Software Engineering

Books or notes are **not** allowed.
Write only on these sheets. **Concise** and **readable** answers please.

Surname, name, matricola ____

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.

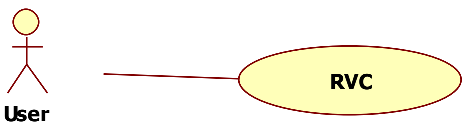
The start button starts a cleaning procedure. The robot, starting from the charging station, covers and cleans all the space in the house, as mapped in the ‘learn’ procedure.

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.

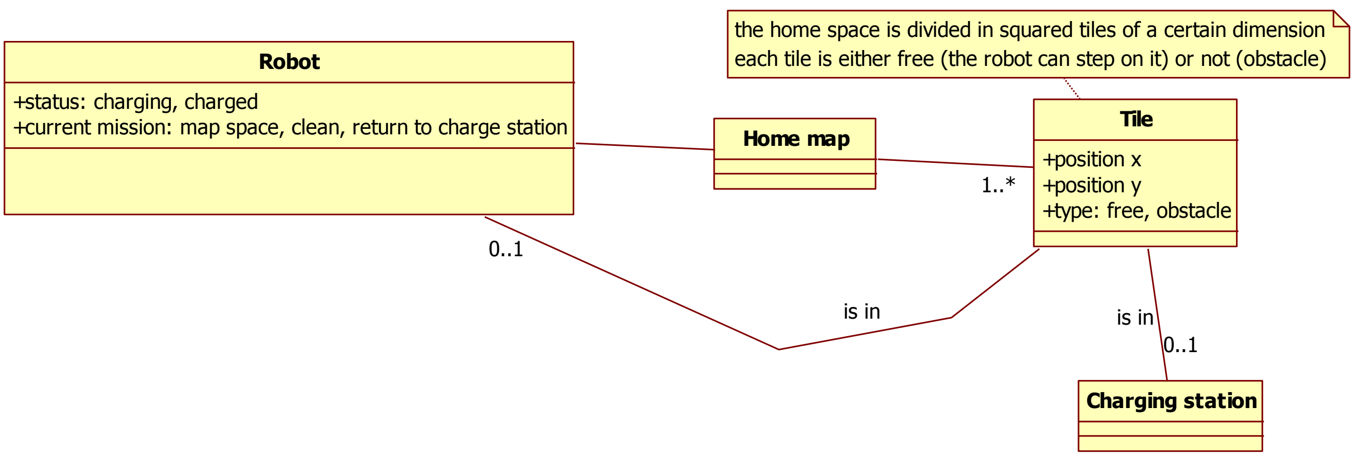
In the following you should analyze and model the RVC system.

1 – a. Define the **context diagram** (including relevant interfaces)

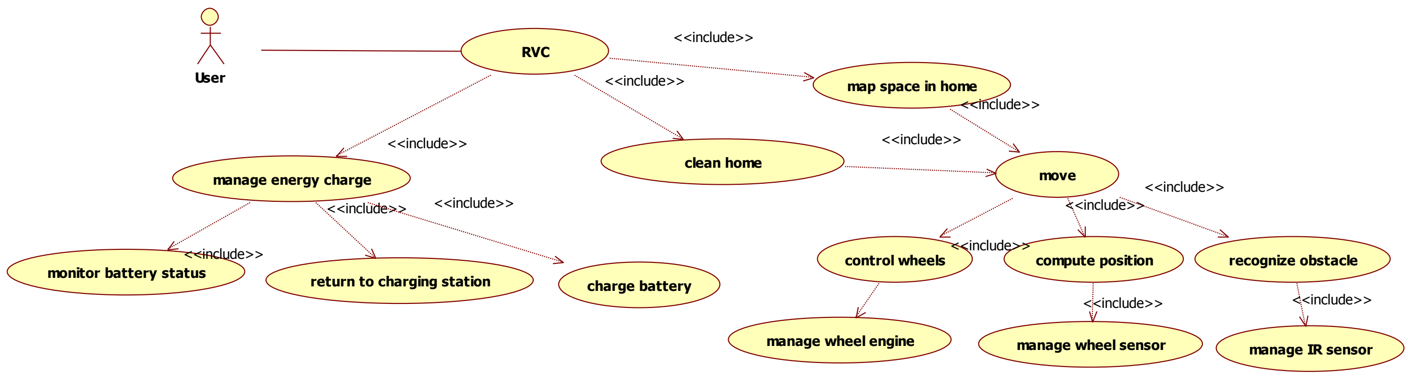
Actor	Physical interface	Logical interface
User	On off, start, learn switches	On off, start, learn command



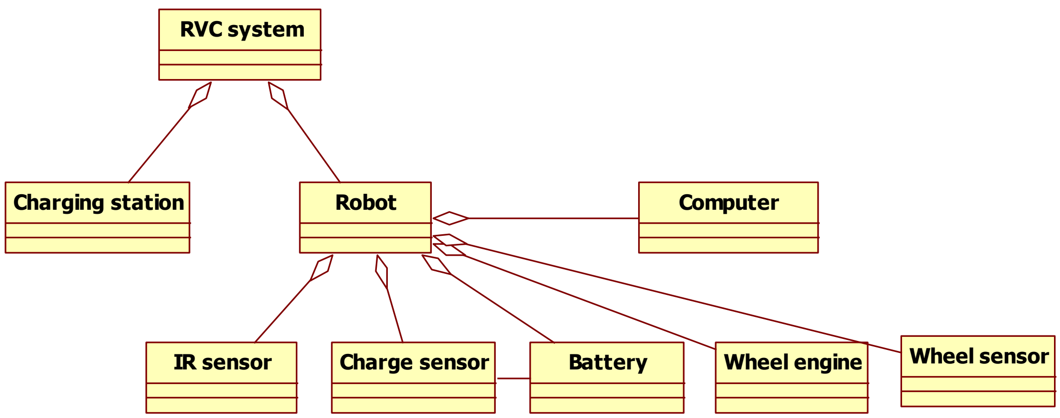
1-b Define the **glossary** (key concepts and their relationships) (UML class diagram) for the RVC System



1-c Draw the Use Case Diagram for the RVC system. For each Use Case give self-explainable long names, or a short textual description



1-d Draw the system diagram for the RVC system



2 black box

Define black box tests for the following function, using equivalence classes and boundary conditions.

boolean canGoTo(int charge, int movingMode, int distance)

The function receives the charge of the battery (can be 0 to 100), the mode (0 slow, 1 fast), the distance to be travelled (from 0 to maxint). The function returns true if the distance can be traveled by the robot, false otherwise. The formula applied is: 1 unit of charge per unit of distance in slow mode, 2 units of charge per unit of distance in fast mode.

Ex canGoTo(100, 0, 50) → true (charge available to go until distance 100/ 1 = 100 >= 50)

canGoTo(100, 1, 60) → false (charge available to go until distance 100/ 2 = 50 < 60)

Charge	Moving mode	distance	Valid / Invalid
[Minint, 0[	[minint, 0[	[minint, 0[	I
		[0, maxint]	I
	0	[minint, 0[	I
		[0, maxint]	I
	1	[minint, 0[	I
		[0, maxint]	I
	[2, maxint]	[minint, 0[	I
		[0, maxint]	I
[0, 100]	[minint, 0[	[minint, 0[	I
		[0, maxint]	I
	0	[minint, 0[	I
		[0, maxint]	V
	1	[minint, 0[	I
		[0, maxint]	V
	[2, maxint]	[minint, 0[	I
		[0, maxint]	I
]100, maxint]	[minint, 0[	[minint, 0[	I
		[0, maxint]	I
	0	[minint, 0[	I
		[0, maxint]	I
	1	[minint, 0[	I
		[0, maxint]	I
	[2, maxint]	[minint, 0[	I
		[0, maxint]	I