



**A Project Report On  
“SDLC Manager”**

Submitted in the partial fulfillment of the requirements for the award of the Degree of Bachelor  
of Science in Software Engineering

Submitted By  
Tonmoy Chiran  
181-35-2289

Under the support & guidance of  
Md. Shohel Arman  
Senior Lecturer  
Department Of Software Engineering

Batch: 25th  
Department Of Software Engineering  
Daffodil International University  
102/1, Sukrabad Mirpur Road, Dhaka 1207.

© All right Reserved by Daffodil International University

## APPROVAL OF PROJECT

This project titled “SDLC Manager”, submitted by Tonmoy Chiran, ID: 181-35-2289 to the Department of Software Engineering, Daffodil International University has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of B.Sc. in Software Engineering and approved as to its style and contents.

### BOARD OF EXAMINERS



-----  
Chairman

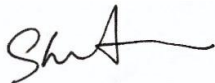
Dr. Imran Mahmud

Associate Professor and Head  
Department of Software Engineering  
Daffodil International University



-----  
Internal Examiner 1

Kaushik Sarker  
Assistant Professor  
Department of Software Engineering  
Daffodil International University



-----  
Internal Examiner 2

Md. Shohel Arman  
Senior Lecturer  
Department of Software Engineering  
Daffodil International University



-----  
External Examiner

Md. Fazle Munim  
Technology Expert  
Access to Information (a2i) Programme

## DECLARATION

I hereby declare that I have done this project under the supervision of Md. Shohel Arman, Senior Lecturer, Department of Software Engineering, Daffodil International University. I also declare that this project or any part of this is unique and has not been submitted elsewhere for the award of any degree.

Tonmoy

Tonmoy Chiran  
181-35-2289  
Department of Software Engineering  
Faculty Of Science & Information Technology  
Daffodil International University

Certified by:

ShA

Md. Shohel Arman  
Senior Lecturer  
Department Of Software Engineering  
Faculty Of Science & Information Technology  
Daffodil International University

## **ACKNOWLEDGEMENT**

At first, I am blessed that I successfully moved towards the last semester. I am pleased with my almighty. First, at the beginning of university life, I have learned a lot about software development as well as computer science-related knowledge from my university's knowledgeable teachers and helpful course mates. Teachers teach us ethics, morality, and politeness as well as software knowledge and related knowledge. I must be thankful to my parents and my family to give me the opportunity and always be to myself. My family always supports me. I am highly indebted to Md. Shohel Arman for his guidance and constant supervision as well as for providing necessary information regarding the project & for his support in completing the project. My supervisor supports me to make this project "SDLC Manager" a successful end. My thanks and appreciations go to my course mates in developing the project and people who have willingly helped me out with their abilities.

## **ABSTRACT**

"SDLC Manager" is a web based project sdhc phase tracker to help individual developers or developer groups to checkout, in which phase currently they are in. This platform encourages developers to follow SDLC phases, so they can make a quality full software. Every development of the software industry follows some rules and methods; I also followed some methods to develop this project. I finished all of the development by following the Agile methodology. I think agile methodology was the best choice for me, as i had to make a particular module and test them and Agile Methodology helped me to reduce the project risk.

## List Of Figures

<b>Figure 1.1: Gantt Chart</b>	5
<b>Figure 2.1: Agile Model</b>	10
<b>Figure 2.2 : Use Case Diagram</b>	12
<b>Figure 3.1: Activity Diagram - Registration</b>	26
<b>Figure 3.2: Activity Diagram - Login</b>	27
<b>Figure 3.3: Activity Diagram - Create Workspace</b>	28
<b>Figure 3.4: Activity Diagram - Registration</b>	29
<b>Figure 3.5: Activity Diagram - Specific Workspace</b>	30
<b>Figure 3.6: Activity Diagram - Workspace Sharing</b>	31
<b>Figure 3.7: Activity Diagram - Create Workspace</b>	32
<b>Figure 3.8: Activity Diagram - Project List</b>	33
<b>Figure 3.9: Activity Diagram - kanban Board</b>	34
<b>Figure 4.1: Sequence Diagram - Registration</b>	35
<b>Figure 4.2: Sequence Diagram - Login</b>	36
<b>Figure 4.3: Sequence Diagram - Create Workspace</b>	37
<b>Figure 4.4: Sequence Diagram - Workspace List</b>	38
<b>Figure 4.5: Sequence Diagram - Share Workspace</b>	39
<b>Figure 4.6: Sequence Diagram - Create Project</b>	40
<b>Figure 4.7: Sequence Diagram - Project List</b>	41
<b>Figure 4.8: Sequence Diagram - Kanban Board</b>	42
<b>Figure 5.1 : Entity Relationship Diagram</b>	43
<b>Figure 6.1: User Manual - Registration</b>	51
<b>Figure 6.2: User Manual - Login</b>	52
<b>Figure 6.3 : User Manual - Workspace List</b>	53
<b>Figure: 6.4: User Manual-Specific Workspace</b>	53
<b>Figure 6.5:User Manual - Create Workspace</b>	54

**Figure 6.6: User Manual - Specific Project**

54

**Figure 6.7: User Manual - Kanban**

55

<b>CHAPTER 1: INTRODUCTION</b>	1
1.1 Project Overview	1
1.2. Project Purpose	1
1.2.1 Background	2
1.2.2 Benefits & Beneficiaries	2
1.3 Stakeholders	2
1.3.1 Owner	2
1.3.2 Member	2
1.4 Modules of SDLC Manager	2
1.4.1 Focused Modules	3
1.5 Project Schedule	3
1.5.1 Gantt Chart	4
1.5.2 Release Plan or Milestone	5
1.6. Glossary	6
1.7 Objectives	7
<b>CHAPTER 2: REQUIREMENTS ENGINEERING</b>	7
2.1 Functional Requirements (FR):	7
2.1.1 Registration	7
2.1.2 Login/Logout	7
2.1.3 Create Workspace	7
2.1.4 View Workspace	8
2.1.5 Create Project	8
2.1.6 Update Sdlc phase	8
2.1.7 Invite Member	8
2.2 Non-Functional Requirements	9
2.2.1 Performance	9
2.2.2 Availability	9
2.2.3 Security	9
<b>CHAPTER 3: SYSTEM ANALYSIS, DESIGN &amp; SPECIFICATION</b>	10
3.1 Development Model	10
3.2 Use Case Diagram	11
3.2 Use Case Description	12
3.2.1 Registration	12
3.2.2 Login	12
3.2.3 User Dashboard	13
3.2.3.1 Create Workspace	13
3.2.3.2 Workspace List	14
3.2.4 Workspace List	14
3.2.4.1 Sharing and Permission	15

3.2.4.2 Specific Workspace	15
3.2.4.3 Remove Workspace	15
3.2.4.4 Rename Workspace	16
3.2.5 Specific Workspace	16
3.2.5.1 Create Project	17
3.2.5.2 Specific Project	17
3.2.5.3 Project List	18
3.2.5.4 Remove Project	18
3.2.5.5 Rename Project	19
3.2.5.6 Sharing and Permission	19
3.2.5.7 Kanban Board	20
3.2.6 Specific Project	20
3.2.6.1 SDLC Phase	21
3.2.7 Kanban Board	21
3.2.7.1 Add	22
3.2.7.2 Remove	22
3.2.7.2 Edit	23
3.2.7.2 Change Status	23
3.4 Activity Diagram	25
3.4.1 Registration	25
3.4.2 Login	26
3.4.3 Create Workspace	27
3.4.4 Workspace List	28
3.4.5 Visit Specific Workspace	29
3.4.6 Workspace Sharing	30
3.4.7 Create Project	31
3.4.8 Project List	32
3.4.9 Kanban Board	33
3.5 Sequence Diagram	34
3.5.1 Registration	34
3.5.2 Login	35
3.5.3 Create Workspace	36
3.5.4 Workspace List	37
3.5.5 Share Workspace	38
3.5.6 Create Project	39
3.5.7 Project List	40
3.5.8 Kanban Board	41
3.6 Entity Relationship Diagram	42
<b>CHAPTER 4: SYSTEM TESTING</b>	<b>43</b>
4.1 Feature Testing	43
4.1.1 Features to be tested	43



4.2 Testing strategies	44
4.2.1 Test Approach	44
4.2.2 Pass/Fail Criteria	44
4.2.3 Testing Schedule	44
4.3 Testing Environment	45
4.4 Test Cases	45
4.4.1 Registration	45
4.4.2 Login	46
4.4.3 Create New Workspace	47
4.4.4 Create New Project	48
4.4.5 Invite Member	49
<b>CHAPTER 5: USER MANUAL</b>	<b>50</b>
5.1 Registration	50
5.2 Login	50
5.3 Workspace	51
5.4 Specific workspace	52
5.5 Create Workspace	52
5.6 Specific Project	53
5.7 Kanban	54
<b>CHAPTER 6: CONCLUSION</b>	<b>54</b>
6.1 Project Summary	55
6.2 Limitations	55
6.3 Obstacles and Achievements	55
6.4. Future Scope	55
<b>CHAPTER 6: REFERENCE</b>	<b>56</b>

## CHAPTER 1: INTRODUCTION

### 1.1 Project Overview

The “SDLC Manager” is developed to help developers track their SDLC phases and build softwares that are quality full and also error free. The project is designed in such a way that a user can have their own workspace, under specific workspace users can have multiple projects and choose who can see or access.

### 1.2. Project Purpose

The purpose of this document is to present a detailed description of the SDLC Manager System. It will explain the purpose and features of the system, the interfaces of the system, what the system will do, the constraints under which it must operate and how the system will react to external stimuli. This document is intended for both the stakeholders and the developers of the system and will be proposed to the Regional Historical Society for its approval.

Functionalities provided by “SDLC Manager”

- An online platform to be used by developers
- Easy and user friendly use experience
- User has the control of their workspace, who they wishes to share with
- Kanban Board
- Member Invitation

### **1.2.1 Background**

Based on the trello board I wanted to build a system that would help developers to track their SDLC phases and also new developers can have knowledge about SDLC. Here a user can create a project and have pre-defined sdlc phases and users can make their status true, when that particular sdlc phase is done.

### **1.2.2 Benefits & Beneficiaries**

This system would be helpful for developers and help them to make less error while developing software and make software bug free.

- The system helps developers to follow SDLC phases
- This systems helps developers to make software which have less error

### **1.3 Stakeholders**

A person who is actively involved in this system and is not a developer. According to project management, project stakeholders “a person, group or organization will be influenced or affected by in a decision, activity or outcome of the project”, I have an Owner user and Member user as the stakeholder of this system.

#### **1.3.1 Owner**

Owner is the user who has all the access to a workspace, like inviting a member. He can remove a user or add a member to a workspace.

#### **1.3.2 Member**

Member is the user who has only the access of a workspace to view and update project status.

### **1.4 Modules of SDLC Manager**

- Workspace Module: Used for managing the workspace of a user

- Project Module: Used for managing project of specific workspace
- Kanban module: Used for managing the kanban board of workspace of a project
- Registration Module: Used for managing the users of the system
- Login Module: Used for used authentication

#### **1.4.1 Focused Modules**

- Registration -

At first, a user has to complete registration providing username, email and password, which will be saved inside the database of the system.

- Login -

After providing correct credentials, users will be redirected to the workspace page, where they can create a new workspace, project or invite members.

- Workspace -

Inside workspace, a user can create a new project, add kanban boards or invite members.

- Project -

In the project, a user can change the status of sdlc phase.

- Kanban -

Inside kanban, users can add, modify or delete board titles or contents.

#### **1.5 Project Schedule**

Considering a short time, I need to read a scheduling plan to finish the project on time. It also refers to making communication with what task needs to be done within a short time.

### 1.5.1 Gantt Chart

Gantt chart is a production time control tool. It remained me to complete my assigned tasks within a certain period. For developing software, it is mostly used. I have designed a Gantt chart for my project.

Activities		W 1	W 2	W 3	W 4	W 5	W 6	W 7	W 8	W 9	W 10	W 11	W 12	W 13	W 14	W 15	W 16
Planning	Ideas																
	Problem identification																
	Proposal Planning																
Requirements	Requirements Specification																
	Requirement Analysis																
QA-1	Quality assurance																
System Design	Sketching																
	Design Specification																
	Database Design																
Implementation-1	Write Review																

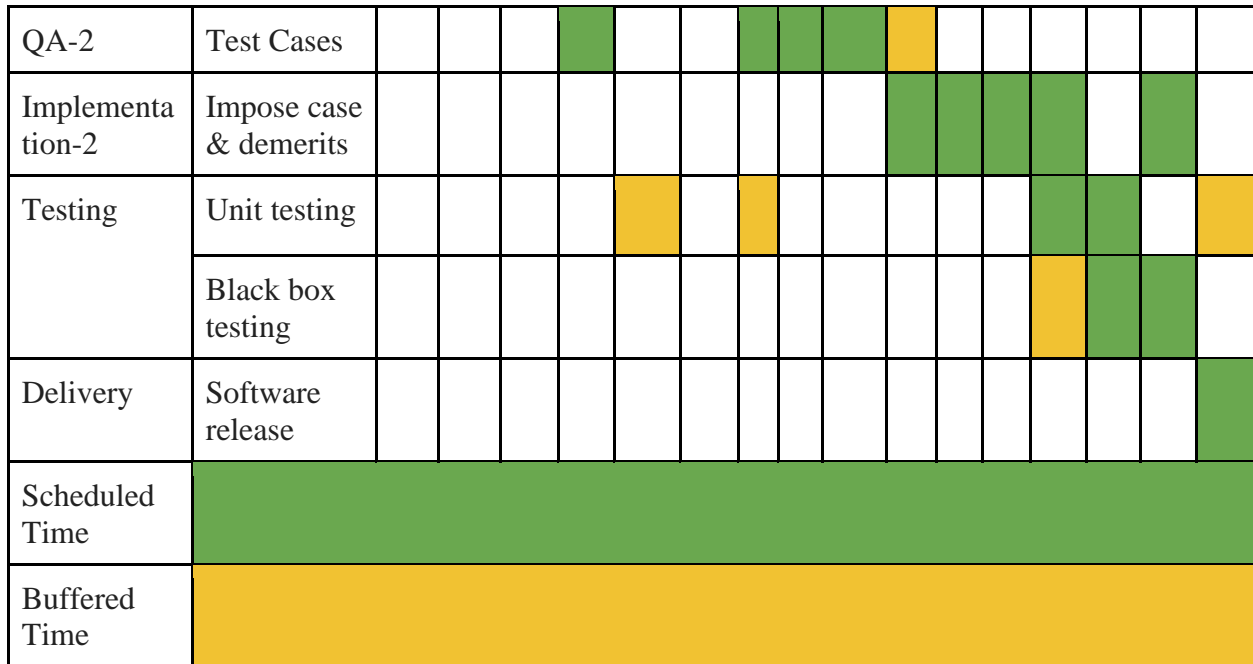


Figure 1.1: Gantt Chart

### 1.5.2 Release Plan or Milestone

The release plan or milestones are given below

Activities	Duration in week	Total Week
Brainstorming	W 1	1
Problem identification	W1, W2	2
Requirement specification	W2	1
Requirement analysis	W2,W3,W4	3
Sketching	W4,W5	2
Design specification	W5	1
Database design	W5,W6	2
Write Review	W5,W5,W6,W7,W8	5

Quality assurance	W3	1
Test case	W3,W6,W7,W8,W9	5
Impose case & demerits	W9.W10,W11,W12,W14	5
Unit testing	W3,W6,W7,W8,W9	5
Black-box testing	W13,W14,W15	3
Software release	W16	1

Table 1.2: Release Plan

## 1.6. Glossary

Term	Definition
SDLC	SDLC or Software Development Life Cycle phases are the standard process, which a software team must follow while developing a software to meet maximum output.
SRS(Software Requirement Specification)	A document that completely describes all of the functions of a proposed system and the constraints under which it must operate. For example, this document.
Stakeholder	Any person with an interest in the project who is not a developer.
End-User	Intended persons for whom the software is built.

## 1.7 Objectives

The main objective behind the project is to provide such an application, that will help software developers to follow SDLC phase and make a software that has less bugs.

## CHAPTER 2: REQUIREMENTS ENGINEERING

### 2.1 Functional Requirements (FR):

Functional requirements referred to as the mandatory functions, a software must have. Functional requirements capture the intended behavior of the system. This behavior can be written as functions, services, tasks, or which system is required to perform.

#### 2.1.1 Registration

FR 1	Registration
Description	End-users need registration for login and access to functionalities of the system.
Stakeholders	End-users

#### 2.1.2 Login/Logout

FR 2	Login/Logout
Description	End-users need registration for login and access to functionalities of the system.
Stakeholders	End-users

#### 2.1.3 Create Workspace

FR 3	Create workspace
Description	End-users can create workspace for specific works



Stakeholders	End-users
--------------	-----------

#### 2.1.4 View Workspace

FR 4	View workspace
Description	When a workspace is created, a user can see all information about that workspace
Stakeholders	End-users

#### 2.1.5 Create Project

FR 5	Create Project
Description	A user can create project and work on that
Stakeholders	End-users

#### 2.1.6 Update Sdlc phase

FR 6	Update Sdlc phase
Description	When a project is created, it includes sdlc phases which users can update the status.
Stakeholders	End-users

#### 2.1.7 Invite Member

FR 7	Invite Member
Description	A user can invite a member to work collaboratively.
Stakeholders	End-users

## 2.2 Non-Functional Requirements

### 2.2.1 Performance

NFR 1	The system will provide all services without any fault
Description	When a user tries to perform particular functionality, then the outcome must appear without any fault and faster.
Stakeholders	End-users

### 2.2.2 Availability

NFR 2	The system must be available all the time
Description	The system should be available to the user all the time
Stakeholders	End-users

### 2.2.3 Security

NFR 3	The system must be secured
Description	The system must be secured and use https, so user form credentials might not be in trouble and also database must be secured
Stakeholders	End-users

## CHAPTER 3: SYSTEM ANALYSIS, DESIGN & SPECIFICATION

### 3.1 Development Model

As my project requires many tests and reviews, I choose an agile model to complete my project. The Agile model offers more flexibility to build a system, where we can change the system easily without any difficulties. The Agile model is more flexible if a system can be changed often.



Figure 2.1: Agile Model

### 3.2 Use Case Diagram



Figure 2.2 : Use Case Diagram

## 3.2 Use Case Description

### 3.2.1 Registration

<b>Use Case Name</b>	User Registration
<b>Trigger</b>	User Click on registration after entering website
<b>Pre Condition</b>	<ol style="list-style-type: none"> <li>1. Have to be at website</li> <li>2. Must be stable internet connectivity</li> </ol>
<b>Basic Path</b>	<ol style="list-style-type: none"> <li>1. User goes to website</li> <li>2. Clicks on register</li> <li>3. Give credentials</li> <li>4. Clicks on register button</li> </ol>
<b>Alternative Path</b>	In step 3 user can register with OAuth(Open Authentication) with gmail
<b>Post Condition</b>	<ol style="list-style-type: none"> <li>1. User will have a success message</li> <li>2. User will be redirected to login page</li> </ol>
<b>Exception Path</b>	<ol style="list-style-type: none"> <li>1. User can cancel registration at any time</li> </ol>
<b>Other</b>	

### 3.2.2 Login

<b>Use Case Name</b>	User Login
<b>Trigger</b>	User Click on login after entering website
<b>Pre Condition</b>	<ol style="list-style-type: none"> <li>1. Have to be at website</li> <li>2. Must be stable internet connectivity</li> </ol>
<b>Basic Path</b>	<ol style="list-style-type: none"> <li>1. Visits WEbsite</li> <li>2. Click on Login</li> <li>3. Give credentials</li> <li>4. Clicks login</li> </ol>
<b>Alternative Path</b>	In step 3 user can login with OAuth(Open Authentication) with gmail
<b>Post Condition</b>	<ol style="list-style-type: none"> <li>1. User will be redirected to dashboard page</li> </ol>
<b>Exception Path</b>	
<b>Other</b>	

### 3.2.3 User Dashboard

<b>Use Case Name</b>	User Dashboard
<b>Trigger</b>	User will be redirected after successfully login
<b>Pre Condition</b>	<ol style="list-style-type: none"> <li>1. Must have stable internet connection</li> <li>2. Must be logged in</li> </ol>
<b>Basic Path</b>	<ol style="list-style-type: none"> <li>1. Visits Website</li> <li>2. Click on Login</li> <li>3. Give credentials</li> <li>4. Clicks login</li> <li>5. Redirect to user dashboard</li> </ol>
<b>Alternative Path</b>	In step 5 user can be redirected to login page if credentials are wrong
<b>Post Condition</b>	<ol style="list-style-type: none"> <li>1. User must create workspace</li> <li>2. User must check workspace list</li> </ol>
<b>Exception Path</b>	
<b>Other</b>	

#### 3.2.3.1 Create Workspace

<b>Use Case Name</b>	Create Workspace
<b>Trigger</b>	Click on create workspace
<b>Pre Condition</b>	<ol style="list-style-type: none"> <li>1. Must have stable internet connection</li> <li>2. Must be logged in</li> <li>3. Must be at user dashboard page</li> </ol>
<b>Basic Path</b>	<ol style="list-style-type: none"> <li>1. Clicks on create workspace</li> <li>2. Gives workspace name input</li> <li>3. Gives sharing option</li> </ol>
<b>Alternative Path</b>	In step 3 user can skip

<b>Post Condition</b>	<ol style="list-style-type: none"> <li>1. Must see success message</li> <li>2. Must see posted workspace name</li> </ol>
<b>Exception Path</b>	
<b>Other</b>	

### 3.2.3.2 Workspace List

<b>Use Case Name</b>	Workspace List
<b>Trigger</b>	Click on workspace list
<b>Pre Condition</b>	<ol style="list-style-type: none"> <li>1. Must have stable internet connection</li> <li>2. Must be logged in</li> <li>3. Must be at user dashboard page</li> </ol>
<b>Basic Path</b>	<ol style="list-style-type: none"> <li>1. Click on workspace list</li> </ol>
<b>Alternative Path</b>	
<b>Post Condition</b>	<ol style="list-style-type: none"> <li>1. Must redirect to workspace list page</li> </ol>
<b>Exception Path</b>	
<b>Other</b>	

### 3.2.4 Workspace List

<b>Use Case Name</b>	Workspace List
<b>Trigger</b>	Click on workspace list
<b>Pre Condition</b>	<ol style="list-style-type: none"> <li>1. Must have stable internet connection</li> <li>2. Must be logged in</li> <li>3. Must be at user dashboard page</li> </ol>
<b>Basic Path</b>	<ol style="list-style-type: none"> <li>1. Click on workspace list</li> </ol>
<b>Alternative Path</b>	
<b>Post Condition</b>	<ol style="list-style-type: none"> <li>1. Must redirect to workspace list page</li> </ol>
<b>Exception Path</b>	
<b>Other</b>	

### 3.2.4.1 Sharing and Permission

<b>Use Case Name</b>	Sharing and permission
<b>Trigger</b>	1. Click on sharing and permission
<b>Pre Condition</b>	1. Must have stable internet connection 2. Must be logged in 3. Must be at workspace list page
<b>Basic Path</b>	1. Sharing and permission
<b>Alternative Path</b>	
<b>Post Condition</b>	1. Must redirect to workspace list page
<b>Exception Path</b>	
<b>Other</b>	

### 3.2.4.2 Specific Workspace

<b>Use Case Name</b>	Specific Workspace
<b>Trigger</b>	1. Click on sharing and permission
<b>Pre Condition</b>	1. Must have stable internet connection 2. Must be logged in 3. Must be at workspace list page
<b>Basic Path</b>	1. Sharing and permission
<b>Alternative Path</b>	
<b>Post Condition</b>	1. Must redirect to workspace list page
<b>Exception Path</b>	
<b>Other</b>	

### 3.2.4.3 Remove Workspace

<b>Use Case Name</b>	Remove Workspace
<b>Trigger</b>	



<b>Pre Condition</b>	<ol style="list-style-type: none"> <li>1. Must have stable internet connection</li> <li>2. Must be logged in</li> <li>3. Must be at workspace list page</li> </ol>
<b>Basic Path</b>	<ol style="list-style-type: none"> <li>1. Click remove</li> <li>2. Press confirm</li> </ol>
<b>Alternative Path</b>	
<b>Post Condition</b>	<ol style="list-style-type: none"> <li>1. Must have successful message</li> </ol>
<b>Exception Path</b>	
<b>Other</b>	

#### 3.2.4.4 Rename Workspace

<b>Use Case Name</b>	Rename Workspace
<b>Trigger</b>	
<b>Pre Condition</b>	<ol style="list-style-type: none"> <li>1. Must have stable internet connection</li> <li>2. Must be logged in</li> <li>3. Must be at workspace list page</li> </ol>
<b>Basic Path</b>	<ol style="list-style-type: none"> <li>1. Click rename</li> <li>2. Give new name input</li> <li>3. Click save</li> </ol>
<b>Alternative Path</b>	
<b>Post Condition</b>	<ol style="list-style-type: none"> <li>1. Must have successful message</li> </ol>
<b>Exception Path</b>	
<b>Other</b>	

#### 3.2.5 Specific Workspace

<b>Use Case Name</b>	Specific Workspace
<b>Trigger</b>	
<b>Pre Condition</b>	<ol style="list-style-type: none"> <li>1. Must have stable internet connection</li> <li>2. Must be logged in</li> <li>3. Must be at workspace list page</li> </ol>
<b>Basic Path</b>	<ol style="list-style-type: none"> <li>1. Click on specific workspace</li> </ol>

<b>Alternative Path</b>	
<b>Post Condition</b>	1. Must have specific workspace features access
<b>Exception Path</b>	
<b>Other</b>	

### 3.2.5.1 Create Project

<b>Use Case Name</b>	Create Project
<b>Trigger</b>	1. Click on create project at specific workspace page
<b>Pre Condition</b>	1. Must have stable internet connection 2. Must be logged in 3. Must be at specific workspace page
<b>Basic Path</b>	1. Click on create project 2. Give project name input 3. Give sharing and permission 4. Comment Input
<b>Alternative Path</b>	At step 3 and 4 user can have alternative path 1. Skip sharing and permission 2. Skip comment
<b>Post Condition</b>	1. Should get popup success message
<b>Exception Path</b>	
<b>Other</b>	

### 3.2.5.2 Specific Project

<b>Use Case Name</b>	Specific Project
<b>Trigger</b>	1. Click on specific project at specific workspace page
<b>Pre Condition</b>	1. Must have stable internet connection 2. Must be logged in 3. Must be at specific workspace page

<b>Basic Path</b>	1. Click on specific project
<b>Alternative Path</b>	
<b>Post Condition</b>	1. Should redirect to specific project page
<b>Exception Path</b>	
<b>Other</b>	

### 3.2.5.3 Project List

<b>Use Case Name</b>	Project List
<b>Trigger</b>	1. Click on project list at specific workspace page
<b>Pre Condition</b>	1. Must have stable internet connection 2. Must be logged in 3. Must be at specific workspace page
<b>Basic Path</b>	1. Click on project list
<b>Alternative Path</b>	
<b>Post Condition</b>	1. Should be redirected to project list page
<b>Exception Path</b>	
<b>Other</b>	

### 3.2.5.4 Remove Project

<b>Use Case Name</b>	Remove Project
<b>Trigger</b>	1. Click on remove beside specific project
<b>Pre Condition</b>	1. Must have stable internet connection 2. Must be logged in 3. Must be at specific workspace page
<b>Basic Path</b>	1. Click on remove icon beside project 2. Press Confirm
<b>Alternative Path</b>	
<b>Post Condition</b>	1. Should get popup success message

	2. Project list must be updated
<b>Exception Path</b>	
<b>Other</b>	

### 3.2.5.5 Rename Project

<b>Use Case Name</b>	Rename Project
<b>Trigger</b>	1. Click on rename beside specific project
<b>Pre Condition</b>	1. Must have stable internet connection 2. Must be logged in 3. Must be at specific workspace page
<b>Basic Path</b>	1. Click on rename beside project 2. Give new name input 3. Press save
<b>Alternative Path</b>	
<b>Post Condition</b>	1. Project name should be changed
<b>Exception Path</b>	
<b>Other</b>	

### 3.2.5.6 Sharing and Permission

<b>Use Case Name</b>	Sharing and Permission
<b>Trigger</b>	1. Click on sharing and permission beside specific project
<b>Pre Condition</b>	1. Must have stable internet connection 2. Must be logged in 3. Must be at specific workspace page
<b>Basic Path</b>	1. Click on sharing and permission 2. Give user email id 3. Press share
<b>Alternative Path</b>	

<b>Post Condition</b>	1. Success message should pop up
<b>Exception Path</b>	
<b>Other</b>	

### 3.2.5.7 Kanban Board

<b>Use Case Name</b>	Kanban Board
<b>Trigger</b>	1. Click on board at top navbar
<b>Pre Condition</b>	1. Must have stable internet connection 2. Must be logged in 3. Must be at specific workspace page
<b>Basic Path</b>	1. Click on board
<b>Alternative Path</b>	
<b>Post Condition</b>	1. Should redirect to kanban board page
<b>Exception Path</b>	
<b>Other</b>	

### 3.2.6 Specific Project

<b>Use Case Name</b>	Specific Project
<b>Trigger</b>	1. Click on specific project
<b>Pre Condition</b>	1. Must have stable internet connection 2. Must be logged in 3. Must be at specific workspace page
<b>Basic Path</b>	1. Click on specific project
<b>Alternative Path</b>	
<b>Post Condition</b>	1. Should redirect to specific project page
<b>Exception Path</b>	

<b>Other</b>	
--------------	--

### 3.2.6.1 SDLC Phase

<b>Use Case Name</b>	SDLC Phase
<b>Trigger</b>	1. Click on SDLC Phase
<b>Pre Condition</b>	1. Must have stable internet connection 2. Must be logged in 3. Must be at specific project page
<b>Basic Path</b>	1. Click on SDLC Phase 2. Change SDLC Phase Status 3. Give Comment input on specific SDLC phase
<b>Alternative Path</b>	Step 3 can be skipped
<b>Post Condition</b>	1. SDLC phase status should change 2. SDLC phase comment should be visible
<b>Exception Path</b>	
<b>Other</b>	

### 3.2.7 Kanban Board

<b>Use Case Name</b>	Kanban Board
<b>Trigger</b>	1. Click on Board at specific workspace or specific project
<b>Pre Condition</b>	1. Must have stable internet connection 2. Must be logged in 3. Must be at specific project page or at specific workspace page
<b>Basic Path</b>	1. Click on board at specific project or at specific workspace
<b>Alternative Path</b>	
<b>Post Condition</b>	1. Should redirect to board page
<b>Exception Path</b>	

<b>Other</b>	
--------------	--

### 3.2.7.1 Add

<b>Use Case Name</b>	Add
<b>Trigger</b>	1. Click on add button
<b>Pre Condition</b>	1. Must have stable internet connection 2. Must be logged in 3. Must be at board page
<b>Basic Path</b>	1. Click add button 2. Select board 3. Give content input 4. Save
<b>Alternative Path</b>	At step 2 there can be alternative path 1. Click create board 2. Give board name input 3. Press save
<b>Post Condition</b>	1. Should have success popup message
<b>Exception Path</b>	
<b>Other</b>	

### 3.2.7.2 Remove

<b>Use Case Name</b>	Add
<b>Trigger</b>	1. Click on remove icon
<b>Pre Condition</b>	1. Must have stable internet connection 2. Must be logged in 3. Must be at board page
<b>Basic Path</b>	1. Click remove icon 2. Press confirm
<b>Alternative Path</b>	

<b>Post Condition</b>	1. Board should be updated
<b>Exception Path</b>	
<b>Other</b>	

### 3.2.7.2 Edit

<b>Use Case Name</b>	Edit
<b>Trigger</b>	1. Click on edit icon
<b>Pre Condition</b>	1. Must have stable internet connection 2. Must be logged in 3. Must be at board page
<b>Basic Path</b>	1. Click edit icon 2. Give new content name 3. Press save
<b>Alternative Path</b>	
<b>Post Condition</b>	1. Board should be updated
<b>Exception Path</b>	
<b>Other</b>	

### 3.2.7.2 Change Status

<b>Use Case Name</b>	Change Status
<b>Trigger</b>	1. Click on change status
<b>Pre Condition</b>	1. Must have stable internet connection 2. Must be logged in 3. Must be at board page
<b>Basic Path</b>	1. Click change status 2. Select new status 3. Press save



<b>Alternative Path</b>	
<b>Post Condition</b>	1. Board should be updated
<b>Exception Path</b>	
<b>Other</b>	

### 3.4 Activity Diagram

I have prepared some activity diagrams according to my SDLC Manager’s use case. These activity diagrams are properly referring to the flow of the individual conditions of my sdlc manager.

#### 3.4.1 Registration

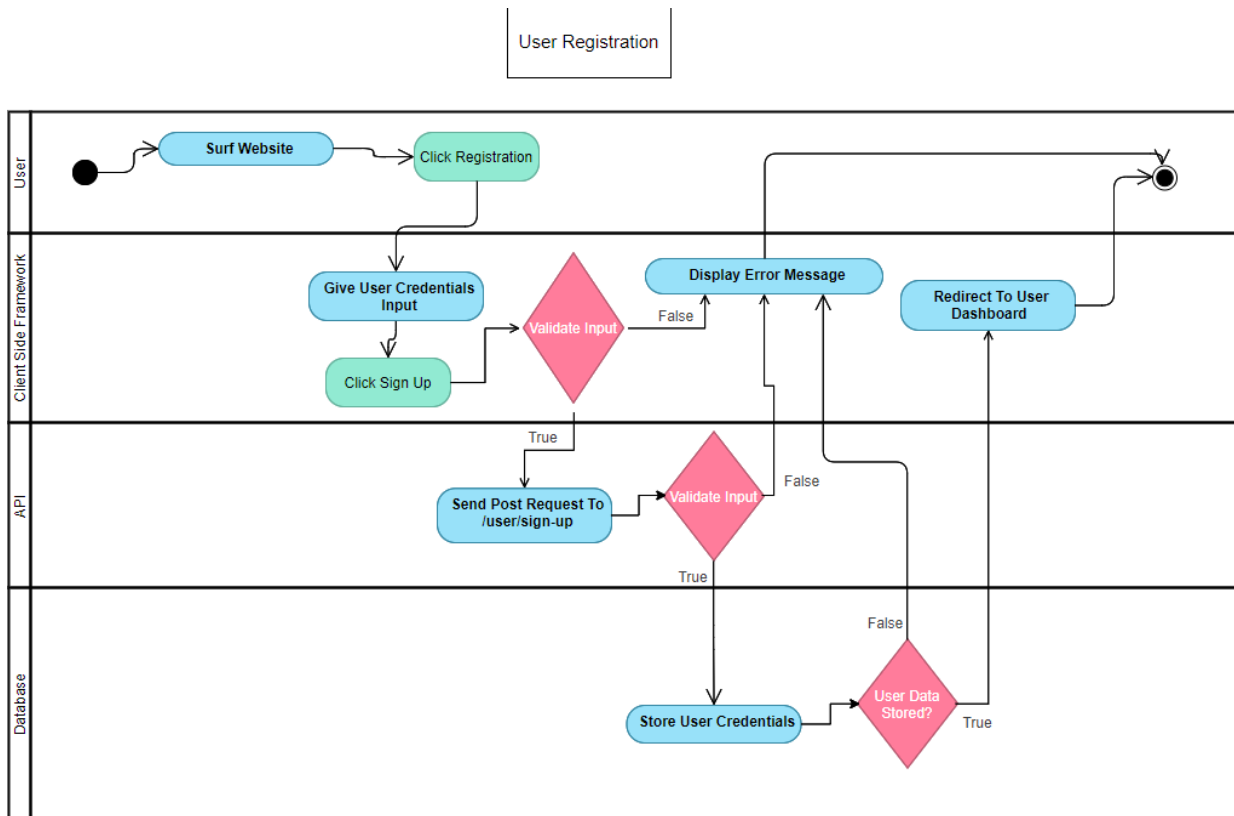


Figure 3.1: Activity Diagram - Registration

### 3.4.2 Login

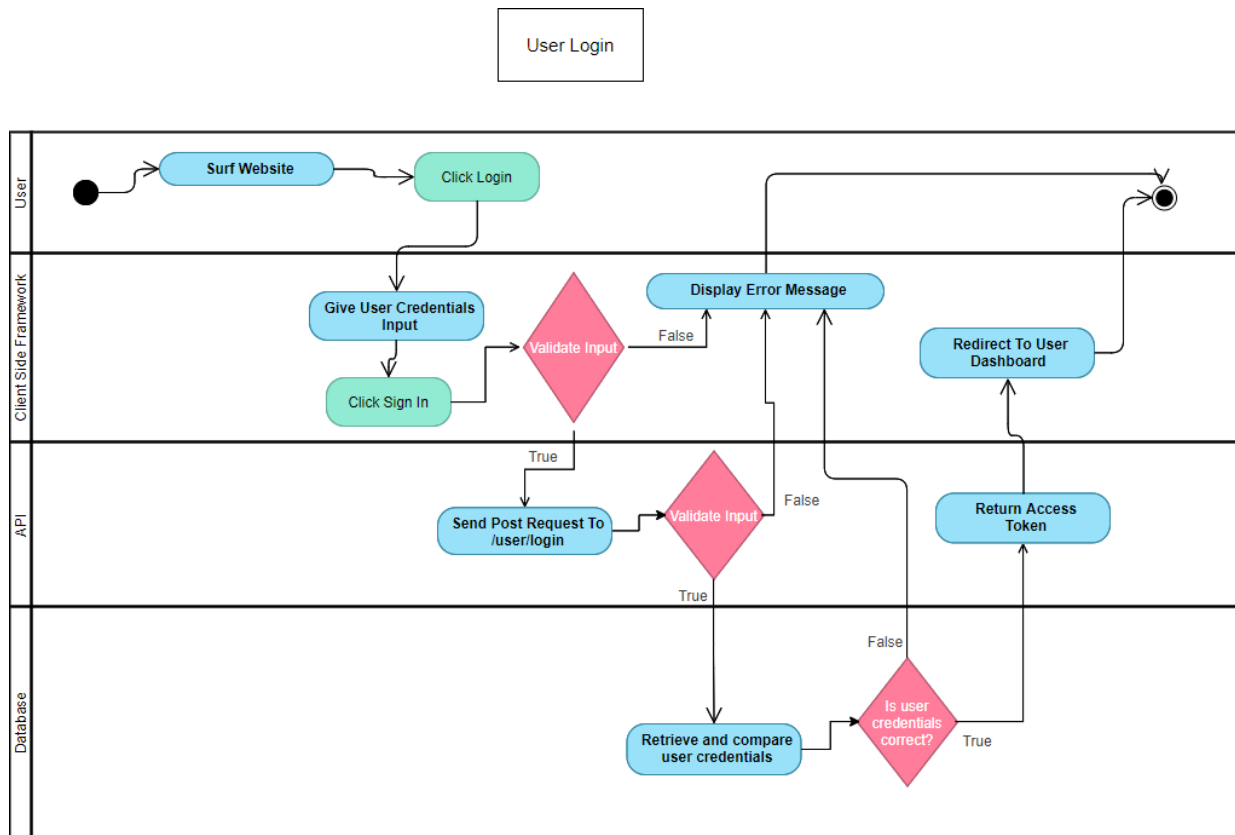


Figure 3.2: Activity Diagram - Login

### 3.4.3 Create Workspace

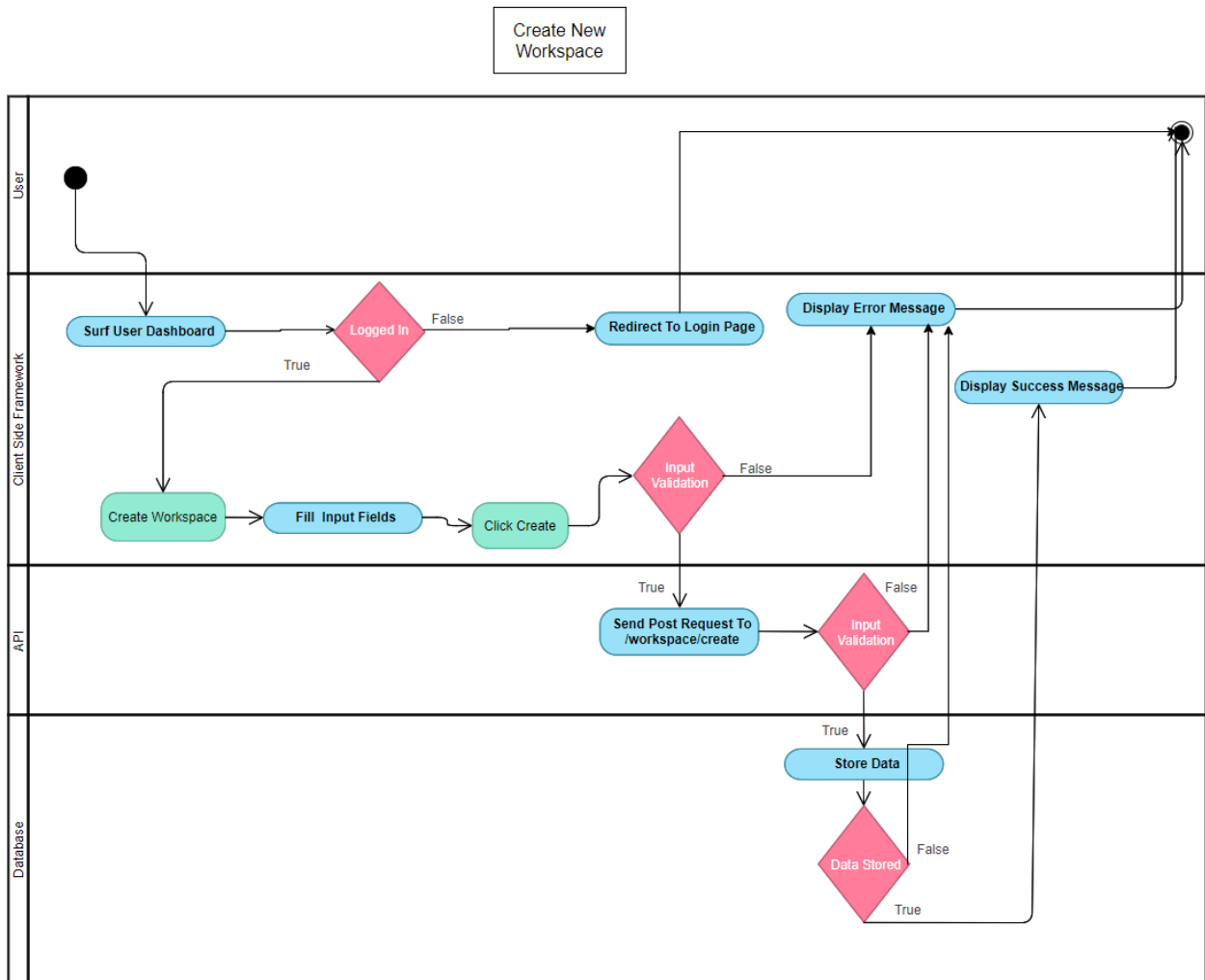


Figure 3.3: Activity Diagram - Create Workspace

### 3.4.4 Workspace List

Visual Paradigm Online Free Edition

Workspace List

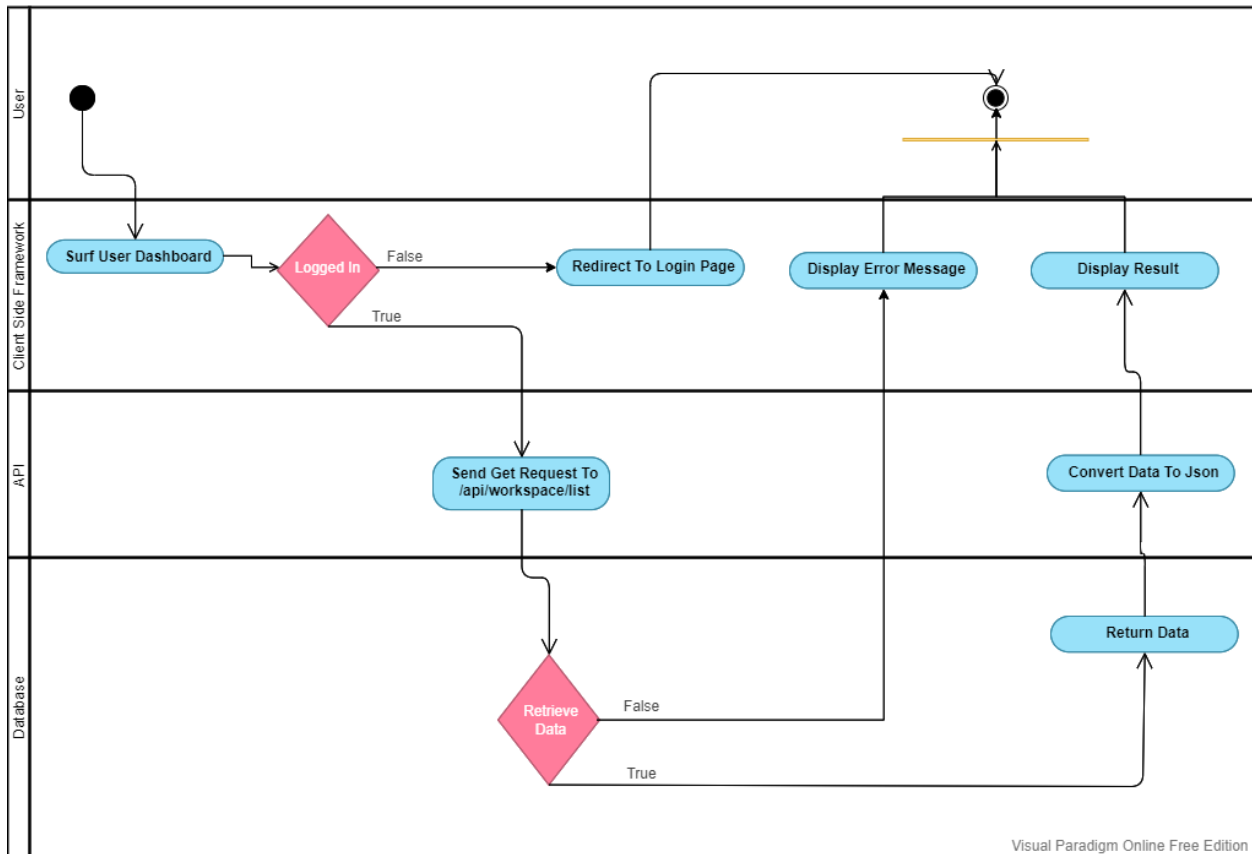
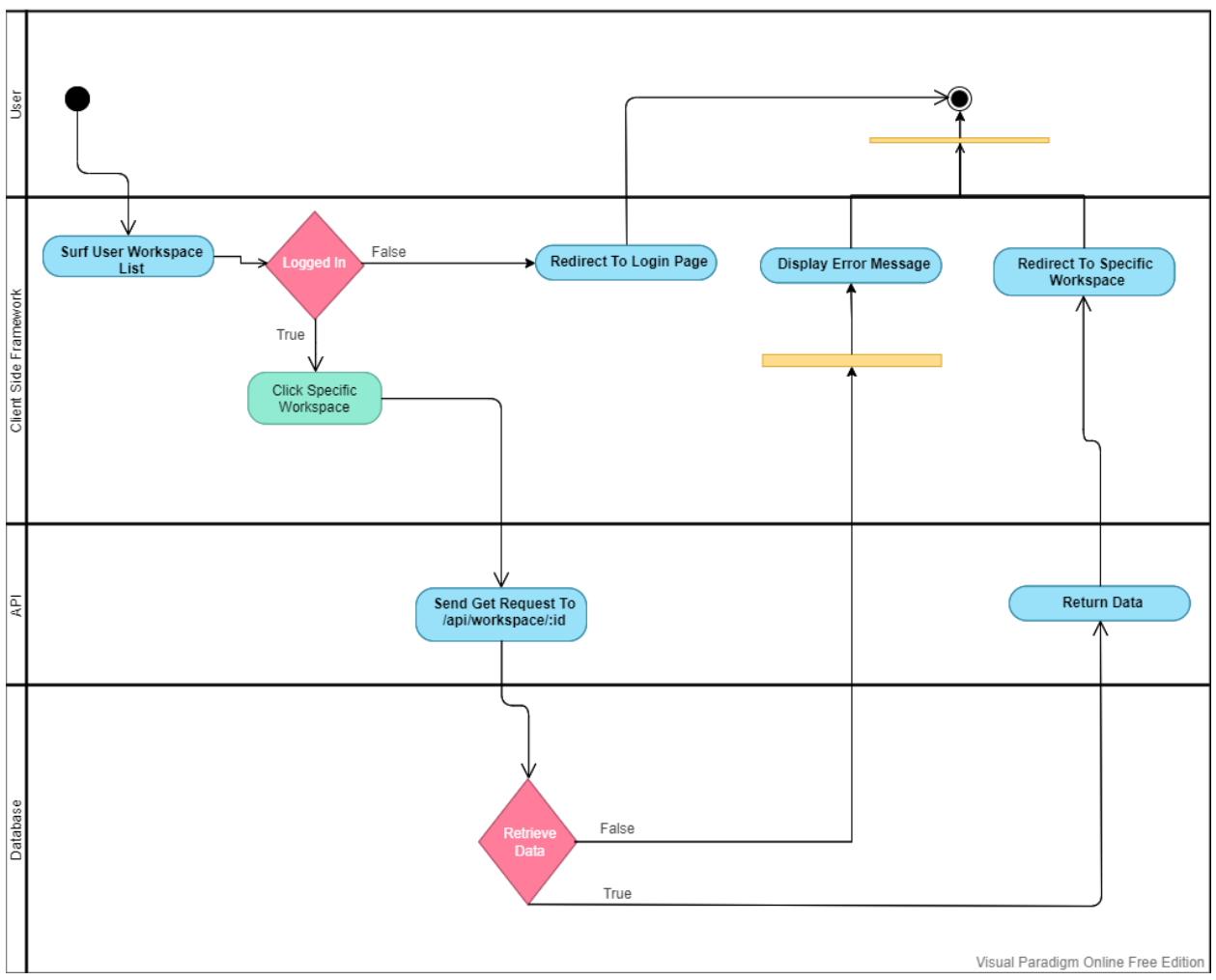


Figure 3.4: Activity Diagram - Registration

### 3.4.5 Visit Specific Workspace

Visual Paradigm Online Free Edition

Visit Particular Workspace



Visual Paradigm Online Free Edition

Figure 3.5: Activity Diagram - Specific Workspace

### 3.4.6 Workspace Sharing

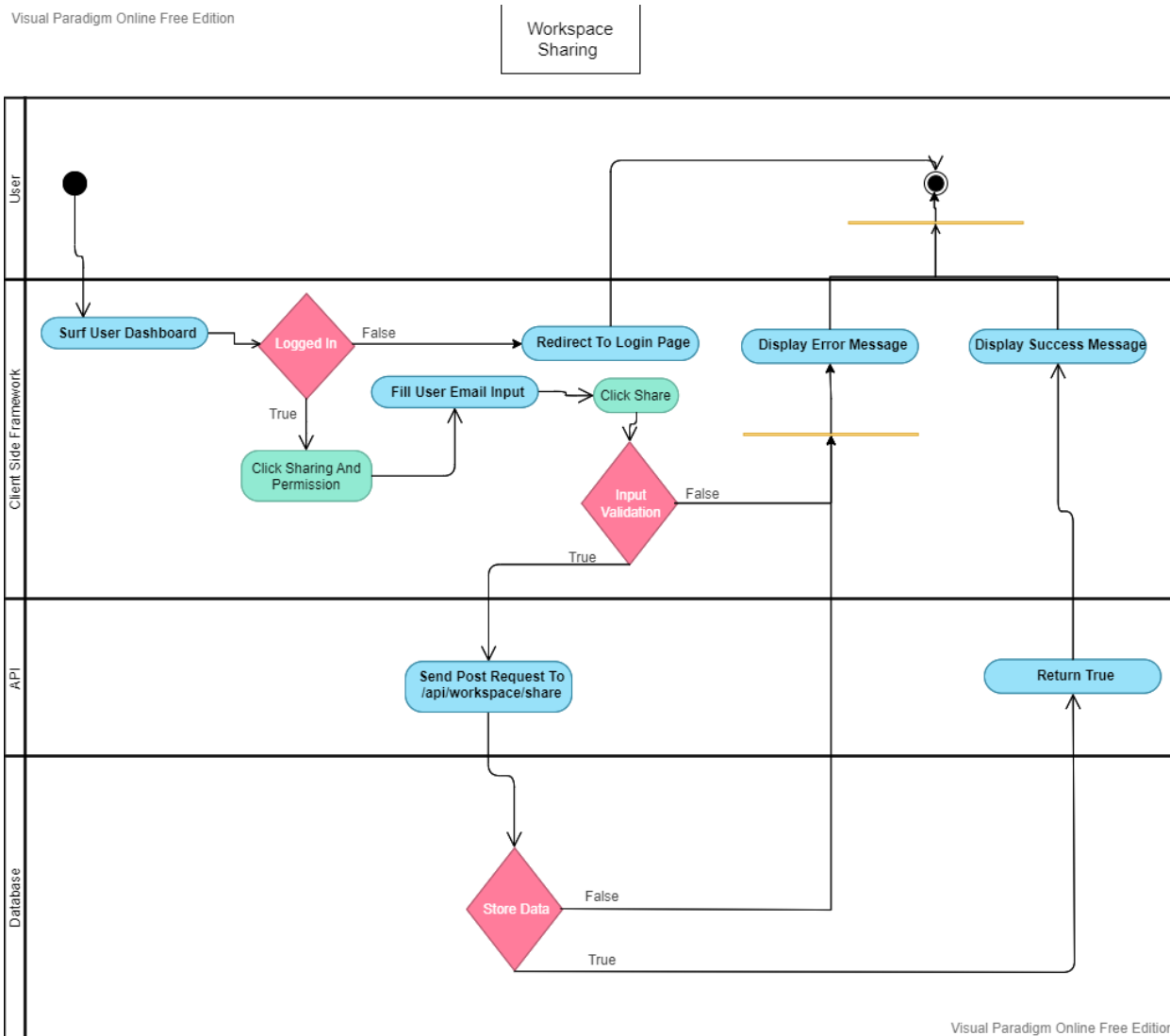


Figure 3.6: Activity Diagram - Workspace Sharing

### 3.4.7 Create Project

Visual Paradigm Online Free Edition

Create Project

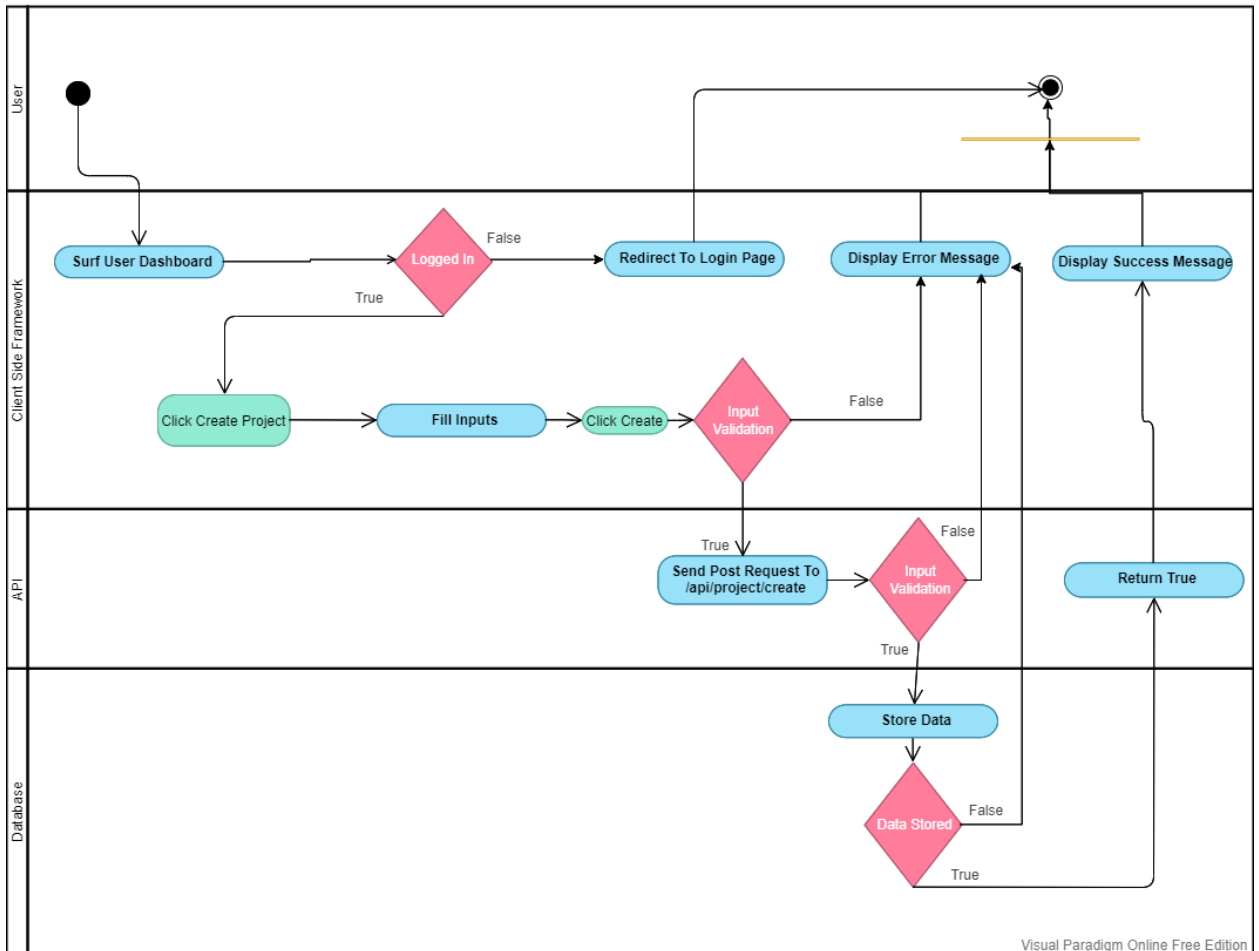


Figure 3.7: Activity Diagram - Create Workspace



### 3.4.8 Project List

Visual Paradigm Online Free Edition

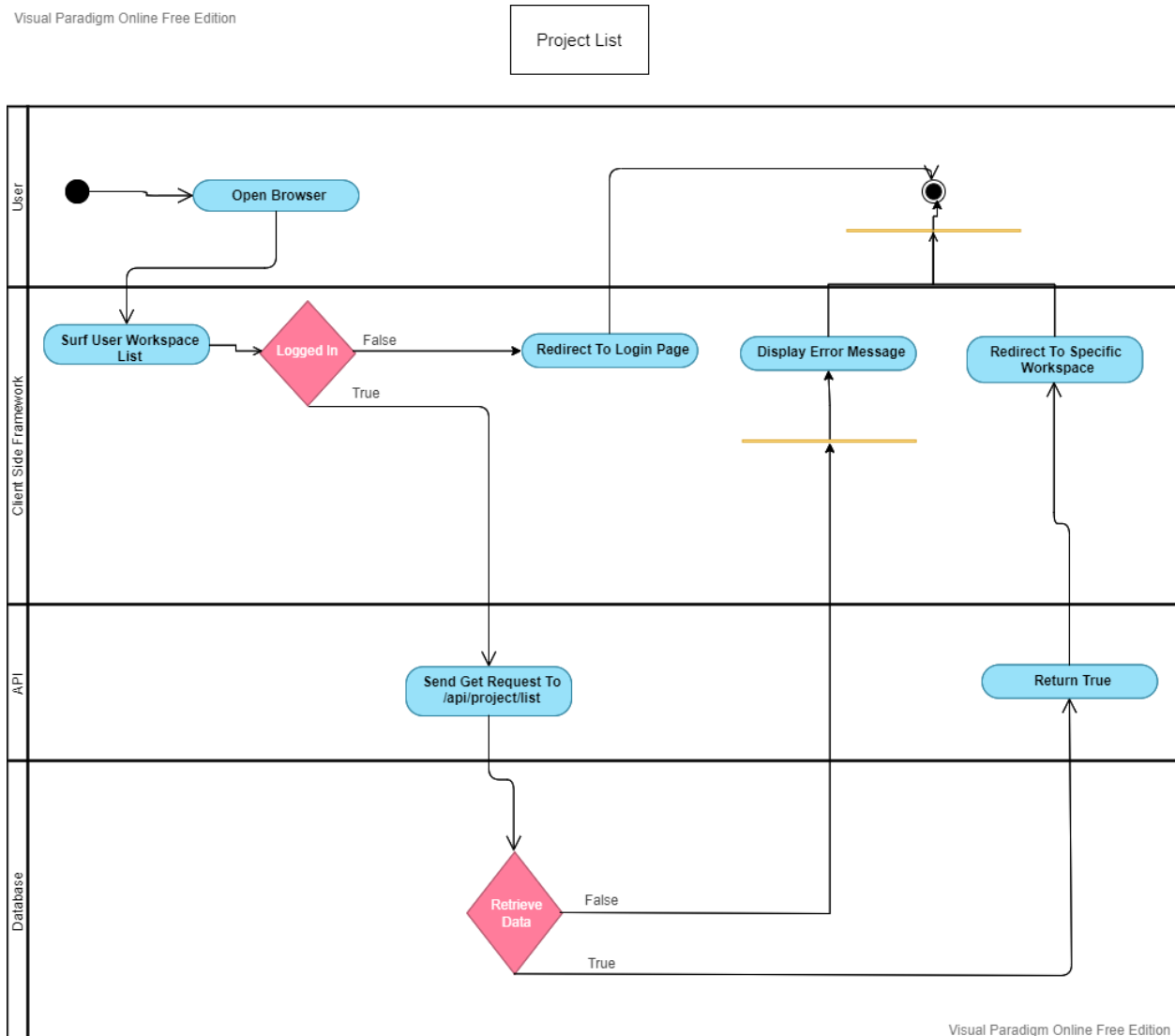


Figure 3.8: Activity Diagram - Project List

### 3.4.9 Kanban Board

Visual Paradigm Online Free Edition

Kanban Board  
List

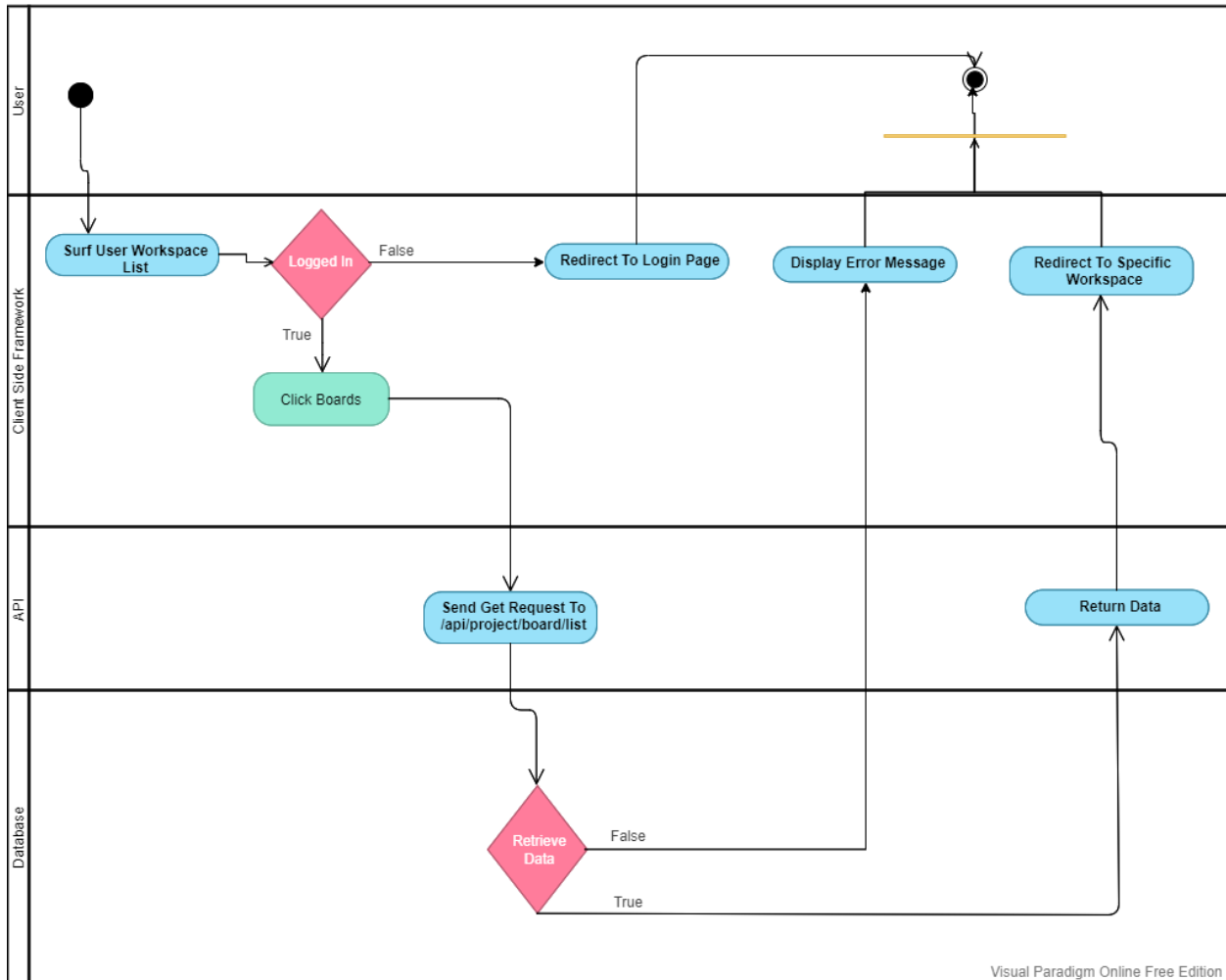


Figure 3.9: Activity Diagram - kanban Board

### 3.5 Sequence Diagram

#### 3.5.1 Registration

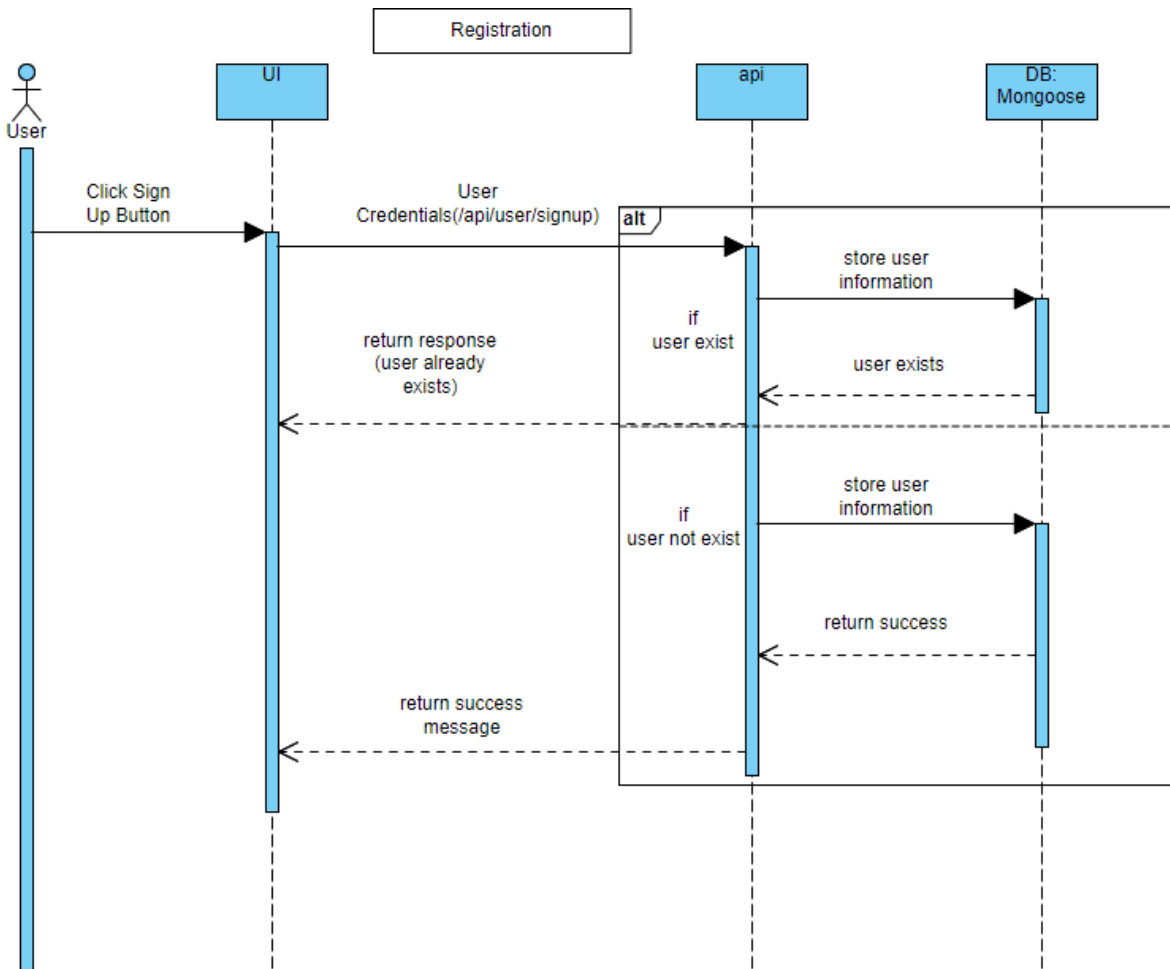


Figure 4.1: Sequence Diagram - Registration

## 3.5.2 Login

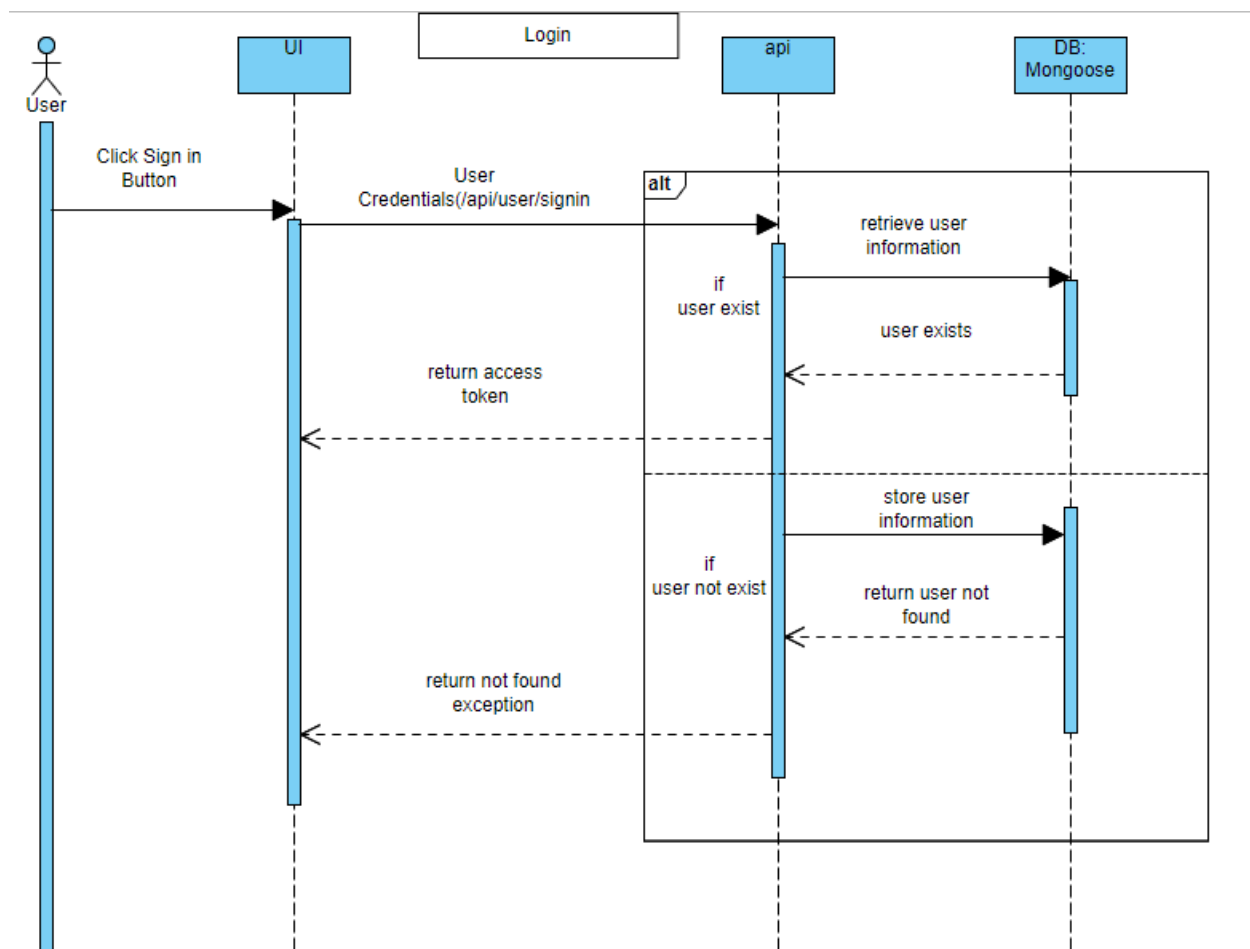


Figure 4.2: Sequence Diagram - Login

### 3.5.3 Create Workspace

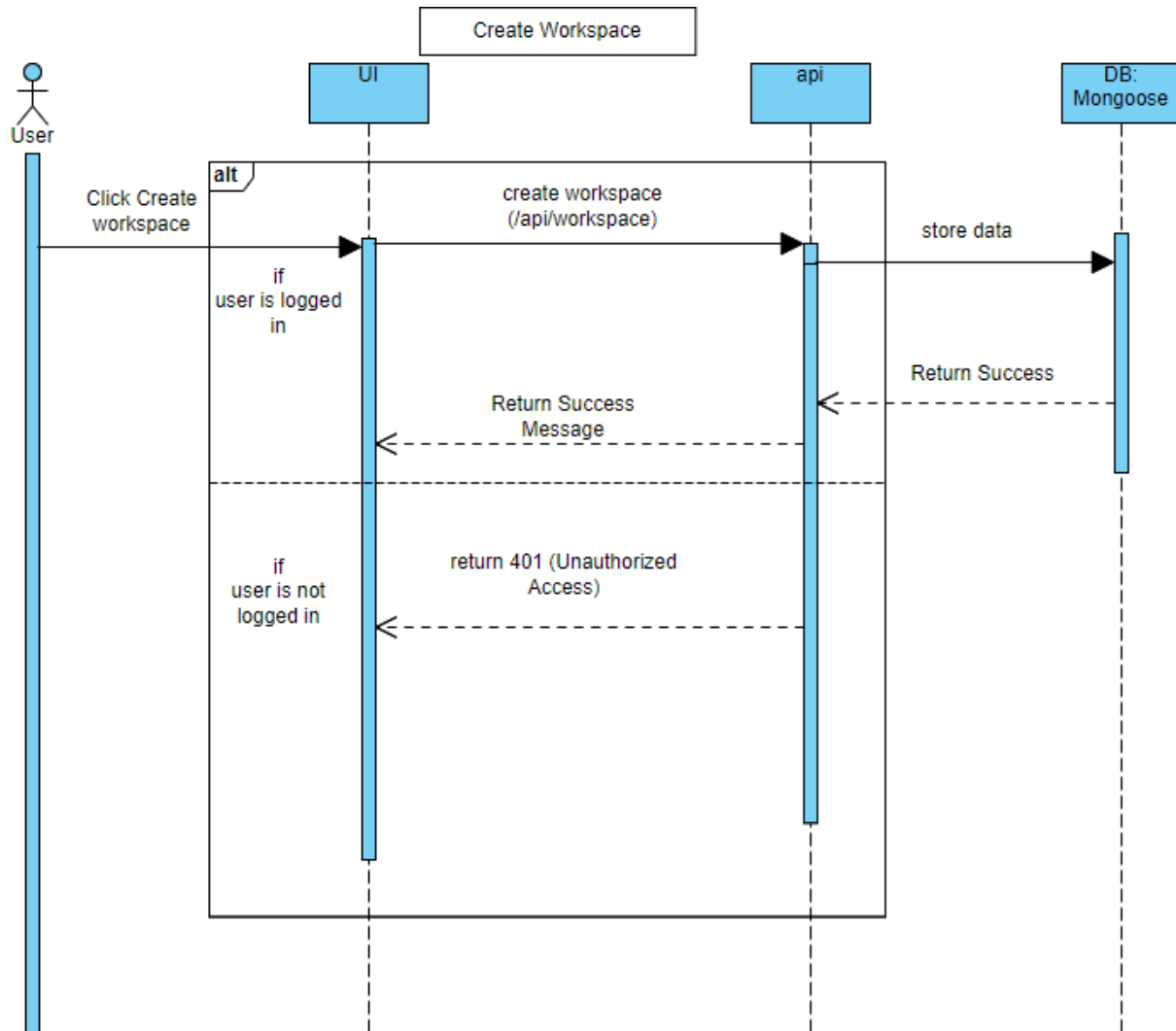


Figure 4.3: Sequence Diagram - Create Workspace

### 3.5.4 Workspace List

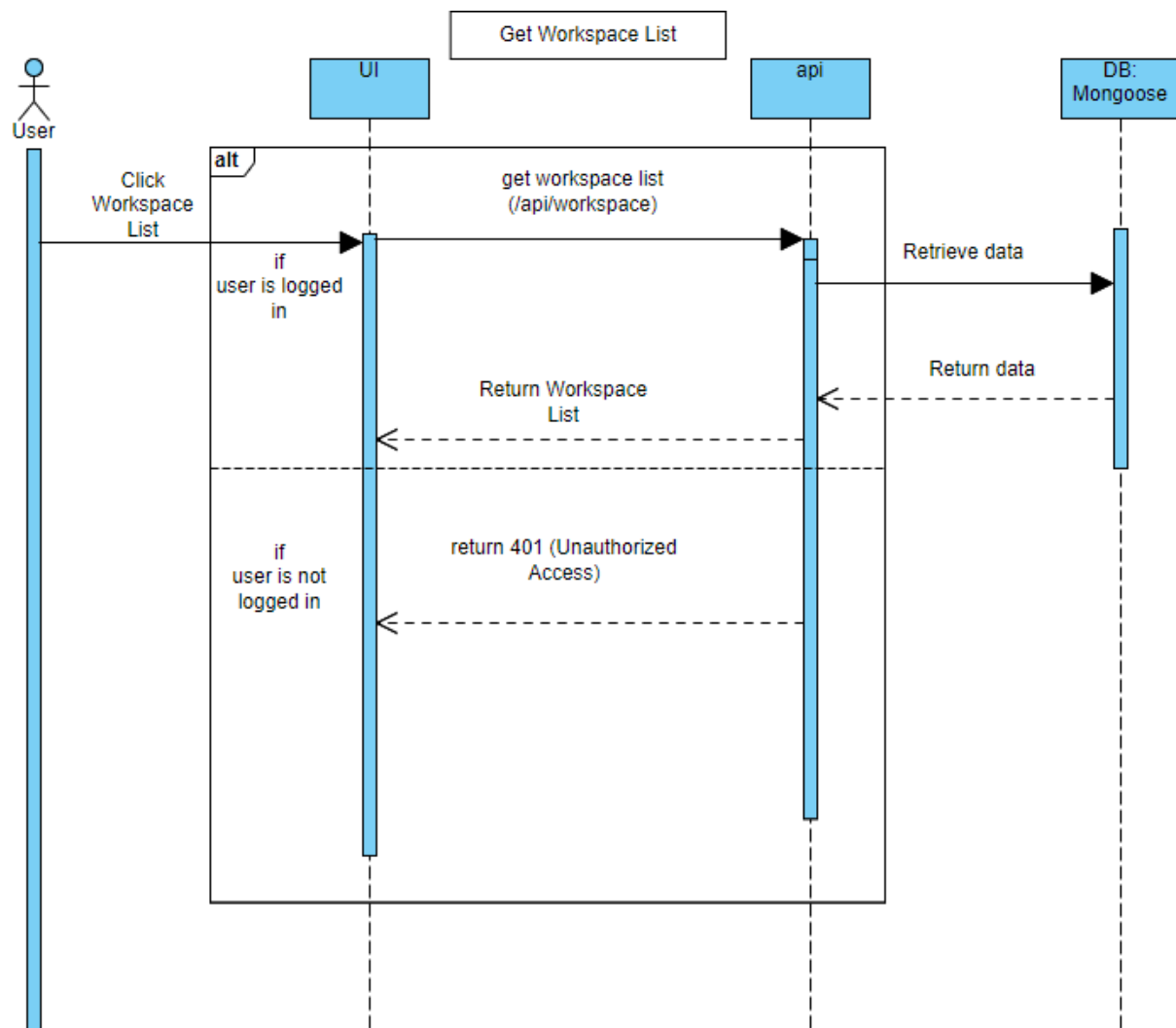


Figure 4.4: Sequence Diagram - Workspace List

### 3.5.5 Share Workspace

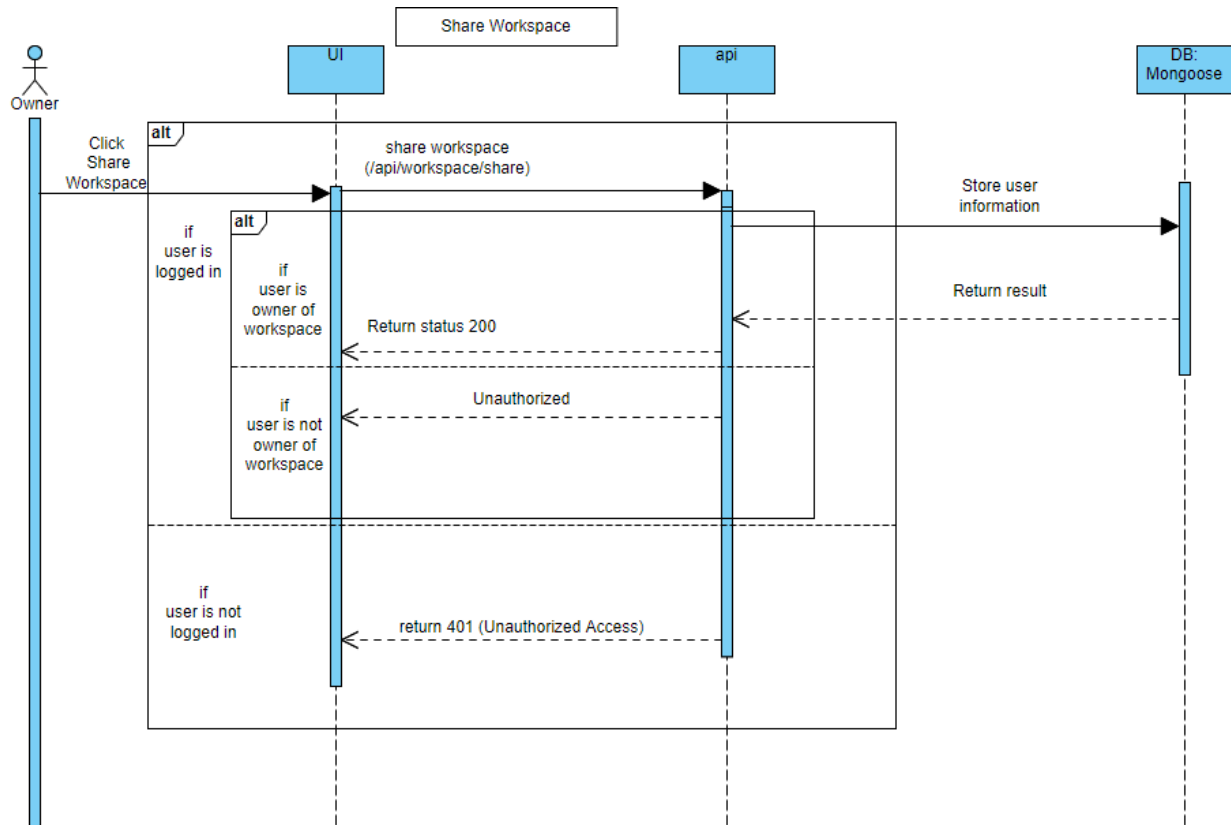


Figure 4.5: Sequence Diagram - Share Workspace

### 3.5.6 Create Project

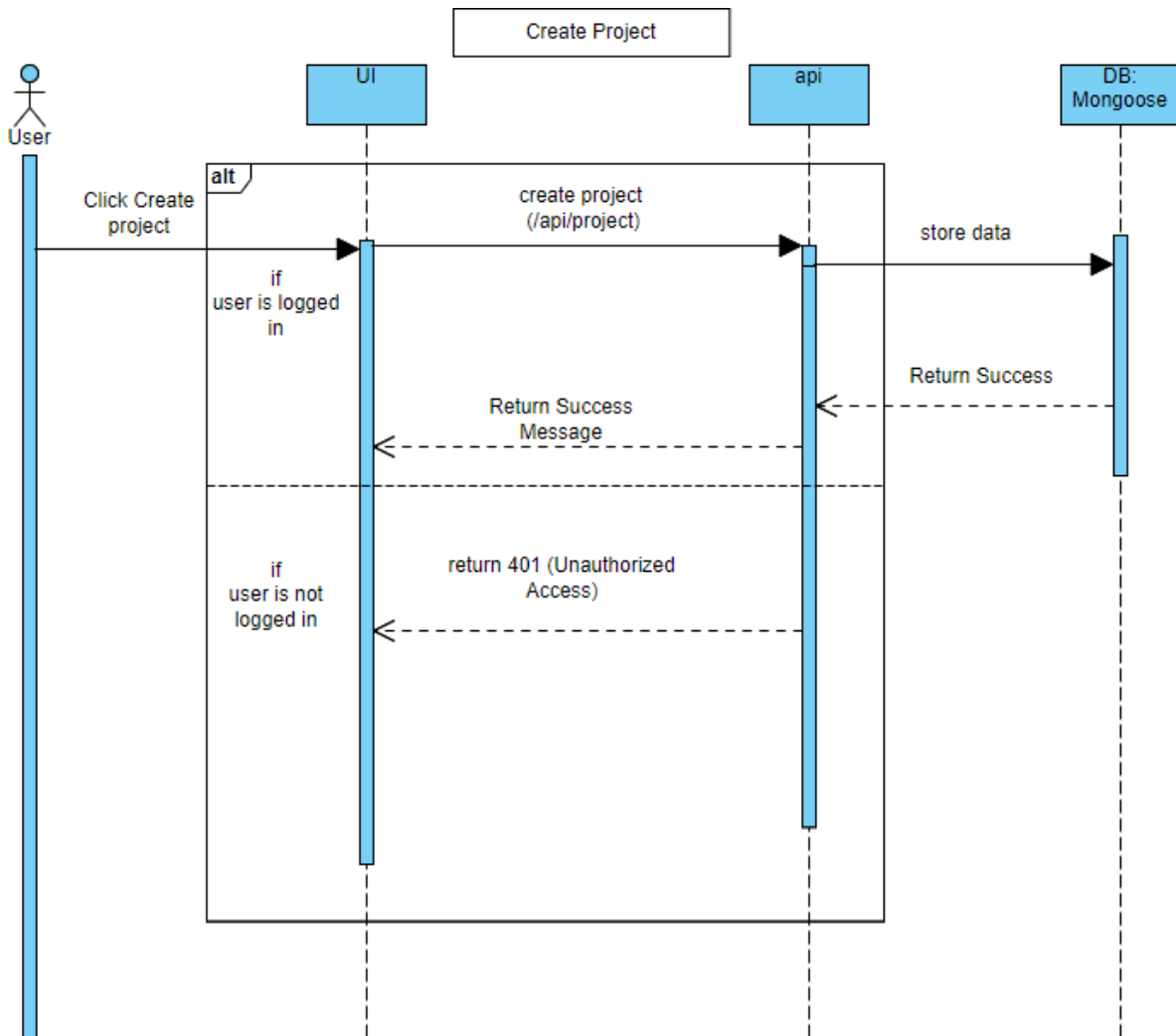


Figure 4.6: Sequence Diagram - Create Project



### 3.5.7 Project List

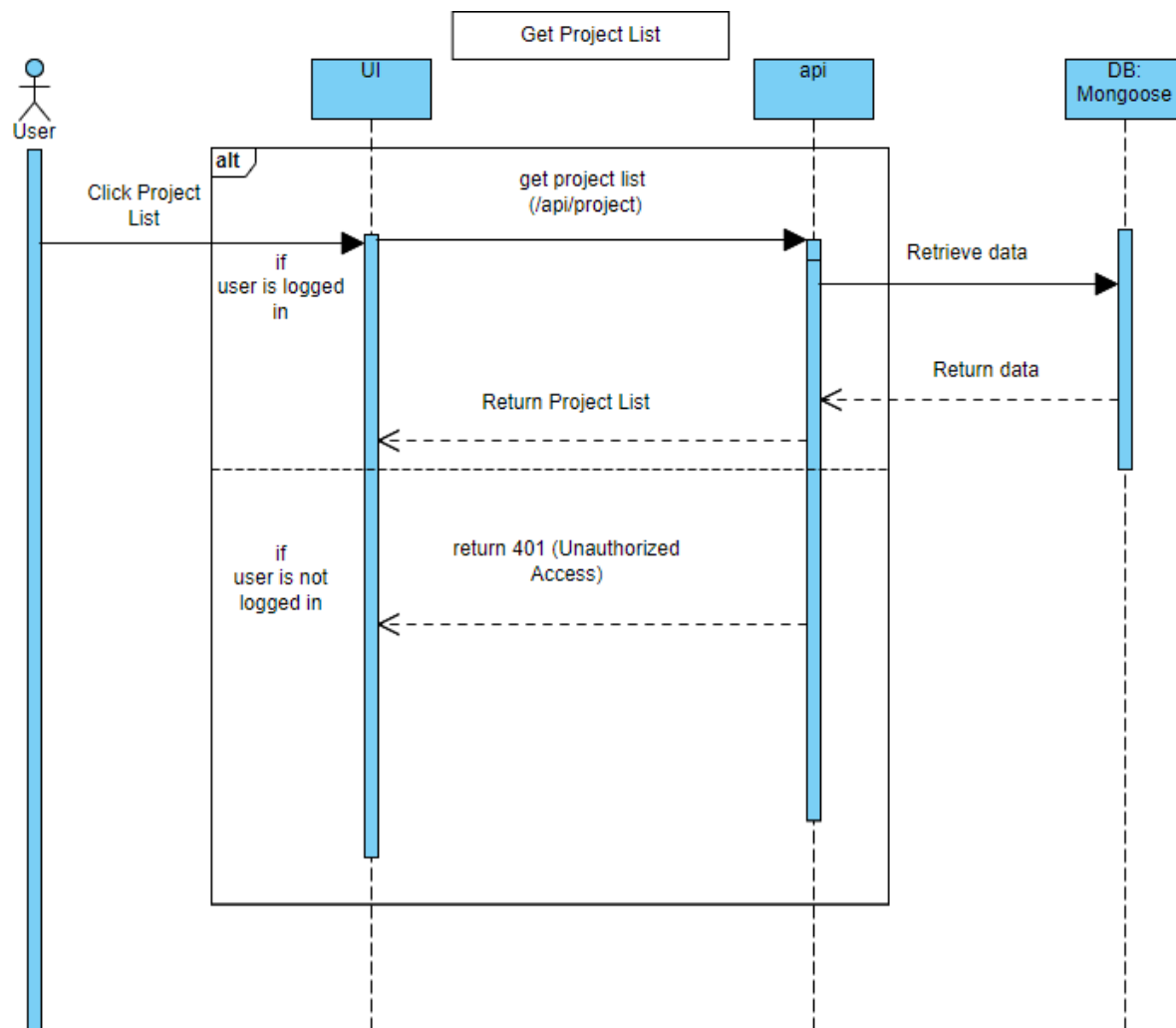


Figure 4.7: Sequence Diagram - Project List

### 3.5.8 Kanban Board

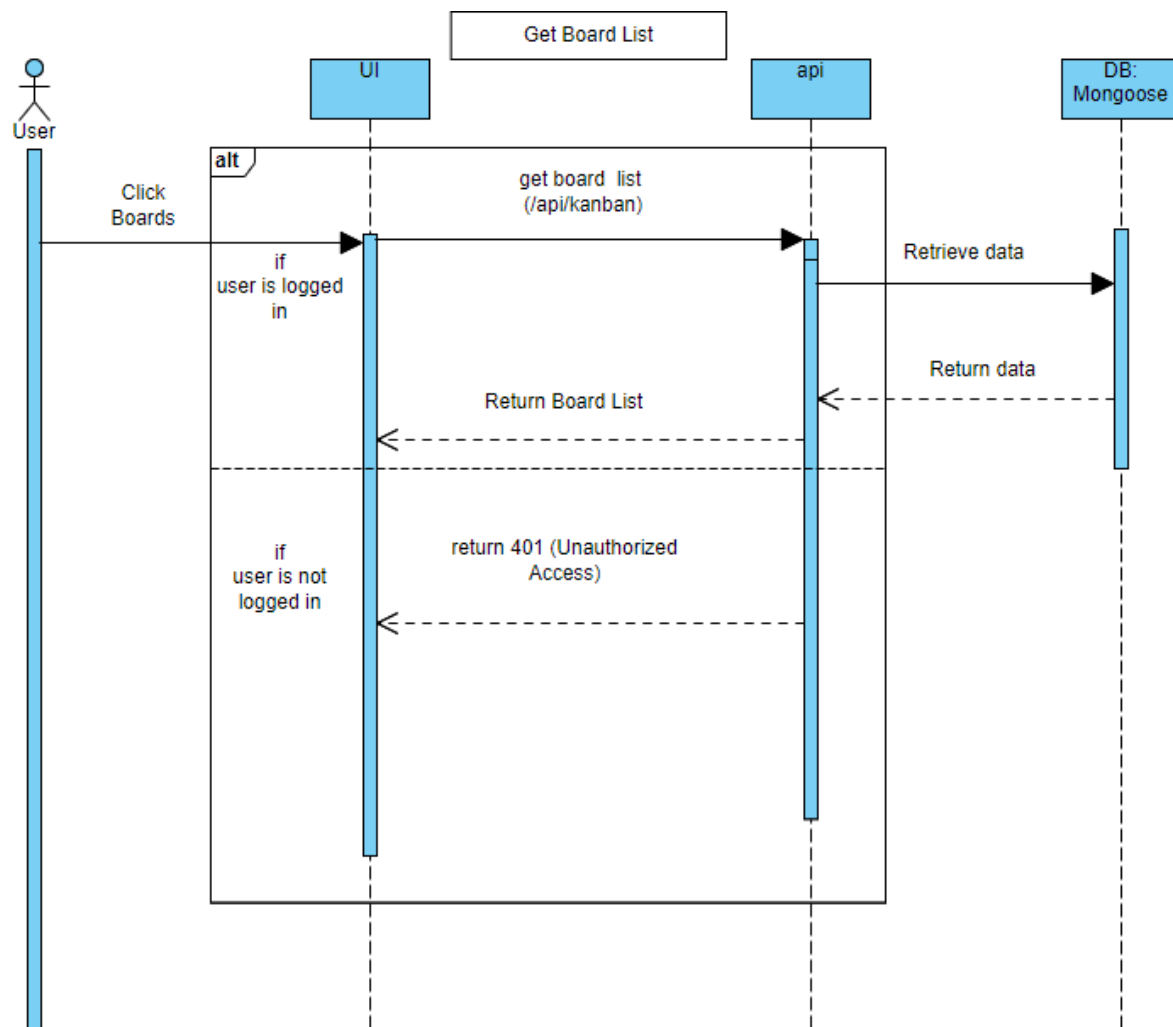


Figure 4.8: Sequence Diagram - Kanban Board

### 3.6 Entity Relationship Diagram

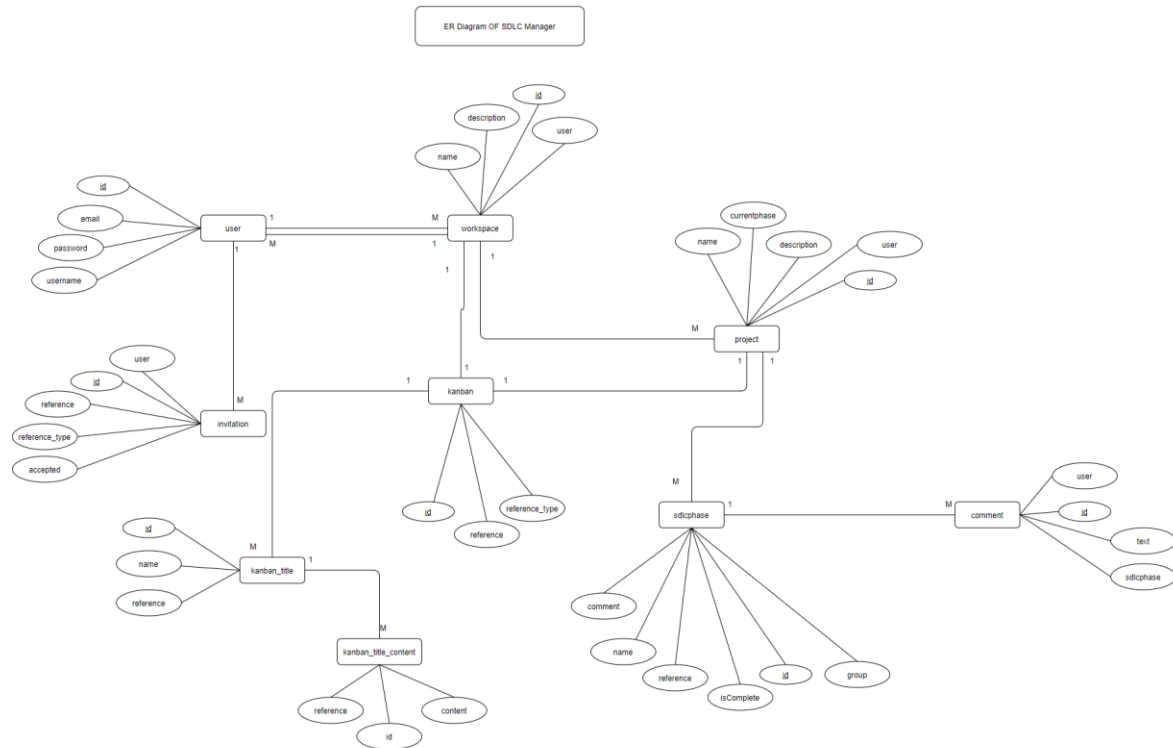


Figure 5.1: Entity Relationship Diagram

## CHAPTER 4: SYSTEM TESTING

### 4.1 Feature Testing

Feature testing is considered to add or modify the new functionality to the existing system. Every feature and functionality have different characteristics. Those are designed to make the web application more useful, reliable, and effective, and secure.

#### 4.1.1 Features to be tested

Features	Priority	Description
Registration	1	User Information must be saved properly
Sign In	1	User must be authenticated properly
Sign Out	2	The token must be destroyed properly
Create New Workspace	1	Newly created workspace must be saved properly
Create New Project	1	Newly created project must be saved properly
Invite Member to workspace	2	Invited member must have invitation message/notification
Change Member role	2	Member role must be changed
Here, 1 = High Priority, 2 = Medium Priority, 3 = Low Priority		

## 4.2 Testing strategies

### 4.2.1 Test Approach

I have two different types of testing to ensure the quality of my system. These two testing systems include functional and structural testing.

- Black Box Testing, which is also called functional testing, is testing that ignores the internal mechanism of a system or component and focuses solely on the outputs.
- White box testing is testing that checks the internal mechanism of a system or component.

### 4.2.2 Pass/Fail Criteria

I set some pass or fail criteria for unit testing. I have prepared the pass/fail criteria based on if input data worked or not. Those which are worked are considered pass and rest of the data is considered a failure.

### 4.2.3 Testing Schedule

Test Phase	Time
Testing plan creation	1 week
Unit Testing	During development time
Component test	During development time
Integration testing	4 days
UI testing	3 days
Load testing	1 week
Performance testing	1 week
Accessibility testing	1 week

### 4.3 Testing Environment

Testing environment means to prepare the environment with hardware and software so that testers can be able to execute test cases as required. For the testing environment, I used these

- Test data
- Api Server(Heroku)
- Database Server (Mongo Atlas)
- Front-end running environment (Netlify)
- Network
- Browser
- System and application

### 4.4 Test Cases

Test cases are those by which a tester can determine whether a system can be able to perform better under test cases properly.

#### 4.4.1 Registration

Test case # 1	Test case name: Registration
System: SDLC Manager	
Designed By: Tonmoy Chiran	Designed Date:
Executed by:	Executed Date:
Short description: System will save user information	

<b>Pre-conditions:</b>						
•						
Serial	Email	Username	Password	Expected result	Pass/Fail	Remarks
1	johndoe@gmail.com	johndoe	abcd1234	User information must be saved	Pass	
2				User must received a message(All field must be filled)	Pass	
3	johndoe@gmail.com	johndoe	abcd1234	User must receive error message(User already registered)	Pass	
<b>Post-conditions:</b> User will have a successfully registered or failed message						

#### 4.4.2 Login

Test case # 2	Test case name: Login
System: SDLC Manager	
Designed By: Tonmoy Chiran	Designed Date:
Executed by:	Executed Date:
Short description: System will retrieve and return access token	

<b>Pre-conditions:</b>					
•					
Serial	Email	Password	Expected result	Pass/Fail	Remarks
1	johndoe@gmail.com	abcd1234	User must be redirected to workspace list	Pass	
2			User must received a message(All field must be filled)	Pass	
3	alex@gmail.com	abcd1234	User must receive error message(Invalid Credentials)	Pass	
<b>Post-conditions:</b> User will have be redirected to workspace list page					

#### 4.4.3 Create New Workspace

Test case # 3	Test case name: Create New Workspace
System: SDLC Manager	
Designed By: Tonmoy Chiran	Designed Date:
Executed by:	Executed Date:



Short description: System will create a new workspace				
<b>Pre-conditions:</b>				
<ul style="list-style-type: none"> <li>• User must be logged in</li> <li>• User must be in workspace list page</li> </ul>				
Serial	Workspace Name	Expected result	Pass/Fail	Remarks
1	Defense	New workspace must be created	Pass	
2		A message “Please fill all the field” must be shown to the user	Pass	
3	Defense	Must return a message “Workspace already exists”	Pass	
<b>Post-conditions:</b> User will have a new workspace list after new workspace creation				

#### 4.4.4 Create New Project

Test case # 4		Test case name: Create New Project			
System: SDLC Manager					
Designed By: Tonmoy Chiran		Designed Date:			
Executed by:		Executed Date:			
Short description: System will create a new project					
<b>Pre-conditions:</b>					
<ul style="list-style-type: none"> <li>• User must be logged in</li> <li>• User must be in specific workspace page</li> </ul>					
Serial	Workspace Name	Select Workspace	Expected result	Pass/Fail	Remarks
1	Sdlc Manager	Defense	New project must be created	Pass	
2			A message “Please fill all the field” must be shown to	Pass	

			the user		
3	Sdlc Manager		Must return a message “Please fill all the fields”	Pass	
<b>Post-conditions:</b> User will have a new project list after new project creation					

#### 4.4.5 Invite Member

Test case # 5		Test case name: Create New Workspace			
System: SDLC Manager					
Designed By: Tonmoy Chiran		Designed Date:			
Executed by:		Executed Date:			
Short description: System will create a new workspace					
<b>Pre-conditions:</b>					
<ul style="list-style-type: none"> <li>• User must be logged in</li> <li>• User must be in workspace list page</li> </ul>					
Serial	Member Email	Expected result	Pass/Fail	Remarks	
1	johndoe@gmail.com	Invitation to the user must be sent	Pass		
2		A message “Please fill all the field” must be shown to the user	Pass		
3	alexa@gmail.com	Must return a message “User Not Found”	Pass		
<b>Post-conditions:</b> User will have a confirmation message					

## CHAPTER 5: USER MANUAL

### 5.1 Registration

Unregistered / new users can register using their email, username and password. After successfully registering, they can login and access all of the features.

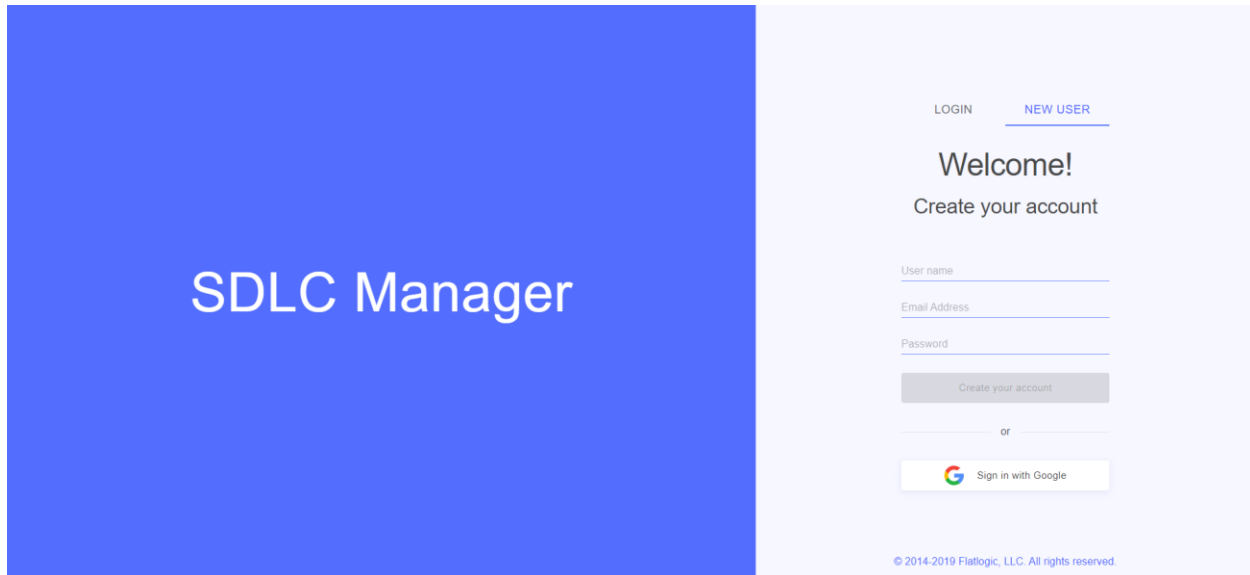


Figure 6.1: User Manual - Registration

### 5.2 Login

After successfully login with correct credentials, users can create workspace, get workspace list or they can access all of the features of the application.

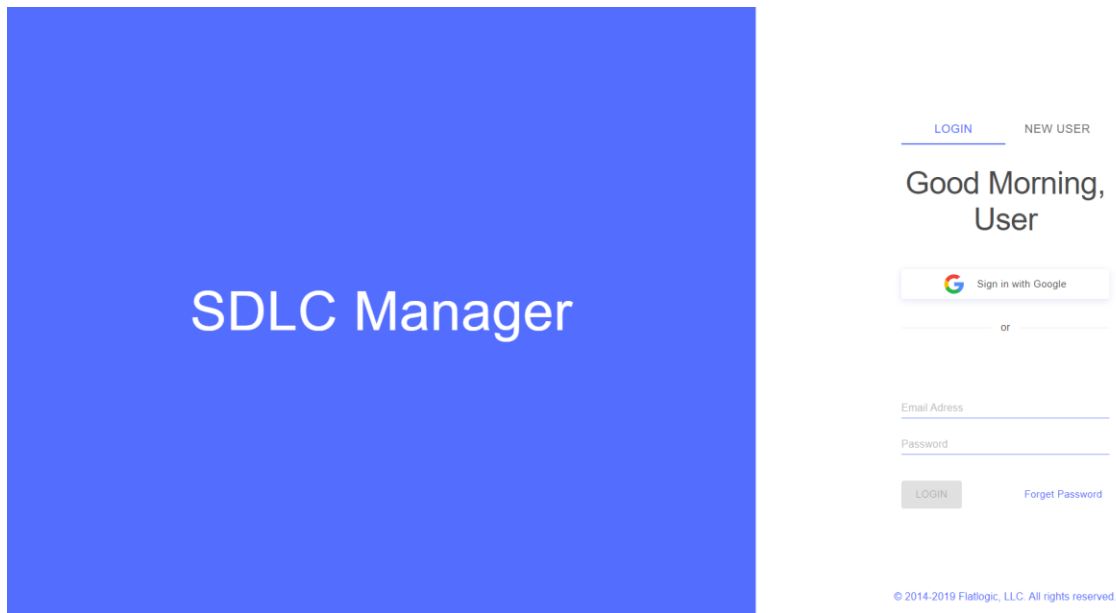


Figure 6.2: User Manual - Login

### 5.3 Workspace

After successfully login with the right credentials, users will be redirected to workspace view, where they can create new workspace, checkout their workspace list and more.



Figure 6.3: User Manual - Workspace List

## 5.4 Specific workspace

In the workspace list, users can click on a specific workspace and see details about that particular workspace like creation of project, kanban board or invite member.

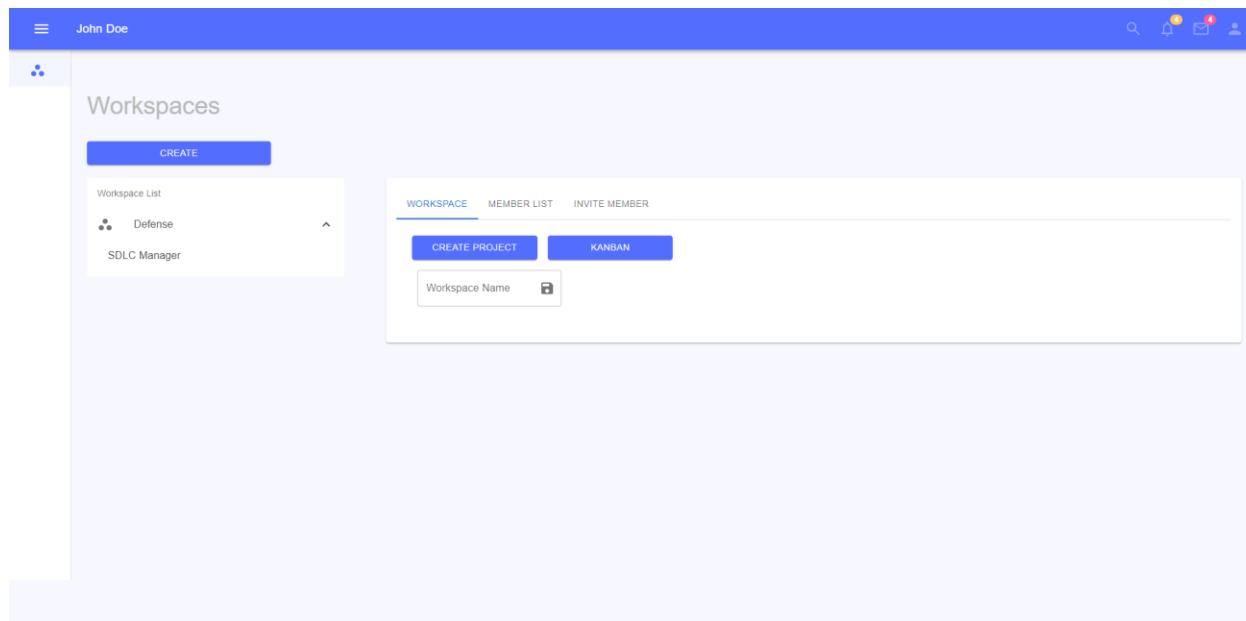


Figure: 6.4: User Manual-Specific Workspace

## 5.5 Create Workspace

If a user wishes to create a new workspace for some particular project, they can create it by clicking create inside workspace page.

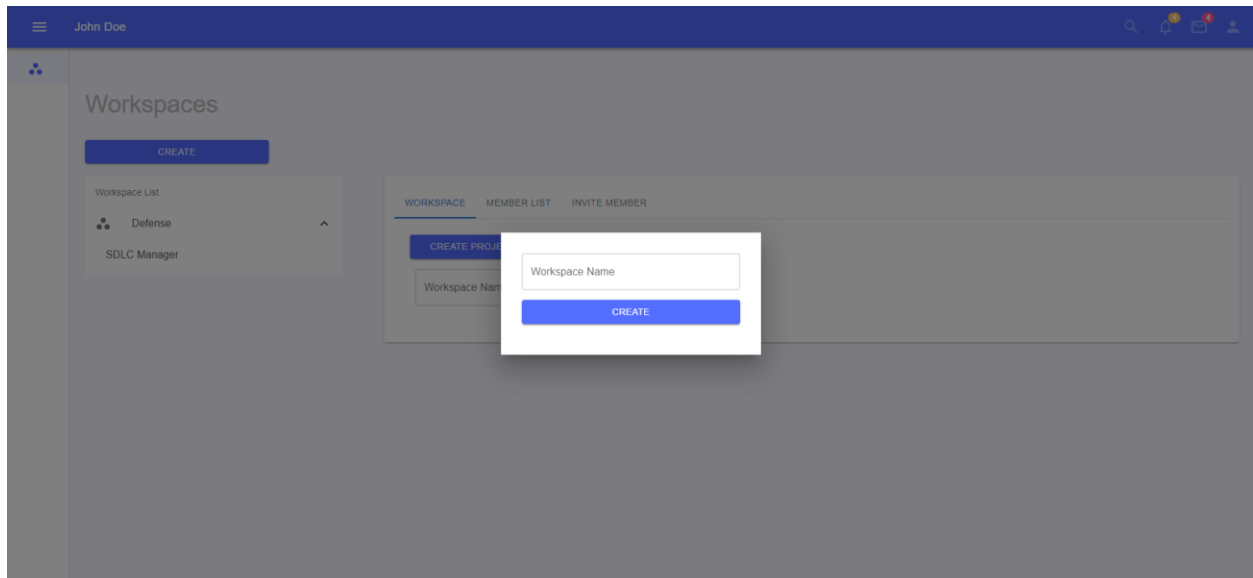


Figure 6.5: User Manual - Create Workspace

## 5.6 Specific Project

Users can directly go to a project by clicking on the project name which is below the workspace on the left side.

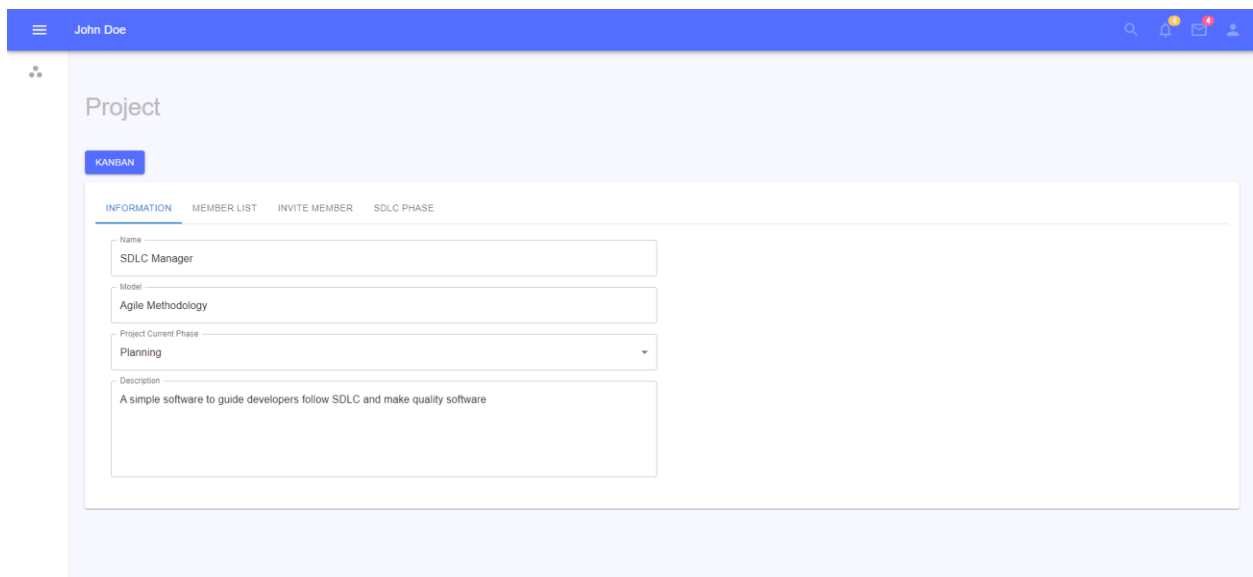


Figure 6.6: User Manual - Specific Project

## 5.7 Kanban

Users can use the kanban board also of specific workspace or of specific project clicking kanban inside specific workspace or project details. In kanban users can create new boards or add content inside precreated titles.

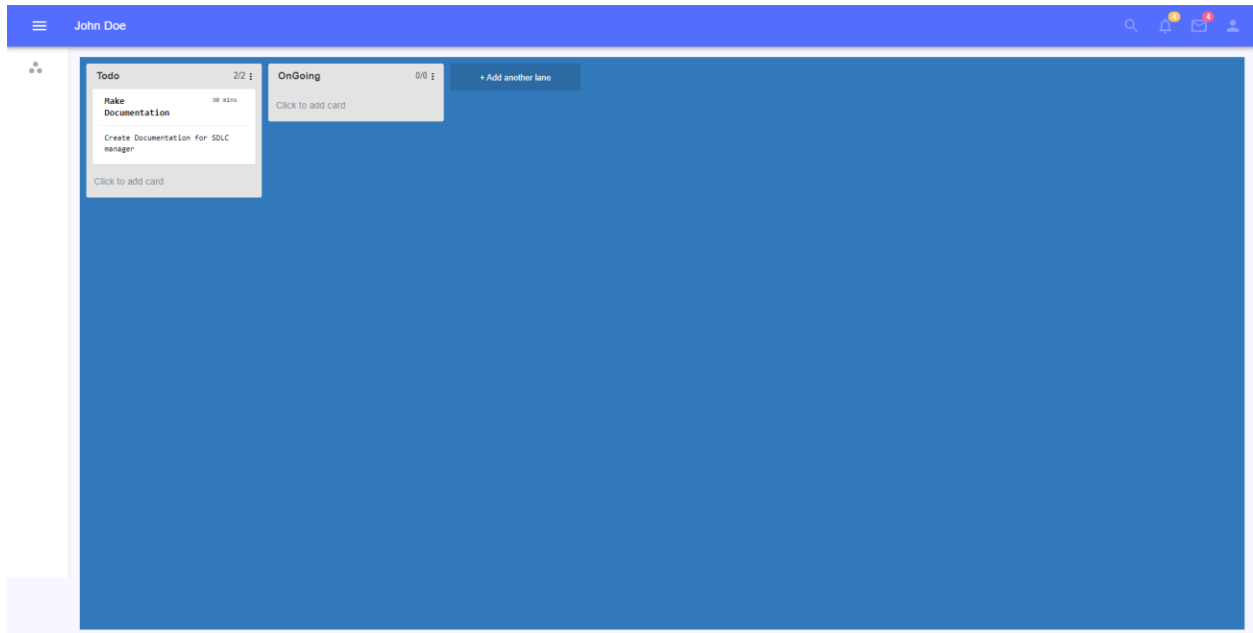


Figure 6.7: User Manual - Kanban

## CHAPTER 6: CONCLUSION

### 6.1 Project Summary

“SDLC Manager” is a web-based application to help/guide developers to follow sdlc standard and minimize bugs of a software. In this system a user can create specific workspace for specific works, so their works can be more maintainable. As sdlc manager guides through all of the sdlc phases, users will follow and make a software that will be with less bugs.

I have completed this project from planning to development within a tight period and completed the system ready to use. This system should be updated regularly with improved and newer features.

### 6.2 Limitations

As there was a limitation of time, I could not include some advanced features for this system, which could make this system more advanced and rich in features.

- Real time push notification
- Real time invitation notification
- It is now only web based, pwa(Progressive web app) and native mobile application is yet not developed.

### 6.3 Obstacles and Achievements

As the system is made with mern(mongo,express,react,node js), I had a lot of obstacles while trying to make the system, as the system includes some array and nested array inside of the database, which were a little difficult at first to maintain. But with time and practice, I overcame them.

### 6.4.Future Scope

This software system will be a SDLC manager for both software industries along with individuals developers and small teams. This system is supposed to ensure that a software system is built maintaining the standard SDLC phases, so there is less chance of software product failure after deploying.



More specifically, this system is designed to ensure that each software firm including small teams and individual developers follow the SDLC.

- A mobile application can be developed with some advanced features
- A progressive application can be developed for this application
- This application can be deployed in a strong server for lots of traffic for a lot users

## CHAPTER 6: REFERENCE

- Trello Board
- ITL Bulletin The System Development Life Cycle (SDLC), April 2009 (nist.gov)