**GLOBAL SOFTWARE PROJECT MANAGEMENT TOOL**

### Anosh Zahid (BCS191088)

**Emaan Saqib (BCS191039)**

**Ahmad Tahir (BCS191070)**



## Spring 2022 Supervised By Hafiz Anas Bilal

**Department of Computer Science**

## Capital University of Science & Technology, Islamabad

Submission Form for Final-Year

PROJECT REPORT

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | V1.0 | **NUMBER OF MEMBERS** | 3 |
|  |  |  |  |
| **TITLE** | **GLOBAL SOFTWARE PROJECT MANAGEMENT TOOL** | |  |
|  |  | |  |
| **SUPERVISOR NAME** | | **Hafiz Anas Bilal** |  |

|  |  |  |
| --- | --- | --- |
| **MEMBER NAME** | **REG. NO.** | **EMAIL ADDRESS** |
| Anosh Zahid | BCS191088 | [anoshzahid@gmail.com](mailto:anoshzahid@gmail.com) |
| Emaan Saqib | BCS191039 | [Saqibemaan2027@gmail.com](mailto:Saqibemaan2027@gmail.com) |
| Ahmad Tahir | BCS191070 | [Ahmadsidiqui008@gmail.com](mailto:Ahmadsidiqui008@gmail.com) |

|  |
| --- |
| **MEMBERS’ SIGNATURES** |
|  |
|  |
| **Supervisor’s Signature** |

# APPROVAL CERTIFICATE

This project, entitled as “GLOBAL SOFTWARE PROJECT MANAGEMENT TOOL” has been approved for the award of

# Bachelors of Science in Computer Science

**Committee Signatures:**

Supervisor:

(Mr. Hafiz Anas Bilal)

Project Coordinator:

(Mr. Abdul Basit)

Head of Department:

(Dr. Nayyer Masood)

# DECLARATION

I/We, hereby, declare that “No portion of the work referred to, in this project has been submitted in support of an application for another degree or qualification of this or any other university/institute or other institution of learning”. It is further declared that this undergraduate project, neither as a whole nor as a part thereof has been copied out from any sources, wherever references have been provided.

**MEMBERS’ SIGNATURES**

# ACKNOWLEGMENT

We are thankful to the Almighty Allah to lend us this chance in life and making us capable of doing it. We also thank our parents, guardians and most specifically Sir Hafiz Anas Bilal for guiding us to the right project and making us enlighten with the knowledge without which we would have been unable to go our Final Year Project.

**Table of Contents**

[**CHAPTER 1** Error! Bookmark not defined.](#_Toc103695838)

[**INTRODUCTION 1**](#_Toc103695839)

[**1.1. Project Introduction 1**](#_Toc103695840)

[**1.2. Existing Examples / Solutions 3**](#_Toc103695841)

[**1.2.1. BitBucket: 3**](#_Toc103695842)

[**1.2.2. DSpace BU Repository: 3**](#_Toc103695843)

[**1.2.3. Final Year Project Management System: 3**](#_Toc103695844)

[**1.3. Comparison of Existing Solutions 5**](#_Toc103695845)

[**1.4. Business Scope 5**](#_Toc103695846)

[**1.5. Useful Tools and Technology 6**](#_Toc103695847)

[**1.5.1. Languages used: 6**](#_Toc103695848)

[**1.5.2. Visual studio and visual studio code: 6**](#_Toc103695849)

[**1.5.3. Asp.net: 6**](#_Toc103695850)

[**1.5.4. MySQL: 7**](#_Toc103695851)

[**1.5.5. Platform: 7**](#_Toc103695852)

[**1.6. Project Work Breakdown 8**](#_Toc103695853)

[**1.7. Project Timeline 9**](#_Toc103695854)

[**Chapter 2 10**](#_Toc103695855)

[**Requirement specification and analysis 10**](#_Toc103695856)

[**2.1 Functional Requirement: 10**](#_Toc103695858)

[**2.2 Non-Functional Requirements 12**](#_Toc103695859)

[**2.3 Selected Functional Requirement 13**](#_Toc103695860)

[**2.4 System Use Case Modeling 15**](#_Toc103695861)

[**2.5 System Sequence diagram 44**](#_Toc103695862)

[**2.6 Domain Model 56**](#_Toc103695863)

[**Chapter 3 57**](#_Toc103695864)

[**System Design 57**](#_Toc103695865)

[**3.1 Layer Definition 57**](#_Toc103695866)

[**3.2 Software Architecture 58**](#_Toc103695867)

[**3.3 Class Diagram 59**](#_Toc103695868)

[**3.4 Sequence Diagram 60**](#_Toc103695869)

[**3.5 Data Flow Diagram 65**](#_Toc103695870)

[**3.6 User Interface Design 66**](#_Toc103695871)

# INTRODUCTION

## Project Introduction

When working on a new project, both the project team and the project team lead come across several problems. The first thing they need is digital repository where they can see which similar projects have been done before. Then arises the problem of determining that to what extent that project idea has been implemented. Moreover, there is also a problem of project monitoring in which the team leader and the project team are communicating verbally, this can lead to clear misunderstanding and delay in essential tasks. The repositories for maintaining a record of the previous projects already do exist but they don’t offer an all in one solution where you can save, monitor, and check similarity of the project artifacts.

To overcome these problems, we are proposing an all-in-one solution, “Global Software Project Management Tool”. This system will have a projects repository that will be accessible to both team leads and the project team members according to their level of authority and domain. The repositories will hold the following elements of the projects:

* + 1. Project Scope
    2. Functional Requirements
    3. Non-Functional Requirements
    4. UML Artifacts
       1. Use Cases
       2. System Sequence Diagrams (SSDs)
       3. Activity Diagrams
       4. Domain Models
       5. Class Diagrams
    5. Data Models
    6. System Architecture
    7. Code
    8. Test Cases

“GLOBAL SOFTWARE PROJECT MANAGEMENT TOOL” also provides similarity measurement functionality. The system will determine to what extent the idea has been implemented by searching the projects already saved in the repository based on the functional requirements and scope provided by the user. This also leads to reusability of artifacts that have already been implemented. In addition to that, the system acts as a platform for the team leads and the project team members to communicate. The team lead will be able to assign tasks with deadlines and respective documents plus he or she can approve or disapprove the completed tasks. This helps the team lead to monitor the project easily.

Our project consists of following three modules:

1. Project repository

* Maintain project repository with software artifacts
* Integration of case tools

1. Project progress monitoring

* List of tasks
* Task allocation
* Task artifacts
* Monitoring progress

1. Identification of similar and matching artifacts on bases of project scope and functional requirements

This proposed system can also benefit the educational sector as the students and the project supervisors face similar problems while selecting a topic for the Final Year Project (FYP) for their degree. The repositories of the FYP artifacts do already exist but in physical form, if a student is willing to look upon a project that has been done, he or she may have to go through all the documents available.

## Existing Examples / Solutions

Some existing mobile applications that provide similar functionalities are listed below:

### BitBucket:



This web application only holds the data of several global users and does not provide any natural language processing techniques to determine how much of the functional requirements and scope of a certain project matches with the given input. And it only provides repositories on a monthly rental basis.

### DSpace BU Repository:



DSpace is a web application based online repository of fyp for bahria university Islamabad. It is a digital service that collects, preserves, and distributes digital material. It lets project coordinators to evaluate students’ fyp related tasks and the coordinators can also schedule their meetings through it.

### Final Year Project Management System:

Implemented in Tunku Abdul Rahman University College, Malaysia, The functionality of Final Year Project Management System is to manage students’ final year projects. It assists in managing all the operations of final year project. The Final Year Project Management System consists of 8 features which include User Management, Appointment Making, Students-Supervisor Pairing, Quota & Workload, Project Allocation, Announcement, Assessment and Reports Generation.

## Comparison of Existing Solutions

None of these applications takes functional requirements or scope of the project as an input and compare it to the existing final year projects in the database to see how much of the scope and functional requirements of a similar idea has already been implemented.

The following Table 1.1 shows the comparison between the existing platforms.

**Table 1.1 Comparison of existing websites**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Sr. No.** | **Characteristics** |  | **BitBucket** | **DSpace BU Repository** | **Final Year Project Management System** | **Proposed System** |
| **1** | **Repository** |  | ✓ | ✓ | ✓ | ✓ |
| **2** | **Comparison** |  | ❌ | ❌ | ❌ | ✓ |
| **3** | **Category based search projects** |  | ❌ | ❌ | ❌ | ✓ |
| **4** | **Paid** |  | ✓ | ❌ | ❌ | ❌ |

## Business Scope

“Higher education is a $1.9 trillion industry. For comparison, this is how the higher education market compares with other large global markets. Healthcare - $1.7 trillion. Information technology - $3.4 trillion. Transportation - $4.7 trillion.”

The system that we are building is capturing a market that big, we make sure to supply packages that make it affordable to small educational institutions as well. There is this gap in the educational sector that falls for this system as there is no such platform that provides this functionality is specific.

“Revenue in the Software market is projected to reach US$626.50bn in 2022. The market's largest segment is Enterprise Software with a projected market volume of US$250.30bn in 2022.”

Our Project also reflects in the field of corporate software industry. There are several numbers of software’s and websites that are currently present in the market, but each software holds just one or two of the functionalities that our proposed system offers. Our devised system can be implemented in a form that can also cover the software industry.

## Useful Tools and Technology

In this section, we are describing the tools and technologies that will be used during this project:

### 1.5.1. Languages used:

Languages used in this project includes, C#, html, bootstrap, css, jQuery, angular js, and Asp.net. The following languages are used to make a website and to support these languages following software are used.

### 1.5.2. File:Visual Studio Code 1.35 icon.svg - Wikimedia CommonsVisual studio and visual studio code:

Visual Studio Code for the Web provides a free, zero-install Microsoft Visual Studio Code experience running entirely in your browser, allowing you to quickly and safely browse source code repositories and make lightweight code changes.

### Asp.net:

We will write our web applications and services with ASP.NET Core using Visual Studio.

### Download MySQL Community Edition Linux 4.1.24 / 4.0.30 / 3.23.58MySQL:

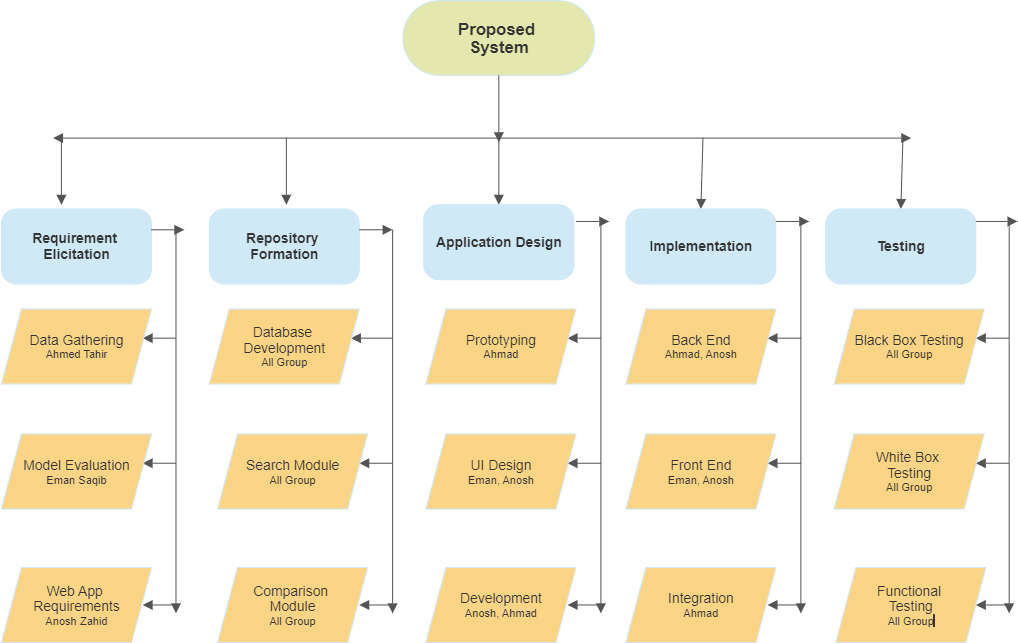
Our project will deal with database systems as well so we will be using MySQL community edition.

### Platform:

As we talk about the operating system, our project will be web-based application so there will be no constraint with respect to operating system. It will be a web-based platform independent application.

## Project Work Breakdown

In Figure 1 below, work breakdown of the project is shown. The project is broken down into five phases, each phase consisting of its own separate tasks that are allocated to the members of the project according to their expertise.



**Figure 1.1 project word breakdown**

## Project Timeline

In Figure 2, the timeline of this project is shown. All of the tasks that have to be done are mentioned along with their deadlines and durations.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | **Project Timeline** | | | | | | | | | |  | | | | | | |
| ***ID*** | ***Task Name*** | | ***Start*** | ***Finish*** | ***Duration*** |  | ***Q2 22*** | | | ***Q3 22*** | | | ***Q4 22*** | | | ***Q1 23*** | |
|  | ***Apr*** | ***May*** | ***Jun*** | ***Jul*** | ***Aug*** | ***Sep*** | ***Oct*** | ***Nov*** | ***Dec*** | ***Jan*** | ***Feb*** |
| **1** | **Requirements Management** | | **3/15/2022** | **4/8/2022** | **19d** |  |  | | | | | | | | | | | |
|  |
| **2** | **System Analysis** | | **4/1/2022** | **4/15/2022** | **11d** |  | | | | | | | | | | | | |
| **3** | **Design & Architecture** | | **4/18/2022** | **5/27/2022** | **30d** |  | | | | | | | | | | | | |
| **4** | **Implementation** | | **5/30/2022** | **12/16/2022** | **145d** |  | | | | | | | | | | | | |
| **5** | **Testing** | | **12/19/2022** | **1/13/2023** | **20d** |  | | | | | | | | | | | | |
| **6** | **Deployment** | | **1/16/2023** | **1/20/2023** | **5d** |  | | | | | | | | | | | | |

**Figure 1.2project work breakdown**

# Chapter 2

# Requirement specification and analysis

# Requirements analysis is a process of determining user expectations for a new or modified product. These features, called requirements, must be quantifiable, relevant and detailed. In software engineering, such requirements are often called functional specifications. In Chapter 2 we will enlist the functional and non-functional requirements and model functional requirements in the form of use case model.

## Functional Requirement:

Functional requirement define a function of a system or its component. Functional requirement may be calculations, technical details, data manipulation and processing and other specific functionality that define what a system is supposed to accomplish.

**Table 2.1 functional requirements**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.**  **No.** | **Functional Requirements** | **Type** | **Status** |
|  | **User Management and Authentication** |  |  |
| 1. | The system administrator shall be able to register into the system. | core | pending |
| 2. | The system administrator shall be able to login to the system using username and password. | core | pending |
| 3. | The system administrator shall be able to create users account into the system. | core | pending |
| 4. | The system administrator shall be able to delete users into the system. | core | pending |
| 5. | Users shall be able to login to the system using their username and password. | core | pending |
| 6. | Users shall receive their username and password via email. | core | pending |
| 7. | User shall be able to change their passwords. | core | pending |
| 8. | User shall be able to view their profiles. | core | pending |
| 9. | User shall be able to update their profiles. | core | pending |
| 10. | The Project manager shall be able to view profiles of team lead and team member. | core | pending |
| 11. | Team leads shall be able view their team member’s profile. | core | pending |
|  | **Project Repository Management** |  |  |
| 12. | The System administrator shall be able to add projects in the repository. | core | pending |
| 13. | The System administrator shall be able to delete projects in the repository. | core | pending |
| 14. | User shall be able to search projects in the project repository. | core | pending |
| 15. | The Project Manager shall be able to update artifacts of the existing projects. | intermediate | pending |
| 16. | The Project Manager shall be able to check similarity between new projects and existing projects in the repository on the basis of functional requirement. | core | pending |
| 17. | The Project Manager shall be able to check similarity between new project and existing projects in the repository on the basis of scope. | core | pending |
|  | **Project Management and Monitoring** |  |  |
| 18. | The Project Manager shall be able to create new projects where it assign the project and roles to the members and set deadlines accordingly. | core | pending |
| 19. | The Project Manager shall be able to Browse projects from projects list. | core | pending |
| 20. | Team member shall be able to view assigned tasks and deadline of the tasks. | intermediate | pending |
| 21. | Team lead shall be able to submit completed tasks to the Project manager. | intermediate | pending |
| 22. | Team member shall be able to submit completed tasks to the Team lead. | intermediate | pending |
| 23. | The Project manager shall be able to add remarks to each assigned project. | intermediate | pending |
| 24. | Team lead shall be able to add remarks to each assigned task in the project. | intermediate | pending |
|  | **Dashboard** |  |  |
| 25. | The User shall be able to get project status reports. | intermediate | pending |
| 26. | The User shall be able to view charts and graphs. | intermediate | pending |

## Non-Functional Requirements

A non-functional requirement is a requirement that specifies criteria that can be used to judge the operation of a system, rather than specific behaviors. They are contrasted with functional requirements that define specific behavior or functions.

**Table 2.2 non-functional requirements**

|  |  |  |
| --- | --- | --- |
| **SR.** | **NON FUNCTIONAL REQUIRMENT** | **CATEGORY** |
| 1 | System shall be online 24 by 7 | Availability |
| 2 | The web application shall be easy to use by all employees including manager, team lead and team members. | Usability |
| 3 | Only authorized person should be able to login into the website | Security |
| 4 | Website will recover in 20 to 30 second in case of crash | Maintainability |
| 5 | Website will roughly take 5 to 10 seconds in searching repositories and giving output | Efficiency |

## Selected Functional Requirement

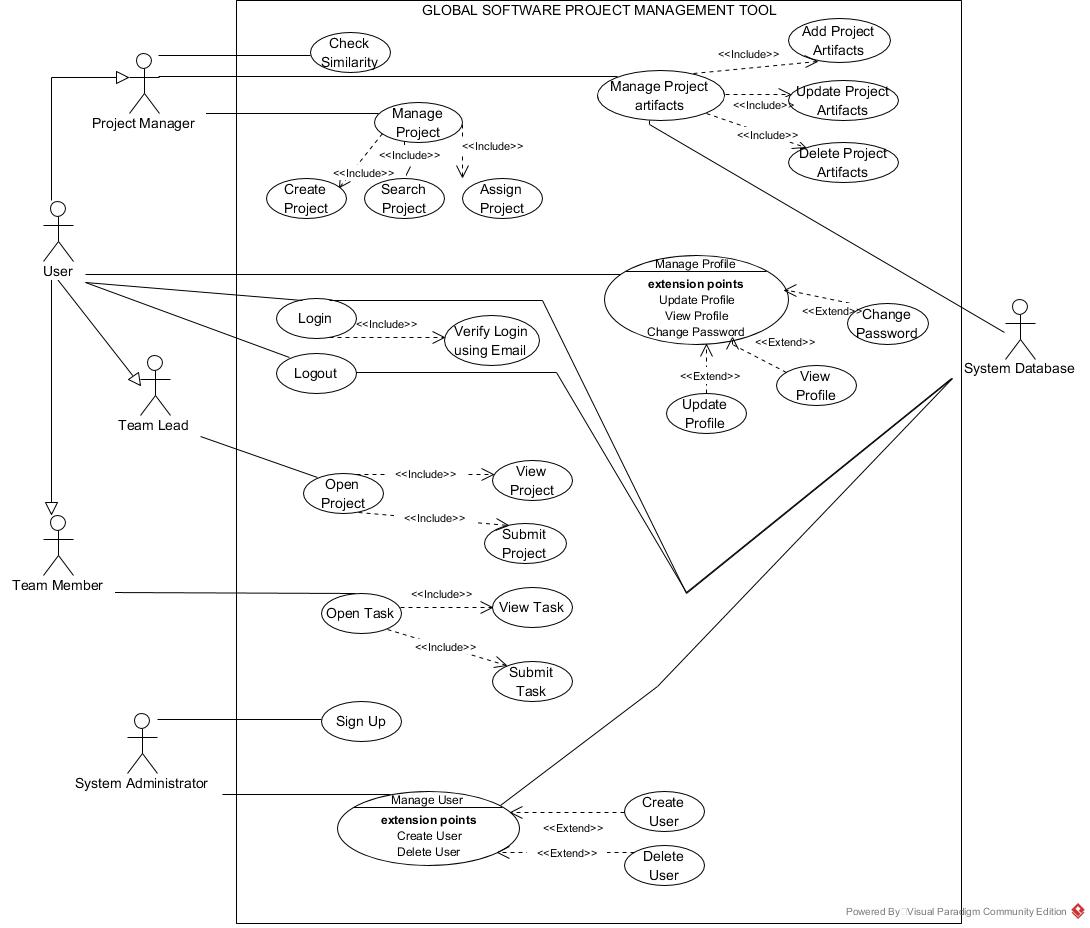
Following is the list of the requirements selected for the current iteration.

**Table 2.3 Selected functional requirement**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.**  **No.** | **Functional Requirements** | **Type** | **Status** |
|  | **User Management and Authentication** |  |  |
| 1. | The system administrator shall be able to register into the system. | core | pending |
| 2. | The system administrator shall be able to login to the system using username and password. | core | pending |
| 3. | The system administrator shall be able to create users account into the system. | core | pending |
| 4. | The system administrator shall be able to delete users into the system. | core | pending |
| 5. | Users shall be able to login to the system using their username and password. | core | pending |
| 6. | Users shall receive their username and password via email. | core | pending |
| 7. | User shall be able to change their passwords. | core | pending |
| 8. | User shall be able to view their profiles. | core | pending |
| 9. | User shall be able to update their profiles. | core | pending |
| 10. | The Project manager shall be able to view profiles of team lead and team member. | core | pending |
| 11. | Team leads shall be able view their team member’s profile. | core | pending |
|  | **Project Repository Management** |  |  |
| 12. | The System administrator shall be able to add projects in the repository. | core | pending |
| 13. | The System administrator shall be able to delete projects in the repository. | core | pending |
| 14. | User shall be able to search projects in the project repository. | core | pending |
| 15. | The Project Manager shall be able to update artifacts of the existing projects. | intermediate | pending |
| 16. | The Project Manager shall be able to check similarity between new projects and existing projects in the repository on the basis of functional requirement. | core | pending |
| 17. | The Project Manager shall be able to check similarity between new project and existing projects in the repository on the basis of scope. | core | pending |

## System Use Case Modeling

A use case is a list of actions or event steps, typically defining the interactions between a role (known in the Global Software Project Management Tool as an actor) and a system, to achieve a goal. The actor can be a human or other external system, use cases are shown in the following the figure.



**Figure 2.1 System Use case diagram**

* + 1. **Register(System Administrator)**

**Table 2.4 Usecase-1 (register admin)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Use Case ID:** | UC1 | | | | |
| **Use Case Name:** | Register System Administrator | | | | |
| **Created By:** | Anosh Zahid | | **LastUpdatedBy:** | | Anosh Zahid |
| **Date Created:** | 4/18/2021 | | **LastRevisionDate:** | | 4/25/2021 |
| **Actors:** | | System Administrator | | | |
| **Description:** | | The System Administrator will register into the system using his/her information i.e., username, password, confirm password, and email. | | | |
| **Trigger:** | | Click Register As System Administrator Button | | | |
| **Preconditions:** | | 1. The system must have online connectivity. 2. The registration page must be displayed by the system. | | | |
| **Post conditions:** | | 1. The system shows a successful registration message on the same page. 2. The system redirects the system administrator to the login page. | | | |
| **Normal Flow:** | | Actors | | System | |
| 1. The system administrator clicks on register as System Administrator button. | | 1. The System directs system administrator to registration form to fill his information. | |
| 1. The system administrator enters his/her username, password, confirm password, and email. | |  | |
|  | | 1. The system administrator presses the submit button.   8. The system administrator enters the OTP within 100 seconds. | | 1. The System validates if the entered password is strong or weak and makes sure that the password is strong. 2. The system confirms the entered email by sending an OTP on the entered email which the user has to enter within 100 seconds. 3. The system validates if the entered OTP is correct. 4. The system shows a “successfully registered” message on the screen and redirects user to login page. | |
| **Alternative Flows:** | | The system administrator can be provided with username and password manually by the system stakeholders. | | | |
| **Exception:** | | Exception (Add Strong Password) is raised in case the system administrator enters a password less than 6 characters.  Exception (Enter valid email) is raised in case the system administrator fails to enter correct OTP.  Exception (OTP timeout) is raised in case the system administrator fails to enter OTP within 100 seconds. | | | |

* + 1. **Login(System Administrator)**

**Table 2.5 Usecase-2 (Login admin)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **UseCase ID:** | UC2 | | | | |
| **UseCase Name:** | Login System Administrator | | | | |
| **CreatedBy:** | Anosh Zahid | | **LastUpdatedBy:** | | Anosh Zahid |
| **DateCreated:** | 4/18/2021 | | **LastRevisionDate:** | | 4/25/2021 |
| **Actors:** | | System Administrator | | | |
| **Description:** | | The System Administrator will login into the system using his/her credentials i.e., username and password. | | | |
| **Trigger:** | | Click Login Button | | | |
| **Preconditions:** | | 1. The login page must be displayed. 2. The system administrator must be registered. 3. The system administrator has an active and working account. | | | |
| **Postconditions:** | | 1. The system shows a successful login message on the same page. 2. The system redirects the user to the user profile page. | | | |
| **NormalFlow:** | | Actors | | System | |
| 1. The system administrator clicks on login button. | | 1. The System directs user to login form to fill in his credentials i.e., username and password. | |
| 1. The system administrator enters his/her username and password. | |  | |
|  | | 1. The system administrator presses the login button. | | 1. The System validates the username and password from the database and redirects user to user profile page. | |
| **AlternativeFlows:** | | There is no specific alternate flow to this use case. | | | |
| **Exception:** | | Exception (wrong username or password) is raised in case of wrong credential entry by the system administrator. | | | |

* + 1. **Create User(System Administrator)**

**Table 2.6 Usecase-3 (Create user)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **UseCase ID:** | UC3 | | | | |
| **UseCase Name:** | Create User Account | | | | |
| **CreatedBy:** | AnoshZahid | | **LastUpdatedBy:** | | Anosh Zahid |
| **DateCreated:** | 4/18/2021 | | **LastRevisionDate:** | | 4/25/2021 |
| **Actors:** | | System Administrator | | | |
| **Description:** | | The System Administrator shall be able to create user accounts using User ID, their role, and their basic information such as their first and last names, email, field of expertise. | | | |
| **Trigger:** | | Add User Button | | | |
| **Preconditions:** | | 1. The system administrator must be logged in. | | | |
| **Postconditions:** | | 1. The system shows a user successfully created message with automatic generated password on the same page. 2. The system adds the user added in the users list. | | | |
| **NormalFlow:** | | Actors | | System | |
| 1. The system administrator clicks on add user button. | | 1. The System directs user to create user page. | |
|  | | 1. The System opens a form to fill out all the information required for creating a new user i.e., User ID, role, and first and last names, email, field of expertise. | |
|  | | 1. The system administrator enters the User ID, role, and first and last names, email, field of expertise. | |  | |
|  | | 1. The system administrator clicks on submit button on form. | | 1. The System generates a random 6 digit password against the entered member id. | |
|  | |  | | 1. The System shows (successfully added user) message. | |
|  | |  | | . | |
| **AlternativeFlows:** | | There is no specific alternate flow to this use case. | | | |
| **Exception:** | | The system administrator misses out any one of the information boxes then the system raises an alert message to fill out all the information fields before submitting the form. | | | |

* + 1. **Login(User)**

**Table 2.7 Usecase-4 (Login User)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **UseCase ID:** | UC4 | | | | |
| **UseCase Name:** | Login User | | | | |
| **CreatedBy:** | Anosh Zahid | | **LastUpdatedBy:** | | Anosh Zahid |
| **DateCreated:** | 4/18/2021 | | **LastRevisionDate:** | | 4/25/2021 |
| **Actors:** | | Project Manager / Team Lead / Team Member | | | |
| **Description:** | | The User (Project Manager / Team Lead / Team Member) will login into the system using his/her credentials i.e., username and password. | | | |
| **Trigger:** | | Click Login Button | | | |
| **Preconditions:** | | 1. The login page must be displayed. 2. The user (Project Manager / Team Lead / Team Member) must be added by the system administrator. 3. The system has sent the credentials to the user via email. 4. The user (Project Manager / Team Lead / Team Member) must have an active and working account. | | | |
| **Postconditions:** | | 1. The system shows a successful login message on the same page. 2. The system redirects the user to the user profile page. | | | |
| **NormalFlow:** | | Actors | | System | |
| 1. The User (Project Manager / Team Lead / Team Member) clicks on login button. | | 1. The System directs user (Project Manager / Team Lead / Team Member) to login form. | |
| 1. The User (Project Manager / Team Lead / Team Member) enters his/her username and password. | |  | |
|  | | 1. The User (Project Manager / Team Lead / Team Member) clicks the login button. | | 1. The System validates the username and password from the database and redirects user (Project Manager / Team Lead / Team Member) to user profile page. | |
| **AlternativeFlows:** | | There is no specific alternate flow to this use case. | | | |
| **Exception:** | | Exception (wrong username or password) is raised in case of wrong credential entry by the user (Project Manager / Team Lead / Team Member). | | | |

* + 1. **Delete(User)**

**Table 2.8 Usecase-5 (delete User)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **UseCase Name:** | Delete user accounts | | | | |
| **CreatedBy:** | Anosh Zahid | | **LastUpdatedBy:** | | Anosh Zahid |
| **DateCreated:** | 4/18/2021 | | **LastRevisionDate:** | | 4/25/2021 |
| **Actors:** | | System Administrator | | | |
| **Description:** | | The System Administrator will delete user accounts using User ID. | | | |
| **Trigger:** | | Click Delete a User Button | | | |
| **Preconditions:** | | 1. The system must have online connectivity. 2. The system administrator must be logged in. | | | |
| **Postconditions:** | | 1. The system shows a successful deletion message on the same page. 2. The system redirects the system administrator to the home page. | | | |
| **NormalFlow:** | | Actors | | System | |
| 1. The system administrator clicks on delete a user button. | | 1. The System directs system administrator to User ID input box. | |
| 1. The system administrator enters User ID of the user that needs to be deleted from the system. | |  | |
|  | | 1. The system administrator presses the submit button.   6. The system administrator enters his password. | | 5. The System asks for system administrator’s password.  7. The system validates the password entered by the system administrator.  8. The system validates if the user against entered user ID exists  9. The system then deletes the user and all his record from the database. | |
| **AlternativeFlows:** | | There is no specific alternative flow to this use case. | | | |
| **Exception:** | | Exception (Add valid User ID) is raised in case the system administrator enters invalid User ID.  Exception (Enter correct password) is raised in case the system administrator enters wrong password. | | | |

* + 1. **Change password(User)**

**Table 2.9 Usecase-6 (Change User’s Password)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **UseCase ID:** | UC6 | | | | |
| **UseCase Name:** | Change User’s Password | | | | |
| **CreatedBy:** | Anosh Zahid | | **LastUpdatedBy:** | | Anosh Zahid |
| **DateCreated:** | 4/18/2021 | | **LastRevisionDate:** | | 4/25/2021 |
| **Actors:** | | Project Manager / Team Lead / Team Member | | | |
| **Description:** | | The user (Project Manager / Team Lead / Team Member) will change his account password. | | | |
| **Trigger:** | | Click Change Password Button | | | |
| **Preconditions:** | | 1. The system must have online connectivity. 2. The user (Project Manager / Team Lead / Team Member) must be logged in. 3. The user (Project Manager / Team Lead / Team Member) must be on account settings. | | | |
| **Postconditions:** | | 1. The system shows a “password successfully changed” message on the same page. 2. The system redirects the user (Project Manager / Team Lead / Team Member) to the home page. | | | |
| **NormalFlow:** | | Actors | | System | |
| 1. The user (Project Manager / Team Lead / Team Member) clicks on change password button. | | 1. The System directs user (Project Manager / Team Lead / Team Member) to change password form. | |
| 1. The user (Project Manager / Team Lead / Team Member) enters existing password, new password, confirm new password. | |  | |
|  | | 1. The user (Project Manager / Team Lead / Team Member) presses the submit button. | | 1. The System validates if the user (Project Manager / Team Lead / Team Member) has entered correct existing password and new password matches confirm new password. 2. The system then displays “password successfully changed” message and redirects user (Project Manager / Team Lead / Team Member) to the home page. | |
| **AlternativeFlows:** | | The user can click on forgot password on the login page and change his password from there with the same procedure. | | | |
| **Exception:** | | Exception (Enter correct existing password/ Enter matching new password) is raised in case the user (Project Manager / Team Lead / Team Member) enters wrong existing password or wrong matching password. | | | |

* + 1. **Receive username and password via email(System Administrator)**

**Table 2.10 Usecase-7 (Receive Username and Password via Email)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **UseCase ID:** | UC7 | | | | |
| **UseCase Name:** | Receive username and password via email | | | | |
| **CreatedBy:** | Anosh Zahid | | **LastUpdatedBy:** | | Anosh Zahid |
| **DateCreated:** | 4/18/2021 | | **LastRevisionDate:** | | 4/25/2021 |
| **Actors:** | | System Administrator | | | |
| **Description:** | | The user (Project Manager / Team Lead / Team Member) will receive their password and username created by the system administrator via email. | | | |
| **Trigger:** | | Successful creation of a user account by the system administrator. | | | |
| **Preconditions:** | | 1. The system administrator must have created user accounts successfully. | | | |
| **Postconditions:** | | 1. The user (Project Manager / Team Lead / Team Member) receives their password and username on their email. | | | |
| **NormalFlow:** | | Actors | | System | |
|  | | 1. The System sends username and password to the user (Project Manager / Team Lead / Team Member) on his/her email. | |
|
|  | |
| **AlternativeFlows:** | | There is no alternative flow to this use case. | | | |
| **Exception:** | |  | | | |

* + 1. **View profile(User)**

**Table 2.11 Usecase-8 (View Profile)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **UseCase ID:** | UC8 | | | | |
| **UseCase Name:** | view Profile | | | | |
| **CreatedBy:** | Anosh Zahid | | **LastUpdatedBy:** | | Anosh Zahid |
| **DateCreated:** | 4/18/2021 | | **LastRevisionDate:** | | 4/25/2021 |
| **Actors:** | | Project Manager / Team Lead / Team Member | | | |
| **Description:** | | The user (Project Manager / Team Lead / Team Member) will view their user profile. | | | |
| **Trigger:** | | Click View profile Button | | | |
| **Preconditions:** | | 1. The system must have online connectivity. 2. The user (Project Manager / Team Lead / Team Member) must be logged in. 3. The user (Project Manager / Team Lead / Team Member) must have his home page displayed on the screen. | | | |
| **Postconditions:** | | 1. The system shows a graphical table with all the details of the user (Project Manager / Team Lead / Team Member) on the screen. | | | |
| **NormalFlow:** | | Actors | | System | |
| 1. The user (Project Manager / Team Lead / Team Member) clicks on view profile button. | | 1. The System displays user (Project Manager / Team Lead / Team Member) a graphical table with his/her details. | |
|
|  | |
| **AlternativeFlows:** | | There is no alternative flow to this use case. | | | |
| **Exception:** | |  | | | |

* + 1. **Update user profile(User)**

**Table 2.12 Usecase-9 (Update User Profile)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **UseCase ID:** | UC9 | | | | |
| **UseCase Name:** | Update User Profile | | | | |
| **CreatedBy:** | Anosh Zahid | | **LastUpdatedBy:** | | Anosh Zahid |
| **DateCreated:** | 4/18/2021 | | **LastRevisionDate:** | | 4/25/2021 |
| **Actors:** | | Project Manager / Team Lead / Team Member | | | |
| **Description:** | | The user (Project Manager / Team Lead / Team Member) will update information in their user profile. | | | |
| **Trigger:** | | Click Update profile Button | | | |
| **Preconditions:** | | 1. The system must have online connectivity. 2. The user (Project Manager / Team Lead / Team Member) must be logged in. 3. The user (Project Manager / Team Lead / Team Member) must have his home page displaying on the screen. | | | |
| **Postconditions:** | | 1. The system shows a “profile successfully updated” message on the same page. 2. The system redirects the user (Project Manager / Team Lead / Team Member) to the home page. | | | |
| **NormalFlow:** | | Actors | | System | |
| 1. The user (Project Manager / Team Lead / Team Member) clicks on update profile button. | | 1. The System directs user (Project Manager / Team Lead / Team Member) to profile information form. | |
| 1. The user (Project Manager / Team Lead / Team Member) enters updated information in the profile information form. | |  | |
|  | | 1. The user (Project Manager / Team Lead / Team Member) presses the submit button. | | 1. The System updates the user profile in the database. 2. The system then displays “profile successfully updated” message and redirects user (Project Manager / Team Lead / Team Member) to the home page. | |
| **AlternativeFlows:** | | There is no specific alternative flow to this use case. | | | |
| **Exception:** | | Exception (fill all the information fields) is raised in case the user (Project Manager / Team Lead / Team Member) leaves one of the information field boxes empty and clicks on submit button. | | | |

* + 1. **Project manager can view other members profile(Project manager)**

**Table 2.13 Usecase-10 (Project manager viewing user profiles)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **UseCase ID:** | UC10 | | | | |
| **UseCase Name:** | Project manager viewing user profiles | | | | |
| **CreatedBy:** | Anosh Zahid | | **LastUpdatedBy:** | | Anosh Zahid |
| **DateCreated:** | 4/18/2021 | | **LastRevisionDate:** | | 4/25/2021 |
| **Actors:** | | Project Manager | | | |
| **Description:** | | The Project Manager will view team members and team leads user profiles. | | | |
| **Trigger:** | | Click View User Profile Button | | | |
| **Preconditions:** | | 1. The system must have online connectivity. 2. The Project Manager must be logged in. 3. The Project Manager must be on view team page. | | | |
| **Postconditions:** | | 1. The system shows user profile information page. | | | |
| **NormalFlow:** | | Actors | | System | |
| 1. The Project Manager clicks on view user profile button. | | 1. The System directs Project Manager to a graphical table with selected user’s details. | |
|  | |  | |
| **AlternativeFlows:** | | There is no specific alternative flow to this use case. | | | |
| **Exception:** | |  | | | |

* + 1. **Team lead can view other members profile(Team lead)**

**Table 2.14 Usecase-11 (Team lead viewing User profiles)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **UseCase ID:** | UC11 | | | | |
| **UseCase Name:** | Team lead viewing user profiles | | | | |
| **CreatedBy:** | Anosh Zahid | | **LastUpdatedBy:** | | Anosh Zahid |
| **DateCreated:** | 4/18/2021 | | **LastRevisionDate:** | | 4/25/2021 |
| **Actors:** | | Team Lead | | | |
| **Description:** | | The team lead will view team members user profiles. | | | |
| **Trigger:** | | Click View User Profile Button | | | |
| **Preconditions:** | | 1. The system must have online connectivity. 2. The team lead must be logged in. 3. The team lead must be on view team page. | | | |
| **Postconditions:** | | 1. The system shows user profile information page. | | | |
| **NormalFlow:** | | Actors | | System | |
| 1. The team lead clicks on view user profile button. | | 1. The System directs team lead to a graphical table with selected user’s details. | |
|  | |  | |
| **AlternativeFlows:** | | There is no specific alternative flow to this use case. | | | |
| **Exception:** | |  | | | |

* + 1. **Add Projects(System Administrator)**

**Table 2.15 Usecase-12 (Add Projects)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **UseCase ID:** | UC12 | | | | |
| **UseCase Name:** | Add Projects | | | | |
| **CreatedBy:** | AnoshZahid | | **LastUpdatedBy:** | | Anosh Zahid |
| **DateCreated:** | 4/18/2021 | | **LastRevisionDate:** | | 4/25/2021 |
| **Actors:** | | System Administrator | | | |
| **Description:** | | The System Administrator will add new projects in the repository. | | | |
| **Trigger:** | | Click Add Project Button | | | |
| **Preconditions:** | | 1. The system administrator must be logged in. 2. The system administrator must be on the home page. | | | |
| **Postconditions:** | | 1. The system shows system administrator “successfully added” message. 2. The system adds the new project in the repository projects list. | | | |
| **NormalFlow:** | | Actors | | System | |
| 1. The system administrator clicks on add project button. | | 1. The System directs system administrator to add project page with all the details required to be uploaded for the project. | |
| 1. The system administrator uploads all the artifacts in right extensions. | | 1. The System validates if the uploaded artifacts are in correct extension. | |
|  | | 1. The system administrator clicks on submit button. | |  | |
|  | |  | |  | |
|  | |  | | 1. The System shows “successfully added” message and saves the project in the repository. | |
|  | |  | | . | |
| **AlternativeFlows:** | | There is no specific alternate flow to this use case. | | | |
| **Exception:** | | The system administrator uploads the project artifacts in a wrong extension/format then the system raises an alert message upload artifacts in correct extension/format before submitting. | | | |

* + 1. **Delete Projects(System Administrator)**

**Table 2.16 Usecase-13 (Delete Projects)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **UseCase ID:** | UC13 | | | | |
| **UseCase Name:** | Delete Projects | | | | |
| **CreatedBy:** | AnoshZahid | | **LastUpdatedBy:** | | Anosh Zahid |
| **DateCreated:** | 4/18/2021 | | **LastRevisionDate:** | | 4/25/2021 |
| **Actors:** | | System Administrator | | | |
| **Description:** | | The System Administrator will delete existing projects from the repository. | | | |
| **Trigger:** | | Click delete Project Button | | | |
| **Preconditions:** | | 1. The system administrator must be logged in. 2. The system administrator must be on the home page. | | | |
| **Postconditions:** | | 1. The system shows system administrator “successfully deleted” message. 2. The system deletes the selected project from the repository list. | | | |
| **NormalFlow:** | | Actors | | System | |
| 1. The system administrator clicks on delete project button. | | 1. The System directs system administrator to project repository list. | |
| 1. The system administrator selects the project that needs to be deleted. | |  | |
|  | | 1. The system administrator clicks on submit button. | |  | |
|  | |  | |  | |
|  | |  | | 1. The System shows “successfully deleted” message and deletes the project in the repository. | |
|  | |  | | . | |
| **AlternativeFlows:** | | There is no specific alternate flow to this use case. | | | |
| **Exception:** | |  | | | |

* + 1. **Search Projects(Project Manager)**

**Table 2.17 Usecase-14 (Search Projects)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **UseCase ID:** | UC14 | | | | |
| **UseCase Name:** | Search Projects | | | | |
| **CreatedBy:** | AnoshZahid | | **LastUpdatedBy:** | | Anosh Zahid |
| **DateCreated:** | 4/18/2021 | | **LastRevisionDate:** | | 4/25/2021 |
| **Actors:** | | Project manager | | | |
| **Description:** | | Project can search existing projects in the repository. | | | |
| **Trigger:** | | Click Search Project Button | | | |
| **Preconditions:** | | 1. The user must be logged in. 2. The user must be on the home page. | | | |
| **Postconditions:** | | 1. The system shows user project repository list. | | | |
| **NormalFlow:** | | Actors | | System | |
| 1. The user clicks on search project button. | | 1. The System directs user to repository list. | |
|  | |  | |  | |
|  | |  | | . | |
| **AlternativeFlows:** | | There is no specific alternate flow to this use case. | | | |
| **Exception:** | |  | | | |

* + 1. **Update Project artifacts(Project manager)**

**Table 2.18 Usecase-15 (Update Project Artifacts)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **UseCase ID:** | UC15 | | | | |
| **UseCase Name:** | Update Project Artifacts | | | | |
| **CreatedBy:** | Anosh Zahid | | **LastUpdatedBy:** | | Anosh Zahid |
| **DateCreated:** | 4/18/2021 | | **LastRevisionDate:** | | 4/25/2021 |
| **Actors:** | | Project Manager | | | |
| **Description:** | | Project Manager will update artifacts of existing projects in the repository. | | | |
| **Trigger:** | | Click Update Project Button | | | |
| **Preconditions:** | | 1. The Project manager must be logged in. 2. The project manager must be on the home page. | | | |
| **Postconditions:** | | 1. The system shows “updated successfully” message. 2. The system updates artifacts of the project in the repository list. | | | |
| **NormalFlow:** | | Actors | | System | |
| 1. The project manager clicks on update project button. | | 1. The System directs project manager to upload artifacts box. | |
|  | | 1. The project manager uploads new updated artifacts in the correct format/extension. | | 1. The system validates if the uploaded artifacts are in correct format/extension. | |
|  | |  | | .5. The system then shows a “updated successfully” message and updates the project in the project repository list. | |
| **AlternativeFlows:** | | There is no specific alternate flow to this use case. | | | |
| **Exception:** | | There is no such exception to this case. | | | |
|  | |  | | | |

* + 1. **Check similarity(Project manager)**

**Table 2.19 Usecase-16 (Check Similarity)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **UseCase ID:** | UC16 | | | | |
| **UseCase Name:** | Check Similarity | | | | |
| **CreatedBy:** | Anosh Zahid | | **LastUpdatedBy:** | | Anosh Zahid |
| **DateCreated:** | 4/18/2021 | | **LastRevisionDate:** | | 4/25/2021 |
| **Actors:** | | Project Manager | | | |
| **Description:** | | Project Manager will check similarity of existing projects and new projects from the project repository on the basis of functional requirements. | | | |
| **Trigger:** | | Click check similarity-FR Button | | | |
| **Preconditions:** | | 1. The Project manager must be logged in. 2. The project manager must be on the home page. | | | |
| **Postconditions:** | | 1. The system shows a percentage of similarity between projects. | | | |
| **NormalFlow:** | | Actors | | System | |
| 1. The project manager clicks on check similarity-FR button. | | 1. The System directs project manager to enter functional requirement input box. | |
|  | | 1. The project manager enters functional requirements. | | 1. The system matches functional requirements with functional requirements of existing projects in the project repository list. | |
|  | |  | | .5. The system then shows a percentage of similarity against each project. | |
| **AlternativeFlows:** | | There is no specific alternate flow to this use case. | | | |
| **Exception:** | | Exception (Enter functional requirements in correct format) is raised if the project manager enters functional requirements in wrong format. | | | |

* + 1. **Check Similarity(Project manager)**

**Table 2.20 Usecase-17 (Check Similarity)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **UseCase ID:** | UC17 | | | | |
| **UseCase Name:** | Check Similarity-Scope | | | | |
| **CreatedBy:** | Anosh Zahid | | **LastUpdatedBy:** | | Anosh Zahid |
| **DateCreated:** | 4/18/2021 | | **LastRevisionDate:** | | 4/25/2021 |
| **Actors:** | | Project Manager | | | |
| **Description:** | | Project Manager will check similarity of existing projects and new projects from the project repository on the basis of scope. | | | |
| **Trigger:** | | Click check similarity-scope Button | | | |
| **Preconditions:** | | 1. The Project manager must be logged in. 2. The project manager must be on the home page. | | | |
| **Postconditions:** | | 1. The system shows a percentage of similarity between projects. | | | |
| **NormalFlow:** | | Actors | | System | |
| 1. The project manager clicks on check similarity-scope button. | | 1. The System directs project manager to enter scope input box. | |
|  | | 1. The project manager enters scope. | | 1. The system matches scope with scope of existing projects in the project repository list. | |
|  | |  | | .5. The system then shows a percentage of similarity against each project. | |
| **AlternativeFlows:** | | There is no specific alternate flow to this use case. | | | |
| **Exception:** | | Exception (Enter scope in correct format) is raised if the project manager enters scope in wrong format. | | | |
|  | |  | | | |

* + 1. **Create new project (Project manager)**

**Table 2.21 Usecase-18 (Create new Project)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Use Case ID:** | UC18 | | | | |
| **Use Case Name:** | Create new project | | | | |
| **Created By:** | Emaan saqib | | **Last Updated By:** | | Emaan saqib |
| **Date Created:** | 4/24/2022 | | **Last Revision Date:** | | 4/24/2022 |
| **Actors:** | | Project manager | | | |
| **Description:** | | The Project manager shall be able to create new projects. Project manager will create teams and assign roles to the member accordingly. Project manager will assign the project to team leads so they can assign task to team members. Project manager and team leads can set deadlines of the tasks assigned. | | | |
| **Trigger:** | | Click on Create Project button | | | |
| **Preconditions:** | | Project manager should be log into the system. | | | |
| **Post conditions:** | | New project will be created. | | | |
| **Normal Flow:** | | Actors | | System | |
| 1. Project manager clicks on Create new project button. | | 1. System will display the form to create a new project on the screen. | |
| 1. Project manager will add name of the project, role of members, and will set deadlines of the project created. | |  | |
|  | | 1. Project manager will click on submit button to create the project. | | 1. New project will be created. | |
| **Alternative Flows:** | | There is no alternative flow for this case. | | | |
| **Exception:** | | Input fields are empty. | | | |

* + 1. **Browse projects (project manager)**

**Table 2.22 Usecase-19 (Browse projects)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Use Case ID:** | UC19 | | | | |
| **Use Case Name:** | Browse projects from projects list | | | | |
| **Created By:** | Emaan saqib | | **Last Updated By:** | | Emaan saqib |
| **Date Created:** | 4/18/2021 | | **Last Revision Date:** | | 4/21/2021 |
| **Actors:** | | Project manager | | | |
| **Description:** | | The Project manager shall be able to browse/search projects from the projects list.  Project manager can search the projects from ‘in progress projects’ assigned to the members. | | | |
| **Trigger:** | | Click on Search projects or Browse projects button | | | |
| **Preconditions:** | | 1. Project manager should be log into the system. 2. There should be projects available to search. | | | |
| **Post conditions:** | | All the ongoing projects will be displayed on the screen. | | | |
| **Normal Flow:** | | Actors | | System | |
| 1. Project manager will click on the search project button. | | 1. System will display the search box. | |
| 1. Project manager will write project names or category that it want to search from. | | 1. System will show all the results according to the search of the user entered. | |
| **Alternative Flows:** | | There is no specific alternate flow to this use case. | | | |
| **Exception:** | | No such project exist. | | | |

* + 1. **View assigned projects (Team lead)**

**Table 2.23 Usecase-20 (View assigned Projects)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Use Case ID:** | UC20 | | | | |
| **Use Case Name:** | View assigned projects | | | | |
| **Created By:** | Emaan saqib | | **Last Updated By:** | | Emaan saqib |
| **Date Created:** | 4/24/2022 | | **Last Revision Date:** | | 4/24/2022 |
| **Actors:** | | Team lead | | | |
| **Description:** | | Team leads can view the projects assigned by the project manager. Team lead can view the task related to it and the deadlines of the project as well. | | | |
| **Trigger:** | | Click on View assigned project button | | | |
| **Preconditions:** | | Project should be assigned to the team leads. | | | |
| **Post conditions:** | | Team members can view every detail regarding the project. | | | |
| **Normal Flow:** | | Actors | | System | |
| 1. Team lead will click on the projects button. | | 1. System will display assigned projects to the team lead. | |
| 1. Team lead can then view the assigned projects. | |  | |
| **Alternative Flows:** | | There is no such alternative flow for this use case. | | | |
| **Exception:** | | System not responding. | | | |

* + 1. **Submit project (Project manager)**

**Table 2.24 Usecase-21 (Submit Project)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Use Case ID:** | UC21 | | | | |
| **Use Case Name:** | Submit project to project manager. | | | | |
| **Created By:** | Emaan saqib | | **Last Updated By:** | | Emaan saqib |
| **Date Created:** | 4/24/2022 | | **Last Revision Date:** | | 4/24/2022 |
| **Actors:** | | Team lead and Project manager | | | |
| **Description:** | | Team lead can submit the tasks they have completed to project manager. | | | |
| **Trigger:** | | Click on submit task button. | | | |
| **Preconditions:** | | Project manager should assign projects to team leads. | | | |
| **Post conditions:** | | Team leads can view every detail regarding the project. | | | |
| **Normal Flow:** | | Actors | | System | |
| 1. Team lead will click on the view assigned project button. | | 1. System will show the project assigned. | |
| 1. Team lead can then click on submit button. | | 1. Project will be then submitted in the system. | |
| **Alternative Flows:** | | There is no such alternative flow for this use case. | | | |
| **Exception:** | | System not responding | | | |

* + 1. **Submit task (Team member)**

**Table 2.25 Usecase-22 (Submit Task)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Use Case ID:** | UC22 | | | | |
| **Use Case Name:** | Submit task to team lead. | | | | |
| **Created By:** | Emaan saqib | | **Last Updated By:** | | Emaan saqib |
| **Date Created:** | 4/24/2022 | | **Last Revision Date:** | | 4/24/2022 |
| **Actors:** | | Team member and team lead | | | |
| **Description:** | | Team member can submit the tasks they have completed to team lead. | | | |
| **Trigger:** | | Click on submit task button. | | | |
| **Preconditions:** | | Team leads should assign task to team members. | | | |
| **Post conditions:** | | Team members can view every detail regarding the project. | | | |
| **Normal Flow:** | | Actors | | System | |
| 1. Team member will click on the view assigned task button. | | 1. System will show the task assigned. | |
| 1. Team member can then click on submit button. | | 1. Task will be then submitted in the system. | |
| **Alternative Flows:** | | There is no such alternative flow for this use case. | | | |
| **Exception:** | | System not responding | | | |

* + 1. **Add project remarks (Project manager)**

**Table 2.26 Usecase-23 (Add Remarks)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Use Case ID:** | UC23 | | | | |
| **Use Case Name:** | Add remarks to the assigned projects to team leads. | | | | |
| **Created By:** | Emaan saqib | | **Last Updated By:** | | Emaan saqib |
| **Date Created:** | 4/24/2022 | | **Last Revision Date:** | | 4/24/2022 |
| **Actors:** | | Project manager | | | |
| **Description:** | | Project manager can add remarks on the projects that are assigned to team leader, once they have submitted it. | | | |
| **Trigger:** | | Click on add remarks button. | | | |
| **Preconditions:** | | Team lead should complete projects assigned to them. | | | |
| **Post conditions:** | | Team lead can see the remarks given to them. | | | |
| **Normal Flow:** | | Actors | | System | |
| 1. Project manager will click on the view assigned project button. | | 1. System will show the project assigned. | |
| 1. Project manager will click on the add remarks button. | | 1. System will display add remarks form on the screen. | |
|  | | 1. Project manager will then fill in the remarks form. | |  | |
|  | | 1. Project manager will click on the submit button. | | 1. System will display a message of remarks submitted successfully. | |
| **Alternative Flows:** | | There is no such alternative flow for this use case. | | | |
| **Exception:** | | System not responding | | | |

* + 1. **Add project remarks (Team lead)**

**Table 2.27 Usecase-24 (Add remarks)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Use Case ID:** | UC24 | | | | |
| **Use Case Name:** | Add remarks to the assigned tasked to team members. | | | | |
| **Created By:** | Emaan saqib | | **Last Updated By:** | | Emaan saqib |
| **Date Created:** | 4/24/2022 | | **Last Revision Date:** | | 4/24/2022 |
| **Actors:** | | Team lead | | | |
| **Description:** | | Team lead can add remarks on the tasks that are assigned to team member, once they have submitted it. | | | |
| **Trigger:** | | Click on add remarks button. | | | |
| **Preconditions:** | | Team member should complete tasks assigned to them. | | | |
| **Post conditions:** | | Team members can see remarks given to them. | | | |
| **Normal Flow:** | | Actors | | System | |
| 1. Team lead will click on the view assigned task button. | | 1. System will show the task assigned. | |
| 1. Team lead will click on the add remarks button. | | 1. System will display add remarks form on the screen. | |
|  | | 1. Team lead will then fill in the remarks form. | |  | |
|  | | 1. Team lead will click on the submit button. | | 1. System will display a message of remarks submitted successfully. | |
| **Alternative Flows:** | | There is no such alternative flow for this use case. | | | |
| **Exception:** | | System not responding | | | |

* + 1. **Project status report(User)**

**Table 2.28 Usecase-25 (Project Status Report)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **UseCase ID:** | UC25 | | | | |
| **UseCase Name:** | Project Status Report | | | | |
| **CreatedBy:** | Anosh Zahid | | **LastUpdatedBy:** | | Anosh Zahid |
| **DateCreated:** | 4/18/2021 | | **LastRevisionDate:** | | 4/25/2021 |
| **Actors:** | | All users | | | |
| **Description:** | | All users will view and have access to project status report. | | | |
| **Trigger:** | | Click view project status Button | | | |
| **Preconditions:** | | 1. The user must be logged in. 2. The user must be on the project status page. | | | |
| **Postconditions:** | | 1. The system shows a percentage of completion of each task of the projects. | | | |
| **NormalFlow:** | | Actors | | System | |
| 1. The user clicks on view project status button. | | 1. The System displays percentage of completion of each task of the project. | |
| **AlternativeFlows:** | | There is no specific alternate flow to this use case. | | | |
| **Exception:** | |  | | | |

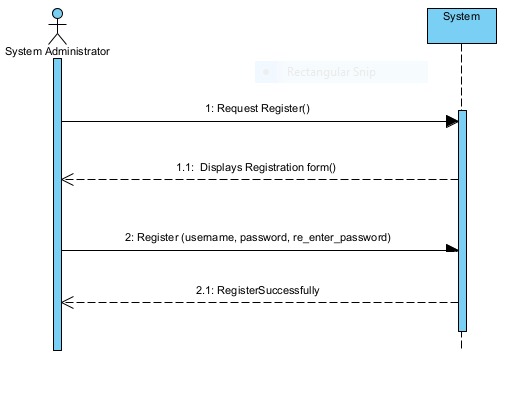
* + 1. **View Charts and Graphs(User)**

**Table 2.29 Usecase-26 (View Charts and Graphs)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **UseCase ID:** | UC26 | | | | |
| **UseCase Name:** | View charts and graphs | | | | |
| **CreatedBy:** | Anosh Zahid | | **LastUpdatedBy:** | | Anosh Zahid |
| **DateCreated:** | 4/18/2021 | | **LastRevisionDate:** | | 4/25/2021 |
| **Actors:** | | All users | | | |
| **Description:** | | All users will view and have access to charts and graphs showing project status. | | | |
| **Trigger:** | | Click view in graphical form Button | | | |
| **Preconditions:** | | 1. The user must be logged in. 2. The user must be on the project status page. | | | |
| **Postconditions:** | | 1. The system shows graphs and charts of each task of the projects related to their status of completion. | | | |
| **NormalFlow:** | | Actors | | System | |
| 1. The user clicks on view in graphical form button. | | 1. The System displays bar graphs and charts of completion of each task of the project. | |
| **AlternativeFlows:** | | There is no specific alternate flow to this use case. | | | |
| **Exception:** | |  | | | |

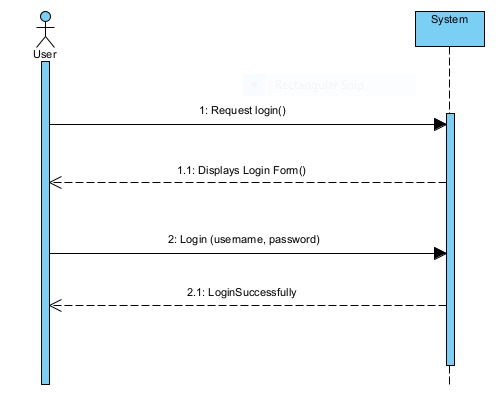
## System Sequence diagram

* + 1. **Register(System Administrator)**

****

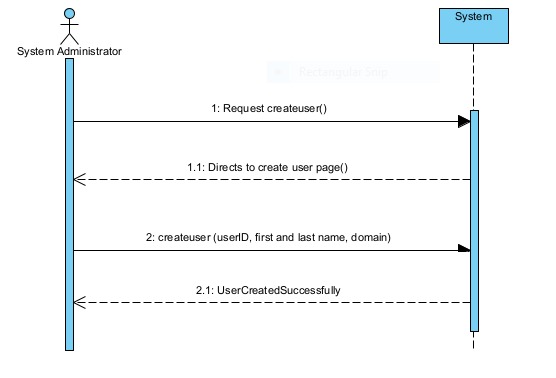
**Figure 2.2 register (admin)**

* + 1. **Login(User)**



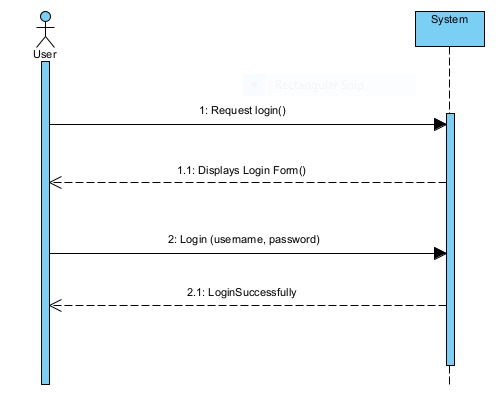
**Figure 2.3 login (admin)**

* + 1. **Create User(System Administrator)**



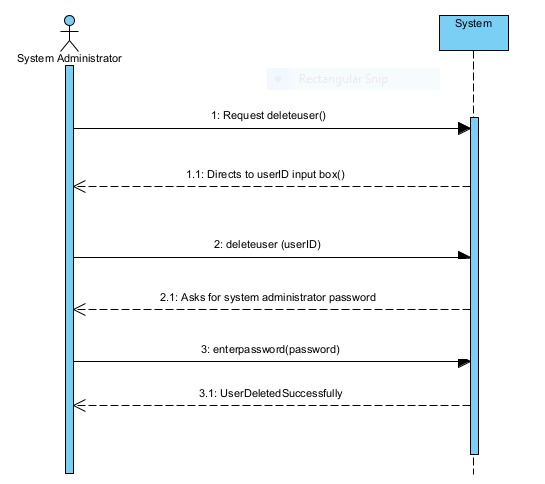
**Figure 2.4 Create User (admin)**

* + 1. **Login(User)**



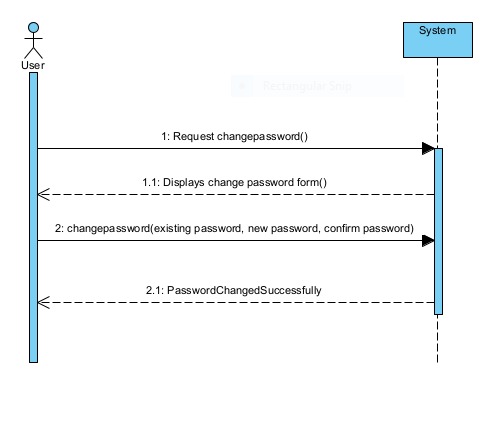
**Figure 2.5 Login (User)**

* + 1. **Delete(User)**

****

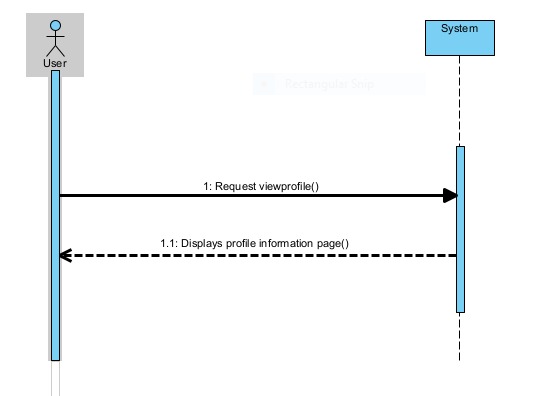
**Figure 2.6 Delete User (admin)**

* + 1. **Change password(User)**



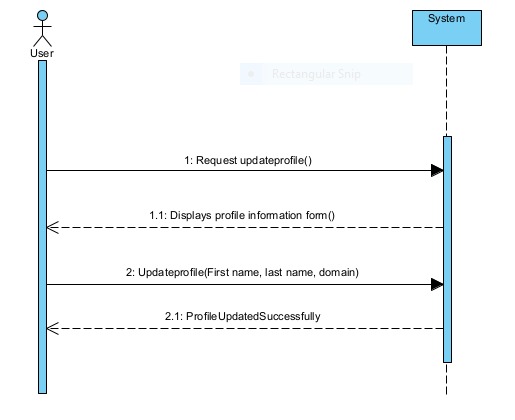
**Figure 2.8 Change Password (User)**

* + 1. **View profile(User)**



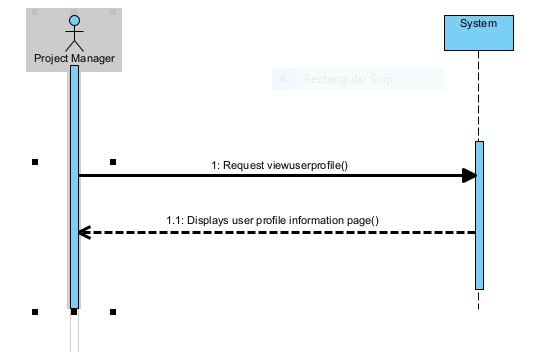
**Figure 2.9 View Profile (User)**

* + 1. **Update user profile(User)**



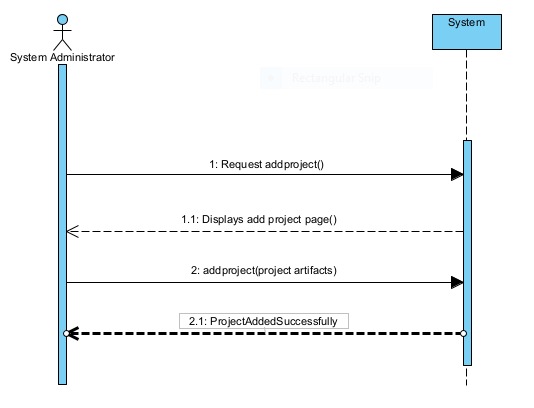
**Figure 2.10 Update Profile (user)**

* + 1. **Project manager can view other members profile(Project manager)**



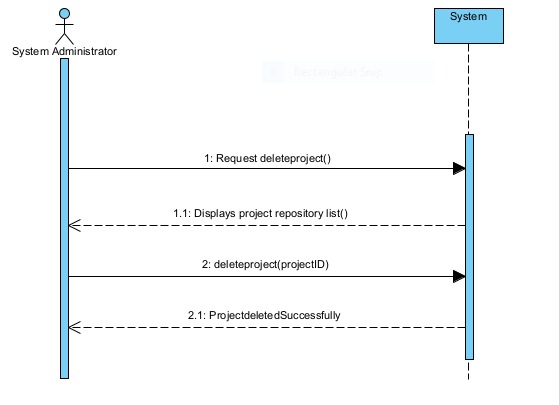
**Figure 2.11 View User Profile (Project Manager)**

* + 1. **Add Projects(System Administrator)**



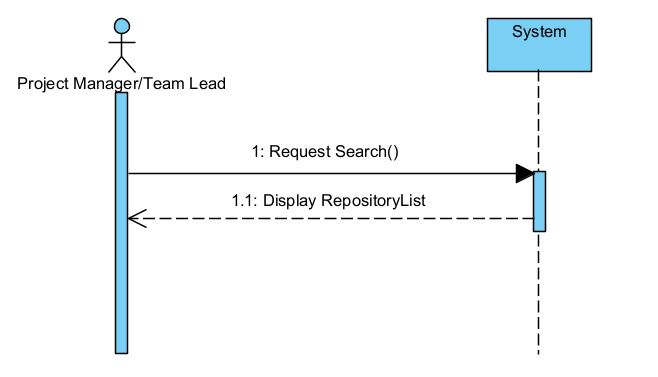
**Figure 2.12 Add Projects (admin)**

* + 1. **Delete Projects(System Administrator)**

****

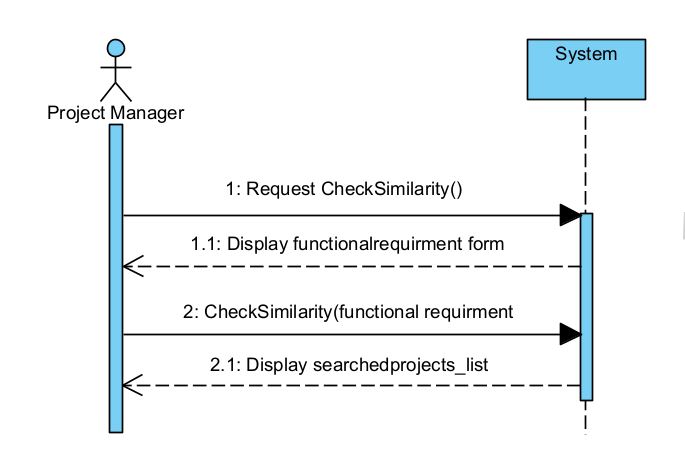
**Figure 2.13 delete project (admin)**

* + 1. **Search Projects(Project Manager)**



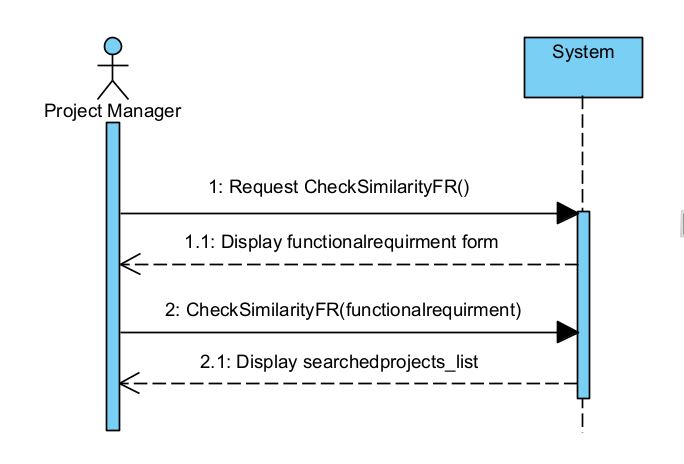
**Figure 2.14 Search Projects (Project Manager)**

* + 1. **Check similarity(Project manager)**

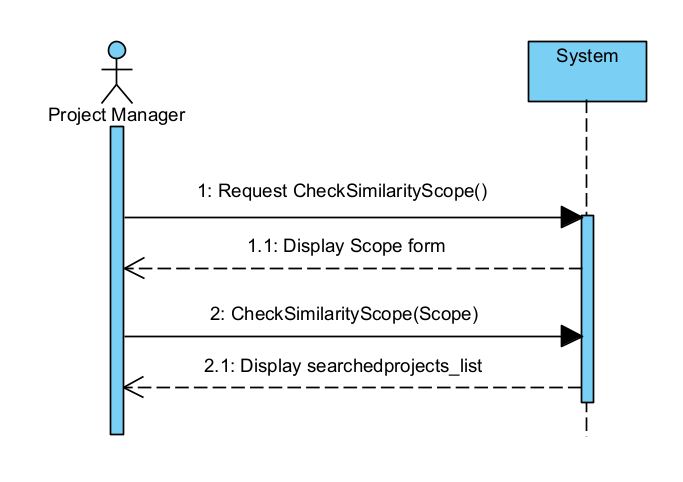


**Figure 2.15 Check similarity (Project Manager)**

* + 1. **Check Similarity by scope and functional requirement(Project manager)**

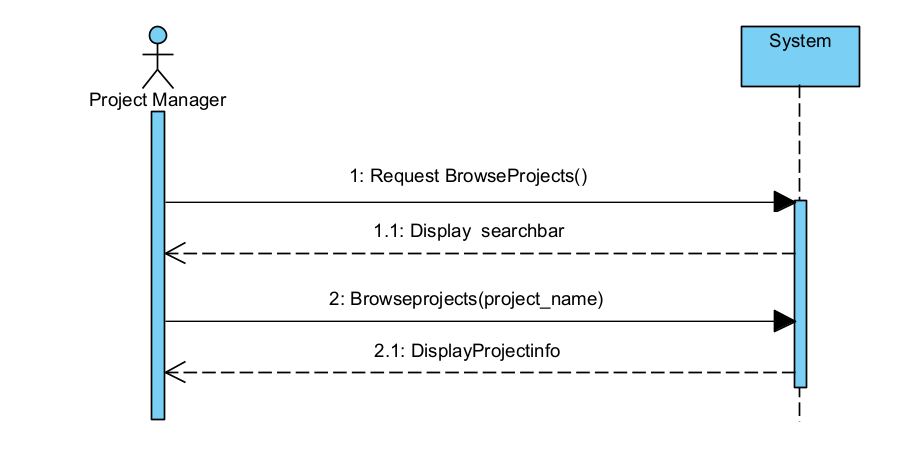


**Figure 2.16 Check similarity (Project Manager)**



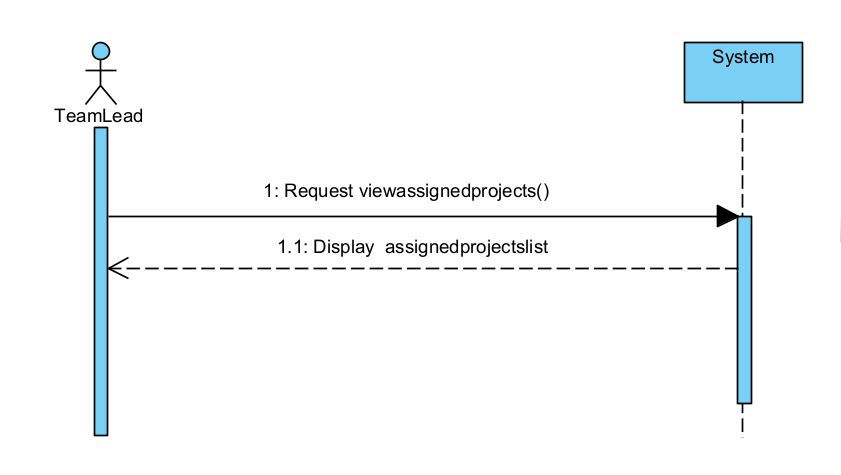
**Figure 2.17 Check Similarity (Project Manager)**

* + 1. **Browse projects (project manager)**



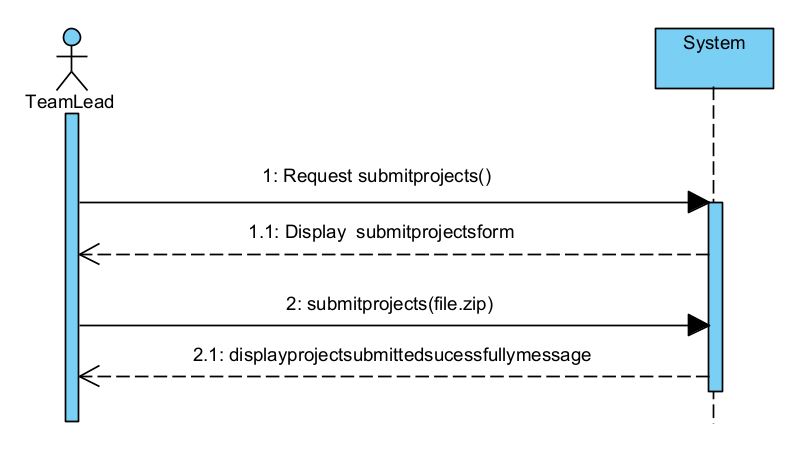
**Figure 2.18 Browse Projects (Project Manager)**

* + 1. **View assigned projects (Team lead)**



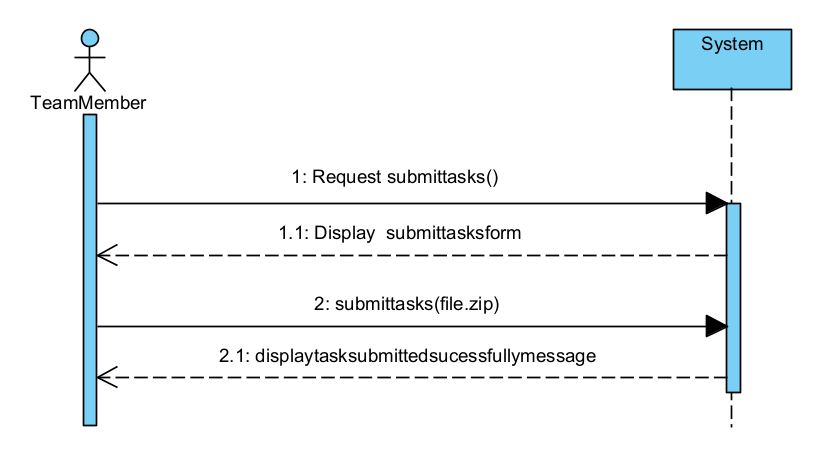
**Figure 2.19 View Assigned Project (Team Lead)**

* + 1. **Submit project (Project manager)**



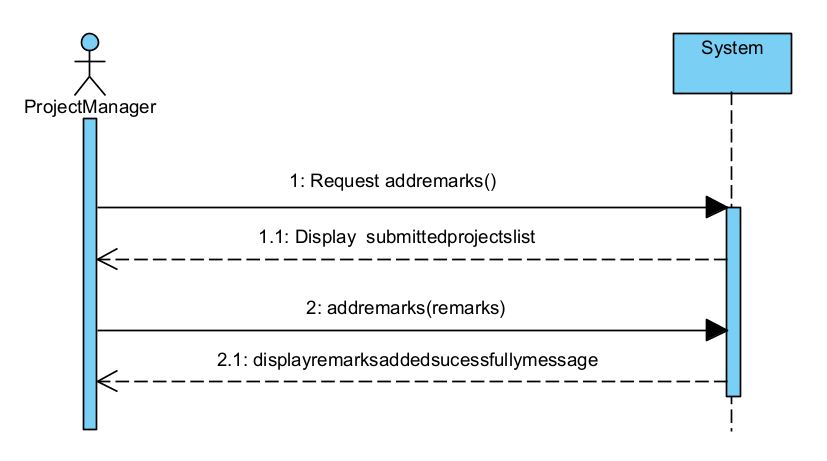
**Figure 2.20 Submit Project (Project manager)**

* + 1. **Submit task (Team member)**



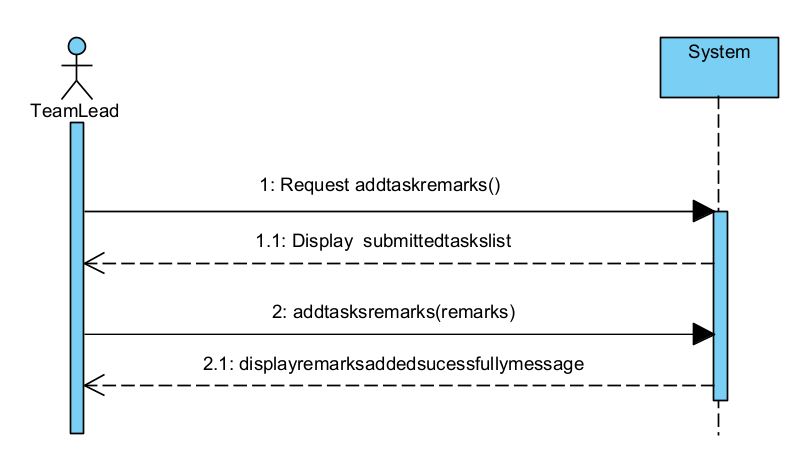
**Figure 2.21 Submit Task (Team Member)**

* + 1. **Add project remarks (Project manager)**



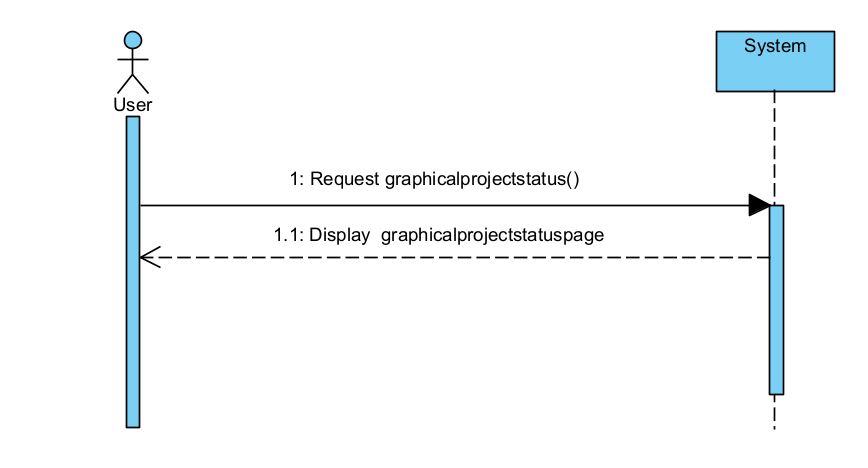
**Figure 2.22 Add remarks (Project Manager)**

* + 1. **Add project remarks (Team lead)**



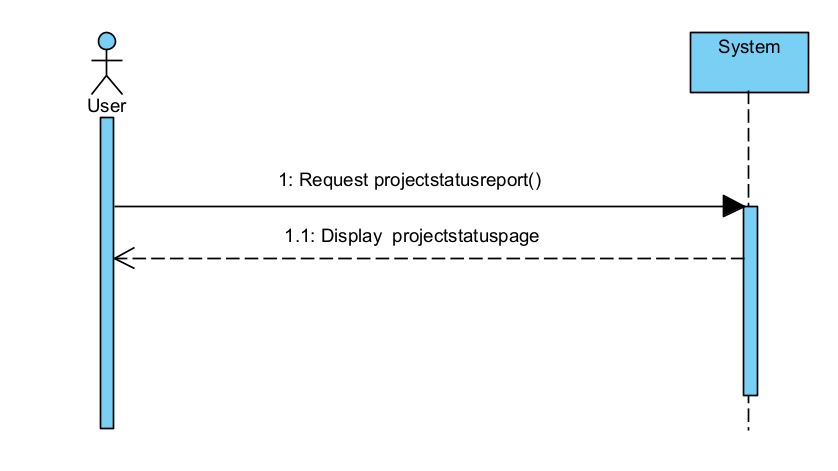
**Figure 2.24 Add remarks (Team Lead)**

* + 1. **Project status report(User)**



**Figure 2.25 Graphical Project Status (User)**

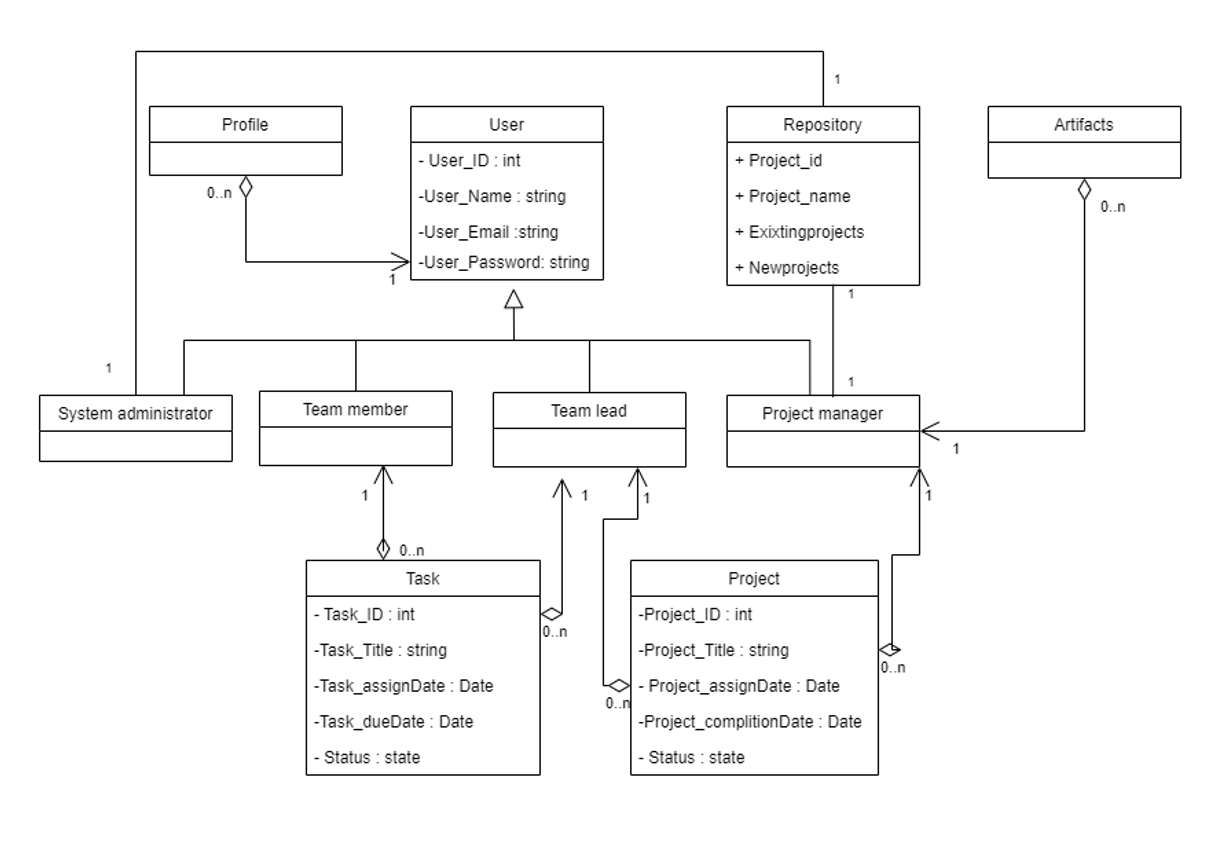
* + 1. **View Charts and Graphs(User)**

****

**Figure 2.27 View Charts and Graphs (User)**

## Domain Model

The basic concepts of the domain are User, Profile, Repository, Artifacts, System administrator, Team Lead, Team member, Project manager, Task and Project are shown in the following figure representing domain model.



**Figure 2.28 domain model**

# Chapter 3

# System Design

The purpose of this chapter is to provide information that is complementary to the development phase. Without an adequate design, that delivers required function as well as quality attributes, the project will fail. However, communicating architecture to its stakeholders is as important a job as creating it in the first place.

## Layer Definition

**Presentation Layer:**

Occupies the top level and displays information related to services available on a website. This tier communicates with other tiers by sending results to the browser and other tiers in the network.

**Business Logic Layer:**

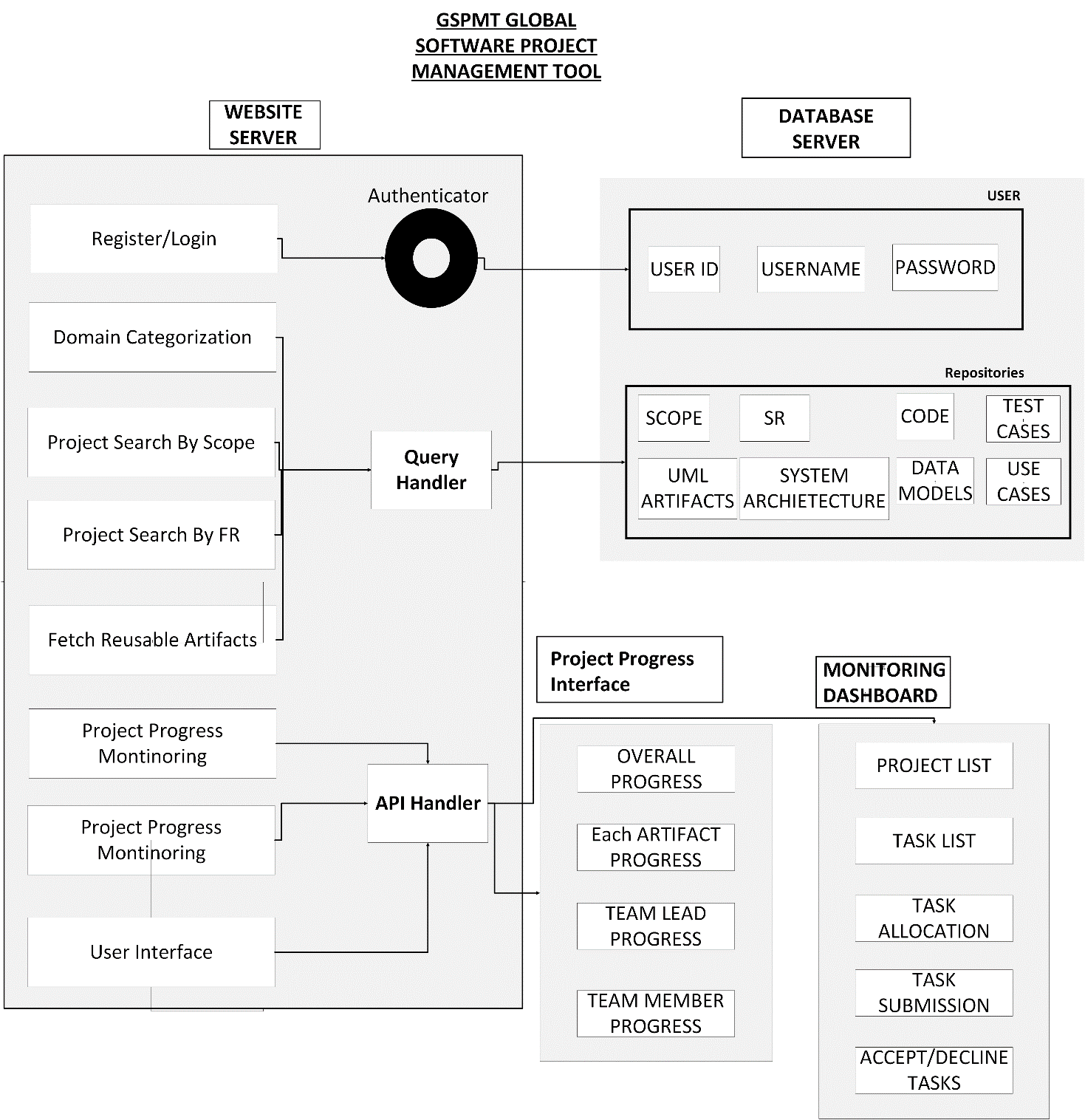
Application Layer also called the middle tier, logic tier, business logic or logic tier, this tier is pulled from the presentation tier. It controls application functionality by performing detailed processing.

**Database Layer:**

Database layer includes database servers where information is stored and retrieved. Data in this tier is kept independent of application servers or business logic.

## Software Architecture

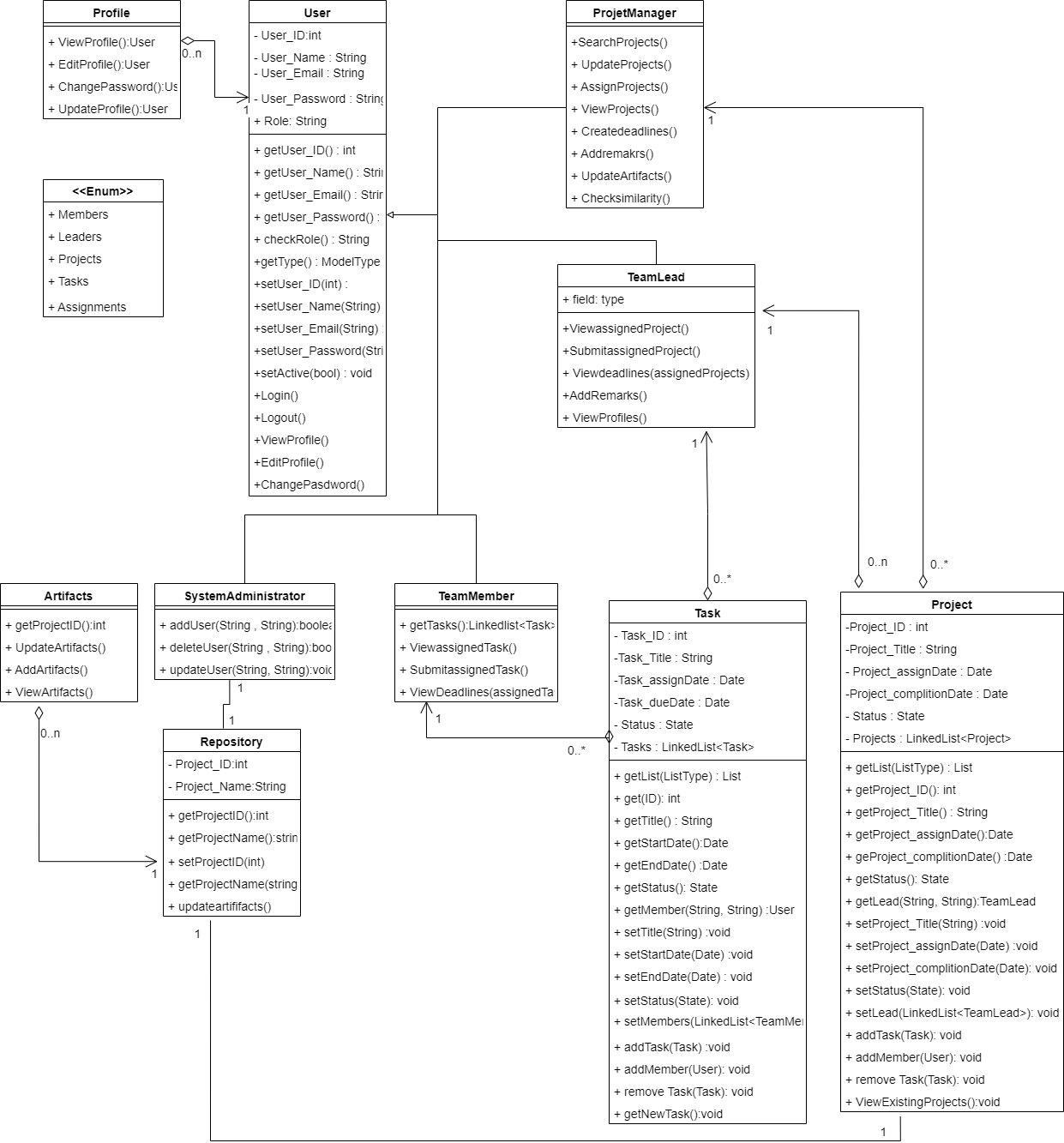
Software architecture is described as the organization or structure of a system, where the system represents a collection of components that accomplish a specific function or set of functions. Below is the architecture diagram of the system:



**Figure 3.1 architecture diagram**

## Class Diagram

The class diagram describes the attributes and operations of a class and the constraints imposed on the system. The class diagrams are widely used in the modeling of object oriented systems because they are the only UML diagrams, which can be mapped directly with object-oriented languages:

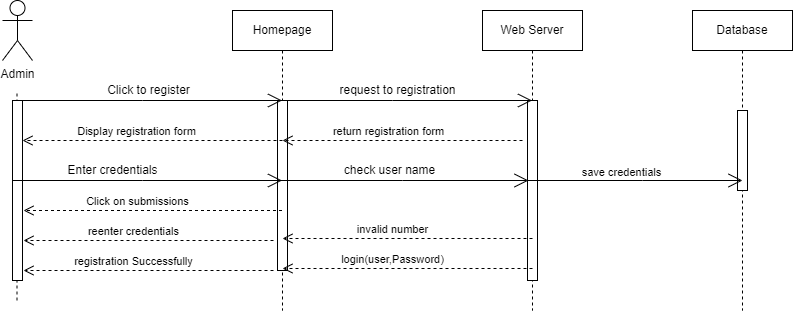


**Figure 3.2 class diagram**

## Sequence Diagram

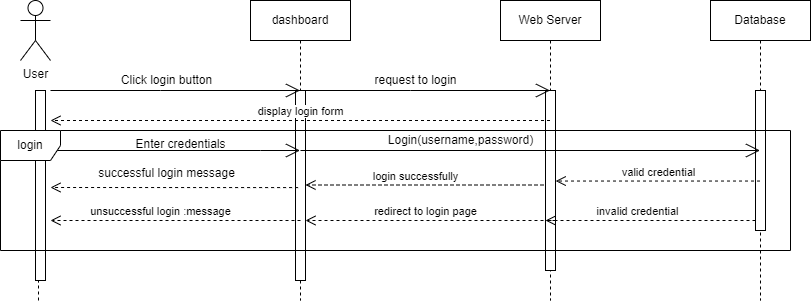
Sequence Diagram model the flow of logic within your system in a visual manner enabling you both to document and validate your logic, and are commonly used for both analysis and design purposes

* + 1. **Register (System Administrator)**



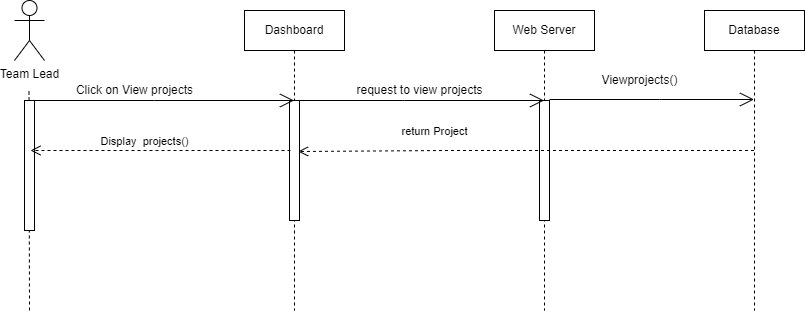
**Figure 3.3 SD-1**

* + 1. **Login (User)**



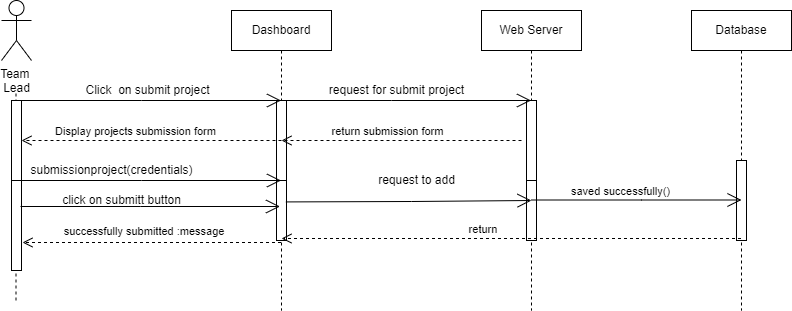
**Figure 3.4 SD-2**

* + 1. **View Projects (Team Lead)**



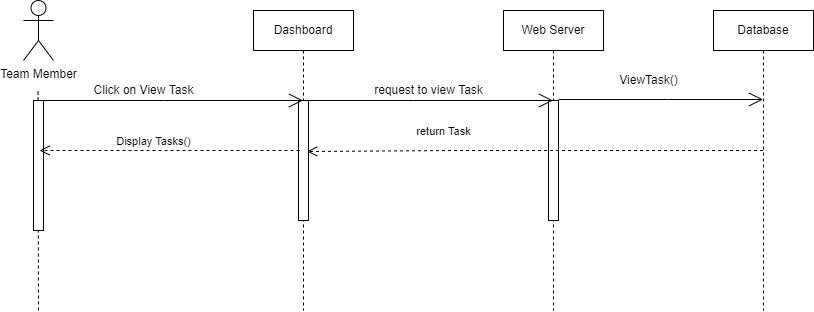
**Figure 3.5 SD-3**

* + 1. **Submit Project(Team Lead)**



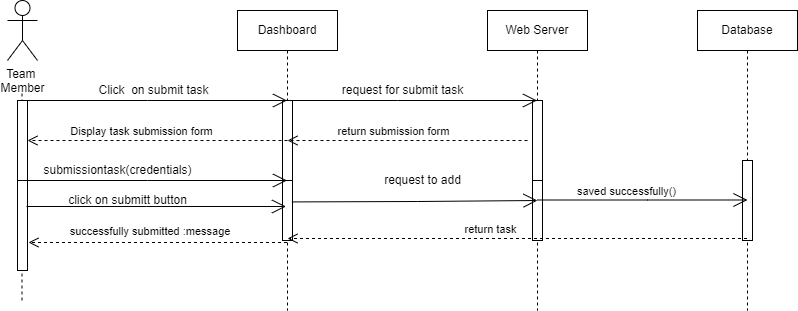
**Figure 3.6 SD-4**

* + 1. **View Task(Team Member)**



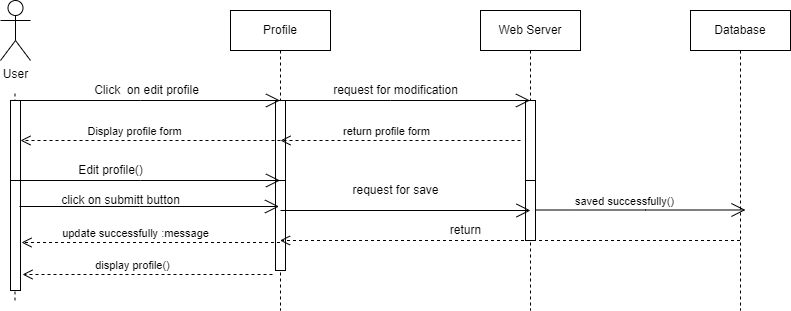
**Figure 3.7 SD-5**

* + 1. **Submit Task(Team Member)**



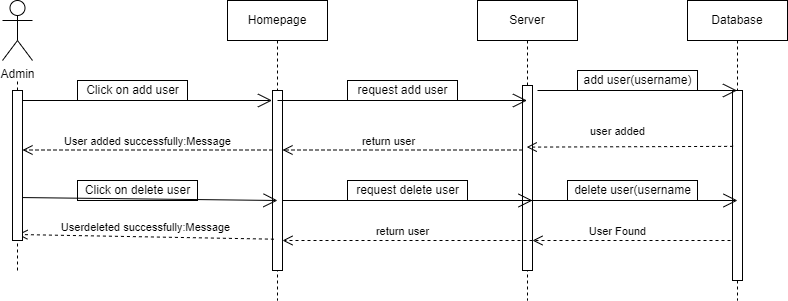
**Figure 3.8 SD-6**

* + 1. **Edit Profile (User)**



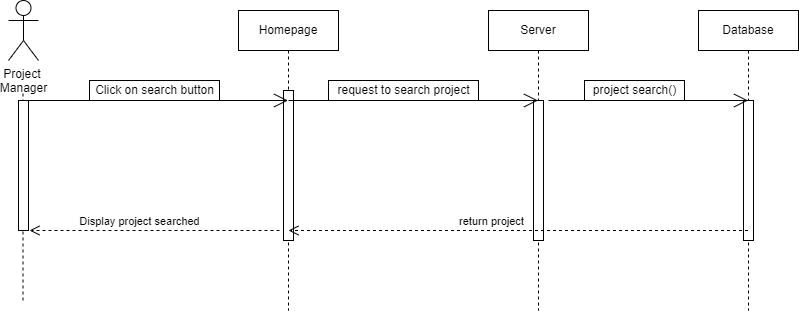
**Figure 3.9 SD-7**

* + 1. **Add User(System Administrator)**



**Figure 3.10 SD-8**

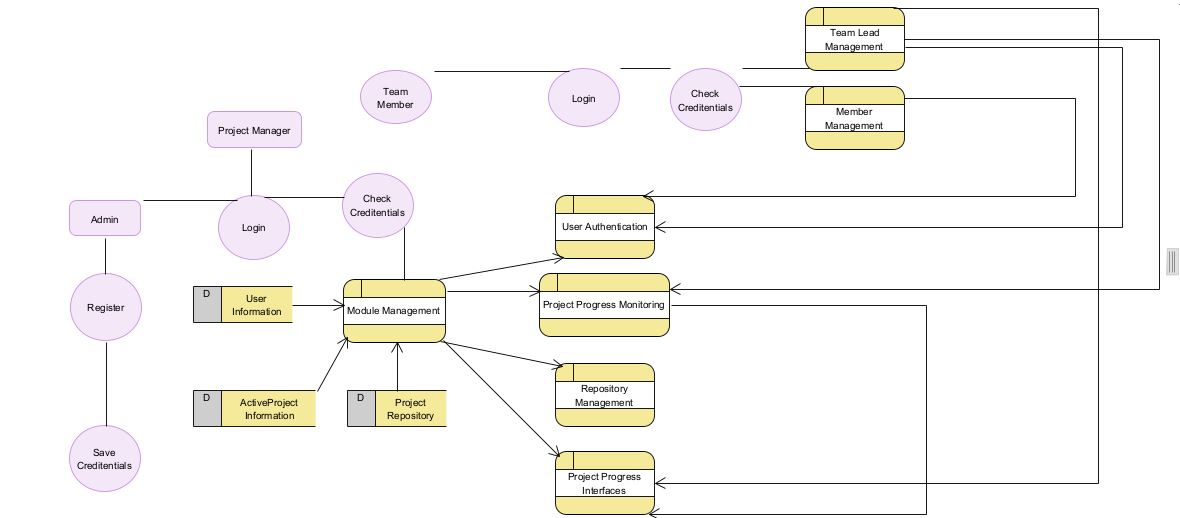
* + 1. **Search Projects(Project manager)**



**Figure 3.11 SD-9**

## Data Flow Diagram

Data-flow diagram is a way of representing a flow of data through a process or a system. The DFD also provides information about the inputs of each entity and the process itself. A data-flow diagram has no control flow — there are no decision rules and no loops.



## User Interface Design

