“Smart Mirror using Raspberry PI”

A Project

Submitted in partial fulfillment of the requirement for the award of Degree of Bachelor of Engineering in Computer Engineering Discipline

Submitted To

* * * *

KAVAYITRI BAHINABAI CHAUDHARI NORTH MAHARASHTRA UNIVERSITY, JALGAON

Submitted By:

Ms. Patil Vaibhavi Chudaman	**Exam Seat No.11111**
Ms. Patil Priyanka Hemant	**Exam Seat No.22222**
Ms. Patil Prerna Manilal	**Exam Seat No.33333**
Ms. Patil Divya Pramod	**Exam Seat No.44444**
Ms. Patil Jagruti Vishram	**Exam Seat No.55555**

Under the guidance of:

Prof. V. T. Patil

DEPARTMENT OF COMPUTER ENGINEERING

P.S.G.V.P. MANDAL’S

D.N.PATEL COLLEGE OF ENGINEERING

SHAHADA, DIST- NANDURBAR (M.S.)

ACADEMIC YEAR 2019-20

P.S.G.V.P. MANDAL'S

D. N. PATEL COLLEGE OF ENGINEERING

SHAHADA, DIST- NANDURBAR (M.S.)

CERTIFICATE

This is to certify that

Ms. Patil Vaibhavi Chudaman	***Exam Seat No.11111***
Ms. Patil Priyanka Hemant	***Exam Seat No.22222***
Ms. Patil Prerna Manilal	***Exam Seat No.33333***
Ms. Patil Divya Pramod	***Exam Seat No.44444***
Ms. Patil Jagruti Vishram	***Exam Seat No.55555***

Has satisfactorily completed Project-II entitled

"Smart Mirror using Raspberry PI"

As prescribed by Kavayitri Bahinabai Chaudhari North Maharashtra University, Jalgaon as a part of syllabus for the partial fulfillment in Bachelor of Computer Engineering for academic year 2019-20.

GUIDE
Prof. V. T. Patil

H.O.D.
Prof. V.S. Mahajan

EXAMINER
_____

PRINCIPAL
Prof. Dr. N.J. Patil

ACKNOWLEDGMENT

The Acknowledgement is just like a drop in the ocean of the deep sense of gratitude within our heart for people who helped us out of most embarrassing part of our life when we were standing at most difficult step towards our dream of life.

Many people have contributed to the success of this project work. Although a single sentence hardly suffices, we would like to thank Almighty God for blessing us with his grace. We extend our sincere and heartfelt thanks to **Prof. V. S. Mahajan**, Head of Department, Computer Engineering, for providing us the right ambience for carrying out this work.

We are grateful and sincerely appreciate the effort of our respected project in-charge **Prof. R.A. Shaikh, Prof. D.B. Shukla & Prof. A.I. Pathan** who acted as a fulcrum for us and supported us during the ups and downs of our project. We are profoundly indebted to our project guide **Prof. V.T Patil** for innumerable acts of timely advice, encouragement and we sincerely express our gratitude to him.

We express our immense pleasure and thankfulness to all the teachers and staff of the Department of Computer Engineering and Information Technology for their cooperation and support.

Smart Mirror

Vaibhavi C. Patil

Priyanka H. Patil

Prerna M. Patil

Divya P. Patil

Jagruti V. Patil

B.E. Comp.

ABSTRACT

The project describe the design, construction and working of smart mirror. Smart mirror is one of the application of Raspberry PI. The raspberry PI stays at back scene and control the data display on the mirror. While looking at mirror you can look at various notification from social sites as well as news, weather forecast and more things. The smart mirror idea is to integrate technology seamlessly into people's lives by putting it where everyone's routine eventually collides. The goal of the smart mirror is to increase the user's productivity by saving time. The smart mirror provides a near effortless experience that allow the user to just walk up and be greeted with information they would typically need another device. Despite the fact this information can be found on the user's devices. The smart mirror has the necessary applications and features needed for the time efficiency focused device. It also provides real time interaction users. The Smart Mirror CPU is the Raspberry pi 3 computer and the framework that retrieves the data from the web through the Wi-Fi connectivity. Through facial recognition model , smart mirror can identify the user.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

Abbreviation	Literal Translation
DB	Database
ERD	Entity-Relationship Diagram
IDE	Integrated Development Environment
LOC	Lines of Code
SDLC	Software Development Life Cycle
SQL	Structured Query Language

CHAPTER 1
INTRODUCTION

1.1. INTRODUCTION TO PROJECT DOMAIN

In this world everyone needs a comfort life. Modern man has invented different technology for his purpose. In today‟s world, people need to be connected and they are willing to access the information easily. Whether it is through the television or internet, people need to be informed and in touch with the current affairs happening around the world. The Internet of Things means interconnection via the internet of computing devices embedded in everyday objects, enabling them to send and receive data. The Internet of Things with its enormous growth widens its applications to the living environment of the people by changing a home to smart home. Smart home is a connected home that connects all type of digital devices to communicate each other through the internet. Our lifestyle has evolved in such a way that optimizing time is the most important thing. Our work is based on the idea that we all look at the mirror when we go out, so why wouldn‟t the mirror become smart**.** A common approach for building a smart mirror is to use a high quality one-way glass, a LCD monitor, a frame to hold the glass and monitor, and a web browser with python to provide the software features and drive the display.

This project has to be developed with the idea of making home smart to save time . The Internet transformed our lives by connecting us more easily to information and other people in the virtual world. The state of innovation currently is to provide more information with less interaction to get it. The device that has been researched and designed is called “Smart Mirror”. It is a wall mounted mirror which displays relevant items to the user such as weather, time,date,temperature,humidity and news and other fields of interest. IoT emerged the idea of remotely monitoring objects through the Internet. When it comes to our home, security is crucial issue to the general public. For enhancing the security of home this framework is used by owner of the house. Assume you are not at home and a thief enters your home then this framework will give a caution through alert message. When thief enters the home, PIR sensor will detect the movement and gives the owner alert message. Wireless Home security and Home automation are the dual aspects of this project. The currently built prototype of the system sends alerts to the owner over message using the Internet if any sort of human movement is sensed near the mirror.

1.2 PROBLEM IDENTIFICATION

1.2.1 Problem Definition

The major problem is with existing mirror is it shows only any object kept in front of that or face of human. People wastes their lot of time standing in front of mirror then after of they read news so all this is time consuming. So we are developing a project which overcomes to time wastage. Staying connected with new information is both important for entertainment and daily life. With such a variety of options, there is difficulty in following all of your data streams. Often, during your day, you may end up in a position where it is inconvenient, or even impossible, to take out your phone or computer and check the newest update. You cannot commit to a slower interaction. You need a display to glance at, with the information you need ready to go. However, aesthetics are just as important as displaying information. Keeping an extra computer in your bathroom or hall would be inconvenient, and would not fit well with the look of a modern room. A sleek, simple display, easy for an average consumer, is a necessity in todays world.

1.2.2 Existing Systems

In the existing system, we can interface Raspberry with a camera and display unit. It can display the date time and calendar also.

1.2.3 Need for New System

There are several products in the market that attempt to be your attractive hub of daily information. The Amazon Echo and the upcoming Google Home present themselves as a small speaker that can relay information through sound. You can request news or music, fulfilling your need to obtain media content in a hands-free manner. However, not all data is suitable for conveyance by voice. Both designs lack the key ability to convey information visually having the news read to you is convenient, but many prefer reading the news at their own pace. A smart display would be a product that would be able to answer all of these concerns, while staying smoothly modern. The Nest thermostat has a small display for information. However, it is not intended for interaction. The interface can be clunky, and not something an average consumer would interact with on a day-to-day basis.

Our solution is an open platform for discrete display development. We offer an aesthetically pleasing mirror, with a hidden smart display underneath. With a generic display, the mirror can be built to any size so the information can be both in your face while showing you your face. By creating a platform open to modification, developers will also be add new functionality at their own pace. This will allow

our display to be a tailorable and adaptable platform. A web application provides the interface that the user sees and interacts with.

1.3 PROJECT 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. Users are able to create a profile and customize the visual interface to display what specific data feeds they want.

The objective is to provide a natural interface in the home environment for accessing various services such as location based weather, time, calendar etc. It includes downloading the Raspbian operating system based on Debian and extracting the image on SD card, inserting the card in the Raspberry PiSD slot and then performing the required steps.

- The working is based on raspberry-pi.

- That screen can be an Android tablet or a computer monitor.

- The project which would collect real world machine data.

1.4 PROPOSED SYSTEM AND METHODOLOGY

1.4.1 System Overview

The aim of designing this model is to create an interactive interface which can be conveniently used in home environment as well as commercial space. Various services like weather, calendar, news stock updates etc. can be accessed and controlled. The Raspberry Pi 3 is connected to a Monitor via HDMI cable and a webcam is attached using a universal serial bus. Raspberry Pi is powered up using a 5V/2A DC supply.

We plan to deliver a working model of Smart mirror by using raspberry pi 3 for smart homes of future as well as commercial uses. The device will look like a normal reflective mirror but would have a monitor attached on one side. A special TFT touch screen is used for this purpose as it can act as normal reflective mirror when the monitor is off and can also display various data as soon as the monitor is turned on. This will thus serve both the purposes.

1.4.2 METHODOLOGY

- **Smart Mirror As A Mirror**

The device is look like a regular mirror but would have a screen inside. A smart mirror is basically a mirror with a screen behind it with high concentration of aluminium content. That screen can be an Android tablet or a computer monitor.

- **Smart Mirror As An Information System**

The project which would collect real world machine data such as location based latest news and headlines, weather reports, and as well as show us the local time. The data would be transmitted from the machine and would be managed in a central database and would be managed by the Raspberry Pi. The Smart Mirror implemented as a personalised digital device equipped with peripherals such as Raspberry PI, speakers, LCD Monitor covered with a sheet of reflective one way mirror provides one of the most basic common amenities such as weather of the city, latest updates of news and headlines and local time corresponding to the location. The mirror display is provided by a flat LED display monitor which displays all the necessary information which are useful for the user. The mirror also provides a picture-in-picture sub-display to facilitate the display of services.

- **Smart Mirror As Security System**

When there is nobody in home it can be switched into security system by using VNC viewer to detect human presence. When someone enters into room ,PIR sensor will detect the movement of the person when he passes by the mirror and capture the image and stores it in the drop box .Also informs the owner by updating captured image in the dropbox ,by this way smart mirror system can also be used as a security system.

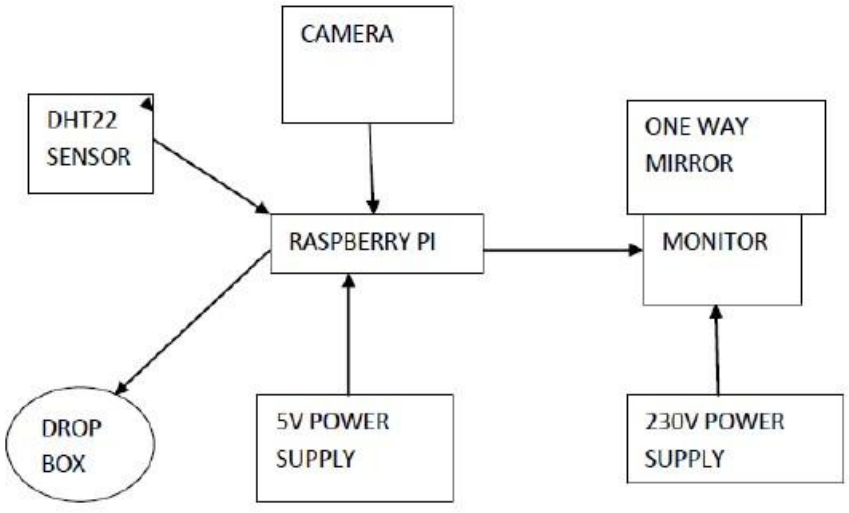


Fig 1.1 Block diagram of smart mirror

1.4.3 System Architecture

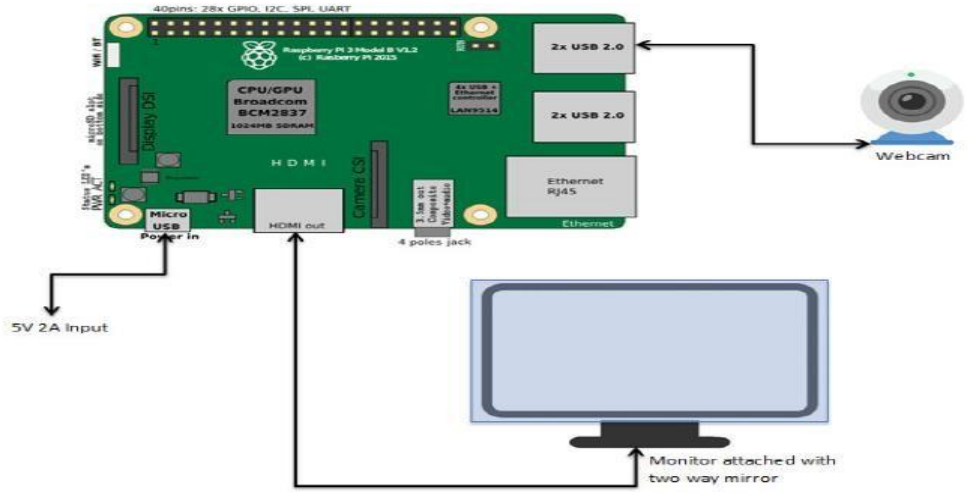


Fig 1.2 System architecture of smart mirror

1.4.4 Modules

Rasberry PI

Raspberry Pi 3 acts as the main control centre for this proposed model. The Raspberry Pi is equipped with a micro SD card which can be loaded with operating systems like Raspbian or Windows 10 IoT core. After the OS is running the Magic Mirror code will can be implemented on it to run the application. The Monitor will be getting input from RPi using HDMI cable.

PI Camera

A simple USB powered camera was used for facial recognition. It was used as an input device to take image as input.

TFT Touch Screen

This screen was placed behind the mirror which was used to display the desired information to the user.

Speaker

The audio output is taken from the audio jack of the raspberry pi. The audio output of the videos or any multimedia is amplified using an audio amplifier. The audio output is taken out from the speaker which is also interfaced with the raspberry pi controller.

1.4.5 Procedure

1.4.5.1 Circuit Diagram

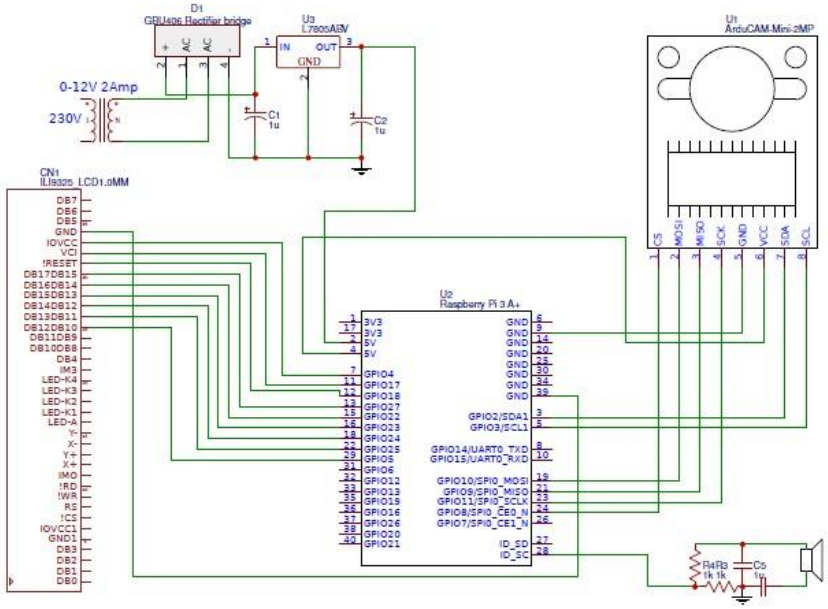


Fig 1.3 Circuit diagram for smart mirror

1.4.5.2 Circuit Description

To drive all the components in this circuit 5V dc and 12V dc are required. The mains give the 230V ac. The 230V ac is stepped down to 12V ac by using step down transformer. Then the output is given to the full wave rectifier. The rectifier eliminates the negative peak voltage of the input voltage. The output of the rectifier is the pulsating dc. The error pulses are eliminating by using capacitor filter. Then the output at the parallel of the capacitor is the 12V dc. But the Micro Controller works on 5V dc .To convert the 12V dc into 5V dc a regulator is used. The output of the regulator is constant irrespective of the input voltage. An LED is connected in between the regulator and the controller to indicate whether the power is on or off. The power supply is provided to the pin number 2 which is VCC pin of the raspberry pi as shown in circuit diagram.

The hardware of the system consists of a camera interfaced with the raspberry pi as shown in the circuit diagram above. The VCC pin of the camera module is connected to the pin number 4 of the raspberry pi controller which has a 5V supply. And the ground supply is provided by pin number 9 of the raspberry pi controller as shown in the above circuit diagram of the system. This pi camera is used to view the face of the user who is using the smart mirror designed.

The system consists of a 7 inch touch screen LCD display act as mirror as shown in the circuit diagram of the system. This display acts as mirror as well as interactive display used to display the data required by the user like weather information, news headlines etc. This display is interfaced with the raspberry pi as shown in the circuit diagram of the system. The data pins of the display are connected to the GPIO number 12, 13, 15, 16, 18, 22 and 29 as shown in the circuit diagram of the system. The VCC and IOVCC pins of the display are connected to GPIO pins 7 and 11 whereas GND terminal is connected to the GND pins in the raspberry pi board as shown in the circuit diagram of the system.

The system also outputs the data in audio form through speaker. This speaker is interfaced with the pi controller as shown in the circuit diagram of the system. This speaker is connected to the GPIO pin 22 of the raspberry pi controller. This speakers are used to provide audio of the multimedia played by the user as per the requirement.

1.4.5.3 Working Principle

Proposed model can perform various functions described as follows:

i) Work as a normal reflective mirror so that the user can use it as a regular mirror.

ii) A two way mirror which can function both as reflective and see through mirror is attached to a TFT screen. This provides two major functionalities ie. Mimicking a normal mirror as well as working as a display for real time data updates.

iii) Personalized data and information services: Anyone using this mirror will be able to get real time updates of news and headlines, date, time, weather updates as well as other reports of our particular interests.

The webcam is attached to the Raspberry pi using the universal serial bus to detect user‟s face using OpenCV. This will help in setting up the personalized profiles for different users and managing them afterwards. The user interface will be show the data on the mirror and the empty space in between will accommodate the reflection of the user.

A LCD was attached to display with the frame and connect the LCD display with the raspberry pi via HDMI cable. USB microphone and USB web were then connected with the raspberry pi. Finally, the power source for both the raspberry pi and LCD display is established. As the raspberry pi has its own operating system, the Raspbian operating system is booted for the project into the raspberry pi. It is updated and upgraded to increase the CPU speed. The default version of the operating system

consisted of an older version of the Node which did not consist of NPM. So, the Node v5.1.1 was reinstalled which included the NPM v1.9. Next, pip was installed which was a package installer of python and it helped to install numerous packages. An OpenCv, a library of programming functions was used which focused on real time computer vision. Now moving to the coding and designing part, HTML, CSS and JavaScript were used to display the information on the
LCD display in such a way that only the information‟s would appear before the user. And for the AI and other modules, python programming language was used. The key features of our design are:

(1) Time and Date: The time of the CPU used (Raspberry Pi) in the mirror is shown.

(2) Calendar: The international calendar in the mirror as well as the upcoming holidays is integrated in the system. The help of open source website to fetch the API of the calendar is taken.

(3) News: The news functionality was integrated, which will show the RSS feed of any newspaper of the world.

- Algorithm For Information System

Step 1: Switch on the power supply.

Step 2: Get the date, time, and weather details from predefined from URL.

Step 3: Get the news from www.TimesofIndia.com

Step 4: In code section write down all the compliments to be displayed on mirror.

Step 5: Display it on mirror via LCD monitor

Step 6: Switch to thief detection mode using VNC viewer.

Step 7: Switch off the power supply when it is of no use.

- Algorithm For Thief Detection

Step 1: Start

Step 2: Setup the Camera

Step 3: Check whether PIR sensor output is high or low

Step 4: If it is low, go to step 3.

Step 5: If it is high, camera is turned ON.

Step 6: Image is captured and stored on raspberry pi.

Step 7: Check for Wi-Fi connection.

Step 8: If it is connected image is uploaded to dropbox.

Step 9: Notification is updated in dropbox.

1.5 APPLICABILITY

Each software feature of the mirror will be designed from scratch and placed appropriately within the UI. The layout of the applications is actually one of the most important aspects of this project. The reason for this is that, if the mirror is not well thought out, easy to use, and does not intrude on daily use of a mirror, then users are unlikely to enjoy the product or utilize it daily. Due to this, much thought has gone into the layout of the applications on the mirror and more changes will likely be made as prototyping continues. The format will allow a wealth of different information to be presented but not in much depth.

The Widget class is the basis of information display for the Smart Mirror. A Widget is a selfcontained module that injects HTML content into a specified location. A primary loop function is called periodically based upon a specified refresh rate. This is handled by the client main. The loop function is modeled to function similarly to the loop used in Raspberry Pi development.

Clock

The clock is one of the simpler components of this project.

Weather

The weather app will display basic weather information including the temperature at your current location along with the high, low, and chance of precipitation. A simple graphic, such as a sun or raincloud, representing current weather conditions will be displayed as well. This information will be updated every hour, providing the user with an accurate estimation of what to expect at any time of the day. This is an important feature in assisting the user in planning for their day as they get ready in the morning. This feature will be displayed on the mirror at all times.

Calendar

The calendar will be implemented as a list of events for the current day. An example of this can be found in the Day view of Google Calendar. The events on the calendar will be updated hourly

to ensure the user is always up to date on their events. As the calendar will implement a day view of events, the current time will be indicated by highlighting the hourly row that corresponds to the current time.

News

In order to provide the user with a few news events, a simple news feed will be displayed to provide three headlines from world news. After researching various different news feed APIs, we have decided to use CNN‟s World News RSS feed. The other news outlets we considered included USA Today, The New York Times, and Associated Press. Our decision is based on simplicity of use and straightforward parsing ability.

CHAPTER 2
LITERATURE SURVEY

2.1 RELATED WORK

The design and the development of an interactive multimedia futuristic Smart Mirror with artificial intelligence for the ambient home environment as well as for commercial uses in various industries. The project which would collect real world machine data and the data would be transmitted from the machine and would be managed by the Raspberry Pi. The Smart Mirror implemented as a personalised digital device equipped with peripherals such as Raspberry PI, speakers, LCD Monitor covered with a sheet of reflective one way mirror provides one of the most basic common amenities such as weather of the city, latest updates of news and headlines and local time corresponding to the location.

Smart Reflect is a similar work carried out by the students of MacEwan University. It basically aimed at providing a platform that can facilitate the development of smart mirror. It acts an alternative option than the sandbox environment. It is light in functioning as compared to already present platforms. Its major advantage is its multiple language and environment support so as to ease end user efforts.

The AwareMirror is an augmented display that is placed in the bathroom for presenting personalized information to the user. It detects the position of a person in the bathroom using a proximity sensor and identifies her/him from the usage of a toothbrush. It provides useful information such as closest schedule, transportation information, and the weather forecast. The mirror is constructed by attaching an acrylic board in front of a monitor. A slider sensor on the mirror is used to navigate the information presented on the mirror. Although it attempts to provide an intuitive interface, it has some limitations that may restrict it from wider usage. For example, the state-of-use of a toothbrush for identifying a person might not provide accurate states to personalize information. Also, the use of magic mirror restricts dark colors from going through it; and hence, requires special attention to the color of the contents to be displayed. In our case there is no such issues as we use a touch screen to mimic a mirror-like interface through the use of touchscreen and video technology. Another project named MagicMirror as carried out by students of NUS, They created a magic mirror which can recommend you appropriate clothing in the morning while you get ready. The Magic mirror model will scan the user and then based on the particular occasion or event it will recommend most suitable

attire and other styling options. The events can be retrieved from user‟s social media account or can be added to the calendar manually.

Philips HomeLab acts as testing platform for interactive and automated home environment. The Philips Hue is one such example of smart lighting which can be controlled using mobile application. Another example is of an Interactive Mirror which can installed in room or washroom to get personalized services depending on the end user. Children can customize it view cartoons, adults can get live news feeds and updates on weather, traffic etc.

An earlier work presents i-mirror that attempts to include information services within the mirror interface as a natural way of providing interactive experiences to people. The i-mirror uses dedicated optical system including a video camera, magic mirrors, and a video projector to imitate a mirror. Its use of magic mirror to superimpose an image to the mirror is similar to the one in AwareMirror. Three potential uses of the i-mirror have been explored such as the one with ability to show images in the dark; one with the capability of providing younger/older looks; and the one with memory.

2.2 THEORETICAL BACKGROUND

The innovation and research work in the field of Artificial intelligence, Machine learning, Internet of things has brought a massive change in the technology we use and paved the way for Smart environment. Kevin Ashton published an article in the RFID Journal in 2009[1] in which he talked about the capabilities of things that a computer can perform if it knew everything there was to know about things by the means of gathering data and track everything. They would be able to reduce loss, waste and cost. We would be able to get updates about machines know when they needed replacing or repairing. There is need to empower the computer by automating them to see them in their full glory This is exactly what has happened after the development in field of IoT.

Also, An efficient, convenient and secure home automation environment can be achieved using collective application of AI and IoT. Artificial intelligence has already received attention for assisted living purposes. Apart from this entertainment, automated home environment, official space and home learning are also affected due to advancement in AI and IoT. Various companies are now launching products aimed at automating the day to day activities we do in our home, offices or industries. Nest Labs launched learning thermostat in 2010 which used to detect smoke and carbon monoxide detector and later was redesigned to sense and control temperature in the house. Magic Mirror also provides solution to our daily routine of getting ready. It uses the concept of Internet of

things to embed various chores like reading newspaper, getting stock updates, traffic updates etc. on a display that will also work as a normal reflective mirror simultaneously.

CHAPTER 3
ANALYSIS

3.1 FEASIBILITY STUDY

Feasibility study is a test of a proposed system according to work ability, impact on the organization‟s ability to meet user needs and effective use of resources. Feasibility study is performed by, considering the factors such as development cost, operating cost, response time, development time, accuracy and reliability. Not all requested projects are feasible. We compare the proposed system with the existing system. In feasibility study we develop more than one way to solve the existing system problems. From this we can select the feasible one and then we prepare detailed description our Feasibility study includes studying the available general purpose.

The objective of feasibility study is to determine whether the proposed system can be developed with available resources. It is the high level capsule version of the entire requirement analysis process. There are two steps to be followed for determining feasibility study of proposed systems.

- Technical feasibility
- Economical feasibility

3.1.1 Technical Feasibility

The system is developed using python :

- **Simple and easy to learn :** Python is a simple and minimalistic language. It allows you to concentrate on solution to the problem rather than the syntax that is the language itself. As you will see, python is extremely easy to get started with. Python has an extraordinary simple syntax as already mentioned.

- **Object Oriented :** Python supports procedure oriented programming as well as object oriented programming. In procedure oriented languages, the program is built around procedures and functions which are nothing but reusable pieces of programs. In object oriented languages, the program is built around objects which combines data and functionality. Python has a very powerful but simple way of doing object oriented programming, especially, when compared to languages like C++ or Java.

- **Extensible :** If you need a critical piece of code to run very fast, you can achieve this by writing that piece of code in C, and then combine that with your Python program.

3.1.2 Economic Feasibility

The system that we are developing is a very cost effective because of following mentioned points:

- The system is developed with Python technology,which is free of cost.
- The system can be called as economically feasible as it has been written in python and python being platform independent we don"t have to take efforts/invest resource or money in redeveloping it for various other platforms.

3.2 PROJECT PLANNING AND SCHEDULING

Project planning involves plotting project activities against a time frame. We exercised a lot to plan project according OO project metrics. The aim of these processes is to ensure that various Project tasks are well coordinated and they meet the various project objectives including timely completion of the project. There are two popular tools to plot the project planning:

- Timeline Chart
- Project Table

3.2.1 Team Structure

Team structure addresses the issue of organization of the individual project teams. Our project team consists of ***5*** members; the efforts assignment to each team member are given the project table, the role of each member is as below:

Table 3.1 Team Structure, Roles & Details

Sr. No.	Name of Team Member	Role in Project-I	Role in Project-II	Email ID
1.	Vaibhavi Chudaman Patil	Designer	Coder	vaibhavipatil200798@gmail.com
2.	Priyanka Hemant Patil	Designer	Coder	patilpriyanka9866@gmail.com
3.	Prerna Manilal Patil	Documentor	Documentor	prernapatil652@gmail.com
4.	Divya Pramod Patil	Documentor	Tester	patildivya8907@gmail.com
5.	Jagruti Vishram Patil	Documentor	Analyst	jagrutipatil8299@gmail.com

3.2.2 Timeline Chart

When creating a software project schedule, the planner begins with a set of tasks. A timeline chart can be developed for the entire project. Alternatively, separate charts can be developed for each project function or for each individual working on the project.It is a way of displaying a list of events in chronological order, sometimes described as a project artifact. It is a special type of bar chart where each bar represents an activity. The bars are drawn along a timeline. The length of each bar is proportional to the duration of time planned for the corresponding activity. When multiple bars occur at the same time on the calendar, task concurrency is implied.

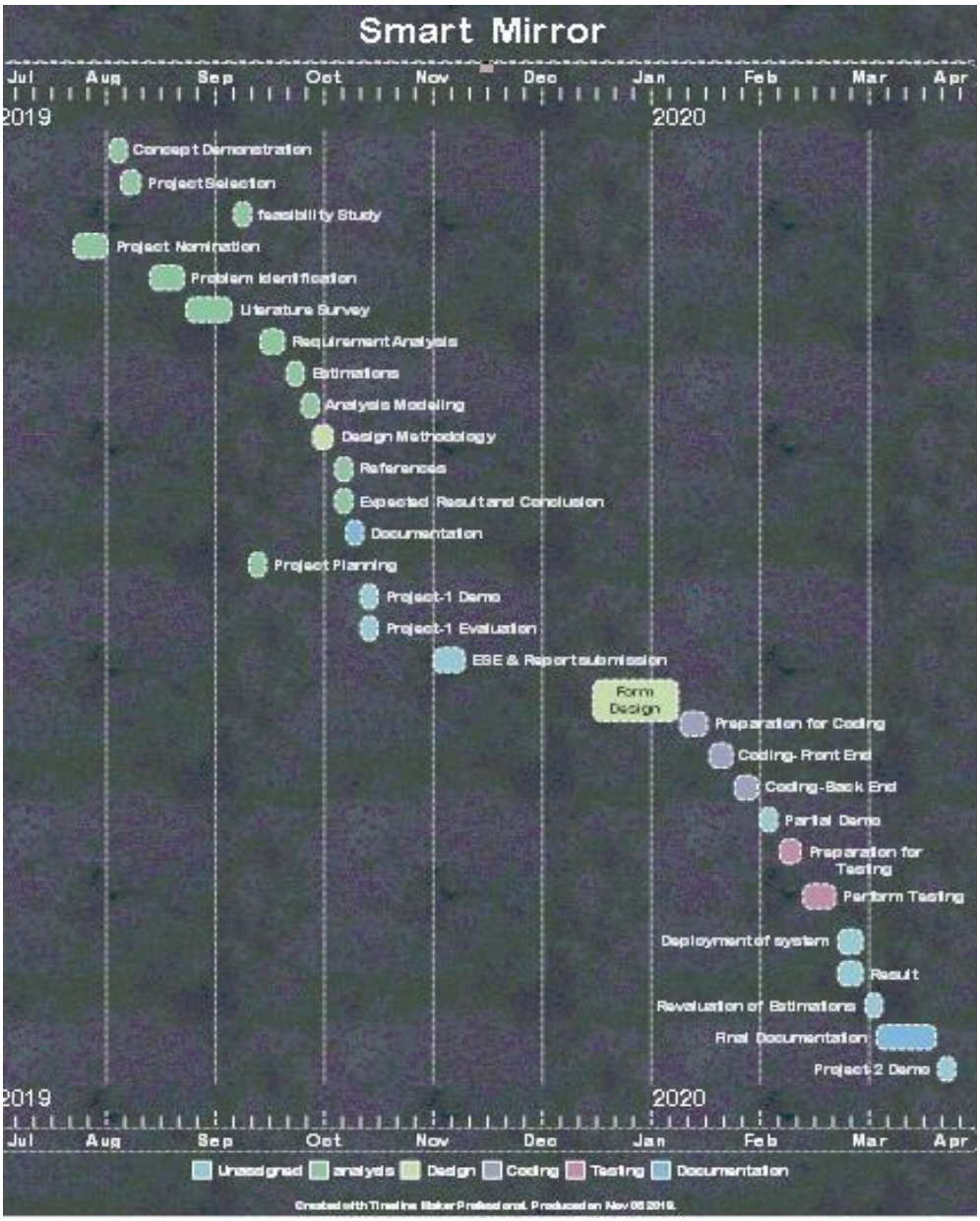


Figure 3.1 Timeline Chart

3.2.3 Project Table

Project table is a tabular listing of all project tasks, their planned and actual start- and end-dates, and a variety of related information. Used in conjunction with the timeline chart, project tables enable the project manager to track progress.

Table 3.2 Project Table

Event Name	Start Date	Actual Start Date	End Date	Actual End Date	Effort Assignment
Problem Definition -Collecting detailed problem definition of the system to be implemented	Jul 13 2019	Aug 12 2019	Aug 01 2019	Aug 22 2019	ALL
Initiation(Literature Survey) -Visiting different websites. -Studying existing system with its limitation -Going through Journals, magazines -Studying the reference books	Aug 23 2019	Aug 24 2019	Sept 05 2019	Sept 19 2019	ALL
Feasibility Study -Techincal& Economical feasibility	Sept 06 2019	Sept 06 2019	Sept 09 2019	Sept 19 2019	ALL
Project Planning & Scheduling -Prepare complete project plan: decide Roles, Schedule of events, Deadlines	Sept 10 2019	Sept 11 2019	Sept 12 2019	Sept 19 2019	ALL
Requirement Analysis -Functional & Non-Functional Requirements -Software & Hardware Requirements	Sept 13 2019	Sept 15 2019	Sept 19 2019	Sept 19 2019	ALL
Estimations -Estimate Size, Effort, Duration, Person & Cost of for the project	Sept 20 2019	Sept 20 2019	Sep 23 2019	Sept 27	Priyanka Patil Vaibhavi Patil Divya Patil
Modelling -Describing relationships between modules and sub modules -Describe the schema of database and the relationship between the various entities in it	Sep 24 2019	Sept 24 2019	Sep 26 2019	Sept 27 2019	Divya Patil Prerna Patil Jagruti Patil
Design -Design various UML Models	Sep 27 2019	Sept 29 2019	Oct 03 2019	Oct 04 2019	Vaibhavi Patil Prerna Patil Jagruti Patil
Project-I Documentation -Prepare Expected Result, Conclusion and References Compile all data into a report -Prepare Presentation	Oct 07 2019	Oct 08 2019	Oct 10 2019	Oct 12 2019	Priyanka Patil Vaibhavi Patil

Form Design -Design the graphical user interface (GUI) of the various modules, show relationship among them.	Dec 15 2019	Dec 15 2019	Jan 01 2020	Jan 02 2020	ALL
Coding -Decide the Programming Language, IDE, Database Server. -Decide the Coding style to be followed. -Creating classes for the system & linking those classes for proper functioning -Coding Back-End -Connecting Back-End & Front-End	Jan 09 2020	Jan 09 2020	Jan 30 2020	Jan 30 2020	Vaibhavi Patil Priyanka Patil Jagruti Patil
Testing -Create Test Plan -Decide various Test Cases describing scenarios of success and failure -Test the performance of the system in all test cases and obtain Test Results	Feb 06 2020	Feb 06 2020	Feb 20 2020	Feb 20 2020	Prerna Patil Divya Patil Jagruti Patil
Deployment -Delivery of Project, Support, Feedback	Feb 21 2020	Feb 21 2020	Feb 27 2020	Feb 27 2020	ALL
Project-II Documentation -Prepare Estimates, Results, Update Report -Writing User Manual for the system	Mar 03 2020	Mar 03 2020	Mar 19 2020	Mar 19 2020	ALL

3.3 REQUIREMENT ANALYSIS

Analysis is concerned with understanding and modeling the application and domain within which it operates. The initial input to the analysis phase is problem statement, which describes the problem to be solved, and provides a conceptual view of the proposed system. Subsequent dialog with the customer and real-world background knowledge are additional inputs to analysis. The output from analysis is a formal model that captures the three essential aspects of the system: the objects and their relationships, the dynamic flow of control, and the functional transformation of data subject to constraints.

Requirement analysis bridges the gap between system engineering and software analysis design. Software requirement analysis involves requirement collection, classification, structuring, prioritizing and validation. Requirement analysis consists of user requirements Analysis is concerned with understanding and modeling the application and domain within which it operates. The initial input to the analysis phase is problem statement, which describes the problem to be solved, and provides a conceptual view of the proposed system.

3.3.1 Software requirements

Raspbian OS

Raspbian is a free operating system based on Debian optimized for the Raspberry Pi hardware. Raspbian comes with over 35,000 packages, pre-complied software bundled in a nice format for easy installation on Raspberry Pi computer.

Python

Python is a widely used high-level programming language for general-purpose programming, created by Guido Can Rossum and first released in 1991. It has lots of support and libraries which makes it very popular among raspberry pi community. Most of the codes of this project were written in python.

3.3.2 Hardware Requirements

Raspberry Pi 3 Model A Plus

Fig 3.2 Raspberry PI model A Plus

The Raspberry Pi 3 Model A+ is the latest product in the Raspberry Pi 3 range. Like the Raspberry Pi 3 Model B+, it boasts a 64-bit quad core processor running at 1.4 GHz, dual-band 2.4 GHz and 5 GHz wireless LAN, and Bluetooth 4.2/BLE. The dual-band wireless LAN comes with modular compliance certification, allowing the board to be designed into end products with significantly reduced wireless LAN compliance testing, improving both cost and time to market. The Raspberry Pi 3 Model A+ has the same mechanical footprint as the Raspberry Pi 1 Model A+.

The New Model A+ replaced the Model A in November 2014. The design is based around a Broadcom BCM2835 SoC, which includes an ARM1176JZF-S 700 MHz processor, VideoCore IV GPU, and 256 Megabytes of RAM. This revision A+ board features four mounting holes for easy installation, a built-in reset circuit, and can be powered via the USB data ports. The design does not include a built-in hard disk or solid-state drive, instead relying on an microSD card for booting and long-term storage. This board is intended to run Linux kernel based operating systems.

Raspberry Pi Model A+ has 40pin extended GPIO so user can build even bigger and better projects than ever before. The first 26 pins are identical to the Model A to provide 100% backward compatibility for your projects. Micro SD slot instead of the full size SD slot for storing information and loading your operating systems. The Board can now provide up to 1.2 AMP to the USB port enabling user to connect more power hungry USB devices directly to the Raspberry PI. (This feature requires a 2Amp micro USB Power Supply) The A+ board now uses less power (600mA) than the Model A Board (750mA) when running.

Technical Specification :-

Processor :

- Broadcom BCM2835 chipset.
- 1.2GHz Quad-Core ARM Cortex-A53 (64Bit)

Microprocessor	700 MHz Broadcom
Chipset	Broadcom BCM2835
RAM	1GB
GPIO	40 Pins
USB	1 port (USB 2.0)
HDMI	Full size (video output at 1080p)
SD Card	Micro SD

Combined 4 Pole Jack	Combined 3.5mm audio jack and composite video
Camera Interface	CSI port for connecting the Raspberry Pi camera
Display Interface	DSI port for connecting the Raspberry Pi touch screen display
Power Supply	5V DC 1A micro USB

802.11 b/g/n Wireless LAN and Bluetooth 4.1 (Bluetooth Classic and LE) :

- IEEE 802.11 b / g / n Wi-Fi. Protocol: WEP, WPA WPA2, algorithms AES-CCMP (maximum key length of 256 bits), the maximum range of 100 meters.
- IEEE 802.15 Bluetooth, symmetric encryption algorithm Advanced Encryption Standard (AES) with 128-bit key, the maximum range of 50 meters.

GPU :

- Dual Core Video Core IV® Multimedia Co-Processor. Provides Open GL ES 2.0, hardwareaccelerated Open VG, and 1080p30 H.264 high-profile decode.
- Capable of 1Gpixel/s, 1.5Gtexel/s or 24GFLOPs with texture filtering and DMA infrastructure

Memory : 1GB LPDDR2

Operating System : Boots from Micro SD card, running a version of the Linux operating system or Windows 10 IoT

Dimensions : 85 x 56 x 17mm

Ethernet : 10/100 BaseT Ethernet socket

Video Output :

- HDMI (rev 1.3 & 1.4)

- Composite RCA (PAL and NTSC) **Audio Output**
- Audio Output 3.5mm jack • HDMI • USB 4 x USB 2.0 Connector

GPIO Connector :

- 40-pin 2.54 mm (100 mil) expansion header: 2x20 strip
- Providing 27 GPIO pins as well as +3.3 V, +5 V and GND supply lines **Camera Connector :** 15-pin MIPI Camera Serial Interface (CSI-2)

Display Connector :

- Display Serial Interface (DSI) 15 way flat flex cable connector with two data lanes and a clock lane
- pole Stereo output and Composite video port
- Full size HDMI CSI camera port for connecting the Raspberry Pi camera
- DSI display port for connecting the Raspberry Pi touch screen display
- Micro SD port for loading your operating system and storing data
- Micro USB power source
- Smaller board footprint and is fully HAT compatible
- Stream and watch Hi-definition video output at 1080P
- Combined 4-pole jack for connecting your stereo audio out and composite video out

Memory Card Slot :

- Push/pull Micro SDIO

Fig 3.3 Memory Card Slot

The GPU provides Open GL ES 2.0, hardware-accelerated Open VG, and 1080p30 H.264 high-profile decode and is capable of 1Gpixel/s, 1.5Gtexel/s or 24 GFLOPs of general purpose compute. What‟s that all mean? It means that if you plug the Raspberry Pi 3 into your HDTV, you could watch BluRay quality video, using H.264 at 40MBits/s.

The biggest change that has been enacted with the Raspberry Pi 3 is an upgrade to a next generation main processor and improved connectivity with Bluetooth Low Energy (BLE) and BCM43143 Wi-Fi on board. Additionally, the Raspberry Pi 3 has improved power management, with an upgraded switched power source up to 2.5 Amps, to support more powerful external USB devices.

The Raspberry Pi 3‟s four built-in USB ports provide enough connectivity for a mouse, keyboard, or anything else that you feel the RPi needs, but if you want to add even more you can still use a USB hub. Keep in mind, it is recommended that you use a powered hub so as not to overtax the on-board voltage regulator. Powering the Raspberry Pi 3 is easy, just plug any USB power supply into the micro-USB port. There‟s no power button so the Pi will begin to boot as soon as power is applied, to turn it off simply remove power. The four built-in USB ports can even output up to 1.2A enabling you to connect more power hungry USB devices (This does require a 2Amp micro USB Power Supply).

Fig 3.4 USB

On top of all that, the low-level peripherals on the Pi make it great for hardware hacking. The 0.1" spaced 40-pin GPIO header on the Pi gives you access to 27 GPIO, UART, I 2C, SPI as well as 3.3 and 5V sources. Each pin on the GPIO header is identical to its predecessor the Model B+.

SoC :

Built specifically for the new Pi 3, the Broadcom BCM2837 system-on-chip (SoC) includes four high-performance ARM Cortex-A53 processing cores running at 1.2GHz with 32kB Level 1 and 512kB Level 2 cache memory, a VideoCore IV graphics processor, and is linked to a 1GB LPDDR2 memory module on the rear of the board.

Fig 3.5 SoC

GPIO :

The Raspberry Pi 3 features the same 40-pin general-purpose input-output (GPIO) header as all the Pis going back to the Model B+ and Model A+. Any existing GPIO hardware will work without modification; the only change is a switch to which UART is exposed on the GPIO‟s pins, but that‟s handled internally by the operating system.

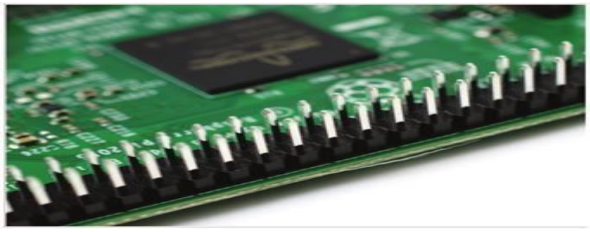


Fig 3.6 GPIO USB

chip :

The Raspberry Pi 3 shares the same SMSC LAN9514 chip as its predecessor, the Raspberry Pi 2, adding 10/100 Ethernet connectivity and four USB channels to the board. As before, the SMSC chip connects to the SoC via a single USB channel, acting as a USB-to-Ethernet adaptor and USB hub.

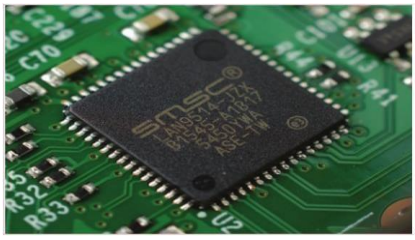


Fig 3.7 USB Chip

Antenna :

There‟s no need to connect an external antenna to the Raspberry Pi 3. Its radios are connected to this chip antenna soldered directly to the board, in order to keep the size of the device to a minimum. Despite its diminutive stature, this antenna should be more than capable of picking up wireless LAN and Bluetooth signals – even through walls.

Fig 3.8 Antenna

The main differences are the quad core 64-bit CPU and on-board Wi-Fi and Bluetooth. The RAM remains 1GB and there is no change to the USB or Ethernet ports. However, the upgraded power management should mean the Pi 3 can make use of more power hungry USB devices.

For Raspberry Pi 3, Broadcom have supported us with a new SoC, BCM2837. This retains the same basic architecture as its predecessors BCM2835 and BCM2836, so all those projects and tutorials which rely on the precise details of the Raspberry Pi hardware will continue to work. The 900MHz 32-bit quad-core ARM CortexA7 CPU complex has been replaced by a custom-hardened 1.2GHz 64-bit quad-core ARM Cortex-A53.

In terms of size it is identical to the B+ and Pi 2. All the connectors and mounting holes are in the same place so all existing add-ons, HATs and cases should fit just fine although the power and activity LEDs have moved to make room for the WiFi antenna.

The performance of the Pi 3 is roughly 50-60% faster than the Pi 2 which means it is ten times faster than the original Pi. All of the connectors are in the same place and have the same functionality, and the board can still be run from a 5V micro-USB power adapter. This time round, we‟re recommending a 2.5A adapter if you want to connect power-hungry USB devices to the Raspberry Pi.

Raspberry Pi Camera Module V2 :

The Raspberry Pi Camera v2 is the new official camera board released by the Raspberry Pi Foundation. The Raspberry Pi Camera Module v2 is a high quality 8 megapixel Sony IMX219 image sensor custom designed add-on board for Raspberry Pi, featuring a fixed focus lens. The Raspberry Pi

Zero now comes complete with a camera port! Using the new Raspberry Pi Zero Camera Adapter, you can now use a Raspberry Pi camera to your Zero. It attaches to Pi by way of one of the small sockets on the board upper surface and uses the dedicated CSi interface, designed especially for interfacing to cameras.

Hardware features

Available	**Implemented**
Chief ray angle correction	Yes
Global and rolling shutter	Rolling shutter
Automatic exposure control (AEC)	No - done by ISP instead
Automatic white balance (AWB)	No - done by ISP instead
Automatic black level calibration (ABLC)	No - done by ISP instead
Automatic 50/60 Hz luminance detection	No - done by ISP instead
Frame rate up to 120 fps	Max 90fps. Limitations on frame size for the higher frame rates (VGA only for above 47fps)
AEC/AGC16-zone size/position/weight control	No - done by ISP instead
Mirror and flip	Yes
Cropping	No - done by ISP instead (except 1080p mode)
Lens correction	No - done by ISP instead
Defective pixel cancelling	No - done by ISP instead
10-bit RAW RGB data	Yes - format conversions available via GPU
Support for LED and flash strobe mode	LED flash
Support for internal and external frame	No

synchronisation for frame exposure mode	
Support for 2 × 2 binning for better SNR in low light conditions	Anything output res below 1296 x 976 will use the 2 x 2 binned mode
Support for horizontal and vertical sub-sampling	Yes, via binning and skipping
On-chip phase lock loop (PLL)	Yes
Standard serial SCCB interface	Yes
Digital video port (DVP) parallel output interface	No
MIPI interface (two lanes)	Yes
32 bytes of embedded one-time programmable (OTP) memory	No
Embedded 1.5V regulator for core power	Yes

Software features

Picture formats	JPEG (accelerated), JPEG + RAW, GIF, BMP, PNG, YUV420, RGB888
Video formats	raw h.264 (accelerated)
Effects	negative, solarise, posterize, whiteboard, blackboard, sketch, denoise, emboss, oilpaint, hatch, gpen, pastel, watercolour, film, blur, saturation
Exposure modes	auto, night, nightpreview, backlight, spotlight, sports, snow, beach, verylong, fixedfps, antishake, fireworks
Metering modes	average, spot, backlit, matrix
Automatic white balance modes	off, auto, sun, cloud, shade, tungsten, fluorescent, incandescent, flash, horizon
Triggers	Keypress, UNIX signal, timeout

Extra modes	demo, burst/timelapse, circular buffer, video with motion vectors, segmented video, live preview on 3D models

Features:

- Fixed focus lens on-board
- 5 megapixel native resolution sensor-capable of 640 × 480 pixel static images
- Supports 1080p30, 720p60 and 640x480p90 video
- Size 25mm x 23mm x 9mm
- Weight just over 3g
- Connects to the Raspberry Pi board via a short ribbon cable (supplied)
- Camera v2 is supported in the latest version of Raspbian, Raspberry Pi's preferred operating system.

7’ Inch Touch Screen Tft Display :

Fig 3.9 7’ Inch Touch Screen TFT Display

Elecrow Raspberry Pi 3 Display 7 Inch Touch Screen HDMI HD LCD TFT 1024X600 Monitor 7inch RPI Display for Raspberry Pi 3. This 7 inch HDMI LCD supports various systems like Raspberry Pi,Banana Pi,Banana Pro,BB Black to provide Lubuntu, Raspbian with and Angstrom images with high resolution of 1024×600 and capacitive Touch Screen. Besides it upgrades to IPS screen with larger visible angle and more clear display effect. Broadly you can apply it to raspberry pi, HDMI display screen and other mini PC or even computer display.

A good TFT LCD display with a high resolution picture and large viewing screen. Resolution:1024X600. 7 inches LCD display with touch function.

Large viewing angle. Fast response time. Full colour display with high quality. Works perfectly for Raspberry Pi B+/2B Raspberry Pi 3B win 7/8.

Features

- 1024X600 high resolution
- Touch control
- Supports Raspberry Pi, comes with Raspbian driver (works with your Raspbian directly), and Ubuntu image
- Supports Raspberry Pi B+/2B Raspberry Pi 3 win 7/8
- HDMI interface for displaying, USB interface for touch control
- Back light control to lower power consumption - Provide the driver for Raspbian.
- 7inch standard display, 1024 x 600 ultra clear resolution
- Capacitive touch screen, maximum support 5 point touch
- Support backlight control alone, can turn off backlight to save power
- Support standard HDMI interface input
- Can be used as a general-purpose HDMI display, such as access to the computer HDMI as a secondary display (resolution output can be adjusted to 1024X600)
- As a raspberry pie display, support Raspbian, Ubuntu, Kodi, win10, IOT, single touch, free drive
- Used as a computer monitor, support win7, win8 system 5 point touch (XP and older version of the system single touch), free drive
- Dimension:164.9*124.27(mm)
- Package list
- 7inch HDMI LCD x 1
- HDMI cable x 1
- USB type A plug to micro B plug cable x 1
- RPi screws pack (4pcs) x 1
- DVD × 1
- A good solution for those seeking for a bigger resolution display
- Good touch response
- Supports Banana Pi / Banana Pro, comes with Lubuntu, Raspbian images
- Supports BB Black, comes with Angstrom image
- Supports Raspberry Pi, comes with Raspbian driver (works with your Raspbian directly), and Ubuntu image

- Not only for mini-PCs, it can work as a computer monitor just like any other general HDMI screen (touch function is unavailable in this case)
- Back light control to lower power consumption
- HDMI interface for displaying, USB interface for touch control

Specifications

- LCD Type:TFT
- 7 inch TFT Capacitive touch screen display, 1024x600 Resolution
- HDMI input
- Usb touch and power, 5V@1A
- Lcd Size ：164.7mm*107.1mm
- Weight:360

Usage

When users connect the Raspberry Pi to use, they need to configure the official system. Or you can also burn the configured system image directly.

3.3.3 Hardware Connection

1. Connect the HDMI Connector to both the HDMI interfaces on the LCD and the Pi.
2. LCD and Pi connect power.
3.Turn on the "backlight" switch on the back of the LCD.

Fig 3.10 Hardware connection

3.3.4 Minimum Hardware Requirements

Table 3.3 Minimum Hardware Requirements

Hardware	Minimum Requirement
Processor	Broadcom BCM2835
Primary Memory	1GB
Secondary Memory	8GB
Internet Connection	1 USB port
Input Device	Webcam

3.3.5 Minimum Software Requirements

Table 3.4 Minimum Software Requirements

Role	Software	Minimum Requirement
Development	**Platform (OS)**	Windows 7 and higher
	Front End (Prog. Lang.)	Python
	Backend (DB)	-
	Development Tool (IDE)	Python Idle
	Testing Tool	Selenium IDE or Firefox Mozilla with Selenium Plugin
Deployment	**Execution Environment**	vnc viewer
	Browser	Any
	Server(Application / Database Server)	Cloud
Documentation	**Documentation Tool**	Microsoft Office 2010or Above
	Estimation Tool	SystemStar 3.0
Design	**UML Design**	Star UML
	DFD, ER, Flows	Edraw 6.1

3.4 ESTIMATIONS

Software estimation is the process of predicting the most realistic amount of effort required to develop or maintain software based on incomplete, uncertain and noisy input. Accurate estimation of the problem size is fundamental to satisfactory estimation of effort, time duration and cost of a software project. In order to be able to accurately estimate the project size, some important metrics should be defined in terms of which the project size can be expressed. The size of a problem is obviously not the number of bytes that the source code occupies. It is neither the byte size of the executable code. The project size is a measure of the problem complexity in terms of the effort and time required to develop the product. Currently two metrics are popularly being used widely to estimate size: lines of code (LOC) and function point (FP).***We will use LOC metric to estimate size.***

Estimation is done using following:

- Estimate the **size in lines of code** of each module
- Estimate the **effort in person-month** or person-hours
- Estimate the **duration in** calendar **month**
- Estimate the **number of person** required
- Estimate the **cost in currency**

3.4.1 Estimation Technique (Basic COCOMO Model)

COCOMO (**Co**nstructive **Co**st Estimation **Mo**del) was proposed by Boehm [1981]. COCOMO predicts the efforts and schedule of a software product based on size of the software. According to Boehm, software cost estimation should be done through three stages: **Basic COCOMO, Intermediate COCOMO** and**Detailed / Complete / Advanced COCOMO.**

- **Basic COCOMO:** It a single-valued, static model that computes software development effort (and cost) as a function of program size expressed in estimated thousand delivered source instructions (KDSI) i.e., Lines of code (LOC).
- **Intermediate COCOMO:** an extension of the Basic model that computes software development effort as a function of program size by adding a set of "cost drivers," that will

determine the effort and duration of the project, such as assessments of personnel and hardware.

- **Detailed COCOMO:**an extension of the Intermediate model that adds effort multipliers for each phase of the project to determine the cost driver‟s impact on each step (analysis, design, etc.) of the software engineering process.

In our project we are going to use "Basic COCOMO" model for estimations. Basic COCOMO categorizes projects into three types:

i. **Organic Mode:**(**Application Programs**such as: data processing, scientific, etc.)
Development projects typically are not complicated and involve small experienced teams. The planned software is not considered innovative (i.e. little innovation) and requires a relatively small amount of DSIs (typically 2000 to 50,000 LOC). The organic projects are those developed in a stable development environment and does not have a tight deadline or constraints.

ii. **Semidetached Mode:** (**Utility Programs** such as: compilers, linkers, analyzers, etc.)
Development projects typically are more complicated than in Organic Mode and involve teams of people with mixed levels of experience. The software requires no more than 50,000 to 300,000 DSI‟s. The projects require minor innovations and has some deadline & constraint restrictions where the development environment is not much stable. Examples of this type are developing new database management system.

iii. **Embedded Mode:** (**System Programs** such as: operating system, etc.)
Development projects must fit into a rigid set of requirements because the software is to be embedded in a strongly joined complex of hardware, software, regulations and operating procedures. Contains a large highly experienced project team which is required to do some highly innovative work with very tight deadlines and severe constraints. The project requires no greater than 300,000 DSI‟s.

The Basic COCOMO formula takes the form:

Effort, E =a_b(KLoC/KDSI) $^{(b_b)}$	**[person-months]**
Duration, D = c_b(E) $^{(d_b)}$	**[months]**
Person, P = E / D	**[persons]**

where,**E** is the effort applied in person-months, **KLoC** is the estimated number of thousands of delivered lines of code for the project, **D** is total time duration to develop the system in months, and **P** is number of persons required to develop that system.

The coefficient a_b,c_b and the exponent b_b,d_b are given in the next table.

Table 3.5 Coefficient values for Basic COCOMO

Software project	a_b	b_b	c_b	d_b
Organic	2.4	1.05	2.5	0.38
Semi-detached	3.0	1.12	2.5	0.35
Embedded	3.6	1.20	2.5	0.32

Our project will fall in the "Embedded" category.

3.4.2 Historical Data Collection

As we do not have access any the coding of smart mirror system developed earlier, we cannot specify the exact modules of such a system. But as our system uses very basic modules we can specify the approximate size of such projects based on our experience. The project is modularized as shown :

Table 3.6 Size Estimation of Historical Data

Software Module	LOC
Hardware coding	3500
Graphical User Interface	2000
Total Estimated Lines of Code (LOC)	**5500**

We are here considering the approximate size of such software would be 5500 LOC.

3.4.3 Size Estimation

Table 3.7Size Estimation of Current System.

Software Module	LOC
Graphical user interface	5500
Hardware coding	4000

Total Estimated Lines of Code (LOC)	9500

Total lines of code of our project will be approximately 9500 LOC or DSI.

3.4.4Effort Estimation

Effort (E) = a_b(KLoC) $^{(b_b)}$ [Person -Month]

The value a_b and b_b according to **embedded** system is: a_b

= 3.6 and b_b = 1.20

Total LOC (approx) of project is: 9500 LOC = 9.5 KLOC

E = (3.6)*(9.5)^1.20

E = 53.65

Person-Month = 53 PM (approx)

3.4.5Duration Estimation

Duration (D) = c_b(E)$^{(d_b)}$ [months]

The value c_b and d_b according to **embedded** system is:

c_b = 2.5 and d_b = 0.32

Effort, E as calculated above is 37.34 PM

D = (2.5)*(53.65)^0.32

D = 2.5*3.57

D = 8.94

Duration ≈ 9 [months]

3.4.6 Person Required

Person Required = Effort Applied (E) / Development Time (D) [count]

= 53/9

= 5.88

Person Required = 5 [Persons]

3.4.7 Cost Estimation

We take the assumption each person charges Rs. 1,000/month and additional 1,000/month for resources required.

Cost Estimation = (1000*5+1000)*8

Cost Estimation = 48000

Total Cost Estimation = Rs 48000

3.4.8 Estimation Summary

Table 3.8 Summary of calculated estimations.

Estimation	Value
Size of the Project	**9500** Lines of Code
Effort Required	**53** Person-Month
Duration Required	**9** months
Person Required	5 persons
Cost Required	Rs **48000**

3.5 ANALYSIS MODELING

3.5.1 Data Modeling (Entity - Relationship Diagram)

Entity-Relationship diagram (ERD) is a graphical technique, which is used to represent entities present in the system, and relationship those are applied between these entities. [15]

The entity-relationship (E-R) data model is based on a perception of a real world that consists of a collection of basic objects, called **entities**, and of **relationships** among these objects.

An entity is a “thing” or “object” in the real world that is distinguishable from other objects. For example, each person is an entity, and bank accounts can be considered as entities.Entities are described in a database by a set of **attributes**.A **relationship** is an association among several entities.

The overall logical structure (schema) of a database can be expressed graphically by an E-R diagram, which is built up from the following components:

- **Rectangles,** which represent entity sets
- **Ellipses,** which represent attributes
- **Diamonds,** which represent relationships among entity sets
- **Lines,** which link attributes to entity sets and entity sets to relationships
- Each component is labeled with the entity or relationship that it represents.

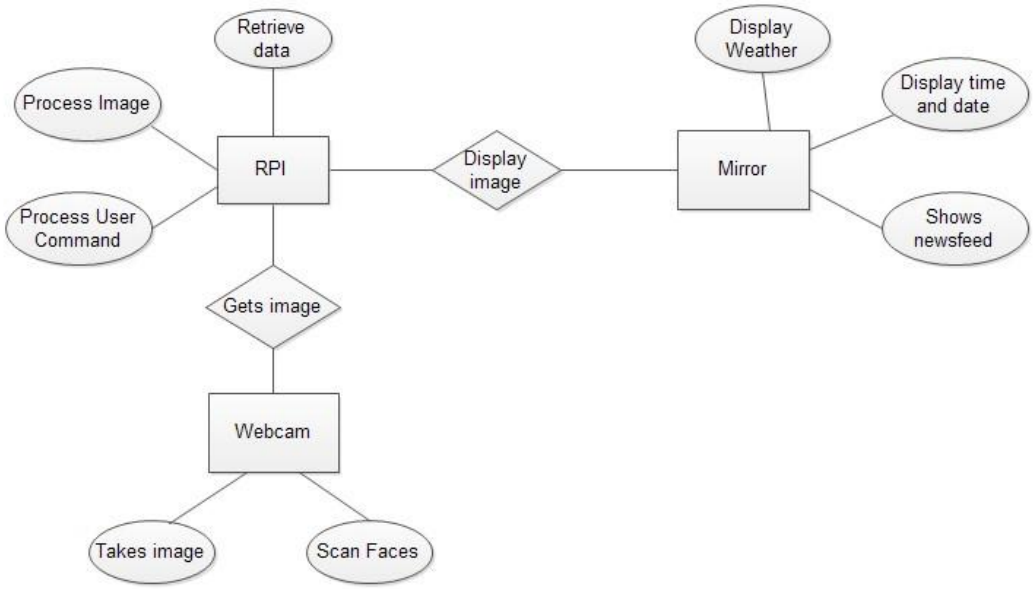


Figure 3.11 Entity-Relationship Diagram

3.5.2 Functional Modeling(Data Flow Diagram)

Data flow diagram (DFD), also called as „Bubble chart" is a hierarchical (or leveled) set of diagrams, used to represent the flow of data elements into and out of the functional units of the program, data stores, environmental sources and sinks.

The data flow diagram (DFD) serves two purposes:(1) to provide an indication of how data are transformed as they move through the system and **(2)** to depict the functions (and sub-functions) that transform the data flow.

The data flow diagram may be used to represent a system or software at any level of abstraction. In fact, DFDs may be partitioned into levels that represent increasing information flow and functional detail.

A level 0 DFD, also called a fundamental system model or a context model, represents the entire software element as a single bubble with input and output data indicated by incoming and outgoing arrows, respectively. Additional processes (bubbles) and information flow paths are represented as the level 0 DFD is partitioned to reveal more detail. For example, a level 1 DFD might contain five or six bubbles with interconnecting arrows. Each of the processes represented at level 1 is a sub-function of the overall system depicted in the context model.

3.5.2.1 Data Flow Diagram - Level 0

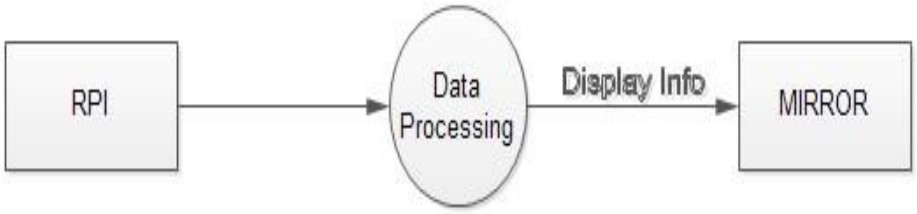


Figure 3.12 Data Flow Diagram level-0

3.5.2.1 Data Flow Diagram - Level 1

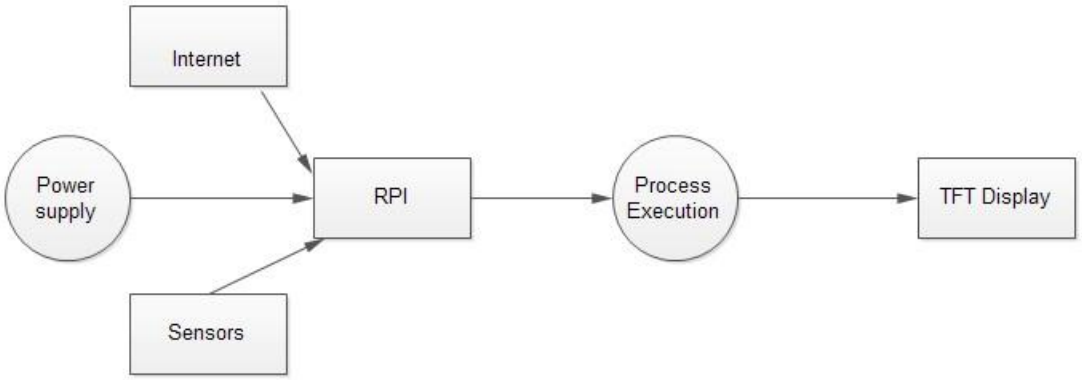


Figure 3.13 Data Flow Diagram level-1

3.5.2.1 Data Flow Diagram - Level 2

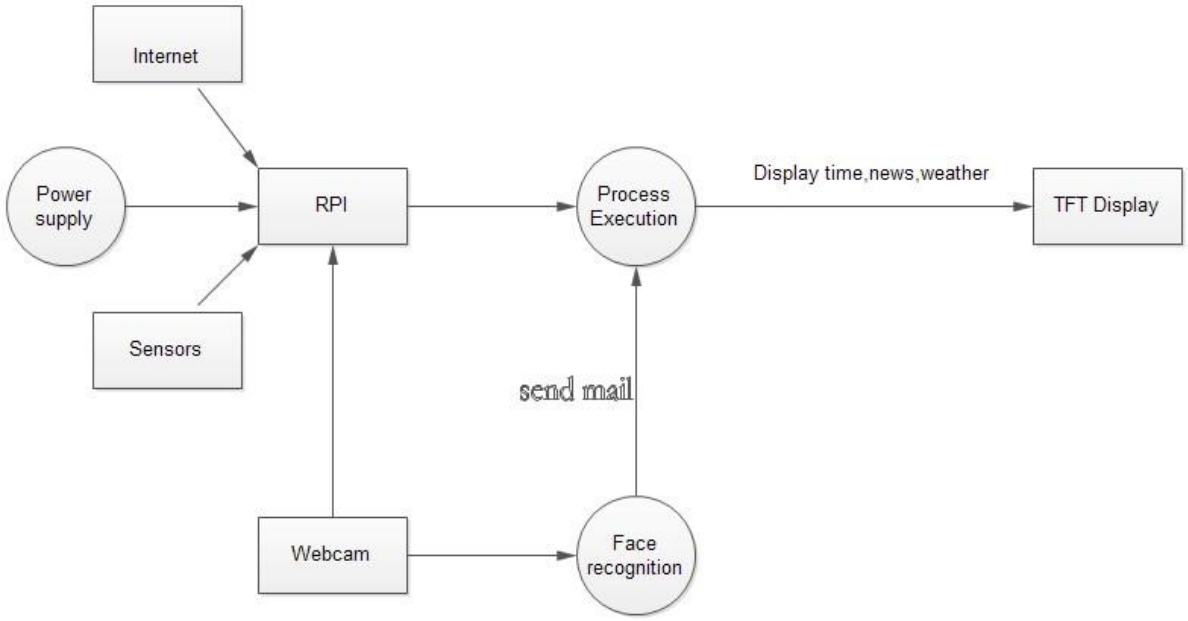


Figure 3.14 Data Flow Diagram level-2

CHAPTER 4
DESIGN

4.1 INTRODUCTION

Design uses a combination of text and diagrammatic forms to depict the requirements for data, function and behavior in a way that is relatively easy to understand and more important, straightforward to review for correctness, completeness and consistency.

A diagram is the graphical presentation of a set of elements most often rendered as a connected graph of vertices (things) and arcs (relationship). These diagrams are drawn to visualize a system from different perspectives so a diagram into a system.

4.2 UML MODELING

The unified modeling language (UML) is a Graphical Language for visualization, Specifying, construction and documenting the artifacts of a software intensive system. The UML gives a standard was to write system‟s blue prints, covering conceptual thing, such as Business Processes & system functions, As well as concrete things, such as classes written in a specific programming language, database schemas, and reusable software components. [17]

4.2.1 Use Case Diagram

A use case defines behavioral features of a system. Each use case is named using a verb phase expresses a goal of the system. A use case diagram shows a set of use casesand actors &their relationships. Use case diagrams address the static use case view of a system.These diagrams are especially important in organizing and modeling tshe behaviors of a system. It shows the graphical overview of functionality provided by the system intents actor.

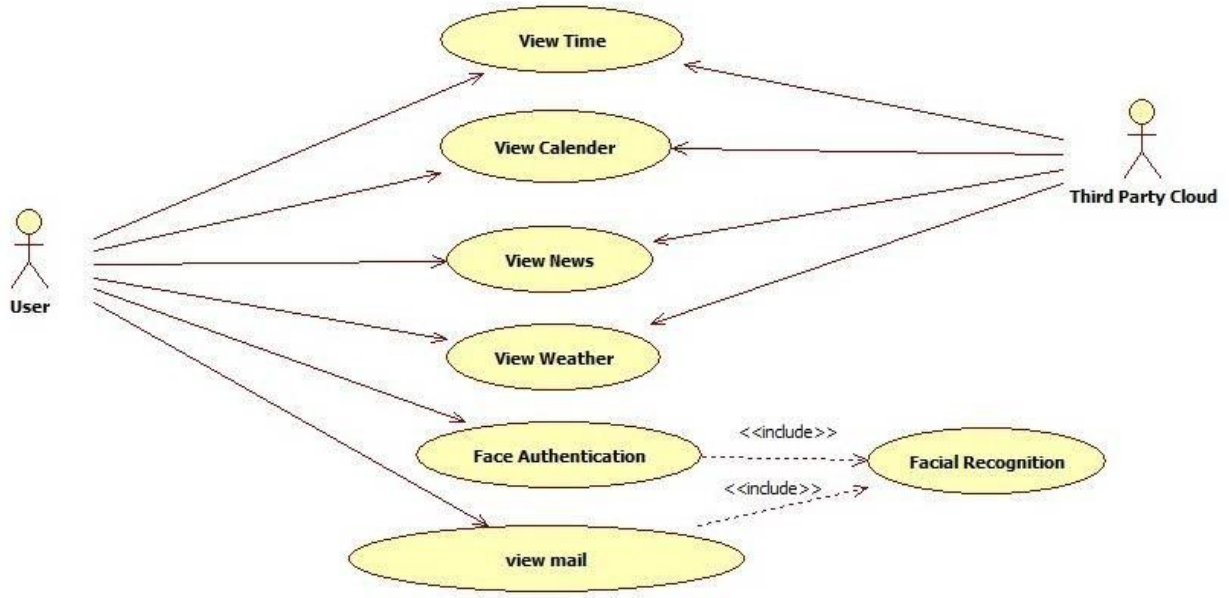


Figure 4.1 Use Case Diagram For Smart Mirror

Table 4.1 Use Case Description for Smart Mirror

Use case	Description
View time	Gives information about the time
View calender	Gives information about the date
View news	Gives information about the newsfeed
View weather	Gives information about the weather
Face authetication	Scan for faces and autheticates
View mail	Gives information about the mail

4.2.2 Activity Diagram

An activity diagram of a special kind of a state chart diagram that shows the flow from activity within a system. An activity addresses the dynamic view of a system. The activity diagram is often seen as part of the functional view of a system because it describes logical processes, or functions. Each process describes a sequence of tasks and the decisions that govern when and they are performed. The flow in an activity diagram is driven by the completion of an action.

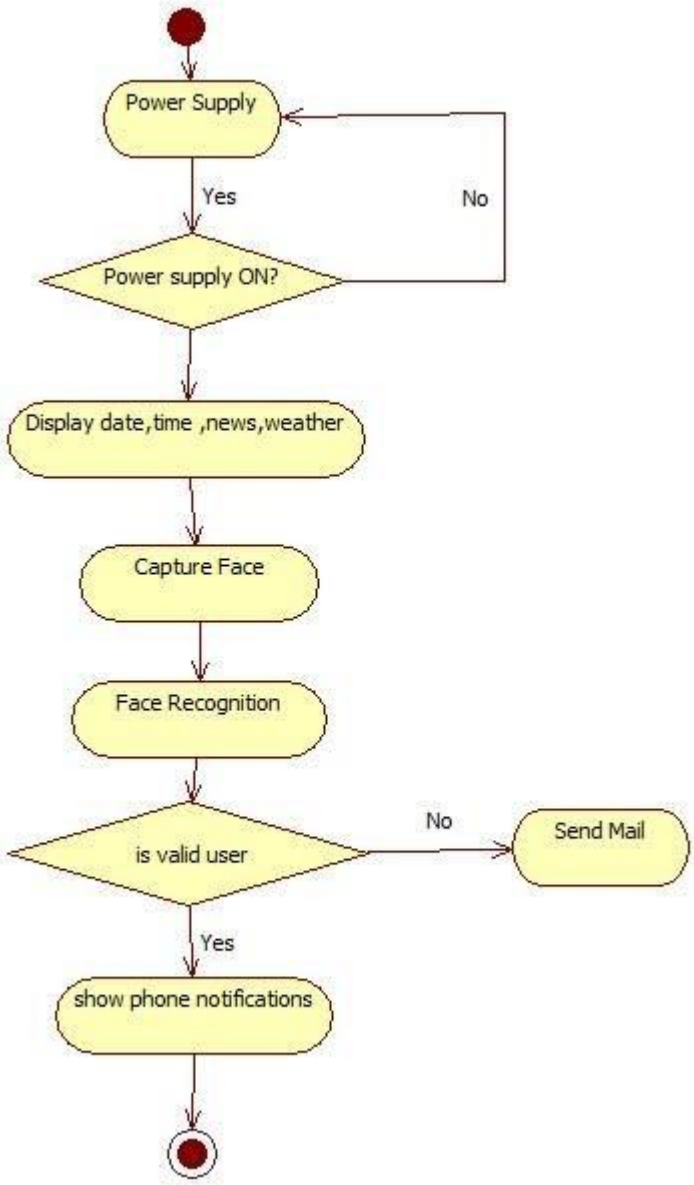


Figure 4.2 Activity Diagram For Smart Mirror

4.2.3 Sequence Diagram

Sequence diagram are a kind of interaction diagram. An shows an interaction, consisting of a set of objects and their relationships, including the message that may be dispatched among them. A sequence diagram emphasizes the time ordering of messages. As shown in figure we can form a sequence diagram by first placing the objects that participate in the interaction at the top of our diagram. The object that initiates the interaction at the left and increasingly more subordinate objects to the right. The messages that these objects send and receive along the Y-axis, in order of increasing time from top to bottom. This gives the reader a clear visual cue to the flow of control over time.

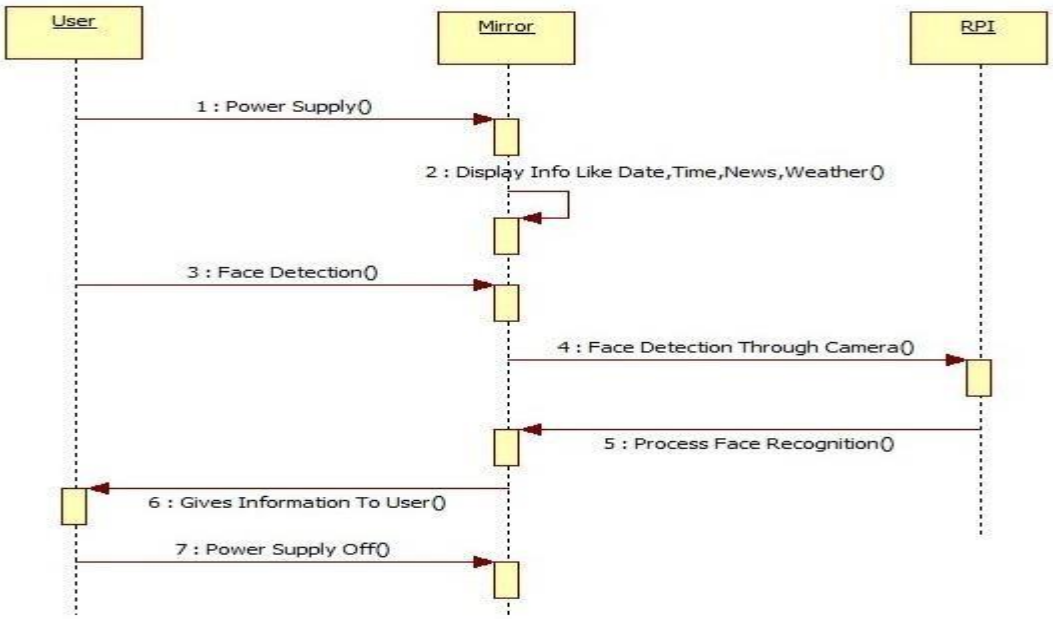


Figure 4.3 Sequence Diagram For Smart Mirror

4.2.4 State Machine Diagram

A state machine diagram models the behavior of a single object, specifying the sequence of events that an object goes through during is lifetime in response to events. The most common purpose for which you will use state machines is to model the lifetime of an object, especially instances of classes, use cases and the system as a whole.

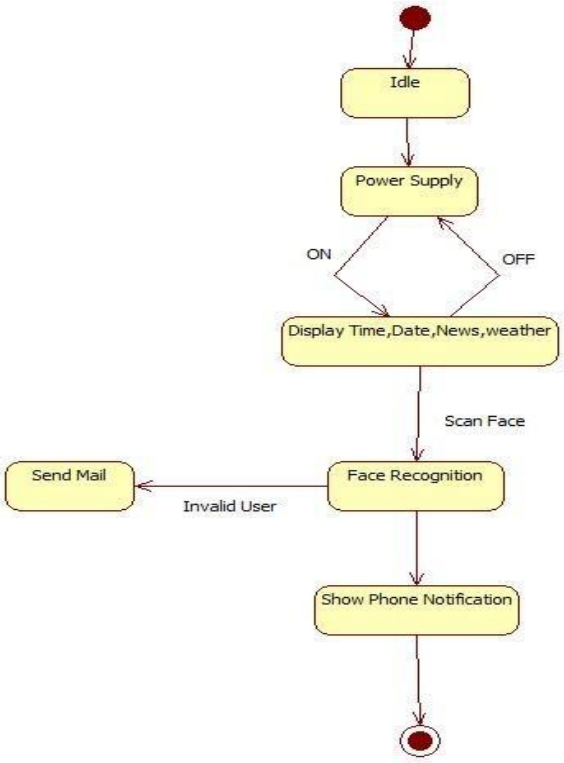


Figure 4.4 State Chart Diagram For Smart Mirror

4.2.5 Class Diagram

A class diagram shows a set of classes, interfaces and collaborations and their relationship. These diagrams are the most common diagram found in modeling object oriented systems. Class diagram addressed the static design view of a system.

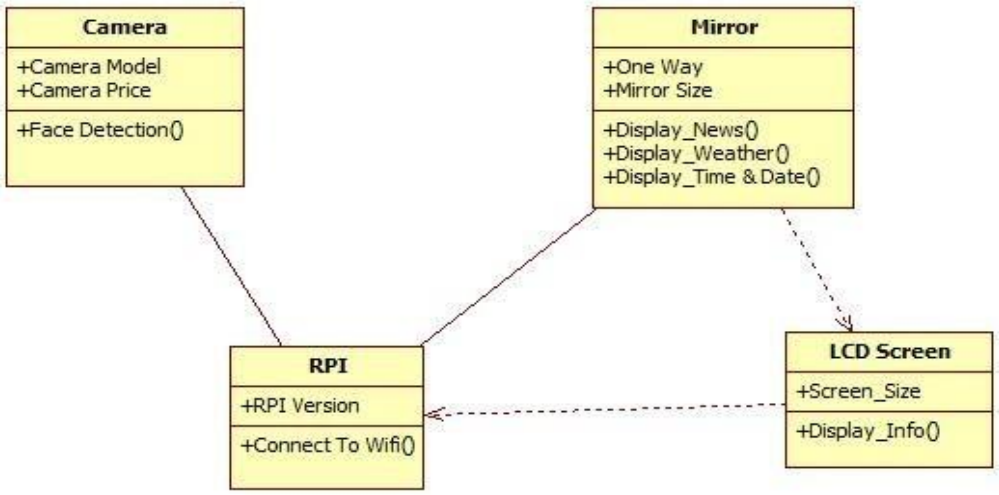


Figure 4.5 Class Diagram For Smart Mirror

Table 4.2 Description of Classes

Class	Description
Camera	Captures faces for authentication or security purpose
Mirror	Shows newsfeed, weather, time and date
RPI	Gives information for the process detail, notifications,email,etc
LED Screen	Display information

4.2.6 Component Diagram

A component diagram shows the organization and dependencies among a set of components. Component diagrams address the static implementation view of a system. Component diagrams are one of the two kinds of diagrams found in modeling the physical aspects of object-oriented systems. A component diagram shows the organization and dependencies among set of

components. You can use component diagrams to model the static implementation view of a system.

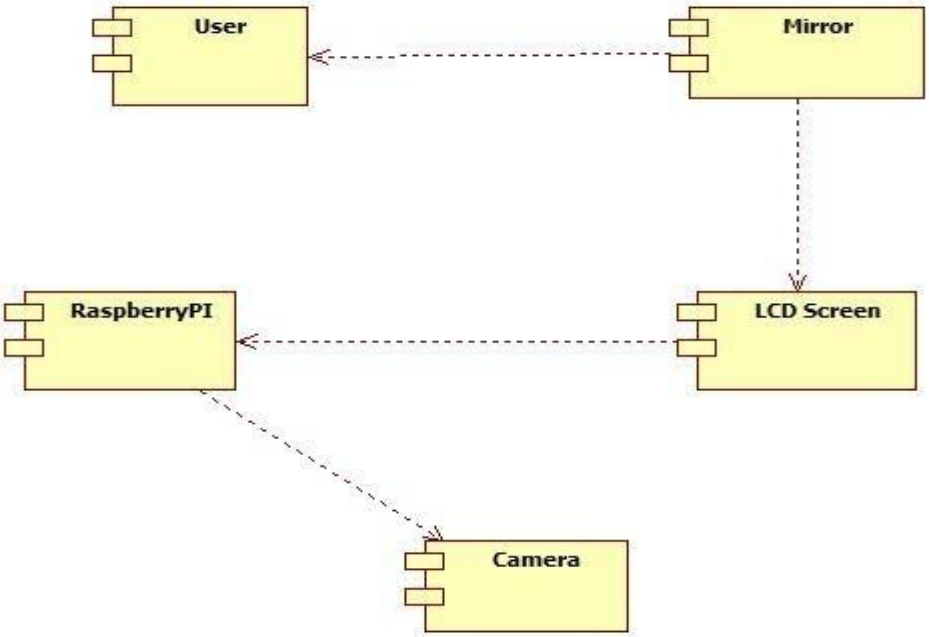


Figure 4.6 Component Diagram For Smart Mirror

4.2.7 Deployment Diagram

Deployment diagram shows the configuration of run time processing nodes and components that live on them. Deployment diagram address the static deployment view of architecture. A deployment diagram shows the configuration of run-time processing nodes and the components that live on them. Deployment diagrams address the static view of architecture. They are related to components diagram in that a node typically encloses one or more components

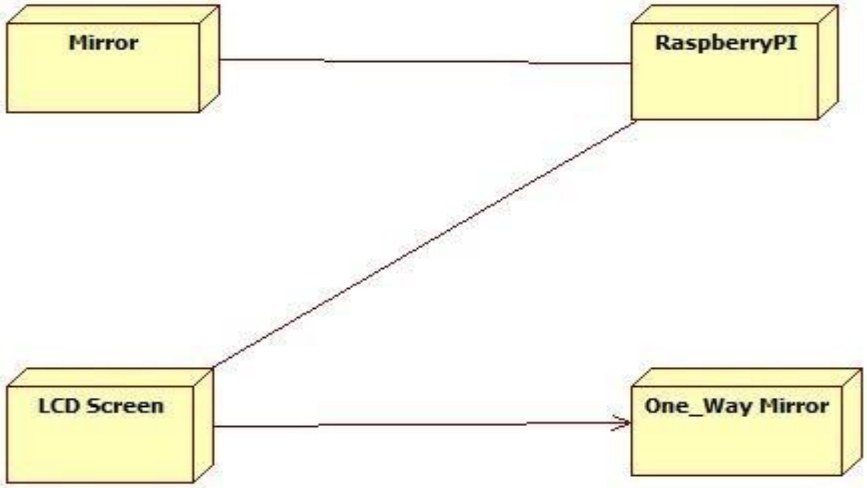


Figure 4.7 Deployment Diagram For Smart Mirror

CHAPTER 5

CODING

5.1 IMPLEMENTATION LANGUAGE

5.1.1 PYTHON

Python is an interpreted, high-level, general-purpose programming language. Created by Guido van Rossum and first released in 1991, Python's design philosophy emphasizes code readability with its notable use of significant whitespace. Its language constructs and object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects.

Python is dynamically typed and garbage-collected. It supports multiple programming paradigms, including structured (particularly, procedural), object-oriented, and functional programming. Python is often described as a "batteries included" language due to its comprehensive standard library.

Python was conceived in the late 1980s as a successor to the ABC language. Python 2.0, released in 2000, introduced features like list comprehensions and a garbage collection system capable of collecting reference cycles. Python 3.0, released in 2008, was a major revision of the language that is not completely backward-compatible, and much Python 2 code does not run unmodified on Python 3.

The Python 2 language was officially discontinued in 2020 (first planned for 2015), and "Python 2.7.18 is the last Python 2.7 release and therefore the last Python 2 release." No more security patches or other improvements will be released for it. With Python 2's end-of-life, only Python 3.5.x and later are supported.

5.1.1.1 Features in Python

- **Easy to code:**

Python is high level programming language. Python is very easy to learn language as compared to other language like c, c#, java script, java etc. It is very easy to code in python language and anybody can learn python basic in few hours or days. It is also developer-friendly language.

- **Free and Open Source:**

Python language is freely available at official web address. The source code is also available. Therefore it is open source.

- **Object-Oriented Language:**

One of the key features of python is Object-Oriented programming. Python supports object oriented language and concepts of classes, objects, encapsulation etc.

- **GUI Programming Support:**

A software is not user-friendly until its GUI is made. A user can easily interact with the software with a GUI. Python offers various libraries for making Graphical user interface for your applications. PyQt5 is the most popular option for creating graphical apps with Python.

- **High-Level Language:**

Python is a high-level language. When we write programs in python, we do not need to remember the system architecture, nor do we need to manage the memory.

- **Extensible feature:**

Python is a Extensible language. We can write our some python code into c or c++ language and also we can compile that code in c/c++ language.

- **Python is Portable language:**

Python language is also a portable language. For example, if we have python code for windows and if we want to run this code on other platform such as Linux, Unix and Mac then we do not need to change it, we can run this code on any platform.

- **Python is Integrated language:**

Python is also an Integrated language because we can easily integrated python with other language like c, c++ etc.

- **Interpreted Language:**

Python is an Interpreted Language. Because python code is executed line by line at a time. Like other language c, c++, java etc there is no need to compile python code this makes it easier to debug our code. The source code of python is converted into an immediate form called bytecode.

- **Large Standard Library:**

Python has a large standard library which provides rich set of module and functions so you do not have to write your own code for every single thing. There are many libraries present in python for such as regular expressions, unit-testing, web browsers etc.

- **Dynamically Typed Language:**

Python is dynamically-typed language. That means the type (for example- int, double, long etc) for a variable is decided at run time not in advance. Because of this feature we don’t need to specify the type of variable.

5.1.1.2 Reason for using python:

- **Multiple Programming Paradigms:** Python supports several programming paradigms similar to other modern programming languages. It is object-oriented and supports structured programming.
- **Readable and Maintainable Code:** Having a quality code is of extreme. For that to happen, developers need to focus on writing code to simplify maintenance and updates.

- **Compatible with Major Platforms and Systems:** Python supports many operating systems. This means that Python can also be used to run codes on specific tools.
- **Many Open Source Frameworks and Tools:** Due to the open - source nature of Python, it helps in delivering cost - effective software solutions. Python frameworks, libraries, and developmental tools can also be used to cut the software development time without increasing the development cost.
- **Robust Standard Library:** The large and robust standard library of Python makes it advanced and precise. You can choose modules as per your requirements. While creating an application using Python, you can make use of specific modules to perform string operations, implement web services, manage operating system interface, and work with internet protocols. Information can also be gathered by browsing through the Python Standard Library documentation.

5.1.2 Node js

Node.js is a server-side platform built on Google Chrome's JavaScript Engine (V8 Engine). Node.js was developed by Ryan Dahl in 2009 and its latest version is v0.10.36. The definition of Node.js as supplied by its official documentation is as follows −

Node.js is a platform built on Chrome's JavaScript runtime for easily building fast and scalable network applications. Node.js uses an event-driven, non-blocking I/O model that makes it lightweight and efficient, perfect for data-intensive real-time applications that run across distributed devices.

Node.js is an open source, cross-platform runtime environment for developing server-side and networking applications. Node.js applications are written in JavaScript, and can be run within the Node.js runtime on OS X, Microsoft Windows, and Linux.

Node.js also provides a rich library of various JavaScript modules which simplifies the development of web applications using Node.js to a great extent.

Node.js is an open-source, cross-platform, JavaScript runtime environment that executes JavaScript code outside of a web browser. Node.js lets developers use JavaScript to write

command line tools and for server-side scripting—running scripts server-side to produce dynamic web page content before the page is sent to the user's web browser. Consequently, Node.js represents a "JavaScript everywhere" paradigm, unifying web-application development around a single programming language, rather than different languages for server- and client-side scripts.

Though `.js` is the standard filename extension for JavaScript code, the name "Node.js" doesn't refer to a particular file in this context and is merely the name of the product. 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 was previously governed by the Node.js Foundation, and has now merged with the JS Foundation to form the OpenJS Foundation, which is facilitated by the Linux Foundation's Collaborative Projects program.

5.1.2.1 Features in node js

- **Synchronous Code Execution :**

Non-blocking code execution is conceptually more difficult to code than code that runs in a straight line because we have blocks of code hanging around waiting for asynchronous events to return.

- **Object-Oriented:**

A huge complaint against NodeJS was down to its JavaScript heritage, which frequently involved lots of procedural spaghetti code. Frameworks like CoffeeScript and TypeScript solved these issues but came as a bolt-on for those who seriously cared about coding standards. Now with the release and general adoption of ES6, Classes are built into the framework and the code looks syntactically similar to C#, Java, and SWIFT.

- **Cross-platform:**

NodeJS is not only cross-platform but when developed with the correct structure can be packaged into an executable containing all its own dependencies.

- **Multi-Thread:**

One of the most common complaints by people who do not understand node JS is that it is single-threaded and if you have a core CPU it will only run on one core. Node.js is non-blocking which means that all functions are delegated to the event loop and they are executed by different threads. That is handled by Node.js run-time. Node.js does support forking multiple processes. It is important to know that the state is not shared between the master and the forked process. We can pass messages to the forked process and to master the process from the forked process with

function send.

- **It is extremely fast:**

The library of Node.js is fast when it comes to code execution as it is built on the V8 JavaScript Engine of Google Chrome.

- **I/O is asynchronous and Event Driven:**

In case of Node.js, all the APIs are asynchronous which means that its server does not wait for the API to come back with data. Here the server calls the APIs one by one and keeps moving to the next one while it uses a notification mechanism of Events to generate a response from the API which was previously called. This makes it fast too.

- **Highly scalable :**

Node.js uses a single threaded model with event looping. Event mechanism helps the server to respond in a non-blocking manner which eventually makes it highly scalable as opposed to traditional servers which create limited threads to handle requests. Node.js uses a single threaded program and the same program can provide service to a much larger number of requests than traditional servers like Apache HTTP server.

- **No buffering:**

When it comes to uploading audio and video files, Node.js cuts down the processing time significantly. It does not buffer any data and here the application simple gets out the data in chunks.

- **Open source:**

Being an open source, the community of Node.js has come up with a number of amazing models which can be used to add in better capabilities into the Node.js applications.

5.1.2.2 Reason for using Node js

- Good for beginner developers, JavaScript is simple to learn, rich framework (Angular, Node, Backbone, Ember)
- It is fast, due to Google innovative technologies and the event loop
- Ability to keep data in native JSON (object notation) format in your database
- Multiple modules (NPM, Grunt, etc.) and supportive community
- Good to create real-time apps, such as chats and games
- Single free codebase
- Good for data streaming, thus for audio and video files, as example

- Sponsored by Linux Foundation, as well as PayPal, Joylent, Microsoft, Walmart
- Wide range of hosting options
- JS is the longest running language, 99% of developers know some of it

5.2 VNC Viewer:

Virtual network computing (VNC) is a type of remote-control software that makes it possible to control another computer over a network connection. Keystrokes and mouse clicks are transmitted from one computer to another, allowing technical support staff to manage a desktop, server, or other networked device without being in the same physical location.

VNC works on a client/server model: A VNC viewer (or client) is installed on the local computer and connects to the server component, which must be installed on the remote computer. The server transmits a duplicate of the remote computer's display screen to the viewer. It also interprets commands coming from the viewer and carries them out on the remote computer.

VNC is platform independent and is compatible with any operating system. Computers must be networked with TCP/IP and have open ports allowing traffic from the IP addresses of devices that may need to connect.

VNC was developed at AT&T Laboratories. The original VNC source code is open source under the GNU General Public License, and other variations are also available commercially.

5.2.1 Features:

- Intuitive remote control
- Cross platform support
- Attended and unattended access
- Direct and cloud connectivity
- Preinstalled or connect on demand
- Secured by design

- 128 bit AES encryption
- System authentication
- Tried and tested performance
- File transfer, printing and Chat
- Multilingual support
- Online team management

5.2.2 Reason for using VNC Viewer :

- VNC is platform independent.
- Uses for this technology include remote technical support and accessing files on once work computer from ones home computer or vice versa.
- VNC viewer is licensed as freeware for the windows OS from remote desktop software without restriction.
- VNC viewer 6.19.325 is a available to all software users as a free download.

5.3 Implementation Tool - Python - IDLE

IDLE (Integrated Development and Learning Environment) is an integrated development environment (IDE) for Python. The Python installer for Windows contains the IDLE module by default. IDLE is not available by default in Python distributions for Linux. It needs to be installed using the respective package managers. For example, in case of Ubuntu:

$ sudo apt-get install idle

IDLE can be used to execute a single statement just like Python Shell and also to create, modify and execute Python scripts. IDLE provides a fully-featured text editor to create Python scripts that includes features like syntax highlighting, autocompletion and smart indent. It also has a debugger with stepping and breakpoints features.

IDLE (short for Integrated Development Environment or Integrated Development and Learning Environment) is an integrated development environment for Python, which has been bundled with the default implementation of the language since 1.5.2b1. It is packaged as an optional part of the Python packaging with many Linux distributions. It is completely written in Python and the Tkinter GUI toolkit (wrapper functions for Tcl/Tk).

IDLE is intended to be a simple IDE and suitable for beginners, especially in an

educational environment. To that end, it is cross-platform, and avoids feature clutter.

5.3.1 Features :

- Multi-window text editor with syntax highlighting, auto completion, multiple undo, Python colorizing, call tips, smart indent and other.
- Integrated debugger with stepping, persistent breakpoints, and call stack visibility, viewing of global and local namespaces.
- IDLE is Python's Integrated Development and Learning Environment.
- Coded in 100% pure Python, using the tkinter GUI toolkit
- Cross-platform: works mostly the same on Windows, Unix, and Mac OS X
- Python shell window (interactive interpreter) with colorizing of code input, output, and error messages
- Search within any window, replace within editor windows, and search through multiple files (grep)
- Configuration, browsers, and other dialogs
- Built-in Developer Tool

5.4 CODING STYLE

5.4.1 PYTHON

General

- Avoid using names that are too general or too wordy. Strike a good balance between the two.
- Bad: data_structure, my_list, info_map, dictionary_for_the_purpose_of_storing_data_representing_word_definitions
- Good: user_profile, menu_options, word_definitions
- Don't be a jackass and name things "O", "l", or "I"
- When using CamelCase names, capitalize all letters of an abbreviation (e.g. HTTPServer)

Packages

- Package names should be all lower case
- When multiple words are needed, an underscore should separate them
- It is usually preferable to stick to 1 word names

Modules

- Module names should be all lower case
- When multiple words are needed, an underscore should separate them

- It is usually preferable to stick to 1 word names

Classes

- Class names should follow the UpperCaseCamelCase convention
- Python's built-in classes, however are typically lowercase words
- Exception classes should end in "Error"

Global (module-level) Variables

- Global variables should be all lowercase
- Words in a global variable name should be separated by an underscore

Instance Variables

- Instance variable names should be all lower case
- Words in an instance variable name should be separated by an underscore
- Non-public instance variables should begin with a single underscore
- If an instance name needs to be mangled, two underscores may begin its name

Methods

- Method names should be all lower case
- Words in an method name should be separated by an underscore
- Non-public method should begin with a single underscore
- If a method name needs to be mangled, two underscores may begin its name

Method Arguments

- Instance methods should have their first argument named 'self'.
- Class methods should have their first argument named 'cls'

Functions

- Function names should be all lower case
- Words in a function name should be separated by an underscore

Constants

- Constant names must be fully capitalized
- Words in a constant name should be separated by an underscore

5.4.2 Node.js :

Formatting

- **Four spaces for indentation :** Use 4 spaces for indenting your code and swear an oath to never mix tabs and spaces — a special kind of hell awaits otherwise.
- **Newlines :** Use UNIX-style newlines (\n), and a newline character as the last character of a file. Windows-style newlines (\r\n) are forbidden inside any repository.
- **No trailing white space :** Just like you brush your teeth after every meal, you clean up any trailing white space in your `.js` files before committing. Otherwise the rotten smell of careless neglect will eventually drive away contributors and/or co-workers.
- **Use semicolons :** According to scientific research, the usage of semicolons is a core value of our community. Consider the points of the opposition, but be a traditionalist when it comes to abusing error correction mechanisms for cheap syntactic pleasures.
- **Use single quotes :** Use single quotes, unless you are writing JSON. This helps you separate your objects' strings from normal strings.
- **Opening braces go on the same line :** Your opening braces go on the same line as the statement. Also, notice the use of white space before and after the condition statement. What if you want to write 'else' or 'else if' along with your 'if'…
- **Declare one variable per var statement :** Declare one variable per var statement, it makes it easier to re-order the lines.

Naming Conventions

- **Use lowerCamelCase for variables, properties and function names :** Variables, properties and function names should use lowerCamelCase. They should also be

descriptive. Single character variables and uncommon abbreviations should generally be avoided.

- **Use UpperCamelCase for class names :** Class names should be capitalised using UpperCamelCase.

- **Use UPPERCASE for Constants** :Constants should be declared as regular variables or static class properties, using all uppercase letters.

Variables

- **Object / Array creation :** Use trailing commas and put *short* declarations on a single line. Only quote keys when your interpreter complains:

Conditionals

- **Use the === operator :** Programming is not about remembering stupid rules. Use the triple equality operator and it will work just as expected.

- **Use descriptive conditions :** Any non-trivial conditions should be assigned to a descriptively named variable or function:

Functions

- **Write small functions :** Keep your functions short. A good function fits on a slide that the people in the last row of a big room can comfortably read.

- **Return early from functions :** To avoid deep nesting of if-statements, always return a function's value as early as possible. Or for this particular example it may also be okay to shorten things even further.

- **Method chaining :** One method per line should be used if you want to chain methods. You should also indent these methods so it's easier to tell they are part of the same chain.

Comments

- **Use slashes for comments :** Use slashes for both single line and multi line comments. Try to write comments that explain higher level mechanisms or clarify difficult segments of your code. Don't use comments to restate trivial things.

5.5 Form Design and Coding

5.5.1 Snapshots

5.5.1.1 Clock

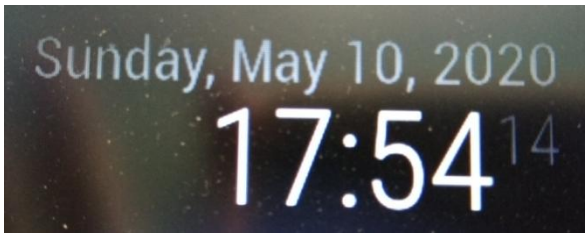


Figure 5.1 Clock

5.5.1.2 Calender

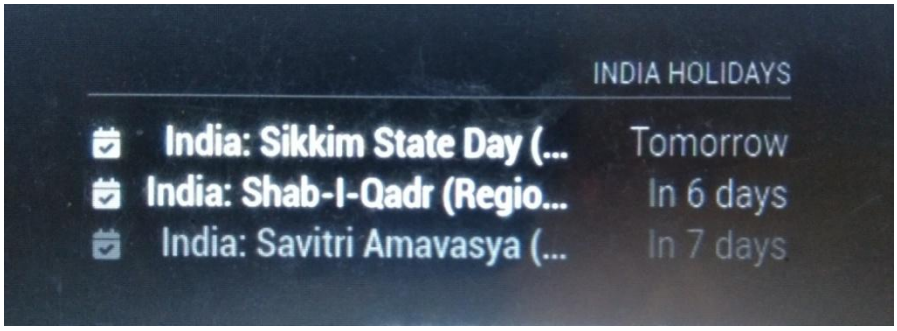


Figure 5.2 Calender

5..5.1.3 Weather Forecast

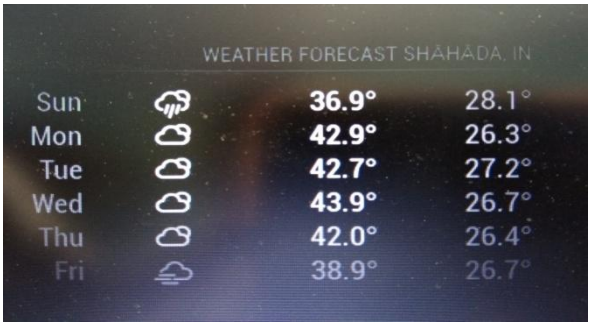


Figure 5.3 Weather Forecast

5.5.1.4 Newsfeed

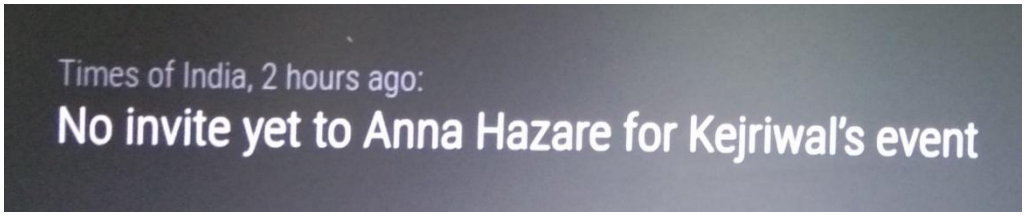
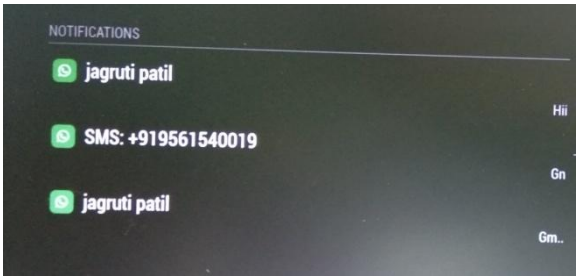


Figure 5.4 Newsfeed

5.5.1.5 Current Weather

Figure 5.5 Current Weather

5.5.1.6 Phone Notifications



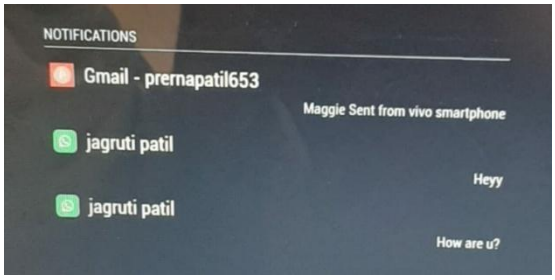


Figure 5.6 Phone Notifications

5.5.2 CODING SNIPPET

//CONFIG

```
/* Magic Mirror Config Sample

*/

var config = {

address: "localhost", // Address to listen on, can be:

// - "localhost", "127.0.0.1", "::1" to listen on loopback interface

// - another specific IPv4/6 to listen on a specific interface

// - "0.0.0.0", "::" to listen on any interface

// Default, when address config is left out or empty, is "localhost"

port: 8080,

ipWhitelist: ["127.0.0.1", "::ffff:127.0.0.1", "::1"], // Set [] to allow all
IP addresses

// or add a specific IPv4 of 192.168.1.5 :
```

```
// ["127.0.0.1", "::ffff:127.0.0.1", "::1", "::ffff:192.168.1.5"],
// or IPv4 range of 192.168.3.0 --> 192.168.3.15 use CIDR format :
// ["127.0.0.1", "::ffff:127.0.0.1", "::1", "::ffff:192.168.3.0/28"],
language: "en",
timeFormat: 24,
units: "metric",
modules: [
{
module: "alert",
},
{
module: "clock",
position: "top_right"
},
{
module: "calendar",
header: "India Holidays",
position: "top_right",
config: {
calendars: [
{
symbol: "calendar-check",
url: "https://www.officeholidays.com/ics/india" }]
maximumEntries: 3
}
},
{
module: "currentweather",
position: "top_left",
config: {
location: "Shahada",
locationID: "1256750", //ID from
```

```
http://bulk.openweathermap.org/sample/city.list.json.gz; unzip the gz file and
find your city

appid: "9420bc8ecbb8c579c4bdc34393d6"

}

},

{

module: "weatherforecast",

position: "top_right",

header: "Weather Forecast",

config: {

location: "Shahada",

locationID: "1256750", //ID from
http://bulk.openweathermap.org/sample/city.list.json.gz; unzip the gz file and
find your city

appid: "9420bc8ecbb8c579c4bdc34393d6"

}

},

{

module: 'MMM-PushbulletNotifications',

header: 'Notifications',

position: 'top_left',

config:{

accessToken: "o.WnID2frnhn3wjkXtDp2frQISMEqVeq1Y",

numberOfNotifications: 3,

filterTargetDeviceName: "",

showPushesSentToAllDevices: true,

onlyAllowCommandsFromSourceDevices: [],

fetchLimitPushBullet: 50;

showPushes: true,

showPushesOnLoad: true,

showDismissedPushes: true,

showMirroredNotifications: true,

onlyShowLastNotificationFromApplication: false,
```

```
showIndividualNotifications: false,
showSMS: true,
showMessage: true,
showIcons: true,
showDateTime: false,
lacalesDateTime: ‘nl-NL’,
playSoundOnNotificationReceived: false,
soundFile: ‘modules/MMM-PushBulletNotifications/sounds/new-message.mp3’,
maxMsgCharacters: 50,
maxHeaderCharacters: 32,
showModuleIfNoNotifications: true,
noNotificationMessage: “No new notifications”,
debugMode: true,
{
module: "newsfeed",
position: "top_left",
config: {
feeds: [
{
title: "Times of India",
url: "http://timesofindia.indiantimes.com/rssfeedstopstories.cms"
}
],
showSourceTitle: true,
showPublishDate: true,
broadcastNewsFeeds: true,
broadcastNewsUpdates: true
}
},
]
};
```

```
/*************** DO NOT EDIT THE LINE BELOW ***************/

if (typeof module !== "undefined") {module.exports = config;}
```

//Modules

//CLOCK

```
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,

showWeek: false,

dateFormat: "dddd, LL",

/* 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,

showSunTimes: false,

showMoonTimes: false,

lat: 47.630539,

lon: -122.344147,

},

// Define required scripts.
```

```
getScripts: function() {

return ["moment.js", "moment-timezone.js", "suncalc.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;

self.second = moment().second();

self.minute = moment().minute();

//Calculate how many ms should pass until next update depending on if seconds
is displayed or not

var delayCalculator = function(reducedSeconds) {

if (self.config.displaySeconds) {

return 1000 - moment().milliseconds();

} else {

return ((60 - reducedSeconds) * 1000) - moment().milliseconds();

}

};

//A recursive timeout function instead of interval to avoid drifting

var notificationTimer = function() {

self.updateDom();

//If seconds is displayed CLOCK_SECOND-notification should be sent (but not
when CLOCK_MINUTE-notification is sent)

if (self.config.displaySeconds) {

self.second = (self.second + 1) % 60;

if (self.second !== 0) {

self.sendNotification("CLOCK_SECOND", self.second);
```

```
setTimeout(notificationTimer, delayCalculator(0));

return;

}

}

//If minute changed or seconds isn't displayed send CLOCK_MINUTE-notification

self.minute = (self.minute + 1) % 60;

self.sendNotification("CLOCK_MINUTE", self.minute);

setTimeout(notificationTimer, delayCalculator(0));

};

//Set the initial timeout with the amount of seconds elapsed as reducedSeconds
so it will trigger when the minute changes

setTimeout(notificationTimer, delayCalculator(self.second));

// Set locale.

moment.locale(config.language);

},

// Override dom generator.

getDom: function() {

var wrapper = document.createElement("div");

/**********************************

* Create wrappers for DIGITAL clock

*/

var dateWrapper = document.createElement("div");

var timeWrapper = document.createElement("div");

var secondsWrapper = document.createElement("sup");

var periodWrapper = document.createElement("span");

var weekWrapper = document.createElement("div");

// Style Wrappers

dateWrapper.className = "date normal medium";

timeWrapper.className = "time bright large light";

secondsWrapper.className = "dimmed";

weekWrapper.className = "week dimmed medium";

// 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();
this.lastDisplayedMinute = now.minute();
if (this.config.timezone) {
now.tz(this.config.timezone);
}
var hourSymbol = "HH";
if (this.config.timeFormat !== 24) {
hourSymbol = "h";
}
if (this.config.clockBold === true) {
timeString = now.format(hourSymbol + "[<span class=\"bold\">]mm[</span>]");
} else {
timeString = now.format(hourSymbol + ":mm");
}
if(this.config.showDate){
dateWrapper.innerHTML = now.format(this.config.dateFormat);
}
if (this.config.showWeek) {
weekWrapper.innerHTML = this.translate("WEEK", { weekNumber: now.week() });
}
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);

}

/************************************************************

* Create wrappers for ANALOG clock, only if specified in config

*/

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 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%";

// clockCircle.style.border = "1px solid black";

clockCircle.style.border = "rgba(0, 0, 0, 0.1)"; //Updated fix for Issue 611
where non-black backgrounds are used

} 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);
}

/*****************************************
* Combine wrappers, check for .displayType
*/
if (this.config.displayType === "digital") {
// Display only a digital clock
wrapper.appendChild(dateWrapper);
wrapper.appendChild(timeWrapper);
```

```
wrapper.appendChild(sunWrapper);
wrapper.appendChild(moonWrapper);
wrapper.appendChild(weekWrapper);
} else if (this.config.displayType === "analog") {
// Display only an analog clock
if (this.config.showWeek) {
weekWrapper.style.paddingBottom = "15px";
} else {
dateWrapper.style.paddingBottom = "15px";
}
if (this.config.analogShowDate === "top") {
wrapper.appendChild(dateWrapper);
wrapper.appendChild(weekWrapper);
wrapper.appendChild(clockCircle);
} else if (this.config.analogShowDate === "bottom") {
wrapper.appendChild(clockCircle);
wrapper.appendChild(dateWrapper);
wrapper.appendChild(weekWrapper);
} 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);
digitalWrapper.appendChild(weekWrapper);
var appendClocks = function(condition, pos1, pos2) {
var padding = [0,0,0,0];
padding[(placement === condition) ? pos1 : pos2] = "20px";
analogWrapper.style.padding = padding.join(" ");
if (placement === condition) {
wrapper.appendChild(analogWrapper);
wrapper.appendChild(digitalWrapper);
} else {
wrapper.appendChild(digitalWrapper);
wrapper.appendChild(analogWrapper);
}
};
if (placement === "left" || placement === "right") {
digitalWrapper.style.display = "inline-block";
digitalWrapper.style.verticalAlign = "top";
analogWrapper.style.display = "inline-block";
appendClocks("left", 1, 3);
} else {
digitalWrapper.style.textAlign = "center";
appendClocks("top", 2, 0);
}
}
// Return the wrapper to the dom.
return wrapper;
}
});
```

//CALENDER

```
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/

showLocation: false,

displayRepeatingCountTitle: false,

defaultRepeatingCountTitle: "",

maxTitleLength: 25,

wrapEvents: false, // wrap events to multiple lines breaking at maxTitleLength

maxTitleLines: 3,

fetchInterval: 5 * 60 * 1000, // Update every 5 minutes.

animationSpeed: 2000,

fade: true,

urgency: 7,

timeFormat: "relative",

dateFormat: "MMM Do",

dateEndFormat: "LT",

fullDayEventDateFormat: "MMM Do",

showEnd: false,

getRelative: 6,

fadePoint: 0.25, // Start on 1/4th of the list.

hidePrivate: false,

hideOngoing: false,

colored: false,

coloredSymbolOnly: false,

tableClass: "small",

calendars: [

{
```

```
symbol: "calendar",

url: "http://www.calendarlabs.com/templates/ical/US-Holidays.ics",

},

],

titleReplace: {

"De verjaardag van ": "",

"'s birthday": ""

},

broadcastEvents: true,

excludedEvents: [],

sliceMultiDayEvents: false,

broadcastPastEvents: false,

nextDaysRelative: false

},

// 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 default modules are defined in the core translation
files.

// Therefor we can just return false. Otherwise we should have returned a
dictionary.

// 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.updateLocale(config.language,
this.getLocaleSpecification(config.timeFormat));

for (var c in this.config.calendars) {

var calendar = this.config.calendars[c];

calendar.url = calendar.url.replace("webcal://", "http://");

var calendarConfig = {

maximumEntries: calendar.maximumEntries,

maximumNumberOfDays: calendar.maximumNumberOfDays,

broadcastPastEvents: calendar.broadcastPastEvents,

};

if (calendar.symbolClass === "undefined" || calendar.symbolClass === null) {

calendarConfig.symbolClass = "";

}

if (calendar.titleClass === "undefined" || calendar.titleClass === null) {

calendarConfig.titleClass = "";

}

if (calendar.timeClass === "undefined" || calendar.timeClass === null) {

calendarConfig.timeClass = "";

}

// we check user and password here for backwards compatibility with old configs

if(calendar.user && calendar.pass) {

Log.warn("Deprecation warning: Please update your calendar authentication
configuration.");

Log.warn("https://github.com/MichMich/MagicMirror/tree/v2.1.2/modules/default/c
alendar#calendar-authentication-options");

calendar.auth = {

user: calendar.user,

pass: calendar.pass

};

}
```

```
this.addCalendar(calendar.url, calendar.auth, calendarConfig);

// Trigger ADD_CALENDAR every fetchInterval to make sure there is always a
calendar

// fetcher running on the server side.

var self = this;

setInterval(function() {

self.addCalendar(calendar.url, calendar.auth, calendarConfig);

}, self.config.fetchInterval);

}

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);

this.loaded = true;

} else if (notification === "INCORRECT_URL") {

Log.error("Calendar Error. Incorrect url: " + payload.url);

}

this.updateDom(this.config.animationSpeed);

},

// Override dom generator.

getDom: function () {
```

```
var events = this.createEventList();

var wrapper = document.createElement("table");

wrapper.className = this.config.tableClass;

if (events.length === 0) {

wrapper.innerHTML = (this.loaded) ? this.translate("EMPTY") :
this.translate("LOADING");

wrapper.className = this.config.tableClass + " dimmed";

return wrapper;

}

if (this.config.fade && this.config.fadePoint < 1) {

if (this.config.fadePoint < 0) {

this.config.fadePoint = 0;

}

var startFade = events.length * this.config.fadePoint;

var fadeSteps = events.length - startFade;

}

var currentFadeStep = 0;

var lastSeenDate = "";

for (var e in events) {

var event = events[e];

var dateAsString = moment(event.startDate, "x").format(this.config.dateFormat);

if(this.config.timeFormat === "dateheaders"){

if(lastSeenDate !== dateAsString){

var dateRow = document.createElement("tr");

dateRow.className = "normal";

var dateCell = document.createElement("td");

dateCell.colSpan = "3";

dateCell.innerHTML = dateAsString;

dateCell.style.paddingTop = "10px";

dateRow.appendChild(dateCell);

wrapper.appendChild(dateRow);

if (e >= startFade) {                //fading
```

```
currentFadeStep = e - startFade;
dateRow.style.opacity = 1 - (1 / fadeSteps * currentFadeStep);
}
lastSeenDate = dateAsString;
}
}
var eventWrapper = document.createElement("tr");
if (this.config.colored && !this.config.coloredSymbolOnly) {
eventWrapper.style.cssText = "color:" + this.colorForUrl(event.url);
}
eventWrapper.className = "normal";
if (this.config.displaySymbol) {
var symbolWrapper = document.createElement("td");
if (this.config.colored && this.config.coloredSymbolOnly) {
symbolWrapper.style.cssText = "color:" + this.colorForUrl(event.url);
}
var symbolClass = this.symbolClassForUrl(event.url);
symbolWrapper.className = "symbol align-right " + symbolClass;
var symbols = this.symbolsForUrl(event.url);
if(typeof symbols === "string") {
symbols = [symbols];
}
for(var i = 0; i < symbols.length; i++) {
var symbol = document.createElement("span");
symbol.className = "fa fa-fw fa-" + symbols[i];
if(i > 0){
symbol.style.paddingLeft = "5px";
}
symbolWrapper.appendChild(symbol);
}
eventWrapper.appendChild(symbolWrapper);
} else if(this.config.timeFormat === "dateheaders"){
```

```
var blankCell = document.createElement("td");

blankCell.innerHTML = "   ";

eventWrapper.appendChild(blankCell);

}

var titleWrapper = document.createElement("td"),

repeatingCountTitle = "";

if (this.config.displayRepeatingCountTitle && event.firstYear !== undefined) {

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;

var titleClass = this.titleClassForUrl(event.url);

if (!this.config.colored) {

titleWrapper.className = "title bright " + titleClass;

} else {

titleWrapper.className = "title " + titleClass;

}

if(this.config.timeFormat === "dateheaders"){

if (event.fullDayEvent) {

titleWrapper.colSpan = "2";

titleWrapper.align = "left";

} else {

var timeClass = this.timeClassForUrl(event.url);

var timeWrapper = document.createElement("td");

timeWrapper.className = "time light " + timeClass;

timeWrapper.align = "left";

timeWrapper.style.paddingLeft = "2px";
```

```
timeWrapper.innerHTML = moment(event.startDate, "x").format("LT");

eventWrapper.appendChild(timeWrapper);

titleWrapper.align = "right";

}

eventWrapper.appendChild(titleWrapper);

} else {

var timeWrapper = document.createElement("td");

eventWrapper.appendChild(titleWrapper);

//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) {

//subtract one second so that fullDayEvents end at 23:59:59, and not at 0:00:00
one the next day

event.endDate -= oneSecond;

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 {

/* Check to see if the user displays absolute or relative dates with their
events

* Also check to see if an event is happening within an 'urgency' time
```

```
frameElement

* For example, if the user set an .urgency of 7 days, those events that fall
within that

* time frame will be displayed with 'in xxx' time format or moment.fromNow()

*

* Note: this needs to be put in its own function, as the whole thing repeats
again verbatim

*/

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").from(moment().format("YYYYMMDD")));

} else {

timeWrapper.innerHTML = this.capFirst(moment(event.startDate,
"x").format(this.config.fullDayEventDateFormat));

}

} else {

timeWrapper.innerHTML = this.capFirst(moment(event.startDate,
"x").from(moment().format("YYYYMMDD")));

}

}

if(this.config.showEnd){

timeWrapper.innerHTML += "-" ;

timeWrapper.innerHTML += this.capFirst(moment(event.endDate ,
"x").format(this.config.fullDayEventDateFormat));

}

} 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 {

if(this.config.timeFormat === "absolute" && !this.config.nextDaysRelative) {

timeWrapper.innerHTML = this.capFirst(moment(event.startDate,
"x").format(this.config.dateFormat));

} else {

// Otherwise just say 'Today/Tomorrow at such-n-such time'

timeWrapper.innerHTML = this.capFirst(moment(event.startDate, "x").calendar());

}

}

} else {

/* Check to see if the user displays absolute or relative dates with their
events

* Also check to see if an event is happening within an 'urgency' time
frameElement

* For example, if the user set an .urgency of 7 days, those events that fall
within that

* time frame will be displayed with 'in xxx' time format or moment.fromNow()

*

* Note: this needs to be put in its own function, as the whole thing repeats
again verbatim

*/

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", {

fallback: this.translate("RUNNING") + " {timeUntilEnd}",

timeUntilEnd: moment(event.endDate, "x").fromNow(true)

})

);

}

if (this.config.showEnd) {

timeWrapper.innerHTML += "-";

timeWrapper.innerHTML += this.capFirst(moment(event.endDate,
"x").format(this.config.dateEndFormat));

}

}

//timeWrapper.innerHTML += ' - '+ moment(event.startDate,'x').format('lll');

//console.log(event);

var timeClass = this.timeClassForUrl(event.url);

timeWrapper.className = "time light " + timeClass;

eventWrapper.appendChild(timeWrapper);

}

wrapper.appendChild(eventWrapper);

// Create fade effect.

if (e >= startFade) {

currentFadeStep = e - startFade;

eventWrapper.style.opacity = 1 - (1 / fadeSteps * currentFadeStep);

}

if (this.config.showLocation) {

if (event.location !== false) {

var locationRow = document.createElement("tr");

locationRow.className = "normal xsmall light";

if (this.config.displaySymbol) {

var symbolCell = document.createElement("td");
```

```
locationRow.appendChild(symbolCell);

}

var descCell = document.createElement("td");

descCell.className = "location";

descCell.colSpan = "2";

descCell.innerHTML = event.location;

locationRow.appendChild(descCell);

wrapper.appendChild(locationRow);

if (e >= startFade) {

currentFadeStep = e - startFade;

locationRow.style.opacity = 1 - (1 / fadeSteps * currentFadeStep);

}

}

}

}

return wrapper;

},

/**

* This function accepts a number (either 12 or 24) and returns a moment.js
LocaleSpecification with the

* corresponding timeformat to be used in the calendar display. If no number is
given (or otherwise invalid input)

* it will a localeSpecification object with the system locale time format.

*

* @param {number} timeFormat Specifies either 12 or 24 hour time format

* @returns {moment.LocaleSpecification}

*/

getLocaleSpecification: function(timeFormat) {

switch (timeFormat) {

case 12: {

return { longDateFormat: {LT: "h:mm A"} };

break;
```

```
}
case 24: {
return { longDateFormat: {LT: "HH:mm"} };
break;
}
default: {
return { longDateFormat: {LT: moment.localeData().longDateFormat("LT")} };
break;
}
}
},

/* hasCalendarURL(url)
* Check if this config contains the calendar url.
*
* argument url string - Url to look for.
*
* return bool - Has calendar url
*/
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()
* Creates the sorted list of all events.
*
```

```
* return array - Array with events.

*/

createEventList: function () {

var events = [];

var today = moment().startOf("day");

var now = new Date();

var future = moment().startOf("day").add(this.config.maximumNumberOfDays,
"days").toDate();

for (var c in this.calendarData) {

var calendar = this.calendarData[c];

for (var e in calendar) {

var event = JSON.parse(JSON.stringify(calendar[e])); // clone object

if(event.endDate < now) {

continue;

}

if(this.config.hidePrivate) {

if(event.class === "PRIVATE") {

// do not add the current event, skip it

continue;

}

}if(this.config.hideOngoing) {

if(event.startDate < now) {

continue;

}

}

if(this.listContainsEvent(events,event)){

continue;

}

event.url = c;

event.today = event.startDate >= today && event.startDate < (today + 24 * 60 *
60 * 1000);

/* if sliceMultiDayEvents is set to true, multiday events (events exceeding at
least one midnight) are sliced into days,
```

```
* otherwise, esp. in dateheaders mode it is not clear how long these events
are.

*/

var maxCount = Math.ceil(((event.endDate - 1) - moment(event.startDate,
"x").endOf("day").format("x"))/(1000*60*60*24)) + 1;

if (this.config.sliceMultiDayEvents && maxCount > 1) {

var splitEvents = [];

var midnight = moment(event.startDate, "x").clone().startOf("day").add(1,
"day").format("x");

var count = 1;

while (event.endDate > midnight) {

var thisEvent = JSON.parse(JSON.stringify(event)); // clone object

thisEvent.today = thisEvent.startDate >= today && thisEvent.startDate < (today
+ 24 * 60 * 60 * 1000);

thisEvent.endDate = midnight;

thisEvent.title += " (" + count + "/" + maxCount + ")";

splitEvents.push(thisEvent);

event.startDate = midnight;

count += 1;

midnight = moment(midnight, "x").add(1, "day").format("x"); // next day

}

// Last day

event.title += " ("+count+"/"+maxCount+")";

splitEvents.push(event);

for (event of splitEvents) {

if ((event.endDate > now) && (event.endDate <= future)) {

events.push(event);

}

}

} else {

events.push(event);

}

}
```

```
}

events.sort(function (a, b) {

return a.startDate - b.startDate;

});

return events.slice(0, this.config.maximumEntries);

},

listContainsEvent: function(eventList, event){

for(var evt of eventList){

if(evt.title === event.title && parseInt(evt.startDate) ===
parseInt(event.startDate)){

return true;

}

}

return false;

},

/* createEventList(url)

* Requests node helper to add calendar url.

*

* argument url string - Url to add.

*/

addCalendar: function (url, auth, calendarConfig) {

this.sendSocketNotification("ADD_CALENDAR", {

url: url,

excludedEvents: calendarConfig.excludedEvents || this.config.excludedEvents,

maximumEntries: calendarConfig.maximumEntries || this.config.maximumEntries,

maximumNumberOfDays: calendarConfig.maximumNumberOfDays ||
this.config.maximumNumberOfDays,

fetchInterval: this.config.fetchInterval,

symbolClass: calendarConfig.symbolClass,

titleClass: calendarConfig.titleClass,

timeClass: calendarConfig.timeClass,

auth: auth,

broadcastPastEvents: calendarConfig.broadcastPastEvents ||
```

```
this.config.broadcastPastEvents,
});
},
/**
* symbolsForUrl(url)
* Retrieves the symbols for a specific url.
*
* argument url string - Url to look for.
*
* return string/array - The Symbols
*/
symbolsForUrl: function (url) {
return this.getCalendarProperty(url, "symbol", this.config.defaultSymbol);
},
/**
* symbolClassForUrl(url)
* Retrieves the symbolClass for a specific url.
*
* @param url string - Url to look for.
*
* @returns string
*/
symbolClassForUrl: function (url) {
return this.getCalendarProperty(url, "symbolClass", "");
},
/**
* titleClassForUrl(url)
* Retrieves the titleClass for a specific url.
*
* @param url string - Url to look for.
*
```

```
* @returns string
*/
titleClassForUrl: function (url) {
return this.getCalendarProperty(url, "titleClass", "");
},
/**
* timeClassForUrl(url)
* Retrieves the timeClass for a specific url.
*
* @param url string - Url to look for.
*
* @returns string
*/
timeClassForUrl: function (url) {
return this.getCalendarProperty(url, "timeClass", "");
},
/* calendarNameForUrl(url)
* Retrieves the calendar name for a specific url.
*
* argument url string - Url to look for.
*
* return string - The name of the calendar
*/
calendarNameForUrl: function (url) {
return this.getCalendarProperty(url, "name", "");
},
/* colorForUrl(url)
* Retrieves the color for a specific url.
*
* argument url string - Url to look for.
*
* return string - The Color
```

```
*/
colorForUrl: function (url) {
return this.getCalendarProperty(url, "color", "#fff");
},
/* countTitleForUrl(url)
* Retrieves the name for a specific url.
*
* argument url string - Url to look for.
*
* return string - The Symbol
*/
countTitleForUrl: function (url) {
return this.getCalendarProperty(url, "repeatingCountTitle",
this.config.defaultRepeatingCountTitle);
},
/* getCalendarProperty(url, property, defaultValue)
* Helper method to retrieve the property for a specific url.
*
* argument url string - Url to look for.
* argument property string - Property to look for.
* argument defaultValue string - Value if property is not found.
*
* return string - The Property
*/
getCalendarProperty: function (url, property, defaultValue) {
for (var c in this.config.calendars) {
var calendar = this.config.calendars[c];
if (calendar.url === url && calendar.hasOwnProperty(property)) {
return calendar[property];
}
}
return defaultValue;
```

```
},

/**

* Shortens a string if it's longer than maxLength and add a ellipsis to the end

*

* @param {string} string Text string to shorten

* @param {number} maxLength The max length of the string

* @param {boolean} wrapEvents Wrap the text after the line has reached
maxLength

* @param {number} maxTitleLines The max number of vertical lines before cutting
event title

* @returns {string} The shortened string

*/

shorten: function (string, maxLength, wrapEvents, maxTitleLines) {

if (typeof string !== "string") {

return "";

}

if (wrapEvents === true) {

var temp = "";

var currentLine = "";

var words = string.split(" ");

var line = 0;

for (var i = 0; i < words.length; i++) {

var word = words[i];

if (currentLine.length + word.length < (typeof maxLength === "number" ?
maxLength : 25) - 1) { // max - 1 to account for a space

currentLine += (word + " ");

} else {

line++;

if (line > maxTitleLines - 1) {

if (i < words.length) {

currentLine += "&hellip;";

}

break;
```

```
}
if (currentLine.length > 0) {
temp += (currentLine + "<br>" + word + " ");
} else {
temp += (word + "<br>");
}
currentLine = "";
}
}
return (temp + currentLine).trim();
} else {
if (maxLength && typeof maxLength === "number" && string.length > maxLength) {
return string.trim().slice(0, maxLength) + "&hellip;";
} else {
return string.trim();
}
}
},
/* capFirst(string)
* Capitalize the first letter of a string
* Return capitalized string
*/
capFirst: function (string) {
return string.charAt(0).toUpperCase() + string.slice(1);
},
/* titleTransform(title)
* Transforms the title of an event for usage.
* Replaces parts of the text as defined in config.titleReplace.
* Shortens title based on config.maxTitleLength and config.wrapEvents
*
* argument title string - The title to transform.
```

```
*
* return string - The transformed title.
*/
titleTransform: function (title) {
for (var needle in this.config.titleReplace) {
var replacement = this.config.titleReplace[needle];
var regParts = needle.match(/^\/(.+)\/([gim]*)$/);
if (regParts) {
// the parsed pattern is a regexp.
needle = new RegExp(regParts[1], regParts[2]);
}
title = title.replace(needle, replacement);
}
title = this.shorten(title, this.config.maxTitleLength, this.config.wrapEvents,
this.config.maxTitleLines);
return title;
},
/* broadcastEvents()
* Broadcasts the events to all other modules for reuse.
* The all events available in one array, sorted on startdate.
*/
broadcastEvents: function () {
var eventList = [];
for (var url in this.calendarData) {
var calendar = this.calendarData[url];
for (var e in calendar) {
var event = cloneObject(calendar[e]);
event.symbol = this.symbolsForUrl(url);
event.calendarName = this.calendarNameForUrl(url);
event.color = this.colorForUrl(url);
delete event.url;
eventList.push(event);
```

```
}
}
eventList.sort(function(a,b) {
return a.startDate - b.startDate;
});
this.sendNotification("CALENDAR_EVENTS", eventList);
}
});
```

//CURRENT WEATHER

```
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,
showWindDirectionAsArrow: false,
useBeaufort: true,
appendLocationNameToHeader: false,
useKMPHwind: false,
lang: config.language,
decimalSymbol: ".",
showHumidity: false,
degreeLabel: false,
```

```
showIndoorTemperature: false,

showIndoorHumidity: false,

showFeelsLike: true,

initialLoadDelay: 0, // 0 seconds delay

retryDelay: 2500,

apiVersion: "2.5",

apiBase: "https://api.openweathermap.org/data/",

weatherEndpoint: "weather",

appendLocationNameToHeader: true,

calendarClass: "calendar",

tableClass: "large",

onlyTemp: false,

hideTemp: 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",

"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 calendar event. Used if no location
is specified.

firstEvent: false,

// create a variable to hold the location name based on the API result.

fetchedLocationName: "",

// 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 default modules are defined in the core translation
files.

// Therefor we can just return false. Otherwise we should have returned a
dictionary.

// If you're trying to build your 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.windSpeed = null;

this.windDirection = null;
```

```
this.windDeg = null;

this.sunriseSunsetTime = null;

this.sunriseSunsetIcon = null;

this.temperature = null;

this.indoorTemperature = null;

this.indoorHumidity = null;

this.weatherType = null;

this.feelsLike = 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");

if (this.config.showWindDirectionAsArrow) {

if(this.windDeg !== null) {

windDirection.innerHTML = "  <i class=\"fa fa-long-arrow-down\"
style=\"transform:rotate("+this.windDeg+"deg);\"></i> ";

}

} else {

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");

wrapper.className = this.config.tableClass;

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 = "light";

var degreeLabel = "";

if (this.config.units === "metric" || this.config.units === "imperial") {

degreeLabel += "°";

}

if(this.config.degreeLabel) {

switch(this.config.units) {

case "metric":

degreeLabel += "C";

break;

case "imperial":

degreeLabel += "F";

break;

case "default":

degreeLabel += "K";

break;

}

}

if (this.config.decimalSymbol === "") {

this.config.decimalSymbol = ".";
```

```
}

if (this.config.hideTemp === false) {

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.replace(".",
this.config.decimalSymbol) + degreeLabel;

large.appendChild(temperature);

}

if (this.config.showIndoorTemperature && this.indoorTemperature) {

var indoorIcon = document.createElement("span");

indoorIcon.className = "fa fa-home";

large.appendChild(indoorIcon);

var indoorTemperatureElem = document.createElement("span");

indoorTemperatureElem.className = "bright";

indoorTemperatureElem.innerHTML = " " + this.indoorTemperature.replace(".",
this.config.decimalSymbol) + degreeLabel;

large.appendChild(indoorTemperatureElem);

}

if (this.config.showIndoorHumidity && this.indoorHumidity) {

var indoorHumidityIcon = document.createElement("span");

indoorHumidityIcon.className = "fa fa-tint";

large.appendChild(indoorHumidityIcon);

var indoorHumidityElem = document.createElement("span");

indoorHumidityElem.className = "bright";

indoorHumidityElem.innerHTML = " " + this.indoorHumidity + "%";

large.appendChild(indoorHumidityElem);

}

wrapper.appendChild(large);

if (this.config.showFeelsLike && this.config.onlyTemp === false){
```

```
var small = document.createElement("div");

small.className = "normal medium";

var feelsLike = document.createElement("span");

feelsLike.className = "dimmed";

feelsLike.innerHTML = this.translate("FEELS") + " " + this.feelsLike +
degreeLabel;

small.appendChild(feelsLike);

wrapper.appendChild(small);

}

return wrapper;

},

// Override getHeader method.

getHeader: function() {

if (this.config.appendLocationNameToHeader && this.data.header !== undefined) {

return this.data.header + " " + this.fetchedLocationName;

}

if (this.config.useLocationAsHeader && this.config.location !== false) {

return this.config.location;

}

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) {

this.firstEvent = false;
```

```
for (var e in payload) {
var event = payload[e];
if (event.location || event.geo) {
this.firstEvent = event;
//Log.log("First upcoming event with location: ", event);
break;
}
}
}
}
if (notification === "INDOOR_TEMPERATURE") {
this.indoorTemperature = this.roundValue(payload);
this.updateDom(this.config.animationSpeed);
}
if (notification === "INDOOR_HUMIDITY") {
this.indoorHumidity = this.roundValue(payload);
this.updateDom(this.config.animationSpeed);
}
},
/* updateWeather(compliments)
* Requests new data from openweather.org.
* Calls processWeather on succesfull response.
*/
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(compliments)
* Generates an url with api parameters based on the config.
*
* return String - URL params.
*/
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(data)
* Uses the received data to set the various values.
*
* argument data object - Weather information received form openweather.org.
*/
processWeather: function(data) {
if (!data || !data.main || typeof data.main.temp === "undefined") {
// Did not receive usable new data.
// Maybe this needs a better check?
return;
}
this.humidity = parseFloat(data.main.humidity);
this.temperature = this.roundValue(data.main.temp);
this.fetchedLocationName = data.name;
this.feelsLike = 0;
if (this.config.useBeaufort){
this.windSpeed = this.ms2Beaufort(this.roundValue(data.wind.speed));
} else if (this.config.useKMPHwind) {
this.windSpeed = parseFloat((data.wind.speed * 60 * 60) / 1000).toFixed(0);
```

```
} else {

this.windSpeed = parseFloat(data.wind.speed).toFixed(0);

}

// ONLY WORKS IF TEMP IN C //

var windInMph = parseFloat(data.wind.speed * 2.23694);

var tempInF = 0;

switch (this.config.units){

case "metric": tempInF = 1.8 * this.temperature + 32;

break;

case "imperial": tempInF = this.temperature;

break;

case "default":

var tc = this.temperature - 273.15;

tempInF = 1.8 * tc + 32;

break;

}

if (windInMph > 3 && tempInF < 50){

// windchill

var windChillInF = Math.round(35.74+0.6215*tempInF-
35.75*Math.pow(windInMph,0.16)+0.4275*tempInF*Math.pow(windInMph,0.16));

var windChillInC = (windChillInF - 32) * (5/9);

// this.feelsLike = windChillInC.toFixed(0);

switch (this.config.units){

case "metric": this.feelsLike = windChillInC.toFixed(0);

break;

case "imperial": this.feelsLike = windChillInF.toFixed(0);

break;

case "default":

var tc = windChillInC + 273.15;

this.feelsLike = tc.toFixed(0);

break;

}
```

```
} else if (tempInF > 80 && this.humidity > 40){
// heat index
var Hindex = -42.379 + 2.04901523*tempInF + 10.14333127*this.humidity
- 0.22475541*tempInF*this.humidity - 6.83783*Math.pow(10,-3)*tempInF*tempInF
- 5.481717*Math.pow(10,-2)*this.humidity*this.humidity
+ 1.22874*Math.pow(10,-3)*tempInF*tempInF*this.humidity
+ 8.5282*Math.pow(10,-4)*tempInF*this.humidity*this.humidity
- 1.99*Math.pow(10,-6)*tempInF*tempInF*this.humidity*this.humidity;
switch (this.config.units){
case "metric": this.feelsLike = parseFloat((Hindex - 32) / 1.8).toFixed(0);
break;
case "imperial": this.feelsLike = Hindex.toFixed(0);
break;
case "default":
var tc = parseFloat((Hindex - 32) / 1.8) + 273.15;
this.feelsLike = tc.toFixed(0);
break;
}
} else {
this.feelsLike = parseFloat(this.temperature).toFixed(0);
}
this.windDirection = this.deg2Cardinal(data.wind.deg);
this.windDeg = 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);
// 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 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()

* Schedule next update.

*

* argument delay number - Milliseconds before next update. If empty,
this.config.updateInterval is used.

*/

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(ms)
* Converts m2 to beaufort (windspeed).
*
* see:
*  http://www.spc.noaa.gov/faq/tornado/beaufort.html
*  https://en.wikipedia.org/wiki/Beaufort_scale#Modern_scale
*
* argument ms number - Windspeed in m/s.
*
* return number - Windspeed in beaufort.
*/
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 > 146.25 && deg <= 168.75) {

return "SSE";

} else if (deg > 168.75 && deg <= 191.25) {

return "S";

} else if (deg > 191.25 && deg <= 213.75) {

return "SSW";

} 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";
```

```
}
},
/* function(temperature)
* Rounds a temperature to 1 decimal or integer (depending on config.roundTemp).
*
* argument temperature number - Temperature.
*
* return string - Rounded Temperature.
*/
roundValue: function(temperature) {
var decimals = this.config.roundTemp ? 0 : 1;
return parseFloat(temperature).toFixed(decimals);
}
});
```

//WEATHER FORECAST

```
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,
decimalSymbol: ".",
fade: true,
```

```
fadePoint: 0.25, // Start on 1/4th of the list.

colored: false,

scale: false,

initialLoadDelay: 2500, // 2.5 seconds delay. This delay is used to keep the
OpenWeather API happy.

retryDelay: 2500,

apiVersion: "2.5",

apiBase: "https://api.openweathermap.org/data/",

forecastEndpoint: "forecast/daily",

appendLocationNameToHeader: true,

calendarClass: "calendar",

tableClass: "small",

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.

fetchedLocationName: "",

// 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 default modules are defined in the core translation
files.

// Therefor we can just return false. Otherwise we should have returned a
dictionary.

// If you're trying to build your 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 = this.config.tableClass;

for (var f in this.forecast) {

var forecast = this.forecast[f];

var row = document.createElement("tr");

if (this.config.colored) {

row.className = "colored";

}

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 degreeLabel = "";

if (this.config.units === "metric" || this.config.units === "imperial") {

degreeLabel += "°";

}

if(this.config.scale) {


switch(this.config.units) {

case "metric":

degreeLabel += "C";

break;

case "imperial":

degreeLabel += "F";

break;

case "default":

degreeLabel = "K";

break;

}

}

if (this.config.decimalSymbol === "" || this.config.decimalSymbol === " ") {

this.config.decimalSymbol = ".";

}

var maxTempCell = document.createElement("td");

maxTempCell.innerHTML = forecast.maxTemp.replace(".",
this.config.decimalSymbol) + degreeLabel;

maxTempCell.className = "align-right bright max-temp";

row.appendChild(maxTempCell);

var minTempCell = document.createElement("td");

minTempCell.innerHTML = forecast.minTemp.replace(".",
this.config.decimalSymbol) + degreeLabel;
```

```
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 {


if(config.units !== "imperial") {

rainCell.innerHTML = parseFloat(forecast.rain).toFixed(1).replace(".",
this.config.decimalSymbol) + " mm";

} else {

rainCell.innerHTML = (parseFloat(forecast.rain) / 25.4).toFixed(2).replace(".",
this.config.decimalSymbol) + " in";

}

}

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.fetchedLocationName;
}

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) {
this.firstEvent = false;
for (var 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(compliments)

* Requests new data from openweather.org.

* Calls processWeather on successful response.

*/

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);

if (self.config.forecastEndpoint === "forecast/daily") {

self.config.forecastEndpoint = "forecast";

Log.warn(self.name + ": Your AppID does not support long term forecasts.
Switching to fallback endpoint.");

}

retry = true;

} else {

Log.error(self.name + ": Could not load weather.");

}

if (retry) {

self.scheduleUpdate((self.loaded) ? -1 : self.config.retryDelay);
```

```
}
}
};
weatherRequest.send();
},
/* getParams(compliments)
* Generates an url with api parameters based on the config.
*
* return String - URL params.
*/
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;
},
/*
* parserDataWeather(data)
```

```
*
* Use the parse to keep the same struct between daily and forecast Endpoint
* from Openweather
*
*/
parserDataWeather: function(data) {
if (data.hasOwnProperty("main")) {
data["temp"] = {"min": data.main.temp_min, "max": data.main.temp_max};
}
return data;
},
/* processWeather(data)
* Uses the received data to set the various values.
*
* argument data object - Weather information received form openweather.org.
*/
processWeather: function(data) {
this.fetchedLocationName = data.city.name + ", " + data.city.country;
this.forecast = [];
var lastDay = null;
var forecastData = {};
for (var i = 0, count = data.list.length; i < count; i++) {
var forecast = data.list[i];
this.parserDataWeather(forecast); // hack issue #1017
var day;
var hour;
if(!!forecast.dt_txt) {
day = moment(forecast.dt_txt, "YYYY-MM-DD hh:mm:ss").format("ddd");
hour = moment(forecast.dt_txt, "YYYY-MM-DD hh:mm:ss").format("H");
} else {
day = moment(forecast.dt, "X").format("ddd");
hour = moment(forecast.dt, "X").format("H");
```

```
}

if (day !== lastDay) {

var forecastData = {

day: day,

icon: this.config.iconTable[forecast.weather[0].icon],

maxTemp: this.roundValue(forecast.temp.max),

minTemp: this.roundValue(forecast.temp.min),

rain: this.processRain(forecast, data.list)

};

this.forecast.push(forecastData);

lastDay = day;

// Stop processing when maxNumberOfDays is reached

if (this.forecast.length === this.config.maxNumberOfDays) {

break;

}

} else {

//Log.log("Compare max: ", forecast.temp.max,
parseFloat(forecastData.maxTemp));

forecastData.maxTemp = forecast.temp.max > parseFloat(forecastData.maxTemp) ?
this.roundValue(forecast.temp.max) : forecastData.maxTemp;

//Log.log("Compare min: ", forecast.temp.min,
parseFloat(forecastData.minTemp));

forecastData.minTemp = forecast.temp.min < parseFloat(forecastData.minTemp) ?
this.roundValue(forecast.temp.min) : forecastData.minTemp;

// Since we don't want an icon from the start of the day (in the middle of the
night)

// we update the icon as long as it's somewhere during the day.

if (hour >= 8 && hour <= 17) {

forecastData.icon = this.config.iconTable[forecast.weather[0].icon];

}

}

}

//Log.log(this.forecast);

this.show(this.config.animationSpeed, {lockString:this.identifier});
```

```
this.loaded = true;

this.updateDom(this.config.animationSpeed);

},

/* scheduleUpdate()

* Schedule next update.

*

* argument delay number - Milliseconds before next update. If empty,
this.config.updateInterval is used.

*/

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(ms)

* Converts m2 to beaufort (windspeed).

*

* see:

*  http://www.spc.noaa.gov/faq/tornado/beaufort.html

*  https://en.wikipedia.org/wiki/Beaufort_scale#Modern_scale

*

* argument ms number - Windspeed in m/s.

*

* return number - Windspeed in beaufort.

*/

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;

},

/* function(temperature)

* Rounds a temperature to 1 decimal or integer (depending on config.roundTemp).

*

* argument temperature number - Temperature.

*

* return string - Rounded Temperature.

*/

roundValue: function(temperature) {

var decimals = this.config.roundTemp ? 0 : 1;

return parseFloat(temperature).toFixed(decimals);

},

/* processRain(forecast, allForecasts)

* Calculates the amount of rain for a whole day even if long term forecasts
isn't available for the appid.

*

* When using the the fallback endpoint forecasts are provided in 3h intervals
and the rain-property is an object instead of number.

* That object has a property "3h" which contains the amount of rain since the
previous forecast in the list.

* This code finds all forecasts that is for the same day and sums the amount of
rain and returns that.

*/

processRain: function(forecast, allForecasts) {
```

```
//If the amount of rain actually is a number, return it

if (!isNaN(forecast.rain)) {

return forecast.rain;

}

//Find all forecasts that is for the same day

var checkDateTime = (!!forecast.dt_txt) ? moment(forecast.dt_txt, "YYYY-MM-DD
hh:mm:ss") : moment(forecast.dt, "X");

var daysForecasts = allForecasts.filter(function(item) {

var itemDateTime = (!!item.dt_txt) ? moment(item.dt_txt, "YYYY-MM-DD hh:mm:ss")
: moment(item.dt, "X");

return itemDateTime.isSame(checkDateTime, "day") && item.rain instanceof
Object;

});

//If no rain this day return undefined so it wont be displayed for this day

if (daysForecasts.length == 0) {

return undefined;

}

//Summarize all the rain from the matching days

return daysForecasts.map(function(item) {

return Object.values(item.rain)[0];

}).reduce(function(a, b) {

return a + b;

}, 0);

}

});
```

//NEWS FEED

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

// Default module config.

defaults: {

feeds: [

{
```

```
title: "Lokmat",
url: "http://info.indiatimes.com/rss/toi.opml",
encoding: "UTF-8" //ISO-8859-1
}
],
showSourceTitle: true,
showPublishDate: true,
broadcastNewsFeeds: true,
broadcastNewsUpdates: true,
showDescription: false,
wrapTitle: true,
wrapDescription: true,
truncDescription: true,
lengthDescription: 400,
hideLoading: false,
reloadInterval: 5 * 60 * 1000, // every 5 minutes
updateInterval: 10 * 1000,
animationSpeed: 2.5 * 1000,
maxNewsItems: 0, // 0 for unlimited
ignoreOldItems: false,
ignoreOlderThan: 24 * 60 * 60 * 1000, // 1 day
removeStartTags: "",
removeEndTags: "",
startTags: [],
endTags: [],
prohibitedWords: [],
scrollLength: 500,
logFeedWarnings: false
},
// Define required scripts.
getScripts: function() {
```

```
return ["moment.js"];

},

// Define required translations.

getTranslations: function() {

// The translations for the default modules are defined in the core translation
files.

// Therefor we can just return false. Otherwise we should have returned a
dictionary.

// If you're trying to build your 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.newsItems = [];

this.loaded = false;

this.activeItem = 0;

this.scrollPosition = 0;

this.registerFeeds();

this.isShowingDescription = this.config.showDescription;

},

// Override socket notification handler.

socketNotificationReceived: function(notification, payload) {

if (notification === "NEWS_ITEMS") {

this.generateFeed(payload);

if (!this.loaded) {

this.scheduleUpdateInterval();

}

this.loaded = true;

}
```

```
},

// Override dom generator.

getDom: function() {

var wrapper = document.createElement("div");

if (this.config.feedUrl) {

wrapper.className = "small bright";

wrapper.innerHTML = this.translate("configuration_changed");

return wrapper;

}

if (this.activeItem >= this.newsItems.length) {

this.activeItem = 0;

}

if (this.newsItems.length > 0) {

// this.config.showFullArticle is a run-time configuration, triggered by
optional notifications

if (!this.config.showFullArticle && (this.config.showSourceTitle ||
this.config.showPublishDate)) {

var sourceAndTimestamp = document.createElement("div");

sourceAndTimestamp.className = "newsfeed-source light small dimmed";

if (this.config.showSourceTitle && this.newsItems[this.activeItem].sourceTitle
!== "") {

sourceAndTimestamp.innerHTML = this.newsItems[this.activeItem].sourceTitle;

}

if (this.config.showSourceTitle && this.newsItems[this.activeItem].sourceTitle
!== "" && this.config.showPublishDate) {

sourceAndTimestamp.innerHTML += ", ";

}

if (this.config.showPublishDate) {

sourceAndTimestamp.innerHTML += moment(new
Date(this.newsItems[this.activeItem].pubdate)).fromNow();

}

if (this.config.showSourceTitle && this.newsItems[this.activeItem].sourceTitle
!== "" || this.config.showPublishDate) {

sourceAndTimestamp.innerHTML += ":";
```

```
}

wrapper.appendChild(sourceAndTimestamp);

}

//Remove selected tags from the beginning of rss feed items (title or
description)

if (this.config.removeStartTags === "title" || this.config.removeStartTags ===
"both") {

for (f=0; f<this.config.startTags.length;f++) {

if
(this.newsItems[this.activeItem].title.slice(0,this.config.startTags[f].length)
=== this.config.startTags[f]) {

this.newsItems[this.activeItem].title =
this.newsItems[this.activeItem].title.slice(this.config.startTags[f].length,thi
s.newsItems[this.activeItem].title.length);

}

}

}

if (this.config.removeStartTags === "description" ||
this.config.removeStartTags === "both") {

if (this.isShowingDescription) {

for (f=0; f<this.config.startTags.length;f++) {

if
(this.newsItems[this.activeItem].description.slice(0,this.config.startTags[f].l
ength) === this.config.startTags[f]) {

this.newsItems[this.activeItem].description =
this.newsItems[this.activeItem].description.slice(this.config.startTags[f].leng
th,this.newsItems[this.activeItem].description.length);

}

}

}

}

//Remove selected tags from the end of rss feed items (title or description)

if (this.config.removeEndTags) {

for (f=0; f<this.config.endTags.length;f++) {

if (this.newsItems[this.activeItem].title.slice(-
this.config.endTags[f].length)===this.config.endTags[f]) {

this.newsItems[this.activeItem].title =
```

```
this.newsItems[this.activeItem].title.slice(0,-this.config.endTags[f].length);

}

}

if (this.isShowingDescription) {

for (f=0; f<this.config.endTags.length;f++) {

if (this.newsItems[this.activeItem].description.slice(-
this.config.endTags[f].length)===this.config.endTags[f]) {

this.newsItems[this.activeItem].description =
this.newsItems[this.activeItem].description.slice(0,-
this.config.endTags[f].length);

}

}

}

}

if(!this.config.showFullArticle){

var title = document.createElement("div");

title.className = "newsfeed-title bright medium light" +
(!this.config.wrapTitle ? " no-wrap" : "");

title.innerHTML = this.newsItems[this.activeItem].title;

wrapper.appendChild(title);

}

if (this.isShowingDescription) {

var description = document.createElement("div");

description.className = "newsfeed-desc small light" +
(!this.config.wrapDescription ? " no-wrap" : "");

var txtDesc = this.newsItems[this.activeItem].description;

description.innerHTML = (this.config.truncDescription ? (txtDesc.length >
this.config.lengthDescription ? txtDesc.substring(0,
this.config.lengthDescription) + "..." : txtDesc) : txtDesc);

wrapper.appendChild(description);

}

if (this.config.showFullArticle) {

var fullArticle = document.createElement("iframe");

fullArticle.className = "";
```

```
fullArticle.style.width = "100vw";

// very large height value to allow scrolling

fullArticle.height = "3000";

fullArticle.style.height = "3000";

fullArticle.style.top = "0";

fullArticle.style.left = "0";

fullArticle.style.border = "none";

fullArticle.src = this.getActiveItemURL();

fullArticle.style.zIndex = 1;

wrapper.appendChild(fullArticle);

}

if (this.config.hideLoading) {

this.show();

}

} else {

if (this.config.hideLoading) {

this.hide();

} else {

wrapper.innerHTML = this.translate("LOADING");

wrapper.className = "small dimmed";

}

}

return wrapper;

},

getActiveItemURL: function() {

return typeof this.newsItems[this.activeItem].url === "string" ?
this.newsItems[this.activeItem].url : this.newsItems[this.activeItem].url.href;

},

/* registerFeeds()

* registers the feeds to be used by the backend.

*/

registerFeeds: function() {
```

```
for (var f in this.config.feeds) {
var feed = this.config.feeds[f];
this.sendSocketNotification("ADD_FEED", {
feed: feed,
config: this.config
});
}
},
/* generateFeed()
* Generate an ordered list of items for this configured module.
*
* attribute feeds object - An object with feeds returned by the node helper.
*/
generateFeed: function(feeds) {
var newsItems = [];
for (var feed in feeds) {
var feedItems = feeds[feed];
if (this.subscribedToFeed(feed)) {
for (var i in feedItems) {
var item = feedItems[i];
item.sourceTitle = this.titleForFeed(feed);
if (!(this.config.ignoreOldItems && ((Date.now() - new Date(item.pubdate)) >
this.config.ignoreOlderThan))) {
newsItems.push(item);
}
}
}
}
newsItems.sort(function(a,b) {
var dateA = new Date(a.pubdate);
var dateB = new Date(b.pubdate);
return dateB - dateA;
```

```
});

if(this.config.maxNewsItems > 0) {

newsItems = newsItems.slice(0, this.config.maxNewsItems);

}

if(this.config.prohibitedWords.length > 0) {

newsItems = newsItems.filter(function(value){

for (var i=0; i < this.config.prohibitedWords.length; i++) {

if
(value["title"].toLowerCase().indexOf(this.config.prohibitedWords[i].toLowerCas
e()) > -1) {

return false;

}

}

return true;

}, this);

}

// get updated news items and broadcast them

var updatedItems = [];

newsItems.forEach(value => {

if (this.newsItems.findIndex(value1 => value1 === value) === -1) {

// Add item to updated items list

updatedItems.push(value);

}

});

// check if updated items exist, if so and if we should broadcast these
updates, then lets do so

if (this.config.broadcastNewsUpdates && updatedItems.length > 0) {

this.sendNotification("NEWS_FEED_UPDATE", {items: updatedItems});

}

this.newsItems = newsItems;

},

/* subscribedToFeed(feedUrl)

* Check if this module is configured to show this feed.
```

```
*
* attribute feedUrl string - Url of the feed to check.
*
* returns bool
*/
subscribedToFeed: function(feedUrl) {
for (var f in this.config.feeds) {
var feed = this.config.feeds[f];
if (feed.url === feedUrl) {
return true;
}
}
return false;
},
/* titleForFeed(feedUrl)
* Returns title for a specific feed Url.
*
* attribute feedUrl string - Url of the feed to check.
*
* returns string
*/
titleForFeed: function(feedUrl) {
for (var f in this.config.feeds) {
var feed = this.config.feeds[f];
if (feed.url === feedUrl) {
return feed.title || "";
}
}
return "";
},
/* scheduleUpdateInterval()
```

```
* Schedule visual update.
*/
scheduleUpdateInterval: function() {
var self = this;
self.updateDom(self.config.animationSpeed);
// Broadcast NewsFeed if needed
if (self.config.broadcastNewsFeeds) {
self.sendNotification("NEWS_FEED", {items: self.newsItems});
}
timer = setInterval(function() {
self.activeItem++;
self.updateDom(self.config.animationSpeed);
// Broadcast NewsFeed if needed
if (self.config.broadcastNewsFeeds) {
self.sendNotification("NEWS_FEED", {items: self.newsItems});
}
}, this.config.updateInterval);
},
/* capitalizeFirstLetter(string)
* Capitalizes the first character of a string.
*
* argument string string - Input string.
*
* return string - Capitalized output string.
*/
capitalizeFirstLetter: function(string) {
return string.charAt(0).toUpperCase() + string.slice(1);
},
resetDescrOrFullArticleAndTimer: function() {
this.isShowingDescription = this.config.showDescription;
this.config.showFullArticle = false;
this.scrollPosition = 0;
```

```
// reset bottom bar alignment

document.getElementsByClassName("region bottom bar")[0].style.bottom = "0";

document.getElementsByClassName("region bottom bar")[0].style.top = "inherit";

if(!timer){

this.scheduleUpdateInterval();

}

},

notificationReceived: function(notification, payload, sender) {

if(notification === "ARTICLE_NEXT"){

var before = this.activeItem;

this.activeItem++;

if (this.activeItem >= this.newsItems.length) {

this.activeItem = 0;

}

this.resetDescrOrFullArticleAndTimer();

Log.info(this.name + " - going from article #" + before + " to #" +
this.activeItem + " (of " + this.newsItems.length + ")");

this.updateDom(100);

} else if(notification === "ARTICLE_PREVIOUS"){

var before = this.activeItem;

this.activeItem--;

if (this.activeItem < 0) {

this.activeItem = this.newsItems.length - 1;

}

this.resetDescrOrFullArticleAndTimer();

Log.info(this.name + " - going from article #" + before + " to #" +
this.activeItem + " (of " + this.newsItems.length + ")");

this.updateDom(100);

}

// if "more details" is received the first time: show article summary, on
second time show full article

else if(notification === "ARTICLE_MORE_DETAILS"){

// full article is already showing, so scrolling down
```

```
if(this.config.showFullArticle === true){

this.scrollPosition += this.config.scrollLength;

window.scrollTo(0, this.scrollPosition);

Log.info(this.name + " - scrolling down");

Log.info(this.name + " - ARTICLE_MORE_DETAILS, scroll position: " +
this.config.scrollLength);

}

else {

this.showFullArticle();

}

} else if(notification === "ARTICLE_SCROLL_UP"){

if(this.config.showFullArticle === true){

this.scrollPosition -= this.config.scrollLength;

window.scrollTo(0, this.scrollPosition);

Log.info(this.name + " - scrolling up");

Log.info(this.name + " - ARTICLE_SCROLL_UP, scroll position: " +
this.config.scrollLength);

}

} else if(notification === "ARTICLE_LESS_DETAILS"){

this.resetDescrOrFullArticleAndTimer();

Log.info(this.name + " - showing only article titles again");

this.updateDom(100);

} else if (notification === "ARTICLE_TOGGLE_FULL"){

if (this.config.showFullArticle){

this.activeItem++;

this.resetDescrOrFullArticleAndTimer();

} else {

this.showFullArticle();

}

} else if (notification === "ARTICLE_INFO_REQUEST"){

this.sendNotification("ARTICLE_INFO_RESPONSE", {

title:  this.newsItems[this.activeItem].title,
```

```
source: this.newsItems[this.activeItem].sourceTitle,

date:   this.newsItems[this.activeItem].pubdate,

desc:   this.newsItems[this.activeItem].description,

url:    this.getActiveItemURL()

});

}

},

showFullArticle: function() {

this.isShowingDescription = !this.isShowingDescription;

this.config.showFullArticle = !this.isShowingDescription;

// make bottom bar align to top to allow scrolling

if(this.config.showFullArticle === true){

document.getElementsByClassName("region bottom bar")[0].style.bottom =
"inherit";

document.getElementsByClassName("region bottom bar")[0].style.top = "-90px";

}

clearInterval(timer);

timer = null;

Log.info(this.name + " - showing " + this.isShowingDescription ? "article
description" : "full article");

this.updateDom(100);

}

});
```

//EMAIL

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

// defaults : [

//         {

//             user: 'a@b.com',

//             password: 'xxx',

//             host: 'jjj.kkk.com',

//             port: 993,
```

```
//             tls: true,
//             authTimeout: 10000,
//             numberOfEmails: 5,
//             fade: true,
//             maxCharacters: 30
//         }
//     ]
payload: [],
start : function(){
console.log("Email module started!");
console.log('MEMEME', this.config);
this.sendSocketNotification('LISTEN_EMAIL',{config: this.config, payload:
this.payload, loaded: this.loaded});
this.loaded = false;
},
socketNotificationReceived: function(notification, payload){
if (notification === 'EMAIL_RESPONSE'){
if(payload){
this.loaded = true;
var that = this;
console.log("NEW PAYLOAD: ", payload);
var payloadIds = that.payload.map(function(m) {return m.id});
payload.forEach(function(m){
if(payloadIds.indexOf(m.id) == -1)
that.payload.push(m);
});
this.payload.sort(function(a,b) {return b.id - a.id; });
this.sendSocketNotification('LISTEN_EMAIL',{config: this.config, payload:
this.payload, loaded: this.loaded});
this.updateDom(2000);
}
}
```

```
},
// Define required scripts.
getStyles: function() {
return ["email.css", "font-awesome.css"];
},
getDom: function(){
var wrapper = document.createElement("table");
wrapper.className = "small";
var that =this;
if(this.payload.length > 0)
{
if (typeof that.config.accounts !== "undefined") {
var indexToRemove = [];
for (var i = 0; i < this.config.accounts.length; i++) {
var maxNumEmails = this.config.accounts[i].numberOfEmails;
var count = 0;
for (var j = 0; j < this.payload.length; j++) {
if (this.payload[j].host === this.config.accounts[i].user) {
count++;
}
if (count > maxNumEmails) {
indexToRemove.push(j);
}
}
}
for (var j = 0; j < this.payload.length; j++) {
if (indexToRemove.indexOf(j) > -1) {
delete this.payload[j];
}
}
this.payload.forEach(function (mailObj) {
var host = mailObj.host.slice(0,1) + '@' +
```

```
mailObj.host.substr(mailObj.host.indexOf('@') + 1)[0];
var name = mailObj.sender[0].name.replace(/['"]+/g, "");
name = name.substring(0, that.config.maxCharacters);
var subject = mailObj.subject.replace(/[\['"\]]+/g, "");
subject = subject.substring(0, that.config.maxCharacters);
var emailWrapper = document.createElement("tr");
emailWrapper.className = "normal";
var senderWrapper = document.createElement("tr");
senderWrapper.className = "normal";
var nameWrapper = document.createElement("td");
nameWrapper.className = "bright";
nameWrapper.setAttribute("data-letters", host);
nameWrapper.innerHTML = name;
senderWrapper.appendChild(nameWrapper);
var addressWrapper = document.createElement("td");
addressWrapper.className = "address xsmall thin dimmed";
addressWrapper.innerHTML = mailObj.sender[0].address;
senderWrapper.appendChild(addressWrapper);
emailWrapper.appendChild(senderWrapper);
var subjectWrapper = document.createElement("tr");
subjectWrapper.className = "light";
subjectWrapper.innerHTML = subject;
emailWrapper.appendChild(subjectWrapper);
wrapper.appendChild(emailWrapper);
// Calculate total possible emails
var totalEmails = 0;
for (var i = 0; i < that.config.accounts.length; i++) {
totalEmails += that.config.accounts[i].numberOfEmails;
}
// Create fade effect.
if (that.config.fade) {
var startingPoint = that.payload.slice(0, totalEmails).length * 0.25;
```

```
var steps = that.payload.slice(0, that.config.numberOfEmails).length -
startingPoint;

if (count >= startingPoint) {

var currentStep = count - startingPoint;

emailWrapper.style.opacity = 1 - (1 / steps * currentStep);

}

}

});

}

}

else{

wrapper.innerHTML = (this.loaded) ? "No new mails" : this.translate("LOADING");

wrapper.className = "small dimmed";

return wrapper;

}

return wrapper;

}

});
```

//PHONE NOTIFICATION

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

defaults: {

accessToken: '',

numberOfNotifications: 5,

displayNotificationIcon: true,

displayMessage: true,

displayCount: false,

alert: false,

fade: true,

maxCharacters: 50,

useEncryption: false,

key: {

password: '',
```

```
ident: ''
}
},
payload: [],
start: function() {
console.log("Phone notifications module started!");
this.sendSocketNotification('LISTEN_PHONE', this.config);
this.loaded = false;
this.originalHeader = this.data.header;
// if(this.config.displayCount) {
//    console.log('OLD_HEADER', this.data.header);
//    this.data.header = this.payload.length + '' + this.originalHeader;
//    console.log('HEADER', this.data.header);
//  }
},
cleanPayload: function(newPayload) {
var application_name = newPayload.application_name;
var that = this;
var dupIndex = 0;
if (this.payload.length > 0) {
this.payload.forEach(function(m) {
// If application_name already exists, increment notification count
if (m.application_name === application_name) {
//m.count++;
that.payload.splice(dupIndex, 1);
}
dupIndex++;
});
}
this.payload.unshift(newPayload);
},
removePayload: function(dismissedPayload) {
```

```
var package_name = dismissedPayload.package_name;
var that = this;
var index = 0;
if (this.payload.length > 0) {
this.payload.forEach(function(m) {
//If package_name exists in Notification list, remove notification
if (m.package_name === package_name) {
that.payload.splice(index, 1);
}
index++;
});
}
},
socketNotificationReceived: function(notification, payload) {
console.log(notification);
if (notification === 'PHONE_RESPONSE') {
if (payload) {
this.loaded = true;
this.cleanPayload(payload);
if (this.config.alert)
this.sendNotification("SHOW_ALERT", {
type: "notification",
title: "New phone notification"
});
this.sendSocketNotification('LISTEN_PHONE', this.config);
this.updateDom();
}
} else if (notification === 'DISMISSAL') {
if (payload) {
this.loaded = true;
this.removePayload(payload);
```

```
this.sendSocketNotification('LISTEN_PHONE', this.config);
this.updateDom();
}
}
},
// Define required scripts.
getStyles: function() {
return ["phone_notification.css", "font-awesome.css"];
},
getDom: function() {
//console.log(this.data.header);
var wrapper = document.createElement("table");
wrapper.className = "small";
var that = this;
if (this.config.displayCount) {
var headerRow = document.createElement("tr");
var headerData = document.createElement("td");
var headerDiv = document.createElement("div");
headerDiv.className = "count";
headerDiv.innerHTML = this.payload.length;
headerData.appendChild(headerDiv);
var headerRest = document.createElement("td");
headerRest.innerHTML = this.originalHeader;
headerRow.appendChild(headerData);
headerRow.appendChild(headerRest);
wrapper.appendChild(headerRow);
}
if (this.payload.length > 0) {
var count = 0;
this.payload.slice(0, this.config.numberOfNotifications).forEach(function(o) {
var name = o.application_name;
//var subject = mailObj.subject.replace(/[\['"\]]+/g,"");
```

```
var notificationWrapper = document.createElement("tr");

notificationWrapper.className = "normal";

if (that.config.displayNotificationIcon) {

var iconWrapper = document.createElement("td");

iconWrapper.className = "icon";

var icon = document.createElement("span");

var iconPath = '/modules/phone_notification/icons/' + o.application_name +
'.jpg';

icon.innerHTML = '<img src="' + iconPath + '" width="25" >';

iconWrapper.appendChild(icon);

notificationWrapper.appendChild(iconWrapper);

}

var nameWrapper = document.createElement("td");

nameWrapper.className = "bright";

nameWrapper.innerHTML = name;

notificationWrapper.appendChild(nameWrapper);

var titleWrapper = document.createElement("td");

titleWrapper.className = "bright";

titleWrapper.innerHTML = o.title;

notificationWrapper.appendChild(titleWrapper);

wrapper.appendChild(notificationWrapper);

if (that.config.displayMessage) {

var bodyWrapper = document.createElement("tr");

var bodyContentWrapper = document.createElement("td");

bodyContentWrapper.colSpan = '3';

bodyContentWrapper.className = "dimmed xsmall address";

bodyContentWrapper.innerHTML = o.body.substring(0, that.config.maxCharacters);

bodyWrapper.appendChild(bodyContentWrapper);

wrapper.appendChild(bodyWrapper);

}

// Create fade effect.

if (that.config.fade) {
```

```
var startingPoint = that.payload.slice(0,
that.config.numberOfNotifications).length * 0.25;

var steps = that.payload.slice(0, that.config.numberOfNotifications).length -
startingPoint;

if (count >= startingPoint) {

var currentStep = count - startingPoint;

notificationWrapper.style.opacity = 1 - (1 / steps * currentStep);

}

}

count++;

});

} else {

wrapper.innerHTML = this.translate("No new notifications");

wrapper.className = "small dimmed";

return wrapper;

}

return wrapper;

}

});
```

//FACE RECOGNITION

//Face _detect_p.py

```
import cv2

cascPath = "C:/Users/patil/PycharmProjects/face_Recognition/venv/Lib/site-
packages/cv2/data/haarcascade_frontalface_default.xml"

face_classifier = cv2.CascadeClassifier(cascPath)

def face_extractor(img):

gary = cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)

faces = face_classifier.detectMultiScale(gary,1.3,5)

if faces is():

return  None

for(x,y,w,h) in faces :
```

```
cropped_face = img[y:y+h ,x:x+w]

return cropped_face

cap = cv2.VideoCapture(0)

count = 0

while True:

ret, frame = cap.read()

if face_extractor(frame) is not None:

count +=1

face = cv2.resize(face_extractor(frame),(200,200))

face = cv2.cvtColor(face,cv2.COLOR_BGR2GRAY)

file_path = 'C:/Users/patil/PycharmProjects/face_Recognition/new
pro/face_samp/user'+str(count)+'.jpg'

cv2.imwrite(file_path,face)

cv2.putText(face,str(count),(50,50),cv2.FONT_HERSHEY_COMPLEX,1,(0,255,0),2)

cv2.imshow("face Cropper",face)

else:

print("face not found")

pass

if cv2.waitKey(1)==13 or count == 70:

break

cap.release()

cv2.destroyAllWindows()

print("dataset collection complete")
```

```
//Face_training_p.py

import cv2

import numpy as np

from os  import listdir

from os.path import isfile,join

file_path = 'C:/Users/patil/PycharmProjects/face_Recognition/new
pro/face_samp/'

onlyfiles = [f for f in listdir(file_path) if isfile(join(file_path,f))]
```

```
Training_Data, Labels = [], []

for i, files in enumerate(onlyfiles):

image_path = file_path + onlyfiles[i]

images = cv2.imread(image_path, cv2.IMREAD_GRAYSCALE)

Training_Data.append(np.asarray(images, dtype=np.uint8))

Labels.append(i)

Labels = np.asarray(Labels , dtype=np.int32 )

model = cv2.face.LBPHFaceRecognizer_create()

model.train(np.asarray(Training_Data), np.asarray(Labels))

model.save("C:/Users/patil/PycharmProjects/face_Recognition/new
pro/train_model/Trainner.yml")

print("model training complete")
```

//Face_recog_p.py

```
import cv2

import yagmail

from time import sleep

import numpy as np

from os  import listdir

from os.path import isfile,join\

file_path = 'C:/Users/patil/PycharmProjects/face_Recognition/new
pro/face_samp/'

onlyfiles = [f for f in listdir(file_path) if isfile(join(file_path,f))]

Training_Data, Labels = [], []

for i, files in enumerate(onlyfiles):

image_path = file_path + onlyfiles[i]

images = cv2.imread(image_path, cv2.IMREAD_GRAYSCALE)

Training_Data.append(np.asarray(images, dtype=np.uint8))

Labels.append(i)

Labels = np.asarray(Labels , dtype=np.int32 )

model = cv2.face.LBPHFaceRecognizer_create()
```

```
model.train(np.asarray(Training_Data), np.asarray(Labels))

print("model tarining complete")

cascPath = "C:/Users/patil/PycharmProjects/face_Recognition/venv/Lib/site-
packages/cv2/data/haarcascade_frontalface_default.xml"

face_classifier = cv2.CascadeClassifier(cascPath)

def face_detector(img, size = 0.5):

gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

faces = face_classifier.detectMultiScale(gray,1.3,5)

if faces is():

return img,[]

for(x,y,w,h) in faces:

cv2.rectangle(img, (x,y),(x+w,y+h),(0,255,255),2)

roi = img[y:y+h, x:x+w]

roi = cv2.resize(roi, (200,200))

return img,roi

cap = cv2.VideoCapture(0)

count = 0

while True:

ret, frame = cap.read()

image, face = face_detector(frame)

try:

face = cv2.cvtColor(face, cv2.COLOR_BGR2GRAY)

result = model.predict(face)

if result[1] < 500:

confidence = int(100*(1-(result[1])/300))

#display_string = str(confidence)+'% Confidence it is user'

#cv2.putText(image,display_string,(100,120),
cv2.FONT_HERSHEY_COMPLEX,1,(250,120,255),2)

if confidence > 80:

cv2.putText(image, "Unlocked", (250, 450), cv2.FONT_HERSHEY_COMPLEX, 1, (0,
255, 0), 2)

cv2.imshow('Face Cropper', image)

else:
```

```
cv2.putText(image, "Locked", (250, 450), cv2.FONT_HERSHEY_COMPLEX, 1, (0, 0,
255), 2)

cv2.imshow('Face Cropper', image)

count +=1

file_path = 'C:/Users/patil/PycharmProjects/face_Recognition/new
pro/unknown_face/unknown'+str(count)+'.jpg'

cv2.imwrite(file_path, image)

filename ='C:/Users/patil/PycharmProjects/face_Recognition/new
pro/unknown_face/unknown'+str(count)+'.jpg'

yag = yagmail.SMTP("alertemailac@gmail.com", "security@123")

yag.send(

to=["jagrutipatil8299@gmail.com"],

subject="Magic Mirror",

contents="Alert Alert!!! Motion DetectedNear your Mirror",

attachments=filename

)

print("Mail Sent", "Mail Send Successfully !!")

sleep(3)

except:

cv2.putText(image, "Face Not Found", (250, 450), cv2.FONT_HERSHEY_COMPLEX, 1,
(255, 0, 0), 2)

cv2.imshow('Face Cropper', image)

pass

if cv2.waitKey(1)==13:

break

cap.release()

cv2.destroyAllWindows()
```

CHAPTER 6
TESTING

Testing is an investigation conducted to provide stakeholders with information about the quality of the product or service under test. Software Testing also provides an objective, independent view of the software to allow the business to appreciate and understand the risks at implementation of the software. Test techniques include, but are not limited to, the process of executing a program or application with the intent of finding software bugs.

Software Testing depending on the testing method employed can be implemented at any time in the development process. However, most of the test effort occurs after the requirements have been defined and the coding process has been completed. As such, the methodology of the test is governed by the Software Development methodology adopted.

6.1 Manual Testing

Testing is the fourth phase in Software Development Life Cycle (SDLC) of the Smart Mirror system. The purpose of testing is to execute a program with the intent of finding an error. To get the software error, the testing is carried out at different levels through entire software development life cycle.

After code implementation and modules integration, it is tested against the requirements to ensure the final product of the smart mirror system is solving needs addressed and gathered during the requirement gathering phase. Theoretically, the testing objective of the smart mirror are grouped into two functional types viz. black box testing and white box testing. Black box testing focuses on the functional and program interface without knowing the internal environment while white box testing focuses on the internal source, implementation and source code of application.

There are four types of testing is carried out at different levels at the development life cycle which namely are unit testing, integration testing, system testing and acceptance testing.

First is automated testing strategy where the tests are run more frequently, issues are caught at earliest stage of the development. Only selected functional code will be tested with automated testing. Second is manual testing strategy where the test is a structured set of steps to test an area of a product, have an expected result and result as well. In software testing, manual testing is the process of manually reviewing and testing software/application for errors, defects and/or vulnerabilities. This type of testing is performed by software developers and testers, without any

automated tools, to identify any defects within the software from the perspective/experience of an end user. Exploratory testing strategy is deployed as third test where the tests fit well with the Agile concept of user stories. A user story is a plain English description of how software should work. This application is implemented on user stories. These tester can easily turn a user story into a test session. Anyone can be tester when this strategy used.

6.1.1 Advantages of Manual Testing

- Manual testing is eye ball testing.
- Applications with short life cycles.
- It requires less time and expenses to begin productive manual testing.
- Automation cannot replace human intuition, inference, and inductive reasoning.
- It is covered in limited cost.
- Manual QA testing can be used in both small and big projects.
- Easily we can update our test case according to project movement.

6.1.2 Disadvantages of Manual Testing

- GUI objects size difference and colour combination etc. is not easy to find out in manual testing.
- Load testing and performance testing is not possible in manual testing.
- Running test manually is very time consuming job.
- Regression Test cases are time consuming if it is manual testing.

6.2 Test Plan

A **test plan** documents the strategy that will be used to verify and ensure that a product or system meets its design specifications and other requirements. A test plan is usually prepared by or with significant input from Test Engineers. [8, 16]

Test plan document formats can be as varied as the products and organizations to which they apply. There are three major elements that should be described in the test plan: **Test Coverage, Test Methods, and Test Responsibilities.** These are also used in a formal test strategy.

Test coverage in the test plan states what requirements will be verified during what stages of the product life.

Test methods in the test plan state how test coverage will be implemented. Test methods also specify test equipment to be used in the performance of the tests and establish pass/fail criteria.

Test responsibilities include what organizations will perform the test methods and at each stage of the product life. Test responsibilities also includes, what data will be collected, and how that data will be stored and reported (often referred to as "deliverables").

Table 6.1 Test Plan

Name of Tester	Name of Test item / Module / Function	Date of Testing
Patil Jagruti Vishram	Smart Mirror GUI on boot	18/03/2020
	Initial application data fetch	19/03/2020
	Node Package Manager (NPM) connection	19/03/2020
	Application data refresh	20/03/2020
	Module and system functionality	21/03/2020
	Loss of Internet Connectivity	21/03/2020
Patil Divya Pramod	Camera	24/03/2020
	User Motion Detection	26/03/2020 27/03/2020
	Facial recognition	01/04/2020 02/04/2020 03/04/2020
	Boot up time	04//04/2020
	Display Brightness	05/04/2020

6.3 Test Case

A test case in software engineering is a set of conditions or variables under which a tester will determine whether an application or software system is working correctly or not. It may take many test cases to determine that a software program or system is functioning correctly. Test cases are often referred to as test scripts, particularly when written. Written test cases are usually collected into test suites.

A test case is a detailed procedure that fully tests a feature or an aspect of a feature. Whereas the test plan describes what to test, a test case describes how to perform a particular test. You need

to develop a test case for each test listed in the test plan. A test case includes:

- The purpose of the test.
- Special hardware requirements, such as a modem.
- Special software requirements, such as a tool.
- Specific setup or configuration requirements.
- A description of how to perform the test.
- The expected results or success criteria for the test.

Table 6.2 Test Cases

MODULE	USAGE	INPUT	OUTPUT	REMARK
Smart Mirror GUI on boot	Power up the mirror by plugging in the power cable.	Power supply	The Smart Mirror app load upon boot.	✓
Initial data application fetch	Boot up the mirror and ensure that the Smart mirror is booted.	Boot up the mirror.	The various software features should fetch their respective data such as weather conditions and news headlines.	✓
Node package manager(NPM)database connection	When the smart mirror program is executed , it tries to connect to NPM database using the data source and catalogue. Successful connection results in the execution of smart mirror application.	The command npm start is used to initialize the smart mirror application.	The connection was established and the smart mirror application executed and run successfully.	✓
Application data refresh	Observe whether the various software features update their contents according to their set refresh cycles. Reduce the refresh intervals in the code to expedite the testing.	Application data refresh cycles.	Each software feature should update within 5 seconds after their specified refresh intervals have passed.	✓
Module and system functionality	The functionality of individual modules as well as combined functionality of all modules are simulated and tested.	The data is accessed through various websites using the browser.	Various modules like weather, calendar, newsfeed and various notifications were displayed on mirror.	✓ .

Loss of internet connectivity test	The internet connectivity is removed to ensure that the smart mirror application should stop working.	Remove WIFI connectivity.	In the event of internet connectivity disruption, the software features that rely on the internet connection should respond as programmed without comprising total system functionality.	✓
Camera	The camera connected to the smart mirror hardware to validate that the camera produced images of defined resolution and sent the frames over USB to the embedded computer without any frame loss.	Camera started and user stand in front of camera.	It was verified that the embedded computer could process the images to perform facial recognition.	✓ .
User motion detection	The user is detected using motion sensor.	Put the mirror into an inactive state with the user interface hidden and then trigger the motion sensor by walking in front of its line of sight.	The mirror should become active with its user interface elements visible within 5 seconds of stepping into its line of sight.	✓
Facial recognition	The smart mirror to work as security system faces are trained and detected, recognized so that only user with authority can access the mirror. The third person if captured by mirror an alert to the owner is sent through mail.	Put an unknown user in front of the mirror.	An alert mail is sent to owner user.	✓
Boot up time	Boot up the mirror by plugging in the power cable.	Supply power.	The smart mirror application should boot within 30 seconds of providing power	✓

Display brightness	Mirror tested in bright environment to ensure the user interface elements still be visible in unfavorable light conditions.	Place the mirror in a bright environment and power it up.	The user interface elements should still be visible despite the unfavorable lighting conditions. The rest of the screen real estate should retain its mirror-like finish as provided by the display's reflective tint.	✓

6.4 Test Results

Table 6.3 Test Cases

MODULE NAME	ERRORS	BUGS	REMARKS
Smart Mirror GUI on boot	No	No	✓
Initial data application fetch	No	No	✓
Node package manager(NPM)database connection	No	No	✓
Application data refresh	No	No	✓
Module and system functionality	No	No	✓
Loss of internet connectivity test	No	No	✓
Camera	No	No	✓
User motion detection	No	No	✓
Facial recognition	No	No	✓
Boot up time	No	No	✓
Display brightness	No	No	✓

P.S.G.V.P. MANDAL'S

D.N. PATEL COLLEGE OF ENGINEERING

SHAHADA, DIST- NANDURBAR (M.S.)

TESTING REPORT

This is to certify that

We have tested the performance of prototype

"Smart Mirror using Raspberry PI"

Developed By

Ms. Patil Vaibhavi Chudaman	***Exam Seat No.11111***
Ms. Patil Priyanka Hemant	***Exam Seat No.22222***
Ms. Patil Prerna Manilal	***Exam Seat No.33333***
Ms. Patil Divya Pramod	***Exam Seat No.33333***
Ms. Patil Jagruti Vishram	***Exam Seat No.33333***

has been successfully tested and is operating as per the specifications.

Date :___/___/2020
Place: Shahada

GUIDE
Prof. V.T.Patil

H.O.D.
Prof. V.S.Mahajan

PROJECT IN-CHARGE
Prof. R.A.Shaikh
Prof. D.B.Shukla
Prof. A.I.Pathan

CHAPTER 7

PROJECT COST AND EFFORT

7.1 Estimation Technique

For the initial estimation of our project we have used the first stage of COCOMO i.e. Basic COCOMO, now since our work is completed we have all the necessary and actual information required for the cost calculation, hence here we will use **Detailed COCOCMO.** [8, 16]

Detailed COCOMO incorporates all characteristics of the intermediate version with an assessment of the cost driver's impact on each step (analysis, design, etc.) of the software engineering process.

The detailed model uses different effort multipliers for each cost driver attribute. These Phase Sensitive effort multipliers are used to determine the amount of effort required to complete each phase. In detailed COCOMO, the whole software is divided in different modules and then we apply COCOMO in different modules to estimate effort and then sum the effort.

In detailed COCOMO, the effort is calculated as function of program size and a set of cost drivers given according to each phase of software life cycle. A Detailed project schedule is never static.

The Six phases of detailed COCOMO are:-

- Plan And Requirement.
- System Design.
- Detailed Design.
- Module Code and Test.
- Integration and Test.
- Cost Constructive Model

Detailed COCOMO incorporates the set of "cost drivers" that include subjective assessment of product, hardware, personnel and project attributes. The 17 cost drivers which are multiplicative factors that determine the effort required to complete our software project. Each of the 17 attributes

receives a rating on a six-point scale that ranges from "very low" to "extra high" (in importance or value).

7.2 DETAILED COCOMO-COST DRIVERS

Table 7.1 Cost Drivers for Detailed COCOMO

Cost Driver	**Ratings**					
Personnel Factors	**Very Low**	**Low**	**Nomi-nal**	**High**	**Very High**	**Extra High**
Analyst Capability **(ACAP)**	1.46	**1.19**	1.00	0.86	0.71	---
Applications Experience **(APEX)**	1.29	**1.13**	1.00	0.91	0.82	---
Programmer Capability **(PCAP)**	1.42	1.17	**1.00**	0.86	0.70	---
Platform Experience **(PLEX)**	1.21	1.10	1.00	**0.90**	---	---
Language and Tool Experience **(LTEX)**	1.14	1.07	1.00	**0.95**	---	---
Personnel Continuity **(PCON)**	1.29	**1.12**	1.00	0.90	0.81	---
Project Factors						
Use of Software Tools **(TOOL)**	1.24	1.10	1.00	**0.91**	0.83	---
Multisite Development **(SITE)**	**1.24**	1.10	1.00	0.91	0.82	---
Development Schedule **(SCED)**	1.23	1.08	1.00	**1.04**	1.10	---
Platform Factors						
Execution Time Constraint **(TIME)**	---	---	**1.00**	1.11	1.30	1.66
Main Storage Constraint **(STOR)**	---	---	**1.00**	1.06	1.21	1.56
Platform Volatility **(PVOL)**	---	**0.87**	1.00	1.15	1.30	---
Product Factors						
Required Software Reliability **(RELY)**	0.75	0.88	1.00	1.15	**1.40**	---

Database Size **(DATA)**	---	0.94	**1.00**	1.08	1.16	---
Product Complexity **(CPLX)**	0.70	0.85	1.00	**1.15**	1.30	1.65
Required Reusability **(RUSE)**	---	**0.95**	1.00	1.07	1.15	1.24
Documentation Match to Lifecycle Needs **(DOCU)**	0.81	0.91	**1.00**	1.11	1.23	---

7.3 COST PER PERSON-MONTH FOR PHASES OF SDLC

Table 7.2 Assumed Cost for each Phase of SDLC

Phase	Cost
Requirement Analysis	500
Product Design	500
Detailed Design	1000
Coding & Unit Test	1500
Integration & Test	500

7.4 DETAILED ESTIMATION REPORT(Obtained using SystemStar)

Smart Mirror - Detail Report

SystemStar 3.0 Demo May 18, 2020 11:25:22 Page: 1

Estimate Name: Smart Mirror
Model Name: COCOMO® II 2000
Process Model: COCOMO® II Model
Estimate ID:
Model ID: 2000
Phases: Waterfall

Component Name: Component1
Increment: 1
Developed Size: 5,000
Component ID:
Level: 1
EAF: 1.0000

Phase	Effort (Person-Months)	Cost (K$)	Duration (Months)	Staffing
RQ -- Requirements	1.1	10.9	1.4	0.8
PD -- Product Design	2.7	13.3	2.1	1.3
DD -- Detailed Design	4.1	6.2	2.0	2.1
CT -- Code & Unit Test	5.6	5.6	2.6	2.1
IT -- Integration & Test	3.2	6.4	1.8	1.8
Development (PD+DD+CT+IT)	15.6	31.5	8.5	
Totals (RQ+PD+DD+CT+IT)	16.7	42.5	9.9	
MN -- Maintenance (per year)	0.0	0.0		0.0

Figure 7.1 Detailed COCOMO Estimation Report

7.5 ESTIMATION SUMMARY

Table 7.3 Summary of calculated estimations.

Estimation	Value
Size of the Project	**11000** Lines of Code
Effort Required	**16.7** Person-Month
Duration Required	**9.9** months
Person Required	**5** persons
Cost Required	**42,500**

CHAPTER 8
RESULT

While looking and grooming in front of mirror the user log in to its account and the mirror provides specified real time data like current weather, temperature, humidity, date, time, phone notifications and latest newsfeed and when there is no known user in front of mirror it acts like security system that is if unknown person stands in front of mirror it will be notified to the owner through an alert mail.

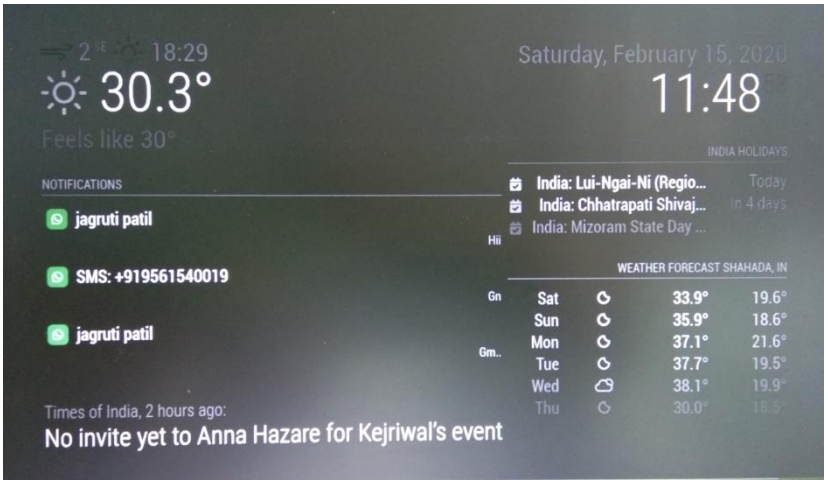


Figure 8.1 Output on the Screen

CHAPTER 9
CONCLUSION

The smart mirror designed will provide the user with an enhanced mirror experience. The device has been designed keeping in mind the advancement in the field of home automation. The mirror is powered and controlled by the Raspberry Pi 3 to give output in the form of real time data feeds like news, weather, calendar, etc are displayed on LCD screen. The PIR sensor ensures that the mirror will always turn on when a person steps up to use it. Also the mirror works as a Security System, when there is nobody in home it can be switched into security system by using PIR to detect human presence. When someone enters the room, it will capture an image and send an alert message to the owner using the face recognition module.

CHAPTER 10

FUTURESCOPE

There are many future possibilities for this project and hopefully will be continued. A layout can be extended to accommodate more functionalities by controlling the mirror using voice commands. In future we will try to add advanced gesture controls, automated salutation using face recognition of the end user and also understand that how advanced artificial intelligence can be implemented to the mirror so that it can automatically take care of all the requirements of the end user.

REFERENCES

[1] P.L. Emiliani and C. Stephanidis , Universal access to ambient intelligence envi-ronments:

Opportunities and challenges for people with disabilities. IBM System-sJournal, 2005.

[2] M. S. Raisinghani, A. Benoit, J. Ding. M. Gomez, K. Gupta, V. Gusila. D. Power, and 0. Schmedding. Ambient intelligence: Changing forms of human computer interaction and their social implications. Journal of Digital Information, 5(4), 2004.

[3] M. Z. Poh, D. McDuff, R. Picard, "A medical mirror for non-contact health monitring," In ACM SIGGRAPH 2011 Emerging Technologies SIGGRAPH "11, NewYork, USA, ACM

[4] "What is a Raspberry Pi?" Raspberry Pi What Is a Raspberry Pi Comments. Accessed May 06, 2016. https://www.raspberrypi.org/help/what-is-a-raspberry-pi/. P.L. Emiliani and C. Stephanidis , Universal access to ambient intelligence envi-ronments:

Opportunities and challenges for people with disabilities. IBM System-sJournal, 2005.

[5] M. S. Raisinghani, A. Benoit, J. Ding. M. Gomez, K. Gupta, V. Gusila. D. Power, and 0. Schmedding. Ambient intelligence: Changing forms of human computer interaction and their social implications. Journal of Digital Information, 5(4), 2004.

[6] F. Bomarius, M. Becker, and T. Kleinberger. Embedded intelligence for ambient-assisted living. ERCIM News, 67:19-20, 2006.

[7] P.L. Emiliani and C. Stephanidis. Universal access to ambient intelligence environments: Opportunities and challenges for people with disabilities. IBM SystemsJournal, 44(3):605-619, 2005.

[8] Roger S. Pressman, "*Software Engineering: A Practitioner's Approach*", Fifth Ed., MGH, ISBN 0-07-365578-3

[9] M. Friedewald, 0. Da Costa, Y. Punie, P. Alahuhta, andS. Heinonen. Perspectives of ambient intelligence in the home environment. Telematics and Informatics, 22(3):221238, 2005.

[10] Derrick Gold, David Sollinger, and Indratmo. SmartReflect: A Modular Smart Mirror Application Platform. IEEE Journal, Nov 2016.

[11] Tatiana Lashina. Intelligent bathroom. In European Symposium on Ambient Intelligence (EUSAI'04), Eindhoven, Netherlands, 2004.

[12] Si Liu, Luoqi Liu, Shuicheng Yan, Department of Electrical and Computer Engineering National University of Singapore. 2013 Second IAPR Asian Conference on Pattern Recognition.

[13] "Python Features" *https://www.javapoint.com/python-features*

[14] "Features of Node.js" *https://www.tutorialspoint.com/nodejs/nodejs_introduction.htm*

[15] "Manual Testing" *https://www.javapoint.com/manual-testing*

[16] Mall, Rajib, "*Fundamentals of software Engineering*", Fourth Edition, ISBN: 978-81-203-4898-1

[17] Grady Booch, James Rumbauch, Ivar Jacobson, "*The Unified Modeling Language User Guide*", Publisher: Addison Wesley, First Edition October 20, 1998, ISBN: 0-201-57168-4, 512 pages

APPENDIX

A. GLOSSORY

Pace

Pace is the rate of activity or movement, such as in running or in the flow of events in an entertainment piece

Dabian

Debian is a Unix-like operating system consisting of free software. Debian is a popular and freely-available computer operating system that uses the Linux kernel and other program components obtained from the GNU project.

TFT Touchscreen

Thin Film Transistor (TFT) is a display screen technology used in liquid crystal display (LCD). A TFT touch screen is a combination device that includes a TFT LCD display and a touch technology overlay on the screen.

HDMI Cable

HDMI is a proprietary audio/video interface for transmitting uncompressed video data and compressed or uncompressed digital audio data from an HDMI-compliant source device, such as a monitor, video projector, digital television, or digital audio device.

VNC Viewer

Virtual Network Computing (VNC) is a type of remote-control software that makes it possible to control another computer over a network connection.

PIR Sensor

A passive infrared sensor (PIR sensor) is an electronic sensor that measures infrared (IR) light radiating from objects in its field of view. They are most often used in PIR-based motion detectors.

Dropbox

Dropbox is a cloud storage service for sharing and storing files including photos, documents and videos, which means you can copy your files to the cloud and access them later, even if you're using a different device

DHT22 Sensor

The DHT22 is a basic, low-cost digital temperature and humadity sensor. It uses a thermistor to

measure the surrounding air

PI Camera

The Raspberry Pi Camera v2 is the new official camera board released by Raspberry Pi Foundation. The Raspberry Pi Camera v2 is a high quality 8 megapixel Sony IMX219 image sensor custom designed add-on board for Raspberry Pi, featuring a fixed focus lens.

Audio Jack of RaspPI

The Pi Model B+, Pi 2, Pi 3 and Pi 4 features a 4-pole 3.5mm audio jack which also includes the composite video signal. This has allowed for the removal of the composite video socket found on the original Model B. The new jack is a 4-pole socket which carries both audio and video signals.

OpenCV

OpenCV (Open Source Computer Vision) is a library of programming functions mainly aimed at real- time computer vision. In simple language it is library used for Image Processing. It is mainly used to do all the operation related to Images.

NPM

NPM (originally short for Node Package Manager) is a package manager for the JavaScript programming language. It is the default package manager for the JavaScript runtime environment Node.js. It consists of a command line client, also called npm, and an online database of public and paid-for private packages, called the npm registry.

PIP

pip is a de facto standard package-management system used to install and manage software packages written in Python. Many packages can be found in the default source for packages and their dependencies. Most distributions of Python come with pip preinstalled.

RSS

It is commonly referred to as "Really Simple Syndication. "RSS is method of providing website content such as news stories or software updates in a standard XML format. Websites such as The Wall Street Journal and CNET's News.com provide news stories to various RSS directories that distribute them over the Internet.

Google Calendar

Google Calendar is the time management and scheduling tool created by Google. It is a free web and mobile calendar that lets you keep track of your own events and share your calendars with others.

CNNS world NEWS RSS Feed

CNN.com offers feeds of story headlines in XML format to visitors who use RSS aggregators. The Domo CNN connector takes those headlines and creates a new DataSet that can be used to provide current headlines into your Domo instance. The CNN RSS connector is a "Cloud App" connector,

meaning it retrieves data stored in the cloud.

Sandbox

A sandbox is a testing environment that isolates untested code changes and outright experimentation from the production environment or repository, in the context of software development including Web development and revision control.

Broadcom Bcm2835 SoC

This is the Broadcom chip used in the Raspberry Pi Model A, B, B+, the Compute Module, and the Raspberry Pi Zero. The Broadcom BCM2835 SoC used in the first generation Raspberry Pi includes a 700 MHz ARM1176JZF-S processor, VideoCore IV graphics processing unit (GPU), and RAM. It has a level 1 (L1) cache of 16 KiB and a level 2 (L2) cache of 128 KiB. The level 2 cache is used primarily by the GPU.

ARM 1176JZF-S

ARM11 is a group of older 32-bit RISC ARM processor cores licensed by ARM Holdings. The ARM11 core family consists of ARM1136J(F)-S, ARM1156T2(F)-S, ARM1176JZ(F)-S, and ARM11MPCore.

CSI Port

The Camera Serial Interface (CSI) is a specification of the Mobile Industry Processor Interface (MIPI) Alliance. It defines an interface between a camera and a host processor. The latest active interface specifications are CSI-2 v3.0, CSI-3 v1.1 and CCS v1.0 which were released in 2019, 2014 and 2017 respectively.

WI-FI Protocol

WI-FI is a family of wireless networking technologies, based on the IEEE 802.11 family of standards, which are commonly used for local area networking of devices and internet access. WI-FI uses multiple parts of the IEEE 802 protocol family, and is designed to network seamlessly with its wired sibling Ethernet.

GPU : OpenGL ES 2.0

OpenGL for Embedded Systems (OpenGL ES or GLES) is a subset of the OpenGL computer graphics rendering application programming interface (API) for rendering 2D and 3D computer graphics such as those used by video games, typically hardware-accelerated using a graphics processing unit (GPU). It is designed for embedded systems like smart phones, tablet computers, video game consoles and PDAs. OpenGL ES is the "most widely deployed 3D graphics API.

24 GFLOPS

Gigaflops is a unit of measurement used to measure the performance of a computer's floating point unit, commonly referred to as the FPU. One gigaflops is one billion (1,000,000,000) FLOPS, or floating point operations, per second. The term FLOP is often used for floating-point operation, for example as a unit of counting floating-point operations carried out by an algorithm or computer hardware.

MIPI

The mobile industry processor interface (MIPI) standard defines industry specifications for the design of mobile devices such as smart phones, tablets, laptops and hybrid devices. MIPI interfaces play a strategic role in connected car and Internet of Things (IoT) solutions.

HDT

HDT-lib is a Java Library that implements the W3C Submission of the RDF HDT (Header-Dictionary- Triples) binary format for publishing and exchanging RDF data at large scale. Its compact representation allows storing RDF in fewer space, providing at the same time direct access to the stored information. This is achieved by depicting the RDF graph in terms of three main components: Header, Dictionary and Triples. The Header includes extensible metadata required to describe the RDF data set and details of its internals.

BCM 43143

Raspberry Pi 3 Model B has a BCM43143 on board. That one should work out of the box. A Wi-Fi adapter will probably need more power than the Raspberry Pi USB port can provide, especially if there is a large distance from the Wi-Fi adapter to the Wi-Fi Access Point, or it is transferring large amounts of data. Therefore, you may need to plug the Wi-Fi adapter into a powered USB hub.

Transformer

A device used to transfer electrical energy from one circuit to another. Transformers used to change the voltage of an alternating current in one circuit to a different voltage in a second circuit.

Rectifier

A rectifier used for powering appliances. Using a rectifier in the power supply helps in converting AC to DC power supply. Bridge rectifiers are widely used for large appliances, where they are capable of converting high AC voltage to low DC voltage.

Capacitor

A device consisting of one or more pairs of conductors separated by an insulator and it is used to store the electrical energy and give this energy again to the circuit when necessary. In other words, it charges and discharges the electric charges stored in it.

Regulator

A voltage regulator is a system designed to automatically maintain a constant voltage level. A device for maintaining a designated characteristic, as voltage or current, at a predetermined value, or for varying it according to a predetermined plan.

Controller

A controller, in a computing context, is a hardware device or a software program that manages or directs that flow of data between two entities. In a general sense, a controller is just something or someone that interfaces between two systems and manages communications between them.

B. USER MANUAL

I. Required Software

1. Python 3.0
2. Node.js 10
3. VNC viewer

II. Environment Setup (Software Installation and their setting)

No Special setup required, ***simply install the above mentioned software normally.***

III. Database Setup (if any)

Our project does not use database, but requires Dataset from browser.

IV. Project Execution Steps

1. Turn on the supply of both raspberry and LCD screen.
2. Turn any hotspot and connect it with Raspberry PI.
3. Now monitor the raspberry pi connect your device raspberry through VNC viewer it require an IP address so we have to enter the correct IP address of raspberry pi so it will be get connected.
4. The mirror starts automatically.

C. BASE PAPER

International Journal of Pure and Applied Mathematics
Volume 118 No. 22 2018, 451-455
ISSN: 1314-3395 (on-line version)
url: http://acadpubl.eu/hub
Special Issue

ijpam.eu

IMPLEMENTATION OF MAGIC MIRROR USING RASPBERRY PI 3

[1]Suryansh Chandel, [2]Ashay Mandwarya, [3]S.Ushasukhanya
[1,2,3]Department of Computer Science and Engineering
SRM Institute of Science and Technology, Kattankulathur, Tamil Nadu, India
[1]suryansh.chandel004@gmail.com, [2]ashaymurceilago@gmail.com,
[3]ushasukhanya.s@ktr.srmuniv.ac.in

Abstract: This paper describes the designing and implementation of an voice controlled wall mirror, called "Magic Mirror". It is a device that can function both as a mirror and an interactive display displaying multimedia content such as time, date, weather and news simultaneously. The user can interact with it using voice commands. The Magic Mirror consists of various functionalities like real time data and information updates, voice commands, face detection/recognition using LCD monitor, microphone and webcam. The user can interact with magic mirror using voice commands.

Keywords: Magic Mirror, Raspberry pi 3, Rasbian, face detection, Internet of Things;

1. Introduction

Day after day we are moving towards a more automated and interconnected world because of various wirelessly connected embedded devices. These are responsible for changing and improving the standards and quality of living. Many devices are being developed which use concepts of multimedia communication, artificial intelligence, internet of things (IoT) to revolutionizing the way we perform our various day to day tasks in our home, offices or even industries. Most of us use mirrors every day to look at ourselves; we psychologically interact with the mirror every day to check how we look and how our attire is while getting ready for our work or colleges. So, The idea of having an interactive mirror that can respond to your commands can excite anyone. Magic Mirror aims at augmenting the basic reflective mirror with embedded intelligence to combine daily routine tasks like reading newspaper, getting stock updates, weather updates etc. and providing all that data to the user while he/she gets ready. The Magic mirror will help in automating our work and development of smart houses. This paper provides a detailed idea of theory of design and practical implementation of Magic Mirror.

2. Theory

The innovation and research work in the field of Artificial intelligence, Machine learning, Internet of things has brought a massive change in the technology we use and paved the way for Smart environment. Kevin Ashton published an article in the RFID Journal in 2009[1] in which he talked about the capabilities of things that a computer can perform if it knew everything there was to know about things by the means of gathering data and track everything. They would be able to reduce loss, waste and cost. We would be able to get updates about machines know when they needed replacing or repairing. There is need to empower the computer by automating them to see them in their full glory This is exactly what has happened after the development in field of IoT. Also, An efficient, convenient and secure home automation environment [2] can be achieved using collective application of AI and IoT. Artificial intelligence has already received attention for assisted living purposes [3]. Apart from this entertainment, automated home environment, official space and home learning [5] are also affected due to advancement in AI and IoT. Various companies are now launching products aimed at automating the day to day activities we do in our home, offices or industries. Nest Labs launched learning thermostat [6] in 2010 which used to detect smoke and carbon monoxide detector and later was redesigned to sense an control temperature in the house. Magic Mirror also provides solution to our daily routine of getting ready. It uses the concept of Internet of things to embed various chores like reading newspaper, getting stock updates, traffic updates etc. on a display that will also work as a normal reflective mirror simultaneously.

3. Proposed System And Components

A. System Overview

Proposed system and block diagram for magic mirror are shown in figure 1. The aim of designing this model is to create an interactive interface which can be

conveniently used in home environment as well as commercial space. Various services like weather, calendar, traffic, news stock updates etc. can be accessed and controlled using voice commands. The Raspberry Pi 3 is connected to a Monitor via HDMI cable and a webcam is attached using a universal serial bus. Raspberry Pi is powered up using a 5V/2A DC supply.

We plan to deliver a working model of Magic mirror by using raspberry pi 3 for smart homes of future
as well as commercial uses. The device will look like a normal reflective mirror but would have a monitor attached on one side. A special two way mirror is used for this purpose as it can act as normal reflective mirror when the monitor is off and can also display various data as soon as the monitor is turned on. This will thus serve both the purposes.

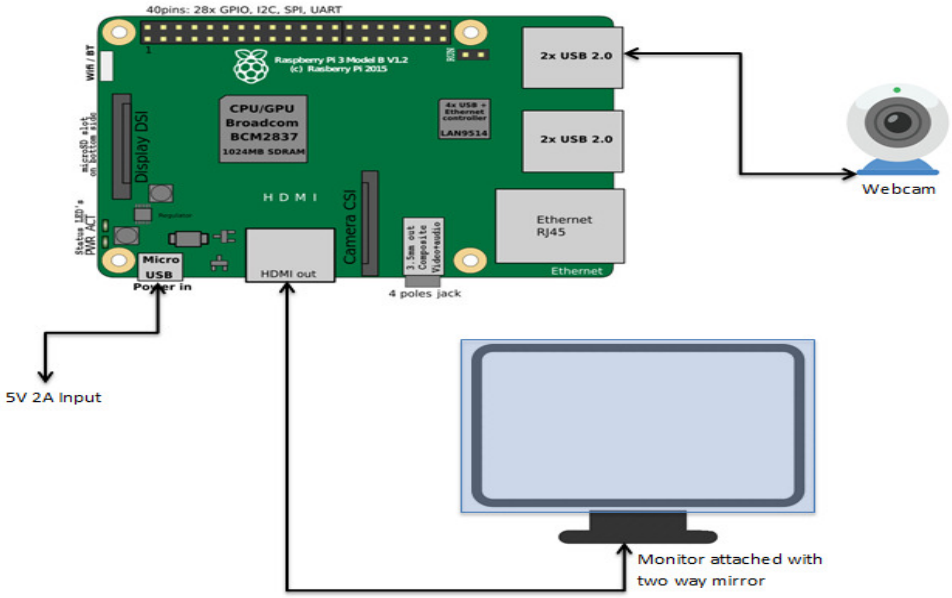


Fig 1 Block Diagram

B. Raspberry Pi 3

Raspberry Pi 3 acts as the main control center for this proposed model. The Raspberry Pi is equipped with a micro SD card which can be loaded with operating systems like Raspbian or Windows 10 IoT core. After the OS is running the Magic Mirror code will can be implemented on it to run the application. The Monitor will be getting input from RPi using HDMI cable and voice commands can be given to RPi using a microphone.

C. Dual Purpose Display

For the purpose of dual functionality, we are using a two way mirror for the display. It will be attached on top of the monitor using a wooden frame to hold the whole system together. The two way mirror can act as normal reflective mirror when the monitor is switched off and the data can be simultaneously displayed while the monitor is switched on.

4. Functionality

Figure 2 provides an ER Diagram for the proposed magic mirror. Proposed model can perform various functions described as follows:
i) Work as a normal reflective mirror so that the user can use it as a regular mirror.
ii) A two way mirror which can function both as reflective and see through mirror is attached to a LED monitor. This provides two major functionalities ie. Mimicking a normal mirror as well as working as a display for real time data updates.
iii) Personalized data and information services:
Anyone using this mirror will be able to get real time updates of traffic, stocks, news and headlines, date, time, weather updates as well as other reports of our particular interests.
iv) Voice Commands:
User will be able to give voice commands to the mirror using a microphone connected to the Raspberry pi 3.

The Magic mirror will display data in accordance to the user commands.

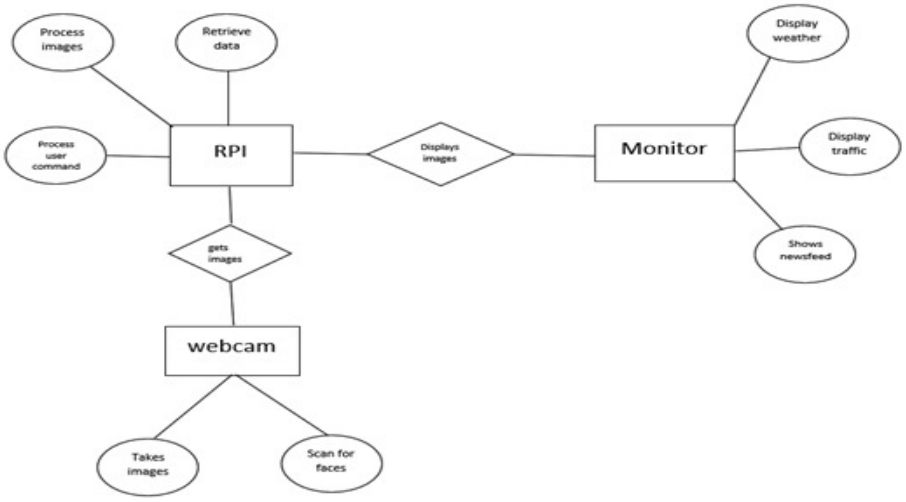


Fig 2 ER Diagram

The webcam is attached to the Raspberry pi using the universal serial bus to detect user's face using OpenCV. This will help in setting up the personalized profiles for different users and managing them afterwards. Figure 3 given below shows the basic user interface of the magic mirror that will be used by the end user. The user interface will be show the data on the mirror and the empty space in between will accommodate the reflection of the user.

Fig 3 Magic Mirror UI

5. Related Work

The proposed Magic Mirror represents a natural interface that provides a platform to access information and data services in a more personalized manner. This project is aimed at contributing to the design and implementation of a Magic Mirror-like interface as well as the automated home environment where user can interact with the mirror interface, we briefly comment on some related work and research in similar direction. SmartReflect[7] is a similar work carried out by the students of MacEwan University. It basically aimed at providing a platform that can facilitate the development of smart mirror. It acts an alternative option than the sandbox environment. It is light in functioning as compared to already present platforms. Its major advantage is its multiple language and environment support so as to ease end user efforts.

Another project named MagicMirror[10] as carried out by students of NUS, They created a magic mirror which can recommend you appropriate clothing in the morning while you get ready. The Magic mirror model will scan the user and then based on the particular occasion or event it will recommend most suitable attire and other styling options. The events can be retrieved from user's social media account or can be added to the calendar manually.

Philips HomeLab [8] acts as testing platform for interactive and automated home environment. The Philips Hue is one such example of smart lighting which can be controlled using mobile application. Another example is of an Interactive Mirror [9] which can installed in room or washroom to get personalized services depending on the end user. Children can customize it view cartoons, adults can get live news feeds and updates on weather, traffic etc.

In comparison to various works as mentioned above, The model proposed in our paper is different as our it's aim is to develop a functional prototype of a interactive and technologically enhanced platform

which can provide personalized services which can be operated and controlled using user's voice commands

6. Conclusion

We have designed an intelligent mirror keeping in mind the up-coming future advancement in the field of home automation environment. The prototype of the magic mirror is powered and controlled by the Raspberry Pi 3 and all the final output in form of real time data feeds are displayed on LED screen fixed with a two way mirror.

We have built a working model to demonstrate various functionalities of the mirror using voice commands. It gives a layout that can be extended in future to accommodate even more functionalities.
In our future work we will try to add advanced gesture controls, automated salutation using face recognition of the end user and also understand that how advanced artificial intelligence can be implemented to the mirror so that it can automatically take care of all the requirements of the end user.

References

[1] K.Ashton, "That 'Internet of Things' Thing" RFID Journal, July 22, 2009. (*references*)

[2] M. S. Raisinghani, A. Benoit, J. Ding. M. Gomez, K. Gupta, V. Gusila. D. Power, and 0. Schmedding. Ambient intelligence: Changing forms of human computer interaction and their social implications. Journal of Digital Information, 5(4), 2004.

[3] F. Bomarius, M. Becker, and T. Kleinberger. Embedded intelligence for ambient-assisted living. ERCIM News, 67:19-20, 2006.

[4] P.L. Emiliani and C. Stephanidis. Universal access to ambient intelligence environments: Opportunities and challenges for people with disabilities. IBM SystemsJournal, 44(3):605-619, 2005.

[5] M. Friedewald, 0. Da Costa, Y. Punie, P. Alahuhta, andS. Heinonen. Perspectives of ambient intelligence in the home environment. Telematics and Informatics, 22(3):221-238, 2005.

[6] Nest Labs Thermostat,"Programs itself, Then pays for itself", 2010 https://nest.com/thermostats/nest-learning-thermostat/overview/

[7] Derrick Gold, David Sollinger, and Indratmo. SmartReflect: A Modular Smart Mirror Application Platform. IEEE Journal, Nov 2016.

[8] Philips Homelab. http:// www.research.philips. com/ technologies/misc/homelab/index.html

[9] Tatiana Lashina. Intelligent bathroom. In European Symposium on Ambient Intelligence (EUSAI'04), Eindhoven, Netherlands, 2004.

[10] Si Liu, Luoqi Liu, Shuicheng Yan, Department of Electrical and Computer Engineering National University of Singapore. 2013 Second IAPR Asian Conference on Pattern Recognition.

[11] S.V.Manikanthan and D.Sugandhi " Interference Alignment Techniques For Mimo Multicell Based On Relay Interference Broadcast Channel " International Journal of Emerging Technology in Computer Science & Electronics (IJETCSE) ISSN: 0976-1353 Volume- 7 ,Issue 1 –MARCH 2014.

[12] T. Padmapriya and V. Saminadan, "Inter-cell Load Balancing technique for multi-class traffic in MIMO-LTE-A Networks", International Journal of Electrical, Electronics and Data Communication (IJEEDC), ISSN: 2320- 2084, vol.3, no.8, pp. 22-26, Aug 2015.

D. PUBLISHED PAPER