# **[homeStruction]**

Feature Team: Ács Dávid, Albert Erika-Timea, Nagy Lilla, Szabó Bálint

Release date: 23 November 2015

# **Problem**

Home automation is a rapidly growing branch in technology, one that allows people to further customize the place they live in. The main idea of our project would be to automate frequently performed tasks all around the house which could be monotone or even time consuming. It also provides safety and comfort. Controlling the lamps and monitoring temperature remotely are some of the most simple yet effective things that can make life easier and more exciting. The user could easily control the whole system from a Web GUI.

# **Solution**

In order to solve this problem, we will put several sensors in a house. The data from the sensors will be logged in a database on a server. The server will send all the data to a webpage, where many details can be viewed and options can be set. The very core of our project will be a Raspberry Pi and an Arduino Uno (or more) that will work together.

The Raspberry Pi will execute the commands from our Web GUI. The Arduino is needed to collect the data from the sensors and send them to the Raspberry Pi, because the Raspberry Pi does not have analog input.

# **Requirements**

· Green – Will be implemented

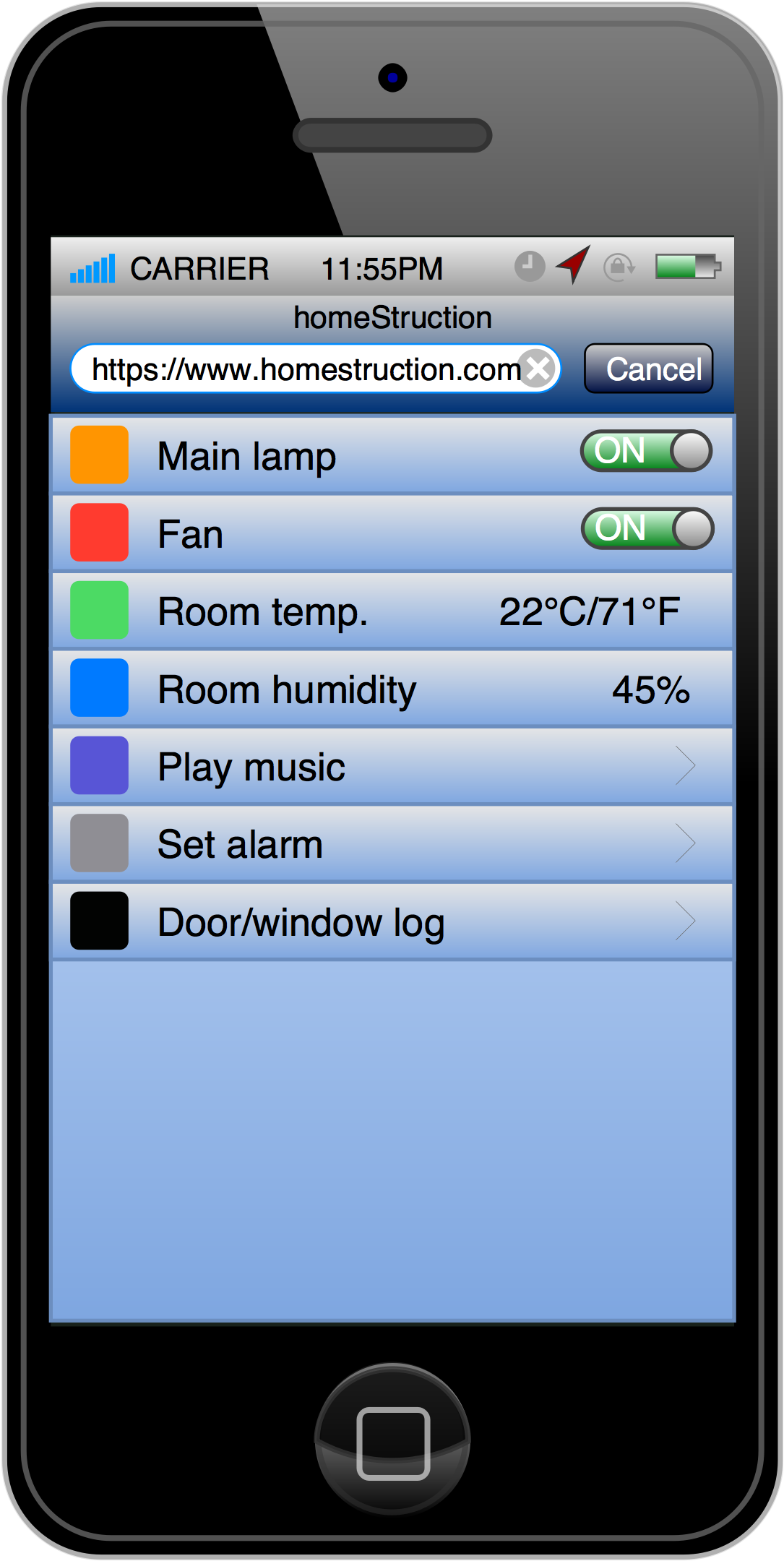
· Yellow – Will be implemented if there’s time

· Red – Won’t be implemented for this release

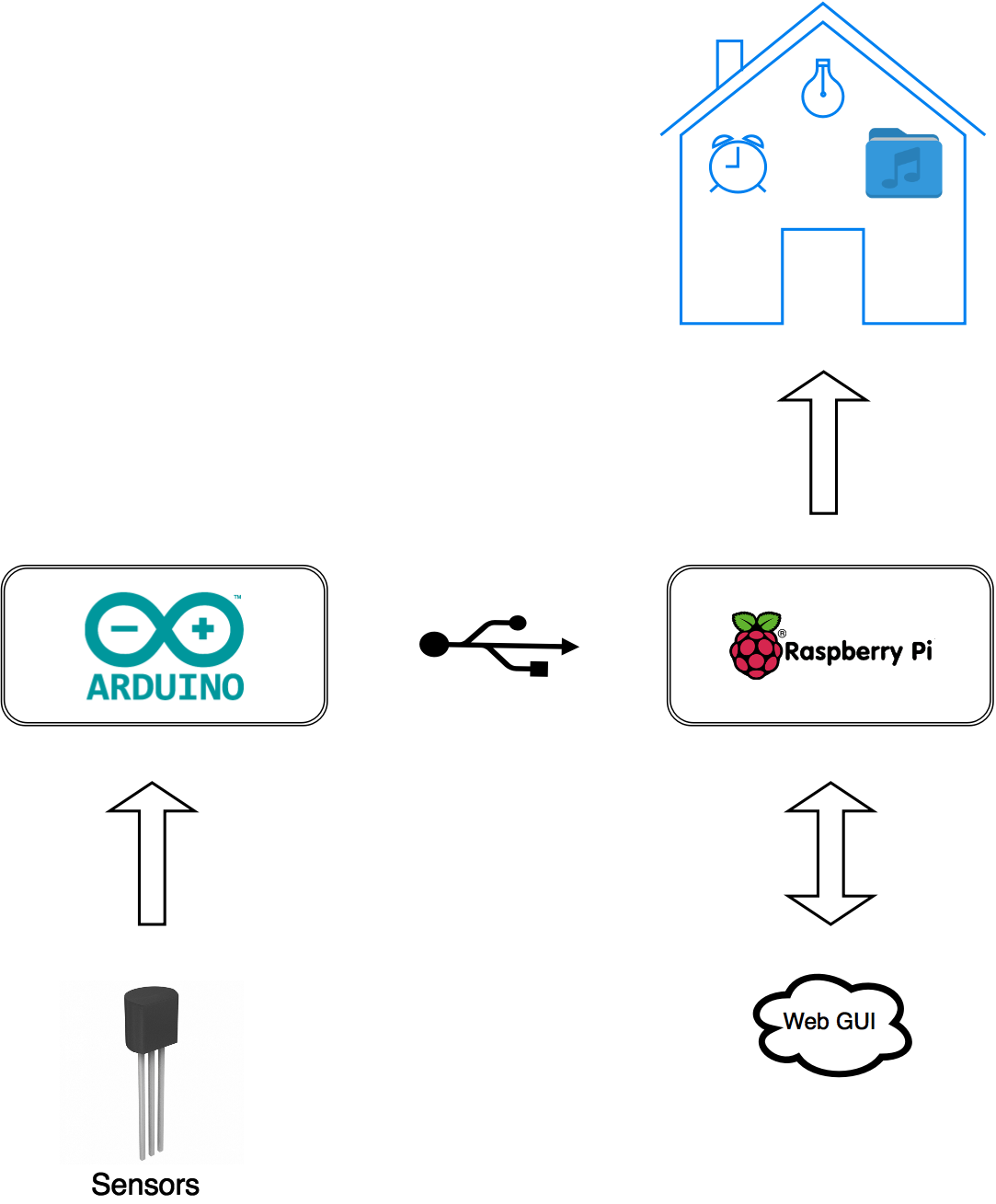
|  |  |  |
| --- | --- | --- |
| **Requirement** | **Priority**  **(G/Y/R)** | **Explanation** |
| Remote control power to different devices | G | Features:  power saving  Allows to switch light on and off remotely. |
| Monitor temperature | G | Logging temperature and humidity in the automated house using sensors. This data will be available in the UI as a graph. |
| Music jukebox | G | Control the music using the website, or using the light sensor (at night, put some relaxing music). |
| Alarm | G | Set an alarm using the webpage and customize it. |
| Make a webpage | G | Create webpage and server in order to provide a UI where the user can monitor and control everything. |
| Database | G | Storing system history for being accessible 100% of the time |
| Fan controller | Y | Based on the room’s temperature, the fan would be turned on/off. |
| Android application | Y | Use an Android application so you can be in control of your room from everywhere. |
| Doors/windows closed/opened log | Y | Have a log of the doors/windows state. |
| Flame detector | Y | Detect a fire and notify the user immediately. |
| Sleep logging | Y | Based on one’s movements, it would log how one sleeps at night. |
| Camera | Y | Check the people who entered the room. |
| Application for all phone OS-s | R | Make a phone application for every OS, from which one can control everything. |
| Turn TV on | R | When you enter the room/get into bed, the TV turns on immediately. |
| Coffee maker | R | As the morning alarm goes off, prepare a coffee. |

# 

# **Detailed description**



Mock up User Interface



The Raspberry Pi will be the center element of the project, it will contain the server, the database, and control the whole system. The user can interact with the system using the Web GUI and get real time data from the system. The analog sensor data will be read by the Arduino and transmitted by USB to the Raspberry Pi.

# **Project phases**

Testing: connection between Raspberry Pi and Arduino, sensors.

Logging data from sensors into a database.

Visualizing data in form of a webpage UI.

Controlling relays (turning on and off devices) using the UI.

# **Testing strategy**

This project offers the possibility of testing each piece of our system even separately at home. First of all we need this system to communicate with our Web GUI, consequently we have to test continuously the connection between the hardware elements (Raspberry Pi/Arduino) and the website developed by us, but also between the hardware elements themselves (Raspberry Pi/Arduino and the sensors). In order to reach the system history, the link between the website and a database is also vital in our case. When controlling the lighting system we need to test carefully the changes we made to it without risking anyone’s safety. The next step in our process would be, that after having everything working well, we are going to set them up as a whole.

# **Software and Hardware**

All of the software we use is free, and we have already bought most of the hardware.

Programming languages:

* Python
* PHP
* HTML
* CSS

Hardware:

* Raspberry Pi ⅔
* Arduino Uno
* Sensors: temperature, humidity, light, flame, sound, magnetic door switch
* Speakers