University of Plymouth

School of Engineering,

Computing, and Mathematics

COMP3000

Computing Project

2023/2024

Multi-modal authentication using facial recognition and keystroke analysis

Alexander Williams

10681389

BSc (Hons) Cyber Security

Acknowledgements

This project was only achievable due to the valuable insights of various people within my academic journey and the motivation to progress within this field of study.

I would like to specifically mention my previous manager Nicholas Sharratt, who helped me develop my interest of user interaction with authentication during my placement year at the University of Plymouth Information Security Team.

I would also like to thank my project supervisor Dr. Nathan Clarke who helped influence my original research-based project into an implemented solution, while encouraging me to develop my skills regarding machine learning to make it a key factor of the product.

Finally, a special thank you to my partner and parents, who encouraged me throughout my entire academic journey, especially when times got tough.

I am extremely grateful for their invaluable insights, encouragement and assistance that has been provided during my academic journey. Your contributions have been core to the creation of my product and academic experience.

Abstract

This document explores the modern-day usage of multi-factor authentication and the intrusions this has on the regular user versus the security needs for the business. The project is a proposition of an alternative method of multi-factor authentication that can be uniquely tailored to any organisation that implements this into their wider system while not being intrusive to the end user. The prototype investigates the fusion of biometric modalities, being facial recognition and keystroke analysis, as an enhancement to accurate authentication, reducing both security vulnerabilities surrounded with single factor authentication and the frustrating user experience of multi-factor authentication. The product explores methods of authentication that is unable to be replicated by the rapidly evolving genre of Artificial Intelligence.

The background of the project stems from the accessibility of the frequently used options for multi-factor authentication and how these can be circumvented by

adversaries. Delving into the capabilities and limitations of current solutions in the industry, which will develop the objectives of the project.

The requirements for the project are then outlined, followed by the legal, social and ethical issues that will be challenged during development of the project. Incorporating the varying factors then informs us to develop the architecture for the application, as well as the method of approach. Furthermore, the report delves into the usability aspects of the multi-modal authentication system, considering user acceptance and experience through qualitative feedback and usability testing. This feedback is reflected within the design of the system to allow for a better user experience which can be tailored to the individual.

The development process is then detailed, discussing the five main stages of development are discussed, which presents the overarching importance of each individual part of development, the challenges, and the importance of each section. Development changes are then further explained by the accompanying testing section, ensuring that the end product met all objectives.

The report concludes with a reflection on how the project developed and which objectives were met. This also discusses the changes that were highlighted throughout development as well as a section detailing future developments and lessons learned.

To conclude, the product contributes within the field of authentication with demonstration of multi-model biometric authentication, with the effectiveness of using key biometrics such as facial recognition and keystroke analysis. The findings can be used for enhanced security and end user experience within system's authentication, offering a practical implementation for policy makers and system developers.

Contents

Word Count: 9962

GitHub Link: https://github.com/Firey56/comp3000-multi-modal-authentication

1. Introduction

The product that has been researched, designed, and implemented is aimed at enhancing security for organisations while also considering end user interaction at the forefront of the design process. The primary requirement of the product is to allow for flexible authentication where the organisation may tailor the experience of the user depending on the authentication resultant of the various methods implemented for access control. The multi-factor approach has been designed to be non-intrusive to users, not requiring either a third-party app or device, and for the interaction to be as simple as them starting their day by logging in.

The primary objective of the product is to build a report on each individual user's individual typing habits when they are trying to login to their system. There are two key biometrics that are unique to all users that are used to build this report – facial recognition and keystroke analysis. The product will result in giving an overall confidence level which is resultant of varying authentication methods, varying from low to high. This value will then be used by an organisation for their own access control needs allowing for stronger security implementation. A discussion on existing methods of multifactor authentication will be explored with the limitations of the solutions and how the product in hand overcomes that challenge. This will allow for an requirements analysis to define the objectives of the product and how these will be achieved.

Following on a discussion of varying legal, social and ethical issues that are approached throughout the project and how they will be incorporated, challenged or how a user’s experience may change. We then discuss the project management approach and the technologies that are used to ensure that the method is successful. Following the methodology, an informed discussion on the design of the application and the architecture underlying.

Implementation is a section that discusses the project lifecycle and how each stage of the project was tackled, discussing any issues that were solved along the way. A series of testing is then implemented to inform further development.

Lastly, a discussion on the success of the project and changes during development, as well as a reflection on the project as a whole.

2. Background and Objectives

2.1 Overview

There are a variety of multi-factor authentication options, ranging from one-time codes to having plug-and-play devices that act as an authenticator. However, many of these variations rely on having a strong foundation ensuring that a unique secret key (seed) is not able to be leaked or cracked, while other variations are vulnerable to risks such as adversary in the middle attacks. This section will discuss some alternative solutions that are frequently used and their limitations. This will then be used to inform our project objectives as to overcome the limitations of the researched methods.

2.2 Alternative Solutions

2.2.1 Google/Microsoft Authenticator/Authy

Google Authenticator, Microsoft Authenticator and Authy are all popular multi-factor authentication solutions that are favoured by many. When a user is adding multi-factor solution to an account, the user would scan a QR code which would integrate the account into the application. The applications then generate unique time-based one-time passwords (TOTP) which act as the multi-factor.

To generate these codes, there are a series of factors in place. A shared secret between the application and the authenticator that is unique to the account and a time vector. Due to the symmetric cryptography in use, it allows for either application to be able to calculate the TOTP, even when there is no network connectivity. This form of authentication is generally favoured by users due to its simplicity in setting up and ability to access as researched by Usenix (Reese et al., 2019) and shown in figure 1. However, the limitation with this study is that their tested audience were university students, who generally can be more tech savvy.

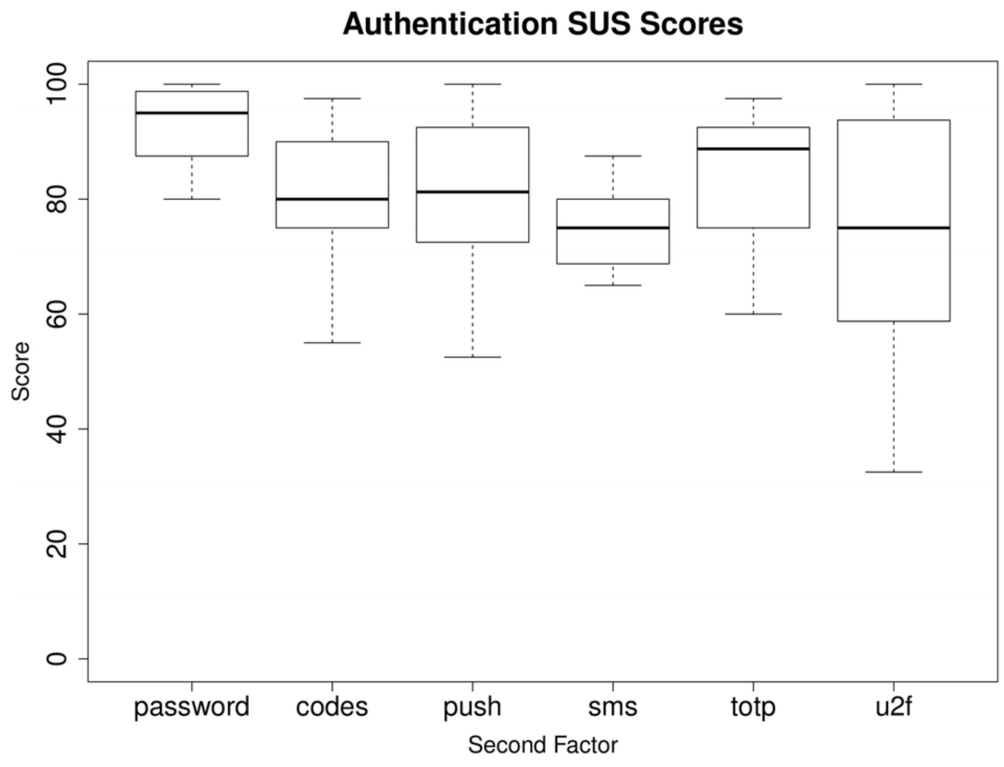


Figure 1: Authentication System Usability Scale Scores - Usenix

2.2.2 One Time Codes via SMS or Email

This is generally the worst rated method of MFA for usability score as shown in the study previously mentioned. OTPs function very similarly to TOTPs as the server-side will be generating an OTP using a variety of functions, however this doesn't take time or a shared-secret in as a factor. The website requesting the OTP will generate the code and send this to the user through their requested channel (Twilio, *One-time passcode verification (OTP)*). This is generally considered less secure due to the channels being more vulnerable to attacks than software tokens as there are other factors at play for generating the codes, as well as access.

2.2.3 Limitations

While these forms of authentication are generally favoured by the population, they are not without drawbacks.

A key example of limitations within using Authenticator apps is the requirement of secondary equipment. If a user is unable to access the device with their authenticator, they are at an impasse. Although this is beneficial for account security as it prevents unauthorised access with compromised credentials, it can cause work-day disruptions.

Secondarily, if a user loses the authenticator device, it can be generally quite difficult to remove from their account without contacting IT support. While this does mitigate the risk of an attacker maintaining persistence if the account becomes compromised, it can lead to end user frustration.

Regarding the limitations of OTP via Email and SMS, these are unable to be computed offline due to there being no shared secret. This then has a reliance on network and internet accessibility to access this code. This generally interferes with the user's experience as they now must wait for external factors at play. Furthermore, both SMS and Email are generally more vulnerable. SMS messages can be intercepted and spoofed so that the end user never receives the codes, but an adversary could instead. If an adversary has compromised multiple accounts and they are able to access the victim's email, they are able to bypass MFA as they would also have access to the codes.

Generally, the use of SMS based OTP is considered less secure due to the risk of adversary's being able to access the code due to external factors. As there is reliance on network service, these codes are often designed to have a longer time to live, while also having less entropy (Lei et al., 2021). Therefore, they are at a higher security risk for an adversary to spoof or intercept the code.

2.3 Objectives

The aim of the project is to create a non-intrusive form of multi-factor authentication using varying biometrics which can be used by companies and privacy-conscious users to enforce access control to systems. Therefore, the objectives for the project are as follows:

- To research user experience (UX) interface design to allow for the application to be accessible to all users.
- Build an application that uses multiple biometric modalities to authenticate.
- Build an application that allows for enforcement of access control for each individual authentication attempt.
- Design a user interface that is accessible and intuitive.
- To follow agile project management techniques and complete a minimum viable product within the allocated time.
- Create an unintrusive method of application to ensure users aren’t disrupted.

3. Requirements

3.1 Functional Requirements

For development of the application, a series of requirements were created aligned with the project objectives. The categories that the requirements were split into were core and desirable to ensure that development stays within the scheduled time. Core requirements are required for the minimum viable product and to achieve functionality aligned with project objectives. The desirable category are other requirements that greatly enhance the application and will aim to be included in the prototype. However, it is not feasible for all desirable requirements to be complete within the time frame of the project.

3.1.1 Core Requirements

3.1.1.1 Login System

The product is required to have an entire login system where the product can generate a confidence score based on the user provided information. This will allow for a user to type their username and password, while also providing a series of images that will be used by the facial recognition system. The system will record the keystrokes and pass the data to the rest of the application.

3.1.1.2 Register System

A register system to allow for user enrolment within the application will be required. This will allow for a user to submit a username, password, and an image of themselves that will be used for all future authentications. The register system will also record the keystroke pattern of the user and generate a user table for future attempts.

3.1.1.3 Facial Recognition

The first main biometric modality of authentication for the product will be facial recognition. The application will have a base user photo from when they registered, and during every attempted authentication attempt, several photos will be taken of the user. The comparison photo and the new user photos will then be tested against each other using the facial recognition technology to determine an overall confidence score of the user specifically for this modality. As facial recognition technology is a very proven method of authentication, this will count more towards the overall confidence score than other modalities.

3.1.1.4 Keystroke Analysis

The second biometric modality of authentication for the product will be keystroke analysis. This will use a machine learning model with binary classification to determine whether the keystroke pattern of the attempted authentication attempt is consistent with what previous successful authentication attempts. Based off the binary classification prediction, a confidence score is then generated to determine whether the authentication is legitimate or not. However, due to the small sample size until consistent use and integration, this will not generate consistent scores. This will count towards less of the overall confidence rating due to this.

3.1.2 Desirable Requirements

3.1.2.1 Face Liveness

This is the third modality of authentication. This would take the image that has been provided during the login system to detect whether the image provided is a live image from a feed. This would allow for detection of AI generated images or videos of the user that may have been used to spoof the system.

3.1.2.2 User Management System

This system would allow for administrators within the application to be able to view all current and deleted accounts. They would have a variety of options ranging from deleting an account or recovering them. They would also force a user to reset their password.

3.1.2.3 Password Reset System

This system would be available for all users and would allow upon login to change their password. This system would update their relevant data to remove any regarding previous passwords. This ultimately would greatly reduce the accuracy of the keystroke analysis system. This would be locked behind a minimum confidence score within the application.

3.1.2.4 Photo Update System

This would allow for users to change the photo that is registered in the database for their account. This would be locked behind a confidence score within the application, requiring a minimum confidence score before allowing for the user to change the photo. This would mitigate the risk of a compromised account being able to spoof the account. If the confidence score wasn’t met, they would have to re-authorise or use an alternative third-party authentication method.

3.1.2.5 Accessibility Settings

To ensure that the application is accessible for all range of users, there will be an additional option within the application that will allow for users to change a variety of accessibility options within the application. The main options would be font, font size and theme i.e. light or dark theme.

3.2 Non-Functional Requirements

Non-functional requirements are defined as overall requirements that are not concerned with the specific functionality of the system, but a list of requirements that the application must meet.

- Accessibility: The application should be accessible for all users.
- Accuracy: The application should be able to accurately classify the user’s authentication attempt.
- Documentation: The system should be documented for future development.
- Effectiveness: The application should reduce the intrusiveness of multi-factor authentication without compromising security.
- Flexibility: The user should be able to have choices with what modalities are in use when authenticating.
- Performance: The application shouldn’t be disruptive to the user and should be quick.
- Testability: All parts of the system should be tested through development.
- Usability: The application is intuitive and user-friendly for all skill levels.

3.3 User Stories

User stories are generalised explanations of features for an application written from the perspective of a user (Atlassian, *User stories: Examples and template*). These user stories act as an end goal rather than tasks that can be ticked off in sprints. Using user stories ensures that the product stays focused on the user and allows for end goals to be broken down into smaller, achievable chunks that can be implemented.

3.3.1 User

- As a user I want to be able to enrol within the programme.
- As a user I want to be able to sign in to the programme.
- As a user I want to be able to view the progress of the authentication.
- As a user I want to be able to change my password.
- As a user I want to be able to update my photo.
- As a user I want to be able to change the font of the application.
- As a user I want to be able to change the font size in the application.
- As a user I want to be able to change the colour theme of the application.

3.3.2 Admin

- As an admin I want to be able to view all users.
- As an admin I want to be able to delete existing users.
- As an admin I want to be able to recover users.
- As an admin, I want to be able to change the admin state of a user.

4. Legal, Social, and Ethical Issues

4.1 Legal

4.1.1 General Data Protection Regulation

General Data Protection Regulation (GDPR) is the required law as enforced by the European Union (EU) defining how information should be stored, protected, and processed. It applies to any and all applications and services which process or hold data of any EU citizen. If the application would to be used within the EU, all laws must be abided by and considered when developing the application. GDPR follows seven main principles (ICO, 2020):

- Lawful, Fairness and Transparency
- Purpose limitation
- Data minimisation
- Accuracy
- Storage limitation
- Integrity and Confidentiality
- Accountability

The application adheres to these principles by only storing data that is required for authentication within the system. Images should either be encrypted when stored in the database, or image vectors should be used to mitigate the risk of personally identifiable information to be leaked. As the keystroke data is not considered personally identifiable, this data is not required to be encrypted.

When a user is deleted from the application, to ensure that the company can be keep records of the user, they are not permanently deleted from the application. Passwords are hashed using NIST recommendations to ensure that the credentials stay secure.

GDPR also outlines the requirement of having a Data Protection Officer (DPO) who will be responsible for data usage and will be the point of contact for any freedom of information requests and queries. However, due to the limited size of the application, a DPO is not currently required.

4.1.2 Licensing

Throughout development of an application, it is a requirement as a developer to ensure that all external assets are either public use or have permission for use. All libraries, code and assets within this application all have free use licensing for use on personal projects, A full list of resources can be found in Appendix A.

4.1.3 Computer Misuse Act

The application fundamentally embodies the computer misuse act to prevent unauthorised access. As explained in section 1 of the computer misuse act, a person is guilty if the actor attempts to access that is intended to secure is unauthorised (UK Public General Acts, 1990). The use of this application minimises the risk of unauthorised access to a user’s use case, while the unauthorised use of biometrics to attempt to gain unauthorised access also would incriminate the adversary with identity theft.

4.2 Social and Ethical

As the application is centred around authentication, there is a requirement that the information that the user enters is stored securely and appropriately. In accordance with GDPR, the user will have the right to erasure, and it will be a requirement that the user is capable of having all data regarding them in the system removed.

The use of biometrics with a machine learning model is also a matter to consider. With AI being currently in mainstream media, there is an air of doubt about the use of AI for processing individual’s data. This worry changes how the machine learning model works, ensuring that for each authentication attempt the model begins anew, and disposing of any previous learning. This ensures that the model is not being used for deep learning, but only to be used for the binary classification with each individual training set.

Lastly, the ethical implications of using facial recognition without the individual user's consent for each authentication attempt. This would be a privacy violation as the user has the right to withdraw at any time. This is incorporated into the application by allowing users to authenticate themselves still successfully without the use of a video device.

5. Method of Approach

5.1 Agile

The project has adopted the agile methodology to ensure that development is completed within the required timeframe and to review progress of the project. There are two main forms of agile development, Scrum and Kanban. Scrum divides development into fixed-length iterations known as sprints, and each sprint has a set of prioritised list of user stories and sprints that are required to be completed. Each sprint follows the cycle of planning, executing and evaluations Atlassian, *What is agile?*). Kanban doesn't work in sprints, but instead works with a visual board (frequently known as a Kanban board) which allows for continuous improvement to tasks within the project and ensuring that workflow is optimised by limiting the number of tasks that are work-in-progress.

5.2 Scrumban

While there are two key methodologies of agile, the project will adopt the intermediate methodology, Scrumban. Scrumban combines the best features of both Scrum and Kanban into a hybrid project management framework. The result is a flexible approach to development due to the range of agile tools that are accessible (Atlassian, *Scrumban: Mastering Two agile methodologies*).

From the scrum approach, we are adopting the use of sprints, also using the sprint planning and retrospectives. From kanban, we are adopting the use of a visual board to keep track of development as well as limiting the number of work-in-progress tasks. The use of a visual board allows for development to be kept accurately tracked of with the use cards including descriptions of tasks and any iterative developments throughout each sprint. The sprint plans can be found in Appendix B.

5.3 Technologies Used

This section will discuss the varying technologies that have been used to meet the techniques that are required for agile development to be successful.

5.3.1 Trello Board

From the benefits of using Scrumban comes the development board from Kanban. To ensure that the project stays organised and to ensure that we limit work in progress tasks, a Trello board was used. Trello is an online visual tool to ensure project-space organisation is well managed and can be easily interpreted. The board is easily able to inform what is in the backlog, what is currently being worked on and what has been completed (Trello, 2023).

The board used was divided into five sections:

- To Do: This acted as the product backlog.
- Doing: This acted as which tasks were being performed during the sprint or required future work.
- Testing: This acted as a section for when tasks were fully developed but required testing to ensure quality.
- Done: This section held every task that had been completed and fully tested.
- Other Tasks: This section contained all non-functional and desirable features.

Figure 2 shows an example portion of the Trello board.

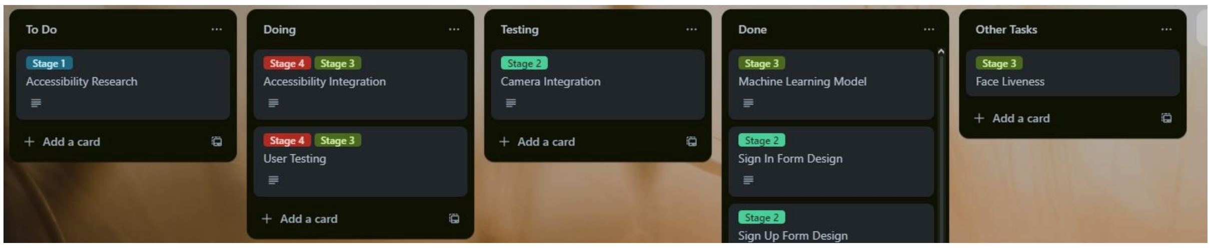


Figure 2 Example of Trello Board

Each kanban card was assigned a label associated with a stage of development. This allowed for consistent review of any work that was behind schedule and what needed to be progressed. With the visual prominence of each column, this helped ensure that the work-in-progress limit was enforced and regularly reviewed.

5.3.2 GitHub

Due to the complexity of the project, it is vital that a method of version control is implemented before any development began. Using version control is industry standard as it allows for larger teams to tackle a project. It includes a complete long-term change history of every file that is within a repository (Atlassian, *What is version control*), along other benefits such as branching and long-term change history of every file.

The product's codebase has been maintained using Git Source Control Management (SCM) and is hosted on GitHub. Having the codebase hosted on an SCM allows for the prevention of loss of data in event of local hardware error or corruption. With the GitHub being a centralized repository, this allows for the access and continued development of the project in event of significant hardware failure.

To ensure that very little is lost in event of hardware failure or corruption, there should be frequent commits to the codebase; allowing for the developer to roll back to previous versions, allowing for quick recovery. GitHub also allows developers to create individual branches from the master branch which allows for individual features to be developed on their own branch without risking the lack of a working solution on the master branch. Finally, GitHub allows for easy management of documentation for the project, allowing for varying files that will be required for the overall project rather than just the codebase itself. This allows for readme files for information regarding the project to be easily displayed for potential users.

6. Design and Architecture

The product is designed with the user experience at the forefront of the design and limiting the intrusion of the multi-factor authentication.

The strongest method of authentication that we can provide is biometrics due to the uniqueness for every individual. It is a challenge for adversaries to spoof biometrics due to them being core to the individual/user and would require excessive reconnaissance and information gathering for the adversary to be able to spoof these biometrics.

6.1 User Interface Design

As the application is an alternative implementation of a system that is used multiple times daily by users, it was important to the design of the application that it followed Human-Computer Interface (HCI) principles to ensure that the application doesn't cause user frustration while maintaining accessibility. This will be implemented by allowing for users to choose between a light and dark theme and allowing for users to choose more accessible font families and font size for the individual.

The eight golden rules of interface design that will be followed were researched (Schneiderman et al., 2017) and are globally followed by designers. The principles being followed are:

- Strive for consistency.
    - Use familiar icons, colours, and user-flows that the user may be familiar with.
- Enable users to use shortcuts.
    - With frequent use, the desire to use quicker methods for completing tasks.
- Offer informative feedback.
    - Ensure that users are aware of what is required of them at all stages of use.
- Design dialogue to yield closure.
    - Ensure that all dialogue provided to the user ensures the user's understanding of what has occurred.
- Offer simple error handling.
    - Ensure that remediation actions are accessible and easy to understand to all end users.
- Easy reversal of actions
    - After a user has completed any action, there should be an obvious but unintrusive option to reverse their action.
- Support internal locus of control.
    - Ensure that the user feels like they are making the application progress. This is done by making sure there are no redirects within the application without control from the user.
- Reduce short-term memory load.
    - Ensure interfaces are simple with proper information hierarchy for quick recognition.

Due to the limited scope of the application, not all the principles will be as applicable as others. The main principles that will be implemented throughout design is consistency, informative feedback, simple error handling and short-term memory load.

6.2 Design Principles and Patterns

The product will follow a handful of design principles and patterns throughout development. Following design patterns allows for the developer to keep code readable and expandable, using proven methods of development rather than requiring the developer to start from scratch.

6.2.1 Observer Pattern

An Observer pattern allows for a subscriber to register and receive notifications from a provider (IEvangelist, *Observer design pattern - .NET*). This will be used frequently throughout the programme to update any interfaces that are gathering information from another form. This will mainly be demonstrated on our real-time form where it subscribes to the login page for information about where in the login process it is.

6.2.2 Singleton Pattern

The singleton design pattern allows for only a single instance of the class to ever be created at once. This ensures that during authentication, there is only ever one user that is stored as the current instance. It also provides a global access point to the instance to ensure that all child processes can access the class. This also ensures that there is no chance of duplicate forms being open, causing problems with webcam feeds.

6.2.3 Strategy Pattern

This design pattern is commonly utilized within software development that enables the definition of groups of algorithms that users are able to select during run time (Tanrıverdi, 2024). Our implementation won't be directly specified by the user, but by what biometrics have been input to us. The key one will be facial recognition, as it is not a requirement and is hardware dependent on the user.

6.3 Database

The database component of the application has been implemented using SQL Server hosted on Amazon Web Services (AWS) Managed Relational Database Service (RDS). An alternative solution for hosting was Microsoft Azure due to their free services as a student, however AWS was selected due to multiple reasons:

- Centralisation – As other aspects of the project will be using AWS; it was deemed more efficient to keep the services centrally hosted rather than using varying services.
- Cost – AWS offers a generous free tier, allowing for the use of a large majority of their services.
- Reliability – With AWS being such a notable company, they have strong incentives to comply with the CIA triad and strong data security to comply with other regulatory requirements.

The use of RDS allows for an always=accessible database that is hosted by a consistent company, allowing for workflow to continue from any device the developer works from.

The design of the database queries was majorly encapsulated using stored procedures, due to their accessibility and security. Using stored procedures, we can reduce the attack surface due to pre-defined logic. Furthermore, it helps prevent SQL injections due to the parameterising of the SQL queries.

6.4 Application

The graphical user interface (GUI) that has been developed for this product has been implemented in .NET with Windows Forms. This also allowed for better hardware utilisation when the application is executed and for a strong integration of a separate Python subsection of code for the machine learning aspect of the project. Python was selected as the machine learning language due to the large number of libraries and documentation available for machine learning, as well as its easy integration into the project. Figure 3 below shows the class diagram of how the overarching system interacts with each other.

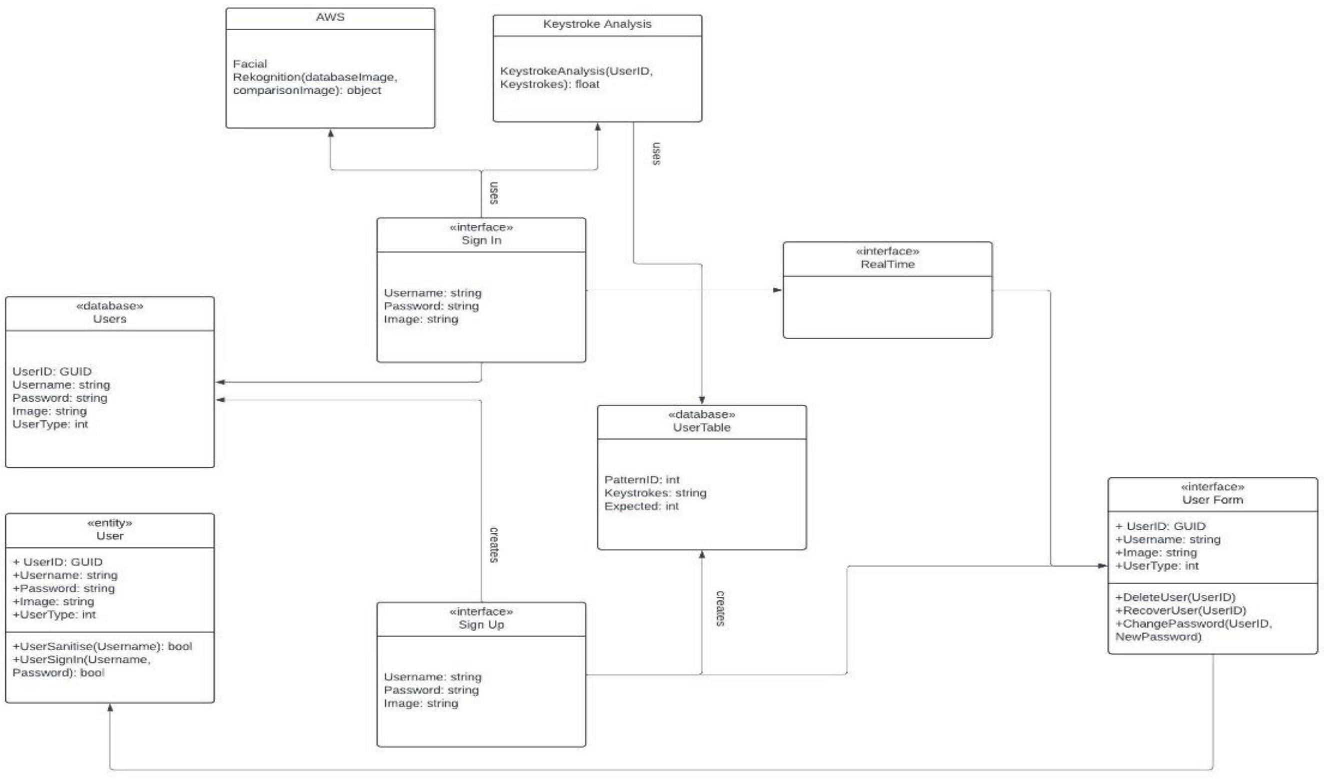


Figure 3 Architecture Interaction

6.5 Security

Due to the handling of personally identifiable information (PII), it is of upmost importance that all data is handled security and with careful consideration to ensure compliance with GDPR.

The main two areas of security that need to be handled are user's passwords and user images. To ensure password security, a strong hashing algorithm should be implemented to minimise the risk of a password being cracked in case of a data leak. At a bare minimum, SHA256 should be used to hash the passwords as advised by the National Institute of Standards and Technology (NIST) (Computer Security Division, 2015). This can be integrated easily within the project to ensure that the passwords are not stored in plaintext. However, SHA256 has many drawbacks such as the lack of salting. Alternatively, a more suitable method of hashing would use an algorithm such as Argon2. Argon2 is a hashing function that won the password hashing competition in 2015 and uses memory-hard functions, specifically designed to hash passwords (Dinu et al., 2015).

Secondly, how the images are stored is a requirement to explore due to the inherent flaws in storing an image in base64. To combat this, rather than storing the image in the database, feature vectors can instead be stored within the database and used for future comparison. Instead of using a facial recognition library/service, an alternative can be used to analyse the initial sign-up image and return the feature vectors, which can then be stored. This process can then be repeated and used for comparison.

7. Implementation

7.1 Research

7.1.1 Accessibility

The graphical user interface (GUI) will be the prominent use case for the entire prototype application. This requires generous research into user design (UX) and how to make it accessible. The main implementation of accessibility that the product is using is font family and font size. The main accessibility option that this is tackling is those with reading difficulties such as poor eyesight and dyslexia.

While searching for dyslexic friendly fonts, OpenDyslexic was a prominent typeface that has been based on research and had had various studies completed on it. This font is an opensource typeface that is designed against common symptoms of dyslexia by using heavy bottom design (OpenDyslexic, 2024). Having this implemented into the application allows for dyslexic users to be able to have an easier time interacting with the application.

As researched by Lexdis, a subsection of ECS Accessibility Team at the University of Southampton, copying codes from one device to another can become problematic where concentration or attention is an issue, as may happen with ADHD or other neurodivergent users (Draffan, 2022). Actions that require a user to recognise and copy a code qualify under the W3C WCAG cognitive function test, therefore requiring an alternative method of authentication. By not using this method of authentication in the solution, this limits the risk of attention loss when authenticating.

Based on the research, having a simple design UI with a variety of font offerings, while also ensuring that the user can change the font size (as generally [web] applications should have at least minimum size of ten) allows for a more tailored experience. While also incorporating everything that is needed within the application reduces the risk of losing attention of users, taking advantage of the concentration to conduct another type of authentication.

7.1.2 Machine Learning

As the developer was not very familiar with the use and implementation of machine learning, a section of research had to be done into the different methods that can be used for the design of the application. There were a variety of different methods of machine learning that can be applied for binary classification. Binary classification is a task to classify input data into two mutually exclusive categories (Keita, 2022). In this case, we are defining our expected data (legitimate activity) and our poisoned samples (not legitimate activity). The model will then be trained each authentication attempt to classify the new input from the user.

While there are several different methods of classification that can be used, the developer chose to use random forest classification. Due to the limited sample size that will be provided to the model until consistent use, using a random forest classifier allows for improved accuracy by reducing overfitting (Al-Rousan, 2022). Ensuring that the model is not overfitted can result in the training data being poisoned with noise and outliers, and therefore may perform poorly on newer data. It is a necessity that this isn’t implemented due to new data being core to the analysis.

7.1.3 Facial Recognition Technology

As there are a variety of different options available regarding facial recognition, research into which solution would be the most accessible had to be conducted. While there are several libraries available for .NET, such a AForge and DlibDotNet, the developer decided to use an external service to ensure that the function would be completed within the timeframe provided. The options that were researched were Amazon Facial Rekognition, Microsoft Azure Face and Google Cloud Vision. However, the developer chose to use Amazon Facial Rekognition to keep consistency with database development. Keeping the application centralised allows for any future development to become modular and less confusing for an additional developer to transition to another service if so desired.

However, using Azure Face API may have been a more beneficial selection due to it being generally more accurate and reliable when using facial analysis/recognition tasks.

7.1.4 GUI Technology

The application could have been built in a variety of different methods. Due to the developer's experience, the code was to be written in either .NET or Python. By using Python, the entire product could have been kept within one coding language. However, to create GUIs there are multiple different libraries that are required to be imported and has a lack of visual development without compiling. Due to this, the developer chose to use Windows Forms, a built-in functionality within Visual Studio that has a visual designer that allows for real-time editing and properties changing without the need of compiling.

However, the challenge with using Microsoft Forms is the lack of creative direction you can implement using the base product. Research into alternative libraries that can be used were available such as MetroFramework.

MetroFramework is a community created library that brings UI updates from Windows 8 to .NET Windows Forms applications. It allows for greater customisation, and therefore the ability to make the application more accessible. It allows for custom colours and the integration of a light and dark mode. Being able to use custom hex colours will allow for us to ensure that the colour pallet used meets the requirements from Web Content Accessibility Guidelines (WCAG). While our application is not web-content based, the same guidelines can be applied to our application. This ensures that the contrast between displayed text and the background can still be ready by those with moderately low vision (*Understanding SC 1.4.3:contrast (minimum) (level AA)* 2023).

7.2 Application Setup

The second stage of development was to begin the setup of the environment and other technologies being used to begin development.

7.2.1 Database

The database began by being held on a locally hosted SQL database. The initial creation was just a simple users table that took in a user's UserID, Username, Password, and Image. This had all been documented into a Jupyter Notebook for ease of transition and understanding; the notebook has been updated throughout development to ensure that the end product can be met.

The second table that had been developed was the Keystrokes table. On sign in, a user's recorded keystroke pattern would be stored in this table along with their unique identifier (UserID). Further into development, this table had been depreciated in favour of using dynamically named tables.

7.2.2 Sign Up and Sign In Functionality

This stage of development created a basic enrolment form within the application. Any images that will be integrated into the function will be stored as base64 strings and all other user information is passed into the database using a stored procedure. This will then create a table named after the user, storing all future keystroke patterns that will be used for the keystroke analysis. Figure 4 shows a flowchart of the interaction of sign up programme when camera integration is implemented:

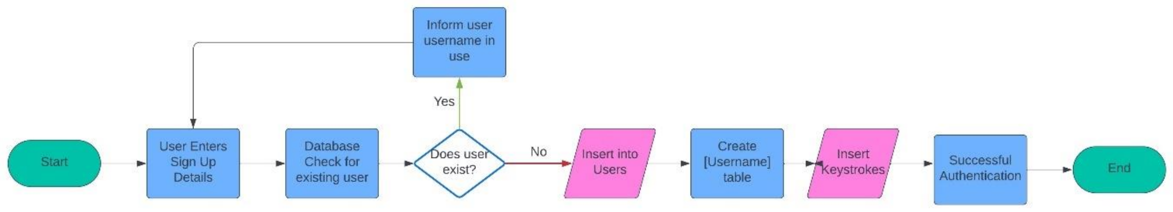

Inform user username in use
Yes
Start
User Enters Sign Up Details
Database Check for existing user
Does user exist?
No
Insert into Users
Create [Username] table
Insert Keystrokes
Successful Authentication
End

Figure 4 Sign Up Flowchart

The next function will be the sign in function. This takes in a user's username and password, and if a video device is detected, it takes up to six photos of the user. The images are taken automatically by the application while the user is typing in their password due to already having their concentration, therefore removing an additional step during the authentication process. This also gives more reliability when using the facial recognition function, as there are more samples to compare. Figure 5 shows a flowchart for the sign in feature.

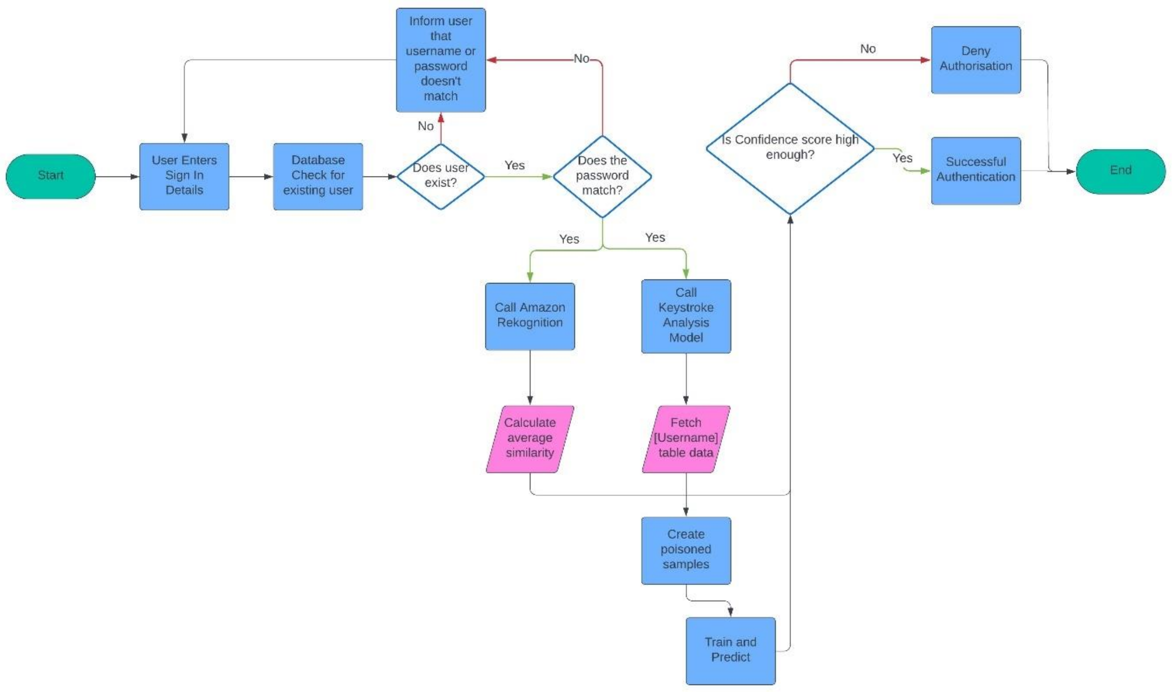

Start
User Enters Sign In Details
Database Check for existing user
Does user exist?
No
Inform user that username or password doesn't match
Yes
Does the password match?
No
Yes
Yes
Call Amazon Rekognition
Call Keystroke Analysis Model
Calculate average similarity
Fetch [Username] table data
Create poisoned samples
Train and Predict
Is Confidence score high enough?
No
Deny Authorisation
Yes
Successful Authentication
End

Figure 5 Sign In Function Flowchart

To ensure security within the application, user inputs (that aren't passwords due to being hashed) are all validated using regex patterns against common patterns such as SQL Injection and Path Traversal. In the case of pattern detection, this will prevent any communication with any other parts of the application.

7.2.3 Video Feed

Before completing the sign-up function, the ability to display a webcam feed was a necessity. This used external library AForge Video to allow the developer to capture any video devices connected to the device and streamed this to a PhotoBox. The challenge that came with using a PhotoBox was the video stream was not resized to fit within the container. This was implemented by creating a duplicate feed, calculating the aspect ratio and resolution of the video stream versus the PhotoBox, and therefore resizing it. This functionality is repeated in both the Sign Up and Sign in feature.

Once the user had taken their photo, to optimize the programme, the stream is then stopped and instead displayed with photo that had been taken.

7.3 Authentication Functionality – Core Functionality

The third stage of the implementation was the various authentication modalities. This part of development took the largest allocation of time due to the challenges that occurred during development.

7.3.1 Facial Recognition Implementation

The implementation for the Facial Rekognition API wasn't a challenge. This took the existing database image and the several new images for a comparison. This function is then called for every image that has been taken. The API will then return a similarity percentage. An average is then calculated for the sum of the images to further validate the images that had been taken.

There is an editable parameter that can be customized for each individual for a "SimilarityThreshold". This parameter determines the response value, the developer has this currently set to a value of zero to ensure that a response is always given. This will be discretionary to the company, as this could be editable to always assume a constant value if the similarity threshold is below what is requested by the access control.

7.3.2 Keystroke Analysis

The keystroke analysis section is by proportion a lot larger. Due to the limited training samples that are available, the function generates its own poisoned samples using standard variations and averages of existing keystroke pairs. This calculation therefore produces a variety of poisoned samples that are similar enough to legitimate data, but with slight variation to teach the model that it should be categorised as unexpected.

To ensure that there is no data imbalance, the number of poisoned samples generated are directly proportional to the amount of legitimate data. Ensuring there is a data balance between legitimate and poisoned samples mitigates the risk of bias during the training process and increases the model accuracy.

In the model, there are a variety of parameters that can be edited to be tailored more to company requirements. The parameters that can be edited are:

- A randomly generated number of standard deviations added to the average (currently between -2 and 2)
- A randomly generated amount of noise that is added (currently between -20 and 20).

Each keystroke pair is then calculated using the formula in Figure 6:

$$NewPattern = ColumnAverage + (RandomNumber * StandardDeviation) + noise$$

Figure 6 Keystroke Set Calculation Formula

When enough new keystroke pairs have been generated, these are concatenated together to generate a new keystroke pattern. Once our poisoned sample has been generated, it is then appended to the data set with a poisoned sample label to be passed into the machine learning model. Once it has completed its training, the model then predicts whether the current attempt is consistent with its training using binary classification. This returns a confidence score on whether it is either legitimate or not, returning to the application.

The first challenge faced within the model was ensuring that integration into the other files were successful, as the outputs that were returned needed to be manipulated to ensure consistent results. The second challenge was fine-tuning the correct level of parameters for the model to ensure accurate predictions without taking a large amount of time, making the user feel frustrated.

7.3.3 Confidence Score

The finalised confidence score generated will be a weighted calculation of the predictions given by the biometrics.

Due to the keystroke analysis having less accurate predictions with smaller data sets, this will be considered a less viable biometric and therefore have less weighting when calculating an overall confidence score. However, as the application is used for longer and the data set grows larger, this shouldn't reduce the weighting by a detrimental amount. Research has shown that while keystroke analysis is not a unique identifier, it is able to provide discriminative information, therefore providing valuable information for authentication purposes (Saevanee et al., 2012). With this reasoning, the current confidence weighting is:

Facial Recognition	60%
Keystroke Analysis	40%

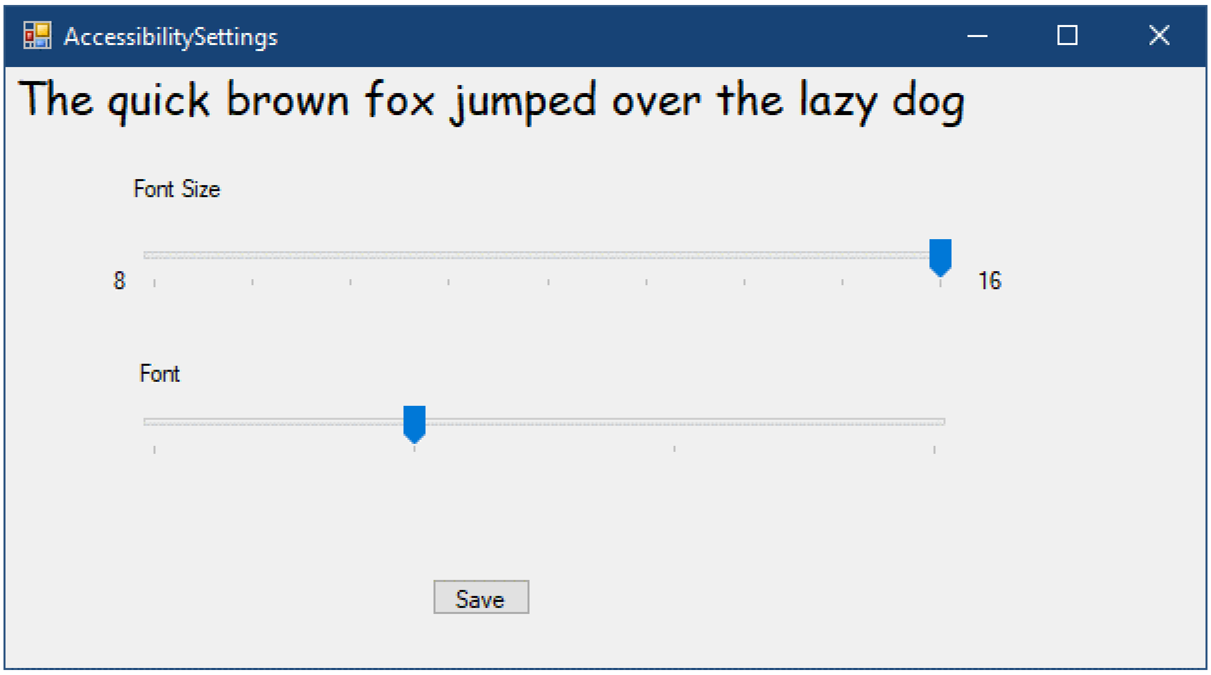
This confidence score is then stored in a session class. This can then be used for further access control within the organisation's systems. An example of this can be found in the AdminForm interface, where administrators are only able to edit user information if their overall confidence score is > 90%. This is an adjustable constraint.

To ensure that the acceptance rate corresponds with the company's needs, the confidence accepted score should be edited in according to their needs. However, the score should be personalised to ensure that both the false acceptance rate isn't too high; but ensuring that the false rejection rate (FRR) isn't as high as to interfere with legitimate users. For a consistent rate, the equal error rate (EER) should be used. The EER is where the both the percentage for the false FAR and FRR are equal (Veenhof, 2023).

7.4 Accessibility Integration

To allow the user the ability to customise their experience with the application and to ensure accessibility when it comes to reading, a separate interface has been developed to allow the user to experiment with different fonts and font sizes. The application utilizes TrackBars to allow the user to drag between font sizes 8 and 16, while selecting from a variety of different fonts. The fonts offered are:

Microsoft Sans Serif, Comic Sans MS, OpenDyslexic 3 and Impact. Sans Serif fonts are generally considered an accessible font and is standard amongst varying different applications. Comic Sans MS is a font that has been used within education to accommodate those with reading difficulties such as dyslexia and is one of the many fonts that is used for inspiration for OpenDyslexic 3. The use of OpenDyslexic 3 has been previously discussed, but the use of bottom-heavy characters is designed to give easier readability. Lastly, Impact was the last font selected due to the boldness of each letter, making it larger and bolder for those who are hard of vision to use. There is a sample text on the page for the user to decide which of the alternatives they would like to use, as shown in figure 7:



The quick brown fox jumped over the lazy dog

Figure 7: Example Accessibility Settings Page

The developer was unable to integrate MetroFramework successfully into the project. Despite the controls being accessible, and visible, they were unable to overwrite properties of each control. For example, a label could not have the size of the label or the font colour changed. It also didn't allow for using custom backgrounds for labels or text boxes, meaning that if a custom background colour was used, there would be white backgrounds around all labels and text boxes, which visually didn't contrast well.

7.5 Maintenance and Report

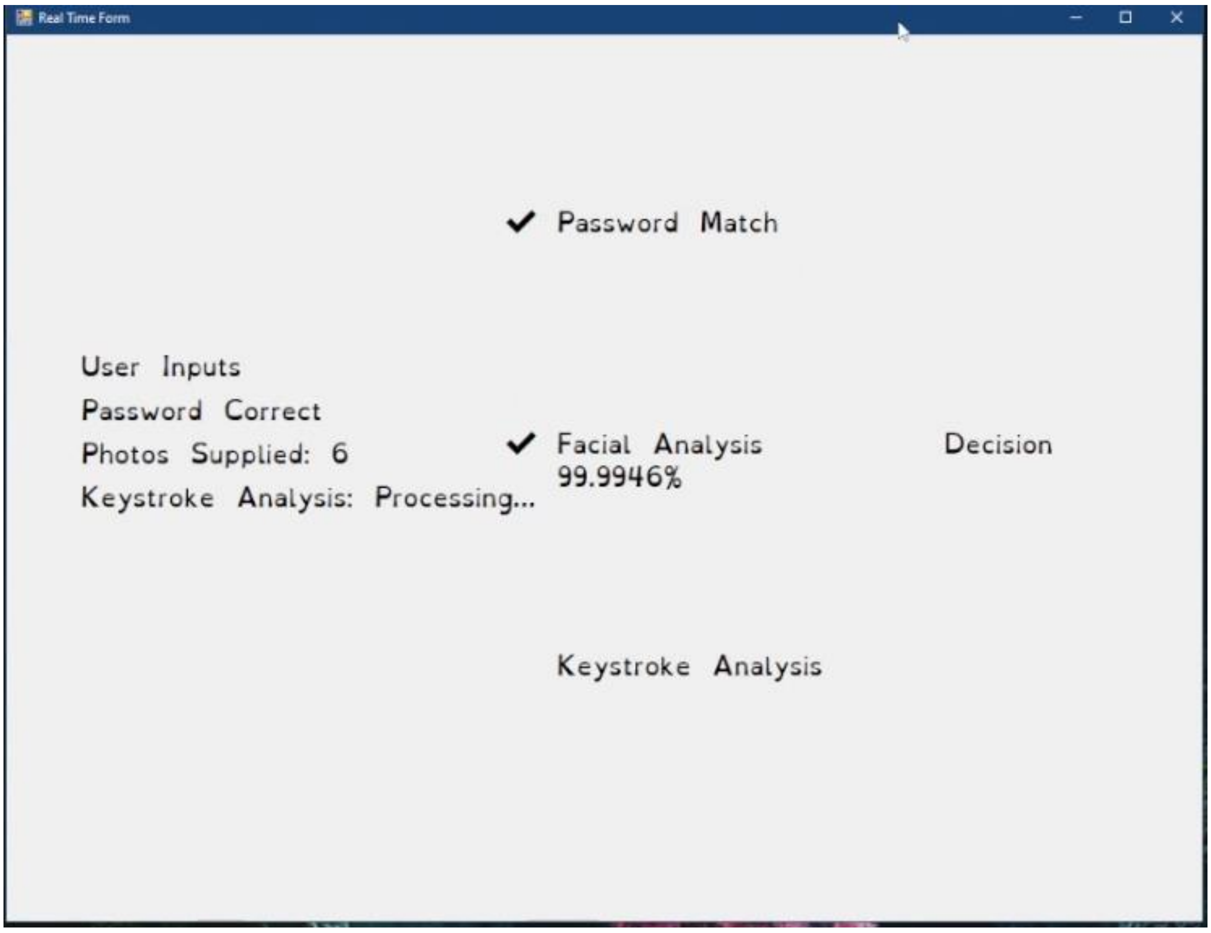
The final stage of the project was used to ensure that user testing results had been implemented, with a key focus on bug fixing and quality of life changes within the application. Once the fixes had been implemented, focus centred around writing the report. This took a large amount of time as it required the centralisation of varying resources that had been throughout the whole of development. There were no challenges faced during this stage of the project.

8. Testing

8.1 Test Driven Development and Integration Testing

To ensure that an effective approach towards development that was robust and well tested, the developer adopted a Test Driven Development (TDD) approach. This allows for an iterative development that uses precise tests to ensure the application performs as expected. Once the code provides successful outputs, refactor the code to ensure that it is optimized and redundant (Unadkat, 2023). While this is traditionally done with unit tests, the approach used console outputs for constant checking that values are as expected.

While this was used for tests throughout the application's development, specifically for integration of the authentication functions, Integration Testing was used to focus on the interaction of the functions overall impact on the applications performance. The result of this testing ultimately changed how the application ran during the authentication process – by instead running the functions as asynchronous tasks, this allowed for the application to continue being interactive for the user instead of locking them out, seemingly as if the application had crashed. The implemented interface can be seen in figure 8.


Real Time Form

✓ Password Match

User Inputs
Password Correct
Photos Supplied: 6
Keystroke Analysis: Processing...

✓ Facial Analysis
99.9946%

Decision

Keystroke Analysis

Figure 8:: Real Time Form

8.2 Functional Testing

To ensure that each function performed as expected, a black box approach was taken to test each subsection of the system. This allows for a versatile approach to testing to ensure that the product works as intended for the end user, rather than how the source code is expecting. Functional testing was used throughout development of the product to ensure that features worked as intended. Feedback provided resulted in quality-of-life changes within the project for the end user. An example of this would be a user managing to crash the programme by pressing backspace too many times during a password input.

To complete functional testing, a plan had been devised for each and all functional requirements. The full set of functional testing can be found in Appendix C.

9. End of Project Report

9.1 Summary

Overall, the project was completed successfully. The project was able to provide a non-intrusive multi-modal authentication application that was consistent and accurate. Although there were challenges along the way regarding interface design and the creation of the machine learning model which caused the product to not be on schedule, all core and many desirable features were implemented, and all product objectives were achieved.

9.2 Objectives Review

1. **To research user experience (UX) interface design to allow for the application to be accessible to all users.**

This was researched and ideas were implemented into the application, however this objective wasn't fully met. The user interface (UI) had a basic design with a monochrome colour palette with no option of a light/dark mode. However, the implementation of allowing the user to customise the font size and family resulted in a partially met objective.

2. Build an application that uses multiple biometric modalities to authenticate.

This objective was considered a success. The application uses both image recognition and keystroke analysis to devise a confidence score based on the login attempt. Face liveness ultimately wasn't implemented into the application due to further research discovering that there were a series of prerequisites that proved intrusive to users.

3. Build an application that allows for enforcement of access control for each individual authentication attempt.

This objective is considered a success – but mainly based on how a function returns the confidence score. The programme currently only returns confidence in the prediction which is two numbers that add up to a total of one, the greater number being either a zero for success or one for failure. This can be manipulated in future development and integration.

4. Design a user interface that is accessible and intuitive.

This objective has been partially met. While the interface has been designed to be accessible using allowing users to edit key aspects such as font size and font family, the visual design of the application is basic. This is due to the technologies that had been chosen to develop the application, and many libraries that are used to re-design the interfaces costing money. Although there was a library that was used, it wasn't consistent with how it applied settings, and therefore was scrapped for a more basic design.

5. To follow agile project management techniques and complete a minimum viable product within the allocated time.

Overall, the objective should be considered a success despite setbacks. Many sprints were behind schedule or not worked on when scheduled. However, the minimum viable product was still completed during the allocated time and completed in an agile and iterative manner, while being version controlled. Albeit there were many core techniques to agile development that had been overlooked, and for future development or to start from scratch, would be incorporated for a more structured approach.

6. Create an unintrusive method of authentication to ensure users aren't disrupted.

This objective is a complete success on sign up. The user follows a basic sign-up and sign-in process no different to a regular sign-up form to any application or web application. The design currently displays the image of the user while it is being taken as a way of alerting the user that their camera is in use.

9.3 Changes in Development

The main changes in development were database hosting, while for a large majority of the project, the database was hosted with AWS, due to unforeseen circumstances it had to be partially locally hosted during some parts of development. This took a chunk out of development at times to update the correct database at time of use.

This generally was taken care of using a Jupyter Notebook, however with micro changes throughout development and bug-fixing, not all were documented fully.

The addition of a real-time form during authentication was a change that was also made during development. This has been updated in the user stories. This was based off feedback that during the authentication process, the user was unsure whether anything was happening or if the application crashed.

10. Reflection

10.1 What Went Well?

The integration of the various biometrics into the application worked smoother than expected, with the developer not having experience in interweaving multiple code files that perform different functions together. Furthermore, the accuracy of the machine learning model also proved quite high; although this was only noticed when the application had a minimum of 30 keystroke patterns to use. The integration of AWS services are well documented and resulted in easy development and manipulation of results.

The research into accessibility proved quite well as to why the development of this project is mindful and can contribute towards accessible authentication in the future. Ensuring that user's have options that don't fall within the WCAG cognitive function testing allows for a wider community of users to use the application.

Adopting agile as the project management was beneficial despite its limited usage. Using this development approach allowed for flexibility in implementation and allowed for an iterative approach with testing and additional requirements were discovered from this

10.2 Lessons Learnt

Moving forward, the developer would want to design future project's interfaces using a high-level design tool such as Figma. While it is not a mobile application, it functions similarly enough. While the developer had a base understanding of how they wanted my application to look – it would have been easier to experiment with positioning and colour design. This would have also allowed for a more intuitive user testing, while also saving valuable time that could be used for further development.

Furthermore, more research should have been done into the technology used to design the GUI, as while the developer was aware of the limitations for designing the interface, it proved to be overly detrimental to acquiring a good UX design. The developer was unable to get MetroFramework to successfully integrate into the project, preventing the use of custom colours and gentler borders.

The project management left more to be desired, as the developer didn't frequent sprint reviews and often left considerable time between development. If the developer was to continue developing the product, they would ensure to follow correct agile features to benefit from the development cycle.

10.3 Future Development

As the application stands, it is a suitable prototype that acts as an authentication app that is unintrusive and can provide a confidence score based on the biometrics provided. For future development:

- Add in an integration to any devices with fingerprint sensors as an optional for the user to potentially use for passwordless login.
- Add in more ways of experimenting with confidence score – allowing for a more human-approachable side of seeing Low, Medium, or High.
- Allow for admins to change passwords of users.
- Ensure that the Singleton design pattern is integrated to prevent webcam feed issues.
- Allow for users to retake their photo during sign up.
- Incorporate more information from keystroke dynamics, including length of key press.

11. Conclusion

The aim of the project was to create a form of authentication that fuses several biometrics to enforce access control, while also being non-intrusive to the end user. The project achieved all but one of the project objectives that were outlined but met all functional requirements. Although there were various setbacks during development, the project managed to finish as a success. The developer had never dealt with a project this large using various unfamiliar technologies and has considered the project a success. For future development of the project, it is hoped that other various types of biometrics could be included, such as speech recognition and fingerprint identification which can be added optional authentication methods for those who are prepared to achieve higher accuracy confidence ratings, while still maintaining its unintrusive nature.

12. References

- Atlassian (2024) *Scrum sprints: Everything you need to know*, *Atlassian*. Available at: https://www.atlassian.com/agile/scrum/sprints (Accessed: 09 April 2024).
- Atlassian (2024) *What is agile?*, *Atlassian*. Available at: https://www.atlassian.com/agile (Accessed: 09 April 2024).
- Atlassian (2024) *User stories: Examples and template*, *Atlassian*. Available at: https://www.atlassian.com/agile/project-management/user-stories (Accessed: 18 April 2024).
- Atlassian (2024) *What is version control*, *Atlassian*. Available at: https://www.atlassian.com/git/tutorials/what-is-version-control (Accessed: 18 April 2024).
- Atlassian (2024) *Scrumban: Mastering Two agile methodologies*, *Atlassian*. Available at: https://www.atlassian.com/agile/project-management/scrumban (Accessed: 19 April 2024).
- Al-Rousan, T. (2022) *(open access) new framework for improving Random Forest Classification Accuracy (2022)*, *SciSpace - Paper*. Available at: https://typeset.io/papers/new-framework-for-improving-random-forest-classification-1zs85mbp (Accessed: 19 April 2024).
- Computer Security Division, I.T.L. (2015) *NIST policy on hash functions - hash functions: CSRC*, *CSRC*. Available at: https://csrc.nist.gov/projects/hash-functions/nist-policy-on-hash-functions (Accessed: 19 April 2024).
- Dinu, D. *et al.* (2015) *P-H-C/PHC-winner-argon2: The password hash argon2, winner of PHC*, *GitHub*. Available at: https://github.com/P-H-C/phc-winner-argon2 (Accessed: 20 April 2024).
- Draffan, E.A. (2022) *Multifactor Authentication – Cognitive Overload?*, *Lexdis 2.0*. Available at: https://www.lexdis.org.uk/2022/02/multifactor-authentication-cognitive-overload/ (Accessed: 18 April 2024).
- ICO (2020) *A guide to the data protection principles*, *ICO*. Available at: https://ico.org.uk/for-organisations/uk-gdpr-guidance-and-resources/data-protection-principles/a-guide-to-the-data-protection-principles/ (Accessed: 22 April 2024).
- IEvangelist (2024) *Observer design pattern - .NET*, *.NET | Microsoft Learn*. Available at: https://learn.microsoft.com/en-us/dotnet/standard/events/observer-design-pattern (Accessed: 19 April 2024).
- Keita, Z. (2022) *Classification in machine learning*, *DataCamp*. Available at: https://www.datacamp.com/blog/classification-machine-learning (Accessed: 19 April 2024).
- Lei, Z. *et al.* (2021) *On the insecurity of SMS one-time password messages ...*, *On the Insecurity of SMS One-Time Password Messages against Local Attackers in Modern Mobile Devices*. Available at: https://www.ndss-symposium.org/wp-content/uploads/ndss2021_3B-4_24212_paper.pdf (Accessed: 18 April 2024).
- OpenDyslexic (2024) *OpenDyslexic*. Available at: https://opendyslexic.org/related-research (Accessed: 09 April 2024).

- Reese, K. *et al.* (2019) *A usability study of five two-factor authentication methods, A Usability Study of Five Two-Factor Authentication Methods*. Available at: https://www.usenix.org/system/files/soups2019-reese.pdf (Accessed: 17 April 2024).
- Saevanee, H., Clarke, N.L. and Furnell, S.M. (2012) *Multi-modal behavioural biometric authentication for mobile devices, SpringerLink*. Available at: https://link.springer.com/chapter/10.1007/978-3-642-30436-1_38 (Accessed: 21 April 2024).
- Schneiderman, B. *et al.* (2017) *Designing the user interface: Strategies for effective human-computer interaction, 6th edition, Designing the User Interface: Strategies for Effective Human-Computer Interaction*. Available at: https://www.pearson.com/en-us/subject-catalog/p/designing-the-user-interface-strategies-for-effective-human-computer-interaction/P200000003485/9780137503889 (Accessed: 20 April 2024).
- Tanrıverdi, B. (2024) *Implementing the strategy design pattern with keyed services in .net 8, Medium*. Available at: https://medium.com/@baristanriverdi/implementing-the-strategy-design-pattern-with-keyed-services-in-net-8-9bf6995651d3 (Accessed: 19 April 2024).
- Trello (2023) *What is Trello?: Trello, Atlassian Support*. Available at: https://support.atlassian.com/trello/docs/what-is-trello/ (Accessed: 20 April 2024).
- Twilio, (2024) *One-time passcode verification (OTP)*. Available at: https://www.twilio.com/code-exchange/one-time-passcode-verification-otp (Accessed: 18 April 2024).
- UK Public General Acts (1990) *Computer misuse act 1990, Legislation.gov.uk*. Available at: https://www.legislation.gov.uk/ukpga/1990/18/contents (Accessed: 22 April 2024).
- Unadkat, J. (2023) *What is Test Driven Development (TDD) ?, BrowserStack*. Available at: https://www.browserstack.com/guide/what-is-test-driven-development (Accessed: 21 April 2024).
- Veenhof, J. (2023) *Everything about far and Frr, Recogtech*. Available at: https://recogtech.com/en/insights-en/far-and-frr-security-level-versus-ease-of-use/ (Accessed: 21 April 2024).
- W3(2023) *Understanding Success Criterion 1.4.3: Contrast (Minimum) | WAI | W3C*. Available at: https://www.w3.org/WAI/WCAG21/Understanding/contrast-minimum.html (Accessed: 19 April 2024).

13. Appendix

Appendix A – Technologies Used

.NET: https://dotnet.microsoft.com/en-us/download/dotnet/8.0

AForge Video : https://github.com/andrewkirillov/AForge.NET

Amazon Web Services: https://aws.amazon.com/

Github: https://github.com/

Jupyter: https://jupyter.org

MetroFramework: https://thielj.github.io/MetroFramework/

NumPy: https://numpy.org/

OpenDyslexic: https://opendyslexic.org/

Pandas: https://pandas.pydata.org/

Pyodbc: https://pypi.org/project/pyodbc/

Python: https://www.python.org/

RegEx: https://learn.microsoft.com/en-us/dotnet/standard/base-types/regular-expression-language-quick-reference

SciKitLearn: https://scikit-learn.org/

SQLServer: https://www.microsoft.com/en-gb/sql-server/sql-server-downloads

Trello: https://trello.com/

Windows Forms: https://learn.microsoft.com/en-us/dotnet/desktop/winforms/overview/?view=netdesktop-8.0

Appendix B – Sprint Planning

Sprint 1:

The plan for the initial sprint is to complete the first aspect of the software development lifecycle as well as to conduct research. After researching the background of multifactor authentication and the existing solutions, a requirements analysis should be conducted to provide objectives and functional requirements for the project.

Tasks:

Existing multifactor authentication solutions benefits and limitations

Research into how to make multifactor authentication accessible

Implemented solutions for facial recognition

Technologies for GUI implementation

Trello board creation and implementation

Sprint 2:

The plan for sprint 2 will be for initial environment setup. Ensuring that the coding environment is setup and database creation. This will allow for experimentation with the chosen technologies and for a base startup idea.

Tasks:

Initial database setup and table creation

GitHub repository creation and initialisation.

Experiment with Windows Forms and how the application will function

Sprint 3:

The plan for sprint 3 will be to develop a basic sign up and sign in system. This will be integrated with the database and have basic error checking and user testing to ensure that the system functions as intended. This should also be used to research applicable password hashing algorithms.

Tasks:

Create sign up and sign in forms.

Create relevant tables within the database and keep Jupyter Notebook up-to-date.

Create sign up and sign in function for a username and password.

Sprint 4:

This section will be for external implementation of AWS services and integration of video sources into the project. Research into which .NET libraries can be used to integrate the devices in and get them accurately displaying the video stream. Experiment with using Amazon Rekognition API and the responses that is received into the application. This will involve several table redesigns. Develop the keystroke capture for future analysis.

Tasks:

Integrate video sources into the project.

Allow users to upload a photo of themselves during sign up.

Allow users to upload a photo of themselves during sign in.

Develop functions to capture keystroke pattern data.

Sprint 5:

This sprint will be used to develop the keystroke analysis machine learning model and for integration into the wider system. By the end of this sprint, the authentication modalities should be fully integrated and working cohesively to generate a confidence score.

Tasks:

Create machine learning model.

Integrate facial analysis and keystroke analysis functions into the system.

Ensure that the keystroke model only records accurate keystroke data.

Generate a weighted confidence score according to user inputs.

Sprint 6:

This sprint will be for user testing and the development of accessibility within the application. Due to time constraints, the main focus will be on enabling users to use custom font families and font sizes and having it apply to the whole application.

Further development into the user side of the system allowing for the user to reset their passwords and for admins to have the ability to delete and recover accounts.

Tasks:

Perform user testing on existing system.

Create user accessibility form and integrate custom fonts.

Create user form for when the user has logged in.

Create admin form for when admin has logged in.

Sprint 7:

The penultimate sprint will be used to integrate user feedback and capture bug fixing. The creation of a real-time form so that the user is able to understand what is happening within the application to be created. Implement extra admin functionality to allow for them to change existing user's admin status.

Tasks:

Bug fixing on sign-up and sign-in functions.

Allow for users to change the admin status of accounts, changing their experience after successful authentication.

Create real-time-form for user experience, including the individual confidence score of each modality.

Sprint 8:

The final sprint is dedicated to writing the report for the final submission, creating a poster, and recording a video demonstrating the product. This will be integrating all previous research and testing into one large document discussing what issues had arisen.

Tasks:

Ensure that all project objectives are met.

Ensure all functional requirements are implemented.

Further functional testing on the system to be performed.

Appendix C – Functional Testing

Test	Instructions	Inputs	Result	Expected Result
User register an account without a webcam	1. Click Register 2. Enter Username 3. Enter Password	N/A	User was unable to access sign up page.	As expected
User register an account with a webcam.	1. Click register 2. Enter username 3. Enter password 4. Take photo 5. Submit	Username: “james” Password: “jamesPassword” Photo: user taken	The user was able to create an account	As expected
User register an account that already exists.	1. Click register 2. Enter username 3. Enter password 4. Take photo 5. Submit	Username: “james” Password: “jamesPassword” Photo: user taken	Receives an error saying the username already exists.	As expected
User sign in with a webcam.	1. Click Sign in 2. Enter details 3. Submit	Username: “james” Password: “jamesPassword”	User was taken to the real time form to see where they are in authentication. Successfully logged in.	As expected.
User sign in without a webcam.	1. Click Sign in 2. Enter details 3. Submit	Username: “james” Password: “jamesPassword”	User was taken to the real time form to see where they are in the authentication process. User failed authentication due to confidence score being too low.	As expected.

User re-types their password during sign in.	1. Click Sign in 2. Enter details 3. Submit	Username: “james” Password: “jamesPassword”	Programme crashed.	Programme should progress as normal.
User re-types their password during sign up.	1. Click Sign up 2. Enter details 3. Submit	Username: “james2” Password: “jamesPassword”	Programme crashed.	Programme should progress as normal.
User attempts to sign in with a user that doesn’t exist.	1. Click Sign In 2. Enter details 3. Submit	Username: “james3” Password: Password	User was informed a user could not be found.	As expected.
User create two log in sessions at once.	1. Click sign in 2. Navigate back to previous screen 3.Click sign in	n/a	User was able to open two forms.	Programme should not allow a second form to be opened.
User create two sign up sessions at once.	1. Click sign up 2. Navigate back to previous screen 3.Click sign up	n/a	User was able to open two forms.	Programme should not allow a second form to be opened.
User attempts to reset password where passwords don’t match	1. Click sign in 2. Enter details 3. Submit 4. Enter new passwords 5. Submit	Username: “james” Password: “jamesPassword” NewPassword:” Password” Confirm Password: “password”	User was informed their passwords don’t match.	As expected
User attempts to reset password where	1. Click sign in 2. Enter details 3. Submit	Username: “james” Password: “jamesPassword” NewPassword:” Password” Confirm Password: “Password”	User’s password was reset.	As expected

passwords match	4. Enter new passwords 5. Submit			
Admin logs in with a confidence score < 80	1. Click Sign in 2. Enter details 3. Submit 4. Click user	Username: “alex” Password: “alexPassword”	User was unable to click on a user.	As expected
Admin logs in with a confidence score > 80.	1. Click Sign in 2. Enter details 3. Submit 4. Click user	Username: “alex” Password: “alexPassword”	User was unable to view the user’s form.	As expected
Admin attempts to delete a user.	1. Click sign in. 2. Enter details 3. Submit 4. Click user from left table 5. Click Delete User	Username: “alex” Password: “alexPassword”	User was deleted, form closed and user view’s reset.	As expected
Admin attempts to recover a user.	1. Click sign in. 2. Enter details 3. Submit 4. Click user from right table 5. Click Delete User	Username: “alex” Password: “alexPassword”	User was recovered, form closed and user view’s reset.	As expected
Admin attempts to make a non-admin admin.	1. Click sign in. 2. Enter details 3. Submit	Username: “alex” Password: “alexPassword”	User was made admin, form closed and user view’s reset.	As expected

	4. Click user from left table 5. Click Change Admin			
Admin attempts make themselves a non-admin	1. Click sign in. 2. Enter details 3. Submit 4. Click user from left table 5. Click Change Admin	Username: “alex” Password: “alexPassword”	User’s admin state changed, form closed and user view’s reset.	User should not be able to change their own admin status.