**A picture containing text, font, logo, screenshot

Description automatically generated**

**Software System Report: Team Vault – Collaborative Project Management System**

**Date: June 21, 2023**

**Author(s): Jaleel Rogers**

**Organization: FileGuard Industries**

**4700 Research Way**

**Lakeland, FL 33805**

## System Functionality

## Problem Definition

### Affected Aspects of the Organization

This sub-section can provide the following information based on the client’s requirements. Understanding the impacts of developing Team Vault is a primary goal of FileGuard Industries.

### Effects and Possible Causes

It is impossible to make a perfect software system but an ideal one that fits the client and organization’s wants and needs. In this regard, software and other possible current implementations implement these needs and wishes.

### Operating Constraints

Operating Constraints will specify constraints involving the software system running on different operating system platforms, security, and performance.

## Goals

Team Vault will be a program communicating with users via a database class.

### File Upload Functionality

The following files users upload will be primarily source code, images, and documents.

### View and Download Files

Users have privileges allowing, at minimum, to view and download files.

### Commenting Functionality

Users can leave comments.

### Bug Reporting Feature

Users can report bugs in the files.

### User Roles and Permissions

Administrators must add and remove users, create projects, and add and remove posts, comments, or bugs. A specific project will assign Project Managers. Moreover, they can add and remove posts, comments, or bugs. One or more projects will assign Normal Users. In addition, they can add posts, comments, or bugs.

## The Proposed Solution

### System Objectives

The objectives of Team Vault are to provide users was a safe form of communicating in projects.

### Primary and Secondary Objectives

Identifies the primary and secondary objects of the software system based on the client’s requirements and the desired outcomes. These objectives will reflect the goals and functionalities specified by the client.

### Addressing End Users’ Needs

How the proposed solution addresses the needs of end users, considering the diverse roles and functionalities required.

## Feasibility

### Technical Feasibility

Consider hardware, operating systems, network configurations, and software dependencies. Assess whether the system can handle increasing user load, file sizes, and concurrent interactions. Consider system resources, database performance, caching mechanisms, and network bandwidth. Consider file transfer protocols, storage capacity, security measures, and backup mechanisms. Determine if the proposed solution provides a reliable and efficient user tool to upload, store, and retrieve files.

## Collaborative File Management System with Role-Based Access Control

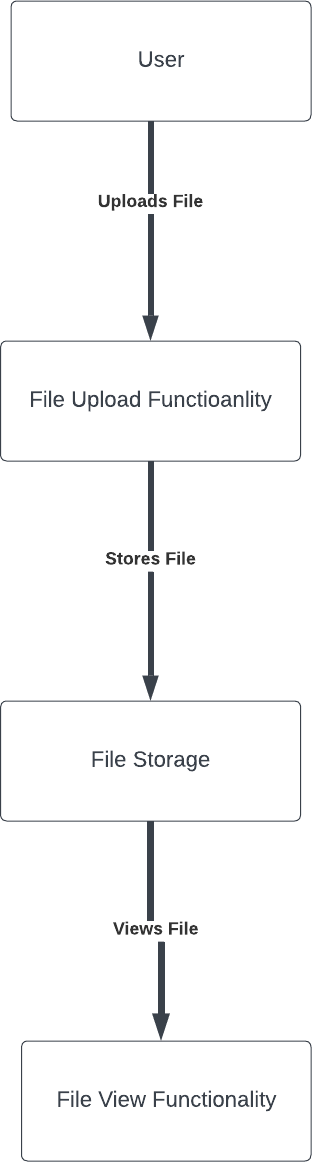
## Acknowledgement

FileGuard Industries sincerely appreciates the invaluable contributions made by our fellow team members toward the success of this project. We want to extend our gratitude to the team manager for his strong leadership and guidance and the unwavering support from our company's stakeholders. Additionally, we are grateful for the consistent feedback provided by our esteemed client, which greatly influenced FileGuard Industries in developing Team Vault as user-friendly software.

## Problem Synopsis

The purpose of Team Vault is to provide a platform where users can upload various types of files, including source code, images, and documents, within projects. The system will allow other users to view and download these files. Additionally, users will have the ability to leave comments and report bugs within projects. The system will implement a role-based access control mechanism with three distinct roles: Administrators, Project Managers, and Normal Users.

### File Upload Functionality



***Figure 1 - OFD for Process of User Uploading File***

As shown in Figure 1, the process for uploading files will enable users to store various file formats within the database, including source code files, images, and documents. This functionality ensures efficient storage and retrieval of files in Team Vault.

### View and Download Files

All users will be privileged to view and download files uploaded by others. However, access to these files will vary based on the user's role and project assignment. This approach facilitates easy file sharing while maintaining user privacy and security.

### Commenting Functionality

Users can leave comments, provide feedback, suggestions, or ask questions. These comments will be visible to other users in their projects, fostering collaboration, discussion, and effective communication.

### Bug Reporting Feature

Team Vault will include a bug-reporting feature that allows users to report encountered issues. Detailed bug reports, including relevant information for debugging and problem resolution, will assist in identifying and promptly resolving these issues.

### User Roles and Permissions

Team Vault will implement a role-based access control mechanism to manage user permissions and restrict access to certain functionalities based on their roles. Administrators will have elevated privileges and responsibilities within the system. They can add and remove users and create and delete projects, posts, comments, and bugs. Additionally, administrators can assign Project Managers to specific projects and Normal Users to one or more projects, ensuring effective system management and control. Project Managers will be assigned to specific projects and have the authority to add and remove posts, comments, and bugs within their assigned projects.

This role facilitates efficient project management and coordination while maintaining appropriate privileges. Normal Users will be assigned to one or more projects and can add posts, comments, and bugs within their given project(s). Although they have limited administrative rights, Normal Users play a vital role in contributing to projects and providing valuable feedback.

## System Overview

### Project Management Flow

**Figure 2 – Activity Diagram for User**

### User Management

Team Vault will have three types of users: Normal Users, Project Managers, and Administrators, each with varying privileges. Upon starting Team Vault, users enter their username and password. If they are not registered users, they will have the option to create an account. The program will verify the user's account and handle cases where the user is not in the system.

### File Management

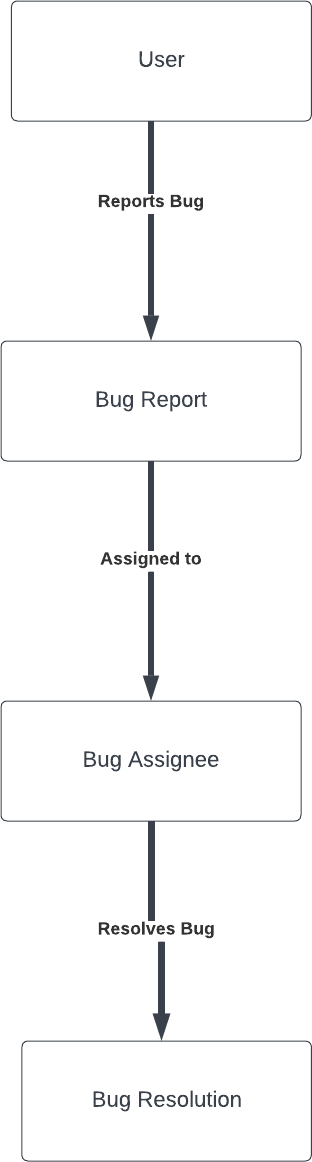
Users can upload and view different types of files in Team Vault. Source code files are text files with the extension ".txt," documents will be in formats such as ".pdf," ".doc," and ".docx," and

images will use the extensions ".jpeg" and ".png." Limiting file extensions is based on reducing the risk of malicious file uploads, considering the popularity and enterprise usage of the specified file formats, and enhancing the user experience by reducing complexity. Users will have the ability to preview files before downloading them.

### Commenting/Posting

Users can create posts that are visible to others, and comments are the replies to these posts. The maximum character limit for comments and posts will be 30,000 characters, including markdown and any other formatting. This limit will accommodate complex questions, statements, instructions, or briefings that users may need to communicate. Posts and comments will display the user, timestamp, and message date.

### Bug Reporting



***Figure 3 – OFD for Process of Bug Reporting***

The bug reporting functionality allows users to report issues within Team Vault. When requesting to make a bug report (BR), users will receive a prompt to enter their device's operating system, provide a description with a character limit of 30,000, enter a title with a limit of 50 characters, and have the option to add a picture. The process for handling bug reports in

Figure 3 depicts the steps involved in processing a BR, assigning users to address it, and marking it as resolved upon completion.

### Data Management

Team Vault will utilize the MySQL DBMS to manage all stored data, including user accounts, posts, comments, bug reports, and files.

## System Components

### User Management Component

Before the user can access Team Vault’s features, a prompt to log in to the system with a username and password. If the user has no account, they will create one with their username and password. User credentials are stored securely in the software system’s database for already established users to enter their credentials to gain. Initially, in the database, there will be an Administrator. When users successfully register, they will, by default, be given the Normal User role. If the user is an Administrator, they can assign roles to other users, such as converting a Normal User into a Project Manager. Users will only have one and only one role. The Administrator is the only role that can assign and change roles. Users will be able to verify their roles by going to their profiles. In addition, the User Profile will allow users to log out, add a name, and change passwords. All passwords must be 12 characters long with at least an uppercase letter, a lowercase letter, a symbol, and a number. Conditions are passable if Team Vault notifies users of their password's strength and verifies it in the database.

### File Management Component

The maximum size of the file users can upload is 3MB (Megabytes). This file size is because most documents (txt, pdf, doc,docx) average 4KB (Kilobytes)-10KB per page. The database will store all files under the user who post them. All users can view and download other users' file uploads as long as both users are on the same project.

### Commenting/Posting Component

Users can create comments, allowing other users to respond to the comments by making posts. Comments and posts by a user will be in the system’s database. Users will have access to a log of their comments and posts. The database will store up to 50 posts and comments before deleting the oldest comment or post to maximize storage efficiency.

### Bug Reporting Component

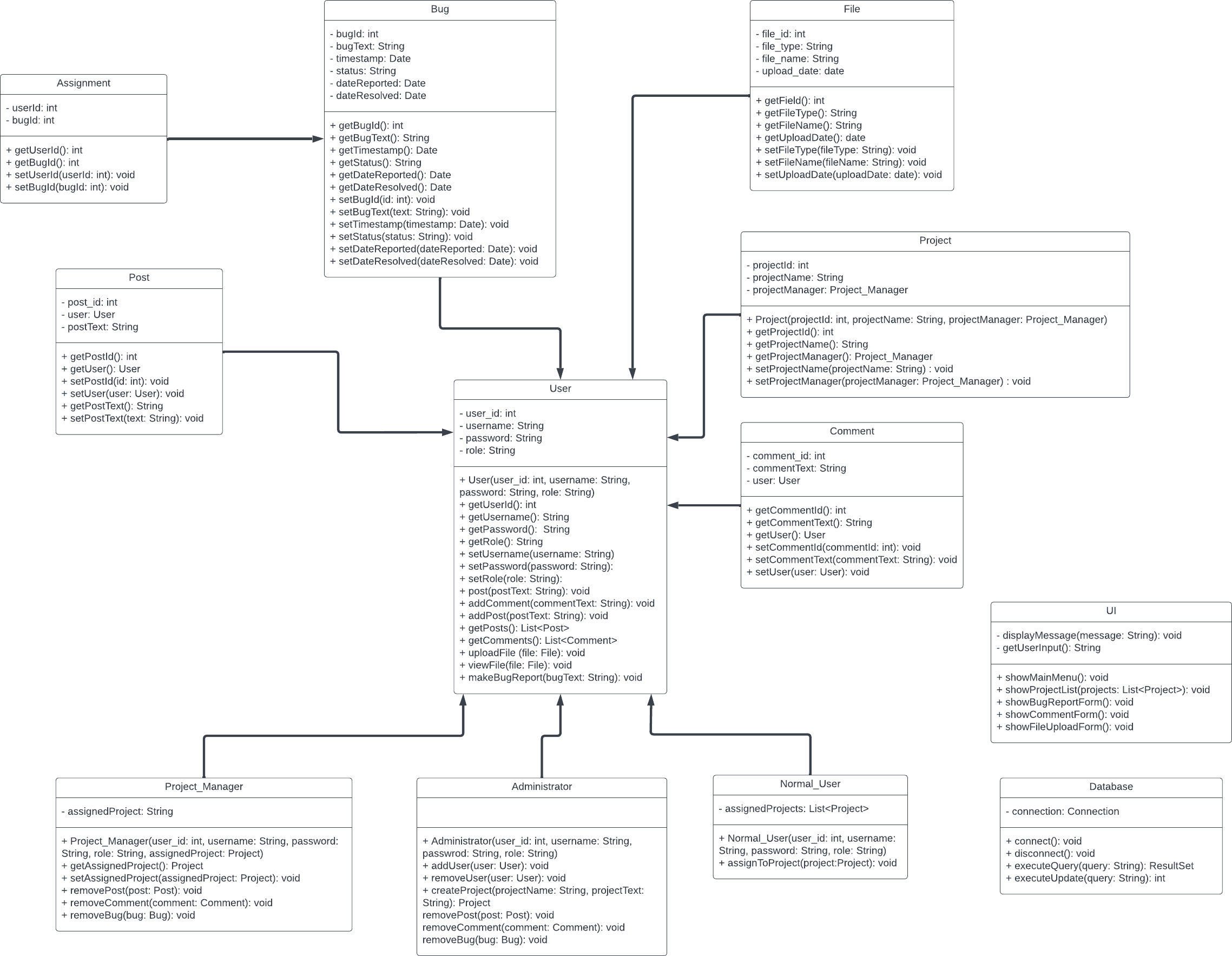
BRs will be another valuable option to create and store in the database. Users who submit a BR receive a message once the BR is complete.

### Data Management Component

Administrators are the only users capable of modifying the database significantly due to their ability to assign users to projects and remove users from Team Vault. Table entities will be auto-incremented so as not to cause any errors with ID allocation. Expected errors cause the prevention of unexpected errors from occurring, causing the entire program to cease functioning or terminate. These errors will be fore expected incorrect events such as not uploading the correct file type, displayed in Figure 2. A prompt with an error message will appear if users do not enter one of the supported file extensions. Error handling will be expanded to when the user enters credentials, reports bugs, and assigns user roles.

## Detailed Requirements

### Team Vault Class Structure



***Figure 4 - Class Diagram of Software System***

### Requirement Matrix

|  |  |
| --- | --- |
| **Requirement** | **Description** |
| **R001[1] – Account Requirement for Software System Usage** | Users must have an account to use the software system. |
| **R002[1] – Normal User Permissions** | Normal Users can upload the supported files, view files within their assigned project(s), download files, and create posts, comments, and bug reports within their given project(s). |
| **R003[1] – Project Manager Permissions** | Project Managers can upload the supported files, view files within their specified project, download and delete files, and create and |

|  |  |
| --- | --- |
|  | delete posts, comments, and bug reports, including those of Normal Users within their specified project. |
| **R004[1] – Administrator Permissions** | Administrators can add and delete users, assign roles to users, assign Normal Users to projects, assign Project Managers to a specified project, download, view, and delete files, and create and delete posts, comments, and bug reports, including those of Normal  Users and Project Managers within projects. |
| **R005[1] – Login Credentials** | All users must sign in with a valid username and password. |
| **R006[1] – Password Complexity** | All passwords must be 12 characters long with at least an uppercase letter, a lowercase letter, a symbol, and a number. |
| **R007[1] – Supported File Extensions** | Users will be capable of uploading and viewing different types of files. For code, it will be a text file with the extension .txt; documents will be in the form of .pdf, .doc, and .docx; for images, it will be .jpeg and  .png. |
| **R008[1] – Post & Comment Limit** | The database will store up to 50 posts and comments before it deletes the oldest comment or post. |
| **R009[1] – Character Limit for Posts, Texts, and Bug Reports** | Posts, Texts, and Bug Reports have a max of 30,000 characters. |
| **R010[1] – Character Limit for Bug Report Title** | Bug Reports must contain a title with a 50- character limit and an option to add a picture. |
| **R011[1] – File Upload Size Limitation** | The maximum size of the file users can upload is 3 megabytes. |
| **R012[1] – Minimum Hardware Specifications** | The software system will require 3.0 gigabytes of available disk space, 4.0 gigabytes of RAM, and at least 1024x768 screen resolution. |
| **R013[1] – Compatible Operating Systems** | The compatible operating systems for the software are Windows 11, Windows 10,  Windows 8 or 8.1, Ubuntu 12.04 or higher, macOS X (10.10) or later, and Red Hat Enterprise Linux 8.0 or higher. |
| **R014[2] – Password Expiration** | Passwords will expire after 12 months of being stored within the software system. |
| **R015[2] – Project Collaboration** | Enable collaborative features within projects, allowing users to invite other users, assign  tasks, and track project progress collectively. |
| **R016[2] – Email Notification** | Provide email notifications to users for important system events, such as new project |

|  |  |
| --- | --- |
|  | assignments, comments, or bug updates, ensuring they stay informed even when they are not actively using the system. |
| **R017[3] – Customizable Color Scheme** | Dark Mode and Light Mode will be two options for color schemes. |
| **R018[3] – Real-time Notifications** | Implement real-time notifications to notify users of new comments, replies, or updates related to their posts or projects. |
| **R019[3] – Advanced Search Functionality** | Enhance the search functionality by implementing advanced search features such as filtering by date, user, project, or specific keywords. |
| **R020[3] – Multi-language Support** | Support multiple languages within the system interface to accommodate users from different regions and facilitate better accessibility. |

***Figure 5 - Requirement Chart for Software System***

### Functional and Non-Functional (Infrastructure) Requirements

Team Vault shall provide a user-friendly interface for users to upload files of various types, including code, images, and documents. The system shall validate the file format and size to ensure compatibility and prevent abuse. In Figure 5 R011, the requirement specifies the functionality of allowing users to upload files and mentions the need to validate file extension and size.

Team Vault shall ensure data security and confidentiality by implementing appropriate authentication and encryption mechanisms. User data, including usernames, passwords, and personal information, will be protected from unauthorized access or disclosure. All users will have an ID automatically assigned once a user is within the database. This process will allow for a comprehensive log of posts, comments, BRs, and files created by users, which will prove valuable during audits for FileGuard Industries.

### Storage Requirements, Operational Requirements, System Rules

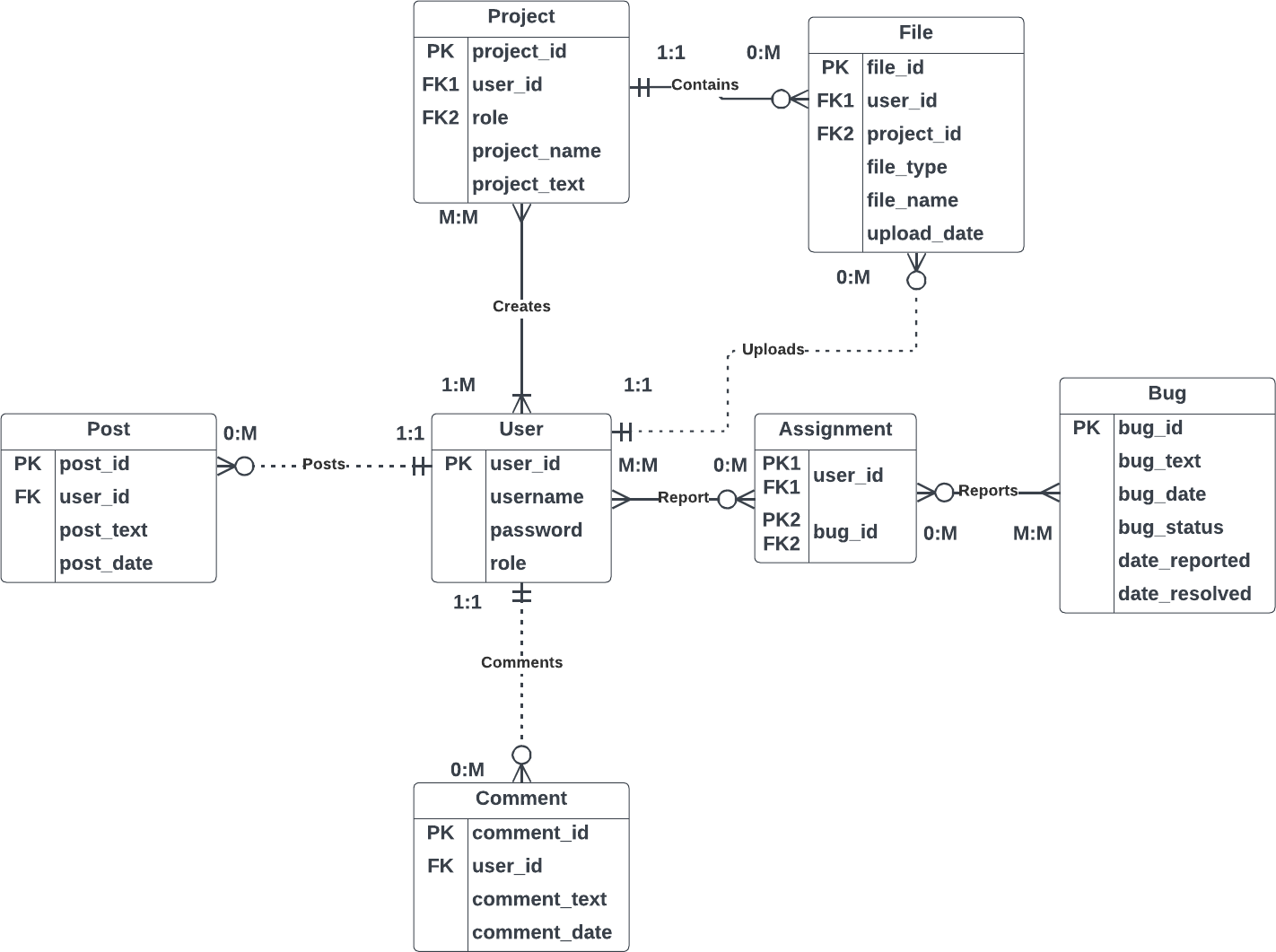
The current version of the software system does not require the user to store Personal Identifiable Information (PII) unless necessary to provide optimal user experience and privacy. Team Vault offers a scalable storage size to accommodate user uploads, including files, images, and documents. Additionally, scalability will be an emphasis to adapt to the growing user base of Team Vault. The software system will ensure high availability and uninterrupted access for users. It provides an optimal response time with user interactions and a responsive and efficient user experience.

Additionally, the system will implement robust security measures, including user authentication and data encryption, to protect sensitive information and ensure data privacy.

Only authenticated users with valid credentials can access the system’s functionalities. In addition, according to R004 in Figure 5, only Administrators have the authority to create and remove user accounts, projects, and system content, as shown in Figure 2 with the roles and abilities of each role and in Figure 4 in which the Administrator class is the only role-based class that can add and remove accounts. Normal Users cannot delete files, BRs, comments, and posts without an Administrator. Users’ passwords must meet specific complexity requirements (minimum length and combination of characters) according to R009 in Figure 5. The system will log all user activities, including post creation, comment submissions, and BR updates.

## Interface Specifications

### Data Model



***Figure 6 - Entity Relationship Diagram of Software System***

### User Interface (UI)

The UI will primarily support a keyboard and mouse for computer use. Capabilities of the UI will include textboxes and buttons on a multitude of panes.

### Application Programming Interface (API)

The API of the software system will be a model of the Class Diagram shown in Figure 4. User roles will be their classes inherited from a parent class called User. Depending on the role within the system, users have additional privileges outside of the User class. For example, an Administrator assigns a Project Manager to a specific project or Normal Users to one or more projects. The User class is the most important class of the Class Diagram as it has many bidirectional associations with other classes such as Bug, Post, and Comment. The UI and

Database classes will be responsible for sending information to the database and providing a visual interaction of the software system to end users.

### Database Interface

In Figure 6, the software system's Entity Relationship Diagram (ERD) has a User as the crucial entity holding other entities together, like Figure 4’s User class. Users have a mix of strong and weak relationships between entities. Unlike Figure 4, Figure 6 does not have three separate table entities for the roles as the difference between the roles are too little to warrant creating additional entities. The Assignment table entity's purpose is to make a bridge between User and Bug because there would be a many-to-many (M: M) relationship between User and Bug and the complexity of the two entities’ relationship.

### Error Handling & Exception Interface

When it comes to the errors in the software system, there will be various types of errors, such as input validation, database connection, file access, and network errors. The priority is to ensure that the software system will keep running if these errors occur and to solve these errors for the best quality assurance for users. To best help users, there will be an error message along with the error number and message stating why the error occurred. The error number keeps track of the number of errors predicted to happen and keep track of. And the error message is used for end users to understand the occurrence of the error and assist in resolving the issue.

## Critical System Constraints

### Resource Constraints

Team Vault will require 3.0 GB (Gigabytes) of available disk space, 4.0 GB RAM, and a requirement of at least 1024x768 screen resolution, according to Figure 5, R012.

### Availability and Reliability Constraints

Team Vault’s database will be available 24/7 using AWS as the cloud-based infrastructure.

### Compatibility Constraints

Operating Systems compatible with the software System are Windows 11, Windows 10, Windows 8 or 8.1, Ubuntu 12.04 or higher, macOS X (10.10) or later, and Red Hat Enterprise Linux 8.0 or higher, according to Figure 5, R013.

### Security Constraints

Team Vault will not contain PII outside of username and password to avoid data leakage. User IDs are kept private.

### Scalability Constraints

Team Vault has no definite user capacity but offers a scalable storage size controlled using the AWS database.

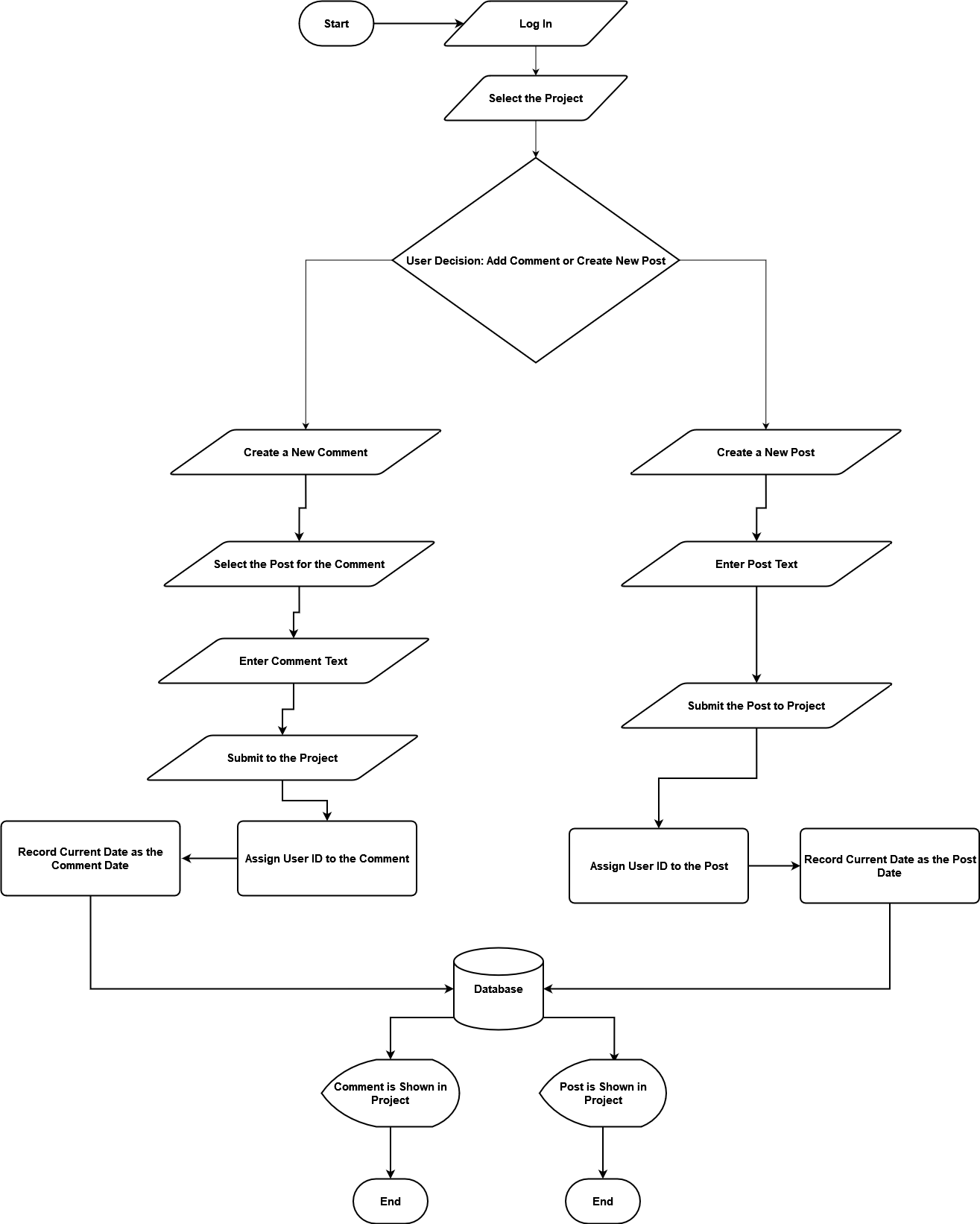
### Regulatory & Compliance Constraints

User information is kept with the organization’s database and will not be sold or distributed outside company use.

## Software Design

## Process-Orientation Flow (POF)

### POF Chart



***Figure 7 - POF Chart Posting and Commenting Process in Team Vault***

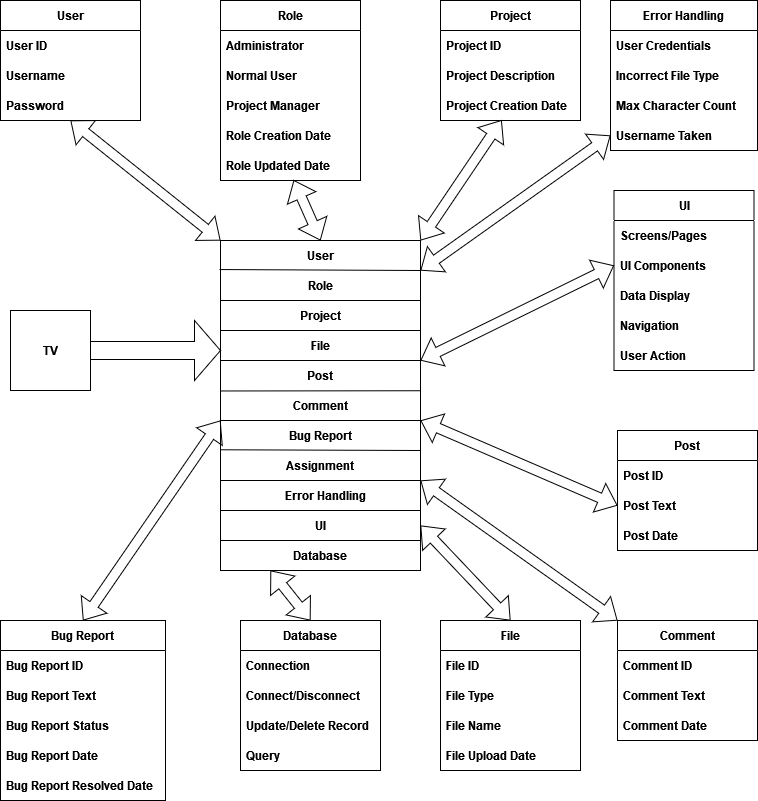
The purpose of Figure 7 is to show how users would post and comment on a Project. After the user logs into Team Vault, they must select a project they have access. The user can either leave a comment on the project or respond to a comment by leaving a post. There is a decision point for the user to decide whether they would like to add a comment or a post to a project.

If the user decides to leave a comment, they will need to select a post and then start creating their text. After the user finishes and enters the comment, it will be assigned the user’s ID within the system and set a date for the comment in the database. Afterward, it is in the database and displayed within the project.

Post follows the same process as a comment; however, when a user creates a post, they will not need to respond to another post but start typing within the text field. This post, after submissions, will be given the user’s ID and date, sent to the database, and visually shown in the project.

## Information Topology Chart (ITC)

### ITC



***Figure 8 - ITC for Team Vault***

The ITC chart has eleven subsystems: User, Role, Project, File, Post, Comment, Bug Report, Assignment, Error Handling, UI, and Database. Database acts like a database, storing user information, while UI is an external source to give users a visual representation of the software system for the server to respond to the client to send user information to the database—all subsystems except Error Handling and UI. Data flows from the subsystems to the system holding all eleven. User information for Team Vault is its most critical data and will be stored primarily in AWS cloud storage due to FileGuard Industries's size and cloud-based infrastructure's scalability.

## Data Flow Diagrams (DFD)

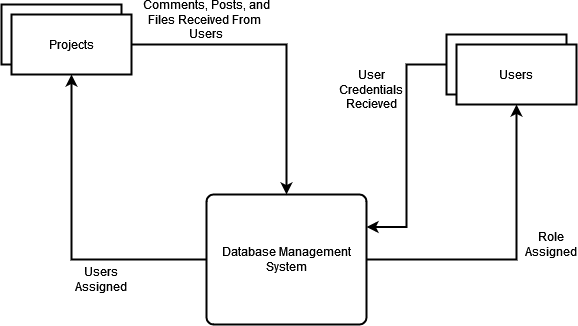
### Important Operations Chart

|  |  |
| --- | --- |
| **Operation/Process** | **Description** |
| **P01: Add Users** | Facilitates the ability to add users to one or more projects. |
| **P02: Remove Users** | Facilitates the ability to removes users from one or more projects. |
| **P03: Receive File** | Facilitates the recording of files received from users. |
| **P04: Receive Comment** | Facilitates the recording of comments received from users for posts. |
| **P05: Receive Post** | Facilitates the recording of posts received from users. |

***Figure 9 - Important Operation in the Database Management System***

The Important Operations Chart covers the processes and their description used in the DFDs. These processes will be visually in one or more DFDs.

### System DFD



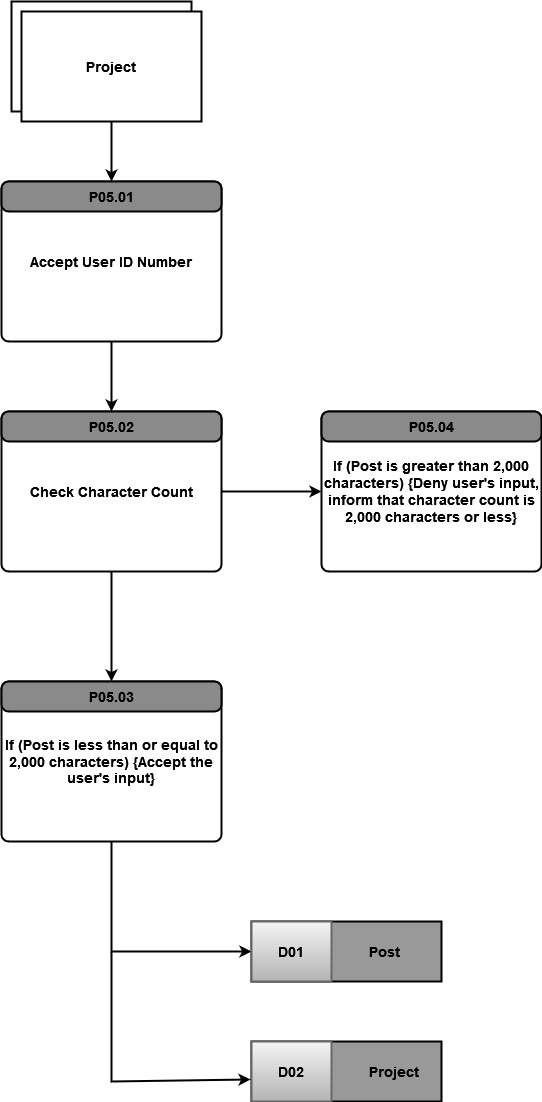
***Figure 10 – Level-0 Context DFD for the Database Management System***

Figure 10 represents DFD illustrating the primary functionality of a project-based software system that facilitates user collaboration. This DFD is a context diagram showing Team Vault as an extensive process. The system involves Users, Projects, and a Database Management System (DBMS).

Users interact with the system by performing various actions, such as logging in with their username and password. These actions are managed and processed by the DBMS, which acts as the central hub for handling user-related information.

The DBMS also plays a crucial role in assigning roles to users and associating them with specific projects. It manages this role assignment process, ensuring users have the appropriate permissions and access levels within each project. Furthermore, the DBMS serves as the gateway for storing and retrieving information related to users and projects. It acts as the intermediary between the users and the database, managing all data flow and ensuring the proper storage and retrieval of information.

### Process DFD

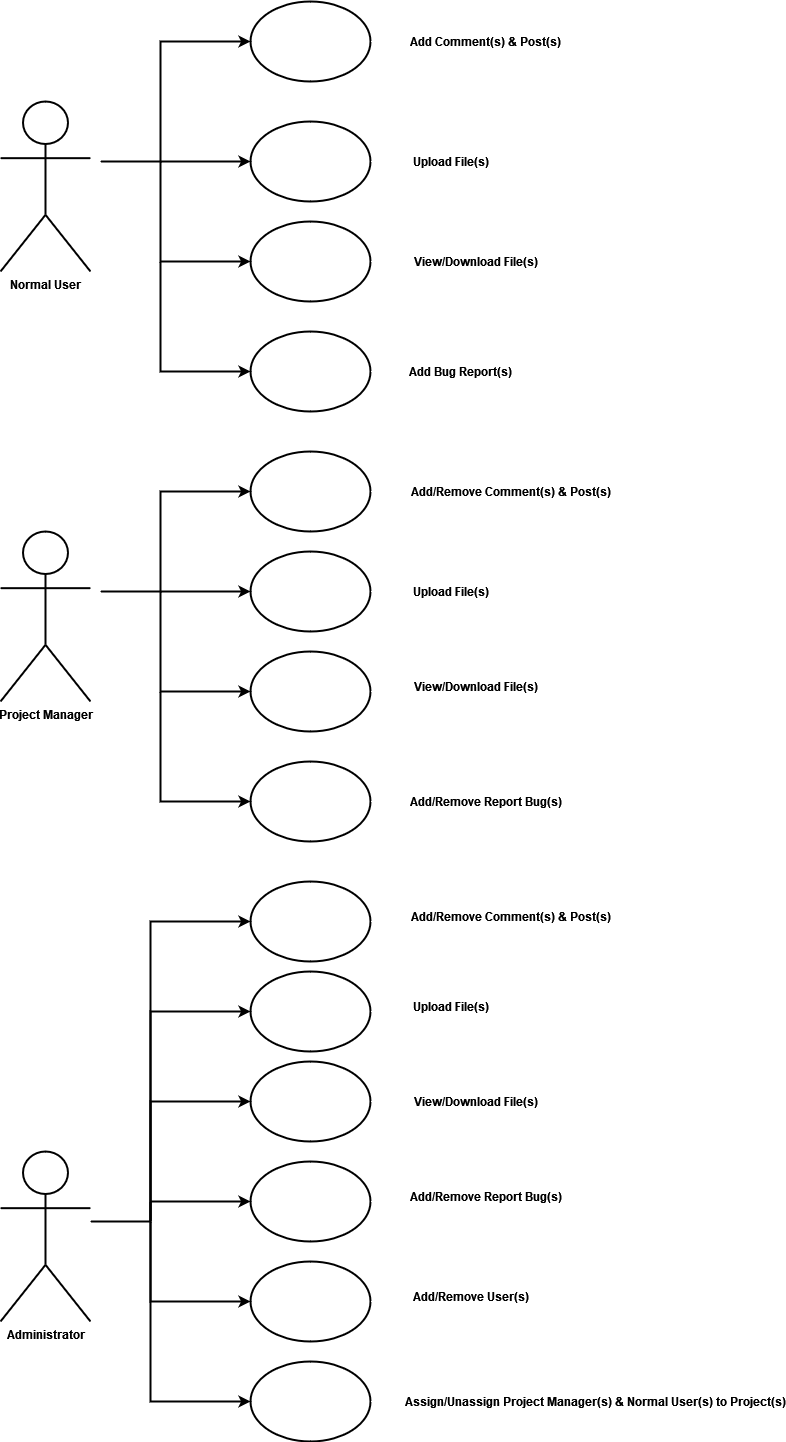


***Figure 11 - Level-2 DFD for Process P05 of the Database Management System***

Figure 11 also shows a DFD is a more detailed-oriented DFD covering a process of Team Vault rather than the entirety of the system. This DFD covers the process of P05, which details a project receiving a post. Before a user can create a post within a project, they must be able to access a project determined by the user ID number assigned to the project. When a user submits a post, there will be a condition that if the characters are more significant than two-thousand characters, a user will receive an error message a deny the submission to the DBMS. If the user’s post is two-thousand characters or less, then the post will be accepted and stored within the Post and Project entities within the database.

## Use Case Diagram

### UCD



***Figure 12 - UCD of User Roles and Privileges***

The UCD in Figure 12 covers the roles and their permissions in Team Vault. There are three roles or actors in the UCD: Normal User, Project Manager, and Administrator. The Normal User actor can add comments and posts and upload and download files within a project while adding bug reports. The Project Manager actor can also do the same permissions that a Normal User can do in addition to removing comments, posts, and bug reports. The Administrator actor has the same permissions as Project Manager but, in addition, can add and remove users from the Team Vault and projects.

## Use Case Scenarios

### Adding Comment(s)

|  |  |
| --- | --- |
| **UC-001** | **User Adds a Comment** |
| **Use Case** | **Comment** |
| **Actor(s)** | **Normal User, Project Manager, Administrator** |
| **Pre-Conditions** | **User must be logged in and have access to a project.** |
| **Details** | 1. **User navigates to the desired project.** 2. **User selects a specific item or section to add a comment.** 3. **User enters the comment content.** 4. **User submits the comment.** 5. **The system validates and stores the comment.** |
| **Post-Condition** | **The system displays the updated item or section with the added comment.** |

### Removing Comment(s)

***Figure 13 - Use Case 001***

|  |  |
| --- | --- |
| **UC-002** | **Remove Comment** |
| **Use Case** | **Comment** |
| **Actor(s)** | **Project Manager, Administrator** |
| **Pre-Conditions** | **Project Manager or Administrator must be logged in.** |
| **Details** | 1. **User navigates to the desired project.** 2. **User selects a specific post to be removed.** 3. **User confirms the action to remove the post.** |

|  |  |
| --- | --- |
|  | **4. The system removes the selected post.** |
| **Post-Condition** | **The system updates the project view to reflect the removal of the post.** |

### Adding Post(s)

***Figure 14 - Use Case 002***

|  |  |
| --- | --- |
| **UC-003 Add Post** | |
| **Use Case** | **Post** |
| **Actor(s)** | **Normal User, Project Manager, Administrator** |
| **Pre-Conditions** | **User is logged in and has access to the project.** |
| **Details** | 1. **User selects the project to which they want to add a post.** 2. **User navigates to the project's page or posts section.** 3. **User clicks on the "Add Post" button or a similar option.** 4. **System presents a form or text editor for creating a new post.** 5. **User enters the post text, attaches any necessary files, and provides relevant information.** 6. **User clicks on the "Submit" or "Publish" button.** 7. **System validates the input and creates a new post associated with the user and project.** 8. **System displays a success message indicating that the post has been added.** |
| **Post-Condition** | **User can view the newly added post in the project's posts section.** |

### Removing Post(s)

***Figure 15 - Use Case 003***

|  |  |
| --- | --- |
| **UC-004 Remove Post** | |
| **Use Case** | **Post** |
| **Actor(s)** | **Project Manager, Administrator** |
| **Pre-Conditions** | **Project Manager or Administrator is logged in and access to the project.** |

|  |  |
| --- | --- |
| **Details** | 1. **User selects the project from which they want to remove a post.** 2. **User navigates to the project's page or posts section.** 3. **User locates the post they want to remove.** 4. **User selects the post and chooses the "Delete" option.** 5. **System prompts the user to confirm the deletion.** 6. **User confirms the deletion.** 7. **System deletes the post and any associated comments or files.** 8. **System displays a success message indicating that the post has been removed.** |
| **Post-Condition** | **User can no longer view the deleted post in the project's posts section.** |

### Uploading File(s)

***Figure 16 - Use Case 004***

|  |  |
| --- | --- |
| **UC-005** | **Upload File** |
| **Use Case** | **File** |
| **Actor(s)** | **Normal User, Project Manager, Administrators** |
| **Pre-Conditions** | **User is logged in and has access to the project.** |
| **Details** | 1. **User selects the project to which they want to upload a file.** 2. **User navigates to the project's page or files section.** 3. **User clicks on the "Upload File" or a similar option.** 4. **System presents a file upload dialog or form.** 5. **User selects the file(s) they want to upload from their local device.** 6. **User clicks on the "Upload" button.** 7. **System validates the file(s) and begins the upload process.** 8. **System stores the file(s) in the appropriate location or storage system.** 9. **System associates the uploaded file(s) with the user and project.** |

|  |  |
| --- | --- |
|  | **10. System displays a success message indicating that the file(s) have been uploaded.** |
| **Post-Condition** | **User can view the uploaded file(s) in the** **project's files section.** |

### Viewing File(s)

***Table 17 - Use Case 005***

|  |  |
| --- | --- |
| **UC-006 View File** | |
| **Use Case** | **File** |
| **Actor(s)** | **Normal Users, Project Managers, Administrators** |
| **Pre-Conditions** | **User is logged in and has access to the project.** |
| **Details** | 1. **User selects the project for which they want to view files.** 2. **User navigates to the project's page or files section.** 3. **User locates the file they want to view.** 4. **User clicks on the file name or selects the file to open it.** 5. **System retrieves and displays the file in an appropriate viewer based on the file type.** 6. **User can scroll through the file, zoom in or out, and perform other relevant actions based on the file type.** |
| **Post-Condition** | **User can close the file viewer to return to the project's files section.** |

### Downloading File(s)

***Table 18 - Use Case 006***

|  |  |
| --- | --- |
| **UC-007 Download File** | |
| **Use Case** | **File** |
| **Actor(s)** | **Normal User, Project Manager, Administrator** |
| **Pre-Conditions** | **User is logged in and has access to the project.** |
| **Details** | 1. **User selects the project from which they want to download a file.** 2. **User navigates to the project's page or files section.** |

|  |  |
| --- | --- |
|  | 1. **User locates the file they want to download.** 2. **User clicks on the file name or selects the file to initiate the download.** 3. **System retrieves the file and prompts the user to save it to their local device.** 4. **User selects the destination folder and confirms the download.** 5. **System transfers the file to the user's device.** |
| **Post-Condition** | **User can access the downloaded file in the specified folder on their device.** |

### Add Bug Report(s)

***Table 19 - Use Case 007***

|  |  |
| --- | --- |
| **UC-008 Add Bug Report** | |
| **Use Case** | **Bug Report** |
| **Actor(s)** | **Normal User, Project Manager, Administrator** |
| **Pre-Conditions** | **User is logged in and has access to the project.** |
| **Details** | 1. **User selects the project for which they want to add a bug report.** 2. **User navigates to the project's page or bug reports section.** 3. **User clicks on the "Add Bug Report".** 4. **System presents a bug report form with fields for entering relevant information.** 5. **User fills in the bug report details, such as title, description, steps to reproduce, and severity.** 6. **User attaches any necessary files or screenshots related to the bug report.** 7. **User clicks on the "Submit" button to add the bug report.** 8. **System validates the input and creates a new bug report associated with the user and project.** |

|  |  |
| --- | --- |
|  | **9. System displays a success message indicating that the bug report has been added.** |
| **Post-Condition** | **User can view the newly added bug report** **in the project's bug reports section.** |

### Remove Bug Report(s)

***Table 20 - Use Case 008***

|  |  |
| --- | --- |
| **UC-009 Remove Bug Report** | |
| **Use Case** | **Bug Report** |
| **Actor(s)** | **Project Manager, Administrator** |
| **Pre-Conditions** | **Project Manager or Administrator is logged in and has access to the project.** |
| **Details** | 1. **User selects the project from which they want to remove a bug report.** 2. **User navigates to the project's page or bug reports section.** 3. **User locates the bug report they want to remove.** 4. **User selects the bug report or clicks on a "Remove".** 5. **System prompts the user for confirmation to remove the bug report.** 6. **User confirms the removal of the bug report.** 7. **System deletes the bug report from the project's bug reports list.** |
| **Post-Condition** | **System displays a success message indicating that the bug report has been removed.** |

***Table 21 - Use Case 009***

### Adding User(s) to System

|  |  |
| --- | --- |
| **UC-010 Add User** | |
| **Use Case** | **User** |
| **Actor(s)** | **Administrator** |
| **Pre-Conditions** | **Administrator is logged in.** |
| **Details** | 1. **Administrator navigates to the user management section or a similar option.** 2. **Administrator selects the "Add User" to initiate user creation.** |

|  |  |
| --- | --- |
|  | 1. **System presents a user creation form with fields for entering user details.** 2. **Administrator fills in the required user information, such as username, password, and role.** 3. **Administrator selects the "Submit" button to add the user.** 4. **System validates the input and creates a new user with the provided details.** 5. **System assigns a unique user ID and generates necessary credentials for the user.** |
| **Post-Condition** | **System displays a success message indicating that the user has been added.** |

***Table 22 - Use Case 010***

### Removing User(s) from System

|  |  |
| --- | --- |
| **UC-011 Add User to System** | |
| **Use Case** | **User** |
| **Actor(s)** | **Administrator** |
| **Pre-Conditions** | **Administrator is logged in.** |
| **Details** | 1. **Administrator navigates to the user management section or a similar option.** 2. **Administrator locates the user they want to remove from the system.** 3. **Administrator selects the user.** 4. **System prompts the administrator for confirmation to remove the user.** 5. **Administrator confirms the removal of the user.** 6. **System deletes the user from the system and removes associated data.** |
| **Post-Condition** | **System displays a success message indicating that the user has been removed.** |

***Table 23 - Use Case 011***

### Assigning Project Manager to a Project

|  |  |
| --- | --- |
| **UC-012** | **Assign Project Manager to Project** |
| **Use Case** | **User** |
| **Actor(s)** | **Administrator** |
| **Pre-Conditions** | **Administrator must be logged in.** |
| **Details** | 1. **Administrator logs into the system.** 2. **Administrator accesses the project management section.** 3. **Administrator selects the desired project to manage.** 4. **Administrator chooses the option to assign a project manager.** 5. **Administrator selects the project manager from the available users.** 6. **Administrator submits the assignment request.** 7. **The system validates the request and updates the project with the assigned project manager.** |
| **Post-Condition** | **The system displays a success message and updates the project details.** |

***Table 24 - Use Case 012***

### Removing Project Manager from a Project

|  |  |
| --- | --- |
| **UC-013** | **Remove Project Manager from Project** |
| **Use Case** | **User** |
| **Actor(s)** | **Administrator** |
| **Pre-Conditions** | **Administrator is logged in.** |
| **Details** | 1. **Administrator selects the project from which they want to remove a project manager.** 2. **Administrator navigates to the project's page or project management section.** 3. **Administrator locates the project manager they want to remove.** 4. **Administrator selects the project manager and click "Remove".** |

|  |  |
| --- | --- |
|  | 1. **System prompts the administrator for confirmation to remove the project manager.** 2. **Administrator confirms the removal of the project manager.** 3. **System removes the project manager from the project and revokes their associated privileges.** |
| **Post-Condition** | **System displays a success message indicating that the project manager has been removed.** |

***Table 25 - Use Case 013***

### Assigning Normal User(s) to a Project

|  |  |
| --- | --- |
| **UC-014 Assign Normal User to Project** | |
| **Use Case** | **User** |
| **Actor(s)** | **Administrator** |
| **Pre-Conditions** | **Administrator is logged in.** |
| **Details** | 1. **Administrator selects the project to which they want to assign normal user(s).** 2. **Administrator navigates to the project's page or project management section.** 3. **Administrator locates the section or option for adding users to the project.** 4. **Administrator selects the "Add User" to initiate user assignment.** 5. **System presents a list of available normal users who are not currently assigned to the project.** 6. **Administrator selects one or more normal users from the list.** 7. **Administrator confirms the selection to assign the user(s) to the project.** 8. **System adds the selected normal user(s) to the project and grants them appropriate privileges.** |
| **Post-Condition** | **System displays a success message indicating that the user(s) have been assigned to the project.** |

***Table 26 - Use Case 014***

### Removing Normal User(s) to a Project

|  |  |
| --- | --- |
| **UC-015** | **Remove Normal User from Project** |
| **Use Case** | **User** |
| **Actor(s)** | **Administrator** |
| **Pre-Conditions** | **Administrator is logged in.** |
| **Details** | 1. **Administrator selects the project from which they want to remove normal user(s).** 2. **Administrator navigates to the project's page or project management section.** 3. **Administrator locates the normal user(s) they want to remove.** 4. **Administrator selects the normal user(s) and clicks "Remove".** 5. **System prompts the administrator for confirmation to remove the normal user(s).** 6. **Administrator confirms the removal of the normal user(s).** 7. **System removes the selected normal user(s) from the project and revokes their associated privileges.** |
| **Post-Condition** | **System displays a success message indicating that the normal user(s) have been removed.** |

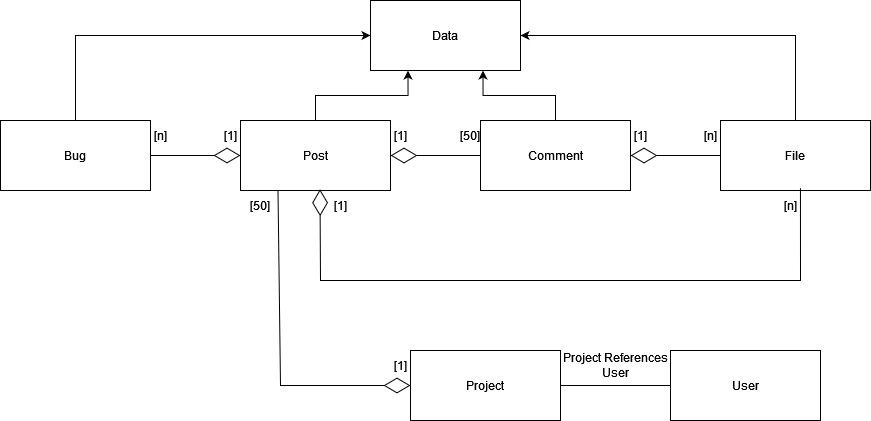
***Table 27 - Use Case 015***

### Use Case Scenario Overview

Figures 13-27 cover scenarios for roles and privileges in Figure 12. The scenarios specify the use cases for comments, posts, bug reports, files, and users. The details give a procedural design in acting the scenario.

## Object Relation Diagram

### ORD



***Figure 28 - ORD for the Database Management System***

Figure 28 displays the object types for Team Vault. The purpose of the ORD is to provide a visual representation of the data structure and relationships within Team Vault. Furthermore, it helps understand how different objects are connected and interact with each other regarding data flow. There are seven object types: Bug, Data, Post, Comment, File, Project, and User. Bug, Post, Comment, and File are inherited from Data, which will contain functions used in these four objects. In addition to inheritance, there is aggregation between Bug, Post, Comment, File, and Project. Post will contain a list of bug reports from Bug, comments from Comment with a limitation of fifty comments, and many files from Files. Comment object will have many files from Files while Project will contain a list of a maximum of fifty posts while Project references User.

## Object Structure Diagrams

### OSD

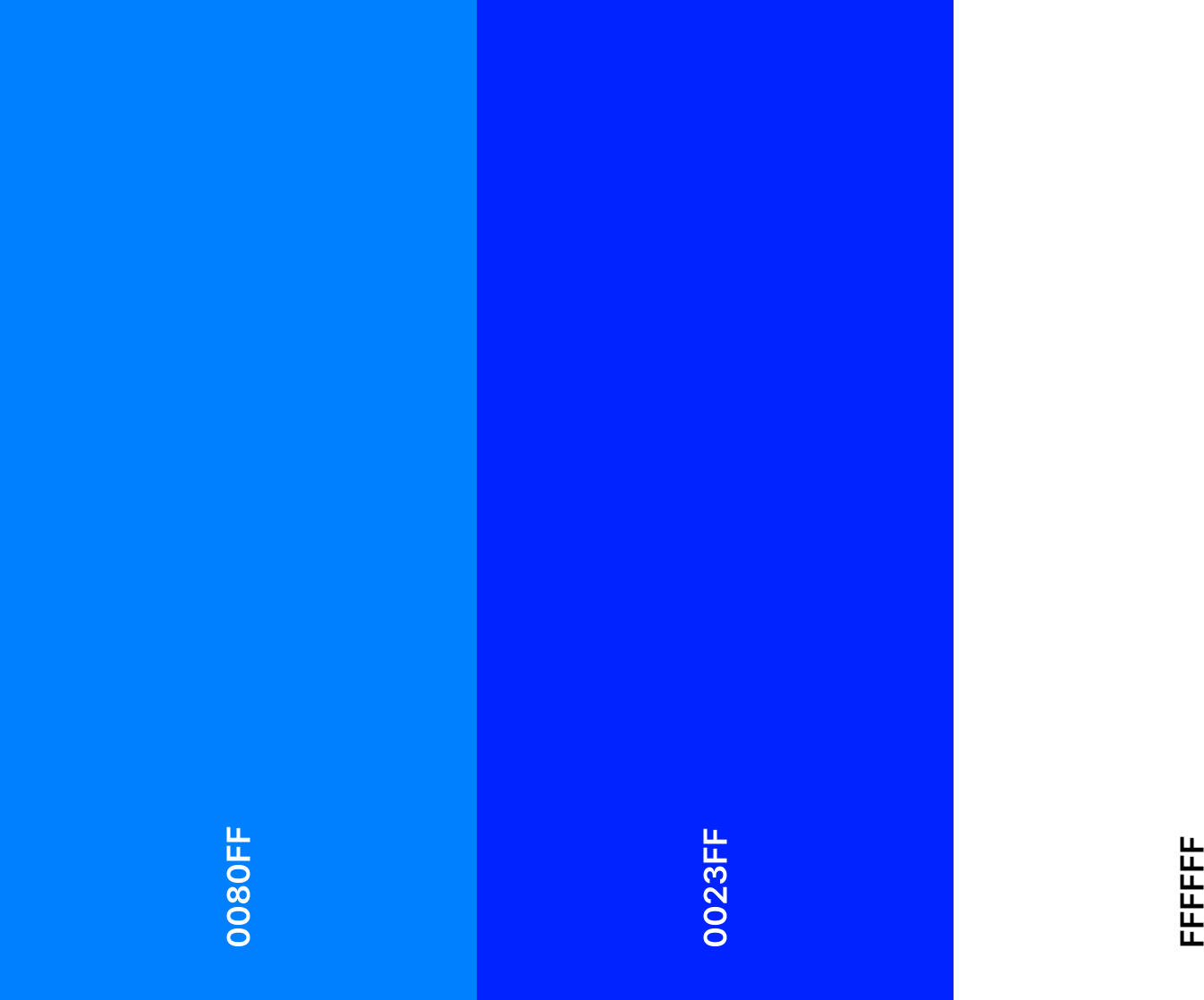


***Figure 29 - OSD of Team Vault***

The OSD provides an overview of the static structure and relationship among objects and classes of Team Vault. The OSD represents several classes: Data, File, Post, Comment, Bug, User, Project, and UI. The Data class serves as the superclass for File, Post, Comment, and ug classes, indicating the inheritance of attributes and behaviors. File class represents uploaded files; Post class represents user posts; Comment class represents user comments; Bug class represents bug reports; User class represents system users; Project class represents collaborative projects, and UI class visually ties in the other classes for the end-user to interact with.

## UI Design

## Color Schema

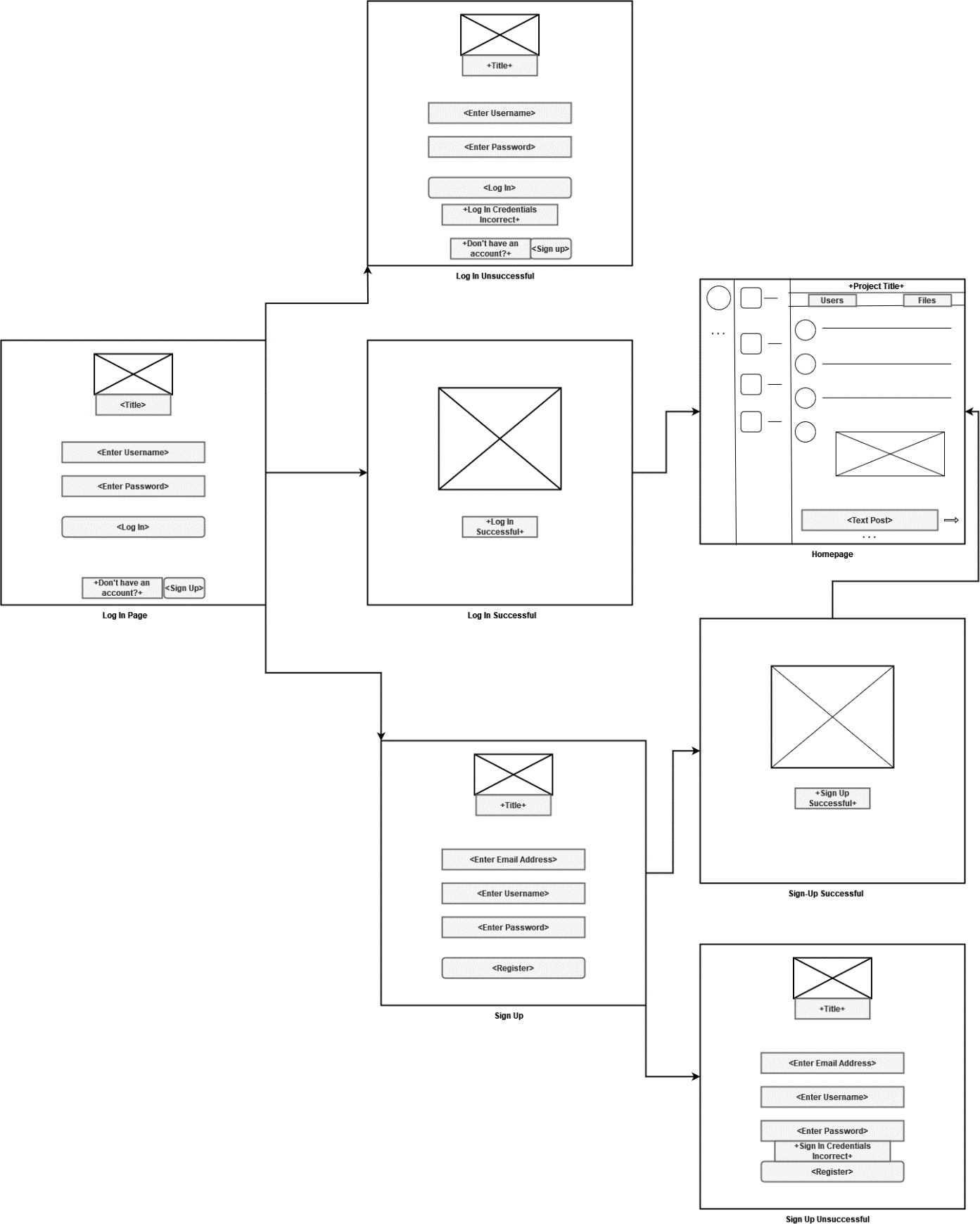


***Figure 30- Team Vault Color Schema***

The color schemes rely on #0023FF for the icon, #0080FF with the slogan, and #FFFFFF with the background. The reason for choosing blue colors is due to the color blue safety and calmness, emphasizing users’ trust in using Team Vault to communicate information to other users within projects and share additional information.

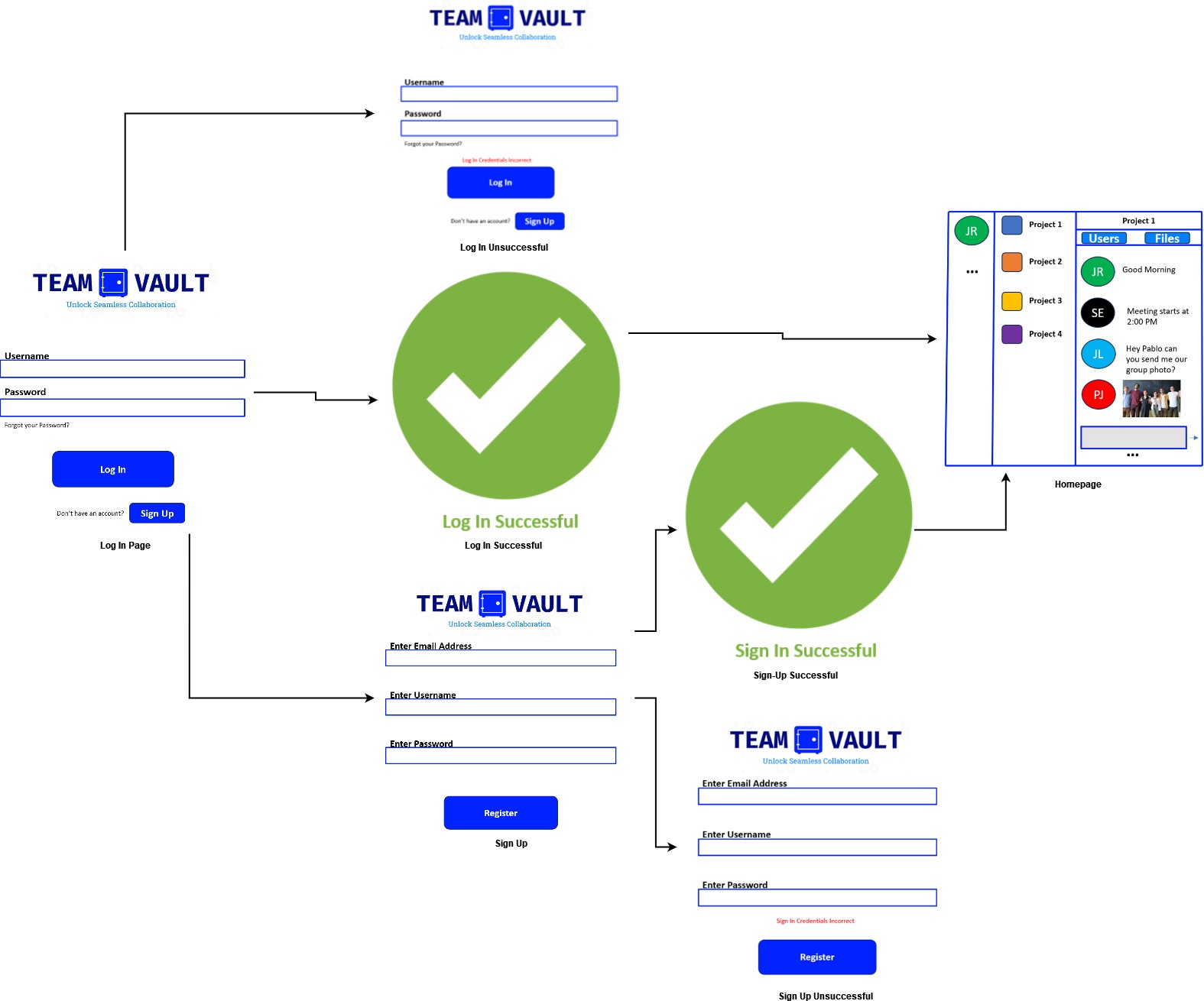
## Logging In

### Logging In Storyboard



***Figure 31 - Log-In Process Storyboard***

### Logging In Desing



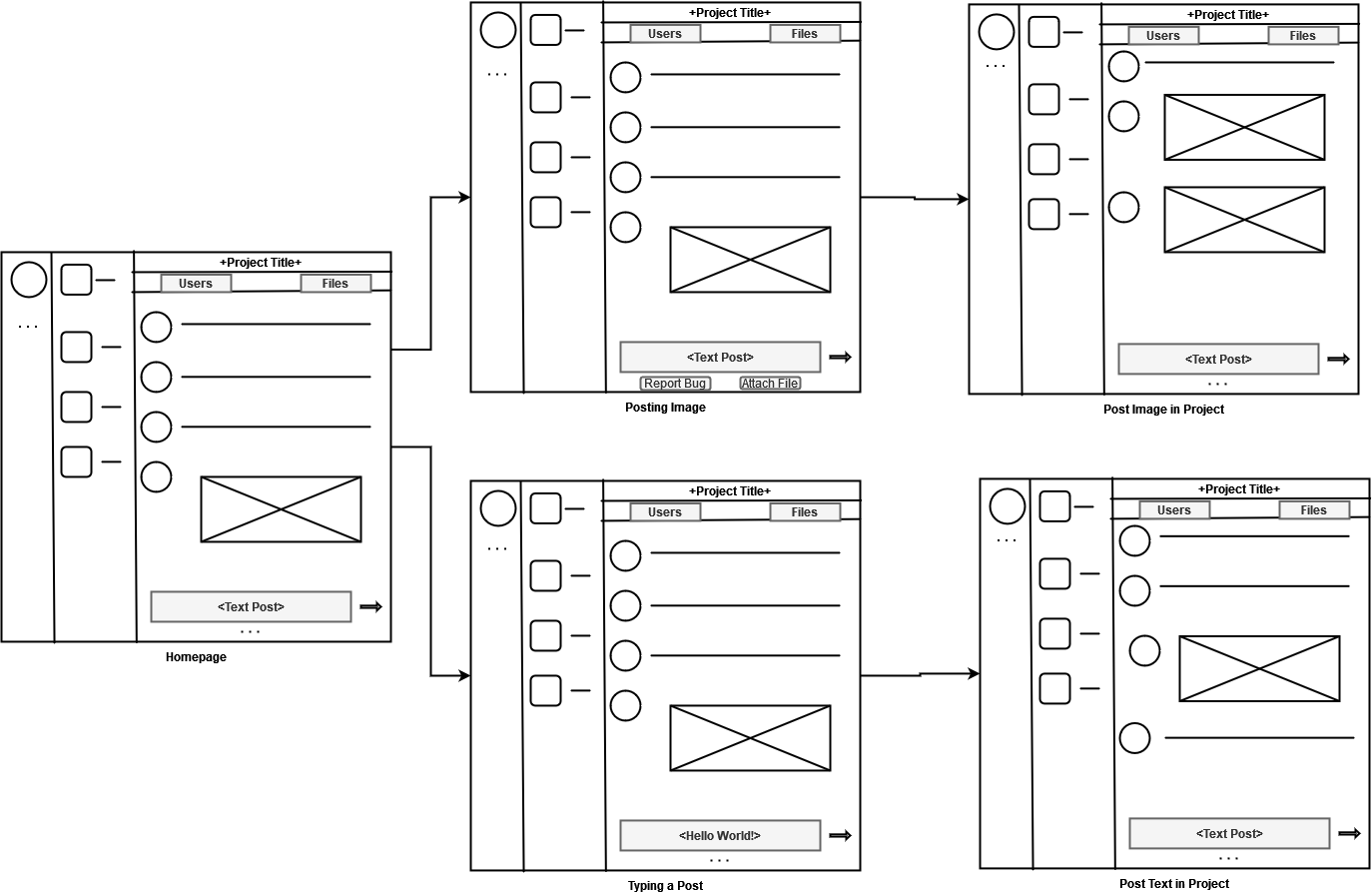
### Logging In Overview

***Figure 32 - Log-In Process Design***

In Figure 31, the user will have multiple options to access Team Vault’s home page. Users can log in and sign up, creating an account. There will be error handling if the user enters incorrect credentials for logging in and signing up, along with an error message for each condition. Figure 32 shows the prototype of using the login page.

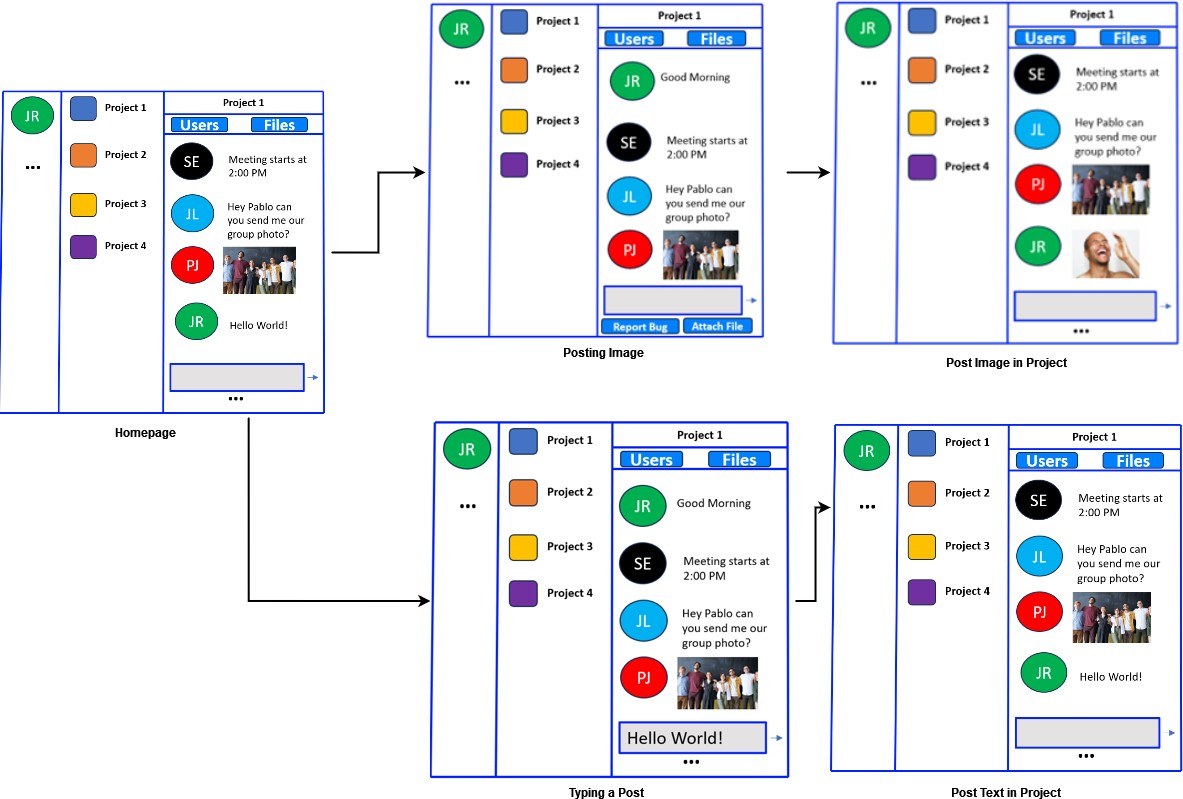
## Creating a Post

### Creating a Post Storyboard



***Figure 33 - Post Creation Storyboard***

### Creating a Post Design



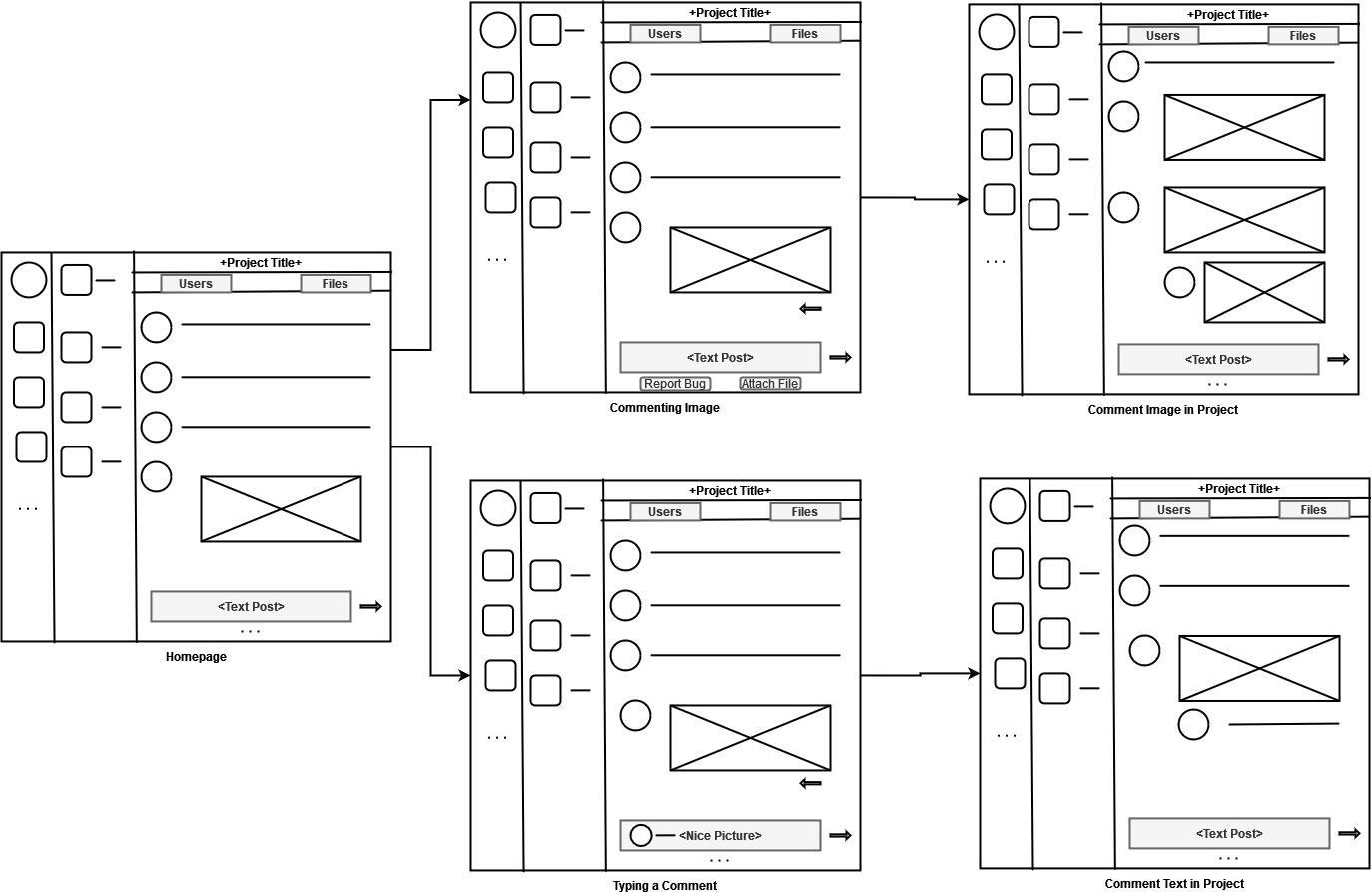
***Figure 34 - Post Creation Design***

### Creating a Post Overview

The homepage is the most important location of Team Vault, with most resources for front-end development going onto the homepage. The left side includes the user's profile picture and a three-dotted button containing settings, the ability to create a project, and adding users depending on the specific user's privileges. The middle section includes all projects the current user is in. The right-most area is where the user can interact with a given project. For posting, the user can type in the textbox within the project. In addition to posting, users can attach files and report bugs within a project.

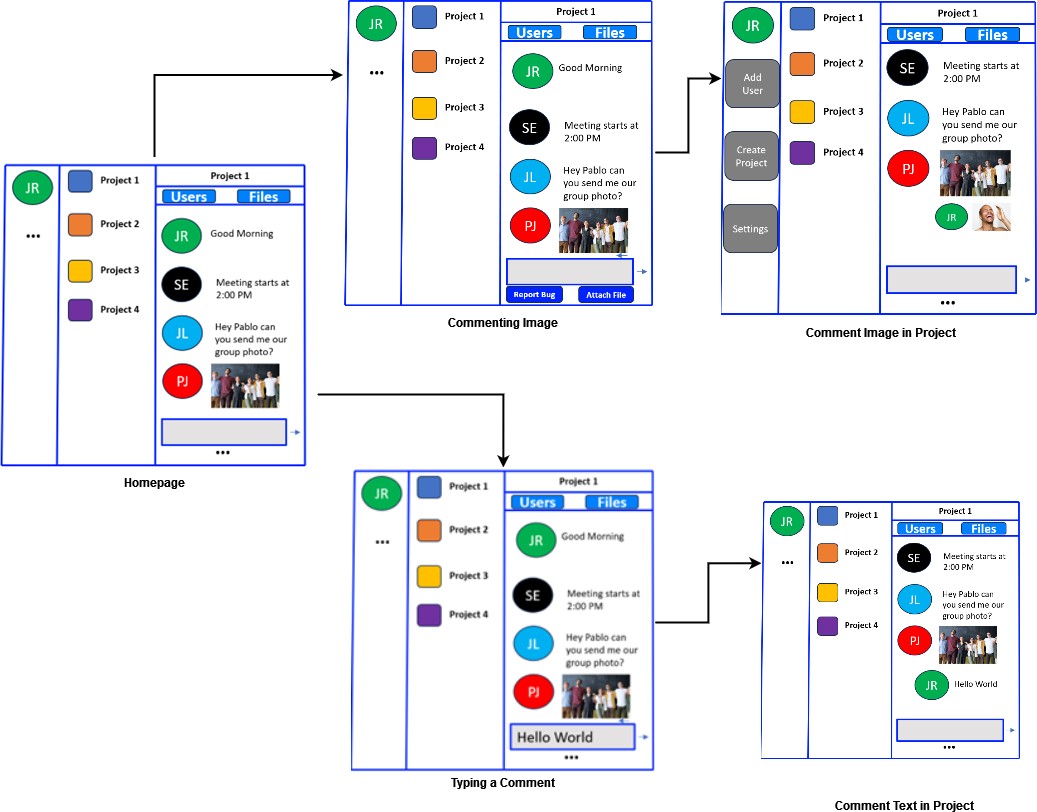
## Creating a Comment

### Creating a Comment Storyboard



***Figure 35 - Comment Creation***

### Creating a Comment Design



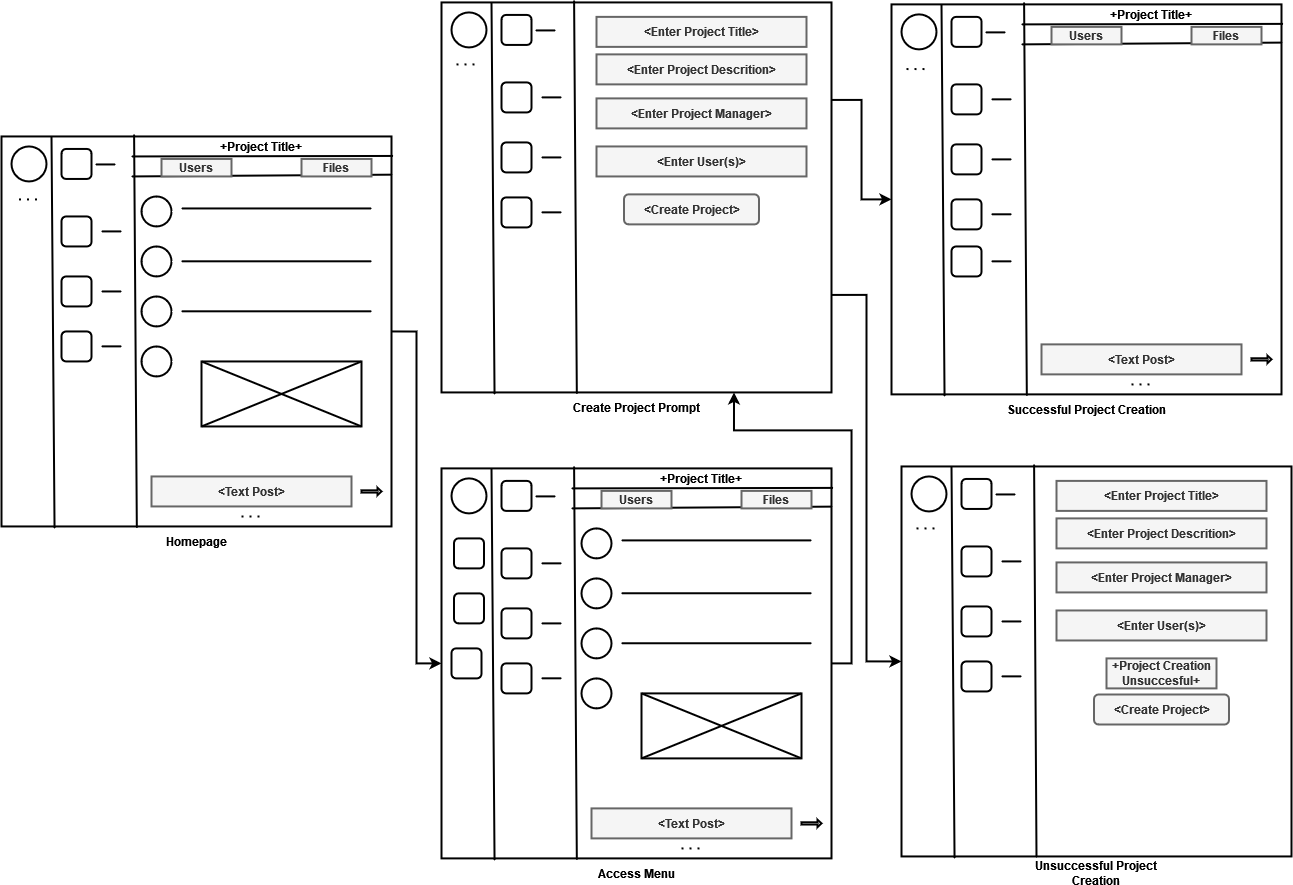
***Figure 36 - Comment Creation Design***

### Creating a Comment Overview

Commenting is like a post; however, the user must hover their mouse over another user’s post, click the arrow, and either attach a file or type within the textbox to create a comment.

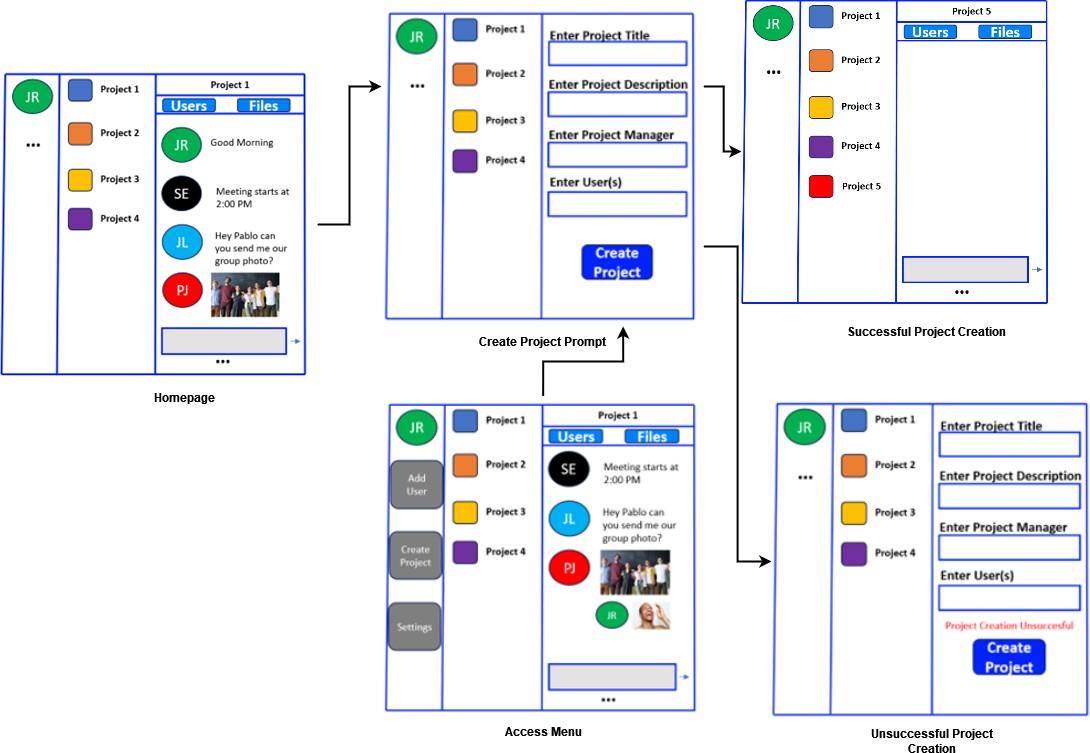
## Administrator Creating a Project

### Administrator Creating a Project Storyboard



***Figure 37 - Administrator Project Creation Storyboard***

### Administrator Creating a Project Design



***Figure 38 - Administrator Project Creation Design***

### Administrator Creating a Project Overview

On the left side of the page, the three-dotted button under the user’s profile picture will give an administrator access to create a project. The administrator must enter the required information to create the project.

## Project Planning

## Tabulation of Team Vault Activities

### Tabulation Chart

|  |  |  |  |
| --- | --- | --- | --- |
| **Activity** | **Predecessor** | **Description** | **Estimated Time (days)** |
| **A** | **-** | **Define project objectives and scope** | **2** |
| **B** | **A** | **Conduct initial requirements gathering** | **3** |
| **C** | **A** | **Identify project stakeholders and their roles** | **1** |
| **D** | **A,B,C** | **Create project schedule and timeline** | **2** |
| **E** | **A** | **Develop project budget and resource allocation** | **2** |
| **F** | **A** | **Define project deliverables and milestones** | **1** |
| **G** | **A** | **Assess project risks and develop risk mitigation plan** | **3** |
| **H** | **A** | **Define project communication plan** | **1** |
| **I** | **A** | **Assign roles and responsibilities to team members** | **1** |
| **J** | **I** | **Conduct team kickoff meeting** | **1** |
| **K** | **J** | **Conduct regular team meetings and progress updates** | **5** |
| **L** | **A** | **Develop project documentation** | **5** |
| **M** | **B** | **Perform initial system analysis** | **5** |

|  |  |  |  |
| --- | --- | --- | --- |
| **N** | **M** | **Design database architecture** | **5** |
| **O** | **M** | **Design system architecture** | **7** |
| **P** | **N,O** | **Develop front-end user interface** | **10** |
| **Q** | **N,O** | **Develop back-end functionality and database integration** | **12** |
| **R** | **P,Q** | **Conduct system testing and quality assurance** | **7** |
| **S** | **R** | **Perform user acceptance testing** | **5** |
| **T** | **S** | **Refine and finalize system based on user feedback** | **5** |
| **U** | **T** | **Prepare system deployment and release** | **3** |
| **V** | **U** | **Implement system deployment and release** | **2** |

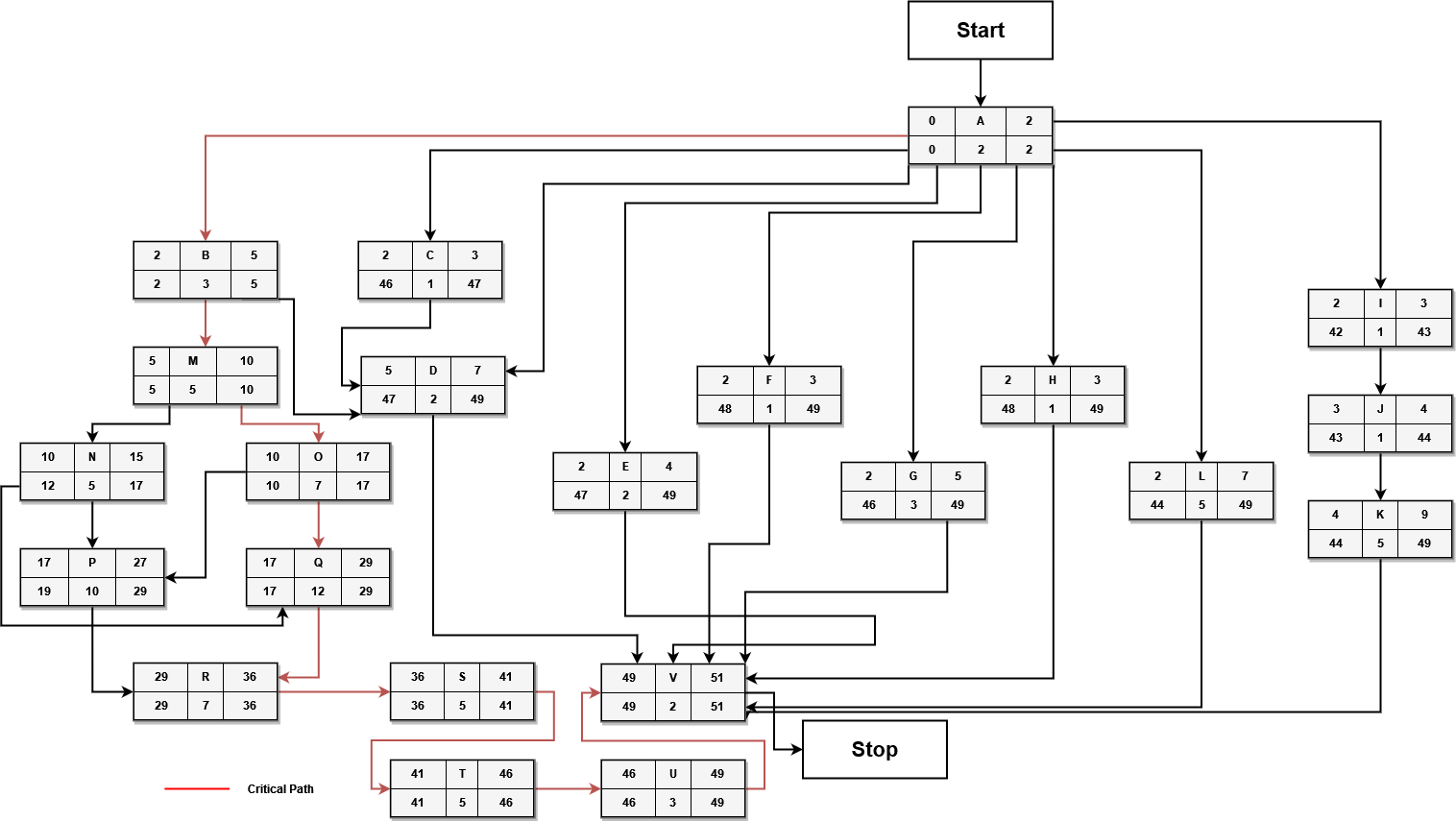
***Figure 39 - Schedule of Activities for Team Vault***

Figure 39 presents the tasks involved in the project planning phase of Team Vault. It outlines various activities required to design, develop, and deploy the software system. The activities are represented by alphabetic identifiers, and dependencies between tasks are specified to indicate their sequencing. The project planning begins with activity A, which involves gathering requirements and conducting initial research, task A is also the predecessor to most of the activities in Table 1. Activities B-K focus on planning and allocating resources to the construction of Team Vault.

Activities L-O continue the next phase of the project by designing the software such as putting the software into the form of a report, analyzing how Team Vault will function, to designing the structure of how the database will be designed. Activities O-R focus on the programming aspect of the project such as UI design, back-end functionality, and how the database will communicate to the software. Finally, activities S-V covers user-feedback pre and launch of Team Vault into the commercial market along with its maintenance.

## Program Evaluation and Review Technique (PERT)

### PERT Diagram



***Figure 40 - Team Vault PERT***

The project begins with task A, which involves defining project objectives and scope.

From there, the chart branches into multiple parallel tasks dependent on the previous tasks' completion. Tasks B, C, and L are parallel to each other and have a linear dependency on task A. Task D depends on both tasks B and C, meaning it can only start once both tasks are completed. Task D is a critical path task that determines the timeline for subsequent tasks.

Tasks M, N, and O are parallel to each other and are dependent on the completion of task

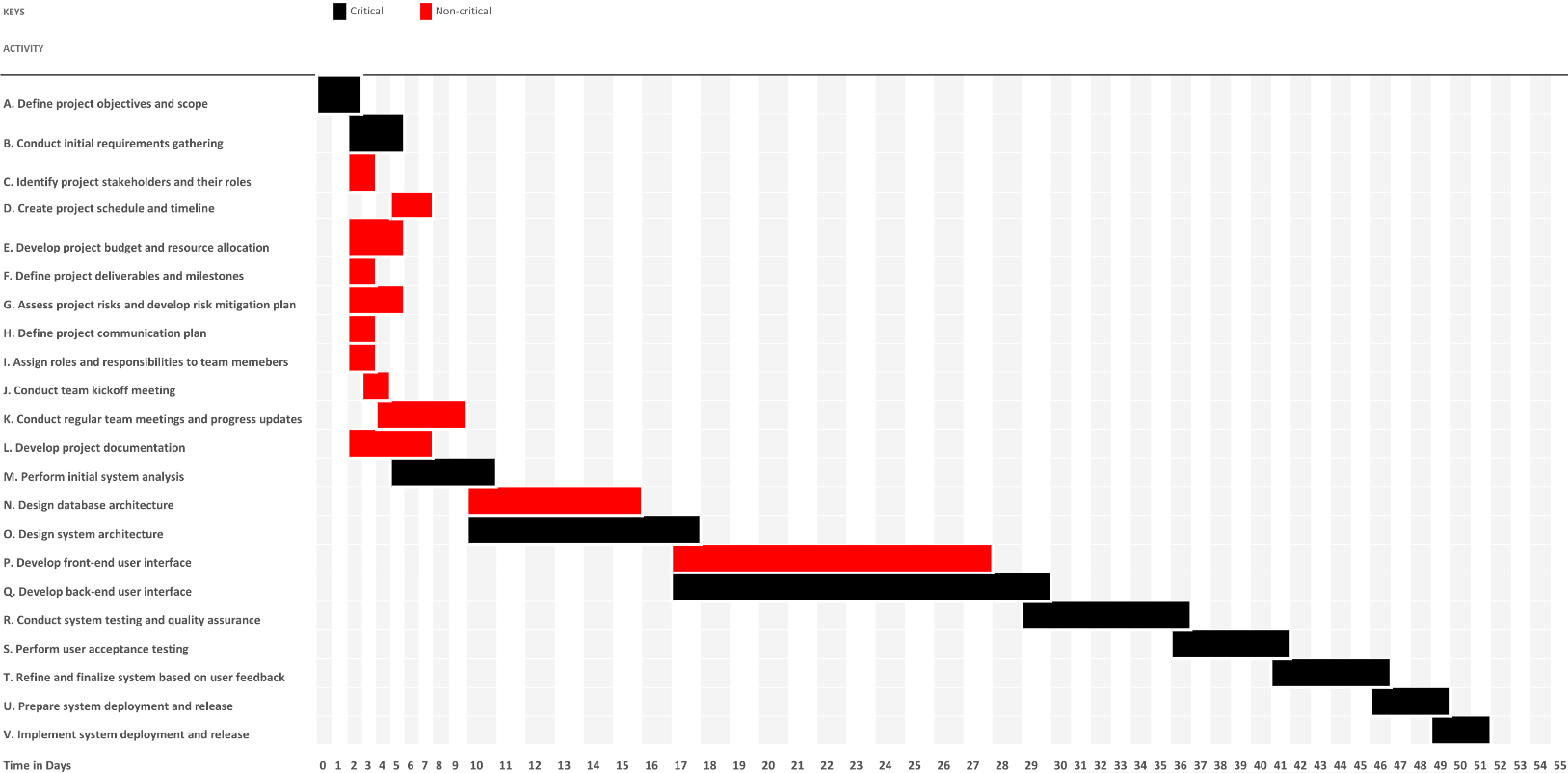
M. Tasks P and Q are dependent on the completion of tasks N and O. Task R is dependent on tasks P and Q, and task S depends on the completion of task R. Finally, tasks T, U, and V are dependent on the completion of task S. The critical path of the project includes tasks A, B, M, O, Q, R, S, T, U, and V. These tasks are crucial for the project's completion within the estimated timeframe.

The critical path represents the most extended sequence of activities that need completion to finish Team Vault in the minimum possible time. In Figure 1, starting from activity A, it is followed by activity B, which leads to activity M. Activity M is then followed by activity N,

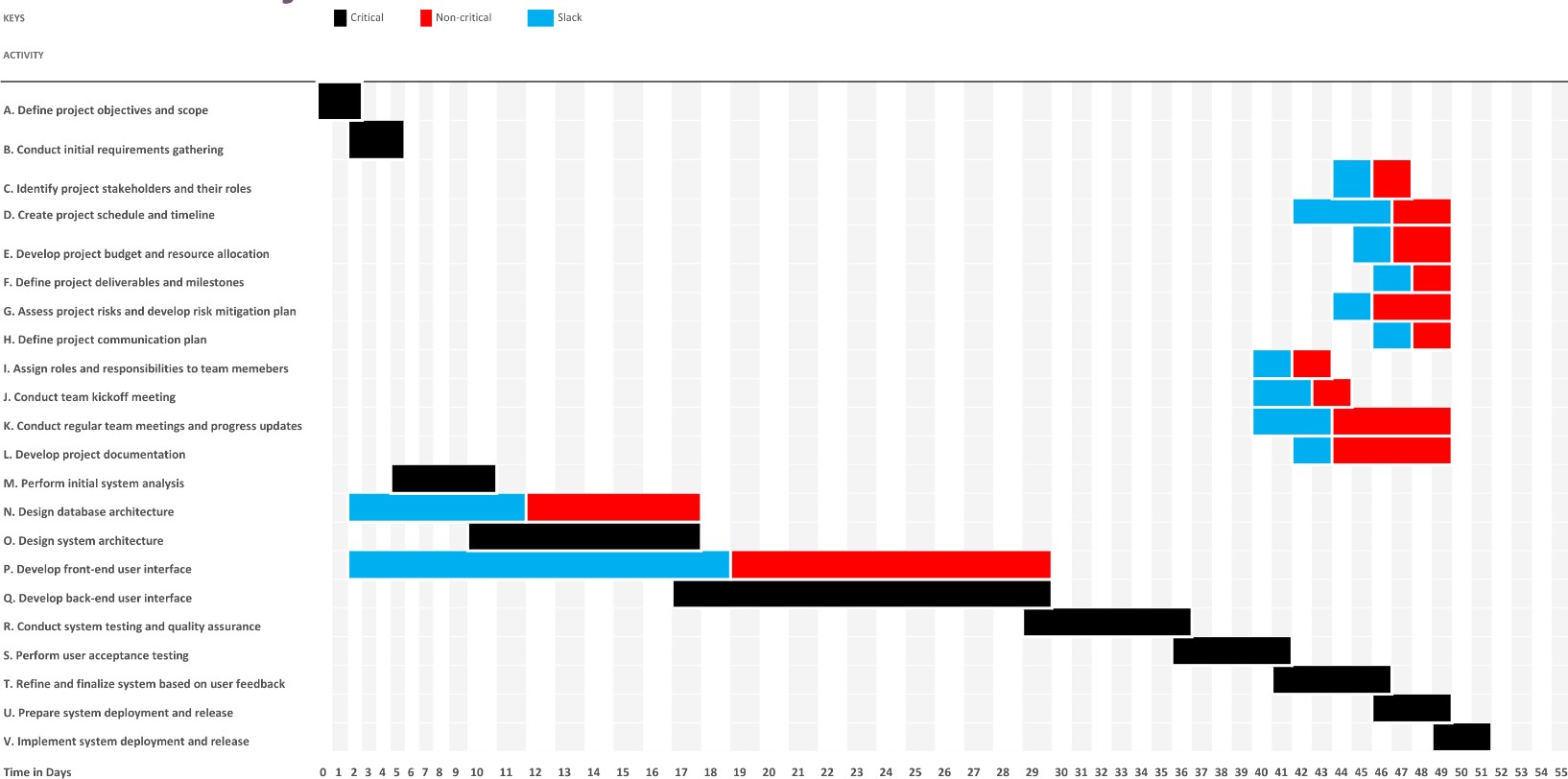
which further connects to activity O. From O, the path continues to activity Q, then to activity R, S, T, U, and finally concludes with activity V.

## Gantt Chart

### Project Planner



***Figure 41 - Team Vault Project Planner (ES/EF)***



***Figure 42 - Team Vault Project Planner (LS/LF)***

The Gantt chart provides a visual representation of the schedule of Team Vault. Figure 41 represents the project's Early Start (ES) and Early Finish (EF); both represent the earliest sequence of time for activities for the project timeline. Figure 3 offers a similar format but adds variable slack along with Late Start (LS) and Late Finish (LF) to the project's timeline. LS and LF offer the latest possible time to complete activities for the project, while slack represents the delaying of activities. As shown by Figures 41 and 42, a more visual representation of Figure 40, there is a mix of critical and non-critical activities linked to the project's overall duration. Due to the length of the slack being comparable to the course of non-critical activities, this allows Team Vault to prioritize the completion of critical tasks over non-critical tasks by allocating significant resources to them while still having enough time to complete non-critical tasks, ensuring the completion of Team Vault within the allotted time.