Tyme Wear iOS App Project Proposal

Team: Roland Gill (PM)
        Rachel Hart
        Anwesh Joshi

Client: Juan Morales, Tyme Wear

I. OVERVIEW

Our project group will be working with Juan Morales on his Tyme Wear smart clothing technology. The product centers around a smart shirt that contains sensors to track movement and breathing as the user exercises. The sensor data is fed into a removable electronic node in the shirt. The node is Bluetooth enabled, which allows it to communicate the information gathered to an iOS App. The data gathered by the user can be used to calculate anaerobic and aerobic thresholds, and ultimately provide the user with customized coaching and a personal training plan.

II. PURPOSE: iOS APP

We will be working on completing development of the companion iOS app. The algorithms for processing user data are stored in a cloud in the back-end. The iOS app must be able to communicate back and forth between the back-end to give the user feedback and calculations based on their performance. The processed data is used to generate charts of progress that can be reviewed by the user. Although we are a separate team from that working on the back-end, we will need to work closely with that
team to develop cohesive communication between the user-facing app and the processing algorithms on the back-end.

III. GOALS

Our main goal is to integrate and optimize this app/backend pipeline. Once the data is processed, the app should display the results to the user. Ideally, a map feature will also be included. There was originally a possibility of a front-end team, but there is not one at this time. Still, we may need to do some integration with the existing online website if need be, though most of that would probably be done on the back-end. An original version of the app has been given to us and we will be building onto the existing code. We will need to study what we have so we can properly integrate with the back-end. Additional UX updates and improvements will also be made as needed throughout the project.

IV. STACK

Since we are building onto an iOS app, we have limited options on the stack. The existing portion of the app uses Objective C, so we will be largely working with that. Swift is also compatible with iOS, and we have team members who have worked with it before, so portions of the program may be written in Swift. The app uses a BLE (Bluetooth Low Energy) library to handle the Bluetooth functionalities. A number of PoDs (AFNetwork, AWS, Chameleon, Charts and Cocoa) are included and will aid us in get and post requests, data collection, UI, generating graphs of collected data, and managing dependencies (respectively).