Author: Razieh Meidanshahi, S274070

Abstract

Nowadays IT technology brings convenience to everyone's life. The energy and fuel industries are not exceptional in this area. Regarding the needs of vehicle drivers to find out the best station considering parameters like distance, fuel, and price in an appropriate time.

Fortunately, a software system called EZGass is developed to satisfy not only the requirements of drivers but also manage the station owners' issues. Tacking in to account the time and expenses that each owner has to put into advertising, accounting management, and, handling the station's issues, EZGass, is the solution to demonstrate the proper station to the driver also ables the owners to do all the related working stuff online.

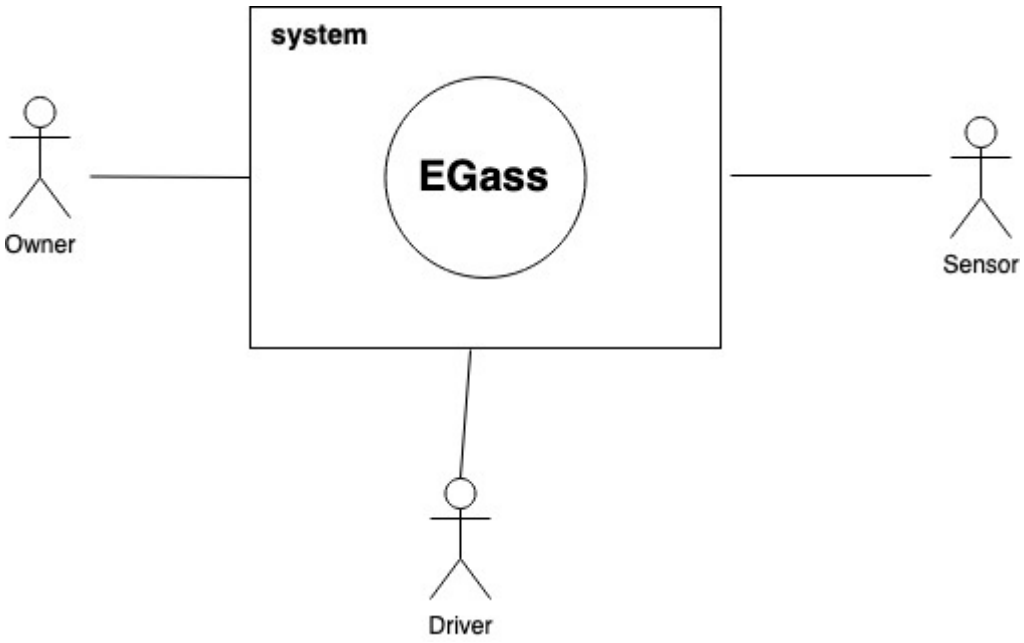
Furthermore, EZGass has a connection to tank sensors in order to notify the owner to order the fuel before the station tank quantity becomes less than the critical storage and recharge tank stations by the application.

Stakeholders

Stakeholder name	Description
Station Owner	Uses the application to manage accounting, quantity monitoring, and order fuel or recharge electricity
Driver	Uses the application to manage accounting, quantity monitoring, and order fuel or recharge electricity

Context Diagram and interfaces

Context Diagram



Interfaces

Actor	Logical Interface	Physical Interface
Station Owner	GUI	Screen, keyboard
Driver	GUI	Screen, buttons
Tank Sensor	TS	main commands are read and send quantity

Stories and personas

Ava is a young talented software system analyst who works in a software development company. She always has meetings with the customer most of the meeting schedules early in the morning.

Once she woke up late and was rushing to the office to attend an important meeting, she noticed that her car was run out of fuel or even electricity charge. As she is smart she called her friend, Alex whom his father owns some stations. Fortunately, Alex sent her the address of one of the latest stations that his father built near Ava's apartment. the station was not famous because of the location but the prices were surprisingly less in comparison with the others in the center also it was not crowded with long queues.

Since Ava could attend the meeting on time and get rewards due to her good job on the analysis, She went to visit Alex's father Jack. Jack talked about his challenges of managing stations and low income compared to stations' expences. Ava developed the proposal of EZGass that is the most famous application for drivers and station owners these days.

Recently, Jack just logs in to the application and check the account balance of his stations also monitors the notifications to order fuel for the tanks with quantity

near to critical savings. Moreover, gets notification 7 days before the rechargedate for each electronic stations. Whenever the tanks are recharged, Jack just need to recharge the tank quantity by the application

Nowadays, millions of drivers become the EZGass' users because as soon as they open the application, they are able to see stations around their location with their prices. The users can fill out the parameters like distance, fuel, price to find out the proper station. EZGass is also famous for its easy ways of payment that is online and works with every kind of bank accounts.

Functional and non functional requirements

Functional Requirements

ID	Description
FR1	Record that a station has an owner, address, type(Electric, Fuel), and, FuelType
FR2	Record that a station has prices for a fuel type
FR3	Record that a fuel station tank has a critical saving for a fuel type
FR4	Record that a notification for a fuel station should be send to the owner when a tank quantity * 0.3 = critical saving
FR5	Record that a notification for an electric station should be send to the owner when currentdate +7 = station account rechargedate
FR6	Record that a fuel station tank has recharged X Gallon on its capacity
FR7	Record that an electronic station has recharged X Euros on its account
FR8	Record that when a fuel tank is used CurrentQuantity= CurrentQuantity-UsedQuantity
FR9	Record that when a tank is used CurrentIncome= CurrentIncome+ (UsedQuantity*FuelTypePrice)
FR10	Produce a report about all income separately for each station and tank over a certain period of time
FR11	Produce a report about all consumption and recharge separately for each station and tank over a certain period of time
FR12	demonstrate stations that location between Userlocation+500Meter and Userlocation-500Meter and TankType in(fuelTypes) and Tankprice<=UserPrice for user search parameters
FR13	Record that an electronic station has a rechargedate
FR14	Record that when a fuel tank is recharged CurrentQuantity= CurrentQuantity+chargedQuantity

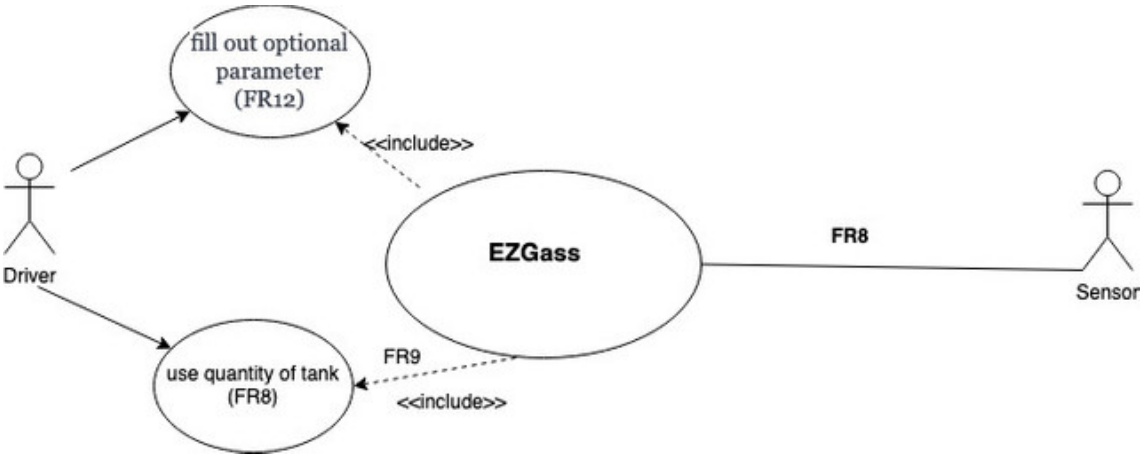
FR15	Record that when electronic station is recharged CurrentAccount= CurrentAccount+chargedAmount
## Non Functional Requirements	

ID	Type (efficiency, reliability, .. see iso 9126)	Description	Refers to
NFR1	Usability	Application should be used with no training by any drivers and station owners	All FR
NFR2	Performance	All functions should complete in < 0.4 sec	All FR
NFR3	Portability	The application runs on MS Windows (7 and more recent) also on smartphones using all versions of Android and IOS	All FR
NFR4	Portability	The application (functions and data) should run properly on every browsers	All FR
NFR5	Security	the application should have 2-way authentication for station owners	FR9,10,11,14,15

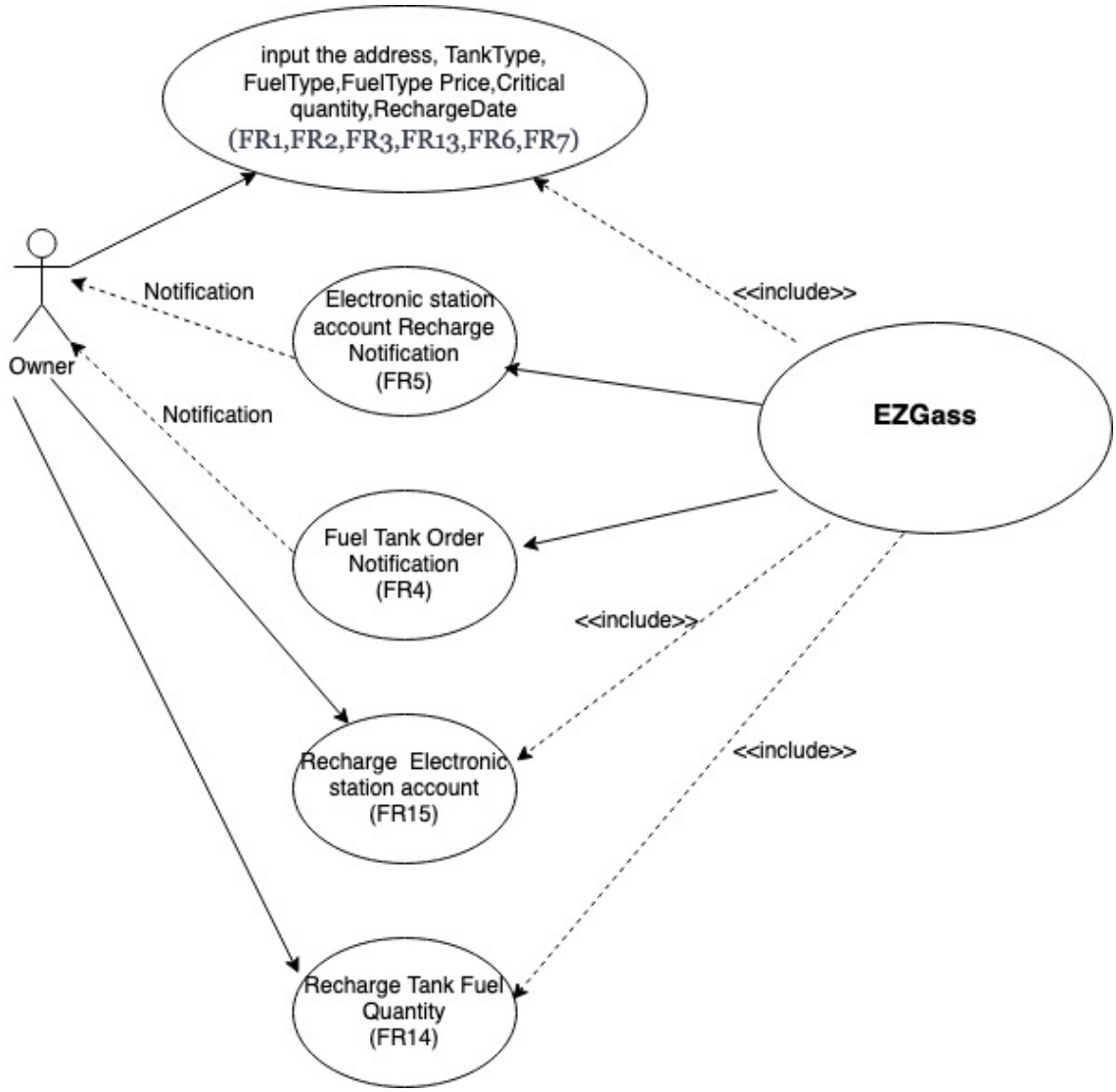
Use case diagram and use cases

Use case diagram

Searching for the station and Use fuel



Notifications and Recharge stations



Use Cases

Use case 1, UC1 - FR8 Record when a fuel tank is used

Actors Involved	Tank Sensor
Precondition	Tank T exists, FuelQuantity Q exists, CurrentQuantity C exists, UsedQuantity U exists, TankIncome exists I,FuelPrice FP exists
Precondition	T.Q_pre > T.Q_post, T.C > T.U
Post condition	T.Q_pre < T.Q_post, T.C > T.U, T.I_pre < T.I_post
Nominal	Tank sensor reads the quantity that is used by the user and send

Scenario	the used quantity to the server, the application set the current quantity of the T.C = T.c - T.U also T.I = T.I + T.FP
Variants	phisical Current Tank Quantity is diffrenet from the application Tank CurrentQuantity, T.C < T.U

Use case 2, UC2 - FR14 Record that when a fuel tank is recharged

Actors Involved	**Station Owner**
Precondition	Tank T exists, FuelQuantity Q exists, CurrentQuantity C exists, ChrgedQuantity X exists
Precondition	T.Q_pre < T.Q_post
Post condition	T.Q_pre > T.Q_post
Nominal Scenario	Station Owner charged the quantity of the tank, then T.C = T.C + T.X
Variants	phisical Current Tank Quantity is diffrenet from the application Tank CurrentQuantity, wrong T.X

Use case 3, UC3 - FR15 Record that when electronic station is recharged

Actors Involved	**Station Owner**
Precondition	Station Account SA exists , quantity >0
Post condition	SA.balance_post = SA.balance_pre + quantity
	StationAccount.balance_post = StationAccount.balance_pre + quantity
Nominal Scenario	Station Owner selects account of station SA, increase account of quantity, increase Station account of quantity
Variants	

Use case 4, UC4 - FR12 demonstrate stations location

Actors Involved	**Driver**
Precondition	driver use smartphone with IOS or Android and GPS Set to On Mode,Userlocation U exists,Stationlocation S exists, UserFuelType UFT optional,TankType TT exists,Tankprice TP exists, UserPrice UP optinal
Post condition	S between U+500Meter and U-500Meter and UFT in (TT) and TP <= UP
Nominal	Driver input optional parameters and the application demonstrates

Scenario	the stations S between U+500Meter and U-500Meter and UFT in (TT) and TP <= UP
Variants	

Relevant scenarios

Scenario 1

Scenario ID: SC1	**Corresponds to UC1**
Description	Sensor reads used quantity of a tank type T
Precondition	Tank of T has enough quantity for UsedQuantity U
Postcondition	quantity of T updated, quantity of U recorded
Step#	Step description
1	Sensor reads one used quantity of a tank type T
2	Sensor Send used quantity of a tank type T to server
3	Deduce current quantity of tank T from UsedQuantity
4	update tank income of T from price of fueltype and UsedQuantity U

Scenario 2

Scenario ID: SC2	**Corresponds to UC1**
Description	station owner recharge one tank of type T, quantity positive
Precondition	Tank of T has quantity near to Criticalsaving Quantity
Postcondition	quantity of T updated
Step#	Step description
1	station owner selects of a tank type T
2	station owner input recharge quantity of type Q
3	update current quantity of tank T from recharge quantity of type Q

Glossary

```
class EZGass
class Station Owner {
+ name
```

```
+ surname
}

class Electronic Station Account {
+ balance
}

class FuelType {
+ name
+ price
+ quantity
}

class TankType {
+ name
}

class Tank {
+ Capacity
+ CriticalQuantity
}

class UseFuel {
+ date
+ Quantity
}

class Recharge
class Consumption
```