GitHub Username: avgasanov

Catch my car!

Description

Catch my car is a very specific social network and challenging game for car owners. It is application for popular challenge among Azerbaijani car owners. The idea is very simple:

1) Make a photo of your friends car when you accidentally see their cars somewhere on your way.
2) Post this photo on social network
3) Ask for reward (have a cup of tea together)

It is like being catched by police radar, but this time by your social network friends in a very nicely manner. Rewards for catching friends as a usual include having a cup of tea together. So it inspires people to socialize and have a great time together.

Intended User

Intended users are car owners. Due to some specifics of people mentality in Azerbaijan and neighbour countries, application especially intended for users from these countries (Russia, Georgia, Turkey)

Programming environment

Application will be written solely in the Java Programming Language in Android Studio.
Android Studio, Gradle and libraries versions:

Android Studio	3.1.0
Gradle	4.4
Chat-sdk	`4.0.26`
Butterknife	8.8.1
Glide	4.7.1
Google Places	`15.0.1`
Google Play Services	`4.0.1`

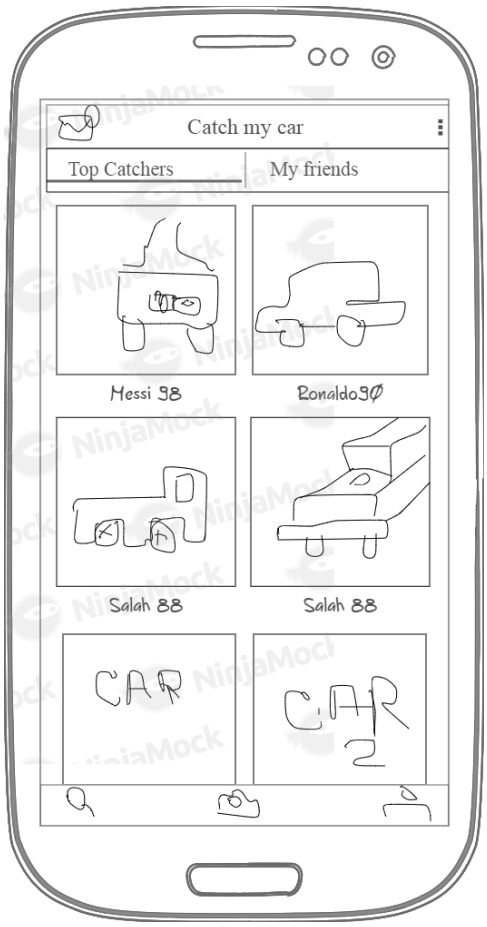
Firebase Analytics	16.0.1
Firebase Realtime Database	16.0.1

Features

- Takes pictures with geoposition data
- Creates user account
- Adds more than one car to be cathed
- User confidentiality settings
- Maintains users friendships
- Chatting features
- Notifies about events (example: when car is “catched”)
- Searches for cars

User Interface Mocks

Screen 1

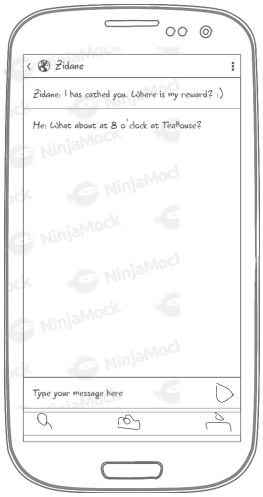


This is the main application screen. As you may see there are two tabs representing the most active users and friends of current. Message button in the right corner is for chat module of an application. The first icon in the bottom left corner is to search users by their cars. Middle icon is to take a picture. The icon in the right bottom corner is for user profile.

Chat contact list screen

This is the screen of chat.

Private chat screen



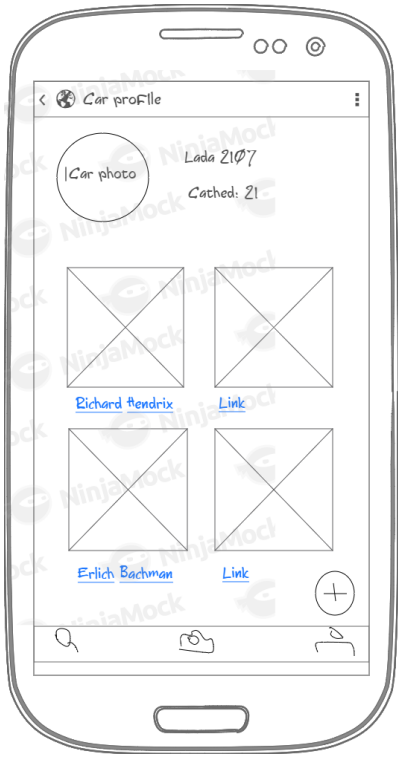
This is the screen of private chatting.

User profile screen

This is the screen of user profile. User can add any number of its cars (by pressing FAB plus button). If any of this cars will be catched (by their plate numbers) he will be notified.

Adding new car screen

Car profile screen



This screen is to represent a car from user profile. On user names click the other user profile will be opened

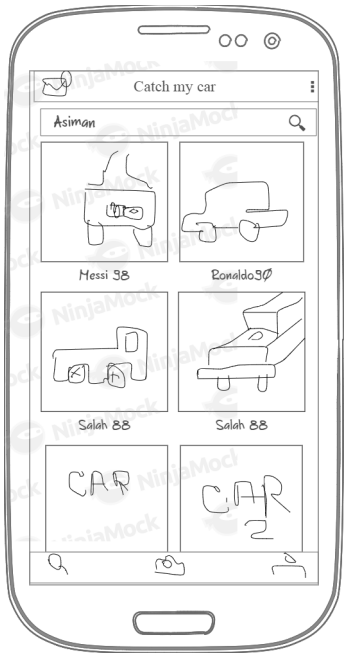
Other user profile screen

The other user profile will be as in this mock. Button is to send friendship request.

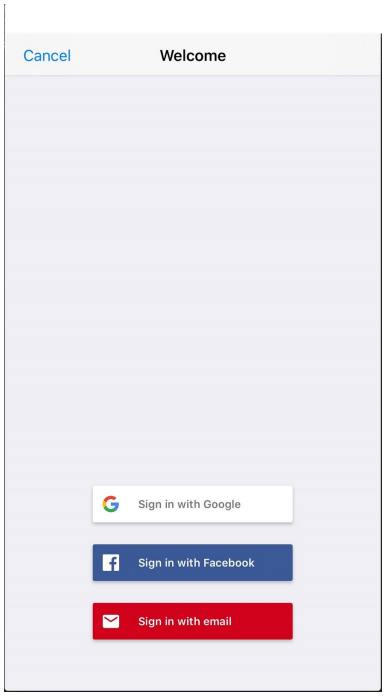
Settings screen

Settings screen is for confidentiality settings. Such as be searchable by car and user name.

Search screen

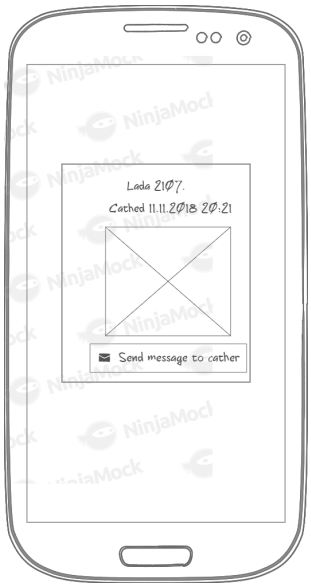


Log in screen



Login screen is from firebase ui

App Widget



App widget will show the last cathed photo. Button is to send message to catcher.

Key Considerations

How will your app handle data persistence?

1) I'll use Firebase Realtime Database to persist user data
2) I'll use Firebase Cloud Storage to store photos

Describe any edge or corner cases in the UX.

All strings will be stored in strings.xml file. Image assets will be stored as drawables in drawable folder. User data images will be stored in Firebase Cloud Storage. Colors will be stored in colors.xml. Themes will be stored in styles.xml
Layouts will use start and end to allow RTL layout. Images will have description. For user data images it will be description as "catched car image", "user avatar" and etc. For other general purpose buttons it will be more descriptive and specific.
Users will be able to use application only after registering. Registering and authentication will be through Firebase UI.
Pressing photo camera icon will create intent and start system activity to take a picture. Picture will be then processed to add geotag and time in my application.
Users will be able to set confidentiality rules. To define whether they want their cars to be found by search.
Users will be able to search other users by their usernames and car plate numbers, without any initial settings.
Intent service will be used to fetch data and show on widget.

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

Glide library - for image loading and caching
Chat SDK - for creating chat module for application.
ButterKnife - to avoid boilerplate code

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

Google location services will be used to add geodata on image during processing
Firebase analytics will be used for analytics purposes
Firebase Authentication will be used for user authentication
Firebase Cloud Functions will be used to notify users about events.

Next Steps: Required Tasks

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

Task 1: Project Setup

Create new project. Add necessary dependencies (new stable versions) for libraries.

Task 2: Implement UI for Each Activity and Fragment

- Build UI for MainActivity
- Build UI for user profile activity
- Build UI for chat activity
- Build UI for conversation activity
- Build UI for search activity

Task 3: Create firebase project

- Create new project
- Set project for new application
- Download json and add it to project

Task 4: Add firebase authentication ui

- Google and facebook login
- Check for infinity loop with back button

Task 5: Add chat sdk

- Initialize chat sdk
- Add Firebase File Storage

- Add Firebase Push Notifications

Task 6: Implement notification with cloud functions

- Write and test server code
- Test notifications

Task 7: Create app widget

- Create app widget model
- Implement app widget provider class
- Implement intent service to fetch data on database events