CALL H2020-SC1-FA-DTS-2018-2020
Trusted digital solutions and Cybersecurity in Health and Care
TOPIC DT-TDS-01-2019
Smart and healthy living at home

SMART BEAR

"Smart Big Data Platform to Offer Evidence-based Personalised Support for Healthy and Independent Living at Home"

D3.6 – Report on SMART BEAR @ Home Enabling Components v3

Due date of deliverable: 31/10/2023
Actual submission date: 31/01/2024

Grant agreement number: 857172
Start date of project: 01/09/2019

Lead contractor: CNR
Duration: 60 months

Project funded by the European Commission within the EU Framework Programme for Research and Innovation HORIZON 2020

Dissemination Level	
PU = Public, fully open, e.g. web	✓
CO = Confidential, restricted under conditions set out in Model Grant Agreement	
CI = Classified, information as referred to in Commission Decision 2001/844/EC.	
Int = Internal Working Document	

D3.6 – Report on SMART BEAR @ Home Enabling Components v3

Editor

Dimitrios G. Boucharas

Contributors

Dimitrios G. Boucharas (UOI)
Ioannis Pappas (UOI)
Fanis Kalatzis (UOI)
Kostas Tsiouris (UOI)
George Zissis (ATC)
Sotiria Antaranian (ATC)
Claudiu Petrescu (ITSS)
Anca Mihaela (ITSS)
Claudio Ardagna (UMIL)
Valerio Bellandi (UMIL)
Paolo Ceravolo (UMIL)
Ernesto Damiani (UMIL)
Jonatan Maggesi (UMIL)
Mattia Occhipinti (UMIL)
Nicola Paganotti (UMIL)

Reviewers

Lefteris Koumakis
Luigi Gallo
Dimitrios I. Fotiadis

The work described in this document has been conducted within the project SMART BEAR, started in September 2019. This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 857172

Executive Summary

The present deliverable provides an overview of the progress achieved with the third and final iteration of the development of a comprehensive federation of devices and sensors designed to seamlessly integrate with SMART BEAR's Home Automation system. The design and selected technology empower personal and home-based sensors to monitor a broad spectrum of physiological and health-related data with minimal user involvement. This monitoring is facilitated through a centralized interface delivered by a mobile application tailored to the end users. The deliverable encompasses six chapters, each of which is dedicated to addressing specific objectives and activities.

Chapter 1 commences by offering an in-depth review of the project's scope and effort allocation required to meet the objectives. Additionally, it acknowledges and confronts the challenges encountered in WP3.

Chapter 2 delves into the advancements achieved in HomeHub automation, harnessing open-source technologies to transcend vendor-specific communication and connectivity protocols. The analysis of the installation stages has led to the development of automation mechanisms that can be seamlessly implemented in the homes of study participants. These mechanisms are presented in the form of SD card-based software, thus simplifying the installation of the sensor ecosystem through equipment deployment.

Chapter 3 elaborates on the development of the mobile application, which strategically adopts key technological frameworks and integration schemes. These strategies enable the monitoring of study participants through a diverse range of smart devices. The application leverages APIs and SDKs from various vendors to establish connections with monitoring devices. This results in the provision of a unified solution, known as the SB@App, which attempts to minimize the need for user intervention.

Chapter 4 represents the finalized version of the aggregated measurement list, encompassing all devices and sensors derived measurements. Specifically, it provides a comprehensive presentation of measurements, field types, data values, and metadata, as requested by the reviewers. However, this chapter also plays a pivotal role in managing the data flow for the related work packages.

Chapter 5 focuses on enumerating the key updates to the Dashboard's functionalities capable of offering an exclusion assessment tool and providing updated analytics. A more detailed elaboration of these updates will be available in Deliverable 4.6.

Chapter 6 serves as a platform for outlining the methodology employed in monitoring activities, conditions, and interventions. This methodology is anticipated to be developed further in WP5. It also introduces use cases and offers a glimpse of envisioned user dashboard functionalities.

Contents

Definitions & Acronyms

Term	Meaning	Term	Meaning
IoT	Internet of Things	**USB**	Universal Serial Bus
openHAB	open Home Automation Bus - "open source, technology agnostic home automation platform which runs as the center of your smart home"	**MQTT message**	A formatted structure of the MQTT protocol used for the machine-to-machine conversation on a topic
Bindings	are the openHAB component that provides the interface to interact electronically with devices	**API**	Application Programming Interface
Things	are the first openHAB (software) generated representation of your devices	**REST**	Representational state transfer
Channels	are the openHAB (software) connection between “Things” and “Items” (see below)	**IP**	Internet Protocol
Items	are the openHAB (software) generated representation of information about the devices	**URL**	Uniform Resource Locator
openHAB Rules	that perform automatic actions (in its simplest form: if "this" happens, openHAB will do "that")	**HTTP**	Hypertext Transfer Protocol
Sitemap	is the openHAB (software) generated user interface (web site) that presents information and allows for interactions	**HTTPS**	Hypertext Transfer Protocol Secure
openHABian	A self-configuring Linux system setup specific to every openHAB user	**APK**	Android Package Format for Android Applications
VPN	Virtual Private Network	**JSON**	JavaScript Object Notation
LAN	Local Area Network	**ECG**	Electrocardiogram
Vue	A progressive framework for building user interfaces	**BP**	Blood Pressure - sBP=Systolic Blood Pressure, dBP=Diastolic Blood Pressure

Term	Meaning	Term	Meaning
JS	JavaScript programming language	**BMI**	Body Mass Index
Zigbee	A standard for low power, self-healing, mesh networks offering a complete and interoperable IoT solution for home and building automation	**SPO2**	Peripheral capillary oxygen saturation
Zigbee2MQTT	A software platform that allows to use your Zigbee devices without the vendors bridge or gateway	**UI**	User Interface
MQTT	Message Queuing Telemetry Transport is a lightweight messaging protocol designed for machine-to-machine communication in low bandwidth environment	**HA**	Hearing Aids
MQTT Broker	Is synonym of an MQTT server	**MVC**	Model-View-Controller
OSGi	A specification that describes a modular architecture for the Java programming language	**IDE**	Integrated Development Environment
Apache Tomcat	A Java based Application Server that implements the web profile of the JAVA EE	**SB**	SMART BEAR
JAVA EE	A full set of enterprise specifications implemented in the Java programming language	**OS**	Operating System
JAKARTA EE	JAKARTA is the new name of JAVA after JAVA EE adoption from Eclipse Foundation	**FCM**	Firebase Cloud Messaging
Spring BOOT	A Java framework (a "competitor" of JAVA EE)	**GDPR**	General Data Protection Regulation
CORS	Cross-Origin-Resource-Sharing	**AES-256**	Advanced Encryption Standard with 256 bits key length
SSL	Secure Sockets Layer	**BLE**	Bluetooth Low Energy
TLS	Transport Layer Security	**hPA**	hectopascal unit of pressure measurement

Term	Meaning	Term	Meaning
Mosquitto	An Open Source MQTT server that implements the MQTT protocol	**CLI**	Command Line Input
Philips Hue Bridge	A device that controls the Philips home automation	**FHIR**	Fast Healthcare Interoperability Resources
Xiaomi Aqara Gateway	A device that controls the Xiaomi home automation	**JPA**	Java Persistence API
Raspberry PI	A small hardware device that operates as an autonomous computer for low-demanding computational labour	**HR**	Heart Rate
ACL	Access Control List	**bpm**	Beats per minute
oAuth	Open-source authentication scheme that defines explicitly the delegated authentication process	**mm**	mili-meters, mili equals one of thousands
MQTT topic	A topic is string (a word) that the broker users to filter messages from the client	**ISO**	International Standards Organization
Vuex	A state management library for Vue.js applications	**HL7**	Health Level Seven International (HL7) is a not-for-profit, ANSI-accredited standards developing organization

List of Tables

List of Figures

1 Introduction

The SMART BEAR deliverable D3.6 details the latest advancements and upgrades made to the HomeHub and Mobile Application (SB@App) software to facilitate the seamless integration of smart home integration hubs and sensors, wearables, and medical/wellness devices utilised by SMART BEAR study participants (study participants or patients) while also providing the methodology utilized for the activities, conditions, and interventions monitoring and presenting new functionalities on the end-users' dashboard. This revision enchases the previous one (D3.4), which provided an in-depth look at the second iteration of the services on offer. This report's scope constitutes the third iteration of the development of the HomeHub automation and Mobile Application modules, polishing features that were previously evaluated in the Pilot of Pilots (PoP) and will be put to use in the main pilots.

Specifically, it details the progress in refining, validating, and otherwise improving the HomeHub and SB@App modules to better support the offered services. These improvements encompass a wide range of aspects, including but not limited to database service upgrades, the provision of additional analytics within the context of the SMART BEAR Mobile application (SB@App), the integration of additional – from the selection of those tested during the PoP period – medical/wellness devices and their respective data structuring, updated data synchronization schemes and improved communication pipelines, improvements to enhance the usability of the use interface of the SB@App, and introducing newly developed features such as cognitive games. To better capture the supported functionality and to facilitate its use by study participants, and the expansion of comprehensive user manuals composed of video and documents.

One of the primary objectives outlined in Work Package 3 (WP3) of the SMART BEAR project is to establish an efficient and secure out-of-the-box federation of smart devices within the SMART BEAR ecosystem. This objective is achieved through the collaborative efforts of two key components: the HomeHub and the SB@App. The HomeHub serves as a critical point of contact within the smart home ecosystem, responsible for configuring various smart home devices, aggregating the raw sensor data that originate from them and facilitating their transmission to the SB@App (and then to the SB@Cloud). This data gathering process is facilitated by the utilization of openHAB, a home automation platform, running on a dedicated Raspberry Pi processing unit. The HomeHub's role is pivotal in ensuring the seamless collection of data from a diverse range of smart home devices such as temperature, humidity, motion, and brightness sensors.

Conversely, the SB@App is designed to interface with both the HomeHub and the devices/sensors within the SMART BEAR ecosystem acting as the centralised node of the SMART BEAR solution where all data communication is unified. Its primary functions encompass the collection, transformation, display, and synchronization of the gathered data to SMART BEAR Cloud (SB@Cloud) once internet access is available. The synchronization process ensures that the usage data is securely transmitted (to the SB@Cloud) for further analysis and utilization, allowing higher level information inference. The capabilities of the SB@App extend beyond serving as a mere data handling module, as it offers to its end-users (study participants) the opportunity to engage in a variety of activities, including educating themselves through the OTAGO exercise program (i.e., a series of strength and balance exercises), enjoying cognitive games, and effectively planning and managing their nutrition.

The most critical aspect of WP3 is taggled with the successful collection, transformation and secure transmission of all study-participant associated (personalised) medical/sensor usage data in accordance with the technical specifications, is supported by developed features of the SB@App. This task is of paramount importance, as it involves the integration of numerous heterogeneous devices, including hubs, wearables, and medical/wellness devices, as well as the (re-)formatting of raw usage data generated by those into meticulously structured types (e.g., FHIR structures). The overarching goal is not solely limited such raw data collection but extends further to engaging study participants in actively monitoring their activities, health conditions, and overall well-being. This, in turn, contributes to an enhancement in their quality of life, as the collected data enables a comprehensive analysis of their status and enables informed decisions and actions.

On the other hand, data aggregation of home sensors raw data is a crucial facet of the data analysis, and summarization procedures within the HomeHub ecosystem enables the extraction of higher-level insights from continuously monitored environmental settings (mainly conditions inside the participant's residence). To achieve this, the ecosystem meticulously calculates aggregated measurements of smart home devices at five-minute intervals, which are then transmitted to a designated smartphone for subsequent relay to the SB@Cloud, where further comprehensive analysis will occur. This process effectively reduces data complexity, size and facilitates a comprehensive understanding of prevailing trends, patterns, and distinctive characteristics within the monitored data. Moreover, these aggregated measurements serve as a vital quality control indicator, summarizing data related to defects, errors, and deviations from established standards, thereby ensuring product quality throughout the smart home ecosystem.

As it is previously outlined in D3.2 and D3.4, the HomeHub sensory ecosystem encompasses both hardware and software components, strategically positioned within the residences of study participants. As indicated then, its primary purpose is to integrate and harmonize a diverse range of smart devices and sensors, facilitating the seamless collection and management of physiological and health-related measurements, including parameters such as temperature, humidity, and brightness levels (list of all parameters is presented in Section 4). The communication infrastructure connecting these devices and sensors is established using the ZigBee[1], which leverages the low-power technology framework as defined in the IEEE 802.15.4-2011 data communication protocol[2]. This protocol is specifically tailored to accommodate devices with low-power consumption profiles, and low-complexity operational needs. The crucial objective of this sensory ecosystem is to autonomously acquire measurements from the various home-based devices and sensors and transmit this data to the SB@App without necessitating direct intervention or interaction from study participants (as not considered necessary to have a technical background for configuring technical parameters of such hardware).

Interventions monitoring, an integral facet of this work package, seeks to formulate and implement mechanisms that enable the platform to effectively monitor the daily activities of study participants, the surrounding environmental context, and parameters associated with the project's specific conditions, alongside the interventions recommended by the platform. This comprehensive process involves evaluating whether these interventions were appropriately adhered to by the participants and assessing their resultant impact, among other aspects.

Over the last year, and taking into account end-user feedback during the PoP, the SMART BEAR dashboard (SB@Dashboard) has undergone significant enhancements both at the level of functionality and at the level of usability to better serve its utilisation from clinicians and technicians (roles: Clinical case manager, Help desk operator). These improvements include a

---

[1] Zigbee, IEEE 802.15.4-based specification: https://csa-iot.org/all-solutions/zigbee/

[2] IEEE Standard for Local and metropolitan area networks--Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs): https://standards.ieee.org/ieee/802.15.4/5050/

refined Exclusion Criteria tab for participant screening, ensuring adherence to Montreal Cognitive Assessment (MoCA) and Dexterity questionnaire scores. The SB@Dashboard outputs clearest notifications for acceptance or exclusion of a candidate. This newer version of the dashboard incorporates several types of descriptive and predictive analytics associated with all data been stored and analysed within the context of SB@Cloud, including those transmitted from HOLOBALANCE and SMART4HEALTH H2020 projects, associated to common study participants.

Moreover, from the perspective of the main pilots' initiation, a plethora of supplementary automation mechanisms have been either devised or enhanced to facilitate the large-scale deployment process. These enhancements encompass a broad spectrum of functionalities, including configuration expansion, the implementation of bash scripts, the utilization of camera scans to identify device MAC addresses, user-friendly authorization and set-up mechanisms, among various other manifestations, all contributing to the provision of robust support for the exceptionally demanding level of automation and scalability essential for this task. Subsequent to the receipt of gathered measurements from the HomeHub, the SB@App initiates a series of transformation procedures encompassing timestamping, device type expansion, and data structuring before effecting transmission to the SB@Cloud. This structured approach which contributes to the monitoring of the frequency of devices/sensors usage data production (supported by the SB@SecurityComponent, first version presented in D5.3), is instrumental in the event of a sensor failure, as the device type aids the identification of the device/sensor not contributing raw data as supposed to. Generally, the SB@App demonstrates its capacity to interpret data sets, by facilitating the provision of personalized recommendations (in the case a specific organisation/pilot wishes to activate this feature) tech-related alerts, or instructions that are harmonized with the users' daily activities.

This deliverable delineates the methodological approach employed to devise these pivotal mechanisms, a process to be further developed within the purview of WP4 and WP5. As a tangible outcome, a dedicated section on the participant dashboard is introduced, providing a granular overview of all devices, each assigned a color-coded status based on historical data and the corresponding intervention undertaken.

In the subsequent sections, comprehensive documentation is provided for the software tools and methodological workflows employed by both study participants and technical administrators across five key domains: the "SB@HomeHub", "SB@App", SB@Dashboard, activities, "conditions and interventions monitoring", and "Measurements, Data Types, Data Values and Metadata".

Table 1: Correspondence to the objective of Work Package 3.

WP3 Objectives	**Current status**
To provide efficient and secure out of the box federation of smart devices present in the SMART BEAR ecosystem;	OpenHAB[3] has consistently maintained its position as the designated software platform, serving as the vendor-agnostic integrator. Similarly, the Zigbee2MQTT[4] software has continued to fulfill its role effectively as a bridge between Zigbee-based smart home devices.

[3] The open Home Automation Bus (openHAB): https://www.openhab.org/docs/

[4] Zigbee Adapter interface: https://www.zigbee2mqtt.io/

To provide monitoring capabilities for end user activities, their conditions and the interventions applied; to provide the local analytics needed to infer higher level information from the raw data monitored;	The process of device selection was meticulously conducted, taking into consideration several key factors. These factors encompassed an in-depth assessment of the devices' supported functionalities, their connectivity features, and the accessibility of open Application Programming Interfaces (APIs). The primary focus was on how these APIs could enhance monitoring capabilities and facilitate advanced analytics within the SB@App, as well as promote user engagement.
To summarise data prior to transmission to the backend cloud;	HomeHub software, specifically openHAB, was strategically chosen for its capability to effectively manage the collection of measurements by overseeing and orchestrating data derived from various devices and sensors. Leveraging its robust capabilities, the software was adeptly employed to aggregate measurements, a process by which sensor data was summarized, prior to its transmission to the backend cloud infrastructure.
To develop the SMART BEAR mobile application and associated submodules;	The deliverable showcases a currently operational and final iteration of the HomeHub configuration, SB@App, users' dashboard, and interventions monitoring.

Table 2: Fulfilment of the expected deliverable expectations.

Deliverable Description as described in the Grant Agreement	Actual Implementation
Report on SMART BEAR @ Home Enabling Components v3: This deliverable will be a report documenting the components included in D3.5. D3.6 will be an amended version of D3.4 and D3.2, which in addition to the material included in these earlier deliverables, will include the documentation of all the new capabilities of the relevant components. The task responsible for the development of each of the relevant components will also	In this document, the components constituting D3.5 are delineated with respect to usability, interconnection and integration. Each Task, which is dedicated to specific aspects of smart home environment creation, makes substantial contributions to the objectives outlined in D3.4. **Chapter 2** expounds upon the integration process involving the second Xiaomi Aqara temperature and humidity sensor positioned within the bedroom area. Furthermore, it offers a comprehensive insight into the unification of home devices, and the subsequent development of automation services that are geared towards enhancing the efficiency and scalability of large-scale deployment. **Chapter 3** of this document presents the advancements achieved in the realm of mobile application development. This progress encompasses the incorporation of two cognitive games, the establishment of a comprehensive data gathering protocol, the implementation of a reset feature tailored for smart scales, the

be responsible for the provision of the documentation of the component in this deliverable.	finalization of data formatting procedures for hearing aids, and the seamless integration of three new Beurer devices, specifically, an oximeter, a thermometer, and a set of scales. Additionally, this chapter underscores the significant adjustments made to the user interface (UI), which includes the introduction of a HomeHub screen dedicated to the second temperature and humidity sensor, as well as the notable upgrades made to the authorization services and OTAGO localization. **Chapter 4** serves as an extensive exposition, providing details pertaining to the measures associated with various sensors. These measures encompass a comprehensive examination of data types, data fields, and metadata for each sensor. These sensor-related details are closely linked with the activities, interventions, and monitoring procedures outlined in **Chapter 6**. Furthermore, they are integral to the local analytics framework as delineated in **Chapter 5**.
The delivered components will include: **at least 5** heterogeneous devices covered by the monitoring solution (KPI-1); **all 9** parameters inferred by basic analytics (KPI-4);	KPI-1: The final iteration of the Home Automation components has been curated t**o encompass a total of 14 diverse devices**, as comprehensively detailed in Table 3. These devices possess the inherent capability to capture pivotal measurements pertaining to daily life and living. Concretely, this ensemble comprises the following elements: a motion sensing device (i.e., Philips Hue Motion) , a light sensing device (i.e., Philips Smart Bulb), two humidity and temperature sensing devices (i.e., Xiaomi Aqara Temperature and Humidity Sensor), an integration hub (i.e., Raspberry Pi utilizing openHAB), two bridge devices (i.e., Philips Hue Bridge, Sonoff USB Dongle), a smartwatch (i.e., Garmin Venu SQ and Garmin Vivo), a smart oximeter (i.e., iHealth Smart Air Plus), a thermometer (i.e., iHealth Themo Pro and Withings Smart Thermometer), hearing aids (e.g., OTICON miniTIRE T), a scale (i.e., Withings Body+ and iHealth Scale Fit), a blood pressure monitor (i.e., OMRON M7 and Withings BPM Core), and a mobile smartphone (i.e., Samsung Galaxy A13 & Xiaomi Redmi 10c), which serves as the operational platform for the mobile application. KPI-4: The SMART BEAR solution sets forth a comprehensive objective, **encompassing a total of nine high-level parameters**. These parameters encompass a wide spectrum of devices, including Smart Home devices, sensory devices, integration hubs, wearables like smartwatches, mobile devices such as smartphones, and medical devices, namely, smart blood pressure monitors, hearing aids, scales, thermometers, and oximeters. The primary aim of these devices is to effectively mitigate and minimize daily risk factors that individuals may encounter. These risk factors include but are not limited to noise exposure, physical inactivity,

	sleep disturbances, malnutrition, fluctuations in body weight, abnormal blood pressure levels, cardiovascular and respiratory symptoms, non-compliance with prescribed medication regimens, suboptimal light levels, adverse weather conditions, behavioral and emotional changes, habitual substance use, as well as monitoring parameters like body temperature and blood oxygen saturation (SpO2), among numerous others.
applications for participant screening and monitoring and delivery of the SMART BEAR platform interventions (KPI-12);	KPI-12: The scope of the screening process has been expanded to encompass dexterity and the Montreal Cognitive Assessment questionnaires (MOCA). Following completion by the clinicians, these assessments yield scores that determine the inclusion or exclusion of participants in the SMART BEAR project (Section 5.1). Apart from this enhanced screening, the SMART BEAR platform is now also equipped for both monitoring and intervention delivery. The current deliverable offers a more comprehensive elucidation of the methodology for participants' monitoring regarding their activities and conditions, resulting in specific interventions, with its subsequent implementation occurred for WP5 (Chapter 6).
and integration of all needed sensing and actuation mechanisms such as those described in Table 2 (KPI-20).	KPI-20: **All needed sensing and actuation mechanisms are incorporated** within the SMART BEAR solution as they are previously presented in detail and specified in Table 2 of the Grant Agreement.

2 SMART BEAR HomeHub (Delta – Updates)

The HomeHub is an integral component of the SMART BEAR solution, encompassing both hardware and software aspects. It plays a pivotal role in the deployment of home automation technology within people's residences, serving as the central hub where various devices and sensors can connect and be coordinated. This kind of technology is commonly referred to by different names, including gateways, bridges, and hubs, depending on the manufacturer. For instance, Philips Hue offers "Hue Bridges," while Xiaomi provides "Aqara Gateways." Regardless of the terminology, the fundamental requirement for a hub to facilitate centralized control of devices from different suppliers remains consistent. This central node enables communication between systems using diverse protocols like WiFi and ZigBee. In the context of the SMART BEAR project, openHAB[5] was selected as part of the solution due to its capacity for interoperability, which is essential to accommodate the wide array of the heterogeneous devices involved.

Employing a Raspberry Pi[6], the openHAB software platform establishes a unified point for administration and access, facilitating seamless interactions between humans and the machine. In this section, we delve into the architecture and communication pipelines of openHAB, shedding light on the built-in services that underpin the automation of the HomeHub. To streamline the handling of vast volumes of data, three pivotal services have been developed in this domain. The Persistence service serves the purpose of centralizing information, collecting and storing it in standardized formats. The Aggregation service, on the other hand, compiles data summaries using mathematical functions as specified by clinical professionals. Cloud computing, as a distinct service, focuses exclusively on data administration and analysis through remote servers. Consequently, the SB@App has the capability to provide users with valuable information and recommendations, such as suggesting the optimal time to turn off lights on a bright day, communicated through notifications and suggestions.

Considering the scale of the SMART BEAR main pilots across Europe, the installation process for HomeHub equipment has been optimized to minimize technical personnel effort, time, and technical expertise. As a result, individuals responsible for the installation do not need to possess advanced technical skills in areas like programming or sensor installation. To illustrate the disparities in automation, effort, and time required for HomeHub setup, this section presents the configuration advancements and user interfaces enhancements. Furthermore, a comprehensive instructional manual is provided to expedite the setup process and ensure proper device functionality and communication.

2.1 *HomeHub New Functionalities*

During the course of HomeHub development in the reported period, a new requirement emerged, necessitating the installation of a second sensor within the participants' residences. Consequently, the integration of a second Aqara Xiaomi sensor was executed, complemented by the collection and aggregation of the corresponding measurements. These measurements were then incorporated into the broader measurement set encompassing various environmental conditions, thus enabling the synthesis of a comprehensive view on the home's status. To streamline the registration process for the Aqara Xiaomi sensors, a fully functional web interface was implemented. Simplification of the acquisition of the Universal Unique Identifier (UUID) and secret key pair, which can now be obtained through an almost automatic bash script, occurs for the process of Cloud registration. Standardization of HomeHub measurements and their

[5] https://www.openhab.org/

[6] https://www.raspberrypi.org/

respective variables on the Raspberry Pi was also undertaken to facilitate compatibility with the mobile application and, subsequently, with the cloud storage infrastructure of the SMART BEAR project.

Furthermore, the rules supervising HomeHub were expanded to encompass all the final measurements, whether they were simple or aggregated in nature. This enhancement holds the potential to facilitate comprehensive analysis of the data collected within the study participants' homes, providing deeper insights into the project's objectives. To ensure the longevity of HomeHub measurements in local storage, a new persistence service was integrated. This service enables the retention of a vast amount of data for extended durations, without posing any issues throughout the duration of the pilot phase, which spans 12 months.

2.1.1 *Second Humidity/Temperature Sensor Integration*

The incorporation of a second Aqara humidity/temperature sensor within the residential spaces of the study participants was advanced further. This strategic deployment of two Aqara humidity/temperature sensors within the participants' homes served the purpose of enhancing the monitoring capabilities by conducting temperature and humidity measurements in distinct rooms. This approach is adopted to capture and account for potential physiological variations that may exist across different rooms within these households. By acquiring data from multiple sensors, a more comprehensive and nuanced understanding of the environmental conditions can be achieved, contributing to the robustness and reliability of the research findings.

The versatility and capabilities of the open-source home automation platform, OpenHAB, played a pivotal role in this endeavor. OpenHAB provided the necessary tools and functionalities to seamlessly integrate and manage a diverse array of smart devices and sensors. In this specific case, the potential of this platform to optimize the utilization of the newly introduced sensor was harnessed. To accomplish this, a meticulously defined set of steps was followed, enabling the integration of the new sensor into the pre-existing setup. The integration of the second Aqara humidity/temperature sensor underscored the adaptability and scalability of OpenHAB, empowering it to refine the data collection processes and enhance the overall efficacy of the SMART BEAR project.

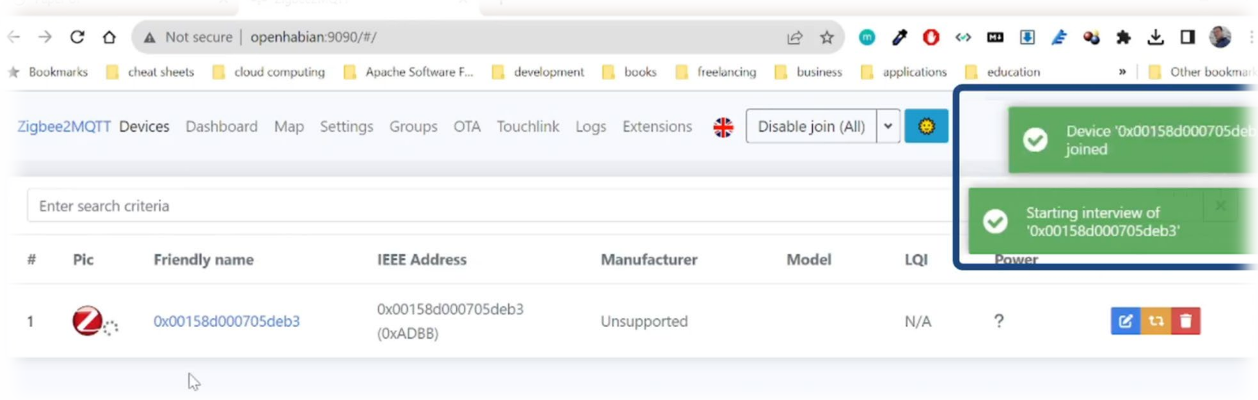


Figure 1: Device discovery.

The process of integrating both Xiaomi Aqara Temperature and Humidity sensors, **each installed separately**, into the SMART BEAR project's system is characterized by a series of meticulously executed steps to ensure seamless functionality and effective data acquisition.

These steps, which have been diligently followed, reflect the project's commitment to maintaining a robust and reliable setup for home automation and environmental monitoring.

1. *Device Discovery*: The initial phase involves the discovery of the new sensor (Figure 1). With the appropriate binding for the device already installed, the Zigbee2MQTT built-in web-based frontend is utilized to detect the sensor. This typically entails activating the new sensor's pairing mode (e.g., pressing the appropriate pairing button) and allowing Zigbee2MQTT to identify it. The next subchapter (e.g., Chapter 2.1.2) will provide detailed instructions regarding this step, ensuring a smooth integration process.
2. *Items Creation*: To represent the sensor's data effectively, new Items are created. Two separate items are generated per sensor, one for the humidity and another for the temperature, along with their corresponding aggregated values. This is achieved through manual editing of the Items configuration files, as exemplified in Table 3.

Table 3: Xiaomi Aqara Temperature and Humidity Sensor Items.

Number AqaraSensorTemperatureLivingroom "Temperature #1 - Living room (Aqara sensor) [%.1f C]" {channel="mqtt:topic:habBroker:homeaqara:temperature",autoupdate="true"}
Number AqaraSensorHumidityLivingroom "Humidity #1 - Living room (Aqara sensor) [%.1f %%]" {channel="mqtt:topic:habBroker:homeaqara:humidity",autoupdate="true"}
Number AqaraSensorAtmosphericPressureLivingroom "Atmospheric pressure #1 - Living room (Aqara sensor) [%.1f hPa]" {channel="mqtt:topic:habBroker:homeaqara:pressure",autoupdate="true"}
Number AqaraSensorBatteryLivingroom "Battery #1 - Living room (Aqara sensor) [%.1f %%]" {channel="mqtt:topic:habBroker:homeaqara:battery",autoupdate="true"}
Number AqaraSensorBatteryBedroom "Battery #2 - Bedroom (Aqara sensor) [%.1f %%]" {channel="mqtt:topic:habBroker:homeaqara:battery2",autoupdate="true"}
Number AqaraSensorTemperatureBedroom "Temperature #2 - Bedroom (Aqara sensor) [%.1f C]" {channel="mqtt:topic:habBroker:homeaqara:temperature2",autoupdate="true"}
Number AqaraSensorHumidityBedroom "Humidity #2 - Bedroom (Aqara sensor) [%.1f %%]" {channel="mqtt:topic:habBroker:homeaqara:humidity2",autoupdate="true"}
Number AqaraSensorAtmosphericPressureBedroom "Atmospheric pressure #2 - Bedroom (Aqara sensor) [%.1f hPa]" {channel="mqtt:topic:habBroker:homeaqara:pressure2",autoupdate="true"}

3. *Link Items to Channels*: The newly created Items are linked to the appropriate Channels provided by the Binding. This linkage is established by editing the Items' configuration files, ensuring a coherent connection between data sources and the system.
4. *Creation of Rules and Automation*: To respond effectively to changes in the newly added sensor data, new rules and automation routines are established within the openHAB framework. These rules govern the system's behavior in response to sensor input, enhancing its adaptability and functionality.
5. *User Interface Configuration*: A user-friendly interface for the openHAB setup is configured to facilitate convenient monitoring of the new sensor's data via the web interface. This step ensures that project stakeholders have an easy understanding to vital information.

6. *Testing and Fine-Tuning*: Testing is conducted to validate the functionality of the newly integrated setup. This process involves verifying the accuracy of sensor data reporting and ensuring that automation rules operate as intended. Fine-tuning is performed to address any discrepancies or optimization opportunities identified during testing.
7. *Documentation*: The documentation of the system is extended to include comprehensive information about the newly added sensor[7]. This documentation serves as a valuable resource for technicians involved in setting up, troubleshooting, and maintaining the openHAB system. It contributes to the long-term sustainability and reliability of the project's infrastructure.

In brief, these steps underscore the commitment to precision and reliability in the integration of the Xiaomi Aqara Temperature and Humidity sensors, ultimately enhancing the SMART BEAR project's capabilities in home automation and environmental monitoring.

2.1.2 *Aqara Devices Registration and Pairing Process User Interface*

Zigbee2mqtt[8] stands as a widely recognized open-source project, serving as a vital enabler for the seamless integration of Zigbee devices, including humidity and temperature sensors, into home automation systems. To facilitate the process for the technicians and streamline the addition of Aqara sensors within the openHAB setup, a new procedure was devised, capitalizing on the capabilities of the Zigbee2mqtt Frontend. A new configuration (e.g., yaml file) was introduced, as it is exhibited in Table 4, to create a frontend web interface.

Table 4: The new configuration yaml file.

```
homeassistant: false
permit_join: true
mqtt:
  base_topic: zigbee2mqtt
  server: mqtt://localhost
serial:
  port: /dev/ttyACM0
frontend:
  port: 9090
```

In this new configuration, the addition of front-end functionality is significant, where port number 9090 has been designated as the entry point. Technicians can conveniently access this web frontend by entering the IP address of the host machine running Zigbee2MQTT, followed by the specified port, within their web browser. The steps involved in integrating an Aqara sensor through the Zigbee2mqtt web interface are outlined below:

1. *Access the Zigbee2mqtt Web Interface*: Initiate the process by opening a web browser and entering the IP address or hostname of the machine where Zigbee2mqtt is operational, along with the designated port number. In this specific scenario, the URL is http://openhabian:9090/ (Figure 2).

[7] https://drive.google.com/file/d/10vmDun03BIvMnDOO7a8ScBZDj9ZGkvS9/

[8] https://www.zigbee2mqtt.io/

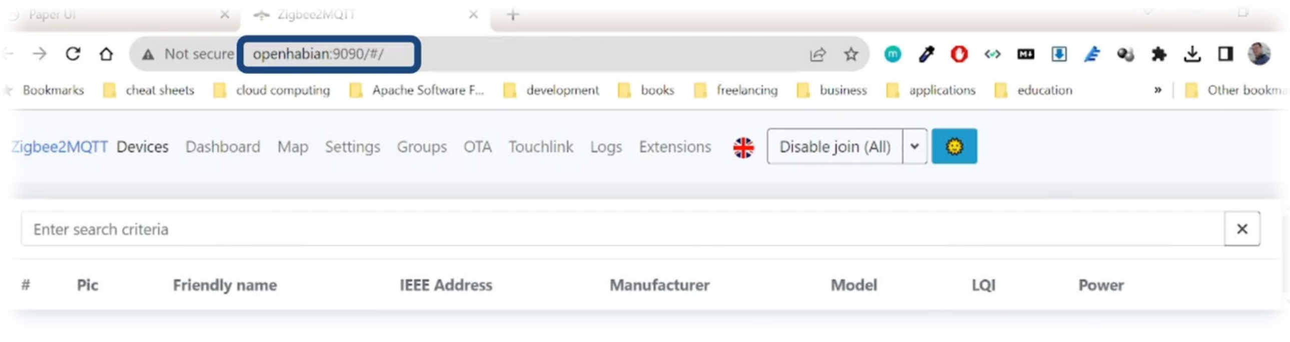


Figure 2: Accessing http://openhabian:9090/.

2. *Preparation of the USB Dongle*: Within the Zigbee2mqtt web interface, navigate to the "Permit Join" section, where the Zigbee coordinator (e.g., USB dongle) is set into pairing mode (Figure 3).

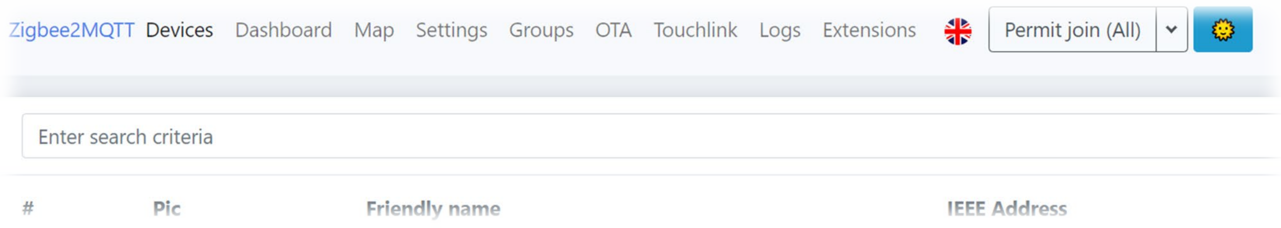


Figure 3: Preparing the USB dongle.

3. *Pairing Process*: Zigbee2mqtt commences the discovery of new devices. Upon detecting the Aqara sensor, it promptly provides information about the device through the web interface. Simultaneously, the technician should engage the pairing mode on the sensor by pressing the designated button (Figure 4).

Figure 4: a) Device information through the web interface; b) device in pairing mode.

4. *Add the Sensor to Zigbee2mqtt's Network*: Within the web interface, assign a friendly name (e.g., aqaraht or aqaraht2) to the Aqara sensor to facilitate specific identification (Figure 5). Subsequently, save the configuration changes.

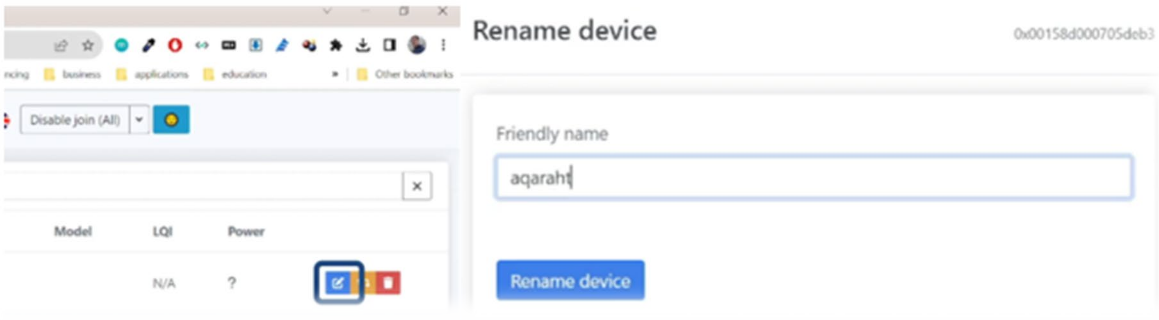


Figure 5: a) Device renaming quick button; b) device renaming.

5. *Integrate with openHAB*: With the Aqara sensor successfully added and configured within Zigbee2mqtt, it is automatically integrated into openHAB based on the predefined configuration files created for openHAB.
6. *Testing and Automation*: Testing of the sensor's data within openHAB is performed to ensure accurate reporting and functionality, further validating the integration process (Figure 6).

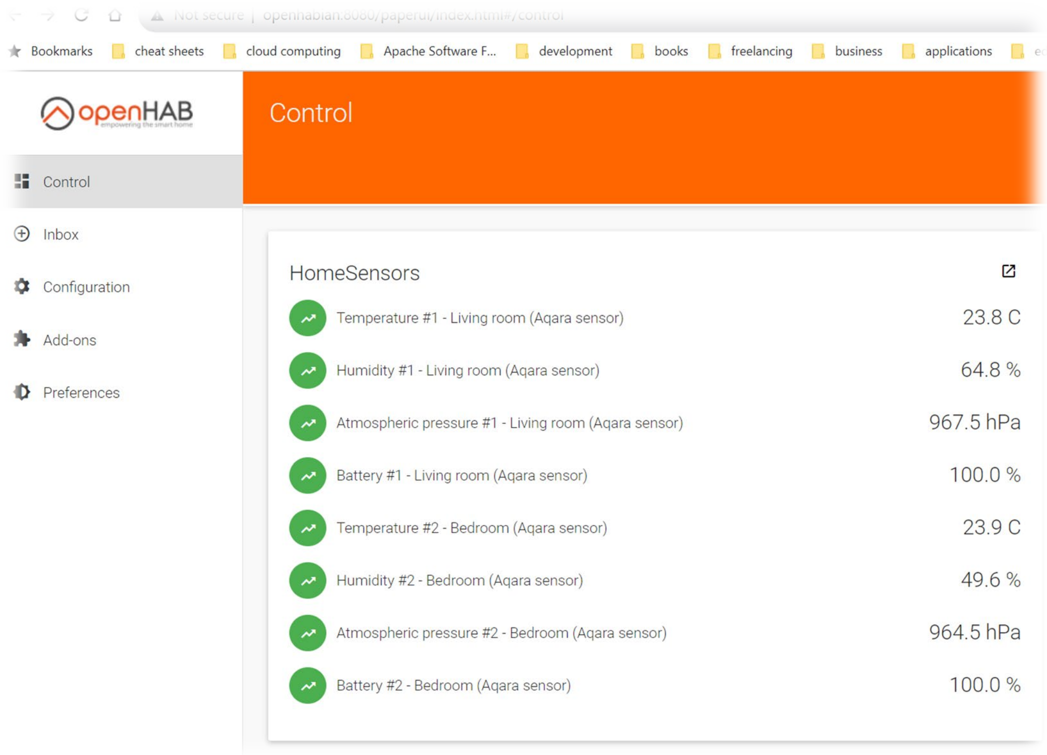


Figure 6: Display of current measurements locally.

This comprehensive procedure significantly simplifies the integration of Aqara sensors into the openHAB environment, leveraging the Zigbee2mqtt platform's capabilities to enhance the SMART BEAR project's home automation system with efficiency and precision.

2.1.3 *Automation of Cloud Registration*

This stage in the project pertains to the seamless integration of participants' residences into the openHAB cloud ecosystem, a critical step in aligning locally gathered measurements with the online domain. Beyond the inherent convenience it offers, this integration holds multifaceted significance. It not only facilitates remote monitoring but also serves as the conduit for data migration to the mobile application and establishes a reliable backup mechanism.

By tethering the residential setups to the openHAB cloud, an avenue for real-time environmental metric observation is unveiled. This means that someone with credentials at hand can remotely access and evaluate ongoing measurements without the need for physical presence. This capability not only augments flexibility but also empowers users with heightened control over environmental parameters, underscoring the value of this integration in optimizing the SMART BEAR project's objectives.

In a deliberate effort to enhance the user-friendliness of this process for technicians, a Bash script has been crafted to streamline the acquisition of essential UUID and Secret values for OpenHAB Cloud Registration. The Bash script operates by monitoring the creation of two specific files, namely /var/lib/openhab2/openhabcloud/secret and /var/lib/openhab2/uuid. Once these files come into existence, the script extracts and displays their contents. It is strategically positioned for execution and endowed with the requisite permissions (e.g., chmod +x script.sh) to function as a standalone script, ensuring a seamless and user-friendly experience for technicians involved in this crucial and previously not-automated phase of the project. An illustrative depiction of this script is presented in Figure 7 as the user triggers the "ohc" in cmd.

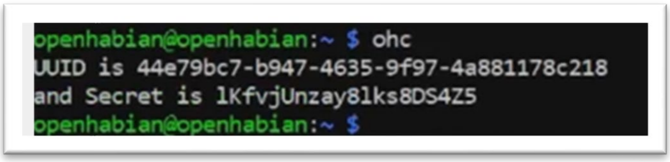


Figure 7: Automation of cloud registration key retrieval.

2.1.4 *Standarization of HomeHub Measurements*

In configuring the openHAB setup, a fundamental step entails the creation of Items configuration files, each bearing specific names tailored to the individual Items within the smart home environment. This process serves to delineate and define the Items (Table 5) that represent various integrated devices and data points. Each Item is intricately linked to a particular device or functionality and can be associated with channels provided by Bindings, facilitating the transmission and reception of data.

Table 5: Items definitions.

```
// Hue light bulb brightness
Dimmer HueLightBulbBrightness     { channel="hue:0100:1:brightness",autoupdate="true" }

// Hue light illuminance (Hue motion sensor)
Number:Illuminance  HueMotionSensorLightIlluminance  {  channel="hue:0106:1:light-level-sensor:illuminance",autoupdate="true" }

// Room temperature (Hue motion sensor)
```

```
Number:Temperature HueMotionSensorTemperature { channel="hue:0302:1:temperature-sensor:temperature",autoupdate="true" }

// Presence (Hue motion sensor)
Switch HueMotionSensorPresence { channel="hue:0107:1:motion-sensor:presence",autoupdate="true" }

//Last Updated (Hue motion sensor)
DateTime HueMotionSensorLastUpdate { channel="hue:0107:1:motion-sensor:last_updated",autoupdate="true" }

//Hue motion sensor Battery percentage
Number HueMotionSensorBatteryPercentage { channel="hue:0107:1:motion-sensor:battery_level",autoupdate="true" }

//Switch MotionSensorLowBattery { channel="hue:0107:1:motion-sensor:battery_low",autoupdate="true" }

// Scenes
//String LightScene { channel="hue:bridge:1:scene"}
```

A noteworthy addition to the Items configuration files is the introduction of new files aimed at the aggregation and tracking of diverse sensor data measurements. These measurements hold great utility in the pursuit of monitoring and analyzing trends within smart home ecosystems. These Items (Table 6) are seamlessly integrated into the openHAB Rules and various facets of the configuration, facilitating interaction with and visualization of the aggregated sensor data.

Table 6: Aggregated Items definitions.

```
Number:Illuminance HueMotionSensorLightIlluminanceAggMin
Number:Illuminance HueMotionSensorLightIlluminanceAggMax
Number:Illuminance HueMotionSensorLightIlluminanceAggAvg

Dimmer HueLightBulbBrightnessAggMin
Dimmer HueLightBulbBrightnessAggMax
Dimmer HueLightBulbBrightnessAggAvg

Number:Temperature HueMotionSensorTemperatureAggMin
Number:Temperature HueMotionSensorTemperatureAggMax
Number:Temperature HueMotionSensorTemperatureAggAvg

Number AqaraSensorHumidityLivingroomAggMin
Number AqaraSensorHumidityLivingroomAggMax
Number AqaraSensorHumidityLivingroomAggAvg
```

```
Number AqaraSensorAtmosphericPressureLivingroomAggMin
Number AqaraSensorAtmosphericPressureLivingroomAggMax
Number AqaraSensorAtmosphericPressureLivingroomAggAvg

Number AqaraSensorTemperatureLivingroomAggMin
Number AqaraSensorTemperatureLivingroomAggMax
Number AqaraSensorTemperatureLivingroomAggAvg

Number AqaraSensorHumidityBedroomAggMin
Number AqaraSensorHumidityBedroomAggMax
Number AqaraSensorHumidityBedroomAggAvg

Number AqaraSensorAtmosphericPressureBedroomAggMin
Number AqaraSensorAtmosphericPressureBedroomAggMax
Number AqaraSensorAtmosphericPressureBedroomAggAvg

Number AqaraSensorTemperatureBedroomAggMin
Number AqaraSensorTemperatureBedroomAggMax
Number AqaraSensorTemperatureBedroomAggAvg
```

For the Aqara sensors a specialized configuration, differentiating between two distinct rooms and their corresponding environmental data has been devised (Table 7).

Table 7: Aqara Items, Channels, Things configuration, and definitions.

```
Number AqaraSensorTemperatureLivingroom "Temperature #1 - Living room (Aqara sensor) [%.1f C]" {channel="mqtt:topic:habBroker:homeaqara:temperature",autoupdate="true"}
Number AqaraSensorHumidityLivingroom "Humidity #1 - Living room (Aqara sensor) [%.1f %%]" {channel="mqtt:topic:habBroker:homeaqara:humidity",autoupdate="true"}
Number AqaraSensorAtmosphericPressureLivingroom "Atmospheric pressure #1 - Living room (Aqara sensor) [%.1f hPa]" {channel="mqtt:topic:habBroker:homeaqara:pressure",autoupdate="true"}
Number AqaraSensorBatteryLivingroom "Battery #1 - Living room (Aqara sensor) [%.1f %%]" {channel="mqtt:topic:habBroker:homeaqara:battery",autoupdate="true"}
Number AqaraSensorBatteryBedroom "Battery #2 - Bedroom (Aqara sensor) [%.1f %%]" {channel="mqtt:topic:habBroker:homeaqara:battery2",autoupdate="true"}
Number AqaraSensorTemperatureBedroom "Temperature #2 - Bedroom (Aqara sensor) [%.1f C]" {channel="mqtt:topic:habBroker:homeaqara:temperature2",autoupdate="true"}
Number AqaraSensorHumidityBedroom "Humidity #2 - Bedroom (Aqara sensor) [%.1f %%]" {channel="mqtt:topic:habBroker:homeaqara:humidity2",autoupdate="true"}
Number AqaraSensorAtmosphericPressureBedroom "Atmospheric pressure #2 - Bedroom (Aqara sensor) [%.1f hPa]" {channel="mqtt:topic:habBroker:homeaqara:pressure2",autoupdate="true"}
```

Specifically:
For the Aqara Sensor in the Living Room:

- Number AqaraSensorTemperatureLivingroom: This item represents the temperature data from the Aqara sensor in the living room. It is linked to the temperature channel of the MQTT topic.
- Number AqaraSensorHumidityLivingroom: This item represents the humidity data from the Aqara sensor in the living room. It is linked to the humidity channel of the MQTT topic.
- Number AqaraSensorAtmosphericPressureLivingroom: This item represents the atmospheric pressure data from the Aqara sensor in the living room. It is linked to the pressure channel of the MQTT topic.
- Number AqaraSensorBatteryLivingroom: This item represents the battery level data from the Aqara sensor in the living room. It is linked to the battery channel of the MQTT topic.

For the Aqara Sensor in the Bedroom:

- Number AqaraSensorBatteryBedroom: This item represents the battery level data from an Aqara sensor in the bedroom. It is linked to the battery2 channel of an MQTT topic.
- Number AqaraSensorTemperatureBedroom: This item represents the temperature data from an Aqara sensor in the bedroom. It is linked to the temperature2 channel of an MQTT topic.
- Number AqaraSensorHumidityBedroom: This item represents the humidity data from an Aqara sensor in the bedroom. It is linked to the humidity2 channel of an MQTT topic.
- Number AqaraSensorAtmosphericPressureBedroom: This item represents the atmospheric pressure data from an Aqara sensor in the bedroom. It is linked to the pressure2 channel of an MQTT topic.

These Item definitions allow openHAB to receive data from the specified MQTT topics and use that data in the home automation Rules, sitemaps, or other configurations. Formatting to the data (e.g., "%.1f C" for temperature) was also applied to control how it is displayed in the user interface.

2.1.5 *Rules Expansion*

A comprehensive array of fundamental analytics has been crafted within the framework of openHAB rules. These rules have been devised to facilitate the systematic collection of a diverse spectrum of sensor data, encompassing both basic and aggregated measurements. Among the key data points gathered are the minimum, maximum, and average values derived from the aggregated sensor readings, which are computed over discrete 5-minute intervals. This analytical framework has been thoughtfully designed to operate at regular intervals, driven by a cron-based trigger.

The configuration file housing these rules serves as the cornerstone of this analytical infrastructure, offering a blueprint for orchestrating the collection, processing, and storage of sensor data. By virtue of this structured framework, openHAB leverages the power of automation to distill meaningful insights from the raw sensor data, enabling users to gain valuable insights into their connected environments. This application of rules-based analytics enhances the overall efficiency, functionality, and intelligence of openHAB-powered smart ecosystems. The rules configuration file is provided in Table 8:

Table 8: Rules configuration.

```
rule "Set 5 minutes max, min and avg HueMotionSensorTemperature"
when
	//Time cron "0 0 0 * * ?"
	//System started or
	Time cron "0 0/5 * * * ?"
	//Time cron "0 0/4 * 1/1 * ? *"
then
   val max5HueMotionSensorTemperature = (HueMotionSensorTemperature.maximumSince(now.minusMinutes(5), "rrd4j").state as DecimalType).doubleValue
   val min5HueMotionSensorTemperature = (HueMotionSensorTemperature.minimumSince(now.minusMinutes(5), "rrd4j").state as DecimalType).doubleValue
   val avg5HueMotionSensorTemperature = (HueMotionSensorTemperature.averageSince(now.minusMinutes(5), "rrd4j") as DecimalType).doubleValue

   if( max5HueMotionSensorTemperature != null && min5HueMotionSensorTemperature != null && avg5HueMotionSensorTemperature !=null) {

HueMotionSensorTemperatureAggMax.postUpdate(max5HueMotionSensorTemperature)

HueMotionSensorTemperatureAggMin.postUpdate(min5HueMotionSensorTemperature)

HueMotionSensorTemperatureAggAvg.postUpdate(avg5HueMotionSensorTemperature)
   }
end

rule "Set 5 minutes max, min and avg HueMotionSensorLightIlluminance"
when
	//Time cron "0 0 0 * * ?"
	//System started or
	Time cron "0 0/5 * * * ?"
	//Time cron "0 0/4 * 1/1 * ? *"
then
   val max5HueMotionSensorLightIlluminance = (HueMotionSensorLightIlluminance.maximumSince(now.minusMinutes(5), "rrd4j").state as DecimalType).doubleValue
   val min5HueMotionSensorLightIlluminance = (HueMotionSensorLightIlluminance.minimumSince(now.minusMinutes(5), "rrd4j").state as DecimalType).doubleValue
```

```
  val avg5HueMotionSensorLightIlluminance = (HueMotionSensorLightIlluminance.averageSince(now.minusMinutes(5), "rrd4j") as DecimalType).doubleValue

  if( max5HueMotionSensorLightIlluminance != null && min5HueMotionSensorLightIlluminance != null && avg5HueMotionSensorLightIlluminance !=null) {

HueMotionSensorLightIlluminanceAggMax.postUpdate(max5HueMotionSensorLightIllumina nce)

HueMotionSensorLightIlluminanceAggMin.postUpdate(min5HueMotionSensorLightIllumina nce)

HueMotionSensorLightIlluminanceAggAvg.postUpdate(avg5HueMotionSensorLightIlluminan ce)
  }
end

rule "Set 5 minutes max, min and avg HueLightBulbBrightness"
when
        //Time cron "0 0 0 * * ?"
        //System started or
        Time cron "0 0/5 * * * ?"
        //Time cron "0 0/4 * 1/1 * ? *"
then
  val max5HueLightBulbBrightness = (HueLightBulbBrightness.maximumSince(now.minusMinutes(5), "rrd4j").state as DecimalType).doubleValue
  val min5HueLightBulbBrightness = (HueLightBulbBrightness.minimumSince(now.minusMinutes(5), "rrd4j").state as DecimalType).doubleValue
  val avg5HueLightBulbBrightness = (HueLightBulbBrightness.averageSince(now.minusMinutes(5), "rrd4j") as DecimalType).doubleValue

  if( max5HueLightBulbBrightness != null && min5HueLightBulbBrightness != null && avg5HueLightBulbBrightness !=null) {
          HueLightBulbBrightnessAggMax.postUpdate(max5HueLightBulbBrightness)
          HueLightBulbBrightnessAggMin.postUpdate(min5HueLightBulbBrightness)
          HueLightBulbBrightnessAggAvg.postUpdate(avg5HueLightBulbBrightness)
  }
end
```

```
rule "Set 5 minutes max, min and avg AqaraSensorHumidityLivingroom"
when
        //Time cron "0 0 0 * * ?"
        //System started or
        Time cron "0 0/5 * * * ?"
        //Time cron "0 0/4 * 1/1 * ? *"
then
  val max5AqaraSensorHumidityLivingroom =
(AqaraSensorHumidityLivingroom.maximumSince(now.minusMinutes(5), "rrd4j").state as
DecimalType).doubleValue
  val min5AqaraSensorHumidityLivingroom =
(AqaraSensorHumidityLivingroom.minimumSince(now.minusMinutes(5), "rrd4j").state as
DecimalType).doubleValue
  val avg5AqaraSensorHumidityLivingroom =
(AqaraSensorHumidityLivingroom.averageSince(now.minusMinutes(5), "rrd4j") as
DecimalType).doubleValue

  if( max5AqaraSensorHumidityLivingroom != null &&
min5AqaraSensorHumidityLivingroom != null && avg5AqaraSensorHumidityLivingroom
!=null) {

AqaraSensorHumidityLivingroomAggMax.postUpdate(max5AqaraSensorHumidityLivingroo
m)

AqaraSensorHumidityLivingroomAggMin.postUpdate(min5AqaraSensorHumidityLivingroo
m)

AqaraSensorHumidityLivingroomAggAvg.postUpdate(avg5AqaraSensorHumidityLivingroom
)
  }
end

rule "Set 5 minutes max, min and avg AqaraSensorAtmosphericPressureLivingroom"
when
        //Time cron "0 0 0 * * ?"
        //System started or
        Time cron "0 0/5 * * * ?"
        //Time cron "0 0/4 * 1/1 * ? *"
then
  val max5AqaraSensorAtmosphericPressureLivingroom =
(AqaraSensorAtmosphericPressureLivingroom.maximumSince(now.minusMinutes(5),
"rrd4j").state as DecimalType).doubleValue
```

```
  val min5AqaraSensorAtmosphericPressureLivingroom = (AqaraSensorAtmosphericPressureLivingroom.minimumSince(now.minusMinutes(5), "rrd4j").state as DecimalType).doubleValue
  val avg5AqaraSensorAtmosphericPressureLivingroom = (AqaraSensorAtmosphericPressureLivingroom.averageSince(now.minusMinutes(5), "rrd4j") as DecimalType).doubleValue

  if( max5AqaraSensorAtmosphericPressureLivingroom != null && min5AqaraSensorAtmosphericPressureLivingroom != null && avg5AqaraSensorAtmosphericPressureLivingroom !=null) {
AqaraSensorAtmosphericPressureLivingroomAggMax.postUpdate(max5AqaraSensorAtmosphericPressureLivingroom)
AqaraSensorAtmosphericPressureLivingroomAggMin.postUpdate(min5AqaraSensorAtmosphericPressureLivingroom)
AqaraSensorAtmosphericPressureLivingroomAggAvg.postUpdate(avg5AqaraSensorAtmosphericPressureLivingroom)
  }
end

rule "Set 5 minutes max, min and avg AqaraSensorTemperatureLivingroom"
when
	//Time cron "0 0 0 * * ?"
	//System started or
	Time cron "0 0/5 * * * ?"
	//Time cron "0 0/4 * 1/1 * ? *"
then
  val max5AqaraSensorTemperatureLivingroom = (AqaraSensorTemperatureLivingroom.maximumSince(now.minusMinutes(5), "rrd4j").state as DecimalType).doubleValue
  val min5AqaraSensorTemperatureLivingroom = (AqaraSensorTemperatureLivingroom.minimumSince(now.minusMinutes(5), "rrd4j").state as DecimalType).doubleValue
  val avg5AqaraSensorTemperatureLivingroom = (AqaraSensorTemperatureLivingroom.averageSince(now.minusMinutes(5), "rrd4j") as DecimalType).doubleValue

  if( max5AqaraSensorTemperatureLivingroom != null && min5AqaraSensorTemperatureLivingroom != null && avg5AqaraSensorTemperatureLivingroom !=null) {
```

```
AqaraSensorTemperatureLivingroomAggMax.postUpdate(max5AqaraSensorTemperatureLivingroom)

AqaraSensorTemperatureLivingroomAggMin.postUpdate(min5AqaraSensorTemperatureLivingroom)

AqaraSensorTemperatureLivingroomAggAvg.postUpdate(avg5AqaraSensorTemperatureLivingroom)
  }
end

rule "Set 5 minutes max, min and avg AqaraSensorHumidityBedroom"
when
        //Time cron "0 0 0 * * ?"
        //System started or
        Time cron "0 0/5 * * * ?"
        //Time cron "0 0/4 * 1/1 * ? *"
then
  val max5AqaraSensorHumidityBedroom = (AqaraSensorHumidityBedroom.maximumSince(now.minusMinutes(5), "rrd4j").state as DecimalType).doubleValue
  val min5AqaraSensorHumidityBedroom = (AqaraSensorHumidityBedroom.minimumSince(now.minusMinutes(5), "rrd4j").state as DecimalType).doubleValue
  val avg5AqaraSensorHumidityBedroom = (AqaraSensorHumidityBedroom.averageSince(now.minusMinutes(5), "rrd4j") as DecimalType).doubleValue

  if( max5AqaraSensorHumidityBedroom != null && min5AqaraSensorHumidityBedroom != null && avg5AqaraSensorHumidityBedroom !=null) {

AqaraSensorHumidityBedroomAggMax.postUpdate(max5AqaraSensorHumidityBedroom)

AqaraSensorHumidityBedroomAggMin.postUpdate(min5AqaraSensorHumidityBedroom)

AqaraSensorHumidityBedroomAggAvg.postUpdate(avg5AqaraSensorHumidityBedroom)
  }
end

rule "Set 5 minutes max, min and avg AqaraSensorAtmosphericPressureBedroom"
when
        //Time cron "0 0 0 * * ?"
        //System started or
```

```
        Time cron "0 0/5 * * * ?"
        //Time cron "0 0/4 * 1/1 * ? *"
then
  val max5AqaraSensorAtmosphericPressureBedroom =
(AqaraSensorAtmosphericPressureBedroom.maximumSince(now.minusMinutes(5),
"rrd4j").state as DecimalType).doubleValue
  val min5AqaraSensorAtmosphericPressureBedroom =
(AqaraSensorAtmosphericPressureBedroom.minimumSince(now.minusMinutes(5),
"rrd4j").state as DecimalType).doubleValue
  val avg5AqaraSensorAtmosphericPressureBedroom =
(AqaraSensorAtmosphericPressureBedroom.averageSince(now.minusMinutes(5), "rrd4j") as
DecimalType).doubleValue

  if( max5AqaraSensorAtmosphericPressureBedroom != null &&
min5AqaraSensorAtmosphericPressureBedroom != null &&
avg5AqaraSensorAtmosphericPressureBedroom !=null) {

AqaraSensorAtmosphericPressureBedroomAggMax.postUpdate(max5AqaraSensorAtmosphe
ricPressureBedroom)

AqaraSensorAtmosphericPressureBedroomAggMin.postUpdate(min5AqaraSensorAtmosphe
ricPressureBedroom)

AqaraSensorAtmosphericPressureBedroomAggAvg.postUpdate(avg5AqaraSensorAtmospher
icPressureBedroom)
  }
end

rule "Set 5 minutes max, min and avg AqaraSensorTemperatureBedroom"
when
        //Time cron "0 0 0 * * ?"
        //System started or
        Time cron "0 0/5 * * * ?"
        //Time cron "0 0/4 * 1/1 * ? *"
then
  val max5AqaraSensorTemperatureBedroom =
(AqaraSensorTemperatureBedroom.maximumSince(now.minusMinutes(5), "rrd4j").state as
DecimalType).doubleValue
  val min5AqaraSensorTemperatureBedroom =
(AqaraSensorTemperatureBedroom.minimumSince(now.minusMinutes(5), "rrd4j").state as
DecimalType).doubleValue
  val avg5AqaraSensorTemperatureBedroom =
(AqaraSensorTemperatureBedroom.averageSince(now.minusMinutes(5), "rrd4j") as
DecimalType).doubleValue
```

```
  if( max5AqaraSensorTemperatureBedroom != null && min5AqaraSensorTemperatureBedroom != null && avg5AqaraSensorTemperatureBedroom !=null) {
AqaraSensorTemperatureBedroomAggMax.postUpdate(max5AqaraSensorTemperatureBedroom)
AqaraSensorTemperatureBedroomAggMin.postUpdate(min5AqaraSensorTemperatureBedroom)
AqaraSensorTemperatureBedroomAggAvg.postUpdate(avg5AqaraSensorTemperatureBedroom)
  }
end
```

An exemplification of the practical application of the temperature rule (e.g., Set 5 minutes max, min, and avg HueMotionSensorTemperature) within the HomeHub smart ecosystem can be elucidated through its utilization in monitoring temperature measurements obtained from the Hue Motion sensor. Specifically, it serves an important role in the smart home environment by providing insightful analytics on temperature data. This rule operates by computing the maximum, minimum, and average temperature values drawn from the HueMotionSensorTemperature Item, effectively encapsulating the temperature dynamics over the preceding 5-minute window. In doing so, it harnesses the power of data aggregation to derive key insights that can be invaluable for monitoring and trend analyzing. Upon execution, this rule updates corresponding aggregation Items, namely HueMotionSensorTemperatureAggMax, HueMotionSensorTemperatureAggMin, and HueMotionSensorTemperatureAggAvg, with the computed values. This data collection and aggregation process lay the foundation for comprehensive and real-time temperature monitoring within the smart home, offering an indispensable tool for enhancing comfort, energy efficiency, and overall functionality. The collective suite of rules employed within the openHAB environment operates in synergy, each rule focused on a specific data dimension, such as temperature in this case. The orchestration enables researchers to effectively keep track of aggregated sensor data, offering a deep understanding of how various parameters evolve over 5-minute intervals. The resulting analytical capability empowers informed decision making and optimizing the smart home ecosystem for maximum convenience and efficiency.

2.1.6 *New Persistence Service*

A new persistence service has been established for the openHAB rules to function correctly for longer periods of time. This service is crucial as it enables the calculation and storage of historical data, similar to what is achieved with the developed aggregation rules. The role of persistence services in openHAB is to provide the means for storing and retrieving data from Items. This functionality is essential for creating charts, graphs, and conducting analysis of smart home data. OpenHAB offers various common persistence services to choose from, including RRD4j[9],

[9] https://www.openhab.org/addons/persistence/rrd4j/

InfluxDB[10], and MariaDB[11], among others. In the previous openHAB version, the RRD4j service was employed for this purpose. In the current and final version of HomeHub, MariaDB is employed.

Table 9: Persistence strategy configuration.

```
Strategies {

   everyMinute : "0 * * * * ?"
}

Items {
   // persist items on every change and every minute
   * : strategy = everyChange, everyMinute, restoreOnStartup
   //* : strategy = everyChange, restoreOnStartup
}
```

The provided persistence configuration encompasses a set of strategies and settings that serve specific purposes within our openHAB system:

- In the **Strategies** Section:
  - \`*everyMinute*\`: This strategy employs a cron expression, designed to initiate persistence updates at the onset of each minute, precisely when the seconds component reaches 0. This periodicity ensures that data is consistently persisted at one-minute intervals.
- In the **Items** Section:
  - \`*\* : strategy = everyChange, everyMinute, restoreOnStartup*\`: This configuration is designed to apply two primary strategies, namely "everyChange" and "everyMinute," to all items within the system. The "everyChange" strategy triggers persistence whenever there is a change in an Item's state, ensuring that any alteration is recorded. The "everyMinute" strategy complements this by persisting Item data at one-minute intervals, regardless of whether a change has occurred. Additionally, the configuration includes \`restoreOnStartup\`, a feature that guarantees the restoration of each Item's state to its most recently persisted value when openHAB initializes.

This comprehensive persistence strategy effectively results in the systematic persistence of data for all items within the openHAB system. It ensures that changes in Item states are reliably captured (\`everyChange\`), and data is periodically persisted at one-minute intervals (\`everyMinute\`). Furthermore, the inclusion of \`restoreOnStartup\` ensures the integrity of item states by restoring them to their last known values during openHAB system startup.

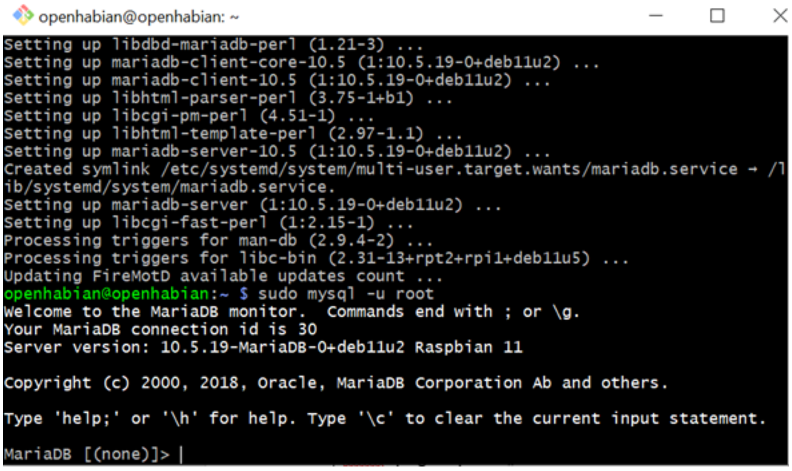
Following this specific strategy, the persistence service has been transitioned to MariaDB, a widely favoured choice for data storage, especially beneficial for long-term data retention and handling complex queries. The following steps outline the configuration and setup procedures

[10] https://www.influxdata.com/

[11] https://mariadb.org/

that were undertaken to ensure the proper functioning of the MariaDB persistence service within the openHAB environment:

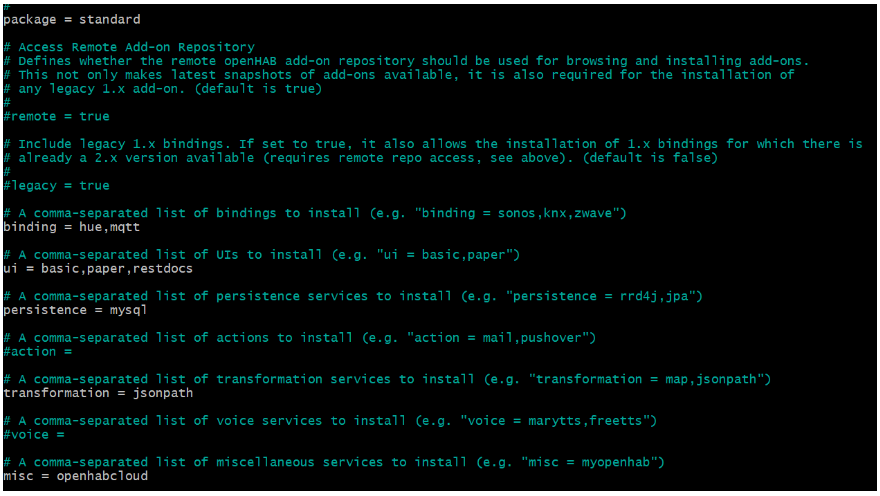
1. Install and Configure MariaDB: The initial step involved the installation and configuration of MariaDB on the system (Figure 8). This was a foundational prerequisite to have MariaDB up and running, ready to handle the storage and retrieval of smart home data.



```
openhabian@openhabian: ~
Setting up libdbd-mariadb-perl (1.21-3) ...
Setting up mariadb-client-core-10.5 (1:10.5.19-0+deb11u2) ...
Setting up mariadb-client-10.5 (1:10.5.19-0+deb11u2) ...
Setting up libhtml-parser-perl (3.75-1+b1) ...
Setting up libcgi-pm-perl (4.51-1) ...
Setting up libhtml-template-perl (2.97-1.1) ...
Setting up mariadb-server-10.5 (1:10.5.19-0+deb11u2) ...
Created symlink /etc/systemd/system/multi-user.target.wants/mariadb.service → /l
ib/systemd/system/mariadb.service.
Setting up mariadb-server (1:10.5.19-0+deb11u2) ...
Setting up libcgi-fast-perl (1:2.15-1) ...
Processing triggers for man-db (2.9.4-2) ...
Processing triggers for libc-bin (2.31-13+rpt2+rpi1+deb11u5) ...
Updating FireMotD available updates count ...
openhabian@openhabian:~ $ sudo mysql -u root
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 30
Server version: 10.5.19-MariaDB-0+deb11u2 Raspbian 11

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> |
```

Figure 8: Installation and configuration of MariaDB.

2. MariaDB Persistence Binding in openHAB: Subsequently, the installation and configuration of the MariaDB Persistence binding within the openHAB setup were carried out. To facilitate this integration, necessary modifications were made to the addons.cfg file (Figure 9). This adjustment was crucial in enabling openHAB to seamlessly interact with the MariaDB database, ensuring the smooth operation of the persistence service.



```
#
package = standard

# Access Remote Add-on Repository
# Defines whether the remote openHAB add-on repository should be used for browsing and installing add-ons.
# This not only makes latest snapshots of add-ons available, it is also required for the installation of
# any legacy 1.x add-on. (default is true)
#
#remote = true

# Include legacy 1.x bindings. If set to true, it also allows the installation of 1.x bindings for which there is
# already a 2.x version available (requires remote repo access, see above). (default is false)
#
#legacy = true

# A comma-separated list of bindings to install (e.g. "binding = sonos,knx,zwave")
binding = hue,mqtt

# A comma-separated list of UIs to install (e.g. "ui = basic,paper")
ui = basic,paper,restdocs

# A comma-separated list of persistence services to install (e.g. "persistence = rrd4j,jpa")
persistence = mysql

# A comma-separated list of actions to install (e.g. "action = mail,pushover")
#action =

# A comma-separated list of transformation services to install (e.g. "transformation = map,jsonpath")
transformation = jsonpath

# A comma-separated list of voice services to install (e.g. "voice = marytts,freetts")
#voice =

# A comma-separated list of miscellaneous services to install (e.g. "misc = myopenhab")
misc = openhabcloud
```

Figure 9: Updated addons.cfg file.

3. Next, a configuration is set at the `services/mariadb.cfg` file as it is depicted in Table 10.

Table 10: MariaDB persistence service configuration.

the database url like 'jdbc:mysql://<host>:<port>/<database>' (without quotes) url=jdbc:mysql://127.0.0.1:3306/OpenHAB # the database user

```
user=openhab

# the database password
password=openhabian

# the reconnection counter
reconnectCnt=3

#sqltype.STRING  =  TEXT
mysql:sqltype.string=VARCHAR(20000)

# the connection timeout (in seconds)
#waitTimeout=

# Use MySQL Server time to store item values (=false) or use openHAB Server time (=true).
# For new installations, its recommend to set "localtime=true".
# (optional, defaults to false)
localtime=true
```

4. Next, a strategy is set at `mysql.persist` as it is depicted in Table 11.

Table 11: MariaDB persistence service strategy.

```
Strategies {
 default = everyUpdate
}

Items { *: strategy = everyUpdate }
```

5. Assess Database Connection: openHAB can successfully connect to the MariaDB database. To ensure this, the openHAB logs were assessed and by using database management tools, the verification of that data being stored occurred.
6. Data Retrieval: Once the data is being persisted, the REST API may be used to retrieve it.

By following these steps, MariaDB can be successfully integrated as the chosen persistence service, enhancing data storage capabilities for long-term retention, and supporting complex queries. This transition allows for the utilization of MariaDB's benefits in the effective management of smart home data within the openHAB ecosystem.

2.1.7 *Minimizing IMG file size*

PiShrink[12] is a bash script that serves as a valuable utility for reducing the IMG file size on Raspberry Pi devices. Employing PiShrink in this context proves advantageous, particularly when the objective is to economize storage space on the SD card or when creating backups of the Raspberry Pi's system.

[12] https://github.com/Drewsif/PiShrink

The deployment of PiShrink entails a structured sequence of steps:

1. *Create a Backup*: As a preliminary measure, it is imperative to generate a backup of the Raspberry Pi's SD card. This safeguard ensures the availability of a functional system copy in the event of any complications during the image size reduction process.
2. *Insert the SD Card*: The Raspberry Pi's SD card is to be inserted into a card reader and subsequently connected to the host computer.
3. *Install PiShrink (If Not Already Installed)*: In the event that PiShrink is not pre-installed on the host computer, it can be conveniently downloaded and installed. Notably, PiShrink typically manifests itself as a shell script, making integration with the system a straightforward task. Adequate execution permissions should be granted as a prerequisite.
4. *Execute PiShrink*: The next step involves the execution of PiShrink through a terminal window. The command should include the path to the target image file. An example command is depicted in Table 12.

Table 12: Example of executing PiShrink through terminal.

```
sudo pishrink.sh /path/to/your-image.img
```

5. *Await the Shrinking Process*: PiShrink will commence the analysis of the image, with the primary objective of eliminating superfluous space. The duration of this process is contingent upon variables such as the SD card's capacity and the volume of data contained therein, potentially requiring a considerable amount of time.
6. *Inspect the Shrunk Image*: Upon the successful completion of the image reduction process, you will be left with a diminished image file, which can be effectively utilized by technicians or for any other intended purpose.

In summary, PiShrink stands as a valuable tool for efficiently managing Raspberry Pi images. Its utility is particularly evident when striving to optimize storage space or work with more compact backup files, as exemplified in the use case of SMART BEAR pilots.

2.1.8 *Cloud Kubernetes Integration*

The openHAB Cloud[13] is an open-source backend service that is used in the SMART BEAR home automation platform. A deployment was created for deploying it on Kubernetes aiming to provide scalability and manageability benefits. For the deployment the following were used:

a) A Kubernetes Cluster (initially configures by ATOS and is currently maintained by ICCS) that ensure a running Kubernetes environment where the Kubernetes services are managed (e.x. Google Kubernetes Engine (GKE), Amazon EKS, or Minikube for local development). Minikube[14] used for building the manifest file in our local development environment.

b) A kubectl[15] toolset that is used to configure command-line tools to interact with your Kubernetes cluster.

The steps to create and integrate the Kubernetes deployment manifest (a Kubernetes specific syntax file) are the following:

[13] https://github.com/openhab/openhab-cloud

[14] https://minikube.sigs.k8s.io/docs/

[15] https://kubernetes.io/docs/reference/kubectl/

- Create Kubernetes deployment manifest: A deployment manifest is the file named "openhab-deployment.yaml" for the SMART BEAR project while "yaml" is the specification language (*YAML is a human-friendly data serialization language for all programming languages*)[16] used for writing the manifest file. Our deployment provides declarative updates for the pods which are the smallest deployable units of computing; that are created and managed in Kubernetes. The SMART BEAR deployment manifest is a declarative specification that was built specifically for the projects' needs based on the Kubernetes' cluster resources developed by ATOS. The resources are including pods, services, deployments, replica sets, and more. The Kubernetes manifest file usually contains configurations and settings for one or multiple Kubernetes resources.
- Apply the Deployment Manifest: The deployment manifest is applied with the command: kubectl apply -f openhab-deployment.yaml
- Expose the Service: The services (web access) is exposed and enabled with the command: kubectl apply -f openhab-service.yam.
- Access the Service: The openHAN services are finally accessible to the users by applying the command: kubectl get services openhab-service
- Configure openHAB: Connect to the openHAB interface and configure it according to the requirements of the project while the selected smart devices and services are configured. The deployment is furthermore customized with persistent storage and SSL/TLS certificates.

The Kubernetes installation of the SMART BEAR openHAB cloud service requires a set of systems and platforms that all are pre-configured and installed by the manifest deployment profile. The following pre-requirements are developed and started before the deployment of the openHAB cloud service: a)MongoDB noSQL database server should be up an running, b) REDIS cache server should be up and running with a selected password configured. The value of the parameter defined in CONFIG_JSON_FILE should be mounted as "config.json" file in the path "/config" to avoid overwrite the complete folder and is copied to "/opt/openhabcloud" with a deployment descriptor before executing the openHAB environment.

The overall deployment configuration is based on 4 YAML manifest files and they are including different kinds of Kubernetes configuration profiles including: ConfigMap, PersistentVolumeClaim, Deployment, Service and Ingress. The main configuration scheme is presented in the table below and it is based on providing an config.json.template and mount it into the folder /data/openhabcloud-config:

Table 13: The openHAB cloud configuration scheme.

```
- Secret ```smartbear-openhabcloud-database-secret```, with the following variables are defined,
e.g.:
   * ```MONGO_INITDB_ROOT_PASSWORD```: The admin password for the MongoDB
   * ```MONGO_INITDB_PASSWORD```: The custom smartbear password for the MongoDB
   * ```REDIS_PASSWORD```: The REDIS password

 - Secret ```smartbear-openhabcloud-secret```, with the following variables are defined, e.g.:
   * ```EXPRESS_KEY```: The Express key
   * ```MONGO_INITDB_PASSWORD```: The custom smartbear password for the MongoDB
```

16 https://yaml.org/

```
   * ```REDIS_PASSWORD```: The REDIS password
   * ```MAILER_PASSWORD```: The MAILER service password

- ConfigMap ```smartbear-openhabcloud-config```, with the following variables are defined, e.g.:
   * ```DOMAIN_NAME```: openhab.smart-bear.eu
   * ```DOMAIN_PORT```: 443
   * ```MONGODB_HOSTS```: smartbear-openhabcloud-database-svc:27017
   * ```REDIS_HOST```: smartbear-openhabcloud-database-redis-svc
   * ```REDIS_HOST_PORT```: '6379'
   * ```CONFIG_JSON_FILE```: containing the content of the config.json.template template file, e.g.:
      ```
      {
          "system": {
              "host": "$DOMAIN_NAME",
              "port": "$DOMAIN_PORT",
              "protocol": "https",
              "logger" : {
                  "type": "console"
              },
              "subDomainCookies": false
          },
          "express":{
              "key" : "$EXPRESS_KEY"
          },
          "apn" : {
              "gateway": "gateway.push.apple.com",
              "cert": "certs/aps/aps_production_cert.pem",
              "key": "certs/aps/aps_production_key.pem",
              "passphrase": "passphrase"
          },
          "gcm" : {
              "jid": "something@gcm.googleapis.com",
              "password": "password"
          },
          "ifttt" : {
              "iftttChannelKey": "key",
              "iftttTestToken": "token"
          },
          "mongodb": {
              "user": "$MONGO_INITDB_USERNAME",
              "password": "$MONGO_INITDB_PASSWORD",
              "hosts": ["$MONGODB_HOSTS"],
              "db": "$MONGO_INITDB_DATABASE"
          },
          "redis": {
              "host": "$REDIS_HOST",
              "port": "6379",
              "password": "$REDIS_PASSWORD"
          },
          "mailer": {
              "host" : "mail.your-server.de",
              "port": 587,
              "secureConnection": true,
              "user": "sbhab@precious-cloud.eu",
              "password": "$MAILER_PASSWORD",
              "from": "openHAB Cloud <sbhab@precious-cloud.eu>"
          },
          "legal": {
              "terms": "/terms",
              "policy": "/policy"
          },
```
```

```
            "registration_enabled": true
        }
        ```
```

## 2.2 *HomeHub Validation*

### 2.2.1 *Testing with 2nd device set*

Additional validation of the proposed technique, commencing with the previous delivery following steps, was required to ensure its correctness and reliability. To achieve this, the generated image was utilized as a reference, along with the second set of devices (Figure 10), which included sensors, to assure that all components were correctly recognized and utilized. This aided in ensuring that no steps were overlooked throughout the installation procedure, since the HomeHub technical handbook was attentively followed. As a result, the established approach was completely validated, and its usefulness was demonstrated by carrying out these tests. As such, the installation team can be confident that everything will be completed correctly and swiftly.

*Figure 10: The two sets of devices.*

The steps undertaken are extensively described in the HomeHub Technical Manual, and the essential time points are presented here to demonstrate that each module is operating properly, resulting in functional HomeHub assembly.

1. The new set of devices are successfully installed with the preconfigured OpenHab setup by following the specified protocol for device recognition. This step ensures that the system can properly recognize and communicate with each device. This is achieved through the incoming messages located at the "Inbox" tab on http://openhabian:8080/paperui/index.html, which showcase that Philips devices are recognized (Figure 11, Figure 12), while Zigbee2mqtt will start searching for new devices
```

automatically and when it detects the Aqara sensors, it will display information about the devices in the web interface at http://openhabian:9090/.

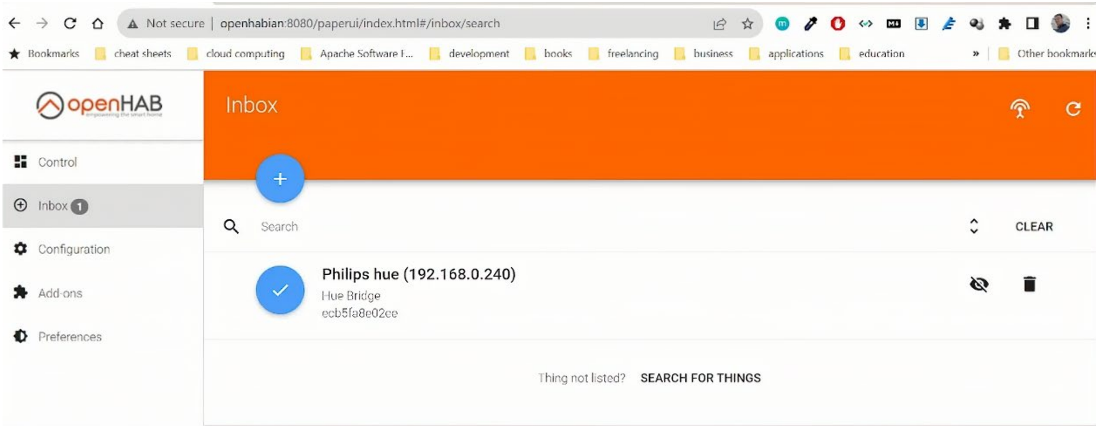


Figure 11: Philips hue bridge device recognition.

2. All measurements from the devices are now displayed on the OpenHab User Interface **locally**, providing real-time monitoring of the system. This step enables personnel to monitor the system and detect any abnormalities or issues, which is achieved through "Control" tab found at http://openhabian:8080/paperui/index.html.
3. The registration of the device set was successfully conducted to the OpenHab Cloud, allowing remote access to the system's data. This step provides personnel with the ability to access the data of the system from anywhere and at any time, increasing the system's accessibility and ease of use. This is achieved through assessing the measurements online located at https://openhabcloud.preciouscloud.eu:8443/.
4. The measurements are now displayed on the cloud, and their validity is verified by their proximity in space (i.e., verifiable through environmental conditions of the room) and time (i.e., verifiable through UNIX time values), ensuring accurate and reliable data. This step ensures that the system's data is reliable and can be used to make informed decisions. This objective can be accomplished through the utilization of external devices, such as thermometers.

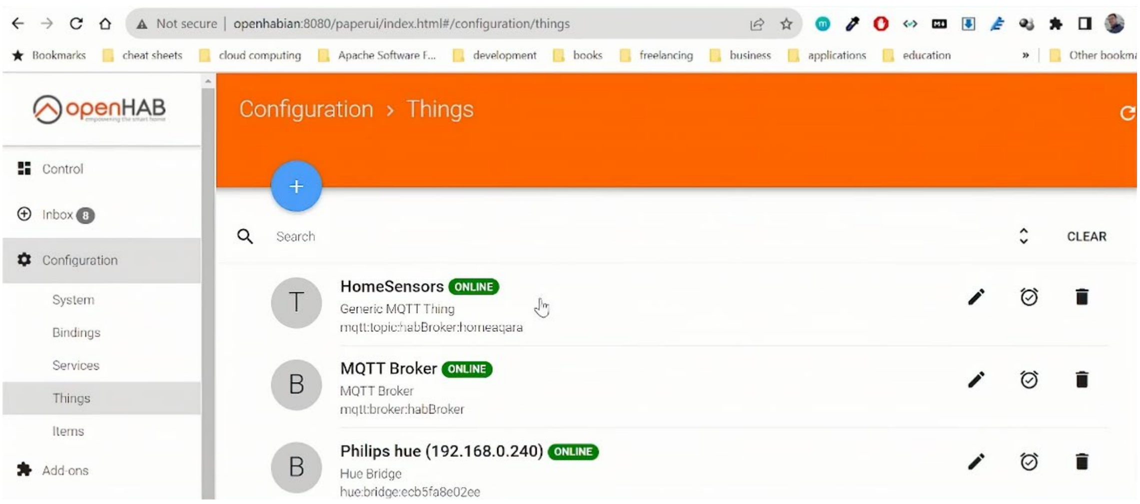


Figure 12: All Philips hue devices recognition.

5. In addition to the individual measurements, aggregated measurements are also displayed, providing valuable insights into the environmental conditions of the home as well as overall performance of the system. This step provides a comprehensive overview of the system's performance, enabling personnel to identify trends and patterns in the home environment.
6. These features can be viewed through the SMART BEAR Mobile Application, which includes screens for monitoring light levels, motion, and various environmental measurements such as humidity, temperature, and pressure. By utilizing these screens, end-users can actively participate in the HomeHub ecosystem by adjusting room lighting, checking the last detected motion, and staying informed about their home's current environmental conditions. Previous measurements can be assessed through the history tab.

By completing these specific smaller tasks, one can ensure that the system is functioning properly, and all individual modules are working as intended.

2.3 *Addressing D3.4 Next Steps*

In the pursuit of advancing the HomeHub capabilities, several critical objectives have been previously identified to bolster its functionality and efficiency. Firstly, a comprehensive validation process had to be executed, which involves extending the testing scope to encompass an additional set of devices and sensors. This expansion, coupled with the previously tested second Raspberry Pi, assessed the method's robustness and reliability. Furthermore, a vital step in enhancing the project overall objectives involves migrating the OpenCloud service from its current deployment on the University of Ioannina's Cloud Infrastructure to the SMART BEAR@Cloud Infrastructure. This transition enables HomeHub to leverage the advanced cloud components and resources provided by SMART BEAR@Cloud, thus elevating its performance and scalability.

Another essential aspect of HomeHub's future development is simplifying the technician experience during installation. To achieve this, a transition away from using hand-written commands for retrieving UUIDs and secret keys will be undertaken. Instead, the aim is to implement user-friendly one-click buttons, automated command prompt scripts, streamlining the cloud registration process and making it more accessible to technical employees. Furthermore, efforts are directed towards enhancing the configuration file (e.g., cfg) to enable the automatic registration of Aqara device IDs. By simplifying these procedures, HomeHub seeks to increase its user-friendliness and accessibility while maintaining its robust functionality. Additionally, the focus also changes to standardizing variable names to facilitate seamless collaboration with partner organizations. This standardization, as outlined in the cross-partner document titled "SB_Devices_Measurements_v6.docx," and updated to v7[17] and supplementary excel file[18] for the HomeHub measurements aims to establish universal conventions that facilitate data sharing and cooperation across all project stakeholders. These multifaceted initiatives collectively represent HomeHub's commitment to continuous improvement and innovation in the field of smart home technology. Lastly, the steps as provided above are aggregated into Table 14, along with the methodology of their implementation and the corresponding section where the reader can refer to.

[17] https://docs.google.com/document/d/1sP8ygQJeDmyyFCVgMvbKuptR6kX2lajR/

[18] https://docs.google.com/spreadsheets/d/1sPxTnimDAhZ3nzxRazS8kMVixa4d5Z6J

Table 14: Next steps, implementation methodology, and corresponding cross-reference.

Next Step	Implementation methodology	Cross-Reference
Testing a second set of sensors in addition to second Raspberry Pi processing unit.	Utilize a second set of sensors and raspberry with the preconfigured software.	2.2.1
Migration to Kubernetes	The Kubernetes installation of the SMART BEAR openHAB cloud service requires a set of systems and platforms that all are pre-configured and installed by the manifest deployment profile.	2.1.8
Avoid using CMD commands to retrieve UUID and secret keys to simplify the cloud registration process.	Bash script "ohc" was created to automatically retrieve UUID and secret key pair.	2.1.3
Simplify the Aqara device ID identification in order for it to be automatically registered in the configuration file.	A web-based interface was generated to automatically identify and register the Aqara devices.	2.1.2
Provide universal variable names for cross-partner facilitation.	Variable names were generated to facilitate not only in the cross-partner collaboration but also streamlines data migration schemas.	2.1.4

2.4 *Supporting Material for the Pilots*

In pursuit of further augmenting assistance for prospective pilots during the pre-recruitment phase, an elaborate video presentation was meticulously crafted. This video serves as a demonstrative showcase, meticulously illustrating the multifaceted capabilities inherent to diverse sensor technologies integrated within the SMART BEAR solution, encompassing lighting sensing, humidity and temperature sensing, and motion sensing devices. Furthermore, the presentation encapsulates the landscape of contemporary smart home ecosystems throughout smart devices and sensors. This collaborative endeavor was orchestrated through the concerted efforts of WP3 partners, each bringing their specialized expertise to the table, and coordinated under the guidance of UOI. Such collaboration underscores the commitment to providing pilots with state-of-the-art tools and knowledge requisite for excellence within the home automation domain.

To meet the project objectives with ease, a series of instructional videos has been synthesized in addition. These videos serve an important role in elucidating installation process, which has been systematically deconstructed into smaller, more comprehensible tasks. This strategic breakdown not only facilitates a clearer understanding of the complex installation procedures undertaken by the technical personnel but also enhances their proficiency in executing these tasks with precision and efficacy. Remarkably, these videos offer an authentic and real-time perspective, as they were filmed during the actual installation process, providing an unfiltered view of the installation journey, including potential challenges that may arise.

Specifically, the installation process encompasses several distinct yet interrelated phases. These encompass the initial step of burning the openHAB image, followed by the crucial processes of device pairing, cloud registration, and mobile application validation. It is imperative to underscore that these phases are intricately intertwined, forming a cohesive workflow that demands attention. These video resources serve as invaluable tools, offering technicians a comprehensive and dynamic insight into the HomeHub installation process, ultimately enhancing their competence and efficacy in this critical endeavor.

In conclusion, the manual pertaining to the HomeHub installation process have undergone a thorough revision to accurately incorporate the most recent modifications. These updates ensure that the manual align precisely with the current installation procedures, providing technicians with up-to-date guidance for HomeHub deployment.

2.5 *Personal Devices Linked (not integrated to the HomeHub)*

Within the context of WP3, the SMART BEAR project encompasses the provisioning of technology and devices, catering to the holistic needs of the project. It's important to note that this chapter includes an updated catalog of devices integrated for use by study participants, even though they may not have a direct association with home automation. Comprehensive insights into the detailed measurements, data types, and fields pertinent to all devices are meticulously presented in Chapter 4.

Of relevance, Table 15 provides an overview of the carefully selected personal health-related tracking devices that are instrumental to the project. These devices are seamlessly integrated into the SMART BEAR ecosystem, with communication facilitated through their respective APIs. The underlying mechanisms for interaction with these devices are elaborated upon in Chapter 3, outlining the intricate interplay between the SMART BEAR mobile application and these essential health tracking tools.

Table 15: The hardware components of HomeHub, along with the medical devices.

No.	Device Selected	Brief Description
1	Garmin Venu SQ Smart Watch	Track activity and location
2	Withings Body+	Measure body related characteristics (weight, bmi, etc)
3	Withings BPM Core	Make a quick Cardio Check-up including ECG
4	Beurer Thermometer	Measure Body Temperature
5	iHealth Smart Air Plus	Measure SPO_2
6	OTICON miniRITE T	Hearing Aids
7	iHealth Thermo Pro	Measure Body Temperature
8	iHealth Scale Fit	Measure body related characteristics (weight, bmi, etc)
9	OMRON M7	Blood pressure monitor

3 SMART BEAR Mobile Application (Delta – Updates)

The SMART BEAR mobile application (SB@App), represents a crucial component designed to furnish study participants with an intuitive and user-friendly graphical interface. Its core functionalities encompass managing communication between medical devices, the HomeHub, and the SB platform, all while accessing various platform utilities and functionalities. The application also plays a pivotal role in formatting transmitted data to comply with the Fast Healthcare Interoperability Resources (FHIR) format. Furthermore, it delivers pertinent data to participants and facilitates the seamless integration of additional apps, such as the Diet App, Medication App, and Rehabilitation App.

Significant enhancements and updates have been undertaken during this development iteration, elevating the SB@App's capabilities further. These updates include the expansion of the user interface to accommodate the incorporation of a second humidity and temperature sensor in an additional room, the implementation of a convenient feature for resetting iHealth scales, the establishment of a comprehensive data gathering protocol within a 2-hour window, and the development of two cognitive games for an enriched user experience. Interoperability efforts have also seen advancements, with the finalization of Hearing Aids data transfers and Beurer devices data formatting in compliance with the FHIR standard.

Moreover, the application's compatibility with the evolving installation environment, component installation sequences, and other parameters has been ensured. The integration of three new Beurer devices, namely the oximeter, thermometer, and scale, has been seamlessly incorporated into the SMART BEAR solution, alongside necessary issues resolve for previously utilized devices, such as Withings/iHealth devices. The mobile application's evolution places a strong emphasis on user interface, functionality, and user experience, manifesting in features like the display of sensor readings from the second room on the "Home Comfort" screen, the introduction of cognitive games, and an upgrade of authorization services for enhanced security and scalability through JWT tokens.

Furthermore, all needed health-related data is now transformed into the FHIR format, ensuring secure transmission to the SMART BEAR platform through the upgraded mobile application, which has transitioned from Android level 12 to level 13. This holistic approach ensures that changes in devices, data transmission pipelines, registration processes, and all related information are accurately reflected in the revised and enriched user manuals.

3.1 *SB@App Deployment*

3.1.1 *Technology Stack*

The Android API level supported by SB@App has been elevated to Android 13, aligning with the recommendation provided by the Android Integrated Development Environment (IDE) for application development and Google Play Store standards. This enhancement comes with a commitment to maintaining backward compatibility, ensuring that the application remains accessible to a wide range of Android devices.

The foundational architecture of the two backend components relied upon by SB@App remains consistent with the structure presented in previous deliverables. Previously, these components were deployed separately, with the Android backend hosted on Heroku and the Notifications mechanism on virtual machines (VMs). However, a significant milestone has been achieved through the seamless migration of these components to the Kubernetes platform. This migration signifies a substantial advancement in terms of deployment efficiency and scalability capabilities.

The migration process encompasses three distinct environments: development, staging, and production. This approach guarantees uniform performance across these environments, thereby assuring a dependable and cohesive experience for both users and stakeholders. Throughout this migration effort, the requisite configuration files and resources were prepared to facilitate the smooth deployment of these components within the Kubernetes infrastructure. This strategic groundwork minimizes disruptions during the transition and optimizes the efficiency of the deployment process. Rigorous testing has been conducted across all environments to ensure the seamless operation of the migrated components within the Kubernetes framework, meeting the stringent performance standards expected. Furthermore, proactive performance monitoring has been implemented to enable continuous optimization, ensuring that these components consistently operate at peak efficiency in their new environment.

3.1.2 *Mobile App Delivery*

The SB@App is readily accessible to the public through the Google Play Store, provided as an open testing release. With each new release of the SMART BEAR mobile application on the Google Play Store, a streamlined process is in place for the **automatic** updating of the application on user devices. This process involves pre-configured settings within the Google Play Store application itself, which autonomously manages the update procedure. Furthermore, the Android operating system on the user's devices collaborates seamlessly with the Google Play Store application, ensuring the efficient and hassle-free deployment of updates to the SMART BEAR application. This approach to updates guarantees that users have access to the latest features and enhancements without manual intervention, enhancing the overall user experience.

3.2 *Device Integration*

Within the current phase of mobile application development, notable strides have been taken to incorporate a diverse array of devices into the SMART BEAR solution. This expansion encompasses the integration of various categories of devices, including smart home devices and sensors, medical devices sourced from different vendors, and smart wearables. This integration serves a dual purpose: firstly, it aims to diversify the SMART BEAR ecosystem, offering a broader spectrum of options to end-users (i.e., both PoP and Pilot users), and secondly, it serves as a rigorous test of the interoperability and compatibility of the existing solution with devices from different manufacturers. By seamlessly connecting with devices from various vendors, SMART BEAR is poised to accommodate a wider and more diverse audience in the future, further enhancing its accessibility and utility within the broader ecosystem.

3.2.1 *HomeHub Devices*

3.2.1.1 HomeHub Integration

Following the verification of the HomeHub user's credentials, a secure and authenticated communication channel is established between the SB@App and the HomeHub component. This communication serves as the conduit for retrieving comprehensive data generated by various home sensors. Furthermore, SB@App provides an intuitive feature enabling users to control the brightness of connected smart light bulbs, enhancing their ability to customize their home environment.

To ensure the consistent flow of data, SB@App implements a mechanism whereby it periodically requests aggregated sensor data from the HomeHub using the REST API at fixed intervals of two hours. In the event of an unsuccessful request, the application is programmed to automatically retry the data retrieval process two hours later. This retrial ensures the retrieval of data pertaining to the previous timeframe, thereby mitigating the risk of data loss, and maintaining a continuous stream of information for users. This approach underscores the project's commitment to data reliability and user satisfaction.

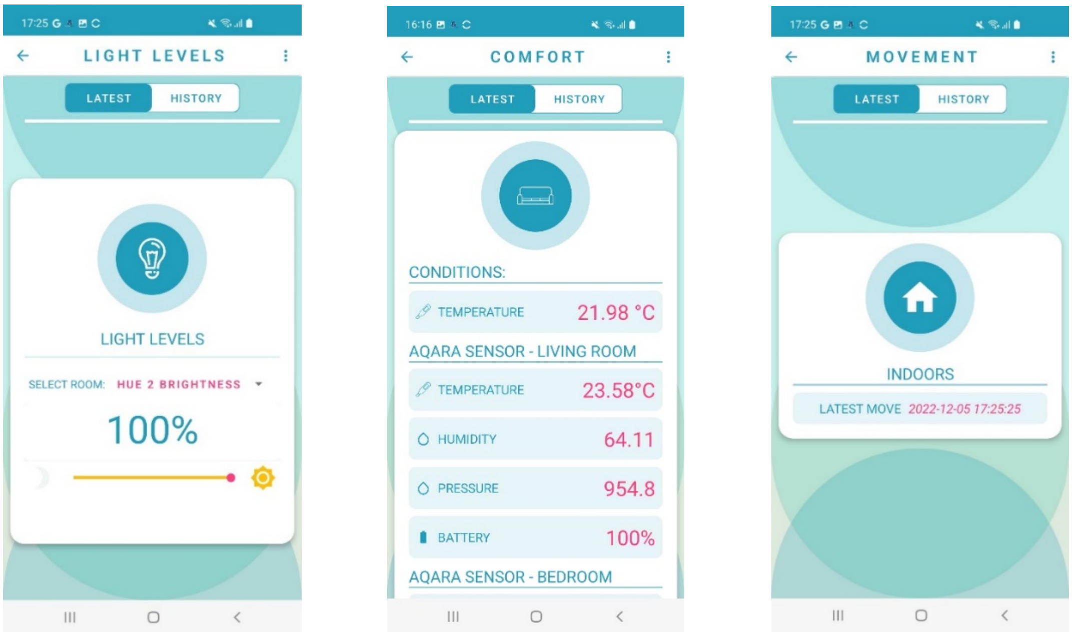


Figure 13: a) Light levels; b) comfort; c) movement screens.

3.2.1.2 HomeHub Screens

The retrieved data is presented through a series of user-friendly screens, each designed to provide specific insights. The "Light Levels" screen offers a comprehensive view of the light levels within various rooms inside the participants' homes. Meanwhile, the "Comfort" screen displays temperature and humidity measurements from sensors positioned in both the living room and bedroom, allowing users to monitor the environmental conditions effectively. In contrast, the "Movement" screen is dedicated to showcasing the most recent changes recorded by door sensors within the user's residence, serving as a valuable security and activity monitoring feature (Figure 13). Importantly, all these screens are accompanied by corresponding history lists, affording users the ability to review and analyze previous measurements, thereby facilitating informed decisions and enhancing the overall user experience (Figure 14).

Figure 14: a) Light levels; b) comfort; c) movement history screens.

3.2.2 *Wearable Devices*

3.2.2.1 Smart Watch Integration

The Garmin Venu SQ wearable plays an essential role in tracking and summarizing user activities on a daily basis. This data is subsequently transmitted to the corresponding vendor application, installed and operating on the user's mobile device through a Bluetooth connection. Once within the vendor app, the data is synchronized with the vendor's cloud platform, specifically the Garmin Connect Platform, whenever an internet connection is available. To facilitate the integration of this data into the SMART BEAR ecosystem, the SB@App leverages the HEALTH API[19] and ACTIVITY API[20] provided by Garmin (Figure 15).

It is worth noting that the integration and functionality of the HEALTH API and ACTIVITY API remain consistent with previous deliverables, providing a seamless and reliable means of data retrieval.

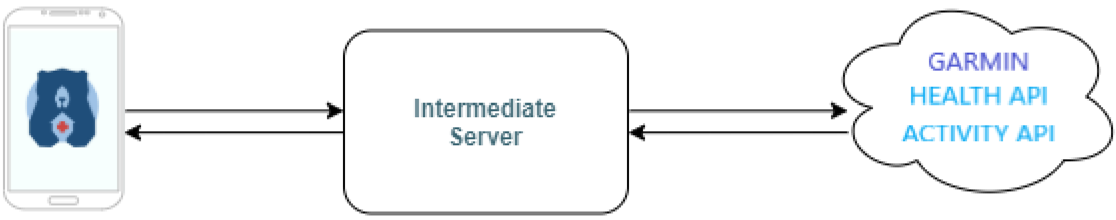


Figure 15: Process of retrieving Garmin wearable data.

[19] https://developer.garmin.com/gc-developer-program/health-api/

[20] https://developer.garmin.com/gc-developer-program/activity-api/

The HEALTH API and ACTIVITY API deliver an array of user measurement data, encompassing daily summaries, sleep summaries, stress levels, respiration, oxygen saturation, and additional activity summary data.

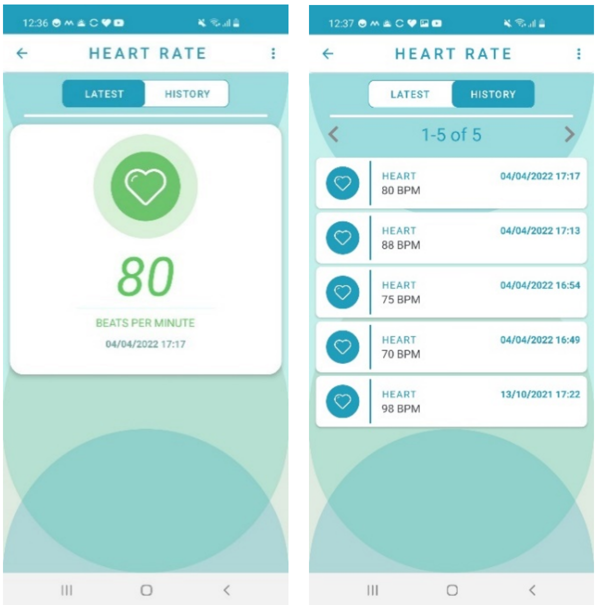


Figure 16: Heart Rate data display Screens.

These values are stored locally within the SB@App, and specific data points including:

- maxHeartRateInBeatsPerMinute
- averageHeartRateInBeatsPerMinute
- minHeartRateInBeatsPerMinute
- steps
- distanceInMeters
- durationInSeconds
- deepSleepDurationInSeconds
- lightSleepDurationInSeconds

are transmitted to the SB@Cloud in the standardized FHIR format. Additionally, all GARMIN data values are transmitted to the SB@Cloud in a non-FHIR repository, ensuring redundancy and comprehensive data storage for further analysis and utilization within the SMART BEAR platform.

3.2.2.2 Smart Watch Screens

The SMART BEAR application features a dedicated "**Heart Rate**" screen (Figure 16), which offers users access to the most recent heart rate measurement synced from the daily summary data. Furthermore, the "Heart Rate" screen provides a comprehensive history of all recorded heart rate measurements, empowering users to review their historical data.

Similarly, the "**Activity**" screen provides users with a detailed overview of their daily physical activity, including step counts and distance walked (Figure 17). This screen not only displays current activity data but also maintains a comprehensive historical record of these measurements, enabling users to track their physical activity trends over time.

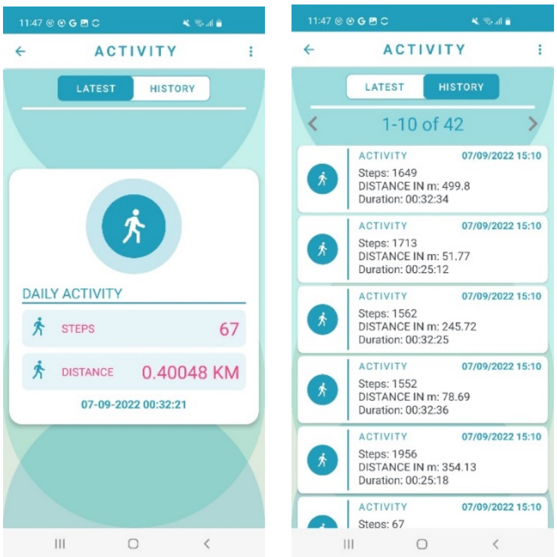


Figure 17: Activity Screens.

Moreover, the "**Sleep Statistics**" screen furnishes users with insights into their sleep patterns by showcasing the latest sync measurement derived from daily sleep summary data (Figure 18). Users can readily access information regarding their most recent sleep duration, and a comprehensive history of all sleep-related measurements is also made available. This feature facilitates a deeper understanding of one's sleep habits and patterns, promoting overall health and well-being awareness.

Figure 18: Sleep Statistics Screens.

3.2.3 *Medical Devices*

3.2.3.1 Beurer Thermometer, Oximeter, and Scales Integration

The SMART BEAR application has expanded its device support to include three new additions: the Beurer non-contact thermometer FT 95, the Beurer BF 600 scale, and the Beurer PO 60 pulse oximeter (Figure 19). These devices utilize Bluetooth GATT (Generic Attribute Profile) services to establish connections and transmit data. The Bluetooth GATT protocol plays a pivotal role in facilitating communication between Android applications and Bluetooth Low Energy (BLE) devices.

The pairing process commences with the smartphone establishing a connection with the Beurer devices via the smartphone's Bluetooth app. Subsequently, the SB@App gains the capability to initiate connections with these devices. Upon successful connection establishment, the Android application proceeds to discover the GATT services provided by the respective Beurer device. GATT services are logically organized into profiles, each of which contains characteristics designed to store specific data of interest. Within the spectrum of discovered services, the Android application gains access to the pertinent characteristics that house the data. These characteristics can be interacted through methods such as readCharacteristic() and writeCharacteristic(), which enable reading or writing data, respectively.

Figure 19: a) Beurer FT 95 thermometer; b) Beurer BF 600 scale; c) Beurer PO60 oximeter.

The app's logic is thoughtfully designed to manage and utilize incoming data in accordance with project requirements. Consequently, measurements obtained from each device are locally stored within the app, transformed into the FHIR format, and securely transmitted to the SB@Cloud, ensuring seamless integration and data standardization. Below, one will find visual representations (Table 16, Table 17, Table 18, Table 19) depicting the measurements obtained from the thermometer, scales, and oximeter, respectively.

Table 16: Thermometer measurements.

Name	Brief Description
Temperature	The temperature in number with a decimal place
Year	The year of the measurement
Month	The month of the measurement
Day	The day of the measurement
Hour	The hour of the measurement
Minute	The minute of the measurement
Second	The second of the measurement

Table 17: Scales measurements.

Name	Brief Description
Weight	The weight with a resolution of 0.01 kg.
Year	The year of the measurement
Month	The month of the measurement
Day	The day of the measurement

Name	Brief Description
Hour	The hour of the measurement
Minute	The minute of the measurement
Second	The second of the measurement
User ID	The user ID of the measurement
BMI	The BMI of the user at the time of the measurement
Height	The height of the user at the time of the measurement

Table 18: Body composition measurements, also provided from the scales.

Name	Brief Description
Body Fat %	The Body Fat value in %.
BMR	The BMR value in kcal
Muscle %	The Muscle value in %
Soft Lean Mass	The Soft Lean Mass in 0.01 kg
Body Water Mass	The Body Water Mass in 0.01 kg
Impedance	The impedance in Ohm

The data structure for an individual measurement, transmitted by the Beurer PO60 pulse oximeter in response to the "acquire measurement data" write command, has been designed to accommodate a maximum of ten measurements. In situations where the data exceeds the capacity of a single notification, the remaining measurement data will be conveyed in the next notifications. This methodology ensures the efficient and continuous transfer of data from the pulse oximeter to the recipient application, guaranteeing the successful transmission of all pertinent measurements, even when they extend across multiple notifications.

Table 19: Oximeter measurements.

Name	Brief Description
Header	The header for acquire measurement data notify
Start year	This value represents the starting year of the measurement.
Start month	This value represents the starting month of the measurement.
Start day	This value represents the starting day of the measurement.
Start hour	This value represents the starting hour of the measurement.
Start minute	This value represents the starting minute of the measurement.
Start second	This value represents the starting second of the measurement.
End year	This value represents the ending year of the measurement.

Name	Brief Description
End month	This value represents the ending month of the measurement.
End day	This value represents the ending day of the measurement.
End hour	This value represents the ending hour of the measurement.
End minute	This value represents the ending minute of the measurement.
End second	This value represents the ending second of the measurement.
SpO2 max	This value represents the SpO2 maximum value.
SpO2 min	This value represents the SpO2 minimum value.
SpO2 avg	This value represents the SpO2 average value.
PR max	This value represents the Pulse Rate maximum value.
PR min	This value represents the Pulse Rate minimum value.
PR avg	This value represents the Pulse Rate average value.
Header	The header for acquire measurement data notify
Start year	This value represents the starting year of the measurement.
Start month	This value represents the starting month of the measurement.
Start day	This value represents the starting day of the measurement.
Start hour	This value represents the starting hour of the measurement.
Start minute	This value represents the starting minute of the measurement.
Start second	This value represents the starting second of the measurement.

3.2.3.2 Thermometer, Oximeter, and Scales Screens

This section pertains to the SMART BEAR mobile application screens that are associated with measurements obtained from the thermometer, oximeter, and scales. The visual layout and design of these screens, as presented in the previous version of this deliverable (referred to as Figure 20, Figure 21, and Figure 22), remain unchanged. However, it is noteworthy that the data displayed on these screens has been updated to reflect information retrieved from different devices. Despite this shift in data sources, the fundamental structure and functionality of these screens have been preserved, ensuring continuity and familiarity for users while enhancing the application's capacity to handle data from various devices.

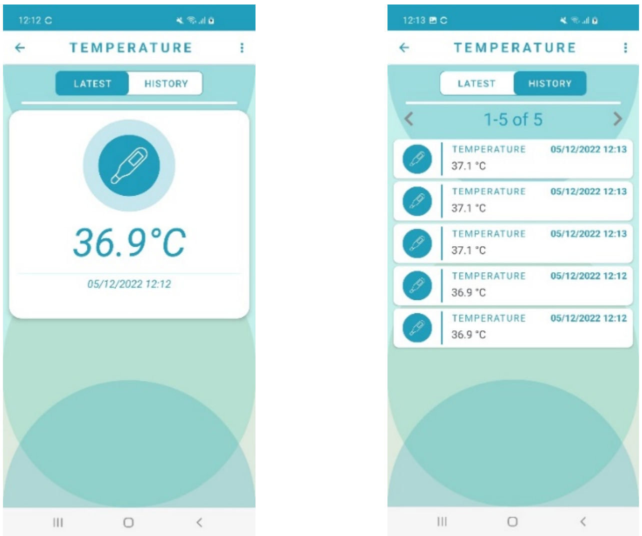


Figure 20: Temperature Screens.

Figure 21: Weight Screens.

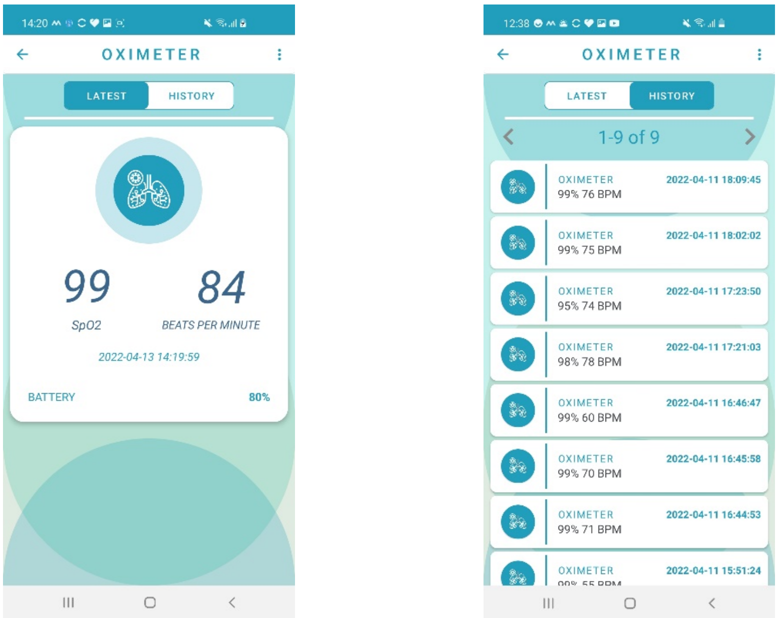


Figure 22: Oximeter Screens.

3.2.3.3 OTICON Hearing Aids

The hearing aids within the SMART BEAR ecosystem (Figure 23) employ Bluetooth GATT services to establish secure connections and facilitate the seamless transfer of data. The initial pairing of these hearing aids with the user's smartphone is facilitated through the smartphone's native Bluetooth application. Subsequently, the SB@App gains the capability to establish connections with the hearing aids, enabling the retrieval of **real-time data** through the Bluetooth GATT service.

Figure 23: OTICON miniRITE T.

The retrieved data is then transformed into the FHIR format, ensuring standardized data representation, and subsequently transmitted to the SB@Cloud for further processing and analysis. This process of data acquisition, transformation, and transmission is orchestrated by a background worker integrated within the application. Notably, this background worker executes the data retrieval and transfer task at regular intervals, with a frequency set to every five minutes. This recurrent process ensures the consistent and up-to-date transfer of hearing aid data to the SB@Cloud, contributing to the real-time monitoring and analysis capabilities of the SMART BEAR platform. A detailed view of the acquired data is depicted in Table 20.

Table 20: Detailed measurements from hearing aids device.

Item Description	Value Type
Main volume	Integer
Muted status	Integer
Main program	Integer
Noise level estimator	array of unsigned floats
Signal to Noise ratio estimator	array of unsigned floats
Broadband level estimator	unsigned float
Time since boot	integer (seconds)
Main volume	Integer

3.2.3.4 Hearing Aids Screens

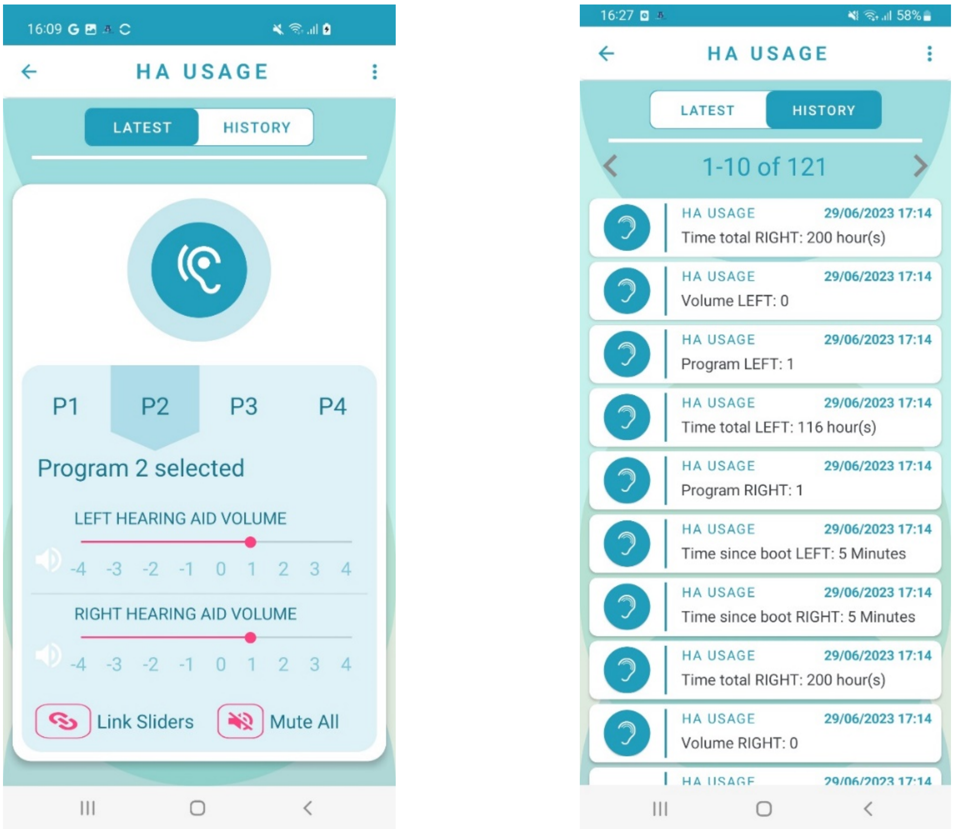


Figure 24: a) Hearing aids usage screen; b) hearing aids history screen.

End users have the capability to select their preferred program and fine-tune the volume settings for both the left and right hearing aids directly from the main screen, as it is depicted in Figure 24. To enhance user convenience, quick-access buttons for muting all sound and linking volume sliders are prominently featured on this main screen. Additionally, users can readily access the historical data, complete with timestamps, from the same screen once navigated to "History", facilitating a comprehensive understanding of their hearing aid usage.

3.2.3.5 Remaining Devices

For the remaining devices, specifically the OMRON M7 Intelli IT hem-7361t–ebk and iHealth devices, it's crucial to note that the established connection, communication, and data transfer pipelines remain consistent with the descriptions provided in the previous deliverable. These devices continue to be fully supported by the SMART BEAR mobile application, ensuring uninterrupted compatibility and data handling as per the previously documented standards.

3.3 *SB@APP Development*

3.3.1 *Cognitive Games*

The incorporation of two cognitive games, "Matching Game" (Figure 25, Figure 26) and "Quiz Game," (Figure 27, Figure 28) within the SMART BEAR mobile application serves multifaceted purposes. Firstly, these games are designed to stimulate mental faculties and support the preservation of cognitive function, particularly in older adults, potentially contributing to the delay of cognitive decline.

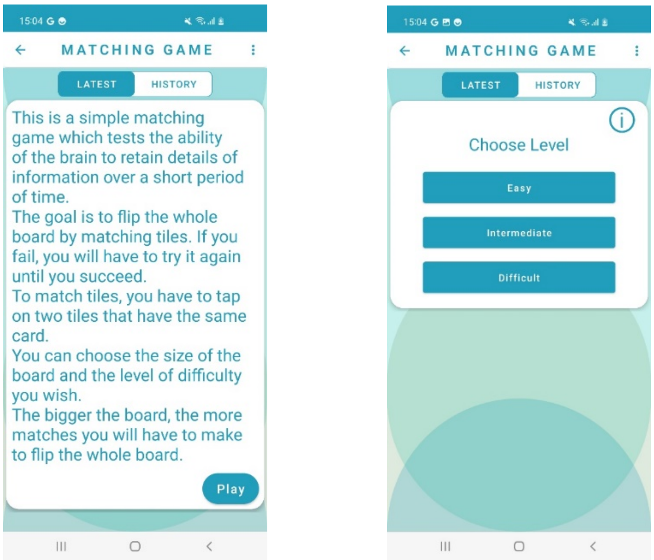


Figure 25: a) Game explanation; b) choosing difficulty level.

Additionally, they offer avenues for entertainment and social engagement, mitigating feelings of isolation among users. Furthermore, these games serve as valuable tools for monitoring cognitive health and gathering pertinent data, enhancing the SB@App's utility. With instructions readily available to users each time they access the games, along with the option to revisit them via the "i" icon, the user experience is intuitively guided. Furthermore, the application maintains a comprehensive history of user scores for both games, with these scores transmitted to SB@Cloud, thus enriching the overall functionality and impact of the SMART BEAR platform.

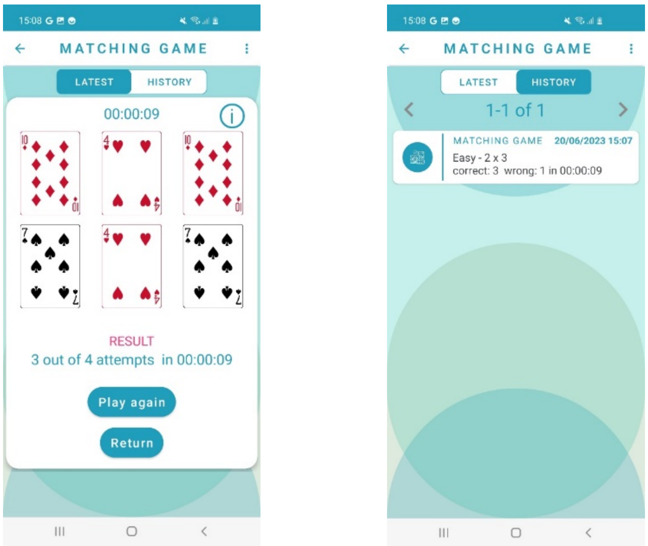


Figure 26: a) Matching game result; b) matching game history.

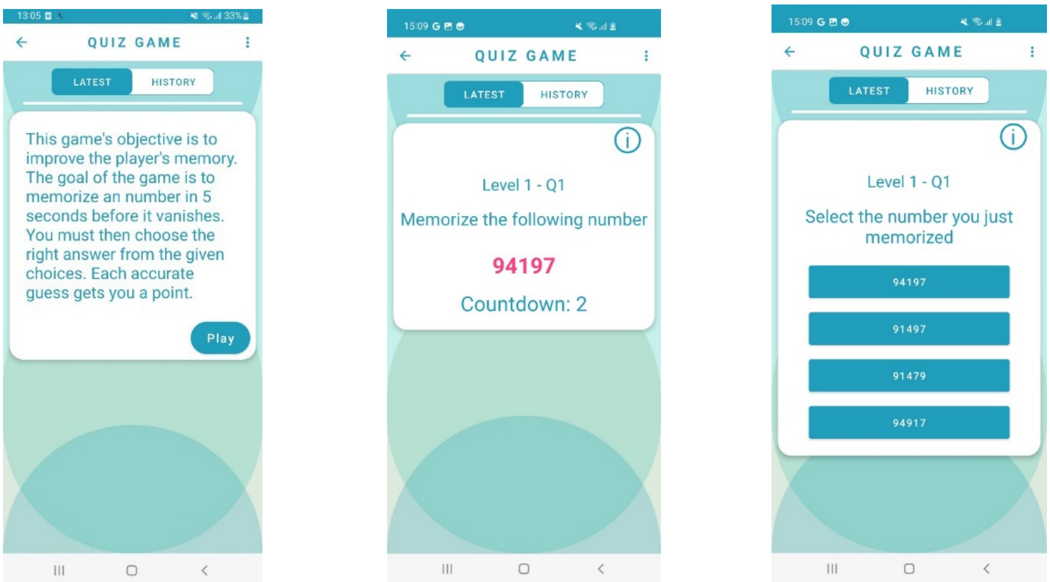


Figure 27: a) Quiz game instructions; b) memorizing the number; c) choosing the answer.

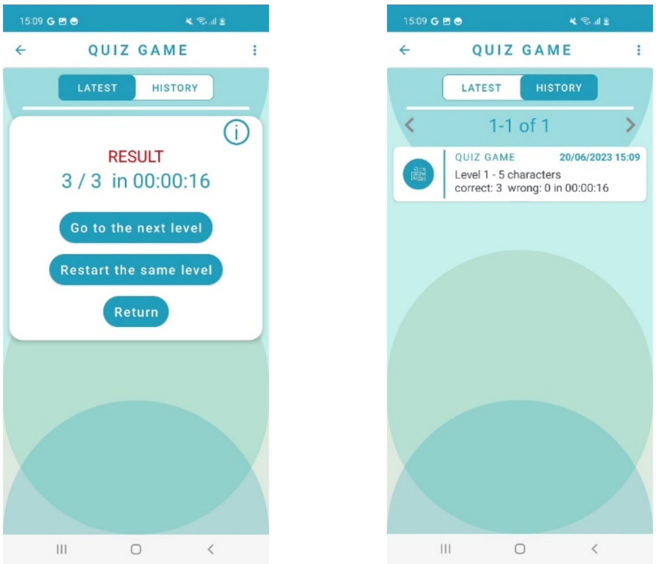


Figure 28: a) Quiz game result; b) quiz game history.

3.3.2 *Authorization Upgrade*

The transition from API keys to JWT (JSON Web Token) tokens within the SB@App's authorization process, prompted by the shift from a VM-based system to Kubernetes, represents a significant enhancement in security, scalability, and flexibility. This transition is paramount due to several compelling reasons. Firstly, JWT tokens offer a higher degree of security as they are cryptographically signed, ensuring the integrity of the data being transmitted and verified. Additionally, JWT tokens are time-bound, reducing the window of vulnerability compared to static API keys. Secondly, JWT tokens adhere to widely recognized and accepted standards, rendering them interoperable with a variety of systems and services, a crucial factor in the ever-evolving technology landscape. Thirdly, JWT tokens enable more granular control over permissions and access, thus enhancing overall system control and ensuring that only authorized entities can access specific resources. Overall, this shift to JWT tokens represents a fundamental step towards robust, secure, and future-proof authorization mechanisms, aligning the SB@App with industry best practices and safeguarding the integrity of user data and interactions.

3.3.3 *Localization Support*

The inclusion of comprehensive language support, encompassing Greek, Italian, French, Spanish, Romanian, Portuguese, and English, within the application marks a crucial step towards enhancing accessibility and inclusivity for all participants in the SMART BEAR project. This multilingual approach acknowledges the diverse linguistic backgrounds of users, ensuring that language barriers do not impede their engagement and understanding of the platform. Furthermore, the incorporation of subtitles for Otago videos, currently available in English, Italian, and Romanian, underscores the commitment to accessibility and education. This feature aids users in comprehending crucial content, promoting proper exercise techniques, and ultimately contributing to their overall well-being. In essence, this robust language and subtitle support not only fosters inclusivity but also enriches the user experience, making the SMART BEAR platform more effective and user-friendly across a broader audience.

3.3.4 *Data Collection Protocol*

Furthermore, the SMART BEAR application has undergone significant enhancements to optimize its connectivity with all available devices. This improved functionality operates seamlessly in the background, establishing connections with devices periodically at two-hour intervals. This periodic retrieval of measurements ensures that data remains current and up to date, enhancing the timeliness and accuracy of the information transmitted to the SB@Cloud. Moreover, the application now incorporates the retrieval of aggregated sensor values from the HomeHub API, also on a two-hour cycle. This integration substantially broadens the scope of sensor data collection, providing a more comprehensive and nuanced understanding of the user's environment and activities. In summary, these enhancements bolster the application's efficiency and data collection capabilities, further reinforcing the SMART BEAR platform's capacity to deliver meaningful insights and support for users' health and well-being.

3.3.5 *Additional Features*

3.3.5.1 Weight Indicator

In scenarios where a user does not possess a smart scale, the SMART BEAR application accommodates their needs by offering the option to manually input their weight (Figure 29). This manual input feature ensures that users without specific devices can still participate in the platform's data collection and health monitoring processes. However, it's important to note that once the necessary configuration for setting up a smart scale has been successfully completed, the manual input option is intentionally deactivated. This design choice encourages users to transition to more automated and precise data collection methods, thereby promoting consistency and accuracy in their health-related measurements within the SMART BEAR ecosystem even without having the specific devices.

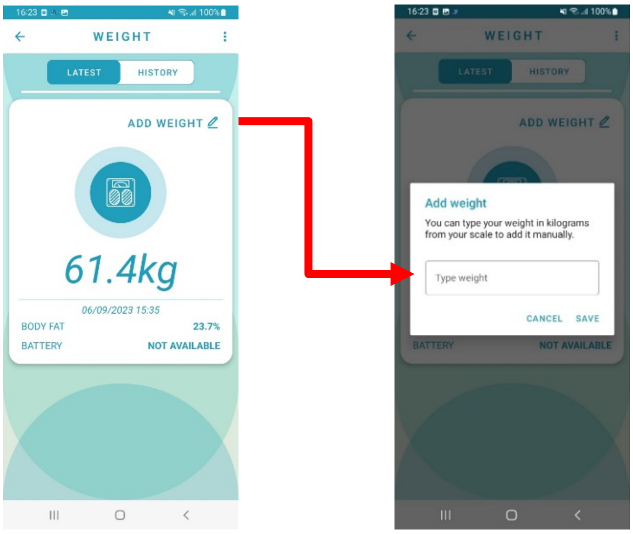


Figure 29: a) The user taps on "ADD WEIGHT"; b) types their weight and taps on "SAVE".

3.3.5.2 Location Indicator

SMART BEAR application offers its end users the flexibility to indicate their current location status within the "My Settings" section of the app, providing an added layer of customization to their experience. This location information requires a one-time input from users, streamlining the process. The data regarding their location is then seamlessly included in the information transmitted to the SB@Cloud platform. By empowering users to specify whether they are presently at home, the application enhances its capacity to contextualize and analyze health-related data in relation to the user's environment. This feature (Figure 30) enriches the platform's ability to provide tailored insights and support, ultimately contributing to a more personalized and effective health monitoring experience for SMART BEAR participants.

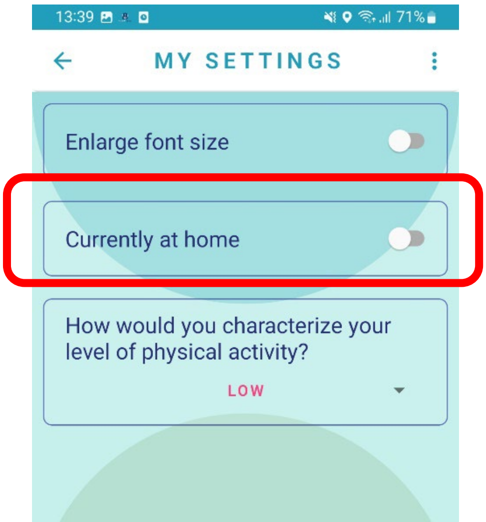


Figure 30: The user drags the switch to the right to indicate they are currently at home.

3.3.5.3 Device Type Inclusion

Another notable enhancement in the SMART BEAR application is the inclusion of device type information in all data transmitted to the SB@Cloud platform (Table 21). This significant improvement enables a critical level of transparency and accountability within the data collection process. By indicating the source device type for each data point, the SMART BEAR technicians gain the ability to verify the expected transmission of data from specific devices, ensuring data integrity and reliability. This, in turn, empowers the SMART BEAR personnel to take prompt and appropriate actions, in accordance with the outlined procedures in the **intervention** section of this deliverable. This carefully crafted tracking and verification process further reinforces the platform's robustness and data quality, contributing to its effectiveness in delivering meaningful insights and support to users' health and well-being.

Table 21: Device Id and device mapping.

Device ID	Description
1	Smartphone
2	Smartwatch
3	Smart Blood Pressure
4	Smart scale
5	Smart Hearing aids - Left
6	Smart Hearing aids - Right
7	Smart Thermometer
8	Smart Oximeter
9	Smart Bulb E27/14
10	Hue Motion Sensor
11	Aqara Sensor Bedroom
12	Aqara Sensor Living Room

3.3.5.4 Other Minor Updates

Within the medication app functionality of the SMART BEAR system, the SB@App plays an essential role by transmitting user responses concerning medication intake to the SB@Cloud platform. This seamless data exchange ensures that medication adherence is readily accessible for analysis and monitoring, contributing to users' health management.

Moreover, the SB@App provides valuable guidance to participants regarding their engagement in Otago exercises, exercising with a "caution" warning, or refraining from exercises altogether, in strict accordance with precise clinician-established guidelines outlined by SB@Cloud. This dynamic guidance empowers participants to make informed decisions about their exercise routines, aligning their activities with recommended healthcare guidelines. Furthermore, the application collects and transmits metrics related to the Otago app to the SB@Cloud platform, enriching the data repository and enabling comprehensive assessment and support for users' well-being and fitness endeavours.

Lastly, it is crucial to emphasize that all of these improvements have been seamlessly integrated into the mobile application, resulting in a comprehensive update to both the technical documentation[21] and end-user manual[22] to align with these elevated features.

[21] https://drive.google.com/file/d/1DImjkmN93Fx5z9pMWZhWuFKi-YSeAKuk/

[22] https://drive.google.com/file/d/1DFe8r-IgsKHNlc0cw4C9d1X1ZzMvDgon/

4 Measurements, Data Types, Data Values and Metadata (Delta – Updates)

In this section, the data collected by the smart home's sensors and devices, including the various kinds of information gathered (i.e., data types), its values (i.e., value type), and its associated metadata, is discussed. Most significantly, a hearing aids device is implemented in SMART BEAR solution alongside the adoption of several other devices. Garmin VENU SQ and Garmin VIVOSPORT were utilised for activity and sleep tracking, while Withings Body+ and iHealth Scale Fit were deployed for body-related measurements. For temperature monitoring, Withings Smart Thermometer and iHealth Thermo Pro were used. For the blood pressure, OMRON M7 was adopted in parallel with Withings BPM Core. The functionalities presented by iHealth Smart Air Plus smart oximeter along with the accompanying software resulted in consistent SPO_2 measuring. Lastly, the Oticon MiniRite T was used as the main hearing aid device. Devices were selected based on their availability to respective SDKs and APIs.

In this iteration of Home Enabling Components, the inclusion of Beurer devices has been implemented for the respective devices. Additionally, an indication of their storage location within SB@Cloud is displayed, which can be categorized as follows: FHIR, non-FHIR, none, or both FHIR and non-FHIR.

4.1 *Body-Related Measurements*

For the body-related measurements, Withings Body+ and iHealth Smart Scale were utilised. The measurements that can also be retrieved by the API are presented in Table 22.

Table 22: List of measurements from the smart body scale devices.

Device	Description	Value Type	Measurement Frequency	SB @ Cloud
Withings Body+	Body weight	kg, floating point	Weekly (x1)	FHIR
	Body muscle mass	kg, floating point	Weekly (x1)	
	Body bone mass	kg, floating point	Weekly (x1)	
	Body fat mass	kg, floating point	Weekly (x1)	
	Body fat free mass	kg, floating point	Weekly (x1)	
	Body fat ratio	%, integer	Weekly (x1)	FHIR
iHealth Smart Scale	Body weight	kg, floating point	Weekly (x1)	FHIR
	Body fat	%, floating point	Weekly (x1)	FHIR
	Body Mass Index	Floating point	Weekly (x1)	FHIR
	Body water	%, floating point	Weekly (x1)	FHIR
	Body lean mass	kg, floating point	Weekly (x1)	FHIR
	Body muscle mass	kg, floating point	Weekly (x1)	FHIR
	Visceral fat rating	Floating point	Weekly (x1)	FHIR
	Protein rate	Floating point	Weekly (x1)	
	Skeletal muscle mass	kg, floating point	Weekly (x1)	FHIR
	Measure time	Unix time format	Weekly (x1)	FHIR
Beurer Scales	Weight	kg, floating point	Weekly (x1)	FHIR
	Measurement year	Integer	Weekly (x1)	FHIR
	Measurement month	Integer	Weekly (x1)	FHIR
	Measurement day	Integer	Weekly (x1)	FHIR

Device	Description	Value Type	Measurement Frequency	SB @ Cloud
	Measurement hour	Integer	Weekly (x1)	FHIR
	Measurement minute	Integer	Weekly (x1)	FHIR
	Measurement second	Integer	Weekly (x1)	FHIR
	User ID	String	Weekly (x1)	
	Height	Cm, Integer	Weekly (x1)	
	Body Fat %	%,	Weekly (x1)	FHIR
	BMR	Kcal,	Weekly (x1)	
	Muscle %	%	Weekly (x1)	
	Soft Lean Mass	0.01 Kg,	Weekly (x1)	
	Body Water Mass	0.01 Kg,	Weekly (x1)	
	Impedance	Ohm,	Weekly (x1)	

Specifications for the devices are available at Withings[23] and iHealth[24] webpages, respectively.

4.2 *Blood Pressure Measurements*

For the blood pressure measurements, Withings BPM Core and OMRON M7 were utilised. The measurements that can also be retrieved by the API are presented in Table 23.

Table 23: List of measurements from the smart blood pressure devices.

Device	Description	Value Type	Measurement Frequency	SB @ Cloud
Withings BPM Core	Diastolic Blood Pressure	mmHg, integer	Daily (x2)	FHIR
	Systolic Blood Pressure	mmHg, integer	Daily (x2)	FHIR
	Heart Pulse	BPM, integer	Daily (x2)	FHIR
	ECG signal	μV, time series (20 sec, Fs=500Hz)	Daily (x2)	
	Afib classification	String	Daily (x2)	
	Digital stethoscope signal	time series (20 sec)	Daily (x2)	
	Digital stethoscope classification	String	Daily (x2)	
OMRON M7	Diastolic blood pressure	mmHg, integer	Daily (x2)	FHIR

[23] https://www.withings.com/gr/en/body-plus

[24] https://ihealthlabs.eu/en/74-smart-body-composition-scale-ihealth-fit-hs2s-856362005135.html

Device	Description	Value Type	Measurement Frequency	SB @ Cloud
	Systolic blood pressure	mmHg, integer	Daily (x2)	FHIR
	Irregular heart rate symbol	integer	Daily (x2)	
	Heart pulse	BPM, integer	Daily (x2)	FHIR
	Irregular pulse detection flag	Integer	Daily (x2)	
	User id	Integer	Daily (x2)	
	Atrial Fibrillation flag for reading	integer	Daily (x2)	
	Atrial Fibrillation mode of device	integer	Daily (x2)	
	Sequence number of readings	Integer	Daily (x2)	
	Number of detected artifacts	Integer	Daily (x2)	
	Movement error symbol	integer	Daily (x2)	
	Movement detection flag	Integer	Daily (x2)	
	Cuff wrap guide	Integer	Daily (x2)	
	Irregular heart beat detection key	Integer	Daily (x2)	
	Measurement timestamp	Unix time format	Daily (x2)	FHIR
	Blood pressure measurement mode	Integer	Daily (x2)	

Specifications for the devices are available at Withings[25] and OMRON[26] webpages, respectively.

4.3 *Body Temperature Measurements*

For the body temperature measurements, Withings Smart Thermometer and iHealth Thermo Pro were utilised. The measurements that can also be retrieved by the API are presented in Table 24.

Table 24: List of measurements from the smart thermometer devices.

[25] https://www.withings.com/gr/en/bpm-core

[26] https://www.omron-healthcare.com/eu/blood-pressure-monitors/m7_intelli_it_2.html

Device	Description	Value Type	Measurement Frequency	SB @ Cloud
Withings Smart Thermometer	Temperature	°C, integer	Daily (x2)	FHIR
	Body temperature	°C, integer	Daily (x2)	
	Skin temperature	°C, integer	Daily (x2)	
iHealth Thermo Pro	Temperature	°C, integer	Daily (x2)	FHIR
Beurer Thermometer	Temperature	°C, integer	Daily (x2)	FHIR
	Measurement year	Integer	Daily (x2)	FHIR
	Measurement month	Integer	Daily (x2)	FHIR
	Measurement day	Integer	Daily (x2)	FHIR
	Measurement hour	Integer	Daily (x2)	FHIR
	Measurement minute	Integer	Daily (x2)	FHIR
	Measurement second	Integer	Daily (x2)	FHIR

Specifications for the devices are available at Withings[27] and iHealth[28] webpages, respectively.

4.4 *SPO_2 Measurements*

For the SPO_2 measurements, iHealth Air Smart Pulse Oximeter was utilised. The measurements that can also be retrieved by the API are presented in Table 25.

Table 25: List of measurements from the smart pulse and oximeter devices.

Device	Description	Value Type	Measurement Frequency	SB @ Cloud
iHealth Oximeter	Blood oxygen saturation	%, integer	Daily (x2)	FHIR
	Pulse rate	BPM, integer	Daily (x2)	
	Date and time of measurement	Time-based format	Daily (x2)	FHIR
Beurer Oximeter	Start year	Integer	Daily (x2)	
	Start month	Integer	Daily (x2)	
	Start day	Integer	Daily (x2)	
	Start hour	Integer	Daily (x2)	
	Start minute	Integer	Daily (x2)	
	Start second	Integer	Daily (x2)	
	End year	Integer	Daily (x2)	FHIR
	End month	Integer	Daily (x2)	FHIR
	End day	Integer	Daily (x2)	FHIR
	End hour	Integer	Daily (x2)	FHIR
	End minute	Integer	Daily (x2)	FHIR

[27] https://www.withings.com/gr/en/thermo

[28] https://ihealthlabs.eu/en/86-smart-non-contact-infrared-thermometer-ihealth-thermopro-nt13b.html

Device	Description	Value Type	Measurement Frequency	SB @ Cloud
	End second	Integer	Daily (x2)	FHIR
	SpO2 max	%, Integer	Daily (x2)	
	SpO2 min	%, Integer	Daily (x2)	
	SpO2 avg	%, Integer	Daily (x2)	FHIR
	PR max	Integer	Daily (x2)	
	PR min	Integer	Daily (x2)	
	PR avg	Integer	Daily (x2)	
	Header	String	Daily (x2)	
	Start year	Integer	Daily (x2)	
	Start month	Integer	Daily (x2)	
	Start day	Integer	Daily (x2)	
	Start hour	Integer	Daily (x2)	
	Start minute	Integer	Daily (x2)	
	Start second	Integer	Daily (x2)	

Specifications for the device is available at iHealth[29] webpage.

4.5 *Activity Tracking Measurements*

For the activity tracking measurements, two Garmin devices were utilised: Venu SQ and VIVOSPORT. These devices are also deployed for information related to stress and sleep cycles. The measurements that can also be retrieved by the API are presented in Table 26. Note that they are the same between the two devices.

Table 26: List of measurements from the smart watch devices.

Device	Description	Value Type	Measurement Frequency	SB @ Cloud
Garmin Venu SQ / VIVOSPORT	Number of steps	number of steps, integer	Continuous	NON-FHIR, FHIR
	User's goal for number of steps	number of steps, integer	-	NON-FHIR
	Distance travelled	meters, floating point	Continuous	NON-FHIR , FHIR
	Active kilocalories (burned through movement)	kCal, integer	Continuous	NON-FHIR
	BMR Kilocalories (burned by existing Basal Metabolic Rate)	kCal, integer	-	NON-FHIR

[29] https://ihealthlabs.eu/en/14-smart-pulse-oximeter-ihealth-air-pom3-855111003910.html

Device	Description	Value Type	Measurement Frequency	SB @ Cloud
	Intensity Minutes	minutes, integer	Continuous	NON-FHIR
	Total duration of vigorous intensity activities (>600 seconds)	seconds, integer	Continuous	NON-FHIR
	Total duration of moderate intensity activities (>600 seconds)	seconds, integer	Continuous	NON-FHIR
	User's goal for duration of moderate to vigorous intensity activity	seconds, integer	-	NON-FHIR
	Floors Climbed	Number of floors, integer	Continuous	NON-FHIR
	User's goal for floors climbed	Number of floors, integer	-	NON-FHIR
	Average heart rate (last 7 days)	BPM, integer	-	NON-FHIR, FHIR
	Average heart rate at rest	BPM, integer	Continuous	NON-FHIR
	Minimum heart rate value	BPM, integer	Continuous	NON-FHIR, FHIR
	Maximum heart rate value	BPM, integer	Continuous	NON-FHIR, FHIR
	Entries of mappings between offset from start time in seconds to a heart rate value in previous 15 seconds	BPM, map	-	NON-FHIR
	Average stress level	number, integer	-	NON-FHIR
	Duration in stress (stress levels 26-100)	seconds, integer	Continuous	NON-FHIR

Device	Description	Value Type	Measurement Frequency	SB @ Cloud
	Duration in no stress (stress levels 1-25)	seconds, integer	Continuous	NON-FHIR
	Duration in low stress (stress levels 26-50)	seconds, integer	Continuous	NON-FHIR
	Duration in medium stress (stress levels 51-75)	seconds, integer	Continuous	NON-FHIR
	Duration in high stress (stress levels 76-100).	seconds, integer	Continuous	NON-FHIR
	Stress duration while engaging in physical activity (non-reliable)	seconds, integer	Continuous	NON-FHIR
	Maximum stress level	number, integer	-	NON-FHIR
	Stress quality	label, string	-	NON-FHIR
	Type of activity	label, string	-	NON-FHIR
	Duration of activity	seconds, integer	During activity (continuous)	NON-FHIR
	Distance covered	meters, floating point	During activity (continuous)	NON-FHIR
	Steps	number of steps, integer	During activity (continuous)	NON-FHIR
	Average heart rate	BPM, integer	During activity (continuous)	NON-FHIR
	Maximum heart rate	BPM, integer	During activity (continuous)	NON-FHIR
	Average speed	meters per second, floating point	During activity (continuous)	NON-FHIR
	Maximum speed	meters per second, floating point	During activity (continuous)	NON-FHIR
	Average pace	minutes per kilometer, floating point	During activity (continuous)	NON-FHIR
	Maximum pace	minutes per kilometer, floating point	During activity (continuous)	NON-FHIR

Device	Description	Value Type	Measurement Frequency	SB @ Cloud
	Average cadence while biking	rounds per minute, floating point	During activity (continuous)	NON-FHIR
	Maximum cadence while biking	rounds per minute, floating point	During activity (continuous)	NON-FHIR
	Average cadence while running	steps per minute, floating point	During activity (continuous)	NON-FHIR
	Maximum cadence while running	steps per minute, floating point	During activity (continuous)	NON-FHIR
	Average cadence while swimming	strokes per minute, floating point	During activity (continuous)	NON-FHIR
	Kilocalories burned during activity	kCal, integer	During activity (continuous)	NON-FHIR
	Number of active lengths	number, integer	-	NON-FHIR
	Starting point latitude	degree, floating point	Once	NON-FHIR
	Starting point longitude	degree, floating point	Once	NON-FHIR
	Total elevation gain	meters, floating point	Continuous	NON-FHIR
	Total elevation loss	meters, floating point	Continuous	NON-FHIR
	Move IQ Events activity duration	seconds, integer	Continuous	NON-FHIR
	Move IQ Events activity type	label, string	Continuous	NON-FHIR
	Move IQ Events activity subtype	label, string	Continuous	NON-FHIR
	Total sleep duration	seconds, integer	Continuous	NON-FHIR, FHIR
	Time spent in deep sleep	seconds, integer	Continuous	NON-FHIR, FHIR
	Time spent in light sleep	seconds, integer	Continuous	NON-FHIR, FHIR
	Time spent in REM sleep	seconds, integer	Continuous	NON-FHIR

Device	Description	Value Type	Measurement Frequency	SB @ Cloud
	Time spent awake	seconds, integer	Continuous	NON-FHIR
	Time that sleep could not be measured	seconds, integer	Continuous	NON-FHIR
	Entries of sleep level in deep, light, and awake	unix timestamps in seconds, map	-	NON-FHIR
	Sleep quality	label, string	-	NON-FHIR
	Maximum volume of oxygen consumed per minute per kilogram of body weight at maximum performance (Vo2Max)	milliliters, floating point	During activity	NON-FHIR

Specifications for the devices are available at Garmin Venu SQ[30] and Garmin VIVOSPORT[31] webpages, respectively.

4.6 *Home sensors*

The measurements that can be retrieved by the openHAB are presented in Table 27.

Table 27: List of measurements from the smart home devices.

Device	Description	Value Type	Measurement Frequency	SB @ Cloud
Smart Bulb E27/14	Brightness	%, integer	1-minute period & upon change	Non-FHIR
	Brightness Min	%, integer	5-minute period	Non-FHIR
	Brightness Max	%, integer	5-minute period	Non-FHIR
	Brightness Avg	%, floating point	5-minute period	Non-FHIR
Philips Hue Motion Sensor	Light Illuminance	Illuminance, integer	1-minute period & upon change	Non-FHIR
	Temperature	°C, floating point	1-minute period & upon change	Non-FHIR
	Presence	detection event, date	1-minute period & upon change	Non-FHIR

[30] https://garmin.gr/p-344

[31] https://www.garmin.com/en-US/p/574602

Device	Description	Value Type	Measurement Frequency	SB @ Cloud
	Last Update	detection event, date	1-minute period & upon change	Non-FHIR
	Battery Percentage	%, integer	1-minute period & upon change	Non-FHIR
	Illuminance Min	Illuminance, integer	5-minute period	Non-FHIR
	Illuminance Max	Illuminance, integer	5-minute period	Non-FHIR
	Illuminance Avg	Illuminance, floating point	5-minute period	Non-FHIR
	Temperature Min	°C, floating point	5-minute period	Non-FHIR
	Temperature Max	°C, floating point	5-minute period	Non-FHIR
	Temperature Avg	°C, floating point	5-minute period	Non-FHIR
Xiaomi Aqara Temperature and Humidity Sensor	Temperature	°C, floating point	1-minute period & upon change	Non-FHIR
	Humidity	%, floating point	1-minute period & upon change	Non-FHIR
	Atmospheric Pressure	hPa, integer	1-minute period & upon change	Non-FHIR
	Battery Percentage	%, integer	1-minute period & upon change	Non-FHIR
	Temperature Min	°C, floating point	5-minute period	Non-FHIR
	Temperature Max	°C, floating point	5-minute period	Non-FHIR
	Temperature Avg	°C, floating point	5-minute period	Non-FHIR
	Humidity Min	%, floating point	5-minute period	Non-FHIR
	Humidity Max	%, floating point	5-minute period	Non-FHIR
	Humidity Avg	%, floating point	5-minute period	Non-FHIR
	Atmospheric Pressure Min	hPa, integer	5-minute period	Non-FHIR
	Atmospheric Pressure Max	hPa, integer	5-minute period	Non-FHIR
	Atmospheric Pressure Avg	hPa, floating point	5-minute period	Non-FHIR

Specifications for the devices are available at Aqara[32], Philips Hue[33], Smart Spotlight[34], and Smart Bulb[35] webpages. Due to the utilization of two Aqara humidity/temperature sensors in the homes of study participants, the temperature, humidity, and atmospheric pressure

[32] https://www.aqara.com/us/temperature_humidity_sensor.html

[33] https://www.lighting.philips.co.id/consumer/p/hue-motion-sensor/8718699721749/specifications

[34] https://www.philips-hue.com/en-us/p/hue-white-ambiance-gu10---smart-spotlight----2-pack-/046677542405#specifications

[35] https://www.philips-hue.com/en-us/p/hue-white-ambiance-a21---e26-smart-bulb---100-w/046677562991#specifications

measurements as presented above occur two times to reflect the potential physiological differences between rooms (i.e., living room and bedroom).

4.7 *Hearing aids*

Table 28 presents the list of measurements that have been identified as available and the SB@app can request from the Hearing Aids using Bluetooth GATT services.

Table 28: List of measurements from Hearing Aids.

Device	Description	Value Type	Measurement Frequency	SB @ Cloud
OTICON Minirite T	Main volume	Integer	5-minute period	FHIR
	Muted status	Integer	5-minute period	FHIR
	Main program	Integer	5-minute period	FHIR
	Noise level estimator	array of unsigned floats	5-minute period	FHIR
	Signal to Noise ratio estimator	array of unsigned floats	5-minute period	FHIR
	Broadband level estimator	unsigned float	5-minute period	FHIR
	Time since boot	integer (seconds)	5-minute period	FHIR
	Total usage time	integer (seconds)	5-minute period	FHIR

Specifications for the Hearing Aids device are available at Oticon[36] webpage.
The hearing aid contains two radio technologies, which are described below:

1) The hearing aid contains a radio transceiver using short range magnetic induction technology working at 3.84 MHz. The magnetic field strength of the transmitter is very weak and is always below -40 dBμA/m at a 10 meter distance.
2) The hearing aid also contains a radio transceiver using Bluetooth Low Energy (BLE) and a proprietary short-range radio technology, both working at 2.4 GHz. The 2.4 GHz radio transmitter is weak and is always below 4 dBm e.i.r.p. in total radiated power. The hearing aid complies with international standards concerning electromagnetic compatibility and human exposure. The hearing aid is in compliance with the essential requirements and other relevant provisions of Directive 2014/53/EU.

[36] https://www.oticon.com/-/media/oticon-us/main/download-center/opn/product-information/179076us_td_opn1-2-3_minirite-minirite-t_85.pdf

5 The SMART BEAR Dashboard (Delta – Updates)

The SMART BEAR dashboard (SB@DashBoard) represents a pivotal component designed to cater to the specialized needs of healthcare practitioners within the platform. This user-friendly graphical user interface (UI) serves as an essential gateway for clinicians, providing them with a comprehensive overview of the system's functionality and capabilities.

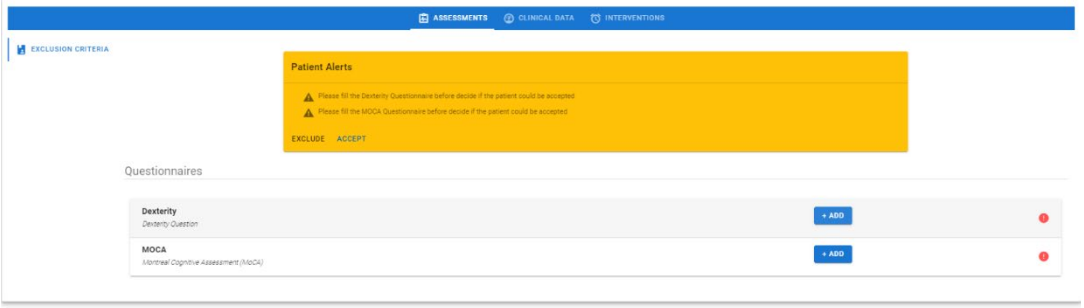


Figure 31: The Exclusion criteria tab.

Within this context, we will delve into the principal enhancements introduced in the SMART BEAR UI, particularly concerning its evolution from its previous iteration, as outlined in D3.4. These enhancements represent a substantial leap forward in terms of user experience, efficiency, and the overall usability of the SMART BEAR platform.

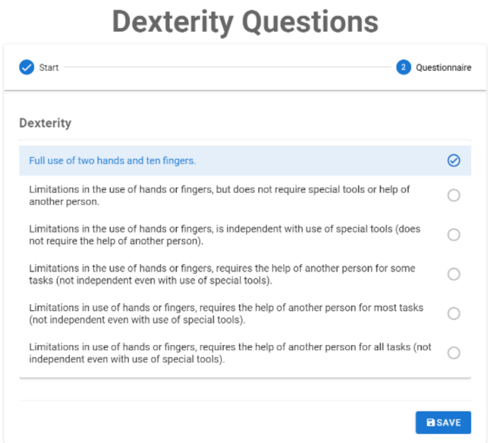


Figure 32: The Dexterity questionnaire tab.

Furthermore, it is essential to note that the information provided here is just a preliminary overview of the SMART BEAR UI, with in-depth details and insights set to be elaborated upon in the forthcoming WP4 deliverables, specifically D4.5 and D4.6. These subsequent documents will comprehensively outline the advanced functionalities that the SMART BEAR UI offers to the clinician community, facilitating their vital roles in the healthcare ecosystem.

5.1 *Participant screening*

Enhancements have been made to the Exclusion Criteria tab, part of the participant screening process, to facilitate the meticulous assessment by clinicians, ensuring that candidates meet the specified prerequisites concerning their MoCA (Montreal Cognitive Assessment) score and Dexterity questionnaire score in order to qualify for participation in the SMART BEAR program.

These improvements are thoroughly illustrated in Figure 31, Figure 32 and Figure 33. This refined functionality plays a pivotal role on the development of a screening tool for the identification of potential new patient candidates.

Upon the completion of both the Dexterity assessment and the MoCA evaluation, several scenarios may unfold:

1. If the patient aligns with the exclusion criteria, a green pop-up notification, as it is depicted in Figure 34, will manifest. In response, the user is afforded the option to either ACCEPT the patient or opt for CANCEL. In instances where the patient does not meet the acceptance criteria, stringent adherence to General Data Protection Regulation (GDPR) mandates necessitates the removal of all related patient data.

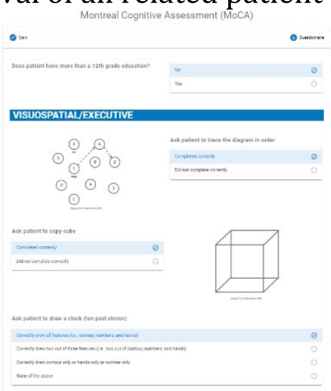


Figure 33: The MoCA questionnaire tab.

2. Conversely, if the patient does not meet the exclusion criteria, an orange pop-up notification, as exemplified in Figure 35, will emerge, bearing a cautionary message. Here, the user is provided with the choice to either proceed with the patient's inclusion by clicking DON'T EXCLUDE or opt to EXCLUDE them. Once again, it is underscored that, in cases where the patient is not accepted, the GDPR-compliant deletion of all pertinent patient data is non-negotiable.

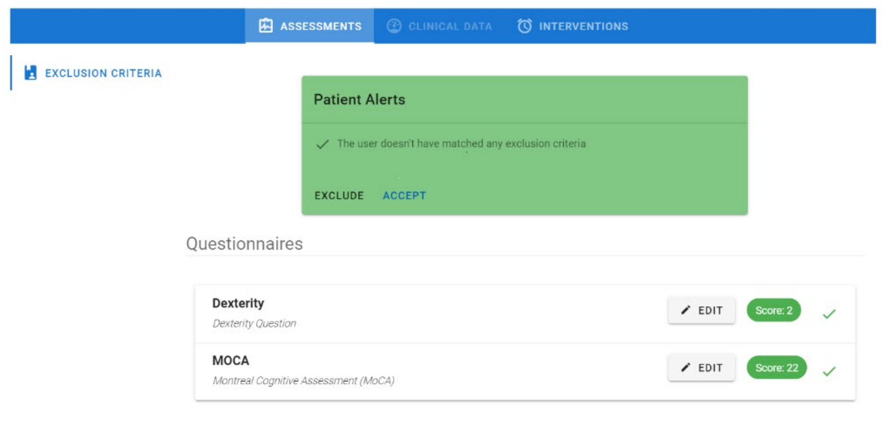


Figure 34: The patient matches the Exclusion criteria.

These refinements in the Exclusion Criteria tab not only streamline the candidate screening process but also adhere to stringent privacy and data protection regulations, ensuring the integrity of patient information and furthering the goals of the SMART BEAR project.

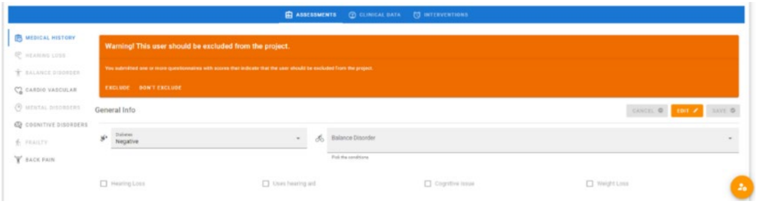


Figure 35: Risk of exclusion.

5.2 *Analytics*

Descriptive analytics are conveniently accessible on the Home page, where they provide a comprehensive overview of the pilot status, as it is illustrated in Figure 36. These analytics are thoughtfully categorized according to various features, including but not limited to generality and application group.

Figure 36: Descriptive analytics in the Dashboard homepage.

Furthermore, the Home page also offers predictive analytics specifically tailored to cardiovascular diseases scenarios. These predictive analytics are designed to be initiated and conducted by users through dedicated Notebooks, as depicted in Figure 37.

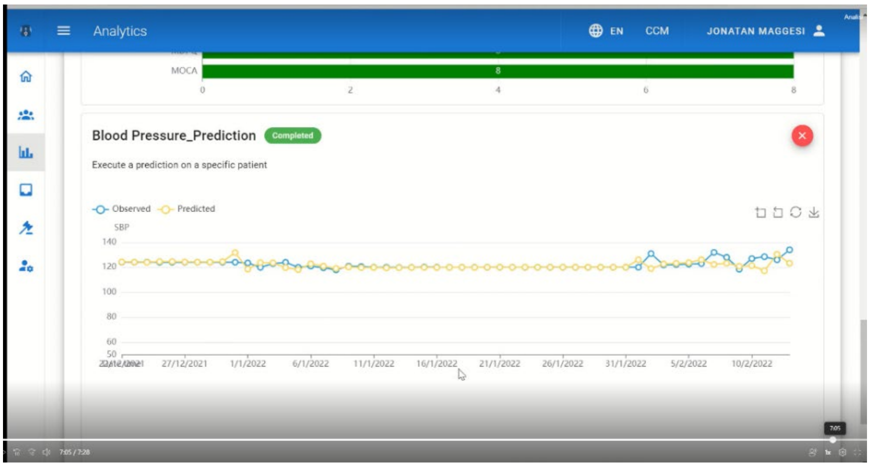


Figure 37. An example of predictive analytics.

The ongoing efforts in this domain are geared toward the development of novel predictive analytics for several comorbidities, among which Cardiovascular diseases, Hearing Loss, and

Balance Disorders are prominently featured. This development initiative seeks to provide essential support for the collaborative synergy between Holobalance and SMART BEAR.

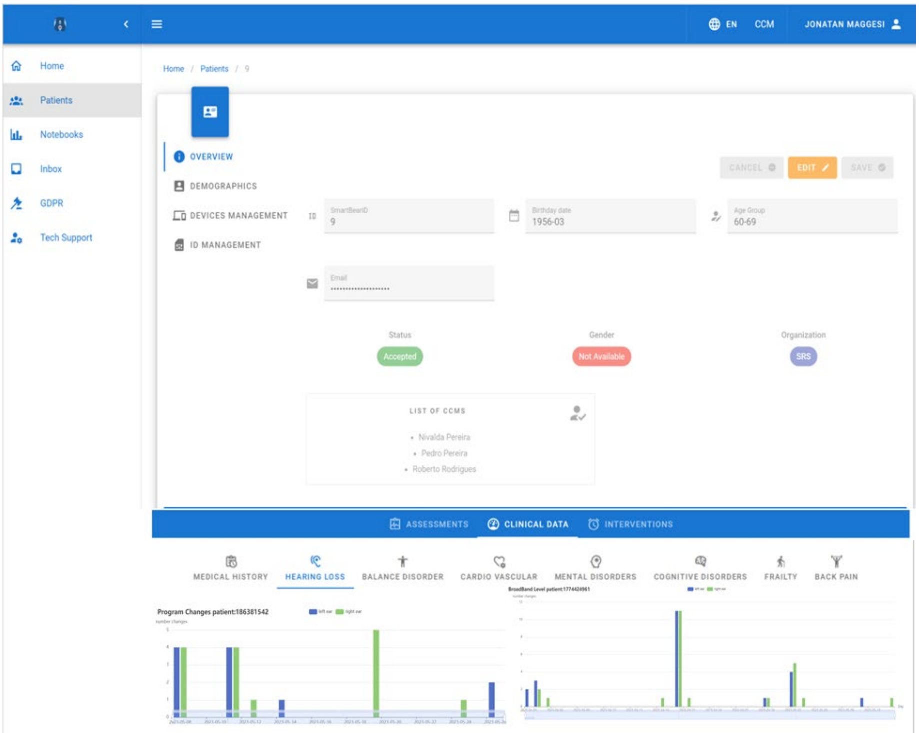


Figure 38. Condition tags.

Notably, these predictive analytics will be efficiently organized and presented by comorbidity categories, employing tags within the Patient's analytics page, a visual representation of which can be observed in Figure 8. This organizational approach is instrumental in enhancing the accessibility and usability of these critical analytics, furthering the objectives of the Holobalance and SMART BEAR integration.

6 Activities, Conditions, and Interventions Monitoring

The purpose of this section is to highlight the implementation work that will provide the technicians the tools and information required in order to prioritize interventions. The scope of assigning priorities is to maximize the data collection process by managing interventions on critical devices that are key points in the data collection infrastructure. To do so we assign an attribute to devices that will indicate if the device is critical or not.

6.1 *Device Monitoring*

To facilitate efficient tracking of devices engaged in data transmission and those encountering adherence issues, a feature supported by the SB@SecurityComponent (presented in D5.3), the SB@Dashboard implements a color-coding system, as exemplified in Table 29. Each individual device is assigned a particular maximum time interval during which it is expected to furnish new data measurements. Devices complying with this schedule are depicted in green, signifying that they have successfully submitted values within the prescribed time frame. Conversely, devices represented in yellow have failed to deliver new data within the stipulated time but are deemed non-critical. Finally, devices appearing in red denote a critical status, as they have not provided data within the designated time interval. This visual classification system aids in swift identification and prioritization of devices based on their data submission status, thereby enabling effective monitoring and troubleshooting.

Table 29: Color coding each condition.

Color	Condition
	Device has submitted values within the maximum time interval.
	Device has not submitted values within the maximum time interval and the device is not critical.
	Device has not submitted values within the maximum time interval and the device is critical.

To establish precise time intervals for each individual device, we drew upon the data collected during the Pilot of Pilots in Madeira. However, it is important to note that as additional data becomes accessible, the potential for a more refined and sophisticated classification of device criticality is anticipated. Presently, the system designates a maximum interval of five days without new data submission for devices classified as critical, as outlined in Table 30. Conversely, non-critical devices are afforded a more lenient maximum interval of 10 days without new data. This framework ensures that the monitoring and tracking processes remain dynamic and adaptable to evolving data patterns and requirements, allowing interventions to occur once needed.

Table 30: Device criticality and maximum interval.

Device is Critical	Maximum Interval
Yes	5 days
No	10 days

6.2 *Dashboard Implementation*

This section highlights the implementation of the dashboard, consisting of two key components: the Participant List View and the Participant Detailed View. The former provides a snapshot of participant status, including data transmission, device status, and interventions conducted. This real-time status tracking system enables efficient prioritization of interventions, ensuring the project's data collection processes remain on track. The latter, the Participant Detailed View, offers a comprehensive overview of device statuses, last read values, and critical intervention-related details, enhancing transparency and real-time data access for technical teams. These tools collectively contribute to the success of the SMART BEAR project by streamlining data collection and intervention management.

6.2.1 *Participant list view*

In the following table, Table 31, a snapshot of the current status of participants in the data collection infrastructure is presented, focusing on their data transmission, device status, and interventions conducted. This information serves as the most crucial tool for tracking and managing data collection processes efficiently.

Table 31: Participant list view.

Participant	Status	Last read value	Device without new data	Intervention done
1st		6 hours ago	N/A	N/A
2nd		6 days ago	Smartphone	Yes – dd / mm / yy
3rd		11 days ago	Oximeter	No

The table includes key details such as the time elapsed since the last data transmission, the device responsible for the interruption in data, and whether an intervention has been performed by the technician personnel. This information is instrumental in **prioritizing** interventions, ensuring seamless data collection, and maintaining the integrity of our infrastructure.

This real-time status tracking system allows us to effectively prioritize interventions by focusing the efforts on critical devices; thereby, critical participants. For instance, in the case of the second participant, where the smartphone has not transmitted data for six days, an intervention was conducted on a specific date (i.e., dd/mm/yyyy) to address the issue. This proactive approach ensures that the efficiency of the data collection process is maximized by promptly addressing any disruptions or issues that may arise. Additionally, for participants like the third one, whose oximeter has not reported data for 11 days, the system indicates that no intervention has been performed yet, prompting the personnel to assess the situation and take appropriate action as needed. This structured approach to monitoring and intervention management contributes to the overall success of the data collection infrastructure, specifically the intervention's monitoring data.

6.2.2 *Participant detailed view*

In this section, an overview of device statuses and last read values for each participant within the data collection infrastructure is showcased (Table 32). This information plays a crucial role in providing a snapshot of data transmission, participant adherence and device health, facilitating efficient monitoring and intervention management. Unlike the previous section, here the concept of a participant detail page is introduced, which includes vital details about interventions performed by the technical team.

Table 32: Participant detailed view.

Device	Status	Last read value
Smartphone		6 hours ago
Smartwatch		6 days ago
Smart Weight Scale		20 days ago
Smart Blood Pressure		5 hours ago
Light Sensors		2 hours ago
Oximeter		18 hours ago
Smart Thermometer		15 days ago
HomeHub		2 hours ago

The above table offers insights into the status of various devices and their respective last read values. This information becomes essential for monitoring the health of the data collection infrastructure. Additionally, it lays the foundation for the participant detail page, which will include critical intervention-related details.

The participant detail page will provide a comprehensive view of interventions conducted by the technical team. This page will also feature the date and time of each intervention, fulfilled by the technician, along with pertinent notes or mentions about the intervention. Moreover, the platform will offer the capability to retrieve **instant** data from the devices. This real-time data retrieval functionality empowers field technicians with the knowledge that an issue has been successfully resolved, as it queries the database for the most recent data submitted by the participant's devices. This feature enhances the efficiency and effectiveness of the intervention management process, ensuring seamless data collection. The following images (Figure 39, Figure 40) depict on how these devices metadata metrics as presented to both the clinicians and help desk operators (the latter responsible for monitoring the devices utilization).

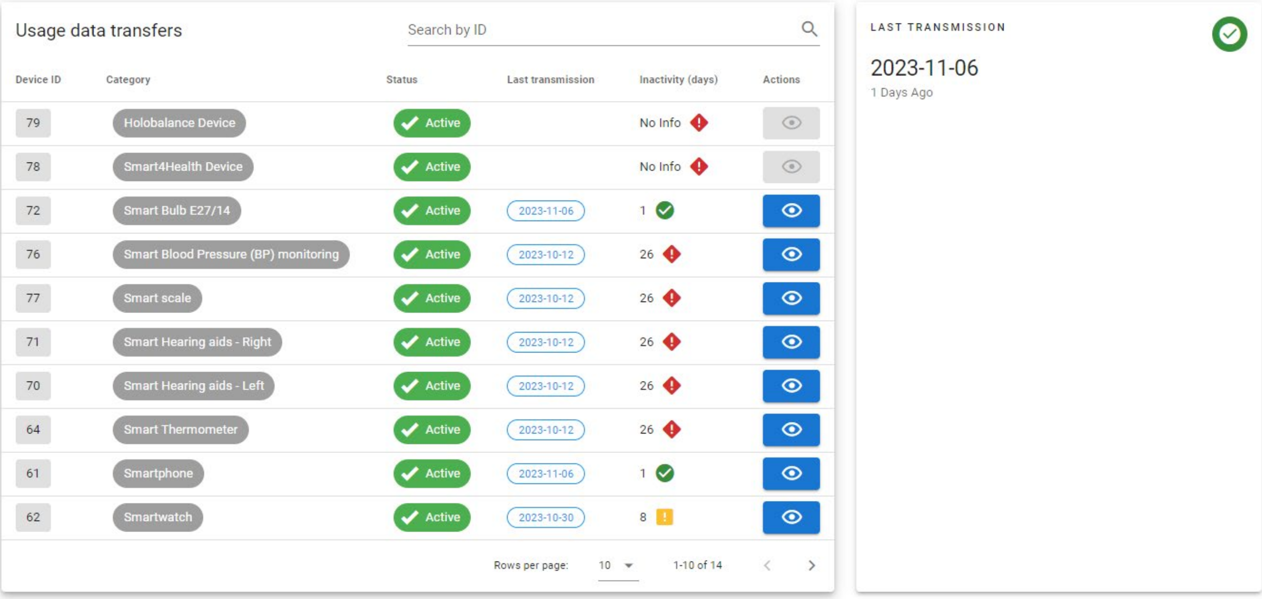


Figure 39: Condition tags.

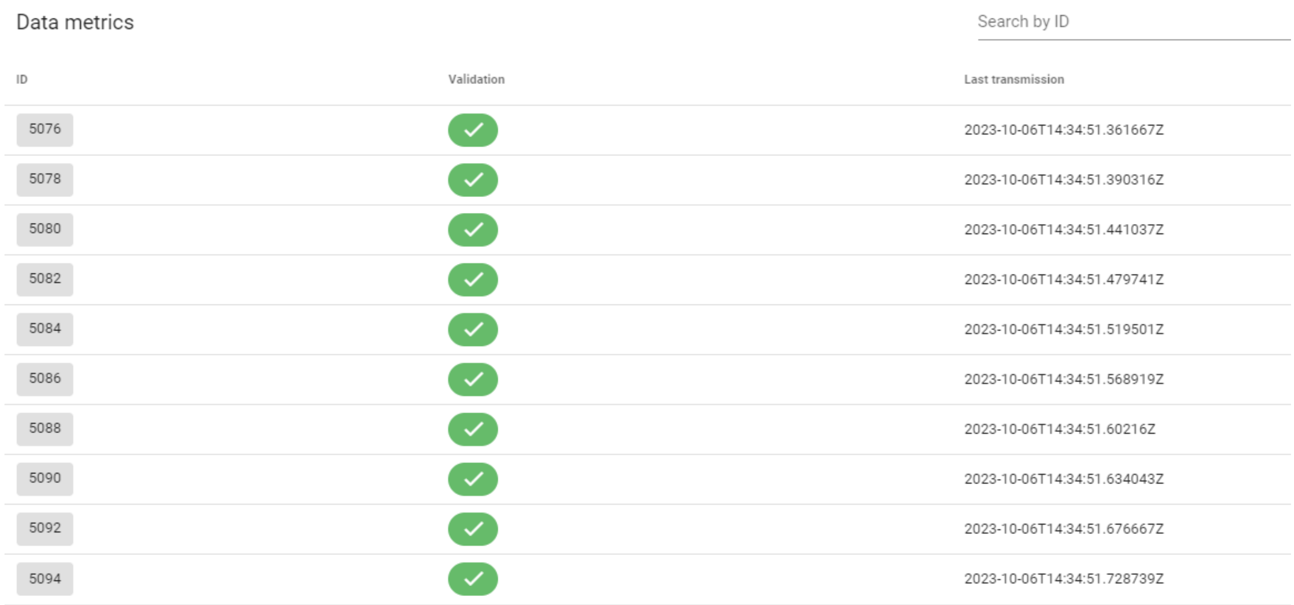


Figure 40: Condition tags.

Compared to the previous section, this segment introduces a more detailed approach to intervention management, offering greater transparency and real-time data access to the technical team. This enhancement enables the personnel to address issues swiftly and comprehensively, further contributing to the success of the data collection.

6.3 *Use Case Scenarios*

The section presents two use case scenarios within the SMART BEAR project, showcasing the practical application of the implemented tools in addressing participants' health monitoring challenges. These scenarios underscore the project's commitment to supporting independent living among participants by swiftly resolving technical issues and ensuring seamless participation in the program.

6.3.1 *The Case of Mr. Alexander Smithe*

Mr. Alexander Smithe, a 68-year-old resident of the rural community within the Municipality of Palaio Faliro, Greece, confronts the daily challenges posed by atrial fibrillation, a cardiac condition that manifests as heightened fatigue, shortness of breath, and palpitations. With a professional background as a technical expert in a prominent IT industry, where he specialized in the design of monitoring devices, Mr. Smithe has a deep-seated interest in such technologies. Eager to enhance his quality of life and better manage his health condition, he enthusiastically enrolled in the Greek pilot of the SMART BEAR project.

Regrettably, after two weeks of active engagement in the SMART BEAR initiative, Mr. Smithe found himself facing an alarming issue. His oximeter measurements, integral for his health monitoring, failed to display on the SMART BEAR application, and concurrently, his smartphone experienced a loss of data connectivity regarding those measurements. These developments have left Mr. Smithe deeply concerned, as he heavily relies on the SMART BEAR platform to effectively manage his health.

Table 33: Mr. Smithe participant list view.

Participant	Status	Last read value	Device without data	Intervention Done
SBGRC01001		14 days ago	Smart Oximeter	No

In parallel, the SMART BEAR team promptly identified the issue from the participants' list and detailed view as they are provided in Table 33 and Table 34, respectively and initiated contact with Mr. Smithe through his registered phone number. Subsequently, upon Mr. Smithe's acceptance of their questions regarding the oximeter measurements, a team of technical experts was dispatched to his residence. They swiftly determined that the oximeter was faulty, necessitating its replacement. Given his technical background, Mr. Smithe deeply appreciates the importance of the SMART BEAR project in supporting the independent living of elderly individuals. Understanding the project's commitment to assisting the elderly population, he is particularly keen on expeditiously resolving the present issue. This will enable him to continue reaping the anticipated benefits of the SMART BEAR initiative, both in terms of improving his quality of life and effectively managing his health.

Table 34: Mr. Smithe participant detailed view.

Device	Status	Last read value
Smartphone		6 hours ago
Smartwatch		2 hours ago
Smart Weight Scale		9 days ago
Smart Blood Pressure		5 hours ago
Light Sensors		2 hours ago
Smart Thermometer		4 hours ago

Device	Status	Last read value
HomeHub		2 hours ago
Oximeter		14 days ago

6.3.2 *The case of Mrs. Kimberly Heart*

Mrs. Kimberly Heart, a 76-year-old resident of Crema, a city near Milan, Italy, confronts the challenges of balance disorders, which occasionally result in rare instances of falling, along with hearing loss. Despite not being particularly tech-savvy, Mrs. Heart possesses a strong desire to embrace technology and its potential benefits. Her lifelong career as a high school teacher had, to some extent, kept her distanced from technological advancements. However, driven by her innate curiosity and a thirst for improvement, she recently became acquainted with the concept of the Hologram-powered rehabilitation program known as Holobalance, which currently collaborates with SMART BEAR. Recognizing the potential to enhance her quality of life and regain stability in her posture and balance, Mrs. Heart eagerly enrolled in the Italian pilot program of SMART BEAR. Her enthusiasm was further heightened when she learned that hearing aids were also integrated into the program, representing a holistic approach to her well-being and making her participation all the more meaningful.

Table 35: Mrs. Heart participant list view.

Participant	Status	Last read value	Device without data	Intervention Done
SBITA01001		6 days ago	Smartphone	No

After a thorough analysis of data collected over a period of six days, the project personnel examined the information from the participants' list view and noted a concerning observation as it is illustrated in Table 35.

This data revealed a conspicuous absence of information gathering from the smartphone associated with participant SBITA01001 corresponding to her SMARTBEAR id. This is further validated from the data showcased at her participant detailed view (Table 36). Such an extended period without data transmission raises concerns regarding the functionality of the device or the possibility of a faulty condition. The absence of any recorded intervention at this stage highlights the need for further investigation and remediation to ensure the participant's continued engagement and the project's objectives.

Table 36: Mrs. Heart participant detailed view.

Device	Status	Last read value
Smartphone		6 days ago
Smartwatch		6 days ago
Smart Weight Scale		6 days ago
Smart Blood Pressure		6 days ago
Light Sensors		6 days ago
Smart Thermometer		6 days ago
HomeHub		6 days ago

Device	Status	Last read value
Oximeter		6 days ago
Hearing Aids		6 days ago

Upon noticing these anomalies, the SMART BEAR technical personnel initiated further investigation leading to potential intervention. Concerns grew when Mrs. Heart failed to answer phone calls, leading to suspicions that she may have experienced a fall or encountered other issues. Subsequently, SMART BEAR personnel physically reached out to Mrs. Heart and discovered that she was unharmed. In a conversation with her, Mrs. Heart explained that her smartphone had ceased to function after the first day of participation. She admitted that her reluctance to charge the device stemmed from a fear of causing harm and a lack of confidence in managing the technology independently.

It is crucial to emphasize that Mrs. Heart's active participation in the SMART BEAR program is vital, not only for her well-being but also for the successful accomplishment of the project's objectives. The challenges Mrs. Heart faced stemmed primarily from her limited familiarity with the technology, rather than any deliberate action on her part.

In light of this situation, immediate action was requested to address the issue and ensure the seamless continuation of Mrs. Heart's participation in the program. To facilitate this, the technical personnel stayed with Mrs. Heart until she felt comfortable and confident in using the smartphone device effectively. They provided the necessary technical support to resolve the smartphone issue and offered reassurance and guidance throughout the process.

Conclusion

In conclusion, the third and final iteration of Home Enabling Components aimed at developing a comprehensive federation of devices and sensors for seamless integration with SMART BEAR's home enabling components system has made significant progress. This deliverable has provided a detailed overview of the objectives and activities accomplished across its six chapters. The initial chapter outlined the project's scope and the allocation of effort required to meet the objectives, while also acknowledging and addressing the challenges encountered in WP3 setting the stage for what is to be presented. At the subsequent chapters, accomplishments in the realm of smart home automation were presented, with a focus on the development of novel functionalities intended for utilization by the technical personnel in the upcoming pilots. Simultaneously, enhancements and refinements to existing features were provided. The development of the SB@App, the mobile application that seamlessly integrates various smart devices including medical ones through the adoption of key technological frameworks and integration schemes was also highlighted. The finalized aggregated measurement list, including all measurements from devices and sensors, along with essential metadata, meeting the requirements set by the reviewers, was presented in detail, which also played a crucial role in managing data flow for related work packages. The key updates to the Dashboard's functionalities include enhanced participant screening through the Exclusion Criteria tab, providing clear notifications for patient acceptance or exclusion while complying with GDPR regulations. Additionally, the dashboard offers descriptive and predictive analytics for various health scenarios, with a focus on cardiovascular diseases, to support collaborative efforts between Holobalance and SMART BEAR. Lastly, the activities, conditions, and interventions monitoring methodology was presented highlighting the importance of specific devices, absence of data, and intervention performed.

This journey to create a federation of devices and sensors has been marked by dedication, innovation, and a commitment to simplify home-based monitoring of health-related data. The achievements and developments detailed in this deliverable are significant steps toward realizing the respective objectives. Looking ahead, the favorable impact that the integration of the smart home ecosystem with the mobile application will have upon end users is anticipated.

We express our gratitude to the entire WP3 team for their hard work and collaboration, as well as to the reviewers for their valuable feedback and guidance. Collectively, significant progress has been achieved, and there is a palpable sense of enthusiasm regarding the forthcoming prospects for SMART BEAR's Home Enabling Components. These components, along with the data they are poised to deliver, are anticipated to result in heightened user quality of life and independent living.

Please note that both D3.5 and D3.6 presently lack the presentation of certain information, which is anticipated to be developed in the coming months and documented in the subsequent Annual Report. This forthcoming development encompasses the SESARAM synergy on the mobile app, along with enhancements to user experience, including reminder customization and adherence achievements. Furthermore, the integration of the new devices due to procurement updates such as hearing aids device is also part of this ongoing initiative, as well as the HomeHub validation.