Adaptive Fitness Application

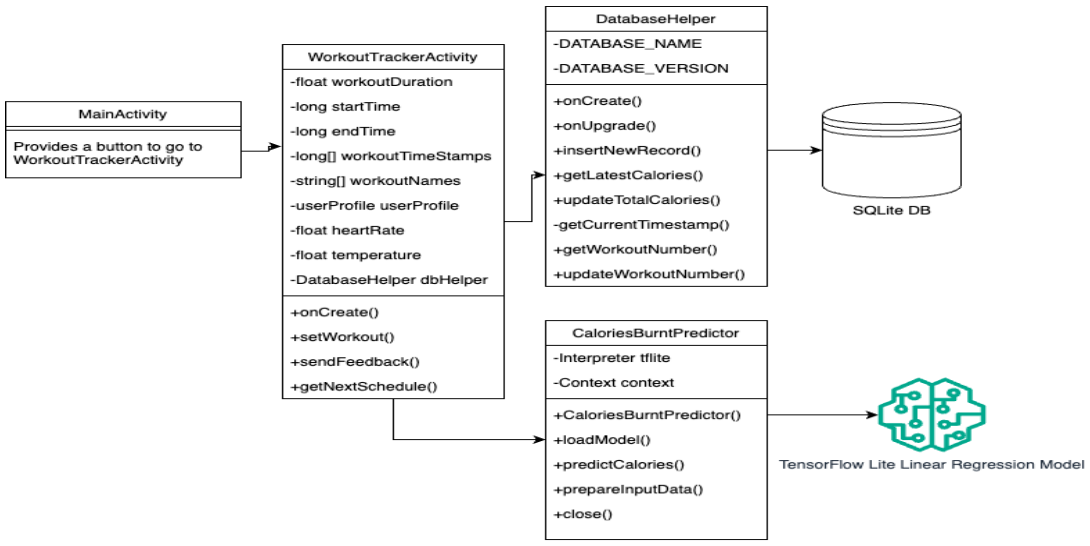
Aksa Elizabeth Sunny
ASU ID: 1227737175
GitHub Branch Name: AksaElizabeth_Sunny

1. Alignment with Guardian Angel

The Adaptive Fitness Application is an innovative fitness platform that leverages advanced technology and individual user data to craft customized workout plans. Emphasizing a user-focused design, the app provides personalized fitness regimens that adapt to each user's health conditions, unique preferences and external environmental conditions and location.

Incorporating a machine learning component, the feedback feature of the Adaptive Fitness Guide Application significantly enhances its capability by accurately calculating calories burned during each workout. This sophisticated approach to data analysis ensures precise calorie tracking. Crucially, this information is then fed back into the recommendation engine, serving as a vital input for personalizing subsequent workout suggestions. By continuously adapting to the user's workout results and progress, the app's machine learning algorithms can more effectively tailor fitness routines, aligning them closely with individual goals and evolving fitness levels.

2. Specifications

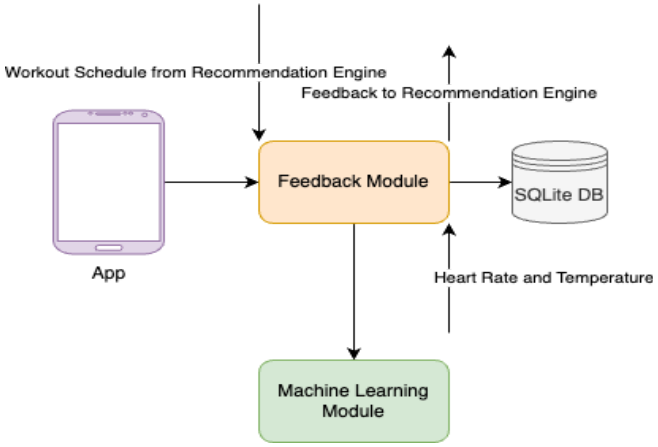


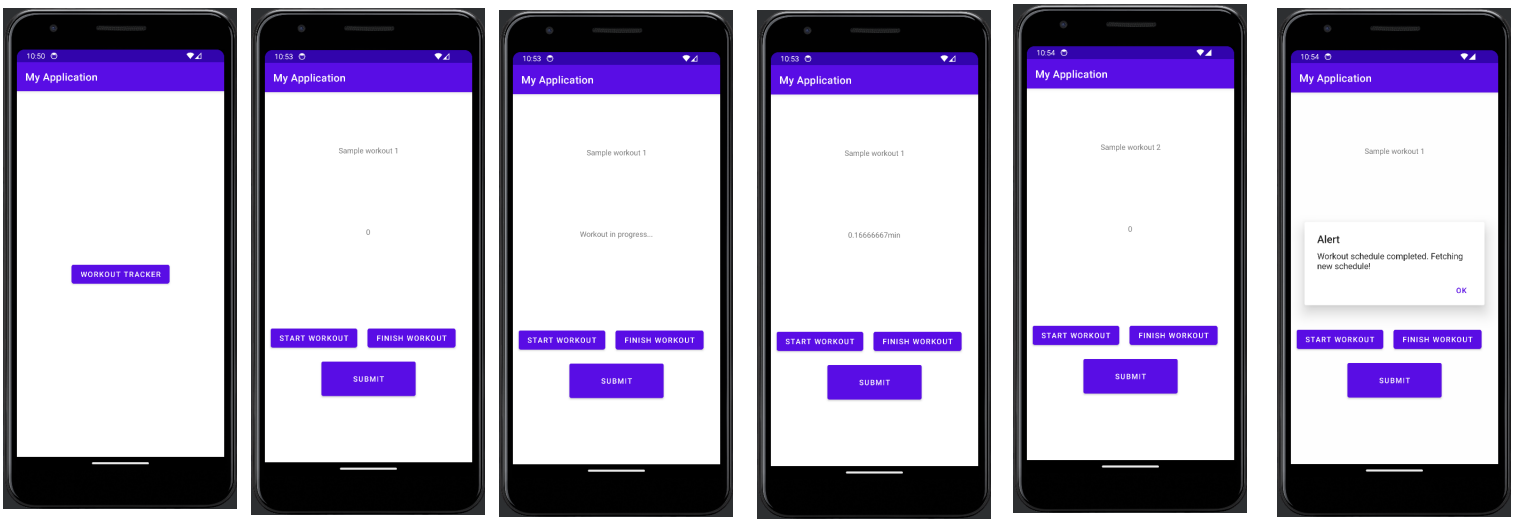
The MainActivity provides a button to view the workout tracking UI. The WorkoutTracker activity initializes DatabaseHelper, user profile, workout schedule and the listeners for Start Workout, Finish Workout and Submit buttons. When a workout completes, its duration, heart rate and temperature values are calculated and sent to CaloriesBurntPredictor to get the calories burnt during workout. This class uses a

TensorFlow Lite Linear Regression Model that was trained over an exercise and calories dataset to predict the calories burnt. This data is added to the weekly_calories table which keeps track of the total calories burnt during a workout schedule for each user. A user_workout_tracker table is maintained which stores the historical total calories burnt for each workout schedule for each user. Whenever a schedule is completed, the total calories burnt is added to the user_workout_tracker and this feedback is sent to the recommendation engine. The next schedule is then fetched from the recommendation engine.

3. Design

The feedback component takes input from the app for tracking calories burnt during a workout as well as from recommendation engine to get a workout schedule and from HealthData module to get heart rate and temperature during workout. A machine learning module is used to run the model to predict calories burnt. An SQLite DB is used to store the data collected.





The UI design is shown above. The starting page is a placeholder for integration with the main app. The workout tracker has a Start Workout, Stop Workout and Submit Button. The Workout name will be displayed on top based on the schedule retrieved. Another TextView item is used to display workout status and duration. An alert is raised when a 5-day workout schedule is completed and the next set is retrieved and the first workout in the schedule will be displayed.

Tech-stack Used

- Android SDK: For building a responsive and intuitive mobile application interface.
- TensorFlow Lite: Used to create the Linear Regression Model to calculate calories burnt during workout.
- Java: Programming language for Android app development.
- SQLite: Database
- Espresso: For conducting comprehensive UI and integration tests.
- Mockito: Used in unit testing for mocking objects and ensuring the reliability of individual components.

4. Testing Strategies

a. Unit Testing: To validate the functionality of individual components independently.
b. ML Model Testing: To evaluate the accuracy and efficiency of the neural network in predicting calories burnt. The Key metrics used are:
    i. Root Mean Square Error (RMSE)
    ii. Validation Tests: Conducted using Test split of the dataset.
c. User Interface and Integration Testing: To verify the interaction between different components of the application. Tools Used are Expresso for Android UI Testing, ensuring that the components work seamlessly together, especially in user flow scenarios like moving from account creation to personal details entry.

5. Navigating Challenges

In developing the Adaptive Fitness Guide Application, several challenges were encountered and navigated. A significant hurdle was incorporating machine learning models into the Android platform, as heavier models were not feasible for on-device predictions. This led to the adoption TensorFlow Lite, offering a lightweight yet effective solution. Developing the feedback component with expected inputs from other components required extensive collaboration, ensuring data compatibility and smooth flow. Furthermore, designing a user-friendly yet functional system to track workout completion and manage scheduling was a complex task. Simplicity in the user interface was balanced with the necessary backend sophistication, resulting in a streamlined and effective workout tracking and adaptive scheduling system.