# Official Requirements Document

Authors: Paolo D’Ambrosio, Lorenzo Lanari, Silvio Girolami, Stefano Cataldi

Date: 28/03/2020

Version: 1

Contents

[Official Requirements Document 1](#_Toc36318445)

[Abstract 1](#_Toc36318446)

[Stakeholders 2](#_Toc36318447)

[Context Diagram and interfaces 2](#_Toc36318448)

[Context Diagram 2](#_Toc36318449)

[Interfaces 3](#_Toc36318450)

[Stories and personas 3](#_Toc36318451)

[Functional and non functional requirements 4](#_Toc36318452)

[Functional Requirements 4](#_Toc36318453)

[Non Functional Requirements 4](#_Toc36318454)

[Use case diagram and use cases 4](#_Toc36318455)

[Use case diagram 5](#_Toc36318456)

[Use Cases 5](#_Toc36318457)

[Use case 1, UC1 – FR1 Insert a new gas station 5](#_Toc36318458)

[Use case 2, UC2 – FR2 Delete a gas station 5](#_Toc36318459)

[Use case 3, UC3 – FR3 Insert/change fuel price 6](#_Toc36318460)

[Use case 4, UC4 – FR4 Leave a feedback 6](#_Toc36318461)

[Use case 5, UC5 – FR5 Compute Route 6](#_Toc36318462)

[Use case 6, UC6 – FR6 Add filters 7](#_Toc36318463)

[Use case 7, UC7 – FR7 Compare fuels price 7](#_Toc36318464)

[Use case 8, UC8 – FR8 Check inserted information 7](#_Toc36318465)

[Relevant scenarios 8](#_Toc36318466)

[Scenario 1 8](#_Toc36318467)

[Scenario 2 8](#_Toc36318468)

# 

# Abstract

A crowdsourcing application that shows gas stations in an area, along with the prices they practice, has to be designed.

The application is meant to be used by everyone who wants to know these information. These informations are inserted and updated by users themselves.   
The application home shows a map of gas stations around user location and a list of all stations that can be sorted by price, alphabetic order or distance. Other filters may be applied (fuel type, fuel company, etc.).

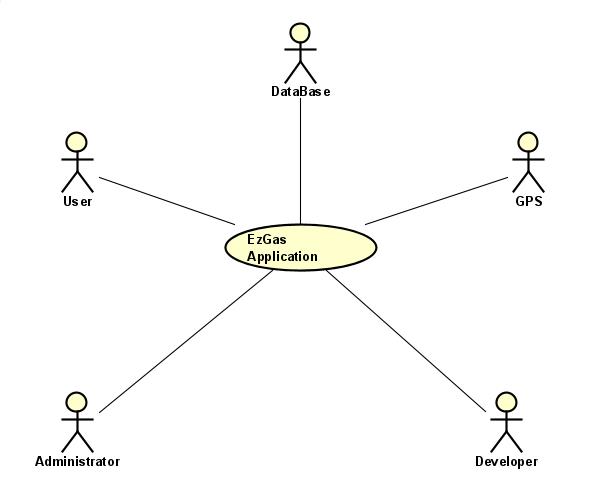
Users can insert also a feedback about the gas station and those feedbacks are readable by other users.

# Stakeholders

|  |  |
| --- | --- |
| **Stakeholder name** | **Description** |
| User | Uses the application to read and insert information |
| GPS | Gives the real time position users and positions where all stations are located |
| Administrator | Checks that all information are correct |
| Developer | Creates the application |
| Application Database | Where all information are stored |

# Context Diagram and interfaces

## Context Diagram



## Interfaces

|  |  |  |
| --- | --- | --- |
| **Actor** | **Logical Interface** | **Physical Interface** |
| User | GUI | Smartphones |
| GPS | APIs | Internet connection |
| Administrator | GUI | Smartphones, screen, keyboard |
| Developer | GUI | Screen, keyboard |
| Database | GUI and APIs | Smartphones |

# Stories and personas

Arturo is a truck driver and is constantly on the road in order to carry goods all over the Europe.  
He used to contact his truck driver friends to receive advices about the cheapest gas station,  
near him. Despite being a truck driver for many years, it is not always easy for him to understand  
 the advices given by his colleagues, in particular the ones about the location of the various   
gas stations. Another problem for Arturo is that most of his friends use a different type of  
fuel, so it is a mess to filter all the incoming information.  
He would really like to have the possibility to use the gps inside his phone  
to reach his destination and moreover to be able to search specifically for GPL stations.

Jennifer is a 37 years old mother of four children. Every day she has to pick up her children from school  
and it's difficult for her to choose the best route in order to be fast in cooking lunch.  
Sometimes, in all this rush, she needs to go to a gas station so it's necessary for her  
to get to the nearest gas station to the school of her children.

# Functional and non functional requirements

## Functional Requirements

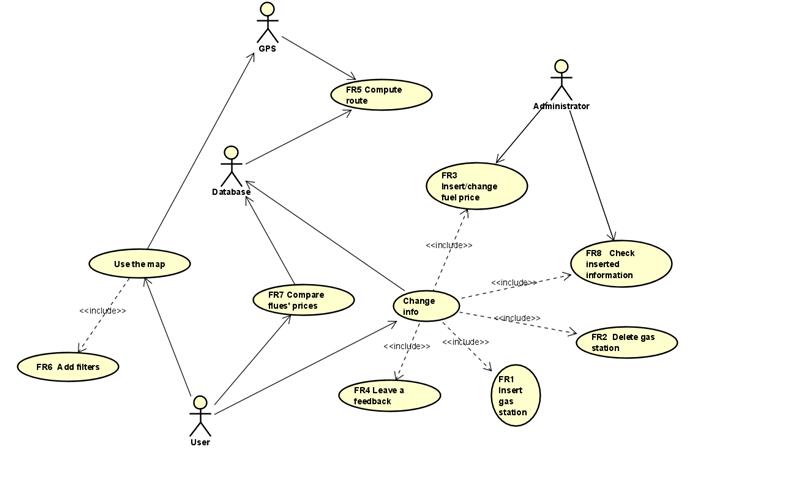
|  |  |
| --- | --- |
| **ID** | **Description** |
| FR1 | Insert a new gas station |
| FR2 | Delete a gas station |
| FR3 | Insert/change price |
| FR4 | Leave a feedback |
| FR5 | Compute the route to the station |
| FR6 | Add filters |
| FR7 | Compare fuels and prices |
| FR8 | Check inserted information |

## Non Functional Requirements

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Type** | **Description** | **Refers to** |
| NFR1 | Usability | Application should be used easily while in a car | All FR |
| NFR2 | Performance | All functions should complete in < 0.5 sec | All FR |
| NFR3 | Portability | The application should run on smartphones | All FR |
| NFR4 | Reliability | The application should be available 99% of times |  |

# Use case diagram and use cases

## Use case diagram



## Use Cases

### Use case 1, UC1 – FR1 Insert a new gas station

|  |  |
| --- | --- |
| **Actors involved** | User, Database |
| **Precondition** | User U exists, database D exists, station S doesn’t exist |
| **Post condition** | S exists, S.fuel > 0 |
| **Nominal scenario** | User inserts a new name and a new price, database adds a new entry |
| **Variants** | Station already exists, issue warning |

### Use case 2, UC2 – FR2 Delete a gas station

|  |  |
| --- | --- |
| **Actors involved** | User, Database |
| **Precondition** | Station S exists |
| **Post condition** | Station S doesn’t exist |
| **Nominal scenario** | User selects station S and delete it |
| **Variants** |  |

### Use case 3, UC3 – FR3 Insert/change fuel price

|  |  |
| --- | --- |
| **Actors involved** | Users, Database, Administrator |
| **Precondition** | Station S exists |
| **Post condition** | S.price > 0 |
| **Nominal scenario** | User selects Station S, Price is inserted or changed |
| **Variants** |  |

### Use case 4, UC4 – FR4 Leave a feedback

|  |  |
| --- | --- |
| **Actors involved** | User, Database |
| **Precondition** | Station S exists |
| **Post condition** | S.feedback.quantity\_pre < S.feedback.quantity\_post |
| **Nominal scenario** | User selects a station S and adds a feedback |
| **Variants** |  |

### Use case 5, UC5 – FR5 Compute Route

|  |  |
| --- | --- |
| **Actors involved** | GPS, Database |
| **Precondition** | Station S exists, User.position is not NULL |
| **Post condition** |  |
| **Nominal scenario** | The application sets the route to the station S |
| **Variants** |  |

### Use case 6, UC6 – FR6 Add filters

|  |  |
| --- | --- |
| **Actors involved** | User, GPS |
| **Precondition** | Filter F exists |
| **Post condition** |  |
| **Nominal scenario** | The user selects a filter |
| **Variants** |  |

### Use case 7, UC7 – FR7 Compare fuels price

|  |  |
| --- | --- |
| **Actors involved** | User, Database |
| **Precondition** | Fuel Fu exists, Fu.price > 0, Station S exists |
| **Post condition** |  |
| **Nominal scenario** | User selects stations or a fuel to compare them |
| **Variants** | Stations doesn’t exist, issue warning |

### Use case 8, UC8 – FR8 Check inserted information

|  |  |
| --- | --- |
| **Actors involved** | Administrator |
| **Precondition** | Station S exists, S.price > 0 |
| **Post condition** | All information are correct |
| **Nominal scenario** | The administrator analyze a Station and its properties |
| **Variants** |  |

# Relevant scenarios

## Scenario 1

|  |  |
| --- | --- |
| Scenario ID:SC1 | Corresponds to UC 1 |
| Description | User finds a Gas Station not inserted in the app yet and insert it. |
| Precondition | User is already signed in. |
| Postcondition | Database is updated with the new station |
| Step# | Step description |
| 1 | User insert his credentials to sign in |
| 2 | User insert data about the new station |
| 3 | Administrator checks the data inserted |
| 4 | The new station is inserted in the database |

## Scenario 2

|  |  |
| --- | --- |
| Scenario ID:SC2 | Corresponds to UC 1 |
| Description | User finds a Gas Station not inserted in the app yet and insert it, with wrong data. |
| Precondition | User is already signed in. |
| Postcondition | Administrator block the insertion on the new station |
| Step# | Step description |
| 1 | User insert his credentials to sign in |
| 2 | User insert wrong data about the new station |
| 3 | Administrator checks the data inserted |
| 4 | Administrator blocks the insertion request |

# Glossary

In this section we describe the principal concepts of the EZgas application through a UML class diagram.

# 

EZgas is a crowdsourcing application that allows users to insert and check information about gas stations. Each user can sign up using his name and surname in order to create an account with an email, a userID and a password to log in. The application collects several gas stations: a gas station is characterized by a name and a position on the map, indicated by 2 coordinates (latitude and longitude). Furthermore, a gas station can offer to his customers several type of fuel: a fuel has a name and a price. Each account can leave a feedback to a gas station: a feedback is composed of a comment and a rating (going from 0 to 5) and is identified by the date it was written and the userID of the user who wrote it. The Map API is used to show a map of the user area with the nearest gas stations.