

FPT Academy International

FPT – Aptech Computer Education

September – 2024

Center Name: ACE-HCMC-2-FPT.

Address: 590 Cach Mang Thang 8, District 3,

Ho Chi Minh City, Viet Nam.

Reddot

Software Report Requirement

Group:	Group 6	
Supervisor	Mr. Le Tuan Xuyen	
Class:	T1.2210.A1	
<u>No:</u>	<u>Full name</u>	<u>Roll No.</u>
1	Nguyen Trong Toan	student1387829

Table of Content

Introduction

Purpose

The purpose of this document is to outline the functional and nonfunctional requirements for *Reddot*, a mobile and web application aimed at providing a community-based question-and-answer platform like Stack Overflow. This document serves as a guide for the development team and stakeholders to ensure that all requirements are captured and met during the application development lifecycle.

Scope

Reddot is a cross-platform application, developed using the Flutter framework, that allows users to ask and answer questions, vote on content, and participate in discussions. The app is designed to run on Android, iOS, and the Web. Its primary objective is to facilitate knowledge sharing in various domains, providing a seamless user experience across devices.

Definitions, Acronyms, and Abbreviations

1. **Q&A:** Question and Answer
2. **UI:** User Interface
3. **UX:** User Experience
4. **CRUD:** Create, Read, Update, Delete
5. **API:** Application Programming Interface

Design Specification

Customer Requirement Specification

User Roles

Reddot will cater to the following user roles:

1. **Registered User:** Can post questions, answer, and vote.
2. **Guest User:** Can browse questions but cannot interact (post, vote, comment).
3. **Moderator:** Can moderate content (remove inappropriate questions, answers).
4. **Admin:** Full access to all features and controls, including user management.

User Stories

1. As a **Registered User**, I want to ask questions and receive answers from the community.
2. As a **Registered User**, I want to upvote or downvote questions and answers based on their usefulness.
3. As a **Moderator**, I want to moderate the content to ensure a healthy community.
4. As an **Admin**, I want to manage the overall system, including user accounts and application settings.

UI/UX Requirements

1. The interface must be intuitive and allow users to easily post, browse, and vote on questions.
2. The design must be responsive and adaptive to different screen sizes (iOS, Android, and Web).

Nonfunctional Requirement

These refer to the app's operational standards and constraints that must be met to ensure proper functionality:

1. **Performance:** The app must be responsive and handle large amounts of expense data without delays.
2. **Security:** All personal data must be encrypted in transit and at rest.
3. **Scalability:** The app should be capable of handling increasing numbers of users and data storage without performance degradation.
4. **Accessible:** The mobile app should have clear and legible fonts, user-interface elements, and navigation elements.
5. **User-friendly:** The design should follow Android UI/UX guidelines, with clear menus and other elements ensuring a smooth user experience and easy to understand.
6. **Availability:** The app must be available with an uptime of at least 99.9% per month.
7. **Cross-platform Compatibility:** The application should provide a consistent experience across iOS, Android, and Web platforms using the Flutter framework.

Functional Requirement

User Management

1. **Registration/Login:** Users can sign up via email or third-party services (Google, Facebook).
2. **Profile Management:** Users can update their profile information (bio, avatar, etc.).
3. **Password Reset:** Users can reset their passwords via email.

Question and Answer System

1. **Post Questions:** Registered users can post questions.
2. **Answer Questions:** Users can submit answers to existing questions.
3. **Voting System:** Registered users can upvote or downvote questions and answers.
4. **Commenting System:** Users can add comments to both questions and answers.
5. **Tagging:** Questions can be tagged with relevant topics.

Content Moderation

1. **Flagging System:** Users can flag inappropriate questions or answers.
2. **Moderation Tools:** Moderators can remove or edit content.
3. **Admin Dashboard:** Admins can manage users, questions, answers, and system settings.

Search and Filtering

1. **Search Functionality:** Users can search for questions based on keywords, tags, and categories.
2. **Filters:** Users can filter questions by tags, popularity, and recency.

Notifications

1. **In-App Notifications:** Users receive notifications for new answers, comments, or votes on their posts.
2. **Email Notifications:** Users receive email notifications for important updates (new answers, flagged content).

System Requirement

This section covers both hardware and software prerequisites for developing and running the application:

Frontend (Mobile and Web)

1. **Platform:** Flutter Framework
2. **Supported Platforms:**
 - Android: API level 21 (Lollipop) and above
 - iOS: iOS 12 and above
 - Web: Support for modern browsers (Chrome, Safari, Edge, Firefox)
3. **UI Framework:** Flutter Widgets

Backend

1. **Server-side:** Developed using Spring Boot Framework
2. **Database:** Firebase, MySQL
3. **Authentication:** JWT, OAuth 2.0 for third-party login (Google, Facebook)
4. **API:** RESTful API for communication between frontend and backend
5. **Hosting:** Cloud-based hosting

Development Tools

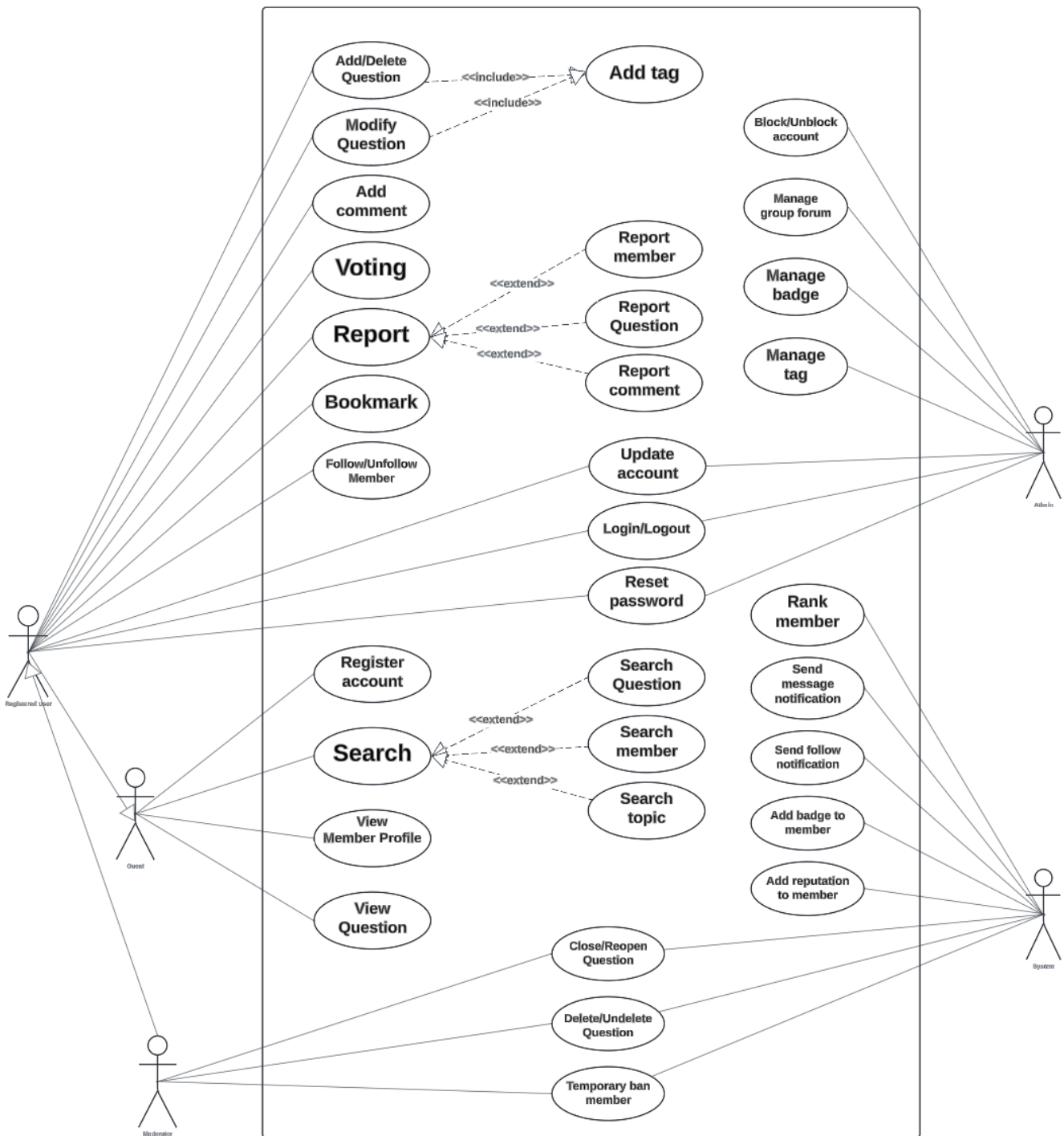
1. **IDE:** Android Studio or Visual Studio Code for Flutter development, IntelliJ for Spring Boot.
2. **Version Control:** Git for version control and collaboration.
3. **CI/CD Pipeline:** GitHub Actions or similar tools for continuous integration and deployment.

Testing

1. **Unit Testing:** Flutter's built-in testing framework
2. **UI Testing:** Automated UI testing using Flutter Test
3. **Backend Testing:** Postman for API testing

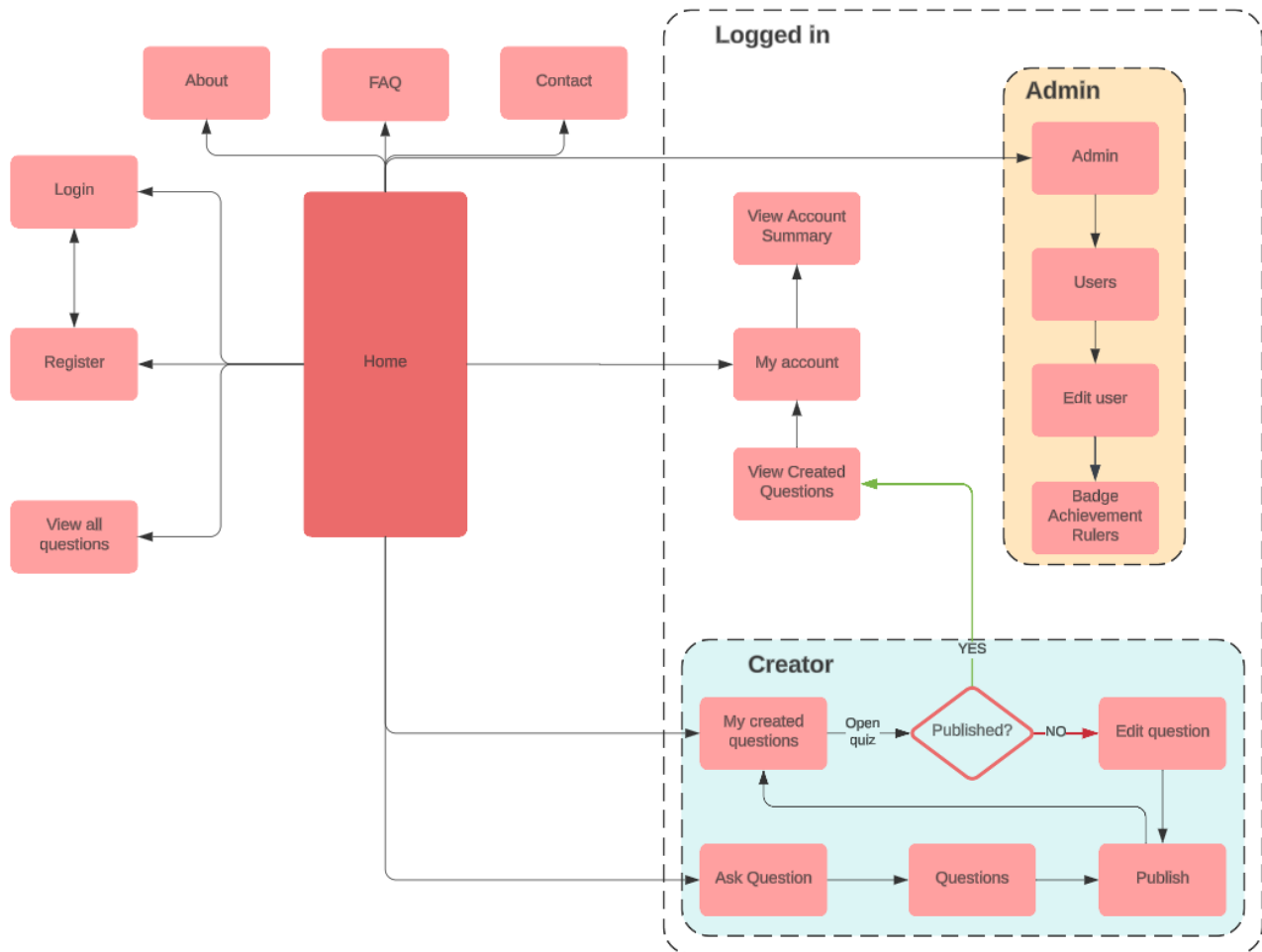
Software Diagram

Use Case Diagram

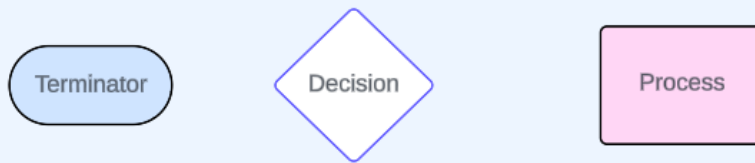


Flowchart

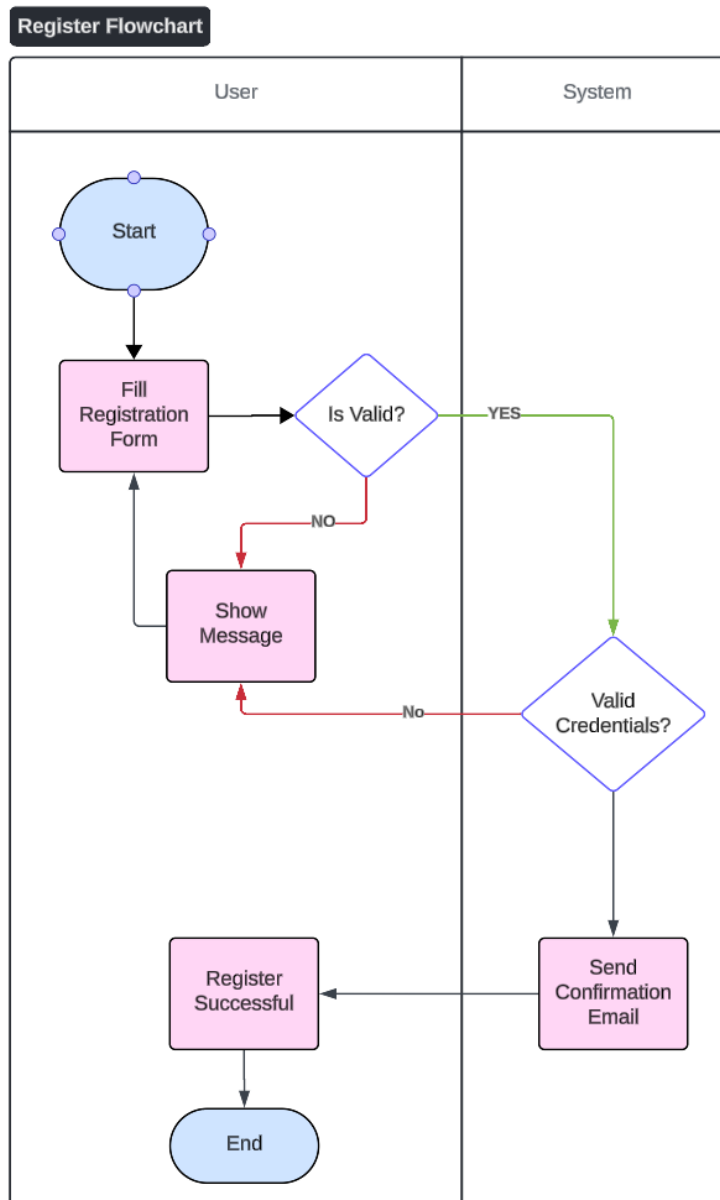
UI Flowchart



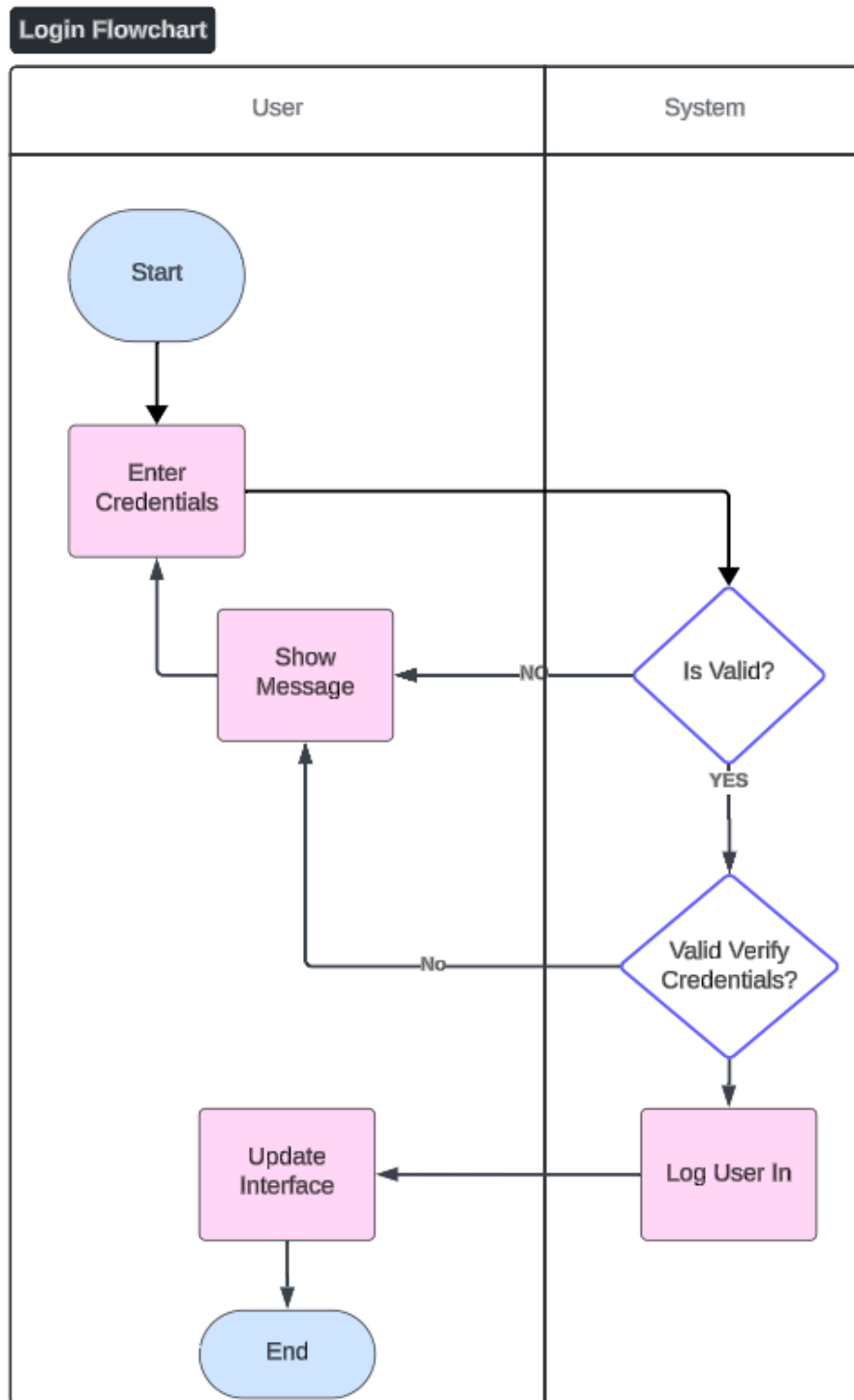
Process Flowcharts



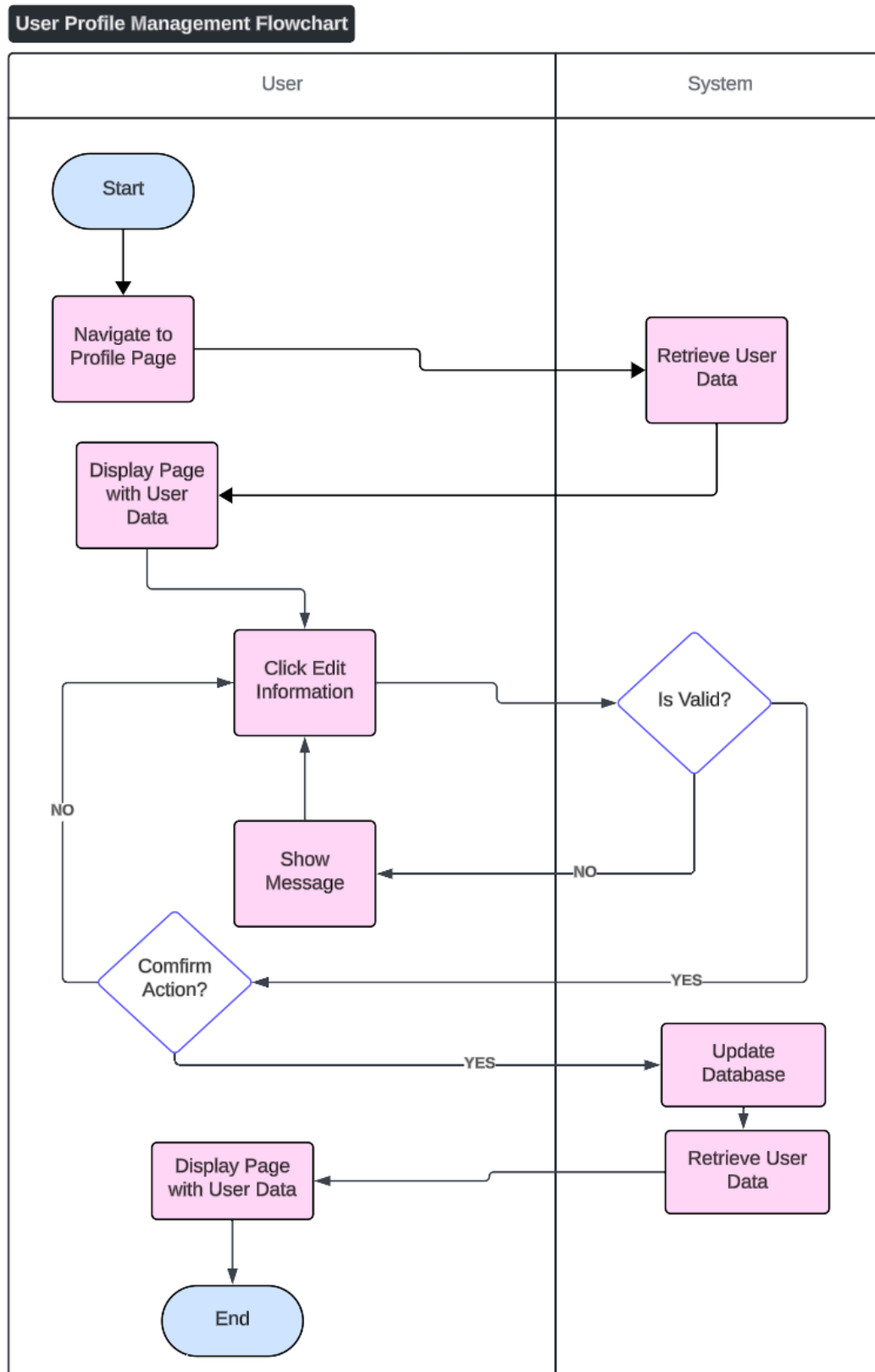
Registration Flowchart



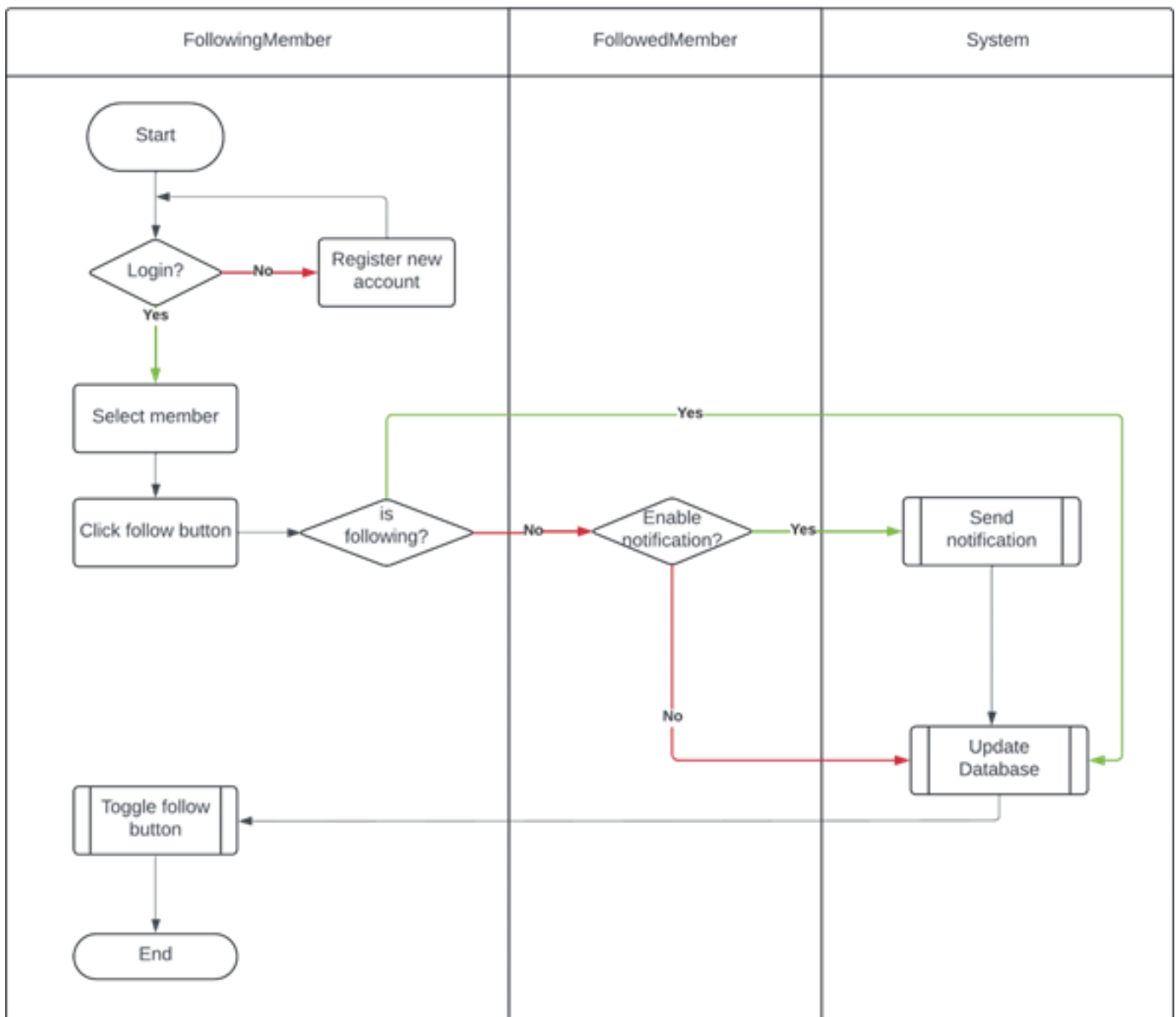
Login Flowchart



User Profile Management Flowchart

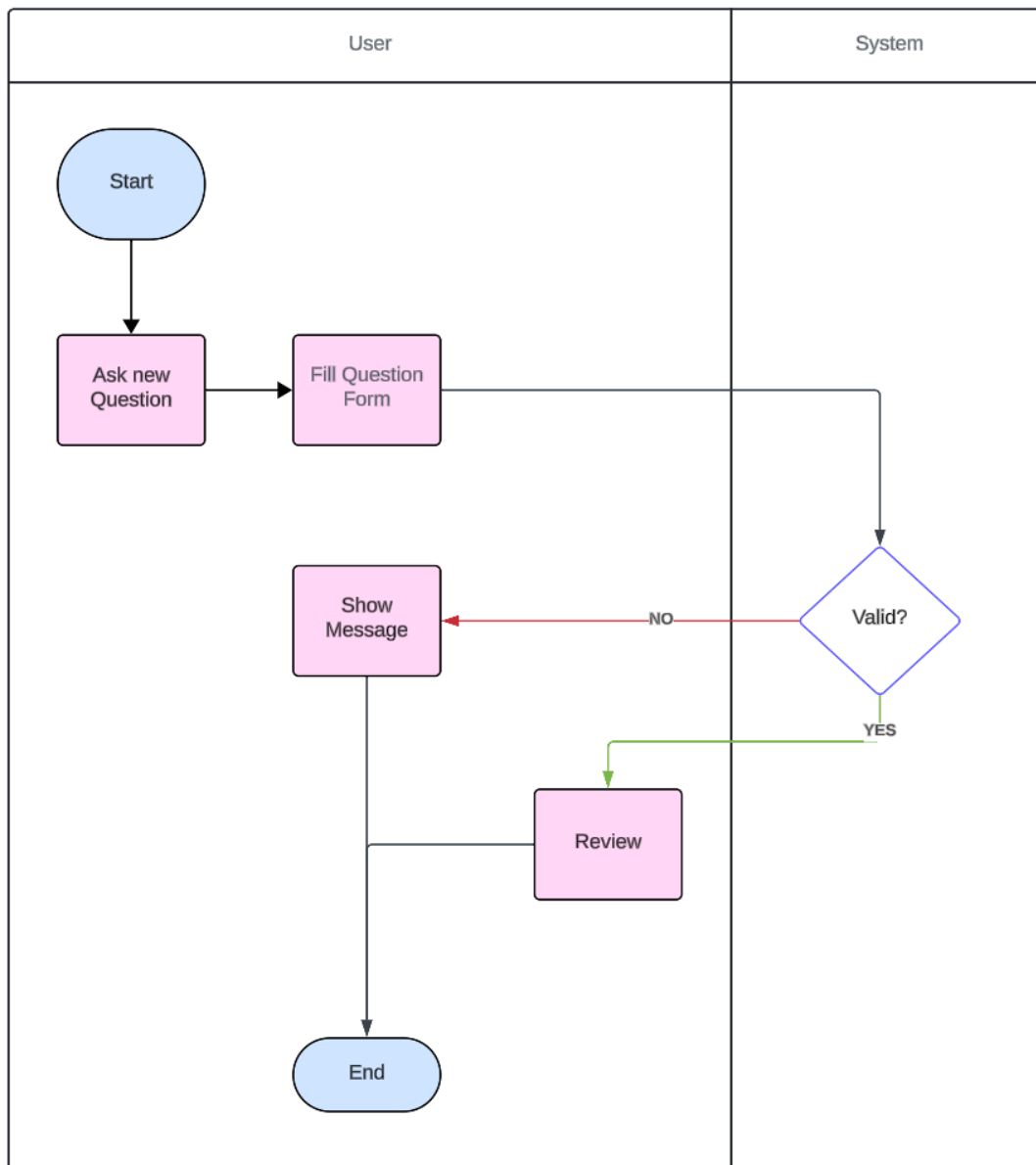


Follow or unfollow users (for users, moderators)



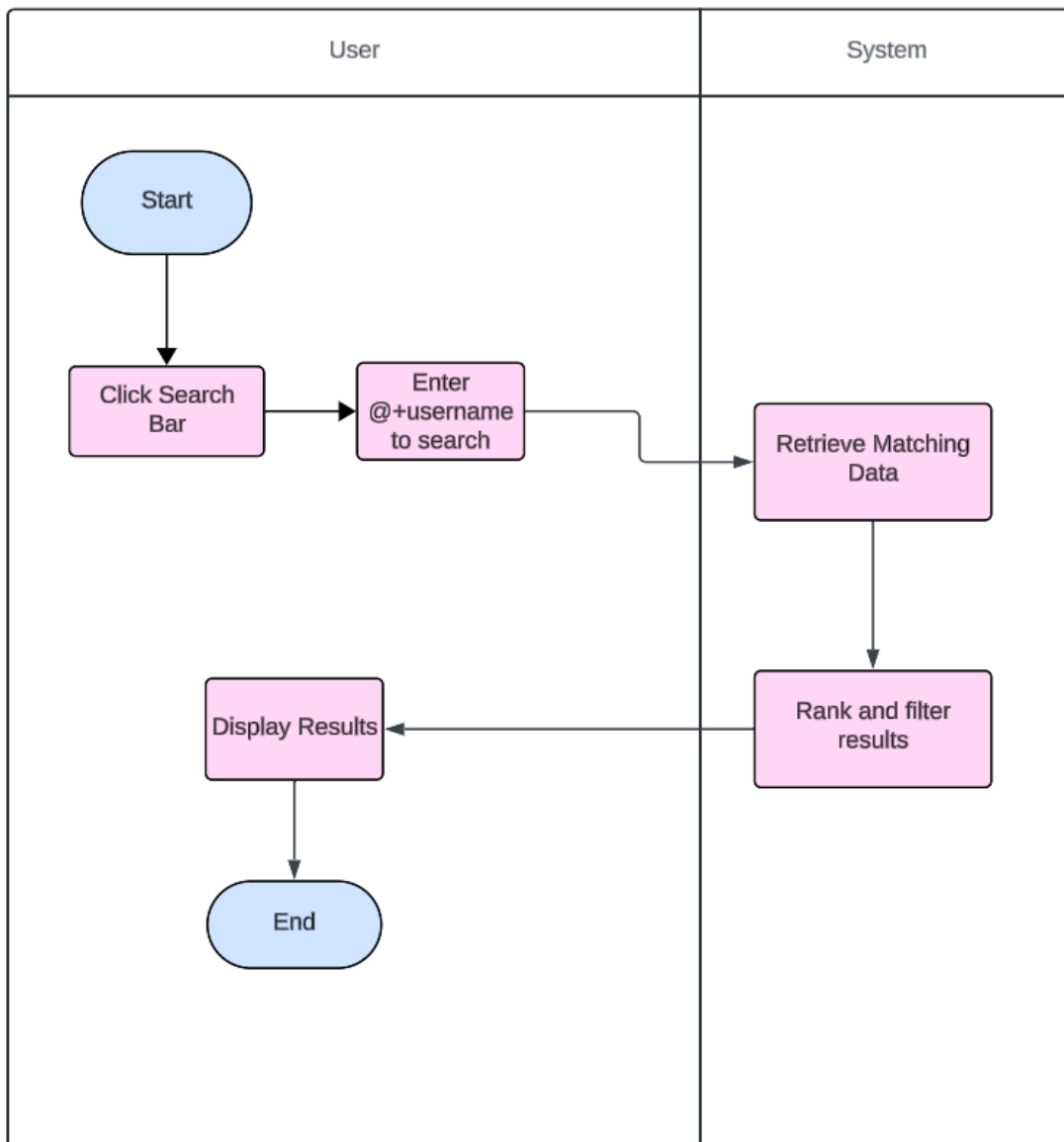
User Ask Question Flowchart

User Ask Question Flowchart



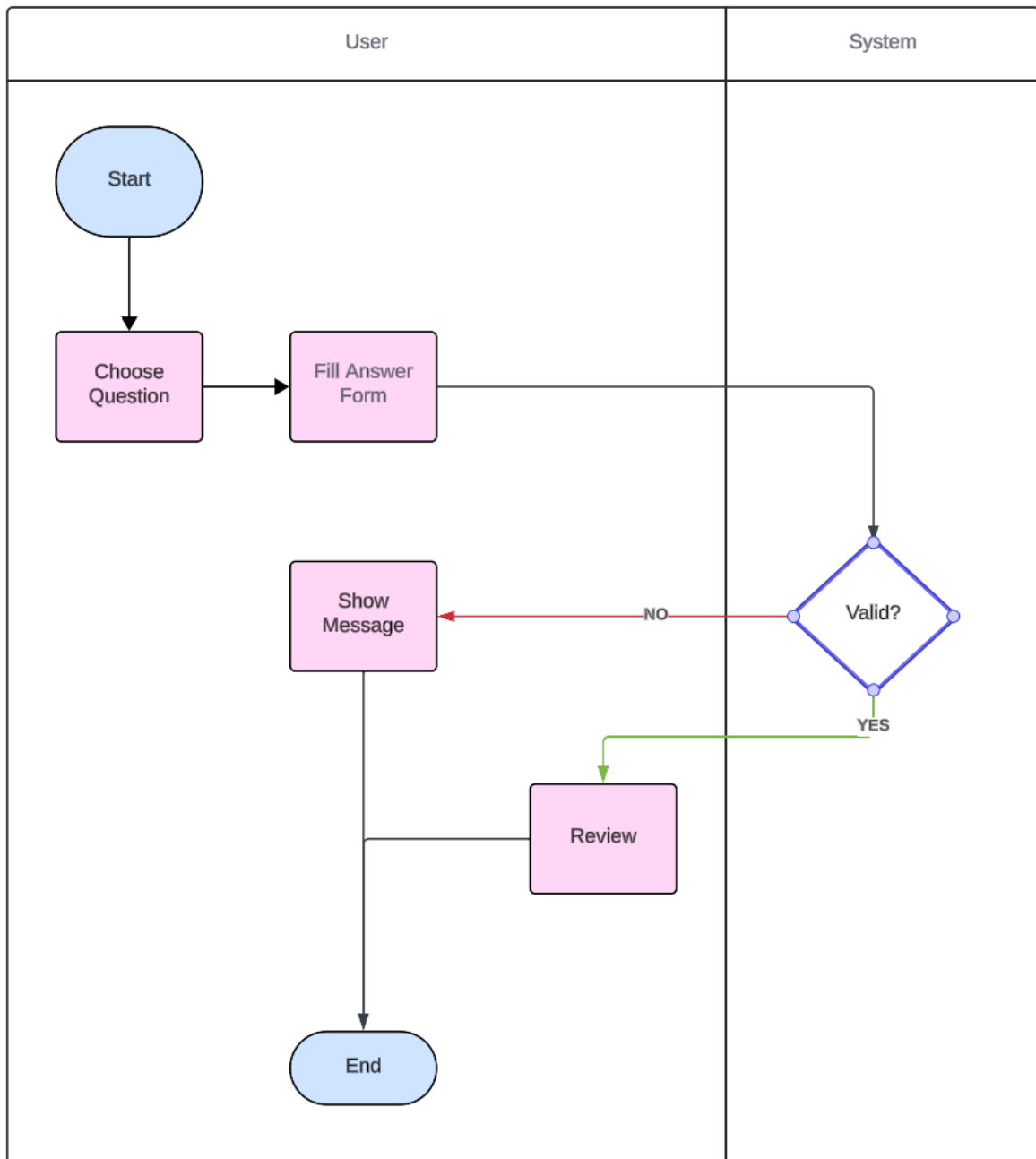
Search other Members

Search Member Flowchart



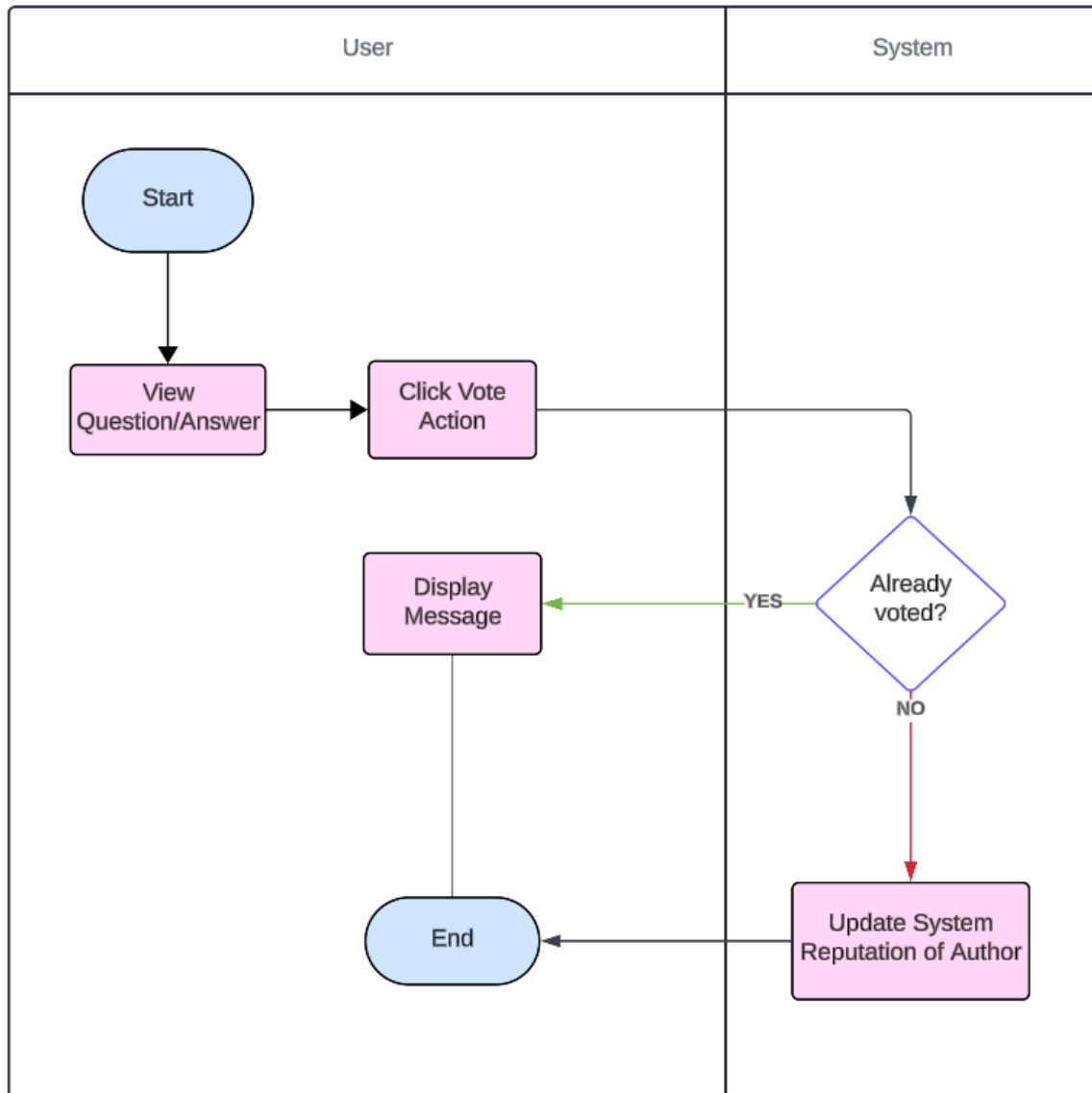
User Answer Question Flowchart

User Answer Question Flowchart

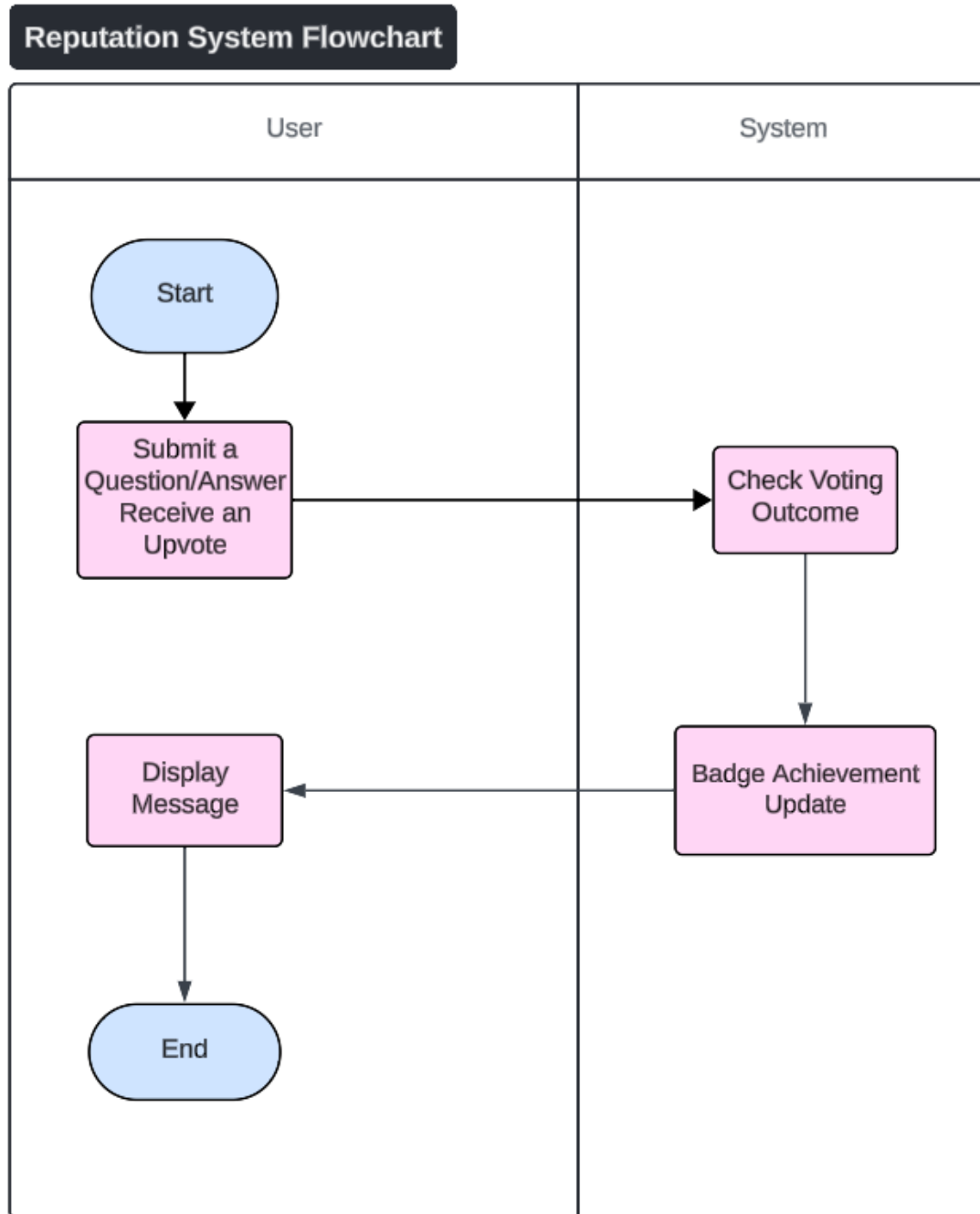


Voting on Questions/Answers Flowchart

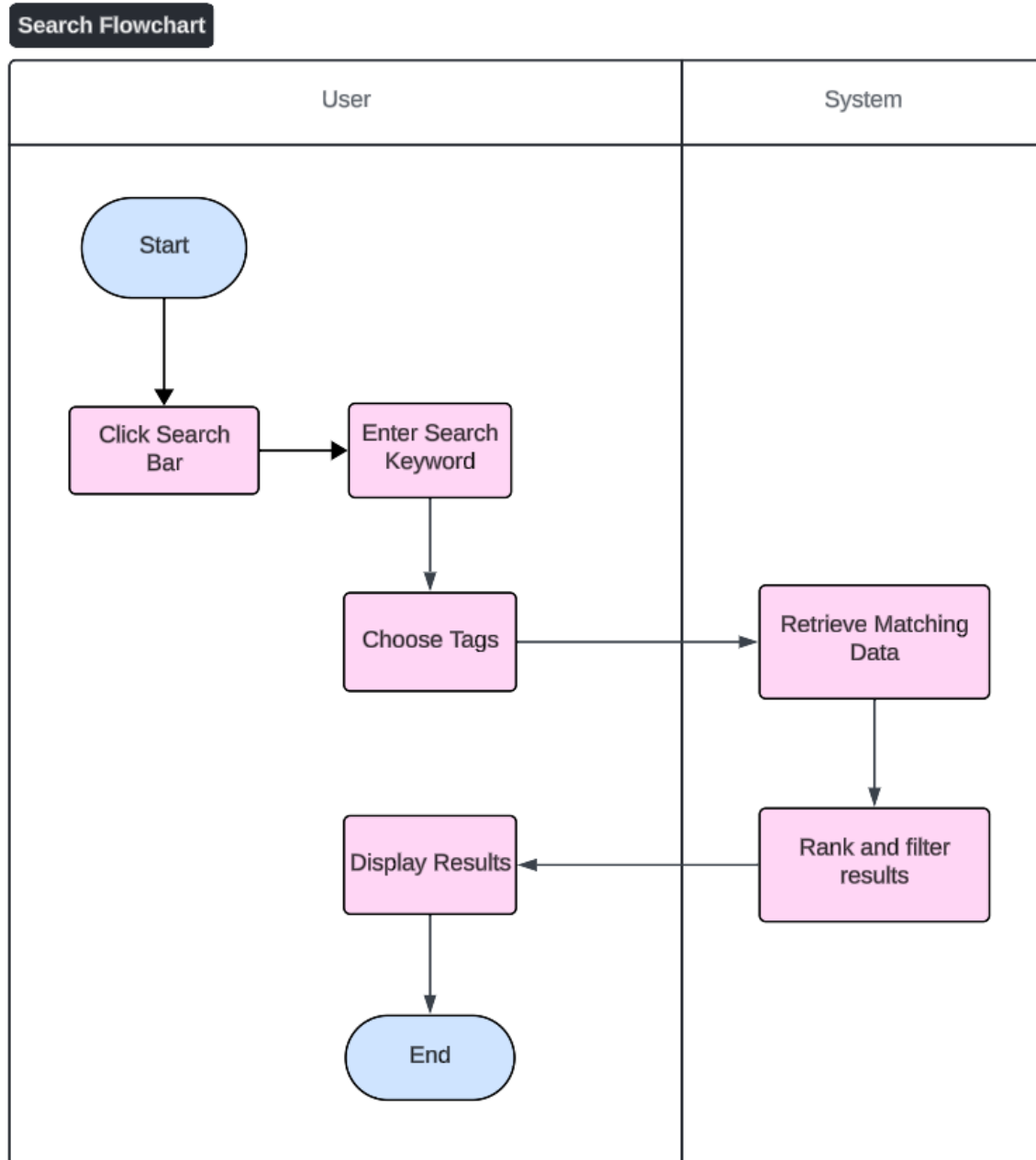
Voting on Questions/Answers Flowchart



Reputation System Flowchart

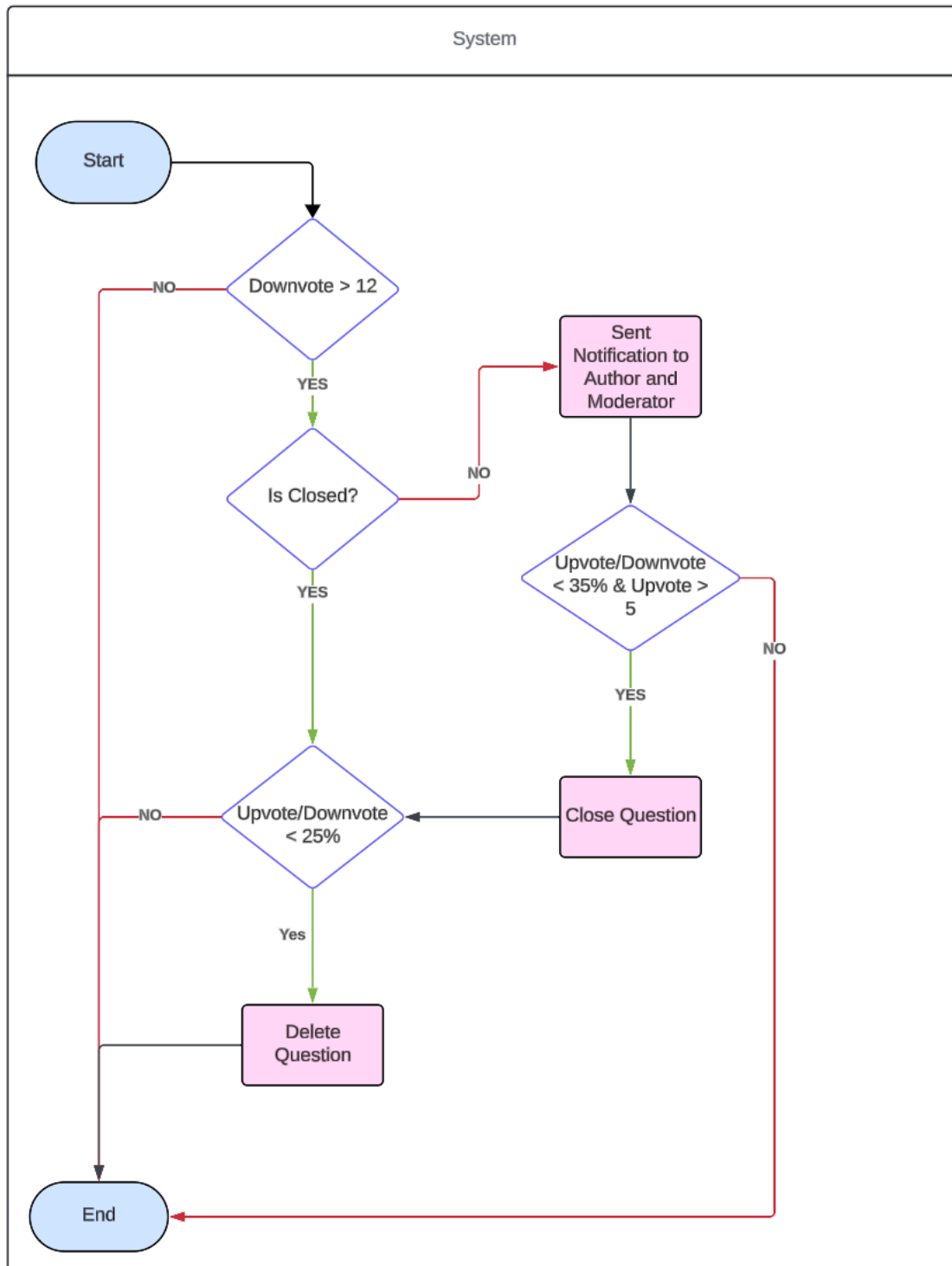


Search Flowchart

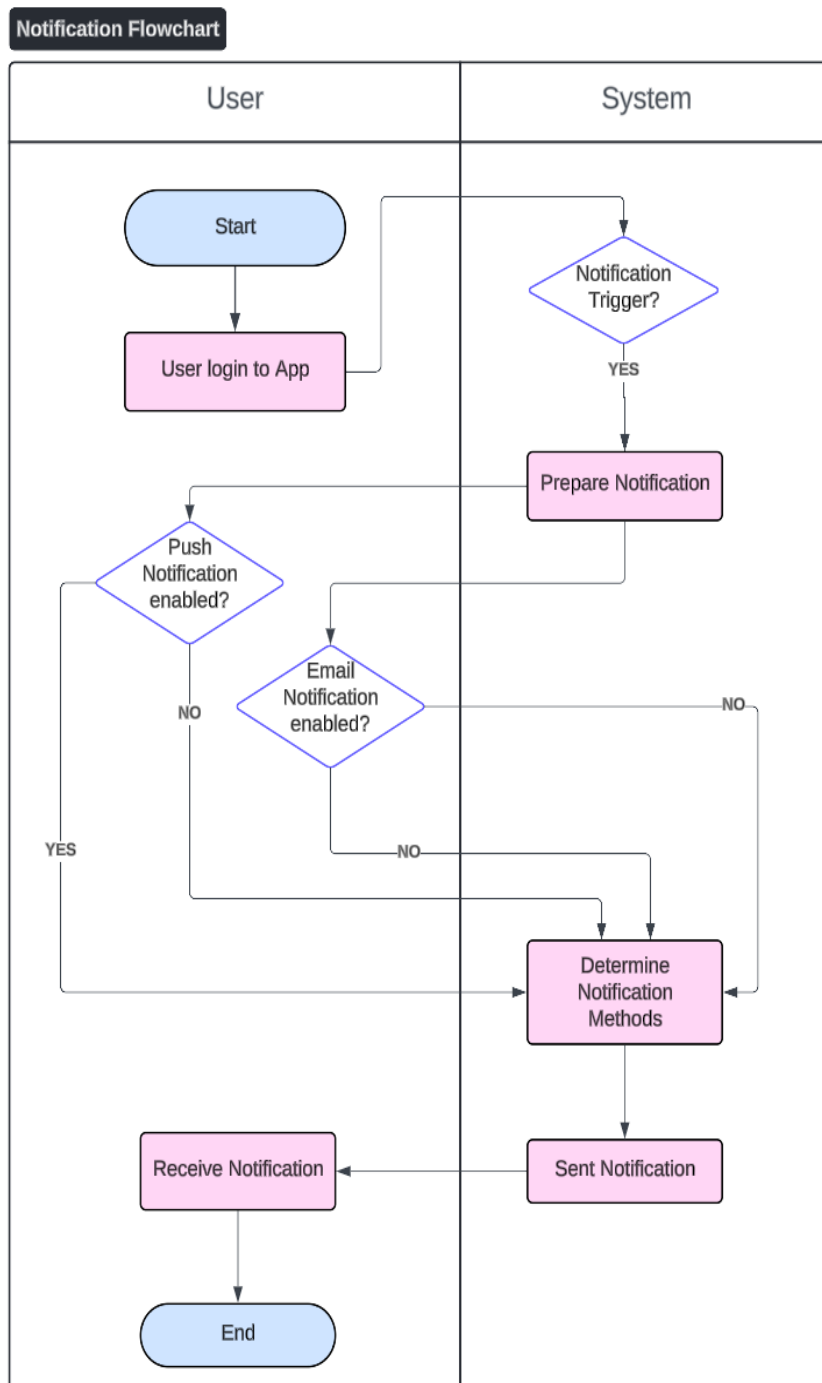


Moderation Flowchart

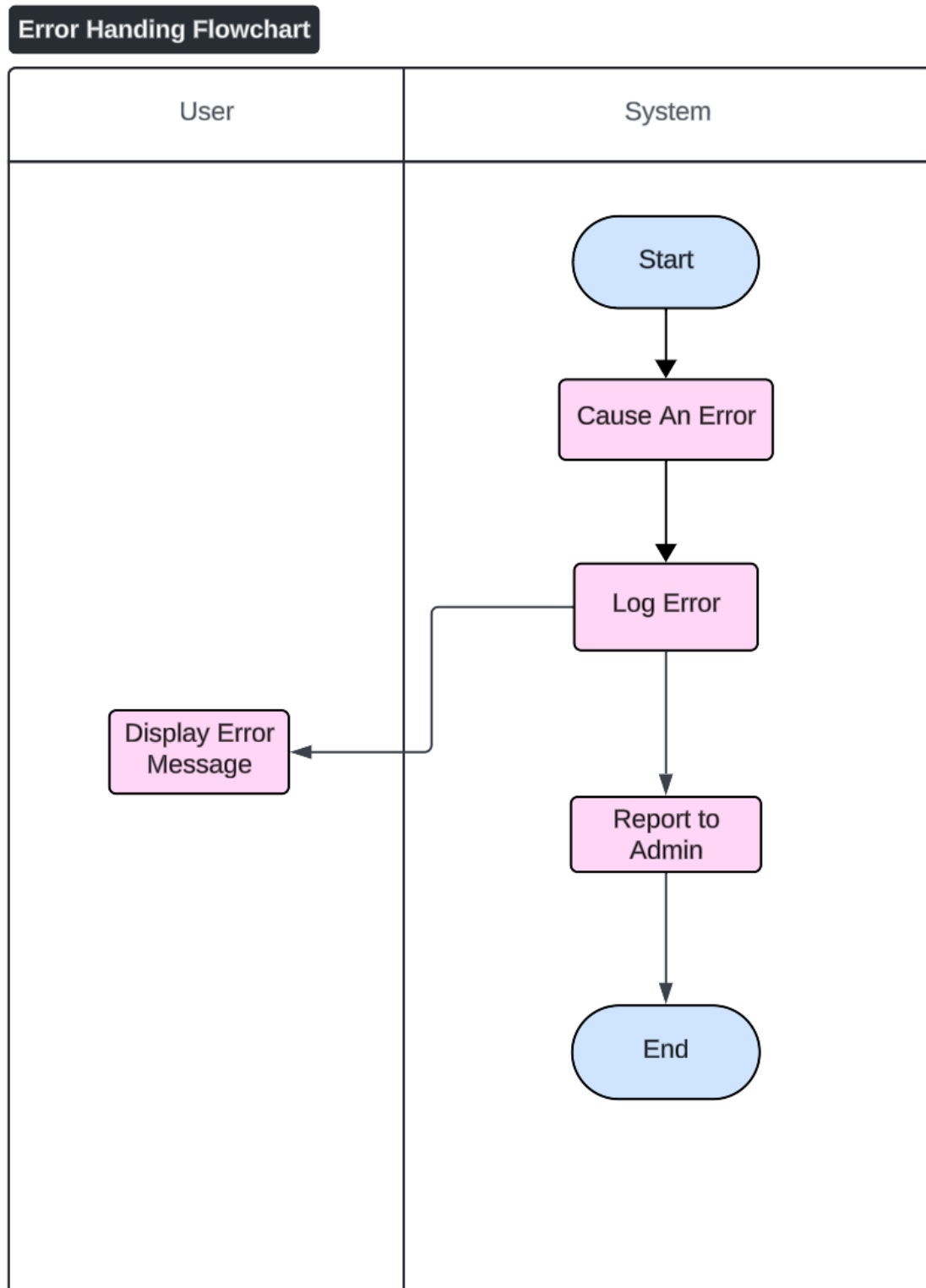
Moderation Flowchart



Notifications Flowchart

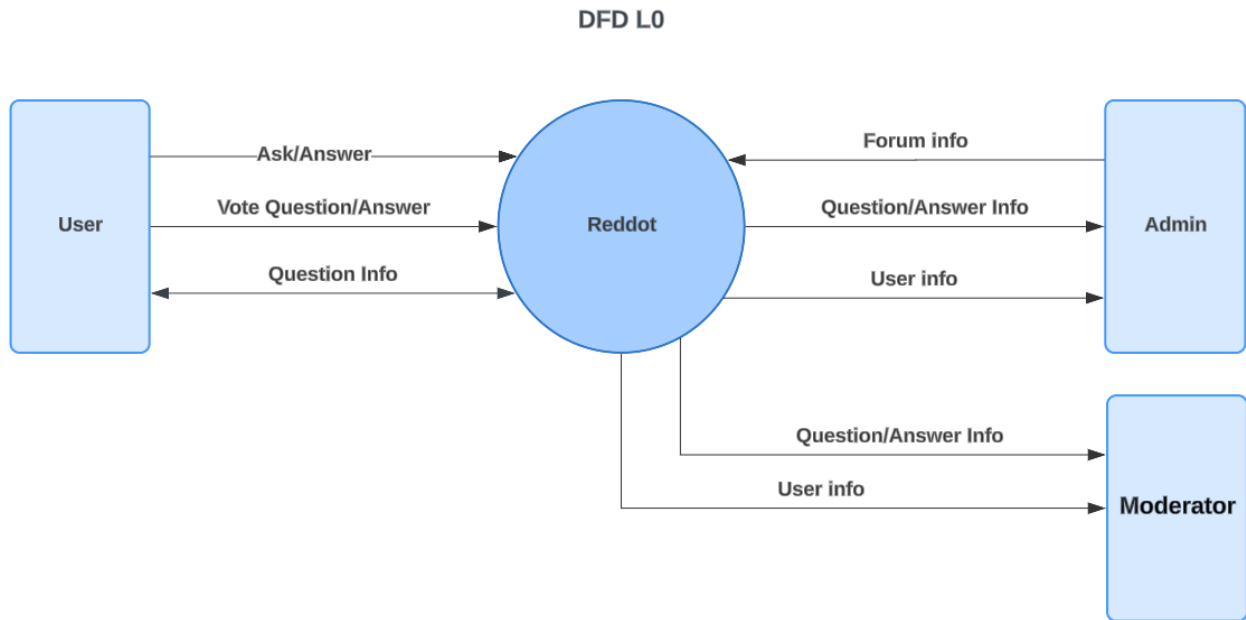


Error Handling Flowchart



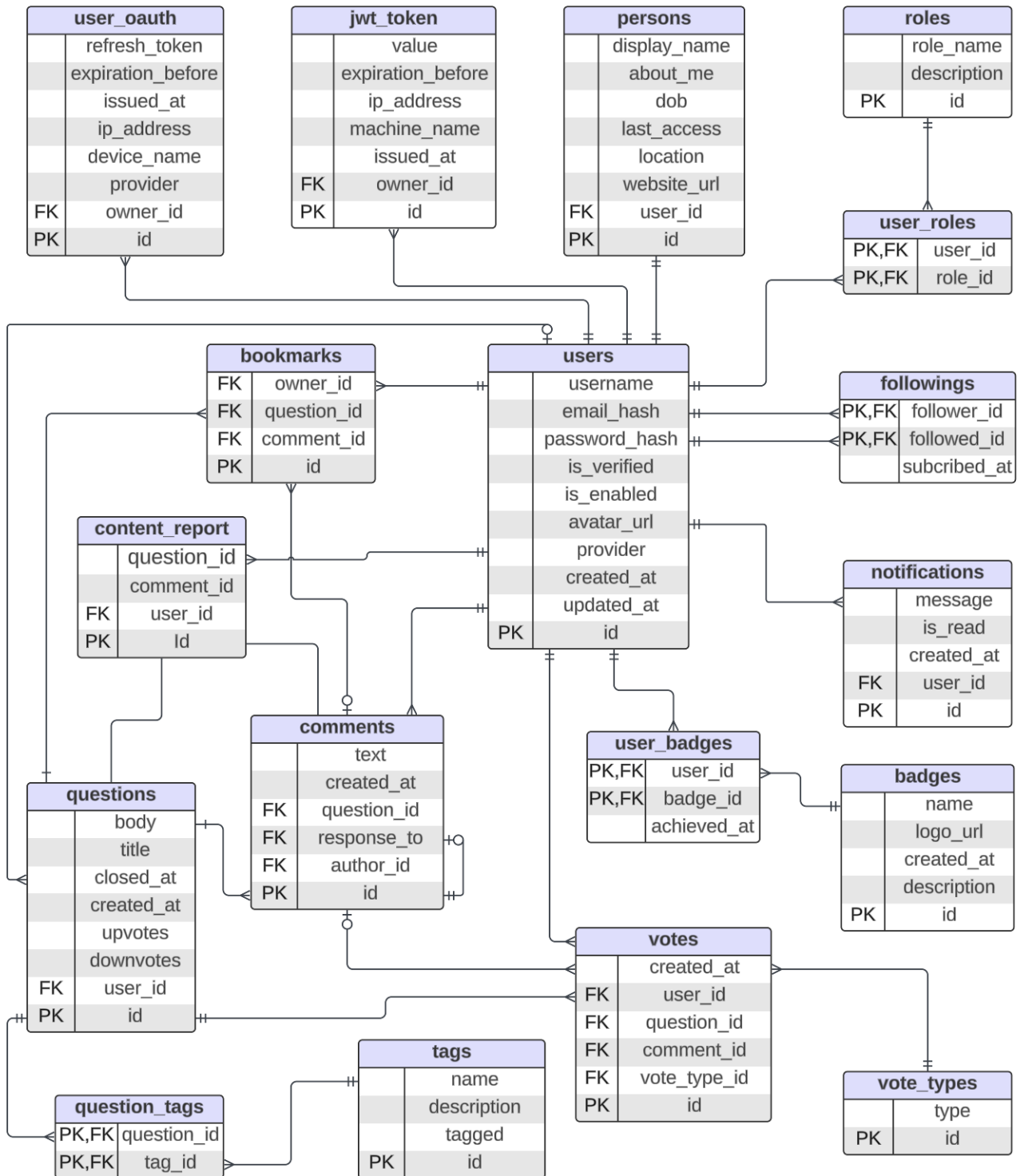
Data Flow Diagram

Data Flow Diagram Level 0



Database Structure

Entity Relationship Diagram



Testing Instruction

GitHub Repository

<https://github.com/TAT2k30/Apolo11.git>