

MEET ALL

Ilayda Kurt – 2021719099

03.01.2023

SWE 573 - Fall 2022 – Term Project

A note-taking and post-sharing application

GitHub Repository: <https://github.com/ilaydakurtboun/swe573>

Git Tag Version: v0.9-latest

Git Tag Link: <https://github.com/ilaydakurtboun/swe573/releases/tag/v0.9-latest>

Deployment (Project) URI: <http://44.201.174.233:8000/>

HONOR CODE

Related to the submission of all the project deliverables for the Swe573 2022 Fall semester project reported in this report, I Ilayda KURT declare that:

- I am a student in the Software Engineering MS program at Bogazici University and am registered for Swe573 course during the 2022 Fall 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.

Ilayda KURT



Content

1. Important Information & Sample Users.....	4
1.1. “.env” file.....	4
1.2. User for Admin Panel.....	4
1.3. Test User in Production.....	4
1.4. Other Information.....	4
2. Declaration: “All work in the Project is performed myself.”.....	5
3. Project Details.....	5
3.1. Overview.....	5
3.2. Software Requirements Specification.....	5
3.2.1. Introduction.....	5
3.2.1.1. The Purpose.....	5
3.2.1.2. The Scope.....	6
3.2.1.3. Definitions.....	6
3.2.2. Overall Description.....	6
3.2.3. System Features and Requirements.....	6
3.2.3.1. Functional Requirements.....	6
3.2.3.2. System Requirements.....	7
3.3. Design & Mockups.....	7
3.3.1. Beginning Stage Designs and Mockups.....	8
Register Screen:.....	8
Login Screen:.....	8
Main Page:.....	9
Own Page:.....	9
3.3.2. Progress Stage Designs and Mockups.....	10
Register Screen:.....	10
Login Screen:.....	10
Main Page:.....	11
3.3.3. Final Stage Designs and Images.....	11
Main Screen without Authentication:.....	11
Register Screen:.....	12
Login Screen.....	12
Main Screen with Authentication.....	13
3.4. Diagrams.....	13
3.4.1. UML Diagram.....	13
3.4.2. Sequence Diagram.....	14
3.5. Requirement Completion Table.....	15

<u>3.6. Deployment Information.....</u>	<u>16</u>
<u>3.6.1. Dockerization.....</u>	<u>16</u>
<u>3.6.2. Deployment.....</u>	<u>16</u>
<u>3.7. System Manual.....</u>	<u>17</u>
<u>3.8. User Manual.....</u>	<u>17</u>
<u>3.9. Test Results.....</u>	<u>18</u>
<u>3.9.1. Unit Test Results.....</u>	<u>18</u>
<u>User Login Unit Tests.....</u>	<u>18</u>
<u>Space, Label, Post Unit Tests.....</u>	<u>19</u>
<u>3.9.2. User Test Results.....</u>	<u>19</u>
<u>3.10. Demo Video Link & Information.....</u>	<u>20</u>
<u>4. References.....</u>	<u>20</u>

1. Important Information & Sample Users

1.1. “.env” file

In this project, I use a .env file to keep secret information about the project such as database user credentials, email hosts, domain urls, etc.

To run this project at local, you need to create .env file in the main directory of project, and copy below information:

```
SECRET_KEY=5ti__g^03$@8#_$_3yd!*r&fyo01tj5$d!&zkv$+-4r+of8fgnd
EMAIL_HOST=smtp.gmail.com
EMAIL_HOST_USER=swe573fall2022@gmail.com
EMAIL_HOST_PASSWORD=bhtv nxhm xedl jhmo
EMAIL_BACKEND=django.core.mail.backends.smtp.EmailBackend
DB_HOST=db
DB_NAME=app
DB_USER=ilayda
DB_PASS=123123
POSTGRES_DB=app
POSTGRES_USER=ilayda
POSTGRES_PASSWORD=123123
DOMAIN_URL=http://127.0.0.1:8000
```

1.2. User for Admin Panel

In this project, I implemented a Django Admin Panel. To use the admin user in production, there is a need for user credentials. Here is the admin credentials:

Admin Panel Username: [“admin@admin.com”](mailto:admin@admin.com)

Admin Panel Password: “admin1.”

1.3. Test User in Production

Test Username/Email: [“ilaydaakurt@gmail.com”](mailto:ilaydaakurt@gmail.com)

Test Password: “1234qwer”

1.4. Other Information

When login with an admin user to the Django Admin Panel, if there is an authorization session of a user in the application side, it gets confused. There might be a bug. I will work on it. During tests, the Django Admin Panel and Application part should not test at the same time because of a confusing authorization session.

2. Declaration: “All work in the Project is performed myself.”

I graduated from Boğaziçi University, Department of Mathematics. At the beginning of my 3rd year in my undergraduate period, I started developing software as an intern in a startup company. I learned Python and Django. I have been developing with Python and Django for 3,5 years in my intern and full-time working life. Django Admin Panel configuration, dockerization process, deployment and use of cloud service provider applications, which I learned through my experiences, and all the developments I have made in this project belong to me. Except for the python libraries I used and the Figma application that I got help for design, I wrote all the codes with the knowledge I gained from my own experience.

3. Project Details

The aim of this project is to gather the posts that people see in different applications, that they are interested in, that they want to read later or that they want to share with their friends under a single platform.

3.1. Overview

In this application, the user sees the titles of posts without registering and logging in, however she cannot like or add comments. If the user wants to do this, the user should register to the application with her email address, name and surname. After logging in, she sees a list of titles where posts on similar topics are shared. If she wishes, she can create a title and start sharing posts under it. On the other hand, she can read the posts in the content of existing titles, bookmark them, like them, write comments, and create a new post. Also, she can visit the profiles of other users in the application, review their posts and topics, and follow that user if he/she wishes. In addition, she can search for a topic, a tag, a word within the application and a list of related titles or posts will appear.

3.2. Software Requirements Specification

3.2.1. Introduction

The aim of this part is to make the system and expectations clear and understandable. The following sections, the scope of projects, the categorized requirements, and the features of the project.

3.2.1.1. The Purpose

The purpose of this part is to define the project and to give information about the user interface and software of the project. With this section, the requirements of the product becomes more understandable and clear, since what users can expect, what users can do, and how the software and hardware behave will be explained.

3.2.1.2. The Scope

The aim in this project is to create an user-friendly environment that users can share any post from other platforms. Also there should be liking, bookmarking, following people, commenting on posts mechanisms. The stakeholders might be students, teaching assistants, and the professors. To develop this product, the specific requirements about the system, functionality, and interfaces will be explained.

3.2.1.3. Definitions

Software: A combined product with Backend, Frontend and Database.

User Interface: A website

Users: Person who use the website

Admin: Person who controls the website

Label: A word or group of words that specify the topic in general

Post: A piece of writing about a topic. It contains image, link, title, description, label, platform where it is taken

Space: A generic topic and description of a group of posts

3.2.2. Overall Description

In this project, our users are able to share any content from other applications with the platform information that content is taken from. There should be liking, commenting, bookmarking and following people features to make the product more user-friendly. There is a problem about the other post-sharing application. They do not bring content from many platforms together. With this project, we aim to solve this problem.

3.2.3. System Features and Requirements

3.2.3.1. Functional Requirements

- When the user clicks "Register", the software should open the form of "Register".
- If the user wants to create an account with an existing email, the software shall give the existing account error.
- If the user's passwords do not match each other in Register Form, the software shall give the do not matching passwords error.
- When the user clicks "Login", the software should open the form of "Login".
- If the user wants to login with non existing email, the software shall give the non existing account email.
- If the user enters a wrong password, the software should give incorrect credentials error.
- When the user registers, the software should redirect the user to the login page.
- When the user logs in, the software shall redirect the user to the main page.

- When the user clicks the "Spaces" tab, the software shall show the recent topics for posts.
- When the user clicks a topic from Spaces screen, the software shall show the topic-related posts.
- When the user wants to create a new space, the software shall save the space that user creates.
- When the user wants to add a post to a space, the software shall save the post in that space.
- When the user clicks the "Recent Post" tab, the software shall show the recent posts from any space.
- When the user likes a post and clicks the "Like "button, the software shall add the post to the "Your List" section.
- When the user wants to add some comment on a post and clicks the button "Comment", the software shall save his comment.
- When the user clicks the "Your List" tab, the software shall show the liked posts.
- When the user clicks the "Your Posts" tab, the software shall show his posts.
- When the user wants to create a post, the software shall get the input from the user.
- When the user clicks "Private" in the create post section, the software shall make the post visible only to the owner user.
- When the user clicks the "Browse Media" button, the software shall open the browser media screen.
- When the user adds a link, the software shall make it url.
- When the user adds a label in the create post section, the software shall assign the post to the label.
- When the user clicks the button "Submit" in the create post section, the software shall create a new post.
- When the user wants to find a word related to posts and writes the word to the search bar, the software shall show the related results.
- When the user wants to add a non-existent label to a post, the software shall make the create label available.

3.2.3.2. System Requirements

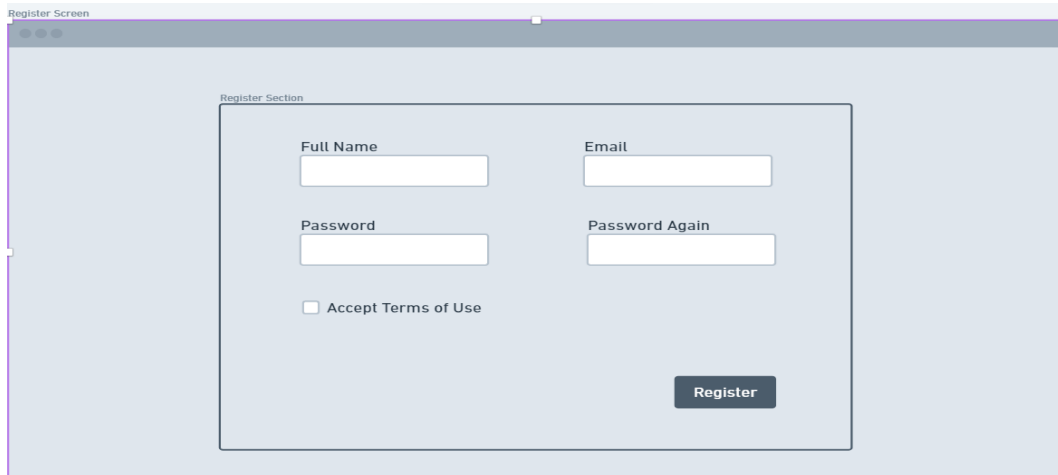
- The system should accept requests.
- The backup of the database is recreated for 3 times of a month.
- The response time of the API requests should not exceed 10 seconds.

3.3. Design & Mockups

In the beginning of the project, I prepared some mockup designs and screens with unreal information, for example I used "John Doe" as a user, and "Lorem insup..." as content. However, while developing the project, I realize that they are not user-friendly. So I try to change them.

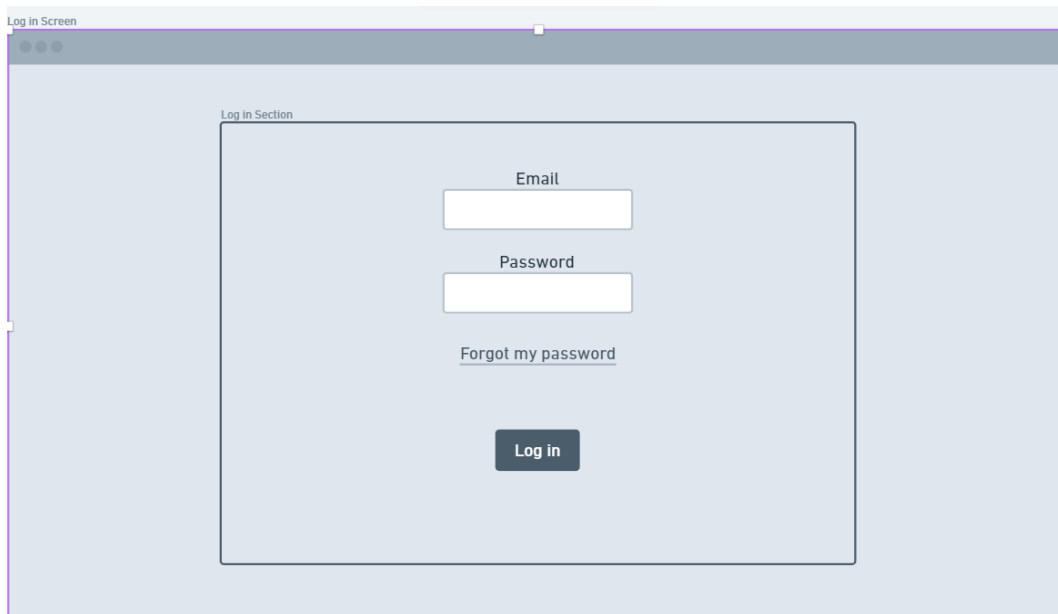
3.3.1. Beginning Stage Designs and Mockups

Register Screen:



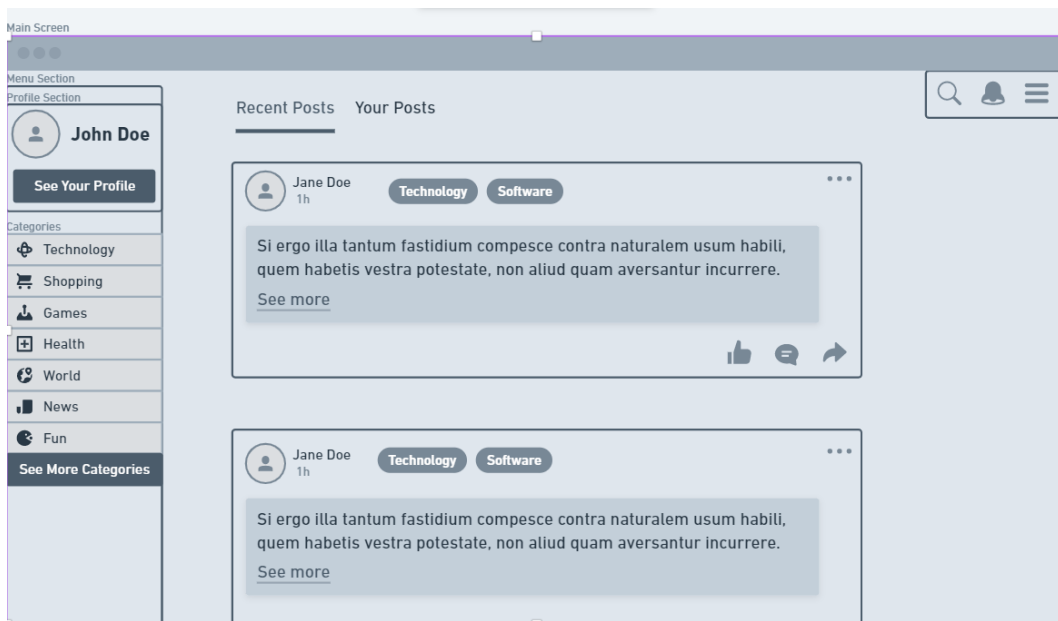
The Register Screen mockup features a light blue background. At the top left, the text "Register Screen" is displayed. Below it, a "Register Section" is enclosed in a rounded rectangle. Inside this section, there are four input fields: "Full Name", "Email", "Password", and "Password Again", arranged in a 2x2 grid. Below the "Password" field, there is a checkbox labeled "Accept Terms of Use". At the bottom right of the "Register Section", there is a dark blue button labeled "Register".

Login Screen:

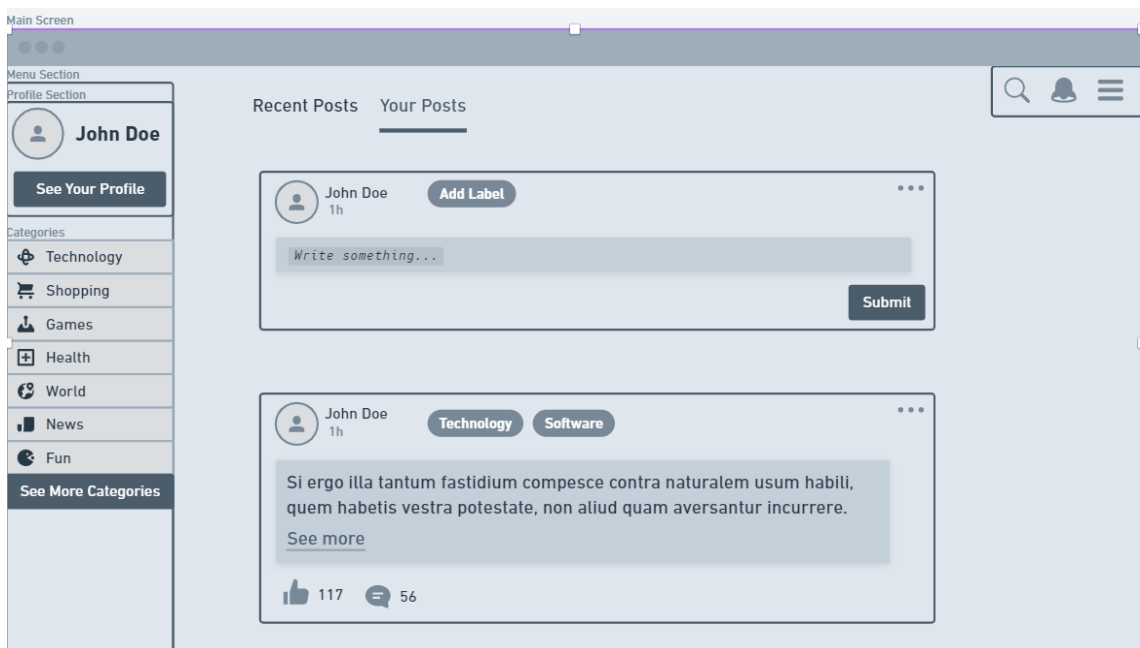


The Login Screen mockup features a light blue background. At the top left, the text "Log In Screen" is displayed. Below it, a "Log in Section" is enclosed in a rounded rectangle. Inside this section, there are two input fields: "Email" and "Password", arranged vertically. Below the "Password" field, there is a link labeled "Forgot my password". At the bottom center of the "Log in Section", there is a dark blue button labeled "Log in".

Main Page:



Own Page:



3.3.2. Progress Stage Designs and Mockups

Register Screen:

WELCOME TO MEET ALL

Register

Login

First Name

Last Name

Email

Password

Password - 2

Submit

Login Screen:

WELCOME TO MEET ALL

Register

Login


Email

Password

Submit

Main Page:

MEET ALL

 Ilayda Kurt

Main Page

Menu

See Your Profile


Notifications

SpacesRecent PostsYour ListYour Posts


Is the question "what is your salary expectation" becoming a thing of the past?

Submit

What's happening in Boğaziçi University?

 1M+

Layoffs on the rise in Europe

 1M+

3.3.3. Final Stage Designs and Images

Main Screen without Authentication:

WELCOME TO MEET ALL

Register

Login

Why iPhone 14 pro is sold out in every EU countries?

There is no stock for iPhone 14 pro.

View Posts

South Indian Cooking

If you are a vegetarian and like spicy food you will love southern Indian food. Come join us.

View Posts

Register Screen:

WELCOME TO MEET ALL

Register

Login

First Name

Last Name

Email

Password

Password

Submit

Login Screen

WELCOME TO MEET ALL

Register

Login

Email

Password

Forgot Your Password?

Submit

Main Screen with Authentication

MEET ALL

İlayda Kurt

Your Profile

Following

Followers

Possible to Know

Your Bookmarks

Logout

Spaces

Recent Posts

Your Spaces

Your Posts

Liked Posts

Search

Title

Description

Submit

South Indian Cooking

If you are a vegetarian and like spicy food you will love southern Indian food. Come join us.

View Posts

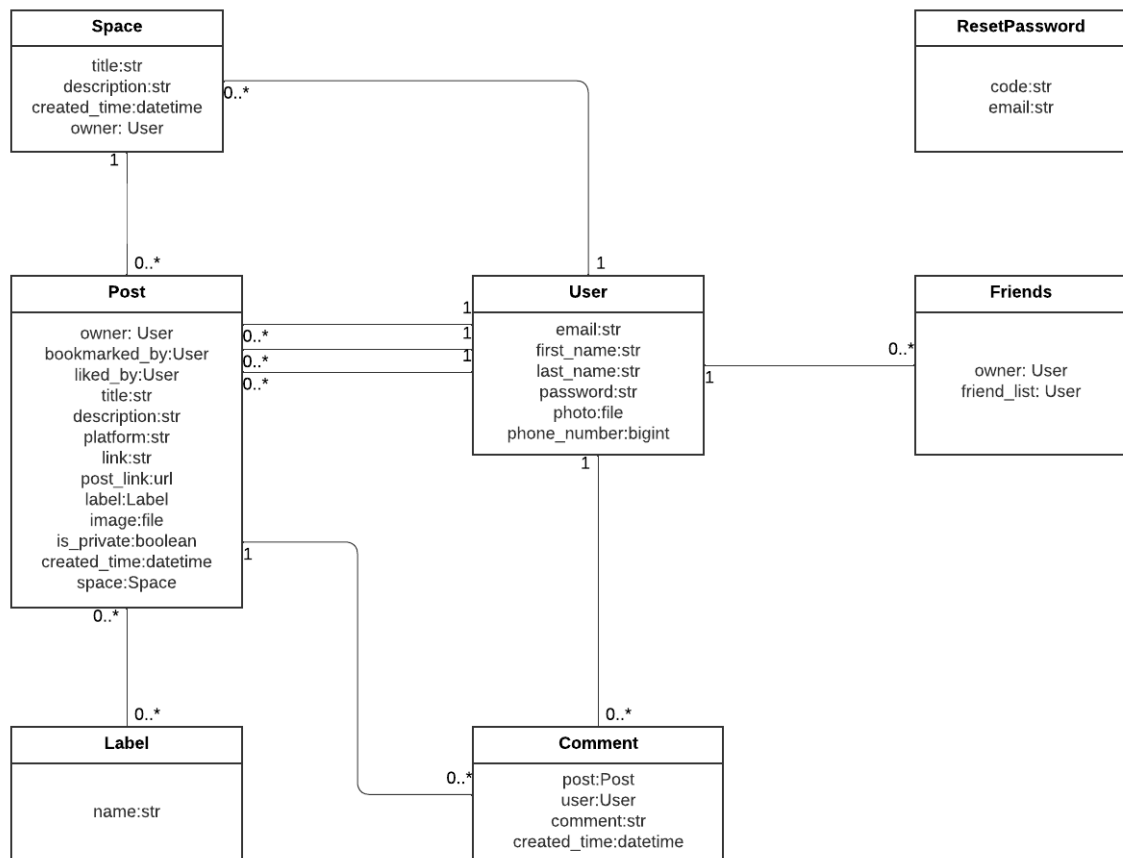
Why iPhone 14 pro is sold out in every EU contries?

There is no stock for iPhone 14 pro.

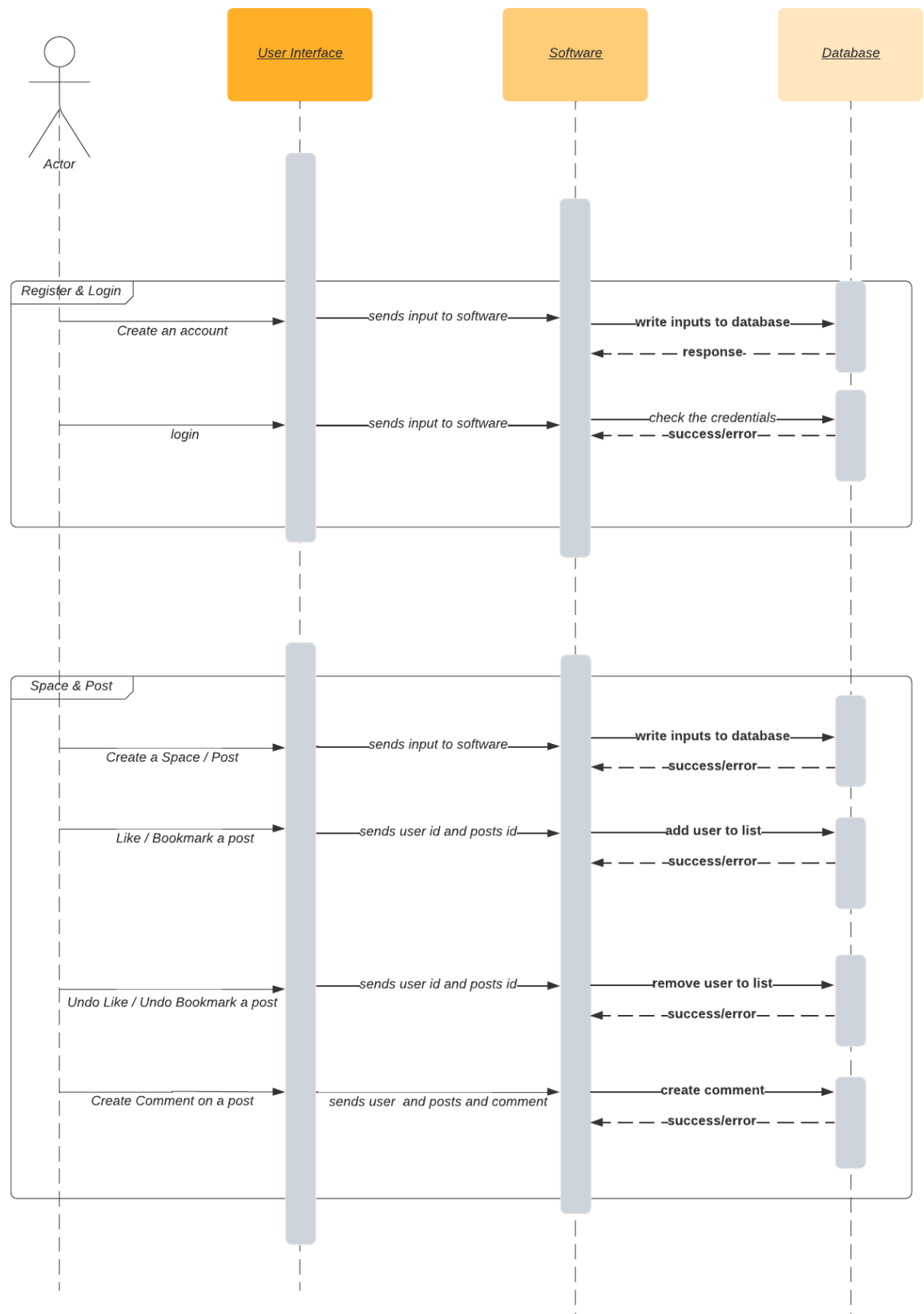
View Posts

3.4. Diagrams

3.4.1. UML Diagram



3.4.2. Sequence Diagram



3.5. Requirement Completion Table

When the user clicks "Register", the software should open the form of "Register".	Completed
If the user wants to create an account with existing email, the software shall give the existing account error.	Completed
If the user's passwords do not match each other in Register Form, the software shall give the do not matching passwords error.	Completed
When the user clicks "Login", the software should open the form of "Login".	Completed
If the user wants to login with nonexisting email, the software shall give the nonexisting account email.	Completed
If the user enters a wrong password, the software should give incorrect credentials error.	Completed
When the user registers, the software should redirect the user to login page.	Completed
When the user logs in, the software shall redirect the user to main page.	Completed
When the user clicks the "Spaces" tab, the software shall show the recent topics for posts.	Completed
When the user clicks a topic from Spaces screen, the software shall show the topic-related posts.	Completed
When the user wants to create a new space, the software shall save the space that user creates.	Completed
When the user wants to add a post to a space, the software shall save the post in that space.	Completed
When the user clicks the "Recent Post" tab, the software shall show the recent posts from any space.	Completed
When the user wants likes a post and clicks the "Like "button, the software shall add the post to "Your List" section.	Completed
When the user wants to add some comment on a post and clicks the button "Comment",the software shall save his comment.	Completed
When the user clicks "Your List" tab, the software shall show the liked posts.	Completed
When the user clicks "Your Posts" tab, the software shall show the his posts.	Completed
When the user wants to create a post, the software shall get the input from user.	Completed
When the user clicks "Private" in create post section, the software shall makes the post visible only the owner user.	Not Completed

When the user clicks "Browse Media" button, the software shall open the browse media screen.	Completed
When the user adds a link, the software shall makes it url.	Completed
When the user adds a label in create post section, the software shall assign the post to the label.	Completed
When the user clicks the button "Submit" in create post section, the software shall create a new post.	Completed
When the user wants to find a word related posts and writes the word to search bar, the software shall show the related results.	Completed
When the user wants to add non-existent label to a post, the software shall makes the create label available.	Not Completed
The system should accept requests.	Completed
The backup of database is recreated for 3 times of a month.	Not Completed
The response time of the API requests should not exceed 10 seconds.	Completed
The registration of the platform should be verified by email verification.	Not Completed
Email or phone number that used for registration should be unique.	Not Completed

3.6. Deployment Information

3.6.1. Dockerization

This project is dockerized. It works as two containers in Docker:

1. API Container: It uses the "python:3.8-alpine" version and runs on port 8000.

2. Database Container: "postgres:12-alpine" version is used as a database and it works on port 5432.

Related Dockerfile link:

<https://github.com/ilaydakurtboun/swe573/blob/main/Dockerfile>

Related docker-compose.yml file link:

<https://github.com/ilaydakurtboun/swe573/blob/main/docker-compose.yml>

3.6.2. Deployment

This project was deployed through Amazon Web Services cloud service provider. EC2 - Micro instance was rented on AWS and the project was cloned into that machine from github. Docker and docker-compose were installed on the machine and deployed with docker on the machine.

3.7. System Manual

To run the project on a local machine with docker, firstly make sure that docker is already installed. After you started the docker and be sure about docker is running then you can simply run these commands:

docker-compose build

then,

docker-compose up

the first time you start the project.

docker-compose up will create a postgresql database and migrate all migrations files for the first time. After that only docker-compose up will be sufficient unless any database or library change occurs. If this is the case you should also run **docker-compose build** before **docker-compose up**. Sometimes build does not install libraries that you need if this is the case, you need to run docker-compose down first then build then up.

In this case the application will be served in <http://0.0.0.0:8000/>

To use the Django Admin Panel, you should enter the backend container. To do this:

- Run **docker ps** command to see the container id of the api service.
- Then you can copy the api service container id and run the **docker exec -it <container_id> sh** command to get in the container.
- After this, you should run **python manage.py createsuperuser** command and give input for your admin user.
- Now you can access the admin panel at <http://0.0.0.0:8000/admin/> with your admin credentials just you created.

To reach the swagger documentation of the project, you should visit:

<http://0.0.0.0:8000/swagger/>

You can also find some of these information in README.md file of the project:

<https://github.com/ilaydakurtboun/swe573/blob/main/README.md>

3.8. User Manual

- Visit the <http://44.201.174.233:8000/> anonymously.
- See the list of Spaces (topics and their descriptions).
- Click the “View Posts” button and list the posts of the selected space.
- If interested, click the “Register” button and enter your information, then click the “Submit” button.
- Then it redirects the login page, enter your credentials.
- If you forget the credentials, click the “Forgot your credentials?” button and enter registered email. Then write the email and code that application sends to you, and reset your password.
- If successfully logged in, it lists the spaces. Click the “View Posts” button and list the posts of the selected space. If interested, click “Like”, “Bookmark” or “See Details” buttons.

- When you click the “See Details” button and if you are the owner of the posts, you can edit or delete the post.
- If you want to add a comment, click the “Comment” and write your comment, then submit it.
- If interested, add new posts to the selected space or create new space and create posts in it.
- To see spaces that you created, click the “Your Spaces” button.
- To see posts that you created, click the “Your Posts” button.
- To see posts that you liked, click the “Liked Posts” button.
- If you want to undo your likes, click the “Undo Like” button.
- To see posts that you bookmarked, click the “Your Bookmarks” button.
- If you want to undo your bookmarks, click the “Undo Bookmark” button.
- To see recent posts without space restriction, click the “Recent Posts” button.
- To discover other people in the application, click the “Possible to Know” button. If interested, click the “View Profile” button to see their profiles, spaces and posts.
- If interested, click the “Follow” button.
- To see people that you follow, click the “Followers” button.
- To see people that follow you, click the “Following” button.
- To edit your personal information or change password, click the “Your Profile” button.
- To search something, click the “Search” button, and select your search area (the default is both space and post).

3.9. Test Results

3.9.1. Unit Test Results

User Login Unit Tests

Test 1: Login User with Correct Credentials by Django Authenticate Function

Test 2: Login User with Wrong Username by Django Authenticate Function

Test 3: Login User with Wrong Password by Django Authenticate Function

Test 4: Login User with Correct Credentials by Request

Test 5: Login User with Wrong Password by Request

All tests are passed:

```

Creating test database for alias 'default'...
System check identified some issues:

WARNINGS:
feed.Post.bookmarked_by: (fields.W340) null has no effect on ManyToManyField.
feed.Post.label: (fields.W340) null has no effect on ManyToManyField.
feed.Post.liked_by: (fields.W340) null has no effect on ManyToManyField.
user.Friends.friend_list: (fields.W340) null has no effect on ManyToManyField.

System check identified 4 issues (0 silenced).
.response <HttpResponse status_code=200, "text/html; charset=utf-8">
....
-----
Ran 5 tests in 1.458s

OK
Destroying test database for alias 'default'...

```

Space, Label, Post Unit Tests

Test 1: Label Create Test

Test 2: Space Name Test

Test 3: Space Owner Assignment Test

Test 4: Post Name Test

Test 5: Post Owner Assignment Test

Test 6: Post Liked By Assignment Test

All tests are passed:

```
Creating test database for alias 'default'...
System check identified some issues:

WARNINGS:
feed.Post.bookmarked_by: (fields.W340) null has no effect on ManyToManyField.
feed.Post.label: (fields.W340) null has no effect on ManyToManyField.
feed.Post.liked_by: (fields.W340) null has no effect on ManyToManyField.
user.Friends.friend_list: (fields.W340) null has no effect on ManyToManyField.

System check identified 4 issues (0 silenced).
.....
-----
Ran 6 tests in 1.154s

OK
Destroying test database for alias 'default'...
```

3.9.2. User Test Results

Register	Passed
Login	Passed
Forgot Password & Reset Password	Passed
Create Space	Passed
List Space's Posts	Passed
Create Post in a Space	Passed
Like Post	Passed
Add Post to Bookmark	Passed
Add Comment to Post	Passed
Edit Own Post	Passed
Delete Own Post	Passed
See Details of Others	Passed
Search in Posts	Passed
Search in Spaces	Passed

Search in Both Space and Posts	Passed
See Own Posts	Passed
See Own Spaces	Passed
See Own Profile	Passed
Change Password	Passed
Edit First Name	Passed
Edit Last Name	Passed
List Possible to Know People	Passed
Follow People	Passed
Unfollow People	Passed
See Following People	Passed
See Followers	Passed
See Other People Profile	Passed
Click Link of Post and Redirect to Link	Passed
Logout	Passed

3.10. Demo Video Link & Information

I prepared a demo video via Zoom Screen Recording. There might be a visual quality problem because of this.

Demo Video Link:

https://drive.google.com/drive/folders/10S9JSxhVoOcXhx8o6Djr_4HGD7dnmAIH?usp=sharing

4. References

Swagger Documentation:

<https://drf-yasg.readthedocs.io/en/stable/readme.html#openapi-3-0-note>

Admin Panel Documentation: <https://django-jazzmin.readthedocs.io/>

Model Viewset Documentation: <https://www.django-rest-framework.org/api-guide/viewsets/>

Django Unit Test Documentation and Examples:

<https://docs.djangoproject.com/en/4.1/topics/testing/overview/>

<https://developer.mozilla.org/en-US/docs/Learn/Server-side/Django/Testing>