Capstone_Stage1

GitHub Username: ericafenyo

Quick Headlines

Description

Get breaking news headlines, current weather and search for articles from over 5,000 fully licensed and trusted sources.
Easily customizable and personalized to give you what you are interested in.
No registration required; choose your country and start right away.

Intended User

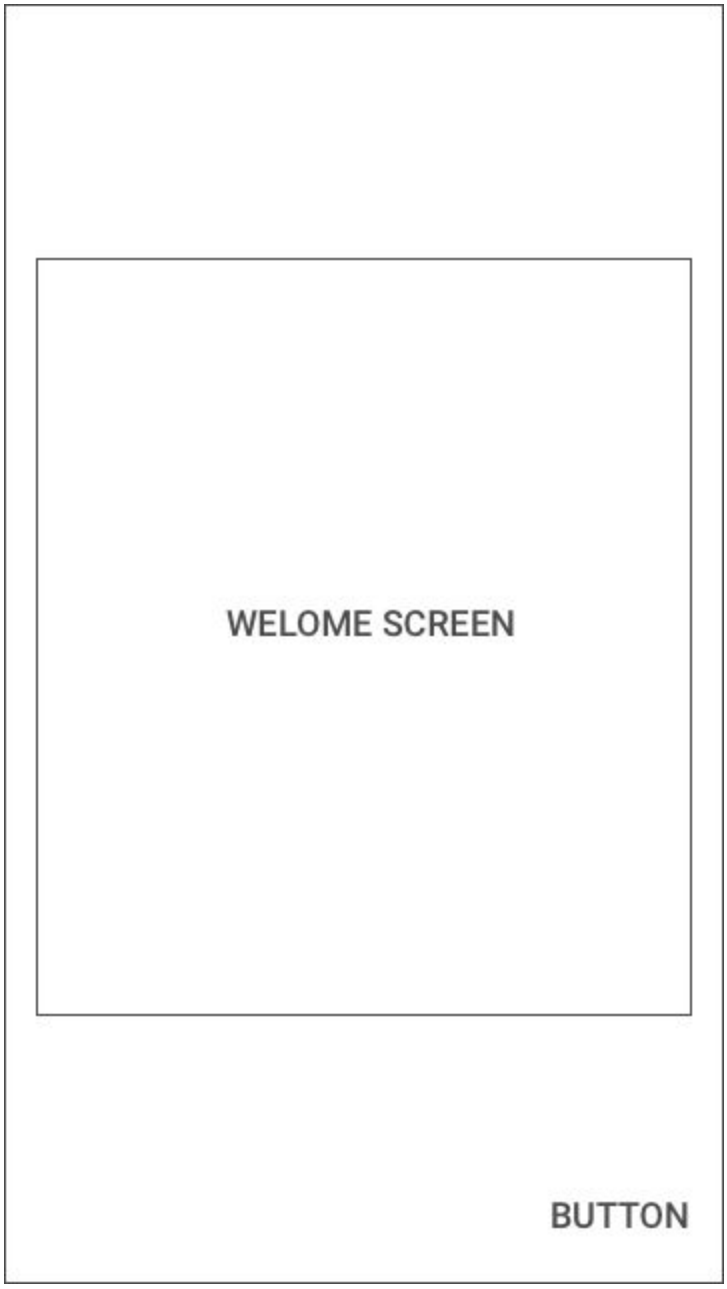
Intended for anyone looking for the latest breaking news and information on top stories, weather, business, entertainment, politics, and more.

Features

- Get news headlines in different languages from different trusted sources
- Save articles for offline reading
- Share stories to your social networks and E-mail/SMS
- Support for dynamic text. Increase or decrease the body text size to suits your preferences
- Get storylines with suggestions for further reading and other range of related topics
- Integrated Search: Quickly search for topics that interest you.
- Get current weather conditions with the very latest temperature, humidity, wind speed and more.

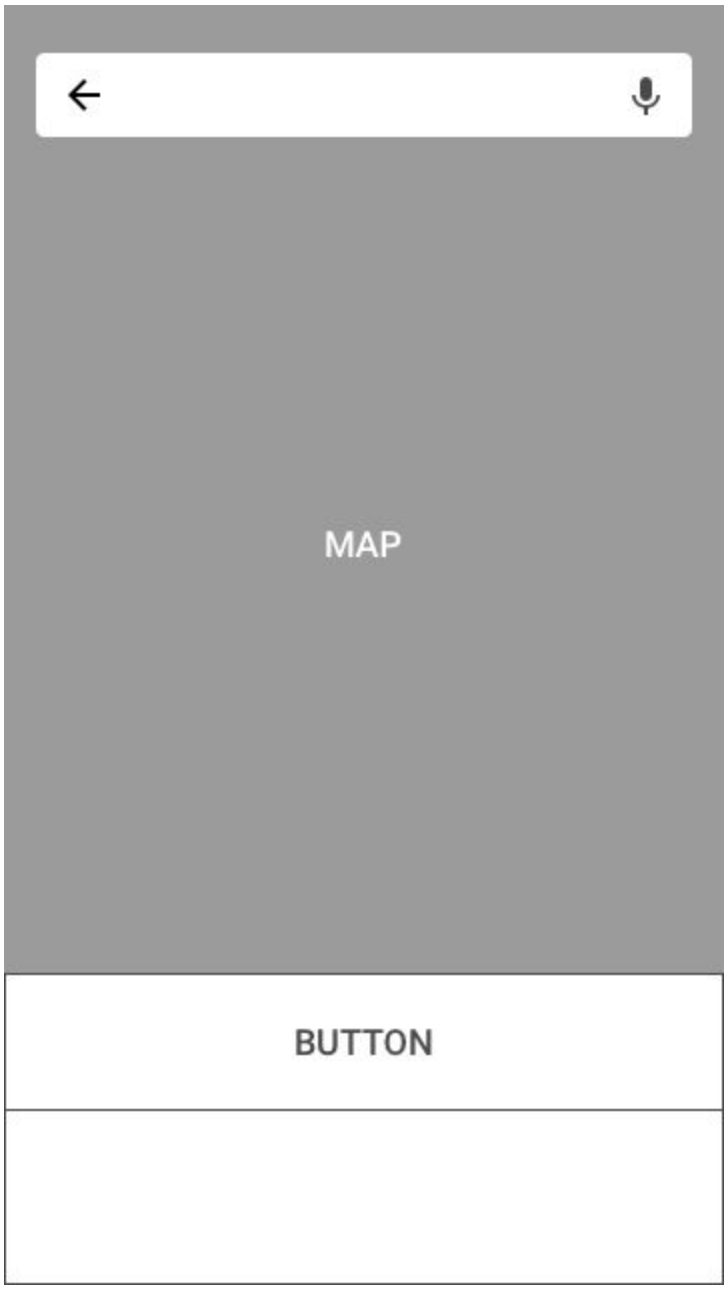
User Interface Mocks

1.Welcome Screen



A welcome screen with a brief introduction

2.Place Picker

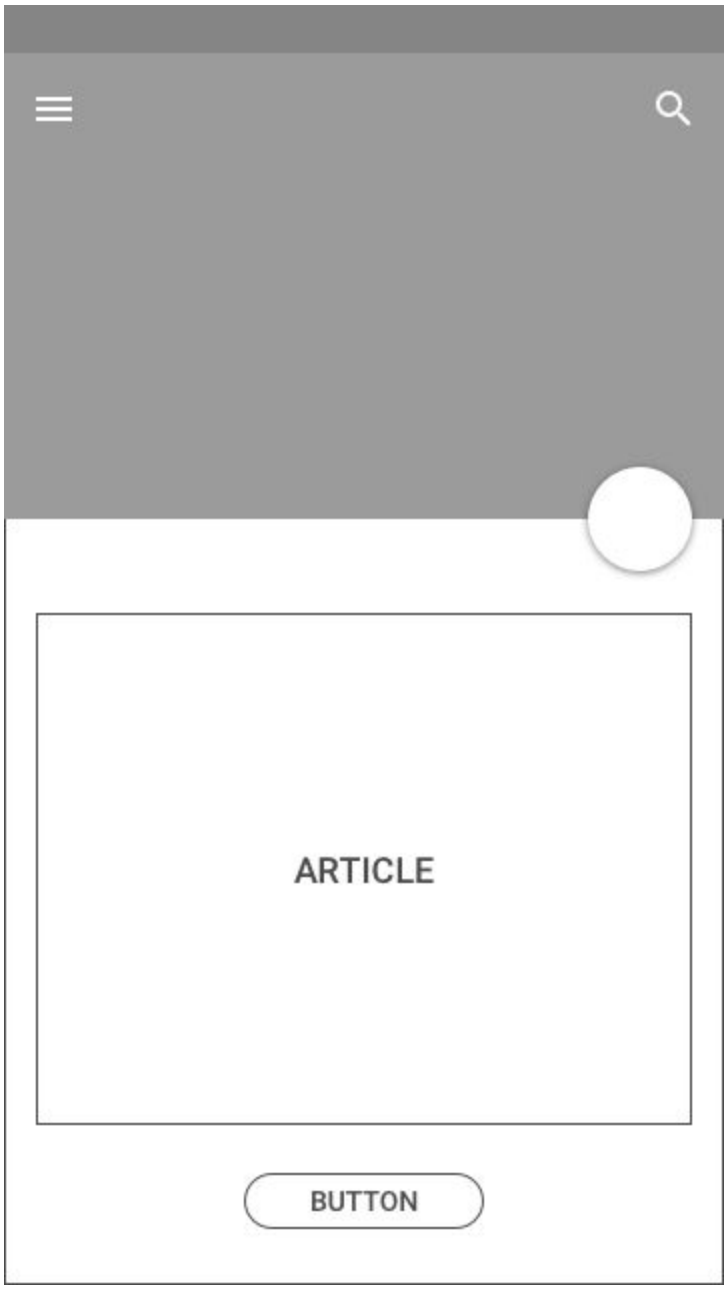


built-in UI widget for Google Places API.
Users can choose a place on the map or from a list of nearby places, including places corresponding to geographical addresses and local businesses.

3.Home Screen

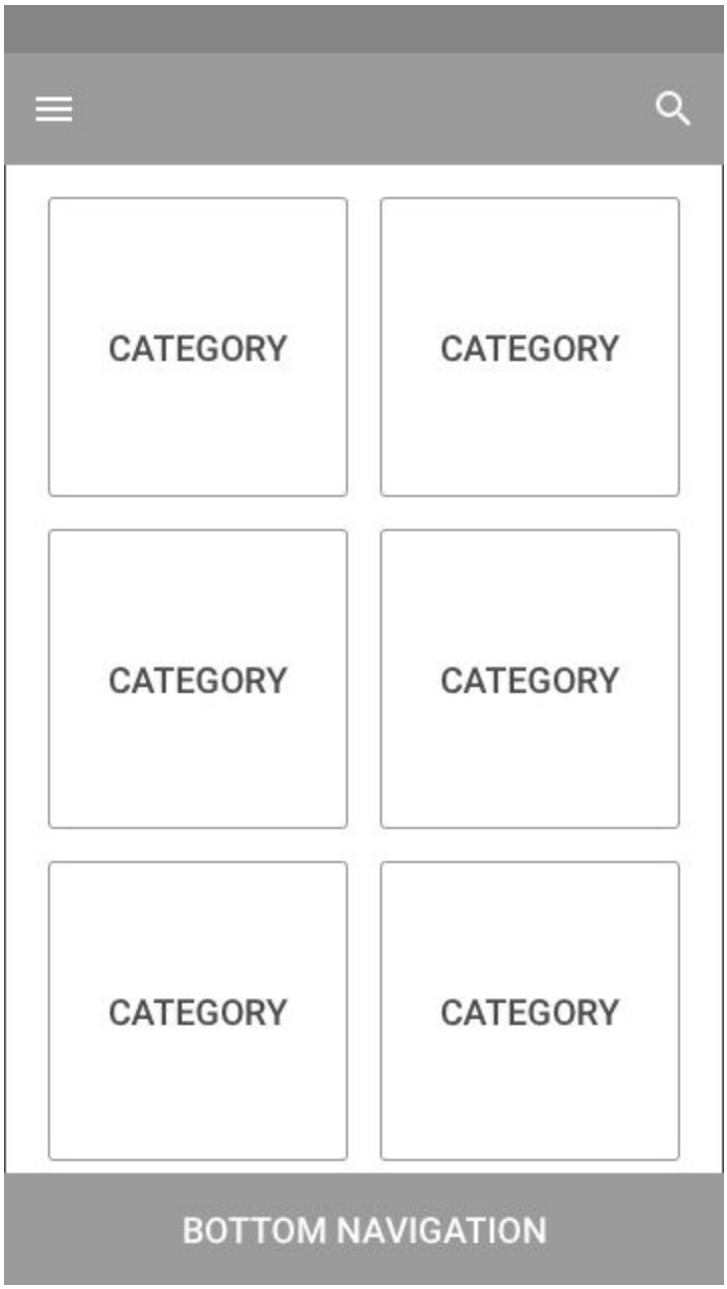
The Home Screen is being populated with a list of article headlines including current weather based on users location.

4.Article Details



Detail Screen for an article: Displays more information about an article . It also has buttons which allow users to add articles as bookmarks and to read full stories inside a webView.

5.Category Screen



Article Categories: For example Science, Politics, Money, etc.

7.Widget

An AppWidget that displays top headlines in a StackView.
A tap on any headline triggers its detailed page to be opened

Key Considerations

How will your app handle data persistence?

The app request data based on the selected location.This data is cached in the device memory and reuse without pulling data from the web service subsequently anytime the app is restarted. However, the data in the cache is updated at regular intervals to fetch the latest article headlines.The location details are also stored using SharedPreferences which can be cleared or changed by the user in a configuration screen.The app also saves articles marked as bookmarks in a local database(SQLite) using a Content provider and Loaders.
The Content Provider manages access to the database by using an address(Url) that points to a table or a path and the Loaders help in catching and running the database processes on a seperate thread.

Describe any edge or corner cases in the UX.

The Bottom Navigation allows users to navigate between different sections of the app.
On the “Article Details Screen”, Users can easily slide to the next page with the help of a ViewPager.

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

- Glide, to handle the loading and caching of images.
- Butter Knife, for view binding
- Retrofit 2 and OkHttp - for communicating with different API services.
- Dagger - Dependency injection to generate and provide runtime and singleton dependencies
- “ViewModel” from the Android Architecture components - to prepare data for the UI and to handle activity lifecycle on device configuration

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

Google Place Picker will allow a user to pick his or her location and generate location details. The location data collected will be filtered using Google Maps Reverse Geocoding and use as query parameters in different API endpoints.

Next Steps: Required Tasks

Task 1: Project Setup

- Create project and configure local git repository
- Setup license agreement and dimension keylines
- Create packages and directories
- Update and configure tools, manifest file, and dependencies
- Commit changes to the local git repository

Task 2: Implement UI for Each Activity and Fragment

- Build UI for MainActivity and a Welcome screen
- Implement Google Place Picker UI Widget with runtime uses permission for location services
- Create UI layouts with their corresponding activities and fragments
- Implement search logic and menus

Task 3: Setup Data Source

- Create a repository and model classes
- Setup network services and make calls to the respective APIs
- Create a local SQLite database and a Content Provider

Task 4: Set Up Test Classes

- Generate Testing Classes
- Create mocks with Mockito library
- Write test for the flow of data within the app

Task 5: Create Animations

- Setup shared element Activity Transition
- Define custom animation

Task 6: Implement bookmark logic

- Create a button(Bookmark) to add and remove articles to and from the database respectively
- Show empty state, if there are no results

Task 7: Create Widget

- Generate and customize default widget settings
- Build the UI and setup a configuration screen
- Setup Click listeners and the data source