**EZgas official requirement document**

Authors: Grottesi Lorenzo

Date: 28/03/2020

Version: 2

Change history:

Version 2: added map interface, added glossary, changed non functional requirements, updated preconditions, shifted up functional and non functional requirements..

**Abstract**

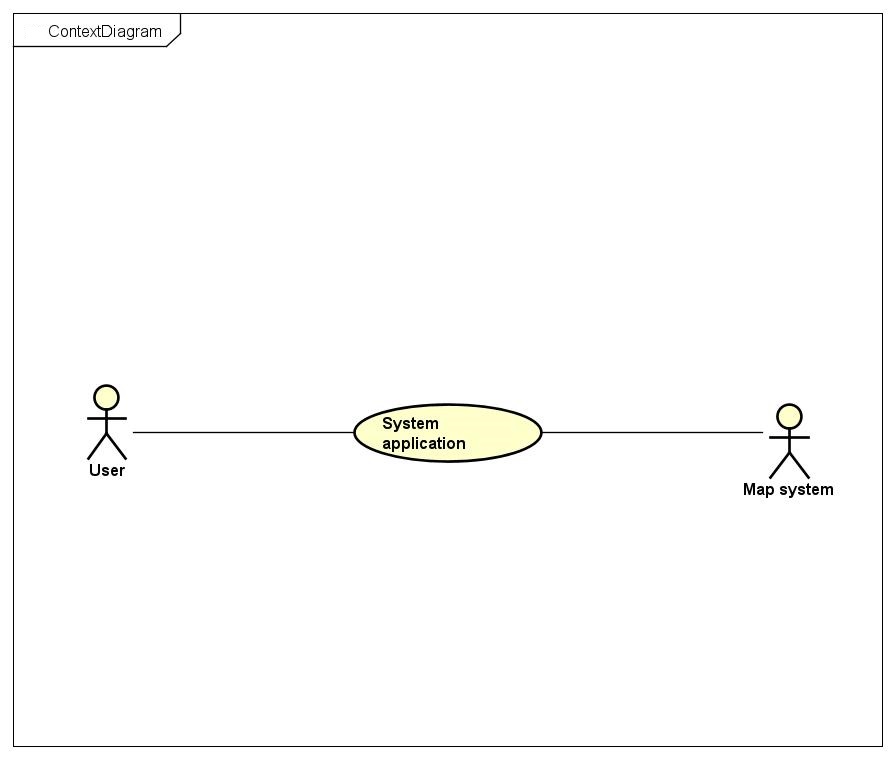
EZGas is a crowdsourcing service used by the users to classify the gas stations. The application takes into account the price and the distance of the stations, each user is able give its own contribution to the community inserting or updating the prices and location of gas stations located in a certain area.

# Stakeholders

The stakeholders

* Administrator: validate the user’s profile.
* Developer: fix software bugs.
* Map API: external map system service.
* User: use the system to find the gas station and classify it.

# Context Diagram



**Interfaces**

|  |  |  |
| --- | --- | --- |
| Actor | Logical interface | Physical interface |
| User | GUI | Screen |
| Map API | Web interface | - |

# Stories and personas

Mario is 27 years old and is an engineer of an important industry where his job is particularly dynamic, indeed often he has to travel by car from one city to another. The company gives to Mario all the money that he spends when traveling among cities, but Mario is particular careful to save the company’s money and he knows that one of the most important cost when travelling is the fuel. Each time Mario goes in a new city he uses the application EZGas application to find the cheapest station closer to him, sometimes happened that in the new city there aren’t gas station recorded to the application so Mario decide to go to the closest station and add it to the application map.

# Functional requirements

|  |  |
| --- | --- |
| ID | Description |
| FR1 | Store a new gas station in the system |
| FR2 | Update attributes of an old gas station |
| FR3 | Show map of gas stations |
| FR4 | Show prices of a selected gas station |
| FR5 | Show the list of gas station in a given range |
| FR6 | Show the list of gas station of given prices |
| FR7 | Login |
| FR8 | Logout |
| FR9 | Register a new user |

# Nonfunctional requirements

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Type | Description | Refer to |
| NF1 | Usability | Application should be able to locate the user | All FR |
| NF2 | Performance | All the operation should complete in <1 s | All FR |
| NF2 | Portability | The application should run for all android environment | All FR |

# Scenarios and use case

**Use case 1: UC1 - FR1 Insert new gas station**

Actors involved: user, map system.

Precondition: Gas station G not registered; user U is registered.

Postcondition: G is registered with position and prices.

Nominal scenario: The user U insert the price, position and name of the gas station.

**Use case 2: UC2 - FR2 Update gas station price**

Actors involved: user.

Precondition: Gas station G registered; user U is registered.

Postcondition: New G.prices.

Nominal scenario: The user U insert the new prices of the selected G gas station.

**Use case 3: UC3 – FR3 Show the gas station map**

Actors involved: user, map system.

Precondition: user U is registered.

Postcondition:

Nominal scenario: User U is able to see in the map its current position and the gas stations.

**Use case 4: UC4 – FR4 Show gas station prices**

Actors involved: user, map system.

Precondition: user U is registered, station G is registered.

Postcondition: U see the G.prices.

Nominal scenario: User U ask for the G.prices selecting G in the map M.

**Use case 5: UC5 – FR5 Show gas station in a range**

Actors involved: user, map system.

Precondition: user U is registered.

Postcondition: List of gas station.

Nominal scenario: User U ask insert a range R of km and the system map show if there is at least one gas station G in that range.

**Use case 6: UC6 – FR6 Show gas station with prices under a certain threshold**

Actors involved: user, map system.

Precondition: user U is registered.

Postcondition: List of gas station.

Nominal scenario: User U ask insert a threshold T of prices and the system map show if there is at least one gas station G that that respect that threshold, if more than one the list is sorted by distance.

**Scenario 1**

Scenario ID: SC1 – UC 1

Description: User insert a new gas station.

Precondition: The user is correctly registered.

Postcondition: The new gas station is stored at the user position with the user inserted prices.

|  |  |
| --- | --- |
| **Step** | **Step description** |
| 1 | Start evaluation |
| 2 | Insert gas station name |
| 3 | Insert gas station prices |
| 4 | Insert gas station position |
| 5 | commit |

**Scenario 2**

Scenario ID: SC2 – UC4

Description: Update gas station prices

Precondition: The station is yet in the system; the user is registered and is physically in the station.

Postcondition: The prices of the gas station are updated.

|  |  |
| --- | --- |
| **Step** | **Step description** |
| 1 | Start evaluation |
| 2 | Select the station |
| 3 | Insert new prices |
| 4 | Commit |

**Scenario 3**

Scenario ID: SC3 – UC 4

Description: Show selected gas station prices.

Precondition: The user is correctly registered.

Postcondition: The list of prices of selected station is showed.

|  |  |
| --- | --- |
| **Step** | **Step description** |
| 1 | Open map |
| 2 | Select gas station |

**Scenario 4**

Scenario ID: SC4 – UC 6

Description: Show gas station which prices are under a given threshold.

Precondition: The user is correctly registered.

Postcondition: The list station is showed.

|  |  |
| --- | --- |
| **Step** | **Step description** |
| 1 | Open map |
| 2 | Select function “filter” |
| 3 | Select threshold |
| 4 | Insert threshold value |

# Use case diagram

# 

# Glossary

