(CSE231) Advanced Computer Programming

Junior CESS

Spring 2024

Major Task

Name: Ahmed Mohamed El Henawy

ID: 21P0298

Social Media Platform

1. Introduction:

Today, social media is a big part of our lives. It changes how we connect, share, and interact with others. This project, called "Social Media Platform," aims to create a simple and user-friendly social media app using Java. I will focus on Object-Oriented Programming (OOP) and building a nice graphical user interface (GUI).

My main goal is to develop a platform where users can create profiles, post updates, interact with posts, and communicate with each other. The project is divided into two main parts. The first part focuses on using OOP principles to build the core features, such as user profiles, posts, and interactions. The second part will add a GUI using JavaFX to make the platform easy to use and visually appealing.

This project is a great way to apply what I have learned in class to a real-world scenario. It helps me gain practical experience in software design, coding, testing, and documentation. Additionally, it improves my project management and communication skills, which are essential in the software development field.

In short, the "Social Media Platform" project is an exciting opportunity to turn my knowledge into a functional and innovative software solution. It allows me to create something useful while showcasing my technical skills and creativity.

2. Project Description:

➢ Milestone 1: OOP Foundations

Objective:

Implement the core structure of the social media platform using OOP principles.

Key Features:

1. **User Profiles:**
   - Attributes: Username, bio, profile picture, and friend list.
   - Functionality: Create a User class that encapsulates user-related functionalities such as managing user information, friend requests, and profile updates.
2. **Post and Interaction Representation:**
   - Classes: Design classes for representing posts, comments, and likes.
   - Methods: Implement methods for creating and interacting with posts, including liking and commenting on posts.

Deliverables:

- UML diagrams illustrating class relationships and hierarchies.
- Core classes for user profiles, posts, and interactions.
- Basic functionalities for user registration, login, post creation, and interactions.

➢ Milestone 2: GUI Development

Objective:

Enhance the application by incorporating a graphical user interface (GUI) using JavaFX.

Key Features:

3. **Graphical User Interface (GUI):**
   - **Design:** Create an interactive and user-friendly GUI using JavaFX.
   - **Screens:** Design screens for user registration/login, user profiles, news feed, and post creation.

Deliverables:

- Fully functional GUI with screens for user registration/login, user profiles, news feed, and post creation.

➢ Technologies and Tools Used

1. **Java:**
   - **Core Language:** Java is used for implementing the backend logic and core functionalities of the social media platform.
   - **OOP Principles:** Emphasis on OOP principles to ensure modular, maintainable, and reusable code.
2. **JavaFX:**
   - **GUI Development:** JavaFX is used for creating the graphical user interface, ensuring a user-friendly and interactive design.
   - **Screens:** Design and implementation of various screens such as user registration, login, profile management, and news feed.

3. Functionalities:

➢ User Registration and Login:

- Users can register by providing their username, email, password, bio, and profile picture.
- Login functionality allows users to access their profiles and interact with the platform.

➢ User Profiles:

- Each user has a profile displaying their information, posts, and friend list.
- Users can update their profile information and manage their friend requests.

➢ Posts and Interactions:

- Users can create posts with text content.
- Other users can like and comment on posts, fostering interaction and engagement.

➢ News Feed:

- A dynamic news feed displays posts from friends and other users.
- Users can scroll through the feed to see the latest posts and interactions.

➢ Friend Management:

- Users can send and receive friend requests.
- Friend lists are managed efficiently, allowing users to add or remove friends.

4. Beneficiaries:

➢ Users:

- **General Public:** Individuals looking to connect with friends, share updates, and interact online.
- **Content Creators:** Users who want to create and share content with their audience.

➢ Developers:

- **Students:** Those learning Java and OOP principles through practical experience.
- **Educators:** Professors using the project to teach programming concepts.

5. Analysis:

Functional and Non-Functional Requirements:

➢ Functional Requirements

1. **User Registration and Login:**
   - The system shall allow users to register with a unique username, email, password, bio, and profile picture.
   - The system shall authenticate users based on their email and password.
2. **Profile Management:**
   - Users shall be able to update their profile information, including username, bio, and profile picture.
   - Users shall be able to view their own profiles as well as other users' profiles.
3. **Post Creation and Interaction:**
   - Users shall be able to create new posts with text content.
   - Users shall be able to like posts.
   - Users shall be able to comment on posts.
4. **Friend Management:**
   - Users shall be able to send friend requests to other users.
   - Users shall be able to accept or decline friend requests.
   - Users shall be able to view a list of their friends.
5. **News Feed:**
   - The system shall display a news feed with posts from the user's friends.
   - The news feed shall be updated with new posts, likes, and comments.
6. **Logout:**
   - Users shall be able to log out of their accounts.

➢ Non-Functional Requirements:

1. **Usability:**
   - The user interface shall be intuitive and easy to navigate.
   - The system shall provide feedback for user actions.
2. **Performance:**
   - The system shall respond to user actions within 2 seconds under normal load conditions.
   - The platform shall be able to handle 100 simultaneous users without significant performance degradation.
3. **Security:**
   - User passwords shall be stored securely using encryption.
   - The system shall validate user inputs.

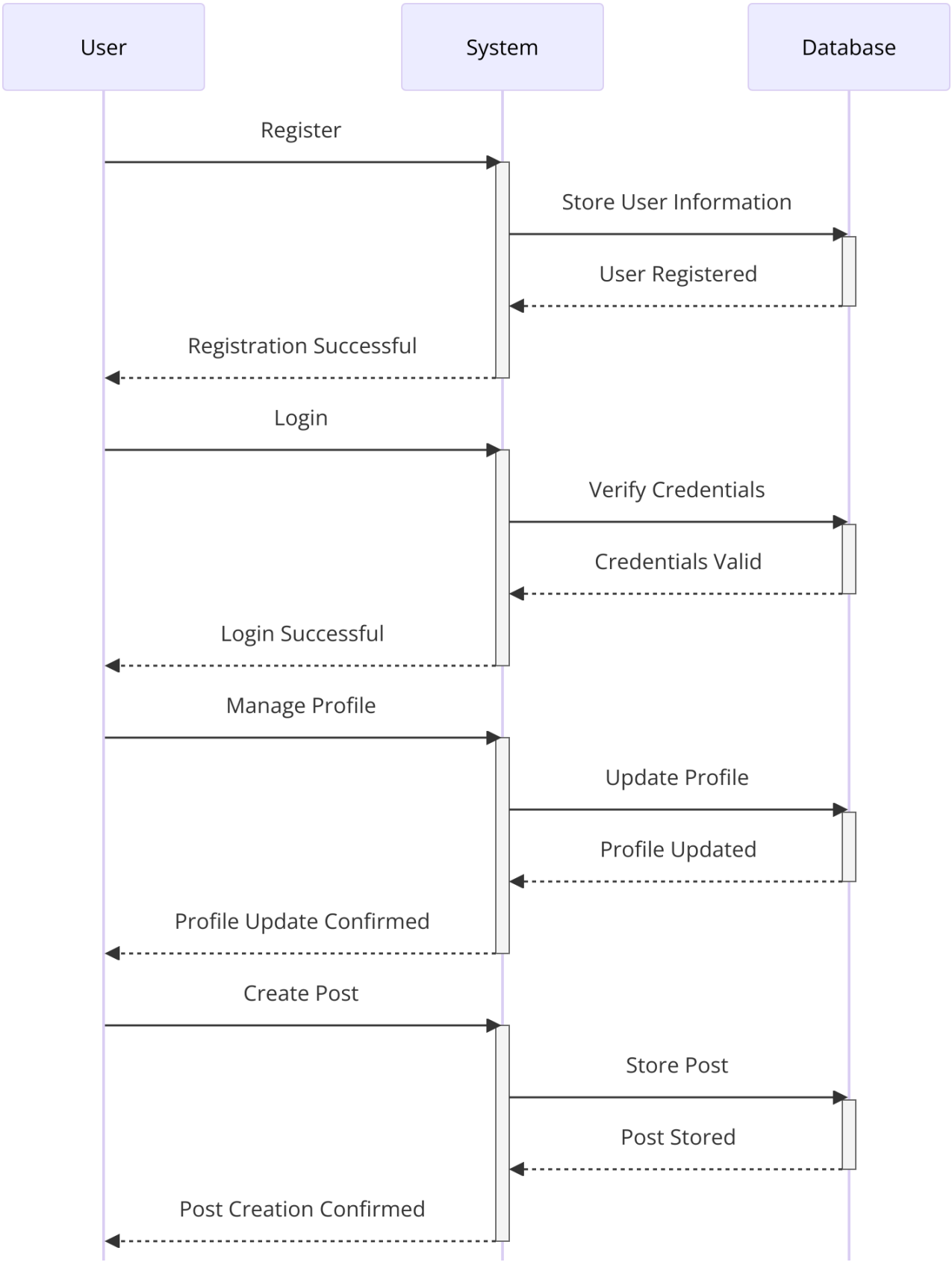
4. Maintainability:

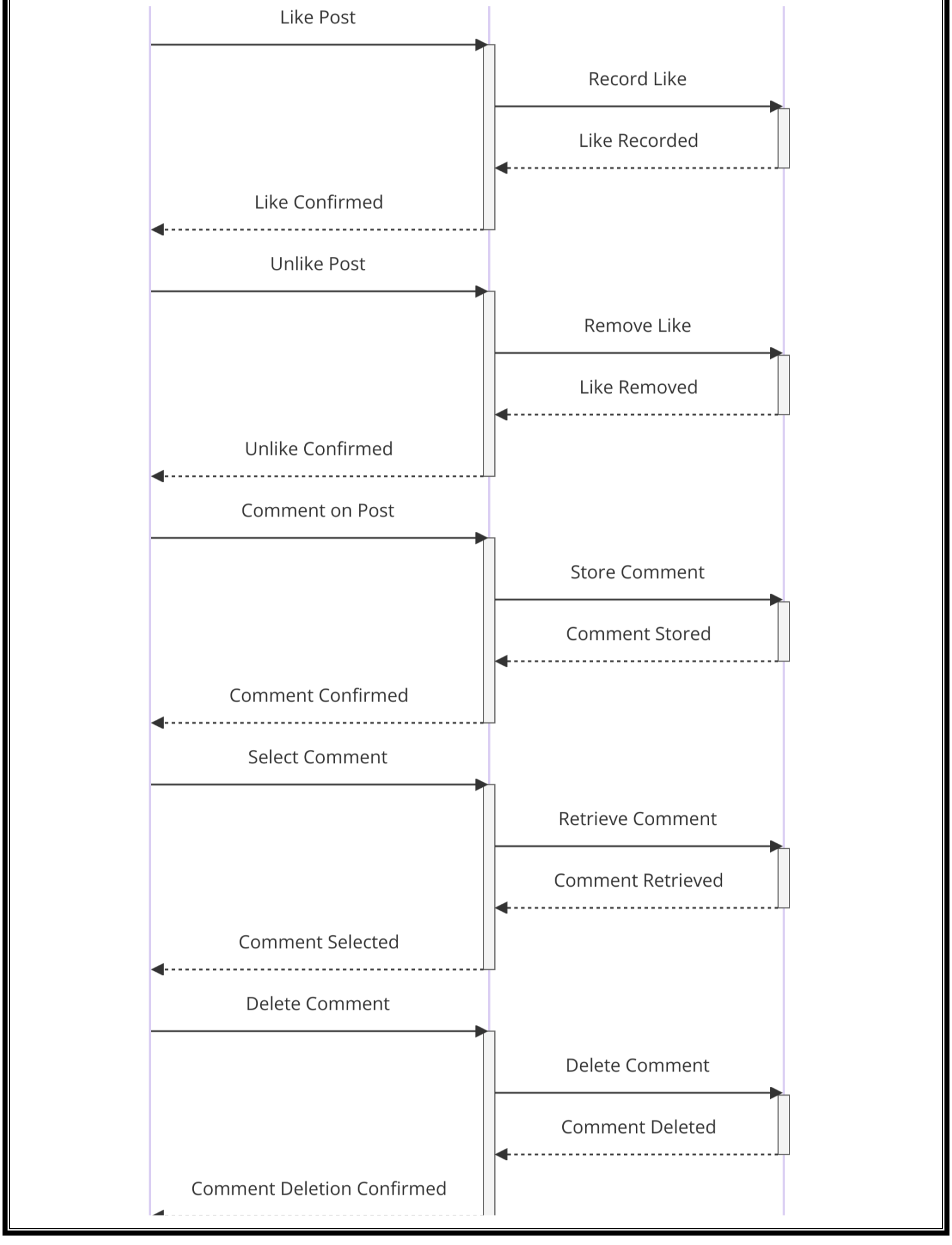
- The codebase shall be modular to allow for easy updates and maintenance.
- The system shall be documented to facilitate future development and troubleshooting.

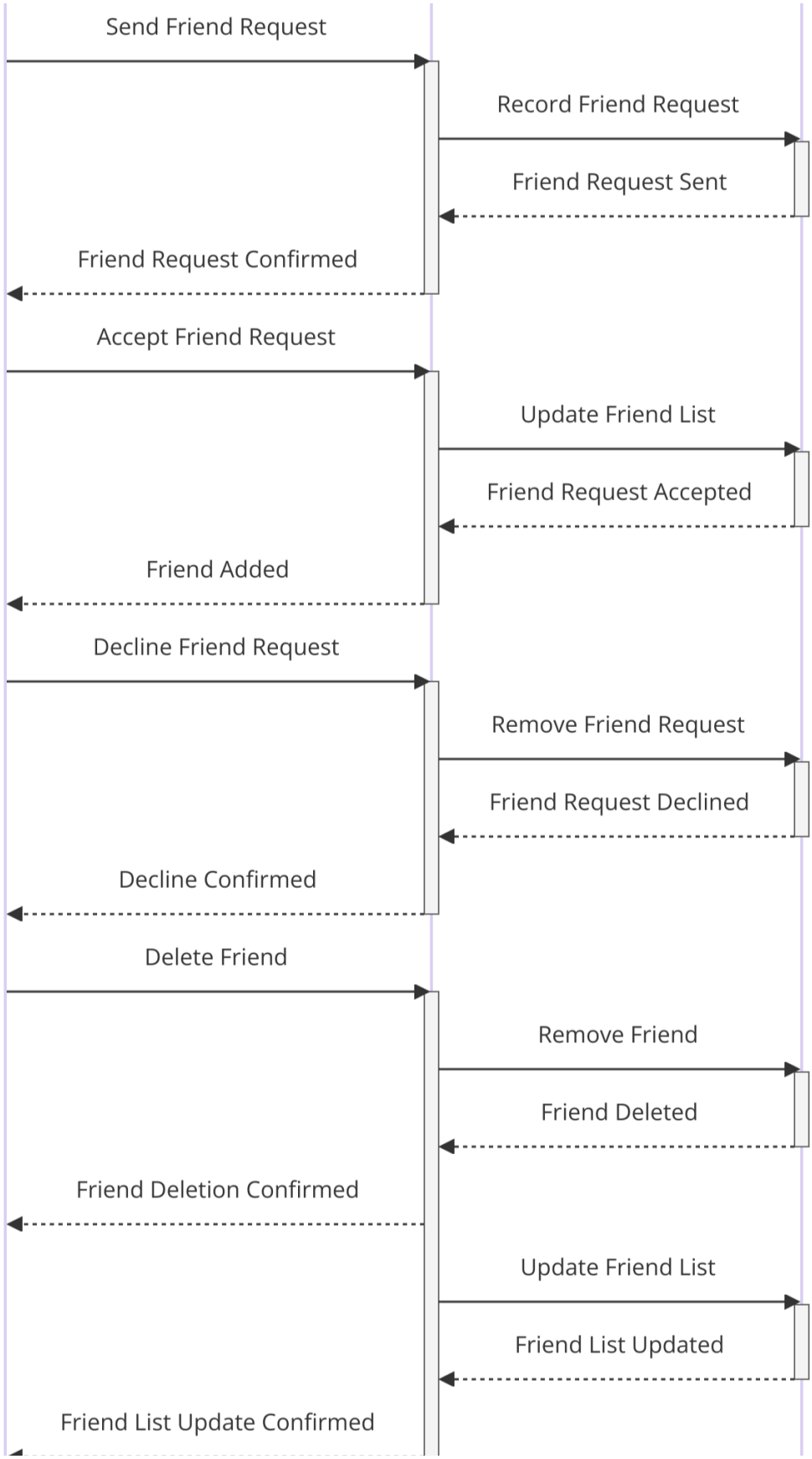
5. Scalability:

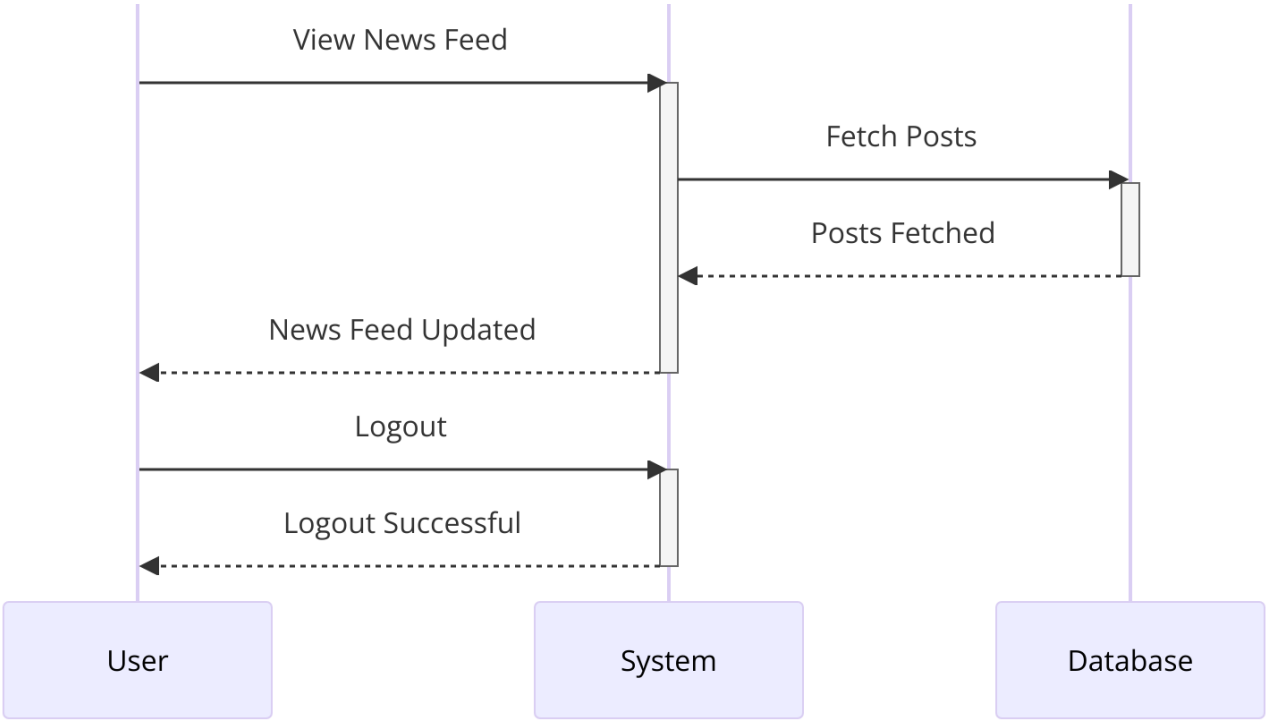
- The system architecture shall allow for easy scaling to accommodate more users as needed.
- The database design shall support efficient storage and retrieval of user data and posts.

Sequence Diagram:

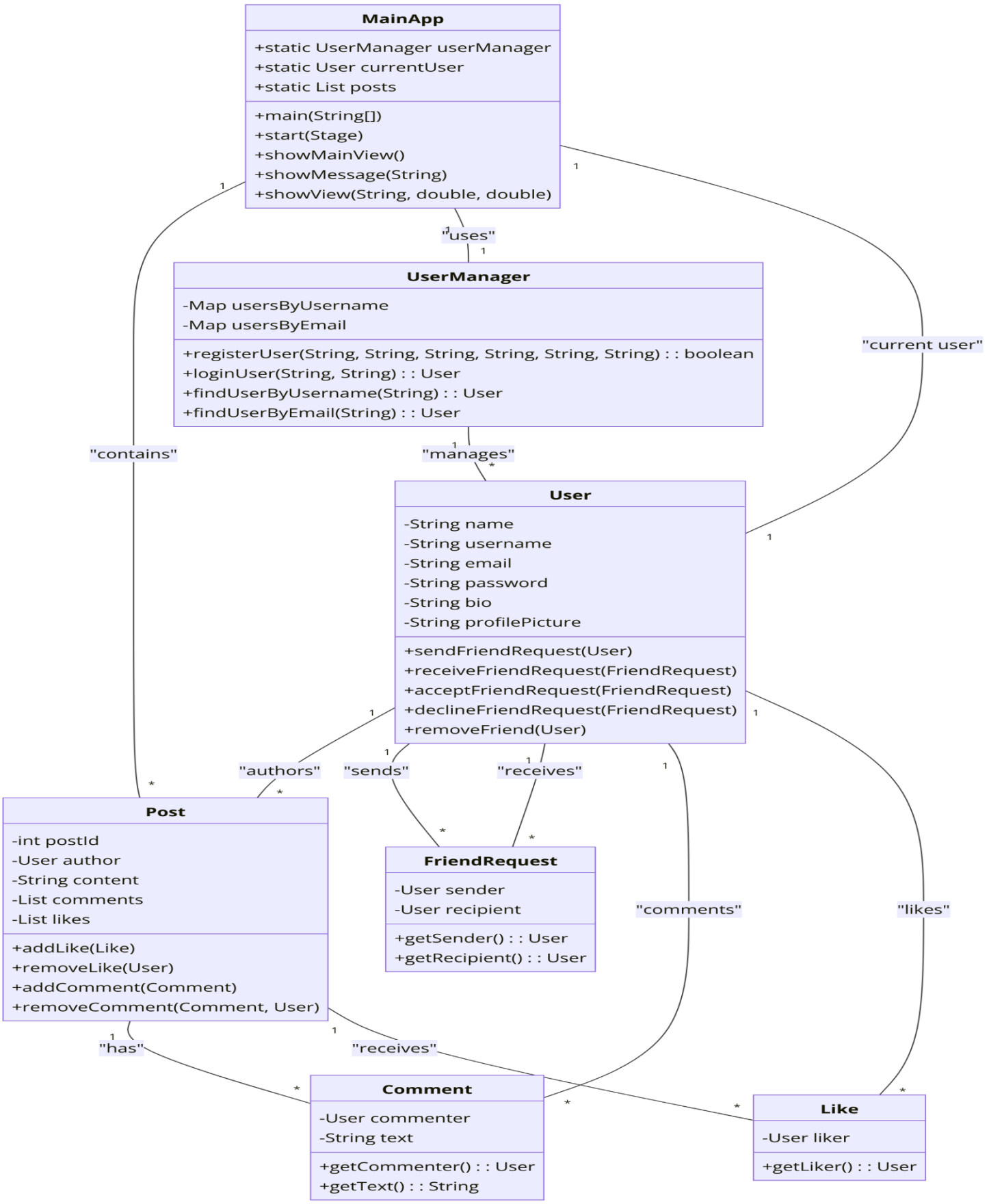



Like Post
Record Like
Like Recorded
Like Confirmed
Unlike Post
Remove Like
Like Removed
Unlike Confirmed
Comment on Post
Store Comment
Comment Stored
Comment Confirmed
Select Comment
Retrieve Comment
Comment Retrieved
Comment Selected
Delete Comment
Delete Comment
Comment Deleted
Comment Deletion Confirmed


Send Friend Request
Record Friend Request
Friend Request Sent
Friend Request Confirmed
Accept Friend Request
Update Friend List
Friend Request Accepted
Friend Added
Decline Friend Request
Remove Friend Request
Friend Request Declined
Decline Confirmed
Delete Friend
Remove Friend
Friend Deleted
Friend Deletion Confirmed
Update Friend List
Friend List Updated
Friend List Update Confirmed


View News Feed
Fetch Posts
Posts Fetched
News Feed Updated
Logout
Logout Successful
User
System
Database

Class Diagram:



6.System Architecture:

➢ Directory Structure:

AdvancedProgrammingProject/

- src
  - SocialMediaPlatform
    - Comment.java
    - CreatePostController.java
    - FriendRequest.java
    - Like.java
    - LoginController.java
    - MainApp.java
    - MainController.java
    - ManageFriendsController.java
    - NewsFeedController.java
    - Post.java
    - RegisterController.java
    - User.java
    - UserDashboardController.java
    - UserManager.java
  - resources
    - SocialMediaPlatform
      - MainView.fxml
      - UserDashboardView.fxml
      - ManageFriendsView.fxml
      - LoginView.fxml
      - RegisterView.fxml
      - CreatePostView.fxml
      - NewsFeedView.fxml
- users.txt

➢ Project Classification:

- Models
- Controllers
- Views

1. Models:

The Model classes represent the core data structures and business logic of the application. These classes define the properties and behaviors of the application's data.

Classes:

- **User.java:** Represents a user in the social media platform.
- **Post.java:** Represents a post created by a user.
- **Comment.java:** Represents a comment made on a post.

- **Like.java:** Represents a like on a post.
- **FriendRequest.java:** Represents a friend request sent between users.
- **UserManager.java:** Manages user registration, login, and data retrieval.

2. Controllers:

The Controller classes handle the user input and interactions, mediating between the views and models. They contain the logic to respond to user actions.

Classes:

- **MainApp.java:** The main application class that starts the JavaFX application and manages navigation between different views.
- **MainController.java:** Controls the main view.
- **CreatePostController.java:** Controls the post creation view.
- **LoginController.java:** Controls the login view.
- **RegisterController.java:** Controls the registration view.
- **UserDashboardController.java:** Controls the user dashboard view.
- **ManageFriendsController.java:** Controls the manage friends view.
- **NewsFeedController.java:** Controls the news feed view.

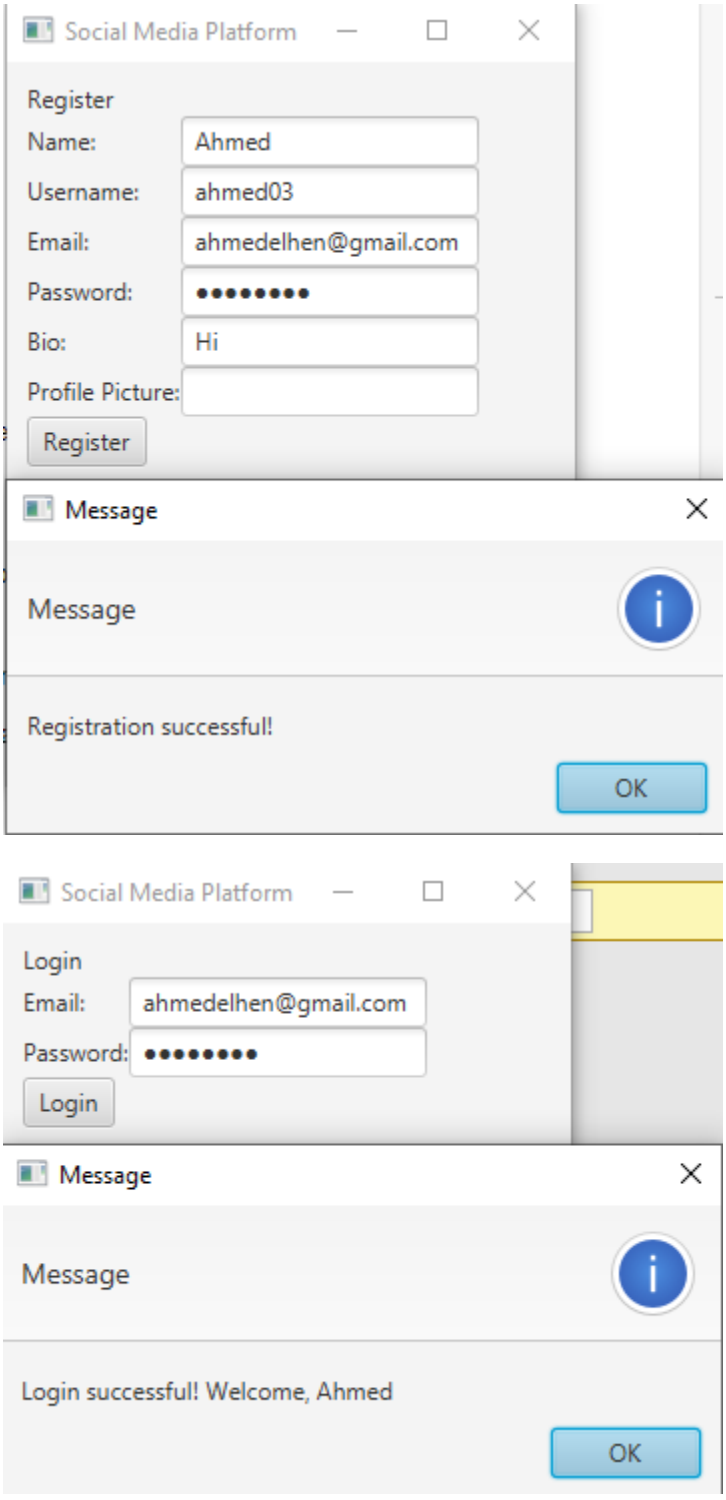
3. Views:

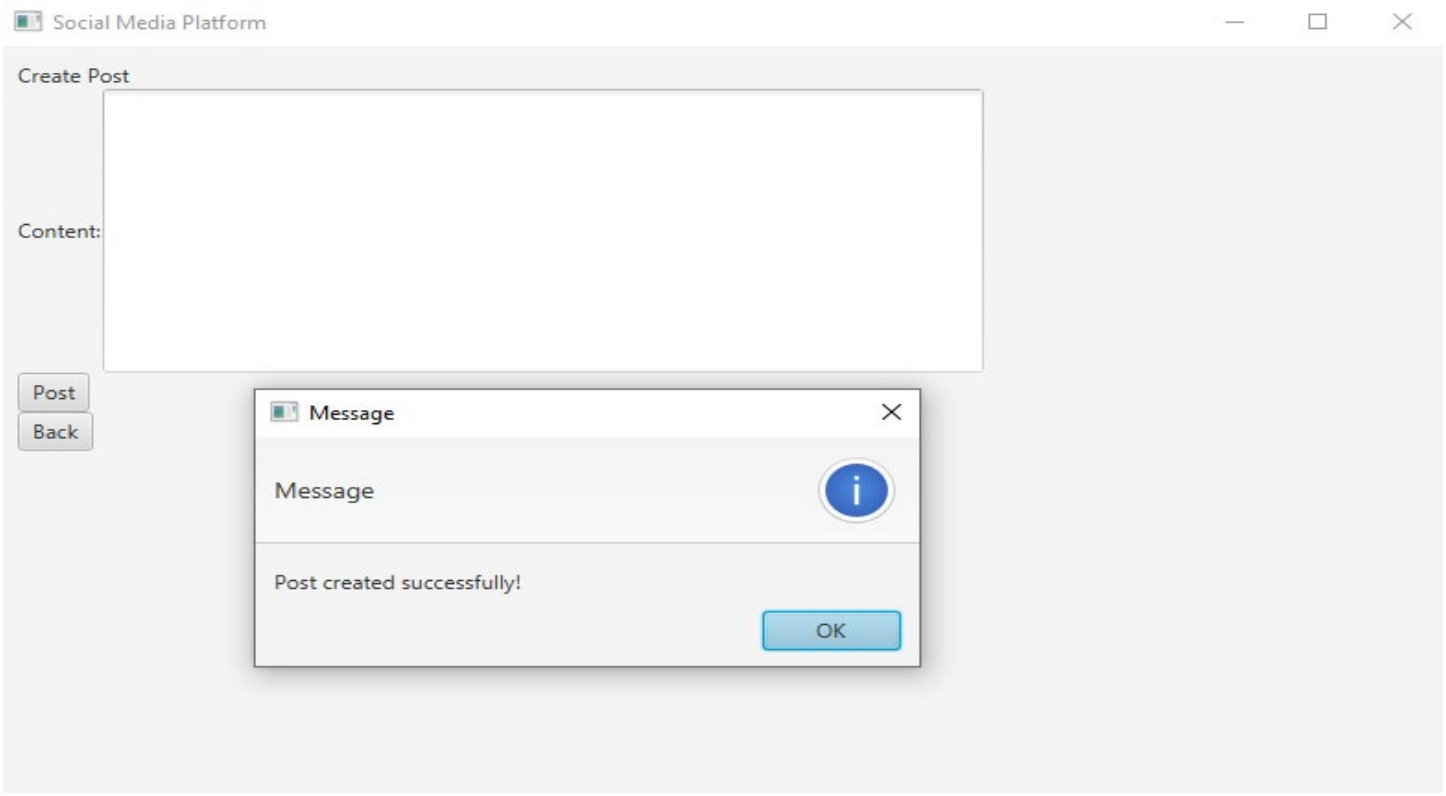
The View components are the FXML files that define the user interface layout. They are loaded by the controllers to display the different screens of the application.

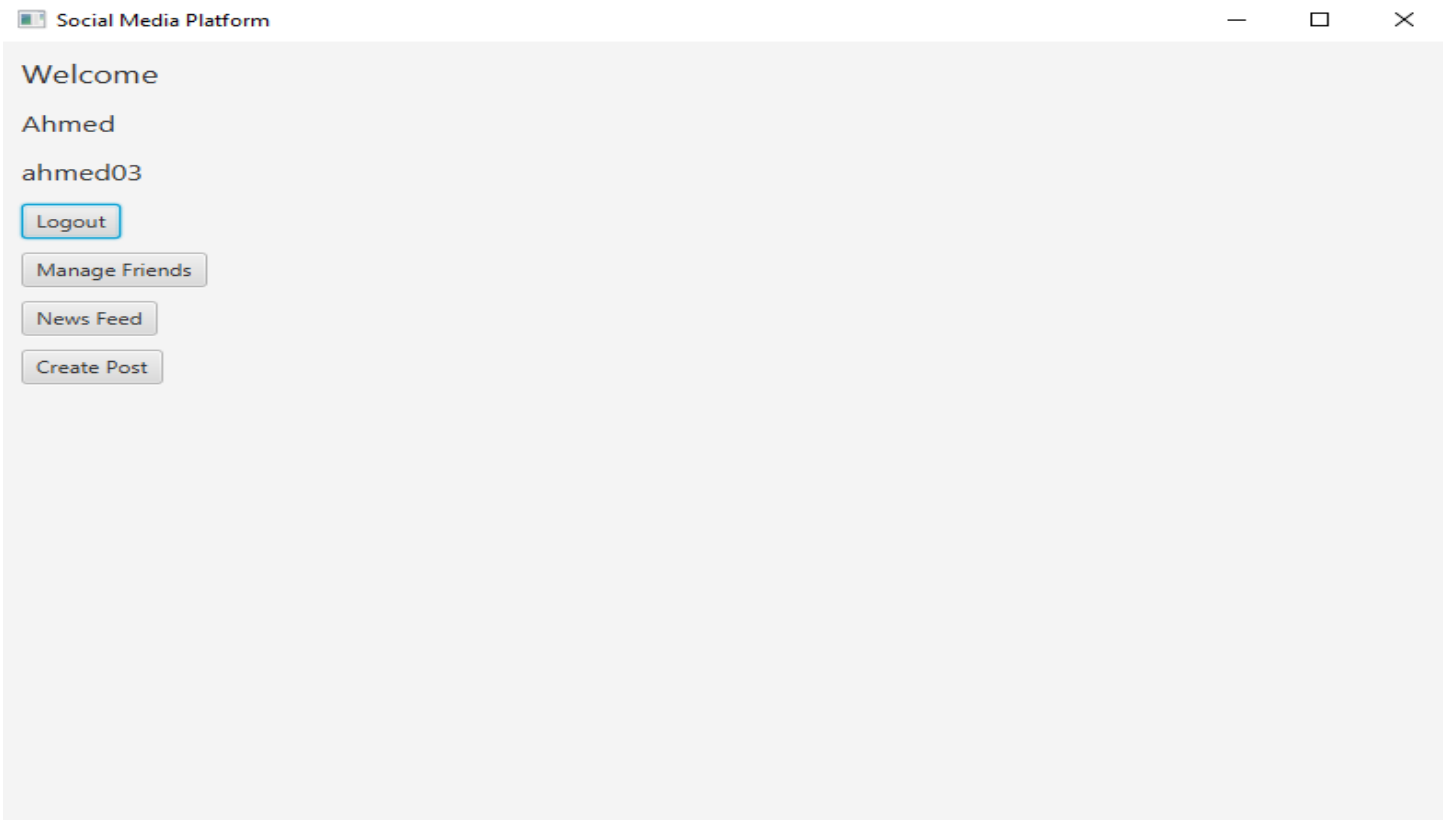
Files:

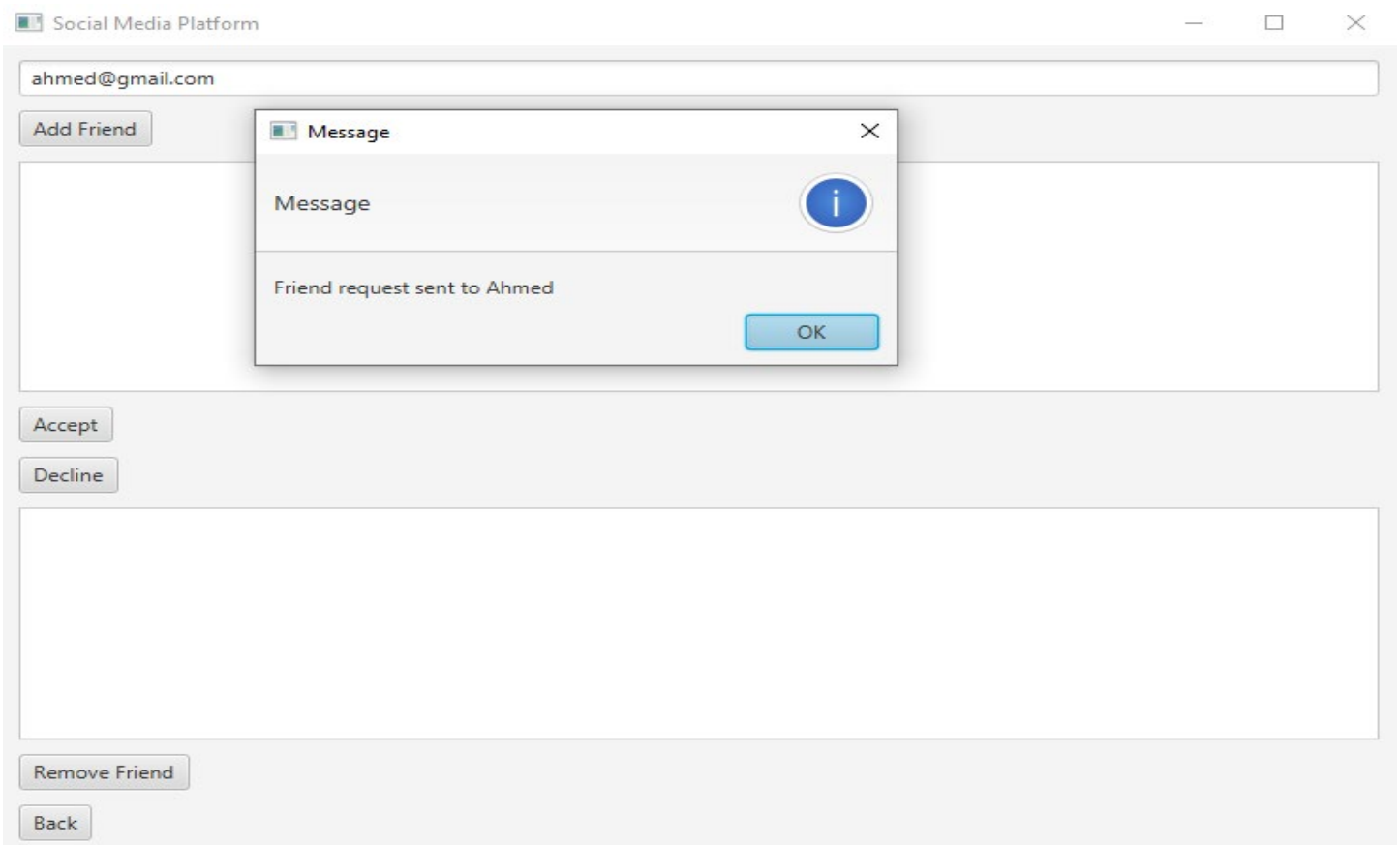
- **MainView.fxml:** The main screen of the application where users can navigate to login or register.
- **UserDashboardView.fxml:** The user dashboard displaying user information and navigation options.
- **ManageFriendsView.fxml:** The screen to manage friends and friend requests.
- **LoginView.fxml:** The login screen for user authentication.
- **RegisterView.fxml:** The registration screen for creating a new user account.
- **CreatePostView.fxml:** The screen for creating new posts.
- **NewsFeedView.fxml:** The news feed displaying posts from users.

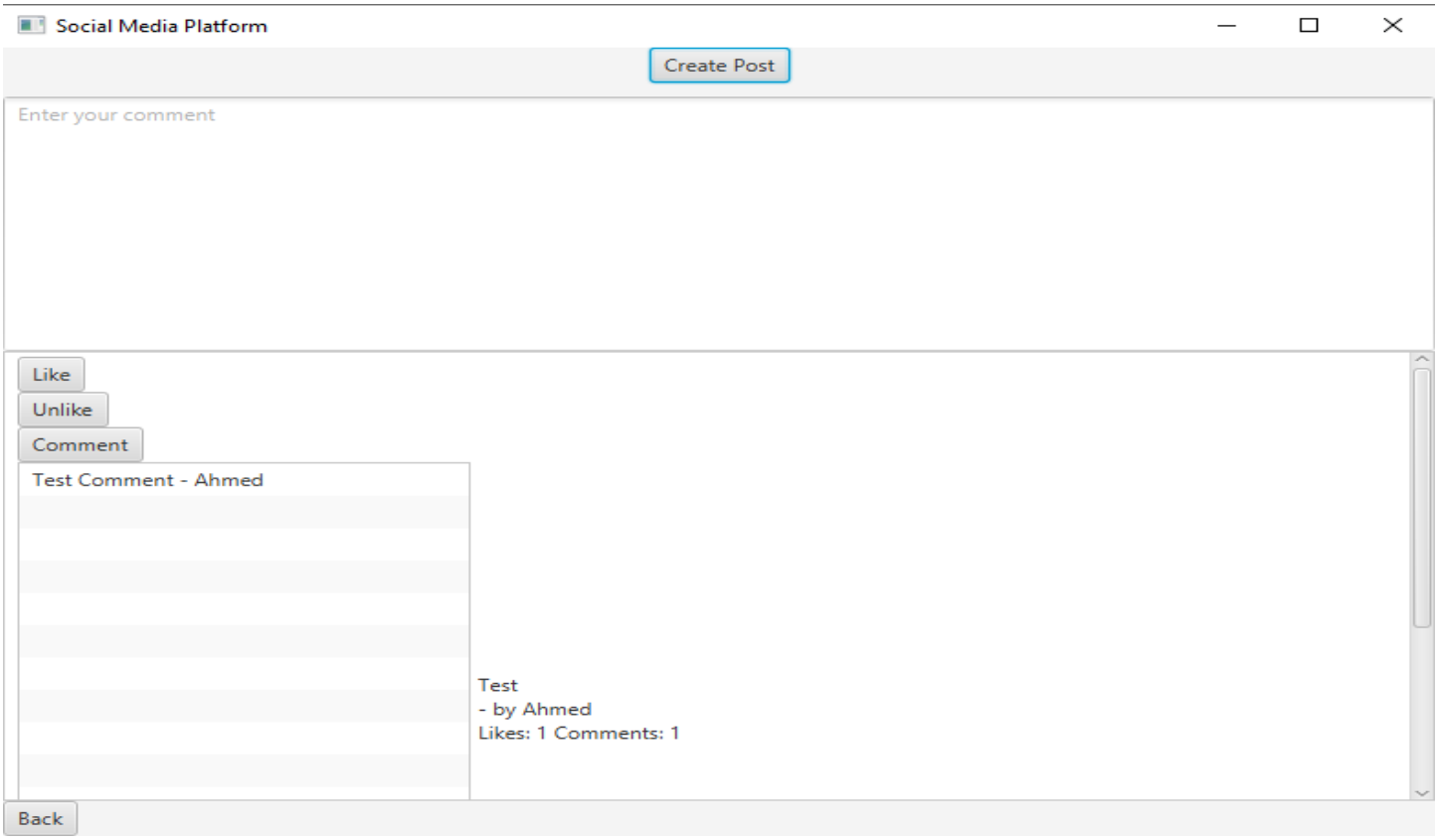
7. Testing:


Social Media Platform
Register
Name:
Ahmed
Username:
ahmed03
Email:
ahmedelhen@gmail.com
Password:
Bio:
Hi
Profile Picture:
Register
Message
Message
Registration successful!
OK
Social Media Platform
Login
Email:
ahmedelhen@gmail.com
Password:
Login
Message
Message
Login successful! Welcome, Ahmed
OK


Social Media Platform
Create Post
Content:
Post
Back
Message
Message
Post created successfully!
OK


Social Media Platform
Welcome
Ahmed
ahmed03
Logout
Manage Friends
News Feed
Create Post


Social Media Platform
ahmed@gmail.com
Add Friend
Message
Message
Friend request sent to Ahmed
OK
Accept
Decline
Remove Friend
Back


Social Media Platform
Create Post
Enter your comment
Like
Unlike
Comment
Test Comment - Ahmed
Test
- by Ahmed
Likes: 1 Comments: 1
Back

8. End-User Guide:

➢ 1. Introduction

Welcome to the Social Media Platform! This guide will help you navigate and use the platform's features effectively. Whether you are a new user or need assistance with specific functionalities, this guide has you covered.

➢ 2. System Requirements

Ensure your system meets the following requirements to run the application:

- Operating System: Windows 7/8/10, macOS, or Linux
- Java Runtime Environment (JRE) 8 or later
- Internet connection (optional for certain features)

➢ 3. Launching the Application

- Double-click the application icon to launch the platform.
- You will be greeted with the main screen where you can choose to register or log in.

➢ 4. User Registration and Login

Registering a New Account

1. On the main screen, click the "Register" button.
2. Fill in the registration form with your details:
   - Name: Enter your full name.
   - Username: Choose a unique username.
   - Email: Provide a valid email address.
   - Password: Create a strong password.
   - Bio: Write a short bio about yourself (optional).
   - Profile Picture: Upload a profile picture (optional).
3. Click the "Register" button to create your account.
4. A confirmation message will appear if registration is successful.

Logging into Your Account

1. On the main screen, click the "Login" button.
2. Enter your registered email and password.
3. Click the "Login" button.
4. If the credentials are correct, you will be redirected to the user dashboard.

➢ 5. Using the Platform

Navigating the User Dashboard

1. The user dashboard is your main hub for all activities.
2. From here, you can manage your profile, create posts, manage friends, and view the news feed.

Managing Your Profile

1. Click on your profile name or picture on the dashboard.
2. You can update your bio, change your profile picture, and update other personal information.
3. Click "Save" to apply changes.

Creating Posts

1. Click the "Create Post" button on the dashboard.
2. Enter the content of your post in the provided text area.
3. Click the "Post" button to publish your post.
4. Your post will appear on the news feed for your friends to see.

Liking and Commenting on Posts

1. On the news feed, you will see posts from your friends.
2. To like a post, click the "Like" button below the post.
3. To comment, enter your comment in the text area below the post and click "Comment."

Managing Friends

1. Click the "Manage Friends" button on the dashboard.
2. To add a friend, enter their username or email in the "Add Friend" field and click "Add Friend."
3. To accept a friend request, find the request in the list and click "Accept."
4. To decline a request, click "Decline."
5. To remove a friend, select the friend from your list and click "Remove Friend."

Viewing the News Feed

1. Click the "News Feed" button on the dashboard.
2. The news feed displays posts from your friends in chronological order.
3. Scroll through to view, like, and comment on posts.

➢ 6. Logging Out

1. Click the "Logout" button on the user dashboard.
2. You will be redirected to the main screen, confirming you have logged out successfully.

➢ 7. GitHub Repository

- AhmedElhenawy/SocialMediaPlatform: This project is a comprehensive social media platform developed in Java, utilizing Object-Oriented Programming (OOP) principles and JavaFX for the graphical user interface. It includes essential features such as user registration, profile management, post creation, interactions through likes and comments, and friend management. (github.com)

9.Conclusion:

In summary, the Social Media Platform project successfully demonstrates the application of Object-Oriented Programming (OOP) principles and JavaFX for creating a functional and user-friendly social media application. The platform includes essential features such as user registration, profile management, post creation, interactions through likes and comments, and friend management. The development process emphasized modularity, maintainability, and scalability, resulting in a robust codebase and a seamless user experience.

The project not only achieved its core objectives but also provided valuable learning experiences in software design, user interface development, and project management. This foundation sets the stage for future enhancements and innovations, ensuring the platform can adapt and grow to meet user needs effectively. Overall, the Social Media Platform project represents a significant step forward in understanding and implementing advanced programming concepts in a practical, real-world context.