

| | | |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------|--------------------------------------------|
|  <p>Project Title:</p> <p>Helping elders to live an active and socially connected life by involving them in the digital society</p> <p>Contract no. AAL_08-1-2011-00011 (Magyar)</p> | Deliverable reference: D5 | Date: 31 th December 2014 |
| | Title: Development and integration of systems, components and services | |
| | Responsible partner: IncreaseTime | |
| | Editors: Nuno Rocha, Gil Gonçalves, Raquel Sousa, Simone Pellegrini | |
| | Approved by: | |
|  AAL-2011-2 | Classification: Confidential | |
| <p>Abstract:</p> <p>This report presents a description of a project to develop a web responsive application directed to the “assisted living” market.</p> <p>In this project we intend to develop a web responsive application with wireless sensors for monitoring health and environmental data. This application must also include a social component to offer functionalities such as chat, video call, shared agenda and games.</p> <p>Keywords:</p> | | |

This page was intentionally left blank

| | |
|--------------------------|------------------------------|
| Project: | HELASCOL |
| Contract no. | AAL_08-1-2011-00011 (Magyar) |
| Start – End dates | |
| Deliverable | D5.1, D5.2, D5.3 |
| Date | December 2014 |
| Version | 0.4 |

NOTE:

Under the terms of contract for the implementation of the project, this report is confidential and may contain references to inventions, know-how, drawings, computer programs, trade secrets, products, formulas, methods, plans, specifications, designs, data or works covered by intellectual/industrial property rights of the consortium members. This report may only be used for evaluation of the project. Any other use requires prior written consent from the consortium.

This page was intentionally left blank

Document History

| Revision | Date | Author | Organization | Description |
|------------|-------------------|--------------------------|--------------|-------------------------------|
| 0.1 | 2014.09.04 | Nuno Rocha | iTime | Initial structure and content |
| 0.2 | 2014.12.11 | Simone Pellegrini | SUPSI | Initial structure and content |
| 0.3 | 2014.12.31 | Gil Gonçalves | iTime | Initial structure and content |
| 0.4 | 2015.01.18 | Simone Pellegrini | SUPSI | Added chapter 4.3.3 |

Statement of Originality

This deliverable contains original unpublished work except where clearly indicated otherwise. Acknowledgement of previously published material and of the work of others has been made through appropriate citation, quotation or both.

This page was intentionally left blank

Table of Contents

| | |
|-------------------------------------------------|----|
| Glossary of terms | 1 |
| 1 Introduction | 2 |
| 1.1 Project scope | 2 |
| 1.2 Purpose of this document | 2 |
| 1.2.1 System and components | 2 |
| 1.2.2 Service development..... | 2 |
| 1.2.3 Integration..... | 2 |
| 1.3 Structure of the document | 3 |
| 2 Technology Overview | 4 |
| 2.1 ASP.NET | 4 |
| 2.2 Microsoft SQL Server 2012..... | 4 |
| 2.3 XSockets.NET | 4 |
| 2.4 AngularJS..... | 4 |
| 2.5 ChatJS | 4 |
| 2.6 D3.js..... | 4 |
| 2.7 Google API's | 4 |
| 2.7.1 Google Calendar API..... | 5 |
| 2.7.2 Google Contacts API..... | 5 |
| 2.7.3 Google Gmail API..... | 5 |
| 2.7.4 Google+ Sign-In | 5 |
| 3 Physical Architecture | 6 |
| 3.1 Introduction..... | 6 |
| 3.2 Component diagram..... | 6 |
| 3.3 Deployment diagram..... | 7 |
| 3.4 Physical architecture description | 8 |
| 3.4.1 WebServer | 8 |
| 3.4.2 Database | 8 |
| 3.4.3 WebBrowser | 8 |
| 3.4.4 Agent's Device | 8 |
| 4 System and components (D5.1)..... | 9 |
| 4.1 Introduction..... | 9 |
| 4.2 Components | 9 |
| 4.2.1 Google services..... | 9 |
| 4.2.2 Authentication | 9 |
| 4.3 Framework | 9 |
| 4.3.1 Zephyr – HeartRate sensor | 9 |
| 4.3.2 Mobile gateway | 10 |
| 4.3.3 SUPSI Wireless Sensors Network..... | 10 |
| 5 Service development completed..... | 18 |
| 5.1 Introduction..... | 18 |
| 5.2 Social services..... | 18 |
| 5.2.1 Contacts | 18 |
| 5.2.2 Calendar..... | 18 |
| 5.2.3 Games | 18 |
| 5.3 Communication services..... | 19 |
| 5.3.1 Non real time - Chat | 19 |
| 5.3.2 Real time - Video Chat | 19 |
| 5.4 Personal services..... | 19 |
| 5.4.1 Health monitoring & Home monitoring | 19 |
| 5.4.2 Email..... | 19 |

| | | |
|--------|----------------------------------|----|
| 5.4.3 | Alerts | 20 |
| 5.4.4 | Reminders..... | 20 |
| 5.5 | Integration with sensors | 20 |
| 6 | Integration (D5.3)..... | 21 |
| 6.1 | Application access | 21 |
| 6.1.1 | Description of Application | 21 |
| 6.1.2 | Navigation Bar..... | 22 |
| 6.1.3 | Notifications | 22 |
| 6.1.4 | Profile | 23 |
| 6.1.5 | Settings | 24 |
| 6.1.6 | Home Monitoring | 25 |
| 6.1.7 | Health Monitoring | 26 |
| 6.1.8 | Heart Rate..... | 26 |
| 6.1.9 | Reminders..... | 28 |
| 6.1.10 | Alerts | 29 |
| 6.1.11 | Contacts | 30 |
| 6.1.12 | Chat..... | 32 |
| 6.1.13 | Calendar..... | 34 |
| 6.1.14 | Games | 35 |
| 7 | Conclusions | 36 |
| 8 | References | 37 |
| 9 | Annexes | 38 |
| 10 | Appendices | 40 |

Glossary of terms

Responsive Web Application: It's a web design approach aimed at crafting sites to provide an optimal viewing experience which consist of an easy reading and navigation with a minimum of resizing, panning, and scrolling across a wide range of devices (from mobile phones to desktop computer monitors).

User Story: In software development and product management, a user story is one or more sentences in the everyday or business language of the end user or user of a system that captures what a user does or needs to do as part of his or her job function

Use Case: In software and systems engineering, a use case is a list of steps, typically defining interactions between a role and a system, to achieve a goal.

Actors: Characterizes a role of a person, a system or some external entity linked to one or more system's use cases.

Fibonacci Scale: A sequence of numbers in which each number equals the sum of the two preceding numbers

1 Introduction

1.1 Project scope

The „Helping elders to live an active and socially connected life by involving them in the digital society” project addresses the objectives of the call by offering a 360 degree user involvement methodology to examine how a new approach towards digital technologies can be harnessed to support the involvement of elderly people in digital society. The proposal intends to synthesize the skills, experience and knowledge of the consortium members in developing a state-of-the-art platform and service package backed with feasible business models which supports the on time and on budget realization and market introduction of the call objectives.

The project focuses on providing an enriched communication experience, anywhere, anytime and to any device with accessible, intuitive, easy to use, multimodal User Interfaces. We believe that the right service and the right content are only accepted by the end users if it is delivered on the right device, one that they are used to. This can be the screen of the television, mobile phones, etc. Our goal is to enable elderly people, their family and social surrounding to share their everyday experiences anytime, anywhere and help them make use of existing and currently developed multimedia services to generate the sense of closeness and community belonging they are searching for. This enriched experience, which allows users to share their emotions and experiences in a vivid and interactive way, requires a new approach both in services and the technology that supports them.

1.2 Purpose of this document

In this document is intended to describe the work and the results obtained with the development of 3 parts of the project:

1.2.1 System and components

Includes a general description the full architecture of the developed system and application services

1.2.2 Service development

Technically describes the details of the services developed by SUSPI and iTime with Kapsch consultation, such as Social services, Communication services, Personal services and sensors Integration.

1.2.3 Integration

Describes the Integration of systems, components and services completed. Also it comprises income statement of the actual webapp and description of the used sensors.

The ultimate goal of this document is to report the results achieved with the tasks and the development undertaken. These results are to be validated through pilot studies to be carried out in the following stages of the project.

1.3 Structure of the document

This report is organized in several sections where section 1 contains the introduction of the report and presents the purpose of the document as well as its scope. Section 2 presents a list of the used technologies (along with a short description). In section 3 there is a general description of the systems's physical architecture and its main components, and section 4 presents a description of the overall system structure, including both of the external APIs and the framework created. The communication and services integration are explained in detail in section 5. Results obtained within the development activities are presented in section 6.

2 Technology Overview

This section includes a brief description of the technologies and frameworks to be used on the development of the HELASCOL application. As many web applications relies on the usual platforms such as HTML5, CSS3, JSON, JQuery, Bootstrap.

2.1 ASP.NET

The application will be constructed on the well know framework .NET. Being a web based app ASP.NET will be the base for the development. .NET is a software framework developed by Microsoft with a large reusable library of classes and a development environment that helps developers rapidly and graphically build applications.

2.2 Microsoft SQL Server 2012

For our database we picked Microsoft SQL Server 2012. It's interactions with .NET are well documented and it's the go-to database when developing for this platform.

2.3 XSockets.NET

For the video module we intend on using XSockets.NET. XSockets.NET is a real time communication platform built on Microsoft.NET Technology that provides integration with WebRTC.

2.4 AngularJS

The client side of the application will be built on AngularJS. AngularJS is an open-source JavaScript framework, maintained by Google that assists with running single-page applications. AngularJS lets you extend HTML vocabulary for your application. The resulting environment is extraordinarily expressive, readable, and quick to develop.

2.5 ChatJS

We will be using ChatJS for our chat module. ChatJS is highly customizable and supports both the SignalR and the long-polling adapter which are essential to important features such as online and offline status.

2.6 D3.js

The visualization of chart will be handled by D3.js. D3.js is a JavaScript library for manipulating data and allows us to create real-time graphics.

2.7 Google API's

We will use several API's supplied by google:

2.7.1 *Google Calendar API*

The Google Calendar API allows a program to perform many of the operations available via Google Calendar web interface. Using this API, it is possible to search for and view public calendar events. Authenticated sessions can access private calendars, as well as create, edit, and delete both events and the calendars that contain them.

2.7.2 *Google Contacts API*

The Google Contacts API allows client applications to view and update a user's contacts. Contacts are stored in the user's Google Account; most Google services have access to the contact list. The application use the Google Contacts API to create new contacts, edit or delete existing contacts, and query for contacts that match particular criteria.

2.7.3 *Google Gmail API*

The Google Gmail API allows client application to manage client's Gmail inbox on behalf of a user.

2.7.4 *Google+ Sign-In*

By adding Google+ Sign-In, you bring the power of Google to your site. When a user is signed in, you get an OAuth.2 token for making API requests on their behalf, which you can use to better understand your user, connect them with their friends, and create a richer and more engaging experience.

The first time a user clicks on the sign-in button, they will see an authorization dialog. This dialog outlines how the application will use their data. The user then can consent to the authorization or cancel. After authorizing, a returning user will not be prompted again for authorization. A user always has the option to revoke access to an application at any time.

3 Physical Architecture

3.1 Introduction

In this section we describe the different components involved and how they are represented in a physical environment. A description of the complete structure of the system is provided.

3.2 Component diagram

The developed framework consists of 4 main structures:

User side:

The user side is composed of both an web application and a set of sensing devices fully integrated with the web application. On the application, it can be found social and personal services for data storage, statistical analyses, safety control and communication services.

Google servers:

This framework relies on a some external services provided by Google api to ensure data security.

WebServer:

Increase time is responsible to maintain and support a server that enable other of functionalities such as a data persistent storage and a request interface for data requests.

Metering sensors:

Other sensors were developed and integrated externally and communicate directly with the increase time server through a gateway devised specifically for this type of communications.

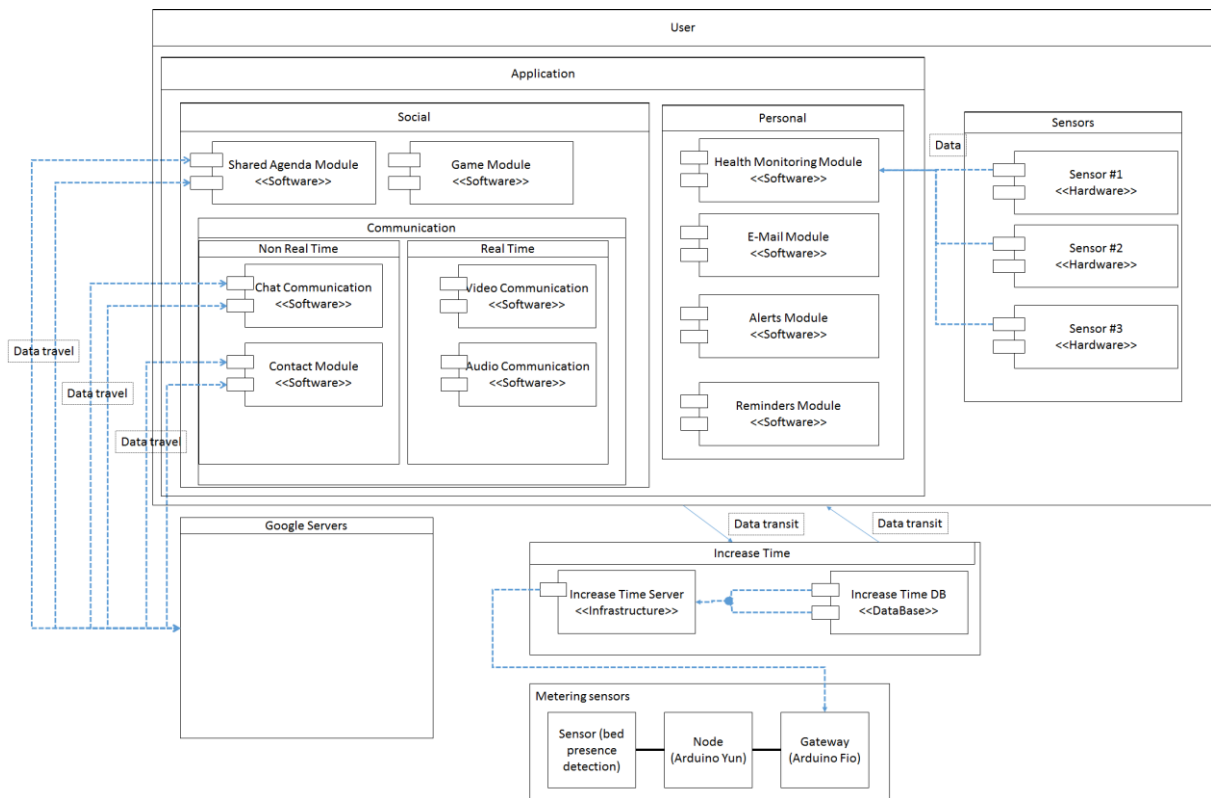


Figure 1. Component diagram

3.3 Deployment diagram

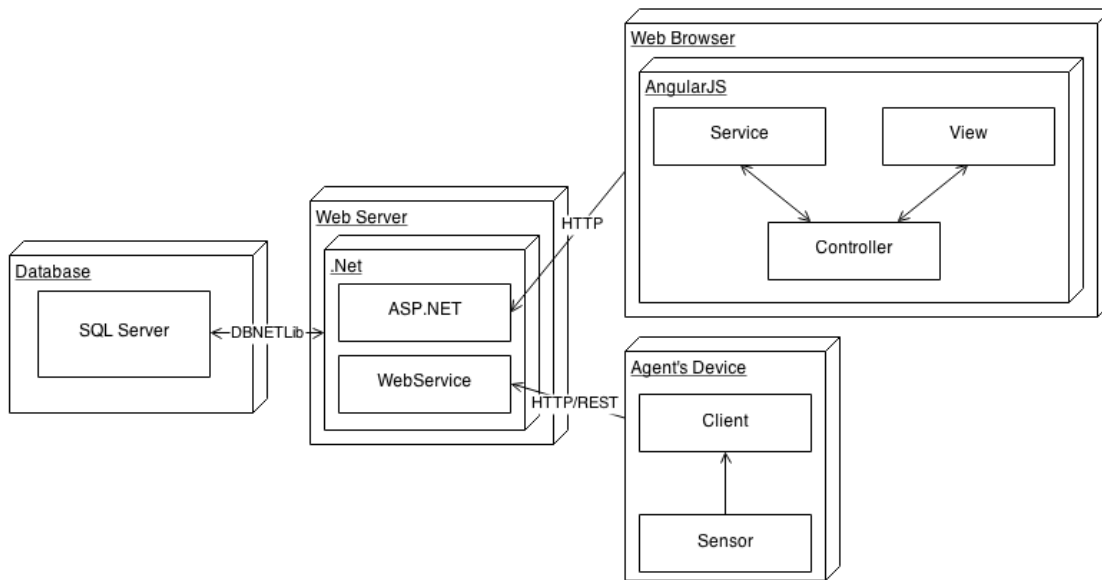


Figure 2. Deployment diagram

This deployment diagram shows how communications between components generically work. All access to database go through a webservice. Other client applications, the web application and sensors' gateways connect to the webservice to access data with insert and get requests.

3.4 Physical architecture description

3.4.1 WebServer

The web server will be running on the .NET platform and providing web support via the ASP.NET framework. It will also provide separate web server that implements the REST protocol for interactions with potential devices such as health and home sensors. This web server will be hosted on the IncreaseTime facilities.

3.4.2 Database

The database will be created on SQLServer 2012 and it too will be hosted on cloud.

3.4.3 WebBrowser

The client side is defined on the user's browser and will be supported by AngularJS to create a seamless interaction with the WebServer. The current version is tested for google chrome Version 39.0.2171.71 m

3.4.4 Agent's Device

The agent's device is a mobile device (smartphone or tablet) with android 4.1 operating system or above that connects to a sensor via Bluetooth. This device communicates via wifi with our WebServer via our WebService in order to insert health and environmental data into our system.

4 System and components (D5.1)

4.1 Introduction

In this chapter it will be described all components built into the system, the third party services used and some protocols used for the hardware integration.

4.2 Components

4.2.1 Google services

A set of google services is used to achieve some features such as user authentication and calendar sharing. Issues with authentication such as account hacking are delegated for the google api. Guest invitations and calendar sharing invitations are also handled by google. Gmail accounts are the core center for this kind of services, forcing the user to firstly have a Gmail account to access the webapp.

4.2.2 Authentication

The authentication uses Gmail register users, but the authentication is achieved through OAuth.2 protocol. After the first login into the application, a token key is generated. This is the key that grants access to the user's data for external applications. Once this key is altered, all external applications lose access privileges as a safety measure.

4.3 Framework

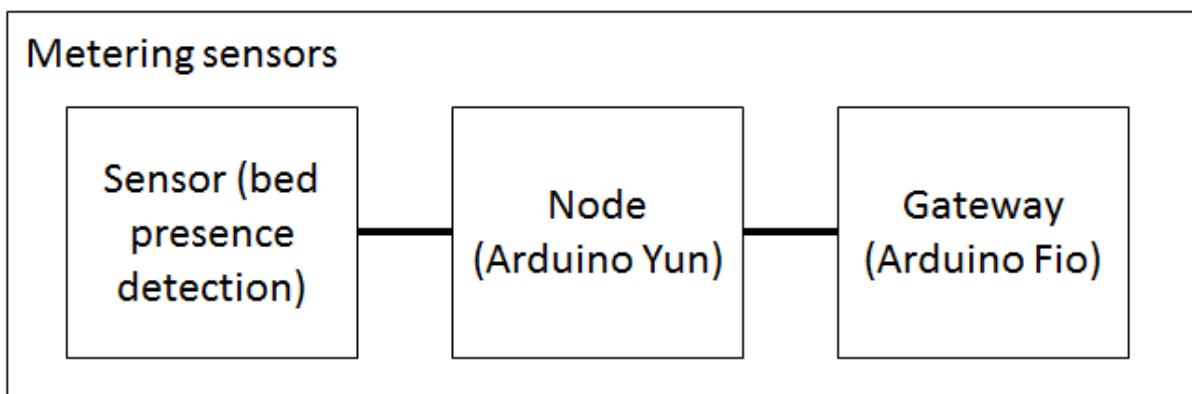


Figure 3: Sensor framework diagram

4.3.1 Zephyr – HeartRate sensor

The vital signs can be gathered by a Zephyr HxM sensor in a chest strap that uses Bluetooth to provide vital signs readings such as heart rate, speed and distance which provide access to users' parameters and potential alerts that may arise based on pre-configured thresholds.

The Zephyr HxM is a physiological monitoring system. The subject wears a Smart Fabric chest strap which incorporates sensors to monitor heart ECG signals.

Attached to the strap is the Zephyr Module. This contains a 3-axis accelerometer for monitoring attitude (subject posture) and activity (acceleration).

Raw sensor data is filtered, processed and analyzed within the device.

4.3.1.1 Transmit mode

Data is transmitted by Class I Bluetooth over a 10 meter range to a corresponding Bluetooth receiver device. This will allow physiological data to be monitored using any suit ably configured Bluetooth mobile device, such as a laptop, phone or PDA.

A Bluetooth Test App is provided with the Developer Kit which displays transmitted data in real time.

4.3.2 Mobile gateway

Through continuous monitoring of health status and location of an individual, the gateway facilitates assistance quickly in emergencies, since when are verified anomalies are automatically issued alerts for the caregiver and / or centralized emergency system.

The gateway communicates with wireless sensors to continuously monitor the vital signs, or any other parameter that is desired to monitor, and processes this information to provide a set of services such as:

- Monitoring management
- Manual values insertion

This module consists of a smartphone with the mobile application KeepCare. It is responsible for the acquisition and processing of data monitored by Bluetooth wireless sensors, and transmitting these to the central server. This module is also responsible for triggering alerts if the values exceed upwards or downwards the previously defined limits. The second module is based on a web application in which the caregiver can check real-time data monitored as well as set up and configure a set of parameters. The movable component KeepCare platform was developed for the Android operating system since, given the required specification is considered the most advantageous platform for the purpose.

The application base is a service that manages the communication with the sensors and the synchronization with the remote server. The service implements an interface in order to allow the main application, communicate with this service, allowing the user to change the state of connection with the sensors through the UI as other relevant parameters for monitoring. When the Gateway connects to the service it retrieves data and the application makes calls to the Service.

This service also uses TimerTasks that allow a portion of code to be executed promptly. Thus it was possible to implement reconnection mechanisms to sensors in the case of sensors are disconnected for any reason, as they leave the reach or come into hibernation.

4.3.3 SUPSI Wireless Sensors Network

System overview

The SUPSI wireless sensor network (WSN) concept consists of a sensor network based on smart nodes (node and connected sensor/actuator) able to read and possibly write general purpose input/output lines either analog or digital. These lines are connected to sensors or actuators. Sensors are devices that produce output for evaluation and or processing by the node while actuators are devices that require input driven by the same node. Smart node communicates via a radio module (XBee) in the 2.4GHz ISM frequency band to a sensor network bridge acting as a coordinator for the underlying network. The coordinator then sends the sensor messages via a Wi-Fi access point (AP) or a portable hot-spot to the collecting database via HTTP queries at the associated web service. The state or histories of the sensor messages are accessible via any user device running a web browser (e.g. laptop, tablet or Smartphone).

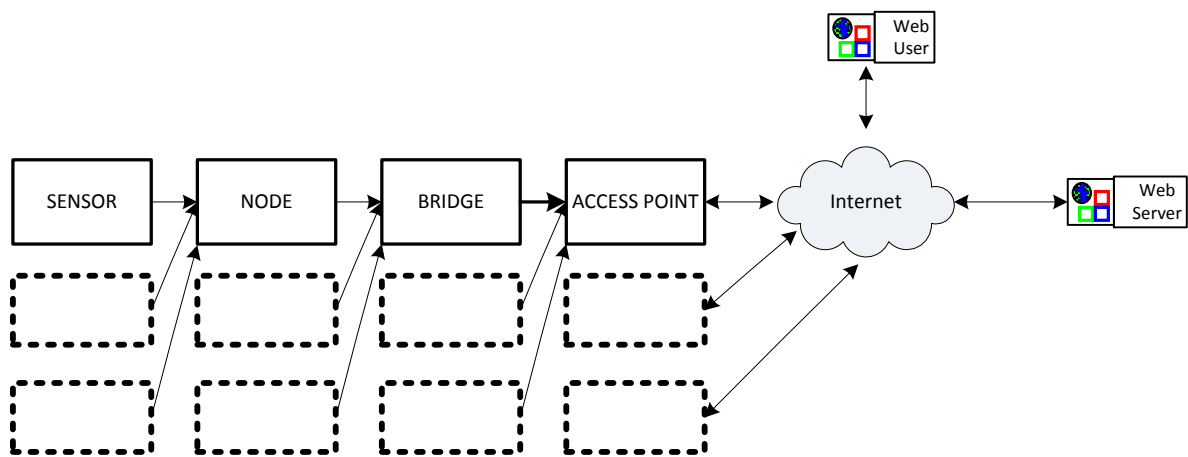


Figure 4 - WSN System overview

Sensor

Sensors are generally output devices reading physical quantities (e.g. temperature, pressure, a.s.o.) or able to detect state changes in a controlled system and translate this information into electrical signals. The scope of our project focuses in monitoring the user environment for safety purposes so we identified two kinds of sensors detecting state changes: a person-in-bed sensor and water-in-bathroom/water-in-kitchen sensor.

Person-in-bed sensor

This sensor is based on a component that changes the electrical resistance when a force is applied vertically. Putting this component under the mattress and measuring his resistance it is possible to detect when a person is or not in bed and then monitor the sleep cycle.

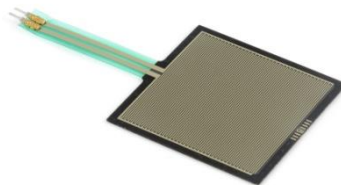


Figure 5 – Person-in-bed sensor

Water-in-bathroom sensor and water-in-kitchen sensor

This sensor is based on a piezoelectric microphone that generates a continuous electric signal proportional to the acoustic noise level. Positioning this sensor on the water pipe it is possible to detect when the water is flowing inside by performing a fast Fourier transform (FFT) analysis running on the node.



Figure 6 – Water-in-bathroom sensor and water-in-kitchen sensor

Node

Nodes are boards hosting a small microcontroller connected to an XBee radio communication module. This board can be powered by plugging a USB cable or a Li-Po battery cell for more flexibility. The USB cable can be used to charge the battery as well.



Figure 7 – Self-powered Arduino Fio

The communication module is an XBee radio configured as a ZigBee router to send and receive messages to/from the bridge using a simple character-based protocol to encode source, type and meaning of messages (i.e. coming from an attached sensor or from the node itself). These messages can notify sensor state changes, power-up events, low-battery events and heart beat events.



Figure 8 – XBee module configured as ZigBee router

Sensor and node combined together build a smart node.

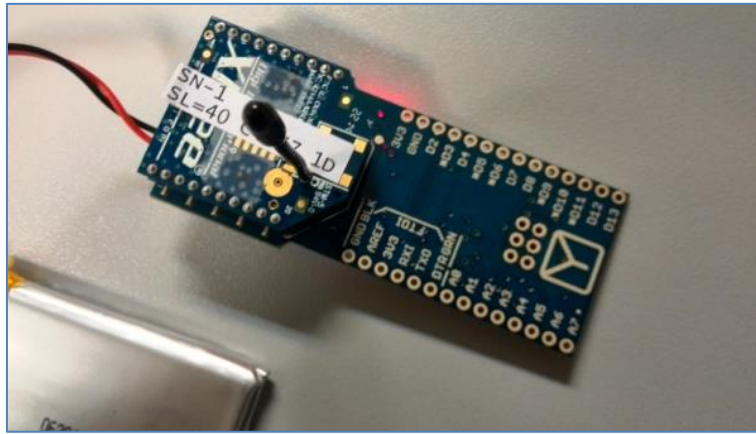


Figure 9 - Arduino Fio (node) with XBee module configured as router

Bridge

The network bridge is a board hosting a small microcontroller connected to an XBee communication module configured as a ZigBee coordinator and a more powerful microcontroller running an open source operating system controlling peripherals like the Wi-Fi network interface to connect to the Internet. The XBee module receives and sends messages from/to the smart nodes inside the same room (or network) while the Wi-Fi interface is used to forward sensor state changes to the web service using RESTful API.

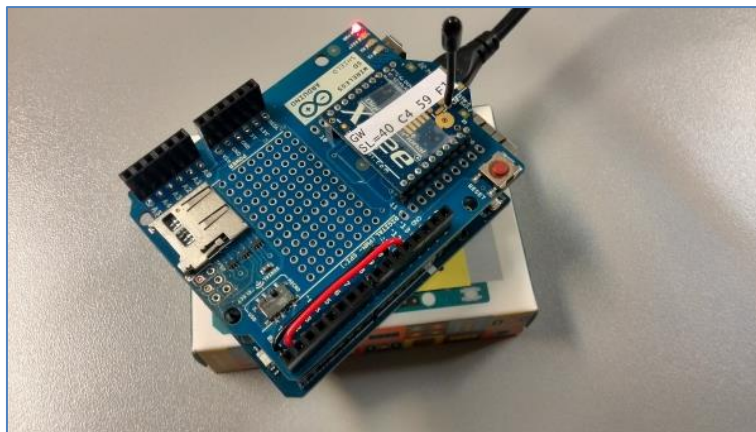


Figure 10 – Arduino Yun (bridge) with XBee module configured as coordinator

Access point

The access point (AP) is the gateway to the Internet; it enables one or more bridges to reach the remote web service. There are two possible scenarios.

In the first case a (modem) router AP is used when the infrastructure has a DSL connection.



Figure 11– Router access point

In the second case a mobile AP is used when the infrastructure don't have a DSL connection like for demonstration setups. The second case connects through 3G mobile network and hence a SIM with a working data plan is required.



Figure 12– Mobile access point

Detailed description

The system overview can be depicted more in detail describing the inside of each piece of hardware used to implement the wireless sensor network. Sensor and node build the smart node, the single device acting as end-point in the WSN, while the bridge act as a gateway between the XBee network and the Wi-Fi/Internet network.

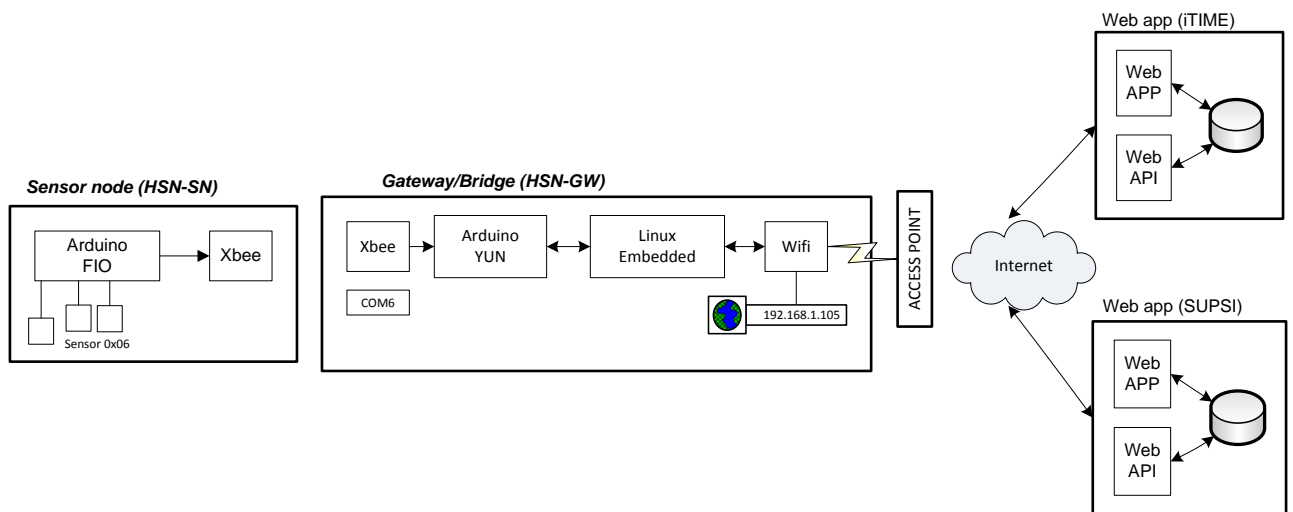


Figure 13- WSN end-point and gateway

Smart node (HSN-SN)

This device is composed by a sensor, a node and an XBee S2 radio module configured as a ZigBee router. The core of the node is an Arduino Fio board with one or more sensors connected on its IOs (input/output lines, either digital or analog). Each sensor node receives broadcast messages from the coordinator (bridge) and sends events only to the coordinator in a so called star network typology. Events generated from sensor nodes are changes in the state of the sensor, power up or reset, low battery and heart beat (a timed event saying the node is alive).

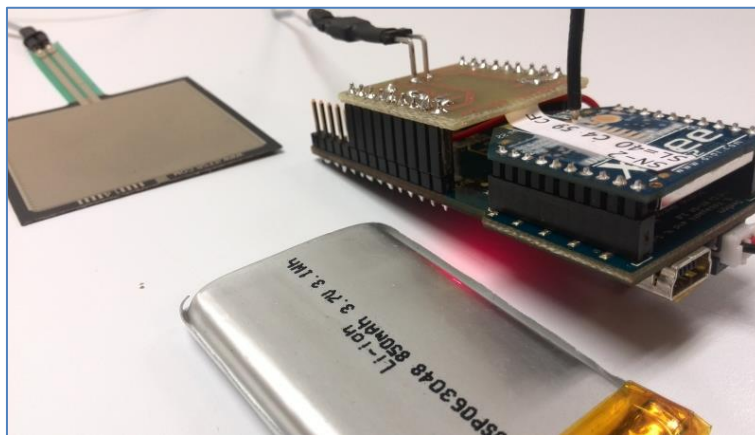


Figure 14– Smart node with person-in-bed sensor

For more details see HSN-SN in the annex part of this document.

Gateway/Bridge (HSN-GW)

This device is composed by an Aduino Yun board and a Wireless SD shield hosting an XBee S2 radio module configured as a ZigBee coordinator for the wireless sensor network. Events coming from the smart nodes need to be forwarded to the web service. This task is delegated to the Linux processor hosted on the Arduino Yun board and connected to the LAN/WAN via a Wi-Fi adapter. Event messages are then converted to HTTP request (RESTful API) and sent to the service back-end.

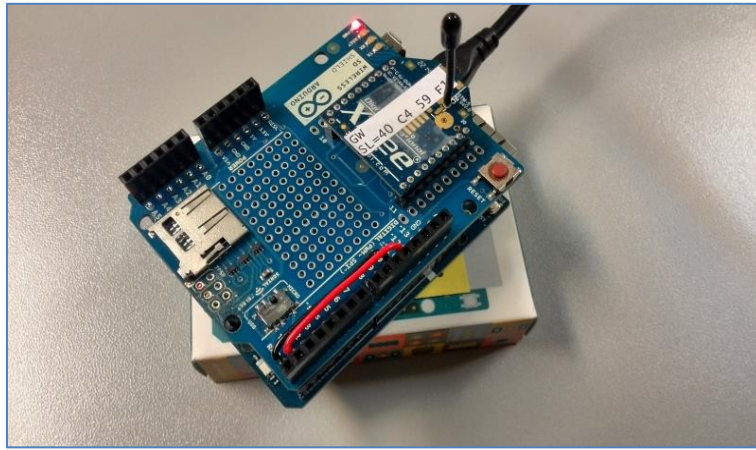


Figure 15 – Gateway/bridge

For more details see HSN-GW in the annex part of this document.

Sensor Node – Bridge protocol

A simple character-oriented communication protocol has been implemented on the smart node for bi-directional communication, network management and control of the input/output lines between smart nodes and gateway/bridge. Notification of events messages are straight forward while the incoming commands require some more tweaks on the server side. Even if the communication protocol is able to manage bidirectional messages (from sensor node to gateway and from gateway to sensor node), only one way is used in this project.

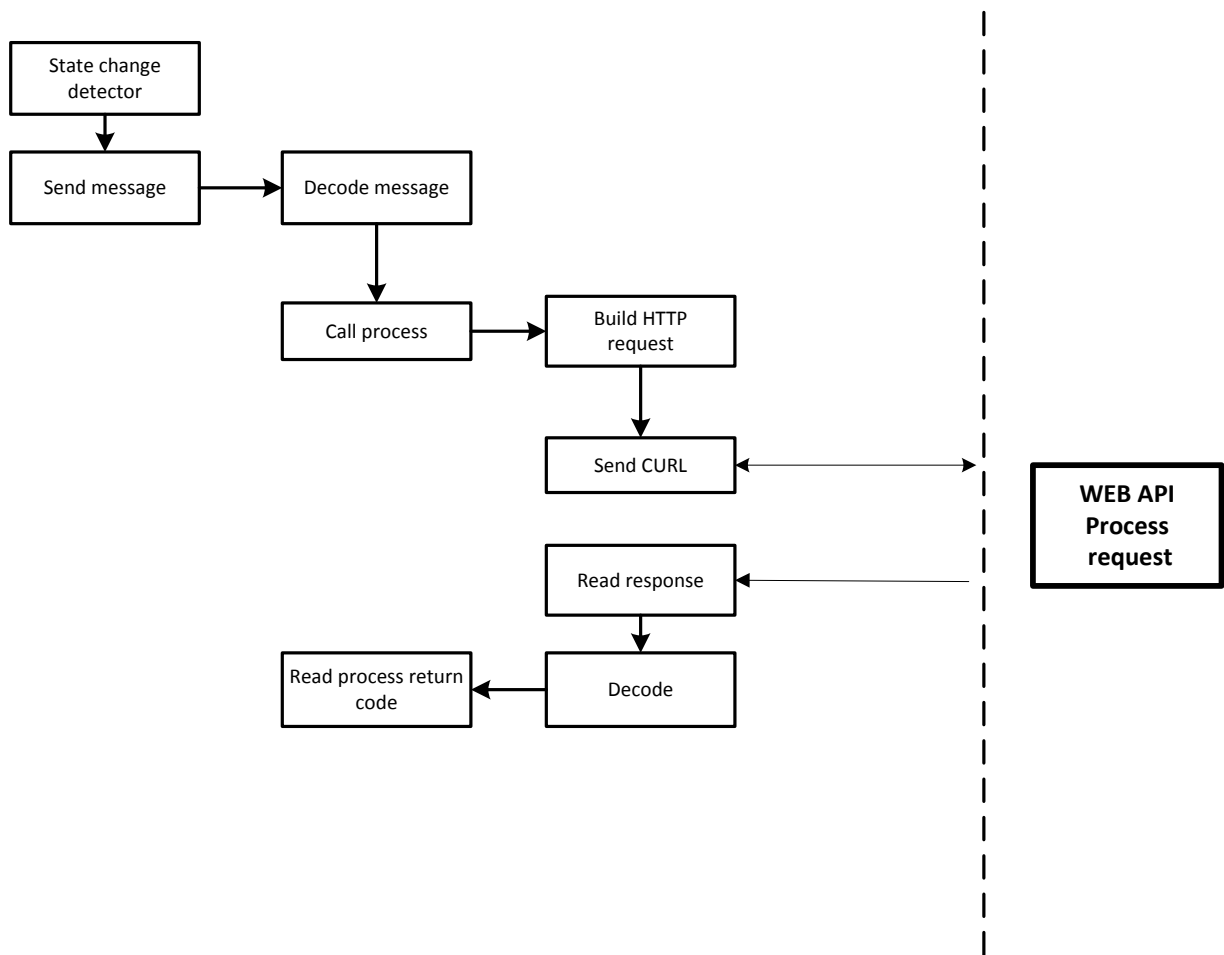


Figure 16– Wireless sensor node communication protocol

For more details about the protocol see HSN-SN in the annex part of this document.

5 Service development completed

5.1 Introduction

These services are built in an application and are separated in Social and Personal services. Social services are the ones used for communication and sharing information. As for more personal information directly connected to the user there are the personal services. Social services are supported by google apis and services and are external to the application.

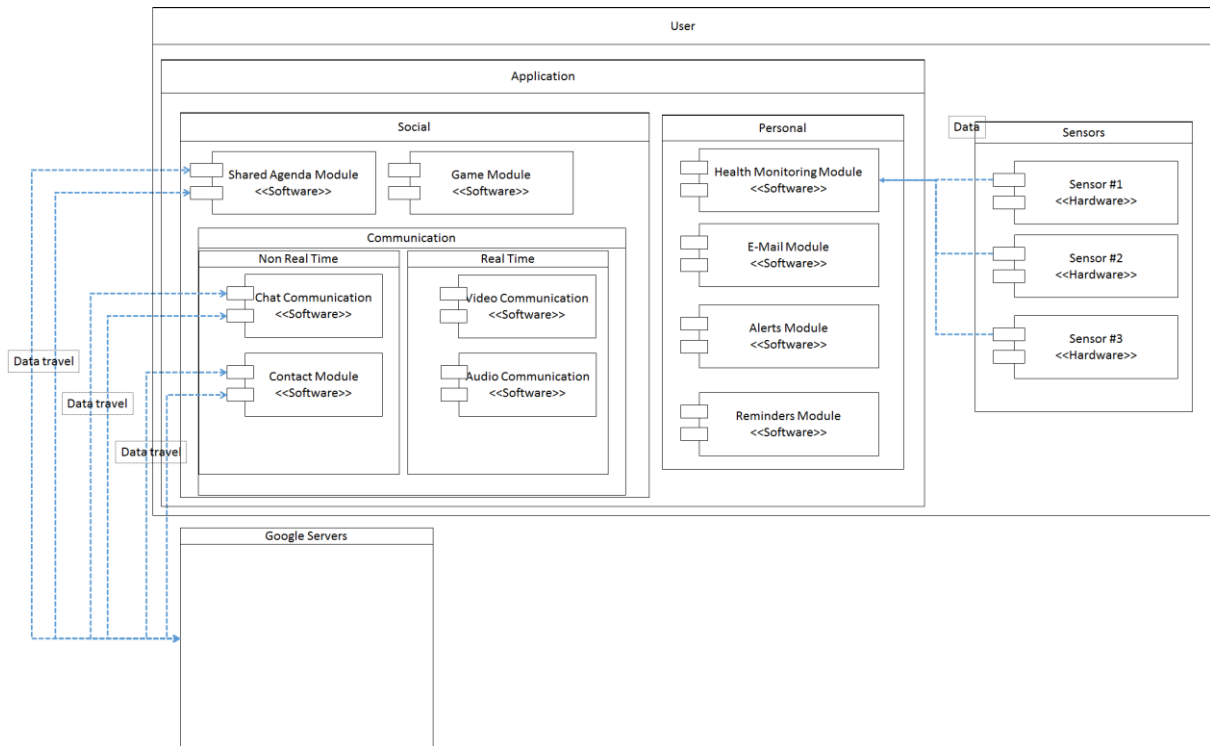


Figure 17: Services and components developed

5.2 Social services

The social section of the web application provides the user access to the configuration of personal contacts, calendar-oriented games for the stimulation of cognitive abilities. It is a complement to the previous section, facilitator of recreational activities and social life, important for healthy and stimulating maintenance of life.

5.2.1 Contacts

This module ensures the display of user contacts, lets you add and delete contacts and con-still associate a user account invited to contact.

5.2.2 Calendar

This module ensures the display of shared calendars with the user, allows subscription of a new calendar, enables the detail view as each calendar events and even add a new calendar

5.2.3 Games

This module provides the user with access to a set of predefined game-oriented maintenance of cognitive functions such as memory and mental calculation function in addition to the fun.

5.3 Communication services

The communication section of the web application provides mechanisms for inter user communications.

The user access to the configuration of personal contacts, the chat service use and video call, calendar-oriented games for the stimulation of cognitive abilities. It is a complement to the previous section, facilitator of recreational activities and social life, important for healthy and stimulating maintenance of life.

5.3.1 *Non real time - Chat*

This feature allows the user to view the status of other users in the chat, send and receive text messages with another application user and even invite a user from the list of contacts but has not yet accessed the chat. Through this service, it's possible for two users to exchange messages with each other instantly and also, messages are stored to be read later.

5.3.2 *Real time - Video Chat*

For real time conversation a video chat room is enabled via WebRTC. Only two users are allowed per video chatroom. This line of conversation enables real time video and audio communication as long as both users are logged in. At any time the call can be terminated, closing the video and audio channels.

5.4 Personal services

5.4.1 *Health monitoring & Home monitoring*

5.4.1.1 Health monitoring

The application provides a self-monitoring service. This service enables data collection and threshold configuration for the following sensors:

- Blood Pressure: data for the user diastolic and systolic blood pressure in mmHg.
- Glucose Levels: data for the user's blood sugar glucose in mg for dL of blood.
- O2 Levels: data for the user's oxygen levels in perceptual values.
- Heart Rate: data for the user heart rate in bpm.
- Weight: data for the user weight in Kg.
- Body Temperature: data for the user temperature in Celsius degrees.
- Bed Sensor: data for the user presence in bed – true or false.

5.4.1.2 Home monitoring

The application provides a house monitoring service. This service enables data collection and threshold configuration for the following sensors:

- Humidity: data for the room humidity levels in perceptual values.
- Temperature: data for the room temperature in Celsius degrees.
- CO Levels: data for the room carbon monoxide levels in perceptual values.
- CO2 Levels: data for carbon dioxide levels in perceptual values.
- Bathroom: data for the Bathroom water usage – true or false
- Kitchen: data for the kitchen water usage – true or false

5.4.2 *Email*

The email service is provided by the google Gmail service. This provides security and a reliable email management.

5.4.3 Alerts

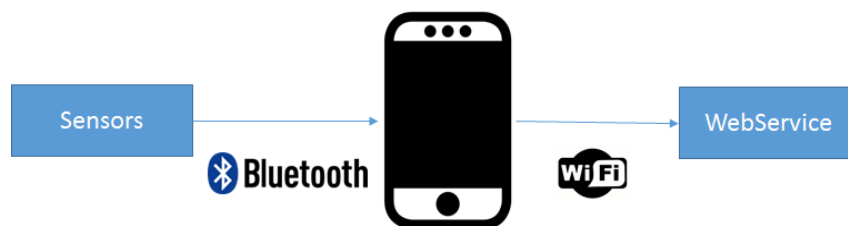
If an irregular or dangerous situation is detected by the system, an alert message is sent for both the user and his or her guest via an alert message service. This service can be subscribed by family members, caregivers or even a tele assistance provider in order to grant the user a safe net in case of emergency.

5.4.4 Reminders

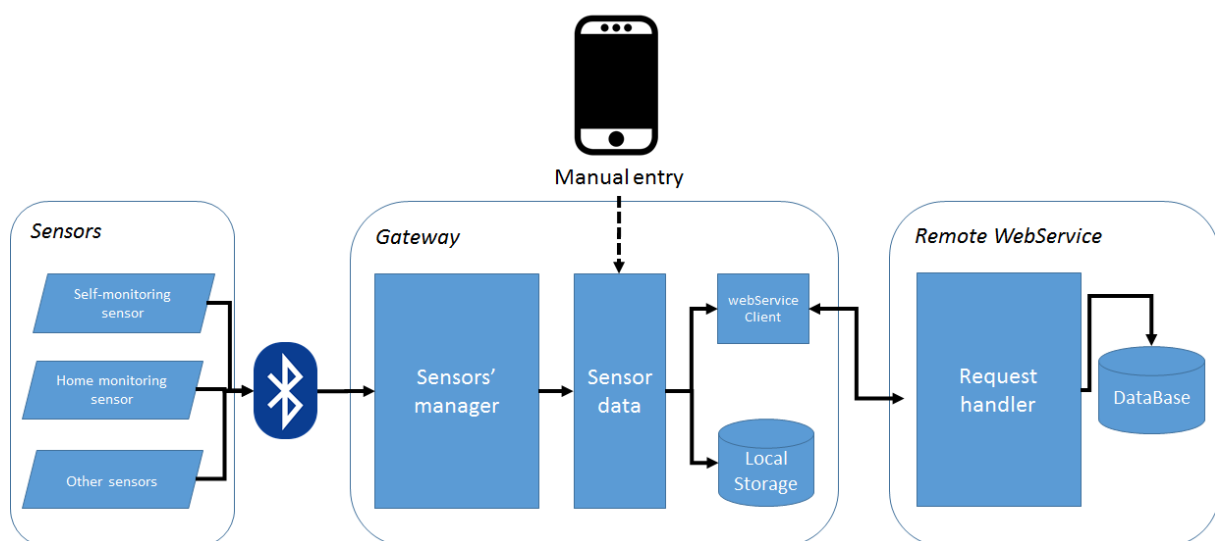
To keep in check and to increase control over the schedule, a reminder service is inbuilt the application. Through this service the user can schedule events, recurring or one time occurrences, and be reminded by the application of its occurrence.

5.5 Integration with sensors

For each type of sensor a distinct class was created to handle communication between the smartphone and the sensor, because each sensor has its communication protocol. In this class were implemented methods for connecting and disconnecting the connection to the sensor and detect if the connection is alive. For this, and knowing that the sensors send data, typically from 1 in 1 second, runs one minute by minute statement that checks when the last transmission, marking the sensor disconnected as if it has been exceeded a defined time interval. So it is possible to service know every moment the state of connection with the sensors.



For greater flexibility is also provided a method of enabling and disabling monitoring certain parameters, so that sensors in more than one type of monitoring is possible to select only the necessary parameters.



6 Integration (D5.3)

6.1 Application access

To access Helascol-webApp the user only needs to open the device browser you are in and access the following address: <http://helascol.no-ip.org>

After entering the website is displayed to the user a page with the button Sign in with Google by clicking the user is redirected to the Google login page where you only need to enter the email address and the corresponding password to your Google account and is able to browse the application. If you do not have a Google account you must register.



Figure 18: Login page

6.1.1 Description of Application



The homepage presents the sections Personal and Social. Each section has the icons to access the different features. To do this, simply click the desired icon to be directed to the page with the desired functionality



Figure 19: HomePage

6.1.2 Navigation Bar

On every page is in the top navigation bar. The current date and time are displayed on the top left and right side will indicate "My Account". To switch users, click the arrow ▼ and choose another user to which you have access. Through this list also access the application output option (logout).

By clicking the button  you will be redirected to the profile page. Although the right of the navigation bar, the button appears with the heart rate information by clicking directs for the Heartbeat page. The button for the room temperature  directs for Environment Monitoring page.


On the left side of the navigation bar is the back button . This button must be clicked each time you want to return to the previous page.



Figure 20: Navigation Bar

6.1.3 Notifications

This bar shows all notifications that are sent to the user: reminders, alerts, contacts, messages and chat invitations. When a new alert is triggered, appear a new bar with warning message.

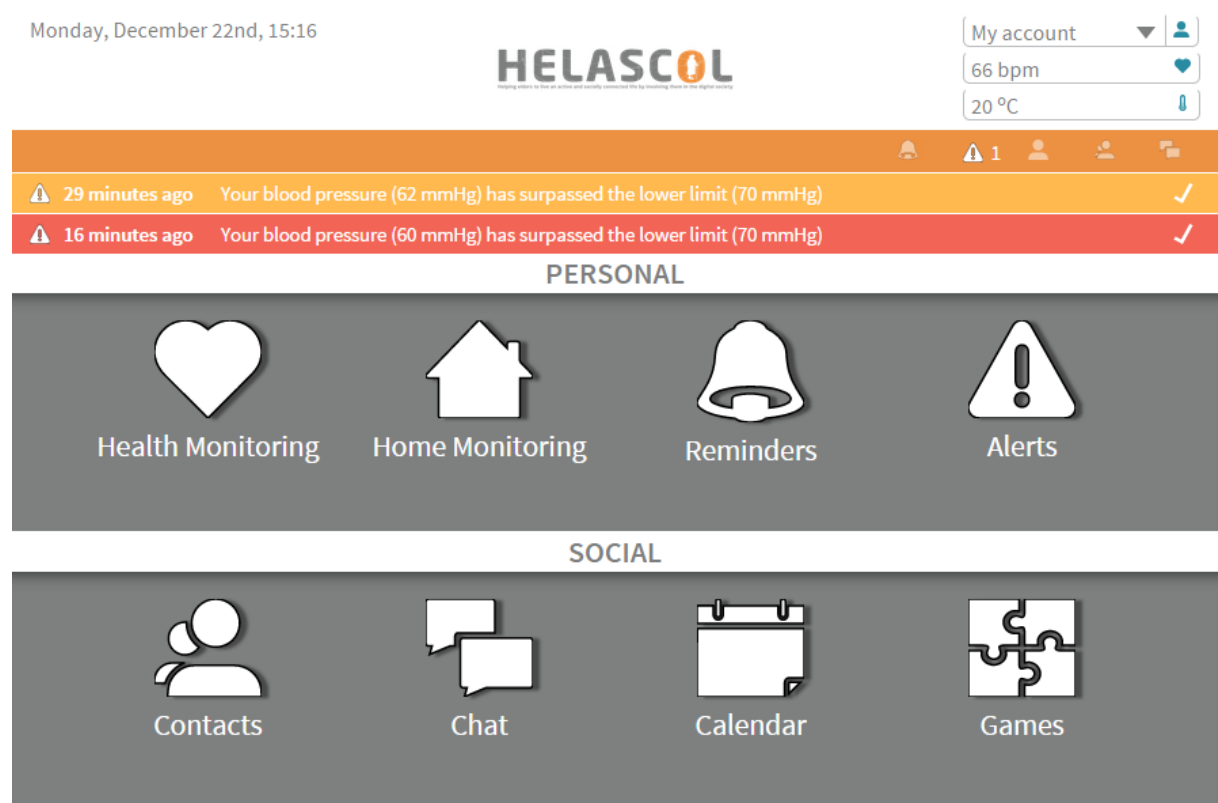


Figure 21: HomePage - Notifications

6.1.4 Profile

On this page the user has access to your information (name and email) and even the option to click the Settings icon and access the settings page.

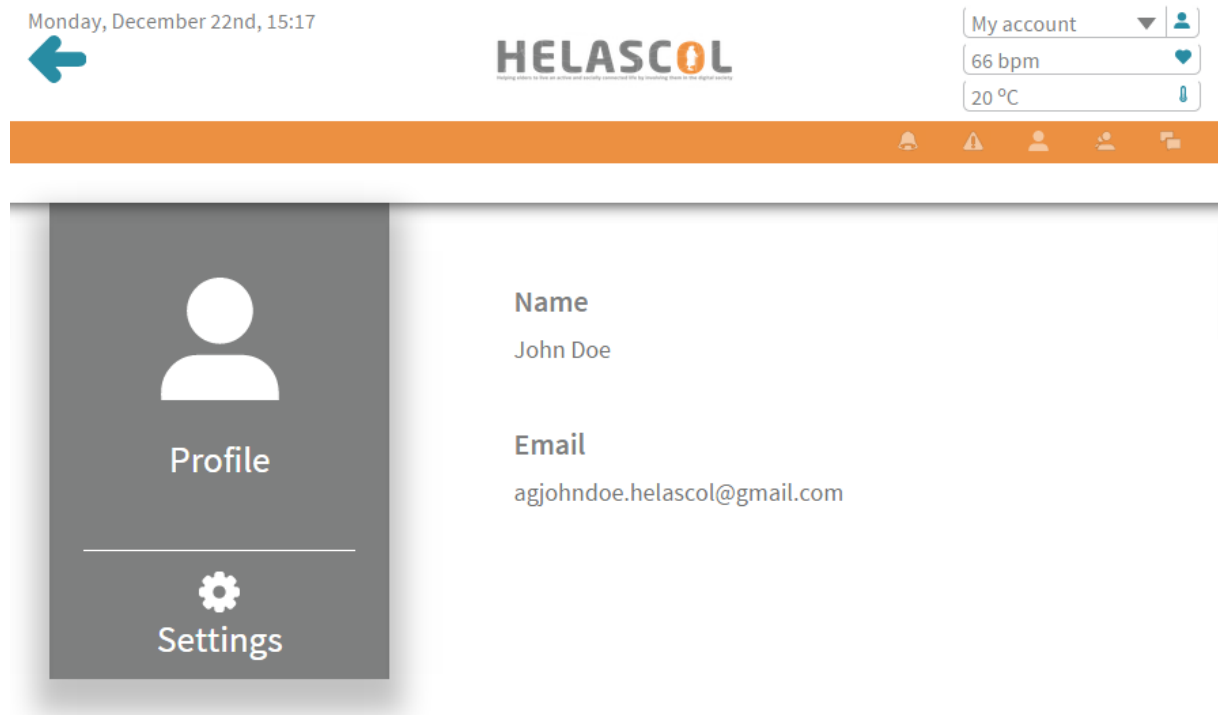


Figure 22: Profile page

6.1.5 Settings

On the Settings are presented four options, the possibility of language change by selecting Portuguese or English, edit the ID and write or add guests invited by clicking the (+) button or remove guests through the (x) button.



Figure 23: Profile settings page

6.1.6 Home Monitoring

On the Home Monitoring are presented to the user several options monitoring of environmental variables: humidity, temperature, CO and CO2. Each of these options directs the user to the respective page.

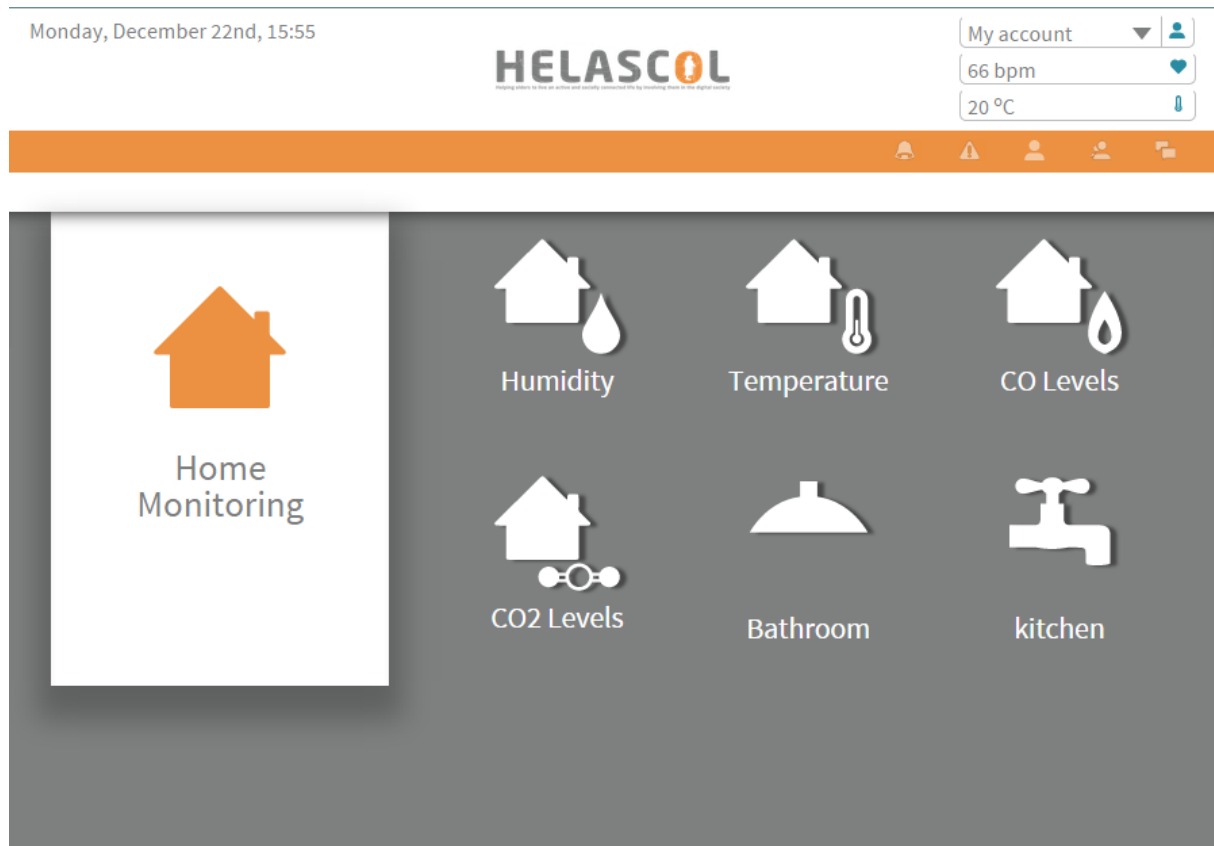


Figure 24: Home monitoring page

6.1.7 Health Monitoring

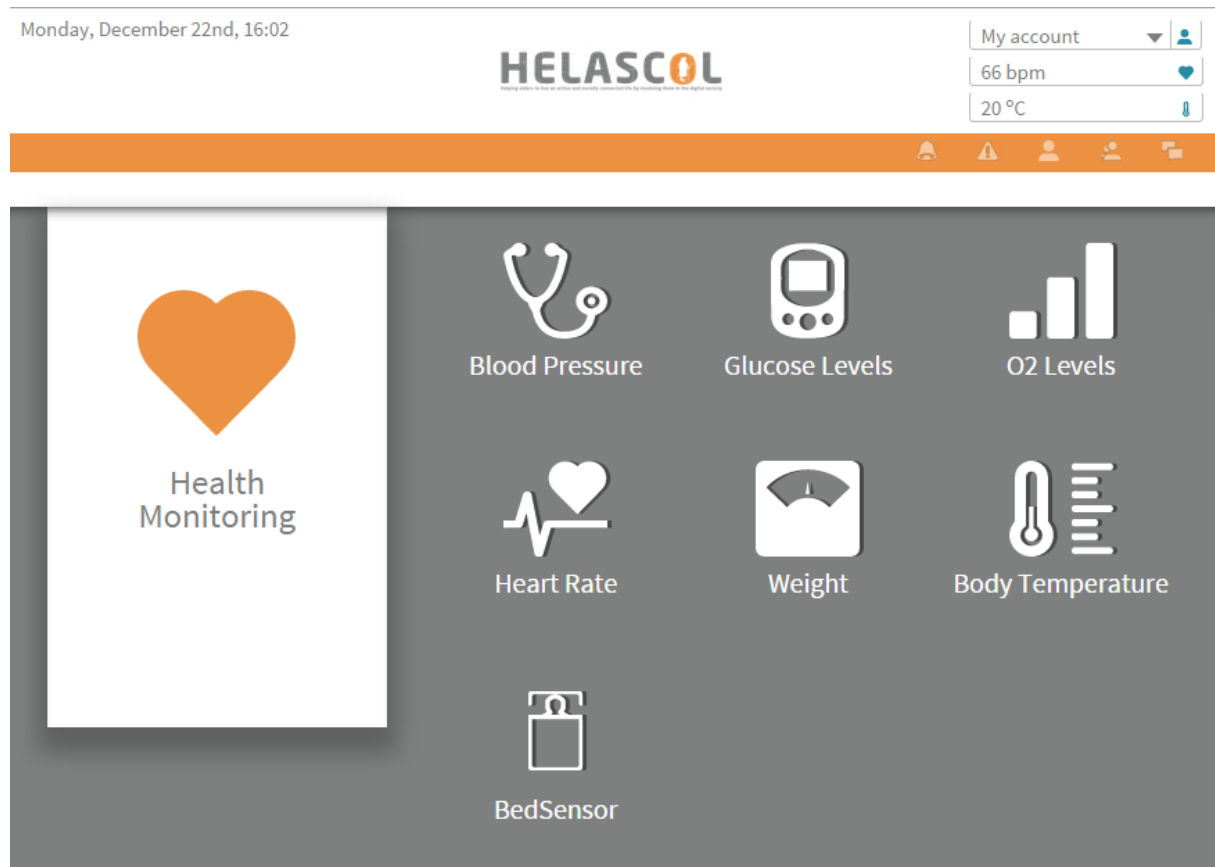


Figure 25: Health monitoring page

On this page you can access to blood pressure options, glucose level, oxygen level, heart rate, weight and body temperature. Also, a bed presence can be monitored.

All these options are accessed by clicking and have very similar sub options. As an example, it is explained how to use the Heart Rate option.

6.1.8 Heart Rate

On the Heart Rate query the last heartbeat value by clicking Last Value. The registration table heart-beat can be query-clicking the Table and clicking Chart see the chart with the registration of the heartbeat. Clicking the pencil button below to Edit Heart Rate page.

The screenshot shows the HELASCOL interface for heart rate monitoring. At the top left, the date and time are "Tuesday, December 30th, 16:05". The HELASCOL logo is in the center. On the top right, there are three status boxes: "My account" with a user icon, "66 bpm" with a heart icon, and "20 °C" with a thermometer icon. Below the header is an orange navigation bar with icons for notifications, alerts, and users. The main content area has three tabs: "LastValue" (selected), "Table", and "Chart". On the left, a grey card displays a heart rate icon, the text "Heart Rate", and an "Edit" button with a pencil icon. On the right, the current heart rate is shown as "66 bpm" in a dark box, with "8 days ago" below it.

Figure 26: Heart Rate monitoring page

6.1.8.1 Edit Heart Rate

On this page you can set the maximum and minimum values for alerts and warning of danger, for this just click the icon (pencil) to edit. After performing the user changes need only click the Save button to save the changes.

The screenshot shows the HELASCOL interface for editing heart rate thresholds. At the top left, the date and time are "Tuesday, December 30th, 16:06". The HELASCOL logo is in the center. On the top right, there are three status boxes: "My account" with a user icon, "66 bpm" with a heart icon, and "20 °C" with a thermometer icon. Below the header is an orange navigation bar with icons for notifications, alerts, and users. The main content area has a grey card on the left with a heart rate icon, the text "Edit Heart Rate", and a pencil icon. On the right, the page is titled "(bpm)" and contains two columns of input fields. The "Alert" column has "Upper Limit" (150) and "Lower Limit" (50). The "Danger" column has "Upper Limit" (160) and "Lower Limit" (40). A "Save" button is located at the bottom right.

Figure 27: Heart Rate thresholds edit page

6.1.9 Reminders

On this page you can view the entire history of reminders that was played through a list with the reminder title the date and time. To add a new reminder the user needs to click the Add button with the + icon and is redirected to the page add reminders.

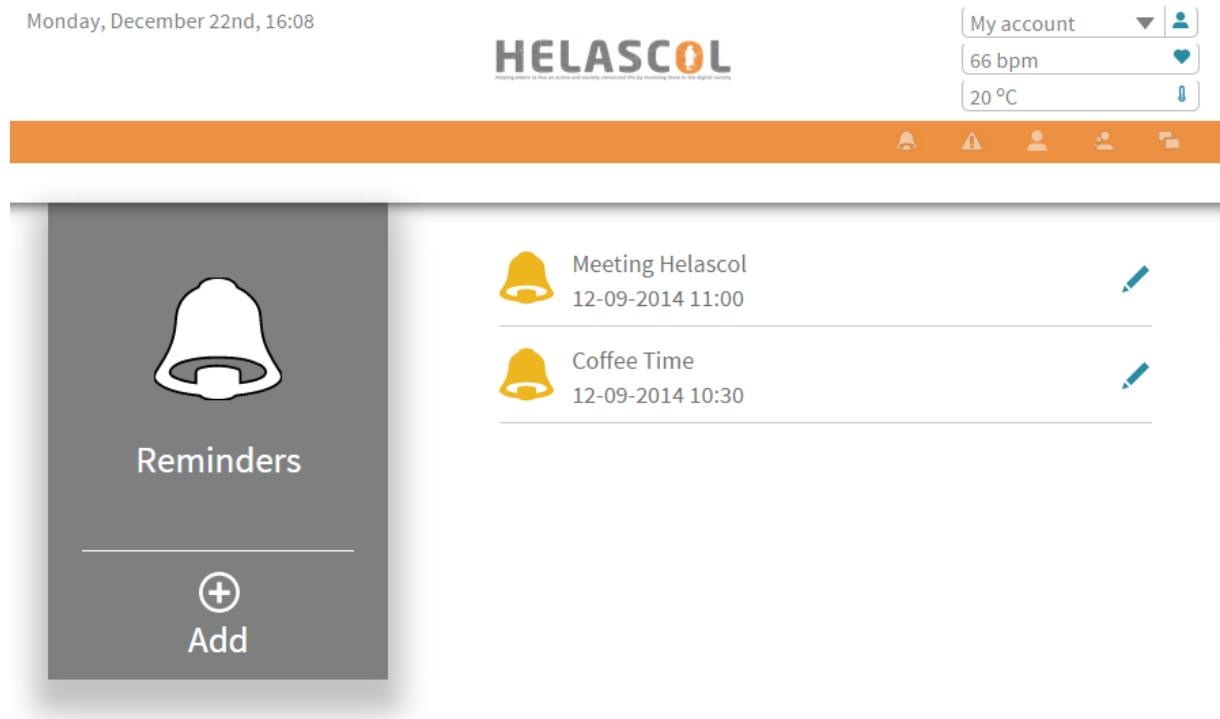


Figure 28: Reminders page

6.1.9.1 Add Reminders

Type the reminder text in the Subject text box and enter the date and time in the text boxes Date and Time, respectively. In the Repeat box must click on to open the list of options and choose one. To validate the reminder click the Add button.

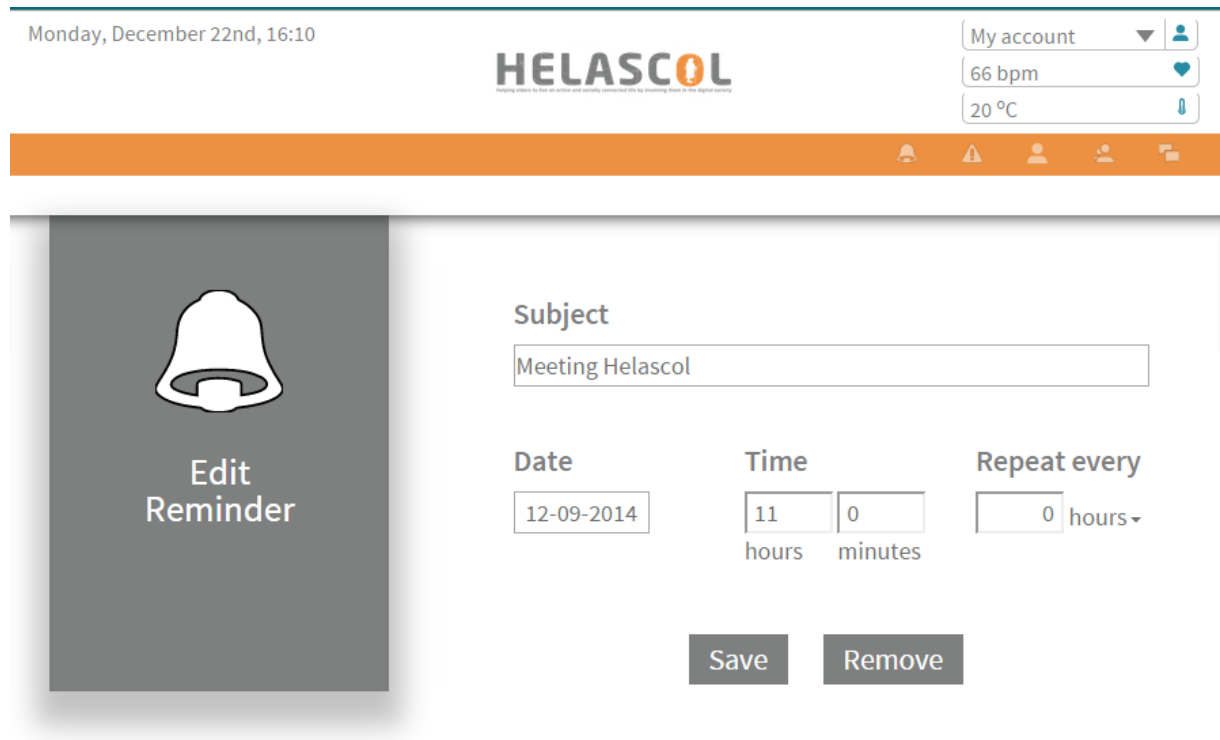


Figure 29

6.1.10 Alerts

In the Alerts page you can access the entire history of orange alerts and warnings see melhos which are differentiated by color alert icon or orange or red. Both have the date the alert description and the time that was played. Each alert contains an icon and is separated from other alerts through separation lines.

Monday, December 22nd, 16:11

HELASCOL

My account 66 bpm 20 °C

Alerts

- Your heart rate (69 bpm) has surpassed the lower limit (70 bpm) 4 months ago
- Your heart rate (62 bpm) has surpassed the lower limit (65 bpm) 4 months ago
- Your house temperature (15 °C) has surpassed the lower limit (17 °C) 3 months ago
- Your house temperature (16 °C) has surpassed the lower limit (17 °C) an hour ago
- Your blood pressure (62 mmHg) has surpassed the lower limit (70 mmHg) an hour ago
- Your blood pressure (66 mmHg) has surpassed the lower limit (70 mmHg) an hour ago

Figure 30: Alerts page

6.1.11 Contacts

On the Contacts the user can view your contacts and have the possibility of using the icon (cross) eliminate the contact you want to edit or through the pencil icon to edit the contact by clicking direct the user to the Edit Contacts page. To add a new contact will have to click the button with the (+) icon that will redirect you to the Add Contacts.



Figure 31: Contacts page

6.1.11.1 Edit Contacts

To edit a contact the user selects to change (either name or email) and make desired changes and click the Save button to save the changes. To delete a contact, select the name to delete and click the Remove button.

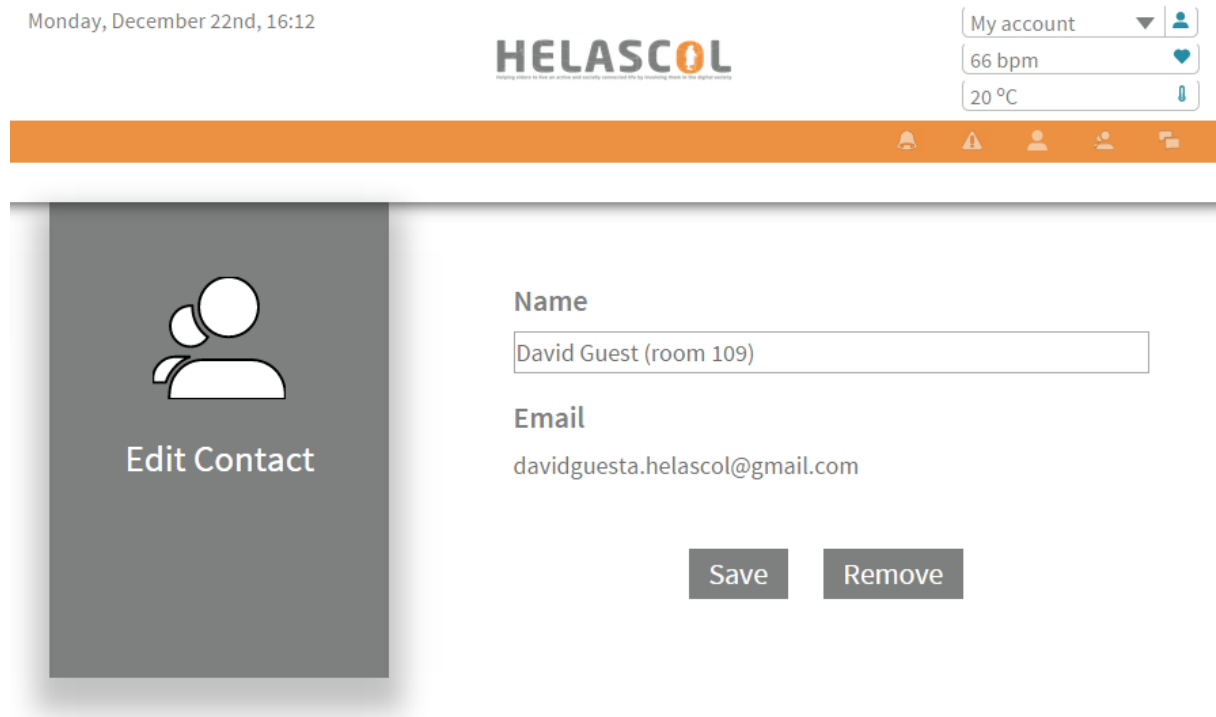


Figure 32: Edit contact page

6.1.11.2 Add Contacts

To add a contact the user only needs to enter the name and email address and click the button Add.

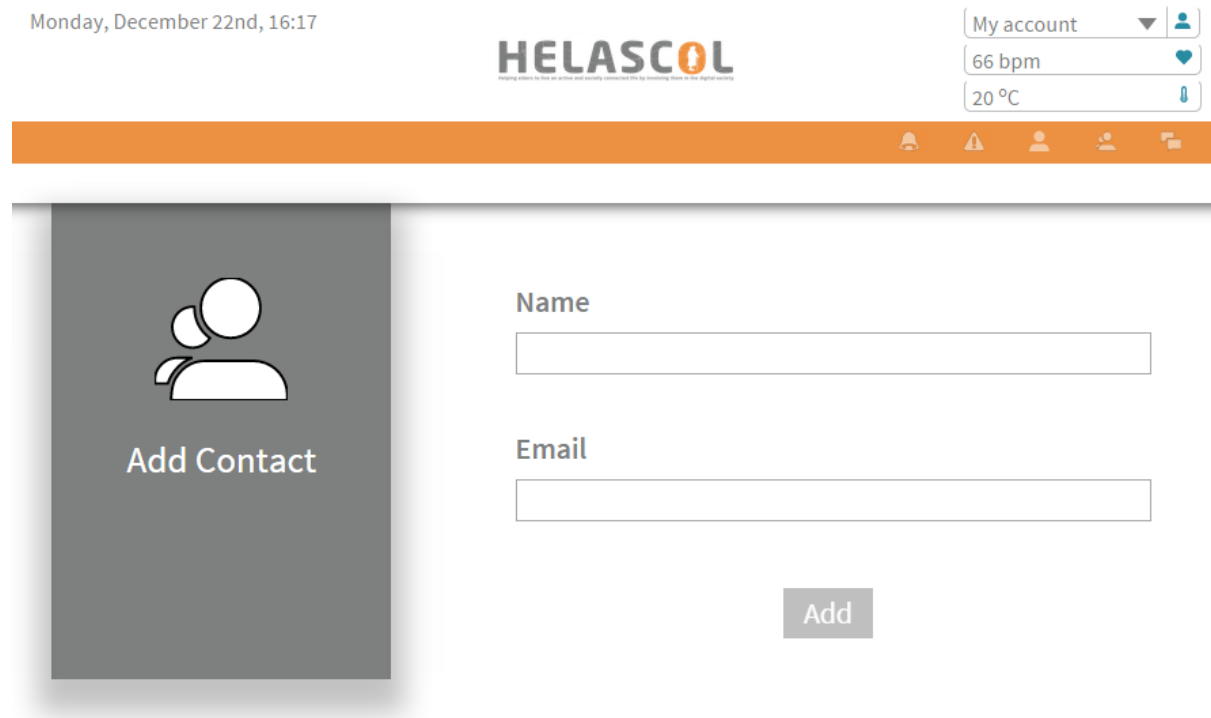


Figure 33: Add contact page

6.1.12 Chat

To access the chat service must click the option Chat and select the desired contact.

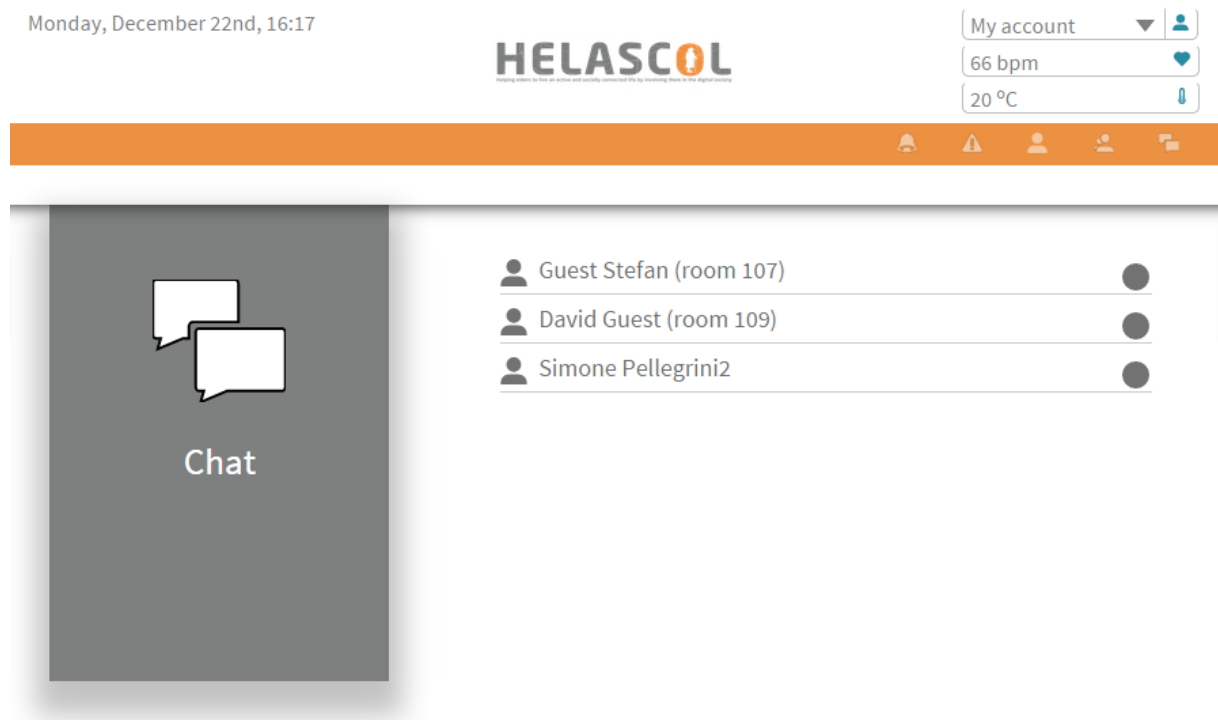


Figure 34: Chat page

If the contact is logged in, a green dot will be shown next to its name, otherwise the dot will be gray.

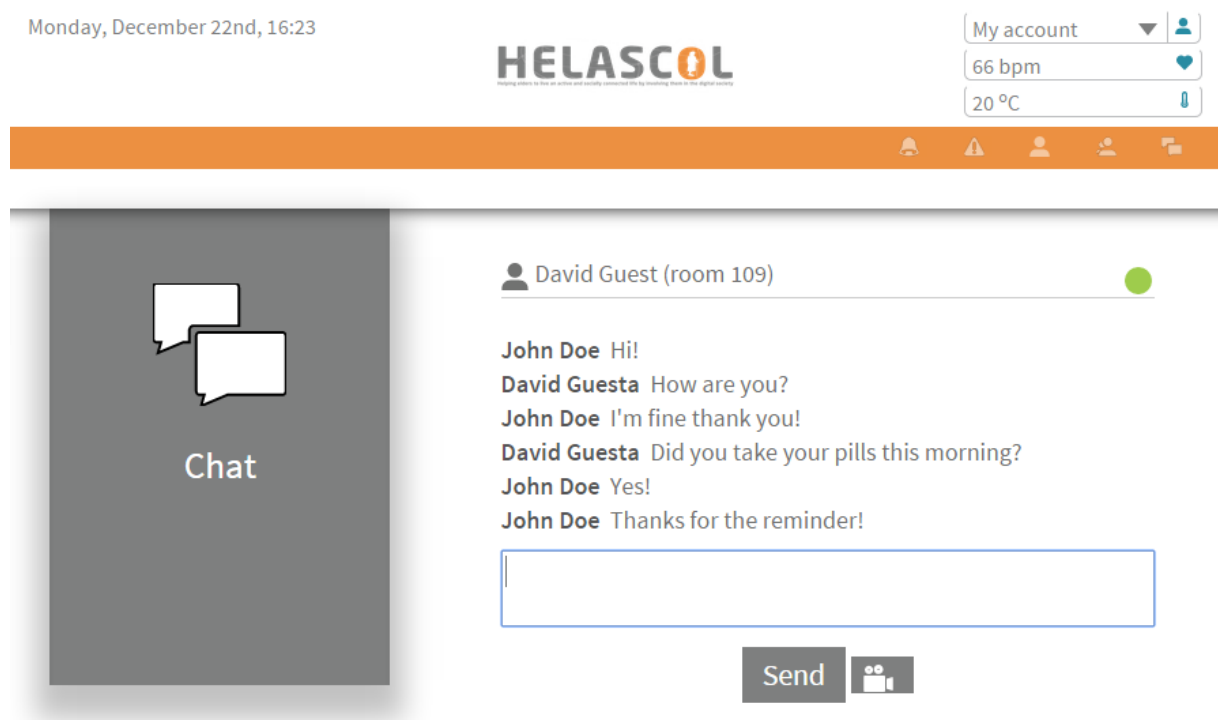


Figure 35: Chat page – Live conversation

In order to start a video call, push the video button. This will activate the webcam and microphone if detected. To end the call just push the close video button.

6.1.13 Calendar

Clicking the Calendar icon on the HomePage page accesses the Calendar page. Here the user can add a public agenda through the Add button and look at the available schedules. The user can also remove an agenda by clicking the unsubscribe bottom.

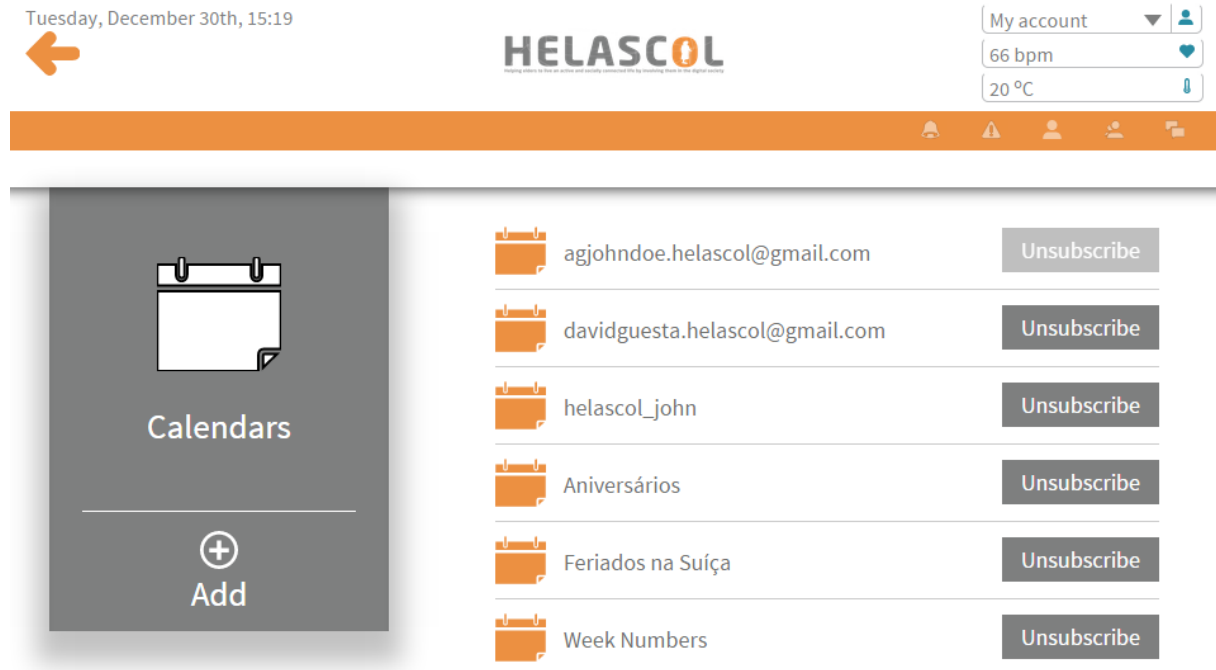


Figure 36: Calendar page – calendar list

To add a new calendar, the user must add the calendar id. This id is generated on the google calendar page (<https://www.google.com/calendar>) on calendar settings, next to calendar address. For a guest calendar, just enter the guest’s email address as the calendar id. If the guest publishes the calendar as public, access to this calendar events will be granted.

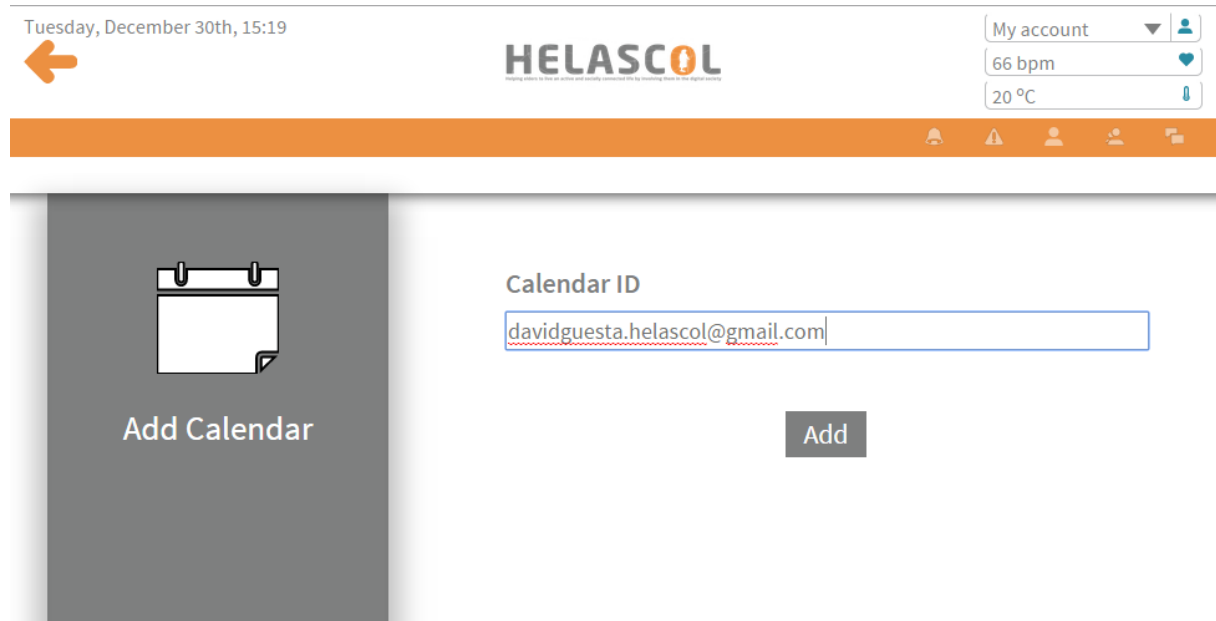


Figure 37: Add calendar page

On the calendar page, the user can access the calendar’s detailed view. This presents an “open calendar” bottom and a list of the next events on the calendar. By clicking the “open calendar” bottom, the user can navigate on the calendar to view past events.



Figure 38: Calendar view page

6.1.14 Games

Clicking the Games icon accesses the games page. To choose a game to click the game icon you want.

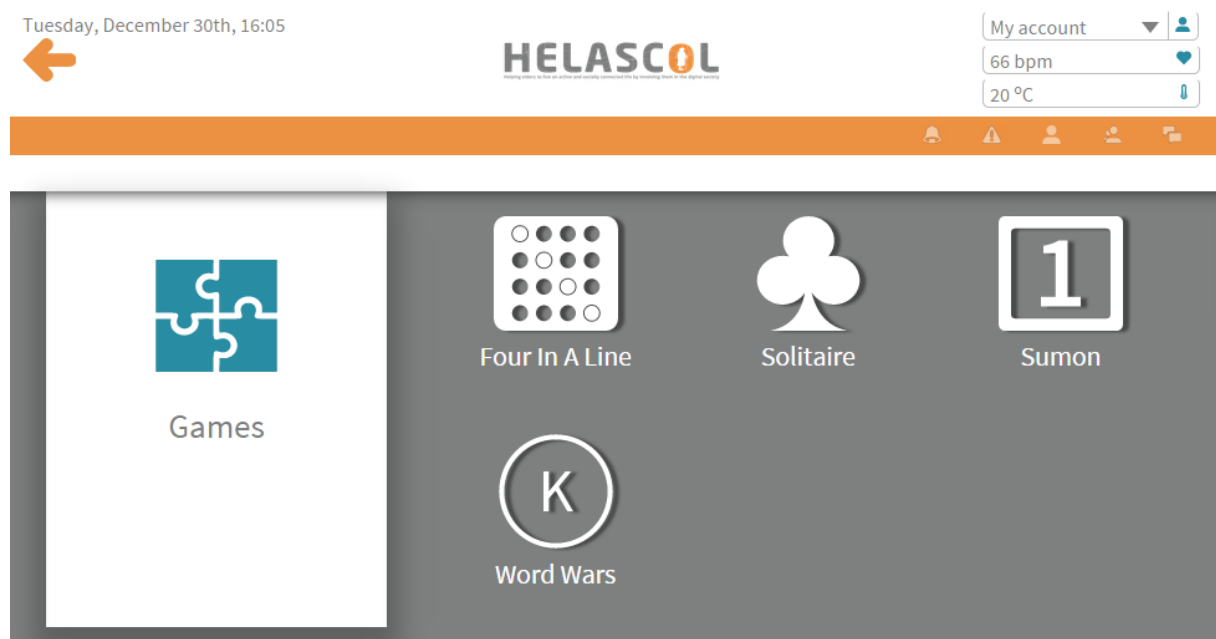


Figure 39: Games page

7 Conclusions

The developed components are fully integrated through established communication lines and standard protocols such as Rest web services and other communication protocols. These components were built to be easy to expand granting this framework a greater longevity and growth to answer other problems or to provide other services.

Two field pilots will take place in

8 References

A. Wood, G. Virone, T. Doan, Q. Cao, L. Selavo, Y. Wu, L. Fang, Z. He, S. Lin, J. Stankovic, "ALARM-NET: Wireless Sensor Networks for Assisted-Living and Residential Monitoring", Technical Report CS-2006-13, University of Virginia, 2006.


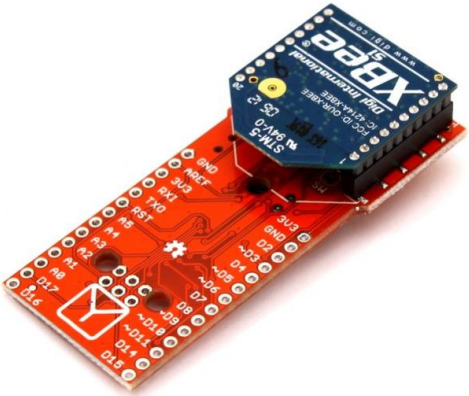
H. Huo , Y. Xu , H. Yan , S. Mubeen and H. Zhang "An elderly health care system using wireless sensor networks at home", Proc. 3rd Int. Conf. Sensor Technologies and Applicat., pp.158 -163 2009.

Suryadevara, N.K.; Mukhopadhyay, S.C. "Wireless sensors network based safe home to care elderly people: A realistic approach", *Recent Advances in Intelligent Computational Systems (RAICS), 2011 IEEE*, On page(s): 001 - 005

Wood A., Stankovic J., Virone G., Selavo L., Zhimin He., Qihua Cao., Thao Doan., Yafeng Wu., Lei F., Stoleru R., "Context-aware wireless sensor networks for assisted living and residential monitoring", *IEEE Network-2008*, Vol:22, No:4, Page(s): 26 – 33.

<http://bradfrostweb.com/blog/web/mobile-first-responsive-web-design/>. Accessed on March 5, 2014.

9 Annexes

|  | HELASCoL Sensor Network | Sensor Node |
|-----------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------|
|  | <p>Short name: HSN-SN</p> <p>Features The HELASCoL Sensor Node is used to interface sensors and/or actuators. It has connections for a Lithium Polymer battery and includes a charge circuit over USB. An XBee radio module is available on the bottom of the board. The Fio has been designed to be wirelessly reprogrammable.</p> <p>Summary ATmega328V microcontroller running at 8MHz XBee radio module 2mW output (+3dBm) 250kbps Max data rate MCP73831T LiPo Charger 14 digital I/O pins 8 analog inputs pins 32 KB Flash memory (of which 2 KB used by bootloader)</p> | |
| Hardware components | Arduino FIO | Version: 2 |
| | XBee 2mW Wire Antenna | Version: Series 2 |
| Firmware components | Sensor_node1_bed | Version: 1 |
| | xb24-802.15.4-10e6-node-57600 | Version: 1 |
| Power Supply | Polymer Lithium Ion Battery - 850mAh | |
| Sensor | Force Sensor | |
| Mechanical dimensions | 66 x 28 x 40 mm | |
| Accessories | Mini-USB cable, Xbee Explorer USB | |
| System Specification: Bed sensing | <p>Refers to the HSN-SN implementation. The node receives the values read from the force sensor. Depending of the threshold determines the presence of the person on the bed. Subsequently, a notification with a '1' or '0' is sent to central node (Arduino YUN) in order to update the HSN-SN current state. The central node will be in charge to communicate this information to the webservice. In the Appendix A is described the communication protocol.</p> | |
| Firmware | <p>This section is referred to the program running on the Bed Sensor, is the core of the sensor node (Arduino Fio). Describes the way of the required features has to be implemented. Features include:</p> <ul style="list-style-type: none"> • Power Up event • Battery level event • Alive event (HeartBeat) | |

| | |
|----------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Notes | The miniUSB connector is used for battery charging only. To bootload a new firmware, you will need an external serial connection over an FTDI Basic, FTDI cable, or other serial connection. Check the Appendices B . |
| Hardware for the complete system | <ul style="list-style-type: none">▪ Arduino FIO: Sparkfun DEV-10116▪ XBee 2mW Wire-Antenna, Series 2▪ Force Sensor▪ Polymer Lithium Ion Battery 850mAh:▪ USB Cable - A-to-miniB 6 Foot Retail: Sparkfun RTL-11302 |

10 Appendices

A. Communication Protocol

Message pattern (8 bytes, 64 bits):

<ACK> <RID> <NID> <SID> <MID> <EID> <CHA> <VAL>

ACK : Acknowledge:

0x06 - Start-of-message

RID : Room ID, unique room (or network) identifier inside the same infrastructure:

0x00 - Reserved to identify the server itself (local room)

0xFF - Reserved for broadcast (discovery).

NID : Node ID, unique node identifier inside the same room (or network):

0x00 - Reserved to identify the bridge itself (local node).

0xFF - Reserved for broadcast (discovery).

SID : Sensor ID, unique sensor identifier inside the same node:

0x00 - Reserved to identify the node itself (local sensor).

0xFF - Reserved for broadcast (discovery).

0x01 - GPI (digital)

0x02 - GPO (digital)

0x03 - GPI (analog)

0x04 - GPO (analog)

0x05 - Shaker (Z-only accelerometer)

MID : Message ID, unique message identifier:

0x00 - Reserved

0x80 - Event (after a state change)

EID : Event ID, identifier of an event generated by the specific sensor:

EID | SID

0x00 | 0x00 : Test value:

CHA - 0xNN - Any value

VAL - 0xNN - Any value

0x00 | 0x?? : Reserved:

CHA - 0x00

VAL - 0x00

0x01 | 0x00 : Power-up or system reset:

CHA - 0x00

VAL - 0x00

0x01 | 0x01 : State changed:

CHA - 0xNN - Number of the affected input

VAL - 0x00 - Button released

0x01 - Button pressed

0x01 | 0x05 : Shake detected:

CHA - 0x00

VAL - 0xNN - Current shake threshold (62.5mG/LSB)

0x02 | 0x00 : Heart beat:

CHA - 0x00

VAL - 0x00

0x02 | 0x05 : No shake after timeout:

CHA - 0x00

VAL - 0xNN - Current shake timeout (1s/LSB)

0x03 | 0x00 : Battery low:

CHA - 0x00

VAL - 0xNN - Charge level (%)

CHA : Channel, the number of the affected channel (if any).

VAL : Value, the value returned (if any).

B. Installation Procedure

Hardware:

- Arduino Fio v1
- Xbee radio, Series 2
- Force Sensitive Resistor (square shape)
- Battery rechargeable 850mAh
- Mini Usb cable
- Xbee explorer USB



Steps:

- Verify all components are properly powered and include the xbee radio.
- Open the file code: **sensor_node1_bed.ino** with Arduino 1.5.5 sketch .
- Connect the Xbee explorer USB to the PC
- In the Arduino sketch verify the following settings:
 - ✓ Tools→ Board→ **Arduino Fio**
 - ✓ Tools→Port → **COM5 (depends which port is assigned)**
- Upload it code
- Finally, open the serial monitor to visualize the data send from the Force sensor.

Recommendation:

Upload the firmware on Arduino with FTDI Basic

The XBee socket and FTDI connection live on the same TX/RX pins of the ATmega328. You will need to remove the XBee module while reprogramming over serial. We recommend using a wireless boot-loader whenever possible to avoid this step.

|  | HELASCoL Sensor Network | Gateway |
|-----------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------|
|  | <p>Short name: HSN-GW</p> <p>Features The HELASCoL Sensor Network Gateway is a board based on the ATmega32u4 and the Atheros AR9331. The board has built-in Ethernet and WiFi support, a USB-A port, micro-SD card slot, 20 digital input/output pins (of which 7 can be used as PWM outputs and 12 as analog inputs), a 16 MHz crystal oscillator, a micro USB connection, an ICSP header, and a 3 reset buttons.</p> <p>Summary >> ATmega32u4 Microcontroller running @16 MHz 20 Digital I/O Pins 7 PWM Channels 12 Analog Input Channels 32 KB Flash Memory (of which 4 KB used by bootloader)</p> >> Linux Microprocessor Atheros AR9331 @400MHz Ethernet: IEEE 802.3 10/100Mbit/s WiFi: IEEE 802.11b/g/n PoE Compatible 802.3af USB Type-A 2.0 Host Micro-SD Card Reader 64 MB DDR2 RAM, 16 MB Flash Memory | |
| Hardware components | Arduino Yún | Version: |
| | XBEE Shield Mod. O for Arduino | Version: - |
| | XBee 2mW Wire Antenna | Version: Series 2 |
| Firmware components | Sensor_node0 | Version: 0.1 |
| | xb24zb-zb-coordinator-at-20A7-57600 | Version: 0.1 |
| Power Supply | Wall Charger - 5V USB (1A) | |
| Mechanical dimensions | Length 73 mm, Width 53 mm | |
| Accessories | AC/DC Power Supply, Micro-USB cable | |
| Order codes | Arduino Yún: Sparkfun, DEV-12053 XBee Shield: Sparkfun WRL-12847 XBee 2mW Wire-Antenna: Sparkfun WRL-10414 Wall Charger - 5V USB (1A): Sparkfun TOL-11456 USB microB Cable - 6 Foot: Sparkfun CAB-10215 Headers supplementari per shield | |