CMPT 276 Proposal 1

09/17/2023 Version 0
10/09/2023 Version 1
10/29/2023 Version 2

Project name: SweatSink

Group members:

Eric Bowerbank	Andy Liu	Dongan Kim	Sam Showkati	Phoebe Han
301576680	301472847	301441678	301443052	301558325

Repository: https://github.com/SamShowkati/CMPT276-SweatSink

Abstract:
We're aiming to create a fitness-tracking app focused on weightlifting and cardio training using the Android Studio IDE and coding in Java and Kotlin. The app will utilize the OpenAI's API and the Google Search API to allow users to access an AI personal trainer and feedback on their training, search for specific workouts and training methods, or to simply do more research on certain exercises. Our app will also make use of the Google Calendar's API, as well as Google map's API which will allow users to schedule workouts throughout the week and plan optimal running routes around their neighborhood. We will also store an offline database of exercises for specific body parts to allow users to target certain areas they may want to work out.

Customer:
The target audience for this Android app is anybody who is looking to create and achieve fitness goals. The app will benefit customers who are looking for fitness guidance but can't or don't wish to seek help from a human. It will be useful for both beginners looking to start weightlifting and cardio as well as intermediates who are already experienced. This app will be free meaning it can be easily accessed by people of all ages and financial status.

Competitive Analysis:
Competitors:

- MyFitnessPal
- Nike Training Pal
- Google Fit
- Peloton

The applications listed above will be our biggest competitors. However, our application strives to provide our users with the use of an AI personal trainer that will provide feedback based on user input. Additionally most of the applications listed above are aimed towards iOS users, while our application will be more geared towards android users. In short, we are aiming to provide our users with all their favorite features together into one app.

Differences that our app brings:

- Most of these apps allow for the creation of a custom workout plan but this can often require a lot of research and experience that is hard to find. Our app will allow for custom workout plans that are created with the help of an AI personal trainer integrated with

CMPT 276 Proposal 1

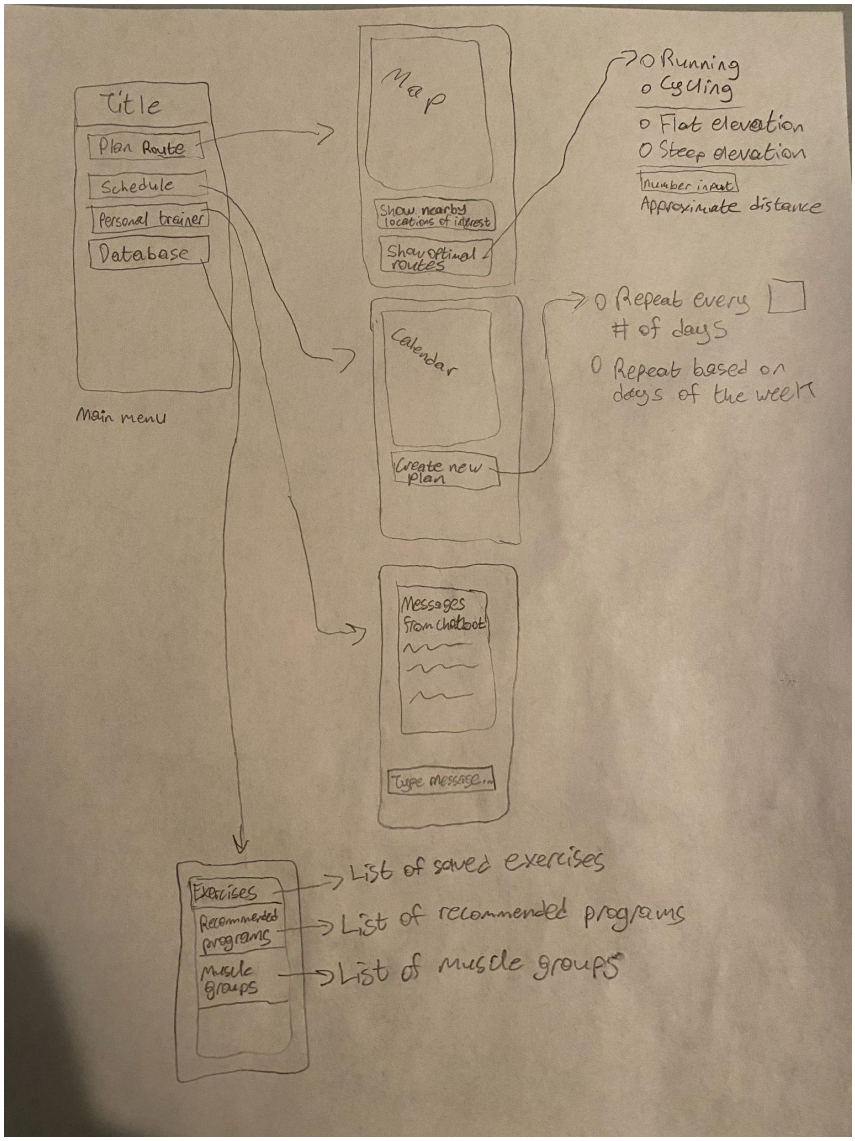
09/17/2023 Version 0
10/09/2023 Version 1
10/29/2023 Version 2

Project name: SweatSink

Chat GPT. This will give the user access to information that they otherwise wouldn't be able to easily receive for free.

- Any workout plans that, for example, an intermediate lifter has been using can be optimized with the AI personal trainer's advice. This also isn't a service that is available on the other apps.
- Provides live guidance and encouragement for people who tend to work out alone.
- Can provide live safety tips to ensure the user is not only working out efficiently but also without the risk of injury
- Using AI, not only helps users to achieve their goals, but also to set realistic and appropriate goals

User Interface:



CMPT 276 Proposal 1

09/17/2023 Version 0
10/09/2023 Version 1
10/29/2023 Version 2

Project name: SweatSink

Documentation

When opening the app, users will be presented with the main menu. From here, they have 4 main options: Plan Route, Schedule, Personal Trainer, and Saved Data. Plan route takes users to the map, where in future iterations they will be able to see nearby fitness-related points of interest and plan routes for running. In the schedule, users can see their calendar. In future iterations, they will be able to use this to plan recurring workouts that will appear on their own Google Calendar. The personal trainer is an AI assistant. Saved data has a locally stored database of exercises that the user can access offline.

The maps page uses Google Maps API to show the user a map with a pin on their current locations. Users can create more pins by tapping any location on the map, and press the "Create Route" button to calculate a route between these pins.

The schedule page contains 2 buttons, one for scheduling a run on the selected date on the calendar, and another one for clearing all previously planned schedules.

In the assistant page, users can send messages to their fitness assistant powered by ChatGPT and OpenAI API. The assistant's responses will appear quickly in the app, and there is a "Save as Exercise" button which will automatically save the assistant's most recent response to the database.

Saved data contains "Exercises," which are notes that are both pre-installed and which can be modified and created by the user. Users can add and remove exercises, with a maximum of 14 exercises at a time.

User Stories

User Suggestion:

As someone with a busy schedule, I would like a scheduling system integrated within the app. This will provide my clients with keeping on track with their workout progress.

Name:	Scheduling system
Actor:	Regular User
Precondition:	User is in the Calendar page, has signed into their Google account, has given the app permission to access their calendar, and has an internet connection.
Iteration:	1
Actions:	Implement a scheduling function using Google Calendar API. Users should be able to make changes that will be reflected in their own, personal Google Calendar. The goal of this feature is to help users stay on top of their training through scheduling.

CMPT 276 Proposal 1

09/17/2023 Version 0
10/09/2023 Version 1
10/29/2023 Version 2

Project name: SweatSink

Functional test:	If the user is on the main menu, when they press the schedule button, they are taken to their calendar as desired. If the user is on the Calendar page and they select a date, a popup message will appear telling them the details of that date. Given the user attempts to schedule a workout, when there is a failure in the Google Calendar API integration (e.g. API downtime or authentication issues), then an error message should be displayed, informing the user that the scheduling action could not be completed, and they should try again later. Given the user attempts to schedule a workout, when they provide invalid data (e.g. missing workout details or conflicting schedules), then an error message should be displayed, indicating the specific issue and guiding the user on how to correct it.
Unit test:	MainActivity.kt: all code functions as expected and is error free. main_activity.xml: all code functions as expected and is error free. calendar_activity.xml: all code functions as expected and is error free. CalendarActivity.kt: all code functions as expected and is error free. AndroidManifest.xml: all code functions as expected and is error free. The calendar page is working bug free, as tested on Google Pixel 3a and 6 emulators.

User Suggestion:

As a user, I would like to see Google Maps iterated into the application to help me plan optimal running routes around the neighborhood or find places to work out near me.

Name:	Integrated Map
Actor:	Regular User
Precondition:	User is in the Maps page, has given the app permission to access their location, and has an internet connection.
Iteration:	1

CMPT 276 Proposal 1

09/17/2023 Version 0
10/09/2023 Version 1
10/29/2023 Version 2

Project name: SweatSink

Actions:	Integrate Google Maps onto our application, use Google Maps API and location services to allow users to plan running routes in their neighborhood and locate places to work out.
Functional test:	If the user is on the main menu and presses the “Plan Your Route” button, they are taken to Google Maps as desired. Given the user attempts to access the map functionality, when there is a failure in the Google Maps API integration (e.g. API downtime or authentication issues), then an error message should be displayed, informing the user that the map service is temporarily unavailable, and they should try again later. Given the user attempts to access location-based features, when they do not have the necessary location permissions enabled on their device, then an error message should be displayed, guiding the user to grant the required location permissions.
Unit test:	MainActivity.kt: all code functions as expected and is error free. main_activity.xml: all code functions as expected and is error free. activity_maps.xml: all code functions as expected. MapsActivity.kt: all code functions as expected. AndroidManifest.xml: all code functions as expected. The maps page is working bug free, as tested on Google Pixel 3a and 6 emulators.

User Suggestion:

As someone who is frequently unable to access internet or cellular data, I would like to be able to access a database of exercises and fitness info while offline.

Actor:	Regular User
Precondition:	User is in the Database page and has sufficient storage space to store the database.

CMPT 276 Proposal 1

09/17/2023 Version 0
10/09/2023 Version 1
10/29/2023 Version 2

Project name: SweatSink

Iteration:	1
Actions:	Implement and display to the user an offline database that tracks saved exercises, recommended programs, and a list of muscle groups. These are saved to the user's devices as .txt files and displayed in the app.
Functional test:	When the user is on the main menu, if they press the "Saved Data" button, then they are taken to the database page as desired. Given the user presses the "Saved Data" button, when the user's free storage space is running low, then a message should be displayed notifying the user of this issue. Given the user has accessed the database page, when the offline database is empty, then a message should be displayed, notifying the user that no data is available and suggesting actions like adding an exercise.
Unit test:	MainActivity.kt: all code functions as expected and is error free. main_activity.xml: all code functions as expected and is error free. database_activity.xml: all code functions as expected and is error free. DatabaseActivity.kt: all code functions as expected and is error free. AndroidManifest.xml: all code functions as expected and is error free. The database page is working bug free, as tested on Google Pixel 3a and 6 emulators.

User Suggestion

As someone with very little experience in weight training, I would like to see AI integrated within the app to help with my workouts and training exercises. This would be an excellent way to help me start my weight training journey, and allow me to ask questions I may not be comfortable asking other people.

Name:	AI personal trainer
Actor:	Regular User
Precondition:	User is in the assistant page and has a valid

CMPT 276 Proposal 1

09/17/2023 Version 0
10/09/2023 Version 1
10/29/2023 Version 2

Project name: SweatSink

	OpenAI key and internet connection.
Iteration:	2
Actions:	Implement an assistant within the application that utilizes the ChatGPT API to allow users to gather information on training methods and workouts and automate creation of workout plans. Responses to the user's queries are sent to GPT 3.5 model via OpenAI API and responses are displayed to the user in the app.
Functional test:	If the user is on the assistant page and submits a query to the assistant, then they are met with a quick and helpful response. If the user sends a message to the assistant and there is no response from OpenAI API, or the response from the API doesn't contain the message from ChatGPT, then a message is shown to the user stating that there's been an API error. If the user has received a message from the assistant and they press the clear button, then the assistant's response will be cleared and reset to default.
Unit test:	AssistantActivity.kt: all code is functioning and error free. When attempting to extract ChatGPT's response from OpenAI API's message, a JSONException is successfully caught if the message is empty or missing the section containing ChatGPT's response.

User Suggestion:

As someone who wants to work out a specific part of my body, I would like pre-planned workouts available at my disposal for simplicity.

Name:	Workout Database
Actor:	Regular User
Precondition:	User is in the Database page and has sufficient storage space to store the database.
Iteration:	2

CMPT 276 Proposal 1

09/17/2023 Version 0
10/09/2023 Version 1
10/29/2023 Version 2

Project name: SweatSink

Actions:	Using local storage and Kotlin file IO, we will store an offline database with pre-planned workouts for a specific muscle group or body part that can be viewed as text within the app.
Functional test:	If the user is on the Saved Data page and they open a saved exercise, then they can view and edit its contents. If the user is on the Saved Data page and they attempt to open an exercise that doesn't exist yet, then a new empty exercise is created.
Unit test:	DatabaseActivity.kt: all code is functioning and error free. DatabaseExercisesActivity.kt: all code is functioning and error free. If the user attempts to open a file that doesn't exist, that file is created and set to empty.

User Suggestion:

As someone with unreliable internet or cellular data, I would like to be able to save any additional notes I may have within the app.

Name:	Workout Database
Actor:	Regular User
Precondition:	User is in the Database page and has sufficient storage space to store the database.
Iteration:	2
Actions:	Using local storage and Kotlin file IO, we will store an offline database with user-defined notes that can be viewed as text within the app.
Functional test:	If the user is on the Saved Data page and they open any saved data, then they can modify the text and save these changes persistently. If the user is on the Saved Data page and they click the Remove button, then the most recently added data is removed. If the user is on the Saved Data page and

CMPT 276 Proposal 1

09/17/2023 Version 0
10/09/2023 Version 1
10/29/2023 Version 2

Project name: SweatSink

	they click the Add button, then a new blank note is created which they can edit. If the user clicks the Remove button and there is nothing to remove, a popup message will inform them of this. If the user clicks the Add button and they are at their maximum amount of saved exercises, a popup message will inform them of this.
Unit test:	DatabaseActivity.kt: all code is functioning and error free. Database_activity.xml: all code is functioning and error free. DatabaseExercisesActivity.kt: all code is functioning and error free. If the user attempts to open a file that doesn't exist, that file is created and set to empty.

User Suggestion:

As a beginner athlete, I would like to have exercises generated by the AI assistant integrated with the rest of the app.

Name:	Workout Database and AI Assistant integration
Actor:	Regular User
Precondition:	User is in the Assistant page, has a valid OpenAI API key, has an internet connection, and has sufficient storage space to store the information to be added.
Iteration:	2
Actions:	Use a button on the Assistant page which will save the Assistant's last response as an exercise in the Exercises page.
Functional test:	If the user is on the Assistant page and they click the "Save as Exercise" button, then the current assistant's response will be saved in the Exercises page. If the user clicks the "Save as Exercise" button and they are at the maximum amount of saved exercises, a message will inform the user of this and it won't be saved.

CMPT 276 Proposal 1

09/17/2023 Version 0
10/09/2023 Version 1
10/29/2023 Version 2

Project name: SweatSink

Unit test:	AssistantActivity.kt: all code is functioning and error free.

Future:

As a regular runner who just moved into a new neighbourhood, I would like to use the application to help me plan optimal running routes with adjustable variance in elevation to best fit my running needs.

As a traveling athlete, I would like to use the app to locate specific fitness facilities (e.g. traditional gym, rock climbing gym, sports equipment store) on the go. This will allow me to continue my training even while traveling

As a busy individual, I would like to use the app to set specific and customizable schedules with a large degree of freedom over how I schedule my workouts (e.g. every other day, weekdays only, specific days of the week, etc).

User Suggestion:

As someone who walks regularly, a step counter would be beneficial to me in tracking my exercise and adjusting my workouts accordingly.

Name:	Step counter
Actor:	Regular User
Precondition:	Counts steps taken by the user daily
Iteration:	2
Actions:	not yet decided