SMART MIRROR

By

Md. Neaz Ahsan Chowdhury

ID: CSE 050 06459

Kanchon Sarker

ID: CSE 050 06444

A Project Submitted in Partial Fulfillment of the Requirements for the Degree of Bachelor of Science in Computer Science and Engineering

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

STAMFORD UNIVERSITY BANGLADESH

March 2017

APPROVAL

The Project Report "SMART MIRROR" submitted by Md. Neaz Ahsan Chowdhury, Student ID: CSE 05006459 and Kanchon Sarker, Student ID: CSE 05006444, to the Department of Computer Science and Engineering, Stamford University Bangladesh, has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of Bachelor of Science (Hons) in Computer Science and Engineering and approved as to its style and contents.

……..	……..
(Supervisor)	(Chairman)
Mohammad Manzurul Islam	Dr. Kamruddin Nur
Assistant Professor	Associate Professor
Department of CSE	Department of CSE

DECLARATION

We, hereby, declare that the work presented in this project is the outcome of the investigation performed by us under the supervision of Mohammad Manzurul Islam, Assistant Professor, Department of Computer Science and Engineering, Stamford University Bangladesh. We also declare that no part of this project and thereof has been or is being submitted elsewhere for the award of any degree or Diploma.

Countersigned

……
(Supervisor)

Mohammad Manzurul Islam
Assistant Professor
Department of CSE

…….
(Candidate 1)

Md. Neaz Ahsan Chowdhury
ID: CSE 050 06459

…….
(Candidate 2)

Kanchon Sarker
ID:CSE050 06444

ABSTRACT

Smart Mirror project has been developed within the context of a time where every day we see more and more connected devices. The Internet transformed our lives by connecting us more easily to information and other people in the virtual world. Mobile phones became Smart phones and since then this concept has erupted and morphed into the Internet of Things, things which connect us to everyday objects. There is no end of objects that could be made smarter, some being more suited to this than others. Mirrors for example, provide a large surface ideal for displaying information to interacting with. Most people have mirrors at home. So the concept of a smart mirror that we can interact with is attractive and has been fantasized in many futuristic movies. Smart mirrors have recently started to be developed by people in the Maker community, with varying degrees of interactivity. However, so far, the features of these mirrors have been limited. This final year project describes how a smart mirror was built from scratch using a Raspberry Pi 3 microprocessor for the hardware and custom software built on top of Raspbian, a Linux distribution. The goal of the project is to create a low cost Smart Mirror device that people could interact with but also to further develop the technology so that it would let you install and develop your own applications for it. Our Smart Mirror is developed in six months, starting with the software and finally integrating it with the hardware. A few problems arose in the physical construction and software side of the project, such as the glass not being reflective enough but these drawbacks can be addressed by doing more tests and trials to further develop the Smart Mirror.

ACKNOWLEDGEMENTS

First and foremost, we are grateful to Allah, the Almighty, the Merciful without whose blessing, this project would not have been successful. He gave us confidence, courage and determination to overcome the obstacles we faced during this journey.

We would like to express our eternal gratitude to Mohammad Manzurul Islam, Assistant Professor, our respected supervisor, for taking the time from his hectic schedule to guide us in this project. He is the one who inspired us to take this project and supported us every step of the way in every possible way. Without him, this project would not have come to fruition. He provided with valuable technical advice and pointed us in the right directions which were much required for a project like this. We are truly grateful for having a teacher like him as our supervisor.

We would also like to thank our parents and our best friend for pushing and encouraging us to do the best in our ability.

TABLE OF CONTENTS

ABSTRACT ……..... iv

ACKNOLEDGEMENT …… v

TABLE OF CONTENTS ……..... vi

LIST OF FIGURES …….. x

Chapter 1 : Introduction 1

1.1 Introduction 2

1.2 Objectiv 2

Chapter 2 : Literature Review 3

2.1 Raspberry Pi 3 Model B 4

2.1.1 Raspberry Pi 3 Model B Hardware 5

2.1.1.1 Hardware of Raspberry Pi Required Accessories 5

2.1.1.2 BCM2837 6

2.1.1.3 Power 6

2.1.1.4 USB 7

2.1.1.5 GPIO pins 8

2.1.2 Raspberry Pi 3 Model B Operating System Setup 10

2.1.2.1 Raspbian 10

2.1.2.2 NOOBS 10

2.2 Remote Access 12

2.2.1 Access over Internet 12
2.2.2 VNC 12
2.2.3 SSH (Secure Shell) 12
2.2.4 FTP 13
2.2.5 SFTP 13
2.2.6 SCP (Secure Copy) 13
2.2.7 RSYNC 13
2.2.8 Web Server 14
2.3 Apache Web Server 14
2.4 NODE JS 14
2.5 Electron 15
2.6 Python 15
2.7 HTML 15
2.8 CSS 16
2.9 Java Scripts 16
2.10 PHP 16
2.11 Smart Mirror 17
2.11.1 Existing System 17
2.11.2 How Smart Mirror Works 17
2.11.2.1 Monitor 17
2.11.2.2 One way mirror 17
2.11.2.3 Wood frame 18

Chapter 3 : System Development **20**

3.1 Hardware Design 21

3.1.1 Level 0 Design of Smart Mirror 21

3.1.2 Level 1 Design of Smart Mirror 21

3.1.3 Devices Interfacing with Computer 22

3.1.4 Making Wood Frame 23

3.1.5 Assembly 24

3.2 Software Design 25

3.2.1 Program Flow 25

3.2.2 Installation of Node js 25

3.2.3 Installation of Electron 26

3.2.4 Calendar Module 26

3.2.5 Clock Module 27

3.2.6 News Feed Module 27

3.2.7 Current Weather Module 28

3.2.8 Weather Forecast Module 28

3.2.9 Compliments Module 39

Chapter 4 : Project Simulation **30**

4.1 Hardware 31

4.2 Software 31

Chapter 5: Challenges and Future Improvement **33**

5.1 Challenges Faced 34

5.2 Future Improvement 35

Chapter 6: Conclusion **36**

6.1 Conclusion 37

Appendix **38**

Reference **82**

LIST OF FIGURES

Chapter 2 : Literature Review

2.1 Raspberry pi 3 pin diagram 09

2.2 NOOBS installation 10

2.3 Schematic diagram of light reflection on a one way mirror 18

2.4 Wood Frame 19

Chapter 3 : System Development

3.1 Level 0 Design of Smart Mirror 21

3.2 Level 1 Design of Smart Mirror 21

3.3 Diagram of Devices Interfacing with Embedded Computer 22

3.4 Making Wood Frame 23

3.5 Assembly (One way mirror) 24

3.6 Assembly (Monitor) 24

3.7 Program Flow 25

Chapter 4 : Project Simulation

4.1 Normal View of Smart Mirror 31

4.2 View of Smart Mirror (When the room is dark) 32

4.3 View of Smart Mirror (When the room is bright) 32

CHAPTER 1

INTRODUCTION

1.1 Introduction

The Internet of Things (IoT) is transforming every corner of life, the home, the office, city streets and beyond. IoT products give us greater control over door locks, lights and appliances. They also offer insights into resource consumption habit, streamline business processes and better connect us to the people, systems and environments that shape our daily lives.

In this project, We have put an effort to build an IoT device named smart mirror which is basically a one-way mirror (like we might have seen in Hollywood depictions of interrogation rooms), made “smart” by a simple LCD display which sits behind the mirror and displays white UI elements with a black background. When the display is on, we can see both our reflection and the white elements, allowing software to present relevant information while you get ready for the day.

1.2 Objective

The objective of this project is to design and prototype a device that acted as a “Smart Mirror” by displaying the user’s image and providing customizable information on the display. A “Smart Mirror” is a device that acts as a traditional mirror while also superimposing informational data, which can be customized by the user. The mirror also allows for touch free user interaction with some of the data displays. We are able to create a profile and customize the visual interface to display what specific data feeds we want.

CHAPTER 2

LITERATURE REVIEW

2.1 Raspberry Pi 3 Model B

The Raspberry Pi is a very small and low-cost computer which has the size of a credit card and can be plugged into any computer monitor or TV. It uses a standard keyboard and mouse. It has all the capability of a desktop like browsing the internet, playing high-definition video, making spreadsheets, word processing and playing games. What is unique to Raspberry Pi is that it allows the user to interact with the outside world and is currently used in many digital projects. Users can learn to write programs by using languages like Scratch and Python.

The Multipurpose Surveillance Robot has utilized the Raspberry Pi 3 Model B which is the second generation Raspberry Pi which replaced the original Raspberry Pi 3 Model B+ [1]. It has:

- A 1.2GHz 64-bit quad-core ARMv8 CPU
- 4 USB ports
- 40 GPIO pins
- Full HDMI port
- Ethernet port
- Combined 3.5 mm audio jack and composite video
- Camera interface (CSI)
- Display interface (DSI)
- Micro SD card slot
- Video Core IV 3D graphics core

2.1.1 Raspberry Pi 3 Model B Hardware

2.1.1.1 Hardware of Raspberry Pi Required Accessories

- **Micro SD Card**

  An 8 GB class 4 micro SD card with NOOBS (New Out Of the Box Software) pre-installed, is recommended. The minimum recommended capacity of an SD card is 8 GB. 4 GB is recommended for image installation. Even smaller cards can be used for some distributions like OpenElec and Arch. The SD card class determines the write speed a card can sustain. A class 4 SD card can achieve 4 MB/s write speed whereas for a class 10 card 10 MB/s is attainable [33]. But this does not mean a class 10 card will perform better than a class 4 one. Because, in many cases, the higher write speed is achieved at the cost of read speed and increased seek time. The Raspberry Pi 3 Model B requires a micro-SD card.

- **Display and connectivity cable**

  Any HDMI/DVI monitor and any TV works as display for pi. But one with an HDMI input is recommended.

- **Keyboard and mouse**

  Raspberry pi works with any standard keyboard and mouse. Wireless keyboard and mouse will work if already paired with Pi. Keyboard layout can be configured through raspi-config [33].

- **Power supply**

  The Pi is powered by a USB micro power supply like most standard mobile phone chargers. A good-quality power supply is required that can supply at least 700mA at 5V.Power supplies with less than 700 mA current should work for basic usage but the Pi might reboot if too much power is drawn [33].

- **Ethernet (network) cable**

  An Ethernet cable is necessary to connect Pi to a local network and the internet.

- **USB wireless dongle**

  Pi can also be connected to a wireless network through use of a USB wireless dongle which will need to be configured.

- **Audio lead**

  A standard 3.5mm audio jack is used to audio through speakers and headphones. An audio is required in the absence of an HDMI cable. If the Pi is connected to the monitor through an HDMI cable no separate audio lead is necessary as audio can be played directly from the display [33]. But if playing audio through speakers is preferable, it will have to be configured.

2.1.1.2 BCM2837

The Raspberry Pi 3 Model B uses the Broadcom processor BCM2837. The underlying architecture in BCM2837 is identical to BCM2836 [1]. It has an ARMv8 CPU.

2.1.1.3 Power

The device is powered by a 5V micro USB supply. Exactly how much current (mA) the Raspberry Pi requires is dependent on what is connected to it. A 1.2A (1200 mA) power supply from a reputable retailer will provide apple power to run a Pi. Typically, the model B uses between 700-1000mA depending on what peripherals are connected; the model A can use as little as 500mA with no peripherals attached. The maximum power the Raspberry Pi can use is 1 Amp. If someone needs to connect a USB device that will take the power requirements above 1 Amp, then they must connect it to an externally-powered USB hub. The power requirements of the Raspberry Pi increase as it makes use of the various interfaces on the Raspberry Pi. The GPIO pins can draw 50mA safely [1]; distributed across all the pins; an individual GPIO pin can only safely draw 16mA. The HDMI port uses 50mA, the camera module requires 250 mA, and the keyboards and mice can take as little as 100mA or over 1000mA [1]. Backpowering occurs when USB hubs do not provide a diode to stop the hub from powering against the host computer. Some hubs will backfeed the Raspberry Pi. This means that the hubs will power the Raspberry Pi through its USB input cable, without the need for a separate micro-USB power cable, and bypass the voltage protection. If the Raspberry Pi is connected to a hub that backfeeds to it and the hub experiences power surge, the Raspberry Pi could potentially be damaged.

2.1.1.4 USB

The Raspberry Pi 3 Model B is equipped with four USB 2.0 ports. These are connected to LAN9512 combo hub/Ethernet chip IC3, which is itself a USB device connected to the single upstream USB port on BCM2837 [1]. The USB ports enable the attachment of peripherals such as keyboard, mice, webcams that provide the Pi with additional functionality. There are some differences between the USB hardware on the Raspberry Pi and the USB hardware on desktop computers or laptop/tablet devices. The USB host port inside the PI is an On-The-Go (OTG) host as the application processor powering the PI, BCM2836, was originally intended to be used in the mobile market: i.e. as the single USB port on a phone for connection to a PC, or to a single device, In essence, the OTG hardware is simpler than the equivalent hardware on PC [1]. OTG in general supports communication to all types of USB device, but to provide an adequate level of functionality for the most of the USB devices that one might plug into a Pi, the system software has to do more work. In general, every device supported by Linux is possible to use with the Pi, subject to a few caveats detailed further down. Linux has probably the most comprehensive driver database for legacy hardware of any operating system. The OTG hardware on Raspberry Pi has a similar level of support for certain devices, which may present a higher software processing overhead. The Raspberry Pi also has only one root USB port: all traffic from all connected devices is funneled down this bus, which operates at a maximum speed of 480mbps. The OTG hardware on Raspberry Pi has a simpler level of support for certain devices, which may present a higher software processing overhead. The Raspberry Pi also has only one root USB port: all traffic from all connected devices is funneled down this bus, which operates at a maximum speed of 480mbps. The USB specification defines three device speeds - Low, Full and High. Most mice and keyboards are Low-speed, most USB sound devices are Full-speed and most video devices (webcams or video capture) are High-speed. Generally, there are no issues with connecting multiple High-speed USB devices to a Pi. The software overhead incurred when talking to Low- and Full-speed devices means that there are soft limitations on the number of simultaneously active Low- and Full-speed devices. Small numbers of these types of devices connected to a Pi will cause no issues. USB devices have defined power requirements, in units of 100mA from 100mA to 500mA. The device advertises its own power requirements to the USB host when it is first connected. In theory, the actual power consumed by the device should not exceed its stated requirement. The USB ports on a Raspberry Pi have a design loading of 100mA each - sufficient to drive "low-power" devices such as mice and keyboards. Devices such as WiFi adapters, USB hard drives, USB pen drives all consume much more current and should be powered from an external hub with its own power supply. While it is possible to plug a 500mA device into a Pi and have it work with a sufficiently powerful supply, reliable operation is not guaranteed. In addition, hotplugging high-power devices into the Pi's USB ports may cause a brownout which can cause the Pi to reset.

2.1.1.5 GPIO pins

One powerful feature of the Raspberry Pi is the row of GPIO (General Purpose Input/Output) pins. These pins are a physical interface between the Pi and the outside world. The GPIO pins can be programmed to interact in amazing ways with the real world. Inputs don't have to come from a physical switch; it could be input from a sensor or a signal from another computer or device, for example. The output can also do anything. In addition to the familiar USB, Ethernet and HDMI ports, the Raspberry Pi offers the ability to connect directly to a variety of electronic devices. These include:

- **Digital outputs**: turn lights, motors, or other devices on or off
- **Digital inputs:** read an on or off state from a button, switch, or other sensor
- Communication with chips or modules using low-level protocols: SPI, I²C, or serial UART

Connections are made using GPIO ("General Purpose Input/Output") pins. Unlike USB, etc., these interfaces are not "plug and play" and require care to avoid miswiring. The Raspberry PI GPIOs use 3.3V logic levels, and can be damaged if connected directly to 5V levels (as found in many older digital systems) without level-conversion circuitry [8]. Note that no analogue input or output is available. However, add-on boards such as the Rpi Gertboard provide this capability. The Raspberry Pi Model A+ and B+ boards, and the Pi 3 Model B, have a 40-pin header marked J8, arranged as 2x20 pins. The first 26 pins are the same as P1 on the A/B boards, with the remaining 14 pins providing additional GPIO and ground pins, and an EEPROM ID feature for auto-configuration with add-on "HAT" boards. GPIO voltage levels are 3.3 V and are not 5 V tolerant. There is no over-voltage protection on the board - the intention is that people interested in serious interfacing will use an external board with buffers, level conversion and analog I/O rather than soldering directly onto the main board.

All the GPIO pins can be reconfigured to provide alternate functions, SPI, PWM, I²C and so. At reset only pins GPIO 14 & 15 are assigned to the alternate function UART, these two can be switched back to GPIO to provide a total of 17 GPIO pins. Each of their functions and full details of how to access are detailed in the chipset datasheet.
Each GPIO can interrupt, high/low/rise/fall/change. There is currently no support for GPIO interrupts in the official kernel, however a patch exists, and requiring compilation of modified source tree. The 'Raspbian "wheezy"' version that is currently recommended for starters already includes GPIO interrupts. GPIO input hysteresis (Schmitt trigger) can be on or off, output slew rate can be fast or limited, and source and sink current is configurable from 2 mA up to 16 mA. Note that chipset GPIO pins 0-27 are in the same block and these properties are set per block, not per pin. See GPIO Datasheet Addendum - GPIO Pads Control. Particular attention should be applied to the note regarding SSO

(Simultaneous Switching Outputs): to avoid interference, driving currents should be kept as low as possible. The available alternative functions and their corresponding pins are detailed below. These numbers are in reference to the chipset documentation and may not match the numbers exposed in Linux. Only fully usable functions are detailed, for some alternative functions not all the necessary pins are available for the functionality to be actually used. A Raspberry Pi 3 pin diagram is given below:

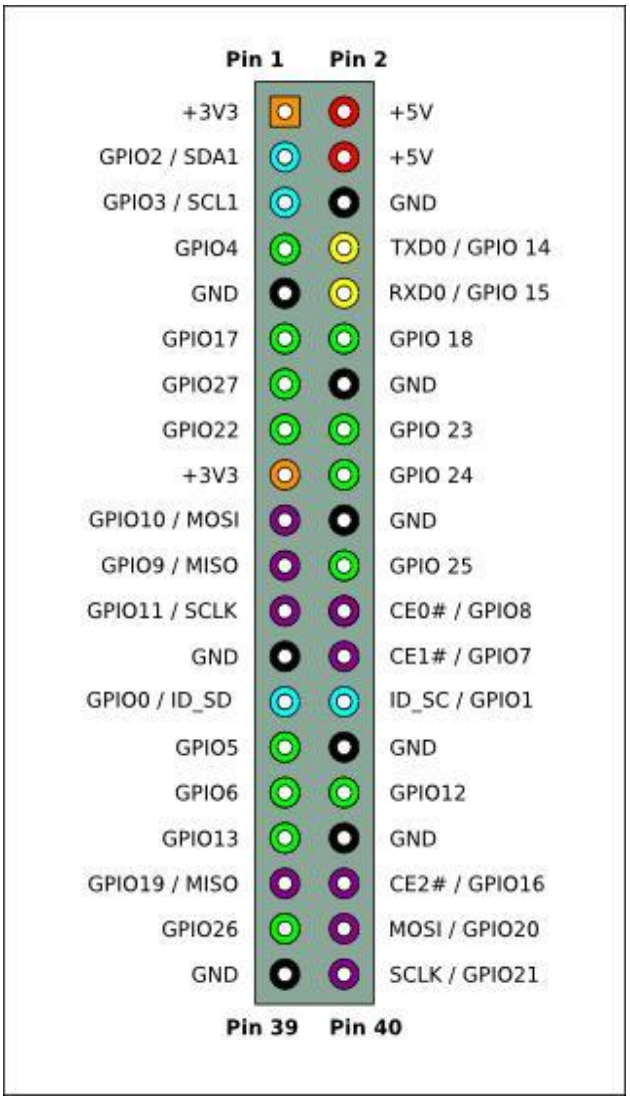

Figure 2.1: Raspberry Pi 3 pin diagram

2.1.2 Raspberry Pi 3 Model B Operating System Setup

There are different ways to install operating system into raspberry pi. The main two systems are discussed below. We use NOOBS to install operating system into raspberry pi.

2.1.2.1 Raspbian

Raspbian is the default operating system for regular use on Raspberry Pi. Raspbian is a free operating system based on Debian and optimized for Raspberry Pi hardware. Raspbian comes with over 35000 packages; precompiled software bundled in a nice format for easy installation on Raspberry Pi. Raspbian is a community project under active development [1], with the emphasis on developing stability and performance of as many Debian packages as possible. (update it. Forcus on both and discuss about the features and benefits. Finally say which method you have used.)

2.1.2.2 NOOBS

New out Of the Box Software is an easy operating system install manager for the Raspberry Pi. The most convenient way to get NOOBS is to purchase a micro SD card with NOOBS pre-installed. It can also be downloaded from the Raspberry Pi website [10]. After downloading the NOOBS zip file, the contents of the file will have to be copied to a formatted SD card on a computer.

Figure 2.2: NOOBS installation

- **Installing NOOBS on an SD card**

    - Format an SD that is 4 GB or larger as FAT. Download and extract files from the NOOBS zip file.

    - Copy the extracted file onto the SD card immediately after formatting so that the files are in the root directory of the SD card.

    - The files might be copied into a single folder. In that case, the files inside the folder must be copied across rather than the folder itself.

    - The ‘’RECOVERY” fat partition will be automatically resized to a minimum and a list of available to install OSs will be displayed.

- **Format an SD card as FAT**

    - **Windows:** The SD Association’s Formatting Tool is recommended for Windows users which can be downloaded from sdcard.org. It is necessary to set “FORMAT SIZE ADJUSTMENT” option “ON” in the “Options” menu so that the entire SD card volume is formatted instead of a single partition.

    - **Mac OS:** The SD Association’s Formatting Tool is also available for Mac users. The default OSX Disc utility can also be used to format the disc. In order to do that, the SD card has to be selected and choose Erase with MS-DOS format.

    - **Linux:** For Linux user’s parted (or the command line version parted) is recommended.

- **Included in NOOBS**

    Only Raspbian is installed in NOOBS by default. The others can be installed with a network connection.

- **NOOBS and NOOBS Lite**

    NOOBS comes in two forms: offline and network install; or network install only. The full version comes with Raspbian included and so it can be installed from the SD card directly without any internet connection. The NOOBS Lite or any other operating system needs to be installed while online [10]. The operating system image on the full version might get outdated. So, if an internet connection is available during installation, the user will be given the option to choose to download the new version if the current one is outdated. An elaborate documentation of NOOBS and NOOBS source code is available on GitHub.

2.2 Remote Access

2.2.1 Access over internet

Raspberry Pi can be connected to another computer or a mobile device. One method is to set-up port forwarding. To set-up port forwarding one must change the configuration of their router to forward all inbound traffic from the internet to a specific port to the local IP address of their Raspberry Pi. One disadvantage of doing this is that it exposes a network port on a private LAN to the public network [34]. This is security vulnerability and must be managed carefully. One secure alternative to port forwarding is the Weaved service. Weaved is software that needs to be install on the Raspberry Pi and it will allow Pi to connect to any device over the internet.

2.2.2 VNC

VNC is a graphical desktop sharing system that allows someone to remotely control the desktop interface of one computer from another. It transmits the keyboard and mouse events form the controller, and receive updates to the screen over the network from the remote host. This enables a user to use the desktop of the Pi inside a window on their computer. They will be able to control it as though they were working on Raspberry Pi itself [35]. VNC is used widely across every industry sector by individuals and organizations for different use cases which include providing IT desktop support to colleagues and friends, and also accessing system and services on the move.

2.2.3 SSH (Secure Shell)

Secure Shell, or SSH is a network protocol which operates at Application layer of OSI model to allow encrypted remote login and other services for secure operations over an unsecured network. SSH provides a secure channel in a unsecure network through a client-server architecture. SSH is capable of securing any network service but typical applications include remote command-line login and remote command execution [29]. The command line of Raspberry Pi can be accessed from another computer in the same network using SSH. SSH server is enabled on Raspberry Pi by default. It can be disabled as well. SSH is built into Linux distributions and Mac OS, and a third-party SSH client is available for Windows.

2.2.4 FTP

The File Transfer Protocol is a standard network protocol to transfer files between a client computer and a host computer. It is based on client-server architecture and utilizes separate connections for control and data between client and server [38]. FTP clients can authenticate themselves with clear-text sign-in usually in the form of username and password but they can also sign in anonymously if the server is configured to do so. Sometime FTP is secured with SSL/TLS (FTPS) in order to protect username and password and encrypt the content. FTP can be used to transfer files between a Raspberry Pi and another computer. Although with default program sftp-server of Raspbian, the users with sufficient privilege can transfer files or directories. And access to the file system of the limited users is also required often.

2.2.5 SFTP

The FTP is a popular file transfer protocol for transferring file between two remote systems. SFTP, which stands for SSH File Transfer Protocol, is a separate protocol packages with SSH that works in a similar way over a secure connection. The advantage is it leverages a secure connection to transfer file and traverse file system on both local and remote system. Almost in all cases, SFTP is preferable to FTP because of its underlying security [36]. The SSH File Transfer Protocol is a network protocol that provides file access, file transfer, and file management functionalities over SSH. By using SFTP one can easily change, browse and edit files on their Raspberry Pi. SFTP is easier to setup that FTP as Raspbian has SSH enabled by default.

2.2.6 SCP (Secure Copy)

Secure Copy or SCP is a means of securely transferring file between a local host and a remote host or two remote hosts. Its utilizes SSH, SCP is a command for sending files over SSH. This means someone can copy files between computers, such as Raspberry Pi to desktop or laptop, or vice-versa [37]. SCP uses the same mechanism as SSH for authentication which ensures the authenticity and confidentiality of the data in transit. A client can upload files to a server and additionally add their basic attributes like permissions or timestamps. A client can request to download files from the server as well.

2.2.7 RSYNC

There is a tool called rsyn to synchronize folders between computers. Someone might want to transfer some files from their desktop computer or laptop to their Pi, and for them

to be kept up to date, or they might want the pictures taken by Pi transferred to their computer automatically. Using rsync over SSH allows someone to transfer files to their computer automatically [39]. RSYNC is widely-used tool to keep copy of a file in two computers simultaneously. It is commonly used in Unix-like operating system and functions both as a file synchronization and file transfer mechanism. It uses a type of delta algorithm for reducing network usage. Zlib may be used for additional compression and Stunnel or SSH can be used for data security.

2.2.8 Web Server

A web server can be installed on a Raspberry Pi to host a full website locally in a network or globally on the internet. It can also be used to display some information that needs to be shared with other machines in the local network. There are various web servers available, with different advantage for usage like Apache and NGINX [40]. This project uses the Apache web server to access Raspberry Pi over the local network.

2.3 Apache Web Server

The Apache HTTP Server, more often called Apache, is the world's most used web server software. It played a key role in the initial growth of World Wide Web (WWW). It quickly overtook NCSA HTTPd as the dominant HTTP server. It is the most popular web server software since April 1996 [41]. In 2009 Apache became the first web server software to serve more than a hundred million websites. Apache can be installed on the Raspberry Pi to allow it to serve web pages. On its own, Apache can serve HTML files over HTTP, and with additional modules can serve dynamic web pages using scripting languages such as PHP.

2.4 NODE JS

Node.js is an open-source, cross-platform JavaScript runtime environment for developing a diverse variety of server tools and applications. Although Node.js is not a JavaScript framework,[4] many of its basic modules are written in JavaScript, and developers can write new modules in JavaScript. The runtime environment interprets JavaScript using Google's V8 JavaScript engine. Node.js has an event-driven architecture capable of asynchronous I/O. These design choices aim to optimize throughput and scalability in Web applications with many input/output operations, as well as for real-time Web applications (e.g., real-time communication programs and browser games). The Node.js distributed development project, governed by the Node.js Foundation, is facilitated by the

Linux Foundation's Collaborative Projects program Corporate users of Node.js software include GoDaddy, Groupon, IBM, LinkedIn ,Microsoft, Netflix,PayPal, Rakuten, SAP, Voxer, Walmart, Yahoo! and Cisco Systems.

2.5 Electron

Electron (formerly known as Atom Shell) is an open-source framework developed by GitHub.[2] It allows for the development of desktop GUI applications using the Node.js runtime and the Chromium web browser, originally used for the development of backend web applications. Electron is the main framework behind two notable open-source source code editors: GitHub's Atom, and Microsoft's Visual Studio Code. A basic Electron app consists of three files: package.json (metadata), main.js (code) and index.html (graphical user interface). The framework is provided by the Electron executable file (electron.exe in Windows, electron.app on macOS and electron on Linux). Developers wishing to add branding and custom icon can rename and/or edit the Electron executable file.

2.6 Python

Python is widely used high-level, interpreted, dynamic programming language. Its design philosophy emphasizes code readability and its syntax structure allows programmers to express concepts in fewer lines of code that would be possible in languages like C++ or Java. The language contains construct to enable clear programs on both small and large scale. Python is a wonderful and powerful programming language that's easy to use (easy to read and write) and Raspberry Pi lets it connect a project to the real world. Python syntax is very clean, with an emphasis on readability and uses standard English keywords.

2.7 HTML

Hyper Text Markup Language, usually referred to as HTML, is the most common markup tool to make any website. HTML defines the contents of a website [13]. Along with CSS and JavaScript, HTML is a cornerstone technology used to create web pages as well as to create web interfaces for mobile and web applications. Web browsers can read HTML files and render them into visible or audible web pages. HTML describes the structure of a website semantically and, before the advent of CSS, included cues for the representation or appearance of the document or web page which makes it a markup language rather than a programming language [14].

2.8 CSS

CSS is a style sheet language for describing the presentation of a document written in a markup language [15]. Although most often used to set the visual style of web pages and user interfaces written in HTML or XHTML, the language is perfectly capable of being applied to any XML document, including plain XML, SVG and XUL, and is applicable to rendering in speech, or on other media. Along with HTML and JavaScript [16], CSS is a cornerstone technology used by most website to create visually engaging web pages, user interfaces for web applications, and user interfaces for many mobile applications. CSS is capable of maintaining style of multiple web pages at once.

2.9 Java Script

JavaScript extends HTML by defining how a webpage should behave when particular HTML events occur. It is the programming language of the web [17]. It is also a high-level and object-oriented language. It is a high-level, untyped, and interpreted programming language. It has been standardized in the ECMAScript language specification. Alongside HTML and CSS, it is one of the tree core technologies of the World Wide Web content production. The majority of websites enjoy it and it is supported by all modern web browsers without plug-ins. JavaScript has a API for working with text, arrays, dates and regular expressions, but does not include any I/O, such as networking, storage or graphics facilities. It relies for these upon the host environment upon which it is embedded [18].

2.10 PHP

PHP is a very popular and widely-used sever scripting language [19]. It is powerful tool for making dynamic websites. It is free and can be used as general purpose programming language as well. PHP originally stood for Personal Home Page, but now stands for the recursive backronym PHP: Hypertext Preprocessor [20]. PHP code may be embedded into HTML code, or it can be used in combination with various web template systems, web content management system and web frameworks. PHP code is usually processed by a PHP interpreter implemented as a module in the web server or as a Common Gateway Interface (CGI) executable. The web server combines the results of the interpreted and executed PHP code, which may be any type of data, including images, with the generated web pages. PHP code may also be executed with a command-line interface (CLI) and can be used to implement standalone graphical applications [20].

2.11 Smart Mirror

2.11.1 Existing System

The Smart Mirror by Evan Cohen and Michel Mach inspired us to make this project. Evan Cohen Smart Mirror offers mainly software part and offers a wide range of voice commands to interact with the mirror. The voice command support weather, time and date information, adding reminders, showing pictures and even controlling the light in the house.

2.11.2 How Smart Mirror Works

2.11.2.1 Monitor

The size of the mirror is really dictated by the kind of monitor we get. We wanted to get something large as possible but also with a removable arm so it could fit inside of a case. Another important aspect is making sure we get a monitor with the inputs towards the center of the monitor rather than the sides.We was cautious to get a monitor with it's inputs on the bottom or side because it would be hard to make the frame to fit and have the raspberry pi's HDMI cord also fit. We use the Samsung B1930 18.5 inch LCD monitor.

2.11.2.2 One way mirror

This is probably the most important part of the hardware because it's responsible for creating the futuristic effect and is the biggest part of the smart mirror. Wikipedia provides the following definition

A one-way mirror, sometimes called two-way mirror, is a mirror that is partially reflective and partially transparent. When one side of the mirror is brightly lit and the other is dark, it allows viewing from the darkened side but not vice versa (Loy, 1999).

This was the most difficult component to find because of these technical requirements, but we make a one-way mirror customly. The one that was we made unfortunately not very reflective so sometimes we can see the interior of the device. This is not ideal but in the right conditions it works well and it can always be replaced with better quality glass in the future

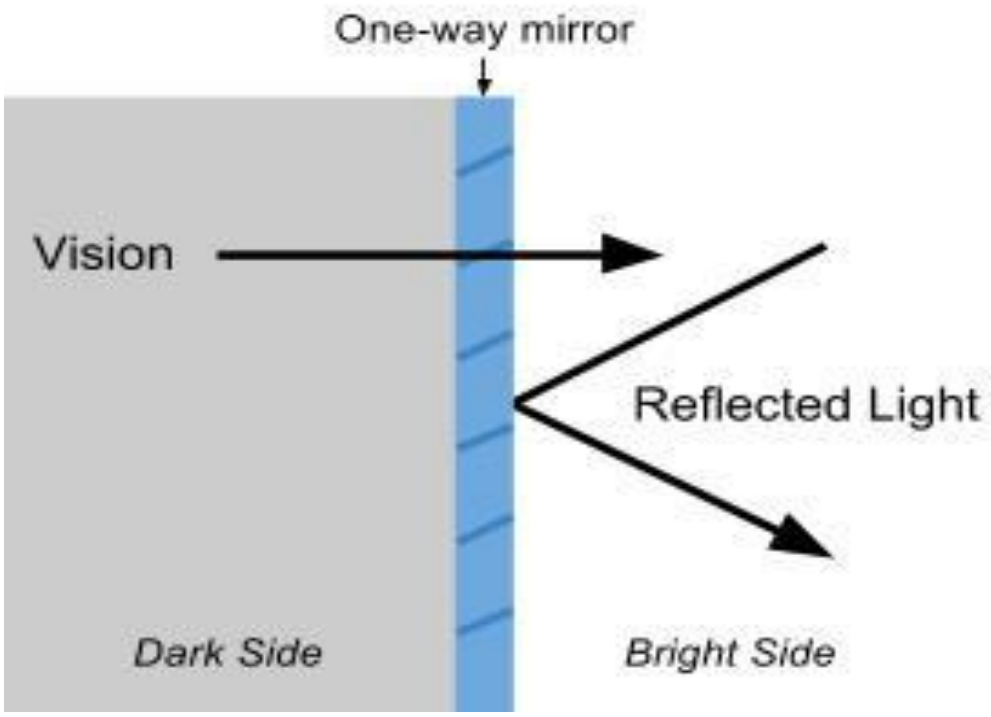

Figure 2.3: Schematic diagram of light reflection on a one-way mirror

In the case of this project this essentially means that the dark or black parts of the screen will be seen as a reflection and the light parts will be seen normally. So if there is white text over a black background the white text will be seen as an overlay with the user reflected in the background.

This was the most difficult component to find because of these technical requirements, but we made a custom made one way mirror. The one that was made unfortunately not very reflective so sometimes we can see the interior of the device. This is not ideal but in the right conditions it works well and it can always be replaced with better quality glass in the future.

2.11.2.3 Wood frame

Wood Frame is another important part of the Smart Mirror. How the whole project look like is depend on the wood frame. We make it reusable, enough scope for adding other sensor for future improvement of this project. How we design the wood frame given in below figure

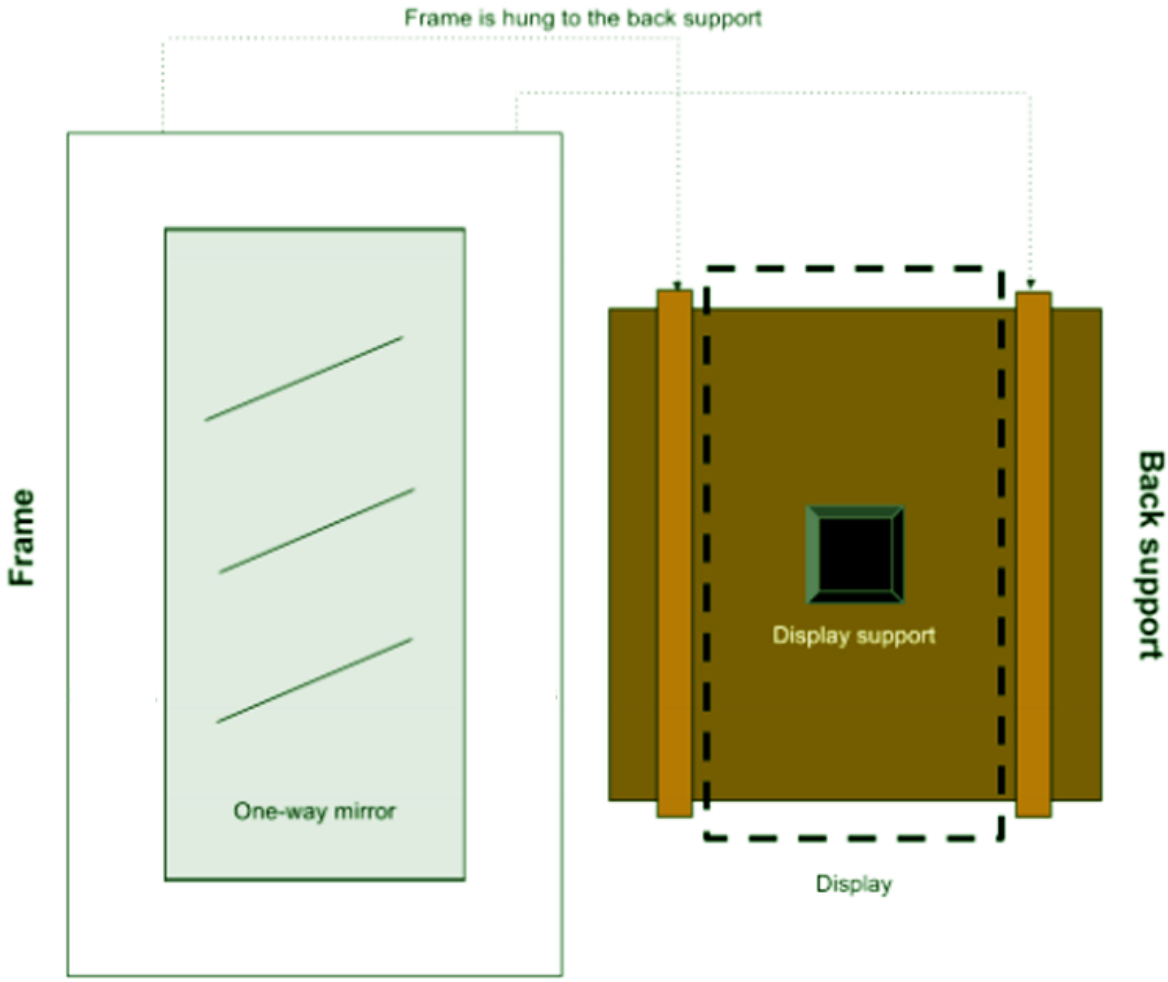

Figure 2.4: Wood Frame

CHAPTER 3

SYSTEM DEVELOPMENT

3.1 Hardware Design

For making smart mirror we make two level design pattern .First we design initial design of the smart mirror. Then we design the first level design pattern of the smart mirror.

3.1.1 Level 0 Design of Smart Mirror

At first we design the level 0 design which is the initial design of the smart mirror. This is the initial idea of making the smart mirror then we expand it in level 1 design.

Figure 3.1: Level 0 Design of Smart Mirror

3.1.2 Level 1 Design of Smart Mirror

After completing level zero level we design the first level design pattern. In first level design pattern we design whole idea and working procedure of the smart mirror.

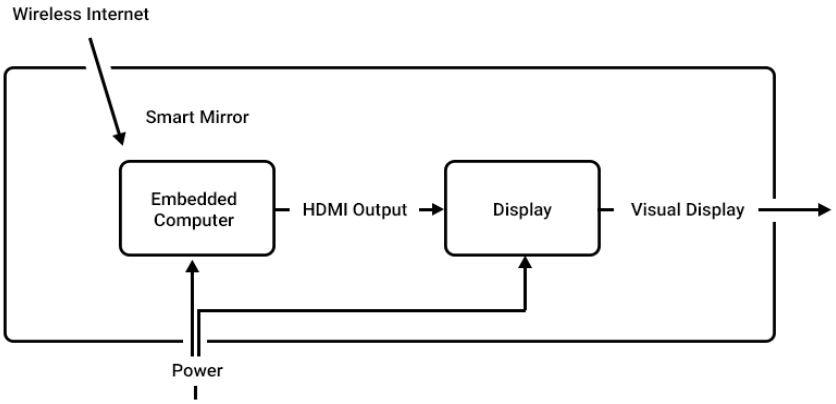

Figure 3.2: Level 1 Design of Smart Mirror

3.1.3 Devices Interfacing with Computer

Based on the diagram, a minimum of two USB ports are required in order to get the Smart Mirror usable for a user without a keyboard and mouse. The mirror's display is connected through HDMI to prevent having to buy an HDMI to VGA/DVI adapter. A high display resolution is also required, meaning graphics processing capabilities are of high importance.

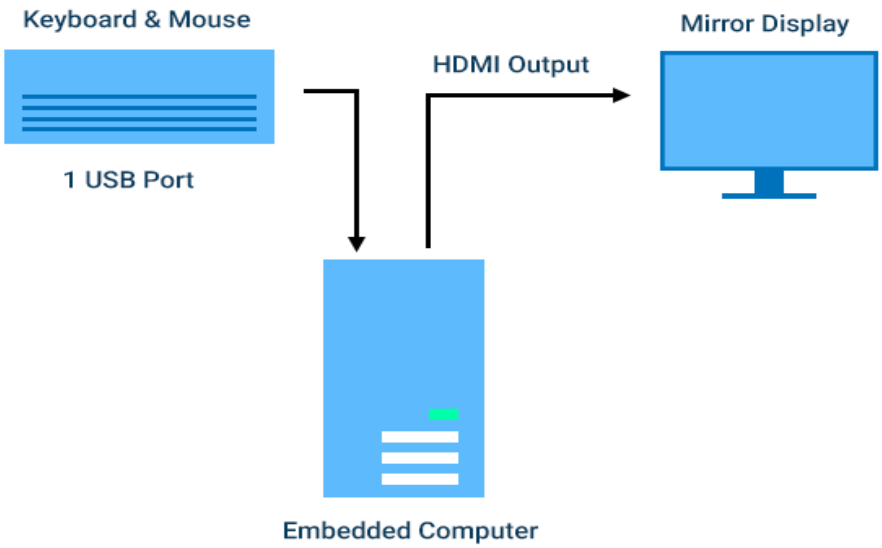

Figure 3.3: Diagram of Devices Interfacing with Embedded Computer

3.1.4 Making Wood Frame

We take the measurements for the frame and start cutting. We used both screws and glue to make sure it was sturdy enough, the final product weighs quite a lot. Put some air holes at the top and a hole for the cord at the bottom.

We also cut the "frame" in the front in an angle, just because we like it. Then we put the front on the frame and we are almost done.

We make 4 small pieces of wood that we later can put behind the screen, to make sure it doesn't fall backwards.

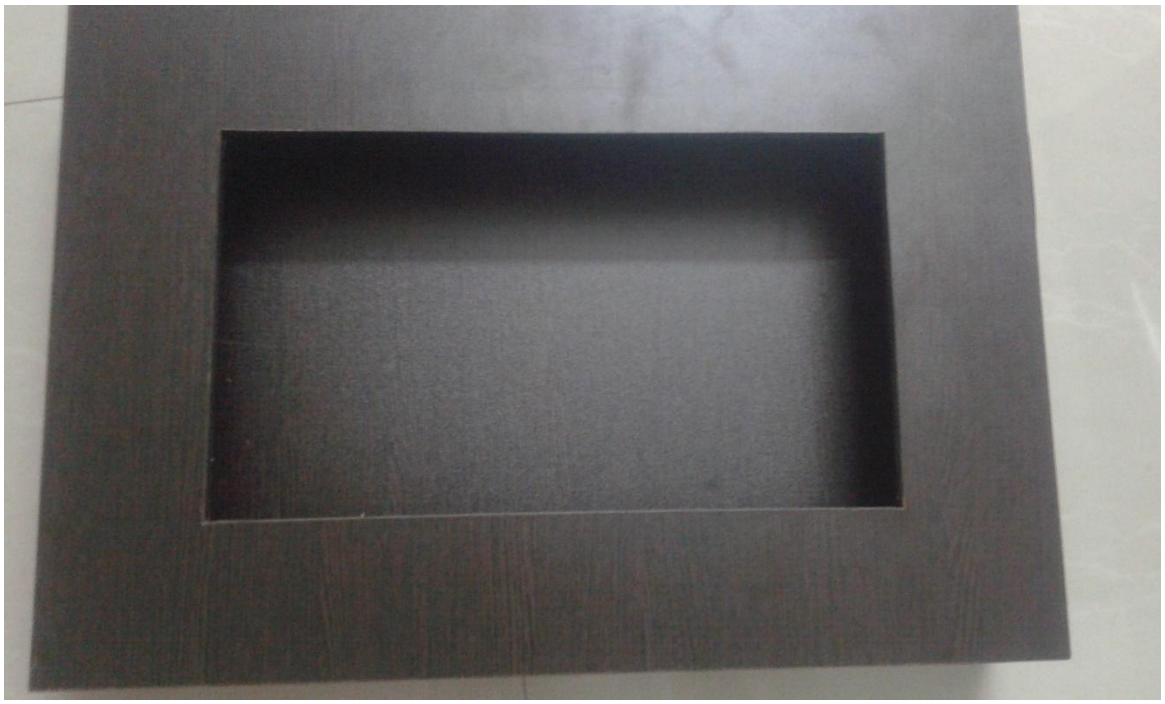

Figure 3.4: Making Wood Frame

3.1.5 Assembly

We take the frame and put the mirror in the front. Place the screen behind it and plug in all the things we need. Start the Raspberry and make sure everything works. Then we turn it off and put the final 4 pieces behind the screen to make it stay up. Then we just plug everything in there and maybe use some cable ties to make it look a little nice.

Figure 3.5: Assembly (One way mirror)

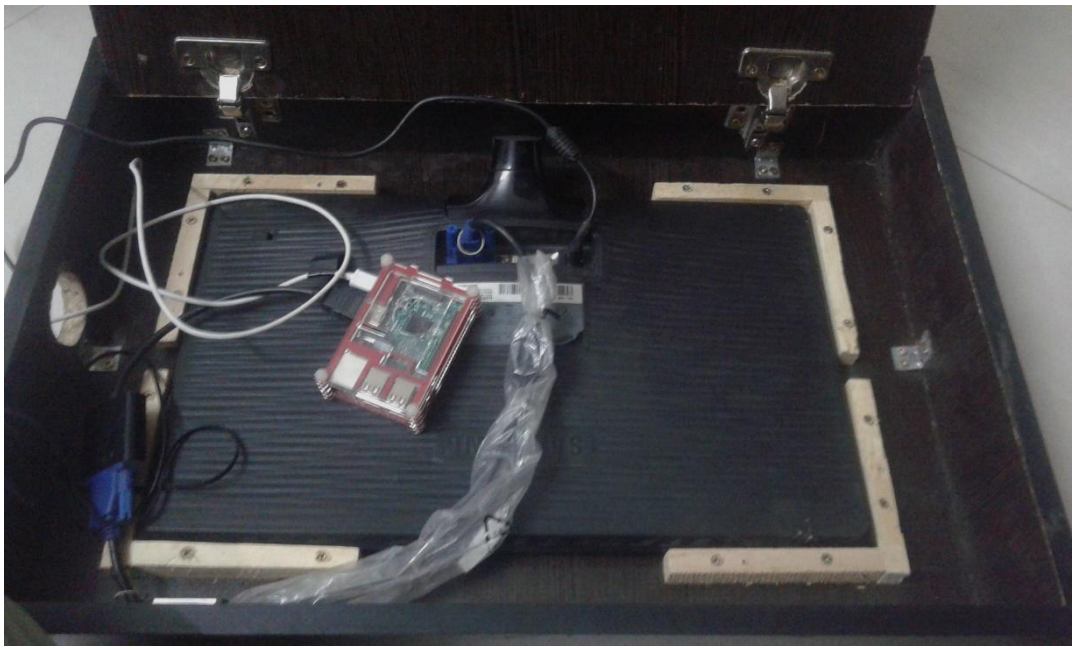

Figure 3.6: Assembly (Monitor)

3.2 Software Design

3.2.1 Program Flow

Our Smart mirror started when the user switched on the raspberry pi. Then device parsed command and translate the API call. In the next step API processed Query and return data to the raspberry pi. After that Raspberry pi received those data and processed into the display widget.

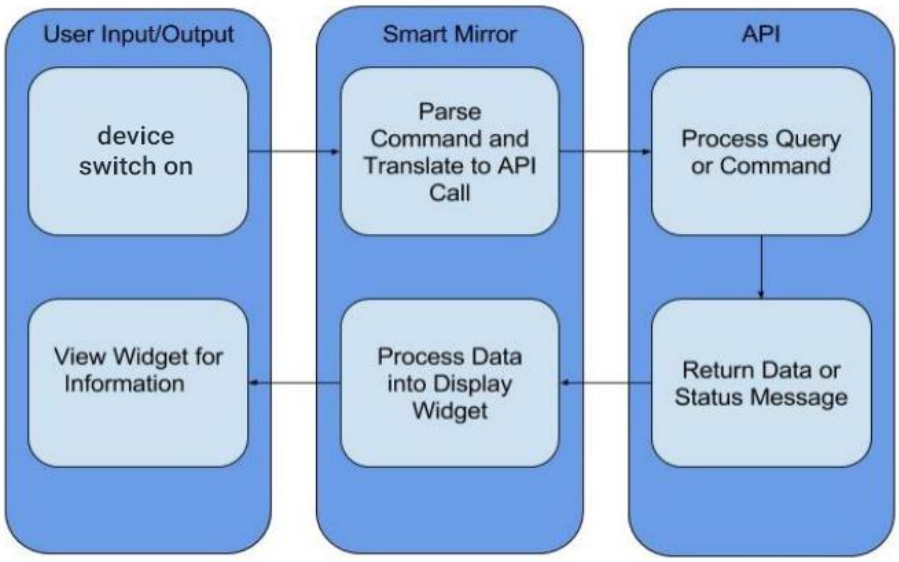

Figure 3.7: Program Flow

3.2.2 Installation of NODE J.S

FOR Installing Node J.S we follow the stops below

```
wget https://nodejs.org/dist/v4.3.2/node-v4.3.2-linux-armv6l.tar.gz

tar -xvf node-v4.3.2-linux-armv6l.tar.gz

cd node-v4.3.2-linux-armv6l
```

Copy to /usr/local

```
sudo cp -R * /usr/local/
```

3.2.3 Installation of Electron

FOR Installing Node J.S we follow the stops below

```
npm install electron --save-dev

npm install electron -g
```

3.2.4 Calendar Module

This module displays events from a public .ical calendar. It can combine multiple calendars.

```
modules: [
    {
        module: 'calendar',
        position: 'top_left',    // This can be any of the regions. Best results in left or right
regions.
        config: {
            // The config property is optional.
                                }
    }
```

3.2.5 Clock Module

This module displays the current date and time. The information will be updated realtime.

```
modules: [
   {
      module: 'clock',
      position: 'top_left',   // This can be any of the regions.
      config: {
         // The config property is optional.
      }
   }
]
```

3.2.6 News Feed Module

This module displays news headlines based on an RSS feed

```
modules: [
   {
      module: 'newsfeed',
      position: 'bottom_bar', // This can be any of the regions. Best results in center
regions.
      config: {
         // The config property is optional.
         // If no config is set, an example calendar is shown.
         // See 'Configuration options' for more information.

         feeds: [
            {
               title: "New York Times",
               url: "http://www.nytimes.com/services/xml/rss/nyt/HomePage.xml",
            },
            {
               title: "BBC",
               url:
"http://feeds.bbci.co.uk/news/video_and_audio/news_front_page/rss.xml?edition=uk",
            },
```

3.2.7 Current Weather Module

This module displays the current weather, including the wind speed, the sunset or sunrise time, the temperature and an icon to display the current conditions.

```
modules: [
    {
        module: 'currentweather',
        position: 'top_right',  // This can be any of the regions.
                                // Best results in left or right regions.
        config: {
            // See 'Configuration options' for more information.
            location: 'Amsterdam,Netherlands',
            locationID: '', //Location ID from http://openweathermap.org/help/city_list.txt
            appid: 'abcde12345abcde12345abcde12345ab' //openweathermap.org API key.
        }
      }
]
```

3.2.8 Weather Forecast Module

This module displays the weather forecast for the coming week, including an an icon to display the current conditions, the minimum temperature and the maximum temperature.

```
modules: [
    {
        module: 'weatherforecast',
        position: 'top_right',  // This can be any of the regions.
                                // Best results in left or right regions.
        config: {
            // See 'Configuration options' for more information.
            location: 'Amsterdam,Netherlands',
            locationID: '', //Location ID from http://openweathermap.org/help/city_list.txt
            appid: 'abcde12345abcde12345abcde12345ab' //openweathermap.org API key.
        }
    }
]
```

3.2.9 Compliments Module

This module displays a random compliment.

```
modules: [
    {
        module: 'compliments',
        position: 'lower_third',    // This can be any of the regions.
                                    // Best results in one of the middle regions like: lower_third
        config: {
            // The config property is optional.

        }
    }
]
```

CHAPTER 4

PROJECT SIMULATION

4.1 Hardware

The hardware looks very good but the glass could be more reflective. In some conditions we can see a bit of the interior of the device

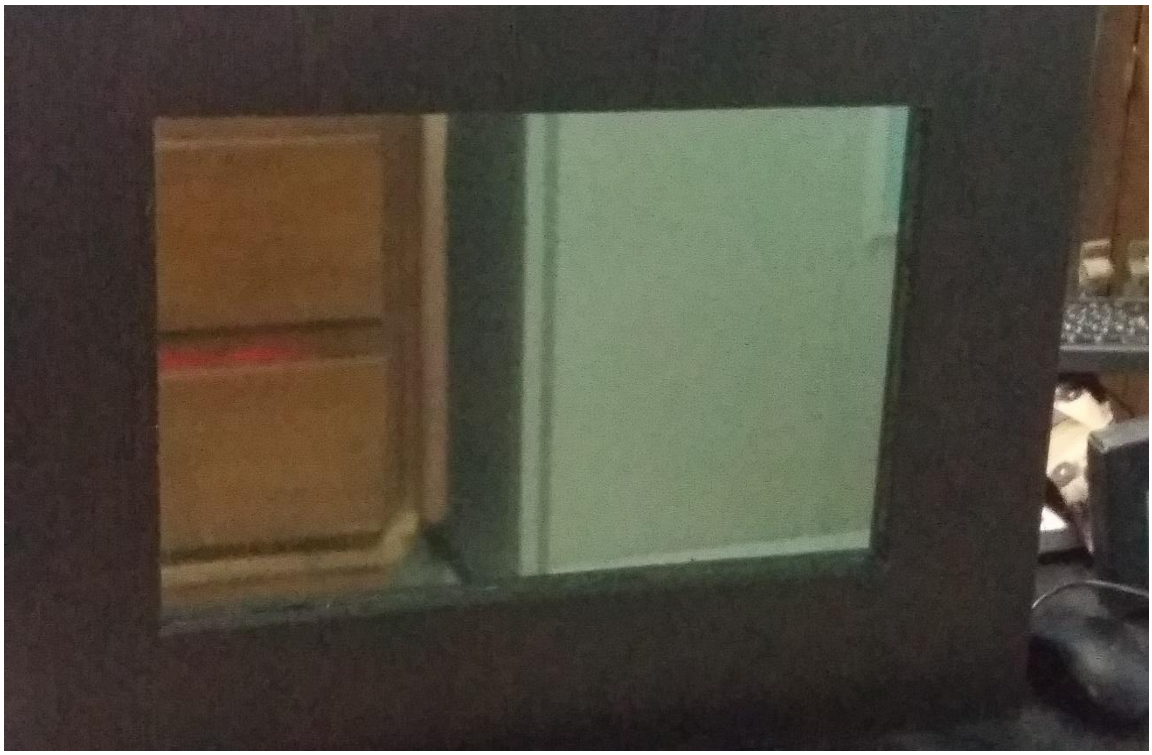

Figure 4.1: Normal view of Smart Mirror

4.2 Software

The webpage consists of clock in top left position, calendar in top left position, compliments lower third position, current weather of Dhaka in top right position, weather forecast of Dhaka in top right position and newsfeed of New York Times in bottom bar.

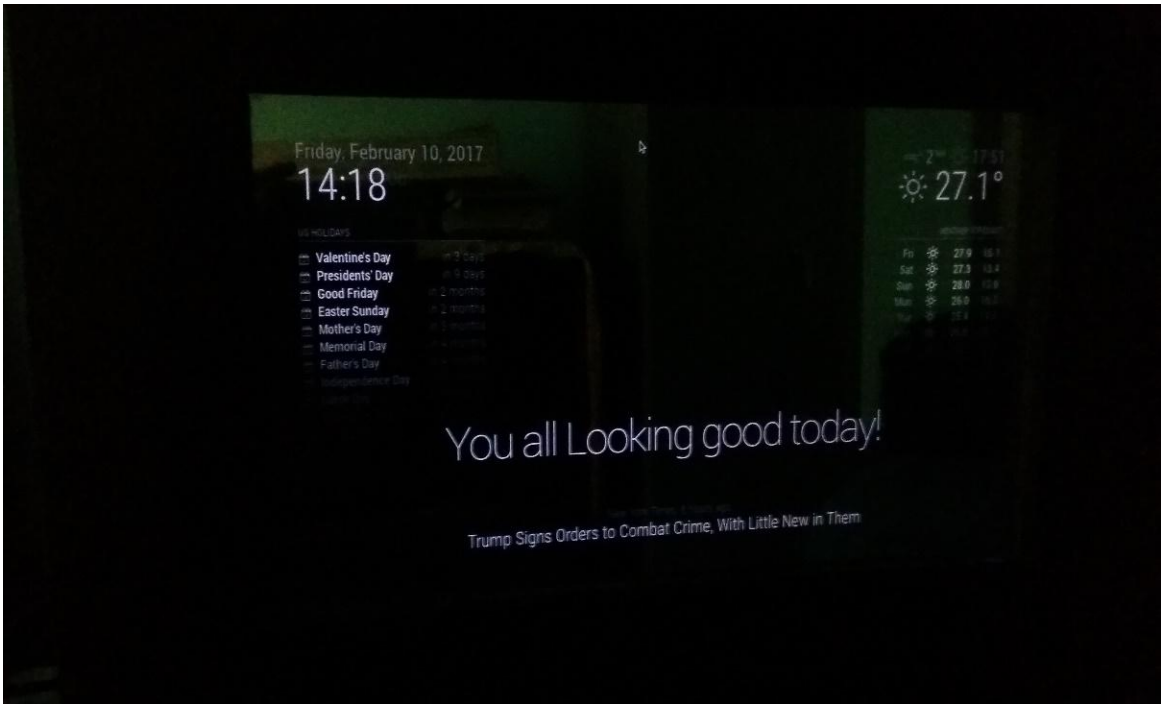

Figure 4.2: View of Smart Mirror (When the room is dark)

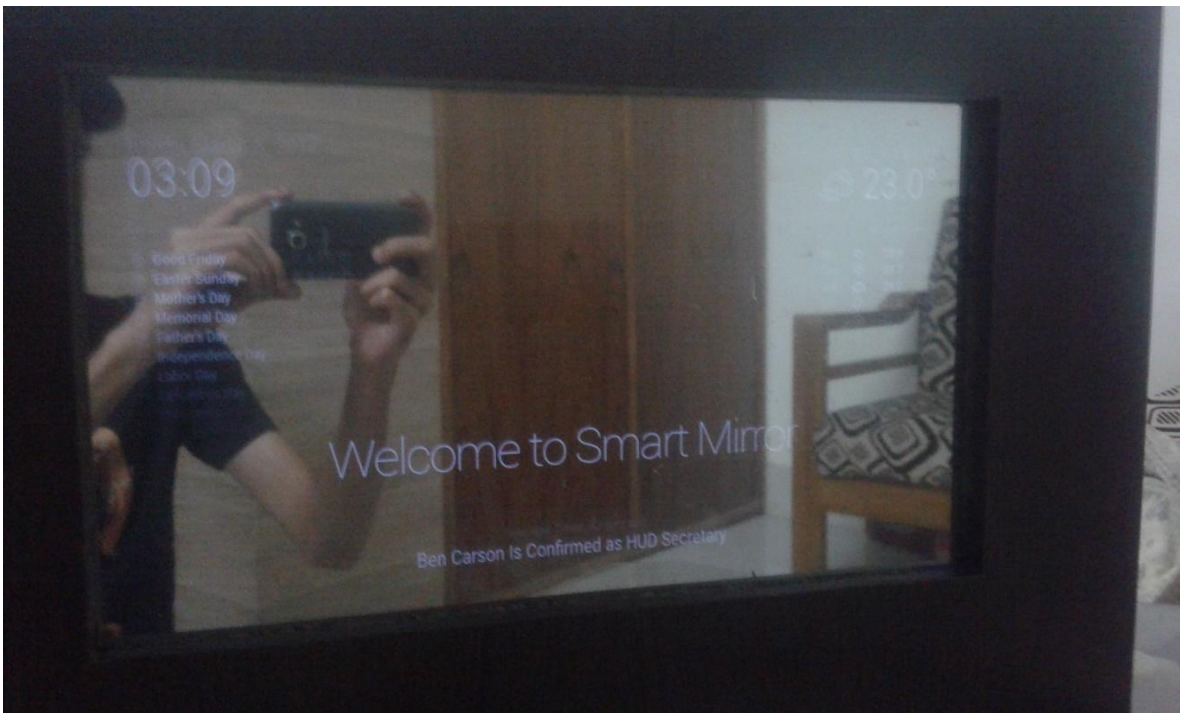

Figure 4.3: View of Smart Mirror (When the room is dark)

CHAPTER 5

CHALLENGES AND FUTURE IMPROVEMENTS

5.1 Challenges Faced

IoT provides a great platform for enablement of social development in varied societies across the world and with the proliferation of Internet across the various sections of the society in developing countries coupled with lowering costs of microprocessors and sensors will make IoT devices accessible to low income households.

But there are lot of shortcomings related to enablement of high speed internet and basic technology services architecture for commercial and business usage in developing countries. Until and unless, a basic infrastructure is in place, devices would be of no value to the users.

While IoT brings about new opportunities; at the same time, it adds multiple layers of complexity. Such a new environment of devices will add a new dimension for policy makers in emerging economies who will need to chalk out a new blueprint for IoT related regulatory concerns.

In an ideal environment, information exchange should take place between all the interconnected IoT devices. But the actual scenario is inherently more complex and depends on various levels of communication protocols stacks between such devices.

Making smart mirrors was not so easy for us. We found lot of difficulty in searching of hardware. One way mirror is not available in our country. We searched one way mirror everywhere in the city but we failed to manage the one way mirror. We made lot of research about one way mirror. Finally we got the solution for making a normal mirror into the one way mirror. Making the wood frame was another challenge for us. Collecting information about this kind of project was not easy because there were not so much information about this kind of project.

- Scarcity of information concerning similar projects
- Unavailability of hardware parts
- Difficulty to manage fund for the project.

5.2 Future Improvements

Time and other resources is always factor in a better product. There is always room for improvement and this project has a lot of areas that needs work. Voice control system must be added to the smart mirror so that anyone can update their calendar, schedule, and so much more. Our future plan is adding facial recognition so that after setting up a profile and sending a picture of themselves to the Cognitive Services database, everyone would be able to step in front of the mirror and get a personalized display showing the information based on our preferences. Our mirror is not hundred percent reflective we will try to make this more reflective. We will add new feature like Email module, Facebook module another module according to the demand of Smart Mirror User.

CHAPTER 6

CONCLUSION

6.1 Conclusion

The main strengths of this project are that this is a new kind of smart device that people don’t see every day and it looks very spectacular. The platform has a very simple API that makes it very easy for developers to make apps. Of course there are also weaknesses the app eco system is currently very small, the glass could be more reflective but it can be easily changed, the swipe gestures are sometimes unreliable and however, this can also solved given enough time by making the software more modular. There are many future possibilities for this project and hopefully it will be continued. For the software, it would be interesting to create an installer for it or even bundle it as a Linux distribution to be able to install it very easily on any Raspberry Pi device.

Appendix

Index.html

```
<!DOCTYPE html>
<html>
<head>
	<title>Magic Mirror</title>
	<meta name="google" content="notranslate" />
	<meta http-equiv="Content-type" content="text/html; charset=utf-8" />
	<link rel="icon" href="data:;base64,iVBORw0KGgo=">
	<link rel="stylesheet" type="text/css" href="css/main.css">
	<link rel="stylesheet" type="text/css" href="fonts/roboto.css">
	<!-- custom.css is loaded by the loader.js to make sure it's loaded after the module   css files. -->
	<script type="text/javascript">
	var version = "#VERSION#";
	</script>
</head>
<body>
	<div class="region fullscreen below"><div class="container"></div></div>
	<div class="region top bar">
		<div class="container"></div>
		<div class="region top left"><div class="container"></div></div>
		<div class="region top center"><div class="container"></div></div>
		<div class="region top right"><div class="container"></div></div>
	</div>
```

```
        <div class="region upper third"><div class="container"></div></div>
        <div class="region middle center"><div class="container"></div></div>
        <div class="region lower third"><div class="container"><br/></div></div>
        <div class="region bottom bar">
            <div class="container"></div>
            <div class="region bottom left"><div class="container"></div></div>
            <div class="region bottom center"><div class="container"></div>
            </div>
            <div class="region bottom right"><div class="container"></div>
            </div>
        </div>
        <div class="region fullscreen above"><div class="container"></div></div>
        <script type="text/javascript" src="/socket.io/socket.io.js"></script>
        <script type="text/javascript" src="js/defaults.js"></script>
        <script type="text/javascript" src="config/config.js"></script>
        <script type="text/javascript" src="vendor/vendor.js"></script>
        <script type="text/javascript"
        src="modules/default/defaultmodules.js"></script>
        <script type="text/javascript" src="js/logger.js"></script>
        <script type="text/javascript" src="translations/translations.js"></script>
        <script type="text/javascript" src="js/translator.js"></script>
        <script type="text/javascript" src="js/class.js"></script>
        <script type="text/javascript" src="js/module.js"></script>
        <script type="text/javascript" src="js/loader.js"></script>
        <script type="text/javascript" src="js/socketclient.js"></script>
        <script type="text/javascript" src="js/main.js"></script>
</body>
</html>
```

main.css

```
html {
            cursor: none;
            overflow: hidden;
}
           ::-webkit-scrollbar {
             display: none;
}
body {
            margin: 60px;
             position: absolute;
             height: calc(100% - 120px);
             width: calc(100% - 120px);
             background: #000;
             color: #aaa;
             font-family: "Roboto Condensed", sans-serif;
             font-weight: 400;
             font-size: 2em;
             line-height: 1.5em;
             -webkit-font-smoothing: antialiased;
}
           .dimmed {
             color: #666;
}
           .normal {
            color: #999;  }
           .bright {
             color: #fff;  }
.xsmall {
             font-size: 15px;
             line-height: 20px;
}
.small {
             font-size: 20px;
             line-height: 25px;
}
```

```
.medium {
	font-size: 30px;
	line-height: 35px;
}
.large {
	font-size: 65px;
	line-height: 65px;
}
.xlarge {
	font-size: 75px;
	line-height: 75px;
	letter-spacing: -3px;
}
.thin {
	font-family: Roboto, sans-serif;
	font-weight: 100;
}
.light {
	font-family: "Roboto Condensed", sans-serif;
	font-weight: 300;
}
.regular {
	font-family: "Roboto Condensed", sans-serif;
	font-weight: 400;
}
.bold {
	font-family: "Roboto Condensed", sans-serif;
	font-weight: 700;
}
.align-right {
	text-align: right;
}
.align-left {
	text-align: left;
}
header {
	text-transform: uppercase;
	font-size: 15px;
	font-family: "Roboto Condensed";
```

```
        font-weight: 400;
        border-bottom: 1px solid #666;
        line-height: 15px;
        padding-bottom: 5px;
        margin-bottom: 10px;
        color: #999;
}
sup {
        font-size: 50%;
        line-height: 50%;
}
.module {
        margin-bottom: 30px;
}
.region.bottom .module {
        margin-top: 30px;
        margin-bottom: 0;
}
.region {
        position: absolute;
}
.region.fullscreen {
        position: absolute;
        top: -60px;
        left: -60px;
        right: -60px;
        bottom: -60px;
}
.region.right {
        right: 0;
}
.region.top {
        top: 0;
}

.region.top .container {
        margin-bottom: 25px;
}
```

```
.region.top .container:empty {
            margin-bottom: 0;
}
          .region.top.center,
          .region.bottom.center {
            left: 50%;
            -moz-transform: translateX(-50%);
            -o-transform: translateX(-50%);
            -webkit-transform: translateX(-50%);
            -ms-transform: translateX(-50%);
            transform: translateX(-50%);
}
          .region.top.right,
          .region.top.left,
          .region.top.center {
            top: 100%;
}
          .region.bottom {
            bottom: 0;
}
          .region.bottom .container {
            margin-top: 25px;
}
          .region.bottom .container:empty {
            margin-top: 0;
}
          .region.bottom.right,
          .region.bottom.center,
          .region.bottom.left {
            bottom: 100%;
}
          .region.bar {
            width: 100%;
            text-align: center;
}
          .region.third,
          .region.middle.center
 {
```

```
        width: 100%;
        text-align: center;
        -moz-transform: translateY(-50%);
        -o-transform: translateY(-50%);
        -webkit-transform: translateY(-50%);
        -ms-transform: translateY(-50%);
        transform: translateY(-50%);
}
      .region.upper.third {
        top: 33%;
}
      .region.middle.center {
        top: 50%;
}
      .region.lower.third {
        top: 66%;
}
      .region.left {
        text-align: left;
}
      .region.right {
        text-align: right;
}
      .region table {
        width: 100%;
        border-spacing: 0;
        border-collapse: separate;
}
```

electron.js

```
 /* jshint esversion: 6 */
"use strict";
const Server = require(__dirname + "/server.js");
const electron = require("electron");
const core = require(__dirname + "/app.js");
        // Config
var config = {};
        // Module to control application life.
const app = electron.app;
        // Module to create native browser window.
const BrowserWindow = electron.BrowserWindow;
        // Keep a global reference of the window object, if you don't, the window will
        // be closed automatically when the JavaScript object is garbage collected.
let mainWindow;
function createWindow() {
                var electronOptionsDefaults = {
                        width: 800,
                        height: 600,
                        x: 0,
                        y: 0,
                        darkTheme: true,
                        webPreferences: {
                                nodeIntegration: false,
                                zoomFactor: config.zoom
                        }
                }
                // DEPRECATED: "kioskmode" backwards compatibility, to be removed
                // settings these options directly instead provides cleaner interface
                if (config.kioskmode) {
                        electronOptionsDefaults.kiosk = true;
                } else {
                        electronOptionsDefaults.fullscreen = true;
                        electronOptionsDefaults.autoHideMenuBar = true;
                }
```

```
var electronOptions = Object.assign({}, electronOptionsDefaults,
config.electronOptions);
// Create the browser window.

        mainWindow = new BrowserWindow(electronOptions);
        // and load the index.html of the app.
        //mainWindow.loadURL('file://' + __dirname + '../../index.html');
        mainWindow.loadURL("http://localhost:" + config.port);
        // Open the DevTools if run with "npm start dev"
        if(process.argv[2] == "dev") {
                mainWindow.webContents.openDevTools();
        }
        // Set responders for window events.
        mainWindow.on("closed", function() {
                mainWindow = null;
        });
        if (config.kioskmode) {
                mainWindow.on("blur", function() {
                        mainWindow.focus();
                });
                mainWindow.on("leave-full-screen", function() {
                        mainWindow.setFullScreen(true);
                });
                mainWindow.on("resize", function() {
                        setTimeout(function() {
                                mainWindow.reload();
                        }, 1000);
                });
        }
}
// This method will be called when Electron has finished
// initialization and is ready to create browser windows.
app.on("ready", function() {
        console.log("Launching application.");
        createWindow();
});
```

```
// Quit when all windows are closed.
app.on("window-all-closed", function() {
		createWindow();
});

app.on("activate", function() {

	// On OS X it's common to re-create a window in the app when the
	// dock icon is clicked and there are no other windows open.
	if (mainWindow === null) {
		createWindow();
	}
});

// Start the core application.
// This starts all node helpers and starts the webserver.
core.start(function(c) {
	config = c;
});
```

modules/index.js

```
var Class = require("../../../js/class.js");
var express = require("express");
var path = require("path");

NodeHelper = Class.extend({
	init: function() {
		console.log("Initializing new module helper ...");
	};
	start: function() {
		console.log("Staring module helper: " + this.name);
	};
	socketNotificationReceived: function(notification, payload) {
		console.log(this.name + " received a socket notification: " + notification +
	" - Payload: " + payload);
	};
	setName: function(name) {
		this.name = name;
	};
	setPath: function(path) {
		this.path = path;
	};
	sendSocketNotification: function(notification, payload) {
		this.io.of(this.name).emit(notification, payload);
	};
	setExpressApp: function(app) {
		this.expressApp = app;
		var publicPath = this.path + "/public";
		app.use("/" + this.name, express.static(publicPath));
	};
```

```
        setSocketIO: function(io) {
                var self = this;
                self.io = io;

                console.log("Connecting socket for: " + this.name);
                var namespace = this.name;
                io.of(namespace).on("connection", function(socket) {
                        // add a catch all event.
                        var onevent = socket.onevent;
                        socket.onevent = function(packet) {
                                var args = packet.data || [];
                                onevent.call(this, packet);    // original call
                                packet.data = ["*"].concat(args);
                                onevent.call(this, packet);      // additional call to catch-all
                        };
                        // register catch all.
                        socket.on("*", function(notification, payload) {
                                if (notification !== "*")
                                //console.log('received message in namespace: ' +
                                namespace);
                                self.socketNotificationReceived(notification, payload);
                        });
                });
        }
});
NodeHelper.create = function(moduleDefinition) {
        return NodeHelper.extend(moduleDefinition);
};
module.exports = NodeHelper ;
```

calendar.js

```
Module.register("calendar", {
	// Define module defaults
	defaults: {
		maximumEntries: 10, // Total Maximum Entries
		maximumNumberOfDays: 365,
		displaySymbol: true,
		defaultSymbol: "calendar", // Fontawesome Symbol see
http://fontawesome.io/cheatsheet/
		displayRepeatingCountTitle: false,
		defaultRepeatingCountTitle: "",
		maxTitleLength: 25,
		fetchInterval: 5 * 60 * 1000, // Update every 5 minutes.
		animationSpeed: 2000,
		fade: true,
		urgency: 7,
		timeFormat: "relative",
		dateFormat: "MMM Do",
		getRelative: 6,
		fadePoint: 0.25, // Start on 1/4th of the list.
		hidePrivate: false,
		calendars: [
			{
			symbol: "calendar",
				url: "http://www.calendarlabs.com/templates/ical/US-
				Holidays.ics",
			},
			],
		titleReplace: {
			"De verjaardag van ": "",
			"'s birthday": ""
		},
		broadcastEvents: true
	},
	// Define required scripts.
	getStyles: function () {
		return ["calendar.css", "font-awesome.css"];
		},
```

```
// Define required scripts.
	getScripts: function () {
		return ["moment.js"];
	},
	// Define required translations.
	getTranslations: function () {
		// The translations for the defaut modules are defined in the core translation files.
		// Therefor we can just return false. Otherwise we should have returned a dictionairy.
		// If you're trying to build your own module including translations, check out the documentation.
		return false;
	},
	// Override start method.
	start: function () {
		Log.log("Starting module: " + this.name);
		// Set locale.
		moment.locale(config.language);
		for (var c in this.config.calendars) {
			var calendar = this.config.calendars[c];
			calendar.url = calendar.url.replace("webcal://", "http://");
			this.addCalendar(calendar.url, calendar.user, calendar.pass);
		}
		this.calendarData = {};
		this.loaded = false;
	}, // Override socket notification handler.
	socketNotificationReceived: function (notification, payload) {
		if (notification === "CALENDAR_EVENTS") {
			if (this.hasCalendarURL(payload.url)) {
				this.calendarData[payload.url] = payload.events;
				this.loaded = true;
				if (this.config.broadcastEvents) {
					this.broadcastEvents();
				}
			}
		} else if (notification === "FETCH_ERROR") {
			Log.error("Calendar Error. Could not fetch calendar: " + payload.url);
```

```
            } else if (notification === "INCORRECT_URL") {
                Log.error("Calendar Error. Incorrect url: " + payload.url);
            } else {
                Log.log("Calendar received an unknown socket notification: " +
                notification);
            }
            this.updateDom(this.config.animationSpeed);
        },  // Override dom generator.
        getDom: function () {
            var events = this.createEventList();
            var wrapper = document.createElement("table");
            wrapper.className = "small";
            if (events.length === 0) {
                wrapper.innerHTML = (this.loaded) ? this.translate("EMPTY") :
                this.translate("LOADING");
                wrapper.className = "small dimmed";
                return wrapper;
            }
            for (var e in events) {
                var event = events[e];
                var eventWrapper = document.createElement("tr");
                eventWrapper.className = "normal";
                if (this.config.displaySymbol) {
                    var symbolWrapper = document.createElement("td");
                    symbolWrapper.className = "symbol";
                    var symbol = document.createElement("span");
                symbol.className = "fa fa-" + this.symbolForUrl(event.url);
                    symbolWrapper.appendChild(symbol);
                    eventWrapper.appendChild(symbolWrapper);
                }
                var titleWrapper = document.createElement("td"),
                    repeatingCountTitle = "";
                if (this.config.displayRepeatingCountTitle) {
                    repeatingCountTitle = this.countTitleForUrl(event.url);
                    if (repeatingCountTitle !== "") {
                var thisYear = new Date(parseInt(event.startDate)).getFullYear(),
                    yearDiff = thisYear - event.firstYear;
                repeatingCountTitle = ", " + yearDiff + ". " + repeatingCountTitle;
                    }
                }
```

```
titleWrapper.innerHTML = this.titleTransform(event.title) + repeatingCountTitle;
                titleWrapper.className = "title bright";
                eventWrapper.appendChild(titleWrapper);
                var timeWrapper = document.createElement("td");
                //console.log(event.today);
                var now = new Date();
                // Define second, minute, hour, and day variables
                var oneSecond = 1000; // 1,000 milliseconds
                var oneMinute = oneSecond * 60;
                var oneHour = oneMinute * 60;
                var oneDay = oneHour * 24;
                if (event.fullDayEvent) {
                if (event.today) {
            timeWrapper.innerHTML = this.capFirst(this.translate("TODAY"));
            } else if (event.startDate - now < oneDay && event.startDate - now > 0) {
            timeWrapper.innerHTML = this.capFirst(this.translate("TOMORROW"));
    } else if (event.startDate - now < 2 * oneDay && event.startDate - now > 0) {
                if (this.translate("DAYAFTERTOMORROW") !==
    "DAYAFTERTOMORROW") {
                timeWrapper.innerHTML =
    this.capFirst(this.translate("DAYAFTERTOMORROW"));
            } else {
    timeWrapper.innerHTML = this.capFirst(moment(event.startDate, "x").fromNow());
            }
            } else {
            if (this.config.timeFormat === "absolute") {
            if ((this.config.urgency > 1) && (event.startDate - now <
    (this.config.urgency * oneDay))) {
            // This event falls within the config.urgency period that the user has set
            timeWrapper.innerHTML = this.capFirst(moment(event.startDate,
    "x").fromNow());
            } else {
            timeWrapper.innerHTML = this.capFirst(moment(event.startDate,
"x").format(this.config.dateFormat));
            }
            } else {
            timeWrapper.innerHTML = this.capFirst(moment(event.startDate,
    "x").fromNow());
                        }
                }
```

```
} else {
        if (event.startDate >= new Date()) {
        if (event.startDate - now < 2 * oneDay) {
        // This event is within the next 48 hours (2 days)
        if (event.startDate - now < this.config.getRelative * oneHour) {
// If event is within 6 hour, display 'in xxx' time format or moment.fromNow()
timeWrapper.innerHTML = this.capFirst(moment(event.startDate, "x").fromNow());
        } else {
                // Otherwise just say 'Today/Tomorrow at such-n-such time'
timeWrapper.innerHTML = this.capFirst(moment(event.startDate, "x").calendar());
        }
        } else {
        if (this.config.timeFormat === "absolute") {
        if ((this.config.urgency > 1) && (event.startDate - now <
(this.config.urgency * oneDay))) {
        // This event falls within the config.urgency period that the user has set
timeWrapper.innerHTML = this.capFirst(moment(event.startDate, "x").fromNow());
        } else {
        timeWrapper.innerHTML = this.capFirst(moment(event.startDate,
        "x").format(this.config.dateFormat));
        }
        } else {
        timeWrapper.innerHTML = this.capFirst(moment(event.startDate,
"x").fromNow());
        }
                }
        } else {
        timeWrapper.innerHTML = this.capFirst(this.translate("RUNNING")) + " "
+ moment(event.endDate, "x").fromNow(true);
                                }
                        }
//timeWrapper.innerHTML += ' - '+ moment(event.startDate,'x').format('lll');
//console.log(event);
        timeWrapper.className = "time light";
        eventWrapper.appendChild(timeWrapper);
                wrapper.appendChild(eventWrapper);
        // Create fade effect.
                        if (this.config.fade && this.config.fadePoint < 1) {
                                if (this.config.fadePoint < 0) {
                                        this.config.fadePoint = 0;
```

```
}var startingPoint = events.length * this.config.fadePoint;
                var steps = events.length - startingPoint;
                if (e >= startingPoint) {
                var currentStep = e - startingPoint;
                eventWrapper.style.opacity = 1 - (1 / steps * currentStep);
                                }
                        }
                }
        return wrapper;
        };
        hasCalendarURL: function (url) {
                for (var c in this.config.calendars) {
                        var calendar = this.config.calendars[c];
                        if (calendar.url === url) {
                                return true;
                        }
                } return false;
        };
        createEventList: function () {
                var events = [];
                var today = moment().startOf("day");
                for (var c in this.calendarData) {
                        var calendar = this.calendarData[c];
                        for (var e in calendar) {
                                var event = calendar[e];
                                if(this.config.hidePrivate) {
                                        if(event.class === "PRIVATE") {
                                        // do not add the current event, skip it
                                             continue;
                                        }
                                }
                                event.url = c;
                                event.today = event.startDate >= today &&
event.startDate < (today + 24 * 60 * 60 * 1000);
                                events.push(event);
                        }
                }
                events.sort(function (a, b) {
                        return a.startDate - b.startDate;
                });
```

```
return events.slice(0, this.config.maximumEntries);
        };
        addCalendar: function (url, user, pass) {
        this.sendSocketNotification("ADD_CALENDAR", {
                url: url,
                maximumEntries: this.config.maximumEntries,
                maximumNumberOfDays: this.config.maximumNumberOfDays,
                fetchInterval: this.config.fetchInterval,
                user: user,
                pass: pass
                });
        };
        symbolForUrl: function (url) {
                for (var c in this.config.calendars) {
                var calendar = this.config.calendars[c];
                if (calendar.url === url && typeof calendar.symbol === "string") {
                        return calendar.symbol;
                        }
                }
                return this.config.defaultSymbol;
        };
        countTitleForUrl: function (url) {
        for (var c in this.config.calendars) {
                var calendar = this.config.calendars[c];
                if (calendar.url === url && typeof calendar.repeatingCountTitle
                === "string") {
                return calendar.repeatingCountTitle;
                        }
                }
                return this.config.defaultRepeatingCountTitle;
        };
        shorten: function (string, maxLength) {
                if (string.length > maxLength) {
                        return string.slice(0, maxLength) + "&hellip;";
                }
        return string;
        };
capFirst: function (string) {
                return string.charAt(0).toUpperCase() + string.slice(1);
```

```
};
        titleTransform: function (title) {
                for (var needle in this.config.titleReplace) {
                        var replacement = this.config.titleReplace[needle];
                        title = title.replace(needle, replacement);
                }
                title = this.shorten(title, this.config.maxTitleLength);
                return title;
        };
        broadcastEvents: function () {
                var eventList = [];
                for (url in this.calendarData) {

                        var calendar = this.calendarData[url];
                        for (e in calendar) {

                                var event = cloneObject(calendar[e]);
                                delete event.url;
                                eventList.push(event);
                        }
                }
                eventList.sort(function(a,b) {
                        return a.startDate - b.startDate;
                });
                this.sendNotification("CALENDAR_EVENTS", eventList);
        }
});
```

clock.js

```
	Module.register("clock",{
	// Module config defaults.
	defaults: {
		displayType: "digital", // options: digital, analog, both
		timeFormat: config.timeFormat,
		displaySeconds: true,
		showPeriod: true,
		showPeriodUpper: false,
		clockBold: false,
		showDate: true,
		/* specific to the analog clock */
		analogSize: "200px",
analogFace: "simple", // options: 'none', 'simple', 'face-###' (where ### is 001 to 012 inclusive)
		analogPlacement: "bottom", // options: 'top', 'bottom', 'left', 'right'
		analogShowDate: "top", // options: false, 'top', or 'bottom'
		secondsColor: "#888888",
		timezone: null,
	}, // Define required scripts.
	getScripts: function() {
		return ["moment.js", "moment-timezone.js"];
	}, // Define styles.
	getStyles: function() {
		return ["clock_styles.css"];
	}, // Define start sequence.
	start: function() {
		Log.info("Starting module: " + this.name);
		// Schedule update interval.
		var self = this;
		setInterval(function() {
		self.updateDom();
		}, 1000);
		// Set locale.
		moment.locale(config.language);
	},
	// Override dom generator.
	getDom: function() {
		var wrapper = document.createElement("div");
		var dateWrapper = document.createElement("div");
```

```
var timeWrapper = document.createElement("div");
var secondsWrapper = document.createElement("sup");
var periodWrapper = document.createElement("span");
// Style Wrappers

dateWrapper.className = "date normal medium";
timeWrapper.className = "time bright large light";
secondsWrapper.className = "dimmed";

// Set content of wrappers.
// The moment().format("h") method has a bug on the Raspberry Pi.
// So we need to generate the timestring manually.
// See issue: https://github.com/MichMich/MagicMirror/issues/181
var timeString;
var now = moment();
if (this.config.timezone) {
	now.tz(this.config.timezone);
}
if (this.config.clockBold === true) {
	timeString = now.format("HH[<span class=\"bold\">]mm[</span>]");
} else {
	timeString = now.format("HH:mm");
}
if (this.config.timeFormat !== 24) {
	// var now = new Date();
	// var hours = now.getHours() % 12 || 12;
	if (this.config.clockBold === true) {
	//timeString = hours + moment().format("[<span
	class=\"bold\">]mm[</span>]");
	timeString = now.format("h[<span class=\"bold\">]mm[</span>]");
	} else {
		//timeString = hours + moment().format(":mm");
		timeString = now.format("h:mm");
	}
}
if(this.config.showDate){
	dateWrapper.innerHTML = now.format("dddd, LL");
}
timeWrapper.innerHTML = timeString;
secondsWrapper.innerHTML = now.format("ss");
```

```
                if (this.config.showPeriodUpper) {
                        periodWrapper.innerHTML = now.format("A");
                } else {
                        periodWrapper.innerHTML = now.format("a");
                }
                if (this.config.displaySeconds) {
                        timeWrapper.appendChild(secondsWrapper);
                }
                if (this.config.showPeriod && this.config.timeFormat !== 24) {
                        timeWrapper.appendChild(periodWrapper);
                }
                 if (this.config.displayType !== "digital") {
                        // If it isn't 'digital', then an 'analog' clock was also requested
                        // Calculate the degree offset for each hand of the clock
                        var now = moment();
                        if (this.config.timezone) {
                                now.tz(this.config.timezone);
                        }
                        var     second = now.seconds() * 6,
                                minute = now.minute() * 6 + second / 60,
                                hour = ((now.hours() % 12) / 12) * 360 + 90 + minute / 12;
                        // Create wrappers
                        var wrapper = document.createElement("div");
                        var clockCircle = document.createElement("div");
                        clockCircle.className = "clockCircle";
                        clockCircle.style.width = this.config.analogSize;
                        clockCircle.style.height = this.config.analogSize;
                        if (this.config.analogFace != "" && this.config.analogFace != "simple"
&& this.config.analogFace != "none") {
                                clockCircle.style.background = "url("+ this.data.path + "faces/" +
this.config.analogFace + ".svg)";
                        clockCircle.style.backgroundSize = "100%";
                        } else if (this.config.analogFace != "none") {
                        clockCircle.style.border = "2px solid white";
                        }
                        var clockFace = document.createElement("div");
                        clockFace.className = "clockFace";
                        var clockHour = document.createElement("div");
                        clockHour.id = "clockHour";
                        clockHour.style.transform = "rotate(" + hour + "deg)";
                        clockHour.className = "clockHour";
```

```
                var clockMinute = document.createElement("div");
                clockMinute.id = "clockMinute";
                clockMinute.style.transform = "rotate(" + minute + "deg)";
                clockMinute.className = "clockMinute"; // Combine analog wrappers
                clockFace.appendChild(clockHour);
                clockFace.appendChild(clockMinute);
                if (this.config.displaySeconds) {
                        var clockSecond = document.createElement("div");
                        clockSecond.id = "clockSecond";
                        clockSecond.style.transform = "rotate(" + second + "deg)";
                        clockSecond.className = "clockSecond";
                        clockSecond.style.backgroundColor = this.config.secondsColor;
                        clockFace.appendChild(clockSecond);
                }
                clockCircle.appendChild(clockFace);
        }
        if (this.config.displayType === "digital") {
                // Display only a digital clock
                wrapper.appendChild(dateWrapper);
                wrapper.appendChild(timeWrapper);
        } else if (this.config.displayType === "analog") {
                // Display only an analog clock
                dateWrapper.style.textAlign = "center";
                dateWrapper.style.paddingBottom = "15px";
                if (this.config.analogShowDate === "top") {
                        wrapper.appendChild(dateWrapper);
                        wrapper.appendChild(clockCircle);
                } else if (this.config.analogShowDate === "bottom") {
                        wrapper.appendChild(clockCircle);
                        wrapper.appendChild(dateWrapper);
                } else {
                        wrapper.appendChild(clockCircle);
                }
        } else {
                // Both clocks have been configured, check position
                var placement = this.config.analogPlacement;
                analogWrapper = document.createElement("div");
                analogWrapper.id = "analog";
                analogWrapper.style.cssFloat = "none";
                analogWrapper.appendChild(clockCircle);
```

```
                digitalWrapper = document.createElement("div");
                digitalWrapper.id = "digital";
                digitalWrapper.style.cssFloat = "none";
                digitalWrapper.appendChild(dateWrapper);
                digitalWrapper.appendChild(timeWrapper);

                if (placement === "left" || placement === "right") {
                        digitalWrapper.style.display = "inline-block";
                        digitalWrapper.style.verticalAlign = "top";
                        analogWrapper.style.display = "inline-block";
                        if (placement === "left") {
                                analogWrapper.style.padding = "0 20px 0 0";
                                wrapper.appendChild(analogWrapper);
                                wrapper.appendChild(digitalWrapper);
                        } else {
                                analogWrapper.style.padding = "0 0 0 20px";
                                wrapper.appendChild(digitalWrapper);
                                wrapper.appendChild(analogWrapper);
                        }
                } else {
                        digitalWrapper.style.textAlign = "center";
                        if (placement === "top") {
                                analogWrapper.style.padding = "0 0 20px 0";
                                wrapper.appendChild(analogWrapper);
                                wrapper.appendChild(digitalWrapper);
                        } else {
                                analogWrapper.style.padding = "20px 0 0 0";
                                wrapper.appendChild(digitalWrapper);
                                wrapper.appendChild(analogWrapper);
                        }
                }
        }

        // Return the wrapper to the dom.
        return wrapper;
    }
});
```

CurrentWeather

```
Module.register("compliments",{
	// Module config defaults.
	defaults: {
		compliments: {
			morning: [
				"Good morning, handsome!",
				"Enjoy your day!",
				"How was your sleep?"
			],
			afternoon: [
				"Hello, beauty!",
				"You look sexy!",
				"Looking good today!"
			],
			evening: [
				"Wow, you look hot!",
				"You look nice!",
				"Hi, sexy!"
			]
		},
		updateInterval: 30000,
		remoteFile: null,
		fadeSpeed: 4000
	},
	// Set currentweather from module
	currentWeatherType: "",
	// Define required scripts.
	getScripts: function() {
		return ["moment.js"];
	},
	// Define start sequence.
	start: function() {
		Log.info("Starting module: " + this.name);
		this.lastComplimentIndex = -1;
		if (this.config.remoteFile != null) {
			this.complimentFile((response) => {
				this.config.compliments = JSON.parse(response);
			});
		}  // Schedule update timer.
		var self = this;
```

```
setInterval(function() {
                        self.updateDom(self.config.fadeSpeed);
                }, this.config.updateInterval);
        },
        randomIndex: function(compliments) {
                if (compliments.length === 1) {
                return 0;
                }
                var generate = function() {
                return Math.floor(Math.random() * compliments.length);
                };
                var complimentIndex = generate();
                while (complimentIndex === this.lastComplimentIndex) {
                complimentIndex = generate();
                }
                this.lastComplimentIndex = complimentIndex;
                return complimentIndex; },
        complimentArray: function() {
                var hour = moment().hour();
                var compliments  = null;
                if (hour >= 3 && hour < 12) {
                        compliments = this.config.compliments.morning;
                } else if (hour >= 12 && hour < 17) {
                        compliments = this.config.compliments.afternoon;
                } else {
                        compliments = this.config.compliments.evening;
                }
                if ( this.currentWeatherType in this.config.compliments) {
                        compliments.push.apply(compliments,
this.config.compliments[this.currentWeatherType]);
                }
                return compliments;
        },
        complimentFile: function(callback) {
                var xobj = new XMLHttpRequest();
                xobj.overrideMimeType("application/json");
                xobj.open("GET", this.file(this.config.remoteFile), true);
                xobj.onreadystatechange = function () {
                        if (xobj.readyState == 4 && xobj.status == "200") {
                        callback(xobj.responseText);
                        }
                };
```

```
        xobj.send(null);
        },
        randomCompliment: function() {
                var compliments = this.complimentArray();
                var index = this.randomIndex(compliments);
                return compliments[index];
        },
        getDom: function() {
                var complimentText = this.randomCompliment();
                var compliment = document.createTextNode(complimentText);
                var wrapper = document.createElement("div");
        wrapper.className = this.config.classes ? this.config.classes : "thin xlarge bright";
                wrapper.appendChild(compliment);
                return wrapper;
        },
        setCurrentWeatherType: function(data) {
                var weatherIconTable = {
                        "01d": "day_sunny",
                        "02d": "day_cloudy",
                        "03d": "cloudy",
                        "04d": "cloudy_windy",
                        "09d": "showers",
                        "10d": "rain",
                        "11d": "thunderstorm",
                        "13d": "snow",
                        "50d": "fog",
                        "01n": "night_clear",
                        "02n": "night_cloudy",
                        "03n": "night_cloudy",
                        "04n": "night_cloudy",
                        "09n": "night_showers",
                        "10n": "night_rain",
                        "11n": "night_thunderstorm",
                        "13n": "night_snow",
                        "50n": "night_alt_cloudy_windy"          };
                this.currentWeatherType = weatherIconTable[data.weather[0].icon];
        },
        notificationReceived: function(notification, payload, sender) {
                if (notification == "CURRENTWEATHER_DATA") {
                        this.setCurrentWeatherType(payload.data); }
        },
});
```

currentweather.js

```
Module.register("currentweather",{
	// Default module config.
	defaults: {
		location: false,
		locationID: false,
		appid: "",
		units: config.units,
		updateInterval: 10 * 60 * 1000, // every 10 minutes
		animationSpeed: 1000,
		timeFormat: config.timeFormat,
		showPeriod: true,
		showPeriodUpper: false,
		showWindDirection: true,
		useBeaufort: true,
		lang: config.language,
		showHumidity: false,
		initialLoadDelay: 0, // 0 seconds delay
		retryDelay: 2500,
		apiVersion: "2.5",
		apiBase: "http://api.openweathermap.org/data/",
		weatherEndpoint: "weather",
		appendLocationNameToHeader: true,
		calendarClass: "calendar",
		onlyTemp: false,
		roundTemp: false,
		iconTable: {
			"01d": "wi-day-sunny",
			"02d": "wi-day-cloudy",
			"03d": "wi-cloudy",
			"04d": "wi-cloudy-windy",
			"09d": "wi-showers",
			"10d": "wi-rain",
			"11d": "wi-thunderstorm",
			"13d": "wi-snow",
			"50d": "wi-fog",
			"01n": "wi-night-clear",
			"02n": "wi-night-cloudy",
			"03n": "wi-night-cloudy",
			"04n": "wi-night-cloudy",
			"09n": "wi-night-showers",
```

```
                        "11n": "wi-night-thunderstorm",
                        "13n": "wi-night-snow",
                        "50n": "wi-night-alt-cloudy-windy"
                },
        },
        firstEvent: false,
        // create a variable to hold the location name based on the API result.
        fetchedLocatioName: "",
        // Define required scripts.
        getScripts: function() {
                return ["moment.js"];
        },
        // Define required scripts.
        getStyles: function() {
                return ["weather-icons.css", "currentweather.css"];
        },
        // Define required translations.
        getTranslations: function() {
        // The translations for the defaut modules are defined in the core translation files.
        // Therefor we can just return false. Otherwise we should have returned a dictionair
                return false;
        },
        // Define start sequence.
        start: function() {
                Log.info("Starting module: " + this.name);  // Set locale.
                moment.locale(config.language);
                this.windSpeed = null;
                this.windDirection = null;
                this.sunriseSunsetTime = null;
                this.sunriseSunsetIcon = null;
                this.temperature = null;
                this.weatherType = null;
                this.loaded = false;
                this.scheduleUpdate(this.config.initialLoadDelay);
        },
        // add extra information of current weather
        // windDirection, humidity, sunrise and sunset
        addExtraInfoWeather: function(wrapper) {
                var small = document.createElement("div");
                small.className = "normal medium";
                var windIcon = document.createElement("span");
                windIcon.className = "wi wi-strong-wind dimmed";
```

```
                small.appendChild(windIcon);
                var windSpeed = document.createElement("span");
                windSpeed.innerHTML = " " + this.windSpeed;
                small.appendChild(windSpeed);
                if (this.config.showWindDirection) {
                        var windDirection = document.createElement("sup");
                        windDirection.innerHTML = " " + this.translate(this.windDirection);
                        small.appendChild(windDirection);
                }
                var spacer = document.createElement("span");
                spacer.innerHTML = " ";
                small.appendChild(spacer);
                if (this.config.showHumidity) {
                        var humidity = document.createElement("span");
                        humidity.innerHTML = this.humidity;
                        var spacer = document.createElement("sup");
                        spacer.innerHTML = " ";
                        var humidityIcon = document.createElement("sup");
                        humidityIcon.className = "wi wi-humidity humidityIcon";
                        humidityIcon.innerHTML = " ";
                        small.appendChild(humidity);
                        small.appendChild(spacer);
                        small.appendChild(humidityIcon);
                }
                var sunriseSunsetIcon = document.createElement("span");
                sunriseSunsetIcon.className = "wi dimmed " + this.sunriseSunsetIcon;
                small.appendChild(sunriseSunsetIcon);
                var sunriseSunsetTime = document.createElement("span");
                sunriseSunsetTime.innerHTML = " " + this.sunriseSunsetTime;
                small.appendChild(sunriseSunsetTime);
                wrapper.appendChild(small);
        },
        // Override dom generator.
        getDom: function() {
                var wrapper = document.createElement("div");
                if (this.config.appid === "") {
                        wrapper.innerHTML = "Please set the correct openweather <i>appid</i> in
the config for module: " + this.name + ".";
                        wrapper.className = "dimmed light small";
                        return wrapper;
                }
```

```
		if (!this.loaded) {
			wrapper.innerHTML = this.translate("LOADING");
			wrapper.className = "dimmed light small";
			return wrapper;
		}
		if (this.config.onlyTemp === false) {
			this.addExtraInfoWeather(wrapper);
		}
		var large = document.createElement("div");
		large.className = "large light";
		var weatherIcon = document.createElement("span");
		weatherIcon.className = "wi weathericon " + this.weatherType;
		large.appendChild(weatherIcon);
		var temperature = document.createElement("span");
		temperature.className = "bright";
		temperature.innerHTML = " " + this.temperature + "&deg;";
		large.appendChild(temperature);
		wrapper.appendChild(large);
		return wrapper;
	},
	getHeader: function() {
		if (this.config.appendLocationNameToHeader) {
		return this.data.header + " " + this.fetchedLocatioName; }
		return this.data.header;
	},
	notificationReceived: function(notification, payload, sender) {
		if (notification === "DOM_OBJECTS_CREATED") {
			if (this.config.appendLocationNameToHeader) {
				this.hide(0, {lockString: this.identifier});
			}
		}
		if (notification === "CALENDAR_EVENTS") {
		var senderClasses = sender.data.classes.toLowerCase().split(" ");
		if (senderClasses.indexOf(this.config.calendarClass.toLowerCase()) !== -1) {
			var lastEvent =  this.firstEvent;
			this.firstEvent = false;
			for (e in payload) {
				var event = payload[e];
				if (event.location || event.geo) {
				this.firstEvent = event;
				//Log.log("First upcoming event with location: ", event);
				break;
```

```
                                }
                        }
                }
        },
        updateWeather: function() {
                if (this.config.appid === "") {
                        Log.error("CurrentWeather: APPID not set!");
                        return; }
                var url = this.config.apiBase + this.config.apiVersion + "/" +
this.config.weatherEndpoint + this.getParams();
                var self = this;
                var retry = true;
                var weatherRequest = new XMLHttpRequest();
                weatherRequest.open("GET", url, true);
                weatherRequest.onreadystatechange = function() {
                        if (this.readyState === 4) {
                                if (this.status === 200) {
                                        self.processWeather(JSON.parse(this.response));
                                } else if (this.status === 401) {
                                        self.updateDom(self.config.animationSpeed);
                                        Log.error(self.name + ": Incorrect APPID.");
                                        retry = true;
                                } else {
                                        Log.error(self.name + ": Could not load weather.");
                                }
                                if (retry) {
                                self.scheduleUpdate((self.loaded) ? -1 : self.config.retryDelay);
                                }
                        }
                };
                weatherRequest.send();
        },
        getParams: function() {
                var params = "?";
                if(this.config.locationID) {
                        params += "id=" + this.config.locationID;
                } else if(this.config.location) {
                        params += "q=" + this.config.location;
                } else if (this.firstEvent && this.firstEvent.geo) {
                        params += "lat=" + this.firstEvent.geo.lat + "&lon=" +
this.firstEvent.geo.lon
                } else if (this.firstEvent && this.firstEvent.location) {
                        params += "q=" + this.firstEvent.location;
```

```
} else {
                        this.hide(this.config.animationSpeed, {lockString:this.identifier});
                        return;
                }
                params += "&units=" + this.config.units;
                params += "&lang=" + this.config.lang;
                params += "&APPID=" + this.config.appid;
                return params;
        },
        processWeather: function(data) {
                if (!data || !data.main || !data.main.temp) {
                        return;
                }
                this.humidity = parseFloat(data.main.humidity);
                this.temperature = this.roundValue(data.main.temp);
                if (this.config.useBeaufort){
                        this.windSpeed = this.ms2Beaufort(this.roundValue(data.wind.speed));
                }else {
                        this.windSpeed = parseFloat(data.wind.speed).toFixed(0);
                }
                this.windDirection = this.deg2Cardinal(data.wind.deg);
                this.weatherType = this.config.iconTable[data.weather[0].icon];
                var now = new Date();
                var sunrise = new Date(data.sys.sunrise * 1000);
                var sunset = new Date(data.sys.sunset * 1000);
        var sunriseSunsetDateObject = (sunrise < now && sunset > now) ? sunset : sunrise;
                var timeString = moment(sunriseSunsetDateObject).format("HH:mm");
                if (this.config.timeFormat !== 24) {
                //var hours = sunriseSunsetDateObject.getHours() % 12 || 12;
                if (this.config.showPeriod) {
                        if (this.config.showPeriodUpper) {
                //timeString = hours + moment(sunriseSunsetDateObject).format(':mm A');
                timeString = moment(sunriseSunsetDateObject).format("h:mm A");
                } else {
                //timeString = hours + moment(sunriseSunsetDateObject).format(':mm a');
                timeString = moment(sunriseSunsetDateObject).format("h:mm a");
                }
                } else {
                //timeString = hours + moment(sunriseSunsetDateObject).format(':mm');
                        timeString = moment(sunriseSunsetDateObject).format("h:mm");
                        }
                }
```

```
this.sunriseSunsetTime = timeString;
this.sunriseSunsetIcon = (sunrise < now && sunset > now) ? "wi-sunset" : "wi-sunrise";
        this.show(this.config.animationSpeed, {lockString:this.identifier});
        this.loaded = true;
        this.updateDom(this.config.animationSpeed);
        this.sendNotification("CURRENTWEATHER_DATA", {data: data});
},
scheduleUpdate: function(delay) {
        var nextLoad = this.config.updateInterval;
        if (typeof delay !== "undefined" && delay >= 0) {
                nextLoad = delay;
        }
        var self = this;
        setTimeout(function() {
                self.updateWeather();
        }, nextLoad);
},
ms2Beaufort: function(ms) {
        var kmh = ms * 60 * 60 / 1000;
        var speeds = [1, 5, 11, 19, 28, 38, 49, 61, 74, 88, 102, 117, 1000];
        for (var beaufort in speeds) {
                var speed = speeds[beaufort];
                if (speed > kmh) {
                        return beaufort;
                }
        }
        return 12;
},
deg2Cardinal: function(deg) {
        if (deg>11.25 && deg<=33.75){
                return "NNE";
        } else if (deg > 33.75 && deg <= 56.25) {
                return "NE";
        } else if (deg > 56.25 && deg <= 78.75) {
                return "ENE";
        } else if (deg > 78.75 && deg <= 101.25) {
                return "E";
        } else if (deg > 101.25 && deg <= 123.75) {
                return "ESE";
        } else if (deg > 123.75 && deg <= 146.25) {
                return "SE";
```

```
    }
        else if (deg > 213.75 && deg <= 236.25) {
                return "SW";
            } else if (deg > 236.25 && deg <= 258.75) {
                return "WSW";
            } else if (deg > 258.75 && deg <= 281.25) {
                return "W";
            } else if (deg > 281.25 && deg <= 303.75) {
                return "WNW";
            } else if (deg > 303.75 && deg <= 326.25) {
                return "NW";
            } else if (deg > 326.25 && deg <= 348.75) {
                return "NNW";
            } else {
                return "N";
            }
        },
        roundValue: function(temperature) {
            var decimals = this.config.roundTemp ? 0 : 1;
            return parseFloat(temperature).toFixed(decimals);
        }
    });
```

helloworld.js

```
Module.register("helloworld",{

		// Default module config.

		defaults: {
			text: "Hello World!"
		},

		// Override dom generator.
		getDom: function() {
			var wrapper = document.createElement("div");
			wrapper.innerHTML = this.config.text;
			return wrapper;
		}
	} );
```

weatherforecast.js

```
Module.register("weatherforecast",{
	//Default module config.
	defaults: {
		location: false,
		locationID: false,
		appid: "",
		units: config.units,
		maxNumberOfDays: 7,
		showRainAmount: false,
		updateInterval: 10 * 60 * 1000, // every 10 minutes
		animationSpeed: 1000,
		timeFormat: config.timeFormat,
		lang: config.language,
		fade: true,
		fadePoint: 0.25, // Start on 1/4th of the list.
		initialLoadDelay: 2500, // 2.5 seconds delay. This delay is used to keep the
OpenWeather API happy.
		retryDelay: 2500,
		apiVersion: "2.5",
		apiBase: "http://api.openweathermap.org/data/",
		forecastEndpoint: "forecast/daily",
		appendLocationNameToHeader: true,
		calendarClass: "calendar",
		roundTemp: false,
		iconTable: {
			"01d": "wi-day-sunny",
			"02d": "wi-day-cloudy",
			"03d": "wi-cloudy",
			"04d": "wi-cloudy-windy",
			"09d": "wi-showers",
			"10d": "wi-rain",
			"11d": "wi-thunderstorm",
			"13d": "wi-snow",
			"50d": "wi-fog",
			"01n": "wi-night-clear",
			"02n": "wi-night-cloudy",
			"03n": "wi-night-cloudy",
			"04n": "wi-night-cloudy",
			"09n": "wi-night-showers",
			"10n": "wi-night-rain",
```

```
                        "11n": "wi-night-thunderstorm",
                        "13n": "wi-night-snow",
                        "50n": "wi-night-alt-cloudy-windy"
                },
        },
        // create a variable for the first upcoming calendaar event. Used if no location is specified.
        firstEvent: false,
        // create a variable to hold the location name based on the API result.
        fetchedLocatioName: "",
        // Define required scripts.
        getScripts: function() {
                return ["moment.js"];
        },
        // Define required scripts.
        getStyles: function() {
                return ["weather-icons.css", "weatherforecast.css"];
        },
        // Define required translations.
        getTranslations: function() {

        // The translations for the defaut modules are defined in the core translation files.
        // Therefor we can just return false. Otherwise we should have returned a dictionairy.
        // If you're trying to build yiur own module including translations, check out the
documentation.
                return false;
        },
        // Define start sequence.
        start: function() {
                Log.info("Starting module: " + this.name);
                // Set locale.
                moment.locale(config.language);
                this.forecast = [];
                this.loaded = false;
                this.scheduleUpdate(this.config.initialLoadDelay);
                this.updateTimer = null;
        },

        // Override dom generator.
                getDom: function() {
                        var wrapper = document.createElement("div");
```

```
if (this.config.appid === "") {
wrapper.innerHTML = "Please set the correct openweather <i>appid</i> in the
config for module: " + this.name + ".";
                wrapper.className = "dimmed light small";
                        return wrapper;
                        }
                if (!this.loaded) {
                wrapper.innerHTML = this.translate("LOADING");
                wrapper.className = "dimmed light small";
                return wrapper;
                }
                var table = document.createElement("table");
                table.className = "small";
                for (var f in this.forecast) {
                var forecast = this.forecast[f];
                        var row = document.createElement("tr");
                        table.appendChild(row);
                        var dayCell = document.createElement("td");
                        dayCell.className = "day";
                        dayCell.innerHTML = forecast.day;
                        row.appendChild(dayCell);
                        var iconCell = document.createElement("td");
                        iconCell.className = "bright weather-icon";
                        row.appendChild(iconCell);
                        var icon = document.createElement("span");
                        icon.className = "wi weathericon " + forecast.icon;
                        iconCell.appendChild(icon);
                var maxTempCell = document.createElement("td");
                maxTempCell.innerHTML = forecast.maxTemp;
                maxTempCell.className = "align-right bright max-temp";
                row.appendChild(maxTempCell);
                var minTempCell = document.createElement("td");
                minTempCell.innerHTML = forecast.minTemp;
                minTempCell.className = "align-right min-temp";
                row.appendChild(minTempCell);
                if (this.config.showRainAmount) {
                        var rainCell = document.createElement("td");
                if (isNaN(forecast.rain)) {
                        rainCell.innerHTML = "";
                } else {
                        rainCell.innerHTML = forecast.rain + " mm";
                                        }
```

```
rainCell.className = "align-right bright rain";
        row.appendChild(rainCell);
        }
if (this.config.fade && this.config.fadePoint < 1) {
if (this.config.fadePoint < 0) {
        this.config.fadePoint = 0;
        }
var startingPoint = this.forecast.length * this.config.fadePoint;
        var steps = this.forecast.length - startingPoint;
if (f >= startingPoint) {
var currentStep = f - startingPoint;
row.style.opacity = 1 - (1 / steps * currentStep);
                                        }
                                }
                        }
                        return table;
        },
        // Override getHeader method.
        getHeader: function() {
        if (this.config.appendLocationNameToHeader) {
                return this.data.header + " " + this.fetchedLocatioName;
                }
                return this.data.header;
        },
        // Override notification handler.
        notificationReceived: function(notification, payload, sender) {
                if (notification === "DOM_OBJECTS_CREATED") {
                if (this.config.appendLocationNameToHeader) {
                this.hide(0, {lockString: this.identifier});
                        }
                }
        if (notification === "CALENDAR_EVENTS") {
        var senderClasses = sender.data.classes.toLowerCase().split(" ");
if (senderClasses.indexOf(this.config.calendarClass.toLowerCase()) !== -1) {
                var lastEvent =  this.firstEvent;
                this.firstEvent = false;
                        for (e in payload) {
                                var event = payload[e];
                        if (event.location || event.geo) {
                                this.firstEvent = event;
                        //Log.log("First upcoming event with location: ", event);
                                                break;
```

```
                        }
                    }
                }
            },
            updateWeather: function() {
                if (this.config.appid === "") {
                Log.error("WeatherForecast: APPID not set!");
                    return;
                    }
            var url = this.config.apiBase + this.config.apiVersion + "/" +
            this.config.forecastEndpoint + this.getParams();
                var self = this;
                var retry = true;
                var weatherRequest = new XMLHttpRequest();
                weatherRequest.open("GET", url, true);
                weatherRequest.onreadystatechange = function() {
            if (this.readyState === 4) {
                if (this.status === 200) {
            self.processWeather(JSON.parse(this.response));
                } else if (this.status === 401) {
                self.updateDom(self.config.animationSpeed);
                Log.error(self.name + ": Incorrect APPID.");
                retry = true;
                } else {
                Log.error(self.name + ": Could not load weather.");
                                    }
            if (retry) {
                self.scheduleUpdate((self.loaded) ? -1 : self.config.retryDelay);
                                    }
                            }
                    };
                    weatherRequest.send();
                },
            getParams: function() {
                var params = "?";
                if(this.config.locationID) {
                params += "id=" + this.config.locationID;
                } else if(this.config.location) {
                params += "q=" + this.config.location;
                } else if (this.firstEvent && this.firstEvent.geo) {
            params += "lat=" + this.firstEvent.geo.lat + "&lon=" + this.firstEvent.geo.lon
            } else if (this.firstEvent && this.firstEvent.location) {
```

```
                    }       params += "q=" + this.firstEvent.location;
                    } else {
                    this.hide(this.config.animationSpeed, {lockString:this.identifier});
                    return;
                            }
                    params += "&units=" + this.config.units;
                    params += "&lang=" + this.config.lang;
                    params += "&cnt=" + (((this.config.maxNumberOfDays < 1) ||
(this.config.maxNumberOfDays > 16)) ? 7 : this.config.maxNumberOfDays);
                    params += "&APPID=" + this.config.appid;
                    return params;
                    },
                    processWeather: function(data) {
                    this.fetchedLocatioName = data.city.name + ", " + data.city.country;
                    this.forecast = [];
                    for (var i = 0, count = data.list.length; i < count; i++) {
                            var forecast = data.list[i];
                            this.forecast.push({
                            day: moment(forecast.dt, "X").format("ddd"),
                            icon: this.config.iconTable[forecast.weather[0].icon],
                                    maxTemp: this.roundValue(forecast.temp.max),
                                    minTemp: this.roundValue(forecast.temp.min),
                                    rain: this.roundValue(forecast.rain)
                                    });
                            }
                            //Log.log(this.forecast);
                                    this.show(this.config.animationSpeed,
                    {lockString:this.identifier});
                            this.loaded = true;
                            this.updateDom(this.config.animationSpeed);
                    },
                    scheduleUpdate: function(delay) {
                            var nextLoad = this.config.updateInterval;
                            if (typeof delay !== "undefined" && delay >= 0) {
                                    nextLoad = delay;
                            }
                            var self = this;
                            clearTimeout(this.updateTimer);
                            this.updateTimer = setTimeout(function() {
                                    self.updateWeather();
                            }, nextLoad);
                    },
```

```
        ms2Beaufort: function(ms) {
                var kmh = ms * 60 * 60 / 1000;
                var speeds = [1, 5, 11, 19, 28, 38, 49, 61, 74, 88, 102, 117,
1000];
                for (var beaufort in speeds) {
                        var speed = speeds[beaufort];
                        if (speed > kmh) {
                                return beaufort;
                                }
                        }
                        return 12;
                },

                roundValue: function(temperature) {
                        var decimals = this.config.roundTemp ? 0 : 1;
                        return parseFloat(temperature).toFixed(decimals);
                }
        });
```

REFERENCES

[1] "Raspberry Pi documentation," [Online]. Available: https://www.raspberrypi.org/documentation/. [Accessed 23 August 2016].

[2] "Electron documentation," [Online]. Available: http://electron.atom.io/docs/tutorial/about/. [Accessed 1 September 2016].

[3] "Set up Kiosk web browser," [Online]. Available: https://www.raspberrypi.org/forums/viewtopic.php?f=36&t=122444. [1 September 2016].

[4] "One Way Mirror Design," [Online]. Available: https://en.wikipedia.org/wiki/One-way_mirror. [Accessed 24 October2016].

[5] "Idea of Smart Mirror" [Online]. Available: http://michaelteeuw.nl/post/80391333672/magic-mirror-part-i-the-idea-the-mirror. [Accessed 1 September 2016].

[6] "Automatically connect Raspberry Pi to a Wi-Fi network," [Online]. Available: http://weworkweplay.com/play/automatically-connect-a-raspberry-pi-to-a-wifi-network/. [Accessed 5 September 2016].

[7] "Node.js," [Online]. Available: https://nodejs.org/en/docs/. [Accessed 19 October 2016].

[8] "An introduction to GPIO and physical computing on Raspberry Pi," [Online]. Available: https://www.raspberrypi.org/documentation/usage/gpio/. [Accessed 23 August 2016].

[9] "Linux Fundamental," [Online]. Available: linux-training.be/linuxfun.pdf [Accessed 24 August 2016].

[10] "NOOBS setup," [Online]. Available: https://www.raspberrypi.org/help/noobs-setup/. [Accessed 23 August 2016].

[11] "Installation of Node js," [Online]. Available: http://thisdavej.com/beginners-guide-to-installing-node-js-on-a-raspberry-pi/. [Accessed 3 November 2016].

[12] "One way mirror film," [Online]. Available: http://www.reflectiv.com/site/en_index.php [Accessed 16 July 2016].

[13] " HTML introduction," [Online]. Available: http://www.w3schools.com/html/html_intro.asp. [Accessed 3 November 2016].

[14] " HTML," [Online]. Available: https://en.wikipedia.org/wiki/HTML. [Accessed 21 January 2016].

15] " Cascading Style Sheets," [Online]. Available: https://en.wikipedia.org/wiki/Cascading_Style_Sheets. [Accessed 3 November 2016].

[16] " CSS introduction," [Online]. Available: http://www.w3schools.com/css/css_intro.asp. [Accessed 3 November 2016].

[17] "JavaScript introduction," [Online]. Available: http://www.w3schools.com/js/js_intro.asp. [Accessed 3 November 2016].

[18] "JavaScript," [Online]. Available: https://www.javascript.com/about [Accessed 3 November 2016].

[19] "PHP introduction," [Online]. Available: http://www.w3schools.com/php/php_intro.asp. [Accessed 3 November 2016].

[20] "PHP," [Online]. Available: https://en.wikipedia.org/wiki/PHP. [Accessed 3 November 2016].

[21] "Bootstrap (front-end framework)," [Online]. Available: https://en.wikipedia.org/wiki/Bootstrap_ (front-end framework). [Accessed 3 November 2016].

[22] "GPIO - Raspberry Pi documentation," [Online]. Available: https://www.raspberrypi.org/documentation/hardware/raspberrypi/gpio/README.md. [Accessed 28 October 2016].

[23] "Mirror Film," [Online]. Available: http://www.reflectiv.com/site/en_catalogue_liste.php?main_category=3. [Accessed 15 November 2016].

[24] "Facial Recognition," [Online]. Available: http://www.ex-sight.com/technology.htm. [Accessed 18 November 2016].

[25] "How voice control works,"]. Available: http://electronics.howstuffworks.com/gadgets/high-tech-gadgets/speech-recognition1.htm [Accessed 27 November 2016].

[26] Build a Smart Mirror"," [Online]. Available: http://lifehacker.com/build-a-magic-mirror-with-a-raspberry-pi-and-an-old-mon-1750468358 [Accessed 23 July 2016].

[27] "IoT World m\Market," [Online]. Available: https://www.forbes.com/sites/louiscolumbus/2016/11/27/roundup-of-internet-of-things-forecasts-and-market-estimates-2016/#f8584d292d51 [Accessed 28 December 2016].

[28] "How to remote desktop onto the Raspberry Pi using Xming," [Online]. Available: http://www.codesynthesis.co.uk/tutorials/how-to-remote-desktop-onto-the-raspberry-pi-using-xming. [Accessed 28 August 2016].

[29] "SSH (Secure Shell) - Raspberry Pi documentation," [Online]. Available: https://www.raspberrypi.org/documentation/remote-access/ssh/. [Accessed 3 November 2016].

[30] "Remotely accessing the Raspberry Pi via SSH – console mode," [Online]. Available: http://www.modmypi.com/blog/remotely-accessing-the-raspberry-pi-via-ssh-console-mode. [Accessed 11 December 2016].

[31] "What is a Raspberry Pi?," [Online]. Available: https://www.raspberrypi.org/help/what-is-a-raspberry-pi/. [Accessed 3 June 2016].

[32] "Raspberry Pi," [Online]. Available: https://en.m.wikipedia.org/wiki/Raspberry-Pi. [Accessed 3 June2016].

[33] "Setting up your Raspberry Pi," [Online]. Available: https://www.raspberrypi.org/help/quick-start-guide/. [Accessed 23 August 2016].

[34] "Access your Raspberry Pi over the internet," [Online]. Available: https://www.raspberrypi.org/documentation/remote-access/access-over Internet/internetaccess.md. [Accessed 29 August 2016].

[35] "VNC (Virtual Network Computing)," [Online]. Available: https://www.raspberrypi.org/documentation/remote-access/vnc/README.md. [Accessed 29 August 2016].

[36] "SFTP," [Online]. Available: https://www.raspberrypi.org/documentation/remote-access/ssh/sftp.md. [Accessed 29 August 2016].

[37] "SCP (Secure Copy)," [Online]. Available: https://www.raspberrypi.org/documentation/remote-access/ssh/scp.md. [Accessed 29 August 2016].

[38] "FTP," [Online]. Available: https://www.raspberrypi.org/documentation/remote-access/ftp.md. [Accessed 29 August 2016].

[39] "RSYNC," [Online]. Available: https://www.raspberrypi.org/documentation/remote-access/ssh/rsync.md. [Accessed 29 August 2016].

[40] "Setting up a web server on a Raspberry Pi," [Online]. Available: https://www.raspberrypi.org/documentation/remote-access/web-server/README.md. [Accessed 29 August 2016].

[41] "Apache HTTP server," [Online]. Available: https://en.wikipedia.org/wiki/Apache_HTTP_Server. [Accessed 29 August 2016].