Funded by EU's Horizon 2020

D 7.1

SAAM Components Verification and Integration Report

Supporting Active Ageing through Multimodal coaching

DISCLAIMER

This document reflects the opinion of the authors only and not the opinion of the European Commission. The European Commission is not responsible for any use that may be made of the information it contains.

AKNOWLEDGEMENT

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No.769661.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 769661

Supporting Active Ageing through Multimodal coaching

DELIVERABLE DOCUMENTATION SHEET

Deliverable:	*D 7.1 – SAAM Components Verification and Integration Report*
WP №	*7*
Title:	*Integration, Verification, and Security*
Research team:	*Klemen Bregar, Matevž Vučnik, Mihael Mohorčič, Timotej Gale, Marko Mihelin, Claus Pribbernow, Andrej Čampa, Miha Mohorčič, Irina Stoyanova, Anton Biasizzo, Andrej Hrovat*
Type:	*REPORT*
Version:	*V1.0*
Author Report:	*Klemen Bregar*
Delivery Due Date:	*31.3.2020*
Dissemination level:	*Public*
Copyright:	*©SAAM Project - All rights reserved*

☒	Approved by the WP Leader
☒	Approved by the Technical Manager
☒	Approved by the Coordinator
☒	Approved by the PSC

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 769661

PUBLISHABLE SUMMARY

System component integration extends the work done at system component verification stage and can be split into three integration testing steps. In the first step interaction between the device and system component is tested. The second step contains a complete system integration testing in a controlled simulated environment which resembles deployment scenarios. The last step is performed at each deployment site where it is confirmed that the deployed system components are operating in accordance with the specifications. When all user-side devices and components are deployed at the location, a complete integration testing of user-side components is done using deployment and integration testing application running on an eGW. This deliverable presents work done behind those three steps.

After the description of the infrastructure components verification and integration procedures in Section 2, the user-side device development, production and assembly testing are explained in Section 3. The document continues with the description of a purposely developed web application for user-side device integration and deployment testing in Section 4, which is followed in Section 5 by a detailed description of the integration tests. Section 6 describes pre-deployment integration testing during the individual deployment packages preparation. In Section 7 the user-side device deployment procedure is given and user-side device health monitoring system is described in Section 8.

This project has received funding from the European Union's Horizon 2020
research and innovation programme under grant agreement No 769661

QUALITY CONTROL ASSESSMENT SHEET

Version	Date	Comment	Name of author/reviewer/contributor
V0.1	2.3.2020	First Draft	Klemen Bregar
V0.2	16.3.2020	Contributions	Matevž Vučnik, Timotej Gale, Claus Pribbernow, Miha Mohorčič, Anton Biasizzo, Andrej Čampa, Marko Mihelin, Jakob Jenko, Ivan Boškov
V0.3	19.3.2020	Second Draft	Andrej Hrovat,
		Contributions	Doroteya Dineva
V0.4	13.05.2020	Third Draft	Petar Yordanov
	18.05.2020	Peer review	Bernard Ženko
	18.05.2020	Peer review	Vera Veleva
	20.05.2020	PSC/Coo consultation	Returned for additions
V0.5	25.09.2020	Version for approval	Irina Stoyanova, Petar Yordanov
	26.09.2020	WP Leader approval	
	30.06.2020	Coordinator approval	
		EAB Review	N.a.
	7.10.2020	PSC approval	
V1.0	14.10.2020	Submission to EC	

HISTORY OF CHANGES

For updating the Deliverable after submission to the EC if applicable

Version	Date	Change

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 769661

Supporting Active Ageing through Multimodal coaching

PROJECT DOCUMENTATION SHEET

Project Acronym:	SAAM
Project Full Title:	Supporting Active Ageing through Multimodal coaching
Grant Agreement:	GA № 769661
Call identifier:	H2020-SC1-2017-CNECT-1
Topic:	Personalised coaching for well-being and care of people as they age
Action:	Research and Innovation Action
Project Duration:	44 months (1 October 2017 – 31 May 2020)
Project Officer:	Jose Albacete VALVERDE
Coordinator:	Balkan Institute for Labour and Social Policy (BILSP)
Consortium partners:	Jožef Stefan Institute (JSI) University of Edinburgh (UEDIN) Paris-Lodron Universität Salzburg (PLUS) Scale Focus AD (SCALE) Interactive Wear AG (IAW) Univerzitetni rehabilitacijski inštitut Republike Slovenije (SOČA) Nacionalna Katolicheska Federacia CARITAS Bulgaria (CARITAS) Bulgarian Red Cross (BRC) Eurag Osterreich (EURAG)
website:	saam2020.eu
social media:	#saam2020, #saamproject

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 769661

ABBREVIATIONS

AP	Access point
ACSI	Automatic coordinate system initialization
CIR	Channel Impulse Response
MDS	Multi-Dimensional Scaling
MQTT	MQ Telemetry Transport
NILM	Non-intrusive Load Monitoring
PCP	Printed circuit board
PDP	Power Delay Profile
PMC	Power Meter and Control
RSS	Received Signal Strength
RSSFP	First path Received Signal Strength
SAAM	Supporting Active Ageing through Multimodal Coaching
UWB	Ultra-Wide Band

This project has received funding from the European Union's Horizon 2020
research and innovation programme under grant agreement No 769661

CONTENTS

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 769661

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 769661

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 769661

TABLE OF FIGURES

ANNEXES

This project has received funding from the European Union's Horizon 2020
research and innovation programme under grant agreement No 769661

1. INTRODUCTION

With the increasing number of system components the susceptibility of the system to various malfunctions increases, which may eventually reduce or disable the operation of the entire system. Systems, especially high complexity system such as SAAM, where different user-side devices are connected into a cloud structure with bidirectional interaction, depend on reliable operation of each individual system component. The reliability of services provided by such a system, in our case the SAAM coaching actions and monitoring features, depend on reliable operation of the system components underneath.

To limit the amount of possible system malfunctions, special attention was given during the development, production and deployment phases of the devices and services. In order to ensure proper operation of devices, decrease the probability of hidden bugs and guarantee correct system operation, all the system components were tested and verified individually as well as after the integration. Each SAAM user-side device and component was verified for the compliance with the specifications several times during development and production process. It starts with the production testing for defects in the electronics and development tests for early detection of bugs in component's software and firmware and ends by a complete device verification.

System component integration extends the work done at system component verification stage and can be split into three integration testing steps. In the first step interaction between the device and system component is tested. An example is a user-side integration test where all communications and interactions between eGW and a sensor are tested or a testing procedure is made that verifies all interactions between the eGW software component and the cloud server for storing the measurement data. The second step contains a complete system integration testing in a controlled simulated environment which resembles deployment scenarios. In order to verify the integration of the whole system, testing procedures can be based on simulated interaction using synthetically generated data. The last step is performed at each deployment site where it is confirmed that the deployed system components are operating in accordance with the specifications. When all user-side devices and components are deployed at the location, a complete integration testing of user-side components is done using deployment and integration testing application running on an eGW. Deployment testing enables the technician to verify the correct operation of the devices or to replace malfunctioned devices if necessary.

This document is organized in several sections. After the description of the infrastructure components verification and integration procedures in Section 2, the user-side device development, production and assembly testing are explained in Section 3. The document continues with the description of a purposely developed web application for user-side device integration and deployment testing in Section 4, which is followed in Section 5 by a detailed description of the integration tests. Section 6 describes pre-deployment integration testing during the individual deployment packages preparation, while in Section 7 the user-side device deployment procedure is given. User-side device health monitoring system is described in Section 8.

This project has received funding from the European Union's Horizon 2020
research and innovation programme under grant agreement No 769661

2. INFRASTRUCTURE AND SOFTWARE COMPONENTS VERIFICATION AND INTEGRATION

Infrastructure and software components within SAAM are responsible for the data storage and processing as well as for providing the means to process data, coach and interact with the user. All infrastructure system components are verified and correctly integrated with other components before deployment to prevent performance and reliability issues.

All devices running in the system produce data and events that have to be collected in a data centre. Data is later accessible to the services and authorized personnel and thus needs to be stored securely and reliably. As the infrastructure components require different criteria for evaluating the test results, specific methods and procedures for each type of testing are used:

- Functional testing - checks if all requirements are implemented according to the requirements specification; the ability to process data according to prescribed rules; data validation; the solution switch correctly from state to state; proper exception handling, error and information message recovery.
- Unit Test – to test an individual unit like method or function with all dependencies mocked up.
- Integration testing – combining different infrastructure components in groups and to expose defects in the components integrated as a group.
- Regression testing – to ensure that no unexpected behaviour of the system's functionalities is introduced after implementing changes to the application.
- Load and performance testing – to ensure that that the application performs to SAAM user expectations (response time, availability, portability, and scalability).
- Acceptance testing – whether a system satisfies the acceptance criteria and to enable the SAAM user to determine whether to accept the system functionalities.

The specific infrastructure components are explained in detail in other deliverables. The components in the cloud-based server are explained in Deliverable 2.1, including data for:

- Load Balancer
- Data Storage/Data Layer
- Application Server
- Microservices
- Platform scalability
- Security

The components of the client-application are explained in Deliverable 2.9. Infrastructure testing is described in Deliverable 2.6. System monitoring verification/integration consists of set of monitoring measures and testing procedures. These procedures include initial testing of the configuration and connectivity of the end-user infrastructure components, APIs Testing, User-side assets monitoring, Alerting management, Hardware monitoring, Resource usage and performance analysis. The SAAM system complexity requires a huge size of development and very complex software and infrastructure specification. Thus, the testing code covers all critical parts of the code. All business requirements and scenarios are covered with tests, e.g. coaching, communication, settings, visualizations, etc. A test-driven development processes with a hierarchical structure is implemented in SAAM software development, so the code is prepared for testing. The development process is consisting of short iterations. The code is simple, each method aims and performs only one action, so single responsibility

This project has received funding from the European Union's Horizon 2020
research and innovation programme under grant agreement No 769661

principle is followed. The code is decoupled so through the dependency injections the needed services are plugged. All code parts are written independently of each other and all of them use abstraction. A container with list of all services is developed and injects the needed abstraction of the specific service.

The testing processes starts form the lowest level of unit testing, goes through integration testing, system testing and acceptance testing. This allows the development team to test all infrastructure components at each level of development. Mocked framework is also used in testing.

The unit tests are written during the coding process of new functionalities development. They are testing the basic components and software units. Such segregation makes a time friendly approach to testing and the result of each unit test can be investigated independently. All results are evaluated after the unit tests were applied, providing another layer of insurance that there will be no issues during each new deploy. All tests are running automatically during the pull requests of the developers. If one or more unit tests fail the pull request is rejected and the code is returned in development phase.

The integration tests are testing group of components corresponding to elements of architectural design or a whole system module to guarantee that the integrated components interact with each other successfully. Integration testing set in SAAM server- side system is at high level. The integration tests are created in more complex way with preparation of staging with input test data, sample scenario, expected results, and mocked up connections. It gives more sophisticated results as these tests are based on business logic level. All tests' results are checked and verified. If one or more of the integration tests fail, the results are analyzed, and respective actions are taken to solve the problem.

System testing is performed on the functional requirements, for example testing a whole pipeline.

The acceptance testing in SAAM's case is considered the first iteration of piloting, during which constant feedback will be gathered to resolve any issues with the Platform. Preliminary acceptance testing is also performed internally within the consortium by the main Social Partners having direct access and functional users in SAAM-platform and providing relevant to the target users feedback.

The verification of the client-application components is completed by:

- Verification of Microservices in server-side through unit and integration tests and for the APIs through Swagger, used to monitor and execute requests. To have a quick, standardized way to validate the status of a service and its dependencies we have introduced HealthCheck API endpoints to each microservice so we can check the service status, performance information, such as component execution times or downstream service connection times.
- Verification of MSSQL and MongoDB: Mongo uses a set of write concerns to ensure that before sending acknowledgment to the client - the data was written successfully, and the changes were committed to multiple nodes (the copy of the data must be written in more than 50% of the nodes before sending acknowledgment).
- Verification of RabbitMQ - Prometheus is used to collect and save data for RabbitMQ and MongoDB and it is the core component used for alerting. Grafana visualizes the system resource usage and performance data.

This project has received funding from the European Union's Horizon 2020
research and innovation programme under grant agreement No 769661

- Verification of logging through Elasticsearch, Fluent Bit, Kibana - all the events that are triggered and processed are ingested by Fluent Bit. Fluent Bit is installed as a Kubernetes daemon set (one instance of FB per Kubernetes node), the data is transformed and sent to Elasticsearch and Kibana used to visualize the data with charts and graphs. The website uses verified and trusted TLS web certificate. The Kubernetes is configured to automatically renew the old certificate from verified certification authority.

The components of the client-application are also tested via a similar hierarchical approach, using Jasmine and Karma (tools/frameworks used to test Angular):

- Initially through unit tests where angular components are tested in isolation without template and services.
- After that, integration tests are implemented for testing the client-application component along with its template.
- Concluding the process with an end-to-end test scenario to test the entire functionalities sets and the application in its entirety.

2.1 Test plan

This section describes the Test plan that has the following elements: test scope, test strategy and resources for carrying out testing of the SAAM Platform and Mobile App within the duration of the project, explained above in the text.

The Test Plan for the **SAAM-platform** project supports the following objectives:

- Describe testing process
- Provide brief review of project goals and functionality and identify the components that should be tested
- Identify the required technical resources
- Specify team member responsibilities
- Provide an estimation of test
- Identify procedures for test execution and reporting
- List the recommended requirements for test activities (high level)

In the scope of testing will be included all features described in Objectives sections below that will be developed by ScaleFocus team.

2.2 Test strategy

The purpose of the testing activities is to identify potential issues, related to development process and to ensure that they are correctly fixed. Test analysis and planning is based on provided project and product documentation. The testing is focused on the components and features in the scope and follows prepared scenarios in different forms. Test cases are attached to the related Module, Functionality or User story (task), and their implementation is based on Annex 1.

This project has received funding from the European Union's Horizon 2020
research and innovation programme under grant agreement No 769661

If testing of newly created or already existing features is required to be performed on desktop browsers and mobile devices, the team defines the testing strategy for the release depending on the changes of the system.
In the following sub sections will be explained all types of testing implemented for the SAAM Platform.

2.2.1 Testing on desktop browsers

SAAM Platform functionality on desktop browsers is covered by test scenarios developed for browser specific behaviour. Test scenarios for desktop browsers may differ from test sets for mobile devices, but most of them are similar. Test cases which are browser specific are grouped in test sets for testing on desktop browsers and are available for regression testing of existing features or can be selected for testing of newly developed functionalities.

2.2.2 Testing on mobile devices

SAAM Platform functionality on mobile devices is covered by test scenarios developed for mobile native app and mobile browsers. While testing on mobile browsers is covered mostly by scenarios for desktop browsers, because of similar behaviour on browsers, the test scenarios on the native mobile application may differ from the cases for browsers.

For different behaviour of the SAAM Platform depending on the Android device OS corresponding test scenarios are developed.

Depending on the developed features and release strategy, test sets for mobile native application may be executed separately or parallel with test sets for desktop/mobile browsers.

During the development and testing phase simulators, emulators and physical mobile devices are used to verify the functionality of the system.

Supported Android versions are Android 7, 8, 9, 10.

Depending on business analysis which mobile devices and Android versions are mostly used by the users corresponding devices and OS are selected. The selection includes 5 most relevant devices. The selection list may be expanded if required and the information should be documented in mobile testing cheat-sheet.

Testing on mobile devices is performed on devices connected to Wi-Fi Internet connection and 3G-4G Internet connection.

The testing on mobile devices is performed manually according to the describes test scenarios.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 769661

2.2.3 Functional Testing

2.2.3.1 Objectives

The functional testing presents front end testing of end-to-end business cases and testing of web-services where requests are sent to the endpoints for retrieving or posting information in order to verify the integration between front end and back end systems.

The following properties were checked:

- Requirements verification – all requirements are implemented according to the requirements specification;
- Correctness - ability to process data according to prescribed rules. Controls over transactions and data field edits provide an assurance on accuracy and completeness of data.
- Boundary conditions – solution response to boundary and exceptional(negative) input;
- State Transitions – does the solution switch correctly from state to state;
- Proper exception handling, error and information message recovery;
- Data validation

The following list presents the features that were tested. The SAAM Platform consists of Web application running on Desktop and Mobile browsers, and Mobile app.:

1. SAAM Platform – User registration
2. SAAM Platform User access, Tokenization - Login, SSO, Access
3. SAAM Platform Profile Management - Manage User and Device settings
4. SAAM Platform Coaching
    a. Sleep – Overview, Coaching, Diary
    b. Cognition – Overview, Mood slider, Coaching
    c. Social – Overview, Coaching
    d. Activity – Overview, Coaching, Electricity consumption
    e. Mobility – Overview, Coaching
5. SAAM Platform Events – Events Setup and Calendar
6. SAAM Platform Calls – VoIP and Messaging
7. SAAM Platform Circles – Manage Social Contacts
8. SAAM Platform Wall – Notifications for Coaching suggestions, Requests, Calls & Messages, Diary Entries and Events
9. SAAM Platform – Care giver and Coach role assignment and activities
10. SAAM Platform Administration panel – Register, Login, Access
11. SAAM Platform Administration panel – User management
12. Sens belt, Sens bed and Sens bracelet sensors devices – Connectivity, Data Transfer and Management

2.2.3.2 Approach to testing

Stories describing single functionality are assigned to the relevant ScaleFocus Quality Engineer for creation of test cases and automation.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 769661

In order to have valid results from test executions ScaleFocus Quality assurance team fabricates test data suitable for the developed test scenarios.
Test verification scenarios are used for functional testing. The test scenarios executions are performed as follows:

- Component testing – single functionality of a developed feature is tested stand-alone against the requirements described in the relevant user story. A single developed feature is tested in the QE environment to validate their correct behavior as soon as the functionality has passed Unit tests in Dev environment and is deployed.
- End- To- End (E2E) testing – where it is applicable. This software testing validates entire software from starting to the end.E2E scenarios are executed to guarantee the correct process flow functioning.

Scope of the E2E testing:

- Verification of Microservices
- Verification of MSSQL and MongoDB
- Validation of RabbitMQ
- Verification of logging
- Verification of MicroHub connectivity

Depending on the result of the test the tasks that have to be implemented by developers are moved between different statuses described in Section 2.3.3.: User story testing.
When a bug is found it is logged with a title, description, severity level, found in version, logging files if available etc. After the bug is fixed the test scenario is applied again to ensure that the bug is no longer present. Regression sets of tests are executed to ensure there is no change in the expected system behavior after the fix is applied.
Risk-based testing approach is adopted. Selection, allocation and prioritization of test artifacts to maximize test effectiveness and efficiency is part of the risk-based strategy. Risk assessment is done via discussions held among all stakeholders.

2.2.3.3 Pass/Fail Criteria

Pass criteria: When all acceptance criteria from the tasks (users stories) are met. Test result equals Expected result.
Fail criteria: When acceptance criteria from the tasks (users stories) are not met. Test result does not equal Expected result .

2.2.4 Integration Testing

2.2.4.1 Objectives

SAAM platform is integrated with hardware devices, backend and external systems.

Integration from SAAM platform to other involved systems will be realized with interfaces allowing communication with web services.

This project has received funding from the European Union's Horizon 2020
research and innovation programme under grant agreement No 769661

Integration testing verifies that the interconnection between the SAAM platform, back end and external systems functions properly. This test proves that all areas of the application interface with other systems correctly and that there are no gaps in the data flow.

Integration testing is performed to ensure that separately developed components can work together and have a successfully integrated application as a result.

Integration with Third-party services:

- Weather Services
- Air-quality data
- Job Offers (job-care.bg) – POC (proof of concept)
- News feeds (RSS Subscription) – POC (proof of concept)

2.2.4.2 Approach to testing

Integration testing validates the developed interfaces and their proper communication with the relevant interfaces.

In cases when integration of some Components is not established with the SAAM platform or some features are not yet developed, mock-ups are used in the development and testing process. This approach guarantees correct operation of the interface when sending requests and receiving the expected responses at earlier stage of integration development process.

The approach during integration testing includes Web services call using specific tool (SoapUI) and validation of the returned response.

2.2.4.3 Pass/Fail Criteria

Pass criteria: For all tests - Test result equals Expected result.

Fail criteria: For some tests - Test result does not equal Expected result.

2.2.5 Unit Testing

2.2.5.1 Objectives

Unit testing is a level of software testing where individual units of source code, sets of one or more computer program modules together with associated control data, usage procedures, and operating procedures are tested. The purpose is to validate that each unit of the software performs as designed. A unit is the smallest testable part of software.

2.2.5.2 Approach

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 769661

Unit testing are performed as a part of the verification process of the software components in the source code which are created by development team members.

2.2.6 Regression Testing

2.2.6.1 Objectives

The goal is to ensure that no unexpected behaviour of the system's functionalities is introduced after deploying newer versions of the application. Automation of regression test cases will ensure that there are no broken functionalities in the system after bug fixing.

2.2.6.2 Approach to testing

After developing new functionality and changes are applied to the application or bug is fixed, regression suit or all the tests in the existing test bucket is executed.

2.2.6.3 Pass/Fail Criteria

Pass criteria: For all tests - Test result equals Expected result.

Fail criteria: For some tests - Test result does not equal Expected result.

2.2.7 Load and Performance Testing

2.2.7.1 Objectives

Performance testing is a practice which objective is to determine how a system performs in terms of responsiveness and stability under a particular workload. It can also serve to investigate, measure, validate or verify other quality attributes of the system, such as scalability, reliability and resource usage.

The general scope of the performance test subproject includes a number of areas:

- Business critical web operations – Web application
- Frequently used interfaces

Separate Performance test plan will be prepared before performance test phase.

The system must be able to support the anticipated numbers of users, concurrent sessions and transaction volumes. The following types of performance tests will be executed in order to guarantee that the required metrics are satisfied:

Load testing

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 769661

Load tests check the application's ability to perform under anticipated user loads in concurrent sessions. The objective is to identify performance bottlenecks before the software application goes live.

Stress testing

Stress testing involves testing the application under extreme workloads to see how it handles high traffic or data processing. The objective is to identify breaking point of an application.

2.2.7.2 Approach to testing

The performance testing tool which will be used for test execution, monitoring and investigating will be defined at a later stage. The tests will be executed over pre-production environment with conditions as close as possible to the real environment. The simulated load is based on predefined requirements.

2.2.7.3 Pass/Fail Criteria

Pass criteria: For all tests – The performance, reliability and sustainability of the system should meet the predefined requirements.

Fail criteria: For some tests - Test Result does not meet the predefined performance, reliability and sustainability of the system.

2.2.8 Acceptance Testing (Pilot Testing)

2.2.8.1 Objectives

Testing conducted to determine whether or not a system satisfies the acceptance criteria and to enable the customer, which in SAAM case are the Pilot Users, to determine whether or not to accept the system. Acceptance testing ensures that customer requirements' objectives are met and that all components are correctly included in a customer package.

2.2.8.2 Pass/Fail Criteria

Pass/Fail criteria will be defined by ScaleFocus Test Team.

2.2.8.3 Acceptance procedure

Acceptance procedure is defined by ScaleFocus Test Team.

2.2.8.4 Testing Tools

The following table presents a list of all project components, which will be tested, and corresponding test methods and tools, used for testing.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 769661

Item	Test method
Unit Testing	Automated
Functional Testing	Manual/Automated
Integration Testing	Manual/Automated
Regression Testing	Manual/Automated
Load and Performance Testing	Automated
Security Testing	Manual/Automated
Acceptance Testing	Manual

2.2.9 Acceptance Criteria

- All test scenarios are executed on minimum one iteration
- All planned tests for the solution are executed successfully, according to their Pass criteria as defined in their corresponding Pass/Fail criteria section
- All identified issues during the planned tests execution have been reported, addressed, and closed
- Solution deployment procedure executes successfully without failure
- All Critical, High and Medium issues are fixed – exceptions are issues with Very Low priority that are agreed for rejection from customer

2.3 Test Process

The development is realized using agile practices and KANBAN methodology. The test process iterations consists of:

- Analyzing user story requirements and acceptance criteria
- Testing design and implementation
- Testing execution
- Defect tracking
- Reporting
- Evaluating Exit Criteria
- Test activity closure

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 769661

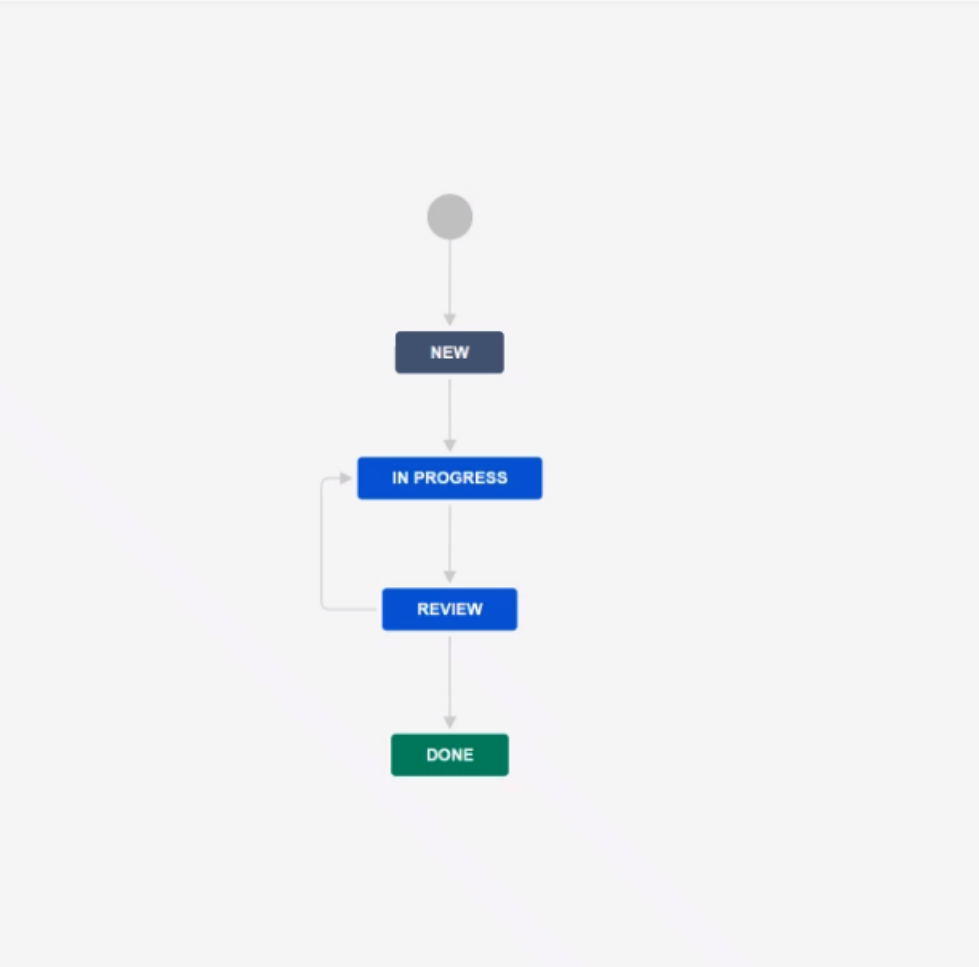
2.3.1 Environments

The test environments used in the project are:

- Development/QE environment:
- The dev team deploy all new functionalities and bug fixes together with the execution of the unit and integration tests on this environment.
- The QA team execute automation frontend and backend on this environment.

2.3.2 User Story Testing – Test Ticket workflow

The following diagram describes the flow of the Test Ticket processing:



- New – the item is newly added to the board, it is groomed and clear and QA can start working on it

This project has received funding from the European Union's Horizon 2020
research and innovation programme under grant agreement No 769661

- Active (In Progress) – the item is in progress by the QA (quality assurance) team. QA should create test cases, describe the steps for testing the item and provide additional testing information if needed. When ready the QA representative should change the status to Review.
- Review – after the QA work is done the item should be reviewed by PM/PO. If approved the status should be changed to Done. If additional QA work is required, the items should be moved to "In Progress"
- Done (Closed) – the item is approved and closed

2.3.3 Defect lifecycle

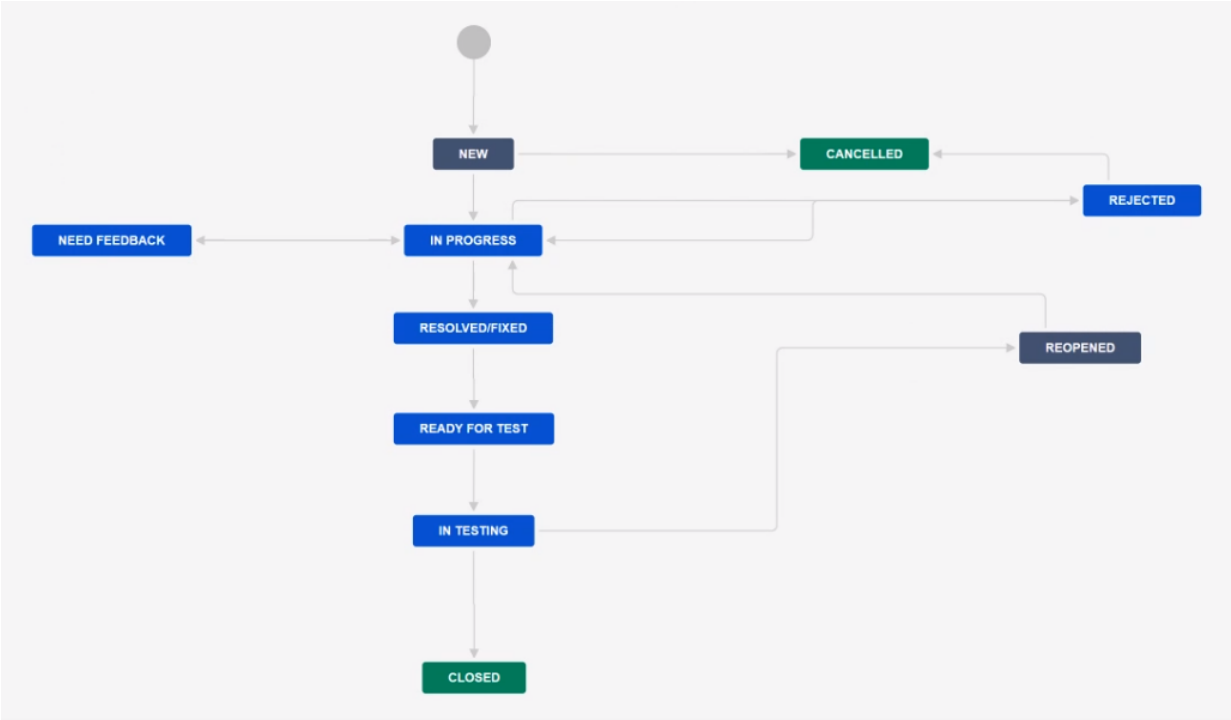
Issues, that come upon during the testing process, are listed as Issue/bug items in the Work Items section in the Project system "Jira". The procedure for issue reporting and fixing is as follows:

- Issue, which is identified during testing/review activities, is created in the Project System by QA (quality assurance) engineer. The issue is opened within dedicated project in Jira and its status is marked as "**New**". All "**New**" issues are assigned to responsible software developer.
- In case the issue is not reproducible or not existing the software developer changes the status to "**Cancelled**".
- The responsible developer changes the status to "**In Progress**".
- In case the developer needs this issue to be reviewed, the status is changed to "**Need Feedback**".
- In case it is determined that the issue is defect and should be resolved the status is changed to "**In Progress**", otherwise the status is changed to "**Rejected**" with Resolution:
  - "**Not a bug**" – in case the issue is not a bug
  - "**Duplicate**" – in case another defect is open regarding the same issue
  - "**Won't fix** "- in case this is a defect but it is decided that the impact of the functionality is not critical
  - "**CR**" – in case it is determined that the issue is a change request
- In cases when the issue is rejected with resolutions "***Not a Bug***", "***Duplicate***" and "***Won't fix***" QA engineer review it and if the issue matches the set resolution, the status is changed to "Cancelled" otherwise the QA returns it in status "**In Progress**".
- In case the issue is rejected with resolution "***CR***", the issue is transformed to type "Change request"
- The developer reviews the issue, collects all available information, reproduces the issue and applies required fixes. The status is changed to "**Resolved**" with Resolution "***Fixed***" and the status is changed to "**Ready for Test**".
- After the issue is fixed and fix is applied on Test environment the status is changed to "**In testing**" by the QA engineer. The issue is assigned to the responsible QA.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 769661

- If QA engineer re-tests fixed issue and reproduces the defect the status is changed to "**Reopened**"; the issue is assigned to the responsible developer. Then developer change the status of the defect to "**In Progress**" and follows the relevant workflow thereafter.
- QA engineer verifies that the bug is fixed, performs corresponding tests to check if issue is fixed correctly and closes the issue by changing its status to "**Closed**".

The following diagram represent the defect workflow management:



2.3.4 Resources and Planning

2.3.4.1 Technical Resources and Test Environment

The following items are available for Testing:

- Test environment with latest version
- Access to test environment Data base for verification purposes
- Access to the test web services
- Technical support

2.3.4.2 Responsibilities

This section lists base project team actors as well as their roles and obligations in testing process.

This project has received funding from the European Union's Horizon 2020
research and innovation programme under grant agreement No 769661

Team (Dev/QA) Lead – Role of this team member is to prioritize testing tasks and review the test deliverables. This person is also responsible for assigning issues to developers, verifying deployment procedure of the system on production environment.

Developer – Role of this team member is to the implement functionality describes in User stories. This person is also responsible for performing white box testing as well as debugging and writing/performing of unit tests, analyze issues and fix bugs.

Quality Assurance (QA) Engineer – Role of this team member is to prepare and perform test activities and create test deliverable documents. This person is also responsible for identification of issues and their creation in the project tracking system.

PM / Business Analyst – Role of this team member is to consult the team when additional information, clarification and confirmation are needed regarding business processes and requirements.

2.3.5 Entry Criteria

- Definition of done from Dev perspective;
- Code is complete and delivered QE environment;
- The story which will be tested is in Ready for test status;
- Test Cases are prepared and completed;
- Test environment is ready and setup correctly;
- Interfaces and connectivity readiness
- Test Data is available.

2.3.6 Exit Criteria

- There are no open bugs with the High or Medium severity in functionalities
- Integration and regression testing passed without the defects with High or Medium severity, or will not be treated with major breakdowns as specified.
- Test cases, including functionality, integration and regression, will be fully executed.
- Defects should be reviewed, discussed and treated accordingly prior to next build deployment and delivery.

2.4 Use Case scenarios

This section describes the user stories for SAAM- Platform that are implemented and tested. User stories contain a description of the SAAM- Platform features.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 769661

2.4.1 SAAM Platform – User registration – split into two user PU/SU

Title:	**User registration – PU Sign Up**	
Description:	***As a***	PU (Primary User)
	I want to	be able to create a SAAM Platform account
	so that	I can be able to Login, use the installed in my home SAAM and have subscription to at least one coaching domain.

Title:	**User registration – SU Sign Up**	
Description:	***As a***	SU (Secondary User)
	I want to	be able to create a SAAM Platform account
	so that	I can be able to Login into to the SAAM Platform and use it as related to a PU or as representative of organization providing services to PU.

2.4.2 SAAM Platform User access, Tokenization – Login, Access

Title:	**Secure Login**	
Description:	***As a***	PU/SU
	I want to	be able to log in to the system via a secure login process
	so that	I can be sure that no one can get hold of my user credentials – neither by getting information from the computer and browser I use for accessing the SAAM Platform nor by capturing information transmitted from my computer to the SAAM Platform server during the login process.

Title:	**User Authorization**	
Description:	***As a***	PU/SU
	I want	only people authorized by SAAM to have access via the SAAM Platform to information related to my company
	so that	I can be sure that my sensitive information is kept secure and it is not possible for unauthorized people to access it.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 769661

Title:	User Access – PU Home page
Description:	***As a*** PU ***I want to*** have access to the sections Coaching, Events, Calls, Circles, Wall ***so that*** I can be able to open the section on my home page.

Title:	User Access – SU Home page
Description:	***As a*** SU ***I want to*** have access to the sections Events, Calls, Circles, Wall ***so that*** I can be able to open the section on my home page.

Title:	User Access – As PU provide access to SU
Description:	***As a*** PU ***I want to*** have ability to provide read and write access rights to SU ***so that*** SU can view or edit my account and data.

Title:	Logout
Description:	***As a*** PU/SU ***I want to*** be able to Sign out of the SAAM Platform ***so that*** other users cannot have access to my profile and other users can login with different account on the same device.

2.4.3 SAAM Platform Profile Management - Manage User and Device settings

Title:	Profile Management – Account information
Description:	***As a*** PU/SU ***I want to*** be able to manage my user account on the Profile page ***so that*** I can change my personal Account information, Notifications, Language settings and change my status.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 769661

Title:	**Profile Management – PU additional settings**
Description:	***As a*** PU ***I want to*** be able to manage my user account on the Profile page ***so that*** I can change my Coaching and Device settings.

Title:	**Profile Management – Reset Password**
Description:	***As a*** PU/SU ***I want to*** be able to reset my password ***so that*** I can change my password if necessary and login with my changed password.

Title:	**Profile Management – Change Email**
Description:	***As a*** PU/SU ***I want to*** be able to change my email ***so that*** I can receive email communication on the changed email address.

Title:	**Profile Management – Change Language**
Description:	***As a*** PU/SU ***I want to*** be able to change the Language ***so that*** I can use the SAAM Platform in my preferred language.

Title:	**Profile Management – Change Away status**
Description:	***As a*** PU/SU ***I want to*** be able to set my Away status to Yes or No ***so that*** I can activate and deactivate the analysis of coaching data when I am available or respectively unavailable to provide data.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 769661

2.4.4 SAAM Platform Coaching

2.4.4.1 Sleep – Overview, Coaching, Diary

Title:	**Social Pipeline - Sleep**
Description:	***As a*** SAAM PU ***I want to*** be able to open the Sleep pipeline from the Coaching section ***so that*** I can be able to track and manage my sleeping activities.

Title:	**Sleep Overview**
Description:	***As a*** SAAM PU ***I want to*** be able to open the Sleep Overview page of the Sleep pipeline ***so that*** I can be able to track my sleeping activity on the Personal Statistics list and have the option for detailed view stats.

Title:	**Sleep Coaching**
Description:	***As a*** SAAM PU ***I want to*** be able to open the Sleep Coaching page of the Sleep pipeline ***so that*** I can be able to check and follow suggestions for coaching actions, see the date of their creation and indicate my intention and capabilities and willingness to follow the coaching advices by opting out by one of the three buttons cannot, decline, accept.

Title:	**Sleep Diary – Record in the Morning and Evening**
Description:	***As a*** SAAM PU ***I want to*** be able to open and fill-in the Sleep Diary page of the Sleep pipeline ***so that*** I can be able to keep a diary of my sleeping activity in mornings and in the evenings.

2.4.4.2 Cognition – Overview, Moodslider, Coaching

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 769661

Title:	**Cognition Pipeline**
Description:	***As a*** SAAM PU ***I want to*** be able to open the Cognition pipeline from the Coaching section ***so that*** I can be able to track my cognition.

Title:	**Cognition Overview**
Description:	***As a*** SAAM PU ***I want to*** be able to open the Cognition Overview page of the Cognition pipeline ***so that*** I can be able to track my cognition on the Moodbox Dashboard and have the option for detailed view stats, and to change the Preferences.

Title:	**Cognition Coaching**
Description:	***As a*** SAAM PU ***I want to*** be able to open the Cognition Coaching page of the Cognition pipeline ***so that*** I can be able to check and follow suggestions for coaching actions, see the date of their creation and indicate my intention and capabilities and willingness to follow the coaching advices by opting out by one of the three buttons cannot, decline, accept.

Title:	**Cognition Moodslider**
Description:	***As a*** SAAM PU ***I want to*** be able to open the Cognition Moodslider page of the Cognition pipeline and indicate my mood ***so that*** I can be able to keep a record of my daily mood.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 769661

2.4.4.3 Social – Overview, Coaching

Title:	**Social Pipeline**
Description:	***As a*** SAAM PU ***I want to*** be able to open the Social pipeline from the Coaching section ***so that*** I can be able to add and manage my social goals.

Title:	**Social Overview**
Description:	***As a*** SAAM PU ***I want to*** be able to open the Social Overview page of the Social pipeline ***so that*** I can be able to add goals for social contacts, track my personal goals, have the option for detailed view stats and to change the Preferences.

Title:	**Social Coaching**
Description:	***As a*** SAAM PU ***I want to*** be able to open the Social Coaching page of the Social pipeline ***so that*** I can be able to check and follow suggestions for coaching actions, see the date of their creation and indicate my intention and capabilities and willingness to follow the coaching advices by opting out by one of the three buttons cannot, decline, accept.

2.4.4.4 Activity – Overview, Coaching, Electricity consumption

Title:	**Activity Pipeline**
Description:	***As a*** SAAM PU ***I want to*** be able to open the Activity pipeline from the Coaching section ***so that*** I can be able to add and manage my social goals.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 769661

Title:	**Activity Overview**
Description:	***As a*** SAAM PU ***I want to*** be able to open the Activity Overview page of the Activity pipeline ***so that*** I can be able to track my Activity statistics on the Personal Statistics list and have the option for detailed view stats.

Title:	**Activity Coaching**
Description:	***As a*** SAAM PU ***I want to*** be able to open the Activity Coaching page of the Activity pipeline ***so that*** I can be able to check and follow suggestions for coaching actions, see the date of their creation and indicate my intention and capabilities and willingness to follow the coaching advices by opting out by one of the three buttons cannot, decline, accept.

Title:	**Activity Electricity consumption**
Description:	***As a*** SAAM PU ***I want to*** be able to open the Activity Electricity consumption page of the Activity pipeline ***so that*** I can be able to check my household electricity consumption for a given period and compared to previous one with equal parameters.

2.4.4.5 Mobility – Overview, Coaching

Title:	**Mobility Pipeline**
Description:	***As a*** SAAM PU ***I want to*** be able to open the Mobility pipeline from the Coaching section ***so that*** I can be able to add and manage my social goals.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 769661

Title:	**Mobility Overview**
Description:	***As a*** SAAM PU ***I want to*** be able to open the Mobility Overview page of the Mobility pipeline ***so that*** I can be able to track my mobility activity on the Personal Statistics list and have the option for detailed view stats.

Title:	**Mobility Coaching**
Description:	***As a*** SAAM PU ***I want to*** be able to open the Mobility Coaching page of the Mobility pipeline ***so that*** I can be able to check and follow suggestions for coaching actions, see the date of their creation and indicate my intention and capabilities and willingness to follow the coaching advices by opting out by one of the three buttons cannot, decline, accept.

2.4.5 SAAM Platform Events – Events Setup and Calendar

Title:	**Events**
Description:	***As a*** PU/SU ***I want to*** be able to open the Events page ***so that*** I can be able to add new events, manage my events and check the events calendar.

Title:	**Events Details**
Description:	***As a*** PU/SU ***I want to*** be able to open the Events Details page ***so that*** I can be able to view the Events Overview and manage the Event detail settings.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 769661

Title:	**Enable/Disable Event Notifications**
Description:	***As a*** PU/SU ***I want to*** be able to manage the receiving of Notifications for Event on Event details page ***so that*** I can receive Notification when they are switched on and respectively not receive Notification when they are switched off.

Title:	**Edit Event**
Description:	***As a*** PU/SU ***I want to*** be able to edit an Event to which I am the owner ***so that*** I can update the Event properties.

Title:	**Event Tags**
Description:	***As a*** PU/SU ***I want to*** be able to add Tags to Events ***so that*** I can choose the topic of the Event.

Title:	**Event Types**
Description:	***As a*** PU/SU ***I want to*** be able to select the Event type ***so that*** I can create Reminder, Meeting, Call and Appointment events.

Title:	**Event Recurrence**
Description:	***As a*** PU/SU ***I want to*** be able to set the Recurrence of Events ***so that*** I can schedule daily, weekly, monthly, and not repeating events.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 769661

Title:	Event Participants	
Description:	*As a*	PU/SU
	I want to	be able to manage the Event Participant on the Event details page
	so that	I can add participants through search option, check list with invited people and cancel the invites, check list with persons going to the event and cancel their participation.

Title:	Event Requests	
Description:	*As a*	PU/SU
	I want to	be able to manage the Event Requests on the Event details page
	so that	I can accept or decline Requests related to an Event of which I am the owner.

Title:	Event Location	
Description:	*As a*	PU/SU
	I want to	be able to choose a Location during the Event creation
	so that	I can connect a Location to an Event through the Location finder and that Location will be presented on the Event details page.

Title:	Event – Weather Information	
Description:	*As a*	PU/SU
	I want to	be able to select a date for the event
	so that	I can see the Weather forecast information for the chosen date and location if the event occurs in the next 3 days.

2.4.6 SAAM Platform Calls – VoIP and Messaging

Title:	Calls	
Description:	*As a*	PU/SU
	I want to	be able to open the Calls page
	so that	I can be able to initiate voice and video calls, and message conversations with my connections.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 769661

Title:	Calls - Connections
Description:	***As a*** PU/SU ***I want to*** be able to open the list with Connections on the Calls page ***so that*** I can be able to select a connection from the list and start video, voice, message chat conversation and manage the call preferences for that person.

Title:	Calls - History
Description:	***As a*** PU/SU ***I want to*** be able to open the History tab on the Calls page ***so that*** I can be able to have a list with connections to track the history of my calls and messages with that connections. From the History tab I can be able to select a connection from the list and start video, voice, message chat conversation and manage the call preferences for that person.

2.4.7 SAAM Platform Circles – Manage Social Contacts

Title:	Circles
Description:	***As a*** PU/SU ***I want to*** be able to open the Circles page ***so that*** I can be able to manage my contact, search and add people to the contact list, and check the status of sent and received invites.

Title:	Circles – Contacts Overview
Description:	***As a*** PU/SU ***I want to*** be able to open the Contacts tab on the Circles page ***so that*** I can be able to check my contacts list, remove a contact, initiate video, voice and message chat conversation.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 769661

Title:	Circles – Contacts Tag management
Description:	***As a*** PU/SU ***I want to*** be able to manage the Tags of my contacts ***so that*** I can be able to add and remove the tags of my contacts and edit already added tags.

Title:	Circles - Requests
Description:	***As a*** PU/SU ***I want to*** be able to open the Requests tab on the Circles page ***so that*** I can be able to check and manage sent and received invites.

Title:	Circles – Circle Explorer
Description:	***As a*** PU/SU ***I want to*** be able to open the Circle Explorer tab on the Circles page ***so that*** I can be able to search and for people and add them to my circles.

2.4.8 SAAM Platform Wall – Notifications for Coaching suggestions, Requests, Calls & Messages, Diary Entries and Events

Title:	Notifications popup for new Events
Description:	***As a*** PU/SU ***I want to*** be able to receive Notifications popup for new Events ***so that*** I can be open the popup and navigate to the Notification item like reminders for pill intake, incoming calls, etc.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 769661

Title:	**Wall**
Description:	***As a*** PU/SU ***I want to*** be able to open the Wall page ***so that*** I can be able to view and manage Notifications.

Title:	**Wall – Coaching Notifications**
Description:	***As a*** PU ***I want to*** be able to open the Coaching section on the Wall page ***so that*** I can be able to view list of coaching suggestions divided by domains and open coaching suggestions through the View button.

Title:	**Wall – Notifications for Diary Entries**
Description:	***As a*** PU ***I want to*** be able to open the Diary Entries section on the Wall page ***so that*** I can be able to view notifications for pending or missing diary.

Title:	**Wall – PU's Coaching Notifications**
Description:	***As a*** SU ***I want to*** be able to open the Wall tab sections PU's Coaching ***so that*** I can be able to view coaching suggestions provided to PU's divided by domains and their status.

Title:	**Wall – Notifications for Requests**
Description:	***As a*** PU/SU ***I want to*** be able to open the Requests section on the Wall page ***so that*** I can be able to view and Accept or Reject invitation requests the same way as in the Circle menu.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 769661

Title:	Wall – Notifications for Calls & Messages
Description:	***As a*** PU/SU ***I want to*** be able to open the Calls & Messages section on the Wall page ***so that*** I can be able to view a list with notifications for messages, voice and audio calls.

Title:	Wall – Notifications for Events
Description:	***As a*** PU/SU ***I want to*** be able to open the Events section on the Wall page ***so that*** I can be able to view a list of notifications for events to which I am invited.

2.4.9 SAAM Platform – Caregiver and Coach role assignment and activities

Title:	Assign roles of coach/caregiver
Description:	***As a*** PU ***I want to*** be able to able to assign and revoke coach/caregiver roles to SU through the option to add tag on the contacts ***so that*** the SU use can facilitate my activities.

Title:	Accept tag assignment
Description:	***As a*** SU ***I want to*** be able to able to accept and later on manage tag assignment as Caregiver, Coach, Emergency Contact ***so that*** I can facilitate PU's activities.

Title:	Decline tag assignment
Description:	***As a*** SU ***I want to*** be able to able to decline tag assignment ***so that*** I can reject the activity facilitation of PU.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 769661

Title:	**My PUs administration panel**	
Description:	***As a***	Caregiver/Coach
	I want to	be able to open the My PUs administration panel
	so that	I can be able to login on behalf of PU with restricted access rights. I will have read only rights if the PU is using the system. And if the user is not using the system due to unavailability, I will have access rights to edit the PU data.

Title:	**My PUs Coaching**	
Description:	***As a***	Coach
	I want to	be able to open the PU's Coaching tab on the Wall page
	so that	I can be able to manage PU's coaching advices.

2.4.10 SAAM Platform Administration panel – Login, Access

Title:	**User Administrator Access** - MA	
Description:	***As a***	Master Administrator (MA)
	I want to	be able to connect to the Database
	so that	I can have Administrator full access rights.

Title:	**User Login –** SA/PS	
Description:	***As a***	System Administrator/Platform Support user (SA/PS)
	I want to	be able to login to the SAAM Platform
	so that	I can access the Administration panel

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 769661

Title:	**User Administrator Access** – SA/PS
Description:	***As a*** SA/PS ***I want to*** be able to have Administrator access rights to the SAAM Platform ***so that*** I can go through the authorization process and access resources based on the Administrator role.

2.4.11 SAAM Platform Administration panel – User management

Title:	**Master Administrator - user management**
Description:	***As a*** MA ***I want to*** have Administrator full rights ***so that*** I can be able to create and manage SA, PS, PU, SU accounts and track the status of the user devices.

Title:	**System Administrator - user management**
Description:	***As a*** SA ***I want to*** have System Administrator rights ***so that*** I can be able to manage PU and SU accounts and manage their access to specific resources.

Title:	**Platform Support - user management**
Description:	***As a*** PS ***I want to*** have Platform Support user rights ***so that*** I can be able only to view PU and SU accounts and register issues.

Title:	**Administrator panel - Search and Filter**
Description:	***As a*** SA/PS ***I want to*** search and filter for users based on email or name ***so that*** I can find users that I need to.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 769661

2.4.12 Sens belt, Sens bed and Sens bracelet sensor devices

2.4.12.1 Sens belt

Title:	**Sens belt**	
Description:	***As a***	SAAM Platform user
	I want to	be able to attach hardware sensor device powered by battery on my belt
	so that	movement data can be collected and transferred to my mobile device

2.4.12.2 Sens bed

Title:	**Sens bed**	
Description:	***As a***	SAAM Platform user
	I want to	be able to put hardware sensor device connected to the electricity under the bed mattress
	so that	sleeping data can be collected and transferred to my mobile device

2.4.12.3 Sens bracelet

Title:	**Sens bracelet**	
Description:	***As a***	SAAM Platform user
	I want to	be able to carry bracelet sensor device on my hand
	so that	movement and body status data can be collected and transferred to my mobile device

2.4.12.4 Sens devices – Connectivity

Title:	**Connectivity – Prompt to switch on mobile device's Bluetooth**	
Description:	***As a***	SAAM Platform user
	I want to	be prompted after Login to switch on the mobile device's Bluetooth if this is not enabled
	so that	the mobile application can trigger the mobile device to search for activated sensor devices

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 769661

Title:	**Connectivity – Switch on sensor devices**
Description:	***As a*** SAAM Platform user ***I want to*** be able to switch on the sensor devices ***so that*** the sensor devices can be discoverable by the mobile application and connection can be es

Title:	**Connectivity – Search automatically for active devices**
Description:	***As a*** SAAM Platform user ***I want*** the used mobile device to search for activated Sens belt, Sens bed and Sens bracelet devices after every login/start of the mobile application ***so that*** the mobile device can discover active sensor devices

Title:	**Connectivity – Establish connection to sensor devices**
Description:	***As a*** SAAM Platform user ***I want*** the mobile application to be able to establish connected to discovered active sensor devices when the mobile device's Bluetooth is switched on ***so that*** the mobile device can connect to the sensor devices and data to be collected for analysis

Title:	**Connectivity – Information when no devices are found**
Description:	***As a*** SAAM Platform user ***I want to*** be notified under the device settings section when no devices are found ***so that*** I can be informed that the connectivity between the mobile device and sensor devices cannot be established

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 769661

Title:	**Connectivity – Bluetooth connection in range of 10m**
Description:	***As a*** SAAM Platform user ***I want*** the mobile application to maintain stable connectivity of the mobile device to the sensor devices within the range of 10 m ***so that*** the data transfer is not interrupted

2.4.12.5 Sens devices – Data transfer

Title:	**Data transfer – Mobile device Idle mode**
Description:	***As a*** SAAM Platform user ***I want*** data from the sensor devices to be cashed locally on the mobile device when the mobile device is disconnected from internet ***so that*** the data is secured and stored

Title:	**Data transfer – Synchronize data after re-establishing of internet connection**
Description:	***As a*** SAAM Platform user ***I want*** the collected on the local storage data when the Internet connection was interrupted to be synchronized after re-establishing the internet connection ***so that*** the data is transferred successfully for analysis

2.4.12.1 Sens devices – Management

Title:	**Management – List with added sensor devices**
Description:	***As a*** SAAM Platform user ***I want to*** have a list with all added sensor devices under the device settings section ***so that*** I can manage the connection to the sensor devices through the mobile application

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 769661

Title:	Management – Add additional sensor device
Description:	***As a*** SAAM Platform user ***I want to*** be able to add an additional sensor device under the device settings section of the mobile application ***so that*** I can connect additional device when such is available

Title:	Management – Remove sensor device
Description:	***As a*** SAAM Platform user ***I want to*** be able to remove sensor device under the device settings section of the mobile application ***so that*** unused device is not appearing under the list with sensor devices

Title:	Management – Manage sensor device
Description:	***As a*** SAAM Platform user ***I want to*** be able to mage already added sensor device under the device settings section of the mobile application ***so that*** I can change/track the device connectivity

3. USER-SIDE DEVICES DEVELOPMENT, PRODUCTION AND ASSEMBLY TESTING

User-side devices have to be tested during the production, assembly, development and deployment of the software that runs on them. Before the integration of the devices into the SAAM system they have to undergo functional tests that confirm their correct operation. Testing procedures for each individual user-side device are explained in details in the following subsections.

3.1 eGW Device

Before deploying the eGW devices they have to go through a testing procedure. The production of an eGW device is separated into three individual steps. The first step includes PCB assembly procedures, the second step concerns the assembly of the produced PCBs inside the plastic enclosures finalized by a start-up test, while the third step contains procedures of flashing eGW device and final testing of completely assembled eGW system.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 769661

3.1.1 eGW Device Production Testing

eGW devices were tested for different defects during the production process. In the first step the electrical testing of all connections on the produced PCB were performed by the automatic testing machine which checks all the connections for short circuits and identifies missing connections. Next, all electronic components are placed and soldered on the produced PCBs automatically by robots or manually (non-standard components) and visually inspected for defects. The visual inspection of manually assembled components is done by the person assembling the components. PCBs with detected flaws are returned for the inspection and repaired if possible.

3.1.2 eGW Device Assembly and Start-up Test

eGW device PCB is mounted in a plastic enclosure with the minimum number of openings in order to limit the risk of the indoor environmental effects or user's actions which could endanger the user himself or the integrity of the device. The plastic enclosure has two openings made by CNC laser cutting machine *(i)* for power cord connector and *(ii)* for connecting the bed sensors via USB socket.

PCB is mounted inside the plastic enclosure using screws. Power connector is placed into the power-cord hole and connected to the board with the wires inside the enclosure. Two PCB antennas connected to the Wi-Fi/Bluetooth RF connectors on the eGW PCB are mounted internally at the opposite sides of the enclosure using adhesive tape.

After inserting bootable SD card with complete SAAM eGW system into the SD card slot on the eGW PCB, the device is connected to the power supply. If there is no status LED activity seen on the eGW PCB, the device is put aside for a fault inspection. If the status LEDs shows the device activity, the successful automatic device registration on SAAM WiFi network has to be checked trough the web dashboard of SAAM Wi-Fi access point (AP). If the device fails to automatically register, it is assumed faulty and put aside for fault investigation.

In the case of successful start-up test, device is powered down and an UWB device is mounted on an eGW PCB UWB connector.

3.1.3 eGW Device Final Test

The procedure starts by flashing the device with the adequate software using a master flashing SD card. The flashing process is indicated by the four LED lights on the eGW that blink consecutively. If the lights abruptly turn off during the flashing process, the flashing process failed, and it needs to be repeated. If the flashing process was successful, all four LEDs will light up and then turn off. After the flashing is done the device will turn off. At this step an empty SD card has to be inserted in the device since eGW formats the SD card after fresh boot and mount it as a storage device. In the last step the device is tested for the internet connection.

Since it is required to keep track of the device and have it accessible and distinguishable from other devices, the ID used for management purpose has to be extracted and saved in the device table linked to the depersonalized location (e.g. BG01).

This project has received funding from the European Union's Horizon 2020
research and innovation programme under grant agreement No 769661

3.2 PMC Device

During the production and deployment phases of the Power Meter and Control unit (PMC) several stages of the hardware, calibration and programming tests are performed. The device consists of two modules; *(i)* measurement module and *(ii)* processing/communication module, which have to be verified by appropriate testing and programming procedures. Processing/communication module is based on ARM-based Octavo system in package (SiP) running Linux Debian 9.

3.2.1 Production Testing and Calibration

In the first step the measurement module printed circuit board (PCB) is installed in the bottom part of the enclosure. In the next step the firmware with testing procedure is loaded to the microcontroller using the programming port. The installed firmware checks the functions and reports the results via programming port.

A one-point calibration of the current and voltage sensing inputs is performed by connecting the module to a calibrated voltage and current sources.

In the next step the processing/communication module is installed on the top of the measurement module and an electrical test on the processing/communication module power busses is performed.

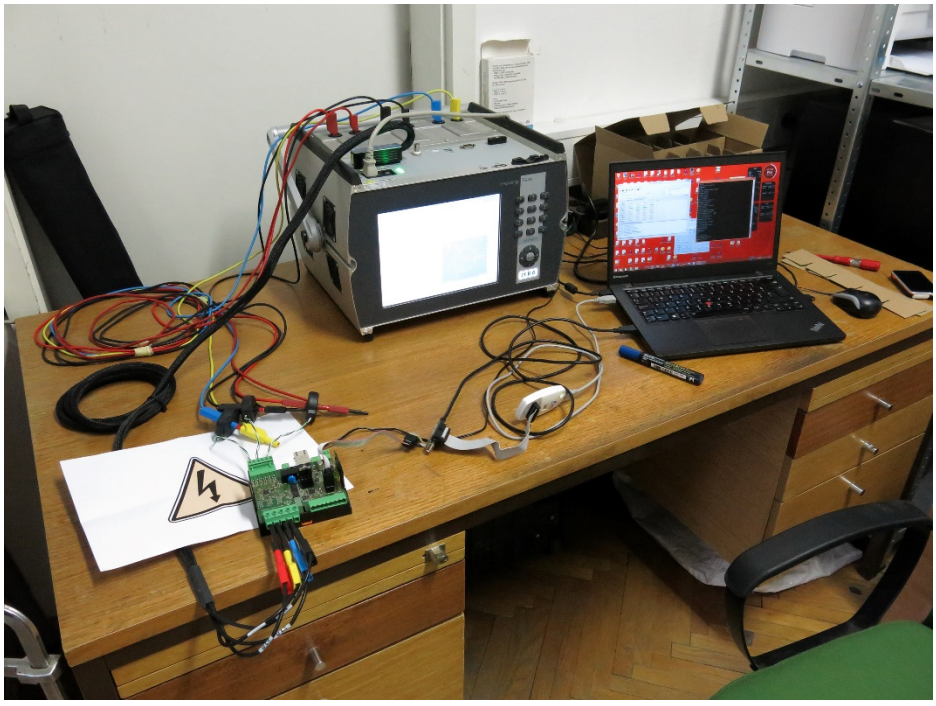


Figure 1: PMC Measurement module calibration

Tested PMCs are now installed in the SAAM enclosure box and prepared for processing/communication module programing and testing.

The procedure for flashing and testing of the processing/communication module is similar to the procedure for the eGW. The PMC is flashed using a master flashing SD card. During the flashing process

This project has received funding from the European Union's Horizon 2020
research and innovation programme under grant agreement No 769661

the LED light on the PMC is blinking. After the flashing procedure is complete only one solid green LED should be on.

The success of flashing process and the device internet connectivity can be checked in two ways. In the first approach we connect to the PMC via a serial port. A correct start-up of the device indicates successful flashing. The alternative is to access the PMC through the eGW by scanning all devices that are connected to the same network. If the PMC is not found in the list of scanned devices, this indicates that it does not have an active internet connection.

In the last step the ID of the PMC is saved in the device database in such a manner that it lines up with the eGW for the correct depersonalized location.

3.2.2 NILM Algorithm and Testing

The NILM (Non-intrusive Load Monitoring) algorithm is based on detecting changes (events) in real and reactive power (*dP-dQ*) on the grid and creating *dP-dQ* events used by different NILM algorithms for detecting various devices. The events are obtained by the analysis of the grid parameters such as voltage, active and reactive power, and energy for all phases in the time domain with the resolution of one second. The data is normalized to the nominal voltage to filter out the dependence of constantly changing grid characteristics on the appliance. Additionally, the spikes in power (e.g., start of the motor) are filtered out to obtain and operate only with the "stable" states with filtered fast transients. From these filtered data, the algorithm can detect simple momentary and long continuous events. In addition, the algorithm tries to distinguish the concurrent events (e.g., momentary event inside ongoing event), which is crucial in the process of identifying or labelling the appliances. The more detailed description of how the algorithm works and description of all its features is available in the deliverable D3.1.

The testing was done in a controlled environment, where one PMC device was installed in the private household with all initial services included. In the first step, an additional PMC device was connected directly to the appliance to obtain the complete picture of the energy consumption of the appliance without any background noise. From these measurements, quite clean response of events related to particular appliance was obtained. These events were then used as the starting point for building the appliance database. When the initial database was established the deployed PMC at the household level was extensively tested for bugs and improvements on the stability. In the second step, the improved services and algorithms were deployed to the PMC. These PMCs were deployed on different SAAM pilot sites where individual installation-time labelling per site was performed. Subsequently, a dedicated database was built for each site from installation-time labelling and remotely deployed to a corresponding site. At this level, the complete solution was additionally tested if all services were working as planned. After the deployment and extensive testing of the algorithm, some minor time related (different time zones) bugs were found and remotely patched.

3.3 UWB Device

During the development and production of the UWB devices, several phases of testing have to be performed. First phase comprises testing and validation of communication protocol during the

This project has received funding from the European Union's Horizon 2020
research and innovation programme under grant agreement No 769661

firmware development. Visual inspection and basic electric testing of the produced device is performed in the second phase, while during the third phase the device is flashed by firmware and the basic device functionality is checked. In the last phase the functional testing of the communication protocol according to the role of the UWB device in the local UWB network is carried out.

3.3.1 UWB Communication Protocol Development Testing

Custom UWB communication protocol enables exchanging information between the UWB devices, which is vital for sensing the changes in living environment. This is the key enabler for person's location and movement detection in indoor environment.

Developing the communication protocol is a tedious and complex process because of the distributed nature of the protocol's operations. Typical embedded firmware development covers only the operations inside a single device, requiring only unit testing inside the embedded device. Communication protocols include at least two devices with different functional roles and thus work as a distributed software.

The development and testing is an iterative process with the feedback loop as shown in **Error! Reference source not found.**. SAAM UWB network c onsists of one MASTER device and two SLAVE devices. Whenever a new feature is added to the protocol or an issue is fixed, the communication protocol undergoes the entire testing procedure. All three devices are flashed with the custom developed firmware and then the network operations are tested for 10 minutes. If the communication fails, the bug must be identified, firmware fixed accordingly and the development loop repeated.

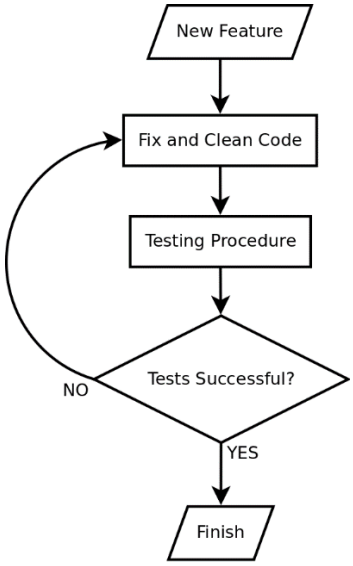


Figure 2: Communication protocol development and testing flowchart.

3.3.2 UWB Device Production Testing

Since the production process of electronic devices is not error-free, all the produced UWB PCBs are tested for production defects where a testing machine verifies all connections and possible short circuit situation.

3.3.3 UWB Device Flashing and Basic Functionality Testing

During the next step the devices are tested for operation defects. Errors during component placement and soldering can alter the device's functionality or completely prevent the operation of the device. Thus, each UWB device is being flashed with the appropriate binary flash file according to the device's communication role (MASTER or SLAVE).

There are two points in the process where the device can be detected as faulty:

This project has received funding from the European Union's Horizon 2020
research and innovation programme under grant agreement No 769661

- Fault in flashing procedure.
- Device operational fault.

If an UWB device is faulty, the flashing process will not finish successfully or it will not start at all. In some cases (e.g. in a case of a faulty UWB radio etc.), the flashing procedure will finish successfully but the device will run erratically due to less critical hardware errors. After successfully finished flashing process and device restart, LED indicator (*Figure 3*) should blink with a constant frequency. If the indicator is not blinking, a flashing process has to be repeated. In the case of unsuccessful second flashing procedure, the device has to be checked for defects.

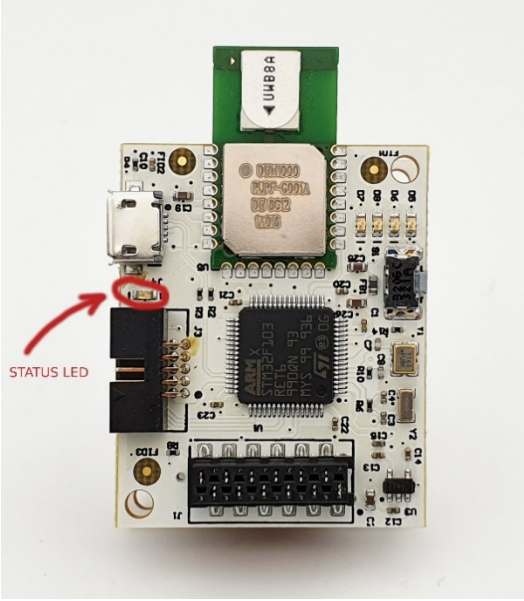


Figure 3: Status LED location on UWB device.

3.3.4 UWB Functionality Testing

The last step of UWB device testing at the production stage is a functionality test. Device under test is powered on and tested for complete functionality.

In the case of testing SLAVE UWB device, a fully functional MASTER UWB device is connected to the testing computer using UART interface with open serial terminal application. If the SLAVE device under test works correctly, the MASTER device prints the network activity of the new SLAVE device. If there are problems with the device under test, the SLAVE device can't establish the connection and is not able to maintain the communication directed by the MASTER device. Faulty SLAVE devices are put aside for defect investigation.

When testing the MASTER UWB device, no other MASTER UWB devices should be powered up at the same location. MASTER UWB device under test is connected to the computer using UART interface with serial terminal application which waits for the activity on UART interface. A reference SLAVE UWB device is powered-up and if MASTER UWB device operates as expected, it should report

This project has received funding from the European Union's Horizon 2020
research and innovation programme under grant agreement No 769661

communication activity of two UWB devices and the number of active devices - two for SAAM use case. The communication activity includes distance measurement reports and channel impulse response report for the link between a MASTER UWB device under test and reference SLAVE UWB device. Faulty MASTER UWB devices are put aside for defect investigation.

3.4 Ambient Sensor Device

The ambient sensor device is assembled from the Raspberry Pi computer, the Matrix Creator daughter board, a micro SD card, an USB speaker, and a wooden casing. All (electronic) components (devices) are purchased off-the-shelf and are initially tested by the manufacturer.

The ambient sensor device is assembled by connecting the Matrix Creator board to the Raspberry Pi extension header, connecting the USB speaker to the USB connector and auxiliary audio connector of the Raspberry Pi. Then a pre-programmed micro SD card is inserted in the Raspberry Pi and all electronic components are placed in a wooden enclosure. Special attention should be given to the alignment of the Matrix Creator connector with the Raspberry Pi header. In the case of wrong alignment, the Matrix Creator board does not function properly.

Each of the micro SD cards is programmed with a customized system software image which includes the operating system, the ambient sensor software, and the custom system parameters (e.g. depersonalized location). The pre-programmed micro SD cards are verified on a testing system with:

- Internet connectivity test – the connectivity of the ambient sensor is tested using both Raspberry Pi Ethernet and Wi-Fi connection.
- Functional test of the Matrix Creator board – basic Matrix Creator functional test is conducted to detect misaligned or defective Matrix Creator board.
- Functional test of the ambient sensor software – Ambient sensor software is built from several independent modules: data acquisition modules, voice command module, and rendering module. The test checks the operation of each module separately, checks the connection of the modules to the cloud, and verifies the gathered data in the cloud database.

3.5 MicroHub Device

3.5.1 MicroHub Hardware Testing

The MicroHub device production testing is based on several levels of verification steps:

- Automated optical inspection (AOI), where the assembled PCBs is verified with optical inspection to justify that the current PCB is assembled correctly.
- Self-Test at power up – the MicroHubs have implemented a self-test procedure at power up. After the boot process they go into a setup mode to test all connected devices such as motion sensor, memory and communication interfaces. Once that test has passed, the green LED is turned on and red LED is turned off; at failed self-test procedure the red LED remains on and the BLE communication is blocked.

This project has received funding from the European Union's Horizon 2020
research and innovation programme under grant agreement No 769661

- Production test with an App – this test is already a system integration test with a small App. The tablet/smartphone connects over BLE with the MicroHub device. That will ensure that BLE functionality is working properly and indicate that the self-test has passed. The App forces the MicroHub into a test mode and sends the status information of the system to the App. Each status has an expected correct result which indicates passed or failed test. The following items are tested in this manner: firmware version, battery level, motion sensor, memory access.

MicroHub devices are programmed with the SAAM Application software in a post-production process. The devices that passed the tests are programmed (flashed) with the SAAM application software. Further verification tests ensure that the individual MicroHub device sends data according to the specified SAAM protocol.

With a test App F009_Protocol_checker, the MicroHub devices are connected to the App and after setting the mode to stream data, the measured data are sent over BLE to the App where the data are displayed. If the MicroHub is oriented correctly and kept stable the accelerator data should display the x,y,z gravity values in g (1,0,0). This step verifies the correctness of the protocol and measured sensor data.

3.5.1 MicroHub Connectivity, Data transfer and Management Testing

Sens belt, Sens bed and Sens bracelet sensor devices are available to end users in order body movement and physical data to be collected for analysis. The data is transferred from the sensor devices through the user's mobile device by Bluetooth Low Energy (BLE) to the backend of the SAAM Platform in order analysis for coaching purposes to be performed.

The MicroHub controller is installed on every sensor device and provides the connectivity and data transfer between the sensor devices and user's mobile device.

The mobile application is developed to maintain stable connection of the mobile device to the sensor devices and connect automatically.

Users can see a list of all connected devices under the device settings section of the mobile application and will be notified if the connection is interrupted. Users can add, remove, and maintain sensor devices through the mobile application and track the status of the connection.

3.6 Wi-Fi Access Point

Since not every household has a pre-existing Wi-Fi access point, a dedicated Wi-Fi access point (AP) is included in each deployment package. Pre-configured Wi-Fi AP simplifies the deployment, since SAAM-specific preconfigured SSID and password simplifies the steps needed in a deployment procedure and thus decreases the number of possible points-of-failure in the system installation. Additionally, a dedicated Wi-Fi AP separates SAAM local network from the pre-existing local network, thus further increasing the security of the SAAM system. A dedicated Wi-Fi AP is a standard low-cost off-the-shelf Wi-Fi AP with standard firmware and only changed SSID and password.

This project has received funding from the European Union's Horizon 2020
research and innovation programme under grant agreement No 769661

4. APPLICATION FOR USER-SIDE INFRASTRUCTURE DEPLOYMENT VERIFICATION

As part of the user-side infrastructure deployment, each device needs to be individually tested for connectivity and basic operation. For the purpose of user-side infrastructure deployment testing, a special web-based application was developed. It runs on an eGW device and serves as the user interface with testing and deployment procedures that guide a deployment engineer through the deployment process. The application puts the initial settings to the eGW device and collects test results. As the last step of the deployment procedure, the deployment app records the installation-time NILM algorithm labelling recordings.

The web application is architecturally split into a business logic layer (back end) and a presentation layer (front end). The back end provides the necessary functionality for various deployment aspects via APIs for the presentation layer, while the front end implements the graphical user interface (GUI). The back end is based on Node.js and Express application framework, whereas the front end is developed using React in connection with the Material Design.

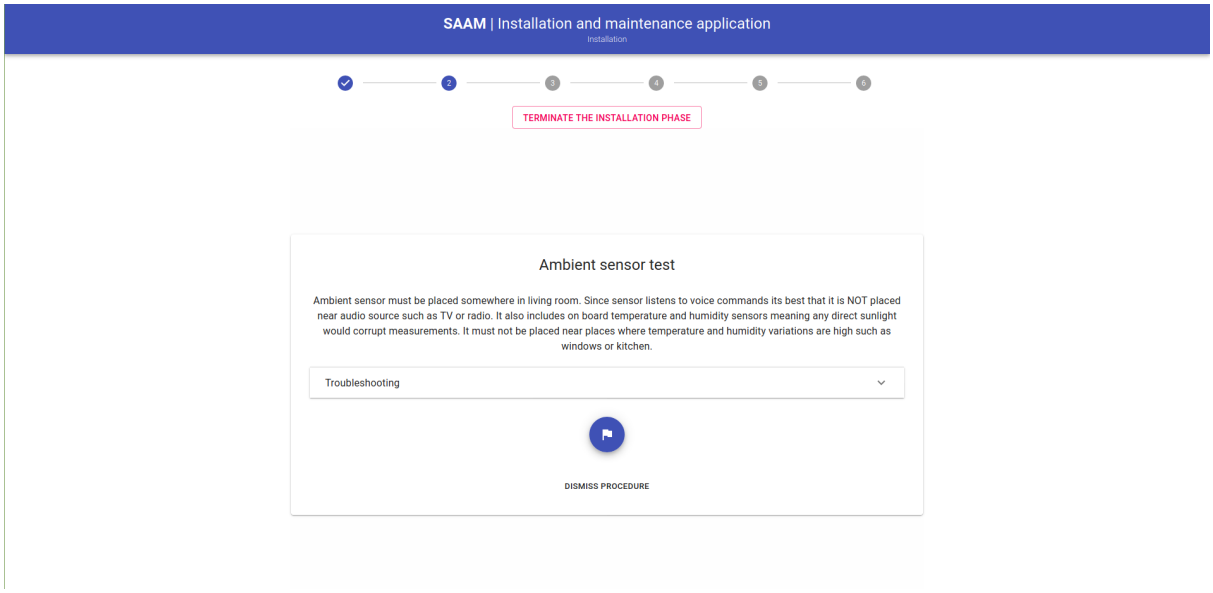


Figure 4: Web application for user-side infrastructure validation and integration; ambient sensor test procedure example.

On the initial application run, the deployment engineer is prompted to select the country and location identifier of the current deployment location. Next, various test and calibration procedures can be executed, either altogether in a sequence as a part of the deployment process or individually as a part of additional maintenance operation. An example test procedure is illustrated in Figure 4.

Each procedure provides simple instructions and a troubleshooting guide. If a procedure fails (e.g., due to device connectivity problem), comprehensive context-dependent guidelines are displayed. All procedure results are reported to a cloud database. The application also provides rich logging

This project has received funding from the European Union's Horizon 2020
research and innovation programme under grant agreement No 769661

capabilities that enable extensive means of deployment failure troubleshooting. When the connection to the cloud is non-existent, the application maintains a local back up storage on the eGW device.

The Web app uses test APIs for individual device status information gathering. A test API calls a script that makes a remote procedure call (RPC) to the desired device. On the called device, RPC triggers a function that makes a corresponding status checking actions (described in Section 7). If a device does not respond or returns an error, the deployment engineer is provided with comprehensive reports and troubleshooting guidelines.

As mentioned before, eGW triggers the test RPC calls, and tested devices are those that are listening and responding to the calls. Thus, eGW acts as an RPC client and tested device as an RPC server.

In order for calls to be made, eGW (RPC client) has to obtain server's IP address. That is achieved through Avahi service, which is a service running on all devices advertising themselves on the network, which ensures devices will find each other on a LAN network.

The deployment process consists of the following procedures:

1. **Internet connectivity test**. Checks whether the eGW device has internet connectivity by connecting to *https://www.google.com*.
2. **Ambient sensor test**. Checks whether Raspberry Pi devices are connected to the Wi-Fi AP and properly configured.
3. **Voice module training**. Configures the voice command module.
4. **PMC test**. Checks whether PMC devices are connected to the Wi-Fi AP and properly installed/configured.
5. **UWB test**. Checks whether UWB devices are connected to the eGW device and properly configured and placed.
6. **Appliance labelling**. Records the installation-time NILM algorithm labelling by using predefined labelling steps. The labelling is performed by following simple on-screen menus with instructions and manipulating various household appliances.

5. USER-SIDE DEVICE DEPLOYMENT/INTEGRATION TESTING

This describes the assembly and deployment tests for individual user-side devices. In addition, the description in the deliverable is complemented by two videos describing each individual user-sided device and their status indicator. They are added to the report as Appendices A7.1 and A7.2.

5.1 Internet Connectivity Test

Internet connectivity test checks if eGW is connected to the Internet. The Internet connection is an essential service for SAAM system since it provides connectivity to the remote cloud where the data from the user-side devices are stored and processed by coaching algorithms. Coaching algorithms generate appropriate coaching actions that are sent back to the user or user's social circle. The internet connectivity is checked by connecting to the known predefined public server that is up and running. During internet connectivity test the connection to *https://www.google.com* is verified.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 769661

The Internet connectivity test indirectly checks the Wi-Fi AP settings, its connection to the Internet and eGW's network settings. Potential problems may occur if eGW is out of Wi-Fi AP's range, thus it should be moved closer or a Wi-Fi range extender should be used.

5.2 UWB Device Test

The SAAM UWB system consists of three UWB devices. The two stand-alone (SLAVE) devices are mounted in a socket-on-socket case while the third (MASTER) is built-in in an eGW device. In order to test the connectivity and to ensure the required functionality of the UWB network deployment, the UWB system is tested for:

- connectivity between all three UWB devices, and
- correct layout of the devices in the apartment/house.

The main problem arises if the UWB modules are out of the range of the coordinating (MASTER) UWB module within the eGW device. The status of UWB modules connectivity is available through a dedicated test application on eGW. If the connectivity problems occur, the devices (eGW and UWB modules) should be moved closer to each other while still trying to assure required layout as described in D3.1, Chapter 3.3 Deployment requirements.

5.3 Ambient Sensing Device Test

The Raspberry Pi based ambient sensing device is connected to the Wi-Fi AP using a Wi-Fi interface. Potential problems may occur if devices are out of each other's range, thus they should be moved closer or a Wi-Fi range extender should be used. The ambient sensor device test is comprised of a sequence of procedures that test basic operation of individual ambient sensor modules. The ambient sensor device test is initiated at the system deployment by the eGW device. The eGW device may also provide current depersonalized location ID. The ambient sensor device test procedures include:

- Update the depersonalized location ID, if given.
- Verify the presence of the Matrix Creator board and its operation.
- Test the operation of data acquisition modules and their connection to the cloud.
- Verify that the voice command module and message rendering module are properly initiated.

5.4 PMC Device Test

The PMC test includes a dedicated procedure for testing the basic operation, correct wiring and connectivity. In terms of connectivity, potential problems may occur if the PMC is installed outside the apartment or in the metal fuse box out of the Wi-Fi range. In such cases either Wi-Fi AP should be moved closer or a Wi-Fi range extender should be used. In addition to the basic operation, the testing procedure identifies mistakes in the installation yielding, for instance, negative power, no power or missing data on one of the channels.

This project has received funding from the European Union's Horizon 2020
research and innovation programme under grant agreement No 769661

Testing of the PMC is done through the dedicated deployment web application. There are two requirements for the installed PMC to be able to pass the test. PMC should be connected correctly and at least a small load on the power line should be detectable (at least one appliance consuming some power). The detectable power requirement ensures the reliability of the tests (i.e., without the load the wiring cannot be verified).

5.5 MicroHub Device Testing

MicroHub devices are individually pre-tested by the partner IAW. Testing of the eGW software service is done during the development. The MicroHub communication protocol is tested separately from MQTT and system BLE parts of the service, which ensures that SW and HW problems can be independently detected and resolved.

5.5.1 eGW Configuration for MicroHub Devices

Pairing of MicroHub devices and eGW is done by an automatic script, which is ran during the kit assembly process. The process is explained in **Error! Reference source not found.**. Devices that are a lready pre-tested by IAW are powered up and recognised by eGW through BLE advertising packets. At this point, the potential problems can emerge if expected devices are not present. This could be caused by a faulty eGW, where BLE misconfiguration or faulty HW will prevent BLE scanning. When the issues are resolved the script should be re-run again. After the successful MicroHub initialization, the configuration is saved on the eGW and its output data (MAC addresses and BLE advertising names) are saved to the deployment documentation. At this point the MicroHub devices can be turned off.

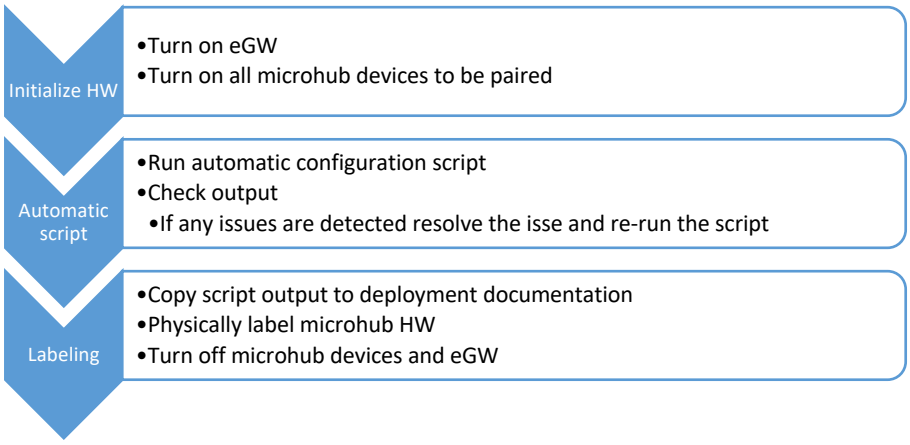


Figure 5: Process of configuring MicroHub devices for each deployment site separately.

5.5.2 eGW-based MicroHub Deployment Issues Handling

When the MicroHub system is deployed, the eGW device automatically establishes connection to the receiving MQTT server and starts searching for the pre-configured MicroHub devices, connects to them when available, and starts transmitting the data. If any issues are detected (variation outside of specified MicroHub BLE protocol), the MicroHub device is disconnected and eGW tries to re-establish a new connection. On each connection, the MicroHub device is sent to IDLE mode and back to S3

This project has received funding from the European Union's Horizon 2020
research and innovation programme under grant agreement No 769661

mode to ensure the known state of the device when sampling starts. Because of the nature of BLE (lossy connection, dropped packets...) a better and more reliable fault correction is not viable during the deployment.

All major events (MQTT connect, BLE connect/disconnect, BLE scan report etc.) during the deployment are logged on eGW, to help pinpointing any raised issues when the devices are already deployed.

Service running on eGW already handles some of the errors on the fly. These issues were detected during test deployments, and are implemented in the service code. Each type of the MicroHub device has its own code-path to handle device specific deviations from the communication protocol defined in the documentation as follows:

- Wristband sensor device is sending BLE packets exceeding 20B length that is governed by BLE 4.1
    - solved by eGW BLE workaround.
- Bed and clip sensor device irreparably hung when connection was lost (e.g. user walked out of the coverage)
    - solved by IAW FW update.
- All MicroHub devices have variable sampling frequency. Expected sampling frequency is 56 Hz while actual sampling frequencies in bed and clip sensor devices are around 52 Hz and in wristband devices around 112 Hz.
    - This is partially handled on the eGW (dropping every second sample from the wristband device) and partially in the pipeline where frequency is discarded.

6. ASSEMBLY OF DEPLOYMENT KITS

To mitigate some risks for possible deployment failures, complete deployment kits were assembled at JSI premises before shipping to corresponding partners handling on-site deployment at piloting locations. Each deployment kit was assembled in a way that resembles the deployment configuration as close as possible. Before packaging of the individual deployment kit, a complete testing of all devices was made to check and confirm complete system for required performance.

6.1 Wi-Fi Access Point Configuration

SSID and password settings are changed in each Wi-Fi AP. All SAAM devices are set in advance to connect to predefined SAAM SSID and thus automatically connect to the Internet. No additional setup steps are needed at the deployment site.

6.2 Setup eGW

Setting up the eGW device starts with flashing the eGW system image with a master micro SD card. The system boots for the first time after the flashing process and obtains its unique ID (machine-id). The device ID is linked with the depersonalized location ID and written down in a spreadsheet for tracking the device status. Since the device's IP address is dynamic and thus can change during the

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 769661

operation, the device ID links the current device IP address and the device ID in the central device management system, which enables the remote access to the running device.

After the initial setup is complete, the device is tested for Internet connectivity using SAAM deployment app. With everything in place, a sticker with the location ID is put on the eGW enclosure and left in operation as a testing device for remaining SAAM device components.

6.3 Setup PMC

Setting up the PMC device starts with flashing the PMC system image on the micro SD card which was later inserted into PMC device itself. In the next step, the image is written to the PMCs built in eMMC memory. Afterwards the system will boot for the first time and obtain its unique ID (machine-id) which is written down to a spreadsheet for tracking its status.

The PMC testing process starts with wiring the PMC device in a test fuse box and continues with running the SAAM deployment web application tests. If tests are successful, the PMU device is labeled with the identification stickers and put in a transport box.

6.4 UWB Setup

The UWB setup starts by plugging-in two socket-on-socket UWB devices in the office resembling triangular shape together with the third UWB device integrated in the eGW device. If the UWB tests in the SAAM deployment web application are successful, the devices are labelled with corresponding identification stickers and put in a transport box.

6.5 Ambient Sensing Device Setup

The micro SD card with a predefined location ID is inserted in the testing ambient sensor device's card slot. The device is powered on and ambient sensor device tests are run in the SAAM deployment web application. Subsequently only the pre-configured micro SD card for ambient sensing device is put into the dedicated transport box since the individual components for the ambient sensing devices are shipped separately and assembled at the deployment location where the preconfigured micro SD card is put into the system.

6.6 Packing MicroHub Devices

Since MicroHub devices were pre-tested (as described in Section 5.5), they were just labeled with identification stickers corresponding to the given deployment site and put in a transport box.

6.7 Check the Inventory List

This project has received funding from the European Union's Horizon 2020
research and innovation programme under grant agreement No 769661

As a last step, the eGW device is powered off and put into a dedicated transport box. The inventory list for a particular deployment site is checked again to ensure all necessary components are included. Finally, the transport box is marked with the corresponding location ID and sealed.

Figure 6: Packaging procedure.

7. SYSTEM DEPLOYMENT PROCEDURE

During the on-site deployment procedure, the deployment team has to follow the pre-prepared guidelines to ensure that the devices are correctly deployed and successfully pass the tests. The purposely developed deployment web application guides the deployment engineer throughout the deployment process with detailed per-device steps. The deployment process is finished by the house appliances labelling process, where appliances fingerprints on power lines are recorded. The process of the deployment is thoroughly described in a video which is added to this deliverable as Appendix A7.3.

7.1 Wi-Fi AP and eGW Deployment

In the first step the SAAM Wi-Fi AP is deployed and connected to the modem device in an apartment using the Ethernet cable and powered up.

When Wi-Fi AP is up and running, the eGW device has to be placed to the foreseen operating location and connected to the external power. The deployment procedure is guided by the SAAM deployment web application running on the eGW device.

7.2 SAAM System Deployment Setup

The deployment procedure continues by starting the SAAM deployment application. When the eGW device is up and running, the deployment application web interface can be accessed on http://saam.local using a standard PC or any mobile device with a web browser. The web interface

This project has received funding from the European Union's Horizon 2020
research and innovation programme under grant agreement No 769661

depicted in *Figure 7* shows the SAAM system installation setup screen, where the deployment engineer selects the country of the deployment site and anonymized location identifier assigned to the site beforehand (e.g., SI99).

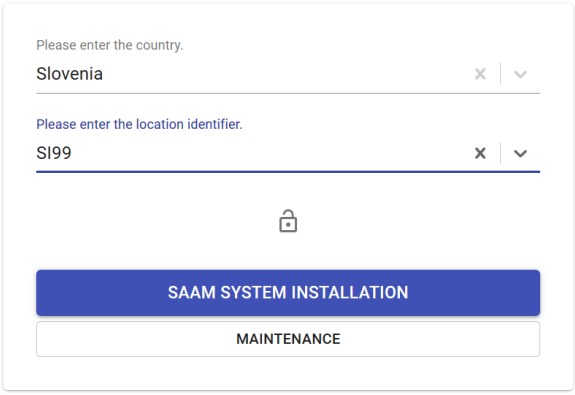


Figure 7: Web application startup screen

7.3 Internet Connectivity

In the next step, the correctness of the Wi-Fi AP and eGW connectivity settings is checked. The Internet connectivity test starts by connecting to the https://www.google.com. By passing the test, the availability of the Internet connection and correctness of the network settings necessary for the basic SAAM system functionality is confirmed. In the case of the Internet connectivity test error, the deployment engineer has to check the Wi-Fi APP deployment and possible issues with the modem device.

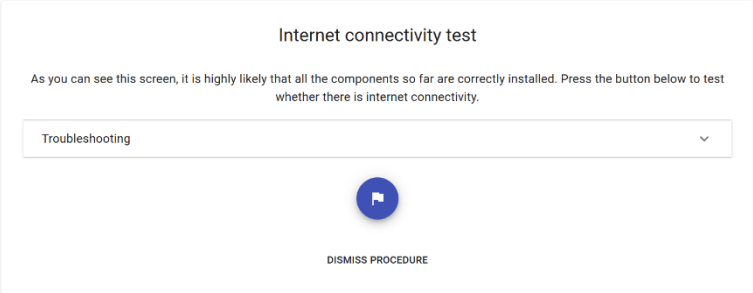


Figure 8: Internet connectivity test interface.

7.4 Ambient Sensing Device Deployment

The ambient sensing device is assembled at the deployment location from the individual components. The Matrix Creator board is connected to the Raspberry Pi extension header, the USB speaker is connected to the USB connector and auxiliary audio connector of the Raspberry Pi, the pre-programmed micro SD card is inserted into the Raspberry Pi, and finally all ambient sensor electronics components are placed into a wooden enclosure.

The Ambient sensing device can be placed on any horizontal surface which is large enough in order to prevent the overheating. There should be a clearance of at least 10 cm on each side of the device and an open space on top.

This project has received funding from the European Union's Horizon 2020
research and innovation programme under grant agreement No 769661

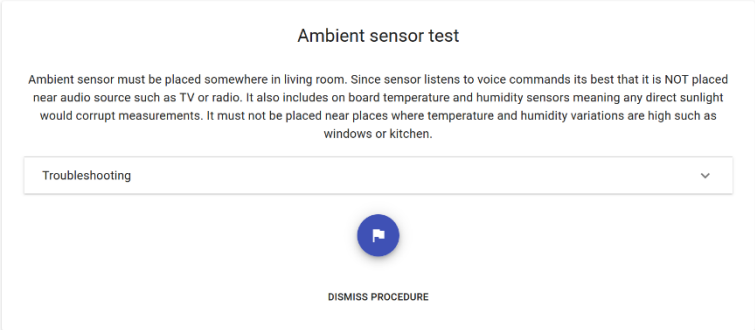


Figure 9: Ambient sensor deployment testing interface.

7.5 Voice Module Training

The voice command module uses the Snowboy Hotword Detection Engine for word recognition. The keyword detection in the Snowboy Hotword Detection Engine is based on personalized hotword models. To train the hotword recognition models the user must pronounce and record each keyword a couple of times and send the recording to the Snowboy Hotword Service. The Snowboy Hotword Service returns trained models which are stored for later use. Since the voice command module has to detect the keywords pronounced by the primary user, the models must be built at the time of deployment.

The voice training procedure, which is triggered by the web application for testing, guides the primary user to build the required models. Since the ambient sensor device is a headless system, the prerecorded sound messages are used for user interaction. For each provided keyword the voice training module instructs the user to pronounce the keyword three times, sends the sound recordings to the Snowboy Hotword Service, tests the built model by additional keyword recording, and stores the model. If the keyword detection fails, the training of the keyword is repeated. However, the number of training repetition of each keyword is limited to three repetitions.

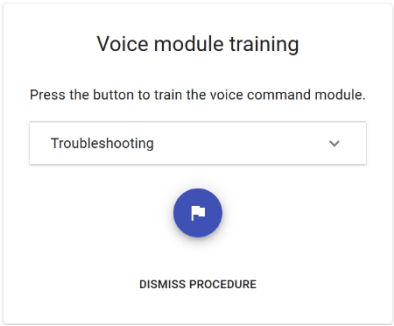


Figure 10: Voice module training procedure interface.

This project has received funding from the European Union's Horizon 2020
research and innovation programme under grant agreement No 769661

7.6 PMC Deployment

The PMC deployment test checks if the PMC device is correctly deployed and has the Internet connectivity. Based on the measurements of the power-line parameters, the erroneous measuring transformer clip orientation and/or invalid voltage input connections are detected. In the case of deployment errors, the interface presented in *Figure 11* reports the most possible cause of an error and suggests troubleshooting actions.

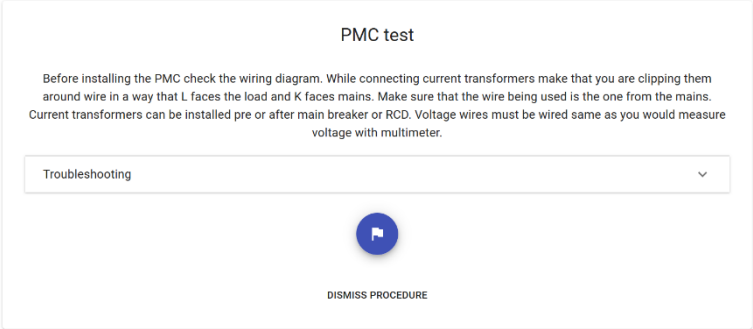


Figure 11: PMC deployment test interface.

7.7 UWB Deployment

The web application interface for UWB deployment test is shown in *Figure 12*. The test checks the number of active UWB devices and tests the positioning of devices. Most errors are due to the large distance between the devices which prevents establishing a reliable communication between them. In case of an error, the application suggests the possible steps needed to overcome the problems.

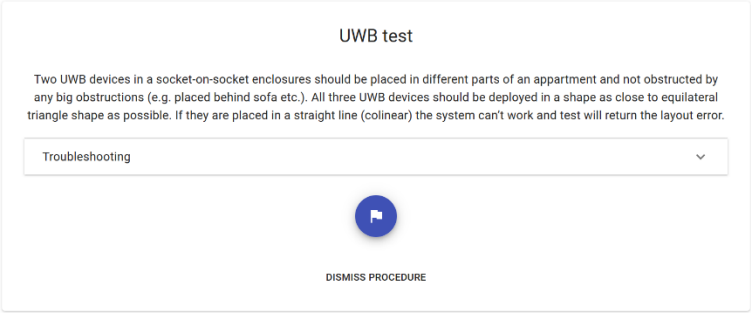


Figure 12: UWB system deployment test interface.

7.8 Household appliance labelling

The installation-time household appliance labelling is done as a last step of the on-site deployment procedure, and it uses already prepared web application. The labelling is performed by following simple on-screen menus with instructions for manipulating various household appliances. The raw data generated by the PMC device (frequency, voltages, currents, powers, …) together with the timestamps of characterizing the appliances events (ON or OFF) are sent to the Postgres database. For each deployment site, the same labelling procedure is used and recordings are distinguished in the database by the predefined depersonalized unique location ID. Because of numerous site specifics,

This project has received funding from the European Union's Horizon 2020
research and innovation programme under grant agreement No 769661

the characterization of appliances using the database is done manually for each piloting location. The generated database of labelled appliances is subsequently deployed back to the PMC device after each characterization.

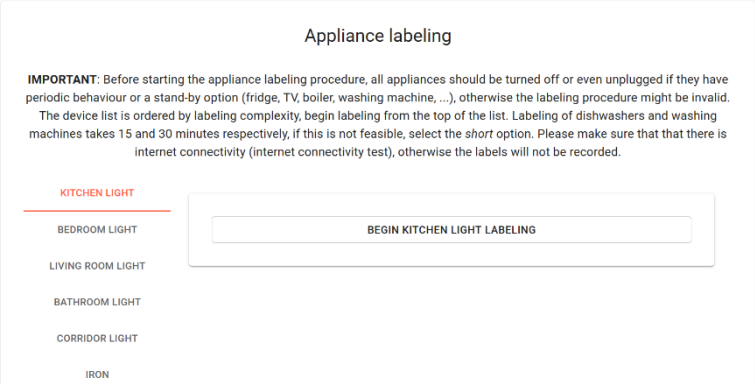


Figure 13: Deployment appliance labeling interface.

7.9 MicroHub Devices

During the deployment procedure the MicroHub devices need to be installed and turned on. They are preconfigured on the eGW device and are automatically detected upon powering up. If the MicroHubs should be used with the mobile application, the paring with the mobile should be done before the eGW device is turned on, or outside of the eGW BLE range. Otherwise, eGW will connect to the MicroHub device and prevent MicroHub BLE advertising and connecting to the mobile application.

8. CONTINUOUS SAAM "SYSTEM HEALTH" MONITORING

After the successful deployment, a dedicated system for health monitoring of the devices was established. The status of high level devices is continuously updated every hour. The system is implemented according to the protocol described in D2.5, Chapter 4.

A service for obtaining the status of the PMC, ambient sensor and eGW devices was created. It runs on eGW and uses avahi service discovery on LAN in order to find IP addresses of devices at each individual piloting site (described in Section 4, paragraph 6). The Python script pings the obtained IP addresses and creates a JSON file according to the responses. If a device responds, the "up" tag is recorded, otherwise it records the "down" tag. The compiled JSON file is subsequently sent to the SCALE database via MQTT protocol, where each device can be found under /saam/health/loc-id. The loc-id is the predefined ID of the SAAM system installation location. In this way a complete overview of the system health is at disposal at any time.

For all backend services there is a health check set up and Kubernetes constantly checks to see if they are working. We use the built-in mechanisms of .net core and Kubernetes to ensure that all our micro services work and if someone accidentally stops it will be launched automatically. Additional information on the System Monitoring can be found in Deliverable 2.7.

This project has received funding from the European Union's Horizon 2020
research and innovation programme under grant agreement No 769661

9. REFERENCES

[1] N. Saeed, H. Nam, M. I. U. Haq in D. B. Muhammad Saqib, „A Survey on Multidimensional Scaling,“ ACM Comput. Surv., zv. 51, pp. 47:1--47:25, 5 2018.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 769661

Introduction

Document Overview

This Test Report provides a summary of the results of test performed as outlined within this document.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 769661