

Swe573 Software Development Practice, Spring 2024 Project Final
Deliverables
Ali Şer Gök
20.05.2024

Project name deployment URL:

Git repository URL: <https://github.com/AliserGok/SWE573-AliserGok.git>

Git tag version URL: <https://github.com/AliserGok/SWE573-AliserGok/releases/tag/v0.9>

Project demonstration video URL:

<https://drive.google.com/file/d/1TvgdNGm0wVFt67LY11EXFkM4tGDAQPSa/view?usp=sharing>

HONOR CODE

Related to the submission of all the project deliverables for the Swe573 2024 Spring semester project reported in this report, I declare that:

- I am a student in the Software Engineering MS program at Bogazici University and am registered for Swe573 course during the Spring 2024 semester.
- All the material that I am submitting related to my project (including but not limited to the project repository, the final project report, and supplementary documents) have been exclusively prepared by myself.
- I have prepared this material individually without the assistance of anyone else with the exception of permitted peer assistance which I have explicitly disclosed in this report.

Ali Şer Gök

Table of Contents

1. Overview	3
1.1 Introduction.....	3
1.2 Key Features.....	3
1.3 Technologies Used.....	3
1.4 Project Structure.....	3
2. Installation Steps.....	3
2.1 Clone the Repository.....	3
2.2 Start the MySQL Database.....	3
2.3 Connect Database to Java Spring Boot Project.....	3
2.4 Access the Application in Browser.....	3
3. User Manual - How to Use System.....	4
3.1 Registration Process.....	4
3.2 Login Procedures.....	4
3.3 Community Creation Procedures.....	4
3.4 Template Creation Procedures.....	4
3.5 Post Creation Procedures.....	4
3.6 User Profile Creation Process.....	4
3.7 Update Community Process.....	4
3.8 Update User Profile Process.....	5
3.9 View Created Community Process.....	5
3.10 View All Communities Process.....	5
3.11 View All User Profiles Process.....	5
3.12 Search Specific Community Process.....	5
3.13 Search Specific User Process.....	5
3.14 Advanced Search for Posts Within a Community.....	5
4. Software Requirements Specification.....	5
4.1 Community Creation and Moderation.....	5
4.2 User Relationships and Interactions.....	7
4.3 User Profile and Transactions.....	8
4.4 Content Creation and Processes.....	8
4.5 Search and Filtering Procedures.....	9
5. Design Documents.....	10
5.1 Mockups.....	10
5.2 Use Case Diagrams.....	17
5.3 Class Diagrams.....	22
5.4 Sequence Diagrams.....	24

1. Overview:

1.1 Introduction:

"Spiny" is a Community Information System developed using Java Spring Boot, Thymeleaf, and MySQL technologies. It serves as a comprehensive platform for community engagement and information dissemination.

1.2 Key Features:

- User Authentication: Secure login and registration system for users.
- Community Interaction: Facilitates communication and collaboration among community members.
- Content Management: Allows users to create, share, and manage various types of content such as posts and events.
- Personalization: Customizable user profiles and preferences.

1.3 Technologies Used:

- Java Spring Boot: Provides a robust framework for building web applications.
- Thymeleaf: A modern server-side Java template engine for rendering dynamic content.
- MySQL: A relational database management system for storing application data.

1.4 Project Structure:

"Spiny" follows a Model-View-Controller (MVC) architecture:

- Model: Represents the data structure and business logic of the application.
- View: Contains Thymeleaf templates for rendering HTML views.
- Controller: Handles user requests, processes data, and interacts with the database.

2. Installation Steps:

2.1 Clone the repository:

git clone <<https://github.com/AliserGok/SWE573-AliserGok.git>>

2.2 Start the MySQL database:

To create the "spiny-database" schema and tables in our database, open and run the sql file named "[01-spiny-directory](#)" in the project file.

2.3 Connect Database to Java Spring Boot project:

if you run application local please change:

application.properties:

```
spring.datasource.url=jdbc:mysql://{DATABASE_URL}/spiny-database
spring.datasource.username={USERNAME_NAME}
spring.datasource.password={PASSWORD}
```

2.4 Access the application in browser: <http://localhost:8080/showMyLoginPage>**3. User manual - How to use system****Username to log into the system:**

mary, susan, john, michael, emma

Password:

'fun123' (Bcrypt encryption)

3.1 Registration Process:

1. Go to the Login Page.
2. Click the Register button.
3. Fill in the form with your information.
4. Click the Register button.

3.2 Login Procedures:

1. Go to the Login Page.
2. Enter your username and password.
3. Click the login button.

3.3 Community Creation Procedures:

1. Go to the Home Page.
2. Click on Create Community.
3. Enter the community name and description.
4. Click the Create button.

3.4 Template Creation Procedures:

Within the Community:

1. Click on Create Template.
2. Enter the template name.
3. Select the template fields and enter field names.
4. Click the Create button.

3.5 Post Creation Procedures:

Within the Community:

1. Select a template.
2. Go to the Create Post page.
3. Fill in the template fields.
4. Click the Create Post button.

3.6 User Profile Creation Process:

1. Go to the Home Page.
2. Click the My Profile button.
3. On the new page, click the Create Profile button.
4. Fill in the form with your information.
5. Click the Create button.

3.7 Update Community Process:

1. Within the Community, click on Update Community.
2. Modify the information.
3. Click the Update button.

3.8 Update User Profile Process:

1. Navigate to the user profile.
2. Click on the Update User Profile option.
3. In the form that appears, update the information.
4. Click the Update button

3.9 View Created Community Process:

1. Go to the Home Page.
2. Click on the "View" link next to the listed communities.

3.10 View All Communities Process:

1. Go to the Home Page.
2. Check the "Communities" checkbox in the basic search form.
3. Click the Search button. All communities in the application will be listed.

3.11 View All User Profiles Process:

1. Go to the Home Page.
2. Check the "Users" checkbox in the basic search form.
3. Click the Search button. All users in the application will be listed.

3.12 Search Specific Community Process:

1. Go to the Home Page.
2. Check the "Communities" checkbox in the basic search form.
3. Enter the keyword in the search field.
4. Click the Search button. Communities matching the keyword will be listed.

3.13 Search Specific User Process:

1. Go to the Home Page.
2. Check the "Users" checkbox in the basic search form.
3. Enter the keyword in the search field.
4. Click the Search button. Users matching the keyword will be listed.

3.14 Advanced Search for Posts Within a Community:

1. Go to the Community Page.
2. Use the advanced search form to select the criteria you want to search for.
3. Enter the keyword.
4. Click the Search button.

4. Software Requirements Specification

4.1 Community creation and moderation

R-1. WHEN a user decides to create a group, the system shall allow the creator to specify a group name, provide a description, upload a group photo, and determine the visibility settings for the group. (not completed)

R-2. WHEN a user decides to create a community, the system shall enable the user to create a private or public community. (not completed)

R-3. WHERE the user creates a community, the system shall designate the owner as the default moderator of the community. (completed)

R-4. WHEN the owner decides to delegate moderation, the system shall allow the owner to assign moderator roles. (not completed)

R-5. WHEN the owner decides to transfer ownership, the system shall enable the owner to transfer the ownership of the community to another user. (not completed)

R-6. WHEN an owner decides to leave a community, the owner shall transfer the community to another user. (not completed)

R-7. WHERE the system permits, a community shall have multiple owners. (not completed)

R-8. The system shall display the description of the community on the community page. (completed)

R-9. The system shall display the number of members on the community page. (completed)

R-10. WHEN a community is created, the system shall automatically assign a default post type to the community. (completed)

R-11. WHERE the owner initiates post settings, the system shall allow the owner to decide the type of posts created by members. (not completed)

R-12. WHEN a moderator decides to edit the group information, the system shall allow the creator to modify the created group name, description, group photo, and visibility settings. (completed)

R-13. WHEN an owner decides to edit the group information, the system shall allow the creator to modify the created group name, description, group photo, and visibility settings. (completed)

R-14. WHEN a user decides to create a group, the system shall allow the creator to specify a group name, provide a description, upload a group photo, and determine the visibility settings for the group. (completed)

R-15. WHEN an owner decides to leave a community, the owner shall transfer the community to another user. (not completed)

R-16. WHEN a moderator receives requests to join the community, the system shall enable the moderator to admit or reject the requests. (not completed)

R-17. WHEN a moderator decides to delete a post, the system shall allow the moderator to delete posts. (completed)

R-18. WHEN a moderator decides to invite a user to the group, the system shall allow the moderator to send an invitation to the specified user. (not completed)

R-19. WHEN a moderator decides to remove a member from the group, the system shall allow the moderator to remove the member from the group. (not completed)

R-20. WHEN a moderator decides to view followers of the group, the system shall allow the moderator to see the list of followers and perform searches within this list. (completed)

R-21. WHEN the owner decides to remove moderators, the system shall allow the owner to remove moderators from the group. (not completed)

4.2 User Relationships and Interactions

R-22. Members shall be able to see the names of other members. (completed)

R-23. WHEN a user chooses to engage or subscribe, the system shall enable the user to join any group. (completed)

R-24. WHERE the system allows different roles, users shall have various roles with distinct privileges. (completed)

R-25. WHEN a post is made in communities/groups that the user is a member of, the system shall notify the user by sending a notification. (not completed)

R-26. WHEN a friend of the user makes a post, the system shall notify the user by sending a notification. (not completed)

R-27. WHEN a user decides to follow another user, the system shall enable the user to follow the other user. (not completed)

R-28. WHEN a user decides to view activities of followed users, the system shall display the activities of the other user they are following. (not completed)

R-29. WHEN a user decides to view followers, the system shall allow the user to see the list of followers and perform searches within this list. (completed)

R-30. WHEN a user decides to unfollow a friend who is following them, the system shall allow the user to remove the friend from their followers list. (not completed)

4.3 User Profile and Transactions

R-31. WHEN a user decides to create a profile, the system shall allow the user to provide information, including a nickname, age, city, description, and gender. (completed)

R-32. WHEN a user decides to update their profile, the system shall allow the user to upload a profile photo. (completed)

R-33. WHEN a user decides to edit their profile, the system shall allow the user to change their profile information. (completed)

R-34. WHEN a user decides to provide information about themselves, the system shall enable the user to write about themselves in their profile. (completed)

R-35. WHEN a user decides to log out, the system shall allow the user to exit the system. (completed)

R-36. WHEN a user attempts to create or update their nickname, the system shall enforce uniqueness, ensuring that each user's nickname is unique. (completed)

R-37. WHEN a user decides to create or update their email, the system shall enforce uniqueness, ensuring that each user's email is unique. (completed)

4.4 Content Creation and Processes

R-38. WHEN users are in a community page, the system shall allow them to create posts in the community page. (completed)

R-39. WHEN users are in a community home page, the system shall allow them to create posts in the home page. (not completed)

R-40. WHEN a user decides to share content, the system shall allow the user to choose from existing post templates and share content based on the selected template. (completed)

R-41. WHERE a user decides to filter search results, the system shall provide the option to categorize the results into distinct categories such as users, groups, and posts. (completed)

R-42. The system shall provide a basic post type for all communities. (completed)

R-43. WHEN the user decides to filter content, the system shall allow filtering by keyword, post type, and date. (completed)

R-44. The system shall include a reporting system. (not completed)

R-45. WHEN a user decides to delete his/her comment, the system shall allow the user to delete his/her comment. (completed)

R-46. WHEN a user decides to edit his/her comment, the system shall allow the user to edit his/her comment. (not completed)

R-47. WHEN a user decides to delete his/her post, the system shall allow the user to delete his/her post. (completed)

R-48. WHEN a user decides to edit his/her post, the system shall allow the user to edit his/her post. (not completed)

R-49. WHEN a user decides to like a post, the system shall allow the user to express liking for the post. (completed)

R-50. WHEN a user decides to comment on a post, the system shall enable the user to add comments to the post. (not completed)

4.5 Search and Filtering Procedures

R-51. WHEN a user decides to search for content, the system shall enable the user to search by typing keywords. (completed)

R-52. WHEN a user decides to search for content, the system shall enable the user to use filtering features. (completed)




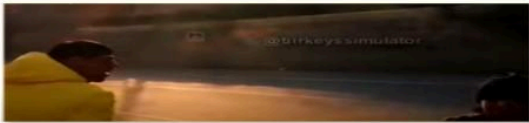
R-53. WHEN a user decides to log out, the system shall allow the user to exit the system. (completed)

R-54. The user shall be able to see recent updates and posts. (completed)


6. Design documents

6.1 Mockups

User Login Page	
User e-mail:	<input type="text" value="alisergok@gmail.com"/>
User Password:	<input type="password" value="*****"/>
<input type="button" value="Login"/>	

Home Page View		
Search	Notifications 14	Logout
Profile		
SamwiseG: 	<div>How logical is it to include orcs in public service? These creatures can overreact.</div> <div></div> <div><input type="button" value="526 Like"/> <input type="button" value="1530 Comment"/></div>	
 hiko21:	<div>https://www.youtube.com/watch?v=ebbiOK16SV8</div> <div></div> <div><input type="button" value="26 Like"/> <input type="button" value="52 Comment"/></div>	

User Own Profile Page View



Edit Profile

525 Friends

Hikmet21


Hello, I am Hikmet. I love traveling and making friends.

40 yo

Diyarbakır

Male

You:




Remove Post

Edit Post

124 Like

62 Comment

You:




Remove Post

Edit Post

11 Like

1 Comment

Create User Profile



Profile Photo

Select Photo

User Name:

Aragorn Gök

Age:

45

City:

Gondor

Sex:


Male

About:

We wrote our love on the walls of Minas Tirith !!!
Arwen Arwen Arwen Arwen Arwen Arwen Arwen

Create

Edit User Profile



Profile Photo
[Select Photo](#)

User Name:

Vito Corleone

Age:

28

City:

Sicily

Gender:


Male

About:

Hi !! I am software engineer and it is my swe573 project

Save

user other people view



525 Friends

Hikmet21

Hello, I am Hikmet. I love traveling and making friends.

40 yo


Diyarbakır

Male

Follow

follow the person to see the content

Group Page (Moderator view)



[Edit Group](#)

14,621 Followers

Followed

Zero black magic for sustainable middle earth
We report those who practice illegal magic to combat black magic and orcs.

Templates

orc news

Black magic notice

Isengard mega project


Minas Tint announcements

[Create Template](#)

SamwiseG:

How logical is it to include orcs in public service? These creatures can overreact.

[Remove Post](#)



526 Like

1530 Comment

Gandalf62:


Due to the drought, everyone needs to cast this rain spell once a day;
Im wen. Telin le thaed. Lasto beth nîn, tolo dan nan galad.

[Remove Post](#)

6084 Like

8550 Comment

Create Group Page



Page Photo
[Select Photo](#)

Group Name:

About:

All posts should only be about Hiko, the rose of Diyarbakır !!!

Community rules;

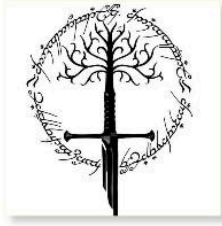
- 1- Comments and content containing swearing and insults will be deleted.
- 2- People who disturb the peace of the community will be expelled from the group.
- 3- Posts that are not related to Hiko Baba will be deleted.

Select post type:

Visibility: ☐ Public ☒ Private

Moderator: [Add Moderator](#)

Moderator Edit Group Page



Page Photo
[Select Photo](#)

Group Name:

About:

Select post type:

Visibility: ☒ Public ☐ Private






Moderators:

Gandalf62
S_Uskudarlı
HikoBaba21

Moderator Invite User Page View

Search User:






Users:






	ali62	<input type="button" value="invite"/>
	Gandalf62	<input type="button" value="invite"/>
	SamwileG	<input type="button" value="invite"/>
	FrodoB	<input type="button" value="invite"/>
	Aragorn Gok	<input type="button" value="invite"/>

Moderator members Page View

Search Member: [invite member](#)

Members:

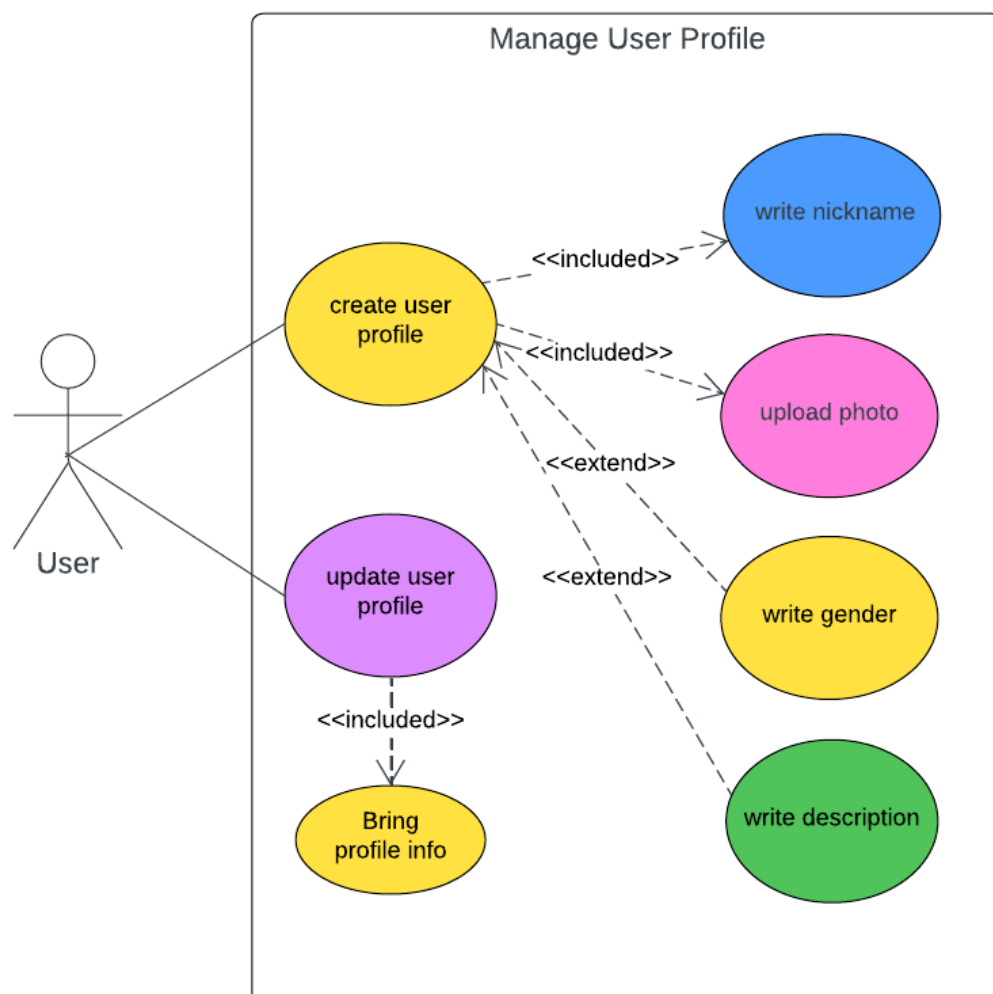
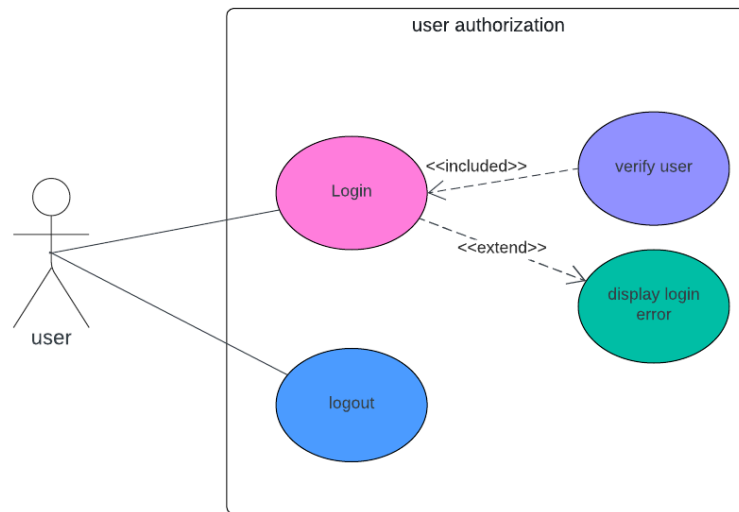
	ali62	<input type="button" value="unsubscribe"/>
	Gandalf62	<input type="button" value="unsubscribe"/>
	SamwileG	<input type="button" value="unsubscribe"/>
	FrodoB	<input type="button" value="unsubscribe"/>
	Aragorn Gok	<input type="button" value="unsubscribe"/>

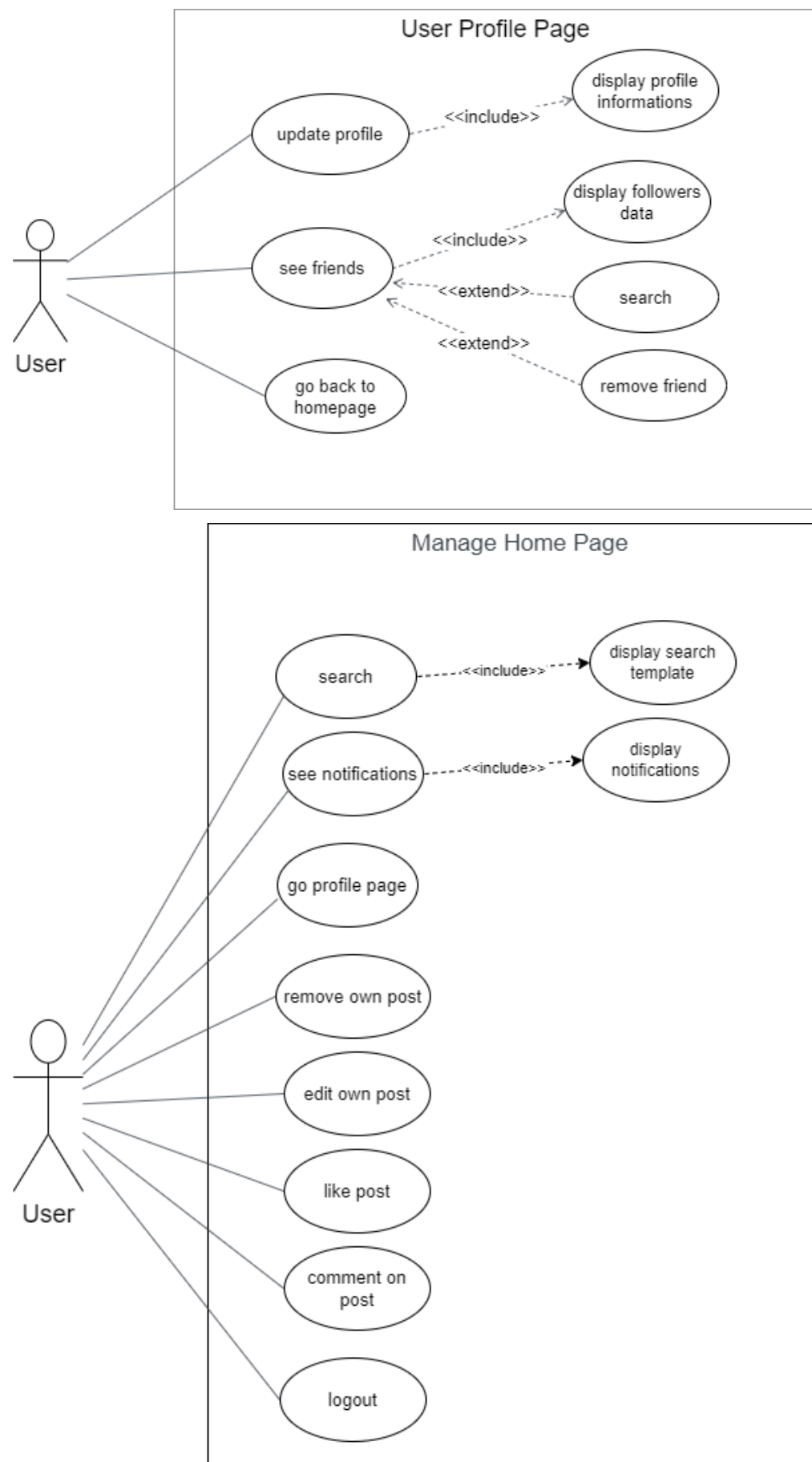
User Friends Page View		
Search friend: <input type="text"/>		
Friends:	 ali62	<input type="button" value="unfollow"/>
	 GandalfB2	<input type="button" value="unfollow"/>
	 SamwileG	<input type="button" value="unfollow"/>
	 FrodoB	<input type="button" value="unfollow"/>
	 Aragorn Gök	<input type="button" value="unfollow"/>

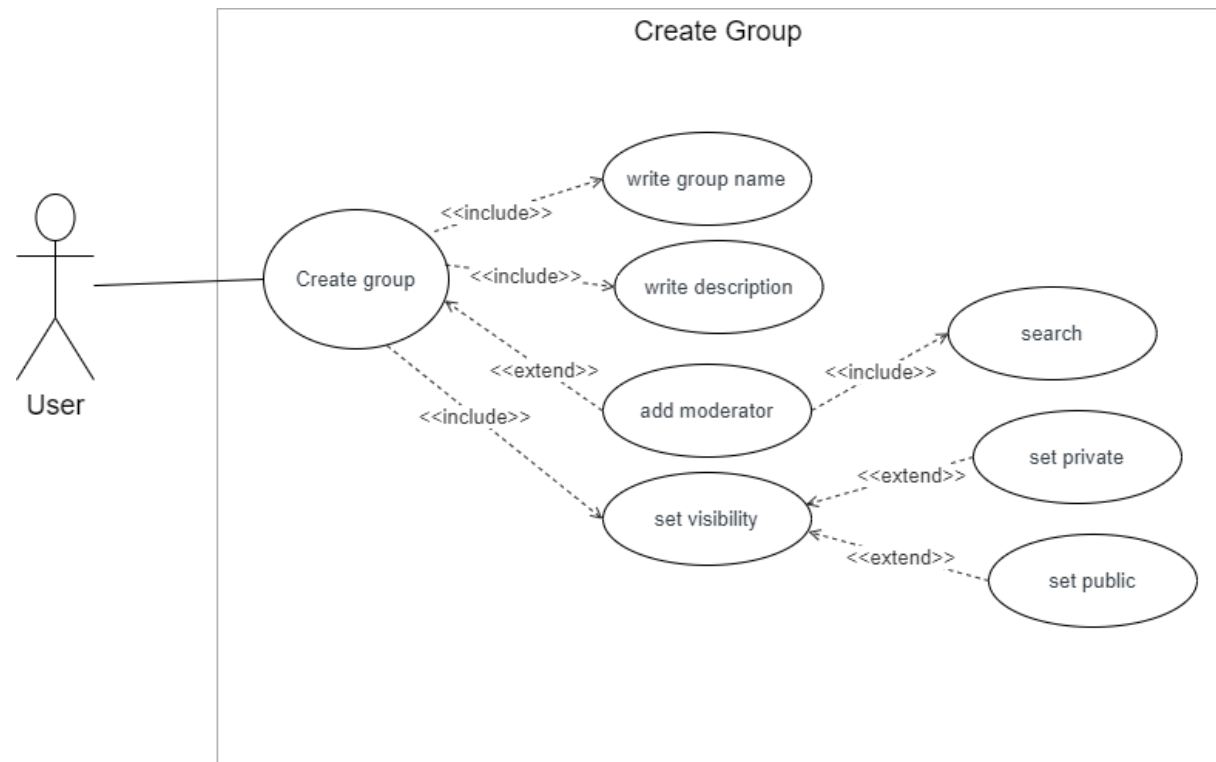
Create Post Template	
Template Name:	<input type="text" value="I saw silverback Uruk-hai"/>
Select field type 1:	<input type="text" value="Text"/> ▼
Select field type 2:	<input type="text" value="Photo"/> ▼
Select field type 3:	<input type="text" value="Video"/> ▼
Add Additional Field	
<input type="button" value="Create"/>	

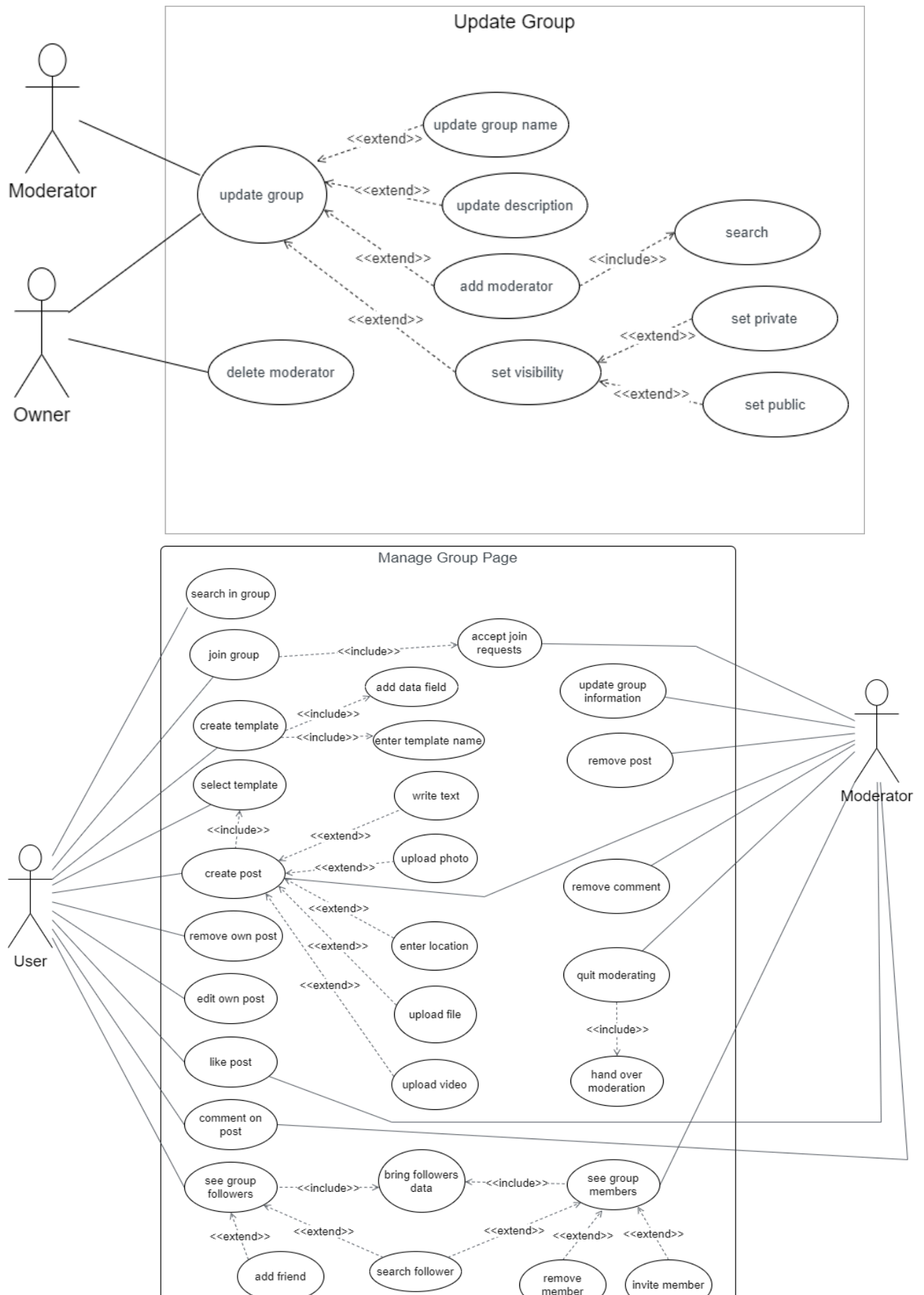
User Search Page	
Keyword :	<input type="text" value="Isengard"/> <input type="button" value="search"/>
Options:	<input type="checkbox"/> Users <input type="checkbox"/> Groups <input type="checkbox"/> Posts <input checked="" type="checkbox"/> All

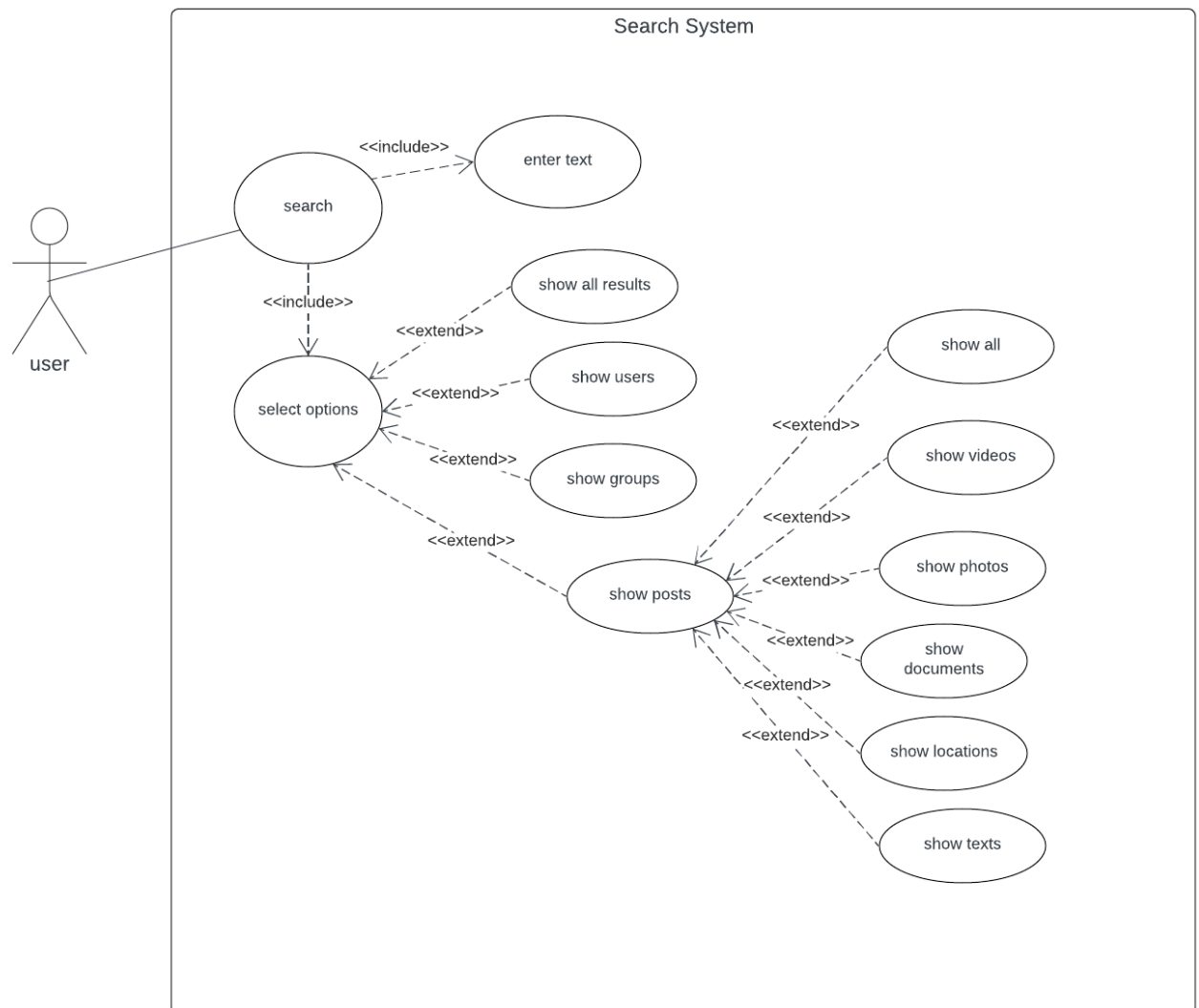
6.2 Use Case Diagrams



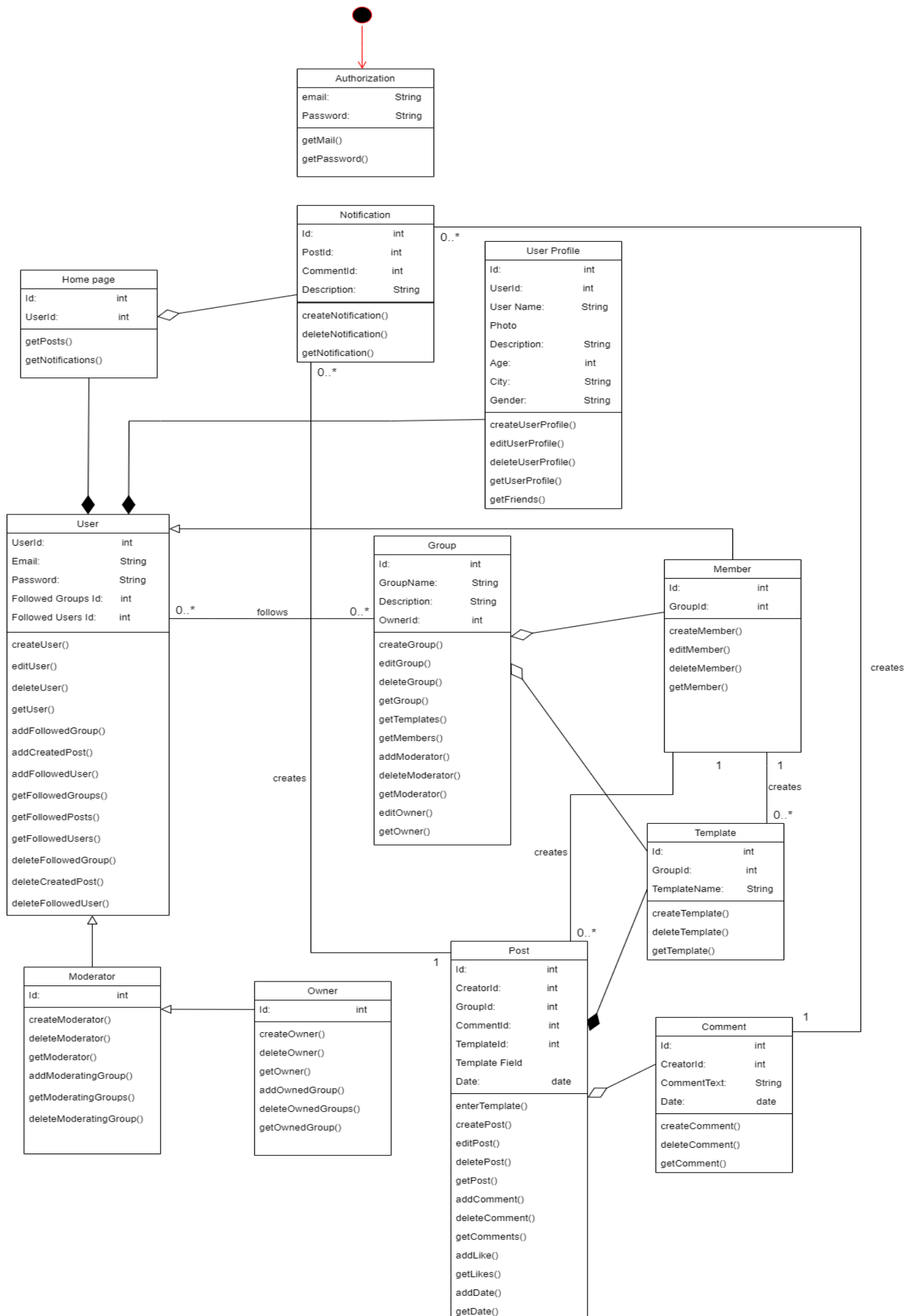




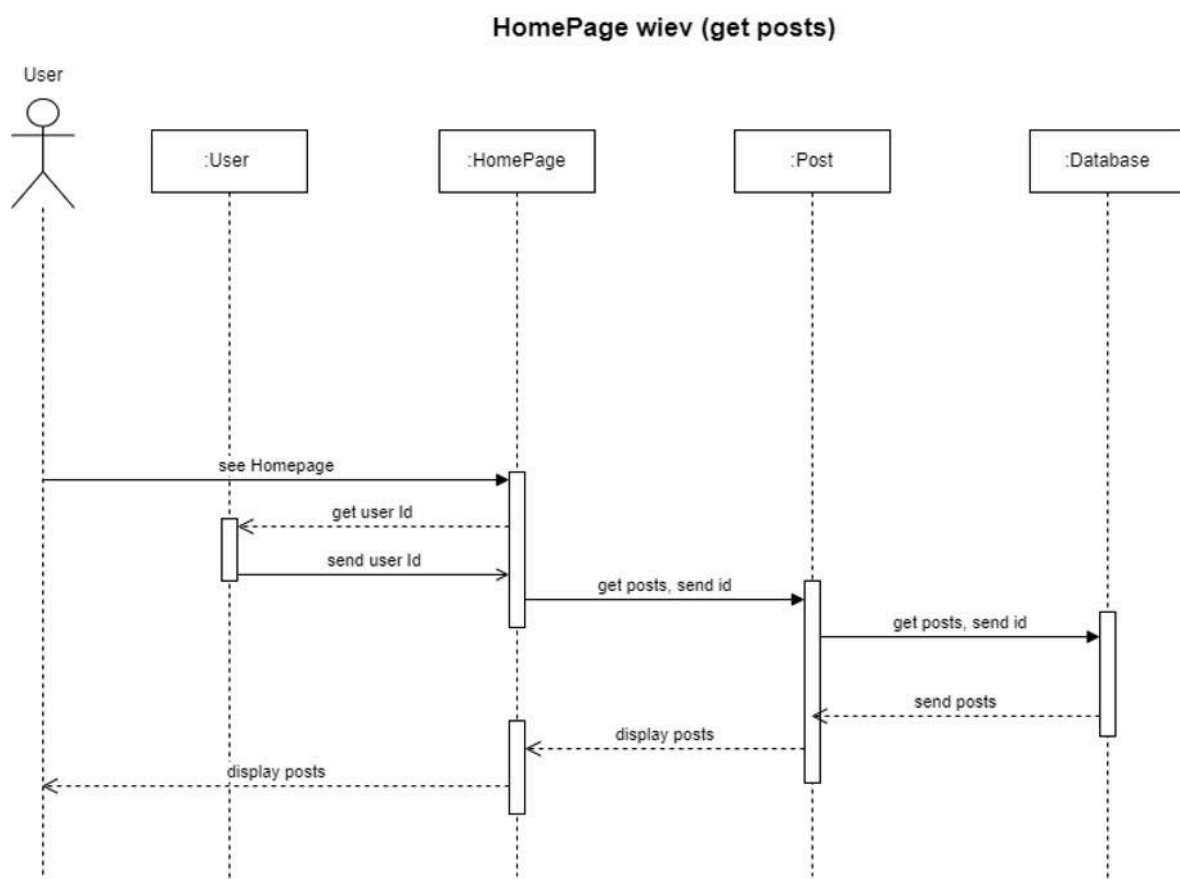
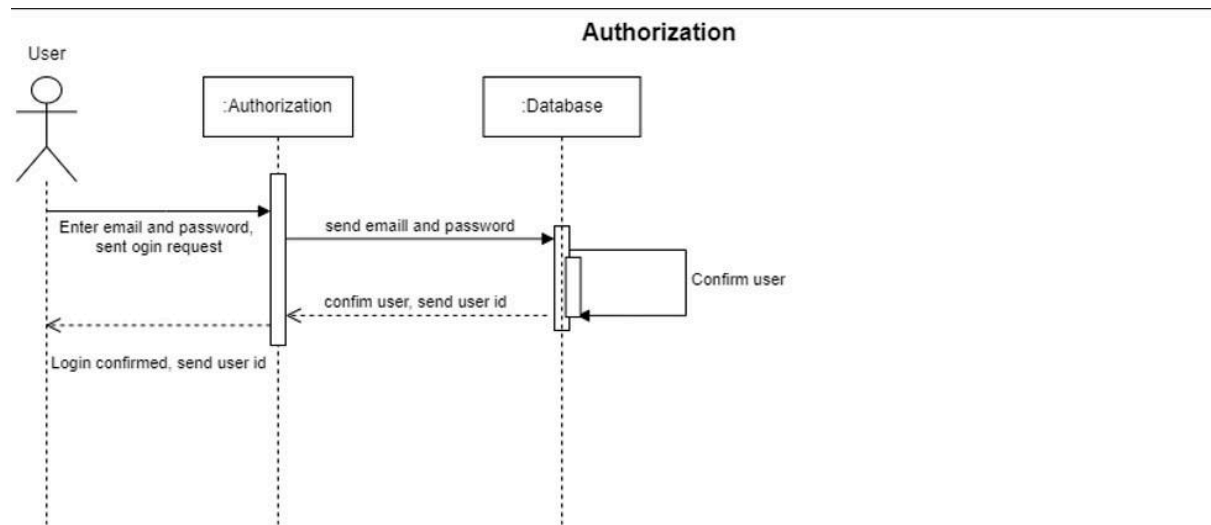


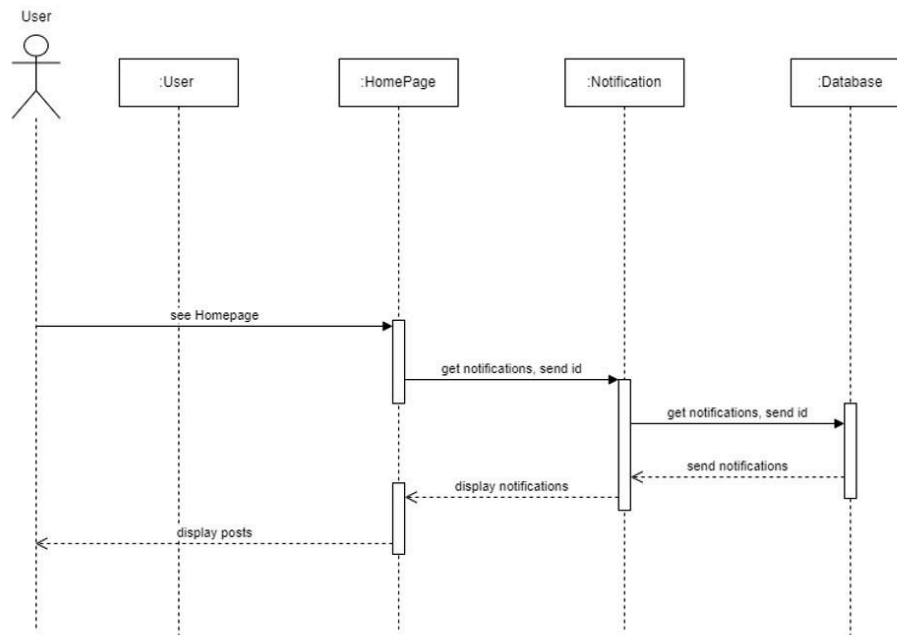
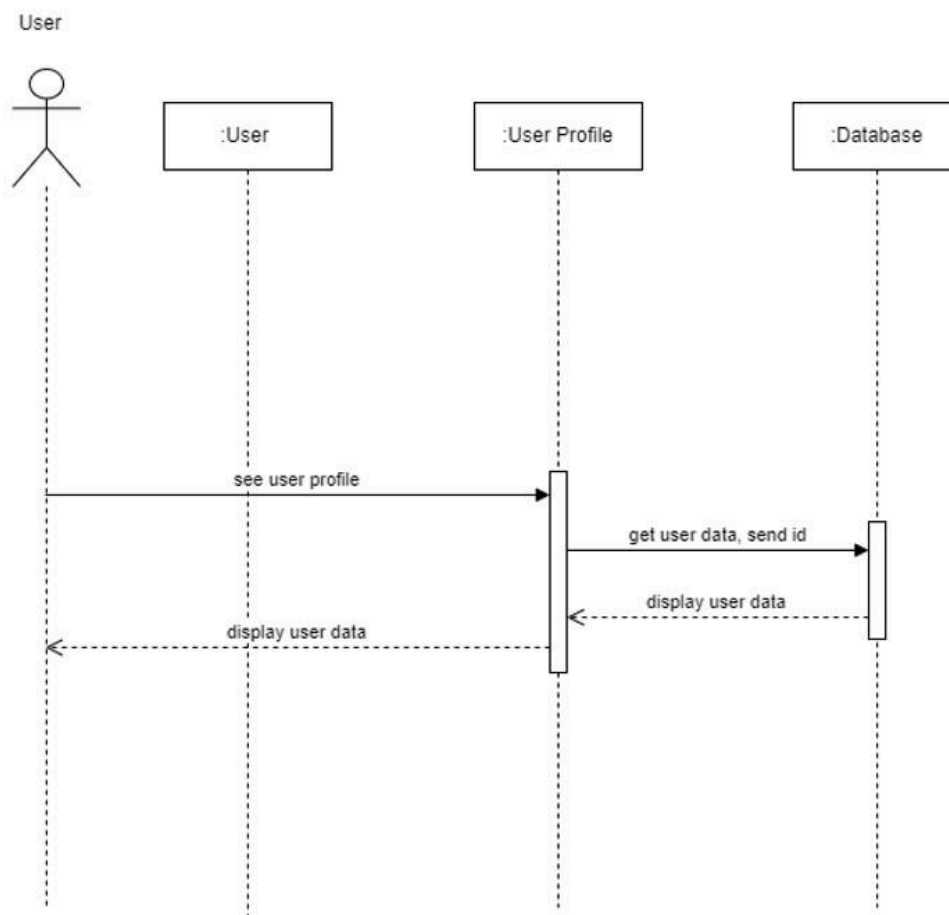


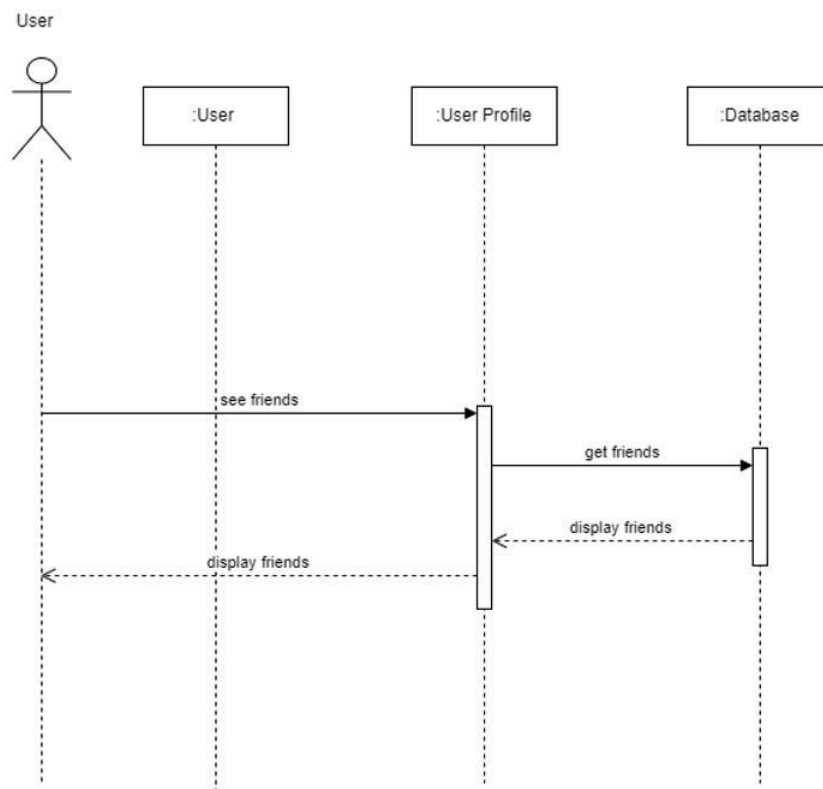
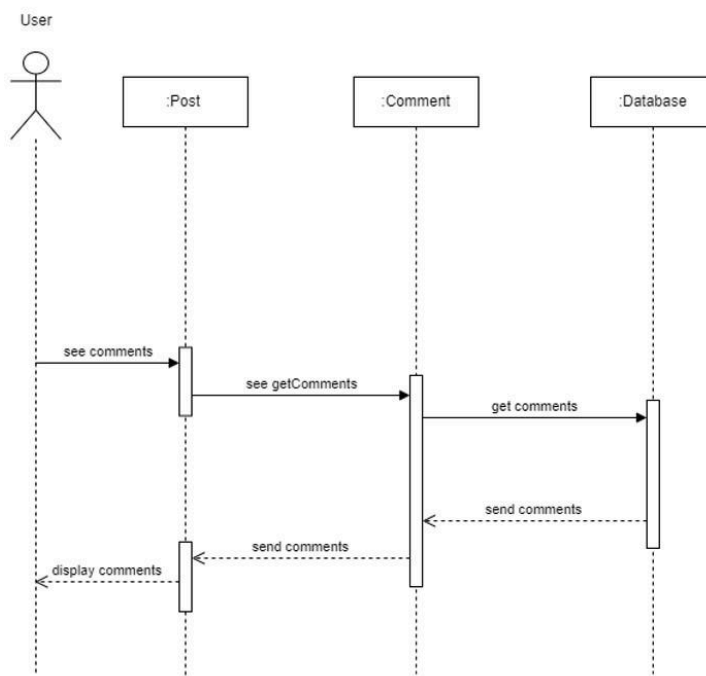
6.3 Class Diagrams

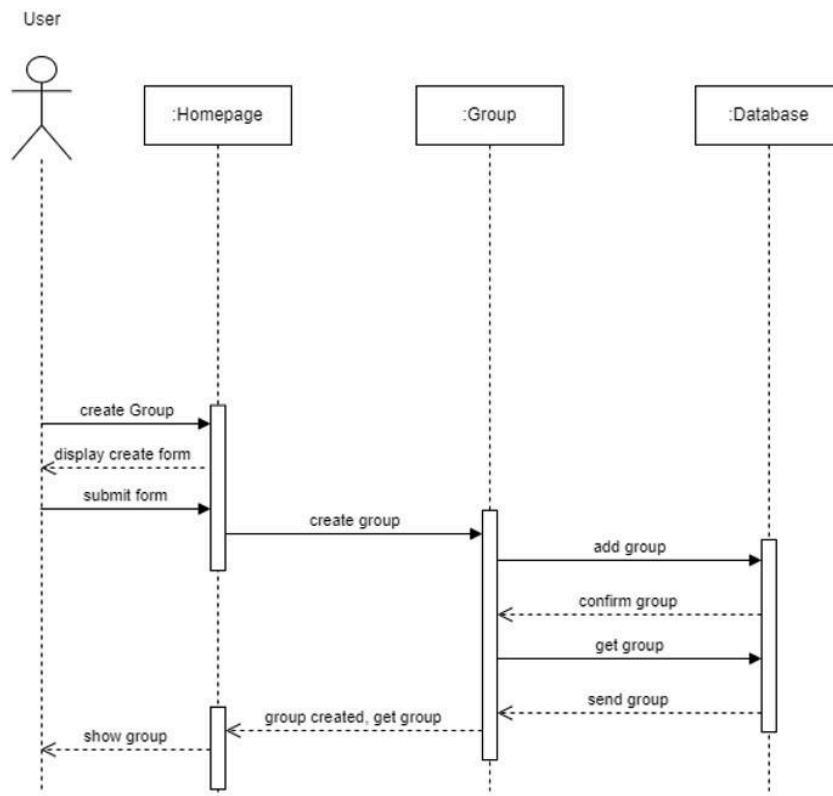
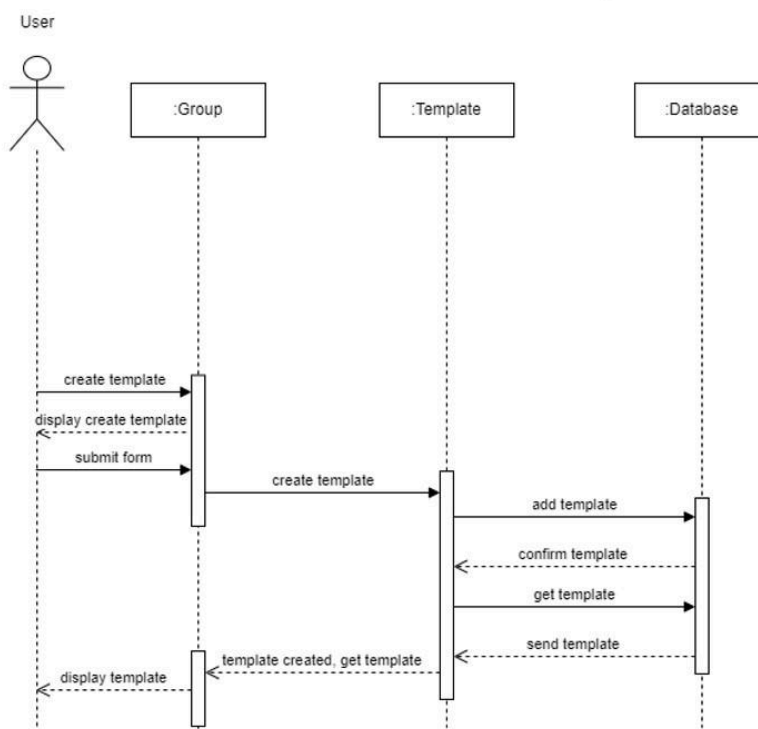


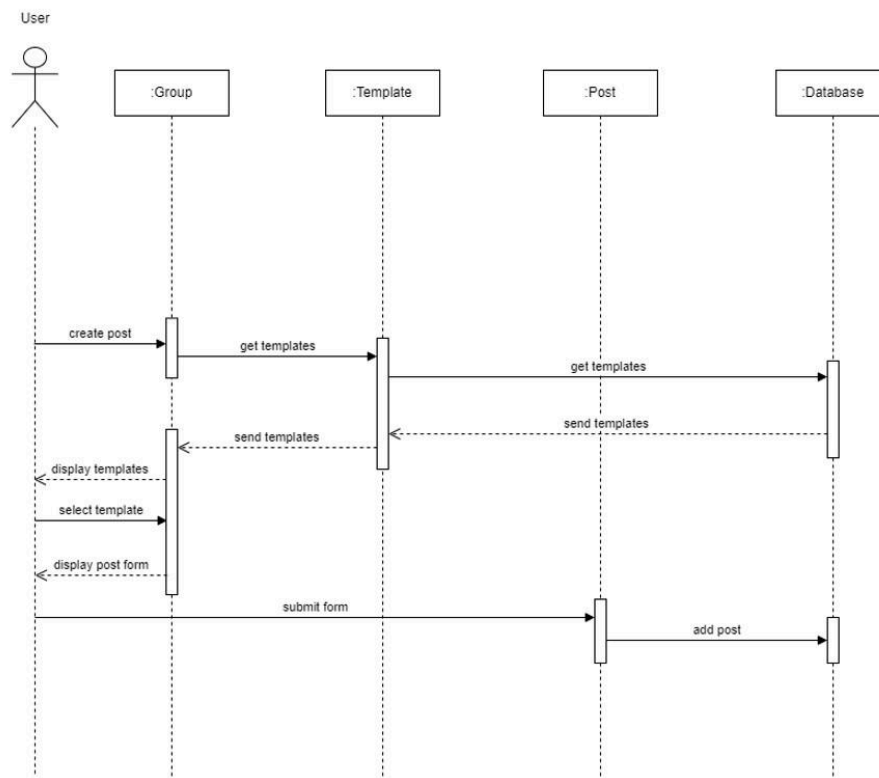
6.4 Sequence Diagrams



HomePage view (notifications)**See User Profile**

See Friends**See comments**

Create Group**Create Template**

Create Post**Group Page(See Posts)**