# Software requirements specification for project “Citylogia”

## 1. Authors

Ilya Volf, Andrew Weiss, Igor Chiesov.

## 2. Introduction

Many people have problems with finding new interesting places for taking photos or just taking a walk in Novosibirsk.

Our goal is to make the process of finding new interesting places worth visiting simpler, more convenient and quick.

To do so, we are going to create a mobile application which will help users to find out about some exciting non-trivial spots according to their personal preferences and preferences of their friends as well as share their opinion about visited places with other users.

## 3. Glossary

* Pronouns related to the word “user” will be in plural: user’s = their, to the user = to them, etc.

1. A **card of a place** is a block which consists of a picture of a place, its name, rating and short description. Such cards may be found on the main page or on the page of search results.

2. **Collections** are a folder on the user’s profile page where the user may store places which have interested them.

3. **Geographical preferences** are preferences which are based on locations of places. For example, the user may want to visit places only downtown, in some specific district or by a river.

4. An **interesting place** is a place which the user considers or would consider as one worth visiting.

5. The **main page** (**“Interesting places” feed**) is a core page which is opened when the application is launched. The feed is a list of cards of interesting places.

6. The **rating** of a place is a number from 1 to 5 which is an arithmetic mean of all grades of this place committed by users.

## 4. Actors

A user of the application which might be a tourist, a photographer, an artist or just an ordinary citizen that is looking for new places to pay a visit to alone or with friends.

A user finds places for visiting. Besides this, they may rate a recently visited place and share their experience about a place with other people.

# 5. Functional requirements

## 5.1. Strategic Use-cases

### 5.1.1. The user finds an interesting place for paying a visit to:

- The user receives information about recommended places for visit in the form of cards on the main page / The user receives information about location of places by means of their indication on the map (see 5.2.1, 5.2.2).
- The user looks for a place using the search with the function of filter (see 5.2.3).
- The user receives detailed information about a specific place such as its description, pictures, rating and reviews (see 5.2.4).
- The user adds a place to collections (see 5.2.5).

## 5.2. Use-cases for User

### 5.2.1.

The user receives information about recommended places for visit in the form of cards on the main page.
**Actor:** user of the application

**Goal:** find an interesting for a visit place

**Main success scenario:**

1. The user goes to the main page.
2. The user scrolls different places in the form of cards from the category “recommended” which are based on the overall rating of a place, the user's history of visited places, the history of visited places of the user's friends and user's specified preferences.
3. The user learns about places which may interest them and adds them to collections (see 5.2.8).

**Alternative scenario 1:**

1. The user does not find interesting for them places or they want to find nearby places or places of some specific categories.
2. The user specifies places which they want to visit using the filter (by distance, by category of places).

**Alternative scenario 2:**

1. The user is not logged in.
2. The cards are given only in accordance with the overall rating of places.
3. The user still may filter places by distance and category.

### 5.2.2.

The user receives information about location of places by means of their indication on the map.
**Actor:** user of the application

**Goal:** examine the location of a specific place on the map or find an interesting place based on own geographical preferences.

**Main success scenario:**

1. The user opens the map.
2.1. The user examines the location of a specific place in which they are interested.
2.2. The user examines the map, finds an interesting for them area and examines designated places within the area.

**Alternative scenario 1:**

1. The user realizes that they do not want to see all places on the map but only parks, for example.
2. The user specifies places which they want to visit using the filter (by distance, by category of places).

### 5.2.3.

The user looks for a place using the search with the function of filter.
**Actor:** user of the application

**Goal:** find a place using the search.

**Main success scenario:**

1. The user goes to the search page.
2. Using filtration by category and remoteness of places, the user receives a list of recommended for visit places.

**Alternative scenario 1:**

1. The user does not find an appropriate place.
2. The user makes a more general request.

### 5.2.4.

The user receives detailed information about a specific place such as its description, pictures, rating and reviews.
**Actor:** user of the application

**Goal:** learn details about some specific place.

**Main success scenario:**

1. The user presses at
    - a card of the place on the main page
    - the place which is designated on the map
    - a card of the place on the page of search results.
2. The user receives multiple pictures of the place as well as its rating and description.
3. The user selects the tab “reviews” and learns about impressions of other people about the place.

4. The user selects the tab “location” and the map is opened with the place marked.

### 5.2.5.

The user adds a place to collections.
**Actor:** user of the application

**Goal:** add a place to collections.

**Precondition:** the user is logged in.

**Main success scenario:**

1. The users presses at
   - a card of the place on the main page.
   - the place which is designated on the map.
   - a card of the place on the page of search results.
2. The user presses the button “add to collections”.
3. The place is added to collections, the user henceforth may open and see this bookmark on his profile page.

**Alternative scenario 1:**

1. The user presses at “add to collections” provided that this button is being already pressed.
2. A second activation of the button removes a place from collections.

### 5.2.6.

The user rates a place and leaves feedback about a place.
**Actor:** user of the application

**Goal:** rate a place and leave feedback.

**Precondition:** the user is logged in.

**Main success scenario:**

1. The user presses at
   - a card of the place on the main page
   - the place which is designated on the map
   - a card of the place on the page of search results.
2. The page about the place is opened, where the rating of the place is shown.
3. The user selects the tab “reviews”.
4. The user rates the place on a scale of 1 to 5.
5. The user writes a review of the place at will.
6. The user presses at “send a review”.
7. The review is saved and sent.

**Alternative scenario 1:**

1. The user presses the button “send a review”.
2. This operation cannot be done since the user did not write a review or rate the place.

3. The user either writes a review or rates the place.

### 5.2.7.

The user creates their own profile page.
**Actor:** user of the application

**Goal:** create the profile page.

**Main success scenario:**

1. The user presses at “creates an account”.
2. The user specifies their email.
3. The user thinks of and enters their password.
4. The user logs into their account.

**Alternative scenario 1:**

1. The user cannot log into the account, since they have forgotten their password.
2. The user resets the password by sending a mail to email.

### 5.2.8.

The user sets up their own profile page.
**Actor:** user of the application

**Goal:** set up the profile page.

**Precondition:** the user is logged in.

**Main success scenario:**

1. The user opens the profile page.
2. The user opens the profile settings.
3. The user sets up their own account, in particular, specifies their own preferences, uploads a profile picture and writes some information about themselves.
4. (Optional) The user adds or deletes friends at will (see 5.3.5, 5.3.6).
5. (Optional) The user may enable or disable transmitting their location to friends.

**Alternative scenario 1:**

1. The user does not want to set up the account for now.
2. The user may complete the set-up whenever they want to.

### 5.2.9.

The user opens their own profile page.
**Actor:** user of the application

**Goal:** use the profile page functionality.

**Precondition:** the user is logged in.

**Main success scenario:**

1. The user opens their profile page.
2. (Optional) The user may manage their list of friends: find and add new friends, delete them (see 5.3.5, 5.3.6).
3. The user may open and manage their collections.
4. The user may open the profile settings and set up the profile.

**Alternative scenario 1:**

1. The account was created recently and it is empty.
2. The user goes to the profile setting and sets up his profile page.

## 5.3. Optional Use-cases

### 5.3.1.

The user receives information about the location of friends on the map.
**Actor:** user of the application

**Goal:** have a look at the current location of friends in order to set up a meeting, for example.

**Precondition:** the user is logged in.

**Main success scenario:**

1. The user opens the map.
2. The user selects an option in the filter “track friends”.
3. The user receives information about the location of friends.

**Alternative scenario 1:**

1. The user does not see the location of some friends.
2. Friends have turned off transmitting their location.

### 5.3.2.

The user exchanges messages with friends.
**Actor:** user of the application

**Goal:** contact another user.

**Precondition:** the user is logged in.

**Main success scenario:**

1. The user opens the tab “chats” and selects a recipient.
2. The chat is opened.

**Alternative scenario 1:**

1. The user tries to contact another user.
2. This operation cannot be done since

- The recipient is not in the friendlist of the user. Besides this, the recipient has restricted receiving messages from unknown users.
- The recipient has added the user to the blacklist.

### 5.3.3.

The user adds another user to the blacklist.
**Actor:** user of the application

**Goal:** add another user to the blacklist.

**Precondition:** the user is logged in.

**Main success scenario:**

1. The user opens a profile of another user they want to block.
2. The user presses at “block a user”.
3. The user is blocked.

**Alternative scenario 1:**

1. The user tries to block another user.
2. This operation cannot be done since the user has already blocked another user.
3. Instead of blocking another user, the user may unblock them.

### 5.3.4.

The user deletes another user from the blacklist.
**Actor:** user of the application

**Goal:** delete another user from the blacklist.

**Precondition:** the user is logged in.

**Main success scenario:**

4. The user opens a profile of another user they want to unblock.
5. The user presses at “unblock a user”.
6. The user is unblocked.

**Alternative scenario 1:**

4. The user tries to unblock another user.
5. This operation cannot be done since another user has not been blocked.
6. Instead of unblocking another user, the user may block them.

### 5.3.5.

The user adds a friend.
**Actor:** user of the application

**Goal:** add a friend.

**Precondition:** the user is logged in.

**Main success scenario:**

1. The user opens the search page.
2. The user sets the filter “people”.
3. The user may enter the name of a necessary user.
4. The user finds a possible friend.
5. The user opens the profile page of another user.
6. The user presses at “add a friend”.
7. The user’s request is approved.
8. They are friends now.

**Alternative scenario 1:**

1. The user wants to add another user as a friend.
2. This operation cannot be done since another user is already a friend.
3. Instead of adding another user as a friend, the user may delete them from their friends list.

**Alternative scenario 2:**

4. The user wants to add another user as a friend.
5. Another user does not want to approve the request.
6. They will not be friends unless another user confirms the request.

### 5.3.6.

The user deletes a friend.
**Actor:** user of the application

**Goal:** delete a friend.

**Precondition:** the user is logged in.

**Main success scenario:**

1. The user opens their friends list.
2. The user finds a friend they want to remove.
3. The user opens the friends profile page.
4. The user presses at “delete a friend”.

**Alternative scenario 1:**

7. The user wants to remove a friend.
8. This operation cannot be done since another user is not a friend.
9. Instead of deleting another user as a friend, the user may add them to their friends list.

# 6. System-wide functional requirements

6.1 Searching by application content.
6.2 Background app updates.
6.3 Background map updates.
6.4 Displaying places on the map.
6.5 Sending notifications about app changes.

# 7. Non-functional requirements

## 7.1. Environment

- Programming languages:
  - C#
  - Kotlin
- Technologies:
  - Google maps API
  - PostgreSQL
  - SignalR
- Design:
  - Figma
  - XML

## 7.2. Performance

- The system should support at least 20 users concurrent connections.
- The response time to user’s requests, provided that they have a stable Internet connection, should not exceed 5 seconds.
- Application launch time should not exceed 5 seconds under ideal conditions.

## 7.3 Extensibility

- The Client-server communication is to be implemented according to the REST standard in order to make adding new functions simple to the extent possible.
- The modular design of the application allows to expand its functionality from the client's side.