Time Tracker Feature List

# Timers

This view only works in portrait mode.

This is the primary view of the app. The goal is to show a list of timers, their status (running/not running), and provide the ability to start and stop them with a single tap.

There are two layouts (which can be switched between in Settings). The default is a grid view, with multiple timers per row. The other option is a list view, with one timer per row.

This view will also be shown in a notification center today widget.

# Timeline

# This view only works in landscape mode. It is activated by changing the orientation of the device in the Timers view (changing the orientation in any other view does not show this view).

The goal of this view is to allow a user to edit previously recorded time windows.

# CRUD Clients

The goal of this view is to allow a user to create clients. Clients are the top level object to organize timers. Clients can have multiple projects, and each project has one timer associated with it.

We should allow a user to view, create, update or delete a client.

Each client also has a color associated with it. It will be the background color of each timer.

# CRUD Projects

This view is accessed by navigating to a project from a list of projects in the CRUD clients view. The goal of this view is to allow a user to view or edit the details of a project.

A project has a color associated with it. It will be the foreground color of its timer.

# Reminders

# Each project can have a set of reminders associated with it. Reminders can be sent by:

* Server side push notification
* Local time based reminder (i.e. remind the user to start/stop a timer at a given time)
* Local location based reminder (i.e. remind the user to start/stop a timer when the enter/leave a location)

# Triggers

# Triggers are like reminders, except that instead of reminding the user to start or stop a timer, they start or stop a timer automatically. These cannot be sent by a server side push notification, but can be triggered based on time and location.

# Login

# Users have to log in to their account to be able to use Time Tracker. Without a valid account, the app does nothing.

# Tutorial

If a user has never logged into the app when it is launched, we should show them a tutorial of the app, and allow them to skip it if they’d like to.

If a user is not currently logged in, but a user was at some point logged in, we should show the login screen with an option to go through the tutorial again.

The tutorial must also be accessible on the Settings screen.

# Analytics

We need to track how users use the app (anonymously) to enable us to make key decisions. For instance, if we make grid view the default view for timers, how many people change the setting to use list view? To start with, we will start reporting these analytics to TimeTrackerWS instead of using a built in analytics package (we can’t really trust what they do since they are black boxes).