GitHub Username: genericjohndoe

Go!

Description

Go! is a voice-activated timer. The idea was conceived while exercising. Isometric exercises sometimes require keeping your hands in a specified position for an extended period of time, making starting the timer cumbersome at times. Go! obviates this problem by allowing the user to get in position and then starting/stopping the timer via a voice command.

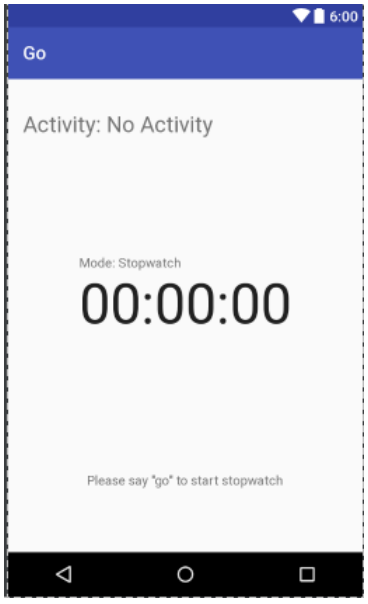
Intended User

Anyone who can see the use of a voice activated timer.
Active people – for isometric exercises and tracking running times

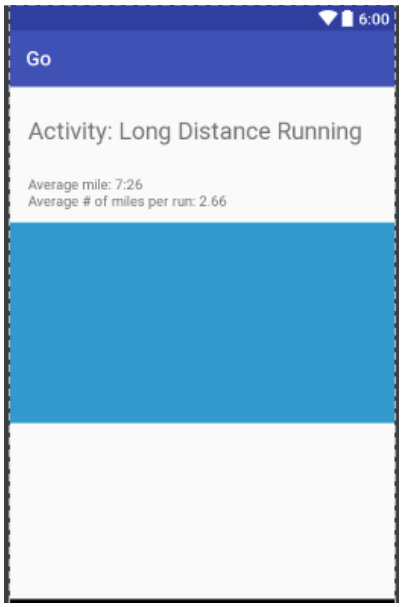
Features

- *Voice activated stopwatch*
- *Tracks distance and pace of runs*
- *Logs fitness info*
- *Has graphs that shows changes in logged data over time*

User Interface Mock



Description: This is a sample activity made with Android Studio. There are two modes (tentatively), that being stopwatch and timer. The "Activity: No Activity" textView shows the activity chosen/made by the user.
The textView at the bottom of the screen is an instruction prompt. When the text changes color, that will be the queue to speak.



This screen shot shows a possible view of the widget/info fragment. The blue rectangle represents the graph showing how the average mile times differ on a trial basis. Numerical data populated via loader

Key Considerations

How will your app handle data persistence?

Data saved in SQLite database and accessed via content provider

Describe any corner cases in the UX.

MainActivity will show timer originally. ViewPager will be use to show the fragment with user data

Describe any libraries you'll be using and share your reasoning for including them.

- *MPAndroidChart by PhilJay. This library will be used to graph user data.*
- *Schematic by SimonVT. This library will be used to set up data storage*

Describe how you will implement Google Play Services.

- *Admob will be used to generate income. Interstitial ads will be used in the main activity*
- *maps/location services for tracking distance ran, average speed, and path ran*

Next Steps: Required Tasks

Task 1: implement voice recognition api

- Add voice recognition API to enable starting/stopping of timer/stopwatch

Task 2: Create fragment to show user data

- Add MPAndroidChart dependency

Task 3: Create SQLite database

- Create SQLite database to log info for long distance runs

Task 4: add finishing touches

- Implement location services and maps api
- Implement admob
- Add weather data using the Open Weather api to help user decide what to wear before their workout, showing new weather data everyday with use of sync adapter