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: ahmed-basyouni

ArkWallpaper

Description

ArkWallpaper is live wallpaper that would change your wallpaper every interval with a famous image from 500px, the app allow you to choose the category you which to see for example (natural, art,Animals etc) and if you are not sure you can choose to see the most popular wallpapers on 500px, it also give you the ability to choose wallpapers from your storage and change between them with a simple click app will also detect if there is any gif image and will animate them

Intended User

App intended for all kind of users who love to see a new wallpaper every morning

Features

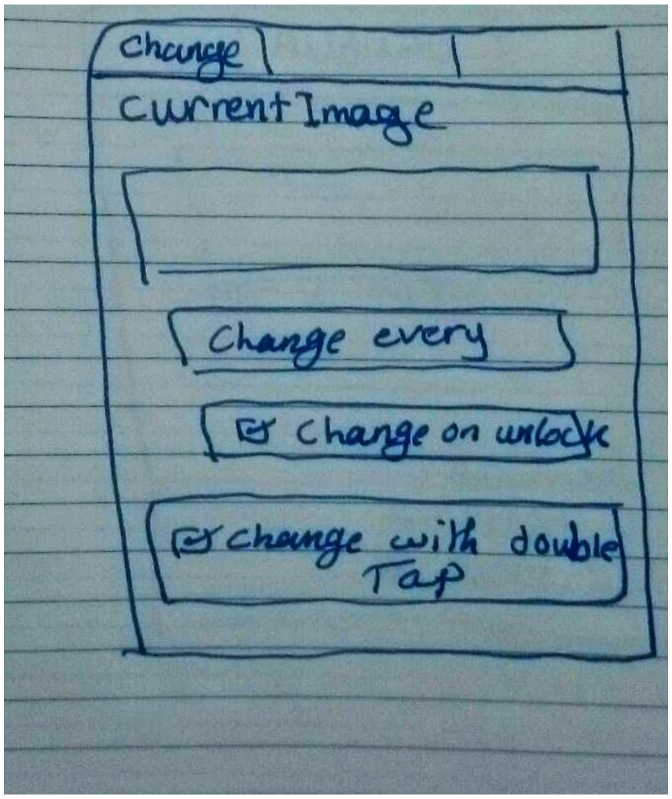
List the main features of your app. For example:

- Easily add images or whole folders
- Change wallpaper with a timer, on each lockscreen unlock, through a customizable widget
- Widgets: change to next wallpaper in rotation list, select the wallpaper you want to see or change the album* with one click
- Ordered rotation or random list
- Blurring the images, grey them or dim the screen
- GIF images integration

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.

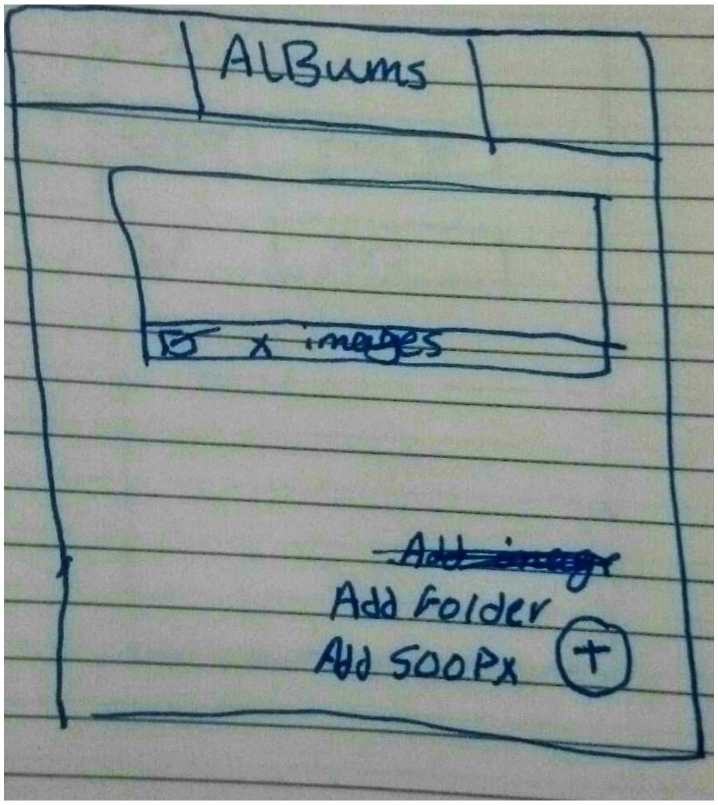
Screen 1



Replace the above image with your own mock [click on the above image, then navigate to Insert → Image…]

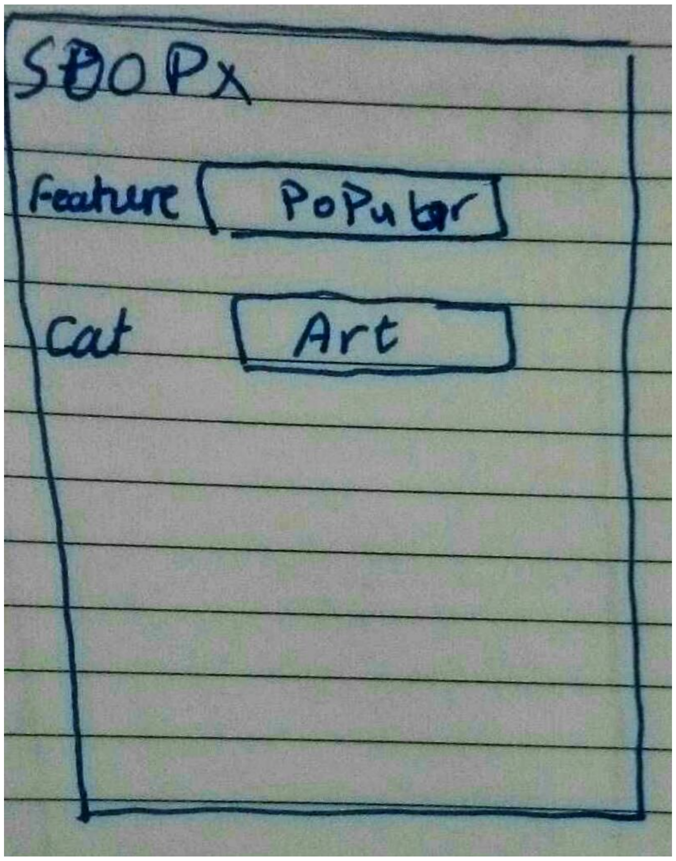
This is the change screen where user can specify which action he would like to use to change the wallpaper either timer, on screen unlock, or with double tap

Screen 2



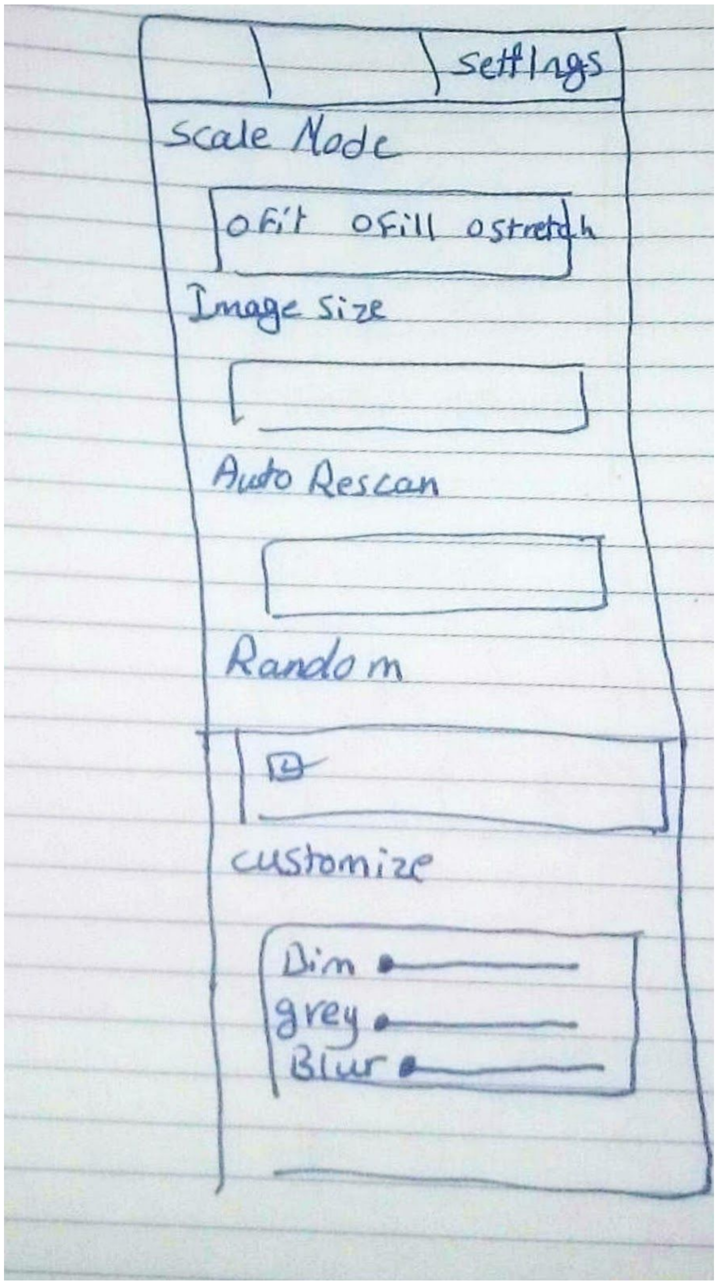
Replace the above image with your own mock [click on the above image, then navigate to Insert → Image…]
This is the album screen where user can see all the albums he specify before and he can also add a new album from a folder of from 500px service

Screen 3



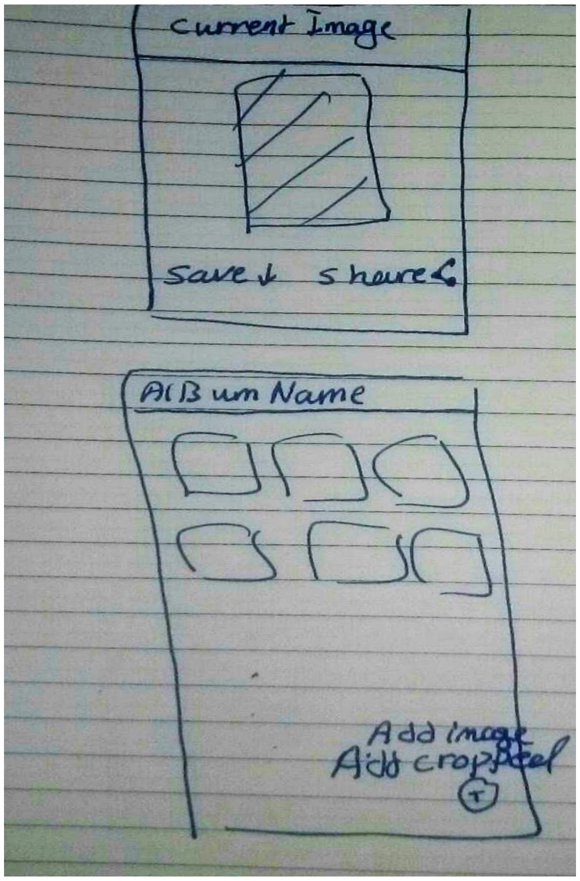
Replace the above image with your own mock [click on the above image, then navigate to Insert → Image…]
This is the 500px configuration screen where user can specify the type of images he would like to see so user can choose for example (popular or highest ranked) , choose the category (Natural or art) or even combine both of them

Screen 4



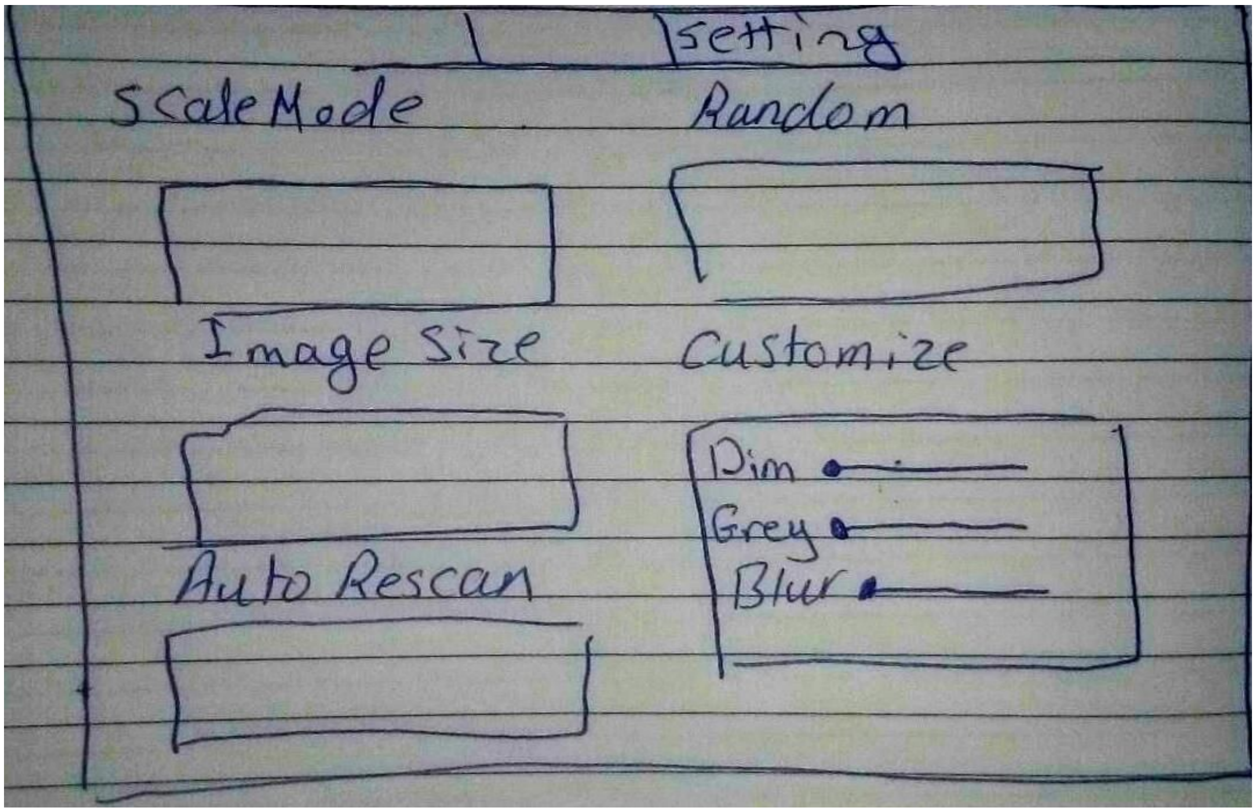
Replace the above image with your own mock [click on the above image, then navigate to Insert → Image…]
This is the setting screen where user can customize his application

Screen 5



When user tap on any image he can share it (if it wasn’t from 500px), in image grid user can add photos or even crop images

Settings screen on tablet



Key Considerations

How will your app handle data persistence?

App uses gallery content provider, also app uses it's own content provider to cache 500px request as 500px request contain 20 images per request

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

- Glide as it can display GIF images.
- Retrofit for network connection to call 500px api.
- android RX to handle observer pattern inside application also it make sense to choose it since we are going to use Retrofit.
- Gson to parse server response

- ButterKnife to bind views
- Facebook rebound for animations

Describe how you will implement Google Play Services.

Well I will implement two google play services:
- Google play mobile ad
- Google Analytics.

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

- Configure libraries

Task 2: Implement UI for Each Activity and Fragment

Build XML for different screens

- Build UI for MainActivity
- Build UI tabs
- Build UI for change fragment
- Build UI for album fragment
- Build UI for settings fragment
- Build UI for 500px configuration

Task 3: Your Next Task

Add Google play services

- Add google play ad to MainActivity

- Add google analytics events to different buttons and views

Task 4: Build the Data Layer

Build the data layer which represent both the caching and Network layers:

- Create Network manager which would connect to 500px webservice and get data
- Create content provider to save 500px response
- Create GallaryManager which would communicate with gallery content provider

Task 5: Build the Logic Layer

Build the logic behind every screen

- Create changeFragment logic
- Create Album logic which would use both content providers
- Create Settings fragment logic and save user preferences

Task 6: Build the Wallpaper service layer

In this task we will build the core of our app the wallpaper service layer in which we will:

- Use user preferences to customize the service (Image scrolling, change time, double tap or wallpaper for gif)

Task 7: Check the tablet UI

In this task we will check the app on tablets and we will fix any bug appear on tablet

Task 8: Add activity transition animations:

In this task we will add some animation between some activities

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**"