

Working Software Report – 22CLC01 – Group 04

1. Introduction

- **Project Name:** The Health and Fitness Web Application
- **Group ID:** Group 4
- **Sprint/Iteration Number:** Sprint 4
- **Report Date:** 08/12/2024

Purpose

This report documents the progress and functionality of the working software version at the end of this sprint. It focuses on the implementation and functionality of the main flow of 5 use cases, as per the project requirements.

2. Tools and Technologies Used

- **Frontend:** Vite.js, React, TypeScript
- **Backend:** Node.js, Express, Firebase
- **Database:** Firestore Database
- **Testing Tools:** Postman for API testing, manual testing
- **Environment:** Chrome browser, Zen Browser (Firefox based)

3. Overview of the Working Software

a) Log in

Features Implemented

1) Main flow: Log in using an email.

- Use Firebase Auth to verify auth credentials.
- User is navigated to Home Page if successful, otherwise show a red error prompt under the email input bar.

2) Alternative Flow: Move to sign up.

Steps to Test the Main Use Case

- 1) **Step 1:** From Home Page, navigate to the log in page.
- 2) **Step 2:** Enter invalid user details and submit.
- 3) **Step 3:** Verify that the user stays on the Log in page with the corresponding error prompt. (Invalid email address, Invalid email or password).
- 4) **Step 4:** Enter valid user details and submit.
- 5) **Step 5:** Verify that the user navigates to the Home Page with a “MY HAF” navigator at the left of the Navigation Bar

Expected Results

- Invalid email format: Red prompt “Invalid email address”.
- Valid email format, but invalid user credentials: Red prompt “Invalid email or password”.
- Valid user credentials: Redirect to Home Page with valid UI.

Actual Results

The use case works as expected

Challenges Faced

There were no critical problems when implemented this use case.

Resolutions

b) Log out

Features Implemented

User logs out from the authenticated session, back to guest session.

Steps to Test the Main Use Case

- 1) **Step 1:** From Home Page, hover to “MY HAF” navigator on the Navigation Bar
- 2) **Step 2:** Click Log out
- 3) **Step 3:** Verify that the Home Page returns to the Public View
- 4) **Step 4:** Verify that “user” variable in AuthProvider is **null** (for now tests are using `console.log(user)` in the right place of the application flow)

Expected Results

- Home Page returns to the guest view
- "user" variable goes to **null**

Actual Results

The use case works as expected

Challenges Faced

There were no critical problems when implemented this use case.

Resolutions

c) Register

Features Implemented

User signs up for a user account

Steps to Test the Main Use Case

- 1) **Step 1:** From Home Page, user clicks Sign up button on the Navigation Bar
- 2) **Step 2:** Fill in Full Name (or username), email address, password and confirm password
- 3) **Step 3:** Verify that the guest can't finish registration process if they typed invalid email format
- 4) **Step 4:** Verify that the guest can't finish registration process if they type in different passwords
- 5) **Step 5:** Verify that if all inputs are valid, the user creates an account and logs in to the Authenticated View

Expected Results

- Invalid email format: Can't sign up
- Wrong confirm password: Can't sign up
- Valid inputs: Sign up successfully and Log in

Actual Results

There is a small UI/UX problem if the user enters an invalid email format: "Manage passwords" box appear and hide the error prompt

Challenges Faced

There were no critical problems when implementing this use case.

Resolutions

Front-end developers (Quang and Dương) will fix the bug.

d) View profile and Edit an account (User Profile Page)

Features Implemented

User can view, edit profile (Current Status and Goals) and edit password on the Profile

Steps to Test the Main Use Case

- 1) **Step 1:** After successfully registering or logging in, the user points to his or her name on the navigation bar and selects profile.
- 2) **Step 2:** Check if the profile page shows the right data of the user's information or not.
- 3) **Step 3:** Click on the change button next to "password", fill in the new password and select confirm.
- 4) **Step 4:** Verify that a new password has been changed on the Firestore Database.
- 5) **Step 5:** On the right corner of the profile page, select edit and change the information the user wants to change, select finish edit after making changes.
- 6) **Step 6:** Verify that after editing, user information is updated on the Firestore Database.

Expected Results

- Invalid Password when changing (current password, length, format, confirm password...): Alert "Failed to change password. Please try again."
- The page shows the right data of the user's information (username, email, status, goal)
- After editing status or goal, the input data is written to Firestore Database successfully

Actual Results

There is a small delay to wait for the data to appear when the user reloads the page. For less than a second, all the fields will be blank, then the actual data fills the screen.

Challenges Faced

- Without a wrapped ProfileProvider, the app could not get user data conveniently at first.
- At first, the team could not make the right data appear again after a reload. The tester had to go back to the Home Page then back to the Profile page every time. We thought the user's auth was corrupted, but it was because of the loading process not being awaited.

Resolutions

The team will consider if a solution of delay loading the page is needed. For now we think that this way of lazy-loading data is acceptable.

e) Search and Filter exercises, View information of an exercise

Features Implemented

- Full Exercise Gallery
- Search Exercise
- Filter based on Muscle and Equipment
- View an exercise and its alternative ones

Steps to Test the Main Use Case

- 1) **Step 1:** From Home Page, navigate to the exercises section.
- 2) **Step 2:** Input the name of the exercise into the search field and enter.
- 3) **Step 3:** Click on the Muscles and Equipment field, and choose the wanted ones.
- 4) **Step 4:** Click on any specific exercise to view its information and related exercises

Expected Results

- View gallery: Each page will display 10 exercises.
- Successful search: Exercise that matches the searched name.

- Successful filter: Exercises that have equipment and muscles that match the filter.
- Successful view: Display information of exercise and related exercises.

Actual Results

All features are implemented successfully

Challenges Faced

Had difficulty in server pagination (Display number of exercises per page)

Resolutions

Using last visible function of Firestore to markdown the number of exercises for each page

4. Status Summary

Use case	Status (Working/Not Working)	Comments
Authentication (Log in, Log out, Register)	Working	Fully functional with a non-critical issue in Register Page
User Profile Page (View and Edit Profile)	Working	Fully functional
Exercises Page (Search and Filter exercises, View information of an exercise)	Working	Fully functional

5. Demo

Link demo:  Intro to SE