INTHER SOFTWARE DEVELOPMENT

ISD IoT Automation

Software Requirements Specification

Version 0.1

Serghei Rusu

1-Feb-16

Software Requirements Specification for ISD IoT Automation project to be implemented during the internship session in February 2016.

Contents

1. Introduction

1.1 Purpose

This document describes general Software Requirements Specification for the ISD IoT Automation project that is to be implemented during the Internship Session in February 2016. The document will serve as a requirements base for the project team. Furthermore requirements and details will be identified, discussed and agreed on in an Agile development method.

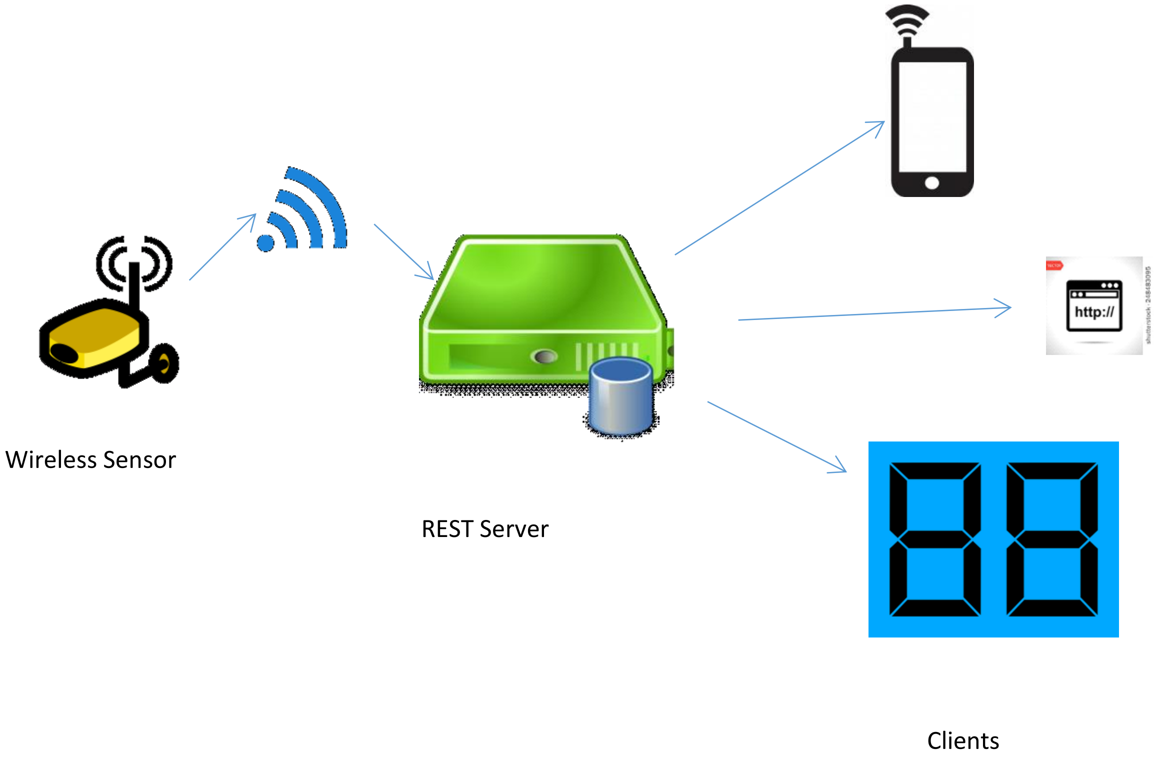
1.2 General Description

Due to evolution of technologies in the IT domain, a new direction of automation and daily life improvement using modern technologies has emerged. The Internet of Things (IoT) is the network of physical objects, devices, vehicles, buildings and other items which are embedded with electronics, software, sensors, and network connectivity, which enables these objects to collect and exchange data.

The ISD IoT Automation project represents a small R&D project, having the main scope to investigate the opportunities in this new domain. As a result the project team should deliver a fully functional product, which can represent a good overview of the IoT opportunities.

The project itself represents a distributed system that has as main scope to remotely detect presence of people in a room. In short, a small server will collect data over network from a wireless autonomous light, movement and presence sensor installed in a room. The server will provide a Web Page and a RESTfull interface for different clients to remotely view the sensors state.

General topology of the product is shown bellow:



2. Specific Requirements

2.1 Functional Requirements

As it results from the short project description, the product that needs to be implemented can be split into several parts.

2.1.1 Sensor

First thing to implement is the remote autonomous sensor. The sensor is based on an Arduino board, that has connected a digital light and PIR sensor (detects movement). The board should reflect sensor state using a set of LEDs, in order to easily see if the sensor is operational.

The sensor firmware and communication protocol should ensure minimal power usage, so the sensor can use a battery pack as a power supply unit.

Wireless communication with the server is implemented using an ESP2866 TCP/IP Wi-Fi board. The actual implementation of the communication protocol will be defined by the project team. The protocol should be easy extendable and documented.

2.1.2 Server

The Arduino will send the state of the sensors to a remote server. The remote server represents a Raspberry PI micro PC that is connected in one LAN with the Arduino.

The server should receive data from the Arduino and store it locally. Data is to be displayed on the LCD display as a first step.

Next step is to run a Web Server on the Raspberry PI that can be accessed from the LAN and that displays the state of the remote sensors. The served web page should have a clean and responsive design.

The server can as well store historical data and display it as statistics.

2.1.3 Clients

Beside the standard HTTP web interface, in order to ease the usage of the product, the server should support connection of different clients via a REST interface. The REST interface API is to be defined by the project team.

Nice to have client applications are:

- Microsoft Windows desktop tray application that displays if there is anybody in the room or not.
- Microsoft Windows desktop widget.
- Android application.
- Android Widget.
- IoT Cloud (check Xively and others).

2.2 Non Functional Requirements

2.2.1 Project Methodology

The project itself will be implemented using Agile SCRUM. There will be 4 Sprints, one week long each. The development team size is limited to 4 people. Specific detailed project requirements will be refined for each sprint apart, in dependency of the actual project progress.

2.2.2 Source Code

Delivered source code should be of high quality, easy maintainable, tested and documented. All the source code will be open source.

Project should be versioned using Git and hosted on GitHub. High quality documentation is a must, both as properly commented code and as a proper documentation document delivered with the project.

Documentation should contain technical knowledge, for future developers, but to not forget end user documentation as well.

2.2.3 Software Tools

The project team is to decide on the tools to be used during the development process. The only requirement is to use freeware user friendly tools. All the used tools should be mentioned in the documentation. All the tools should fully operational on Microsoft Windows.

2.2.4 Hardware

Hardware is basically limited to what the company has already in stock, defined by the list below:

- **Raspberry PI** – to be used as the server hosting solution. The microcomputer is equipped with a WiFi dongle, MicroSD card, power adapter and LCD display.
- **LCD Display** – LCD Display for Raspberry PI.
- **Arduino** – base controller board for the sensor side, equipped with power adapter
- **Light sensor (digital)** - used by the Arduino to detect light.
- **PIR sensor (digital)** – used by the Arduino to detect presence/movement.
- **ESP2866** – Wi-Fi interface for the Arduino board.
- **5V <-> 3.3 V DC-DC converter**
- **5V/3.3V DC power supply**
- **Breadboard**
- **Additional assets** – Digital multimeter, connecting cables, DuPont connectors, different analog components (LEDs, resistors, capacitors), other tools.

Future needed hardware components can be acquired based on team request.