Barrie the bartender

"A brand new way of ordering drinks"

Minor Robotics 2017
Requirements Engineering Assignment

Team 7: Bartender Robot

Casper Cromjongh	4494156
Teun Bakker	4494490
Joppe de Waart	4365399
Vincent van Parijs	4454022
Jonathan Raes	4300343

Science Centre Delft

TU Delft

Table of Contents

1. Mission level

The first step in defining how a mechatronic system should work is by determining cases in which a person has interaction with the machine. These are called use cases. As laid out after this section, we denote 6 such use cases.

Each use case has the capabilities listed that the system should have/be able to do in that specific interaction with the user.
Next are preconditions. A precondition is a condition that has to be satisfied in order for the use case to take place.
Postconditions are conditions to the state the system and user should be in after the interaction has finished.
Scenario steps denote different phases of the interaction. Actions of both user and system are listed here
Alternative scenarios classify situations in which an alternative choice has been made in the process, or if something in the system went wrong. They are situations that are generally avoided, but still represent a possibility.

1.1 Explanation of use cases

Use case 1: System boot up
Capability: When the system is turned on, the robot can execute some self checks in order to guarantee robustness of operation.
Preconditions: The robot is turned off, but connected to the grid.
Postconditions: The robot is turned on, the systems are calibrated and the robot is ready for use.
Scenario steps:

1. The Science Centre employee turns the robot on
2. The robot calibrates its systems
3. The robot notifies the employee that the calibration was successful

Alternative scenarios:

- The robot doesn't turn on due to malfunctioning systems
- The calibration isn't successful due to malfunctioning hardware

Use case 2: Getting product inventory filled
Capability: The robot can be refilled with the supplies needed for the products
Preconditions: The stock has (nearly) run out for some or all of the available products
Postconditions: The stock is sufficiently filled by the staff
Scenario steps:

1. The science centre employee notices the stock is insufficient
2. The employee fills the stock

Alternative scenarios:

- Employee doesn't refill stock sufficiently
- The hardware is accidently damages or changed during refilling, causing an error

Use case 3: Processing orders and payments

Capability: The robot takes orders from the customer and asks to pay

Preconditions: The robot has the desired product(s) in inventory and the customer is able to pay

Postconditions: The robot starts product preparation

Scenario steps:

1. Customer interacts with the robot to see the product options
2. Customer interacts with the robot to choose and confirm selected products
3. Customer pays for the chosen product(s)
4. The chosen product(s) are confirmed and product preparation starts

Alternative scenarios:

- The customer cancels the order
- The customer is not able to pay (doesn't have NFC card)
- The order is cancelled because the payment can't be verified due to a software error
- The chosen products are out of stock and the order is cancelled by the robot

Use case 4: Preparing and presenting orders

Capability: The robot is able to prepare the different products and present them to the customer in a convenient way

Preconditions: The products chosen by the customer are confirmed upon payment and need to be prepared by the robot

Postconditions: The robot has prepared the chosen products and presents them to the customer

Scenario steps: This use case has three separate preparation steps: depending on the choice of the customer:

1. The user chose a hot drink
    a. The robot takes a cup
    b. The cup is placed at the coffee dispense location
    c. The robot signals the coffee subsystem to fill up the cup with the selected drink
    d. The robot transports the cup to the presentation location
2. The user chose a cold drink
    a. The robot signals the cold drink subsystem to dispense the selected drink
    b. The robot transports the dispensed drink to the presentation location
3. The user chose a snack
    a. The robot signals the snack dispense subsystem to dispense the selected snack
    b. The robot transports the snack to the presentation location
4. The robot presents the selected product(s) to the user

Alternative scenarios:

- An error occurs in the preparation, letting it fail
- The drink is spilled during transport in the robot
- The product isn't presented successfully to the user

Use case 5: Erroneous product preparation

Capability: The robot is able to detect an internal error during the preparation, transporting or presentation of the products.

Preconditions: The preparation of drinks is about to start

Postconditions: Something went wrong while preparing the order or delivering it

Scenario steps:

1. An error occurs while preparation, transporting or presentation of the products
2. The robot notices the error
3. The robot stops its action
4. The user and/or Science Centre employee are made aware of the error

Alternative scenarios:

- The system does not/cannot detect the occurred error
- The system detects a false error (there is no error, yet the system reports one or more)

Use case 6: Cleaning of system

Capability: All the robots' systems are able to be cleaned, including external surfaces

Preconditions: The robot is turned off and in need of cleaning (note: the cooling elements are not turned of).

Postconditions: The robot is clean and ready to be turned on again.

Scenario steps:

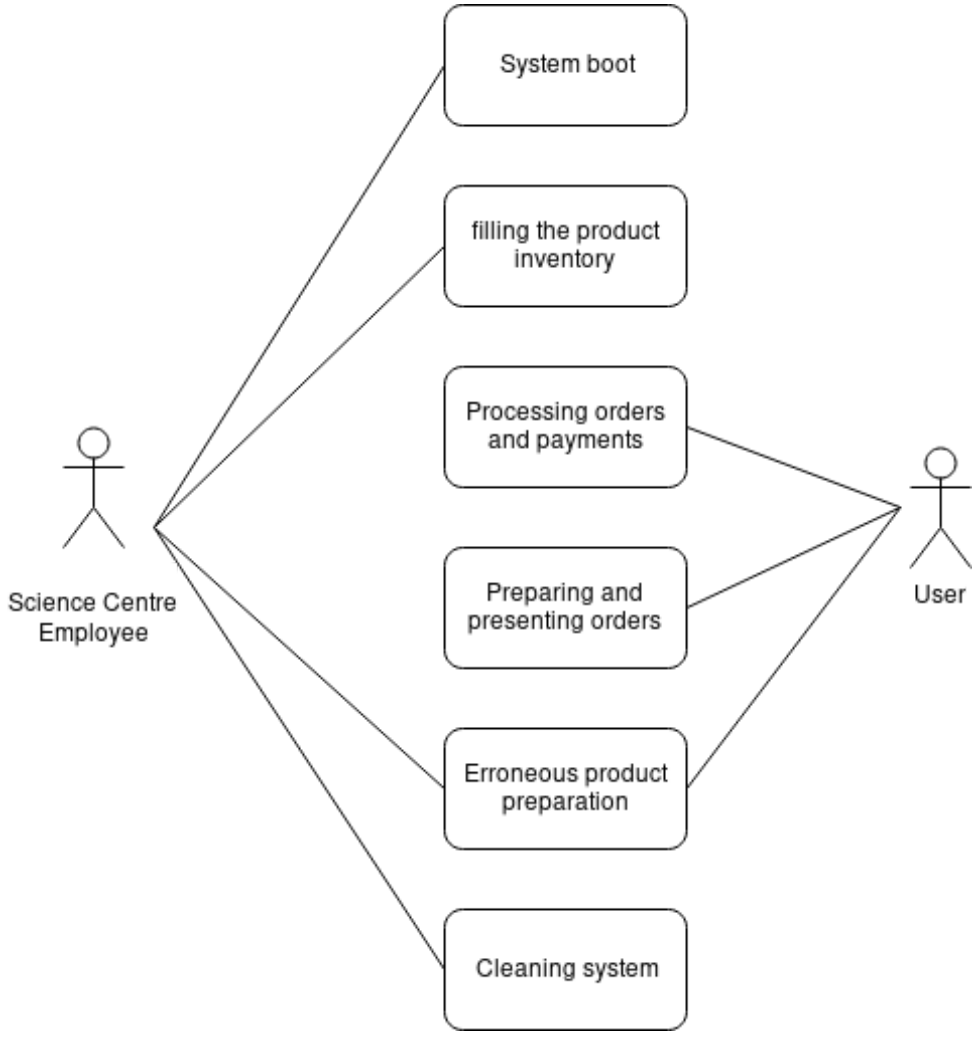
1. An employee opens the robot
2. The employee cleans the necessary subsystems of the robot
3. The employee closes the robot
4. The robot is turned on and the system boots up

Alternative scenarios:

- The outside of the robot is cleaned without opening the robot
- An internal component is removed to be cleaned outside the robot, after which it is placed back again
- An internal component is removed and replaced by an new part

1.2 Use cases diagram

In the image below, a scheme is shown that depicts what users can interact with the machine. It also displays the processes that the user has interaction with. For example, when the system gets stuck, a Science Centre employee is most likely called to resolve the problem.



1.3 Operational needs

The use cases outlined above give rise to properties that the system needs to comply to. These properties can be both humane in nature, logistical, technical, safety. They all concern *usability* of the system in the specific use case they belong to. Thus, operational needs are defined for every use case individually.

Use case 1: System boot up

Operational needs:

- Get power
- One button press should suffice

- Automatic start of all software systems
- Hardware self check after software load

Use case 2: Getting product inventory filled
Operational needs:
- Know the current inventory status
- Should not require technical knowledge of the subsystems
- Possibility to fill inventory
- Possibility to remove inventory

Use case 3: Processing and recording orders
Operational needs:
- Present different orders to choose
- Save chosen order
- Conclude payment
- Confirm order
- Decline order

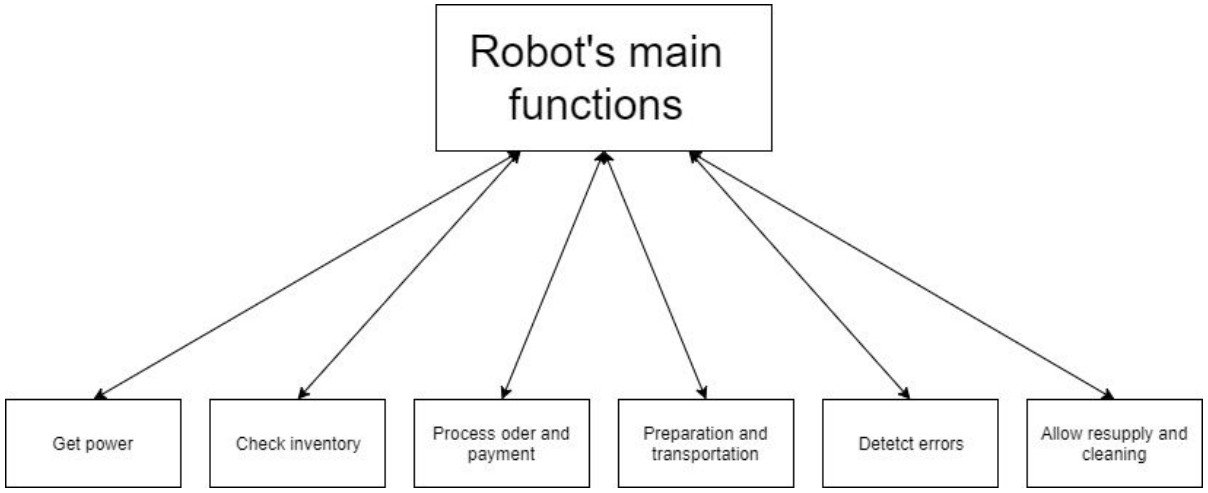
Use case 4: Preparing and presenting orders
Operational needs:
- Autonomous coffee preparation
- Autonomous cold drink preparation
- Autonomous snack preparation
- Move order to presenting area
- Present order to the user

Use case 5: Erroneous product preparation
Operational needs:
- Be able to stop any ongoing processes
- Show error message or indicator lights

Use case 6: Cleaning of system
Operational needs:
- Ability to easily clean necessary subsystems

1.4 Main functional capabilities

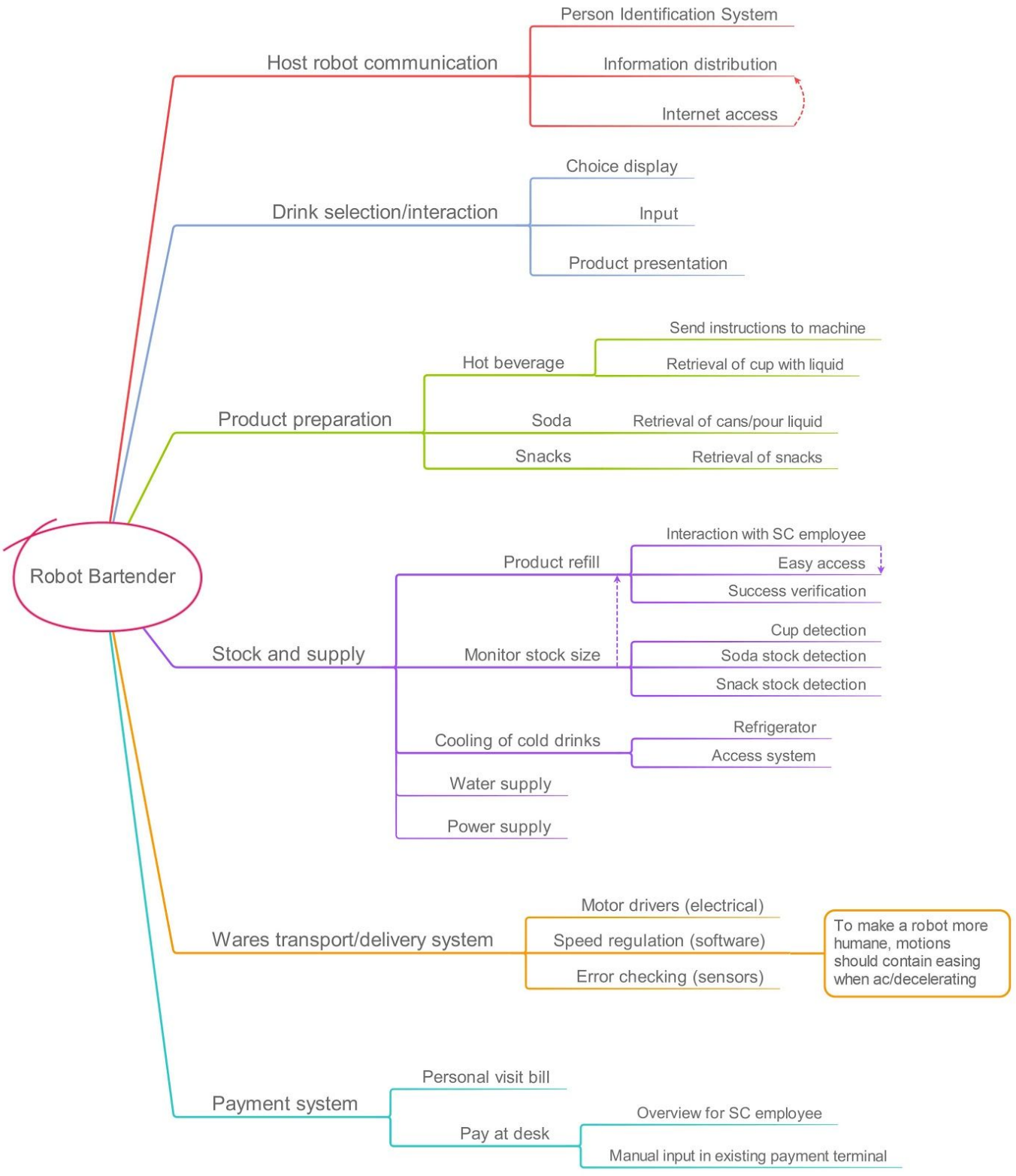
The following figure shows the main functional capabilities of the system.



2. System level

2.1 System Requirements

To ensure that all operational needs are properly satisfied, system requirements are composed. This serves the purpose of establishing clear specifications the design should meet. A diagram of all systems and subsystems is used to outline which (sets of) requirements



exist for different parts of the system.

2.2 Functional Requirements

The system needs to prepare hot drinks.
- The system needs a machine-controlled coffee preparation machine.
- The system needs an automatic coffee cup dispenser.

The system needs to dispense cold drinks.
- The system needs a cooling machine that can dispense single units.

The system needs to dispense snacks.
- The system needs a snack storage that can dispense single units.

The system needs to deliver orders to the customer.
- The system needs a delivery system from the dispense system to the presentation system.

The system needs to present choices to the user.
- The system needs a display that shows choices and current selection.
- The systems needs an input method to make a selection between the shown options.

The system needs to process and record orders to allow for payment.
- The system needs a NFC reader to read customers' NFC tags.
- The system needs a connection to the internal network of the Science Centre, using ethernet or WiFi, in order to record orders in the central database.

2.3 Non-functional Requirements

Performance:
The impact on speed of drink preparation should not be significant.

Safety:
The robots needs to have as much components as possible shielded from touch by users.

Fun:
The system needs to improve the experience of ordering a drink.

Easy to maintain:
The system needs to be easy to open up to clean the internal subsystems.

Contained:
The system needs to be self-contained in a single piece of furniture.

2.4 Tentative concept

We will now allocate the functions in the functional architecture to physical devices, using concept 2 of Team Assignment 1. We have augmented the diagram shown in section 2.1: System requirements to include some physical devices. Of course, since not all solutions are known yet, the model is still incomplete. These, at present, are merely guidelines which have been set up for the system to be developed.