

Requirements Specification for Project 'Smartfridge'

by Melanie, Liuba, Nils, Jörn, Chris
based on [IEEE SRS Template](#)

1. Introduction

1.1 Purpose
1.2 Document Conventions
1.3 Intended Audience and Reading Suggestions
1.4 Product Scope
1.5 References

2. Overall Description

2.1 Product Perspective
2.2 Product Functions
2.3 User Classes and Characteristics
2.4 Operating Environment
2.5 Design and Implementation Constraints
2.6 User Documentation
2.7 Assumptions and Dependencies

3. External Interface Requirements

3.1 User Interfaces
3.2 Hardware Interfaces
3.3 Software Interfaces
3.4 Communications Interfaces

4. System Features

4.1 Taking a picture of the fridge interior
4.2 Detection of food aging process
4.3 Notification of critical food status
4.4 Tracking of entry and healthiness

5. Other Nonfunctional Requirements

5.1 Performance Requirements
5.2 Safety Requirements
5.3 Security Requirements
5.4 Software Quality Attributes
5.5 Business Rules

1. Introduction

1.1 Purpose

This document specifies the software requirements for the SmartFridge project (no release number yet). It describes the entire system.

~~##### 1.2 Document Conventions~~

1.3 Intended Audience and Reading Suggestions

This document is intended for the class of Systems and Software Engineering (WS 2017/2018) at University of Frankfurt. It should be read in whole as each section is relevant for the students' task.

1.4 Product Scope

A device that determines the freshness of food in a fridge. It can optionally be used to track and show the current fridge content. Since we aim to deliver a proof of concept prototype, the examined fruits and vegetables will be bananas and tomatoes initially.

1.5 References

2. Overall Description

2.1 Product Perspective

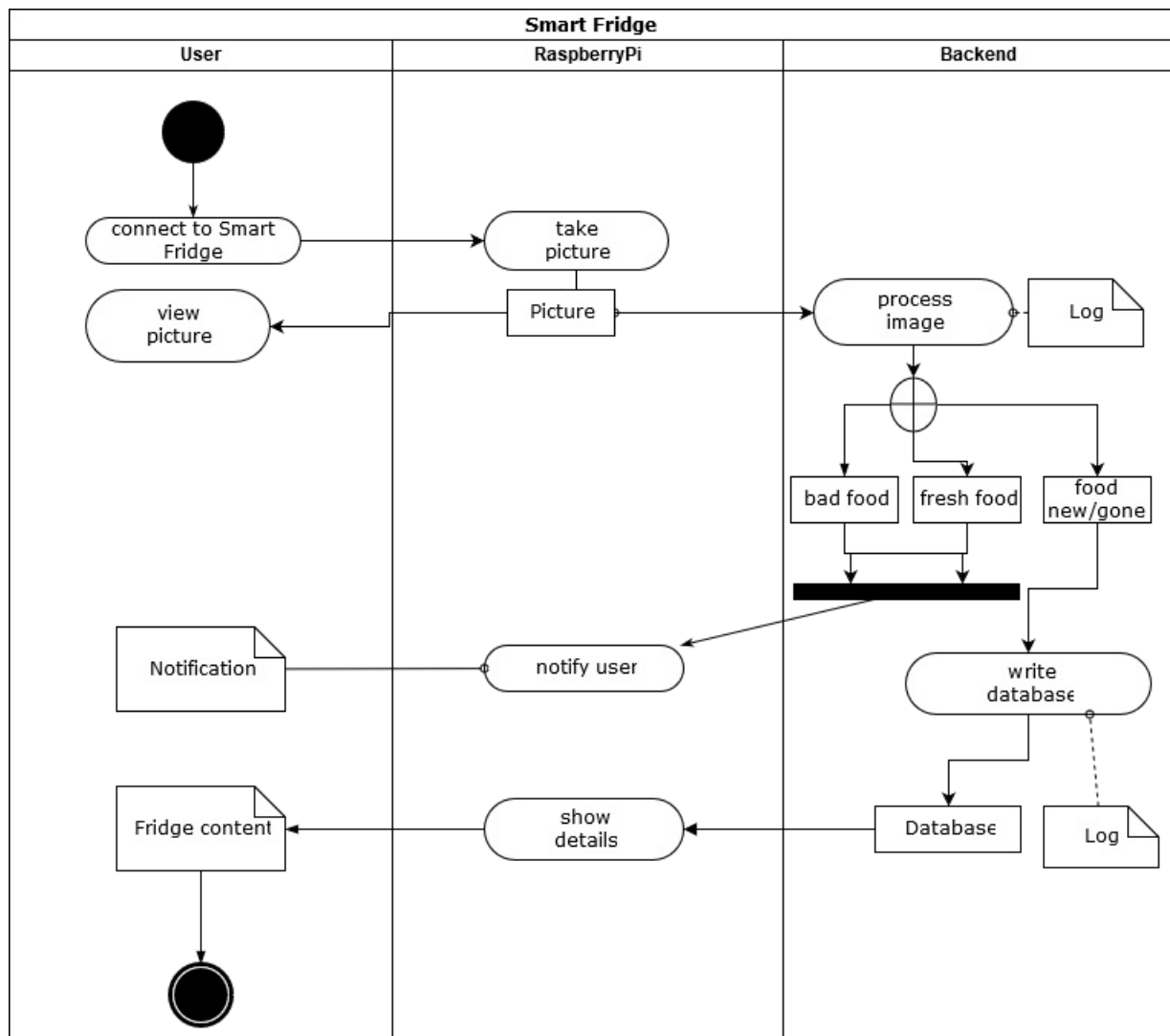
The described product is a university class project. It can serve as an add-on to fridges already equipped with "smart" technology like a touchpad and internet connection. It can also be used as a stand-alone product. It has a prototype nature and will not be ready to be shipped.

2.2 Product Functions

The system's basic functionalities will be:

- Tracking of freshness and edibility of fruits and vegetables within refrigerators via optical recognition of the food items' changing colour and shape.
- Reporting regarding food the current status of these food items via a web-based user interface.
 - Prediction of a 'best before date'
 - Statistical overview via one basic chart visualization.

Please take note of the following graphic for the concept.



2.3 User Classes and Characteristics

We strive for user-centric systems. Hence we elaborated several user groups that share the following attitudes:

- Early adopters. (Technology-savvy and curious users...)
 - are open-minded and willing to try out unfamiliar products
 - are likely to provide valuable feedback on functionalities that might be improved or added
 - appreciate the new product experience as individual benefit
 - User group importance: high
- Consciousness about food consumption. Users...
 - want to have an detailed and exact overview of their food consumption.

- care about food not being wasted
 - are most likely to be a long-time user if they are satisfied
 - benefit the product provides: logs of food consumption
 - User group importance: high
- Housewives / homemakers
 - are in charge of grocery shopping
 - like to show off new kitchen equipment to peers (marketing)
 - benefit the product provides: notifies/reminds on what food needs to be bought
 - User group importance: medium

2.4 Operating Environment

The software will run on a Raspberry Pi 3 Model B with a 1.2GHz Quad Core ARM Cortex-A53, 1 GB LPDDR2 RAM and a WLAN module. Its operating system is Raspbian Stretch (Kernel version 4.9) currently accessible [here](#) and installed on a 16GB SD card.

Attached to it is a camera module with a 5MP sensor that is able to take pictures with a resolution of 2592 x 1944 (4:3).

A power bank is used for energy supply.

2.5 Design and Implementation Constraints

- the RaspberryPi's limited RAM and CPU power might hamper the image processing
- the knowledge of used programming language(s) might be insufficient
- the camera module has no auto focus
- the inside of the fridge is usually not illuminated while the fridge is closed
- putting the Pi into the fridge for a longer period will be harmful due to humidity and temperature

2.6 User Documentation

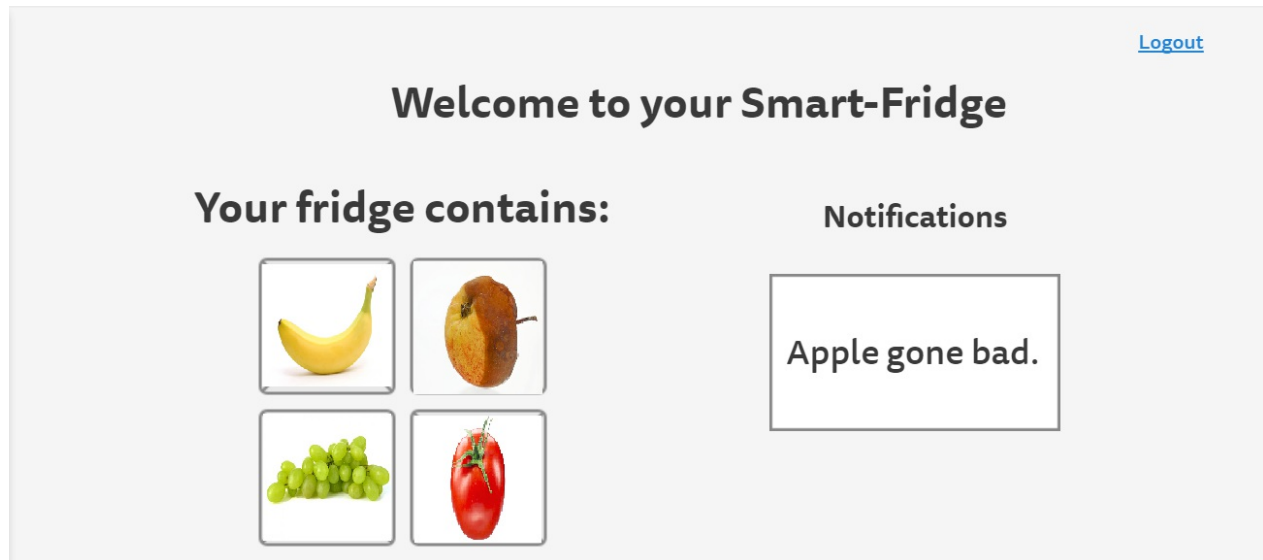
Currently no user documentation is planned.

2.7 Assumptions and Dependencies

3. External Interface Requirements

3.1 User Interfaces

The web-based user interface enables the user to view the content of his refrigerator shelf via a browser. Moreover text notifications and one chart visualize the fruit's edibility at first sight.



3.2 Hardware Interfaces

3.3 Software Interfaces

The picture recognition and freshness prediction software is accessible via a web-based user interface. Hence it is feasible to host this website on a local web server on the device.

3.4 Communications Interfaces

The Raspberry Pi is equipped with a WLAN interface. Thus it is able to offer the web interface provided by a local web server service via a local wifi network. In order to provide a high radio range, the Raspberry Pi might be connected to a local access point. Alternatively it could also be configured as access point itself and deliver a one-to-one connection with the end user device, such as a smartphone, or a laptop.

4. System Features

4.1 Taking a picture of the fridge interior

4.1.1 Description and Priority

The camera module placed inside the fridge takes a picture of a single shelf and everything on it.
Priority: high

4.1.2 Stimulus/Response Sequences

- User puts fruits in the fridge
- User activates the smartfridge-software

4.1.3 Functional Requirements

REQ-1:

REQ-2:

4.2 Detection of food aging process

4.2.1 Description and Priority

The pictures of the food are categorized by their state of freshness. If there are not enough current pictures, it updates this information to the website.
Priority: high

4.2.2 Stimulus/Response Sequences

- The images, taken in 4.1, trigger this process.

4.2.3 Functional Requirements

The pictures get analysed by the software. Therefore the software must be able to analyse and store images.

4.3 Notification of critical food status

4.3.1 Description and Priority

If the food has matured significantly the user gets alerted.
Priority: medium

4.3.2 Stimulus/Response Sequences

- The outcomes of 4.2 trigger this event.

4.3.3 Functional Requirements

The notification system (or the website) must be implemented.

4.4 Tracking of entry and healthiness

4.4.1 Description and Priority

The residence time of the food on the shelf will be tracked and its healthiness will be determined.
Priority: medium

4.4.2 Stimulus/Response Sequences

- Event triggered by the analysis in 4.2, if the pictures show the food has reached a tipping point the healthiness-state will be updated

4.4.3 Functional Requirements

A Database about different states of freshness must be accessible.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

5.2 Safety Requirements

One requirement would be to guarantee the safety of the hardware that needs to be put in the fridge, it must not get too hot, to minimize the energy consumption of the fridge.

5.3 Security Requirements

The data regarding the fridge content must be only accessible by the fridge owner.

5.4 Software Quality Attributes

5.5 Business Rules

6. Other Requirements

Appendix A: Glossary

Appendix B: Analysis Models

Appendix C: To Be Determined List
