TESTING PLAN FOR GIFHUB - CachingFeelings

0 ChangeLog

Version	Change Date	By	Description
Version Number	Date of Change	Name of Person Who Made Changes	Description of Changes Made
1	2024-02-19	Dingyuan Zhang	Created
2	2024-02-27	Heejeong Kim	Edit
3	2024-02-27	Breanna Brown	Edit
4	2024-03-11	Heejeong Kim	Edit
5	2024-03-12	Heejeong Kim	Edit

1 Introduction

1.1 Scope

This test plan will test the following core features:
Functional Requirements:
1. Account Creation
2. Student (Soul) Matching
3. Interact with Matches
4. Community Discovery

Non-functional Requirements:
1. Non-Functional Secure Authentication

1.2 Roles and Responsibilities

Name	UofM ID	GitHub Username	Role
Breanna Brown	millettb	bre9425	Developer
Rahul Kumar	kumarr7	bochacho	Developer
Oluwademilade Akinsola	akinsol2	Demi-AK	Developer
Heejeong Kim	kimh4	Heejoy	Developer
Dingyuan Zhang	zhangd7	AllNEWAREAS	Developer

2 Test Methodology

2.1 Test Levels

Unit Tests:
Core Feature 1: Account Creation
- Unit test on successful user creation
- Unit test to validate account creation needs a password
- Unit test to validate account creation needs a username
- Unit test to prevent duplicate users
- Unit test on a successful login
- Unit test to validate login credentials
- Unit tests to get users credentials
- Unit test to get current user id
- Unit tests to change password
- Unit test to make sure DOB is needed for modifying password

Core Feature 2: Student (Soul) Matching
- Unit test to get all my matches to view
- Unit test to test liking other profiles
- Unit tests to view users preferences
- Unit test to test not able to like yourself
- Unit test to make sure both id and like are needed
- Unit test to delete a message
- Unit test to validate can't re-like someone liked already
- Unit test to dislike a profile/user

- Unit test to validate need the user id (user id shouldn't be missing) to dislike or like them
- Unit test to prevent unauthorised access due to improper token

Core Feature 3: Interact with Matches
- Unit test to chat with matches
- Unit test to test sending a message
- Unit test to make sure if we both liked each other we can chat
- Unit test to not allow unauthorised access to newConve
- Unit test to not allow unauthorised access to getConve
- Unit test to allow users to get their message history in a conversation
- Unit test to not allow users that are not mutually linked to converse in a chat
- Unit test to get you all your conversation history with all your current matches
- Unit test to delete a message
- Unit test to start a new conversation

Core Feature 4: Community Discovery
- Unit test to test posting in community discovery
- Unit test to load up all posts in community discovery
- Unit test to test deleting a post
- Unit test to test liking a post
- Unit test to dislike a post
- Unit test to test reporting a post
- Unit test to ensure unauthorised access is denied by improper token
- Unit test to ensure posts are stored
- Unit test to ensure you can delete your post but not others
- Unit test to ensure post id is required when sending the body

Integration Test:
Core Feature 1: Account Creation
- Test sign up form information whether stored in database or not when successfully signed up for the new user
- Test whether backend sent appropriate information back to frontend, such as token
- Test login page when no valid user, backend sent error message
- Test when try to create an account with an already registered
- Test the sign up process enforces password strength requirements
- Test that all mandatory fields are validated on the backend
- Test the successful account creation
- Test that user creation is successful upon login, and are appropriately terminated upon logout.

Core Feature 2: Student (Soul) Matching
- Test the matching algorithm to ensure that users are matched based on shared interests
- Test that cached users can see each other in the match lists
- Test the system response when there are no available matches
- Test that users can update their matching preferences.
- Test reported or blocked users are no longer visible
- Test that user profiles are accessible and accurate within the matching interface

- Test that matched users can initiate conversations without issues

Core Feature 3: Interact with Matches
- Test the messaging functionality to ensure that users can send and receive messages with their matches
- Test that the conversation history is correctly saved and displayed for each match
- Test the ability to close or end a conversation.
- Test for deleting a match and confirm that deleted matches are removed from the match list
- Test the images are correctly sent and received with proper display
- Test the responsiveness of the messaging interface across different devices and screen sizes

Core Feature 4: Community Discovery
- Test the functionality for users to select and explore topics of interest within the "Community Discovery" page
- Test that thoughts related to the selected topic are correctly displayed to the user
- Test that user can like or dislike posts and these are correctly recorded
- Test that share thoughts to the topic are visible to others
- Test the date of posting sorted correctly

Acceptance Test:

Core Feature 1: Account Creation
- User Story: As a user, I want to sign up for a service to have my own account.
- Acceptance Criteria:
  1. Given that I am a new user,
  2. When I click the signup button,
  3. Then I am prompted to enter the following information in sequence:
     - o Username, password, and confirm password.
     - o Postal code and birthday.
     - o After entering each section, I click next.
  4. And I select my preferences and click next.
  5. And I type a Bio in the text box, select my interests, and click next.
  6. And I click choose a file to upload a profile picture.
  7. And I click the submit button after providing all the required information.
  8. Then I should be redirected to the 'Try' page to explore the service.

- User Story: As a user, I want to be able to change my password, so that I can use a new password to log in.
- Acceptance Criteria:
  1. Given that I am a user,
  2. When I click the 'userconfig' button in the top right corner,
  3. Then I am prompted to enter the following information to modify my password:
     - o Enter my new password.
     - o Confirm my new password.
     - o Enter my current password to authorise changes.

4. And I click the 'Update Info' button after entering all required fields,
5. Then my password should be changed, and I can use the new password to log in.

- User Story: As a user, I want to be able to log in, so that I can access the account that I created.
- Acceptance Criteria
  1. Given that I am a registered user,
  2. When I click the 'Login' button,
  3. Then I am prompted to enter the following information to log in
     o Enter my username.
     o Enter my password.
  4. And When I click the 'Login' button after filling in my username and password,
  5. Then I successfully logged in and redirected to the 'Try' page to explore the service.

- User Story: As a user, I want to be able to sign out, so that I can log in back later or log into a different account.
- Acceptance Criteria:
  1. Given that I am a user logged into this system,
  2. And I decide to log out of my account,
  3. And I click the 'Sign Out' button located in the top right corner of the main page,
  4. Then, I am redirected to the landing page, where I have the option to log in again or sign up for a new account.

- User Story: As a user, I want to update photo to my profile, so that I can show my photo to others
- Acceptance Criteria:
  1. Given that I am a user logged into this system,
  2. And I decide to update my profile photo,
  3. And I click the 'userconfig' button located in the top right corner of the main page,
  4. And, click 'Choose Files',
  5. And, click 'Update Images',
  6. Then, my profile picture will be updated and will show the updated picture to others.

Core Feature 2: Student (Soul) Matching
- User Story: As a user, I want to find people that match my interests and preferences, so that I can get to know them.
- Acceptance Criteria:
  1. Given that I am a user logged into the system,
  2. When I navigate to the 'Try' page,
  3. Then I can explore other users by interacting with the sphere.
     o Can drag the sphere to browse through user profiles.
  4. And when I select a user profile (node)
  5. Then I can view that user's detailed information, specifically their interests.

- User Story: As a user, I want to like users, so that I get to pick the user I like.
- Acceptance Criteria:
  1. Given that I am a user logged into this system,
  2. When I click the user's profile by clicking on their node within the interactive sphere,
  3. Then I am presented with two options:
     o Can click the 'Like' button to add the user to my favourites.
     o Can choose to close the profile view if I do not wish to favour the user.

- User Story: As a user, I want to view my matches, so that I can interact with them and know them better.
- Acceptance Criteria:
  1. Given that I am a user logged into this system,
  2. When I navigate to the 'catch' page,
  3. Then I can view the profiles of the users I have liked. On the 'try' page.
     o This includes full access to their information, allowing me to learn more about them and initiate interaction.

Core Feature 3: Interact with Matches
- User Story: As a user, I want to send and receive messages with my matches, so that we can know each other more.
- Acceptance Criteria:
  1. Given that I am a user logged into this system,
  2. When I navigate to the 'catch' page,
  3. Then, I can view all the users that I have previously liked.
  4. When I choose a user to start a chat with,
  5. And I click the 'Chat Now' button,
  6. Then I am taken to the 'finally' page where I can communicate with the user.
  7. And I can type my message into the chat input field,
  8. And When I press the 'Send' button,
  9. Then my message is sent to the user, enabling us to converse.

- User Story: As a user, I want to send a read-after-burn message, so that others can only see it for a certain time.
- Acceptance Criteria:
  1. Given that I am a user logged into this system,
  2. When I navigate to the 'finally' page,
  3. And I am in a chat room with a selected user,
  4. Then I can select the 'Burn after read' option before sending a message.
  5. When I send the message,
  6. Then the recipient will see a notification indicating a 'Read hidden message'.
  7. When the recipient opens the message,
  8. Then the 'Burn after read' message will be displayed in a pop-up window.
  9. When the recipient closes the pop-up window,
  10. Then the message will automatically be deleted from the chat room.

- User Story: As a user, I want to view all chats with different matches, so that I can continue to chat with them later.
- Acceptance Criteria:
  1. Given that I am a user logged into this system,
  2. When I navigate to the 'finally' page,
  3. Then I should be able to see a list of all users I have matched with, from which I can select a user to start or continue a conversation.

- User Story: As a user, I want to block the matches, so that I will no longer receive messages from them.
- Acceptance Criteria:
  1. Given that I am a user logged into this system,
  2. When I navigate to the 'catch' page,
  3. And decide to block a user from the list,
  4. And click 'Block' button,
  5. Then the user will no longer exist in my list.

- User Story: As a user, I want to send images in messages, so that the conversations will become more fun.
- Acceptance Criteria:
  1. Given that I am a user logged into this system,
  2. When I navigate to the 'finally' page,
  3. When I decided to send image to a user through chat,
  4. And When I click 'Add Attachments' button,
  5. And select image,
  6. And click 'Send',
  7. Then, the image I select will be sent in the chat room.

Core Feature 4: Community Discovery
- User Story: As a user, I want to discover people in the community where everyone can post thoughts so that I can find interesting people.
- Acceptance Criteria:
  1. Given that I am a user logged into this system,
  2. When I click the 'Community" button located in the top right corner,
  3. Then I am taken to the community page where I can view all posts and thoughts shared by users.

- User Story: As a user, I want to post my thoughts to the community, so that everyone can see my thoughts.
- Acceptance Criteria:
  1. Given that I am a user logged into this system,
  2. When I navigate to the community page,

3. Then I can enter my thoughts into a text box that is positioned at the bottom of the community page.
4. And When I click the 'Post' button after typing my thoughts,
5. Then my thoughts should be posted publicly on the community page for all users to view.

- User Story: As a user, I want to like and dislike thoughts in the community, so that I can express my attitude to that thought.
- Acceptance Criteria:
  1. Given that I am a user logged into this system,
  2. When I navigate to the community page,
  3. And I can view all the thoughts that users have posted,
  4. When I choose to express my attitude toward a thought,
  5. And I can click the 'Like' button to like the post or click the 'Dislike' button to dislike the post.
  6. Then the number of likes or dislikes will increase on that post.

- User Story: As a user, I want to delete my thoughts, so that others will not see that thought.
- Acceptance Criteria:
  1. Given that I am a user logged into this system,
  2. When I navigate to the community page,
  3. Then I can view all the thoughts posted by users, including my own.
  4. And When I decide to remove one of my thoughts,
  5. Then I can click on the 'Delete' button associated with my post.
  6. Then the thought that I post will be removed from the community page.

- User Story: As a user, I want to report inappropriate or offensive content in the community, so that the community can stay in peace and love.
- Acceptance Criteria:
  1. Given that I am a user logged into this system,
  2. When I navigate to the community page,
  3. Then I can view all the thoughts posted by users,
  4. And When I decide to report one of the user's thoughts,
  5. Then I can click on the 'Report' button,
  6. Then the thought that I report will be removed from the community page.

Regression Test:
We configured the CI pipeline using GitHub Actions to automatically build the project and execute all tests whenever a pull request is submitted.

Load Balance Test:
The system is expected to efficiently manage the specified volume of concurrent requests without compromising performance. Initiate the creation of a new user profile at a consistent

rate of 1 user per second, engage 100 active users within the app and direct these users to send 10 messages per second.

2.2 Test Completeness

Criterias for Test Completeness:
- Automated testings should be executed
- All the bugs and failed test should be checked and fixed
- All tests should cover 100% of the codebase
- Bugs should be fixed and reported

3 Resource & Environment Needs

3.1 Testing Tools

Following tools are used in our project for testing:
- Automation: GitHub Actions
- Backend Test: Jest
- Frontend Test: Jest

3.2 Test Environment

Following environments are used in our project for testing:
- MacOS Sonoma
- Docker

4 Terms/Acronyms

Term/Acronym	Definition
API	Application Program Interface
AUT	Application Under Test
CI	Continuous Integration