

Exercise Break App

Ash Kandari, Ben Deschand, Nilesch Jain, Soham Pradhan

Group 29

Project Overview

The idea of this app is to use the heartbeat sensor technology found in most smart watches to calculate and find the optimal time to take a break and either get a drink or use its GPS to find a place that could sell a sports drink. This way the user takes a break at the optimal time on their bike ride or run to take a break to steady their heart rate and rehydrate during the trip.

This app was made with the intention of helping endurance trainers to those who just want to go on jog to have a better exercise experience. The hope is by making endurance exercise more fun and less strenuous, they will feel more inclined to do these types of exercises and foster healthier habits.

Project Requirements

The User must be allowed to create an account that is identified by an email and password. They must then be able to put in their age, height, and weight so the program can find the overexertion heart rate. This information will be stored locally and on a server.

The User picks what drink they are bringing and what type of exercise it is and then starts the exercise on the program, alerting the program to start watching their heart rate. The program must be able to observe the User's heartbeat through the device they are wearing, like a smartwatch.

During the time an exercise is initiated, when the User's heartbeat reaches the calculated rate, the program alerts the User that they have a high heart rate and should take a break. The User should then be able to start the break on the program, so the program can watch the User's heart rate to alert them to end the break when it lowers to a calculated lower limit.

When the program finds the User's heart rate has gone down to an acceptable range, it alerts the User to start their exercise again. It will also ask if they finished their drink.

The program should be able to access Google Maps to find the closest store to the User according to their current location.

Project Design

In this application, multiple users and a single server will be involved. We propose a system where clients are responsible for taking a user input, then sending over the input to the server, which is responsible for all the business logic and sending all the relevant information back to the client. The server will retrieve this information from the database. Therefore we think, Client Server Architecture is the appropriate choice here.

In this application, we have 2 subsystems. The first subsystem is the User Interface. It consists of LoginPage, which contains the login. Inside the UserDashboard we can start our exercise. Upon starting exercise, we have the option of ending the exercise or starting a break in ExerciseDashboard. If we start a break, we have the option of ending a break inside BreakDashboard. The second subsystem is Application. In this, we have the User details such as Height, Weight, Age. After the user starts an exercise and takes a break, we record the starting time of break, ending time of break, and drink from the Break Class. We also use the Route Class to check the attributes of the route the user has selected in order to calculate the distance of the selected route.

The primary UI that takes care of users using the application and interacting with the application, it does so from the features available in the smartwatch. These features of the UI include prompt for the break, starting and ending break, prompt for locating a store and navigating a route to the store. The smartwatch is connected to the smartphone with a Bluetooth connection. The smartphone is where the calculation for the break is done. The smartphone however is connected to both a server and google map application. The server holds the User Information stored on the server which is relayed back and forth to the application on the smartphone, whereas the Google Maps is a third party application that is used for navigating a route to local nearby stores. Both Google Maps and server are connected through HTTP for proper encryption and secure data transfer of necessary information and functionality.

We believe that since this application is primarily going to be a mobile application, the developers would need to use Android Studio for making android application and Swift for iOS. We recommend using IntelliJ IDEA as the IDE for Android Studio and XCode for iOS. We recommend deploying the code on the AWS Server, and using Amazon S3 for creating the database. We also recommend using a relational database management system such as MySQL since the data which is currently being stored in the database is heavily structured.

Figure. 1 -Activity Diagram

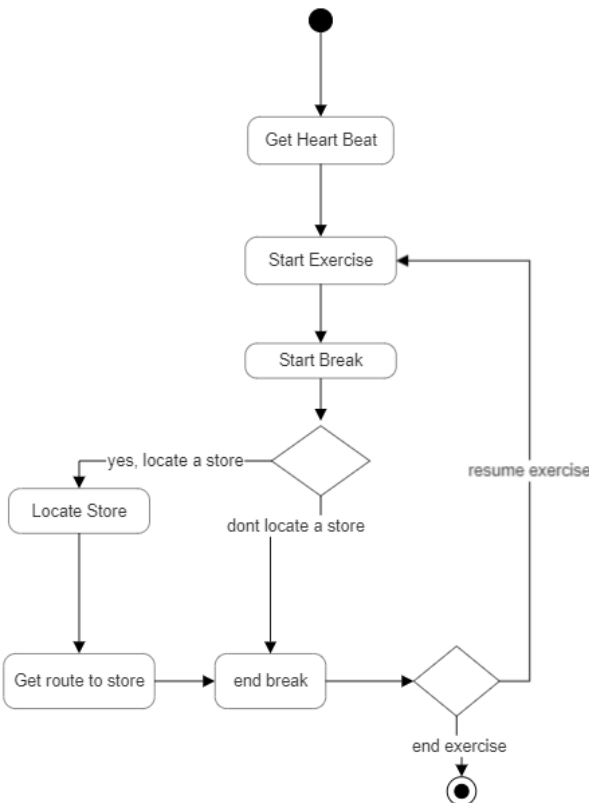


Figure2 - Deployment Diagram

