Fingerprint Based Check-In System



Interim Project Report

**Authors**: Bilal Nasir & Muhammad Amash Khan

**Course**: Technical Project (TPJ655)

**Program**: Computer Engineering Technology (ECT)

**Instructor**: Benjamin Shefler

**Date**: 17th of March 2023

Table of Contents:

[Executive Summary: 3](#_Toc131700931)

[Introduction: 4](#_Toc131700932)

[Project Functional Features: 5](#_Toc131700933)

[Product Specifications: 6](#_Toc131700934)

[Operating Instructions: 7](#_Toc131700935)

[Block Diagram: 8](#_Toc131700936)

[Software Diagram: 9](#_Toc131700937)

[Components: 10](#_Toc131700938)

[Theory of operation: 13](#_Toc131700939)

[Maintenance Requirements: 17](#_Toc131700940)

[Conclusion: 18](#_Toc131700941)

[Further Developments: 19](#_Toc131700942)

[Appendix: 20](#_Toc131700943)

[Schematics: 20](#_Toc131700944)

[Bill of Material: 21](#_Toc131700945)

[Password/Usernames used: 22](#_Toc131700946)

[References: 23](#_Toc131700947)

[Contact Information: 24](#_Toc131700948)

# Executive Summary:

The process of checking in can sometimes be tedious and inefficient, quicker methods do exist in places like airports, but are never implemented on a wide scale in local businesses. To combat this, we made a finger-print based check-in system for everyday businesses mainly targeted towards membership based ones like gyms and spas. This system uses a fingerprint scanner or manually entering your credentials to unlock the door lock with all of it being controlled by a RaspberryPi making it relatively inexpensive. If the user’s fingerprint or their user ID and password match, the door unlocks allowing them to enter; while if it doesn’t, they get a message to try again. Not only does this system automate the process of checking-in and make it faster, but it also adds a new layer of security for these business using biometrics making them more secure.

# Introduction:

In many organisations where check in is required, it can usually be very slow and inefficient. There are pre-existing ways to combat this, for instance airline online check in systems which stream line the process. However, this project aims at streamlining the check in system for membership-based clubs like gyms, spas, and etc. Instead of the old-time consuming ways, where a couple of attendants were required at the front desk to check members in; this method would allow them to check them selves in very quickly, saving them time which they can use for the activity they are there for. This project will not only benefit the client but instead would also help reduce operational costs for the business that adapt it as they will not need to have receptionist solely for the purpose of checking customers in; they’ll just need one to be on call whenever a customer faces issues, and for the remaining time they can take part in other duties that concern the business. Moreover, it is very easy to implement and use too as once the system is installed, all it will need is to be connected to a screen and then it will be ready to authenticate customers either by a fingerprint scanner, or by their credentials.

# Project Functional Features:

* Fingerprint scanner used to authenticate members. The R307 finger print scanner used in this project waits for a finger to be placed on the scanner, once done, if authenticated, it activates the solenoid to unlock the door. The scanner is also used during the enrollment process to store alongside the user ID.
* Web based application with the GUI that the user will use to interact with the application. Uses Flask to run a web application that will give the user the option to either log in with just the credentials, or to create a new account.
* The ability to sign in with credentials or creating a new account. As mentioned earlier, this will be included in the web application’s GUI.
* Door lock that unlocks when authenticated. A 12V solenoid that is activated when the fingerprint is authenticated causing it to contract, in turn unlocking the door.
* LCD that displays interim messages during the authentication and registering parts of the process. Connects to the 5V of the pi to get power while the SDA and SCL pins connect to their corresponding ones on the Pi.

**Optional Feature (If the main features can be accomplished in time.):**

* Proximity sensor on the inside so that anyone leaving can just leave without having to authenticate themselves.

# Product Specifications:

**RaspberryPI 4B:**

Source Voltage: 5.1V

Max power: 15.3 W

Signal Output: 5V and 3.3V

Dimensions: 85mm X 56 mm

**R307 fingerprint scanner:**

Input Voltage: 5V

Interface: UART

Current consumption: 50mA

Dimensions: 10mm X 10mm

**LCD:**

Input Voltage: 5V

Current consumption: 24mA

Dimensions: 92mm X 90mm X 27 mm

**5V Relay:**

Supply Voltage: 3.75V to 6V

Dimension: 12.9 x 8.41 x 3.3 cm

**12V solenoid:**

Input Voltage: 12V

Dimensions: 10mm X 10mm X 10mm

**Final enclosure:**

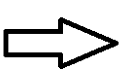
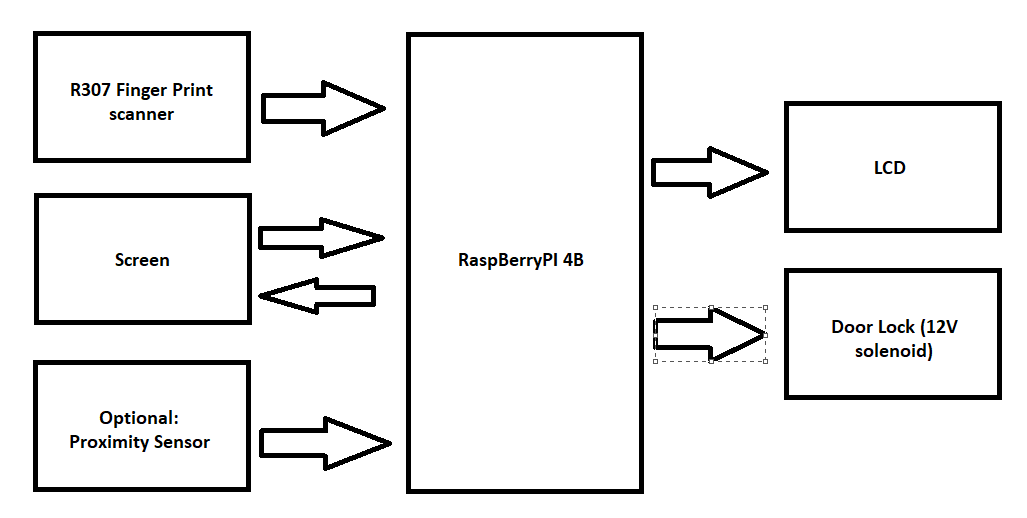
22.86cm X 17.78cm X 13.97cm

# Operating Instructions:

1. User chooses between the options presented in the web application GUI. They either log in with their credentials, unlock the lock by scanning their fingerprint by selecting “Authenticate”, or create a new account.
2. If the first two options are selected, and the system authenticates the user, the door is unlocked for 5 seconds (this value can be modified to the needs of the customer) allowing the user to open the door and go in
3. If the third option is selected, a new screen appears asking the user to enter their Email ID, User and Password after which they are prompted to enroll their fingerprint by following the instructions on the screen.
4. In order to view the GUI, the web application will be accessed by entering the provided IP address of the raspberry Pi into the browser.

For this purpose, any screen can be used, either a monitor with a keyboard or mouse, or for a more modern look, an iPad or tablet which will allow for touch screen use while looking sleek and being efficient.

# Block Diagram:

****

# Software Diagram:

**User**

**Options**

**Enter Credentials**

**Scan Finger**

**Authorized**

**YES**

**NO**

**Door Unlocked**

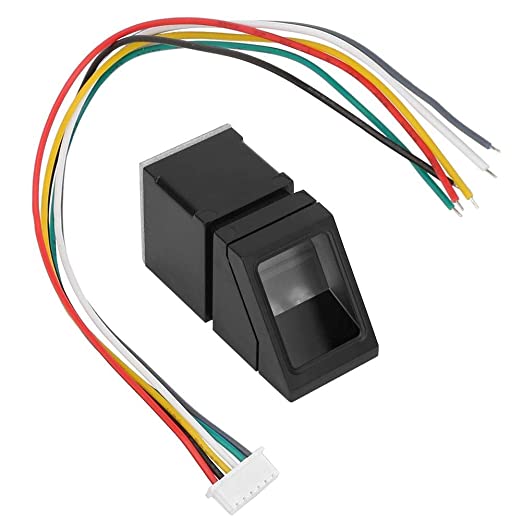
**Try Again**

# Components:

Raspberry Pi 4b 4GB RAM:

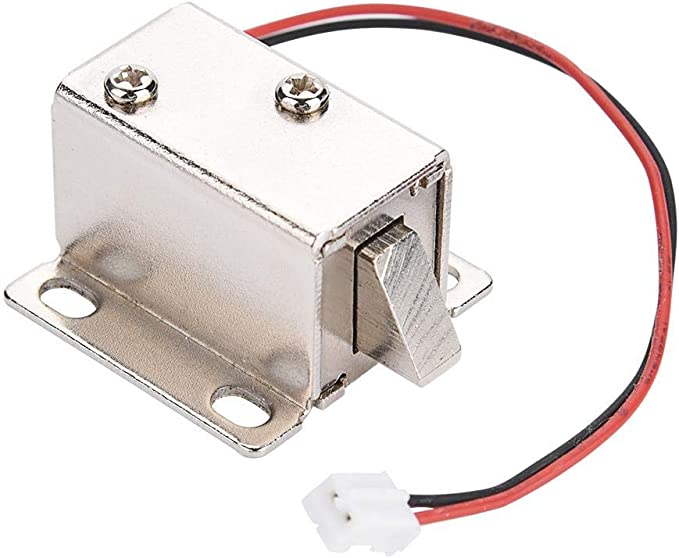
The 4 GB RAM version of the raspberryPi retails for around 75 CAD before tax, but that is the board only. The 4GB version is the one that is used in this project; it has a 64-bit quadcore processor, 2.4/5.0 GHz dual band wireless LAN which is preferred as the 2.4 provides a greater range while the 5.0 gives a better performance. This was used by installing xrdp onto the pi which allowed remote access through the Microsoft remote desktop app which allowed for a more efficient use through the GUI.

R307 Fingerprint Scanner:

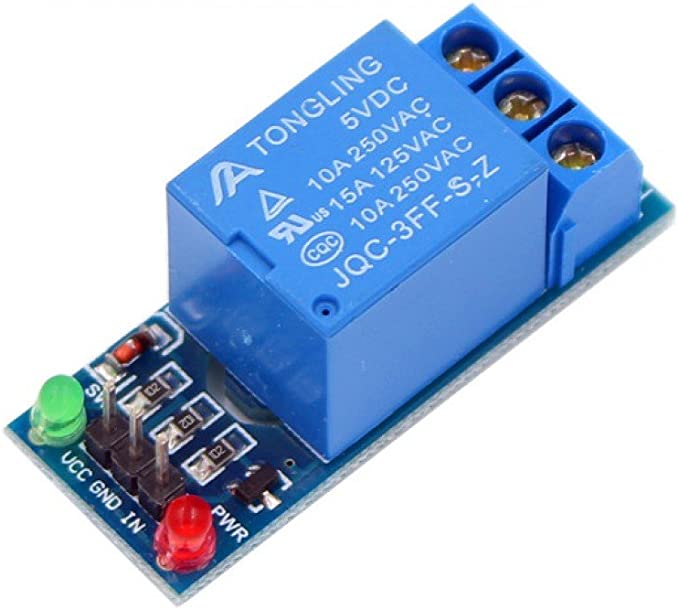


This is the fingerprint scanner used in this project. It was chosen as up-to 1000 fingerprints can be stored on board which are more than enough for the purpose of this product. It is also relatively accurate as it is an optical fingerprint scanner.

12 V DC Solenoid:

A straightforward DC 12volts solenoid used as the lock for this system. When this is activated using a relay, it contracts causing the door to unlock, and after the prescribed time period as ended, it deactivates resulting in the door to lock again.

5V Relay:



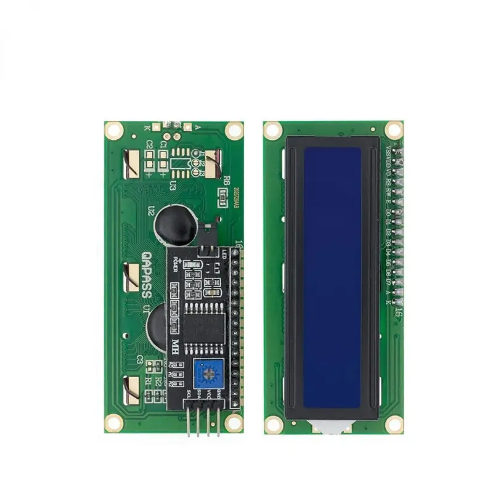
Relatively unexpensive relay which allows the solenoid to run properly, the vcc of the relay connects to the 3.3V pin of the raspberryPi while the IN connects to any GPIO pin, in this case pin number 18 while the relay also connects to the positive of the solenoid and 12V power source.

Screens:

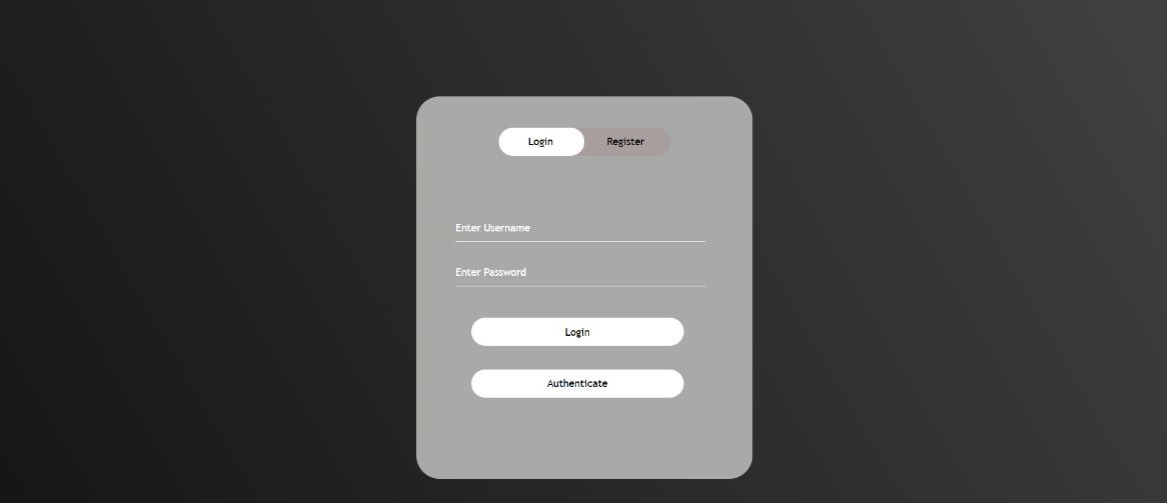


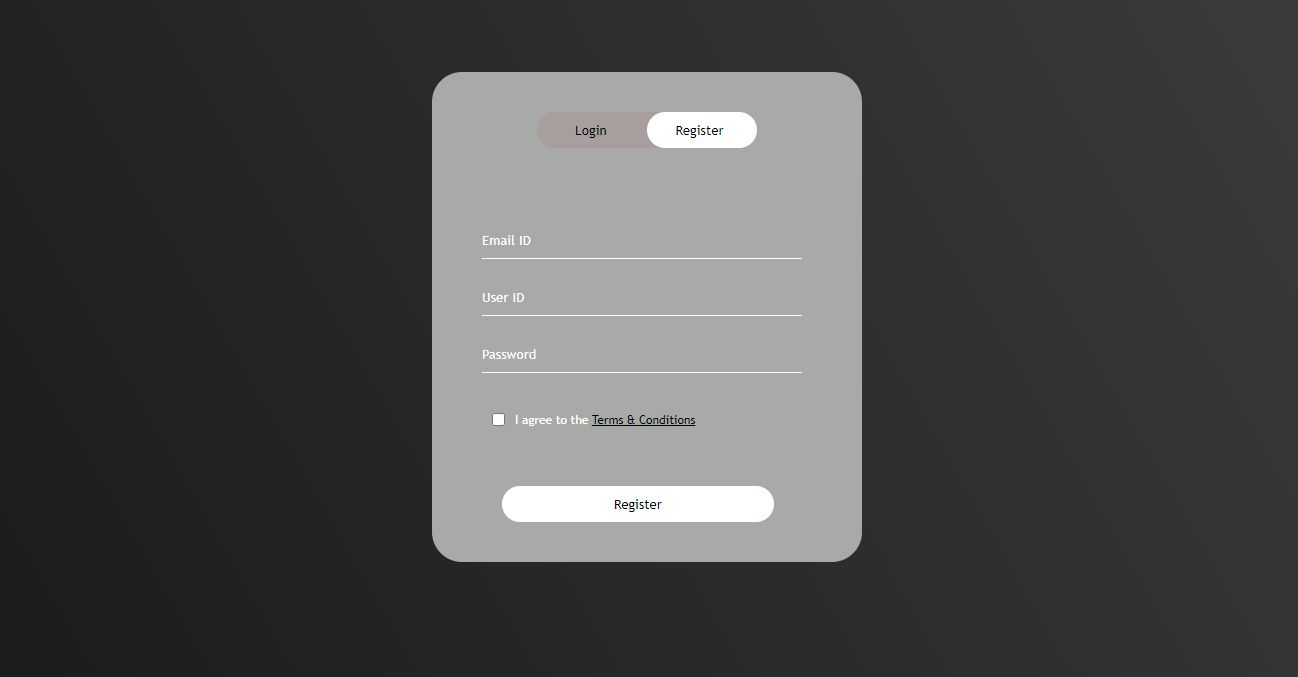
Any screen that is available at the moment can be used, as the raspberryPi it self can connect to a monitor using an HDMI and display the website.

However, for this demo a Samsung Galaxy s8 Tablet is used and the web application is displayed via the IP address of the raspberryPi.

The LCD is used to display interim messages during the authentication and enrollment segments of the process. When clicked on Authenticate the LCD shows a message prompting the user to place their finger. Where else while registering it shows messages requesting the placement of finger, or to show if it was a success and if it has been templated, or if it couldn’t it indicates an error with the message. The VCC connects to 5V while SDA and SDL connect to the corresponding pins on the Pi respectively.

GUI:



This is the home page that the user is greeted with when they look at the screen. They are prompted to either login with their credentials or select the “Authenticate” option to authenticate with their fingerprints. Once authenticated it will display a welcome message with their name. Or they can opt to create a new account by clicking on the register button, after which they will be prompted to create a new account by entering an email ID, user ID, password, and accepting the terms and conditions, which can be customised according to the policies of the establishments that adapt the system.

# Theory of operation:

For this project, the micro controller used is a RaspberryPI 4b as mentioned earlier. It was chosen because of how efficient it is, the number of GPIO pins for input and output, it’s portability and the fact that it can run on either Wi-Fi or ethernet, the last point holding a great deal of importance as many locations where the customers would like to implement it may not have access to ethernet connection, so a possibility of using Wi-Fi is key. The number of GPIO pins was a great factor in the decision-making process as well as all components are connected to the Pi using the pin layout.

The main component of this project is the R307 fingerprint scanner, which is an optical scanner which works on the principle of total internal reflection; meaning when the finger is placed on the sensor the light reflects off the ridges of the finger print to document it. The R307 is a great choice as it can store up to a 1000 finger prints on its local memory while being relatively quick for its price range as it can acquire an image in less than 0.5 seconds which is important as the entire purpose of this project is to stream line the process of checking in. The fingerprint is quite secure too as the False Recognition rate is at less than 1% while the average search time for the prints is less than 1 second.

As for the connections, the VCC and ground pins of the scanner connect to the 5V and ground GPIO pins of the Pi respectively, while the TX of the scanner connects to GPIO pin 15 (RX) of the Pi and the RX of the scanner connects to GPIO pin 14 (TX) of the Pi.

The LCD connects to the Pi using GPIO pin 2 (SDA) for the SDA pin on the LCD and pin 3 (SCL) for the SCL pin, while the ground an VCC connect to the 5V and GND respectively. It displays messages after either authenticate or register is pressed on the web application; it is mainly there to guide the users to make sure they follow the right steps.

The 12V selonoid connects to the 5V relay which connects to the ground and 3.3V using the GND and VCC pins respectively, while the N1 one connects to GPIO18.

Finally, the last screen can be any device as the web application uses flask to create a server and can be displayed on any browser using the IP Address of the Pi, for this project a Samsung Galaxy Tab 8 is being used as it was available and can give a touch interface to the website making it feel more modern and sleeker.

As for the software, the RaspberryPi uses python3 code to control all the components, using libraries like Flask, Pyfingerprint, LCD, GPIO, and time along with many others to simplify the code to a great extent. Flask is used to create the web server while html, CSS, and JavaScript are used as the languages for the code for the actual web application.

The program uses a SQL database to store the created accounts so they can be looked up and matched with to authenticate the user.

Some important functions in the python code include:

**def enroll\_finger(location):**

This function as the name suggests is used for enrolling the fingerprint. It is called after register is hit on the web application, prompting the user to enroll their fingerprint along with the account they are creating. It works by taking two images of the fingerprint by asking the user to place the finger twice; it then templates it and stores it inside the R307 fingerprint scanner itself.

**def isT():**

isT which stands for is True is called in the def login(): when the fingerprint is authenticated. Once it is called, it prints a message on the LCD screen signalling success and unlocks the door by retracting the solenoid.

**def isF():**

isF which stands for is False is called in the def login(): when the fingerprint can not be authenticated. Once it is called, it prompts the LCD to display a message requesting the user to retry.

Both of these are called in def login(): using the following IF statement:

if ( positionNumber == -1 ):

isF()

else:

isT()

Over here, if the fingerprint does not belong to any position, which is showed by “positionNumber == -1” it calls the isF function. But if it does not equal to -1, it calls the isT function.

This works as the fingerprints are stored in positions onboard the scanner itself (each finger print is assigned a position), and anything below 0 does not exist, so if it can not be authenticated, the fingerprint automatically is assigned to the position of negative one triggering the isF function.

If it is authenticated, the scanner locates which position its stored at hence it does not equal to negative one, in turn triggering the isT function.

The positions also relate to the users in the SQL Database allowing the system to authenticate the specific user while also enabling the method of authenticating with credentials.

# Maintenance Requirements:

The maintenance required for this product isn’t high at all. All the firm needs is for their employees to be familiar with the GPIO pin layout on the raspberryPi which can be done by just a simple google search so in case of any wire being damaged, it can be easily replaced free of cost.

Other than that, they should make sure accurate power supplies are being used, while monitoring the temperatures, even though passive cooling is provided by the raspberryPi, it is better to be vigilant as it can overheat if left on for a long period of time. Would be recommended to turn the system off during off hours of the firm. It is also recommended that someone familiar with the raspberry Pi does a full system check every 6 months to ensure everything is healthy, this can be done either free of cost if a tech savvy employee works at the firm or with a small fee of hiring either us or other experts.

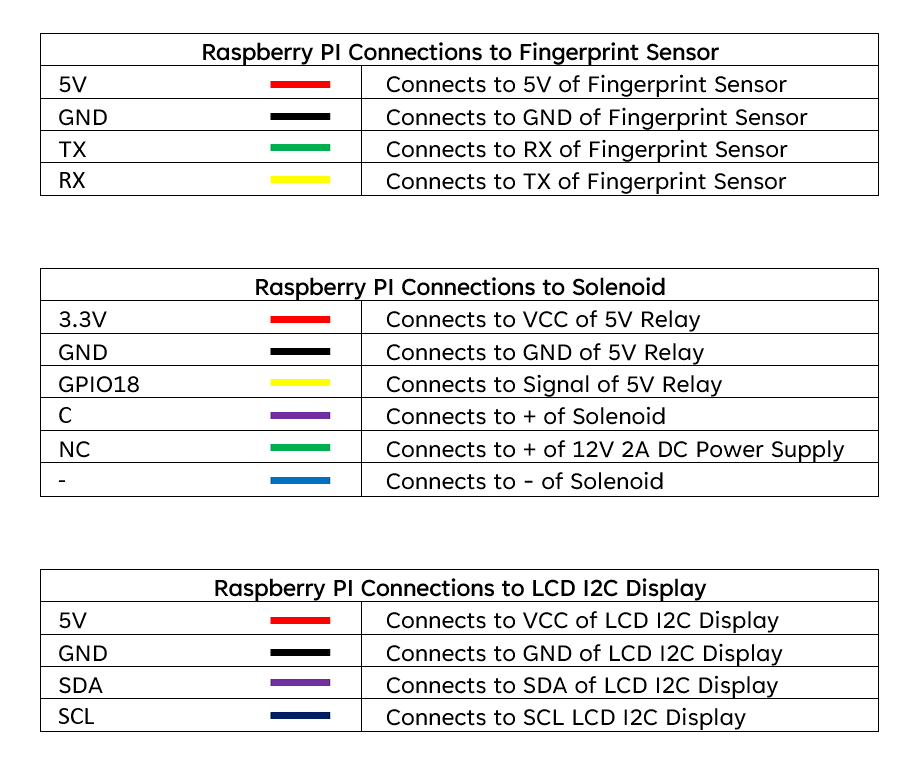
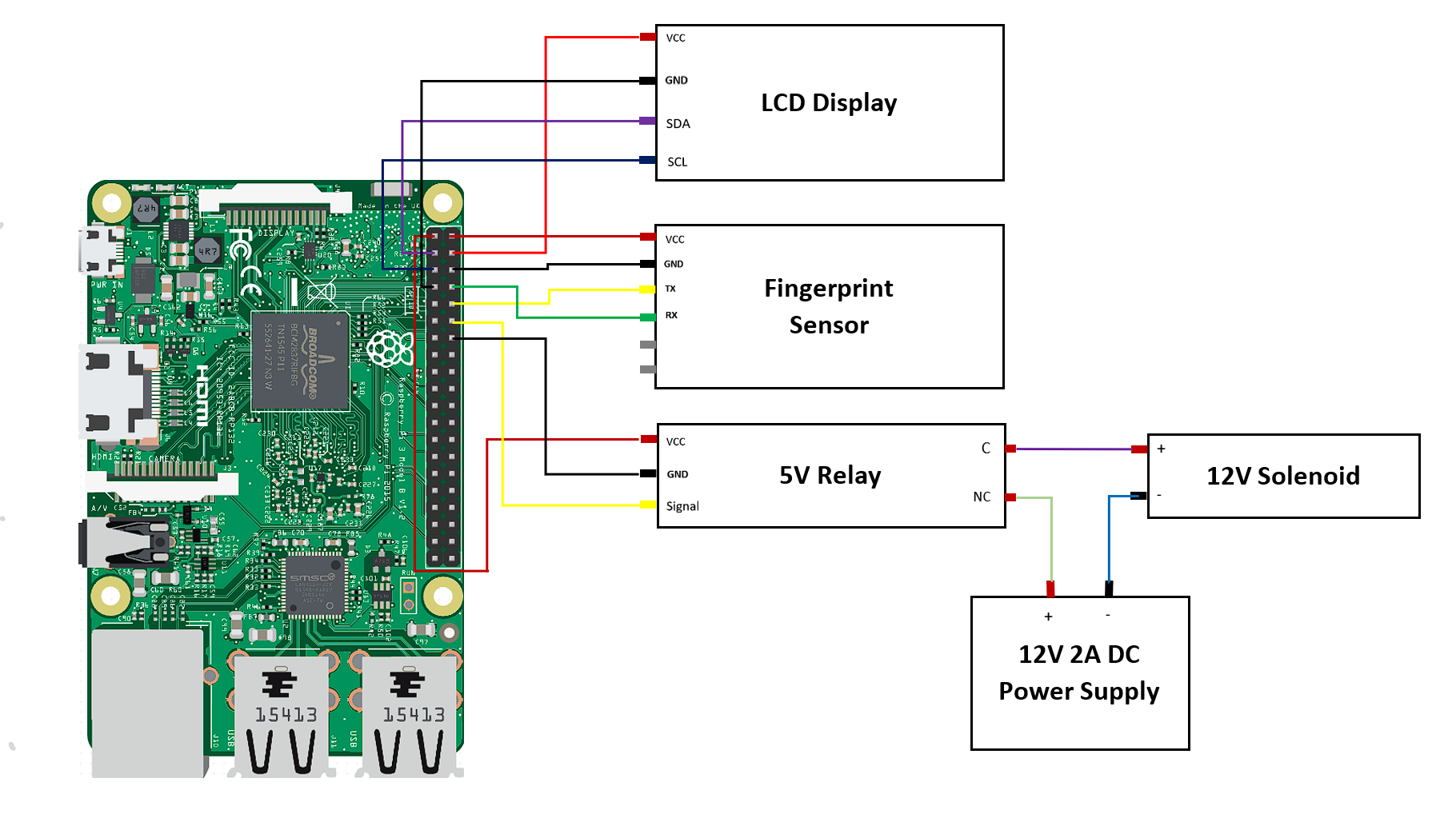
# Conclusion:

In conclusion, not only does this project make the checking in process more stream lined, but it also increases security as it uses biometrics while the false recognition rate is less than 0.1 percent. It is also relatively inexpensive along with requiring minimal maintenance which makes it an attractive option for business attempting in decreasing their operational costs, as the implementation of this would require fewer employees on the front desk. The project full fills the requirements outlined in the initial report only leaving out the optional feature of the proximity sensor, but that is made up for by including an extra LCD, which helps provide a cleaner look for the graphical user interface. The project also runs as intended and without any errors. It is also compact and can be implemented anywhere while connected to a power source. Moreover, the ability to display the GUI anywhere allows for the web application to be displayed on any screen simply using a browser, which can allow the customer to use it as a touch screen (as in the prototype) or with a keyboard. If implemented, the device would be a success and reduce the workload for the employees while saving time for the customers.

# Further Developments:

For further developing the project, the proximity sensor can be implemented which would allow customers to just walk up to the door unlocking it, this would save them form the hassle of authenticating while leaving the establishment. A security camera can also be added along the door so it can provide live feed to the employees to make sure no body tries anything suspicious. Temperature sensors should also be added along with moisture sensors to make sure the environment for the pi is optimal and that the pi will not be damaged. A PCB can also be used to make the system look neater and even more compact. An account locking mechanism can also be added where after a consecutive amount of failed log-in attempts, the account freezes and cannot log-in without the assistance of an employee making the system more secure. Finally, with time, more capital can be invested into the project allowing for better quality components being implemented which would improve the over all quality of the system.

# Appendix:

Schematics:

## Bill of Material:

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Name** | **Link** | **Cost (CAD)** |
| 1 | Raspberry Pi 4 B | https://www.pishop.ca/product/raspberry-pi-4-model-b-4gb/?src=raspberrypi | $76.95 |
| 2 | R307 Fingerprint Scanner | https://www.amazon.ca/gp/product/B07SFYDM4M/ref=ox\_sc\_act\_title\_4?smid=AWD4L551SADGP&psc=1 | $22.03 |
| 3 | WITMOTION UART | https://www.amazon.ca/gp/product/B07TFFCDH4/ref=ox\_sc\_act\_title\_3?smid=AHXBXFML1U6DK&psc=1 | $23.80 |
| 4 | SC 12V Solenoid | https://www.amazon.ca/gp/product/B07XD31J99/ref=ox\_sc\_act\_title\_2?smid=A1GHAO7RQQJPUO&psc=1 | $14.76 |
| 5 | Garosa HC-SR04 | https://www.amazon.ca/gp/product/B07QM7VQ3Z/ref=ox\_sc\_act\_title\_1?smid=A22TNJTMHYRDX4&psc=1 | $8.42 |
| 6 | 5V Relay Module | https://www.amazon.ca/dp/B07KXKMLLW?psc=1&ref=ppx\_yo2ov\_dt\_b\_product\_details | $14.50 |
| 7 | LCD | <https://www.amazon.ca/DSD-TECH-SH-D1602-Interface-Raspberry/dp/B07T8S3P1M/ref=asc_df_B07T8S3P1M/?tag=googleshopc0c-20&linkCode=df0&hvadid=292924016385&hvpos=&hvnetw=g&hvrand=11717179044295344286&hvpone=&hvptwo=&hvqmt=&hvdev=c&hvdvcmdl=&hvlocint=&hvlocphy=9000966&hvtargid=pla-813299064470&psc=1> | $16.99 |
|  | Total Before Tax |  | $177.45 |

## Password/Usernames used:

**For Raspberry Pi:**

Host: masked.local

User: mak

Password: Amk123456

**In SQL database:**

Email: amashkhan@saqib.com

User: mash

Password: Mash123

Email: bilalnasir2712@gmail.com

User: Owner

Password: Pass123

Email: billiuddin357@gmail.com

User: Bilal

Password: BM123

Email: amashkhan@saqib.com

User: mash

Password: Mash123

# References:

Anon. (-). Raspberry Pi 4 4GB. CanaKit.

<https://www.canakit.com/raspberry-pi-4-4gb.html>

Heidenreich, M (2020, 25 Nov). How to Use the LCD1602 I2C Display with Raspberry Pi (Python Tutorial with Multi-Threading). YouTube.

<https://www.youtube.com/watch?v=DHbLBTRpTWM>

Rovai, M (2018, 17 Mar). Python WebServer With Flask & Raspberry Pi. Medium.

<https://towardsdatascience.com/python-webserver-with-flask-and-raspberry-pi-398423cc6f5d>

Mohanan, V. (2021, 11 May). Interfacing R307 Optical Fingerprint Scanner with Arduino Boards for Biometric Authentication. Circuitstate.

<https://circuitstate.com/tutorials/interfacing-r307-optical-fingerprint-scanner-with-arduino-boards-for-biometric-authentication/#r307-specifications>

Neupane, A (2021, 12 Mar). Python Flask Authentication Tutorial - Learn Flask Login. YouTube.

<https://www.youtube.com/watch?v=71EU8gnZqZQ&t=854s>

Tim. (2023, 15 Feb). Raspberry Pi Imager - How to Use. Core Electronics.

<https://core-electronics.com.au/guides/raspberry-pi/raspberry-pi-imager/>

Tim. (2023, 16 Feb). Controlling a Solenoid with Raspberry Pi and a Relay. Core Electronics.

<https://core-electronics.com.au/guides/solenoid-control-with-raspberry-pi-relay/>