**Abstract**

A group of developers are still forced to take their cars in order to reach the workplace. Thus, they decided to develop a simple application that will allow them to save some money. The idea is a crowdsourcing that permits users both to upload the prices of the gas stations and read the prices uploaded by other users. In order to make the application simple to use, it should contain the map with all the available gas stations. Moreover, users should be able to find the gas station that better satisfies their needs, such as position, maximum distance and price.

**Stakeholders**

|  |  |
| --- | --- |
| **Stakeholder** | **Description** |
| User | Upload and read prices, suggest new gas stations, report bugs (to developers) and vandalisms (to moderators) |
| Moderator | Supervise the behaviour of the users in order to prevent vandalisms, validate new stations |
| Administrator | Involved in database management |
| Developer | Develop the application and maintains it by fixing bugs with the help of user reports. |
| Gas station owner | Does not interact directly with the application, but he chooses the prices that will be uploaded by the users. |
| OpenStreetMap | Open database that provides map information |

**Stations database**

**Context diagram**

**User**

**Administrator**

**Moderator**

**OpenStreetMap**

**Interfaces**

|  |  |  |
| --- | --- | --- |
| **Actor** | **Logical Interface** | **Physical Interface** |
| User | GUI | Screen, mouse and keyboard or touch-screen |
| Administrator | CLI | Screen and keyboard |
| Moderator | GUI | Screen, mouse and keyboard |
| OpenStreetMap | Mapsforge API | Internet connection |
| Stations database | SQL | Internet connection |

**Functional requirements**

1. Users must be able to find all the open gas stations that satisfy a set of user-defined requirements, i. e., distance from a given position and price.
   1. It must be possible to select the position either by entering an address or by retrieving the current position by GPS (if available)
   2. It must be possible to select the prices for a user-specified fuel type
2. Users must be able to share the information about prices of gas stations
3. Users must be able to suggest a new gas station
4. Users must be able to report stations that no longer exist, vandalisms (such as wrong prices) and bugs
5. Moderators must be able to validate and insert and remove gas stations
6. Moderators must be able to revert changes done by users
7. Moderators must be able to ban users that do not keep a correct behaviour
8. Administrator must be able to manage users and moderators

**Non-functional requirements**

1. Usability Application should be used with no training by users
2. Performance User functions should complete in less than 2 seconds
3. Portability Application runs on desktop browser and Android

**Stories**

Gianfranco has to do shopping. But is car is near to be out of fuel. Unfortunately the gas station he was used to refuel at is closed today. But he can use this fancy app, that can find a gas station on the road to the supermarket for him.

**Use case diagram**

UC1

**User**

**Stations database**

**OpenStreetMap**

UC2

**User**

**Stations database**

UC3

**Moderator**

**User**

**Stations database**

UC4

**User**

**User**

**Moderator**

**Stations database**

**Use cases**

UC1

|  |  |
| --- | --- |
| **Actors Involved** | User U, Stations database DB, OpenStreetMap OSM |
| **Precondition** | Station S exists in DB |
| **Post-condition** | List of stations is shown in OSM |
| **Nominal Scenario** | U specify the search parameters, the stations that satisfy the parameters are searched in the DB and shown in OSM |
| **Variants** | No station satisfies the parameters |

UC2

|  |  |
| --- | --- |
| **Actors Involved** | User U, Stations database DB |
| **Precondition** | Station S exists in DB |
| **Post-condition** | Prices referred to S in DB are updated |
| **Nominal Scenario** | U uploads new prices, the entry relative to S is updated in DB. |
| **Variants** |  |

UC3

|  |  |
| --- | --- |
| **Actors Involved** | User U, Moderator M, Stations database DB |
| **Precondition** | Station S do not exists in DB |
| **Post-condition** | Station S exists in DB |
| **Nominal Scenario** | U suggest a new station S to M  M add S to DB |
| **Variants** | Station S do not really exists and in it not added to DB by M |

UC4

|  |  |
| --- | --- |
| **Actors Involved** | User U1, User U2, Moderator M, Stations database DB |
| **Precondition** | U1 found in DB wrong prices uploaded intentionally by U2 |
| **Post-condition** | Wrong prices are removed, U2 get banned from the system |
| **Nominal Scenario** | U1 report U2 to the M  M revert the changes done by U2  M ban U2 |
| **Variants** | U2 uploaded correct prices and the report of U1 was wrong. |

**Scenarios**

SC1 – UC1

|  |  |
| --- | --- |
| **Description** | Find a list of stations that satisfy a set of parameters |
| **Precondition** | Station S exists in DB |
| **Post-condition** | List of stations is shown in OSM |
| **Step 1** | User specify the search parameters, and shown in OSM |
| **Step 2** | An SQL query is done to retrieve the list of gas stations |
| **Step 2** | The stations are shown in the map as markers |