



UNIVERSITY  
OF WOLLONGONG  
AUSTRALIA

# 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	<a href="mailto:cyatan005@mymail.sim.edu.sg">cyatan005@mymail.sim.edu.sg</a>
Terrence Yap Yong Ting	6573307	<a href="mailto:yttyap002@mymail.sim.edu.sg">yttyap002@mymail.sim.edu.sg</a>
Tng Min Li	6649737	<a href="mailto:mltng001@mymail.sim.edu.sg">mltng001@mymail.sim.edu.sg</a>
Chea Cui Hui	6647546	<a href="mailto:chchea001@mymail.sim.edu.sg">chchea001@mymail.sim.edu.sg</a>
Joel Teo Han Wen	6456467	<a href="mailto:jhwteo002@mymail.sim.edu.sg">jhwteo002@mymail.sim.edu.sg</a>

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

## 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.

## 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

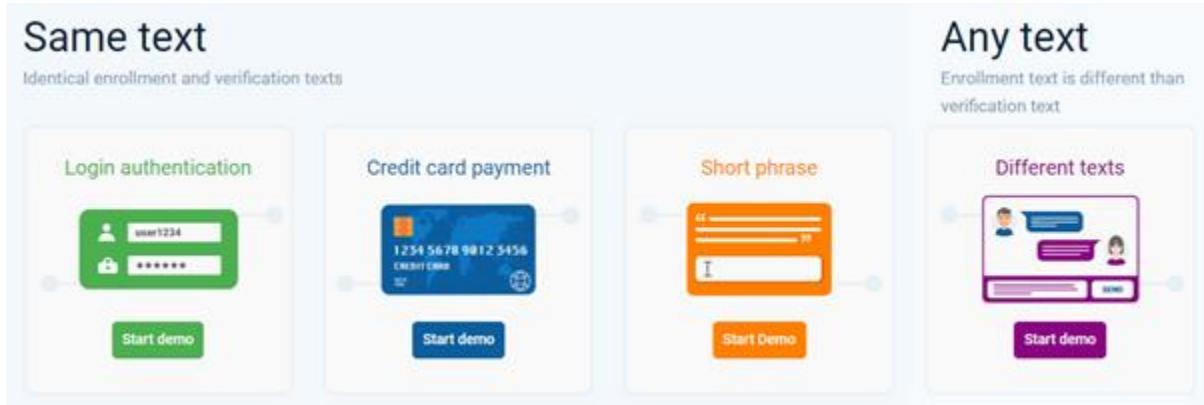


Figure 1. Screenshot from TypingDNA showing their 4 different demonstrations available.

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.

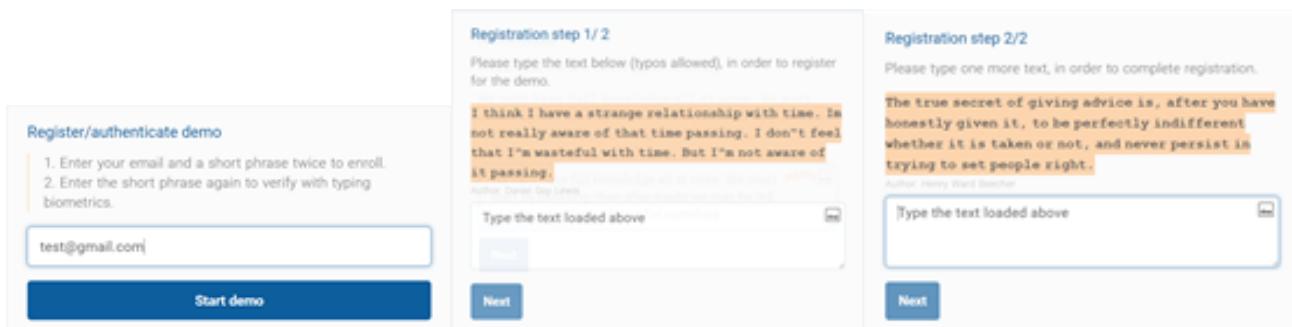


Figure 3. Screenshot of Different Text demo from typing DNA.

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 Phrase 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.

## 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: ratio 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 ( $3 \times 4$ ) 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.

## 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.

## 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

# Technical Manual

Potential Responses	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

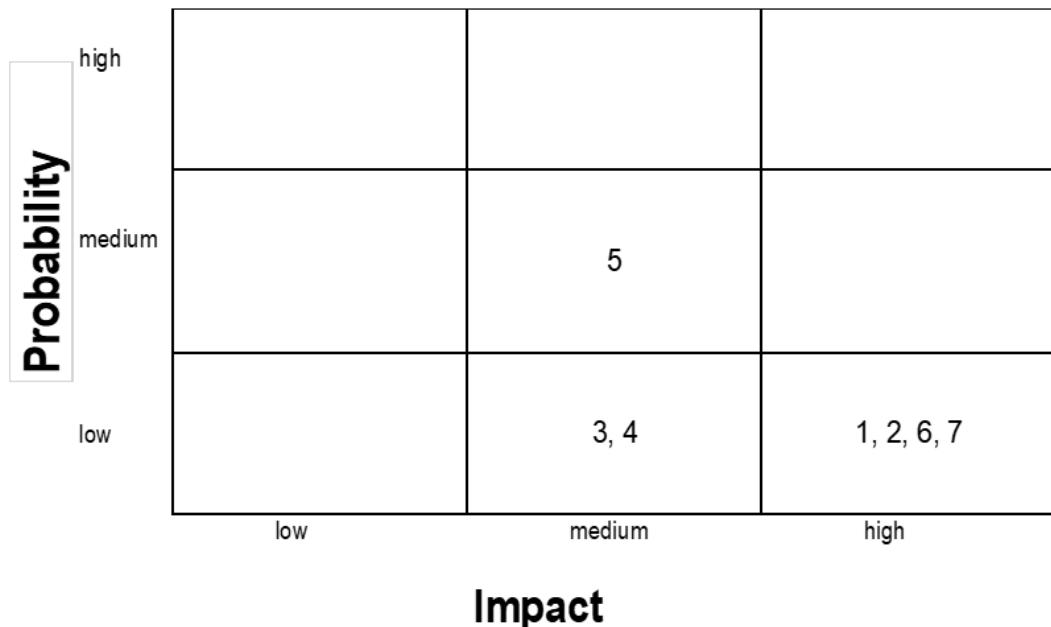
**Probability Matrix**  
**1/15/2021**

Figure 5. Probability matrix showing the 7 identified

## 4.2. Final Work Breakdown Structure

### WBS Categories

#### 1. Initiating

- 1.1 Hold project kick off meeting
- 1.2 Create project charter
- 1.3 Project scope statement
- 1.4 Project Description
- 1.5 Team charter
- 1.6 Team member profiles

#### 2. Planning

- 2.1a Work breakdown Structure
- 2.1b Project Schedule
- 2.1c Roles & responsibilities
- 2.2 Research/Literature review
- 2.3 FURP functional and non-functional requirements
- 2.4 User stories
- 2.5 Hold project meeting minutes (1)
- 2.6 Use case diagram
- 2.7 Use Case description
- 2.8 Milestone meeting
- 2.9 Project methodology
- 2.10 Hold project meeting minutes (2)
- 2.11 Prepare stakeholder register
- 2.12 Create project communications management
- 2.13 Create project risk register
- 2.14 Probability Impact Matrix
- 2.15 System Architecture
- 2.16 Operating Environment
- 2.17 Hold project meeting minutes (3)
- 2.18 Project Requirement Specifications Document Submissions
- 2.19 Sequence diagram
- 2.20 State diagram
- 2.21 Class diagram
- 2.22 Activity diagram
- 2.23 Hold project meeting minutes (4)

# Technical Manual

## 3. Executing

- 3.1 Sprint 1
- 3.2 Hold project meeting minutes (5)
- 3.3 Sprint 2
- 3.4 Project Progress Documentation
- 3.5 Hold project meeting minutes (6)
- 3.6 Sprint 3
- 3.7 Prototype System design and development
- 3.8 Document application development
- 3.9 Hold project meeting minutes (7)
- 3.10 Sprint 4
- 3.11 Test and integrate system components
- 3.12 Review and evaluate the system
- 3.13 Hold project meeting minutes (8)
- 3.14a Prototype presentation
- 3.14b Submit Project Progress Documentation
- 3.15 Hold project meeting minutes (9)
- 3.16 Final System design and development
- 3.17 Hold project meeting (10)
- 3.18 Make any final tweaks to system
- 3.19 Test and integrate system components
- 3.20 Hold project meeting minutes (11)
- 3.21 Review and evaluate the system
- 3.21 Finalize testing and deployment

## 4. Closing

- 4.1 Review inputs, outputs and procedures
- 4.2 Hold project meeting minutes (12)
- 4.3 Final Project Presentation
- 4.3 Submission of final product documentation and weekly diaries
- 4.4 Project closing and completed by date

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)

Task No.	Task Name	Assigned To / In Charge	Progress	Start	End	Days
<strong>Phase 1 - Initiation</strong>						
Task 1	Project kickoff meeting	Alicia	100%	1/15/21	1/15/21	1
Task 2	Create project charter	Alicia	100%	1/15/21	1/15/21	1
Task 3	Project scope statement	Alicia	100%	1/15/21	1/15/21	1
Task 4	Project description	Alicia	100%	1/15/21	1/15/21	1
Task 5	Team charter	Alicia	100%	1/15/21	1/15/21	1
Task 6	Team member profiling	Alicia	100%	1/15/21	1/15/21	1
Task 7	Roles & responsibilities	Alicia	100%	1/15/21	1/15/21	1
<strong>Phase 2 - Planning</strong>						
Task 1	Work breakdown structure (WBS)	Alicia	100%	1/15/21	1/15/21	1
Task 2	Project Schedule	Alicia	100%	1/15/21	1/22/21	8
Task 3	Do risk register	Alicia	100%	1/15/21	1/15/21	1
Task 4	Conduct literature review	Alicia	100%	1/16/21	1/30/21	15
Task 5	Create marketing website	Joel	100%	1/16/21	1/21/21	
Task 6	Hold project meeting (1a)	Alicia	100%	1/22/21	1/22/21	1
Task 7	FURPS+ (functional/non-functional) requirements	Alicia	100%	1/22/21	1/27/21	6
Task 8	User stories	Alicia	100%	1/15/21	1/22/21	

# Technical Manual

Task 9	Hold project meeting (1b) - same meeting minute document as meeting 1a	Alicia	100%	1/25/21	1/25/21	1
Task 10	Project methodology	Joel	100%	1/22/21	1/22/21	1
Task 11	Prepare stakeholder register	Alicia	100%	1/29/21	1/29/21	1
Task 12	Prepare probability impact matrix	Alicia	100%	1/29/21	1/29/21	1
Task 13	System architecture	Terrence	100%	1/29/21	1/29/21	1
Task 14	Operating environment	Alicia	100%	1/29/21	1/29/21	1
Task 15	Supervisor meeting (1)	Alicia	100%	1/27/21	1/27/21	1
Task 16	Milestone meeting (1)	Alicia	100%	1/29/21	1/29/21	1
Task 17	Hold project meeting (2)	Alicia	100%	2/5/21	2/5/21	1
Task 17	Hold project meeting (3)	Alicia	100%	2/10/21	2/10/21	1
Task 18	Project requirement specifications (Document submission)	Alicia	100%	2/13/21	2/13/21	1
Task 19	Hold project meeting (4)	Alicia	100%	2/19/21	2/19/21	1
Task 20	Prepare state diagram	Minli	100%	1/22/12	1/25/12	4
Task 21	Create user stories on Taiga	Joel/Cuihui	100%	2/10/21	2/13/21	4
Task 22	Create tasks on Taiga	Joel/Cuihui	100%	2/10/21	2/13/21	4
<b>Phase 3 Execution</b>						
Task 1	Sprint 1	Alicia	100%	2/15/21	2/24/21	10
Task 2	Hold project meeting (5)	Alicia	100%	2/26/21	2/26/21	1

# Technical Manual

Task 3	Prototype system design and development	Terrence	100%	2/15/21	3/24/21	38
Task 4	Hold project meeting(6) (After exam break)	Alicia	100%	3/12/21	3/12/21	1
Task 5	Sprint 2	Alicia	100%	3/15/21	3/24/21	10
Task 6	Hold project meeting (7)	Alicia	100%	3/19/21	3/19/21	1
Task 7	Document application development	Alicia	100%	2/15/21	3/25/21	39
Task 8	Hold project meeting (8)	Alicia	100%	3/26/21	3/26/21	1
Task 9	Document Project progress	Alicia	100%	2/15/21	3/25/21	39
Task 10	Prototype presentation	Alicia	100%	3/27/21	3/27/21	1
Task 11	Submit progress report and presentation slides	Alicia	100%	3/27/21	3/27/21	1
Task 12	Sprint 3	Alicia	100%	3/29/21	4/16/21	19
Task 13	Hold project meeting (9)	Alicia	100%	4/1/21	4/1/21	1
Task 14	Hold project meeting (10)	Alicia	100%	4/9/21	4/9/21	1
Task 15	Hold project meeting (11)	Alicia	100%	4/16/21	4/16/21	1
Task 16	Sprint 4	Alicia	100%	4/19/21	5/7/21	19
Task 17	Hold project meeting (12)	Alicia	100%	4/23/21	4/23/21	1
Task 18	User manual	Minli	100%	3/29/21	5/7/21	40
Task 19	Hold project meeting (13)	Alicia	100%	4/30/21	4/30/21	1
Task 20	Technical Manual	Cuihui	100%	3/29/21	5/7/21	40
Task 21	Supervisor meeting (2)	Alicia	100%	5/5/21	5/5/21	1
Task 22	Hold project meeting (14)	Alicia	100%	5/7/21	5/7/21	1
Task 23	Conduct UAT Survey	Joel	100%	5/7/21	5/14/21	8
Task 24	Final system tweaks	Alicia/Terrence	100%	5/13/21	5/15/21	3
Task 25	Hold project meeting (15)		100%	5/14/21	5/14/21	1
Phase 4 Conclusion/Wrap up						
Task 1	Review inputs, outputs and procedures	Alicia	100%	5/15/21	5/20/21	6
Task 2	Create videos for demonstration and marketing of our product	Joel	100%	5/21/21	5/21/21	1
Task 3	Hold project meeting (16)	Alicia	100%	5/21/21	5/21/21	1
Task 4	Submission of final product documentation and v	Alicia	100%	5/22/21	5/22/21	1
Task 5	Final Project Presentation	Alicia	100%	5/29/21	5/29/21	1

## **Technical Manual**

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

## Chapter 5: System Requirements

### 5.1. Functional Requirement

#### Website System

- First visit:  
Site visitors would be redirected to log into the website.
- Log in
  - Step 1: Verification of username with database
  - Step 2: Verification of typing habit
  - Step 3: Storing of new data (from successful login) into user's typing habit data.
- Home/Index page (only accessible after logging in)
  - Display options for user to choose.
  - Options are: Recalibrate and logout
- Register
  - 1<sup>st</sup>: Registration of user details
    - Details include:
      - Username
      - First name
      - last name
      - Email
      - Date of birth
      - Security question
      - Security answer
  - 2<sup>nd</sup>: Registering of new user's typing habit.
- Recalibration:
  - Recording of user's typing habits and overwriting the file storing their typing habit.
- User
  - Able to recalibrate their typing habits

## 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.

## 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.

## 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.

## 6.2. Use-Case Diagram

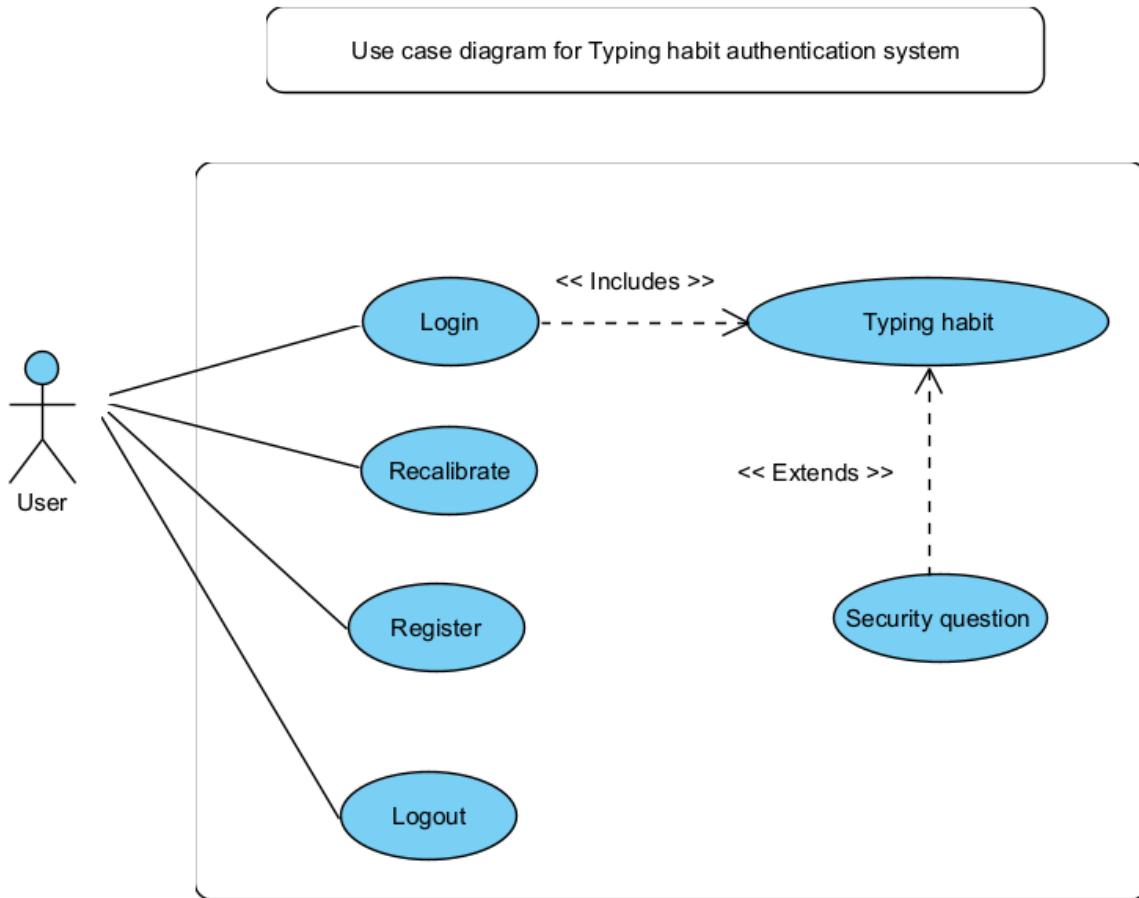


Figure 8. Use-Case diagram for users.

## 6.3. Use-Case Descriptions

### User story 1

<b>User Story #1:</b> As a user, they should be able to get authenticated so that they can log in to the application.	
<b>Name:</b> Login	<b>Taiga ID:</b> 1
<b>Stakeholders and Goals:</b> User wants to login into the Typing Habit Gesture Authentication System.	
<b>Descriptions:</b> The user is able to login into the system with the credentials that they have registered with the System.	
<b>Actors:</b> User	
<b>Trigger:</b> <ol style="list-style-type: none"><li>1. User visits the website.</li><li>2. Website redirects to Login page.</li></ol>	
<b>Normal Flow:</b> <ol style="list-style-type: none"><li>1. User visits the website</li><li>2. Website redirects the user to Login page.</li><li>3. User enters their username in the input field found on the Login page and clicks on the Login button to submit.</li><li>4. Website retrieves the user's information and displays a sentence for user to verify their typing habit on the Typing Habit Verification page.</li><li>5. User enters the sentence generated by the website and clicks on the Validate Habit button to submit.</li><li>6. The system verifies and validates typing habit with the model (unique to the user) and redirects User to the homepage.</li><li>7. Data is saved to the user's typing habit data and model is updated.</li><li>8. End.</li></ol>	
<b>Sub flows:</b> -	
<b>Alternate/Exception flows:</b> <ol style="list-style-type: none"><li>A. Unable to display Login page<ol style="list-style-type: none"><li>A1. The website is unable to load the Login page.</li><li>A2. The website will display an error message. The use case terminates.</li></ol></li></ol>	

# 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.

<b>Name:</b> Logout	<b>Taiga ID:</b> 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.

<b>Name:</b> Sentence typing	<b>Taiga ID:</b> 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.

<b>Name:</b> Sentence view	<b>Taiga ID:</b> 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.

<b>Name:</b> Recalibration	<b>Taiga ID:</b> 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 10<sup>th</sup> 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. User can repeat use case from step 3 onwards and try to recalibrate habit again.

# Technical Manual

## User story 6

<b>User Story #6:</b> As a user, they should be able to register account so that they can log in to the system	
<b>Name:</b> Register	<b>Taiga ID:</b> 6
<b>Stakeholders and Goals:</b> User wants to register an account.	
<b>Descriptions:</b> The user will be able to create an account.	
<b>Actors:</b> User	
<b>Trigger:</b> <ol style="list-style-type: none"><li>1. User visits the website and gets redirected to log in.</li><li>2. User clicks on the Register button.</li><li>3. The website displays the Register page.</li></ol>	
<b>Normal Flow:</b> <ol style="list-style-type: none"><li>1. The website displays the register page.</li><li>2. User enters first name, last name, username, email, security question, security question answer.</li><li>3. User clicks on the 'Register' button.</li><li>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.</li><li>5. User types according to the generated sentence and presses 'Next' button.</li><li>6. User repeats Step 5 for 10 times.</li><li>7. After the 10<sup>th</sup> time, user clicks 'Register' button.</li><li>8. The website processes the user's typing habit data and creates a unique model for the user.</li><li>9. End.</li></ol>	
<b>Sub flows:</b> <ol style="list-style-type: none"><li>A. User decides to click on Login page.<ol style="list-style-type: none"><li>A1. All registration details will be discarded.</li><li>A2. The use case terminates.</li></ol></li></ol>	
<b>Alternate/Exception flows:</b> <ol style="list-style-type: none"><li>A. User decides to click on Login button instead of continuing registration.<ol style="list-style-type: none"><li>A1. All details filled in during registration will be discarded.</li><li>A2. The use case terminates.</li></ol></li></ol>	

- 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.

<b>Name:</b> Security Login	<b>Taiga ID:</b> 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.

## 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.

<b>Name:</b> Register security question and answer	<b>Taiga ID:</b> 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.

## 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.

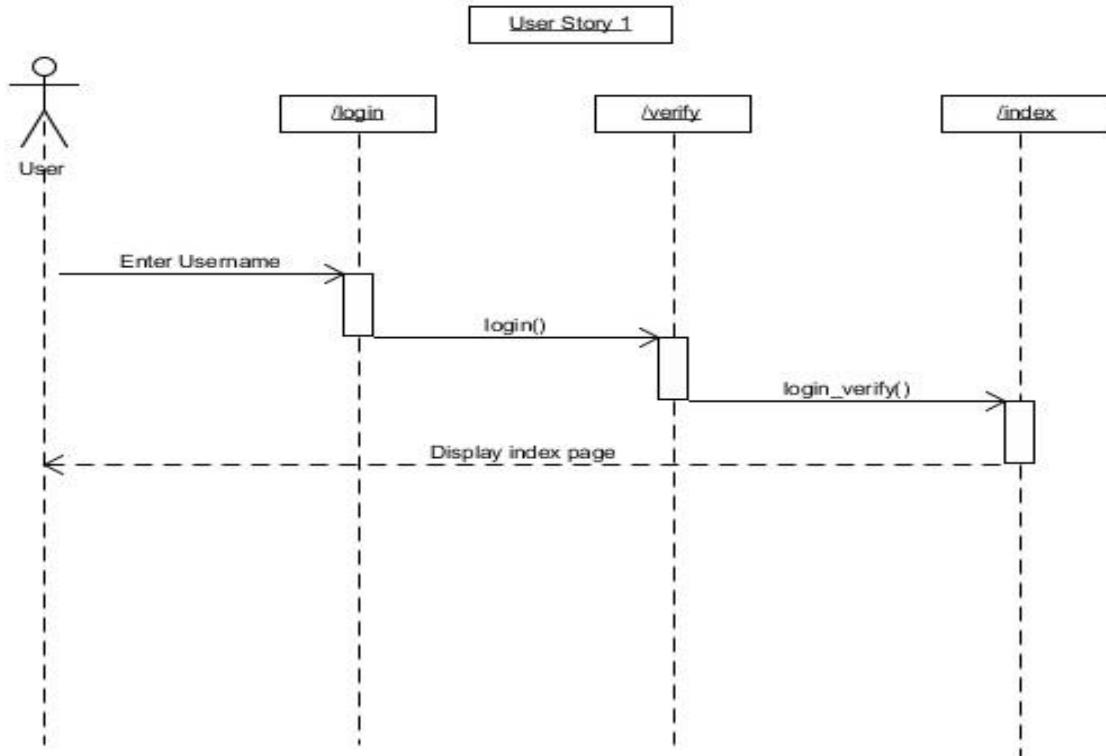


Figure 9. Sequence diagram of user story #1.

# Technical Manual

**User Story #2:** As a user, they should be able to log out so that they can exit the application safely.

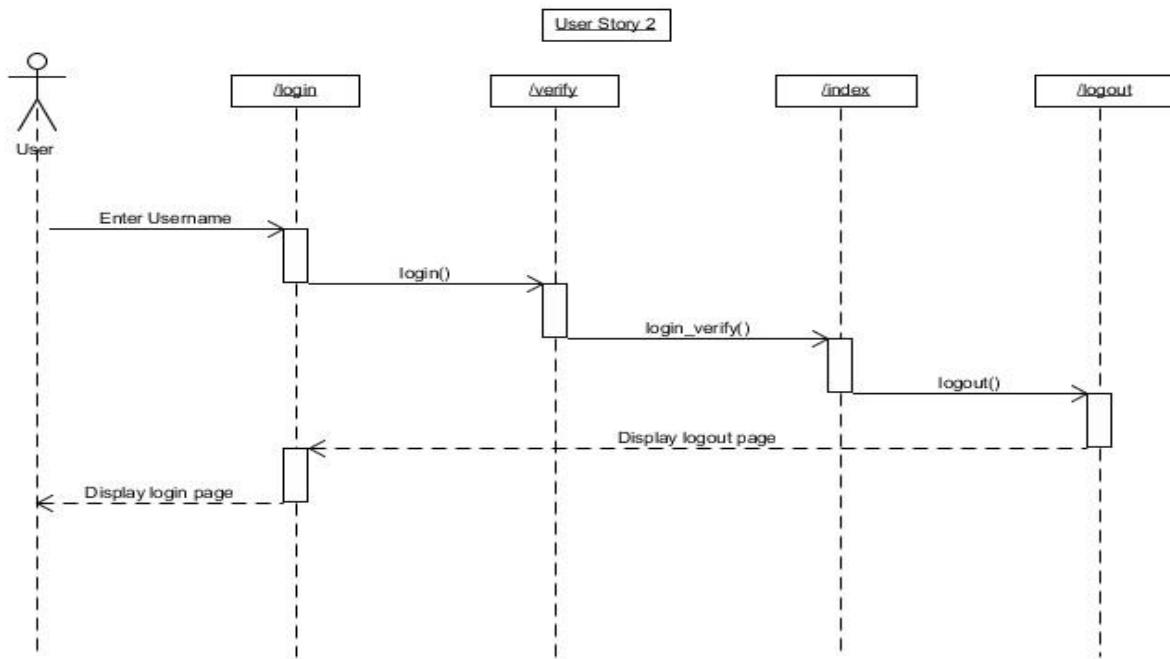


Figure 10. Sequence diagram of user story #1.

# 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.

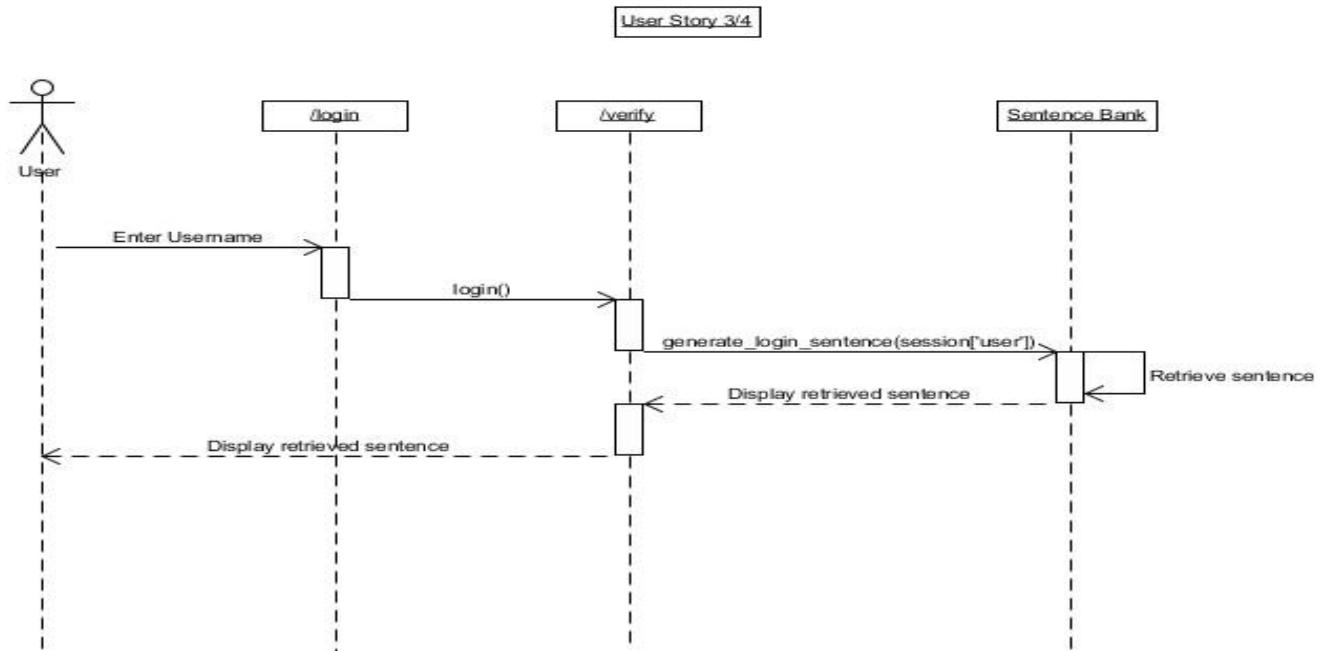


Figure 11. Sequence diagram of user story #3 & #4.

# 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.

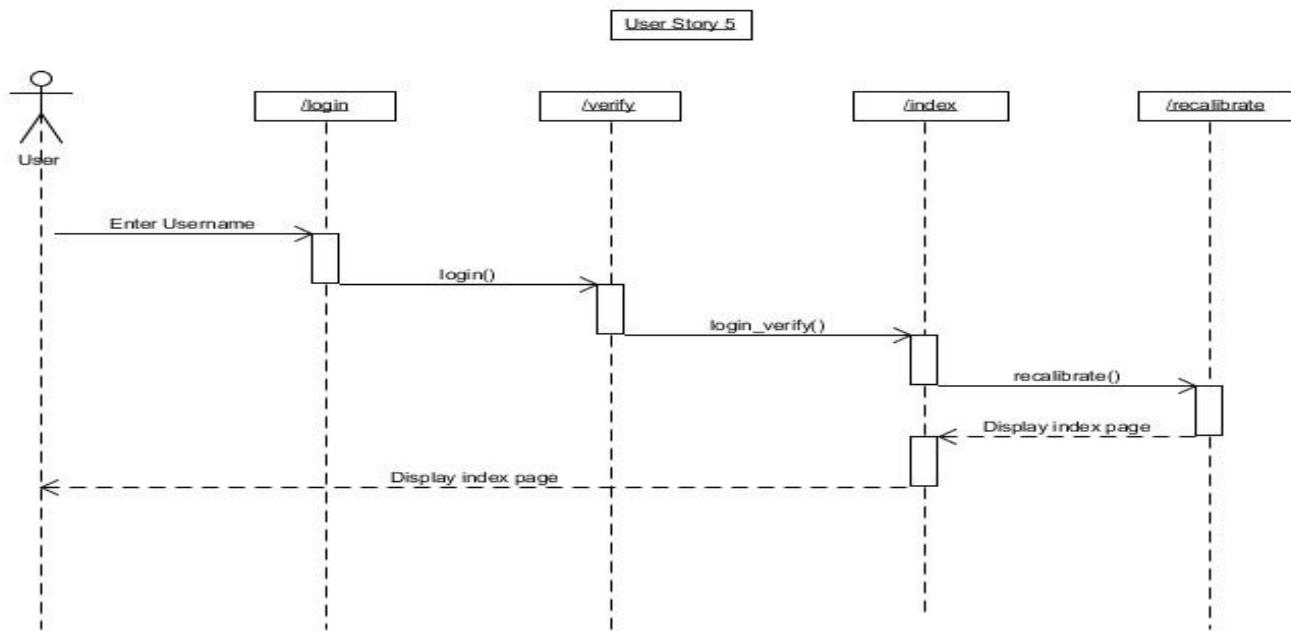


Figure 12. Sequence diagram of user story #5.

# Technical Manual

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

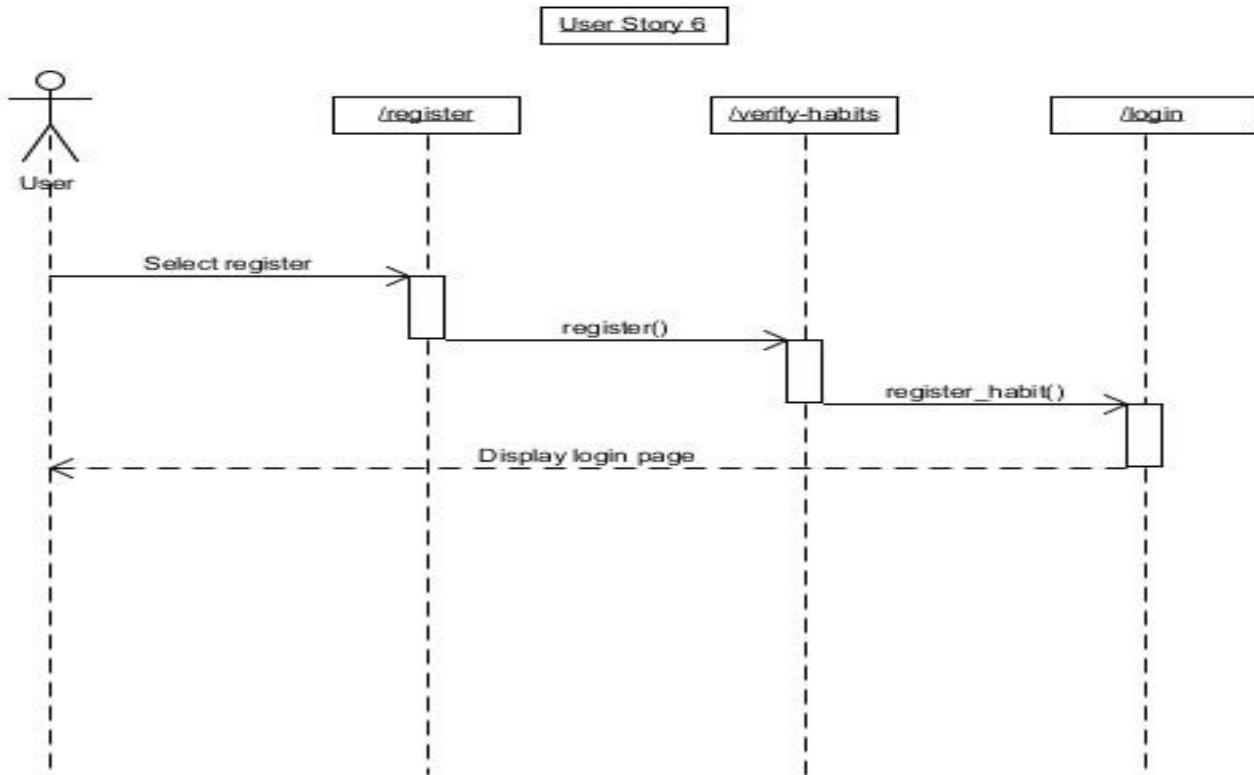


Figure 13. Sequence diagram of user story #6.

# 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.

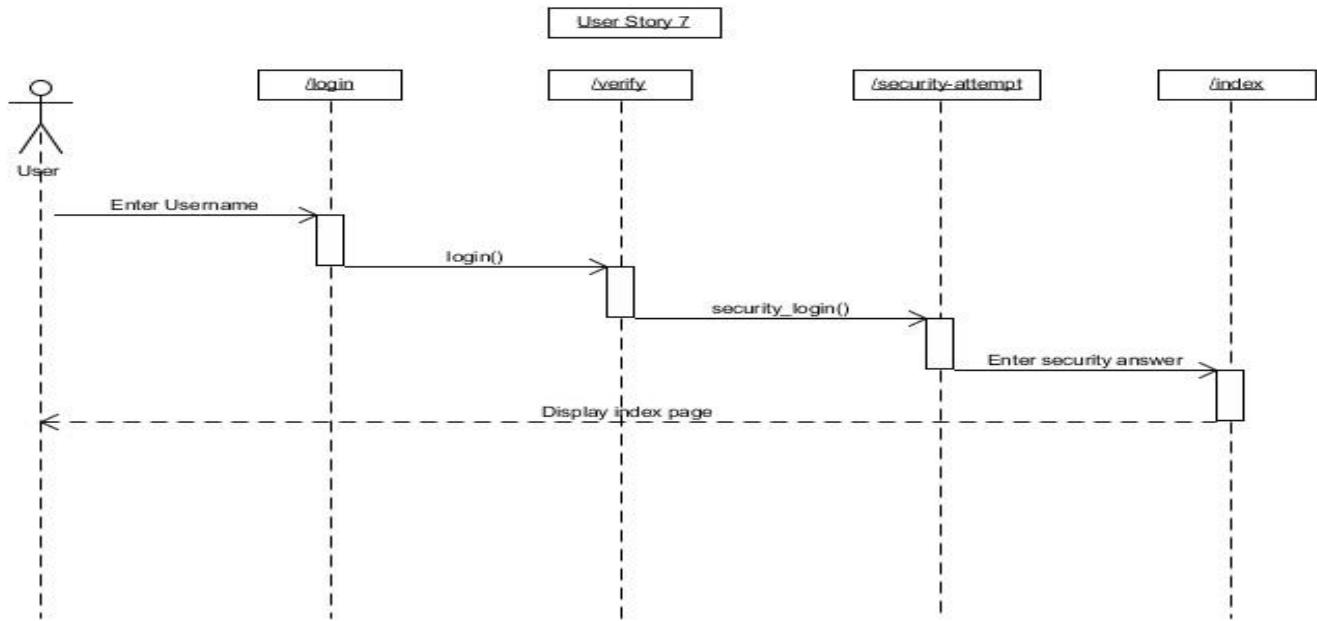


Figure 14. Sequence diagram of user story #7.

# 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.

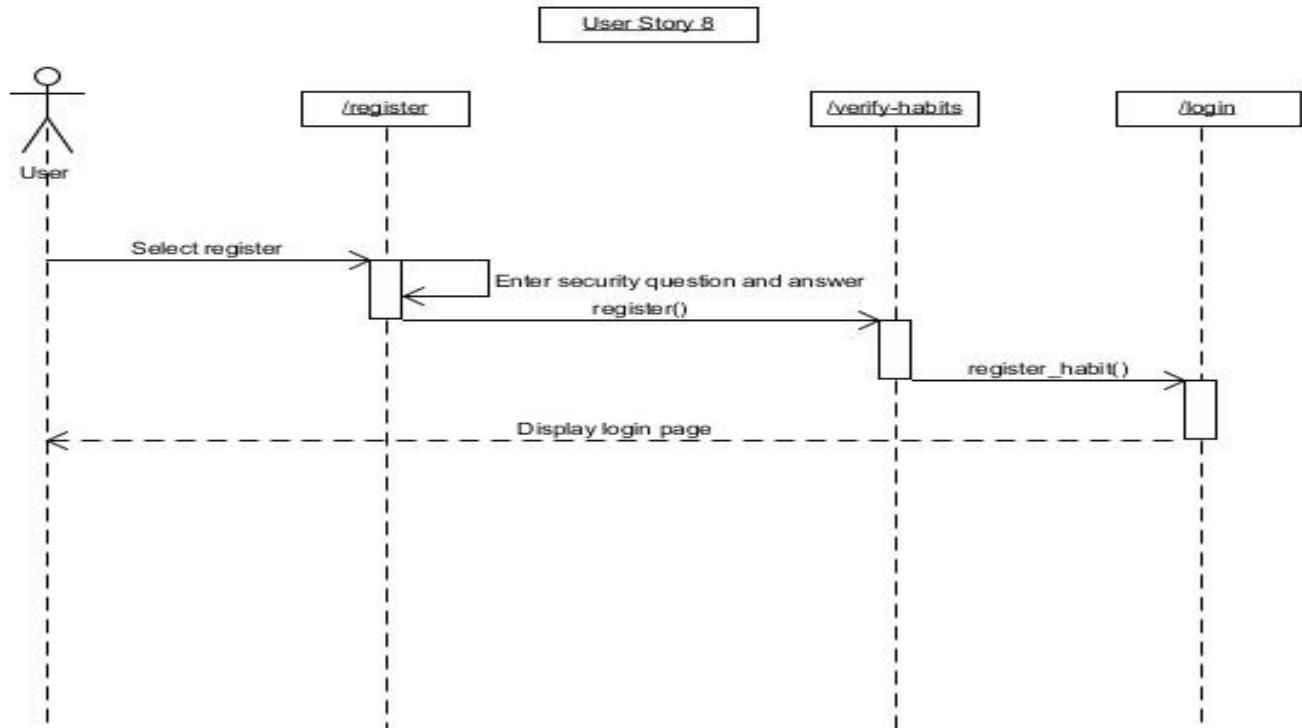


Figure 15. Sequence diagram of user story #8.

## 6.5. State Diagram

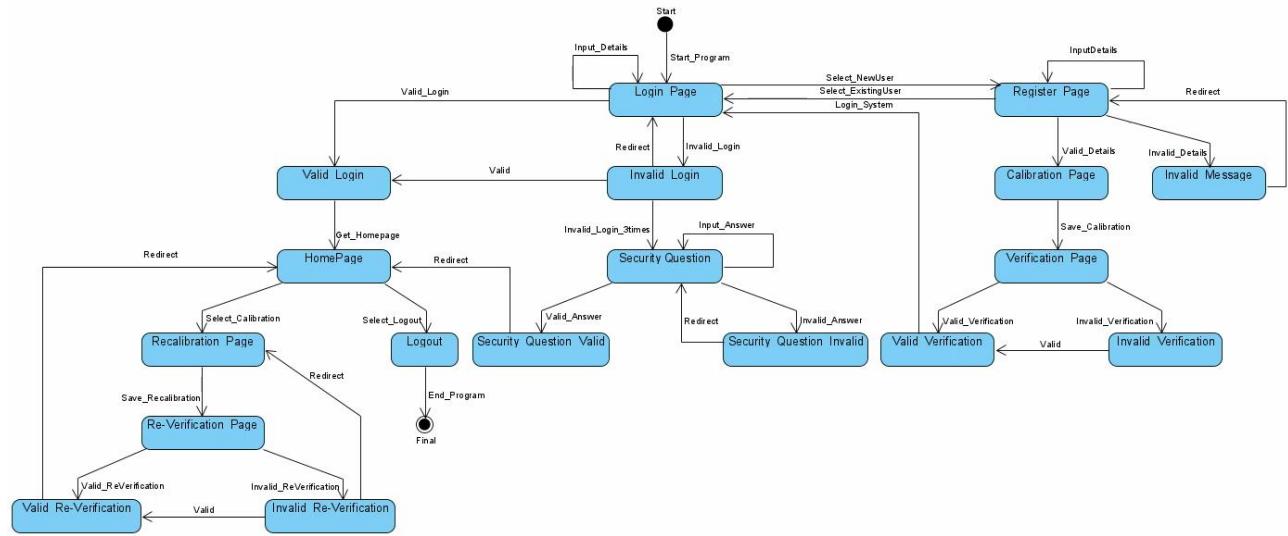


Figure 16. State diagram for the typing habit authentication system.

# Technical Manual

## 6.6. Class Diagram

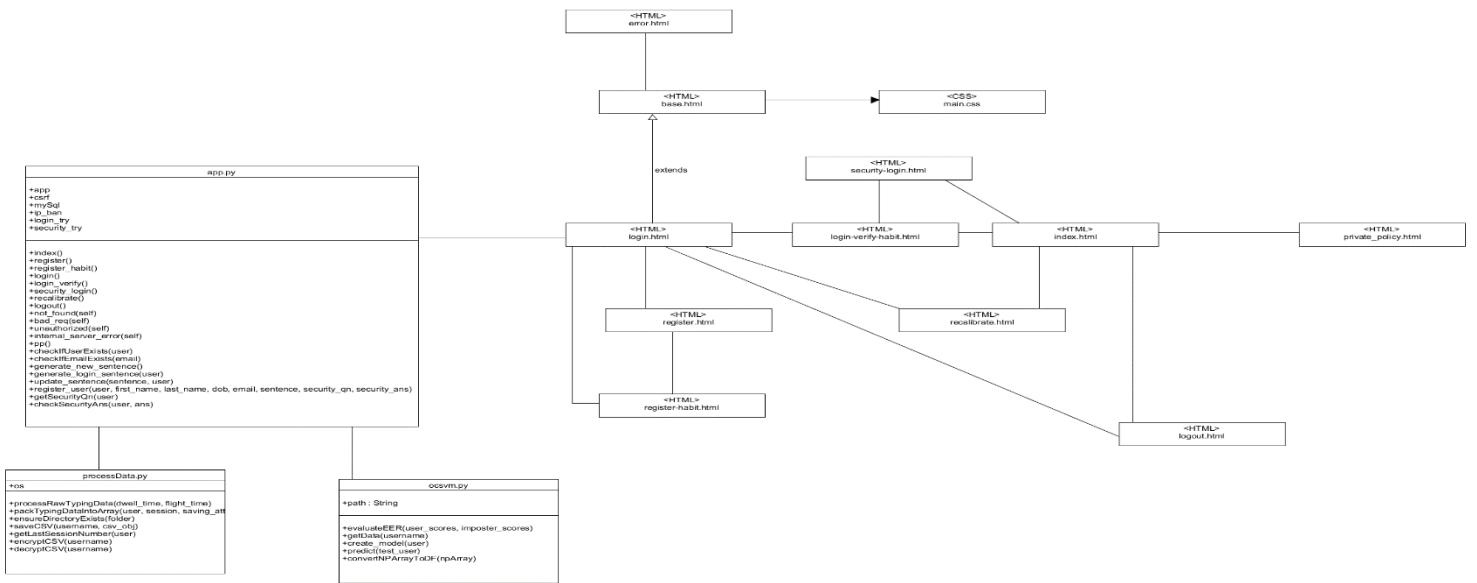


Figure 17. Class diagram for the typing habit authentication system.

## 6.7. Wireframes

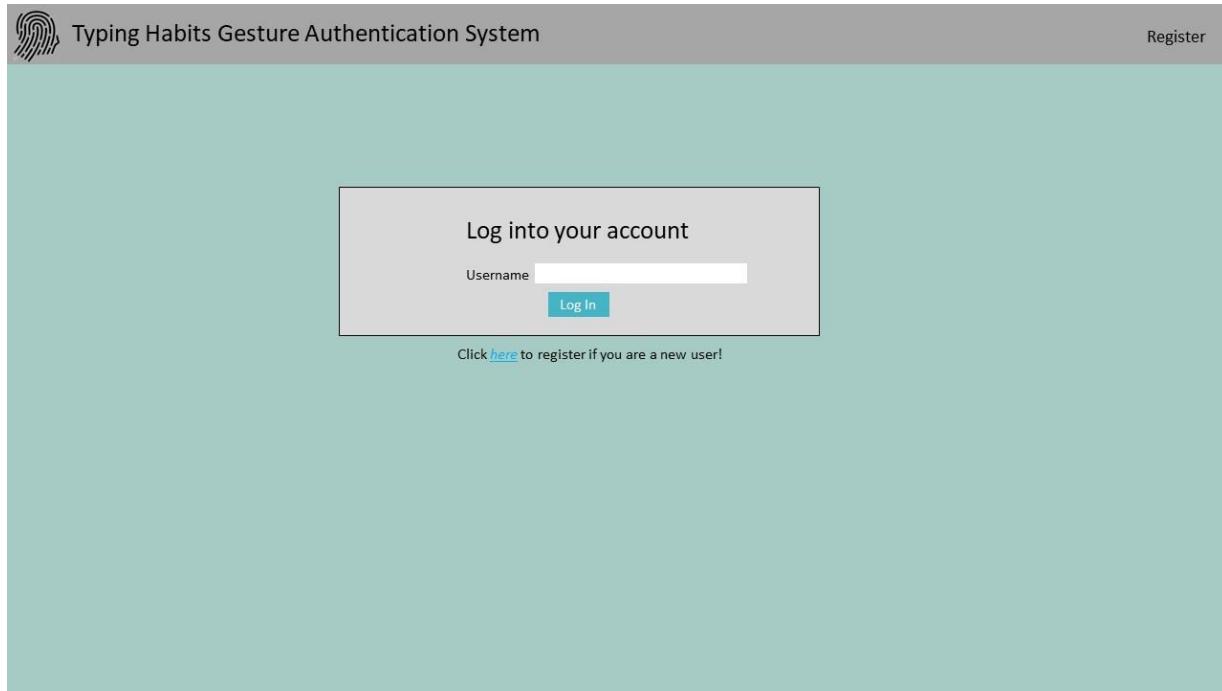


Figure 18. Wireframe showing the login page.

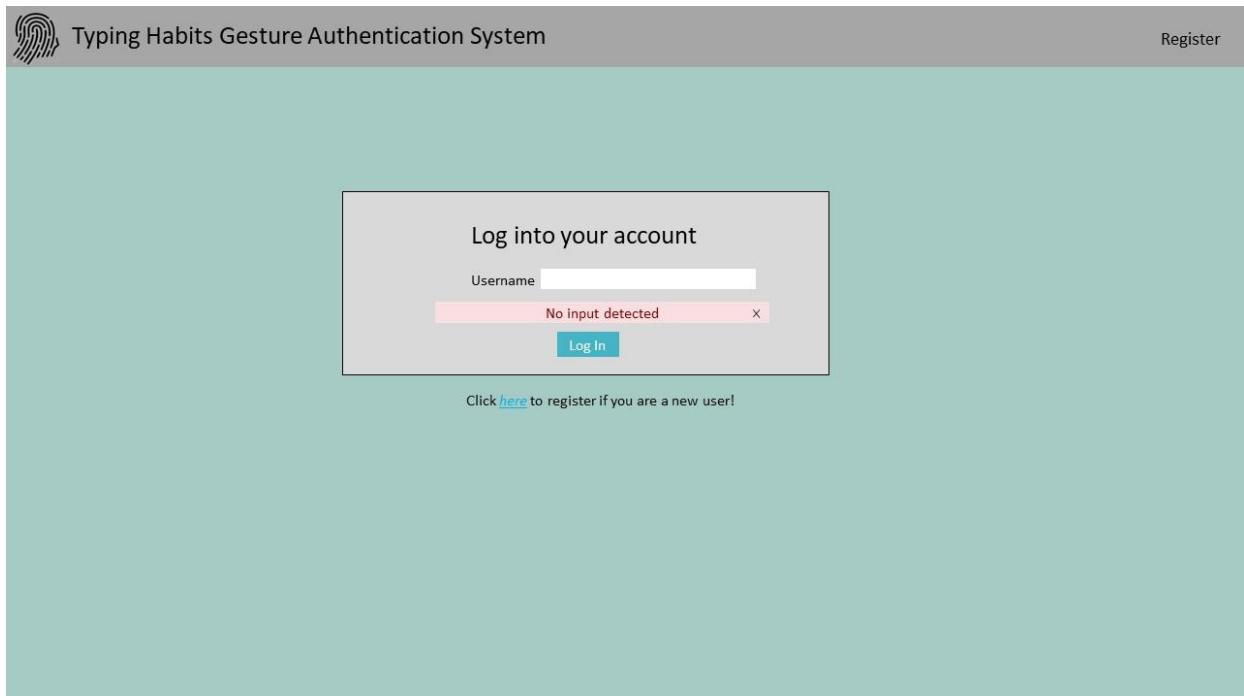


Figure 19. Wireframe showing if user clicks "Log In" button without typing any username.

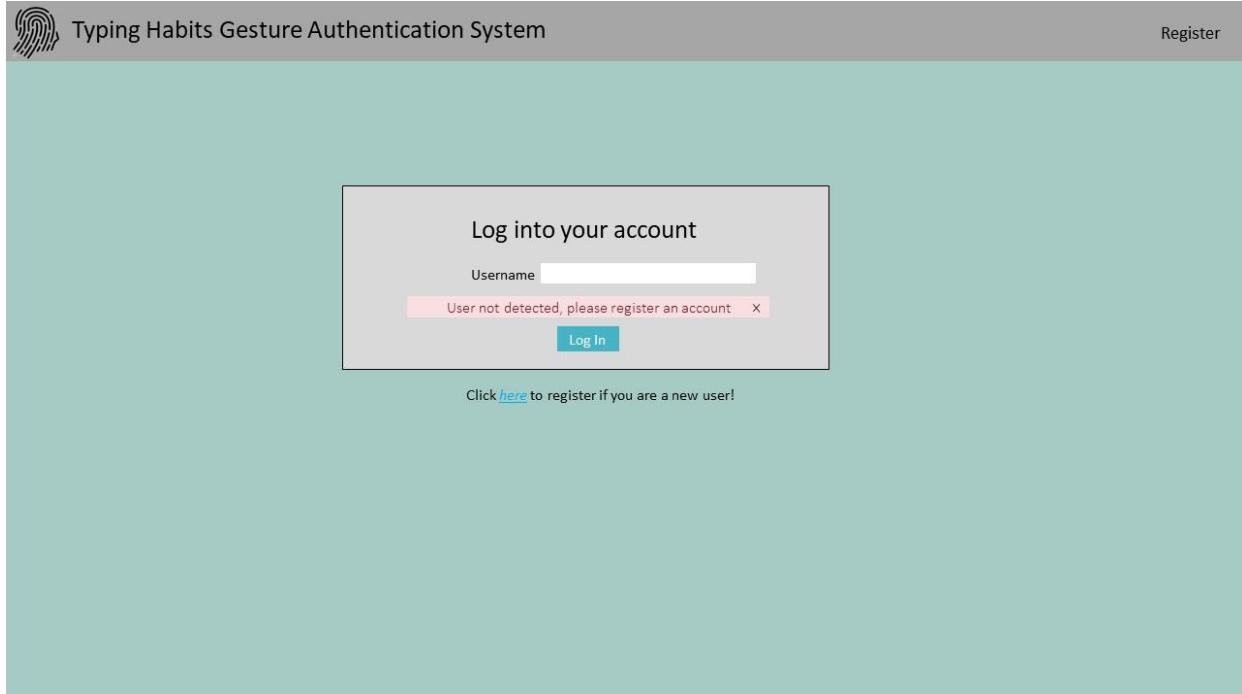


Figure 20. Wireframe showing if username is not registered with the system.

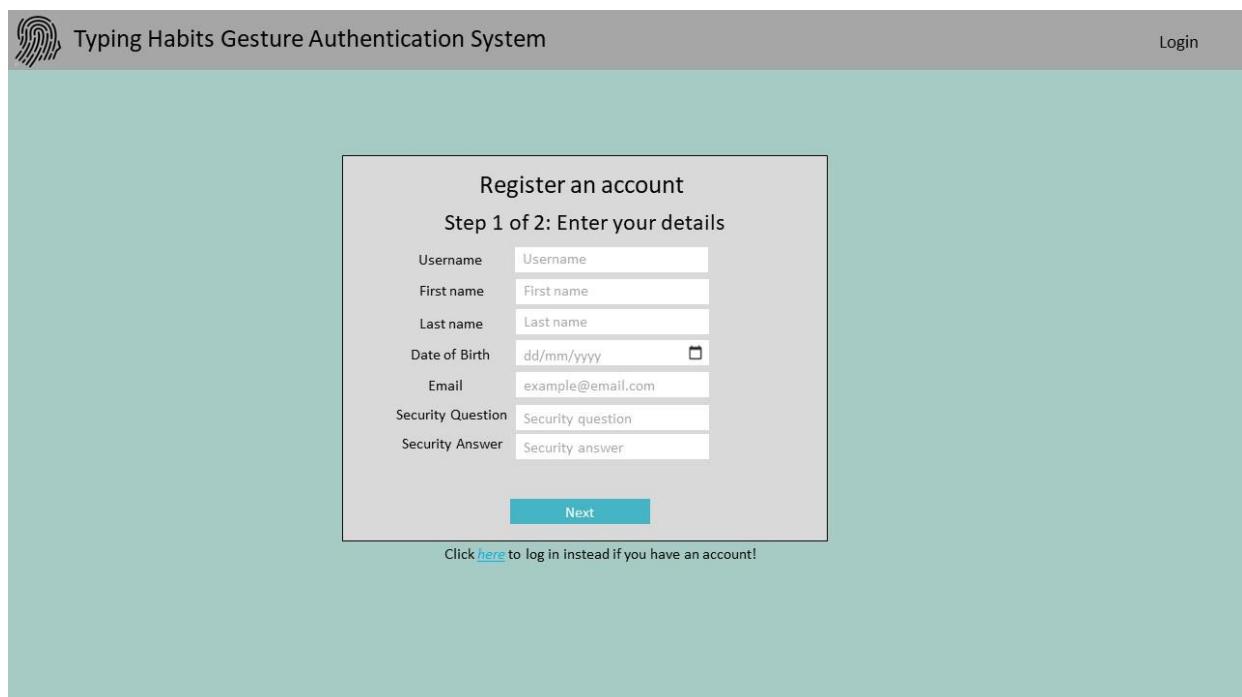


Figure 21. Wireframe showing step 1 of registering an account. User is required to enter username, first name, last name, date of birth, email, security question and security answer.

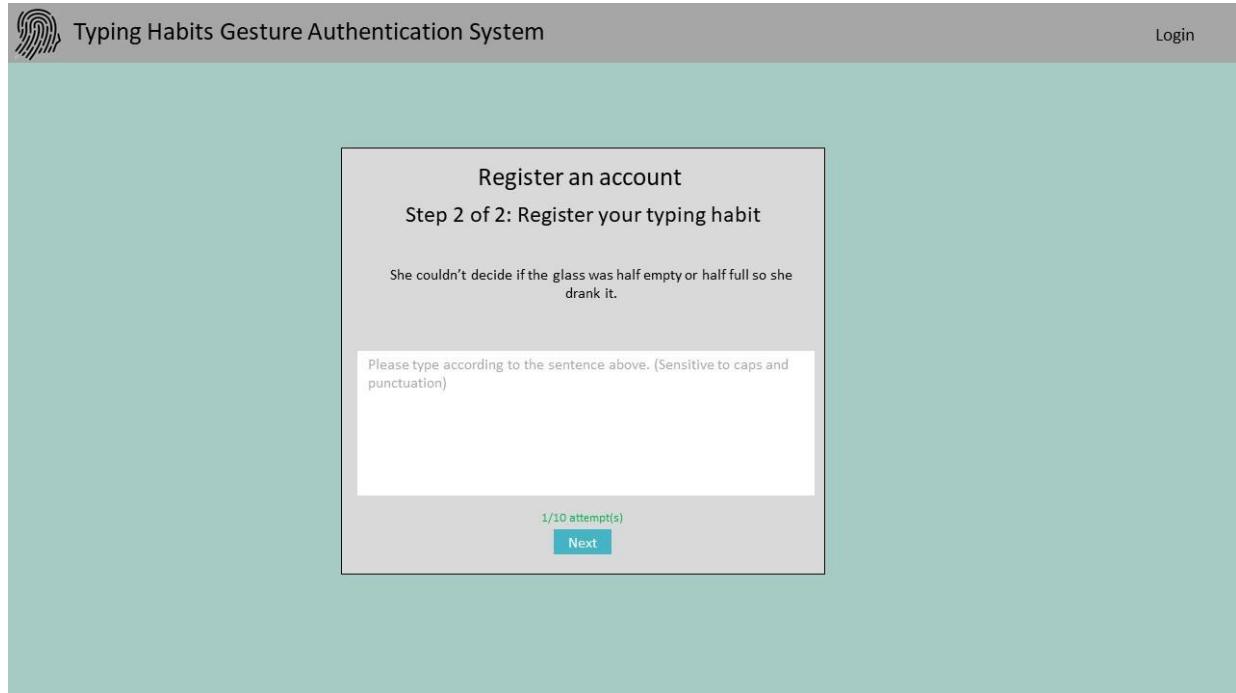
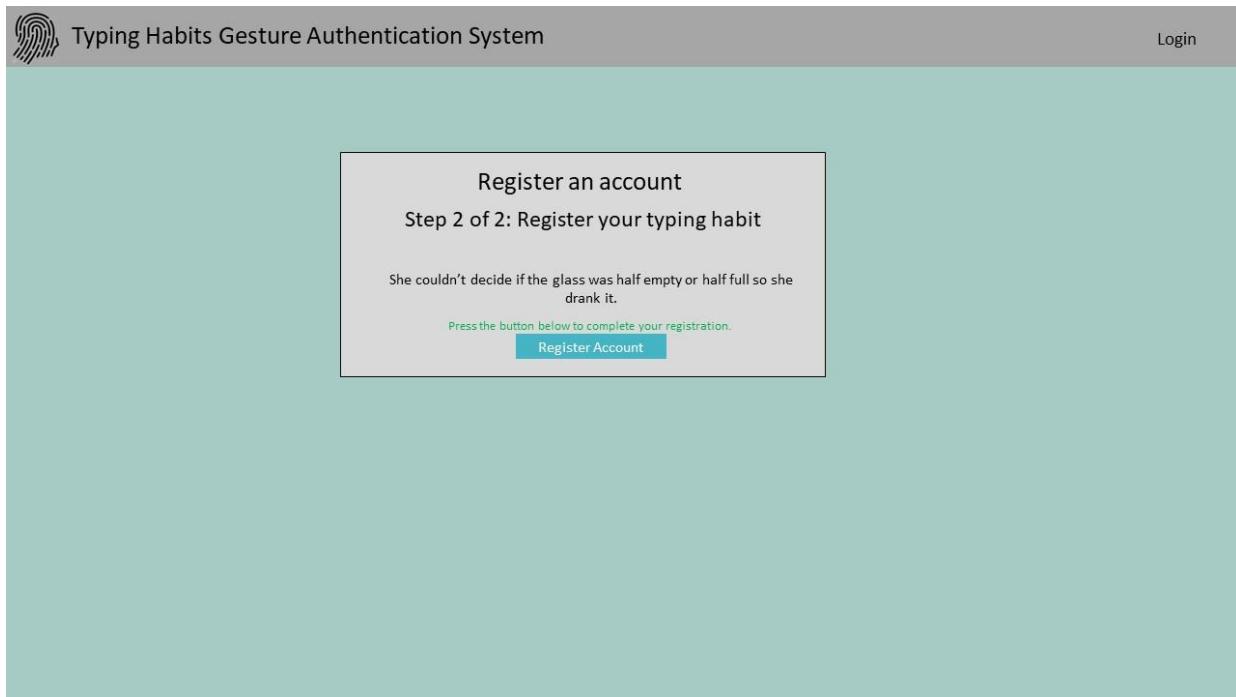


Figure 22. Wireframe showing step 2 of registering an account. User is required to type the sentence shown 10 times. There is a number in green below the area where the user types the sentence to show how many times more are required.



# Technical Manual

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

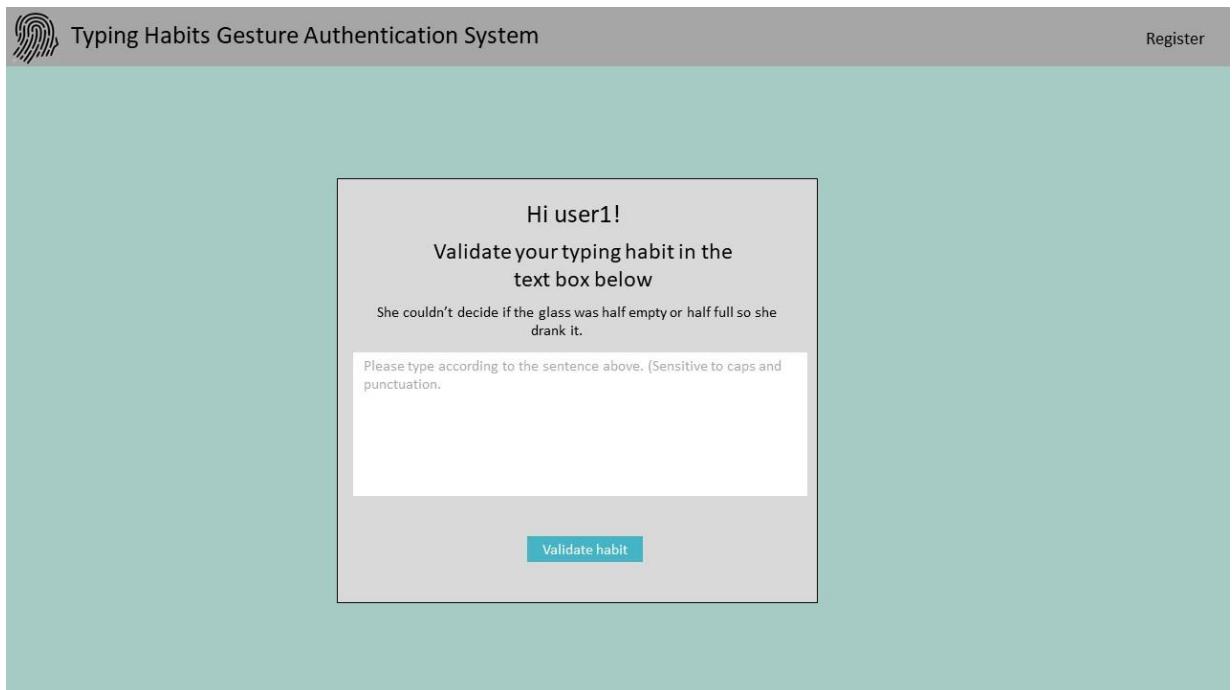
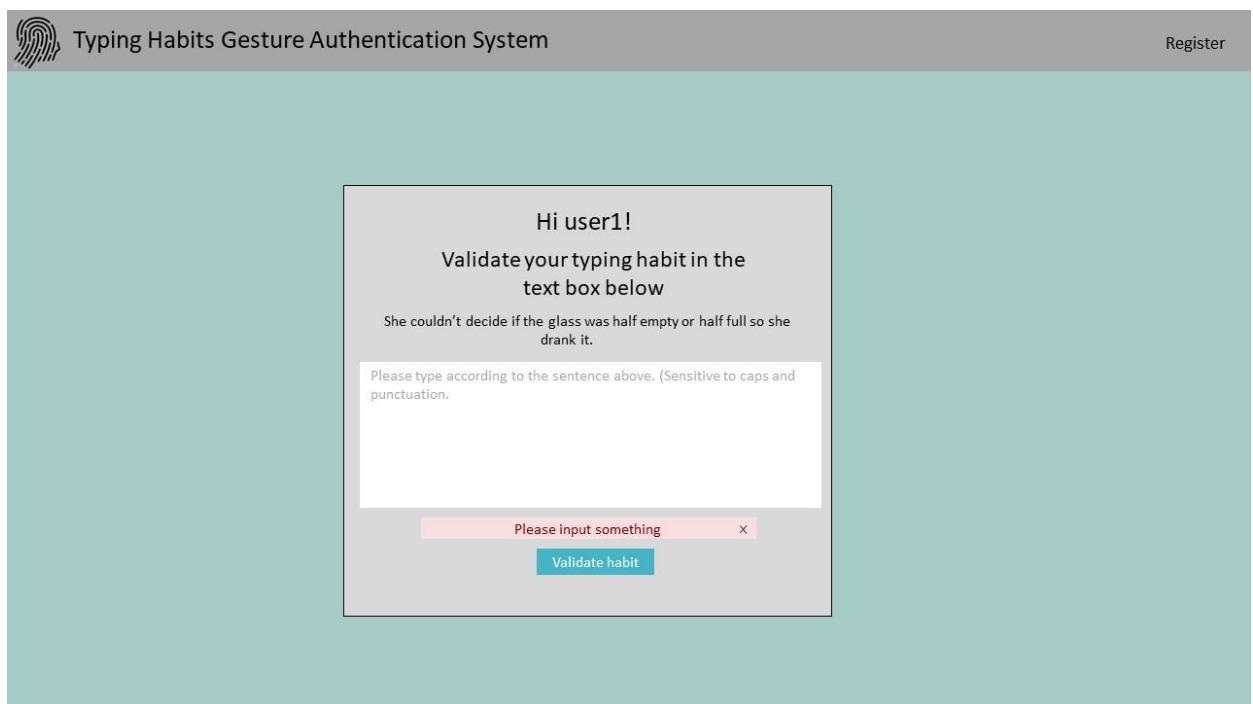


Figure 24. Wireframe showing the validation page that the user will see upon entering his username.



# Technical Manual

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

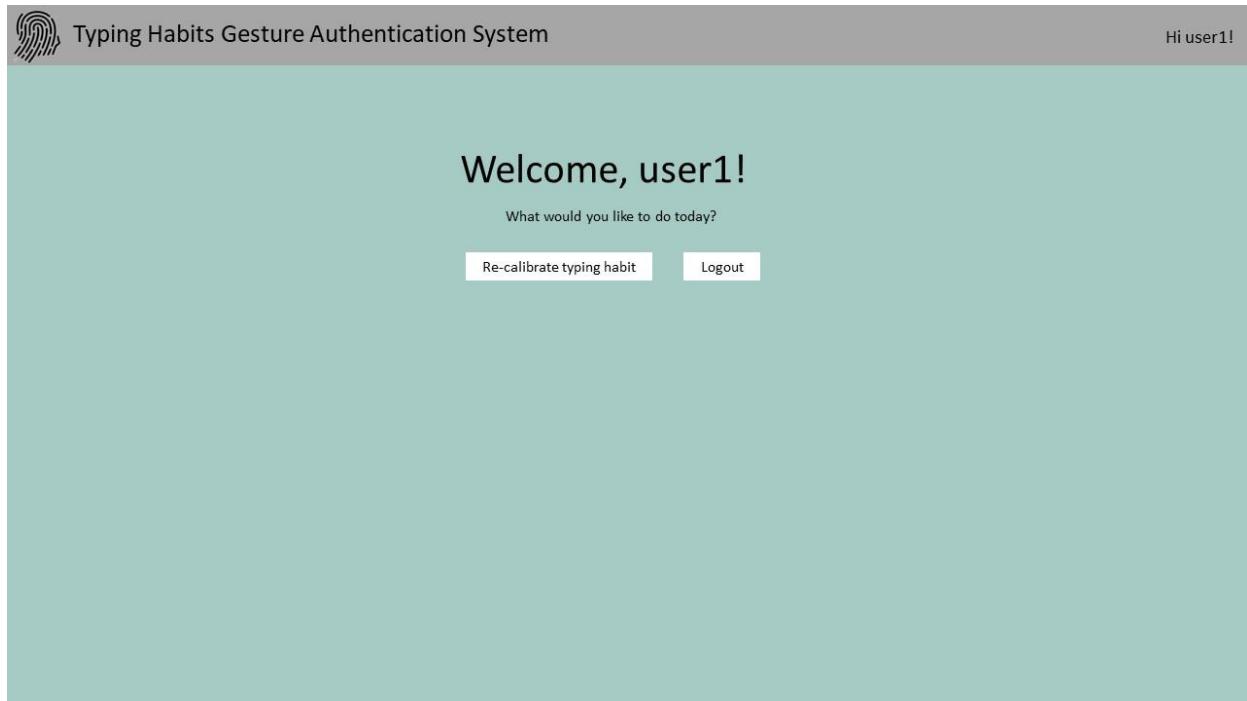
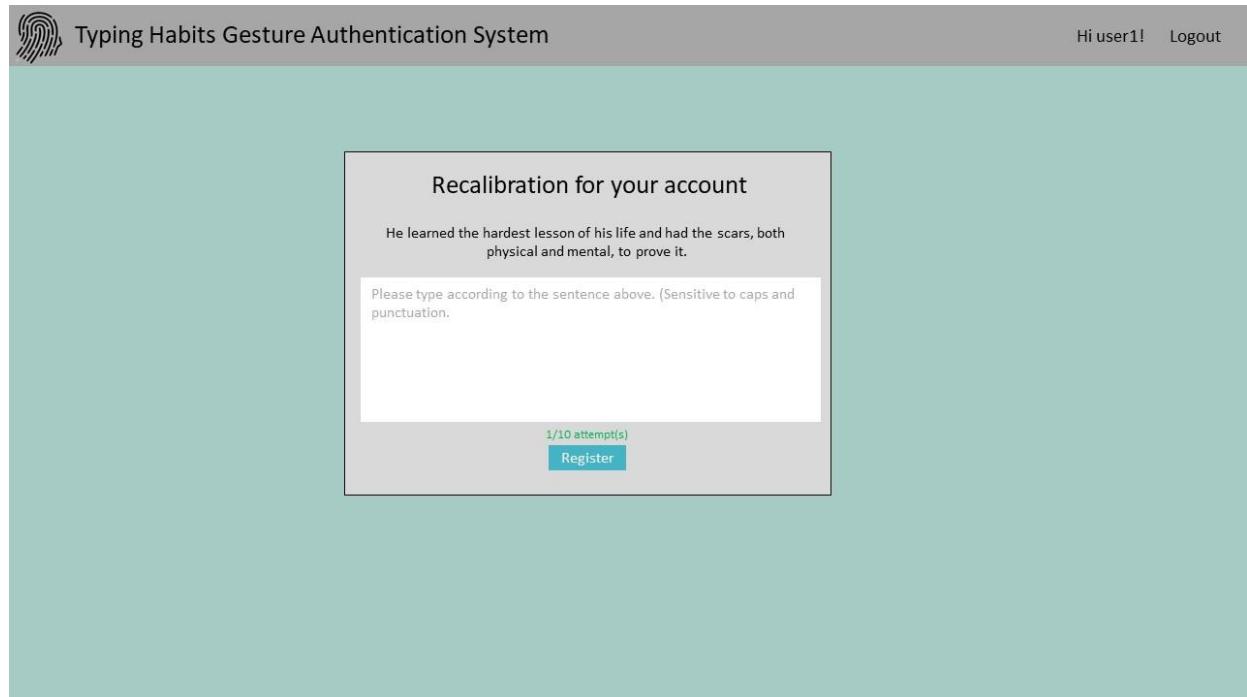


Figure 26. Wireframe showing the home page after successful validation of users typing habits.



# 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.

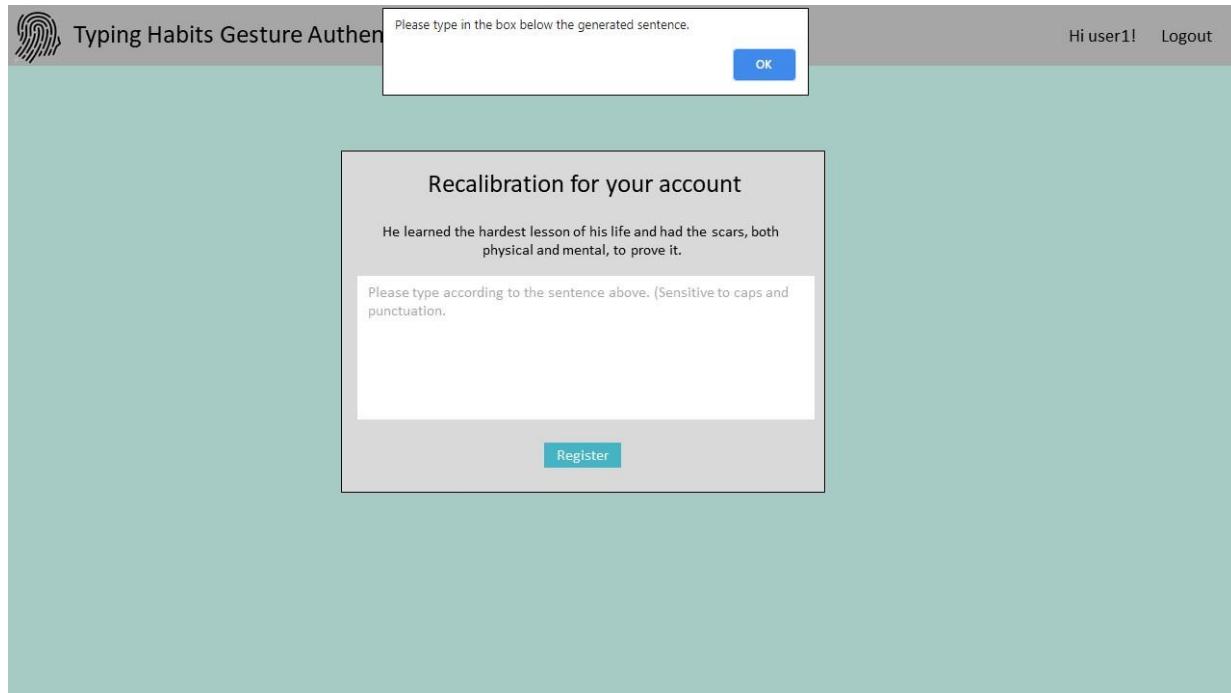
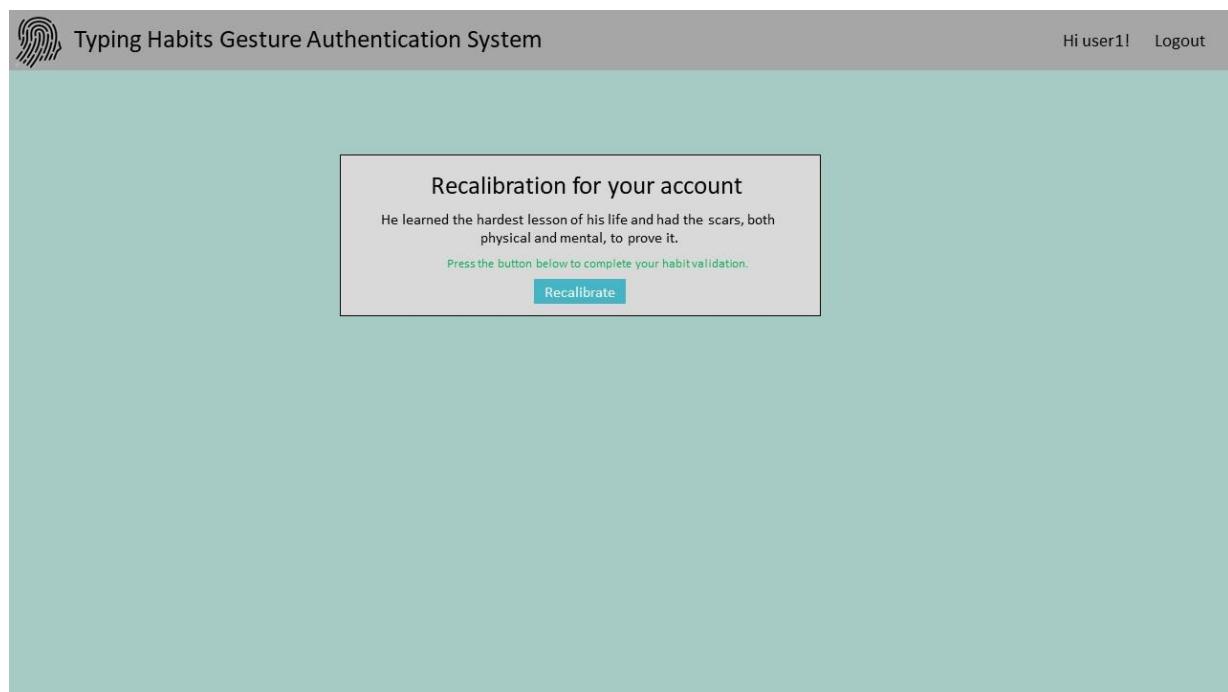


Figure 28. Wireframe showing the recalibration page if user does not type anything.



# Technical Manual

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

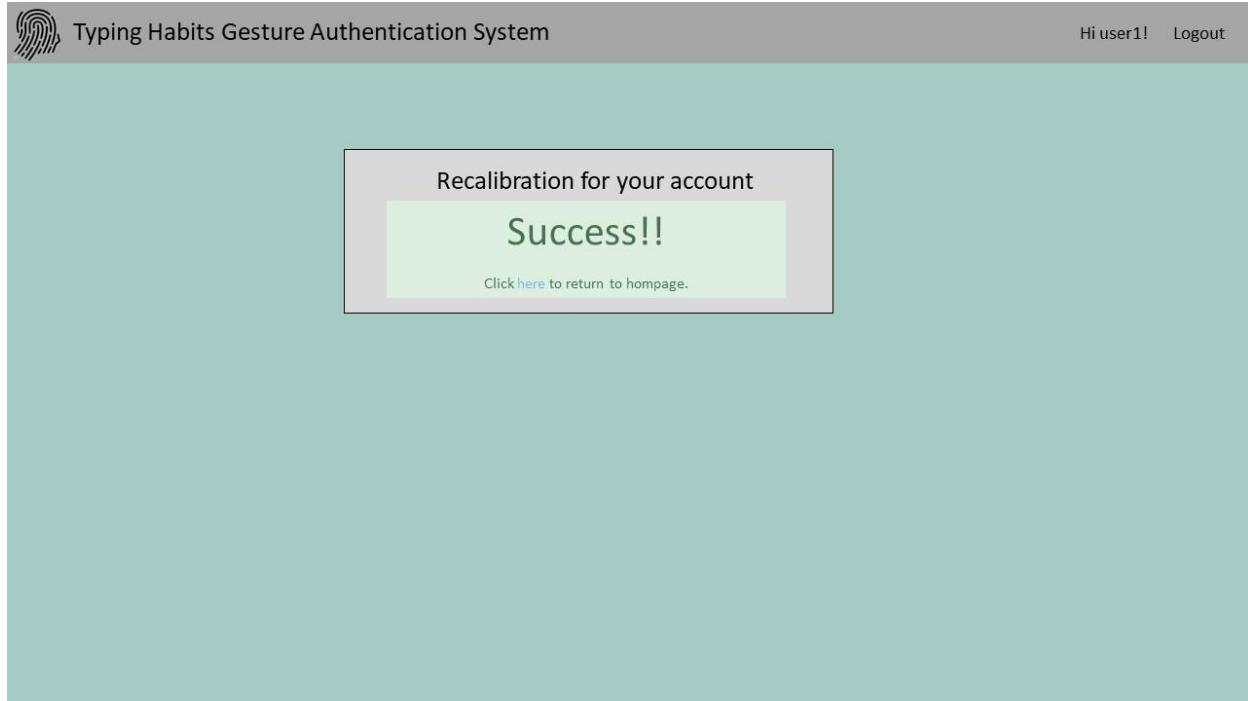
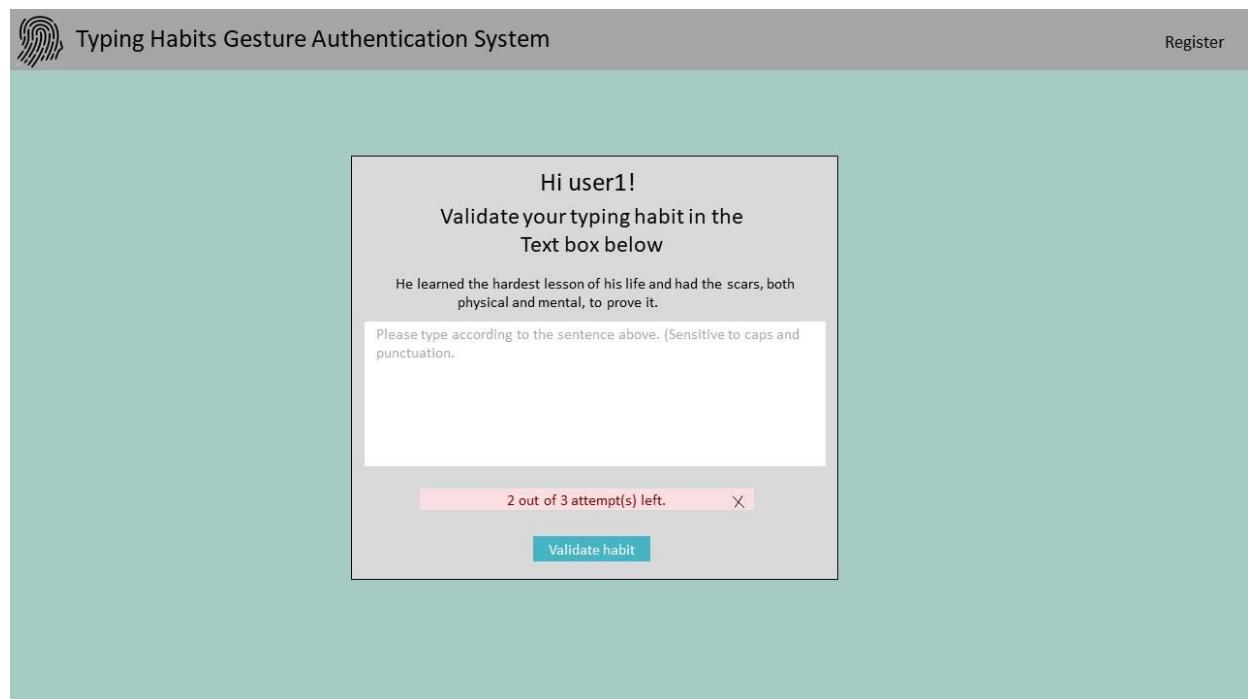


Figure 30. Wireframe showing successful recalibration of users typing habits.



# 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.

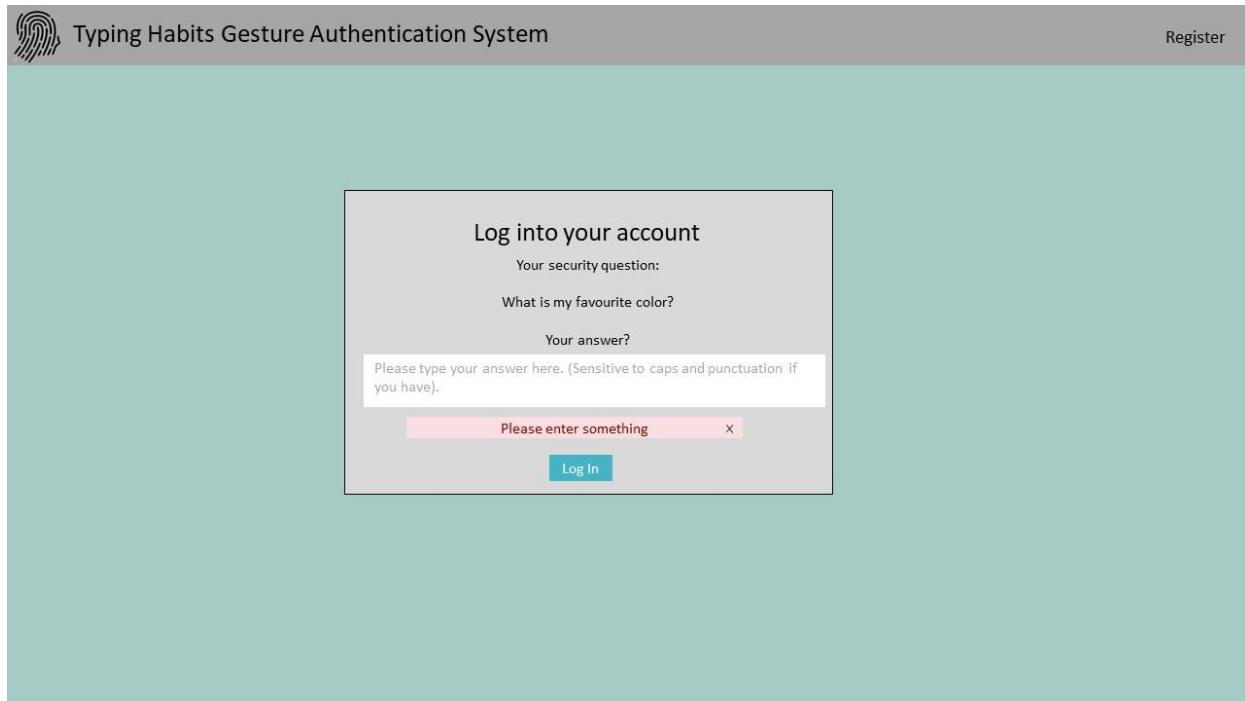


Figure 32. Wireframe showing after the 3 attempts to log in using typing habits are used up and user is redirected to the security question.

## 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.

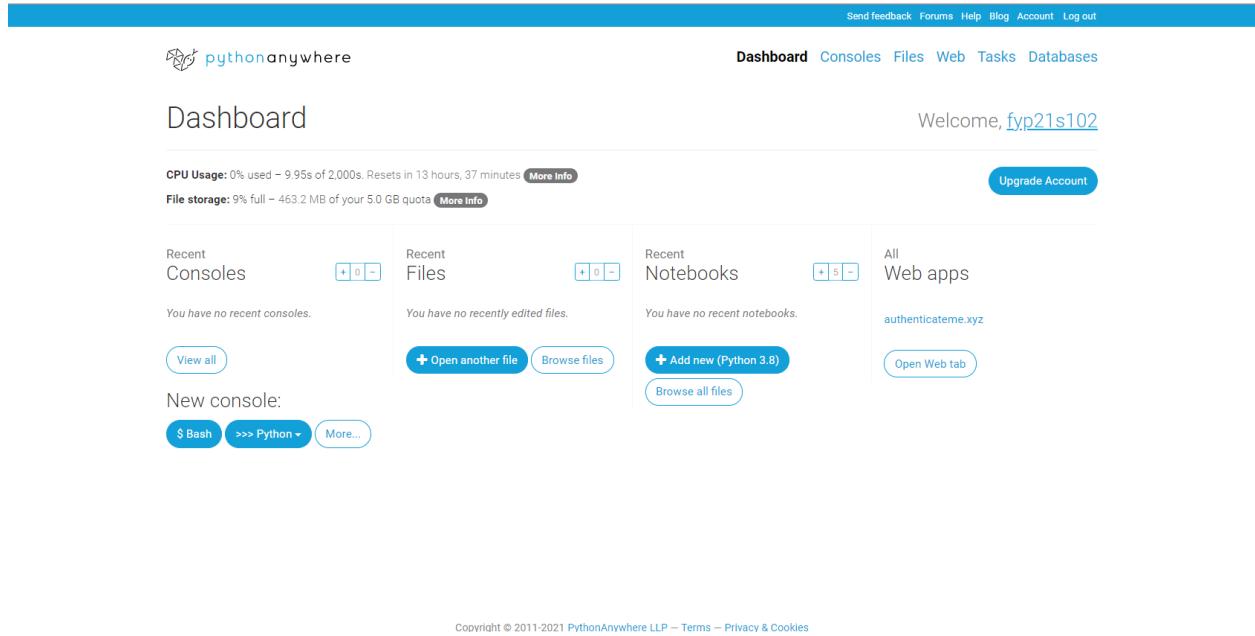


Figure 33. Screenshot of the hosting website (Pythonanywhere).

## 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.

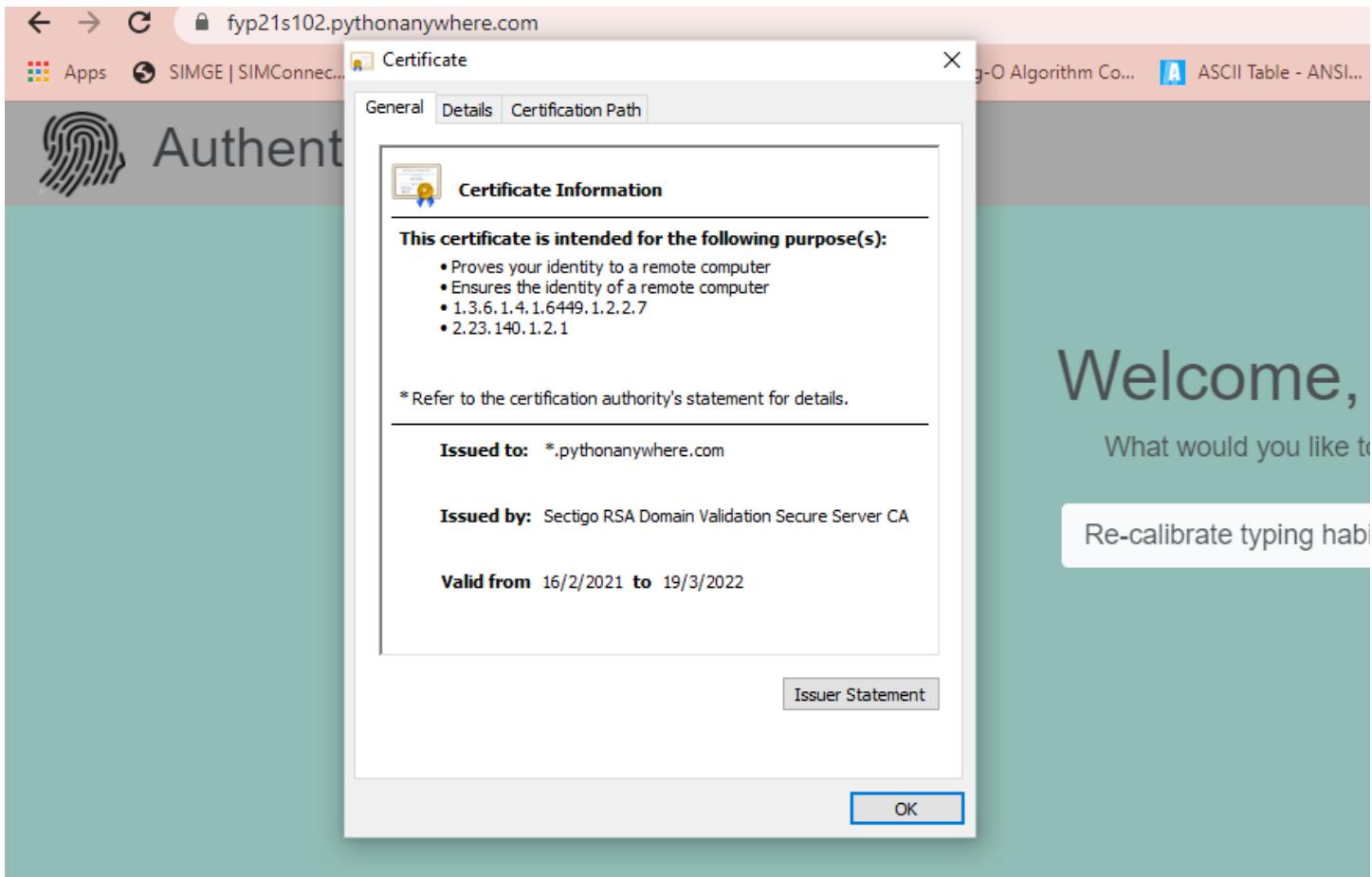


Figure 34. Screenshot of the SSL certificate.

# 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.).

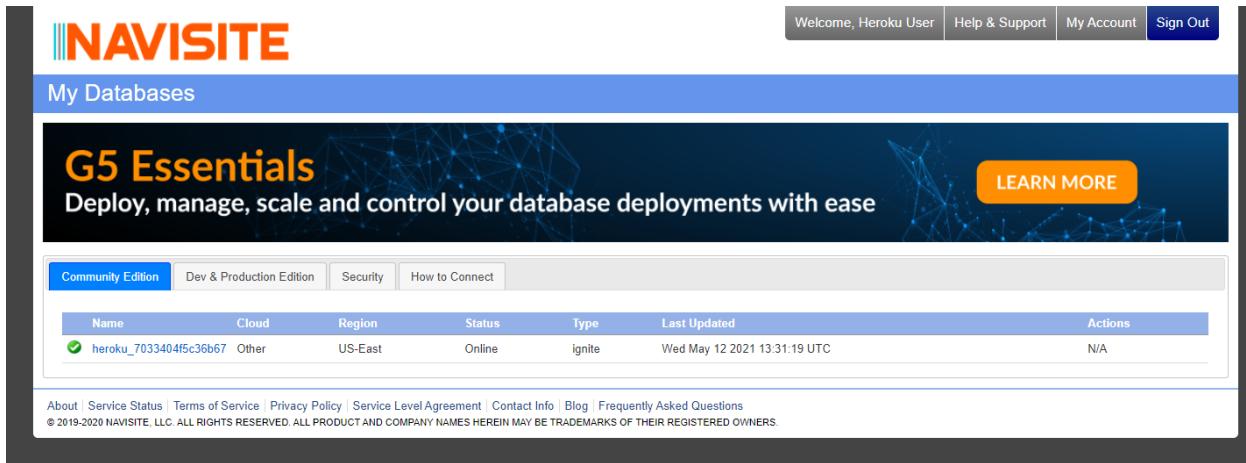
The screenshot shows the Google Cloud Platform Storage interface. The top navigation bar includes the 'Google Cloud Platform' logo, a dropdown menu for 'typinghabits', a search bar, and various status icons. The left sidebar has a 'Cloud Storage' section with 'Browser' selected, and links for 'Monitoring' and 'Settings'. The main content area is titled 'Bucket details' for 'typing-habit'. It features tabs for 'OBJECTS' (selected), 'CONFIGURATION', 'PERMISSIONS', 'RETENTION', and 'LIFECYCLE'. Below these tabs is a breadcrumb trail: 'Buckets > typing-habit'. There are buttons for 'UPLOAD FILES', 'UPLOAD FOLDER', 'CREATE FOLDER', 'MANAGE HOLDS', 'DOWNLOAD', and 'DELETE'. A filter bar allows filtering by name prefix. The main table lists objects in the bucket, with columns for Name, Size, Type, Created time, Storage class, Last modified, Public access, Encryption, and Retention expiration date. Each row shows a folder named after a user, such as 'KimHao/' through 'tar?/'. The table has a header row and approximately 15 data rows. A vertical scrollbar is visible on the right side of the table.

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.



The screenshot shows the NAVISITE database management interface. At the top, there is a navigation bar with links for "Welcome, Heroku User", "Help & Support", "My Account", and "Sign Out". Below the navigation bar, the title "NAVISITE" is displayed in large orange letters. A blue header bar contains the text "My Databases" and a promotional banner for "G5 Essentials" with the subtext "Deploy, manage, scale and control your database deployments with ease" and a "LEARN MORE" button. The main content area is a table listing databases. The table has columns for Name, Cloud, Region, Status, Type, and Last Updated. One row is visible, showing "heroku\_7033404f5c36b67" as the Name, "Other" as the Cloud, "US-East" as the Region, "Online" as the Status, "ignite" as the Type, and "Wed May 12 2021 13:31:19 UTC" as the Last Updated date. The Actions column shows "N/A". At the bottom of the page, there is a footer with links to "About", "Service Status", "Terms of Service", "Privacy Policy", "Service Level Agreement", "Contact Info", "Blog", and "Frequently Asked Questions". It also includes a copyright notice: "© 2019-2020 NAVISITE, LLC. ALL RIGHTS RESERVED. ALL PRODUCT AND COMPANY NAMES HEREIN MAY BE TRADEMARKS OF THEIR REGISTERED OWNERS."

Name	Cloud	Region	Status	Type	Last Updated	Actions
heroku_7033404f5c36b67	Other	US-East	Online	ignite	Wed May 12 2021 13:31:19 UTC	N/A

Figure 36. Screenshot of the database in ClearDB.

## 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.

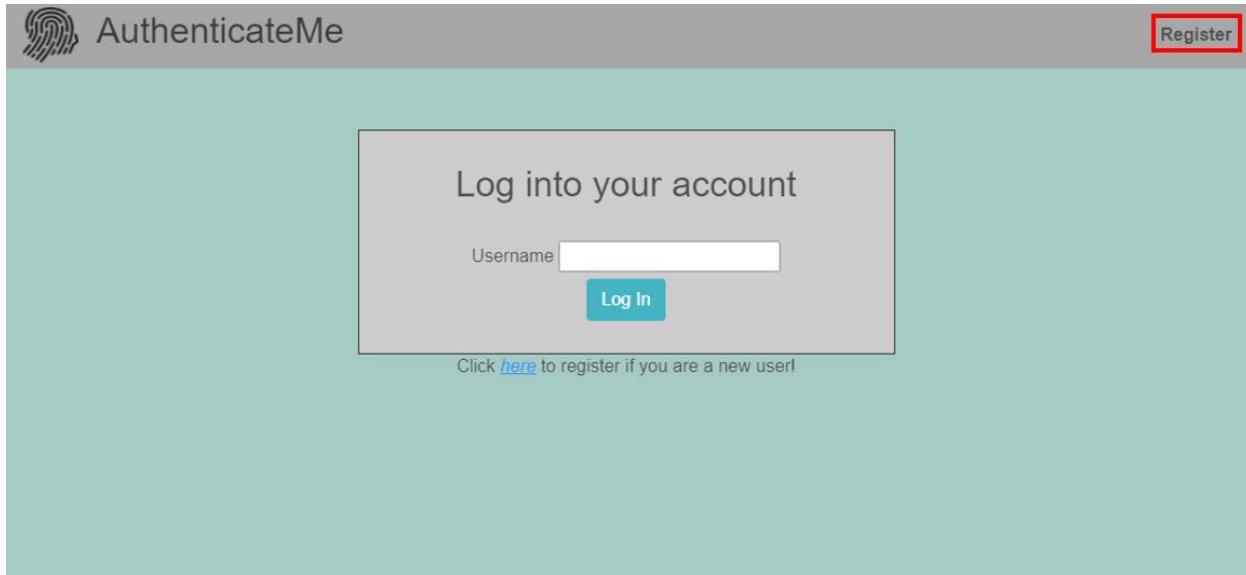


Figure 37. Screenshot from AuthenticateMe website showing the 'Register' button.

# Technical Manual

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

The screenshot shows a web page titled 'AuthenticateMe' with a logo of a fingerprint. In the top right corner, there is a 'Login' link. The main content area has a light gray background and contains a form for 'Register an account'. The form is labeled 'Step 1 of 2: Enter your details'. It includes fields for 'Username', 'First name', 'Last name', 'Date of Birth' (with a date input field and a calendar icon), 'Email' (containing 'example@email.com'), 'Security Question' (containing 'Security question'), and 'Security Answer' (containing 'Security answer'). A large blue button at the bottom of the form is labeled 'Next'. This 'Next' button is highlighted with a red rectangular border. Below the form, a small note says 'Click [here](#) to log in instead if you have an account!'. The overall background of the page is a teal color.

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

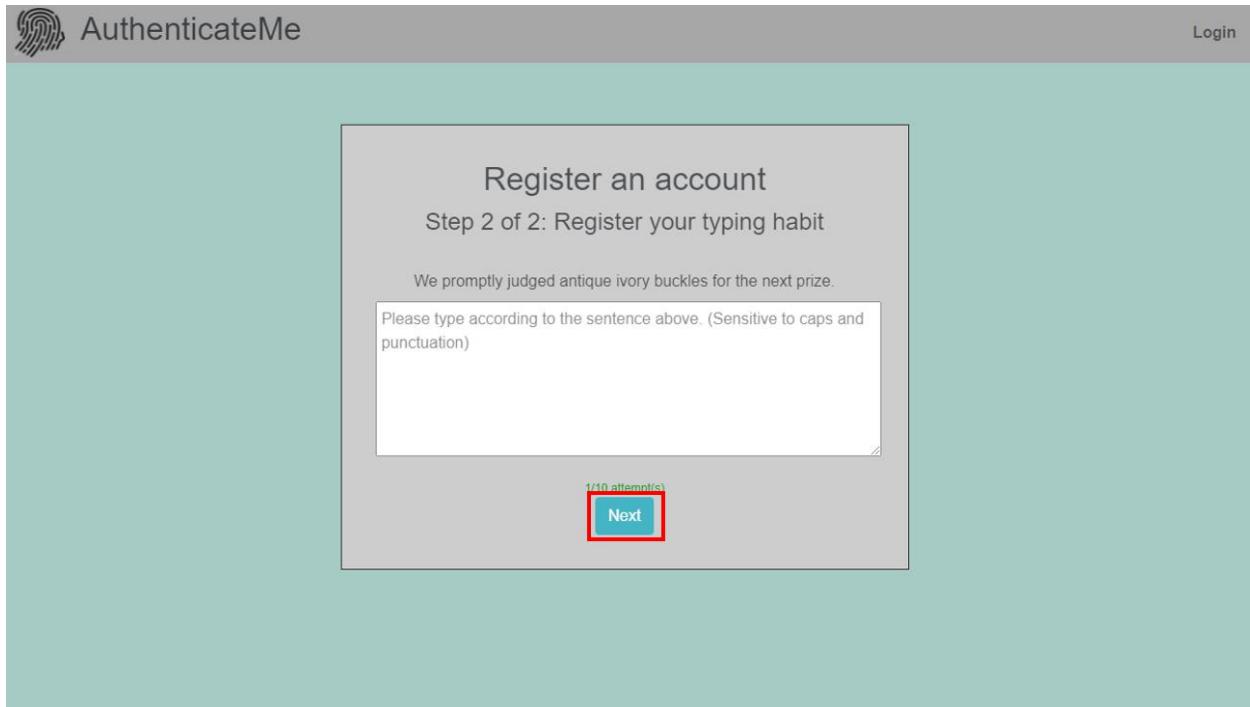


Figure 39. Screenshot from AuthenticateMe website showing step 2 of registering an account and the 'Next' button.

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

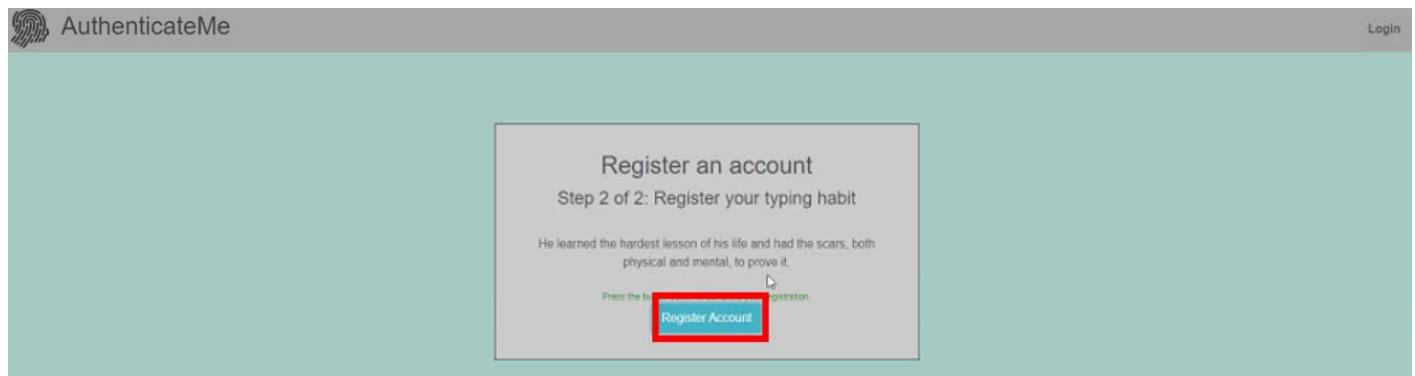


Figure 40. Screenshot from AuthenticateMe website showing step 2 of registering an account and the 'Register Account' button.

# Technical Manual

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

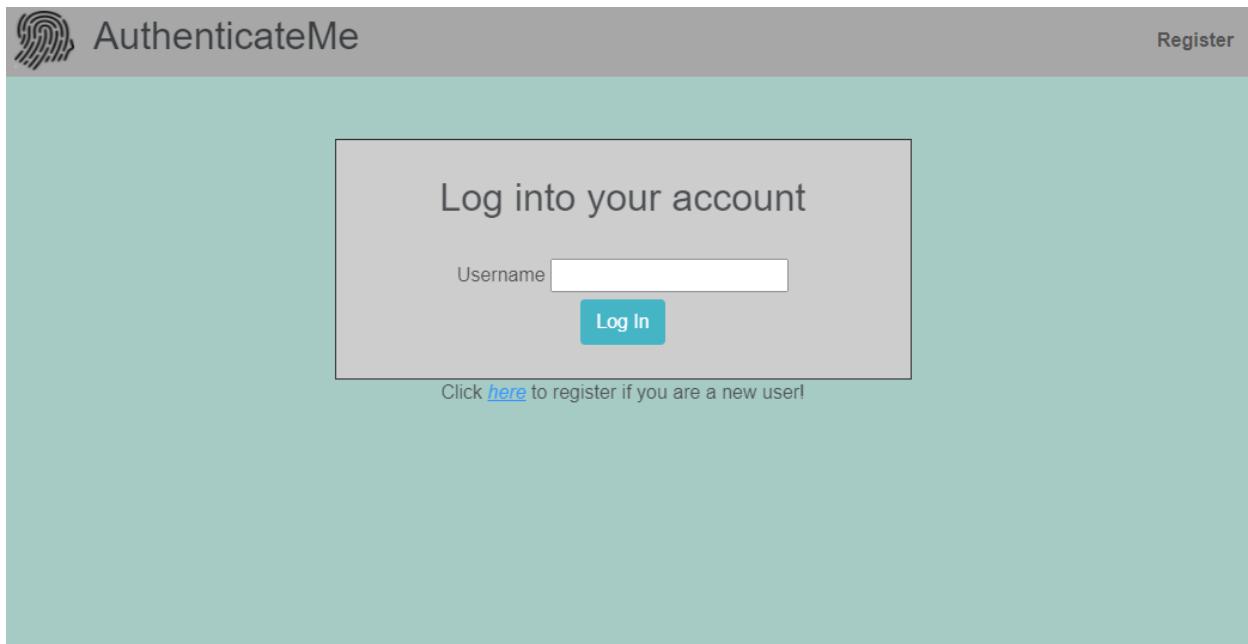


Figure 41. Screenshot from AuthenticateMe website showing step 2 of registering an account and the 'Register Account' button.

## 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.

The screenshot shows the 'AuthenticateMe' login page. At the top, there is a logo of a fingerprint and the text 'AuthenticateMe'. On the right side, there is a 'Register' button. The main area has a light gray background with a centered 'Log into your account' heading. Below it is a 'Username' input field. A pink error message box contains the text 'No input detected' with a close button 'x'. Below the input field is a blue 'Log In' button. At the bottom of the form, there is a link 'Click [here](#) to register if you are a new user!'

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.

The screenshot shows the 'AuthenticateMe' login page. At the top, there is a logo of a fingerprint and the text 'AuthenticateMe'. On the right side, there is a 'Register' button. The main area has a light gray background with a centered 'Log into your account' heading. Below it is a 'Username' input field. A pink error message box contains the text 'User not detected, please register an account.' with a close button 'x'. Below the input field is a blue 'Log In' button. At the bottom of the form, there is a link 'Click [here](#) to register if you are a new user!'

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

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

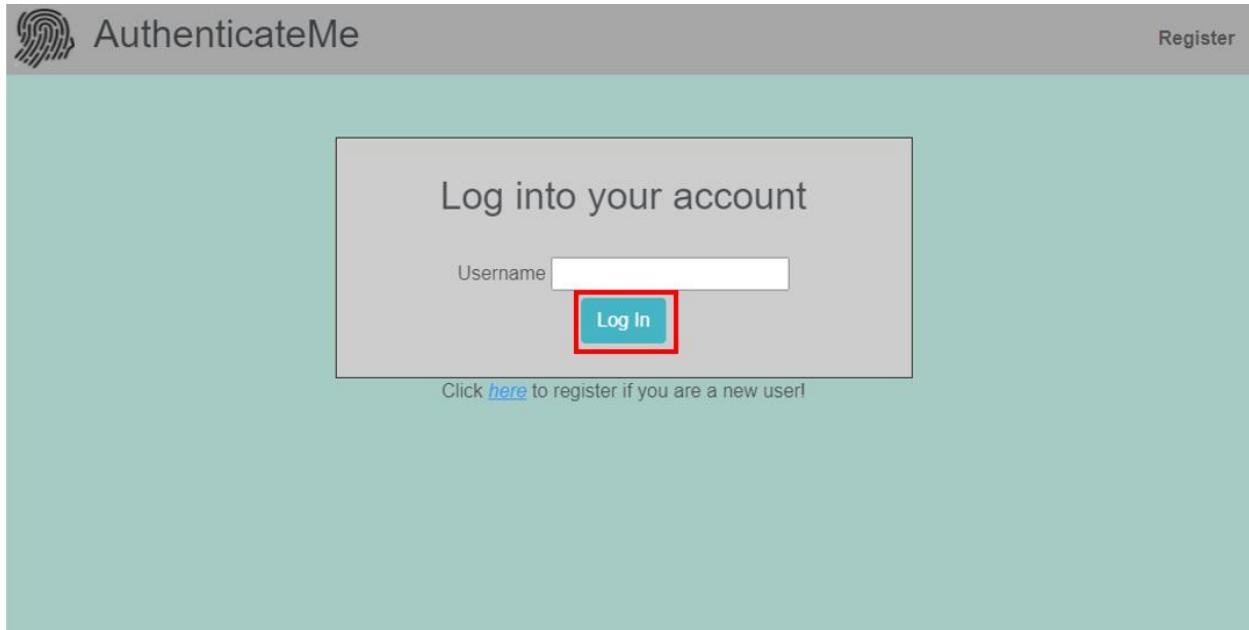


Figure 44. Screenshot from AuthenticateMe website showing the 'Log In' button.

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.

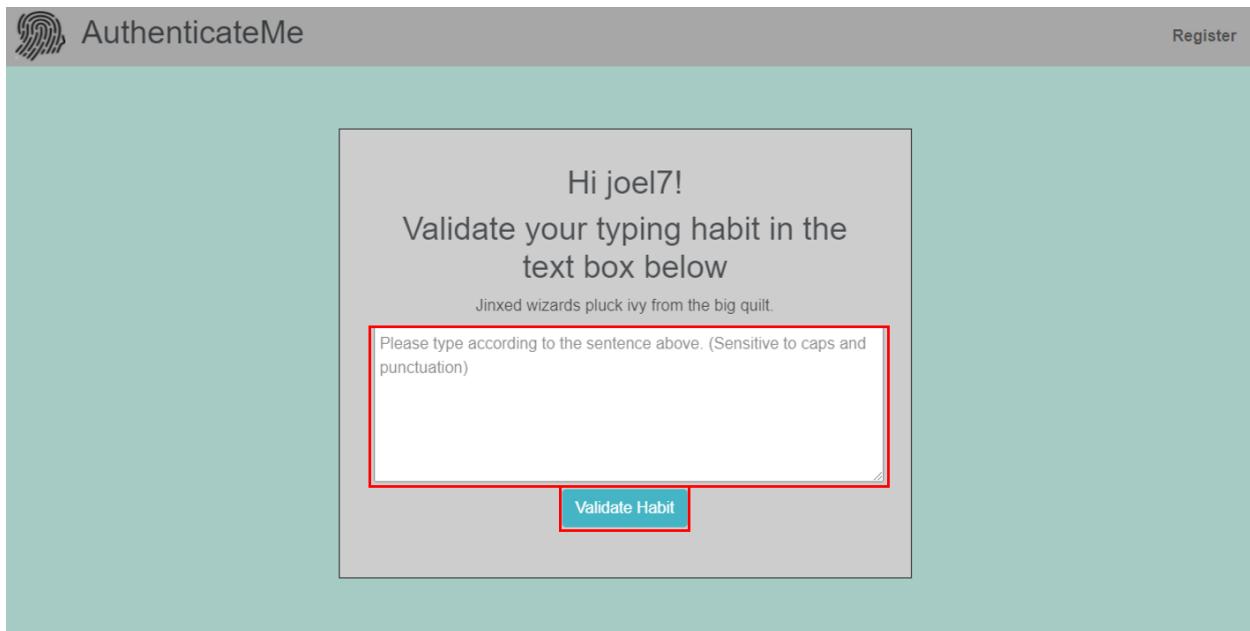


Figure 45. Screenshot from AuthenticateMe website showing the area which the user can type the sentence and the 'Validate Habit' button.

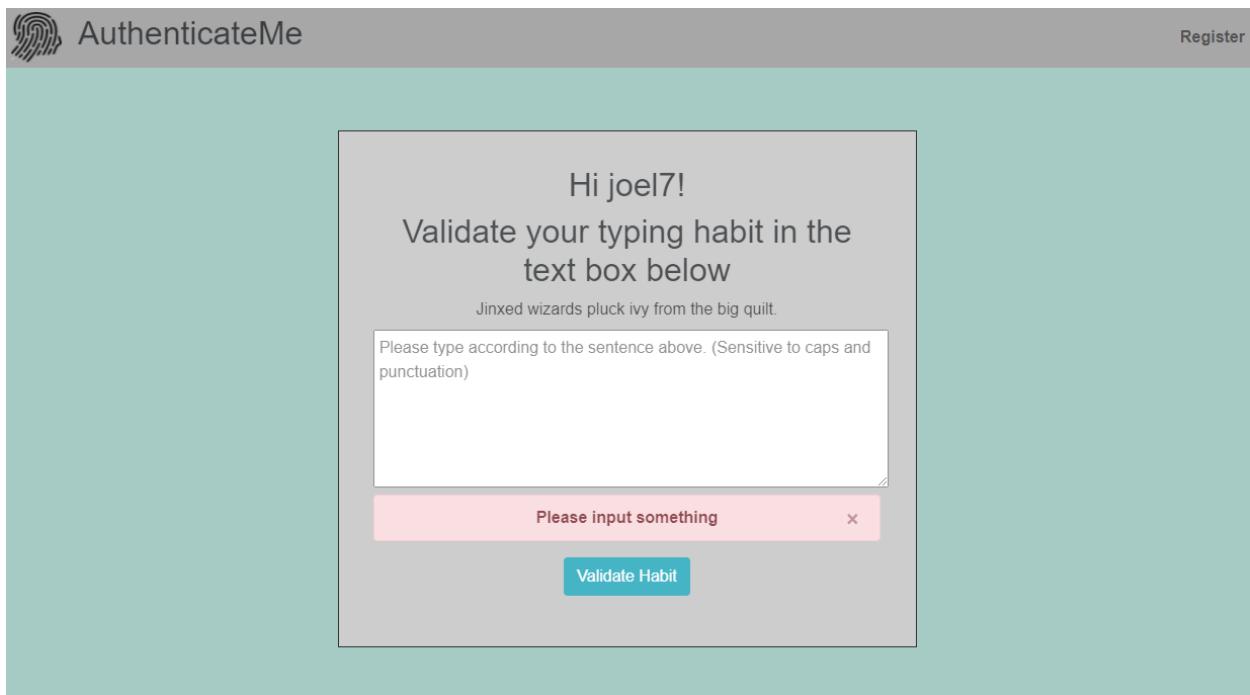


Figure 46. Screenshot from AuthenticateMe website showing an error if user clicks on the 'Validate Habit' button while the box is empty.

## **Technical Manual**

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

## 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.

The screenshot shows a login form titled "Log into your account". At the top, it asks "Your security question: What is my favourite color?". Below that, it asks "Your Answer:" and provides a text input field with the placeholder "Please type your answer here. (Sensitive to caps and punctuation if you have)". A blue "Log In" button is at the bottom of the form. The background is light grey, and the entire form is contained within a white rectangular box.

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.

The screenshot is identical to Figure 47, showing the login form with the security question and answer fields. However, the answer field now contains a red error message: "Please enter something" followed by a small "x" icon. The rest of the form and background are the same as in Figure 47.

## **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.

## 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.

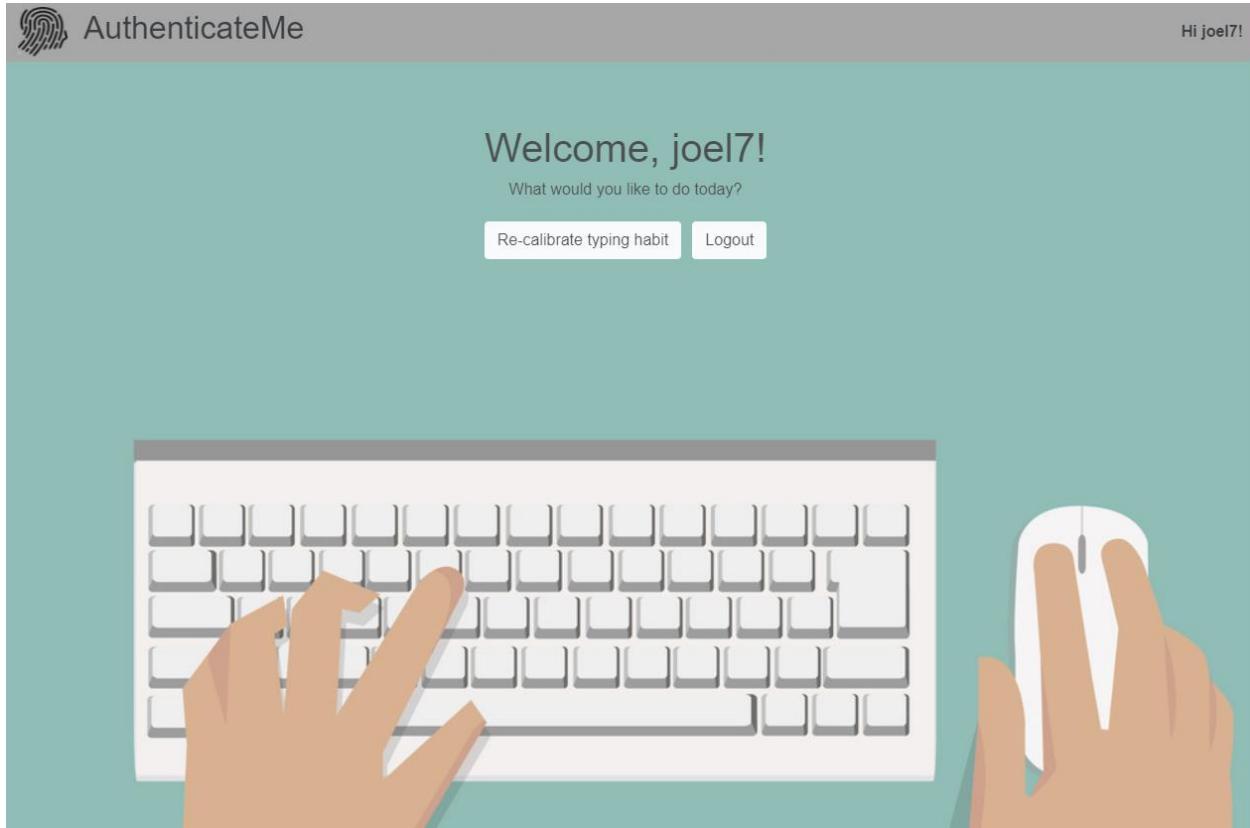


Figure 49. Screenshot from AuthenticateMe website showing the homepage.

### 7.2.4.1 Calibrations Page

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

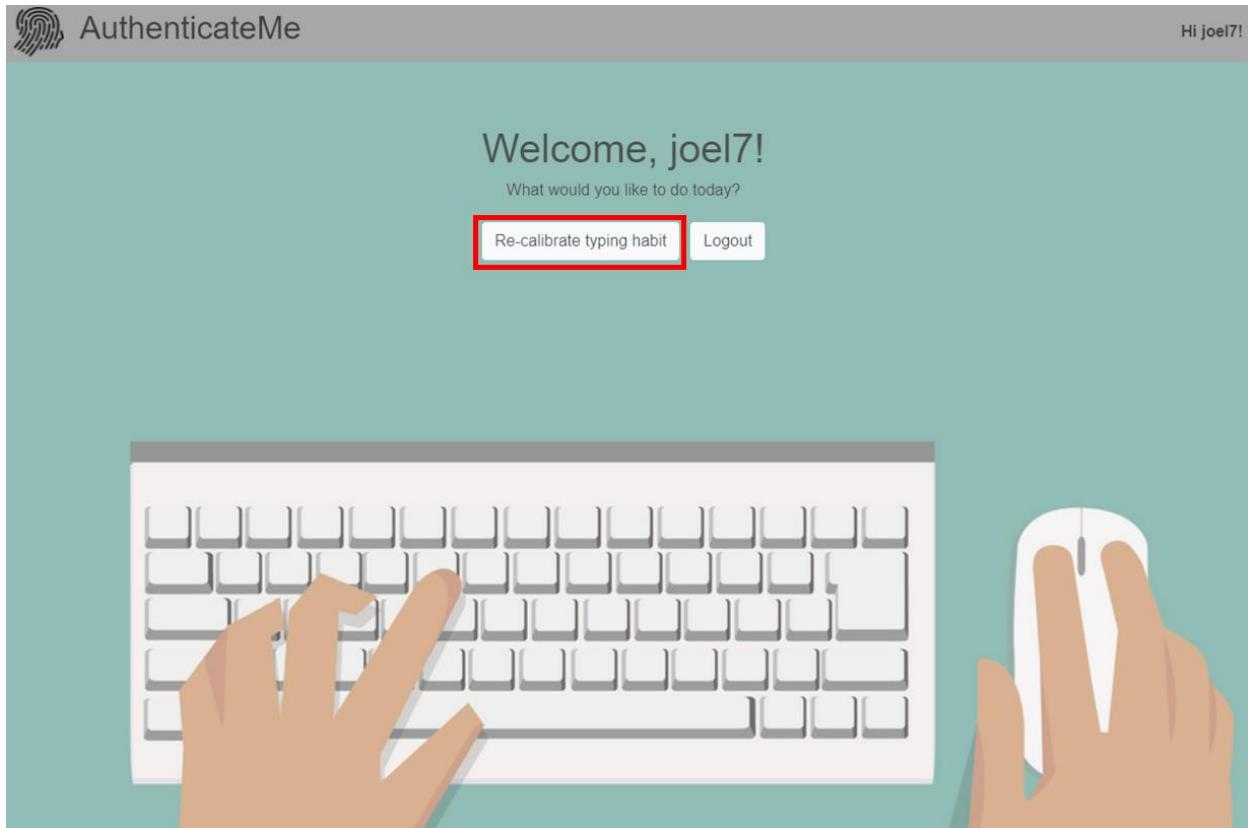


Figure 50. Screenshot from AuthenticateMe website showing the 'Recalibrate typing habit' button.

When users choose to recalibrate, 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.

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

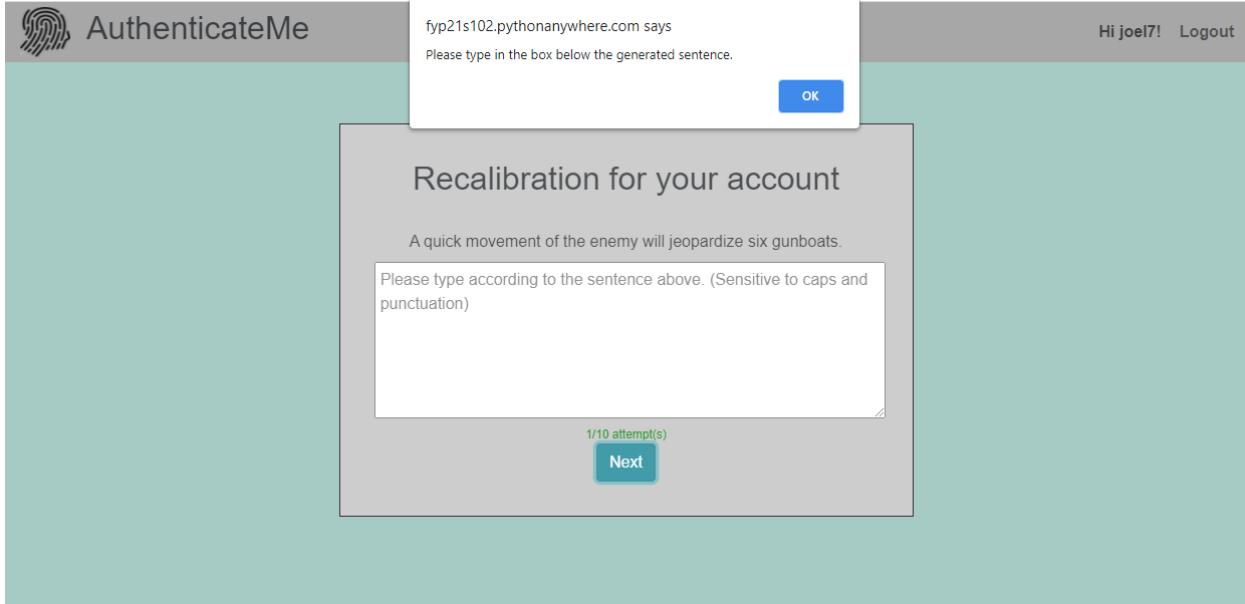


Figure 52. Screenshot from AuthenticateMe website showing an error if the user clicks on the 'Next' button while the box is empty.

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.

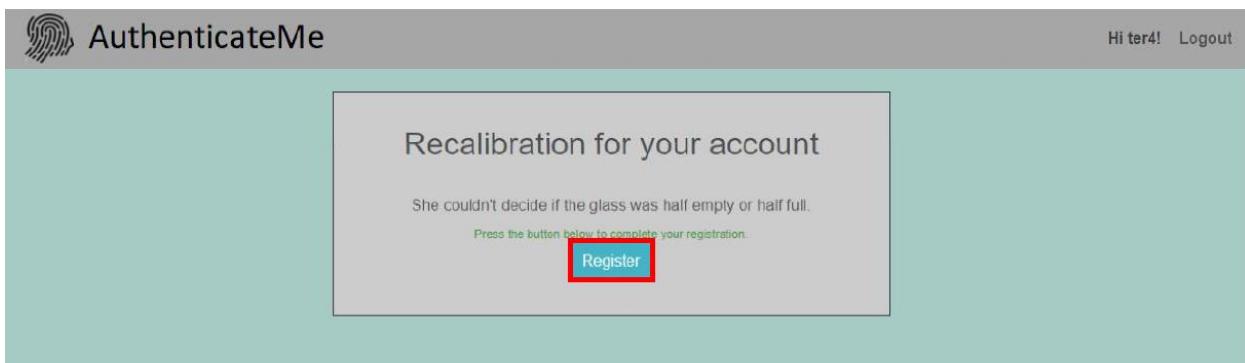


Figure 53. Screenshot from AuthenticateMe website showing the 'Recalibrate' button.

# 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.

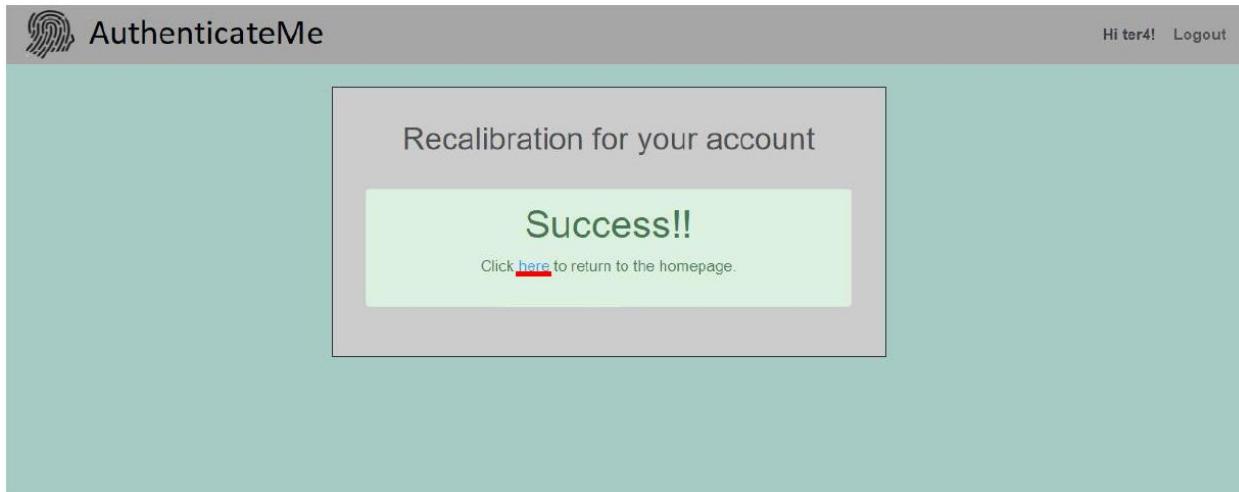


Figure 54. Screenshot from AuthenticateMe website showing successful recalibration of users typing habits.

## 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.

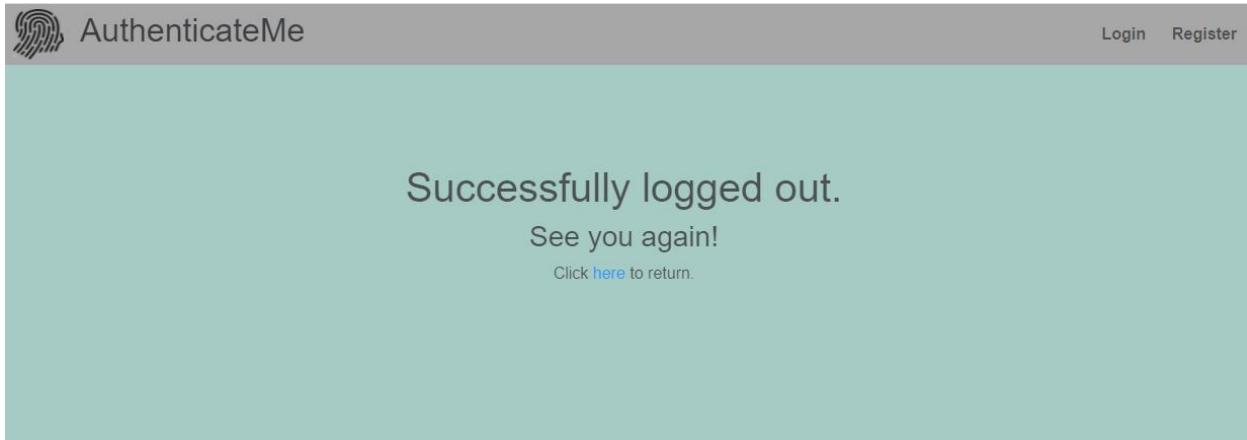


Figure 55. Screenshot from AuthenticateMe website showing that the user has successfully logged out.

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

# Technical Manual

## 7.3. Website Application Testing

### 7.3.1. Sprint 1

<b>Test Plan:</b>	Test Plan Sprint 1							
<b>Project Name:</b>	Behavioural Password Protection	Typing Habit Gesture Authentication						
<b>Test Case ID:</b>	TC-2	<b>Test Case Description:</b>	As a user, they should be able to log out so that they can exit the application safely.					
<b>Created By:</b>	Cheah Cui Hui	<b>Reviewed By:</b>	Terrence Yap				<b>Version</b>	2.0
<b>Tester's Name:</b>	Alicia Tan	<b>Date Tested:</b>	19/5/2021	<b>Test Case (Pass/Fail)</b>	Pass			
<b>No.</b>	<b>Prerequisites</b>			<b>Test Scenario No.</b>	<b>Test Data</b>			
1	Access to Chrome Browser			1	username used for testing: alitcy			
2	Login to Typing Habit Gesture Authentication System							
3								
4								
<b>Test Scenario 1:</b>	Verifying Valid Username and Typing Habit. User can login. User can logout.							
<b>Test Step</b>	<b>Test Step Description</b>	<b>Expected Results</b>	<b>Actual Results</b>	<b>Pass/Fail</b>	<b>Remarks</b>			
TC-2-1	Navigate to Homepage website	Site should open	Site loaded and displayed	Pass				
TC-2-2	Enter Username	Credential can be entered	Able to enter username	Pass				
TC-2-3	Click "Log in"	User is navigated to Typing Habit Verification site	Able to validate username	Pass				
TC-2-4	Enter sentence generated using correct typing habit	Credential can be entered	Able to type	Pass				
TC-2-5	Click "Log in"	User is logged in	Able to validate typing habit	Pass	Homepage is displayed with the Username.			
TC-2-6	Click "Logout"	User is logged out	Able to log out	Pass	Logout page is displayed with the "Successfully logged out" message.			

Figure 56. Diagram showing sprint 1.

# **Technical Manual**

## 7.3.2. Sprint 2

# Technical Manual

<b>Test Plan:</b>	Test Plan Sprint 2			
<b>Project Name:</b>	Behavioural Password ProtectionTyping Habit Gesture Authentication			
<b>Test Case ID:</b>	TC-3	<b>Test Case Description:</b>	As a user, they should be able to register account so that they can log in to the system.	
<b>Created By:</b>	Cheo Cui Hui	<b>Reviewed By:</b>	Terrence Yap	Version 2.0
<b>Tester's Name:</b>	Alicia Tan	<b>Date Tested:</b>	19/5/2021	<b>Test Case (Pass/Fail)</b>
				Pass
<b>No.</b>	<b>Prerequisites</b>		<b>Test Scenario No.</b>	<b>Test Data</b>
1	Access to Chrome Browser		1	Username: tcyan First name: Alicia Last name: Tan
2	User clicks on the "Register" option on the Welcome page.		2	Username: alicy (exists in database)
3			3	Username: tcyal
4			4	Username: tcyal
			5	Username: tcyal
			6	
			7	
			8	
			9	Username: tcyal First name: Alicia Last name: Tan Date of birth: 21/02/1999
<b>Test Scenario 1:</b> Verifying Empty inputs. User cannot proceed to register their typing habit.				
<b>Test Step</b>	<b>Test Step Description</b>	<b>Expected Results</b>	<b>Actual Results</b>	<b>Pass/Fail</b> <b>Remarks</b>
TC-3-1	Navigate to Register website	Site should open	Site loaded and displayed	Pass
		User is navigated to		
TC-3-2	Click "Register"	Registration site	Site loaded and displayed	Pass
		Unable to proceed to registering		
TC-3-3	Click "Next"	typing habit. Prompted to fill up details	System prompt user to enter	Pass Error message "Please fill out this fill" will display on the browser.
<b>Test Scenario 2:</b> Verifying Missing inputs. User cannot proceed to register their typing habit.				
<b>Test Step</b>	<b>Test Step Description</b>	<b>Expected Results</b>	<b>Actual Results</b>	<b>Pass/Fail</b> <b>Remarks</b>
TC-3-4	Navigate to Register website	Site should open	Site loaded and displayed	Pass
		User is navigated to		
TC-3-5	Click "Register"	Registration site	Site loaded and displayed	Pass
		Tested submitting with one empty input field each time (repeated 6x because 6 input fields)		
TC-3-6	User fills out the register form with missing inputs	Credential can be entered	Able to enter	Pass
		Unable to proceed to registering		
TC-3-7	Click "Next"	typing habit. Prompted to fill up details	User is not registered	Pass Error message "Please fill out this fill" will display on the browser.
<b>Test Scenario 3:</b> Verifying Existing Username. User cannot proceed to register their typing habit.				
<b>Test Step</b>	<b>Test Step Description</b>	<b>Expected Results</b>	<b>Actual Results</b>	<b>Pass/Fail</b> <b>Remarks</b>
TC-3-8	Navigate to Register website	Site should open	Site loaded and displayed	Pass
		User is navigated to		
TC-3-9	Click "Register"	Registration site	Site loaded and displayed	Pass
		Tested submitting with one empty input field each time (repeated 6x because 6 input fields)		
TC-3-10	User fills out the register form with an existing username	Credential can be entered	Able to enter	Pass
		Unable to proceed to registering		
TC-3-11	Click "Next"	typing habit. Error message appears.	User is not registered	Pass Error message "Username is taken" will display on the browser.
<b>Test Scenario 4:</b> Verifying Invalid Email. User cannot proceed to register their typing habit.				
<b>Test Step</b>	<b>Test Step Description</b>	<b>Expected Results</b>	<b>Actual Results</b>	<b>Pass/Fail</b> <b>Remarks</b>
TC-3-12	Navigate to Register website	Site should open	Site loaded and displayed	Pass
		User is navigated to		
TC-3-13	Click "Register"	Registration site	Site loaded and displayed	Pass
		Tested submitting with one invalid email.		
TC-3-14	User fills out the register form with an invalid email.	Credential can be entered	Able to enter	Pass
		Unable to proceed to registering		
TC-3-15	Click "Next"	typing habit. Error message appears.	User is not registered	Pass Error message "Invalid Email" will display on the browser.
<b>Test Scenario 5:</b> Verifying Existing Email. User cannot proceed to register their typing habit.				
<b>Test Step</b>	<b>Test Step Description</b>	<b>Expected Results</b>	<b>Actual Results</b>	<b>Pass/Fail</b> <b>Remarks</b>
TC-3-16	Navigate to Register website	Site should open	Site loaded and displayed	Pass
		User is navigated to		
TC-3-17	Click "Register"	Registration site	Site loaded and displayed	Pass
		Tested submitting with one existing email.		
TC-3-18	User fills out the register form with an existing email.	Credential can be entered	Able to enter	Pass
		Unable to proceed to registering		
TC-3-19	Click "Next"	typing habit. Error message appears.	User is not registered	Pass Error message "Email is taken" will display on the browser.
<b>Test Scenario 6:</b> Verifying Valid Email. User proceeds to register their typing habit.				
<b>Test Step</b>	<b>Test Step Description</b>	<b>Expected Results</b>	<b>Actual Results</b>	<b>Pass/Fail</b> <b>Remarks</b>
TC-3-20	Navigate to Register website	Site should open	Site loaded and displayed	Pass
		User is navigated to		
TC-3-21	Click "Register"	Registration site	Site loaded and displayed	Pass
		Tested submitting with one valid email.		
TC-3-22	User fills out the register form	Credential can be entered	Able to enter	Pass
		User is navigated to Register		
TC-3-23	Click "Next"	Typing Habit site	Able to proceed to registering typing habit.	Pass
<b>Test Scenario 7:</b> Verifying Valid Registration details and Empty Registering typing habit.				
<b>Test Step</b>	<b>Test Step Description</b>	<b>Expected Results</b>	<b>Actual Results</b>	<b>Pass/Fail</b> <b>Remarks</b>
TC-3-24	Navigate to Register website	Site should open	Site loaded and displayed	Pass
		User is navigated to		
TC-3-25	Click "Register"	Registration site	Site loaded and displayed	Pass
		Tested submitting with one valid registration details.		
TC-3-26	User fills out the register form	Credential can be entered	Able to enter	Pass
		User is navigated to Register		
TC-3-27	Click "Next"	Typing Habit site	Able to proceed to registering typing habit.	Pass
		Tested submitting with one empty sentence.		
TC-3-28	Click "Next"	User is not registered	Unable to proceed with registering typing habit. Error message appears.	Pass Error message "Please type in the box below the generated sentence." will display on the browser.
<b>Test Scenario 8:</b> Verifying Valid Registration details and Invalid Registering typing habit.				
<b>Test Step</b>	<b>Test Step Description</b>	<b>Expected Results</b>	<b>Actual Results</b>	<b>Pass/Fail</b> <b>Remarks</b>
TC-3-29	Navigate to Register website	Site should open	Site loaded and displayed	Pass
		User is navigated to		
TC-3-30	Click "Register"	Registration site	Site loaded and displayed	Pass
		Tested submitting with one invalid registration details.		
TC-3-31	User fills out the register form	Credential can be entered	Able to enter	Pass
		User is navigated to Register		
TC-3-32	Click "Next"	Typing Habit site	Able to proceed onto part 2 of registration.	Pass
		Tested submitting with one invalid sentence.		
TC-3-33	Enter sentence generated with errors	Credential can be entered	Able to enter	Pass
		User is navigated to Register		
TC-3-34	Click "Next"	User is registered	Unable to proceed with registering typing habit. Error message appears.	Pass Error message "Sentence entered does not match generated sentence. Please try again." will display on the browser.
<b>Test Scenario 9:</b> Verifying Valid Registration details and Successful Registering typing habit. User is registered.				
<b>Test Step</b>	<b>Test Step Description</b>	<b>Expected Results</b>	<b>Actual Results</b>	<b>Pass/Fail</b> <b>Remarks</b>
TC-3-35	Navigate to Register website	Site should open	Site loaded and displayed	Pass
		User is navigated to		
TC-3-36	Click "Register"	Registration site	Site loaded and displayed	Pass
		Tested submitting with one successful registration details.		
TC-3-37	User fills out the register form	Credential can be entered	Able to enter	Pass
		User is navigated to Register		
TC-3-38	Click "Next"	Typing Habit site	Able to proceed onto part 2 of registration.	Pass
		Tested submitting with one successful sentence.		
TC-3-39	Enter sentence generated	Credential can be entered	Able to enter	Pass
		User is navigated to Register		
TC-3-40	Click "Next"	User is registered	Able to click next and proceed on with registration	Pass This repeats for 10 cycles
TC-3-41	Click "Register Account"	Typing Habit site	Site loaded and displayed	Pass Login page is displayed.

# Technical Manual

Figure 57. Diagram showing sprint 2.

### 7.3.3. Sprint 3

<b>Test Plan:</b>	Test Plan Sprint 3				
<b>Project Name:</b>	Behavioural Password				
<b>Test Case ID:</b>	TC-1	<b>Test Case Description:</b>	A user, they should be able to get authenticated so		
<b>Created By:</b>	Chea Cui Hui	<b>Reviewed By:</b>	Terrence Yap	<b>Version:</b>	2.0
<b>Tester's Name:</b>	Alicia Tan	<b>Date tested:</b>	19/5/2021	<b>Test Case (Pass/Fail)</b>	Pass
<b>No.</b>	<b>Prerequisites</b>			<b>Test Scenario No.</b>	<b>Test</b>
1	Access to Chrome Browser			1	-
2				2	Invalid Username
3				3	- Invalid Password
4				4	
				5	
				6	
<b>Test Scenario 1:</b>	cannot proceed to login with their				
<b>Test Step</b>	<b>Test Step Description</b>	<b>Expected Results</b>	<b>Actual Results</b>	<b>Pass/Fail</b>	<b>Remarks</b>
TC-1-1	Navigate to website URL	Site should open	Site loaded and displayed	Pass	
TC-1-2	Click "Log in"	System prompt user to enter Username	Able to type username	Pass	Error message "No input detected" will display on the browser.
<b>Test Scenario 2:</b>	cannot proceed to login with their				
<b>Test Step</b>	<b>Test Step Description</b>	<b>Expected Results</b>	<b>Actual Results</b>	<b>Pass/Fail</b>	<b>Remarks</b>
TC-1-3	Navigate to website URL	Site should open	Site loaded and displayed	Pass	
TC-1-4	Enter incorrect Username	Credential can be entered	Able to type username	Pass	
TC-1-5	Click "Log in"	User is not logged in	Error message loaded. Could not proceed onto login verification part 2	Pass	Error message "User not detected, please register an account." will display on the browser.
<b>Test Scenario 3:</b>	proceeds to exit Typing Habit				
<b>Test Step</b>	<b>Test Step Description</b>	<b>Expected Results</b>	<b>Actual Results</b>	<b>Pass/Fail</b>	<b>Remarks</b>
TC-1-6	Navigate to website URL	Site should open	Site loaded and displayed	Pass	
TC-1-7	Enter Username (existing in database)	Credential can be entered	Able to enter username	Pass	
TC-1-8	Click "Log in"	Verification site	onto validating typing habit	Pass	
TC-1-9	Click the Typing Habit Gesture Authentication System Logo on the top left.	User is navigated to Login page	Empty login page (as seen in TC-1-6) displayed	Pass	Username entered previously is discarded.
<b>Test Scenario 4:</b>	Typing Habit. User cannot login.				
<b>Test Step</b>	<b>Test Step Description</b>	<b>Expected Results</b>	<b>Actual Results</b>	<b>Pass/Fail</b>	<b>Remarks</b>
TC-1-10	Navigate to website URL	Site should open	Site loaded and displayed	Pass	
TC-1-11	Enter Username	Credential can be entered	Able to enter username	Pass	
TC-1-12	Click "Log in"	Verification site	onto validating typing habit	Pass	
TC-1-13	Click "Validate Habit"	User is not logged in	Unable to proceed to homepage.	Pass	Error message "Please input something" will display on the browser.
<b>Test Scenario 5:</b>	Typing Habit. User cannot login.				
<b>Test Step</b>	<b>Test Step Description</b>	<b>Expected Results</b>	<b>Actual Results</b>	<b>Pass/Fail</b>	<b>Remarks</b>
TC-1-14	Navigate to website URL	Site should open	Site loaded and displayed	Pass	
TC-1-15	Enter Username	Credential can be entered	Able to enter username	Pass	
TC-1-16	Click "Log in"	User is navigated to Typing Habit Verification site	Able to validate username and proceed onto validating typing habit	Pass	
TC-1-17	Enter sentence generated using incorrect typing habit	Credential can be entered	Able to submit typing habit	Pass	
TC-1-18	Click "Validate Habit"	User is not logged in	Unable to log in due to incorrect typing habit detected	Pass	Error message "2 out of 3 attempt(s) left." will display on the browser.
<b>Test Scenario 6:</b>	Habit. User can login.				
<b>Test Step</b>	<b>Test Step Description</b>	<b>Expected Results</b>	<b>Actual Results</b>	<b>Pass/Fail</b>	<b>Remarks</b>
TC-1-19	Navigate to website URL	Site should open	Site loaded and displayed	Pass	
TC-1-20	Enter Username	Credential can be entered	Able to enter username	Pass	
TC-1-21	Click "Log in"	User is navigated to Typing Habit Verification site	Able to validate username and proceed onto validating typing habit	Pass	
TC-1-22	Enter sentence generated using correct typing habit	Credential can be entered	Able to submit typing habit	Pass	
TC-1-23	Click "Validate Habit"	User is logged in	Able to log in	Pass	Homepage is displayed with the Username.

Figure 58a. Diagram showing sprint 3

# Technical Manual

Test Plan:	Test Plan Sprint 3				
Project Name:	Behavioural Password Protection	Typing Habit Gesture Authentication			
Test Case ID:	TC-5	Test Case Description:	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.		
Created By:	Cheah Cui Hui	Reviewed By:	Terrence Yap	Version	2.0
Tester's Name:	Alicia Tan	Date Tested:	19/5/2021	Test Case (Pass/Fail)	Pass
No.	Prerequisites			Test Scenario no.	Test Data
1	Access to Chrome Browser			1	username: alityc
2	Login to Typing Habit Gesture Authentication System			2	
3				3	
4				4	
<b>Test Scenario 1:</b> Verifying Empty inputs. User cannot proceed to recalibrate their typing habit.					
Test Step	Test Step Description	Expected Results	Actual Results	Pass/Fail	Remarks
TC-5-1	Navigate onto Recalibration page	Site should open	Site loaded and displayed	Pass	
TC-5-2	Click "Next"	User unable to proceed with recalibration	Unable to proceed with recalibration. Error message shown.	Pass	Error message "Please type in the box below the generated sentence." will display on the browser.
<b>Test Scenario 2:</b> Verifying Invalid inputs. User cannot proceed to register their typing habit.					
Test Step	Test Step Description	Expected Results	Actual Results	Pass/Fail	Remarks
TC-5-3	Navigate to Recalibration page	Site should open	Site loaded and displayed	Pass	
TC-5-4	Enter sentence generated with errors	Credential can be entered	Able to type	Pass	
TC-5-5	Click "Next"	User is registered	Unable to proceed with recalibration. Error message shown.	Pass	Error message "Sentence entered does not match generated sentence. Please try again." will display on the browser.
<b>Test Scenario 3:</b> Verifying Valid inputs 10 times. User does not want it to be recalibrated					
Test Step	Test Step Description	Expected Results	Actual Results	Pass/Fail	Remarks
TC-5-6	Navigate to Recalibration page	Site should open	Site loaded and displayed	Pass	
TC-5-7	Enter sentence generated	Credential can be entered	Able to type	Pass	
TC-5-8	Click "Next"	User is registered	Able to register	Pass	This repeats for 10 cycles
TC-5-9	On the 10th time, click on the logo at the top left	User is navigated to their Homepage	User habit is not recalibrated with new sentence	Pass	
<b>Test Scenario 4:</b> Verifying Valid Registration details and Successful Registering typing habit. User is registered.					
Test Step	Test Step Description	Expected Results	Actual Results	Pass/Fail	Remarks
TC-5-10	Navigate to Recalibration website	Site should open	Site loaded and displayed	Pass	
TC-5-11	Enter sentence generated	Credential can be entered	Able to type	Pass	
TC-5-12	Click "Next"	User is registered	Able to continue recalibration for 10 times	Pass	This repeats for 10 cycles
TC-5-13	On the 10th time, click the button to recalibrate	User successfully registers habit and is able to navigate to homepage.	Successful recalibration. Able to move onto homepage.	Pass	Recalibration page is displayed with the "Successfully logged out" message.
TC-5-14	Log out	Successfully logged out	Successfully logged out	Pass	Recalibration page is displayed with the "Successfully logged out" message.
TC-5-15	Navigate to Login page	User successfully registers habit and is able to navigate to homepage.	Login sentence changed to the new one generated during recalibration	Pass	Recalibration page is displayed with the "Successfully logged out" message.
TC-5-16	Enter username	Able to enter credentials	Able to enter	Pass	
TC-5-17	Click "Login"	User able to validate username and move onto the typing habit validation login page. The typing habit validation page shows a new sentence that user typed during recalibration	Able to validate username and move onto the typing habit validation login page. The typing habit validation page showed a new sentence that was typed during recalibration	Pass	Recalibration page is displayed with the "Successfully logged out" message.

Figure 58b. Diagram showing sprint 3.

# Technical Manual

## 7.3.4. Sprint 4

Test Plan:	Test Plan Sprint 4	Project Name:	Behavioural Password ProtectionTyping Habit Gesture Authentication System				
Test Case ID:	TC-4	Created By:	Cheo Cui Hui	Test Case Description:	Reviewed By:	As a user, they should be able to authenticate themselves using a security answer in case their typing habits change.	
					Terrence Yap	Version	2.0
Tester's Name:	Alicia Tan	Date Tested:	19/5/2021	Test Case (Pass/Fail)		Pass	
No.	Prerequisites			Test Scenario No.	Test Data		
1	Access to Chrome Browser			1	username: alitcy		
2	User fails the Typing Habit Authentication 3 time and is redirected to the Security Question Login page.			2	username: alitcy	answer: fake-answer	
3				3	username: alitcy		
4							
<b>Test Scenario 1:</b> Verifying <b>Empty</b> answer. User cannot proceed to login with their typing habit.							
Test Step	Test Step Description	Expected Results	Actual Results	Pass/Fail	Remarks		
TC-4-1	Navigate to Security Question website	Site should open	Form loaded and displayed	Pass			
TC-4-2	Click "Log in"	System prompt user to enter answer	Unable to login. Error message shown.	Pass	Error message "Please enter something" will display on the browser.		
<b>Test Scenario 2:</b> Verifying <b>Invalid</b> answer. User cannot proceed to login with their typing habit.							
Test Step	Test Step Description	Expected Results	Actual Results	Pass/Fail	Remarks		
TC-4-3	Navigate to Security Question website	Site should open	Form loaded and displayed	Pass			
TC-4-4	Enter incorrect answer	Credential can be entered	Able to enter	Pass			
TC-4-5	Click "Log in"	User is not logged in	Unable to login. Error message shown.	Pass	Error message "Security answer is incorrect." will display on the browser.		
<b>Test Scenario 3:</b> Verifying <b>Valid</b> username. User can login.							
Test Step	Test Step Description	Expected Results	Actual Results	Pass/Fail	Remarks		
TC-4-6	Navigate to Security Question website	Site should open	Form loaded and displayed	Pass			
TC-4-7	Enter correct answer	Credential can be entered	Able to enter	Pass			
TC-4-8	Click "Log in"	Verification site	Able to proceed with website displaying homepage	Pass	Homepage is displayed with the Username.		

Figure 59. Diagram showing sprint 4.

## 7.4 Machine Learning Development

### 7.4.1 Self-Organizing Map

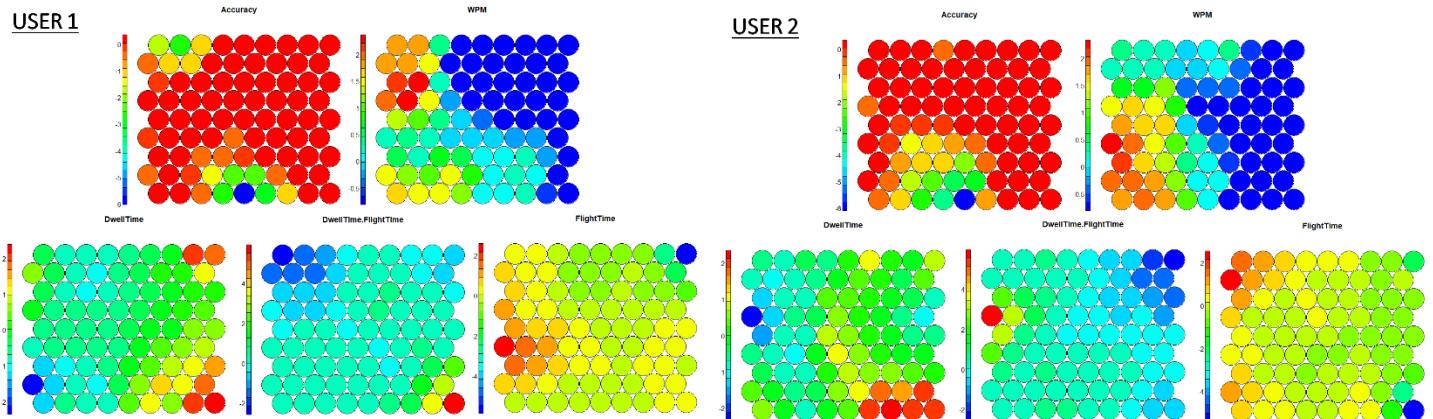
Self-Organizing Map (SOM), or Kohonen Map, is an unsupervised machine learning method that is used to perform a topology preserving feature mapping. SOM is used in this project to determine features to use to help us develop a more accurate model to authenticate users.

Thus, from the Behaviour Analysis we did before, we choose to use features such as Accuracy, Words per Minute (WPM), Dwell Time, Flight Time and a derived value, the difference of Dwell and Flight Time, as part of our analysis to use for our system.

Name	Accuracy	WPM	DwellTime	DwellTime,FlightTime	FlightTime	TrueFlase
1	98.85057	97.40444	0.182	-0.047	0.229	0
1	100	117.473	0.112	-0.036	0.148	0
1	100	105.7485	0.124	-0.098	0.222	0
1	100	114.7942	0.119	-0.036	0.155	0
1	100	103.3654	0.169	-0.127	0.296	0
1	97.72727	102.3911	0.122	-0.045	0.167	0
1	97.72727	100.7714	0.15	0.012	0.138	0
1	94.50549	96.61112	0.11	-0.012	0.122	0
1	95.55556	98.71819	0.108	-0.014	0.122	0
1	100	123.3564	0.112	-0.004	0.116	0
2	100	90.67744	0.152	0.075	0.077	0
2	100	88.57609	0.128	0.12	0.008	0
2	100	81.52947	0.122	0.128	-0.006	0
2	100	89.08071	0.198	0.464	-0.266	0
2	100	81.81386	0.142	0.127	0.015	0
2	100	92.53945	0.129	0.122	0.007	0
2	100	92.80576	0.11	0.129	-0.019	0
2	100	92.37379	0.191	0.054	0.137	0
2	100	86.16515	0.147	0.129	0.018	0
2	100	87.42058	0.132	0.11	0.022	0
3	100	40.43095	0.234	0.21	0.024	0
3	100	36.31757	0.264	0.096	0.168	0
3	100	37.21735	0.278	0.092	0.186	0
3	100	36.93497	0.238	1.204	-0.966	0
3	100	42.75948	0.256	0.083	0.173	0
3	100	41.16802	0.234	0.082	0.152	0
3	100	44.38901	0.227	0.097	0.13	0
3	100	41.65321	0.281	0.629	-0.348	0
3	100	45.42853	0.283	0.164	0.119	0
3	100	50.86249	0.256	0.091	0.165	0

Figure 60. Typing Biometric Sample.

After getting samples of typing biometrics, we used them to do the subsequent SOM Heatmaps for each feature, between different users.



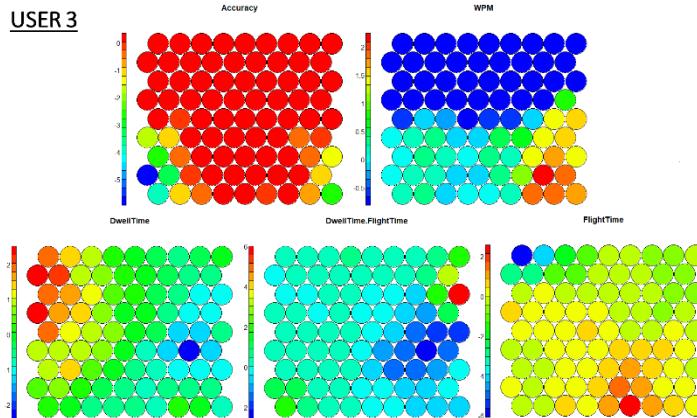


Figure 61. SOM Heat map for Sample User 1,2 & 3.

For our project, we are required to differentiate between each user based on their typing biometric and thus we need to be able to differentiate between each small difference that the user makes. Therefore, from these models, we can see that Accuracy and WPM have many neurons that has the similar color within each model and thus it is not sensitive enough to achieve what we need in this project. As for Dwell Time, Flight Time and Dwell- Flight Time, we can see more variations and thus better to understand and differentiate between users.

#### 7.4.2 Logistic regression

10 users who typed the given sentence 10 times each were used to form a dataset of 100 rows. This dataset was used for logistic regression. Of the 10 users, 1 user was selected and labelled “Genuine” and the rest labelled as “Imposter”. The database was split into train set and test set of 80% and 20% respectively. The results of the test set prediction are shown as a confusion matrix in figure 73. The model has 100% accuracy, predicting the 2 “Genuine” and 18 “Imposter” correctly.

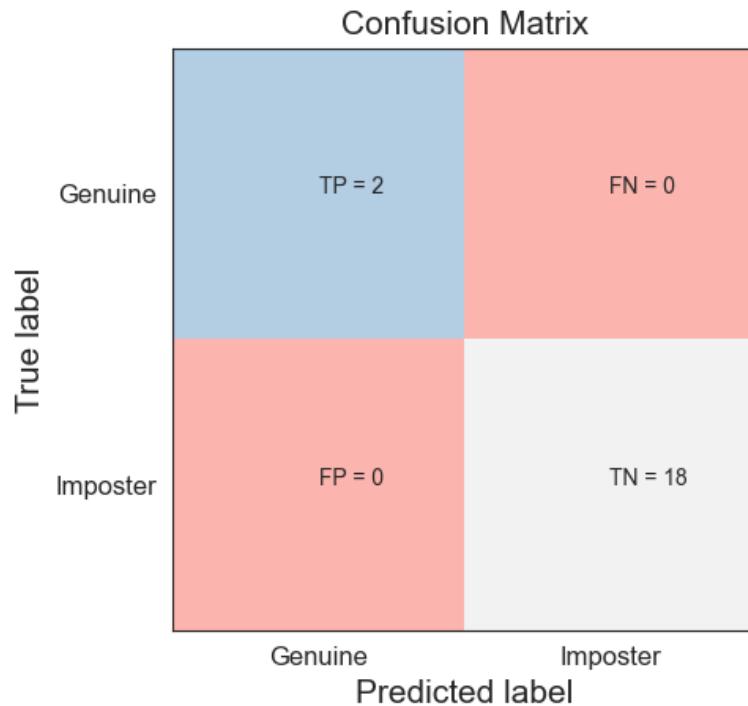


Figure 62. Confusion matrix for logistic regression.

## 7.4.3 One Class SVM (OCSVM)

The same dataset used for logistic regression is also used for OC SVM, except with the last column containing the labels removed. One user's typing habit was used to train the OC SVM to produce a model specific to that user. Then, all other 90 rows, excluding the user whose data was used to train the model, were tested against the model. This is shown as "Accuracy compared to other users" in figure 74. The user whose data was used to train the model was also had another set of 10 rows tested against the model previously created. Note that this new set is not the same set used to train the model. This is shown as "Accuracy of same user" in figure 74.

```
Accuracy of same user: 0.7  
Accuracy compared to other users: 1.0
```

Figure 63. Accuracy of the linear regression model.

## 7.4.4 Choosing OCSVM over logistic regression

- OCSVM seeks to locate the “best” margin (distance between the line and the support vectors) that separates the classes, reducing the possibility of data error, but logistic regression does not, instead having many decision boundaries with various weights that are close to the optimal point (Bassey, 2019).
- OCSVM works effectively with unstructured and semi-structured data like as text and images, whereas logistic regression works with independent variables that have already been identified (Bassey, 2019). Typing habit data is very dynamic and varies from user to user. This leads to semi-structured data because while it may be similar for one user, it may be very different from another user.
- Logical regression is based on statistical techniques, whereas OCSVM is based on geometrical properties of the data.
- OCSVM is able to maximize the distance of the hyperplane from the origin by separating all data points from the origin. This reduces the risk of overfitting for OCSVM. Logistic regression, however, is more prone to overfitting than OCSVM.
- On top of these, OCSVM allowed training with a single person's data. This is an advantage as the model only required single person's data to generate a model as compared to regression that require more than 1 person's data to learn the typing habits.
- Furthermore, linear regression required the outcome to be known. Additional data processing and combining would be required to prepare the data in order for the model to use it.

## Chapter 8: Post-development Data Collection and Analysis

The importance of post-development data collection and analysis is that it allows us to learn more about how our website application has impacted users. Since this project was completed during COVID-19, qualitative data would be collected by sending questionnaires to friends and students via an online platform rather than on the SIM campus.

### 8.1 Contents of survey

#### Section A - User demographics

Question A1: "How old are you?"

Question A2: "What is your gender?"

Question A3: "What is your occupation?"

Question A4: "Do you have web development experience?"

Question A5: "Number of hours per week spent on the computer?"

These questions aim to understand the characteristics of the surveyors and how adept they are at using the computer. Also, through question A3, we aim to differentiate web developers and software developers from the rest of the surveyors as they are our target audience.

#### Section B(i, ii, iii, iv, v, vi, vii, viii, ix) - Testing application section

##### *Section B(i) - Registration*

Question Bi1: "Are you redirected to Register page?"

Question Bi2: "Are you able to proceed to Step 2: Register your typing habit?"

Question Bi3: "Are you able to see that the number of attempts, located above the "Next" button, change from 1/10 attempt(s) to 10/10 attempt(s) accordingly for the next 10 attempt(s)?"

Question Bi4: "After 10 times, the system is able to register you as our user and redirects you to the Login page. Are you able to complete this action successfully?"

##### *Section B(ii) - Login*

Question Bii1: "Are you able to proceed to Step 2: Validate your typing habit?"

Question Bii2: "After verifying habit, it will load the homepage with the username. Are you able to complete this action successfully?"

##### *Section B(iii) - Logout*

Question Biii1: "After you have successfully logout "Successfully logged out" is displayed, are you able to complete this action successfully?"

# Technical Manual

## *Section B(iv) - Return to Login page*

Question Biv1: "Are you redirected to Login page?"

## *Section B(v) - Security Question Login*

Question Bv1: "Are you able to proceed to Step 2: Validate your typing habit?"

Question Bv1: "After you have fail Typing Habit Authentication 3 time, are you redirected to Security Question Login page?"

Question Bv1: "After verifying Security Answer, it will load the homepage with the username. Are you able to complete this action successfully?"

## *Section B(vi) - Recalibration*

Question Bvi1: "Are you redirected to the Re-calibration page?"

Question Bvi2: "Are you able to see that the number of attempts, located above the "Next" button, change from 1/10 attempt(s) to 10/10 attempt(s) accordingly for the next 10 attempt(s)?"

Question Bvi3: "The Recalibration page will displayed with the "Success!!" message. Are you able to see the display message successfully?"

## *Section B(vii) - Logout*

Question Bvii1: "Are you redirected to Homepage?"

Question Bvii2: "Are you redirected to Login page?"

## *Section B(viii) - Test Recalibration*

Question Bviii1: "After verifying habit, it will load the homepage with the username. Are you able to complete this action successfully?"

## *Section B(ix) - Logout*

Question Bix1: "Are you redirected to Login page?"

## *At the end of section B – conclusion*

Question: "Any improvements to be made to improve user experience?"

The questions in section B aim to get users of different demographics to test our website and gather feedback regarding the functionality of the website and their user experience toward the website.

**Section C - User knowledge on typing habit authentication**

# **Technical Manual**

Question C1: "Are you aware of any similar products (on user authentication using typing biometrics) in the market?"

Question C2: "Have you used this type of authentication system before?"

The questions in this section aim to understand if our surveyors have knowledge or concept of using typing biometrics as an authentication system.

## **Section D – Feedback**

Question D1: "Do you feel typing the sentence 10 times (during registration and recalibration of typing habit) is too many times?"

Question D2: "Do you have trouble logging in into the Typing Habit Gesture System?"

Question D3: "Prior to using the Typing Habit Gesture System, do you prefer to use password or a sentence to login to a system?"

Question D4: "After using the Typing Habit Gesture System, would you be open to using this system as a method to log into a system?"

Question D5: "On a scale of 1-10 (1 being least likely and 10 being most likely), would this be your preferred way of logging in?"

Question D6: "Is there anything you do not like about the application?"

Question D7: "(ONLY FOR WEB DEVELOPERS) Will you implement the Typing Habit Gesture System inside your application?"

The questions in this section aim to conclude the potential marketability of our product and gather overall user feedback regarding their experience with our product as well as the concept of authenticating themselves using typing biometrics. Question D7 is aimed at our target audience, web developers, if they would be interested in implementing this as an add on inside their website application in the future.

## 8.2 Survey Results

### 8.2.1. Errors & Issues

#### Common Errors

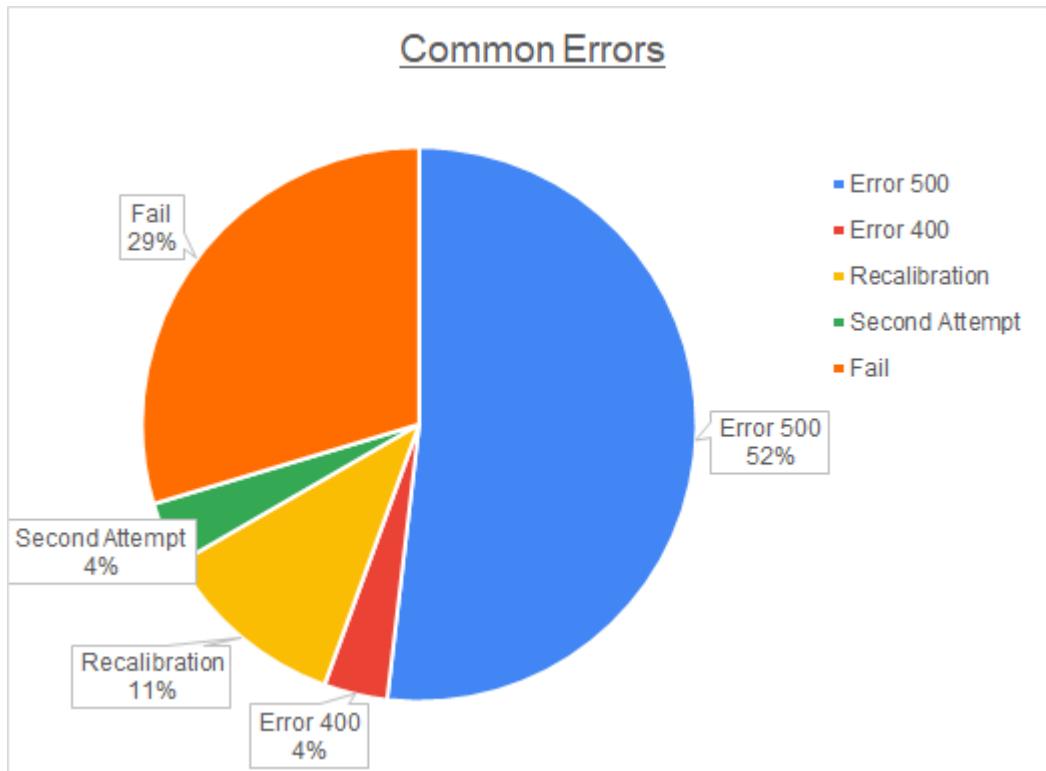


Figure 64. Survey results for common error.

This chart shows the common error that the user experience.

#### Issues Faced

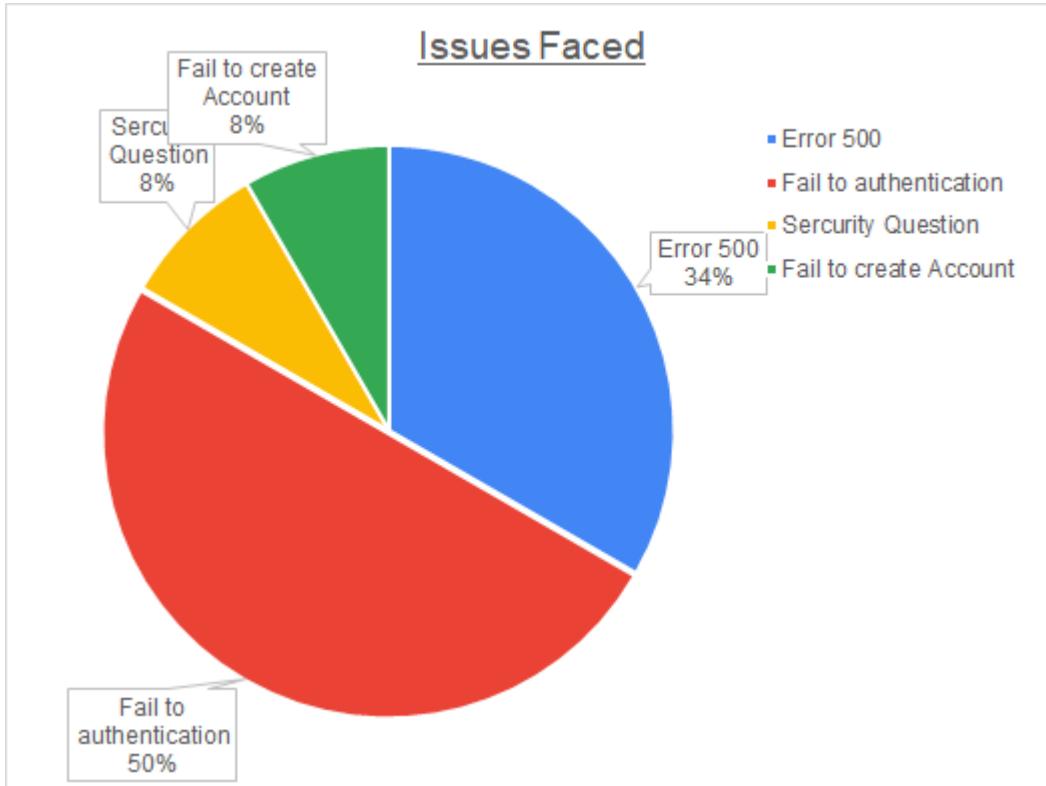


Figure 65. Survey results for the Issues that user faced.

This chart shows the issues that the user experience.

### 8.2.2. Typing Habit Cycle

#### Typing Habit Registration Cycle

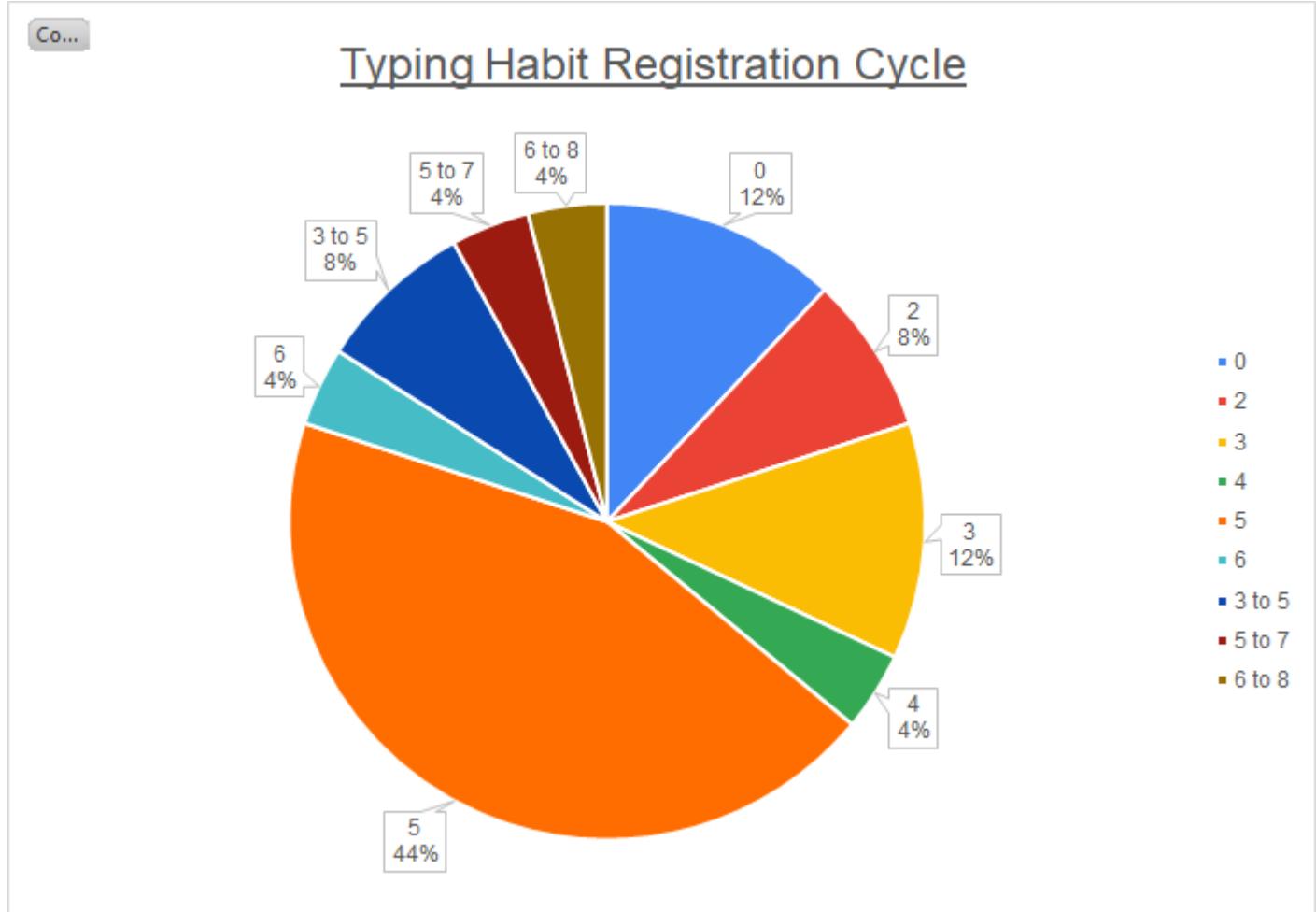


Figure 66. Survey results for the user typing habit registration cycle.

This chart shows the number of cycles the users think is acceptable to them.

## 8.2.3. Before and After Login

### Before vs After using AuthenticateMe

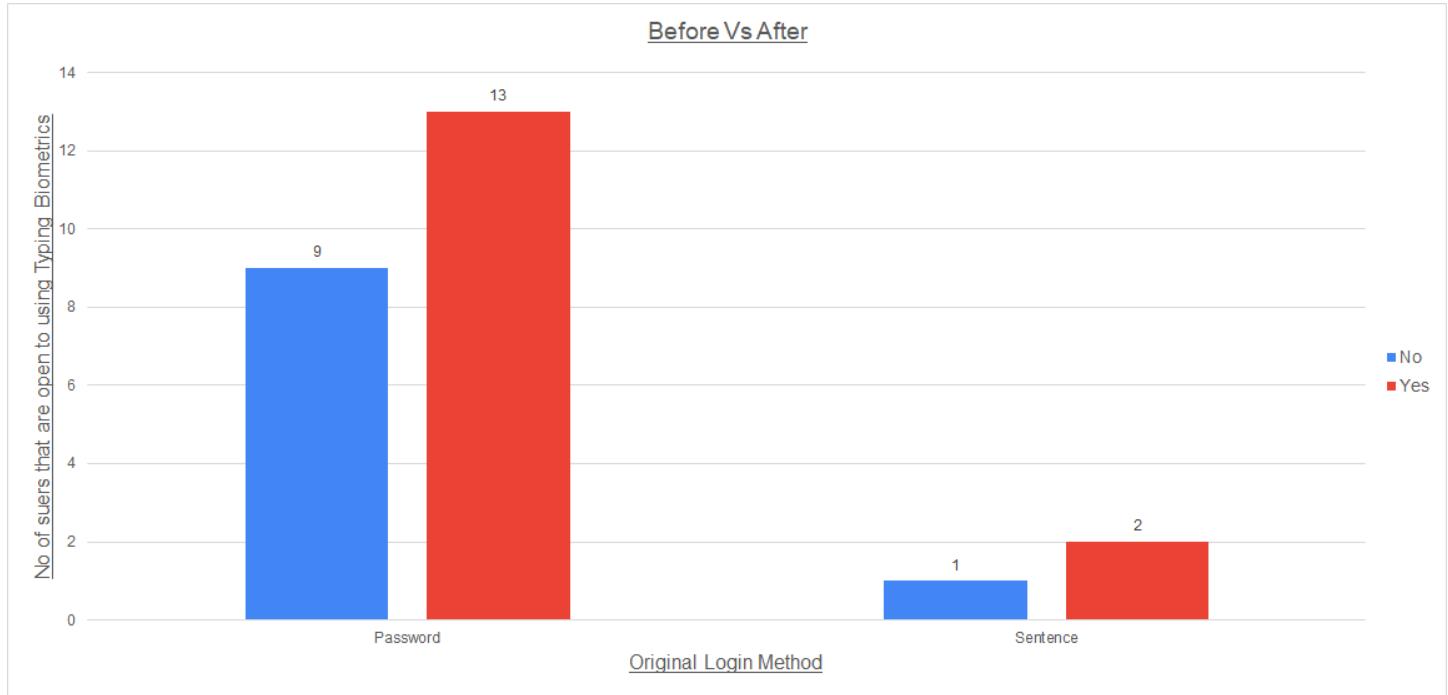


Figure 67: Survey results for the before login method and after for using typing biometrics.

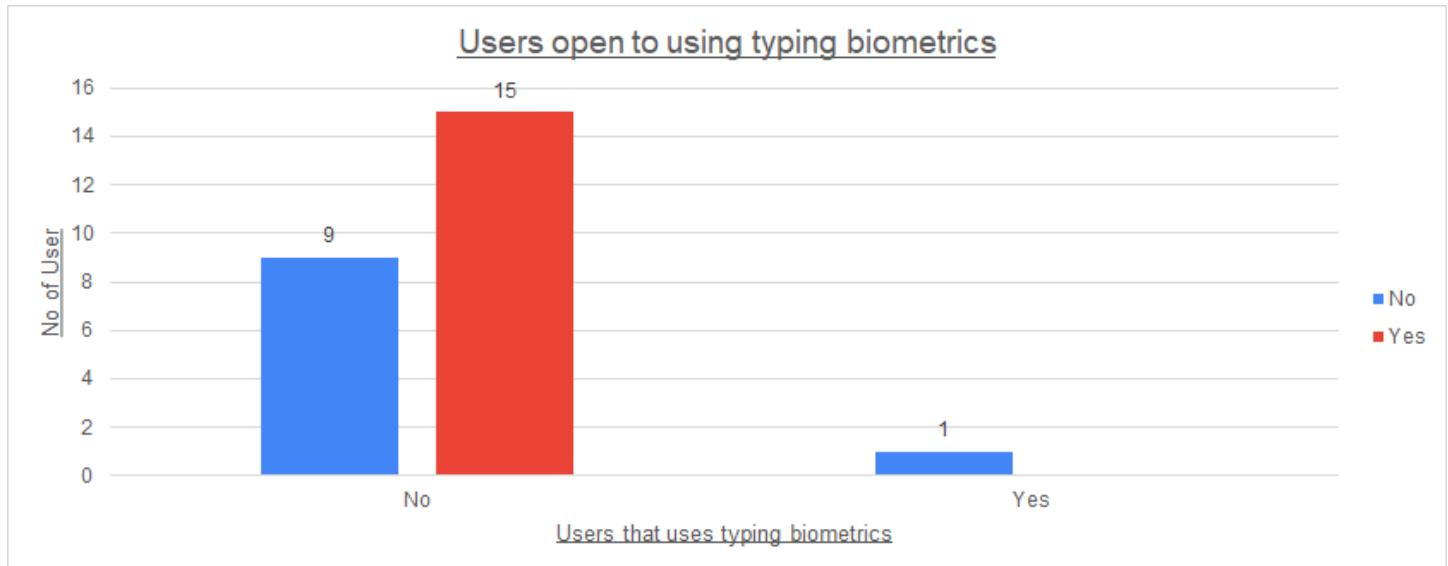


Figure 68: Survey results showing users that have used typing biometric authentication. Users are further differentiated to whether they are open to using typing biometrics as an authentication system.

This chart shows the number of users that are open to using typing biometrics despite their original login method of using password or sentences.

## Occupation + Hour Per Week

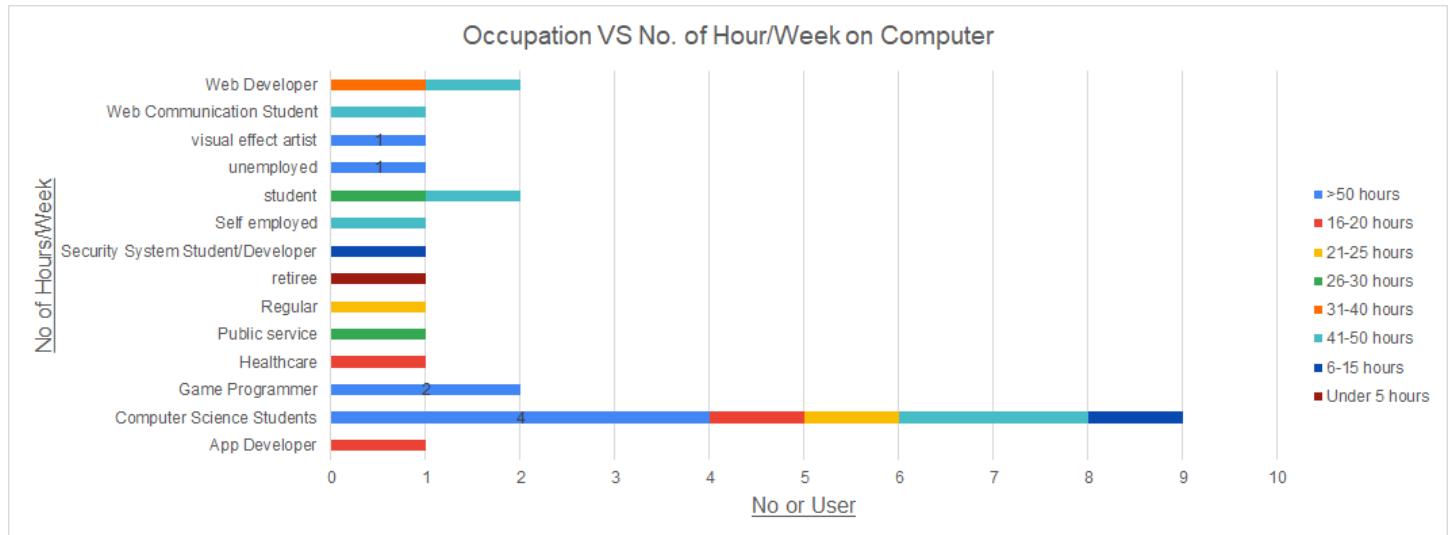


Figure 69: Survey results showing the distribution of hours and users spend on the computer, as well as the number of users in that category.

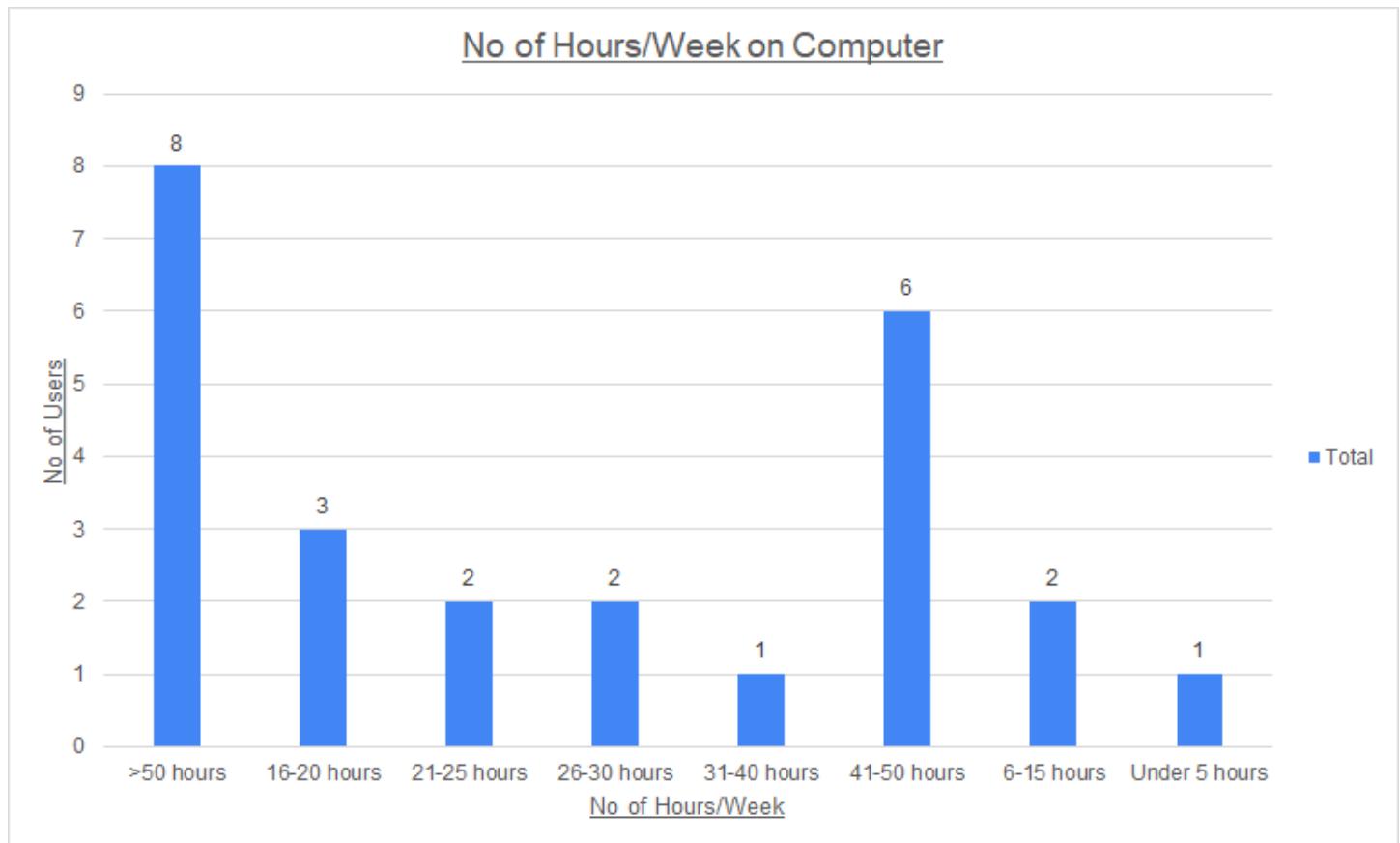


Figure 70. Survey results for Occupation + Hour Per Week.

This chart shows the number of the users from different occupation and the number of hours spent on computer per week.

From the results shown in figures 69 and 70, we have a good range of survey respondents who are familiar and not so familiar with spending time on the computer. This gives us a relatively good range to test our system on the public.

## Ratings/Preference on logging in via typing habits

(1 – least preferred; 10 – most preferred)

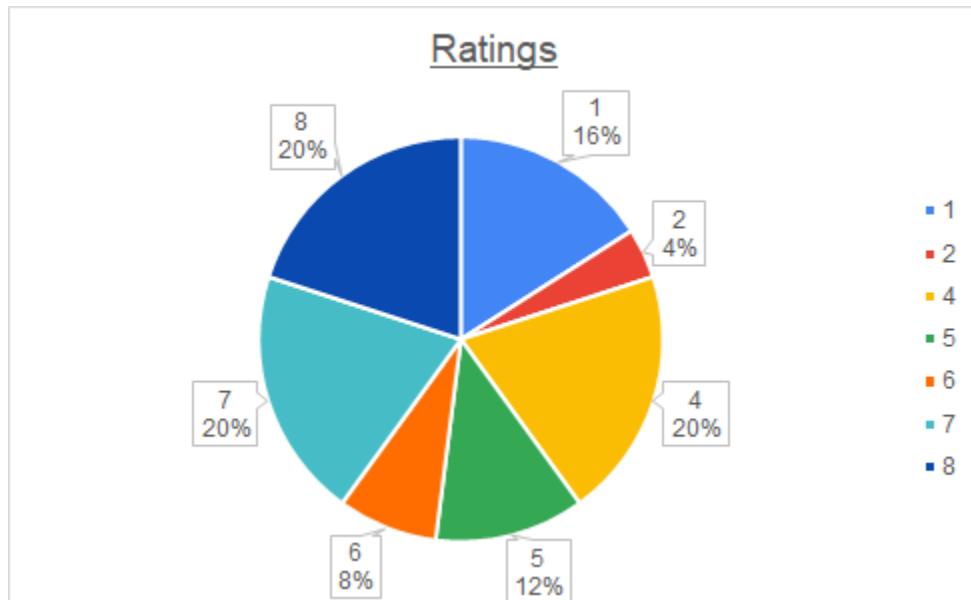


Figure 71. Survey results for the ratings.

This chart shows the ratings of our system by the user.

According to figure 71, it shows that users that prefer authenticating themselves via typing habits are 48%, while users that do not prefer authenticating themselves via typing habits (ratings lower or equal to 5) is 52%. This may be because users are not familiar with this type of login. As seen in figure 68, the number of people that do not use typing habits as an authentication system are 24 out of 25 survey respondents. Thus, the users that do not prefer this authentication method may be due to a lack of awareness in the public about this type of authentication system.

# Technical Manual

## Chapter 9: Marketing Website

This section features our marketing website and the sections that it contains.

The link for the marketing website is: <https://fyp21s102.wixsite.com/fyp21s102>

### 9.1 Home Page

This is our home page for our marketing website. It shows a short description about Typing Habit Gesture Authentication System and the features that are being offered.

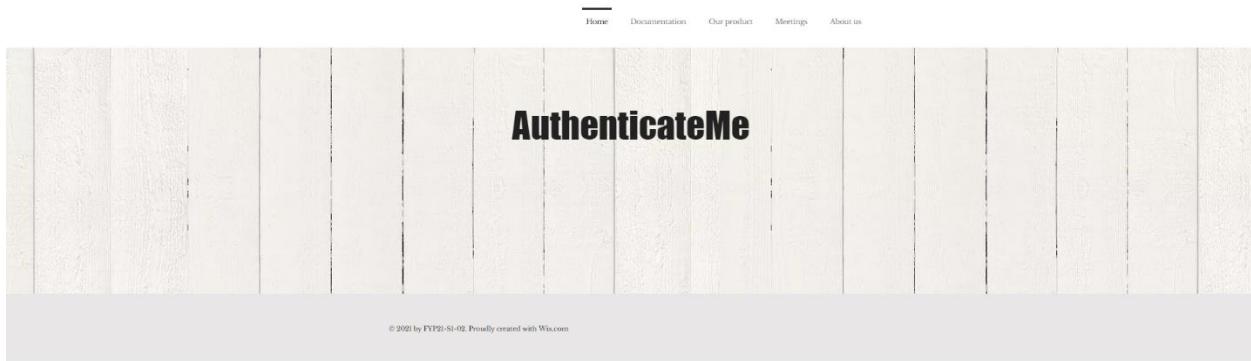


Figure 721. Screenshot showing the main page of AuthenticateMe website.

### 9.2 Documentation Page

Our documentation page consists of our project documentation and diagrams such as User Stories, Use Cases Diagram, Requirement Specification and Design Specification.

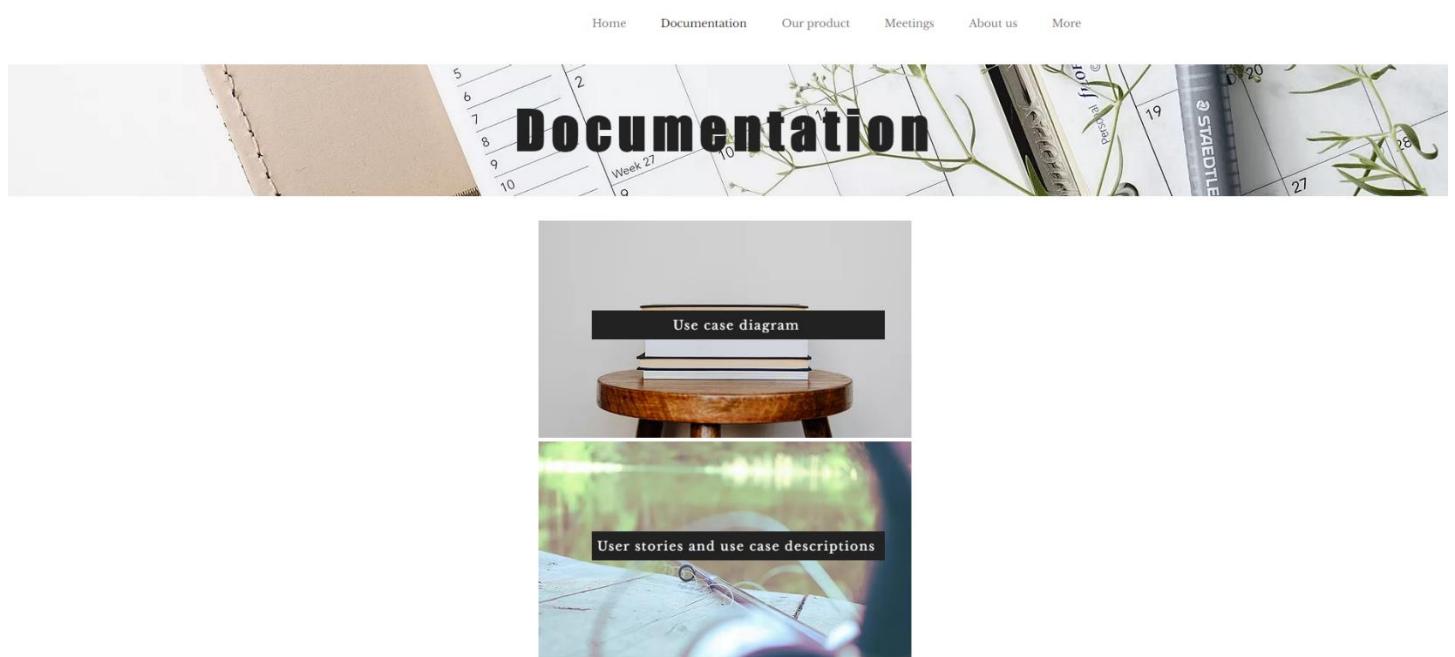




Figure 73. Screenshot showing the documentation page of AuthenticateMe website.

Home      Documentation      Our product      Meetings      About us

## 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 re-calibrate their typing habit 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 an account so that they can log in to the system.
7. As a user, they should be able to authenticate themselves using a security answer in case their typing habits change.
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 74. Screenshot showing on our user stories under “Documentation”.

## 9.3 Product Page

Our Product Page consists of our product video where it showcases our website application's walkthrough to show what to expect when the users visit our website. We also included screenshots of all our different web pages such as register (for details and typing habit) and login.

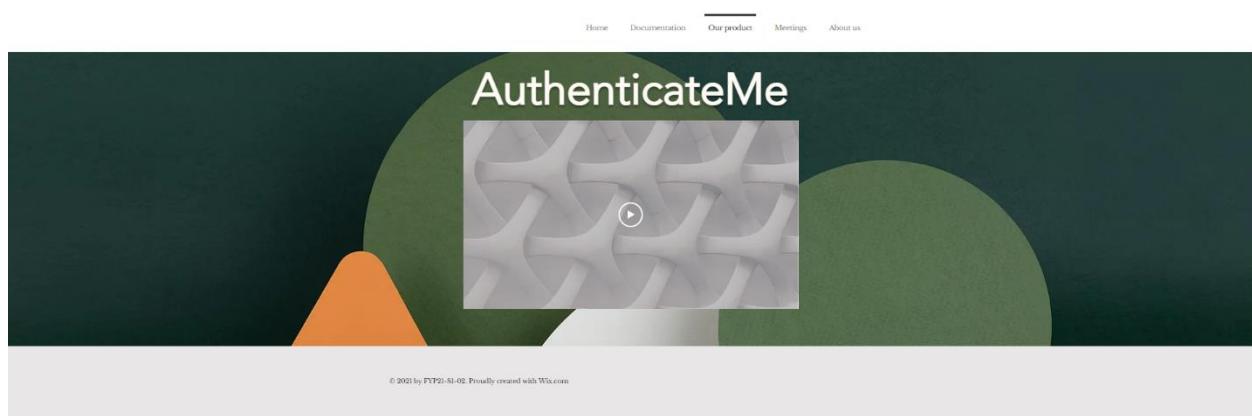


Figure 75. Screenshot showing our product page of AuthenticateMe website.

# Technical Manual

## 9.4 Meeting Page

You can take a look on our meeting minutes on our marketing website too.

The screenshot shows a website header with navigation links: Home, Documentation, Our product, Meetings (which is underlined), About us, and More. Below the header is a large banner with the word "Meetings". Underneath the banner is a grid of 10 rectangular boxes, each containing a meeting summary. The boxes are arranged in two columns of five. The first column contains: "Week 1 - Kickoff meeting", "Week 3 - Milestone meeting", and "Week 4 - Meeting 2". The second column contains: "Week 2 - Meeting 1", "Week 3 - Supervisor meeting 1", and "Week 5 - Meeting 3". The third column contains: "Week 6 - Meeting 4", "Week 7 - Meeting 5", and "Week 8 & 9 (Exam break)". The fourth column contains: "Week 10 - Meeting 6". The background of the grid is a light gray marble texture.

- Home
- Documentation
- Our product
- Meetings
- About us
- More

## Meetings

- Week 1 - Kickoff meeting
- Week 2 - Meeting 1
- Week 3 - Milestone meeting
- Week 3 - Supervisor meeting 1
- Week 4 - Meeting 2
- Week 5 - Meeting 3
- Week 6 - Meeting 4
- Week 7 - Meeting 5
- Week 8 & 9 (Exam break)
- Week 10 - Meeting 6

# Technical Manual

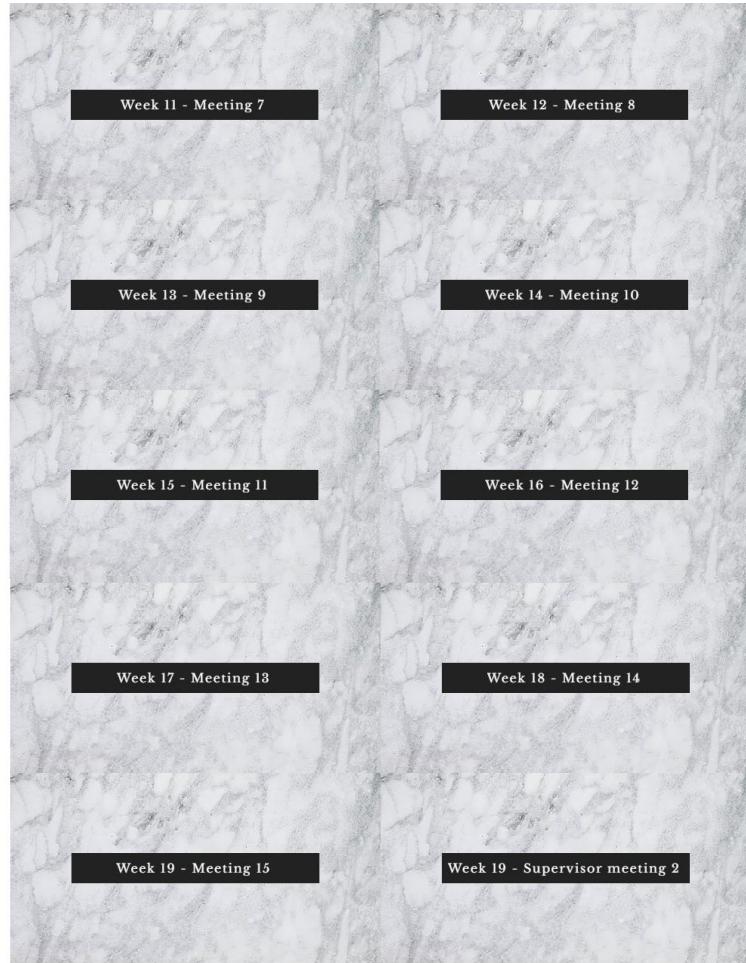


Figure 76. Screenshot showing the meetings page of AuthenticateMe website.

# Technical Manual

## 9.5 About Us Page

Our About Us Page shows the team members and their roles and responsibilities. Also, you can view our weekly diary as well.

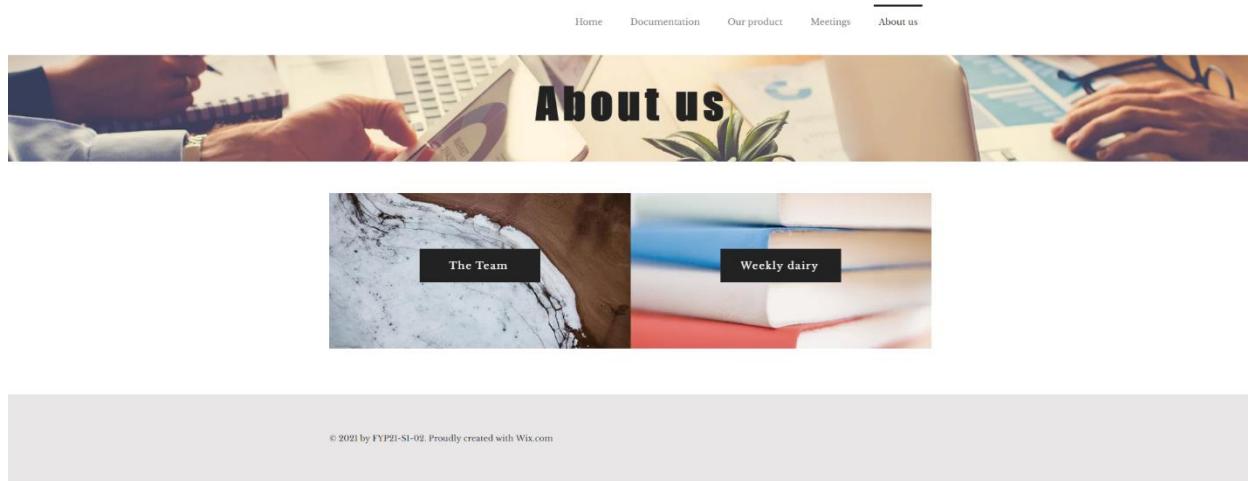


Figure 77. Screenshot showing the “About us” page of AuthenticateMe website.

# Technical Manual

## Our Team roles:

Home      Documentation      Our product      Meetings      About us

### Alicia Tan Chun Ying

Project manager / Developer

UOW ID:  
cyatan005@mymail.sim.edu.sg

Leads and directs the team to achieve the goal and has overall responsibility for the project's success.

Responsibilities include but not limited to:

- Planning
- Analysis
- Research
- Design
- Execution
- Monitoring
- Controlling
- Closure
- External and internal activities

The project manager is also responsible for undertaking the role of decision-maker for the team and delivers the expectations to the stakeholders. She has the responsibility, authority, and accountability of the overall project management and to coordinate with the rest of the internal (project members) and external stakeholders to obtain required resources.

#### **Developer**

Responsible for project design and development activities according to user specifications. He produces clean, efficient code based on specifications. He tests programs to ensure that they meet the requirements of the specification and that they are error-free.

She is responsible for troubleshooting, debugging, and upgrading existing software if necessary. Also, she creates technical documentation for reference and reporting.

### Terrence Yap Yong Ting

Lead developer

UOW ID:  
yttyap002@mymail.sim.edu.sg

Responsible for project design and development activities according to user specifications. He produces clean, efficient code based on specifications. He tests programs to ensure that they meet the requirements of the specification and that they are error-free.

He is responsible for troubleshooting, debugging, and upgrading existing software if necessary. Also, he creates technical documentation for reference and reporting.

As lead developer, he is accountable for the project members involved in programming (developers and testers), keeping them on track.



# Technical Manual

## Tng Min Li

Lead UX Designer / Documentation

UOW ID:

mltng001@mymail.sim.edu.sg

Responsible for overseeing the meeting minutes. These minutes are required for a formal account of the members who attended the meeting and the items discussed, as well as the actions agreed upon and to be carried out. She should plan, organize and update the relevant documents and reports.

## Che Cui Hui

Lead tester / Documentation

UOW ID:

chchea001@mymail.sim.edu.sg

Lead tester is responsible for reviewing software requirements and preparing test scenarios as well as executing tests on software usability. She prepares reports on all aspects related to the software testing carried out and reports to the design/developer team.

She is required to interact with the team to understand product requirements.

She participates in design reviews, providing input on requirements, product design and potential problems.

## Joel Teo Han Wen

Lead documentation

UOW ID:

jhwteo002@mymail.sim.edu.sg

Responsible for overseeing the meeting minutes. These minutes are required for a formal account of the members who attended the meeting and the items discussed, as well as the actions agreed upon and to be carried out. He should plan, organize and update the relevant documents and reports

As lead documentation, he is accountable for the project members involved in documentation, keeping them on track.



© 2021 by FYP21-S1-02. Proudly created with Wix.com

Figure 78. Screenshot showing on our team role under “About us.”

# Technical Manual

## Our Weekly Diary:

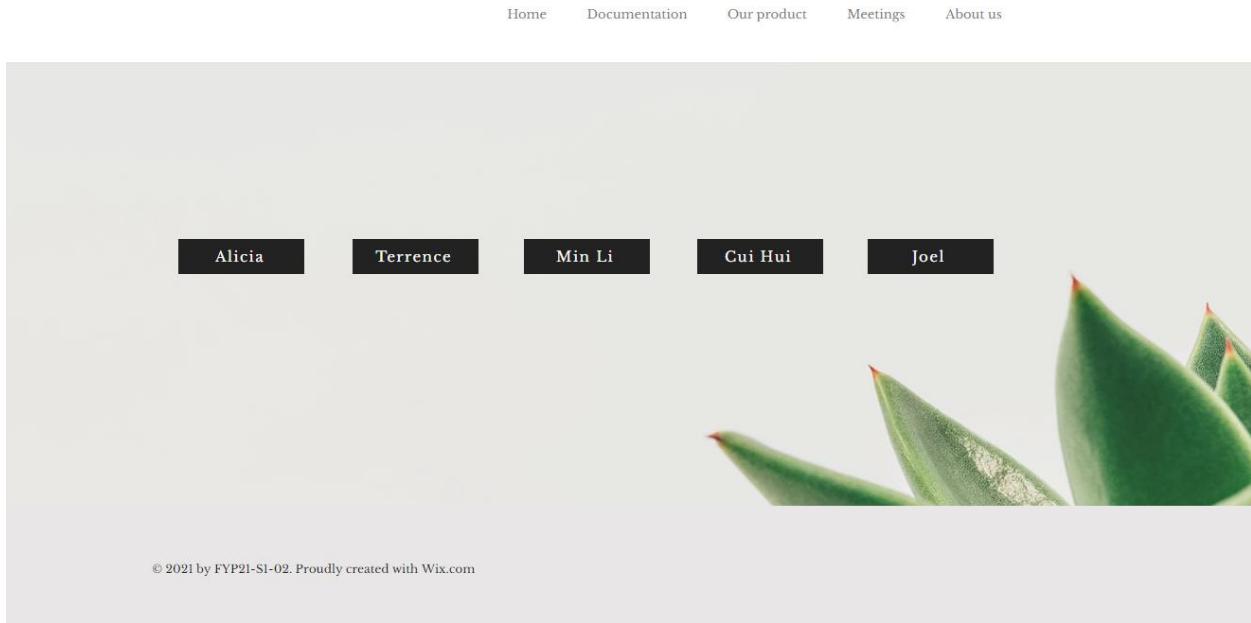


Figure 79. Screenshot showing on our team weekly diary under “About us”.

## Chapter 10: Security considerations

### 10.1. Privacy of Users

#### 10.1.1. Connection

Establishing a secure connection between the application and user is essential as it ensures that any individuals with malicious intentions are unable to eavesdrop and cause harm. Thus, the use of SSL (Secure Sockets Layer) certificate (JavaEE, n.d.) was adopted achieve a secure connection. SSL takes care of important considerations related to security:

1. Authentication of the website application and client

During the initial attempt to communicate with AuthenticateMe over a secure connection, the user's web browser will be presented with a public key certificate to verify that the site is legitimate. Sometimes, a request to the client may be made to prove that the client is legitimate.

2. Confidentiality between client and website application

The presence of SSL certificate ensure that third parties are unable to intercept data passed between users and AuthenticateMe. It ensures that the exchange is encrypted and confidential.

3. Integrity of data shared between client and website application

SSL certificate ensures that the data will not be modified when it is exchanged between client and AuthenticateMe.

With these security considerations in mind, SSL certificate was applied and implemented in AuthenticateMe to ensure a secure connection.

#### 10.1.2. Privacy of sensitive information

##### 10.1.2.1. Information stored in database

Users' login credentials are stored in the database to use in the website application. Login credentials include the following: username, first name, last name, security question, security answer.

Login credentials are securely stored in ClearDB using SSL encryption when communicating with our website application. The data stored in ClearDB are also encrypted, which enhances the safety of our users' data (ClearDB, n.d.).

The login credentials stored in ClearDB are stored in plain text except for security answer. Security answer is hashed using SHA-256 cryptographic hash algorithm. In the event users log in using their security question and answer, the hash of their answer is compared to the hash stored in the database.

SHA-256 algorithm is a cryptographic hash algorithm that outputs a 256-bit long value. SHA-256 is a one-way hash (N-Able, 2019). Thus, if the database is compromised, attackers would need a considerable amount of time to compare each answer with the hashed security answer.

# Technical Manual

## 10.1.2.2. Data collected about users' typing habit

Data on users' typing habit should be private and securely stored to ensure privacy of our users.

The data collected about users' typing habit will be securely stored in a separate file system. This is to ensure in the event of a breach in the system, the attackers would not be able to retrieve the users' data because Google Storage encrypts data on client-side and server-side, therefore data are securely stored (Cloud, n.d.).

## 10.2. Keylogger

AuthenticateMe makes use of keylogging activities to record the users' typing habits (accuracy, WPM, flight time and dwell time). The project team will ensure that this data will be collected solely for functional purposes during login, registration, and recalibration of habits.

The use of the keylogger in the website application is also stated transparently and clearly to users in the private policy document available for all website visitors to view.

## 10.3. Logging in

Users are given 3 tries to validate their typing habit before they are redirected to the different login page where they would have to answer their security question entered during registration. During the security login, users are given a maximum attempt of 3 tries before they are locked out of the website for 1 minute.

This is to reduce the possibility of brute force attacks (Esherdan, et al., n.d.), as attackers would need to try 3 answers at a time and get locked out before they can try again.

## Chapter 11: Future directions

Based on the survey results shown in section 8.2, there are several points of improvement to be made that have not been worked on due to the scope or time frame.

### 11.1. Keylogger

#### 11.1.1. Improvement on key detection

Currently, our keylogger has issues occasionally as seen in the survey results in section 8.2. The errors faced are mainly “Error 500”. This is due to errors faced in the keylogger as the keylogger might detect more keys than typed due to errors with regards to the detection of backspace. Users might hold down the backspace to delete more than 1 key. However, the keylogger detects the backspace as deleting one key and thus there will be extra typing habit variables recorded. This returns a logic error as the model is unable to accept the extra keys.

Thus, future suggestions for this project would be to come up with a solution to improve the keylogger to encompass simultaneous key deletions.

#### 11.1.2. Include more types of keyboards

This project focuses on the US computer keyboard with QWERTY layout. Thus, if a user were to use another keyboard layout, such as a German keyboard layout or non-QWERTY the keylogger might face issues detecting arrow keys or backspace as they might be located differently on different keyboard layouts (Keyboard: keydown and keyup, 2021).

### 11.2. Machine learning algorithm

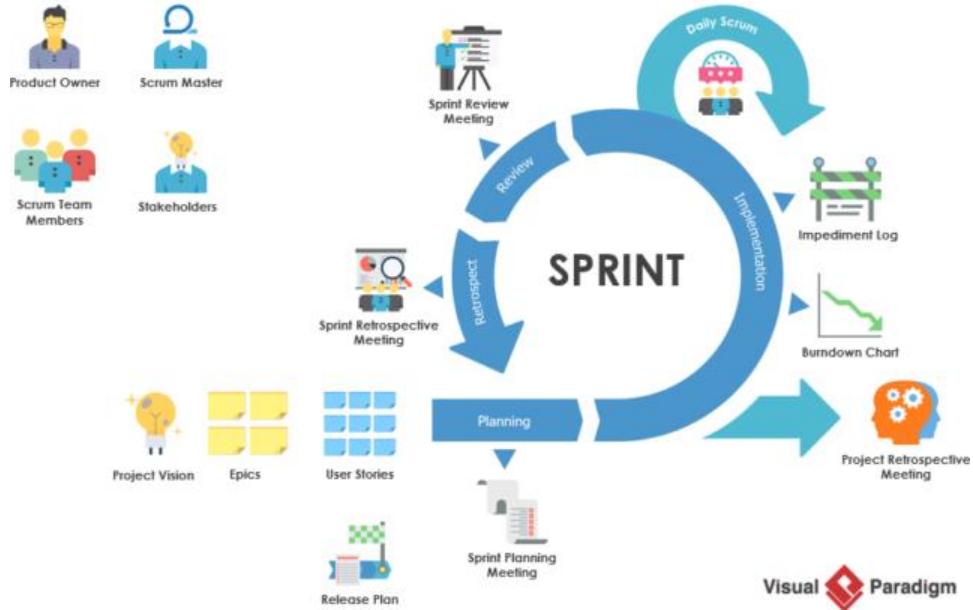
#### 11.2.1. Reducing the number of attempts

In the survey results, quite a number have provided feedback that they felt typing the sentence 10 times was too tedious as a user. During development, we had tried to reduce the number of attempts users needed to type during registration and recalibration. However, this significantly affected the model accuracy to authenticate users. This was the same even with longer sentences.

Thus, one of the future directions of this project is to reduce the number of attempts while maintaining the accuracy.

## Chapter 12: Appendix

### The Agile – Scrum Framework



Source: (Blerot, 2020)

## Chapter 13: References

- Bassey, P. (2019, September 19). *Logistic Regression Vs Support Vector Machines (SVM)*. Retrieved May 10, 2021, from Medium: <https://medium.com/axum-labs/logistic-regression-vs-support-vector-machines-svm-c335610a3d16#:~:text=SVM%20tries%20to%20finds%20the,are%20near%20the%20optimal%20point>.
- Blerot, O. (2020, May 26). *About Agile Organization*. Retrieved May 10, 2021, from Ossia Belgium: <https://www.ossia-belgium.com/about-agile-organisation/>
- ClearDB. (n.d.). *ClearDB FAQs*. Retrieved May 18, 2021, from ClearDB: [https://w2.cleardb.net/faqs/#general\\_17](https://w2.cleardb.net/faqs/#general_17)
- Cloud, G. (n.d.). *Data encryption options*. Retrieved from Google Cloud: <https://cloud.google.com/storage/docs/encryption>
- CloudFlare. (n.d.). *What is SSL?* Retrieved May 18, 2021, from CloudFlare: <https://www.cloudflare.com/learning/ssl/what-is-an-ssl-certificate/>
- Daily, B. N. (2020, 25 February). *What Is Agile Scrum Methodology?* Retrieved May 10, 2021, from Business News Daily: <https://www.businessnewsdaily.com/4987-what-is-agile-scrum-methodology.html>
- Esheridan, KirstenS, McMillan, P., Raesene, Adedov, Dinis.Cruz, . . . kingthorin. (n.d.). *Blocking Brute Force Attacks*. Retrieved May 16, 2021, from OWASP: [https://owasp.org/www-community/controls/Blocking\\_Brute\\_Force\\_Attacks](https://owasp.org/www-community/controls/Blocking_Brute_Force_Attacks)
- Guru99. (n.d.). *Agile Vs Scrum: Difference Between Agile Methodology & Scrum*. Retrieved May 20, 2021, from Guru99: <https://www.guru99.com/agile-vs-scrum.html>
- JavaEE. (n.d.). *Establishing a Secure Connection Using SSL*. Retrieved May 17, 2021, from The Java EE Tutorial: <https://javaee.github.io/tutorial/security-intro006.html>
- Keyboard: keydown and keyup. (2021, January 3). Retrieved May 18, 2021, from Javascript Info: <https://javascript.info/keyboard-events>
- Md Liakat Ali, J. V. (2017, March). Keystroke Biometric Systems for User Authentication. *J Sign Process Syst*, 86 (2), 176-179. doi:10.1007/s11265-016-1114-9
- Muslihat, D. (2018, March 2). *Agile Methodology: An Overview*. Retrieved May 2, 2021, from Zenkit: <https://zenkit.com/en/blog/agile-methodology-an-overview/>
- N-Able. (2019, September 12). *SHA-256 Algorithm Overview*. Retrieved May 15, 2021, from N-Able: <https://www.n-able.com/blog/sha-256-encryption>
- Perrig, A. (2015, June 15). *Shortcomings of Password-Based Authentication*. Retrieved from Usenix: [https://www.usenix.org/legacy/publications/library/proceedings/sec2000/full\\_papers/dhamija/dhamija\\_htm/node2.html](https://www.usenix.org/legacy/publications/library/proceedings/sec2000/full_papers/dhamija/dhamija_htm/node2.html)
- Pin Shen Teh, A. B. (2013, August 29). A Survey of Keystroke Dynamics Biometrics. *The Scientific World Journal*, 2013, 3. Retrieved from <http://dx.doi.org/10.1155/2013/408280>
- TypingDNA. (2020, November 6). *Authentication API Documentation*. Retrieved May 10, 2021, from TypingDNA: <https://api.typingdna.com/index.html>

## **Technical Manual**

Varghese, D. (2018, December 7). *Comparative Study on Classic Machine learning Algorithms*. Retrieved 10 May, 2021, from Towards data science: <https://towardsdatascience.com/comparative-study-on-classic-machine-learning-algorithms-24f9ff6ab222#:~:text=Logistic%20regression%20vs%20SVM%20%3A,outperforms%20log%20loss%20in%20LR>.