Time Trial Tracker

By Terry Ridgway

Overview

My bike club runs a Bike Driving Academy and one of the exercises we have the student perform is a time trial. Unlike a normal bike racing time trial, where the participants are attempting to ride a course in the fastest time, the Academy time trial is meant as a consistency exercise.

Riders navigate a 2.2 mile circuit with the aim of riding each of 5 laps as consistently as possible.

The basic requirement of an application is to time each of these laps and then calculate the mean difference between each of a rider's laps and the average of their lap times. The rider with the lowest mean difference is considered the most consistent rider.

This App has already been coded up for Android and I'm using the idea as my iPhone App project because the user base at my bike club is split between iOS and Android users, so it is an advantage to have the App for BOTH platforms.

Database Scheme

Table: RIDER

Columns:

RiderNumber	NUMBER
Rider Name	TEXT
Last Seen Time	NUMBER
ETA	NUMBER
Average Lap Time	NUMBER
Mean Difference	NUMBER
Standard Deviation	NUMBER

Table: LAPS

Columns:

LapID	NUMBER
RiderNumber	NUMBER
LapSplit	NUMBER

>> ONE Rider has MANY Laps.

User Interface

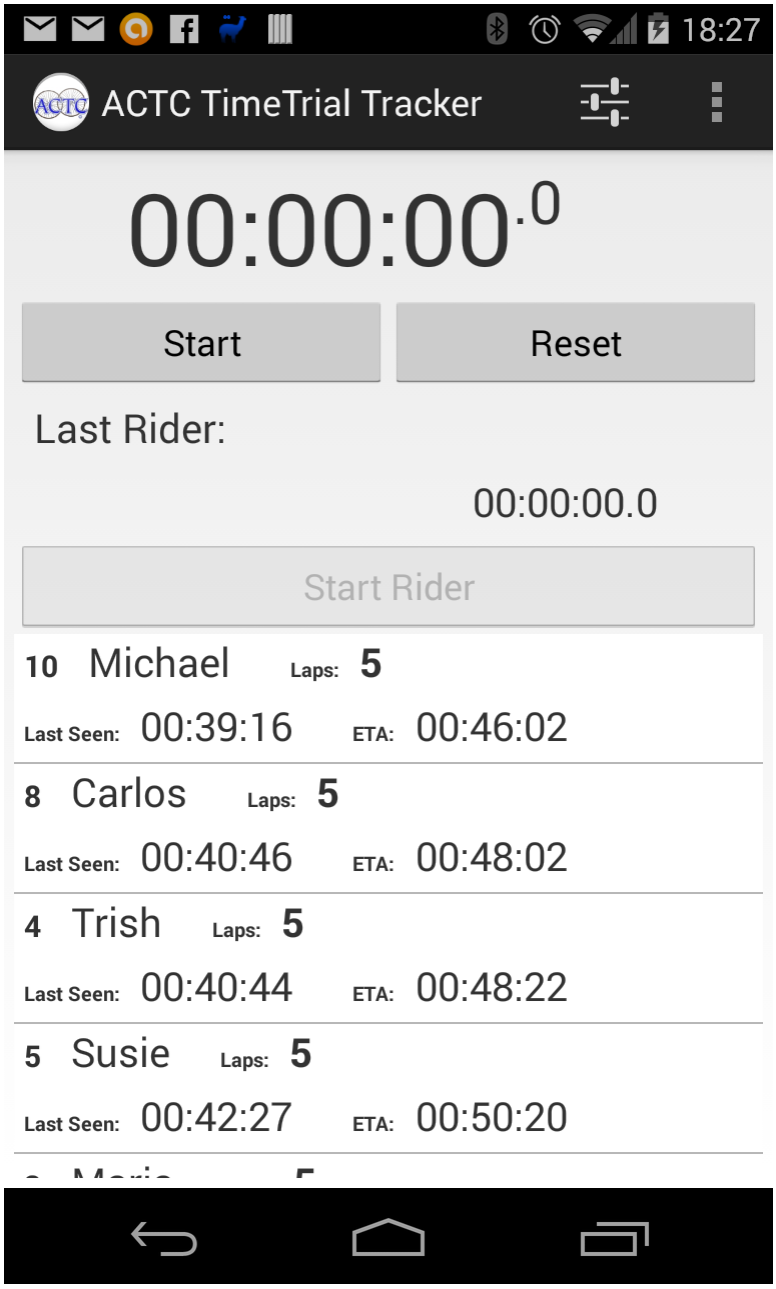

Fig 1.

In Fig 1. We have the main Application screen.

We have a main timer that begins counting up after the user clicks the Start button. Once the Start button is hit, the Start and Reset buttons are replaced by a Stop button, which required a long press to activate, rather than a short press, so as to reduce the likelihood of an accidental shutdown.

Once the timer has been started, the first rider in the list is automatically selected and you're ready to start the riders. Initially the rider list is ordered by the rider number. As the riders are started, the sorting comes from ETA and Last Seen times, so that riders drop to the bottom of the list initially and the non-started riders bubble to the top.

The Last Time and Rider Name area above the Start Rider button show the most recently seen rider and their last seen time.

As the riders come around their laps, we add new lap times to the database. The rider last seen value is updated and an ETA is calculated based on the average time to ride a lap and the last time the rider was seen. At the same time the Mean Difference and the Standard Deviation values are also recalculated and the updated values are written to the Rider record.

Because of the sorting of the SQLite query the rider list is ordered such that the top rider in the list is most likely to appear next.

Selecting a rider in the list and clicking the Start Rider button adds a new lap record for that rider and the current elapsed clock time.

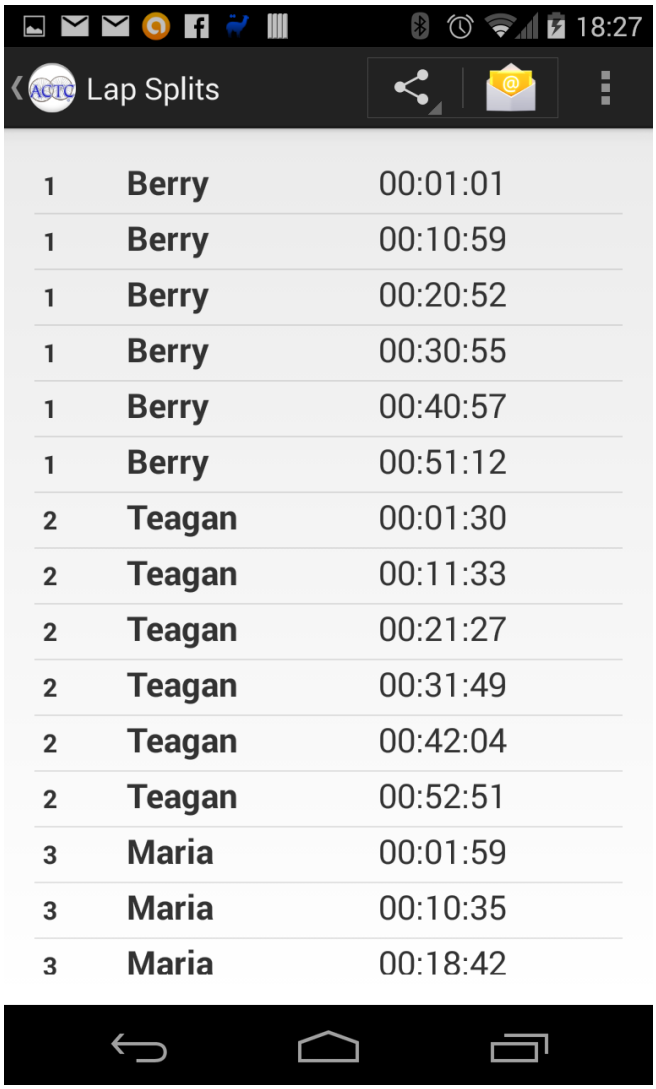

Fig. 2 – Lap Splits view

The Lap Splits screen shows the scrolling table view of riders and their recorded times, in order of rider number and time, so that these values can be viewed and recorded.

Also, this screen allows you to share the data via any text-based sharing method available on your device.

Fig. 3 – Rider Statistics view.

On this screen we see the Rider number, rider name, their average lap time and the standard deviation of their lap times versus their average time. I also propose adding the mean difference value to this screen.

Also, this screen allows you to share the data via any text-based sharing method available on your device.

Technical Implementation Notes

The Application has to store data for the riders and their lap times. The data needs to be displayed to the user with the ability to edit the elements of the data.

I'll implement the Data storage using the sqlite database packaged with the iOS operating system.
I'll be using table views to display the results from the database queries.
I plan to use the NSTimer class to handle the stopwatch processing portion of the application.
I'll be using menus, buttons and navigation controllers to move around the application scene.