Password Manager with
Keystroke Dynamics Authentication System

A Minor Project Report in partial fulfillment of the degree

Bachelor of Technology

in

Computer Science & Engineering

By

19K41A0517	Mohammed Raamizuddin
19K41A0513	K. Rishinandan
19K41A0560	V. Sai Likhitha

Under the Guidance of

Dr. T. Sampath Kumar

Submitted to

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
S.R ENGINEERING COLLEGE(A), ANANTHASAGAR, WARANGAL
(Affiliated to JNTUH, Accredited by NBA)

May-2022

SR Engineering College
Innovation . Creativity . Entrepreneurship

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

CERTIFICATE

This is to certify that the Minor Project Report entitled "Password Manager using Keystroke Dynamics Authentication System" is a record of bonafide work carried out by the student(s) K.Rishinandan, Mohammed Raamizuddin, V.Sailikhitha bearing Roll No(s) 19K41A0513, 19K41A0517, 19K41A0560 during the academic year 2021-22 in partial fulfillment of the award of the degree of ***Bachelor of Technology*** in **Computer Science & Engineering** by the Jawaharlal Nehru Technological University, Hyderabad.

Project Guide **Head of the Department**

External Examiner

ACKNOWLEDGMENT

We owe an enormous debt of gratitude to our project guide **Dr. T. Sampath Kumar, Assistant Professor,** as well as Head of the CSE Department **Dr. Sheshikala, Associate Professor** for guiding us from the beginning through the end of the minor project with their intellectual advice and insightful suggestions. We truly value their consistent feedback on our progress, which was always constructive and encouraging and ultimately drove us to the right direction.

We express our thanks to project coordinators **Dr. Mohammed Ali Shaik, Asst. Prof., Dr. P. Praveen, Asst. Prof.** for their encouragement and support.

We wish to take this opportunity to express our sincere gratitude and deep sense of respect to our beloved principal, **Dr. V. Mahesh,** for his continuous support and guidance to complete this project in the institute.

Finally, we express our thanks to all the teaching and non-teaching staff of the department for their suggestions and timely support.

ABSTRACT

In the last decade with the evaluation of the internet, computer security importance appeared rapidly with the need to provide higher level of security for information systems due to the network attackers and intrusions. Finding new techniques and currently upgrading ones those used to allow only the legitimate users to access sensitive data take a big area of computer security research. The usual way of authentication system that depends on password validation still not enough to provide required high level of security. So, we need more intelligent ideas to achieve our purposed security goals.

Keystroke dynamic technique is one of these ideas that can provide us with better security for our protected information. Keystroke identification differ from other biometric techniques where it doesn't require any special security hardware, it requires only the already existing computer keyboard to measure keystroke dynamic which differ from user to user. Each user has a typing signature for keystroke which relate to the keystroke latencies and pressure measures. These two measures are unique for each user.

In our project, we implemented a keystroke dynamics technique as a way of user's identifications enchantments. Statistical data is collected from the user typing inputs to create unique signatures for user and saved in special profiles. In our implementation we compare generated typing authentication trials with constructed user's profiles stored in database and authenticate it. This technique is applied to the application Password manager where the authenticated user stores the passwords and can perform all the primary operations like add, delete, update and retrieve.

TABLE OF CONTENTS

1. INTRODUCTION:

With the increasing of sensitive information systems and network intruders the trends of protecting the sensitive data. Preventing unauthorized users from accessing your secured data is one of the main requirements of modern security research. Our problem is to provide high level of security authentication and better user identification. Verifying the user who is accessing information resources. So, we need more robust techniques to authenticate and identify the system users before accessing the resources.

The commonly known technique usernames and passwords which is used for user authentication has a need to a lot of improvements. Usernames and passwords can be stolen and can be guessed using brute force attack algorithms. Especially in the case of low complexity passwords. There are many alternatives to passwords technique, one of them is the user biometrics just like figure print, voice signature, iris scanning and hand geometry. There is a common issue between these alternatives that it needs a special hardware.

Using the keystrokes dynamics, we can achieve a new away of user identifications with small cost. No need for additional equipment because it is already use existing keyboard. Keystrokes dynamics used to describe the unique typing pattern of users. This pattern hardly unique for each user and it can be same in some cases for similar typing text. With implementing this technology, we can increase the security accessibility of our information system using inexpensive and easy technique.

This technology is applied to the application password manager for demonstration which can hold the passwords of the user who is using it, the user registers for the application with the process of keystroke biometrics identification system and then log’s in with the same system to enter the password manager module, this module will be specific to that user only which shows the passwords stored by him/her at any point of login previously. The user can add a password, update a password, delete a password, and retrieve a password from the password manager module. The passwords are stored in a database. Multiple users can create an account for using this system.

All the details related to the account like the Domain, username, password shall be displayed in a tabular format. The password is visible in text format. The user can copy the password and view the password details. The user can save any number of accounts information and any number of domains using this system. By this password manager user doesn’t have to remember all the passwords he/she has.

2. LITREATURE SURVEY:

The primary part of authentication for password i.e., Keystroke dynamics have been discussed in many papers of identity verification with differences in users input samples and information and in the way of analysis and implementation. In the following part we will list the previous work in keystroke dynamics:

In 1985:

Umphress and Williams made a digraph for latencies between the captured keystrokes. They worked on two different data sets. Each of them consists of different number of characters, where the reference profile has 1400 characters, and the test profile has 300 characters. The results of this method proved the validity of keystrokes biometrics as an identity verification technique. But this study has some limitations because it requires large amount of input characters, and the False Acceptance Rate (FAR) achieved was 6% which is high for a verification method because says FAR should be less than 1% for systems accessibility.

In 1988:

Williams and Leggett extended their previous study in 1988. They increased the number of users in the study and reduced the experimental variables and discarded some not useful digraphs. By these modifications they reduced the FAR to 5% which still not applicable for verifying login.

In 1990:

Leggett et alia took the concept of the keystroke stroke dynamics into dynamic environment testing verification. Dynamic environment means that the users' verification occurs while they type the test profile, and this allow real-time verification of identity. This fixed the problem of time of check to time of use (TOCTTOU). This problem is defined as that the verification is done only once at login only, while someone else can use this access during the same session later. The results of Legget et alia were based on sequential statistical theory and it were a FAR rate of 12.8% and a False Reject Rate (FFR) of 11.1%. These experiments proved that dynamic user identification is possible and can be better with more enhancements.

In 2000:

Haider et alia discussed the approaches of analyzing user keystroke pattern. These approaches are neural networks, statistical methods, and fuzzy logic. With username and passwords, the best FAR was 2% using a combination of our approaches and for sure not acceptable. The best result was for the statistical approach. To increase the system performance Haider suggests recording the keystrokes period.

3. DESIGN:

3.1 - REQUIREMENT SPECIFICATION (S/W & H/W):

Hardware Requirements:

✓ **System**	:	Pentium 4, Intel Core i3, i5, i7 and 2GHz Minimum
✓ **RAM**	:	4GB or above
✓ **Hard Disk**	:	10GB or above
✓ **Input**	:	Keyboard and Mouse
✓ **Output**	:	Monitor

Software Requirements:

✓ **OS**	:	Windows 7 or Higher Versions
✓ **Platform**	:	Eclipse IDE
✓ **Program Language**	:	Java (JDK17)

3.2 - FLOW CHART:

The whole approach is depicted by the following flowchart:

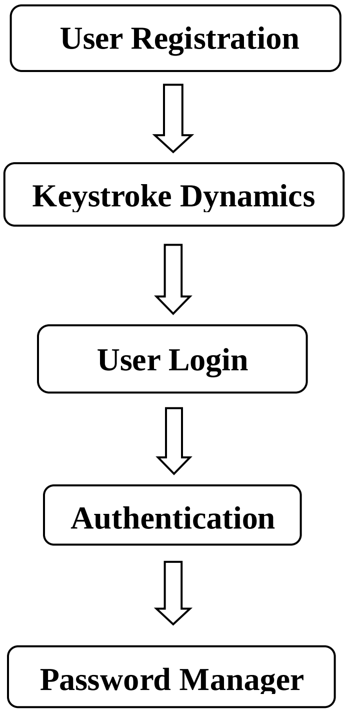


Figure 1 - Flow chart of the System**.**

3.3 – USE-CASE DIAGRAM:

This diagram represents the overall use-cases that are present in the system, the fingerprint use case represents the keystroke dynamics system part.

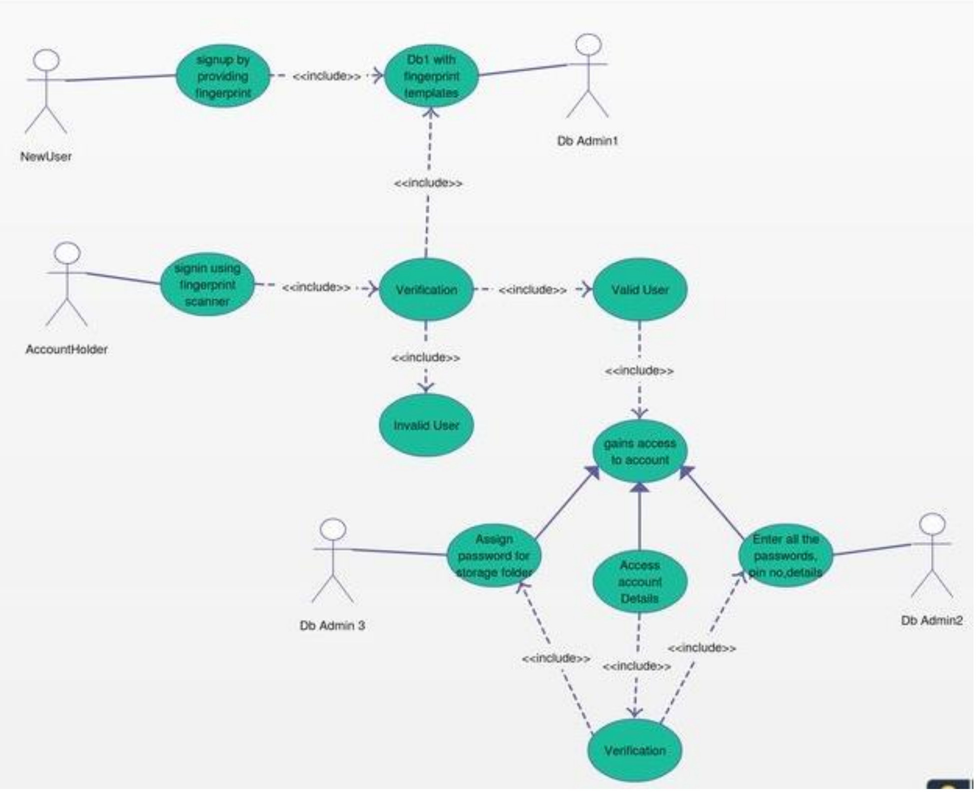


Figure 2 – Use-case diagram**.**

3.4 – ER DIAGRAM:

The ER Diagram represents the databases utilized in the system.

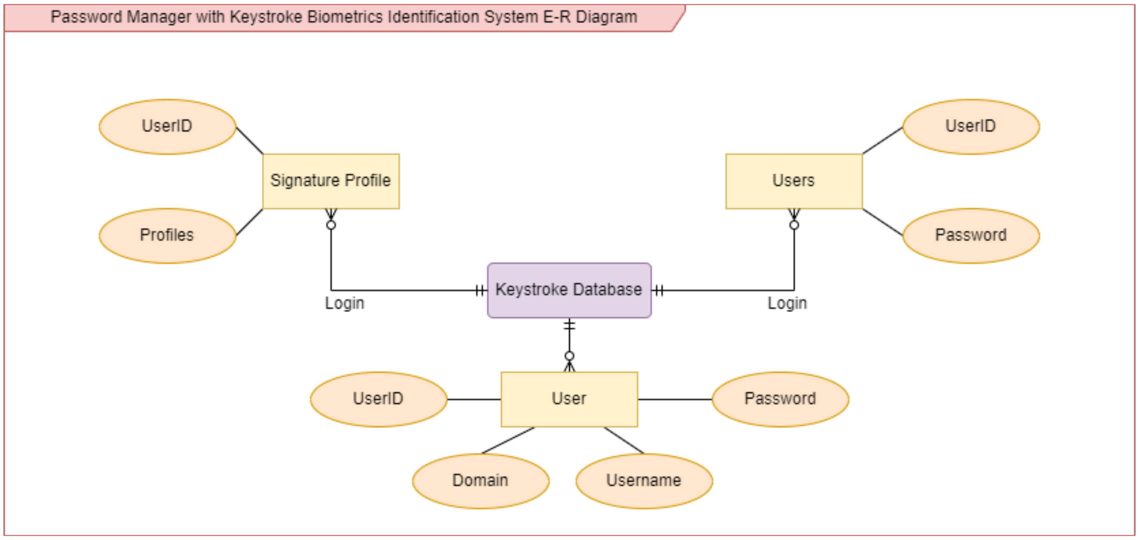
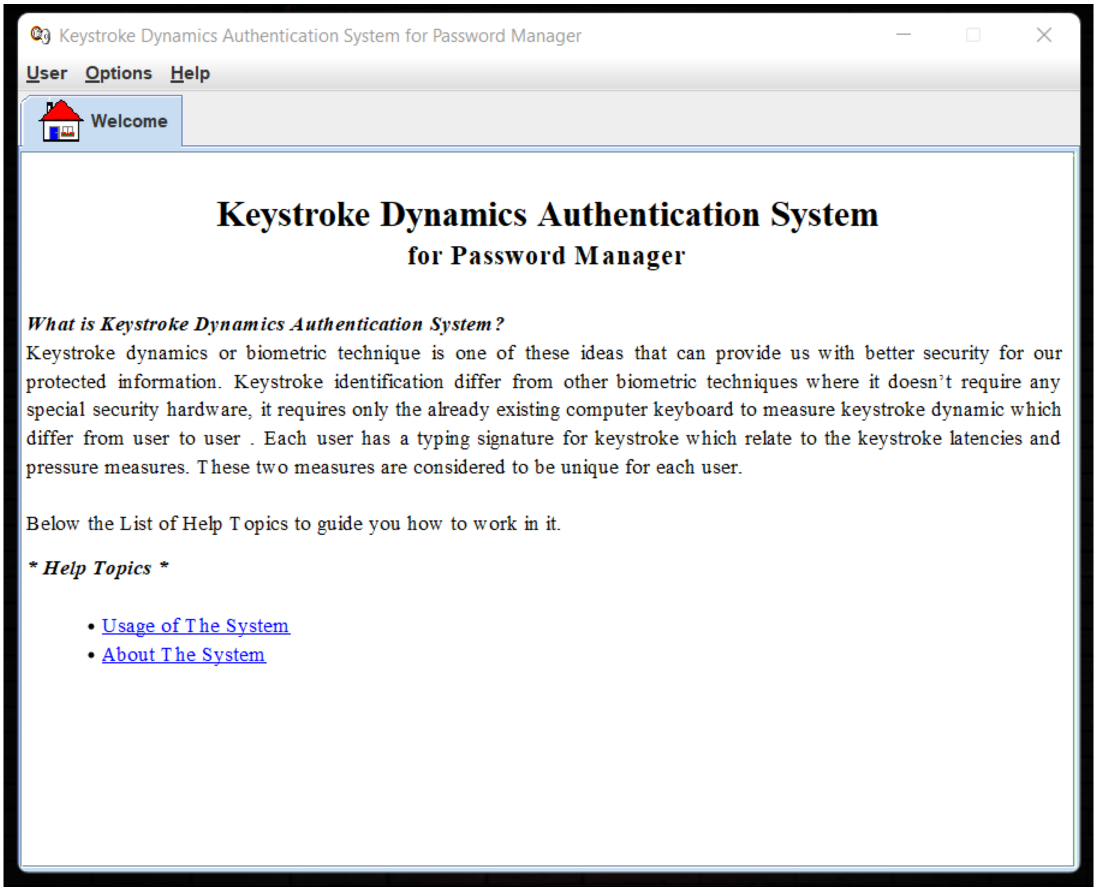


Figure 3 – E-R diagram**.**

4. IMPLEMENTATION:

4.1 Outline

The program has two operating modes, a learning mode where an initial profile is generated for the username, user phrase and standard phrase and a verification mode where the user input is compared to the reference template and is classified as being a successful or unsuccessful login into the password manager. A screenshot of the main entry point to the program is given in *Figure 4*. To give a realistic indication of a real login process the username selected by the user must be between 5 and 12 characters and the user phrase must be between 8 and 25 characters long. Finally, the standard phrase that all users must type in is “the brown fox”. The actual phrase is essentially irrelevant, it was merely chosen because the characters of the phrase are spread out across the keyboard, thereby giving larger inter-key delays that are useful for analysis. A longer phrase such as “the quick brown fox jumped over the moon” was considered, the problem was that the average user kept making typographical errors and would have to restart the login process, as well as the fact that for the larger string the variability tended to be much higher for less advanced typists.


Keystroke Dynamics Authentication System for Password Manager
User
Options
Help
Welcome

Keystroke Dynamics Authentication System

for Password Manager

What is Keystroke Dynamics Authentication System?

Keystroke dynamics or biometric technique is one of these ideas that can provide us with better security for our protected information. Keystroke identification differ from other biometric techniques where it doesn’t require any special security hardware, it requires only the already existing computer keyboard to measure keystroke dynamic which differ from user to user . Each user has a typing signature for keystroke which relate to the keystroke latencies and pressure measures. These two measures are considered to be unique for each user.

Below the List of Help Topics to guide you how to work in it.

**** Help Topics ****

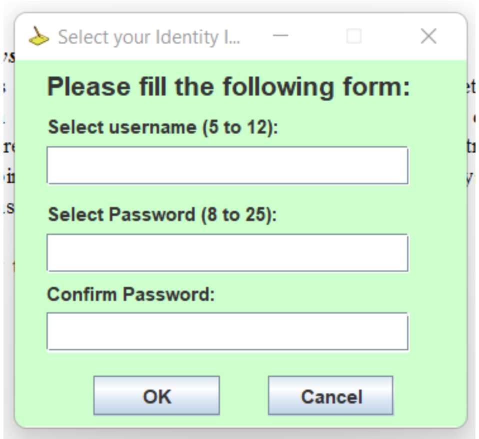
- Usage of The System
- About The System

Figure 4 – Entry point of Program**.**

4.2 Modules

4.2.1 Module-1: Learning Login

To build an initial keystroke profile for each user the program enters the learning login mode. First a user is asked to select a username and a user phrase as described in section 4.1. Once this has occurred the user is requested to enter their login credentials a total of 5 times. During the developmental phase of the program all five logins were used to develop a profile. The profile for the string “goldenau” is shown in figure 6. All login attempts are appended to a log file so that further analysis is possible, this is in additional to the signature file that store the user’s signature for each string, and finally there is a configuration file that stores the number of logins, the number of accepted logins and the how many of each of the strings have matched their respective profile.



Select your Identity I...

Please fill the following form:

Select username (5 to 12):

Select Password (8 to 25):

Confirm Password:

Figure 5 – Learning Login

Latency Digraph	Mean Latency	Sum of X	Sum of X Squared
G	530	19089	10276930
G-O	907	32644	30398770
O	434	15629	6879291
O-L	1794	64588	116126600
L	564	20301	1160640
L-D	960	34573	34414060
D	447	16101	7350737
D-E	1915	68924	132168900
E	538	19354	10513950
E-N	884	31825	29311090
N	582	20959	12379290
N-A	1678	60418	102627900
A	564	20292	11749020
A-U	1069	38480	43294730
U	534	19213	10584790

Figure 6 – Profile for “goldenau”.

Initially, all the 5 logins in the learning mode were used to generate the template. The value of the latencies was added to the sum of X column and the square of the latency was added to the sum of the square column. What was discovered was that there was a high degree of variability in the initial login of users. This would create a very high tolerance and would therefore more easily allow False Accepts from invalid users. One limitation of the program is that it does not allow users to use the delete or backspace key to correct any mistakes they have made while typing in the three strings. While it is possible to add the ability for the program to handle this, the user's typing pattern would be disturbed by using the Delete and Backspace and the latencies would probably deviate too far from the profile anyway. During the input if a user tries to modify previously typed text the login is reset, and they are required to start again.

4.2.2 Module-2: Verification Login

The verification login forms the core of the program. A screen shot of this is shown in figure 6.

From the user's perspective the learning login and the verification login are the same. In both cases the strings that have been selected by the user are displayed on the form. This is done so that they don't have to worry about memorizing the word, as this is not the purpose of the study. The first stage in the verification process is to timestamp and stores all key press and release events for each login. Assuming the user has not made any typographical errors or used the delete or backspace key, which will reset the form anyway, the raw key traces are reordered into key depress events and inter-key delay digraphs in a similar fashion as is shown in *figure 7*.

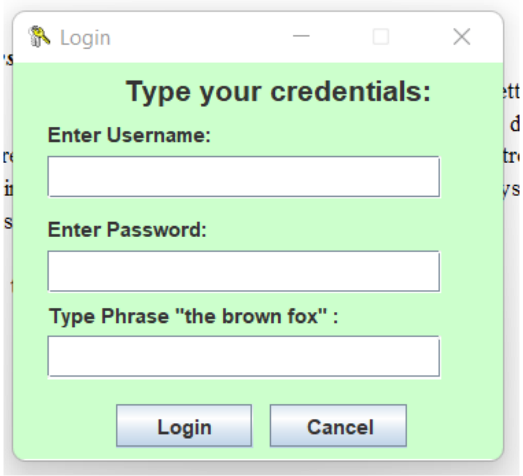
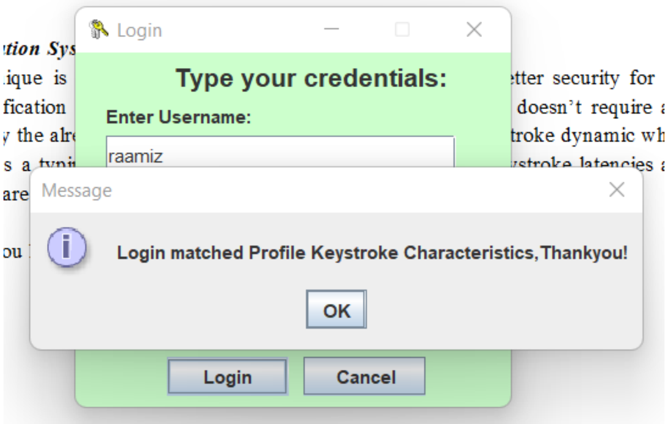


Figure 7 – Verification Login**.**

The algorithm established in section 4.2.1 is used to perform the validation process for each string. Initially, the tolerance level was set to 0.5 standard deviations. This was found to be too high and so the criteria for latency match were reduced and now for an approval for a login string to occur only 50% of latencies must fit within one standard deviation away from the mean of the reference profile. Another change was made so that for each valid latency that fits within the standard deviation, the sum of the weights as described in *figure 6* must add now add up to at least 75%. This gives slightly more emphasis to the weighted system. For a login to be valid, at least two of the three login strings must meet the criteria of having greater than 50% of the latencies within one standard deviation of the mean or the weighting system must give a score of 75% or greater. A screenshot of a matched profile is given in *figure 8.*



Enter Username: raamiz

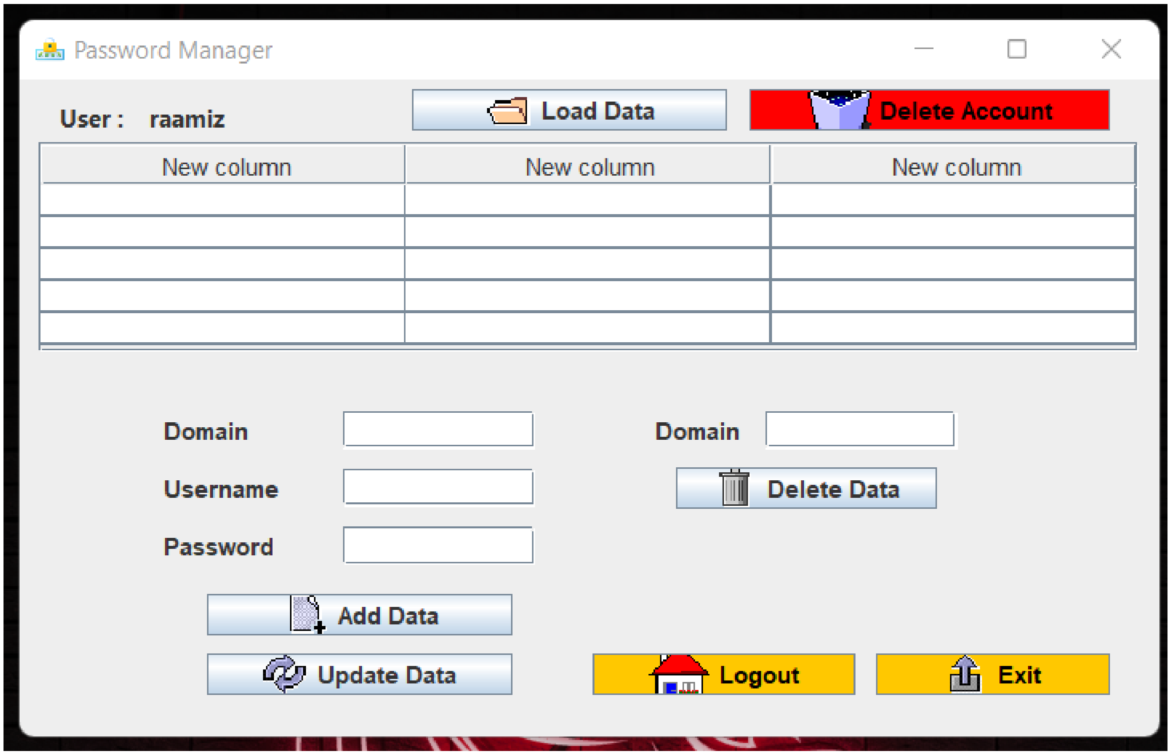
Login matched Profile Keystroke Characteristics,Thankyou!

Figure 8 – Login Message**.**

As mentioned in the requirements the profile had to be able to adapt to gradual changes to the users' typing patterns. This is implemented by adding the key latencies value and the square of the key latency value as described in the Learning Login section to update the mean and standard deviation for the profile. Even if the overall login is not approved, if an individual username, user phrase or standard phrase meets the verification criteria the data will be merged into the signature profile for that string, therefore continuously modifying itself to changes in the user's typing patterns.

4.2.3 Module-3: Password Manager

After the successful login from the verification part the user will be redirected to the password manager module where the user can perform multiple operations like storing, deleting retrieving the data from the database. The screenshot of this module is given in *figure 9.*



User : raamiz

Figure 9 – Password Manager**.**

In the above figure, the various buttons depict the various operations that can be performed on the data in the database for adding and creating passwords the three fields domain, username and password must be filled and to delete the password the domain field should be filled, and the respective options will be executed. The load data option will populate the table present in the module with the usernames and passwords that are stored by the user. The delete account button will delete the users account from the keystroke biometric identification system and the password manager.

4.3 Methodology (Overview of technologies used)

4.3.1 Source Data Used

During a user typing session, we can collect different types of data and information that can be considered as a unique pattern for each user. Most of these data to be gathered are related to time while the user is typing on the keyboard. Such as the latencies between successive keystrokes that represent the time from pressing some key on the keyboard until the time of depressing another key. There are no limitations about the number of keys that we measure the latencies between them where we can calculate the latencies between more than two successive keys not only to form digraphs but to tri-graphs, and so on. The second type of data that we can collect during user typing session is the keystroke duration that can be considered as unigraphs where it represents the time from pressing some individual key until it is released.

The third measure is the keystroke pressure, but this biometric measure has some problem because it requires special types of keyboards that allow calculating the pressure on the keyboard keys and because most of the currently used computer's keyboards have only two statuses for their keys that indicate the position of the key is on or off, this means that the keystroke pressure is not applicable for these types of keyboards. And if we built systems that require special types of keyboards these lead for losing one of the keystroke biometric advantages mentioning that keystroke biometric does not require special type of hardware.

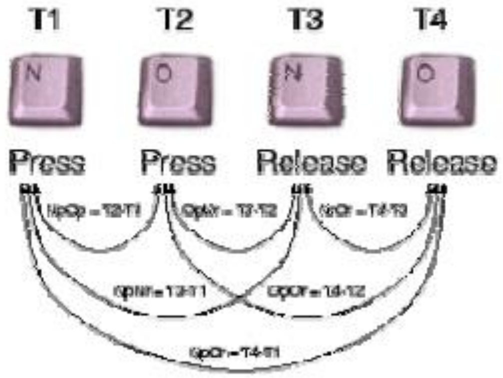


Figure 10 - Digraph latencies for the Typed Pattern "no".

4.3.2 Keystroke Analysis Methods

Some of Keystroke analysis techniques that can be used as a data analysis for the keystroke and identity verification process are fuzzy logic, neural networks, and statistical techniques as well as combinations of these approaches. the statistical methods have been proved as the most accurate. The statistical methods can be used in two different paradigms, comparing by using any distance measure and classification using the Bayes classification algorithm. Using the neural network as an analysis technique is effective but it has a major disadvantage, the neural network must be retrained when a new user is proposed and added to the system database.

The keystroke analysis method used for verification is statistical analysis. The exact nature of the analysis is not determined until after some initial testing has been performed. To achieve the objectives of the project as outlined in section 1.3 there are several requirements that the software needs to achieve.

- The program needs to use statistical analysis methods to perform keystroke identity verification.

- A signature profile for each token i.e., username, user phrase/password and standard phrase needs to be generated during the learning process and then used for comparison in the verification mode.
- The program will capture a username, user phrase/password and standard phrase for all users.
- The standard phrase is used as a point of comparison for latencies of the same phrase and may demonstrate the possible uniqueness of an individual’s keystroke patterns.
- The program will have a learning mode where a user will type the strings in several times as well as a normal login mode.
- It is also needs to incorporate a continues learning feature so that the user profile will adapt to changes in the user’s keystroke patterns over time.
- A log of all user attempts at identity verification needs to be kept; this is mainly all the keystroke’s latencies for the username, user phrase/password and standard phrase.
- The response time for verification needed to be short.

4.3.3 Development of Keystroke Verification Algorithm

The basis for the algorithm is to develop a profile signature of keystroke latencies for each of the three strings that the user would type during the login process. The basic process is to compare the latencies of each login and see if they fall within a certain number of standard deviations from the mean reference latency for each digraph. If more than 80% of all the possible latencies passed this test, then input for that string would be considered valid. The usual way of calculating standard deviations, *figure 10* requires that all the login digraph times need to be stored; the population standard deviation can then be calculated from all these digraphs. This process means that every previous login needs to be stored just so that the standard deviation can be calculated.

$$s = \sqrt{\frac{\sum\left(x-\overline{x}\right)^2}{n}}$$

Figure 11 – Standard Deviation Formula.

Manipulating the above standard deviation formula, a method can be used that does not require all the previous latency values for the standard deviation to be calculated. The formula, shown in *figure 11*, only uses the value for the sum of the values for each latency and the squared sum for each latency. This enables the standard deviation to be counted more quickly and reduces the amount of storage space required for the user’s profile.

$$s = \sqrt{\frac{\sum X}{n} - \left(\frac{\Sigma X}{n}\right)^2}$$

Figure 12 – Sample Standard Deviation Formula.

The main problem with using the standard deviation directly is that a digraph maybe be approved even though it has a high variability and consequently allows many more digraphs to be approved because the allowance for error is high. So that even if more than 80% of latencies fit within one standard deviation of the reference profile, it may not be significant if the variability is high. A way to alleviate this problem is to assign a weight to each of the deviations with regards to how the standard deviation compares with the mean as a percentage. Meaning that if the standard deviation, in comparison to the mean is high, then the approval for that latency would have a low weighting. The weight for each latency using this approach is given below in *figure 12*.

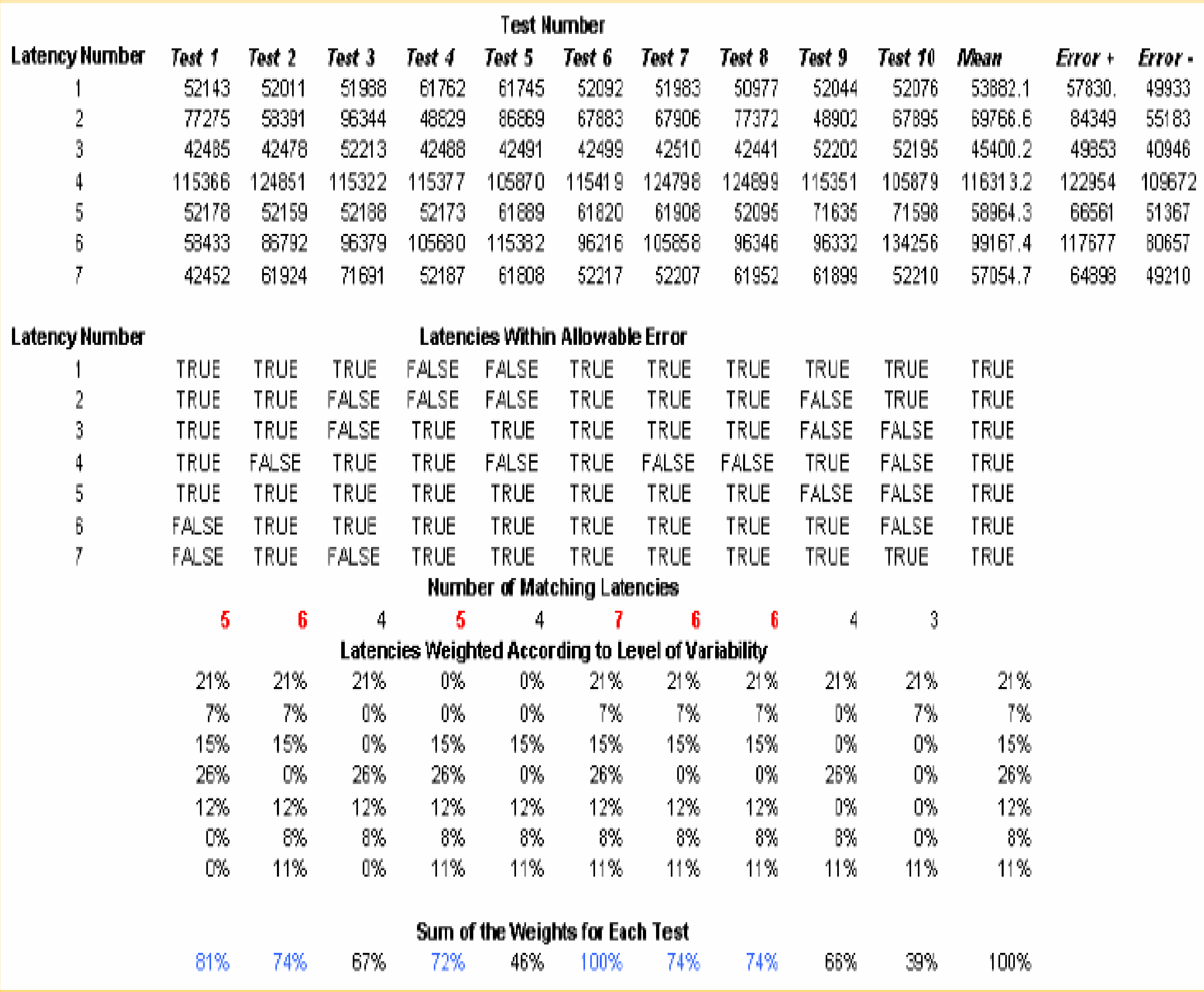
$$W = \left(\frac{1}{\sum \left(\frac{\overline{x}}{s} \right)} \times \left(\frac{\overline{x}}{s} \right) \right) \times 100\%$$

Figure 13 – Formula for verification using weighted latencies.

$$Approved = \sum W \geq 70\%$$

Where x dash is the mean, ‘s’ the standard deviation and W the weighted value for each latency.

This approach can be illustrated using the following table that displays how the keystroke latencies in figure two are approved using the previously mentioned formulas. To be approved more than 80% of latencies must fit within one standard deviation of the reference profile and for each valid latency that fits within the standard deviation the sum of the weights must add up to at least 70%. These figures may be modified during the development of the software and initial user trials.



Test Number

Latency Number	Test 1	Test 2	Test 3	Test 4	Test 5	Test 6	Test 7	Test 8	Test 9	Test 10	Mean	Error +	Error -
1	52143	52011	51988	61762	61745	52092	51983	50977	52044	52076	53882.1	57830.	49933
2	77275	58391	96344	48829	86869	67883	67906	77372	48902	67895	69766.6	84349	55183
3	42485	42478	52213	42488	42491	42499	42510	42441	52202	52195	45400.2	49853	40946
4	115366	124851	115322	115377	105870	115419	124798	124899	115351	105879	116313.2	122954	109672
5	52178	52159	52188	52173	61889	61820	61908	52095	71635	71598	58964.3	66561	51367
6	58433	86792	96379	105680	115382	96216	105858	96346	96332	134256	99167.4	117677	80657
7	42452	61924	71691	52187	61808	52217	52207	61952	61899	52210	57054.7	64898	49210

Latencies Within Allowable Error

Latency Number											
1	TRUE	TRUE	TRUE	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
2	TRUE	TRUE	FALSE	FALSE	FALSE	TRUE	TRUE	TRUE	FALSE	TRUE	TRUE
3	TRUE	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE	FALSE	TRUE
4	TRUE	FALSE	TRUE	TRUE	FALSE	TRUE	FALSE	FALSE	TRUE	FALSE	TRUE
5	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE	FALSE	TRUE
6	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE	TRUE
7	FALSE	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE

Number of Matching Latencies

	5	6	4	5	4	7	6	6	4	3

Latencies Weighted According to Level of Variability

	21%	21%	21%	0%	0%	21%	21%	21%	21%	21%	21%
	7%	7%	0%	0%	0%	7%	7%	7%	0%	7%	7%
	15%	15%	0%	15%	15%	15%	15%	15%	0%	0%	15%
	26%	0%	26%	26%	0%	26%	0%	0%	26%	0%	26%
	12%	12%	12%	12%	12%	12%	12%	12%	0%	0%	12%
	0%	8%	8%	8%	8%	8%	8%	8%	8%	0%	8%
	0%	11%	0%	11%	11%	11%	11%	11%	11%	11%	11%

Sum of the Weights for Each Test

	81%	74%	67%	72%	46%	100%	74%	74%	66%	39%	100%

Figure 14 – Results of Verification Algorithm on Initial Data.

5. TESTING:

5.1 User Trails:

5.1.1 Reasons for a User Trial

A user trial is necessary to determine the False Accept Rates and False Reject Rates of the biometric system. These two measures are some of the most important measures of the effectiveness of an identity verification system. Since these figures cannot be determined accurately and conclusively using simulations a trial must be used to determine the effectiveness of the program with real users by performing some type of user trial.

5.1.2 Description of the User Trial

The user trial involves the participation of at least ten users of varying typing ability. Each user will be given a copy of the program and will be required to use the program and enter at least 50 valid logins. The User Survey gathers information about the user's typing ability and various other associated data. This can help determine why a user has certain characteristics, for instance a beginner typist may have high variability in their signature profiles for the login process. Once at least 10 users have completed the trials and attempted at least fifty logins the information gathered can be analyzed and the False Reject Rate i.e., the rate of rejection for valid users, can be determined.

The next stage of the user trial is to have at least two users at random pick two other users to impersonate and try to login as an impostor. Analysis of these results will determine the False Acceptance Rate, which is the rate at which an invalid user is able to login successfully.

5.2 Test Cases:

A **Test Case** is a set of actions executed to verify a particular feature or functionality of your software application. A Test Case contains test steps, test data, precondition, postcondition developed for specific test scenario to verify any requirement. The test case includes specific variables or conditions, using which a testing engineer can compare expected and actual results to determine whether a software product is functioning as per the requirements of the customer. Test Case acts as the starting point for the test execution, and after applying a set of input values, the application has a definitive outcome and leaves the system at some end point or also known as execution postcondition. The test cases for this system will be based on the keystroke dynamics verification and the password manager functions like adding, deleting, and updating.

Below, in the next page is the table of some of the test cases defined for the Keystroke biometrics identification system with password manager:

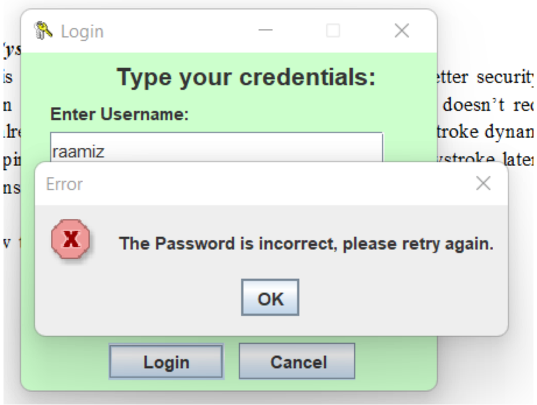
TEST CASES	INPUT	EXPECTED OUTPUT	ACTUAL OUTPUT	SUMMARY/ RESULT
TC1	Provide user as Raamiz and password as something(for not registered account).	Invalid Username and password	Invalid username and password	Verifying whether the user is registered or not.
TC2	Provide user as raamiz and password as the password which the user gave and biometric match.	Successfully logged in	Valid username and password with valid biometrics	Verifying whether the user is registered or not.
TC3	Provide user as raamiz and password as the password which the user gave but biometric did not match.	Invalid Biometrics	Invalid Biometrics	Keystroke Biometrics verification.

TEST CASES	INPUT	EXPECTED OUTPUT	ACTUAL OUTPUT	SUMMARY/ RESULT
TC4	Logged in to password manager user loads the data from database.	Passwords loaded from database into the table	Passwords loaded from database into the table	Password manager operation
TC5	Logged in to password manager user loads the data from database but no data to show.	No data available to display	No data available to display	Password manager operation
TC6	Logged in to password manager user performs adding, deleting, or updating the data from database.	Operation Performed.	Operation Performed.	Password manager operation

TEST CASES	INPUT	EXPECTED OUTPUT	ACTUAL OUTPUT	SUMMARY/ RESULT
TC7	Logged in to password manager user performs deleting, or updating the data from database but domain does not match	Domain could not be found for updating or deleting the data.	Domain could not be found for updating or deleting the data.	Password manager operation
TC8	Logged in to password manager users clicks on logout.	Back to login page	Back to login page	Password manager operation, back to keystroke biometric identification system
TC9	User wants to exit from the system and clicks on the exit button.	Exit Message shown and Application is exited.	Exit Message shown and Application is exited.	Keystroke biometrics identification and password manager exit operation

5.3 Test Results:

Below are some of the test case results which are shown with screenshots:

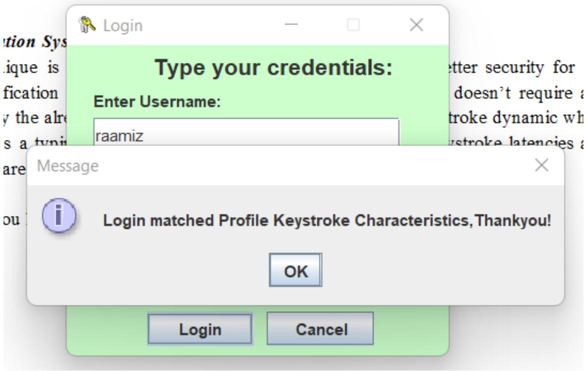


Type your credentials:

Enter Username: raamiz

The Password is incorrect, please retry again.

Figure 15 – Testcase-1

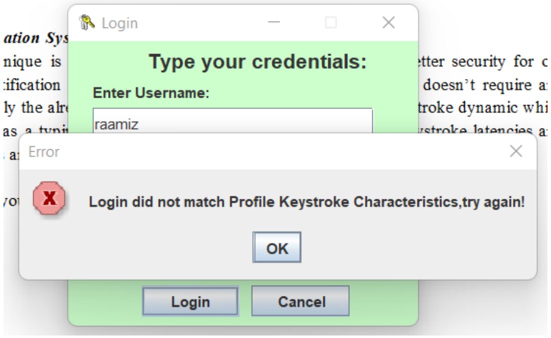


Type your credentials:

Enter Username: raamiz

Login matched Profile Keystroke Characteristics,Thankyou!

Figure 16 – Testcase-2

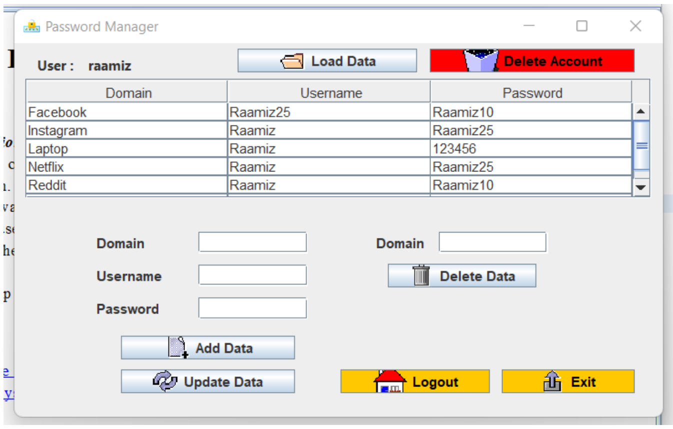


Type your credentials:

Enter Username: raamiz

Login did not match Profile Keystroke Characteristics,try again!

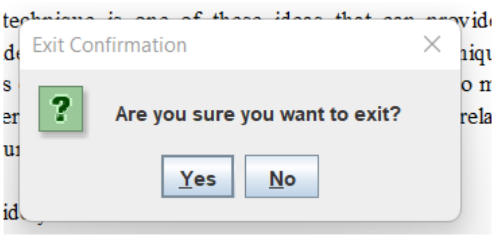
Figure 17 – Testcase-3



User : raamiz

Domain	Username	Password
Facebook	Raamiz25	Raamiz10
Instagram	Raamiz	Raamiz25
Laptop	Raamiz	123456
Netflix	Raamiz	Raamiz25
Reddit	Raamiz	Raamiz10

Figure 18 – Testcase-4



Are you sure you want to exit?

Figure 19 – Testcase-9

6. RESULTS:

The program as described in section three has had varying levels of success. Below in figure 19 is a summary of the login attempts by the trial users.

User Number	Number of Logins	Successful Logins	Username Accepted	User Password Accepted	Common Phrase Accepted
1	51	53%	69%	63%	35%
2	51	76%	80%	43%	69%
3	52	83%	54%	75%	90%
4	60	82%	55%	80%	78%
5	51	80%	78%	67%	73%
6	51	76%	37%	88%	78%
7	53	92%	91%	68%	92%
8	51	67%	57%	80%	49%
9	53	91%	51%	92%	85%
10	51	67%	47%	43%	96%
11	54	72%	76%	72%	56%
	User Average:	76%	63%	70%	73%

Figure 20 – Summary of logins by a user.

This table indicates that the False Reject Rate is in the order of 24% as the average percentage of successful logins for the trial users was only 76%. This is a disappointing result considering most of the previous studies discussed in the Literature Review has been able to achieve False Reject Rates below 10%. Part of the reason for these poor results is that users who have poor typing skills were included in the user trial.

The second phase of the user trial program was to get two users to pose as impostors and try to logon as other users. One point to note is that the impostors did not observe other users typing into the program. By observing a user type the chances of making a successful login as an impostor increase significantly as the impostor now has an idea of what the approximate. Latencies are and this is especially true for slower typists because is easier to distinguish their individual keystrokes. After each user to be impersonated had finished their 50 logins, the two impostors then tried to log in another 50 times using the original signature profiles for the impersonated users. For these four situations almost, all attempts failed except for the username when user two attempted to login as user 9. Overall, this gives a False Acceptance Rate of 0%. While these results seem quite impressive, they are not conclusive. More impostors were needed, and all users should have been tried to be impersonated by all the other users. This is the only way to have real confidence in the results. Also, many more users would be needed in the user trials of the program. Unfortunately, the number of users required to prove confidence in the False Accepts Rate beyond reasonable doubt would be quite large and beyond the means and scope of this project.

User Number	Number of Logins	Successful Logins	Username Accepted	User Password Accepted	Common Phrase Accepted
User 1 Trying to log on as User 3	52	0%	0%	0%	0%
User 1 Trying to log on as User 4	60	2%	4%	2%	1%
User 2 Trying to log on as User 5	51	0%	0%	0%	0%
User 2 Trying to log on as User 9	54	1%	4%	1%	1%

Figure 21 – Summary of logins by imposters.

To show how distinct the user's profile signatures are, figure 14 shows the comparison of the reference profile of the standard phrase "the brown fox" between all users in the trial. While most of them look very similar all profiles have a unique pattern. The main problem is trying to distinguish between them.

The main area where the program does not function well is when there is high variability. Outlying logins that deviate markedly from the mean tend to skew the reference profile rendering logins that would otherwise be accepted to be invalidated or vice-versa. One reason for this is that the initial login numbers 5 to 10 have a high variability as the user gets accustomed to typing in the various words and phrases. This was partially help by ignoring the first five logins, but it would also be useful to have an additional five more logins for the initial generation of the template. This is because during this phase there is no verification of the input so that the signature can adjust itself more easily, rather than when verification is used, and change is incremented much more slowly.

Another method to reduce the variability would be to remove the outlying latencies periodically, especially if they are from earlier logins so that the signature profile may more closely match that of the user.

This program as well as many other programs and algorithms respond best to typists that have a reasonable level of skill and consistency. Some of the users in the trial especially user ten had poor typing skills.

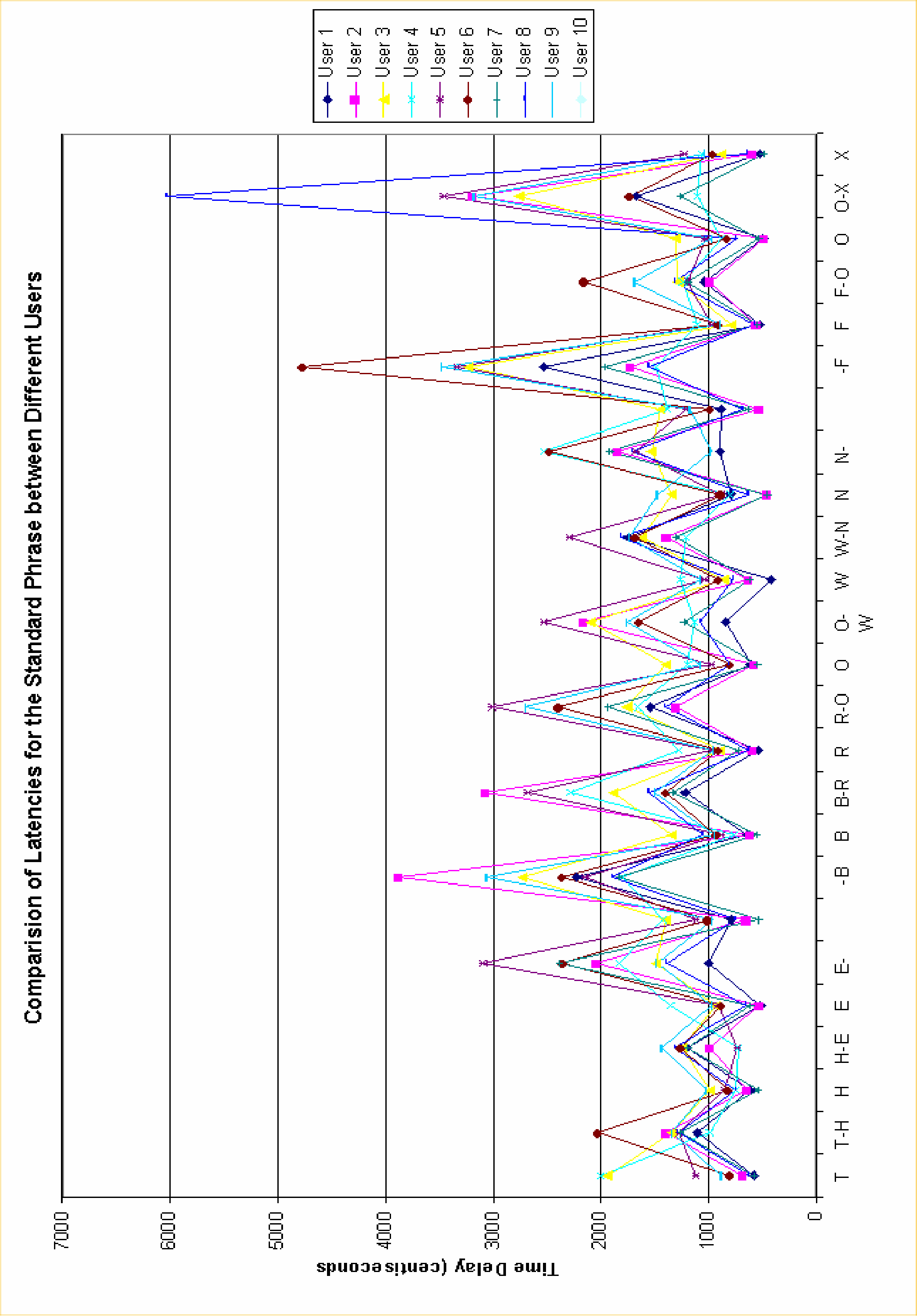


Figure 22: Comparison of Users Signature Profile for “the brown fox” standard Phrase.

7. CONCLUSION

The program developed for the capture and analysis of the typing dynamics was not overly complicated or fancy. It was able to do its objectives of logging data and to verify login attempts by using statistical analysis. It was also able to continuously adapt to changes in the users’ typing patterns. Another important objective was that the response time was virtually immediate.

The results from the user trial showed promise. While they were not as good in terms of False Reject Rates as contemporary research papers have achieved, it is still no less significant. The 24% False Reject Rate was quite high, and this may have been due to very restrictive algorithms. In part, due to this restrictive algorithm, the False Accept Rate proved to be 0% however, as mentioned earlier this is not conclusive as the tests for this were limited due to the small user trial size. What is required is a larger and more in-depth study that has more users so that the validity of the False Reject Rate can be proved beyond reasonable doubt.

Overall, the project has been a small success; the program has shown that using typing dynamics analysis is a useful method in providing identity authentication. If it is simply viewed as strengthening the login process it is has shown to be quite adequate in this regard. While the actual program was not as effective as some programs and algorithms that have been published it manages to show some ability and demonstrates the vast potential using the technology.

8. FUTURE SCOPE

In addition to the statistical analysis approach used for this project other methods that could be used are neural networks and fuzzy logic. Ideally, this project should have had a larger user trial to properly validate the findings. This could be considering for future works. Other areas that could be considered for future work is the effect that different types of keyboards, times of day and other variables have on the ability for a user to type a consistently recognizable keystroke pattern. One of the more interesting and advanced areas of typing dynamics biometrics is that of dynamic analysis. Using this approach, a user may constantly be verified throughout their session at the computer, so that a person cannot just sit down in front of a computer that has been left unattended and gain access. This area seems to be the logical conclusion for the technology and may prove one day to be very widespread. And this technology can be used to any other application also not only password manager but also any other application which use the traditional authentication technique.

9. BIBILOGRAPHY

[1] Haider, S., Abbas, A., Zaidi, A., A Multi-Technique Approach for User Identification through Keystroke Dynamics, IEEE International Conference of Systems, Man and Cybernetics, Vol 2, p1336-1341, 2000.

[2] Monrose, F., Rubin, A., Authentication via Keystroke Dynamics, Proceedings of the 4th ACM Conference on Computer and Communications Security, p 48-56, April 1997.

[3] Gollman, Dieter, Computer Security, Chichester ; New York : Wiley, c1999, QA76.9.A25 G65 1999.

[4] Allen, J. D. (2010). An analysis of pressure-based keystroke dynamics algorithms, Master's thesis,Southern Methodist University, Dallas, TX.Araujo, L., Sucupira, L.H.R., J., Lizarraga, M., Ling, L. & Yabu-Uti, J. (2005).

[5] Userauthentication through typing biometrics features, IEEE Transactions on SignalProcessing 53(2 Part 2): 851–855.Azevedo, G., Cavalcanti, G., Carvalho Filho, E. & Recife-PE, B. (2007).

[6] An approach to featureselection for keystroke dynamics systems based on pso and feature weighting,Evolutionary Computation, 2007. CEC 2007. IEEE Congress on.Balagani, K. S., Phoha, V. V., Ray, A. & Phoha, S. (2011).

[7] On the discriminability of keystrokefeature vectors used in fixed text keystroke authentication, Pattern Recognition Letters32(7): 1070 – 1080.Bartmann, D., Bakdi, I. & Achatz, M. (2007).

[8] On the design of an authentication system basedon keystroke dynamics using a predefined input text, Techniques and Applications forAdvanced Information Privacy and Security: Emerging Organizational, Ethical, and HumanIssues 1(2): 149.Bello, L., Bertacchini, M., Benitez, C., Carlos, J., Pizzoni & Cipriano, M. (2010).

[9] Collectionand publication of a fixed text keystroke dynamics dataset, XVI Congreso Argentinode Ciencias de la Computacion (CACIC 2010).Bergadano, F., Gunetti, D. & Picardi, C. (2002).

[10] User authentication throughkeystroke dynamics, ACM Transactions on Information and System Security (TISSEC)5(4): 367–397.Bleha, S. & Obaidat, M. (1991).