

[Description](#)

[Intended User](#)

[Features](#)

[User Interface Mocks](#)

[Screen 1](#)

[Screen 2](#)

[Key Considerations](#)

[How will your app handle data persistence?](#)

[Describe any corner cases in the UX.](#)

[Describe any libraries you'll be using and share your reasoning for including them.](#)

[Describe how you will implement Google Play Services.](#)

[Next Steps: Required Tasks](#)

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

Intended User

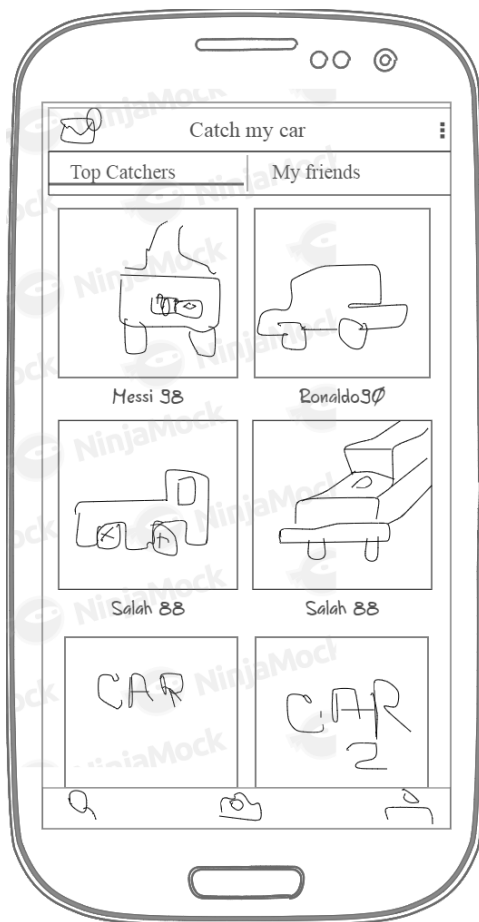
Intended users are car owners. Due to some specifics of people mentality in Azerbaijan and neighbour countries, application especially intended for users from these countries (Russia, Georgia, Turkey)

Features

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

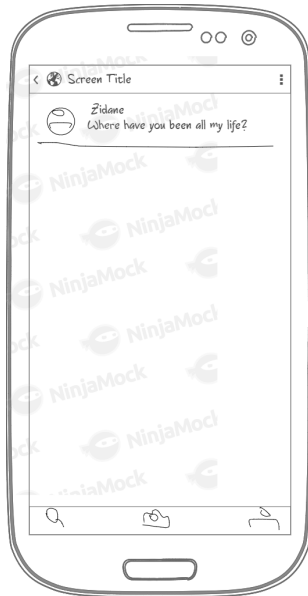
User Interface Mocks

Screen 1



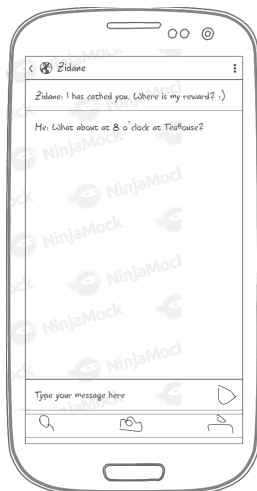
This is the main application screen. As you may see there are two tabs representing the most active users and friends of current. Message button in the right corner is for chat module of an application. The first icon in the bottom left corner is to search users by their cars. Middle icon is to take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

Screen 2



This is the screen of chat.

Screen 3



This is the screen of private chatting.

Screen 4



This is the screen of user profile. User can add any number of its cars. If any of this cars will be caught (by their plate numbers) he will be notified. The screen where will be his car photos will be very same to the Screen 1.

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.

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.

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