



This Project has received funding from the European Union’s Horizon 2020 Research and Innovation Programme under Grant Agreement N. 818329



Sun coupled innovative Heat pumps D5.2 – Thermal Comfort driven Smart Home End-User Interface

Due date of deliverable: **30/09/2020**

Actual submission date: **30/09/2020**

Organisation name of lead contractor for this deliverable: CW

List of contributors: SE, IES, CARTIF, IVL

Dissemination Level (Specify with “X” the appropriate level)		
PU	Public	X
CO	Confidential	

Project Contractual Details

Project Title	Sun coupled innovative Heat pumps
Project Acronym	SunHorizon
Grant Agreement No.	818329
Project Start Date	01-10-2018
Project End Date	30-09-2022
Duration	48 months
Supplementary notes:	This document will be publicly available (on CORDIS and project official websites as soon as approved by EC

Table of Contents

1	Introduction	5
1.1	EXECUTIVE SUMMARY	5
2	Step 1: App Requirements	7
2.1	INPUT - DATA, AND INFORMATION GOING INTO THE APP	7
2.2	OUTPUT - DATA, AND INFORMATION COMING OUT OF THE APP	7
2.3	FEEDBACK APP	8
2.4	USER REQUIREMENTS	9
2.5	USAGE OF THE FEEDBACK APP ON TWO DIFFERENT BUILDING TYPES	9
3	Step 2: Mock-Ups	10
3.1	FIRST WIREFRAMES	10
3.2	UPDATED WIREFRAMES	10
4	Step 2: Frameworks And Demo App	13
4.1	FRAMEWORK SELECTION	13
4.2	FIRST DEMO APP	14
5	Step 3: Final Implementation	16
5.1	BACKEND	16
5.2	FRONTEND (GUI)	19
5.2.1	Register And Login	19
5.2.2	Overview	21
5.2.3	Feedback	23
5.2.4	Energy	25
5.2.5	The Learning Scale	25
5.2.6	More	27
5.2.7	Policy Privacy	27
5.2.8	Settings	28
6	Conclusion	29
A.	Annexes	30
A.1	PRIVACY POLICY FOR FEEDBACK APP	30

List of figures

FIGURE 1. FIRST WIREFRAME	10
FIGURE 2. UPDATED WIREFRAME MENU	11
FIGURE 3. UPDATED WIREFRAME, LOGIN AND REGISTER	11
FIGURE 4. UPDATED WIREFRAME, FEEDBACK, COMFORT, ENERGY CONSUMPTION, LEADERBOARD AND MORE.	12
FIGURE 5. FIRST DEMO APP.....	ERROR! BOOKMARK NOT DEFINED.
FIGURE 6. APP, DB, API AND ISCAN FLOW	ERROR! BOOKMARK NOT DEFINED.
FIGURE 7. FINAL IMPLEMENTATION OVERVIEW	16
FIGURE 8. MONGODB COLLECTIONS.....	16
FIGURE 9. SCREENSHOT PF BUILDING COLLECTION FROM MONGODB ATLAS	17
FIGURE 10. SCREENSHOT PF USER COLLECTION FROM MONGODB ATLAS	17
FIGURE 11. SCREENSHOT PF FEEDBACK COLLECTION FROM MONGODB ATLAS.....	17
FIGURE 12. MODELS AND FUNCTIONS	18
FIGURE 13. HEROKU SCREENSHOT	18
FIGURE 14. APP: REGISTER OR LOGIN.....	19
FIGURE 15. APP: REGISTER FORM.....	20
FIGURE 16. APP: LOGIN	20
FIGURE 17. APP: REGISTER ERRORS	20
FIGURE 18. APP: LOGIN ERRORS.....	20
FIGURE 19. APP: OVERVIEW COMPONENT DETAIL VIEW	21
FIGURE 20. APP: OVERVIEW AIR TEMPERATURE	22
FIGURE 21. APP: OVERVIEW HUMIDITY	22
FIGURE 22. APP: OVERVIEW CO2 AND LIGHT LEVEL	22
FIGURE 23. APP: FEEDBACK SECTION, ROOMS LIST	23
FIGURE 24. APP: FEEDBACK SECTION, FEEDBACK COMPONENTS FOR A SPECIFIC ROOM	23
FIGURE 25. APP: FEEDBACK TEMPERATURE AND HUMIDITY.....	24
FIGURE 26. APP: FEEDBACK LIGHT LEVELS AND ACTIVITY LEVELS.....	24
FIGURE 27. APP: FEEDBACK CLOTHING AND NUMBER OF PEOPLE	24
FIGURE 28. APP: FEEDBACK MESSAGE AFTER FEEDBACK	24
FIGURE 29. APP: ENERGY HOURLY AND DAILY	25
FIGURE 30. APP: ENERGY MONTHLY AND YEARLY	25
FIGURE 31. APP: STATUS PART 1	26
FIGURE 32. APP: STATUS PART 2	26
FIGURE 33. APP: MORE SECTION.....	27
FIGURE 34. APP: POLICY PRIVACY SECTION	27
FIGURE 35. APP: SETTINGS CONNECT ACCOUNT TO BUILDING.....	28
FIGURE 36. APP: SETTINGS DISCONNECT FROM BUILDING	28
FIGURE 37. APP: SETTINGS SESSIONS	28

List of tables

TABLE 1. INPUT REQUIREMENTS	7
TABLE 2. OUTPUT REQUIREMENTS	8
TABLE 3. APP REQUIREMENTS	8
TABLE 4. USER REQUIREMENTS	9
TABLE 5. COMPARISON BETWEEN NATIVE AND HYBRID APPS.....	13
TABLE 6. SELECTED FRAMEWORK FOR FEEDBACK APP FRONTEND	13
TABLE 7. FRAMEWORKS AND SERVICES FOR BACKEND.....	14
TABLE 8. APP: STATUS, LEARNING SCALE RANGE.....	26

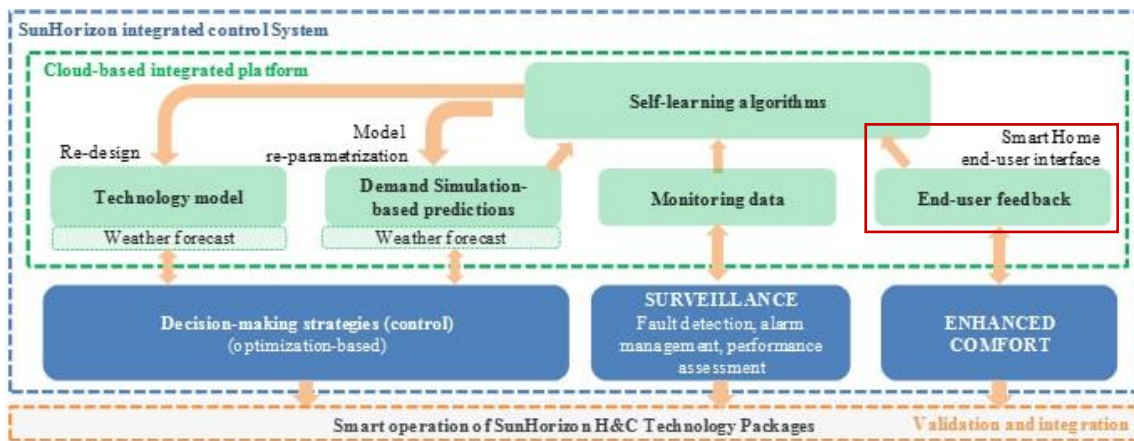
1 Introduction

1.1 Executive summary

The aim of SunHorizon control activities is to create a system able to control and monitor heating and cooling systems using prediction, self-learning, and end-user feedback. The activities described in this deliverable focuses on the latter aspect, creating an end-user interface also referred to as the SunHorizon Feedback app that allows the user to provide the system with feedback about their perception of the indoor comfort levels (e.g. temperature, air quality, light). The feedback is then used by the self-learning algorithms to predict thermal comfort of the end-users.

In particular, the SunHorizon Feedback App will be integrated in the full SunHorizon control system presented here below with the following purposes:

- Provide feedback to the self-learning algorithms to proper predict the thermal comfort of the end-users
- Interact with the control systems that, based on the feedback from the end-users (i.e. they feel warm/cold) modified the control strategies and set point of SunHorizon technologies to enhance the comfort of the occupants
- Provide available data, from the installed SunHorizon sensors, to the occupants about their energy consumption



The SunHorizon Feedback App has two main objectives:

1. To present the actual parameter of comfort and energy consumption to the end-user to make the user aware of their energy consumption.
2. To collect their feedback, to be transferred to the self-learning algorithms

The work has been an iterative process with three steps, each one detailed described in the following Chapters:

- **Step 1** – The first iteration focused on gathering requirements (both user and technical) determining the input needed from the sensors (sensor values) and output expected from the users (user feedback) as well as a brainstorming session to gather ideas for a user-friendly app. (**Section 2**)
- **Step 2** – In the second iteration, the requirements where updated, and a first wireframe was created based on the ideas from the brainstorming. The application requirements were also prioritized as part of the implementation plan as well as the selection of the platform on which the app would be implemented. A first demo app to test the integration with the iSCAN system as well as to test the selected platform (IONIC) was also performed during this iteration. (**Section 3 and 4**)
- **Step 3** – During the third iteration the implementation took place. With lessons learned from the first demo, the implementation of the final app could start. The final implementation consists of two parts: the backend and the frontend. The backend provides APIs for the app for sending/receiving information to/from the database as well



as to communicate with the iSCAN APIs. The frontend is the graphical user interface (GUI), thus what the end-user sees and interacts with. (**Section 5**)

2 Step 1: App Requirements

2.1 Input - Data, and information going into the app

One of the objectives of the work performed is to measure abiotic factors that affect the users' indoor comfort levels and visualize it in the app as an added value for the user.

This information comes by the sensors that will be installed in all the project buildings and which data will be collected in the SunHorizon monitoring platform. The following table lists the sensors desired for the app as well as their availability on the demo sites. The information about which sensors would be possible to install at the demo sites was provided by Schneider Electric.

Sensors	Availability
Air Temperature sensors	In all buildings
CO2 sensors	In all buildings
Humidity sensors	In all buildings
Motion sensors	Based on the building
Windows opening sensors	Based on the building
Lux sensors	Based on the building
Noise sensors	Based on the building
Radiative Temperature sensors	Based on the building

Table 1. Input requirements

Learning Scale

In order to increase the active participation of the end-users and help them in adopting smart behaviour towards reduction of the energy consumption, a learning scale (E.g. Your building has reached level 5/20 in learning) has been created that also would serve the purpose of giving the user an understanding of the correlation between *more feedback = more efficient system* and thus keep track of the progress. This would also open the possibility to compare or compete among users/households. A first version was implemented using a dummy learning scale calculator (see Status section in **Error! Reference source not found.**) but will be updated during the integration process.

2.2 Output - Data, and information coming out of the app

The output of the App is the data/information that can be sent from the app out to other parts of the SunHorizon system, in other words, feedback from the user. The requirements for the information are defined by the needs of the self-learning algorithm. The table below lists the types of feedback needed, both required and optional as well as the scales used in the app. Temperature and humidity use the ASHRAE standard for the scale, for the other types of feedback an arbitrary scale is suggested but can be adjusted during the coming integration with the whole SunHorizon control system.

Feedback	Required	Comment	Scale/Values
Temperature comfort	Yes		Using the ASHRAE thermal sensation scale ¹ : -3 (Cold), -2

¹ [ASHRAE Standard 55: Thermal Environmental Conditions for Human Occupancy](#)

			(Cool), -1 (Slightly cold), 0 (Comfortable, neutral), +1 (Slightly warm), +2 (Warm), +3 (Hot)
Humidity comfort	Yes		0%-30% (low), 30%-60% (Comfortable), 60%-100% (High) ²
Illuminance comfort	Optional		1 (Dark), 2 (Comfortable), 3 (Bright)
Activity level	Optional	Ex. cooking, watching TV, working/studying, sleeping.	1 (Light), 2 (Moderate), 3 (Vigorous)
Clothing level	Optional	Amount of clothes Ex. light, medium, high. Together with activity, it may be useful to see if there are relationships between activities and complaints.	1 (Nude), 2 (Light), 3 (Moderate), 4 (Heavy)
Number of people	Optional	This could also be calculated by the system ex. use the app to track the number of people in the building based on the number of connections. This info is important, especially for the simulation tasks.	1 (One person), 2 (Two persons) ... 10 (Ten persons)

Table 2. Output requirements

2.3 Feedback App

To proper reach the objectives of the App (visualize comfort levels and energy consumption, allow user to leave feedback about comfort levels) the following functions (requirements) have been identified and integrated in the App.

Function	Description
Dashboard	Visualize current measured indoor comfort levels (e.g. temperature in the kitchen, humidity in the living room)
Feedback	A user-friendly way for the user to send feedback about their current comfort level by room (e.g. Bedroom feels too hot, the office feels too dark)
Energy consumption	Visualize user's energy consumption (e.g. hourly, daily, monthly, yearly)
Notifications	Send push notifications to the app to remind the user to provide feedback
Status	A way to show the user the importance of providing feedback to get a more effective system using the learning scale and statistics.
Register/Edit account, Login/Logout	Increases security and makes it possible to keep track of who provides the feedback.
Connect to building	This would be the link between the app and the iSCAN system, where the app gets most data from.
Privacy Policy	Standard
About the project	Might be interesting for the user to read more about the project

Table 3. App requirements

² <https://www.centralhtg.com/blog/managing-home-humidity-for-maximum-comfort>

2.4 User requirements

The user's requirements and needs have been a main concern and important part during the work process. The definition process was centred around answering some key questions, such as *How do we create added value for the user within the app? What would make the user use the app more? What concerns might the user have about using the app?* The table below summarizes the main user requirements and their solutions.

Requirement	Description	Solution
Transparency	The user needs to feel comfortable with how their personal information is used and have the option of using the app anonymously.	To register only email and password are required, everything else (name, age, gender) is optional to fill in. A privacy policy is available in the app as well as a link to the project.
Usability	The app needs to feel interesting to view and intuitive to use.	Creating a user-friendly app has been one of the main goals during the work process of D5.2.
Value	The user needs to feel it is worth using the app	Dashboard, status and energy consumption visualization
Reminders	The user will need reminders to provide feedback	Push notifications
Progress	The user might want to know how well their system is performing	Status section (uses learning scale)

Table 4. User requirements

2.5 Usage of the Feedback app on two different building types

Since SunHorizon building typology is different from one demo to another, the App has been developed considering the constraints and needs of each of them. In particular, how the feedback App will be used will differ based if it is related to a residential (private) or tertiary (public) building. The difference will be mainly on how the App will be used and who will have access to it, and hence could provide feedback.

For the private building two different options have been considered: the first option is to create an account for each household, this means that any family member can give feedback and is somehow anonymous. The other option is to give each household member an account, this would be the better option result-wise but it all comes down to finding a privacy level that the user is comfortable with.

For the public buildings, it is most likely that the feedback will be anonymous and the users (visitors) need very easy access to the Feedback app. One scenario here could be that the building owner downloads the app to a tablet, registers, and places it on a frequently accessed location, where the visitors can leave their feedback on the go.

For both case, a bi-lateral discussion with the demo responsible is still on-going and it will be defined before the integration with the SunHorizon integrated system.

Moreover, to suit both types of buildings as well as different sensors installations the app has been implemented modularly: this means that it is easy to add or remove features (ex. If the building only has temperature sensors, the user will only see current temperature values in the Dashboard and only be prompted to leave temperature related feedback.)

3 Step 2: Mock-ups

3.1 First wireframes

The first wireframe that was created focused on the apps two main functions *Overview/Dashboard* and *Feedback* (see Figure 1). The Overview section would show interesting real-time values related to the building and use colours to show how “comfortable” each value is. For instance, if the indoor temperature is 15° C , it shows in blue colour.

The Feedback section (under preparation) should be kept short, with <5 questions and about 5 alternatives for each question. In the case that the questions are more than 5, they would be grouped. For instance, if the temperature feedback would be required for each room in the house, then a Temperature-category could be arranged. Another idea is to use colours, icons and shapes to make it easier and more intuitive for the user to send their feedback.

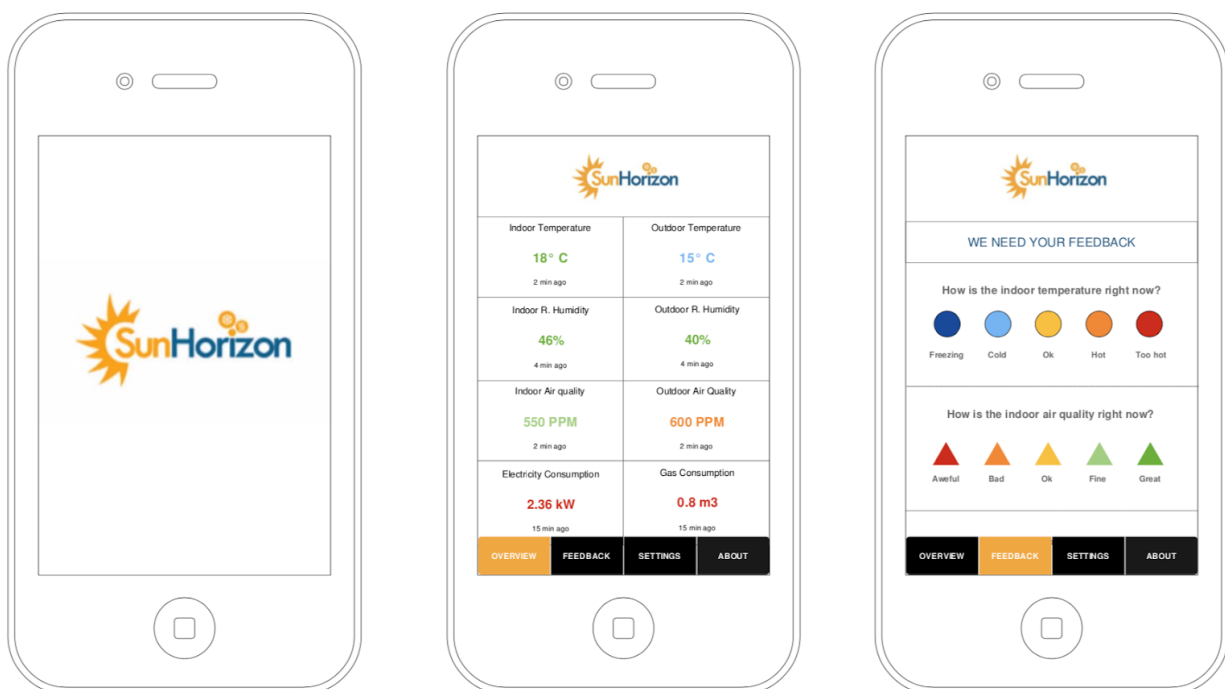


Figure 1. First wireframe

3.2 Updated wireframes

For the second iteration, the wireframes were updated and extended. This included a registration scenario for creating an account per household member. This wireframe also included sections for energy consumption, leaderboard, and more.

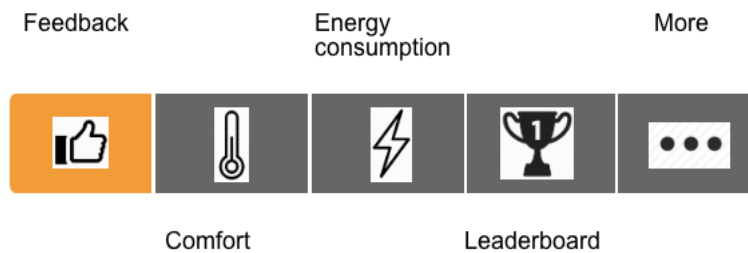


Figure 2. Updated wireframe menu

Register scenario (see Figure 3)

First, the user needs to create a “master” account by selecting the “Register new building” option. Here the user needs to fill in:

- Building name: this would be the name of the building or household that will later be used by other household members to connect to the same building
- Email and password: for confirmation and to log in
- Age and gender: optional

Next, the rest of the household members need to register by selecting the “Connect to building” option where they are required to enter the Building name (registered before) and then email, password, age, and gender. In this way it is possible to have multiple accounts connected to one building/household.

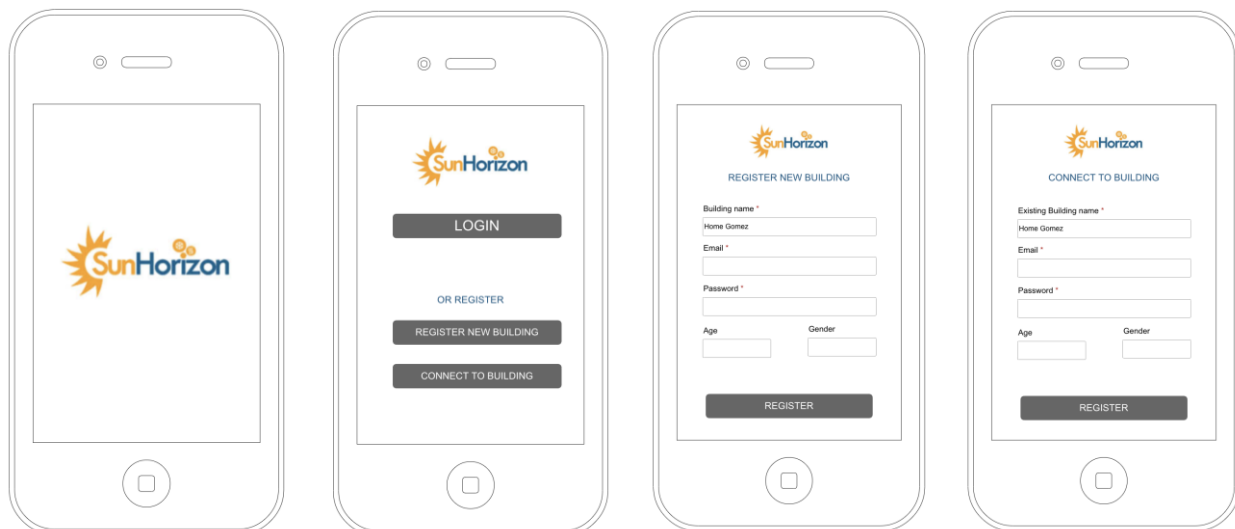


Figure 3. Updated wireframe, login and register

Feedback section

Since this would be the most important view in the app it was important to get it right. But there were still many open questions before the design process could continue.

- What input (in the form of feedback responses) does the learning algorithm require. This will determine the questions for the feedback
- How often do we require feedback from the user?
- Is it different depending on the question? Do we need temperature feedback more often?
- Should we disable the feedback controls for a time after feedback has been given to avoid spam?

Comfort section

This displays the measured comfort parameters or measurements gathered by the installed sensors. If possible, standards to display comfort levels should be used.

Energy consumption section

For this section there were some considerations about the granularity to display the energy consumption in (minutely/hourly/daily/monthly) that needed to be taken. This is also where we could show benchmarks (compare heating technologies) and simulations/estimations.

Leaderboard section

This is where the weekly competition (using the learning scale mentioned in section 2.1) could be displayed, the user can follow the progress of “teaching” their system and thus improving the systems' ability to regulate, for the user, a comfortable indoor environment. This would also be a benchmark between all installations.

More section

The last view could be used to give the user as much information as possible about how their data is managed and what the project is all about. As well as allow the user to configure their account and notifications.



Figure 4. Updated wireframe, feedback, comfort, energy consumption, leader board and more.

4 Step 2: Frameworks and demo app

4.1 Framework selection

Frontend

For the first demo, an important decision had to be taken, in terms of app development platform: whether to use a native programming language, developed for a specific platform ex. IOS, Android, or a hybrid one, developed for multiple platforms. For this purpose, advantages and disadvantages of both types have been considered, and reported in Table 5.

App type	Description	Advantage	Disadvantage
Native	Native apps are native to the target platform for which the app is created. Hence, they can only run on a single platform. So, a Native Android app can only run on the Android devices, and a native iOS app can only run on iOS devices.	The advantage of creating a native app is that the app is faster and most reliable when it comes to user experience. Another advantage is that native apps can interact with all of the device's operating system features, such as the microphone, camera, etc.	The costs in time and effort are higher since multiple apps need to be built (one for IOS and one for Android) and also maintained.
Hybrid	Hybrid apps can run on multiple platforms like Android, iOS, and Windows mobile, as they are built using platform-independent web technologies. Hybrid apps are enclosed inside a native shell which can connect with devices' sensors, just like a native app.	Hybrid apps are typically easier and faster to develop than native apps. They also require less maintenance. Another advantage is that you can build hybrid apps on a single base, allowing the developer to add new functionalities to multiple versions of the app.	Since the speed of a hybrid app depends on the speed of the user's browser, a hybrid app will rarely run as fast as a native app. Low speed can thereby affect the user experience.

Table 5. Comparison between native and hybrid apps

The final decision landed on developing a **hybrid app**, which works for both Android and IOS (the two most common operating systems) using the Ionic Framework³ with an Angular⁴-based codebase (see Table 6).

Type of app	Platform Support	Development Framework	Codebase
Hybrid	Android and IOS	Ionic (open-source mobile UI toolkit for building, cross-platform apps)	Angular Framework (Ionic supports different type of web application frameworks such as Angular, React)

Table 6. Selected framework for Feedback app frontend

³ <https://ionicframework.com/>

⁴ <https://angular.io/>

The decision was taken considering that: i) the Feedback app is a prototype and as such it is more important to focus on creating the best design and functionality possible, rather on the implementation itself, ii) the Feedback app is rather small with fairly simple functionality, iii) there is no need of using device features (camera, contact list, microphone). These characteristics suits the hybrid development better.

Backend

A first version of the backend was also developed for the first demo app, this was done using Node.js⁵ (asynchronous event-driven JavaScript runtime) with Express⁶ (web framework) and deployed to Heroku⁷ (Cloud application platform). The APIs are created using the RESTful API approach.

For the first demo app there was no login page developed, but it was decided that JSON Web Token⁸ (JWT) was to be used for authentication and authorization with a cloud-based MongoDB for storage.

Runtime	Web framework	API	Authentication & Authorization	Database	Deployment
Node.js	Express	RESTful API	JWT	MongoDB (provided by MongoDB Atlas ⁹)	Heroku

Table 7. Frameworks and services for backend

4.2 First demo app

A first demo app was created to test the integration with the iSCAN system. iSCAN is an application developed by IES, that will be the core component particularly for control, surveillance and KPI processing. The app was created with IONIC and retrieved the test data setup in iSCAN. This simple app showed building info, building measurements (list and charts), and tested a simple feedback feature (see **Error! Reference source not found.**). The demo app used the iSCAN REST API both to get sensor data (e.g. temperature, energy consumption) but also to send the feedback provided by the end-user. Due to some issues, the app was not able to communicate directly with iSCAN, this was solved by using the Feedback app API as a proxy API (see **Error! Reference source not found.**).

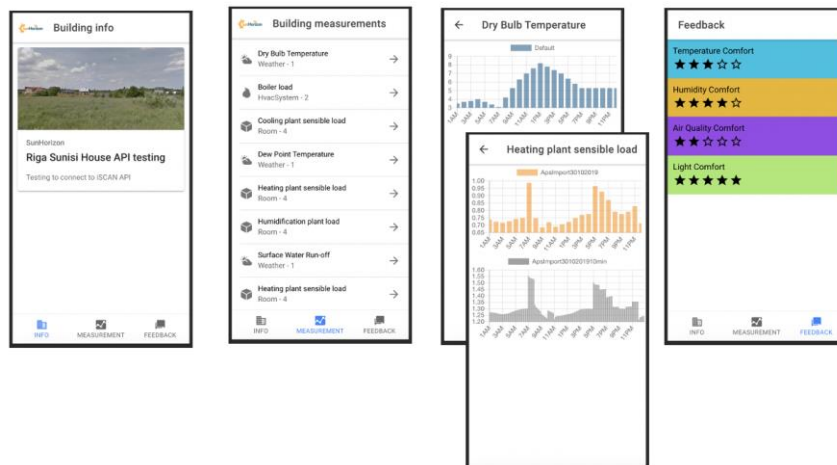


Figure 5. First demo app

⁵ <https://nodejs.org/>

⁶ <https://expressjs.com/>

⁷ <https://www.heroku.com/>

⁸ <https://jwt.io/>

⁹ <https://www.mongodb.com/cloud/atlas>

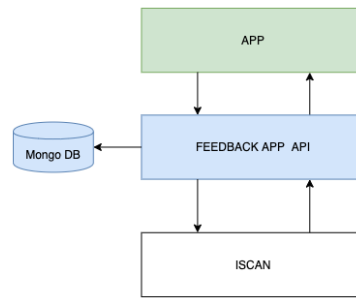


Figure 6. App, DB, API, and iSCAN flow

5 Step 3: Final implementation

Three main components were created during the final implementation, a database (MongoDB), an API for the Feedback app to communicate with the database as well as the iSCAN APIs and the App GUI.

The current state of the app (before integration in the overall control system) uses dummy data and placeholders in the GUI. Once integrated the data will be accessed through the iSCAN APIs.

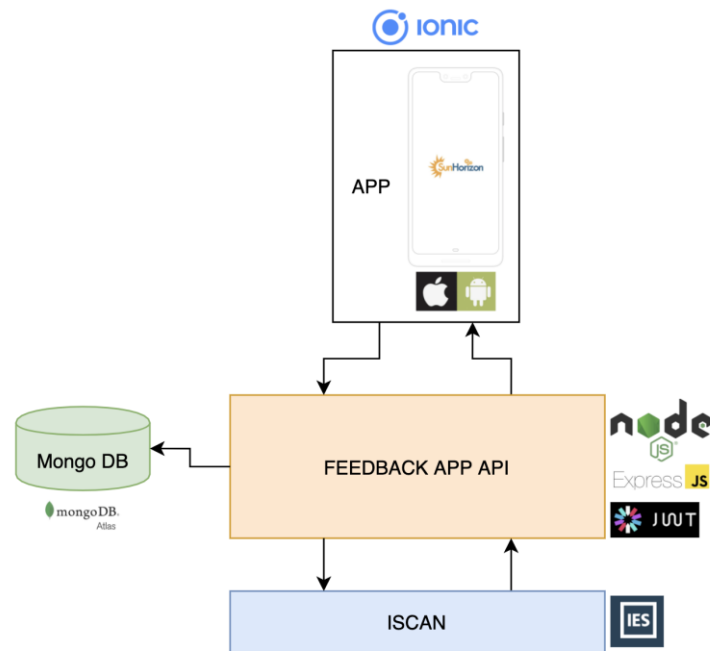


Figure 7. Final implementation overview

5.1 Backend

Database

The database, a MongoDB provided by MongoDB Atlas, was selected for its simplicity to use and availability in the cloud. It consists of three Collections (see Figure 8):

- **Building** where the endpoint to iSCAN is stored, as well as the code used to connect multiple accounts to one building (see Figure 9).
- **User** where the user account is stored (see Figure 10).
- **Feedback** where the users' feedback is stored and used to get statistics (see Figure 11).

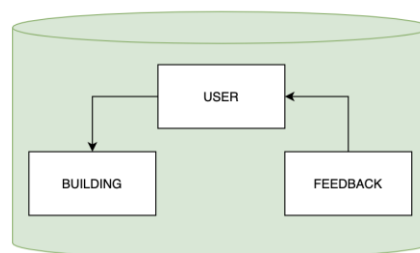


Figure 8. MongoDB Collections

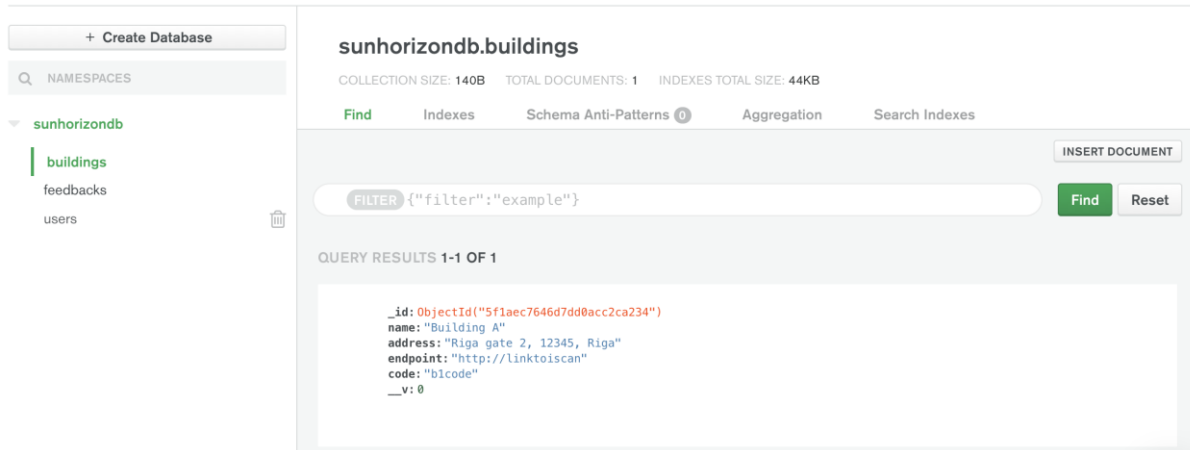


Figure 9. Screenshot of Building collection from MongoDB Atlas

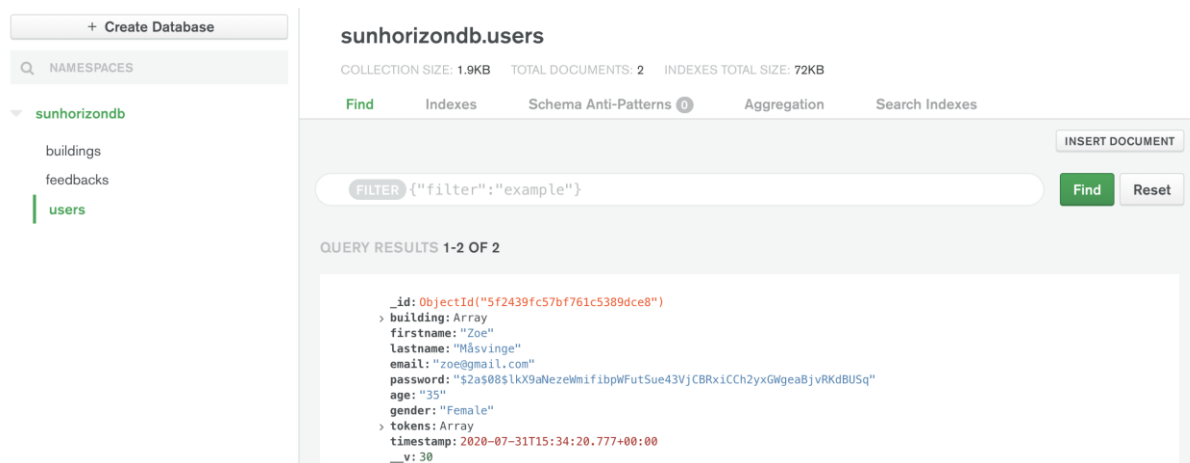


Figure 10. Screenshot of User collection from MongoDB Atlas

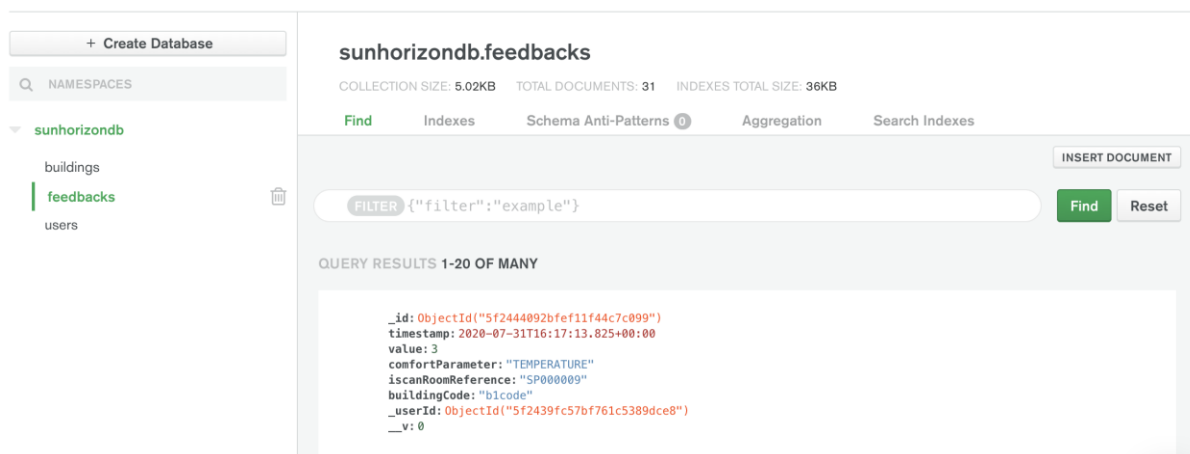


Figure 11. Screenshot of Feedback collection from MongoDB Atlas

RESTful API

The APIs can be divided into 4 main parts: *User*, *Feedback*, *Building*, and *iSCAN*. The first two (user and feedback) are complete and are not expected to change much during the coming integration process. The last two (building and iSCAN) on the other hand are not fully completed yet, these will be completed during the integration process in T5.6. *Building* will be the main bridge for the integration with the iSCAN system, this is where the iSCAN endpoint and credentials will be

stored to be used when calling the iSCAN APIs. The iSCAN part is currently used to retrieve sample data from the iSCAN API, this part will also be updated and extended during the integration process.

Figure 12 provides a more detailed view of the APIs, including both fields and methods.

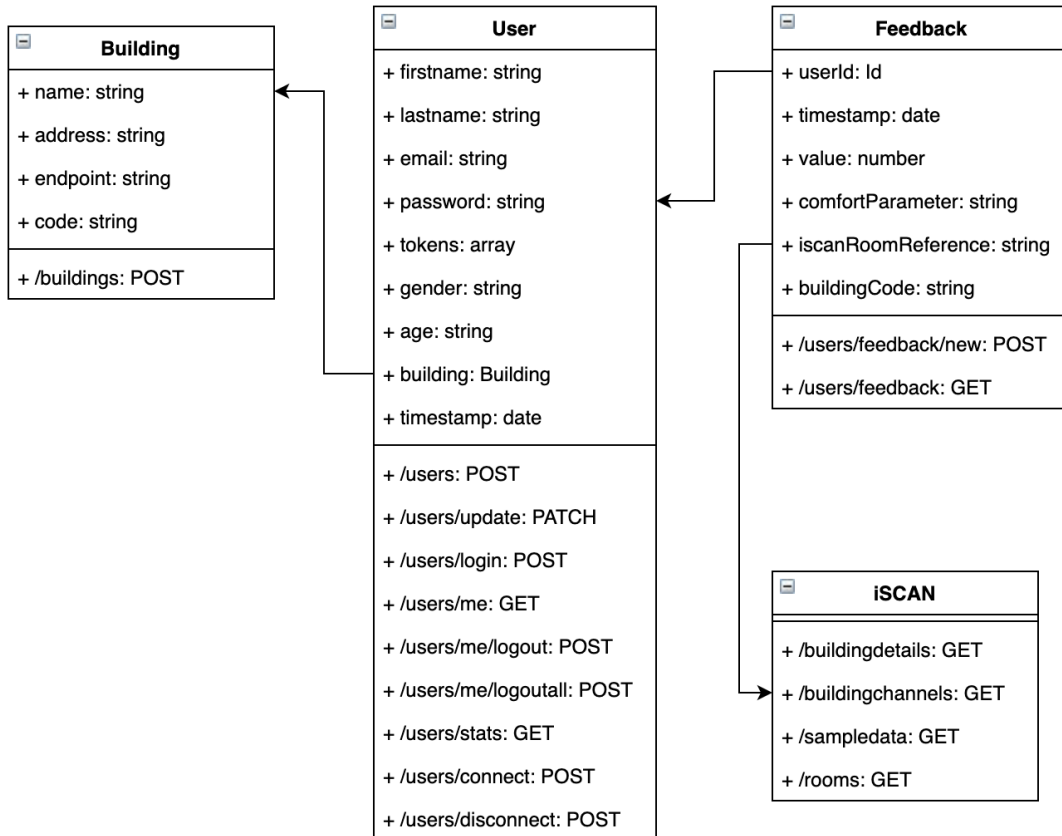


Figure 12. Models and functions

Deployment

The Node.js server including the APIs are deployed to Heroku, Figure 13 shows a screenshot of the current deployment.

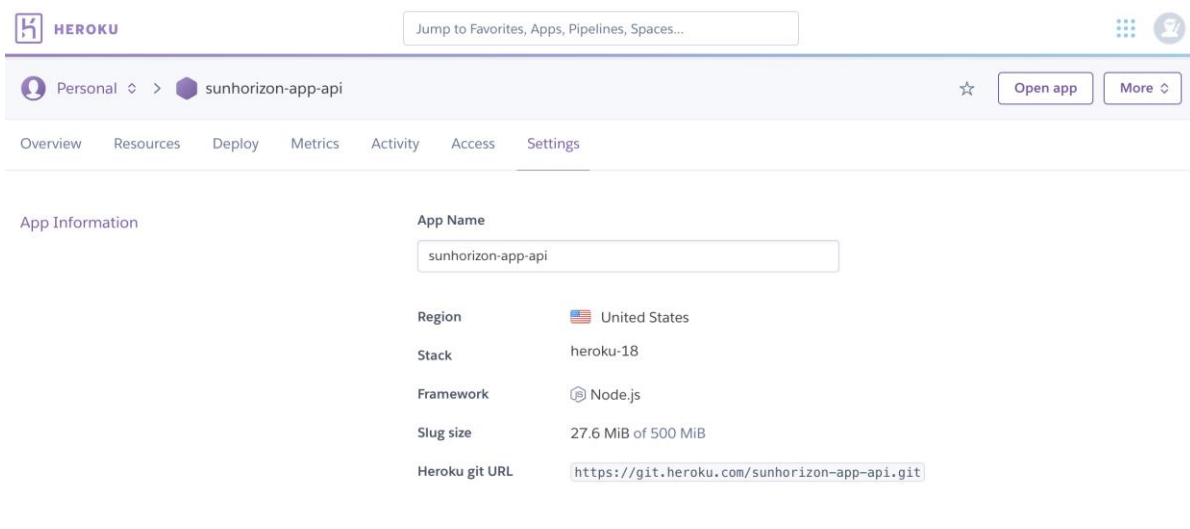


Figure 13. Heroku screenshot

5.2 Frontend (GUI)

The general look and feel of the Feedback App is clean and simple and uses the SunHorizon palette. There are some graphical differences between the Android and iOS versions (as can be seen in Figure 14), this is due to a feature by Ionic that provides platform-specific styles based on the device the application is running on to look and feel native to the user.

5.2.1 Register and Login

The first page in the app gives the user two choices, *Register* (for new users) or *Login* (if the user already has an account).

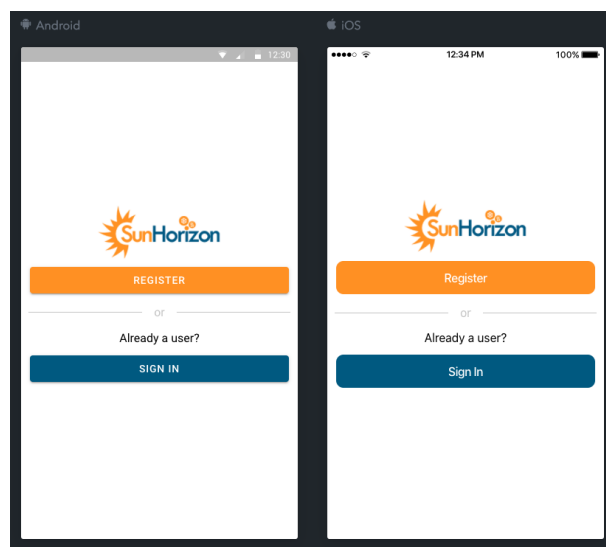


Figure 14. App: Register or Login

The registration process is fairly straight forward, and as previously mentioned, only two fields are required Email and Password while the rest of the information is optional. If the required fields are filled in, the button *Sign up* will be activated and by clicking it a user account is created (see Figure 15). The user is then signed in and redirected to the *Overview* page (see Figure 20). Error messages are displayed below the incorrect field, in case of errors ex. wrong email format, missing email, or password (see Figure 17).

There is however one more step that the user needs to do before the app is fully functional, and that is to connect their account to a building. This is done in the *Settings* page (see Figure 35). This is necessary to avoid that a user connect to a building which is not the one where it lives.

The login process is also straight forward (see Figure 16), once the user enters their email and password the *Sign-in* button is activated and the user is logged in and redirected to the *Overview* page. If the user enters the wrong credentials, an error will appear (see Figure 18).

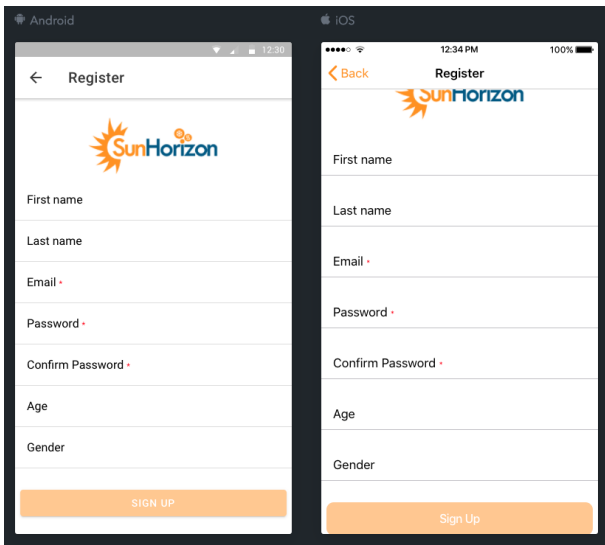


Figure 15. App: Register form

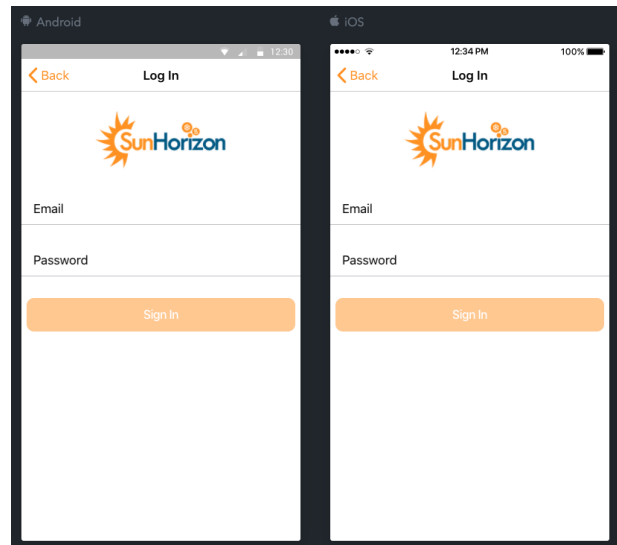


Figure 16. App: login

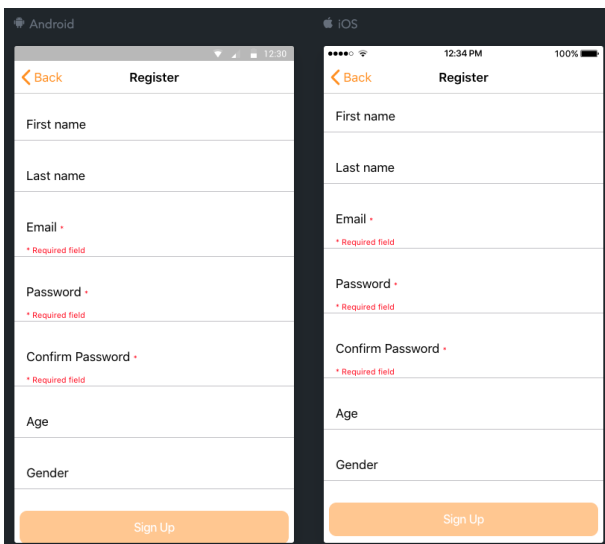


Figure 17. App: Register errors

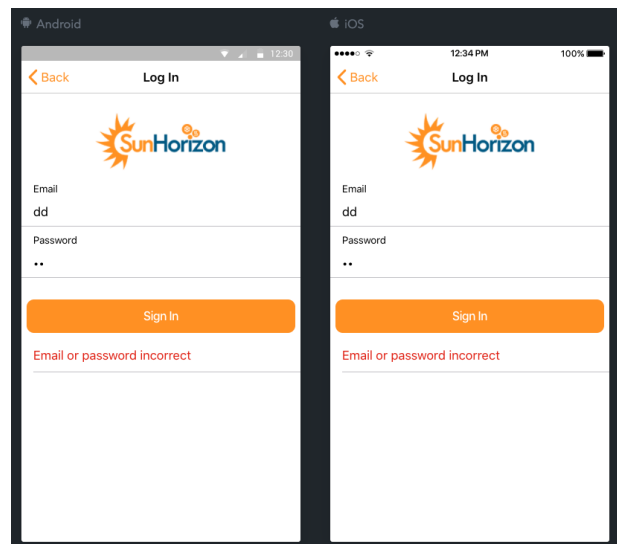


Figure 18. App: Login errors

5.2.2 Overview

The Overview page, also referred to as Dashboard during the design process, is the first page in the main menu and the starting point for the signed-in user. This is where the user can see the current values of the comfort parameters (ex. temperature, humidity) measured by the installed sensors. This section of the app is fully functional but uses dummy data for now. Once the integration with the iSCAN system is completed, these values will be retrieved through the iSCAN APIs.

There are four categories in the Overview page, Air temperature, Humidity, CO2, and light level (see Figure 20, Figure 21, and Figure 22). Each category shows the current measurement for each room and placement (indoor/outdoor). Figure 19 shows a detailed view of each Overview category component including icon, placement, room name, measurement, text, and timestamp. The text below the measurement value is intended as an aid for the user to understand if the value is within a comfortable/acceptable range. Ex. a temperature value of 16.2 °C feels *Cool* or a humidity value of 30.3% falls within the *Comfortable* range.

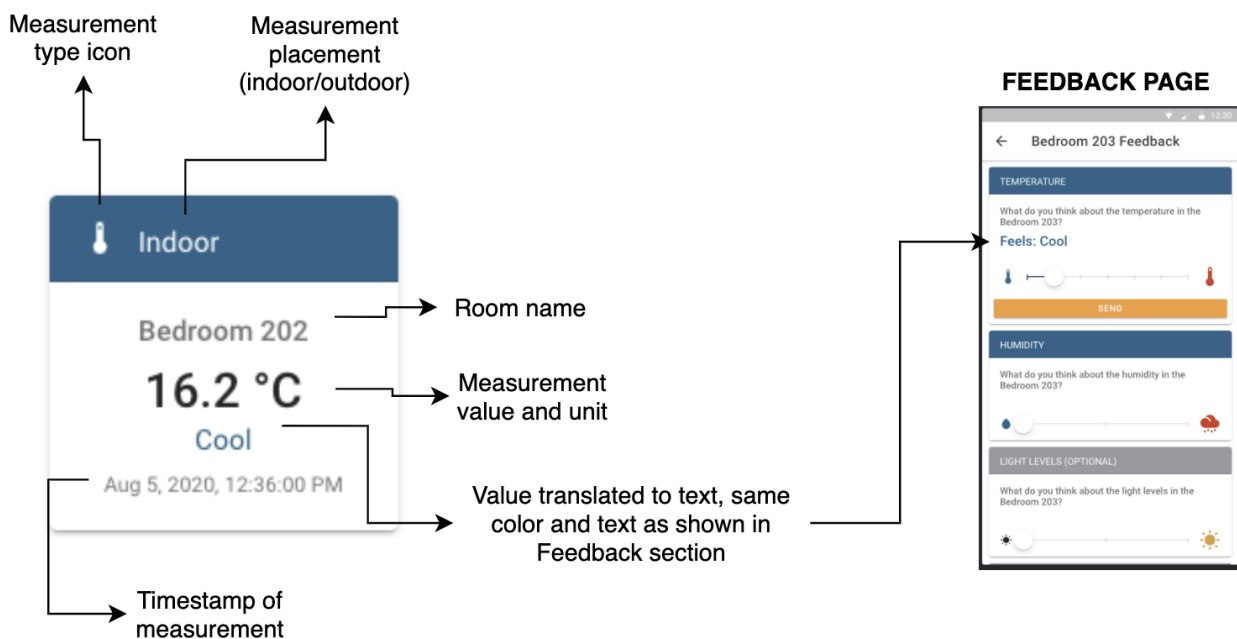


Figure 19. App: Overview component detail view

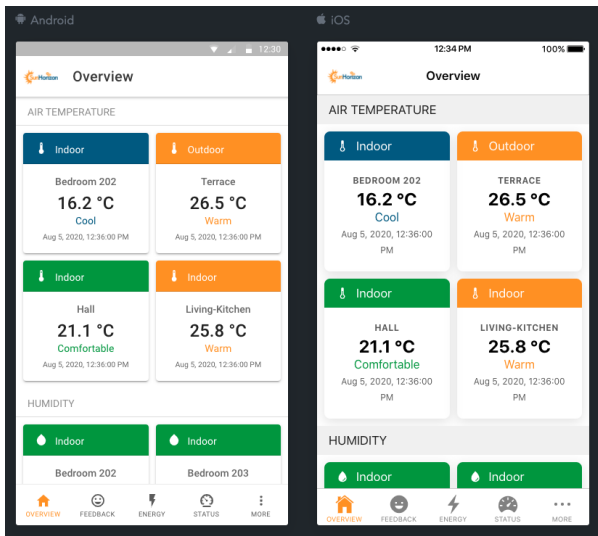


Figure 20. App: Overview Air temperature

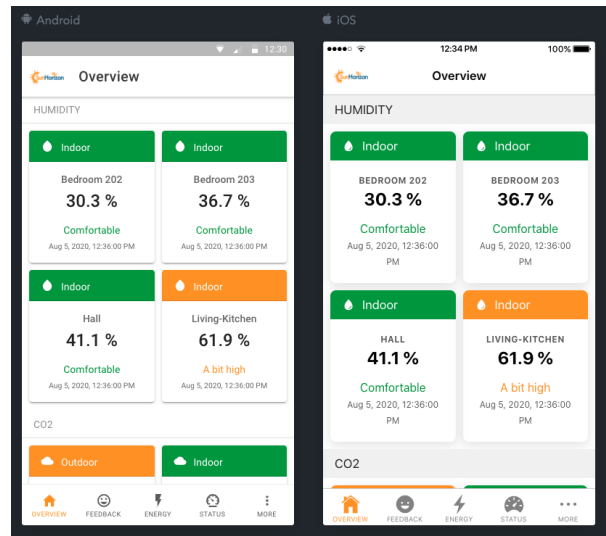


Figure 21. App: Overview humidity

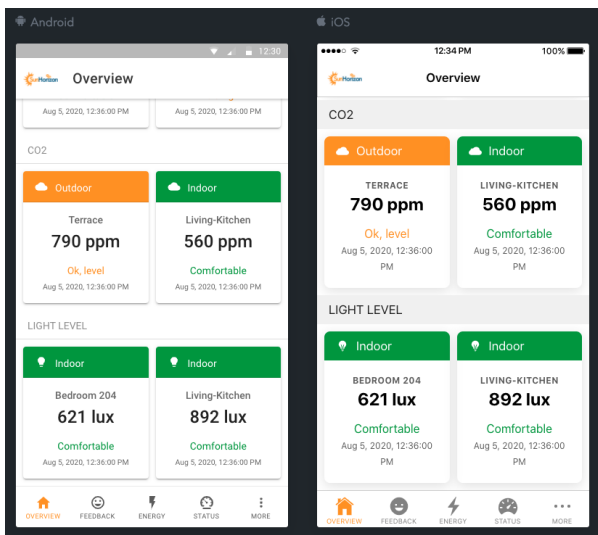


Figure 22. App: Overview CO2 and light level

5.2.3 Feedback

The *Feedback* section, a key aspect of the app as it provides input to the Learning algorithm that in turn is part of the SunHorizon smart control.

This section starts with a list of all available rooms and a counter (red badge to the right) showing the number of feedbacks that have been made for that room (independent of who made it) (see Figure 23). When the user selects a room from the list, they will come to the feedback controls section where the feedback is provided (see Figure 24).

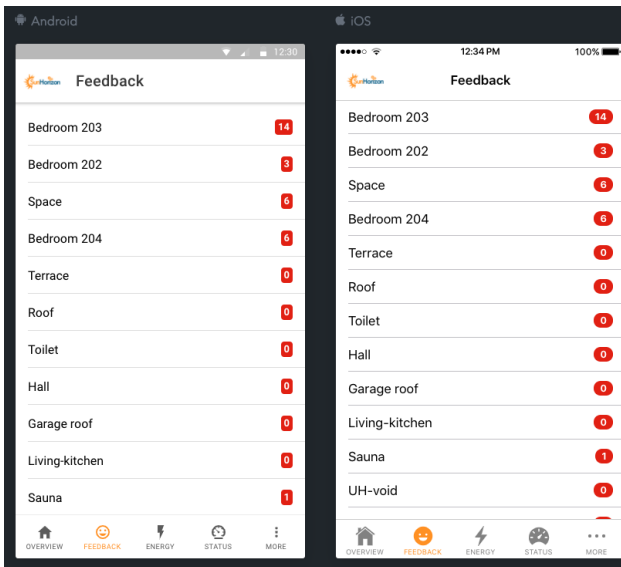


Figure 23. App: Feedback section, rooms list

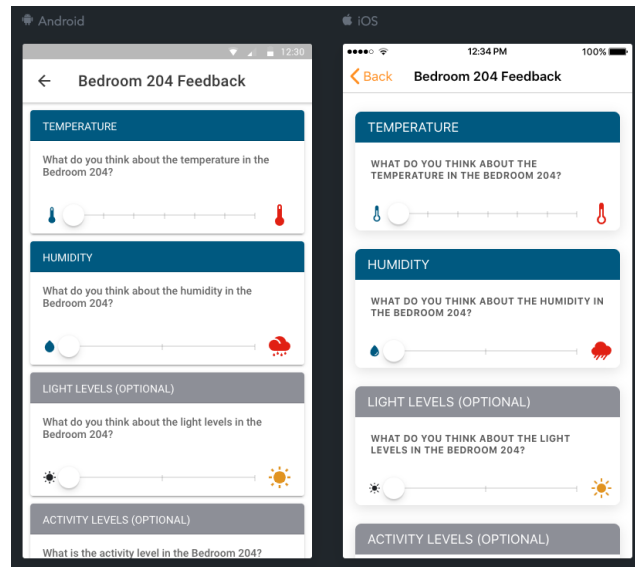


Figure 24. App: Feedback section, feedback components for a specific room

To provide feedback the user selects the ambient category (ex. temperature, light levels), and using the slider selects the most accurate option (see Figure 25, Figure 26 and Figure 27). The options for each ambient category can be found in Table 2 in the scale/values column.

Once the slider is activated, the send button will be visible, and the user can tap it to send their feedback. Finally, a message with the text “Thank you, your feedback is noted.” will appear as a Toast-message at the bottom of the page and the header of that ambient category will turn green with a check-mark icon to the right (see Figure 28). This is a feature to make it easy for the user to keep track on which categories they have provided feedback for.

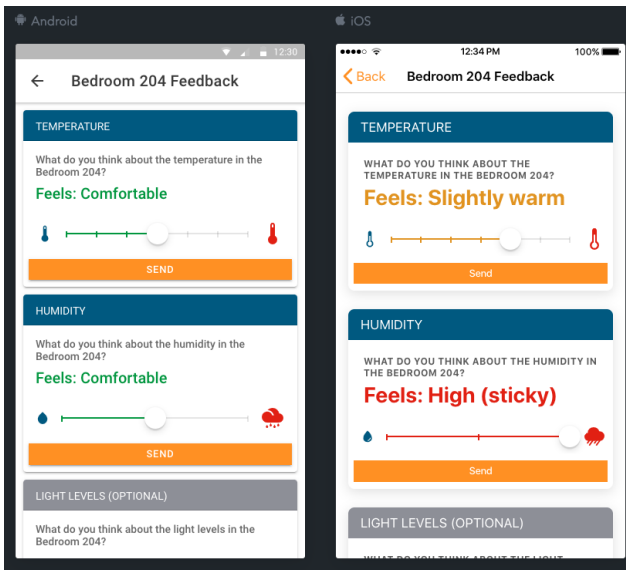


Figure 25. App: Feedback temperature and humidity

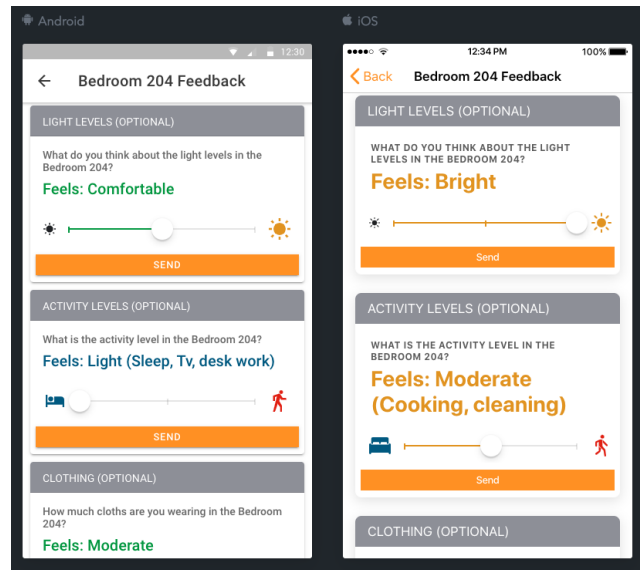


Figure 26. App: Feedback light levels and activity levels

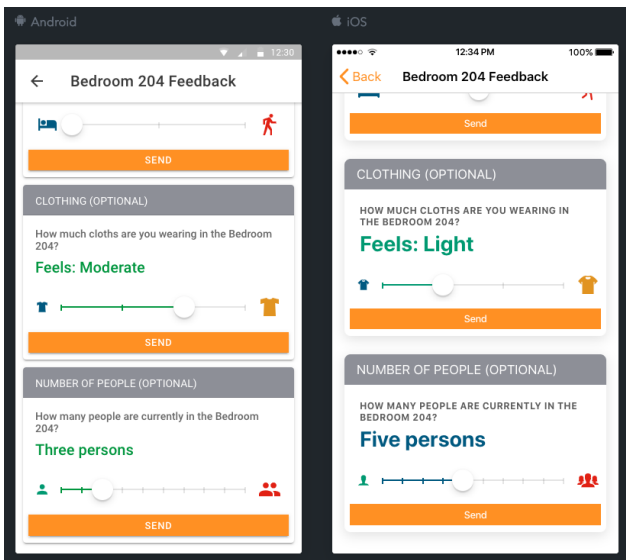


Figure 27. App: Feedback clothing and number of people

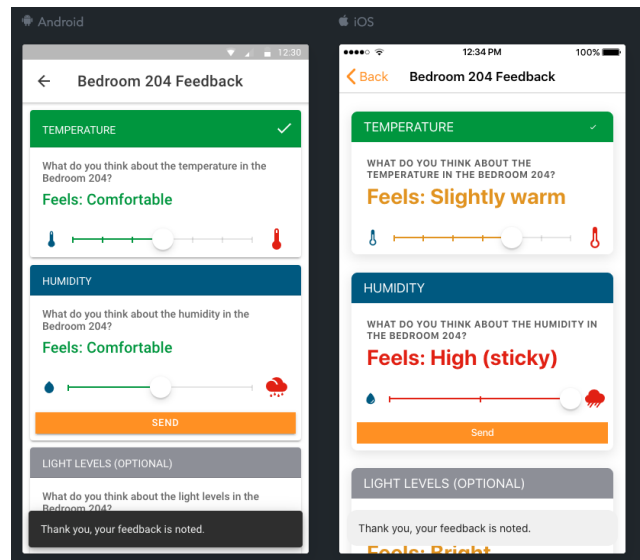


Figure 28. App: Feedback message after feedback

5.2.4 Energy

The *Energy* section is also implemented with dummy-data and will be fully implemented during the integration process. The current idea of this section is to show the user their actual energy consumption in bar charts with different granularities: hourly, daily, monthly, and yearly. Different colours have been used to enhance each bar chart and make it more readable. (See Figure 29 and Figure 30). Another way the colours can be used is to indicate that the energy consumption is, for instance, too high a particular day (red bar).

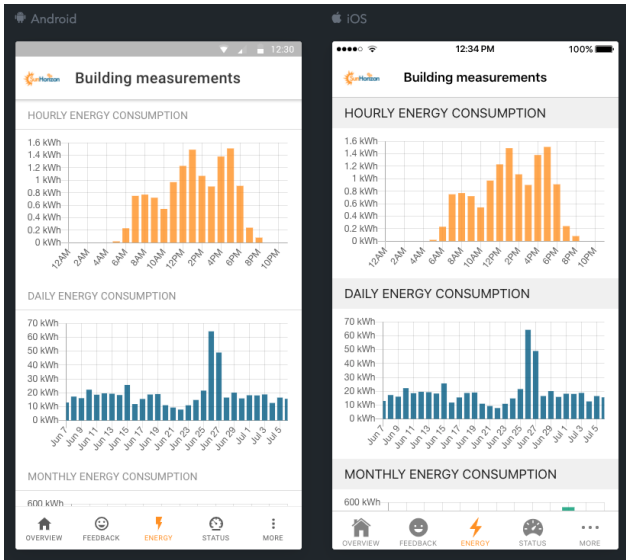


Figure 29. App: Energy hourly and daily

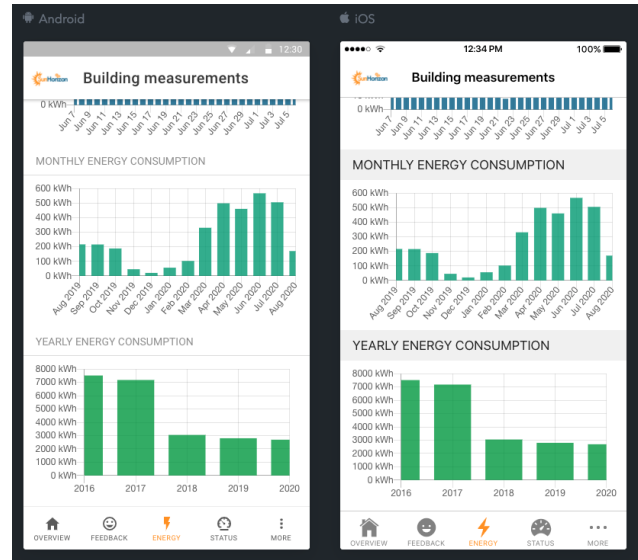


Figure 30. App: Energy monthly and yearly

5.2.5 The Learning Scale

The Status section (see Figure 31) is also fully functional but missing a real Learning Scale (see description in section 2.1) that will later be defined and created by the iSCAN system, the app is currently using a temporary learning scale calculator. The learning scale, or Learning Level, as it is called in the app consists of a progress bar with an accompanying badge showing a value between 0% (lowest level) and 100% (highest level) and a text (see Table 8).

Learning scale range	Display
< 20%	LEARNING LEVEL 5% Not so good, you need to provide more feedback
20% - 40%	LEARNING LEVEL 30% Ok, good job with the feedback so far.
40% - 60%	LEARNING LEVEL 50% Good, keep up your feedback intensity.

60% - 80%	LEARNING LEVEL 65% <hr style="width: 65%; margin: 5px auto;"/> <p>Very good! Thanks to all feedback you provided.</p>
> 80%	LEARNING LEVEL 90% <hr style="width: 90%; margin: 5px auto;"/> <p>Optimal! Your system has reached an optimal learning level and will perform very efficiently.</p>

Table 8. App: Status, learning scale range

There are also some statistics listed in the Status section, such as:

- Points: calculated using sent feedback, active days, and days since the start.
- Sent feedback: Number of feedbacks sent by the user
- Active days: days in a row that the user has provided feedback
- Days since start: number of days since the users' account was created

Finally, there is also an info section describing what the learning level is and why it matters with a button that redirects the user to the Feedback section when pressed (see Figure 32).

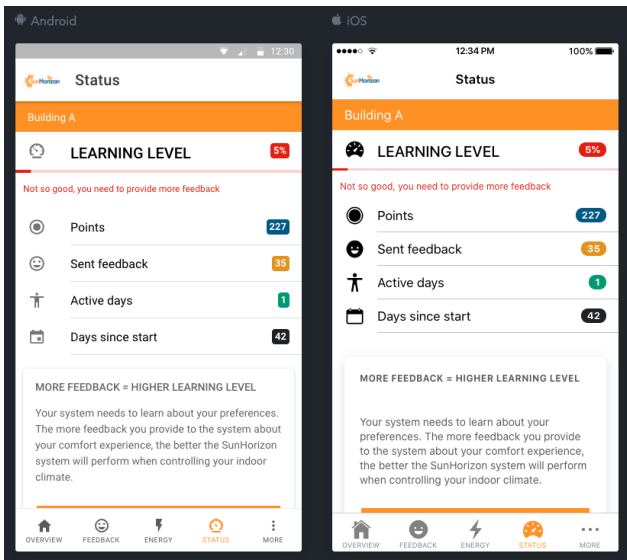


Figure 31. App: Status part 1

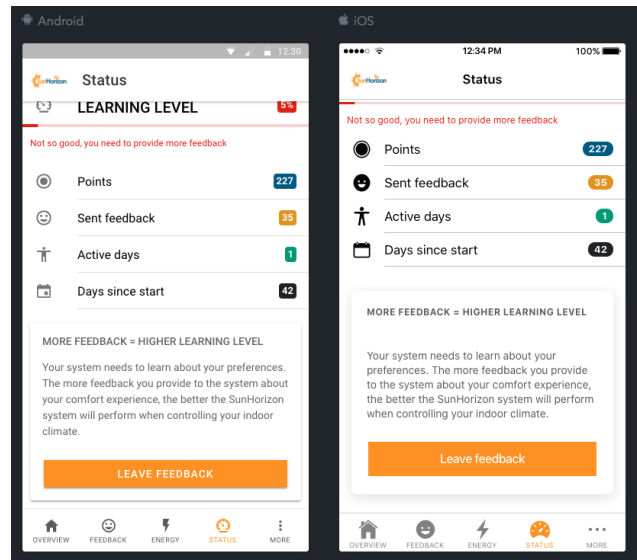


Figure 32. App: Status part 2

5.2.6 More

The fifth menu selection, called More, leads to a section containing: Privacy Policy, About the project (link to SunHorizon web), Settings, and Logout (see Figure 33).

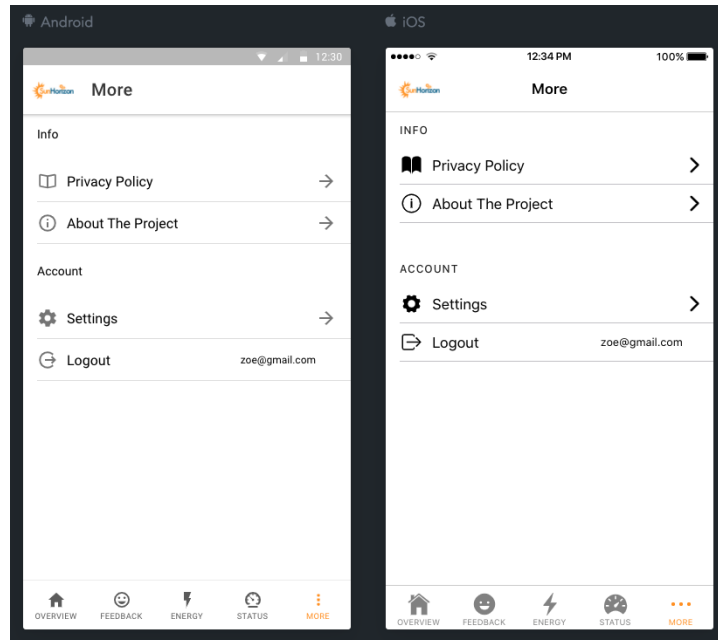


Figure 33. App: More section

5.2.7 Policy privacy

The Privacy policy was created using a template online, the complete text is found in Annex: A.1 Privacy Policy for Feedback app.

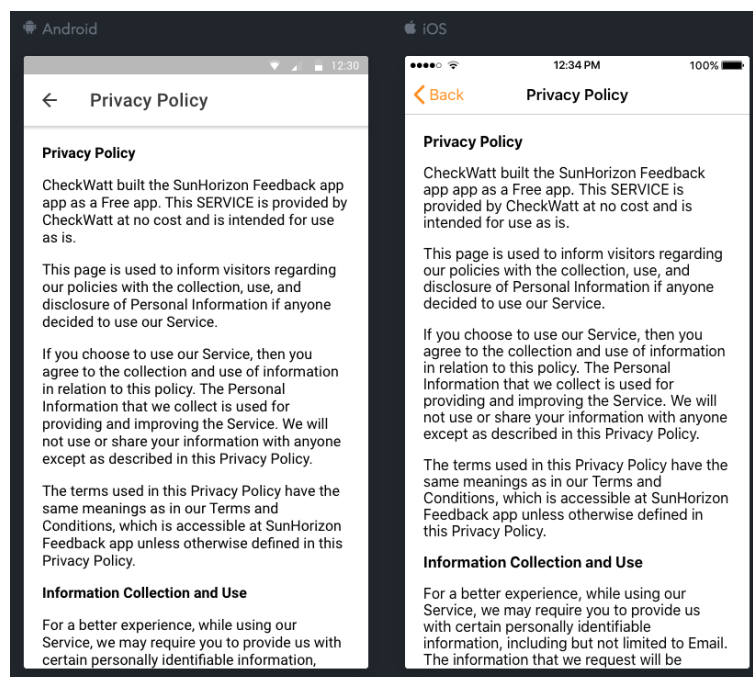


Figure 34. App: Policy Privacy section

5.2.8 Settings

The *Settings* section consists of three parts, the first is *Building/Household*, the second is *Account* and the third is *Sessions*. The Building/household part is where the user connects their account to a specific building, using a code. This code is unique for each building/household. To connect to an account, the user enters the building code and presses the *Connect-button*, a green text with the building name appears when the connection is completed. It is also possible to disconnect, simply by pressing the *Disconnect-button* once connected. (See Figure 35 and Figure 36).

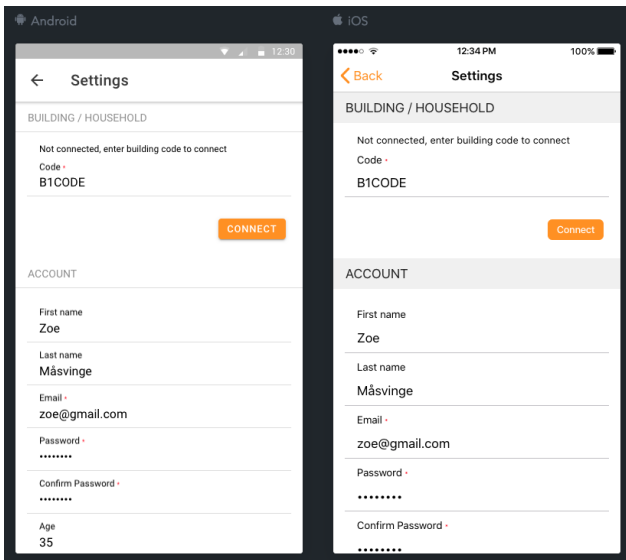


Figure 35. App: Settings connect account to building

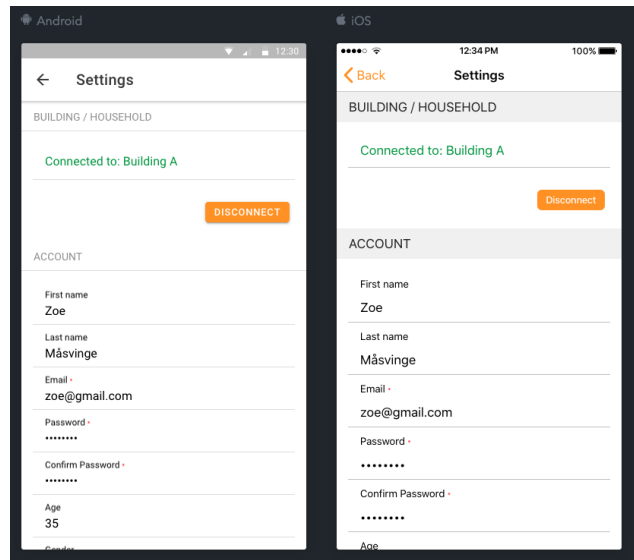


Figure 36. App: Settings disconnect from building

The Account part of the Settings section is a fairly straight forward function where the user can see and edit their account information. Once a field is updated it is also saved automatically without any other action required.

As for the Sessions part, this is where the user can see how many devices their account is used on with the possibility to logout from all (see Figure 37). The feature of allowing users to log in on multiple devices can also be used to create a family account. Ex. A person in a household creates an account and uses this to log in on other family members' devices, this way other family members can provide feedback completely anonymously with only one account.

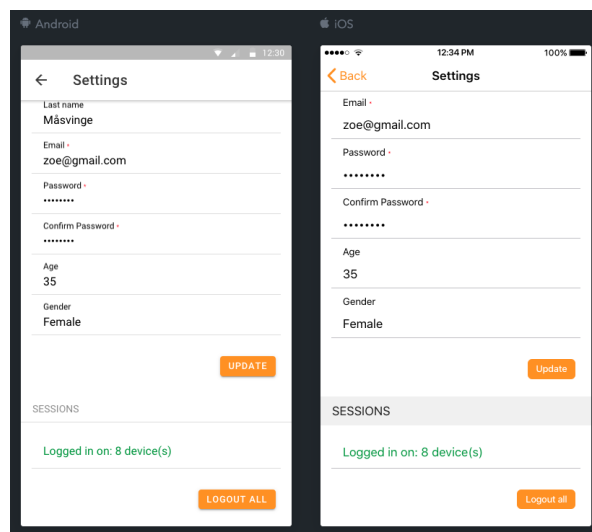


Figure 37. App: Settings sessions

6 Conclusion

The strategy of focusing the design and implementation of the app on the users' needs was the right way to go. Of course, this needs to be validated with real end-users, but the internal tests and discussions about the current version of the app have been positive. All requirements listed in the DOA have been implemented, some with dummy-data for now but with layout and functionality ready. Modularity has also been an important factor during the implementation, this to create a flexible app that can easily be modified to fit adjusted requirements and different types of installations.

One feature that has not been discussed in this deliverable is Notifications, the basic functionality has been implemented in the app but needs more testing and a server-side (that triggers notifications) to be finalized. This is planned to be done during the integration session.

The current version of the app uses dummy data and placeholders in some parts but will be replaced with real data (from the iSCAN APIs) after the integration. The next steps, integration and validation are the final steps to have a complete and ready to use app.

A. ANNEXES

A.1 Privacy Policy for Feedback app

Privacy Policy

CheckWatt built the SunHorizon Feedback app as a Free app. This SERVICE is provided by CheckWatt at no cost and is intended for use as is.

This page is used to inform visitors regarding our policies with the collection, use, and disclosure of Personal Information if anyone decided to use our Service.

If you choose to use our Service, then you agree to the collection and use of information in relation to this policy. The Personal Information that we collect is used for providing and improving the Service. We will not use or share your information with anyone except as described in this Privacy Policy.

The terms used in this Privacy Policy have the same meanings as in our Terms and Conditions, which is accessible at SunHorizon Feedback app unless otherwise defined in this Privacy Policy.

Information Collection and Use

For a better experience, while using our Service, we may require you to provide us with certain personally identifiable information, including but not limited to Email. The information that we request will be retained by us and used as described in this privacy policy.

The app does use third-party services that may collect information used to identify you.

Link to the privacy policy of third party service providers used by the app

[Google Play Services](#)

[Google Analytics for Firebase](#)

Log Data

We want to inform you that whenever you use our Service, in a case of an error in the app we collect data and information (through third party products) on your phone called Log Data. This Log Data may include information such as your device Internet Protocol ("IP") address, device name, operating system version, the configuration of the app when utilizing our Service, the time and date of your use of the Service, and other statistics.

Cookies

Cookies are files with a small amount of data that are commonly used as anonymous unique identifiers. These are sent to your browser from the websites that you visit and are stored on your device's internal memory.

This Service does not use these "cookies" explicitly. However, the app may use third party code and libraries that use "cookies" to collect information and improve their services. You have the option to either accept or refuse these cookies and know when a cookie is being sent to your device. If you choose to refuse our cookies, you may not be able to use some portions of this Service.

Service Providers

We may employ third-party companies and individuals due to the following reasons:

To facilitate our Service;

To provide the Service on our behalf;

To perform Service-related services; or

To assist us in analyzing how our Service is used.



We want to inform users of this Service that these third parties have access to your Personal Information. The reason is to perform the tasks assigned to them on our behalf. However, they are obligated not to disclose or use the information for any other purpose.

Security

We value your trust in providing us your Personal Information, thus we are striving to use commercially acceptable means of protecting it. But remember that no method of transmission over the internet, or method of electronic storage is 100% secure and reliable, and we cannot guarantee its absolute security.

Links to Other Sites

This Service may contain links to other sites. If you click on a third-party link, you will be directed to that site. Note that these external sites are not operated by us. Therefore, we strongly advise you to review the Privacy Policy of these websites. We have no control over and assume no responsibility for the content, privacy policies, or practices of any third-party sites or services.

Children's Privacy

These Services do not address anyone under the age of 13. We do not knowingly collect personally identifiable information from children under 13. In the case we discover that a child under 13 has provided us with personal information, we immediately delete this from our servers. If you are a parent or guardian and you are aware that your child has provided us with personal information, please contact us so that we will be able to do the necessary actions.

Changes to This Privacy Policy

We may update our Privacy Policy from time to time. Thus, you are advised to review this page periodically for any changes. We will notify you of any changes by posting the new Privacy Policy on this page.

This policy is effective as of 2020-09-09

Contact Us

If you have any questions or suggestions about our Privacy Policy, do not hesitate to contact us at support@checkwatt.se.