

[Description](#)

[Intended User](#)

[Features](#)

[User Interface Mocks](#)

[Splash screen](#)

[Main menu](#)

[Favorites](#)

[All characters](#)

[Seen](#)

[Character details](#)

[Character pictures](#)

[Character comics](#)

[Comic Details](#)

[Image viewer](#)

[Search](#)

[Key Considerations](#)

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

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

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

[Next Steps: Required Tasks](#)

[Task 1: Project Setup](#)

[Task 2: Implement All characters activity](#)

[Task 3: Implement Character Details activity](#)

[Task 4: Implement Photo Viewer activity](#)

[Task 5: Implement Character Photos activity](#)

[Task 6: Implement Character Comics activity](#)

[Task 7: Implement Comic Details activity](#)

[Task 8: Implement Seen activity](#)

[Task 9: Implement logic to add to favorites](#)

[Task 10: Implement logic to remove from favorites](#)

[Task 11: Search](#)

[Task 12: Implement widget for displaying favorite characters \[OPTIONAL\]](#)

[Task 13: Implement option to save photo to device \[OPTIONAL\]](#)

**GitHub Username:** ivamluz

# Marvel Shelf

## Description

Marvel Shelf allows users to search for their favorite Marvel characters and add them to their shelf, so they can quickly find relevant information, such as description, photos and stories about their favorite characters.

## Intended User

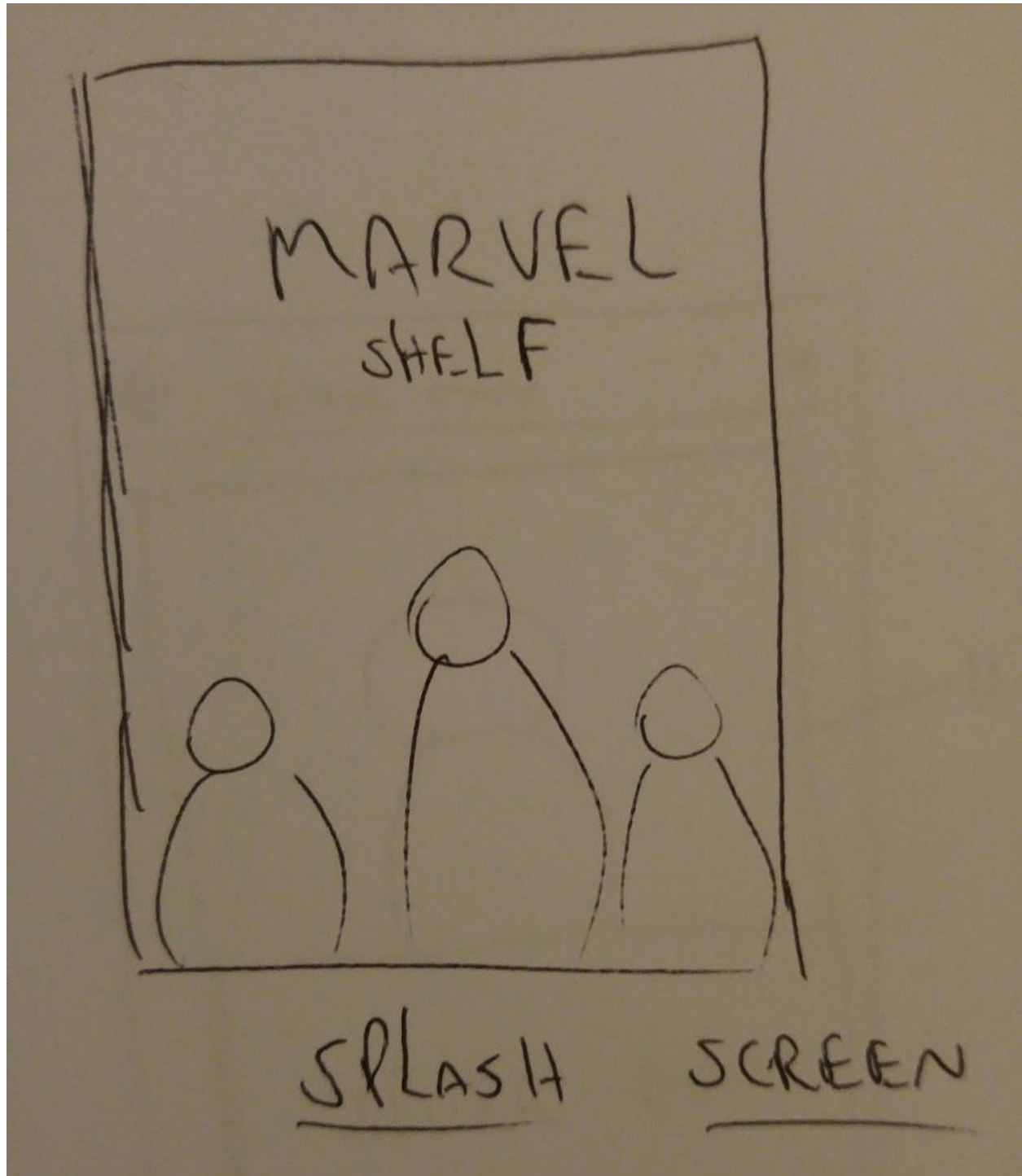
This is app intended to be used by Marvel fans.

## Features

- Character search;
- Character details, such as: name, description and comics.
- Character photos;
- List of favorite characters;
- Ability to add a character to favorites list;
- List of characters already seen;
- Collection widget for your favorite characters [OPTIONAL]
- Ability to save character photos to device [OPTIONAL];

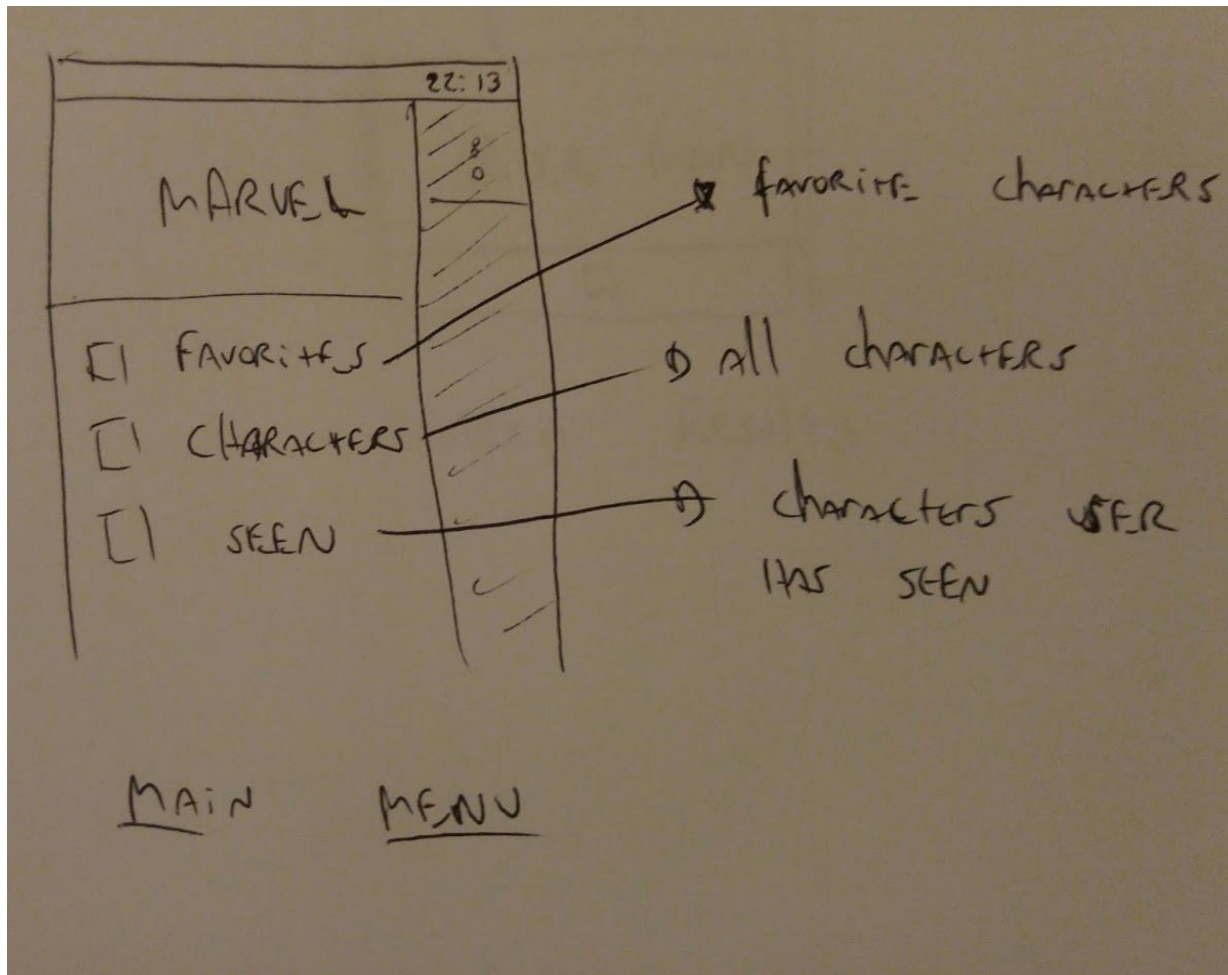
## User Interface Mocks

### Splash screen



First screen displayed when the app is opened. This is just a splash screen that shows the app logo and description, plus a image with some Marvel characters.

## Main menu



This is the main menu for the app. It initially has 3 items planned:

**Favorites:** Displays the characters the user has added to his/her favorites list;

**All characters:** Displays all characters in a recycler view. This screen also displays a button for launching the Search activity.

**Seen:** Display a list of all characters the user has already seen

## Favorites

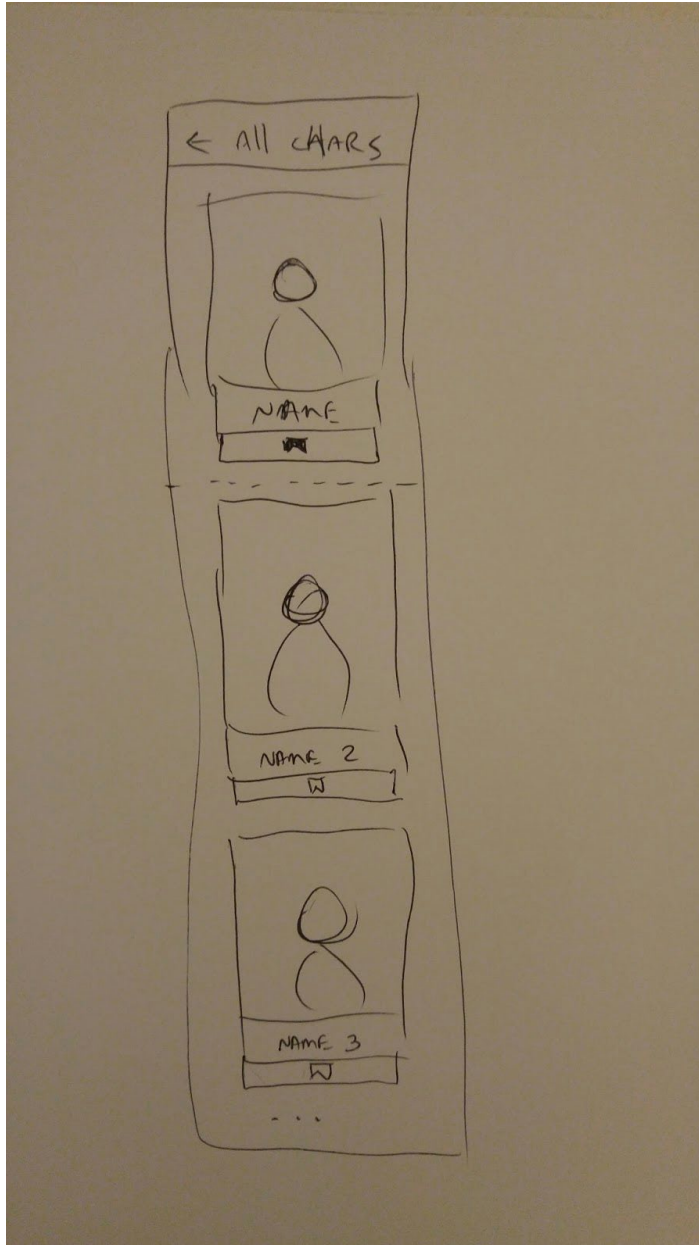


Displays a list of all the characters the user has added to his favorites list. If no character has been added to the favorites list yet, a friendly empty state message is displayed.

When the users clicks in a character card, a new activity is launched, showing the details for that specific character. Also, that character will be added to the Seen list, if not already.

If user clicks the bookmark button (at the bottom of the card), the card is removed from the favorites list.

## All characters



Displays a list with all the characters available in the API. The characters are displayed in a recycler view, for improving performance. Images will be lazy-loaded using Glyde.

This activity has a search icon in the header, so users can launch the search screen and search for specific characters of their interest.

When the users clicks in a character card, a new activity is launched, showing the details for that specific character. Also, that character will be added to the Seen list, if not already.



If users clicks the bookmark button, the card should be added/removed to/from his/her favorites list.

## Seen

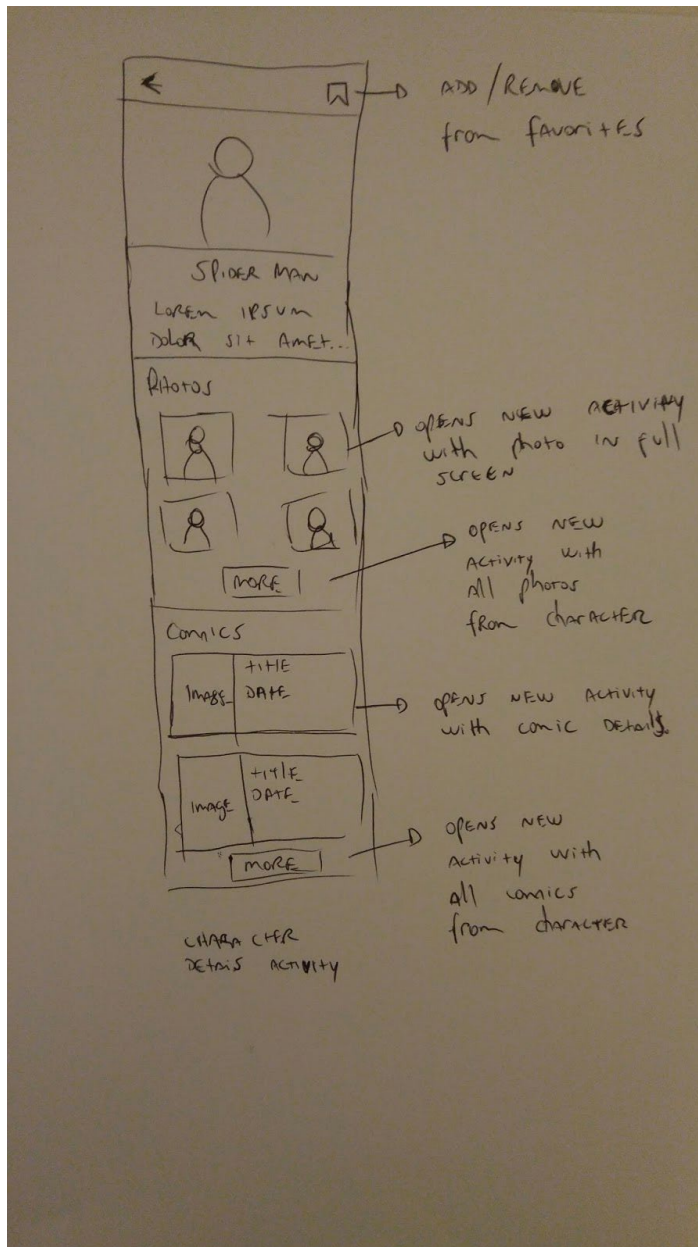


Displays a list (recycler view) with all characters that have already been seen (details opened) by the user.

When the users clicks in a character card, a new activity is launched, showing the details for that specific character. Also, that character will be added to the Seen list, if not already.

If users clicks the bookmark button, the card should be added/removed to/from he favorites list.

## Character details



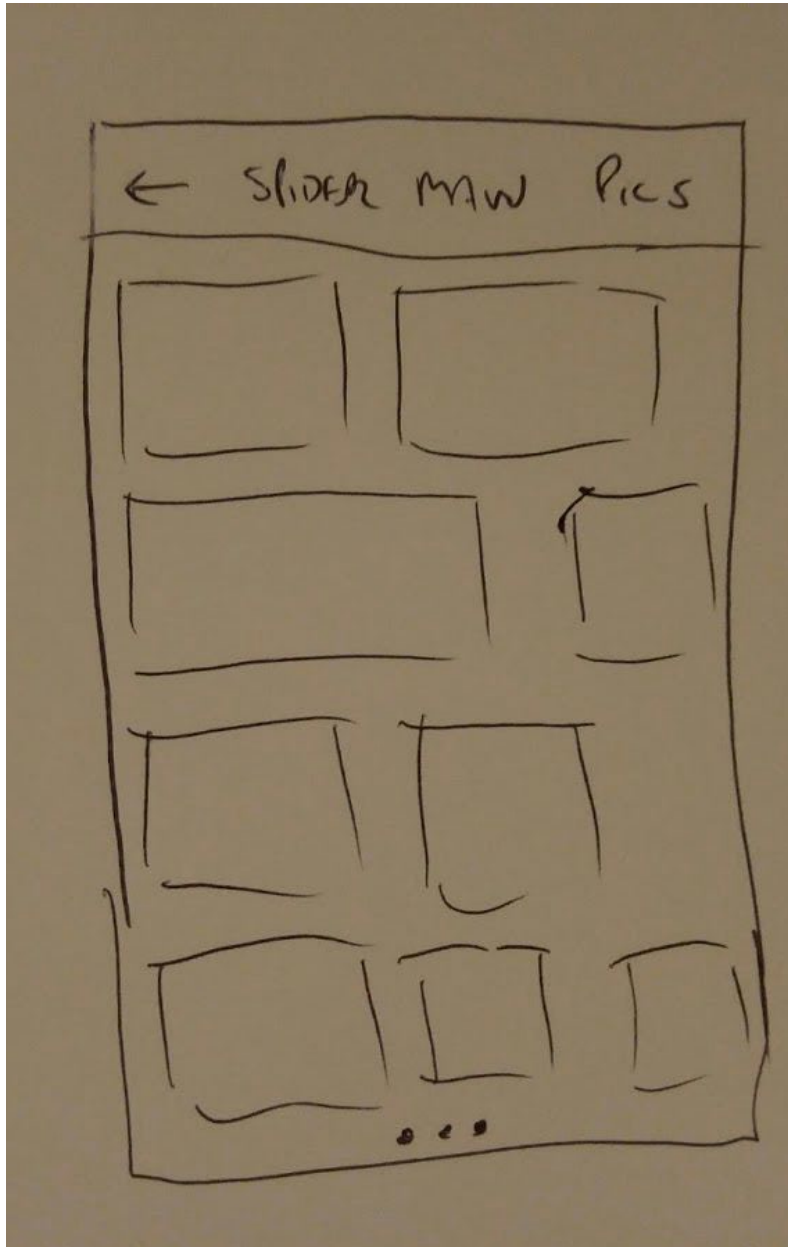
Display the details for the selected character. This activity will display:

- Main photo for the character
- Character name
- Character description
- Character photos: limited to 4 or 6 photos and a MORE button, which launches a new activity that shows all the available photos for the character.
  - If a photo is clicked, the Image viewer activity will be opened, displaying the photo in full screen;



- Character Comics: limited to 4 or 6 items and a MORE button, which launches a new activity showing all the available comics associated to the character.
  - If an item is clicked, a new Activity is launched, containing detailed information for that specific comic.

## Character pictures



Lists all available photos related to a character.

## Character comics



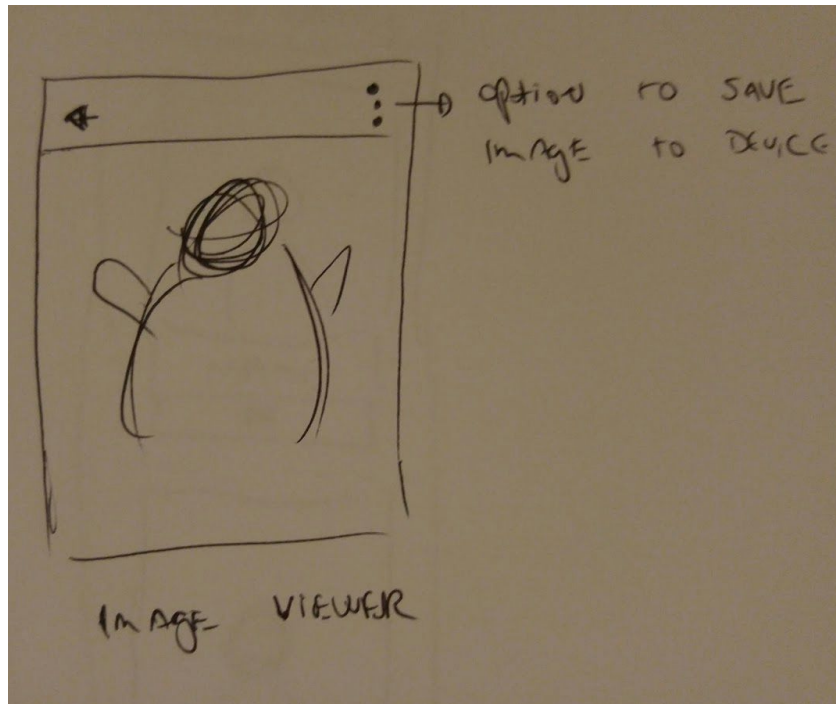
Lists all available comics associated to a character. When a specific item is clicked, a new activity is launched with details for the selected comic.

## Comic Details

Display details for a specific comic of the character. Details include:

- Name of the comic
- Image of the folder
- Date
- Description
- List of characters that appear in the comic.

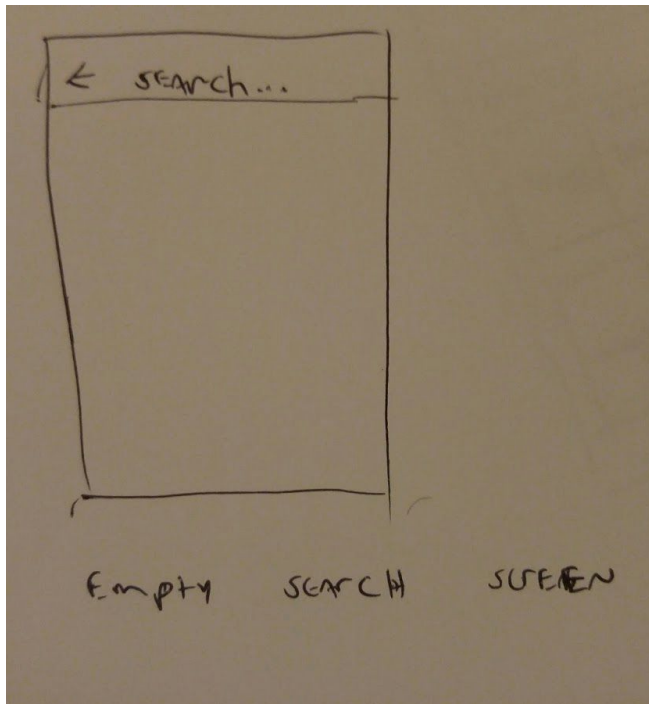
## Image viewer



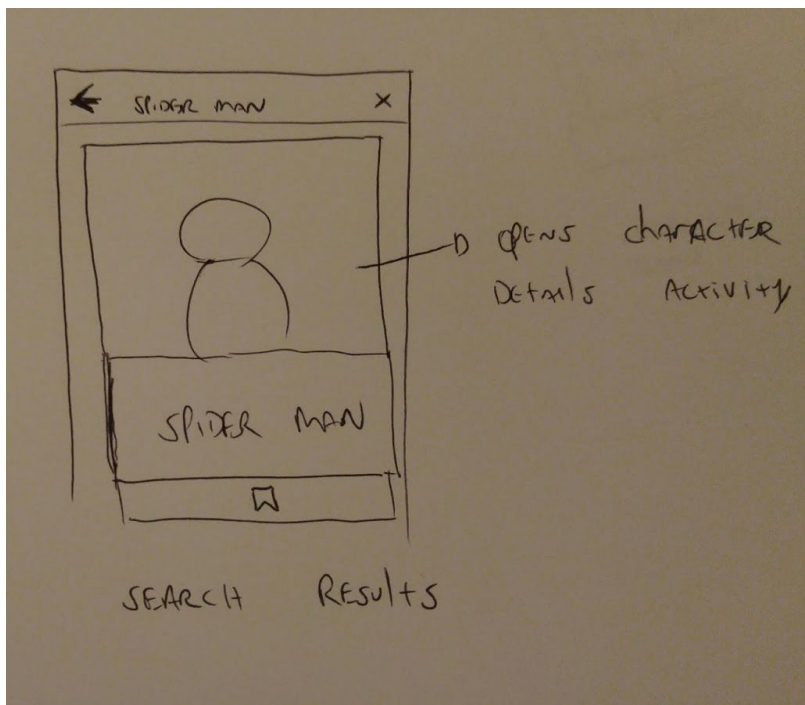
Opens a character image in fullscreen.

This activity also allows for users to save the image to their devices [OPTIONAL]

## Search



*Empty search*



*Filled search*

Allows users to search for specific characters. Initially, this screen displays only a text field for users to enter the search term. After the search term is inserted, a search is performed and:

1. A list of characters found is displayed; or
2. An empty state message is displayed, if there are no results available.

The behavior for the each item in the list of characters is the same as previously described for the the **All Characters** activity.

## Key Considerations

### How will your app handle data persistence?

Characters data will come from Marvel's API. The app will save some data (such as characters info, for caching, and favorite chars) internally using SQLite and a Content Provider. Settings will be saved using SharedPreferences.

### Describe any corner cases in the UX.

- Search button will only be available in the All Characters section. Once hit, a new activity will be open, allowing the user to search for specific characters.
- Users will be able to add characters to favorites list both from listings (All characters, Seen, Search results etc) and from the Details activity for the character.
- If any listing is empty, a empty state screen should be shown;
- If user remove a character from his favorite list, the character should be removed from the list immediately.

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

- Glide, for image loading;
- Butter Knife, for views binding;
- MarvelApiClientAndroid (<https://github.com/Karumi/MarvelApiClientAndroid>) for making it easy to communicate with Marvel API;

## Next Steps: Required Tasks

### Task 1: Project Setup

- Create project and add to repository;
- Set up libraries;
- Structure the project in packages: view, data, model, rest, etc;
- Create structure for displaying main menu across the app;



## **Task 2: Implement All characters activity**

- Integrate with API;
- Create view for displaying individual characters;
- Create models;
- Implement adapter;
- Show items in a recycler view;
- Launch details activity when character is clicked.

## **Task 3: Implement Character Details activity**

- Implement integration with API;
- Create view for displaying character details;
- Display character photos (limited to 4 or 6);
- Display character comics (limited to 4 or 6);

## **Task 4: Implement Photo Viewer activity**

- Implement logic to launch activity when photo is clicked;
- Create views;
- Load photo async using Glyde;

## **Task 5: Implement Character Photos activity**

- Implement MORE photos button logic on Character Details activity;
- Create views
- Integrate with API (if needed);
- Display items in a recycler view;

## **Task 6: Implement Character Comics activity**

- Implement MORE comics button logic on Character Details activity;
- Create views
- Integrate with API (if needed);
- Display items in a recycler view;

## **Task 7: Implement Comic Details activity**

- Implement integration with API;
- Create views;
- Populate UI with data;

## **Task 8: Implement Seen activity**

- Create models
- Create database contract
- Create content provider

- Update Character Details activity to track seen characters
- Create views
- Display items in a recycler view;

### Task 9: Implement logic to add to favorites

- Create models;
- Create database contract;
- Create content provider;
- Update existing activities (**All Characters**, **Seen** and **Character Details**) with logic for the bookmark button;
- Create views
- Display items in a recycler view;

### Task 10: Implement logic to remove from favorites

- Update data layer with service to remove item from favorites
- Update existing activities (**All Characters**, **Seen**, **Favorites** and **Character Details**) with logic for the bookmark button;
- On Favorites, implement logic to remove item from list when unbookmarked.

### Task 11: Search

- Implement logic to launch **Search** activity from **All Characters** activity;
- Create views;
- Implement integration with Marvel API;
- Show results in a recycler view;
- Show empty state, if there are no results;
- Implement logic to add to/remove from bookmarks;

### Task 12: Implement widget for displaying favorite characters [OPTIONAL]

- Create content provider
- Create views
- Create widget;

### Task 13: Implement option to save photo to device [OPTIONAL]

- Add menu to activity;
- Update manifest with required permissions;
- Implement menu item callback for saving image to device;