Requirements Specification for Project 'Smartfridge'

by Melanie, Liuba, Nils, Jörn, Chris
based on IEEE SRS Template

Table of contents

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 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.

1.3 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.

2. Overall Description

2.1 Product Perspective

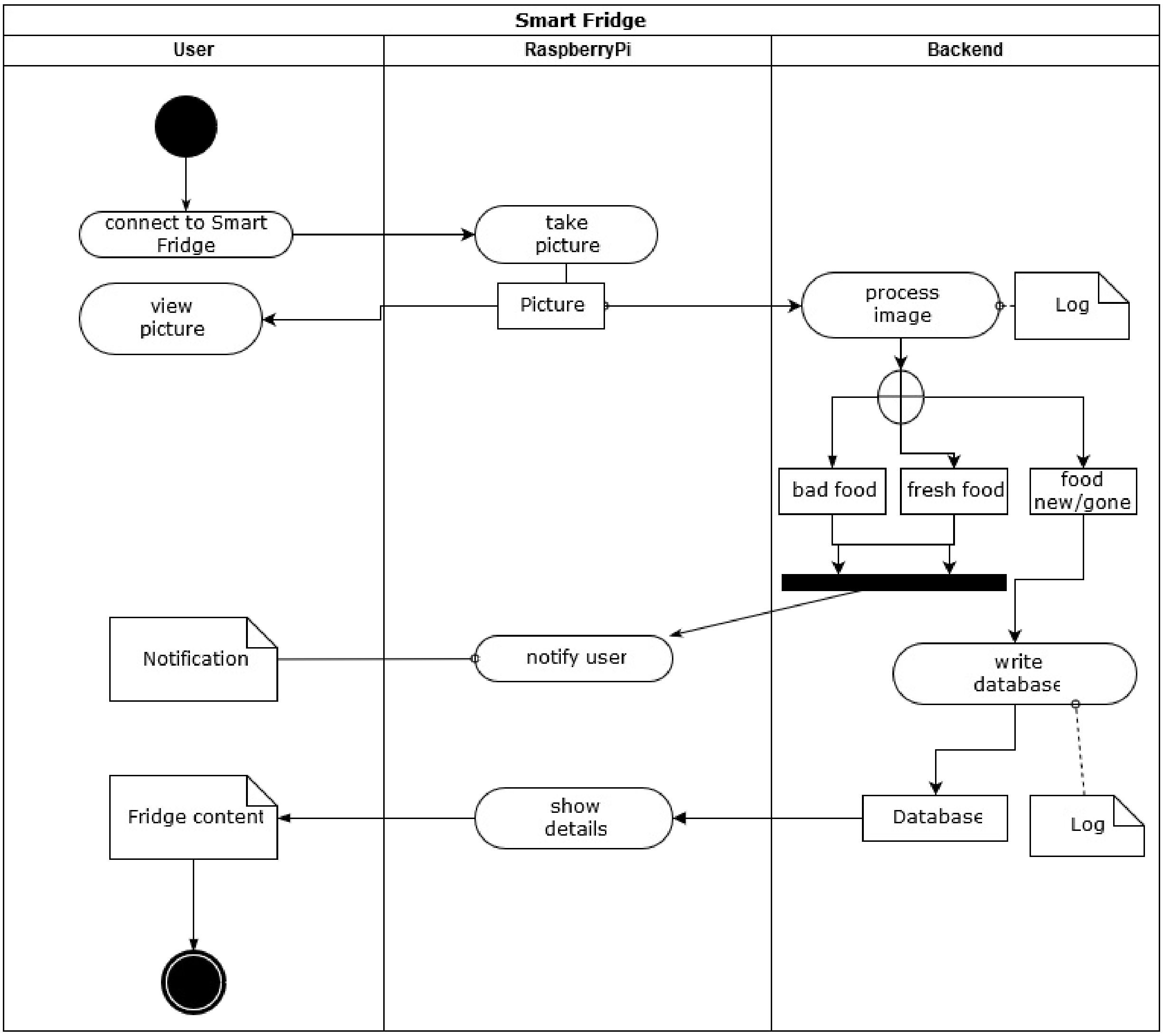
The described product is a university class project. It can serve as an add-on to fridges already equipped with "smart" technology like a touchpad and internet connection. It can also be used as a stand-alone product. It has a prototype nature and will not be ready to be shipped.

2.2 Product Functions

The system's basic functionalities will be:

- Tracking of freshness and edibility of fruits and vegetables within refridgerators via optical recognition of the food items' changing colour and shape.
- Reporting regarding 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 adaptors. (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 detailled 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
- Housewifes / 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.

The hardware will operate within the fridge to reduce the overhead of cabling.

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. We aim to build a user interface that is user friendly enough to be self-explanatory.

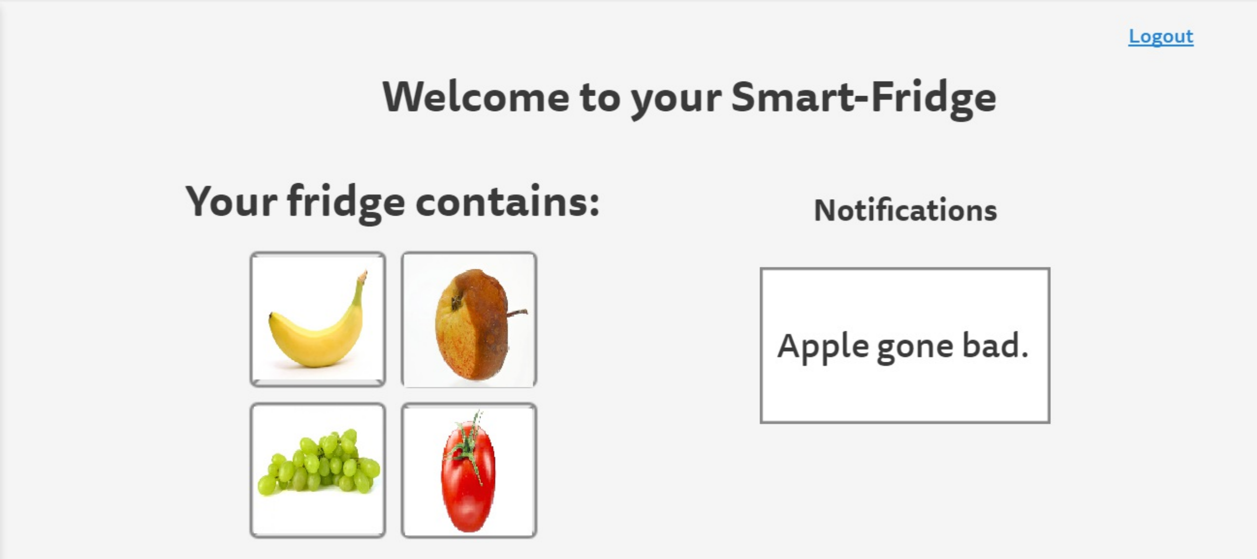
2.7 Assumptions and Dependencies

It is assumed that some open-source machine learing libraries and packages are available to facilitate the development and coding process.

3. External Interface Requirements

3.1 User Interfaces

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



Welcome to your Smart-Fridge

Your fridge contains:

Notifications

Apple gone bad.

3.2 Hardware Interfaces

The camera is attached to the CSI-2 (Camera Serial Interface Type 2) of the Raspberry Pi via cable. The powerbank is plugged in to the micro-USB port of the Raspberry Pi. Output will be displayed over wifi on user devices with a browser.

3.3 Software Interfaces

The backbone of our system will be a picture recognition and freshness prediction software. The output regarding status and predictions will be accessible via a web-based user interface. Hence, for the sake of efficiency it is feasible to host this website on a local web service within the Rasperry Pi along with our analytics software.

3.4 Communications Interfaces

The Raspberry Pi is equipped with a wifi interface. Thus it is able to offer the web interface provided by a local web server service via a local wifi connection. In order to provide a high radio accessibility 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 laptop computer.

4. System Features

4.1 Data aquisition and storage

4.1.1 Description and Priority

The camera module placed inside the fridge takes a picture of food items on one shelf.

Priority: high

4.1.2 Stimulus/Response Sequences

- User puts fruits in the fridge
- User activates the smartfridge-softw are by accessing the user interface via w eb brow ser.

4.1.3 Functional Requirements

- REQ-1.1: Take picture w ithin fixed environment
- REQ-1.2: Store images durably
- REQ-1.3: Define Region-of-interest (ROI) for each image
- REQ-1.4: Provide access to images and metadata (e.g. timestamps, ID, type of fruit, ROI,…) to other processes
- REQ 1.5: Create timeseries of images for unique fruits

4.2 Detection and tracking 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 w ebsite.

- The analytics softw are must consist of components that can provide the follow ing tasks.
  - Taking pictures w ith the
  - Recognize the food items to be tracked.
  - Recognize the aging process w ith picture analytics techniques (w hich need to be further elaborated).
  - Predict the foods edibility.
  - Constantly improving the prediction process. The user must be able to give a simple feedback, if the predicted edibility deviates from its actual state of freshness.
- A Database about different states of freshness must be accessible.

Priority: high

4.2.2 Stimulus/Response Sequences

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

4.2.3 Functional Requirements

- REQ 2.1: Extract features from ROIs of timeseries of images
- REQ 2.2: Build aging models for different type of fruits
- REQ 2.3: Compute state of age for individual fruit stored
- REQ 2.4: Update model w ith user input (e.g. "still fresh", "not fresh anymore", …)
- REQ 2.5: Update notification database

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 w ebsite) must be implemented.

- REQ 3.1: Create database for fruits in fridge
- REQ 3.2: Watch database for updates
- REQ 3.3: Notify frontend (e.g. w ebsite, RSS-feed, …)

5. Other Nonfunctional Requirements

5.1 Performance Requirements

- REQ N1.1: Modular setup to facilitate separation of computing and data aquisition
- REQ N1.2: Reduce energy consumption to less than 25% of the energy consumption of the fridge
- REQ N1.3: Minimize exhaust heat that would increase the fridge cooling

5.2 Safety Requirements

- REQ N2.1: Prevent shortlinks within electronics in the fridge environment
- REQ N2.2: Prevent condensation within power supply

5.3 Security Requirements

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

- REQ N3.1: Prevent unauthorized access to stored data
- REQ N3.2: Prevent unauthorized access to computing hardware
- REQ N3.3: Prevent unauthorized access to connected networks and computers

5.4 Software Quality Attributes

The software must consume few enough resources to work on a system-on-chip. If this is unattainable, the software must be portable to a different environment. To prevent a bad user experience it also must deliver results quickly.

5.5 Business Rules

- REQ N5.1: User controls data storage and usage