

Fingerprint Authentication Project

**Name: Ben Frost**

**Student ID: S246228**

**Submission Date: 4th June 2025**

**Supervisor: Dr Kakia Chatsiou**

Declaration

All content quoted in this paper from other authors publications have been specifically acknowledged by clear cross-referencing to author, work and page(s). Any illustrations that are not the work of the author of this paper have been used with the explicit permission of the originator and are specifically acknowledged. I understand that failure to do this could be considered plagiarism and will be considered grounds for failure in this project and the degree as a whole.

Ben Frost

Abstract

This project aims to explore the benefits and possibilities of having biometric input as the log in method to bank accounts. It aims to prove that biometrics are a more secure method than the current system and makes it socially easier to gain access to personal bank accounts.

[The abstract briefly summarises your project. It should provide a concise description of the need for such a project, what this project intends to achieve (aims) and the results. The abstract is usually the last chapter of the report to be completed once you have an overview of the whole project. The abstract should be 6 to 12 sentences. For examples, consult a journal article to see how an abstract is constructed.]

**Keywords**: [Add about six keywords which would help a search engine find your report]

Acknowledgements

I would like to give thanks to my supervisor, Kakia, for the support I received. The scheduled meetings, for updates on this dissertation project, helped immeasurably. Due to the support, solutions to my problems were concluded very quickly. This allowed me to overcome hurdles as I had someone to talk to about the struggles, as well what has been tried and failed.

Table of Contents

[Declaration 1](#_Toc196403208)

[Abstract 2](#_Toc196403209)

[Acknowledgements 2](#_Toc196403210)

[Table of Contents 3](#_Toc196403211)

[List of Figures 6](#_Toc196403212)

[List of Tables 7](#_Toc196403213)

[List of Abbreviations 8](#_Toc196403214)

[1 Project Introduction, Motivations and Background 9](#_Toc196403215)

[1.1 Relevance to Security Implementation 9](#_Toc196403216)

[1.2 Project Aim and Expected Outcome 10](#_Toc196403217)

[2 Literature Review and/or Technology Review (other systems like it that use fingerprint scanners, distributed systems etc) 10](#_Toc196403218)

[2.1 Authentication 10](#_Toc196403219)

[2.2 Data Security 11](#_Toc196403220)

[2.3 Biometric Techniques (non deterministic) 12](#_Toc196403221)

[2.4 Existing Technology 13](#_Toc196403222)

[3 Project Plan and Methodologies (talk about how I done my project, breaking up tasks etc, agile waterfall, rapid prototyping) 14](#_Toc196403223)

[3.1 Project plans 14](#_Toc196403224)

[3.2 Methodologies 15](#_Toc196403225)

[3.2.1 Hardware 15](#_Toc196403226)

[3.2.2 Software 16](#_Toc196403227)

[4 System Analysis and Requirements 17](#_Toc196403228)

[5 System Design 20](#_Toc196403229)

[5.1 System Flow 20](#_Toc196403230)

[5.2 UML Diagram 21](#_Toc196403231)

[6 Implementation + progression (talk implementation of hardware/software) 23](#_Toc196403232)

[Testing and Evaluation 28](#_Toc196403233)

[6.1 Tests 28](#_Toc196403234)

[6.1.1 Hardware 28](#_Toc196403235)

[6.1.2 Capturing/Converting Fingerprint 28](#_Toc196403236)

[6.1.3 Storing Fingerprint 29](#_Toc196403237)

[6.1.4 Uploading Fingerprints 29](#_Toc196403238)

[6.1.5 Verifying Fingerprint 30](#_Toc196403239)

[6.1.6 Deleting Fingerprint 30](#_Toc196403240)

[7 Maintenance and Deployment 31](#_Toc196403241)

[8 Results and Discussion 32](#_Toc196403242)

[9 Future Work 33](#_Toc196403243)

[10 References 34](#_Toc196403244)

[11 Bibliography 37](#_Toc196403245)

[12 Appendix A 38](#_Toc196403246)

List of Figures

[Figure 1: Easy Colours 4](#_Toc143255913)

List of Tables

[Table 1test 3](#_Toc143256679)

[Table 2: Test Table 1 6](#_Toc143256680)

[To add entries to this list:

* insert/create the figure, then place the cursor in a newline below the figure. In the REFERENCES menu, select INSERT CAPTION.
* In the caption window, enter a name for the figure. I recommend that you add a colon then a space character before the name as the table number will automatically be inserted before you caption, for example if you want the name of the caption to be ‘Original Data’ enter ‘: Original Data’, Word will then place the caption “Table 2: Original Data’ into the document.]
* In the LABEL option choose FIGURE, then click OK. This places the caption below the figure.
* To update the LIST OF FIGURES, place the cursor anywhere in the List of Figures and press F9. Then select UPDATE THE ENTIRE TABLE and press OK.

List of Abbreviations

[Place abbreviations in alphabetical order.]

# Project Introduction, Motivations and Background

Two foundational pillars of cybersecurity are authentication and authorization, alongside the principles of the CIA triad: confidentiality, integrity, and accessibility. According to the National Institute of Standards and Technology (NIST) in standard 1800-25A, confidentiality ensures that stored data remains private and accessible only to authorised personnel. Integrity guarantees that the data remains unaltered and is safeguarded against destruction. Accessibility ensures that authorised parties can reliably and consistently access the data when required (Elkstrom *et al.,* 2020).

Authentication and authorisation are critical components of access control, which involves granting individuals or accounts the rights to access systems or resources. Authentication verifies the identity of a person or entity, typically through mechanisms such as passwords. Authorization determines whether the authenticated individual has the necessary permissions to access a resource. Authentication methods fall into three main categories: what you know (e.g., passwords), what you own (e.g., keyless fobs), and what you have (e.g., biometrics) (Idrus *et al.,* 2013).

## Relevance to Security Implementation

The importance of these concepts lies in their role in maintaining robust access control mechanisms. However, traditional methods like passwords can be forgotten or stolen, key fobs can be lost or cloned, and even biometrics can be subject to replication. This project explores the use of biometrics, specifically fingerprints, as a more secure alternative for authentication systems. Unlike passwords or fobs, biometrics are inherently tied to the individual and are difficult to lose or replicate. The improved safety aspect comes from using two prints from two different fingers as the password. This then means that there are 9,072 different combinations you can have with 10 prints. Although less combinations than a password, fingerprints cannot be forgotten or guessed by a threat actor.

## Project Aim and Expected Outcome

The project aims to design and implement a biometric authentication system utilizing a fingerprint scanner capable of recognizing multiple fingerprints during the login process. The implementation will rely on off-the-shelf hardware and open-source software to ensure accessibility and reproducibility. The expected outcome will demonstrate the viability of biometrics as a highly secure solution for systems requiring stringent security measures. The project emphasizes the operational functionality of biometrics rather than focusing on extensive software development. Wouldn’t it be far easier if you could just walk into a bank or an establishment and withdraw/pay with the fingers on your hand? The main issue, that the project is aiming to solve, is how unreliable outdated forms of banking can be. Cards are easily lost or stolen, leading to the potential misuse by strangers who either find it, or have stolen it. The project has the idea that biometrics can be used for banking transactions and, could potentially reduce security risks along with the added benefit of not having the burden of remembering cards and pin codes/passwords.

# Literature Review and/or Technology Review (other systems like it that use fingerprint scanners, distributed systems etc)

Through this review, we will discuss the importance of data protection. Using different formats of securing information, as well as authentication, authorisation, and biometric techniques. Using existing literature, this review aims to compare different literature and findings, to evaluate their effectiveness in system authentication and security. This will help decide what the best approach for biometric authentication methods and data storing are.

## Authentication

Authentication is a must for secure design of systems. Policies vary greatly across different systems. For example, some systems may optimise the use of a default deny policy. This policy means that the system denies them at every instance. The user must almost have to prove why they deserve to gain access, using standard methods such as a username and password to gain authentication and authorisation. Authentication systems as a concept, answer the questions of who is the person trying to gain access? and are they who they say they are? Idrus et al (2013). The standard of authentication currently, is the use of two step authentication. This is where the user has another piece of information to prove they are who they say they are. They are based on three different factors for such a system to work effectively. These are such factors as knowledge, possession and inherence. These are simply put as something they know, something they own or something they have. Passwords are something they know, key fob is something they own and fingerprint is something they have Velasquez et al (2018). Keeping this as robust as possible is vital to keeping data secure.

## Data Security

It is important for any sensitive data to be kept hidden from anyone, especially from someone that may misuse it. An established way to achieve this is to encrypt data at rest, as said by Shingari and mago (2024) in their paper on the importance of encryption for the medical sectors data, it is proven, that following strict security protocols lowers the threat of a data breach/leak. Later, they go on to mention that data encryption is a vital part of prevention. The kinds of threat prevention are unauthorised access, use, and disclosure of sensitive data. This means that encryption is a good standard to use for the project, to store the highly sensitive print data. However, another option that is to be considered for storing the data would be the use of a hash function. As described by kamal, P (2019) in security of password hashing, the hash function is used to store the plaintext password without being able to see what the password really is, as it has been hashed by an algorithm and isn’t readable to humans. It does this by storing the password with a fixed length output, regardless of the input size. By doing this when a user needs to be authenticated, their entered password is hashed by the same algorithm and compared to the stored hash. If they’re the same, then it has a matched password and the user gains access. This is also a good method to use and has become a standard, along with other techniques, to store passwords for standard methods as well as biometric authentication techniques.

## Biometric Techniques (non deterministic)

Biometric authentication has been a developing technology for decades. As discussed in the publishment, an introduction to evaluating biometric systems, Phillips et al (2000), there are many forms and use cases for biometrics. They discuss the main and most common biometrics, such as facial, fingerprint and vocal. There is also a discussions about how useful biometrics are. Even to the extent of potentially using fingerprints to sign key pairs for encryption. This supports the idea of biometrics being an ideal replacement for the generic alpha-numeral standard for passwords. Although some systems, particularly the facial recognition, has its flaws. Currently some of the issues with this system found in the same report, include lighting conditions. Some faults have been noticed with the performance under light differences. Faults with fingerprint biometrics systems found by M.H.Ali et al (2016) and Choi et al (2016) can be attributed to the type of scanning that you would choose, as well as the issues with image quality playing a role in the success of the matching process. There is the payoff that having a very fast system will have a lower accuracy rate. Which means that inversely, to have a better accuracy rate it is more robust to make it slower. This is something that needs to be considered when designing the system, as the project is based for finances, security is of upmost importance. As backed by the journal from Unar et al (2014) the methods have different subsets of biometrics. Having various means of biometric authentication opens the use of biometric properties, this is likely to decrease attack surface for any one person. For example, if your online banking has fingerprint technology but your social media uses retina scanning, if an attacker gains access to your social media, they wouldn’t be able to get your bank account as they would be different formats. In the past, this would be slightly different. Although the advice for more security is to have different passwords for everything, no one can remember all their passwords. This means that repeat passwords are very often used.

A diagram of a biometrical structure

AI-generated content may be incorrect.

Figure x: Biometric Types

## Existing Technology

In the modern day, fingerprint scanners are in the everyday smartphone. They are used not only for accessing your phone, but they are also used for payments and logging into different applications. There are many stipulations with fingerprint scanners integrated in mobile devices. They must consider the position of the scan, normal fingerprint scanners being mounted a specific way, they are likely to catch the print image the same way. Whereas on a mobile device, the scanner can be in many different positions. The scanner would also have different conditions to deal with, such as moisture differences and lighting differences Saavedra et al (2016). Through the paper they discuss a study, in this study it was found that there is a limit to the size of the fingerprint imaging size. The lowest size recommended for a fingerprint image is 320x320 pixels, as less than this recommended size found considerable degrading in matching performance. In another research paper, image quality plays a vital role in the success rate of the matching process. To enforce the highest standards, a high quality image should be taken for the base enrolment image. This then allows any image used for matching to be lower quality and still have a high success on matching Ordani and McCabe (2009). Fingerprint systems are already being designed and implemented for payment methods. In Japan, they are ready to start trialling this system, with 300 firms being involved with the trial Utzhanova (2016). The fingerprints of the customer will be linked to their bank account and used as the currency to pay for items. This supports the aim of this project as the basis of the fingerprint system is being trialled in other parts of the world. In the same journal, it goes on to talk about how Sweden is also trialling something similar. They will trial the use of biometrics to pay for public transport. This will make payments quicker and more efficient.

# Project Plan and Methodologies (talk about how I done my project, breaking up tasks etc, agile waterfall, rapid prototyping)

Provide a brief overview of the project management methodology and tools you have selected (agile/waterfall, MoSCoW prioritisation, critical path analysis etc.). Make it clear why you are using the selected approaches. Explaining why you rejected other techniques can help with the justification of your choices.

Provide an initial plan (Gantt chart or other) which you created at the start of the project. Provide an update final version of this plan in the review 9 Results and Discussion of the report to show how the plan changed.

## Project plans

During the creation of the project, the agile-waterfall hybrid development process was adopted. This method suited the project progression, as it involves incremental improvements over the lifecycle of the process. I found it more beneficial to use than either waterfall of agile on their own. This helps with making robust requirements but then also the added benefit of being able to compartmentalise sections of the project and ensuring that each part works independent of each other before trying to bring it all together but allowing flexibility with creation process. This ensures that any faults within each section is fixed. This means that rather than having a large subset of, for example, code to run through every time there is a fault, it limits the possibilities as to where and error is occurring from. For example, an extra step in the fingerprint scanning function was required, after it was thought that scanning the print function was finished, to improve the print image quality. Adding the extra efforts of the MoSCoW method for requirements prioritisation meant I had a clear outlined goal of the essential parts for the project. This enabled me to stay on task with greater efficiency and keep the focus on topic and not on adding to the workload needlessly.

A screenshot of a computer program

Description automatically generated

## Methodologies

In this section I will outline the methodology that I followed, only depicting the information that was found to be useful and the parts of the process that were successful. This shows how the project can be made from start to finish without any faults, allowing replication in a successful manor and time frame.

### Hardware

The set up of the Arduino hardware was a simple task. The hardware included the set-up instruction in the form of a website link, to a file of the specifications and the instructions on how to set it up. The useable cables that this project requires from the Arduino are the black one (1) which is ground, the red one (4) which is power in using 5V, the white one (2) which is the serial in and the green one (3) which is serial out. The use of the two serial wires enables the Arduino to communicate with hardware or other software that Is being used to send and receive commands and data. In this case it is python being used to send and receive data with the Arduino (see figure for graphical representation of set up, also a link to the entire pdf).

[A white background with black text

Description automatically generated](https://www.farnell.com/datasheets/3216194.pdf)

Figure x: Datasheet Wire Information

### Software

The interface for the scanner project needed to be a friendly and useable language. Python was therefore used for this purpose, as it has a lot of support for software development capabilities. The added benefit of choosing python for this use case over java, for example, is that pythons support for fingerprint scanning projects is larger. They have specific libraries to help in this project. It has a very simple serial connection creation, for the communication between scanner and interface, as well as being able to create a GUI, if necessary.

# System Analysis and Requirements

To ensure timely delivery of the project outputs and The MoSCoW prioritisation method is used to define the prioritisation of features for the application. The method splits requirements into “must haves” (M), “should haves” (S), “could haves” (C) and “wish to haves” (W) helping developers with prioritising these towards a minimum viable product (MVP). Requirements will also be split and organised using the functional vs non-functional methodology: as Kurtanovic and Maalej said, Functional requirements are system functionalities, things that will be interacted with such as, a way to add users. Non-functional requirements are system properties, things that happen in the background such as security for the biometric passwords (Kurtanovic and Maalej, 2017).

|  |  |  |  |
| --- | --- | --- | --- |
| Mo | S | Co | W |

|  |  |  |
| --- | --- | --- |
| Functional Feature | Description | priority |
| User authentication | Using some form of biometrics to log into the ‘banking app’. | Mo |
| Add users from an admin login | Have a way to add users to database to be able to authenticate. | Mo |
| a banking home page | Make a home page to log into to show the authentication working | Co |
| Be able to manipulate data | The ability to delete users from the database | S |
| Economy of mechanism/ease of use | The interface for the end user to be simple and easy to navigate, text or icons will make it as self-explanatory to be able to navigate around | Co |

|  |  |  |
| --- | --- | --- |
| Non-functional features | Description | priority |
| Storage | Have a database with the biometrics stored in | Mo |
| scalability | Be able to handle as many login details as needed | W |
| Security | Make sure the fingerprints are stored securely and not in a raw image | Mo |
| Availability | The systems need to be useable at least 90% of the time | W |
| Back up and recovery | Periodically backup the fingerprints as so they have a fresh recovery point, in case of data error | Co |
| Error handling  Fault tolerance | Appropriate error handling messages for the user | S |
| Visual and/or confirmation of fail/success logging in | Give some feedback as to the state of logging in, e.g. fail try again or success | S |

In this chapter you explain to the reader the specification of the system you intend to build and, just as importantly, **how** you arrived at those requirements.

Depending on your project, you could include subsections covering:

1. Technical description of existing system/s which your project will integrate with and how it will integrate with those systems.
2. System analysis process (how you acquired the system requirements)
3. Functional Requirements
4. Non-Functional Requirements
5. Requirements Prioritisation and/or specification of Minimum Viable Product (MVP)
6. Technology Choices

# System Design

This section will be used to display the initial design of the system. Throughout the process, there will be displays of the system changing as issues arise and functionality is added and changed.

## System Flow

The fingerprint scanner project needed to prove the concept that biometric scans can work for passwords. As it was more of a practical working system, the interface isn’t of high priority, so a command line style interface works well. As figure x shows the initial flow idea, it is to simply be able to enrol fingerprints, have them saved to a database and then be able to authenticate the user, if they are in the database, when they present their print to the scanner.

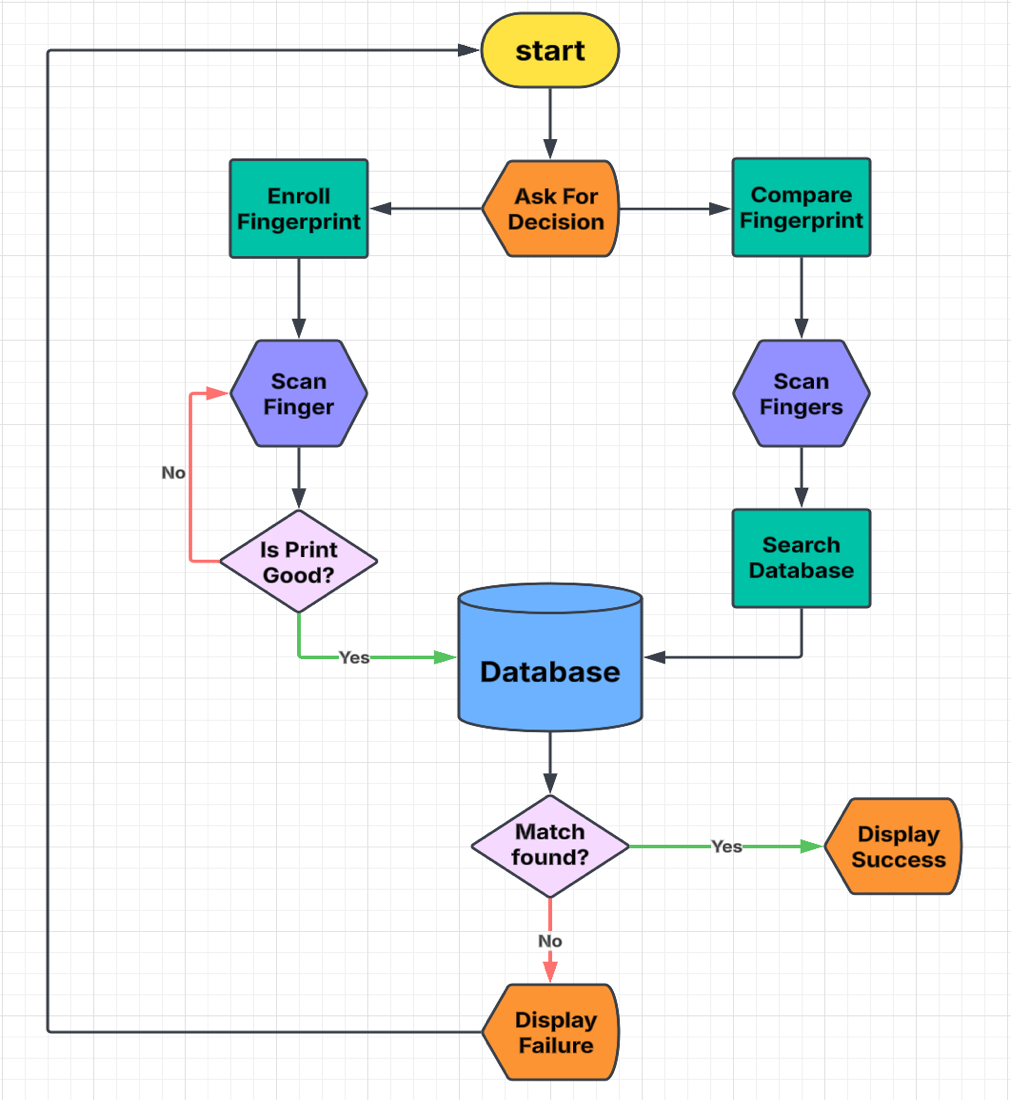


Figure x: Fingerprint system flow

## UML Diagram

Figure X is a class diagram, this diagram shows the methods that are within the python code. This allows the reader to see what methods are used and which objects and variables have been created inside each method, as well as what data type the objects and variables are. It has been hybridised to show the flow of the system. This makes it clearer how the functions interact with each other

A diagram of a data flow

AI-generated content may be incorrect.

Figure x: Class Diagram Python

Figure X is the second hybridised class diagram, this is to represent the Arduino code. As seen, it depicts the methods, variables and objects. This diagram shows how the language used to code slightly different, as everything is controlled by the loop function. After each function has run, it goes back to the loop and then executes the next function in order. This enabled the project to be more organised as it had a better execution structure to visualise than the python code does.

A diagram of a computer program

AI-generated content may be incorrect.

Figure x: Class Diagram Arduino

# Implementation + progression (talk implementation of hardware/software)

Work has been undertaken up until now to resource the hardware and decide on the software that would be compatible to work on as well as setting up the development pipeline. Hardware has been set up to work together and work on implementing the scanner and hashing / encryption components with some positive results. During this first development stage, several challenges that required iterative problem-solving and technical adjustments have been encountered. Below an overview of the issues is presented together with the solutions or steps taken to solve these issues.

**Issue 1: Fingerprint Scanner Model Creation.** The initial challenge was the inability of the fingerprint scanner to convert the finger image into a usable model. To resolve this, a solution was devised where two fingerprint scans were combined to generate a single, successful model. This approach overcame the initial hurdle effectively.

**Issue 2: Exporting the Model to Python.** The next obstacle involved exporting the fingerprint model in a format that could be read by Python. This required enabling data transmission via the serial interface. After generating the model, a method was developed to write the data to the serial port, enabling Python to read it successfully.

**Issue 3: Initiating the Fingerprint Scanner from Python.** A further challenge was initiating the fingerprint scanner directly from Python. To achieve this, the Arduino system was configured to respond to a specific command sent via the serial port. Python code was then written to send the command, allowing the Arduino to interpret it as an instruction to start the program. This integration allowed seamless interaction between Python and the scanner.

**Issue 4: Reading Data in Python.** Reading the serial data in Python posed significant difficulties. Various methods were tested to establish a connection with the Arduino and display incoming serial data. Issues with data flow necessitated introducing delays to ensure proper data transmission. A dedicated Python function was created to continuously read the serial data while it was being populated. This approach ensured that the data could be printed and utilized effectively.

**Issue 5: Saving Fingerprint Data.** The fingerprint data generated by the scanner was initially a raw image, which was converted into binary data for processing. While this data could be stored in a database, encryption was required prior to storage. A method was implemented to encrypt the data after capture and before writing it to the serial port. On the Python side, adjustments were made to specify which portion of the data stream to save. The Arduino code was modified to include markers indicating the start and end of relevant data. A Boolean flag controlled the data collection process, ensuring only the desired data was saved to a variable.

**Issue 6: Testing Data Saving.** To validate the saving functionality, a YAML file format was used for testing purposes. YAML, being human-readable, allowed for quick verification of the saved data. Once the functionality was confirmed, the focus shifted to storing the encrypted fingerprint data in a MySQL database.

**Issue 7: Developing a Verification Function.** A verification function was necessary to compare newly captured fingerprints with those stored in the database for authentication. The Arduino code was updated to include an if statement that executed a verification routine when a specific command was received. This routine accessed the database to perform fingerprint comparisons. However, initial implementations faced issues, including failure to return data during enrolment due to serial conflicts.

**Issue 8: Secure Saving.** Initially the data was going to be saved in an encrypted format, later this was changed to using a hash of the fingerprint, as the idea to solve the problem of finding a matching print in the database. The theory behind it was based off normal passwords matching. The print would be hashed and compared with all the hashes to find a match, as they would come out the same. This was an issue however as small differences, such as pressure, on the imaging of a finger would cause a big change in the hash output. To combat these issues, I investigated ways to have an error acceptance level. I researched into this idea and found hammering distancing, this is an algorithm that checks the binary format of two equal length strings, it counts how many bits are different and you can set the limit on these. After implementing, I found it didn’t work, the difference in print data proved too much, further research showed me other methods, such as mini hashing which also didn’t work. The data saving format therefore went back to encryption.

**Issue 9: sending data to Arduino.** As a method to try and authenticate users, I planned to use the Arduino built in method that can compare prints. the only issue with this idea, was that the Arduino can only compare with a print on the buffer/hardware database. My initial idea was to send the prints over to the Arduino scanner database. To do this I would need to use the same method that was used to send them to python, chunking data. There was an issue when the Arduino code contained the method for comparing prints. It was not running at all with this included. This was confirmed by removing the verify function from the loop and running it. When it successfully run after removing the function, I knew that the problem was with this. While trying to apply the method for sending data back to the Arduino however, the issue of overhead came into play, as the Arduino has a preferred reading speed. This means that on a small scale it would most likely be okay to use but scaling is a huge problem for this method.

Several debugging techniques, such as using print statements in the Arduino code, identified the root cause as buffer overflow during serial communication. The Arduino was reading the serial constantly and the extra function was causing issue. To address this, a chunking method was implemented, which divided the data into smaller, manageable pieces for transmission. An extra step to make sure the serial was empty before running the next function, to avoid an overflow. This adjustment resolved the issue, enabling successful enrolment while maintaining the verification method in the code.

**Issue 10: sending fingerprints from python to Arduino/scanner.**

The fingerprints had to be chunked, in the same way they were for enrolment. They couldn’t be saved to a variable in C++ as this couldn’t be used to then save the print to the scanner. After trial and error, the realisation that there needed be a buffer for the print to be saved to. This would enable the print to maintain the correct data type and be handled correctly for the scanner. After creating the buffer object, data was appended to it, successfully saving the fingerprint back into C++ (Arduino).

Saving the print to the scanners buffer for comparison displayed some difficulty. The buffer size was not the right size. It had been set to the size of the fingerprint template size so it was strict on the size of input it would be passing. Using the index as a delimiter, to keep track of the buffer received data size, it was shown to be either far too much or not enough. By utilising the python function to check the length of the data, it was confirmed to be the correct length in the primary stage of sending. This means that issues occur within the behaviour of the C++ code. Further investigation revealed that the serial may contain small amounts of data and would need to be cleared prior to receiving the print. To do this, a read function is used to make sure it is completely empty.

**Issue 11: scope issue:** now the Arduino can save a print, we needed to be able to send all prints to the scanner. The program could successfully iterate and send all the prints through. The Arduino then handled the prints and was able to safely save them. There are however issues with speed. Due to the limitations of the hardware and connection speeds, this process was very slow. With a small scope it would be more than acceptable, but when the customer base surpasses fifty people, also globally, this wouldn’t be acceptable realistically. To improve upon this idea, the ability to search for one specific print set made this process a lot faster and more manageable for the hardware. Using a username associated with the user’s details, allows the program to retrieve and send just the expected customers prints to the scanner. Reducing the amount of data being transferred and stored on the scanner, which is good security practise.

**Issue 12: comparing fingerprints.** Using the built in Arduino comparison algorithm follows the secure coding principle of ‘don’t reinvent the wheel’. It is proven to work as it is standard in the Ada fruits library. This also saves time for the project, as making a comparison algorithm is possible but very time consuming. While trying to make the scanner accept the fingerprints, it was discovered that it wouldn’t accept them. While diagnosing the possible reasons, it was discovered that it was to do with the method as to how the scanner accepts prints. The scanner has an order that it ‘likes’ to have the prints in. This being the process on how it captures a print. To bypass this error and make it possible to send an external print, the solution was to create a custom method withing the Arduino source code. Creating the ability for the scanner to accept the external print, if it is in the format that it likes. This is not an issue, as the prints are all scanned by the method that the Arduino runs on.

**Issue 13: accuracy of the enrolment print.** when trying to compare prints, there were problems with how well detailed the saved print was. This can reduce the confidence score in the comparison as the print is not as complete as it could be. To avoid this issue entirely, the solution was to add more scans of the print on the enrolment process. This then increased the data point imagery on the overall template, as it had a more accurate reading of the fingerprint. This also meant that when they re-applied their finger onto the scanner, any tiny movements meant that more of the finger was being processed. This was useful as it means the scanner would still have a high confidence score for authenticating, even if the user has their finger at a less than ideal orientation.

**Issue 14: clearing the fingerprint cache.** The prints that the scanner uses to compare to a newly scanned print cannot stay on the scanners database. This could be a big data leak possibility, if someone is able access the scanner in some manner. This also caused issues with being able to repeat the verify process indefinitely, as the first register was the only one being used and it wouldn’t allow overwriting. There must be an automatic clearing of the prints from the scanner after the process has finished, regardless of a successful authentication or a failed one. The method to clear the scanner was introduced to mitigate these errors. Allowing the scanner to be able to accept a new print every-time the verify process gets run.

**Issue 15: continually running the program.** C++ has a different run style than python. The run order is set up in the loop function, as it acts as a return spot for the programs. To make sure the program can be run without having to keep pressing the run button, the order had to be precise. Within the loop, there was a method that is purely to clear the serial, so each function starts with a clean buffer and doesn’t have excess noise.

# Testing and Evaluation

## Tests

This section will depict the testing methods used to make sure the project worked suitably and effectively. This will be a methodological account of how each section was worked on to completion.

### Hardware

While building the hardware, it became apparent that an adapter was needed so the cable from the Arduino, can connect to this specific laptop. This is due to the fact that it doesn’t have a USB connector, only a USB-C connector. A USB to USB-C connector was used. The Arduino hardware was assembled, following the instructions from the purchase link that came with the scanner. There were obvious signs that the Arduino was powered on correctly, with the visual aid of a blinking LED. To test that the scanner was operational, a built in program to make the scanner blink was installed and run, this was successful. To then see if the scanner would capture images of prints, a program was written to just take an image. Again, this was successful in creating an image, so it was proven to be fully operational.

### Capturing/Converting Fingerprint

Now that the scanner has been confirmed to be fully operational, the next challenge was to capture fingerprints to then encrypt and save in a database. The issue is the format that the print needs to be saved in. The print cannot be transferred as an image to python. This is a limitation due to security, but also the scanner, as the scanner reads the print in a binary format. The scanner takes an image of the print and changes the image into a numerical representation of the print image. More specifically a feature extraction-based representation. These points are the properties that all fingerprints have, the arches, loops, ridges, valleys, swirls, and triangles (these triangles are called delta points) Hong and Jail (1998). As all fingerprints are unique to every individual person, there are no two fingers that have the same scanned image, making them unique to one person. After the print has been taken, the built in function to convert the image to a template is used, this is where the print is turned to the binary format ready to be encrypted and then externally store.

### Storing Fingerprint

Now the fingerprint has been transferred into the correct format, allowing it to be encrypted and stored securely. A way of sending the print over the serial to the python program needs to be created, this is for further handling of the data. This needed to be done primarily to be able to save the print on the external database, rather than on the scanners internal memory, which is a more secure way of handling the data. The database setup occurs inside the python code, it is initialised as soon as the program is run. Using the connection function and SQL commands in python to make the database tables and set the parameters of the tables. The creation of the database was confirmed by going into the database application and checking the schema. There are extra steps to check python has been connected, using error handling. If it is connected, then it displays a print message in the terminal. If it fails to connect to the database, it falls into error handling messages. The fingerprint is then encrypted using a secure method, in the case of the project, for simplicity, it uses the built in fernet encryption. But would also work with other methods. The print is 512 bytes in data size, which is then saved in the database with either the BLOB (stores up to 64KB) or LONGBLOB (stores up to 4GB) data type, along with the id, name and creation date of the account.

### Uploading Fingerprints

To be able to verify a fingerprint, the scanner must have access to the enrolled fingerprint of the user, for comparison. The small issue of having the fingerprints stored in an external database, means that there needs to be a method of sending specific prints back to the scanner. To make this possible, the program takes a name/username as the variable, searches the database and gets the print of the matching username. It then decrypts it, chunks it up into more manageable pieces, so there isn’t a synching issue between python sending it and the Arduino program receiving it. While receiving the chunks, the Arduino appends the print to a preset buffer. It is confirmed that the print is the correct size as it doesn’t cause an overflow issue. This is proven because the error handling checks if the index of the data received, is larger than the buffer size. Another way this has been proven is printing the length of the data when python retrieves the print as well as checking the data length in the buffer from the Arduino program. They both match up with the correct size expected, which is 512 bytes.

### Verifying Fingerprint

Arduino has a variety of build in methods, these are available to upload to the hardware and use. To prove that the fingerprint was in the correct format and useable, the opensource method was used, just for the purpose of testing. Upon successfully uploading the print back to the scanner, I used this method for testing the matching of the fingerprints. This worked and proved that the print was in the correct format and could then be used for a custom made fingerprint verification method. The proof of success was shown by the program displaying that it has a fingerprint in the sensor’s memory, it then fed back that it matches the fingerprint with a success rating. It was tested again, but using the wrong finger, which fit displayed that it didn’t match. After this success, it was decided to use this built in comparison algorithm, as it made more sense to use a proven working method than to make one myself. This is due to the fact of the time frame of the project.

### Deleting Fingerprint

To make sure that no prints were left on the sensor and not allowing the risk of the data being stolen, the prints needed to be cleared off the memory. This was tested while the method for loading the print from the external database onto the sensor was being executed. In the Arduino, there is a function to be able to get the number of prints that is stored on the sensor. It’s in two parts, first it is finger.getTemplateCount() and then displayed by printing it to the serial. Using this, I could see that after storing the print it showed a total of one. To delete the print it, I used the built in method of finger.emptyDatabase(). This is used in its own method, as it needed to be a conditional check, to see if the print was acceptable. This is then checked by invoking the fingerprint check again, to see if it has been successful. This was a success as it displayed zero prints on the scanner after the code was run.

In this chapter you should document the results of the tests you have carried out against the requirements and design, as specified in previous chapters. This may also include results of user testing if appropriate.

Testing may result with large volumes of test results which would not be appropriate to include in the main text and would make the appendices excessively large. If this is the case, it may be worth considering placing large volumes of test data as a separate digital file upload into Brightspace, or include a sample of tests in your appendices if there are too many. However, do not expect assessors to read every page of your tests, they will be more interested in the outcomes of those tests rather than the documentation of vast amounts of individual tests.

# Maintenance and Deployment

This chapter is optional depending on the project. If you do use this chapter, provide an overview of how the system is deployed, and any limiting factors on deployment issues which a user should be aware of. Which other software or libraries must be installed? A full installation guide would be useful. Describe how the system can be patched and upgraded. Are if there are licencing issue to consider?

# Results and Discussion

In this chapter you critically evaluate the overall effectiveness of the initial design. Assessors will be looking for evidence that you can ‘objectively critically evaluate’ your work. This means considering both the positive and negative aspects of your project to demonstrate your problem solving and decision-making skills. Whenever possible, you should support your discussion with citations to demonstrate that you are evaluating your work in the context of your wider field of study.

You should also address how many of the initial requirements were achieved, and why some were not. Think carefully about using phrases such as “***Due to time constraints*** *some requirements were not achieved*”. This leaves yourself open to criticism that you initially over specified the initial design within the available time, or that you suffered from feature creep or that you did not adhere to your proposed plan. Any of these scenarios suggest poor project management. This is why it is important to plan your requirements realistically.

See this library link to help you with Critical Thinking: [Critical Thinking & Writing - Academic Writing - Learning and Teaching at University of Suffolk (uos.ac.uk)](https://libguides.uos.ac.uk/academic-writing/critical-thinking-writing)

# Future Work

Use the limitations of your project, as outlines in the ‘Results and Discussion’ chapter, to identify further improvements to the existing implementation as well as new functionality. This chapter should act as a jumping-off point for other readers (and/or yourself) to continue your work.

# References

[RefWorks & Reference Management - Academic Skills - Learning and Teaching at University of Suffolk (uos.ac.uk)](https://libguides.uos.ac.uk/academicskills/RefWorks)

Choi, H., Negar, A and Jain, A.K. (2011) 'on the evidential value of fingerprints', *2011 International Joint Conference on Biometrics (IJCB). DOI:10.1109/JCB.2011.6117550*

Elkstrom, M., Lusty, L., Sexton, J,. Sweetnam, J,. Townsend, A,. (2020) *Data Integrity: Identifying and Protecting Assets Against Ransomware And Other Destructive Events*. Available at: https://www.nccoe.nist.gov/publication/1800-25/VolA/index.html (Accessed: 20th November 2024).

Hong, L and Jain, A. (1998) ResearchGate, *Classification of Fingerprint Images.* p.p. 1-8, Available at: https://www.researchgate.net/profile/Lin-Hong-12/publication/2929466\_Classification\_of\_Fingerprint\_Images/links/54733a5a0cf216f8cfaeb8b7/Classification-of-Fingerprint-Images.pdf?\_\_cf\_chl\_tk=gGIjmRBQoDbqiets\_XPaf1NI5aUH6oiWiYZoZOklu1s-1744128421-1.0.1.1-uLH7OAoK7SaRpJUvpLHVv\_ljGoKV.sX0w7LGrzvq7K8.

Idrus, S. Z. S., Cherrier, E., Rosenburger, C. and Schwartzmann, J. (2013) '*A Review On Authentication Methods*', HAL open science, Australian Journal of basic and Applied science pp 95-107. (Accessed: 13th December 2024)

Kamal, P. (2019) *security of passwords hashing in cloud*. Cloud State University, USA. Journal Of Information Security. Vol.10 No.2

(Accessed 2nd January 2025)

Kurtanovic, Z,. Maalej, W. (2017) *Automatically Classifying Funtional and Non-functional Requirements Using Supervised Machine Learning.* Lisbon, Portugal., Publisher: IEEE, p 1(Accessed: 8th January 2025)

Random Nerd Tutorials(2019) *Guide To Fingerprint Module with Arduino (FPM10A)*  Available at: https://randomnerdtutorials.com/fingerprint-sensor-module-with-arduino/ (Accessed: 13th December 2024).

M.H.Ali, M., Mahale, V and Yannawar, P., A.T, G (2016) *Overview of Fingerprint Recognition System*. Chennai, India: IEEE. **DOI:**[10.1109/ICEEOT.2016.7754900](https://doi.org/10.1109/ICEEOT.2016.7754900) (Accessed 2nd April 2025)

Orandi, S and McCabe, R.M. (2009) *Mobile ID device best practice recommendation version 1.0*. National Institute of Standards and Technology (NIST), NISTSpecial Publication 500–280.

Phillips, J., Martin, A., Wilson, C.L. and Przybocki, M. (2000) *An Introduction to Evaluating Biometric Systems*. Computer. IEEE. Volume: 33, Issue 2. pp. 56-63. DOI: [10.1109/2.820040](https://doi.org/10.1109/2.820040)

(Accessed 1st April 2025)

Saavedra, B.F., Reillo, R.S., Gomez, R.R., Jimenez, J.L. (2016) *small fingerprint scanners used in mobile devices: the impact on biometric performance*. University of Madrid, Spain. IET Biometrics.

(Accessed 4th April 2025)

Shingari, N,. Mago, B. (2024) *The Importance Of Data Encryption in Ensuring the Confidentiality and Security of Financial Records of Medical Health*. Gwalior, India: IEEE.

(Accessed: 3rd January 2025)

Unar, J.A., Seng, W.C. and Abbasi, A. (2014) Pattern Recognition, *A review of biometric technology along with trends and prospects*. Available at: DOI: <https://doi.org/10.1016/j.patcog.2014.01.016>

(Accessed 4th April 2025)

Utzhanova, A. (2016) European Journal of Sustainable Development, *Fingerprint Technology and Sustainable Development*, Volume 5 Number 4, DOI: <https://doi.org/10.14207/ejsd.2016.v5n4p325>.

(Accessed 7th April 2025)

Velasquez, I., Caro, A and Rodriquez, A. (2018) Information and Software Technology. *Authentication Schemes and Methods: A Systematic Literature Review.* Science Direct. Volume: 94. pp 30-37. DOI: <https://doi.org/10.1016/j.infsof.2017.09.012>

(Accessed 4th April 2025)

# Bibliography

A bibliography is optional. A bibliography includes all sources which were consulted in the process of the project but not cited in the main text. This should be presented in the same format as the References.

# Appendix A

Appendices are optional.

If you have any information which is **OPTIONAL** and **NOT ESSENTIAL** for the reader, but you think they may find it interesting, place it in an appendix. The reader will refer to it **IF** they want to. The reader is not expected to read anything in the appendix as this is optional information. In the same sense, do not assume that anything you place in the appendix will be read by the assessor. Assessors are not obliged to read the appendices or consider any of their contents in the marking process. Markers will check the appendices are not being overused, but not necessarily consider their content.

The appendices are not a dumping ground for more ‘stuff’ which you cannot fit within the word limit in the main document. It is wrong to believe that the more ‘stuff’ you cram into the appendices, the higher marks you will receive. Quite the opposite, overuse of the appendices will be penalised. The appendices should not be longer than the report itself.

Use letter identification for each appendix: Appendix A, Appendix B…..

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

Table 2: Test Table 1