# Annex 2: Software Requirement Specification

# Introduction

The following section provides a specification of the software’s overall requirements. Showing its functions, limitations and user interaction, the catalogue can be considered as a contract between clients and developers. Aside from giving textual descriptions, use case diagrams help to clarify the requirements and interactions in more detail.

# 2. Project Objectives

*(copy and adapt from part 2 – project objectives)*

# 3. Software Requirement Catalogue

This section outlines the software requirements. At first, the participating actors are introduced. Then, the requirements are examined, dividing them into functional and non-functional requirements.

## 3.1 Participating actors

This software has only one main actor, the user. Using the Telegram messenger, it interacts with the chatbot. There are no different user roles, so each user has the same range of features to use.

## 3.2 Functional Requirements

This section describes the software’s full range of features:

* Using the Telegram interface, messages to the chatbot can be introduced which are answered accordingly.
* Using the Telegram interface, a location can be introduced which is used as the basis for the chatbot recommendations.
* The proximity of the recommendations can be specified by the user. If no radius is entered, the default value of 1 km is used as a distance between the user location and examined point of interests.
* The chatbot provides a recommended point of interest within the chosen proximity. The recommendation result contains the name, location and type of point of interest as well as additional descriptions from OpenStreetMaps (if available).
* A desirable feature is the retrieval of additional data from external sources to enrich the POI descriptions. Information can be retrieved from Wikipedia to give the user a first impression or pictures from platforms.
* Recommended points of interests are rated by the users. The rating is saved and used as a basis for future recommendations.
* To refine recommendations, user chat messages are evaluated using natural language processing and saved in the user profile.

## 3.3. Non-functional Requirements

This section describe the so-called non-functional requirements which contain technical as well as operational requirements.

* To use the chatbot, the Telegram messenger app has to be installed on a smart device (smartphones or tablets). Although Telegram is also available as a desktop application, these versions do not support sending locations and are therefore not suitable.
* The device must be connected to the internet to use the chatbot.
* The device must be capable of receiving GPS information to calculate its current location.
* The chatbot interface should be intuitive, so the user is able to communicate with the chatbot without previously reading an exhaustive tutorial. To facilitate user decisions, mutually exclusive keyboard buttons are used.
* The chatbot should be able to handle user requests adequately. Questions and demands concerning travelling are understood and answered satisfyingly. Other requests are rejected politely.

## 3.4 Limitations

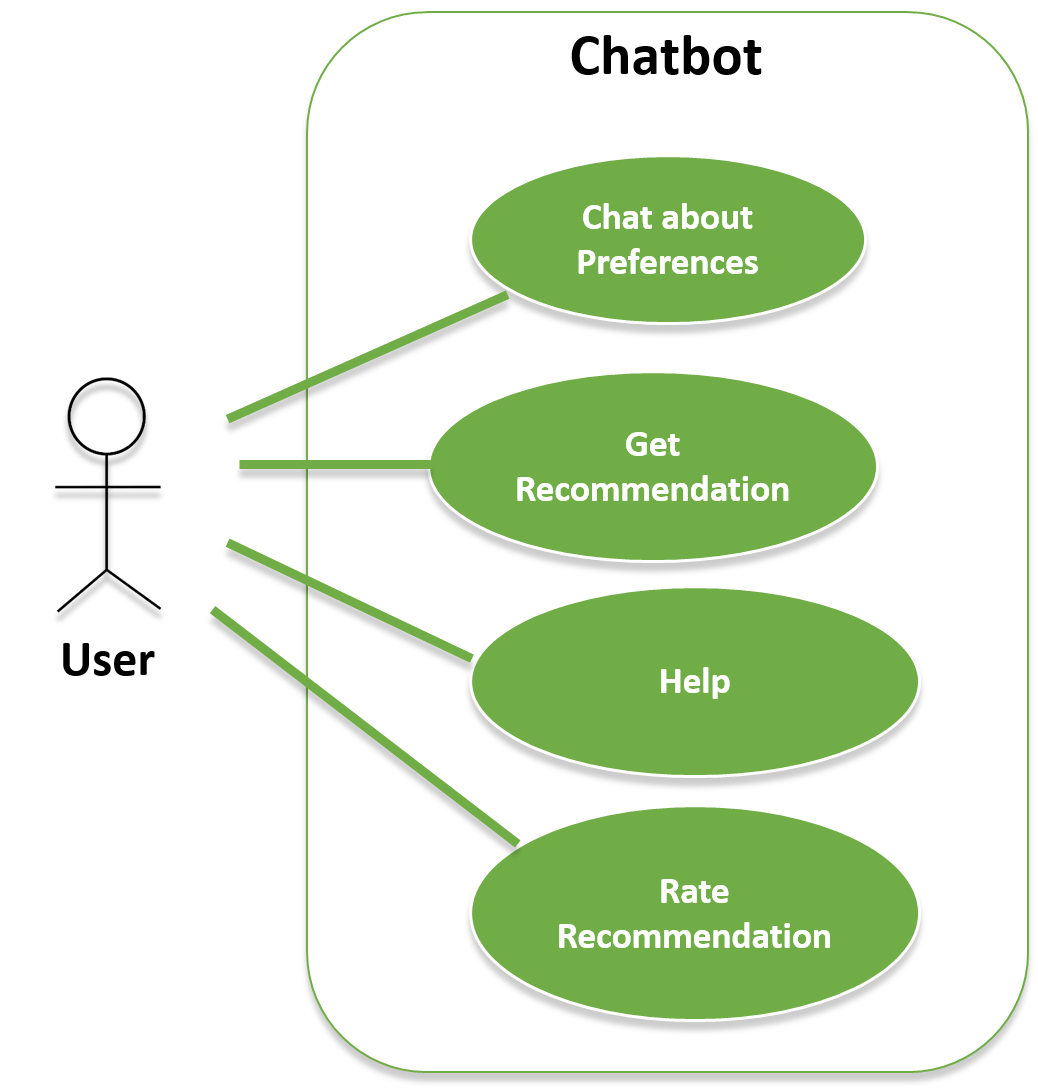
There are multiple limitations present due to the fact that the software can be still considered as a prototype. In a future enhancement, most of these limitations should be remedied.

* The used OpenStreetMaps data was downloaded once and then used offline. To keep the data up-to-date, an automatic update mechanism should be setup.
* Due to performance reasons and sparse user rating data, the prototype only gives recommendations for the city of *Barcelona, Spain*. In a future enhancement, a bigger OSM region should be covered.
* Due to the fact that a chatbot is available to large amount of users at the same time, it should be able to handle multiple user requests. This prototype is capable of handling requests from multiple users at the same time, but will have a delayed response. A desirable feature for a future enhancement would be to integrate concurrent access without significant time delay.

# 4. Requirement Specification

## 4.1 Use Case Diagrams

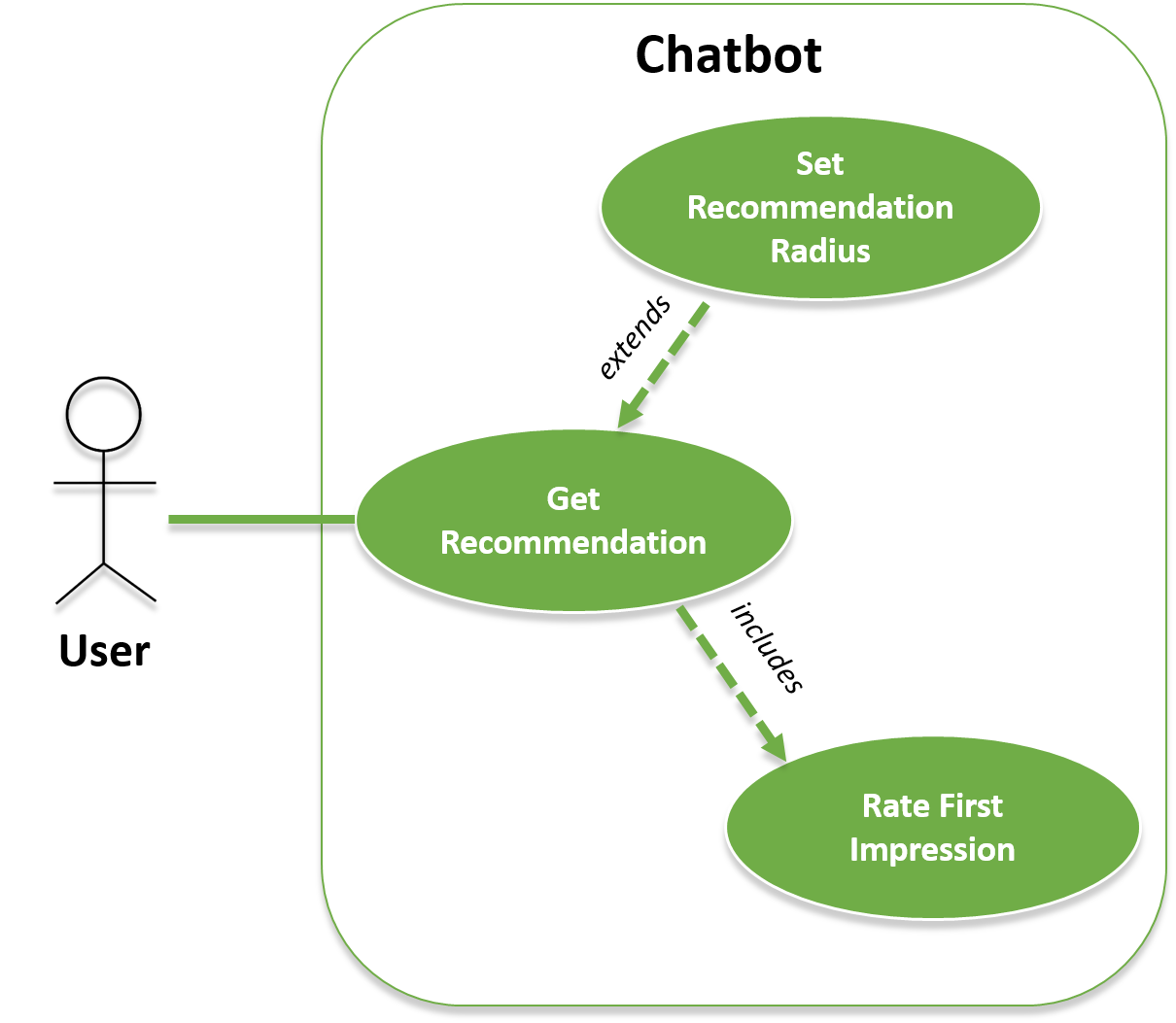
### 4.1.1 General Use Case Diagram



The shown use case diagram contains the four main interactions the user performs with the chatbot. The use case “Help” is used to show the user the main features of the chatbot. In “Chat about Preferences” the chatbot collects information about the user by chatting with him. This data is used to complete the user profile for recommendations. On the other hand, “Rate Recommendation” shows how the user rates previously recommended points of interests. The use c

The use case “Get Recommendation” represents the interactions between user and chatbot that lead to the provision of user adapted recommendations. Because of its complexity, this use case is shown in more detail in its own diagram next.

### Use Case Diagram – “Get Recommendation”



This use case shows the involved components in providing the user with recommendations. In order to get recommendations, the user is able to specify a recommendation radius to set the maximal distance between himself and the point of interest. This step is optional. After showing the user the recommended point of interest, he is asked to give a first impression of the point of interest in order to refine future recommendations.

## 4.2 Use Case Templates

The previously defined use cases are explained in detail in the following templates.

#### UC-01: Chat about Preferences

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | **UC-01** | |
| Use Case Name | Chat about Preferences | |
| Description | The user chats with the chatbot about his preferences | |
| Trigger | User greets chatbot and has no open ratings left or shows proactively intention to chat about his preferences | |
| Precondition | The chatbot is in a state in which the user is allowed to type messages independently. | |
| Flow of Events | Step | Action |
| 1 | Chatbot: *“I would like to get you know you better, is that ok?”* |
| 2 | User answers Yes |
|  | 3 | Chatbot asks question out of his question catalogue |
|  | 4 | User answers and interest is filtered |
|  | 5 | Steps 3-4 are repeated until all questions are asked or User quits question process |
| Alternate Flow | 4a | User answer is not understood, so the chatbot asks the user to rephrase the answer |
| Postcondition | The chatbot is in a state in which the user is allowed to type messages independently.  User Profile is updated with user interests. | |
| Exceptions | Step | Action |
| 2 | User answers No, so the use case is aborted |
| Frequency of Use | High | |
| Importance | Medium | |

#### UC-02: Get Recommendation

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | **UC-02** | |
| Use Case Name | Get Recommendation | |
| Description | The user asks for a recommended point of interest. Based on the user information the recommender returns points of interests close to the user. | |
| Trigger | The user types in a message that is interpreted as a “recommendation” intent by the language processing platform. | |
| Precondition | The chatbot is in a state in which the user is allowed to type messages independently. | |
|  | Step | Action |
| 1 | Chatbot: *“Where are you at the moment?”* |
| 2 | User enters his location after showing him the “Send location” button |
| Flow of Events | 3 | Chatbot: *“I’m going to look for places close to your current location. This could take a moment.”* |
|  | 4 | Chatbot presents user a recommended point of interest |
|  | 5 | Use Case -> Rate First Impression |
|  | 6 | Chatbot: *“Do you want to see another recommendation?”* |
|  | 7 | User enters No |
|  | 8 | Chatbot: *“Ok, have fun!”* |
| Alternate Flow | 1.a | User starts recommendation process by entering its current location which is followed by step 3 |
|  | 7.a | User enters Yes, so steps 4-6 are repeated. |
| Postcondition | The chatbot is in a state in which the user is allowed to type messages independently. | |
| Exceptions | Step | Action |
| 4 | Chatbot does not find any (more) points of interests for the user and cancels recommendation process with the message:  *“I couldn’t find any places for you. Maybe if I knew you better…?”* |
| Frequency of Use | Medium | |
| Importance | High | |

#### UC-03: Rate Recommendation

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | **UC-01** | |
| Use Case Name | Rate Recommendation | |
| Description | The user rates a points of interest that was previously recommended to him. | |
| Trigger | The user greets the chatbot or indicates that he wants to rate a recommendation. | |
| Precondition | 1. The chatbot is in a state in which the user is allowed to type messages independently. 2. The user was given a recommendation before that he has not rated yet. | |
| Flow of Events | Step | Action |
| 1 | Chatbot: “*How did you like (insert name of recommended POI)?*” |
| 2 | User chooses from mutually exclusive rating buttons (e.g. “Fantastic”, “Ok.”, “Didn’t like it”) |
|  | 3 | Chatbot: “Thanks for rating!” |
| Alternate Flow | - | - |
| Postcondition | 1. The rating is saved in the ratings file and the corresponding recommendation marked as rated. 2. The chatbot is in a state in which the user is allowed to type messages independently. | |
|  | Step | Action |
| Exceptions | - | - |
| Frequency of Use | Low | |
| Importance | High | |

#### UC-04: Specify Recommendation Radius

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | **UC-01** | |
| Use Case Name | Specify Recommendation Radius | |
| Description | The user sets the radius in which he wants the recommended points of interests to be in. | |
| Trigger | User enters messages which is identified as a distance intent by the natural language processing platform (e.g. *“I don’t want to walk that far”, “Let me set the distance”*…) | |
| Precondition | The chatbot is in a state in which the user is allowed to type messages independently. | |
| Flow of Events | Step | Action |
| 1 | Chatbot: *“How many meters are you willing to go to see recommended places at a max?”* |
| 2 | User answers with a positive numeric value |
|  | 3 | Chatbot repeats*: “Fine, I set the maximal radius to (repeat value)”.* |
| Alternate Flow | - | - |
| Postcondition | The maximal radius for that user is saved and used for the next recommendations.  The chatbot is in a state in which the user is allowed to type messages independently. | |
| Exceptions | Step | Action |
| 2 | User doesn’t answer with a positive numeric value, so the chatbot has to ask again or abort the use case |
| Frequency of Use | Low | |
| Importance | Low | |

#### UC-05: Rate First Impression

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | **UC-05** | |
| Use Case Name | Rate First Impression | |
| Description | When the user is given a recommended point of interest, he is immediately asked of his first impression to refine future recommendations. | |
| Trigger | - | |
| Precondition | The user has just received a recommended point of interest (UC-02: Get Recommendation) | |
| Flow of Events | Step | Action |
| 1 | Chatbot: “What do you think of this place?” |
| 2 | User chooses from mutually exclusive buttons: “Sounds good!” / “Don’t like it” |
| Alternate Flow | - | - |
| Postcondition | The first impression is added to the recommendation user ratings data. A good first impression is interpreted as a rating of value 4 in the user ratings table and “Don’t like it” is interpreted as a value 1 rating.  The recommended point of interest is added to the user’s open (= not rated) points of interests. | |
| Exceptions | Step | Action |
| - | - |
| Frequency of Use | Medium | |
| Importance | Medium | |

#### UC-06: Help

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | **UC-06** | |
| Use Case Name | Help | |
| Description | The user asks for help and gets an overview of the chatbot features | |
| Trigger |  | |
| Precondition | The chatbot is in a state in which the user is allowed to type messages independently. | |
| Flow of Events | Step | Action |
| 1 | The user types in a message that is interpreted as a “help” intent by the language processing platform. |
|  | 2 | Chatbot gives an overview of the chatbot features (Rating, chatting, recommendations) |
| Alternate Flow | - | - |
| Postcondition | The chatbot is in a state in which the user is allowed to type messages independently. | |
| Exceptions | Step | Action |
| - | - |
| Frequency of Use | Low | |
| Importance | Low | |