BIO TOUCH PASS: HANDWRITTEN PASSWORDS FOR TOUCH SCREEN BIOMETRICS

A PROJECT REPORT

Submitted by

SELVA SATHISH T 3 1 2 0 1 9 2 0 5 0 2 8

RAJADURAI S 3 1 2 0 1 9 2 0 5 0 2 2

in partial fulfillment for the award of the degree
of

BACHELOR OF TECHNOLOGY

in

INFORMATION TECHNOLOGY

JEPPIAAR SRR ENGINEERING COLLEGE, PADUR

ANNA UNIVERSITY : CHENNAI 600 025

APRIL 2023

BONAFIDE CERTIFICATE

Certified that this project report **"BIO TOUCH PASS: HANDWRITTEN PASSWORDS FOR TOUCH SCREEN BIOMETRICS"** is the bonafide work of **"SELVA SATHISH T (312019205028), RAJADURAI S (312019205022) "** who carried out the project under my supervision.

Mrs. J.Gunaseeli M.E.,	**Mrs. J.Gunaseeli M.E.,**
HEAD OF DEPARTMENT	**INTERNAL GUIDE**
Department of Information Technology	Department of Information Technology
Jeppiaar SRR Engineering College,	Jeppiaar SRR Engineering College,
Old Mamallapuram Road,	Old Mamallapuram Road,
Padur, Chennai – 603 103	Padur, Chennai – 603 103

Submitted for the examination held on ________________________________

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ACKNOWLEGEMENT

We take this opportunity to express our profound gratitude and deep regard to our beloved **Founder Chairman (Late) Col. Dr. JEPPIAAR M.A., B.L., Ph.D.,** for enlightening our lives and showering heavenly blessings forever.

We also express our heartfelt thanks to our **Chairman and Managing Director Dr. REGEENA JEPPIAAR B.E., M.B.A., Ph.D.,** for her kind cooperation and keen interest for the success of the project.

We are immensely happy to accord the warmth of gratitude to our **Director Mr. MURLI SUBRAMANIAN** for being the beacon in all our endeavors.

We express our profound gratitude to our **Principal Dr. M. SASIKUMAR M.TECH., Ph.D.,** for bringing out novelty in all executions.

We express our thanks to our **Head of the Department Mrs. J. Gunaseeli M.E.,** for his valuable suggestions and guidance for the development and completion of this project.

We are highly thankful to our project **Internal Guide Mrs. J. Gunaseeli M.E.,** for guidance and encouragement in carrying out this project work.

We are much obliged to all our teaching and non-teaching staff members for their valuable information and constructive criticism that immensely contributed to the development of the project.

Above all, we wish to avail this opportunity to express a sense of gratitude and love to our beloved parents and friends for their moral support and constant strength at various stages of our project

TABLE OF CONTENTS

ABSTRACT

"Efficient password Mechanism To Overcome Spyware Attacks" is to develop user-friendly mobile applications ensuring data protection and high security. In addition to the high level of technological advancement that enables real-time social media usage and communication, Personal Identification Numbers and One- Time Passwords have become the two most popular methods of user authentication.This has led to the rapid and widespread deployment of mobile devices throughout the world. Our suggested solution focuses on offering mobile applications that are easy to use while still delivering excellent security and data protection. Instead of inputting the password as normal, the user should sketch each digit on the touch screen. In this approach, the static handwriting biometric data of the standard authentication systems is improved. Our system has two levels of authentication, and the pin that is drawn must match the pin that was entered during registration.Our second authentication stage offers a variety of alternatives based on user preference, allowing the user to set up numerous sets of combinations. The second stage password can be specified as a stroke, a timer, the brightness of the screen, or a sensor-based authentication mechanism. Traditional password-based systems that incorporate biometric data can increase security by adding a second level of user authentication.

LIST OF FIGURES

LIST OF ABBREVATIONS

JDK	Java Development Toolkit
DEX	Dalvik Executables
TCP	Transmission Control Protocol
IP	Internet Protocol
HTTP	Hyper Text Transfer Protocol
ADT	Android Development Tool

LIST OF TABLES

CHAPTER-1

INTRODUCTION

1.1 OVERVIEW OF THE PROJECT

Mobile devices have become an indispensable tool for most people nowadays . Today's population relies heavily on mobile devices as a tool. In addition to the high level of technological advancement and newly added capabilities, new internet infrastructures like 5G, which enable real-time social media use and communication, have also been a driving force behind the rapid and continual deployment of mobile devices around the world. So, both the public and private sectors are conscious of the value that mobile devices have for society and are working to make their services available through convenient mobile applications that guarantee data security and privacy. Personal Identification Numbers (PIN) and One-Time Passwords have historically been the two user authentication methods that are most often used (OTP). In contrast to PIN-based authentication methods, which demand that users memorise their When the system chooses and provides the user with a different password each time, for example, when delivering messages to personal mobile devices or special tokens, OTP-based solutions prevent users from memorising personal passwords. Although PIN- and OTP-based authentication systems are widely used and widely accepted, numerousresearch have shown that these methods have drawbacks. First off, it's standard practise to choose passwords that are extremely easy to decipher, such as terms like "password" or "qwerty," or personal information like birth dates. Second, "smudge attacks"—whereby the deposition of finger grease marks on the touchscreen can be used—affect passwords entered on mobile devices like tablets or smartphones. for the imposters to figure out the password. However, "shoulder surfing" can also be a problem with password-based security. When the impostor has direct observational access or can gather user data through the use of external recording devices, this type of attack is generated. Because to the rising use of handheld recording devices and public monitoring infrastructures in recent years, this approach has caught the interest of many researchers. By combining ease and a high level of security, biometric recognition systems are able to meet these requirements. The conventional meet these requirements. The conventional authentication techniques are improved in this way. Online credit card payments are one application that inspires our suggested strategy. The user's mobile device receives a numerical password from the bank that is normally between 6 and 8 digits long. To complete the payment, the user must enter this numerical password into the security platform. By adding a second authentication factor based on the user's biometric data while drawing the digits, our suggested method improvessuch a scenario. A typical architecture of our suggested password-based mobile authentication method is shown in Fig. 1. In this study, the enrolment set, password generation, and touch biometric system—three major modules—are all examined. Depending on the intended use (i.e., PIN or OTP), In order to confirm the validity of the password, the handwritten digits can first be identified using, for instance, an Optical Character Recognition (OCR) system. Following this initial authentication stage, a second authentication stage compares the handwritten digits' biometric data to the claimed user's enrollment information, comparing each digit individually. Although the identification of numerical digits has been demonstratedto be a nearly solved problem with mistakes close to 0%, the second authenticationstage based on the user's activity information is the subject of this study. Becauseof this, we assume in this study that impostors successfully circumvent the first security measure (i.e., know the password). As a result, we assume in this study that impostors get beyond the first security checkpoint (i.e., they know the password of the target user) and that, absent our suggested strategy, the attack would be a complete success.

1.2 NEED OF THE PROJECT

“Efficient password mechanism to overcome spyware attacks” is to secure a mobile application with Password drawing pattern with various ways of implementation.Securing an ecommerce mobile application with password and drawing pattern is essential to protect sensitive information such as customer data, payment information, and order details. Without proper security measures in place, the ecommerce mobile application can become vulnerable to cyberattacks, leading to financial loss, data breaches, and reputational damage.There are several ways to implement password and drawing pattern security in an ecommerce mobile application.When implementing password or drawing pattern security in an ecommerce mobile application, it is important to ensure that the password ordrawing pattern is secure, encrypted, and stored in a safe manner. Additionally, themobile application should provide a way for users to reset their password ordrawing pattern in case it is forgotten or compromised. It is also recommended to provide users with information about best practices for creating strong passwords and protecting their accounts.

1.3 OBJECTIVE OF THE PROJECT

In this work we evaluate the potential of touch biometric verification systems based on time functions . Signals captured by the digitizer (i.e., X and Y spatial coordinates) are used to extract a set of 21 time functions for each numerical digit sample . Information related to pressure, pen angular orientations or pen upsbroadly used in other biometric traits such as handwriting and handwritten signature is not considered here as this information is not available in all mobile devices when using the finger touch as input. Sequential Forward Floating Search (SFFS) algorithm is used for the DTW algorithm in some of the experiments in order to select the best subsets of time functions for each handwritten digit and improve the system performance in terms of EER (%). So with the help of gesture what we drawn in the mobile , all the numbers patters will be stored in the backend. So whenever a user signin to the ecommerce application , this password security mechanism will be enacting with every product purchase.

1.4 SCOPE OF THE PROJECT

Our proposed system focus on providing user-friendly mobile applications ensuring data protection and high security. User should draw each digit of the password on the touch screen instead of typing them as usual. This way, the traditional authentication systems are enhanced by incorporating dynamic handwritten biometric information. Our system involves two stages of authentication the drawn pin should be similar to pin entered during registration process.

Our second stage of authentication involves multiple options based on user preference where user can set multiple set of combinations. User can set second stage password as stroke, time, screen brightness or sensor based authentication system. The incorporation of biometric information on traditional password-based systems can improve the security through a second level of user authentication.

CHAPTER 2

LITERATURE SURVEY

2.1 Benchmarking Touchscreen Biometrics for Mobile Authentication.

AUTHOR: Julian Fierrez, Ada Pozo

YEAR:2018

We study user interaction with touchscreens based on swipe gestures for personal authentication. In a series of disjointed and constrained works over the past few years, this strategy has only recently been examined. We review these recentefforts before contrasting them with three new systems that utilize independent processing of swipes based on orientation and are based on support vector machines and Gaussian mixture models using selected features from the literature. For the analysis, four public databases consisting of touch data obtained from gestures sliding one finger on the screen are used. We first analyse the contents of the databases, observing various behavioral patterns, e.g., horizontal swipes are faster than vertical independently of the device orientation. Next, we examine two scenarios: an intra-session one in which users are registered and authenticated on the same day, and an inter-session one in which registration and testing are done ondifferent days. The benchmarks and processed data that are produced are made available, enabling the key findings to be replicated using the score files and scripts that were supplied. The results provide new insights into the distinctiveness of swipe interaction in addition to the outstanding performance made possible by the proposed orientation-based conditional processing. For example, some gestures hold more user-discriminant information than others, data from landscape orientation is more stable, and horizontal gestures are generally more discriminative than vertical ones.

2.2 A New Multimodal Approach for Password Strength Estimation

AUTHOR: Javier Galbally, Iwen Coisel

YEAR:2017

One obvious conclusion can be derived from more than two decades of study in the area of password strength estimation: no single password strength metric is superior against all other metrics for every possible password. In this research, we present a novel multimodal strength metric that combines numerous flawed individual measures to take advantage of their strengths in order to overcome manyof their weaknesses, building on this certainty and also utilising the knowledge acquired in the field of information fusion. The final multimodal measure consists of many modules based on statistics and heuristics that, when combined, aresuccessful in giving real-time input regarding the "guess ability" of passwords. In acompanion publication, the validation process and test results are given and analysed.

2.3 Synchronous One Time Biometrics With Pattern Based Authentication.

AUTHOR: Patrick Lacharme, Christophe Rosenberger

YEAR:2016

In electronic transactions, one-time passwords are frequently used for authentication. A one-time password can be provided by a fraudster, therefore offering one is not actually a reliable authentication evidence. because a phoney might provide the token that generates the passwords. The use of biometric recognition is growing in order to address this issue. Even if biometric data are tightly linked with the user, their revocability nor diversity is conceivable, without an adequate post-processing. The BioHashing algorithm is one of the biometric template protection systems used to handle the underlying privacy and security concerns. Although used to safeguard a number of biometric modalities, these methods may not be suitable for all of them. In this paper, we propose a new protocol combining protected biometric data and a classical synchronous one time password to enhance the security of user authentication while preserving usability and privacy. Behavioral biometrics is used to provide a fast and a usable solution for users. We show through experimental results the efficiency of the proposed method.

2.4 Improving Accuracy, Applicability and Usability of Keystroke Biometrics on Mobile Touchscreen Devices.

AUTHOR: Alexander De Luca ,Florian Alt

YEAR:2015

Personal behavioural clues can help to improve authentication techniques. Particularly, physical keyboards have been extensively examined for user verification based on typing habits. The same ideas have been used on mobile touchscreens with few modifications thus far. The first study on mobile keystroke biometrics to compare touch-specific parameters across three distinct hand postures and evaluation frameworks is presented in this publication. Wedemonstrate that incorporating spatial touch features reduces implicit authentication equal error rates (EER) by 26.4 - 36.8% compared to the previously employed temporal features using 20.160 password entries from a research with 28 participants over two weeks. We also demonstrate that some hand positions work better for authentication than others. We additionally quantify the impact of common evaluation presumptions, such as known attacker data, training and testing on data from a single typing session, and fixed hand postures, in order to increase application and usability. We demonstrate how these methods can resultin unduly optimistic assessments. As a result, we discuss assessment suggestions, a probabilistic framework to deal with unidentified hand postures, and suggestions for future advancements.

2.5 Beware, Your Hands Reveal Your Secrets

AUTHOR: Diksha Shukla, Rajesh Kumar,Abdul Serwadda

YEAR:2014

Traditional research on assaults that use video-based side-channels to decode text entered on a smartphone included the assumption that the adversary could make use of some information from the screen display (say, a reflection of the screen ora low resolution video of the content typed on the screen). The PIN entry procedure on a smartphone is subjected to a novel sort of side-channel attack inthis research, which completely depends on the spatiotemporal dynamics of the hands during typing to decode the written text. The attack, which we implemented on a dataset of 200 films of the PIN entry procedure on an HTC One phone, resultsin the average PIN being broken over 50% of the time on the first try and over 85%of the time after ten tries. We believe that the attack is very likely to be used by adversaries who want to steal important private information covertly because itmay be carried out in a fashion that does not arouse suspicion The report urges thecommunity to look more closely at the dangers posed by the now-ubiquitous camera-enabled gadgets as users conduct an increasing amount of their computing activities on mobile devices in public.

2.6 Surveying the Development of Biometric User Authentication on Mobile Phones.

AUTHOR: Weizhi Meng ,Duncan S.Wong, Steven Furnell

YEAR:2014

To safeguard users' personal information and data, it is becoming more and more crucial to design reliable user authentication for mobile devices. Biometric approaches have gained popularity in both academics and industry because they provide numerous advantages over conventional authentication techniques. Authenticating legitimate users and spotting impostors based on physiological and behavioural traits is the main objective of biometric user authentication. We outline a taxonomy of ongoing projects for mobile biometric authentication and assess their viability for use with touch-enabled devices.Also, we investigate real attacks and potential defences on mobile phones while methodically characterising a generic biometric authentication system with eight potential attack points. Additionally, we offer a methodology for developing a trustworthy authentication system by properly incorporating multimodal biometric user authentication. When multimodal biometrics are used on touch-enabled phones, the false rate of a single biometric system can be drastically reduced. Experimental findings are shown to confirm this framework utilising touch dynamics. Finally, we identify obstaclesand open concerns in this field and predict that touch dynamics will become a mainstream feature in developing future user authentication on mobile phones.

2.7 The Quest To Replace Passwords: A Framework For Comparative Evaluation Of Web Authentication Schemes.

AUTHOR: Joseph Bonneau, Cormac Herley, Paul C.van Oorschot

YEAR: 2012

Using a wide range of twenty-five usability, deployability, and security advantages that an ideal scheme might provide, they assess twenty years' worth of ideas to replace text passwords for general-purpose user authentication on the web. We cover a wide range of approaches in our survey, including biometrics, hardware tokens, federated login protocols, graphical password schemes, cognitive authentication schemes, one-time passwords, and phone-assisted authentication. Our thorough methodology yields important insights regarding the difficulties of changing passwords. No known strategy even manages to keep the entire set of advantages that existing legacy passwords already offer, which means that none even comes close to giving all needed benefits. There is a wide range of schemes,in particular, from those that offer just minimal security gains over historicalpasswords to those that offer substantial security benefits.

2.8 Exploring Touch-Screen Biometrics for User Identification on Smart Phones.

AUTHOR: Julio Angulo,Erik Wastlund

YEAR:2012

Mobile smart devices are increasingly being used to store sensitive data and access online services. At the same time, mechanisms for user authentication onto their devices and online services are required. These techniques must be safe, private, and user-friendly. We describe our preliminary findings in this work about the usage of lock pattern dynamics as a convenient and secure two-factor authentication technique. With the purpose of gathering information about how people draw lock patterns on touchscreens, we created an app for the Android mobile operating system. This method achieves an average Equal Error Rate (EER) using a Random Forest machine learning classifier of about 10.39%, indicating that lock patterns biometrics can be used to identify users towards their device but may also pose a threat to privacy if the users' biometric information is handled without their consent.

2.9 Unobservable Reauthentication for Smartphones

AUTHOR: Lingjun Li,Xinxin Zhao, Guoliang Xue

YEAR:2013

New privacy and security concerns are being raised by the growing use of cellphones. Smartphones are becoming into individual network access points and have the potential to retain sensitive data. Due of its small size, a smartphone could be quickly stolen away and used by an assailant. Attackers can initiate impersonation attacks using the victim's smartphone, endangering the security of current networks, particularly online social networks. Thus, it is essential to create a system for smartphones that will periodically re-verify the identity of the current user and notify the owner. Such a system could aid in preventing smartphone theft and protecting the data kept on them. In this research, we present a unique biometric-based method for smartphones that allows for continuous and imperceptible re-authentication. The system compares the current user's finger movement patterns to the owner's finger movement patterns after learning the owner's finger movement patterns using a classifier. The technology keeps the current user authenticated without interfering with user-smartphone conversations. Tests demonstrate the effectiveness of our technology on smartphones and its excellent accuracy.

2.10 Multitouch Gesture-Based Authentication

AUTHOR: Ramabathina Girish Kumar ,priya

YEAR:2014

This study examines multitouch gestures for touch-sensitive devices' user authentication. By defining a canonical set of 22 multitouch gestures based on hand and finger movement traits. To assess the idea, two different investigations were conducted. To test the viability of multitouch gestures for user authentication, a single session experiment was first conducted. Tests on the canonical set revealed that the system could discriminate between movements made by various users with good performance. In addition, the studies indicated a desirable alignment of usability and security as movements that were more secure from a biometric point of view were assessed as more desirable in terms of convenience, pleasure, and excitement. Second, a three-session experiment for a study was conducted. The results show that biometric data obtained from a brief user-device contact persists during intervals of many days, but there is a substantial performance degradation when the authentication is carried out over several sessions. The study also demonstrated that among all gestures, user-defined gestures offer the highest recognition rates, although using numerous gestures in a sequence helps to increase verification accuracy.

2.11 Feasibility study

A feasibility study is carried out to select the best system that meet performance requirements. The main aim of the feasibility study activity is to determine that it would be financially and technically feasible to develop the product. The document provides the feasibility of the project that is being designed and lists various areas that were considered very carefully during the feasibility study of this project such as Economic feasibility, technical feasibility and Operational feasibility.

Economic Feasibility

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into there search and development of the system is limited. The expenditures must be justified. Thus, the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased. The fund required by the company for the research and development of the system. The expenditures faced by the company. For the developed system to be within the budget, the technologies which are freely available and the customized products are to be used.

The following are some of the important financial questions asked during preliminary investigation:

• The costs conduct a full system investigation.

• The cost of the hardware and software.

• The benefits in the form of reduced costs or fewer costly errors.

Technical Feasibility

The system must be evaluated from the technical point of view first. The assessment of this feasibility must be based on an outline design of the system requirement in the terms of input, output, programs and procedures. Having identified an outline system,the investigation must go on to suggest the type of equipment, required method developing the system, of running the system once it has been designed. The project should be developed such that the necessary functions and performance are achieved within the constraints. The project is developed within latest technology. Through the technology maybe come obsolete after some period of time, due to the fact that never version of same software supports older versions, the system may still be used. So, there are minimal constraints involved with this project. As the system has been developed using Java the project is technically feasible. This study is carried out to check the technical feasibility, that is, the technical Requirements of thesystem. Any system developed must not have a high demand on the available technical resources. This will lead to high demands being placed on the client. The developed system must have modest requirements, as only minimal or null changes are required for implementing this system.

CHAPTER-3

SYSTEM DESIGN

3.1 EXISTING SYSTEM

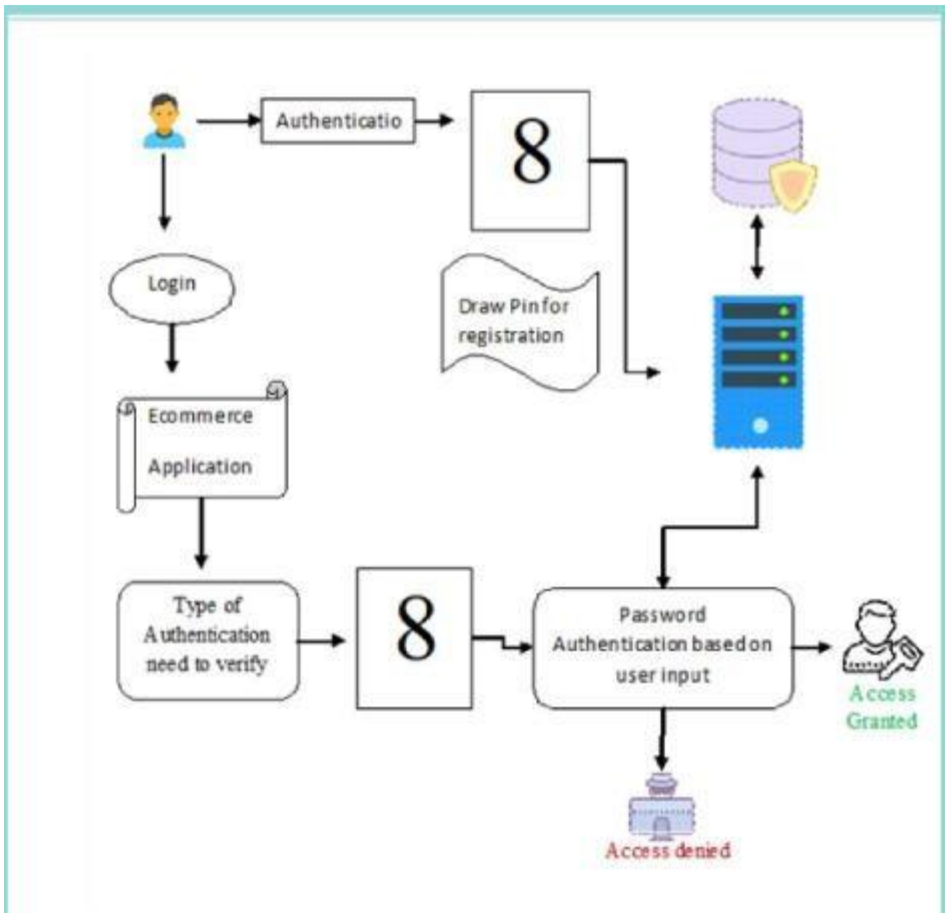
In existing system handwritten signature is one of the most socially accepted biometrics as it has been used in financial and legal agreements for many years and it also finds applications in mobile scenarios. These approaches are based on the combination of two authentication stages. The security system checks that the claimed user introduces its unique password correctly, and its behavioral biometric information is used for an enhanced final verification. The software for capturing handwritten numerical digits was developed in order to minimize the variability of the user during the acquisition process. The selection of a password that is robust enough for a specific application is a key factor. The number of digits that comprise the password depends on the scenario and level of security considered in the final application.

This effect has proven to be very important for many behavioral biometric traits such as the case of the handwritten signature.

3.1.1 Problem Definition:

- The amount of data requested to the user during the enrolment.
- The security level provided by the biometric system. From the point of view of the security system, it seems clear that the ideal case would be to have as much information of the user as possible.

3.2PROPOSED SYSTEM



Authenticatio
8
Login
Draw Pin for registration
Ecommerce Application
Type of Authentication need to verify
8
Password Authentication based on user input
Access Granted
Access denied

Fig:3.1 Architecture Diagram

Our proposed system focus on providing user-friendly mobile applications ensuring data protection and high security. User should draw each digit of the password on the touch screen instead of typing them as usual. This way, the traditional authentication systems are enhanced by incorporating dynamic handwritten biometric information. Our system involves two stages of authentication the drawn pin should be similar to pin entered during registration process.

Our second stage of authentication involves multiple options based on user preference where user can set multiple set of combinations. User can set second stage password as stroke, time, screen brightness or sensor based authentication system. The incorporation of biometric information on traditional password-based systems can improve the security through a second level of user authentication.

Advantages:

- These approaches enable active or continuous authentication schemes, in which the user is transparently authenticated.
- Handwritten signature is one of the most socially accepted biometrics.
- The incorporation of biometric information on traditional password-based systems can improve the security through a second level of user authentication.

3.2.1 Algorithm

OCR stands for Optical Character Recognition, which is a technology that allows computers to recognize printed or handwritten text characters from digital images of physical documents or scanned paper documents. OCR algorithms are designed to analyze the image of the text characters, identify the patterns in the characters, and convert them into machine-encoded text that can be used for further processing.

Working Of OCR

The basic steps involved in the OCR algorithm used in the above code:

- Load the image: The input image containing the text is loaded into the program.
- Preprocessing: The image is preprocessed by converting it to grayscale, applying noise reduction techniques, and adjusting the contrast to enhance the quality of the image.

- Text detection: The program detects the regions of the image where text is present using various techniques like the Stroke Width Transform (SWT), Edge Detection, or Connected Component Analysis (CCA).
- Text Recognition: The detected text regions are then processed using Optical Character Recognition (OCR) algorithms, which extract the text from the image and convert it into machine-readable text.
- Postprocessing: The extracted text is post-processed to correct errors and improve the accuracy of the OCR output. Techniques like language models, spell-checking, and grammar checking can be used for this purpose.
- Output: The final output is the extracted and processed text, which can be used for further analysis or storage.

3.3 Modules

- User Authentication and Ecommerce View Product
- Cart and Payment Using Biometric Hand Written Password
- Biometric Password Using Strokes
- Biometric Password Using Screen Brightness and Time

3.3.1 User Authentication and Ecommerce View Product

User has an initial level Registration Process. The users provide their own personal information for this process. The server in turn stores the information in its database and user can view a list of products in their page multiple list of products and their details.

3.3.2 Cart and Payment Using Biometric Hand Written Password

User can select a list of product they wish to purchase the selected product will be listed in a cart page and user can initiate general purchase information has to be filled. Completing general detail user has to draw their four digit pin one by one on screen. The drawn password then converted into an image through optical character recognition numbers from each image fetched and verified with user password.

3.3.3 Biometric Password Using Strokes

User has to register their four digit password with multiple strokes during their registration process once the process completed during confirm password .User has to confirm their password with same password with stroke has to be verified. Strokes for each drawn digits should match with strokes given at time of registration.

3.3.4 Biometric Password Using Screen Brightness and Time

Spyware attack will be avoided by proposing the idea that uses the screen brightness as an authentication tool. The android secure environment generates the 6 digit binary value. Based on the binary digit the brightness of the screen gets changed to high or low. If the screen brightness is high the user should input the correct PIN digit. Else the user should give the wrong and random PIN number. The system will remove the digits which inserted while the screen brightness is low and apply the HMac algorithm for the PIN given by user and generate the Signature for the user PIN which is a digestible Value in order to avoid MAN-IN- MIDDLE attack. The server gets the signature of user generated PIN and generates
the signature value for the Original PIN and compare two signatures. If the two

Signatures are equal the user can access the Profile of the user. If not user can't access the profile.

3.4 System Feature

Here we concentrate on password mechanism with user drawing patterns. The user need to register their pin number in the application with proper strokes. All the pin numbers with proper strokes will be getting stored in database server.

We implement random changes in screen brightness for securing the user's confidential passwords from hackers. We use ultrasound for OTP transfer , so that OTP will be sent to a host mobile from there it send that data through sonic waves to the user's mobile. Thus security mechanism for user drawing patterns in android application.

3.5 UML Diagrams

3.5.1 Use Case Diagram:

Unified Modeling Language (UML) is a standardized general-purpose modeling language in the field of software engineering.

Use case diagram consists of two parts:

Use case: A use case describes a sequence of actions that provided something of measurable value to an actor and is drawn as a horizontal ellipse.

Actor: An actor is a person, organization or external system that plays a role in one or more interaction with the system.

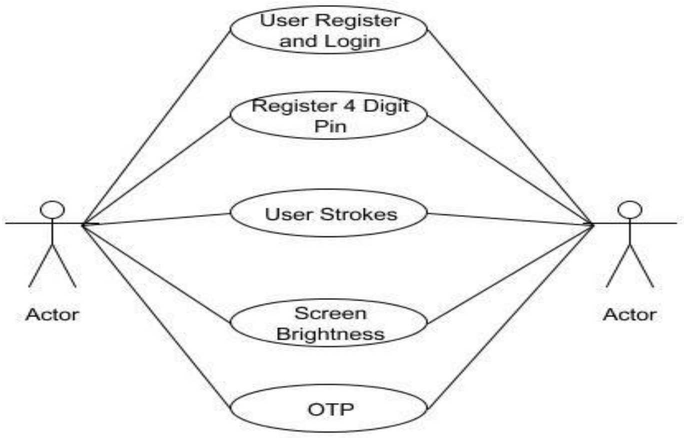


Fig:3.2 Use case Diagram

3.5.2 Activity Diagram:

Activity diagram is a graphical representation of workflows of stepwise activities and actions with support for choice, iteration and concurrency. An activity diagram shows the overall flow of control.

The most important shape types:

- Rounded rectangles represent activities.
- Diamonds represent decisions.
- Bars represent the start or end of concurrent activities.
- A black circle represents the start of the workflow.
- An encircled circle represents the end of the workflow.

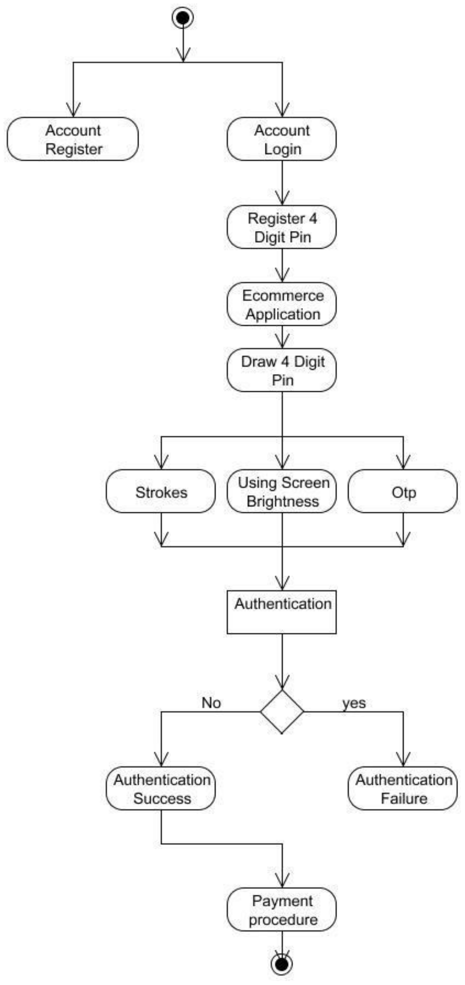
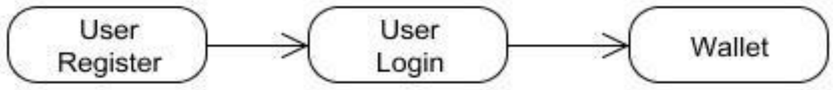


Fig:3.3 Activity Diagram

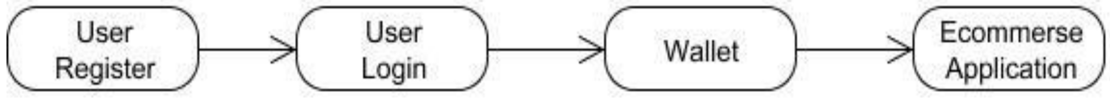
3.5.3 Data Flow Diagram:

A Data Flow Diagram (DFD) is a graphical representation of the "flow" of data through an information system, modeling its aspects. It is a preliminary step usedto create an overview of the system which can later be elaborated DFDs can alsobe used for visualization of data processing.

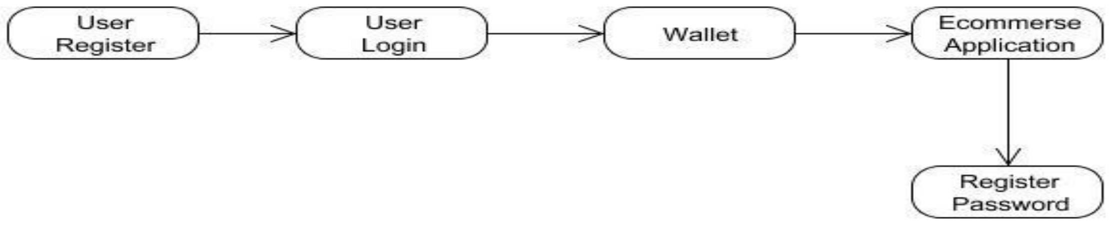
Level 0:



Level 1:



Level 2:



Level 3:

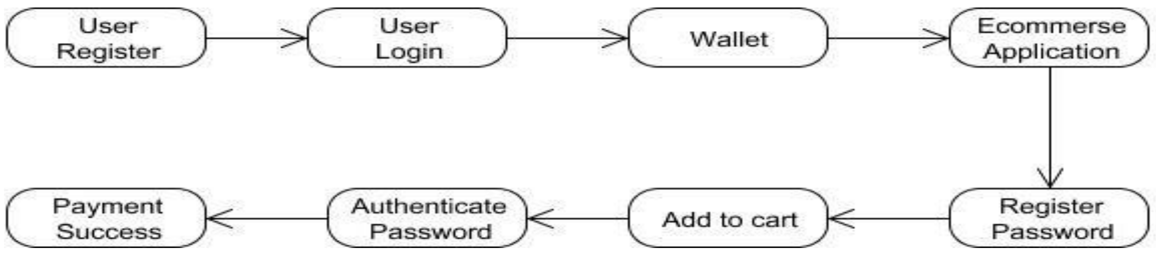


Fig:3.4 Data flow diagram

3.5.4 Class Diagram

A Class diagram in the Unified Modeling Language is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects.

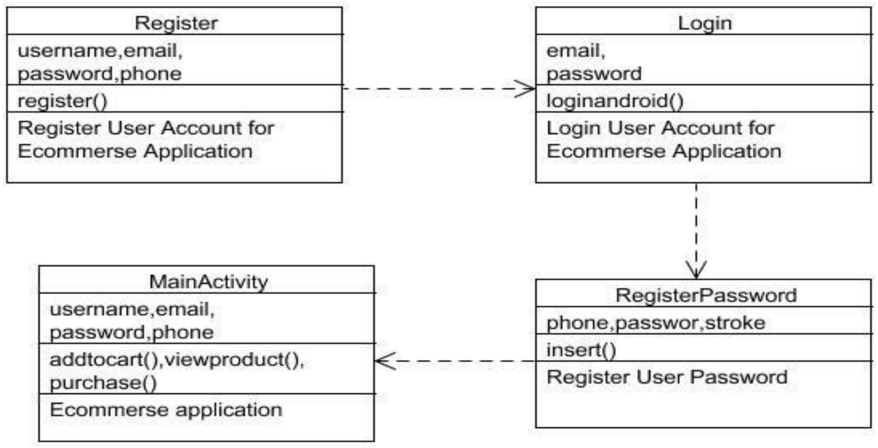


Fig: 3.5 Class Diagram

CHAPTER-4

REQUIREMENT SPECIFICATION

4.1 HARDWARE REQUIREMENTS

The hardware requirements may serve as the basis for a contract for the implementation of the system and should therefore be a complete and consistent specification of the whole system. They are used by software engineers as the starting point for the system design. It shows what the system does and not how it should be implemented.

- PENTIUM IV 2.6 GHz, Intel Core 2 Duo.
- 4GB DD RAM
- LAPTOP 15" COLOR
- Hard Disk:40 GB
- Android Mobile Phone

4.2 HARDWARE REQUIREMENTS DESCRIPTION

4.2.1 PENTIUM IV 2.6 GHz, Intel Core 2 Duo.

Pentium 4 is a series of single-core CPUs for desktops, laptops and entry level servers manufactured by Intel. The processors were shipped from November 20, 2000 until August 8, 2008. All Pentium 4 CPUs are based on the Net Burst micro architecture. The Pentium 4 Willamette (180 nm)introduced SSE2, while the Prescott (90 nm) introduced SSE3. Later versions introduced Hyper Threading Technology (HTT).

4.2.2 4GB DD RAM

The PC has a 64-bit Windows 10 operating system (OS), at least 4GB of memory is a must. You can easily get by with 4GB as long as you aren't playing advanced games and tackling large data files. Of course, it wouldn't hurt to jump up to 8GB if you want your computer to run as smoothly as possible.

4.2.3 LAPTOP 15"

This 15 inch Full HD laptop delivers best-in-class picture quality with excellent 178° viewing angles. The 15 inch laptop has a lightweight housing and is ideal for working. The 15HD7 can be connected via HDMI, VGA, BNC or RCA connections

4.2.4 HARD DISK 40 GB

Hard disks are flat circular plates made of aluminium or glass and coated with a magnetic material. Hard disks for personal computers can store terabytes (trillions of bytes) of information. Data are stored on their surfaces in concentric tracks. The hard drive is where all your permanent computer data is stored.Whenever you save a file, photo, or software to your computer, it's stored in your hard drive.

4.2.5 ANDROID MOBILE PHONE

Android is a mobile operating system based on a modified version of the linux kernel and other open-source software, designed primarily for touchscreen mobile devices such as smartphones and tablets.at its core, the operating system is known as android open source project and is free and open-source software (foss) primarily licensed under the apache license. however most devices run on the proprietary android version developed by google, which ship with additional proprietary closed-source software pre-installed,most notably google mobile services which includes core apps such as google chrome, the digital distribution platform google play, and the associated google play services development platform

4.3 SOFTWARE REQUIREMENTS

- Windows 7 and above
- JDK 1.7
- Xampp
- Android Studio 3.4

4.4 SOFTWARE REQUIREMENTS DESCRIPTION

4.4.1 JDK

The JDK is a development environment for building applications, applets, and components using the Java programming language. The JDK includes tools useful for developing and testing programs written in the Java programming language and running on the Java platform.

Features:

- Binary Literals.
- Strings in switch Statements.
- The try-with-resources Statement.
- Catching Multiple Exception Types and Rethrowing Exceptions with Improved Type Checking.
- Underscores in Numeric Literals.
- Type Inference for Generic Instance Creation.

4.4.2 XAMPP:

XAMPP helps a local host or server to test its website and clients via computers and laptops before releasing it to the main server. It is a platform that furnishes a suitable environment to test and verify the working of projects based on Apache, Perl, MySQL database, and PHP through the system of the host itself.XAMPP is an open source package that is widely used for PHP development. XAMPP contains MariaDB, PHP, and Perl; it provides a graphical interface for SQL (phpMyAdmin), making it easy to maintain data in a relational database.

4.4.3 ANDROID STUDIO

Android Studio provides a selection of code samples and templates for you to use to accelerate your app development. Browse sample code to learn how to build different components for your apps. Use templates to create new app modules, individual activities, or other specific Android project components.Android Studio is the official integrated development environment (IDE) for Android app development built and distributed by Google. An IDE contains tools that make it easy for software developers to design, build, run, and test software, in this case, apps for the Android platform.Android Studio is the official integrated development environment (IDE) for Google's Android operating system, built on JetBrains' IntelliJ IDEA software and designed specifically for Android development. It is available for download on Windows, macOS and Linux based operating systems.

4.5 TECHNOLOGIES USED

- Core java
- Android
- PHP

JAVA

Java is an object-oriented programming language developed initially by James Gosling and colleagues at Sun Microsystems. The language, initially called Oak (named after the oak trees outside Gosling's office), was intended to replace C++, although the feature set better resembles that of Objective C.

Working of java

For those who are new to object-oriented programming, the concept of a class will be new to you. Simplistically, a class is the definition for a segment of code thatcan contain both data (called attributes) and functions (called methods). When the interpreter executes a class, it looks for a particular method by the name of **main,** which will sound familiar to C programmers. The main method is passed as a parameter an array of strings (similar to the argv [] of C), and is declared as a static method.

To output text from the program, we execute the **println** method of **System.out,** which is java's output stream. UNIX users will appreciate the theory behind such a stream, as it is actually standard output. For those who are instead used to the Wintel platform, it will write the string passed to it to the user's program.

PHP

PHP is mainly focused on server-side scripting, so you can do anything any other CGI program can do, such as collect form data, generate dynamic page content, or send and receive cookies. But PHP can do much more. There are three main areas where PHP scripts are used. Server-side scripting.It is integrated with a number of popular databases, including MySQL, PostgreSQL, Oracle, Sybase, Informix, and Microsoft SQL Server.

ANDROID

Android is a Linux based operating system it is designed primarily for touch screen mobile devices such as smart phones and tablet computers. The operating system has developed a lot in last 15 years starting from black and white phones to recent smart phones or mini computers. One of the most widely used mobile OS these days is android. The android is software that was founded in Palo Alto of California in 2003.The android is a powerful operating system and it supports largenumber of applications in Smartphones. These applications are more comfortable and advanced for the users. The hardware that supports android software is based on ARM architecture platform. The android is an open source operating system means that it’s free and any one can use it. The android has got millions of apps available that can help you managing your life one or other way and it is available low cost in market at that reasons android is very popular.The androiddevelopment supports with the full java programming language. Even other packages that are API and JSE are not supported. The first version 1.0 of android development kit (SDK) was released in 2008 and latest updated version is jelly bean.

CHAPTER 5

IMPLEMENTATION

5.1 CODING

File Dao.java

```
package logic;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.util.ArrayList;

import java.util.List;

public class FileDao

{

public static Connection getConnection(){

Connection con=null;

try{

Class.forName("com.mysql.jdbc.Driver");

con=DriverManager.getConnection("jdbc:mysql://localhost/biotouchpass", "root",
"root");

//System.out.println("------------CLOUD USER CREATED ----------- ");

}catch(Exception e){System.out.println(e);}

return con;

}        public static boolean ownerCheck(OwnerPojo op) throws SQLException

{

boolean emailexists=false;
```

```
Connection con=FileDao.getConnection();
String query="select * from users where email = ?";
PreparedStatement st = con.prepareStatement(query);
st.setString(1, op.getEmail());
ResultSet rs=st.executeQuery();
if(rs.next()) {
emailexists = true;
}        return emailexists;
}public static int ownerSave(OwnerPojo op){
int status=0;a
try{
Connection con=FileDao.getConnection();
String query="insert into users(username,email,password,phone) values (?,?,?,?)";
PreparedStatement ps=con.prepareStatement(query);
ps.setString(1, op.getUsername());
ps.setString(2, op.getEmail());
ps.setString(3, op.getPassword());
ps.setString(4, op.getPhone());
status=ps.executeUpdate();
con.close();
}catch(Exception ex){ex.printStackTrace();}
return status;
}public static String ownerLogin(OwnerPojo olog)
{
```

```
String status="";
try {
Connection con=FileDao.getConnection();
String email="";
String password="";
String name="";
String query="select * from users where email='"+olog.getEmail()+"'";
PreparedStatement st = con.prepareStatement(query);
ResultSet rs=st.executeQuery();
while(rs.next())
{
//   name=rs.getString("name");
email=rs.getString("email");
password=rs.getString("password");
// phone=rs.getString("password");
System.out.println(email+"     "+password);
System.out.println(olog.getEmail()+"       "+olog.getPassword());
}
```

Login Android.java

```
  package logic;
import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.RequestDispatcher;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import org.json.simple.JSONObject;
public class LoginAndroid extends HttpServlet {
public void doPost(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
response.setContentType("text/html");
PrintWriter out = response.getWriter();
String email=request.getParameter("username");
String password=request.getParameter("password");
String[] detail;
JSONObject jsonObject = new JSONObject();
JSONObject jsonObject2 = new JSONObject();
OwnerPojo olog=new OwnerPojo();
olog.setEmail(email);
olog.setPassword(password);
try {
String status=FileDao.ownerLogin(olog);
if(!status.equals("no"))
{
String userdetails=FileDao.LoginDetails(email);
detail = userdetails.split(",");
jsonObject2.put("username", detail[0]);
jsonObject2.put("email", detail[1]);
jsonObject2.put("password", detail[2]);
jsonObject2.put("phone", detail[3]);
jsonObject.put("error", "false");
jsonObject.put("message", "Login Successfull");
jsonObject.put("user", jsonObject2);
}
else
{
jsonObject.put("error", "true");
jsonObject.put("message", "Login failed");
}
out.print(jsonObject);
} catch (Exception e) {
// TODO: handle exception
}}}
```

Register Android.java

```
package logic;
import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.RequestDispatcher;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import org.json.simple.JSONObject;
public class RegisterAndroid extends HttpServlet {
public void doPost(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
response.setContentType("text/html");
PrintWriter out = response.getWriter();
String username = request.getParameter("username");
String email = request.getParameter("email");
String password = request.getParameter("password");
String phone = request.getParameter("phone");
String[] detail;
JSONObject jsonObject = new JSONObject();
JSONObject jsonObject2 = new JSONObject();
OwnerPojo op=new OwnerPojo();
op.setUsername(username);
op.setEmail(email);
op.setPassword(password);
op.setPhone(phone);
try {
boolean emailexists=FileDao.ownerCheck(op);
if(emailexists==true)
{
jsonObject.put("error", "false");
jsonObject.put("message", "email already exist");
}
else
{
int status=FileDao.ownerSave(op);
if(status>0)
{
String userdetails=FileDao.LoginDetails(email);
detail = userdetails.split(",");
jsonObject2.put("username", detail[0]);
jsonObject2.put("email", detail[1]);
jsonObject2.put("password", detail[2]);
jsonObject2.put("phone", detail[3]);
jsonObject.put("error", "false");
jsonObject.put("message", "Registration Successfull");
jsonObject.put("user", jsonObject2);
}
else {
jsonObject.put("error", "true");
jsonObject.put("message", "Registration faild");
}}
out.print(jsonObject);
} catch (Exception e) {
// TODO: handle exception
```

```
}}
```

Check UPI.java

```
package logic;
import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import org.json.simple.JSONArray;
import org.json.simple.JSONObject;
public class CheckUpi extends HttpServlet {
public void doPost(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
response.setContentType("text/html");
PrintWriter out = response.getWriter();
String phone = request.getParameter("phone");
OwnerPojo ownerPojo = new OwnerPojo();
ownerPojo.setPhone(phone);
JSONObject jsonObject = new JSONObject();
JSONArray jsonArray = new JSONArray();
JSONObject users = new JSONObject();
try {
boolean upiexists=FileDao.upiCheck(ownerPojo);
if(upiexists==true)
{
users.put("upiid",phone+"@okbank");
jsonObject.put("error", "false");
jsonObject.put("message", "upiid");
jsonObject.put("users",users);
//out.print("");
}
else {
jsonObject.put("error", "true");
jsonObject.put("message", "noupi");
}
out.print(jsonObject);
} catch (Exception e) {
// TODO: handle exception
}}}
```

Enter Password.java

```
package logic;
import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import org.json.simple.JSONObject;
public class EnterPassword extends HttpServlet {
public void doPost(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
response.setContentType("text/html");
PrintWriter out = response.getWriter();
String uid= request.getParameter("uid");
String[]pass;
JSONObject jsonObject = new JSONObject();
JSONObject jsonObject2 = new JSONObject();
try {
String passstr=FileDao.password(uid);
if(!passstr.equalsIgnoreCase("")){
System.out.println("----- "+passstr);
pass = passstr.split(",");
jsonObject2.put("password", pass[0]);
jsonObject2.put("stroke", pass[1]);
jsonObject.put("error", "false");
jsonObject.put("message", "password");
jsonObject.put("user",jsonObject2);
}
else {
jsonObject.put("error", "true");
jsonObject.put("message", "password not registered");
}
out.print(jsonObject);
} catch (Exception e) {
// TODO: handle exception
}}}
```

Ip Address.java

```
package com.example.handwrittenpassword;
public class IpAddress {
 public static String Ip_Address;
}
```

File Upload.java

```
package logic;
import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
public class FileUpload extends HttpServlet {
public void doPost(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
response.setContentType("text/html");
PrintWriter out = response.getWriter();
String test = request.getParameter("hash");
System.out.println(test);
}}
```

Password.java

```
package logic;
import java.io.IOException;
import java.io.PrintWriter;
import java.sql.SQLException;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import org.json.simple.JSONObject;
public class Password extends HttpServlet {
public void doPost(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
response.setContentType("text/html");
PrintWriter out = response.getWriter();
String phone =  request.getParameter("phone");
String password = request.getParameter("password");
String stroke =  request.getParameter("stroke");
String upiid = phone+"@okbank";
JSONObject obj = new JSONObject();
OwnerPojo op=new OwnerPojo();
op.setPhone(phone);
op.setPassword(password);
op.setStroke(stroke);
op.setUpiid(upiid);
try {
boolean emailexists=FileDao.ownerCheck(op);
if(emailexists==true)
{
obj.put("error", "true");
obj.put("message", "upi id generated");
//out.print("");
}
else
{
int status=FileDao.ownerSave(op);
if(status>0)
{
obj.put("error", "false");
obj.put("message", "Register Success");
}
else {
obj.put("error", "true");
obj.put("message", "Error");
}}
out.print(obj);
} catch (SQLException e) {
e.printStackTrace();
}
out.close();
}}
```

Screen Brightness.java

```
package com.example.handwrittenpassword;
import android.content.Context;
import android.content.Intent;
import android.content.SharedPreferences;
import android.graphics.Bitmap;
import android.net.Uri;
import android.os.Build;
import android.os.Bundle;
import android.provider.Settings;
int count = 0;
//public ArrayList<Integer> stroke;
//public ArrayList<String> password;
String[] upassword;
String[] ustroke;
HashMap<Integer,String> password;
HashMap<Integer,Integer> stroke;
HashMap<Integer,Integer> hstroke;
HashMap<Integer,String> hpassword;
```

```
int brightcount = 0;

int scount = 0;

ArrayList<String>isnum;

int strokeCount =0;

public String uemail;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity_register_password);

FirebaseApp.initializeApp(this);

SharedPreferences sharedPreferences =
getSharedPreferences("mypref",MODE_PRIVATE);

uemail = sharedPreferences.getString("phone",null);

insert();

mSignaturePad = (SignaturePad) findViewById(R.id.signature_pad);

password = new HashMap<>();

stroke = new HashMap<>();

hstroke = new HashMap<>();

hpassword = new HashMap<>();

isnum = new ArrayList<>();

isnum.add("1");

isnum.add("2");

isnum.add("3");

isnum.add("4");
isnum.add("5");
isnum.add("6");
isnum.add("7");
isnum.add("8");
isnum.add("9");
isnum.add("0");
settingPermission();
if(rannumbers[bright].equalsIgnoreCase("2")){
changeScreenBrightness(getApplicationContext(),10);
}
else if(rannumbers[bright].equalsIgnoreCase("4")){
changeScreenBrightness(getApplicationContext(),10);
}
else {
changeScreenBrightness(getApplicationContext(),120);
}
```

UPI.java

```
package  com.example.handwrittenpassword;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
public class Upi extends AppCompatActivity {
@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_upi);
}
}
```

Update Amount.java

```
package logic;
import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import org.json.simple.JSONObject;
public class UpdateAmount extends HttpServlet {
public void doPost(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
response.setContentType("text/html");
PrintWriter out = response.getWriter();
String accountno = request.getParameter("accountno");
String amount = request.getParameter("rem");
JSONObject jsonObject = new JSONObject();
FilePojo ftype = new FilePojo();
ftype.setAmount(amount);
ftype.setAccountno(accountno);
try {
```

```
int uamount = FileDao.updateamount(ftype);
if(uamount>0){
jsonObject.put("error", "false");
jsonObject.put("message", "amount updated");
}else {
jsonObject.put("error", "true");
jsonObject.put("message", "error");
}
out.print(jsonObject);
} catch (Exception e) {
// TODO: handle exception
}}}
```

Wallet.java

```
package logic;
import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import org.json.simple.JSONObject;
import sun.awt.SunHints.Value;
public class Wallet extends HttpServlet {
public void doPost(HttpServletRequest request, HttpServletResponse response)
```

```
throws ServletException, IOException {
response.setContentType("text/html");
PrintWriter out = response.getWriter();
String email = request.getParameter("email");
System.out.println("====="+email);
JSONObject jsonObject = new JSONObject();
JSONObject jsonObject2 = new JSONObject();
String[] values;
try {
String data = FileDao.wallet(email);
values = data.split(",");
System.out.println("---- "+values[0]);
if(!values[0].equalsIgnoreCase("")){
jsonObject.put("acccountno", values[0]);
jsonObject.put("mailid", values[1]);
jsonObject.put("name", values[2]);
jsonObject.put("amount", values[3]);
jsonObject2.put("user", jsonObject);
jsonObject2.put("error", "false");
jsonObject2.put("message", "wallet");
}
else {
jsonObject2.put("error", "false");
jsonObject2.put("message", "error");
}
out.print(jsonObject2);
} catch (Exception e) {
// TODO: handle exception
}}}
```

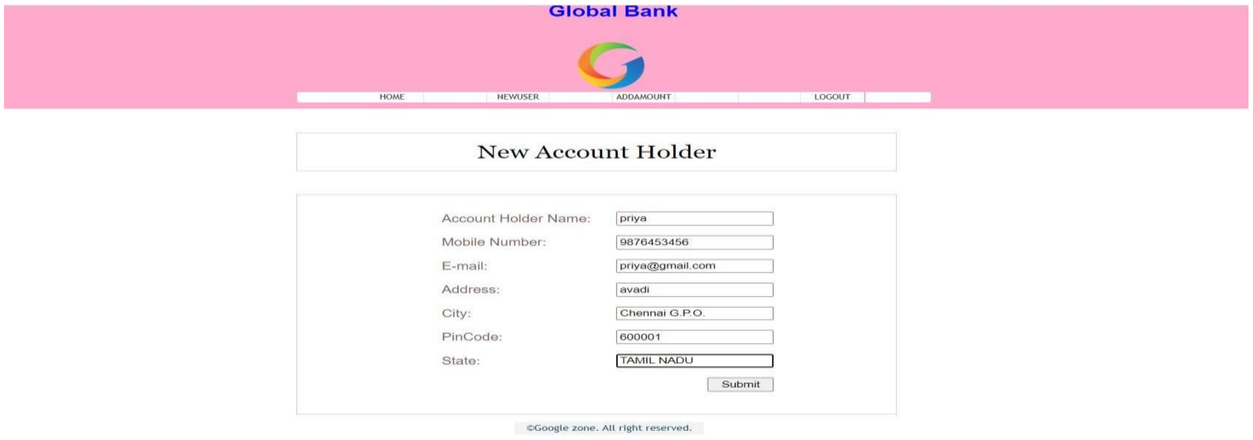
Payment Successfil.java

```
package com.example.handwrittenpassword;
import android.content.Intent;
import android.content.SharedPreferences;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;
import android.widget.Toast;
import com.android.volley.AuthFailureError;
import com.android.volley.Request;
import com.android.volley.Response;
import com.android.volley.VolleyError;
/*    public  String insert(){
StringRequest stringRequest = new StringRequest(Request.Method.POST,
URLs.WALLET,
new Response.Listener<String>() {
```

```
@Override

public void onResponse(String response) {

try {

//converting response to json object

JSONObject obj = new JSONObject(response);

Log.e("test",response);

//if no error in response

if (!obj.getBoolean("error")) {

Toast.makeText(getApplicationContext(), obj.getString("message"),
Toast.LENGTH_SHORT).show();

JSONObject userJson = obj.getJSONObject("user");

accountno = userJson.getString("acccountno");

aamount = userJson.getString("amount");

SharedPreferences sharedPreferences =
getSharedPreferences("mpref",MODE_PRIVATE);

SharedPreferences.Editor editor = sharedPreferences.edit();

editor.putString("acc",accountno);

editor.apply();

String mailid = userJson.getString("mailid");

String aname = userJson.getString("name");

} else {

Toast.makeText(getApplicationContext(), obj.getString("message"),
Toast.LENGTH_SHORT).show();

}

} catch (JSONException e) {
```

```
e.printStackTrace();
}}},
new Response.ErrorListener() {
@Override
public void onErrorResponse(VolleyError error) {
Toast.makeText(getApplicationContext(), error.getMessage(),
Toast.LENGTH_SHORT).show();
}
})
```

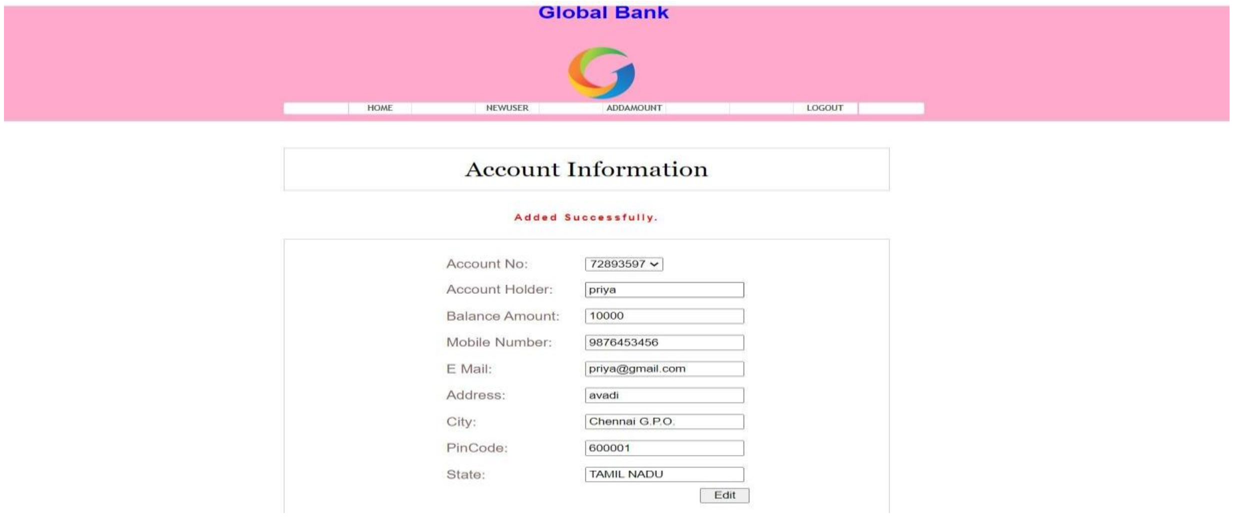
5.2 SCREENSHOTS



New Account Holder

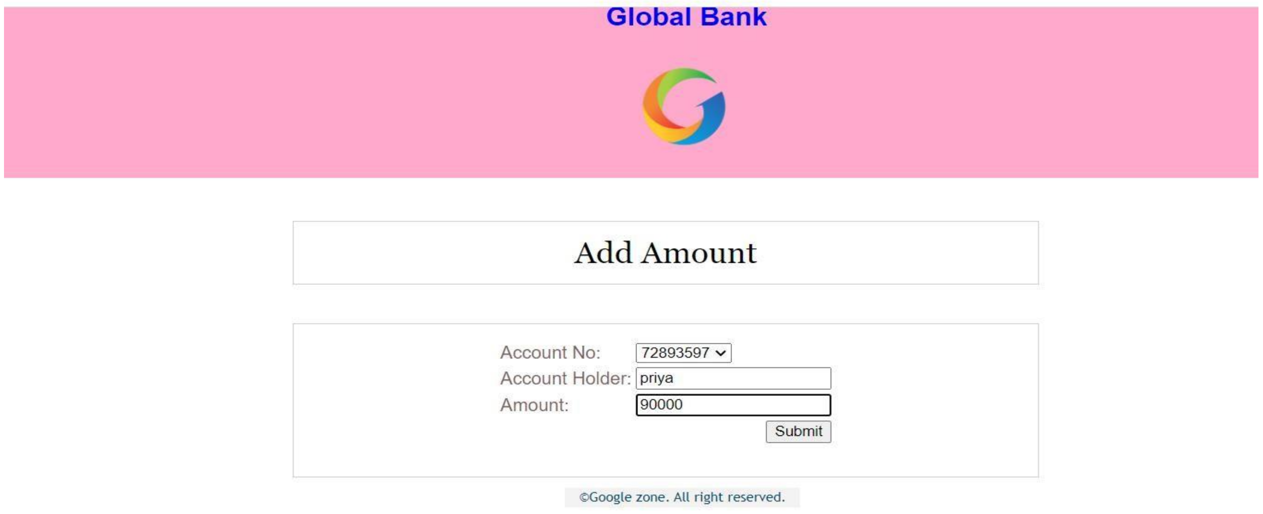
Account Holder Name:	priya
Mobile Number:	9876453456
E-mail:	priya@gmail.com
Address:	avadi
City:	Chennai G.P.O.
PinCode:	600001
State:	TAMIL NADU

Fig:5.1 Create Account



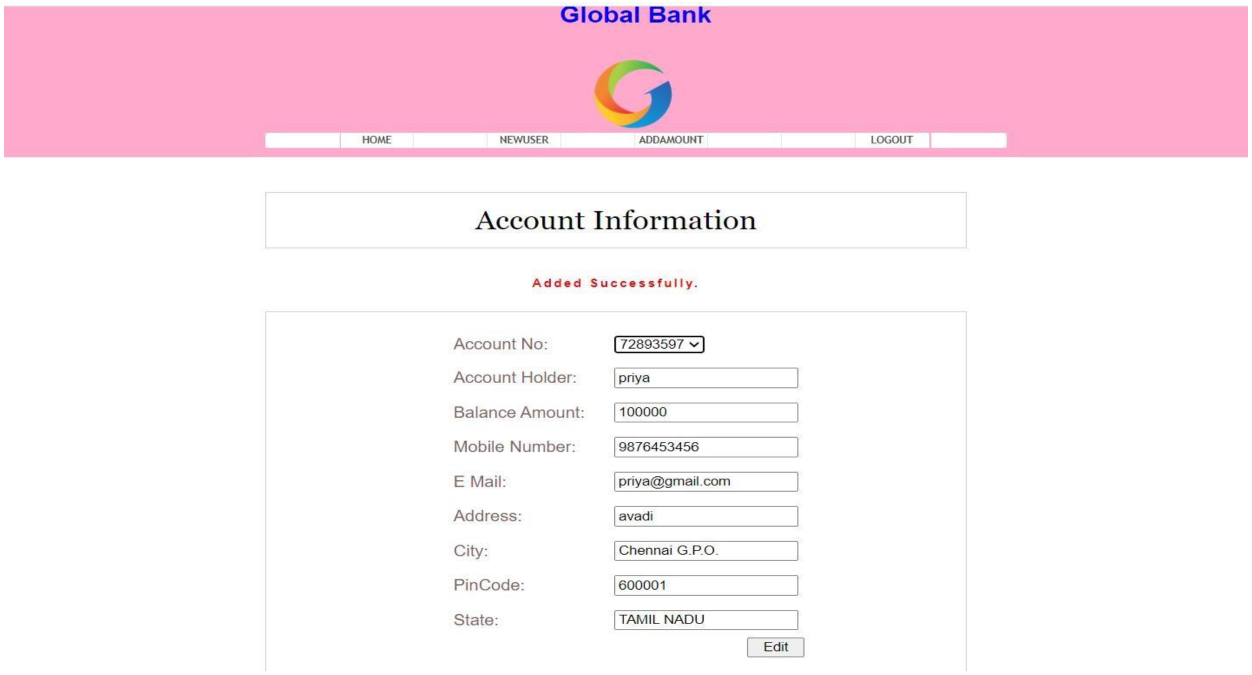
Account No:	72893597
Account Holder:	priya
Balance Amount:	10000
Mobile Number:	9876453456
E Mail:	priya@gmail.com
Address:	avadi
City:	Chennai G.P.O.
PinCode:	600001
State:	TAMIL NADU

Fig:5.2 Account Information



Account No:	72893597
Account Holder:	priya
Amount:	90000

Fig: 5.3 Amount Adding



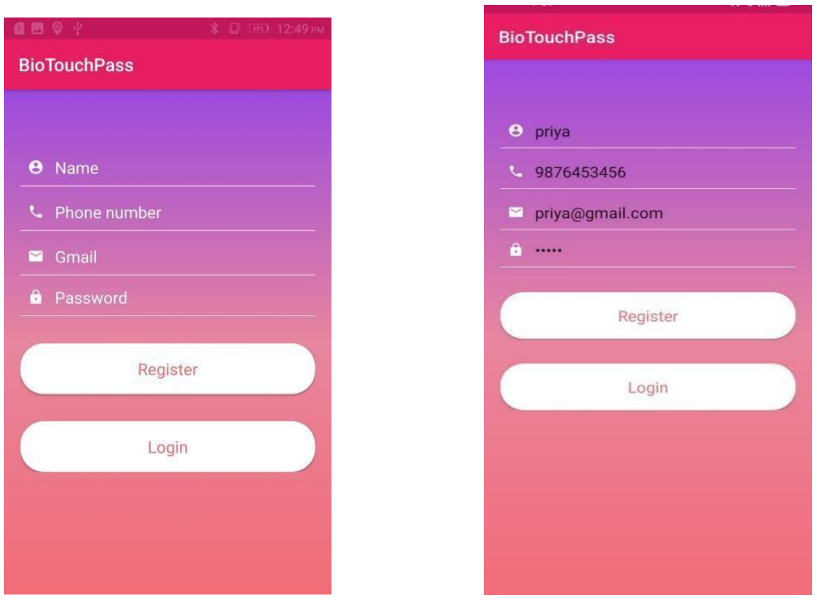
Global Bank

Account Information

Added Successfully.

Account No:	72893597
Account Holder:	priya
Balance Amount:	100000
Mobile Number:	9876453456
E Mail:	priya@gmail.com
Address:	avadi
City:	Chennai G.P.O.
PinCode:	600001
State:	TAMIL NADU

Fig:5.4 After Adding Amount



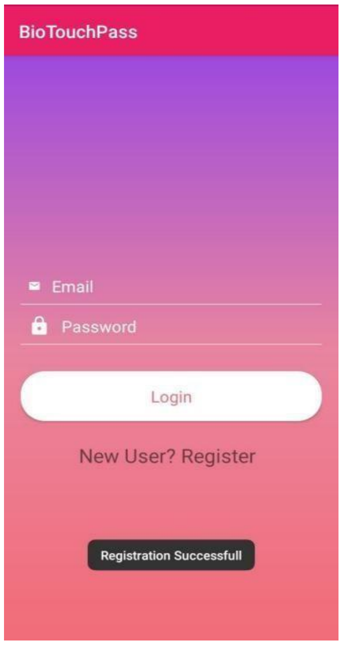
BioTouchPass

Name
Phone number
Gmail
Password

BioTouchPass

priya
9876453456
priya@gmail.com
•••••

Fig:5.5 Register



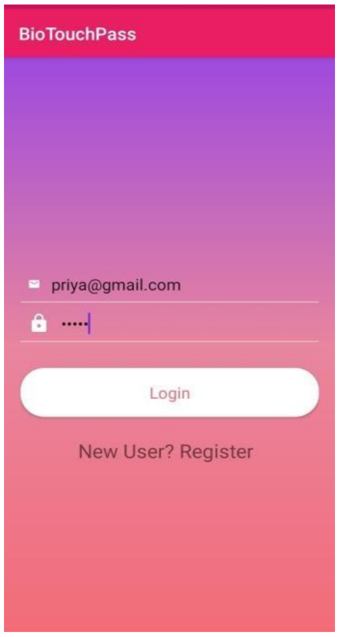
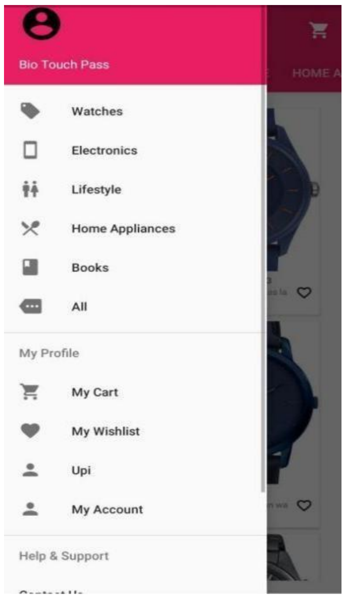
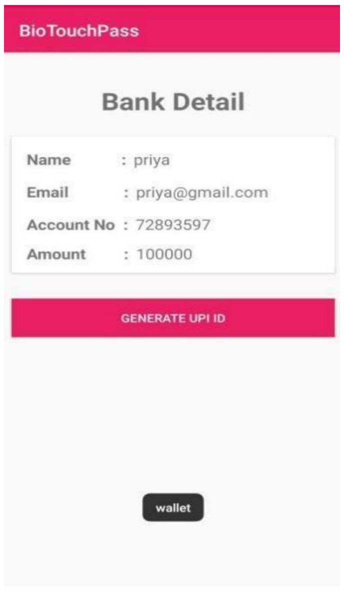


Fig:5.6 Login





Name : priya
Email : priya@gmail.com
Account No : 72893597
Amount : 100000

Fig:5.7 UPI Generation

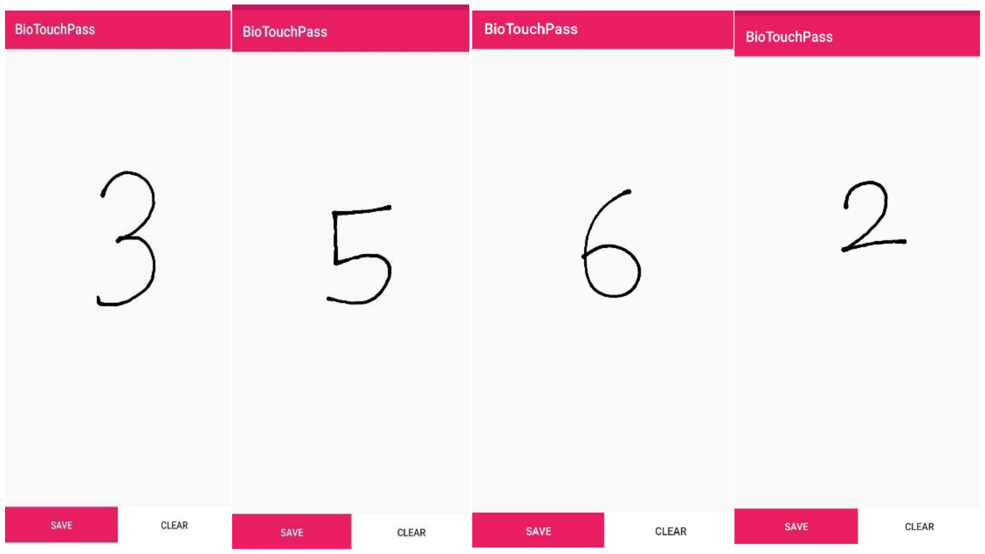
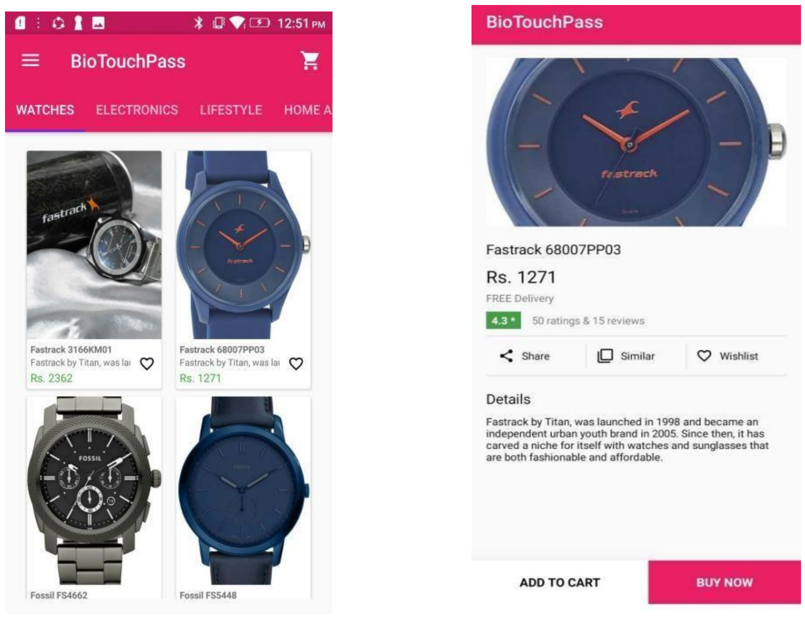


Fig:5.8 Create UPI



Fastrack by Titan, was launched in 1998 and became an independent urban youth brand in 2005. Since then, it has carved a niche for itself with watches and sunglasses that are both fashionable and affordable.

Fig:5.9 Product

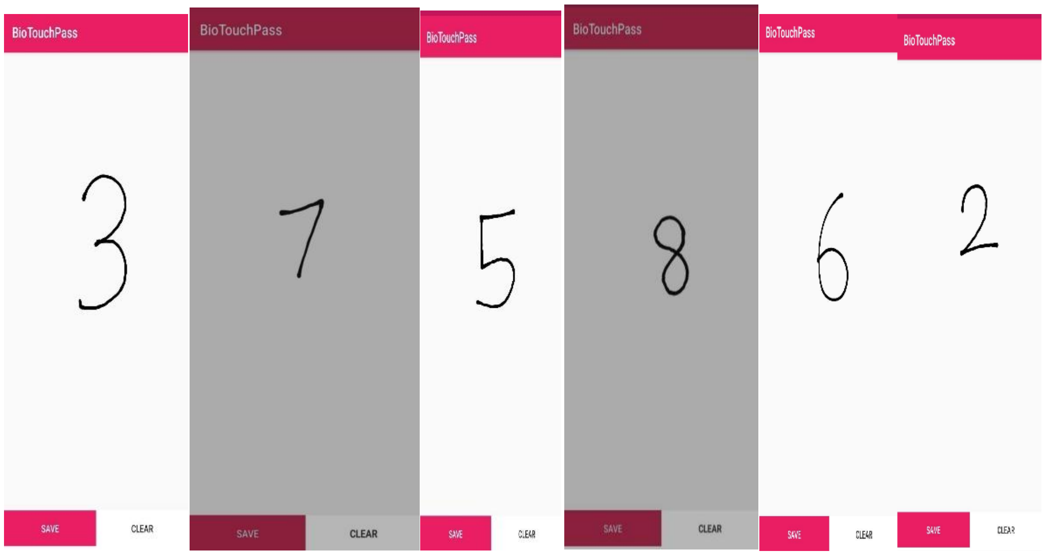
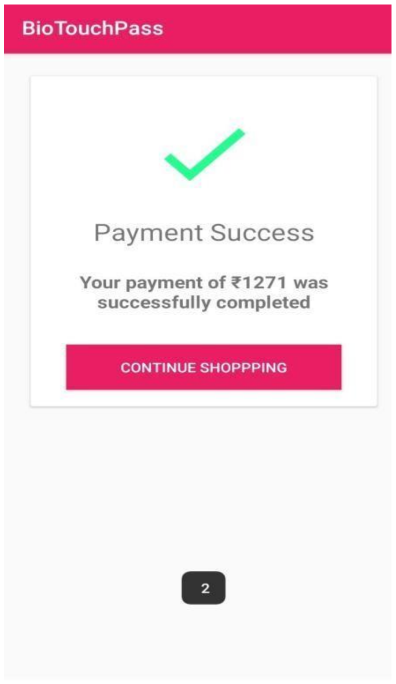


Fig:5.10 Enter Password



Payment Success

Your payment of ₹1271 was successfully completed

Fig:5.11 payment successful

CHAPTER 6

TESTING AND MAINTENANCE

6.1 WHITE BOX TESTING

White-box testing, sometimes called glass-box, is a test case design method that uses the control structure of the procedural design to derive test cases. Using White Box testing methods, we can derive test cases that

- Guarantee that all independent paths within a module have been exercised at least once.
- Exercise all logical decisions on their true and false sides.
- Execute all loops at their boundaries and within their operational bounds.
- Exercise internal data structures to assure their validity.

6.2 BLACK BOX TESTING

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box. You cannot "see" into it. The test provides inputs and responds to outputs without considering how the software works.In this testing by knowing the internal operation of a product, test can be conducted to ensure that "all gears mesh", that is the internal operation performs according to specification and all internal components have been adequately exercised. It fundamentally focuses on the functional requirements of the Software.

6.3 UNIT TESTING

Unit testing is a method by which individual units of source code, sets of one or more computer program modules together with associated control data, usage procedures, and operating procedures are tested to determine if they are fit for use. Intuitively, one can view a unit as the smallest testable part of an application. In procedural programming, a unit could be an entire module,but it is more commonly an individual function or procedure. In object- oriented programming, a unit is often an entire interface, such as a class, but could be an individual method. Unit tests are short code fragments created by programmers or occasionally by white box testers during the development process. Unit testing is software verification and validation method in which the individual units of source code are tested fit for use. A unit is the smallest testable part of an application. In this testing, each classis tested to be working satisfactorily. Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual softwareunits of the application .it is done after the completion of an individual unit before integration.

6.4 FUNCTIONAL TESTING

Functional testing is a quality assurance (QA) process and a type of black box testing that bases its test cases on the specifications of the software component under test. Functions are tested by feeding them input and examining the output, and internal program structure is rarely considered (not like in white-box testing). Functional Testing usually describes what the system does. Functional testing differs from system testing in that functional testing Functional testing typically involves five steps. The identification of functions that the software is expected to perform.

- The creation of input data based on the function's specifications
- The determination of output based on the function's specifications
- The execution of the test case
- The comparison of actual and expected outputs.

6.5 PERFORMANCE TESTING

In general testing performed to determine how a system performs in terms of responsiveness and stability under a particular workload. It can also serve to investigate, measure, validate or verify other quality attributes of the system, such as scalability, reliability and resource usage. Performance testing is a subset of performance engineering, an emerging computer science practice which strives to build performance into the implementation, design and architecture of a system.

6.6 INTEGRATION TESTING

Integration testing is a systematic technique for constructing the program structure while at the same time conducting tests to uncover errors associated with. Individual modules, which are highly prone to interface errors, should not be assumed to work instantly when put together. There may be the chances of data lost across on another's sub functions, when combined may not produce the desired major function; individually acceptable impression may be magnified to unacceptable levels; global data structures can present problems. Integration testing takes as its input modules that have been unit tested, groups them in larger aggregates, appliestests defined in an integration test plan to those aggregates, and delivers asits output the integrated system ready. Simulated usage of shared data areas and inter-process communication is tested and individual subsystems are exercised through their input interface.

6.7 VALIDATION TESTING

Verification and Validation are independent procedures that are used together for checking that a product, service, or system meets requirements and specifications and that it full fills its intended purpose. These are critical components of a quality management system such as ISO 9000. The words "verification" and "validation" are sometimes preceded with "Independent" (or IV&V), indicating that the verification and validation is to be performed by a disinterested third party. It is sometimes said that validation can be expressed by the query "Are you building the right thing?" and verification by "Are you building it right?” In practice, the usage of these terms varies. Sometimes they are even used interchangeably.

6.8 SYSTEM TESTING

System testing of software or hardware is testing conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements. System testing falls within the scope of black box testing, and as such, should require no knowledge of the inner design of the code or logic. As a rule, system testing takes, as its input, all of the "integrated" software components that have passed integration testing and also the software system itself integrated with any applicable hardware system. The purpose of integration testing is to detect any inconsistencies between the software units that are integrated together or between any of the assemblages and the hardware. System testing is performed on the entire system in the context of a Functional Requirement Specification(s) (FRS) and/or a System Requirement Specification (SRS). System testing tests not only the design, but also the behavior and even the believed expectations of the customer. It is also intended to test up to and beyond the bounds defined in the software/hardware requirements specification.

6.9 OUTPUT TESTING

After performing the validation testing, next step is output testing of the proposed system since no system could be useful if it does not produce the required output generated or considered in to two ways. One is on screen and another is printed format. The output comes as the specified requirements by the user. Hence output testing does not result in any correction in the system.

6.10 USER ACCEPTANCE TESTING

User acceptance of a system is the factor for the success of any system. The system under consideration is tested for the user acceptance by constantly keeping in touch with the prospective system users at the time of developing and making changes wherever required.

- Input screen design.
- Output screen design.
- Online message to guide user.
- Format of the ad-hoc reports and other outputs.

Taking various kinds of test data does the above testing. Preparation of test data plays a vital role in the system testing. After preparing the test data the system under study is tested using the test data. While testing the system by using test data errors are again uncovered and correct.

6.11 TEST CASES

Table 6.1: Test Cases

MODULES	**INPUT**	**EXPECTED O/P**	**ACTUAL O/P**	**RESULT**
REGISTER	User name, mobile number, email id and password	Redirect to login screen	Display login screen	Pass
LOGIN	Registered email id and password	Redirect to home page	Display product in the home page	Pass
GENERATE UPI	Draw 4 digit pin	UPI generated successfully	UPI generated successfully	Pass
VIEW PRODUCTS	Select the product to purchase	View the product details	Display the product details	Pass
PURCHASE PRODUCT	Re-enter the generated UPI	Product purchased successfully	Payment successful	Pass

CHAPTER 7

CONCLUSION AND FUTURE ENHANCEMENTS

7.1 CONCLUSION

The achievement of good results with EERs of approximately 4.0% in the proposed approach of bio touchpass is a significant milestone. The fact that this approach outperforms other traditional biometric verification traits such as handwritten signatures or graphical passwords on similar mobile scenarios is a testament to its efficacy. Moreover, specific details for the deployment of the proposed approach on current PIN- and OTP-based authentication systems have been discussed. This is important as it allows for the integration of this newapproach with existing authentication systems, making it more accessible andeasier to implement. Overall, the success of the proposed bio touchpass approach and its potential integration with existing authentication systems can have a significant impact on the security of mobile devices and applications. It provides a new, robust and convenient authentication method that can help prevent unauthorized access and protect sensitive information from being compromised.

7.2 FUTURE ENHANCEMENTS

Future work is to enlarge the e-BioDigit database in Bio touchpass to consider lower and upper case letters and to train more complex deep learning architectures.By doing so, the accuracy of the system can be improved, and it can be more effective in recognizing and classifying different characters. To enlarge the e- BioDigit database, one approach would be to collect more samples of different characters in both lower and upper case letters. This can be done by either manually collecting the samples or by using automated methods such as web
scraping. Once the samples have been collected, they can be labeled and added to

the existing database. Training more complex deep learning architectures can also improve the accuracy of the system. One approach would be to use convolutional neural networks (CNNs), which are known to be effective in image recognition tasks. Another approach would be to use recurrent neural networks (RNNs) to capture temporal dependencies in the data. In addition to enlarging the database and using more complex architectures, other techniques such as data augmentation

and transfer learning can also be used to improve the performance of the system. (Data augmentation can be used to create additional training samples by applying transformations such as rotation, scaling, and cropping to the existing samples. Transfer learning can be used to leverage pre-trained models for similar tasks, which can reduce the amount of training data needed and improve the accuracy of the system.) Overall, there are many ways to improve the e-BioDigit database in Bio touchpass and the deep learning architectures used to recognize and classify characters. By continually refining these approaches, we can create more accurate and reliable systems for character recognition in a variety of applications, including touch-based biometric authentication systems.

REFERENCES

REFERENCES

[1] J. Fierrez, A. Pozo, M. Martinez-Diaz, J. Galbally, and A. Morales, “Benchmarking Touchscreen Biometrics for Mobile Authentication,” IEEE Trans. on Information Forensics and Security, vol. 13, 2018.

[2] J. Galbally, I. Coisel, and I. Sanchez, “A New Multimodal Approach for Password Strength Estimation Part I: Theory and Algorithms,” IEEE Transactions on Information Forensics and Security, vol. 12, pp. 2829– 2844, 2017.

[3] P. Lacharme and C. Rosenberger, “Synchronous One Time Biometrics With Pattern Based Authentication,” in Proc. 11th Int. Conf. on Availability, Reliability and Security, ARES, 2016.

[4] A. Aviv, K. Gibson, E. Mossop, M. Blaze, and J. Smith, “Smudge Attacks on Smartphone Touch Screens,” in Proc. of the 4th USENIX Conference on Offensive Technologies, 2010, pp. 1–7.

[5] M. Liang and X. Hu, “Recurrent Convolutional Neural Network for Object Recognition,” in Proc. of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 3367–3375.

[6] P. Lacharme and C. Rosenberger, “Synchronous One Time Biometrics With Pattern Based Authentication,” in Proc. 11th Int. Conf. on Availability, Reliability and Security, ARES, 2016.

[7] D. Buschek, A. D. Luca, and F. Alt, “There is more to Typing than Speed: Expressive Mobile Touch Keyboards via Dynamic Font Personalisation,” in Proc. of the International Conference on Human- Computer Interaction with Mobile Devices and Services, 2015, pp. 125– 130.

[8] “Improving Accuracy, Applicability and Usability of Keystroke Biometrics on Mobile Touchscreen Devices,” in Proc. of the CHI Conference on Human Factors in Computing Systems, 2015, pp. 1393–

[9] N. Sae-Bae, N. Memon, K. Isbister, and K. Ahmed, “Multitouch Gesture-Based Authentication,” IEEE Transactions on Information Forensics and Security, vol. 9, no. 4, pp. 568–582, 2014.

[10] C. Shen, Y. Zhang, X. Guan, and R. Maxion, “Performance Analysis of Touch-Interaction Behavior for Active Smartphone Authentication,” IEEE Transactions on Information Forensics and Security, vol. 11, no. 3, pp. 498–513, 2016.

[11] R. Tolosana, R. Vera-Rodriguez, J. Fierrez, A. Morales, and J. Ortega- Garcia, “Benchmarking Desktop and Mobile Handwriting across COTS Devices: the e-BioSign Biometric Database,” PLOS ONE, 2017.

[12] M. Martinez-Diaz, J. Fierrez, and J. Galbally, “Graphical Passwordbased User Authentication with Free-Form Doodles,” IEEE Trans. On Human-Machine Systems, vol. 46, no. 4, pp. 607–614, 2016.

[13] T. Kutzner, F. Ye, I. Bonninger, C. Travieso, M. Dutta, and A. Singh, “User Verification Using Safe Handwritten Passwords on Smartphones,” in Proc. 8th International Conference on Contemporary Computing, IC3, 2015.

[14] T. Nguyen, N. Sae-Bae, and N. Memon, “DRAW-A-PIN: Authentication using finger-drawn PIN on touch devices,” Computers and Security, vol. 66, pp. 115–128, 2017.

[15] R. Tolosana, R. Vera-Rodriguez, J. Fierrez, and J. Ortega-Garcia, “Incorporating Touch Biometrics to Mobile One-Time Passwords: Exploration of Digits,” in Proc. IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2018