AAL Programme

AAL-CP-2018-5-149-SAVE

01/09/2019 - 31/08/2022

AAL Programme

Project - SAfety of elderly people and Vicinity Ensuring - "SAVE"

Deliverable: D1.6. Final Technology Report

Version: 2.0

Main editor: VS

Contributing partners: VS, UnitBv, ISS

Table of contents

1. SAVE technical solution overview

The SAVE system is a comprehensive and integrated solution dedicated to enhancing the well-being of elderly individuals by facilitating their ability to remain in their familiar environments for an extended period. The primary objective of SAVE is to ensure their safety and provide optimal care throughout their aging journey. Additionally, this innovative system aims to support informal caregivers, such as relatives, in offering the best possible care to their loved ones while effectively managing their personal and professional commitments.

One of the core features of SAVE is its capability to empower professional caregivers in creating well-structured support plans tailored to the specific needs of each elderly individual. By involving volunteering associations, the system fosters a collaborative approach to caregiving, leveraging collective efforts to maximize the quality of care provided.

The foundation of the SAVE solution lies in cutting-edge technologies, specifically designed to operate efficiently within cloud environments, particularly in the context of *Infrastructure as a Service* (IAAS). Utilizing the power of containerization, the system minimizes overhead, ensuring increased portability and consistent operation across diverse platforms. This not only enhances scalability and speed but also accelerates application development, allowing for timely updates and improved responsiveness.

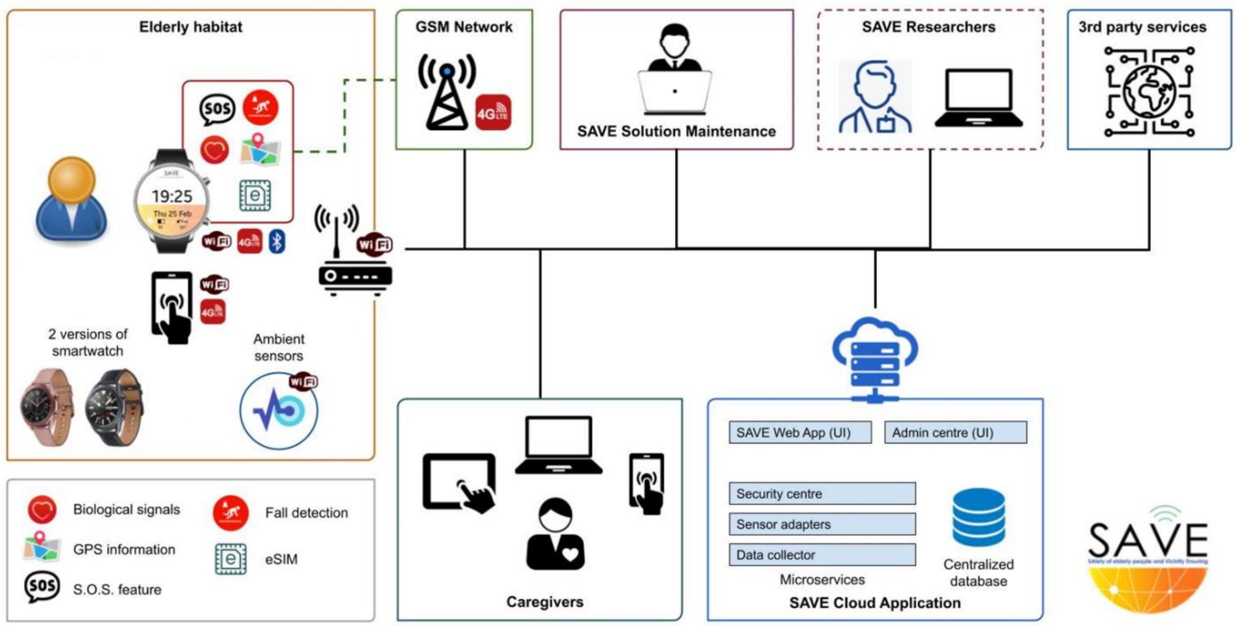


Figure 1 - SAVE solution overview

Through its innovative approach and utilization of the latest technologies, the SAVE system addresses the evolving needs of elderly individuals, informal caregivers, and professional caregiving organizations alike. By promoting a harmonious balance between optimal care, safety, and the preservation of familiar surroundings, SAVE strives to create a compassionate and sustainable caregiving ecosystem for the elderly.

An overview of the SAVE solution is depicted synthetically in Figure 1. The sensors included in the SAVE kit (wearable and ambient) provide raw data about the elderly (end-user) (his/her well-being, activity, environment, location) by connecting through Internet to dedicated services of the SAVE cloud application. The users of the SAVE solution have dedicated user interfaces, in form of responsive web applications and mobile applications, for accessing its features:

- For the end-users: the SAVE smartwatch face and application and the SAVE web application;
- For the caregivers: the SAVE web application;
- For the SAVE solution maintenance staff and for the SAVE researchers: the SAVE Admin Centre web application.

A Manager user profile has been introduced, whose role is to associate Caregiver users with primary institutionalized users. In addition, the Manager user can view all data of primary users associated with the Caregiver users they are responsible for.

All web applications are designed with responsiveness in mind, ensuring they can adapt and be accessed seamlessly from a wide array of devices, including desktops, laptops, tablets, and smartphones.

2. SAVE solution architecture

From an architectural standpoint, the SAVE solution adopts a service-based architecture, employing microservices as independent and smaller software components that fulfil specific tasks. These microservices communicate with each other through a standardized interface, using HTTP or HTTPS protocols for message-based interactions. Typically, JSON format is used for the messages exchanged between services, providing flexibility and interoperability.

Microservices serve as a strategic alternative to monolithic application development. In contrast to monolithic applications, where all components are tightly interconnected, microservices offer greater flexibility and ease of maintenance. By breaking down the application into smaller, independent units, modifications and updates can be implemented more efficiently. Additionally, the microservices architecture facilitates migration to cloud environments, enabling better scalability and management (Figure 2).

The adoption of a microservice-based architecture for the SAVE system offers a multitude of technical and implementation advantages, streamlining application development in a cloud environment. This approach takes into account crucial factors such as implementation efficiency, availability, effective management, adaptability, reliability, and scalability.

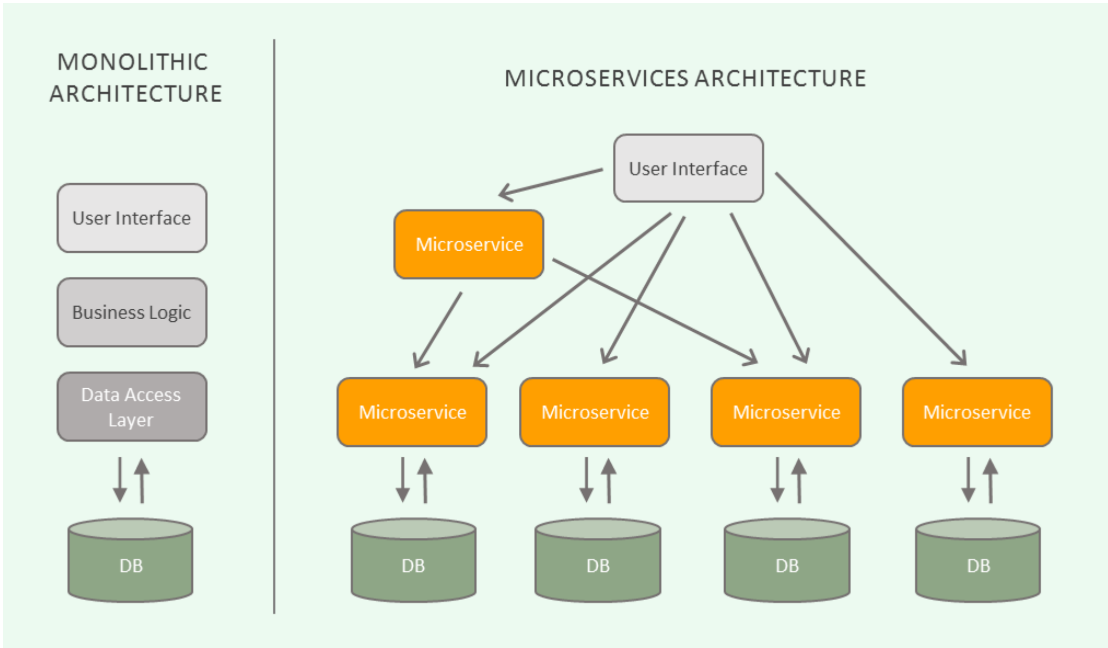


Figure 2 - Monolithic architectures vs. microservices

The microservices approach also facilitates rapid development of new features and functionalities. By breaking down the application into smaller, self-contained units, each microservice can be developed and updated independently. This accelerates the development process, as teams can focus on specific functionalities without disrupting other parts of the system. Consequently, new facilities can be rolled out more quickly, enhancing the system's capabilities and responsiveness to user needs.

One of the primary benefits of employing a microservices-based architecture in the SAVE system is its compatibility with cloud infrastructure. This allows the system to leverage the scalability and flexibility offered by cloud environments, enabling efficient resource allocation and distribution of workloads. As a result, the system can seamlessly accommodate growing demands and handle increased user traffic, ensuring optimal performance.

The SAVE system's architecture is designed to harness the power of cloud hosting, allowing various components to be hosted in the cloud. Each of these components is dedicated to specific tasks and functionalities, ensuring a streamlined and efficient system. Moreover, the microservices-based structure permits seamless integration with external systems provided by third parties. These external systems may include sensors, mobile devices, or web applications that contribute essential data to the SAVE system. The ability to interact with and process data from these external sources enhances the system's overall functionality and empowers it to deliver comprehensive monitoring and analysis capabilities.

The general architecture of the SAVE system (as depicted in Figure 3) is structured around several independent components, which collaborate and communicate with each other to fulfil specific functionalities.

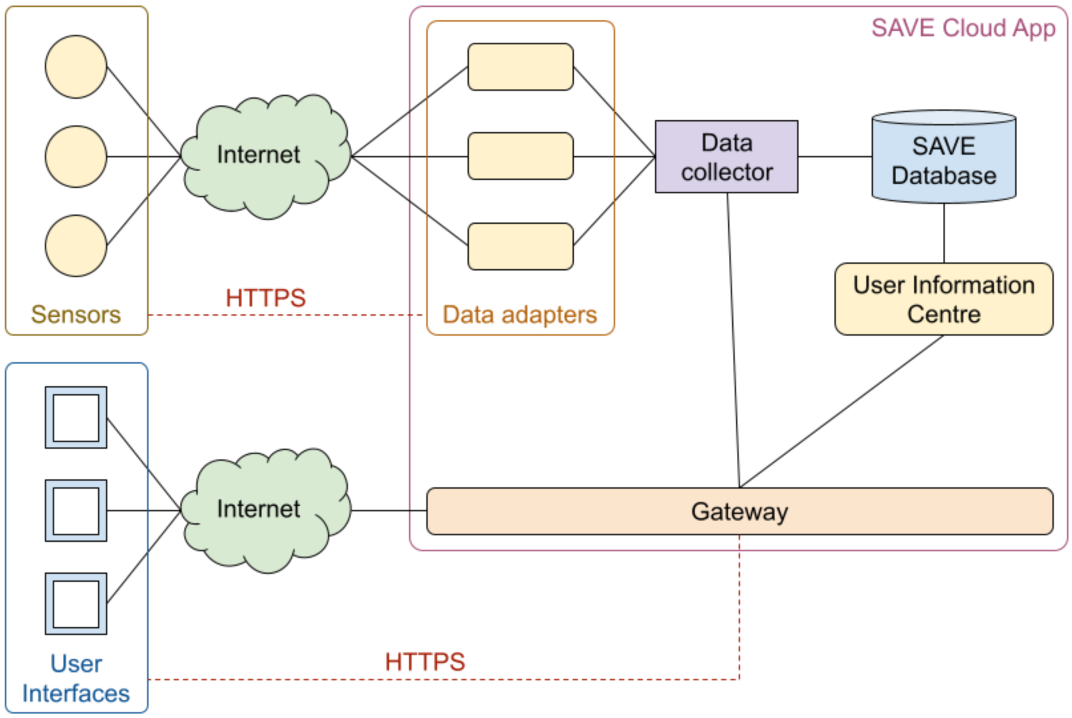


Figure 3 - General architecture of the SAVE system

The SAVE cloud application employs a modular approach, dividing its functionalities into distinct functional units (microservices). Each of these microservices is responsible for handling specific aspects of the business logic:

- *Data Collector*: This microservice gathers data from various sensors and efficiently stores it in the database. It also provides a communication interface through a REST API, enabling easy retrieval of the stored data.
- *Sensor Adapters*: These arrays of microservices establish direct connections with sensing devices and relay the collected data to the Data Collector in a standardized format, ensuring smooth and consistent data exchange.
- *User Information Centre*: This microservice is dedicated to handling authentication and authorization operations, offering REST APIs for seamless access control.
- *SAVE Web App*: Serving as the web-based user interface, this microservice caters to end-users and caregivers, providing a user-friendly experience for managing and monitoring the SAVE system.
- *SAVE Admin Centre*: As another web-based user interface, this microservice is designed for maintenance staff and researchers, facilitating administrative tasks and research-related activities.

For data storage, the SAVE solution utilizes a relational database management system, specifically Oracle's MySQL. To ensure rapid access to the data, a partitioning algorithm has been implemented at the cloud application level, leveraging the kit identifier to optimize data retrieval.

The SAVE system is intentionally designed to be inclusive, enabling the incorporation of other sensor kits developed by third-party providers, as long as they don't require permanent maintenance and configuration. Successful piloting have been conducted with the *Xiaomi Aqara Smart Home Set*, demonstrating compatibility and integration with good results. Also the SAVE solution included a smartwatch as a wearable sensor for monitoring the activity and location of the end-users; a smartphone application connects to the cloud app and feeds the database. Additionally, the *Data Collector* service offers an inclusive REST API, facilitating easy connection with any other sensor systems to be integrated into the SAVE solution. Such an example is the link with the Technological Club device (*eHealth* and *CRT – Choice Reaction Time*).

Figure 4 shows the communication connections between microservices, sensors and user interfaces for the SAVE solutions.

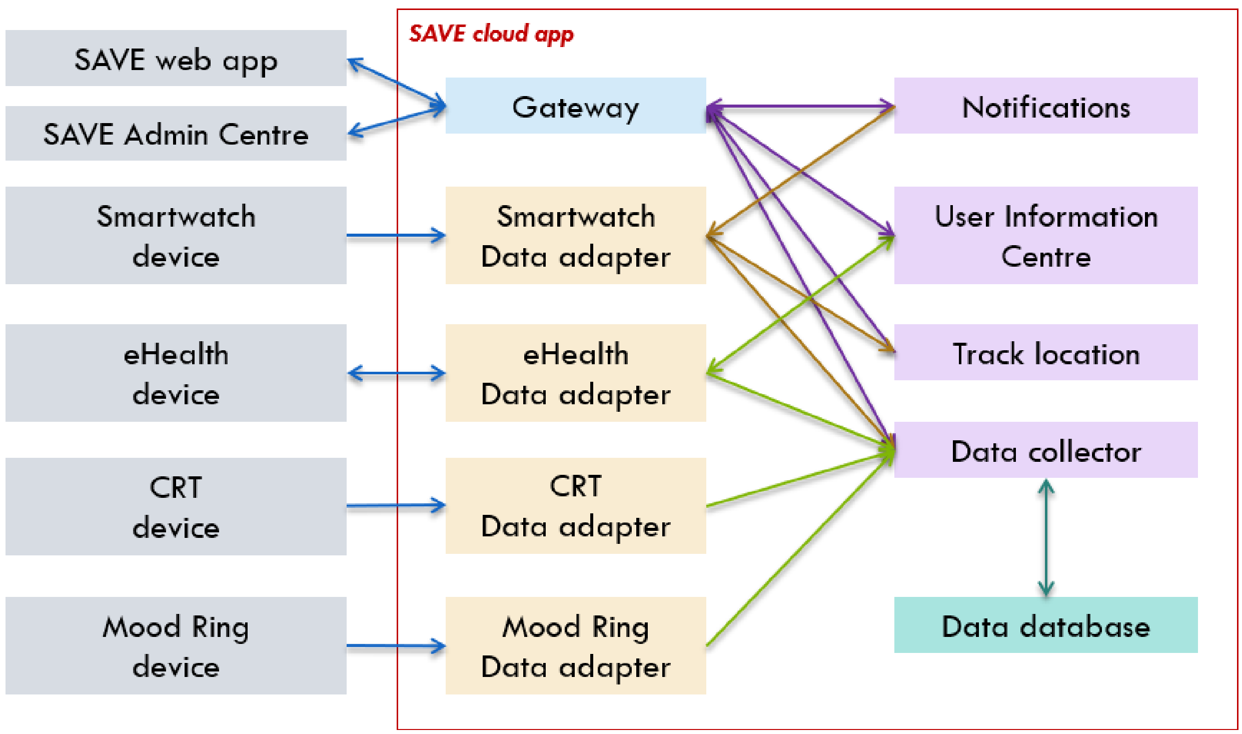


Figure 4 - Communication connections between microservices

The implementation of these components was done using the *Java* programming language and the *Spring Boot* framework. This combination offers numerous advantages for developing microservices applications, ensuring their smooth operation and straightforward maintenance over time. Additionally, the Spring Boot framework simplifies the process of updating individual microservices independently without causing any disruptions to other components, enhancing overall system flexibility and efficiency.

For intercommunication between microservices, REST (*Representational State Transfer*) communication interfaces are utilized, employing the HTTP or HTTPS protocols. Standardizing the messages exchanged between microservices using the JSON format ensures seamless and straightforward communication, both among internal components and with external systems. This setup simplifies the integration of various services and external devices into the SAVE system, promoting easy data exchange and interaction.

3. Technology stack

Figure 5 depicts the simplified technology stack for the microservices included in the SAVE cloud application.

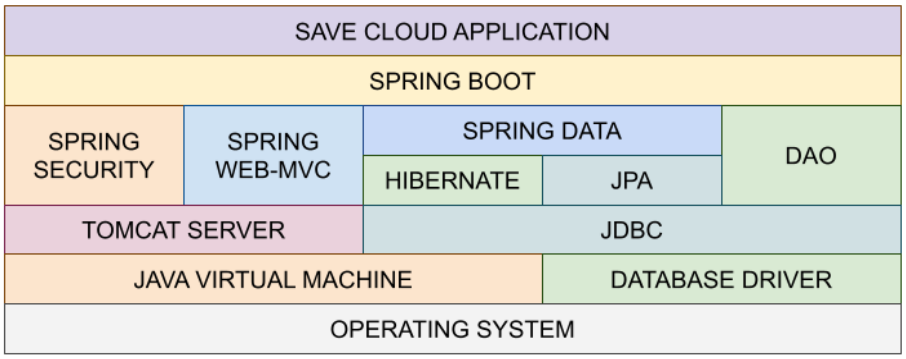


Figure 5 - Technology stack

The user interfaces (the frontend) of the web applications are developed using the Angular framework and the data is retrieved using the REST APIs of the backend (developed using the Spring Boot framework).

The smartwatch face application and the smartwatch application are Tizen web application, developed in HTL, CSS and JavaScript.

4. SAVE database

To manage data persistence, the SAVE system relies on a relational database management system, Oracle's MySQL. This choice provides an efficient solution for organizing and storing the system's information. The database structure includes tables for kits, device types, and devices (figure 6). Kits are registered in the kits table and receive user-friendly names for easy identification. Device types are stored in the *device_types* table, receiving acronyms and default descriptions. The acronyms help user interfaces identify the relevant components for handling data from specific types of devices. Devices are registered in the devices table, associated with a particular kit and linked to a device type. Short and long descriptions provided by end-users facilitate identification and differentiation of multiple sensors of the same type.

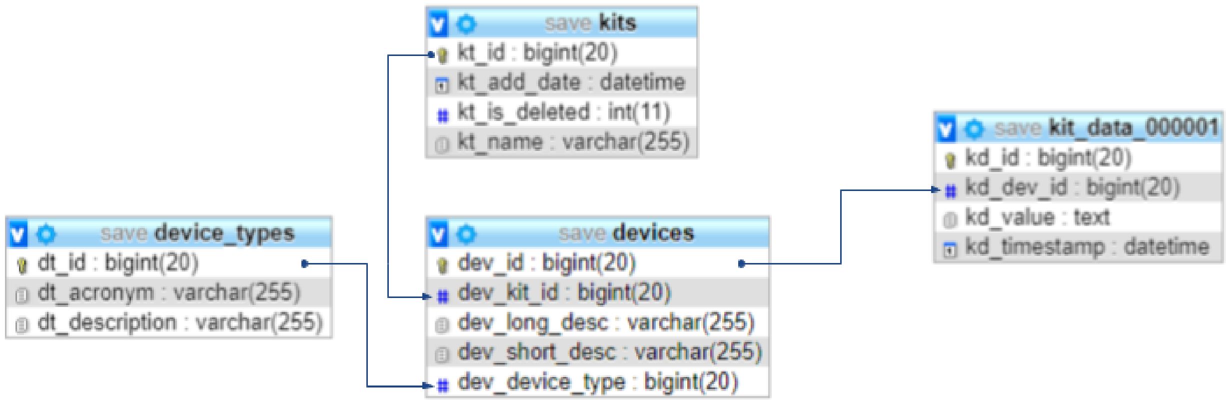


Figure 6 - Tables for the data collecting system

The values read from the sensors (the „data”) are persisted, partitioned at kit level, in the „kit_data_*” tables; there is one table for each kit. The *kd_value* field contains the sensor data, as transmitted by the sensors itself or a sensor adapter.

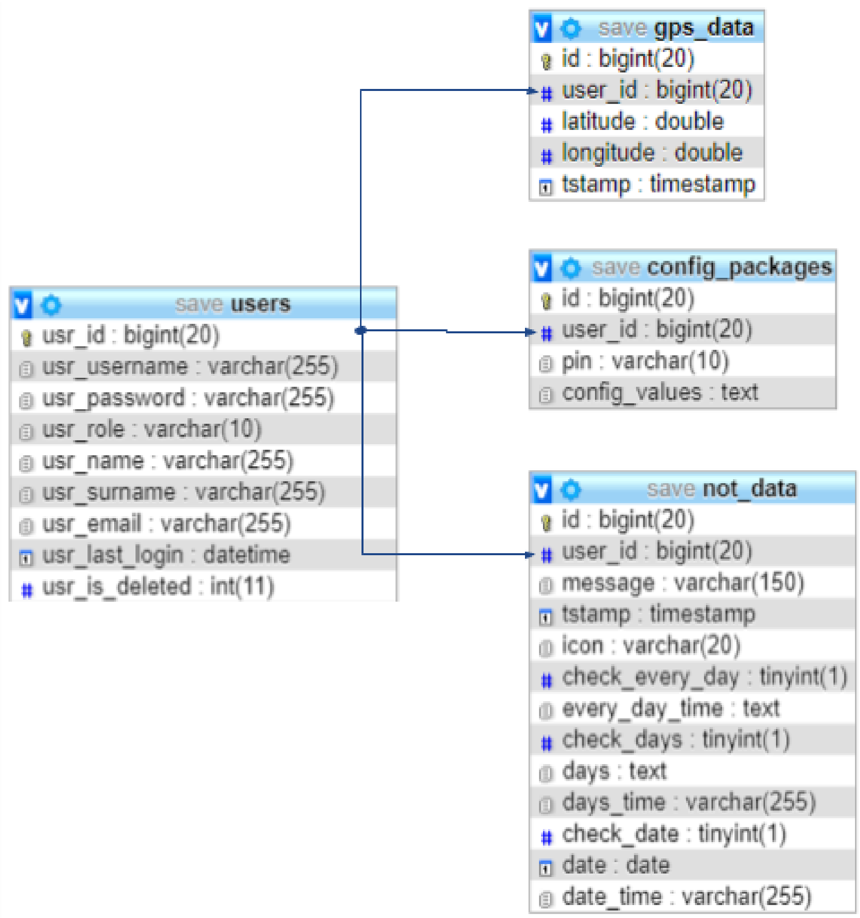


Figure 7 - Tables for the users, notifications, smartwatch app configuration, GPS data

Figure 7 presents the structure of users table and several other tables used by the microservices that implement the user interfaces and the link to the smartwatch applications. The password is not kept in clear text, but only the SHA-1 hash is stored.

The *not_data* table stores the scheduled notifications of the end-users. These are transferred periodically (every 10 minutes or soon as the smartwatch connects to the SAVE cloud application) to the smartwatch. The SAVE web application user interface manages these records.

It must be highlighted that the data coming from the sensors are not linked directly to the end-users, but to the kits. Accessing only this data does not reveal anything about the users' identities.

The temporary GPS data is collected in the *gps_data* table. The maximum duration for which the records are kept is 10 days.

5. The Data collecting system - Data collector

One of the most important components of the SAVE system is the data collection service. This service is implemented as a microservice, to be easily scalable with the increase of number of sensors. The communication interface is a REST API (over HTTPS), that accepts JSON formatted data coming from devices that are registered inside SAVE's database.

The minimal structure of the JSON message accepted by the data collecting API is:

```
{
"kdDevId": <integer_value>,
"kdValue": <string_value>
}
```

When the sensors cannot provide the data in the accepted format, a data adapter must be implemented, so it will wrap the sensor's native format (binary, text, JSON, XML, etc) into the accepted structure. The eHealth, Mood rings and CRT devices have a data adapter running inside the SAVE cloud application that intermediates between the devices and the *Data collecting system* (*Data collector*).

As illustrated in figure 8, the devices connect via the HTTPS protocol to a device type-specific adapter interface that transmits the data to the collection system, which will persist it in the database.

The data adapter interface is a software module that is developed specifically for each type of device recognised by the system. Its role is to expose a communication interface with external devices and to convert the format of the received data into a standard format recognized by the data collection component and to send it to the latter.

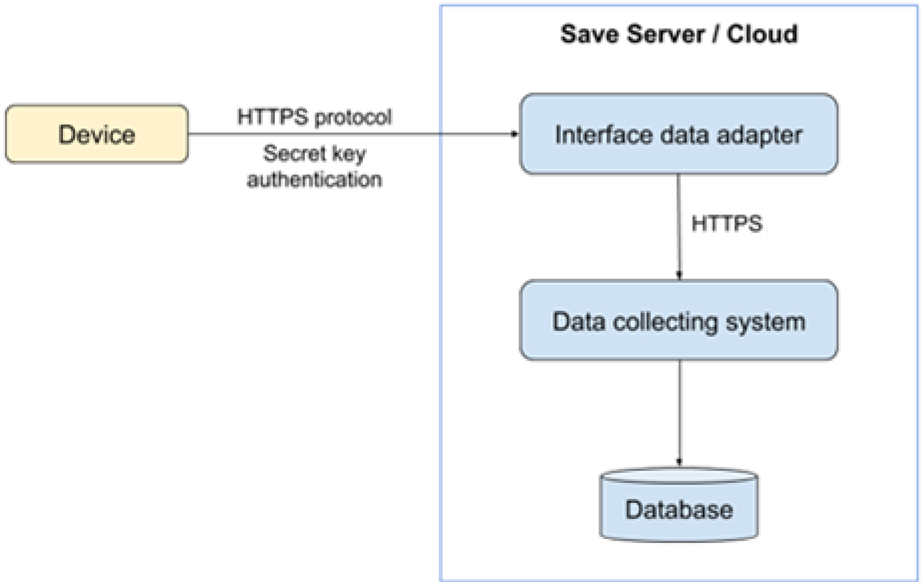


Figure 8 - System communication model

The adapter streamlines communication between devices and system, abstracting the data format and providing it to be persisted. In this sense, the adapter is specific to the type of device with which it interacts and can be developed using any technology; it must, however, maintain the standard communication with the central system.

The communication between the adapter and the *Data collector* is done via the HTTPS protocol, doubled by an authentication based on a secret key (API Key).

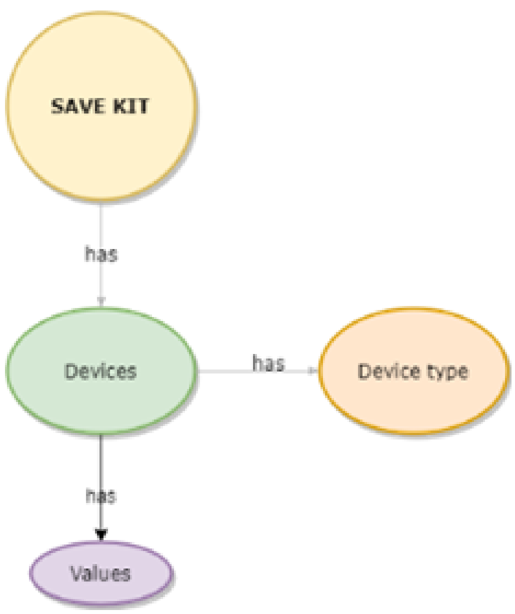


Figure 9 - Data organization in the SAVE system

The adapters are implemented as microservices, using the Java language and the *Spring Boot* framework.

Once in the system and transposed into a standard format, data from the devices are organised (figure 9) and stored.

A device must be registered in advance; it is associated with a device type (which defines the corresponding adaptation interface). This device will receive a name so that users can easily identify it.

The collected data are stored in tables into the database. To optimise the speed of data access, kits partition them, so that the data from a kit is stored in the same table. The data storage mode is a general one, allowing the saving of any data structure.

6. Security aspects

Regarding the security aspects, the SAVE solution uses the current standards to protect the data and the access to its features. Using HTTPS and the authentication by API key does the communication between the sensors and its adapter or the *Data collector*, and between the adapter interfaces and the *Data collector*.

The web applications (*SAVE Web* and *Admin Centre*) use authentication by the username-password pair, JWTs for authorization and work on HTTPS.

The data coming from the sensors are not linked in the database directly to the end-users, but to the kits. Accessing only this data does not reveal anything about the users' identities.

The passwords are not kept in clear text, but only the SHA-1 hash is stored in the users' table.

The SAVE database is designed keeping in mind the GDPR constraints, so the collected data is kept in an anonymized fashion, separating the identity of the end-user from their data. This database design allows the data to be processed without knowing the actual identity of the user it belongs to. Also, data from different users are isolated (through custom partitioning), so even if one of the end-user accounts is compromised, the attacker will not gain access to others' user data. Furthermore, the user accounts are not directly linked to the users' details, so these can be used in processing without knowing/accessing the identity of the user. If an end-user decides to opt-out of the SAVE solution, their account and collected data is deleted from the system and the backups.

7. Deployment environment

The SAVE cloud application, being design on the microservices architecture and built using the *Spring Boot* framework, can be deployed o vast number of infrastructures. The microservices can run independently of a servlet engine/web server (e.g. Tomcat) or can be deployed as a classic Java web application. Also, the microservices can be deployed using the *Docker* containerization engine, *Kubernetes* being employed for the orchestration.

8. SMS Notification System

Following the preliminary testing of the SAVE system, arose the need to notify caregiver-type users through the GSM short message service (SMS) in case of certain events. The notifications considered useful in the context of the SAVE project are as follows:

- Notification in case of flooding.
- Notification if the entrance door is left open (door remains open for more than 10 minutes).

This generation of notifications requires checking the most recent data from all flood and contact sensors. To avoid the traversal of all data tables in the database to generate notifications, a new table called *crt_data* (figure 10) was introduced, which stores the most recent data concerning flood (leak) and contact sensors for all users. Data insertion into the new table is done concurrently with insertion into the tables associated with the SAVE kits, but asynchronously, so as not to significantly impact the response time of the data collection API.

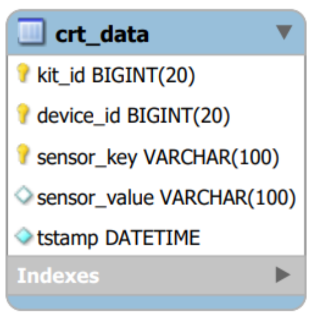


Figure 10 - crt_data table

A schedule-type service has been developed to generate notifications based on the data from the *crt_data* table. The generated notifications are stored in a waiting queue (implemented as a table in the database - figure 11) to be transmitted through the GSM network.

The SMS messages are sent through a GSM modem. Communication with the modem is achieved via a serial port, using AT commands.

A Windows service has been developed, which periodically identifies new messages from the *sms_notifications* table and attempts to send them through the GSM modem.

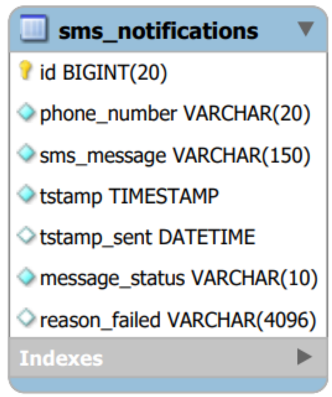


Figure 11 - sms_notifications table

9. SAVE Sensor Adapter

The solution adopted within the SAVE project for collecting and analysing user data consists of the following components:

- Smartwatch
- Data capturing sensors
- Web-based graphical interface for displaying and analysing collected data.

Sensors have been acquired for data collection (figure 11). These are:

- Flood detection sensor (2 pieces - for the kitchen, bathroom)
- Presence (human) sensor (2 pieces)
- Contact sensor (1 piece - for the entrance door)
- Centralization device (1 piece)

Figure 12 - Sensors included in the SAVE solution piloting

To collect data from sensors for the purpose of connecting them to the acquired devices, a "sensor adapter" was developed and designed by the partner VS.

The purpose of the *SAVE Sensors Adapter* is to read, indirectly, the status of the sensors included in the provided sensor kit. The *Aqara Hub 2* is capable of relaying events through IR (infrared) codes. The *SAVE Sensors Adapter* reads the IR codes emitted by the *Aqara Hub* and sends them to the *SAVE cloud application*, through Wi-Fi. The configuration of the Wi-Fi link is done using WPS (Wireless Protected Setup).

9.1. Components

The SAVE *Sensors Adapter* is built around an ESP32 development board, readily available on the market (the NodeMCU 32S and LOLIN32 variants)(figure 13). It also includes:

- a monochrome OLED graphic display, with I2C communication
- 2 push-buttons are used for the user interface
- the VS1838B IR sensor is employed to read the IR codes

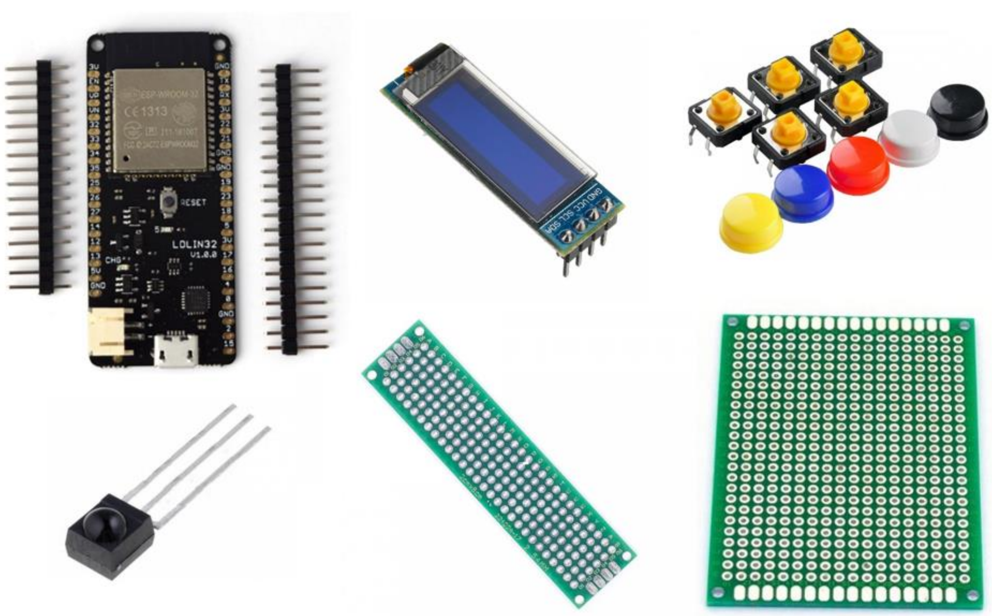


Figure 13 - SAVE Sensor Adapter components

9.2. The case

The case is built from black and transparent plexiglass sheets, 3mm thick, for maximum sturdiness (figure 14). The case is held together by M3 screws.

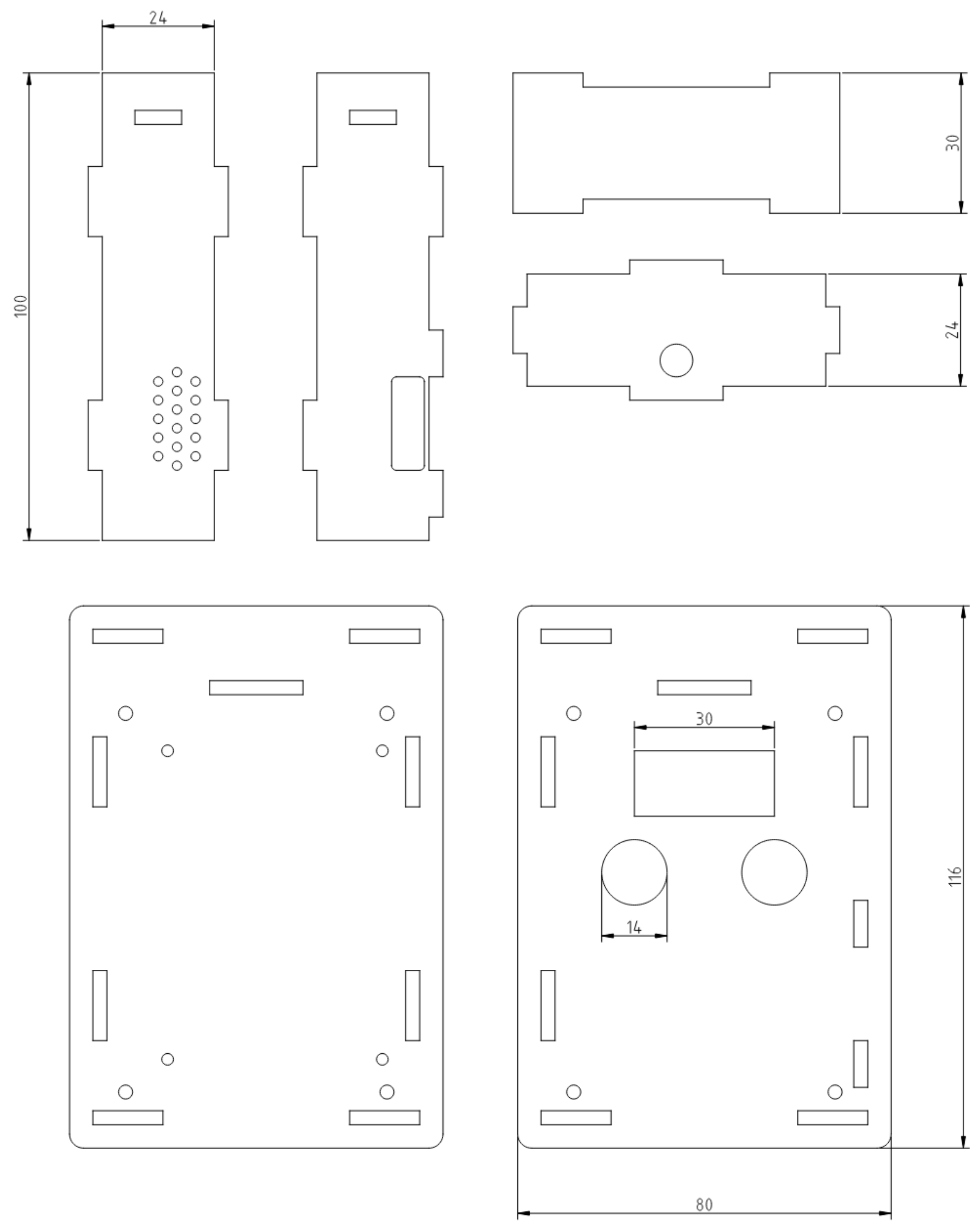


Figure 14 - SAVE Sensor Adapter case - main dimension of the case

Figure 15 shows the physical aspect of the case parts.

Figure 15 - SAVE Sensor Adapter case parts

9.3. Electrical schematic

Figure 16 shows the electrical schematic:

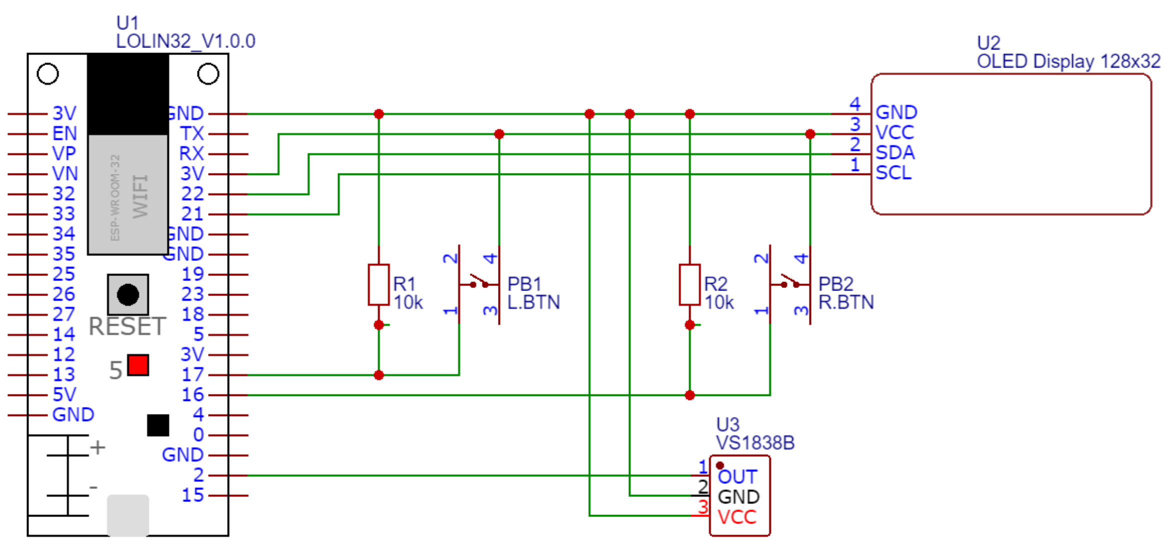


Figure 16 - SAVE Sensor Adapter schematic

The *SAVE Sensor Adapter* is powered from the USB port of the *Aqara Hub 2*.

Figure 17 presents the electronics of the device.

Figure 17 - SAVE Sensors Adapter – electronics

9.4. Software

The *SAVE Sensor Adapter* is programmed using Arduino and runs over Free RTOS OS. The software application makes use of the dual-core feature of the ESP32 microcontroller, separating the collection of data (through IR) from the user interface (figure 18).

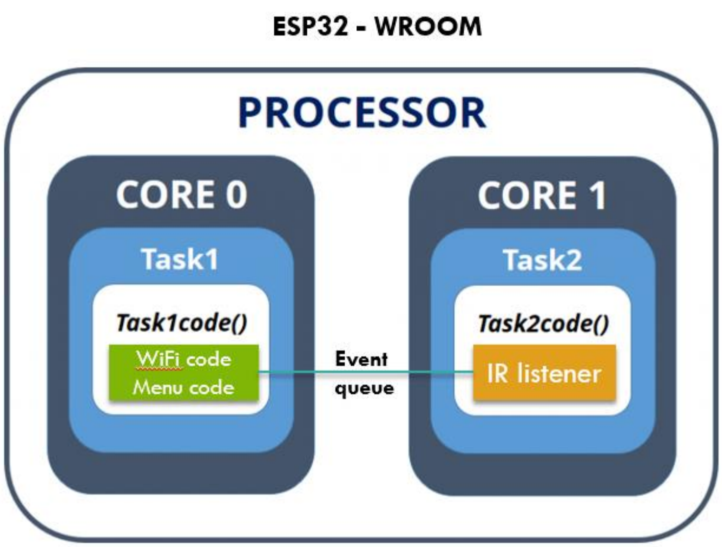


Figure 18 - Microcontroller software organization

9.5. The menu

The SAVE *Sensor Adapter* offers a user interface based on 2 buttons (left and right). The left button chooses an option, and the right button executes or confirms.

Figure 19 presents the structure of the user interface, as a menu. The blue screens only report information, the yellow ones allow the user to change the settings of the device.

The *Native ID* is the unique code by which the device is recognized by the SAVE solution.

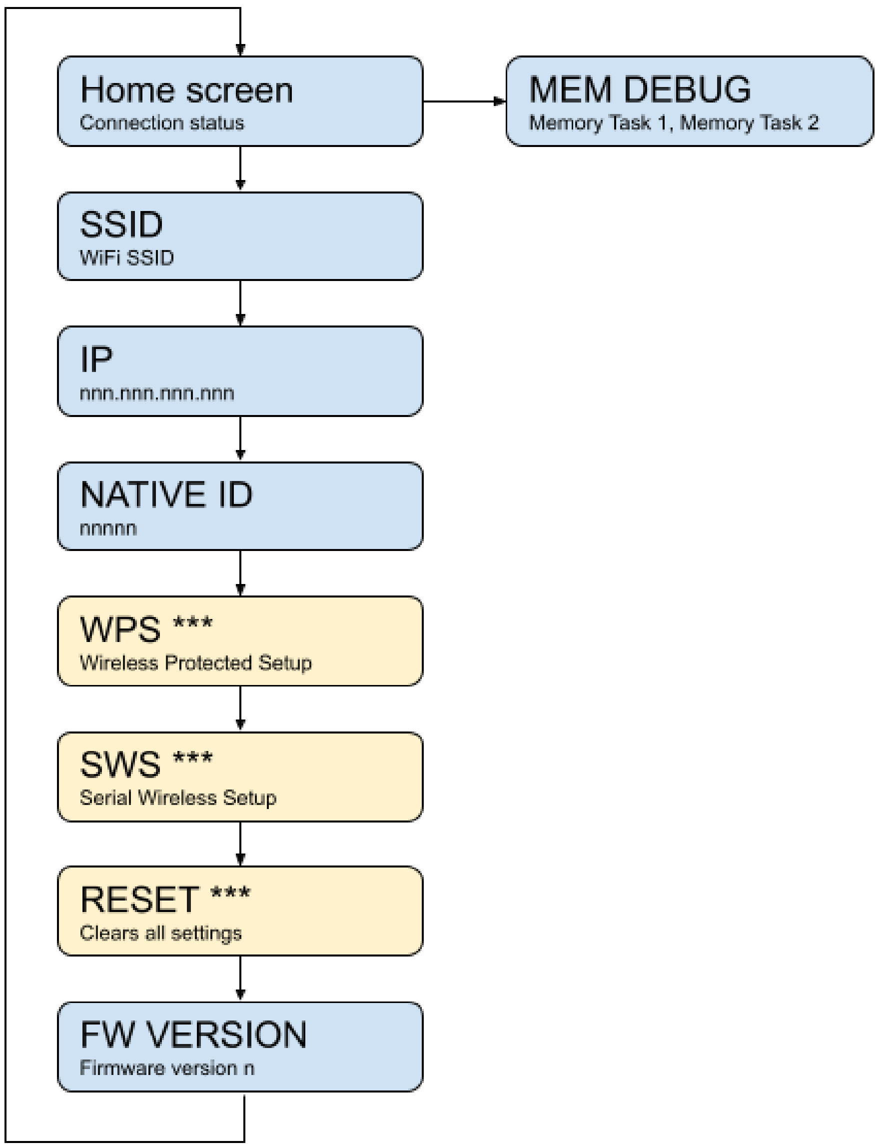


Figure 19 - Menu structure for the user interface

The WiFi credentials can be set by WPS or by using an USB connection to a computer running the SAVE *SWS* (*Serial Wireless Setup*) software, developed as a Windows application especially for this device (figure 20).

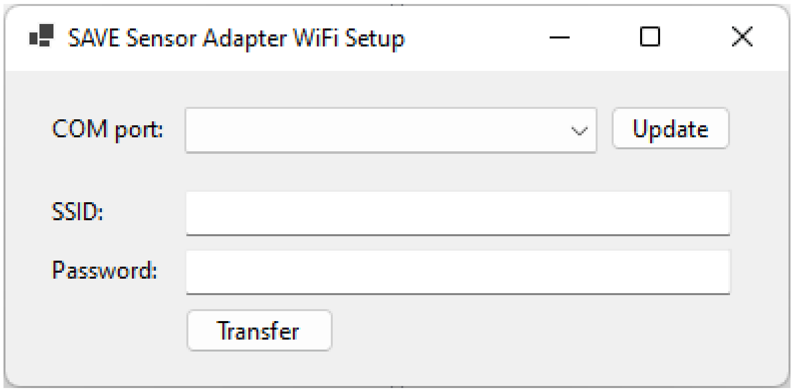


Figure 20 - SAVE Sensors Adapter Configuration application

The *Reset* option erases all configuration data of the device (i.e. WiFi credentials).

9.6. Final version of the device

Figure 21 shows the *SAVE Sensor Adapter* final device.

Figure 21 - SAVE Sensors Adapter

10. eHealth System

10.1. eHealth System functionality

The eHealth Monitoring System (figure 22) has the scope to acquire biometric sensors data by different wireless and wired eHealth sensors, to automatically collect data on-site and transmit the data via Wi-Fi to a router modem in the area of a LAN (Local Area Network) and further via Internet to the cloud server of the project.

The eHealth Monitoring system is using wireless and wired interfaces for sensor data acquisition (in current version: temperature, oxygen saturation, blood pressure, spirometry) together with a cloud-based platform and a TFT display for both remote and on-site data monitoring.

Figure 22 - The eHealth Monitoring System

10.2. eHealth System architecture

As settled at the level of technology, for complying with one of the main requirements of the European AAL perspectives in terms of interoperability and open interfaces for achieving a European market, the SoI concept (figure 23) is oriented towards Open-Source Hardware (OSHW) and COTS eHealth Platforms including COTS biosensors. The decision was also made considering other criteria, such as lowering costs and increasing scalability and interoperability.

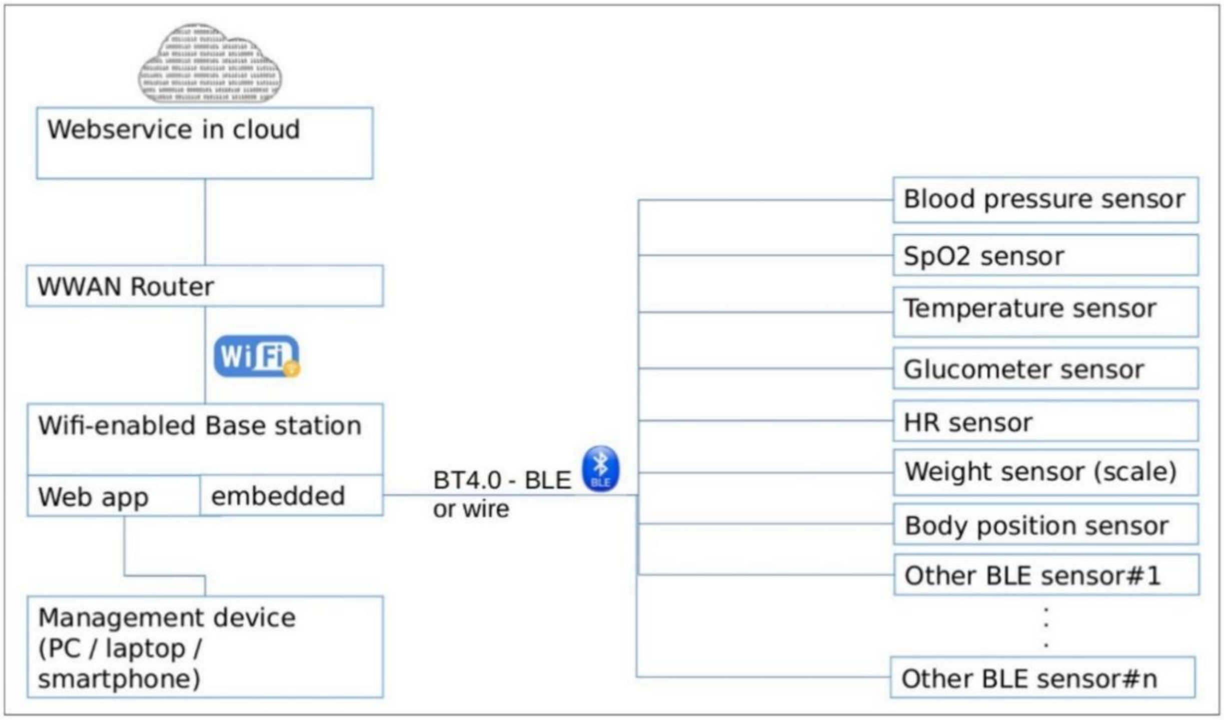


Figure 23 - eHealth System-of-Interest (SoI) conceptual perspective

By identifying and decomposing primary and secondary value functions of the eHealth SoI together with the elements of form that sustain them, the system architecture is emerging via Object-Process Methodology (OPM), where the structural, functional, and behavioural views in a single coherent architecture (figure 24) are modelled. Hence, in the figure below, an Object-Process Diagram (OPD) for the eHealth SoI is shown where the specific relation of elements of form in terms of building blocks are linked to the processes and operands for achieving the intended system emergence and functions, as well notifying the system boundary, accompanying systems, and interfaces.

Based on the eHealth system architecture OPD view, the emergence, as the desired goal, suggests the primary value functions as automatically collecting biometric data in a non-obtrusive approach (e.g., unique physiological, physical, behavioural data) from elders at their dwelling place and further transmitting the data to specific caregivers/volunteers according to communication protocols established for the proper decisions of elders.

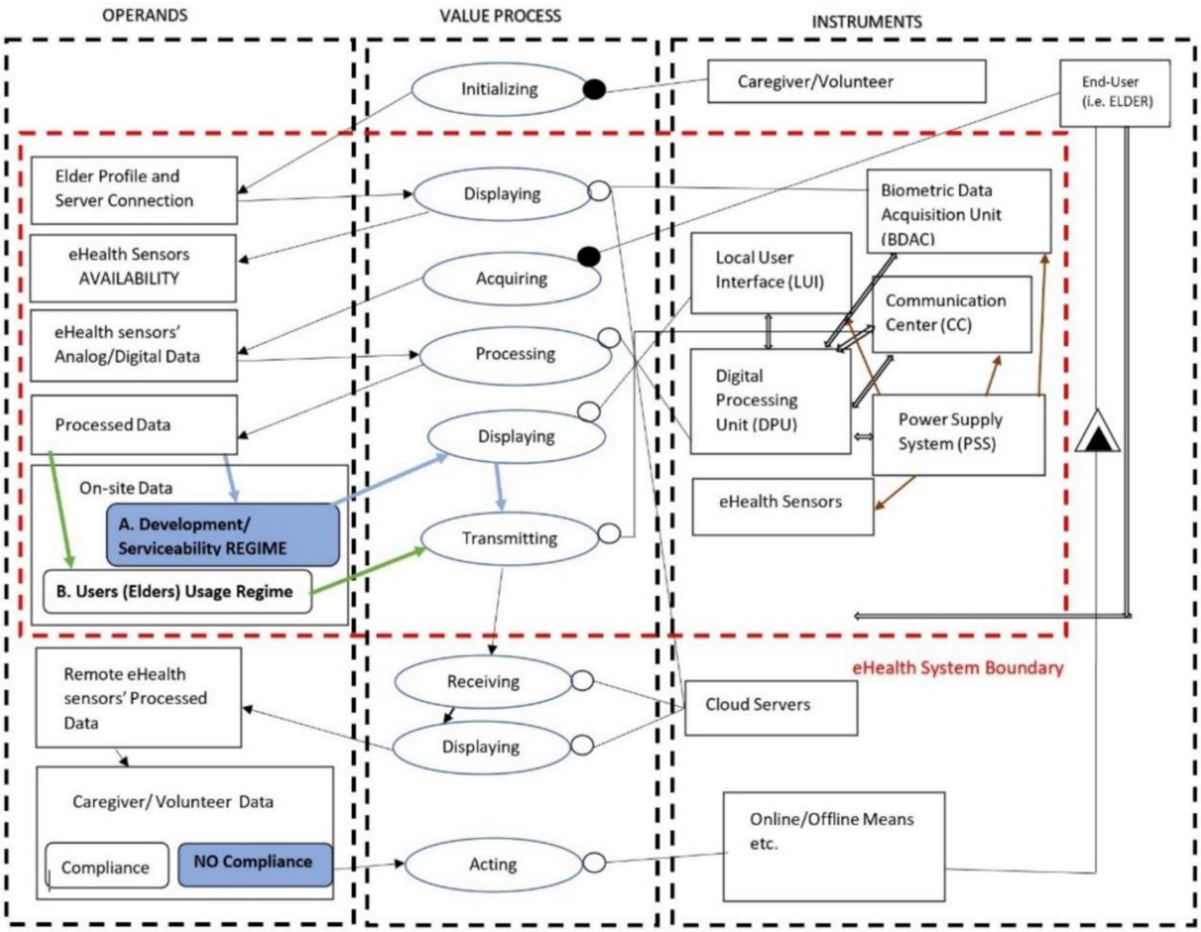


Figure 24 - Object-Process Diagram (OPD) for the eHealth system architecture

Having an in-depth analysis on system architecture OPD, the value pathway starts when the system is initialized by the second target user (volunteer/caregiver) with the elder's profile and server network connections of the elder's dwelling-place. Further, when the elder is properly using the system, there is the display function for elders exclusively in terms of eHealth sensors availability, the sensors being grouped in a list on the screen. The elder is able to acquire biometric data, the data are processed by the system, and two additional scenarios can occur in parallel:

- In the first regime of system development and serviceability, the biometric data are displayed and further transmitted via the internet exclusively to Information and Communications Technology (ICT) development teams. In the second regime of daily usage by the elders, the biometric sensor data are not displayed on the unit screen, just transmitted to the internet to caregivers/volunteers outside the eHealth system boundary for specific decisions to be made.
- The second regime stems from co-design performed sessions, as required by stakeholders. Thus, the system may accommodate changing needs, fitting the scenario of internet absence and communication failure for elders if the two regimes will be combined.

Considering the form aspects, namely instruments along with structural relationships among them, as illustrated in OPD system architecture, the functional blocks identified and defined by specific system requirements are the following:

- Biometric Data Acquisition Unit, which must integrate COTS OSHW components for data acquisition in digital and/or analog format, data processing, and transmission using analog-to-digital converters, multiplexers, standard communication protocols, etc.
- eHealth Sensors that could be presented as a customizable biometric sensors package for the primary target group (i.e., elders). According to the system requirements, the considered biometric sensors are: Blood pressure sensor, pulse oximetry sensor, body position sensor, temperature sensor, glucometer, spirometer, body weight scale, and CRT sensor for cognitive assessment. The acquired data by the e-Health central unit from the sensors (intended to be used individually) are collected, processed, and automatically sent to the Cloud.
- Power Supply System (PSS), which must provide with power autonomy the eHealth system with the screen ON and all sensors in operating condition. The user must beable to charge the system while it is in use; the system must function in "always-on" mode and not enter in standby mode; the e-Health system is turned OFF using an ON/OFF button.
- Digital Processing Unit (DPU), which must communicate to transmit information between the Communication Center, Local User Interface, Biometric Data Acquisition System, and Power Supply blocks.
- Client–Server Communication Unit uses the network card Wi-Fi System-on-a-Chip (SoC) in order to send information to the Cloud server, and the system will communicate via Wi-Fi with the user's home wireless router. The system is intended to communicate with the server using an encrypted channel with HTTP protocol and secured with SSL/TLS certificates, to send data only when it has real values recorded from the sensors and at a configurable period of about 10 s.
- Local User Interface (LUI) must display locally the biometric sensors availability, if it has found at least one Bluetooth-enabled sensor and the sensor pairing. The system, according to the system architecture, will pursue the parallel functional regimes described above.

11. Wellbeing System

11.1. Wellbeing System functionality

The well-being system (figure 25) scope consists in assessing intraindividual variability across reaction time (RT) tasks performance and the corresponding galvanic skin response (GSR).

Intraindividual variability, according to literature, is considered a risk factor predictive of successful ageing, implicitly well-being, and it is significant in assessing individuals whose disorders are mild.

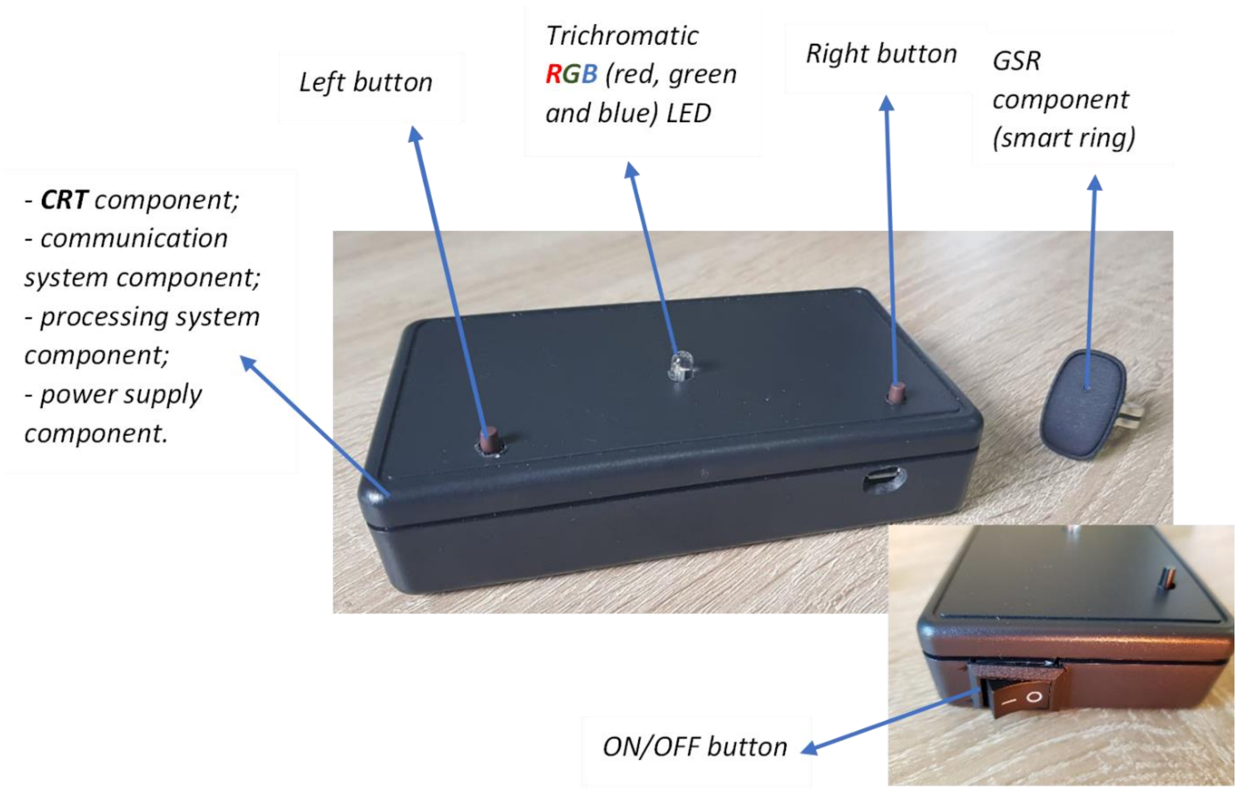


Figure 25 - Wellbeing system

The proposed well-being assessment system consists of two major components, as Choice Reaction Time (CRT) and Galvanic Skin Response (GSR) system components.

Visual choice reaction time (CRT) tasks performance has been widely analysed to measure age-related declines in processing speed.

The Well-being system is using wireless and built-in sensor data acquisition, together with a cloud-based platform for both remote and on-site data monitoring.

Choice Reaction Time (CRT) methodology is based on several visual stimuli and two response buttons. In the Figure 1, a trichromatic RGB (red, green, and blue) light-emitting diode (LED) is emitting a stimulus with respect to the following procedure:

- The blue LED color represents the target, and when the blue stimulus lights up, the elder has to discriminate, select, and execute the right button (R);
- The red or green LED colors represent distractors and when the red or green stimulus lights up, the elder has to discriminate, select, and execute the left button (L).

Stress assessment Technologies are based on wearable devices measuring galvanic skin response (GSR) in order to evaluate specific changes and detect stress level and phase (excitement, stress and recovery). Stress assessment Technologies give a way to find a balance between work and free time in order to enhance the well-being state of the person.

The elder wears the stress assessment device, a ring in our case, that analyses the evolution of his stress level. The elder can be advised to stop working, relax and/or breathe in order to diminish stress level and thus, to enhance its well-being state.

Stress Assessment Services evaluate specific changes of stress level and/or stress phase (excitement, stress and recovery) and usually store the data in a cloud for further offline assessments.

The Moodmetric smart ring, see Figure 26, was developed for measuring electrodermal activity accurately in a convenient, wearable form able to provide data in real time.

Figure 26 - Moodmetric smart ring

The Moodmetric measurement is related to the the sympathetic nervous system activation – the fight-or-flight response. High activation indicates positive or negative stress, low in turn relaxation and calmness. The Moodmetric measurement is intended for a long term and continuous follow-up on stress and recovery levels and shows clearly the impact of emotional stress on the overall load.

The Moodmetric ring is not a medical device and should not be used to diagnose or treat any medical conditions. The device has an internal, non-removable, rechargeable Li-Ion battery. Normal device operating temperature is between -20º and 35º C / (-4º to 95º F). Charging temperature is between 5ºC and 35ºC (40º to 95º F).

A green indicator LED should blink once when inserting to finger: device is now working and recording and the battery is charged. When the device is removed from finger or when the contact with the skin is lost, the green indicator LED blinks twice. When the battery is low, the red indicator LED starts to blink periodically. If worn with a low battery for extended periods, corruption of recorded data may occur. When the battery is empty, the red indicator LED blinks twice and the ring will shut down. Charge before using. When the charge plug is inserted, an orange light will appear until battery is full. Take the charger off when the orange light shuts off. If the orange light does not light up when the power plug is inserted and the plug is powered, this means the battery is already full and no need to charge, or the temperature is outside the allowed charging temperature window 5ºC to 35ºC (40º to 95º F).

Moodmetric measures your alertness level. A simple reading tells whether you are experiencing stress/excitement or being calm. Note that the measurement can't tell whether your emotion positive or negative, only the intensity of it. The measurement is shown on a scale from 1 to 100, named the MM Level (the Moodmetric level). On the app´s main screen this number displays the stress level at a precise moment, and it updated continuously.

The MM level describes the load you are experiencing. Low electrodermal activity (EDA) means that your mind is at peace. The more intense the feeling, the higher the EDA. The Moodmetric smart ring captures this biosignal and shows the stress/excitement level with the Moodmetric number. The Moodmetric level ranges from 1 to 100. The in-built algorithm learns from the user and gives 100 to the highest experienced load and 1 to lowest. The MM level 100 means being extremely stressed, excited, anxious or frightened. At about 50 the mind is active while below 30 means being relaxed. Level 1 can be reached e.g. at deep sleep.

11.2 Wellbeing System architecture

Following the design and interaction with users, new functional requirements have emerged for improving the Wellbeing system.

a) The following two operating modes exist:
   - regime 1 (UPGRADE version for research purposes) for scientific research with large-scale data extraction: the CRT subsystem should provide response time for each button press, along with the correct/incorrect verdict. In the case of an incorrect press, the reaction time until the correct button press should be monitored. The GSR subsystem should continuously monitor electrodermal activity (3 samples per minute), particularly during CRT activity.
   - regime 2 (EXISTING version for caregiver information) for monitoring the elderly by tutors/volunteers on various cloud and mobile platforms using data extracted from the existing version (without upgrade): the CRT subsystem should provide the number of incorrect and correct button presses, minimum and maximum response times, average time, and standard deviation. The GSR subsystem should provide an epidermal stress score.

b) Both operating modes should be able to function simultaneously as follows:
   - Regime 1 (UPGRADE version) needs to be adapted to the specific requirements of the Technology Club pilot, displaying on-site data through a USB connection to a PC within the Technology Club infrastructure. The displayed data should be exclusively intended for the research team.
   - Regime 2 (EXISTING version) operates with remote data display through the cloud or a mobile web application for tutors/volunteers.

In accordance with Model Based Systems Engineering (MBSE), the architecture of the Wellbeing system was developed using the Object Process Methodology (OPM) methodology, employing the specific software OPCAT. Thus, the system diagram, which represents the highest level of architecture, is described in the figure 27:

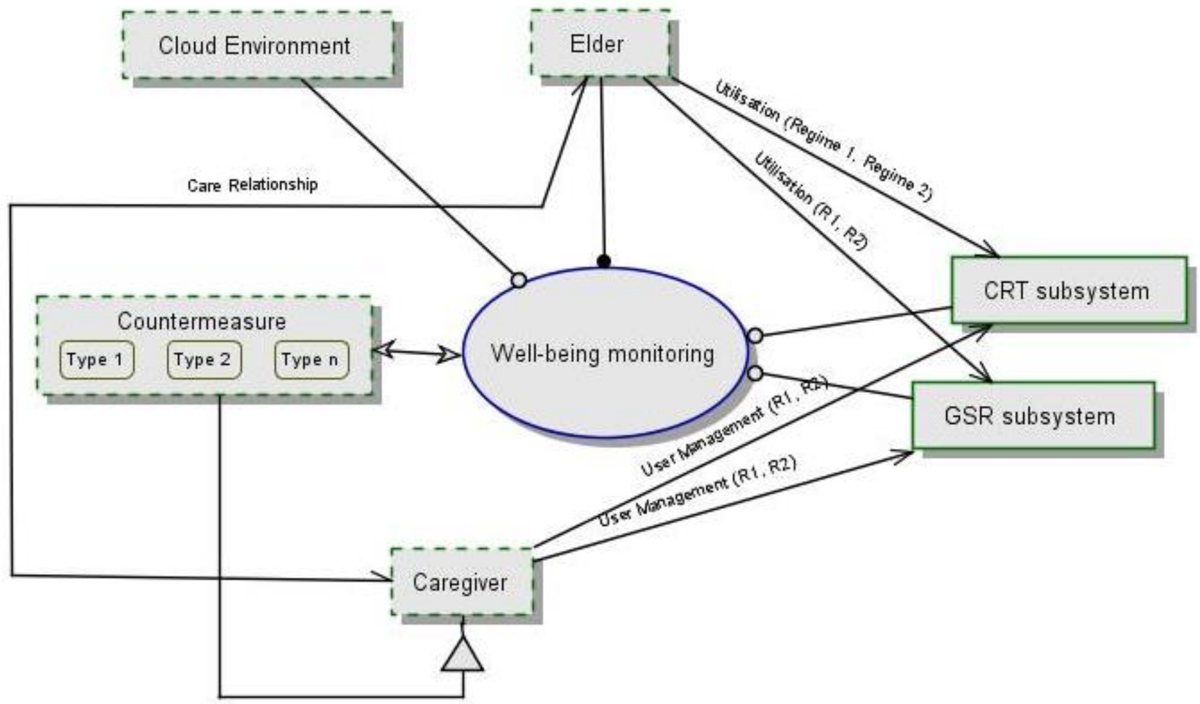


Figure 27 - System Diagram (SD) – well-being system

The specifications for the SD - Wellbeing, according to the Object Process Language, are as follows:

CRT subsystem is physical.
GSR subsystem is physical.
Cloud Environment is environmental and physical.
Elder is environmental and physical.
Elder Utilisation (Regime 1, Regime 2) CRT subsystem.
Elder Utilisation (R1, R2) GSR subsystem.
Elder and Caregiver are Care Relationship.
Elder handles Well-being monitoring.
Caregiver is environmental and physical.
Caregiver User Management (R1, R2) CRT subsystem.
Caregiver User Management (R1, R2) GSR subsystem.
Countermeasure is environmental and physical.
Countermeasure is a Caregiver.
Countermeasure can be Type 1 by default , Type 2, or Type n.
Well-being monitoring is physical.
Well-being monitoring requires Cloud Environment, GSR subsystem, and CRT subsystem.
Well-being monitoring affects Countermeasure.

The Wellbeing system diagram at level 1.1, as shown in figure 28, provides a detailed overview of the Wellbeing monitoring process:

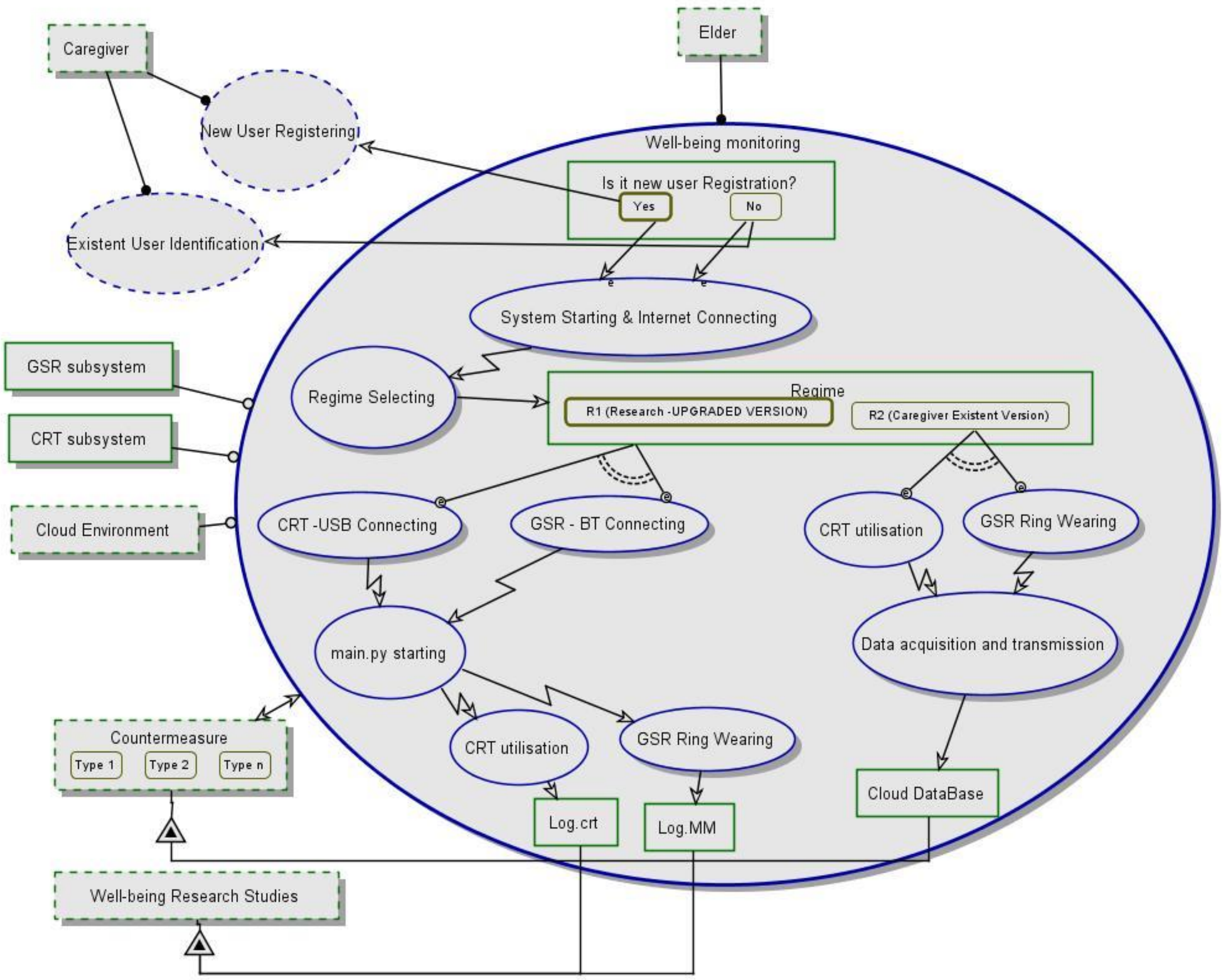


Figure 28 - System Diagram (SD) 1.1 – well-being system

Specifications SD 1.1 – well-being, according to Object Process Language are:

CRT subsystem is physical.
GSR subsystem is physical.
Cloud Environment is environmental and physical.
Elder is environmental and physical.
Elder handles Well-being monitoring.
Countermeasure is environmental and physical.
Countermeasure can be Type 1 by default , Type 2, or Type n.
Countermeasure exhibits Cloud DataBase.
Caregiver is environmental and physical.
Caregiver handles Existent User Identification and New User Registering.

Well-being Research Studies is environmental and physical.
Well-being Research Studies exhibits Log.crt and Log.MM.
New User Registering is environmental.
New User Registering consumes Yes Is it new user Registration?.
Existent User Identification is environmental.
Existent User Identification consumes No Is it new user Registration?.
Well-being monitoring is physical.
Well-being monitoring exhibits Is it new user Registration?, Regime, Log.crt, Log.MM, and Cloud DataBase.
Well-being monitoring consists of System Starting & Internet Connecting, Regime Selecting, CRT -USB Connecting, GSR - BT Connecting, main.py starting, CRT utilisation, GSR Ring Wearing, and Data acquisition and transmission.
Well-being monitoring requires Cloud Environment, GSR subsystem, and CRT subsystem.
Well-being monitoring affects Countermeasure.
Well-being monitoring zooms into System Starting & Internet Connecting, Regime Selecting, CRT -USB Connecting, GSR - BT Connecting, Data acquisition and transmission, main.py starting, GSR Ring Wearing, GSR Ring Wearing, CRT utilisation, and CRT utilisation, as well as Cloud DataBase, Log.MM, Log.crt, Regime, and Is it new user Registration?.

Regime can be R1 (Research -UPGRADED VERSION) or R2 (Caregiver Existent Version).

R1 (Research -UPGRADED VERSION) is initial.

Regime triggers GSR - BT Connecting or CRT -USB Connecting when its state changes.

Regime triggers GSR Ring Wearing or CRT utilisation when it enters R2 (Caregiver Existent Version).

Is it new user Registration? can be Yes or No.

Yes is initial.

Is it new user Registration? triggers System Starting & Internet Connecting when it enters Yes.

Is it new user Registration? triggers System Starting & Internet Connecting when it enters No.

System Starting & Internet Connecting is physical.

System Starting & Internet Connecting consumes No Is it new user Registration? and Yes Is it new user Registration?.

System Starting & Internet Connecting invokes Regime Selecting.

Regime Selecting is physical.

Regime Selecting yields Regime.

CRT -USB Connecting is physical.

CRT -USB Connecting requires Regime.

CRT -USB Connecting invokes main.py starting.

GSR - BT Connecting is physical.

GSR - BT Connecting requires Regime.

GSR - BT Connecting invokes main.py starting.

Data acquisition and transmission is physical.

Data acquisition and transmission yields Cloud DataBase.

main.py starting invokes GSR Ring Wearing and CRT utilisation.

GSR Ring Wearing is physical.

GSR Ring Wearing requires R2 (Caregiver Existent Version) Regime.

GSR Ring Wearing yields Log.MM.

GSR Ring Wearing invokes Data acquisition and transmission.

CRT utilisation requires R2 (Caregiver Existent Version) Regime.

CRT utilisation yields Log.crt.

CRT utilisation invokes Data acquisition and transmission.

12. End-user SAVE Kit package content

Each end-user will receive a SAVE kit which contains the following:

- Aqara Smart Home Starter Kit
- Sensor Adapter (to be connected to the Aqara Hub 2)
- Samsung Galaxy Watch3
- Android-based Smartphone (optional)
- A card containing Kit key (KIT ID), Smartwatch ID (SW ID) and Sensor Adapter ID (SA ID) (optional – figure 29)

Figure 29 - SAVE kit card

13. Installation and configuration of the SAVE Kit (used in piloting)

For the piloting phase, the smartwatch app was delivered using the debug function of the smartwatches, so the installation procedure required a trained person with some technical knowledge. The application packages are signed using Samsung certificates. However, the app can be uploaded to the store app.

13.1. Preparing the environment

13.1.1. Tizen Studio

On a computer, open an Internet browser and type the following link: https://developer.tizen.org/ko/development/tizen-studio/download (figure 30)

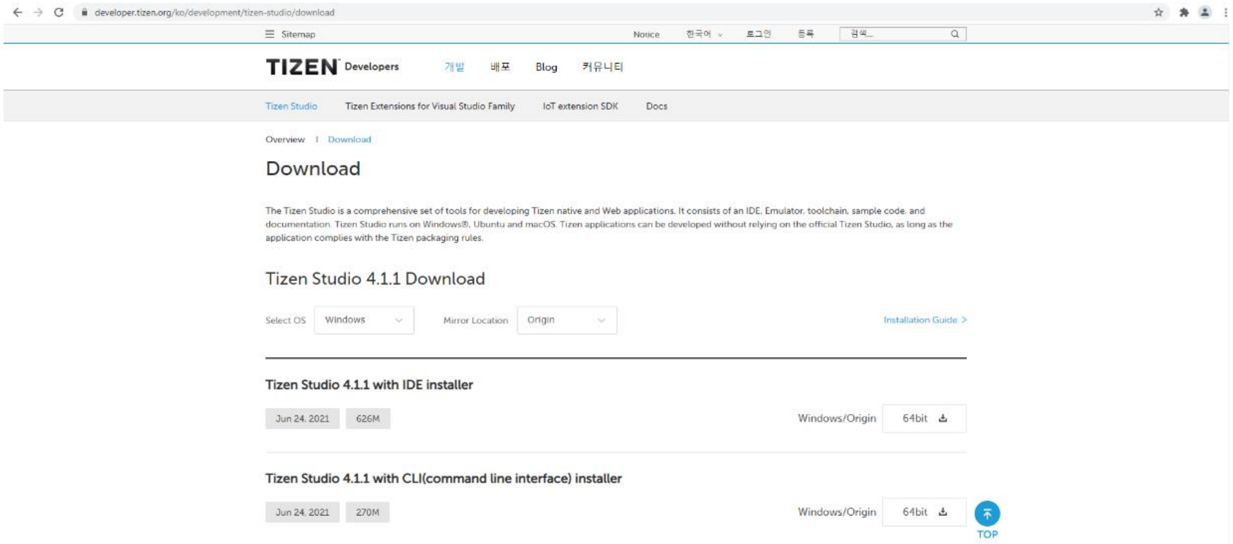


Figure 30 - Tizen Studio download

Download the ***CLI installer*** package by clicking on "Download" icon. After the download is finished, run the installer. Type "*y*" when being asked to read license agreement policy.

```
Loading... Please wait.

Installation information
        * License Agreement : false
        * Install directory : C:\tizen-studio, Validation [false]
        * Data directory    : C:\tizen-studio-data.1

Attention
Tizen Studio is required to agree with software license.
Do you want to read a license agreement policy? (Y/n) :  y
```

Figure 31 - Tizen Studio installation – license policy

Type "*y*" to agree with software license agreement.

```
Do you agree with software license agreement? (Y/n) :  y
```

Figure 32 - Tizen studio installation – license policy agreement

The default destination directory is "c:\tizen-studio". Type "*y*" to install to default directory or "*n*" to set a different one.

```
Destination directory : C:\tizen-studio
Default destination directory is (C:\tizen-studio)
Do you want to install to default directory? (Y/n) :  y
```

Figure 33 - Tizen Studio installation – default directory

If you chose to set another path, you are being asked to type the full path of the installation directory. Type it and hit "Enter".

```
Destination directory : C:\tizen-studio
Default destination directory is (C:\tizen-studio)
Do you want to install to default directory? (Y/n) :  n

Full path is required to set destination directory
Please type destination directory : c:\my-directory
```

Figure 34 - Tizen Studio installation – custom directory

A confirmation is required. Type "*y*" if the path is correct, "*n*" to retype it and "*q*" to cancel the installation.

```
Full path is required to set destination directory
Please type destination directory : c:\my-directory
You typed : c:\my-directory, is this correct? (Y/n) (quit:q) :
y
```

Figure 35 - Tizen Studio installation – custom directory confirmation

Once the path confirmed, the installation will begin. After the installation is completed, press "Enter" to close the window.

```
Installation has been completed!
Thank you for using Installer
Press enter to exit...
```

Figure 36 - Tizen Studio installation finished

11.1.2. Setting up Samsung Galaxy Watch3 device

Turn on the Samsung Galaxy Watch3 and once the home screen is displayed, tap on it.

Figure 37 - Galaxy Watch 3 home screen

On the resulted screen, tap on "Question mark" icon.

Figure 38 - Accessibility, Info and Language

On the next screen, instructions will be displayed. Tap on “here” link, right below the screen.

Figure 39 - Instructions

Next, information about what you will miss by not using smartwatch with your phone is displayed. You won’t get emails, calendar and music player support. Tap “Continue”.

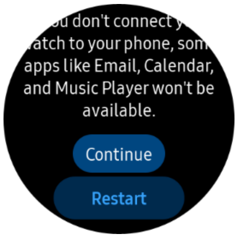


Figure 40 - Not available apps

The terms and conditions are displayed. Tap “Agree”.

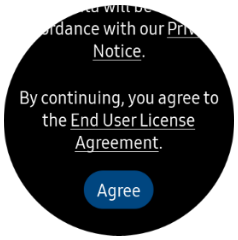


Figure 41 - End User License Agreement

On the next screen, agree with the requirements and tap “Next”.

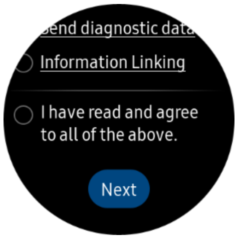


Figure 42 - Agreements

At this step you can log in to your Samsung account in case you want to backup and restore your saved data. It is optional, so tap “Skip”.

Figure 43 - Samsung account log in

Next, you can set up a mobile plan. This is also optional, so tap “Skip”.

Figure 44 - Mobile plan

Now, you are required to create a PIN. The PIN is useful when you will want to pair smartwatch with the smartphone. That time you will require this PIN. Tap “Next”, set the 4-digit PIN and remember it for future purposes.

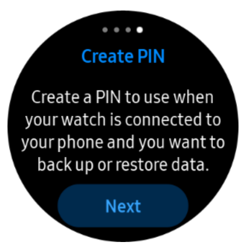


Figure 45 - PIN creation

13.1.3. Connect the Galaxy Watch3 to Wi-Fi network

Go to “Menu -> Settings -> Connections -> Wi-Fi -> Wi-Fi networks”. Scan and connect to the same network as your computer.

Figure 46 - Settings menu

Figure 47 - Connections menu

Figure 48 - Wi-Fi option

After the connection is successfully, tap on the network name you just connected, scroll to "IP address" and write it down because this IP is needed in the SAVE applications installation process.

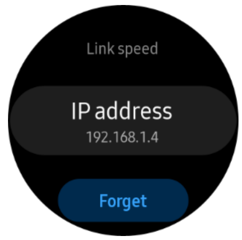


Figure 49 - IP address

13.1.4. Activating debugging option

Go to "Menu -> Settings -> About watch". Scroll to the bottom, tap on "Debugging" and set this option "On".

Figure 50 - About watch menu

Figure 51 - Debugging option

13.1.5. Get the smartwatch DUID code

After the Galaxy Watch3 was connected to the Wi-Fi network (it must be in the same network as your computer), the DUID (Device Unique Identifier) code must be obtained. The DUID must be included in the distributor certificate to be able to install the SAVE applications to the smartwatch.

On your computer, open a Command Prompt window, use "cd" command and set the directory to the "tools" directory where the Tizen Studio was installed (e.g. "cd C:\tizen-studio\tools").

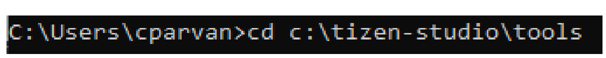


Figure 52 - Directory change

Now type "*sdb.exe connect IP*", IP = the IP your smartphone has in the network and that you write it down in a step earlier (e.g. 192.168.1.4).

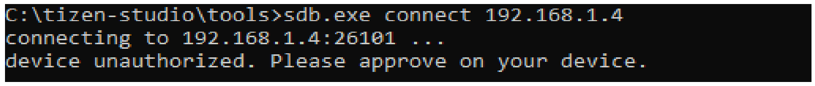


Figure 53 - Connecting to smartwatch

Check your smartwatch and approve the connection.

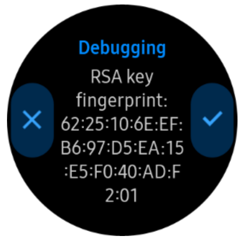


Figure 54 - Connection approving

In command window type "*sdb shell /opt/etc/duid-gadget vs*". The obtained code is sent to Vision Systems, who provide the kits for the smartwatch app.

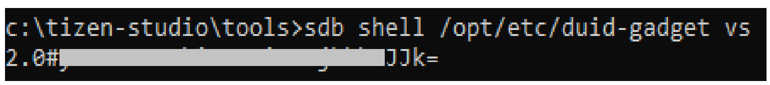


Figure 55 - Getting DUID code

13.1.6. Pairing Android smartphone with Samsung Galaxy Watch3

To get Samsung Galaxy Watch3 working with a compatible smartphone, it is required to follow some steps.

Make sure the Bluetooth connection is active on both smartphone and smartwatch.

On your smartphone open an application store (e.g., Google Play – Figure 56), then search and install "*Galaxy Wearable*".

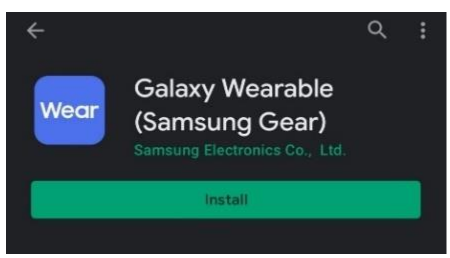


Figure 56 - Galaxy Wearable app

After the installation process is successfully, open the installed app and tap on "Start" button. A pop-up window will prompt you to allow *Galaxy Wearable* app to use your phone location as shown in Figure 30. Choose "While using the app" and the pairing process will continue.

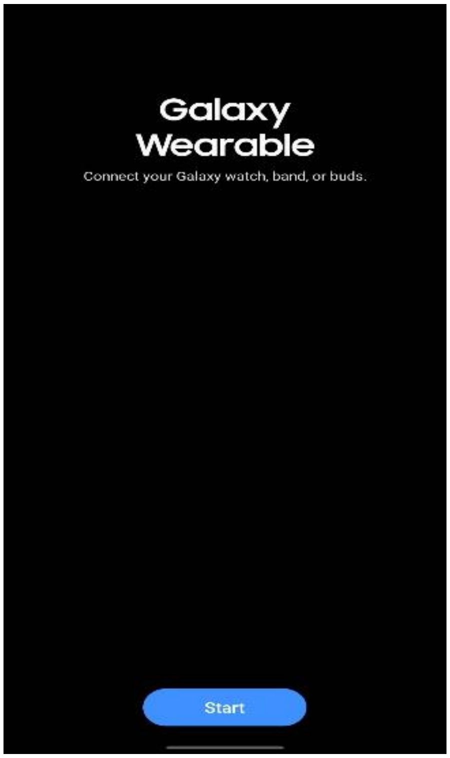


Figure 57 - Galaxy Wearable app

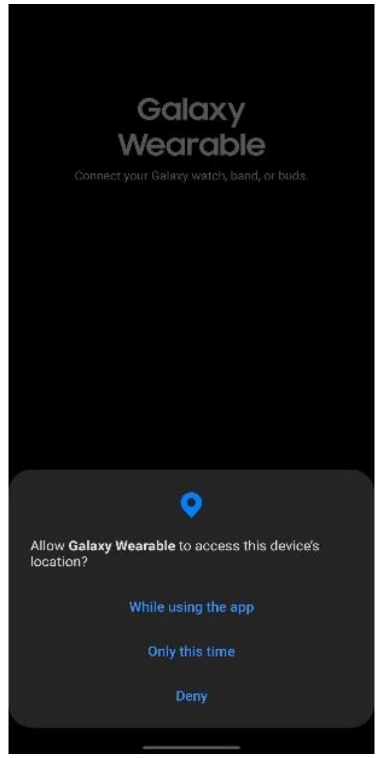


Figure 58 - Galaxy Wearable requesting location access

At this step, the application will scan for your Samsung Galaxy Watch3. Make sure the smartwatch is nearby your phone. After the scanning process is finished and your smartwatch name is visible in the list, tap on it. An example is shown in Figure 61.

If you set up your Galaxy Watch3 to use it without a phone, you will be asked to enter the PIN you set previously. After that, you will be asked if you want to keep using your personal data and content from the watch after the pairing process will be successfully.

Figure 59 - PIN code

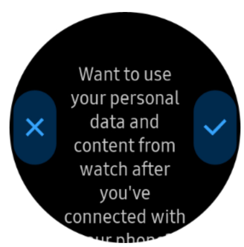


Figure 60 - Use personal data and content from watch

At this point, you will have to check both your smartphone and your smartwatch. A number will be displayed on both devices screen. If the numbers are the same, on your phone tap on the "Confirm" button (Figure 63) and on your Watch3 tap the orange icon located in the middle of the screen (Figure 62).

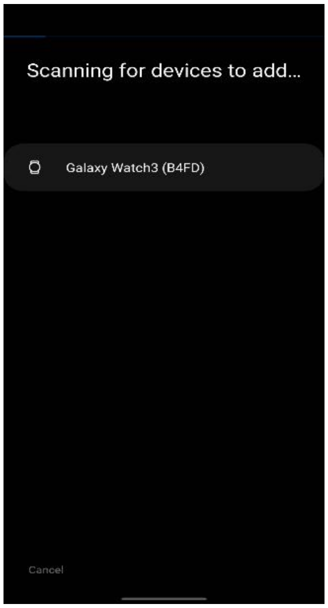


Figure 61 - Scanning for devices

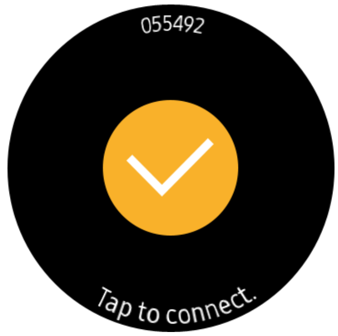


Figure 62 - Smartwatch pairing confirmation

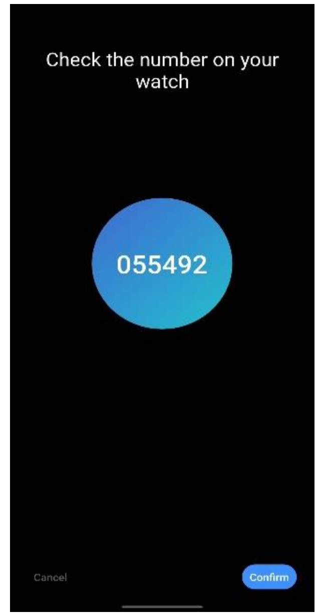


Figure 63 - Smartphone pairing confirmation

Required software is being downloaded and installed on your smartphone (*Galaxy Watch3 Plugin* and *Samsung Accessory Service*). It will take a few moments for the pairing process to finish.

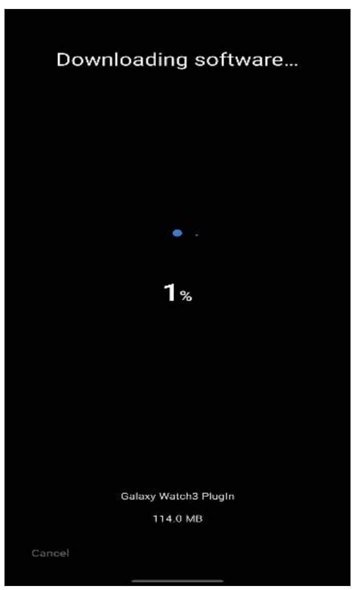


Figure 64 - Downloading required software

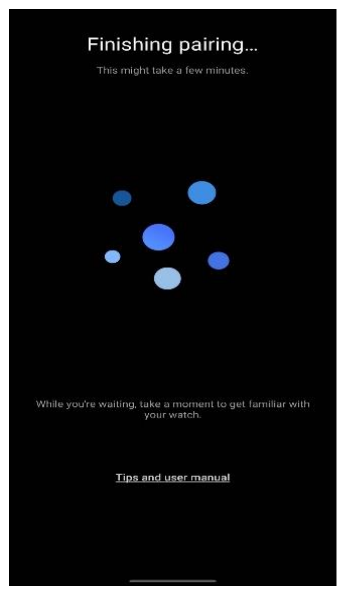


Figure 65 - Finishing pairing

Next, a window where you can see the Privacy Notices is displayed (Figure 38). You have the possibility to agree or not to the sending of diagnostic data and mapping technologies by Here. After tapping on "Continue" button you will be prompted to link "Galaxy Watch3 Plugin" with your Galaxy Watch. Tap "OK".

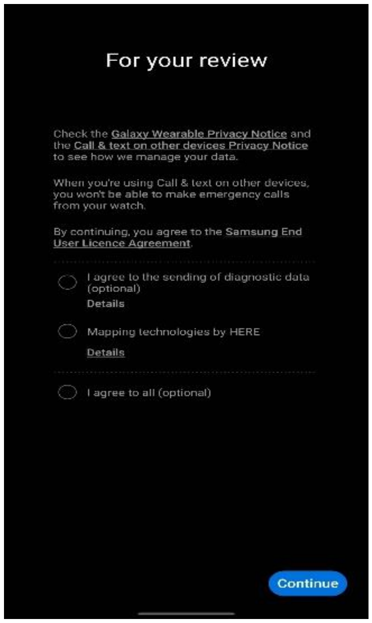


Figure 66 - Privacy Notice

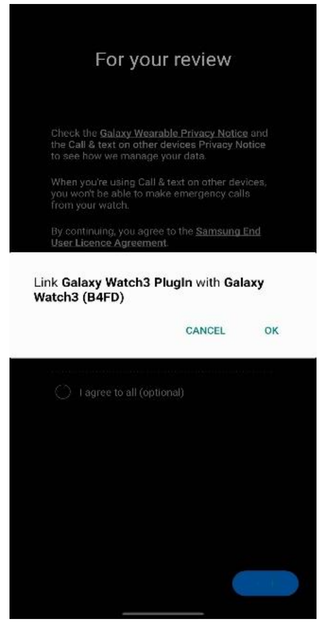


Figure 67 - Linking Galaxy Watch3 Plugin with smartwatch

You will get a screen where you will be prompted to allow *Galaxy Wearable* app access to Location all the time. Access must be granted because *Galaxy Wearable* application uses this permission to locate your phone with your watch. Tap “Allow” and you will be redirected to a new screen where you need to check “Allow all the time” option and then tap the “go back” icon (“<”) located top left.

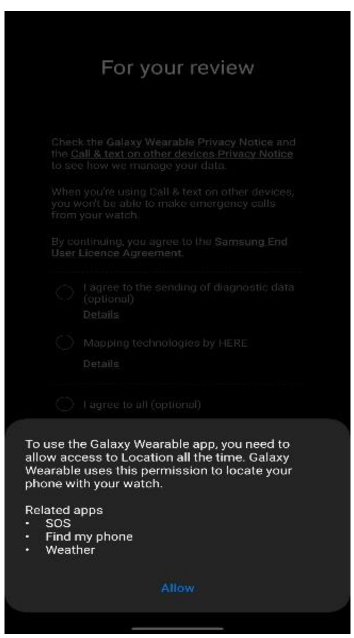


Figure 68 - Allowing Location access

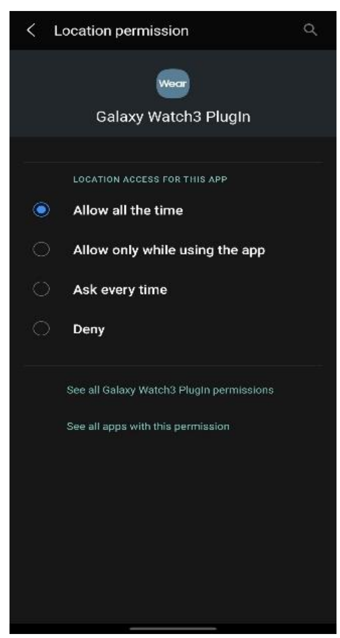


Figure 69 - Location permissions

In order to use all the available smartwatch functionalities (making phone calls, sending SMS, accessing your calendar, etc.) you will need to allow more permissions. The requested permissions are shown in figures from figure 63 to figure 68.

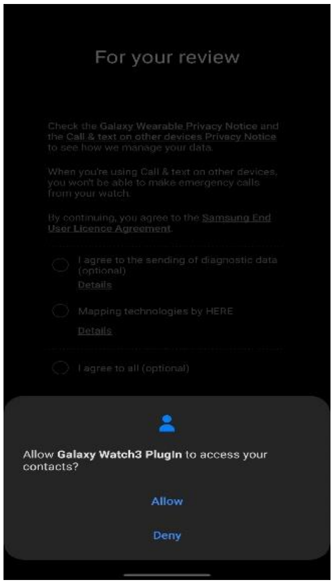


Figure 70 - Contacts permission

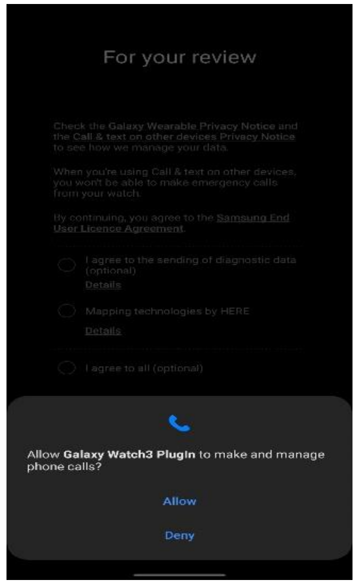


Figure 71 - Phone calls permission

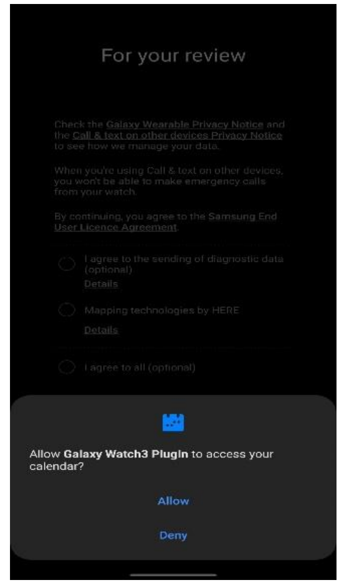


Figure 72 - Calendar permission

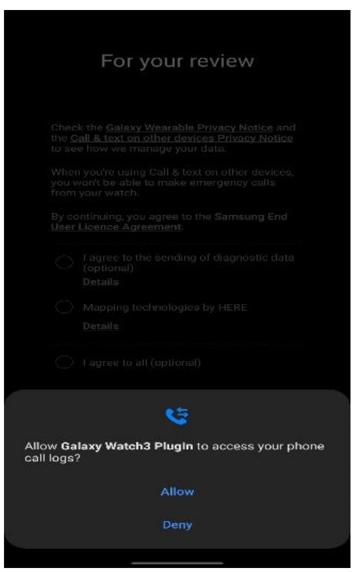


Figure 73 - Call logs permission

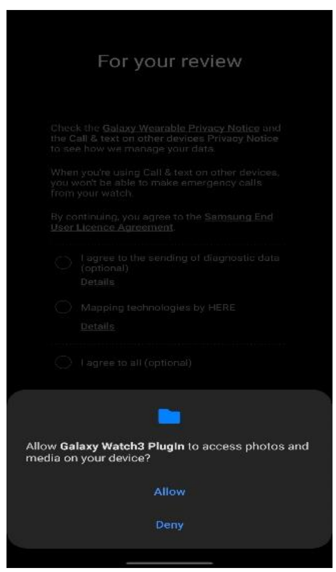


Figure 74 - Media permissions

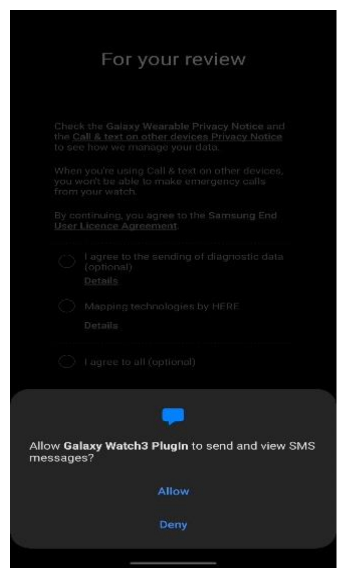


Figure 75 - SMS permission

After the permissions were allowed, you will be redirected to the backup screen. On this screen you have the possibility to restore a backup (if backups were made) or to set backups to be made automatically. With this option enabled, the watch will check once every 24 hours to see if the backup conditions are met. Hit "Next" button.

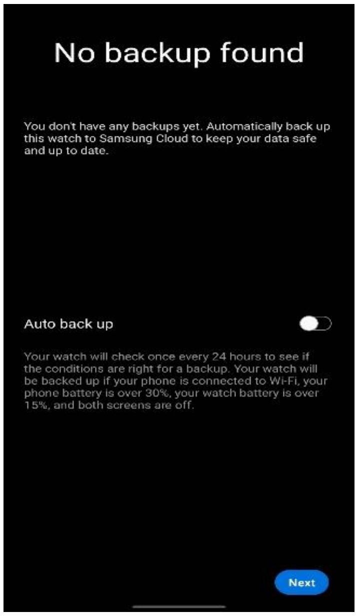


Figure 76 - Backup screen

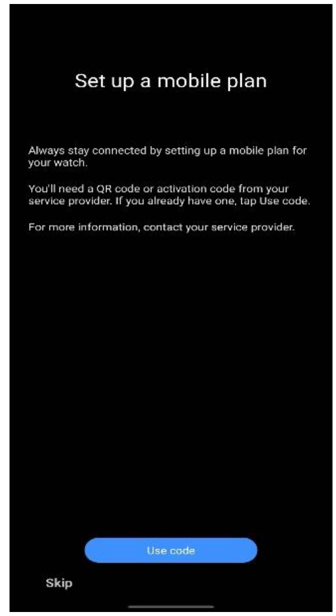


Figure 77 - Setting up a mobile plan

Setting up a mobile plan screen is displayed. Just tap “Skip” button.

At this point, *Galaxy Wearable* home screen will be displayed. You will just be requested to allow *Samsung Health Monitor* to access your Samsung accessory. Allow it and the pairing process is finished.

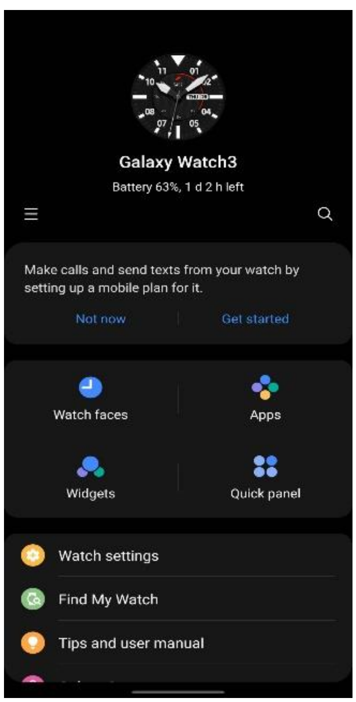


Figure 78 - Galaxy Wearable home screen

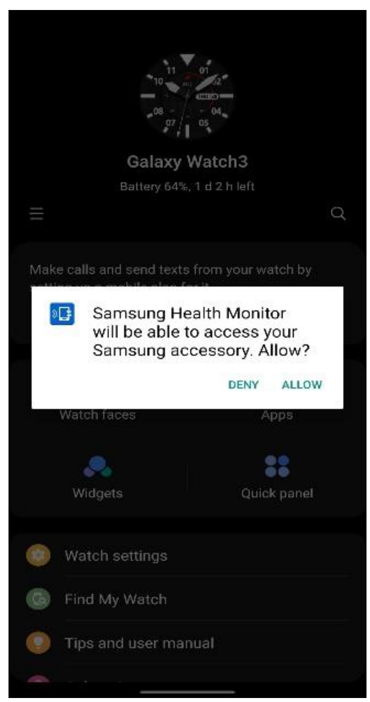


Figure 79 - Samsung Health Monitor approval

Now that you paired Watch3 with your smartphone, the smartwatch will use the phone Wi-Fi connection or its mobile data (if it is turned on). If you want to avoid that the smartwatch to use your mobile phone's Internet connection, you have the possibility to connect Galaxy Watch3 directly to a Wi-Fi network.

13.2. Installing SAVE smartwatch applications

Download SAVE applications from the location provided by Vision Systems and copy the files into "tools" directory located at the path where you installed Tizen Studio (e.g. C:\tizen-studio\tools).

On the computer, open a Command Prompt window, use "cd" command and set the directory to the "tools" directory where the application files were copied (e.g. "*cd C:\tizen-studio\tools*").

```
C:\Users\cparvan>cd c:\tizen-studio\tools
```

Figure 80 - Directory change

Now type "*sdb.exe connect IP*", IP = the IP your smartphone has in the network and that you write it down in a step earlier (e.g. 192.168.1.4).

```
C:\tizen-studio\tools>sdb.exe connect 192.168.1.4
connecting to 192.168.1.4:26101 ...
device unauthorized. Please approve on your device.
```

Figure 81 - Connecting to smartwatch

Check the smartwatch and approve the connection (if you approve it on an earlier step, it is possible that you won't be prompted again for an approval).

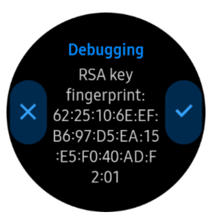


Figure 82 - Connection approving

Type "*sdb.exe install save-configuration.wgt*" and wait for the installation to finish.

```
C:\tizen-studio\tools>sdb.exe install save-configuration.wgt
WARNING: Your data are to be sent over an unencrypted connection and could be read by others.
pushed          save-configuration.wgt   100%       1079KB           0KB/s
1 file(s) pushed. 0 file(s) skipped.
save-configuration.wgt           1176KB/s (1104961 bytes in 0.916s)
```

Figure 83 - Save-configuration app installation

Type “*sdb.exe install watch-face.wgt*” and wait for the installation to finish.

```
C:\tizen-studio\tools>sdb.exe install watch-face.wgt
WARNING: Your data are to be sent over an unencrypted connection and could be read by others.
pushed                  watch-face.wgt   100%       1992KB           0KB/s
1 file(s) pushed. 0 file(s) skipped.
watch-face.wgt                   1531KB/s (2040733 bytes in 1.301s)
```

Figure 84 - SAVE watch face installation

The installation process is completed.

Before start using SAVE watch face, you need to turn on location on smartwatch. Go to “Settings -> Location” and set “Location” to “On”.

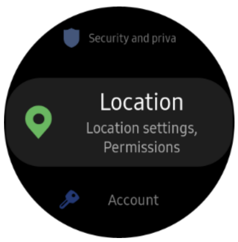


Figure 85 - Location menu

Figure 86 - Location activation

13.3. Registering users in the SAVE web app

The platform is available at http://saveaal.eu:4200/web and can be accessed through popular web browsers. For optimal interaction, it is recommended to use the Chrome browser.

A new user has the possibility to register an account by clicking on "Register" menu option. After the required data is completed, clicking on the "Register" button will create an account. The email field is not mandatory.

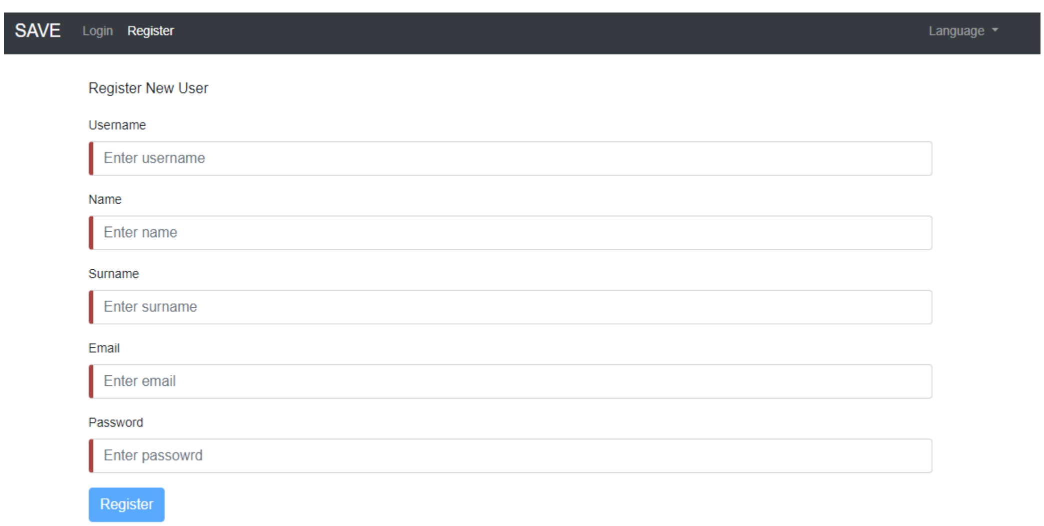


Figure 87 - User registration form

13.4. Assigning a kit to an account in SAVE web app

This page is dedicated to display and submit user data through a form. All fields are required, and the information entered can be changed later.

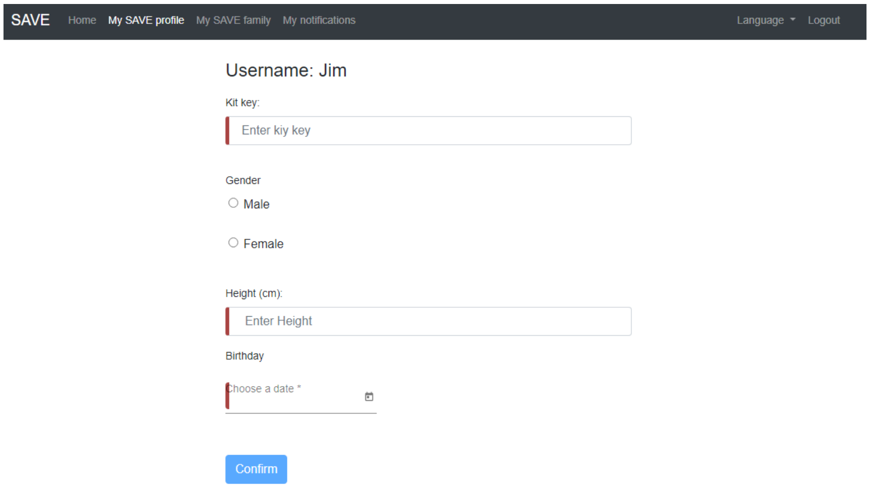


Figure 88 - "My SAVE profile" page without saved data

The kit's key, which the user will find it in the kit package, ensures the security and at the same time it makes the connection between the user and the kit in which the data coming from the sensors are saved. Without a kit key entered, certain sections of the platform such as the data displayed on the "Home" page will be restricted.

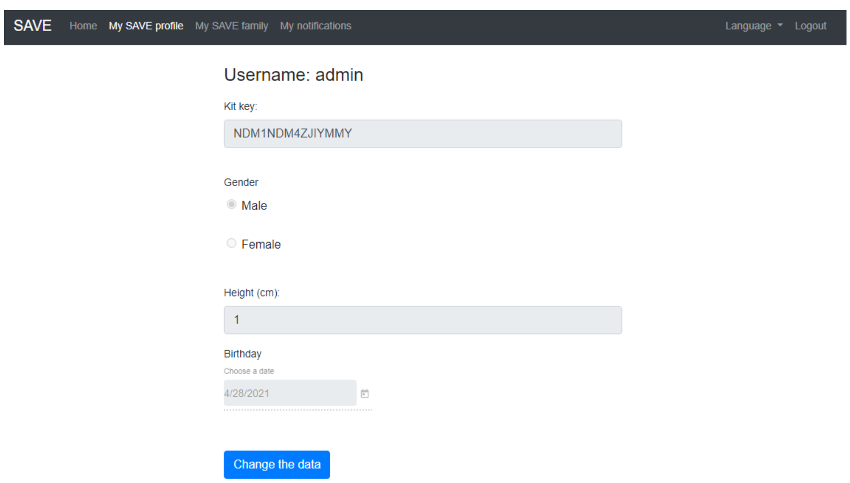


Figure 89 - "My SAVE profile" page with saved data

13.5. Managing caregivers of an account in SAVE web app

Only manager can associate users to caregivers. Log in to your account and press the "Manage caregivers"

Figure 90 - "My SAVE family" page

On this page, the manager can associate a caregiver to a user by selecting their names.

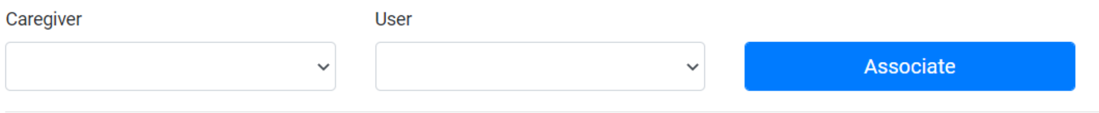


Figure 91 - "User family" page

After selecting a caregiver, you can see names and usernames of users that are linked to selected caregiver account. From this card you can delete association by pressing the button.

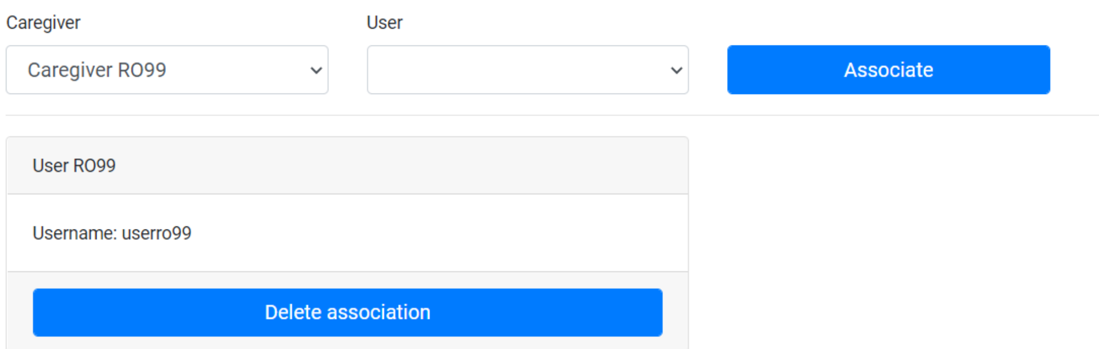


Figure 92 - "User family" page

13.6. Registering the SAVE smartwatch

For a user to be able to have access to SAVE smartwatch features, he/she must register the device using the received PIN code (which can be found in the kit's package). In order to do that, the user will access the smartwatch menu by pressing the Home/Menu button.

Figure 93 - Smartwatch buttons

Then he/she will search through the smartwatch installed applications (by rotating the smartwatch bezel or swiping right on the display) for the "SAVE Configuration" app and then tap on it to open.

After the application is opened, the user will enter the received code into the input box, and then will tap on submit icon, as shown in the figure 95.

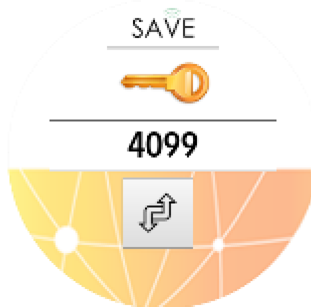


Figure 94 - SAVE Auth application home screen

If the registration was successful, the user will be shown a new screen as in figure 91. If the user is required to register with another code, he/she has the possibility to do that by tapping on change icon . A new screen will be shown and the user can input the new code and submit it.

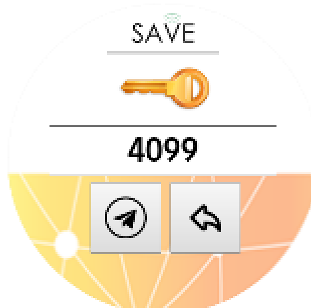


Figure 95 - Registering a new PIN code

By tapping the icon, the user will be redirected to application home screen.

14. SAVE applications features

14.1. Login to SAVE web app

The platform is available at http://saveaal.eu:4200/web and can be accessed through popular web browsers. For optimal interaction, it is recommended to use the Chrome browser.

The platform opens with the login page, and the access is realized based on a username and a password.

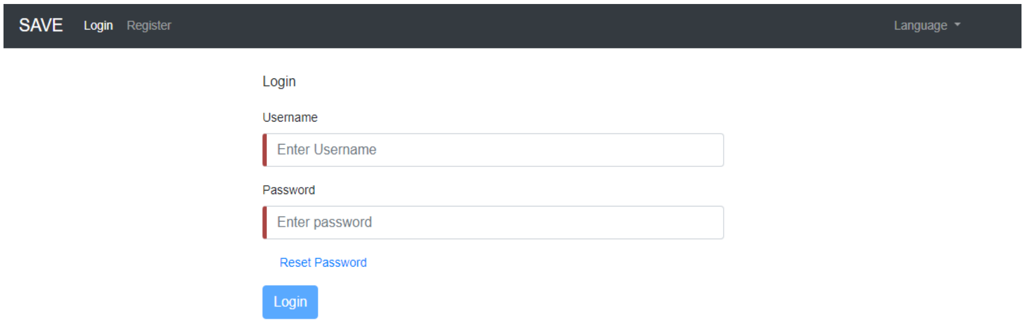


Figure 96 - SAVE web app login page

After a successful authentication, the user is redirected to the "Home" screen.

14.2. Reset your password

The password can be changed clicking on the "Reset Password" link, which will redirect the user to a new page where he/she will have to enter its username, old password and new password, then press the "Reset Password" button. The success or failure of the action is indicated by a notification.

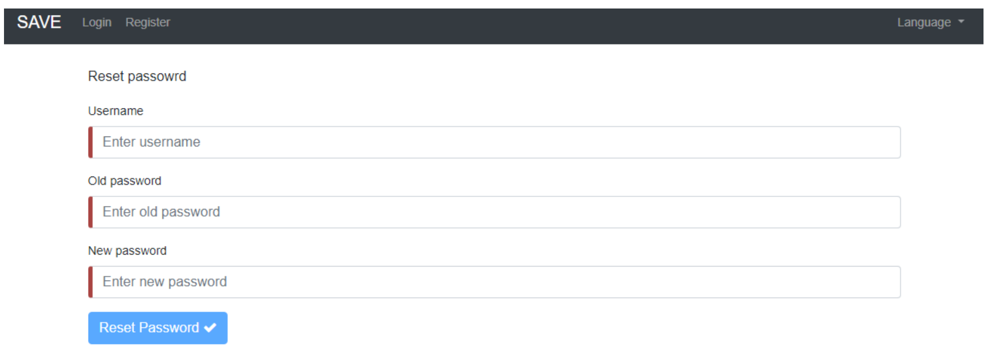


Figure 97 - Reset password page

12.3. Changing the SAVE web app displaying language

Click on the “Language” option from the menu bar and select the desired language.

12.4. Logout from the SAVE web app

At the end of the activity, the user will logout pressing the “Logout” located on the top right in the menu bar.

12.5. View a person’s data in SAVE web app as a caregiver

After the login, the home page contains at the top a list of all the people that the logged in user is taking care of. The first tab is dedicated to the logged in user. If the logged user is a manager, you can view all users’ data in his/her country.

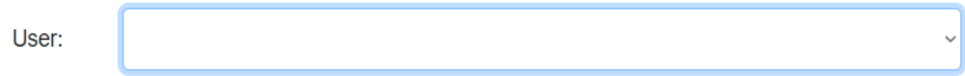


Figure 98 - Users tabs on "Home" screen

If the selected person does not have an assigned kit, instead of the data, a notification message will be displayed. If you are not a caregiver, the only tab available will be "My data".

User does not have assigned kit

Figure 99 - User without assigned kit

The first data set displayed on a user tab is the current day notifications of the selected user. The pink background of the notifications indicates that the scheduled notification was sent on the current day and the blue background means that the notification is to be sent.

Today

	It's time for a walk	15:35
	Have a nice day	20:21
	It's time for a walk	22:58
	Don't forget to take your pills	23:20

Figure 100 - Current day notifications

The next data set displayed illustrates 3 charts with different statistics: the number of steps over the last 7 days, the speed and the number of heartbeats for the last 2 days (figure 101).

The "Show user location" Show user location button located at the bottom of the page, redirects the caregiver to a page where he/she can see in real time the location of the person whose tab is selected (figure 102).

12.6. Notifications

From a caregiver's account, select the user you want to manage notifications and from the page, access "Notifications" (figure 96).

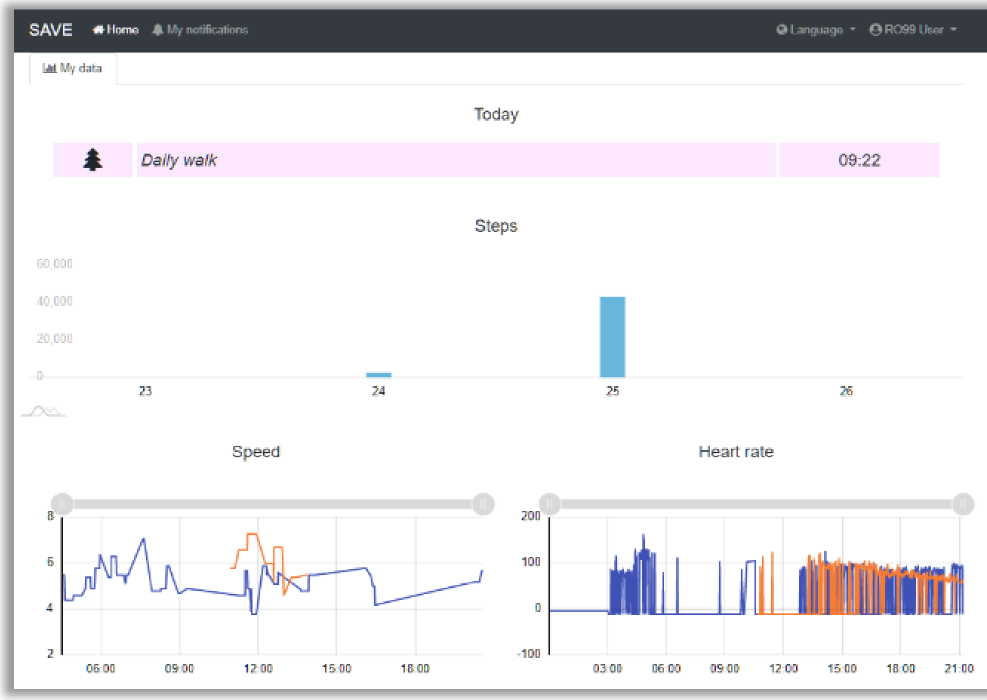


Figure 101 - Graphic representation of the collected data

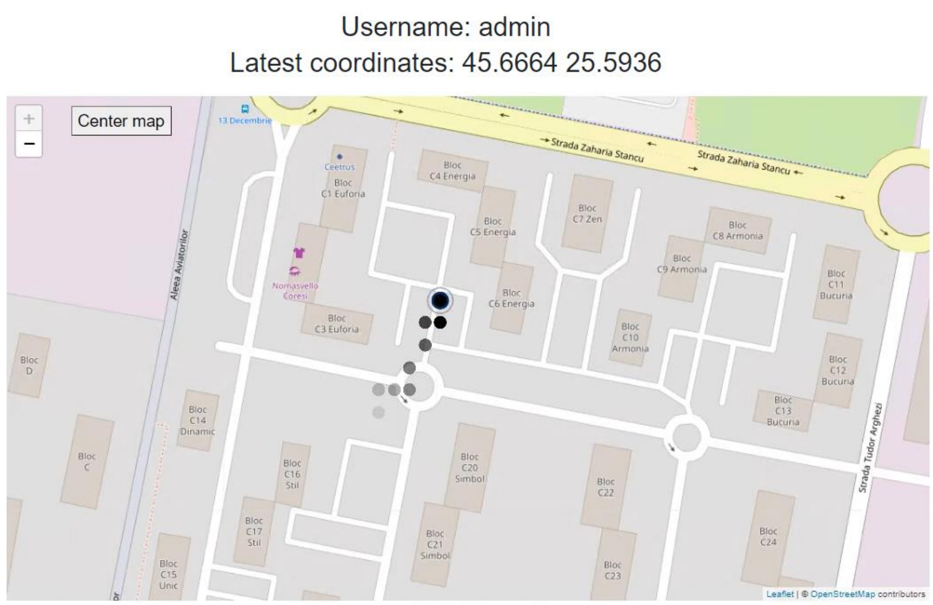


Figure 102 - Selected user's current location

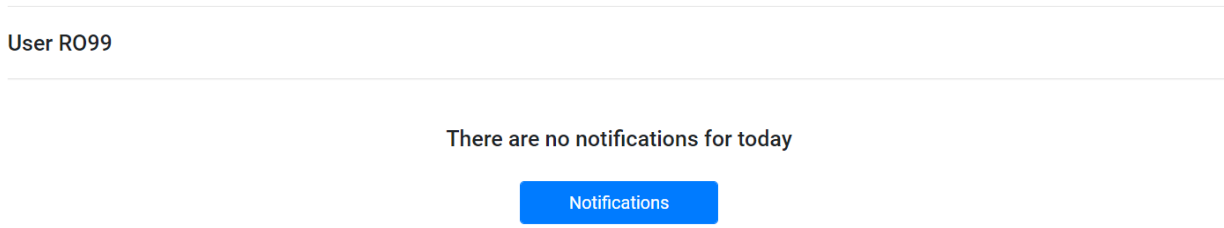


Figure 103 - Selected user's notifications

Access to this page is granted only for caregivers. This section is dedicated to notifications management, which will be sent to the user smartwatch, and it displays the user's notifications in tabular format. The first column contains the notification icon, the second one the notification text to be displayed, the third column indicates when the notification will be sent on the smartwatch device: every day, on certain days of the week and on a certain date (figures 105 & 106).

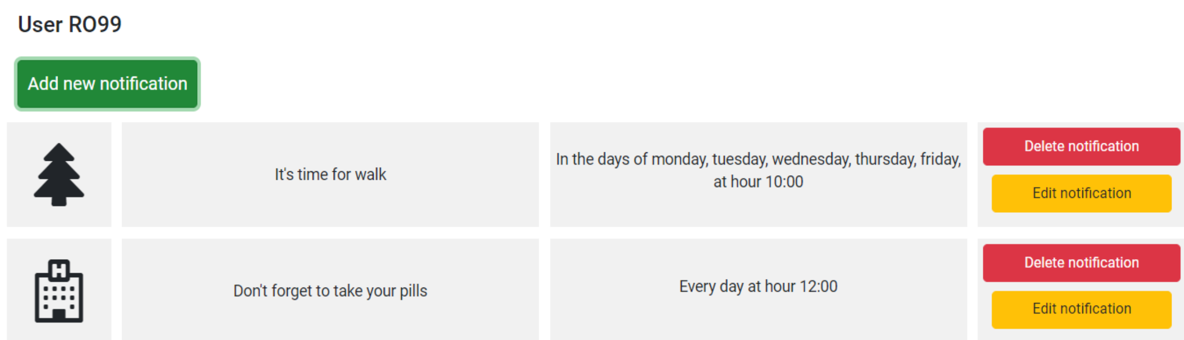


Figure 104 - Notifications management

The "Delete notification" button will delete the notification.

To add or edit a notification the user will press the corresponding button and a window will open in which the notification's text, the desired icon and the options regarding the date and time of the notification must be completed. Also, in this window there are a "Reset" button to clear the entered data, a "Go back" button to close the window without saving the changes and a "Save" button to save the notification.

Notifications that are scheduled to ring on the current day will also appear in a simplified format on the "Home" page.

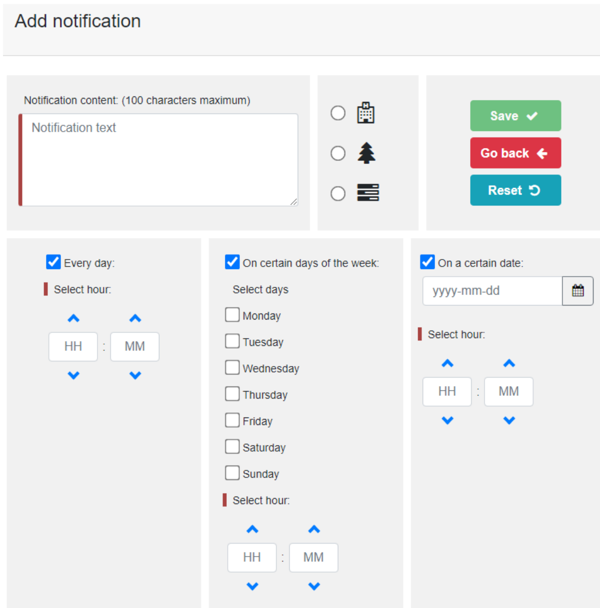


Figure 105 - Add notification window

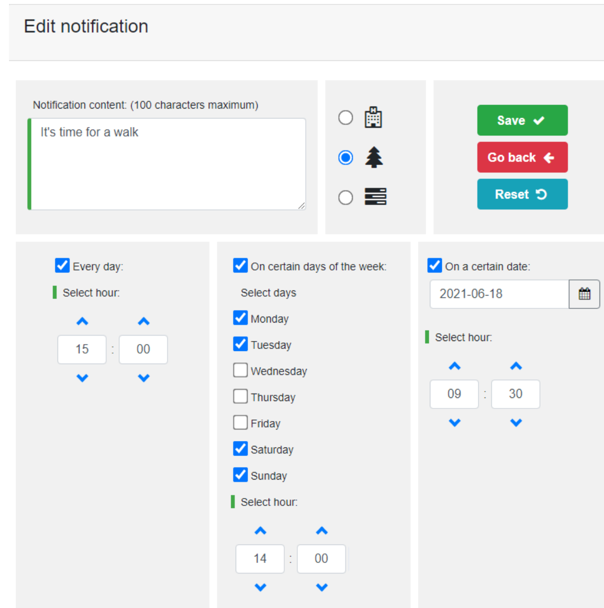


Figure 106 - "Edit notification" window

15. Setting SAVE watch face on Galaxy Watch3

Make sure that SAVE smartwatch registration was successfully and that all permissions are set for the SAVE Face app.

With the watch turned on and its screen active, touch and hold screen to enter edit mode.

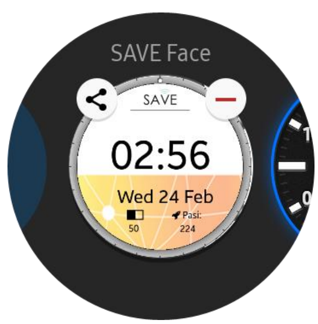


Figure 107 - Set SAVE watch face

Rotate the bezel or swipe left/right and touch on "SAVE Face".

The watch face has a simple user interface. It displays the date, time, the number of user's walked steps and the battery status (figure 107).

Figure 108 - SAVE watch face

16. Getting SAVE smartwatch notifications

After SAVE watch face has been set, the user will receive on the smartwatch the notifications that were set in “My Notifications” page in the SAVE Web application. One notification example is shown in the next figure.

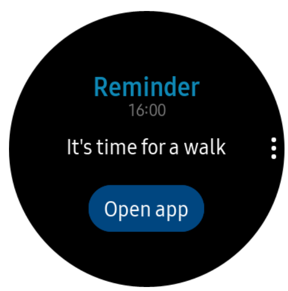


Figure 109 - SAVE notification

17. Setting emergency contacts for calls and SMS

The user has the possibility to set contacts in case of emergencies. The contacts will be set on smartphone in Galaxy Wearable application, “SOS” section.

After accessing “SOS” section, the user can set contacts for emergencies by choosing people from phone’s contact list. For SOS calls the user can choose one person from emergency contacts set previously.

When “Send SOSes” option is activated, if there is an emergency the user can press the smartwatch “Home” button 3 times quickly and then a call will be initiated to the emergency contact along with an SOS message.

When “Detect falls” option is activated, if a hard fall is detected, the watch will alert the user for 60 seconds with a popup, sound and vibration. He/she can cancel the alert or swipe to send the SOS right away. If the user doesn’t respond for 60 seconds, an SOS call will be made and SOS messages will be sent to the emergency contact.

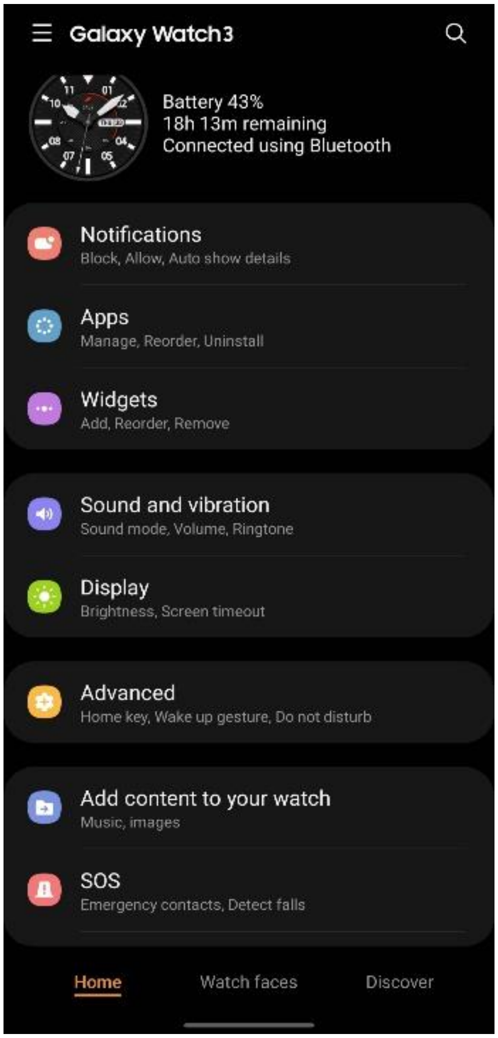


Figure 110 - Galaxy Wearable app home page

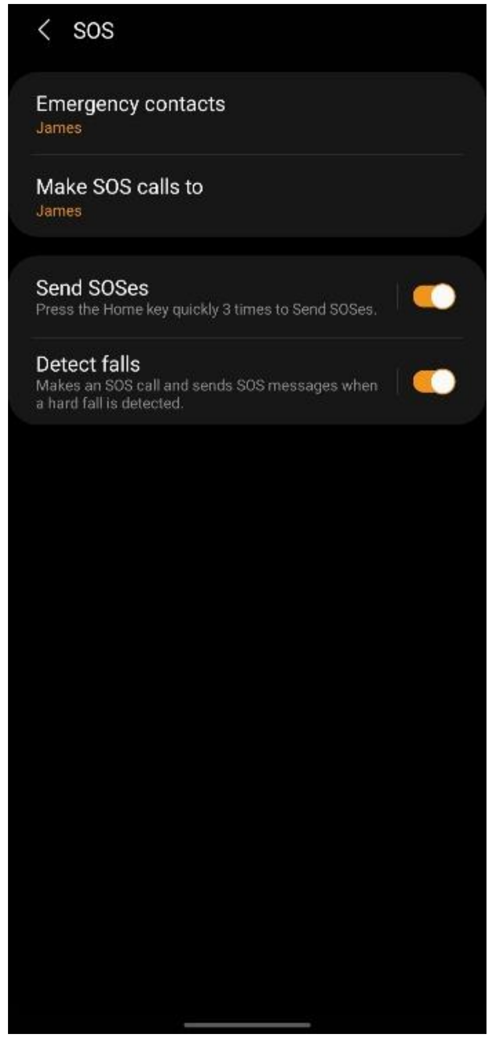


Figure 111 - SOS options

Figure 112 - Emergency call

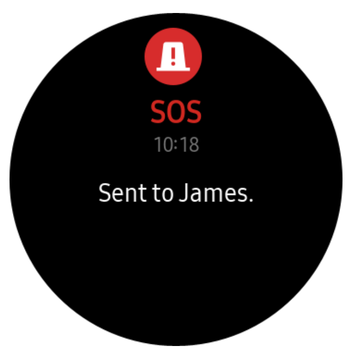


Figure 113 - SOS message

The SOS messages contain person's current location. An example is shown in figure 114.

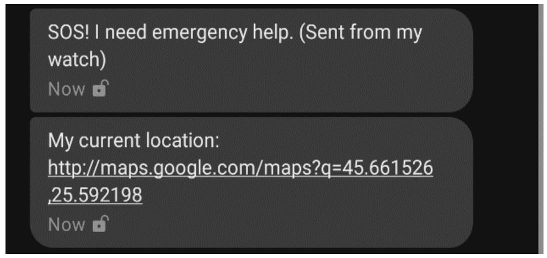


Figure 114 - SOS message example

18. Setting Automation to the Aqara kits

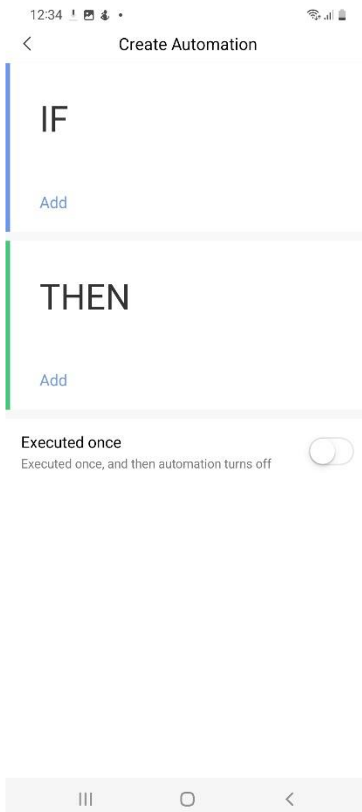
- Install *Aqara Home* app from *Google Play*.
- Connect the *Aqara Hub 2* by adding it as an accessory in the app.
- Add all sensors by following the instructions from the *Aqara Home app*.
- Add a remote control for the *Sony projector* on the Hub; the app will require the user to confirm three buttons.
- Add the following *Automations*, for each sensor:

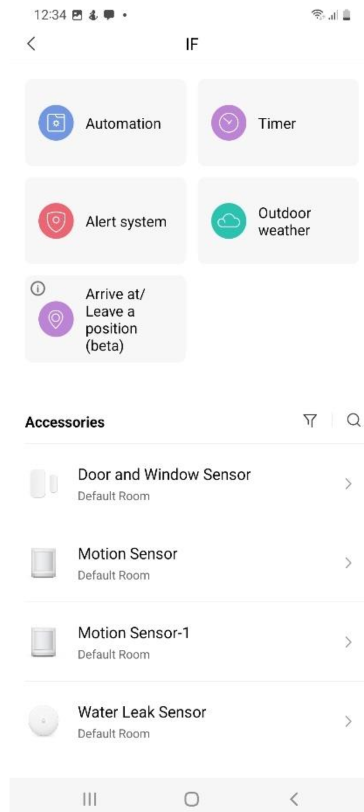
Sensor	Event	SONY Projector Remote Control Button
DOOR & WIN	Open	RESET
	Close	User 1
DOOR & WIN-1	Open	User 2
	Close	User 3
MOTION	Motion Detected	Digital Zoom +
	No motion has been detected in over	CINEMA
MOTION-1	Motion Detected	Digital Zoom -
	No motion has been detected in over	WIDE
WATER LEAK	Leak is detected	INPUT
	Water recedes	NET
WATER LEAK-1	Leak is detected	FREEZE
	Water recedes	USB

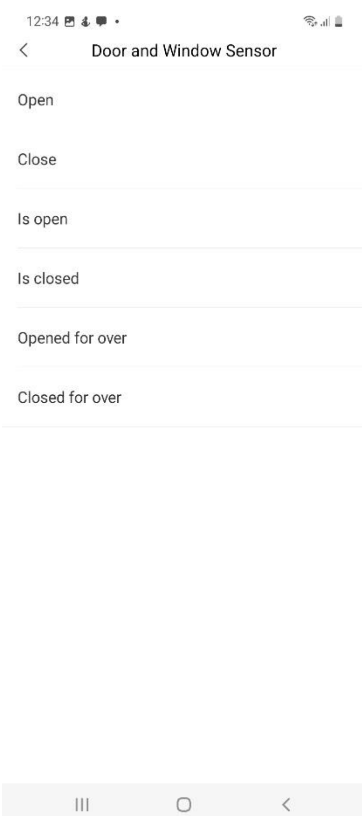
18.1. Creating an Automation

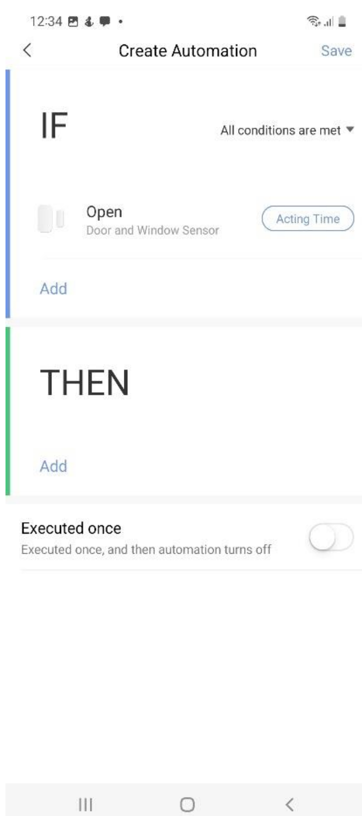
Figure 115 is an example for setting the Automation for the Open event of the Door and Window Sensor (DOOR & WIN). The same must be done for all events in the preceding table.

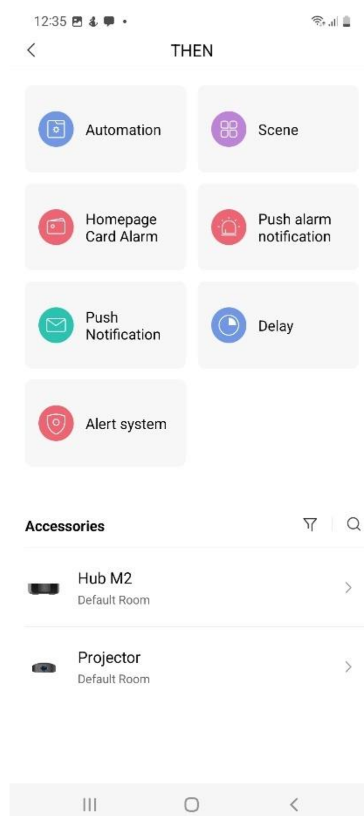

12:33
Automation
Automatic Execution
More>
Add Automation
Home
Accessories
Automation
Scene
Profile


12:34
Create Automation
IF
Add
THEN
Add
Executed once
Executed once, and then automation turns off


12:34
IF
Automation
Timer
Alert system
Outdoor weather
Arrive at/ Leave a position (beta)
Accessories
Door and Window Sensor
Default Room
Motion Sensor
Default Room
Motion Sensor-1
Default Room
Water Leak Sensor
Default Room


12:34
Door and Window Sensor
Open
Close
Is open
Is closed
Opened for over
Closed for over


12:34
Create Automation
Save
IF
All conditions are met
Open
Door and Window Sensor
Acting Time
Add
THEN
Add
Executed once
Executed once, and then automation turns off


12:35
THEN
Automation
Scene
Homepage Card Alarm
Push alarm notification
Push Notification
Delay
Alert system
Accessories
Hub M2
Default Room
Projector
Default Room

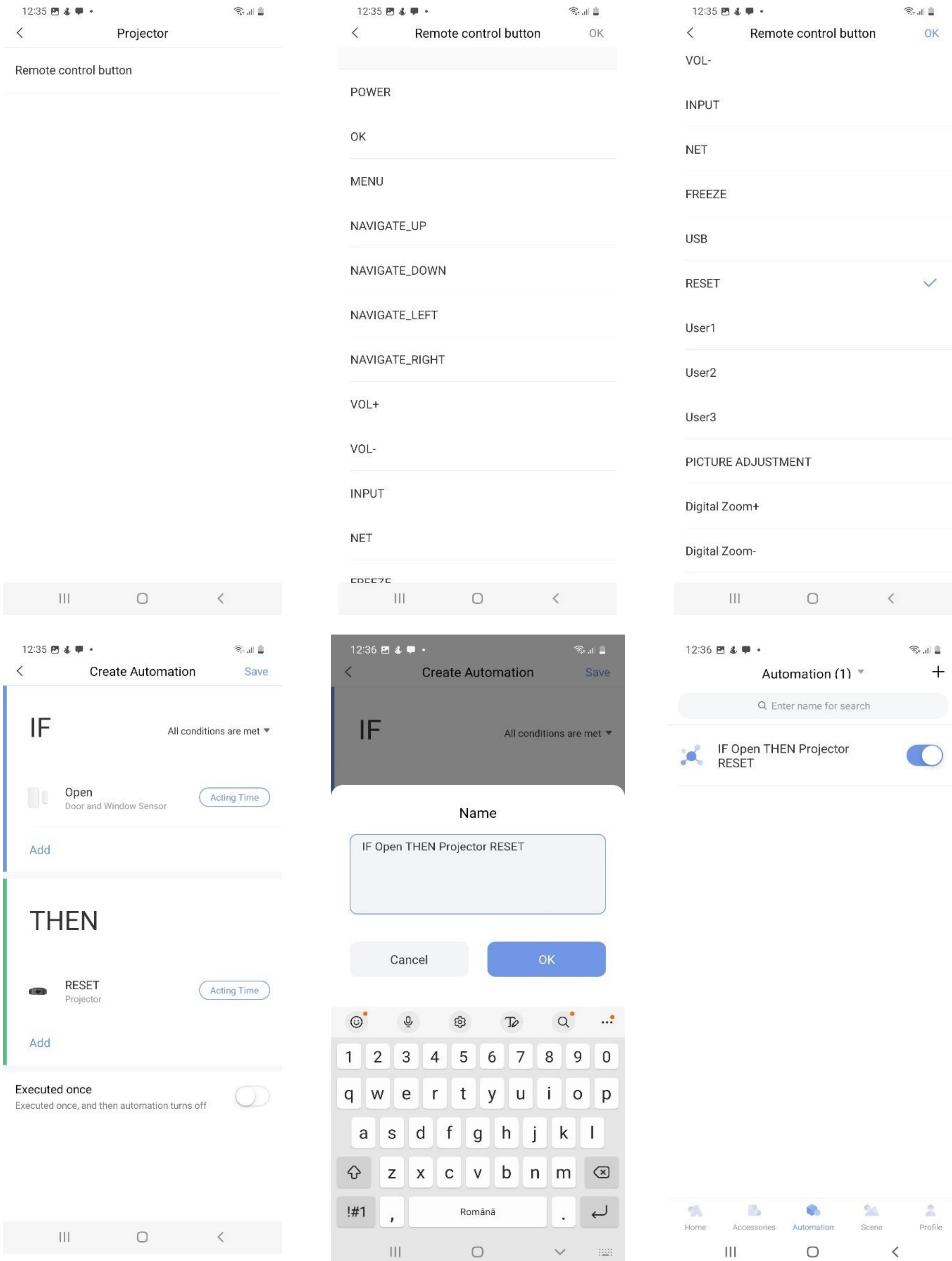


Figure 115 - Setting Automation for an event

19. Configuring the SAVE Sensors Adapter

The main configuration of the SAVE Sensor Adapter is the WiFi credentials.

19.1. Configuring using WPS

If the local router allows / can perform WPS, press the WPS button on the router, then navigate onto the SAVE Sensor adapter to the WPS option (by using the Menu – left – button), then press 3 times the OK – right – button. If the process is successful, the SAVE Sensors Adapter will reset itself and will connect to the network (on the home screen there will be visible the texts: SAVE and Connected – as in Figure 116).

Figure 116 - SAVE Sensor Adapter in connected state

19.2. Configuring using serial communication

If the router cannot perform WPS or the WPS process fails, the serial communication between a PC and the SAVE sensor adapter cand be employed. This can be done only for devices with software version 7+.

First, a piece of software is needed (the *SAVESAConfig* application) that can be downloaded from *http://visionsystems.ro/save/SAVESAConfig.zip* . The application requires *Microsoft's .Net Framework 5.0*.

A device driver must be installed: from the *driver* folder install *CP210xVCPInstaller_x64.exe*. Then the app can then be started (figure 117).

The *SAVE Sensors Adapter* must be connected through an USB data cable to the PC, then the *Update* button must be pressed.

A COM port name corresponding to the COM port associated with the device should appear in the *COM port* dropdown list.

The SSID of the wireless network and the password must be filled in, and then the transfer button must be clicked.

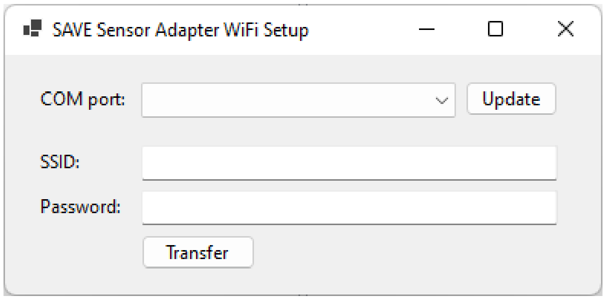


Figure 117 - SAVE Sensors Adapter Configuration application

On the *SAVE Sensors Adapter*, navigate to the SWS option and then press the OK button 3 times. The message *RECEIVED* should be visible on its screen and the device will restart and connect to that network.

20. Technological Club user interface

20.1. Registering a new user

The platform is available at http://saveaal.eu:4200/web/#/tc and can be accessed through popular web browsers. For optimal interaction, it is recommended to use the Chrome browser.

A new user has the possibility to register an account by clicking on "Register" menu option. After the required data is completed, clicking on the "Register" button will create an account.

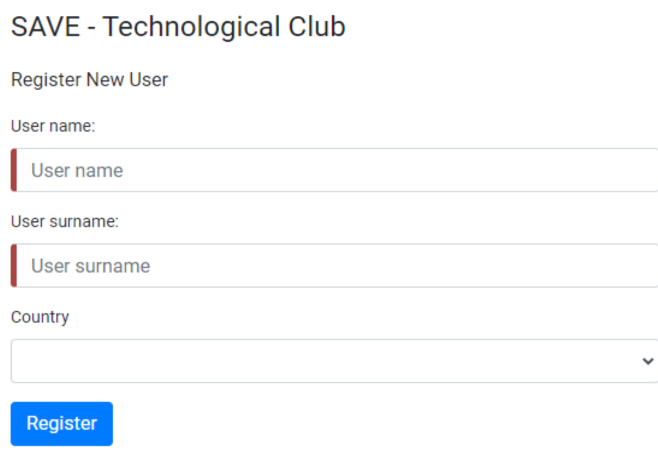


Figure 118 - Tehnological Club register page

20.2. Transferring devices to another account

From dropdown lists select your user country and devices you want to migrate to the new user.

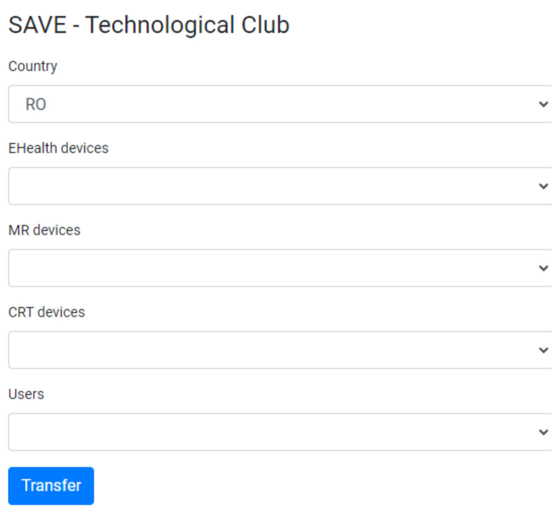


Figure 119 - Tehnological Club transfer devices page

After pressing transfer you will be notified that devices were transferred to new user

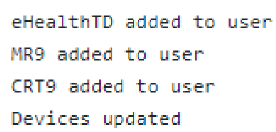


Figure 120 - Tehnological Club transfer notification

21. Admin Centre application

21.1. Dashboard

Once you logged on admin centre application(saveaal.eu:4200/admincentre/) you will be automatically redirected to dashboard page. Here you can view the last 30 records from any device.

You can select what device you want to see from dropdown list situated in the top of the page. After you selected the device press filter and you will see a bunch of cards with last 30 data from that device.

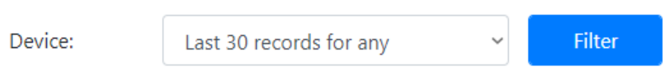


Figure 121 - Dashboard device selection

In the card is displayed the exact date when that data was sent to server, together with the *Device Id*, *Native Id* and the *Kit* of which it is belonging.

Figure 122 - Dashboard sample of card

21.2. System status

In the system status tab is displayed when a kit sent last data. From dropdown list select the kit you want to check. Press filter button after you select a kit.

Figure 123 - Kit selection

This page is divided in two parts. In left half you will see the last data from Aqara sensors and in right half is showed the last data from the Galaxy Watch 3. If too much time has passed, the line from respectively data will be coloured in red colour.

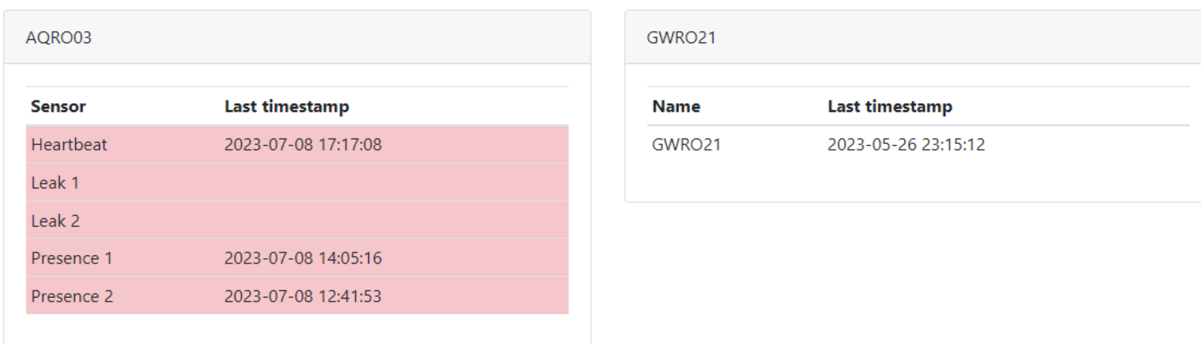


Figure 124 - Status of a kit

21.3. Raw data

In the raw data tab is displayed all data from a limited period for a specific kit. Select kit you want to see data and specify the date you want to view the data.

Figure 125 - Raw data selection

This table contains the raw data sent by the sensors.

Kit name	Device	Data	Timestamp
SAVE-RO-33	GWRO05	{"EQUID":"4005","heartRate":88}	15/07/2023 20:12:18
SAVE-RO-33	GWRO05	{"EQUID":"4005","stepStatus":"NOT_MOVING","walkingFrequency":"0.00","speed":0,"steps":1324}	15/07/2023 19:55:01
SAVE-RO-33	GWRO05	{"EQUID":"4005","heartRate":87}	15/07/2023 19:52:18
SAVE-RO-33	GWRO05	{"EQUID":"4005","stepStatus":"NOT_MOVING","walkingFrequency":"0.00","speed":0,"steps":1324}	15/07/2023 19:35:01
SAVE-RO-33	GWRO05	{"EQUID":"4005","heartRate":88}	15/07/2023 19:32:18
SAVE-RO-33	GWRO05	{"EQUID":"4005","stepStatus":"WALKING","walkingFrequency":"0.60","speed":3.400000095367432,"steps":1196}	15/07/2023 19:17:49
SAVE-RO-33	GWRO05	{"EQUID":"4005","stepStatus":"NOT_MOVING","walkingFrequency":"0.00","speed":0,"steps":1189}	15/07/2023 19:15:01

Figure 126 - Raw data example

21.4 Kits

In this page you can see all kits and also you can create a new kit. To create a new kit press Add button in top left corner. To search for a specific kit type in top right corner in textbox what kit do you want to find.

Figure 127 - Search and add kit bar

In *Add* new *kit* page just type what name do you want for kit and after press add kit button. If you want to cancel just press back button.

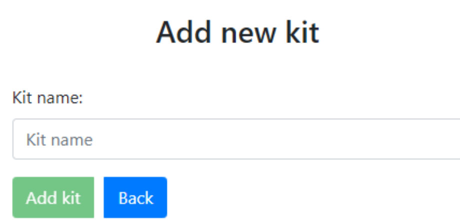


Figure 128 - Adding new kit

The kit menu has 3 operations for each kit, there operations are: see devices from that kit, edit the name of the kit and delete that kit.

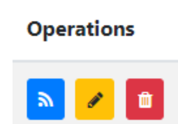


Figure 129 - Operation buttons

Editing is similar to kit creation, you can edit only the kit name.

Delete button triggers a popup from where you can cancel the operation or to confirm deleting.

Devices operation will redirect the user to the Devices tab.

21.5. Devices

In this page is displayed all devices from all kits. If you want to see a specific kit you have to select from dropdown list.

In add new device tab you will have to select from what kit is part of, what device it is, a long description and a short description, including the Native Id of the device.

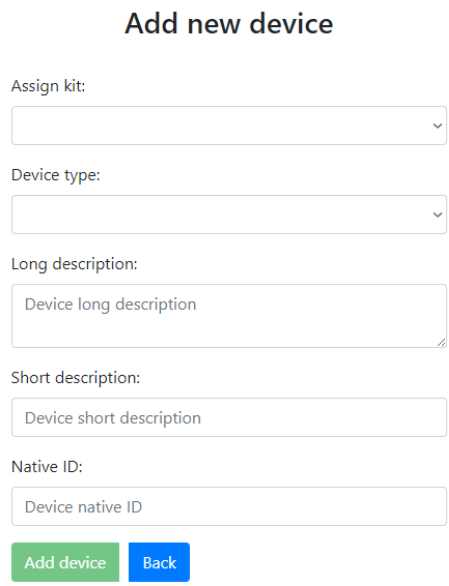
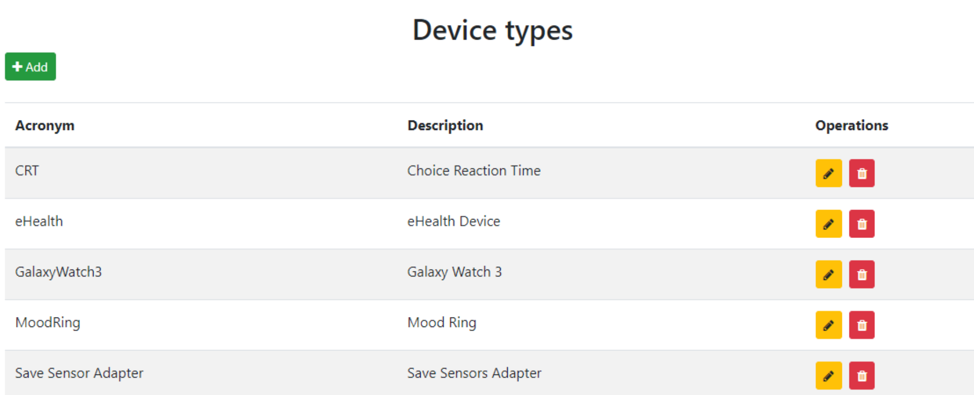


Figure 130 - Adding new device

21.6. Device type

Here is displayed all the device type existing. You can add a new device by typing an acronym and a description.



Acronym	Description	Operations
CRT	Choice Reaction Time	
eHealth	eHealth Device	
GalaxyWatch3	Galaxy Watch 3	
MoodRing	Mood Ring	
Save Sensor Adapter	Save Sensors Adapter	

Figure 131 - Device type table

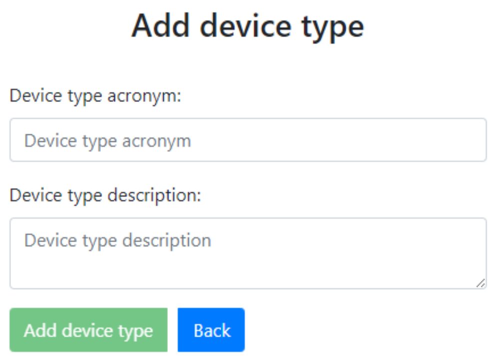


Figure 132 - Adding new device type

21.7. Users

In this page is showed all users and information about them. In adding a new user the email and phone number are not mandatory.

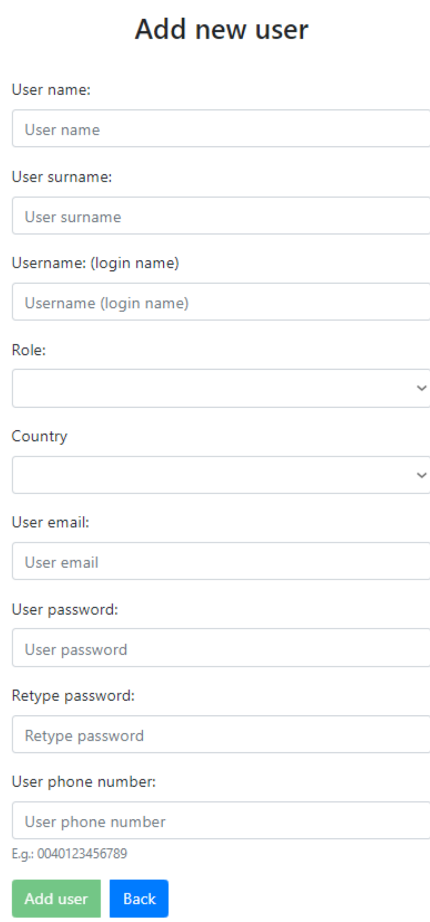


Figure 133 - Adding new user

22. Conclusions

The SAVE system is designed to be a comprehensive and integrated solution that addresses the evolving needs of elderly individuals, informal caregivers, and professional caregiving organizations. It aims to enhance the well-being of elderly individuals by providing them with the means to stay in their familiar environments for an extended period, fostering a sense of comfort and independence. At the same time, it ensures their safety and offers optimal care throughout their aging journey.

The adoption of a microservices-based architecture for the SAVE system allows for greater flexibility, scalability, and ease of maintenance. By breaking down the application into smaller, independent units (microservices), development teams can focus on specific functionalities without affecting other parts of the system. This approach streamlines the development process and enables the rapid rollout of new features, enhancing the system's capabilities and responsiveness to user needs.

The foundation of the SAVE solution lies in cutting-edge technologies, particularly the ones related to cloud environments and containerization. Cloud hosting offers scalability, flexibility, and efficient resource allocation, allowing the system to handle increased user traffic and growing demands. Containerization, through technologies like Docker and Kubernetes, ensures consistent operation across diverse platforms, minimizing overhead and enhancing portability.

The SAVE cloud application, being built using the Spring Boot framework and following a microservices architecture, offers flexibility in its deployment environment. It can be hosted on various infrastructures, including traditional servlet engines/web servers or as a Docker container orchestrated by Kubernetes. This adaptability ensures that the system can be deployed on diverse infrastructures to meet specific requirements and constraints.

Data security is of utmost importance in the SAVE system, and it adopts current standards to protect data and user access. Anonymization of data and separation of user identities from their collected data ensure privacy and GDPR compliance. The communication channels use the current standards and user authentication, and authorization are handled through secure methods like JWTs (JSON Web Tokens).

The *SAVE Sensor Adapter* plays an important role in the data collection process from various sensors in the piloting phase. It reads IR codes emitted by the *Aqara Hub* and relays them to the *SAVE cloud application* through Wi-Fi. The adapter's user-friendly interface allows easy configuration and interaction, enabling seamless monitoring and analysis of user data from the sensors.

The SAVE system provides dedicated user interfaces in the form of responsive web applications and mobile applications. For end-users, there is the SAVE smartwatch face and application, as well as the *SAVE web application*. Caregivers can access the SAVE web application, while the *SAVE Admin Centre* web application is designed for maintenance staff and researchers. The system also offers a *SMS Notification System* for specific events, ensuring caregivers are promptly informed in certain situations.

The SAVE system is intentionally designed to be inclusive and compatible with third-party sensor kits that meet certain requirements. This approach allows for the incorporation of other sensor kits, expanding the system's capabilities and ensuring it can adapt to changing technological advancements.

The eHealth System included in the SAVE solution offers a comprehensive solution for monitoring the health and well-being of elders. Through the integration of wireless and wired eHealth sensors, the system efficiently acquires biometric data, including temperature, oxygen saturation, blood pressure, and spirometry, among others. This data is automatically collected on-site and transmitted via Wi-Fi to the SAVE cloud application, enabling remote monitoring and analysis. The system's architecture, designed using Object-Process Methodology (OPM), ensures scalability, interoperability, and cost-effectiveness. By focusing on the primary value function of collecting non-obtrusive biometric data from elders and transmitting it to caregivers, the eHealth System holds potential in enhancing elderly care and enabling timely and informed decision-making.

The Wellbeing System aims to assess intraindividual variability in reaction time tasks performance and galvanic skin response (GSR) to evaluate the well-being of individuals, especially elders. This system comprises two major components: the Choice Reaction Time (CRT) subsystem and the Galvanic Skin Response (GSR) subsystem, along with the utilization of wearable devices like the Moodmetric smart ring to measure stress levels. By adopting the Object-Process Methodology (OPM) for system architecture, the Wellbeing System ensures clarity in design and interaction with users. It facilitates two operating modes: Regime 1 for research purposes and Regime 2 for caregiver information. The Wellbeing System holds great promise in aiding scientific research with large-scale data extraction and providing caregivers with valuable insights to support the elderly in managing stress and improving their overall well-being.

Overall, the SAVE system represents an innovative and inclusive approach to caregiving, leveraging technology to empower elderly individuals, informal caregivers, and professional caregiving organizations to provide the best possible care and support, while allowing seniors to stay in their familiar environments and maintain their independence.