**Logo

Description automatically generated**

*AUTHOR Student-ID*

Mümtaz Eralp Ünver 2021719162

**Project Final Deliverables**

**SWE-573 Software Development Practice / 2022 Fall Semester**

**Project Name: Research Box ( Django Web Development Project)**

**Git Repository:** [**https://github.com/eralp85/SWE-573.01**](https://github.com/eralp85/SWE-573.01)

**Git Tag Version:** [**https://github.com/eralp85/SWE-573.01/releases/tag/0.9**](https://github.com/eralp85/SWE-573.01/releases/tag/0.9)

**Deployed to URL :** [**http://52.23.241.177:8000/login/**](http://52.23.241.177:8000/login/)

**Test-User ID/Password = test /tester123**

**Admin-User ID / Password = eralp /admin123!**

**Jan 03, 2023**

**HONOR CODE**

Related to the submission of all the project deliverables for the Swe573 2022 Fall semester project reported in this report, I, Mumtaz Eralp Unver 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.

**Mumtaz Eralp Unver**

**Table Of Contents**

**1-) Overview of Research Box -------------------------------------------------------------------------------------**3

**2-) Software Requirements Specification ------------------------------------------------------------------------**3

A-) Terminology for Requirements **-----------------------------------------------------------------------** 3

B-) Functional Requirements **-------------------------------------------------------------------------------**3

i-) Authentication Requirements **------------------------------------------------------------------**3

ii-) Posts Requirements **----------------------------------------------------------------------------**3

iii-) User Requirements **----------------------------------------------------------------------------**4

1. *User Posts* **-----------------------------------------------------------------------------**4
2. *Profile* **---------------------------------------------------------------------------------**5

C-) Non-functional Requirements **--------------------------------------------------------------------------**5

i-) System Requirements **---------------------------------------------------------------------------**5

**3-) Design ---------------------------------------------------------------------------------------------------------------**6

A-) UML Diagram **-------------------------------------------------------------------------------------------**6

B-) Use-Case Diagram **--------------------------------------------------------------------------------------**6

C-) UML Sequence Diagram **-------------------------------------------------------------------------------**7

D-) Mockups **-------------------------------------------------------------------------------------------------**7

i-) Login Page **--------------------------------------------------------------------------------------**7

ii-) Home Page **-------------------------------------------------------------------------------------**8

iii-) *My Research* **-----------------------------------------------------------------------------------**8

*iv-) Friends* **-----------------------------------------------------------------------------------------**10

*v-) My Account* **-------------------------------------------------------------------------------------**10

**4-) Project’s Status --------------------------------------------------------------------------------------------------**11

A-) Requirements Check List **------------------------------------------------------------------------------**11

**5-) Deployment Status ----------------------------------------------------------------------------------------------**14

**6-) System Manual --------------------------------------------------------------------------------------------------**14

A-) System Manual **-----------------------------------------------------------------------------------------**14

B-) Docker Manual **-----------------------------------------------------------------------------------------**14

**7-) User manual ------------------------------------------------------------------------------------------------------**14

**8-) Test results --------------------------------------------------------------------------------------------------------**16

**9-) Video Link --------------------------------------------------------------------------------------------------------**17

**10-) References -------------------------------------------------------------------------------------------------------**17

**1-) Overview of Research Box**

The Research Box project, a web application developed under Django framework, aims to create a system that facilitates different users to aggregate information that they have encountered during their social media interaction under a hub.

Research Box project is able to save social media information that each user has collected on internet. Furthermore, Research Box enables users to share, tag and label the information with other users.

The main drivers of the project are users, and they will bring the content to the web application.

The Requirements of Research Box project is given in section 2.

**2-) Software Requirements Specification**

***A-) Terminology for Requirements***

**User** = User describes people who have registered the website with their valid e-mail and username.

**Post** = Post describes the content that users share in the system.

**Title =** Name of the post.

**Content**= Content describes the social media information that users have find in their daily social media journey.

**Like**= Like is a gesture that indicates users in the system enjoyed the social media content that a specific user shared.

**Prioritize**= Prioritize is a gesture that indicates post is important and has to be prioritized.

**Award**= Award is a gesture that indicates post is rich in content.

**Flash**= Flash is a gesture that indicates post is a new and might have an impact on markets.

**Tag** = Tag is a word or phrase that provides a generalization of the description of a post.

**Label =** Label is a word or phrase that categorizes a post.

***B-) Functional Requirements***

**i-) Authentication Requirements** (= [Login Page](https://github.com/eralp85/SWE-573.01/wiki/3.-Mock-Up#login-page))

• Users shall be able to register by providing e-mail and password.

• Users shall receive a verification e-mail after registration.

• Users shall be able to login with e-mail and password they have registered.

• Users shall be able to change password by using their previous password and the e-mail that they have registered.

• Users shall be able to log-out securely.

**ii-) Posts Requirements** (= [Home Page](https://github.com/eralp85/SWE-573.01/wiki/3.-Mock-Up#home-page))

• Users shall be able see the content on its own space.

• Users shall be able to view all the content shared by different users on a time base.

• Users shall be able to view sub-contents shared by different users on a time base.

• Users shall be able to select the shared-content and display it in any format.

• Users shall be able to tag, label and make comment on a shared content.

• Users shall be able to get tagged, labeled and made commented on a shared content.

• Users shall be able to like, prioritize, award, and flash content shared.

• Users shall browse the whole content.

• Users shall undertake advanced search the whole content and sub-contents.

• Users shall undertake the search through a specific order that is Topic, Time, Tag, and Source.

• Users shall be able to classify or group the shared content.

• Users shall see sub-categories of the social media content like Macro, FI, EQ etc.

• Users should be able to track other users’ activities.

• Users shall be able to navigate through different pages.

**iii-) User Requirements**

**a-)User Posts** (= [My Research Page](https://github.com/eralp85/SWE-573.01/wiki/3.-Mock-Up#my-research), [Friends Page](https://github.com/eralp85/SWE-573.01/wiki/3.-Mock-Up#friends))

• Users shall have their own space.

• Users shall be able to copy a link of post to create a content.

• Users shall be able to give description or a name the post they share.

• Users shall be able to label a content.

• Users shall be able to tag a content.

• Users shall be able to commend on a content.

• Users shall be able to edit a content.

• Users shall be able to delete a content.

• Users shall be able to see their own activities on a time base.

• Users shall be provided with information recommendations based on content labels and tags.

• Users shall be able to see other users' profile.

• Users shall be able to see other users' activities.

• Users shall be able to tag other users and their activities.

• Users shall be able to label other users and their activities.

• Users shall be able to comment other users and their activities.

• Users shall be able to like, prioritize, award, and flash other users activities.

#### b-) Profile (= [My Account](https://github.com/eralp85/SWE-573.01/wiki/3.-Mock-Up#my-account))

• Users shall see and update their own accounts.

• Users shall be able to communicate with each other.

• Users shall be able to follow any sub-content that they are interested in.

• Users shall be able to hold the history lag even though their accounts are erased.

***C-) Non-Functional Requirements***

**i-) System Requirements** (= [Frequently Asked Questions](https://github.com/eralp85/SWE-573.01/wiki/3.-Mock-Up/_edit#frequently-asked-question))

• The system shall be in English.

• The system shall be web-based.

• The system shall be usable via any common browser.

• The system shall use Python as programming language and Django as its framework.

• The system shall use postgresql as database.

• The system shall be up 23.5 hours for every day in a week.

• The system shall get response time from server under 5 seconds.

• The system shall get a back-up every day.

• The system shall be accessed by only-users.

• The system shall be protected by any sort of malicious internet attacks.

**3-) Design**

***A-) UML Class Diagram***

***Diagram

Description automatically generated***

***B-) Use-Case Diagrams:***

***Diagram

Description automatically generated***

***C-) UML Sequence Diagrams:***

***Timeline

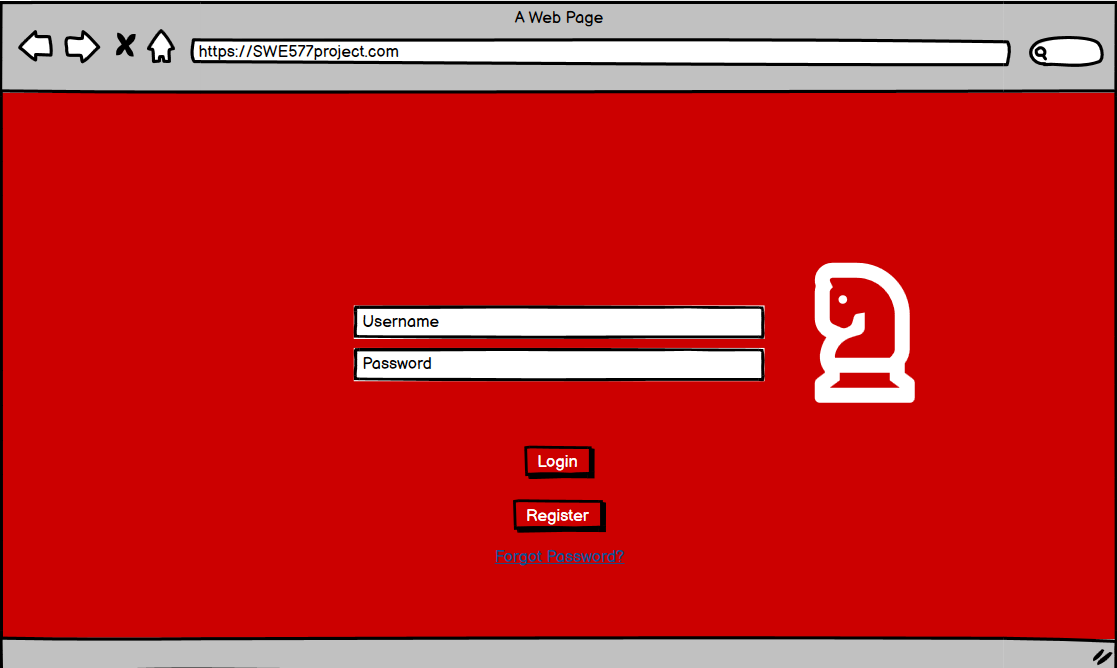
Description automatically generated***

***D-) Mockups***

Balsamiq is being used to prepare mockups at the beginning of the Research Box project. Though the front-end has changed throughout the development, the idea of the frame and mapping has been kept.

**i-) Login Page**

Login page offers 3 functionalities. Login, register, and change of password. Users require to have a valid e-mail address and password. Please see [Authentication Requirements](https://github.com/eralp85/SWE-573.01/wiki/2.3-Requirements-of-Research-Box#1--authentication-requirements--login-page)for detailed information.

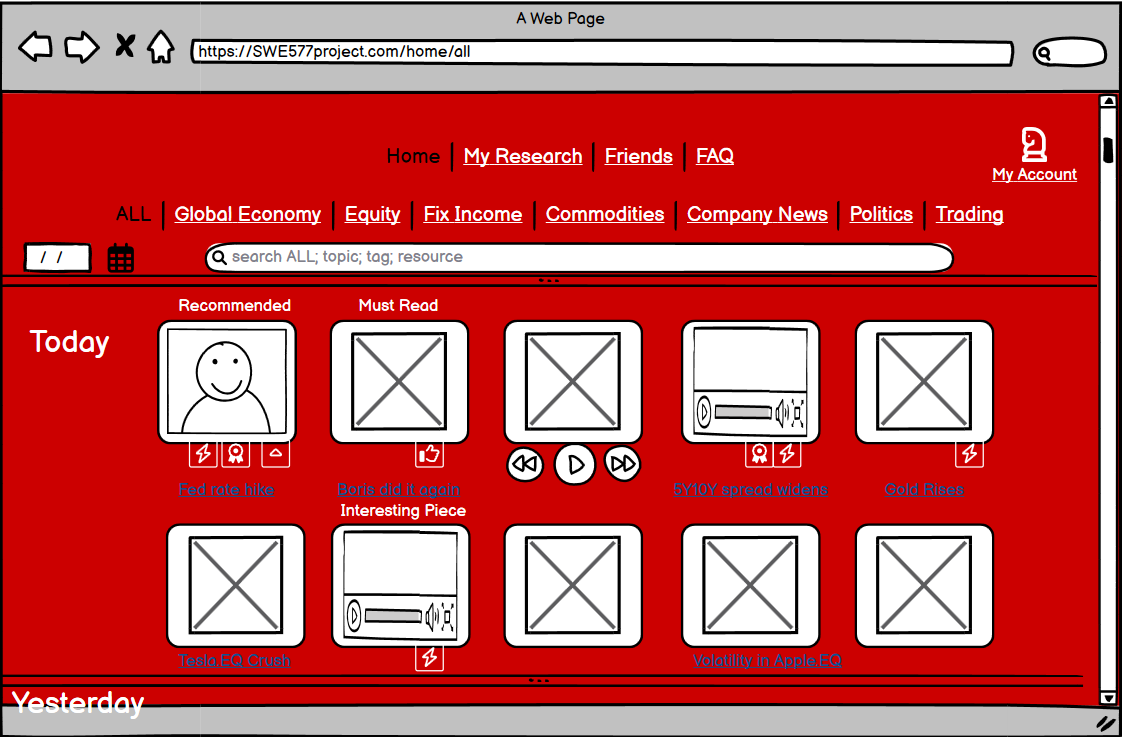


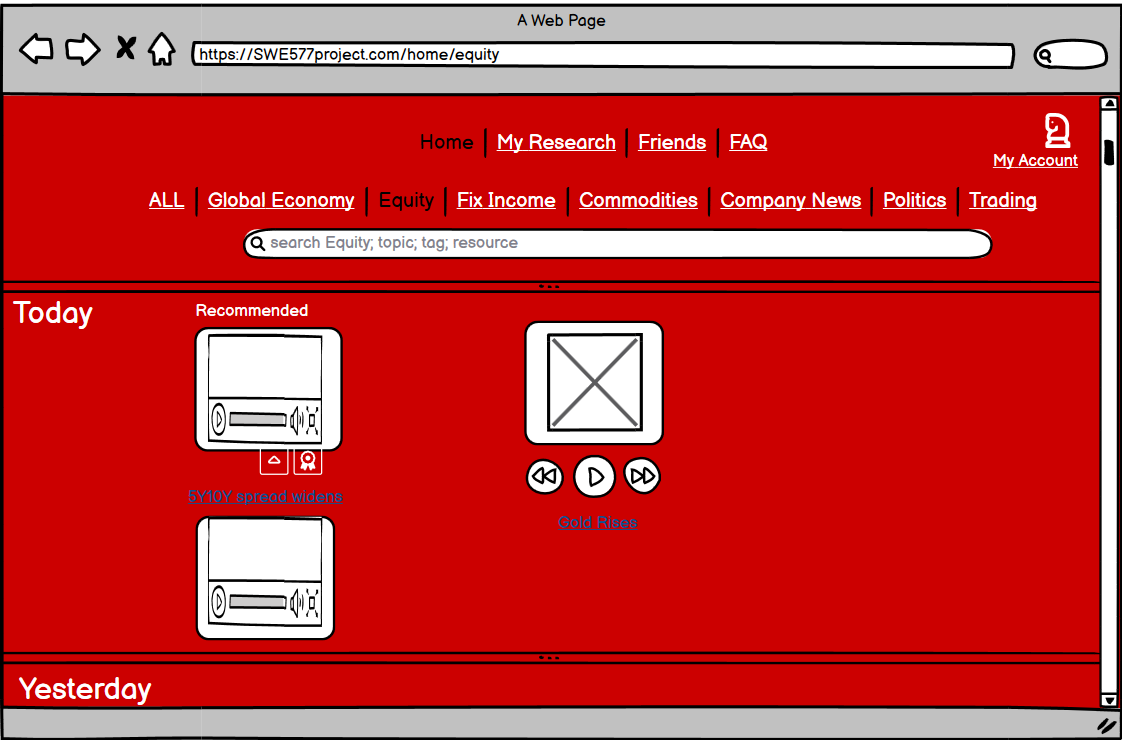
**ii-) Home Page**

Homepage is a space for shared-content. Home page enables users to see and classify the content that other users share in any format on a time base. Users will be able to see shared-content title, description of it and the owner of it. Furthermore, users will be able to share, go to linked page, comment and like the post that is shared.

Further, classification is done by label based asset categories. If the topic in mind is going to be researched, it can also be conducted via topic selection.

Please see [Posts Requirements](https://github.com/eralp85/SWE-573.01/wiki/2.3-Requirements-of-Research-Box#2--posts-requirements--home-page)for detailed information.





**iii-) My Research**

My Research is a space for users. Each user have its own space. My research shows the recent content that is shared by user's. Also, it is the page that user can **ADD**, **EDIT** and **DELETE** a content that s/he shared.

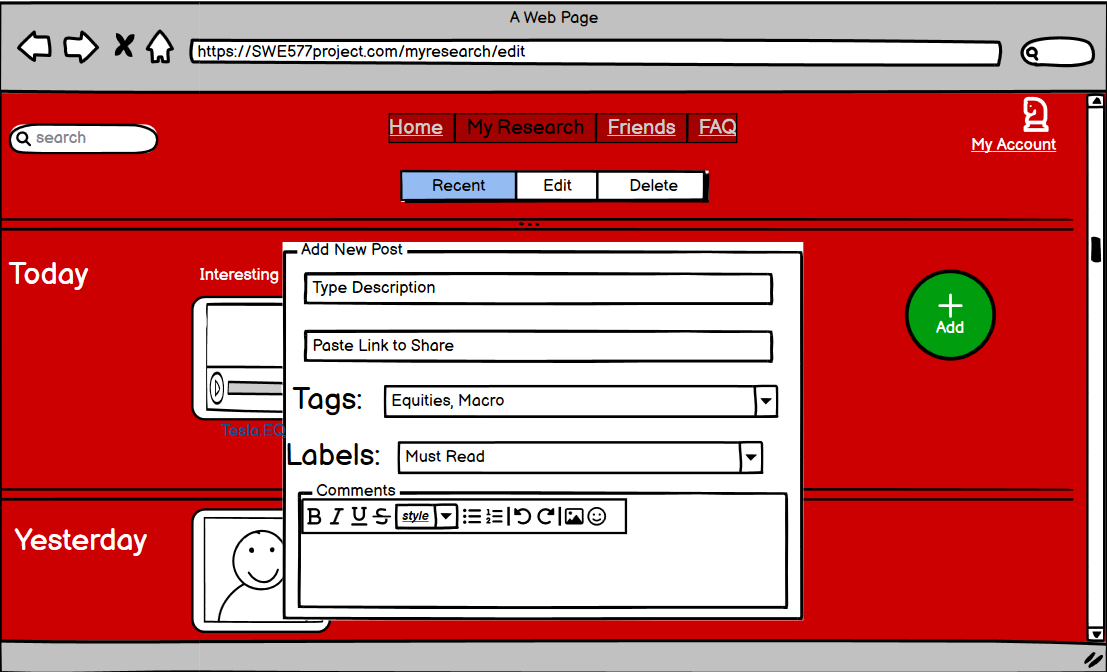
**ADD** function enables user to add a content by copying and pasting the link of the content they find in other social media content. They can put a description of a link they shared.

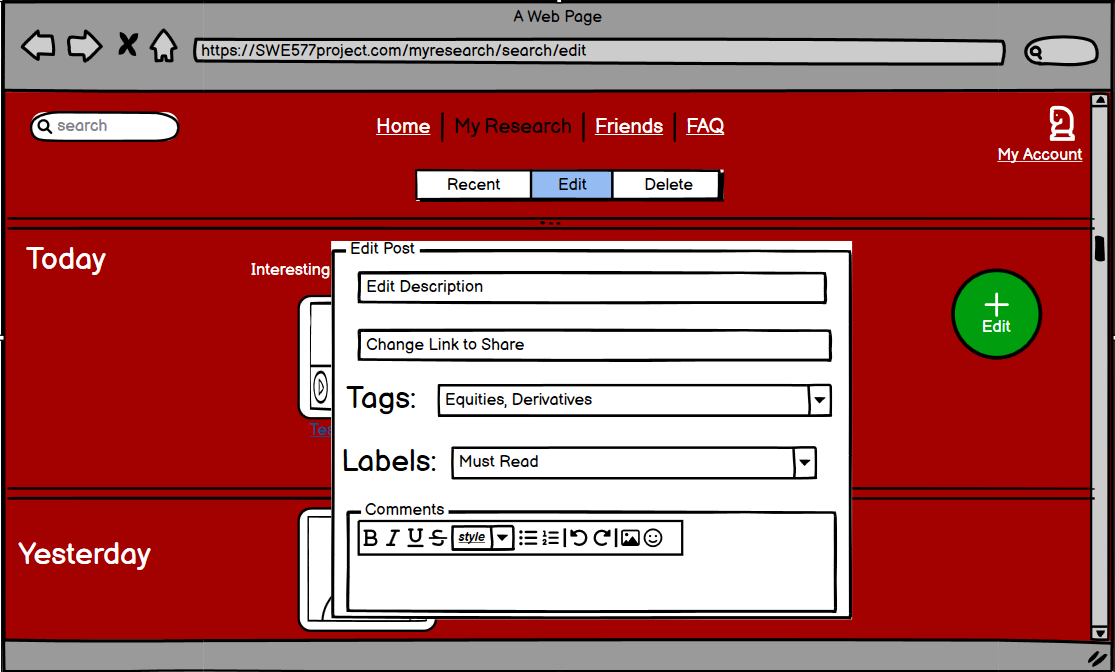
**EDIT** function enables user to change or make amendments of the previously added content.

**ADD** and **EDIT** functions also enables users to tag, label and comment what they shared.

**DELETE** function enables users to erase the content they previously shared.

Please see [User Posts Requirements](https://github.com/eralp85/SWE-573.01/wiki/2.3-Requirements-of-Research-Box#3--user-requirements)for detailed information.

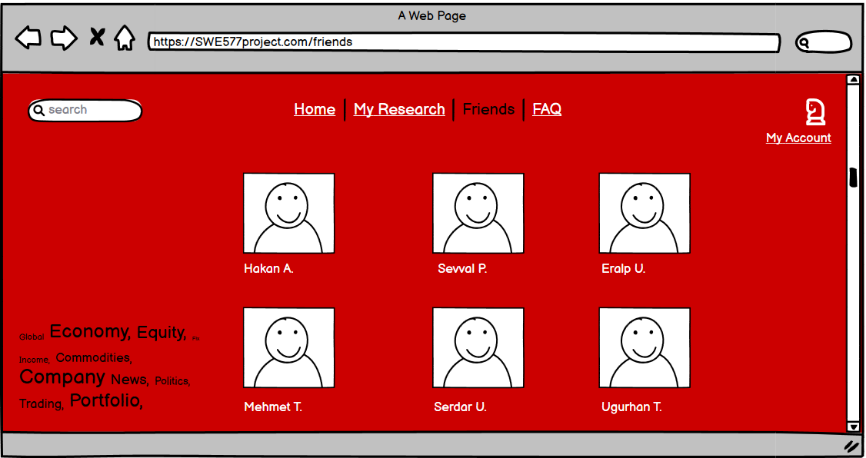




**iv-) Friends**

Users can see the other users and can see and tag what other users have shared.

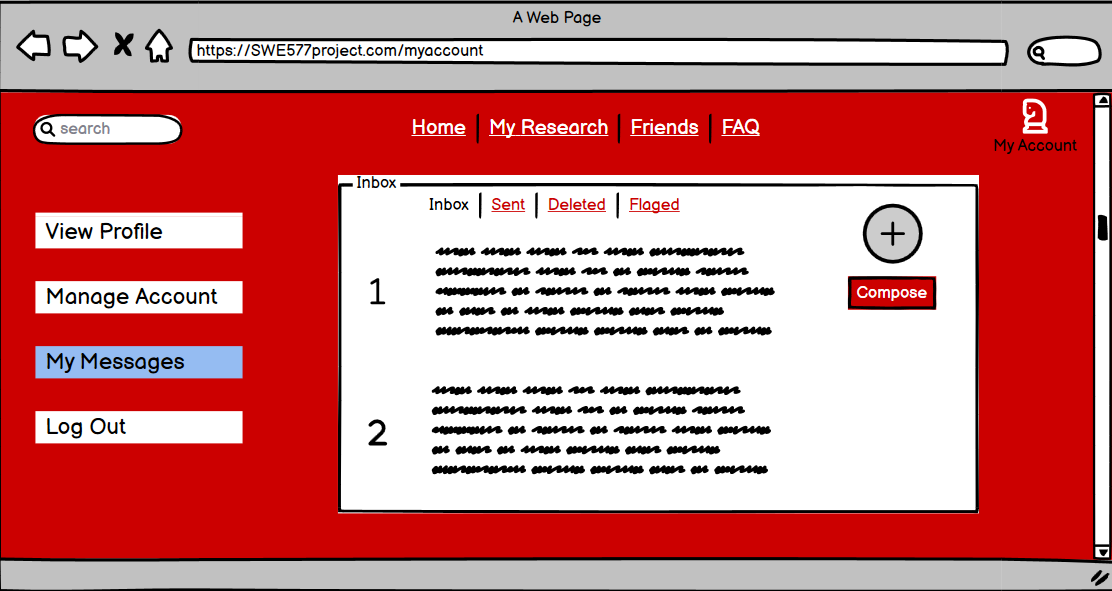
Please further see  [User Posts Requirements](https://github.com/eralp85/SWE-573.01/wiki/2.3-Requirements-of-Research-Box#3--user-requirements)



**v-) My Account**

My Account is also another user space. My Account page provides users to see and manage their profiles. Users can safely log-out from their account by clicking Log Out button.

Please see [Functional Requirements](https://github.com/eralp85/SWE-573.01/wiki/2.3-Requirements-of-Research-Box#b--functional-requirements)for further information.



**4-) Project’s Status**

|  |  |  |  |
| --- | --- | --- | --- |
| **Item Number** | **Requirement** | **Status** | **Detail** |
| Authentication - 1 | Users shall be able to register by providing e-mail and password. | 🗹 | [Issue: Authentication Requirements](https://github.com/eralp85/SWE-573.01/issues/19) |
| Authentication - 2 | Users shall receive a verification e-mail after registration. | 🗹 | [Issue: E-mail Services](https://github.com/eralp85/SWE-573.01/issues/8) |
| Authentication - 3 | Users shall be able to login with e-mail and password they have registered. | 🗹 | [Issue: Authentication Requirements](https://github.com/eralp85/SWE-573.01/issues/19) |
| Authentication - 4 | Users shall be able to change password by using their previous password and the e-mail that they have registered. | 🗹 | [Issue: Authentication Requirements](https://github.com/eralp85/SWE-573.01/issues/19) |
| Authentication - 5 | Users shall be able to log-out securely. | 🗹 | [Issue: Authentication Requirements](https://github.com/eralp85/SWE-573.01/issues/19) |
| Post - 1 | Users shall be able see the content on its own space. | 🗹 | [Issue: User-Space Configurations ; Issue: User-Spaces and related functions](https://github.com/eralp85/SWE-573.01/issues/16;%20https:/github.com/eralp85/SWE-573.01/issues/15) |
| Post - 2 | Users shall be able to view all the content shared by different users on a time base. | 🗹 | [Issue: Add new fundamental features](https://github.com/eralp85/SWE-573.01/issues/12) |
| Post - 3 | Users shall be able to view sub-contents shared by different users on a time base. | 🗹 | [Issue: Add new fundamental features](https://github.com/eralp85/SWE-573.01/issues/12) |
| Post - 4 | Users shall be able to select the shared-content and display it in any format. | 🗹 | [Issue: User-Space Configurations ; Issue: User-Spaces and related functions](https://github.com/eralp85/SWE-573.01/issues/16;%20https:/github.com/eralp85/SWE-573.01/issues/15) |
| Post - 5 | Users shall be able to tag, label and make comment on a shared content. | 🗹 | [Issue: Add new fundamental features](https://github.com/eralp85/SWE-573.01/issues/12) |
| Post - 6 | Users shall be able to get tagged, labeled and made commented on a shared content. | 🗹 | [Issue: Add new fundamental features](https://github.com/eralp85/SWE-573.01/issues/12) |
| Post - 7 | Users shall be able to like, prioritize, award, and flash content shared. | 🗷 | Only Like feature is added. |
| Post - 8 | Users shall browse the whole content. | 🗹 | [Issue : Add Search function to make query](https://github.com/eralp85/SWE-573.01/issues/13) |
| Post - 9 | Users shall undertake advanced search the whole content and sub-contents. | 🗹 | [Issue : Add Search function to make query](https://github.com/eralp85/SWE-573.01/issues/13) |
| Post - 10 | Users shall undertake the search through a specific order that is Topic, Time, Tag, and | 🗹 | [Issue : Add Search function to make query](https://github.com/eralp85/SWE-573.01/issues/13) |
| Post - 11 | Users shall be able to classify or group the shared content. | 🗹 | [Issue : Add Search function to make query](https://github.com/eralp85/SWE-573.01/issues/13) |
| Post - 12 | Users shall see sub-categories of the social media content like Macro, FI, EQ etc. | 🗹 | [Issue : Add Search function to make query](https://github.com/eralp85/SWE-573.01/issues/13) |
| Post - 13 | Users should be able to track other users’ activities. | 🗷 | Follow Function is not added. |
| Post - 14 | Users shall be able to navigate through different pages. | 🗹 | [Issue: Add new fundamental features](https://github.com/eralp85/SWE-573.01/issues/12) |
| User / Post - 1 | Users shall have their own space. | 🗹 | [Issue: User-Space Configurations ; Issue: User-Spaces and related functions](https://github.com/eralp85/SWE-573.01/issues/16;%20https:/github.com/eralp85/SWE-573.01/issues/15) |
| User / Post - 2 | Users shall be able to copy a link of post to create a content. | 🗹 | [Issue: Add new fundamental features](https://github.com/eralp85/SWE-573.01/issues/12) |
| User / Post - 3 | Users shall be able to give description or a name the post they share | 🗹 | [Issue: Add new fundamental features](https://github.com/eralp85/SWE-573.01/issues/12) |
| User / Post - 4 | Users shall be able to label a content. | 🗹 | [Issue: Add new fundamental features](https://github.com/eralp85/SWE-573.01/issues/12) |
| User / Post - 5 | Users shall be able to tag a content. | 🗹 | [Issue: Add new fundamental features](https://github.com/eralp85/SWE-573.01/issues/12) |
| User / Post - 6 | Users shall be able to commend on a content. | 🗹 | [Issue: Add new fundamental features](https://github.com/eralp85/SWE-573.01/issues/12) |
| User / Post - 7 | Users shall be able to edit a content. | 🗹 | [Issue: Add new fundamental features](https://github.com/eralp85/SWE-573.01/issues/12) |
| User / Post - 8 | Users shall be able to delete a content. | 🗹 | [Issue: Add new fundamental features](https://github.com/eralp85/SWE-573.01/issues/12) |
| User / Post - 9 | Users shall be able to see their own activities on a time base. | 🗹 | [Issue: User-Space Configurations ; Issue: User-Spaces and related functions](https://github.com/eralp85/SWE-573.01/issues/16;%20https:/github.com/eralp85/SWE-573.01/issues/15) |
| User / Post - 10 | Users shall be provided with information recommendations based on content labels and tags. | 🗷 | Since follow function is not added, users cannot be differentiated. |
| User / Post - 11 | Users shall be able to see other users' profile. | 🗹 | [Issue: User-Space Configurations ; Issue: User-Spaces and related functions](https://github.com/eralp85/SWE-573.01/issues/16;%20https:/github.com/eralp85/SWE-573.01/issues/15) |
| User / Post - 12 | Users shall be able to see other users' activities. | 🗷 | Since follow function is not added, users cannot be differentiated. |
| User / Post - 13 | Users shall be able to tag other users and their activities. | 🗹 | [Issue: Add new fundamental features](https://github.com/eralp85/SWE-573.01/issues/12) |
| User / Post - 14 | Users shall be able to label other users and their activities. | 🗹 | [Issue: Add new fundamental features](https://github.com/eralp85/SWE-573.01/issues/12) |
| User / Post - 15 | Users shall be able to comment other users and their activities. | 🗹 | [Issue: Add new fundamental features](https://github.com/eralp85/SWE-573.01/issues/12) |
| User / Post - 16 | Users shall be able to like, prioritize, award, and flash other users activities. | 🗹 | [Issue: Add new fundamental features](https://github.com/eralp85/SWE-573.01/issues/12) |
| User / Profile - 1 | Users shall see and update their own accounts. | 🗹 |  |
| User / Profile - 2 | Users shall be able to communicate with each other. | 🗷 | Since follow function is not added, users cannot be differentiated. |
| User / Profile - 3 | Users shall be able to follow any sub-content that they are interested in. | 🗷 | There is no content following |
| User / Profile - 4 | Users shall be able to hold the history lag even though their accounts are erased. | 🗹 | [Issue: E-mail Services](https://github.com/eralp85/SWE-573.01/issues/8) |
| System - 1 | The system shall be in English. | 🗹 | [Issue: Set Up a Django Framework for the project](https://github.com/eralp85/SWE-573.01/issues/8) |
| System - 2 | The system shall be web-based. | 🗹 | [Issue: Set Up a Django Framework for the project](https://github.com/eralp85/SWE-573.01/issues/8) |
| System - 3 | The system shall be usable via any common browser. | 🗹 | [Issue: Set Up a Django Framework for the project](https://github.com/eralp85/SWE-573.01/issues/8) |
| System - 4 | The system shall use Python as programming language and Django as its framework. | 🗹 | [Issue: Set Up a Django Framework for the project](https://github.com/eralp85/SWE-573.01/issues/8) |
| System - 5 | The system shall use postgresql as database. | 🗹 | [Issue: Set Up a Django Framework for the project](https://github.com/eralp85/SWE-573.01/issues/8) |
| System - 6 | The system shall be up 23.5 hours for every day in a week. | 🗹 | [Issue: Set Up a Django Framework for the project](https://github.com/eralp85/SWE-573.01/issues/8) |
| System - 7 | The system shall get response time from server under 5 seconds. | 🗹 | [Issue: Set Up a Django Framework for the project](https://github.com/eralp85/SWE-573.01/issues/8) |
| System - 8 | The system shall get a back-up every day. | 🗹 | [Issue: Set Up a Django Framework for the project](https://github.com/eralp85/SWE-573.01/issues/8) |
| System - 9 | The system shall be accessed by only-users. | 🗹 | [Issue: Set Up a Django Framework for the project](https://github.com/eralp85/SWE-573.01/issues/8) |
| System - 10 | The system shall be protected by any sort of malicious internet attacks. | 🗹 | [Issue: Set Up a Django Framework for the project](https://github.com/eralp85/SWE-573.01/issues/8) |

**5-) Deployment Status**

Most of the requirements of the Research Box Project has been fulfilled and implemented. The requirements related to “following a user” has not been implemented due to the problems occurred in terms of database. Therefore, database has been changed from MySql to PostGreSql. Yet, those functions has been implemented due to time constraint of the project.

Research box is dockerized and deployed on AWS service.

Docker-compose.yml, dockerfile and requirements.txt can be reached from [here.](https://github.com/eralp85/SWE-573.01/wiki/2.3-Requirements-of-Research-Box)

URL that Research Box works on is <http://52.23.241.177:8000/login>.

The steps to how run Research Box on a system will be discussed in the next section.

**6-) System Manual**

***A-) System Manual***

Research Box project is developed in Django, a high-level Python framework. Therefore, in order to run the project, latest version of Python shall be downloaded ([here](https://www.python.org/downloads/)) and on a IDE that users prefer. The project developed with django==3.1.1 version.

However, dependencies are listed on requirements.txt file and can be found on the [Git Repository](https://github.com/eralp85/SWE-573.01/blob/master/requirements.txt) . Use the pip install -r requirements.txt command to install all of the Python modules and packages listed in requirements. txt file.

***B-) Docker Manual***

In order to run Research Box on a local environment, dockerization is required. Below steps offer users how to dockerize the project.

1-) Install Docker Compose to your system from [here](https://www.docker.com/). (Follow the [guidance](https://docs.docker.com/desktop/install/windows-install/) provided by Docker itself). After installation please run the Docker on your desktop.

2-) To run the project, the source codes of the Research Box needs to be on local environment. You can download it from Github manually in [here](https://github.com/eralp85/SWE-573.01/blob/master/requirements.txt) or clone the code by running following command. Git clone <https://github.com/eralp85/SWE-573.01.git>

3-) Navigate until you reach root directory cd SWE-573.01 and run docker-compose build.

4-) The make the Docker containers running please run docker-compose up.

5-) On your browser, type 127.0.0.1:8000 and you will be able to use the application.

**7-) User Manual**

User manual will provide guidance users over the functionalities and features of the Research Box. User Manual will start from the beginning of registration of a user to the logout.

**Registration Page:** A new coming users has to register to system before they start sharing a content. They have to first go to login page at <http://52.23.241.177:8000/login/> and click on register button. New users have to provide a user name, e-mail address and password that has specific complexity security requirements. New users have to fill all the fields in registration form to open an account. After registration, application will direct new user to home page.

**Login Page:** For registered users to use the system, they need to login to system by entering their user name and password. After successful login, users will be redirected to home page. If a user forgets his/her password, they can reclaim it by entering their valid e-mail address. New password will be sent to their e-mail address.

**Home Page**: Registered users will be able to see all the content that has been posted at home page. In other words, home page shows the list of the posts sorted on a time base. New posts will be at the top of the page. At home page, users can see My Research, Friends, FAQs and My Account subsections. Further, users will be able to see the sub-categories of posts and a search box.

At the home page, users can add a post by clicking on Box\_it Button. Further, users can share the post by e-mail, comment, go to link or like a post on the home page.

In order to see what is shared in the post, user can click on the individual post and see what is inside of the content in a more detailed way.

**My Account:** My account is the profile page that a user can enter their personal information, update their user photo and change their password. Personal information includes first name, last name, e-mail address, date of birth and a picture. Users are strongly advised to update their profile as soon as they log in to application.

**Friends:** Friends page is where users can see other users. List of users will appear on this page. However, due to unfinished requirement users cannot see other users activities or see the details of the users.

**My Research:** My Research page shows the total posts that a user have posted since they have registered. Users can see post title, post link, post label and post text on what they have shared. They can also share, edit, delete, comment the posts they have shared. Users can only edit and delete their own comments. Other users cannot make any changes on the private posts.

**Search:** Search function works by entering the name of the title or a related name referring to title. Users can also search through tags that are on the posts.

**Box\_it:** Box\_it page offers user to post a content. When a user wants to share a content they should enter title, link, related tag and label and text. They can also upload files.

**Post View:** In order to see the details of a shared content, users can just click on the related post. When the post is clicked, users will be able to see subsections of the content. They can also comment on a post with more detailed information. From the post view, users can also give a like to a post.

**8-) Test Results:**

12 different unit-tests were executed under 4 different categories by using assert statements.

1-) First category unit – tests are about Post function. Assert statements are placed to check;

1. If a post exists or created,
2. if an authorized user exists,

2-) Second category unit – tests are about authentication processes. Assert statements are used to control

1. Login,
2. Registration,
3. Registration with common password.
4. Valid password and username

3-) Third category unit - tests about Search and Tag functions. Assert statements are set to check;

1. If searched function works correctly.
2. if tag is correct and has a name.

4-) Fourth category unit - tests about Comment function and profile functions. Assert statements are placed to check;

1. Comment has an author,
2. Comment has a text
3. Profile is valid
4. Profile can be updated.

Furthermore, 12 different user acceptance tests are undertaken to see if the functions are working. These tests include;

|  |  |  |  |
| --- | --- | --- | --- |
| **Item Number** | **Requirement** | **Status** | **Detail** |
| /Registration-1 | If a new user can with the same name. | þ | No, they cannot |
| /Registration-2 | If a user can register with poor password. | þ | No, they cannot |
| /Home-1 | If a user can see last content. | þ | Yes, they can |
| /Home-2 | If a user can go to a link. | þ | Yes, they can |
| /Home-3 | If a user can comment. | þ | Yes, they can |
| /Home-4 | If a user like. | þ | Yes, they can |
| /Home-5 | If a user can share. | þ | Yes, they can |
| /Search-1 | If a user can find relevant content. | þ | Yes, they can |
| /Search-2 | If a user can use tags to search. | þ | Yes, they can |
| /Post-1 | If a user can edit or delete others content. | þ | No, they cannot |
| /Post-2 | If a user can see their own content on a separate space. | þ | Yes, they can |
| /Post-3 | If a content has its own space in terms of label | þ | Yes, it has |

**9-) Functionality Video:**

You find the demo video of the functionalities in [here.](https://drive.google.com/drive/folders/1eOjmWPBsVahqWNeWnuvgf176MmJKPMuz?usp=sharing)

**10- References:**

Throughout the project, below sources are being used to get a referenced knowledge.

* Django Girls Website, a tutorial. <https://tutorial.djangogirls.org/en/django/>
* Melé, Antonio, Django 3 by Example, March 2020
* Sommerville, Ian, Software Engineering, 10th edition, 2020.
* <https://docs.djangoproject.com/en/4.1/topics/testing/tools/>
* <https://realpython.com/python-assert-statement/>
* <https://aws.amazon.com/tr/>
* <https://docs.aws.amazon.com/codedeploy/latest/userguide/tutorials-windows-deploy-application.html>
* <https://www.docker.com/>
* <https://docker-curriculum.com/>
* <https://docs.djangoproject.com/en/3.2/topics/i18n/>