

SWE573 - SOFTWARE DEVELOPMENT PRACTICE - FINAL REPORT

Mehmet Eyüpoğlu

Instructor: Suzan Üsküdarlı

May 19, 2024

TABLE OF CONTENTS

1. Introduction (Title Page)	ii
1.1. HONOR CODE	ii
2. Project Details	iii
2.1. Overview	iii
2.2. Accounts to test the system:	iii
2.3. Design documents	iii
2.4. Status of the project	iii
2.5. Status of Deployment	iii
2.6. Full installation instructions including docker	iv
2.6.1. Client	iv
2.6.2. Server	iv
2.6.3. Environment Variables	iv
2.7. User manual	v
2.7.1. System requirements:	v
2.7.2. Installation	v
2.7.3. Features	v
2.7.3.1. Create a community	v
2.7.3.2. Interact with a community	vi
2.7.3.3. Post	vi
2.7.3.4. Interact with posts	vii
2.7.3.5. Search functionality	vii
2.8. Test results	vii
2.8.1. Test Case: Create a Community on Communiche	vii
2.8.2. Test Case: Create a Post on Communiche	ix
2.8.3. Test Case: Like and Comment Functionality on a Post	xi
2.9. Link to demo video	xii
REFERENCES	xiv

1. Introduction (Title Page)

You will find the main information about the course and the project.

Name	Mehmet Eyupoglu
Course	SWE573 - Software Development Practice
Date	May 20, 2024
Project Name	https://communique.vercel.app
Git Rep. URL	https://github.com/mehmeteyupoglu/Swe573-eyupoglu
Git Tag V. URL	https://github.com/mehmeteyupoglu/Swe573-eyupoglu/releases/tag/v0.0.1

1.1. HONOR CODE

Related to the submission of all the project deliverables for the Swe573 2024 Spring semester project reported in this report, I Mehmet Eyupoglu declare that: - I am a student in the Software Engineering MS program at Bogazici University and am registered for Swe573 course during the Spring 2024 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.

Mehmet Eyupoglu



2. Project Details

2.1. Overview

2.2. Accounts to test the system:

- username: test1234
- password: test1234

2.3. Design documents

Link to design documents in the wiki of the project:

<https://github.com/mehmeteyupoglu/Swe573-eyupoglu/wiki/6>

2.4. Status of the project

Link to status of the project in the wiki :

<https://github.com/mehmeteyupoglu/Swe573-eyupoglu/wiki/8>

2.5. Status of Deployment

Since the application is constructed on a typical client-server architecture, two separate deployments are configured.

- Client* is deployed as a Vercel project
- Server* (api) that serves the data is on an AWS EC2 instance and could be accessed at <https://www.kodigate.com>

Only the server needs to be dockerized. Therefore, docker container is kept and maintained for the server.

2.6. Full installation instructions including docker

Be mindful of the fact that the whole application is utilizing the client-server architecture. A separate installation for each folder in the repository (client and server) is needed.

2.6.1. Client

```
cd client
npm install
npm run start
```

2.6.2. Server

```
cd server
docker-compose build
docker-composer up
```

This will run your docker application in the server.

2.6.3. Environment Variables

Please check the *.env* files submitted through Moodle.

2.7. User manual

2.7.1. System requirements:

Make sure:

- You own a laptop or a desktop computer.
- You have one of the following operating systems installed on your device: Windows, macOS, or Linux.
- You have access to a web browser and are connected to the internet.
- Your internet download speed should be higher than 10 Mbps (Megabits per second) for a better user experience.

2.7.2. Installation

If you plan to install the application on your local machine, please find the github repo in this link:

<https://github.com/mehmeteyupoglu/Swe573-eyupoglu>

clone it to your local machine and follow the installation specifications shared above. Otherwise, go to <https://communiche.vercel.app> and login with the given *username* and *password*.

2.7.3. Features

2.7.3.1. Create a community.

- In the navigation bar, click on the create community button
- After you are directed to the create community page, you can fill in the inputs provided and create button to create the community

- Once it's created, you can enter more templates to your community, making it more peculiar

2.7.3.2. Interact with a community.

- Click on the communities button on the navigation bar
- Once you are in the communities page and all the communities are visible, you can click on the view icon and see the community details
- If the community is private, you can send a request to join the community by clicking on the request to join .
- Your requests will be responded by the moderator or the owner of that community.
- If the community is public, you can simply click on the join button.
- If you are the owner or the moderator of the community, you can invite other people by clicking on the invite tab item
- Once you are in the invite tab, search for the user
- Once you find the user you are searching for, you can invite the person by clicking on the invite button
- If you are invited to a community, you can respond to the invitations in <https://communiche.vercel.app/app/account/settings/myInvitations>
- You can find this page by clicking on the profile avatar and select *My Invitations* from the opened dropdown
- If you are the owner of a community, you can assign a moderator or remove a member

2.7.3.3. Post.

- Once you visit a community, you can post under that community. Post must be related with a community
- Once you click on the post button in community detail page, you will be directed to a path `/community/:communityId/post`

- Once you are in the post page, you can select a template amongst the available community-specific templates and fill in the inputs and click on post button to complete the post action.

2.7.3.4. Interact with posts.

- You will be directed to home page once you click on home page button on the navigation bar
- Once you are in the home page, you can see the posts listed by the date they are posted.
- Choose the one and you will be able to see them in detail in */post/:postid*
- You will see that there is a thumb-up icon for the user to either like or unlike. There is no dislike button.
- Right next to the thumb-up icon, there is a comment icon to list all the comments
- If you would like to comment on a post, you can type anything you like in the comment section. Once you are done, you can click on the comment button to send the comment.

2.7.3.5. Search functionality.

- You can search communities, posts or users. You can simply select one of them in the provided filter area and query. For posts, additional data type filtering is available.

2.8. Test results

2.8.1. Test Case: Create a Community on Communiche

Test Case ID: TC01_Create_Community

Description: This test case verifies the functionality of creating a new commu-

nity on the Communiche platform.

Pre-conditions:

- User has a registered account on Communiche with username "test1234" and password "test1234".
- User is logged in to the platform.
- User is automatically redirected to URL: <https://communiche.vercel.app/home>

Steps:

(i) **Navigate to Create Community Page:**

- Click on the "Create Community" button located on the navigation bar.
- **Expected Result:** The user is redirected to the "/create-community" path displaying the Create Community form.

(ii) **Fill in the Community Details:**

- Enter a unique community name in the "Name" field.
- Enter a descriptive community description in the "Description" field.
- Select appropriate options from any other relevant form fields (e.g., privacy settings).
- **Expected Result:** All form fields accept user input and display the entered data.

(iii) **Submit the Community Creation:**

- Click on the "Create" button.
- **Expected Result 1:** A success notification appears via a toaster message (e.g., "Community created successfully!").
- **Expected Result 2:** The user is automatically redirected to the "/communities" path displaying the list of communities, including the newly created one.

Post-conditions:

- The newly created community should be listed on the "/communities" page.
- The user should be able to access and manage the newly created community as an owner

Pass/Fail Criteria:

- **Pass:**
 - All steps are executed successfully as described in the expected results.
 - The new community is created and displayed on the "/communities" page.
- **Fail:**
 - Any step encounters an error or unexpected behavior.
 - The new community is not created or is not accessible.

2.8.2. Test Case: Create a Post on Communiche

Test Case ID: TC02.Create.Post

Description: This test case verifies the functionality of creating a new post within a specific community on the Communiche platform.

Pre-conditions:

- User has a registered account on Communiche with username "test1234" and password "test1234".
- User has access to an existing community on the platform (you can join a public community for this test).

Steps:

(i) **Navigate to the Community Page:**

- Open the Communiche application URL: <https://communiche.vercel.app>

- Locate the desired community and navigate to its dedicated page.
- **Expected Result:** The user lands on the specific community page displaying existing posts (if any).

(ii) **Access the Post Creation Feature:**

- Locate the button or link designated for creating a new post within the community (e.g., "Create Post," "+ New Post").
- Click on the button/link to access the post creation interface.
- **Expected Result:** The user is directed to a dedicated post creation interface within the community page.

(iii) **Compose and Submit the Post:**

- Choose the template available in the community
- Enter a *title* and *description* (default values) for the post in the designated field.
- Enter the content for the post body (text, images, or attachments as allowed by the platform).
- (Optional) Add relevant tags or categories to the post (if available).
- Click on the or "Submit" button to create the post.
- **Expected Result 1:** The post is successfully created and displayed within the community page, showcasing the title, content, and any additional elements (tags, attachments).
- **Expected Result 2:** An optional success notification might appear (e.g., "Post created successfully!").

Post-conditions:

- The newly created post should be visible on the community page.
- The user should be able to view and manage the created post (edit, delete, etc., based on permissions).

Pass/Fail Criteria:

- **Pass:**
 - All steps are executed successfully as described in the expected results.
 - The new post is created, displayed correctly, and accessible within the community page.
- **Fail:**
 - Any step encounters an error or unexpected behavior.
 - The new post is not created or is not displayed as expected.

2.8.3. Test Case: Like and Comment Functionality on a Post

Test Case ID: TC03_Like_and_Comment_Post

Description: This test case verifies the functionality of liking and commenting on a post within a community on the Communiche platform.

Pre-conditions:

- User has a registered account on Communiche with username "test1234" and password "test1234".
- User has access to an existing community on the platform with at least one post (you can join a public community with existing posts for this test).

Steps:

(i) **Navigate to a Post:**

- Open the Communiche application URL: <https://communiche.vercel.app>
- Locate the desired community and navigate to its dedicated page displaying existing posts.
- Identify a specific post for testing like and comment functionalities.
- **Expected Result:** The user lands on the community page with the chosen post displayed.

(ii) **Like the Post (Optional):**

- Locate the "Like" button or icon associated with the chosen post.
- Click on the "Like" button/icon.
- **Expected Result 1:** The post is visually updated to indicate it is liked by the user (e.g., change button color, like counter increment).

(iii) **Leave a Comment:**

- Locate the comment section or designated area for adding comments to the post.
- Enter a comment message in the text field provided.
- Click on the "Comment" or "Submit" button to submit the comment.
- **Expected Result 1:** The user's comment is successfully added and displayed beneath the post, showing the username and comment content.

Post-conditions:

- The post should be updated to reflect the user's like (if applicable).
- The user's comment should be displayed beneath the post.

Pass/Fail Criteria:

- **Pass:**
 - All steps are executed successfully as described in the expected results.
 - The user can like the post (visually and functionally) and leave a comment that appears correctly.
- **Fail:**
 - Any step encounters an error or unexpected behavior.
 - The like functionality doesn't work, or the comment is not submitted/displayed.

2.9. Link to demo video

You can find the demo in this link:

[https://drive.google.com/file/d/1Ihd55₄WoepgITYs37YKkVHVMS1jBuJ/view?usp =
sharing](https://drive.google.com/file/d/1Ihd55₄WoepgITYs37YKkVHVMS1jBuJ/view?usp=sharing)

REFERENCES