

# Specification for DynaMap

## 1 Overview

DynaMap is an Android application that allows users to view the activities, specifically, the type of motion, of their friends on a Google-Maps-like interface. The app develops a motion classifier tailored to each user in the initial phase of usage by actively collecting accelerometer data and asking the user to label the type of activities associated with the data. It then uses this trained classifier to predict the type of motion the user is currently carrying out and outputs this classification along with the user's location on the map of the user's friends. The app also outputs aggregate statistics on the user's daily activity level in order for the user to track the amount of exercise that he or she is getting every day. The goal of this app is to encourage people to lead a more active and healthy lifestyle and use the activity levels of friends to motivate the user to strive towards this goal.

## 2 Welcome Screen

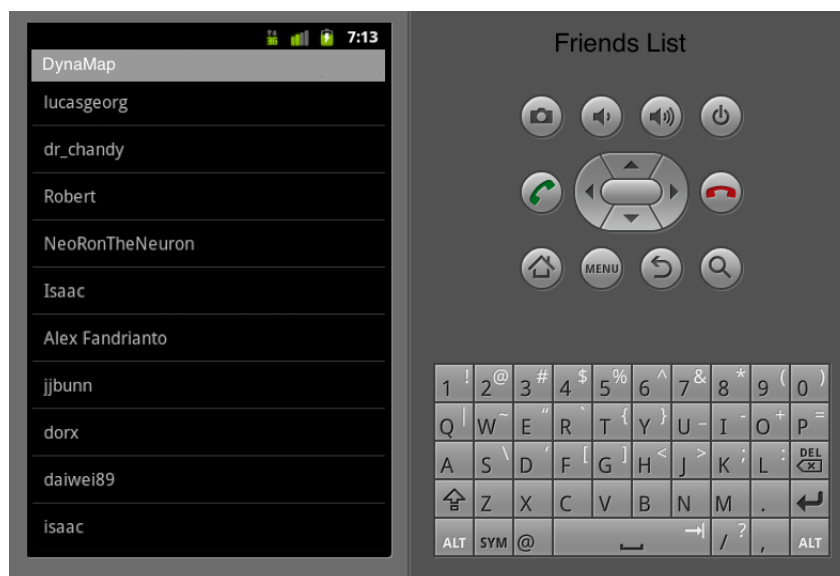
Upon launching the application, the user will be asked to either create an account or login with an existing account. Logging in with a personal account is absolutely crucial for this app since the app works with a lot of user-specific content.

## 3 Main Menu



As seen in the mockup UI above, the main menu allows the user to navigate either to the friends list, the screen for motion detection (manage recordings), or view the map on which their friends activity levels are displayed. In place of the map on the current mockup, the user should see their own profile with statistics regarding their activity level.

## 4 Friend List



On the friend list screen, the user would see a list of their friends that they can click on to view their profiles and have the option to unfriend or block the person. There would also be a search box to allow users to either search for new friends to be added to their list or for a particular friend on the existing list, much like how the Facebook search box works.

## 5 View Map

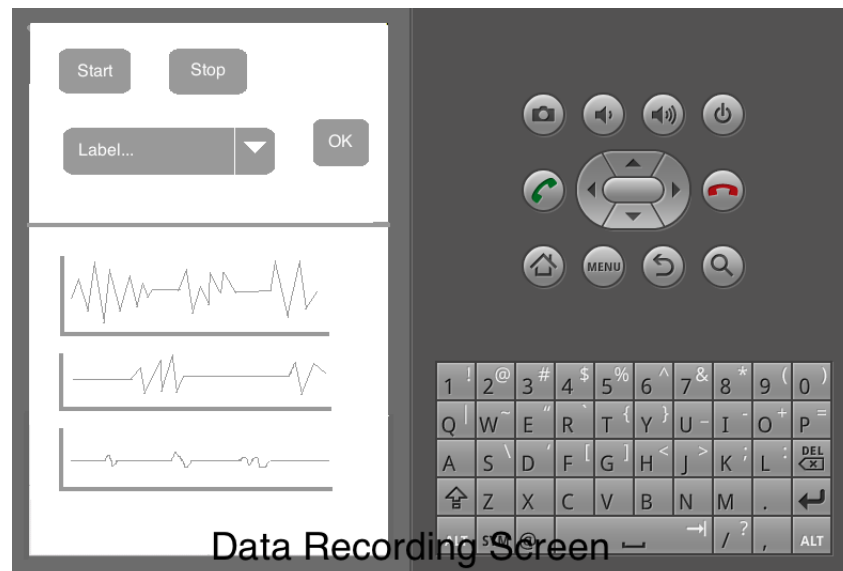


This is a map view of the friends and their activity levels. Each type of icon corresponds to an activity level, and we provide a legend at the bottom left hand corner of the map. Each icon is hyperlinked to the friend's profile, and a box displaying some basic information regarding the friend would pop up when the user hover over the icon in the map view.

## 6 Machine Learning for Android Motion

### 6.1 Data collection

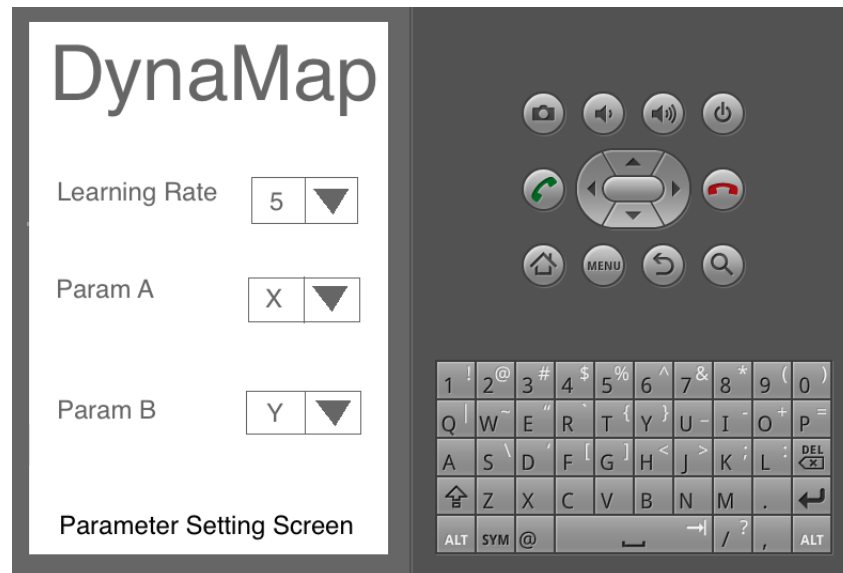
The app attempts to collect accelerometer data constantly in the background and deposit this data in the server at a regular time interval to avoid overwhelming the Android phone with data storage overflow. In the initial phase of running the app, the user will be prompted frequently to annotate the data with the type of activities they were engaging in at the time of collection.



As shown in the UI mockup above, the user will be shown the actual time series and asked to identify the start and stop time of a specific segment of data and choose a label for that data from a dropdown menu.

### 6.2 Learning and Classification

The training of the classifier takes place in the server. The server creates a unique process for each user and trains on the data submitted to the server by the user's Android phone. The server then sends the trained classifier to the user for classification. Since an online learning algorithm is used for training the classifier, the weights representing the classifier would be updated at a regular interval. The classifier is then used to predict the type of activities associated with unlabeled accelerometer data once we're able to establish a certain level of accuracy in the classifications, and we rely on the classifier to identify user motion for the friends map as well as providing statistics on the user's average activity level. The user is periodically prompted to provide feedback on the performance of the classifier, and more training data will be collected to update the classifier accordingly. The user also has the option of manually setting some of the learning parameters to customize the classifier.



The parameters will be packaged and presented in such a way that a person who's not familiar with the concept of machine learning would be able to make a decent judgment on the range of parameters suited for their lifestyle.