GitHub Username: ogasimli

Manat

Description

Manat is an easy to use and free app, which shows daily exchange rates of foreign currencies relatively to Azerbaijani manat (AZN). Manat allows users to convert AZN into other currencies and vice versa.

Manat uses official daily exchange rates published by the Central Bank of Azerbaijan (CBAR). However, since the public API of CBAR does not allow to retrieve exchange rates for more than one particular date at once, Manat will get the data from a NoSQL database created and maintained by me specifically for this application. The values in database are kept automatically up to date with the help of a special backend application.

Intended User

Manat targets any user living in or travelling to Azerbaijan. Furthermore, it will be especially useful for professionals dealing with exchange rates on a daily basis (i.e. accountants, auditors and other professionals of finance sector).

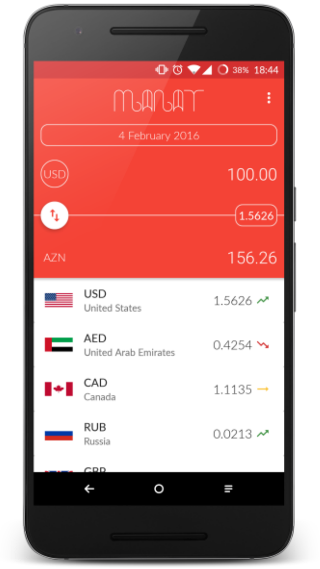
Features

Main features of Manat:

- Discover exchange rates by date;
- Convert AZN into foreign currencies and vice versa;
- Discover exchange rates directly from your home screen via collection widget;
- Share currency rates.

User Interface Mocks

Screen 1



4 February 2016

USD 100.00

1.5626

AZN 156.26

USD United States	1.5626	↗
AED United Arab Emirates	0.4254	↘
CAD Canada	1.1135	→
RUB Russia	0.0213	↗

This is the main view when the user opens the app. The screen consists of 2 sections:

1. Currency convertor;
2. List of exchange rates of other currencies;

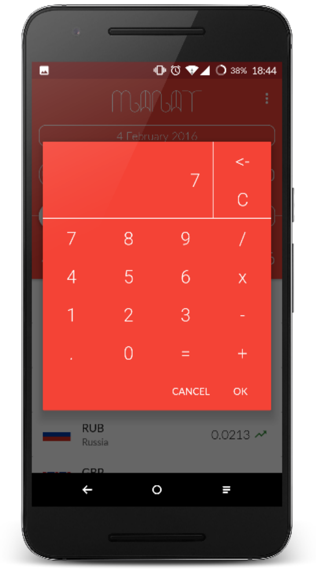
To enter the amount the user taps on one of two amount fields, which opens the calculator fragment shown in Screen 2.

The user can change the date by taping on the date button, which will open the calendar fragment shown in Screen 3.

In order to change the currency to be converted into AZN, the user is required to tap on the foreign currency button. This action will result in opening of a new activity containing the list of available currencies. Please see Screen 4.

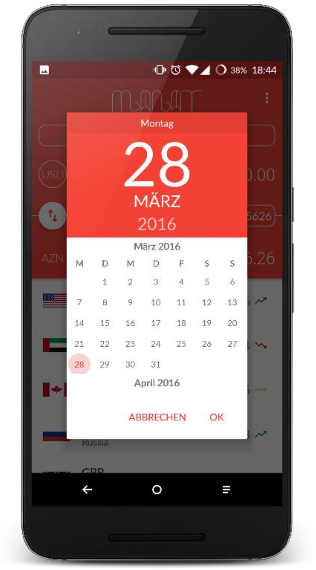
Tapping on one of the list items will redirect the user to the details activity shown in Screen 5.

Screen 2



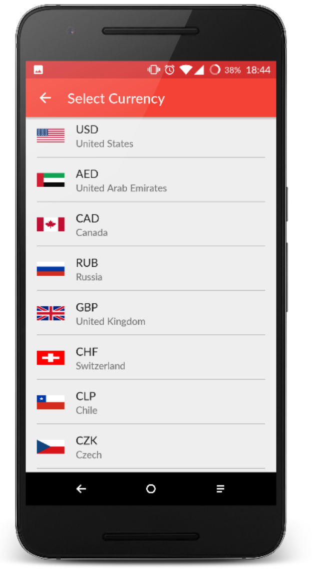
With the help of this fragment the user will be able to make basic calculations or just enter the amount to be converted.

Screen 3



The calendar fragment will help the user to get the exchange rates for desired dates.

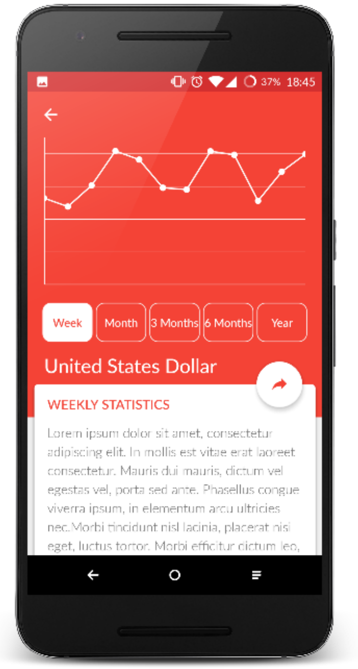
Screen 4



USD	United States
AED	United Arab Emirates
CAD	Canada
RUB	Russia
GBP	United Kingdom
CHF	Switzerland
CLP	Chile
CZK	Czech

The user will be able to select the desired currency and make conversion from/to it from the main activity.

Screen 5



United States Dollar

WEEKLY STATISTICS

Lorem ipsum dolor sit amet, consectetur adipiscing elit. In mollis est vitae erat laoreet consectetur. Mauris dui mauris, dictum vel egestas vel, porta sed ante. Phasellus congue viverra ipsum, in elementum arcu ultricies nec.Morbi tincidunt nisl lacinia, placerat nisi eget, luctus tortor. Morbi efficitur dictum leo,

Details activity will give the detailed statistical information on a currency during the selected period. The user will be able to share the entire statistical information or the current exchange rate with the help of the floating action button.

Key Considerations

How will your app handle data persistence?

The app will use SQLite and a Content Provider to store some data internally for caching purposes. Settings will be saved using SharedPreferences.

Describe any corner cases in the UX.

The user will be able to open the app by tapping on a widget. This will take the user to the main activity.

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

- **Android Design Support Library** to follow Material design guidelines;
- **Google Analytics** to analyse users' behaviour;
- **Google App Invites** to integrate app invites;
- **Google Mobile Ads** to insert banner ads;
- **ButterKnife** to bind views;
- **MongoDB Java Library** to fetch data from online database;
- **Inquiry** to handle and use SQLite database;
- **Glide** for image loading;
- **MPAndroidChart** to implement graphics to show the rate statistics.

Next Steps: Required Tasks

Task 1: Project Setup

- Create project with simple activity;
- Add the library dependencies described in the below to the build.gradle file;
- Create an application Class to handle the initialization of required libraries;
- Create Helper class for libraries according to the Business Logic;
- Create project structure: Entities, Listeners, UI -> Fragment, Activities, Custom Views.

Task 2: Implement UI for Each Activity and Fragment

List the subtasks. For example:

- Build UI for MainActivity:
  - Build UI for Currency Convertor;
  - Build UI for Calculator DialogFragment;
  - Build UI for list of currency items.
- Build UI for SelectCurrencyActivity;
- Build UI for DetailActivity.

Task 3: Pull Data

- Implement an IntentService to pull the information from online NoSQL database.

Task 4: Implement Data Models and Content Provider

Build up the data models and implement data persistence.

- Create data model classes;
- SQLite database setup and CRUD;
- Loader/adapters and UI.

Task 5: Implement Google Play Services

- Add Google Mobile Ads to the app to generate revenue;
- Add Google App Invites to facilitate the user acquisition process;
- Add Google Analytics to analyse users and their behaviour.

Task 6: Other Tasks

- Build a collection widget;
- Accessibility and RTL;
- Testing.