Software project documentation

# Background and current status of development

The software project will be related to a new startup company, eloSpaces, where I’m one of the founder. While the most important outcome of the project is to widen my personal skill set, it will also help the company by providing an actual prototype. The company consist of 5 persons both from Finland and China. My responsibilities are focused on the technical concepts and development.

eloSpaces focuses on smart spaces and everything related to smart living. Our first product is eloSeed, an inflatable capsule which creates a personal space around the user. This space will be fully controllable and synchronized with the user. Embedded sensors, actuators and controllers adapt the environment according to the users needs. Measured data from the user and the environment will also be stored in a database for later analyzing and visualization purposes. More information and pictures can be found on our company website <www.elospaces.com>.

The product development is split into separate categories: mechanical structure, embedded electronics, user interface and web & cloud services. The mechanical structure of the product is developed mainly in China and at the moment they are focusing on the manufacturability and how to embed the electronics to the structure. The two most promising solutions to integrate the electronics are “pillowcase” of cloth on the product which contains the electronics and the other is new Nano carbon technologies which are available to us through a Finnish university.

Embedded electronics have been prototyped with prototyping boards such as Microduino (tiny Arduino) and Raspberry Pi (PI). The main idea is that there is one control unit for each capsule which collects the sensor data, controls the devices and communicates to the user and database. At the moment there are few separate parts which will be integrated together and into the capsule. These parts consist of lighting, speakers and basic sensors. Temperature, air pressure, moisture and motion sensors are read by the PI and written to the database or to a local storage if the internet connection is missing. Sensor data resolution is between 100 ms to 10 s depending on the sensor type.

For user interface I have developed a multiplatform mobile application with Apache Cordova (which enables the use web technologies such as html5, css3 and JavaScript instead of native languages). The goal of the user interface was to have one remote controller for all the devices with clear controls and visualization of the important data. However, we have decided that our product is best fit with more flexible solutions than a separate mobile application. ZeroUI and other user interfaces are being looked into to find the best way of interacting. The vision is that the environment would adapt automatically to optimize the environment for the user. This can be achieved through the measurements and analyzing the data. So at the moment the user interface is not highly related to this project.

The cloud services prototyping is at the very early state. There is a mongoDB located in the Microsoft azure virtual server and the schema for the JSON is somewhat decided. Otherwise the development has not yet been started, thus it is a good scope for the school software project and can be done separately from the other prototypes. Only connection to the capsule is the database where to sensor and user data are stored.

# Proposal of the software project

The goal of the software project is to develop a working full stack one page web application for mongoDB sensor data visualization. It will consist of the server backend where data queries, user authentication and rending are done and the client frontend where the data is presented in an informative way. As the software development has heavily shifted from low level coding to efficient usage of open sourced code and frameworks, it is ideal to use as much as possible the frameworks and their examples for the development. However, to maximize my own learning and understanding, it may be better to develop some components from the scratch and not to use existing solutions. The items below show the phases and their content for the project.

* First the specification based on research. All of the items depend on each other.
  + Architecture of application
  + Technologies for development: templates, frameworks
  + Features to be developed
* The development based on the specification
  + Feature driven development to keep software working all the time
  + Unit tests and some ideas from test drive development to keep the quality good
  + Setting up the development environment
* Documentation updated constantly according to the progress
* All the materials for the project are stored in GitHub repository.

The first task is to define the user requirements and write the feature descriptions based on them. The features are used to find the suitable tools and platform to build them on. This means researching the modern web technologies and their fit to this project. Some of the frameworks use very different approaches and that will affect the software architecture dramatically. That’s why both of these items should be kept in mind in the beginning.

After the features and structure are clear, it is best to use some effort to have all the necessary tools for the developing environments. By this I mean tools such as automated build scripts, unit tests, debuggers and other tools. Some of the frameworks offer these tools as the part of the solution, but many of them require 3rd party solutions. This might sound obvious but I have learned that it is a waste of time to develop anything without proper tools.

After everything is set up, the application is developed feature by feature. This should keep every iteration of the application working and it is easier to keep the focus this way. After each step/item is done, this document is updated and in the end returned to the school for grading together with the developed application.

# Specification

## Features

As the purpose of the application is data visualization, a simple dashboard view will be used to show all the relevant data. An example of dashboard view is shown in figure 1.



Figure Example of a dashboard view (<http://www.cyfe.com/images/dashboard-startup.png>)

In the project application the data will consist of user and sensor data, both connected to a unique user. The data is loaded from a database. The key features are listed below.

* User authentication
  + Login with user name and password
  + Creating a new user (Additional)
* Connection to database
  + Loading data according to the user logged in
* User data visualization
  + Basic information of user
  + Averages and key figures
* Sensor data visualization
  + A few different styles of graphs depending on signal (bars, line, etc…)
  + Tabs where you can choose the signal to be shown
  + A zoom functionality (additional)
  + Live data feed (additional)
* User settings (additional)
  + Customization options for charts and figures
  + Saves are changed to the database

## Architecture and Frameworks

As discussed earlier, the modern web development is heavily based on third party components such as frameworks, plugins and templates. In this chapter the selected components are explained together with the architecture of the application.