

SMART PIN SECURITY IN ATM TRANSACTION ON GRAPHICAL PASSWORD INTERCHANGE

Submitted in partial fulfillment of the requirements for the award of

Bachelor of Computer Science

By

SWATHI.M

REG.NO. 40290099



**DEPARTMENT OF COMPUTER SCIENCE
SCHOOL OF SCIENCE AND HUMANITIES**

SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**

Accredited with Grade "A" by NAAC | 12B Status

By UGC | Approved by AICTE

**JEPPIAR NAGAR, RAJIV GANDHI SALAI,
CHENNAI 600119**

MAY 2023



SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)

Accredited "A" Grade by NAAC | 12B Status by UGC | Approved by

www.sathyabama.ac.in

DEPARTMENT OF COMPUTER SCIENCE

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafied work of the **SWATHI.M**
0290099) who carried out the project entitled "**SMART PIN SECURITY IN ATM**
TRANSACTION ON GRAPHICAL PASSWORD INTERCHANGE " under our
supervision from FEBRAUARY 2023 TO MAY 2023.

S. Rajeshwary.
INTERNAL GUIDE

MS. RAJESHWARY.S, M.Sc., Assistant Professor

Rekha
DEAN AND HEAD OF THE DEPARTMENT
Dr.REKHA CHAKRAVARTHI M.E., Ph.D.

Submitted for Viva voce examination held on 08/05/2023

6/5/2023
Internal Examiner

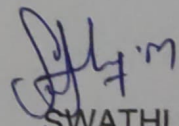
B. Dishant
External Examiner

DECLARATION

I am, **SWATHI.M (40290099)** hereby declare that the Project Report entitled "**SMART PIN SECURITY IN ATM TRANSACTION ON GRAPHICAL PASSWORD INTERCHANGE**" done by me under the guidance of **Ms.Rajeshwary.S, M.Sc.,** Department of Computer Science at **SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY**, Jeppiar Nagar, Rajiv Gandhi Salai, Chennai-600119, is submitted in partial fulfillment of the requirements for the award of Master of Science Degree in Computer Science.

DATE:

PLACE: CHENNAI


SWATHI.M

SIGNATURE OF THE CANDIDATE

TABLE OF CONTENTS

CHAPTER	TITLE	PAGE NO
	ABSTRACT	VI
	LIST OF FIGURES	V
1	INTRODUCTION	1
	1.1 OBJECTIVES	1
	1.2 ATM SECURITY	1
	1.3 HYBRID PIN KEYPAD	2
	1.4 AUTHENTICATION	2
	1.5 RECALL-BASED GRAPHICAL PASSWORD VERIFICATION	3
	1.6 LITERATURE SURVEY	4
2	AIM AND SCOPE OF THE PROPOSED SYSTEM	6
	2.1 AIM	6
	2.2 SCOPE	7
	2.3 EXISTING SYSTEM	7
3	SYSTEM CONFIGURATION	8
	3.1 SYSTEM ARCHITECTURE	8
	3.2 HARDWARE REQUIREMENTS	9
	3.3 SOFTWARE REQUIREMENTS	9
4	EXPERIMENTS, METHODS AND ALGORITHM	10
	4.1 METHODS USED	10
	4.1.1 Phase I: REGISTRATION PHASE	10
	4.1.2 Phase II: LOGIN PHASE	11
	4.2 ALGORITHM USED	15
5	RESULT AND DISCUSSION	17
	5.1 DISCUSSION	17
	5.2 RESULT	17
	5.3 TEST SCREENS	19
6	CONCLUSION	21
	6.1 APPENDIX	22
	6.2 SOURCE CODE	22
	6.3 SCREEN SHOT	45
7	BOOK REFERENCE	54

LIST OF FIGURES

FIGURE NO.	FIGURE NAME.
3.1.	System Architecture
4.1.	Flow Diagram of the given first phase user account credentials
4.2.	Flow Diagram of the Login Phase of ATM Transaction System
4.3.	Access Phase System using URL System

ABSTRACT

The vital issue in data security is client confirmation. There are numerous confirmation strategies are printed, graphical or biometric passwords etc. The content based watchword is effectively speculated by the assailant over to adjacent bear; assailants watched specifically or observe a few outside gadgets.. The nearby people watching pin can be made confusion where the numbers appeared as the digital value. Here the shoulder based resistant attack is reduced where the hybrid pin images will randomly shuffle as they cannot be easily known. The efficient pin based security can be added from the behind person where they will not able to see the details of the pin. And as of the technology developed the pin can be used in the digitalized display itself. The system developed as software and can be executed out in all ATM .

CHAPTER 1

INTRODUCTION

ATM pin entry system are not more secure since anyone can view the pin number from back side. Thus the security is maintained for password recovery, digital pin system and authorization access system is added. The system can be added for user security and the system can be well developed for future enhancement. This can be used for the authentication in public places (ATM, Banks and so on). This resists against many attacks and eavesdropping problems.

1.1 OBJECTIVES

Textual passwords containing alphanumeric words are strong enough to resist against brute force attacks. Graphical password is one of the alternative solutions to alphanumeric passwords because it is easy to remember. Existing methods use an authentication system called PassMatrix, based on graphical passwords. Proposed a method that uses hybrid keypad and biometric authentication for overcoming shoulder surfing attacks. The unknown user attacks and misused ATM cards are avoided by webpage-based user access. When the ATM card is known lost or misplaced then the user can make their own access.

1.2 ATM SECURITY

These are getting prevalent, where clients can use additional individual gadgets, such as mobile phones, to perform ATM exchanges. Shoulder-surfing attacks, also known as perception assaults, are the most common risk for ATM confirmation. In this case, the aggressor basically views the screen strategy of the PIN by the authorized client to get hold of the mystery data. Credit card and charge card frauds due to character robberies are expanding each year. Additionally, there are scamming strategies utilizing

fake terminals, credit card cloning, and inaccessible transfer which make the handle of client security harder. The attacker can be standing in line behind the authenticating individual and looking at the Pin section and execute a shoulder-surfing or perception attack. The attacker may too introduce a little camera on the best surface of the ATM terminal to record Stick sections of clients at the point-of-service.

1.3 HYBRID PIN KEYPAD

Bear surfers can too see at a remove utilizing spyware, binoculars or other visionary equipment. Presently a day's numbers of graphical secret word methods are used to secure from the bear surfing assailants. In spite of a later study report on innovative world, a security team measures the proportion is 80% of employees' watchword is cracked inside organization in 30 seconds. Unfortunately, existing graphical secret word plans are not proficient to specialized aggressor. The proposed work Hybrid Pin keypad method is exceptionally straightforward and more secured to the authenticated user. The client will make a smart card and the scanner to make login to keep money exchange effectively and highly secured. This proposed hybrid pin keypad technique has a visual keypad for safe bear surfers. This visual keypad is planned to computerized number organize.

1.4 AUTHENTICATION

Authentication can be defined as verifying the validity of a user by using at least one form of the identification methods. To grant access to the system, the users' identity should be verified by determining the following factors.

- i) Knowledge-based factors: It is defined as what the user knows. Some of them are any forms of a password, personal identification number, and the project to the secret questions and many more.
- ii) Possession-based factors: It is defined as what the user has. Some of them are an identification card, security token, device token or any unique hardware identifier.

iii) Inherence-based factors: It is defined based on what the user is or how he does. Some of the physiological factors are fingerprint; iris and DNA patterns and some of the behavioral factors are biometric identifiers, signatures, voice and face.

Authentication can be a combination of the above. The types of authentication categories include:

- Single-factor authentication: It makes use of one factor to authenticate the user trying to login to the system. It is more prone to different cyber-attacks.
- Two-factor authentication: It combines any two authentication factors to increase the level of security in the system. A practical example of this implementation is the real – time banking login where some banks generate a one-time password (OTP) while the user logs in by typing the correct password. Only if the password entered is valid the OTP gets generated, and if the user enters the generated number from his device correct, he gains access to the system.
- Multi-factor authentication:- It combines many authentication mechanisms to form a layered approach. The plethora of functionalities offered by multi-factor authentication includes protection from intrusion, enhancement of security, and reliable false proof system. The idea is to add three factors of authentication by entering the correct password, verifying the device, and identifying the users typing pattern.

1.5 RECALL-BASED GRAPHICAL PASSWORD VERIFICATION

A range of approaches for recall-based graphical password authentication have been evaluated using measures including resilience to forgeries, memorability, user acceptance, error rates and time to enroll. Recall-based authentication can be divided in two categories. Exact-match approaches assume that during authentication, a user produces exactly the same drawing provided during enrollment.

1.6 LITERATURE SURVEY

1. TITLE IMAGE BASED AUTHENTICATION USING ILLUSION PIN FOR SHOULDER SURFING ATTACK

AUTHORS: K.Divyapriya, Dr.P.Prabhu

YEAR: 2021

Shoulder surfing attack is the direct observation of user from far distance by hacker. Traditional methods use personal identification number (pin) consists of a sequence of digits for authentication. This method is used for Digital Authentication of touch screen devices. The applications of touch screen devices include ATM machines, Smart phones and Kiosk. Shoulder surfing attack suffers from various issues, Challenges and limitations like security and privacy. There are various algorithms and techniques have been proposed in the literature to overcome these difficulties and still needs improvement. Hence in this work a novel algorithm using illusion pin with hybrid images for shoulder surfing attack authentication scheme has been proposed. This proposed method using Illusion-pin (I-pin) blends of two keypads with different ordering digits using hybrid images. The user keypads is shuffled in every authentication attempt. This method is used to restrict the shoulder surfing attack by implementing this visibility algorithm. Hence hackers are unable to recognize or learning the user pin which provides more security and authentication.

2. TITLE DIRECTIONAL BASED GRAPHICAL AUTHENTICATION METHOD WITH SHOULDER SURFING RESISTANT

AUTHOR: Anis Shobirin Abdullah Sani, Noor Ashitah Abu Othman ; Muhammad Akmal Abdul Rahman ; _

YEAR: 2020

Authentication has been advanced by implying advancement in security such as cryptography to protect password against data leaking and sniffing; captcha to prevent

robot system; Secure Sockets Layer designed to protect users against phishing website and many other solution invented for different types of vulnerabilities. Although there are other methods suggested for preventing shoulder surfing (i.e. exposure of password to nearby observing-person), the techniques still have some drawback. Some of the proposed techniques also required high computational process with high range of possible answer. This paper proposed a shoulder-surfing-proof graphical based authentication with direction scheme that combines technique adapted from Passfaces scheme and a selection of direction. User is required to choose four faces and one direction to serve as the secret authentication.

CHAPTER 2

AIM AND SCOPE OF THE PRESENT INVESTIGATION

2.1 AIM OF THE PROPOSED SYSTEM

The Using at least one of the identification methods, authentication is the process of confirming the legitimacy of a user. Verifying the users' identities is necessary before granting them access to the system.

i) Factors based on knowledge: These are things the user is aware of. Some of these include various passwords, PINs, the and the project to secret riddles, and many others.

ii) Possession-based factors: These are determined by the user's possessions. A security token, device token, or any other special hardware identifier is one of them.

iii) Inherence-based factors: It is determined by the characteristics of the user or his actions. Biometric identifiers, signatures, voices, and faces are some examples of behavioral elements. Physiological factors include fingerprint, iris, and DNA patterns.

Any of the aforementioned methods may be used for authentication. The various categories of authentication include:

- Single-factor authentication: It uses just one factor to confirm that the person seeking to enter into the system is who they say they are. It is more vulnerable to various cyber attacks.
- Two-factor authentication (two-factor authentication) combines any two authentication elements to boost system security. Real-time banking logins, where some banks generate one-time passwords (OTPs) as a user enters the right password, serve as a concrete illustration of its application. The OTP can only be generated if the password entered is accurate, and if the user enters the generated number correctly from his device, he is granted access to the system.

2.2 SCOPE OF THE PROPOSED SYSTEM

Proposed a method which uses hybrid keypad and a unknown access identification for authentication purpose. User has to register using the biometric method as the project ll as username and PIN method. At the login time, the user can use any one of these methods for login. If the username and PIN, it makes the keypad to look in two ways when typing the PIN. The keypad will look like a normal one for the user but at the same time, it will look different for the shoulder – surfing person. This keypad uses the technique of hybrid images to blend two keypads with different digit orderings. This is designed in such a way that it looks different to the user and the others who is observing from a considerable distance.

ADVANTAGES OF THE PROPOSED SYSTEM

- The shoulder surfer can't predict the password either by brute force attack or by guessing attack.
- The PIN can't be guessed even if it's recorded through surveillance cameras.
- Biometric provides unique authentication facility.

2.3 EXISTING SYSTEM

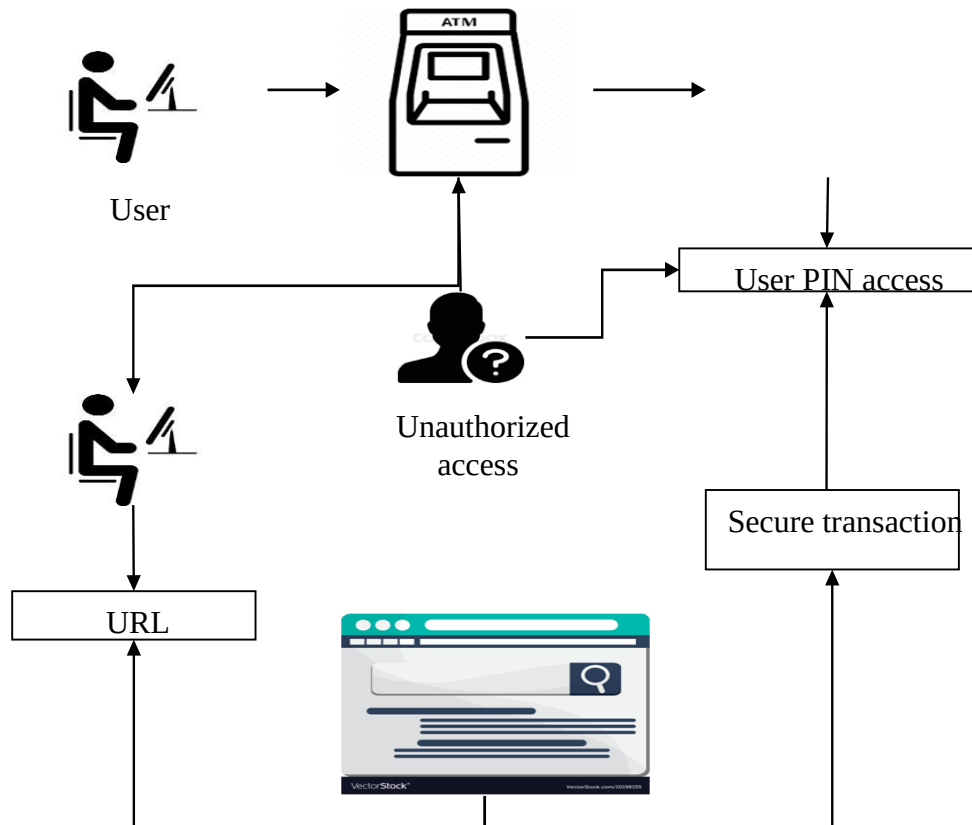
Authentication based on passwords is used largely in applications for computer security and privacy. However, human actions such as choosing bad passwords and inputting passwords in an insecure way are regarded as “the weakest link” in the authentication chain. Rather than arbitrary alphanumeric strings, users tend to choose passwords either short or meaningful for easy memorization.

DISADVANTAGES

- Assumed that the server and the client devices in our authentication system are trustworthy.
- Assumed that the surveillance cameras that are not under proper management

CHAPTER 3

3.1 SYSTEM ARCHITECTURE



3.2 HARDWARE REQUIREMENTS

- Processor : Dual core processor 2.6.0 GHz
- RAM : 1GB
- Hard disk : 160 GB
- Compact Disk : 650 MB
- Keyboard : Standard keyboard
- Monitor : 15 inch color monitor

3.3 SOFTWARE REQUIREMENTS:

- Front End : JAVA
- Back End : My SQL
- Platform : Windows 8

CHAPTER 4

EXPERIMENTS, METHODS AND ALGORITHM

4.1 METHODS USED

The system works with three stages

- Registration Phase
- Random Pin phase
- Access Phase

4.1.1Phase I: Registration Phase

In this stage the authorized client is attending to make an account for get to the keeping money application utilizing the special username and secret word. This enrollment module is putting away the user profile data and account points of interest. It creates report for registered client data and keeping money exchange prepare. Here the user credited details will be added with the smart scanner and the card which they will be read to this process. The registration process added with the email transferring system where the account number and the pin number will be generated to the user email where they can access their pin using the system.

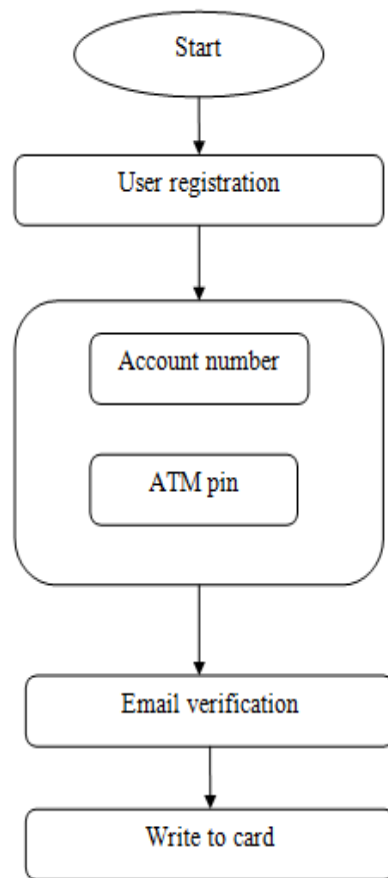


Fig4.1 Flow diagram of the given first phase user account credentials

4.1.2 Phase II: Login Phase

Here the login phase will be given with an smart card swiping system where their account number and the pin are already written with the card during the registration process. The thing the user has to use here is that they have to get the pin to the system.

4.1.2.1Hybrid Pin images:

In this module implement visibility algorithm to create Hybrid PIN. In this PIN pattern is designed to the digital number format, the visibility algorithm analyzed the human visual capacity and generate embedded PIN images that can be seen only on-screen user. The shoulder surfers view only the digit 8. The shoulder surfer may not able to view the Hybrid PIN Keypad. Hybrid PIN keypad uses the technique is to combine two keypads with a different number patterns. The authorized on-screen user is to seeing one keypad pattern to enter the PIN digit, while the shoulder surfers is watching at the device from a long distance is seeing on another keypad pattern. This keypad is shuffled in every attempt.

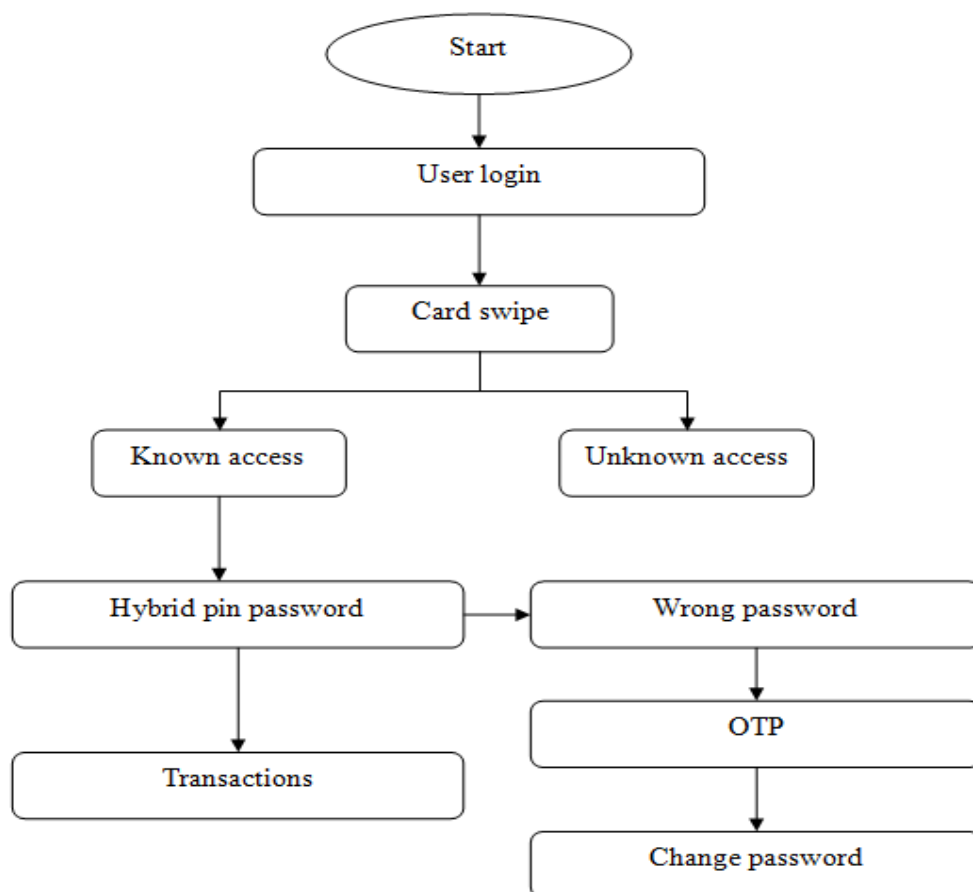


Fig4.2 Flow diagram of the Login phase of ATM transactions system

4.1.2.2 Random Shuffling Keypad

The Hybrid Pin design is outlined to 4×3 lattice format. Each client has been log out the application, the Pin pattern is re-shuffled. This rearranging design is against for shoulder surfers. When start the login handle, the authorized client pass the login ask for getting to the keeping money administrations, the system visualize a Hybrid Pin Keypad. It is outlined to 4×3 matrix format appears on Figure: 4. each client exchange points of interest are stored in database and each person client Pin is put away in a server organize system. This standard of the pin system is progressing the additional security. The client has entered the precise Pin, after access the managing an account administrations (deposit, withdraw, transaction). The pin may be presented in an illusion where all the numbers are get shuffled at each login session.

Visibility challenges

The distance-as-filtering speculation state is mimicking the different seeing position. Each client and spectator is looking directly. The on-screen client as it the project re distinguished the Stick digit and the spectator seeing position is spoken to the Stick design is completely distinctive by assailant seeing separate. The visual angle is completely shifted. The shoulder surfer may not be able to see the Hybrid Pin design. The visual data speaks to the discernment in changed for human perception. The visibility esteem is portrayed the eyewitness is able to marginally note the Pin design. This visual expectation is shifting for different eyewitnesses. The spectator is anticipating the mistakenly PIN design that's planned to human obvious capabilities

Wrong pin clearance

Since, the password may be in a confused manner the user will change the password where they can give wrong pin. If the user gives more than 3 times then the card will be blocked where they cannot take money from the card they have to visit the nearby bank. In order to clear these issues the OTP generation system is placed where they can change their password within their ideal time. When their card gets blocked they can

made access of to the email based OTP system and they can generate a new password system. When the exchange is completed and logout the screen, the PIN design is re-shuffled. The witticism of rearranging strategy is to no clue for shoulder surfers. After logout or press the reset button. The Numeric Pin design is rearranged.

4.2.3 Phase III: Access Phase

The third party may access the card and sometimes due to our inability the project may too give card to get cash. But the serious disadvantage here is that they will access the card where they will know the pin. To secure the pin from unauthorized system a the project b server based authentication system is made.

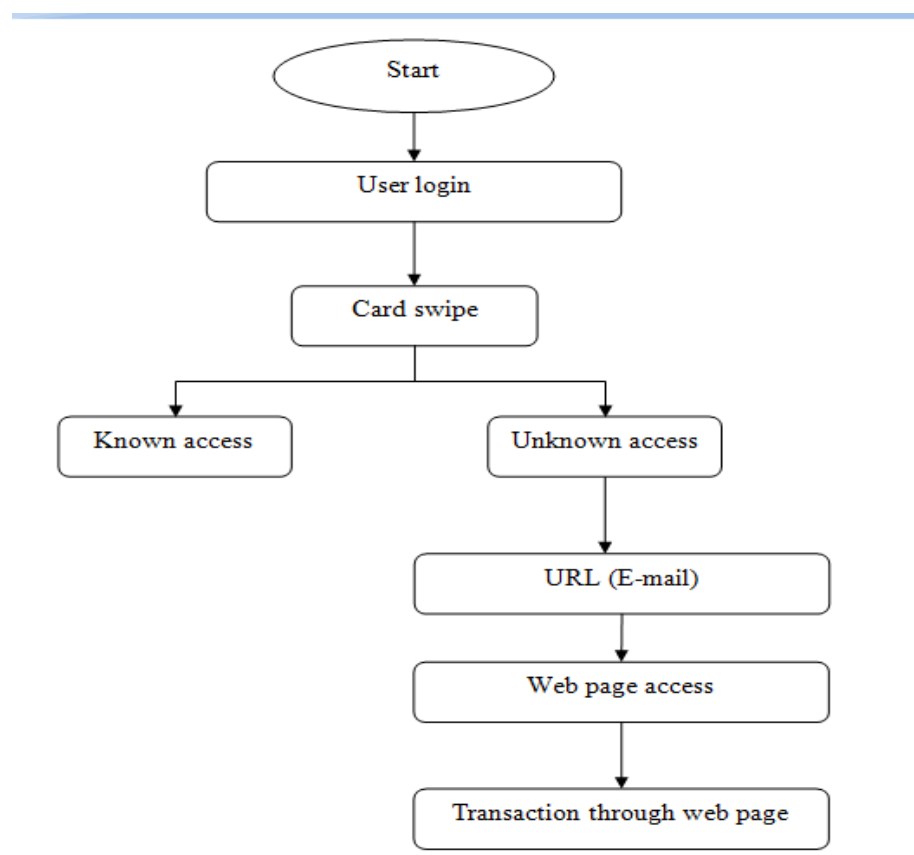


Fig 4.3 Access Phase system using URL system

When the unknown access is made then the user will be given with the system access. The user will get an URL link where they can access using the the project b page system. The URL can be made with the username and password login system for security reasons. The access can be made and the pin number can be entered by the user where they can access it. Here if the user knows that the access is made by the unknown user then the access will be blocked where the user only can open the system.

4.2 ALGORITHM USED

HYBRID PIN ALGORITHM

IllusionPIN is a PIN-based authentication scheme for touchscreen devices which offers shoulder-surfing resistance. The design of IllusionPIN is based on the simple observation that the user is viewing the screen of her device from a smaller distance than a shoulder-surfer. Based on this, the core idea of IllusionPIN is to make the keypad on the touch screen to be interpreted with a different digit ordering when the viewing distance is adequately large. This way, when the shoulder surfer is standing far enough, he is viewing the keypad as being different from the one that the user is utilizing for her authentication, and consequently he is unable to extract the user's PIN.

As the project can see, the gain distribution of the CSF favors the perceived spectrum of I , h and u and as a result I is interpreted as being Iu . the project depict the same perceived spectrums for a bigger viewing distance. As the project can see, the perceived spectrums are stretched to higher perceived spatial frequency magnitudes and the CSF favors.

Require: hybrid keypad I , shoulder-surfer's keypad I_s , DAF filter,

viewing position N ,

visibility index threshold value v_{th}

1: $I_{sp} \leftarrow$ calculate 2D perceived spectrum of I given N

- 2: $I_{s,sp} \leftarrow$ calculate 2D perceived spectrum of I_s given N
- 3: $IDAF_{sp} \leftarrow$ apply DAF filter to I_{sp}
- 4: $IDAF_{s,sp} \leftarrow$ apply DAF filter to $I_{s,sp}$
- 5: $IDAF \leftarrow$ transform I
- DAF sp to spatial domain
- 6: $IDAF_s \leftarrow$ transform $IDAF_{s,sp}$ to spatial domain
- 7: $Buttons(IDAF) \leftarrow$ segment the buttons of $IDAF$
- 8: $Buttons(IDAF_s) \leftarrow$ segment the buttons of $IDAF_s$
- 9: $v \leftarrow \text{mean}(\text{MSSIM}(Buttons(IDAF), Buttons(IDAF_s)))$
- 10: if $v \geq v_{th}$ then
- 11: return No
- 12: else
- 13: return Yes

CHAPTER 5

RESULT AND DISCUSSION

5.1 DISCUSSION

For the design of IllusionPIN the project follow the principle of obscurity, since the shoulder-surfer's keypad is obscures the user's keypad. The project could use any image in the place of Is, but the project decided to always use the image of keypad because this way I, h and u are visually aligned. This means that then digits in I s overlap, and the less dominant keypad is perceived as noise, providing more clear interpretations of the hybrid keypad. The project also apply the principle of alteration by shuffling the user's keypad in every authentication attempt, or after every digit entry. Otherwise, it would be enough for a shoulder-surfer to memorize just the spatial arrangement of the pressed digits.. Out of all the possible digit orderings that Is may have, the project choose the regular digit ordering. The reason is that this is the ordering that the project expect an attacker to be the most familiar with, and as a consequence, to have the tendency to recognize

5.2 RESULTS

The proposed Hybrid Pin Keypad method is actualized in java. Once total the effective login handle, client can see the Hybrid Pin design framework. Here digital numbers are appeared to the client. Each advanced number has one numeric digit in their structure. This will only identified by the individuals who display to approach to the screen. This numeric design planned to ensure the clients Pin from shoulder surfing attack. The security of the experimental results may quiet low and it takes higher time to get the login session in the existing system. The user existing system might be given with different processing capacity.

Method	Time in seconds	
	Registration	Login
Pass point	47.86	42.26
Graphical textual	38.66	39.18
Hybrid pin	31.13	28.18

Table 1 denotes the time taken for the execution of the given system

The client may get to spend more time for existing procedure Pass-points [3]. The pass point procedures take 47.86 seconds and the proposed Hybrid Pin keypad takes 431.13 seconds for registration/Practice Stage. At that point the Pass point takes 42.26 seconds and Hybrid Pin keypad takes 28.18 seconds for Login Stage. The registration and the login system of graphical textual authentication are also high compared with the proposed solution.

5.1.1 UNIT TESTING

The first test in the development process is the unit test. The source code is normally divided into modules, which in turn are divided into smaller units called units. These units have specific behavior. The test done on these units of code is called unit test. Unit test depends upon the language on which the project is developed. Unit tests ensure that each unique path of the project performs accurately to the documented specifications and contains clearly defined inputs and expected results. Functional and reliability test in an Engineering environment producing tests for the behavior of components (nodes and vertices) of a product to ensure their correct behavior prior to system integration.

5.1.2 SYSTEM TESTING

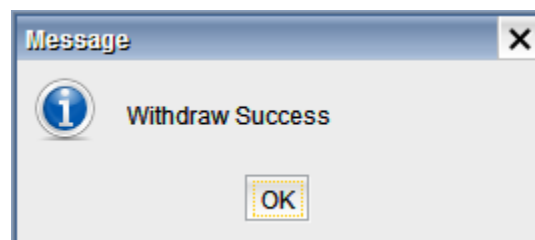
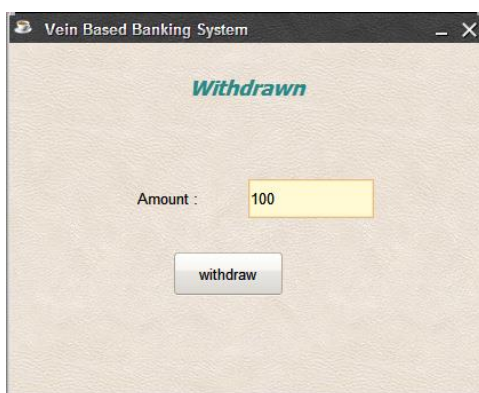
Several modules constitute a project. If the project is long-term project, several developers write the modules. Once all the modules are integrated, several errors may arise. The testing done at this stage is called system test. System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points. Testing a specific hardware/software installation. This is typically performed on a COTS (commercial off the shelf) system or any other system comprised of disparate parts where custom configurations and/or unique installations are the norm.

5.1.3 INTEGRATION TESTING

Testing in which modules are combined and tested as a group. Modules are typically code modules, individual applications, source and destination applications on a network, etc. Integration Testing follows unit testing and precedes system testing. Testing after the product is code complete. Betas are often widely distributed or even distributed to the public at large in hopes that they will buy the final product when it is release.

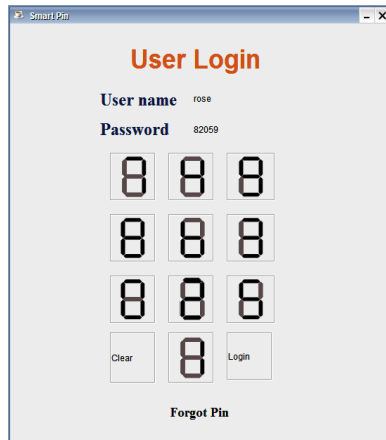
5.3 TEST SCREENS

UNIT TESTING



User withdraw amount

INTEGRATION TESTING



Smart Pin

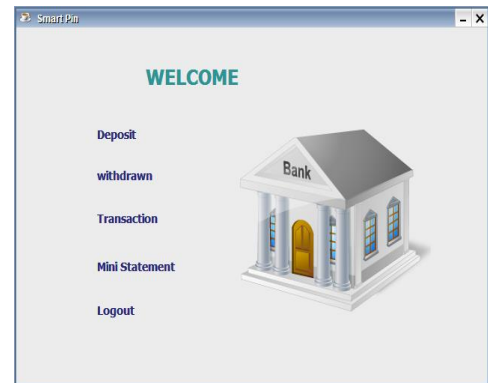
User Login

User name: rosh
Password: 82059

8 8 8
8 8 8
8 8 8

Clear 8 Login

Forgot Pin



Smart Pin

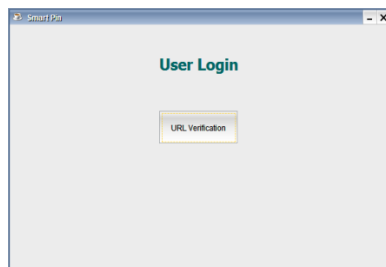
WELCOME

Deposit
withdrawn
Transaction
Mini Statement
Logout

Bank

If user enter correct pin home page will be open

SYSTEM TESTING



Smart Pin

User Login

URL Verification

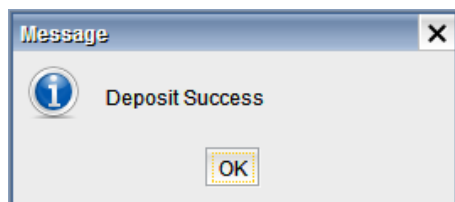


Vein Based Banking System

Deposit

Amount: 1000

deposit



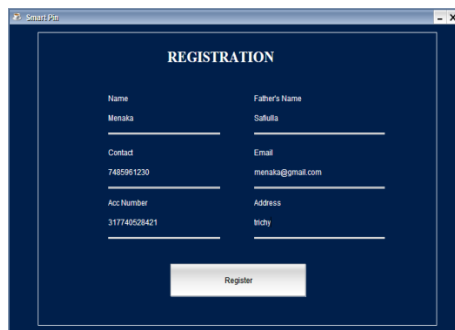
Message

1 Deposit Success

OK

User login and deposit amount

VALIDATION TESTING



Smart Pin

REGISTRATION

Name	Father's Name
Mensaka	Sakula
Contact	Email
7480981230	mensaka@gmail.com
Acc Number	Address
31774020421	richy

Register

User registration

CHAPTER 6

CONCLUSION

In this paper illustrate the proposed framework named as Hybrid Pin Keypad which is valuable to diminish the bear surfing attack. The most reason of this verification method is, the client can login the framework without caring approximately shoulder surfers. The client has effortlessly memorized the Stick and the login process is expended less time. This proposed strategy is breaking a few conventional attacks such as word reference attack and brute constrain attack. The future work is to make strides the higher proficiency and executed may be entryway locks, mobile careen locks and get to the person applications. The process of identifying an individual usually based on a username and password. In security systems, authentication is distinct from authorization, which is the process of giving individuals access to system objects based on their identity. There are three common factors used for authentication: Password, Smart card and URL verification. The proposed method uses the PIN details and the biometric details for registration and login into a system. This method provides complete security as it uses unique biometric method. Computational cost is less for this method. The transactions can be initiated with storage data. Also, the number of invalid attempts can be fixed.

6.1 APPENDIX

a) SOURCE CODE

User Registration:

```
import java.awt.Dimension;
import java.awt.Toolkit;
import java.io.IOException;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.time.LocalDateTime;
import java.time.format.DateTimeFormatter;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JOptionPane;
import static project.Project.conf;
public class user_registration extends javax.swing.JFrame {
    boolean d=conf();
    public static String uid="";
    public static String uname="";
    java.util.Calendar calendar = java.util.Calendar.getInstance();
    public user_registration() {
        initComponents();
        this.setLocationRelativeTo(null);
        this.setResizable(false);
        this.setTitle(home.title);
```

```

String day = String.valueOf(calendar.get(java.util.Calendar.DATE));
String month = String.valueOf(calendar.get(java.util.Calendar.MONTH) + 1);
String year = String.valueOf(calendar.get(java.util.Calendar.YEAR));
String hrs = String.valueOf(calendar.get(java.util.Calendar.HOUR));
String min = String.valueOf(calendar.get(java.util.Calendar.MINUTE));
String sec = String.valueOf(calendar.get(java.util.Calendar.SECOND));
String rn=Crypt.getotp(4);
String acc=month+""+day+""+hrs+""+min+""+sec+""+rn;
accno.setText(acc);
accno.setEditable(false);
}
@SuppressWarnings("unchecked")
// <editor-fold defaultstate="collapsed" desc="Generated Code">//GEN-
BEGIN:initComponents
private void initComponents() {

jPanel1 = new javax.swing.JPanel();
jPanel2 = new javax.swing.JPanel();
jPanel3 = new javax.swing.JPanel();
jLabel1 = new javax.swing.JLabel();
jLabel2 = new javax.swing.JLabel();
name = new javax.swing.JTextField();
jLabel3 = new javax.swing.JLabel();
fname = new javax.swing.JTextField();
jSeparator1 = new javax.swing.JSeparator();
jSeparator2 = new javax.swing.JSeparator();
jLabel4 = new javax.swing.JLabel();
jLabel5 = new javax.swing.JLabel();
cont = new javax.swing.JTextField();

```

```

email = new javax.swing.JTextField();
jSeparator3 = new javax.swing.JSeparator();
jSeparator4 = new javax.swing.JSeparator();
jLabel6 = new javax.swing.JLabel();
accno = new javax.swing.JTextField();
jLabel7 = new javax.swing.JLabel();
address = new javax.swing.JTextField();
jSeparator5 = new javax.swing.JSeparator();
jSeparator6 = new javax.swing.JSeparator();
jButton1 = new javax.swing.JButton();

javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);
jPanel1.setLayout(jPanel1Layout);
jPanel1Layout.setHorizontalGroup(
    jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGap(0, 100, Short.MAX_VALUE)
);
jPanel1Layout.setVerticalGroup(
    jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGap(0, 100, Short.MAX_VALUE)
);

setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE_ON_CLOSE);

jPanel2.setBackground(new java.awt.Color(0, 31, 75));

jPanel3.setBackground(new java.awt.Color(0, 31, 75));
jPanel3.setBorder(javax.swing.BorderFactory.createLineBorder(new
    java.awt.Color(255, 255, 255)));
jLabel1.setFont(new java.awt.Font("Times New Roman", 1, 24)); // NOI18N

```

```
jLabel1.setForeground(new java.awt.Color(255, 255, 255));  
jLabel1.setText("REGISTRATION");
```

```
jLabel2.setForeground(new java.awt.Color(255, 255, 255));  
jLabel2.setText("Name");
```

```
name.setBackground(new java.awt.Color(0, 31, 75));  
name.setForeground(new java.awt.Color(255, 255, 255));  
name.setToolTipText("Enter Your Name");  
name.setBorder(null);
```

```
jLabel3.setForeground(new java.awt.Color(255, 255, 255));  
jLabel3.setText("Father's Name");
```

```
fname.setBackground(new java.awt.Color(0, 31, 75));  
fname.setForeground(new java.awt.Color(255, 255, 255));  
fname.setToolTipText("Enter Your Father Name");  
fname.setBorder(null);
```

```
jLabel4.setForeground(new java.awt.Color(255, 255, 255));  
jLabel4.setText("Contact");
```

```
jLabel5.setForeground(new java.awt.Color(255, 255, 255));  
jLabel5.setText("Email");  
cont.setBackground(new java.awt.Color(0, 31, 75));  
cont.setForeground(new java.awt.Color(255, 255, 255));  
cont.setToolTipText("Enter Your Contact");  
cont.setBorder(null);  
email.setBackground(new java.awt.Color(0, 31, 75));
```

```
email.setForeground(new java.awt.Color(255, 255, 255));  
email.setToolTipText("Enter Your Email Id");  
email.setBorder(null);
```

```
jLabel6.setForeground(new java.awt.Color(255, 255, 255));  
jLabel6.setText("Acc Number");
```

```
accno.setBackground(new java.awt.Color(0, 31, 75));  
accno.setForeground(new java.awt.Color(255, 255, 255));  
accno.setBorder(null);
```

```
jLabel7.setForeground(new java.awt.Color(255, 255, 255));  
jLabel7.setText("Address");
```

```
address.setBackground(new java.awt.Color(0, 31, 75));  
address.setForeground(new java.awt.Color(255, 255, 255));  
address.setToolTipText("Enter Your Address");  
address.setBorder(null);  
jButton1.setText("Register");  
jButton1.addActionListener(new java.awt.event.ActionListener() {  
    public void actionPerformed(java.awt.event.ActionEvent evt) {  
        jButton1ActionPerformed(evt);  
    }  
});
```

```
javax.swing.GroupLayout jPanel3Layout = new javax.swing.GroupLayout(jPanel3);  
jPanel3.setLayout(jPanel3Layout);  
jPanel3Layout.setHorizontalGroup(  
    jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)  
        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
```

```

jPanel3Layout.createSequentialGroup()
.addGap(119, 119, 119)
.addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
.addGroup(jPanel3Layout.createSequentialGroup()
.addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
.addGroup(jPanel3Layout.createSequentialGroup()
.addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
.addComponent(name, javax.swing.GroupLayout.PREFERRED_SIZE, 191,
javax.swing.GroupLayout.PREFERRED_SIZE)
.addComponent(jSeparator5, javax.swing.GroupLayout.Alignment.TRAILING)
.addComponent(accno)
.addComponent(jSeparator3, javax.swing.GroupLayout.Alignment.TRAILING)
.addComponent(cont)
.addComponent(jLabel6)
.addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel3Layout.createSequentialGroup()
.addGap(0, 0, Short.MAX_VALUE)
.addComponent(jSeparator1, javax.swing.GroupLayout.PREFERRED_SIZE, 191,
javax.swing.GroupLayout.PREFERRED_SIZE)))
.addGap(58, 58, 58))
.addGroup(jPanel3Layout.createSequentialGroup()
.addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
.addComponent(jLabel4)
.addComponent(jLabel2))

```

```

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)))
.addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
.addComponent(jLabel7)
.addComponent(address, javax.swing.GroupLayout.DEFAULT_SIZE, 223,
Short.MAX_VALUE)
.addComponent(jSeparator6)
.addComponent(jSeparator4, javax.swing.GroupLayout.Alignment.TRAILING)
.addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
.addComponent(jLabel5)
.addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
.addComponent(jSeparator2, javax.swing.GroupLayout.Alignment.TRAILING,
javax.swing.GroupLayout.DEFAULT_SIZE, 223, Short.MAX_VALUE)
.addComponent(jLabel3)
.addComponent(fname, javax.swing.GroupLayout.Alignment.TRAILING))
.addComponent(email, javax.swing.GroupLayout.PREFERRED_SIZE, 145,
javax.swing.GroupLayout.PREFERRED_SIZE))))
.addGroup(jPanel3Layout.createSequentialGroup())
.addComponent(jLabel1)
.addGap(189, 189, 189))
.addGap(88, 88, 88))
.addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel3Layout.createSequentialGroup())
.addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
.addComponent(jButton1, javax.swing.GroupLayout.PREFERRED_SIZE, 232,
javax.swing.GroupLayout.PREFERRED_SIZE)
.addGap(222, 222, 222))

```

```

);
jPanel3Layout.setVerticalGroup(
jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
.addGroup(jPanel3Layout.createSequentialGroup())
.addGap(21, 21, 21)
.addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED_SIZE, 32,
javax.swing.GroupLayout.PREFERRED_SIZE)
.addGap(39, 39, 39)
.addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BA
SELINE)
.addComponent(jLabel2)
.addComponent(jLabel3))
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
.addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LE
ADING)
.addComponent(fname, javax.swing.GroupLayout.PREFERRED_SIZE, 33,
javax.swing.GroupLayout.PREFERRED_SIZE)
.addComponent(name, javax.swing.GroupLayout.PREFERRED_SIZE, 33,
javax.swing.GroupLayout.PREFERRED_SIZE))
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
.addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LE
ADING)
.addComponent(jSeparator1, javax.swing.GroupLayout.PREFERRED_SIZE, 10,
javax.swing.GroupLayout.PREFERRED_SIZE)
.addComponent(jSeparator2, javax.swing.GroupLayout.PREFERRED_SIZE, 10,
javax.swing.GroupLayout.PREFERRED_SIZE))
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
.addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LE
ADING)

```

```

.addComponent(jLabel4)
.addComponent(jLabel5, javax.swing.GroupLayout.Alignment.TRAILING))
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
.addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
.addComponent(cont, javax.swing.GroupLayout.DEFAULT_SIZE, 33,
Short.MAX_VALUE)
.addComponent(email))
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
.addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
.addComponent(jLabel6)
.addComponent(jLabel7))
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
.addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
.addComponent(accno, javax.swing.GroupLayout.DEFAULT_SIZE, 33,
Short.MAX_VALUE)
.addComponent(address))
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
.addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

```

```

.addComponent(jSeparator5, javax.swing.GroupLayout.PREFERRED_SIZE, 10,
javax.swing.GroupLayout.PREFERRED_SIZE)
.addComponent(jSeparator6, javax.swing.GroupLayout.PREFERRED_SIZE, 10,
javax.swing.GroupLayout.PREFERRED_SIZE))
.addGap(31, 31, 31)
.addComponent(jButton1, javax.swing.GroupLayout.PREFERRED_SIZE, 51,
javax.swing.GroupLayout.PREFERRED_SIZE)
.addContainerGap(43, Short.MAX_VALUE))
);

```

```

javax.swing.GroupLayout jPanel2Layout = new javax.swing.GroupLayout(jPanel2);
jPanel2.setLayout(jPanel2Layout);
jPanel2Layout.setHorizontalGroup(
jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
.addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel2Layout.createSequentialGroup()
.addContainerGap(44, Short.MAX_VALUE)
.addComponent(jPanel3, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
.addGap(38, 38, 38))
);
jPanel2Layout.setVerticalGroup(
jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
.addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel2Layout.createSequentialGroup()
.addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
.addComponent(jPanel3, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,

```

```

javax.swing.GroupLayout.PREFERRED_SIZE)
.addGap(115, 115, 115))
);

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(
layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
.addComponent(jPanel2, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
);
layout.setVerticalGroup(
layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
.addComponent(jPanel2, javax.swing.GroupLayout.PREFERRED_SIZE, 474,
javax.swing.GroupLayout.PREFERRED_SIZE)
);

pack();
} // </editor-fold> // GEN-END: initComponents

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) { // GEN-
FIRST:event_jButton1ActionPerformed
try {

if(d)
{
String day = String.valueOf(calendar.get(java.util.Calendar.DATE));
String month = String.valueOf(calendar.get(java.util.Calendar.MONTH) + 1);
String year = String.valueOf(calendar.get(java.util.Calendar.YEAR));
String hrs = String.valueOf(calendar.get(java.util.Calendar.HOUR));

```

```

String min = String.valueOf(calendar.get(java.util.Calendar.MINUTE));
String cdate = day + "-" + month + "-" + year;
String pass=Crypt.getotp(5);
String name1=name.getText().toString().trim();
String fname1=fname.getText().toString().trim();
String cont1=cont.getText().toString().trim();
String email1=email.getText().toString().trim();
String accno1=accno.getText().toString().trim();
String address1=address.getText().toString().trim();
Mail ob = new Mail();
ob.SendMail(email1, "Smart pin - Account Registration", "Account No :
"+accno1+"\nPassword : "+pass, "");
String url="jdbc:mysql://localhost:3306/Hybrid_pin";
Class.forName("com.mysql.jdbc.Driver");
Connection Con=DriverManager.getConnection(url,"root","");
Statement stmt=Con.createStatement();
Statement stmt1=Con.createStatement();
String qry = "select max(id) as maxid from user_reg";
ResultSet rs = stmt1.executeQuery(qry);
int id1 = 0;
if (rs.next()) {
id1 = rs.getInt("maxid");
}
int id2 = id1 + 1;
String ins = "insert into user_reg
values('"+id2+"','"+name1+"','"+fname1+"','"+cont1+"','"+email1+"','"+accno1+"','"+address1+"','"+cdate+"','"+pass+"','0','0','0','0','0','0')";
int n = stmt.executeUpdate(ins);
if(n==1)

```

```

{
uid="" + id2;
uname=name1;
//JacspcscLoader js=new JacspcscLoader();
ACSMModule js1=new ACSModule();
js1.write(accno1);

JOptionPane.showMessageDialog(this,"Registration success\n\nCheck Your Mail ID");
//new Enrollment().setVisible(true);
// new home().setVisible(true);
this.setVisible(false);
}
else
{
JOptionPane.showMessageDialog(this,"Registration failed");

}
}
else
{
JOptionPane.showMessageDialog(this,"Registration Success");
this.setVisible(false);
}
} catch (ClassNotFoundException ex) {
Logger.getLogger(user_registration.class.getName()).log(Level.SEVERE, null, ex);
} catch (SQLException ex) {
Logger.getLogger(user_registration.class.getName()).log(Level.SEVERE, null, ex);
}
}
} //GEN-LAST:event_jButton1ActionPerformed

```

```

//<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
/* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and
feel.
* For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
*/
try {
for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
if ("Nimbus".equals(info.getName())) {
javax.swing.UIManager.setLookAndFeel(info.getClassName());
break;
}
}
} catch (ClassNotFoundException ex) {
java.util.logging.Logger.getLogger(user_registration.class.getName()).log(java.util.loggi
ng.Level.SEVERE, null, ex);
} catch (InstantiationException ex) {
java.util.logging.Logger.getLogger(user_registration.class.getName()).log(java.util.loggi
ng.Level.SEVERE, null, ex);
} catch (IllegalAccessException ex) {
java.util.logging.Logger.getLogger(user_registration.class.getName()).log(java.util.loggi
ng.Level.SEVERE, null, ex);
} catch (javax.swing.UnsupportedLookAndFeelException ex) {
java.util.logging.Logger.getLogger(user_registration.class.getName()).log(java.util.loggi
ng.Level.SEVERE, null, ex);
}
//</editor-fold>

/* Create and display the form */
java.awt.EventQueue.invokeLater(new Runnable() {

```

```

public void run() {
    new user_registration().setVisible(true);
}
});
}
// Variables declaration - do not modify//GEN-BEGIN:variables
private javax.swing.JTextField accno;
private javax.swing.JTextField address;
private javax.swing.JTextField cont;
private javax.swing.JTextField email;
private javax.swing.JTextField fname;
private javax.swing.JButton jButton1;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel4;
private javax.swing.JLabel jLabel5;
private javax.swing.JLabel jLabel6;
private javax.swing.JLabel jLabel7;
private javax.swing.JPanel jPanel1;
private javax.swing.JPanel jPanel2;
private javax.swing.JPanel jPanel3;
private javax.swing.JSeparator jSeparator1;
private javax.swing.JSeparator jSeparator2;
private javax.swing.JSeparator jSeparator3;
private javax.swing.JSeparator jSeparator4;
private javax.swing.JSeparator jSeparator5;
private javax.swing.JSeparator jSeparator6;

```

```

private javax.swing.JTextField name;
// End of variables declaration//GEN-END:variables
}

```

User Transaction:

```

package smart_pin;

import java.awt.Dimension;

import java.awt.Toolkit;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.Statement;
import java.time.LocalDateTime;
import java.time.format.DateTimeFormatter;
import javax.swing.JOptionPane;

public class user_transaction extends javax.swing.JFrame {
    String accn="";

    public user_transaction() {
        initComponents();
        this.setTitle("Banking");
        accn=home.accnumber;
        setResizable(false);

        Dimension dimension = Toolkit.getDefaultToolkit().getScreenSize();
        int x = (int) ((dimension.getWidth() - this.getWidth()) / 2);
        int y = (int) ((dimension.getHeight() - this.getHeight()) / 2);
        this.setLocation(x, y);
    }

    @SuppressWarnings("unchecked")

```

```
// <editor-fold defaultstate="collapsed" desc="Generated Code">//GEN-BEGIN:initComponents
private void initComponents() {
jLabel1 = new javax.swing.JLabel();
recev = new javax.swing.JTextField();
jLabel2 = new javax.swing.JLabel();
jButton1 = new javax.swing.JButton();
jLabel3 = new javax.swing.JLabel();
amt = new javax.swing.JTextField();
setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE_ON_CLOSE);
jLabel1.setFont(new java.awt.Font("Tahoma", 3, 18)); // NOI18N
jLabel1.setForeground(new java.awt.Color(40, 137, 137));
jLabel1.setText("Transaction");
jLabel2.setText("Amount : ");
jButton1.setText("Transaction");
jButton1.addActionListener(new java.awt.event.ActionListener() {
public void actionPerformed(java.awt.event.ActionEvent evt) {
jButton1ActionPerformed(evt);
}
});
jLabel3.setText("Acc Number :");
amt.addActionListener(new java.awt.event.ActionListener() {
public void actionPerformed(java.awt.event.ActionEvent evt) {
amtActionPerformed(evt);
}
});
javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(
```

```

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
.addGroup(layout.createSequentialGroup())
.addGap(141, 141, 141)
.addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED_SIZE, 113,
javax.swing.GroupLayout.PREFERRED_SIZE)
.addContainerGap(146, Short.MAX_VALUE))
.addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
layout.createSequentialGroup())
.addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING,
false)
.addComponent(jButton1, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
.addGroup(layout.createSequentialGroup())
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
.addComponent(jLabel3, javax.swing.GroupLayout.PREFERRED_SIZE, 90,
javax.swing.GroupLayout.PREFERRED_SIZE)
.addComponent(jLabel2, javax.swing.GroupLayout.PREFERRED_SIZE, 71,
javax.swing.GroupLayout.PREFERRED_SIZE))
.addGap(18, 18, 18)
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING,
false)
.addComponent(recev, javax.swing.GroupLayout.DEFAULT_SIZE, 112,
Short.MAX_VALUE)
.addComponent(amt))))
.addGap(79, 79, 79))
);
layout.setVerticalGroup(
layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
.addGroup(layout.createSequentialGroup())
.addContainerGap()

```

```

.addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED_SIZE, 30,
javax.swing.GroupLayout.PREFERRED_SIZE)
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 44,
Short.MAX_VALUE)
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
.addComponent(jLabel3, javax.swing.GroupLayout.PREFERRED_SIZE, 30,
javax.swing.GroupLayout.PREFERRED_SIZE)
.addComponent(recev, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
javax.swing.GroupLayout.PREFERRED_SIZE))
.addGap(20, 20, 20)
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
.addComponent(amt, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
javax.swing.GroupLayout.PREFERRED_SIZE)
.addComponent(jLabel2, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
.addGap(35, 35, 35)
.addComponent(jButton1, javax.swing.GroupLayout.PREFERRED_SIZE, 42,
javax.swing.GroupLayout.PREFERRED_SIZE)
.addGap(50, 50, 50))
);
pack();
} // </editor-fold> // GEN-END: initComponents

private void amtActionPerformed(java.awt.event.ActionEvent evt) { // GEN-
FIRST:event_amtActionPerformed
// TODO add your handling code here:
} // GEN-LAST:event_amtActionPerformed

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) { // GEN-
FIRST:event_jButton1ActionPerformed
try

```

```

{
java.util.Calendar calendar = java.util.Calendar.getInstance();
String day = String.valueOf(calendar.get(java.util.Calendar.DATE));
String month = String.valueOf(calendar.get(java.util.Calendar.MONTH) + 1);
String year = String.valueOf(calendar.get(java.util.Calendar.YEAR));
String hrs = String.valueOf(calendar.get(java.util.Calendar.HOUR));
String min = String.valueOf(calendar.get(java.util.Calendar.MINUTE));
//String year = String.valueOf(calendar.get(java.util.Calendar.YEAR));
String ty=String.valueOf(calendar.get(java.util.Calendar.AM_PM));
LocalTime localTime = LocalTime.now();
DateTimeFormatter dateTimeFormatter = DateTimeFormatter.ofPattern("hh:mm a");
String tk=localTime.format(dateTimeFormatter);
String cdate = day + "-" + month + "-" + year;
int am=Integer.parseInt(amt.getText().toString());
int rec=Integer.parseInt(recev.getText().toString());
int bal=0;
//String mobile="";
String url = "jdbc:mysql://localhost:3306/hybrid_pin";
String userid = "root";
String password = "";
////
String sql = "SELECT * FROM user_reg where accno=" + accn + "";
Connection connection = DriverManager.getConnection( url, userid, password );
Statement stmt = connection.createStatement();
Statement stmt1 = connection.createStatement();
Statement stmt2 = connection.createStatement();
Statement stmtr = connection.createStatement();
ResultSet rs = stmt.executeQuery( sql );
while (rs.next()) {
bal= rs.getInt("balance");

```

```

//mobile= rs.getString("contact");
}
if(bal>=am)
{
int in=bal-am;
String sql1 = "update user_reg SET balance = '"+in+"' WHERE accno='"+accn+"'";
stmt1.executeUpdate(sql1);
////
int bal1=0;
String sql2 = "SELECT * FROM user_reg where accno=" + rec + " ";
ResultSet rs1 = stmt2.executeQuery( sql2 );
while (rs1.next()) {
bal1= rs1.getInt("balance");
}
int in1=bal1+am;
String sql4 = "update user_reg SET balance = '"+in1+"' WHERE accno='"+rec+"'";
stmt2.executeUpdate(sql4);
////
String ins = "insert into mini
values('"+accn+"','Debited','"+am+"','"+rec+"','"+cdate+"','"+tk+"')";
int n = stmt.executeUpdate(ins);
////
String insr = "insert into mini
values('"+rec+"','Credited','"+am+"','"+accn+"','"+cdate+"','"+tk+"')";
int n1 = stmtr.executeUpdate(insr);
if(n1==1)
{
JOptionPane.showMessageDialog(this,"Success");
}
else
{

```

```

JOptionPane.showMessageDialog(this,"Failed");
}
}
else
{
JOptionPane.showMessageDialog(this,"Low Bal");
}
this.setVisible(false);
}
catch(Exception e)
{
System.out.println(e.getMessage());
}
}
//GEN-LAST:event_jButton1ActionPerformed
try {
for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
if ("Nimbus".equals(info.getName())) {
javax.swing.UIManager.setLookAndFeel(info.getClassName());
break;
}
}
} catch (ClassNotFoundException ex) {
java.util.logging.Logger.getLogger(user_transaction.class.getName()).log(java.util.logging.
Level.SEVERE, null, ex);
} catch (InstantiationException ex) {
java.util.logging.Logger.getLogger(user_transaction.class.getName()).log(java.util.logging.
Level.SEVERE, null, ex);
} catch (IllegalAccessException ex) {
java.util.logging.Logger.getLogger(user_transaction.class.getName()).log(java.util.logging

```

```

g.Level.SEVERE, null, ex);
} catch (javax.swing.UnsupportedLookAndFeelException ex) {
java.util.logging.Logger.getLogger(user_transaction.class.getName()).log(java.util.logging
g.Level.SEVERE, null, ex);
}
//</editor-fold>
/* Create and display the form */
java.awt.EventQueue.invokeLater(new Runnable() {
public void run() {
new user_transaction().setVisible(true);
}
});
}
// Variables declaration - do not modify//GEN-BEGIN:variables
private javax.swing.JTextField amt;
private javax.swing.JButton jButton1;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JTextField recev;
// End of variables declaration//GEN-END:variables
}

```

6.2 SAMPLE OUTPUTS



Fig 1 User login screen

The image shows a web application window titled "Smart Pin". The main content area is a white rectangular box with the heading "REGISTRATION" at the top. Below the heading, there are two columns of form fields. The left column contains fields for "Name" (filled with "Menaka"), "Contact" (filled with "7485961230"), and "Acc Number" (filled with "317740528421"). The right column contains fields for "Father's Name" (filled with "Safiulla"), "Email" (filled with "menaka@gmail.com"), and "Address" (filled with "trichy"). At the bottom center of the form is a "Register" button.

fig 2 User registration

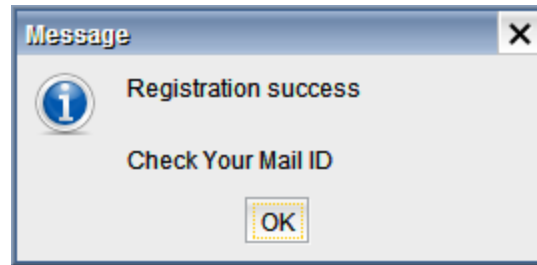


Fig 3 Mail ID verification

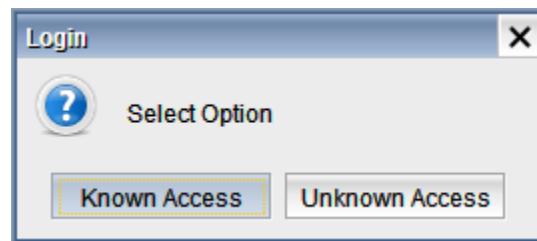


Fig 4 Access verification

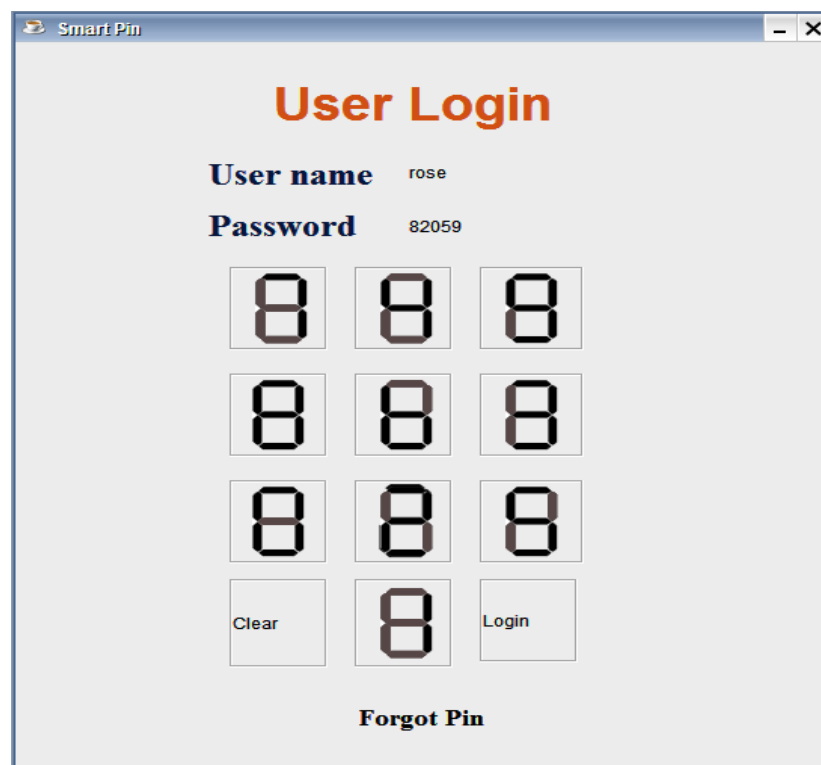


Fig 5 hybrid PIN shuffling

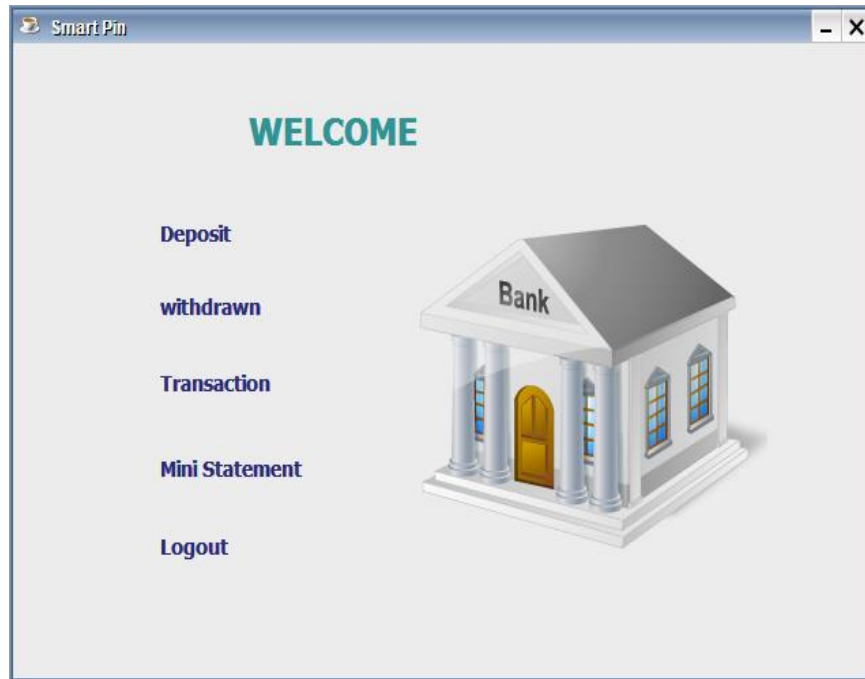


Fig 6 User Transaction

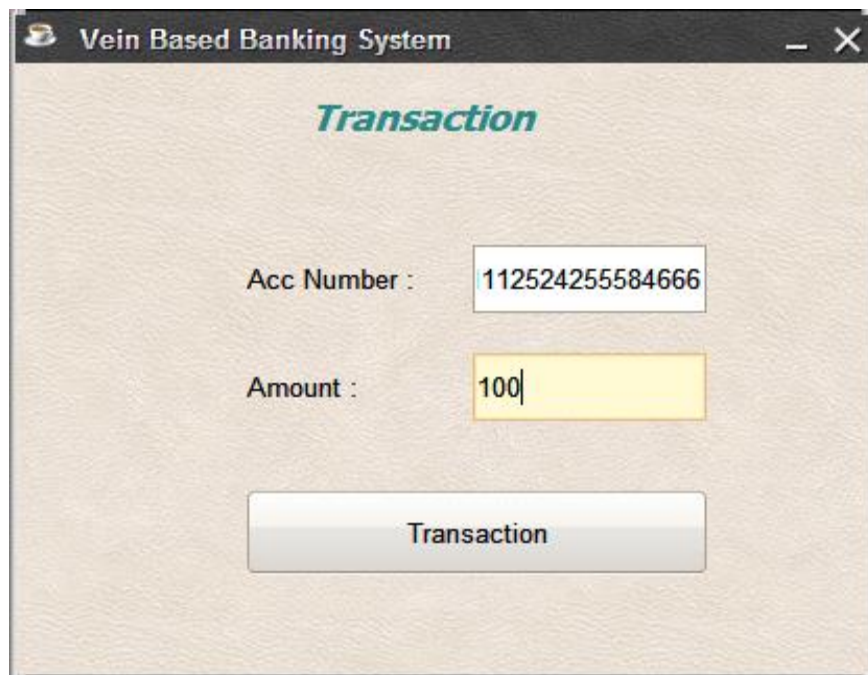


Fig 7 Amount Deposit



The image shows a software window titled "Vein Based Banking System". The main heading is "Withdrawn" in a green, italicized font. Below this, there is a label "Amount :" followed by a yellow rectangular input field containing the number "100". At the bottom center, there is a grey button with the text "withdraw".

Fig 8 Amount withdraw



The image shows a software window titled "Vein Based Banking System". The main heading is "Transaction" in a green, italicized font. Below this, there are two input fields. The first is labeled "Acc Number :" and contains the number "112524255584666". The second is labeled "Amount :" and contains the number "100". At the bottom center, there is a grey button with the text "Transaction".

Fig 9 Amount Transaction

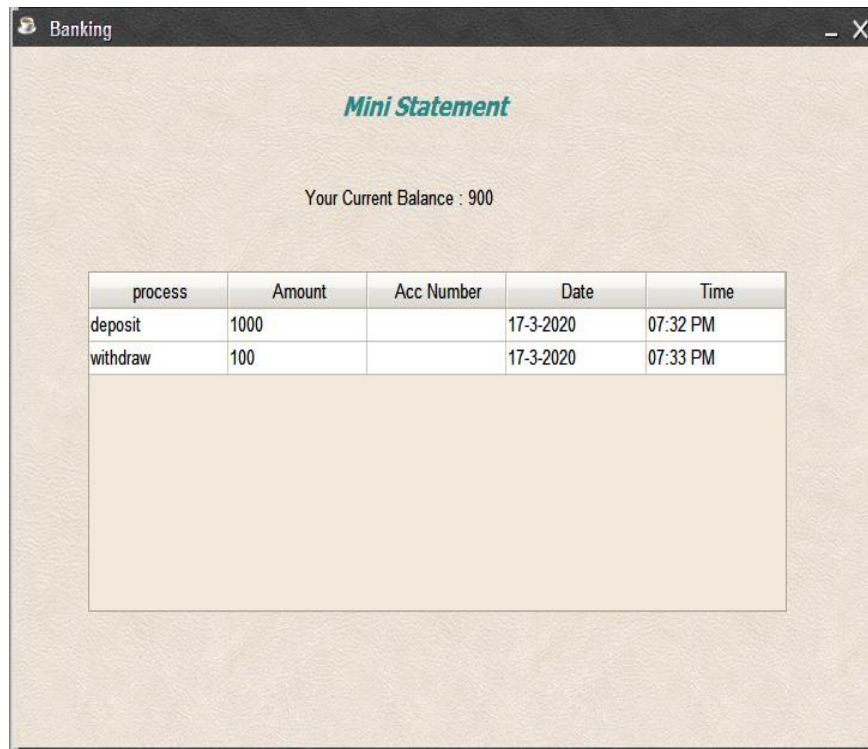


Fig 10 Mini Statement

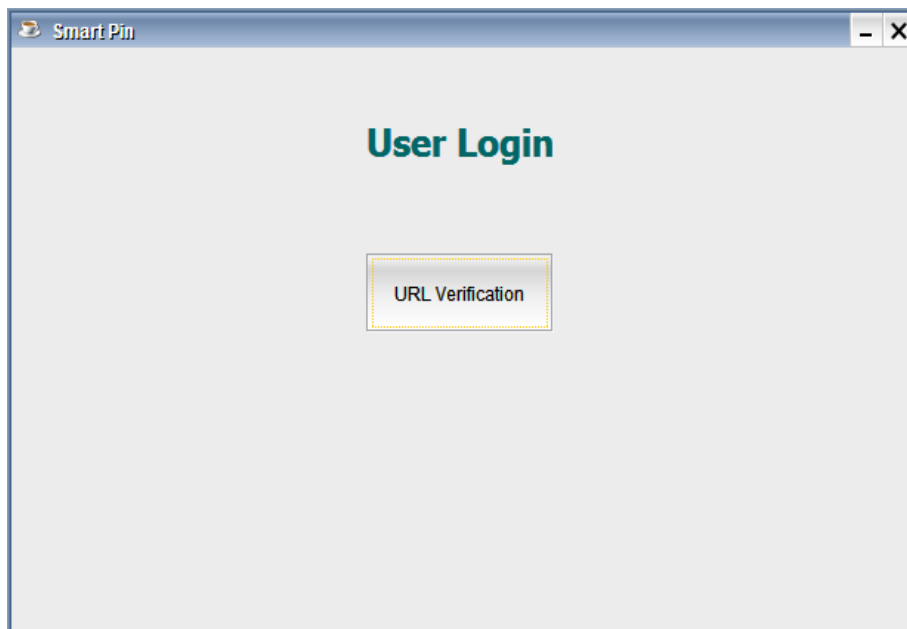


Fig 11 Unknown Accesses

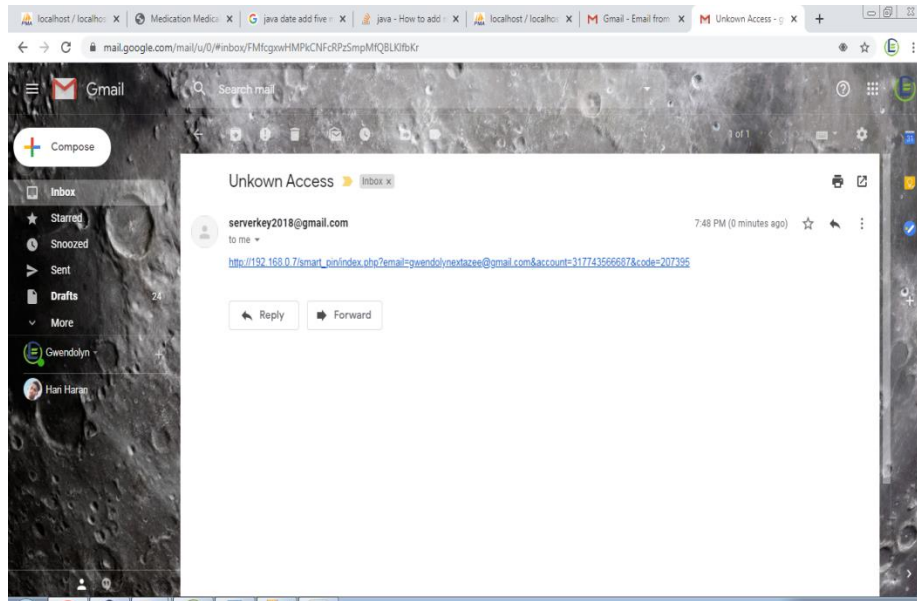


Fig 12 URL verification

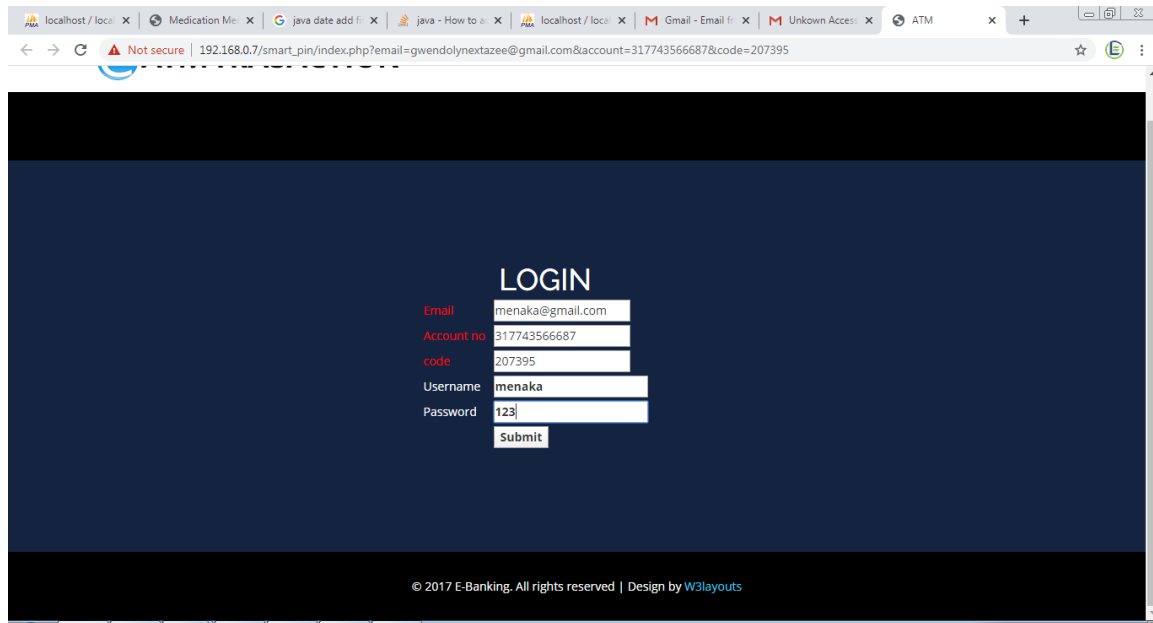


Fig 13 User remote Login

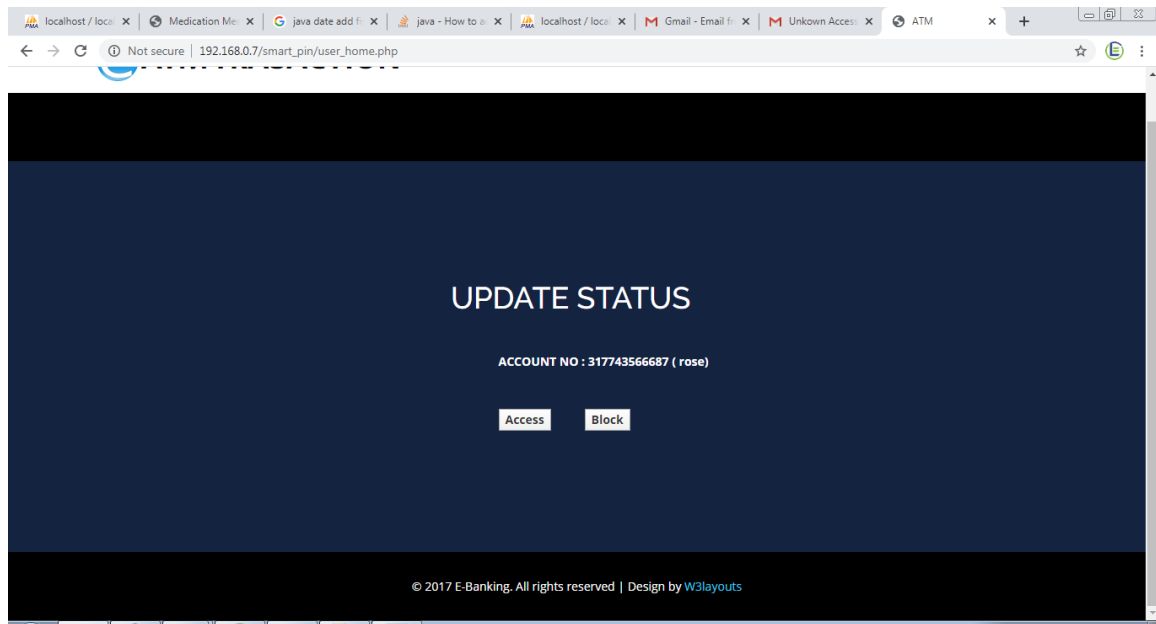


Fig 14 User access or block

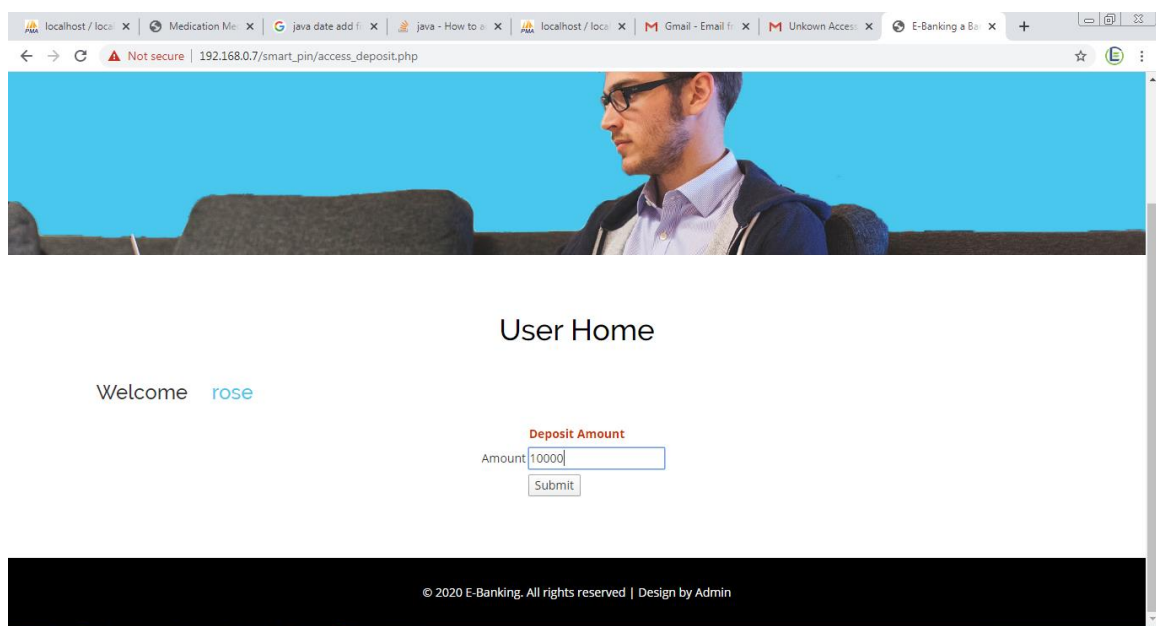


Fig 15 User deposit

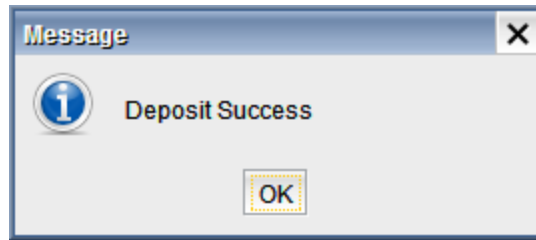


Fig 16 Unknown user access

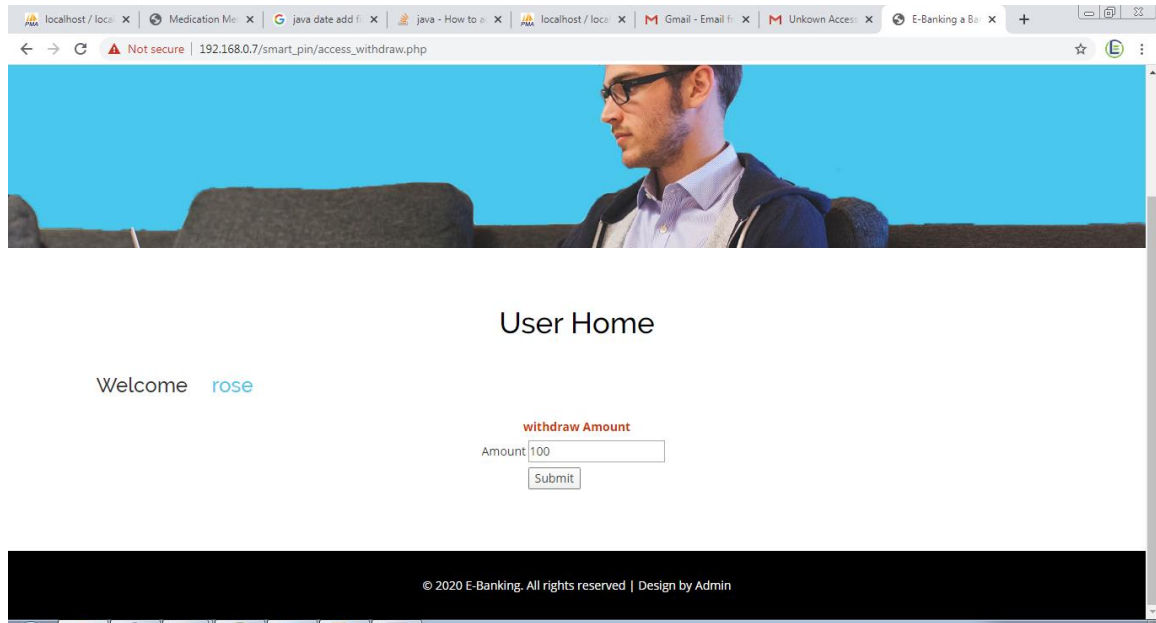


Fig 17 User remote withdraw

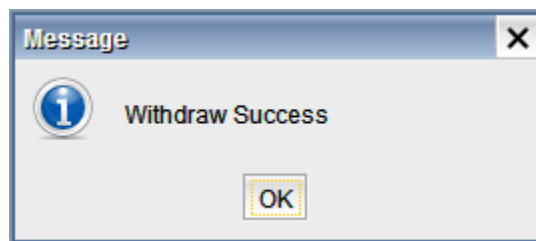


Fig 18 Unknown user withdraw

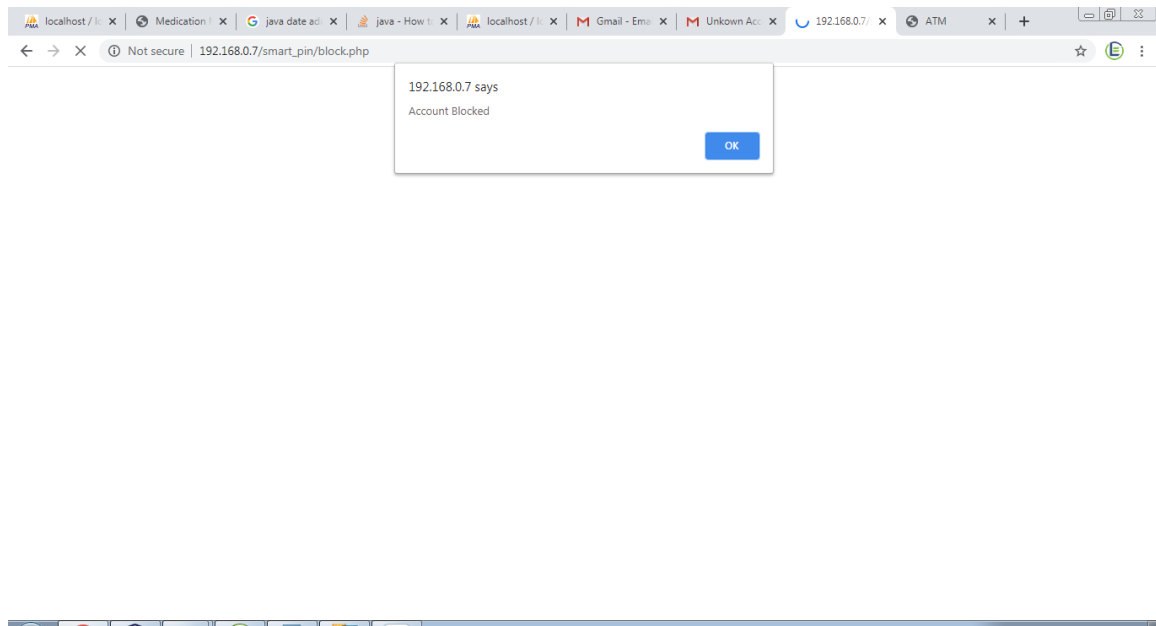


Fig 19 User card Blocked

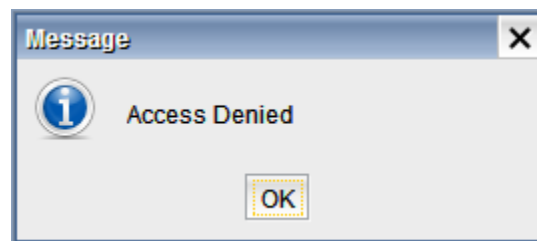


Fig 20 Access will be denied

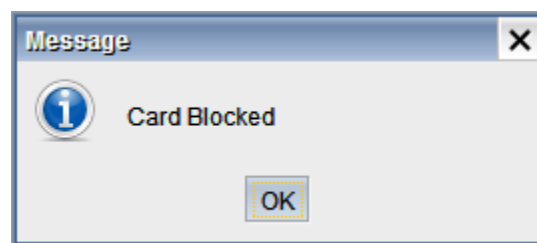


Fig 21 Card Blocked

CHAPTER 6

REFERENCES

Book Reference

- [1] Alvin J Bruney M V P,"The Microsoft Office The project b Components Black Book with .NET ", Lulu.com (9 April 2005)
- [2] Brennan Stehling," Pro ASP.NET for SQL Server: High Performance Data Access for The project b Developers (Expert's Voice in SQL Server)", Apress; 1st ed. edition (26 September 2007)
- [3] Robin Dewson," Beginning SQL Server 2005 Express for Developers: From Novice to Professional (Expert's Voice in .NET) ", Apress; 1st ed. edition (8 February 2007)
- [4] Matthew Macdonald, " ASP.NET: The Complete Reference", McGraw Hill Education (1 July 2017)

The project b reference

- [1] <https://dotnet.microsoft.com/>
- [2] https://en.wikipedia.org/wiki/.NET_Framework
- [3] <https://www.geeksforgeeks.org/introduction-to-net-framework/>