GitHub Username: avandriets

Environment monitor

Description

This application designed for monitoring ecology situation in city and registration different fact violation of ecology for example rubbish dump. Also people can complain about working municipal services. You can register fact and attach photos to it. Everyone can comment facts by messages inside fact.

Intended User

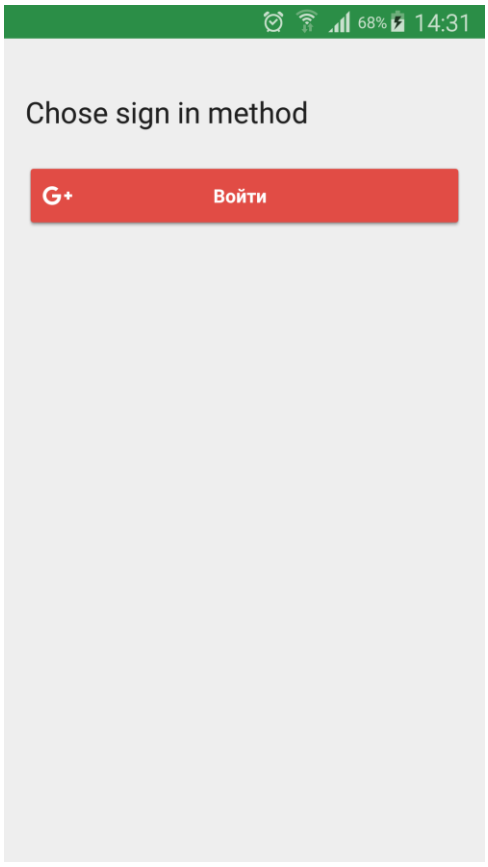
This application for people who lives in city and who want to change environment.

Features

- Save information about facts.
- To view information about facts on map and list
- Takes pictures.
- User can comment facts.

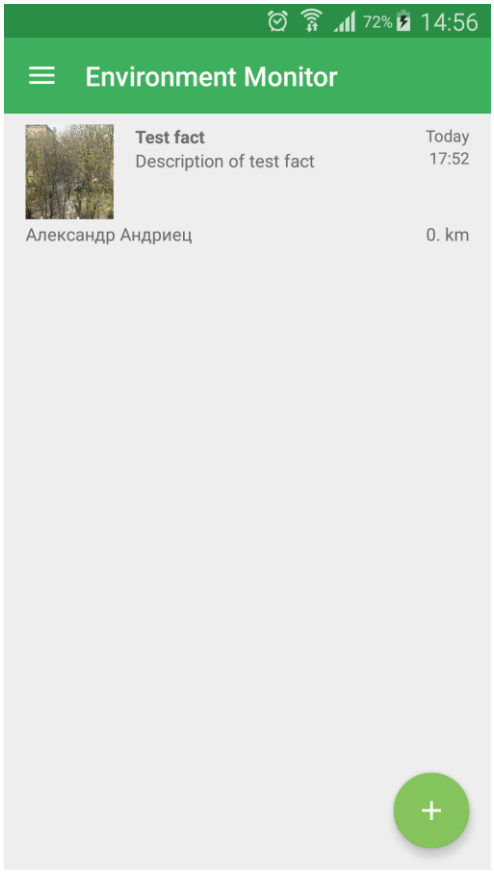
User Interface Mocks

Registration screen



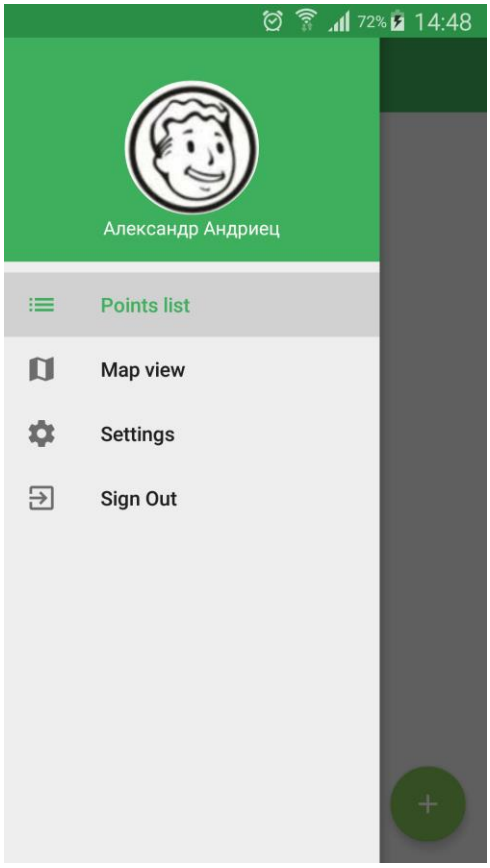
Users can register in program by their Google profile.

Main screen



It is main screen of program all facts (points) show in list. User can add new points to list.

Drawer screen

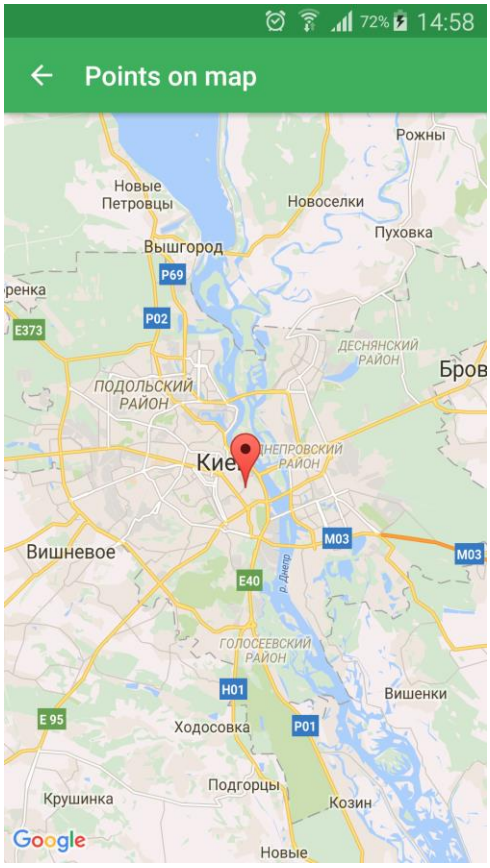


Represents information about user such as first name, last name and avatar from google profile.

Help user to navigate in application:

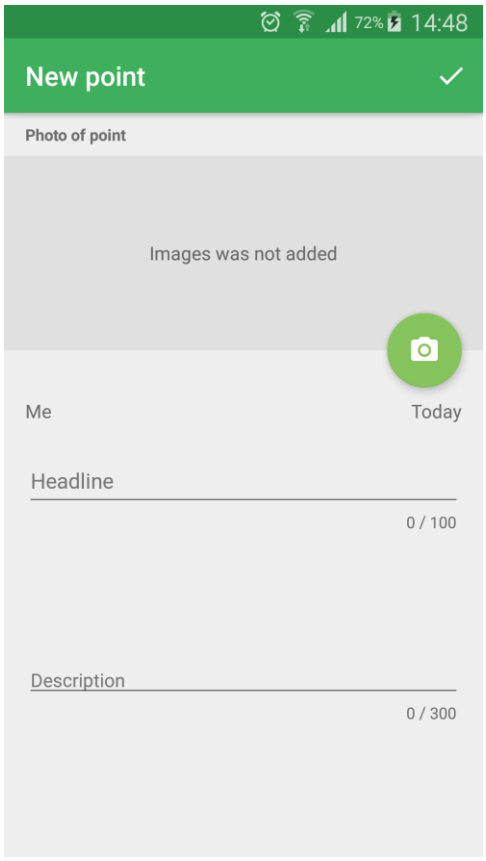
- Switch between list, map or settings screen

Map screen



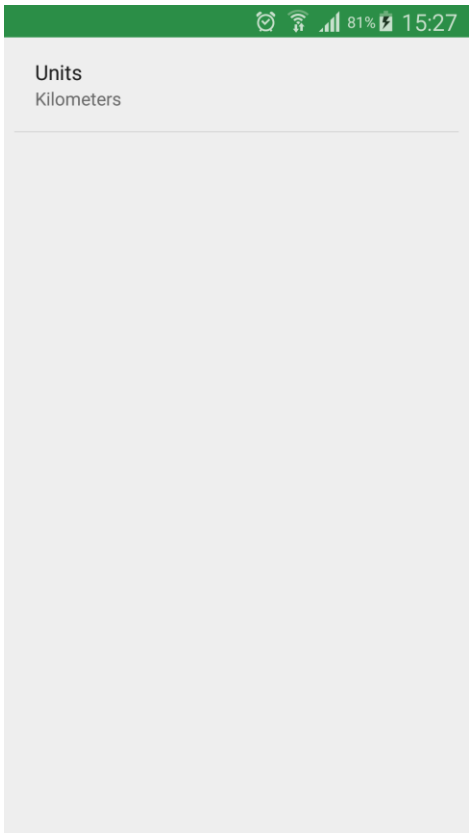
Shows facts (points) on the map.

Screen "new point"



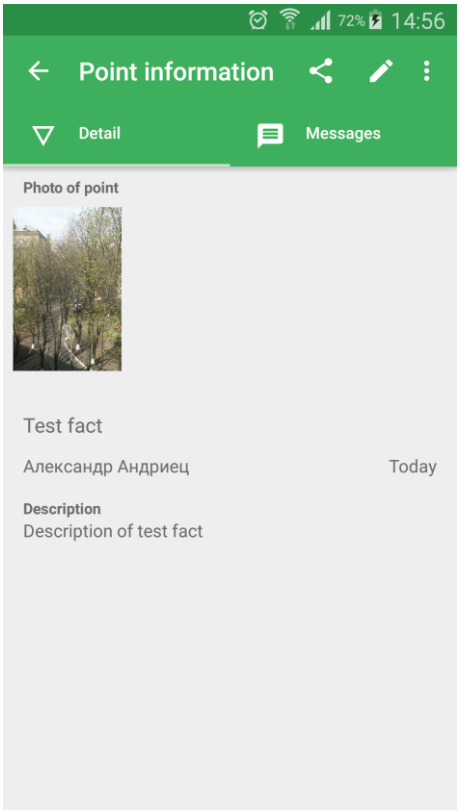
Create new point in local data base. User can add headline of point, short description and add photos from camera or from local starage.

Settings screen



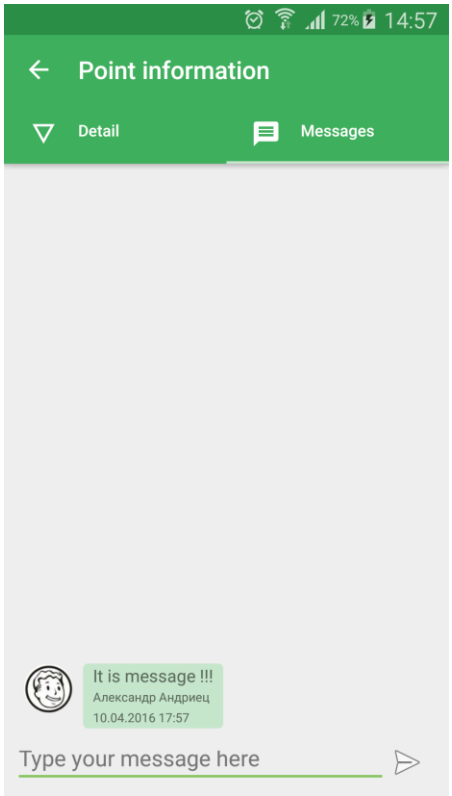
User can choose in with units will show distance from him to this point.

Detail point view



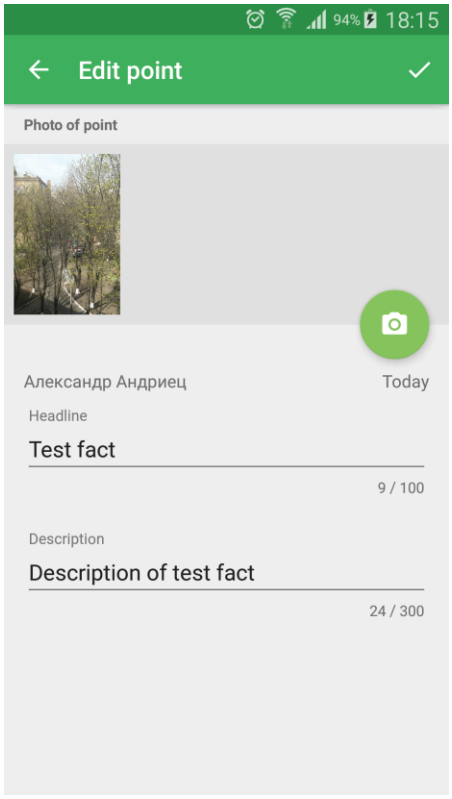
Detail view of fact. User can share information about fact also user can edit or delete point if he is owner.

Send message screen



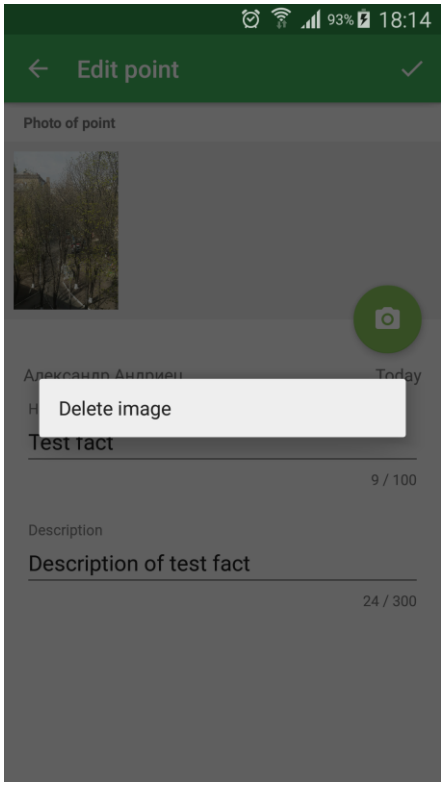
User can write messages about point.

Edit point



User can edit own records

Delete photo



Key Considerations

How will your app handle data persistence?

In my app data store in local sqlite database. I use OrmLite to get data from DB. Also I use Content Provider to get data from database when I use Loader. I combine Content Provider and OrmLite in my application. Data store in local database and than send to the server in the Internet.

Describe any corner cases in the UX.

Detail point view

User can view photos on full screen if he/she taps on picture in list.

In detail point view user can delete photo if he/she long press on a picture, after that show menu whit actions.

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

com.j256.ormlite - to get data more easy from local SqLite database then with Content Provider

Butterknife - it helps me decrease amount of code when I bind UI and local variables.

Hdodenhof - helps make round avatars of users

Okhttp3 - HTTP client

Picasso - helps download images from the internet

Retrofit - helps to work with REST services

Next Steps: Required Tasks

Task 1: Project setup

- Add support libraries (ormlite, Butterknife, Hdodenhof, Okhttp3 , Picasso, Retrofit)
- Configure ORMLite
- Create support classes for database

Task 2: Implement UI for Each Activity and Fragment

- Build UI for MainActivity
- Add support TABLETS to MainActivity
- Build UI for PointsList
- Build UI for New Point
- Build UI for Edit Point
- Build UI for Settings activity.

Task 2: Implement Google Sign In

- Implement Google Sign In
- Build Login activity

Task 3: Implement data synchronization with web service

- Create interfaces for REST service
- Add support methods for data sync
- Create SyncAdapter for sync local data and data in the Internet (web service)

Task 4: Implement Google GCM

- Configure application for support GCM
- Add support classes (MyGcmListenerService, MyInstanceIDListenerService, RegistrationIntentService)
- Add device registration on web service

Task 5: Implementation Google Maps

- Configure application for support Google Maps
- Build Map Activity (on it will show points)

Task 6: Connect Points List Activity and Map Activity

- Create drawer for switch between views and represent information about

Task 7: Prepare application for publication

- Configure key store for application
- Generate signed application
- Make icons
- Publish to Google Play