GitHub Username: greedypinky

You can do it everyday exercise

Description

No time to go to gym but want to do easy exercise at home, this app updates exercise list from the cloud regularly and user can save the favourite exercises list.
Provide easy steps and videos to guide you through.

Intended User

People who do not have much time to do exercise but want to stay healthy and strong by doing easy exercises.

Features

List the main features of your app:-

- allow user to update exercise list regularly and notify the users the list is updated.
- allow user to mark exercise as the favourite exercise
- allow user to share the exercise to family and friends.

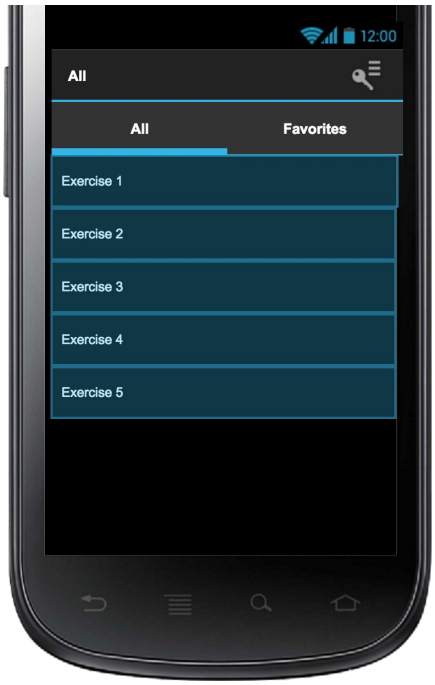
- allow user to select the favourite exercise and display the steps in the Widget.

User Interface Mocks

Screen 1

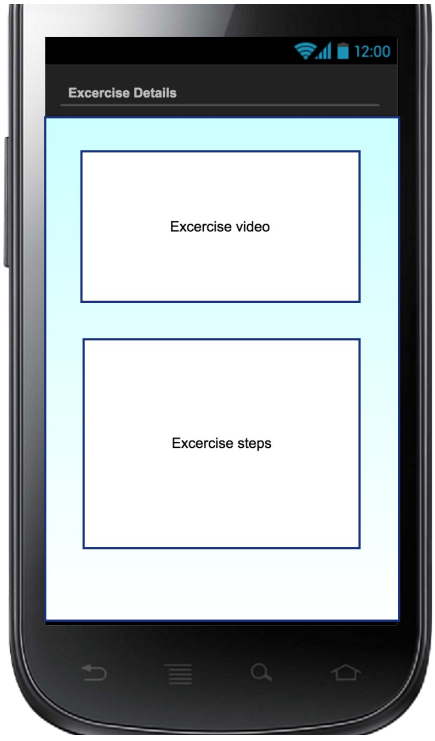
Main activity which shows the exercise category as the CardView.

Screen 2



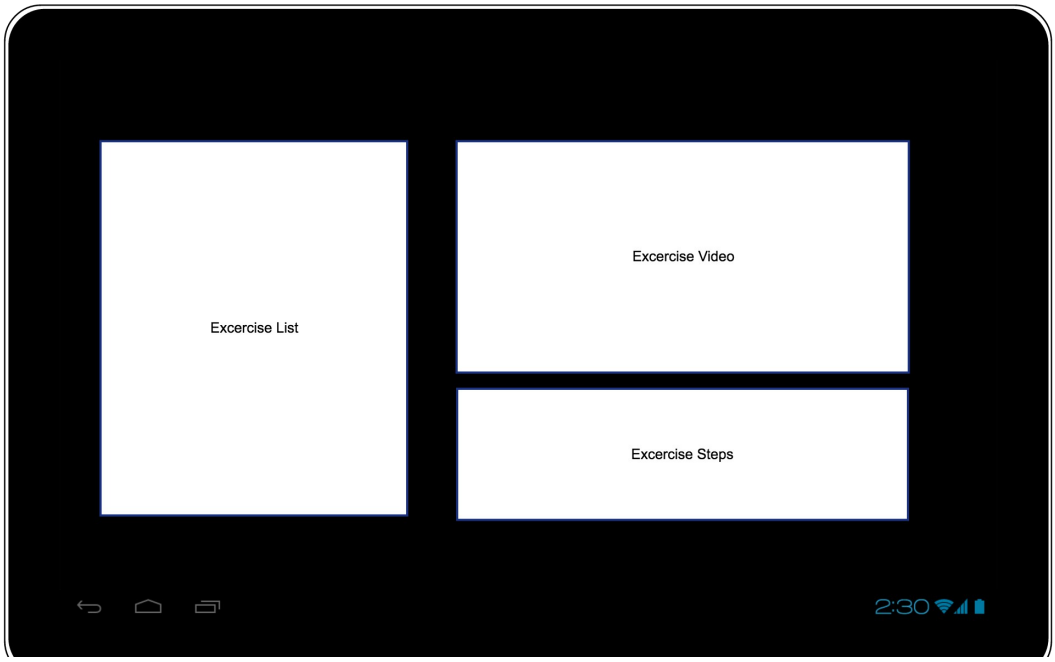
After clicking on the CardView will show the exercise list in swipe view with 2 tabs - All and Favourites

Screen 3



Exercise detail view in portrait mode.

Screen 4



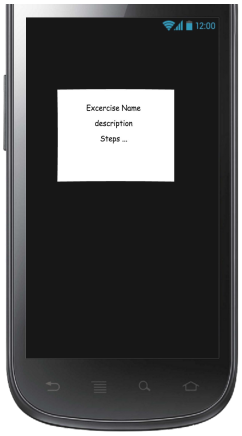
Exercise detail view in landscape mode which shows the exercise list on the master view and the video and steps on the detail view.

Screen5

Create a Configuration Activity to show the Widget that allow user to select a favourite exercise from the favourite exercise list.

Screen6

Display the favourite exercise title, description and steps.



Key Considerations

How will your app handle data persistence?

●Create a database and content provider to store the favourite exercises saved by the user in the database.

●Allow user to delete the favourite exercise by id or allow them to remove all the favourite exercise.

Describe any edge or corner cases in the UX.

●When user rotate the detail view, the app should persist the position of the video.
●When app is in offline mode, the app should show no network placeholder message on the view whenever the content is not able to retrieve for eg. The exercise list from the cloud or the video from youtube.
●when exercise list has no data, it should show a text placeholder instead of an empty list.

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

●Picasso handle the loading and caching of images.
●ExoPlayer to handle the video playing.

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

●Use Google Mobile Ads to show the ads because the app is free.
●Use Google Analytics to analyze the app usage.

Next Steps: Required Tasks

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

Task 1: Project Setup

Write out the steps you will take to setup and/or configure this project. See previous implementation guides for an example.

- Configure libraries Google Play Services (Google Mobile Ads and Google Analytics)
- Configure 3rd libraries such as Picasso and ExoPlayer
- Setup Google Cloud Endpoint project
- Setup Gradle task installRelease to build and deploy the app
- Add dependencies in Gradle file.

Task 2: Implement UI for Each Activity and Fragment

- Build UI layout for MainActivity which requires Card View.
- Build UI layout for Exercise list activity which requires tab view and recycler view.
- Build UI layout for detail view for portrait and landscape which requires fragments for responsive UX design.

- Define app's theme styles and colors that conform to the material design guidelines.

- Add the app bar menus options for the apps

Task 3: Add related classes for DB persistence

- Add subclass extends from ContentProvider class
- Add database contract class to define the table name and columns.
- Add subclass to extends SQLiteOpenHelper to define the db name and create query etc.
- Create IntentService to retrieve exercise data the very first time
- Create Job and JobDispatcher to retrieve exercise data periodically.

Task 4: Add API methods for Google Cloud Endpoint

- Design what JSON format of the exercise data that returns from the server.

- Add API methods to return the exercise data.

Task 5: Add utilities classes

- Add recycler class and recycler adapter class for the recycler view.
- Add adapter class for the card view
- Add a POJO class for the exercise.
- Add a JSON utility class to parse the JSON data from the server.

Task 6: Create Activity and Fragment classes

- Create a main activity class that includes the Card Views, the Google Mobile Ads will display in the main activity.
- Create a exercise list Activity class which includes a fragment class. The fragment class includes a tab view which requires 2 fragment class respectively.
- Create a exercise detail view activity class for Portrait mode and it requires a fragment class that includes the detail layout
- Create a exercise master-detail view activity class for landscape mode and it requires to include 2 fragments.

- Add notification when scheduled data retrieval is successful.

Task 7: Use LoaderManager to initiate loader to load data

- Fragment to load all exercises - Add loader to load data from Google Cloud Endpoints and add data into the local db's table.
- Fragment to load favourite list – Add loader to load data from the local db

Task 8: Add Widget UI and Widget Activity class

- Add Widget UI layout

- Add Widget Activity to configure the widget and show the favourite exercise

Task 9:

- Add UI Tests