GitHub Username: delorenzo

Hit The Gym

Description

This app allows you to track your progress at the gym to see your gains over time, and helps you build a workout schedule to be successful towards your fitness goals.

Intended User

Gym users, runners, fitness enthusiasts

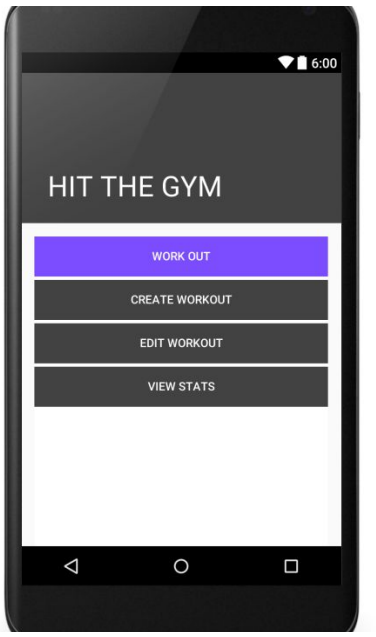
Features

- Create, save, and edit workouts
- Graphs progress with weight, distance, speed, etc over time
- Tracks frequency of gym visits

User Interface Mocks

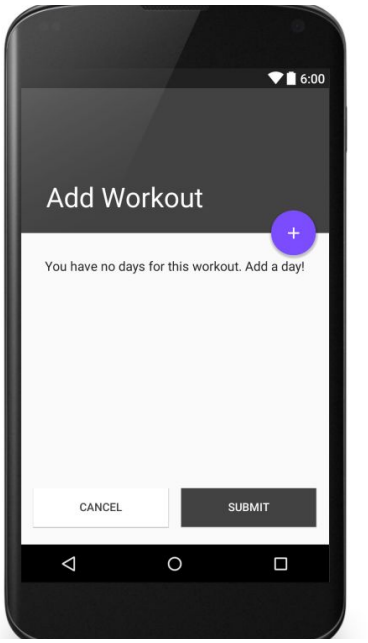
These can be created by hand (take a photo of your drawings and insert them in this flow), or using a program like Photoshop or Balsamiq.

Screen 1



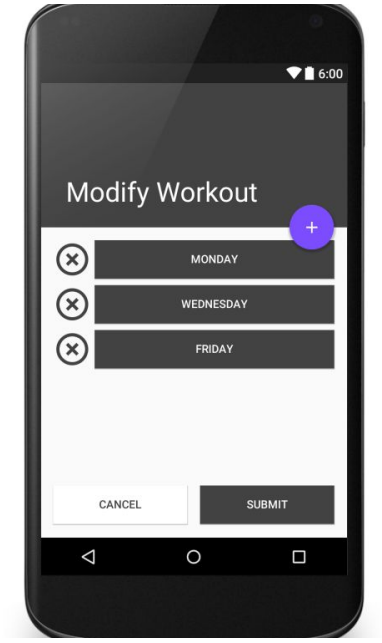
Main screen - navigate to workout, editing a workout, or looking at statistics

Screen 2



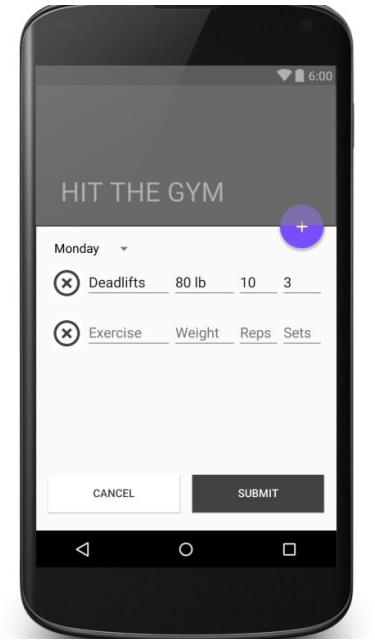
Add Workout - selecting a day brings you into modifying that day, and you can easily add or remove days from this view.

Screen 3



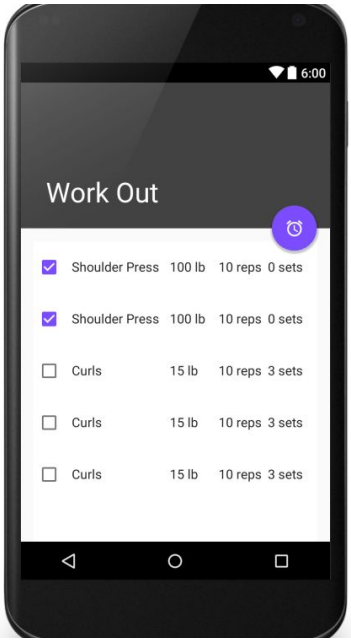
Modify workout - selecting a day brings you into modifying that day, and you can easily add or remove days from this view.

Screen 4



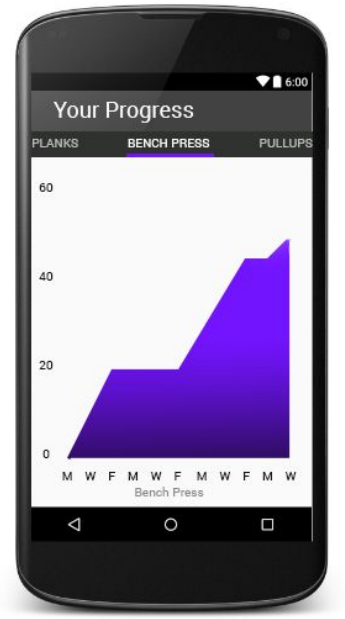
Add or modify day - selecting a day or adding a day brings you into this view, which can be displayed as a fragment or a floating dialog based on the device size, and you can easily add or remove exercises from this view. The weight units can be modified in settings.

Screen 5



Workout - selecting the alarm FAB starts the rest timer between sets, configurable by the user. Selecting a checkbox indicates 1 set completion, decrementing the number of sets (the checkbox only remains checked if all sets are completed). To end the workout prematurely, an option can be selected from the menu. From this view if the weight or rep count change by set number (say, the user uses increasing weight throughout the workout) the list will be updated on the checkbox selection.

Screen 6



View Stats - view graphs of your progress at various exercises. You can also view your consistency in frequenting the gym and other metrics.

Key Considerations

How will your app handle data persistence?

I will build a content provider and store data in an SQLite database.

Describe any corner cases in the UX.

1. “Edit workout” does not appear as an option if no workouts exist
2. In an empty workout in the add or edit workout UI, a text view instructs the user to add a workout day.
3. The statistics displays an instructive text view if no data exists.

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

I’ll be using ButterKnife for cleaner UI code and the v7 and design support libraries for backwards compatibility. I’ll also use GraphView for graphing workout statistics.

Next Steps: Required Tasks

This is the section where you can take the main features of your app (declared above) and decompose them into tangible technical tasks that you can complete incrementally until you have a finished app.

Task 1: Project Setup

- Configure libraries

Task 2: Implement UI for Each Activity and Fragment

- Build UI for MainActivity
- Build UI for ModifyWorkoutActivity
    - Build UI for AddWorkoutFragment
    - Build UI for ModifyWorkoutFragment
    - Build UI for AddDayFragment
- Build UI for WorkoutActivity
- Build UI for ViewStatsActivity

Task 3: Data Persistence

- Create data contract
- Build DB Helper
- Build content provider
- Build adapters to use with Loaders in the UI

Task 4: Graphing Data

- Figure out how the GraphView library works
- Decide on the data that needs to be collected
- Graph the data in a visually pleasing way consistent with the app theme
- Add configuration options to the statistics

Task 5: Build Widget

- Create an informational widget for the home screen that displays the current or upcoming day's workout

Task 6: Integrate Google Analytics

- Configure Google analytics to measure app crashes and exceptions.

Task 7: Integrate App Invites

- Leverage app invites to allow users to easily invite their friends to use the app.

Task 6: Signing Application

- Create keystore
- Create private key
- Add signing configuration to the build file
- Build with assembleRelease