



Your Freedom in Learning

COMP 302

Software Engineering

Twitter Project

*Ece Özaydin 041801092
Mustafa Recep Ertuğrul 041901004
Oğuzhan Güngör 041901016
Aslı Yılmaz 041901019
Ahmet Faruk Turhan 041901038
Aydın Burak Kuşçu 041901043*

Submitted: 03.06.2022

Executive Summary

In this term project, the assignment was to create a social media platform similar to Facebook and Twitter, and work as an agile team in order to deploy the project and create the documents while developing the social media platform. In addition this report includes diagrams such as package diagram, class diagram, activity diagram, sequence diagram, component diagram, timing diagram and ER diagram.

Table of Contents	
Executive Summary	2
Table of Contents	3
Introduction	4
Goals and Objectives	4
Statement of Scope	4
Software Context	5
Project Organization	5
Unit Tests	6
Usage Scenario	6
User Profiles	6
User Stories and Acceptance Tests	6
Use Case Diagram	8
Use Case Descriptions	9
Agile Development & Daily Scrum Notes	10
Meeting Minutes	11
COCOMO II Time Estimation	13
Architecture	13
Logical View	14
Class Diagram	14
Communication Diagram	15
State Diagram	16
Process View	16
Activity Diagram	16
Sequence Diagram	22
Timing Diagram	29
Development View	30
Package Diagram	30
Component Diagram	31
ER Diagram	32
Deployment Diagram	33
Physical View	34
Physical View Diagram	34
Restrictions, Limitations, Constraints	35
User Manual	35
Conclusion	40

1. Introduction

The contents of this report include the UML diagrams for the Twitter project and the project organization, user manual for the project. This is the project for the software engineering lecture and as a team, we implemented agile development because in each iteration, we needed to implement new features to the website and with each implementation brought new bugs. Fixing bugs and improving the quality of the website required agile methodology to finish the project in time.

a. Goals and Objectives

In this project, the main goal of our team was to tweet successfully and create an account successfully. We performed those. One of the most crucial aspects of the social media website is adding friends as well. With native PHP codes, it proved to be a challenge and thus the secondary goal for our project has been born.

With the Twitter project, users can tweet, send messages to each other, share photos, comment under their friends' posts, like each other's posts and change their account details.

b. Statement of Scope

This chapter of the report explains the statement of the scope for the Twitter project.

Project Name	Twitter
Scope Description	<p>IN SCOPE:</p> <ul style="list-style-type: none"> -A server and domain from 000webhost to make our website accessible from everywhere. -A MySQL database provider to store our data.
Project Deliverables	<p>Fully functional Twitter website.</p> <p>Allow people to share their opinions around the globe for free.</p>
Acceptance Criteria	The user should be able to use every functionality of Twitter.
Constraints	Time is the only constraint.
Assumptions	Website should work fine without major errors.

The statement of scope of the Twitter project.

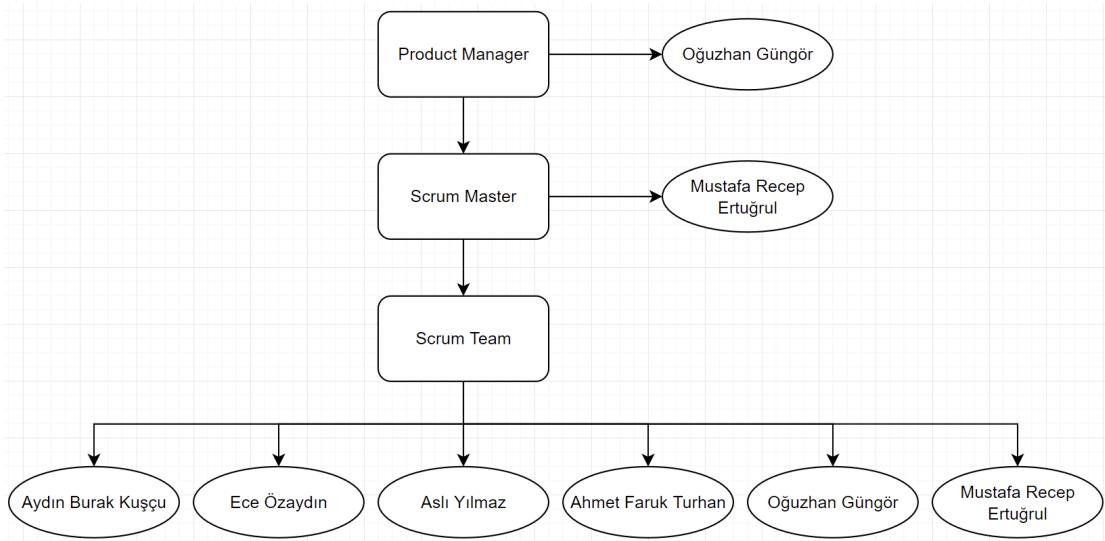
c. Software Context

This chapter of the report explains the software context of the Twitter project. Twitter is a social media platform that can be accessible via the web. The requirements for using the Twitter social media platform is an active internet connection. Users can send their tweets, a message about their ideas to share, like other users' tweets, add other users as friends, get notifications from other users, send messages to other users.

In the project, we used PHP, javascript, css, ajax and MySQL to perform the functionalities.

d. Project Organization

This chapter of the report explains the project organization. Oğuzhan was the product manager who managed the product backlog. Mustafa was the scrum master who held the scrum values and defended the scrum values. The scrum team was all of the project members since this was a group project and we all participated as a scrum team. This decision was made due to the time limitations and limited manpower for the project.



The Twitter project's project organization

e. Unit Tests

For the unit testing of our project, we wanted to utilize PHPUnit however due to the lack of knowledge of our team members, we couldn't figure out how to use PHPUnit. That's why we performed the unit tests manually after implementing a new feature to the website by trying to break down each of the components.

2. Usage Scenario

This term project helped us to understand problems when designing a program and how to overcome those problems. This project helped us how to plan as a team, how to work as a team. These skills that we've obtained from this project taught us how to approach a real life problem, and how to overcome those problems in a limited time.

a. User Profiles

This chapter of the report explains the targeted audience for the Twitter project. Since the project is based on social media, the targeted audience is the younger generation aged 18-30. The users must have an active internet connection in order to use the application. Those who are interested in connecting with friends are expected to use the application.

b. User Stories and Acceptance Tests

This chapter of the report explains the user stories and acceptance tests generated by the team members of our project.

User stories are as follows:

- As a user I want to search people's names or tweets so that I can find what I look for.
- As a user I want to be able to send messages privately so that I can chat with my friend freely.
- As a user I want to be able to customize my profile so that I can represent myself to other people in the best way possible.
- As a user I want to follow other users so that I can see their tweets on my feed
- As a user I should be able to moderate my tweets so I can change my tweets.
- As a user, I should be able to login to my account, so that I can use the website properly.
- As a user, I should be able to register to the website so that I can use the website properly.
- As a user, I should get notifications for users that I follow so that I can stay on topic with content creators.
- As a user I can tweet so that the other users can see my opinion
- As a user I want to post a video and then play the video so that others can play my videos or I can play other videos.
- As a user I want to see what is trending on the website so that I can better understand what's going on with the world.

As for the acceptance tests for the Twitter project, they are as follows:

- Given a user registers to our website, when they enter their username and given password to our website, the home page should display their username as well as the homepage should allow them to post anything under their username.
- Given a user sends a message to their friend, when the other user wants to see it, other users shouldn't be seeing the sent message and the friend should be able to see the message.
- Given when a user wants to customize their profile, they should be able to change their profile picture, change their account name, change their email or delete their profile.
- Given a user wants to delete their tweet, when they click remove they should be able to remove that tweet from the website.
- Given a user wants to share an image, when they post it on the Twitter page, then their friends should be able to see that image.

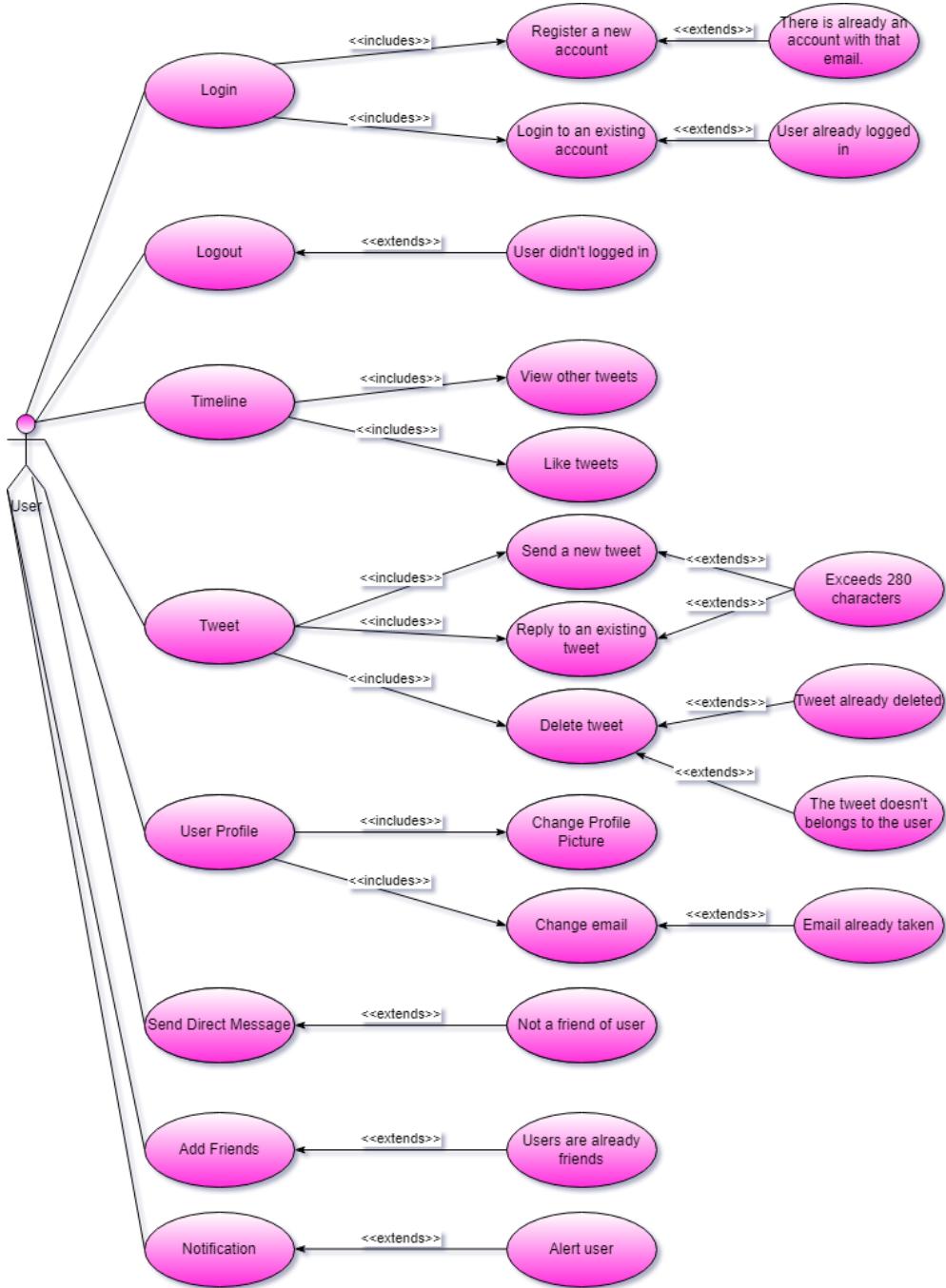
- Given a user wants to share a youtube video, when they copy a youtube video's link and post it on the Twitter page, then their friends should be able to run it on the Twitter page.

As for the non functional requirements for the Twitter project, they are as follows:

- Twitter page should support sending images to the server.
- Twitter should support sending 1000 private messages.
- Twitter page should load 10 posts per page.
- Twitter page should support 1000 users.
- Twitter page should respond to sending tweets in 10 ms.

c. Use Case Diagram

This chapter of the report is dedicated to the use case diagram for the Twitter Project. Our use case diagram consists of 8 main ellipses which symbolize the core functionality of the Twitter project. These core functionalities are: Login, logout, timeline, tweet, user profile, send direct message, add friends and finally notifications. The rest of the use case diagram is self explanatory.



d. Use Case Descriptions

This chapter of the report explains the use case descriptions for the Twitter project.

The first 3 parts of the use case descriptions are the login, logout and timeline. In the login event, the actor logs in the Twitter website using their already registered account. The logout event occurs when the user wants to logout from the website. In that case, the user clicks to the logout button and the user logs out. As for the timeline, the user adds friends, and their posts will show up on the user's feed.

Login

Use Case	Login
Primary Actor	<i>User</i>
Goal in Context	To successfully login into an existing account
Preconditions	Account must exist
Scenario	1.User enters the website. 2.User logs in.
Exceptions	-There is already an account with that email. -User already logged in.

Logout

Use Case	Logout
Primary Actor	<i>User</i>
Goal in Context	To successfully logout from an account.
Preconditions	User must be logged in
Scenario	1.User clicks to logout button. 2.User logs out.
Exceptions	-User already logged out.

Timeline

Use Case	Timeline
Primary Actor	<i>User</i>
Goal in Context	To be able to see other tweets and interact with them.
Preconditions	User must be logged in.
Scenario	<ol style="list-style-type: none">1. User enters the website.2. User logs in.3. User redirected to timeline.
Exceptions	-User not logged in.

Tweet

Use Case	Tweet
Primary Actor	<i>User</i>
Goal in Context	To send a new tweet. To reply to an existing tweet. To delete a tweet.
Preconditions	Account must exist
Scenario	1.User enters the website. 2.User logs in. 3. User redirected to timeline. 4.User writes a new tweet or deletes one.
Exceptions	-Tweet already deleted. -New tweet exceeds 280 characters.

User Profile

User Profile	
Use Case	User Profile
Primary Actor	<i>User</i>
Goal in Context	To change profile picture. To change username.
Preconditions	User must be logged in.
Scenario	1. User enters the website. 2. User logs in. 3. User redirected to timeline. 4. User clicks on his profile link.
Exceptions	-Username already taken.

Send Direct Message

Use Case	Send Direct Message
Primary Actor	<i>User</i>
Goal in Context	Sending a direct message to another user.
Preconditions	<ul style="list-style-type: none"> -User must be logged in. -Other user must be added as a friend.
Scenario	<ol style="list-style-type: none"> 1.User enters the website. 2.User logs in. 3. User redirected to timeline. 4.User clicks on the DM button.
Exceptions	-Other user not added as a friend.

Add Friends

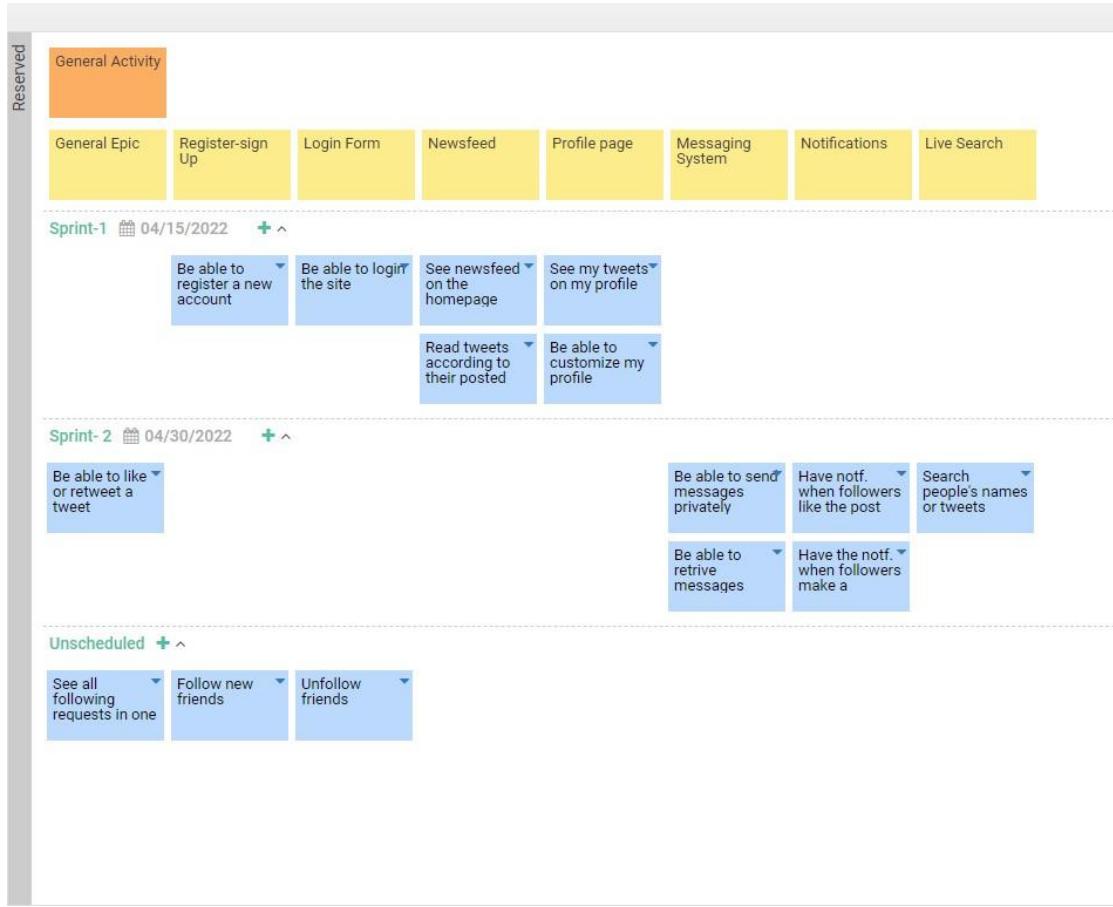
Use Case	Add Friends
Primary Actor	<i>User</i>
Goal in Context	To add another user as a friend.
Preconditions	User must be logged in.
Scenario	<ol style="list-style-type: none"> 1.User enters the website. 2.User logs in. 3.User redirected to timeline. 4.User clicks add friend button.
Exceptions	-Both users are already friends.

Notification

Use Case	Notification
Primary Actor	<i>User</i>
Goal in Context	To notify user, if a new message is received or someone replied to a tweet of the user.
Preconditions	User must be logged in.
Scenario	<ol style="list-style-type: none">1. User enters the website.2. User logs in.3. User redirected to timeline.4. User clicks on notifications.
Exceptions	-User doesn't log in.

3. Agile Development & Daily Scrum Notes

In this part of the report, the agile development steps, daily scrum notes, poker planning game and meeting minutes will be documented. The product backlog and sprint backlog can be observed in the 2 sprint cycles for the project.



2 Sprint Cycles for the Twitter Project.

a. Meeting Minutes

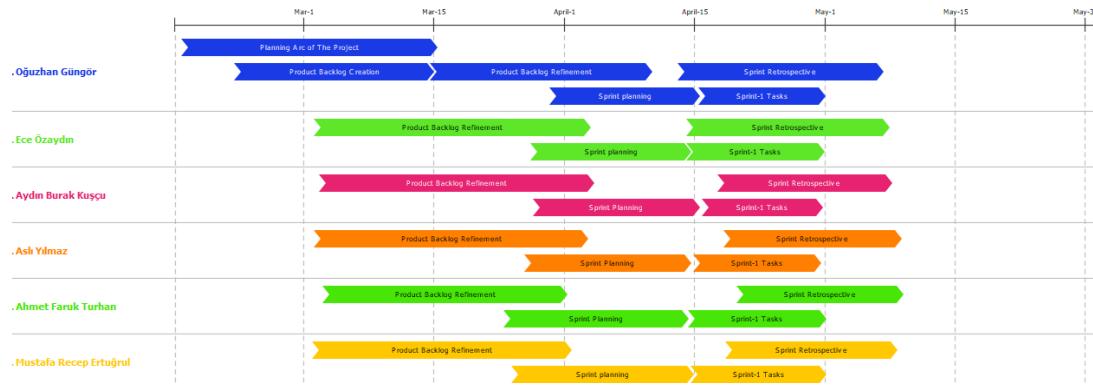
This chapter of the report is dedicated to the team's meeting minutes. The frequency of the meetings has been decided by the result of the COCOMO II time estimation. The COCOMO II time estimation will be mentioned in its corresponding chapter.

1st meeting conducted on 10.04.2022: We planned how to approach the twitter project, we gathered some articles, some documents about the project. We rewatched our lectures from the beginning.

2nd meeting conducted on 17.04.2022: We calculated daily sprint velocity. Our product manager created our product backlog and prioritized our sprint backlog. Then today's backlog tasks expanded by team. For today's sprint backlog, we prioritized the login and register pages due to the product manager's prioritization. Then the team with the monitoring of the scrum master, played the planning game. We decided that the login handler function, the function that checks for the user that enters the credentials, has a difficulty of 5 units. As for the register handler, we reached a consensus that it is similar to the login handler therefore it has a difficulty of 5 units. As for the register page's frontend, it has a difficulty of 8 units because we needed to use javascript and the team's familiarity with javascript isn't that high, that's why we pointed it to be 8. Since we reached a total of 18 units of work and our velocity is 20, we started to work on our sprint backlog.

3rd meeting conducted on 22.04.2022: We calculated today's daily sprint velocity. Our product manager assigned us the index page task. We played the planning game and decided that making the index page with navbar and the static version of the index page is 13 units of work therefore making the navbar and index page with their respected documents would overflow our velocity because creating the user page, sending tweet functions were 13 units of work as well. We started today's daily sprint with creating the navbar, the static html version of the index page as well as optimizing the database.

The sprint cycle conducted after 3rd meeting has been shown down below:



The sprint cycle after 3rd meeting

4th meeting conducted on 26.05.2022: We calculated today's daily sprint velocity. Since our deadline is close, we increased our velocity and manpower. We completed the infinite scrolling, posting comments.

5th meeting conducted on 27.05.2022: We calculated today's daily sprint velocity. In today's daily sprint, fixing bugs was the priority for the project. We separated the team by bug fixers and feature implementers. Feature implementers kept implementing new features and bug fixers fixed the bugs as we moved on with the project. Today, feature implementers implemented the profile page and bug fixers fixed the infinite scrolling bug and posting bugs.

6th meeting conducted on 28.05.2022: We calculated today's daily sprint velocity and feature implementers started to implement the messaging system and bug fixers tried to find bugs in the project.

7th meeting conducted on 29.05.2022: We implemented the search and notification feature since there were no found bugs, bug fixers joined the feature implementers.

8th meeting conducted on 30.05.2022: We implemented sending jpg, jpeg and png files to the website and opening youtube videos.

9th meeting conducted on 31.05.2022: We finished documentation of the Twitter project.

b. COCOMO II Time Estimation

This chapter of the report explains the COCOMO II time estimation used to manage the time of the project. The calculation used for the project as follows:

$$\text{PM} = 2.94 \times \text{Size}^{\{B\}} \times M$$

$$TDEV = 3 \times \text{PM}^{(0.33 + 0.2 \times (B - 1.01))}$$

In our project:

- Size of code : 1000 lines = 1 KLOC,

- **B → 1.11**

Precedentedness - High (2)

Development flexibility - Very High (1)

Architecture/risk resolution - Nominal (3)

Process maturity - Nominal (3)

Team cohesion - Very High (1)

- **M → 1**

RCPX - product reliability and complexity - Nominal (1)

RUSE - the reuse required - Nominal (1)

PDIF - platform difficulty - Nominal (1)

PREX - personnel experience - Nominal (1)

PERS - personnel capability - Nominal (1)

SCED - required schedule - Nominal (1)

FCIL - the team support facilities - Nominal (1)

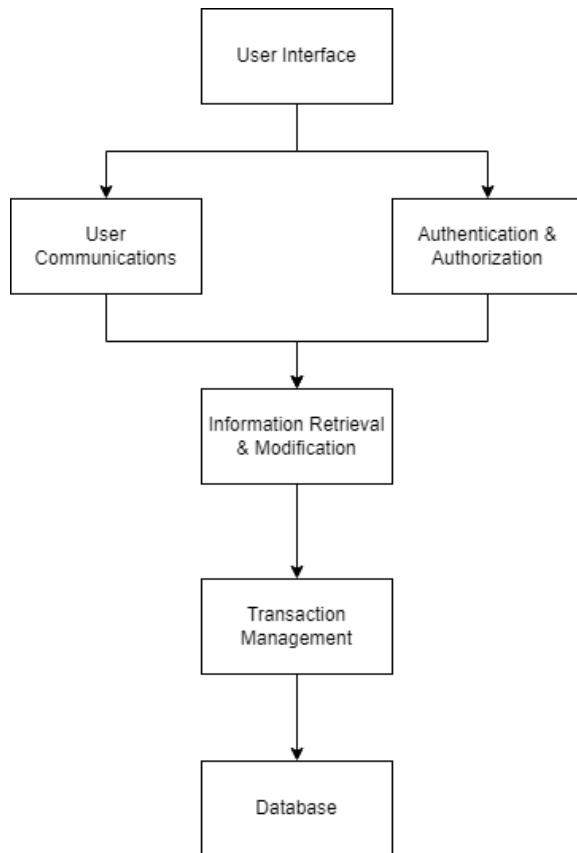
$$\text{PM} = 2.94 \times 1^{(1.11)} \times 1 \rightarrow 2.94 \text{ person - month}$$

$$TDEV = 3 \times 2.94^{(0.33 + 0.2 \times (1.11 - 1.01))} \rightarrow 4.37 \text{ months}$$

Since the time to complete the project exceeds the project's deadline, we decided to remove some of the features from the Twitter project and focused on the core concepts to make the website functional.

4. Architecture

For the Twitter project, we chose the Layered Information System Architecture because it suited our knowledge and understanding of the inner workings of web development. In the architecture, User interface has been branched into authentication & authorization because without authentication & authorization, users cannot interact with the website. User communications is an important aspect of the project because we are working with social media therefore communication is key. These aspects communicate with the database.



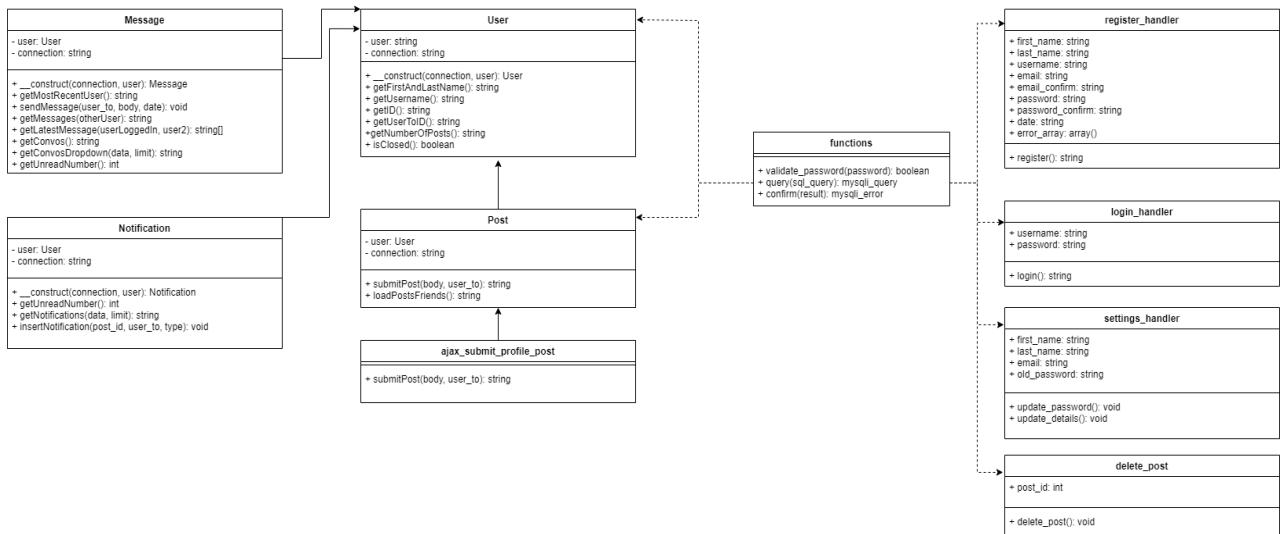
The Layered Information System Architecture used in this project.

a. Logical View

This chapter of the report contains the class diagram and the communication diagram for the Twitter project. Under the logical view, the Twitter project's class diagram, communication diagram and state diagram can be seen.

Class Diagram

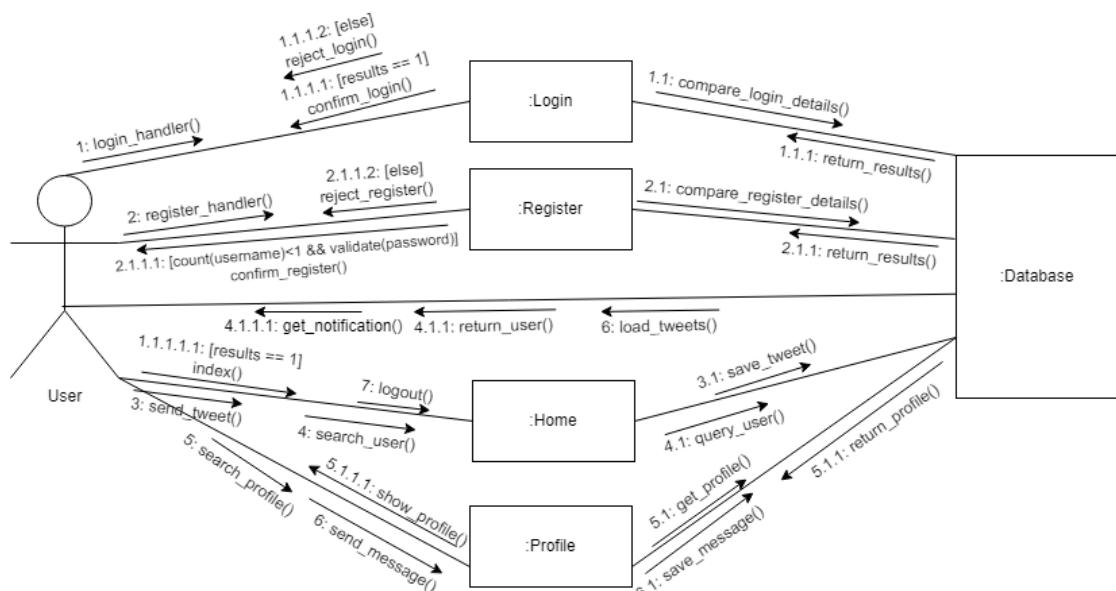
This chapter of the report shows the class diagram for the Twitter project. We have a functions class which contains the frequently used functions. We also have classes that use other classes and these are ajax classes. Their functionality is that, collecting functions at separate files so that when debugging, finding bugs would be easier. Other than that, we have 4 main classes and these are User, Post, Notification, Message. User is the main object since the whole project is based on a user, we thought that putting user to the center would be best fit for the project. Under the Post class, we have the ajax_submit_profile_post and other ajax classes as well. These classes, like we mentioned previously, are only separating the functions from the classes.



The class diagram for the Twitter Project.

Communication Diagram

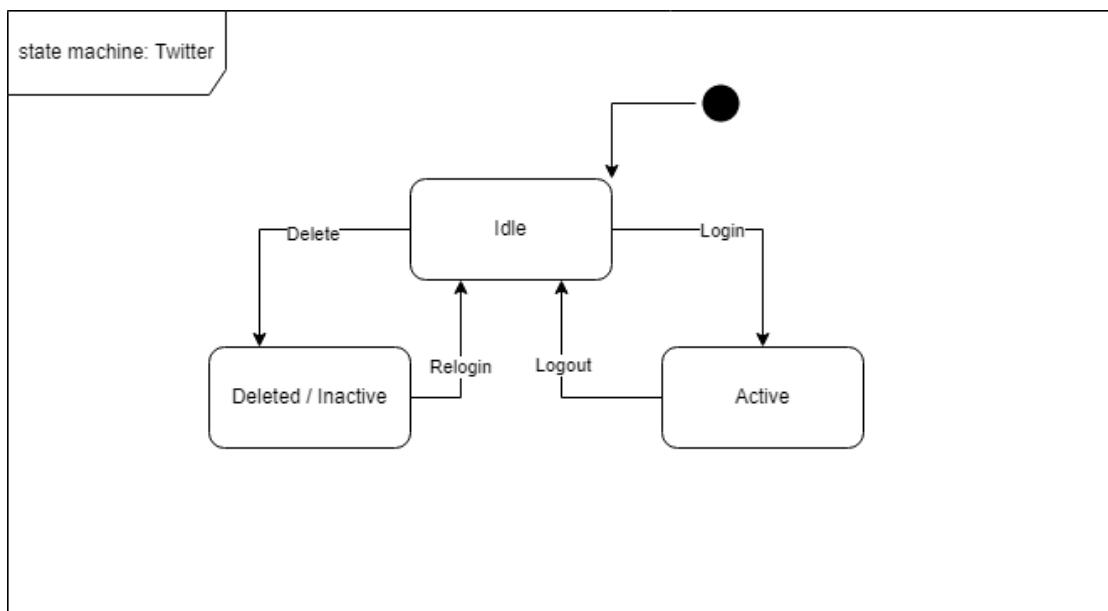
This chapter of the report explains the communication diagram for the Twitter project. The communication diagram consists of 5 main components and these are Login, Register, Home, Profile and Database. If a user interacts with another user via private messaging, it involves the Profile component. If a user interacts with another user via commenting under their post, it involves the Home component. And the login and register components, as their name suggests, handle the logging in and registration processes.



The communication diagram for the Twitter project

State Diagram

This chapter of the report shows the state diagram of the Twitter project. The following is the state diagram for a user account. It has three states; Idle, Active and Deleted. And transitions between these states. When a user opens an account, it is in the idle state. Then if the user logs into their account, its state becomes active. And if they log out from an active account, the account's state returns to idle. Also when the users delete their accounts, the account's state becomes deleted or inactive, which can be reverted if they login again.



The State Diagram of the Twitter Project.

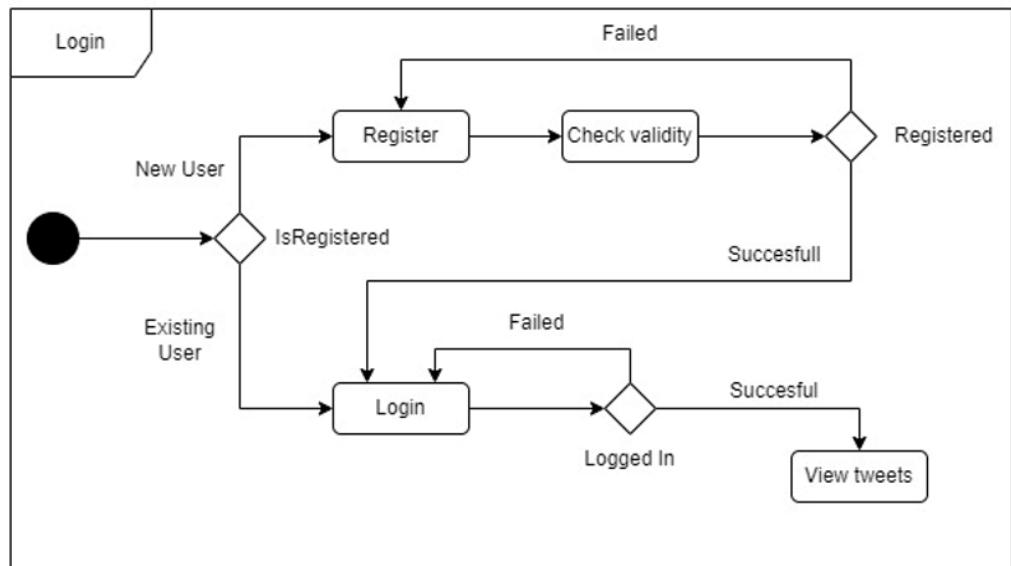
b. Process View

This chapter of the report contains the activity diagram sequence diagram and the timing diagram for the Twitter project.

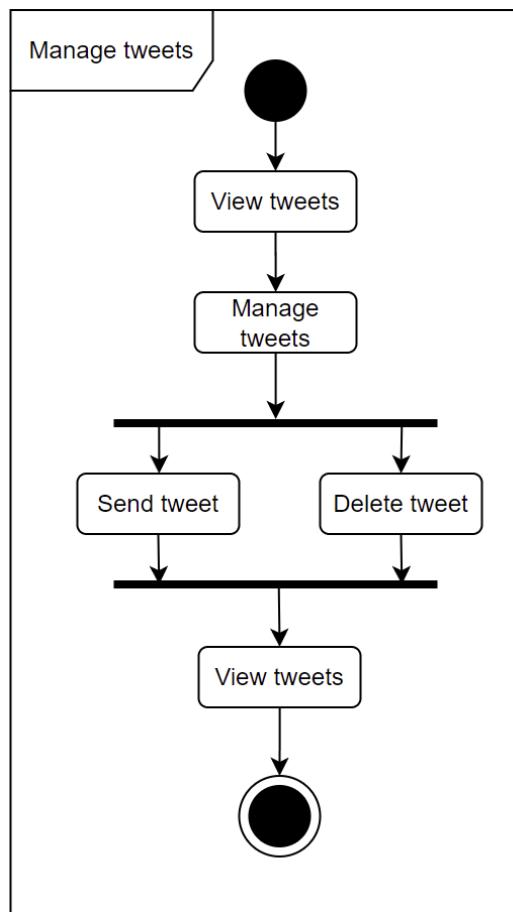
Activity Diagram

This chapter of the report explains the activity diagram of the Twitter project.

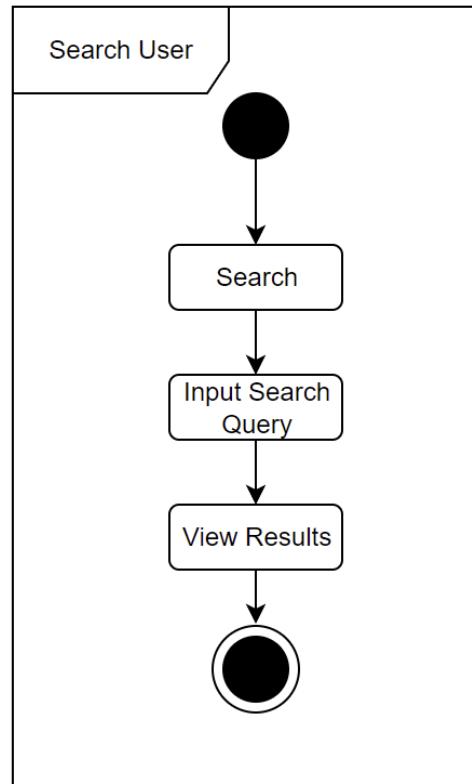
The activity diagrams represent the flow of control within our website. And the following diagram shows the login process for our website. It starts from the left and checks if the user is registered. If not, they go through the registration process. After they successfully register, they can login to their accounts and view the tweets.



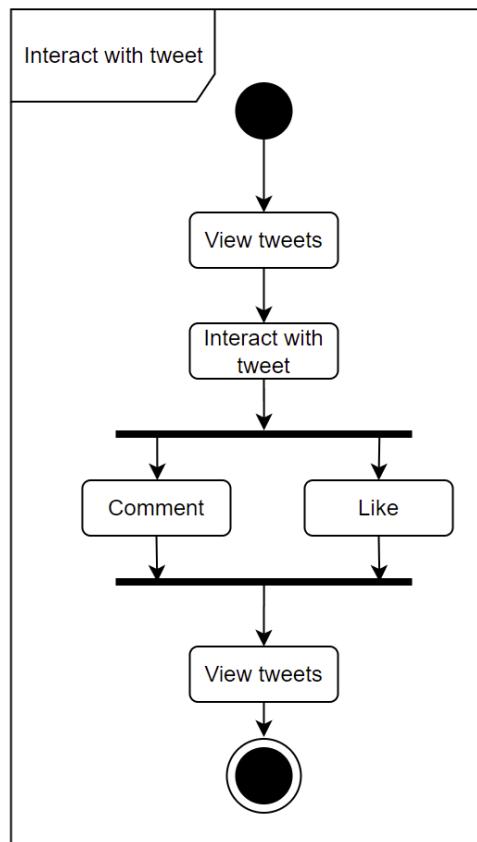
The activity diagram for the login process of the Twitter project.



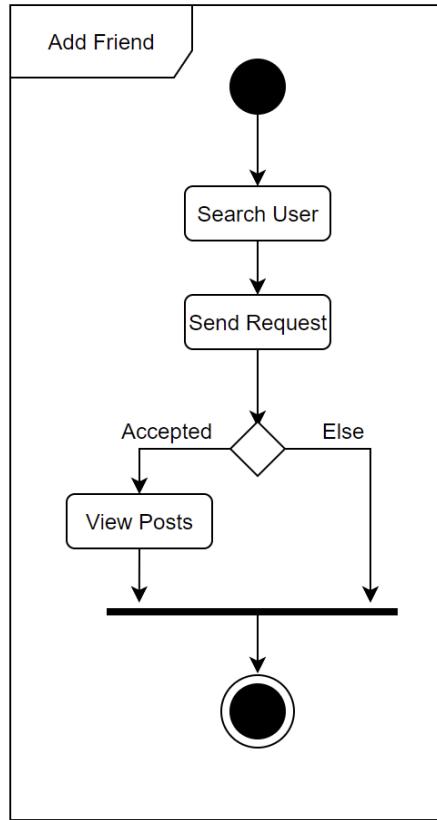
The activity diagram for managing tweets.



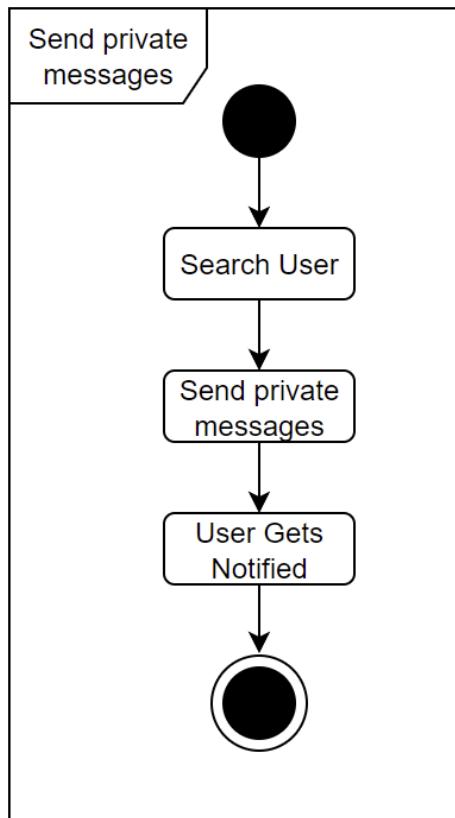
The activity diagram for searching users.



The activity diagram for interacting with a tweet.



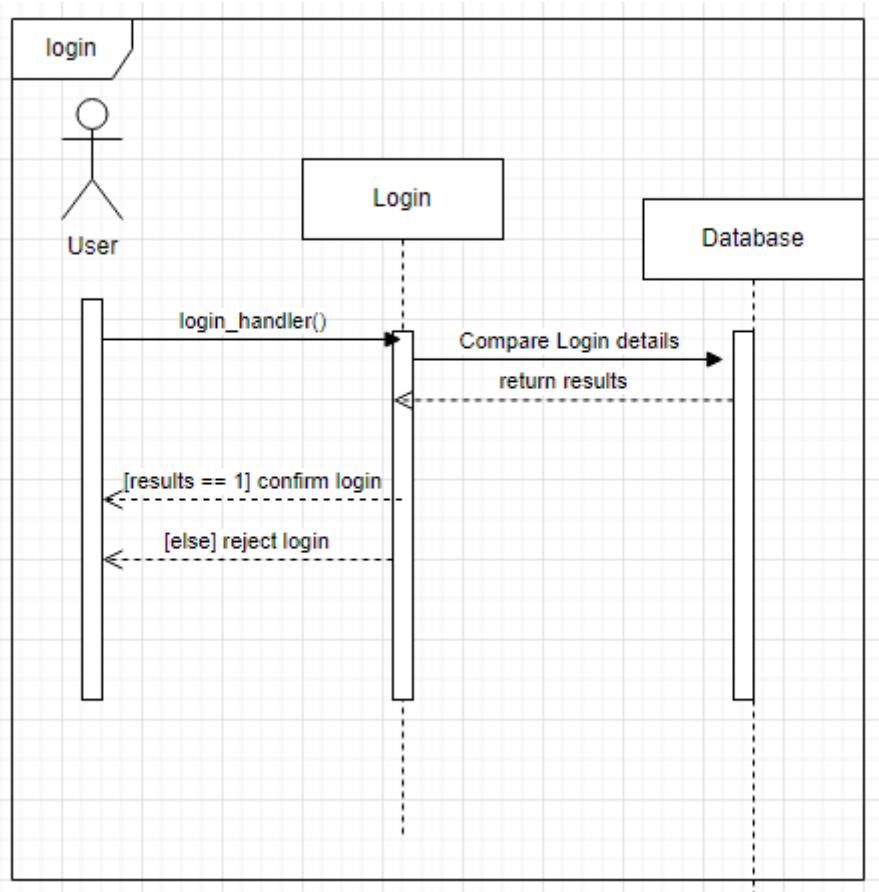
The activity diagram for adding a friend.



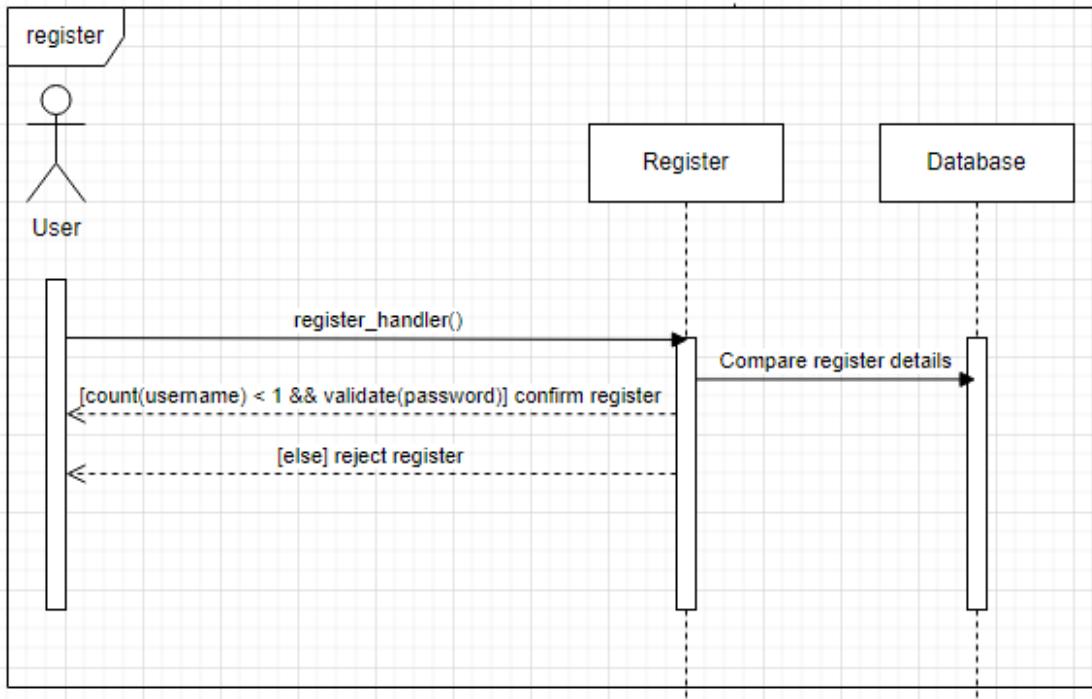
The activity diagram for sending private messages.

Sequence Diagram

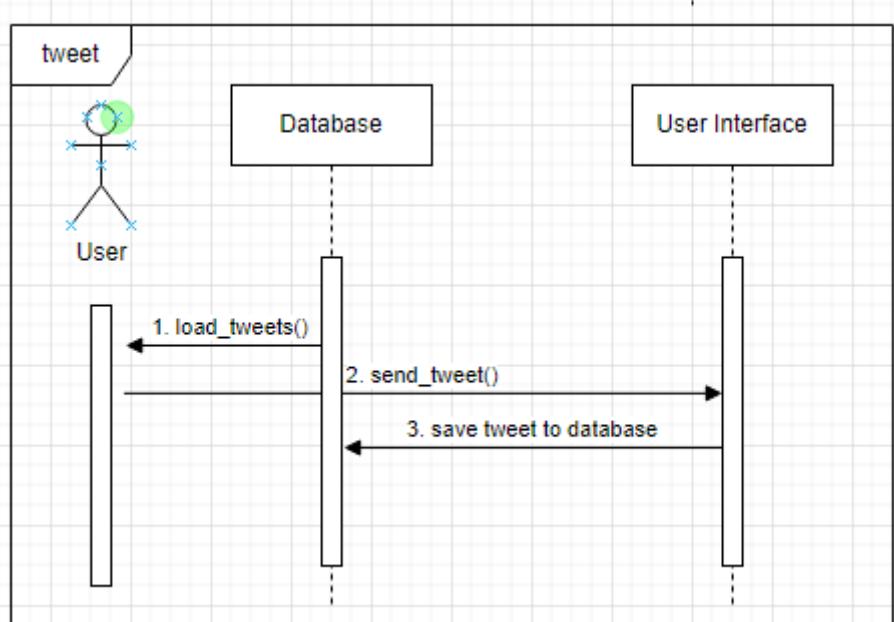
This chapter of the report is dedicated to the sequence diagrams of the Twitter project. There are 14 main operations that Twitter website can perform and these can be observed in the sequence diagrams.



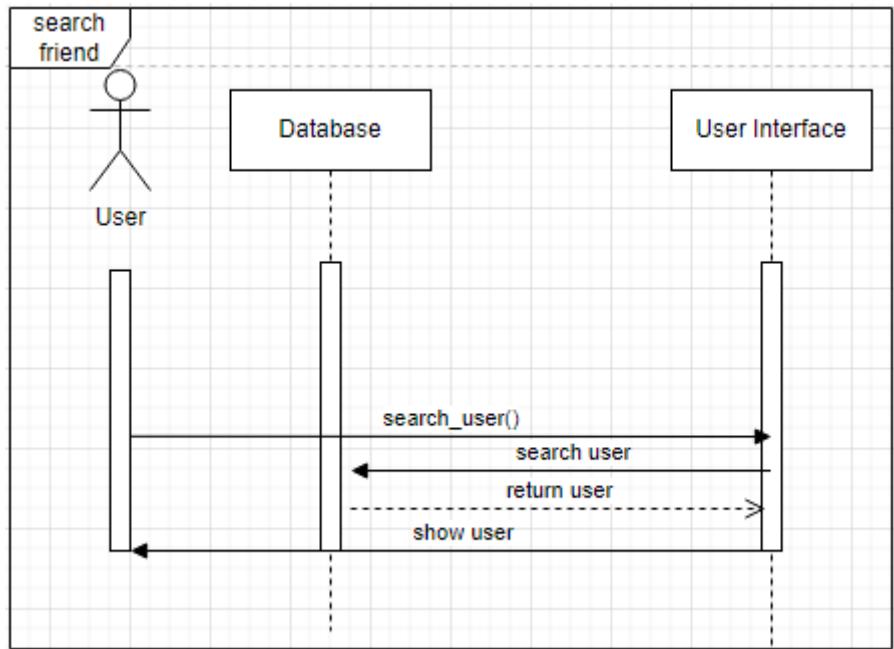
The sequence diagram for login operation



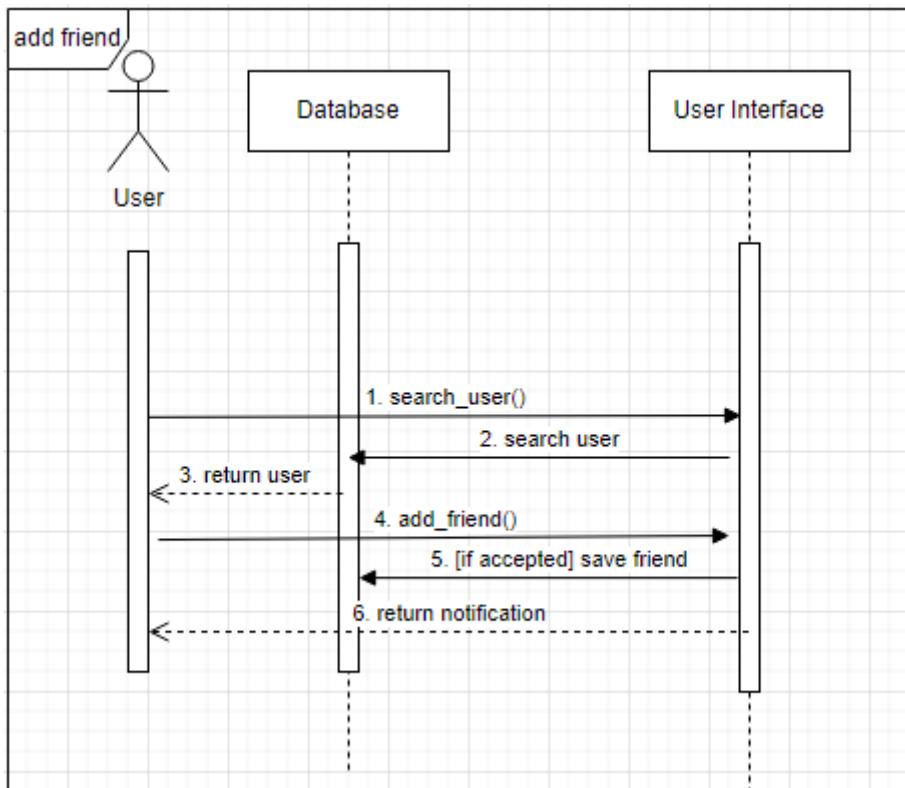
The sequence diagram for register operation



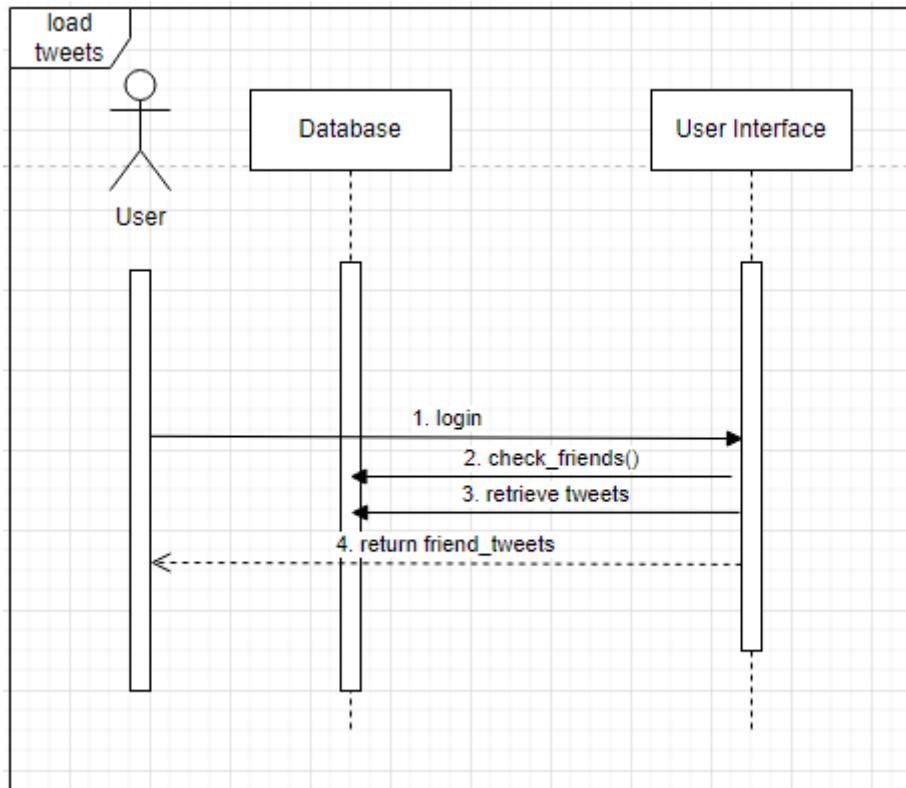
The sequence diagram for sending tweets



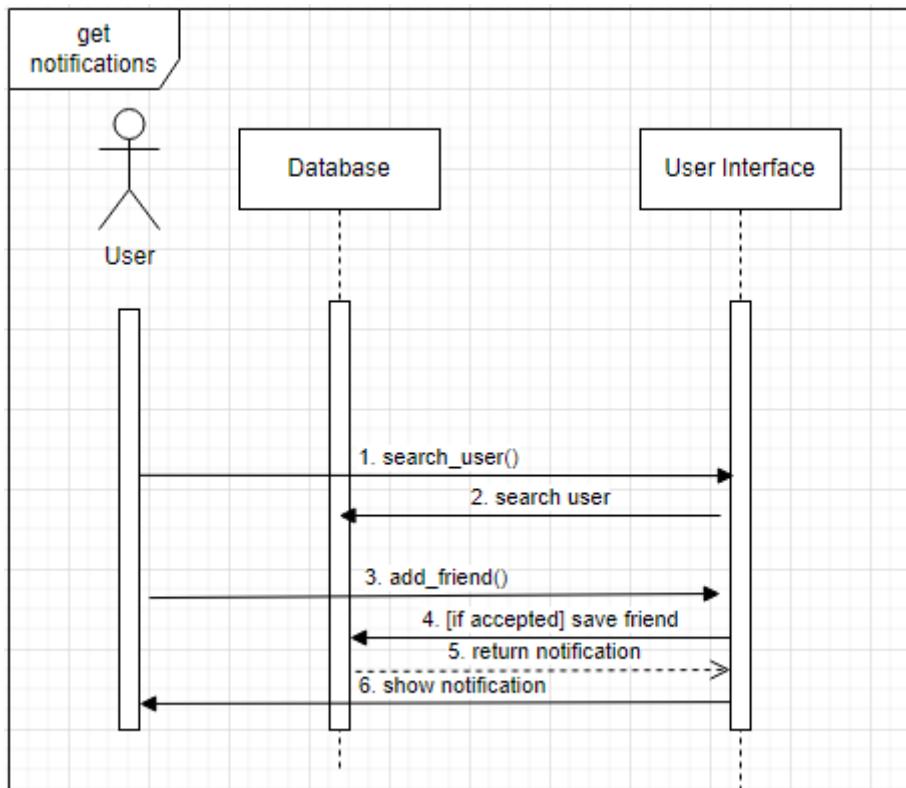
The sequence diagram for searching friend



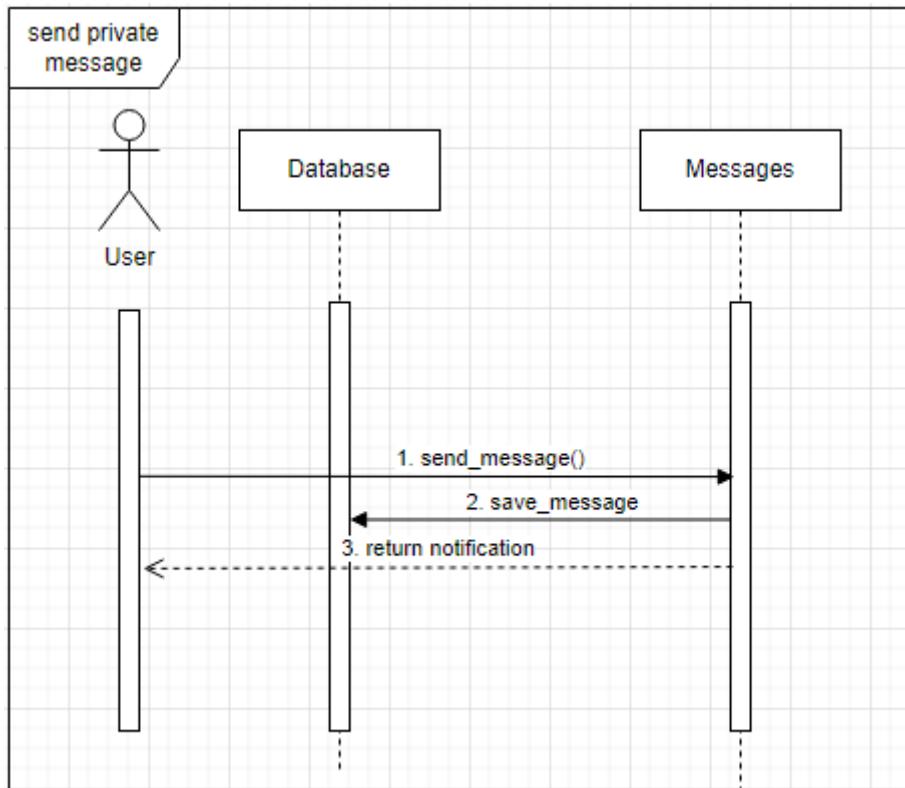
The sequence diagram for adding friend



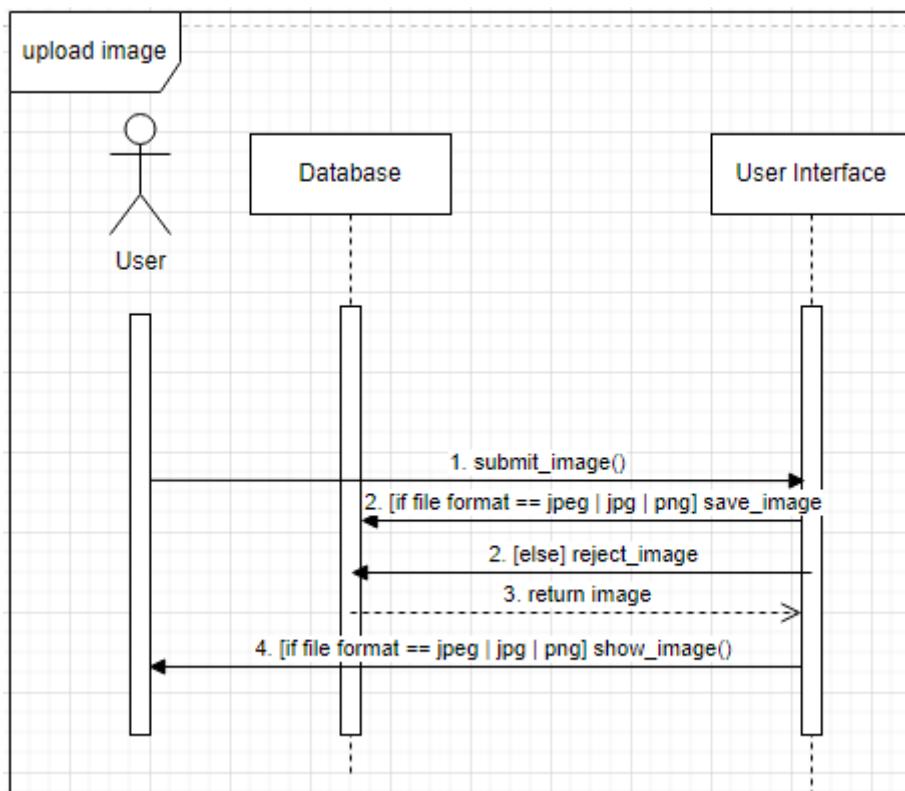
The sequence diagram for loading tweets



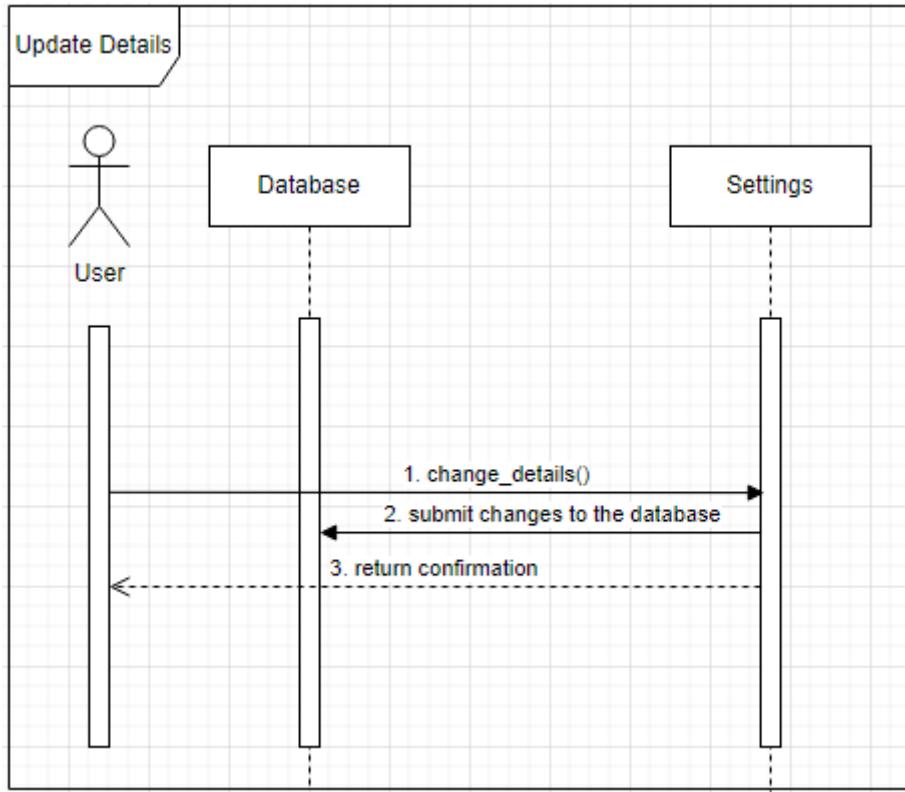
The sequence diagram for getting notifications.



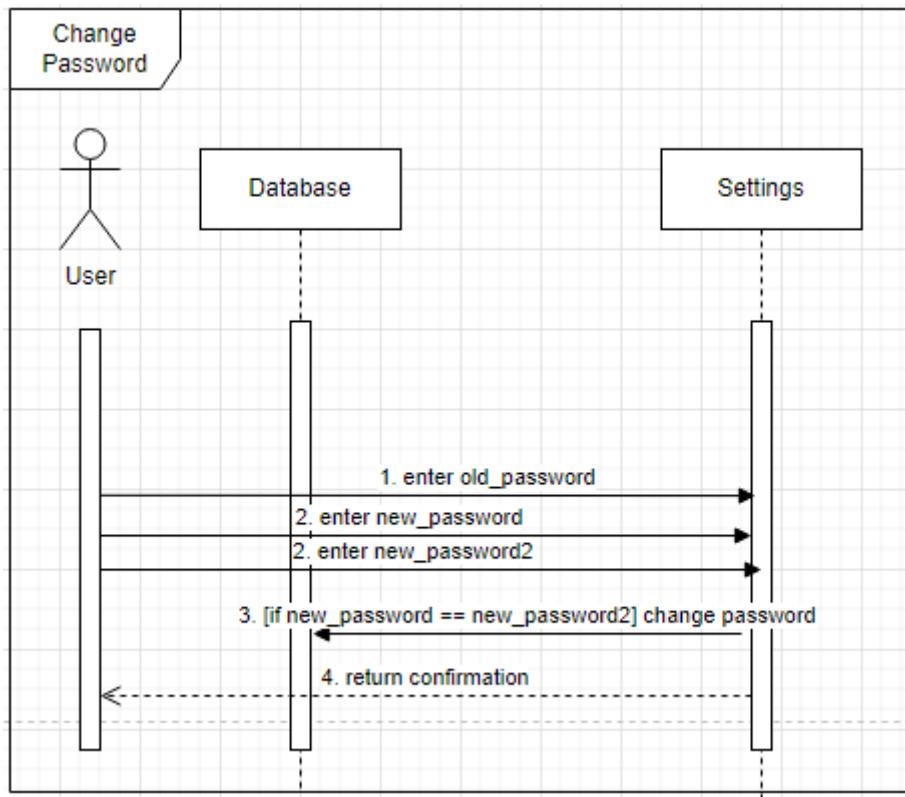
The sequence diagram for sending private messages



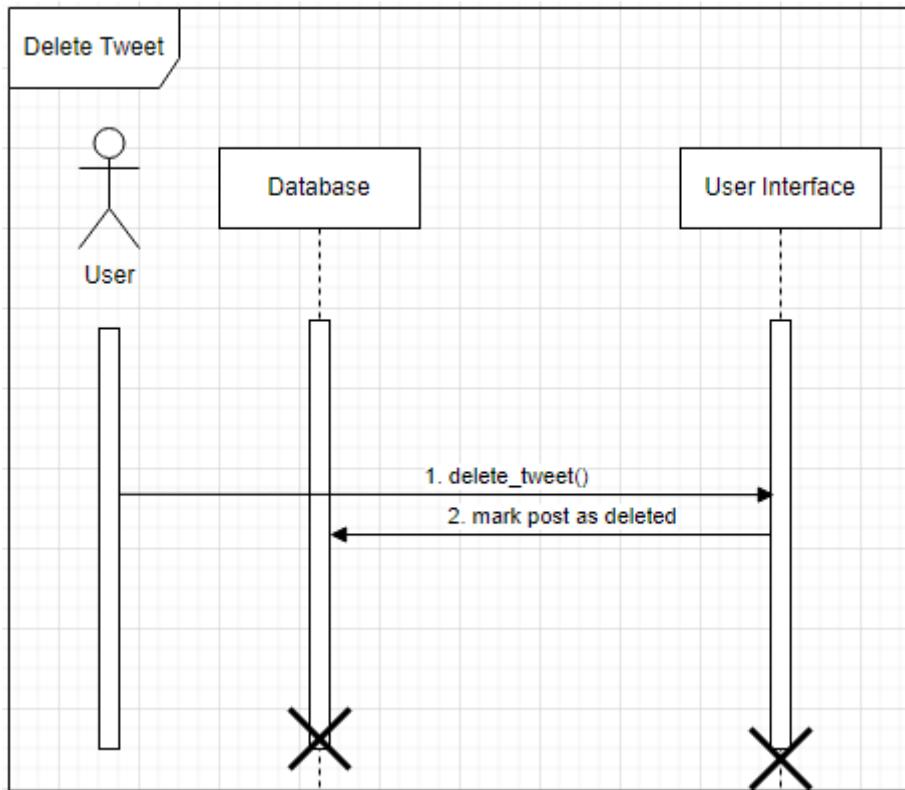
The sequence diagram for uploading an image



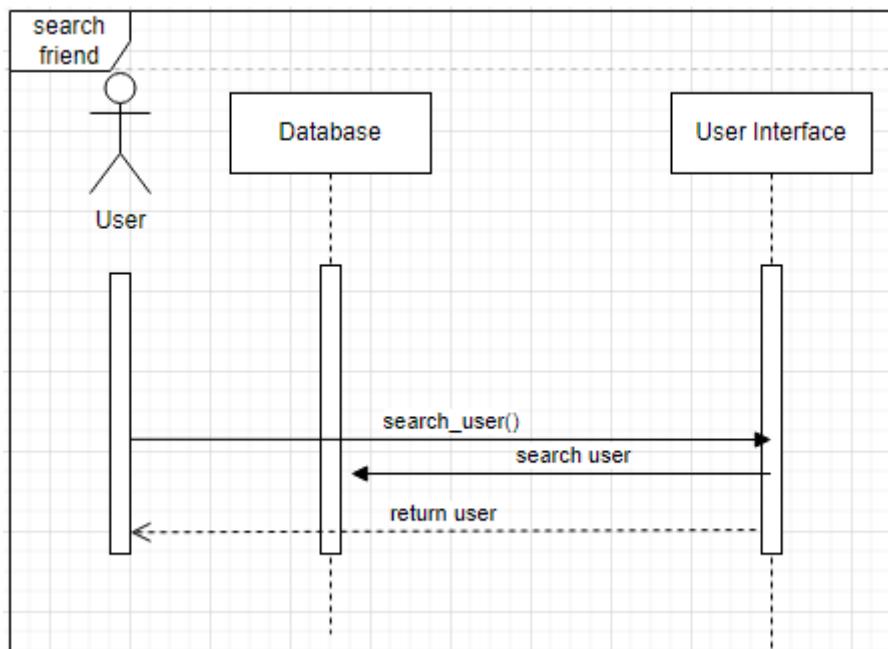
The sequence diagram for updating details



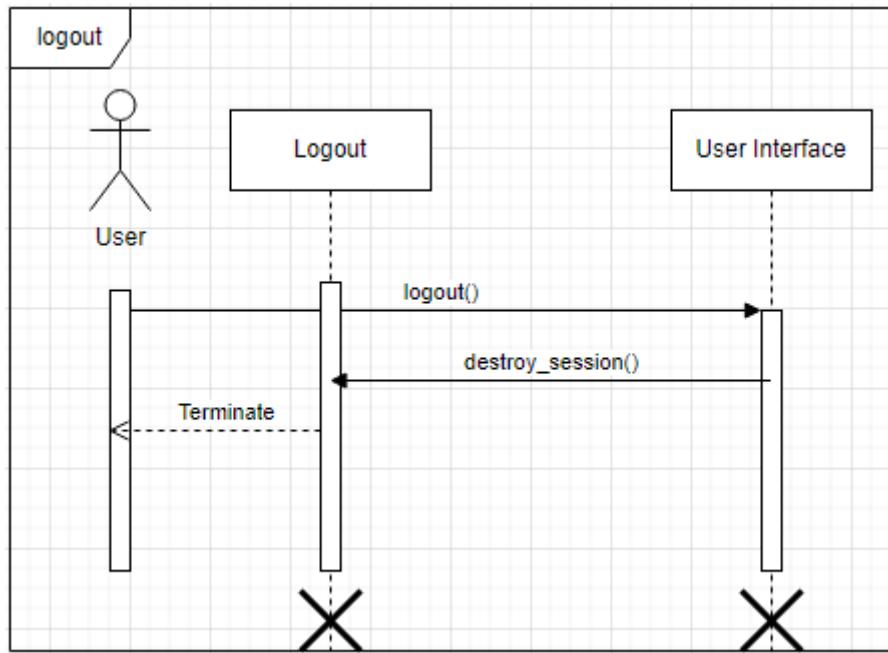
The sequence diagram for changing password



The sequence diagram for deleting tweet



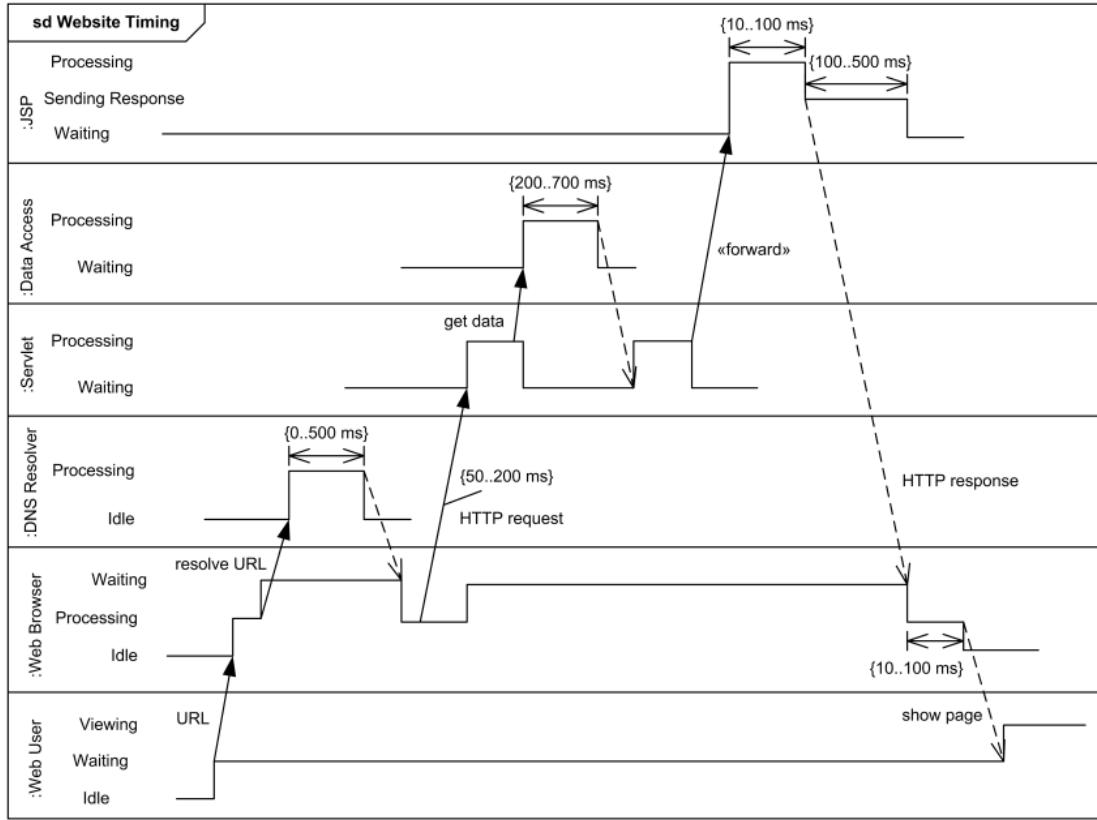
The sequence diagram for searching a friend



The sequence diagram for logout

Timing Diagram

This chapter of the report is dedicated to the timing diagram of the Twitter project. The timing diagram has been observed using Opera's inspect mode.



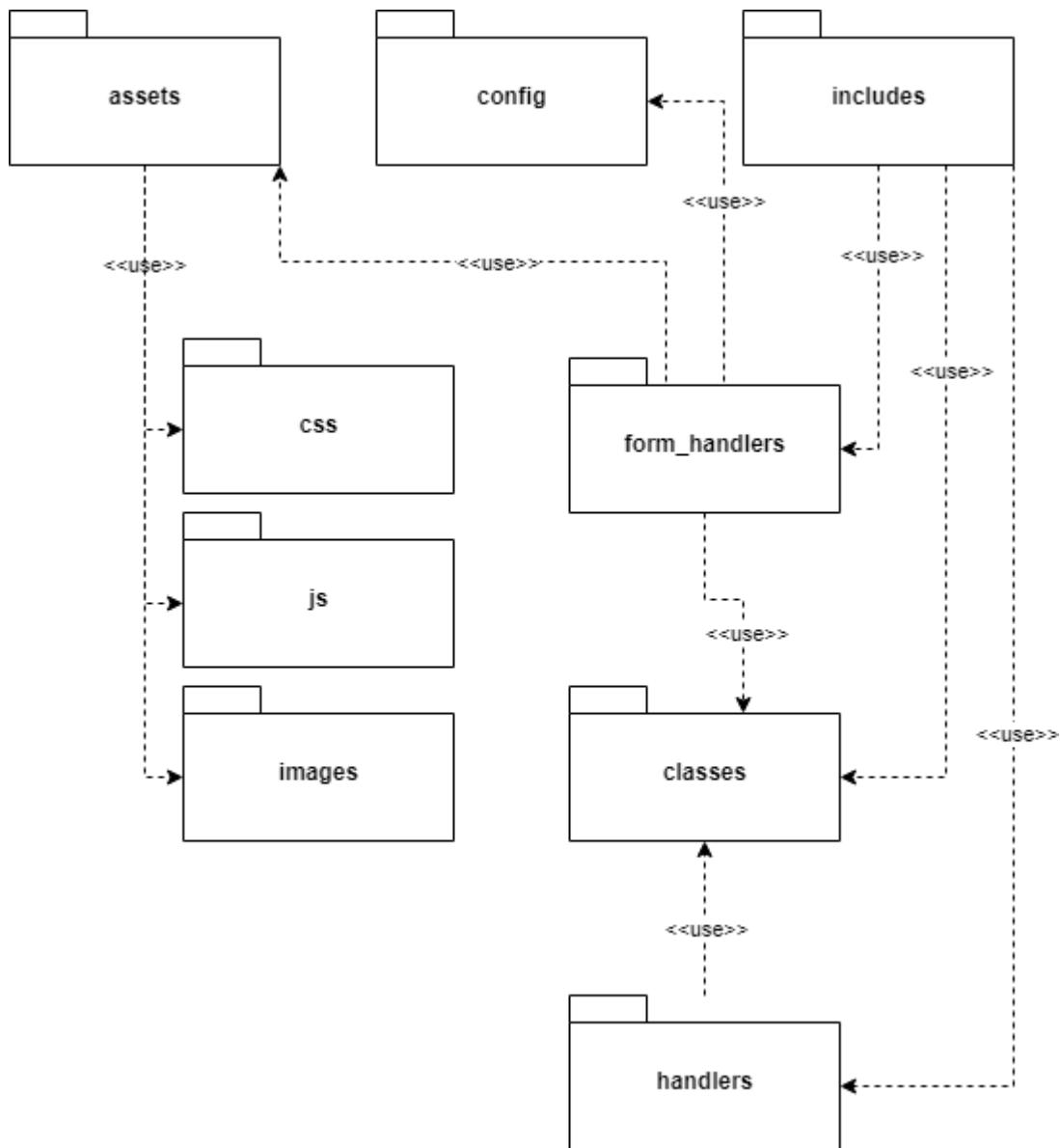
The timing diagram for the Twitter project.

c. Development View

This chapter of the report is dedicated to the development view of the project. These will include the component diagram and the package diagram for the Twitter project.

Package Diagram

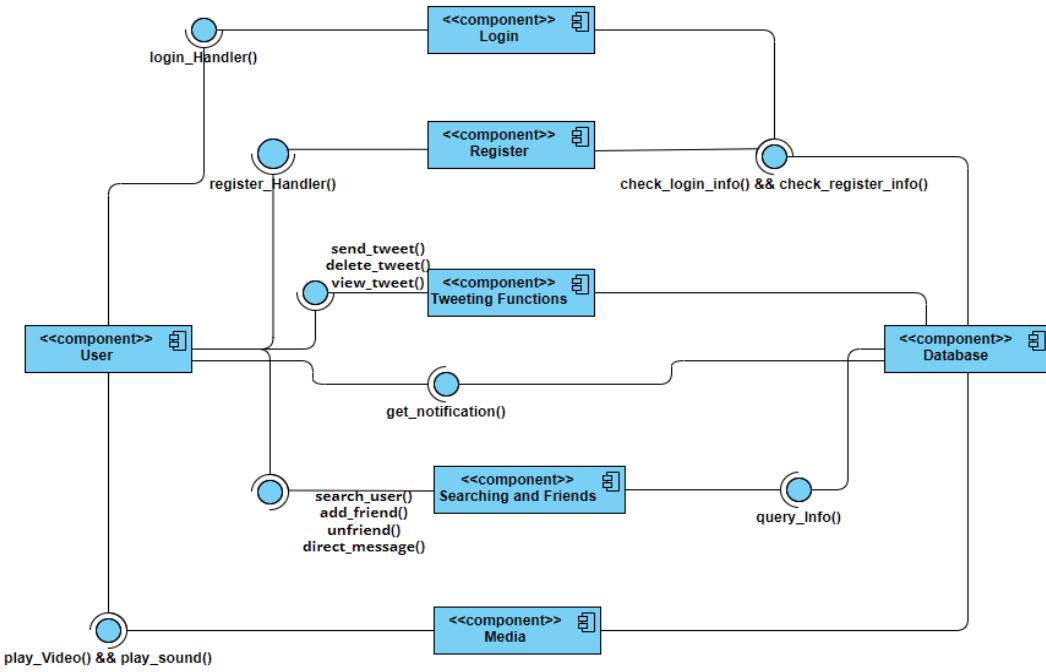
This chapter of the report explains the package diagram for the Twitter project. In the Twitter project, we have 3 main packages and these are assets, config and includes. Under the assets, we have the styling files for the project. Under the config folder, we have the config files for the project and lastly under the includes folder, we have the classes, form_handlers and handlers. Classes contains the objects for the website such as User, Post etc. Form_handlers contains the handlers about an account such as deleting a post, login, register and changing account information which is under the settings_handler.



The Package Diagram for the Twitter project.

Component Diagram

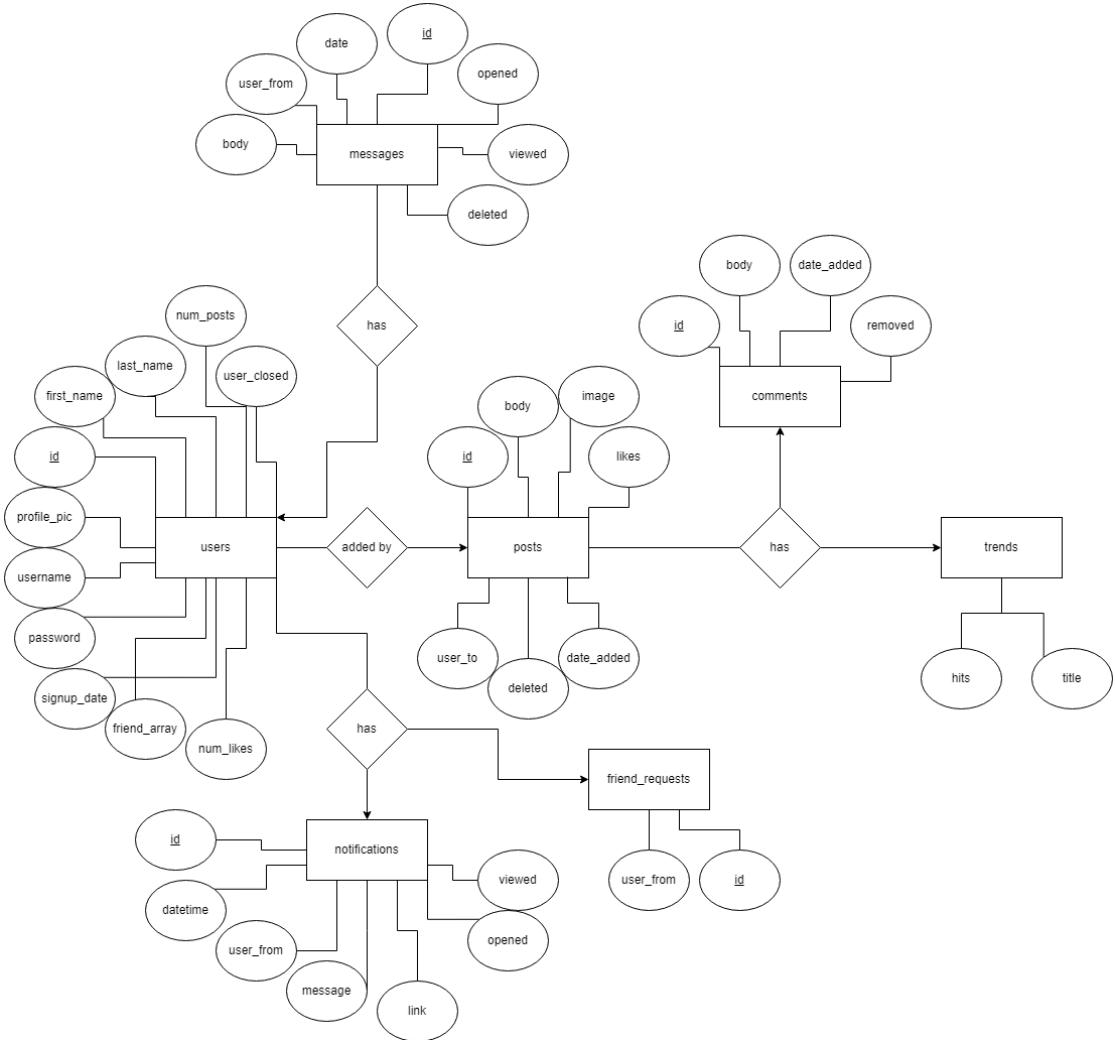
This chapter of the report explains and shows the component diagram of the Twitter project. The component diagram is self explanatory.



The component diagram of the Twitter project.

ER Diagram

This chapter of the report explains and shows the ER Diagram of the Twitter project. In the project, we have 8 entities and these are users, messages, posts, comments, trends, notifications and friend requests. In order to remove redundancy we used relations and id's to keep track of the entities and their stored values. The rest of the ER Diagram is self explanatory.

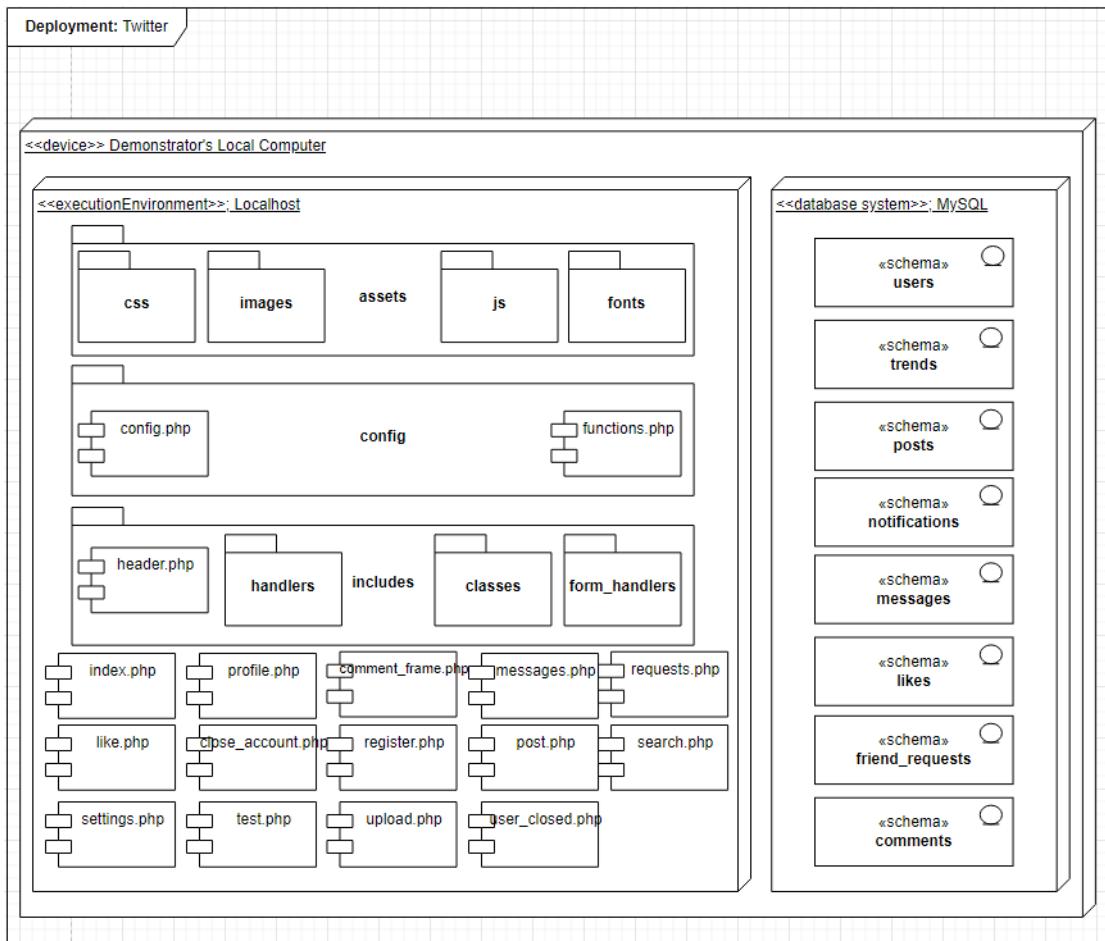


The ER Diagram of the Twitter Project.

Deployment Diagram

This chapter of the report is dedicated to the deployment diagram of the Twitter project. The device that our website will be run on is the demonstrator's local computer and both the database and the execution environment will be run at the demonstrator's local computer. Under the assets folder, we deployed the css, js files for our website's front end. Fonts contain the website's fonts for the project and the images contain the default profile pictures for the Twitter project. Under the config folder, config.php allows us to generate the connection string to the database and the functions.php contains some of the most used functions for the project. Under the includes folder, the header.php can be seen. This is the header of the Twitter and under the handlers folder, we stored all of our ajax classes. Under the form_handlers folder, the php files that perform operations about registering, deleting accounts, logging to the account have been stored. Lastly, under the classes folder, we stored our website's objects which are User object, Post object, Notification object and

Message object classes. As for the database, MySQL contains 8 tables and these are users, trends, posts, notifications, messages, likes, friend_requests and comments.



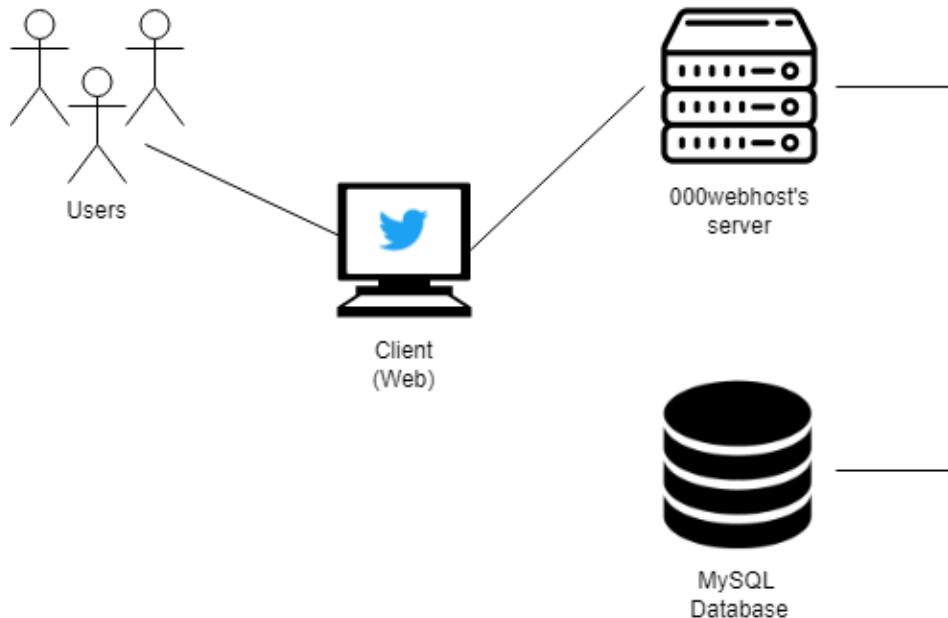
The Deployment Diagram of the Twitter project.

d. Physical View

This chapter of the report is dedicated to the physical view of the Twitter project.

Physical View Diagram

The physical view of the Twitter project as follows. Users connect to our web client. Then the web client connects to 000webhost's server which allows users to retrieve the contents of the Twitter project. Then 000webhost server communicates with our database server in order to retrieve the user data.



The physical view diagram of the Twitter project.

5. Restrictions, Limitations, Constraints

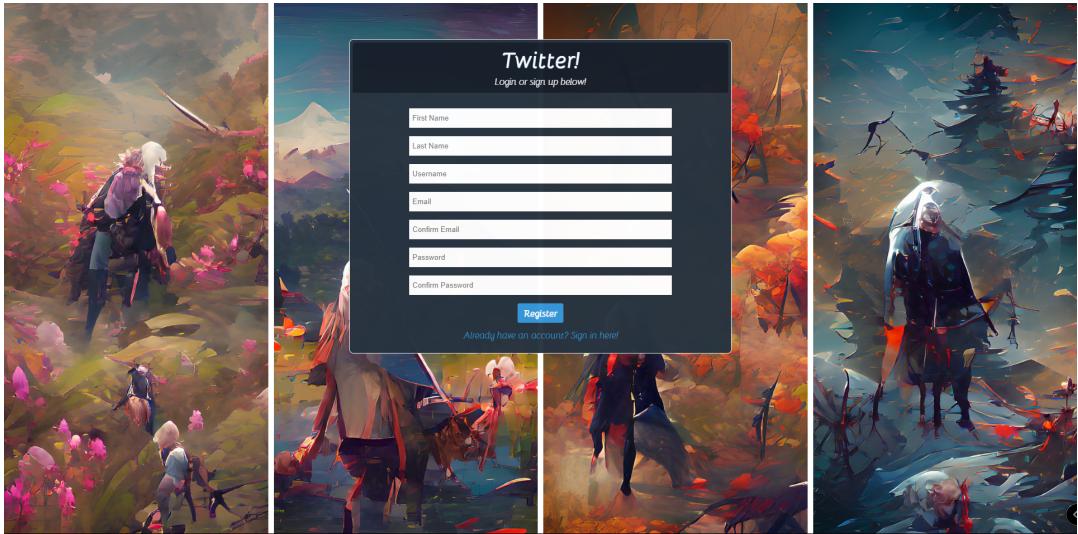
Due to the time constraints and limited budget for the project, some features such as changing user name, implementing security features that cover SQL injections and hosting the Twitter project on dedicated servers couldn't be implemented.

- Database is stored in phpMyAdmin through PHP models
- CSS, Bootstrap 5 and Javascript are used for styling and animation.
- HTML5 is used for the pages.
- Server-side is coded with PHP.

6. User Manual

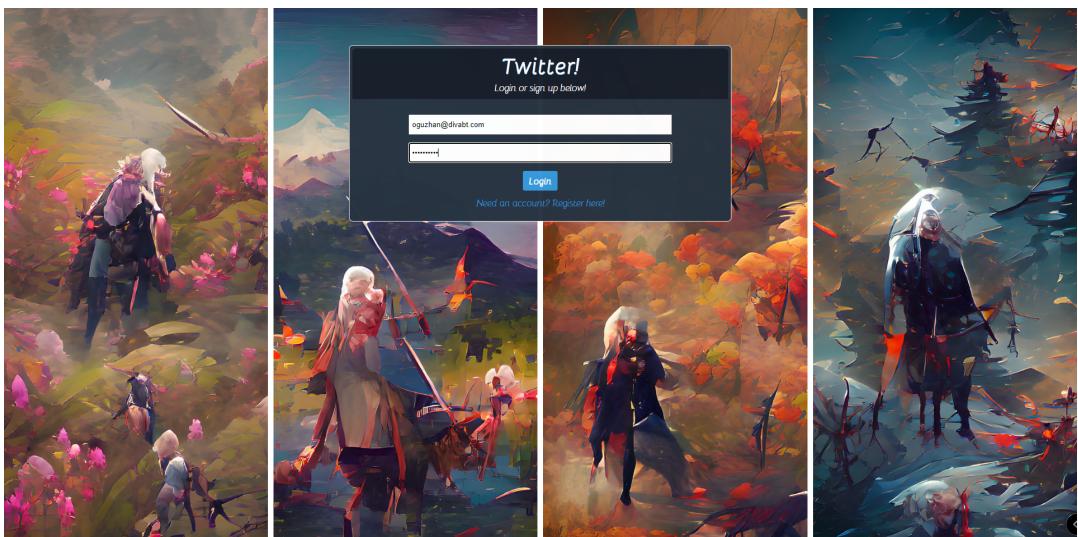
In this chapter, the user manual to operate the Twitter project website can be found.

1st step: User opens the Twitter website. If they don't have an account, they register on Twitter. If they have an account, they proceed with the login process.



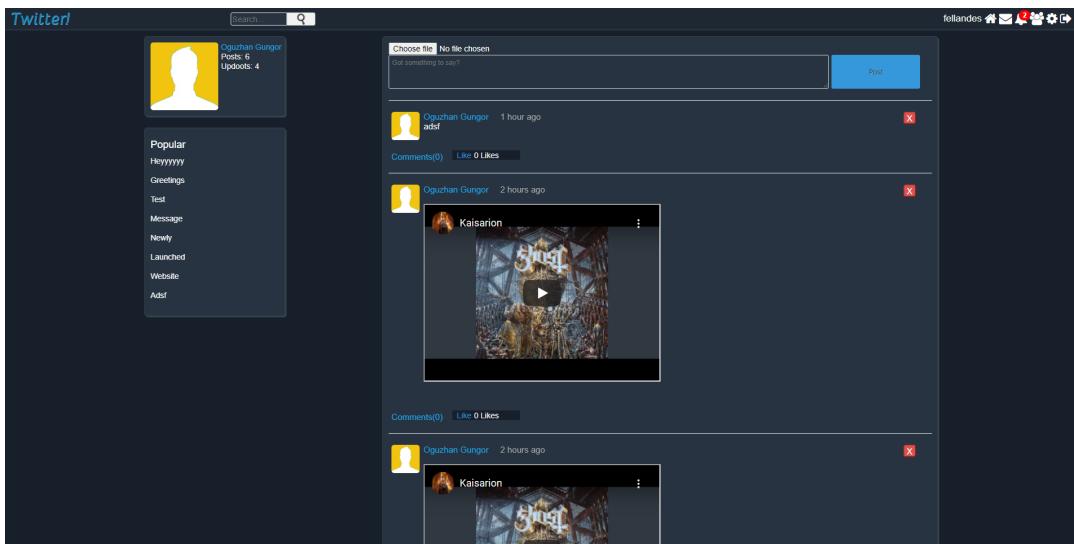
The register page for the Twitter project.

2nd step: The user logs in with their account.



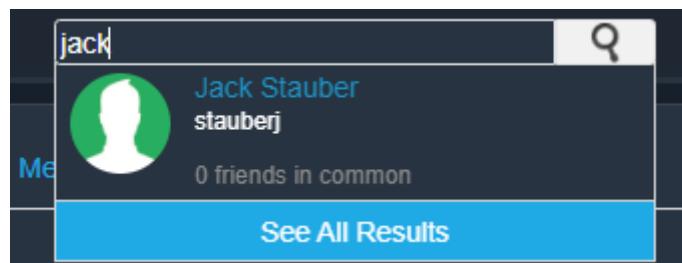
The login page for the Twitter project.

3rd step: User enters the home page.

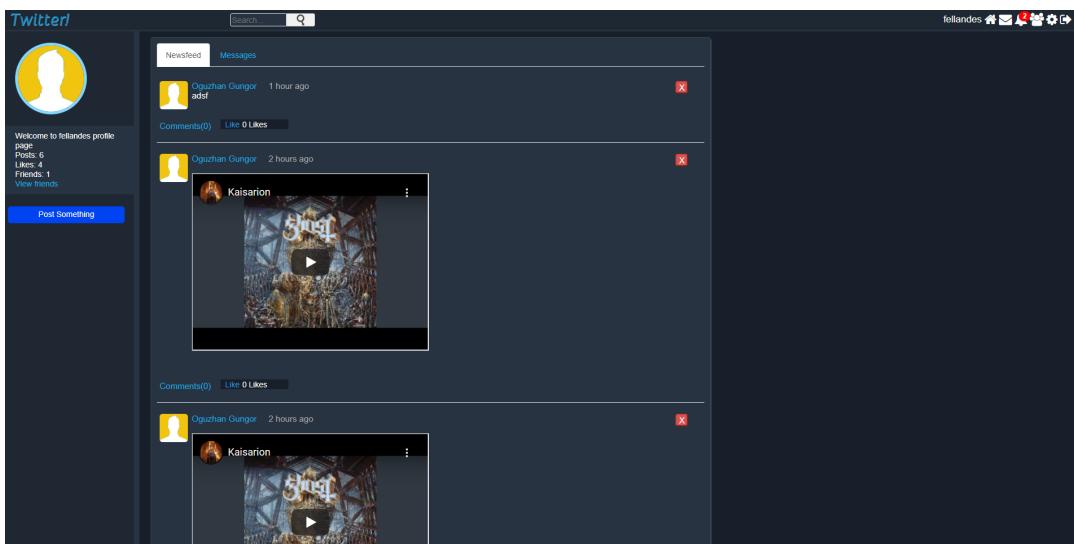


The home page for the Twitter project.

4th step: User adds their friend by searching their name.

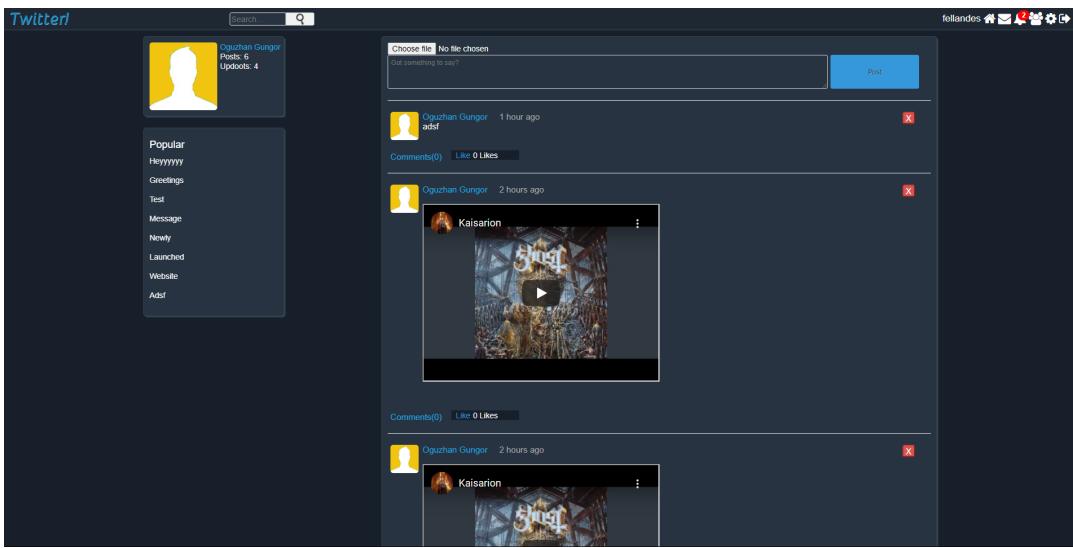


The profile searching friend by their name.



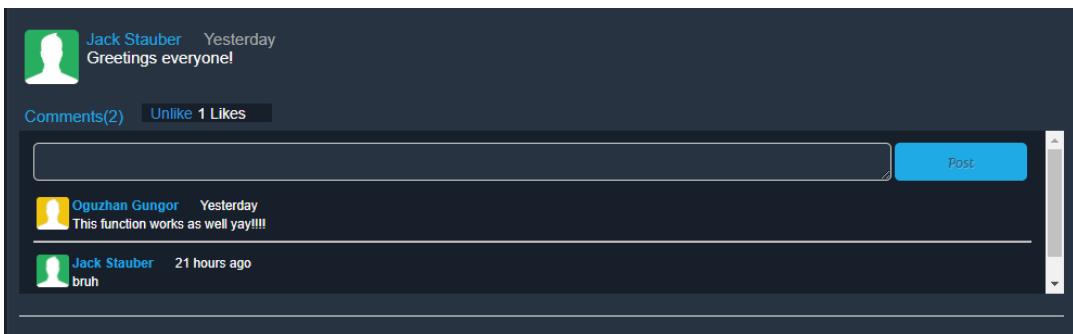
The profile page and Add friend button for adding friends.

5th step: If a user adds friends, they can see their posts on their front page.



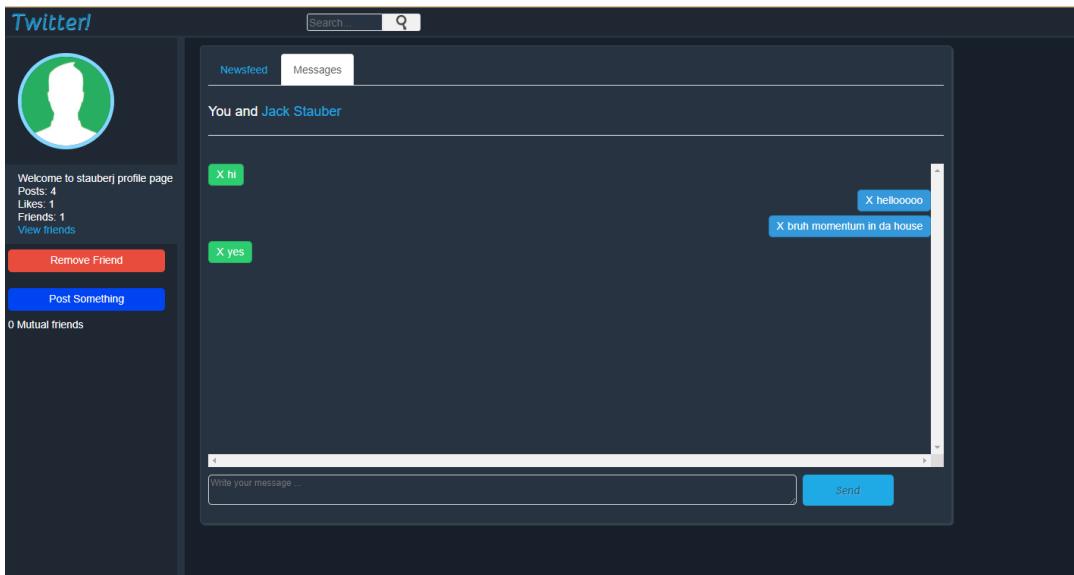
Front page after adding friends.

6th step: User can comment on posts



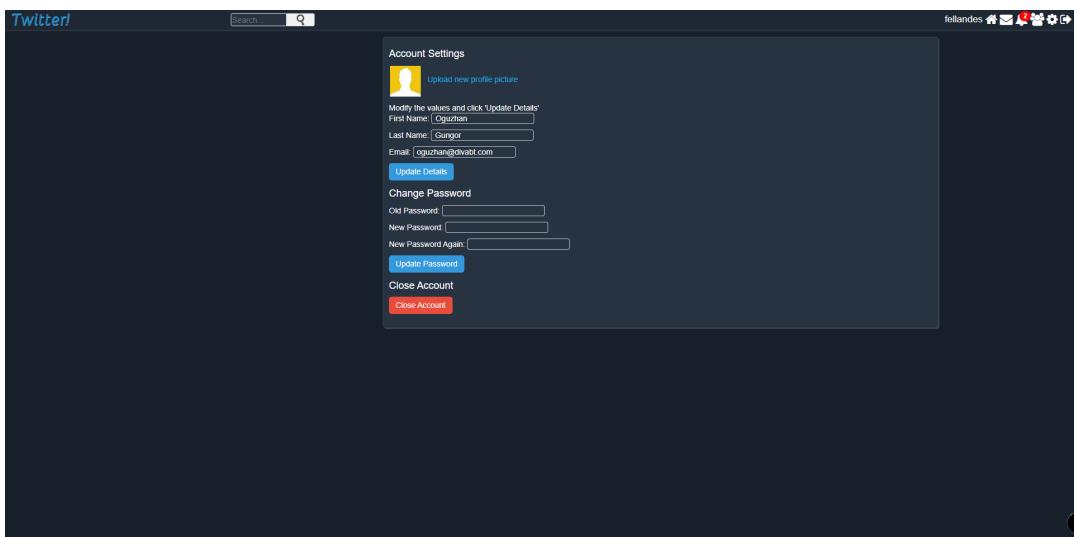
Commenting under a post

7th step: Users can message other users.



The messaging screen of the Twitter project.

8th step: User can delete or change information about their account



The settings page for the user's account

Lastly users can see notifications on the right side of the page.



The notifications can be seen on the right corner of the page.

7. Conclusion

This term project helped us to understand problems when designing a program and how to overcome those problems. This project helped us how to plan as a team, how to work as a team. These skills that we've obtained from this project taught us how to approach a real life problem, and how to overcome those problems in a limited time as well as this project taught us how to divide tasks and conquer those tasks as a team.