D6.3a CommonSense Dashboard V1.0

Project	SWELL
Project leader	Wessel Kraaij (TNO)
Work package	WP6
Deliverable number	D6.3a
Authors	Joris Janssen (Sense-OS), Jan Geert van Hall (Almende)
Reviewers	Wessel Kraaij, John Schavemaker
Date	December 2013
Version	1.0
Access Rights	Internal (project only)
Status	Definitief for deliverables Q4 2013

SWELL Partners:

Noldus, Novay, Philips, TNO, Radboud Universiteit Nijmegen, Roessingh Research and Development, Almende, Sense Observation Systems, Universiteit Twente.

Summary

In October 2012 Ericsson has withdrawn from the SWELL project, and has been replaced by Almende BV and Sense Observation Systems BV. The role of the M-to-M platform of Ericsson has therefore been replaced by the CommonSense platform. This platform will play a central role in SWELL to store, aggregate and process all sensor data to monitor the well-being at work of the SWELL target group, being knowledge workers. This data can then be used by PhD students to build & test applications that display and interpret behaviour.

One of the applications on the output side is a Dashboard, that displays progress of parameters such as steps, sleep and social interaction. This Dashboard will be an element of the Golden Demo of the SWELL project. As such, this is a software deliverable, of which this document is a functional and technical outline. The document also gives an outline of the further developments towards the Golden Demo on the mid-term review, chief amongst which is the implementation of the privacy and authentication policy.

Contents

1 SWELL Objectives

The SWELL project within the COMMIT program aims to research ways to improve the well-being at work of knowledge workers. Knowledge workers are defined as people who use and produce information as their main task. Well-being at work is defined as 'being and feeling in control'.

The methodology of SWELL consists of the gathering of information on the physical and mental state and context of the knowledge worker, interpret this information and, through smart reasoning, provide feedback and advice on how to improve well-being.

The uniqueness of SWELL lies in the reasoning with input from *multiple* sensors and information sources providing a coherent *context*, and the *personalisation* of the feedback with continuous improvement.

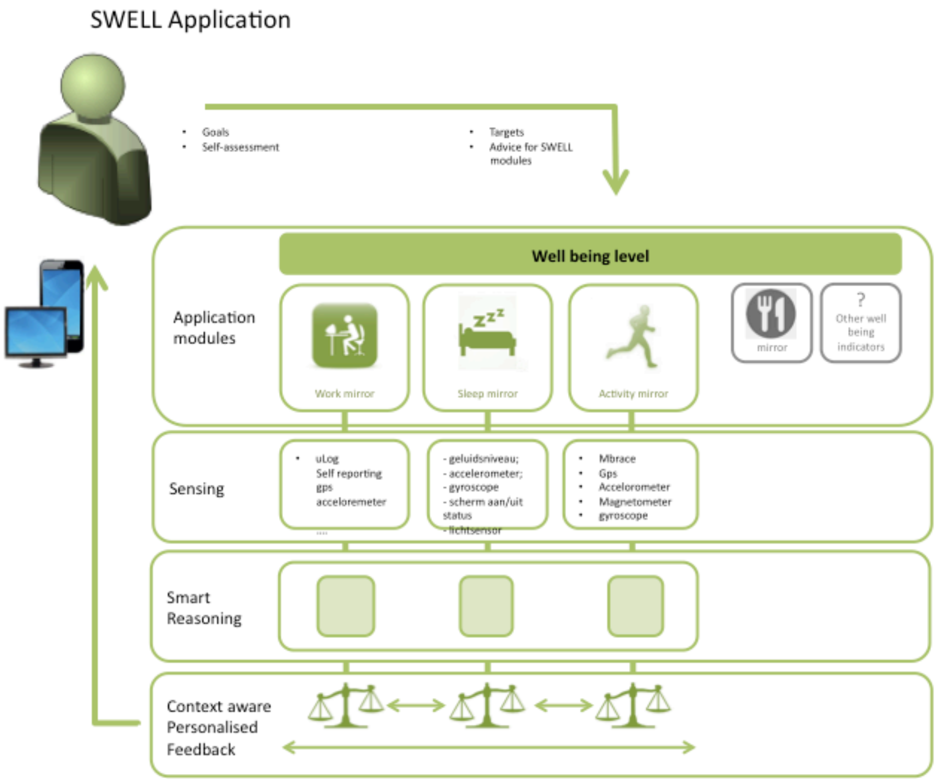


Figure 1. Swell methodology: closed loop of continuously improving feedback

1.1 Context of this deliverable

In October 2012 it became clear that Ericsson would not contribute further to the project. Replacement was found by adding Sense Observation Systems BV and its parent company, Almende BV to SWELL. Both companies were already part of the COMMIT consortium. The technology of the initial plan to be substituted was the Ericsson M-to-M platform. To do this, CommonSense was selected as the central platform in SWELL to store and process sensor data, to be used by the SWELL PhD students as primary source of data for their applications, to be connected by the CommonSense API.

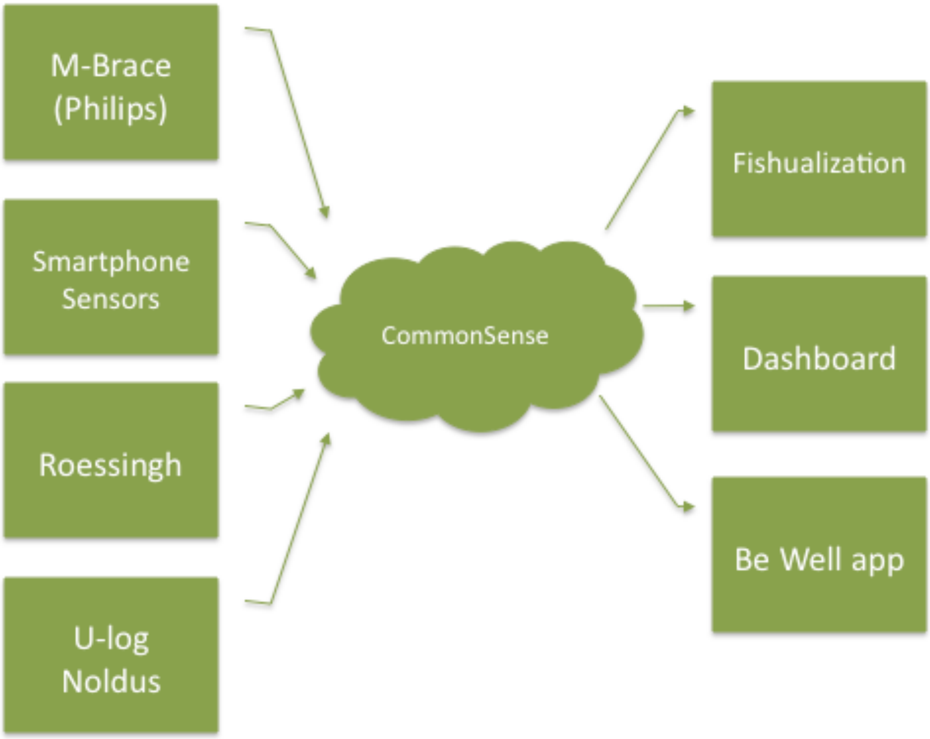


Figure 2. CommonSense serves as the central Hub for data exchange and processing in SWELL. Applications as selected preliminary for the Golden Demo need to be connected to this platform

During the Project Management Team discussion of the scope of the first SWELL demo during the mid-term review we selected a set of modules to develop, with the initial focus on ‘mirroring’ or ‘displaying’ behavioural patterns. A central place in this set has to be taken by the ‘Dashboard’, a personal application that tracks various types of activity, tied to place and time. This document outlines the functional and technical specifications of this Dashboard.

As such, this document is the description of what is essentially a software deliverable.

There is a strong connection with the coincidental SWELL deliverable D4.6 & D4.8 “Design and evaluation of context privacy controls” (Hulsebosch et al.) which describes in more detail the security and privacy policies and user interaction of this dashboard.

1.2 CommonSense

CommonSense is a platform that enables individuals to keep track of all their sensor data, store it in a central location, and play with it. CommonSense also processes raw sensor data into meaningful things like sleep, exercise, or your top locations.. With CommonSense tracker a smartphone can be turned into an advanced tracking device.

The data from smartphone sensors can be combined with data from external sensors, such as a Fitbit. Sense integrates any Sensors on the market it deems interesting. Additionally, external users and developers can add their own sensors via an API.

The API enables developers and 3rd party individuals and companies to develop their own applications using CommonSense. As such, it is a generic platform. Sense Observation Systems do develop applications on this platform themselves as well, notably in the mental health care sector.

The general structure of the platform is such that all data collected by an individual will remain the property of said individual. Sense Observation Systems has no rights whatsoever to the data and thus cannot resell this data. Only the individual owning the data can read, share and delete this data.

2 Functional description and method - Dashboard

2.1 User stories and method

The dashboard provides mirroring of lifestyle functionality in order to change behaviour. To reach this, users should be able to complete the following tasks:

A Create and manage an account

B Download and install an app that tracks smartphone data

C Daily review of tracked data in an easy and intuitive format

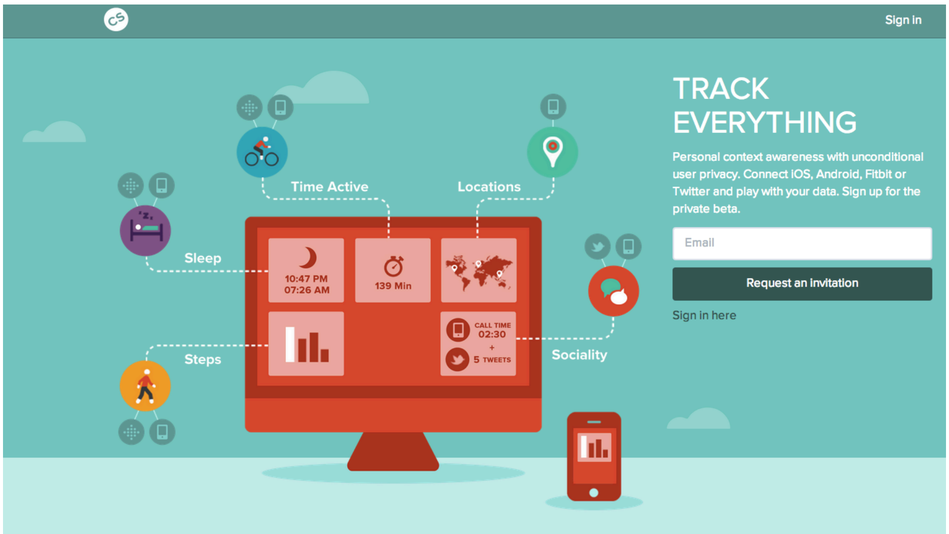
D Monthly reflection over longer periods

E Have control over their own stored data

After defining the user stories, wireframes (analogue and later digital) were created. These were tested with several naïve and non-naive potential users. Wireframes were adjusted and updated based on the feedback that was received. Implementation started from the wireframes. The wireframes were later translated to a graphic design, which was subsequently implemented.

2.2 Result

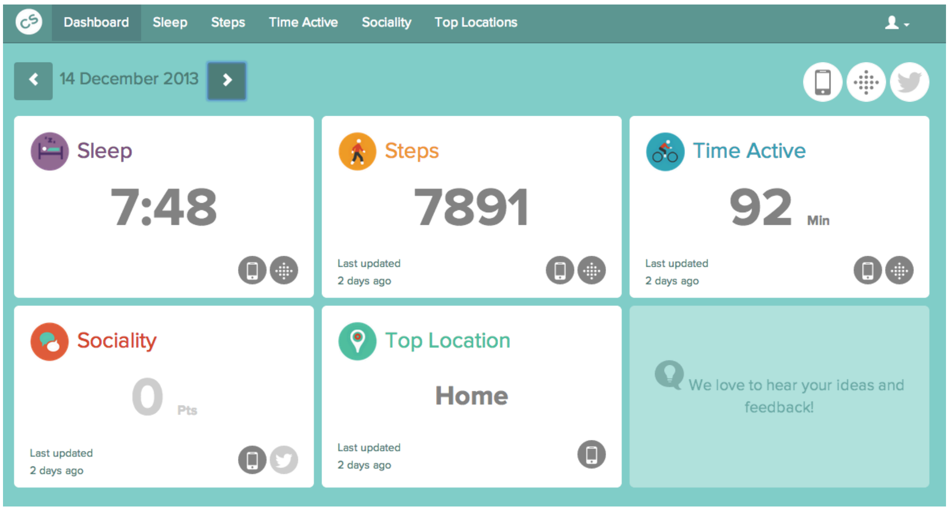
Resulting functional description will be done by user story through several screenshots that visualize the resulting functionality.



TRACK EVERYTHING

Personal context awareness with unconditional user privacy. Connect iOS, Android, Fitbit or Twitter and play with your data. Sign up for the private beta.

Figure 3. User story A is implemented in the interface above. Creating an account is kept simple, users only have to fill out an email address. They will receive an email that will guide them through the different steps. Passwords can be changed when logged in.



14 December 2013

Sleep
7:48

Steps
7891
Last updated
2 days ago

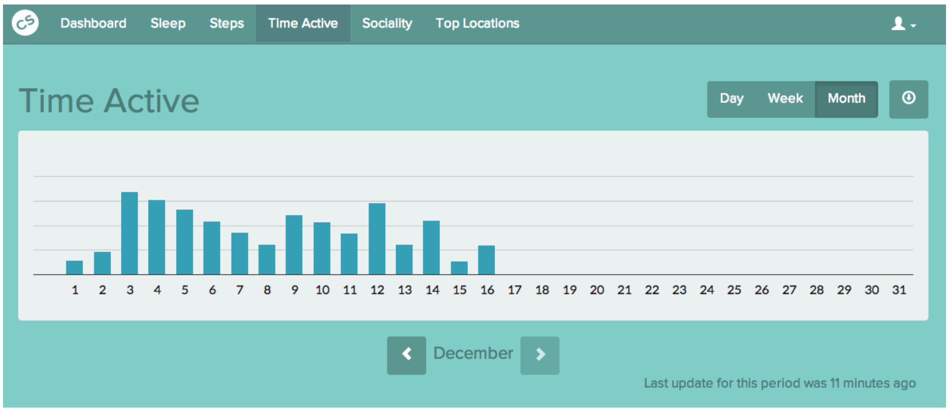
Time Active
92 Min
Last updated
2 days ago

Sociality
0 Pts
Last updated
2 days ago

Top Location
Home
Last updated
2 days ago

We love to hear your ideas and feedback!

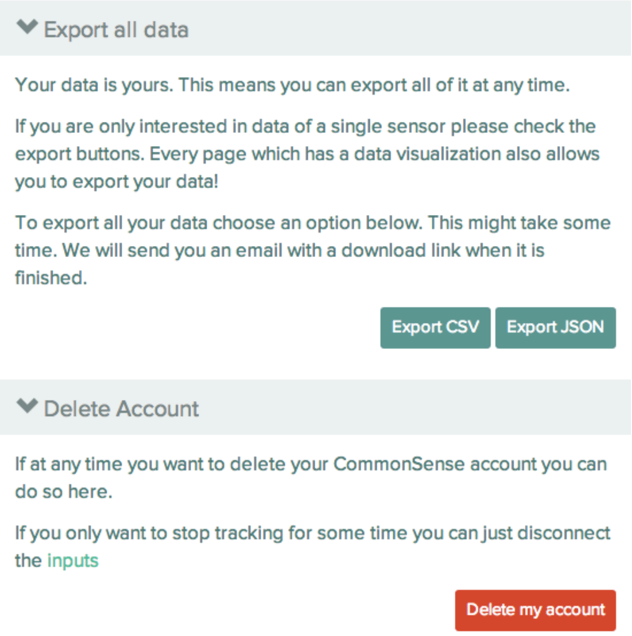
Figure 4. The interface above shows the implementation of user story C – Daily review of tracked data in an easy and intuitive format. For each personal state, the general score for that day is presented. The interface provides a glanceble solution to getting a quick and intuitive overview for daily scores.



Time Active

Last update for this period was 11 minutes ago

Figure 5. The interface above describes the implementation of user story D – Monthly overview for reflection. By clicking on one of the modules presented in Figure [C], the user gets into this longer term view, where daily, weekly, or monthly graphs can be viewed for longer term reflection.



Export all data

Your data is yours. This means you can export all of it at any time.

If you are only interested in data of a single sensor please check the export buttons. Every page which has a data visualization also allows you to export your data!

To export all your data choose an option below. This might take some time. We will send you an email with a download link when it is finished.

Delete Account

If at any time you want to delete your CommonSense account you can do so here.

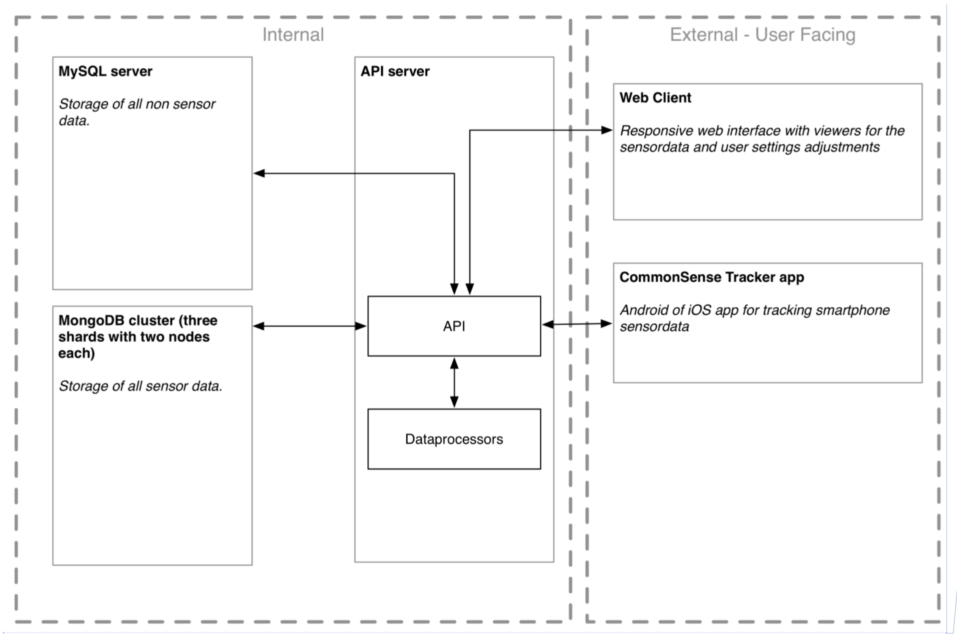
If you only want to stop tracking for some time you can just disconnect the inputs

Figure 6. Users have full control over their data. This means they can always download it or delete it (User Story E). The interface above presents the current implementation of this.

3 Technical description

3.1 General architecture

The general architecture of the CommonSense infrastructure consists of different client side (internal) and server side (external) elements. Details of the most important elements for the current deliverable are elaborated below.


Internal
External - User Facing
MySQL server
Storage of all non sensor data.
API server
Web Client
Responsive web interface with viewers for the sensordata and user settings adjustments
CommonSense Tracker app
Android of iOS app for tracking smartphone sensordata
MongoDB cluster (three shards with two nodes each)
Storage of all sensor data.
API
Dataprocessors

Wessel Kraaij 12/19/13 11:39 AM
Comment: Of→ or

Figure 7. Graphical depiction of the general architecture of the CommonSense infrastructure.

3.2 Client side implementation

The client side of the Dashboard consists of two apps and a web interface. The two apps (one for iOS and one for Android) provide tracking functionality for the smartphone sensors and run on the phone in the background. If available, they use the phone's accelerometer, gyroscope, magnetometer, noise sensor, ambient light sensor, GPS, and call states.

The development framework for the web interface is AngularJS. It is a mature framework enjoying widespread adoption, corporate backing and community support. Other items that played a critical role in its selection were:

- Testability via dependency injection

- Backwards compatibility with older browsers via polyfills and shims
- Built-in localization (l10n) and internationalization (i18n) support
- Adaptability to Content Security Policies (CSP)
- Clean separation of concerns

The application is implemented as a stand-alone HTML5 application and accesses data over the network via CORS secure requests against the Sense API server.

For the user interface the CSS framework Bootstrap (v3) is used as a foundation. CS-dashboard is developed with Bootstrap's mobile-first responsive strategy to provide an interface that works well on devices of all sizes and inputs.

The CommonSense dashboard currently supports the following browsers.

- Chrome (Desktop/mobile, latest)
- Safari (iOS/OSX, latest)
- Firefox (latest)
- Internet Explorer (older versions on Internet Explorer are not supported)
- Opera (latest)
- Android Browser (2.0+)

3.3 CommonSense API

CommonSense functionality is exposed through a RESTful JSON API that employs HTTP requests. With this API all functionality of CommonSense can be precisely controlled, as CRUD operations are available for each of the resources of the data model. An exhaustive listing of the available API endpoints can be obtained from http://developer.sense-os.nl.

3.4 Sleep algorithm

The sleep time algorithm computes the amount of hours a user has slept in the last 24 hours. It uses several sensors from the mobile phone to do this:

- Motion energy
- Audio noise level
- Proximity
- Gyroscope
- Accelerometer

- Camera light values

Features like the mean and variance are computed for each of the sensors after which weighted prototypes for sleep and awake are created. Based on the user's daily pattern, these prototypes are updated over time. Using nearest-neighbour classification every 5 minutes of data is classified as sleep or awake. The total time and start and end time of the nightly sleep is calculated taking the whole classified period in perspective.

4 Future developments

The current dashboard constitutes a minimum viable product for a lean learning methodology. This means that we use it to learn which features provide users most benefits and which features might not be necessary. Moreover, we can use it to monitor problems and bugs.

Now that the main functionality is implemented, as a next step we want to improve the stability of the clients and algorithm. There are still many bugs in the clients, as is generally the case for a beta software product. We have setup monitoring tools to track the most common bugs and will be implementing fixes to these bugs over the coming period.

We also want to improve the sleep algorithm. Currently, we have tested it with a limited group of people, resulting in accuracies around 75% of classifications being within 30 minute accuracy. The classification accuracy depends mostly on the use of the smartphone during the day and night. By gathering and analyzing data from more individuals we hope to learn how the algorithm can be further improved.

At present the social interaction that can be used to derive work-related activity is still rather limited, monitoring phone calls for pick-up and call-back. This will be extended to reflect activity on social networks and other communication channels such as mail.

Finally, we are working in workpackage 4 on implementing sharing functionality in the dashboard so that users will be able to share their data with others. Initial wireframes and testing of the wireframes have been finished and updated designs are currently in development for this.