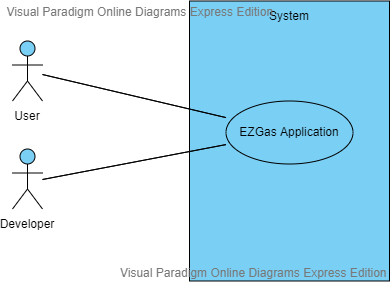
**Official Requirements Document**

**#STAKEHOLDERS**

* User: uses the application to see the prices of fuels in different gas stations and to locate gas stations in an area, along with the prices they practice. Everyday an user refuel, he updates the prices for that gas station or insert the new gas station with its prices;
* Developer: handles the service and manages some possible errors (gas station position or prices).

**#CONTEXT DIAGRAM AND INTERFACES**

**##CONTEXT DIAGRAM**



**#INTERFACES**

User | Gui | Screen, touch |

Developer | Gui | Screen, touch, keyboard |

**#STORIES AND PERSONAS**

Mike is a young business owner and he has a great civic sense that lead him to create a platform to help people saving money. The software collects prices of fuels in different stations and provides them to people in order to get gas as cheaper as possible relating with their position. When people need to refuel, they consult the platform to find the nearest station with the lowest price. After that, they update the prices (if they change with the old ones) or insert a new gas station in the platform with the prices.

Mike also manages all users reports concerning errors related with gas station position and gas price and he has to handle users.

Paul lives in Monza, he is 29th and works for Unicredit, and his office is in Milan. Every day he drives to work with his own car and every morning he has to refuel. While having breakfast, he consults the platform to find the nearest station with the best price. When he refuels, if there prices are different from the one in the platform he communicates the errors as fast as possible. In some cases, Paul forgot to refuel in the morning and make this in the evening in Milan before come backing home. If he refuels in a gas station that is not present in the application, he shares his experience as soon as possible. He enjoys help people.

Michael, which travels three/four times a week, exploits EZGas application to discover the nearest gas station with the better prices, but he is not an active user. He doesn’t communicate updates of prices and if he refuels in a gas station which is not in the application (with good prices) doesn’t’ share his experience. He is a passive user.

Frank is the head master of Vodafone Italia, and he travels around Italy every day. He doesn’t take care of prices because Vodafone covers every costs. So he uses the app to find the nearest station and he is not interested on sharing information with the community.

**#FUNCTIONAL AND NON FUNCTIONAL REQUIREMENTS**

**##FUNCTIONAL REQUIREMENTS**

* FR1:Handle users;
* FR1.1: registers users to EZGas application;
* FR1.2: log-in users to EZGas application, in order to share their experience;
* FR1.3: provides service to guest user (people that use the app only to have the best price without sharing anything);
* FR1.4: remove an user;
* FR1.5: record user’s activities;
* FR2:Handle service
* FR2.1:Insert location where to find out stations;
* FR2.2:Displays gas stations(with prices) which follow the requirements;
* FR3:Share experience
* FR3.1:Insert new gas station with prices;
* FR3.2: Update existing information;
* FR3.3:Insert some comments;

**##NON FUNCTIONAL REQUIREMENTS**

* Usability: application should work easily without any training for the user;
* Performance: functions should be complete in reasonable time (<0.5s);

**#SCENARIOS, USE CASES AND USE CASE DIAGRAM**

**##SCENARIOS**

**###SCENARIO 1**

Description: developer corrects an error

Precondition: developer knows the error

Post condition: some information have been changed

Steps:

1. Developer reads the error report;
2. Developer selects the gas station;
3. Developer update information.

**###SCENARIO 2**

Description: user consults the application to find the cheapest gas station around him;

Precondition: user has access to application (registered and installed it);

Post condition: user knows the most convenient station.

Steps:

1. User open the app and log in;
2. Insert information concerning its position;
3. Chose one of the displayed gas stations;
4. After he refuel, if there are any changes update the prices.

**###SCENARIO 3**

Description: user find out a gas station that is not in the application;

Precondition: user has access to the application (registered and installed it);

Post condition: the application has a new gas station.

Steps:

1. User open the application and log in;
2. Insert information concerning his position;
3. Search the gas station from the one around him;
4. If it is not present, insert the new gas station with respective prices.

**###SCENARIO 4**

Description: a guest user use application to find the nearest gas station

Precondition: user has access to application as guest;

Post condition: the user refuels his car

Steps:

1. User open the application and access as guest;
2. He chooses the nearest gas station;

**##USE CASES**

**###USE CASE 1, UC1-FR1.1 REGISTRATION OF USER**

Actor involved: User

Precondition: user hasn’t an account yet and he has a valid e-mail address.

Post condition: user will have an account;

Nominal scenario: user insert his personal data to perform registration and then he will have an account to use.

Variants: user has been already registered; e-mail is invalid; some other fields are invalid

**###USE CASE 2, UC2-FR1.2 LOG-IN USER**

Actor involved: User

Precondition: user has already an account;

Post condition: user performed the log in;

Nominal scenario: user inserts username and password to log in in the application;

Variants: user is not registered; user has forgot his password.

**###USE CASE 3, UC3-FR1.3 GUEST USER**

Actor involved: User

Precondition: user has installed the app

Post condition: read the nearest gas stations

Nominal scenario: user enters in the app as guest, without having an account.

Variants

**###USE CASE 4, UC4-FR1.4 REMOVE USER**

Actor involved: User

Precondition: User has an account

Post condition: User has deleted his account

Nominal scenario: user performs this action to delete his account

Variants: the account doesn’t exist; information are wrong

**###USE CASE 5, UC5-FR2,1 INSERT LOCATION**

Actor involved: User

Precondition: User has an account and has performed the login

Post condition: Location is known

Nominal scenario: user insert information about the range of research.

Variants: the information inserted are wrong (the address doesn’t exist)

**###USE CASE 6, UC5-FR2,2 DISPLAY STATIONS**

Actor involved: User

Precondition: User has an account, has performed the login and has inserted information about the location

Post condition: List of gas station is displayed

Nominal scenario: a list of gas station appears on the screen

Variants: there aren’t gas stations in the area.

**###USE CASE 7, UC5-FR3.1 INSERT NEW GAS STATION**

Actor involved: User

Precondition: User has an account, has performed the login

Post condition: A new gas station will be in the list

Nominal scenario: user insert a new gas station

Variants: the gas station is already present

**###USE CASE 8, UC5-FR3.2 UDATE PRICES**

Actor involved: User

Precondition: User has an account, has performed the login and has refuelled

Post condition: Prices of the selected gas station are updated

Nominal scenario: user update the prices of a gas station due to an error or a changing of prices

Variants:

**###USE CASE 9, UC5-FR3.3 COMMENTS**

Actor involved: User

Precondition: User has an account, has performed the login and has refuelled

Post condition: Gas station, where the user has refuelled, will have a new comment

Nominal scenario: user insert a comment describing his experience

Variants:

**##USE CASE DIAGRAM**