ALGODETECTIUM

We're hiring

ALGODETECTIUM

FINAL REPORT

Submitted to Assoc. Prof. Elif Vural

Alperen Kalay	2232106	kalay.alperen@metu.edu.tr
Buğra Eryılmaz	2231926	bugra.eryilmaz@metu.edu.tr
Hüseyin Berat Karakaya	2232171	huseyin.karakaya@metu.edu.tr
Kutay Uğurlu	2232841	kutay.ugurlu@metu.edu.tr
Özgür Diyar Kozan	2232353	kozan.ozgur@metu.edu.tr

Contents

EXECUTIVE SUMMARY

Nowadays, humanity goes through one of its toughest times due to the COVID pandemic. To decrease the spread of this disease, people should obey some rules such as self-quarantine while at risk, keeping the social distance, and wearing masks properly. These rules are even more critical in crowded public places such as shopping malls and plazas. We, as AlgoDetectium, intend to ensure that people entering these buildings obey the rules with our surveillance system HealthPass. Our system automatically detects and restricts the entrance of people who do not follow the rules or show any detectable symptoms of COVID-19. In this document, we will describe our solution in detail.

We have divided the problem into subproblems in order to achieve a more structured solution. Our main solution includes:

• Face Mask Detection categorizes people into three as in, properly worn mask, improperly worn mask, no mask. This is achieved by making use of Convolutional Neural Networks. (CNN)

• Loss of Smell Detection is testing the smelling ability of the entrant. This is achieved by asking them to use a smelly disinfectant and identify the odor.

• Fever Detection Subsystem includes a Touchless IR Sensor that measures the body temperature of the person.

• People Counter Subsystem updates the log in the database when a person enters the building or exits the building. The People Counter Subsystem increases the number of people inside the building by one when Decider Subsystem allows the person and decreases it by one when the distance sensor detects a leaving subject.

• Decider Subsystem combines all the data from other subsystems and the database. Based on an algorithm, it evaluates if a person can enter the building or not. Then it unlocks the door depending on the outcome.

To our customers, we will provide the necessary hardware and software as a bundle. The hardware will include a camera, Raspberry Pi device as a platform, distance sensors, a locking mechanism, foot buttons, LCD display, disinfectant stands, and a contactless temperature sensor. The software we will provide is the HealthPass software.

Each of the discussed subjects will be explained in more detail in their own respective sections. We are planning to implement all our solutions in a robust way. The system will be easy to use and maintainable. It will not expose any risk of COVID-19 infection.

INTRODUCTION

Since SARS-Cov2 started to spread, many scientists and engineers have been studying on various projects to eliminate its effects on our daily lives. As AlgoDetectium, we aim to design a system called HealthPass which helps the fight against the pandemic by detecting its symptoms to keep the possibly infected subject from spreading the virus and by forcing people to wear face masks in publicly used buildings. All the subsystems conducting HealthPass tests are run on Raspberry Pi 4B.

According to our design, at first, we detect whether the subject who wants to enter the building shows the symptoms of the virus which are loss of smell and fever. Those are both accepted as the symptoms by the World Health Organization. For loss of smell detection, the subject is presented with some disinfectants with a certain scent which they choose by pressing the foot buttons. We expect the subject to smell and detect which odor the smell belongs to. A screen will be displaying several options for the scent and the subject will show their choice by pressing the foottons. Then, we measure their body temperature with the help of a temperature detector. These two subsystems together decide whether the subject shows the symptoms of the virus.

After the detection part, we check if the entrant wears a mask properly by which we mean nose and mouth are covered completely. The face of the subject is obtained by the camera and the algorithm that we use which is trained by many images decides whether the mask is worn properly. If all health and mask conditions are satisfied, we allow the building door to open and let the person enter while increasing the number of people inside the building by one. This information is kept in a database. We also count the people who are leaving with the help of a distance sensor. That way, we are able to keep the building from being so crowded that the rules of social distancing are violated.

This report consists of an overall description of the project, including a block diagram showing the subsystems, their inputs/outputs, and the information flow between them. A flowchart explaining the main decision logic and technical drawings of the whole system is also included. Each subsystem is explained with individual block diagrams. Functional and performance requirements are provided as well. Moreover, we include test results for those subsystems.

After detailed explanations of each subsystem, a cost-breakdown analysis is presented with the reasons for our choices. Power consumption and necessary power sources are also analyzed and included in the report.

OVERALL SYSTEM DESIGN

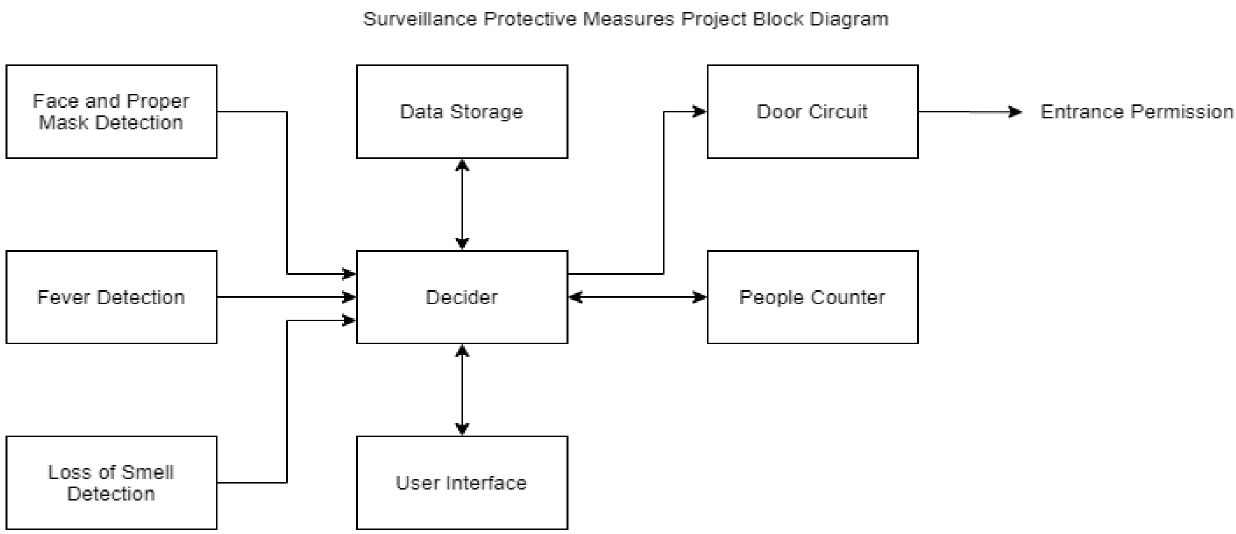

Figure 1. The Overall System Block Diagram

Our system is a fully automatic entrance permission system that is going to decide based on the tests for COVID-19. This system aims to detect people showing the symptoms of COVID-19 to prevent them from entering the building to protect public health in an automatic manner.

Regardless of their symptoms, people will not be allowed to enter the building until they wear their masks correctly. In addition to this, it is going to be made sure that the hands of the incoming person will be sterilized via disinfectants. Another usage this system provides is to track the number of people entering the building to avoid overcrowding, providing an automatic way for keeping social distance between the people inside the building.

The General Flow of Information

The general input-output relationship, information flow, and decision mechanism of the overall system are given in Figure 2 below. The relevant information regarding the system variables and signals are as follows:

- **Check.smell:** The system checks if the disinfectant buttons are pressed.
- **Prompt.options:** The system prints the smell options to the screen.
- **get.answer:** The system takes the response from the smell selection buttons.
- **correct.smell:** If the subject has identified the smell correctly.
- **Prompt.temp:** The system guides the user to measure body temperature via UI.
- **Get.temp:** The body temperature is measured.
- **Temp.safe:** The measured body temperature is below 37.5℃.
- **Temp.notCritical:** The measure body temperature is below 38℃.
- **Prompt.mask:** The system guides the user to show his face to the camera via UI.
- **GetCapture:** A frame is received from the video stream.
- **IdentifyMask:** The received frame is classified through the neural network.
- **ProperMask:** If the subjects wear their mask properly.
- **Prompt.Improper:** The system guides the user to adjust his mask via UI.
- **DB.getPeople:** The decider unit receives the number of people inside the building from the database.
- **Entrance.Allow:** The decider unit has reached the conclusion to allow the entrant.
- **DB.Log:** A log describing the test results and the relevant decision is sent to the database.
- **DB.People.Count:** A signal indicating the number of people should be incremented is sent to the database.
- **Entrance.deny:** The subject fails to satisfy the HealtPass test conditions.

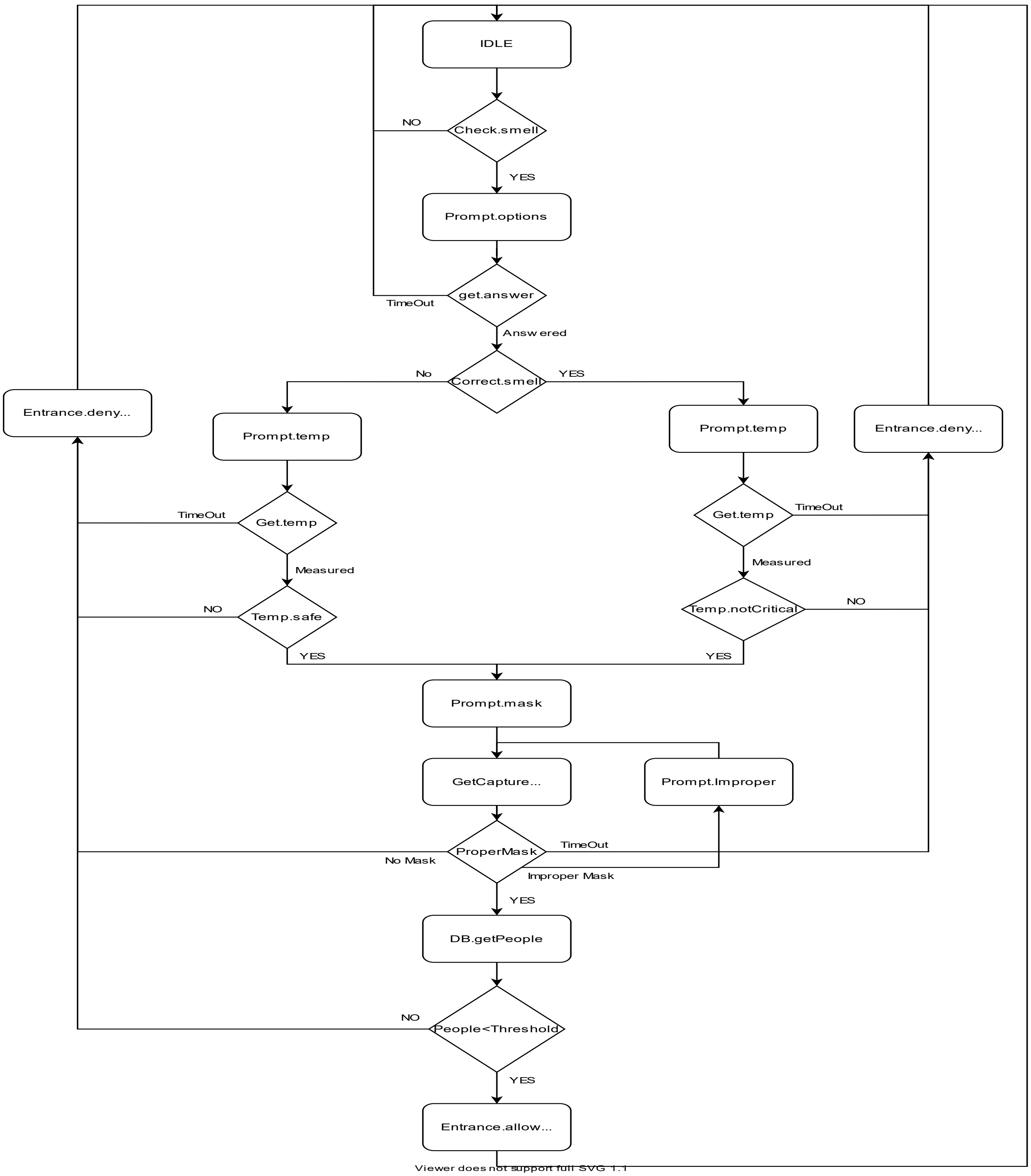

Figure 2. The general information flow in HealthPass

The person approaching the building is going to be first subjected to the temperature test. The body temperature of the subject is going to be measured via a contactless sensor that is placed at the entry of the test cabinet. After this test is complete, the person is going to be subjected to the Loss of Smell test. A disinfectant that has a sharp and easy-to-distinguish smell is going to be released via pedal, based on the choice made by the subject, on the subject's hands to sterilize their hands and they will be asked to identify this smell again by a pedal to choose from the options presented by a screen. For the last step, the person is going to be subjected to the mask test, by being asked to approach a camera to show their faces. The faces and the masks are going to be analyzed via the camera and if they wear the mask correctly is going to be determined. Until their masks are not properly worn, they will not be allowed to enter the building. A decider unit, which is going to allow the subject to enter the building, is going to produce a signal for the door-lock circuitry subsystem, based on an algorithm that the decider subsystem uses to combine all the data coming from different subsystems. In addition to what was described above, the people counter subsystem is also going to count the number of people allowed to enter inside the building. The People Counter subsystem decides on the number of people using the inputs coming from the Decider Unit and the distance sensor that is utilized to count the exiting people from the building.

System-Level Requirements

Functional Requirements

The functions that all proposed solutions have to accomplish are given below:

- Contactless body temperature sensing
- Directing to subject to sterilize their hands
- Releasing disinfectants with odor
- Presenting options for different smells
- Getting the user input from the mechanical platform
- Detecting the face of the subject
- Detecting the mask on the face of the subject
- Determining the properness of the mask
- Counting the number of people inside
- Deciding if the subject is allowed to enter the building
- Generating an unlock signal for the door-lock system

Constraints

This system is proposed for detecting risky people and blocking the risky behavior inside the buildings. Having been designed to slow down the infection rate of the illness in the building, our system should be designed according to this purpose.

• The tests should not contain any process that requires physical contact with the devices.

• The system should take inputs from the subjects contactless or without contact of bare hands.

• The overall cost should not exceed 200USD.

Performance Requirements

• The decision on the entrance allowance should be concluded in less than 30 seconds.

• The accuracy of the overall system should be above %70.

• Test cabinet should be ready for being reused by the next incoming person.

• The overall system should provide enough space for the tests to be taken.

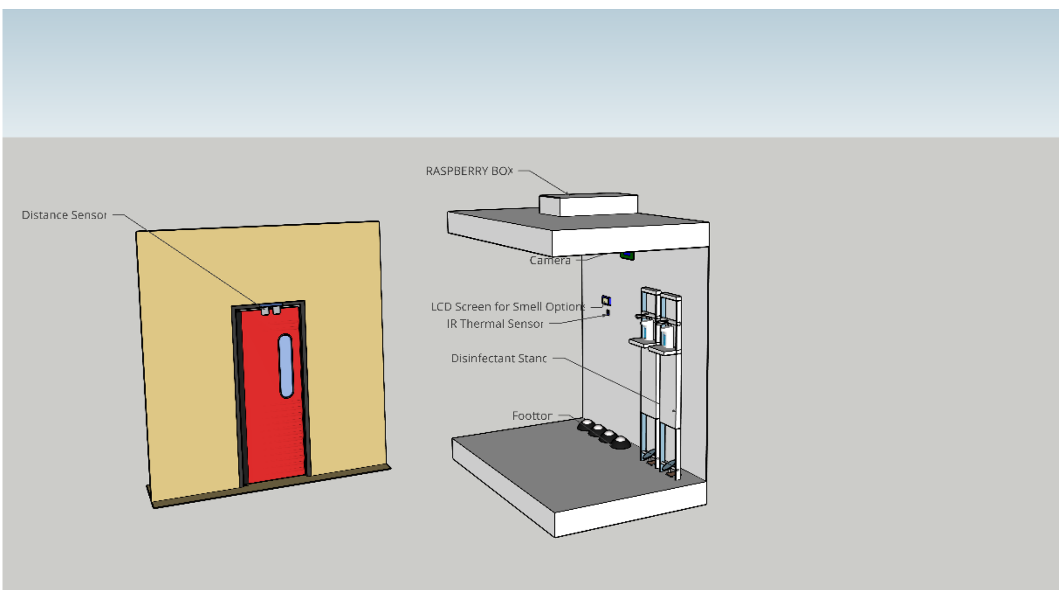

Figure 3. The overall view of HealthPass

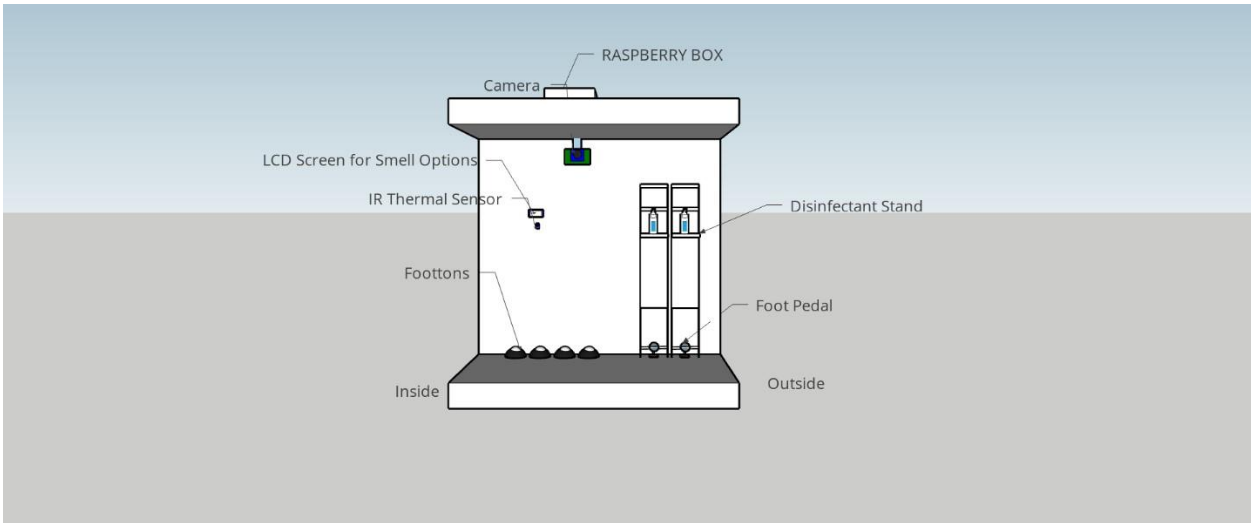

Figure 4. The test cabinet of HealthPass

Circuit Diagram

We used Raspberry Pi 4B, which has pinout configuration as below:

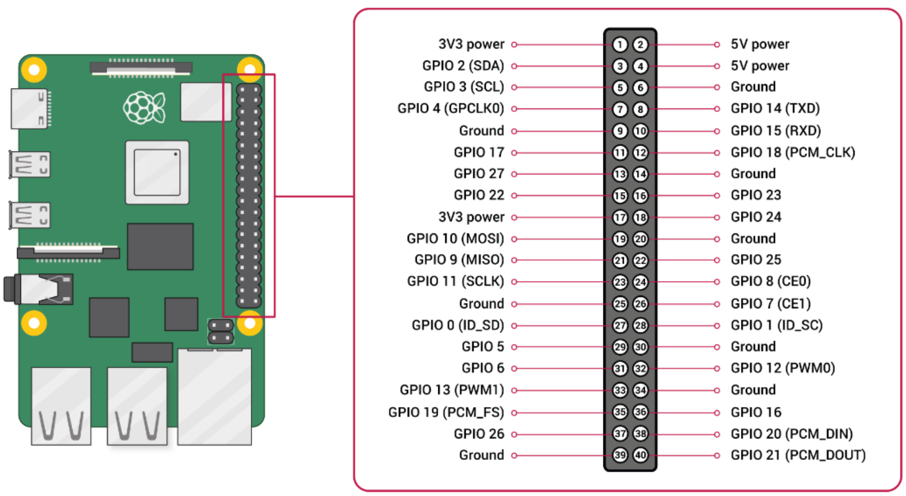

Figure 5. The Pinout Diagram of Raspberry Pi 4B [8]

Connections are made considering available pins and communication protocols of sensors. In Table 1, yellow parts represent device pins that we connected to Raspberry Pi.

Table 1. Raspberry Pi 4B Pinout Used Pins

Temp_Vin	**3.3V**	**5V**	5V
Temp_SDA	**SDA**	**5V**	5V
Temp_SCL	**SCL**	**Gnd**	Ground
-	**GPCLK0**	**TXD**	-
-	**Gnd**	**RXD**	-
Footton (Yellow 3)	**GPIO 17**	**PCM_CLK**	-
Footton (Yellow 2)	**GPIO 27**	**Gnd**	-
Footton (Green 1)	**GPIO 22**	**GPIO 23**	*Distance(Temp)-Echo
3.3V	**3.3V**	**GPIO 24**	*Distance(Temp)-Trigger
LCD_Din	**MOSI**	**Gnd**	-
-	**MISO**	**GPIO 25**	Relay(Lock)
LCD_CLK	**SCLK**	**CE0**	LCD CE
-	**Gnd**	**CE1**	-
-	**ID_SD**	**ID_SC**	-
LCD_Reset	**GPIO 5**	**Gnd**	-
LCD_DC	**GPIO 6**	**PWM0**	-
-	**PWM1**	**Gnd**	-
Footton (Green 2)	**PCM_FS**	**GPIO 16**	LED
Footton (Yellow 1)	**GPIO 26**	**PCM_DIN**	*Distance-Trigger
Ground	**Gnd**	**PCM_DOUT**	*Distance-Echo

*We used 2 distance sensors. Distance(Temp) is the one working collaborative with temperature sensor. Other one is for people counter subsystem.

SUBSYSTEMS

In the following sections, we analyze and explain each subsystem of the HealthPass.

Face Mask Detection Subsystem

Face Mask Detection Subsystem is the most critical subsystem in our project. This subsystem is responsible for detecting the faces and determining if that person wears the mask properly or not. The implementation will be done on a Raspberry Pi 4 with a camera module. The process consists of image processing and classifying tasks. With the increased availability of the data and new algorithms, neural networks became more successful in classifying tasks. Considering also that the usage of them has become extremely common in the past decade and many projects have been done using this approach, we decided that our approach to solve this classification task is using neural networks. However, neural networks come with a trade-off. That is, increasing accuracy on test data requires more complex functions to be learned, more complex and deep networks, yielding more parameters and more computational power.

This subsystem will have mainly 2 inputs, namely inputs from the fever detection subsystem and camera. The fever detection subsystem will prompt this subsystem to start the face-mask detection procedure. Then, the camera will send 3 photographs of the subject to the face detection unit. The face detection unit will determine how many faces there are and where they are in the photograph. After determining the faces, the face detection unit will crop and resize them to match the mask detector's input which is 224x224 pixel. The resultant face images will be sent to the mask detection unit and storage unit. The mask detector unit will get all the faces from the face detector and classify them as masked, improperly masked, and not masked. Finally, the consecutive decision unit will take classifications over the 3 pictures and send a single signal of masked or not masked to the decider subsystem.

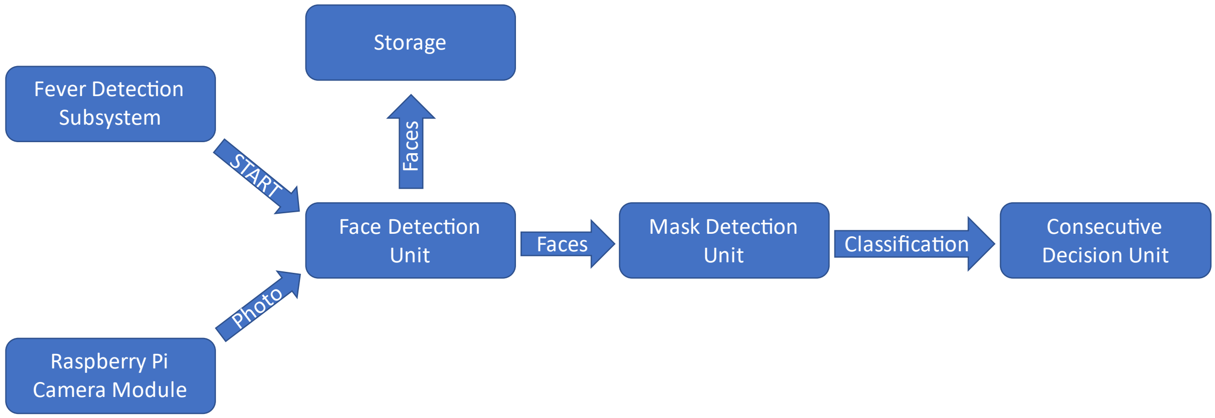

Figure 6. The subsystem diagram Face Mask Detection Subsystem

Requirements

In this subsection, the requirements of the Face Mask Subsystem will be discussed in detail. Firstly, the functional requirements will be presented. After this section, the performance requirements will be described.

Functional Requirements

- The Face Mask Detection Subsystem should detect the faces and their relative positions in the image.
- The Face Mask Detection Subsystem should classify the cropped face pictures in terms of mask status.
- The Face Mask Detection Subsystem should store the cropped face pictures in the local storage.
- The Face Mask Detection Subsystem should produce a final decision based on 3 consecutive face mask classifications.

Performance Requirements

- The detection should be done in less than 10 seconds in order to meet the requirement that the overall process should finish in less than 30 seconds.

- The accuracy of the mask classifier should be at least 75% to meet the requirement that the overall system has at least 70% accuracy.
- The maximum angle of detection span should be at least 30°.
- The mask detection should be done in a range of 20-50 centimeters to meet the requirement that all tests can be conducted within the platform designed for the whole system.

Solution

The proposed solution consists of two neural network models. The first model is for detecting the faces from the images. After this operation is done, the face is cropped and resized so that it can be fed to the second neural network that classifies if the face has a mask or not. The first model is implemented using a Resnet10_SSD architecture using Caffe. The second network is a fine-tuned MobileNetv2 network initialized with Image-Net weights, following a transfer-learning approach. We will also consider 3 consecutive images with these neural networks, and we will generate a final decision as masked if all three results are masked. To achieve the desired accuracy, we grouped different masked faces under 3 class, being Properly worn, Improperly Worn and Not Worn, and designed the network accordingly.

We have decided to use these network architectures since they provide high accuracy and high speed for mobile applications. Since our system will run on a Raspberry Pi 4B, we needed a lightweight model with high enough accuracy. Our test results have shown that these architectures are suitable for our application since provide 83% accuracy within 0.15 seconds of processing time.

The described system also satisfies the detection angle requirement. As it can be seen in the test results section, the system can detect the faces in the range of 0° to approximately 35° which is more than enough to satisfy our needs. All our tests are conducted in the range of 20-50 cm distance, so it is safe to say that the system is able to satisfy the distance requirement.

Test Results

Keeping in mind that the images to be used in the test procedure should not have the same distribution as the ones used in training, we have searched for a few numbers of sources on the internet to be used as test dataset. However, for improper mask class, the sources on the internet did not provide enough number of pictures compared to other classes. Thus, we decided to gather more images from our social circle to the test dataset. Finally, we formed a dataset consisting of 477 images collected from different sources including the photos taken by the individuals in the team members' social circle.

The test procedure consists of two parts for two models. First, we feed the images to the face detector model, whose outputs later to be used in mask classification. In this procedure, the model detected duplicate faces in 5 of the images. Except those, 38 of the remaining images were removed from the dataset due to low resolution caused by the cropping operation of relatively small faces in the image. The model was successful at detecting the remaining faces in terms of both the number of faces and the location of faces. The detected faces were cropped out of the image and resized to match the input dimensions of the mask classifier, which is 224x224 pixels. Then, the faces were fed to the mask classifier network and the following evaluation metrics in Table 1 were used to validate the success of this subsystem.

Table 2. Precision Recall table regarding the Mask Classification subsystem tests

	Precision	Recall	F1-Score	Support
Improper Mask	0.86	0.51	0.64	96
With Mask	0.89	0.89	0.89	228
Without Mask	0.73	0.98	0.84	110
Overall Accuracy			0.83	434

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative}$$

$$F1 - Score = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

Another way to visualize the success of the model is using confusion matrix which can be seen in Table 2.

Table 3. Confusion Matrix regarding the test of Face Mask Classifier

		Predicted Labels		
		Improper Mask	With Mask	Without Mask
Ground Truth Labels	Improper Mask	49	24	23
	With Mask	8	204	16
	Without Mask	0	2	108

The predictions regarding the without mask class images were successful in terms of recall value of 0.98. The predictions regarding the with mask class images also were mostly successful in terms of its recall value of 0.89. However, since this class is more critical in the final decision process, we wanted to decrease its false positive rate, so we decided to use 3 consecutive video frames and expect all of them to be with a mask to grant permission. Thus, the rare inaccuracies are handled without a problem. For the improper mask classification, the main objective is to distinguish this class from with mask class, a confusion between improper and without mask classes is not a problem while allowing the subject to enter the building. In that sense, the improper masks and without mask class can be considered as a single class, yielding a recall value of 0.75. Moreover, we have attributed the reason behind the model predicting improper mask images as with mask to the fact that the improper mask dataset contains images where masks cover faces from slightly under the nose to the chin which we expect to observe rarely in real-life cases and consider as edge cases. One such example is illustrated in Figure 7.

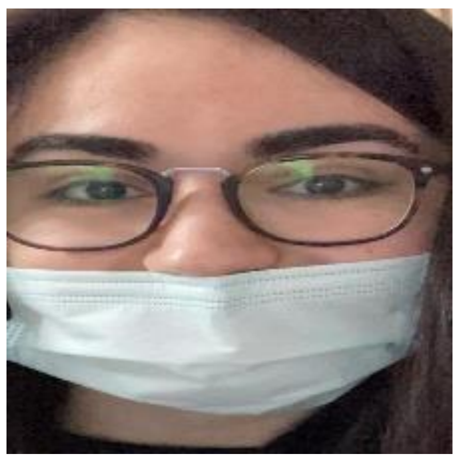

Figure 7. An example edge case for Face Mask Classifier

The tests are done in the Raspberry Pi 4B device and in total it took 66 seconds to classify 434 images. Thus, we need 0.15 seconds to process an image on average. To sum up, the results of the tests are as expected, and they comply with our performance requirements.

Database Subsystem

The database subsystem is a crucial block for the scalability of the HealthPass system. Without the database subsystem, the proposed solution would work for a single entrance single exit building. Using a common database, we provide the necessary communication between raspberry devices that are installed in each of the entrances. All the Raspberry Pis will be connected to the same database, and so they can share information between them. Whenever a person enters or leaves the building from a door, the raspberry pi used for that door will update the number of people inside of that building in the database. Thus, other Raspberry Pis at the other entrances can get the correct number of people and decide on granting permission to entrances accordingly.

Moreover, this subsystem is also crucial for the maintainability of our system. When or if a problem occurs in one of the products, we should be able to track the root of the problem. That is why keeping the logs of the system at an external database will enable us to identify and work on problems without the need of going to the physical location. The logs will be desensitized before storing in an external database due to legal requirements. In other words, the logs will only include anonymous data. Also, again to increase the maintainability, the database will be responsible for providing a User Interface to the customers of the HealthPass system for changing the smell options shown on the screen.

The subsystem will have 2 inputs and 2 outputs in the overall system. The inputs will come from the People Counter subsystem and Decider subsystem. The People counter subsystem will send an update query to the number of people inside that store data whenever a person leaves so that the system will be able to maintain the correct number of people inside information. The Decider subsystem will send an update query to the same data whenever it permits a person to enter the building. We expect all the people who are granted this permission to enter the building since their purpose for using our product is to enter the building. The Decider subsystem will also send a log of the tried entrance to the database whenever a person tries to enter the building. The log will include information about their responses and all the variables used in the decision phase. One of the outputs of this system will be to the decider subsystem since the decision process needs to access to the number of people inside information to conclude the assessment of the incoming subject. The other output of this subsystem is to the User Interface subsystem since the customers will change the options shown on the screen via the Database Subsystem.

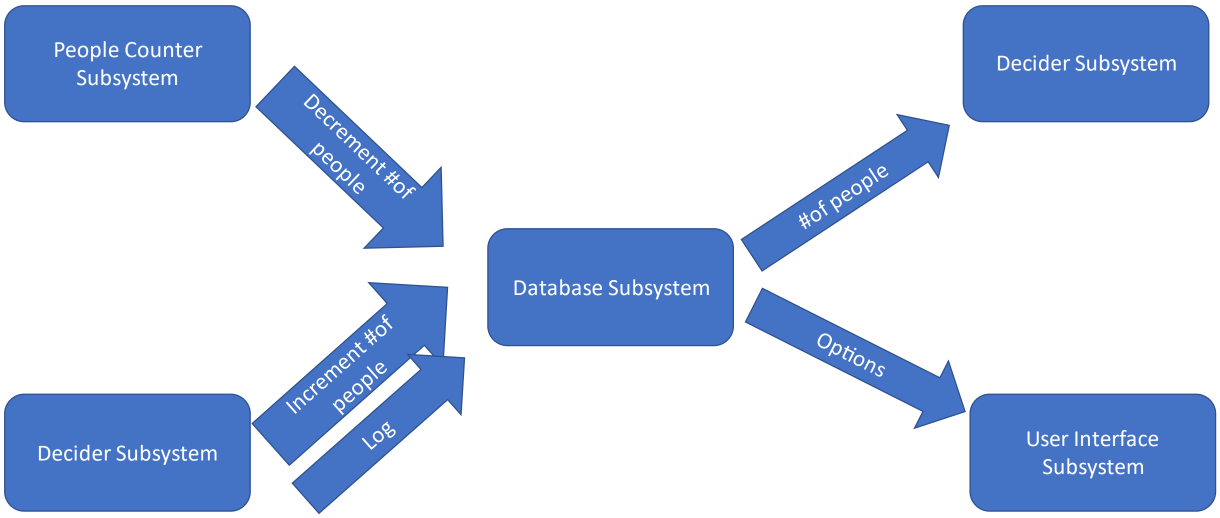

Figure 8. Database Subsystem Input/Output Diagram

Requirements

In this subsection, the requirements of the Database subsystem will be discussed in detail. Firstly, the functional requirements will be presented. After this section, the performance requirements will be described.

Functional Requirements

- The Database Subsystem should enable information exchange among different Raspberry Pis.
- The Database Subsystem should provide an interface to the customer to choose User Interface options.
- The Database Subsystem should be accessible from anywhere.
- The customers should have restricted access to the database.
- The Database Subsystem should store data.

Performance Requirements

- The Database Subsystem should be able to store at least 1 week of data.
- The Database Subsystem should not introduce a delay of more than 1 second.

Solution

Due to the previous experience our team members had; we have decided to use NoSQL databases. The database MongoDB has a free-to-use plan that consists of 512MB storage and 100 operations per second throughput. This plan will be more than enough since we will not store any image, instead, we will store the logs storing the information regarding current time, temperature, response to the smell test with the expected response and a binary data representing proper mask usage and the number of the people inside the building at that time. An exemplary log and the number of people information can be seen in Figure 9. Moreover, we will be keeping the log of the last 7 days, so 512 MB storage limit is not expected to be a constraint.

```
_id: ObjectId("5ff99e973b71bdf9a0f0acb9")
date: "Sat, 13 Jun 2020 18:30:00 GMT"
temp: 37.2
expected_ans: "strawberry"
ans: "banana"
facemask: "improper"
peopleInside: 42
```

```
_id: ObjectId("603a3c6f28563f4fa4317270")
storeName: "AlgoDetectium5"
peopleInside: 1
```

Figure 9. An example database log

The database will also store an entry that is only accessible by the store owner to change options presented in the User Interface module. That entry will include 3 options to be changed, 2 correct smell names, and the maximum allowed number of people inside information stored. When the customer wants to change them, the customer will open the MongoDB application and enter a store-specific URL to the application to connect and update the configurations entry. An exemplary configurations entry is illustrated in Figure 9.

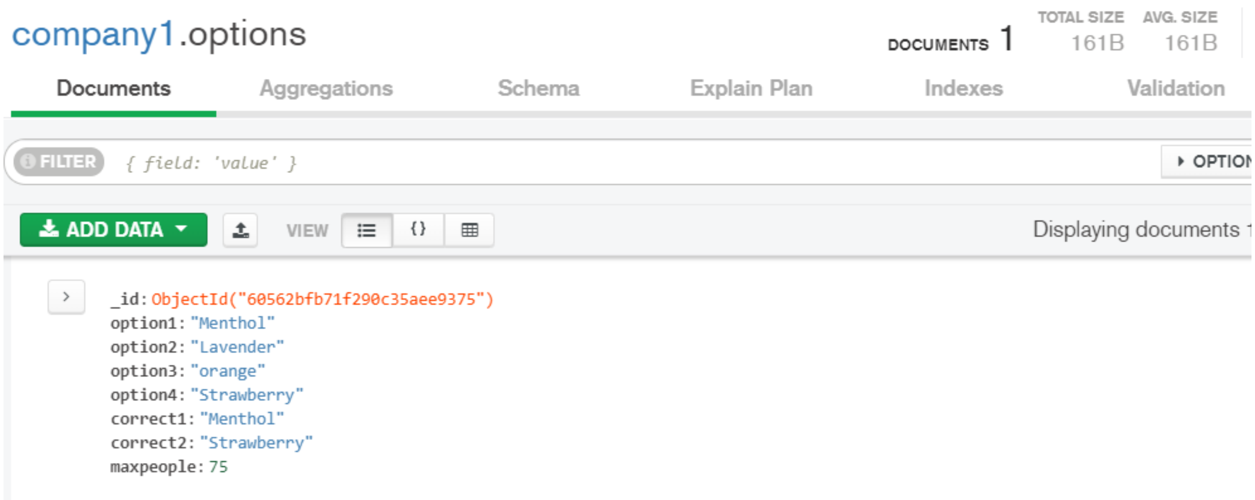

Figure 10. Interface for companies to change their options.

In order to increase the functionality of our system, we decided that the options shown on the screen should be changeable by our customer. If the options were hard-coded, whenever the smell of disinfectants changes, we had to manually adjust the options. That is why we needed an interface for customers to change the options. The database already provided us a graphical interface to see the logs and the security options to adjust so that the user will only see their information. Thus, adding a new user for the customer and adjusting their privileges over the database was more trivial than other options for providing an interface. That is why we decided to have that functionality over the database subsystem.

Test Results

The subsystem is implemented on the Raspberry Pi 4B to see if it satisfies the functional requirements. The MongoDB database has provided us all the functional requirements over a Python script which was ideal for us since the whole system is implemented on Python. After assessing the database for our functional requirements, we tested it to verify the fact that it also complies with our performance requirements as well.

We had 2 performance requirements for this subsystem. One of the requirements is it should be able to store at least 1 week of data. For this requirement, we tested how much storage each entry requires. We also used 21.000 customers per week to estimate how much storage we need for a week of data. The results are then tabulated. As can be seen in Table 3, we need 2.8 MB of storage per week for storing logs. Since MongoDB provides 500 MB of storage, it is enough to store more than 100 building's data. Moreover, the clipped images will be stored in the local storage. Therefore, the 32 GB local storage Raspberry Pi has is more than enough. Thus, this test is regarded as a success.

Table 4. Test results for the required storage

Data to be stored	Required Storage per entry in average	Expected number of entries per week	Expected total storage per week
Clipped Images	20 KB	21.000	410 MB
Logs	140 B	21.000	2.8 MB

The other performance requirement of this subsystem was not introducing more than 1 second of delay. To test this requirement, we have sent 100 queries to the database to store log, increment number of people, decrement number of people, and took the average of the delay. The results are then tabulated. As can be seen in Table 5, the average time of all the operations is 0.19 seconds. When we consider we will only log and increment the number of people once per entrant, the total time it is expected to take is 0.37 seconds per entrance trial. That result is less than the 1-second requirement of this subsystem, so it can be considered a success as well.

Table 5 Introduced delay for each database operation.

Operation	Average Time
Log	0.19 s
Increment number of people	0.18 s
Decrement number of people	0.21 s

Loss of Smell Detection

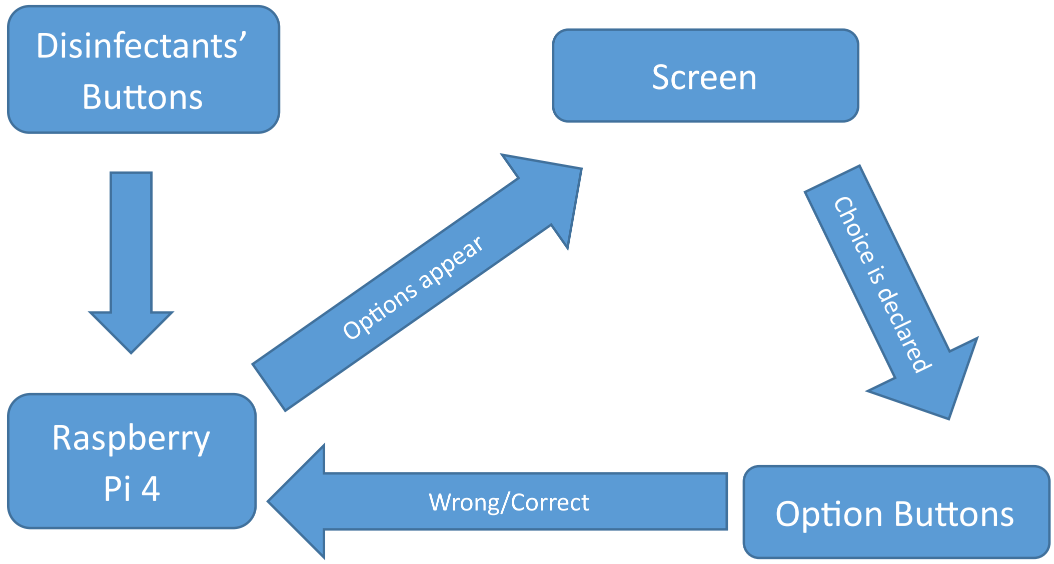

Figure 11. Loss of Smell Detection Subsystem Diagram

Solution

Loss of smell is accepted to be a symptom of COVID-19 by the World Health Organization and local authorities. We decided to use this information to increase our detection accuracy. According to our design, the subject who enters the building is supposed to put disinfectant that we provide on their hands. Our design suggests that there will be two options for disinfectants. The subject chooses one of them by pressing the corresponding footton. We considered that it would be unhygienic for people to touch the same button with their bare hands, so we decided to utilize foot buttons, i.e., foottons. 3D Drawing of these foottons can be seen in Figure 12. The reason why we have chosen to use disinfectant as a smell source is that it is already needed to keep the hands clean, especially in a publicly used building. We expect the subject to smell the scent of the liquid and decide which odor the smell belongs to.

Choice of the subject is stored in the system and after they press the button, four options will appear on a screen that is connected to the main system by Raspberry Pi. Each option will declare a certain odor and they will all be represented by a letter such as A, B, C, D. Same letters will be written on the buttons which we expect the subject to press with their foot to show their decision. If their choice matches with the previously stored information, they will be deemed to have passed this stage successfully.

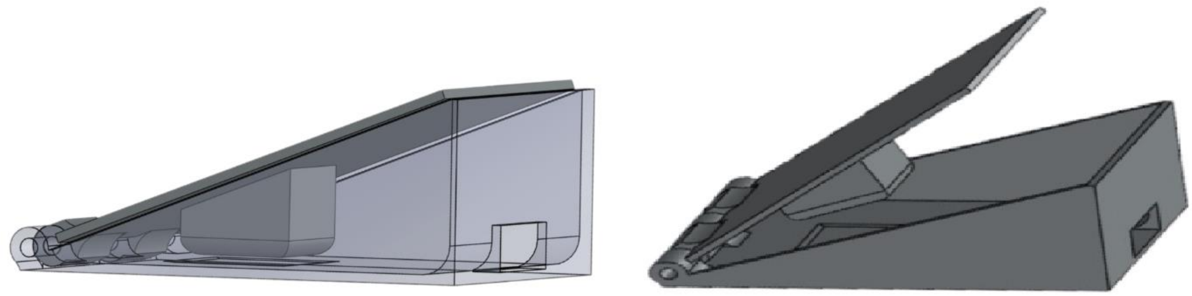

Figure 12: Footton 3D Drawing

Functional Requirements

The scent of the disinfectant that we use must be known by the people and distinguishable from the other options that will appear on the screen. Some mistakes may occur if the subject cannot understand what kind of odor that the disinfectant has even though they can smell it.

Performance Requirements

- This subsystem should be done in less than 15 seconds in order to meet the requirement that the overall process should finish in less than 30 seconds.

Test Procedure

Since it is not possible to ask an infected person to test our system, the only part we can check is the connections with the screen. For that, at first, we press the buttons of the disinfectants to indicate the choice. We expect to see the options on the screen. Then, we choose the right answer first and the screen must display “CORRECT”. When we press one of the wrong buttons, we expect to see “INCORRECT”. We have encountered no problems while testing.

Fever Detection Subsystem

This subsystem is responsible for detecting the body temperature of incoming people. The person who attempts to enter the building is going to be subjected to the fever detection subsystem after the loss of smell detection. An infrared thermometer sensor will sense the temperature of the person and inform the decider unit about this measurement. This data is going to be used as a parameter in the decider unit. Also, the person in the cabinet will be informed about this measurement via the LCD screen.

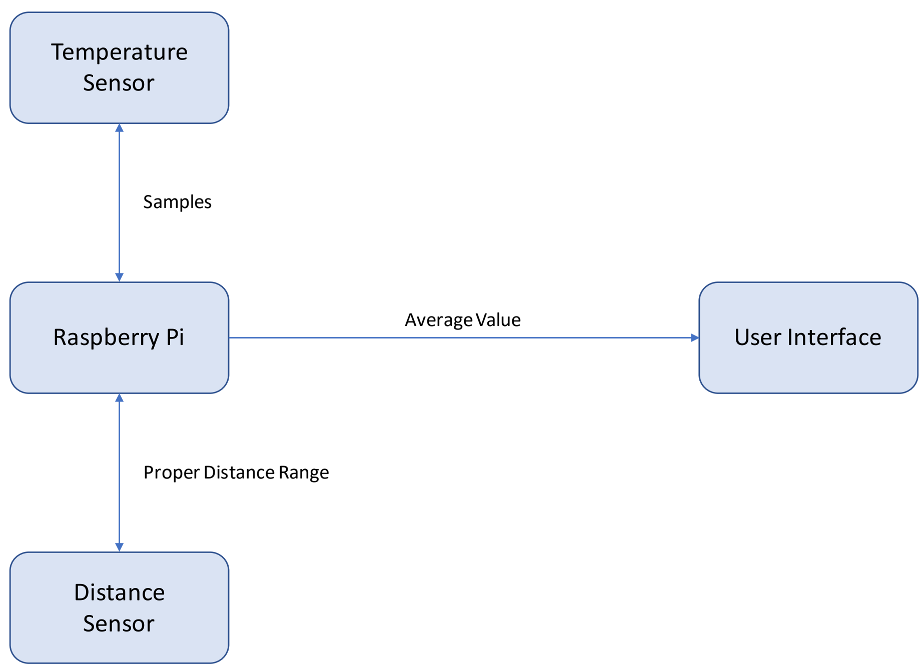

Figure 13. Fever Detection Subsystem Block Diagram

Functional Requirements

- This subsystem should be able to detect the temperature in a contactless way.
- Fever Detection Subsystem should work in four seasons.

Performance Requirements

- The error of temperature detection should be between ±0.5°C.
- Body temperature should be measured between 2-3 cm.
- Measurements should be done in a 30° field of view between specified distance and error range.
- The detection should be done in less than 3 seconds if a person follows the directives.

Solution

A contactless procedure will be applied during measuring temperature. After loss of smell detection subsystem, person in the cabinet will be informed clearly in order not to disrupt the sterile atmosphere in the cabinet. People will show their wrist or hand to sensor. Depending on the proximity, sensor is going to measure the temperature. MLX90614 works as IR thermal sensor. This sensor is selected considering the performance requirements of the whole system. Maximum performance with minimum cost is aimed. Flowchart of this subsystem is as follows:

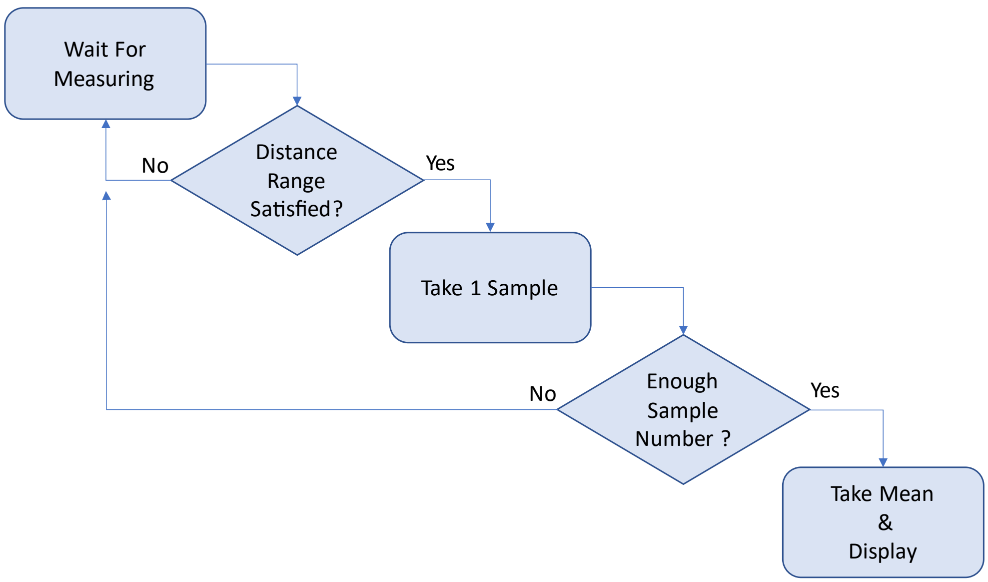

Figure 14. Flowchart for Fever Detection Subsystem

We, as AlgoDetectium, designed our system as follows:

- To minimize the measurement error and achieve the ±0.5°C range in error values, several consecutive measurements will be done, and the mean of it will be taken into consideration.
- A distance sensor has been included in the subsystem. Since measurements vary with distance, they will be taken into consideration if only values are in the proper distance range indicated in the datasheet of the sensor. A small LED will also indicate that the measurement at that instant is going to be taken. After taking enough samples, the average value will be shown on the LCD screen.

Test & Results

Distance Test

Test Objective: The main objective is to measure error value depending on the distance between object and sensor.

Test Plan & Procedure: To be able to keep the only variable as distance, this test will be done without changing the ambient and the object will be placed directly towards the sensor to have the same angle of incidence at every trial. Then, a distance between 2-3 cm will be measured and the corresponding error value for that measurement is to be noted.

Test Result:

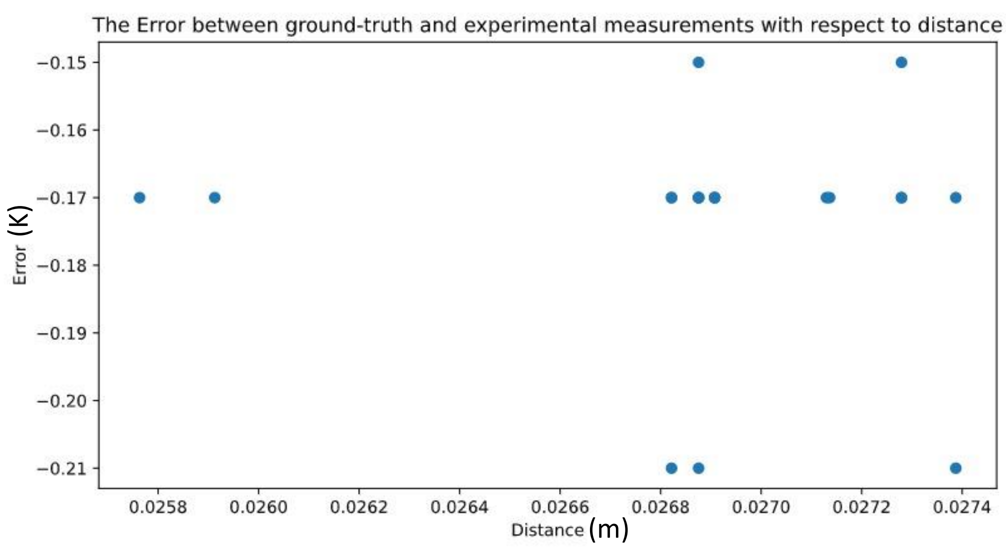

Figure 15. Scatter plot showing Measurement Error vs Distance.

In this figure, the x-axis shows distance, and the y-axis shows the error value in °C. Since a distance sensor is used, it works only between 2-3 cm range. Within this range, the temperature sensor takes samples. Each dot represents a trial. For all these dots, the performance requirement for the ±0.5°C error is satisfied. Moreover, the average of these samples will be used for the decider unit. So, the error will be minimized also with this approach.

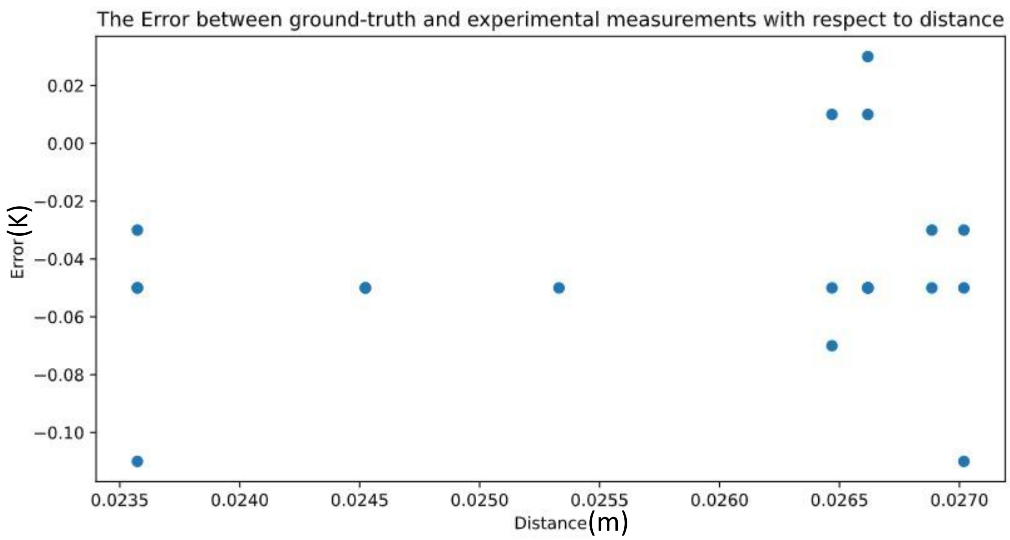

Figure 16. The second trial showing the same metrics in Figure 15.

Another test is applied, and the result is as expected again. In these two trials, 40 samples are tested, and the error is always below ±0.5°C.

The User Interface Subsystem

The main objective of this subsystem, composing of only one component being Nokia 5510 LCD Screen, is informing and guiding the user through the HealthPass' test procedures. The subsystem works in close collaboration with Smell Loss Detection Subsystem. Through this subsystem, the options regarding the sensed smell will be presented to the subject. An example screenshot can be seen in the below figure:

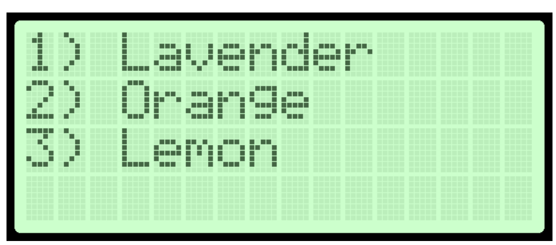

Figure 17. LCD Screen showing the smell options.

In addition to this operation, it also serves an informative and directive purpose. Just before the body temperature sensing procedure is conducted via the Fever Detection subsystem, the user will be guided on how to use the subsystem through the LCD screen. The body temperature sensed via the Fever Detection subsystem will be presented to the subject just to notify the subject. The relevant information and guidance are illustrated in the below figures:

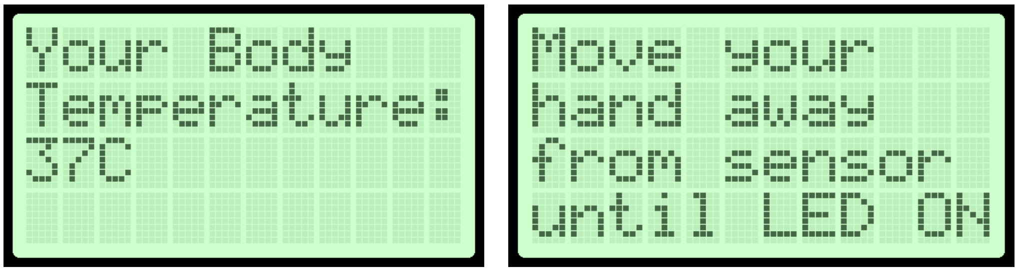

Figure 18. LCD Screen used for informative and directive purposes.

Functional Requirements

The subsystem should track which stage of HealthPass tests the subject is taking and provide the relevant information and guidance on time.

Performance Requirements

The system should refresh the information shown on the screen according to a suitable refresh rate depending on the subject's usage of HealthPass.

Test Procedure

Since this subsystem is a helper subsystem, it does not carry any critical measurement procedures, instead, it exists for the subject to interact with the Smell Loss Detection Subsystem and for the sake of increased user-friendliness.

Door Circuit Subsystem

The door circuit is the last step of our design. It is responsible for the connection between buildings and our Health Pass system. This subsystem aims to produce a signal to open the door. This door can be in various types. The main idea is to produce an unlock signal so that it works for different kinds of door designs. Input and output diagram of this subsystem can be shown as below:

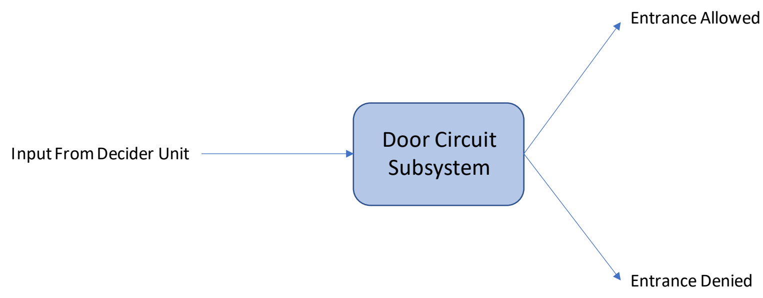

Figure 19. Door Circuit Subsystem Block Diagram

Functional Requirements

- This subsystem must be applicable to various door designs.
- The output signal should carry enough power to open the door.

Performance Requirements

- Input from the decider unit should be turned into a mechanical output instantaneously. There should not be any observable duration.

Solution

Decider Unit will produce an output about permission to the entrance. The door circuit will use this signal as input and if the entrance is authorized it will send a signal to a magnetic lock. A solenoid lock is selected for this purpose. It represents the affirmative signal from the decider unit in a mechanical way. Besides, it can be integrated into different door types. This lock is driven by a relay for switching operations between high and low. An adaptor is also used for drawing enough power. Selected solenoid lock and relay are shown below:

Figure 20. Magnetic Lock and Single Channel Relay Board [7,6]

Test & Result

The duration for producing a mechanical motion is tested. It is tested 30 times and traction was produced immediately in every trial. This result satisfies the performance requirements.

People Counter Subsystem

People Counter Subsystem is a crucial subsystem in counting the people inside the building. The overall system can keep track of the incoming people since it allows entrance or not. However, to detect people going out of the building, our system needs another subsystem mounted on the exit doors.

This subsystem consists of a microsound distance sensor. It measures the distance from the top of the door to the floor. When starting the system, this subsystem measures the height of the door. Then, it measures the distance continuously. Whenever this distance is lower than the door height, it means that someone is going out through the door. After the measured distance is equal to the door height again, the People Counter Subsystem sends a signal to the main circuit indicating the number of the people inside the building is decreased.

Requirements

In this section, the requirements of the subsystem will be discussed in detail.

Functional Requirements

People Counter Subsystem should detect the outgoing objects.

People Counter Subsystem should send the information to the Decider Unit.

Performance Requirements

The error of not detecting a person should be less than 1%.

The accuracy of differentiating two consecutive people should be over 99%.

Solution

The solution consists of an HC-24 Distance Sensor. This sensor will be mounted on top of the door. At the start of the HealthPass system, the first measurement will record the height of the door. After that, it will continuously measure the distance from the top of the door to the floor. Whenever an object passes through the door, this measurement goes below the door height. Until the distance value is restored to the door height, it is considered as only one object is passing through the door. When the measurement is equal to the door height, a signal is sent to the decider unit, indicating a person has exited the building. Then, the Decider Unit decreases the number of people inside the building.

Test Results (Scenario)

In the test scenario, a person passed in front of the sensor 100 times in both ways. The number of people in the building started as 120. After the test, the number of people remaining in the building is expected to be 20.

In 100 trials, there were no miscounting one person as two or missing a passage. This result suits the performance requirements.

Decider Subsystem

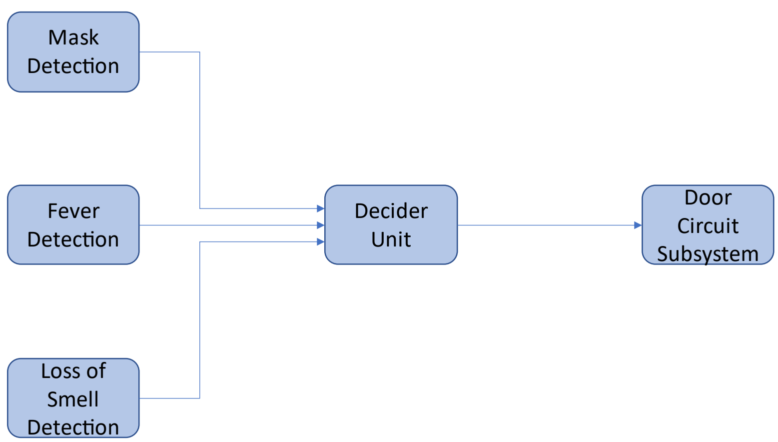

Figure 21. The Decider Subsystem Block Diagram

The decider subsystem will be responsible for augmenting the information coming from other subsystems, namely the face mask detection subsystem, fever detection subsystem, loss of smell detection subsystem, and people counter subsystem. Moreover, this subsystem will be responsible for making a decision of giving permission to the person or not. The decision will be made according to some prespecified conditions on the measurements taken from tests such as the temperature data obtained from the fever detection subsystem. The symptoms of severe COVID-19 cases include body temperatures above 38°C, and some cases may include loss of smell. Taking this information into consideration, we may set some reasonable conditions for these measurements. Such conditions are given in Table 6.

Table 6: Decision Mechanism when the number of people inside is under a threshold.

	Body Temperature<37.5°C	37.5°C <Body Temperature<38°C	38°C <Body Temperature
Improper Mask	No entrance allowed	No entrance allowed	No entrance allowed
Proper Mask & No Loss of Smell	Entrance allowed	Entrance allowed	No entrance allowed
Proper Mask & Loss of Smell	Entrance allowed	No entrance allowed	No entrance allowed

The decision mechanism will also check the number of people inside the building. Each building will have an adjustable threshold representing a number above which the social distancing cannot be maintained. When the number of people inside is under this threshold, the rules given in Table 6 are valid. When the number of people reaches this limit, the decider unit will not allow anybody to enter the building.

The decider subsystem will output a control signal to control the door circuitry according to the decision it made. The output will be binary, either unlock the door or not. Moreover, this subsystem will also output a signal whenever it permits a subject to enter the building. This output signal will be input to the people counter subsystem to keep track of the people inside the building.

This decider subsystem will be implemented on a Raspberry Pi 4B device in Python 3.

Performance Requirements for Decider Subsystem

- Once all the measurements required for this subsystem are done, the decision should be made in less than a second in order to satisfy the performance requirement that the overall process should take less than 30 seconds.

Since this subsystem is only responsible for evaluating the output of other subsystems with a lightweight algorithm, it does not have any further requirements.

Test Procedure

Since this subsystem is only responsible for combining outputs and reaching a conclusion based on them, there was no need for any tests for it.

POWER CONSUMPTION

Table 7. Current (Amp) drawn by Raspberry Pi 4B for different conditions. [3]

	Boot	Idle	Video-Playback	Stress
Average	0.7	0.6	0.78	1.2
Max	0.85	-	0.85	1.25

For Face Mask Detection Subsystem, the main component that dissipates power is the Pi Camera. It is supplied with 3.3V and it draws 120mA, leading to **396mW** power dissipation. [1]

For Fever Detection Subsystem, the IR temperature sensor draws 1.3mA with a 5V power supply, summing up to **6.5mW** while measuring. [2]

For People Counter Subsystem, we have used the ultrasound sensor HC-SR04 which is supplied with 5V, draws 15mA, resulting in **75mW**. [3]

For Decider Subsystem, there is no extra component that draws current other than Raspberry Pi itself. Except for Face Mask Detection Subsystem, the Raspberry Pi does not run using full capacity. Therefore, it is safe to consider the Raspberry device in IDLE, dissipating 3W. [4]

For User Interface Subsystem, the main component is Nokia 5510 LCD Screen. There are two possible usages of LCD Screen, with backlight and without backlight. For the former case, it draws 50mA under 3.3V Supply, resulting in **165mW**. For the latter case, it dissipates an extra **400mW**, drawing 80mA current with a 5V supply. [5]

Lastly, for Door Circuit Subsystem, the magnetic lock 12V 0.5A, 89mA 5V on average for the relay, adding up to **6.445W**. [6]

Table 8 illustrating the average power consumption of the overall system is presented below. On average the whole system dissipated 4.4 Watts. This amount of power will be directly supplied from the city's electrical grid.

Table 8. Power Consumption of Subsystems

	Time(sec)	Energy(mJ)	Power(mW)
Face Mask Detection Subsystem	10	3960	396
Fever Detection Subsystem	3	19.5	6.5
People Counter Subsystem	30	2250	75
Raspberry Pi	30	90000	3000
User Interface Subsystem	30	16950	565
Door Circuit Subsystem	3	19335	6445
Total	30	132514.5	4417.15

COST ANALYSIS

Table 9. The cost of different subsystems

Component	Unit Price ($)	Quantity	**Total Price ($)**
Raspberry Pi 4B 4GB	49	1	49
Pi Camera Module	9.5	1	9.5
MLX90614 IR Temperature Sensor	10	1	10
SD Card 32 GB	5.5	1	5.5
HC-SR04	1	2	2
Nokia LCD Display Screen	10	1	10
Membrane Button	1.13	6	6.78
Magnetic Lock	5.5	1	5.5
Single Channel Relay Board	0.7	1	0.7
Foottons	1.15	3	3.45
Disinfectant Stands	15	2	30
		Grand Total	**132.45**

DELIVERABLES

The customer is going to be delivered the items in the list below:

Raspberry Pi 4B 4GB

- Raspberry Pi Camera Module
- IR Thermal Sensor
- Distance Sensor
- Disinfectant Stand
- MongoDB Compass Software
- LCD Display
- Foottons
- Door-Lock
- HealthPass Software
- MongoDB Credentials

In addition to these, a guide explaining the usage of the system will be provided in both hard and soft copy. 2 years of warranty will be provided. The system will receive maintenance and updates for 2 years.

CONCLUSION

As AlgoDetectium, we aimed to design a system called HealthPass to help fight against COVID19. Apart from the studies on vaccines, the main aim is to decrease the rate of infection, for which we need to detect a person who has the virus as fast as possible, and we need to stay at home. Despite taking all the possible precautions, life goes on a certain pace which requires more attention as we have to keep social distancing. Our main focus while working on this project is publicly used buildings which are likely to be crowded. We want to design a system which detects the infected person according to certain symptoms recognized by the World Health Organization, which are loss of smell and high body temperature. We also want to make sure that whoever comes into the building wears a face mask which covers their nose and mouth properly. Another factor we consider is that the building does not get crowded in order not to violate the rules of social distancing. For that, we count the number of people inside the building and make sure that the population of the building does not become too high.

There are several subsystems included in Health Pass with different roles. According to our design, the subject who wants to enter the building needs the follow certain steps to be permitted. At first, we detect the loss of smell which is accepted to be a symptom of the virus by the World Health Organization. Some disinfectant with a certain scent that is well-known and distinguishable is presented to the subject. We expect them to smell it and choose which odor that the smell belongs from several options. Then, we measure their body temperature which is also a known symptom. The information that those subsystems generate is utilized to decide whether the subject shows symptoms of COVID-19 to be denied entrance.

As it is forced by the law, everyone must wear face masks in public places. Our system detects whether the subject wears a mask which is covering their nose and mouth. We use a camera and a detection model, trained by many samples, takes the visual input to make the decision. The electronic lock that keeps the door closed is not opened until the person wears their mask properly. Also, with the help of the camera, we count the people entering the building. At the same time, a distance sensor which is located at the exit of the building counts the people who leave. By doing so, we keep the knowledge of the number of people inside. That is necessary because if the building becomes too crowded, the rules of social distancing may be violated.

Although the first months of the pandemic caused anxiety, which led to a will to stay at home and avoid crowded places, the necessity to proceed our jobs and the need to interact with other people started to overcome this anxiety. HealthPass is a system that helps people to both stay safe and keep the daily life at a certain pace at the same time. We guarantee that only the people who do not show the symptoms of the coronavirus are allowed inside the publicly used buildings with the help of the detection and decision algorithms. We also make sure that people who are inside the building, had worn their masks properly while entering the building. Aside from these, HealthPass keeps counting the number of people inside the building so that the building is not too crowded. This will help people believe that obeying the rules of social distancing is easier if the building they are in uses HealthPass. With people feeling safer, negative effects of the pandemic can be eliminated a little bit more and quality of our lives will be increased.

HealthPass System is designed to implement some safety measures in case of a pandemic disease such as COVID-19. If HealthPass System becomes widespread, it can help to fight against pandemic diseases. Especially COVID-19 spreads mostly in closed and crowded buildings and when people do not wear their masks correctly. Our system detects possible patients, people without mask and does not allow them to enter the building. Moreover, it also limits the crowdedness of the building. As a result, having HealthPass System will potentially decrease the velocity of spread. On the other hand, the security was responsible from these safety measures before the HealthPass System was created. If the system becomes widespread, it can decrease the need of security personnel and take their jobs. The system can possibly have an effect over unemployment problem.

Although the pandemic will not be a part of our lives, it will still be able to serve a purpose to maintain the hygiene in public places due to the public concern remaining to exist. In addition to this, the system can also be utilized as an extra security layer in the form of an ID Recognition system to track the employees' entrance and exiting times from the building, eliminate the need for carrying ID cards, solving the problems may arise in the absence of these cards.

HealthPass system's power usage is considerably low when it is compared to modern everyday devices employed in the public buildings. In that manner, HealthPass is system is eco-friendly. On the other hand, this system will not probably be needed as much as it is needed in the present. Therefore, the pollution caused by this system is going to decrease over the years.

- Pictures of the entrants are stored locally on the Raspberry Pi and also on the database server. Privacy of the people is our priority. The saved data is not shared with third parties under no circumstances. The database of the entrants is cleared in monthly periods. Also, MongoDB, the database we are using, guarantees the safety of the data it is holding.

- There are cables carrying high power in this system. In any hazardous case, these cables with open ends may be harmful. In order to prevent this problem, we were careful not to leave any open end in cables.
- Raspberry Pi is vulnerable to physical damage, liquids, sunlight etc. In order to protect it, it should be hold in a closed area. Additionally, it should be kept in its protective case all the time.
- Touching surfaces such as buttons or LCD screen with hand may increase the risk of infection. Therefore, we came up with the foottons such that entrants do not need to touch anything with their hands.

REFERENCES

[1] “Raspberry Pi Zero 1.3 Power Usage with camera,” *RasPi.TV*, 20-May-2016. [Online]. Available: http://raspi.tv/2016/raspberry-pi-zero-1-3-power-usage-with-camera. [Accessed: 09-Apr-2021].

[2] Melexis Microelectronic Systems, “Single and Dual Zone Infra Red Thermometer,” MLX90614 datasheet, Feb. 2007 [Revised Oct. 2007].

[3] ETC, “HC-SR04 Ultrasonic Sensor,” HC-SR04 datasheet, Nov. 2014.

[4] Raspberrypi.org. (2019). FAQs - Raspberry Pi Documentation. [online] Available at: https://www.raspberrypi.org/documentation/faqs/ [Accessed 20 Dec. 2019].

[5] L. Ada, “Nokia 5110/3310 Monochrome LCD,” Adafruit Learning System. [Online]. Available: https://learn.adafruit.com/nokia-5110-3310-monochrome-lcd/power-requirements. [Accessed: 09-Apr-2021].

[6] SONGLERELAY, “Songle Relay,” SRD05VDCSLC datasheet.

[7] https://www.robitshop.com/urun/selenoid-kapi-kilit-seti-dc-12v

[8] https://www.raspberrypi.org/documentation/usage/gpio/

APPENDIX

HealthPass Surveillance System User Manual

HealthPass System

COVID-
19 brought need for health concerned precautions in public buildings such as wearing proper masks ,not having fever and having smelling ability. HealthPass is designed by AlgoDetectium as a Surveillance and Protective Measurement system that monitors the mentioned precautions and controls the access to enter the building.

This system includes a Raspberry Pi 4B microcontroller, an (MLX90614) contactless thermal sensor, 2 HC-SR04 Ultrasound Distance Sensor, a Raspberry Pi Camera Module, a Nokia 5110 LCD Screen, 5 Foottons, 2 Disinfectant Stand, a Protective Case, 2 5V Power Adapters, a Solenoid Controllable Lock Circuit, a database software.

Before Using for the first time

There are a number of steps to be followed before starting and using HealthPass System.

1. Database Configuration

First go to this website: https://www.mongodb.com/products/compass. Download and install Mongo DB Compass.

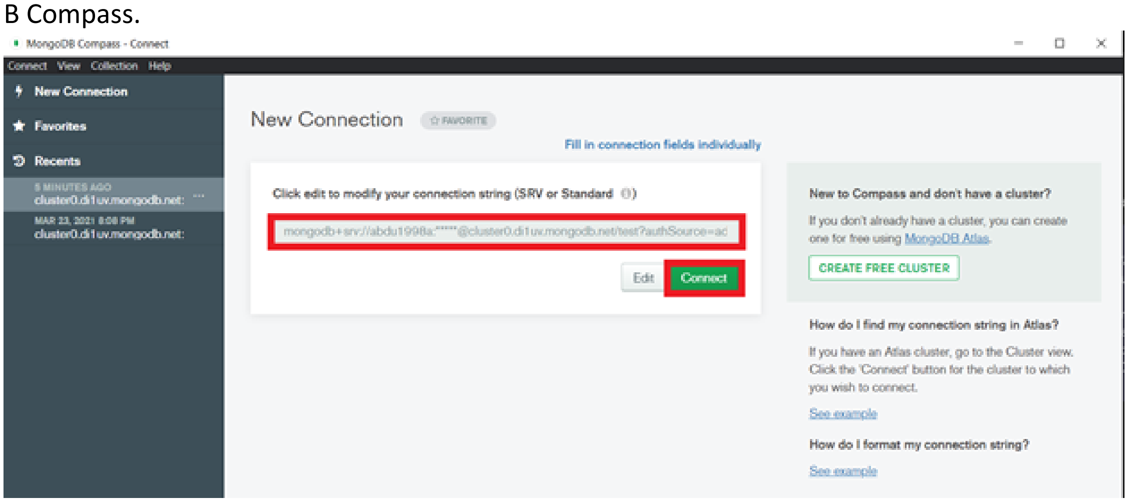

Figure 1. Mongo DB Compass Login Screen

Enter the unique code given to you in the red box in Figure 1.

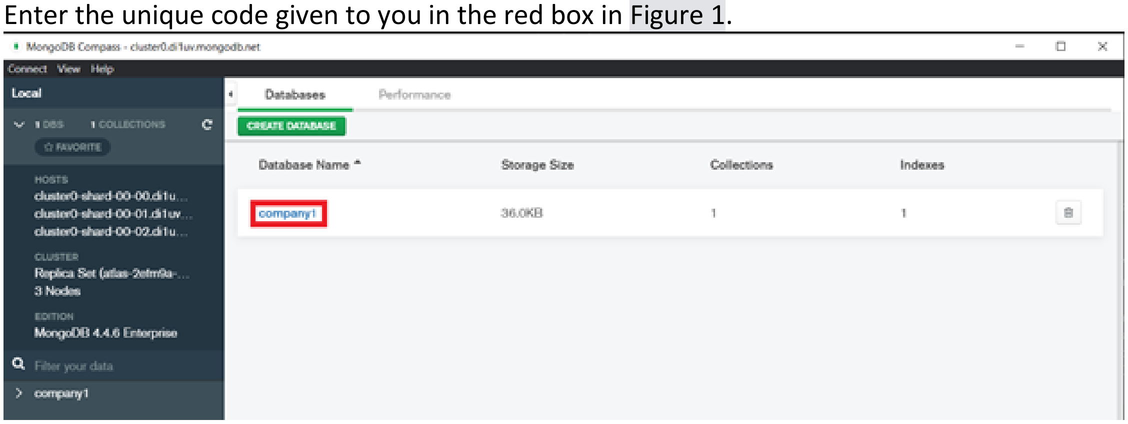

Figure 2. Database Interface

Click on your company name.

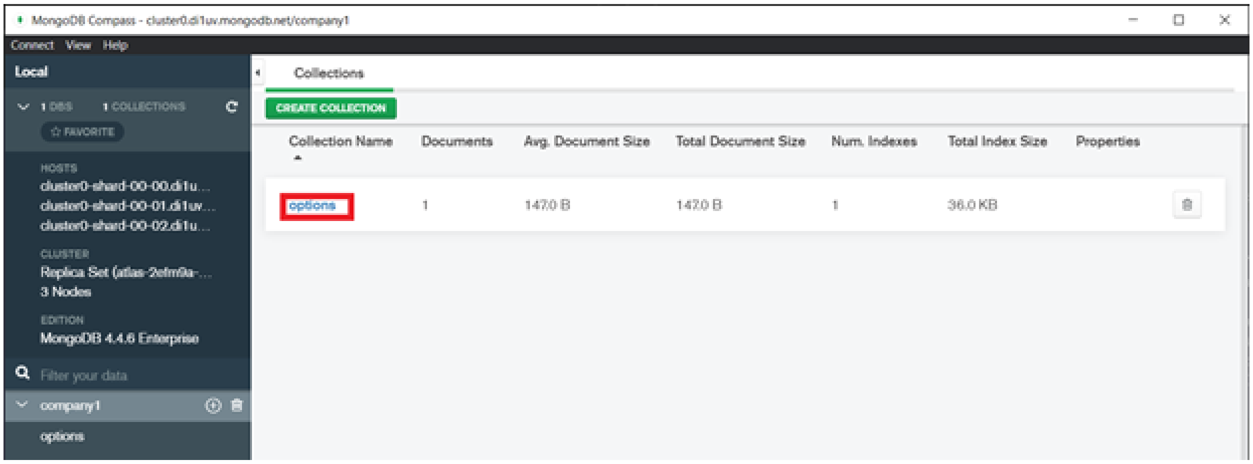

Figure 3. Collections Page

Click on options button.

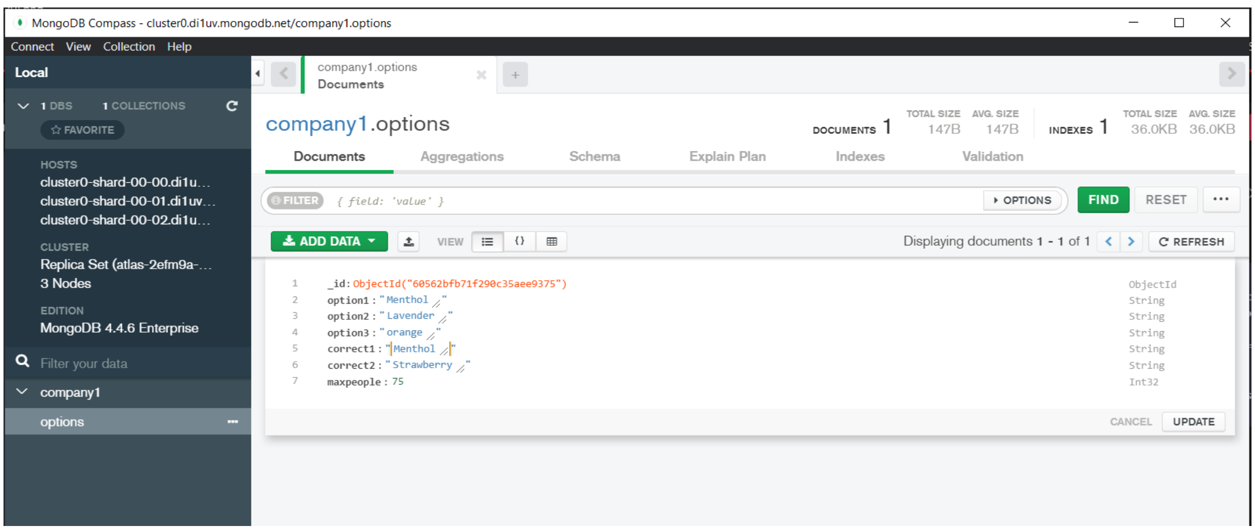

Figure 4. Options Interface

In Figure 4, options interface is shown. You can double click on the options to change their content. To save the changes, click on the update button.

2. Proper Setup

Before the first use, the system should be set properly.

- Its both power adapters should be plugged into a power outlet.
- The internet connection should be made either by connecting an ethernet cable or by providing the Wi-Fi credentials of the store to the technical stuff while installing the product.
- The environment should be properly illuminated so that the mask recognition software works as expected.
- The distance sensors for exit door should be seeing a smooth surface at a fixed distance and should be calibrated to count exiting people correctly.
- If there are multiple entrance or exit doors, each of the entrance and exit doors should have a HealthPass device, and each device should be configured with the same store name to count number of people inside correctly.

- The disinfectants used in this system should be chosen to have a distinct smell.
- If there are any problems after following these steps, please contact either directly to the seller or one of the technical supports.

3. First boot

The HealthPass system boots as soon as the main power adapter is connected. The booting procedure can take some time. When the screen has “Disinfect Your Hands!” printed as in Figure 5, the system is ready to operate.

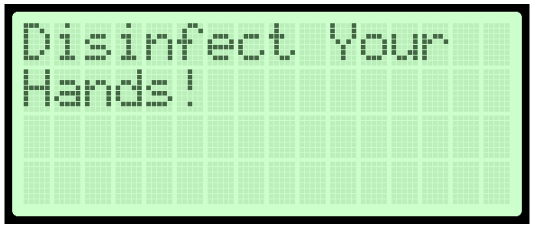

Figure 5: LCD Screen

Guidance for Entrants

Entrants should follow the instructions stated below. LCD screen in HealthPass System will guide entrants already. However, company can keep track of procedure as below.

- Enter the platform.
- Check the screen to see ‘Disinfect your hands!’ statement.
- Select a footton randomly and press it with your feet.
- Follow the screen to see correct number of corresponding odors which is diffused.
- Depending on the number, press the related footton.
- Check the screen if your answer is correct, even it is not, finish the test.
- Smell test is done, check the screen for fever test.
- Extend your hand to 2 or 3 cm front of temperature sensor.
- Show your palm or wrist towards temperature sensor.
- If LED is on, keep your hand there until LED is off for consecutive measurements.
- If not, move your hand back and forth slightly to be in appropriate range.
- After LED is off, measurement is done.
- Check the screen for your body temperature.
- Move on to face mask detection unit.
- Adjust your mask and move your face towards camera.
- Wait until measurement is done.
- Screen will also inform you about process.
- After face mask test is done, lock will open depending on your test results.
- If your test results are not risky, lock will open, and you can enter the building.
- If lock does not open after mask test, leave the platform, do not wait inside.
- If you fail, you can queue up again.

Safety Information

Raspberry Pi is a sensitive device in both electrical and physical manner. Its memory and the main chip should not be exposed to direct sunlight, wet surfaces, or any type of liquid. There should be enough ventilation in the environment since it can overheat and break-down.

High current on the lock relay and Pi Adapter. Beware and do not touch.

Raspberry Pi can heat up and be dangerous. Do not touch the chips on the board.

While operation do not touch the screen or any part of the system to prevent any risk of transmission of diseases.

Maintenance and Warranty

System does not need any human administration to perform. Therefore, user must not interfere with the Raspberry part of the HealthPass System. During any malfunctioning or error please contact the Warranty Service.

AlgoDetectium offers 10 years guarantee. Technical service includes updating the software, changing the foottons and screen that are broken.

Information on HealthPass

This section describes the rules the client and the users should follow to complete the HealthPass test successfully. The “user” defines the person who is going to be subjected to HealthPass tests and the “client” refers to the side who holds the administration of the HealthPass system in the following context.

1. The system becomes ready to use when “DISINFECT YOUR HANDS” written on the screen.
2. The users should show their bare hands to the temperature sensor to get the most accurate measurement.
3. The infrared temperature sensor takes measurements only in previously specified distances. The instances that it takes measurements are indicated by a red LED. Users should hold their hands still in front of the sensor while the LED is on.
4. People should not be allowed to exit the building in group, instead they should be directed to leave the building one by one to have the most accurate number of people inside, simultaneously maintaining the social distancing practices.
5. The disinfectant smells provided by the client should be easily distinguishable by a healthy user to avoid false negative entrance permission. In that sense, the client is strongly advised to use disinfectants having distinct smells.
6. The users are required to show their faces directly to the camera, as indicated by the signs inside the cabinet.
7. The smell loss detection test and mask classification tests have 20 seconds of waiting time for the user to specify what they smell via pressing the foottons and showing their faces to the camera. The tests that are not completed in this period will result in repetition of all tests from the beginning.
8. Once the users are allowed to enter the building, they have 7 seconds to enter the building, then the lock becomes activated again.