[Description](#_sm4ra97uwo11)

[Intended User](#_aws88pzfmqca)

[Features](#_zheq5430xrpq)

[User Interface Mocks](#_giquerrw6g46)

[Screen 1](#_a4jdupabry3k)

[Screen 2](#_dpcbbkx5yry)

[Key Considerations](#_gvcvmae8jn8u)

[How will your app handle data persistence?](#_v8my7nhtvz0m)

[Describe any corner cases in the UX.](#_gw69vjn1ico0)

[Describe any libraries you’ll be using and share your reasoning for including them.](#_6yqqubmw5bs)

[Describe how you will implement Google Play Services.](#_qrxg682nywe6)

[Next Steps: Required Tasks](#_v518bncmggeg)

[Task 1: Project Setup](#_8oe8zpk3qsmp)

[Task 2: Implement UI for Each Activity and Fragment](#_rzllsk6uqztx)

[Task 3: Your Next Task](#_fdmohs7hes)

[Task 4: Your Next Task](#_umfwsvmx7tpn)

[Task 5: Your Next Task](#_kjidlkq4xm3u)

**GitHub Username**: jerielng

FitMe

# Description

FitMe provides users of all fitness levels a tool that will help generate workouts based on selected inputs such as muscle group, equipment available, desired duration, etc. Data is pulled from the Workout Manager API (<https://wger.de/en/software/api>).

In addition, since exercise is a regular activity that requires consistency to progress, users can save their workouts to return to them during future sessions.

With this app, users will also be able to find nearby gyms to streamline the starting process instead of having to worry about the details.

With a clean, intuitive user interface, users can spend less time figuring out what exercises to do and focus on the routines themselves.

# Intended User

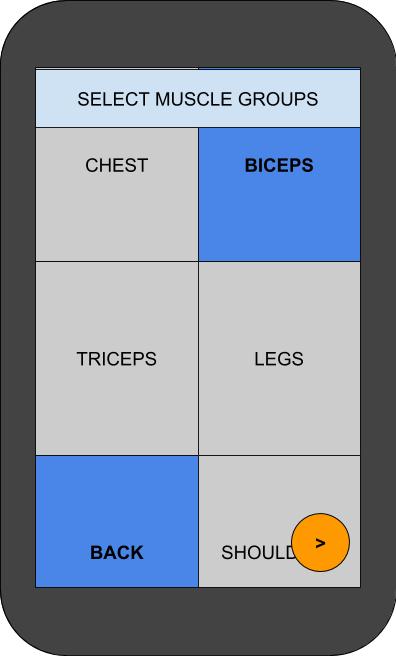
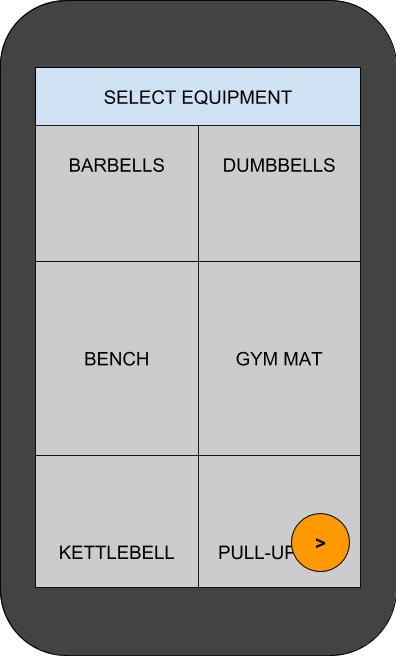
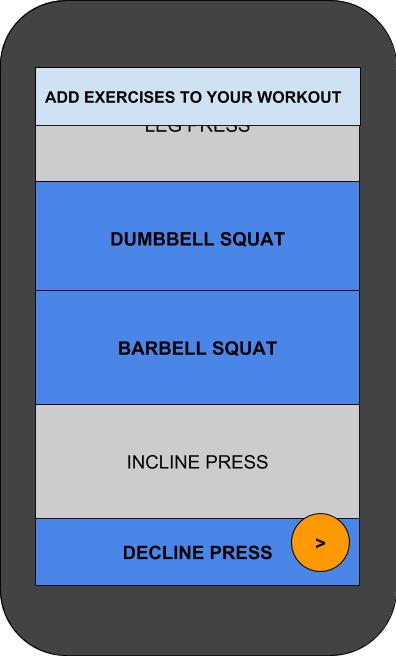
Target users will include people of any fitness level who wish to take a serious approach to their exercise regimes.

# Features

* Accepts user-selected parameters to generate workouts
* Saves favorite workout plans
* Find nearby gyms based on user-input location

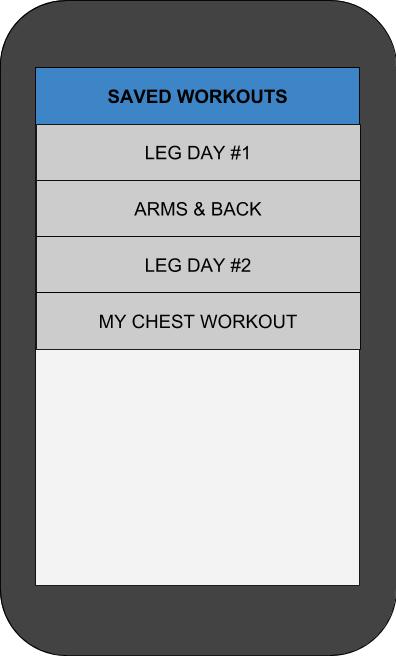
# User Interface Mocks

## Generate a Workout

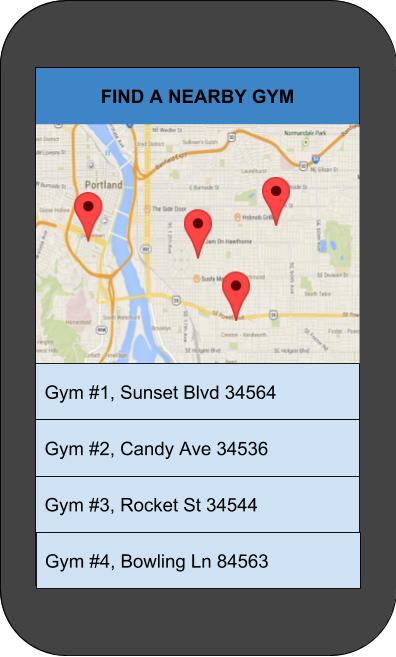
This fragment will follow a simple flow that allows users to select input muscle groups and equipment to generate available exercises that they can select and add to their custom workout.

## View Saved Workouts



When users have created their own custom workouts, they can view the list of saved workouts and return to them in future sessions.

## View Saved Workouts



When users have created their own custom workouts, they can view the list of saved workouts and return to them in future sessions.

# Key Considerations

### How will your app handle data persistence?

Data persistence will involve connecting to a SQLite database using a ContentProvider. This will be used to save stored user workouts.

### Describe any edge or corner cases in the UX.

Navigating between each “Activity” (implemented as Fragments for this app) will be handled with a Bottom Navigation Bar.

When a user generates a workout, they will be able to follow the flow until the final screen, which will allow them to add any exercises they wish to their workout. At this screen, the submit button will take them back to the beginning of the generate fragment.

For the other two fragments (finding a gym and viewing saved workouts), they will only be one-screen fragments, although the detail view for saved workouts will be able to return back to the parent via the up button.

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

The app will use ButterKnife to focus more on logic code than having to bind each view manually. It will also use Picasso to easily load images from the workout API into their respective ViewHolders.

### Describe how you will implement Google Play Services or other external services.

Maps, Location, and Admob will be used in this app. Maps will provide the location information of all the nearby gyms based on the user’s current Location, or from a given input location. Admob will be used throughout the application, sparingly, to provide users with ads.

# Next Steps: Required Tasks

## Task 1: Project Setup

You may want to list the subtasks. For example:

* Create project in Android Studio
  + Add a Bottom Navigation Activity as the MainActivity
* Configure libraries
  + Add ButterKnife dependency for easy view binding
  + Add Picasso dependency for image loading capabilities

## Task 2: Implement UI for Each Activity and Fragment

* Create Fragments for each feature of the app
  + Generate a workout
  + View saved workouts
  + Find a nearby gym

## Task 3: Implement tasks to retrieve and load data from external API

* Add RecyclerView to Saved Workouts Fragment
  + Create adapter and layout manager for RecyclerView
* Extend AsyncTask class to fetch data from Workout Manager API

## Task 4: Add data persistence via a SQLite database

* Implement ContentProvider and SQLite database helper
  + Connect to SQLite database

## Task 5: Use Google Play Services to help with location-based features

Describe the next task. List the subtasks. For example:

* Create layout
* Something else