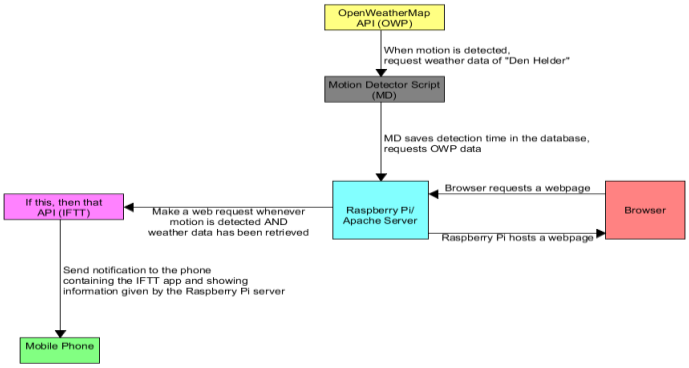
Name Laurent Kleering van Beerenbergh
Student nr 500701043
E-mail Laurent.kleering.van.beerenbergh@hva.nl
Date Friday 14th of January, 2016
Lecturer Bas Pijls

Application - Final Assignment

Application Development

Web application description



The documentation is written is for a web application running on a Raspberry Pi server. The web application serves as a control panel for the Raspberry Pi that the user can control and see various information about the Raspberry Pi.

Within the web application the user is able to shut down and reboot the Pi and he is able to control a motion detector that is attached to the Pi.

Whenever the button has been pressed to activate the motion detector, it will simply start a python script, which will sent timestamps of detections to the database and send a request for the current weather with an API.
And lastly, the script makes a web request to IFTT to send a notification containing the weather information to connected phones with the IFTT application installed.

On the web application itself the user can see a table and graph about when motion has been detected. You can also see the frequency of motion detections depending on the day's motion has been detected.

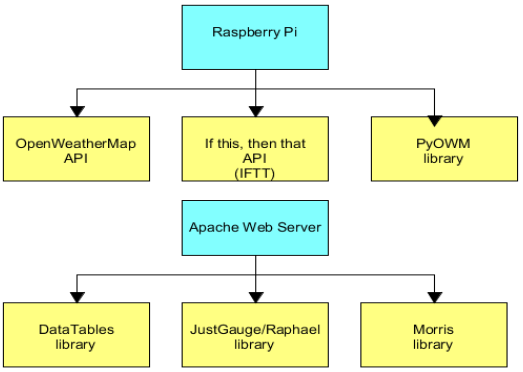
The whole is meant to be run on a Raspberry Pi 3 with Apache, MySQL installed with administrator (sudo) rights.

IOT Theme

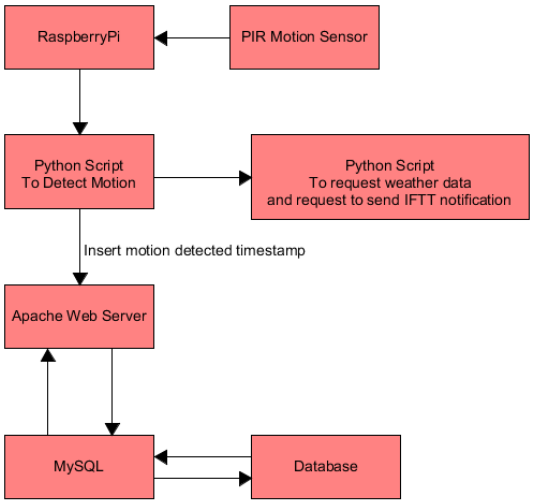
For the project we had to integrate IoT in it. The IOT part of the project is that the web server for the project is installed on the Raspberry Pi with apache and PHP.
The Raspberry Pi also has a motion detection sensor attached to it that can be enabled to detect motion on the click of a button on the webpage.

Libraries and API's

For this project libraries and an API were used. These libraries and API are listed in the picture.



Server-side implementation



On the Raspberry Pi we have the server installed where we can retrieve the webpage from. This server has MySQL installed so we can save data in a database. PHPMyAdmin is installed so we can easily create database and tables and insert, delete and alter records.

The database is being used to insert a timestamp when motion was detected, which will then be used to show in a table and a graph using AJAX and a JavaScript library to visualize the data.

Another thing that will be done on the server side is that a request will be made for the current weather in Den Helder with the OpenWeatherMap API. In Python these requests will be made and then via IFTT we will make a web-request to send a notification with the weather information to phone via a notification.

Client-side implementation

For the project we had to develop using client-side scripting. Here we used JavaScript in combination with JQuery and AJAX. Also, in the project we implemented localization. This means that the whole page is translatable from English to Dutch and vice versa. This is done with click on a button, which adds a GET parameter in the URL for which language you want webpage to have. This makes the page use another language file other than the default one.

AJAX is also used when clicking on the “Start/Stop Motion Detector”. The AJAX requests for both buttons to make a GET request for a page depending on the function of the button. For example, on the “Stop Motion Detector” button it will simply request to execute a PHP file with the id, 2, of the method to stop the motion detection. Lastly, it changes the active/inactive button text and class on the webpage. This code is written in a separate file called “onclick.js”.

HTML, CSS and Bootstrap

For the web application Bootstrap is used for the front-end. Bootstrap is used, because by using bootstrap the web application will be compatible with any browsers and mobile platform browsers.