Software Engineering
Software Requirements Specification (SRS) Document

IndieDev

09/20/2023

1.0

By: Jonathan Moreno, Maddison Tidball, Callie Hampton

My words and actions will reflect Academic Integrity.
I will not cheat or lie or steal in academic matters.
I will promote integrity in the UNCG community.

Table of Contents

1. Introduction

1.1. Purpose

[The goal of your project and the objectives it wishes to accomplish]

The goal of the IndieDev application is to allow Game Developers, Game Designers, Artists, Musicians, and Creatists (Creators) to collaborate on game projects or to share their expertise/experience. Furthermore, users can create their own profiles to share their portfolio that is public on the site. In doing so, these users have an application where they can find talent or information without having to go on various different sites.

1.2. Document Conventions

[Full description of the main objectives of this document in the context of your project.
Here's how you should begin this section:
"The purpose of this Software Requirements Document (SRD) is to..."
"In it, we will . . ., . . ., and . . .."]

The purpose of this Software Requirements Document (SRD) is to describe the client-view and developer-view requirements for IndieDev. Client-oriented requirements describe the system from the client's perspective. These requirements include a description of the different types of users served by the system. Developer-oriented requirements describe the system from a software developer's perspective. These requirements include a detailed description of functional, data, performance, and other important requirements.

1.3. Definitions, Acronyms, and Abbreviations

[Include any specialized terminology dictated by the application area or the product area.
For example:]

Java	A programming language originally developed by James Gosling at Sun Microsystems. We will be using this language to build the Restaurant Manager.
MySQL	Open-source relational database management system.
.HTML	Hypertext Markup Language. This is the code that will be used to structure and design the web application and its content.
SpringBoot	An open-source Java-based framework used to create a micro Service. This will be used to create and run our application.
MVC	Model-View-Controller. This is the architectural pattern that will be used to implement our system.
Spring Web	Will be used to build our web application by using Spring MVC. This is one of the dependencies of our system.
Thymeleaf	A modern server-side Java template engine for our web environment. This is one of the dependencies of our system.
IntelliJ	An integrated development environment (IDE) for Java. This is where our system will be created.
API	Application Programming Interface. This will be used to implement a function

	within the software ...

1.4. Intended Audience

[Describe which part of the SRS document is intended for which reader. Include a list of all stakeholders of the project, developers, project managers, and users for better clarity.]

The intended audience for this application will be Game Developers, Game Designers, Artists, Musicians, and Creatists (Creators).

1.5. Project Scope

[Specify how the software goals align with the overall business goals and outline the benefits of the project to business.]

The goal of IndieDev is to provide a simple to use website to meet other indie creators but also provide simple yet complex profile and portfolio customization tools. IndieDev will be used to overcome the lack of connections between individual game developers and other creators.

The benefits of the project to business include:

- Reducing the stress of game development by connecting creators who want to contribute to a project but may not be proficient in other fields to creators in those other fields.
- Increasing the popularity of individual developers by including user interactions like those in social media
- Obtaining experience by interacting and meeting other creators in the wide spanning umbrella of game development

1.6. Technology Challenges

[Any technological constraints that the project will be under. Any new technologies you may need to use]

1.7. References

[Mention books, articles, web sites, worksheets, people who are sources of information about the application domain, etc. Use proper and complete reference notation. Give links to documents as appropriate. You should use the APA Documentation model (Alred, 2003, p. 144).]

General Description

1.8. Product Perspective

[Describe the context and origin of the product.]

IndieDev was created to connect game developers to other creators under the game development umbrella. Many game developers might not be proficient in other aspects of game development, such as 3D modeling or composing scores or sound effects. IndieDev will provide a space to overcome these challenges by connecting creators.

1.9. Product Features

[A high-level summary of the functions the software would perform and the features to be included.]

The application features include the ability to create posts, like, and reply to posts. This allows users to interact with each other on posts. Another feature is users can create profiles with the portfolio feature. This allows users to display their portfolio on their profile for users to view. In addition, users can report and/or block other users. Users can chat with other users or create group chats to collaborate on projects. For administrators, they can review reports and determine the punishment such as lock or remove users. In addition, administrators can add/update the terms of service for the site (with agreement from other admins). As for moderators, they can pause accounts if a violation of the TOS was found. They can lock profiles as well. For both admins and mods, they can remove or lock posts.

1.10. User Class and Characteristics

[A categorization and profiling of the users the software is intended for and their classification into different user classes]

IndieDev will prompt users to select their areas of expertise when they first create their profile. This allows other users to be searched by classification and connections between game developers and creators will be fastly made.

1.11. Operating Environment

[Specification of the environment the software is being designed to operate in.]

The web application is designed to operate on the web across different operating systems.

1.12. Constraints

[Any limiting factors that would pose challenge to the development of the software. These include both design as well as implementation constraints.]

Due to the security of the system, users will only be allowed to upload text, picture, and video files.

1.13. Assumptions and Dependencies

[A list of all assumptions that you have made regarding the software product and the environment along with any external dependencies which may affect the project]

Our software will be dependent on Spring Web and Thymeleaf in order to create and run the web application that will be developed in IntelliJ.

We will be using the Google Translate API to translate the user functionalities on the front end, while also translating people posts to their native language.

2. Functional Requirements

[Statements of services the system should provide, how the system should react to particular inputs and how the system should behave in particular situations.]

2.1. Primary

[All the requirements within the system or sub-system in order to determine the output that the software is expected to give in relation to the given input. These consist of the design requirements, graphics requirements, operating system requirements and constraints if any.]

- FR0: The system will allow the user to upload text, image and video files and post them to the site.
- FR1: The system will allow the user to send messages to other users
- FR2: The system will allow the user to report other users

2.2. Secondary

[Some functions that are used to support the primary requirements.]

- Administrators will be allowed to ban users from the site
- Moderators will be allowed to put pauses on accounts

3. Technical Requirements

4.1. Operating System and Compatibility

The application will be compatible with any operating system that is able to view and interact with traditional web pages.

4.2. Interface Requirements

4.2.1. User Interfaces

[The logic behind the interactions between the users and the software. This includes the sample screen layout, buttons and functions that would appear on every screen, messages to be displayed on each screen and the style guides to be used.]

IndieDev will allow users to view, like, and comment on other users posts on the home page. The database will store information for each post made by a user and communicate changes to the front end. Users will be allowed to access their own and others porfolios and profiles through their respective buttons. Users can also access their chats via the “Chats” button. On the Portfolio screen, users can view past projects and posts.

4.2.2. Hardware Interfaces

[All the hardware-software interactions with the list of supported devices on which the software is intended to run on, the network requirements along with the list of communication protocols to be used.]

The web application will run on any comptuer operating sytem that has access to the internet, the ability to upload files, and can interact with the web page. This includes desktop computers and laptops.

4.2.3. Communications Interfaces

[Determination of all the communication standards to be utilized by the software as a part of the project]

The software must be able to communicate over the internet as well as to the local database on phpMyAdmin and our chosen API. ----------INSERT API NAME HERE

4.2.4. Software Interfaces

[The interaction of the software to be developed with other software components such as frontend and the backend framework to the used, the database management system and libraries describing the need and the purpose behind each of them.]

We will use Spring Boot ThymeLeaf to help build the frontend, as well as JPA for the backend database functionality. We will also use Spring Boot with Java to connect the frontend to the backend.

5. Non-Functional Requirements

[Constraints on the services or functions offered by the system (e.g., timing constraints, constraints on the development process, standards, etc.). Often apply to the system as a whole rather than individual features or services.]

5.1. Performance Requirements

[The performance requirements need to be specified for all the functional requirements.]

- NFR0: The system will allow the user to upload text, image and video files and post them to the site in less than 5 minutes.
- NFR1: The system will allow the user to send messages to other users
- NFR2: The system will allow the user to report other users
- NFR3: The system will consume no memory on the user's device.

5.2. Safety Requirements

[List out any safeguards that need to be incorporated as a measure against any possible harm the use of the software application may cause.]

- Children under the age of 13 will not be permitted to use the system.
- Posts or profiles containing explicit content must be flagged as such.
- The system will filter out explicit content from user feed by default.
- The system will not allow explicit content to be turned on for any user under the age of 18.
- The system will not allow hate speech of any kind.

5.3. Security Requirements

[Privacy and data protection regulations that need to be adhered to while designing of the product.]

- The system will only be usable by authorized users.
- Regular users will not be able to access the areas of the app deisignated for Admins or Moderators.
- No user will be able to access information of other users listed as private.
- Passwords will be implemented to stop users from logging into other users' accounts.
- Users will not be required to enter their full name to sign up.
- User emails will be private by default.
- Private projects should not be able to be accessed by users not affiliated with the founding group or individual of said project.

5.4. Software Quality Attributes

[Detailing on the additional qualities that need to be incorporated within the software like maintainability, adaptability, flexibility, usability, reliability, portability etc.]

5.4.1. Availability

[Details]

The system must be available to all users 99% of the time unless prior notice of maintenance downtime is given at least one hour in advance.

5.4.2. Correctness

[Details]

The system will follow the design plans and function as intended without errors.

5.4.3. Maintainability

[Details]

The mean time to restore the system following a system failure must not be greater than 24 hours.

5.4.4. Reusability

[Details]

The functionality of the system's ability to allow users to make posts can be reused on other social media systems. The functionality of the system's ability to send messages to other uses can be reused for other messaging systems.

5.4.5. Portability

[Details]

The system will be accessable via any Linux, Windows, or Apple personal computer by the use of any working web browser.

5.5. Process Requirements

5.5.1. Development Process Used

[Software Process Model]

Scrum subset of Agile Processing Model.

5.5.2. Time Constraints

The system prototype should be ready by October 3, 2023. The system should be fully functional and running by December 5, 2023.

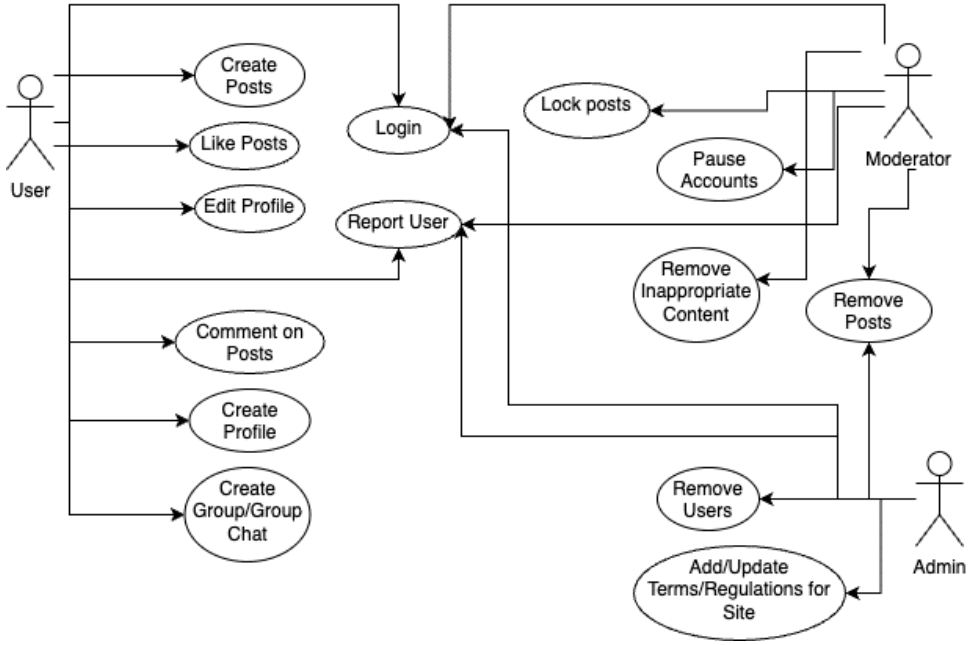
5.5.3. Cost and Delivery Date

The system will not have any monetary cost. The Delivery Date is December 5, 2023.

5.6. Other Requirements

- The time it takes for a new user to locate any features should not exceed 5 minutes.
- The time it takes for an experienced user to locate any feature should not exceed 1 minute.
- The error rate of users making a post must not exceed 10%.

5.7. Use-Case Model Diagram



5.8. Use-Case Model Descriptions

5.8.1. Actor: User (Maddison)

- **Create Posts**: User can upload text, image and video files and post them to the site
- **Like Posts**: User can like posts to provide feedback to other users
- Edit Profile: User can makes changes to their own profile
- Login: User can login to their account
- Report User: User can report another user for inappropriate behavior
- Comment on Posts: User can comment on other users' posts
- Create Profile: User can create a profile for their account
- Create Group/Group Chat: User can create a group for collaboration or chat with other users via a private group messaging feature

5.8.2. Actor: Moderator (Jonathan)

- Login: Moderator can login to their account
- Pause Accounts: Moderator can put a pause user accounts of different lengths for offenses
- Remove Inappropriate Content: Moderator can remove content deemed inappropriate or hateful
- Remove Posts: Moderator can remove posts deemed inappropriate or hateful

5.8.3. Actor: Administrator (Callie)

- Remove Posts: Administrator can remove posts deemed inappropriate or hateful
- **Login**: Administrator can login to their account
- Remove Users: Administrator can remove User accounts for too many violations to user guidelines
- Add/Update Terms/Regulations for Site: Administrators can add or update the user terms and regulations for the system.

5.9. Use-Case Model Scenarios

5.9.1. Actor: User (Maddison)

- **Use-Case Name**: Users registering an account
    - **Initial Assumption**: A user registered an account
    - **Normal**: a user doesn't require assistance
    - **What Can Go Wrong:** A user requests for their account information to be changed
    - **Other Activities**: Users entered invalid data
    - **System State on Completion**: The account is created
- **Use-Case Name**: Users report a post
    - **Initial Assumption**: Users report a post with a valid request
    - **Normal**: The report is reviewed
    - **What Can Go Wrong**: The report is false
    - **Other Activities**: Users report multiple posts for the same reasons
    - **System State on Completion**: The report is reviewed and decided to be removed
- **Use-Case Name**: User sends a message
    - **Initial Assumption**: Users send a valid message
    - **Normal**: The message is delivered to the correct recipient
    - **What Can Go Wrong**: The message is not sent, inappropriate content
    - **Other Activities**: User sends many duplicate messages
    - **System State on Completion**: The message is sent, key word delivered is displayed

5.9.2. Actor: Moderator (Jonathan)

- **Use-Case Name**: Mod pause an account
    - **Initial Assumption**: Mod pauses an account for a valid reason
    - **Normal**: The account is being paused due to an investigation or violation of TOS
    - **What Can Go Wrong**: Mod can pause the incorrect account or falsely pause an account
    - **Other Activities**: Mods can review reports to decide if an account should be paused
    - **System State on Completion**: The account is paused
- **Use-Case Name**:Mod removes inappropriate content
    - **Initial Assumption**: A user posted inappropriate content
    - **Normal**: The mod removes that content
    - **What Can Go Wrong**: the mod can abuse their power
    - **Other Activities**: Mod can remove the wrong content
    - **System State on Completion**: The content is removed

5.9.3. Actor: Administrator (Callie)

- **Use-Case Name**: Remove Posts
    - **Initial Assumption**: The Administrator removes a post.
    - **Normal**: The Administrator removes an inappropriate post.
    - **What Can Go Wrong**: The Administrator cannot find where to remove the post.
    - **Other Activities**: The Administrator removes several posts at once.
    - **System State on Completion**: The targeted post was removed.
- **Use-Case Name**: Add/Update Terms/Regulations for Site
    - **Initial Assumption**: The Administrator adds a new regulation for the site.
    - **Normal**: The Administrator adds a new regulation and the site's terms and regulations page is updated with a notification sent out to all users.

- **What Can Go Wrong**: The Administrator accidentally deletes a regulation.
- **Other Activities**: The Administrator adds several regulations.
- **System State on Completion**: The new regulation has been added.

6. Design Documents

6.1. Software Architecture

Folder: User
- Class: User
- Interface: UserRepo
- Class: UserController
    - registerUser
    - login
- Class: UserService
    - getAllUsers
    - getUsersByRole

Folder: Post
- Class: Post
- Interface: PostRepo
- Class: PostController
    - likePost
    - deletePost
    - commentPost
- Class: PostService
    - getAllPosts
    - getPostByUser

Folder: Report
- Class: Report
- Interface: ReportRepo
- Class: ReportController
    - createReport
- Class: ReportService
    - getAllReports
    - getAllReportsByUser

Folder: Project
- Class: Project
- Interface: ProjectRepo
- Class: ProjectController
    - Methods: deleteProject
    - createProject
- Class: ProjectService
    - getAllProjects
    - getAllProjectsByUser

Folder: Message

- Class: Message
- Interface: MessageRepo
- Class: MessageController
    - sendMessage

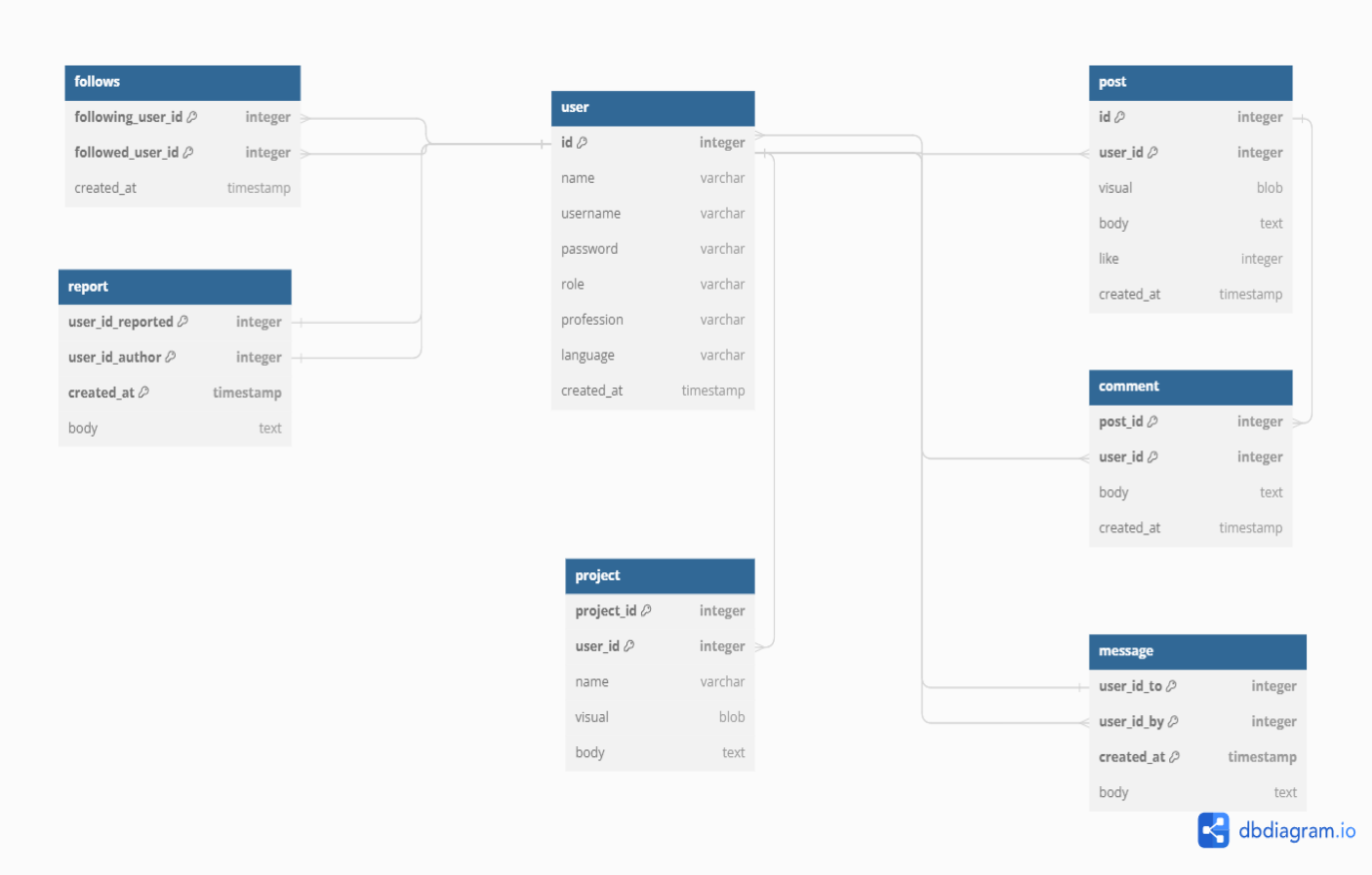
- Class: MessageService
    - getAllMessages
    - getAllMessagesSentBy
    - getAllMessagesSentTo

Folder: Security

- Class: SecurityConfig

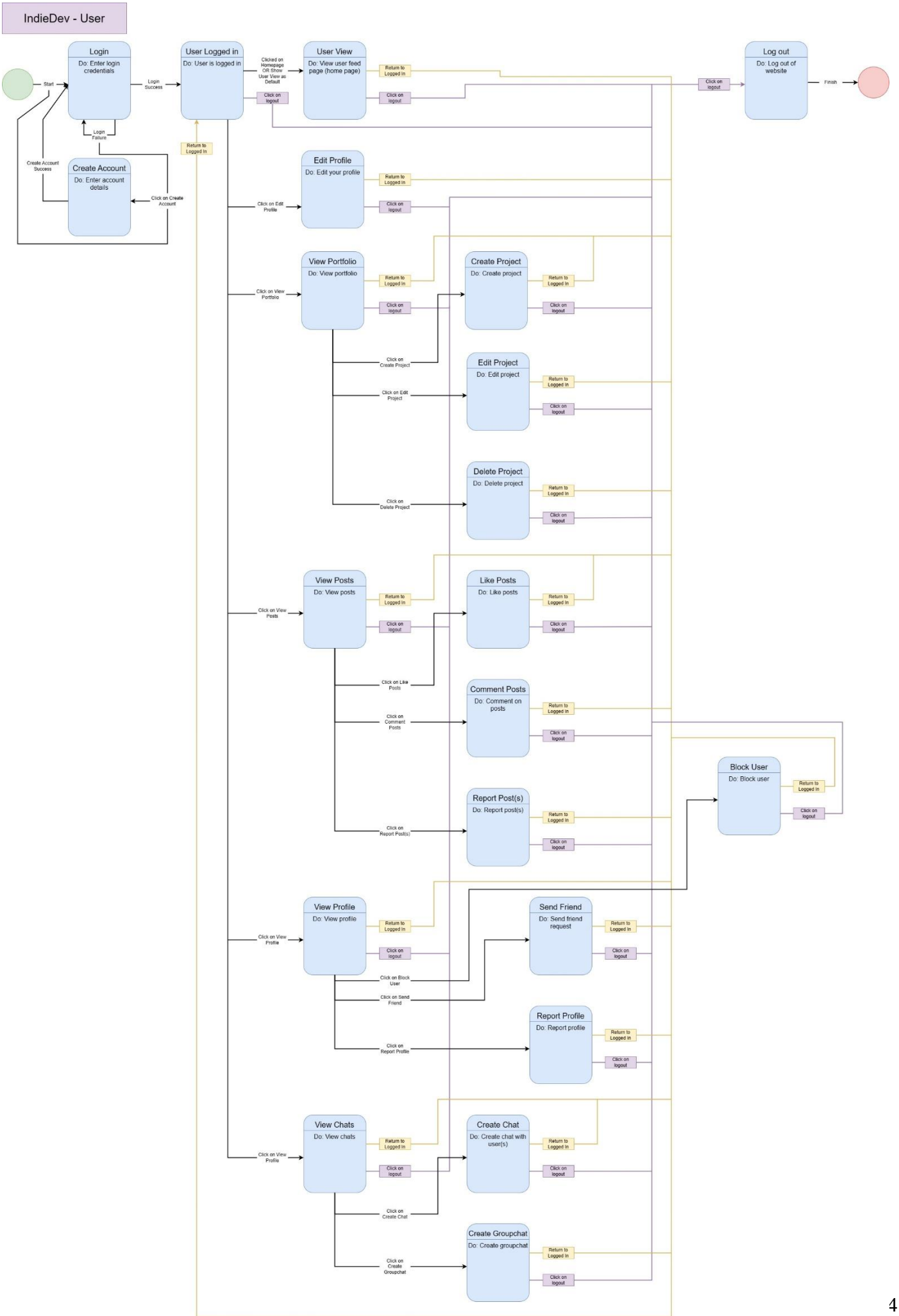
Class: Router (used for home page and log in directories)

6.2. High-Level Database Schema

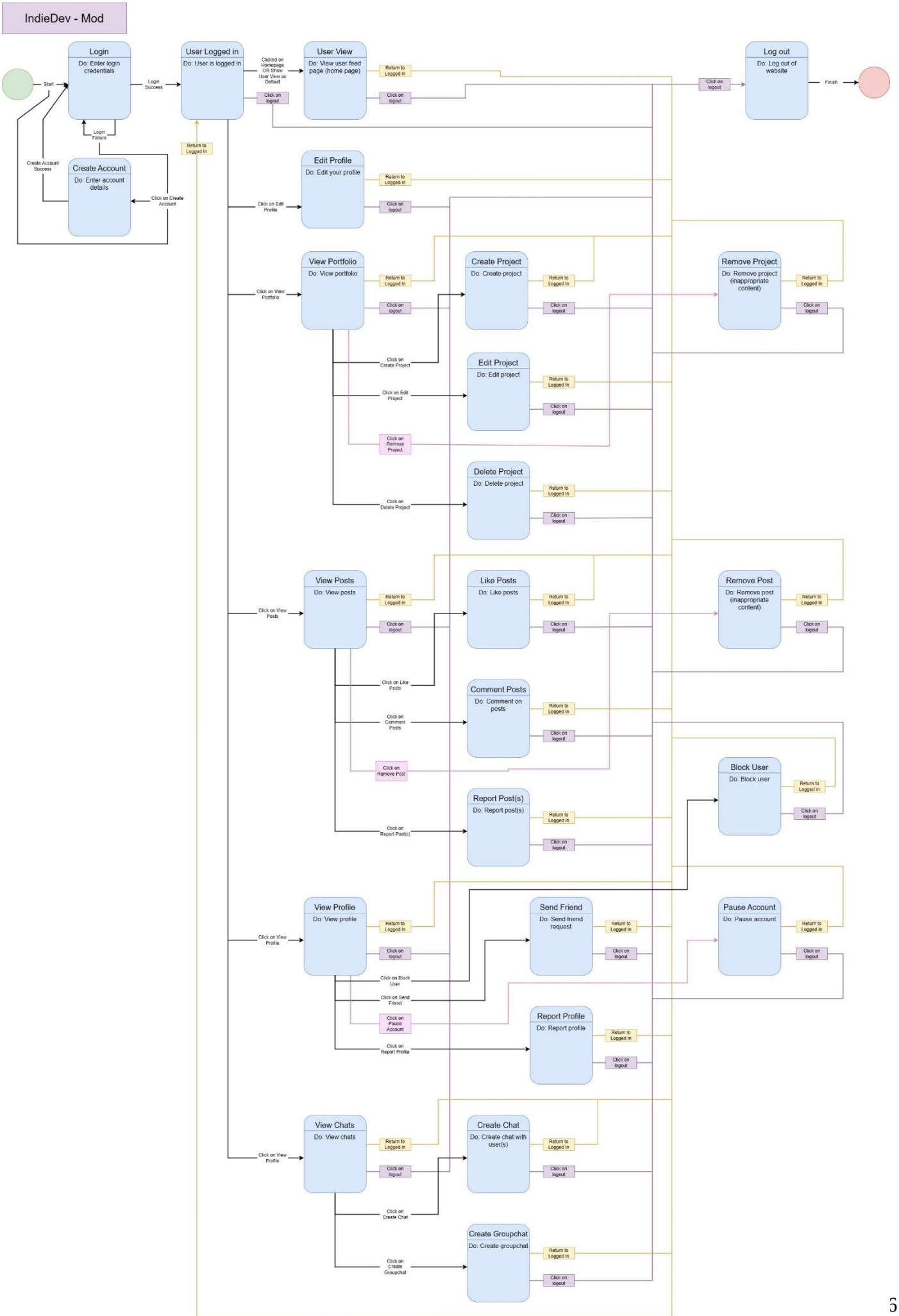


6.3. Software Design

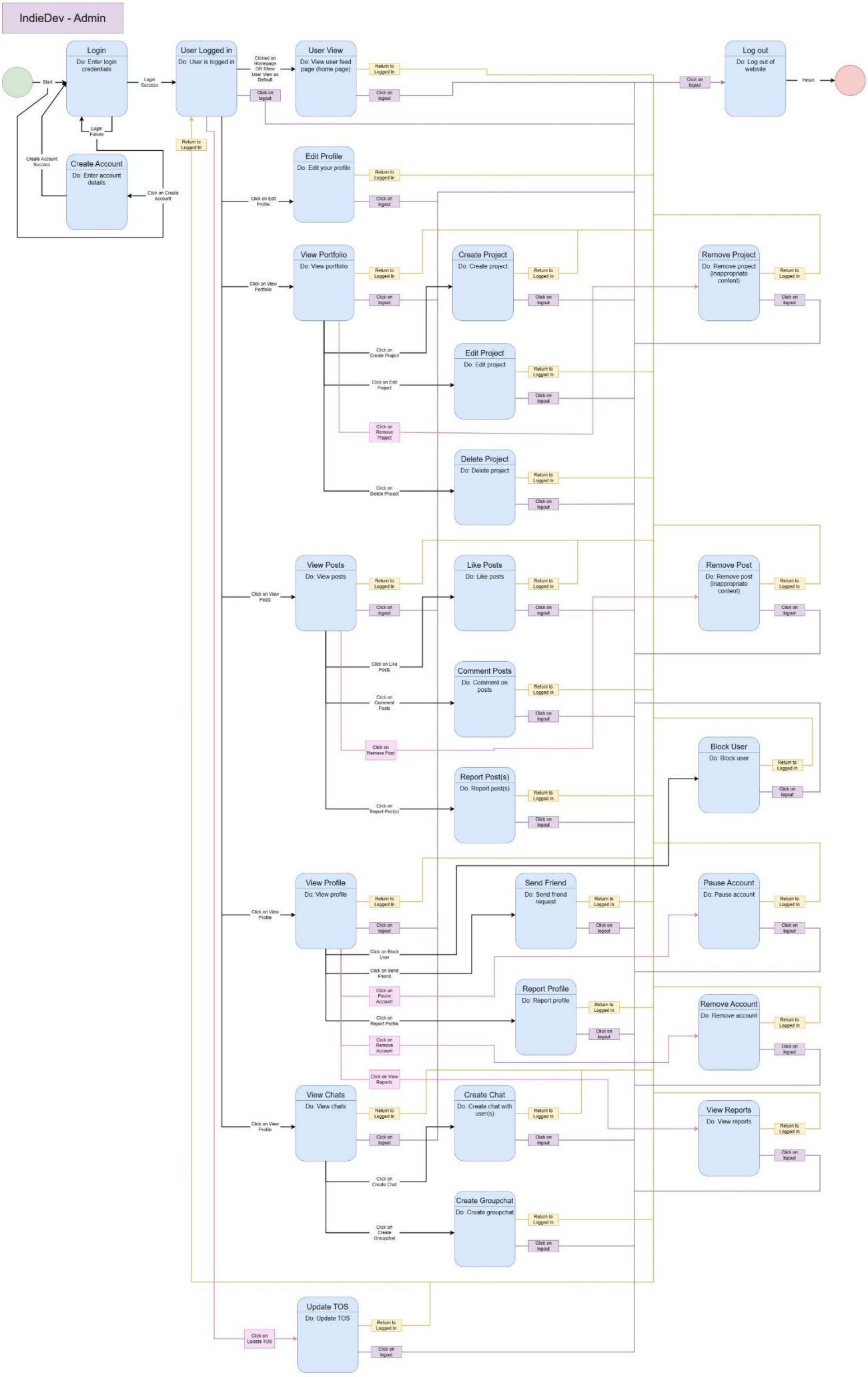
6.3.1. State Machine Diagram: User (Maddison)


IndieDev - User
Start
Login
Do: Enter login credentials
Login Success
Login Failure
Create Account Success
Create Account
Do: Enter account details
Click on Create Account
User Logged in
Do: User is logged in
Clicked on Homepage OR Show User View as Default
Click on logout
Return to Logged In
User View
Do: View user feed page (home page)
Return to Logged In
Click on logout
Click on logout
Log out
Do: Log out of website
Finish
Edit Profile
Do: Edit your profile
Return to Logged In
Click on logout
Click on Edit Profile
View Portfolio
Do: View portfolio
Return to Logged In
Click on logout
Click on View Portfolio
Create Project
Do: Create project
Return to Logged In
Click on logout
Click on Create Project
Edit Project
Do: Edit project
Return to Logged In
Click on logout
Click on Edit Project
Delete Project
Do: Delete project
Return to Logged In
Click on logout
Click on Delete Project
View Posts
Do: View posts
Return to Logged In
Click on logout
Click on View Posts
Like Posts
Do: Like posts
Return to Logged In
Click on logout
Click on Like Posts
Comment Posts
Do: Comment on posts
Return to Logged In
Click on logout
Click on Comment Posts
Report Post(s)
Do: Report post(s)
Return to Logged In
Click on logout
Click on Report Post(s)
Block User
Do: Block user
Return to Logged In
Click on logout
View Profile
Do: View profile
Return to Logged In
Click on logout
Click on View Profile
Send Friend
Do: Send friend request
Return to Logged In
Click on logout
Click on Block User
Click on Send Friend
Report Profile
Do: Report profile
Return to Logged In
Click on logout
Click on Report Profile
View Chats
Do: View chats
Return to Logged In
Click on logout
Click on View Profile
Create Chat
Do: Create chat with user(s)
Return to Logged In
Click on logout
Click on Create Chat
Create Groupchat
Do: Create groupchat
Return to Logged In
Click on logout
Click on Create Groupchat

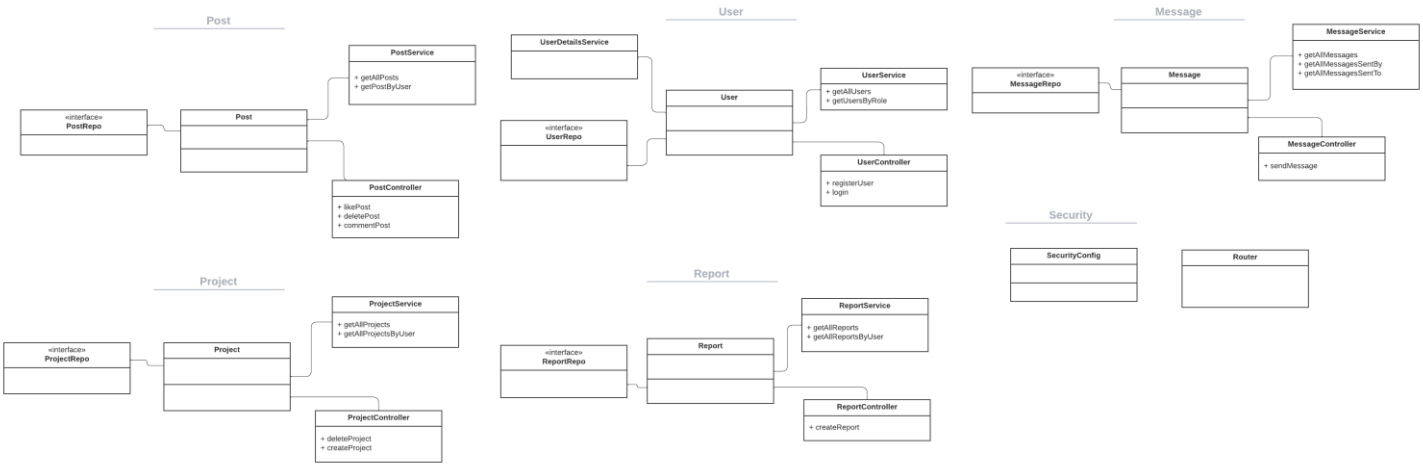
6.3.2. State Machine Diagram: Mod (Jonathan)


IndieDev - Mod
Start
Login
Do: Enter login credentials
Login Success
Login Failure
Create Account Success
Create Account
Do: Enter account details
Click on Create Account
User Logged in
Do: User is logged in
Clicked on Homepage OR Show User View as Default
Click on logout
Return to Logged In
User View
Do: View user feed page (home page)
Return to Logged In
Click on logout
Click on logout
Log out
Do: Log out of website
Finish
Edit Profile
Do: Edit your profile
Click on Edit Profile
View Portfolio
Do: View portfolio
Click on View Portfolio
Create Project
Do: Create project
Remove Project
Do: Remove project (inappropriate content)
Click on Create Project
Edit Project
Do: Edit project
Click on Edit Project
Click on Remove Project
Delete Project
Do: Delete project
Click on Delete Project
View Posts
Do: View posts
Click on View Posts
Like Posts
Do: Like posts
Remove Post
Do: Remove post (inappropriate content)
Click on Like Posts
Comment Posts
Do: Comment on posts
Click on Comment Posts
Click on Remove Post
Block User
Do: Block user
Report Post(s)
Do: Report post(s)
Click on Report Post(s)
View Profile
Do: View profile
Click on View Profile
Send Friend
Do: Send friend request
Pause Account
Do: Pause account
Click on Block User
Click on Send Friend
Click on Pause Account
Report Profile
Do: Report profile
Click on Report Profile
View Chats
Do: View chats
Click on View Profile
Create Chat
Do: Create chat with user(s)
Click on Create Chat
Create Groupchat
Do: Create groupchat
Click on Create Groupchat

6.3.3. State Machine Diagram: Admin (Callie)


IndieDev - Admin
Start
Login
Do: Enter login credentials
Login Success
Login Failure
Create Account
Do: Enter account details
Create Account Success
Click on Create Account
User Logged in
Do: User is logged in
Return to Logged In
Clicked on Homepage OR Show User View as Default
Click on logout
User View
Do: View user feed page (home page)
Log out
Do: Log out of website
Finish
Edit Profile
Do: Edit your profile
Click on Edit Profile
View Portfolio
Do: View portfolio
Click on View Portfolio
Create Project
Do: Create project
Remove Project
Do: Remove project (inappropriate content)
Click on Create Project
Edit Project
Do: Edit project
Click on Edit Project
Click on Remove Project
Delete Project
Do: Delete project
Click on Delete Project
View Posts
Do: View posts
Click on View Posts
Like Posts
Do: Like posts
Remove Post
Do: Remove post (inappropriate content)
Click on Like Posts
Comment Posts
Do: Comment on posts
Click on Comment Posts
Click on Remove Post
Block User
Do: Block user
Report Post(s)
Do: Report post(s)
Click on Report Post(s)
View Profile
Do: View profile
Click on View Profile
Send Friend
Do: Send friend request
Pause Account
Do: Pause account
Click on Block User
Click on Send Friend
Click on Pause Account
Report Profile
Do: Report profile
Click on Report Profile
Remove Account
Do: Remove account
Click on Remove Account
Click on View Reports
View Chats
Do: View chats
Click on View Profile
Create Chat
Do: Create chat with user(s)
View Reports
Do: View reports
Click on Create Chat
Create Groupchat
Do: Create groupchat
Click on Create Groupchat
Update TOS
Do: Update TOS
Click on Update TOS

6.4. UML Class Diagram



7. Scenario

7.1. Brief Written Scenario with Screenshots