**Technical Manual**

****

CSIT321 Final Year Project

Technical Manual

22 Mar 2021

Project Group No: FYP-21-S1-02

Project Topic: Typing Habit Gesture Authentication System

Project Supervisor: Mr Sionggo Japit

Project Members:

| Name | UOW ID | Email |
| --- | --- | --- |
| Alicia Tan Chun Ying | 6457794 | cyatan005@mymail.sim.edu.sg |
| Terrence Yap Yong Ting | 6573307 | yttyap002@mymail.sim.edu.sg |
| Tng Min Li | 6649737 | mltng001@mymail.sim.edu.sg |
| Chea Cui Hui | 6647546 | chchea001@mymail.sim.edu.sg |
| Joel Teo Han Wen | 6456467 | jhwteo002@mymail.sim.edu.sg |

Project Marketing Website: https://fyp21s102.wixsite.com/fyp21s102

**Technical Manual**

Chapter 1: Introduction........................................................................................................................................................ 5 1.1. Overview.................................................................................................................................................................... 5 1.2. Project Scope............................................................................................................................................................ 5

1.2.1 Project Objective ................................................................................................................................................ 5 1.2.2 Goals .................................................................................................................................................................... 5 1.2.3 Milestones ........................................................................................................................................................... 5

1.3. Project Description ................................................................................................................................................... 5 Chapter 2: Literature Review.............................................................................................................................................. 6 2.1. Target Users.............................................................................................................................................................. 6 2.2 Competitors Analysis ................................................................................................................................................ 7 2.3 Behavioral Analysis.................................................................................................................................................10 2.3.1 Text entry speed: Words per minute .............................................................................................................10 2.3.1 Accuracy / Error rate: ration of incorrect characters to total characters ..................................................10 Chapter 3: Methodology....................................................................................................................................................12 3.1. Software Development Life Cycle Methodology................................................................................................12 3.2. Research Methodology..........................................................................................................................................12 3.3. Data Gather Methodology .....................................................................................................................................12 3.4. Development Methodology ...................................................................................................................................13 Chapter 4: Pre-development Data Collection and Analysis.........................................................................................14 4.1. Final Project Risk Analysis and Assessment ....................................................................................................14 4.2. Final Work Breakdown Structure ........................................................................................................................18 4.3. Final Detailed Overview of the Project Schedule (Gantt Chart).....................................................................20 Chapter 5: System Requirements....................................................................................................................................24 5.1. Functional Requirement .......................................................................................................................................24 5.2. Non-Functional Requirements.............................................................................................................................25 5.3. External Interface Requirements ........................................................................................................................26 Chapter 6: System Design Specification ........................................................................................................................27 6.1. User Stories............................................................................................................................................................27 6.2. Use-Case Diagram................................................................................................................................................28 6.3. Use-Case Descriptions.........................................................................................................................................29 6.4. Sequence Diagram ...............................................................................................................................................44 6.5. State Diagram ........................................................................................................................................................51 6.6. Class Diagram .......................................................................................................................................................52 6.7. Wireframes .............................................................................................................................................................53

**Technical Manual**

Chapter 7: Application Development and Testing.........................................................................................................61 7.1. Website Application Set-up..................................................................................................................................61 7.1.1 Hosting...............................................................................................................................................................61 7.1.2 SSL Certificate..................................................................................................................................................62 7.1.3 Hosting of typing habit data ............................................................................................................................63 7.1.4 Database ...........................................................................................................................................................64 7.2. Website Application Development Process.......................................................................................................65 7.2.1 Register Page .......................................................................................................................................................65 7.2.2 Login Page ............................................................................................................................................................69 7.2.3 Security Question Page.......................................................................................................................................73 7.2.4 Home Page ...........................................................................................................................................................75 7.2.4.1 Calibrations Page..............................................................................................................................................75 7.3. Website Application Testing ................................................................................................................................80 7.3.1. Sprint 1.................................................................................................................................................................80 7.3.2. Sprint 2.................................................................................................................................................................81 7.3.3. Sprint 3.................................................................................................................................................................83 7.3.4. Sprint 4.................................................................................................................................................................85 7.4 Machine Learning Development............................................................................................................................86 7.4.1 Self-Organizing Map ........................................................................................................................................86 7.4.2 Logistic regression ...........................................................................................................................................87 7.4.3 One Class SVM (OCSVM)..............................................................................................................................89 7.4.4 Choosing OCSVM over logistic regression ..................................................................................................90 Chapter 8: Post-development Data Collection and Analysis.......................................................................................91 8.1 Contents of survey...................................................................................................................................................91 Section A - User demographics ...............................................................................................................................91 Section B(i, ii, iii, iv, v, vi, vii, viii, ix) - Testing application section ......................................................................91 Section B(i) - Registration .....................................................................................................................................91 Section B(ii) - Login................................................................................................................................................91 Section B(iii) - Logout ............................................................................................................................................91 Section B(iv) - Return to Login page ...................................................................................................................92 Section B(v) - Security Question Login...............................................................................................................92 Section B(vi) - Recalibration .................................................................................................................................92 Section B(vii) - Logout ...........................................................................................................................................92 Section B(viii) - Test Recalibration ......................................................................................................................92

**Technical Manual**

Section B(ix) - Logout ............................................................................................................................................92 At the end of section B – conclusion ...................................................................................................................92 Section C - User knowledge on typing habit authentication ................................................................................92 Section D – Feedback ...............................................................................................................................................93 8.2 Survey Results.........................................................................................................................................................94 8.2.1. Errors & Issues ................................................................................................................................................94 8.2.2. Typing Habit Cycle..........................................................................................................................................96 8.2.3. Before and After Login....................................................................................................................................97 Chapter 9: Marketing Website........................................................................................................................................100 9.1 Home Page.............................................................................................................................................................100 9.2 Documentation Page.............................................................................................................................................100 9.3 Product Page..........................................................................................................................................................102 9.4 Meeting Page .........................................................................................................................................................103 9.5 About Us Page.......................................................................................................................................................105 Chapter 10: Security considerations .............................................................................................................................109 10.1. Privacy of Users .................................................................................................................................................109 10.1.1. Connection...................................................................................................................................................109 10.1.2. Privacy of sensitive information ....................................................................................................................109 10.1.2.1. Information stored in database ..............................................................................................................109 10.1.2.2. Data collected about users’ typing habit ..............................................................................................110 10.2. Keylogger.............................................................................................................................................................110 10.3. Logging in ............................................................................................................................................................110 Chapter 11: Future directions.........................................................................................................................................111 11.1. Keylogger.............................................................................................................................................................111 11.1.1. Improvement on key detection..................................................................................................................111 11.1.2. Include more types of keyboards..................................................................................................................111 11.2. Machine learning algorithm...............................................................................................................................111 11.2.1. Reducing the number of attempts ............................................................................................................111 Chapter 12: Appendix......................................................................................................................................................112 Chapter 13: References ..................................................................................................................................................113

**Technical Manual**

Chapter 1: Introduction

1.1. Overview

In the digital age, security is of great importance. Authentication has become a common activity in daily life. As passwords pose a risk of being compromised due to a variety of reasons such as reusing passwords or using simple passwords, exploring a different approach to authentication could be useful. The purpose of this project is to develop a typing gesture habit authentication system that can authenticate a person based on his typing habits.

1.2. Project Scope

1.2.1 Project Objective

Complete a working typing gesture habit authentication system that can recognize users based on typing habits.

1.2.2 Goals

The website must be able to recognize the user based on their typing habits.

1.2.3 Milestones

• Literature review

• Project specification report

• Progress report

• Coding of system

1.3. Project Description

This project is to authenticate users based on their typing habits instead of using passwords. Through analysis of the typing patterns using attributes that are recorded from the user, analysis would be performed on them and then determined if they match the typing habits of the user when he initially registered.

**Technical Manual**

Chapter 2: Literature Review

2.1. Target Users

2.1.1 Who will want to use a typing gesture habit authentication system?

Web developers that want to implement a complementing primary authentication means with an additional layer of security using typing biometrics.

2.1.2 Why are they the main target audience?

Our system focuses on typing biometrics as a means of authenticating the user. This reduces the burden placed on the end user, as they would not need to remember a difficult password to log in. With this we can help our targeted audience to implement an additional layer of security and protect their users better.

2.1.3 Why should they use a typing gesture habit authentication system?

Passwords have traditionally been the general method used to authenticate users. Apart from the fact that they are easy and cheap to implement, password authentication is a traditional go-to method as other methods of authentication may not be as popular. However, password-based authentication relies on something the user knows. Thus, this results in:

1) Users forgetting their password

Most website applications require their user to have difficult passwords (eg: minimum 8 characters; 1 uppercase and 1 lowercase letter; 1 symbol). However, this burdens the user with remembering a difficult password and more than often, users tend to forget their passwords. They would have to reset their passwords and they would have to rack their brains to come out with another password to remember.

2) Users may pick weak passwords or reuse passwords for every site because it is easier to remember

Most users tend to prefer remembering just one password and will reuse it for most of their accounts across sites and applications (Perrig, 2015). Thus, if their password is cracked, they might be compromised on multiple accounts.

3) Passwords relies on something-you-know

Password authentication relies on a secret shared between the user and the application. This may be prone to shoulder surfing attacks where the password is stolen by peering over the user’s shoulders.

With the growing concerns over password-based authentication, there has been interest in developing newer methods to authenticate users. Authenticating users based on their typing habits has several benefits:

**Technical Manual**

1) Unique typing habits

Typing habits are measured up to milliseconds by software. Thus, it is difficult to mimic another user’s typing habits at such a level of precision without massive efforts (Pin Shen Teh, 2013).

2) Transparent and non-invasive

It requires none or minimal alteration to users’ usual behavior since the evaluation of typing habits is calculated in the backend of the website application. This also reduces the effort and burden towards users as they would not need to remember another password (Md Liakat Ali, 2017).

3) Added security

Since it is established that keystrokes are challenging to replicate, it adds a level of security on top of usual security measures such as locking users temporarily after multiple failed attempts. Stolen credentials become almost insignificant as attackers would need to spend a huge effort to replicate the typing habit. If the typing habit gets compromised, user would be able to register a new typing habit with a new set of words (Pin Shen Teh, 2013).

2.2 Competitors Analysis

TypingDNA is able to analyze typing patterns and accurately determine if they are a match with a known, enrolled user. This allows TypingDNA to protect trusted user accounts with powerful typing biometrics analysis, both accurately and passively without end-users suffering any drawbacks for added protection. Thus, enabling frictionless keystroke authentication in the background of any typed text without the need for additional browser permissions and thus no impact on the User Experience

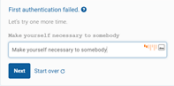
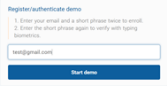
(TypingDNA, 2020).

TypingDNA targets the financial service and education sector and thus, can be deployed in scenarios in which the identity of a user needs confirmation – such as enforcing password resets, complimenting primary authentication means with an additional layer of security, or in the place of OTP flows. The API is not constrained to specific use-cases or authentication stacks, and therefore can be incorporated anywhere within the architecture that end-users are typing.

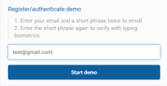
**Technical Manual**

****Figure 1. Screenshot from TypingDNA showing their 4 different demonstrations available.

Typing DNA has 4 demos that recategorized into 2 groups, Short Phrase text – identical enrollment and verification text, and Different Text – different enrollment and verification text. With these 4 demos, it allows multiple uses cases such as User Authentication, Email Security, Fraud Prevention, 2FA and electronic signatures to be created.

Figure 2. Screenshot of Short Phrase demo from typing DNA.

Different Text Demo: Different text verification works with typed texts that do not need to be the same each time.

 Figure 3. Screenshot of Different Text demo from typing DNA.

From these figures, we can see that Phase and Different Text have similar flows where we register our emails and the authentication segment after that. The difference is in the authentication segment where the Short Phase demo shows the same statement, whereas Different Text demo shows 2 different paragraphs.

**Technical Manual**

The small icon that you see inside on the right side of the input box show our typing pattern which consists of data on the timing and durations of various key press events. This data is then used for the verification process.

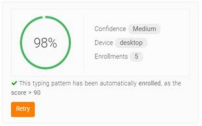
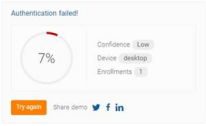


Figure 4. Screenshots showing the accuracy rate, confidence, device and enrolments from TypingDNA of a registered user.

After the Verification process, TypingDNA is then able to analyze and a true/false match response will be returned instantly as shown in these 2 different outputs, Pass and Failed. As seen, the ones that have passed would have the typing pattern automatically enrolled into the system.

Therefore, from our Competitor Analysis we have decided to have a similar flow as Short Phrase where we use a short sentence to recognise our user. However, each user will have a sentence of their own, which differs from other users, and thus function as another layer of protection as the typing biometrics differ even more between users. The registration of typing biometrics will also increase so that we would be able to have a better grasp of our user typing biometrics and thus allowing us to better authenticate the users.

**Technical Manual**

2.3 Behavioral Analysis

2.3.1 Text entry speed: Words per minute

There are 5 things to consider:

1. When did the timing for a phrase begin?

2. When did it end?

3. Did timing begin with the first character or with a START button or some other signal before the first character?

4. Did timing end with the last character or by pressing the ENTER after the last character? 5. If timing begins with the first character and ends with the last character, then, arguably, the first character should not count, since timing excludes the time leading up to the entry of the first character.

2.3.1 Accuracy / Error rate: ration of incorrect characters to total characters

Insertion errors & deletion errors may lead to accuracy problems and hence, might affect the recognition of the user.

2.3.3 Dwell time: text entry by eye typing

Things to consider:

1. A study carried out in 1996 reported that “for people with severe disabilities it can take anywhere from 15 minutes to many months to acquire eye control skill to run the system.” (Tecce, 1998, p. 320). The speed of real experts has not been systematically measured for any of the eye-controlled text entry systems.

2. As another example, consider (Wigdor & Balakrishnan, 2005, p. 212) TiltText, a technique for mobile phone text entry that uses the orientation of the device to resolve the ambiguity of letters on keys on the mobile phone keypad. In addition to conventional error rate analyses, they defined and used “button error” and “tilt error” as dependent variables. Button errors were the ratio of errors due to pressing the wrong button, and tilt errors were the ratio of errors due to tilting the device in the wrong direction.

*PiePad* is a gesture-based entry method for numeric entry (Quinn & Zhai, 2016, p. 235; (MacKenzie & Tanaka-Ishii, 2007, pp. 1–3). Gestures were stylus strokes conforming to a clock metaphor: right for 3, down for 6, left for 9, and so on. In addition to analyzing the time to make gestures, they defined and used “preparation time” and “scripting time” as dependent variables. Preparation time was the time between gestures, from stylus up after the previous character to stylus down for the current character. Scripting time was the gesturing time, from stylus down to stylus up.

**Technical Manual**

3. Experiments have been performed with text entry by gaze using a common, “off-the-shelf” video camera with the GazeTalk system (Itoh et al., 2006, p. 65). They observed text entry speeds of 3–5 WPM by untrained users using large (3 × 4) on-screen buttons. It is not surprising that systems requiring hierarchical navigation or multiple gestures are slower than on-screen Qwerty keyboards with accurate eye trackers, although the prediction and completion features can improve the text entry speed. Their use depends heavily on both individual styles and extended use, so they are difficult to evaluate without longitudinal studies.

4. The language of the text also has an effect on text entry speed. All the figures given so far are for English. GazeTalk and Dasher also support entering text in Japanese. A study with the results on the typing speed of 22–24 Kanji characters per minute, with performance improving from 19 to 23–25 characters per minute over seven short trials over 3 days was also tested (Itoh et al., 2006, p. 65). Both systems reached these text entry rates.

**Technical Manual**

Chapter 3: Methodology

3.1. Software Development Life Cycle Methodology

The Software Development Life Cycle (SDLC) is a set of procedures for the systematic development of a software project, including design and maintenance that meets user requirements while using the least number of resources possible. SDLC is a well-structured sequence of phases that assists a software project in producing high-quality, well-tested, and ready-to-use software quickly

(Muslihat, 2018). There are a variety of SDLC models available, and which one is ideal for the project will depend on the requirements we obtain, the end users we will be working with, and the scale of the project.

We will use the Agile Scrum Model (Guru99, n.d.) as the development methodology for this project. Agile software development enables for changes in requirements to be made even late in the development process. Agile is a project management method that emphasizes collaboration with stakeholders, as well as continuous improvements and iterations at each stage.

Scrum is an Agile method that comprises of self-organizing teams and advances via a series of month-long "sprints" in which the product is developed. The project's requirements are recorded as "user stories" in a "product backlog" list. A sprint usually lasts 2-4 weeks, or a calendar month at the maximum. During the sprint, the product is developed, coded, and tested. It's vital to realize that in a sprint, there will be no changes. Our team, on the other hand, has Weekly Scrum Meetings rather than Daily Scrum Meetings.

The Agile Scrum Model is suitable for this project because requirements are not clear due to lack of extensive research on behavioral of typing gesture habit authentication system. Collaboration between team members and project stakeholders is critical to the project's success. Throughout the development process, Agile Scrum will ensure that value is optimized. Teams can easily adapt to changing requirements throughout the process by leveraging iterative planning and feedback findings (Daily, 2020).

3.2. Research Methodology

We will use the Google Search and UOW Library to assist us in obtaining useful case studies and information for this project's research and development. Using key words to find relevant journals or articles related to the project. We will discuss various software to help us decide which software to use for product development, comparing them to see which one would be best for our project. Similarly, I would follow the same procedure when deciding which software language and database software to use.

3.3. Data Gather Methodology

As this project was done during the COVID-19 period, we will be collecting qualitative data by sending questionnaires to our classmates, friends, and students through online platform instead of physical SIM campus. Through gathering questionnaires, it will help us in deciding which features to include in the application. We will then be sent after the users try the application which will help us

**Technical Manual**

gather post-development data collection and analysis about their experience with this application. This will help to determine what kinds of features we should add in our next future work.

3.4. Development Methodology

For the development of this project, we will be using XAMPP, MySQL, PHP, CSS, HTML and Python 3 as development environment. XAMPP provides free and open-source cross-platform web server solution. MySQL is a free-to-use, open-source database that facilitates effective management of databases by connecting them to the software. It is a stable, reliable, and powerful solution with advanced features. PHP for our web development. Firstly, it is a free language with no licensing fees so the cost of using it is minimal. A good benefit of using a PHP is that it can interact with many different database languages including MySQL. Both PHP and MySQL are compatible with an Apache server which is also free to license. PHP runs on Windows, Linux and Unix servers. CSS is an integral part of the modern web development process. It is a highly effective HTML tool that provides easy control over layout and presentation of website pages by separating content from design. Python 3 has good standards and a powerful library over Python 3. Many Machines Learning and Deep Learning libraries are also improved and enhanced for good model development.

**Technical Manual**

Chapter 4: Pre-development Data Collection and Analysis 4.1. Final Project Risk Analysis and Assessment

Risk ID No. 1

Rank 1

Risk Unable to get the message across. Risk of miscommunication. Category People

Potential Responses

Project manager to intervene and understand source of miscommunication

Risk Owner Alicia

Probability Low

Impact High – Dispute results in zero progress Residual Risk Low

Risk ID No. 2

Rank 2

Risk Inability to achieve the programming requirement. Category Technical

Potential Responses

Do proper research and seek advice

Risk Owner Terrence

Probability Low

Impact High – Unable to deliver project specifications Residual Risk Low

Risk ID No. 3

Rank 5

Risk Unable to maximize individual’s roles for the project. Unable to delegate roles to specific individual.

Category People

Potential Responses

**Technical Manual**

Project manager to step in and know what specialties each individual has

Risk Owner Alicia

Probability Low

Impact Medium - Requires more communication and impede progress. Residual Risk Low

Risk ID No. 4

Rank 4

Risk Inability to achieve users’ expectation

Category People

Potential Responses

Gathering of users’ feedback and suggestions to make corrections and improve our product.

Risk Owner Joel

Probability Low

Impact Medium – Readdressing the requirements will result in time taken.

Residual Risk Low

Risk ID No. 5

Rank 3

Risk Change of project scope.

Category People

Potential Responses

Team members to discuss on the changes required for the prototype programmed.

Risk Owner Joel

Probability Medium

Impact Medium - Need to add in and change some of the information and the programmed.

Residual Risk Low

**Technical Manual**

Risk ID No. 6

Rank 6

Risk Programmer's computer break down during development Category Technical

Potential Responses

There should also be an online copy of the program on the internet / Github where the developers can retrieve from.

Risk Owner Terrence /Alicia/Cuihui

Probability Low

Impact High - Unable to deliver project specifications Residual Risk Low

Risk ID No. 7

Rank 7

Risk Programmer's computer break down during presentation Category Technical

Potential Responses

Make sure that there are at least 2 computers that has the software pre-loaded into their computers. There should also be an online copy on the internet / Github and a hard copy in a external hard disk.

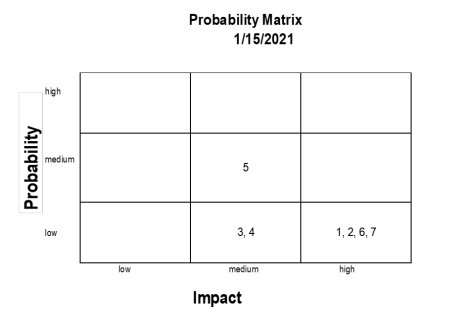
Risk Owner Terrence /Alicia/Cui hui

Probability Low

Impact High - Unable to present our code, hence significant deduction of marks

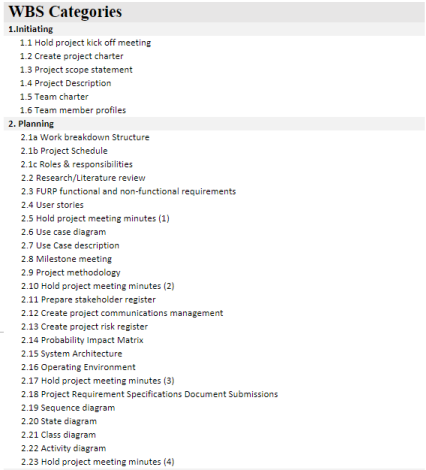
Residual Risk Low

**Technical Manual**

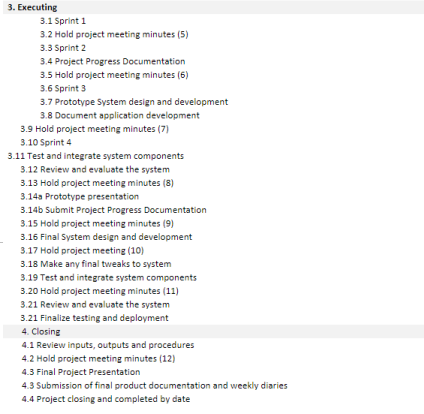
****Figure 5. Probability matrix showing the 7 identified

**Technical Manual**

4.2. Final Work Breakdown Structure



**Technical Manual**

****Figure 6. Work breakdown structure. There are 3 main categories, initiating, planning and executing.

**Technical Manual**

4.3. Final Detailed Overview of the Project Schedule (Gantt Chart)

**Technical Manual**

****

**Technical Manual**

****

**Technical Manual**

Figure 7. There are 4 phases. Initiation, planning, execution and conclusion.

**Technical Manual**

Chapter 5: System Requirements

5.1. Functional Requirement

Website System

• First visit:

Site visitors would be redirected to log into the website.

• Log in

o Step 1: Verification of username with database

o Step 2: Verification of typing habit

o Step 3: Storing of new data (from successful login) into user’s typing habit data.

• Home/Index page (only accessible after logging in)

o Display options for user to choose.

o Options are: Recalibrate and logout

• Register

o 1st: Registration of user details

▪ Details include:

• Username

• First name

• last name

• Email

• Date of birth

• Security question

• Security answer

o 2nd: Registering of new user’s typing habit.

• Recalibration:

o Recording of user’s typing habits and overwriting the file storing their typing habit.

• User

o Able to recalibrate their typing habits

**Technical Manual**

5.2. Non-Functional Requirements

5.2.1. Usability

• The website interface will be simple and straightforward with a few buttons.

5.2.2. Reliability

• The application should always be available whenever authentication is required.

5.2.3. Performance

• The authentication process should be as fast as possible while maintaining high accuracy in identifying anomalies.

5.2.4. Supportability

• All users should be able to run the application on their computer’s browser. • The application is easy to test with various test cases.

• Application is only supported on the computer.

5.2.5. Security

• Users should not be able to log into another user’s account.

• Users should only be able to log into their own account based on their typing habit. • Typing habit data collected from user for application purposes will be protected and encrypted • Users’ security answer will be hashed inside database

• Users’ typing habit data are inside a different file system, separate from the website’s file system.

• File system chosen to store user data should be secure.

• After user fails logging in using security answer three times, they will be locked out temporarily.

5.2.6. Scalability

• Code for machine learning should be simple for future implementation of this authentication system.

• Machine should learn users’ habits even during usage of programs (example: during login). • Updates on the machine learning system (to improve the system / remove bugs from the system) should only be accessible by admin/programmers with the given permission.

**Technical Manual**

5.3. External Interface Requirements

Software requirements

Our team has decided to implement the authentication system using the following software: Table 1: Table depicting our software requirements

Environment Software used Description

Operating System Windows OS We have chosen Windows operating system for several reasons: the most popular

operating system, having the best support

and user-friendliness.

Windows version required: Windows 10

(.NET Framework 3.5) or later.

Database ClearDB (MySQL) To save the user records and pre-built sentences we have chosen an SQL

database.

Web Server IIS (internet Information

Services)

To implement the system, we have chosen IIS as it is a web server that hosts websites and web applications.

Machine Learning Python3 Python is one of the most versatile programming languages. Python code is

understandable by humans, which makes it

easier to build models for machine learning.

Browser Google Chrome Python is a browser which supports HTML, CSS, JavaScript and other web languages.

Hardware requirements

Computers should have a window operating system. Internet connection is not necessary to show the implementation of the authentication system.

**Technical Manual**

Chapter 6: System Design Specification

6.1. User Stories

1) As a user, they should be able to get authenticated so that they can log in to the application. 2) As a user, they should be able to log out so that they can exit the application safely. 3) As a user, they should be able to view a sentence upon login so that they can be able to type and authenticate themselves.

4) As a user, they should be able to see the generated sentence so that they can type it for authentication.

5) As a user, they should be able to recalibrate their typing habits so that they are able to pass the authentication system even if their typing habits change.

6) As a user, they should be able to register account so that they can log in to the system. 7) As a user, they should be able to authenticate themselves by answering their security question saved during registration so as to be able to log into their accounts whenever their typing habit authentication fail.

8) As a user, they should be able to register their security question and answer so as to be able to log into their accounts in the future should their typing habit authentication fail.

**Technical Manual**

6.2. Use-Case Diagram

Figure 8. Use-Case diagram for users.

**Technical Manual**

6.3. Use-Case Descriptions

User story 1

| **User Story #1:** As a user, they should be able to get authenticated so that they can log in to the application. | |
| --- | --- |
| **Name:** Login | **Taiga ID:** 1 |
| **Stakeholders and Goals**: User wants to login into the Typing Habit Gesture Authentication System. | |
| **Descriptions:** The user is able to login into the system with the credentials that they have registered with the System. | |
| **Actors:** User | |
| **Trigger:**  1. User visits the website.  2. Website redirects to Login page. | |
| **Normal Flow:**  1. User visits the website  2. Website redirects the user to Login page.  3. User enters their username in the input field found on the Login page and clicks on the Login button to submit.  4. Website retrieves the user’s information and displays a sentence for user to verify their typing habit on the Typing Habit Verification page.  5. User enters the sentence generated by the website and clicks on the Validate Habit button to submit.  6. The system verifies and validates typing habit with the model (unique to the user) and redirects User to the homepage.  7. Data is saved to the user’s typing habit data and model is updated. 8. End. | |
| **Sub flows:** - | |
| **Alternate/Exception flows:**  A. Unable to display Login page  A1. The website is unable to load the Login page.  A2. The website will display an error message. The use case terminates. | |

**Technical Manual**

| B. No username entered in username input field  B1. User clicks the Login button.  B2. The website will prompt the User to re-input the missing text field at Step 3.  C. User enters a username that does not exist  C1. User enters an invalid username in the username text field.  C2. The website will display an error message and prompt User to register an account.  D. Unable to display Typing Habit Verification page  D1. The website is unable to load the Typing Habit Verification page. D2. The website will display an error message. The use case terminates.  E. No sentence entered in the sentence input field  E1. User clicks the Validate Habit button.  E2. The website will prompt the User to re-input the missing text field at Step 5.  F. User types a sentence that does not match the generated sentence shown F1. Website will notify user that it does not match with the number of attempts left. F2. User resubmits upon rectifying their mistake.  G. Model does not authenticate user  G1. The system is able to find a matching username. However, the System is unable to match the sentence’s typing biometrics against User’s typing biometrics in the database.  G2. The system will prompt the User to try Step 5 again with the number of attempts left.  H. User is unable to log in based on their typing habit and is redirected to the Security Question page. |
| --- |

**Technical Manual**

| H1. The system is unable to match the sentence’s typing biometrics against the user’s model after trying 3 times.  H2. The system will redirect the user to the Security Question page to log in by answering their security question that they had typed during registration. |
| --- |

**Technical Manual**

User story 2

| **User Story #2:** As a user, they should be able to log out so that they can exit the application safely. | |
| --- | --- |
| **Name:** Logout | **Taiga ID:** 2 |
| **Stakeholders and Goals**: User wants to logout from the Typing Habit Gesture Authentication System. | |
| **Descriptions:** The user is able to logout from the system. | |
| **Actors:** User | |
| **Trigger:**  1. User clicks on the Logout button on the Homepage. | |
| **Normal Flow:**  1. User clicks on the Logout button on their Homepage.  2. Website retrieves and display the successful Logout page.  1. The website redirects the user to Login page.  2. End. | |
| **Sub flows:** - | |
| **Alternate/Exception flows:**  A. Website unable to load Logout page  A1. The website is unable to redirect users and load the Logout page. A2. The website will display an error message.  A3. The use case terminates.  B. Website unable to load Login page  B1. The website is unable to redirect users and load the Login page. B2. The website will display an error message.  B3. The use case terminates. | |

**Technical Manual**

User story 3

| **User Story #3:** As a user, they should be able to enter their typing habits so that they can be authenticated. | |
| --- | --- |
| **Name:** Sentence typing | **Taiga ID:** 3 |
| **Stakeholders and Goals**: User needs to be able to type the generated sentence. | |
| **Descriptions:** User is able to type according to the generated sentence for the website to record their typing habit and validate their habit. | |
| **Actors:** User | |
| **Trigger:**  1. User clicks on the Login button on first part of Login page after entering username. | |
| **Normal Flow:**  1. User clicks on the Login button on the Login page after entering username.  2. The website displays the second part of the login form which consists of the generated sentence.  3. User types according to the sentence shown on the second part of the Login page.  4. User clicks “Validate Habit” button to submit and validate their habit. 5. End. | |
| **Sub flows:**  A. User chooses to not log in and goes to registration page instead A1. Website will load Registration page and current login details will be cleared. | |
| **Alternate/Exception flows:** - | |

**Technical Manual**

User story 4

| **User Story #4:** As a user, they should be able to view a sentence upon login so that they can be able to type and authenticate themselves. | |
| --- | --- |
| **Name:** Sentence view | **Taiga ID:** 4 |
| **Stakeholders and Goals**: User needs to be able to view the generated sentence. | |
| **Descriptions:** The user is able to view the generated sentence. | |
| **Actors:** User | |
| **Trigger:**  1. User clicks on the submit button on their login page after entering username. | |
| **Normal Flow:**  1. User clicks on the submit button on their login page after entering username.  2. The system retrieves and displays the generated sentence page. 3. User types the sentence in the blank space provided.  4. End. | |
| **Sub flows:** - | |
| **Alternate/Exception flows:**  A. The website is unable to display the login page.  A1. The system will display an error message.  A2. The use case terminates.  B. The website is unable to display the login sentence.  B1. The system will display an error message.  B2. The use case terminates. | |

User story 5

| **User Story #5:** As a user, they should be able to re-calibrate their typing habits so that they are able to pass the authentication system even if their typing habits change. | |
| --- | --- |
| **Name:** Recalibration | **Taiga ID:** 5 |

**Technical Manual**

| **Stakeholders and Goals**: User wants to re-calibrate their typing biometrics. |
| --- |
| **Descriptions:** The user will be able to re-calibrate their typing biometrics with the website. |
| **Actors:** User |
| **Trigger:**  1. User clicks on the Recalibration button on the index page (after logging in). |
| **Normal Flow:**  1. User clicks on the Recalibration button on the index page (after logging in). 2. The website loads the recalibration page with a generated sentence retrieved from the sentence bank in the database.  3. User types according to the sentence generated into the input text field. 4. User clicks on the Submit button.  5. The user will repeat Steps 3 to 4 for 10 times.  6. After the 10th time, user presses register button to register habit. 7. The website will verify and register the user’s typing habits and develop a new model unique to the user.  8. Upon successful recalibration, the website will display a success message for the user.  9. User clicks Okay button.  10.The website redirects user to the Index page.  11.End. |
| **Sub flows:**  A. User decides not to recalibrate their habit  A1. User clicks on the logo to go back to the homepage.  A2. The use case terminates. |
| **Alternate/Exception flows:**  A. The website is unable to display the Recalibration page.  A1. The website will display an error message.  B. User tries to register without typing anything.  B2. Website will alert user to type according to generated sentence. C. User’s typed sentence does not match generated sentence. |

**Technical Manual**

| C1. The website will alert user that their typed sentence does not match generated sentence.  D. User unable to recalibrate the user’s habit.  D1. The website will display an error message.  D2. Use can repeat use case from step 3 onwards and try to recalibrate habit again. |
| --- |

**Technical Manual**

User story 6

| **User Story #6:** As a user, they should be able to register account so that they can log in to the system | |
| --- | --- |
| **Name:** Register | **Taiga ID:** 6 |
| **Stakeholders and Goals**: User wants to register an account. | |
| **Descriptions:** The user will be able to create an account. | |
| **Actors:** User | |
| **Trigger:**  1. User visits the website and gets redirected to log in.  2. User clicks on the Register button.  3. The website displays the Register page. | |
| **Normal Flow:**  1. The website displays the register page.  2. User enters first name, last name, username, email, security question, security question answer.  3. User clicks on the ‘Register’ button.  4. The website loads the second part of registration to register user’s typing habit, which consists of a generated sentence from the database and an input text field for user to type in.  5. User types according to the generated sentence and presses ‘Next’ button. 6. User repeats Step 5 for 10 times.  7. After the 10th time, user clicks ‘Register’ button.  8. The website processes the user’s typing habit data and creates a unique model for the user.  9. End. | |
| **Sub flows:**  A. User decides to click on Login page.  A1. All registration details will be discarded.  A2. The use case terminates. | |
| **Alternate/Exception flows:**  A. User decides to click on Login button instead of continuing registration. A1. All details filled in during registration will be discarded.  A2. The use case terminates. | |

**Technical Manual**

| B. User chooses a username that has already been taken.  B1. Website will alert user that username is invalid as it has already been taken. B2. User will continue Step 2 of the use case.  C. User does not fill up any of the column.  C1. Website will not allow user to submit unless all columns are filled. C2. User will continue Step 2 of the use case.  D. User chooses an email that has already been taken.  D1. Website will alert user that email is invalid as it has already been taken. D2. User will continue Step 2 of the use case.  E. User fills up email with an invalid input (eg: ‘1’ instead of an email address) E1. Website will alert user to fill up field with a valid email.  E2. User will continue Step 2 of the use case.  F. User fills up birthday with alphabets  F1. Website input field will not show the input as it only accepts numbers. F2. User will continue Step 2 of the use case.  G. User fills up with invalid security question (Less than 3 char)  G1. Website will alert user to fill up field with a valid email.  G2. User will continue Step 3 of the use case.  H. User fills up with invalid security answer (Less than 3 char)  H1. Website will alert user to fill up field with a valid email.  H2. User will continue Step 3 of the use case. |
| --- |

**Technical Manual**

**Technical Manual**

User story 7

| **User Story #7:** As a user, they should be able to authenticate themselves using a security answer in case their typing habits change. | |
| --- | --- |
| **Name:** Security Login | **Taiga ID:** 167 |
| **Stakeholders and Goals**: User wants to login into the Typing Habit Gesture Authentication System using security login. | |
| **Descriptions:** The user is able to login into the system with the security answer that they have registered with the website. | |
| **Actors:** User | |
| **Trigger:**  1. User is unable to authenticate themselves after 3 tries. | |
| **Normal Flow:**  1. User is unable to verify and authenticate their typing habit after the third time.  2. User is redirected to security login page.  3. User types in the answer to the security question entered during registration.  4. User submits the answer.  5. If User answers correctly, the system verifies and validates the security answer with the database and redirects User to the index page.  6. End. | |
| **Sub flows:**  A. User decided to click on Register button  A1. Website will discard login credentials.  A2. The use case terminates. | |
| **Alternate/Exception flows:**  A. Website is unable to display the Security Login page.  A1. The system will display an error message.  A2. The use case terminates.  B. User submits an empty input field. | |

**Technical Manual**

| B1. The website will alert the user to answer the security question.  C. User answers incorrectly  C1. The website will alert the user that security answer was incorrect, and they have 2 more tries.  C2. Use case continues from Step 3.  D. User answers incorrectly after 3 times.  D1. The system will display an error message.  D2. The user will be locked out of the website for 1 minute.  D3. The use case terminates. |
| --- |

**Technical Manual**

User story 8

| **User Story #8:** As a user, they should be able to register their security question and answer so as to be able to log into their accounts in the future should their typing habit authentication fail. | |
| --- | --- |
| **Name:** Register security question and answer | **Taiga ID:** 169 |
| **Stakeholders and Goals**: User wants to register a security question and answer in case they are unable to authenticate through their typing habits | |
| **Descriptions:** The user is able to register a security question that they can answer in case their typing habit authentication fails during log in. | |
| **Actors:** User | |
| **Trigger:**  1. User clicks on Registration button. | |
| **Normal Flow:**  1. User clicks on Registration button.  2. Website loads the registration page to load the first part of  registration.  3. User fills up registration details (username, first name, last name, email, date of birth, security question, security answer)  4. User clicks Submit  5. End | |
| **Sub flows:** - | |
| **Alternate/Exception flows:**  A. User decides to click on Login button instead of continuing registration. A1. All details filled in during registration will be discarded.  A2. The use case terminates.  B. User chooses a username that has already been taken  B1. Website will alert user that email is invalid as it has already been taken. B2. User will continue Step 3 of the use case.  C. User does not fill up any of the column  C1. Website will not allow user to submit unless all columns are filled. | |

**Technical Manual**

| C2. User will continue Step 3 of the use case.  D. User chooses an email that has already been taken  D1. Website will alert user that email is invalid as it has already been taken D2. User will continue Step 3 of the use case.  E. User fills up email with an invalid input (eg: ‘1’ instead of an email address) E1. Website will alert user to fill up field with a valid email.  E2. User will continue Step 3 of the use case.  F. User fills up birthday with alphabets  F1. Website input field will not show the input as it only accepts numbers. F2. User will continue Step 3 of the use case.  G. User fills up with invalid security question (Less than 3 char)  G1. Website will alert user to fill up field with a valid email.  G2. User will continue Step 3 of the use case.  H. User fills up with invalid security answer (Less than 3 char)  H1. Website will alert user to fill up field with a valid email.  H2. User will continue Step 3 of the use case. |
| --- |

**Technical Manual**

6.4. Sequence Diagram

**User Story #1:** As a user, they should be able to get authenticated so that they can log in to the application.

Figure 9. Sequence diagram of user story #1.

**Technical Manual**

**User Story #2:** As a user, they should be able to log out so that they can exit the application safely. Figure 10. Sequence diagram of user story #1.

**Technical Manual**

**User Story #3:** As a user, they should be able to enter their typing habits so that they can be authenticated.

**&**

**User Story #4:** As a user, they should be able to view a sentence upon login so that they can be able to type and authenticate themselves.

Figure 11. Sequence diagram of user story #3 & #4.

**Technical Manual**

**User Story #5:** As a user, they should be able to re-calibrate their typing habits so that they are able to pass the authentication system even if their typing habits change.

Figure 12. Sequence diagram of user story #5.

**Technical Manual**

**User Story #6:** As a user, they should be able to register account so that they can log in to the system

Figure 13. Sequence diagram of user story #6.

**Technical Manual**

**User Story #7:** As a user, they should be able to authenticate themselves using a security answer in case their typing habits change.

Figure 14. Sequence diagram of user story #7.

**Technical Manual**

**User Story #8:** As a user, they should be able to register their security question and answer so as to be able to log into their accounts in the future should their typing habit authentication fail.

Figure 15. Sequence diagram of user story #8.

**Technical Manual**

6.5. State Diagram

Figure 16. State diagram for the typing habit authentication system.

**Technical Manual**

6.6. Class Diagram

Figure 17. Class diagram for the typing habit authentication system.

**Technical Manual**

6.7. Wireframes

Figure 18. Wireframe showing the login page.



Figure 19. Wireframe showing if user clicks "Log In” button without typing any username.

**Technical Manual**

****Figure 20. Wireframe showing if username is not registered with the system.



Figure 21. Wireframe showing step 1 of registering an account. User is required to enter username, first name, last name, date of birth, email, security question and security answer.

**Technical Manual**

****

Figure 22. Wireframe showing step 2 of registering an account. User is required to type the sentence shown 10 times. There is a number in green below the area where the user types the sentence to show how many times more are required.



**Technical Manual**

Figure 23. Wireframe showing the ”Register Account” button after the user has completed typing the shown sentence 10 times.



Figure 24. Wireframe showing the validation page that the user will see upon entering his username.



**Technical Manual**

Figure 25. Wireframe showing if user does not input anything and clicks the” Validate habit” button.



Figure 26. Wireframe showing the home page after successful validation of users typing habits.



**Technical Manual**

Figure 27. Wireframe showing the recalibration page. User is required to type the sentence shown 10 times. There is a number in green below the area where the user types the sentence to show how many times more are required.

Figure 28. Wireframe showing the recalibration page if user does not type anything.



**Technical Manual**

Figure 29. Wireframe showing the” Register” button after user completes typing the shown sentence 10 times.

Figure 30. Wireframe showing successful recalibration of users typing habits.



**Technical Manual**

Figure 31. Wireframe showing if users typing habits does not match those he registered or recalibrated with. User has a total of 3 attempts to login using typing habits.



Figure 32. Wireframe showing after the 3 attempts to log in using typing habits are used up and user is redirected to the security question.

**Technical Manual**

Chapter 7: Application Development and Testing

7.1. Website Application Set-up

7.1.1 Hosting

Pythonanywhere was used to host our files and server as it is suitable to compile Python 3.8 and supports python-based hosting website.

Figure 33. Screenshot of the hosting website (Pythonanywhere).

**Technical Manual**

7.1.2 SSL Certificate

To ensure a secure session for users, an SSL certificate was applied for under our hosting website, Pythonanywhere. SSL (CloudFlare, n.d.) ensures that the connection between the website and users are encrypted and secure. The SSL certificate ensures that our product’s URL is using HTTPS.

Figure 34. Screenshot of the SSL certificate.

**Technical Manual**

7.1.3 Hosting of typing habit data

The files containing user typing habits are located in Google Storage. Sensitive user data is separated from the main hosting platform for security purposes. This was implemented in case our authentication system is breached by malicious actors, all the information would not be readily available to them.

Google Storage encrypts data on both sides: client-side and server-side. Client-side encryption ensures that data is encrypted when users’ files are retrieved by or sent from our website. Server-side encryption ensures that the data is encrypted after Google Storage has received the data before it is stored (Cloud, n.d.).

Figure 35. Screenshot of the Google Storage file system.

**Technical Manual**

7.1.4 Database

MySQL database was adopted for this project as the team was more familiar with the platform. ClearDB was used to store user’s website-related data such as: username, first name, last name, email, security question, and hashed security answer.

Figure 36. Screenshot of the database in ClearDB.

**Technical Manual**

7.2. Website Application Development Process

7.2.1 Register Page

Click on the ‘Register’ button which will lead the user to the registration page. Figure 37. Screenshot from AuthenticateMe website showing the ’Register’ button.

**Technical Manual**

Next, the user would have to key in their particulars and click ‘Next’.



Figure 38. Screenshot from AuthenticateMe website showing step 1 of registering an account and the ’Next’ button that user clicks on after entering the information.

Once the system is able to verify the user particulars, the system will require the user to register his typing biometrics. Key in the system generated sentence and click ‘Next’. The user will be required to repeat this process 10 times.

If the user has made a mistake during their typing, they can use arrow keys or the mouse to navigate to the error and press the backspace button once for each character that they would like to remove. Do not hold the backspace button to remove 2 or more characters.

**Technical Manual**

****

Figure 39. Screenshot from AuthenticateMe website showing step 2 of registering an account and the ’Next’ button.

After 10 times, the system will prompt the user to click on the ‘Register Account’ button to complete their registration.



Figure 40. Screenshot from AuthenticateMe website showing step 2 of registering an account and the ’Register Account’ button.

**Technical Manual**

After the system is able to register the user, it will redirect them back to the Login page. 

Figure 41. Screenshot from AuthenticateMe website showing step 2 of registering an account and the ’Register Account’ button.

**Technical Manual**

7.2.2 Login Page

After entering the URL (http://fyp21s102.pythonanywhere.com/), users will be directed to the Login page. Part 1 of logging in features 2 types of error handling: handling of empty inputs and unregistered users.

After submitting an empty input, the outcome is as shown below in figure 42.



Figure 42. Screenshot from AuthenticateMe website showing an error the user clicks on the 'Log In' button while no username is entered.

After submitting an unregistered username that is not found in the database, the outcome is as shown below in figure 43.



Figure 43. Screenshot from AuthenticateMe website showing an error the user clicks on the 'Log In' button but username if not recognized.

**Technical Manual**

During a normal login flow, user enters username and clicks ‘Login’.

Figure 44. Screenshot from AuthenticateMe website showing the 'Log In' button.

After submitting a valid username that exists in the database, AuthenticateMe will load the typing habit verification page. Similarly, to the part one of logging in, part two of login contains error handling of an empty input.

**Technical Manual**

As shown below in figure 45, type the system generated sentence in the white box and click ‘Validate Habit’. However, if a user submits an empty string, the outcome will be as shown below in figure 45.



Figure 45. Screenshot from AuthenticateMe website showing the area which the user can type the sentence and the ’Validate Habit’ button.



Figure 46. Screenshot from AuthenticateMe website showing an error if user clicks on the 'Validate Habit' button while the box is empty.

**Technical Manual**

Once the system has confirmed user typing habits, user will be granted access to the system and user homepage will be displayed.

**Technical Manual**

7.2.3 Security Question Page

Users will be redirected to this page after they have failed logging in via typing habits three times. Their unique security question will be displayed. User would have to enter their and click ‘Login’. Their security answer provided would be answer have to be exactly what they entered during registration.

Figure 47. Screenshot from AuthenticateMe website showing the security question page.

If an empty string is submitted, the outcome is as shown below in figure 48.



**Technical Manual**

Figure 48. Screenshot from AuthenticateMe website showing an error if the user clicks on the ’Log In’ button while the box is empty.

If the user has entered the correct security answer, they will be allowed to access into the system and user homepage will be displayed.

**Technical Manual**

7.2.4 Home Page

If users are authenticated, they will be granted access to the system and user homepage will be displayed. The index page features the user’s username in the top right and middle of the page. It also features 2 functions: recalibrate habit and logout.

Figure 49. Screenshot from AuthenticateMe website showing the homepage.

7.2.4.1 Calibrations Page

User can recalibrate their typing biometrics by clicking on the ‘Recalibrate typing habit” icon at the homepage.

**Technical Manual**

****Figure 50. Screenshot from AuthenticateMe website showing the 'Recalibrate typing habit’ button.

When users choose to reclibrate, they will be directed to this page where they can register their new typing habit, as seen below in figure 51.

Figure 51. Screenshot from AuthenticateMe website showing the recalibration page.

**Technical Manual**

After submitting an empty string, users will expect to see error message as seen below in figure 52. 

Figure 52. Screenshot from AuthenticateMe website showing an error if the user clicks on the ’Next’ button while the box is empty.

Users are required to key in the text box according to sentence generated by the website and click ‘Next’. This will be repeated for 10 times.

If the user has made a mistake during their typing, they can use arrow keys or the mouse to navigate to the error and press the backspace button once for each character that they would like to remove. They cannot hold the backspace button to remove 2 or more characters.

After 10 times, the system is prompt the user to click on the ‘Register‘ button to complete their typing biometric re-calibration.

Figure 53. Screenshot from AuthenticateMe website showing the 'Recalibrate’ button.

**Technical Manual**

After AuthenticateMe is able to recalibrate user’s typing biometrics, it will show a success message. Clicking ‘here’ will allow user return to their homepage.



Figure 54. Screenshot from AuthenticateMe website showing successful recalibration of users typing habits.

**Technical Manual**

7.2.4.2 Logout Page

When users choose to log out from their accounts and have logged out successfully, AuthenticateMe will display a successful logout page.



Figure 55. Screenshot from AuthenticateMe website showing that the user has successfully logged out.

Clicking “here” brings user back to the login page.

**Technical Manual**

7.3. Website Application Testing

7.3.1. Sprint 1

Figure 56. Diagram showing sprint 1.