GitHub Username: **dcronin202**

App Title: **In a Giffy**

Description

In a Giffy **is an app that allows users to search a database of GIF images, and either send them via a messaging or Social Media app (such as Slack, Facebook/FB Messenger, etc), or save them to a local database as a favorite.**

Intended User

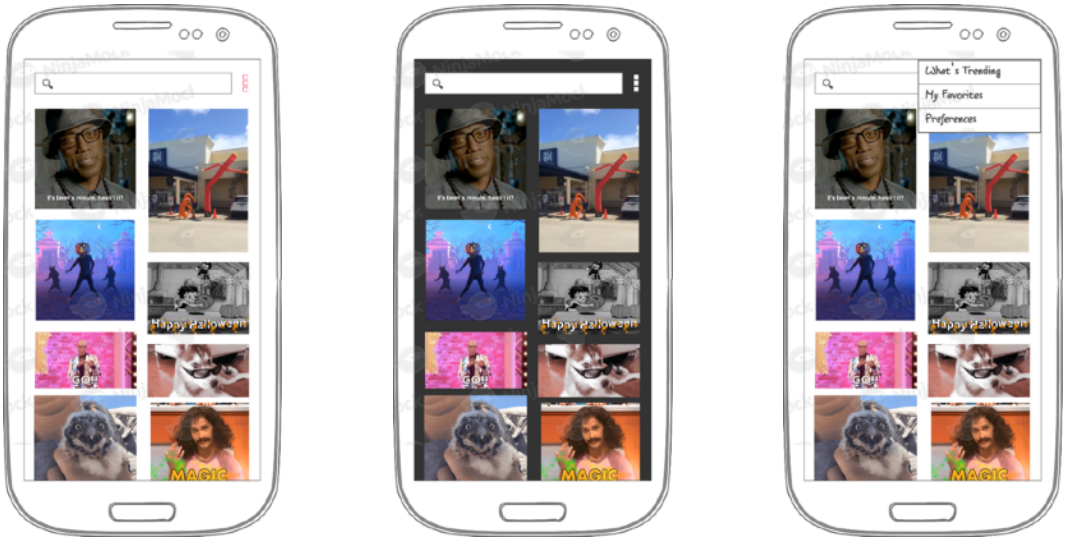
Anyone who enjoys using GIFs in their messages, but would like a single app to search and save their favorite GIFs for future use.

Features

- **See what GIFs are currently popular/trending**
- **Search and send from the world's largest database of GIFs (GIPHY API)**
- **Save favorite GIFs to a local database for future use**
- **Set user preferences, such as Light or Dark mode**

User Interface Mocks

Main Activity - Light & Dark Mode and Menu Options



The main activity will be a simple screen with a search bar at the top, and two columns of GIF images shown in a RecyclerView/GridView. The app will also have a user preference for a Light and Dark mode, as well as a dropdown menu to select this preference, as well as sort by Favorites and Trending GIFs.

GIF DialogFragment/Popup

When a user clicks on a GIF, the image will enlarge to fit the screen, and display two icons on the bottom corners for the user to either save as a favorite, or send the GIF via an app of their choosing (using an intent resolver).

Widget

The Widget will be a simple menu screen when the user long-presses the app, allowing them to create shortcuts to their Favorites or the Search/Main Activity screen. Although the Search feature is probably a bit redundant in this early state, this format will be useful if new features are added in the future, such as Categories or Upload.

Key Considerations

How will your app handle data persistence?

The app will use the Room persistence library to handle the user's ability to save their favorite GIFs.

Describe any edge or corner cases in the UX.

The UX should be fairly straightforward and intuitive. Clicking on a GIF thumbnail will enlarge the image, and to go back, the user can click the software back button or just click anywhere off the enlarged image to return to the main activity. The menu dropdown will also give users the ability to alternate between their favorites and what's currently popular/trending.

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

- ***Data Binding Library:*** **Will be used to declaratively bind observable data to the UI elements, making it simpler and easier to maintain.**
- ***Retrofit:*** **Will be used for handling network requests and interacting with the GIPHY API.**
- ***Glide:*** **Will be used to handle fetching and displaying the animated GIF images.**

Describe how you will implement Google Play Services or other external services.

This app will implement Google Analytics and Google Mobile Ads. The test ad will be displayed as a banner at the bottom of the main screen. Additionally, GIF images will be fetched using the GIPHY API.

Next Steps: Required Tasks

Task 1: Project Setup

- **Create the Main Activity and Main Activity Fragment.**
- **Add necessary build.gradle dependencies (such as Retrofit, RecyclerView, Glide, etc).**
- **Set up initial layout xml file with a TextView to display the url text for "embed_url" from the giphy.com JSON.**

Task 2: Set Up Data Retrieval

- **Create a "data" package that includes the following:**
  - **A GiphyApi interface for Retrofit to make the HTTP GET requests**
  - **A "Gif" object that will store the url to the GIF and whether or not it is a user's favorite**
- **Create a "viewmodel" package that includes the following:**
  - **A Repository class to hold the logic to fetch the data from the GIPHY API and synchronize it with the favorites data persisted on disk**

    - **A ViewModel class to manage the data for the Activities and Fragments, that uses LiveData to dynamically update the UI throughout the activity lifecycle**

- **Set breakpoint and run debug to make sure data is being retrieved from the API**

Task 3: Display Retrieved GIFs

- **Set up RecyclerViewAdapter for GIFs, add RecyclerView to the main activity xml layout, and create a list_item xml for the individual GIFs views.**
    - **Set up Data Binding to dynamically update the RecyclerView from the ViewModel (this will apply to all UI components)**
    - **Run a simple String of the url as the TextView to confirm RecyclerView is correctly displaying data**

- **Set up Glide to display url as GIF images, and change the TextViews into an ImageView or CardView**

- **Set up Search feature**

- **Create DialogFragment to display GIFs when clicked, and add an intent to share the image**

Task 4: Add Database for Favorites

- **Create a "database" package with a DAO interface, and a database class that extends RoomDatabase**
    - **Connect database to the Repository**
    - **Add heart icon to dialog view and create a method to add image to the database when clicked**
    - **Add a dropdown menu to the Main Activity so the user can switch between "Trending" and saved "Favorites"**

Task 5: Final Touches

- **Add Google Analytics and Google Mobile Ads**
    - **Add a banner AdView to the main activity's xml and run the test ad unit ID**
    - **Add a screen tracking event for Google Analytics**

- **Create the Widget**

- **Finish UI polish (to include responsive views for landscape and tablets), and add the Preferences for Light and Dark mode**