

System Specification

Dahoam

Project Name	Dahoam
Project Leader	I. Gattringer
Document state	In process
Version	V. 1.0

Revisions

Date	Author	Change
November 03, 2019	P. Bauer	First version

Inhaltsverzeichnis

1 Initial Situation and Goal	5
1.1 Initial Situation	5
1.1.1 Application Domain	5
1.1.2 Glossary	6
1.1.3 Model of the Application Domain	6
1.2 Goal	7
2 Functional Requirements	8
2.1 Overview	8
2.2 Use Case 1: Delete User	8
2.2.1 General Description	8
2.2.2 UI to Call the Use Case	9
2.2.3 The Standard Use	9
2.2.4 The Non-Standard Use	10
2.3 Use Case 2: First Configuration of the Mirror	10
2.3.1 General Description	10
2.3.2 UI to Call the Use Case	11
2.3.3 The Standard Use	11
2.3.4 The Non-Standard Use	12
2.4 Use Case 3: Install Smart Home Interfaces	13
2.4.1 General Description	13
2.4.2 UI to Call the Use Case	14
2.4.3 The Standard Use	14
2.4.4 The Non-Standard Use	15
2.5 Use Case 4: User Administration	15
2.5.1 General Description	15
2.5.2 UI to Call the Use Case	16
2.5.3 The Standard Use	16
2.5.4 The Non-Standard Use	17
2.6 Use Case 5: Sleep Mode	17
2.6.1 General Description	17
2.6.2 The Standard Use	17
2.6.3 The Non-Standard Use	18
2.7 Use Case 6: Wake up from Sleep Mode	18
2.7.1 General Description	18
2.7.2 The Standard Use	18

2.7.3	The Non-Standard Use	18
2.8	Use Case 7: Creating First Admin	18
2.8.1	General Description	18
2.9	Use Case 8: Defining Personal Location	19
2.9.1	General Description	19
2.9.2	The Standard Use	19
2.9.3	The Non-Standard Use	20
2.10	Use Case 9: Registration Flow	20
2.10.1	General Description	20
2.10.2	The Standard Use	20
2.10.3	Registration aborted by timeout	21
2.10.4	Registration aborted by user	21
2.10.5	Activity Diagram	22
2.11	Use Case 10: Alternative to QR-Code	23
2.11.1	General Description	23
2.11.2	The Standard Use	23
2.11.3	The Non-Standard Use	23
2.12	Use Case 11: Admin Account	23
2.12.1	General Description	23
2.13	Use Case 12: Demo Mode	24
2.13.1	General Description	24
2.13.2	The Standard Use	24
2.13.3	The Non-Standard Use	24
2.14	Use Case 13: Kiosk Auto Start	24
2.14.1	General Description	24
2.14.2	The Standard Use	24
2.14.3	The Non-Standard Use	25
2.15	Use Case 14: Session Management	25
2.15.1	General Description	25
2.15.2	The Standard Use	25
2.15.3	The Non-Standard Use	25
3	Non-Functional Requirements	26
3.1	NFR 1: Enable Full Dockerized Runtime Environment	26
3.2	NFR 2: Development Environment	26
3.3	NFR 2.1: Isolated Testing with Node-RED	26
3.4	NFR 2.2: Improved logging	27
3.5	NFR 3: 24/7 Runtime Performance	27

3.6	NFR 4: Profile Picture	27
3.7	NFR 5: Encrypt User Password before Storing in the Database	27
3.8	NFR 6: Improve Speech Recognition Quality	28
3.9	NFR 7: Dahoam-Flaws	28
4	Quantity Structure	29
5	System Architecture and Interfaces	30
6	Acceptance Criteria	31
6.1	Delete User	31
6.2	First Configuration of the Mirror	31
6.3	Install Smart Home Interfaces	31
6.4	User Administration	32
6.5	Sleep Mode	32
6.6	Wake up from Sleep Mode	32
6.7	Creating First Admin	32
6.8	Defining Personal Location	32
6.9	Registration Flow	32
6.10	Alternative to QR-Code	33
6.11	Admin Account	33
6.12	Demo Mode	33
6.13	Kiosk Auto Start	33
6.14	Session Management	33

1 Initial Situation and Goal

1.1 Initial Situation

An initial prototype of Dahoam was built between 2018 and 2019. It combines a commercial mirror with a smart home hub. The mirror is capable of:

- recognizing when a person stands in front of it. If the person is unknown, they can register themselves through Dahoam-Connect-App.
- understanding when the user is speaking to him (only with a predefined word pool).
- showing weather data of a static location.
- adding an email address via POP3 to a user account.
- adding a calendar to a user account and showing upcoming events on the mirror.
- speaking to the user after commands with a predefined word pool.
- integrating static MQTT smart home interfaces (temperature sensor and light)

The reasons for carrying out the project was to enhance the mirror in all aspects and create a smart mirror which is practically useful. It is in interest of the headmaster and head of department of HTL Leonding to have a project like Dahoam which represents most of the technical focuses of the school.

1.1.1 Application Domain

Dahoam operates with 5 microservices. Face recognition spots a person in front of the mirror and compares it to all known user. When the person is already registered, they will be logged into the mirror. If the person is unknown to the mirror, they can register them. While registration and other processes, interaction is done through voice commands. The mirror understands spoken words via speech recognition, which sends the understood words to the third microservice, the intent parser. It tries to find out what the intention of the user is and sends the intent to the master service. The master service stands in between all services and handles the whole system. For confirmation after startup of the mirror or successfully done tasks the mirror gives feedback to the user through speaking to them via Text-To-Speech service.

Each user can operate their own account where they can also add a specific email address via POP3 and a calendar. The user is able to log in on the mirror with

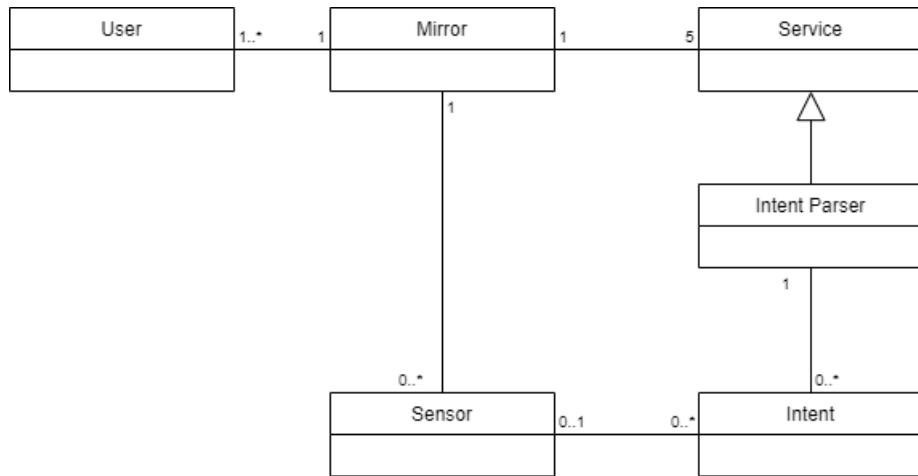
just standing in front of the camera or into the Dahoam-Connect-App with their nickname and password. Managing their account is only possible in the app.

Smart home interfaces are added statically. Sensor values can be requested from the user via speech interaction directly with the mirror. Same goes for turning on/off the lights.

1.1.2 Glossary

Text-To-Speech	Converts output to spoken words
Speech Recognition	Converts spoken input to text
Intent Parser	Parses input to executable commands
Face Recognition	Compares faces in front of the camera with saved faces
Master Service	Interface of all other services
Dahoam-Connect-App	Smartphone app for user/mirror configurations

1.1.3 Model of the Application Domain



1.2 Goal

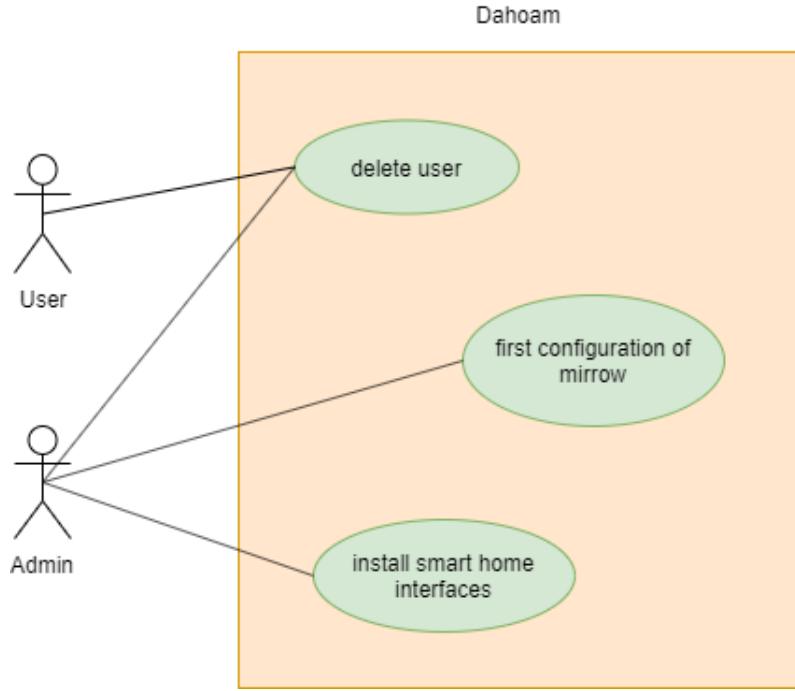
Dahoam should give the customer a solid smart home hub integrated in a common mirror. The intelligent mirror will run 24/7 and is only active when a person is in front of it. If there is one, a movement sensor will wake up all services. By switching into a sleep mode, the power consumption is reduced.

When starting the mirror the first time, the user has to connect to the hot spot from Dahoam. When connecting to the network, a website will open where the user has to enter an admin account and enter the mirror location for weather information. An advanced Dahoam-Connect-App has following additional features:

- Integration of new smart home gadgets
- Configuration of an individual weather location for each user
- Adding an email account for each user via IMAP
- Deleting users
- Admin role page where general mirror configuration can be done

2 Functional Requirements

2.1 Overview



2.2 Use Case 1: Delete User

2.2.1 General Description

In the user settings exists a “delete” button which deletes all related user data.

ID:	Delete User/Admin
Goal:	The user wants to be able to delete their own account.
Precondition:	The use case will be triggered by pressing the “delete” button in the Dahoam-Connect-App.
Postcondition:	The user is deleted and will not be recognized by the mirror anymore. Also the user will not be able to log in with the deleted account in the app.
Involved Users:	Role name: user, admin

2.2.2 UI to Call the Use Case

This is the design of the user settings, where the delete button will be added. With this button it will be possible to delete the account logged in.

Benutzer-Einstellungen

Benutzername
Mustermaxi

Vorname
Max

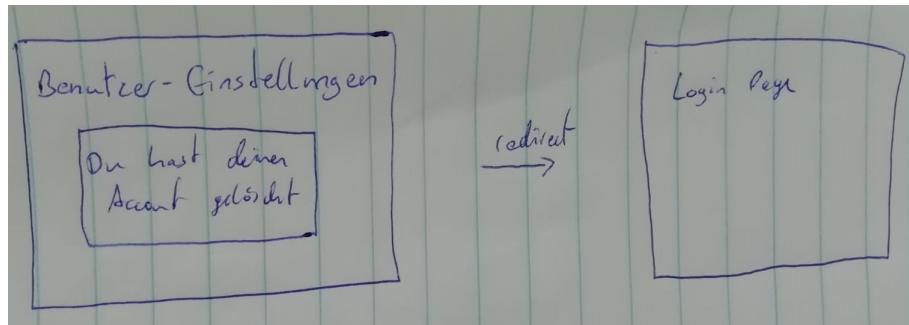
Nachname
Mustermann

ÄNDERUNGEN SPEICHERN

ACCOUNT LÖSCHEN

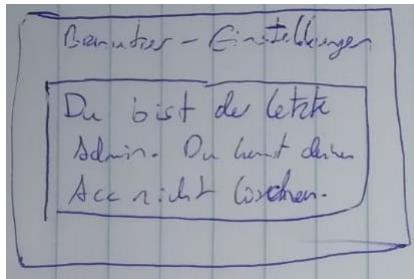
2.2.3 The Standard Use

The user clicks the button and their account will be deleted. The app will automatically log out the account and the mirror does not recognize the user again.



2.2.4 The Non-Standard Use

If the last admin of the mirror wants to delete their account, an error message must occur, because without an admin, the mirror can not change the smart home interfaces.



2.3 Use Case 2: First Configuration of the Mirror

2.3.1 General Description

The user has to connect to the hot spot which the mirror provides. Then a website will be opened on the gadget from which the user connected to the hot spot. There the mirror can be connected to the actual network and define its location.

After this, the problem of having a non-static IP still exists. Therefore it is recommended to give the mirror a static IP. This also makes it a lot easier for the Dahoam-Connect-App to handle.

ID:	First configuration of mirror
Goal:	After the mirror has been installed, it must be configured, so that the mirror knows in which network it works and what its location is.
Precondition:	The mirror has to be built together and started for the first time.
Postcondition:	The mirror knows in which network it has to operate and where it is located.
Involved Users:	Role name: admin

2.3.2 UI to Call the Use Case

Verbindungseinstellungen

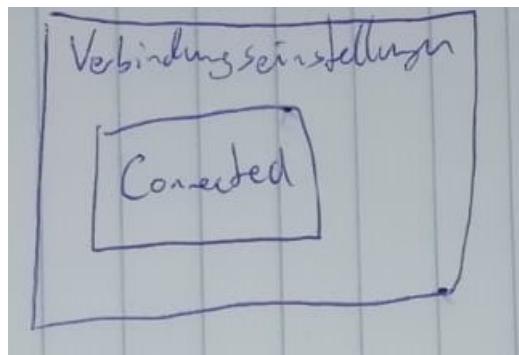
Wlan

Passwort

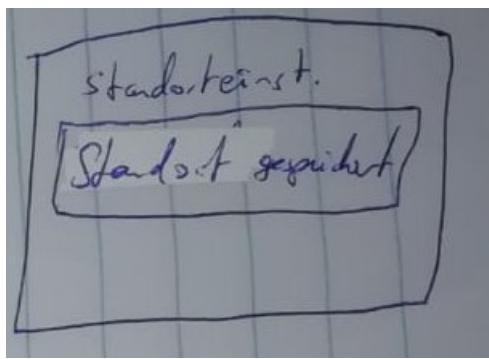
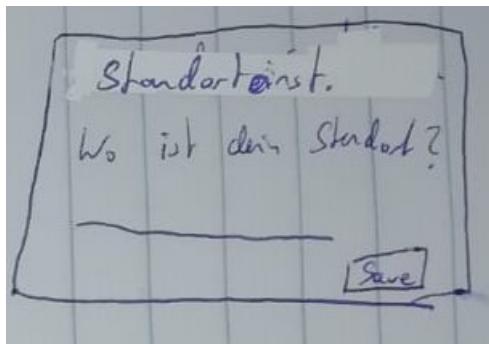
VERBINDEL

2.3.3 The Standard Use

A WiFi is found and the correct password was submitted.

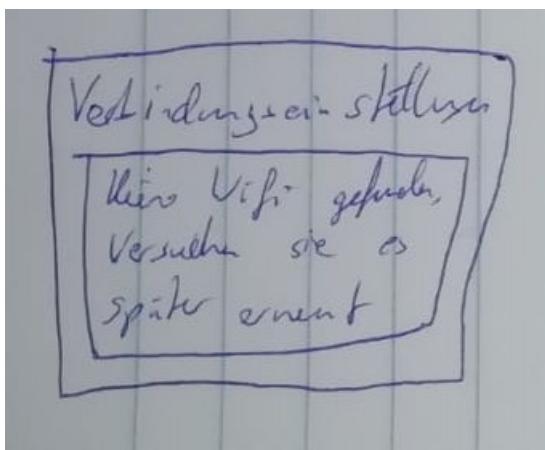


Then the location of the mirror has to be entered. By entering a letter, a drop down list will give options for existing locations.



2.3.4 The Non-Standard Use

If there is no WiFi to connect to, the mirror cannot fulfill its main purpose - being a smart home hub. Therefore an error message will be shown to the user. Also he will not be able to continue the process of configuring the mirror until it is connected to a WiFi.



If a WiFi got selected, but the user entered a wrong password, an alert will appear to communicate this to the user.



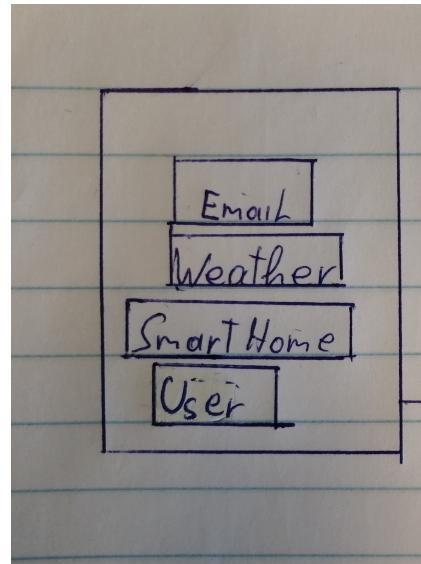
2.4 Use Case 3: Install Smart Home Interfaces

2.4.1 General Description

Connecting Smart Home Devices to the mirror with the help of MQTT.

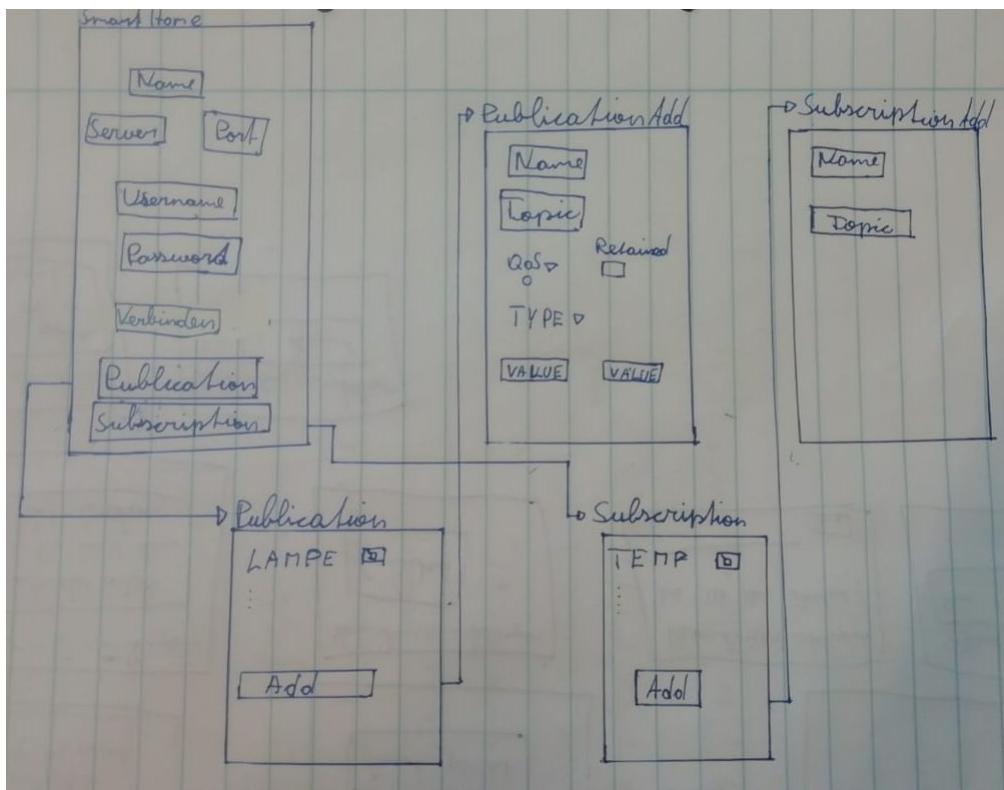
ID:	Install smart home interfaces
Goal:	Connecting the mirror with smart home devices.
Precondition:	Admin presses the button for installing the interfaces.
Postcondition:	The mirror knows all different smart home devices which are installed.
Involved Users:	Role name: admin

2.4.2 UI to Call the Use Case



2.4.3 The Standard Use

First, a MQTT broker has to be set in the Dahoam-Connect-App. After pressing the connect button a small alert will be displayed. When connected, devices can be added. First a list of all existing devices will be shown with an add button at the bottom. When a publication is added, it is needed to specify a name as well as the topic. QoS and retained are optional. To finish the procedure it is necessary to select, which type of values will be sent. Subscription is more or less the same, but there has to be set the name and topic, since data will be received.



2.4.4 The Non-Standard Use

When the user does not enter the right credentials for the broker, an alert occurs and warns him.

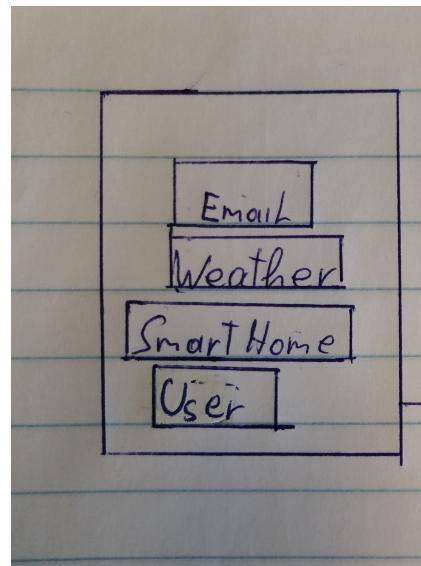
2.5 Use Case 4: User Administration

2.5.1 General Description

Admin accounts are able to administrate users. By clicking the user button, a list of all users in the system appears.

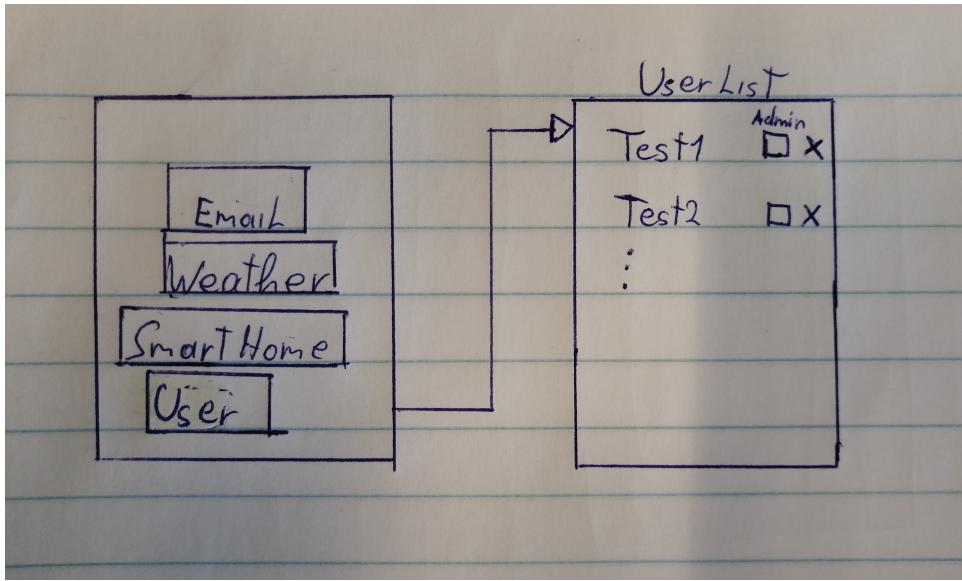
ID:	User administration
Goal:	The admin should be able to see a list of all users and be able to delete users that are not marked as admin. It is also possible to give another user admin rights.
Precondition:	The admin clicks the user button to get all users.
Postcondition:	A User has been deleted or has gotten admin rights.
Involved Users:	Role name: admin

2.5.2 UI to Call the Use Case



2.5.3 The Standard Use

- Users can be deleted by a admin as long as it is not an admin account.
- Admins can give other users admin rights.



2.5.4 The Non-Standard Use

- Admins can not delete other admins. If they try, an error will occur.

2.6 Use Case 5: Sleep Mode

2.6.1 General Description

When the mirror does not detect any movement for a certain amount of time it enters the sleep mode to save energy.

ID:	Sleep mode
Goal:	The mirror does use a lot less energy and does not heat up to fast.
Precondition:	The mirror is running.
Postcondition:	The mirror is in sleep mode.
Involved Users:	Role name: user

2.6.2 The Standard Use

The mirror is in the sleep mode whenever no one has used it recently and waits for a user to move in front of it. This is important to do something against the overheating

problem. For sleeping mode unused services will be stopped.

2.6.3 The Non-Standard Use

The mirror does not enter the sleep mode correctly or never enters it.

2.7 Use Case 6: Wake up from Sleep Mode

2.7.1 General Description

ID:	Wake up from sleep mode
Goal:	When user stands in front of the mirror, it “wakes up”.
Precondition:	Mirror is in sleep mode and user stands in front of the mirror.
Postcondition:	Mirror is running.
Involved Users:	Role name: user

2.7.2 The Standard Use

The mirror recognizes movement via a motion sensor and wakes up from sleep mode. After this, the user is able to use the mirror. All stopped services start again.

2.7.3 The Non-Standard Use

- The motion sensor does not detect any movement, despite a user standing in front of it.
- One of the stopped containers got unhealthy and cannot be started anymore.

2.8 Use Case 7: Creating First Admin

2.8.1 General Description

The first user that registers himself, will automatically be an admin.

ID:	First admin account creation
Goal:	After set up an admin account exists.
Precondition:	No user is registered.
Postcondition:	One user exists which is an admin.
Involved Users:	Role name: admin

2.9 Use Case 8: Defining Personal Location

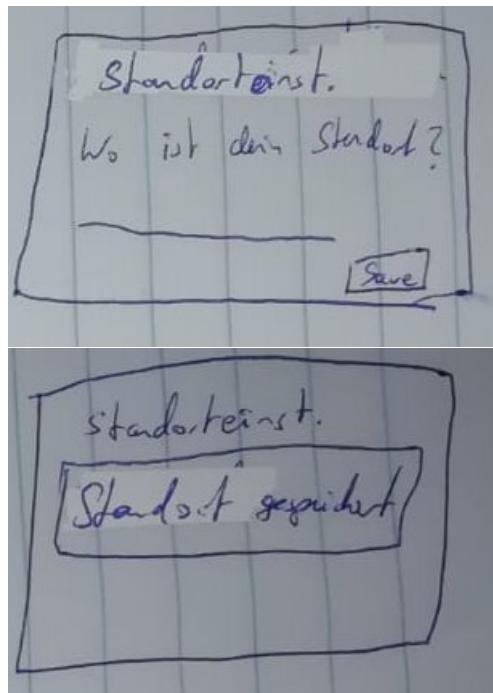
2.9.1 General Description

When a user registers, the mirror sets their location. This is used for the weather forecast of that particular user.

ID:	Defining personal location
Goal:	The weather for the selected location of the user is displayed instead of the default mirror location.
Precondition:	The mirror displays the default location.
Postcondition:	Personal location is set instead of the default location.
Involved Users:	Role name: user

2.9.2 The Standard Use

The user changes the default location to their personal location in the user settings. The weather forecast for the personal location will be displayed.



2.9.3 The Non-Standard Use

The wished location could not be found or does not have a weather forecast which can be displayed at the moment.

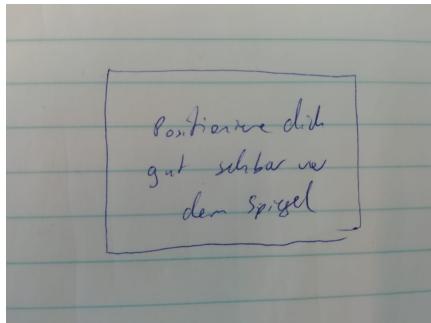
2.10 Use Case 9: Registration Flow

2.10.1 General Description

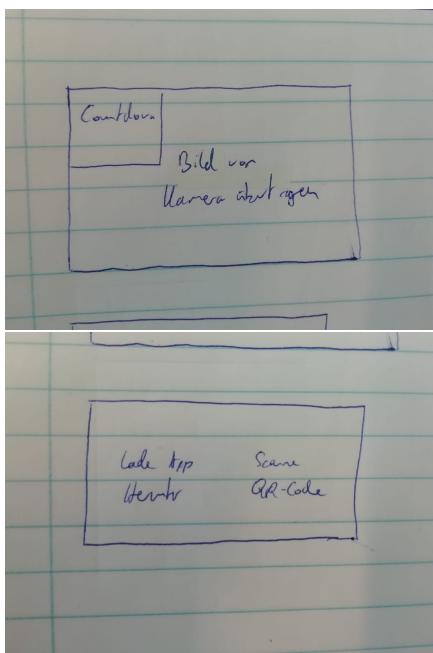
ID:	Registration flow
Goal:	It is easier for the user to register.
Precondition:	Unregistered user wants to register and stands in front of the mirror.
Postcondition:	Unregistered user is registered and gets recognized by the mirror with the face recognition.
Involved Users:	Role name: user

2.10.2 The Standard Use

The user says the keyword “register”, so that the mirror knows that the user wants to register. Then a window appears, which tells the user to position correctly in front of the mirror.



After the message, a live broadcast of the mirror's camera appears and the user has 5 seconds to prepare for taking a picture.



After this step, the user has to connect the Dahoam-Connect-App to the mirror by either scanning the qr-code with the App or using the alternative. Then the user has to enter their credentials.

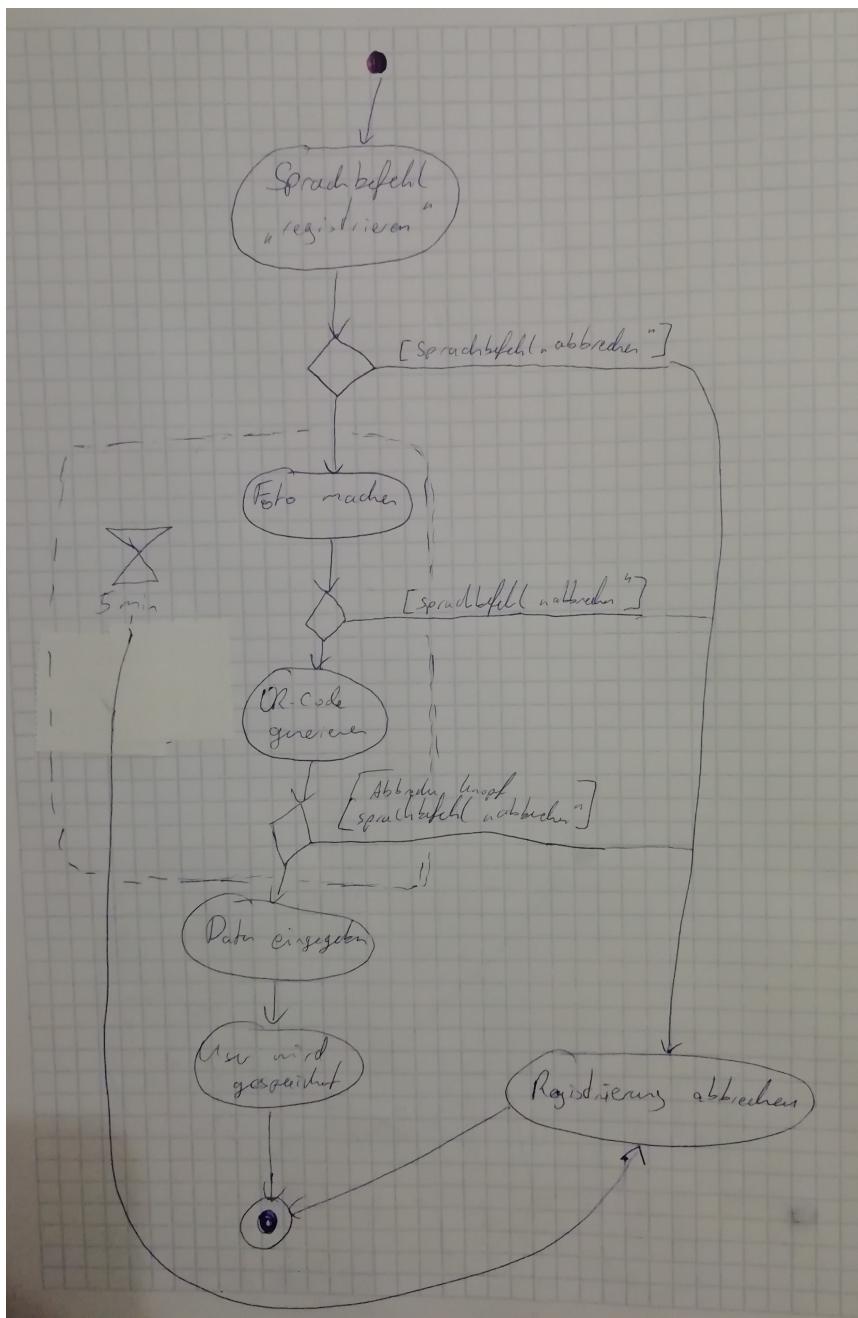
2.10.3 Registration aborted by timeout

If the user does not finish the registration process within 5 minutes, the registration will be aborted and the taken picture and information will be deleted.

2.10.4 Registration aborted by user

The user is able to abort the registration by saying “abbrechen” or pressing the button “abbrechen” on the Dahoam-Connect App. After aborting the taken picture and information will be deleted.

2.10.5 Activity Diagram



2.11 Use Case 10: Alternative to QR-Code

2.11.1 General Description

ID:	Alternative to QR-code
Goal:	Be able to register via the app without scanning a qr-code.
Precondition:	Registration has started.
Postcondition:	The user has registered via the qr-code alternative.
Involved Users:	Role name: user

2.11.2 The Standard Use

During the registration process, the app has to get the information, which IP the IP of the mirror owns. It either can get acquired by scanning the QR-code or an alternative, which is still in search.

2.11.3 The Non-Standard Use

The Dahoam-Connect App is not connected to the same WiFi as the mirror.

2.12 Use Case 11: Admin Account

2.12.1 General Description

An admin is a user with more permissions than a normal user for example:

- set smart home devices
- manage other users

ID:	Admin account
Goal:	To have a user with more permissions to manage users and smart home devices.
Precondition:	No one is able to manage the users or install new smart home devices.
Postcondition:	A user that is able to manage everything exists.
Involved Users:	Role name: admin

2.13 Use Case 12: Demo Mode

2.13.1 General Description

The demo mode is used on fairs to display the functionality of the mirror.

ID:	Demo mode
Goal:	To have a different mode than the normal use of the mirror for showing features under harder conditions (loud noise, many people).
Postcondition:	The functionality and all features are getting correctly showcased.
Involved Users:	Role name: user

2.13.2 The Standard Use

Start the demo mode on a fair so that the features can be presented to the audience.

2.13.3 The Non-Standard Use

The features are not working correctly (because of other conditions than for normal use, like loud noise and many people)

2.14 Use Case 13: Kiosk Auto Start

2.14.1 General Description

The raspberry pi 4 just has to be plugged in and the mirror boots automatically.

ID:	Kiosk auto start
Goal:	Plug it in and everything is ready to go.
Precondition:	The mirror has to be installed in a room and has power.
Postcondition:	The Raspberry Pi 4 has booted and started Dahoam automatically.
Involved Users:	Role name: user

2.14.2 The Standard Use

The raspberry pi boots and starts the mirror application without anything else to be done. Without kiosk the user would have to write the command to start the mirror themselves, open the browser on localhost and connect on the right port. But to be

the able to do this, it would be necessary to connect a mouse, keyboard and screen to the raspberry pi 4.

2.14.3 The Non-Standard Use

When the raspberry pi can not connect to the last used network or was never connected to any, it starts the network configuration in order to let the user connect it to an available network.

2.15 Use Case 14: Session Management

2.15.1 General Description

User should be logged in and out as fast as possible. Also it should not be possible to use all commands by a person that is not registered.

ID:	Session management
Goal:	Improve the session management for login and register and restrict some commands to logged in users.
Precondition:	Everyone can use all commands and it takes quit a long time to be logged in and out.
Postcondition:	Only authorised users can use the mirror to its full extent.
Involved Users:	Role name: user

2.15.2 The Standard Use

When a user is registered, all commands can be used.

2.15.3 The Non-Standard Use

In demo mode, commands are still available, even if no one is logged in.

3 Non-Functional Requirements

3.1 NFR 1: Enable Full Dockerized Runtime Environment

ID:	Full dockerized runtime environment
Name:	Enable full dockerized runtime environment
Type :	USE
Description:	It is really complicated and takes a lot of time to install all required software and programming languages on the Raspberry Pi 4. Therefore, a Docker runtime environment should fix this.

3.2 NFR 2: Development Environment

ID:	Development environment
Name:	Development environment
Type :	MAINT
Description:	Instead of always compiling or running the whole project, a development environment gets offered for developer, so different parts of the project can be build and tested independent of each other.

3.3 NFR 2.1: Isolated Testing with Node-RED

ID:	Isolated testing with node-RED
Name:	Isolated testing with node-RED
Type :	USE
Description:	Developers will be able to test different sections of the mirror like the speech or masterservice separately from each other. There are different test cases prepared on node-RED so that testing is easier. This will improve the time it takes to test single components and identify the exact location of bugs and errors.

3.4 NFR 2.2: Improved logging

ID:	Improved logging
Name:	Improved logging
Type :	USE
Description:	Each component will log data in a separated log file, which will make finding problems easier. The developer shall also be able to change the log level of each component, so that only the important information for their specific problem are logged.

3.5 NFR 3: 24/7 Runtime Performance

ID:	24/7 runtime performance
Name:	24/7 runtime performance
Type :	EFFIC
Description:	Programs that run for a long time have to cope with memory leaks and docker container that get unhealthy. Therefore, a technique has to be developed that is able to cope with those kind of problems.

3.6 NFR 4: Profile Picture

ID:	Profile picture
Name:	Profile picture
Type :	USE
Description:	The picture that was taken while registration will be shown to the user. The user is also able retake this picture.

3.7 NFR 5: Encrypt User Password before Storing in the Database

ID:	Encrypt user password before storing in the database
Name:	Encrypt user password before storing in the database
Type :	SEC
Description:	For a higher security the password has to be encrypted or at least stored as a hash code.

3.8 NFR 6: Improve Speech Recognition Quality

ID:	Improve speech recognition quality
Name:	Improve speech recognition quality
Type :	EFFIC
Description:	The speech recognition sometimes does not understand the spoken words correctly. If there is a lot of background noise, this problem gets worse and worse. In order to achieve best perform this problem has to be fixed.

3.9 NFR 7: Dahoam-Flaws

ID:	Dahoam-Flaws
Name:	Dahoam-Flaws
Type :	USE
Description:	<p>When speaking to the mirror it often understands wrong things. E.g.: “Schalte das Licht in Raum 2 und 3 ein” then it adds the two numbers and turns on the light in room 5. Furthermore a problem occurs when several registered users are in front of the mirror because it can not decide which of them should be logged in. When an unregistered user kicks off registration process and denies it, the whole registration process is blocked for the running session.</p> <p>While registration, a live camera feedback should help the user positioning himself in the right place. This prevents bad profile pictures.</p>

4 Quantity Structure

Because of high data processing, the Raspberry Pi 4 gets very hot. That's caused by incoming voice data and camera feedback.

5 System Architecture and Interfaces

Dahoam is embedded either in a common home without any smart home interfaces or in a modern home with some smart home lights or sensors installed.

The user is able to communicate with the mirror either with speech recognition or Dahoam-Connect-App.

The mirror has a connection to a weather API and has an interface to get the emails of the user via IMAP. It is also capable of addressing the calendar which is saved on the phone of the user.

6 Acceptance Criteria

6.1 Delete User

If a user is deleted, all related data must be removed. If a user removes their account through Dahoam-Connect-App, a confirmation message appears and the user will be redirected to the login view. If they are meanwhile logged into the mirror, they should be logged out after the deletion process.

First Configuration of the Mirror

The configuration must include:

- Create an admin account for all further configurations
- Connect to an existing network (on-/offline does not matter) in order to communicate with Dahoam-Connect-App
 - Online: Enter location for the weather display

6.2 First Configuration of the Mirror

The mirror has to be able to connect to a WiFi via a website on their smart phone.

6.3 Install Smart Home Interfaces

For configuration of smart home devices, an own page in Dahoam-Connect-App is required. Following information is required:

- MQTT-Broker IP and port
- A username and password if needed
- Publish
 - Name
 - Topic
 - QoS and retained
 - Values
- Subscription

- Name
- Topic
- QoS

After entering the required information the data from the smart home device should be received correctly.

6.4 User Administration

The admin has to be able to delete other accounts and giving them admin rights. An admin is also capable of seeing all other accounts.

6.5 Sleep Mode

The mirror is able to go into sleep mode and save power by doing so.

6.6 Wake up from Sleep Mode

When the motion sensor detects movement, the mirror starts the face recognition and checks if a registered user stands in front of it.

6.7 Creating First Admin

While first registration process, the first user will automatically be set as an admin.

6.8 Defining Personal Location

The user is capable of defining an own location to be shown at the mirror screen instead of seeing the mirrors default location.

6.9 Registration Flow

The user starts the registration flow with a spoken command and can also abort it by a timeout or by a speech command or pressing the “abbrechen” button in the Dahoam-Connect App.

6.10 Alternative to QR-Code

An alternative to QR-code exists to connect the Dahoam-Connect App with the mirror.

6.11 Admin Account

The admin should be able to delete other users, install smart home devices and configure the default mirror weather location

6.12 Demo Mode

At a fair the mirror can be presented without any problems and all features can be showcased to the audience.

6.13 Kiosk Auto Start

When plugging in the mirror the setup and starting of the application is done automatically.

6.14 Session Management

The mirror logs out the user automatically, when the interaction between user and mirror stops.