System and Unit Test Report

Spotify Match

7th March 2023

System Test Scenarios

User Stories

1. User story 1 from Sprint 1 - As a person, I would like to have my own personal profile and a way to change it later on Spotify Match, so I can have a more personalized experience.
2. User story 2 from Sprint 1 - As a person, I would like to navigate from one screen to another. Eg- after logging in to the app I would like to go to the edit profile page from the home page.
3. User story 1 from Sprint 2 - As a person, I would like to have potential matches based on my spotify music interests.
4. User story 2 from Sprint 2 - As a person I would like to see what music I have in common with a potential match.
5. User story 1 from Sprint 3 - As a person, I would like to easily swipe through the user cards on the homepage.
6. User story 2 from Sprint 3 - As a person, I would like to have a place where I can see my matches, so I can keep track of who is interested in me as well.

7. User story 1 from Sprint 4 -As a person I would like to have a matches page that displays all the matches which include their social media such that I can contact them if I am interested in them.
8. User story 2 from Sprint 4 - As a person I would like to be able to logout and login to the Spotify Match App such that the app can store my information whether I'm logged in or not.

Scenarios

1. Start Spotify Match; select ‘Log In’;type
    - Email = <matchspotify.test3@gmail.com>
    - Password = <test1234@>
    - Press Log In button
    - The user should be redirected to the survey page where the user can answer the questions.
2. The user should be able to see a navigation bar on the bottom on the screen.
    - Click the third button on the navigation bar.
    - The user should be navigated to another screen.
3. The user should be able to see a user card.
    - Swipe through the card
4. The user should be able to see a user card in the home screen where the user would be able to see their potential matches.
    - On the user card, Click the percentage circle.
    - The user will be able to see the music they have in common with their potential matches.

5. The user should be able to see a user card.
    - Swipe left or right to like or dislike a user.
6. The user should be able to see a navigation bar on the bottom on the screen.
    - Click the first button on the navigation bar.
    - The user should be navigated to the matches screen where the user would be able to see their matches.
7. The user should be able to see their matches on the screen.
    - Click on any circle, you should be navigated to a new socials screen.
    - The user should be able to see the socials of the match they clicked on.

8. The user should be able to see a navigation bar on the bottom of the screen.
    - Select the third bar which is the user profile page.
    - Scroll down to the end of the page.
    - Click the logout button.
    - The user should now be logged out and should have to sign in again to use the app.

Unit tests:

1. Navigation and Render Test - Checks whether the navigation of the app works fine. It also checks whether all the screens render properly along with the navigation.
2. Database Test - Checks the database and gets the user information by id and checks the validity of the information.
3. Redux Store Initial Test - Checks the initialization of the store.
4. Redux Store Storing Test - Checks the storage of the data in the store.
5. Create User Test - Simulates a PUT request and creates a user in the database.
6. Get Home Users Test - Simulates a GET home request for a user and checks the response for validity. Returns status code 200.
7. Get Profile Questions Test- Simulates a GET profile questions request and checks the response for validity. Returns status code 200.
8. Get Profile Questions After Insertion Test - Simulates a GET profile questions request after inserting some data into the database and checks the response for validity. Returns status code 200.
9. Update User Test - Simulates a PUT request to update the User’s data. Returns status code 200.
10. Post Interest Test - Simulates a POST request to add an interest of a user. Returns status code 200.