WallyPaper

GitHub Username: Antariksh

Description

WallyPaper is a wallpaper app that lets you refresh your homescreen with the best and most favorited work of art introduced by the big and creative community of COLOURlovers.com where people from around the world create and share colors, palettes and patterns.

Intended User

Anyone who loves patterns, mosaic or fabric like materials.

Features

List the main features of your app. For example:
- List most favorite patterns by COLOURlovers.com community.
- Make a wallpaper from any pattern that applies to any screen size.
- Save wallpaper on any physical storage on the device.
- Apply a wallpaper from any chosen pattern.

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

This is an initial design of the home screen, the user will see grid of number of designs from different users that is most favorated by the COLOURlovers.com community.
The user can tap any image to see a details page of the selected design.

Screen 2
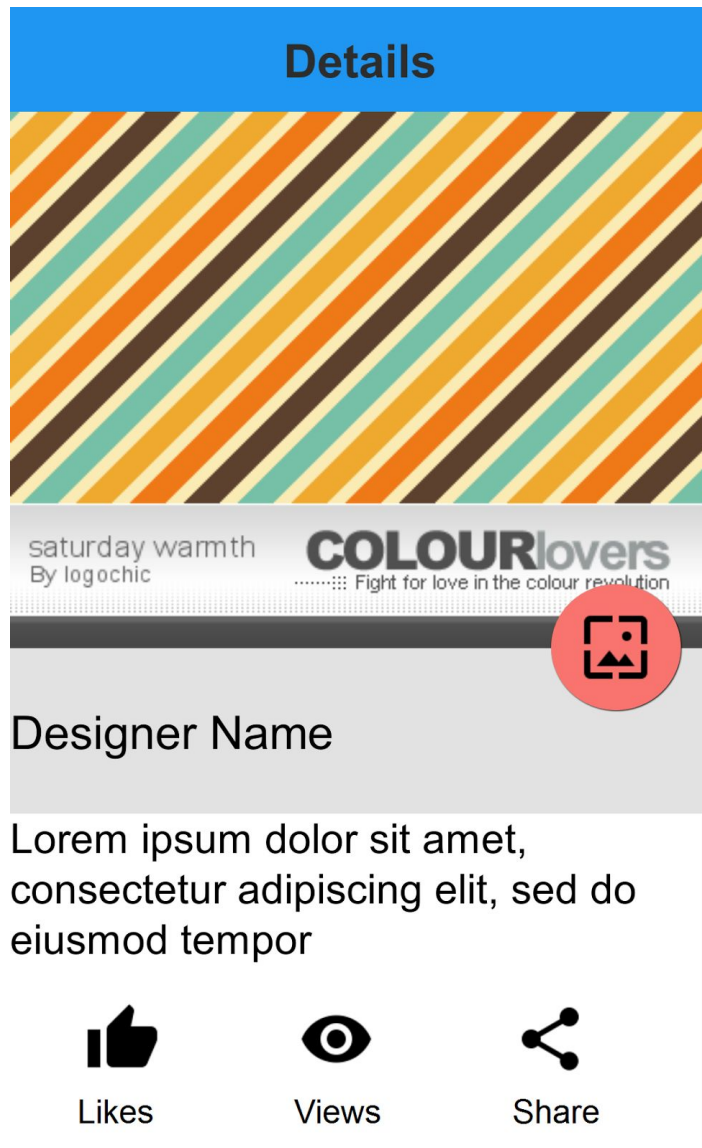
This is the details screen of the selected design, the user can see the number of likes and views of each design, along with the ability to share.
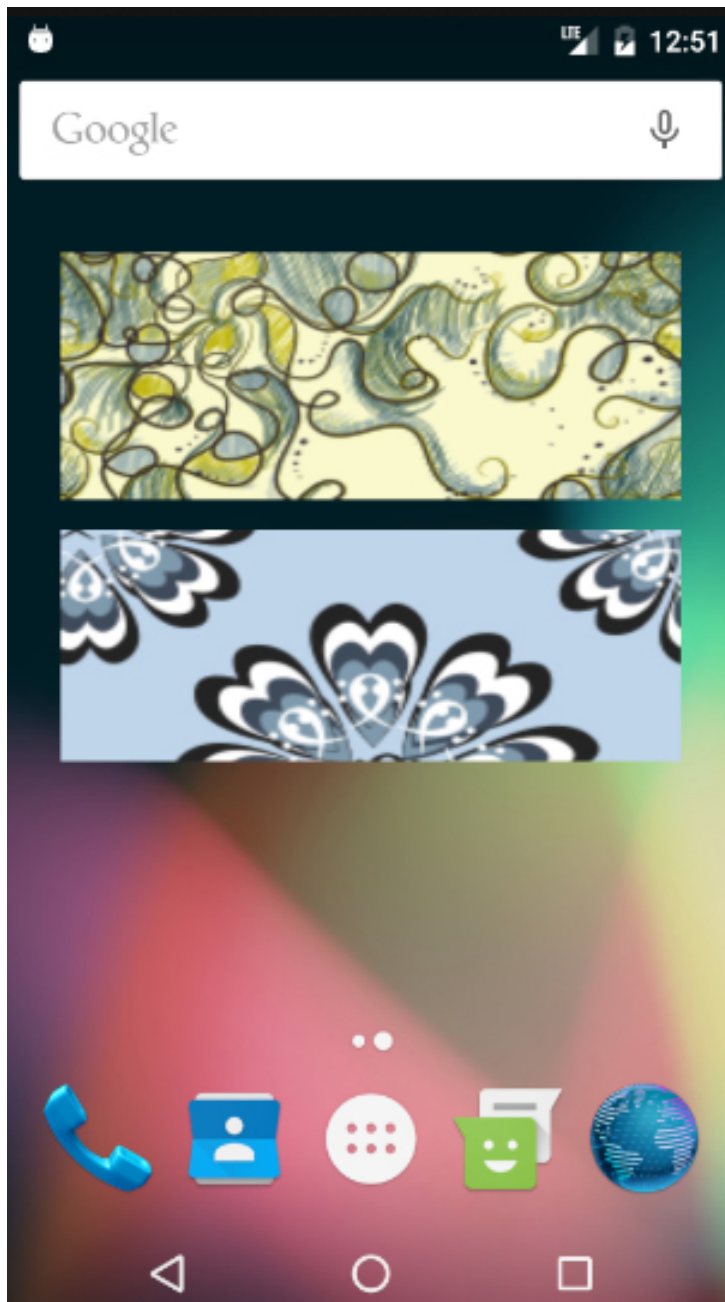User can make use of the biggest feature of the app here, which is setting a wallpaper of the selected design.
In Marshmallow devices, the user will see a permission at first use requiring permission for setting a wallpaper.

Widget Mock 1
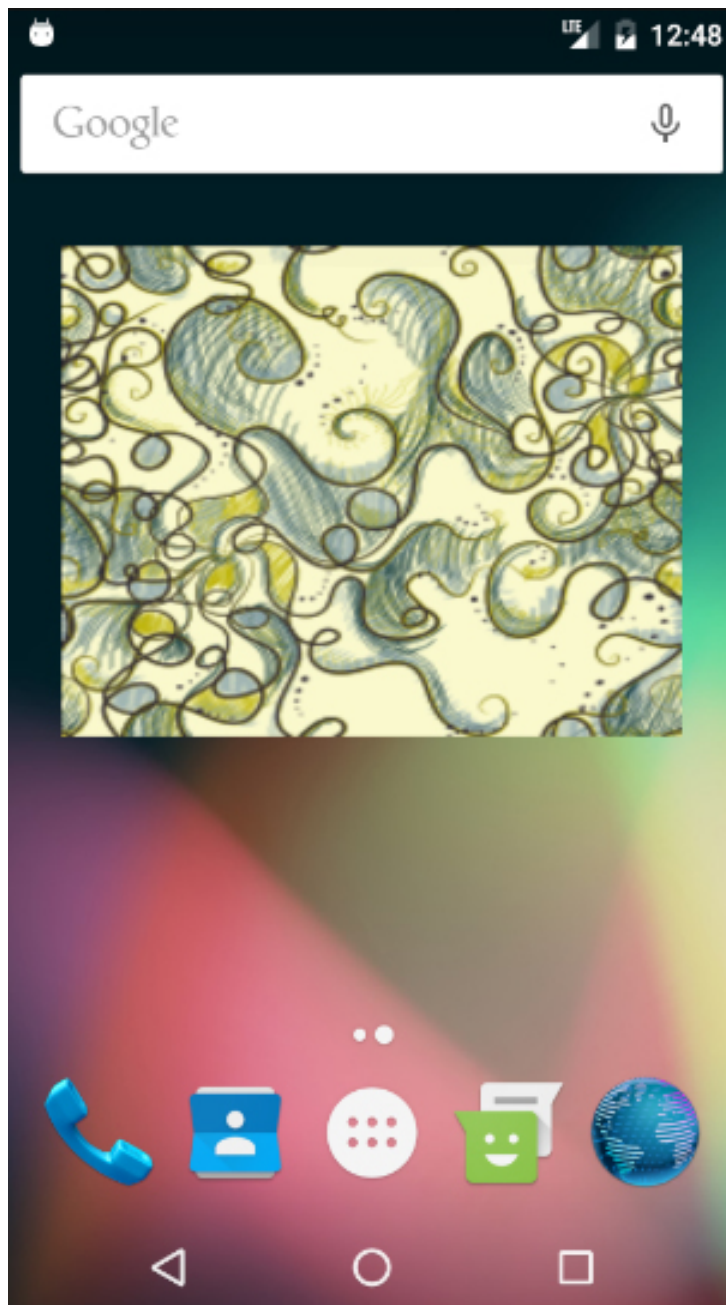
Widget Mock 2

Key Considerations

How will your app handle data persistence?

- Content provider will be created to handle data fetched from COLOURlovers.com api.
- App updates data in its cache at regular intervals using a SyncAdapter.

Describe any corner cases in the UX.

User can save to storage or apply a wallpapaer from icons inside a details screen that shows a bigger version of the pattern.

In Marshmallow devices, the user will see a permission at first use requiring permission for setting a wallpaper.

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

Glide: to handle the loading and caching of images.
Butterknife: To find annotated views and automatically cast corresponding view in your layout.
Schematic : To Automatically generate a ContentProvider backed by an SQLite database.

Next Steps: Required Tasks

Task 1: Project Setup

- Configure libraries
- Build a database scheme
- Design architecture of the app.
- Research apps with the same goal and gain best practices knowledge.

Task 2: Implement UI for Each Activity and Fragment

- Build UI for MainActivity
- Build UI for Details Activity
- Build transition animations for different platform versions.
- Handling UI variations in larger screens.

Task 3: Implement Networking and Data layer

- Build a sync adapter that will fetch data from COLOURlovers.com.
- Build a sqlite database to cache data fetched from the syncadapter.
- Build content provider to handle data fetched from database and update UI.

Task 4: Bug fixing and error handling

- Handle error cases that might happen.
- Bug fixing.