Marvel Characters app

GitHub Username: Jamargle

Marveleando

Description

Marveleando will be an app for Marvel movies fans and any other people interested on Marvel characters. The users will be able to search and learn everything related to any superhero or villain in the Marvel universe.

Intended User

The target user is everyone willing to know about any Marvel character.

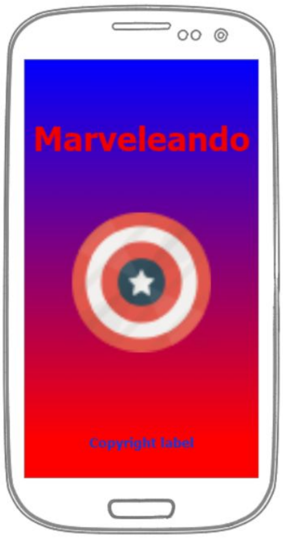
Features

The app will be developed completely in Java and it will have these features:

- Splash screen with Marvel logo and copyright information that will be shown during the downloading of the first characters to be shown.
- A grid screen with the characters pictures and their names. They will appear ordered by name.
- If the user tap a character, a screen with the following details will be shown:
  - Picture
  - Name
  - Description
  - Links to related information
- Settings screen to set things like the quality of the images and the order of the characters (by name or reversed by name)

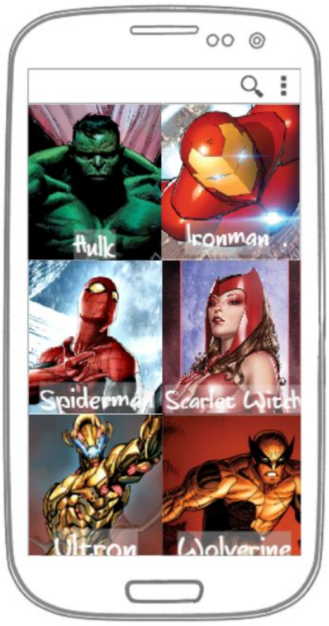
User Interface Mocks

Splash screen



This screen will be shown for a second or something like that during the loading of the next screen.

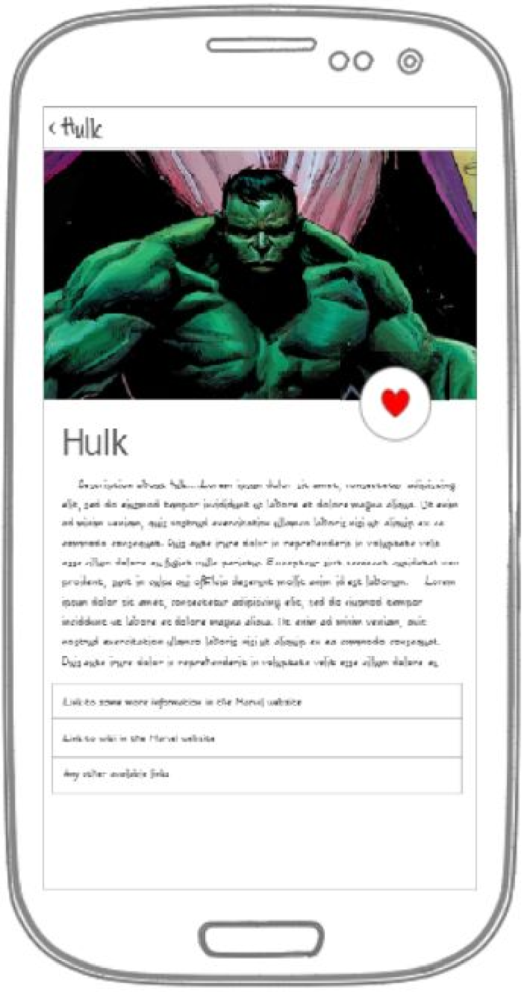
List screen



This screen will show a list of characters to the user.
If the user scrolls down, more characters will be loaded and shown.
If the user clicks over one character, the details screen will be shown.
If the user clicks over the search icon, the user will be able to search characters by name and the results will appear in this screen.

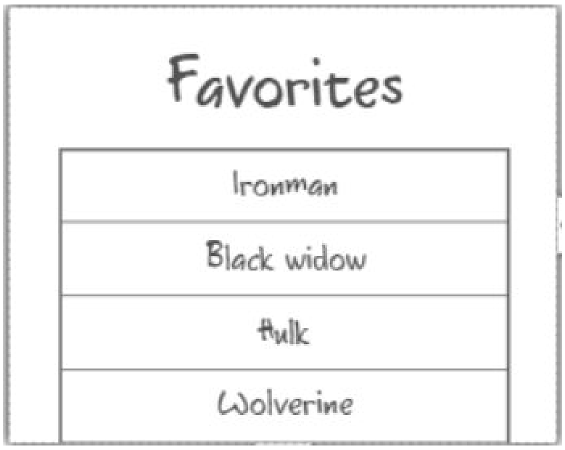
In the 3 dots menu the user will have an option to see only the characters marked as favorites.

Details Screen


Hulk
Hulk
Link to some more information in the Marvel website
Link to wiki in the Marvel website
Any other available links

The details screen will show the whole picture of the character in landscape mode, the name, a description if exists (it is an optional attribute regarding the API documentation) and a list of links to more information in the Marvel website.
I have included a button for the user to mark the character as favorite. So the character will appear in the list of favorites characters in the widget and in the

Widget



The widget will be a simple list that will show the names of the characters marked as favorite in the application.
If the user clicks one of these characters in the widget, the details screen of that character will be shown.

Key Considerations

As per rubric requirements, all the project will be developed using Java language and a stable version of the libraries I will mention later being prefered the latest stable ones.

How will your app handle data persistence?

I will fetch the data from the Marvel API and I will store the first 30 characters in a local Database using Room so they will be downloaded only for the first time instead of perform a download every time the user opens the app.
I also will have stored the characters the user mark as favorites.

Describe any edge or corner cases in the UX.

I will add animated transitions for the pictures when the details screen is shown or when the user goes back.
During the downloading of characters an animated progress bar will be shown.
I will use some image for the while the images last to be loaded and another one in case the image of a character does not exist for the character images in both list screen and detail screen.
When there is no Internet connection, the service or permissions are not available an empty screen with a message explaining the problem will be shown.
For the search of characters, I will add a search icon in the toolbar of the list screen and the search will start with 2 characters or more.
For the list screen I will add a landscape layout for mobile with more columns and besides that for tablets I will implement the master-flow design.

Regarding accessibility, the image views will contain a content description field. The colors will be selected accordingly to follow Material Design guidelines and have good contrast to be as much readable as possible.
I will use as primary color something like this dark blue #000aa6 and as secondary a red like this #a31715.
All the resources such as colors, strings, style, etc. will be in xml files in the res folder of the project.

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

I will establish the API 19 as minimum SDK supported and API 27 as compile/target SDK.
I will use Android Studio 3.1.2 as IDE for development of the project.
Gradle version 3.1.3 for building the project.
The Android Support Library version will be 27.1.1 to use android features not supported in old Android versions.
Picasso 2.5.2 for loading the images.
Butterknife 8.8.1 to bind views and string resources.

Room 1.1.0 for data persistence.
Retrofit 2.3.0 with Gson converter to download and parsing json resources from the Marvel API.
Dagger 2 2.14.1 for dependency injection
Espresso 3.0.2 and mockito 2.16.0 for UI testing and unit testing.
RxJava2 2.1.10 and RxAndroid 2.0.2 for handling background work in separate threads mainly for download resources from Internet.

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

I will use the Marvel API to download the information about the characters that will be shown in the app. I had to register as developer in their web to get the public and private API keys. After read documentation and see the samples, I will be able to fetch the information in my app by using Retrofit. I will use:

- **/v1/public/characters** to fetch the characters shown in the list screen. I will have to implement some pagination logic to allow the user download more when scrolls down.
- **/v1/public/characters?nameStartsWith=´searchText´** to fetch characters related to the text the user is typing in the search box.

About Google Services, I will create two flavors of the app (free and paid) and I will use Admob in the free version.
About the point of using a SyncAdapter/JobDispatcher or IntentService or Asynctask for backend communication, I have thought to use an IntentService to wrap the logic of the download of the data from the Marvel API.

Next Steps: Required Tasks

Task 1: Project Setup

- First I will add libraries and a skeleton of the application following the Model View Presenter architecture pattern and I will configure Dagger 2 library to be able to inject dependencies in the classes.
- Configure RxJava 2 to handle business logic in a background thread.
- Configure the network gateway to fetch resources from the Marvel API in the list screen.
- Configure data persistence to store the data coming from network.

Task 2: Implement UI for Each Activity and Fragment

- Create the list screen to show the characters.
- Implement logic to search the characters by name.
- Create the details screen and the navigation logic to it from the list screen.
- Use unit tests if needed to avoid breaking the existing features.

Task 3: Implement tablet and landscape screens

- Implement the tablet layout master/detail flow.
- Check that the layouts in landscape mode are ok and adapt everything to be working properly.

Task 4: Implement animations and detailed design

In this step I will add resources and implement the logic needed for the improvement of the user experience of the app like animations, enhancements in layouts, etc.

Task 5: Implement the widget

In this step I will add resources and implement the logic needed for the widget to show the list of favorites characters.