Software Architecture Report

Team: FT_3
Authors:

- Bart de Man
- Felix Barten
- Joel James Bartholomew
- Spiros Tzavaras

Table of Contents

Stakeholders

In order to find the necessary requirements and functionalities we have composed a list of stakeholders in our project. We've decided to split the stakeholders into small groups for better traceability.All the internal company stakeholders have the prefix QC. Our stakeholders for this project are:

(Less-abled) product users

The end users of our product are people who are physically or mentally less able to perform household tasks. The goal of our product is to improve the Quality of Life of our customers. People who can still live at home instead of a full care nursing facility but have difficulties with performing certain (household) tasks. Our product should be able to alleviate the amount of work a resident has to perform or even reduce the amount of visits required per week from outside help.

- The robot should perform in a reasonable capacity
- Usability
- Reliability
- It should be easy for them to learn robot's functionalities
- Availability
- Security
- The robot must not hurt the product user.
- They want a robot made of durable materials (Can survive falling, is water resistant)

Relative or family member

A relative or family member is a stakeholder in our project because these people are likely responsible for interacting with the robot when it is initially configured (when the user receives one most likely). These relatives should be able to interact with the robot without much prior knowledge and should be able to customize the robot's settings

- Usability
- Modifiability of behaviour

QC: Quality Assurance Team

Quality assurance will be involved with the whole development of the product to ensure the product adheres to the requirements and to test and improve the entire development process.

- Testability
- Well defined requirements
- Making a better product

QC: Software Testers

This group of stakeholders might have insights to make sure the software of the product is and remains testable.

- Clear details about QC-Robot's functionality and design should be provided
- Testability (for the whole product and for the separate modules)
- Interoperability with other systems

QC: Hardware Testers

This group of stakeholders might have insights to make sure the product is and remains testable .

- Clear details about QC-Robot's functionality and design should be provided
- Testability

QC: Developers

The developers will be responsible for creating the software behind the robot. The developers are a stakeholder for this project because they would like a well maintainable software architecture which can be maintained and modified in the future without major overhauls.

- Interoperability with other systems
- Platform compatibility
- Memory utilization

- Modifiability
- Maintainability

QC: Researchers

They will be conducting the research that will clarify uncertainties surrounding the product.

- Clearly specified requirements
- Subjects for research

QC: Hardware Specialists

They will be responsible for the hardware designs needed by the product

- Clear specifications about the functions hardware should be able to perform

QC: Managers

Management wants a profitable product and to preserve the continuity of the company.

- Profitable product
- Better than similar products
- Preserve company’s status

QC: Investors

Investors are the reason our company is able to conduct research and build products. As Quality care is a small company we need a lot of outside funds to get our company off the ground. Investors have a large stake in our first commercial product as it will indicate if they will be rewarded for their investment in our company. The main concerns of the investors are:

- Schedule and budget estimation
- Requirements should be properly implemented

QC: Software Architects (FT_3)

They want a well-functioning system because they are responsible for the architecture

- Requirements traceability
- Consistency of architecture (parts should be combined successfully)

Doctor

They will administer the medicine required by the patient and track his health. They will be able to administer medicine and adjusting or adding prescriptions from their office using a web interface. Less need for the less-abled to visit, thus saving time and money.

- Usability

QC: Software maintainers

They will be maintaining the software for the product

- Interoperability with other systems
- Platform compatibility
- Memory utilization
- Maintainable system
- Modifiability
- Reusability

QC: Service engineers

Service engineers are a stakeholder in this project as they will be responsible for servicing the robots after launch of the project. The service engineers would like the robot to have easy to replace parts so the replacement of certain components is as easy as possible. The service engineer can use his/her experience to influence the development to make hardware that is easy to service. The service engineers’ main interests are:

- Remote analytics
- Modular Hardware
- Easy (dis)assembly of the product

- Clear maintenance instructions

QC: Customer Support

They will get called when any users run into difficulty using the product

- well documented instructions for troubleshooting
- clear documentation of QC-Robot

Health Insurance Providers

Health insurance is a main stakeholder for our product. Health insurance companies may be able to save money if the product is cheaper (and thus saves them money) than care provided by humans. This will allow medical personnel to perform less mundane tasks and focus on other aspects of care.

Health insurance companies might subsidize robots if they want people to transition to other types of care. This method would incentivize end users to buy the product if a part is refunded. Or alternatively the insurance company may offer the robot as a substitution of human care. Because of the position of the health insurance companies these stakeholders will be viewed as customers in this case.

- A robot is cost effective alternative to real personnel and they would like to make more accessible for users in need of help
- Availability
- Security
- Robot should be safe for public use (see Regulatory Authority)
- Robot should not harm patients
- Licensing

External Manufacturer for the Hardware

The manufacturer has to make sure the product can be produced once the designs are finalized.

- Clear specifications on the hardware needed for the product
- Feasible given design

Regulatory Authority (Nederlandse Zorgautoriteit(NZa))

The NZa is a stakeholder in this project because the product will be used in a medical fashion and can have effects on public health if it is not deemed safe. To make sure the product will comply with the latest rules and regulations the NZa is a definite stakeholder in this project.

- Product should be safe for use and should comply with all health care regulations
- Security

Suppliers/Distributor/Trader/Retailer/Wholesaler

The distributor will deliver the product to retail stores or directly to the customers

- Distributors want the product to be easily moveable without any special ways of transportation required

Third party developer

Third party developers will add additional behaviour to the robot's default behaviour. With the help of third party developers the robot could provide extra methods of care or more social interaction with users. Third party development is also a good way of gaining additional exposure for the product if a lot of interesting new functionalities are developed. However, these additions need to be curated otherwise the product will not be safe for the public anymore as external developers might not adhere to the health regulations which the product's programming is bound by.

- Modifiability of the code or rather good interfaces/API's to interact and extend the existing system.
- Interoperability with other systems

Future Stakeholders

Third party addition curator

The third party curator is a future stakeholder of our project. The 3rd party curator's job will entail checking packages to ensure the public's safety. This stakeholder is not listed in the current stakeholders because it is dependant on the adoption rate of the project. If the project is widely adopted after being completed it is likely more third party development will occur which will lead to more packages that need curation.

- Modifiability
- Testability
- Supportability

Requirements

After deciding on the stakeholders we have created a list of requirements in order to give our architecture more meaning. These requirements are split in functional and nonfunctional requirements. These were used to formulate the main functionalities.

Functional requirements

- Users should be able to customize how they like their meals
- Users should be able to customize the reminder settings
- It must be able to open doors and receive guests
- Unit must be able to notify the user that it is in need of servicing
- Unit must be able to analyze faults
- Unit must be able to send diagnostics back to service engineers for analysis
- Unit must have modular hardware
- Unit must be easily disassembled
- Unit must have a (maintenance) instruction manual
- Unit must have facial recognition
- Unit must be able to recognize voice commands
- Unit must be able to recognize hand gestures (sign language) commands
- Unit must be able to be controlled remotely via a remote.
- Unit must be able to be configured
- Unit must have a display to be able to accommodate deaf or near deaf people
- Unit must be able to respond to events 0-10 seconds
- Unit must be able to move around at a reasonable pace
- Unit must have a simple user interface
- Unit's behaviour can be modified from extra software defined by third parties
- Unit must be able to choose what the next task is that it will perform (scheduler/decision judgement)

- Unit must be able to show its status (either by voice or a graphical representation)
- Unit must be able to send and receive secure video from and to QC headquarters
- Unit must be able to store medical information about the user.
- Unit must be able to detect heart rate of the user
- Unit must be able to measure blood pressure of the user
- Unit must be able to take the user’s temperature
- Unit must have a bluetooth connection
- Unit must have a WiFi connection
- Unit must have a microphone
- Unit must have motion detection sensors
- Unit must have an inventory of available medicine
- Unit must be able to order additional medicine if the medicine runs out.
- QC Control Centre employees must be able to view a secure video from the robot
- The product must be able to monitor health information such as heartbeat and blood pressure

Non-Functional requirements

- Unit must comply with health and safety regulations
- A detailed description of unit’s functionalities
- Unit must be water resistant
- Unit must be assembled from components that are able to be tested individually
- Unit must have replaceable parts
- Unit must be able to withstand shocks and possible falls
- Parts of the unit must be able to be recycled
- There must be documentation to give information about the robot
- User must be able to get acquainted with the basic robot functionalities in less than 15 minutes
- Unit’s power consumption must be minimized where possible

Main Functionalities

After documenting the stakeholders of this project and its requirements, it was decided to create a list of functionalities for the robot. These functionalities are focussed on the end user and should all raise the quality of life for the user of this product.

1. Must be able to help the resident get back up if he/she has fallen down

2. Must be able to remind users of (important) subjects/appointments
3. Must be able to respond to a doorbell if the resident/end-user may not be physically able to respond to the door bell
4. Must be able to make meals and/or drinks available the resident may not be able to cook for him/herself
5. Must be able to clean the house using a built-in vacuum, thus removing the need for (extra) personnel to do so.
6. Must be able to administer medicine to the resident and keep the medicine stocked via prescriptions.
7. Your QC Robot should show (secure line) video in the QC control centre located in the (nearby) area.

Referenced Architectures

- JAUS reference architecture: uses a message passing protocol to provide interoperability
  among subsystems and components that compose systems resulting from this architecture
- ACROSET reference architecture: it is a component-oriented reference architecture for teleoperated service robots and its main characteristic is the reuse of components from different systems. This architecture is composed by subsystems such as UIS (User Interaction Subsystem)
- Servicebots reference architecture: it was designed to service robots in indoor environments and is composed by 3 subsystems, servicebots, fixbots and softbots. The type servicebots is a robot capable of driving autonomously in an average complex environment only with sensorial information. The fixbots present sensor and actuators distributed all over the environment, having their own intelligence, whereas the softbots refers to the software agents executing various tasks for the requesting user
- Robot teleoperation reference architecture: its main identified components are graphical representation, collision detection , user interface, communications and controller.

Below are provided some more links about reference architecture:

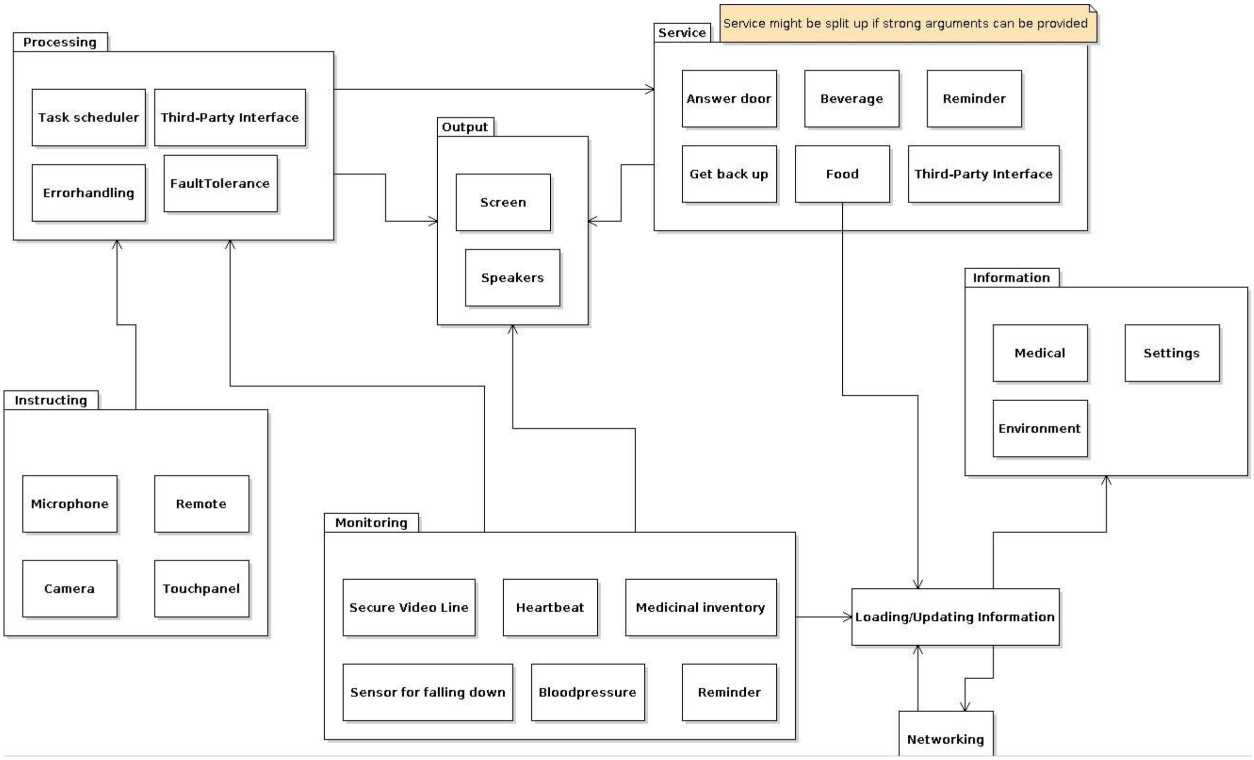
- http://www.mdpi.com/robotics/robotics-03-00181/article_deploy/html/images/robotics-03-00181-g007-1024.png
- http://research.cens.ucla.edu/projects/2005/NIMS/software_arch/Figure1.gif
- http://raweb.inria.fr/rapportsactivite/RA2009/lagadic/1.png

- http://www.eng.newcastle.edu.au/eecs/fyweb/Archives/2004/c2007000/images/softwareA.png
- http://cens.ucla.edu/projects/2005/Actuation/testbeds/Figure25.GIF

4+1 Architectural View Model

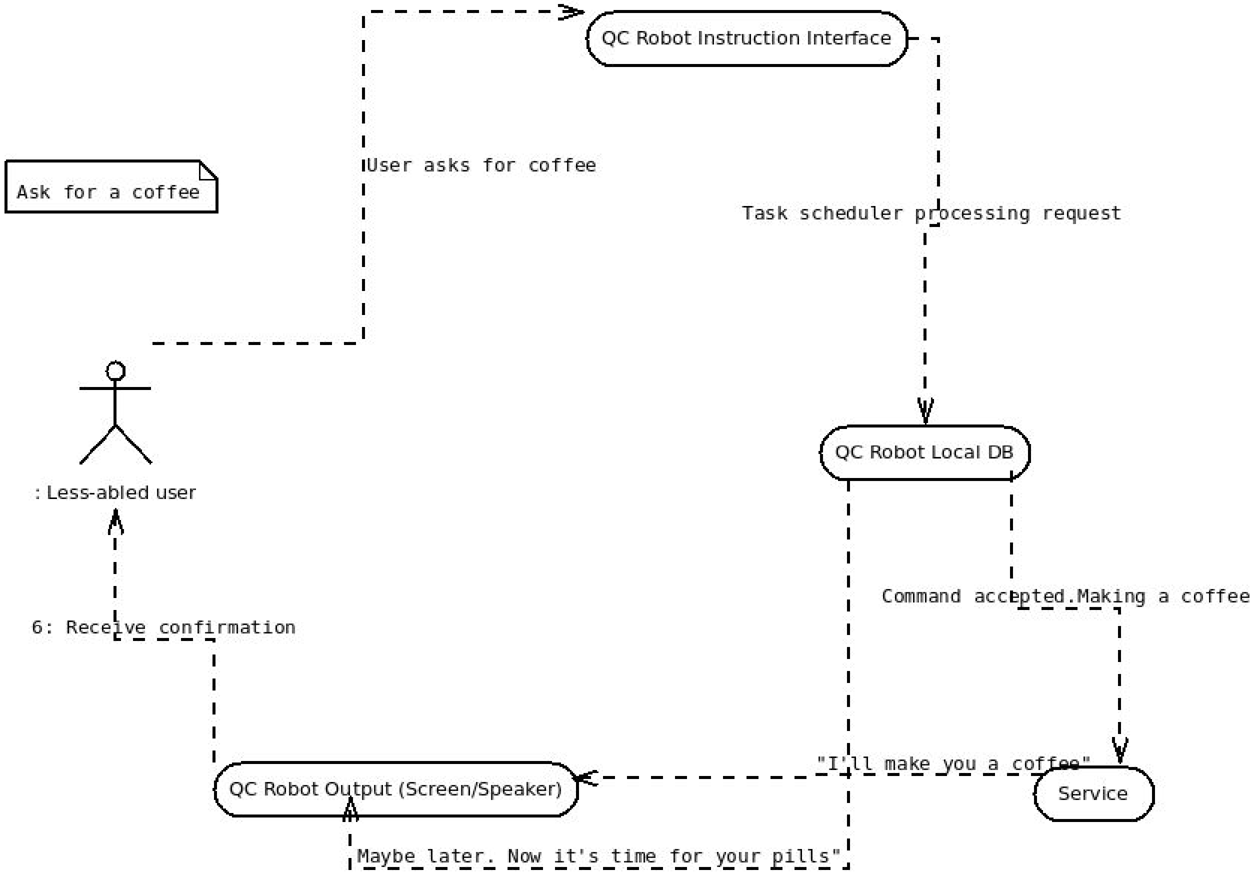
Logical view

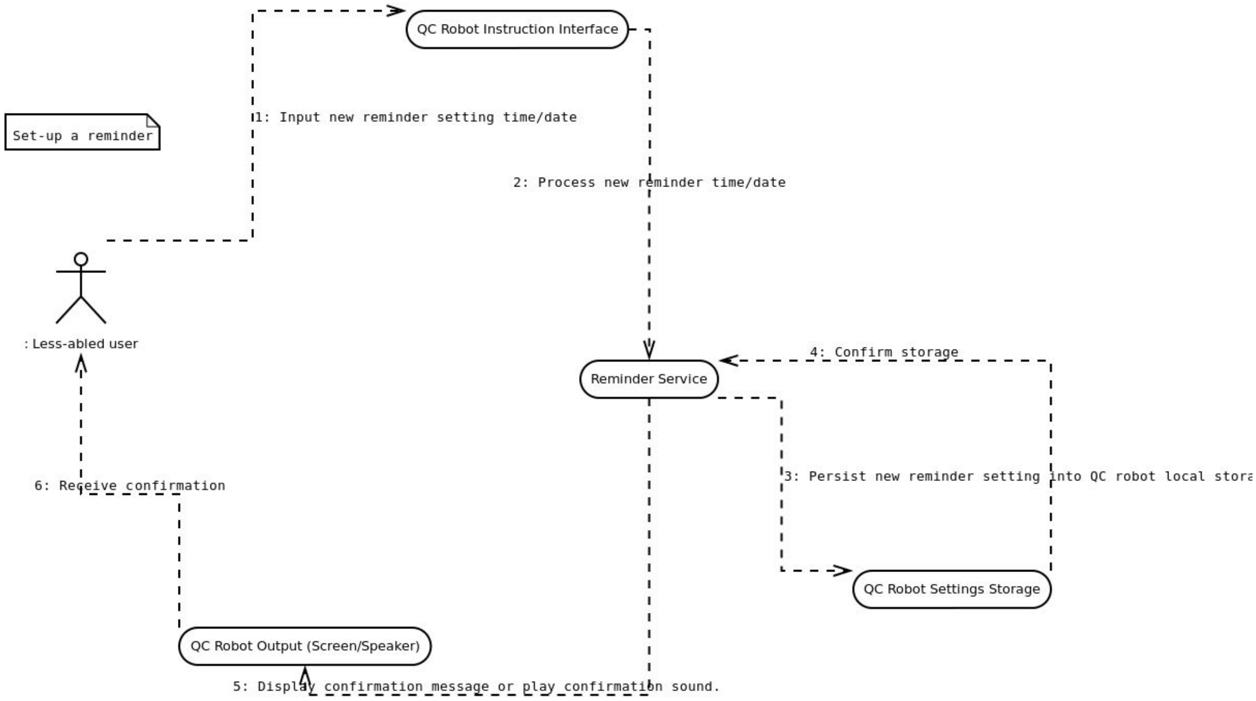
Below a class diagram is provided, that represents the functionalities that QC-Robot provided to the system. The logical view has been designed in such a way that the end user can understand the workings of the robot.

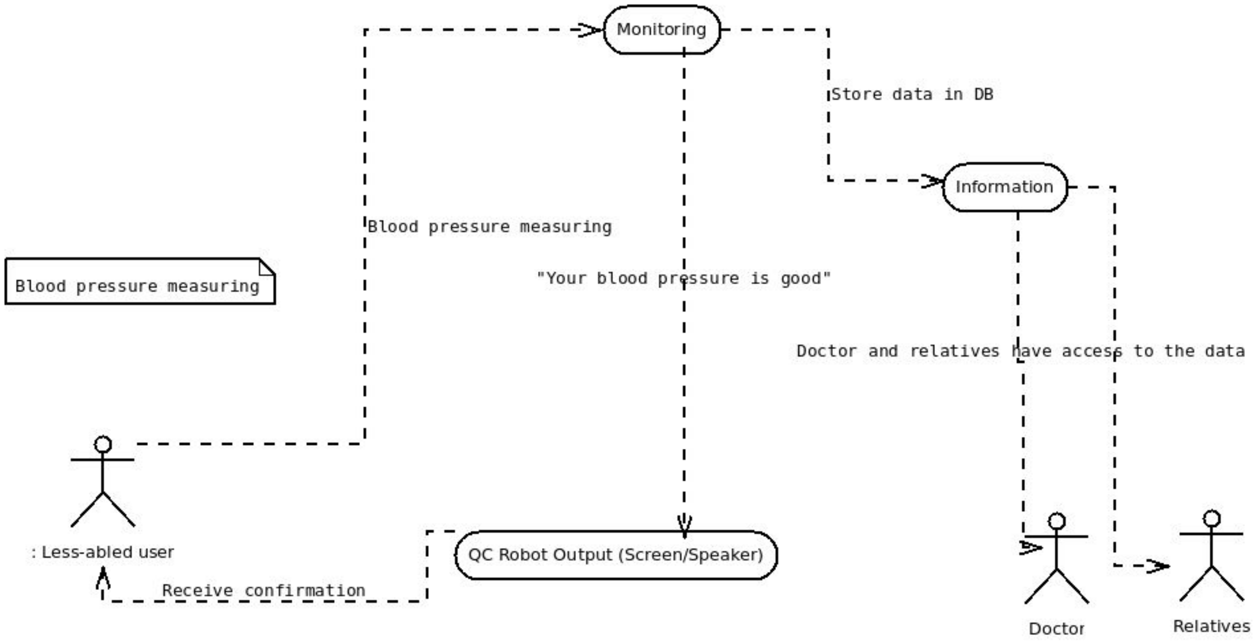

Service might be split up if strong arguments can be provided

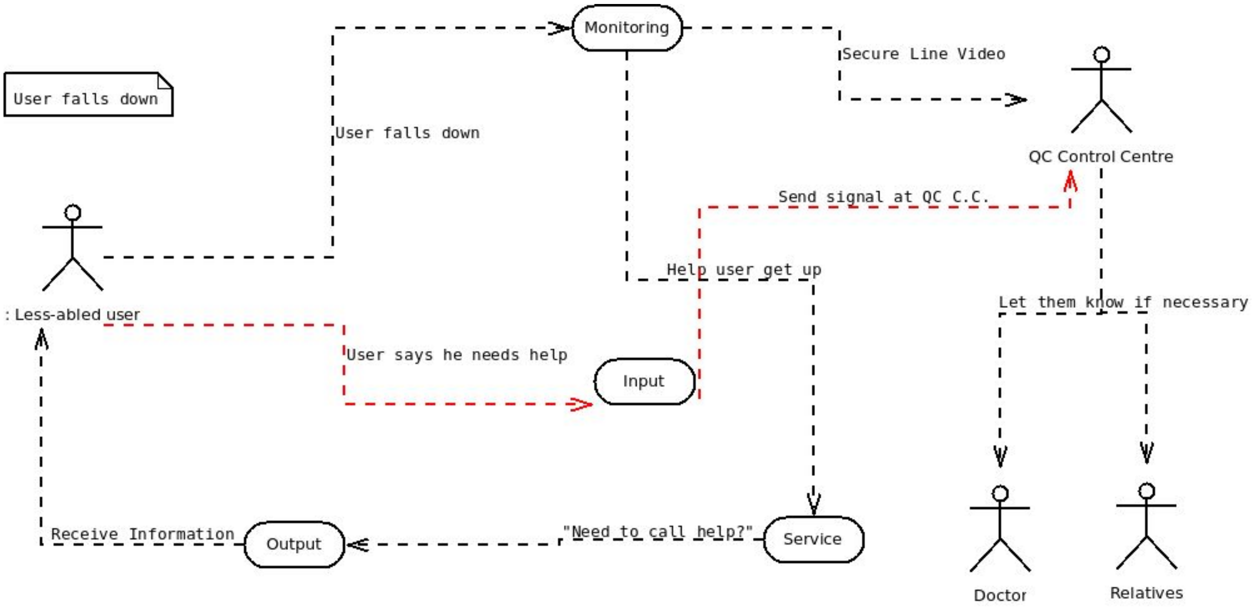
Process view

As for the process view, we decided it would be better if we created more than one diagram, to show how QC-Robot reacts in different occasions, and so the behavior of the system becomes easily understandable. We give examples such as: user asking for a coffee, user falling down and user setting up a reminder.


QC Robot Instruction Interface
Ask for a coffee
User asks for coffee
Task scheduler processing request
QC Robot Local DB
: Less-abled user
Command accepted.Making a coffee
6: Receive confirmation
"I'll make you a coffee"
QC Robot Output (Screen/Speaker)
Service
Maybe later. Now it's time for your pills"


Set-up a reminder
QC Robot Instruction Interface
1: Input new reminder setting time/date
2: Process new reminder time/date
: Less-abled user
4: Confirm storage
Reminder Service
3: Persist new reminder setting into QC robot local stora
6: Receive confirmation
QC Robot Settings Storage
QC Robot Output (Screen/Speaker)
5: Display confirmation message or play confirmation sound.


Monitoring
Store data in DB
Information
Blood pressure measuring
Blood pressure measuring
"Your blood pressure is good"
Doctor and relatives have access to the data
: Less-abled user
QC Robot Output (Screen/Speaker)
Receive confirmation
Doctor
Relatives


User falls down
Monitoring
Secure Line Video
QC Control Centre
User falls down
Send signal at QC C.C.
Help user get up
Let them know if necessary
: Less-abled user
User says he needs help
Input
Receive Information
Output
"Need to call help?"
Service
Doctor
Relatives

Deployment view

The diagram describes the software components of the system , as well as the way these are connected to each other.

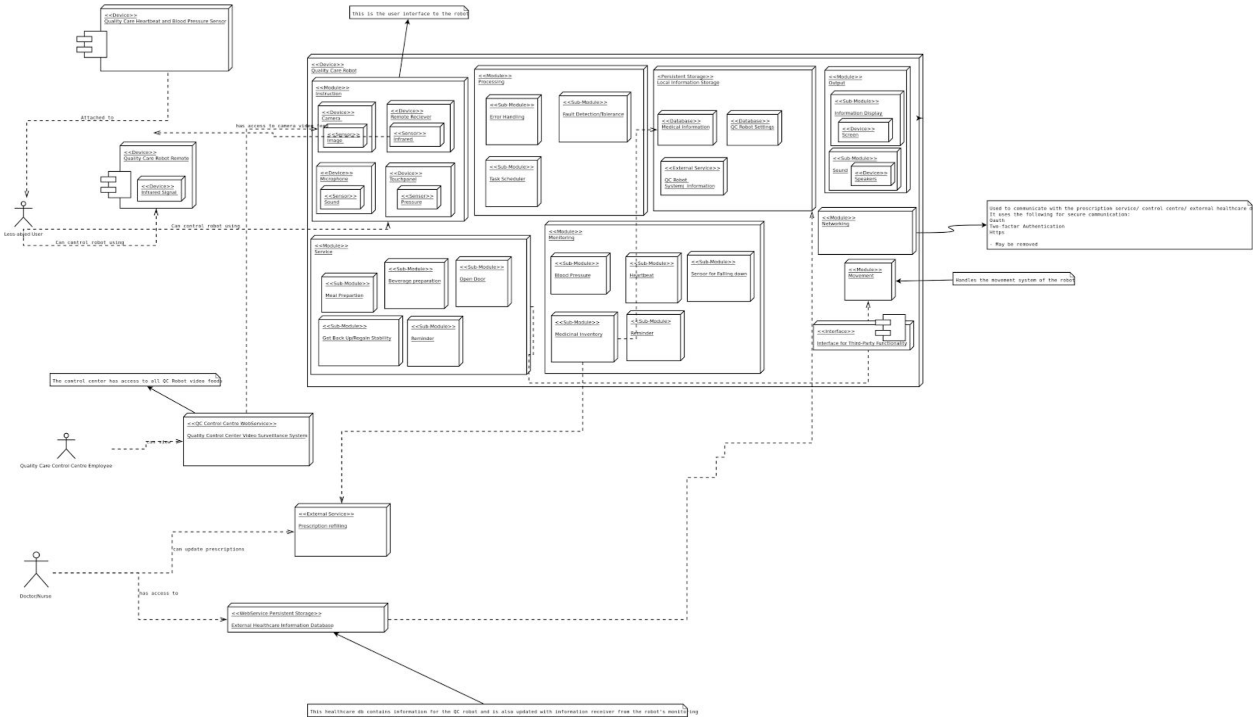
The deployment has the following main elements:

1. A QC sensor for monitoring heartbeat and blood pressure
2. A remote control for the QC robot
3. The QC robot itself
4. A web service which allows the robot to make its video feed available to the QC control centre

There are also two more external services which provide access to the health information of the primary user of the robot and another service which can be contacted to refill prescriptions.

The robot itself has the following software components:

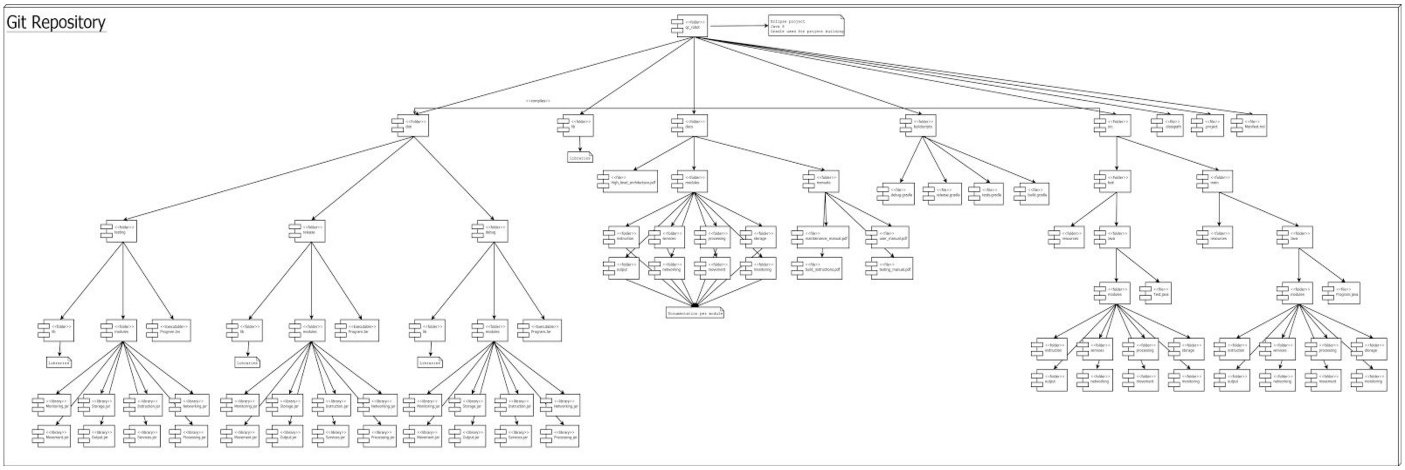
1. Instruction Module - The user interface which has several sensors which the user can use to input commands
2. Processing Module - This module validates the input provided by the instruction module and calls the service corresponding to the command issued
3. Service Module - Contains the main functionalities of the robot.
4. Local information Storage - Contains information relevant to the robot’s settings and user information
5. Monitoring Module - Used to monitor the product user’s health metrics
6. Networking Module - Allows the robot communicate over a network connections
7. Movement Module - Used by other modules to allow it to move
8. Third-party Interface - Can be used to extend the functionalities of the robot in a relatively easy way
9. Output Module - Used to provide feedback(information) to the user via the speakers and the display screen



View image on github

Implementation view

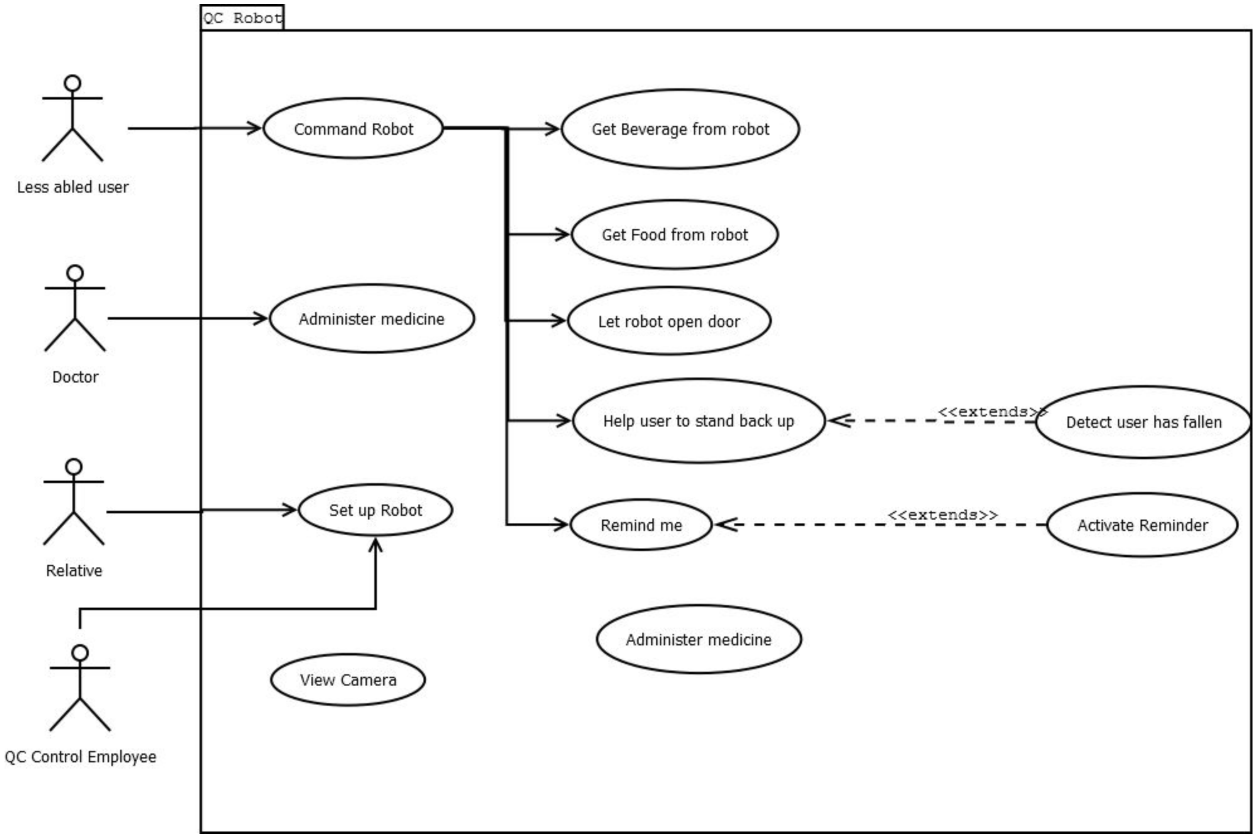
The structure of the code, the directories and files of the system, the environment and in general everything that concerns the programmers of the system, is represented in the following diagram.



View image on github

Use case view

Below we can see the interactions between the main stakeholders and the processes of QC-Robot. Two of the use-cases below are attached with an automated process which checks if actions are required.


QC Robot
Less abled user
Doctor
Relative
QC Control Employee
Command Robot
Get Beverage from robot
Get Food from robot
Let robot open door
Help user to stand back up
<<extends>>
Detect user has fallen
Remind me
<<extends>>
Activate Reminder
Administer medicine
Set up Robot
Administer medicine
View Camera