
OAuth	-	√	√	√	-	√
Import/ export in multiple formats	-	√	√	-	-	√
Requirement management	-	-	-	√	√	√
Diagramming Tools	√	√	√	√	-	√
Version Control	√	-	-	-	√	√

The table is displaying different existing tools comparison in terms their features.

2.2 Stakeholders list (Actors)

The key stakeholders in this project are following:

- System designer
- Software developers
- Students / Teachers
- System analysts

2.3 Requirement Elicitation

2.3.1 Functional Requirements

Table 2 Functional Requirements for User Management and Authentication

Req. No	Functional Requirements
FR-01-01	The system shall allow users to get registered via emails and valid password (minimum 8 characters with at least 1 lowercase and uppercase) or OAuth (Google and GitHub)
FR-01-02	The system shall allow users to log in using registered email and password set, OAuth and two factor authentications. The two-factor authentication code is sent via email.
FR-01-03	The system shall allow users to reset their password using OTP that is sent via email. The correct OTP allows the user to insert a new valid password for the account.
FR-01-04	The system shall allow users to log out securely and end the session.

2.3.2 User Dashboard and Personaliza

Table 3 Functional Requirements for User Dashboard and Personalization

Req. No	Functional Requirements
FR-02-01	The system shall have a dashboard display that gives access to recent activities, notifications and projects.
FR-02-02	The system shall allow users to edit personal information, themes and storage settings.

Table 4 Functional Requirements for Collaboration and Team Management

Req. No	Functional Requirements
FR-03-01	The system shall allow users to create new project teams by specifying project names and team members.
FR-03-02	The system shall allow users to edit and archive projects.

Table 5 Functional Requirement for Requirements Management

Req. No	Functional Requirements
FR-04-01	The system shall allow users to enter requirements by entering name, description and assign priority (High, Medium, and Low) to them.
FR-04-02	The system shall allow users to edit requirements.
FR-04-03	The system shall allow the user to search requirements via keywords.
FR-04-04	The system shall view requirements in standard format.
FR-04-05	The system shall enable version control to requirements via entering version number. The system allows the user to navigate to any previously saved version.
FR-04-06	The system shall allow users to annotate requirements.
FR-04-07	The system shall enable collaborative access to requirements via a unique generated link.
FR-04-08	The system shall create use case diagrams using requirements.
FR-04-09	The system shall download format documents or store them in Google drive.
FR-04-10	The system shall allow the user to upload previously downloaded files to extract requirements.

Table 6 Functional Requirements for Diagramming & Visual Design

Req. No	Functional Requirements
FR-05-01	The system shall allow users to create diagrams via selecting diagrams objects and arrows.
FR-05-02	The system shall allow users to edit and delete diagrams.
FR-05-3	The system shall let users customize canvas via changing canvas colors.
FR-05-04	The system shall enable version control to diagram via entering version number. The system allows the user to navigate to any previously saved version.
FR-05-05	The system shall allow users to comment on and annotate diagrams.
FR-05-06	The system shall enable collaborative access to diagrams via a unique generated link.
FR-05-07	The system shall let users download diagrams in PNG format.
FR-05-08	The system shall let users upload previously downloaded diagram files.

2.3.3 Discussion Form

Table 7 Functional Requirements for Discussion Forum

Req. No	Functional Requirements
FR-06-01	The system shall allow users to create new discussion threads by specifying a title, description, and category.
FR-06-02	The system shall allow users to edit or delete their own posts. It includes basic text editing functionalities (Bold, Italic, Underline etc.).
FR-06-03	The system shall allow users to search within forums based on keywords or topics.
FR-06-04	The system shall allow users to comment and vote for existing threads.

2.3.4 Non-Functional Requirements

2.3.4.1 Security

Table 8 Non-Functional Requirements for Security

Req. No	Non-Functional Requirements
NFR-01-01	All sensitive data including passwords and OTP must be encrypted.
NFR-01-02	No user can view data of another user using any source.

2.3.4.2 Reliability

Table 9 Non-Functional Requirements for Reliability

Req. No	Non-Functional Requirements
NFR-02-01	System must have 99.9% uptime ensuring availability to users.

2.3.4.3 Usability

Table 10 Non-Functional Requirements for Usability

Req. No	Non-Functional Requirements
NFR-03-01	The system shall provide tooltips for key functionalities.
NFR-03-02	The user interface shall follow modern design principles to ensure intuitive navigation and ease of use.

2.3.4.4 Maintainability

Table 11 Non-Functional Requirements for Maintainability

Req. No	Non-Functional Requirements
NFR-04-01	The system codebase shall follow standard coding practices for MVC architecture and be well-documented for ease of maintenance and future updates.

2.3.4.5 Performance

Table 12 Non-Functional Requirements for Performance

Req. No	Non-Functional Requirements
NFR-05-01	The system shall support up to 100 concurrent users without performance degradation.
NFR-05-02	The system shall respond to all user requests within 1 second on average.

2.3.4.6 Data Integrity

Table 13 Non-Functional Requirements for Data Integrity

Req. No	Non-Functional Requirements
NFR-06-01	The system shall prevent unauthorized access or modification of data and ensure data consistency across different modules.

2.3.4.7 Interoperability

Table 14 Non- Functional Requirements for Interoperability

Req. No	Non-Functional Requirements
NFR-07-01	All system components shall adhere to a common, standard set of data exchange formats (JSON) to ensure seamless interoperability.

2.3.5 Requirement Traceability Matrix

Requirement Traceability Matrix

Table 15 Requirement Traceability Matrix

Sr#	Requirement	Technical Assumption	Functional Requirement	Actors	Priority	Assign to	Use case ID	Activity Diagram	Sequence Diagram
1	FR-01-01	User has an email	Register user	User	High	Hassan	001	01	01
2	FR-01-02	User is already registered	Login user	User	High	Hassan	002	01	02
3	FR-01-03	User has entered valid OTP	Reset password	User	Medium	Hassan	003	02	-
4	FR-01-04	User is already logged in	Logout user	User	Low	Hassan	004	-	-
5	FR-02–01	User is already logged in	Display dashboard	User	High	Ilyas	005	-	-
6	FR-02-02	User is logged in	Edit user details	User	Medium	Ilyas	006	-	03
7	FR-03-01	User is logged in	Create team	User	High	Faris	007	03	04
8	FR-03-02	User is project leader	Edit project	User	Medium	Faris	008	04	-
9	FR-04-01	User is team member	Create requirement	User	High	Ilyas	009	05	-
10	FR-04-02	User is team member	Edit requirement	User	Medium	Hassan	010	-	05
11	FR-04-03	Requirement exists	Search requirement	User	Medium	Ilyas	011	-	-
12	FR-04-04	Requirement exists	View requirement	User	Medium	Faris	-	-	-
13	FR-04-05	Requirement exists	Version control for requirements	User	Medium	Hassan	-	-	-
14	FR-04-06	Requirement exists	Comment requirement	User	Medium	Ilyas	012	-	-
15	FR-04-07	User is a team member	Access requirement	User	Medium	Faris	-	-	-

16	FR-04–08	Requirement exists	Create use case diagram	User	High	Hassan	013	-	-
17	FR-04-09	Document exists	Store documents	User	High	Faris	014	06	06
18	FR-04-10	User has file	Upload files	User	Medium	Ilyas	-	-	-
19	FR-05-01	User is a team member	Create diagram	User	High	Hassan	021	-	-
20	FR-05-02	User is a team member	Edit/delete diagram	User	Medium	Faris	022	09	09
21	FR-05-03	User is a team member	Customise canvas	User	Medium	Ilyas	025	-	-
22	FR-05-04	User is a team member	Version control for diagrams	User	Medium	Hassan	-	-	-
23	FR-05-05	User is a team member	Comment on diagram	User	Medium	Faris	026	-	-
24	FR-05-06	User is a team leader	View diagram	User	Medium	Ilyas	-	-	-
25	FR-05-07	User is a team member	Download diagram	User	High	Hammad	024	08	-
26	FR-05-08	User has diagram file	Upload diagram	User	Medium	Faisal	-	-	-
27	FR-06-01	User is registered	Create thread	User	High	Arsal	015	-	07
28	FR-06-02	User created thread	Edit thread	User	Medium	Hassan	016	07	-
29	FR-06-03	Forum exists	Search forum	User	Medium	Faris	020	-	08
30	FR-06-04	Forum exists	Comment or vote on threads	User	Medium	Ilyas	018	-	-

Requirement Traceability Matrix (RTM) is a table that **maps and tracks requirements** through the entire project lifecycle.

It ensures:

- Every requirement → has at least one test case
- Nothing important is missed
- Testing covers all functionality
- Developers, testers, and stakeholders stay aligned

2.4 Use Cases of System

2.4.1 Register User

Table 16 Use case description for registering user

Use Case ID:	001	Use case Name:	Register User
Priority:	Medium		
Actor:	User		
Use case Summary:	The user registers by providing a valid email, password, or using OAuth (Google/GitHub).		
Pre-conditions:	The user must not already be registered in the system.		
Normal Flow of Path		Alternative Path	
1. User accesses the registration form.		1. User selects OAuth service.	
2. User enters a valid email and password (minimum 8 characters, at least 1 uppercase and lowercase)		2. Users authenticate with Google or GitHub.	
3. System validates the email and password		3. System retrieves the user profile from the provider and creates an account.	
4. An email is sent on provided email with an OTP. The user inserts OTP in the provided form.		Same as normal path	
5. If valid, the user account is registered, confirmation email is sent to the user.		Same as normal path	
Exceptions/Alerts:			
1. If the email is already registered, the system shows an error. 2. If the email/password format is invalid, the system prompts the user to correct it.			
Post-conditions:			
Step number		**Description**	
01		User is logged in and redirected to dashboard	
Use case Cross References:			
Includes:		None	
Extends:		None	

2.4.2 Login User

Table 17 Use case description for logging user in

Use Case ID:	002	Use case Name:	Login User
Priority:	High		
Actor:	User		
Use case Summary:	The user logs in using email/password, OAuth, or two-factor authentication.		
Pre-conditions:	The user must have a registered account.		
Normal Flow of Path		Alternative Path	
1. User accesses the login page. 2. User enters an email and password. 3. System validates the email and password. and prompt two factor authentication code (sent to email) 4. User enters the code sent via email. 5. The user login to system		1. User selects OAuth service. 2. Users authenticate with Google or GitHub. 3. System retrieves the user profile from the provider and logs in the user. Same as normal path Same as normal path	
Exceptions/Alerts:			
1. Incorrect email/password: System shows an error. 2. Incorrect two-factor code: System prompts for re-entry.			
Post-conditions:			
Step number		**Description**	
01		User is logged in, their session is started securely, and user is redirected to dashboard	
Use case Cross References:			
Includes:		None	
Extends:		Two Factor Authentication	

2.4.3 Reset Password

Table 18 Use case description for resetting password

Use Case ID:	003	Use case Name:	Reset Password
Priority:	High		
Actor:	User		
Use case Summary:	The user resets their password using an OTP sent via email.		
Pre-conditions:	The user must have a registered email address.		
Normal Flow of Path		Alternative Path	
1. User accesses the "forgot password" page.		Same as normal path	
2. User enters a registered email address.		Same as normal path	
3. System validates the email and sends an OTP to the email if two factor authentications is activated otherwise jumps to step 6.		Same as normal path	
4. User enters the code sent via email.		Same as normal path	
5. System verifies the OTP.		5a. Invalid OTP: System prompts for re-entry or re-sending.	
6. User sets a new password that meets the format requirements.		6a. New password format invalid: System prompts for corrections.	
Exceptions/Alerts:			
1. Incorrect email: System shows an error. 2. Did not receive code: User can ask to resend OTP after 1-minute duration.			
Post-conditions:			
Step number		**Description**	
01		The password is updated, and the user can log in with the new password.	
Use case Cross References:			
Includes:		OTP Verification	
Extends:		None	

2.4.4 Logging Out User

Table 19 Use case description for logging out user

Use Case ID:	004	Use case Name:	Log out user
Priority:	High		
Actor:	User		
Use case Summary:	The user securely logs out and ends their session.		
Pre-conditions:	The user must be logged in.		
Normal Flow of Path		Alternative Path	
1. User clicks the log out button.		Same as normal path	
2. System terminates the session.		Same as normal path	
3. User redirected to Landing Page		Same as normal path	
Exceptions/Alerts:			
1. System error during logout: Notify user of failure.			
Post-conditions:			
Step number		**Description**	
01		The session is securely ended, and user data is cleared from the device.	
Use case Cross References:			
Includes:		None	
Extends:		None	

2.4.5 View Dashboard

Table 20 Use case description for viewing dashboard

Use Case ID:	005	Use case Name:	View Dashboard
Priority:	Medium		
Actor:	User		
Use case Summary:	The user views a personalized dashboard with owned and shared projects.		
Pre-conditions:	The user must be logged in.		
Normal Flow of Path		Alternative Path	
1. User logs in and accesses the dashboard.		Same as normal path	
2. System retrieves notifications, project details and relevant data.		Same as normal path	
3. Dashboard displays the data.		Same as normal path	
Exceptions/Alerts:			
1. If data retrieval fails, show a relevant error message.			
Post-conditions:			
Step number		**Description**	
1.		The user can view and interact with their dashboard.	
Use case Cross References:			
Includes:		None	
Extends:		None	

2.4.6 Edit Personal Information

Table 21 Use case description for editing personal information

Use Case ID:	006	Use case Name:	Edit Personal Information
Priority:	Medium		
Actor:	User		
Use case Summary:	The user updates their personal information, themes, or storage settings.		
Pre-conditions:	The user must be logged in.		
Normal Flow of Path		Alternative Path	
1. User navigates to the settings page.		Same as normal path	
2. User edits personal information, theme, or storage settings.		Same as normal path	
3. System validates the input.		3a. Invalid input: System prompts for corrections.	
4. System saves the changes.			
Exceptions/Alerts:			
1. If the data update fails, show a relevant error message.			
Post-conditions:			
Step number		**Description**	
1.		Updated information is saved and reflected in the system.	
Use case Cross References:			
Includes:		None	
Extends:		None	

2.4.7 Creating Teams

Table 22 Use case description for creating teams

Use Case ID:	007	Use case Name:	Creating teams
Priority:	Medium		
Actor:	User		
Use case Summary:	User can create teams		
Pre-conditions:	User must be signed in		
Normal Flow of Path		Alternative Path	
1. Use case starts when user wants to create a new team		Same as normal path	
2. The user will press the create team button		Same as normal path	
3. The system responds by asking the user to specify team name, members and assign roles		Same as normal path	
4. The user confirms the creation of a new team		Same as normal path	
5. The system responds by creating a new team according to user specifications		Same as normal path	
Exceptions/Alerts:			
1. The system will not allow user to create teams without required information			
Post-conditions:			
Step number		**Description**	
1.		A new team will be created by the system	
2.		The team members will be notified by the system	
Use case Cross References:			
Includes:		None	
Extends:		None	

2.4.8 Editing Teams

Table 23 Use case description for editing teams

Use Case ID:	008	Use case Name:	Editing teams
Priority:	Medium		
Actor:	User		
Use case Summary:	User can edit or delete existing teams		
Pre-conditions:	1. User must be signed in 2. Team must already exist		
Normal Flow of Path		Alternative Path	
1. Use case starts when the user wants to edit a team		1a. User wants to delete a team	
2. The user selects the team			
3. The system responds by asking the user to specify team name, members and assign roles		3a. User presses delete button	
4. The user confirms the changes		4a. System responds by asking user to confirm deletion	
5. The system responds by editing the team according to user specifications		5a. System responds by deleting the team	
Exceptions/Alerts:			
1. The system will not allow user to edit teams without required information			
Post-conditions:			
Step number		**Description**	
1.		Team is modified or deleted	
Use case Cross References:			
Includes:		None	
Extends:		Create team	

2.4.9 Entering Requirements

Table 24 Use case description for entering requirements

Use Case ID:	009	Use case Name:	Entering requirements
Priority:	Medium		
Actor:	User		
Use case Summary:	User can add requirements along with descriptions and assign priorities to them		
Pre-conditions:	1. User must be signed in 2. Project must already exist		
Normal Flow of Path		Alternative Path	
1. Use case starts when the user wants to enter some new requirements		Same as normal path	
2. User will select the project of which user wants to enter requirements		Same as normal path	
3. The user then clicks on the new requirement button		Same as normal path	
4. The system responds by asking using to enter the requirement details		Same as normal path	
5. The user then clicks on the save button to save the requirement		Same as normal path	
6. The system responds by creating a new requirement		Same as normal path	
Exceptions/Alerts:			
1. The system will not allow user to create a new requirement without compulsory details 2. Two requirements cannot have the same ID			
Post-conditions:			
Step number		**Description**	
1.		A new requirement is created	
Use case Cross References:			
Includes:		None	
Extends:		None	

2.4.10 Editing Requirements

Table 25 Use case description for editing requirements

Use Case ID:	010	Use case Name:	Editing requirements
Priority:	Medium		
Actor:	User		
Use case Summary:	User can edit already existing requirements		
Pre-conditions:	1. User must be signed in 2. A project with specified requirements must exist		
Normal Flow of Path		Alternative Path	
1. Use case starts when user wants to edit an already specified requirement		Same as normal path	
2. The user selects the project for which the requirement needs to be edited		Same as normal path	
3. The user selects the requirement which they want to edit		Same as normal path	
4. The user presses the edit button		Same as normal path	
5. The system responds by asking using to enter the requirement details		Same as normal path	
6. User edits the details and presses the save button		Same as normal path	
7. The system responds by updating the requirement details		Same as normal path	
Exceptions/Alerts:			
1. The system will not allow user to edit requirement without compulsory details 2. Two requirements cannot have the same ID			
Post-conditions:			
Step number		**Description**	
1.		A requirement is updated	
Use case Cross References:			
Includes:		None	
Extends:		Entering requirements	

2.4.11 Search Requirements

Table 26 Use case description for searching requirements

Use Case ID:	011	Use case Name:	Search requirements
Priority:	Medium		
Actor:	User		
Use case Summary:	User can search for existing requirements using keywords		
Pre-conditions:	1. User must be signed in 2. A requirement containing the keyword must exist		
Normal Flow of Path		Alternative Path	
1. Use case starts when user wants to search for a requirement in a project		Same as normal path	
2. The user selects the project in which the requirement exists		Same as normal path	
3. The user enters the keyword in the search bar		Same as normal path	
4. System responds by showing search results		Same as normal path	
Exceptions/Alerts:			
1. If the keyword is not present in any of the requirements the system will show no results			
Post-conditions:			
Step number		**Description**	
1.		The system displays a list of the requirements containing the keyword	
Use case Cross References:			
Includes:		None	
Extends:		Entering requirements	

2.4.12 Commenting Requirements

Table 27 Use case description for commenting requirements

Use Case ID:	012	Use case Name:	Comment requirements
Priority:	Low		
Actor:	User		
Use case Summary:	User can add comments		
Pre-conditions:	1. User must be signed in 2. User must be a member of the project team		
Normal Flow of Path		Alternative Path	
1. Use case starts when a user wants to comment on a requirement		Same as normal path	
2. The user selects the project in which the requirement exists		Same as normal path	
3. The user selects the requirement on which they want to comment		Same as normal path	
4. The user then writes their comment under “Add comment” and presses the comment button		Same as normal path	
5. The system responds by adding the comment under the requirement		Same as normal path	
Exceptions/Alerts:			
Post-conditions:			
Step number		**Description**	
1.		The system displays a list of all comments under the requirement	
Use case Cross References:			
Includes:		None	
Extends:		None	

2.4.13 Creating Use Case Diagram from Requirements

Table 28 Use case description for creating use case diagrams from requirements

Use Case ID:	013	Use case Name:	Create use case diagrams from requirements
Priority:	Medium		
Actor:	User		
Use case Summary:	User can create use case diagrams by entering requirements		
Pre-conditions:	1. User must be signed in 2. User must be a member of the project team		
Normal Flow of Path		Alternative Path	
1. Use case starts when the user wants to automate the creation of use case diagrams		Same as normal path	
2. The user fills out the requirement form		Same as normal path	
3. The user then presses the generate button		Same as normal path	
4. The system responds by generating a use case diagram		Same as normal path	
Exceptions/Alerts:			
1. The form is incomplete			
Post-conditions:			
Step number		**Description**	
1.		The use case diagram will be generated and will be ready for use	
Use case Cross References:			
Includes:		None	
Extends:		None	

2.4.14 Download Documents

Table 29 Use case description for downloading or uploading files/documents

Use Case ID:	014	Use case Name:	Download documents
Priority:	Medium		
Actor:	User		
Use case Summary:	User can download files in select formats		
Pre-conditions:	1. User must be signed in		
Normal Flow of Path		Alternative Path	
1. Use case starts when the user wants to download a file		Same as normal path	
2. User presses the download icon		Same as normal path	
3. User selects the format and presses the download button		Same as normal path	
4. System responds by downloading the file to the system in the specified format and location		Same as normal path	
Exceptions/Alerts:			
Post-conditions:			
Step number		**Description**	
1.		The file will be downloaded to the specified location in the selected format	
Use case Cross References:			
Includes:		None	
Extends:		None	

2.4.15 Create Thread

Table 30 Use case description for creating/posting threads

Use Case ID:	015	Use case Name:	Create Thread
Priority:	Medium		
Actor:	User		
Use case Summary:	Users can create a thread in the Collab Design community by filling out the title and description of the thread.		
Pre-conditions:	1. Users must be signed in.		
Normal Flow of Path		Alternative Path	
1. User navigates to the discussion forum page.		Same as normal path	
2. User inputs the title and description for the thread.		Same as normal path	
3. User posts the thread.		Same as normal path	
Exceptions/Alerts:			
1. Thread will not be created if the network is not available.			
Post-conditions:			
Use case Cross References:			
Includes:		None	
Extends:		None	

2.4.16 Editing Threads

Table 31 Use case description for editing threads

Use Case ID:	016	Use case Name:	Edit Thread
Priority:	Medium		
Actor:	User		
Use case Summary:	Users can edit the content of the thread and its styling after the thread is posted.		
Pre-conditions:	1. Users must be signed in. 2. Threads to be edited should be already created.		
Normal Flow of Path		Alternative Path	
1. Use case starts when the user wants to edit a thread they created.		Same as normal path	
2. User navigates to the discussion forum page.		Same as normal path	
3. User selects the filter called '*my threads*' to see the threads they created.		Same as normal path	
4. User selects the thread they want to edit.		Same as normal path	
5. User makes the changes.		Same as normal path	
6. User saves the changes they made.		Same as normal path	
Exceptions/Alerts:			
1. If the network is not available changes will not be saved. 2. Users will be notified that the edited thread is now live.			
Post-conditions:			
Use case Cross References:			
Includes:		Create Thread	
Extends:		None	

2.4.17 Deleting Threads

Table 32 Use case description for deleting threads

Use Case ID:	017	Use case Name:	Delete Thread
Priority:	Medium		
Actor:	User		
Use case Summary:	Users can delete the thread they created.		
Pre-conditions:	1. Users must be signed in. 2. Thread to be deleted should already be created.		
Normal Flow of Path		Alternative Path	
1. Use case starts when the user wants to delete a thread they created.		Same as normal path	
2. User navigates to the discussion forum page.		Same as normal path	
3. User selects the filter called '*my threads*' to see the threads they created.		Same as normal path	
4. User selects the thread they want to delete.		Same as normal path	
5. A confirmation message pops up asking the user if they are sure they want to delete that thread.		Same as normal path	
6. After the user selects "yes" the confirmation message pop up closes.		Same as normal path	
7. The selected thread gets deleted.		Same as normal path	
Exceptions/Alerts:			
1. If the network is not available, the selected thread will not be deleted.			
Post-conditions:			
Use case Cross References:			
Includes:	Create Thread		
Extends:	None		

2.4.18 Commenting Threads

Table 33 Use case description for commenting threads

Use Case ID:	018	Use case Name:	Comment Thread
Priority:	Medium		
Actor:	User		
Use case Summary:	Users can comment on the thread they created or other users created.		
Pre-conditions:	1. Users must be signed in. 2. Thread to be commented on should already be created.		
Normal Flow of Path		Alternative Path	
1. Use case starts when the user wants to comment on a thread.		Same as normal path	
2. User navigates to the discussion forum page.		Same as normal path	
3. User selects the thread they want to comment on.		Same as normal path	
4. User inputs the text for the comment.		Same as normal path	
5. User posts the comment on the thread.		Same as normal path	
Exceptions/Alerts:			
1. If the network is not available, comments will not be posted.			
Post-conditions:			
Step number		**Description**	
1.		The owner of the thread will be notified about the comment.	
Use case Cross References:			
Includes:		Create Thread	
Extends:		None	

2.4.19 Voting Threads

Table 34 Use case description for voting threads

Use Case ID:	019	Use case Name:	Vote Thread
Priority:	Medium		
Actor:	User		
Use case Summary:	Users can upvote or downvote on a thread to show their feedback to the creator of the thread.		
Pre-conditions:	1. Users must be signed in. 2. Thread to be voted should already be created.		
Normal Flow of Path		Alternative Path	
1. Use case starts when the user wants to up or downvote a thread.		Same as normal path	
2. User navigates to the discussion forum page.		Same as normal path	
3. User selects the thread they want to give their feedback on.		Same as normal path	
4. User selects the down or upvote option accordingly.		Same as normal path	
Exceptions/Alerts:			
1. If the network is not available, the vote will not be counted.			
Post-conditions:			
Step number		**Description**	
1.		The owner of the thread will be notified about the vote.	
Use case Cross References:			
Includes:		Create Thread	
Extends:		None	

2.4.20 Search Forum

Table 35 Use case description for searching through discussion forum

Use Case ID:	020	Use case Name:	Search Forum
Priority:	Medium		
Actor:	User		
Use case Summary:	Users can upvote or downvote on a thread to show their feedback to the creator of the thread.		
Pre-conditions:	1. Users must be signed in. 2. Thread to be voted should already be created.		
Normal Flow of Path		Alternative Path	
1. Use case begins when the user wants to search a thread.		Same as normal path	
2. User navigates to the discussion forum page.		Same as normal path	
3. User inputs the keywords or title of the thread in the search bar.		Same as normal path	
4. System responds by showing search results.		Same as normal path	
Exceptions/Alerts:			
1. If the keyword is not present in any of the requirements the system will show no results.			
Post-conditions:			
Step number		**Description**	
1.		The system displays a list of the threads with matching titles or descriptions containing the keyword.	
Use case Cross References:			
Includes:		Create Thread	
Extends:		None	

2.4.21 Create Diagram

Table 36 Use case description for creating a diagram

Use Case ID:	021	Use case Name:	Create Diagram
Priority:	High		
Actor:	User		
Use case Summary:	Users can create a diagram in a project that the user is a member or owner of.		
Pre-conditions:	1. Users must be signed in. 2. A project must be created. 3. Users must be a member of the created project.		
Normal Flow of Path		Alternative Path	
1. Use case begins when the user wants to create a diagram in a project, they are a member or owner of.		Same as normal path	
2. User navigates to the dashboard.		Same as normal path	
3. User selects the project in which they want to create the diagram.		Same as normal path	
4. User clicks on the ‘new diagram’ button.		Same as normal path	
Exceptions/Alerts:			
1. If the network is not present the user will not be navigated to the canvas page.			
Post-conditions:			
Step number		**Description**	
1.		This navigates the user to an empty canvas where the user can draw the diagram they want.	
Use case Cross References:			
Includes:		Create Project	
Extends:		None	

2.4.22 Edit Diagram

Table 37 Use case description for editing a diagram

Use Case ID:	022	Use case Name:	Edit Diagram
Priority:	High		
Actor:	User		
Use case Summary:	Users can edit a diagram in a project that the user is a member or owner of.		
Pre-conditions:	1. Users must be signed in. 2. Users must be a member of that project		
Normal Flow of Path		Alternative Path	
1. Use case starts when a user wants to edit a diagram present in a project.		Same as normal path	
2. User navigates to the dashboard page.		Same as normal path	
3. User selects the project that contains the diagram.		Same as normal path	
4. User selects the diagram they want to edit.		Same as normal path	
5. User is navigated to the canvas page where the selected diagram file is opened for the user to edit.		Same as normal path	
Exceptions/Alerts:			
1. If network is not available changes to the diagram are not saved			
Post-conditions:			
Step number		**Description**	
1.		The edited diagram is saved to the user's Google drive.	
2.		Members of the project are notified about the changes made to the diagram.	
Use case Cross References:			
Includes:		Create Project, Create Diagram	
Extends:		None	

2.4.23 Delete Diagram

Table 38 Use case description for deleting a diagram

Use Case ID:	023	Use case Name:	Delete Diagram
Priority:	High		
Actor:	User		
Use case Summary:	Users can delete a diagram in a project that the user is an owner or member of.		
Pre-conditions:	1. Users must be signed in. 2. User must be a member or owner of that project.		
Normal Flow of Path		Alternative Path	
1. Use case starts when the user wants to delete a diagram in a project, they're a member or owner of.		Same as normal path	
2. User navigates to the dashboard page.		Same as normal path	
3. User selects the project that contains the diagram they want to delete.		Same as normal path	
4. User selects the diagram they want to delete.		Same as normal path	
5. User clicks on the delete button.		Same as normal path	
6. Users select 'yes' on the confirmation message asked by the system on deletion of a diagram.		Same as normal path	
Exceptions/Alerts:			
1. The selected diagram is not deleted if the network is not available.			
Post-conditions:			
Step number		**Description**	
1.		All of the team members of the project are notified about the deletion of the specific diagram.	
Use case Cross References:			
Includes:		Create Project, Create Diagram	
Extends:		None	

2.4.24 Download Diagram

Table 39 Use case description for downloading a diagram

Use Case ID:	024	Use case Name:	Download Diagram
Priority:	High		
Actor:	User		
Use case Summary:	Users can download a diagram in a project that the user is an owner or member of.		
Pre-conditions:	1. Users must be signed in. 2. A project that contains the diagram must exist. 3. User must be a member or owner of that project 4. Diagrams to be downloaded must exist.		
Normal Flow of Path		Alternative Path	
1. Use case starts when the user wants to download a diagram in a project that they're a member or owner of.		Same as normal path	
2. User navigates to the dashboard page.		Same as normal path	
3. User selects the project that contains the diagram.		Same as normal path	
4. User selects the diagram they want to download.		Same as normal path	
5. Users specify the file format in which they want to download the diagram.		Same as normal path	
Exceptions/Alerts:			
1. The user is notified when the download commences.			
Post-conditions:			
Step number		**Description**	
1.		The downloaded diagram is saved on the user's device.	
Use case Cross References:			
Includes:		Create Project, Create Diagram	
Extends:		None	

2.4.25 Customize Canvas

Table 40 Use case description for customizing diagram canvas

Use Case ID:	025	Use case Name:	Customize Canvas
Priority:	Medium		
Actor:	User		
Use case Summary:	Users can customize the background colour and enable or disable grids of the diagram canvas.		
Pre-conditions:	1. Users must be signed in. 2. A project must be created. 3. User must be a member of that project		
Normal Flow of Path		Alternative Path	
1. Use case starts when the user is creating a diagram and they want to customize the look and feel of the canvas.		Same as normal path	
2. User clicks on the paint brush icon.		2- User clicks on the grid icon.	
3. User specifies the colour of the canvas by hex code or by colour name in the pop-up menu that opens upon clicking the icon.		Same as normal path	
Exceptions/Alerts:			
1. The colour name specified does not correspond to any colour.			
Post-conditions:			
Step number		**Description**	
1.		User specified changes are reflected real-time.	
Use case Cross References:			
Includes:		Create Project, Create Diagram	
Extends:		None	

2.4.26 Comment Diagram

Table 41 Use case description for commenting a diagram

Use Case ID:	026	Use case Name:	Comment diagrams
Priority:	Low		
Actor:	User		
Use case Summary:	Users can add comments to a diagram in a project that they're a member or owner of.		
Pre-conditions:	1. User must be signed in 2. Users must be a member or owner of the project.		
Normal Flow of Path		Alternative Path	
1. Use case starts when a user wants to comment on a diagram		Same as normal path	
2. The user selects the project in which the diagram exists		Same as normal path	
3. The user selects the diagram on which they want to comment		Same as normal path	
4. The user then writes their comment under "Add comment" and presses the comment button		Same as normal path	
5. The system responds by adding the comment under the diagram		Same as normal path	
Exceptions/Alerts:			
Post-conditions:			
Step number		**Description**	
1.		The system displays a list of all comments under the diagram	
Use case Cross References:			
Includes:		Create Project, Create Diagram	
Extends:		None	

2.5 Use Case Diagram

2.5.1 User Management and Authentication

Figure 1 illustrates how users interact with the system to perform essential actions such as registration, login, profile management, and authentication. It highlights the roles involved and the key functionalities required to ensure secure access control.

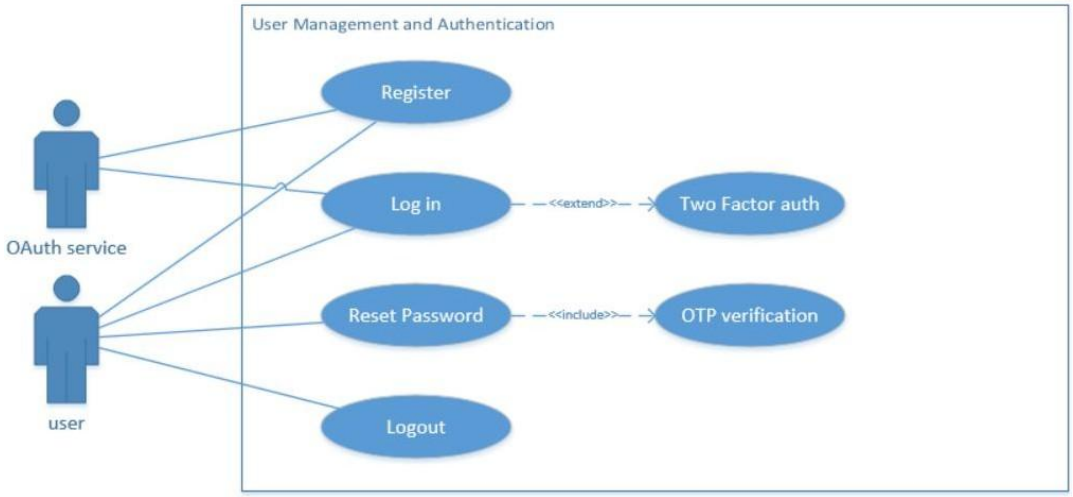


Figure 1 Use case diagram for User Management and Authentication

As shown in figure 2 how users customize their experience by managing preferences, selecting design settings, and modifying personalized options. It outlines the main interactions that enable a tailored and user-centric experience within the system.

2.5.2 Personalization

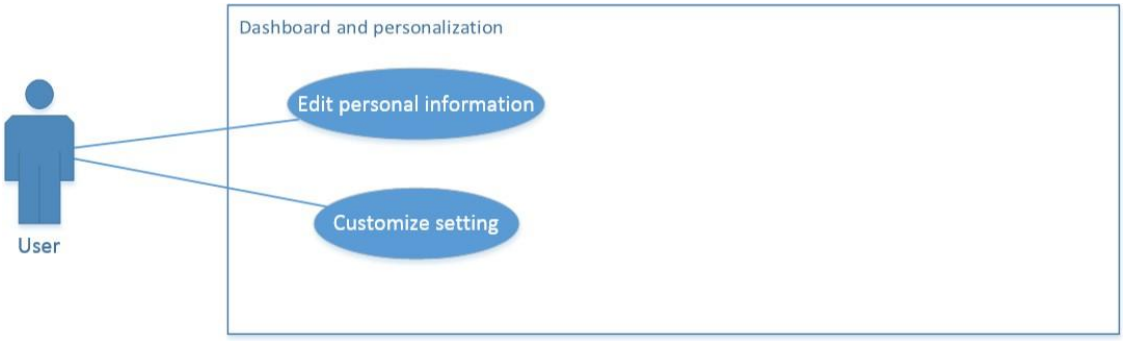


Figure 2 Use case diagram for Personalization

2.5.3 Team Management

The figure 3 illustrates how users manage teams by creating groups, assigning roles, and organizing members. It highlights the core interactions that support collaboration and coordinated work within the system.

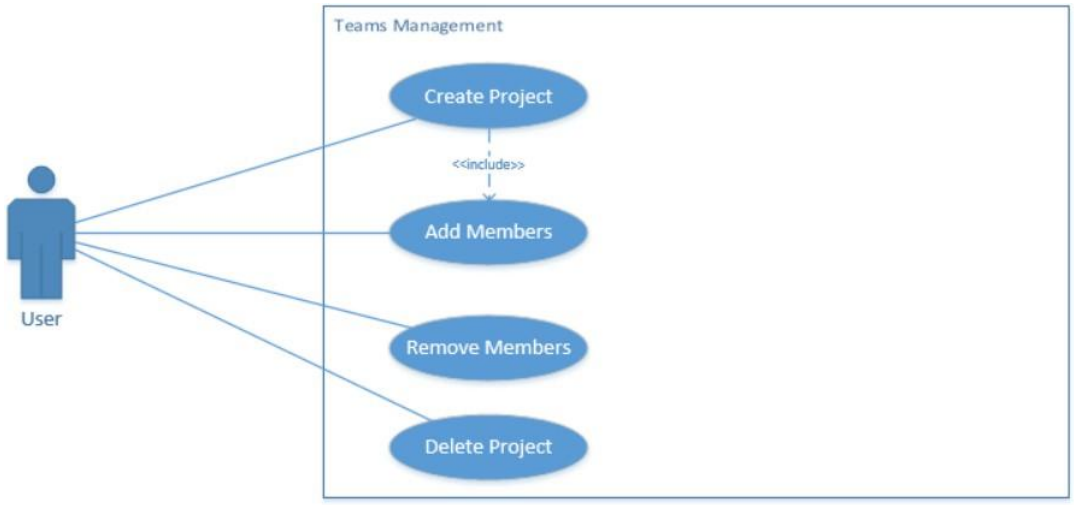


Figure 3 Use case diagram for Team Management

As shown in figure 4 how users participate in discussions by creating posts, commenting, and engaging in collaborative conversations. It outlines the main actions that facilitate communication and knowledge sharing within the platform.

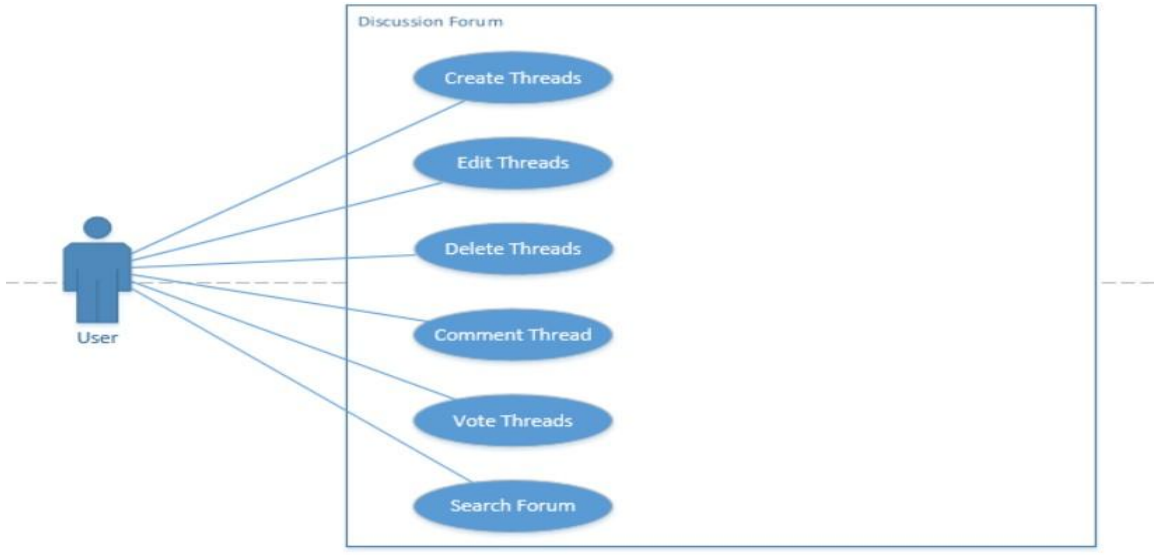


Figure 4 Use case diagram for Discussion Forum

2.5.5 Requirement Management

This figure 5 illustrates how users create, update, and track project requirements throughout the development process. It highlights the key interactions that ensure effective requirement handling and project documentation.

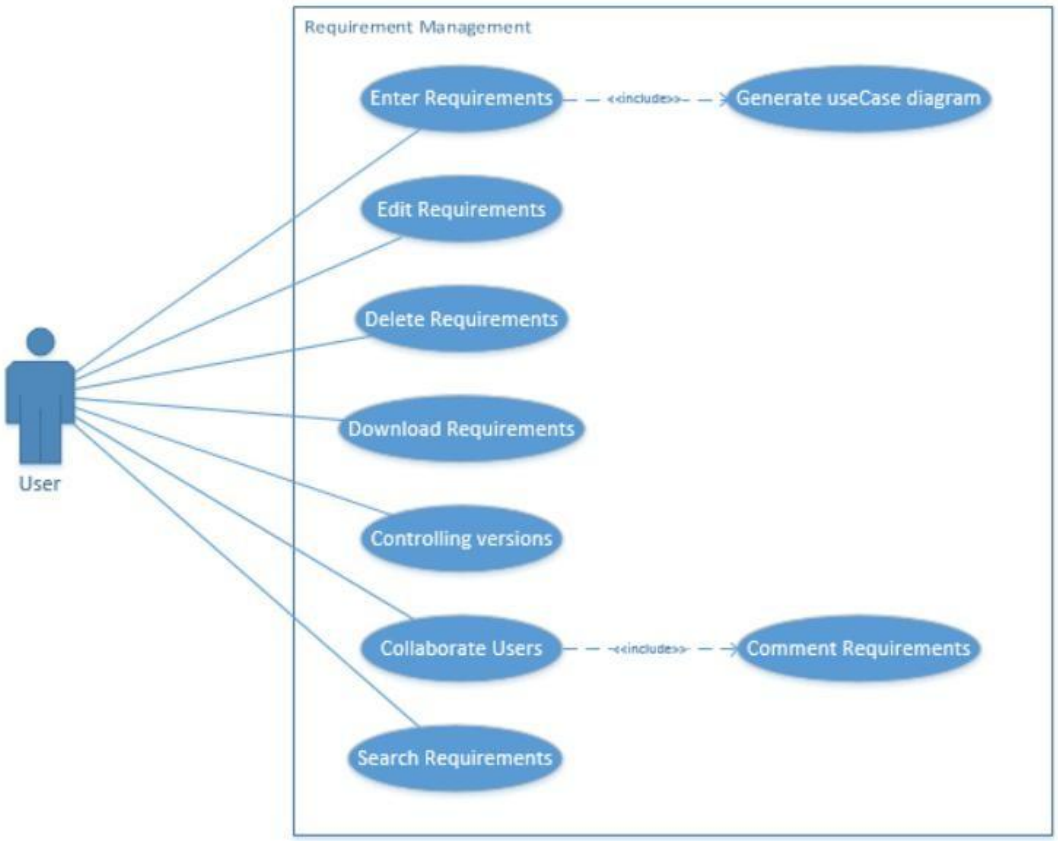


Figure 5 Use case diagram for Requirement Management

2.5.6 Diagramming

As shown in figure 6 how users create, edit, and manage visual diagrams within the system. It highlights the key interactions that support collaborative modelling and design activities.

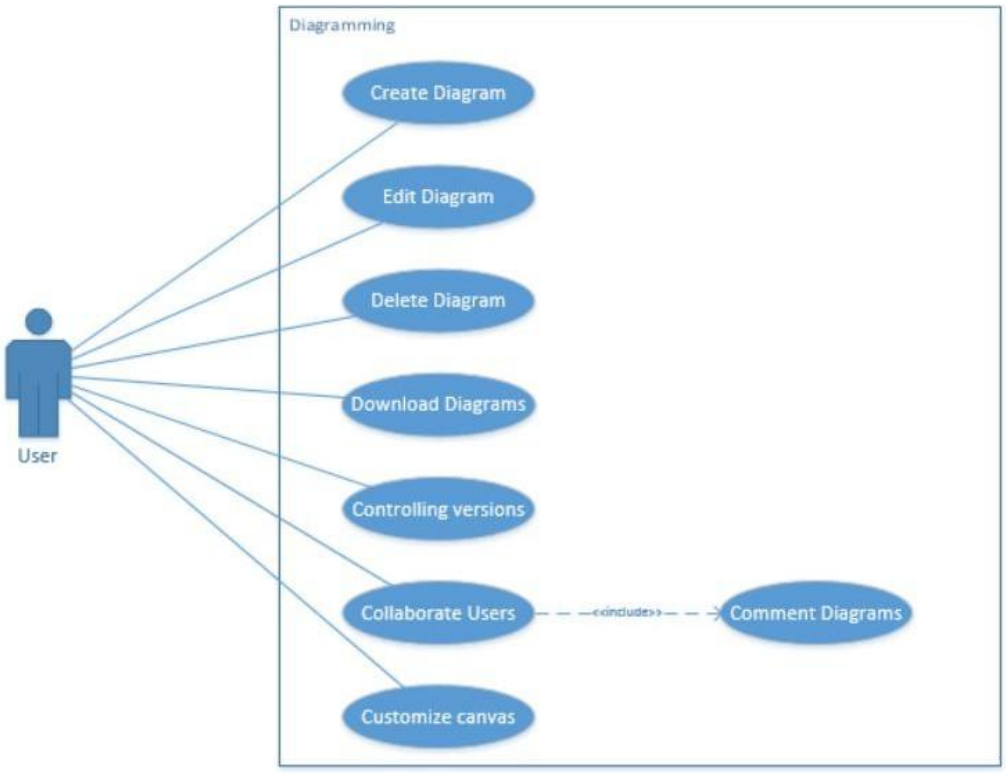


Figure 6 Use case diagram for Diagramming

CHAPTER 3

3 DESIGN AND METHODOLOGY

3.1 Software development Methodology

For this project, we intend to use the agile model for the software development process. The Agile model combines iterative and incremental process methodologies. In the Agile approach, development is carried out in small iterations, with a functional product delivered at the end of each iteration; this is the primary reason for selecting this model. It is very apt for our project because it supports flexibility, frequent feedback, and the ability to adapt to changing requirements as needed.

Using this model, we will systematically develop our project in stages. Each iteration will be focused on a specific set of features or modules. We will have a functional version of the product at the end of each iteration, which we can review and test. This will enable us to catch problems early on, minimize errors, and generally increase the quality of the project. We will be regularly reviewing our progress with our supervisor to ensure that it is on track with the project goals and incorporate any adjustments necessary. Each working version will add more features until we have built up to the final product. In this process, iteration is made sure so that we are always prepared to give results in every stage of development.

Using this model, we will systematically develop our project in stages. Each iteration will be focused on a specific set of features or modules. We will have a functional version of the product at the end of each iteration, which we can review and test. This will enable us to catch problems early on, minimize errors, and generally increase the quality of the project. We will be regularly reviewing our progress with our supervisor to ensure that it is on track with the project goals and incorporate any adjustments necessary. Each working version will add more features until we have built up to the final product. In this process, iteration is made sure so that we are always prepared to give results in every stage of development.

System Design

3.2 Work Breakdown Structure

3.2.1 User Management and Authentication

As shown in figure 7, it breaks down the user management and authentication module into smaller, manageable tasks. It outlines the hierarchy of activities required to implement secure user access and account handling.

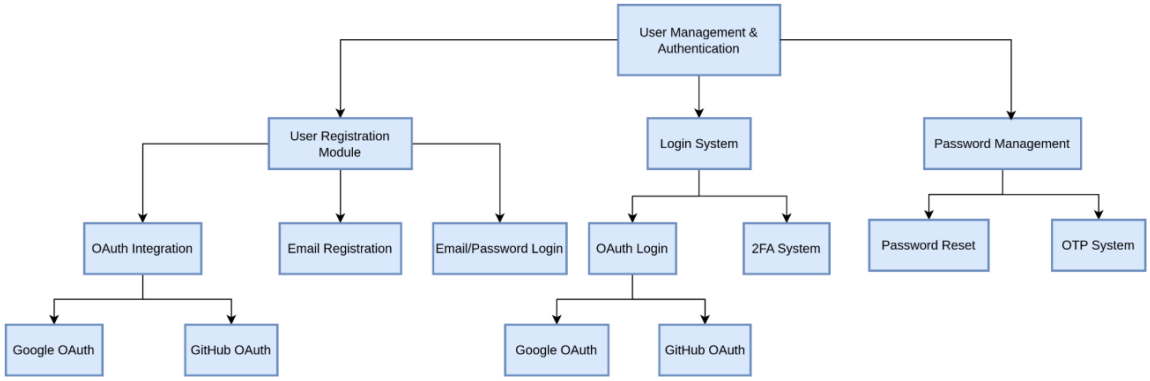


Figure 7 WBS for user management and authentication

3.2.2 User Dashboard

As shown in figure 8, it outlines the major tasks involved in developing the user dashboard and its related features. It presents a structured breakdown of components required to deliver an interactive and user-friendly interface.

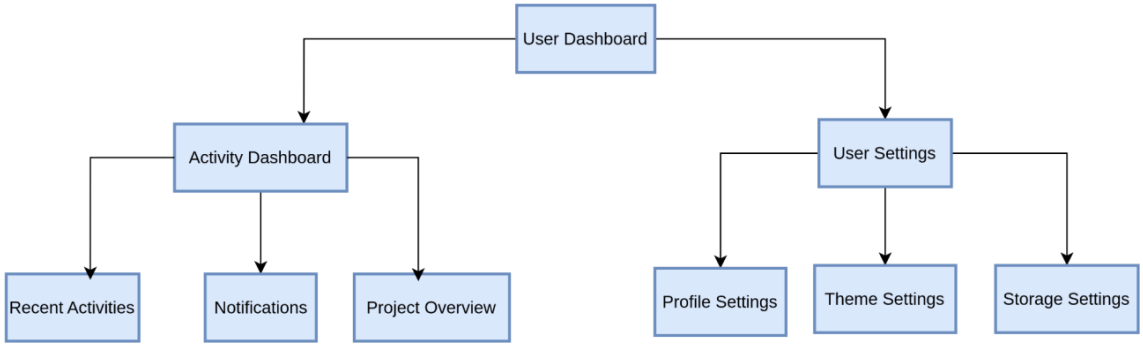


Figure 8 WBS for user dashboard

3.2.3 Coordination and Team Management

As shown in figure 9, it breaks down the tasks involved in managing teams and coordinating activities. It highlights the hierarchical structure of activities required for effective collaboration and team organization.

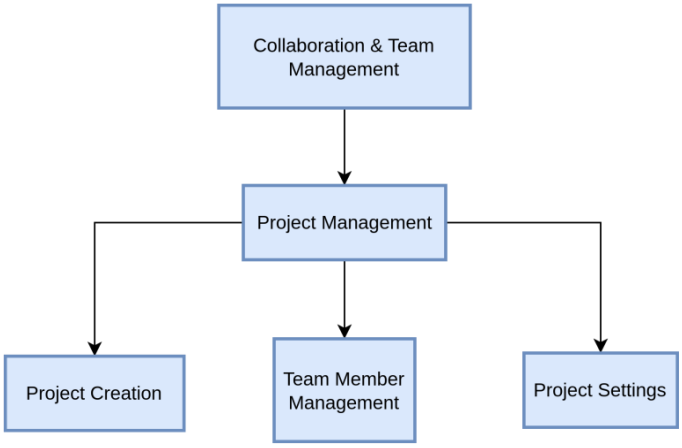


Figure 9 WBS for coordination and team management

3.2.4 Requirement Management

Figure 10 presents a hierarchical breakdown of tasks for managing project requirements. It outlines the key activities needed to capture, track, and maintain requirements effectively throughout the project lifecycle.

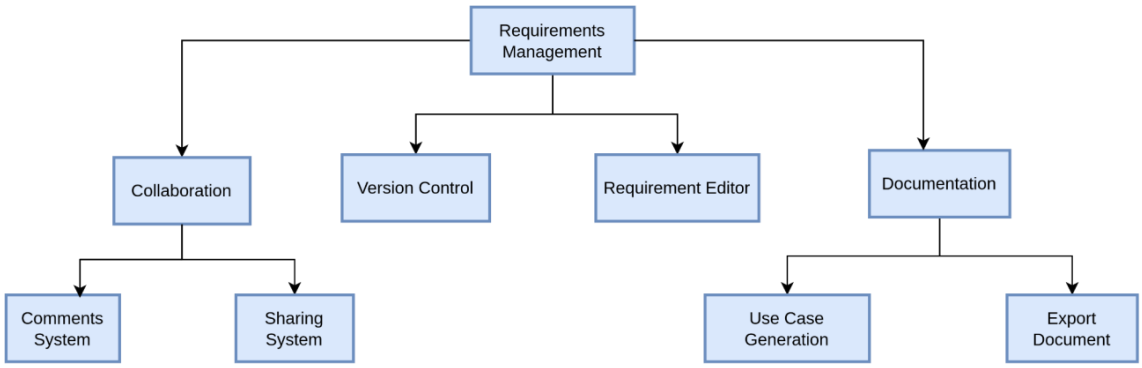


Figure 10 WBS for requirement management

3.2.5 Diagram Management

Figure 11 illustrates the breakdown of tasks involved in creating, editing, and organizing diagrams. It highlights the structured activities necessary for efficient diagram management within the system.

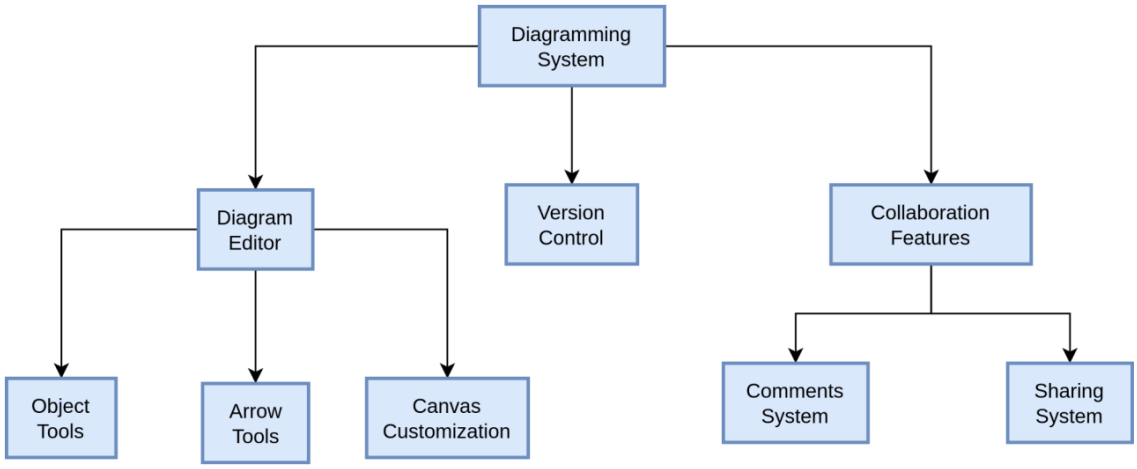


Figure 11 WBS for diagram management

3.2.6 Discussion Forum

As shown in figure 12, it breaks down the tasks required to implement and manage the discussion forum. It outlines the key activities that enable users to create posts, comment, and engage in collaborative discussions.

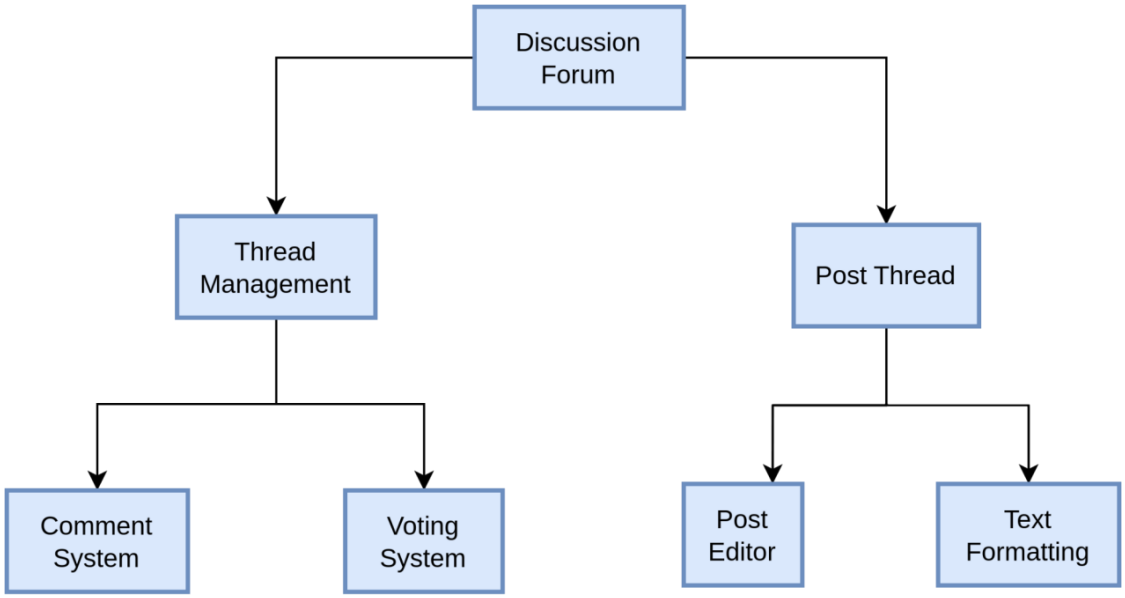


Figure 12 WBS for discussion forum

3.3 Activity Diagram

3.3.1 User authentication

As shown in figure 13, it depicts the step-by-step flow of actions a user follows to register in the system. It highlights decision points, processes, and interactions to ensure successful account creation.

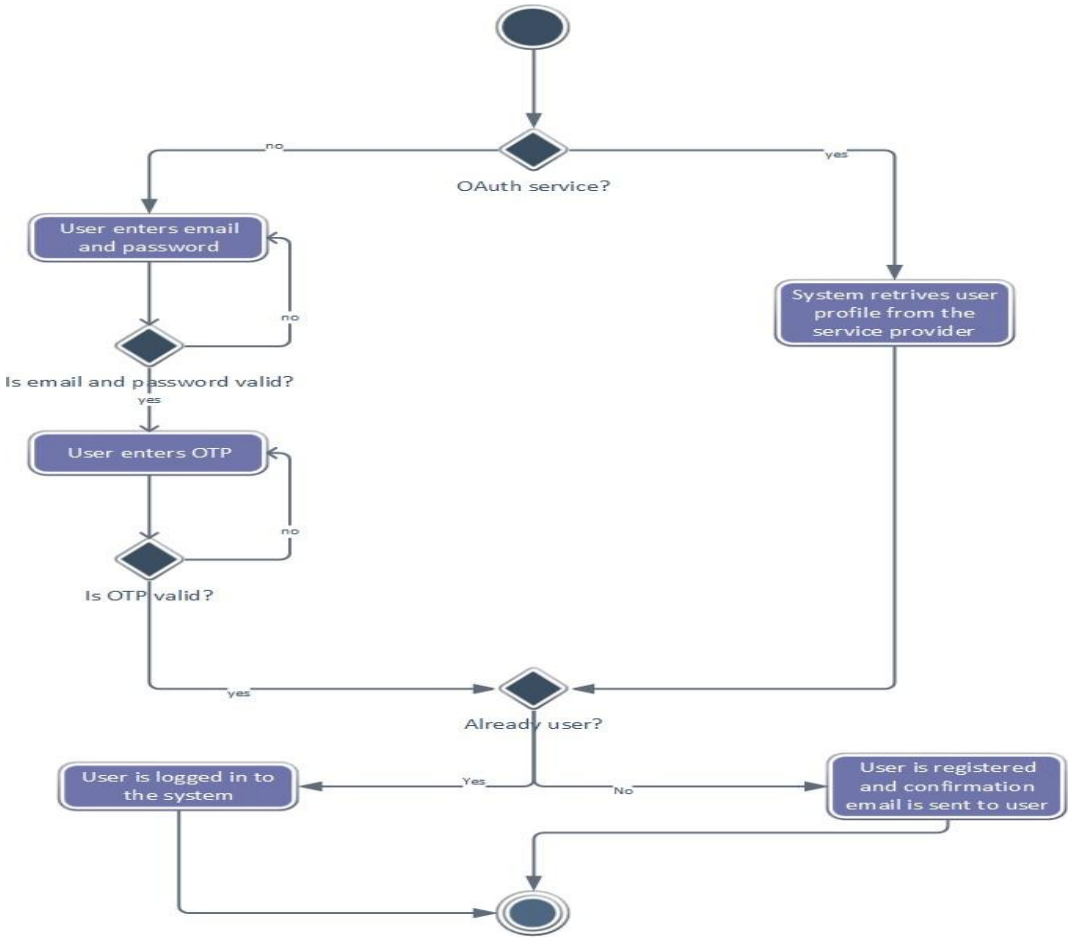


Figure 133 Activity diagram for forgot password

3.3.2 Forgot Password

As shown in figure 14, it illustrates the process a user follows to reset a forgotten password. It shows the sequence of steps, including verification and password update, to securely regain account access.

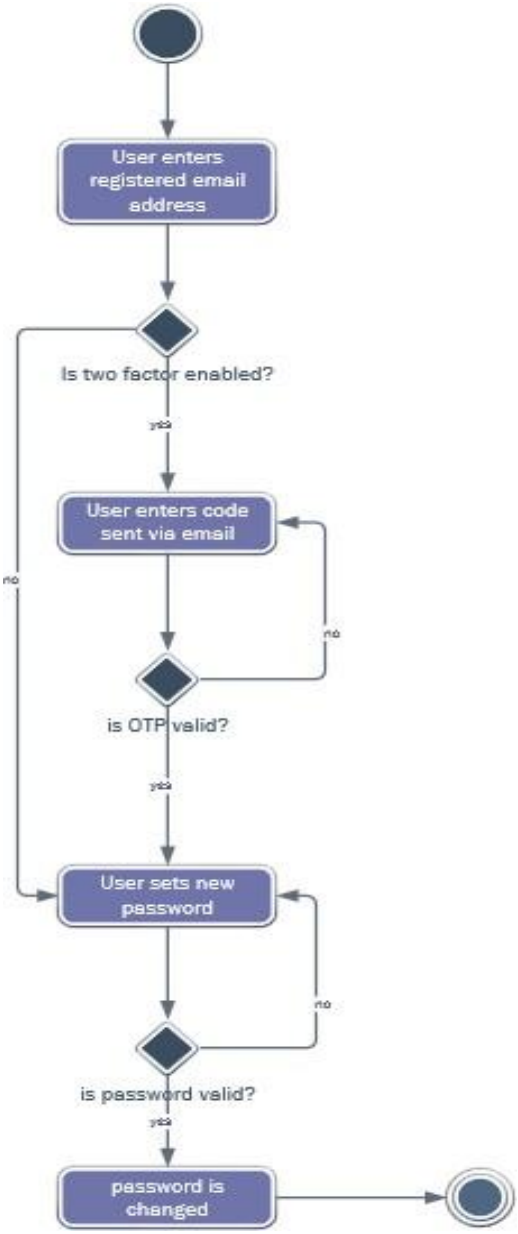
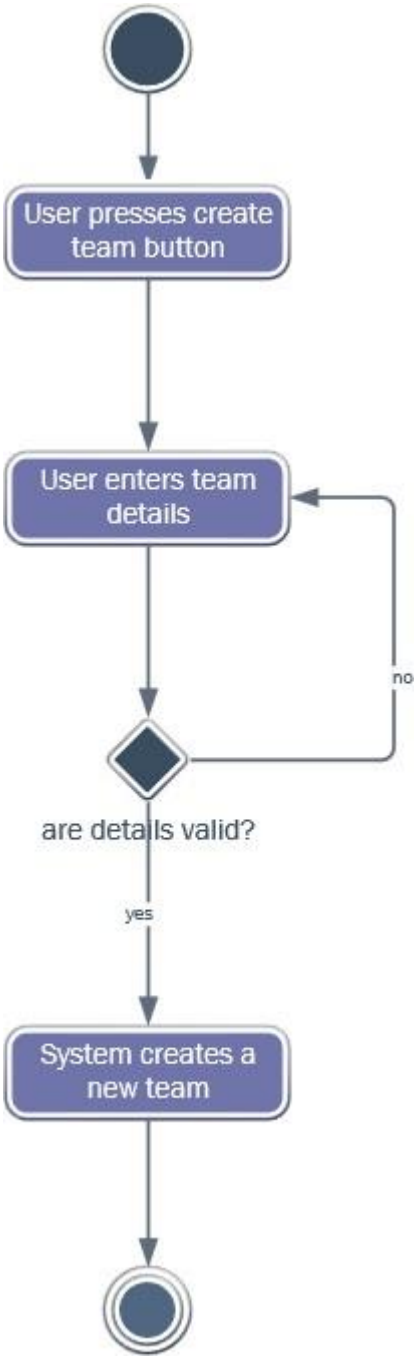


Figure 14 Activity diagram for forgot password

3.3.3 Team Management

As shown in figure 15, it depicts the sequence of actions involved in creating a new team within the system. It highlights the steps for defining team details, assigning members, and finalizing the team setup.



3.3.4 Updating Team Details

As shown in figure 16, it illustrates the process of modifying existing team information. It shows the sequence of actions for editing team details, updating member roles, and saving the changes in the system.

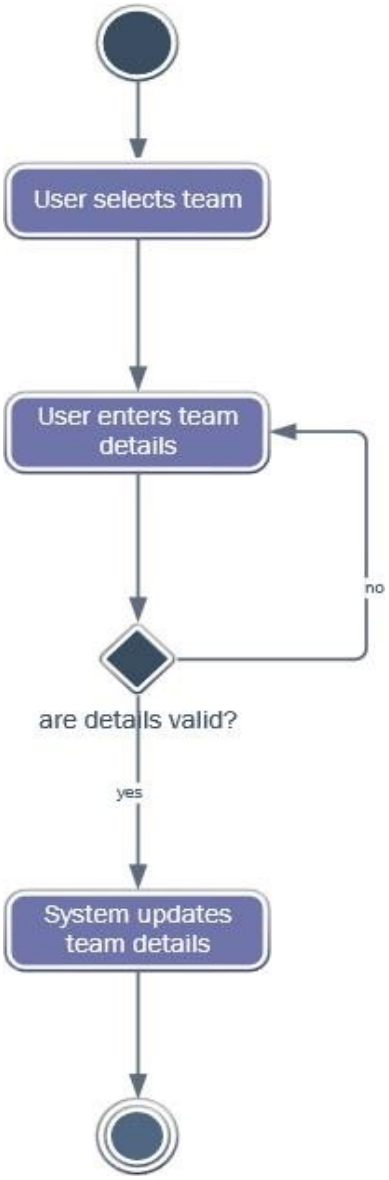


Figure 16 Activity diagram for updating team details

3.3.5 Project Management

As shown in figure 17, it shows the step-by-step process of adding new project requirements. It highlights the actions involved in defining, validating, and recording requirements within the system.

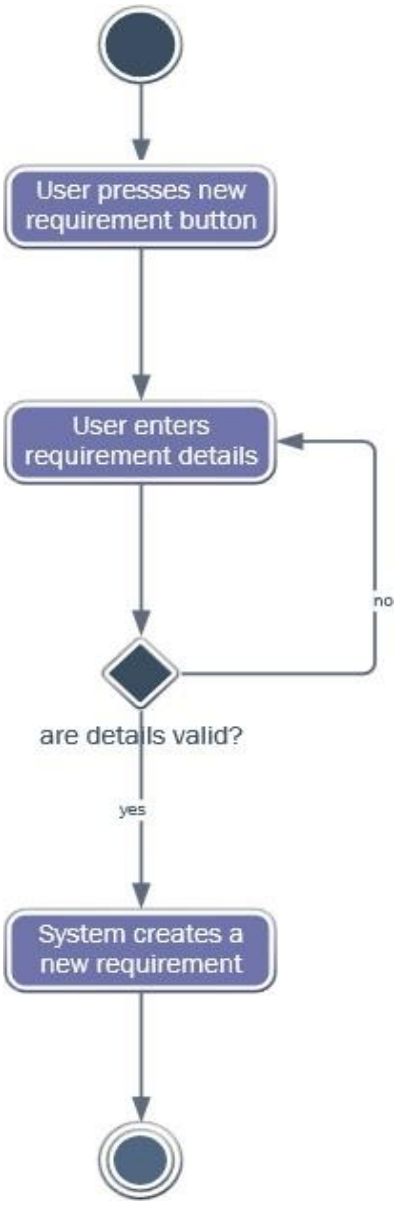


Figure 17 Activity diagram for creation of new requirements

3.3 6 File management

The diagram 18 illustrates the process a user follows to download a file from the system. It highlights the steps from selecting the file to completing the download successfully.

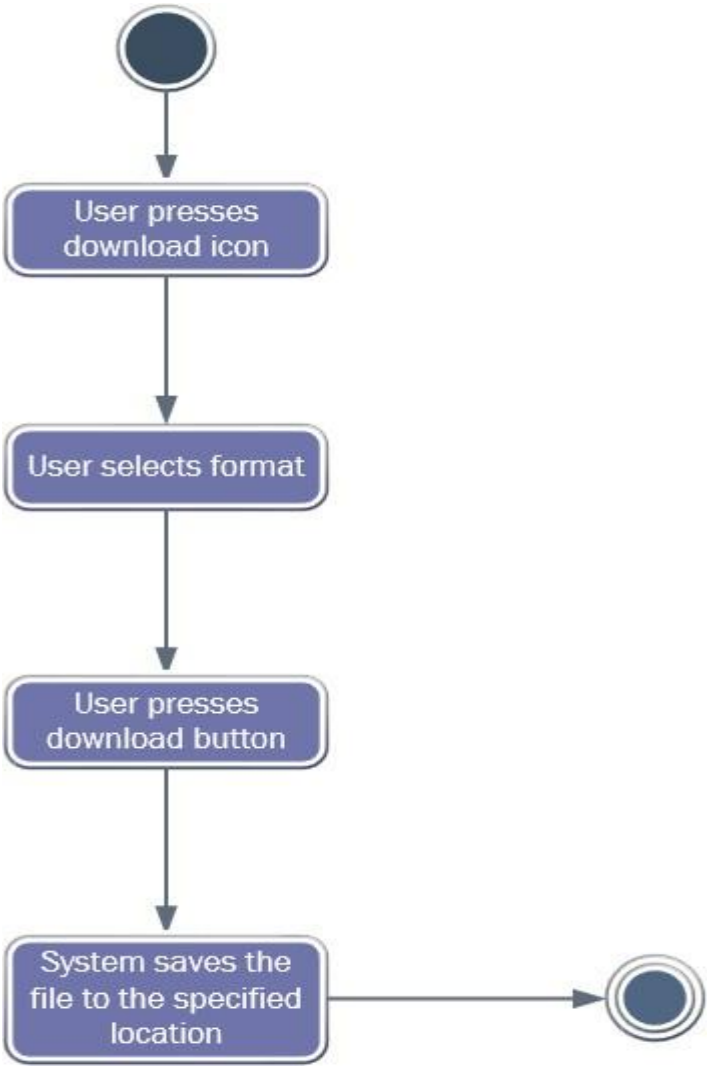


Figure 18 Activity diagram for file download

3.3 7 Discussion Forum

As shown in figure 19, it depicts the process for modifying or removing discussion threads in the forum. It highlights the sequence of actions, including verification, editing, and deletion, to manage content effectively.

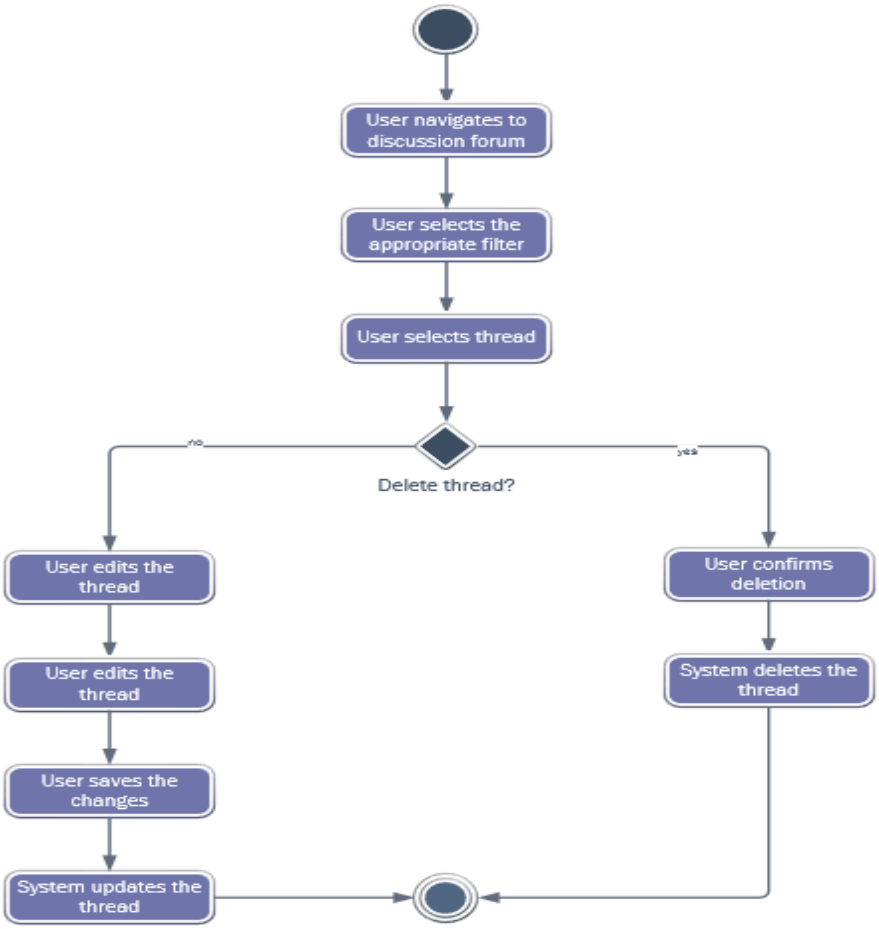


Figure 19 Activity diagram for thread deletion or editing

3.3.8 Diagramming

Diagram 20 illustrates the sequence of actions a user follows to download diagrams from the system. It highlights steps from selecting the diagram to completing the download successfully.

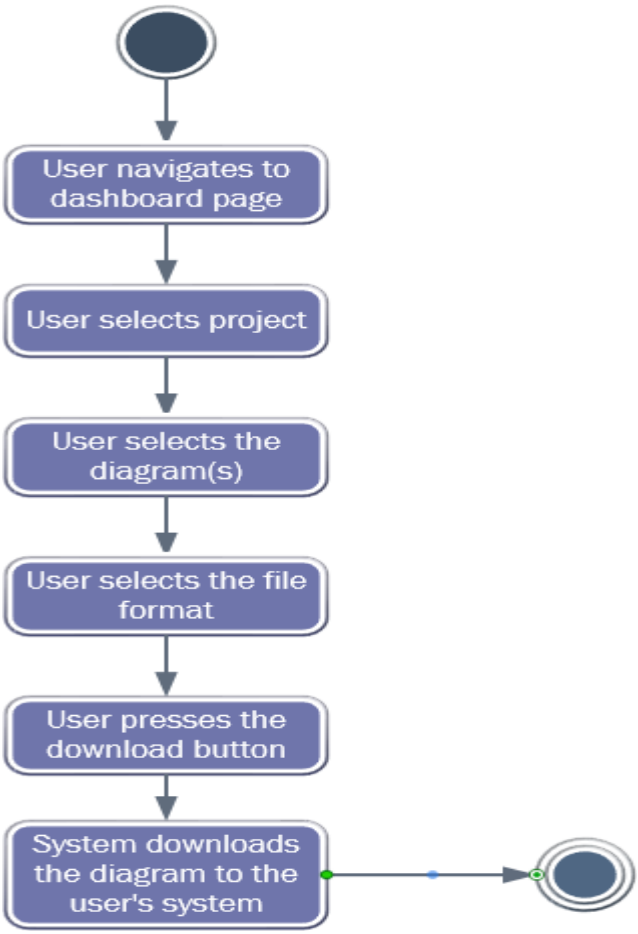


Figure 20 Activity diagram for thread deletion or editing

3.3.9 Editing or Deleting Diagrams

Figure 21 depicts the process for modifying or removing diagrams within the system. It shows the sequence of actions, including selecting the diagram, making changes or confirming deletion, and saving the updates.

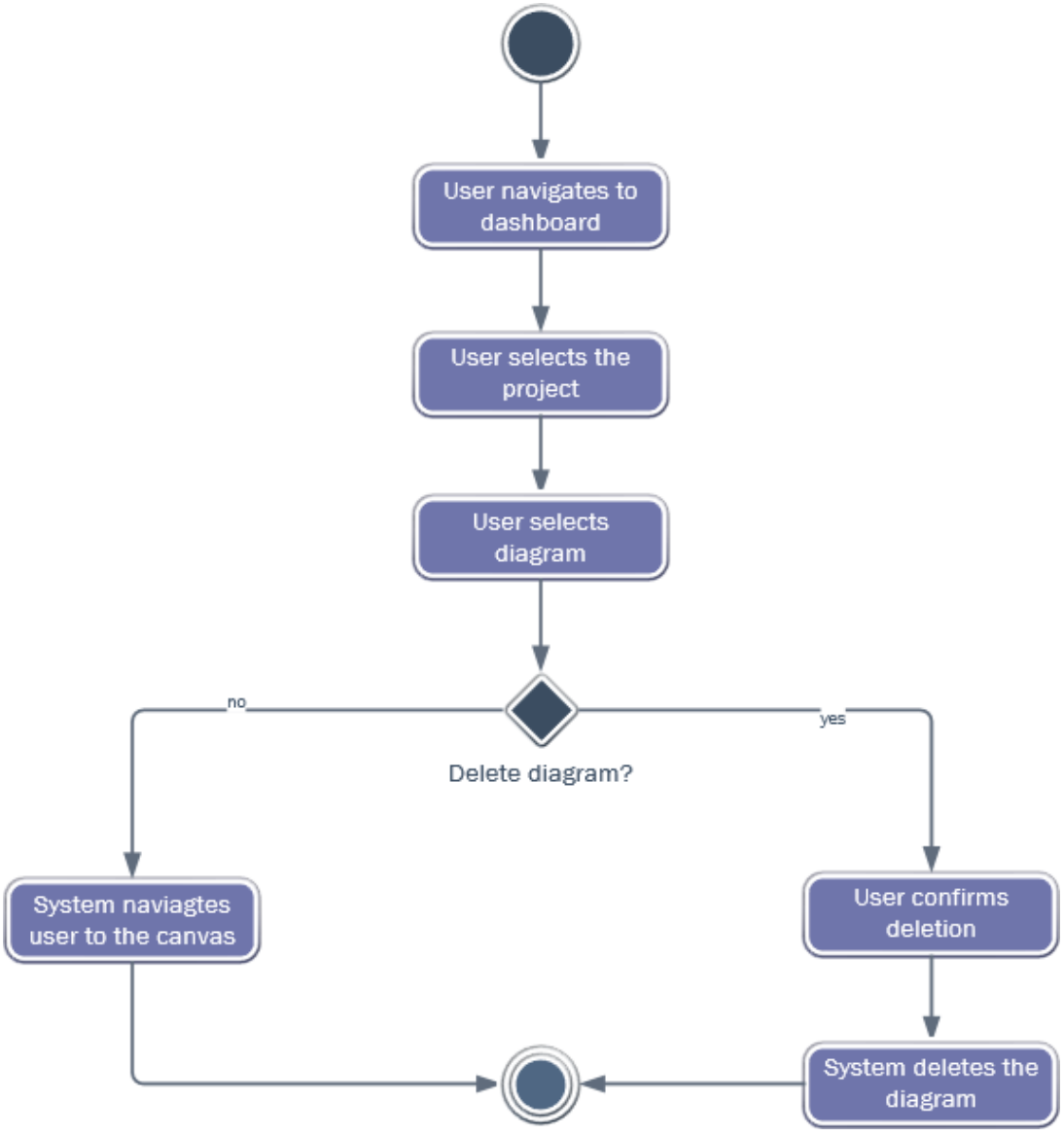


Figure 22 Activity diagram for editing or deleting diagrams

3.4 Sequence Diagram

3.4.1 Registration

Diagram 22 illustrates the interaction between the user and the system during the registration process. It shows the sequence of messages exchanged to successfully create a new user account.

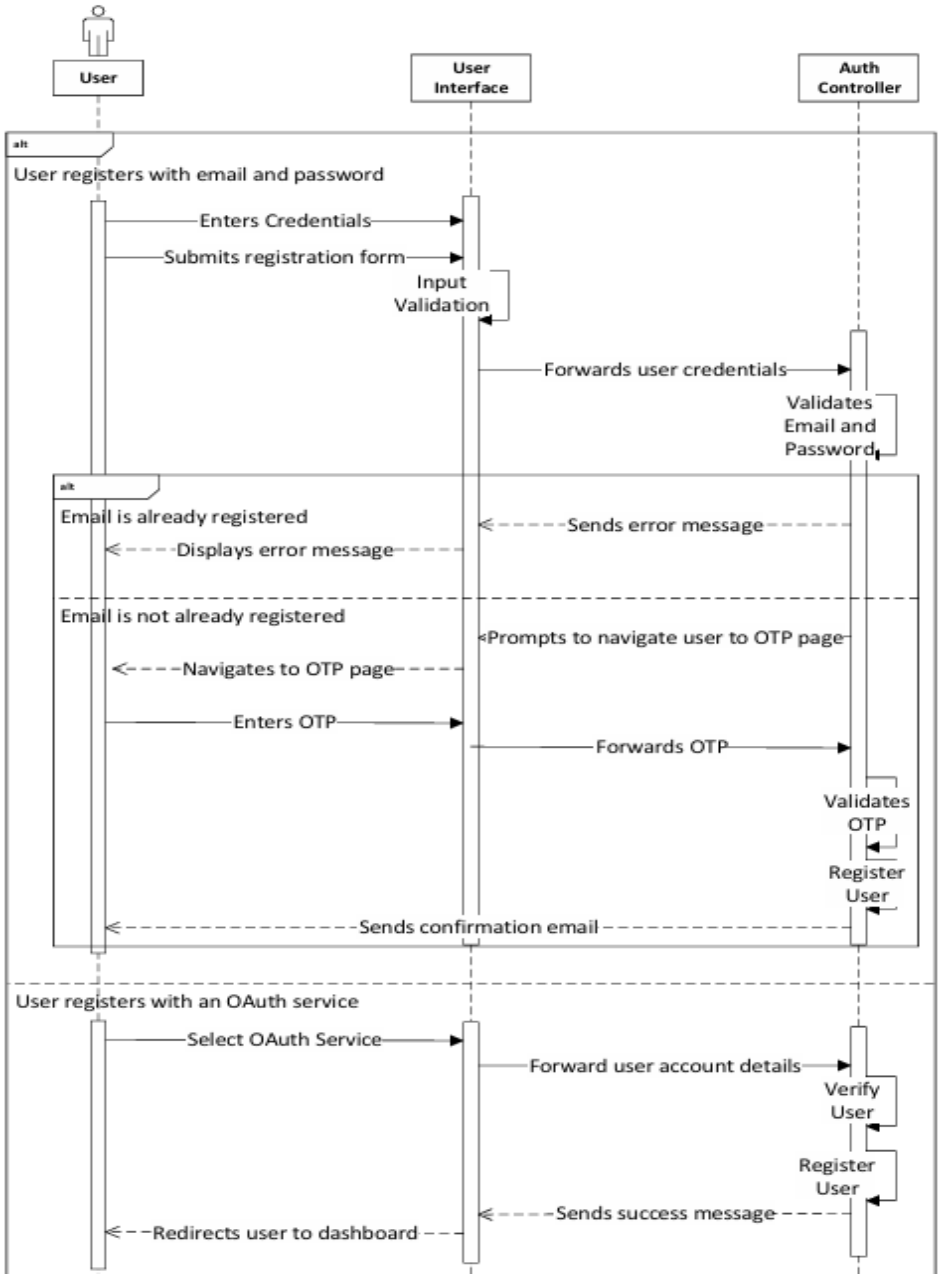


Figure 23 Sequence diagram for registering user

3.4.2 Login

Diagram 23 depicts the interactions between the user and the system during the login process. It shows the sequence of messages for verifying credentials and granting access to the user account.

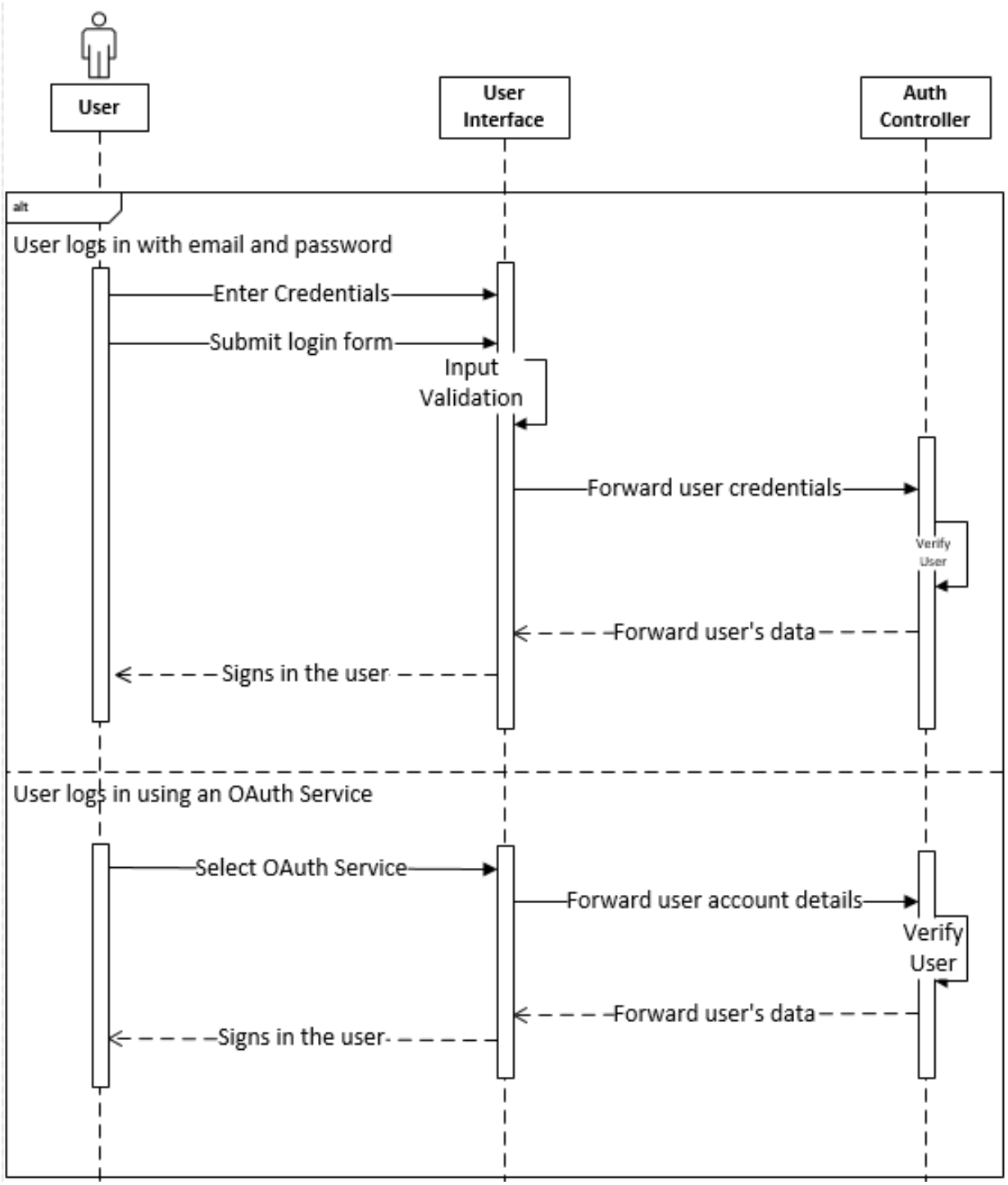


Figure 24 Sequence diagram for login user

3.4.3 Editing Personal Information

Diagram 24 illustrates the interaction between the user and the system when updating personal details. It shows the sequence of messages for submitting changes, validating input, and saving the updated information.

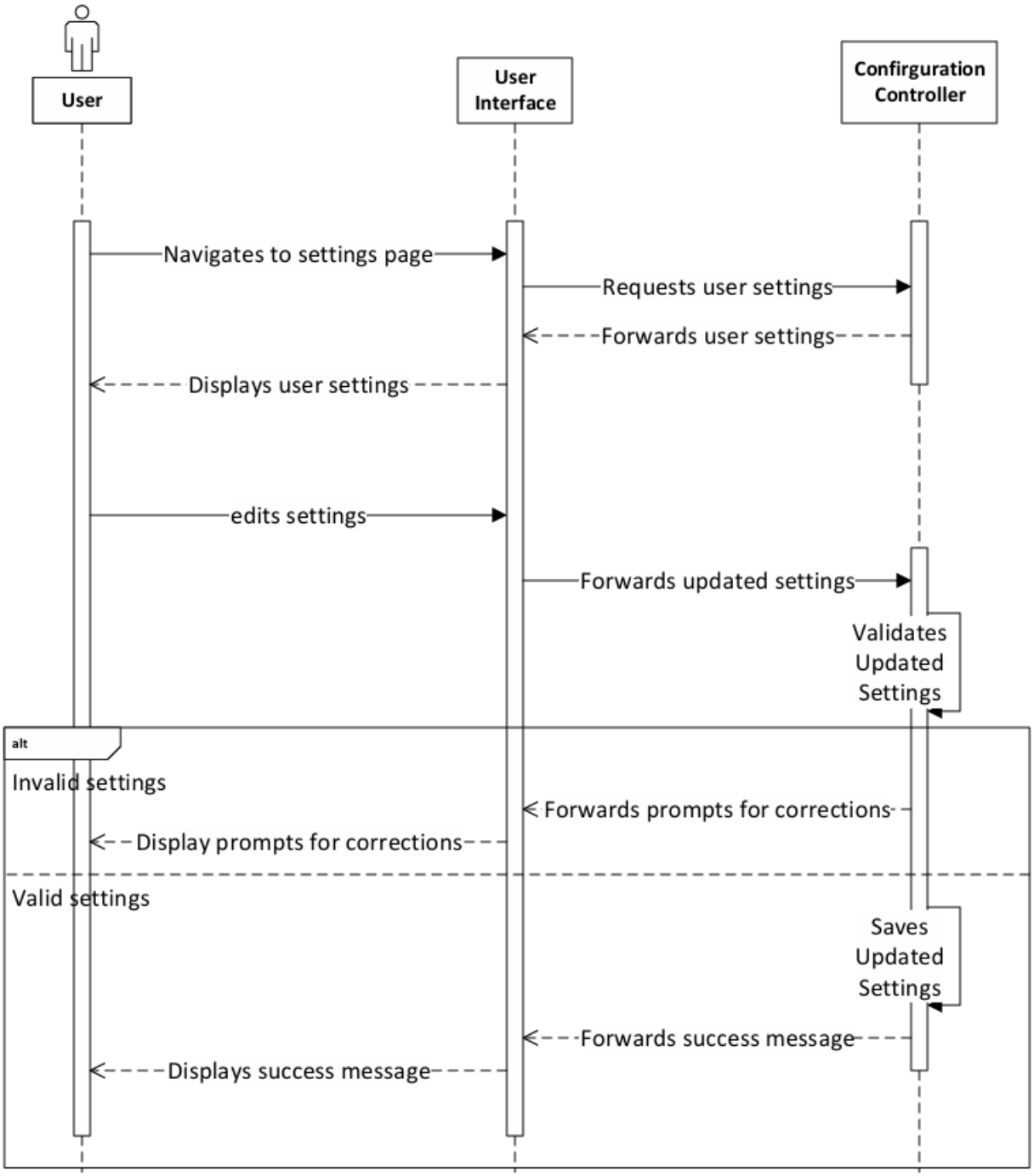


Figure 25 Sequence diagram for editing personal information

3.4 4 Creating Teams

Diagram 25 depicts the interactions between the user and the system during team creation. It shows the sequence of messages for entering team details, assigning members, and confirming the team setup.

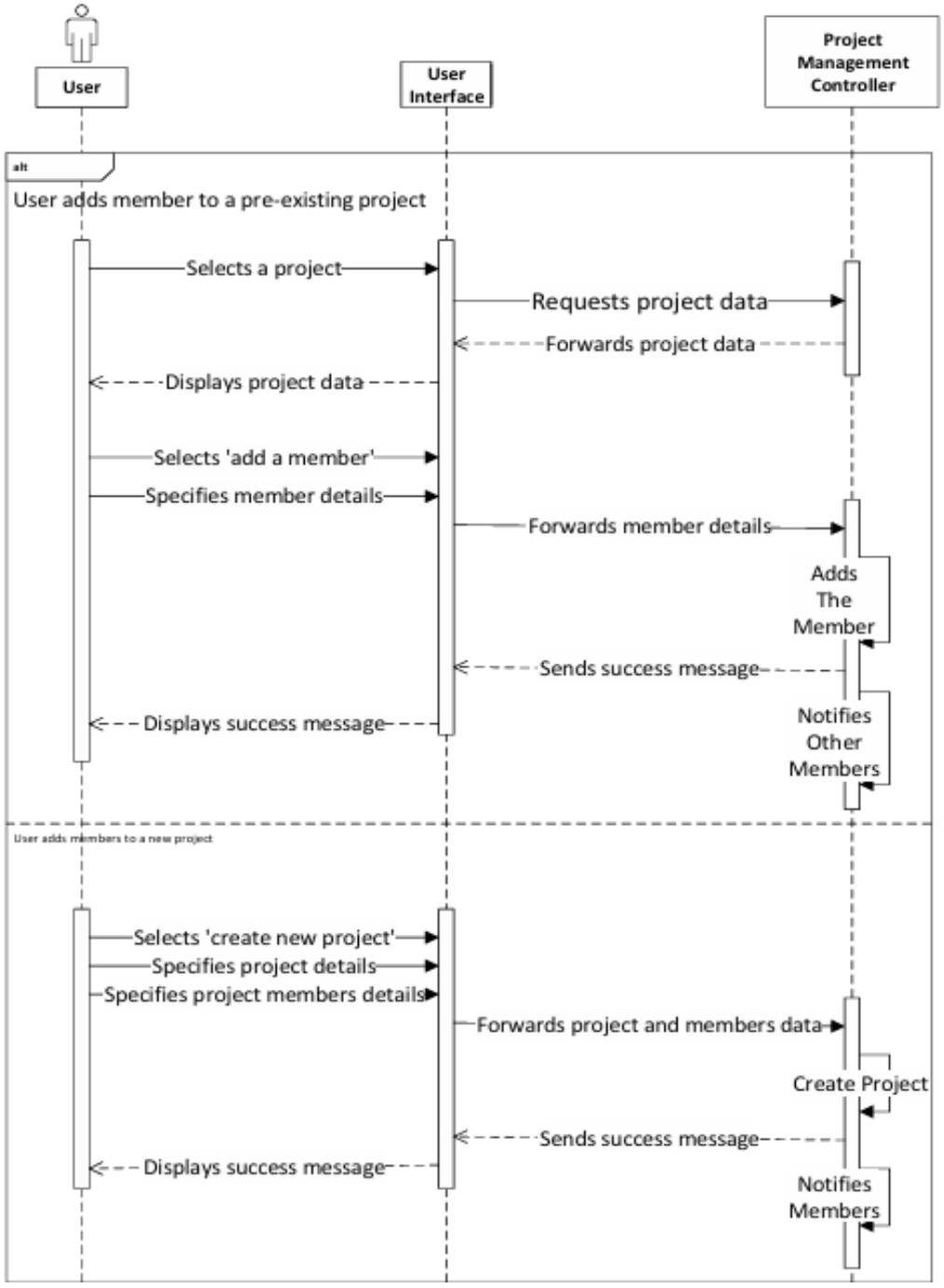


Figure 26 Sequence diagram for creating teams

3.4 5 Editing Requirements

Diagram 26 illustrates the interaction between the user and the system when modifying existing project requirements. It shows the sequence of messages for selecting a requirement, making changes, and saving the updates.

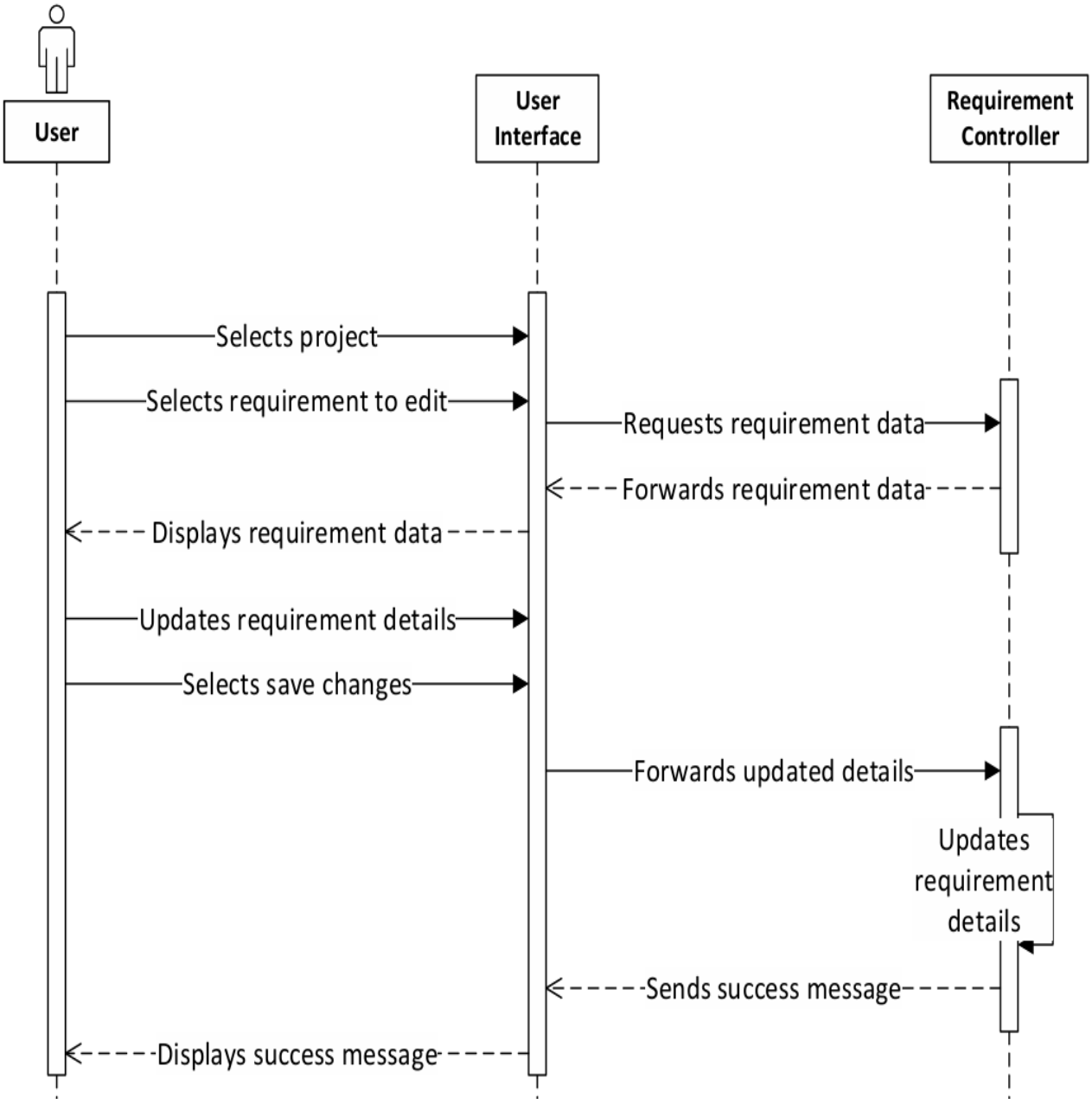


Figure 27 Sequence diagram for editing requirements

3.4.6 Download Documents

Diagram 27 depicts the interactions between the user and the system during the document download process. It shows the sequence of messages from selecting a document to completing the download successfully.

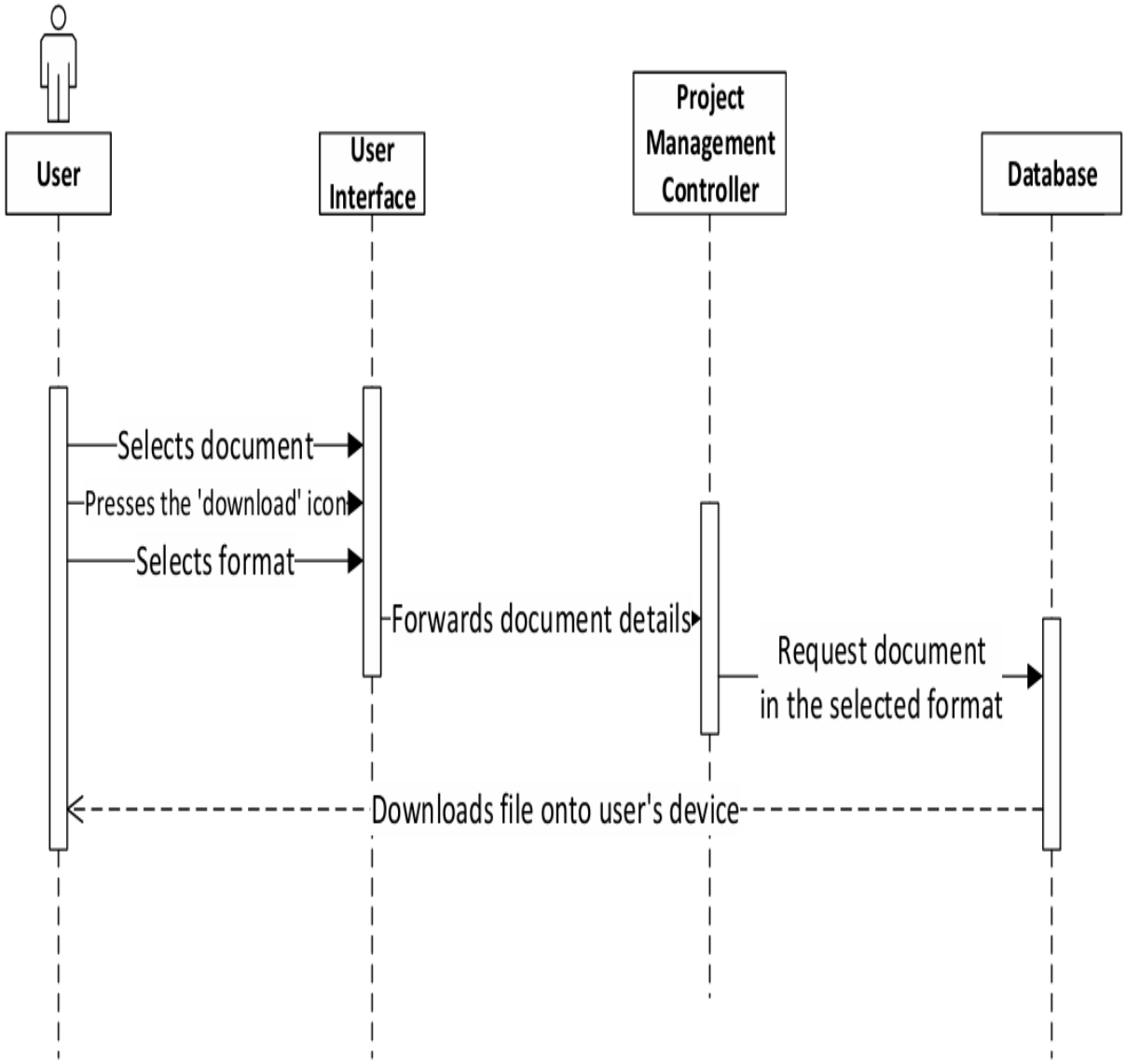


Figure 28 Sequence diagram for download documents

3.4.7 Create Thread

Diagram 28 illustrates the interactions between the user and the system when creating a new discussion thread. It shows the sequence of messages for entering thread details, submitting the thread, and confirming its creation.

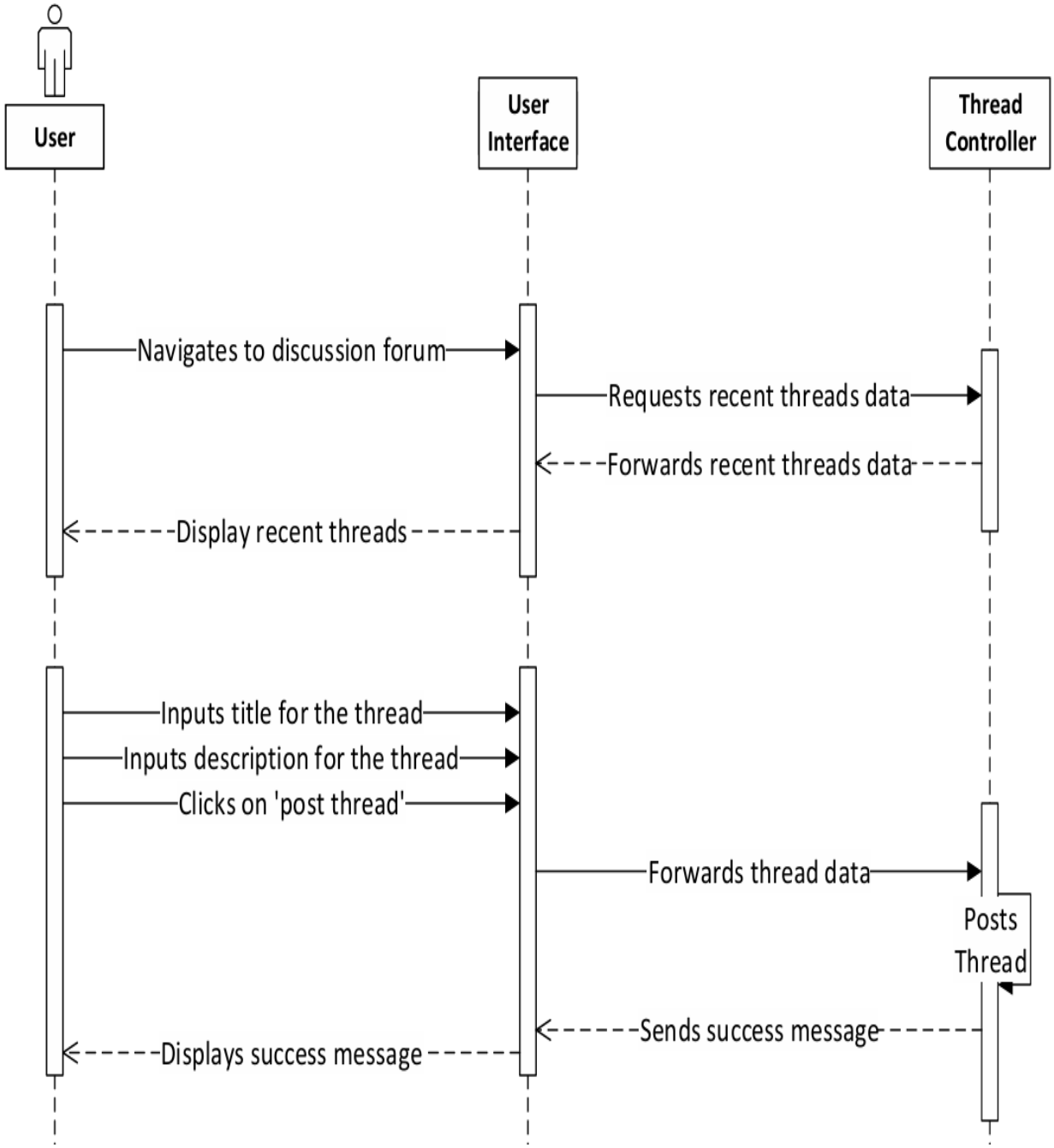


Figure 29 Sequence diagram for create thread

3.4.8 Search Forum

Diagram 29 depicts the interactions between the user and the system during a forum search. It shows the sequence of messages for entering search criteria, processing the query, and displaying relevant results.

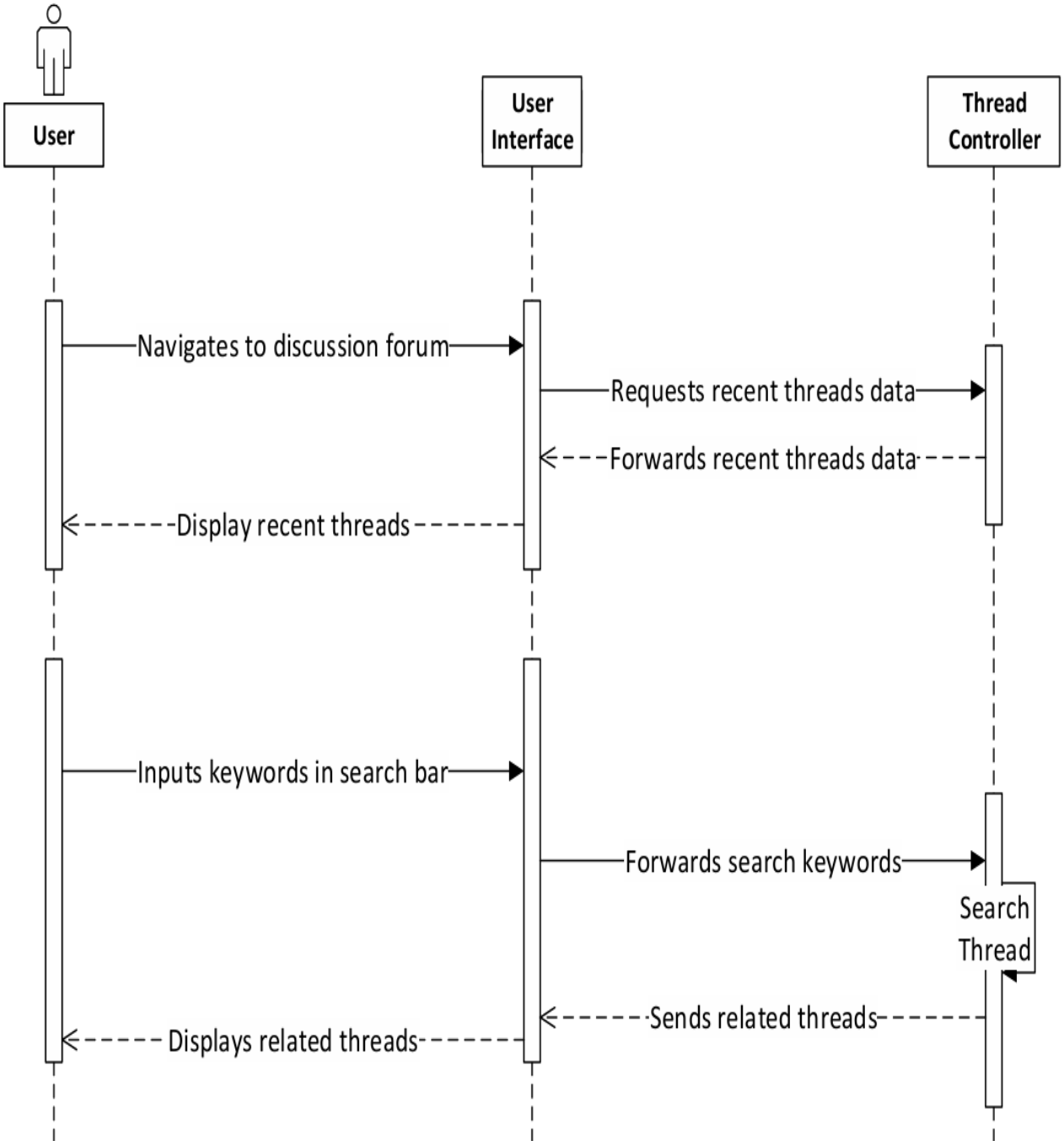


Figure 30 Sequence diagram for search forum

3.4.9 Edit Diagram

Diagram 30 illustrates the interactions between the user and the system when editing an existing diagram. It shows the sequence of messages for selecting the diagram, making modifications, and saving the updated version.

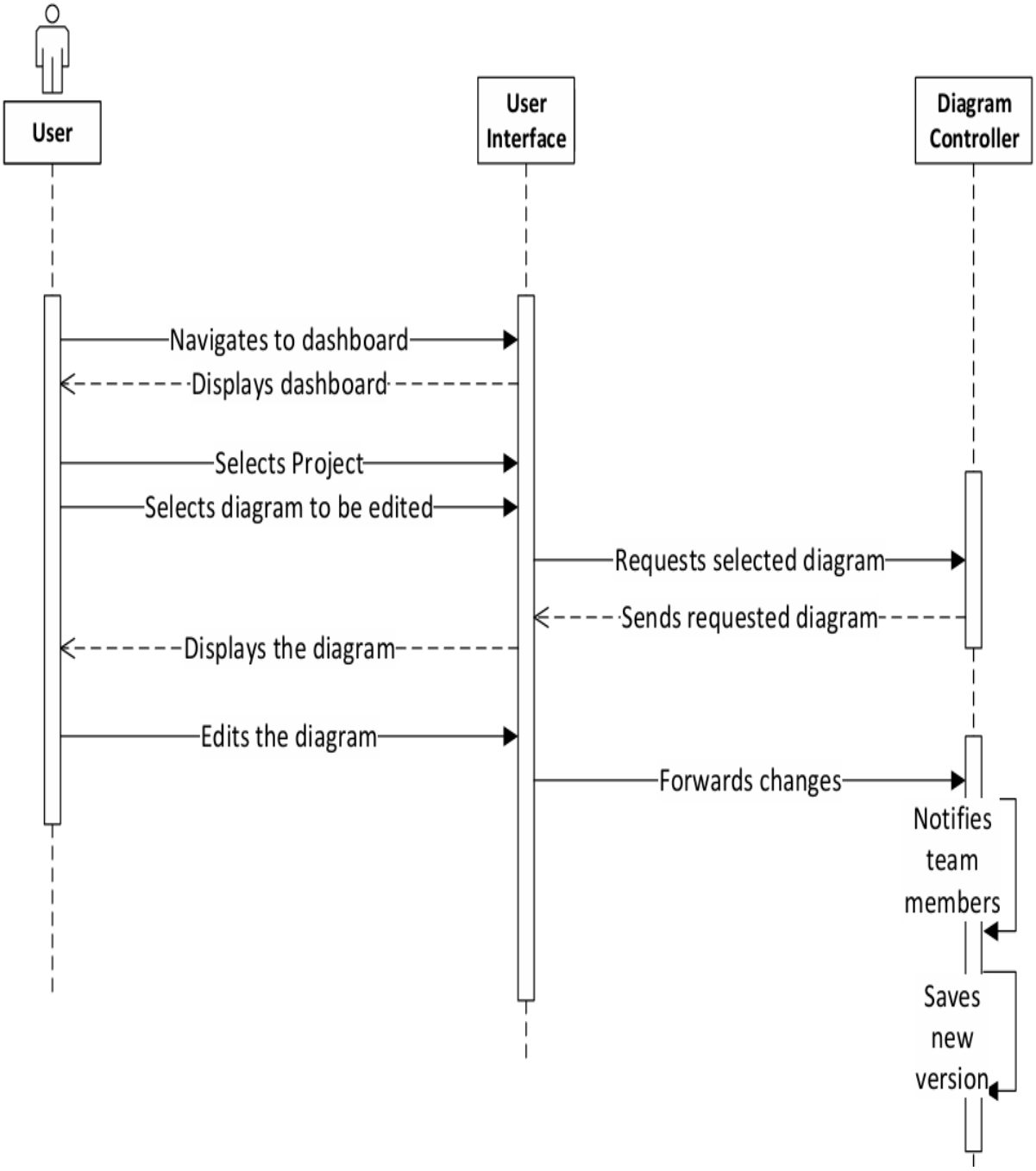


Figure 31 Sequence diagram for edit diagram

3.5 Software Architecture

The software architecture that we have selected for this project is MVC architecture. MVC stands for model view controller. It is a frequently used development framework to create extensible projects. MVC is a software design pattern for designing web applications. MVC architecture composed of three components that are:

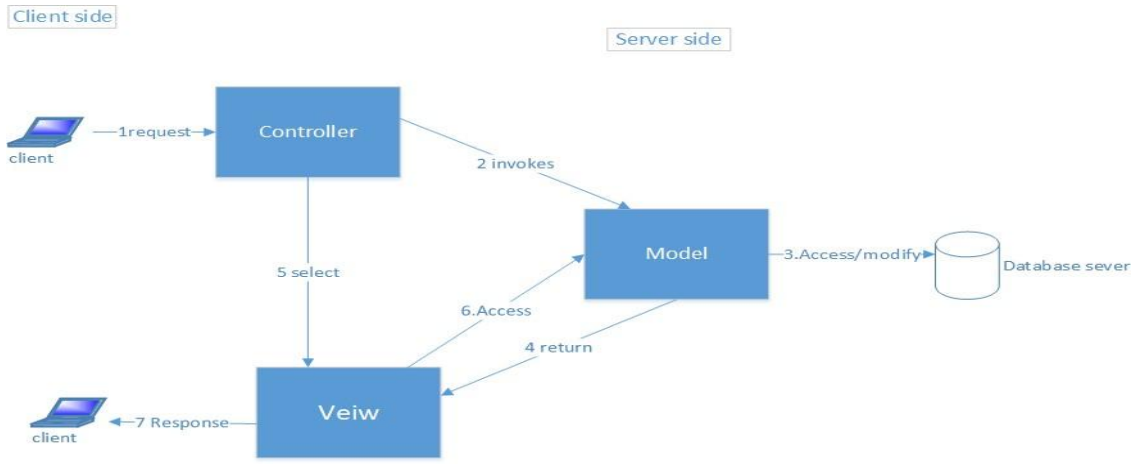
- Model
- View
- Controller

Model: It corresponds to data-oriented logic that users work on. The data represented by the model can be business related data or data that is being transferred between view and controller. Model layer is also known as the business layer.

View: This component deals with the UI (user interface) of the application.

Controller: This component acts as an interface between the model component and view component. It processes all business logic and incoming requests, manipulates data using model component and interacts with the views to render output.

Diagram 31 illustrates the MVC architecture, separating the system into Model, View, and Controller components. It shows how user actions flow through the Controller, interact with the Model, and update the View.



Software Architecture Diagram 31

3.6 Class Diagram

Diagram 32 shows the system's main classes along with their attributes and methods. It highlights the relationships between classes to represent the system's overall structure and behaviour.

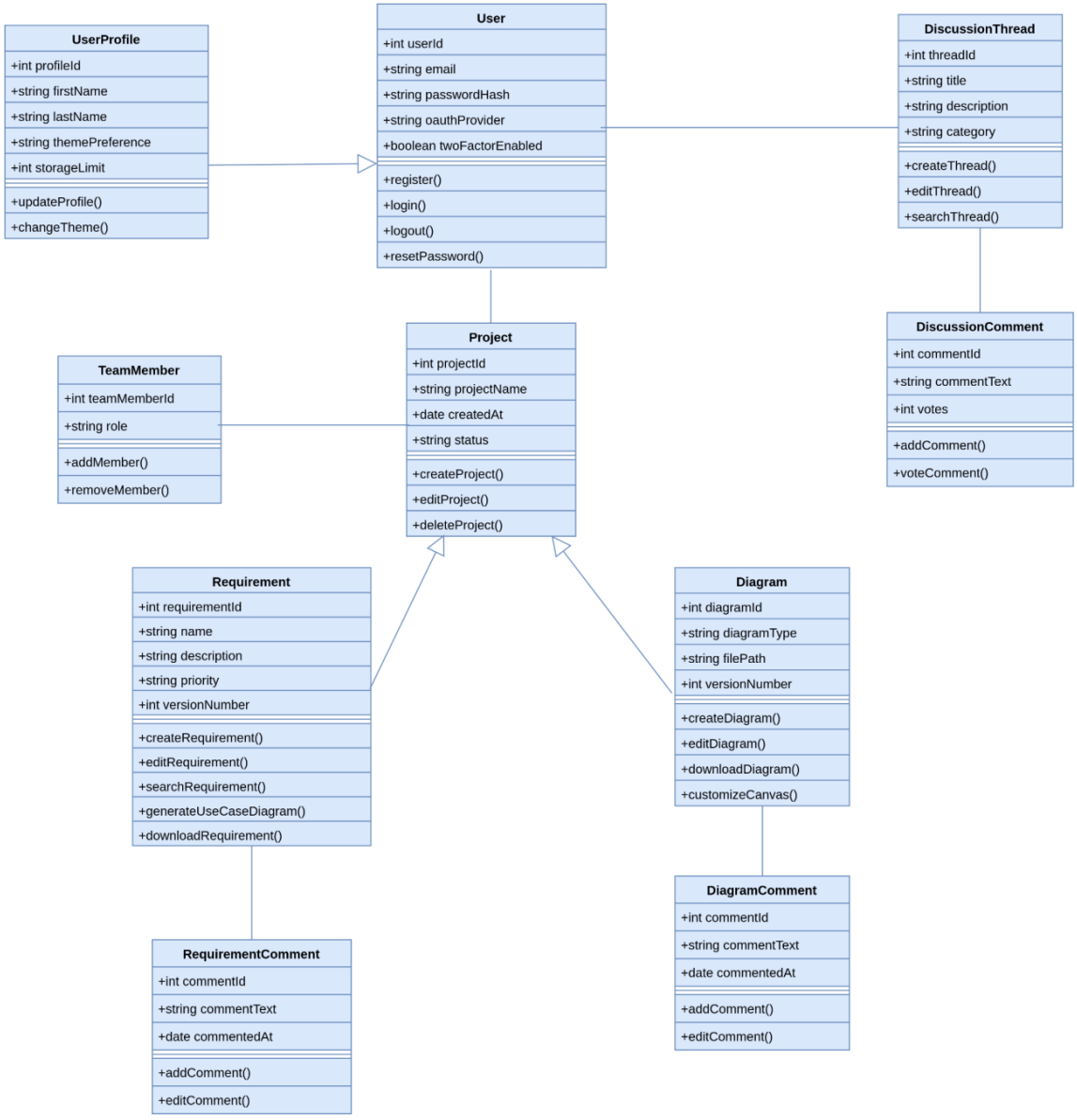


Figure 32 Class Diagram

3.7 Database Diagram

Diagram represents the system's database structure by showing entities, their attributes, and the relationships between them. It helps visualize how data is organized and connected within the system.

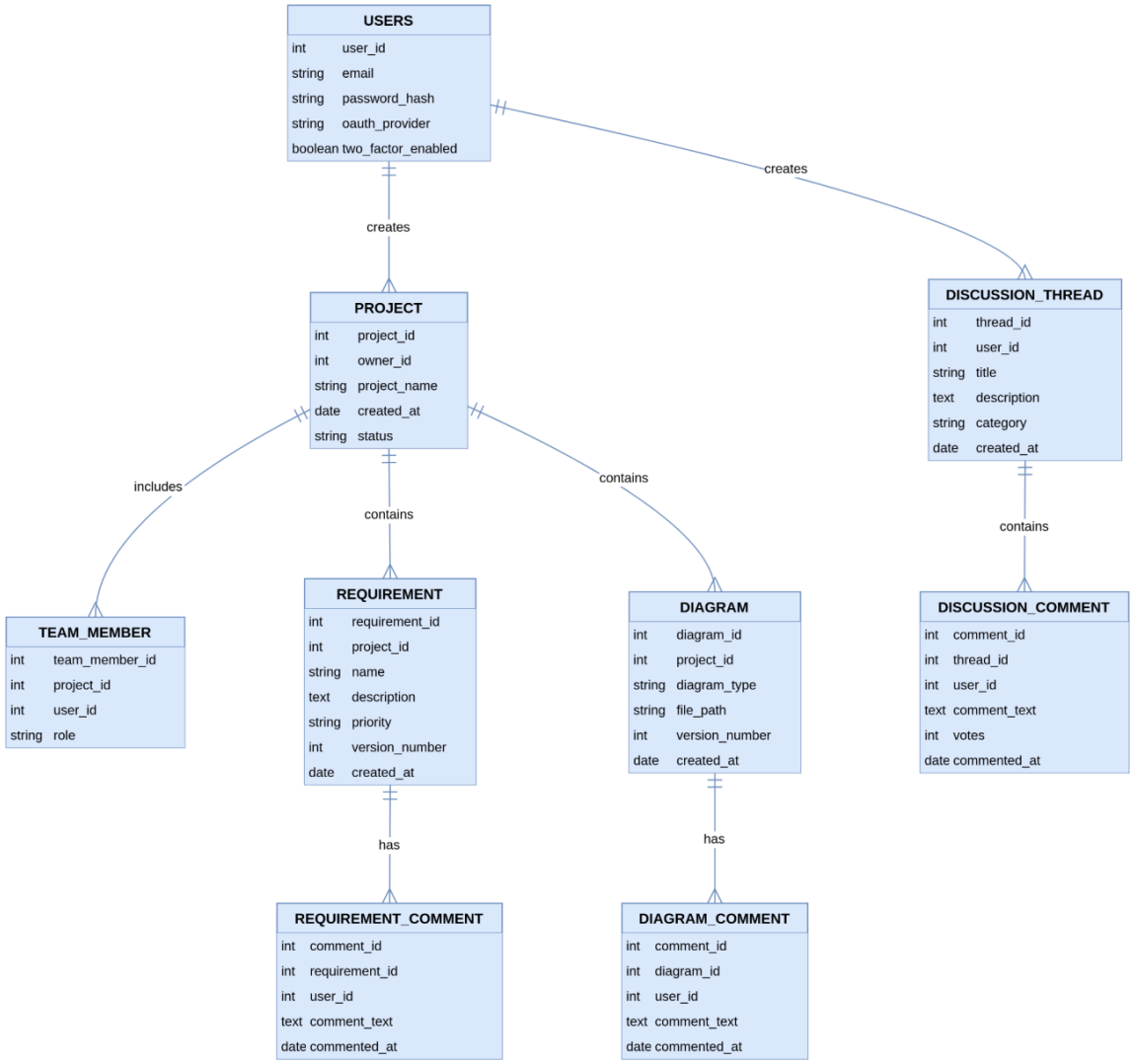


Figure 33 ERD Diagram for system

3.8 Network Diagram

Diagram 34 illustrates the network setup used by the system, showing how different components communicate across servers and devices. It highlights the flow of data and the connections that ensure smooth system operation.

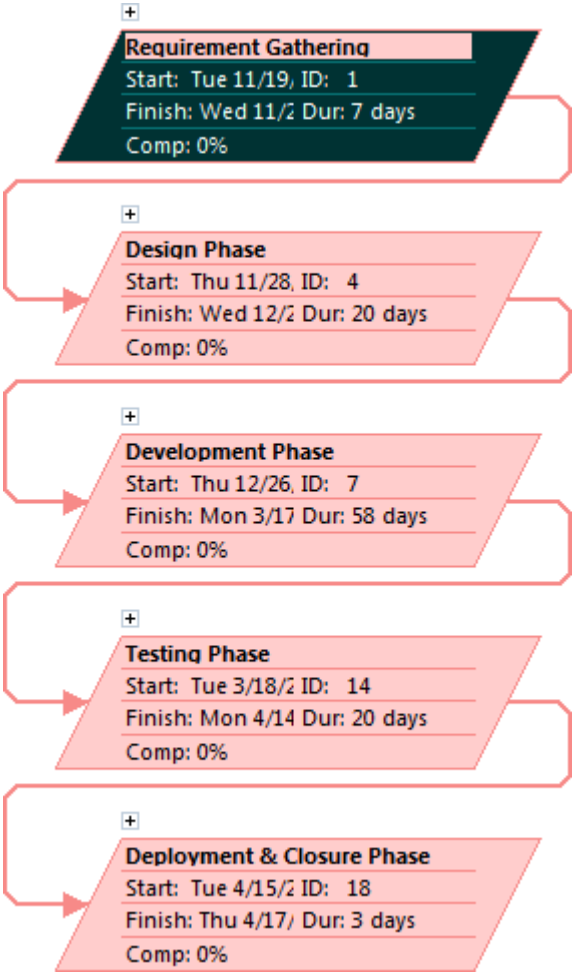


Figure 34 Network diagram for system

3.9 Collaboration Diagram

3.9.1 User Authorization

Diagram 35 shows the interaction between objects involved in authorizing a user. It highlights the message flow and relationships that ensure secure access control within the system.

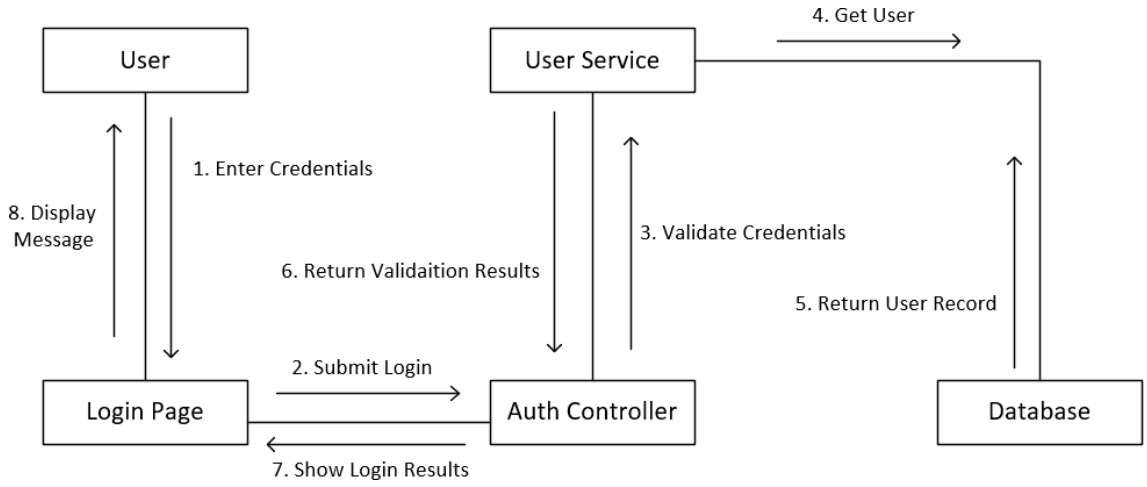


Figure 35 Collaboration Diagram for User Authorization

3.9.2 User Personalization

Diagram 36 illustrates how objects interact to manage user preferences and customization actions. It highlights the message flow that supports a personalized experience within the system.

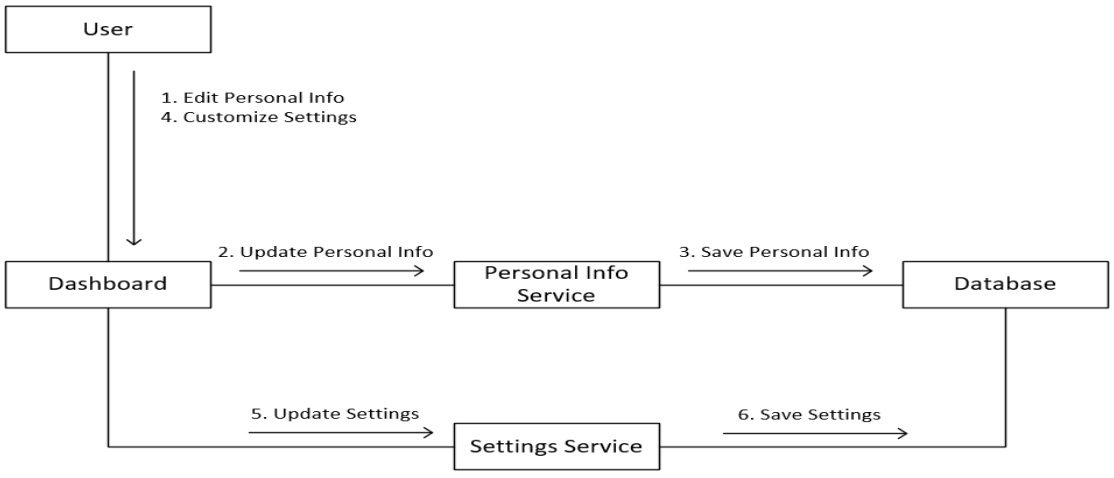
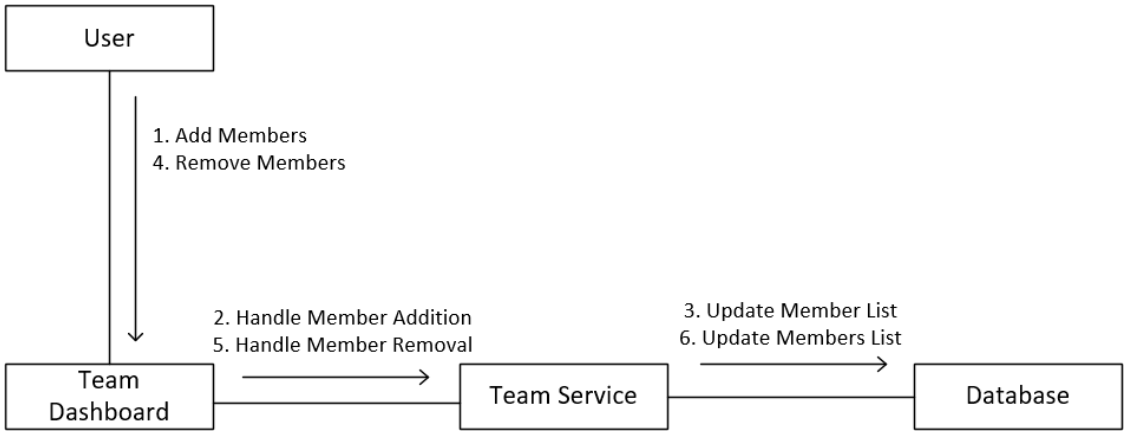


Diagram 36

3.9.3 Team Management

Diagram 37shows how different system objects interact to support team creation, updates, and coordination. It highlights the message flow required for effective team management operations.



37 Collaboration diagram fro team management

3.9.4 Discussion Forum

Diagram 38 illustrates the interactions between objects that handle forum discussions, including creating posts, commenting, and managing threads. It highlights how messages flow to support smooth communication within the forum.

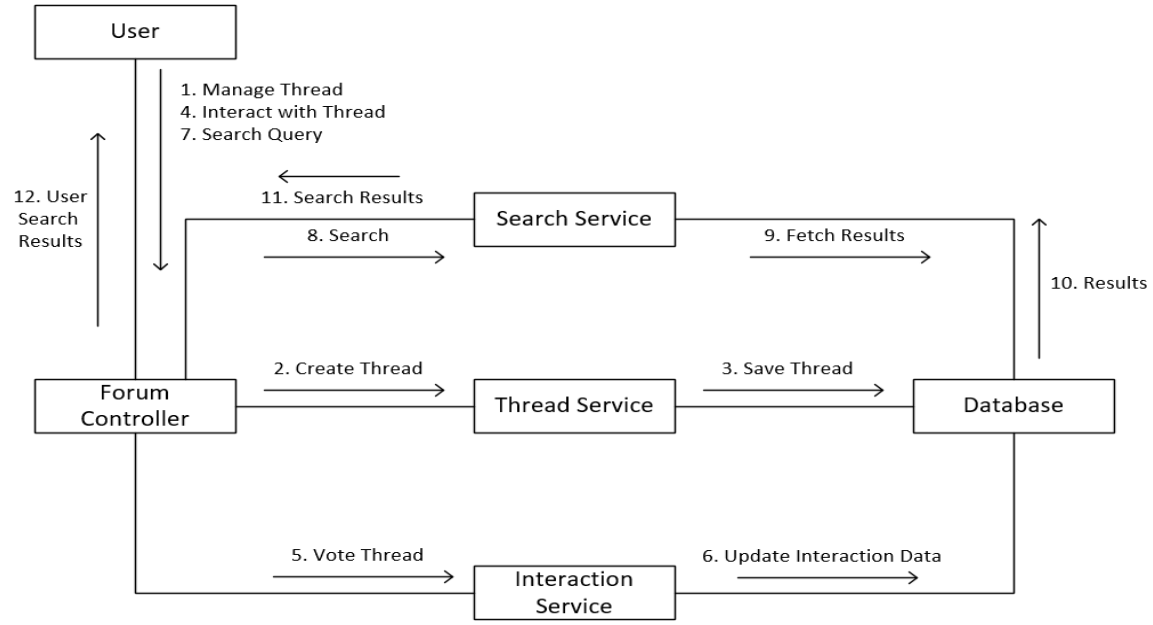


Figure 38 Collaboration Diagram for Discussion Forum

3.9.5 Requirement Management

Diagram 39 illustrates the interactions between objects that handle forum discussions, including creating posts, commenting, and managing threads. It highlights how messages flow to support smooth communication within the forum.

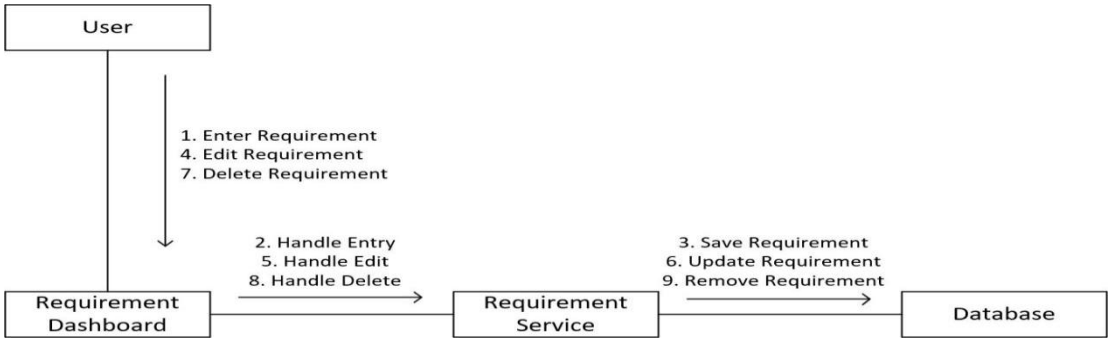


Figure 39 Collaboration Diagram for Requirement Management

3.9.6 Diagramming

Diagram 40 illustrates the interactions between objects involved in creating, editing, and managing diagrams. It highlights the message flow that enables efficient and collaborative diagram management within the system.

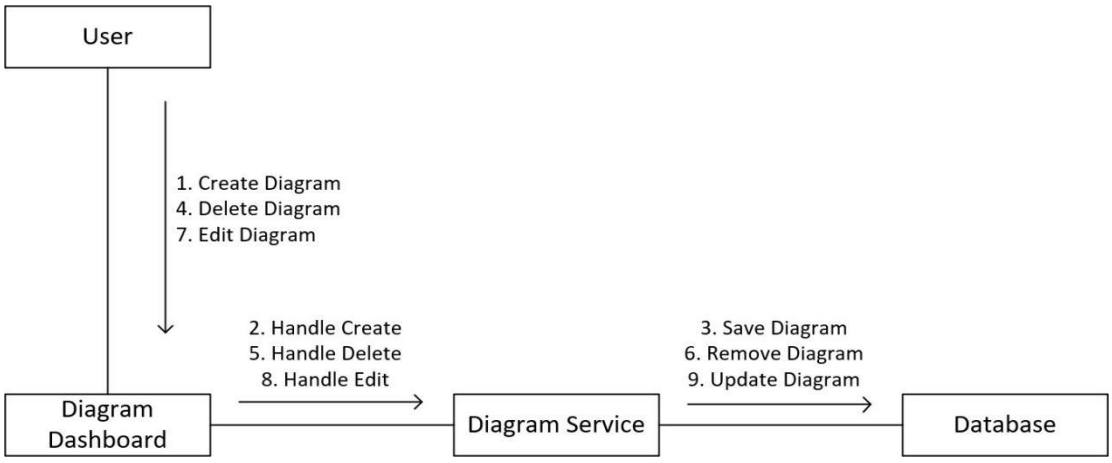


Figure 40 Collaboration Diagram for Diagramming

CHAPTER 4

DATA AND EXPERIMENTS (and/or IMPLMENTATION)

4.1 Tools and Technologies

4.1.1 Frontend Technologies

React.js

React.js was used to build the user interface of the application. Its reusable components, virtual DOM, and state management capabilities made it suitable for creating dynamic and interactive diagram-editing screens.

Tailwind CSS / CSS Frameworks

A modern utility-first CSS framework was used to design clean, responsive, and consistent UI layouts with minimal custom styling effort.

4.1.2 Backend Technologies

Node.js & Express.js

The backend was implemented using Node.js with Express.js to handle API routing, business logic, authentication, and server-side communication.

JWT (JSON Web Token)

JWT was used to implement secure login and role-based access control, ensuring that only authorized users can access or modify diagrams and requirement documents.

4.1.3 Database Technologies

MongoDB

MongoDB served as the primary database for storing:

- User accounts and roles
- UML diagrams
- Requirement documents (FRS/SRS)

- Collaboration history
  Its flexible document-based structure made it suitable for storing complex diagram objects.

4.1.4 Development & Collaboration Tools

Git & GitHub

Git was used for version control, enabling team collaboration. GitHub repositories helped track changes, manage branches, and maintain code history.

Postman

Postman was used for API testing, validating backend routes, authentication flows, and data exchange between the frontend and backend.

Figma / Mockup Design Tools

Figma or similar design tools were used to create initial UI wireframes and plan interface structure before development.

4.1.5 Hosting & Deployment Tools

Vercel / Netlify (Frontend)

The web frontend was deployed on modern cloud platforms for fast global access and automatic builds.

Render / Railway / Node Server (Backend)

Backend services were hosted using a cloud service capable of handling Node.js workloads.

Google Gemini API

The system uses the **Google Gemini API** to assist in automatic UML diagram generation. Gemini reads the user's textual requirements, identifies key elements, and returns structured information that is converted into UML components.

System Testing

4.2 Test cases

4.2.1 Test Case for User Login

Project name: CollabDesign

Module name: User authentication **Created by:** Muhammad Ilyas **Created date:** 20/08/2025**Reviewed by:** Muhammad Ilyas

Reviewed date: 20/11/2025

Test case ID	Test case description	Action/Steps	Test data	Expected results	Status	Actual results	Post-conditions
TC_001	Enter valid email and password	Enter valid email 2- Enter valid password 3- Press login button	1- email: xyz@gmail.com 2- password: abc123	Login should work with valid credentials	Pass	Login works	User should be able to see the landing page
TC_002	Enter valid email and invalid password	Enter valid email 2- Enter invalid password 3- Press login button	1- email: xyz@gmail.com 2- password: abc125	An error message should appear saying that the password is wrong	Pass	Error message appeared	The user should see a message saying wrong password
TC_003	Enter invalid email and valid password	1- Enter invalid email 2- Enter valid password 3- Press login button	1- email: abc@gmail.com 2- password: xyz123	An error message should appear saying that the user does not exist	Pass	Error message appeared	The user should see a message saying that user is not registered

Table 42 Test case for user login

4.2.2 Test Case for OTP Verification

Project name: CollabDesign
Module name: User authentication **Created by:** Muhammad Ilyas **Created date:** 20/08/2025 **Reviewed by:** Muhammad Ilyas
Reviewed date: 20/11/2025

Table 43 Test case for OTP verification

Test case ID	Test case description	Action/Steps	Test data	Expected results	Status	Actual results	Post-conditions
TC_001	Enter valid OTP	1- Enter valid OTP Press submit button	1-OTP: 547	Login should work with valid OTP	Pass	Login works	User should be able to see the landing page
TC_002	Enter invalid OTP	1- Enter invalid OTP Press submit button	1-OTP: 999	An error message should appear saying that the OTP is wrong	Pass	Error message appeared	The system will prompt the user for re-entering the OTP

4.2.3 Test Case for Two-Factor Authentication

Project name: CollabDesign

Module name: User authentication **Created by:** Muhammad Ilyas **Created date:** 20/08/2025 **Reviewed by:** Muhammad Ilyas **Reviewed date:** 20/11/2025

Table 44 Test case for two-factor authentication

Test case ID	Test case description	Action/Steps	Test data	Expected results	Status	Actual results	Post-conditions
TC_001	Enter valid OTP	1-Enter valid email 2-Enter valid password 3-Enter valid OTP	1-email: xyz@gmail.com 2-password: abc123 3-OTP: 547	Login should work with valid credentials	Pass	Login works	User should be able to see the landing page
TC_002	Enter invalid OTP	1-Enter valid email 2-Enter valid password 3-Enter invalid OTP	1-email: xyz@gmail.com 2-password: abc123 3-OTP: 999	An error message should appear prompting user for re- entry of OTP	Pass	Error message appeared	The user should see a message saying invalid OTP

4.2.4 Test Case for Editing Personal Information

Project name: CollabDesign

Module name: Personalisation

Created by: Muhammad Ilyas

Created date: 20/08/2025

Reviewed by: Muhammad Ilyas

Reviewed date: 20/11/2025

Table 45 Test case for editing personal information

Test case ID	Test case description	Action/Steps	Test data	Expected results	Status	Actual results	Post-conditions
TC_001	User edits personal information	1-User navigates to settings page 2-User makes valid changes	1-Username: Umer	System will validate and save the changes	Pass	Changes were saved	Personal information is updated and user will be able to see it
TC_002	User edits personal information	1-User navigates to settings page 2-User makes invalid changes	1-Username: Um£4	An error message should appear prompting user for sufficient and valid details	Pass	Error message appeared	The user should see a message saying what details are missing or incorrect

4.2.5 Test Case for Team Creation

Project name: CollabDesign

Module name: Team management

Created by: Muhammad Ilyas

Farooq **Created date:** 20/08/2024

Reviewed by: Muhammad Ilyas

Reviewed date: 20/11/2025

Table 46 Test case for team creation

Test case ID	Test case description	Action/Steps	Test data	Expected results	Status	Actual results	Post-conditions
TC_001	User creates a new team	1-User clicks on the create team button 2-User enters sufficient team details 3-User confirms the creation of a new team	1-Team name: A Team	System will validate the details and create a new team	Pass	New team was created	A new team will be created
TC_002	User creates a new team	1-User clicks on the create team button 2-User enters insufficient team details 3-User confirms the creation of a new team	1- Team name: Null	An error message should appear prompting user for sufficient details	Pass	Error message appeared	The user should see a message saying what details are missing or incorrect

4.2.6 Test Case for Team Editing

Project name: CollabDesign
Module name: Team management **Created by:** Muhammad Ilyas **Created date:** 20/08/2024 **Reviewed by:** Muhammad Ilyas
Reviewed date: 20/11/2025

Table 47 Test case for team editing

Test case ID	Test case description	Action/Steps	Test data	Expected results	Status	Actual results	Post-conditions
TC_001	User edits a team	1- User selects the team 2- User makes valid changes 3- User confirms the changes	Team name: Backup team	System will validate the required information and edit team details	Pass	Team details were edited	User will be able to see the team with updated details
TC_002	User creates a new team	1- User selects the team 2- User makes invalid changes 3- User confirms the changes	Team name: Null	An error message should appear prompting user for required information	Pass	Error message appeared	The user should see a message saying what details are missing or incorrect

4.2.7 Test Case for Requirement Management

Project name: CollabDesign

Module name: Requirement management

Created by: Muhammad Ilyas

Created date: 25/08/2024

Reviewed by: Muhammad Ilyas

Reviewed date: 20/11/2025

Table 48 Test case for requirement management

Test case ID	Test case description	Action/Steps	Test data	Expected results	Status	Actual results	Post-conditions
TC_001	User creates a new requirement	1- User clicks on the new requirement button 2- User enters valid requirement details 3-User clicks the save button	Requirement ID: 01	System will validate the details and create a new requirement	Pass	New requirement was created	A new requirement will be created
TC_002	User creates a new requirement	1- User clicks on the new requirement button 2- User enters invalid requirement details 3 - User clicks the save button	Requirement ID: x5	An error message should appear prompting user for valid details	Pass	Error message appeared	The user should see a message saying what details are missing or invalid

4.2.8 Test Case for Editing Requirements

Project name: CollabDesign

Module name: Requirement management

Created by: Muhammad Ilyas

Created date: 25/08/2025

Reviewed by: Muhammad Ilyas

Reviewed date: 20/11/2025

Table 49 Test case for editing requirements

Test case ID	Test case description	Action/Steps	Test data	Expected results	Status	Actual results	Post-conditions
TC_001	User edits an existing requirement	1- User selects the requirement 2- User edits requirement details with valid information 3-User presses the save button	Requirement ID: 01	System will validate the details and edit the requirement	Pass	Requirement was edited	The user will be able to see the updated requirement
TC_002	User edits an existing requirement	1- User selects the requirement 2- User edits requirement details with invalid information 3-User presses the save button	Requirement ID: x5	An error message should appear prompting user for valid details	Pass	Error message appeared	The user should see a message saying what details are missing or invalid

4.2.9 Test Case for Searching Requirements

Project name: CollabDesign

Module name: Requirement management

Created by: Muhammad Ilyas

Created date: 25/08/2025

Reviewed by: Muhammad Ilyas

Reviewed date: 20/11/2025

Table 50 Test case for searching requirements

Test case ID	Test case description	Action/Steps	Test data	Expected results	Status	Actual results	Post-conditions
TC_001	User searches an existing requirement	1-User selects the project 2-User enters the requirement keyword	Keyword: Team	System will search for the keyword in existing project requirements	Pass	System searched for keyword	The user will be able to see the requirements containing the keyword
TC_002	User searches for a non-existing requirement	1-User selects the project 2-User enters the requirement keyword	Keyword: Death	System will show a message saying no such requirements were found	Pass	Message appeared	The user should see a message saying that no requirements containing the keyword exists

4.2.10 Test Case for Commenting Requirements

Project name: CollabDesign

Module name: Requirement management

Created by: Muhammad Ilyas

Created date: 28/08/2025

Reviewed by: Muhammad Ilyas

Reviewed date: 25/11/2025

Table 51 Test case for commenting requirements

Test case ID	Test case description	Action/Steps	Test data	Expected results	Status	Actual results	Post-conditions
TC_001	User comments on a requirement	1-User selects the requirement 2-User writes a comment 3-User presses the comment button	Comment: Good work	System responds by adding the comment under the requirement	Pass	Comment was added	The user will be able to see the added comments
TC_002	User searches does not comment on a requirement	1-User selects the requirement 2- User does not write a comment 3-User presses the comment button	Comment: Null	System will show an error message saying comment was not added	Pass	Error message appeared	No comment will be added and system will show user an error message saying comment is empty

4.2.11 Test Case for Use Case Diagram Generation

Project name: CollabDesign

Module name: Requirement management

Created by: Muhammad Ilyas

Created date: 28/08/2025

Reviewed by: Muhammad Ilyas

Reviewed date: 28/11/2025

Table 52 Test case for use case diagram generation

Test case ID	Test case description	Action/Steps	Test data	Expected results	Status	Actual results	Post-conditions
TC_001	User wants to generate a use case diagram	1-User completely fills out the requirement form 2-User presses the generate button	Complete form	System responds by generating a use case diagram	Pass	Use case diagram generated	The user will be able to see the use case diagram
TC_002	User wants to generate a use case diagram	1-User partially fills out the requirement form 2-User presses the generate button	Incomplete form	System will show a error message saying necessary fields are missing in the requirement form	Pass	Use case diagram not generated	Use case diagram will not be generated and system will highlight missing fields

4.2.12 Test Case for Downloading Files

Project name: CollabDesign

Module name: File management

Created by: Muhammad Ilyas

Created date: 28/08/2025

Reviewed by: Muhammad Ilyas

Reviewed date: 28/11/2025

Table 53 Test case for downloading files

Test case ID	Test case description	Action/Steps	Test data	Expected results	Status	Actual results	Post-conditions
TC_001	User wants to download a file	1- User presses download icon 2- User selects format 3-User presses download button	format: PDF	System will save the file to the user's specified location	Pass	System saved file	User will be able to see the downloaded file on their computer

4.2.13 Test Case for Creating Threads

Project name: CollabDesign

Module name: Discussion forum

Created by: Muhammad Ilyas

Farooq **Created date:** 28/08/2025

Reviewed by: Muhammad Ilyas

Reviewed date: 28/11/2025

Table 54 Test case for creating a thread

Test case ID	Test case description	Action/Steps	Test data	Expected results	Status	Actual results	Post-conditions
TC_001	User creates a thread	1-User navigates to discussion forum page 2- User inputs title and description for the thread 3-User posts thread	1-Title: Diagram 2-Description: Lorem Ipsum	A new thread should be crated with the specified title and description	Pass	New thread was created	User will be able to see the newly created thread
TC_002	User creates a thread	1-User navigates to discussion forum page 2-User skips title and inputs description for the thread 3-User posts thread	1-Title: Null 2-Description: Lorem Ipsum	An error message should appear saying that a title is required	Pass	Error message appeared	The user should see a message saying thread title is missing
TC_003	User creates a thread	1-User navigates to discussion forum page 2-User inputs title and skips description for the thread 3-User posts thread	1-Title: Format 2-Description: Null	An error message should appear saying that a description is required	Pass	Error message appeared	The user should see a message saying that thread description is missing

4.2.14 Test Case for Editing Threads

Project name: CollabDesign

Module name: Discussion forum

Created by: Muhammad Faris Hamza

Created date: 05/09/2025

Reviewed by: Muhammad Ilyas

Reviewed date: 05/11/2025

Table 55 Test case for editing threads

Test case ID	Test case description	Action/Steps	Test data	Expected results	Status	Actual results	Post-conditions
TC_001	User wants to edits a thread	1-User navigates to discussion forum page 2-User selects the thread 3-User makes changes 4-User saves changes	Save changes	The system will update the thread according to user changes	Pass	Thread was updated	User will be able to see the edited thread
TC_002	User wants to edit a thread	1-User navigates to discussion forum page 2-User selects the thread 3-User makes changes 4-User does not save changes	Do not save changes	System will not update the thread	Pass	Thread was not updated	The user will see a message saying changes were cancelled

4.2.15 Test Case for Deleting Threads

Project name: CollabDesign

Module name: Discussion forum

Created by: Muhammad Faris Hamza

Created date: 05/09/2025

Reviewed by: Muhammad Ilyas

Reviewed date: 05/11/2025

Table 56 Test case for deleting threads

Test case ID	Test case description	Action/Steps	Test data	Expected results	Status	Actual results	Post-conditions
TC_001	User wants to delete a thread	1- User navigates to discussion forum page 2- User selects the thread 3- User presses delete button 4- User confirms deletion	Confirm deletion	The system will delete the selected thread	Pass	Thread was deleted	User will not be able to see the deleted thread anymore and system will show a message saying the thread was deleted
TC_002	User wants to delete a thread	1- User navigates to discussion forum page 2- User selects the thread 3- User presses delete button 4- User cancels confirmation	Cancel deletion	System will not delete the thread	Pass	Thread was not deleted	User will still be able to see the thread and the thread will not be deleted

4.2.16 Test Case for Commenting on Threads

Project name: CollabDesign

Module name: Discussion forum

Created by: Muhammad Arsal Farooq **Created date:** 05/15/2024

Reviewed by: Muhammad Faisal

Reviewed date: 11/06/2025

Table 57 Test case for commenting on threads

Test case ID	Test case description	Action/Steps	Test data	Expected results	Status	Actual results	Post-conditions
TC_001	User wants to comment on a thread	1- User navigates to discussion forum page 2- User selects the thread 3- User writes their comment 4- User presses the comment button	Comment: How did you get this to work?	The system will show the user's comment(s) under the thread	Pass	Comments can be seen under the thread	User will be able to see their comment under the thread and system will show a message saying comment was posted
TC_002	User wants to comment on a thread	1- User navigates to discussion forum page 2- User selects the thread 3- User writes nothing in the comment 4- User presses the comment button	Comment: Null	System will not add the user comment and will instead show a message to the user	Pass	Comment was not added	The system will highlight the comment section

4.2.17 Test Case for Voting on Threads

Project name: CollabDesign

Module name: Discussion forum

Created by: Muhammad Faris Hamza

Created date: 05/09/2025

Reviewed by: Muhammad Ilyas

Reviewed date: 05/11/2025

Table 58 Test case for voting on threads

Test case ID	Test case description	Action/Steps	Test data	Expected results	Status	Actual results	Post-conditions
TC_001	User wants to upvote a thread	1- User navigates to discussion forum page 2- User selects the thread 3- User selects the upvote option under the thread	Select upvote option	System will highlight the upvote button under the thread	Pass	Upvote button was highlighted	User will be able to see the thread and their reaction which will be highlighted
TC_002	User wants to downvote a thread	1- User navigates to discussion forum page 2- User selects the thread 3- User selects the downvote option under the thread	Select downvote option	System will highlight the downvote button under the thread	Pass	Downvote button was highlighted	User will be able to see the thread and their reaction which will be highlighted

4.2.18 Test Case for Searching Discussion Forum

Project name: CollabDesign

Module name: Requirement management

Created by: Muhammad Faris Hamza

Created date: 05/09/2025

Reviewed by: Muhammad Ilyas

Reviewed date: 05/11/2025

Table 59 Test case for searching discussion forum

Test case ID	Test case description	Action/Steps	Test data	Expected results	Status	Actual results	Post-conditions
TC_001	User searches an existing thread	1- User navigates to discussion forums page 2- User enters the keyword for an existing thread	Management	System will search for the keyword in existing threads and show results	Pass	System shows results	The user will be able to see the thread(s) containing the keyword
TC_002	User searches for a non-existing thread	1- User navigates to discussion forums page 2- User enters the keyword for a non-existing thread	Asteroid	System will show a message saying no such threads containing keyword were found	Pass	System shows message	The user should see a message saying that no threads containing the keyword exists

4.2.19 Test Case for Creating a Diagram

Project name: CollabDesign

Module name: Diagramming

Created by: Muhammad Faris Hamza

Created date: 05/09/2025

Reviewed by: Muhammad Ilyas

Reviewed date: 05/11/2025

Table 60 Test case for creating a diagram

Test case ID	Test case description	Action/Steps	Test data	Expected results	Status	Actual results	Post-conditions
TC_001	User wants to create a diagram	1- User navigates to the dashboard page 2- User selects the project 3- User presses the new diagram button	Press new diagram button	System navigates user to an empty canvas	Pass	User is navigated to an empty canvas	The user will be redirected to an empty canvas where they can create a diagram

4.2.20 Test Case for Editing a Diagram

Project name: CollabDesign

Module name: Diagramming

Created by: Hassan Mubasher

Created date: 05/09/2025

Reviewed by: Muhammad Ilyas

Reviewed date: 05/11/2025

Table 61 Test case for editing a diagram

Test case ID	Test case description	Action/Steps	Test data	Expected results	Status	Actual results	Post-conditions
TC_001	User wants to edit a diagram	1- User navigates to the dashboard page 2- User selects the project 3- User selects the diagram they want to edit	Select diagram	System navigates user to the canvas where the diagram is opened	Pass	System opens the diagram	The user will be redirected to the canvas where they can edit the diagram

4.2.21 Test Case for Deleting a Diagram

Project name: CollabDesign

Module name: Diagramming

Created by: Hassan Mubasher

Created date: 05/09/2025

Reviewed by: Muhammad Ilyas

Reviewed date: 05/11/2025

Table 62 Test case for deleting a diagram

Test case ID	Test case description	Action/Steps	Test data	Expected results	Status	Actual results	Post-conditions
TC_001	User wants to delete a diagram	1- User navigates to the dashboard page 2- User selects the project 3- User selects the diagram they want to delete 4- User confirms deletion	Confirm deletion	System responds by deleting the selected diagrams	Pass	Diagrams are deleted	The deleted diagrams will no longer be visible to the user
TC_002	User wants to delete a diagram	1- User navigates to the dashboard page 2- User selects the project 3- User selects the diagram they want to delete 4- User cancels deletion on confirmation	Cancel deletion	System responds by closing the confirmation message	Pass	Message is closed	The diagram will not be deleted and will still be visible to the user

4.2.22 Test Case for Downloading a Diagram

Project name: CollabDesign

Module name: Diagramming

Created by: Hassan Mubasher

Created date: 05/09/2025

Reviewed by: Muhammad Ilyas

Reviewed date: 05/11/2025

Table 63 Test case for downloading a diagram

Test case ID	Test case description	Action/Steps	Test data	Expected results	Status	Actual results	Post-conditions
TC_001	User wants to download a diagram	1- User navigates to the dashboard page 2- User selects the project 3- User selects the diagram they want to download 4- User specifies file format in which they want to download diagram 5- User presses download button	Format: PNG	System responds by downloading diagrams to the user's specified location in the user's specified format	Pass	Diagram is downloaded	The user will be able to see the diagram in the specified location and format on their system

4.2.23 Test Case for Commenting on a Diagram

Project name: CollabDesign

Module name: Diagramming

Created by: Hassan Mubasher

Created date: 05/09/2025

Reviewed by: Muhammad Ilyas

Reviewed date: 05/11/2025

Table 64 Test case for commenting on a diagram

Test case ID	Test case description	Action/Steps	Test data	Expected results	Status	Actual results	Post-conditions
TC_001	User wants to comment on a diagram	1- User navigates to the dashboard page 2- User selects the project 3- User selects the diagram on which they want to comment 4- User writes their comment 5- User presses add comment button	Comment: Edit this diagram	System responds by adding the user comments to the diagram	Pass	Comments are added	The user can see a list of all the comments under the diagram including their own comment(s)
TC_002	User wants to comment on a diagram	1- User navigates to the dashboard page 2- User selects the project 3- User selects the diagram on which they want to comment 4- User does not write a comment 5-User presses add comment button	1-Comment: Null	System responds by saying that the comment field is empty	Pass	Message is shown	No comment will be added to the diagram and the system will prompt the user to write something in the comment field

Chapter 5.

RESULTS AND DISCUSSIONS (or USER MANUAL)

5.1 Problem Faced and lesson learned

The early stages of the Collab Design project were faced by difficulties in requirement gathering and making sure all team members had a unified understanding of the project's goals. Sometimes, the interpretation of features and functionalities differed, which led to delays as getting everyone on the same page required multiple discussions and clarifications. Also, gathering comprehensive and clear requirements from potential users was difficult because their expectations and needs were diverse and sometimes contradictory. These challenges emphasized the significance of good communication and structured elicitation techniques. We learned the value of documenting requirements in a standardized format and Hi-Fi, which resulted in fewer ambiguities and served as a reference point throughout the development process. Moreover, the use of collaborative tools and setting up regular team meetings helped align our efforts and ensured a shared understanding, which eventually led to more cohesive and efficient teamwork.

5.2 Project Summary

The Collab Design platform focuses on meeting the increasing demand for collaborative tools in software development. Its features include requirement management, UML diagram generation, and team coordination, making it a powerful tool for teams and small organizations. The project required designing and implementing modules for user authentication, team management, requirement tracking, and diagram customization. The combination of role-based access control, and community features like discussion forums ensured a secure yet user-friendly experience. It brings the best of existing tools with unique capabilities such as auto-generating use case diagrams from requirements and enables secure coordination.

5.3 Future Work

In the future, we will definitely include AI-driven features, for example, intelligent suggestions for the refinement of requirements and automated correction of diagrams. Integration of third-party project management tools, such as Jira or Trello, could further improve workflow efficiency. The new range of supported diagram types and customization options could be more appealing to more users. In addition, we would further introduce advanced analytics for tracking project progress and team performance. This enhancement will ensure that Collab Design continues to grow and develop as a holistic, innovative tool for collaborative software development

5.4 Screenshots of our project

5.4.1 Home Page

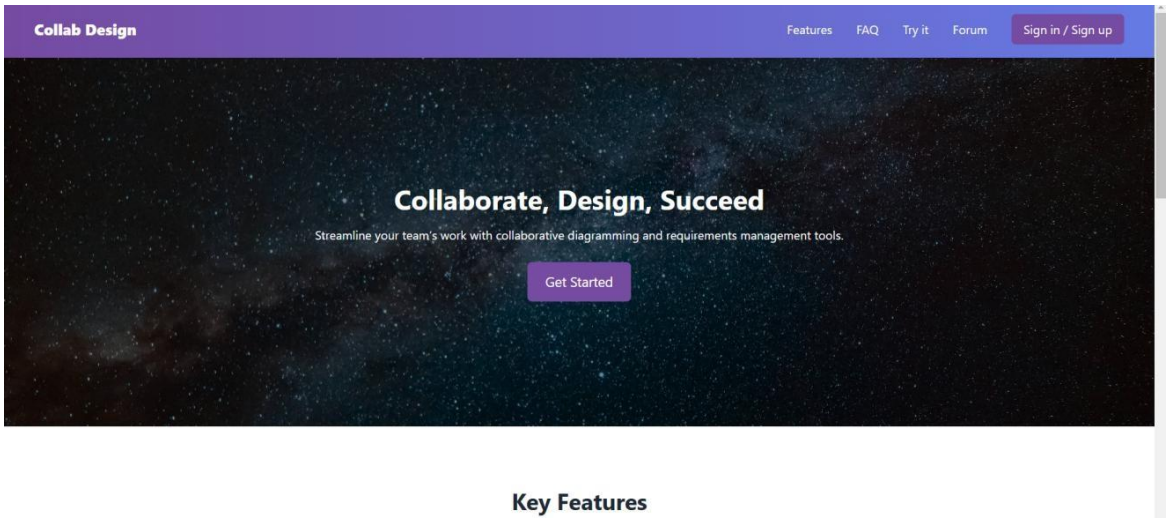


Figure 32 Home Page

5.4.2 OAuth (Sign Up)

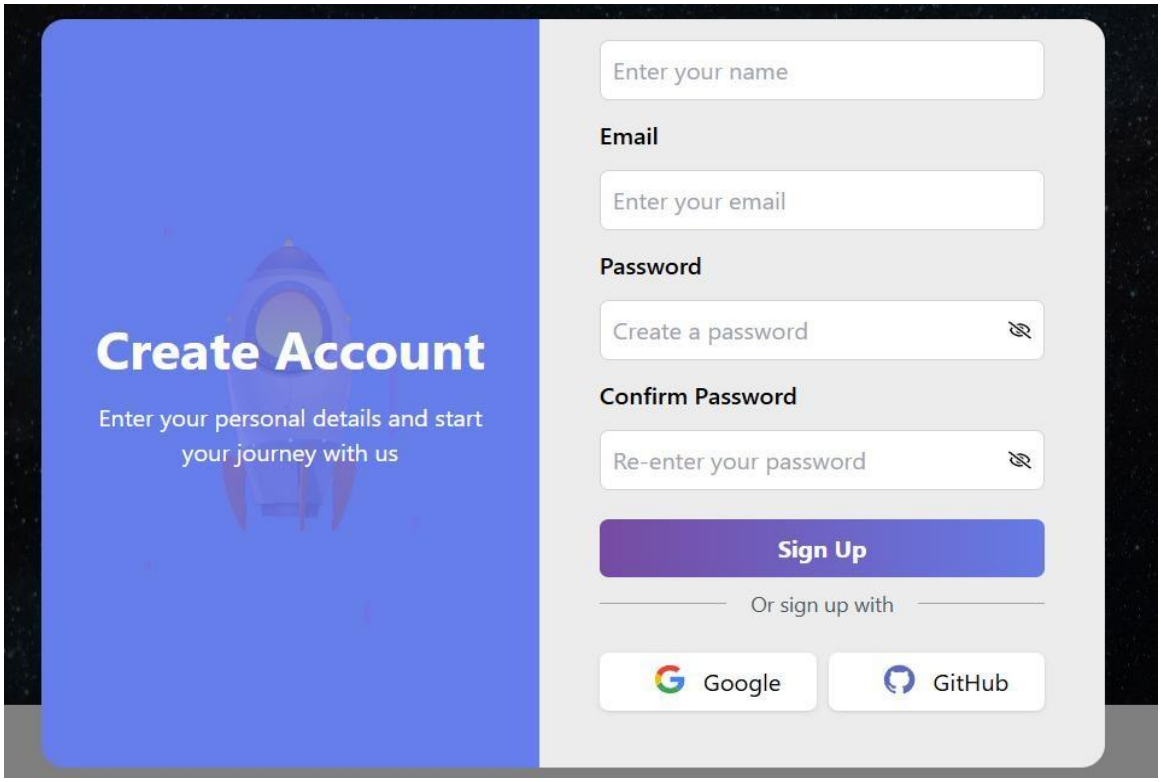


Figure 33 SOAuth (Sign Up

5.4.3 OAuth (Login)

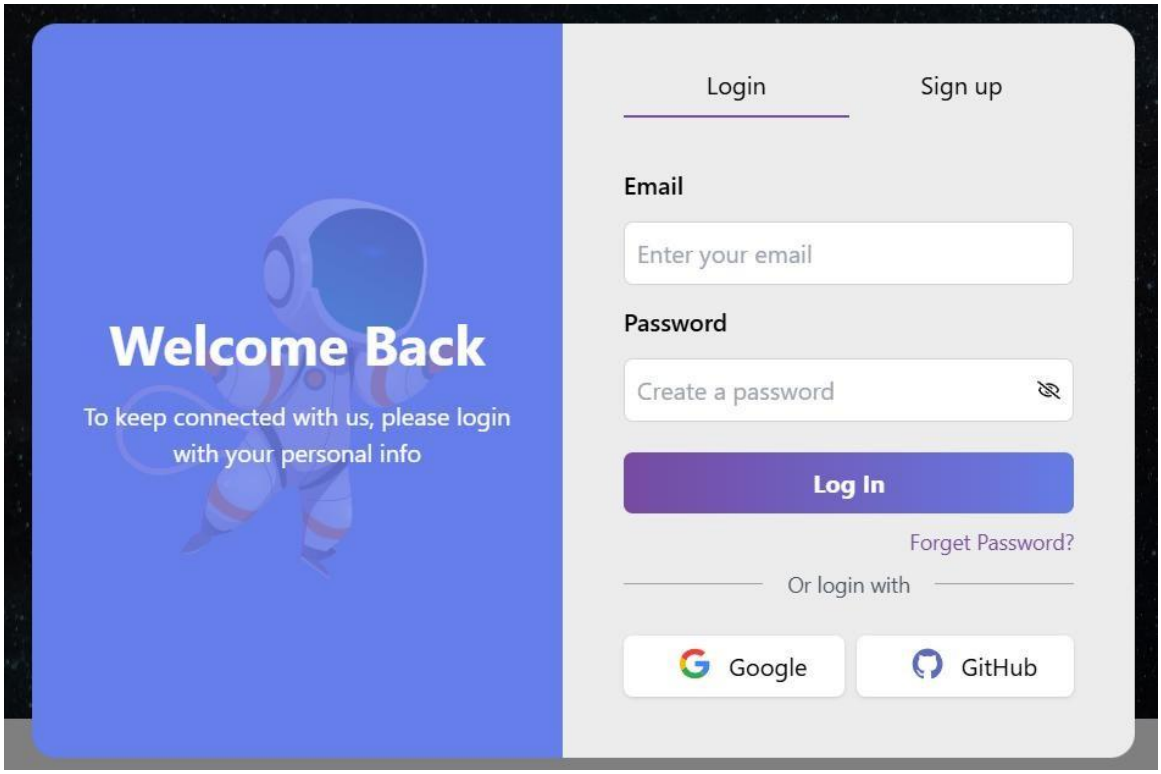


Figure 34 OAuth (Login)

5.4.4 Dashboard

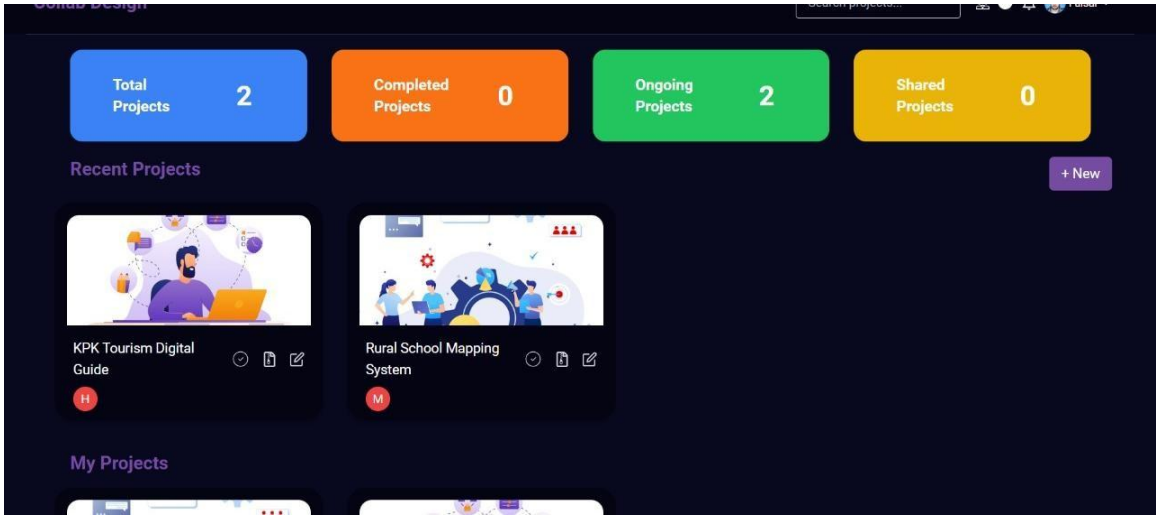


Figure 35 Dashboard

5.4.5 Requirement Management

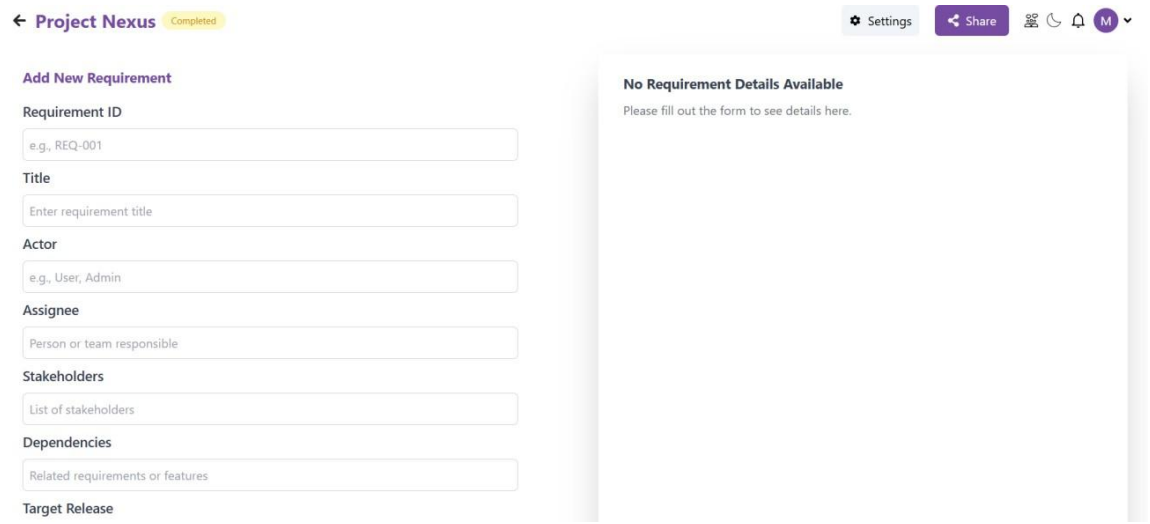


Figure 36 Requirement management

5.4.6 Team Management

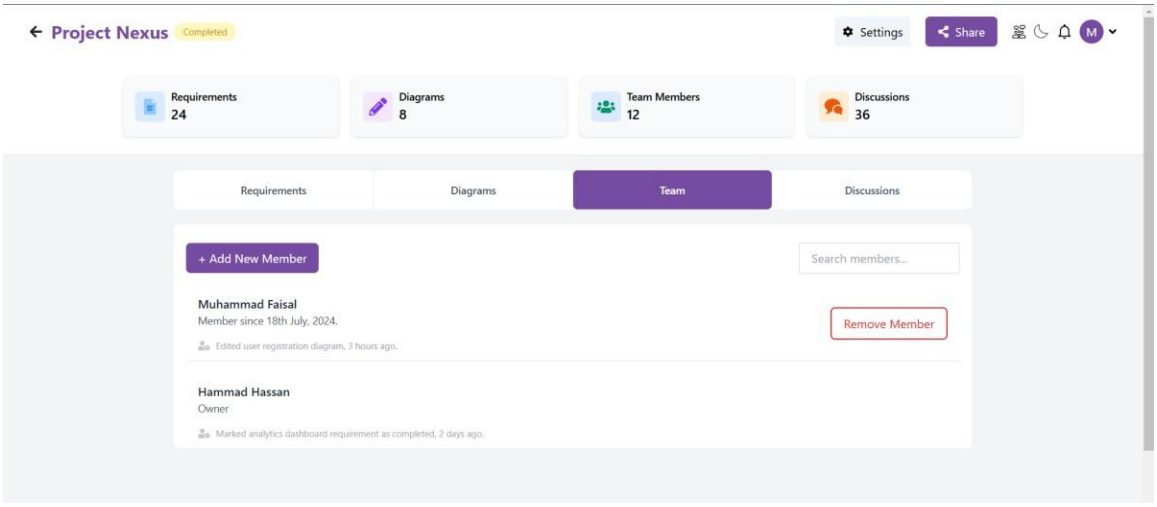


Figure 37 Team management

5.4.7 Discussion Forum

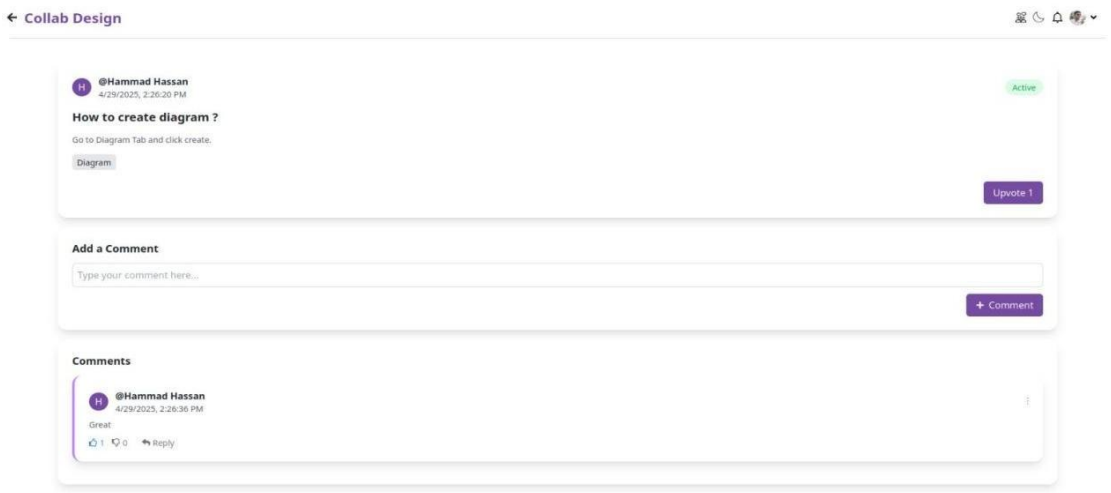


Figure 38 Discussion forum

5.4.8 Team Discussion

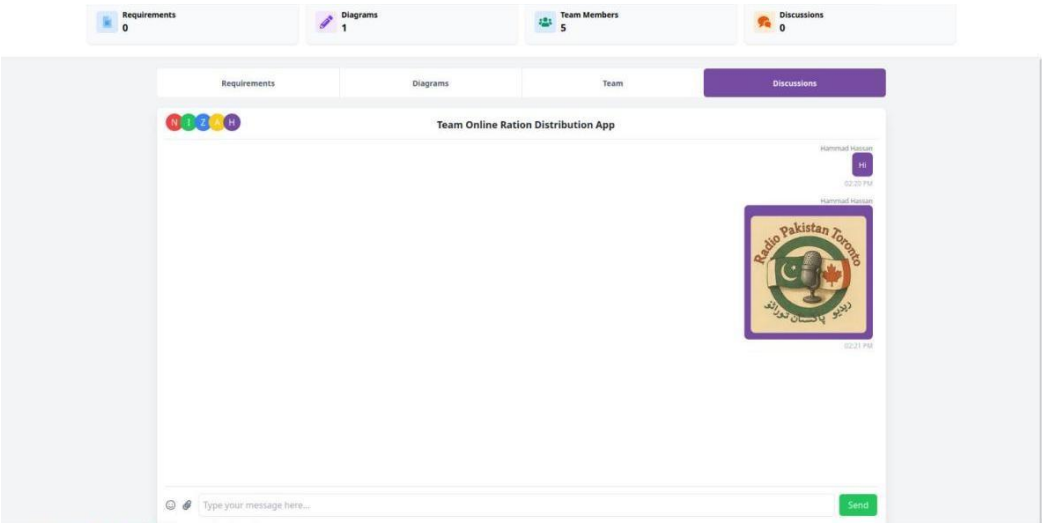


Figure 39 Team discussion

5.4.9 Diagramming

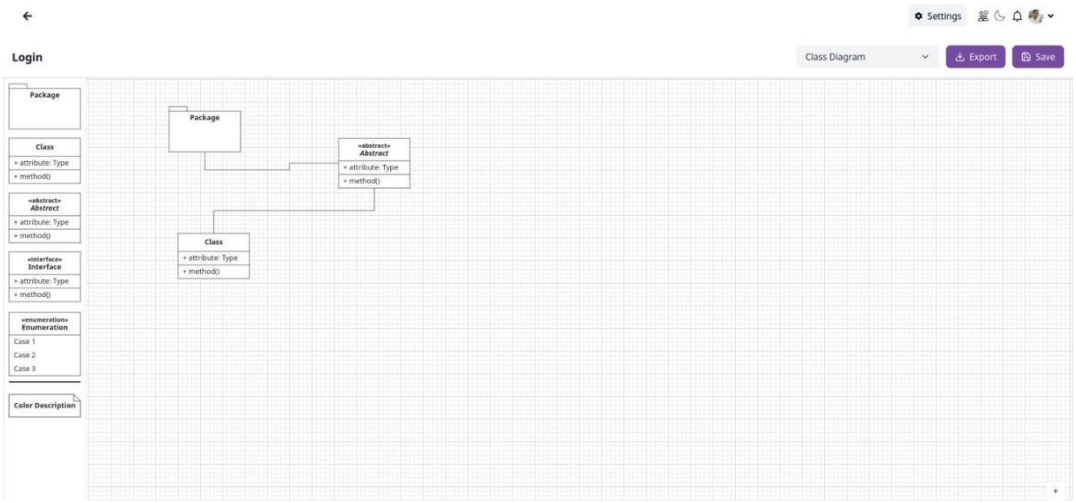


Figure 40 Diagramming

5.4.10 Project Details

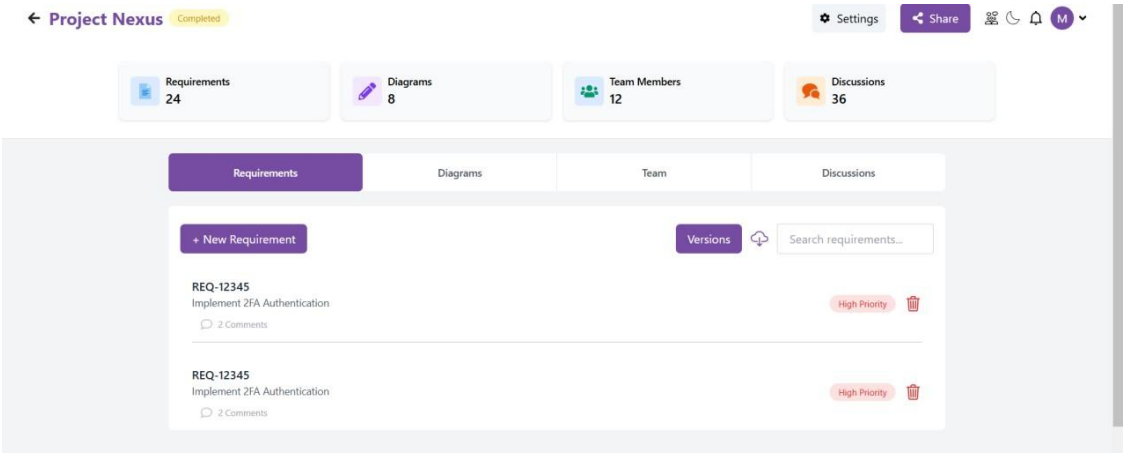


Figure 41 Project details

5.4.11 Dark Mode

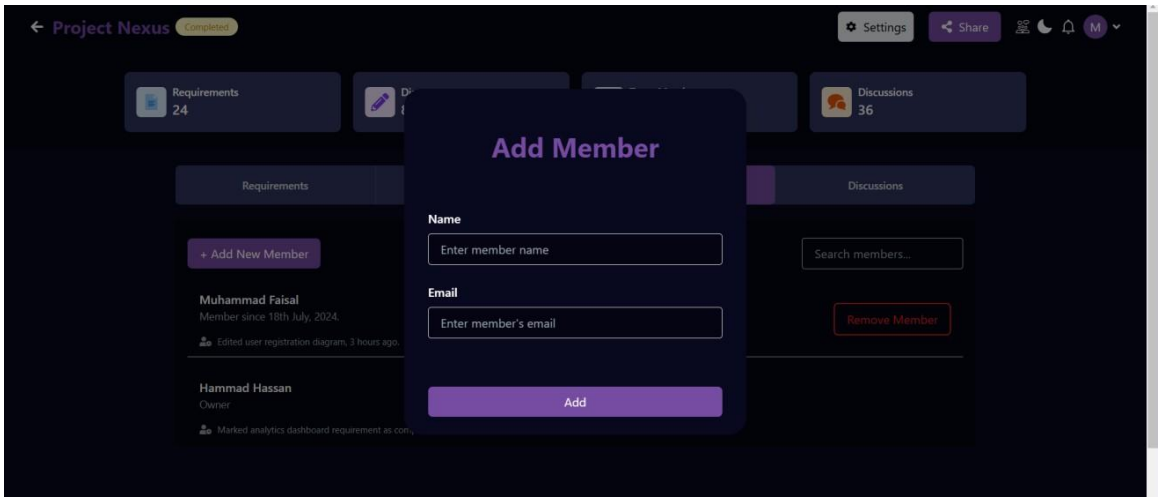


Figure 42 Dark mode

5.4.12 Version Control

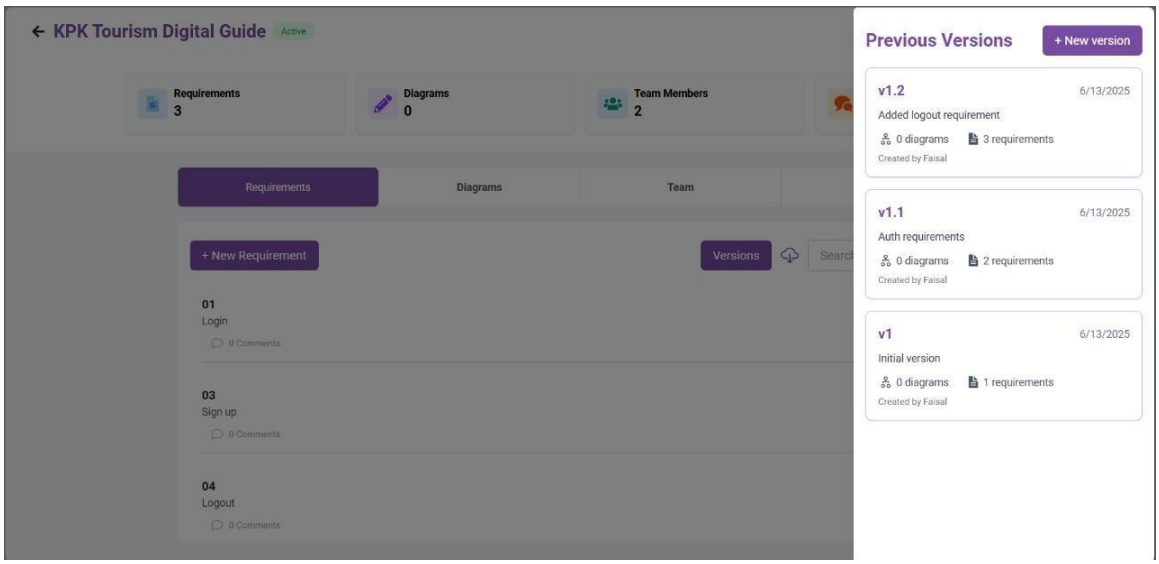


Figure 43 *Version Control*

6. CONCLUSION AND RECOMMENDATIONS

6.1 Problem Faced and lesson learned

The early stages of the Collab Design project were faced by difficulties in requirement gathering and making sure all team members had a unified understanding of the project's goals. Sometimes, the interpretation of features and functionalities differed, which led to delays as getting everyone on the same page required multiple discussions and clarifications. Also, gathering comprehensive and clear requirements from potential users was difficult because their expectations and needs were diverse and sometimes contradictory. These challenges emphasized the significance of good communication and structured elicitation techniques. We learned the value of documenting requirements in a standardized format and Hi-Fi, which resulted in fewer ambiguities and served as a reference point throughout the development process. Moreover, the use of collaborative tools and setting up regular team meetings helped align our efforts and ensured a shared understanding, which eventually led to more cohesive and efficient teamwork.

6.2 Project Summary

The Collab Design platform focuses on meeting the increasing demand for collaborative tools in software development. Its features include requirement management, UML diagram generation, and team coordination, making it a powerful tool for teams and small organizations. The project required designing and implementing modules for user authentication, team management, requirement tracking, and diagram customization. The combination of role-based access control, and community features like discussion forums ensured a secure yet user-friendly experience. It brings the best of existing tools with unique capabilities such as auto-generating use case diagrams from requirements and enables secure coordination.

6.3 Future Work

In the future, we will definitely include AI-driven features, for example, intelligent suggestions for the refinement of requirements and automated correction of diagrams. Integration of third-party project management tools, such as Jira or Trello, could further improve workflow efficiency. The new range of supported diagram types and customization options could be more

appealing to more users. In addition, we would further introduce advanced analytics for tracking project progress and team performance. This enhancement will ensure that Collab Design continues to grow and develop as a holistic, innovative tool for collaborative software development

References

[1] "Umletino – UML diagram online tool," Umletino, [Online]. Available: https://www.umletino.com/umletino.html. [Accessed: Sep. 26, 2025].

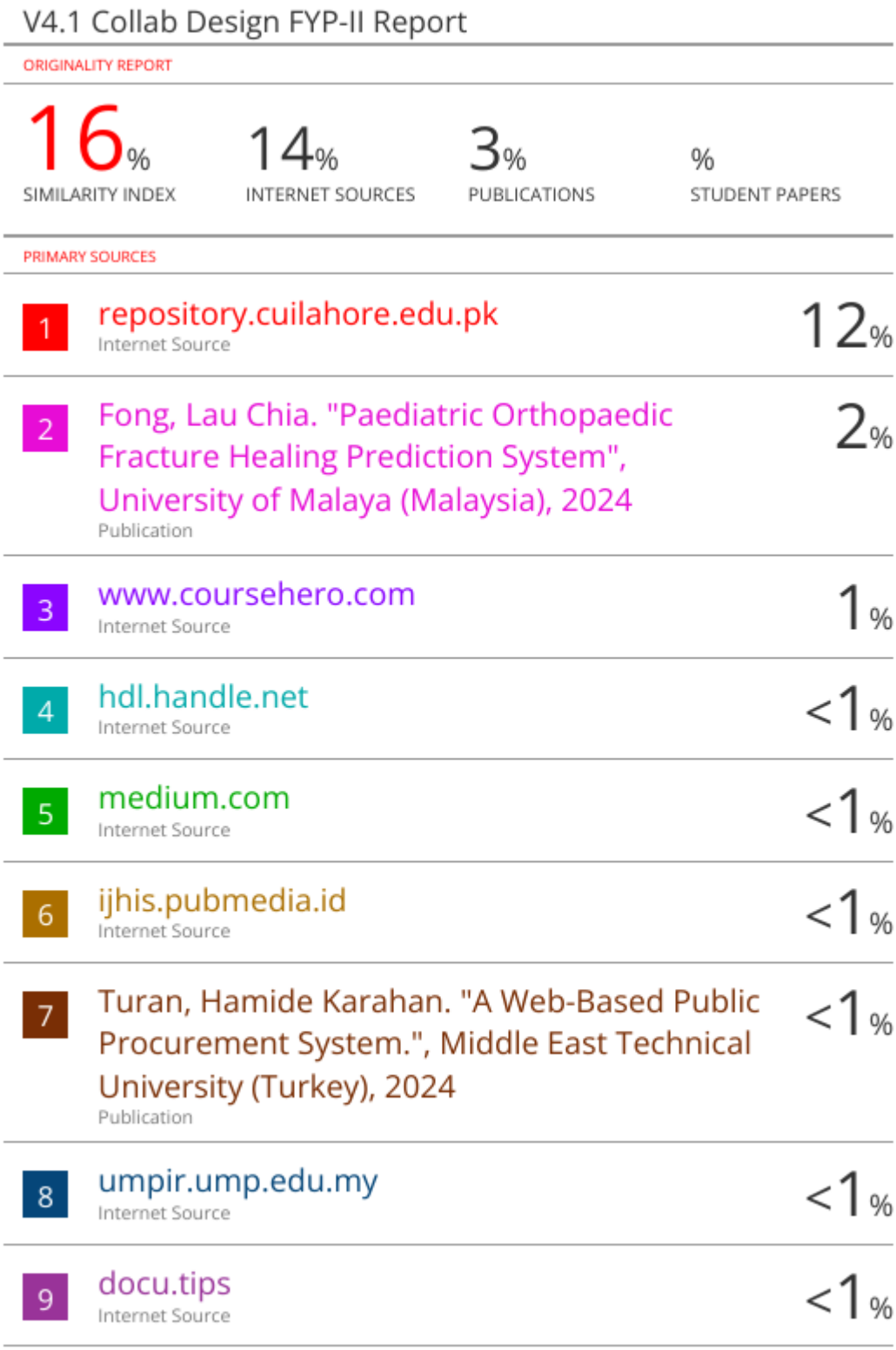
[2] "Erase – The fastest way to make your writing concise," Eraser, [Online]. Available: https://www.eraser.io/. [Accessed: Sep. 26, 2025].

[3] "diagrams.net – Free online diagram software," diagrams.net, [Online]. Available: https://app.diagrams.net/. [Accessed: Sep. 26, 2025].

[4] "IBM Engineering Requirements Management DOORS – Overview," IBM, [Online]. Available: https://www.ibm.com/docs/en/engineering-lifecycle-management-suite/doors/9.7.0?topic=overview-doors. [Accessed: Sep. 26, 2025].

[5] "Microsoft Visio – Flowchart software," Microsoft, [Online]. Available: https://www.microsoft.com/en-us/microsoft-365/visio/flowchart-software. [Accessed: Sep. 26, 2025].

APPENDICES



V4.1 Collab Design FYP-II Report

ORIGINALITY REPORT

16%	14%	3%	%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	repository.cuilahore.edu.pk Internet Source	12%
2	Fong, Lau Chia. "Paediatric Orthopaedic Fracture Healing Prediction System", University of Malaya (Malaysia), 2024 Publication	2%
3	www.coursehero.com Internet Source	1%
4	hdl.handle.net Internet Source	<1%
5	medium.com Internet Source	<1%
6	ijhis.pubmedia.id Internet Source	<1%
7	Turan, Hamide Karahan. "A Web-Based Public Procurement System.", Middle East Technical University (Turkey), 2024 Publication	<1%
8	umpir.ump.edu.my Internet Source	<1%
9	docu.tips Internet Source	<1%

10	Efe, Pinar. "A Unification Model and Tool Support for Software Functional Size Measurement Methods.", Middle East Technical University (Turkey), 2024 Publication	<1%
11	www.termpaperwarehouse.com Internet Source	<1%
12	Constantine Stephanidis, Gavriel Salvendy. "Designing for Usability, Inclusion and Sustainability in Human-Computer Interaction", CRC Press, 2024 Publication	<1%
13	Sujith Samuel Mathew, Mohammad Amin Kuhail, Maha Hadid, Shahbano Farooq. "The Object-Oriented Approach to Problem Solving and Machine Learning with Python", CRC Press, 2025 Publication	<1%
14	www.geeksforgeeks.org Internet Source	<1%

Exclude quotes	Off	Exclude matches	Off
Exclude bibliography	Off		

*% detected as AI

AI detection includes the possibility of false positives. Although some text in this submission is likely AI generated, scores below the 20% threshold are not surfaced because they have a higher likelihood of false positives.

Caution: Review required.

It is essential to understand the limitations of AI detection before making decisions about a student's work. We encourage you to learn more about Turnitin's AI detection capabilities before using the tool.

Disclaimer

Our AI writing assessment is designed to help educators identify text that might be prepared by a generative AI tool. Our AI writing assessment may not always be accurate (it may misidentify writing that is likely AI generated as AI generated and AI paraphrased or likely AI generated and AI paraphrased writing as only AI generated) so it should not be used as the sole basis for adverse actions against a student. It takes further scrutiny and human judgment in conjunction with an organization's application of its specific academic policies to determine whether any academic misconduct has occurred.

Frequently Asked Questions

How should I interpret Turnitin's AI writing percentage and false positives?

The percentage shown in the AI writing report is the amount of qualifying text within the submission that Turnitin's AI writing detection model determines was either likely AI-generated text from a large-language model or likely AI-generated text that was likely revised using an AI-paraphrase tool or word spinner.

False positives (incorrectly flagging human-written text as AI-generated) are a possibility in AI models.

AI detection scores under 20%, which we do not surface in new reports, have a higher likelihood of false positives. To reduce the likelihood of misinterpretation, no score or highlights are attributed and are indicated with an asterisk in the report (*%).

The AI writing percentage should not be the sole basis to determine whether misconduct has occurred. The reviewer/instructor should use the percentage as a means to start a formative conversation with their student and/or use it to examine the submitted assignment in accordance with their school's policies.

What does 'qualifying text' mean?

Our model only processes qualifying text in the form of long-form writing. Long-form writing means individual sentences contained in paragraphs that make up a longer piece of written work, such as an essay, a dissertation, or an article, etc. Qualifying text that has been determined to be likely AI-generated will be highlighted in cyan in the submission, and likely AI-generated and then likely AI-paraphrased will be highlighted purple.

Non-qualifying text, such as bullet points, annotated bibliographies, etc., will not be processed and can create disparity between the submission highlights and the percentage shown.