Remote Security System

System Requirements Specification

DAT231 Requirements Engineering

Version 3

2015-10-19

Anna Arbman
Anton Rose
Dominik Einkemmer
Mathias Bylund
Sebastian Blomberg

Change history

Changes from R1 to R2

- Product description has changed
- Some feature requirements have been turned into use cases.
- New use cases regarding Account and Access management
- The tasks in the previous release have been removed and requirements regarding scheduling and notification have been turned into tasks.
- Virtual windows has been modified.
- Context diagram has been added.
- Prioritization has been added.
- Release Planning has been added.
- The document has been restructured.
- Quality requirements has been added.
- Definitions has been added.
- Data model is basically the same, but we added Keys and Log items, as well as a few other relations concerning them.

Changes from R2 to R3

- The System Requirements Specification has been reviewed by the customer and by us
- Changed style for quality requirements to planguage
- Grouped the quality requirements into four top-categories; security, maintenance, usability and efficiency requirements
- Rewritten all quality requirements
- Added what is excluded from our system in the scope
- The product functions section now includes an explanation of how new devices are added
- Added quality requirements for portability as wished from the customer
- Added references for some things in the glossary
- Added User-Goals
- Added (high) domain-level events to link user goals to specific requirements
- Moved description of prioritization methods into the appendix
- Added more description on the AHP prioritization method
- Changed and added Mock-Ups and included a link to the interactive Mock-Ups
- Added UC15. View a user's profile.
- Removed requirement FR.A1 and FR.A3, since they are now covered in UC15
- Added feature requirements FR.D22-24
- Added feature requirement FR.S7-10 to cover product-level requirements for scheduling tasks
- Converted FR.AR1 into FR.D25
- Removed Administrator from the data model and replaced it by a relation which shows for which households a user is the administrator of
- Added survey results as an appendix

Table of content

1 Introduction

Remote Security System is a smart-home system that allows users to remotely operate locks, home appliances, and other devices, as well as to be informed of conditions in a home, such as air quality and temperature, whether windows are open, or doors are locked. The system comprises of components, which possesses the ability to automate devices, read conditions in a household, and communicate amongst each other. However, the system, as of now, does not possess any user interface, nor does it have any implemented structure to support users to operate the system.

This documents is a software requirements specification for a mobile client application which will become a user interface to the Remote Security System. This introductory section contains descriptions of the overall purpose, scope, and definition of abbreviations and acronyms, as well as a sub-section describing the content and layout of this requirement specification.

1.1 Purpose

The purpose of this documents is to clearly and extensively describe the requirements for the mobile client application of the Remote Security System, as well as the constraints and intercommunication with external systems in a context of use. This documents is intended to describe the requirements of the client application in a way that is adequate for developers to use as a reference when implementing the application, but also understandable for non-developers.

1.2 Scope of the application

The Remote Security System is a mobile application that is an interface that enables users to use the Remote Security System. Since the hardware of the Remote Security System is already defined, it will be beyond the scope of this document to take hardware functionality into account, as well as the software that enables communication with the hardware components. The mobile application will also not include the functionality for finding a key card as stated in the project mission. Furthermore, it will also be beyond the scope to handle data storage, meaning that the data inside the Remote Security System will be stored on a remote location and the application will use an existing interface to fetch and modify that data. The product’s business goals, according to the customer, is to be integrated with the user’s smart home and provide the user with information about locked/unlocked doors and windows, switched on/off light sources and other electrical devices, as well as rooms’ temperatures and air quality. The application shall also allow the user to operate these devices from their mobile phones.

1.3 Definitions, acronyms, and abbreviations

This section describes commonly used terms in this report.

Term	Definition
Household	A house, apartment, or other living arrangement which has a Remote Security System installed.
Administrator	A user of the system who manages the Remote Security System in a certain household.
Resident	A user of the system who permanently lives in a household.
Guest	A user of the system who temporarily lives in or visits a household.
User	A person with an account who interacts with the product. This could be an Administrator, a Resident, or a Guest, unless stated otherwise.
Product owner	The company that shall develop and sell the product on an open market and has ordered the requirements specification.
Device	Any device that is connected to the product
Critical device	A device that can not be left unattended, e.g. stove, flatiron
NFC	Near Field Communication, a wireless method to transfer data over short distances. [11]
RFID	Radio-Frequency Identification, a wireless method that identify and tracks tags attached to objects [12]
REST	Representational state transfer, a hybrid software architectural style for distributed hypermedia systems. [10]
JSON	JavaScript Object Notation, a text format that stores data objects in a way that is readable by humans. [9]
⚙	This symbol next to a datafield in a virtual window means that the data field is editable
Bcrypt	Bcrypt is a key derivation function specifically developed for storing hashed passwords. [6]

HTTPS	An encrypted version of HTTP to ensure secure transactions. [7]
SSL	Denotes the predominant communication security protocol of the Internet particularly for World Wide Web. [6]
Blackbox	The Blackbox is the central control unit of the household which all the devices used in the Remote Security System are connected to. Every household with an installed system has one Blackbox. Furthermore, every Blackbox has its own unique serial number in order to identify which household it belongs to.
Activation code	An activation code is given to the administrator of a household in order to install a new system in the mobile application.

1.4 Overview

This report consists of five chapters. The first chapter gives an introduction to the report and the Remote Security System. The second chapter describes the perspective and functions of the Remote Security System, the user characteristics and the constraints of this project. The third chapter present all external interfaces, functional, data, quality, and design requirements of the application. The fourth chapter includes prioritization and release palling of the functional requirements. Lastly, the fifth chapter includes references and appendices.

The explicit requirements, specified with an ID, are not necessarily ordered in numerical sequences. It might be that there is a missing id, e.g. a sequence [1,2,3,4,6], if a requirement has been removed in other versions of this document. The reason for not re-numbering the requirements is to keep traceability between different versions of this document.

Explicit requirements are indicated by an id **XX.XX**

2 Overall description

This chapter aims to give an overview of the whole system; the product perspective, product functions, user characteristics and constraints of the product are explained.

2.1 Product perspective

The purpose of the Remote Security System is making lives of potential end users more comfortable by enabling them to control, as well as to monitor, electrical devices at home through an application. Furthermore, the Remote Security System will work as an extra security measure by enabling the users to lock unlocked doors and shutting of critical devices in a household when not at home. Described at a high level, the user goals of the Remote Security Systems are:

- UG1. to schedule operations of devices, such as lights, in cases when going away for a longer time
- UG2. to remotely see the state of a households
- UG3. to remotely operate devices in a households
- UG4. to be assisted in ensuring security of a household
- UG5. to simplifying key- and device-management

The mobile application, specified by this document, will be the user interface for the Remote Security System. It will communicate with central control units, Blackboxes, inside the households that has installed the Remote Security Systems. The application will allow users to manage, operate, and be informed of the state of their households and their devices.

2.2 Product functions

Through the Remote Security System application, a user shall be able to remotely control a devices in a household from a smartphone. A user shall be able to check statuses of devices in a household, such as door locks, whether windows are open, if light bulbs are switched on or off, as well as their light color and brightness levels, the temperature and air quality in different rooms, and if general electrical devices are on or off. A user shall also be able to change the statuses of the all the devices, except for the air quality and window sensors which are

The application shall enable marking of electrical devices as critical, and users shall receive a notification from the application if leaving the house and the critical device is turned on. The notification function shall also work for the door locks and notify a user when leaving home without locking.

Every household, in which the Remote Security System is installed, has a central control unit called a Blackbox. The Blackbox interacts all components of the system and has knowledge and information of every component. Therefore, the application will communicate with the Blackbox to operate and get information about the devices in a household. New devices are detected by the Blackbox after they are plugged in in the household, and thus they will automatically show up in the application.

This section further describes the components of the Remote Security System.

2.2.1 Locks

All locks are installed in exterior doors and consist of a RFID reader and a NFC reader. There are three ways of unlocking a lock. Firstly, a key card with an RFID tag can be used by placing the key card in close distance to the lock, and then entering a pin code on the hardware number pad of the lock for validation. These key cards are handed out by the product owner and can be managed through the application. Secondly, the application will allow virtual keys to be used with NFC. Thus, by holding an NFC enabled smartphone close to the lock, the phone will act as a physical key card, and then the user needs to enter a pin code for validation. Thirdly, a virtual key can be used directly in the application by pressing a button for unlocking a door via the application and then entering a pin code directly in the application. The same key (in the Remote Security System) can thus be used in three ways.

Locks also have two modes: *Normal Mode* and *Auto-Lock Mode*. The Normal Mode works as a regular lock. The lock is unlocked until someone locks it, and it is locked until someone unlocks it. However, Auto-Lock Mode keeps the door locked from the outside, but unlocked from the inside. Thus, users do not have to lock when they leave home.

2.2.2 Light bulbs

The light bulbs included in the Remote Security System can turned on and off, and be dimmed by setting a level of brightness. Furthermore, the light bulbs can change their projected light color.

2.2.3 Power switch adapter

A power switch adapter is device that enables remote power switching (on and off) of any high-voltage, e.g.110-240V, electrical device that can be plugged into a socket.

2.2.4 Temperature gauge/regulator

A unit which can measure the temperature of a room, as well as to adjust the temperature. This can be for example, an Air Condition unit.

2.2.5 Window Sensor

A sensor which detects if a window is opened or closed. This sensor cannot operate the window itself.

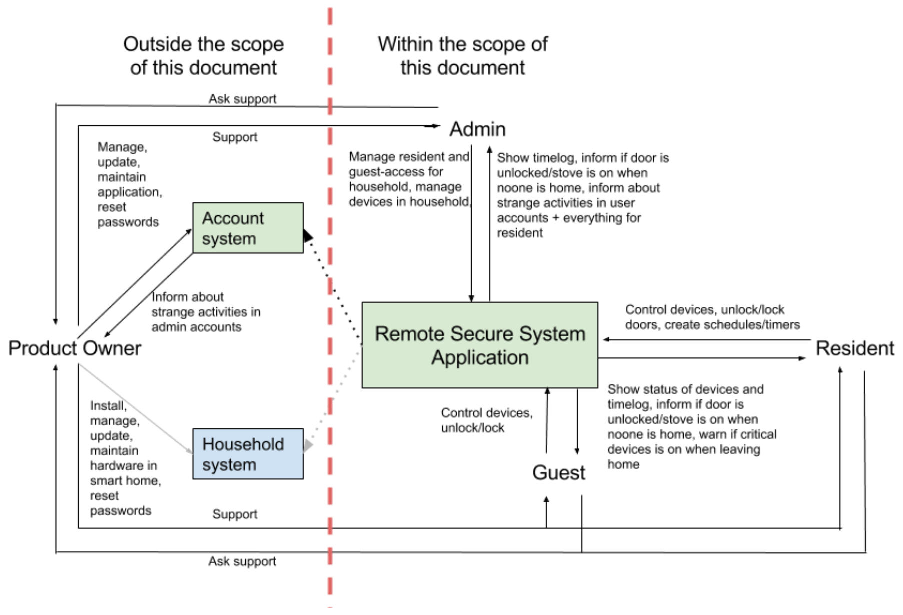
2.2.6 Air quality sensor

A sensor which measures the air quality of a room. The air quality is measured according the European standard CITEAIR [X]

2.3 User characteristics

Potential users of the Remote Security System are people that are interested in technical and futuristic solutions for their home. The application requires the users to have some experience with smartphone applications, but should not require a high level of technical expertise. The targeted audience for the Remote Security System are people from the middle and upper class society, who are able to afford the purchase and install the system.

2.4 Product context



Outside the scope of this document
Within the scope of this document
Ask support
Support
Admin
Manage, update, maintain application, reset passwords
Account system
Manage resident and guest-access for household, manage devices in household
Show timelog, inform if door is unlocked/stove is on when noone is home, inform about strange activities in user accounts + everything for resident
Inform about strange activities in admin accounts
Product Owner
Remote Secure System Application
Control devices, unlock/lock doors, create schedules/timers
Resident
Install, manage, update, maintain hardware in smart home, reset passwords
Household system
Control devices, unlock/lock
Show status of devices and timelog, inform if door is unlocked/stove is on when noone is home, warn if critical devices is on when leaving home
Guest
Support
Ask support

Fig. 1 - Context diagram of the Remote Security System, with emphasis on the user application

As illustrated by the context diagram in figure 1, the Remote Security System application communicates mainly with two other systems: the account system and the household system. The household system directly handles the hardware devices and has access to control and read all electrical devices in a household. Furthermore, the household system is connected to the internet and enables, through an interface it provides, remote controlling of the electrical devices the household system has access to. The household system is provided by the product owner, hence it is not part of the scope of this document. The application, specified in this document, will communicate with a household system and expose a graphical interface to control its devices. There is one instance of a household system, i.e. a Blackbox, for every household. Moreover, the product owner has their own interface to communicate with a household system, in cases where the product owner needs to intervene. However, this is also beyond the scope of the user application.

The account system handles the credentials of the users using the system, i.e. it handles access rights, user login sessions, etc. The user application will expose a graphical user interface for users to create and maintain their accounts in the system. Furthermore, it will use the account system to enforce access rights to different households and devices in said households. The account system also exposes an interface for the product owner to be informed if there are any strange activities surrounding the user account, e.g. if someone is trying to hack the account. The interface also gives the product owner full authority to modify any account details, disable accounts, or remove accounts. However, similarly to the household system, the the account system, including the product owner’s interface, is provided by the product owner, and specifications for it is beyond the scope of this document.

Figure 1 also shows the different user roles of the application, and their respective interfaces. Every household has one administrator who handles access for other users in the household (residents and guests). The administrator also manages devices in a household and grants keys to users. Moreover, the admin is informed of strange user behavior if such should arise, and is able to see a time log of user interaction for every lock in the household. All users can operate and show statuses of devices in a household. However, only residents and administrators can create schedules for operating the devices. All users can can interact with the product owner outside the application for support.

3 Specific requirements

This chapter contains requirements that give a detailed description of the system. All interfaces used by the application are presented and explained to put the application in context. The application is then explained through data requirements, functional requirements at different abstraction levels, quality requirements, and design requirements.

3.1 External interfaces requirements

This section presents all inputs and outputs to the system.

3.1.1 Software interfaces

The application will be developed for mobile operating systems and requires at least Android v4.0, iOS 7 or Windows Phone 7.8.

As explained in section 2.4 Product context, the application will function as a user interface towards the Remote Security System. It is assumed that the Remote Security System provides a central cloud solution which implements all desired functionality for managing accounts in the system, as well as functionality for communicating with all households, and thus allowing remote controlling of households. The application will communicate with this could service through a REST API transferring JSON encoded data.

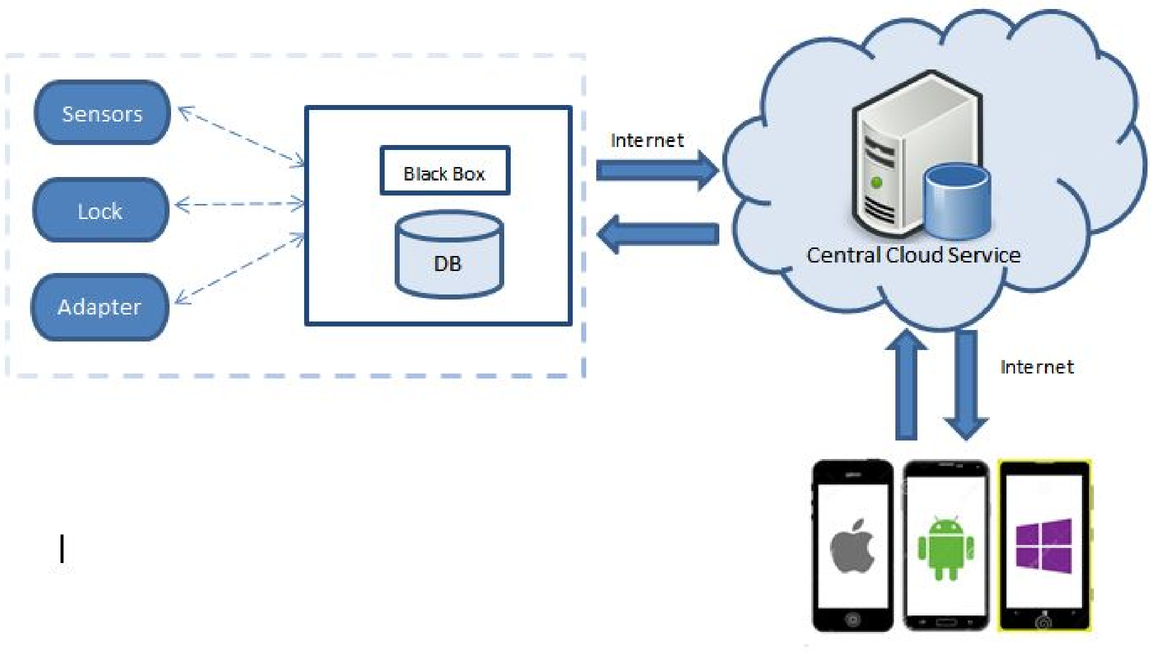


Fig. 2 - An illustration of the communication between the application and the Remote Security System. The application communicates with a central cloud service, which in turn connects to a desired household.

The REST API, provided by the cloud services is assumed to provide traditional REST methods over HTTP, as specified in the tables below.

Query type	Collection	Single item
Syntax	/entities	/entities/{id}
Example	/users, /locks, /rooms	/users/145, /locks/1874, /rooms/10
Comment	Retrieves the entire collection of the specified entity as an array of objects, encoded in JSON	Retrieves a specific instance of an entity, with the specified id
Associations	-	entities/{id}/associatedEntities entities/{id}/associatedEntities/{id} e.g. *households/12/locks* to retrieve all locks of household 12, or *households/12/locks/1* to retrieve lock 1 of household 12 (although locks/1 would work too).

Table 1. The REST query protocol required by the application.

Method	**CRUD equivalent**	**Normal Responses: Collection**	**Normal Responses: Single item**
POST	Create	201 (Created)	404 (Not Found)
PUT	Update	404 (Not Found) - cannot update entire collection	200 (OK), 404 (Not Found)
GET	Read	200 (OK)	200 (OK), 404 (Not Found)
DELETE	Delete	404 (Not Found) - cannot delete entire collection	200 (OK), 404 (Not Found)

Table 2. The REST operations and responses required by the applation.

SWI1.	The application shall support the REST interface described in tables 1 and two.
Comment	This REST API is presumed to provide all entities contained in the Data Model in 3.2.1 Data model (and further described in the Data Dictionary), and be available in the cloud service.

3.1.2 Hardware interfaces

The application should be able to run on devices compatible with running either Android 4.0, iOS 7 or Windows Phone 7.8. However, this requirement is based on respective operating system supporting different hardware devices, i.e. the application makes no assurances that it will be able to run on every device with respective operating system, but rather that works on respective operating system.

Certain functionality in the application will not be available in devices which do not support NFC (Near Field Communication). However, all the other functionalities of the application should still work as expected if the device, which the application runs on, does not support NFC.

3.1.3 Communications interfaces

The application will use two communication interfaces, NFC for communicating directly with locks in a house, and HTTPS to communicate with the Remote Security System cloud service via its REST API. The NFC communication will be encrypted using Remote Security System's proprietary encryption standard. The HTTPS communication will be encrypted using SSL (Secure Socket Layer) over port 443.

CI1.	The application requires an internet connection and port 443 to be able to remotely connect, and thus allow operating and viewing of statuses of devices

CI2.	The smartphone on which the application is run must support NFC in order for the NFC functionality of the application to work.
Comment	The other functionality will work as usual

3.1.4 Memory constraints

The application will be a service that is in sync with a cloud service to periodically gather information about the status of the devices of a household. Each time when the application is used the background service of the application will fetch the data from the database in the cloud service. Cached data may be stored on the smartphone, however the size of this data will be insignificant, thus no strict memory constraints needs to be specified.

3.2 Data requirements

In this section the data requirements of the application are presented and explained. First a data model gives an overview of the entities of the application, as well as their relation to one another. The entities in the data model are further described in a data dictionary.

DR1	The application should support the data model described in 3.2.1 and 3.2.2.

3.2.1 Data model

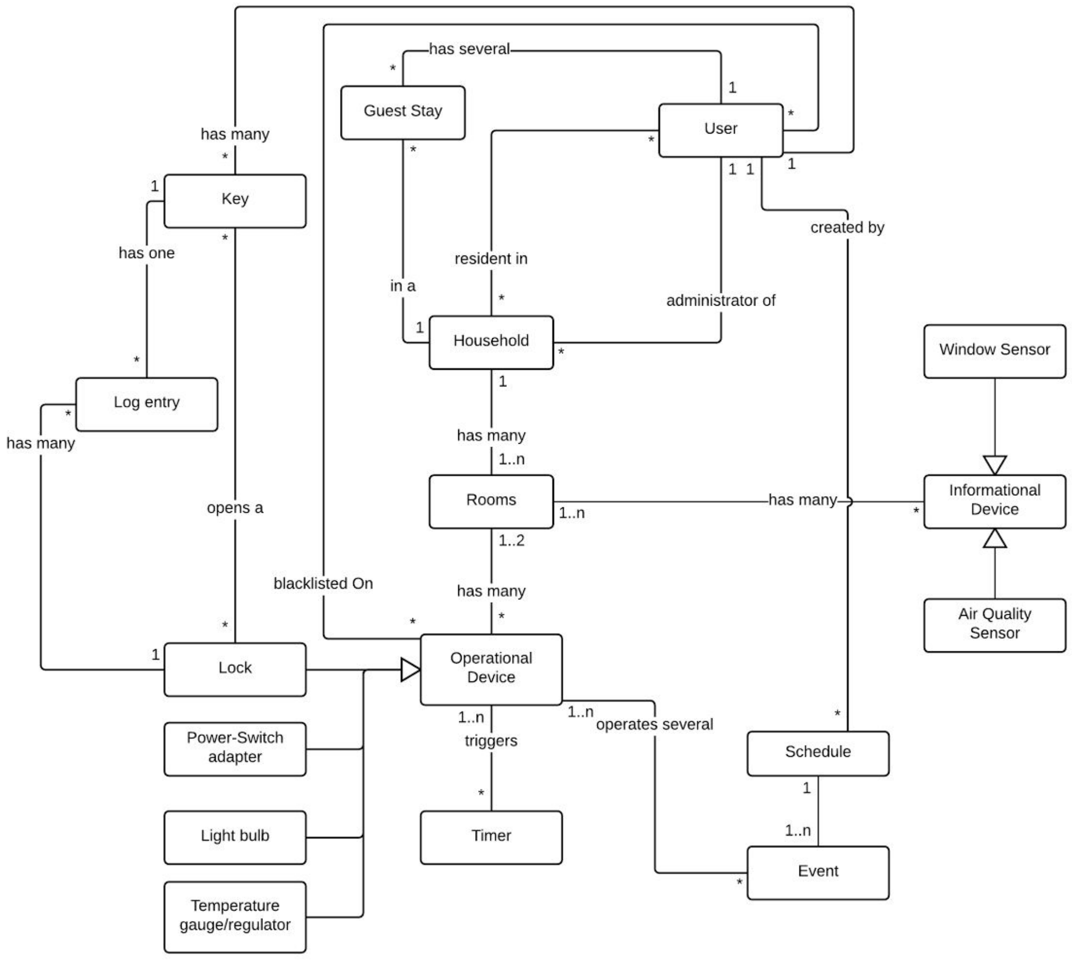


Figure 3. The high-level UML domain model diagram of the system.

Illustrated in figure 3 is the domain model diagram for the system. The domain model diagram illustrates the different entities and their relation to each other, but does not include any attributes. The attributes, along with further description of the entities can be found in section 3.2.2 data dictionary.

3.2.2 Data dictionary

Class: User
A user is a person whom has access to the app, is registered in and can operate the system. A user can be registered to operate one or more households, thereby being a resident of those households. A user can also have a guest stay at, and thereby be granted access to, another household which the user is not registered to. Furthermore, a user can also be an administrator of several households, however a user cannot be an administrator and resident of the same household.

Examples
1. A person who lives in the household of the system
2. A family member who is allowed access to the house
3. A friend who is a guest at a household in which the system is used

Attributes

first name	Text, 20 chars The first name of the person registered as a user
last name	Text, 30 chars The last name of the person registered as a user
email address	Text, 100 chars The email address of the person registered as a user. Must be in the format xxx@xxx.xxx
phone number	Text, 100 chars The phone number of the person registered as a user
password	Text, 60 chars The (encrypted) password for the user
last password change	Date The date of which the user's password was last changed

Class: Household
A household in which a specific instance of the Remote Security System operates. A household has multiple users, which are allowed to operate the system and its components in the household. Furthermore, a household always has exactly one administrator, which manages access for the other users (See #Administrator). A

household also has multiple rooms, which in turns contains the different components of the system.

Example

1. A household on “Imaginary drive 784” with zip code “66016” and city “Kansas city”
2. An apartment on “Imaginary street 487” with apartment number “334101” and zip code “66017” and city “Kansas city”

Attributes

address	Text, 50 chars The street address of the household. Used for legal reasons and emergency responders (fire dept. and police)
apartment number	Text, 10 chars The apartment number of an apartment, if the household is an apartment. Blank by default.
zip code	Text, 10 chars The zip code of the household
city	Text, 30 chars The city of the household

Class: Room
A room represents a physical room in a household, and exists as a means for categorizing system components by location. A room has zero or more instances of operational devices, which are devices that a user can remotely operate (See #OperationalDevice), and zero or more instances of informational devices, which are devices that provides information about the physical presence of the room and its entities to the user (See #InformationalDevice). Furthermore, a room always belongs to (located in) exactly one house.

Examples

1. Living room
2. Hallway
3. East bedroom

Attributes

name	Text, 50 chars The name of the room

Class: Operational Device (abstract)
A device in a household which a user can operate. All operational devices possesses a state of a physical entity they control, e.g. a light bulb, a stovel. All operational devices are also able to change the state of the physical entity they control; this could involve, for example, turning the power on or off to an electrical device, adjusting the temperature, etc.

An operational device possesses a blacklist, which is a list of users (See #User) that are not allowed to operate said device.

An operational device can be operated/automated by an event, i.e. a scheduled operation the device.

An operational device is located, and belongs to, a Room (See #Room)

Examples

1. A light bulb which can be turned on or off, located in the Living Room
2. A air-condition unit which can be set to regulate the temperature of a room, located in the Bed Room

Attributes

name	Text, 50 chars The name of the operational device
state	State The state of operational device (depends on what kind of operational device)

Class: Schedule
A gathering of one or more events which takes place over time, the intention of which is to schedule operations of Operational Devices (See #Event).

Examples

1. A schedule which contains an event that every day at 7pm turns on a window lamp
3. A schedule which contains an events that from sunday to friday turn down the heating at 11pm, to lower the temperature during sleeping hours.

Attributes

name	Text, 50 chars The name of the operational device

Class: Event
An event is a scheduled operation of one or more operational devices to change state at a specified time, thus an Event always has at least one operational device to operate on. An event belongs to a Schedule, which is a grouping/gathering of different events. Furthermore, an event can be repeated on a regular day-basis.

Examples
1. Change the state of a light bulb to be “On”, at 18:48 2016-05-19, repeated every 5th day until 18:48 2017-05-19
2. Change the state of Lock to “Open” at 22:18 2015-05-15

Attributes

start time	Date The time to start the event
end time	Date The time to end the event
repeated	Integer Indicates the amount of days before this event is repeated, i.e. this event is repeated every X days. Set to 0 by default, indicating that it is not repeated.
repeated-end-date	Data The date which this events stops repeating itself
state	State The state to be set on the operational device

Class: Guest Stay
A representation of a guest staying at, and having access to, a household. A guest stay always has exactly one User, which is the Guest, and one Household, which is where the User is a guest at. Furthermore, a guest stay contains information of the time period of when the guest is allowed access to the Household.

Examples
1. User “Mr. Donaldson” has access to the household at “Livington street 829” from 18:00 2015-06-18 to 10:00 2015-06-20

Attributes

access from	Date The date (including time) from which the guest (User) has access to the household
access to	Date The date (including time) to which the guest (User) has access to the household

Class: Informational Device
A device in a household from which the user can retrieve the state of a physical entity. Compared to an Operational Device, an Information Device can only retrieve the state of a physical entity, but not operate its state. Informational Devices can often be thought of as sensors.

An informational device is located, and belongs to, a Room (See #Room)

Examples

1. An air quality sensor
2. A temperature sensor.

Attributes

name	Text, 50 chars The name of the informational device

Class: Lock (inherits from Operational Device)
A lock in an exterior door to a household. The lock can be in one of two states: unlocked or locked. The lock also has zero or more keys, which can unlock and lock the door. (A lock should always have at least one associated key, however cases exist, e.g. when the system is being set up, where the lock will temporarily operate with no keys.) Furthermore, a lock can operate in one of two modes: normal and auto lock. In normal mode a lock is only locked or unlocked when manually applying an associated key. In auto-lock mode, the lock is automatically locked whenever the door is closed, and can always, without a key, be opened from the inside.

Attributes

state	State [“Locked”, “Unlocked”] The state of the lock

mode	Mode [“Normal”, “Auto lock”]

Class: Key
A representation of a key, which may take the physical form of a key card, or as a virtual key in the application.

A key is always associated with exactly one User (See #User), and one or more Locks (See #Lock)

Attributes

key number	Integer A globally unique number assigned to the key. Must be of format “XXXX-XXXX-XXXX-XXXX”
pin	Integer A numerical combination which the user needs to enter in order to use the key. Must be at least 4 digits long.

Class: Log entry
An item which represent a recording of a state change in a Lock. Every time a lock is unlocked or locked a log entry is created. The log entry is associated to exactly Lock, and one Key (and thereby a user).

Example

1. Lock #45 was “Locked” at “2015-09-09 18:09:00” by Key #89

Attributes

state	State [“Locked”, “Unlocked”] The state that the lock was changed to
timestamp	Date The exact date (with time) that the lock’s state was changed

Class: Power-Switch adapter (inherits from Operational Device)

An adapter to the system which can enable any external high-voltage (110-240V) electrical device to be remote controlled, by switching power on or off, by the system.

Examples

1. A power switch adapter which has a stove connected to it has a state “On” and is critical.
2. A power switch adapter which has lamp connected to it has a state “Off” and is non-critical.

Attributes

state	State [“On”, “Off”] The state of the power supply to the external device.
isCritical	Boolean if the external electrical devices which is connected to this power switch adapter is of a critical nature and should not be left unattended.

Class: Light bulb (inherits from Operational Device)
A specifically manufactured light bulb which can be remotely controlled by the system. The light bulb can be dimmed, as well as change the color of the light it emits.

Attributes

state	State [“On”, “Off”] The state of the light bulb
dim value	Number [0-1] The dim value of the lamp. Ranges from 0 to 1
color	Color The color of the light emitted by the lamp

Class: Temperature gauge/regulator (inherits from Operational Device)
A device which can read the temperature and is able to change the temperature of a room, both by cooling and warming the air, e.g. an AC unit.

Attributes

state	State [“On”, “Off”] The state of the temperature gauge/regulator
actual temperature	Number The actual temperature, in Celsius, of the air in a room which this device is located in.
target temperature	Number The target temperature which the device is set to try to keep
minimum target temperature	Number The minimum temperature that is allowed to be set as “target temperature”
maximum target temperature	Number The maximum temperature that is allowed to be set as “target temperature”

Class: Window sensor (inherits from Informational Device)
An device which can sense if a window is opened or closed.

Attributes

window state	State [“Opened”, “Closed”] The state a window to which this sensor is attached

Class: Timer
A timer which makes a state change on an operational device after a given time. Compared to an event, a timer does not have an end-time, and it is not saved after the state change has been executed.

Attributes

time of triggering	Date The timestamp on which the state change of the operational device is enforces. Measured in milliseconds

Class: Air Quality Sensor (inherits from Informational Device)

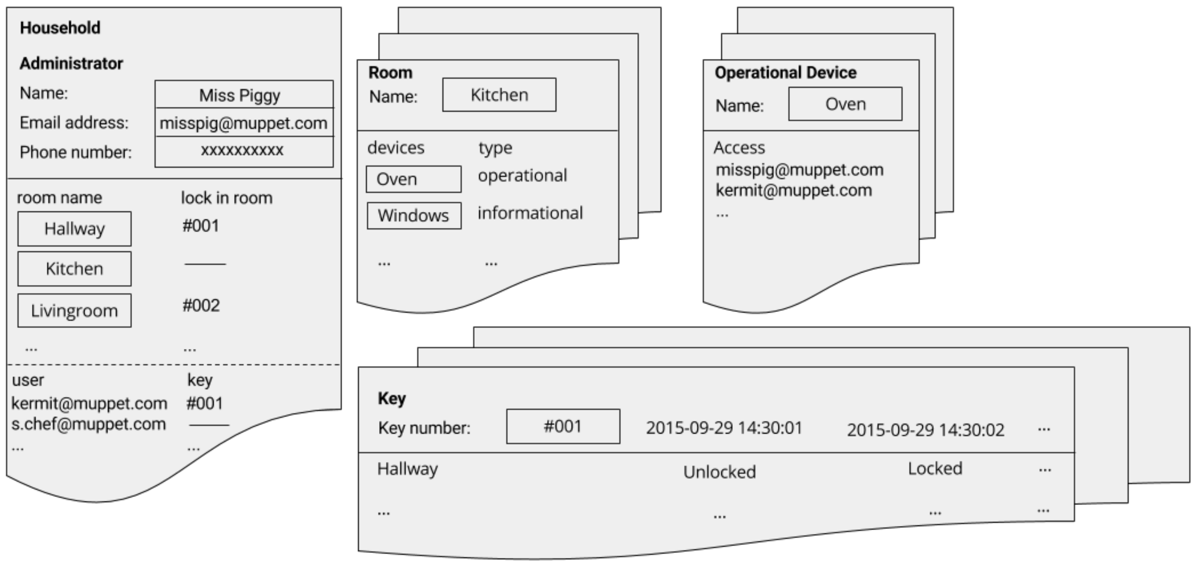
An device which can measure the air quality in a room

Attributes

air quality	Number Air quality index, measured in the air-quality index this sensor is programmed to use.
air-quality index	Air Quality in Europe - CITEAIR [13]

3.2.3 Virtual Windows

This section contains the virtual windows that are used in this SRS. Their purpose is to provide means of easier tracing between data from the data dictionary to the tasks. They are only meant to be seen as data representations and not design requirements.



Household
Administrator
Name: Miss Piggy
Email address: misspig@muppet.com
Phone number: xxxxxxxxxx

room name	lock in room
Hallway	#001
Kitchen	——
Livingroom	#002
...	...

user	key
kermit@muppet.com	#001
s.chef@muppet.com	——
...	...

Room
Name: Kitchen

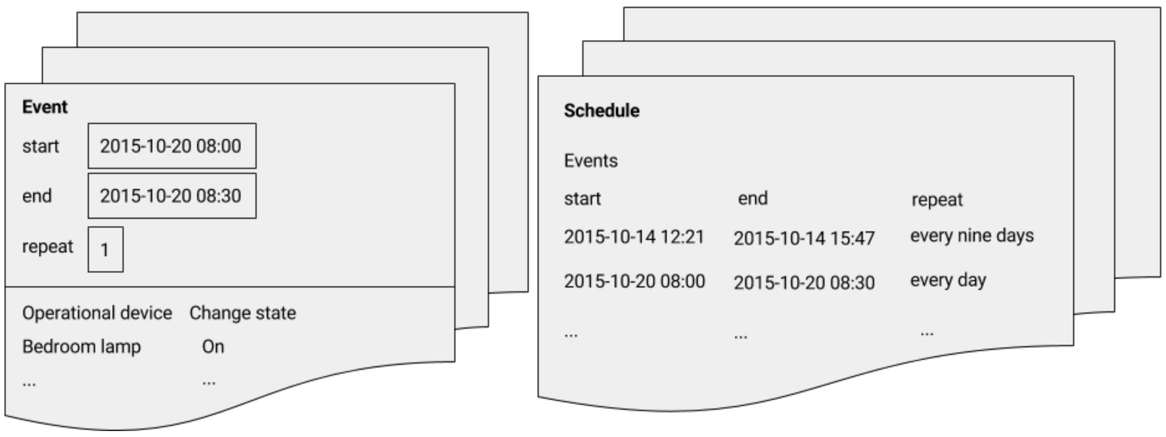
devices	type
Oven	operational
Windows	informational
...	...

Operational Device
Name: Oven
Access
misspig@muppet.com
kermit@muppet.com
...

Key
Key number: #001

	2015-09-29 14:30:01	2015-09-29 14:30:02	...
Hallway	Unlocked	Locked	...
...	...	...	...

Figure 4 - Virtual windows describing the data for Household, Room Operational Device and Key.



Event
start 2015-10-20 08:00
end 2015-10-20 08:30
repeat 1

Operational device	Change state
Bedroom lamp	On
...	...

Schedule
Events

start	end	repeat
2015-10-14 12:21	2015-10-14 15:47	every nine days
2015-10-20 08:00	2015-10-20 08:30	every day
...	...	...

Figure 5 - Virtual windows describing the data for Event and Schedule.

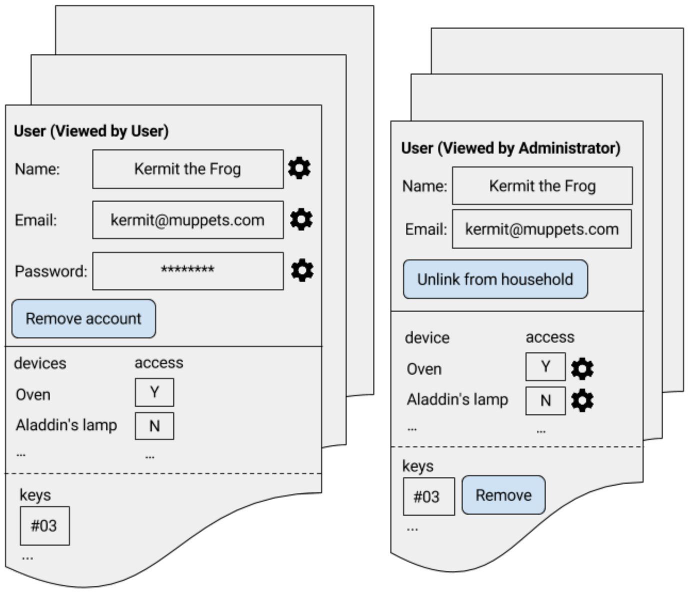


Figure 6 - Virtual windows describing User as viewed by User and User as viewed by admin.

3.3 Functional requirements

The following section describes the functional requirements of the system. Following is an event list of high-level business/domain events that the application should support. These events closely relates to the high-level user goals, described in section 2.1 Product perspective.

ELV1: The product should support the following business events/user activities/tasks:

ELV1.1: User operates lock *(Relates to UG3, UG4, UG5)*
ELV1.2: User turns operates temperature *(Relates to UG3)*
ELV1.3: User operates light bulb *(Relates to UG3)*
ELV1.4: User operates electrical device *(Relates to UG3)*
ELV1.5: User looks over the state of their household *(Relates to UG2, UG4)*
ELV1.6: User leaves home without locking *(Relates to UG4)*
ELV1.7: Guest comes and stays over *(Relates to UG2, UG3, UG5)*
ELV1.8: New family member starts using devices in a household (e.g. family member moves in or comes of age), *(Relates to UG2, UG3, UG5)*
ELV1.9: Family member moves out *(Relates to UG5)*
ELV1.10: Family moves out *(Relates to UG3)*
ELV1.11: Family installs system *(Relates to all user goals)*
ELV1.12: User loses key *(Relates to UG4, UG5)*
ELV1.13: User schedules lights or other electrical device to be turned on/off *(Relates to UG1, UG4)*
ELV1.14: User schedules temperature to be at a certain level *(Relates to UG1, UG4)*
ELV1.15: User changes/removes schedule *(Relates to UG1)*
ELV1.16: Family moves in *(Relates to all user goals)*

On a product level, the application is specified by three different subsystems/components: *Account and Access Management, Device Management and Operating,* and *Scheduling*. These three subsystem are a logical grouping of functionality in the application. The *Account and Access Management* subsystem communicates with the Account system (to handle accounts) and Household systems (to manage access to devices in different households), both described in section 2.4 Product context. The *Device Management and Operating* subsystem, as well as the *Scheduling* subsystem communicates only with Household systems, in order to operate devices and manage scheduling in the different households of which the user is a member.

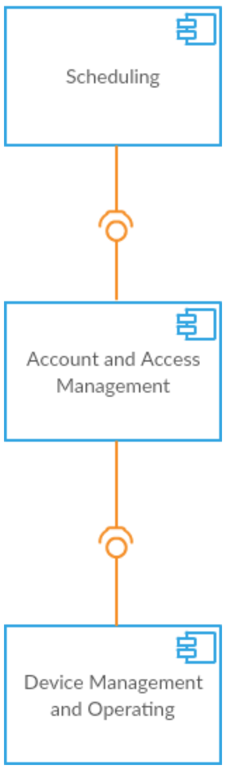

Scheduling
Account and Access Management
Device Management and Operating

Figure 7 An abstract component diagram showing the relations between the three different subsystems/components.

In a system design perspective, as illustrated by figure 7, the three subsystems can be thought of as components that are structured in a hierarchical manner. The Device Management and Operating subsystem is the fundamental component in the application; without it the other components are not useful. The Account and Access Management subsystem requires an interface for Device Management and Operating in order to handle access for different devices. Likewise, Scheduling requires the same interface in order to schedule operations of devices. The Scheduling subsystem also requires an interface for the Account and Access Management system to know which devices a user is allowed to schedule operations for.

The hierarchical structuring of the subsystems provides several benefits. Firstly, if implemented correctly the application could operate without a higher ranking subsystem, for example, the scheduling subsystem and only use the lower two. This way, a version of the application without scheduling functionality could be assembled, while maintaining the same code for the lower two subsystems. In the same way, the Account and Access Management could also be removed and the application could be used purely to control devices without restrictions. Secondly, the hierarchical structuring also gives guidance in prioritization. Without some implemented functionality Device Management and Operation, the system will nott be useful. The hierarchical structuring of the subsystem is, however, not a requirement for implementation; it is merely a basis for thinking of and discussing system architecture, and it is helpful to have in mind when reading the requirements in this document.

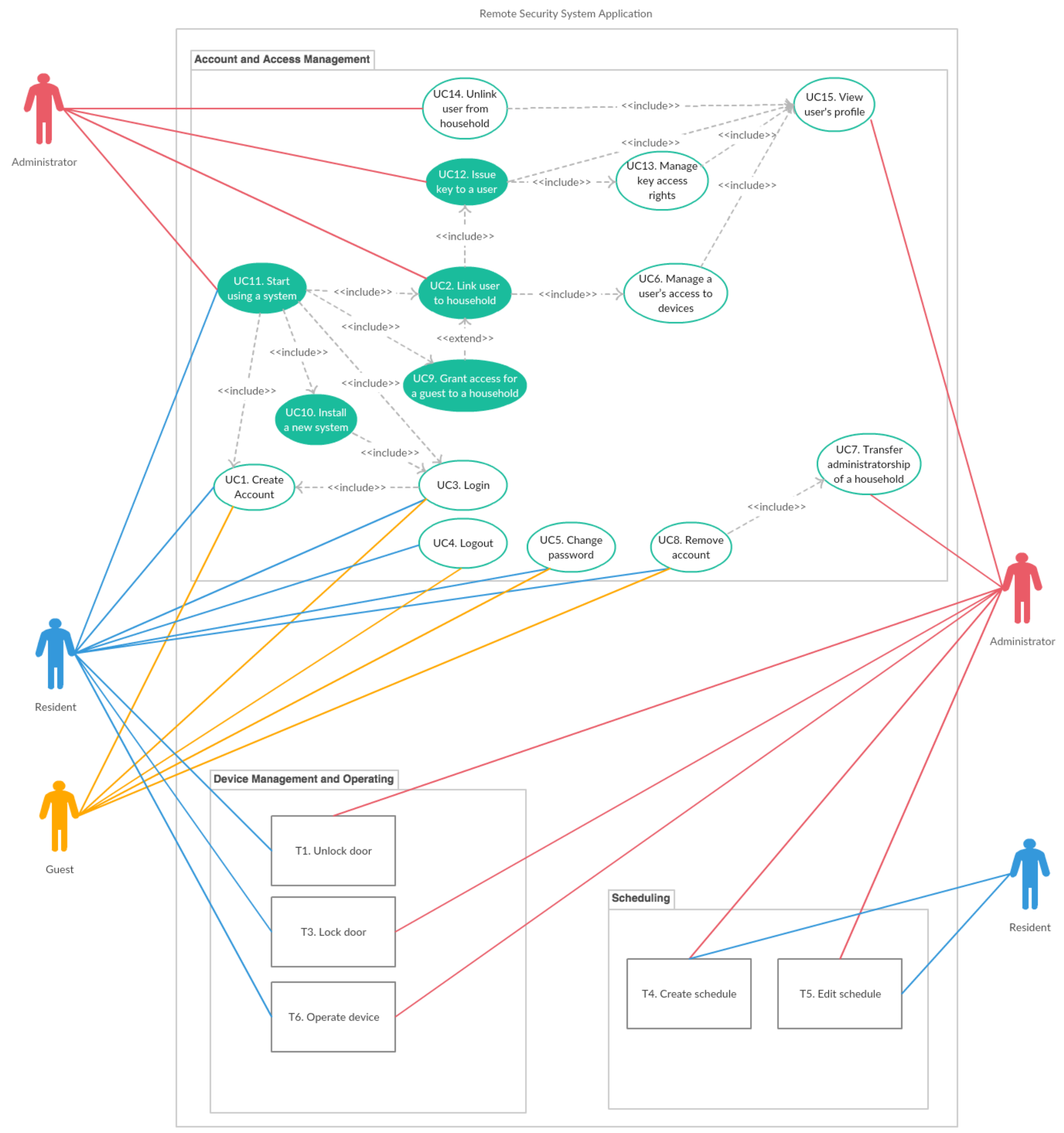

Remote Security System Application
Account and Access Management
Administrator
UC14. Unlink user from household
UC15. View user's profile
<<include>>
UC12. Issue key to a user
UC13. Manage key access rights
UC11. Start using a system
UC2. Link user to household
UC6. Manage a user's access to devices
<<extend>>
UC9. Grant access for a guest to a household
UC10. Install a new system
UC1. Create Account
UC3. Login
UC4. Logout
UC5. Change password
UC8. Remove account
UC7. Transfer administratorship of a household
Resident
Guest
Device Management and Operating
T1. Unlock door
T3. Lock door
T6. Operate device
Scheduling
T4. Create schedule
T5. Edit schedule

Figure 8 - Use case and Task Description diagram. The diagram illustrates the relation between use cases, tasks and the actors. A rectangle represents a task description, and an oval represents a use case. Use cases that are filled in green are use cases that relate more closely to user goals than the ones which only have a green border. The use cases with green borders describes tasks which are needed to support other tasks, which are more related to user goals.

As described in figure 8, the Account and Access Management subsystem is described by use cases. These uses cases describes tasks mainly as product level requirements, since these tasks requires detailed specification in order to be sufficiently useful. The Device Management and Operating, as well as the Scheduling subsystems are described by Task Descriptions in combination with Feature Requirements, since they involve more trivial tasks than the Account and Access Management subsystem. More information of why these methods were chosen is available in the project experience report. Figure 8 also shows, as a limitation, that guests are not allowed to create or edit schedules.

3.3.1 Account and Access Management

The Account and Access Management subsystem is mainly specified by use cases. The system is required to support the functionality of these use cases (UC1 - UC15).

FR.A2	The system shall allow users to view their profiles.

FR.A4	The system shall allow a user to operate multiple households
Limitation	Only operate one household at a time
Comment	Most use cases found below have as pre-conditon that the household to operate on is chosen. Thus, those use cases are dependant on this requirements. An example of how this could be implemented can be found in section 4.5.1 Mock-ups.

Use case	UC1. Create Account
Purpose:	Enables a user to use the Remote Security System
Trigger:	A new user wants to use the Remote Security System
Frequency:	Once per user
Critical:	Non-critical
Actor:	Administrator, Resident, Guest
Pre-condition:	The user is not logged in
Post-condition:	Provided that all input data is valid, the user did not already exist, and that a connection to the system could be established at the time of registration, a new user will be registered in the system.

Main flow
1. The system displays the signup form
2. The actor enters his/hers email address, password, name
3. The system validates the input data
4. Assume: the input data is valid
5. The system registers the user
6. Assume: the registration succeeded

7. The system informs the user that the registration went through

Alternative flows

4a: The email address is invalid

1. The system informs the actor that an invalid email address was entered
2. Go to action step 2

4b: The password does not match the criteria

1. The system informs the actor that the password did not match the criteria according to QR.S2
2. Go to action step 2

4c: Any field is empty

1. The system informs the actor that all fields are required to be filled out
2. Go to action step 2

6a: The user already exist

1. The system informs the actor that a user with the input email address already exist, based on a unique e-mail address
2. Choice: Re-enter email address (Go to action step 2)

6b: A connection to the system could not be established

1. The system informs the actor that a connection to the system could not be established (and possible reason why not) and presents the actor with a choice to try again.
2. Assume: The actor chooses to try again
3. The application tries to establish a connection with the system

6b.2a: The actor chooses not to retry establishment of connection

1. The application disregards any changes made
2. Quit use case

Use case	UC2. Link user to household
Includes	UC6. Manage a user’s access to devices; UC12. Issue a key to a user
Purpose:	Enables a user to access the functionality of a household
Trigger:	A user wants access to a new household, i.e. become a resident
Frequency:	Low
Critical:	Non-critical
Actor:	Administrator
Pre-condition:	The actor is logged in and the household is chosen
Post-condition:	Provided that a connection to both the system can be established and at least one user which exist was chosen, the user(s) will be linked and thereby allowed access to the household

Related Requirements	ELV1.8: New family member starts using devices in a household (e.g. family member moves in or comes of age). *How:* enables the new user to gain access to the household

Main flow

1. The actor navigates to the option for linking users
2. The system presents a form for searching users
3. The actor enters the email address of the desired user
4. Assume: a connection to the system can be established
5. Assume: the user with the entered email address exist
6. Choice: link multiple users (Go to action step 3)
7. The actor manages user(s) access to operational devices. Include UC6.
8. The actor issues a key to to user(s). Include UC12.
9. The system links the chosen user(s) to the chosen household
10. Assume: the linking succeeded.
11. The system informs the user that the linking succeeded

Alternative flows

4a: A connection to the system could not be established

1. The system informs the actor that a connection to the system could not be established (and possible reason why not) and presents the actor with a choice to try again.
2. Assume: The actor chooses to try again
3. The application tries to establish a connection with the system

4a.2a: The actor chooses not to retry establishment of connection

1. The application disregards any changes
2. Quit use case

5a: The user with the entered email address does not exist

1. The system informs the actor that the desired user does not exist
2. Go to action step 3.

10a: Linking failed due to possible temporary error (such as no internet connection)

1. The system informs the actor that the linking failed, and possible reason why.
2. Choice: The actor chooses to try again (Go to action step 10)

10b: Linking failed due possible fatal error (such as the household being removed, administrator rights removed etc.)

1. The system informs the actor that the linking failed, fatally, and possible reason why.

Use case	UC3. Login
Purpose:	Enables a user to access the system

Trigger:	A user wants to operate or be informed of the state of a household
Frequency:	Medium
Critical:	Non-critical
Actor:	Administrator, Resident, Guest
Pre-condition:	Actor is not logged in
Post-condition:	Provided that all input data is valid and that the actor has valid credentials, the actor is assigned a login session

Main flow

1. Assume: the actor has an account in the system
2. The system asks the actor to input email and password
3. The actor enters credentials
4. The system validates credentials
5. Assume: the credentials are valid
6. The system initializes a session for the actor

Alternative flows

1a: Actor has no account

1. Actor navigates to the signup form
2. System presents form for signing up. Include use case UC1.

4a: A connection to the system could not be established

1. The system informs the actor that a connection to the system could not be established (and possible reason why not).
2. Choice: The actor chooses to try again (Go to action step 3)

4b: Invalid email-address format

1. The system informs the actor that the email address has an invalid formatting
2. Choice: The actor chooses to re-enter the email address (Go to action step 2)

4c: Invalid password format

1. The system informs the actor that password is too short, according to QR.S2
2. Choice: The actor chooses to re-enter the password (Go to action step 2)

4d: Invalid credentials

1. The system informs the actor that the email or the password were incorrect
2. Choice: The actor chooses to re-enter the email or the password (Go to action step 2)

Use case	UC4. Logout
Purpose:	Enables a device (smartphone) to be used by another user and makes access to the system unavailable through the logged in actor's account

Trigger:	The actor is temporarily finished using the system
Frequency:	Low
Critical:	Non-critical
Actor:	Administrator, Resident, Guest
Pre-condition:	Actor is logged in
Post-condition:	The actor’s login session is ended

Main flow

1. The actor communicates to the system that he wants to log out
2. Assume: no unfinished or pending tasks
3. The system ends the actors session

Alternative flows
1a: There is a pending task

1. The system informs the actor that a task (such as a saving of data, or an unsaved form) is unfinished, pending or being executed. Furthermore, the system asks if the actor if to proceed with the logout and informs the actor of the risk of doing so.
2. Assume: The actor wants to finish the unfinished, or pending task.
3. The actor completes the task
4. Go to action step 3.

Alt. flow 1a.2a: The actor wants to force a logout

1. Go to action step 3.

Use case	UC5. Edit profile
Purpose:	Enables a user to change password, name, and phone number.
Trigger:	The actor wants to change password (for security reasons), name or phone number. The password hasn’t been changed in the past 3 months.
Frequency:	Automatically invoked every 3 months, or manually very occasionally
Critical:	Non-critical
Actor:	Administrator, Resident, Guest
Pre-condition:	Actor is logged in
Post-condition:	Provided that the actor enters the correct current password, a valid new password, valid first name, last name, and password, as well as

	a connection to the system could be established, the actor's password will be changed to the new password entered, and the first name, last name, and phone number will be updated if they were changed.

Main flow

1. The actor navigates to the option for editing profile.
2. The system prompts the actor to input the current password, the new password, and confirm the new password by inputting it again, as well as the first name, last name, and phone number
3. The system validates the new password
4. Assume: the new password matches the password security criteria QR.S2
5. Assume: the confirmation of the new password is successful
6. The system validates first name, last name, and phone number.
7. Assume: the first name, last name, and phone number are valid.
8. Assume: a connection to the system could be established
9. Assume: the current password is correct
10. The system changes the current password to become the new password, as well as updates first name, last name, and phone number if they have been changed.

Alternative flows

4a: The password does not match the criteria

1. The system informs the actor that the new password does not match the security criteria according to QR.S2.
2. The actor enters another password
3. Go to action step 2.

5a: The new password confirmation did not match

1. The system informs the actor that the confirmation of the new password was unsuccessful
2. The actor re-enters the new password
3. Go to action step 2.

7a: The first name, last name, or phone number are not valid

1. The system informs the actor that the first name, last name, or phone number are invalid
2. The actor re-enters the invalid information.
3. Go to action step 6.

8a: A connection to the system could not be established

1. The system informs the actor that a connection to the system could not be established (and possible reason why not) and presents the actor with a choice to try again.
2. Assume: The actor chooses to try again
3. The application tries to establish a connection with the system

8a.2a: The actor chooses not to retry establishment of connection

1. The application disregards any change
2. Quit use case

9a: The current password is incorrect

1. The system informs the current password is incorrect
2. The actor re-enters the current password
3. Go to action step 5.

Use case	UC6. Manage a user's access to devices
Includes	UC15. View user's profile
Purpose:	To enable an administrator to restrict or extend a user's access to devices in a household
Trigger:	A new user has been registered to a household and his access rights need to be managed in order to be able to use the system. An existing user should gain or lose access to certain device for security reasons.
Frequency:	Low
Critical:	Non-critical
Actor:	Administrator
Pre-condition:	The actor is logged in and the household is chosen
Post-condition:	Provided that a connection to the system could be established, any changes made to the user's access rights are set in motion.
Related Requirements:	ELV1.8: New family member moves in or comes of age (starts using devices in a household). *How:* modify the new user's access ELV1.7: Guest comes and stays over. *How:* modify guest's access

Main flow

1. Assume: a user, whose access is to be managed, has been chosen
2. The system retrieves a list of all operational devices, except locks, in the household
3. Assume: a connection to the system could be established
4. The system presents a list of all operational devices, except locks, and indicates which of these devices the user has access to.
5. The actor checks or unchecks the operational devices in order to allow or disallow access to the devices
6. Assume: a connection to the system can be established
7. The system changes the access rights of the user

Alternative flows
1a: A user has not been chosen

1. Select user to unlink by view his profile. Include UC15.

2. The actor chooses the option for managing the user's device access.

2a,6a: A connection to the system could not be established

1. The system informs the actor that a connection to the system could not be established (and possible reason why not) and presents the actor with a choice to try again.
2. Assume: The actor chooses to try again
3. The application tries to establish a connection with the system

2a,6a.2a: The actor chooses not to retry establishment of connection

1. The application disregards any changes to the user's access right
2. Quit use case

Use case	UC7. Transfer administratorship of a household
Purpose:	To enable an administrator to quit his duties as an administrator
Trigger:	The administrator moves out of the household. The administrator, for some reason, does not want to administrate the household
Frequency:	Very Low
Critical:	Non-critical
Actor:	Administrator
Pre-condition:	The actor is logged in, the household is chosen
Post-condition:	Provided that a connection to the system could be established, the actor confirms the transfership, the actor's password is correct, and that there are some other user to transfer the administratorship to, the system will transfer ownership.
Related requirements:	ELV1.9: Family member moves out. *How:* transferal of administratorship if the administrator moves out. ELV1.10: Family moves out. *How: the administrator moves out, transfers the administratorship to new owner*

Main flow

1. Assume: a connection to the system can be established
2. Assume: there are other users than the administrator in the household
3. The system presents a list of the other users in the household
4. The actor chooses a user to transfer administratorship to
5. The system asks for the actor's password
6. The actor enters current password
7. The system asks for confirmation of transfer
8. Assume: the actor confirms transfer
9. Assume: a connection to the system can be established
10. The system validates the actors password

11. Assume: the password is correct
12. The system transfers administratorship to the chosen user, and the administrator becomes a resident
13. The system notifies the chosen user

Alternative flows
1a, 9a: A connection to the system could not be established
1. The system informs the actor that a connection to the system could not be established (and possible reason why not) and presents the actor with a choice to try again.
2. Assume: The actor chooses to try again
3. The application tries to establish a connection with the system

1a, 9a.2a: The actor chooses not to retry establishment of connection
1. The application disregards any changes to the user's access right
2. Quit use case

2-13a: No other users than the administrator in the household
1. System informs the actor that a transfer is not possible

8-13a: The actor does not confirm transfer of administratorship
(Quit use case)

11a: The password is incorrect
1. The system informs the password is incorrect
2. The actor re-enters the current password
3. Go to action step 6.

Use case	UC8. Remove account
Includes:	UC7. Transfer administratorship of a household
Purpose:	To allow an actor to permanently discontinue usage of the system
Trigger:	The actor moves out of the household. The actor, for some reason, does not want to use the system anymore
Frequency:	Very Low
Critical:	Non-critical
Actor:	Administrator, Resident, Guest
Pre-condition:	Actor is logged in
Post-condition:	Provided that a connection to the system could be established, and the actor enters a correct password and confirms deletion, the actor's account will be deleted from the system.
Related requirements:	ELV1.9: Family member moves out. *How:* Users might want to remove their accounts if they don't use the system in any other household

Main flow

1. The actor navigates to the option for deleting their account
2. The system presents the consequences when deleting an account and asks for the actor's password
3. The actor enters their password
4. The system asks for confirmation of deletion
5. Assume: the actor confirms deletion
6. Assume: a connection to the system can be established
7. The system validates the actors password
8. Assume: the password is correct
9. Assume: the actor is not an administrator
10. The system removes the actor's account

Alternative flows
5-10a: The actor does not confirm removal
(Quit use case)
6a: A connection to the system could not be established

1. The system informs the actor that a connection to the system could not be established (and possible reason why not) and presents the actor with a choice to try again.
2. Assume: The actor chooses to try again
3. The application tries to establish a connection with the system

6a.4: The actor chooses not to retry establishment of connection

1. The application disregards any changes to the user's access right
2. Quit use case

8a: The password is incorrect

1. The system informs the password is incorrect
2. Go to action step 3.

9a: The actor is an administrator

1. The system informs the actor that transfer of administratorship must be performed before his account can be deleted.
2. Include use case UC7.

Use case	UC9. Grant access for guest to a household
Extends	UC2. Link user to household
Purpose:	To allow a guest temporary access to a household
Trigger:	A guest comes to visit and wants access to the household's devices

Frequency:	Medium
Critical:	Non-critical
Actor:	Administrator
Pre-condition:	(The same as for the base use case)
Post-condition:	Provided that the post-conditions of the base use case are met and that the date input is valid, a guest will be granted access to the household
Related Requirements:	ELV1.7: Guest comes and stays over. *How:* links guest to household (guest must be linked to household to access devices)

Main flow
After step 6 in the base use case, after the users have been chosen:

1. The system presents a form in which to specify the guest-access duration, i.e. the start date, and the end date
2. Assume: input is valid
3. The system creates a guest stay with specified access duration.

Resume on step 7 in the base use case.

Alternative flows
2a: The input date is not correctly formatted

1. The system informs the user that the input date is not a valid date
2. Go to action step 1.

2b: The start date is a date before today

1. The system informs the user that the start date needs to be a date after or today.
2. Go to action step 1.

Use case	**UC10. Install a new system**
Include	UC3. Login
Purpose:	To allow a person to become the administrator of a household, and thus configure the household settings, i.e. instal a new system.
Trigger:	A part of the installation process
Frequency:	Once per household
Critical:	Non-critical
Actor:	Administrator

Pre-condition:	-
Post-condition:	Provided that the serial number of the household's blackbox and one-time code are valid, and a connection to the system could be established, the actor will become the administrator of the household.
Related Requirements:	ELV1.11: Family installs system

Main flow

1. The actor opens the application
2. Assume: the actor is logged in
3. The actor navigates to the option for installing a new system
4. The system presents a form for entering first-time credentials
5. The actor enters the serial-nr of the Blackbox of the household he wishes to enroll as an administrator of, as well the one-time code received from the product owner.
6. Assume: a connection to the system could be established
7. The system validates the serial number and the one-time code
8. Assume: the credentials are valid
9. The system registers the actor as an administrator of the household and shows the details of the household

Alternative flows

2a: The actor is not logged in

1. The system ask the user to log in. Include use case UC3

6a: A connection to the system could not be established

1. The system informs the actor that a connection to the system could not be established (and possible reason why not) and presents the actor with a choice to try again.
2. Assume: The actor chooses to try again
3. The application tries to establish a connection with the system

6a.2a: The actor chooses not to retry establishment of connection

1. The application disregards any changes to the user's access right
2. Quit use case

8a: The credentials are invalid

1. The system informs the actor that the credentials are invalid
2. Go to action step 5.

8-9a: The one-time code has already been used

1. The system informs the actor that the one-time code has already been used and that the household thus already have an administrator. That administrator can transfer administratorship to the actor (UC7) if desired.

Use case	**UC11. Start using the system**
Include	UC1. Create account; UC3. Login; UC2. Link resident to household; UC9. Grant access for guest to household; UC10. Install a new system
Purpose:	To allow a resident or a guest to begin using the system
Trigger:	A resident or guest wants access to the system
Frequency:	Low
Critical:	Non-critical
Actor:	Administrator, Resident, Guest
Pre-condition:	-
Post-condition:	Provided that the post-conditions of all included use cases holds, a user can begin using the system.
Related requirements:	ELV1.11: Family installs system. *How*: if the house is brand new and the system hasn't been installed. ELV1.8: New family member starts using devices in a household (e.g. family member moves in or comes of age). *How*: the family member creates account, logs in, and the administrator enables the family member to start using devices ELV1.7: Guest comes and stays over. *How*: if it is a guest that will start using the system.

Main flow
1. Assume: the resident or guest has an account in the system
2. Assume: there is an administrator and that the administrator is logged in
3. Assume: it is a resident which will start using the system
4. The administrator links the resident to the household, thus allowing access. Include use case UC2
5. Assume: the resident or guest is logged in
6. The resident or guest gets notified that they have access to the household and can start using the system.

Alternative flows
1a: The actor is not logged in
1. The resident or guest creates an account. Include use case UC1.

2a: There is no administrator for the household
1. The administrator to be installs a new system. Include use case UC10.

2b: There administrator is not logged in
1. The administrator logs in. Include use case UC3.

3-4a: It is a guest that will start using the system
1. The administrator grants temporary access to the guest. Include use case UC9.

5a: The resident or guest it not logged in

1. The resident or guest logs in. Include use case UC3.

Use case	**UC12. Issue a key to a user**
Include	UC15. View user's profile UC13. Manage key access rights
Purpose:	To allow a user to access locks
Trigger:	A user wants access to operate locks
Frequency:	Low
Critical:	Non-critical
Actor:	Administrator
Pre-condition:	The actor is logged in, the user and household is chosen
Post-condition:	Provided that a connection to the system could be established, that a physical key card exist and it's card number is entered if the actor wants to bind a physical key card, a key will be issued to the user
Related requirements:	ELV1.8: New family member starts using devices in a household (e.g. family member moves in or comes of age). *How*: the new family member is issued a key ELV1.7: Guest comes and stays over. *How*: the guest is issued a key ELV1.12: User loses key. *How*: user is issued a new key

Main flow

1. Select user for whom to issue the key by viewing their profile. Include UC15.
2. The actor chooses the option for issuing a key.
3. The system asks if this new key should be bound to a physical key card
4. Assume: the actor wants to bind new key to a physical key card
5. The system asks for the key card number
6. The actor enters the key card number
7. Assume: the key card number has a valid format
8. Assume: a connection to the system could be established
9. The system validates the key card number
10. Assume: the key card number exists and isn't bound by another key
11. The system indicates a successful binding.
12. The system lets the actor choose which locks the key shall give access to. Include use case UC13.

Alternative flows
4-11a: The actor does not want to bind the key to a physical key card
(Continue directly to step 12)
7a: The key card number does not have to correct format
1. The system informs the actor that the format of the entered key card number is incorrect.
2. Go to action step 4.

8a: A connection to the system could not be established
1. The system informs the actor that a connection to the system could not be established (and possible reason why not) and presents the actor with a choice to try again.
2. Assume: The actor chooses to try again
3. The application tries to establish a connection with the system

8a.2a: The actor chooses not to retry establishment of connection
1. The application disregards any changes to the user's access right
2. Quit use case

10-11a: The key card number does not exist
1. The system informs the actor that the entered key card number does not exist and asks if the actor wants to enter another
2. Choice: actor wants to enter another. Go to action step 6.

10-11b: The key card number is bound to another key, administered by the actor
1. The system asks if the key card should instead be bound to the new key
2. Choice: actor wants to re-bind the key card. Go to action step 11.

10-11c: The key card number is bound to another key, not administered by the actor
1. The system informs that the key card is bound to another key and cannot be re-bound for the key card is not administered by the actor.

Use case	**UC13. Manage key access rights**
Includes:	UC15. View user's profile
Purpose:	To restrict or expand access of a key
Trigger:	A resident or guest wants expanded access. An administrator wants to restrict access
Frequency:	Low
Critical:	Non-critical

Actor:	Administrator
Pre-condition:	The actor is logged in, the user and household is chosen
Post-condition:	Provided that a connection to the system could be established, any changes made to the key's access rights are applied
Related Requirements:	ELV1.12: User loses key. *How*: revoke access to the lost key ELV1.9: Family member moves out. *How:* remove access to the members key

Main flow

1. Assume: a user's key, for which the access is to be managed, has been chosen
2. Assume: a connection to the system can be established
3. The system retrieves a list of all operational devices in the household
4. Assume: the system retrieved at least one lock
5. The system presents a list of all locks, and indicates which of these locks the key has access to.
6. The actor checks or unchecks the locks in order to allow or disallow access for the key
7. Assume: a connection to the system can be established
8. The system changes the access rights of the key

Alternative flows

1a: A user has not been chosen

1. Select user to unlink by viewing the user's profile. Include UC15.
2. The actor chooses the option for managing one of the user's keys

2a, 7a: A connection to the system could not be established

1. The system informs the actor that a connection to the system could not be established (and possible reason why not) and presents the actor with a choice to try again.
2. Assume: The actor chooses to try again
3. The application tries to establish a connection with the system

2a, 7a.2a: The actor chooses not to retry establishment of connection

1. The application disregards any changes to the user's access right
2. Quit use case

4-8a: No locks retrieved

1. The system informs the actor that no locks are connected to the household

Use case	**UC14. Unlink a user from a household**
Includes	UC15. View user's profile

Purpose:	To revoke a user’s access from a household
Trigger:	A resident or guest has misbehaved. A resident moves out.
Frequency:	Very low
Critical:	Non-critical
Actor:	Administrator
Pre-condition:	The actor is logged in and a household is chosen
Post-condition:	Provided that a connection to the system could be established, and the actor confirms unlinking the user from the household, the user gets unlinked from the household
Related Requirements:	ELV1.9: Family member moves out. *How:* remove the family members access from a household.

Main flow

1. Select user to unlink by viewing his profile. Include UC15.
2. The actor chooses the option for unlinking a user from a household.
3. The system presents a form for unlinking a user from a household, including the consequences of doing so.
4. The actor confirms the unlinking by inputting his password
5. Assume: a connection to the system can be established
6. The system validates the actors password
7. Assume: the password is correct
8. The system unlinks the user from the household

Alternative flows

5a: A connection to the system could not be established

1. The system informs the actor that a connection to the system could not be established (and possible reason why not) and presents the actor with a choice to try again.
2. Assume: The actor chooses to try again
3. The application tries to establish a connection with the system

5a.2a: The actor chooses not to retry establishment of connection

1. The application disregards any changes to the user’s access right
2. Quit use case

7a: The password is incorrect

3. The system informs the password is incorrect
4. The actor re-enters the current password
5. Go to action step 4.

Use case	**UC15. View user’s profile**
Purpose:	To view a user’s profile and go on to manage user’s access
Trigger:	An administrator wants the contact information for a user. An administrator wants to navigate to functions for handling access for the user
Frequency:	Medium
Critical:	Non-critical
Actor:	Administrator
Pre-condition:	The actor is logged in and household is chosen
Post-condition:	Provided that a connection to the system could be established, and the actor confirms unlinking the user from the household, the user gets unlinked from the household

Main flow

1. Assume: a connection to the system could be established
2. The system presents a list of all users in the household
3. The actor chooses one user to view their profile
4. The system presents the chosen user’s profile, including name, email address, as well as options for managing the user’s access, unlinking the user from the household, issuing a new key, and managing key access’ for every key belonging to the user.

Alternative flows

1a: A connection to the system could not be established

1. The system informs the actor that a connection to the system could not be established (and possible reason why not) and presents the actor with a choice to try again.
2. Assume: The actor chooses to try again
3. The application tries to establish a connection with the system

1a.2a: The actor chooses not to retry establishment of connection

1. The application disregards any changes to the user’s access right
2. Quit use case

3.3.2 Device Management and Operating

The Device management and Operating subsystem is specified by feature requirements FR.D1 - FR.D25. Furthermore, some of these requirements, which are not trivial, are put in context by Task Descriptions T1, T2, and T3. A pre-condition for all the following requirements is that the user has access to the household and device which the requirements concern.

FR.D1	The system shall allow a user to see the status of an operational device
Related (high-level) requirements	ELV1.5

FR.D2	The system shall allow a user to see the brightness level of a light bulb
Related (high-level) requirements	ELV1.5

FR.D3	The system shall allow a user to see the color code of the light coming from a light bulb
Related (high-level) requirements	ELV1.5

FR.D4	The system shall allow a user to see the air quality in a room according to the Air Quality in Europe - CITEAIR, along with an associated color (green to red) if good or bad.
Related (high-level) requirements	ELV1.5

FR.D5	The system shall allow a user to see the actual temperature of a room
Related (high-level) requirements	ELV1.5

FR.D6	The system shall allow a user to see the name of a device (operational and informational device)
Related (high-level) requirements	ELV1.5

FR.D7	The system shall allow a user to see the location, i.e. room, in which a device (operational and informational device) is located, if that information is available
Related (high-level) requirements	ELV1.5

FR.D8	The system shall allow a user to turn on/off a power switch adapter
Related (high-level) requirements	ELV1.4

FR.D9	The system shall allow a user to adjust brightness level of a light bulb
Related (high-level) requirements	ELV1.3

FR.D10	The system shall allow a user to adjust color of the light coming from a light bulb
Related (high-level) requirements	ELV1.3

FR.D11	The system shall allow a user to adjust temperature in a room, by setting a targeted temperature
Related (high-level) requirements	ELV1.2

FR.D12	The system shall allow a user to lock and unlock a door via NFC, given that one of the user’s keys grants access to said door.
Related (high-level) requirements	ELV1.1

FR.D13	The system shall allow a user to lock and unlock a door from the application, given that one of the user’s keys grants access to said door.
Related (high-level) requirements	ELV1.1

FR.D14	The system shall allow a user to enable auto-lock mode on a door
Related (high-level) requirements	ELV1.1

FR.D15	The system shall allow a user to disable auto-lock mode on a door
Related (high-level) requirements	ELV1.1

FR.D16	The system shall allow an administrator to mark a power switch adapter as critical
Related (high-level) requirements	ELV1.4
Comment	When an administrator marks a power switch adapter as critical, anytime a user leaves the household with the critical device being turned on, the user gets a warning. See FR.D17.

FR.D17	The system shall give a user a notification if the user leaves the house with a critical power switch adapter turned on
Related (high-level) requirements	ELV1.4

FR.D18	The system shall allow an administrator to view a time log of lockings and unlockings of a lock (door)
Related (high-level) requirements	ELV1.5

FR.D19	The system shall give an administrator a notification if there have been 5 consecutive unsuccessful unlock attempts (wrong pin code) by a resident or guest of a household which the administrator administrates.
Related (high-level) requirements	ELV1.12

FR.D20	The system shall allow a user to find, by searching, operational and informational devices in a household
Related (high-level) requirements	ELV1.1, ELV1.2, ELV1.3, ELV1.4, ELV1.5

FR.D21	The system shall allow a user to browse operational and informational devices in a household
Related (high-level) requirements	ELV1.1, ELV1.2, ELV1.3, ELV1.4, ELV1.5

FR.D22	The system shall give a user a notification when leaving home without locking, giving the option to lock, given that one of the user's keys has access to the unlocked lock.
Related (high-level) requirements	ELV1.6

FR.D23	The system shall allow an administrator to rename a device.
Related (high-level) requirements	ELV1.11

FR.D24	The system shall give an administrator a notification whenever a new devices has been connected (installed) to the system
Related (high-level) requirements	ELV1.11

Comment	New devices (hardware) are connected to the blackbox and thereafter automatically discovered by the application. For efficiency of automatic discovery, see QR.E4.

FR.D25	The system shall allow a user to see an illustrational layout of a home with all connected equipment in it.
Related (high-level) requirements	ELV1.5
Comment	The illustrational layout should preferably be a of a blueprint nature, with all equipment connected to the system illustrated on top of the blueprint. See section 4.5.1 Mock-ups for an example.

Task Description	T1. Unlock door
Purpose	Unlock a door for oneself or someone else
Trigger	User wants to enter the household
Frequency	High
Critical	Non-critical
Actor	Administrator, Resident, Guest
Related requirements	ELV1.1 (High level requirement) FR.D12, FR.D13, FR.D19

Sub-tasks	Visible data	Agreed upon solutions	Virtual windows
1. Indicate intention for unlocking the door	Lock	1. Have the application invoke the phone's NFC capabilities (if there are any) by swiping the phone by the lock. 2. List of locks, then include an "Unlock" button on each lock in the application	Operational Device (Lock)
1a. No access	Lock's name	1. The hardware will	Operational Device

		indicate. 2. Without access ser won't be able to press "Unlock"	(Lock)
2. Enter pin	Lock's details, Key's number	1. Hardware num-pad 2. Prompt virtual num-pad	Operational Device (Lock), Key
2a. Wrong pin	Lock's details, Key's number	1. The hardware will indicate. User have to swipe phone by lock again 2. Prompt alert box and let user enter pin again	Operational Device (Lock), Key

Task Description	T3. Lock door
Purpose	To lock a door in order to prevent unauthorized persons from entering the house
Trigger	User leaving home. User just got home and wants to lock
Frequency	High
Critical	Non-critical
Actor	Administrator, Resident, Guest
Related requirements	ELV1.1, ELV1.6 (High level requirements) FR.D12, FR.D13, FR.D22

Sub-tasks	Visible data	Agreed upon solutions	Virtual windows
1. Indicate intention for unlocking the door	Locks' name	1. Have the application invoke the phone's NFC capabilities (if there are any) by swiping the phone by the lock. 2. List of locks, then	Operational Device (Lock)

		include an “Lock” button on each lock in the application	
1a. No access	Lock’s name	1. The hardware will indicate. 2. Without access the user won’t be able to press “Lock”	Operational Device (Lock)
1b. User forgot to indicate intention for locking door upon leaving	Lock’s, Household’s name	Application will inform the user by prompt that the door is unlocked, if no other users is in the house, and ask if user wants to lock. Use GPS to determine coordinates. NOTE: Applies only of auto-lock mode is not enabled.	Household, Lock

Task Description	T6. Operate device
Purpose	To remotely operate a device in a household
Trigger	User wants to operate lamps, temperature gauge, or other electrical devices
Frequency	High
Critical	Non-critical
Actor	Administrator, Resident, Guest
Related Requirements	ELV1.1, ELV1.2, ELV1.3, ELV1.4 (High level requirements) FR.D8, FR.D9, FR.D10, FR.D11

Sub-tasks	Visible data	Possible solutions	Virtual windows

1. Find operational device to operate	Operational devices for a household (name)	Search field. List of all operational devices, possibly categorized	Household
1a. No operational devices	-	Alert and quit	
2. Operate device	Operational Devices (specifics)	Present details and options for operating the operational device, as a form.	Operational device
2a. Operational devices is scheduled to be operated now or soon - conflict arose.	The event that is scheduled and conflicts with operation	Ask user for confirmation to override scheduled operation.	Event

3.3.3 Scheduling

The Scheduling subsystem is specified by feature requirements FR.S1 - FR.S10. Furthermore, some of these requirements, which are not trivial, are put in context by Task Descriptions T4 and T5. Guests are not allowed to access schedules (i.e. delete, edit, create or view schedules).

FR.S1	The system shall allow a resident or administrator to see the schedules of a device
Clarification	The times at which the state of the device is scheduled to change.
Related (high-level) requirements	ELV1.5

FR.S2	The system shall allow a resident or administrator to remove a schedule created by the user
Exception	The administrator can do this for all the schedules in a household.
Related (high-level) requirements	ELV1.15

FR.S3	The system shall allow a resident or administrator to set a timer for a power switch adapters to be turned on/off
Related (high-level) requirements	ELV1.13
Comment	A timer, compared to a regular schedule event does not have an end time, and is specified to invoked in a certain amount of time from now, instead of at an exact timestamp. Timers are only used once, then forgotten.

FR.S4	The system shall allow a resident or administrator to set a timer for a light bulb to be turned on with specified brightness and color levels.
Related (high-level) requirements	ELV1.13

FR.S5	The system shall allow a resident or administrator to set a timer for a light bulb to be turned off.
Related (high-level) requirements	ELV1.13

FR.S6	The system shall allow a resident or administrator to create a schedule with events that sets the targeted temperature of rooms
Related (high-level) requirements	ELV1.14

FR.S7	The system shall allow a resident or administrator to create a schedule with events that sets the color and brightness of light bulbs
Related (high-level) requirements	ELV1.13

FR.S8	The system shall allow a resident or administrator to create a schedule with events that turns on/off power switch devices
Related (high-level) requirements	ELV1.13

FR.S9	The system shall allow a resident or administrator to create a schedule with repeated events (the events being repeated every X days)
Related (high-level) requirements	ELV1.13, ELV1.14

FR.S10	The system shall allow a resident or administrator to change a schedule the user created.
Exception	An administrator can change all schedules, including those created by other residents.
Related (high-level) requirements	ELV1.15

Task Description	T4. Create schedule
Purpose	To schedule operation of different devices in a household
Trigger	User wants to schedule lights in to turn on. User wants to schedule temperature changes
Frequency	Medium
Critical	Non-critical
Actor	Administrator, Resident
Related requirements	ELV1.13, ELV1.14 (High level requirements) FR.S6, FR.S7, FR.S8, FR.S9

Sub-tasks	Visible data	Possible solutions	Virtual windows
1. Create schedule - record name of schedule	Schedule’s name	Regular input form. Clickable calendar view	Schedule
2. Find devices to schedule	Operational devices’ name	List of all available device - possibly filter by type. Search.	Operational device
2a. No devices found	-	Alert and quit.	-
3. Record when the devices’ state changes are to take place (events)	Event, date and time details	A clickable calendar view. Datepicker. Include option to periodically schedule, e.g. “every Tuesday”	Event
3a. Scheduled events collides with another schedule	Event date collision details	Inform user. Ask if user want to change or ignore the colliding events. If all events collide, alert and quit.	Event
4. Set the schedules states of the devices. (e.g. on/off for lightbulbs, target temperature, etc)	Operational device, name and state as well as other information (e.g. temperature, brightness)	On/off button. Regular input form for more devices with more specific settings (temperature etc)	Operational device

Task Description	T5. Edit schedule
Purpose	To change scheduled operation of different devices in a household
Trigger	User wants to change schedule due to a change in circumstances.

Frequency	Medium
Critical	Non-critical
Actor	Administrator, Resident
Related requirements	ELV1.15 (High level requirements) FR.S10

Sub-tasks	Visible data	Possible solutions	Virtual windows
1. Find schedule created by user	Schedules by user (name)	Search field. List of all schedules created by user.	Schedule
1a. No schedules found	-	Alert and quit	
2. Choose events to edit	Event time and operational devices	Presents events in a clickable calendar view.	Event, Operational device
3. Change date/time of event(s)	Event, date and time details	Input forms. Datepicker. Include option to change for all future events of the same kind.	Event
3a. Re scheduling of event(s) collides with another schedule	Event date collision details	Inform user. Ask user to input another time	Event
4. Change scheduled states of devices in event	Operational device, name and state as well as other information (e.g. temperature, brightness)	On/off button. Regular input form for more devices with more specific settings (temperature etc)	Operational device

3.4 Software system attributes

This section presents the quality requirements necessary for the application. First, a quality grid presents the traits priority ranking of the product, and then quality requirements grouped in four categories of security, maintainability, usability and efficiency are presented.

3.4.1 Quality grid

The non-functional requirements of the application have been derived using the McCall and Matsumoto quality factors [1]. Each one of these factors was assessed, by the suppliers, based on its importance for the application in a quality grid, as can be seen in Table 3. The suppliers assessed *security* and *reliability* as the two most critical quality factors for the application and *maintainability, testability, flexibility* and *interoperability* as important. A further motivation and explanation of the critical concerns and why some quality factors are considered unimportant for the application can be read under the quality grid. These specific concerns worked as a guideline for the development team to derive the quality requirements for the application which can be found in section 3.4.2 Quality Requirements.

Quality factors for Remote Security System	Critical	Important	As usual	Unimportant	Ignore
Operation					
Security/Integrity	1.				
Correctness			X		
Reliability/Availability	2.				
Usability			X		
Efficiency				7.	
Revision					
Maintainability		3.			
Testability		4.			
Flexibility		5.			
Transition					
Portability			X		

Interoperabili ty		6.			
Reusability				8.	
Installability					9.

Table 3 - Quality grid showing the criticality of quality factors for the application

Critical

1. If someone is able to get access to an account connected to a household he is able to perform operations on household devices, such as unlocking a door. Therefore, it is critical to prevent this.
2. If the application is down it is hard to control the household. This can be very critical if for example the door was unlocked or the stove was on when the user left home. Furthermore, if the user forgot their physical keycard and the application is down the user is locked out of the household.

Important

3. Because the application is operating devices in a household, errors should be found fast in order to prevent miscommunication with the devices. For example an error where the application accidently sends a *Turn-on* command instead of a *Turn-off* command can be fatal for some devices (stove, etc.).
4. As some errors can have fatal consequences to a house (such as turning on a stove, or an iron), the application needs to be easy to test in order to prevent this from accidentally happening.
5. It is important to add new features cautiously because new features to be added should not affect the existing functionality or the performance of the current system in all the three platforms the application is developed for.
6. The application primarily interacts with the centralised cloud service to send and receive data through a specific message format. The application code base must therefore be easy to adapt if there is any change to the cloud service or the data it transfers.

Unimportant

7. As the efficiency of the mobile application is strongly related to the smartphone used, the programmer does not need to take care of this too much.
8. As the mobile application is specifically designed for the smart home solutions of the customer, reusability is not important for this application.

Ignore

9. As the mobile application is installed through the app-stores of the operating systems, installability for the application is of no concern and can be ignored.

3.4.2 Quality Requirements

This section presents all quality requirements for the application. Each requirement have been formulated using the Planguage method [1]. The Gist concept is a bit modified. It is not only a description of what the requirement is about in broad terms, it is also description of why the requirement is important. For some of the requirements the Wish concept is left out as it becomes redundant to the Must concept.

3.4.2.1 Security requirements

This section presents the security and reliability requirements of the application.

QR.S1 Password encryption
Gist: As it would be insecure to store passwords in plain text, the passwords in the system are encrypted with the Bcrypt standard.
Scale: The percentage of the passwords that are encrypted with the Bcrypt standard.
Meter: Measured by the number of passwords stored encrypted divided by the total number of passwords.
Must: 100 %.

QR.S2 Preventing unauthorized usage: password strength
Gist: To avoid unauthorized people to quickly hack an account, the password shall have a length above 10 characters and has to include at least one uppercase as well as one lowercase letter, a special character which is not at the beginning or the end of the password, and at least one numerical character.
Scale: The percentage of passwords that has the strength stated in the gist.
Meter: Measured by the number of password with the strength stated in the gist divided with the total number of passwords. Tested before the passwords are encrypted.
Must: 100 %.

QR.S3 Preventing unauthorized usage: regular change of password
Gist: The application shall require a user to change password on a regular basis to make it more difficult for unauthorized people to get access to the household.
Scale: The time between the user creates his account or change his password until the application requires that he change his password again.
Meter: Measured by examining the time it takes until the application requires a new password.
Must: Every three month.

QR.S4 Secure communication
Gist: It is very important that the application communicates securely with the account and household subsystems since sensitive data is sent on a regular basis. The communication shall be encrypted using SSL.
Scale: The percentage of the communication that is encrypted using SSL.
Meter: Measured by taking the amount of communication that is encrypted using SSL divided by the total amount of communication.
Must: 100 %.

QR.S5 Prevention of unintentional actions
Gist: Users shall not be able to accidentally operate devices in a household, especially not the main door and the critical devices.
Scale: How many times the application accidently adjust the devices in a household during a month's time.
Meter: User studies conducted for a month in ten randomly chosen household.

Must: Maximum of 10 unthought actions, where maximum 0 of these were for the main door or critical devices
Wish: 0 unthought actions

QR.S6 Reliability
Gist: It is important that the application is reliable and has a high availability since it enables users to control whole households including critical events, such as locking the main door and turning off critical devices.
Scale: The percentage of time the application is up and running
Meter: Measured by dividing the total time the application has been up and running with the total time the application has been on the market. Test conducted on accounts for ten randomly chosen households.
Must: System availability of 95 %, hardware and network failures excluded
Wish: System availability of 100 %, hardware and network failures excluded

3.4.2.2 Maintainability requirements

This section presents the maintainability, testability and flexibility requirements of the application.

QR.M1 Cyclomatic complexity
Gist: The cyclomatic complexity of the code shall be small to in order to increase modularity and thereby also maintainability.
Scale: The cyclomatic complexity number.
Meter: Measured by calculating the cyclomatic complexity for each function (McCabe). If a function has higher cyclomatic complexity than required it shall be modified.
Must: Shall not exceed 15.
Wish: Shall not exceed 10.

QR.M2 Duplicated code
Gist: The source code shall contain little duplications to avoid risks of spreading bugs to other parts of the code and increase the understandability of the code.
Scale: Percentage of duplicated code.
Meter: Measured by calculating the percentage of duplicated code (duplicated statements) in the application.
Must: Less than 5 % duplicated code.
Wish: 0 % duplicated code.

QR.M3 Testability
Gist: The requirements for the application shall be able to be tested for verification.
Scale: The percentage of requirements that is able to be verified.
Meter: Measured by dividing the amount of requirements that can be verified by the total amounts of requirements.
Must: 90 % of the requirements are able to be verified.
Wish: 100 % of the requirements are able to be verified.

QR.M4 New versions
Gist: Installation of a new version of the application shall not change personal settings or data contents
Scale: Percentage of how much a new version can change of personal settings and data contents
Meter: Measured by taking the amount of changes of personal settings and data contents of a new version divided by that of an old version
Must: Maximum of 5 %
Wish: 0 %

QR.M5 Flexibility
Gist: To enable a user to be more flexible in his use of the application a user shall be able to be logged in to his account on several smartphones at the same time without losing settings
Scale: Percentage of smartphones that is logged in to an account that has the same settings
Meter: Measured by taking the amount of smartphones that the account is logged in to that have the same settings, divided by the total amount of smartphones that the account is logged in to. Test conducted with 20 randomly chosen smartphones that handles the application.
Must: 90 %
Wish: 100 %

3.4.2.3 Usability requirements

This section presents the usability and portability requirements of the application. A novice user is defined as a user that is using the functionality of the application for the first time, else the user is defined as experienced.

QR.U1 Create a user or guest account
Gist: It shall be easy to add a new user or guest to the household
Scale: The time it takes to create a new user or guest account
Meter: Measured by taking the time it takes for ten novice users and ten experienced users to create a user or a guest account.
Must: Maximum 7 minutes for a novice user and maximum 2 minutes for an experienced user
Wish: Maximum 5 minutes for a novice user and maximum 1 minute for an experienced user

QR.U2 Finding and selecting a device
Gist: It shall be easy to find and select a device in the application
Scale: Number of clicks
Meter: Measured by examining the maximum number of clicks, from anywhere in the application, that has to be made to be able to find and select a device
Must: Maximum of 5 clicks to find a device and 1 click to select a device
Wish: Maximum of 3 clicks to find a device and 1 click to select a device

QR.U3 Operating a device
Gist: It shall be easy to operate a device in the application

Scale: Number of clicks
Meter: Measured by examining the maximum number of clicks that has to be made from selecting a device to have adjusted it
Must: Maximum of 1 click for on/off and lock/unlock (excluding pin code) devices, maximum of 4 clicks for other devices
Wish: Maximum of 1 click for on/off and lock/unlock (excluding pin code) devices, maximum of 2 clicks for other devices

QR.U4 Overview all devices statuses
Gist: It shall be quick and easy to get an overview of the status of all devices in a certain category in a household.
Scale: Number of clicks.
Meter: Measured by examining the maximum number of clicks, from anywhere in the application, that has to be made to get an overview of the status of all the devices in a certain category.
Must: Maximum of 5 clicks.
Wish: Maximum of 3 clicks.

QR.U5 Adjusting a timer
Gist: It shall be easy to adjust a timer for a device.
Scale: Number of clicks.
Meter: Measured by examining the maximum number of clicks that has to be made from selecting the timer for a device until the timer is adjust.
Must: Maximum of 5 clicks.
Wish: Maximum of 3 clicks.

QR.U6 View the time log
Gist: It shall be quick and easy to view the time log for the main door.
Scale: Number of clicks.
Meter: Measured by examining the maximum number of clicks, from anywhere in the application, that has to be made to be able to view the time log.
Must: Maximum of 5 clicks.
Wish: Maximum of 3 clicks.

QR.U7 Set up a schedule for a device
Gist: It shall be easy to set up a schedule for a device.
Scale: Number of clicks.
Meter: Measured by examining the maximum number of clicks that has to be made from selecting the schedule functionality for a device until the schedule is adjust.
Must: Maximum of 5 clicks.
Wish: Maximum of 3 clicks.

QR.U8 Finding the lock button for the main door and off button for critical devices
Gist: It is critical for safety reasons that the lock button for the main door and the off button for the critical devices is very easy to find in the application.

Scale: Time it takes for a user to find the lock button for the main door or the off button for critical devices.
Meter: Measured by taking the mean time for ten novice users and ten experienced users to find the lock button for the main door or the off button for the critical devices.
Must: Maximum of 10 seconds for novice users and maximum 2 seconds for experienced users.
Wish: Maximum of 4 seconds for a novice users and maximum of 1 second for experienced users.

QR.U9 Easy installation and Portability
Gist: The application shall be portable and easy to install
Scale: Number of hardware and software platforms the application support.
Meter: Measured by the number of smartphones the application is able to operate on.
Must: Be available in App Store, Play Store, and Marketplace, on smart devices with at least Android v4.0, iOS 7 or Windows Phone 7.8
Wish: Be available to use on all smartphones.

3.4.2.4 Efficiency requirements

This section presents the performance, capacity and accuracy requirements for the application. The requirements under this section assume that the system is working under normal conditions (no network failures, bad reception etc.). Many of these requirements are dependant also on the cloud services of the system, i.e. the Account Management system and the Household systems described in 2.4 Product context. Therefore, these requirements does not solely concern the application, but rather the whole Remote Security System. However, the application can largely affect the outcome of these requirements, and they are therefore stated here.

QR.E1 Response time
Gist: The system has a reasonable response time.
Scale: The time it takes from adjusting a device in the application until the actual device responds.
Meter: Measured by taking the average time it takes for someone to operate a device in the application until the physical device responds. Test conducted on accounts for ten randomly chosen households.
Must: Maximum of 2 second in 95 % of the tests.
Wish: Maximum of 500 milliseconds 95 % of the tests.

QR.E2 Response time for locking the main door
Gist: The response time for locking the main door is a critical task and should be faster, if possible, than the response time for other devices.
Scale: Time it takes from clicking the lock button for the main door until the main door is locked.
Meter: Measured by taking the average time it takes from someone clicks the lock button until the main door responds. Test conducted on accounts for ten randomly chosen households.
Must: Maximum of 800 milliseconds in 95 % of the tests.

Wish: Maximum of 400 milliseconds.

QR.E3 Response time for critical devices
Gist: The response time for turning a critical device off is a critical task and should be faster, if possible, than the response time for other devices.
Scale: Time it takes from clicking the off button for a critical device until the critical device is turned off.
Meter: Measured by taking the average time it takes from someone clicks the off button until the critical device responds. Test conducted on accounts for ten randomly chosen households.
Must: Maximum of 500 milliseconds in 95 % of the tests.
Wish: Maximum of 100 milliseconds in 95 % of the tests.

QR.E4 Time to detect new devices
Gist: The application shall detect new devices fast that is added to the household, so the users can start to control them from the application as soon as possible.
Scale: The time it takes from that a new devices is installed until the application has detect it.
Meter: Measured by taking the mean time it takes for ten new devices to be detected in ten randomly chosen households.
Must: Less than 5 minutes.
Wish: Less than 30 seconds.

QR.E5 Devices simultaneously controlled in each household
Gist: There was stated, after talking with customers, that a room often consists of around 5 devices that can be controlled by the application. A house often consists of around 7-8 rooms. This is around 40 devices per household to be controlled. To be safe to even cover real big household the 40 devices per household estimation was multiplied by 5. This led to a conclusion that the application must be able to control at least 200 devices. The system have to handle simultaneously usage of these devices.
Scale: Number of devices that can be simultaneously (within < 1 second) controlled in a household.
Meter: The amount of devices which can be simultaneously controller. Measured by issuing simultaneously requests to different devices in a household.
Must: 20 devices.
Wish: As many devices as possible without having a negative effect on the usability.

QR.E6 Validity of guest accounts
Gist: Guest accounts shall only be temporary and should have an expiry date. This is to avoid that accounts with access shall be forgotten or be used by unauthorized users.
Scale: The time it takes from creating a guest account to expire.
Meter: Measured by that ten different guest account in ten randomly chosen households expires by their expiry date, or if not having one, six months after it was created.
Must: Maximum validity of 6 months.
Wish: Only valid when the guest shall be allowed to access a household.

3.5 Design-level requirements

The design-level requirements of the application is in this section presented in form of mock-ups.

3.5.1 Mock-ups

Figure 9 and 10 shows storyboards from the prototype of the application, of two different use cases. They serve the purpose of illustrating how some of the use cases can be implemented in the application. All use cases are not included, the ones that are included are those in Release 1 in the release planning. The prototype is interactable and can be found in its entirety at http://domein.co.at/remotesecurity/. Its main intention is to show how the application will provide the data as well as the functionality required by this SRS. To see the prototype as intended you may have to zoom out in the browser. The final application is not required to look like, or function as the prototype. However, the final application should aim to implement a solution similar to the prototype.

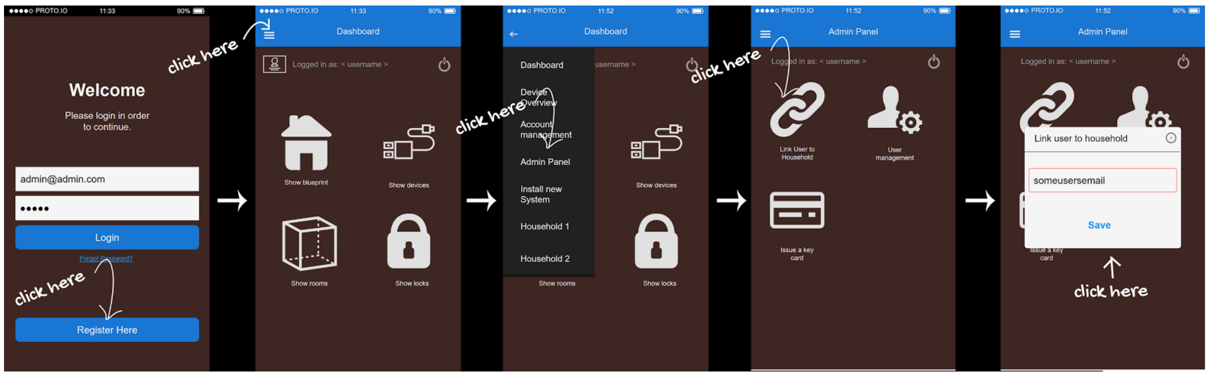


Figure 9 - A storyboard on how UC1, link a user to a household, would be performed by a user. Going from left to right, Login screen, Dashboard, Dashboard with menu, Admin panel, Admin panel linking user.

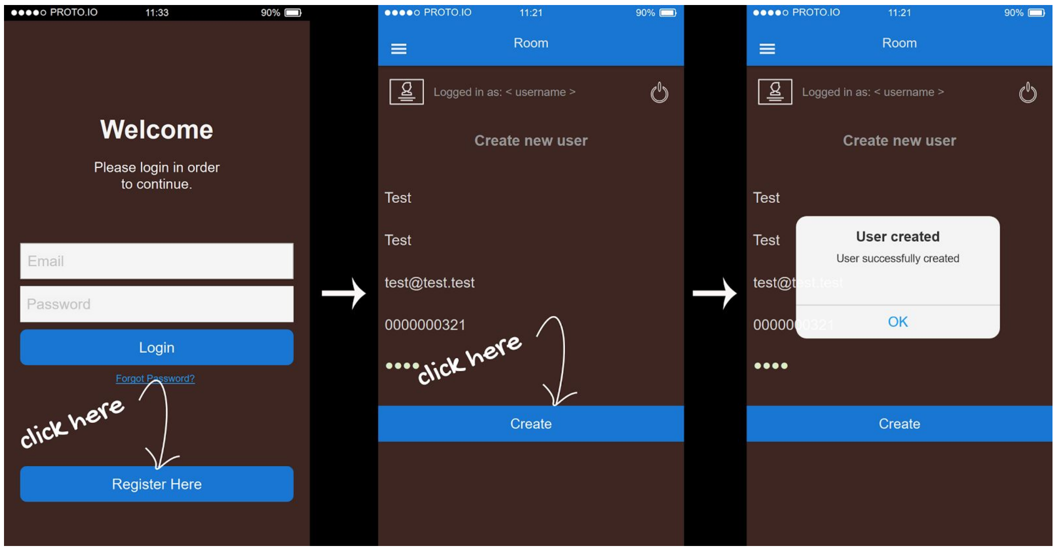


Figure 10 - A storyboard on how UC2, create an account, would be performed by a user. Going from left to right, Login screen, Registration form, Registration form confirming new user created.

4 Prioritization and Release Planning

This section will present the results from some of the prioritization techniques used for this project. Furthermore, it will explain how the results of the prioritization was used for the *Release Planning* of the future releases.

4.1 Prioritization

Using the values obtained by the numerical assignment and the hundred-dollar test explained in Appendix A, the requirements were sorted based on their final value. Requirements with higher values were prioritized. Table 4 shows the top twenty requirements which were prioritized the highest, and with respect to that used as a basis for the release planning.

1.	UC2	Link user to household	690.9
2.	UC1	Create Account	651.7
3.	UC3	Login	567.42
4.	UC10	Install a new system	477.75
5.	UC12	Issue a key to a user	409.15
6.	FR.D1	A user should be able to see the status of an operational device	372.45
7.	UC4	Logout	356.23
8.	T4	Create schedule	312.6
9.	UC13	Manage key access rights	260.19
10.	FR.D13	A user should be able to lock a door from the application.	259.35
11.	UC6	Manage access to devices	242.55
12.	FR.D8	A user should be able to turn on/off a power switch adapter	226.2
13.	UC5	Change password	219.52
14.	FR.D6	A user should be able to see the name of a device (operational and informational device)	214.5
15.	T1	A user should be able to unlock a door via NFC, given that a key of his grants access to said door	214.5
16.	FR.D20	A user should be able to find, by searching, operational and informational devices in a household	208.65
17.	FR.D12	A user should be able to lock a door via NFC, given that a key	198.9

		of his grants access to said door.	
18.	T5	Edit schedule	197.4
19.	T2	A user should be able to unlock a door from the application, given that a key of his grants access to said door given that he has entered a valid personal pin code	195
20.	FR.D21	A user should be able to browse operational and informational devices in a household	185.25

Table 4 - Top twenty requirements with highest prioritizing

The requirements were also prioritized using the Numerical Assignment (Grouping) method. In this method the requirements were divided into groups to define the priority of requirements. The requirements are mainly classified into groups [4] such as

1. Mandatory (project cannot go live without it)
2. Very Important (project needs them and customers want them)
3. Important (customer appreciates the inclusion)
4. Not Important (can live without them)
5. Optional (one hardly notices its presence or absence)

The following table has the requirement IDs grouped and prioritized as per the Numerical Assignment (Grouping) method. In order to avoid the customers grouping most of the requirements under the Mandatory category, the customers are were required to group at least 6 requirements in each group [5].

S.No	Mandatory	Very Important	Important	Not important	Optional
1	UC2.	UC6	UC4	UC9	FR.S4
2	UC1	FR.D6	T4.	FR.S2	FR.D10
3	UC3	T2	UC5	FR.A2	FR.D3
4	UC10	FR.D21	FR.D11	FR.D9	FR.S6
5	UC12	T3	FR.A3	UC7.	FR.A4
6	FR.	FR.A1	FR.S3	FR.S5.	FR.D18
7	UC13.	FR.D15	FR.D17	FR.D2	FR.S1
8	FR.D13	FR.D14	FR.D16	FR.D4	-
9	FR.D8	FR.D5	FR.D19	FR.D7	-

10	T1	FR.D12	FR.D20	UC8	-
11	-	-	T5	UC14	-

Table 5 -

In order to improve the consistency of the results of the prioritization, the requirements are further prioritized using AHP (Analytic Hierarchy Process). In the AHP method the requirements are compared with one another to know their relative importance to each other. This involves pair-wise comparison along with a cost-value approach. The requirements were pairwise compared with each other with the *value* aspect assigned by the customers and *cost* aspect by the developers.

The Results from the AHP are more clear than the other prioritization methods as a requirement are compared based on both cost and value. Detailed explanation of this method is given in appendix A. The results from the AHP process are plotted in graphs and are shown in figure 11.

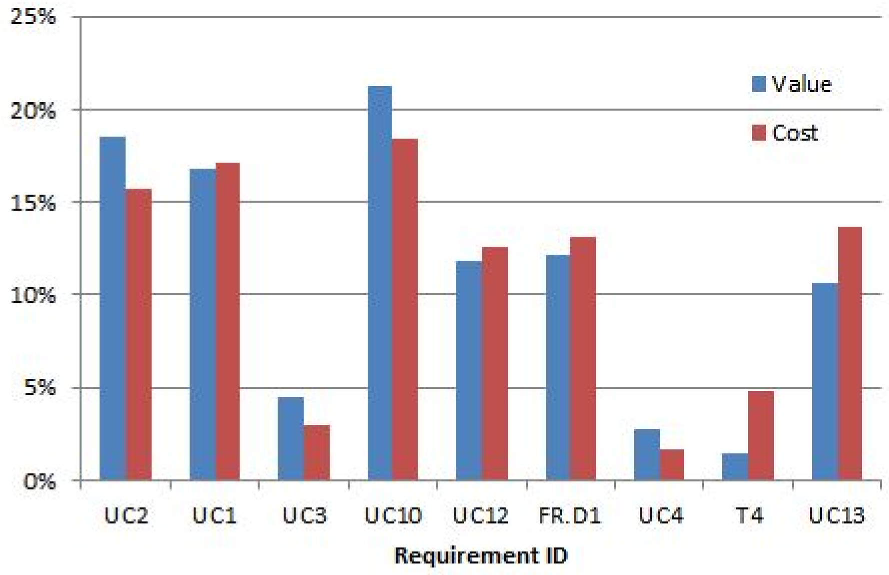


Figure 11 - Graph comparing each requirement with its cost and value

It is inferred from Figure 11 that requirements UC2, UC1, and UC 10 together cost 50% of the total cost of the nine requirements and they contribute to around 57% of the value amongst the top nine requirements. For further analysis the scores of value and cost were plotted against each other in a XY graph as seen in Figure 12.

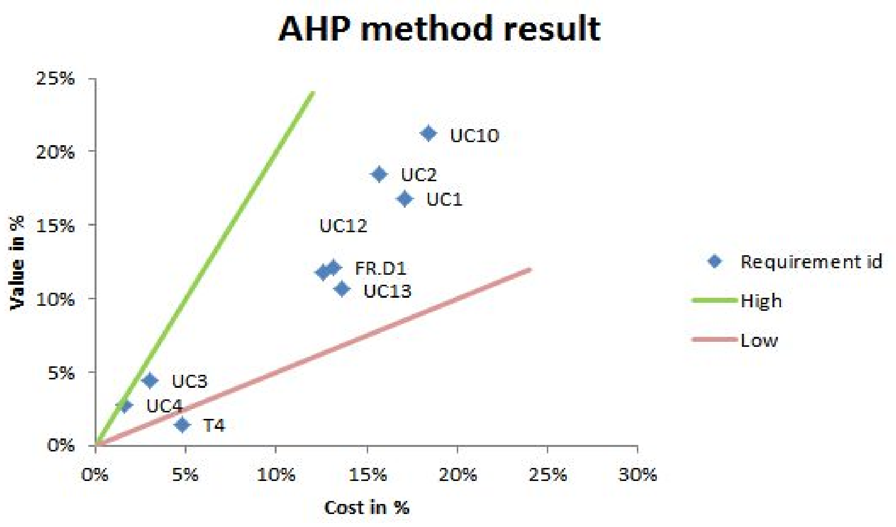


Figure 12 - AHP results plotted in a graph showing cost (%) on the x-axis, and value (%) on the y-axis. The High and Low lines indicate ideal high and low prioritization respectively.

From Figure 12 it is inferred that the requirement id UC4 and UC3 lies close to the high priority line. The requirement id T4 is under the low priority line. The requirements between high and low priority lines are the requirements with medium priority.

4.2 Release Planning

The release planning was performed for three future releases, R1, R2 and R3. The requirements was planned by calculating and maximizing a *Weighted Average Satisfaction* value. More details regarding the methods used can be found in Appendix A. The order of the requirements included in the releases are based on priority, with highest priority on top.

Release 1

Requirement	**Description**
UC2	Link user to household
UC1	Create account
UC3	Login
UC10	Install a new system
UC12	Issue a key to a user
FR.D1	A user should be able to see the status of an operational device
FR.D13	A user should be able to lock a door from the application
FR.D20	A user should be able to find, by searching, operational and informational devices in a household

Release 1 includes functionality for installing a new system (creating account, logging in, linking new users to household), finding devices and locking doors. The release also includes functionality for issuing a key to a user, thus physical key cards can be bound to keys and be used to unlock doors. After release 1 the system's most essential functionality will be implemented and the system will be able to be used in a somewhat restricted manner. The locking functionality is there, which is the most important functionality according to the customers. However, operating other devices will be included in release 2.

Release 2

Requirement	**Description**
T1	A user should be able to unlock a door via NFC, given that a key of his grants access to said door
FR.D12	A user should be able to lock a door via NFC, given that a key of his grants access to said door.
T2	A user should be able to unlock a door from the application, given that a key of his grants access to said door given that he has entered a valid personal pin code
FR.D21	A user should be able to browse operational and informational devices in a household
FR.D8	A user should be able to turn on/off a power switch adapter

Release 3

Requirement	**Description**
T4	Create Schedule
T5	Edit Schedule
UC4	Logout
UC13	Manage key access right
UC6	Manage access to devices
UC5	Change password
FR.D6	A user should be able to see the name of a device (operational and informational device)
FR.S2	A user should be able to remove a schedule created by him

Table 6 - Specification for the three different releases

5 Supporting information

5.1 References

[1] S. Lauesen, ”Software Requirements Styles and Techniques”, 2002.
[2] G.Ruhe, M.O.Saliu, “The Art and Science of Software Release Planning”, 2005.
[3] Nancy Mead,”Requirements Prioritization Case Study Using AHP,2006”
[4] Aniket S Sharma, “BA Techniques,Requirements Prioritization”,20131]
[5] Patrik Berander, Anneliese Andrews, “Requirements Prioritization,” in Engineering and Managing Software Requirements, First ed.2005 Springer Berlin Heidelberg
[6] Henk C.A. van Tilborg, E Jajodia, Sushil, Encyclopedia of Cryptography and Security 978-1-4419-5905-8, I Springer US, 2011 , 157-158
[7] Dubitzky Werner, Wolkenhauer Olaf, Cho Kwang-Hyun, Yokota, Hiroki, Encyclopedia of Systems Biology, 978-1-4419-9862-0, Springer New York, 2013 ,926-926
[8] Richard Berntsson Svensson , Tony Gorschek, Björn Regnell , Richard Torkar, Ali Shahrokni , Robert Feldt , Aybuke Aurum ,”Prioritization of Quality Requirements: State of Practice in Eleven Companies” presented at Requirements Engineering Conference (RE), 2011 19th IEEE International
[9] JSON, http://www.json.org/, accessed: 2015-10-15
[10] Hui Xiong, Sasha Shekhar, Representational State Transfer Services, Encyclopedia of GIS, 978-0-387-30858-6, Springer US, 2008
[11] Li, StanZ., Jain, Anil, Near Field Communication, Encyclopedia of Biometrics, 978-0-387-73002-8, Springer US, 2009, 1001-1001
[12] HowStuffWork, http://electronics.howstuffworks.com/gadgets/high-tech-gadgets/rfid.htm, accesed: 2015-10-16
[13]

5.2 Appendix A - Prioritization and release planning methods explained

5.2.1 Numerical Assignment and The Hundred-dollar test

Methods used for the prioritization were *numerical assignment* and the *hundred-dollar test*. The first method was used to divide the requirements into different categories, namely *account*-, *device*- and *schedule-management*. This was done because of their relationships to each other as explained in 4.3. The latter method was used when all the requirements had been assigned to a category. A *hundred-dollars* were to be divided amongst the *three categories*, while a *thousand-dollars* were to be divided amongst the requirements inside a category. Equation [1] was used to evaluate the final value for a requirement where v_i is the final value for requirement *i*, c_j is the total amount of dollars given to one of the subcategories, c_{tot} is the total amount spent amongst all the subcategories and ri_{tot} is the total amount given to requirement i. Basically this means that they are weighted based on what category the requirements belongs to. They idea with weighting them this way is enable the developers to prioritize the components that are the most important for a working system whilst the customers can weight in a way that make some requirements more valuable even if they are not as important for a working system.

$$v_i = c_j/c_{tot} * ri_{tot} \quad (1)$$

5.2.2 Methods for release planning

A normalization was done on the prioritization values of the requirements so they would range from value 1-9. The stakeholders were also separated into *customers* and *suppliers* to make it possible to weight the values according to the importance of the different stakeholders. With the stakeholders grouped into *customers* and *suppliers* the values for each requirement were recalculated using equation 2. The values obtained were then used as part of the release planning.

Values representing the urgency for a requirement to be done for a certain release was added in addition to the prioritization values. Urgency values were divided amongst the three releases and the total amount placed on each release added up to 9. A *weighted average satisfaction* value was then calculated.

The *weighted average satisfaction* value was then maximized which assigned the requirements to one of the releases. However some of the requirements were postponed to release 3 since they depend on other requirements to be implemented. The results obtained for the release planning is presented in the table below.

The *weighted average satisfaction* given by equation 3 where:

- $\xi(k)$ is the importance of a release
- $\lambda(p)$ the importance of a stakeholder or in our case stakeholders
- $value(p, i)$ is the normalized prioritization value mentioned above
- $urgency(p,\ i,\ k)$ the urgency for a stakeholder to have a requirement *i* in release *k*.

$$F(x) \ = \sum_{k\,=\,1...K} \ \sum_{i:x(i)\,=\,1...K} WAS(i,\ k) \quad (2)$$

$$WAS(i,k) \ = \ \xi(k)[\sum_{p=1...q} \lambda(p) * value(p,i) * \ urgency(p,\ i,\ k)] \quad (3)$$

This method of release planning was presented by [2] but have been implemented with slight modifications as this SRS do not take resources into account when doing the planning.

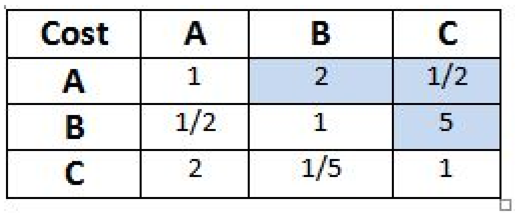
5.2.3 Analytical Hierarchy Process (AHP)

The AHP involves pair wise comparison of requirements. As the requirements are compared with each other , it is very difficult to perform the comparisons when there are more number of requirements. In other words higher the number of requirements higher the number of comparisons. In order to avoid that only the top 9 requirements are chosen from the 100 dollar test. This nine requirements are compared with one another and appropriate weights(Table 6.3.2) are assigned to each pair.For nine requirements there will be 36 pairs of comparisons. This can also be understood from the formula

Total number of comparisons to perform with AHP = $n \times (n-1)/2$
(where n is the number of requirements) [3].

As said earlier to make this prioritization less sophisticated the top nine requirements are chosen from the result of 100 dollar test This set of requirements will be compared with each other based on cost and value. The cost-value approach uses a two dimension graph that displays the requirements value against its cost. AHP is used from a customer and user perspective, to assess the value of each requirement. This is further followed by an assessment of the requirements cost from an implementation perspective ie from the side of developers.

To compare the requirements ids are projected over a matrix (row and column wise) . This matrix is known as the prioritization matrix. In this case it will be a matrix with 81 (9 x 9) cells for nine requirements. The upper half of the matrix is filled with weights for every pair wise comparison and each requirement had a value of "1" when compared to itself. Diagonally the values will always be one as the requirements are compared to itself. The lower part of the matrix are filled with the reciprocals of the value from the appropriate cells of the upper half of matrix.



Cost	A	B	C
A	1	2	1/2
B	1/2	1	5
C	2	1/5	1

Table. 6.3.1 Sample table explaining the pair wise comparison

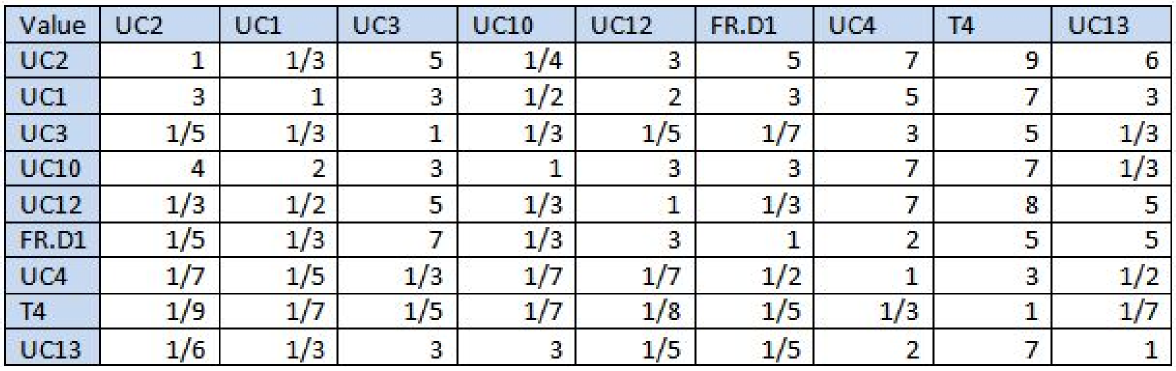
A sample of pair wise comparison is shown in Table. 6.3.1 . As explained earlier the diagonal values will be one as the requirements are compared to itself . The upper part of the matrix ie the highlighted area should be filled by the user by comparing the requirements. The lower half can be filled by taking the reciprocals of the corresponding cells from the upper half. Example in table 6.3.1 shows a pair wise comparison . When A is compared to B the value is 2 so when B is compared to A it is ½ which is nothing but the reciprocal value.

An arbitrary entry in row i and column j of the matrix, labeled a_{ij} , indicates how much higher (or lower) the value/cost for requirement i is than that for requirement j [3].The value/cost is measured on an integer scale from 1 to 9, with each number having the interpretation shown in Table 6.3.2

Intensity of Value	Interpretation
1	Requirements i and j are of equal value or equal cost .
3	Requirement i has a slightly higher value or higher cost than j.
5	Requirement i has a strongly higher value or higher cost than j.
7	Requirement i has a very strongly higher value or higher cost than j.
9	Requirement i has an absolutely higher value or higher cost than j.
2, 4, 6, 8	These are intermediate scales between two adjacent judgments.
Reciprocals	If Requirement i has a lower value or lower cost than j

Table 6.3.2

Values are substituted based on the interpretation table 6.3.2 while doing pair wise comparison and two matrixes are obtained ie one with the aspect value and other with the aspect Cost.Table 6.3.3 shows the comparison with the aspect value and Table 6.3.4. shows the comparison with the aspect cost.



Value	UC2	UC1	UC3	UC10	UC12	FR.D1	UC4	T4	UC13
UC2	1	1/3	5	1/4	3	5	7	9	6
UC1	3	1	3	1/2	2	3	5	7	3
UC3	1/5	1/3	1	1/3	1/5	1/7	3	5	1/3
UC10	4	2	3	1	3	3	7	7	1/3
UC12	1/3	1/2	5	1/3	1	1/3	7	8	5
FR.D1	1/5	1/3	7	1/3	3	1	2	5	5
UC4	1/7	1/5	1/3	1/7	1/7	1/2	1	3	1/2
T4	1/9	1/7	1/5	1/7	1/8	1/5	1/3	1	1/7
UC13	1/6	1/3	3	3	1/5	1/5	2	7	1

Table 6.3.3 Prioritization matrix (value)

Cost	UC2	UC1	UC3	UC10	UC12	FR.D1	UC4	T4	UC13
UC2	1	2	5	1/2	3	3	7	4	1/3
UC1	1/2	1	7	3	1/3	3	7	2	2
UC3	1/5	1/7	1	1/3	1/3	1/5	3	1/2	1/5
UC10	2	1/3	3	1	3	5	9	3	2
UC12	1/3	3	3	1/3	1	1/3	5	5	2
FR.D1	1/3	1/3	5	1/5	3	1	3	7	3
UC4	1/7	1/7	1/3	1/9	1/5	1/3	1	1/7	1/7
T4	1/4	1/2	2	1/3	1/5	1/7	7	1	1/7
UC13	3	1/2	5	1/2	1/2	1/3	7	7	1

Table 6.3.4 Prioritization matrix (Cost)

From the prioritization matrix, the normalised matrix (Table 6.3.5 & 6.3.6) is obtained. For each cell will be divided by the sum of the cells in that particular column. This is done for values in both the matrices (Table 6.3.3 & Table 6.3.4).

	UC2	UC1	UC3	UC10	UC12	FR.D1	UC4	T4	UC13	SCORE
UC2	0.1092	0.0644	0.1816	0.0414	0.2368	0.3738	0.2039	0.1731	0.2816	0.18508
UC1	0.3277	0.1932	0.1090	0.0828	0.1579	0.2243	0.1456	0.1346	0.1408	0.16843
UC3	0.0218	0.0644	0.0363	0.0552	0.0158	0.0107	0.0874	0.0962	0.0156	0.04482
UC10	0.4370	0.3864	0.1090	0.1657	0.2368	0.2243	0.2039	0.1346	0.0156	0.21258
UC12	0.0364	0.0966	0.1816	0.0552	0.0789	0.0249	0.2039	0.1538	0.2346	0.11845
FR.D1	0.0218	0.0644	0.2542	0.0552	0.2368	0.0748	0.0583	0.0962	0.2346	0.12181
UC4	0.0156	0.0386	0.0121	0.0237	0.0113	0.0374	0.0291	0.0577	0.0235	0.02766
T4	0.0121	0.0276	0.0073	0.0237	0.0099	0.0150	0.0097	0.0192	0.0067	0.0145
UC13	0.0182	0.0644	0.1090	0.4970	0.0158	0.0150	0.0583	0.1346	0.0469	0.10657

Table 6.3.5 Normalised matrix (Value)

	UC2	UC1	UC3	UC10	UC12	FR.D1	UC4	T4	UC13	SCORE
UC2	0.1289	0.2515	0.1596	0.0792	0.2594	0.2248	0.1429	0.1349	0.0308	0.15688
UC1	0.0644	0.1257	0.2234	0.4754	0.0288	0.2248	0.1429	0.0675	0.1849	0.17086
UC3	0.0258	0.0180	0.0319	0.0528	0.0288	0.0150	0.0612	0.0169	0.0185	0.02987
UC10	0.2577	0.0419	0.0957	0.1585	0.2594	0.3747	0.1837	0.1012	0.1849	0.18418
UC12	0.0430	0.3772	0.0957	0.0528	0.0865	0.0250	0.1020	0.1687	0.1849	0.12619
FR.D1	0.0430	0.0419	0.1596	0.0317	0.2594	0.0749	0.0612	0.2361	0.2773	0.13167
UC4	0.0184	0.0180	0.0106	0.0176	0.0173	0.0250	0.0204	0.0048	0.0132	0.01614
T4	0.0322	0.0629	0.0638	0.0528	0.0173	0.0107	0.1429	0.0337	0.0132	0.04772
UC13	0.3866	0.0629	0.1596	0.0792	0.0432	0.0250	0.1429	0.2361	0.0924	0.13643

Table 6.3.6 Normalised matrix (Cost)

Once the normalised matrix is filled then the scores of the requirement can be calculated from it. Here a row wise operation is performed. Add the values in the cells row wise and divide ti with the number of cells which is nothing but the row wise average .The average value of each row in normalised matrix gives the score for each requirement id. This process is done for both the cost and value based comparisons. Thus there will be two normalised matrices ,one for cost and the other for value. The scores from the corresponding matrices (Table 6.3.5 & 6.3.6) are converted into percentage values. This scores in percentage is used for plotting against each other in graphs

5.3 Appendix B - Results from the end-user survey conducted

Q1 - Autlock-Mode		**Q2 - Timer functionality (1 = low, 5 = high)**				
Yes	**No**	**1**	**2**	**3**	**4**	**5**
12	7	4	1	5	5	4
Comments						
Not all doors in the house, just the doors leading to the outside should be locked						
I like my home how it is now as simple as possible						
depending on the key to get in. Has to be safe						
That's smart, but at the same time I can see that you would be locked out because the key is still left inside.						
If i forget the key then??						
It should not be possible to lock yourself out.						
I would like the ability to do so but would probably not use it. I tend to forget my keys sometimes and an autolock function may increase the chances of getting locked out.						

Q3 - Notification if door unlocked (1 = low, 5 = high)					Q4 - Guest Accounts	
1	**2**	**3**	**4**	**5**	**Yes**	**No**
0	1	3	6	9	14	5
					Comments	
					In my view thats too dangerous. People can steel ur phone and steel your things	
					Provided I could revoke that guest access at any time.	

Q5 - Timelog	**Q6.1 - Adjust brightness of lights**				
	1	**2**	**3**	**4**	**5**
	2	2	1	6	8
Comments	**Q6.2 - Adjust temperature of each room**				
It has its ups and downs, you know when something fishy is going on but your gitlfriend etc. exactly knows when you got home drunk as fuck last night or in the morning.	**1**	**2**	**3**	**4**	**5**
definitely useful - maybe luxury stuff but i would buy it :P	1	5	3	0	5
half half.... it is good to see it, but it depends how you are using it by yourself, as controll for the kids or for other situations. But yes i would like to have this function!	**Q6.3 - See air quality**				
It depents were the data is stored. If the data is on a local server in the House and you can acces it just for a local terminal it is ok. If the data is stored on a server where other people or companies can acces the data I would not like to have this data stored.	**1**	**2**	**3**	**4**	**5**
Good idea	2	5	5	1	6
would be okay	**Q6.4 - Turn on/off electronical devices**				
Reasonable	**1**	**2**	**3**	**4**	**5**
gooood	1	1	3	4	10
jk	**Q6.5 - Lock/Unlock doors**				
Not necessary.	**1**	**2**	**3**	**4**	**5**
It feels like there's a possibility to hack such a log so that unauthorized is able to get to much and wrong information.	2	3	5	0	9
Works good at companies where staff has different disarming alarm code. Good as a selectable choice even in private homes, could be important that this information is given to the users when unlocking/locking the door.	**Q6.6 - Operate blinds/curtains**				
sounds pretty good	**1**	**2**	**3**	**4**	**5**
ofcourse, for the admin	1	3	2	8	5
I don't really see the usefulness, but why not.	**Q6.7 - Open/Close windows**				
Not a necessity, but it'd be convenient.	**1**	**2**	**3**	**4**	**5**
Awesome	5	5	4	1	4
Not a necessity but something that would be useful. Kind of like, if you need it and	**Q6.8 - Restrict access for children**				

it's not there, it would be worse than having it and not needing it.							
Not good for privacy.			**1**	**2**	**3**	**4**	**5**
			5	2	3	5	4

Q6.1 - Adjust brightness of lights					**Q7 - Other functionalities**
1	**2**	**3**	**4**	**5**	
2	2	1	6	8	
Q6.2 - Adjust temperature of each room					**Comments**
1	**2**	**3**	**4**	**5**	Lights go on when you enter the house A smartphone controlled audiosystem individual for some rooms but also linkable A panic button when shit goes down or pulling of awesome pranks
1	5	3	0	5	smart purchase list on the fridge - when groceries become empty it's showing up on the purchase list on the fridge . which is connected with the smartphone ..
Q6.3 - See air quality					In the moment nothing...
1	**2**	**3**	**4**	**5**	An integrated sound system for individual music in each room controlled with a smartphone.
2	5	5	1	6	I wouldn't want these things to be controlled by my mobile device
Q6.4 - Turn on/off electronical devices					nothing
1	**2**	**3**	**4**	**5**	Sound system, cameras, self-ordering devices meaning for example a light bulb will fail soon it already orders a new one
1	1	3	4	10	video streaming (surveillance) burglary, start lawn mower/vaccuum cleaner robot, start coffee brewer, open/close cat/dog flap, control/time stove/oven
Q6.5 - Lock/Unlock doors					npjk
1	**2**	**3**	**4**	**5**	None right now
2	3	5	0	9	I could imagine that you could for example turn on the coffe brewer with a push of a button, or adjust the temperature as time passes, namely you could make it more energy efficient by having everything turned on only when used.
Q6.6 - Operate blinds/curtains					Possiblity to schedule activities such as light, music, alarm clock, coffe ect.
1	**2**	**3**	**4**	**5**	Statistics in the smartphone app, average temperature in each room, how does it change over time, after opening a window, how and how fast does it effect temprature and air qiality?
1	3	2	8	5	CCTV for the front door and house entrance. In case of fire the alarm will be sent to your phone and CCTVs inside will turn on and show you. A smart fridge which senses what foods you need to buy.
Q6.7 - Open/Close windows					- To see electrical usage for each room/device.

1	**2**	**3**	**4**	**5**	Monitor smoke/gas alarms (if that's not included in 'air quality') and water taps.
5	5	4	1	4	Notifcation if I forgot to turn oven off or the pans
Q6.8 - Restrict access for children					Adjust the humidity of each room with humidifier/de-humidifiers and sensors to know what lights are on in each room.
1	**2**	**3**	**4**	**5**	Security is the most important things for my house and enjoying hobby utilities would be useful at a sweet g=home. Suitably changing the light depends on my conditions every day can make me feel more good as light can make people feel different.
5	2	3	5	4	

Q7 - What other functionality would be important for you / can you think of?
Lights go on when you enter the house A smartphone controlled audiosystem individual for some rooms but also linkable A panic button when shit goes down or pulling of awesome pranks
smart purchase list on the fridge - when groceries become empty it's showing up on the purchase list on the fridge . which is connected with the smartphone ..
In the moment nothing...
An integrated sound system for individual music in each room controlled with a smartphone.
I wouldn't want these things to be controlled by my mobile device
nothing
Sound system, cameras, self-ordering devices meaning for example a light bulb will fail soon it already orders a new one
video streaming(övervakning) inbrott, starta gräsklippare/damsugare(robot), starta kaffebryggare, öppna/stänga katt/hundlucka, styrning/timer spis/ugn... glhf //Rocco
npjk
None right now
Jag skulle tänka mig att tex sätta igång kaffebryggaren med en tryckning, eller justera temperatur överlag med tiden, dvs kunna energieffektivisera så att allt bara är igång när det används
Möjlighet till att Schemalägga aktiviteter så som ljus, musik, väckarklocka, kaffe etc
Statistics in the smartphone app, average temperature in each room, how does it change over time, after opening a window, how and how fast does it effect temprature and air

qiality?
CCTV for the front door and house entrance. In case of fire the alarm will be sent to your phone and CCTVs inside will turn on and show you. A smart fridge which senses what foods you need to buy.
- To see electrical usage for each room/device.
Monitor smoke/gas alarms (if that's not included in 'air quality') and water taps.
Notifcation if I forgot to turn oven off or the pans
Adjust the humidity of each room with humidifier/de-humidifiers and sensors to know what lights are on in each room.
Security is the most important things for my house and enjoying hobby utilities would be useful at a sweet g=home. Suitably changing the light depends on my conditions every day can make me feel more good as light can make people feel different.

Q8 - What age are you?
18
25
32
22
24
27
29
24
21
21
25
26
19
25
24
35
20
27
24

Q9 - Gender (M/F)?
M
M
F
M

F
F
F
M
F
F
F
F
M
F
F
M
F
F
M

Q10 -Have you been an owner of a smart home before (Y/N)?
N
N
N
N
N
N
N
N
N
N
N
N
N
N
N
N
N
N
N