# Stakeholders

## RVC User

### Final user of the system

## RVC Company

### Includes software/electrical/mechanical engineers, marketing people…

## RVC Retailer

### Store that buys from company and sells to final users

## RVC Maintainer

### Technicians for installation and maintenance

## Certification Authority

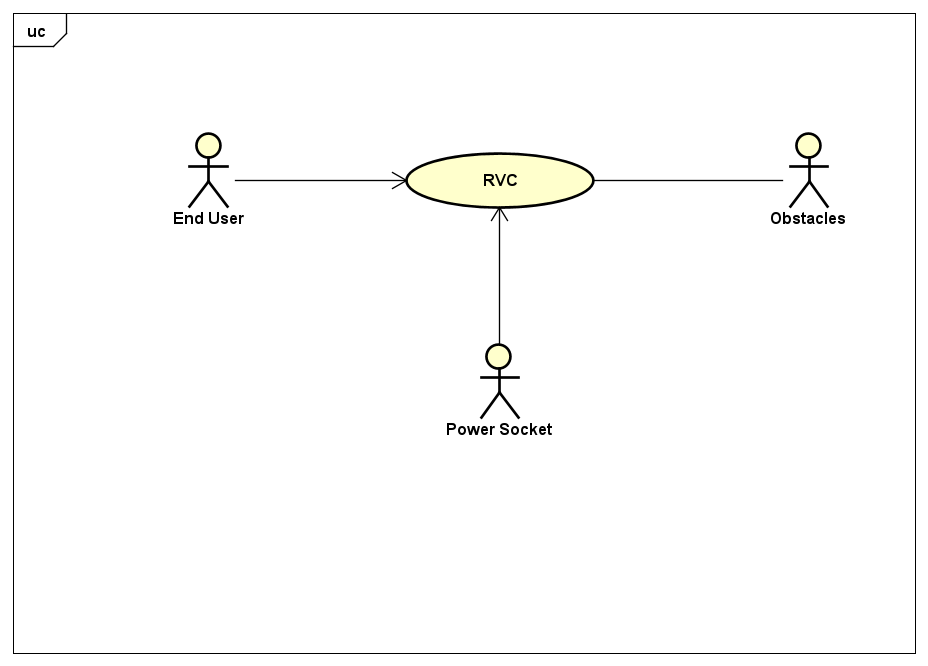
### Entity that issues safety/energy certificates

## Obstacles

### Walls, objects, animals, doors…

# Context diagram and interfaces

## Context diagram



## Interfaces

### End user

#### On/Off switch: On/Off command

#### Start switch: “Start” command

#### Learn switch: “Learn” command

### Power socket

#### Shuko standard: 220V 50 HzO

### Obstacle

#### IR sensor: Detect obstacle

# Functional and Non-Functional Requirements

## Functional Requirements

### FR1 - Clean the house

#### 1.1 Brush engine On/Off

#### 1.2 Detect when dust bag is full

### FR2 - Move autonomously

#### 2.1 Detect obstacle

#### 2.2 Detect gap

#### 2.3 Compute path

##### 2.3.1 Compute path for cleaning

##### 2.3.2 Compute path to return to charging station

#### 2.4 Implement the path

#### 2.5 Compute current position

#### 2.6 Read wheel sensor (direction and speed)

#### 2.7 Control wheels (direction and speed)

#### 2.8 Detect when stuck

#### 2.9 Determine best path

### FR3 - Learn the map

#### 3.1 Map each tile as free or obstacle

### FR4 - Manage battery level

#### 4.1 Read battery level sensor

#### 4.2 Charge battery

### FR5 - Avoid harming people – Safety

#### 5.1 Stop immediately in case of potentially dangerous situation for people or pets

#### 5.2 Check battery temperature

#### 5.3 Check CPU temperature

## Non-Functional Requirements - Refer to ISO 9126

### NFR1 - RVC should be able to clean at least 100 sqmt with a single charge

### NFR2 - Energy efficiency: Energy consumed per cleaned square meter should be less than ??

### NFR3 - Noise level should not exceed 30 dB

### NFR4 - Cleaning efficiency: No space should be left uncleaned

### NFR5 - Usability: should not take more than 5 minutes to learn how to use by non-engineer

### NFR6 - Efficiency: response time to any button should not exceed 1 second

### NFR7 - Safety: no harm to people should be allowed under any circumstance

#### 7.1 Safety: speed should never exceed x m/s

### NFR8 - Dependability = safety + availability

#### 8.1 Availability should be no less than 99%

# UCD – Use Case Diagram

## UC1 – Clean the house

### PRECONDITIONS – Common for all Scenarios in this Use Case

#### Robot is ON

#### Learning procedure already completed (House mapped and cleaning path defined)

#### Battery NOT empty

### POSTCONDITIONS – Common for all Scenarios in this Use Case

#### Robot back in charging station

### ACTORS

#### End user

#### Obstacle

### Scenario UC1.1 - Nominal case – Successful cleaning

|  |  |
| --- | --- |
| PRECONDITIONS | Battery level is high enough to clean all the houseDust bag is empty enough to clean all house – We don’t have a dust bag sensor though |
| Actor “end user” pushes “start” button |  |
| Robot moves to tile 1 (first tile in a path) and cleans it |  |
| Robot moves to tile 2 and cleans it |  |
| … repeat for all tiles … |  |
| Robot returns to charging station |  |
| POSTCONDITIONS | All tiles in the house have been cleanedBattery not empty |

### Scenario UC1.2 – Unsuccessful cleaning due to low battery level

|  |  |
| --- | --- |
| PRECONDITIONS | Battery level is NOT enough to clean all the houseDust bag is empty enough to clean all house – We don’t have a dust bag sensor though |
| Actor “end user” pushes “start” button |  |
| Robot moves to tile 1 (first tile in a cleaning path) and cleans it |  |
| Robot moves to tile 2 and cleans it |  |
| Robot detects low battery level |  |
| Robot returns to charging station |  |
| POSTCONDITIONS | Battery nearly emptyHouse partially clean |

## UC2 – Learn the house

## UC3 – Manage battery

## UC4 – Don’t harm people

# Deployment diagram

# System Design & Glossary

# 