GitHub Username: maxim-yudin

VK Flow

Description

VK Flow is a simple fast client for the largest European social network - VK (www.vk.com). It allows you to follow some news from a life of your friends, make your likes and read/write comments.
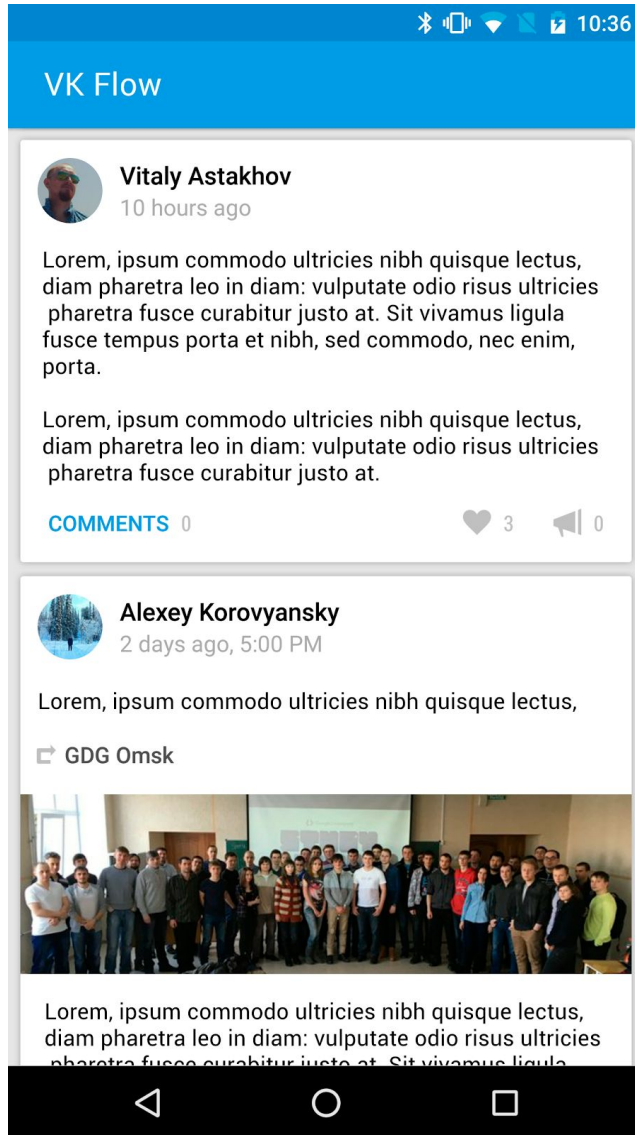
Intended User

A user of this social network who is interested in following the news of his friends.

Features

- See the news flow
- Make some likes
- Read/write comments

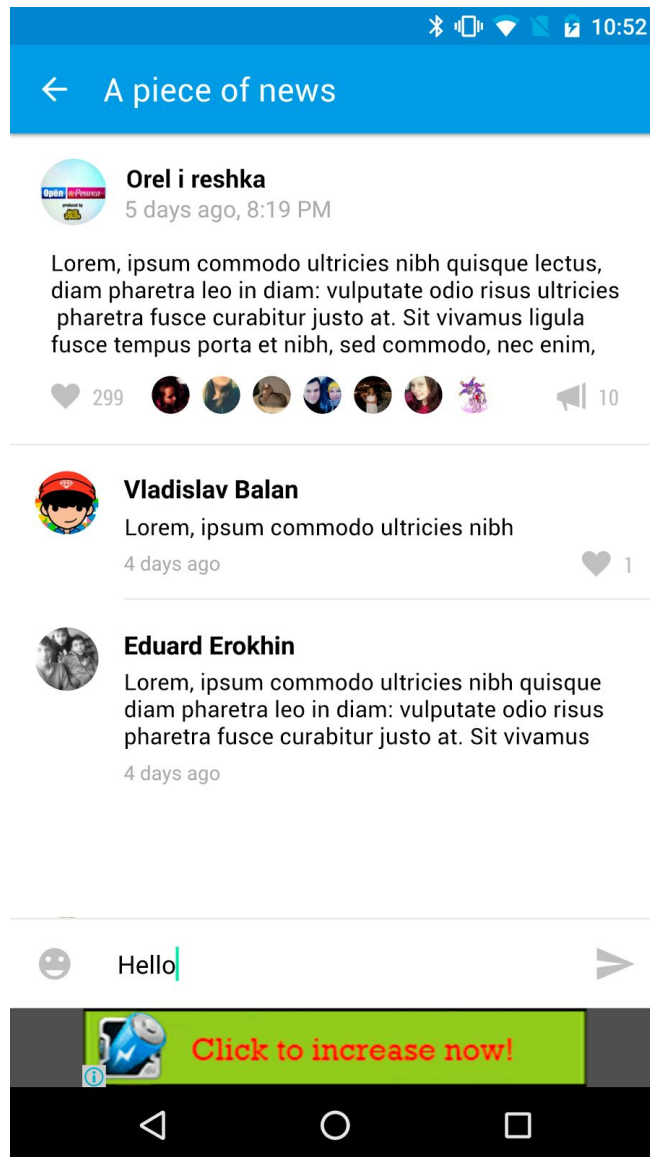# User Interface Mocks

Screen 1. Main activity

It's main activity where you can see a list of news card of your friends after VK authorization.

Screen 2. News activity

It's news activity where you can see a piece of news, some comments of another users. You can also write a comment to this news, share it and make likes. Here an user will be seeing some AdMob ads in the bottom of the screen.

Screen 3. Login activity
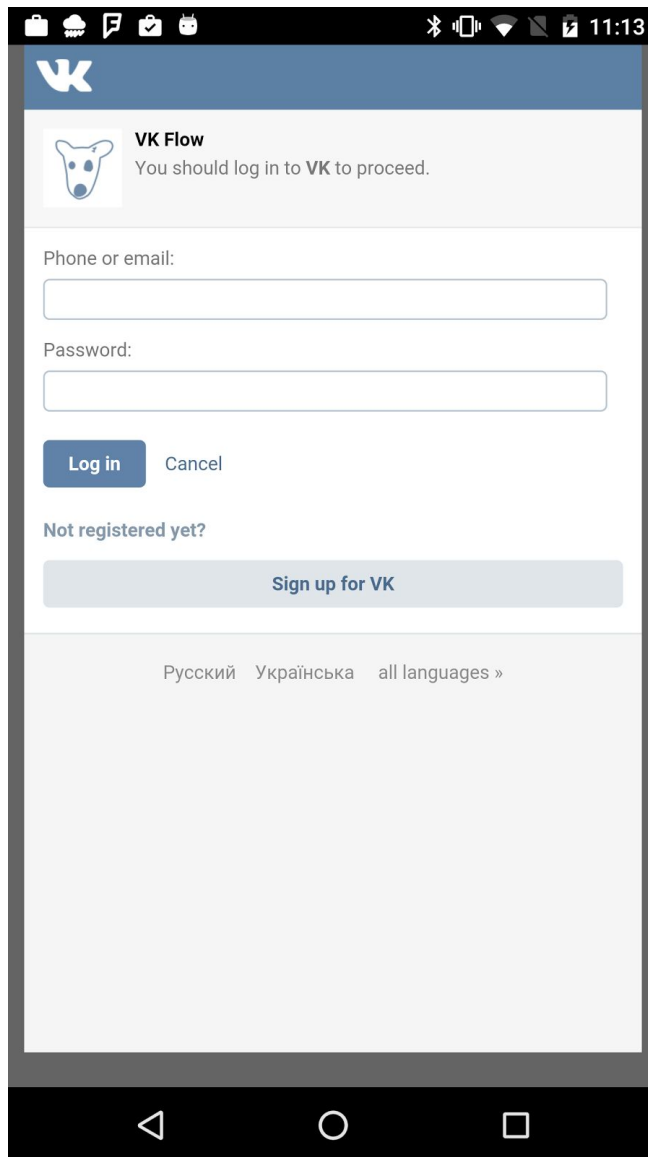
It's a simple webview window, where you can login to VK.com for an API access.

Key Considerations

How will your app handle data persistence?

I will build my own content provider for storing data and inflate it from API using Intent service.
In some cases I will be using SharedPreferences for storing some preferences.

Describe any corner cases in the UX.

- User can update news list using swipe-to-refresh pattern.
- User can click on any piece of news that displaying news window where he can read some comments and make some actions (likes, share)
- User can make 'like' action from news list or from one piece of news.
- User can post some comment when he is on one piece of News window.
- User can share news

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

- **VK Android SDK** for using API of VK.com for reading news and making some another actions like making likes.
- **ProviGen** for creating content provider.
- **Picasso** for handling the loading and caching of images.
- **Google Support Library (AppCompat, Design, CardView, RecyclerView)** for using Material Design UI components, font styles etc.
- **Google Gson** for parsing JSON representation from API to Java objects
- **Google Play Services library** for integrating AdMob for showing some ads and Google Analytics for tracking user activity.
- Probably, I will be using **Retrofit** for some API operations.

Next Steps: Required Tasks

This is the section where you can take the main features of your app (declared above) and decompose them into tangible technical tasks that you can complete incrementally until you have a finished app.

Task 1: Project Setup

- Add library dependencies to build.gradle file.

Task 2: Implement UI for Each Activity and Fragment

- Build UI for LoginActivity for VK authorization.
- Build UI for empty screens (no news, no internet connection etc.)
- Build UI for a news card row and realize some list adapter for displaying this news card list.
- Build UI for MainActivity with news card list using created adapter.
- Build UI for NewsActivity (one piece of news)
- Build UI for a widget who will be displaying last 5-10 news

- Build layouts for using created UI forms for tablets

Task 3: Implement Google Play Services

- Set up Google Analytics and creating some methods for sending user events.
- Set up Google AdMob ads. We will be using one interstitial ad after we will be returning to MainActivity from NewsActivity. We will also be using some banner on NewsActivity in the bottom of the screen.

Task 4: Creating content provider, models and api calls helper methods

- Creating data models
- Creating Content Provider (db provider and contracts) using ProviGen
- Using VK SDK for some api calls.

Task 5: Handle Error Cases

Handle some errors like no internet connection, no data etc and correct displaying this information for users.