GitHub Username: aldajo92

NotesGraph

Description

This is an app designed to be able to save values, with notes and be able to graph it in a time series. Useful if you need to save values for your expenses or if you have a lost weight plan, this app can store your values and visualize it with graphs. The app is written solely in the Java Programming Language.

Intended User

This app is designed for people that want to save reports with specific values, could be money, weight, volume or any dimension that needs to store with a note, and visualize in general with a graph.

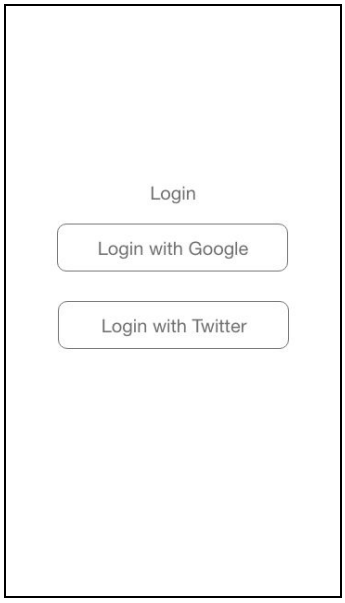
Features

- Save notes with values.
- Edit and delete information.
- Take screenshots to complete information.

- Visualize all values in a simple dashboard.
- Share content on social media.

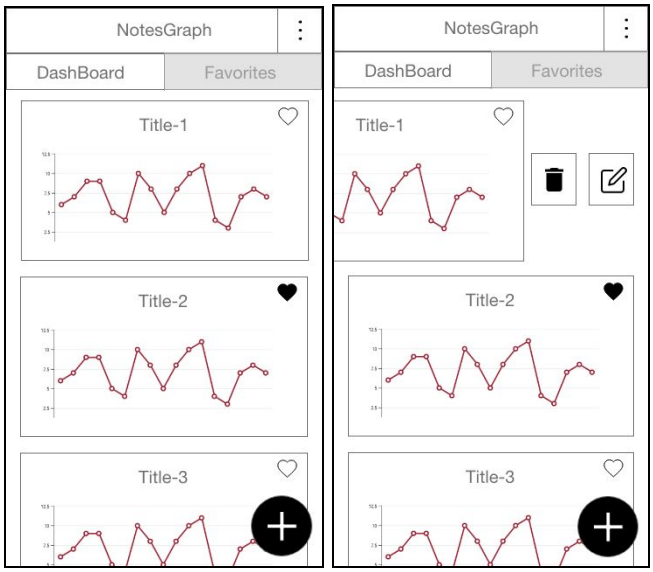
User Interface Mocks

Screen Login



Manage login authentication using firebase.

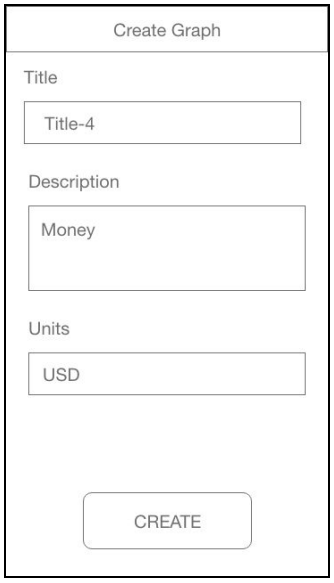
Screen Dashboard



This is the home screen to see all the dataset that the user has in the application represented by cards with the graph associated and the title. Users can mark as favorite each graph and can

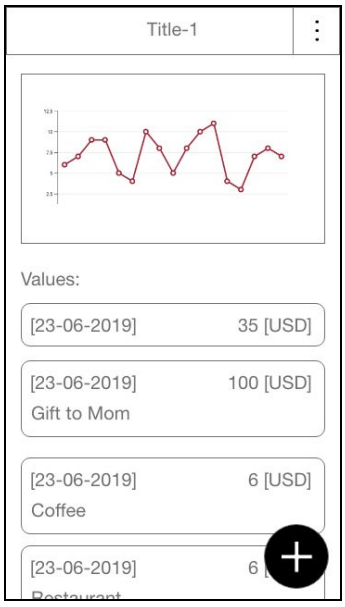
show it in the favorite section. The floating button allows for creating a new dataset. Swapping the card, shows hidden options to edit general information of the card and remove it.

Screen to Create a new dataset.



This form shows what kind of information is required to create a new dataset.

Screen Detail dataset.



This screen shows the detail information when the user selects a card, that allows interacting with the graph and lets to add and remove entries.

Screen Add/Edit Entry.

Those screens show how to view, edit or create an entry.

Widget

This screen represents the widget, that shows the last values added in general for the application.

Key Considerations

Data persistence:

Data persistence is implemented using Room Library, which stores each dataset with its corresponding id. Each dataset, have an array of each entry that has the date, note, and location of each image.

Corner case in the UX.

When a user creates a dataset or deletes all entries, a place holder should be placed to replace the missing data.

Libraries to include.

- MPAndroidChart (version `1.7.5`): Library to manage each graph, including the option to interact with it with gestures and animations.
- Room, LiveData and ViewModel (version `2.2.0`): Libraries used to persist data in the application, and manage the business logic.
- Play Services Ads (version `10.0.1`): Google Play services library to have ads in application.
- Firebase Storage (version `18.0.0`): To store images added by the user.
- Firebase Auth (version `18.0.0`): To manage the login process.
- Firebase Database (version `18.0.0`): To manage the database in the cloud.

Google Play Services.

- Play Services Adds
- Google Drive for backup.

Required Tasks

Task 1: Project Setup

Create project from scratch with the following specs:

- application will be written solely in the Java Programming Language.

- Java version 1.8
- Android SDK 28 [Minimum SDK 21]
- Gradle version 5.1.1

Create some basic screens.

- Create a splash screen
- Create the dashboard screen with mock data.
- Create cards and the detail screen to visualize the mock data.
- Include tests to validate functionality.
- The app keeps all strings in the strings.xml file.

Task 2: Implement Room Library

Define each entry and data set as entities in Room.

- Create the Screen to include new datasets.
- Create the Screen to include new entries for each dataset.
- Add and remove behavior using the Room Library, defining the entities for each entry and dataset. Each dataset, have an array of each entry that has the date, note, and location of each image.

Task 3: Implement MPAndroidChart Library.

Create the Graphs based on each dataset.

- Add graphs to dashboard Screen and detail Screen.
- Edit feature for datasets and entries.

Task 4: Allows Users to add pictures.

Add the feature to allows Users to take pictures or upload images to complement information for each Entry in the dataset. This picture will show only in the detail Screen:

- Add or edit picture for each entry.
- Upload images using Firebase Storage.

Task 5: Add Accessibility features.

Includes support for accessibility with content descriptions.

Task 5: Backup Feature.

Configure the app to allows backup.

- Create a backup and export to Drive.
- Sync with Firebase Database.