How to Use this Template

1. Make a copy [File → Make a copy...]
2. Rename this file: "**Capstone_Stage1**"
3. Replace the text in green

Submission Instructions

1. After you've completed all the sections, download this document as a PDF [File → Download as PDF]
2. Create a new GitHub repo for the capstone. Name it "**Capstone Project**"
3. Add this document to your repo. Make sure it's named "**Capstone_Stage1.pdf**"

GitHub Username: jgodort

Graffpaper

Description

Do you love wallpapers?? Are you tired of bored wallpapers?? You have an awesome phone but your wallpaper it's a pity…Then, you need Graffpaper!!! An awesome wallpaper app that allows you to search awesome images and know who is behind that images you love.
Well… Okay, there are many apps to find wallpapers but… If I told you that you can synchronize your devices with that awesome wallpaper you found and only pressing a button, Cool!! Right??
Begins now to gives the style that your phone deserves

Intended User

For those who love wallpapers and high-quality images.

Features

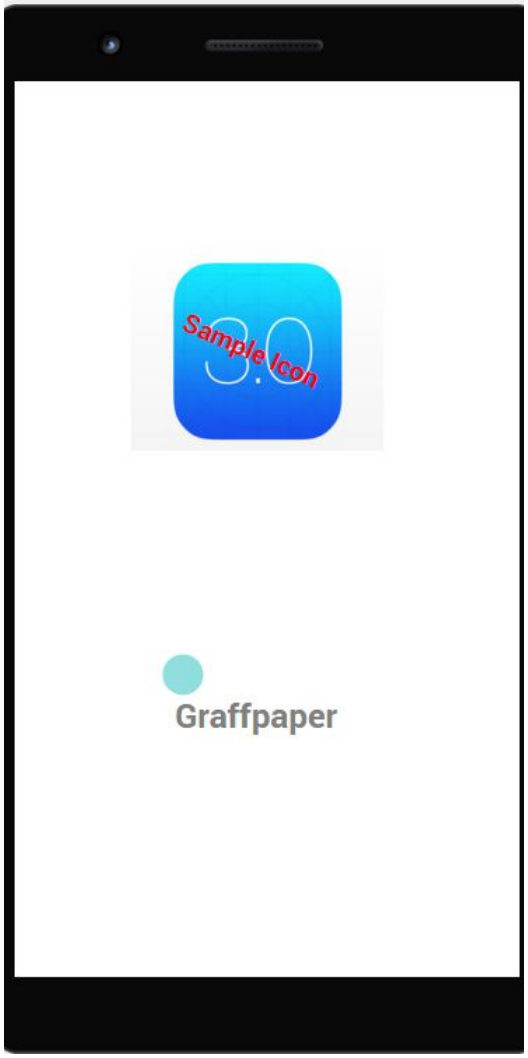
List the main features:

- Search Images.
- Take a look at the photographer behind.
- Download your images.
- Set and sync the wallpaper on all of your devices.

User Interface Mocks

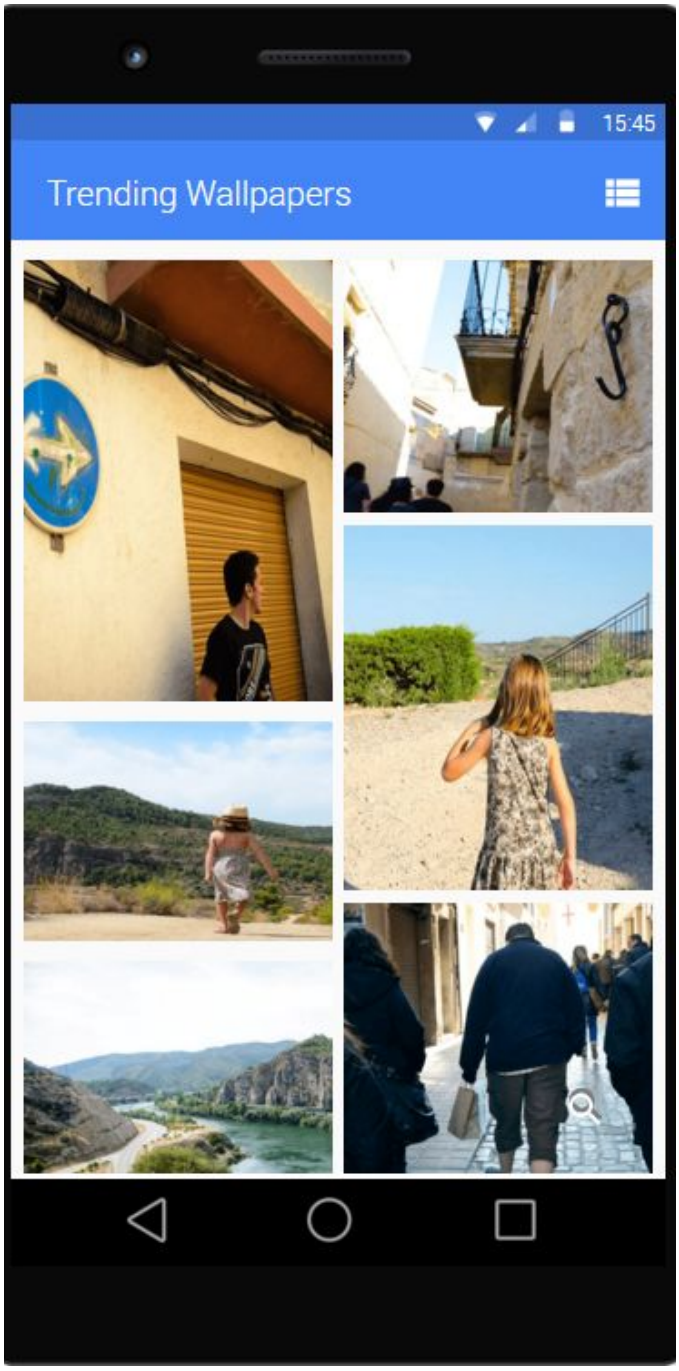
These can be created by hand (take a photo of your drawings and insert them in this flow), or using a program like Photoshop or Balsamiq.

Branded Launch Screen



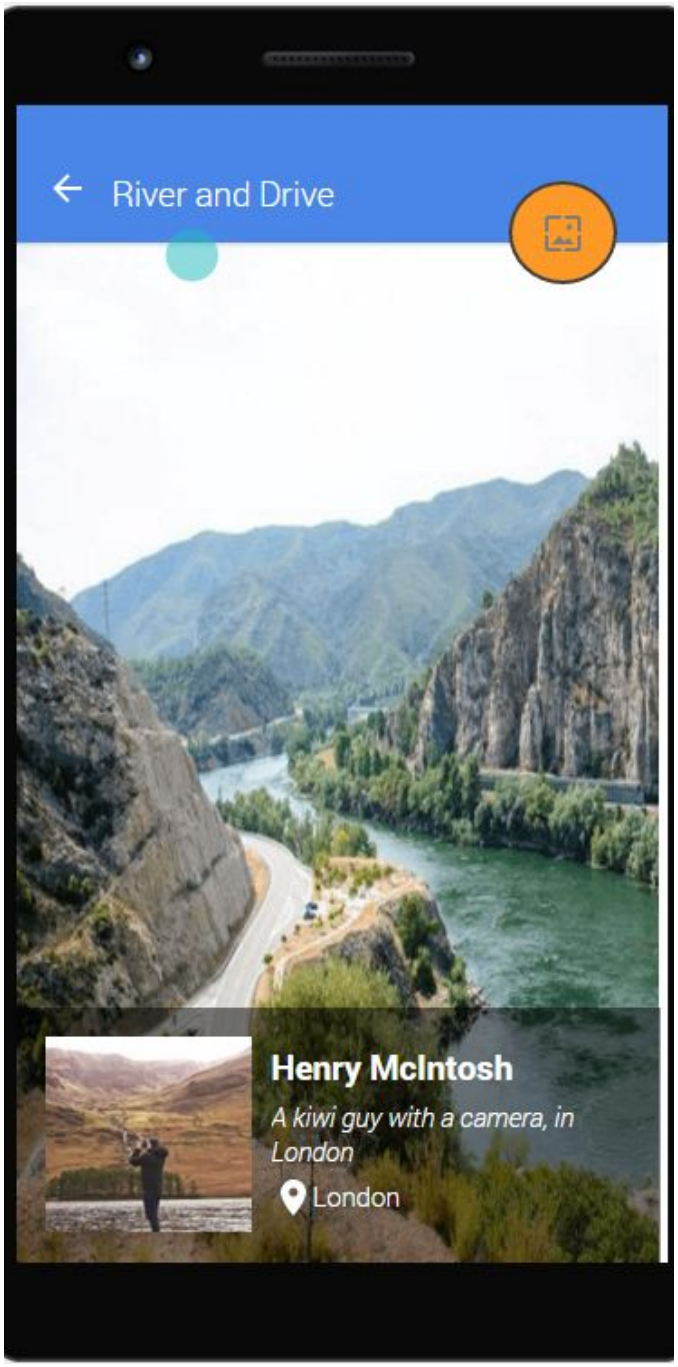
A Branded Screen of the App.

Trending Wallpapers



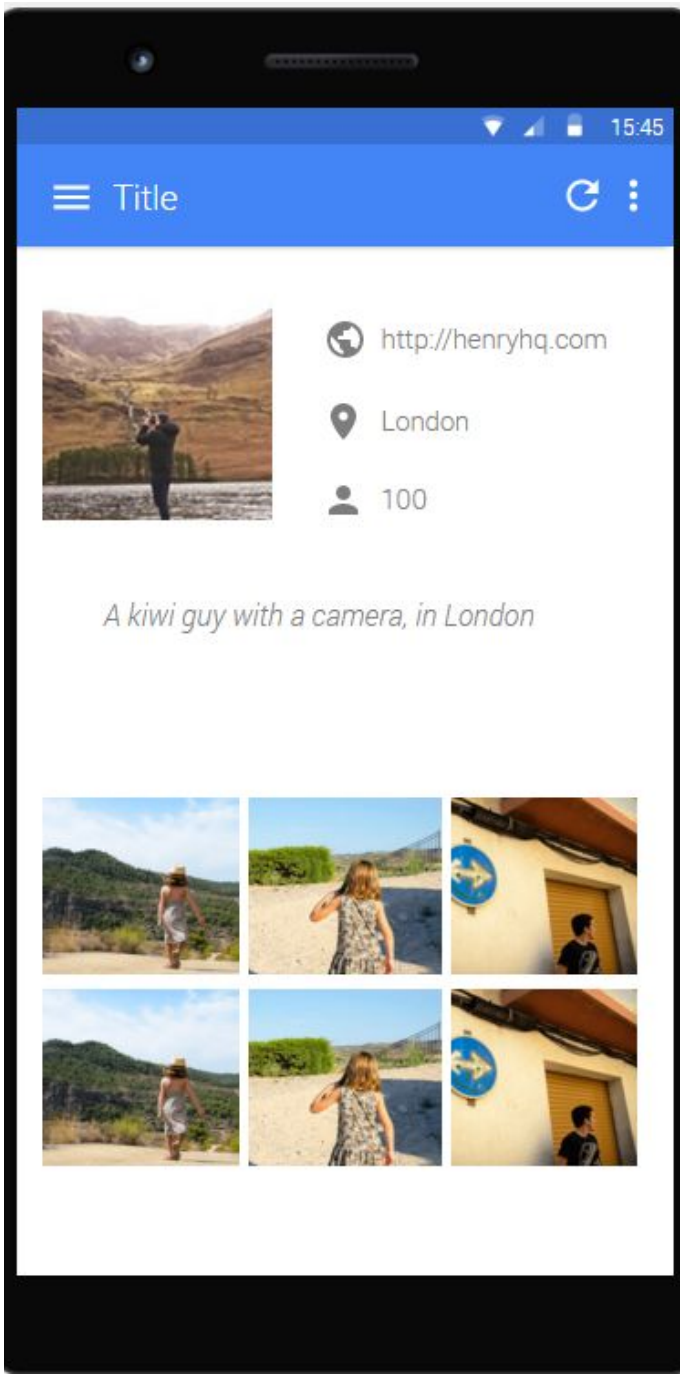
The main Activity shows the trending wallpapers of the moment and allows you to search wallpapers

Image Detail



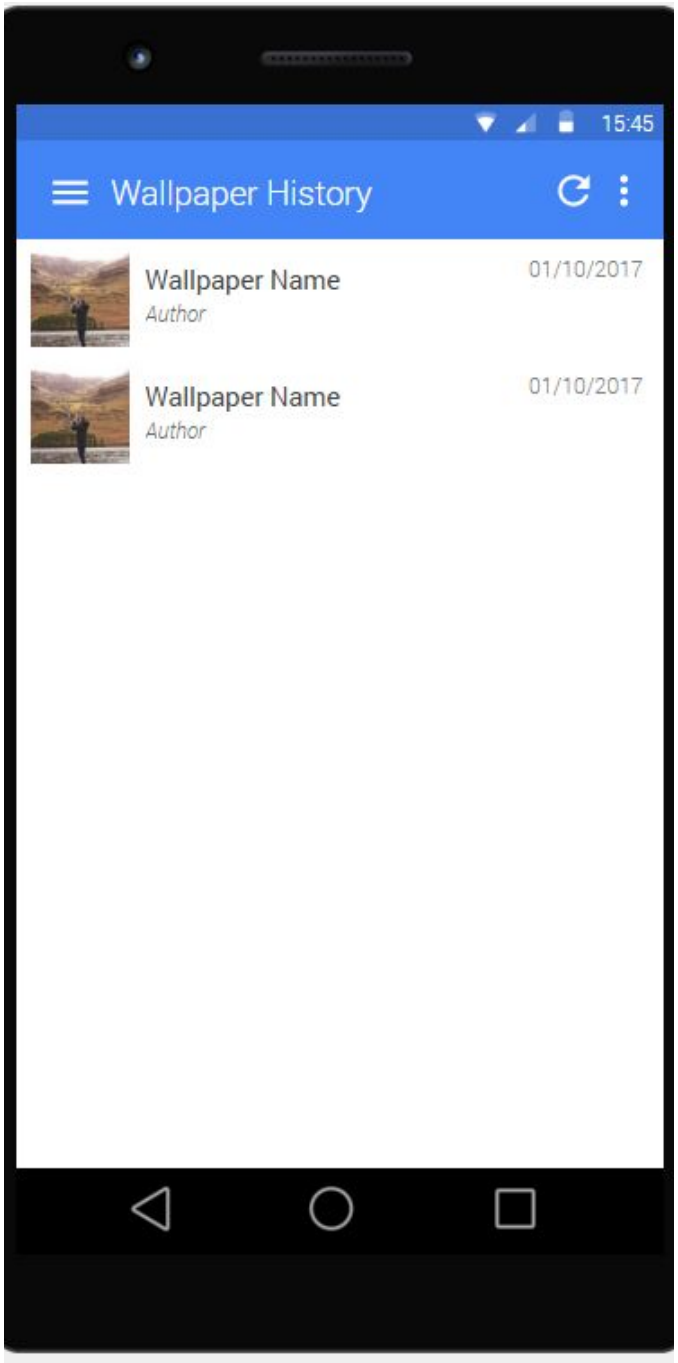
On this Activity you view the wallpaper on full screen mode, set it as your device wallpaper and know more about the author.

Author Profile



A detailed view of the author profile and a curated selection of his photos.

Wallpapers History



A list with the history of wallpapers that you configured on your device.

Widget

A widget that allows you to take a quick view of the history.

Key Considerations

How will your app handle data persistence?

The application will persist the application on the device using GreeDao library.

Describe any corner cases in the UX.

A corner case will be how to retrieve the sync wallpaper notification on the device and set the user selected image as device wallpaper.

Another corner case will be when the user lost the internet connection and tries to search any wallpaper.

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

- Glide: For image download and caching.
- Butterknife: For view binding.
- Retrofit: For HTTP connections.
- Dagger: For dependencies Injection.
- Mockito: For testing mocks.
- Stetho: For Development purpose (database view, http request efficiency, etc...)
- Firebase UI Auth: For login and signup process.
- Firebase Realtime Database: to persist the wallpaper history and the sync information.
- GreenDao: For data persistence.

Describe how you will implement Google Play Services.

Firebase Auth

I'll use Firebase Auth module to manage the user Login and signup process. Also for allow me to manage the device synchronization.

Firebase Realtime Database

I'll use Firebase Realtime Database to manage the historic information about wallpapers and the sync information.

Firebase Crash Report

I'll use Firebase Crash Report to manage the information about unhandled exceptions.

Next Steps: Required Tasks

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

Task 1: Project Setup

- Create a new empty Android Studio Project for Graffpaper
- Create a new CVS repository for the project and configure Readme and .gitignore files.
- Analyze the API Documentation to determine which methods fit with the app requirements.
- Configure Unsplash API: get an api_key from the service.
- Configure the api_key in the gradle files.

Task 2: Implement Model API Client for Unsplash

- Create the model entities to get information from the API service.
- Create the API client Interface.

Task 3: Implement Unsplash Data API Integration.

- Implement the catalog of services to retrieving information from the API.

Task 4: Implement Getting Data from the API.

- The app will implement search by the user through AsyncTask

Task 5: Implement UI for Each Activity and Fragment

- Create UI for the Branded Launch Screen.
- Create UI for the Search Activity.
- Create UI for the Author Detail Activity.
- Create UI for the Historical Wallpapers Activity.

Task 6: Build data persistence support

- Implement all classes needed to handle data persistence, etc.

Task 7: Implement Firebase Integration.

- Implement Firebase Auth integration to manage the user accounts.
- Implements Firebase Realtime Database integration to storage the user wallpaper historic.
- Implements Firebase Crash Report integration to obtain information about unhandled errors.

Task 8: Implement Accessibility.

- Configure support for RTL users.
- Configure multi-language support (English and Spanish in the early stages).
- Configure disabilities aids to the users (Content Descriptions).

Task 9: Implement Graffpaper widget.

- A handsome widget to discover a random wallpaper.

Task 10: Signing and Building.

- Configure app signing process.
- Generate the apk through the “Install Release” task.

Submission Instructions

1. After you’ve completed all the sections, download this document as a PDF [File → Download as PDF]
2. Create a new GitHub repo for the capstone. Name it “**Capstone Project**”
3. Add this document to your repo. Make sure it’s named “**Capstone_Stage1.pdf**”