Contents

[1. Abstract 2](#_Toc37431846)

[2. Stakeholders 2](#_Toc37431847)

[3. Context diagram and interfaces 2](#_Toc37431848)

[4. Stories and personas 2](#_Toc37431849)

[5. Functional requirements 3](#_Toc37431850)

[6. Nonfunctional requirements 3](#_Toc37431851)

[7. Use cases 3](#_Toc37431852)

[8. Use case diagram 5](#_Toc37431853)

[9. Relevant scenarios 5](#_Toc37431854)

[10. Glossary 7](#_Toc37431855)

# Abstract

Most individuals prefer using their own cars instead of public transport. Even car prices are getting more and more affordable, the increasing fuel prices force people drive their cars more efficiently. Changing driving habits may help to reduce the fuel consumption. An additional solution could be to minimize gas expenses by filling the tank from the cheapest nearby gas station.

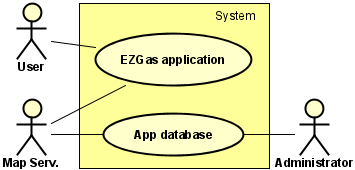
EZGas is a crowdsourcing application. It shows fuel prices at nearby gas stations to the users. Instead of establishing a direct communication with gas station holders, the application allows users to insert new prices to the application database. The other users are notified when an update request is sent to the application maintainers. In this way, the other users can check whether the update request is correct or not. The application is a simple, yet useful tool that helps the users to select the best option to fill their cars’ gas tank.

# Stakeholders

**User:** Car owner who will use the application to find the best nearby gas station. User can be registered or unregistered. A registered user can also involve activities to update the prices in database.

**Administrator:** All the price information must be stored in a database. It must serve continuously. The database administrator manages database and fix any problem related to it as soon as possible.

# Context diagram and interfaces

In the context diagram, “System” represents EZGas application itself. The whole system consists of the application GUI for local devices (smart phone) and the database to store all the information. The application itself does not store any data on the local device. All the user preferences and other sensitive data is stored on a remote server. Thus, the application GUI requires internet connection.

Context diagram

|  |  |  |
| --- | --- | --- |
| **Actor** | **Logical interface** | **Physical interface** |
| User | GUI | Touch screen |
| Administrator | Database computer | Ethernet wires and other devices |
| Mapping service | APIs | Smartphone |

Interfaces

# Stories and personas

Carl works in a company as a part time employee and he serves as a cab driver with his car in his spare time. With another application, he checks whether someone needs a taxi and utilizes his car as a taxi to gain extra money in his spare time. He usually uses his car in urban traffic. Thus, he loses a lot of time while waiting in traffic and this causes a considerable increase in fuel consumption. He wants to increase his benefit by paying as much less as possible for gasoline while serving as a cab driver. The best way is using a simple yet easy to use application to find the nearby station which offers the lowest price.

Carl also strictly relies on the power of communities. He believes he can make a positive impact on others’ life by helping them to reduce their fuel expenses. For this reason, he also wants to share his experiences and fuel prices in different gas stations. In this way, more people might reach relatively cheap gasoline and the other stations would decrease their prices. As a consequence, he could also make a positive impact also on income injustice.

He uses this application not only for his own benefit but also to help to the others for reducing their expenses and he loves this marvelous application as it was his baby. Hence, he would love to involve any activity to improve the application. He is keen on to send feedback about his experiences on using the application and to improve the functionality of the application.

Anyone is free to install the application and become a member of this life changing community. An intruder may also try to be a member of it. The intruder may attempt to manipulate the prices in order to fool the precious members of this great community and canalize them to prefer wrong gas stations. Hence, Carl would also want to take responsibility about price update requests. On the other hand, even he believes equivalence in all part of his life, he always considers that there might be some irresponsible people. It is good to keep the application anonymous but some people must have a little bit more authority to keep the application useful for the community members. For this reason, he thinks it is better to be a registered user to take supervisor role and confirm or reject any price update request.

# Functional requirements

|  |  |
| --- | --- |
| **FR ID** | **Functional requirement description** |
| FR1 | Registered user sends an update request |
| FR2 | Filter fuel type |
| FR3 | Filter gas station |
| FR4 | Show other services (w.c, market etc.) |
| FR5 | Registered user rates the station |
| FR6 | Registered user confirms or rejects any update request |
| FR7 | Registered user sends request to add new station |
| FR8 | Remove user |
| FR9 | Block user |

# Nonfunctional requirements

|  |  |  |
| --- | --- | --- |
| **Type** | **NFR ID** | **Nonfunctional requirement description** |
| Fidelity | NFR1 | User can obtain virtual ranks based on application usage frequency |
| Size | NFR2 | Application should be able to be downloaded and installed in a minute |
| Simplicity | NFR3 | GUI should have a few buttons |
| Support | NFR4 | Application should serve in different languages |
| Usability | NFR5 | Application should be used without any training |
| Portability | NFR6 | Application should run on Android and iOS OSes |
| Efficiency | NFR7 | GUI should be simple in order to reduce CPU utilization |
| Maintainability | NFR8 | It must be able to serve during database update |

# Use cases

|  |  |
| --- | --- |
| Use case 1, UC1 – FR1 Send update request | |
| Actors involved | Registered |
| Pre condition |  |
| Post condition | Registered user sends an update request |
| Nominal scenario | Registered user sends any kind of update request |
| Variants | User sends a request to update a price |
|  | User sends a request to update station status (open/closed) |
|  | User sends a request to update additional services |

|  |  |
| --- | --- |
| Use case 2, UC2 – FR2 Filter fuel type | |
| Actors involved | User |
| Pre condition |  |
| Post condition | Stations, which provide the filtered fuel type, are shown on the map |
| Nominal scenario | User opens menu and selects a fuel type |
| Variants |  |

|  |  |
| --- | --- |
| Use case 3, UC3 – FR3 Filter stations | |
| Actors involved | User |
| Pre condition |  |
| Post condition | Stations, belonging to the selected brand are shown on the map |
| Nominal scenario | User opens menu and selects a brand |
| Variants |  |

|  |  |
| --- | --- |
| Use case 4, UC4 – FR4 Show additional services | |
| Actors involved | User |
| Pre condition | A station is selected on the map |
| Post condition | Provided services are shown |
| Nominal scenario | User taps on a station on the map |
| Variants |  |

|  |  |
| --- | --- |
| Use case 5, UC5 – FR5 User rates the station | |
| Actors involved | User |
| Pre condition | User must be logged in |
| Post condition | User rates the station |
| Nominal scenario | User selects a station on the map and rates it |
| Variants |  |

|  |  |
| --- | --- |
| Use case 6, UC6 – FR6 User rejects or confirms a request | |
| Actors involved | Registered user |
| Pre condition | An update request must be sent |
| Post condition | The request is uprated or down rated |
| Nominal scenario | Registered user sees the request and confirms o rejects it |
| Variants |  |

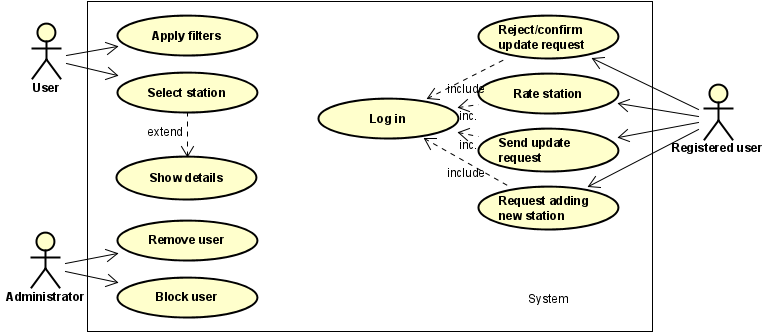
|  |  |
| --- | --- |
| Use case 7, UC7 – FR7 User adds a new station | |
| Actors involved | Registered user |
| Pre condition |  |
| Post condition | Administrator receives request to add a new station |
| Nominal scenario | Registered user sends a request to remove the station |
| Variants |  |

|  |  |
| --- | --- |
| Use case 8, UC8 – FR8 Remove user | |
| Actors involved | Administrator |
| Pre condition | Administrator receives account remove request |
| Post condition | User is completely removed from the database |
| Nominal scenario | Registered user sends a request to remove his/her account |
| Variants |  |

|  |  |
| --- | --- |
| Use case 9, UC9 – FR9 Block user | |
| Actors involved | App database |
| Pre condition | Registered user has very low fidelity |
| Post condition | Registration authority is suspended for 1 week |
| Nominal scenario | Registered user continuously sends unreliable update requests |
| Variants |  |

# Use case diagram

The application is simple. In the nominal case the user applies the filters as he wants and sends a request to the database to see the nearby gas stations. The request is sent through the application user interface. The database applies the request filters and returns results to the application.

Some of the registered users have supervisor authorities. Any user can send a price update request. Only the supervisors can confirm or reject an update request. All actions and information stored in the application database. 

Use case diagram

# Relevant scenarios

|  |  |
| --- | --- |
| **Scenario ID: SC1** | **Corresponds to UC1** |
| Description | Registered user wants to send an update request |
| Precondition | User taps on a station |
| Post condition | User sends a request to update price |
| **Step#** | **Step description** |
| 1 | User taps on a station |
| 2 | In the menu selects a fuel type |
| 4 | Inserts new price and taps on send |

|  |  |
| --- | --- |
| **Scenario ID: SC2** | **Corresponds to UC2** |
| Description | User wants to see the stations which only sells a specific type of fuel |
| Precondition |  |
| Post condition | Only corresponding stations are shown on the map |
| **Step#** | **Step description** |
| 1 | On the main menu the user selects filter fuel type |

|  |  |
| --- | --- |
| **Scenario ID: SC3** | **Corresponds to UC3** |
| Description | User wants to see only one brand |
| Precondition |  |
| Post condition | Only corresponding stations are shown on the map |
| **Step#** | **Step description** |
| 1 | On the main menu the user selects filter brand |

|  |  |
| --- | --- |
| **Scenario ID: SC4** | **Corresponds to UC4** |
| Description | User wants to see extra services provided by the station |
| Precondition | A station must be selected |
| Post condition | The extra services at the selected station are shown |

|  |  |
| --- | --- |
| **Scenario ID: SC5** | **Corresponds to UC5** |
| Description | User wants to rate the services |
| Precondition | A station must be selected |
| Post condition | User rates the station from 0 to 5 |

|  |  |
| --- | --- |
| **Scenario ID: SC6** | **Corresponds to UC6** |
| Description | User wants to reject or confirm an update request |
| Precondition | A station must be selected |
|  | Un update request must be sent for that station |
| Post condition | User confirms or rejects the request |

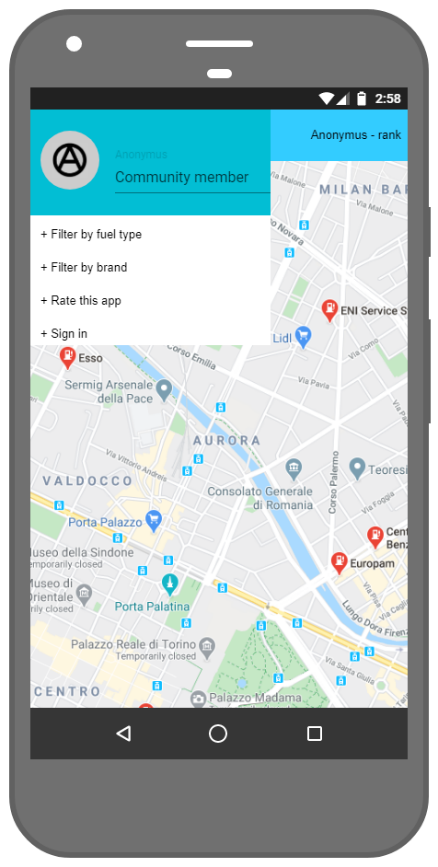
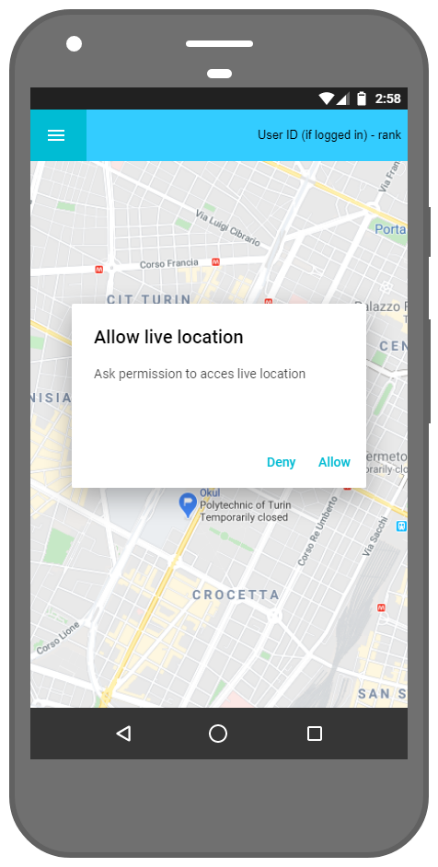
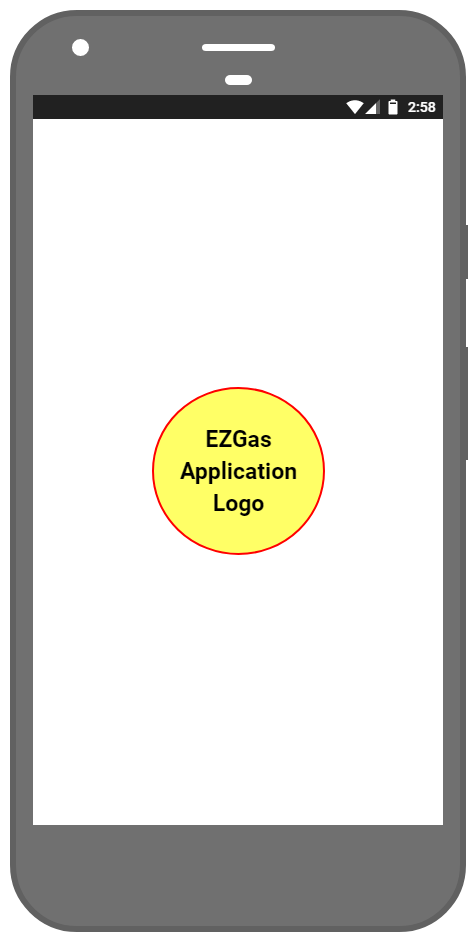
|  |  |
| --- | --- |
| **Scenario ID: SC7** | **Corresponds to UC7** |
| Description | User wants add a new station to the system |
| Precondition |  |
| Post condition | User sends s request to add the new station |

|  |  |
| --- | --- |
| **Scenario ID: SC8** | **Corresponds to UC8** |
| Description | Administrator removes a user |
| Precondition | User sends a request to remove his/her account |
| Post condition | Account is removed |

|  |  |
| --- | --- |
| **Scenario ID: SC9** | **Corresponds to UC9** |
| Description | Administrator blocks an account |
| Precondition |  |
| Post condition | Account is blocked for a certain time |

# Glossary

# User interface



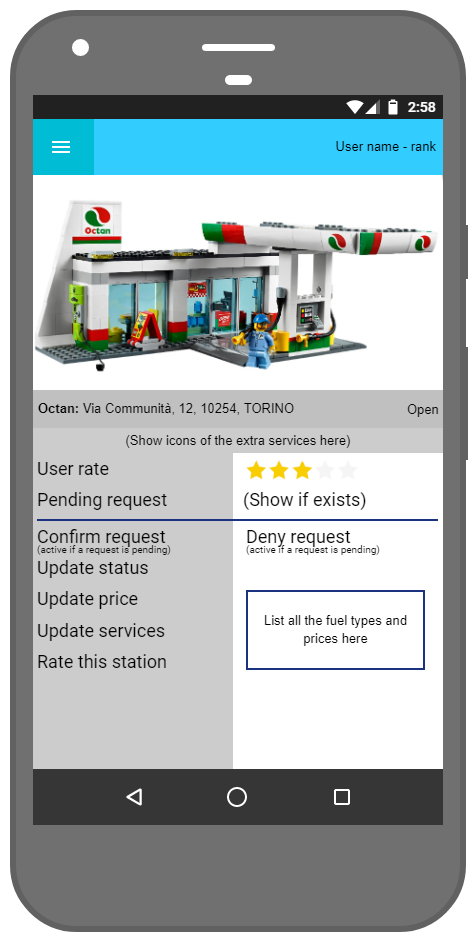
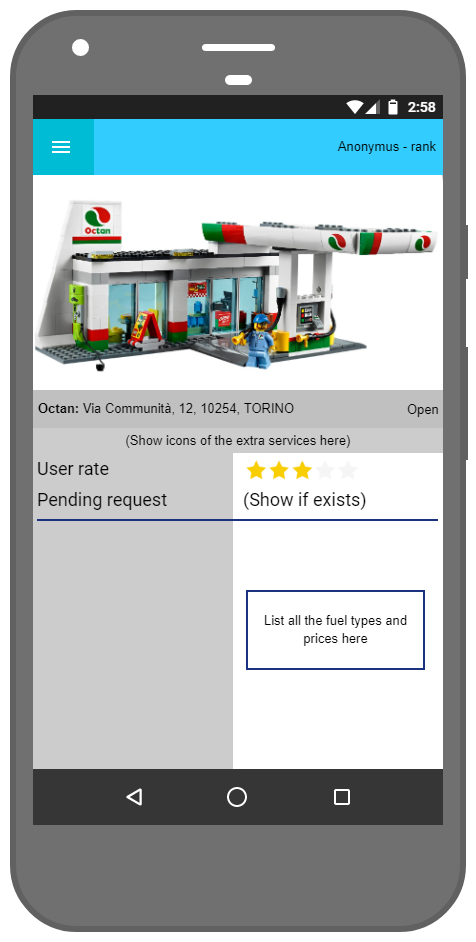
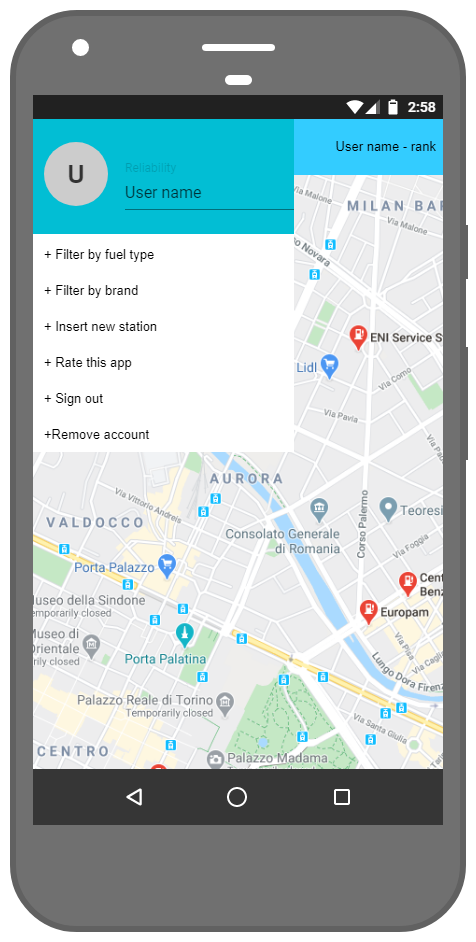
First page

Home page

Anonymous user combo box

When the application is started, application logo is shown to the user. In the home page map is shown directly. If live location is of the device is switched of, it asks to activate it. Every user is signed as anonymous user if not signed in. In the top up menu, anonymous user sees filtering options, app rate button and sign in button.

If the user is registered, s/he sees insert new station, sign out and remove accounts options in the pop up menu in addition to the default options. When a station is selected, every user see the address, status, extra services, user rate and pending update requests. In the right side of the screen all fuel type prices shown in a list. If the user is registered, s/he sees confirm and deny request, update status, update price, update services and station rate buttons in addition to the default ones.



Registered user combo box

User selected station

Registered user selected station