# 4. Tools & Technologies

In this section, the tools and technologies used during the course of the projects are presented. The different components of the geographic database are described as well as the tools to design the conversational interface. Additionally, some alternatives to the eventually used tools are discussed in order to explain why the specific tools are chosen.

In contrast, the tools used in the development process such as version control repository, integrated development environment or build-management tools are not explained in detail as they could be interchanged and are not essential for the application’s characteristics.

## 4.2. Chatbot platform

As already discussed in the previous chapter, there are several competing platforms that permit developers to build their own chatbot. Most of the major software big-players nowadays, such as Google, Facebook, Microsoft, are taking part in the development of these platforms and provide their own solutions, for example API.ai (Google), wit.ai (Facebook), luis.ai (Microsoft) or IBM Watson. Out of these contestants, api.ai and wit.ai seem to be the most widely known and easiest to use. Therefore, both options were investigated to see which one would be the most suitable for this project.

### 4.2.1 API.ai

API.ai 10 is a natural language processing platform by Google that facilitates creating conversational user interfaces. In order to model conversations, entities and intents are used as key concepts. API.ai provides a rich management toolset as well as a simply one-click-integration mechanism to import the chatbot into a variety of mobile apps. Also, the modeled conversation flow in api.ai can be accessed by submitting queries through the REST-like HTTP endpoints.

API.ai offers a variety of prebuilt agents that can be used as a base for the own conversational interface. On top of it, the conversational interface can be enabled to support *small talk*, allowing it to reply to basic, unspecific user input without any further development.

### 4.2.2 Wit.ai

Like api.ai, wit.ai is a natural language processing platform. It was acquired by Facebook in January 2015 and is free to use ever since then 11. Its key concepts for language processing include entities, intents and, additionally, stories. Stories help to create basic user scenarios in which typical user and chatbot conversations are designed­­­. Regarding integration, wit.ai provides a web service API to integrate the designed conversational interface in messaging apps.

### 4.2.3. Choice of Platform (api.ai)

After testing both platforms, both left a good impression and seem to be suitable to use. On the one hand, Wit.ai’s story feature seems promising to design conversations in an easy way, but is still in beta status. However, API.ai convinces with its rich management toolset, an extensive documentation and the integration of simple conversational features such as *small talk*

## 4.1 Geographic database

In order to extract tourist points of interests from *OpenStreetMap*, a geographic database was set up as the application’s backend. The following tools form either a part of the mentioned database or were used to set it up.

### OpenStreetMap

*OpenStreetMap5* (or short OSM)is a collaborative project with the aim to collect and update free to use geographic data. Its main purpose is to be a central data source which can be e.g. used for rendering maps. The stored data contains infrastructural information such as roads or buildings as well as variety of additional informational tags. In this project, the OpenStreetMap data is used to extract necessary tourist information in order to create user recommendations.

There is a big number of data dumps available that store the above mentioned data for either the whole planet or smaller regions or cities. These dumps can be downloaded in the file formats XML and PBF and imported into a PostgreSQL database.

### 4.1.1 PostgreSQL 9.5.5

*PostgresSQL*1 is an object-relational database system. It is an open source software that is most extensively conform to the SQL standard ANSI-SQL 2008 ­. In this project, PostgreSQL is used to store and manage the geographic database.

### 4.1.2 PostGIS 2.2.1

*PostGIS* 2 is an open source spatial extender for the object-relational database system *PostgreSQL*. The extension allows the database system to manage geographic objects and run spatial queries in *SQL*. Combined, *PostgreSQL* and *PostGIS* are used to create and manage spatial databases which in turn can be used in geographic information systems, such as *QGIS*.

### 4.1.3 pgAdmin III 1.22.0

*pgAdmin III* 5 is an open source tool to facilitate the management of *PostgreSQL* databases by providing a graphic user interface. Among many features, it comes with an SQL editor tool to create and run queries and displays data entries.

### 4.1.4 Osmosis 0.44.1

*Osmosis* 7 is an open source command-line based application that is able to process data from *OpenStreetMap.* In this project, it was principally used to import data from OSM files into the PostgreSQL database.

At first, the similar tool osm2psql was investigated and used in this project. However, during research it turned out that osm2psql is not the right fit because its main objective is the import of .osm files for rendering purposes. Due to this reason, only data that are render relevant are imported into the PostgresSQL database. On the other hand, Osmosis imports all of the raw .osm data, namely Nodes, Ways, Relations and their corresponding tags 8.

## Messenger - Telegram

*Telegram*[x] is an instant messaging app for smartphones, tablets or computers. There are available versions for iOS, Android, Windows Phone, as well as desktop applications for Windows, OSX and Linux. Telegram concentrates on speed and security of its messages.

In June 2015, Telegram introduced its Bot API, allowing third-party developers to integrate their own bots into the messenger. The bots are controlled sending HTTPS request. In this project, incoming updates from Telegram are received via an outgoing webhook. [x1]

## 4.4 Platform as a Service - Heroku

Heroku (https://www.heroku.com/platform)is a platform-as-a-service that enables users to run their applications in the cloud. It supports several programming languages, among them Node, Ruby and Java. There are several pricing models offering a range of different features. In this case, the free plan is used which comes with the inconvenience that the application sleeps after 30 minutes of inactivity, leading to a short delay every time the chatbot is accessed after not using it.

Likewise, *Heroku Postgres* is a database-as-a-service enabling the upload of the project’s database into the cloud.

## 4.3 API and Libraries

### Foursquare

*Foursquare* is a mobile service application which provides personalized recommendations of places for its users. In this project, Foursquare’s RESTful API is used to access additional information OpenStreetMap does not provide, more precisely extracting photos for points of interests. [https://developer.foursquare.com/docs/]

### Apache Mahout

Apache Mahout is an open-source project which provides a set of machine-learning algorithms for classification, clustering and recommendation. Mahout plays a vital part in the project’s development as Mahout’s Java library forms the base for the application’s recommender. The recommender’s collaborative filtering approach is based on Mahout’s simple user-based recommender (<https://mahout.apache.org/users/recommender/userbased-5-minutes.html>) whereas some of Mahout’s interfaces where adapted to create a content-based mechanism that fit this project’s needs (more precisely in Relevant Aspects and Annex C – Design Specification).

### Spark

Spark is a micro framework that is used to create Java web applications. Its most outstanding feature is its lightweight design which enables the setup of a web application with minimal overhead relying on Java 8 language features. Spark’s main component is a set of routes which defines how the application is accessed by clients. A route consists of a HTTP verb, a path and a callback. In this project, a POST route is set up using Spark which receives the incoming messages from Telegram.

# Bibliography

1: <https://www.postgresql.org/about/>, 02.02.17

2: http://postgis.net/

5: https://www.pgadmin.org/

7: <https://wiki.openstreetmap.org/wiki/Osmosis>

8: <https://me4bruno.wordpress.com/2012/03/25/osmosis-osm2postgresql-osm2pgsql-openstreetmap-daten-datenbanken-und-spielplatze/>

10:

[https://docs.api.ai/docs/welcome](https://docs.api.ai/docs/welcome 11)

[11](https://docs.api.ai/docs/welcome 11): https://wit.ai/blog/2015/01/05/wit-ai-facebook

12: <https://wit.ai/docs/quickstart>

5:https://www.openstreetmap.org/about

X: <https://telegram.org/faq>

X1: https://core.telegram.org/bots/api#getting-updates