

# Final Project Report

## Designing and Creating of a Collaborative Learning Space



Bogazici University  
SWE 573 – Software Development Practice

Assist. Prof. Suzan Uskudarli

Project Name: CoLearn (Collaborative Learning) Platform

Git repo: <https://github.com/turanlioglu/bounswe573-2022.git>

Git tag version: <https://github.com/turanlioglu/bounswe573-2022/releases/tag/v0.9>

Deployment URI: <http://ec2-34-220-67-213.us-west-2.compute.amazonaws.com>

### HONOR CODE

Related to the submission of all the project deliverables for the Swe573 2022 Spring semester project reported in this report, **I, Sabahattin Erdem Turanlioğlu declare that:**

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

---

Prepared by: Sabahattin Erdem Turanlioglu  
Due Date: 31.05.2022

Signature



## **Table of contents**

<b>DECLARATION OF WORK SUBMITTED .....</b>	<b>4</b>
<b>NECESSARY INFORMATION TO USE DEPLOYED SYSTEM .....</b>	<b>5</b>
<b>PROJECT DETAILS .....</b>	<b>6</b>
<b>Overview .....</b>	<b>6</b>
<b>Software Requirements Specification .....</b>	<b>6</b>
<b>Design (Software &amp; Mockups).....</b>	<b>8</b>
First system design by mockups: .....	8
Use case, Class, and Sequence Diagrams: .....	9
Use case diagram: .....	9
Class diagram:.....	10
Sequence diagrams:.....	10
1. Registration .....	10
2. Log-in .....	11
3. Log-out .....	11
4. Posting new messages .....	12
5. Commenting to posts.....	12
6. Liking post.....	13
7. Displaying user posts .....	13
8. Creating and displaying challenges.....	14
9. Submitting an answer to the challenge.....	14
10. Giving feedback to the challenges .....	15
11. Adding resources.....	15
<b>STATUS OF PROJECT .....</b>	<b>16</b>
<b>STATUS OF DEPLOYMENT .....</b>	<b>16</b>
<b>SYSTEM MANUAL .....</b>	<b>17</b>
<b>USER MANUAL .....</b>	<b>18</b>
<b>TEST RESULTS.....</b>	<b>19</b>
<b>FUTURE WORK.....</b>	<b>20</b>

## Declaration of work submitted

I confirm that the work for the following report with the title: “Software Project Development of a Web Application” was solely undertaken by myself. All sections of the paper that describe or concept developed by another author have been referenced.

List of 3<sup>rd</sup> party software with their licenses:

- psycopg2 -- <http://www.gnu.org/licenses/lgpl-3.0-standalone.html>
- asgiref -- <https://pypi.org/project/asgiref/3.5.2/>
- django -- <https://www.djangoproject.com/download/>
  - bootstrap4
  - ckeditor
  - crispy forms
  - hitcount
  - js-asset
  - taggit
- pillow -- <https://pillow.readthedocs.io/en/stable/about.html>
- css files -- based on tutorials that I shared below.

PostgreSQL installation

Used below link to connect my created project with PostgreSQL database.

- <https://stackpython.medium.com/how-to-start-django-project-with-a-database-postgresql-aaa1d74659d8>

Tutorials for Django environments and its applications

During the project, I tracked different tutorials from different sources and applied the needs on to my project, also developed some code blocks which is needed for the final work:

- [https://www.youtube.com/playlist?list=PLEsfXFp6DpzRMby\\_cSoWTFw8zaMdTEXgL](https://www.youtube.com/playlist?list=PLEsfXFp6DpzRMby_cSoWTFw8zaMdTEXgL)
- <https://www.youtube.com/channel/UCmrvAIpk1lL8WlalusTRlnw/videos>
- <https://www.youtube.com/playlist?list=PLCC34OHNCotr025c1kHSPnP18YPB-NFi>
- <https://www.youtube.com/playlist?list=PLCC34OHNCotqW9BJmgQPPzUpJ8hl49AGy>
- [https://www.youtube.com/playlist?list=PLCC34OHNCotpyEQqVVVoGu8QVvAU\\_6cgGp](https://www.youtube.com/playlist?list=PLCC34OHNCotpyEQqVVVoGu8QVvAU_6cgGp)
- [https://www.youtube.com/playlist?list=PLCC34OHNCotoYVT2654KIzait8\\_eYO\\_j5](https://www.youtube.com/playlist?list=PLCC34OHNCotoYVT2654KIzait8_eYO_j5)

Dockerization & Deployment

After every feature added (or tried to develop its latest version), the project dockerized before the deployment.

Dockerization:

- <https://testdriven.io/blog/dockerizing-django-with-postgres-gunicorn-and-nginx/>
- [https://www.youtube.com/watch?v=mScd-Pc\\_pX0&t=1748s](https://www.youtube.com/watch?v=mScd-Pc_pX0&t=1748s)

Deployment:

For deployment, based on my research I continued with AWS EC2 platform. Beside that, I also tried to use AWS EB. Related documents that I followed on EC2:

- <https://www.youtube.com/watch?v=DNS6OcwcxV8>
- [https://www.youtube.com/watch?v=mScd-Pc\\_pX0&t=1748s](https://www.youtube.com/watch?v=mScd-Pc_pX0&t=1748s)

## Necessary information to use deployed system

The zip file includes the following:

- Source code of project
- Secret-key for AWS machine
- Demo
- Final Report

For using deployed system:

User information:

<b>Username</b>	<b>Password</b>
admin	G2vLmhTXr3
turanlioglu	erdem123
mete.k	asedc123
sibel_ss	Qazws123

You can follow the instructions, which are stated in the demo, in given deployed URL of the project.

## Project details

### Overview

**Goal:** Creating co-learning platform for users, who wants to share information and drive personal improvements on web. The platform is structured in a manner, which users:

- can create rooms
- enrolls the courses they interested in
- in rooms, they can be able to share posts, create challenges or share resources with each other

There are some prerequisites for construct a platform like:

- Prior knowledge on python
- Knowledge on Django
- Knowledge of database (PostgreSQL is used in the project)
- Knowledge on GitHub

When all prerequisites met, software lifecycle methods are used to sustain a continuous improvement and continuous development of the system.

- Creation of milestones
  - Planning how many milestones should be needed
  - What will be the content of milestones, how should they be separated from each other?
  - Each milestone should have a deadline to follow up the project timeline
  - Creation of issues with specific to milestones to increase the following up process

### Software Requirements Specification

Below, there is a list of requirements, which is set in the beginning of the project to create a work plan. Some of the requirements are successfully added into the platform and some not. I will give detailed information later.

- The system shall support web.
- The system shall have an easy registration.
- The system shall have scheduling future for meetings.
- The system shall have provided users to ask & answer to the questions.
- The system shall have voting feature.
- The system shall have calendar information that shows people schedules in the spaces.
- The system shall enable users to edit their calendars.
- The system shall enable users to send direct messages.
- The system shall enable users to send meeting invitations.
- The system shall have user profiles that users can edit their information.
- The system shall have crowd filtering for false information and harassment.
- The system shall have collaborative editing in the channels.
- The system shall have search motor for to search specific subjects in the main page.
- The system shall have search motor for to search subjects in the channel.
- The system shall support file sharing with the limit of 25 MB.

- The channels(rooms) have multimedia message sharing.
- The channels(rooms) shall have unlimited member.
- The web application shall have a learning tracking system by personally created quizzes.
- The web application shall enable users to create badges on their achievements.
- The application shall send notifications to users.
- The application shall enable users to set notifications as their preferences.
- The application shall show topics to users based on their interests.
- The space owners shall add space prerequisites into the space information.
- The space subscribers shall edit space information with permission.
- The application shall have whiteboard for taking notes collaboratively in the meetings.
- The application shall enable uses to have activity performance.
- The application shall show the followers to users.

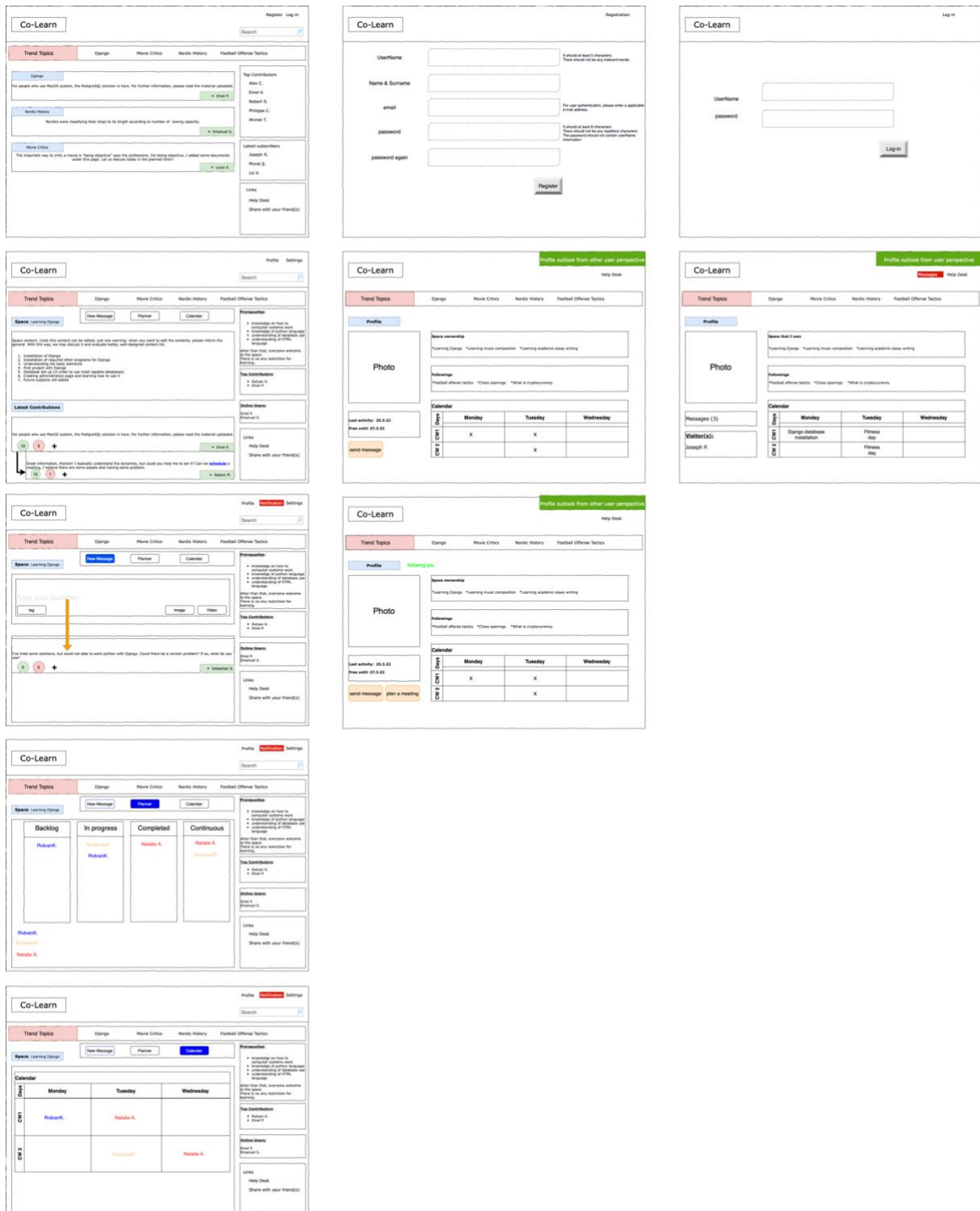
Requirements such notifications, meetings, learning tracking system, voting, badges for achievements, could not added in the system regarding their complexity is high and hard to connect the backend structures. Beside them, different features added in the application, which were not in the requirements system firstly. These are:

- Comments on posts,
- Like/unlike feature on posts,
- View numbers of posts,
- Counting of comment numbers to post specific,
- Counting enrolled users in the rooms,
- Creation of challenges

Later, these new requirements added in the design of the platform, which they can be seen in the [Design](#) section of this report.

# Design (Software & Mockups)

First system design by mockups:



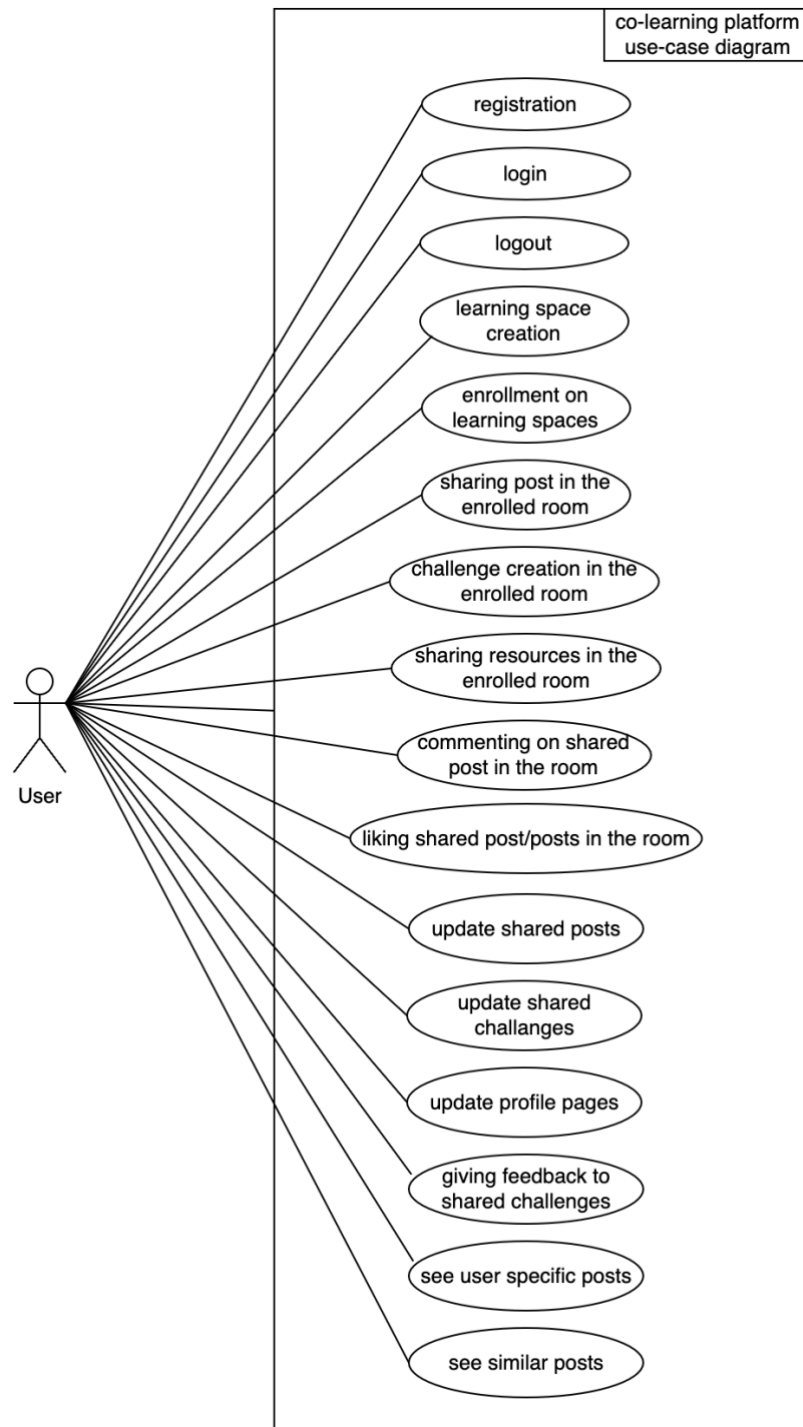


Use case, Class, and Sequence Diagrams:

All designs are updated or changed regarding to the requirements, that are set in the final version of the application.

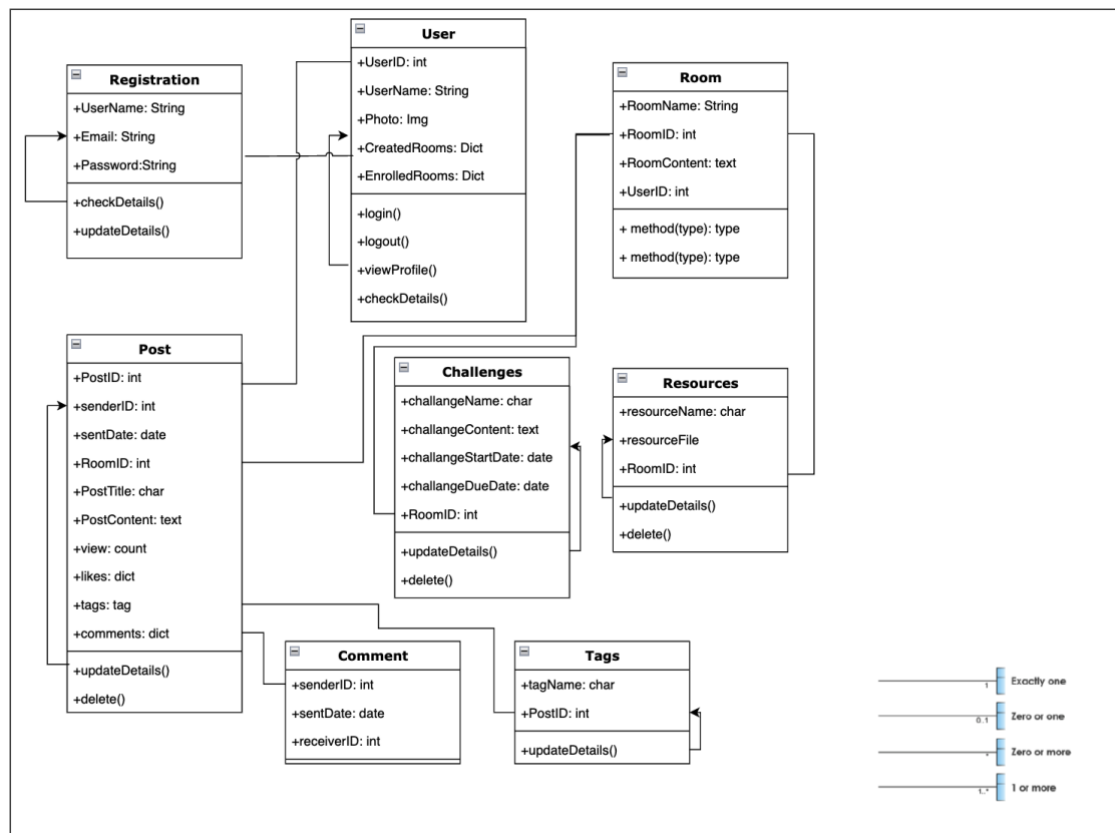
Use case diagram:

The diagram is updated regarding with new requirements, also previously defined requirements are removed, if the final version of the platform does not include it.



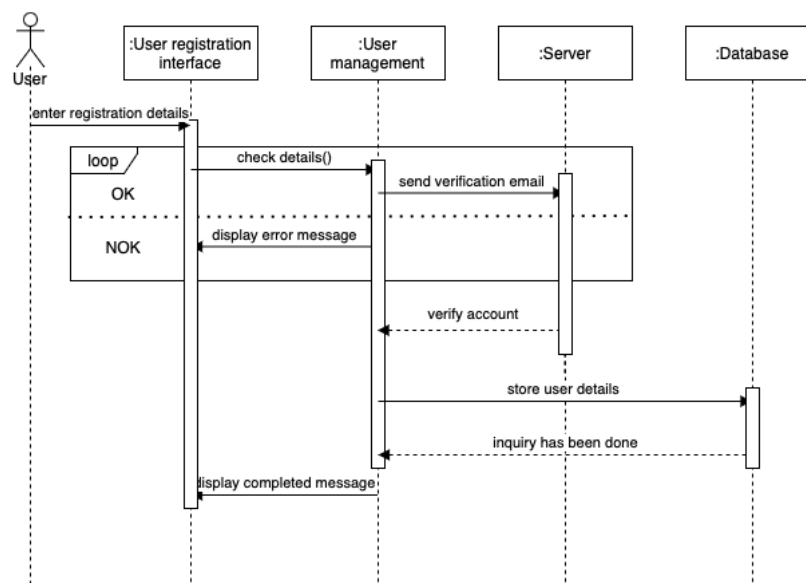
Class diagram:

As stated, new requirements added and not used requirements are removed from the diagram.

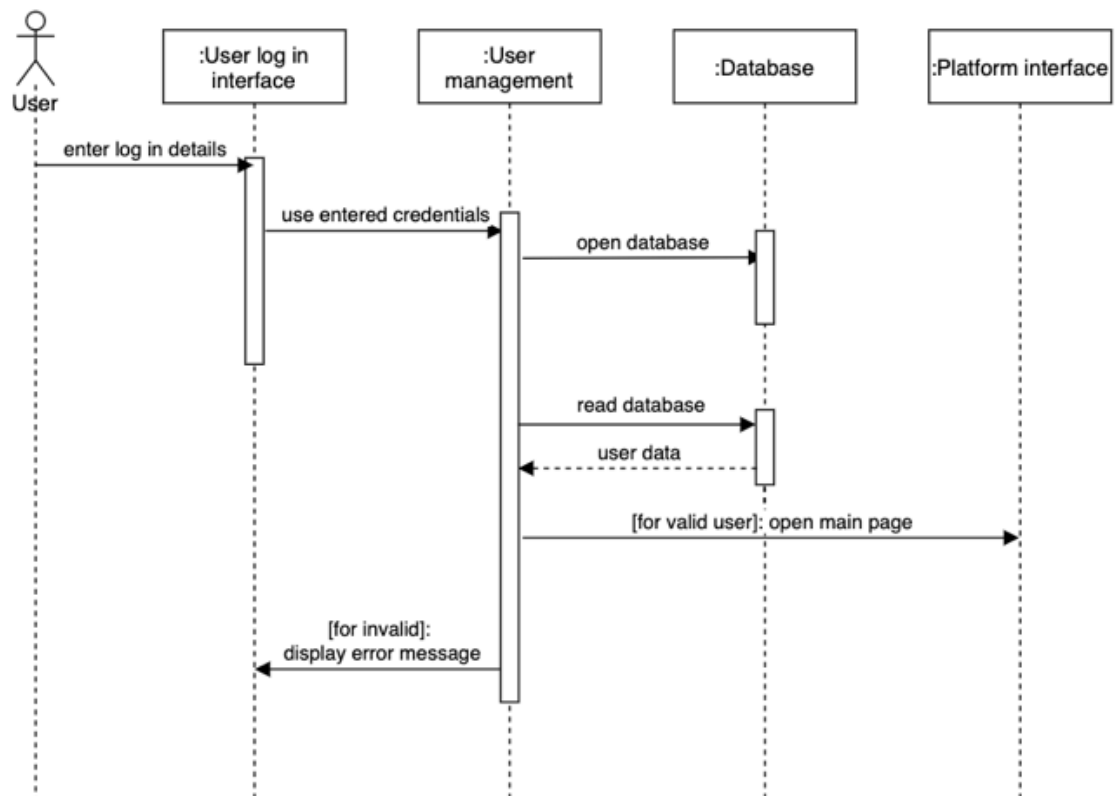


Sequence diagrams:

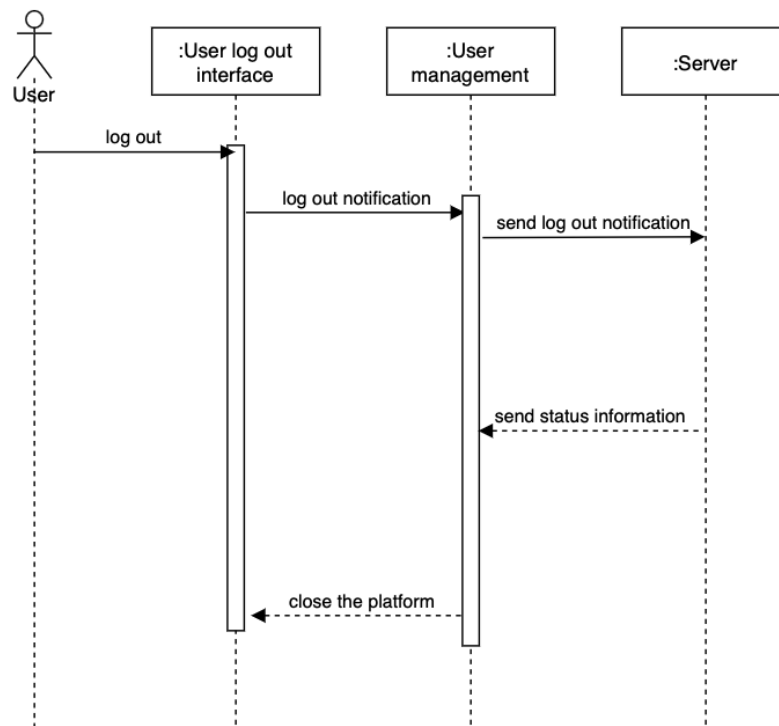
## 1. Registration



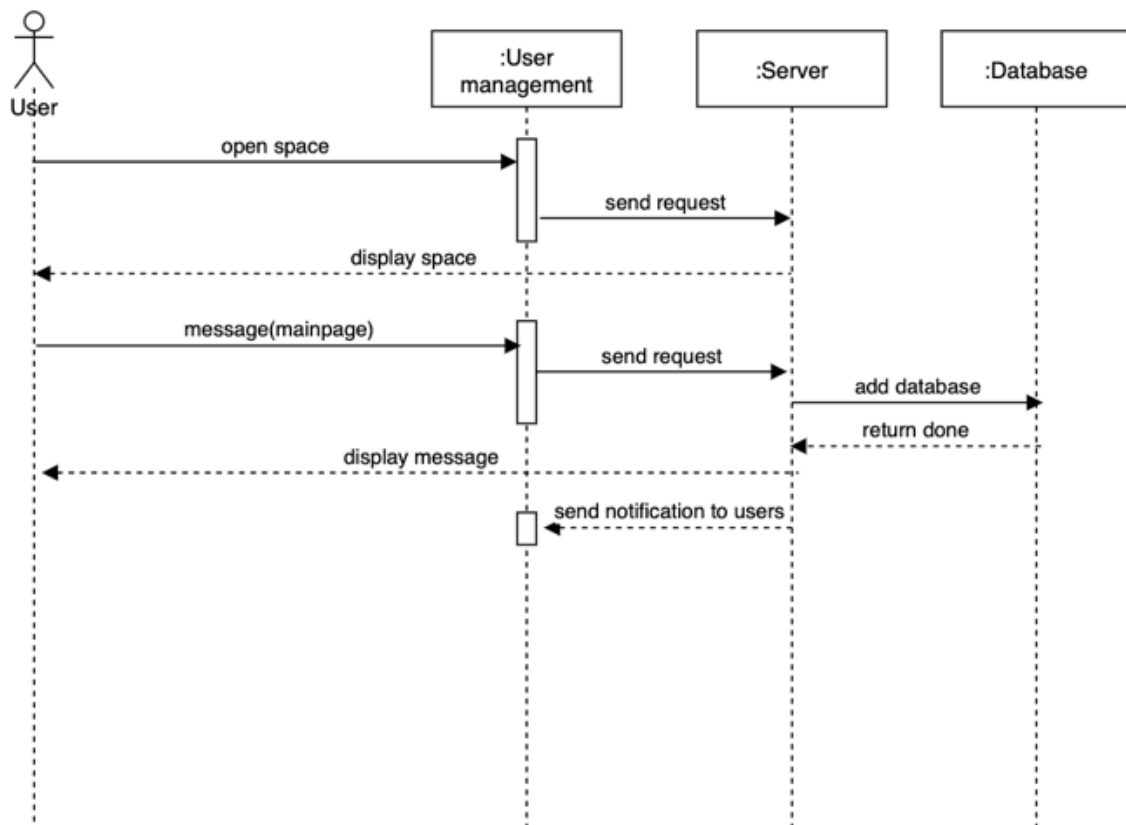
## 2. Log-in



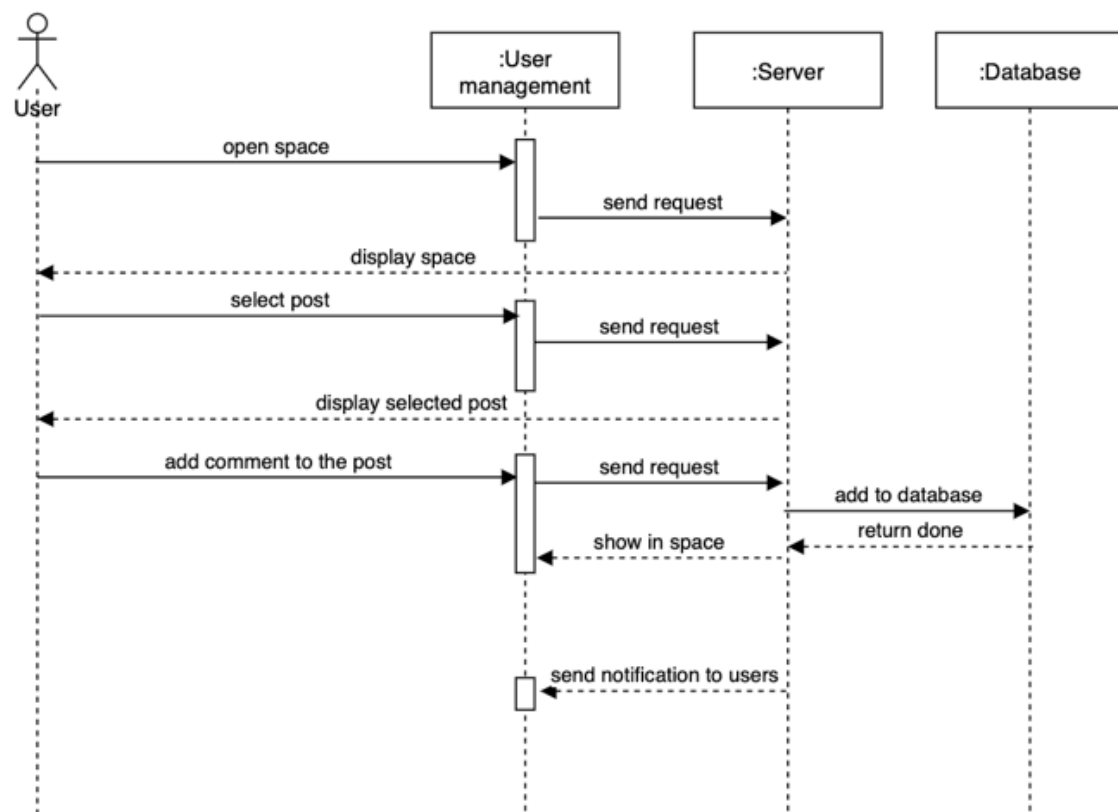
## 3. Log-out



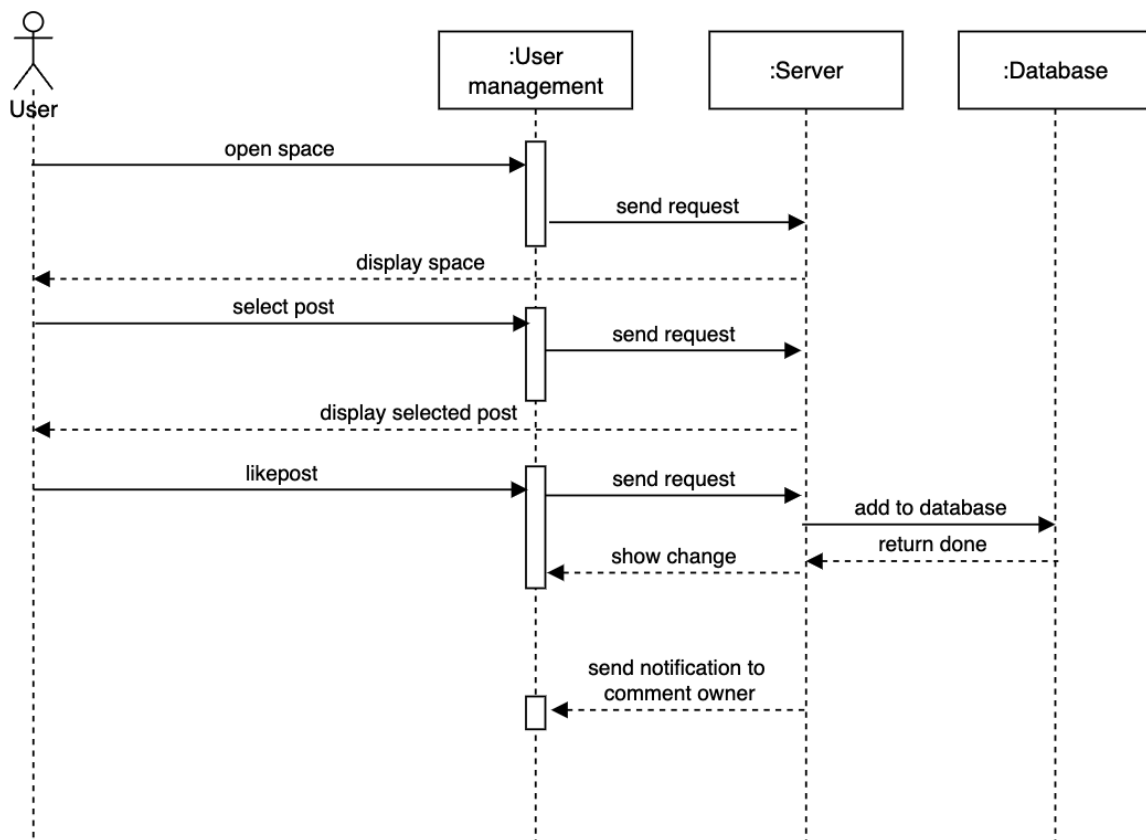
#### 4. Posting new messages



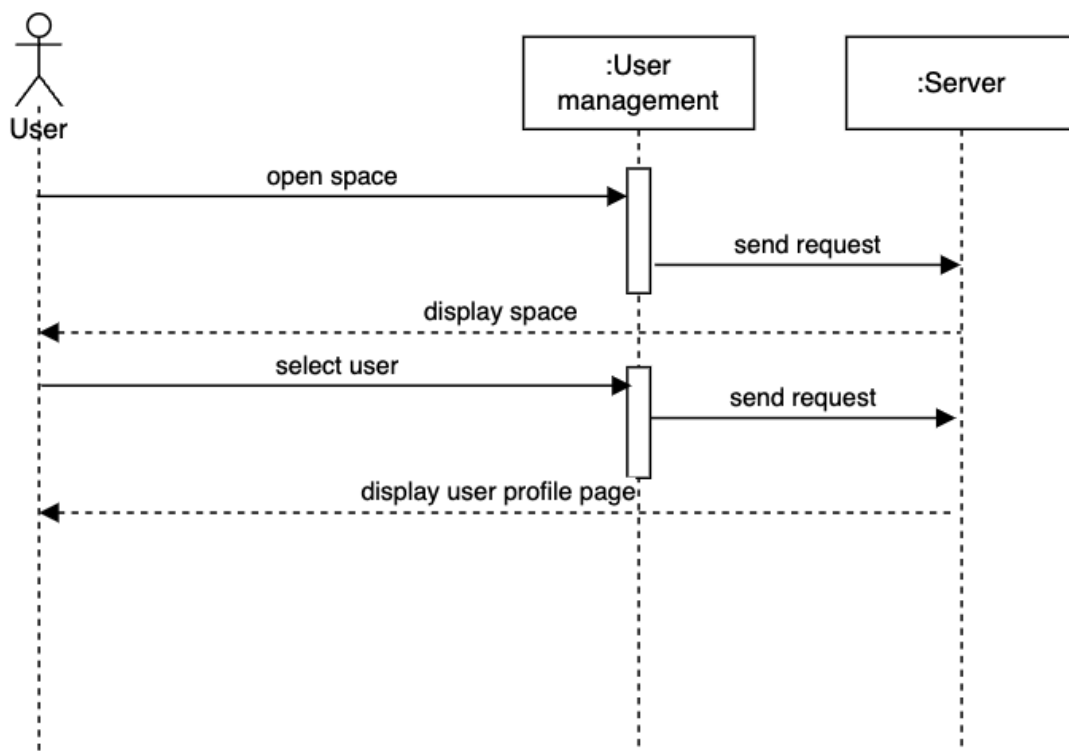
#### 5. Commenting to posts



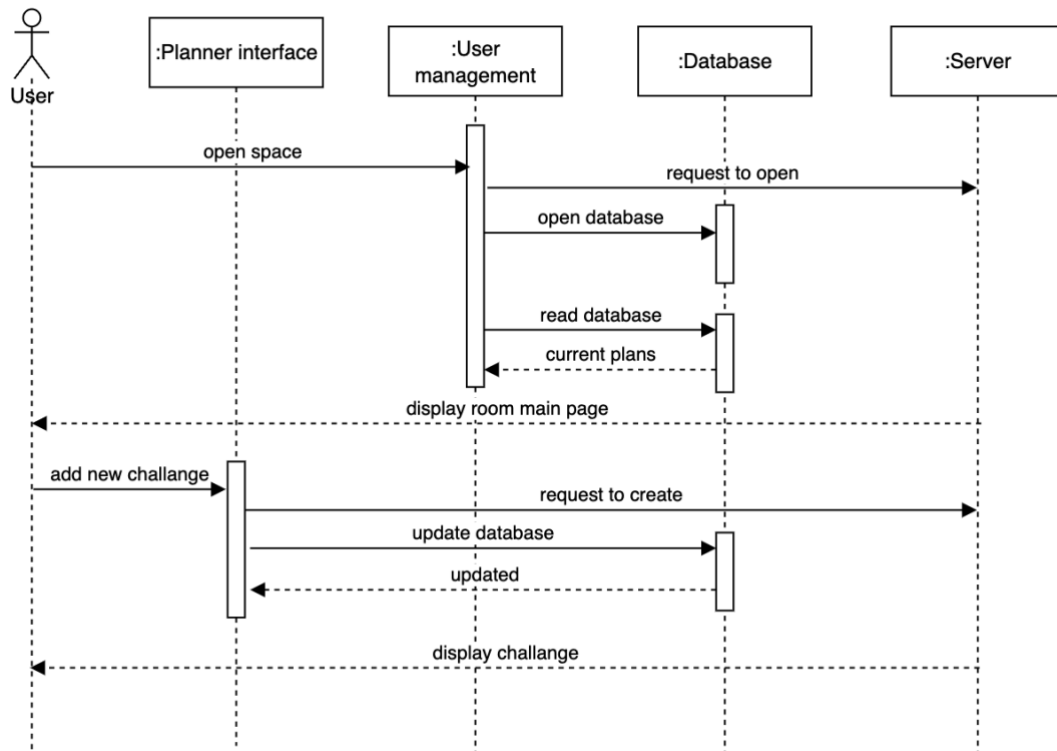
## 6. Liking post



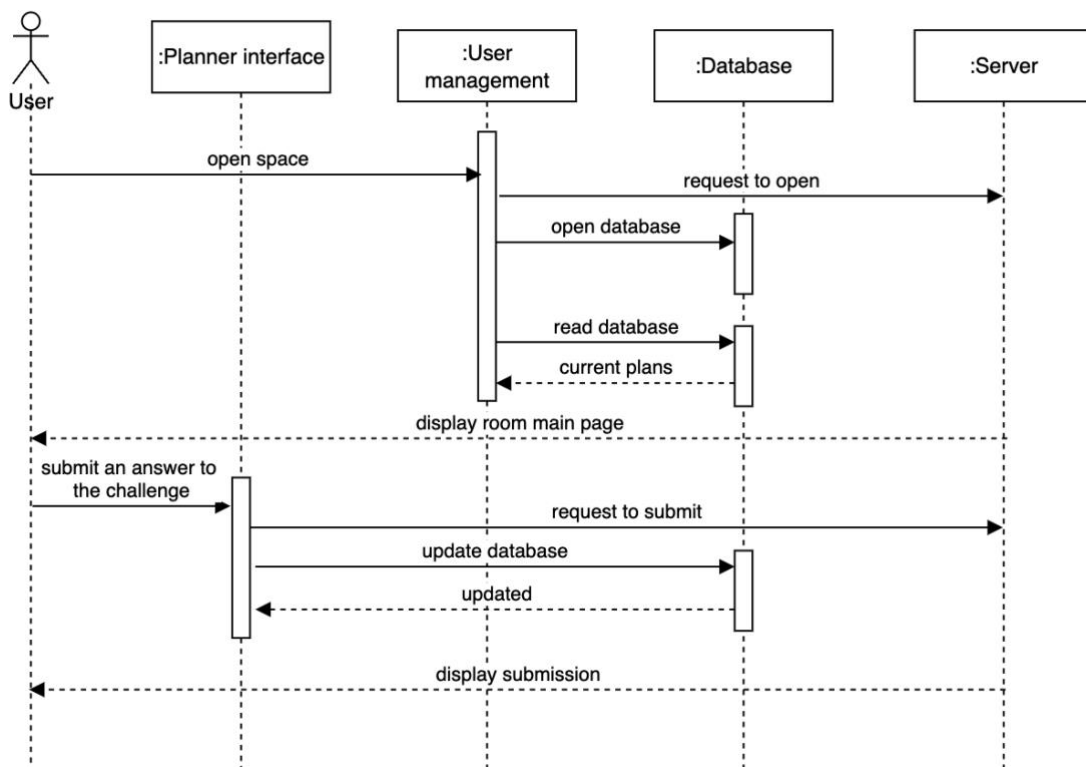
## 7. Displaying user posts



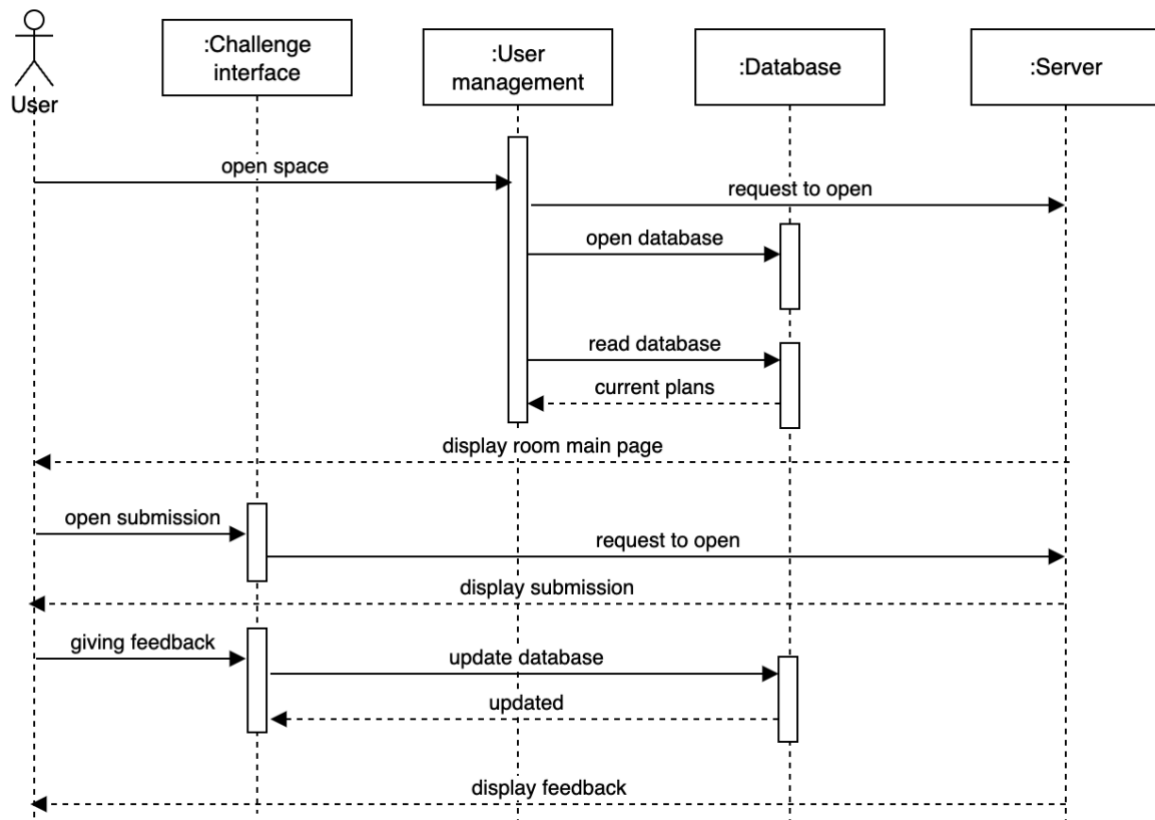
## 8. Creating and displaying challenges



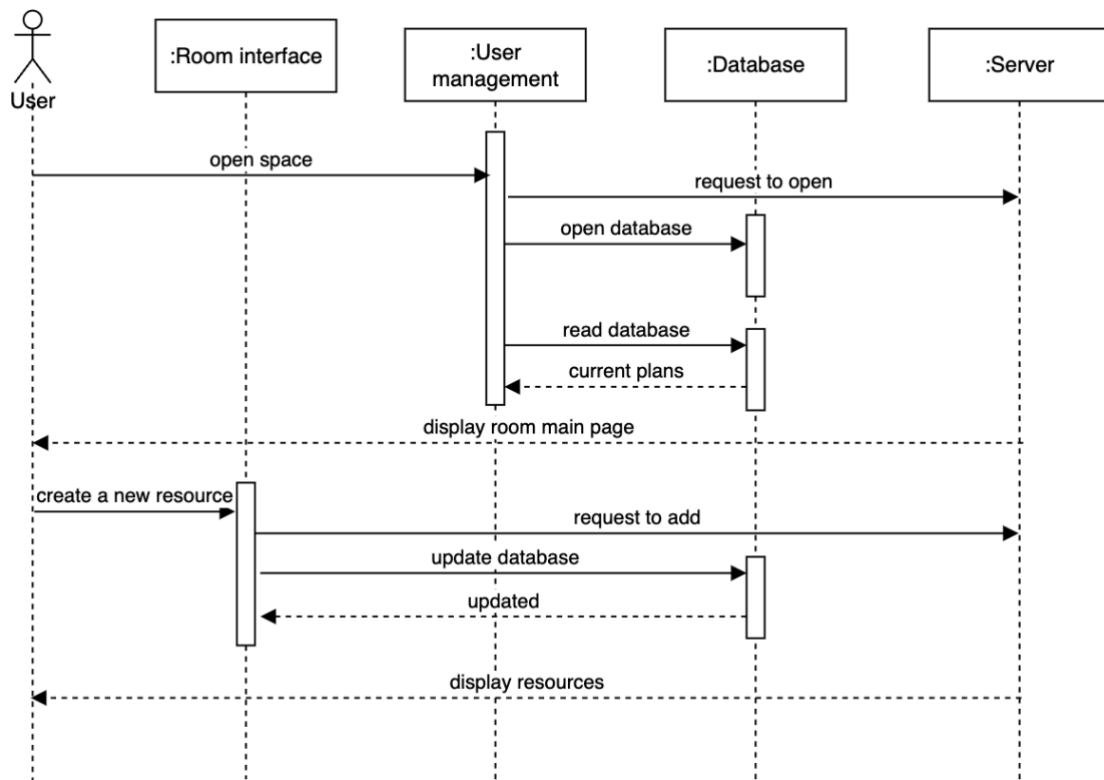
## 9. Submitting an answer to the challenge



## 10. Giving feedback to the challenges



## 11. Adding resources



## Status of Project

As stated before, some requirements could not be able to add in the final application. Also, some functionalities are not work as desired.

- Searching rooms  
This functionality has bugs. The search motor response unlike the logical way.
- Enrolled rooms  
This feature is added to show user the list of rooms that he/she is enrolled. But it returns “None” value, so we cannot see the values.
- Feed page does not show enrolled room’s post, but shows every post, that is sent to rooms.  
Basically, it is related with the previous problem. If I would gather the values on the second problem that I faced, I was going to create if loop to overcome this.
- When user wants to create post, challenge, or room, logically there is no need to select to room again. However, I could not manage to connect these different apps into each other. Thus, user would need to select the room correctly to share information with others.

Beside coding errors above, dockerization is done.

Dockerfile in GitHub:

<https://github.com/turanlioglu/bounsw573-2022/blob/main/Dockerfile>

Docker-compose.yml file in GitHub:

<https://github.com/turanlioglu/bounsw573-2022/blob/main/docker-compose.yml>

## Status of Deployment

For deployment, I firstly, tried out AWS Elasticbeanstalk to understand the functionality of my work. But this platform does not require any dockerization prior. Thus, I changed to use AWS EC2.

Regarding to change, I developed a DockerFile and docker-compose.yml file. In the first step, in my local I did not face any problem to dockerized it. However, in the AWS it returned an error of “***WARNING: Running pip as the 'root' user can result in broken permissions and conflicting behavior with the system package manager.***”. Later, I managed to create a user in the docker file, and with that implementation the bug is fixed.

You can find the Deployed URI:

<http://ec2-34-220-67-213.us-west-2.compute.amazonaws.com>



## System Manual

GitHub page:

The URL of the GitHub repository:

<https://github.com/turanlioglu/bounswe573-2022>

When you click the URL above, you will be directed to the home page of the repository. You may find all the related code in here.

My project consists of several applications, and these are connected inside the database. Inside the requirements.txt page, the required programs can be found.

DockerFile and docker-compose.yml files are contributed with the dockerization.

To run the code in your local:

After you cloned the repository (from: <https://github.com/turanlioglu/bounswe573-2022>), and change the directory of to this downloaded file, you may need to add 127.0.0.1 host into the settings file. Then, you can basically run **python manage.py runserver**, with this command you need to be able to run and expect to open the host (127.0.0.1) in your web browser.

Important note: The debug setting is False currently for the production, you also need to change to True in the settings.py file.

GitHub wiki page:

<https://github.com/turanlioglu/bounswe573-2022/wiki>

In this page, you can find the progress of project in weekly basis and you can see the details by clicking to given links.

GitHub issues page:

<https://github.com/turanlioglu/bounswe573-2022/issues?q=is%3Aopen+is%3Aissue>

There is no open issue now, but some issues are closed with fail. For future work, these problems will be investigated and I plan to manage to fix them all. Thus, you can follow all closed issues:

<https://github.com/turanlioglu/bounswe573-2022/issues?q=is%3Aissue+is%3Aclosed>

GitHub milestones page:

<https://github.com/turanlioglu/bounswe573-2022/milestones>

Each issue is specified to one milestone to have an organized structure to see the project progress in a whole. All the milestones completed for this project, of course, as indicated before there are some issues that needed to look after, but for the final deliverable these are passed.

GitHub project page:

<https://github.com/turanlioglu/bounswe573-2022/projects/1>

In here, as you can see, there are four buckets to track the issues: ToDo, InProgress, Done, Fail. The main idea is to see which issues are needed to do, and which are in under progress etc.

For AWS machine:

With opening secret key folder in the terminal of your pc, which is shared in the zip folder, enter:

```
ssh -i colearn-server-key.pem ec2-user@34.220.67.213
```

This will open the AWS machine in the terminal, then you need to change your home directory with `cd bounswe573-2022/`. This is also the repository name of the project inside the GitHub. After changing directory, you can enter `docker-compose up` command to run the dockerized project in the AWS machine.

## User Manual

The URL of the platform: <http://ec2-34-220-67-213.us-west-2.compute.amazonaws.com>

This link will direct you to the home page of the CoLearn platform and in here, you can create a new account, or you can login with your credentials.

After login, the blog page welcomes you:

<http://ec2-34-220-67-213.us-west-2.compute.amazonaws.com/feed/>

In this page, mainly we can see the posts in related rooms, which are created before. With search functionality you can explore posts with your interest. Later, you can click to the “All rooms” in sidebar. This link will direct you to the rooms page:

<http://ec2-34-220-67-213.us-west-2.compute.amazonaws.com/courses/all/>

Search functionality has bug, so when you want to search a topic, It does not direct you to the findings. This functionality will be worked on it to solve it.

With this page, you can see the created rooms, you can whether enroll or not depending on your interest. In each room, you can see the description, also with “More Info” selection, you can see more details about the room.

After enrollment to the course, you can see the created posts, challenges, and resources inside of the room. If there is any shared information, you can see the related content in the main room page. If not, you may see there is no content description under each section.

With clicking content, it directs you the detail of that shared information. For the sake of collaborative environment, I permit to update or delete the shared information by everyone, just resources are not able to delete or update by everyone, just the owner can update and delete it.

Giving feedback to challenges, every challenge can be graded by feedbacks. This is added to increase the interaction in between users of the room.

## Test Results

**Software Testing** is a method to check whether the actual software product matches expected requirements and to ensure that software product is defect free. It involves execution of software/system components using manual or automated tools to evaluate one or more properties of interest.

My project involves tests as follows:

### 1. Test functions in the code block

The aim is to control the code functionality in deleting, updating of posts, resources, and challenges in the views.py.

### 2. Unit testing

Inside of the blog application, there are several of unit tests. All applied tests are working well. However, there should be testing implementations for all applications inside of the project.

These will be updated for self-learning and indicated as future work for the project. Besides that, my project involves user tests, which is done by myself, and there are some lagging features:

- Searching rooms, this function has a bug that the searched topic is not be found with the system. Most probably, the dictionary is returning empty and does not show any matched values.
- “My rooms”, if condition on the base template gives empty result, which iterates a no information on the UI. Most probably, the condition is not well stated inside the code block.
- “My enrolled rooms”, the database does not hold any string information, but holds integer. This may lead a problem when I want to collect data from the database. Because when I command to pull the data, it is returning “None” value. And by that, I could not be able to apply this into my final version.
- Room (Course) selection, I have revised my code from top to bottom, and figured out that the models.py is working fine. However, in the views.py there is no stating of the post and room, challenge and room, or resource and room.

And I have tried to connect this information together, but I could not to manage to. Thus, the user of the platform should be able to select the rooms to share posts, challenges, or resources.

## Future Work

Throughout the semester, I tried to get maximum benefit from the course and beside classes I watched lots of tutorial, read other external sources.

However, as I can look at to my final work, there is still some failing, even I can see the improvement of myself. Thus, my next aim is to resolve these problems and learn other possible features to add to the web applications.

My roadmap will consist of:

- Stating problems that I faced throughout the semester,
- Have external resources, tutorials to watch how others manage to do,
- Imply all learnings into my project
- And write a basic documentation for myself to improve my understanding, also documentation will help me for other future works.

Later, I want to learn more about GitHub. Because I believe that I did not use all the functionality of GitHub efficiently. So, my second aim will be to search different possible features and learn them all. Also, there will be a documentation too.

Finally, I will improve my understanding on AWS and different tools of the platform. Because, this source will be very important for my general knowledge for the future.