

## Lab1-1: Requirements

Goal of this lab is to practice the techniques to formalize the requirements of a product. One case is a pure software application (EZGas), the other is a product that contains software (RVC).

Always consider the possible defects in a requirement document (omissions, inconsistencies, ambiguities, incorrect facts ..).

Consider that the document must be sufficient to:

- allow another team (not you) to design and code the application
- allow another team (not you) to test the application

For the diagrams you can use tools like Signavio, Argo UML, Astah, Star UML, Plant UML.

Use the slack channel for questions that could interest other students.

For the cases described later (last page) define:

0 business model

1 Stakeholders

2 Context diagram and interfaces

3 Functional requirements

You can use stories to clarify your ideas and define the functional requirements. You can use hierarchy to group functional requirements that are logically (not temporally) related.

F1	
----	--

F1.1	
F1.2	
F1.2.1	
F2	
F3	
F4	
F4.1	
F4.2	
F4.3	
F4.4	
F5	

4 Non functional requirements

5 table of rights

actor	Gas station owner	driver	admin
-------	-------------------	--------	-------

F1 write price of fuel for a gas station	yes	no	
F2 read price of fuel			
F3 create account	no	no	yes

## CASES TO BE DESCRIBED

### EZGas

EZGas is an application to help drivers find gas at lowest prices.

Gas station owners can register their gas station with prices and eventually discounts. Users look for gas stations closest to them and with best prices and quality of service.

### Robotic vacuum cleaner

Since several years robotic vacuum cleaners (RVC) are available. An RVC is capable of cleaning the floors of a house in autonomous mode.

An RVC system is composed of the robot itself and a charging station. The charging station is connected to an electric socket in the house, and allows charging the battery on board of the robot.

The robot itself is composed of mechanical and electric parts, a computer, and sensors. One infrared sensor in the frontal part recognizes obstacles, another infrared sensor always on the frontal part recognizes gaps (like a downhill staircase). A sensor on the battery reads the charge of the battery. The computer collects data from the sensors and controls the movement of four wheels. Another sensor on one of the wheels computes direction and distance traveled by the robot.

Finally on top of the robot there are three switches: on-off, start, learn.

The learn button starts a procedure that allows the robot to map the space in the house. With a certain algorithm the robot moves in all directions, until it finds obstacles or gaps, and builds an internal map of this space. By definition the robot cannot move beyond obstacles, like walls or closed doors, and beyond gaps taller than 1cm.

The starting point of the learn procedure must be the charging station. When the map is built the robot returns to the charging station and stops.

The start button starts a cleaning procedure. The robot, starting from the charging station, covers and cleans all the space in the house, as mapped in the 'learn' procedure.

In all cases when the charge of the battery is below a certain threshold, the robot returns to the charging station. When recharged, the robot completes the mission, then returns to the charging station and stops.