





Design Document Template

System Overview

Provide a brief description of the purpose of the system, its target users and the environment in which the system will be used.

The purpose of our system is to increase workplace efficiency by formatting projects into easy to understand task lists which can be implemented into larger projects encompassing multiple task lists. The target users of this system are generally software developers and tech companies, however this system can prove beneficial for companies in most fields.

References

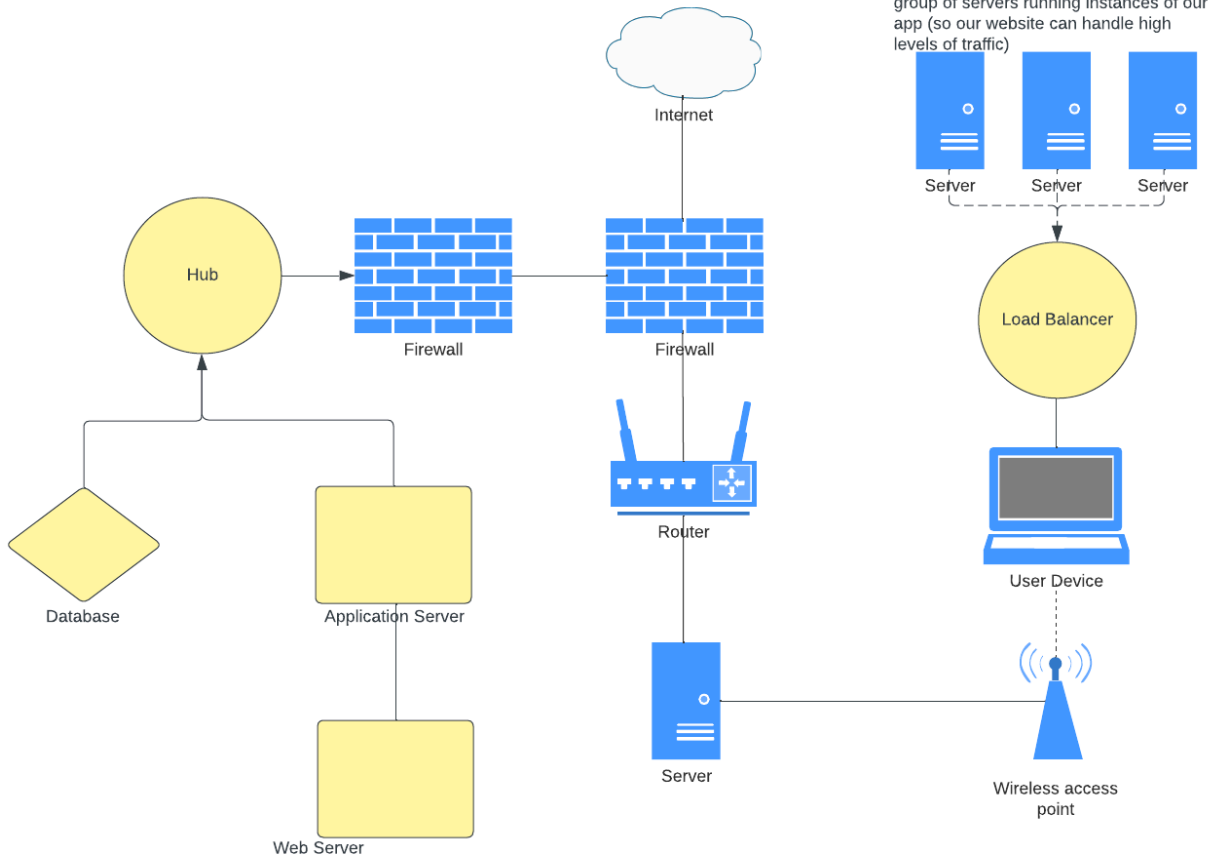
-  USE CASE DIAGRAM
-  Requirements
-  Nonfunctional Requirements
-  Task Management Website

Environment Overview

Network Diagram

NetworkDiagram

Vine Boom | October 8, 2023



The application we are developing is to be in a web based environment. It will reside on a remote server and delivered to the user's device via the internet; it will be accessed through a network. A library should be developed in order to simplify code management and make the project itself more reusable and maintainable. Here are some key aspects of the library to consider:

- Common Functionality
 - Different components should be reusable across different parts of the project manager application.
- API Client (if the project is to work with an API)
 - Creating a library for the API centralizes everything in terms of making requests, handling responses, and managing any necessary user authentication.
- UI Framework
 - Develop a library of UI components (buttons, forms, etc.) ensuring a consistent user experience
- Data Models/ Validation
 - To define data models and validation rules to ensure data integrity
- Utility Functions
 - A separate library for date/time manipulation, string formatting, and data transformation
- Testing
 - To streamline the testing process and make it easier to guarantee the product is functioning as intended

The library would be something that pays off in terms of maintaining the project and developing it efficiently, however it does come with the cost of time. The library would entail the necessary coding, documentation, design, testing, maintenance, and identifying the reusable components of the code.

Data Storage

The data will be stored in nine json files: userList.json, users.json, projects.json, projectManagementSystem.json, projectList.json, column.json, tasks.json, and comments.json, in order on this document.. Each json file holds an array of objects, and each object has a unique identifier. The user list json is the connection between the users json. The project management system json is the overarching system of the projects. The project list json is the connection between the users and projects. The task json is the connection between the tasks and the users. The column json is the connection between the physical columns on our application and the tasks themselves. The comments are the connection between the user and the comments they make on tasks.

```
[
  {
    "users": ["JSmith99", "SkyWalker33"]
  }
]
```

```
[
  {
    "userId": 1,
    "username": "JSmith99",
    "password": "asdf99",
    "firstName": "John",
    "lastName": "Smith",
    "email": "jsmith@gmail.com",
    "projects": ["project1", "project2"]
  },
  {
    "userId": 2,
    "username": "SkyWalker33",
    "password": "asdf33",
    "firstName": "Sara",
    "lastName": "Walker",
    "email": "swalker@gmail.com",
    "projects": ["project1", "project2"]
  }
]
```

```
[
  {
    "projectId": 1,
    "projectName": "project1",
    "foonotes": "",
    "comment": ["", ""],
    "owner": "JSmith99",
    "developers": ["JSmith99", "SkyWalker33"],
    "scrumMaster": "JSmith99"
  },
  {
    "projectId": 2,
    "projectName": "project2",
    "foonotes": "",
    "comment": ["", ""],
    "owner": "SkyWalker33",
    "developers": ["SkyWalker33", "JSmith99"],
    "scrumMaster": "SkyWalker33"
  }
]
```

```
[
  {
    "projects": ["project1", "project2"]
  }
]
```

```
[
  {
    "projectId": 1,
    "projectName": "project1",
    "fooonotes": "",
    "comment": ["", ""],
    "owner": "JSmith99",
    "developers": ["JSmith99", "SkyWalker33"],
    "scrumMaster": "JSmith99"
  },
  {
    "projectId": 2,
    "projectName": "project2",
    "fooonotes": "",
    "comment": ["", ""],
    "owner": "SkyWalker33",
    "developers": ["SkyWalker33", "JSmith99"],
    "scrumMaster": "SkyWalker33"
  }
]
```

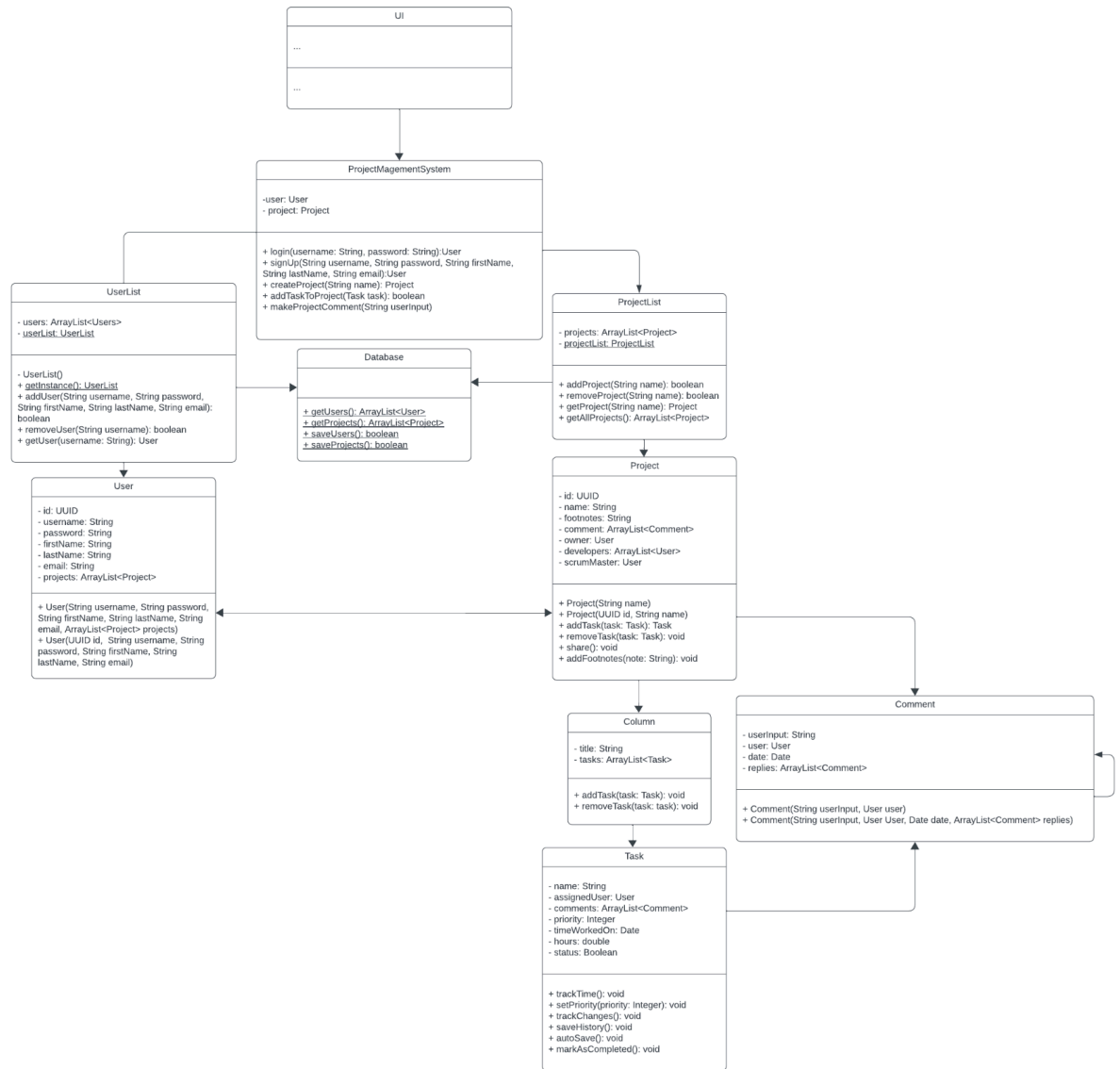
```
[
  {
    "user": "JSmith99",
    "project": "project1"
  },
  {
    "user": "SkyWalker33",
    "project": "project2"
  }
]
```

```
[
  {
    "title": "Task Column",
    "tasks": ["task1", "task2"]
  }
]
```

```
[
  {
    "taskId": 1,
    "userId": 1,
    "taskName": "task1",
    "assignedUser": "JSmith99",
    "comment": "",
    "priority": 2,
    "estimatedTimeWorked": "10/16/2023",
    "hours": 2.3,
    "finished": false
  },
  {
    "taskId": 2,
    "userId": 2,
    "taskName": "task2",
    "assignedUser": "SkyWalker33",
    "comment": "",
    "priority": 1,
    "estimatedTimeWorked": "10/20/2023",
    "hours": 4.6,
    "finished": false
  }
]
```

```
[
  {
    "userInput": "",
    "user": "JSmith99",
    "date": "10/16/2023",
    "replies": ["", ""]
  },
  {
    "userInput": "",
    "user": "SkyWalker33",
    "date": "10/20/2023",
    "replies": ["", ""]
  }
]
```

Class Diagram – Structural Design



Scenarios

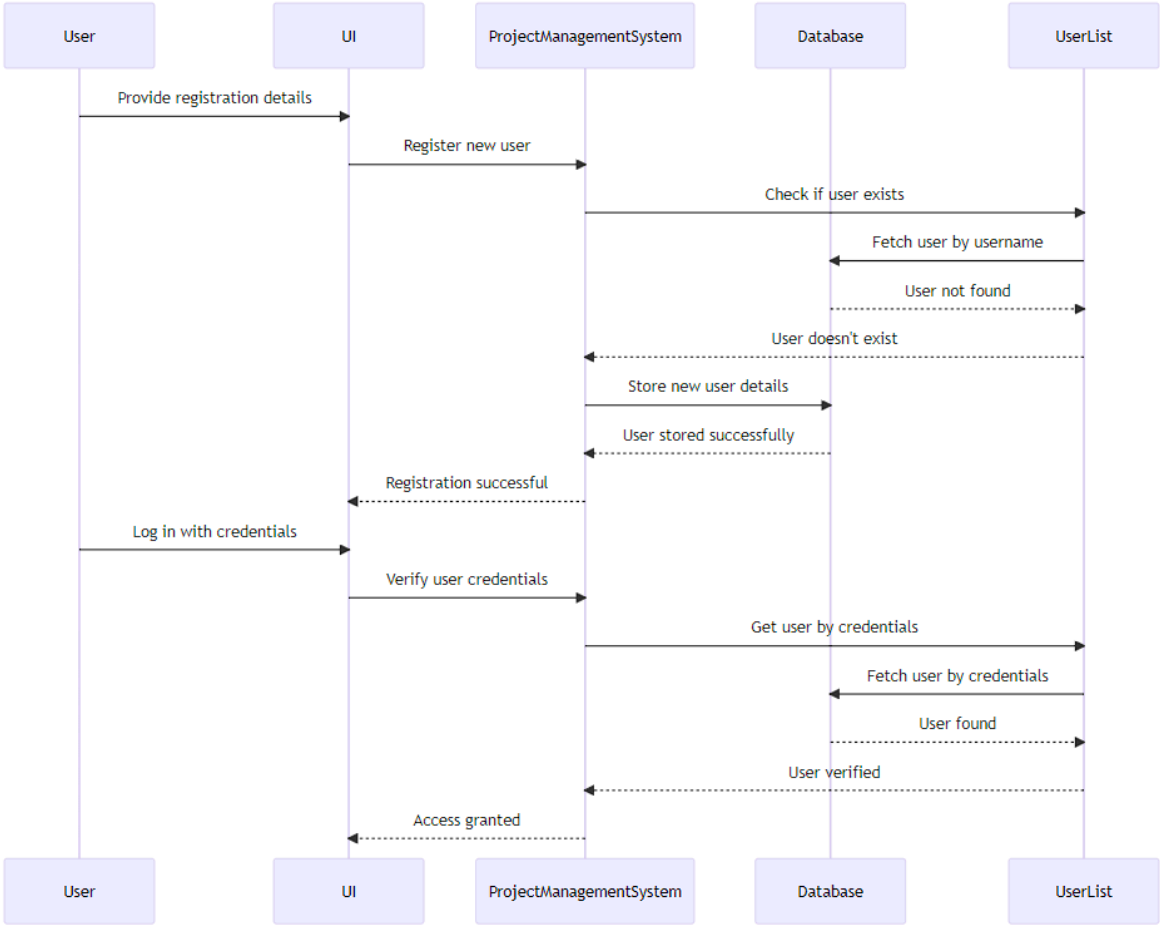
1. User Registration and Login:

Description: A new user wants to register to the ProjectManagementSystem. After registration, the user logs in.

Steps:

- User provides registration details.
- The system checks if a user with the given username already exists.
- If not, the system stores the new user's details in the database.
- The user then logs in using the provided credentials.
- The system verifies the credentials and allows/denies access.

Diagram:



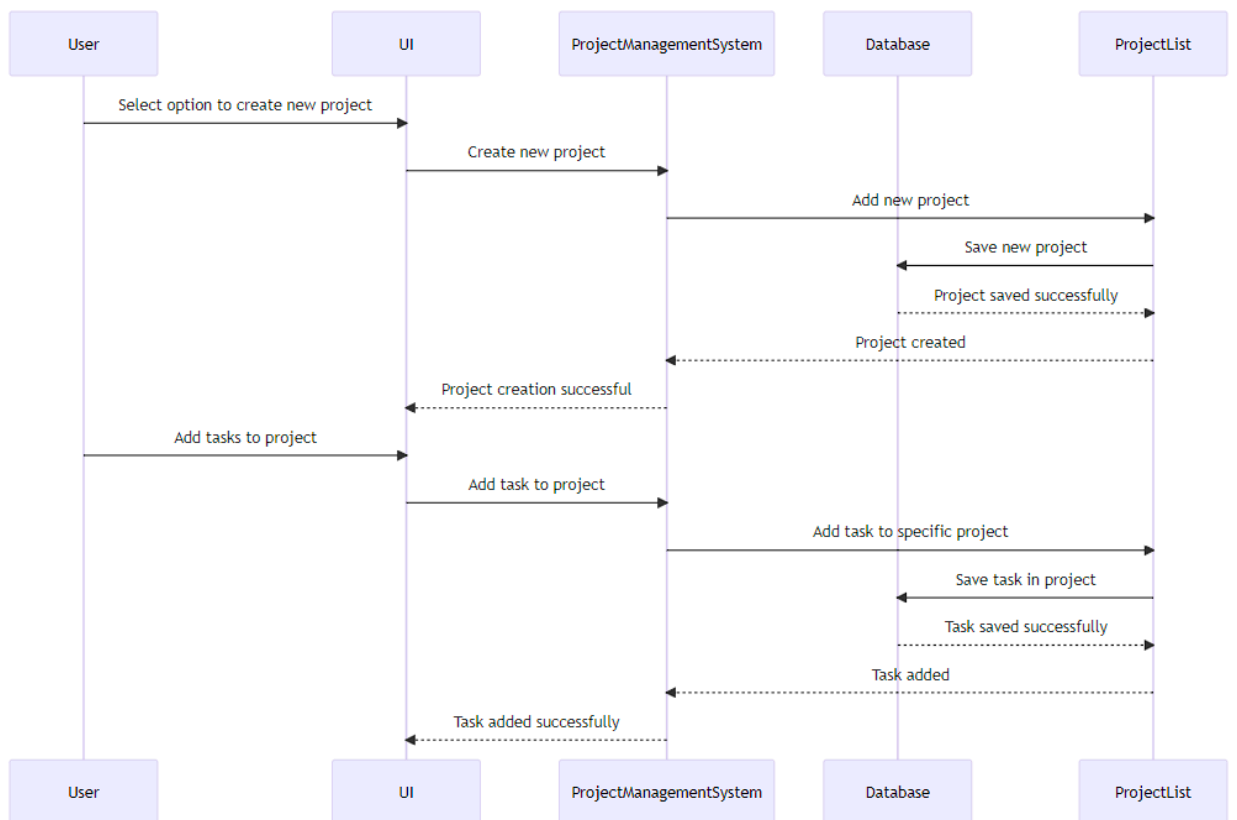
2. Project Creation and Task Assignment:

Description: An existing user logs into the ProjectManagementSystem. The user creates a new project and adds tasks.

Steps:

- User selects the option to create a new project.
- The system prompts for project details.
- The user provides the details, and the system saves the project.
- The user then adds tasks to the project.
- Each task is saved in the respective project.
- User Commenting on Task:

Diagram:



3. Commenting on a user's task

Description: An existing user logs into the ProjectManagementSystem, selects an existing project, and comments on a task.

Steps:

- User navigates to a particular project.
- User selects a task from the project.
- The system provides an option to comment on the task.
- The user writes a comment and submits it.
- The system saves the comment associated with the task.
- Project Sharing and Collaboration:

Diagram:



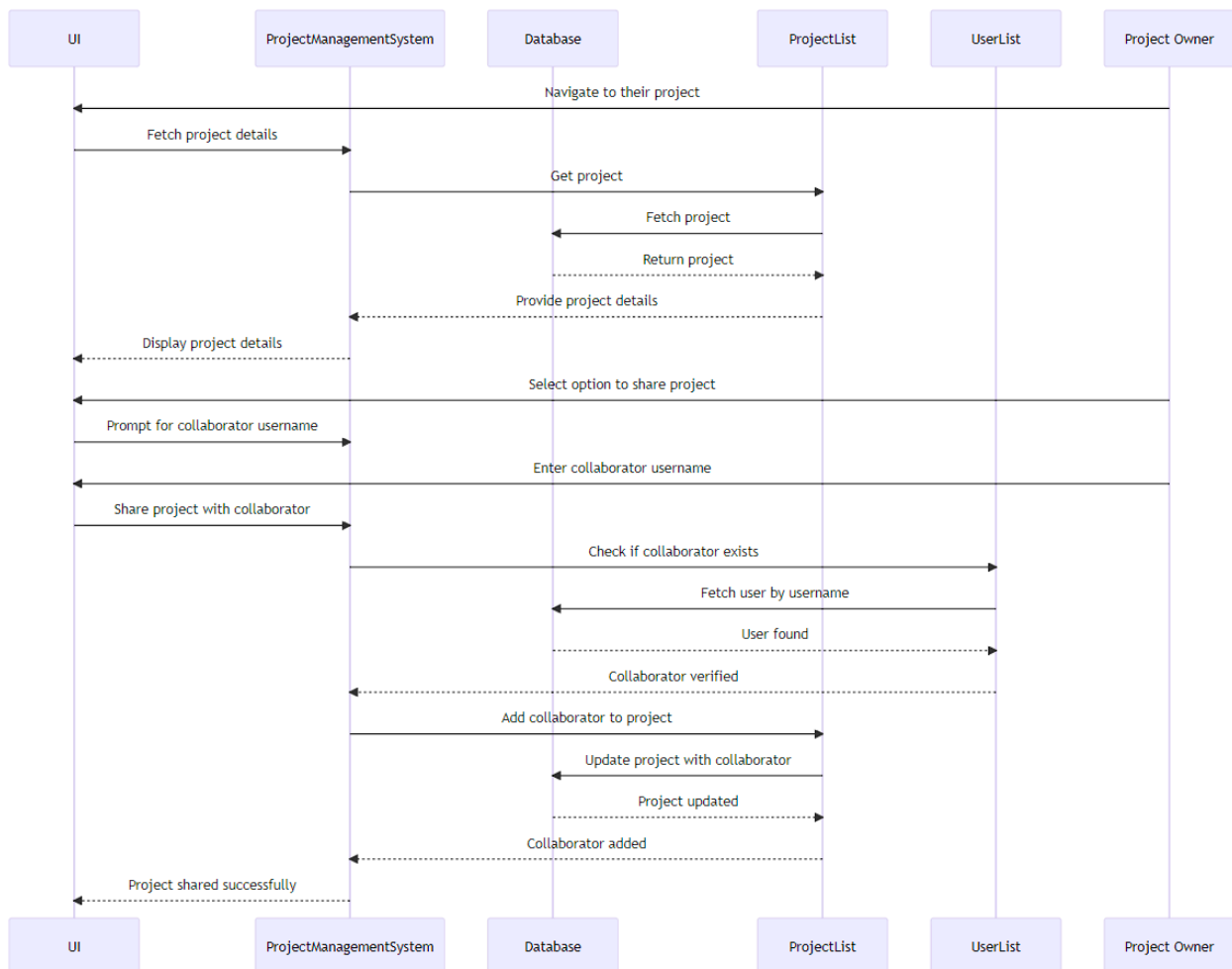
4. Sharing a project

Description: A user wants to share a project with another user, enabling collaboration.

Steps:

- The project owner navigates to their project.
- The system provides an option to share the project.
- The project owner enters the username of the collaborator.
- The system adds the collaborator to the project and provides them access rights.

Diagram:



5. Project Footnotes Addition:

Description: A user wants to add footnotes or additional notes to a project.

Steps:

- The user selects a project.
- The system provides an option to add footnotes.
- The user enters the footnotes and submits them.
- The system saves the footnotes associated with the project.

Diagram:

