Design Manual

Digital Signage Based User Targeted Advertising

Version 1.0
May 2021

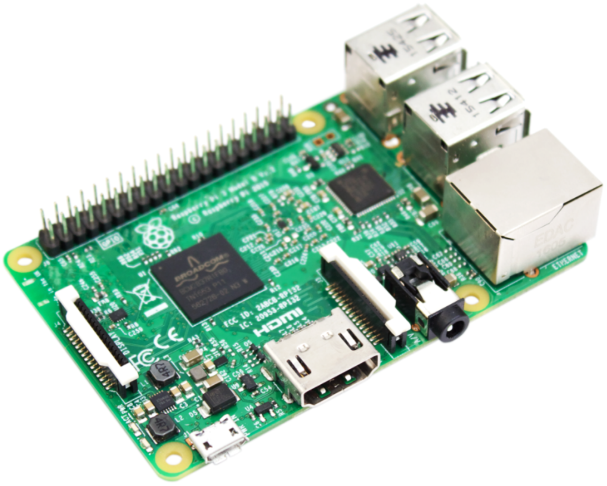
Prepared by
● Dhanushka S.M.V.
● Thisanke M.K.H.
● Lakmali B.L.S.

Table of Contents

1. Hardware Information

1.1. RaspberryPi 3B Board

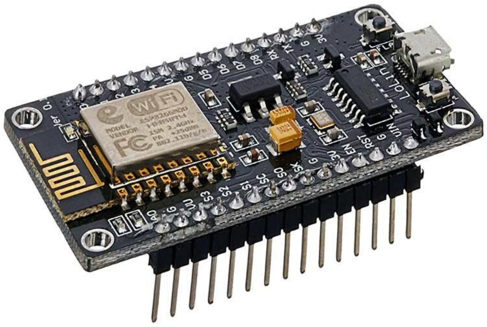
Raspberry Pi is the name of a series of single-board computers made by the Raspberry Pi Foundation. The Raspberry Pi is a very cheap computer that runs Linux, but it also provides a set of GPIO (general purpose input/output) pins, allowing you to control electronic components for physical computing and explore the Internet of Things (IoT).



- **Features and Specifications**
  CPU: Quad-core 64-bit ARM Cortex A53 clocked at 1.2 GHz
  GPU: 400MHz VideoCore IV multimedia
  Memory: 1GB LPDDR2-900 SDRAM (i.e. 900MHz)
  USB ports: 4
  Video outputs: HDMI, composite video (PAL and NTSC) via 3.5 mm jack
  Network: 10/100Mbps Ethernet and 802.11n Wireless LAN
  Peripherals: 17 GPIO plus specific functions, and HAT ID bus
  Bluetooth: 4.1
  Power source: 5 V via MicroUSB or GPIO header
  Size: 85.60mm × 56.5mm
  Weight: 45g (1.6 oz)

1.2. ESP8266 (Node MCU)

Espressif Systems (A Semiconductor Company) has released a bite-sized WiFi enabled microcontroller – ESP8266, it can monitor and control things from anywhere in the world. Commonly used for IoT projects.



- **Features and Specifications**

The development board equips the ESP-12E module containing,

- ESP8266 chip having Tensilica Xtensa® 32-bit LX106 RISC microprocessor
- Operates at 80 to 160 MHz adjustable clock frequency

128 KB RAM and 4MB of Flash memory (for program and data storage) Enough to cope with JSON/XML data and programming codes and everything.

The ESP8266 Integrates 802.11b/g/n HT40 Wi-Fi transceiver

- It can not only connect to a WiFi network and interact with the Internet, but it can also set up a network of its own, allowing other devices to connect directly to it.

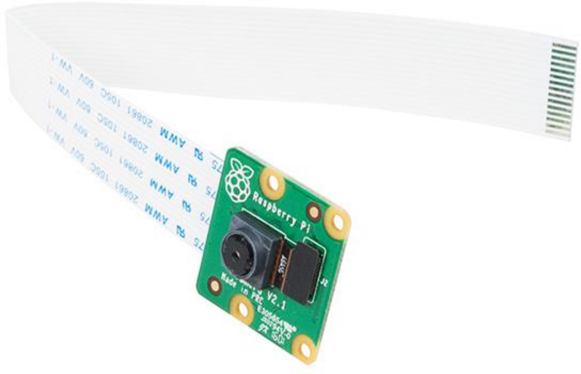
Power to the ESP8266 NodeMCU is supplied via the on-board MicroB USB connector.

- Alternatively, in our case we use a regulated 5V voltage source to the VIN pin to power up the ESP8266 and its peripherals.

These features make the ESP8266 NodeMCU even more versatile and are idle for building the smart power supply unit to receive signals via wifi connection.

1.3. RaspberryPi Camera Module

The upgraded Raspberry Pi Camera Module v2 is a camera board released by the Raspberry Pi Foundation which connects to any Raspberry Pi Module.



- **Features and Specifications**

    - Fixed focus lens on-board
    - Improved resolution - 8 megapixel native resolution sensor-capable of 3280 x 2464 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

The user detecting and analyzing unit captures persons coming in front of the camera module with better quality using the v2 camera module. The image analysis provides better results(predict age,gender) with this camera quality rather than using a webcam.

1.4. 5V Relay Module

5V DC Relay modules allow for easy switching of higher voltages and are ideal for use with microcontrollers.

The digital screen connects to the smart power supply unit through the 5v relay to trigger the screen on/off according to the signals received to the relay from the nodeMCU.

1.5. LED Push Buttons and Rocker Switch

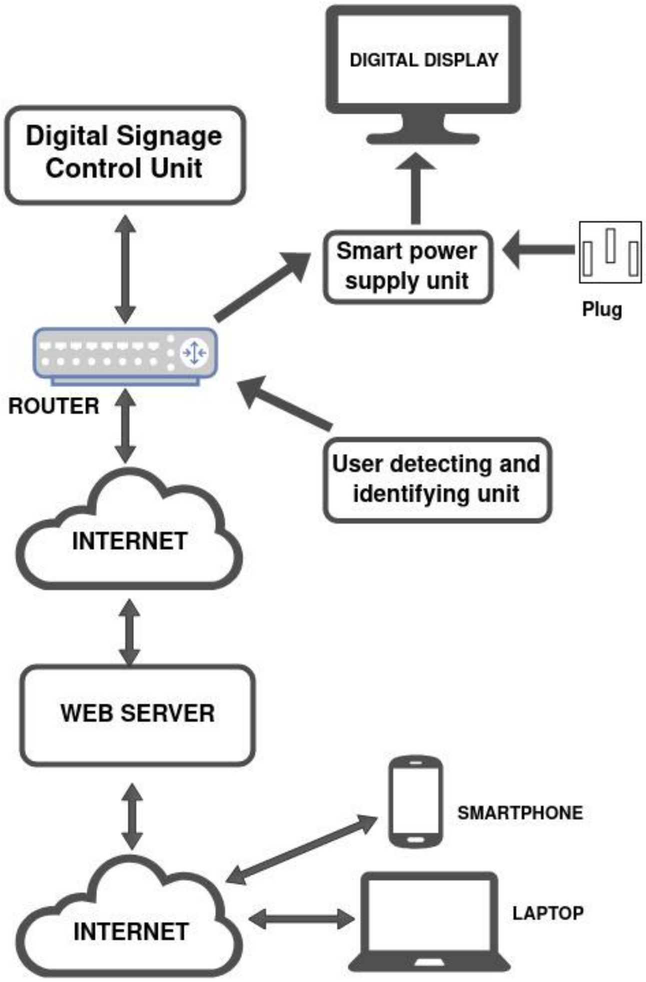
Manual switches are deployed in each of the units.

- Signage Control Unit and User Detecting and Identifying Unit used to power up the Raspberry Pi boards inside.
- Smart power supply unit uses it to create a normal power supply by switching off the nodeMCU power in case the user doesn't need it.

1.6. Cooling Fans and Heat Sinks

Embedded to the box coverings of the unit to circulate airflow and minimize damage to the components due to generated heat.

2. System Overview (Schematic Diagram)

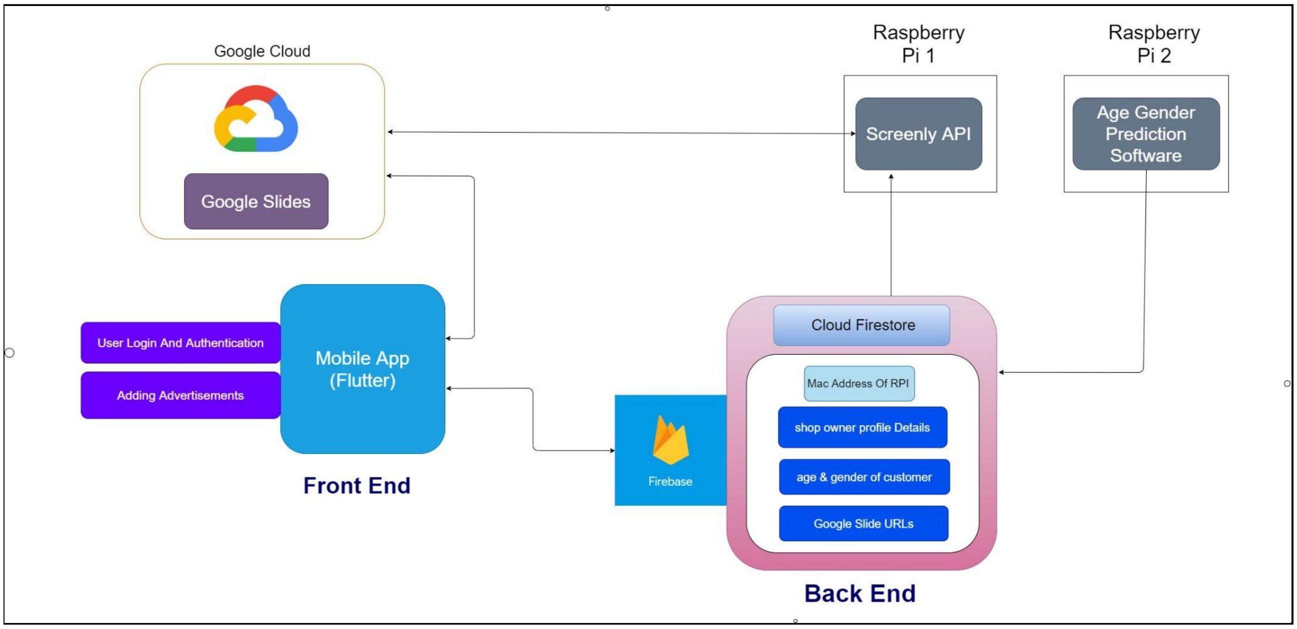


- The **Digital Signage Control unit** and the **User Detecting and Identifying unit** both work together by communicating with each other through the web server through the internet.

- The **Smart Power Supply unit** connects and powers up the digital screen. Digital Signage Control Unit connects with the digital display through an HDMI Connection and displays advertisements.

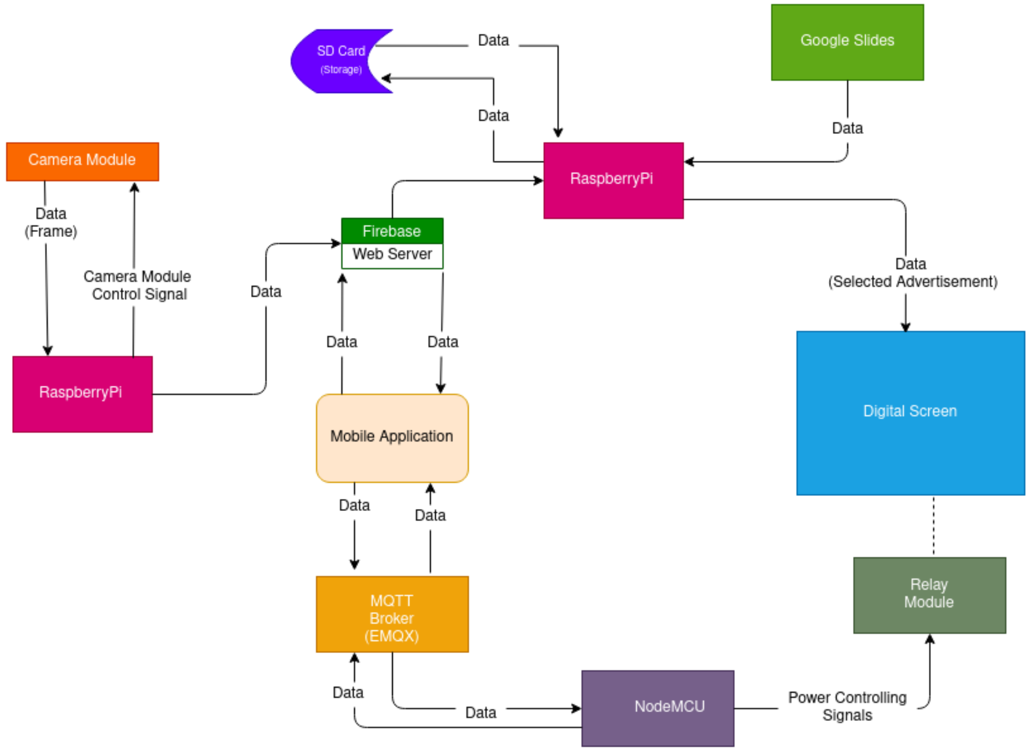
- The whole system can be controlled through the mobile application by adding the connected devices to the user's login account.

3. Software Overview



- ❑ Front End software of the system is the mobile app which is developed with Flutter.
- ❑ The system backend software is developed with Firebase.
- ❑ All login credentials and other information about the digital signage owners are stored in cloud firestore in firebase.
- ❑ The age and gender predictions about the people in front of the screen are also updated real time in Firebase.
- ❑ Digital signage owners can upload their advertisements in the form of Google slides using the mobile app.
- ❑ So, uploaded advertisements are stored in google cloud.
- ❑ Screenly OSE and a configured API is installed to the Raspberry Pi board and it fetches the identified persons' age and gender from Firebase and it's API is configured to request for the specific advertisement category.
- ❑ Then it selects the relevant advertisement category from google slides and displays it in real-time.

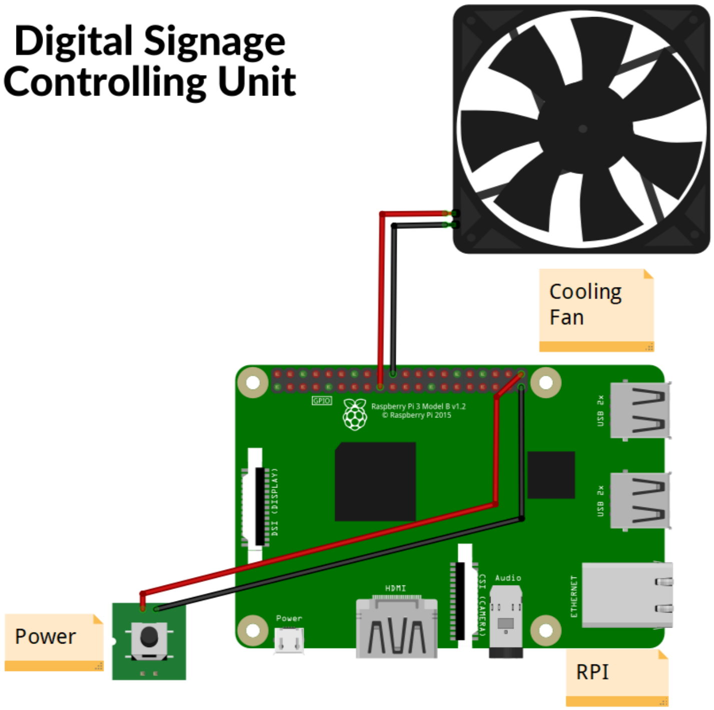
4. Data Flow Diagram



- User is supposed to upload the advertisements using the mobile app.
- Those advertisements are uploaded into the selected google slides page according to age and gender.
- Image of the person who is in front of the screen is captured by the camera module and processed in the RPi to extract the face by running haar-cascade classifiers
- That image is processed with a pre-trained deep learning model to predict the gender and age group of the person.
- The age,gender details are updated in the firestore database in realtime and the signage control unit listens and query these new data.
- According to the received results, matching google slides presentation is selected and displayed on the digital screen.

5. Implementation Details

5.1. Digital Signage Controlling Unit



Actual Implementation

Hardware Components

- ❑ Digital signage controlling unit mainly consists of a Raspberry Pi board. A SD card is inserted into the RPI board.
- ❑ Screenly OSE is installed in RPI that fetches the correct google slide according to the gender and age of the person.
- ❑ When more than one face appears in front of the screen, it is supposed to display generic advertisements.

- Preparing a wpa_supplicant.conf file for deploying devices preloaded with WiFi settings.

  Use a website that calculates WPA pre-shared keys, such as, https://www.wireshark.org/tools/wpa-psk.html

As an example, we have an SSID called “Mobitel”, and password Password123. After we input these values into that generator form, the psk-key value is: dbedc6954827ad7187466e42f4df78d9108955778959b5db11920f8c328dd716

We take this value and create a file called wpa_supplicant.conf and fill it with the following contents:

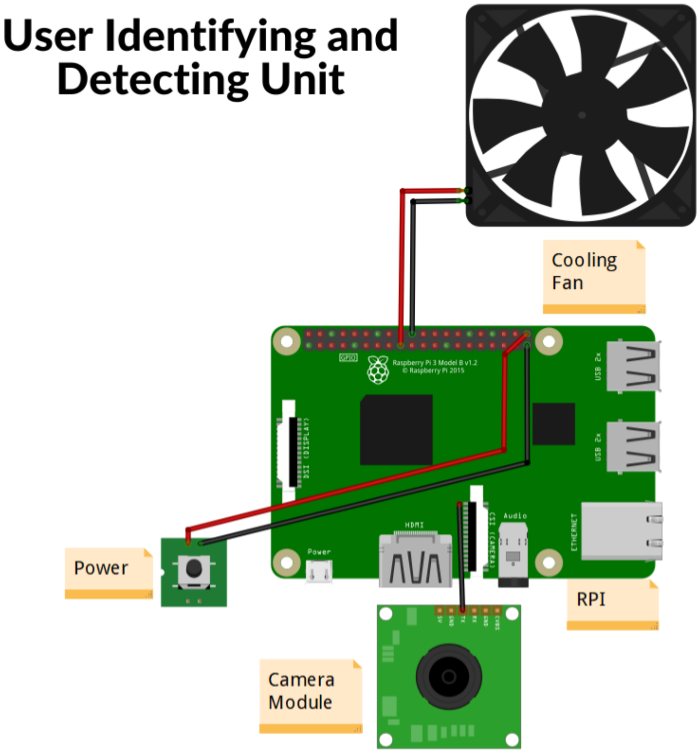
```
country=LK
ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev
update_config=1

network={
ssid="Mobitel"
psk=dbedc6954827ad7187466e42f4df78d9108955778959b5db11920f8c328dd71
6}
```

Then, we save that file inside the Raspberry Pi SD card in the /boot/ volume, so it will be /boot/wpa_supplicant.conf.

Functional Details and operations

Functions/Code	Process / Output
on_snapshot(doc_snapshot, changes, read_time)	callback function to capture changes in firebase fields in real-time
create_slides.sh	bash script to get all the created playlists (google slide links)
female_(ageRange).sh	bash script to get all the female playlists with a given age range
male_(ageRange).sh	bash script to get all the male playlists with a given age range
generic_slide.sh	bash script to get all the generic playlists
delete_slides.sh	bash script to remove all the playlists (google slide links)
showAd(adType,adAge)	function to show ad from firestore

5.2. User Detecting and Analyzing Unit



Hardware Components

Actual Implementation

- User detecting and analysing unit mainly consists of a Raspberry Pi board and a camera module.
- Images of the people in front of the screen are processed inside the RPI board.
- The Haar Cascade classifier is used to detect faces from the images.
- Open CV libraries are used with a trained data model to obtain the predicted age and gender.
- Predicted age and gender is updated real time in Firebase.

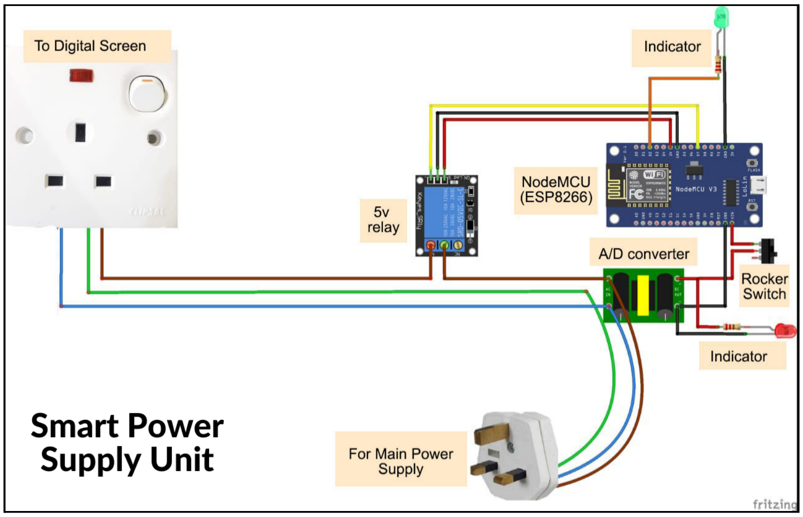
Functional Details and operations

Functions/Code	Process / Output
cv2.dnn.readNetFromCaffe (protoPathage, modelPathage)	Initialize the trained data model
cv2.CascadeClassifier(face_cascade_path)	Haar-cascade detection in opencv
cv2.imread	loads an image from the specified file
cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)	convert an image from one color space to another
cv2.dnn.blobFromImage (imgFace, 1, (227, 227), MODEL_MEAN_VALUES, swapRB=False)	creates a 4-dimensional blob from the image. Optionally resizes and crops image from center, subtract mean values, scales values by scalefactor , swap Blue and Red channels.
face_cascade.detectMultiScale(imgGray, 1.2 , 2)	Detects objects of different sizes in the input image
cv2.rectangle(image, (x,y),(x+w,y+h),(0,0,255),2)	used to draw a rectangle on the image

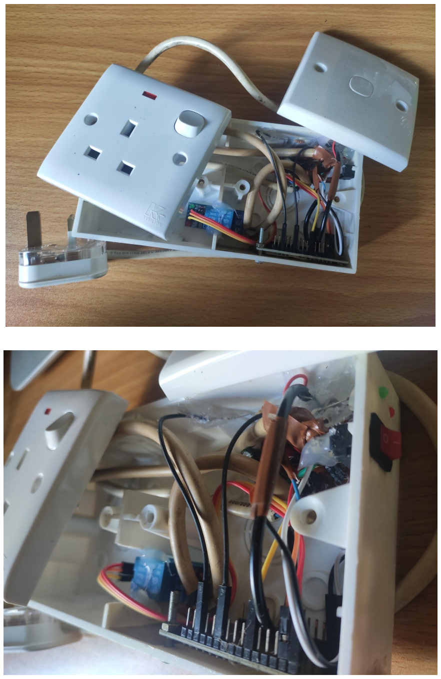
age_net.forward()
gender_net.forward()
	Runs a forward pass to compute the net output

customerAnalysis(adType,adAge)	function to count users according to gender and age

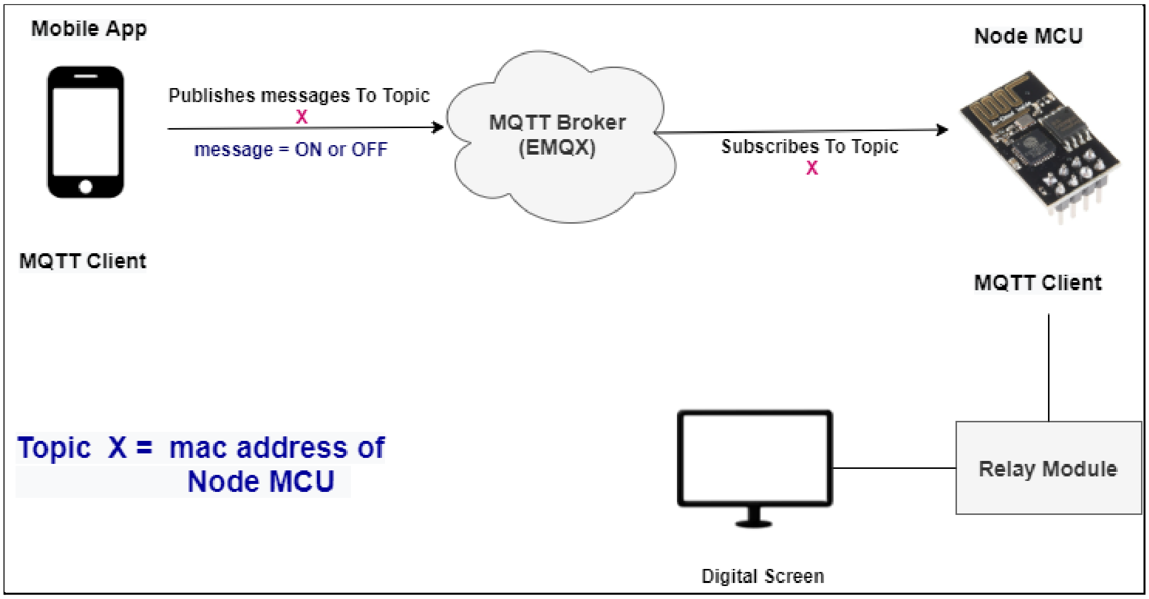
5.3. Smart Power Supply Unit



Hardware components



Actual Implementation



Software Implementation flow diagram

- The digital signage owner can turn on/off the digital screen using the provided mobile app.

- ❑ Main hardware components of this unit are NodeMCU and the relay module, A/D converter and rocker switch.

- ❑ MQTT protocol is used to implement this unit. MQTT client (mobile app) publishes on/off messages to a specific topic.

- ❑ The Mac address of the NodeMCU is used as the topic to control each unit separately.

- ❑ MQTT client (NodeMCU) subscribes to the particular topic and receives the on/off messages.

- ❑ Those signals trigger the relay module accordingly to control the power supply for the digital screen.

Functional Details and operations

Functions/Code	Process / Output
void setup_wifi()	Setup the wifi connection using wifimanager
void reconnect()	(A Loop) Attempt to reconnect when the wifi connection drops.
void callback(char*,byte*,int)	Callback function to receive payload(message) from mqtt clients
WiFi.macAddress()	Fetch the MAC address of the esp8266 nodeMCU
client.setServer(server,port)	Setting up the mqtt server
client.setCallback(callback)	Callback function initialize
wifiManager.autoConnect(username,password)	Create a local hotspot if configured wifi is not available.
Used header files: `<ESP8266WiFi.h> , <PubSubClient.h> , <WiFiManager.h> , <DNSServer.h> , <ESP8266WebServer.h>`	EMQX used as the MQTT broker

6. Tests and Performance Measures

6.1. Unit tests

- For unit testing each of the functionalities in the 3 units has been tested using mock tools and black box testing. Based on the results made conclusions to improve the overall design to minimize drawbacks of the system and provide the required functionality in an effective manner.

❑ **Test 1**

Objective:

- Front end application optimization, by user entry validations for invalid form fills in login and profile pages.

Testing Tools:

flutter_test packages

Results:

- Username,age,contact details are validated with testing carried for various invalid inputs.
- Informative guidance for valid form fills are added to the front end to improve user experience and avoid inaccurate database storing of user details.

❑ **Test 2**

Objective:

- Validating each user's MacAddresses from firestore retrieval.
- Devices add/drop functionality and drop down menu in app syncing with firestore collections.

Testing Tools:

flutter_test packages
cloud firestore mocks

Results:

- Purchased devices by user are added correctly to the dropdown menu inside the app once logged in with correct user credentials.
- Add/Remove devices from the active devices menu is tested using mocks and functions retrieve from firestore collections.
- Analytics for the count of male/female detected is synced with the user detecting unit and updates the app analytics page through firestore realtime retrieve.

❑ Test 3

Objective:

mqtt messages retrieving through the app to the power supply is tested here. Specifically,

- Subscribing to a specific MacAddress topic and reconnecting in wifi failure with last published message status capturing.

Testing Tools:

MQTT X desktop test client tool

Results:

- Reconnecting with a timeout (a loop) added for the smart power supply unit once the wifi is disconnected.
- Uniqueness of MacAddress used as the topic to subscribe by each device.
- Current status of each digital screen(on/off) is stored in firestore collections once the on/off message is successfully published.

6.2. Face detection algorithms accuracy measures

For the face detection process several haar-cascade trained algorithms were used. Each has been tested with a sample image set and choosed the most accurately predicted model to use in the user detecting and identifying unit.

- **Test**

Objective:

To check accuracy of predictions when multiple faces appear in front of the camera.

Testing Tools:

Python unit testing tool

Unit testing done with Black box testing with the pre-trained cnn model (developed by Gil Levi and Tal Hassner using caffe framework)
https://talhassner.github.io/home/publication/2015_CVPR

Results:

No. of Faces per captured frame	No. of test cases	Passed test cases
1	20	19
2	20	15
3	20	11
4 or more	20	10

- Input images with one person in different positions are identified up to 85-95% accuracy. When more than that, people in a frame were captured with accuracy around 50-70%.
- Different haar-cascade models tested for face detection with sample images and the most accurate model identified and used.
- With the results obtained, the conclusion was to show a generic ad category when more people are detected.

6.3. User experience enhancements

In mobile app,

- Two factor authentication used with shared preferences option.
    - Users need to login initially using phoneAuth and Google sign in. The credentials are saved using shared preferences for easy login to the main page in subsequent logins.

- Rename features for each added device.
    - Both Digital Signage units and Smart Power Supply Units can be renamed to a desired name by the user to easily recognize and use. (Only in initial addition of units need to enter the serial number)

- Customer Analytics feature
    - Shop owners can identify each day about the customer count that came to the shop in each gender and age group. Using analytics can use strategies and customize the advertisements for maximum outcome.

- Customize user account
    - Users can keep an account of their own and also login from different google accounts as fits to the user.

In smart power supply,

- Wifi manager library feature
    - Users can configure the smart power supply unit easily to a desired wifi network following simple steps.

- Change configured wifi to another
    - Saved wifi credentials can be changed through the wifi manager portal.

- Can operate in low bandwidth conditions.
    - Using mqtt lightweight messaging protocol can function properly without interruption.

- Wifi connection drop outs managed.
    - In case of a wifi failure the power supply keeps the screen in the previously known state until wifi reconnects.
    - After connection is stable again automatically connects and is able to receive messages instantly through the mobile app.

7. References

- For Raspberry Pi 3B configuration and usage
  https://www.instructables.com/Ultimate-Raspberry-Pi-Configuration-Guide/

- Wireless network configuration with raspberry Pi
  https://www.raspberrypi.org/documentation/configuration/wireless/wireless-cli.md
  https://www.raspberrypi.org/documentation/configuration/wireless/headless.md

- Esp8266 NodeMCU datasheet and configuration
  https://components101.com/asset/sites/default/files/component_datasheet/ESP8266-NodeMCU-Datasheet.pdf

- Wifi manager Library configure with esp8266
  https://github.com/tzapu/WiFiManager

- Trained CNN models for age and gender prediction.
  https://talhassner.github.io/home/publication/2015_CVPR