

SWE573 Software Development Practice

2022 Fall

Project Final Deliverables

STUDENT: Dağlar Berk Erdem

COURSE: SWE573

Student ID: 2021719045

Date: 31.12.2022

Project Name: LinkMe

Git Repository: [GitHub Repo](#)

Git Tag Version: [v0.9](#)

Deployment URI: [52.211.173.214:8000](#)

Test Credentials:

- Username: daglar
- Password: daglardaglar

Admin Panel Credentials of the Website:

- Username: admin
- Password: admin

HONOR CODE

Related to the submission of all the project deliverables for the Swe573 2022 Fall semester project reported in this report, I **DAĞLAR BERK ERDEM** 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.

Dağlar Berk Erdem



Table of Contents

1. Overview	3
2. Software Requirements Specification	3
2.1.1. Functional Requirements	3
2.1.1.1. The User	3
2.1.1.2. The System	4
2.1.1.3. The bLink	5
2.1.1.4. The Space	6
2.1.1.5. The Tag	6
2.2.1. Non-Functional Requirements	6
3. Design	7
3.1. Mockups	7
3.2. UML Diagrams	9
3.2.1. UML Class Diagram	9
3.2.2. UML Use Case Diagram	9
4. Status of the project	10
5. Status of Deployment	16
6. System manual	16
7. User manual	17
8. Test Results	18
9. Functionality Video	18

1. Overview

LinkMe is a web based note-taking and posting application. Users can share, categorize contents from social media or blogs and interact with other users through contents. Moreover, users can browse through spaces or tags to discover contents in a specific area.

2. Software Requirements Specification

2.1.1. Functional Requirements

2.1.1.1. The User

The User should be able to create an account.

The User should have a unique email address.

The User should have a unique username.

The User should have a password with at least 8 characters.

The User should provide matching passwords for registration.

The User should be able to login to the application.

The User should provide valid username-password pair in order to login.

The User should have a Profile page.

The User should be able to edit its bLinks.

The User should be able to delete its bLinks within its profile page.

The User should be able to create bLinks by providing link(mandatory), caption(mandatory), tag(optional) and space(optional).

The User should be able to track number of followers in its profile page.

The User should be able to track number of followings in its profile page.

The User should be able to track number of bLinks in its profile page.

The User should be able to book a bLinks.

The User should be able to see its bio in its Profile page.

The User should be able to see its username in its Profile page.

The User should be able to change its username in settings page.

The User should be able to change its profile picture in settings page.

The User should be able to change its background picture in settings page.

The User should be able to change its bio in settings page.

The User should be able to change its email in settings page.

The User should be able to delete its account in settings page.

The User should be able to logout.

The User should be able to go to Search page through navbar.

The User should be able to go to Feed page through navbar.

The User should be able to go to Tags page through navbar.

The User should be able to go to Spaces page through navbar.

The User should be able to go to Profile page through navbar.

The User should be able to go to Settings page through navbar.

The User should be able to search other users through search page.

The User should be able to search tags through search page.

The User should be able to search spaces through search page.

The User should be able to search tags through tags page.

The User should be able to sign up with third party applications. (Google and Facebook).

The User should be able to sign in with third party applications. (Google and Facebook)

The User should be able to sign in with email address.

The User should be able to create tags.

The User should be able to delete its' tags.

The User should be able to create spaces.

The User should be able to delete its' spaces.

2.1.1.2. The System

The System shall redirect the user to its Profile page if registration attempt is valid.

The System shall redirect the user to signup page with error message if registration attempt is invalid.

The System shall redirect the user to its Profile page if login attempt is valid.

The System shall redirect the user to signin page if login attempt is invalid.

The System shall list the bLinks with a ascending order descending order of creation date of the user under its Profile page.

The System shall redirect the User to the according Tag page when clicked to the tag of the bLink.

The System shall redirect the User to the according Space page when clicked to the space of the bLink.

The System shall list latest created tags when a user access Tags page through navbar.

The System shall list latest created spaces when a user access Spaces page through navbar.

The System shall contain a follow mechanism.

When a user browses the Feed page, the System shall list all available tweets of the users' followings with a descending order of creation date.

When user enters a tag in tags page search, the System shall list all related content available.

When user enters a space in spaces page search, the System shall list all related content available.

When a user creates a space, the System shall check whether the space name already exist.

When a user attempts to create a space that space name is already exist, the System shall warn the user.

When a user attempts to create a space that space name is already exist, the System shall redirect the user to the according spaces' page.

When a user creates a tag, the System shall check whether the tag name already exist.

When a user attempts to create a tag that tag name is already exist, the System shall warn the user.

When a user attempts to create a tag that tag name is already exist, the System shall redirect the user to the according tags' page.

When a new user registers, the system should create a Profile for user.

When a new user registers, the system should assign a default profile picture to users' Profile.

When a new user registers, the system should assign a default background picture to users' Profile.

When a new user registers, the system should assign a unique user id to users' Profile.

When a new user registers, the system should assign a default bio to users' Profile.

When a new user registers, the system should assign a joined at date to users' Profile.

When a new user registers, the system should assign an empty list of followers to users' Profile.

When a new user registers, the system should assign an empty list of followings to users' Profile.

When a new bLink is created, the system should assign a unique id to bLink.

When a new bLink is created, the system should assign the date of creation to bLink.

When a new bLink is created, the system should assign the owners' username to bLink.

2.1.1.3. The bLink

The bLink should have a unique id.

The bLink should have a date of creation.

The bLink should have a owner username.

The bLink should have a link.

The bLink should have a caption.

The bLink should contain a preview of the link if it is valid.

The bLink preview should be consist of title, description and image.

The bLink should contain an embedded button in preview which redirects the user to the link.

The bLink should contain the number of bookmarks.

The bLink should contain a bookmark button which is black if bookmarked and white otherwise.

The bLink should contain at most 1 space name of the bLink if assigned.

The bLink should contain the tag names of the bLink if assigned.

2.1.1.4. The Space

The Space should have unique name.

2.1.1.5. The Tag

The Tag should have a unique name.

2.2.1. Non-Functional Requirements

The App should be in English.

The App shall be developed with Python v3.8.

The App shall be developed by Django framework.

The frontend of the App should be developed with Bootstrap framework.

The App shall have a PostgreSQL database.

The App shall have a cold storage database.

The App should be containerized.

The Database should be containerized.

The App and Database should be orchestrated with docker-compose.

The App should be deployed to a VM.

The App shall be hosted on AWS.

The App should be deployed using a continuous integration/continuous delivery (CI/CD) pipeline to streamline the development and deployment process.

The software should have docstrings.

The Docstrings should have written in Epytext format.

The code should be modular and scalable.

3. Design

3.1. Mockups

Contextual Mockups are of the application can be accessed from [here](#). Also shown below.

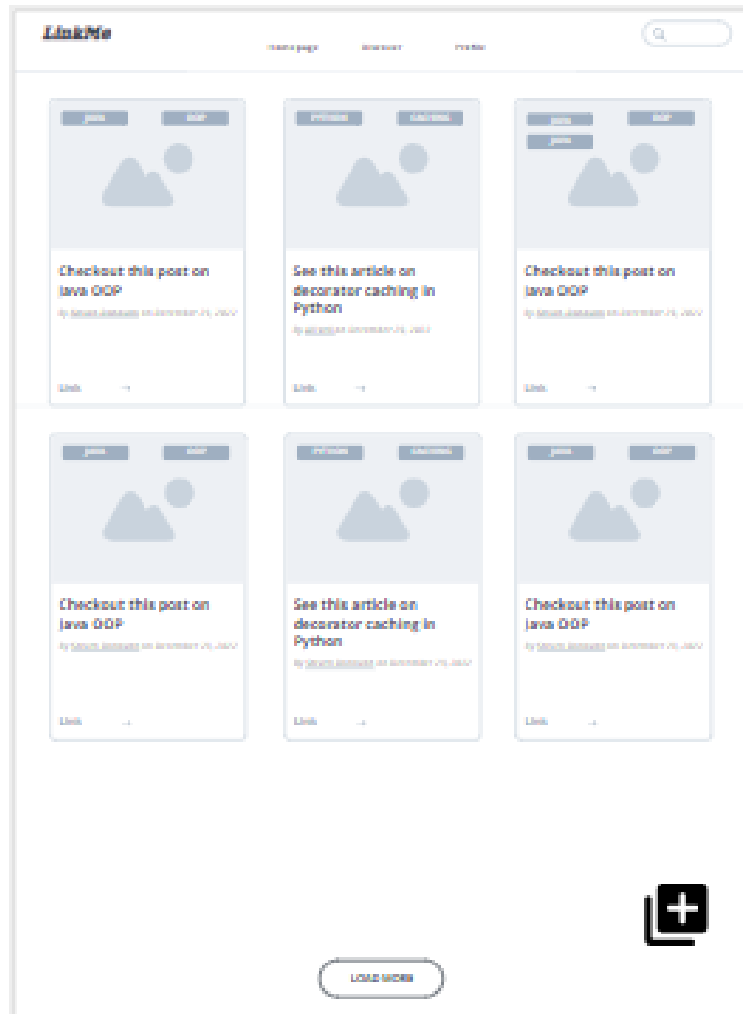


Figure 1 Feed Page Mockup

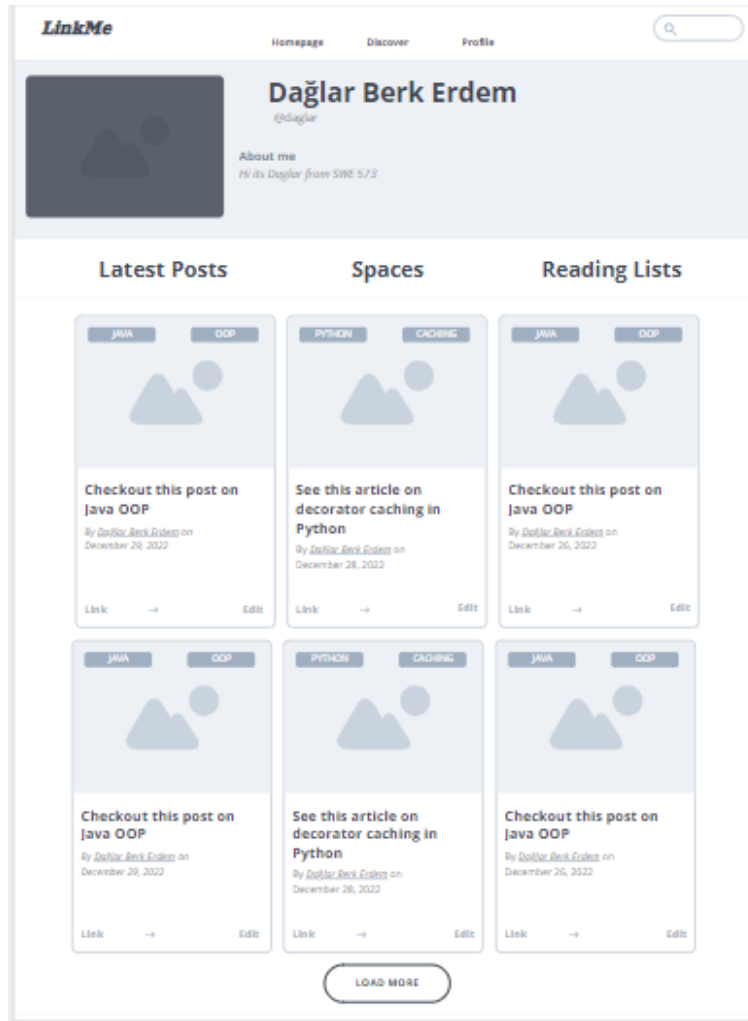


Figure 2 Profile Page Mockup

3.2. UML Diagrams

3.2.1. UML Class Diagram

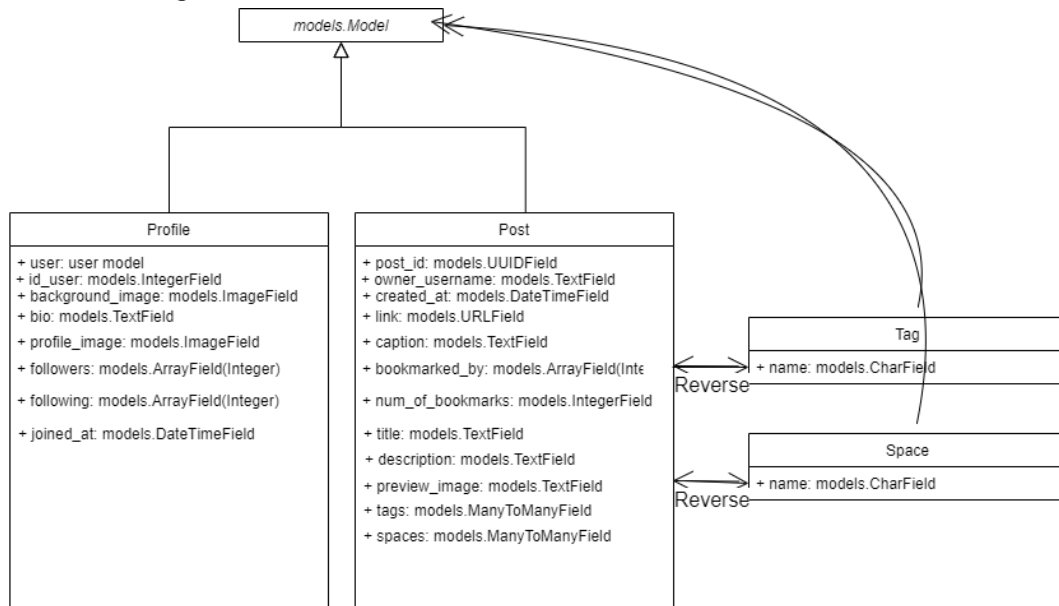


Figure 3 UML Class Diagram

3.2.2. UML Use Case Diagram

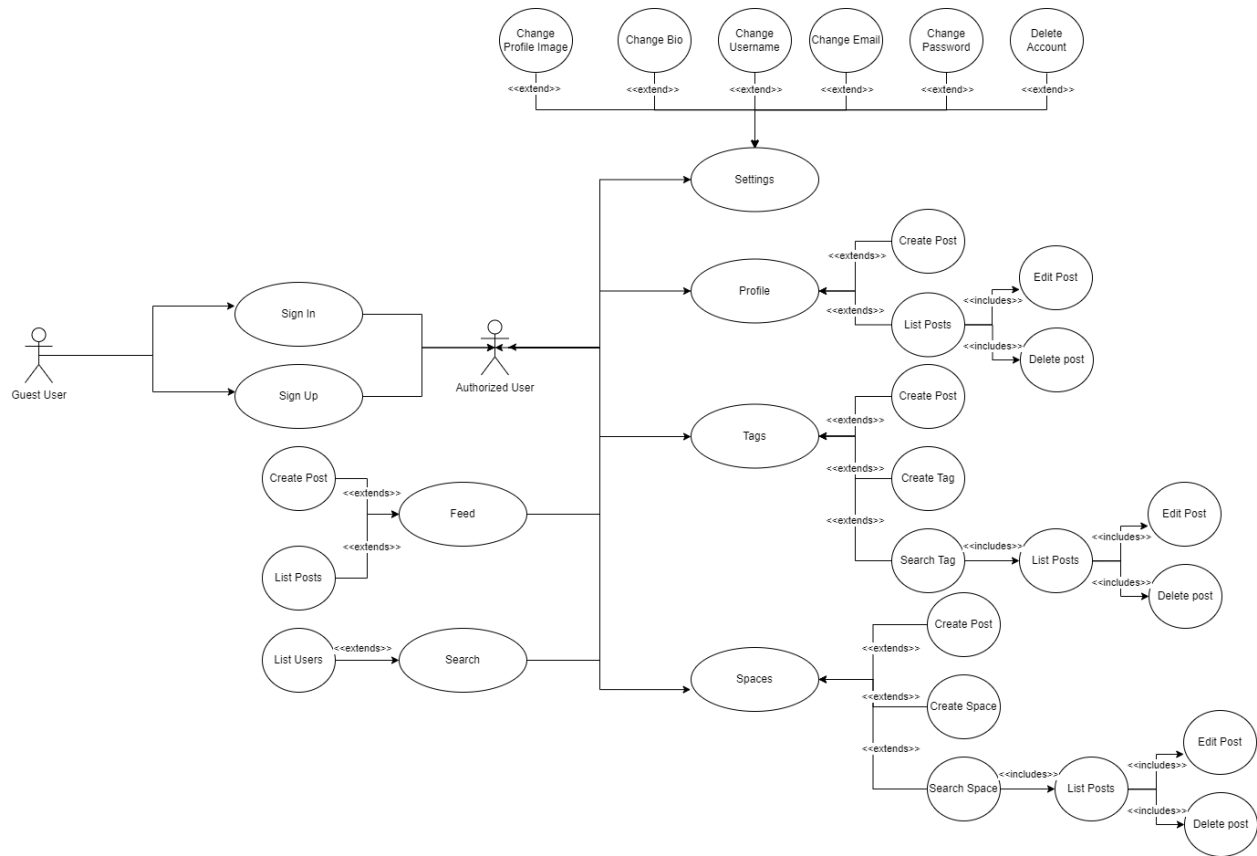


Figure 4 UML Use Case Diagram

4. Status of the project

In this section, status of requirements in section 2 is described and the corresponding GitHub Issue is linked.

#	Requirement	Status	Notes
1	The User should be able to create an account.	Completed	Issue: Create a sign-up page
2	The User should have a unique email address.	Completed	Issue: Create a sign-up page
3	The User should have a unique username.	Completed	Issue: Create a sign-up page
4	The User should have a password with at least 8 characters.	Completed	Issue: Create a sign-up page
5	The User should provide matching passwords for registration.	Completed	Issue: Create a sign-up page
6	The User should be able to login to the application.	Completed	Issue: Create a sign-in page

7	The User should provide valid username-password pair in order to login.	Completed	Issue: Create a sign-in page
8	The User should have a Profile page.	Completed	Issue: Create Profile Model
9	The User should be able to edit its bLinks.	Completed	Issue: Create Post Model
10	The User should be able to delete its bLinks within its profile page.	Completed	Issue: Create Post Model
11	The User should be able to create bLinks.	Completed	Issue: Create Post Model
12	The User should be able to track number of followers in its profile page.	Completed	Issue: Create Profile Model
13	The User should be able to track number of followings in its profile page.	Completed	Issue: Create Profile Model
14	The User should be able to track number of bLinks in its profile page.	Completed	Issue: Create Profile Model
15	The User should be able to book a bLinks.	Completed	Issue: Create Post Model
16	The User should be able to see its bio in its Profile page.	Completed	Issue: Create Profile Model
17	The User should be able to see its username in its Profile page.	Completed	Issue: Create Profile Model
18	The User should be able to change its username in settings page.	Completed	Issue: Add features to change User attributes , Create Settings Page
19	The User should be able to change its profile picture in settings page.	Completed	Issue: Add features to change User attributes , Create Settings Page
20	The User should be able to change its background picture in settings page.	Not Completed	Issue: Create Settings Page
21	The User should be able to change its bio in settings page.	Completed	Issue: Add features to change User attributes , Create Settings Page
22	The User should be able to change its email in settings page.	Completed	Issue: Add features to change User attributes , Create Settings Page
23	The User should be able to delete its account in settings page.	Completed	Issue: Add features to change User attributes , Create Settings Page
24	The User should be able to logout.	Completed	Issue: -
25	The User should be able to go to Search page through navbar.	Completed	Issue: Improvements in Frontend UI
26	The User should be able to go to Feed page through navbar.	Completed	Issue: Improvements in Frontend UI

27	The User should be able to go to Tags page through navbar.	Completed	Issue: Improvements in Frontend UI
28	The User should be able to go to Spaces page through navbar.	Completed	Issue: Improvements in Frontend UI
29	The User should be able to go to Profile page through navbar.	Completed	Issue: Improvements in Frontend UI
30	The User should be able to go to Settings page through navbar.	Completed	Issue: Improvements in Frontend UI
31	The User should be able to search other users through search page.	Completed	Issue: Search Mechanism
32	The User should be able to search tags through search page.	Not Completed	-
33	The User should be able to search spaces through search page.	Not Completed	-
34	The User should be able to search tags through tags page.	Completed	Issue: Create Tags
35	The User should be able to sign up with third party applications. (Google and Facebook).	Not Completed	-
36	The User should be able to sign in with third party applications. (Google and Facebook)	Not Completed	-
37	The User should be able to sign in with email address.	Not Completed	-
38	The User should be able to create tags.	Completed	Issue: Create Tags
39	The User should be able to delete its' tags.	Not Completed	-
40	The User should be able to create spaces.	Completed	Issue: Create Spaces
41	The User should be able to delete its' spaces.	Not Completed	-
42	The System shall redirect the user to its Profile page if registration attempt is valid.	Completed	Issue: Create Signup Page
43	The System shall redirect the user to signup page with error message if registration attempt is invalid.	Completed	Issue: Create Signup Page
44	The System shall redirect the user to its Profile page if login attempt is valid.	Completed	Issue: Create Signin Page
45	The System shall redirect the user to signin page if login attempt is invalid.	Completed	Issue: Create Signin Page

46	The System shall list the bLinks with a descending order of creation date of the user under its Profile page.	Completed	Issue: Create Profile Model
47	The System shall redirect the User to the according Tag page when clicked to the tag of the bLink.	Completed	Issue: Create Post Model
48	The System shall redirect the User to the according Space page when clicked to the space of the bLink.	Completed	Issue: Create Post Model
49	The System shall list latest created tags when a user access Tags page though navbar.	Not Completed	-
50	The System shall list latest created spaces when a user access Spaces page though navbar.	Not Completed	-
51	The System shall contain a follow mechanism.	Completed	Issue: Create Follow Mechanism
52	When a user browses the Feed page, the System shall list all available tweets of the users' followings with a descending order of creation date.	Completed	Issue: Create Feed Page
53	When user enters a tag in tags page search, the System shall list all related content available.	Completed	Issue: Create Tags
54	When user enters a space in spaces page search, the System shall list all related content available.	Completed	Issue: Create Spaces
55	When a user creates a space, the System shall check whether the space name already exist.	Completed	Issue: Create Spaces
56	When a user attempts to create a space that space name is already exist, the System shall warn the user.	Not Completed	-
57	When a user attempts to create a space that space name is already exist, the System shall redirect the user to the according spaces' page.	Completed	Issue: Create Spaces
58	When a user creates a tag, the System shall check whether the tag name already exist.	Completed	Issue: Create Tags

59	When a user attempts to create a tag that tag name is already exist, the System shall warn the user.	Not Completed	-
60	When a user attempts to create a tag that tag name is already exist, the System shall redirect the user to the according tags' page.	Completed	Issue: Create Tags
61	When a new user registers, the system should create a Profile for user.	Completed	Issue: Create Profile Model
62	When a new user registers, the system should assign a default profile picture to users' Profile.	Completed	Issue: Create Profile Model
63	When a new user registers, the system should assign a default background picture to users' Profile.	Completed	Issue: Create Profile Model
64	When a new user registers, the system should assign a unique user id to users' Profile.	Completed	Issue: Create Profile Model
65	When a new user registers, the system should assign a default bio to users' Profile.	Completed	Issue: Create Profile Model
66	When a new user registers, the system should assign a joined at date to users' Profile.	Completed	Issue: Create Profile Model
67	When a new user registers, the system should assign an empty list of followers to users' Profile.	Completed	Issue: Create Profile Model
68	When a new user registers, the system should assign an empty list of followings to users' Profile.	Completed	Issue: Create Profile Model
69	When a new bLink is created, the system should assign an unique id to bLink.	Completed	Issue: Create Post Model
70	When a new bLink is created, the system should assign the date of creation to bLink.	Completed	Issue: Create Post Model
71	When a new bLink is created, the system should assign the owners' username to bLink.	Completed	Issue: Create Post Model
72	The bLink should have a unique id.	Completed	Issue: Create Post Model

73	The bLink should have a date of creation.	Completed	Issue: Create Post Model
74	The bLink should have an owner username.	Completed	Issue: Create Post Model
75	The bLink should have a link.	Completed	Issue: Create Post Model
76	The bLink should have a caption.	Completed	Issue: Create Post Model
77	The bLink should contain a preview of the link if it is valid.	Completed	Issue: Create Post Model
78	The bLink preview should consist of title, description and image.	Completed	Issue: Create Post Model
79	The bLink should contain an embedded button in preview which redirects the user to the link.	Completed	Issue: Create Post Model
80	The bLink should contain the number of bookmarks.	Completed	Issue: Create Post Model
81	The bLink should contain a bookmark button which is black if bookmarked and white otherwise.	Completed	Issue: Create Post Model
82	The bLink should contain at most 1 space name of the bLink if assigned.	Completed	Issue: Create Post Model
83	The bLink should contain the tag names of the bLink if assigned.	Completed	Issue: Create Post Model
84	The Space should have unique name.	Completed	Issue: Create Spaces
85	The Tag should have a unique name.	Completed	Issue: Create Tags
86	The App should be in English.	Completed	-
87	The App shall be developed with Python v3.8.	Completed	-
88	The App shall be developed by Django framework.	Completed	Issue: Learn About Django Framework , Set up the General Skeleton
89	The frontend of the App should be developed with Bootstrap framework.	Completed	Issue: Improvements in Frontend UI
90	The App shall have a PostgreSQL database.	Completed	Issue: Setup a PostgreSQL database
91	The App shall have a cold storage database.	Not Completed	
92	The App should be containerized.	Completed	Issue: Dockerize the app and postgres database
93	The Database should be containerized.	Completed	Issue: Dockerize the app and postgres database
94	The App and Database should be orchestrated with docker-compose.	Completed	Issue: Dockerize the app and postgres database

95	The App should be deployed to a VM.	Completed	Issue: Deploy to AWS
96	The App shall be hosted on AWS.	Completed	Issue: Deploy to AWS
97	The App should be deployed using a continuous integration/continuous delivery (CI/CD) pipeline to streamline the development and deployment process.	Not Completed	-
98	The software should have docstrings.	Completed	Issue: Generate Epytext style Docstrings
99	The Docstrings should have written in Epytext format.	Completed	Issue: Generate Epytext style Docstrings
100	The code should be modular and scalable.	Completed	Issue: Increase Modularity

5. Status of Deployment

The application is [dockerized](#) and [deployed](#), and can be accessed through <http://52.211.173.214:8000>.

The docker-compose.yml, Dockerfile and more details can be found

6. System manual

This project is built with Python 3.8. Site packages i.e., Dependencies of the project are listed in requirements.txt. Requirements.txt can be found at the root folder of the GitHub repository.

In order to setup and run this system, users need to follow step by step guide below.

1. Install Docker Desktop
2. Start Docker Desktop
3. Open a terminal and run following commands
4. git clone <Repository-URL >
5. cd SWE573_Term_Project
6. vim SWE573_Term_Project/LinkMe/.env
7. Press A and Paste Following


```
SECRET_KEY=django-insecure-aeijp&ejbr^^5w6k@8g&l)p&9#a3k=z2yo&l$0-
vbu3bv6i6i&
# FOR DOCKER-COMPOSE
DATABASE_NAME=postgres
DATABASE_USER=postgres
DATABASE_PASSWORD=postgres
DATABASE_HOST=db
DATABASE_PORT=5432
```

8. Press CTRL+C and type :wq then press ENTER
9. docker-compose up --build
10. Open your browser and go to 127.0.0.1:8000
11. Enjoy!

7. User manual

In order to start the application users must create an .env file as mentioned above. After starting the application by following the steps mentioned above, users must run the following commands to initialize spaces and tags features.

1. docker ps
2. Copy the the container id of container named swe573_term_project
3. docker exec -it <container-id> /bin/bash (this command opens the containers terminal)
4. python manage.py createsuperuser (this command creates a super user)
5. Write down your superuser name and password then hit enter
6. Write exit to return to the outer terminal
7. Go to 127.0.0.1:8000/admin and sign in with your superuser credentials
8. Create a Space and Create a Tag by hand
9. Restart the browser and go to 127.0.0.1:8000/
10. Register, the system will redirect the user to its' profile
11. In the profile page create a bLink with tag and space that were created by superuser
12. Click the name of the tag on the bLink while pressing CTRL (This will open the corresponding tags' page in a new tab)
13. In the newly opened tab, Create a new Tag with name " " (this indicates a blank space do not include quotation marks). This is very **important** since by default navbar redirects the user to the tag with name: " " page. Otherwise the system will raise an error when users try to get to the Tags page
14. Do the same for spaces
15. Beware of the **case sensitive** search functionality
16. Since there exist bugs with preview feature in v.09, users must be very careful with the links. Otherwise, the system would raise an error it the link is valid but no preview is available with the web scraper. However, this bug is fixed in v0.91, which cannot be released due to time constraints of this project. A kind suggestion for not to encounter this bug is to use YouTube or StackOverflow links.

Another pre-caution should be not to delete users with posts, since there exist a bug where **deleting a user** does not yield deleting its' bLinks. All known bugs are listed in [here](#).

17. You are ready to use the app.

8. Test Results

In order to run the test user need to use following commands in terminal.

1. Make sure of that the current directory is the root folder of the project
2. `cd SWE573_Term_Project`
3. `python manage.py test`

The results should be,

```
(.env) PS C:\Users\02485955\SWE573_Term_Project\swe573_term_project> python manage.py test
Found 8 test(s).
System check identified no issues (0 silenced).
.....
-----
Ran 8 tests in 0.235s

OK
(.env) PS C:\Users\02485955\SWE573_Term_Project\swe573_term_project>
```

The unit test for the system are for ensuring that,

- 1- Asserting create a Post model functionality is working properly
- 2- Asserting Post models owner username is equal to the owner username given by user while creating.
- 3- Asserting Post models link is equal to the link given by user while creating.
- 4- Asserting Post models caption is equal to the caption given by user while creating.
- 5- Asserting change username functionality is working properly.
- 6- Asserting change caption functionality is working properly.
- 7- Asserting delete Post model functionality is working properly.
- 8- Asserting that the Model object raises an AttributeError when user tries to reach an invalid attribute of the Model

9. Functionality Video

The demo video of the functionalities is [here](#).