Official Requirements Document for EzGas

Author: s236076

Date: 5/04/2020

Version: 2

Content of Document:

1. Description
2. Stakeholders
3. Stories and personas
4. Context diagram and interfaces
5. Functional and non functional requirements
6. Use Case Diagram and use cases with relevant scenarios
7. Glossary (conceptual diagram)

1. Description

EzGas is a crowdsourcing service that allows users to collect prices of fuels in different gas stations and locate gas stations in an area along with their relative prices.

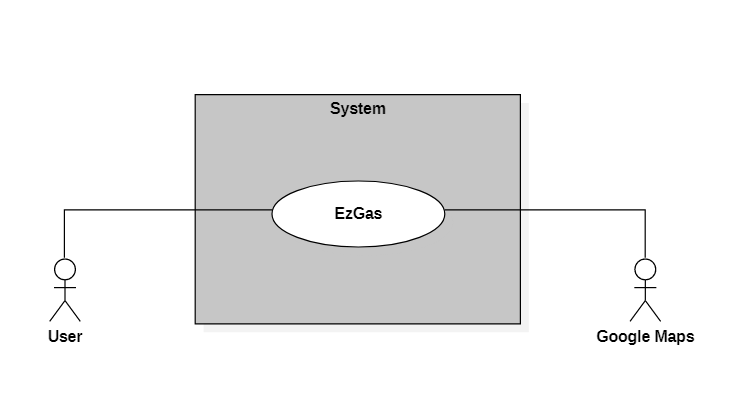
2. Stakeholders

|  |  |
| --- | --- |
| **Stakeholder Name** | **Description** |
| User | He who uses application to check the price of fuel in gas stations and to search for the nearest or most convenient one in his area |
| Developer | He who develops the application |
| Google Maps | System that provides maps to the application |
| Administrator | He who manages the application |

3. Stories and Personas

* John leads a very busy life and is always in a hurry.   
  He uses his car to get around and to him time is of the essence. For this reason, he would like to be able to find the nearest gas station whenever he needs to refill his vehicle so as to not waste time.
* Viviana is a single mum of three.  
  She alone must provide for her family and consequently cannot afford to spend too much money on fuel. Being able to locate where the most convenient gas stations is in her area would help her.
* Larry is a truck driver.  
  Everyday he travels long distances through areas that most of the time he is not familiar with. It would be useful for him to be able to know where gas stations near him are located when he need to refill, and maybe have the option of locating the most convenient one.
* Abigail has always believed that knowledge is power.  
  That’s why when she sees an offer, she thinks everyone should know about it. She works an hour from home and knows first-hand how expensive fuel can be as she relies on her car to get her there. For this reason, when she comes across a gas stations with great fuel prices she would love to be able to share this information with whoever could benefit from it.

3. Context Diagram and interfaces



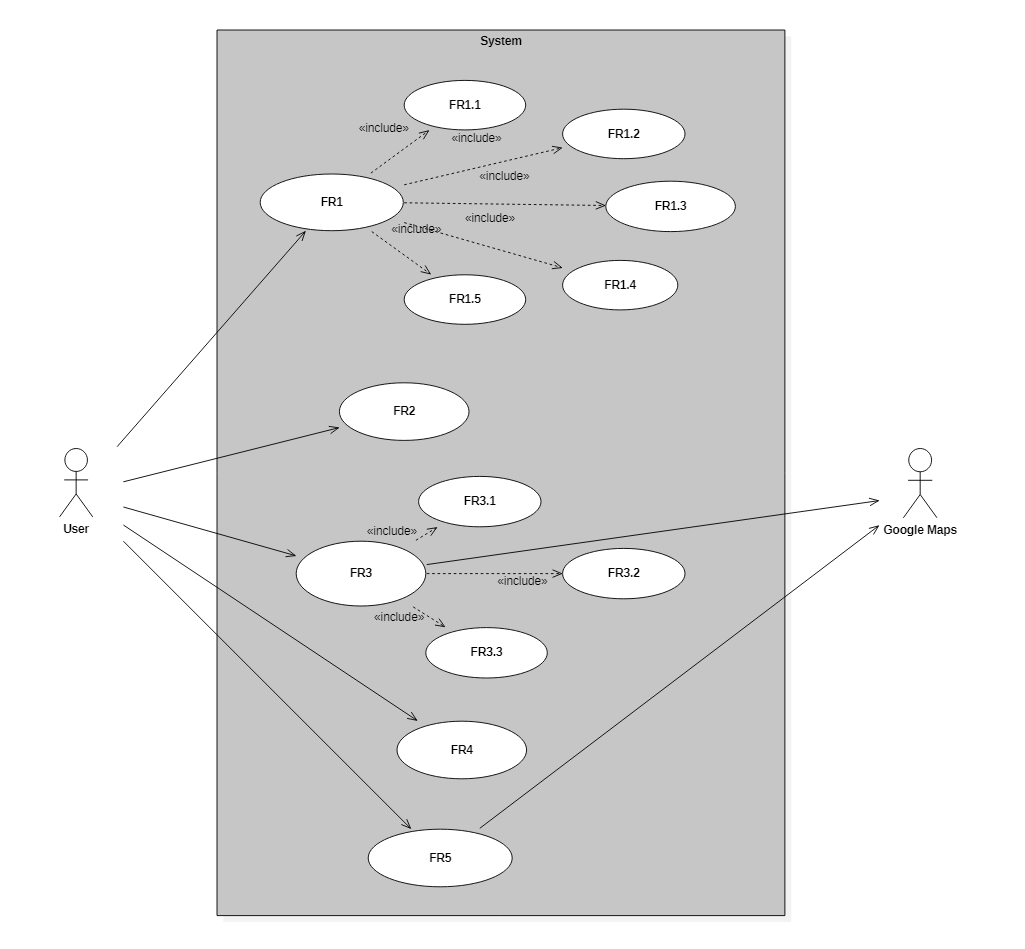
|  |  |  |
| --- | --- | --- |
| **Actor** | **Logical Interface** | **Physical Interface** |
| User | GUI | Screen of Smartphone |
| Google Maps | Web Service | Internet Connection |

4. Functional and non functional requirements

|  |  |
| --- | --- |
| **Functional Requirement ID** | **Description** |
| FR1 | Authorize and authenticate user |
| FR1.1 | Create EzGas Account |
| FR1.2 | Login to EzGas |
| FR1.3 | Logout of EzGas |
| FR1.4 | Change account password |
| FR1.5 | Change account username |
| FR2 | Set fuel type |
| FR3 | Search for gas stations in the area (for each one show fuel price and distance to reach it) |
| FR3.1 | Search based on distance to gas station (results in ascending ordere according to distance to reach the station from current location) |
| FR3.2 | Search based on price of fuel (results ordered according to ascending price of fuel of the station) |
| FR3.3 | User can view the gas stations in their area and their prices on a map directly from the application |
| FR4 | Update fuel price of a station |
| FR5 | Validate price of fuel at a station |
| FR6 | Add gas station |

|  |  |  |  |
| --- | --- | --- | --- |
| **Non functional requirement ID** | **Type** | **Description** | **Refers to** |
| NFR1 | Usability | User should be able to use the application with no previous training | All FR |
| NFR2 | Efficiency | Each function should respond in less than 0,3 seconds | All FR |
| NFR3 | Reliability | The downtime of the application should be maximum 1 hour per year | All FR |
| NFR4 | Portability | The application runs on Android (7.0 to 10.0) and iOS (9.3.6 to 13.4) | All FR |
| NFR5 | Domain | Currrency (euro, dollar, ecc.) and units (liter, gallon, ecc.) depend on the country the user is in | All FR |

6. Use Case Diagram and use cases with relevant scenarios



*Use Case*: Create Account;  
*References:* FR1.1;  
*Primary Actor*: User;  
*Support Actors:* None;  
*Precondition:* Account for that user must not already exist;  
*Postcondition:* Account for user must exist;  
*Nominal scenario:*

1. User sets email;
2. User inserts name and surname;
3. User chooses a username;
4. Users sets password;
5. System validates and confirms;
6. User receives email of confirmation;

*Exception Scenarios:* email or username not valid, issue warning;

*Use Case:* Login;  
*References:* FR1.2; *Primary Actor:* User; *Support Actors:* None; *Precondition:* Account for the user exists, user is logged out; *Postcondition:* User is logged in; *Nominal scenario:* User enters email and password and logs in;  
*Exceptional scenarios:* user inserts invalid email or incorrect password, issue warning;

*Use Case: Logout;  
References:* FR1.3; *Primary Actor:* User; *Support Actors:* None; *Precondition:* User must be logged in; *Postcondition:* User is logged out; *Nominal scenario:* User logs out; *Exception Scenarios:* None;  
  
  
  
*Use Case: Change account password;  
References:* FR1.4; *Primary Actor:* User; *Support Actors:* None; *Precondition:* User must be logged in; *Postcondition:* None; *Nominal scenario:*

1. User goes to general settings -> account settings -> change password;
2. User inserts current password;
3. User inserts new password to substitute;
4. User saves changes;

*Exception Scenario: User inserts incorrect current password. Issue warning telling user to try again;  
  
  
Use Case: Change account username;  
References:* FR1.5; *Primary Actor:* User; *Support Actors:* None; *Precondition:* User must be logged in; *Postcondition:* None; *Nominal scenario:*

1. User goes to general settings -> account settings -> change username;
2. User inserts new username;
3. User saves changes;

*Exception Scenario: User inserts username that is already taken. Issue warning to user to insert valid username;*

*Use Case:* Set fuel type;  
*References:* FR2; *Primary Actor:* User; *Support Actors:* None; *Precondition:* None; *Postcondition:* Fuel type is specified; *Nominal scenario:*

1. User opens app and specifies how he wants to search for gas stations in the area (distance or price);
2. User inserts fuel type he is interested in searching for;
3. User proceeds with the search;
4. Application displays results;

*Exception Scenarios:* None;

*Use Case:* Search gas stations by price;  
*References:* FR3.1; *Primary Actor:* User; *Support Actors:* Google Maps; *Precondition:* None (searching by price is default option);  
*Postcondition:* None; *Nominal scenario:*

1. User searches for gas stations;
2. Application returns results of search, from cheapest option to most expensive;

*Exceptional scenarios:*

1. No gas station in the surrounding area. Issue warning;
2. Localisation of the device is turned off. Notify the user to turn on location to keep using the application;

*Use Case:* Search gas stations by distance;  
*References:* FR3.2; *Primary Actor:* User; *Support Actors:* Google Maps; *Precondition:* User must has specified he wants to search by distance (default option is by price);  
*Postcondition:* None; *Nominal scenario:*

1. User searches by distance;
2. Application returns results of search, from closest option to farthest away;

*Exceptional scenario:*

1. No gas station in the surrounding area. Issue warning;
2. Localisation of the device is turned off. Notify the user to turn on location to keep using the application;

*Use Case:* View gas stations in the area;

*References:* FR3.3; *Primary Actor:* User; *Support Actors:* Google Maps; *Precondition:* Phone location must be on;  
*Postcondition:* None; *Nominal scenario:*

1. User opens application;
2. Application shows map of surrounding area with gas stations and their relative prices;

*Or:*

1. User opens application;
2. User performs a search (by distance or price);
3. Results of the search (gas stations with their prices) are displayed as a list;
4. User has the option to “view results on map” directly in the app;

*Exceptional scenarios:* Localisation of the device is turned off. Notify the user to turn on location to keep using the application; *Use Case:* Update price of fuel in gas station;  
*References:* FR4; *Primary Actor*: User; *Support Actors:* None; *Precondition:* User is logged in, gas station exists; *Postcondition:* Price of the gas station has been updated; *Nominal scenario:*

1. User selects gas station he wants to update price of fuel for;
2. User specifies type of fuel of which he wants to update price;
3. User inserts current price*;*

*Exception Scenarios:* price inserted isn’t valid (for example the user inserted letters and not numbers), issue warning and tell user to insert valid data;  
  
  
  
*Use Case:* Validate price of fuel in a gas station;  
*References:* FR5; *Primary Actor*: User; *Support Actors:* None; *Precondition:* User is logged in, gas station exists; *Postcondition:* None; *Nominal scenario:*

1. User selects gas station he wants to validate price of fuel for;
2. User specifies what price of fuel he wants to validate of that station;
3. User validates price*;*

*Exception Scenarios:* None;  
  
  
  
*Use Case:* Add a gas station;  
*References:* FR6; *Primary Actor*: User; *Support Actors:* None; *Precondition:* User is logged in, gas station exists; *Postcondition:* None; *Nominal scenario:*

1. User opens application;
2. User adds gas station, specifying the location (either by inputting the address or by using localisation of smartphone if the user is currently at the gas station);
3. Optional: user can also specify price of fuel at the station*;*

*Exception Scenarios:* None;

7. Glossary

