

Trending on Imgur

Table of Contents

[Description](#)

[Intended User](#)

[Features](#)

[User Interface Mocks](#)

[Gallery Activity](#)

[Details Activity](#)

[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.](#)

[Required Tasks](#)

[Task 1: Project Setup](#)

[Task 2: Implement Gallery Activity](#)

[Task 3: Implement UI for Detail Activity](#)

[Task 4: Implement Persistence](#)

[Task 5: Implement Google Play Services](#)

[Task 6: Add Material Design Elements](#)

[Task 7: Finishing Touches](#)

Trending on Imgur

Description

Trending on Imgur let's you will discover the most awesome images currently on the internet.

Intended User

This app is for people who want to be entertained and inspired by funny, heartwarming and helpful images and stories from all around the world.

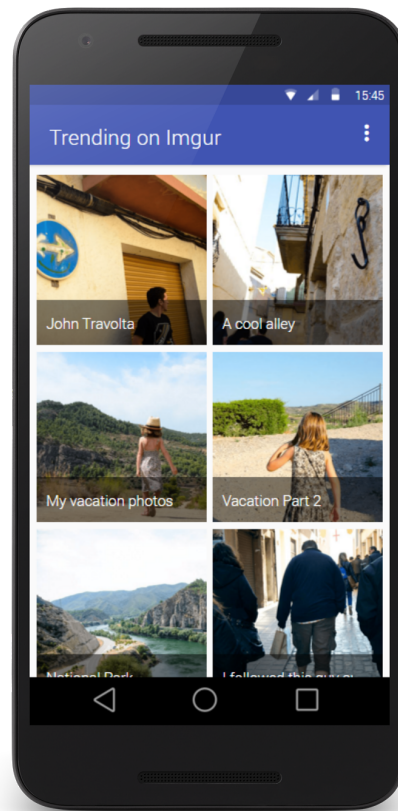
Features

- Shows a preview of currently trending images, albums and stories in a gallery
- Images, albums and stories can then be viewed in detail, along with its top comments and conversations
- Users can share posts that they like over social media

User Interface Mocks

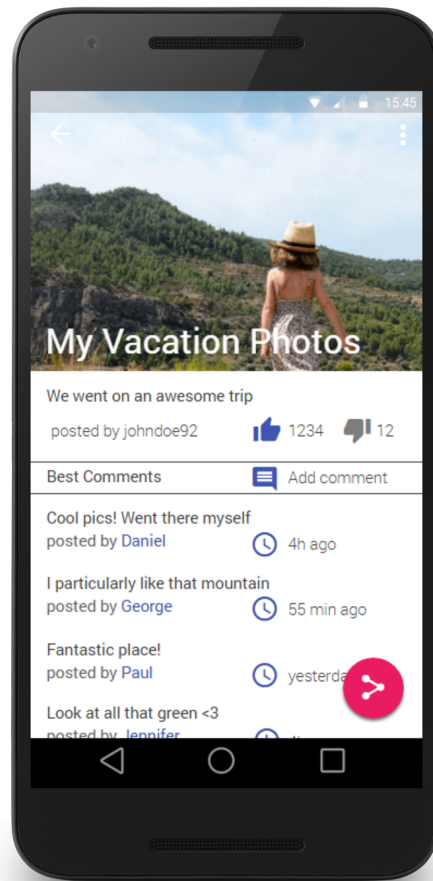
The interface mocks were created using Justinmind Prototyper:

Gallery Activity



The Gallery Screen will show a preview of the currently hot posts on Imgur in a grid. Each grid element will show the thumbnail and the title of the post. Tapping a grid element will take the user to the Post Details Screen.

Details Activity



The Post Details Screen will show more details of the previously tapped post. These details, besides the title and the image (images in case the post is an album) will show:

- the creator of the post
- the post's number of upvotes and downvotes
- the top comments that were added by users to the post

Users can share the post using the floating action button from the bottom right corner.

Key Considerations

How will your app handle data persistence?

A SyncAdapter will be used to query the Imgur API periodically and store information about posts in the database. A CursorLoader will be used to query the database, extract the needed data which then will be displayed on the screens.

Describe any corner cases in the UX.

The information displayed in the Gallery Screen is composed of two types of post objects: Image Type and Gallery Type. Gallery Type posts have more than 1 image, and the thumbnail of the post is almost never the first image from the gallery, so the shared element transition animation will be tricky to implement.

Users can not only comment on the images, but they can also comment on any comment. This tree type comment hierarchy might also prove to be a challenging when implementing the Post Detail Screen.

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

- Butter Knife: to reduce boilerplate code and to bind views and methods easily with annotations
- Glide: a library to load gifs and images with any extension, but also for caching these
- Retrofit 2: a type-safe HTTP client library, for easy json to java object conversion
- Android Support Libraries: to make the app have the same look and feel across supported versions

Required Tasks

Task 1: Project Setup

Create the project files and do all necessary setup, so work can begin:

- Establish minimum and target SDK
- Create the project in Android Studio
- Plan which libraries will be used
- Gather these libraries' Gradle distribution URIs
- Add dependencies to the build.gradle file
- Configure these libraries
- Create project on imgur.com to obtain API key

Task 2: Implement Gallery Activity

- Build UI for Gallery Activity
- Implement user interaction logic for Gallery Activity
- Implement Imgur API to retrieve posts that will be displayed in a grid

Task 3: Implement UI for Detail Activity

- Build UI for Detail Activity
- Implement user interaction logic for Detail Activity
- Use the Imgur API to retrieve details and comments of the selected post
- Implement a ShareActionProvider to share the contents of the selected post

Task 4: Implement Persistence

Since post data doesn't change too often on Imgur, implementing a persistent storage would be a good idea:

- Create a SyncAdapter to periodically update data in our database from the Imgur API
- Create a ContentProvider so access to data is a lot faster, can be done offline and maybe a widget can make use of it in the future

Task 5: Implement Google Play Services

Integrate a few Google Play Services:

- Add dependencies in the build.gradle file
- Use Admob to display dummy ads on both activities
- Create a single Analytics instance and track main interaction events

Task 6: Add Material Design Elements

Theme the application using elements and colors from Material Design:

- Use AppCompat theme (if it's not used already)
- Play around with colors to see what fits the application best
- Implement shared element transitions between the Gallery Activity and the Detail Activity
- Use parallax scrolling in the Detail Activity

Task 7: Finishing Touches

Apply finishing touches where and if needed:

- Make sure all strings are in the strings.xml file
- Make sure RTL is implemented on all layouts
- Create a widget that can be placed on the home screen that links to the current top post
- Create a keystore, and provide the app with a signing configuration
- Make sure app builds with "installRelease" Gradle task