BOĞAZİÇİ UNIVERSITY
SWE573 – Final Project Report

What is This Thing? Application

Mustafa Bektaş

21/12/2024

Deployment Link

Git Repository

Git Tag Version

HONOR CODE

Related to the submission of all the project deliverables for the Swe573 Fall 2024 semester project reported in this report, I, Mustafa Bektaş, declare that:

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

Mustafa Bektaş

Contents

1. Project Overview

This application was designed to help users identify mysterious objects via community discussions. Users can create posts which describe the mysterious object in detail and other users can interact with the post by comments/replies and upvotes/downvotes. The application also has a search function which searches for post titles, descriptions and tags for keywords.

Key Features:

- Create and manage posts with images and detailed descriptions.
- Dynamically fetch tags from Wikidata during post creation.
- Comment, reply and vote on posts.
- Receive notifications for actions such as comments, upvotes and best answers.
- Search for posts using keywords in titles, descriptions or tags.
- Profile page for displaying user posts and comments.

2. Software Requirements Specifications

2.1. User Authentication

- Enable user registration via email.
- Provide a secure login system using passwords.

2.2. Posting System

- Allow users to create posts with text descriptions and image uploads.
- Include options to categorize posts (e.g., object, plant, tool) and add relevant tags.

2.3. Commenting System

- Support threaded comments with upvote/downvote functionality.
- Allow users to edit or delete their own comments.

2.4. Voting Mechanism

- Implement an upvote/downvote system for posts and comments.
- Display the most upvoted comment at the top of the comment section.

2.5. Search and Filtering

- Enable full-text search across all posts.
- Offer filtering options by categories, tags, popularity, and recency.

2.6. Profile Management

- Provide user profile pages that display their posts and comments.
- Ensure all profiles are public to allow interaction and evaluation by the community.

2.7. Notifications

- Send notifications for new comments, upvotes, or when a comment is marked as the best answer.

2.8. Best Answer System

- Allow the post creator to mark a comment as the best answer.
- Display a visual indicator to highlight the best answer selected.

3. Design Documents

3.1 Entity-Relationship Diagram

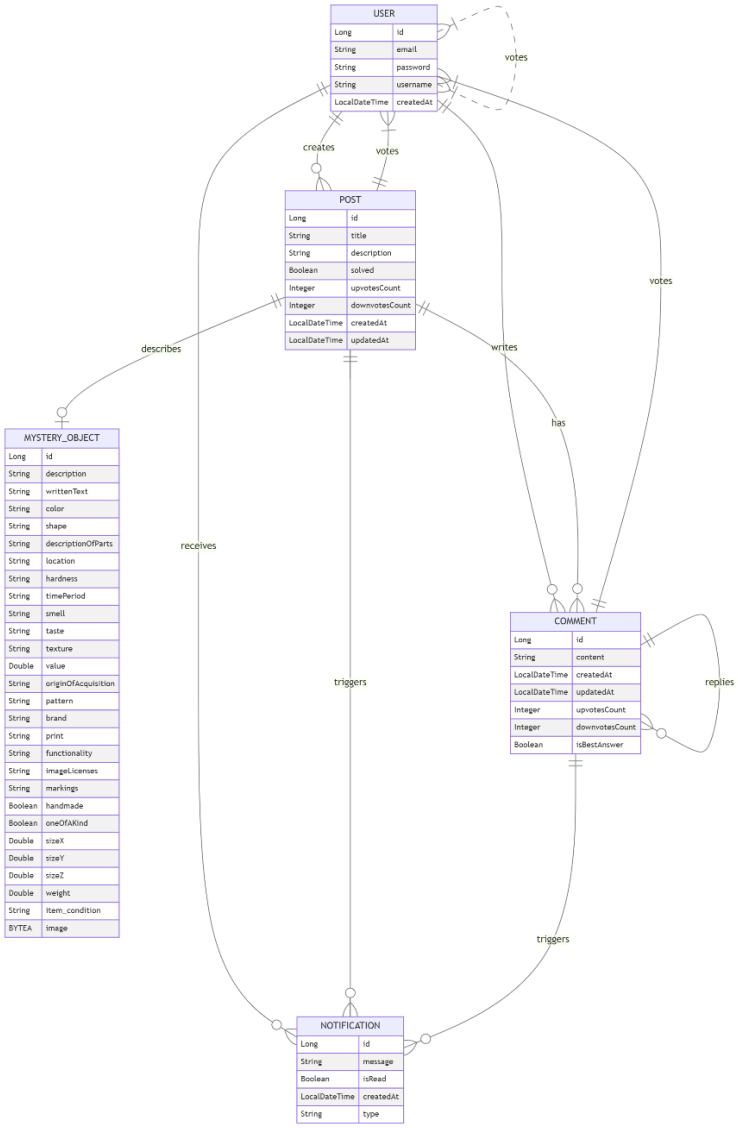


Figure 3.1 – Entity-Relationship Diagram.

Key Relationships:

1. **User and Post** – A user can create multiple posts.
2. **User and Comment** – A user can write multiple comments.
3. **User and Notification** – A user can receive multiple notifications.
4. **Post and Mystery Object** – Each post can have one associated mystery object.
5. **Post and Comment** – Each post can have multiple comments.
6. **Comment and Replies** – Comments can have nested replies.
7. **Votes** – Both posts and comments can have upvotes and downvotes from users.
8. **Notifications** – Notifications are triggered by actions related to comments, upvotes, and best answers.

3.2 Sequence Diagrams

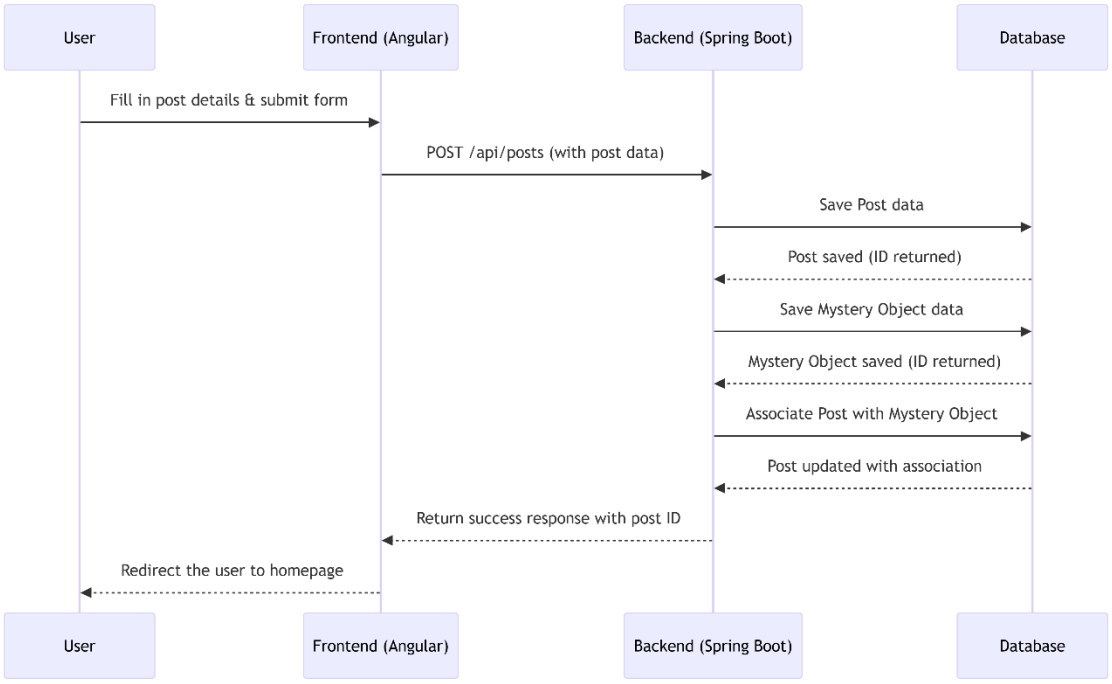


Figure 3.2 – Sequence diagram for creating a post.

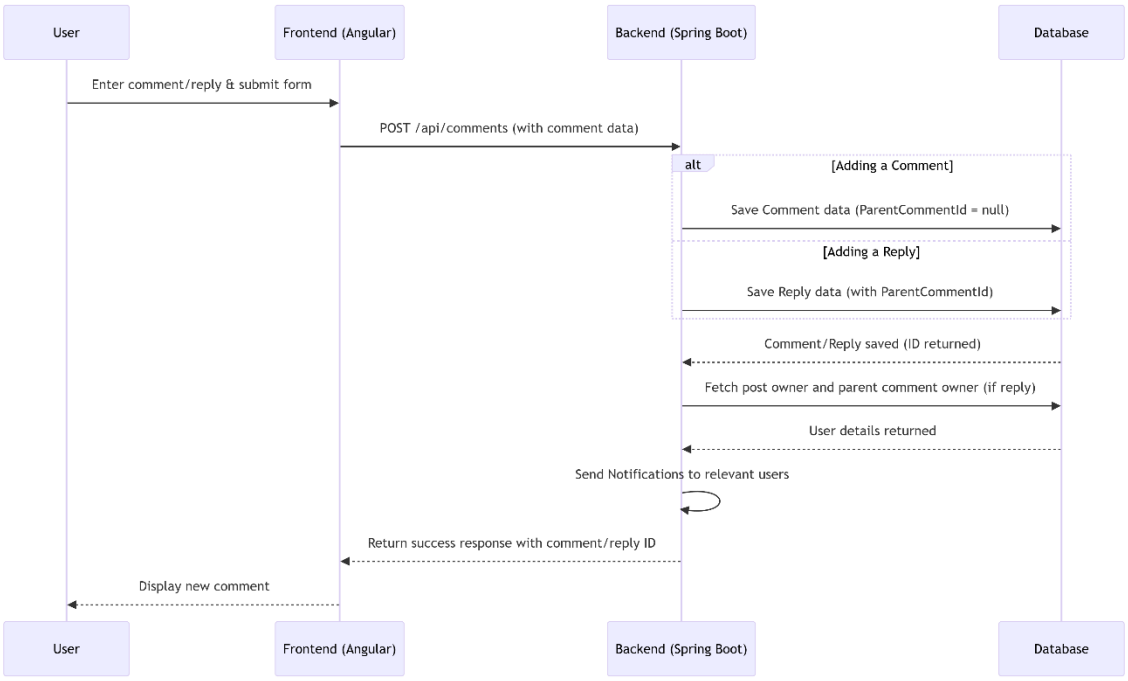


Figure 3.3 - Sequence diagram for creating a comment/reply.

4. Project Status

Requirement	Status
Enable user registration via email	Completed
Provide a secure login system using passwords	Completed
Allow users to create posts with text descriptions and image uploads	Completed
Include options to categorize posts and add relevant tags	Partial – Users can add tags to posts but there is no post categorization option
Support threaded comments with upvote/downvote functionality	Completed
Allow users to edit or delete their own comments	Not Completed – No edit or delete function for comments
Implement an upvote/downvote system for posts and comments	Completed
Display the most upvoted comment at the top of the comment section	Not Completed/Ignored – Best comment is shown at top
Enable full-text search across all posts	Completed
Offer filtering options to search results by categories, tags, popularity, and recency	Partial – Search results are automatically sorted by recency
Provide user profile pages that display their posts and comments	Completed
Ensure all profiles are public to allow interaction and evaluation by the community	Completed
Send notifications for new comments, upvotes, or best answers	Completed
Allow the post creator to mark a comment as the best answer	Completed
Display a visual indicator to highlight the best answer selected	Completed

5. Status of Deployment

The application is fully dockerized and deployed on cloud. The database runs on AWS RDS, and both backend and frontend applications run on Google Cloud. The deployment link is:

https://swe573-frontend-594781402587.us-central1.run.app/

6. Installation Instructions

- Make sure you have Docker installed on your computer
- Clone the repository to a local folder using:

  ```
  git clone https://github.com/mustafa-bektas/SWE573.git
  ```

- Go to the root repository folder and run:

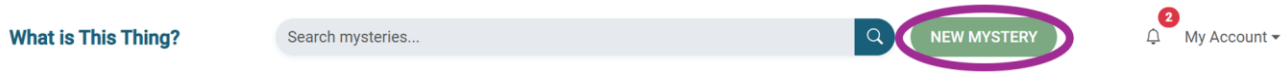
  ```
  docker-compose up --build
  ```

- Wait for all the packages to be installed and deployed
- After everything finishes, visit `http://localhost:4200/` to use the application. It will still be connected to the same database on the cloud, so you don't need to run the database locally.

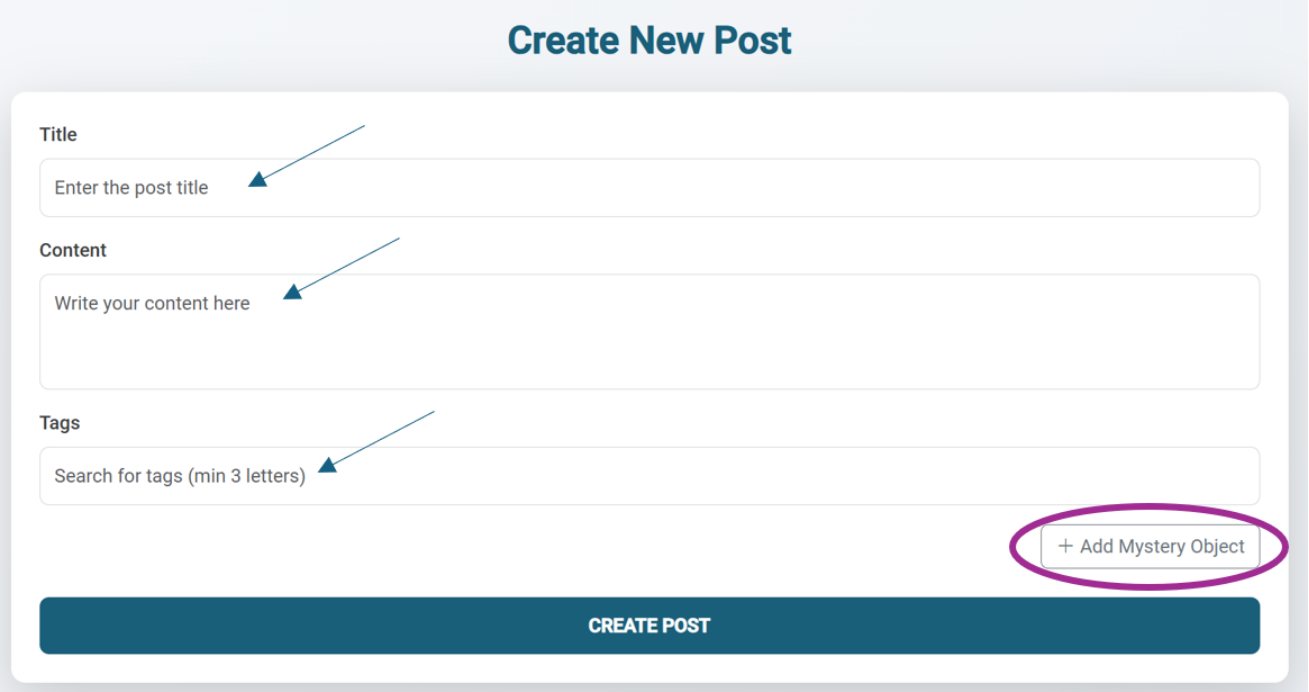
7. User Manual

7.1 Creating a Post

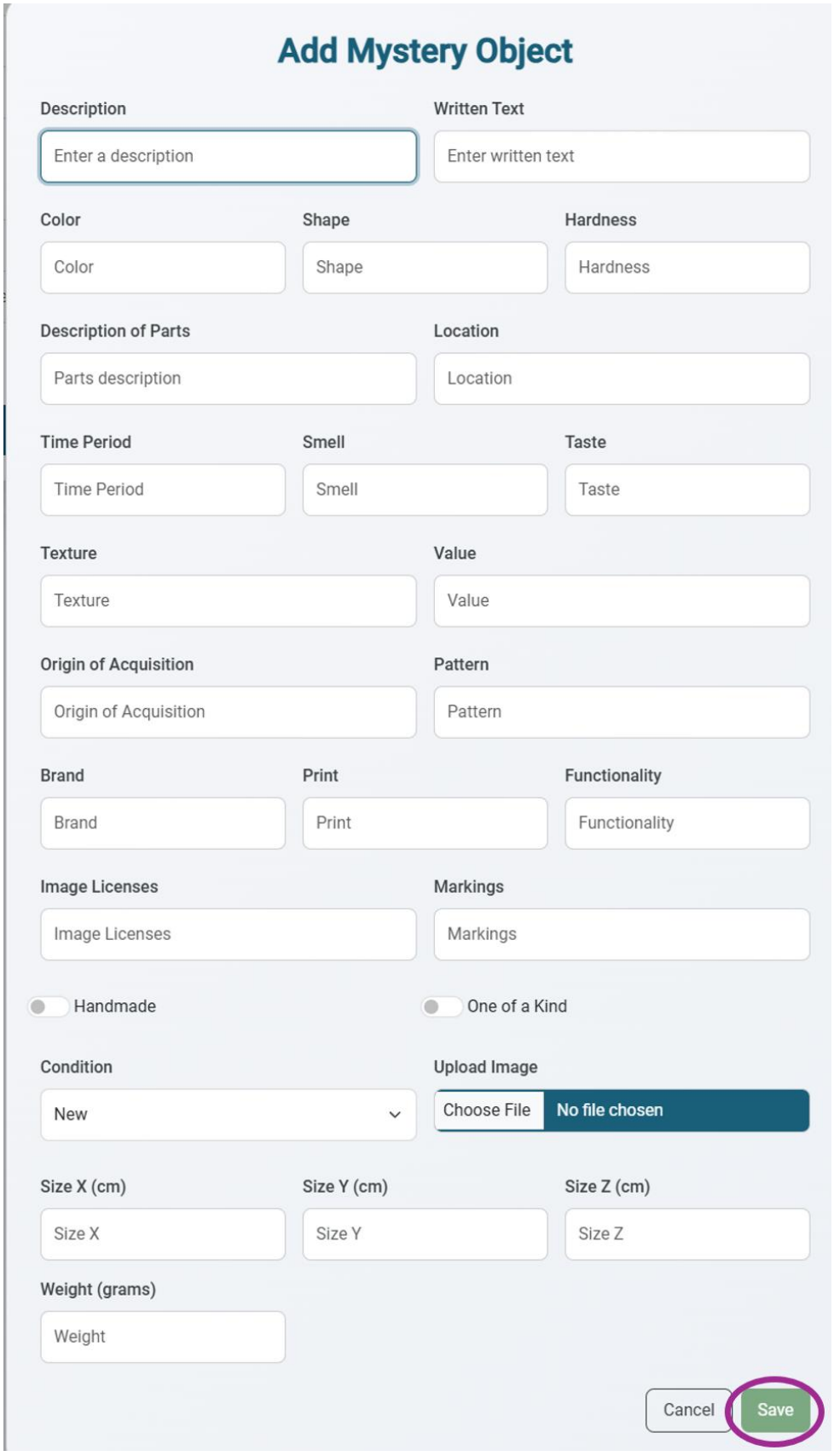
- After logging in, click the NEW MYSTERY button.



- After filling the post details, click the "+ Add Mystery Object" button.

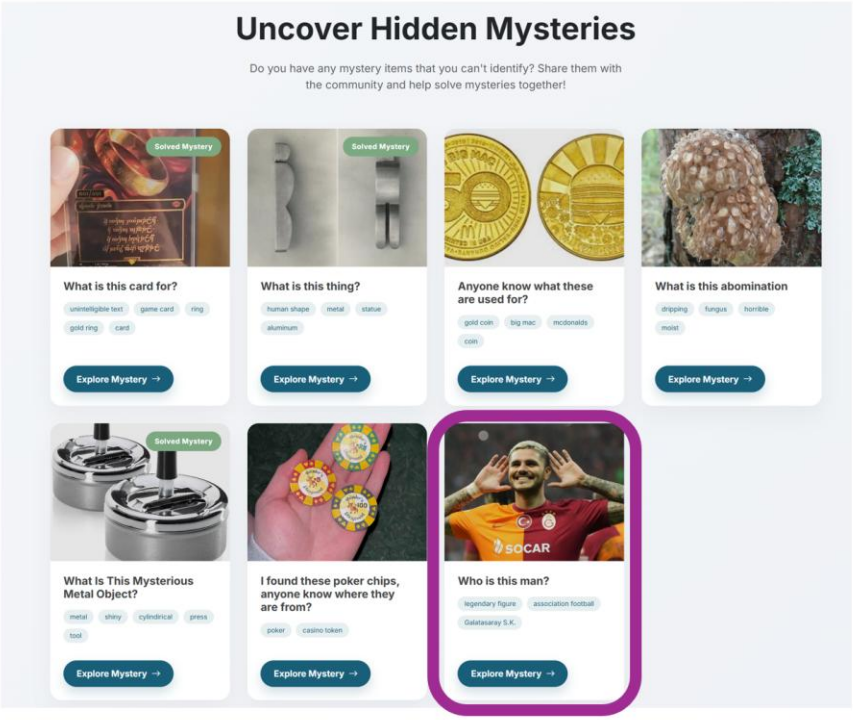


- Fill all necessary details about the mystery object and press “Save”



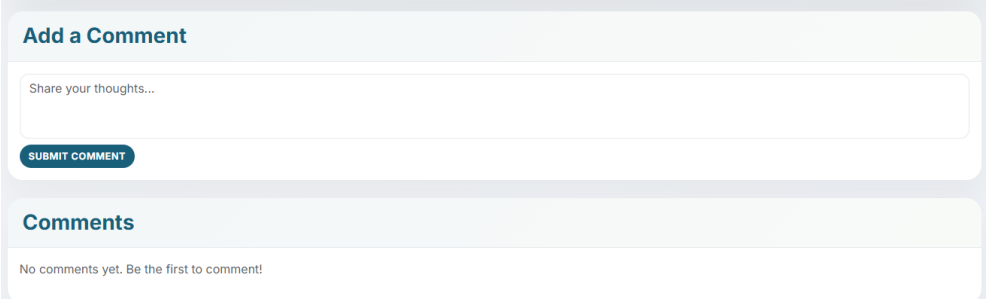
- Preview the added mystery object below. Click “+ Edit Mystery Object” for any modifications. If ready, press “Create Post” to create the post.

- You will be redirected to the home page. See the created post in here.

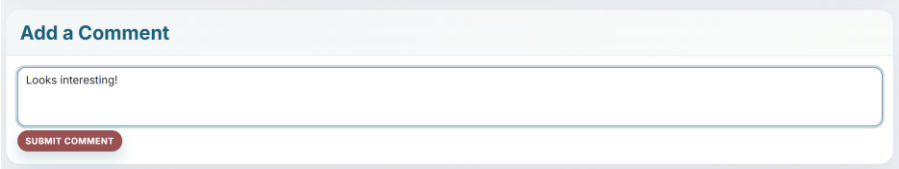


7.2 Creating a Comment / Reply

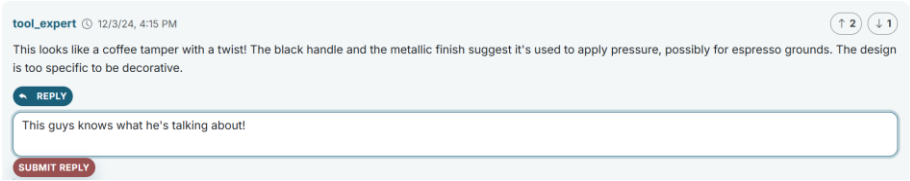
- Go to any post page and locate the comment section at the bottom.



- Use the “Add a Comment” box to create a comment and click “Submit Comment” to submit.

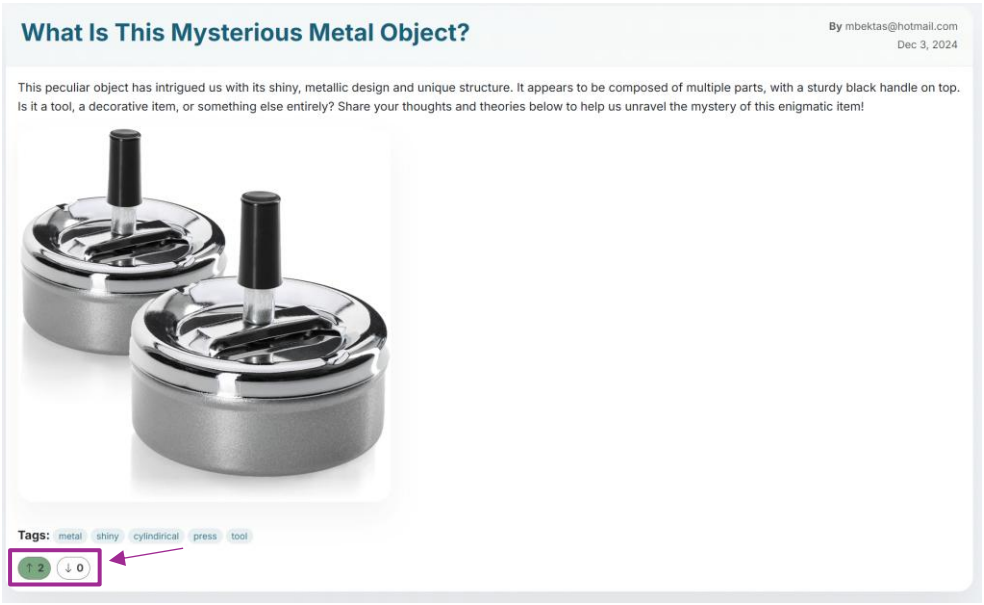


- To reply to a comment, locate the desired comment and click the “Reply” button. After that, enter your reply to the text box that appears and click “Submit Reply”



7.3 Upvoting / Downvoting

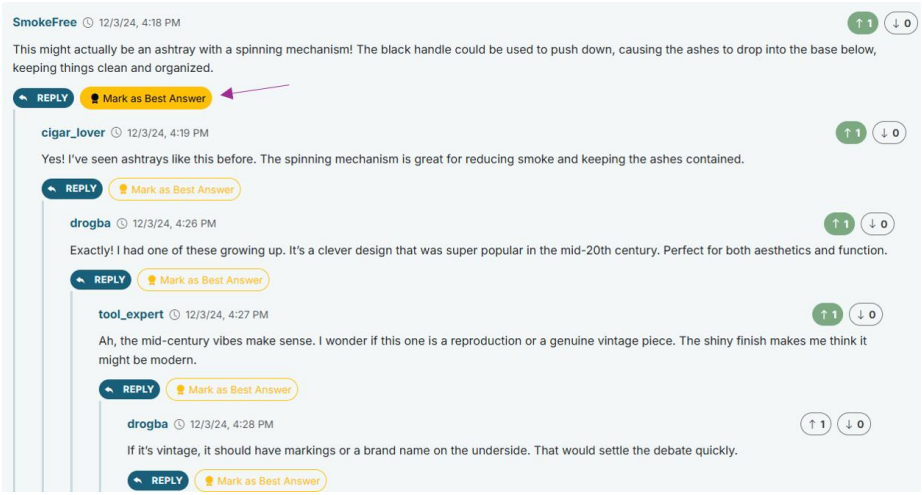
- To upvote or downvote a post, click on the respective button just below the post tags in the post details page.



- To upvote or downvote a comment, click on the respective button just next to the comment.

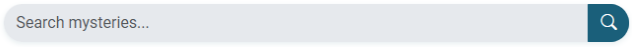
7.4 Marking the Best Answer

- As a post author, click on the “Mark as Best Answer" button to mark a comment as the best answer of a post. The best answer will always appear on top.

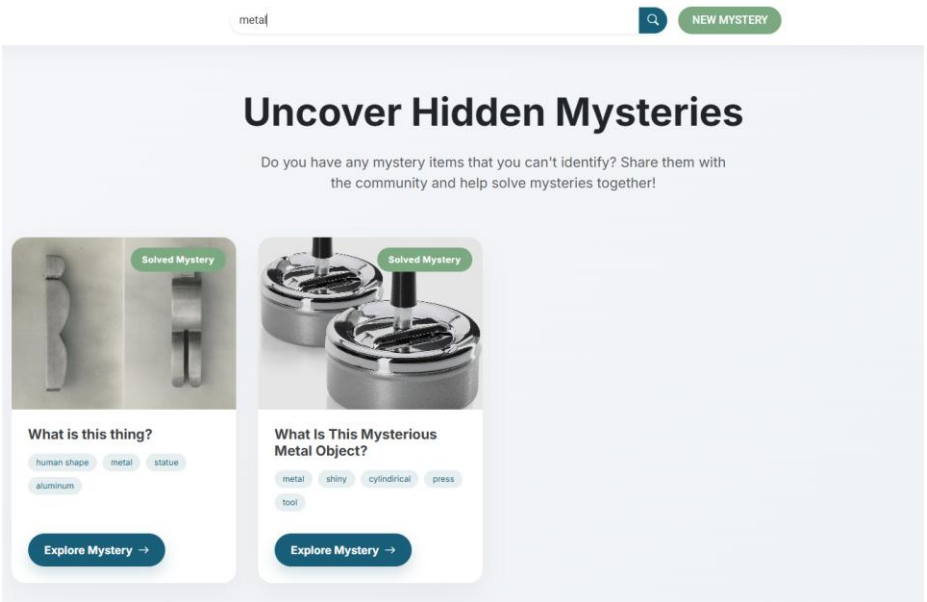


7.5 Searching

- Use the search bar to search keywords in post titles, descriptions or tags.



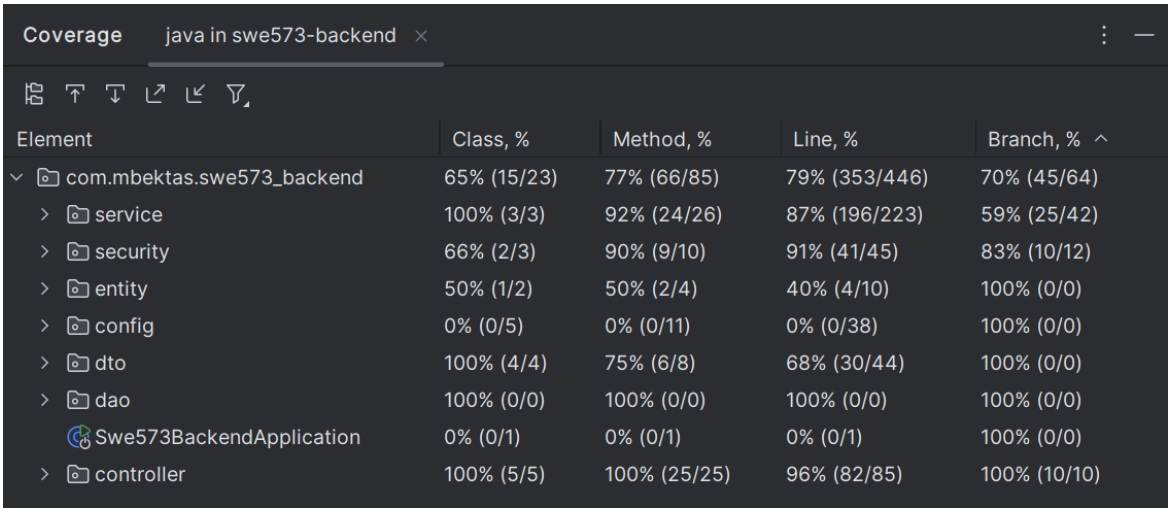
- Search results will show up sorted by recency.



8. Test Results

8.1 Unit Tests

A total of 59 unit tests were performed for the backend services. All of these tests pass when run, and overall they have 70% branch coverage.



Element	Class, %	Method, %	Line, %	Branch, %
com.mbektas.swe573_backend	65% (15/23)	77% (66/85)	79% (353/446)	70% (45/64)
service	100% (3/3)	92% (24/26)	87% (196/223)	59% (25/42)
security	66% (2/3)	90% (9/10)	91% (41/45)	83% (10/12)
entity	50% (1/2)	50% (2/4)	40% (4/10)	100% (0/0)
config	0% (0/5)	0% (0/11)	0% (0/38)	100% (0/0)
dto	100% (4/4)	75% (6/8)	68% (30/44)	100% (0/0)
dao	100% (0/0)	100% (0/0)	100% (0/0)	100% (0/0)
Swe573BackendApplication	0% (0/1)	0% (0/1)	0% (0/1)	100% (0/0)
controller	100% (5/5)	100% (25/25)	96% (82/85)	100% (10/10)

Figure 8.1 – Class, method, line and branch coverage of the unit tests

A report showing the results of the unit tests can be found in the project repository at:

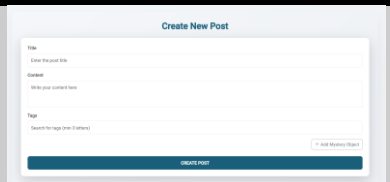
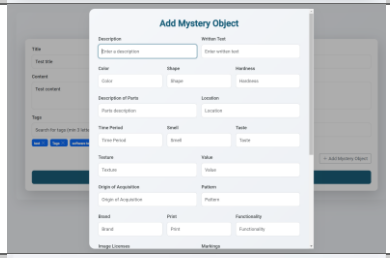
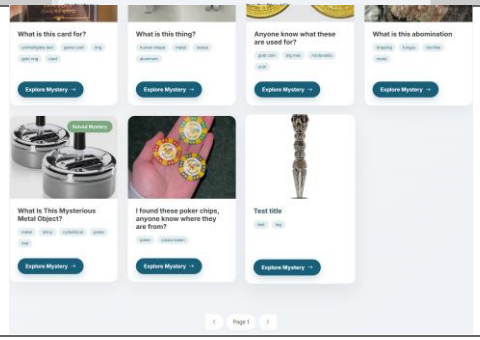
`swe573-backend/Test Results - java_in_swe573-backend.html`

GitHub doesn't render .html files in the browser but downloading this file and opening it in a browser will show all the test results.

8.2 User Acceptance Tests

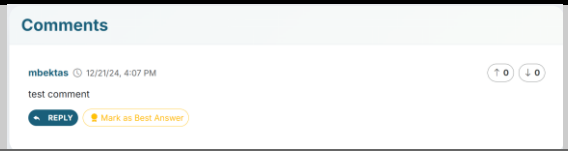
Post Creation (PASSED)

Background: User is logged in as a user and is on the main page.

Step	Expected Result	Actual Result	Status
Click the “NEW MYSTERY” button	Create New Post page appears with input boxes for Title, Content and Tags. Also, a button named “+ Add Mystery Object” should appear.		Passed
Enter Title, Content and Tags. Then click the “+ Add Mystery Object” button.	A modal named “Add Mystery Object” should appear with all mystery object attribute fields.		Passed
Enter Mystery Object details and click “Save”	The modal should disappear, and a post preview should be shown below the post creation box.		Passed
Click the “Create Post” button	User should be redirected to the main page and the new mystery should appear there.		Passed

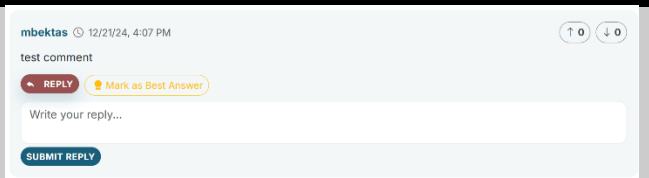
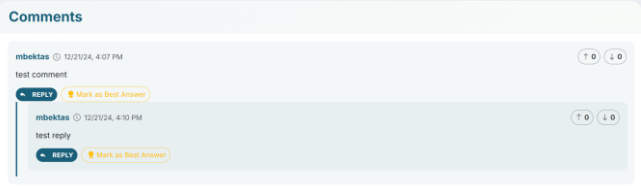
Adding a Comment (PASSED)

Background: User should be logged in and should be in a post details page.

Step	Expected Result	Actual Result	Status
Enter the comment to the “Add a comment” text box and click “Submit Comment”	Comment should be saved and shown in the Comments section.		Passed

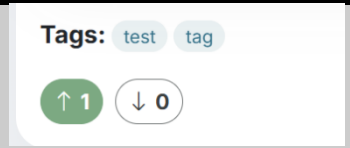
Adding a Reply (PASSED)

Background: User should be logged in and should be in a post details page.

Step	Expected Result	Actual Result	Status
Click on the “Reply” button under a comment	A text box for entering the reply should be shown.		Passed
Enter a reply and click “Submit Reply”	Reply should be saved and shown under the parent comment.		Passed

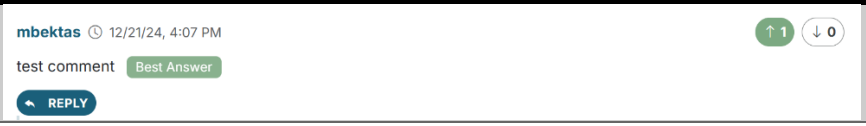
Upvoting a Post (PASSED)

Background: User should be logged in and should be in a post details page.

Step	Expected Result	Actual Result	Status
Click the upvote just below the tags of the post.	The upvote button should be greyed out while the upvote is saving, then should be green once upvote is saved and upvote count should increase by 1.		Passed

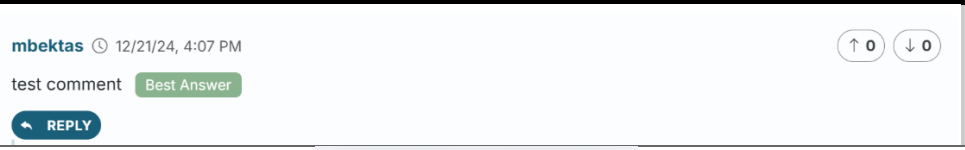
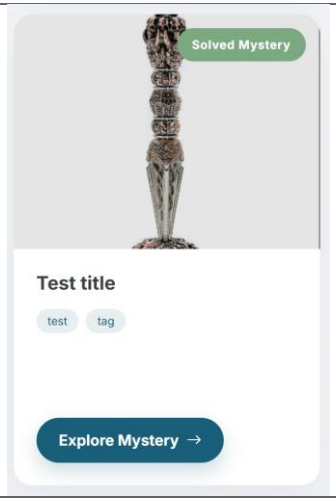
Upvoting a Comment (PASSED)

Background: User should be logged in and should be in a post details page.

Step	Expected Result	Actual Result	Status
Click the upvote just next to the comment	The upvote button should be greyed out while the upvote is saving, then should be green once upvote is saved and upvote count should increase by 1.		Passed

Marking the Best Answer (PASSED)

Background: User should be logged in and should be in a post details page of one of their posts (as post author).

Step	Expected Result	Actual Result	Status
Click “Mark as Best Answer” just below the desired comment.	The comment should be marked as the best answer with a badge.		Passed
Go to the main page	The post should be marked with “Solved Mystery”		Passed

9. Credentials and Posts for Testing

Posts named “I found these poker chips, anyone know where they are from?” and “What Is This Mysterious Metal Object?” can be tested with their authors:

Author for “I found these poker chips, anyone know where they are from?”:

Email: mb101@gmail.com

Password: 123456

Author for “What Is This Mysterious Metal Object?”:

Email: mustafabektas@hotmail.com

Password: 123456

Other posts and users were used for debugging and testing during the development of the application, but left there for a more populated home page.

Of course, users are welcome to create their own accounts and test the application.