CSE 115A – Introduction to Software Engineering

Test Plan and Report

Product Name: RepIt
Team Name: RepIt
November 29, 2024

User Stories:

1. As a new fitness enjoyer, I should be able to log in, so that I can save my progress
2. As a gym-goer, I would like to be able to see/edit session history, so that I can see what I worked on last week
3. As a gym-goer, I should be able to create a session, so that I can track my journey
4. As a new gym member, I should be able to see the machines in my current gym, so that I know which gym to visit
5. As a local gym member, I should be able to add machine to a gym, so that what machines there'd be at that gym for my sessions
6. As a consistent athlete, I should be able to see graphs with my weight/sets/bodyweight progressions, so that I can see if I'm on track for my goals
7. As a consistent athlete, I should be able to set goals and see previous goals, so that I can plan out my progression
8. As an athlete, I should be able to create/see previous lists of supplements + reminders, so that I don't forget anything

System Test Scenarios:

Running tests:

- cd GymerApp/ && npm run test
    - (npm run test ⇒ jest --selectProject frontend backend)
- For specific features
    - npm run test:facilities
    - npm run test:goals
    - npm run test:sessions
    - npm run test:supplements

Scenario 1: Landing Page

1. User will see elements unique to their sign-in status
    a. If signed out, user will expect a Clerk sign-in modal when clicking "Get Started" button
    b. If signed out, user will expect a sign-in button in the navbar
    c. If signed in, user will expect a redirect to the dashboard when clicking "Go to Dashboard" button
    d. If signed in, user will expect a Clerk user button instead of the sign-in button in the navbar

2. User will expect to be redirected to respective links when clicking the navbar buttons
    a. Sessions redirects to /sessions, goals to /goals, etc.
3. User will expect a different navbar style based on screen width
    a. When screen width is set to (less than) 720px, a dropdown button should replace the navbar buttons, which opens the dropdown menu

Relevant tests:

- /GymerApp/src/app/_components/__tests__/*

Scenario 2: Sessions page

1. When user loads into the sessions page, they can expect a list of sessions that they have created, made up of session cards
2. User can expect and click the edit button on a card to open up the edit session form
    a. In order to submit the edit form, the user must input valid values into the fields. Otherwise, they will be prompted until valid
        i. Clicking the submit button will update the session and update the card
    b. The user can expect a confirmation modal to be gone when they click the delete button and a full removal of the session when they click the red button on the modal
3. User can click the “+ add session” button and expect a session form to appear
    a. In order to submit the creation form, the user must input valid values into the fields. Otherwise, they will be prompted until valid
        i. Clicking the submit button will create the session and appear in the list

Relevant tests:

- /GymerApp/src/app/sessions/_components/__tests__/*
- /GymerApp/src/app/components/forms/__tests__/SessionForm.ts
- /GymerApp/src/app/components/forms/__tests__/EditSessionForm.ts
- /GymerApp/src/server/api/__tests__/sessions.test.ts

Scenario 3: Goals page

1. When user loads into the goals page, they can expect a list of goals that they have created, made up of goals cards
2. User can expect and click the edit button on a card to open up the edit goals form
    a. In order to submit the edit form, the user must input valid values into the fields. Otherwise, they will be prompted until valid
        i. Clicking the submit button will update the goal and update the card
    b. The user can expect a confirmation modal to be gone when they click the delete button and a full removal of the goal when they click the red button on the modal
3. User can click the “+ add goal” button and expect a goal form to appear
    a. In order to submit the creation form, the user must input valid values into the fields. Otherwise, they will be prompted until valid
    b. The due date for the goal must be today’s date or a date in the future. Otherwise, the user will be prompted to select a date that falls under that.
        i. Clicking the submit button will create the goal and it appear in the list

ii. Depending on the date selected, the goal will sorted into the various sections: ‘Past Due’, ‘Due This Week’, ‘Upcoming Goals’

4. User can mark their goal as being completed
    a. The user can expect to see their goal will be sorted into a ‘Completed Goals’ category.

Relevant tests:

- /GymerApp/src/app/goals/_components/__tests__/*

Scenario 4: Supplements page

1. When user loads into the supplements page, they can expect a list of supplements that they have created, made up of supplements cards
2. User can expect and click the edit button on a card to open up the edit supplements form
    a. In order to submit the edit form, the user must input valid values into the fields. Otherwise, they will be prompted until valid
        i. Clicking the submit button will update the supplement and update the card
    b. The user can expect a confirmation modal to be gone when they click the delete button and a full removal of the supplement when they click the red button on the modal
3. User can click the “+ add goal” button and expect a supplement form to appear
    a. In order to submit the creation form, the user must input valid values into the fields. Otherwise, they will be prompted until valid
    b. The optional end date, if added, must fall after the given start date. If this isn’t the case, the user will be prompted until valid.
        i. Clicking the submit button will create the goal and it appear in the list
4. User can mark their supplement as being active or inactive
    a. The user can expect to see their supplements will be sorted into its respective section.

Relevant tests:

- /GymerApp/src/app/supplements/_components/__tests__/*

Scenario 5: Facilities page

1. When user loads into the facilities page, they can expect a facility search bar, a search button, and a “I don’t see my facility” button
    a. When the user clicks the search button, they can expect a list of nearby facilities via facility cards.
        i. If location services are disabled, the user can expect an error regarding it.
        ii. If the user inputs a value into the search bar and clicks search, they can expect facilities matching that value by name or leisure type.
2. On the facility card, the user can expect to see facility details and add/view buttons for equipment.
    a. When the user clicks the add equipment button, they can expect an equipment form to fill in, and with valid inputs, it submits the form.
    b. vWhen the user clicks the view equipment button, they can expect a sheet to appear with all the current equipment of the facility.

Relevant tests:

- /GymerApp/src/app/facilities/_components/__tests__/*

Unit Tests (that are failing):

- SessionForm.test.tsx
  - SessionForm Component
    - “Submits form with valid data”
    - “Shows error when session”
- FacilityCard.test.tsx
  - FacilityCard Component
    - “Fetches equipment when AddEquipmentButton triggers callback”
    - “Handles fetch errors gracefully”
- ViewEquipmentsButton.test.tsx
  - ViewEquipmentsButton
    - “Fetches equipment data and opens the sheet on success”
- FacilitySearchBar.test.tsx
  - FacilitySearchBar
    - “Navigates to /facilities when the input is cleared”