

Description

Intended User

Features

User Interface Mocks

- Main Activity View with RecyclerView

- Saved Workout View

- Map View

- Tablet Views

- Widget

Key Considerations

- How will your app handle data persistence?

- Describe any edge or corner cases in the UX.

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

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

Next Steps: Required Tasks

- Task 1: Project Setup

- Task 2: Implement UI for Each Activity and Fragment

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

- Task 4: Implement Data Models and Content providers

- Task 5: Implement Google Play Services

- Task 6: Designing for Tablets

- Task 7: Add Widget

GitHub Username: AI3xSav

Stay Fit

Description

Stay Fit is an app for passionate in Gym. It provides users a variety of options to use while exercising such as muscle groups, equipment to use, exercises to do with the possibility to save data. This way users can create a weekly schedule plan for the workout.

In addition, the progress of the workout can be saved so that users return to them during future sessions.

Also, with this app users can search for nearby gyms through Google Maps, instead of worrying to find one.

Data will be pulled from the Workout Manager API. (<https://wger.de/en/software/api>).

Intended User

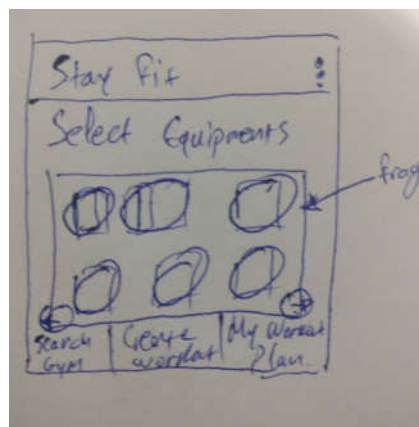
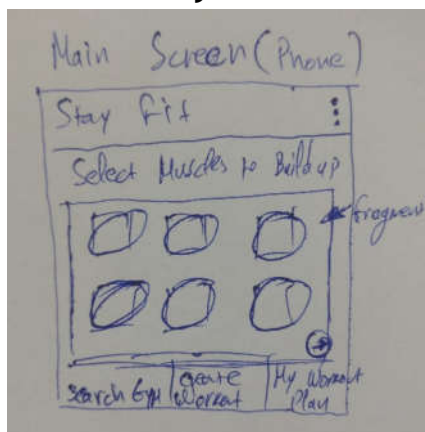
Any user interested in exercising can use this app.

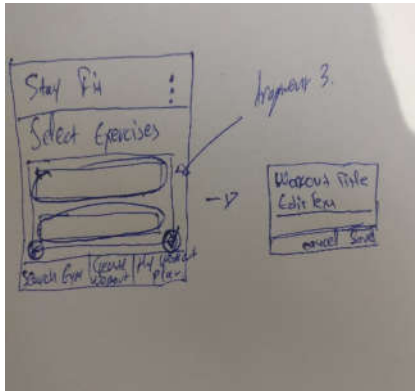
Features

- List of muscles to build, equipment to use and exercise to do.
- Saves information of your workout plans
- Search gyms through Google Maps based on device's location
- Generate workout plans given the parameters by the user
- Widget that provides the chosen exercise

User Interface Mocks

Main Activity View with RecyclerView



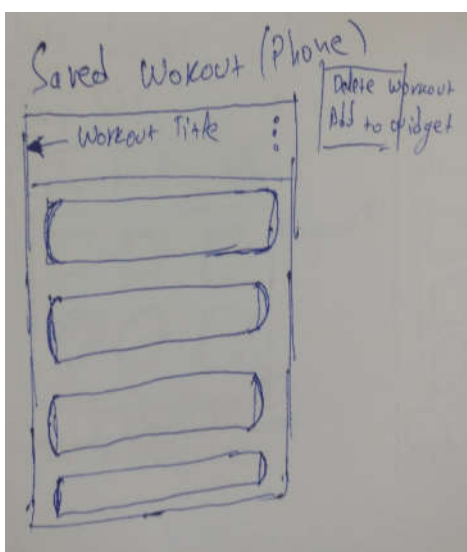


Main Screen that includes a fragment (RecyclerView) with Muscles to select and with bottom navigation between Create Workout, Search Gym and My Workout Plan.

After selecting Muscles, fragment changes to Equipment selection (still on Main Screen of the app)

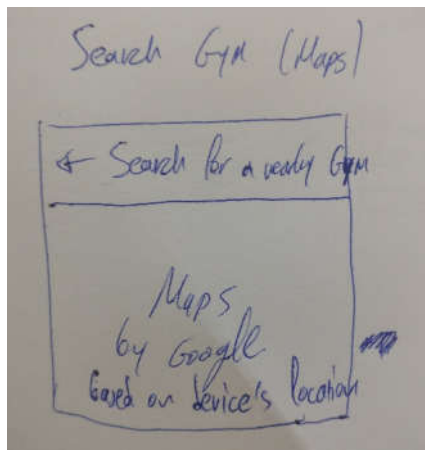
Finally, changes to Exercises selection fragment based on the 2 previous choices (Muscles & Equipments) with the option to save the custom workout and later add to widget.

Saved Workout View



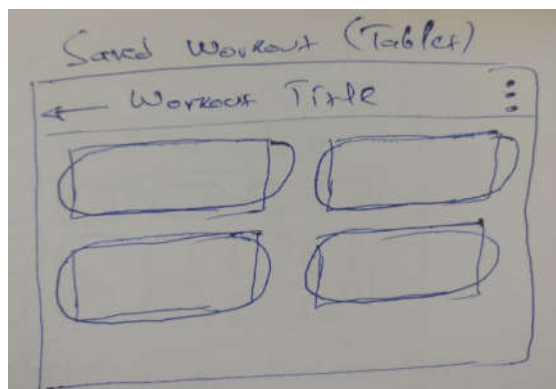
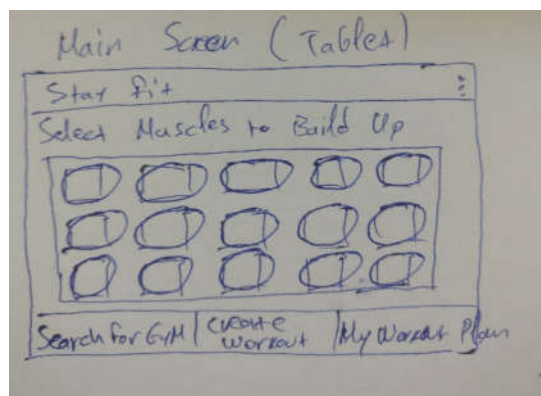
With the option of My Workout Plan on bottom navigation, user can see his saved custom workout(s).

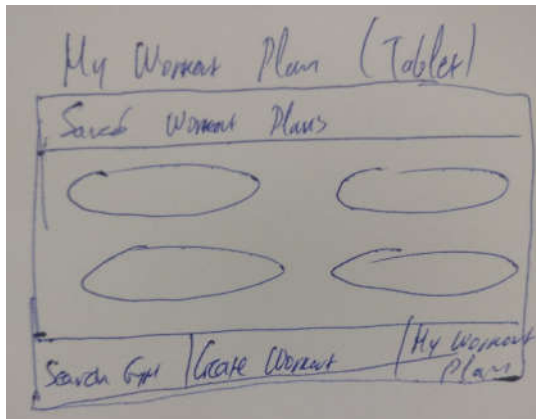
Map View



By allowing the location of user's device, the nearby gyms can be found with the option of opening Google Maps. This can be selected from the bottom navigation in the Main Screen by pressing Search Gym.

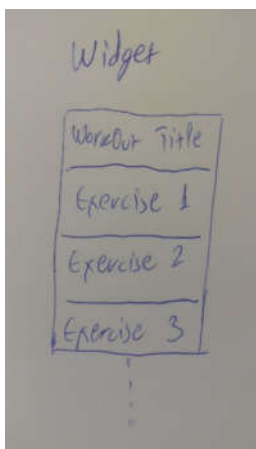
Tablet Views





Same functionality with the Phone's Screen but more appropriate for a tablet, utilizing all the available extra space on tablet's screen.

Widget



Users can add any of their saved workouts to widget to their lock screen. By clicking on it, app launches into their saved workout.

Key Considerations

Common Project Requirements

App is written solely in the Java Programming Language.

App keeps all strings in a strings.xml file and enables RTL layout switching on all layouts.

App includes support for accessibility. That includes content descriptions, navigation using a D-pad, and, if applicable, non-audio versions of audio cues.

How will your app handle data persistence?

Data persistence will require a connection with SQLite database by extending with Content Provider.

Describe any edge or corner cases in the UX.

App is navigating between each Activity- Fragment with Bottom Navigation Bar.

When user creates a workout, they will be able to follow the flow until the final screen to save the chosen exercise.

User can search for a gym based on device's location through navigation button that starts the activity for Find the nearest Gyms.

Also, by pressing the navigation button My workout plan, user starts the activity with the details of his saved exercises.

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

App uses Picasso. – stable version: 2.71828

App uses databinding. – stable version: 27.1.1

App uses RecyclerView-v7. – stable version: 27.1.1

App uses AppCompat-v7. – stable version: 27.1.1

App uses GridLayout-v7. – stable version: 27.1.1

App uses Gradle version: 4.4

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

App uses Location to get device's current location.

App uses Maps to get nearest gyms from the device's Location

App uses AdMob for banner ads

Next Steps: Required Tasks

Task 1: Project Setup

- Update Android Studio to the latest Stable Version
- Create project in Android Studio
- Add a Bottom Navigation Activity for the MainActivity
- Setup Gradle Dependencies

- Configure libraries
- Add permissions to manifest (such as Internet)

Task 2: Implement UI for Each Activity and Fragment

- Build Fragments for each feature of the app (such as Muscles to build, Equipment to use)
- Connect each Fragment to the MainActivity
- App uses an AppBar and associated toolbars
- App provides individual RecyclerView for each fragment
- App provides Map placeholder for Gym Fragment
- App provides a widget to provide relevant information to the user on the home screen

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

- Extend AsyncTask to fetch data from Workout Manager API
- Create Network Utilities to build URLs and make API requests
- Create Connection Utilities to check Internet Connectivity for every required action

Task 4: Implement Data Models and Content providers

- Create Data models
- Create SQLite db
- Create Loaders and Adapters

Task 5: Implement Google Play Services

- App uses Admob with placeholder views
- App uses Location feature for getting device's current location
- App uses Maps feature with map placeholder to find nearby gyms (combo with Location)

Task 6: Designing for Tablets

- Design app for tablets
- Check compatibility

Task 7: Add Widget

- Setup Widget to display given exercises