SECURE VOICE TRANSACTION USING MACHINE LEARNING AND IOT

A PROJECT REPORT

Submitted by

KABILESHWARAN K(811721243022)

MELWIN A.B(811721243027)

MOHAMED FAIZUL S(811721243030)

RAVIDHARSHEN M.K(811721243043)

in partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY

in

ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

K.RAMAKRISHNAN COLLEGE OF TECHNOLOGY

(An Autonomous Institution, affiliated to Anna University Chennai and Approved by AICTE, New Delhi)

SAMAYAPURAM – 621 112

MAY, 2025

K.RAMAKRISHNAN COLLEGE OF TECHNOLOGY

(AUTONOMOUS)

SAMAYAPURAM – 621 112

BONAFIDE CERTIFICATE

Certified that this project report titled " **SECURE VOICE TRANSACTION USING MACHINE LEARNING AND IOT** " is the bonafide work of **KABILESHWARAN K (811721243022), MELWIN A.B (811721243027), MOHAMED FAIZUL S (811721243030), RAVIDHARSHEN M.K (811721243043)** who carried out the project under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form part of any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

SIGNATURE

Dr. T.Avudaiappan M.E., Ph.D.,

HEAD OF THE DEPARTMENT

Department of Artificial Intelligence
K.Ramakrishnan College of Technology
(Autonomous)
Samayapuram – 621 112

SIGNATURE

Mrs. P.Jasmine Jose M.E.,

SUPERVISOR

ASSISTANT PROFESSOR
Department of Artificial Intelligence
K.Ramakrishnan College of Technology
(Autonomous)
Samayapuram – 621 112

Submitted for the viva-voce examination held on ………………

INTERNAL EXAMINER **EXTERNAL EXAMINER**

DECLARATION

We jointly declare that the project report on **“SECURE VOICE TRANSACTION USING MACHINE LEARNING AND IOT”** is the result of original work done by us and best of our knowledge, similar work has not been submitted to **“ANNA UNIVERSITY CHENNAI”** for the requirement of Degree of **BACHELOR OF TECHNOLOGY**. This project report is submitted on the partial fulfilment of the requirement of the award of Degree of **BACHELOR OF TECHNOLOGY**.

Signature

____

KABILESHWARAN K

____

MELWIN A.B

____

MOHAMED FAIZUL S

____

RAVIDHARSHEN M.K

Place: Samayapuram

Date:

ACKNOWLEDGEMENT

It is with great pride that we express our gratitude and in-debt to our institution "**K.Ramakrishnan College of Technology (Autonomous)**", for providing us with the opportunity to do this project.

We are glad to credit honourable chairman **Dr. K.RAMAKRISHNAN, B.E.,** for having provided for the facilities during the course of our study in college.

We would like to express our sincere thanks to our beloved Executive Director **Dr. S. KUPPUSAMY, MBA, Ph.D.,** for forwarding to our project and offering adequate duration in completing our project.

We would like to thank **Dr. N. VASUDEVAN, M.E., Ph.D.,** Principal, who gave opportunity to frame the project the full satisfaction.

We whole heartily thank to **Dr. T.AVUDAIAPPAN**, **M.E., Ph.D.,** Head of the department, **ARTIFICIAL INTELLIGENCE** for providing his encourage pursuing this project.

We express our deep and sincere gratitude to our project guide **Mrs. P. JAMINE JOSE, M.E.,** Department of **ARTIFICIAL**

INTELLIGENCE, for her incalculable suggestions, creativity, assistance and patience which motivated us to carry out this project.

We render our sincere thanks to Course Coordinator and other staff members for providing valuable information during the course.

We wish to express our special thanks to the officials and Lab Technicians of our departments who rendered their help during the period of the work progress.

ABSTRACT

In the rapidly evolving domain of financial technology, secure voice transactions are emerging as a pivotal innovation to enhance user convenience while maintaining stringent security standards. This project presents the development of a secure, voice-guided Android application that integrates a multi-layered authentication framework and voice-based payment processing. The system leverages advanced machine learning techniques, including Convolutional Neural Networks (CNNs) for fingerprint recognition and Recurrent Neural Networks (RNNs) for voice authentication, ensuring robust user verification. Additionally, Dynamic Time Warping (DTW) and Hidden Markov Models (HMM) are employed for accurate voice PIN recognition. Developed using Java and XML in Android Studio, the application incorporates a sequential authentication process voice PIN verification, RFID-based authentication, and fingerprint verification to establish high reliability and user-specific access. After authentication, the application enables users to select shops and enter payment amounts through voice input, providing a seamless, secure transaction experience. The system is trained on a diverse dataset to accommodate different accents and voice variations and further enhanced with supervised and reinforcement learning techniques for fraud detection and continuous improvement. Experimental evaluations demonstrate high accuracy in both user authentication and transaction processing, establishing the solution as a scalable, user-friendly platform for voice-activated payments across mobile and retail environments.

TABLE OF CONTENTS

CHAPTER **TITLE** **PAGE NO.**

LIST OF FIGURES

LIST OF ABBREVIATIONS

API	**-**	Application Programming Interface
ASR	**-**	Automatic Speech Recognition
DTW	**-**	Dynamic Time Warping
HMM	**-**	Hidden Markov Model
IOT	**-**	Internet of Things
SMS	**-**	Short Message Service
SVM	**-**	Support Vector Machine
TTS	**-**	Text-To-Speech
UI	**-**	User Interface

CHAPTER 1
INTRODUCTION

In today’s rapidly evolving digital landscape, ensuring secure, seamless, and accessible transactions has become a critical priority. Traditional authentication methods often involve multiple manual steps, increasing the risk of errors and unauthorized access. To address these challenges, the "Secure Voice Transaction System Using Machine Learning" project introduces a voice-guided, multi-factor authentication Android application designed to offer an intuitive and robust user experience.

Leveraging recent advancements in machine learning (ML), this system integrates voice recognition, fingerprint verification, and RFID authentication to establish a multi-layered security framework. Developed using Java and XML in Android Studio, the application ensures that only authorized users can complete transactions through a step-by-step process: voice PIN verification, RFID scan, and fingerprint authentication. Once verified, the user is prompted to speak the shop name and payment amount, followed by a voice-based confirmation to complete the transaction.

The system incorporates cutting-edge ML algorithms and deep learning models such as convolutional neural networks (CNNs), recurrent neural networks (RNNs), dynamic time warping (DTW), and hidden Markov models (HMMs) to enhance voice recognition and fraud detection. By merging biometric and voice technologies, the project aims to create a secure, user-friendly payment solution suitable for retail environments, smart kiosks, and self-service stations. Its hands-free interface also promotes accessibility and inclusivity, especially for differently-abled individuals.

1.1 BACKGROUND

The need for secure and seamless payment systems has grown with the rise of digital and voice-based transactions. Traditional methods like passwords and PINs pose security risks and limit user convenience. In contrast, biometric technologies—such as voice recognition, fingerprint verification, and RFID—combined with machine learning models like CNNs offer a more reliable and accessible alternative.

This project addresses these needs by developing an Android application with multi-factor authentication. Using Java and XML, the app integrates voice PIN input, RFID scanning, and fingerprint recognition to provide a secure, touch-free payment experience suitable for smart retail and inclusive environments.

1.2 PROBLEM STATEMENT

In the current digital and financial landscape, ensuring secure, fast, and user-friendly transactions remains a major challenge. Traditional authentication methods such as passwords, PINs, and touch-based systems are prone to security breaches, user errors, and accessibility issues. As voice-activated and mobile payments grow, existing systems often lack the accuracy, security, and multi-modal integration needed to prevent fraud and enhance usability. There is a pressing need for a seamless, multi-factor authentication system that combines voice recognition, fingerprint verification, RFID, and voice PIN in a cohesive and hands-free solution.

1.3 AIM AND OBJECTIVE

1.3.1 Aim

- Develop a voice-guided Android application that enables secure, multi-layered user authentication for payment processing.
- Integrate biometric techniques such as voice recognition, fingerprint verification, and RFID scanning to enhance transaction security.
- Implement the system using Java and XML in Android Studio to ensure cross-platform compatibility and responsiveness.
- Leverage advanced machine learning and deep learning models, including NLP techniques, to improve the accuracy of voice and fingerprint recognition.
- Ensure the application delivers a seamless and accessible user experience by minimizing manual input and promoting hands-free interaction.
- Design the system to be intuitive, scalable, and suitable for real-time environments like retail stores, smart kiosks, and self-service stations.

- Develop a secure, voice-guided Android application using Java and XML, integrating multi-layered authentication techniques including voice recognition, RFID scanning, and fingerprint verification.
- Enhance the security and user experience of voice-based transactions by combining advanced biometric authentication methods, ensuring high accuracy and resistance to fraud.
- Create a seamless, hands-free payment system by enabling voice input for shop name, payment amount, and providing real-time voice feedback for transaction confirmation.
- Ensure the accessibility and usability of the system, particularly for differently-abled users and in fast-paced environments, by leveraging voice-based interfaces and biometric security.

1.3.2 Objective

- Analyze and evaluate existing biometric and machine learning techniques for voice recognition, fingerprint verification, and voice PIN security in payment systems.
- Develop a secure, voice-guided Android application using Java and XML that supports multi-layered user authentication.
- Build a machine learning-based voice authentication model utilizing deep learning techniques such as CNNs and RNNs for accurate user verification.
- Integrate voice-based PIN verification, RFID scanning, and fingerprint authentication into a single, cohesive system.
- Enhance model robustness against spoofing attacks and fraud using optimization techniques for both voice and fingerprint recognition.
- Design and implement a user-friendly interface that supports hands-free transactions through voice input of shop name and payment amount.
- Provide real-time voice feedback for transaction confirmation to improve user confidence and interaction
- Incorporate advanced NLP models such as BERT or Transformers to strengthen the system’s voice understanding and adaptability.

- Test the system across different user scenarios, environments, and voices to ensure accuracy, accessibility, and reliability.
- Evaluate system performance through user feedback, stress testing, and comparison with traditional authentication methods.
- Ensure the system is scalable, ethically developed, and capable of handling diverse users and transaction scenarios.
- Promote accessibility by designing the system to be usable by differently-abled individuals and in fast-paced or hands-busy environments.

CHAPTER 2

LITERATURE SURVEY

2.1 SECURE VOICE TRANSACTION SYSTEM USING MULTI- FACTOR AUTHENTICATION

P. Babakhani, D. Sacker, F. Sivrikaya, and S. Albayrak

The Secure Voice Transaction System enables secure voice-based financial transactions using advanced NLP and multi-factor authentication. It utilizes transformer models like Whisper and BERT for speech-to-text conversion and natural language understanding. User identity is verified through voice biometrics and deep-learning-based speaker verification. Once authenticated, transactions are processed securely via API communication with banking systems, ensuring both safety and efficiency.

Merits

- The integration of multi-factor authentication (MFA) significantly enhances security for voice-based transactions, reducing fraud risk.
- The use of transformer models like Whisper and BERT ensures high accuracy in speech recognition and understanding, even in complex or noisy environments.
- Voice biometrics provide an additional layer of identity verification, enhancing trust in the transaction process.

Demerits

- Although the model is powerful for news articles, its generalizability to other domains (e.g., social media, academic texts) may be limited unless further fine-tuned.
- Background noise or speech variations could still impact accuracy, leading to errors in both speech recognition and authentication.

2.2 DEFENSE AGAINST AI-SYNTHESIZED VOICE ATTACKS

S.Rathi

This paper presents an adversarial training approach to detect AI-generated voices using spectrogram analysis to identify synthetic artifacts. It employs Generative Adversarial Networks (GANs) for accurate classification of voice signals. Convolutional Neural Networks (CNNs) are integrated within the GAN framework to enhance detection accuracy. Feature extraction is performed through detailed spectrogram analysis of voice patterns. This method significantly strengthens the security of voice authentication systems against AI-synthesized voice attacks.

Merits

- Effectively counters state-of-the-art voice synthesis tools through adversarial training, providing robust protection for voice authentication systems.
- The combination of spectrogram analysis and GANs enhances the ability to detect subtle synthetic artifacts that are difficult to spot with traditional methods.
- The system improves the overall security and reliability of voice-based authentication, making it more resistant to deepfake attacks.

Demerits

- The model may have a high false-positive rate, especially with low-quality recordings, potentially causing legitimate users to be falsely flagged.
- Adversarial training may require significant computational resources for training and real-time deployment, making it less efficient for systems with limited hardware.

2.3 NOTEPAL-A CURRENCY DETECTOR APPLICATION FOR VISUALLY IMPAIRED

G. Kranthi Kumar, V. L. Durga Keerthi Pampani

Notepal is a mobile app designed to help visually impaired individuals identify currency denominations. It uses a Random Forest Classifier for accurate note classification and OpenCV for efficient image processing. Feature extraction is performed to enhance detection accuracy. Text-to-Speech (TTS) technology provides clear audio feedback to users. This system promotes financial independence through accessible and user-friendly interaction.

Merits

- The system is specifically tailored for visually impaired users, offering a practical and easy-to-use solution for currency recognition, improving financial independence.
- The combination of machine learning (Random Forest Classifier) and image processing (OpenCV) ensures accurate and efficient currency detection.
- TTS technology ensures that users receive instant, accessible feedback, making the system interactive and helpful in real-world scenarios.

Demerits

- The system's accuracy can be influenced by external factors such as poor lighting and subpar camera quality, affecting the overall reliability of currency detection.
- The application may struggle with distinguishing between visually similar denominations or worn-out currency notes, leading to potential errors.
- Dependence on mobile devices with adequate processing power and camera quality may limit accessibility for users with older or lower-end smartphones.

2.4 INNOVATIVE CURRENCY IDENTIFIER FOR THE BLIND THROUGH AUDIO OUTPUT USING DEEP LEARNING

R. Maganti, J. Gutla, Y. Mallimpalli, S. Yellisetti

This paper highlights the limitations of existing currency identification systems for the visually impaired. To address these challenges, a new Currency Identification System (CIS) is proposed using CNN and ResNet architectures. ResNet was chosen for deployment due to its superior accuracy and ability to overcome vanishing gradient issues. The system achieved 90% classification accuracy in identifying currency denominations. This approach offers a reliable and efficient solution for blind users.

Merits

- The system demonstrates high effectiveness with 90% classification accuracy, leveraging deep learning models to provide robust currency recognition for blind individuals.
- The use of Residual Networks (ResNet) ensures improved model performance, overcoming challenges like vanishing gradients that may affect traditional CNN models.
- The audio output feature ensures seamless and accessible interaction for blind users, providing real-time feedback for currency identification.

Demerits

- The system requires significant computational resources for model training, which may not be feasible for all developers, especially those with limited hardware.
- The model's performance could be hindered by environmental factors, such as low-quality images or distortions in currency notes, leading to potential misidentification.
- The need for a well-trained model and substantial data sets may limit the speed of deployment and scalability in real-world applications.

2.5 IMPLEMENTATION OF THE VOICE-BASED CONTROL AND DETECTION OF THE CURRENCY IN ATM TRANSACTION

N. Tejashwini

This paper proposes a voice-enabled ATM system to enhance user experience and security during transactions. It uses speech recognition models like Google Speech-to-Text or DeepSpeech to interpret spoken commands. For currency denomination detection, machine learning techniques such as CNN or OCR are employed. These technologies enable accurate classification of currency notes. The system ensures a more accessible and secure ATM interaction.

Merits

- The system significantly enhances accessibility for visually impaired users, allowing them to perform ATM transactions hands-free using voice commands, promoting independence.
- Voice-based control offers a convenient and user-friendly method of interacting with ATMs, improving the overall user experience.
- The use of machine learning algorithms (CNN, OCR) ensures accurate detection and classification of currency denominations, contributing to the system's reliability.

Demerits

- One major drawback is that background noise can interfere with the accuracy of voice recognition, potentially leading to errors in transaction processing.
- Variations in speech (e.g., accents or speech impediments) could lead to incorrect command recognition, affecting the reliability of the system.
- The system may require high computational resources for real-time speech processing and image recognition, which could limit its feasibility for certain ATM hardware.

CHAPTER 3

SYSTEM ANALYSIS

3.1 EXISTING SYSTEM

Current voice-based transaction and ATM systems incorporate a range of technologies to facilitate secure financial operations, yet each has distinct limitations. Modern voice transaction systems typically leverage machine learning and biometric authentication, utilizing deep learning models such as Convolutional Neural Networks (CNNs) for fingerprint recognition and Recurrent Neural Networks (RNNs) for voice recognition. These systems often employ techniques like Dynamic Time Warping (DTW) and Hidden Markov Models (HMM) for voice PIN analysis, enhancing identity verification accuracy. Despite these advancements, such systems still face challenges in robustness against voice spoofing and maintaining accuracy in noisy environments. On the other hand, traditional ATM systems heavily rely on manual interactions via keypads and screens, which poses usability issues for individuals with visual or physical impairments. These systems are prone to security threats like shoulder surfing and PIN theft and lack automation in detecting currency denominations. The limitations of both current approaches emphasize the need for a unified, intelligent system that integrates multi-modal biometrics and voice-driven interfaces to offer a more secure, inclusive, and efficient transaction experience.

3.1.1 Algorithm Used

Convolutional Neural Network

CNN are widely used for fingerprint recognition in biometric authentication systems. In the context of the secure voice transaction system, CNN are employed to analyze and verify fingerprint data with high accuracy.

The model is designed to learn spatial hierarchies of features from fingerprint images, making it effective for distinguishing between genuine and spoofed biometric data.

Recurrent Neural Network

RNN particularly Long Short-Term Memory (LSTM) networks, are utilized for voice recognition in the secure voice transaction system. RNN are ideal for processing sequential data, like voice signals, and they can capture temporal dependencies, allowing them to accurately recognize speech patterns and user voice features.

Hidden Markov Model

HMM are powerful techniques used for securing voice PINs within the secure voice transaction system. DTW is employed to measure similarity between voice inputs, even when there are slight variations in speed or pronunciation. HMM on the other hand, are used to model sequences of speech patterns over time, making them particularly effective for identifying and validating voice PINs.

3.1.2 Disadvantage of Existing System

- Lacks accessibility for visually impaired or physically challenged individuals due to reliance on visual and manual input.
- Traditional PIN-based authentication is vulnerable to security threats like card skimming, shoulder surfing, and unauthorized access.
- Complex and non-intuitive user interface can be difficult for elderly users or those unfamiliar with technology.
- Absence of voice assistance makes navigation and transaction processing challenging for many users.
- Manual verification of currency denominations increases the risk of human error and disputes over dispensed cash.
- No feedback mechanism is available to confirm successful or failed transactions in real time.
- ATM systems lack biometric authentication (e.g., fingerprint or voice recognition), reducing overall security.
- Cannot support hands-free operation, limiting usability in fast-paced or differently-abled user environments.
- Maintenance and updating of physical interfaces can be time-consuming and costly.

- Limited personalization or adaptability based on user behavior or needs.

3.2 PROPOSED SYSTEM

The proposed system is a comprehensive, multi-modal voice transaction platform designed to enhance security, accessibility, and user experience in digital financial operations. It integrates advanced biometric authentication methods including voice recognition, fingerprint verification, and RFID-based identification. Voice PIN recognition is employed using sophisticated models like Dynamic Time Warping (DTW) and Hidden Markov Models (HMMs) to secure the authentication process against spoofing and voice variation attacks. Convolutional Neural Networks (CNNs) are used for accurate fingerprint authentication, while Recurrent Neural Networks (RNNs) handle the voice recognition tasks effectively.

The system is developed as a hands-free Android application using Java and XML in Android Studio, incorporating voice guidance throughout the transaction flow. After multi-factor authentication, the user can speak the merchant's name and the transaction amount, and the system processes the payment seamlessly. A voice-based confirmation is provided at the end to inform the user about the transaction status.

This design ensures secure, fast, and intuitive transactions, especially benefiting visually impaired and physically challenged users. The integration of speech, biometrics, and contactless technologies enables the system to operate effectively in smart retail environments, public kiosks, and self-service scenarios, promoting both inclusivity and enhanced digital security.

3.2.1 Techniques Used

Convolutional Neural Network

CNN are widely used for fingerprint recognition in biometric authentication systems. In SecureVoice, CNN analyze fingerprint data to verify user identities with high accuracy. These models are excellent at recognizing spatial hierarchies within images, enabling secure, real-time fingerprint authentication. CNN are crucial for distinguishing between genuine users and fraudulent attempts in voice-based transactions, ensuring robust security for the system

Recurrent Neural Network

RNN particularly Long Short-Term Memory (LSTM) networks, are used for voice recognition in the SecureVoice system. RNN excel in processing sequential data, like speech signals, capturing temporal dependencies between voice features. These networks are effective in analyzing user speech patterns and verifying vocal identity, which is essential for authenticating users during voicebased transactions, especially in noisy environments or with various accents.

Dynamic Time Warping

DTW is a technique used to measure similarity between sequences, such as voice recordings, even when there are slight variations in pronunciation or speaking speed. SecureVoice utilizes DTW to accurately match voice PIN inputs with stored voice models, ensuring that users can securely input PINs by voice, while resisting variations in their speech patterns.

Hidden Markov Model

HMM are employed in SecureVoice for modeling speech patterns over time. These models are particularly useful for recognizing sequential data like speech and can capture the transitions between different spoken elements (such as words or phonemes). By using HMMs the system ensures accurate recognition and validation of voice PINs, even in the presence of environmental noise or slight changes in the user's speech

Voice Biometric

Template-based question generation involves creating question structures or templates that are filled with relevant content based on the input text. By defining question types such as "What is...?" or "Explain how...", this technique can produce accurate questions, although it may lack the creativity of more advanced models.

Voice Spoofing Detection

Voice spoofing detection is critical in ensuring that fraudulent actors cannot impersonate legitimate users. The SecureVoice system utilizes machine learning models to detect voice manipulation or synthetic voices, ensuring that the authentication process remains secure against spoofing attempts. This is achieved by analyzing inconsistencies and irregularities in the voice signal, ensuring robust fraud detection.

Multi-Modal Authentication

Secure Voice incorporates a multi-modal approach that combines fingerprint authentication with voice recognition to provide a higher level of security. The system utilizes deep learning models to evaluate both voice and biometric data, ensuring that the user is authenticated using multiple verification methods.

3.2.2 Advantages of Proposed System

- Enhances transaction security by integrating multi-factor biometric authentication, including voice recognition, fingerprint verification, and RFID-based identity checks.
- Utilizes advanced machine learning models such as CNNs for fingerprint analysis and RNNs for accurate voice recognition, ensuring high reliability.
- Employs Dynamic Time Warping (DTW) and Hidden Markov Models (HMMs) to secure voice PIN entry, effectively preventing spoofing and unauthorized access.
- Hands-free operation through voice commands supports seamless transaction flow, especially in situations where physical interaction is difficult or limited.
- Provides voice-guided feedback throughout the transaction process, improving usability and user confidence, particularly for visually impaired or elderly users.
- Adapts to various languages and accents, making it accessible to a diverse and global user base.
- Reduces fraud risk using voice biometrics and spoof detection mechanisms, ensuring the authenticity of the user.
- Real-time processing enables faster transactions with minimal delays, improving the overall efficiency of digital payments.

- Compatible with Android devices and existing hardware such as fingerprint sensors and RFID readers, facilitating cost-effective deployment.
- Ideal for smart retail, self-service kiosks, and inclusive banking services, promoting accessibility and financial autonomy.
- Scalable and adaptable for large-scale applications, capable of handling high transaction volumes securely.
- Continuously trainable and upgradable to counter emerging threats and adapt to evolving user behaviors, ensuring long-term reliability.

CHAPTER 4

SYSTEM SPECIFICATION

4.1 HARDWARE SPECIFICATION

- ESP32 Microcontroller: Handles fingerprint, RFID, Bluetooth, and Wi-Fi communication efficiently in the IoT module
- High-Performance CPU (AMD Ryzen 9 or Intel Core i9): Powers ML model training and inference tasks
- NVIDIA RTX 3090 GPU: Accelerates deep learning operations, especially voice recognition and feature extraction
- Fingerprint Module (R305 or GT-521F52): Provides secure biometric authentication
- RFID Reader (RC522): Enables contactless card-based user identification
- High-Quality Microphone (USB or XLR): Captures clear voice input for secure transaction validation
- LCD Display (16x2 or 20x4 with I2C): Displays system messages, authentication status, and prompts
- 1 TB NVMe SSD: Offers fast read/write speeds for storing models, voice data, and system logs
- 5V 2A Power Supply: Provides stable power to the embedded IoT module
- 1 Gbps Internet Router with Wi-Fi: Ensures reliable and fast communication between the IoT system and the central server

4.2 SOFTWARE SPECIFICATION

- Operating System: Windows 10 or Windows 11 – for development, deployment, and compatibility with Java tools
- Programming Language: Java – used for both backend logic and application development
- Frontend Framework: XML – for designing UI layouts in Java-based environments (e.g., Android, JavaFX)

- Java Development Kit (JDK): JDK 11 or higher – to compile and run Java code efficiently
- Integrated Development Environment (IDE): IntelliJ IDEA or Eclipse – for writing, testing, and debugging Java/XML code
- Voice Processing Library: CMU Sphinx (Sphinx4) or Google Speech API – for voice recognition and authentication
- Machine Learning Library: Weka or Deeplearning4j – for implementing and training ML models in Java
- Database: MySQL or SQLite – for storing user credentials, voiceprints, RFID tags, and transaction logs
- Build Tool: Apache Maven or Gradle – for managing project dependencies and build automation
- Security Framework: Java Cryptography Extension (JCE) – for encrypting voice data, fingerprints, and secure transactions

4.3 SOFTWARE DESCRIPTION

The Secure Voice Transaction Using Machine Learning software system integrates Java-based development with advanced machine learning and biometric technologies to deliver a secure, intelligent transaction platform. Built primarily using Java for both frontend and backend with XML for UI design, the system ensures platform independence, robustness, and strong object-oriented capabilities. Voice recognition is handled using deep learning models (RNN/LSTM) trained with TensorFlow and PyTorch, leveraging librosa and pyAudio for audio feature extraction, while fingerprint authentication uses CNN models with OpenCV for high-accuracy image analysis. The ML models are deployed via Flask or FastAPI and accessed by the Java application through secure REST APIs. Biometric data and voiceprints are encrypted and stored in a secure database (MySQL or SQLite), with authentication further strengthened using Java Cryptography Extension (JCE). Hardware modules like RFID readers, fingerprint sensors, LCDs, and ESP8266 are integrated via Bluetooth and serial communication, enabling seamless multi-modal authentication. This fusion of biometric security and Java technology ensures real-time, secure voice-driven transactions across platforms.

4.3.1 Library

TensorFlow

TensorFlow is a popular open-source machine learning framework developed by Google. In this project, it is used to build deep learning models such as Recurrent Neural Networks (RNN) and Long Short-Term Memory (LSTM) networks for voice recognition. It is also used for training Convolutional Neural Networks (CNN) for fingerprint verification. TensorFlow's flexibility and performance allow easy deployment of models to edge devices if needed.

PyTorch

PyTorch is another deep learning framework developed by Facebook. Like TensorFlow, it is used for building and training neural networks. PyTorch is often preferred for research and rapid prototyping because of its dynamic computation graph and ease of debugging. In this system, PyTorch is particularly useful when experimenting with custom architectures for voice and biometric models.

Librosa

Librosa is a Python library specialized for music and audio analysis. It is used here to extract audio features such as MFCC (Mel Frequency Cepstral Coefficients), chroma, and spectral contrast from the voice recordings. These features are essential for training voice recognition models.

PyAudio

PyAudio is used for capturing audio input in real-time through microphones. It provides a simple Python interface to PortAudio, allowing real-time streaming and recording, which is crucial for live voice-based authentication.

OpenCV

OpenCV (Open Source Computer Vision Library) is used for image processing tasks related to fingerprint recognition. It helps in operations such as image enhancement, edge detection, and feature extraction from fingerprint images before feeding them into CNN models for classification.

scikit-learn

Scikit-learn is a comprehensive machine learning library in Python. It is used for various tasks such as data preprocessing (e.g., normalization, splitting datasets), model evaluation (e.g., accuracy, confusion matrix), and sometimes for implementing classical machine learning algorithms for comparison or baseline models.

Flask / FastAPI

Flask and FastAPI are lightweight web frameworks for Python. They are used to develop the backend of the system, which exposes APIs for voice, fingerprint, and RFID authentication. FastAPI, in particular, is known for its speed and support for asynchronous operations, making it suitable for real-time secure transactions.

Pandas

Pandas is a powerful data manipulation and analysis library. It is used to manage and process structured data such as user profiles, authentication logs, and transaction records stored in tabular format.

NumPy

NumPy provides support for large, multi-dimensional arrays and matrices, along with a collection of mathematical functions. It works in conjunction with other libraries like TensorFlow and PyTorch to perform efficient numerical computations on biometric data.

Hugging Face Transformers

This library provides pre-trained models for natural language processing (NLP) and speech processing. In this project, it can be used to integrate pre-trained speech-to-text or speaker identification models that help in validating the speaker's identity as part of voice authentication.

4.3.2 Developing environment

Operating System

The development environment for the Secure Voice Transaction Using Machine Learning project is built on Ubuntu 20.04 LTS, a stable and versatile platform known for its compatibility with various machine learning and biometric tools. It is ideal for running machine learning frameworks such as TensorFlow and PyTorch. Additionally, Windows 10/11 is used for tasks involving Java programming, particularly when dealing with the front-end and hardware integration like RFID and fingerprint modules.

Programming Environment

The project employs a combination of Python, Java, and C/C++ for different parts of the system. Python is used for machine learning tasks such as voice recognition and biometric authentication, leveraging powerful libraries like TensorFlow and PyTorch. Java handles hardware integration, specifically for controlling RFID and Bluetooth modules, and also supports the development of Android applications. C/C++ are used when dealing with embedded systems and programming microcontrollers.

IDE and Development Tools

The development is carried out using specialized Integrated Development Environments (IDEs). PyCharm Professional is used for Python-based machine learning development, while Eclipse IDE is employed for Java-based development, especially for Android applications and hardware integration. For embedded systems, the Arduino IDE or PlatformIO is used to program microcontroller boards such as the ESP8266 or ESP32.

Machine Learning Libraries

Machine learning is a core component of the project. TensorFlow and PyTorch are the primary libraries used for developing deep learning models, particularly for voice recognition and fingerprint verification. Librosa is used to extract audio features such as Mel-frequency cepstral coefficients (MFCCs) from voice data. PyAudio and SpeechRecognition libraries enable real-time voice input processing, while OpenCV handles image processing and fingerprint matching.

Hardware Interface

For hardware integration, ESP8266 or ESP32 microcontroller boards are utilized to interface with RFID and fingerprint sensors. These microcontrollers communicate with the system using Wi-Fi or Bluetooth, ensuring secure multi-factor authentication. The Arduino IDE is used to program the microcontroller boards, while USB-to-serial drivers facilitate communication between hardware components and the main system.

Database and Data Management

The project employs SQLite for local storage of authentication logs and small datasets on mobile devices. For more extensive data management, MySQL or PostgreSQL is used to store transaction history, biometric data, and user profiles securely on cloud servers. Data encryption ensures the protection of sensitive information throughout the transaction process.

Backend and API Integration

The backend is built using web frameworks like Flask or FastAPI, which allow the deployment of machine learning models as web APIs. These APIs enable voice and fingerprint authentication to be processed in real time. For larger, enterprise-level applications, Spring Boot can also be utilized to manage complex system integrations. Nginx is used as a reverse proxy to handle API traffic, ensuring seamless interaction between the front end and backend systems.

Testing and Monitoring Tools

To ensure the system functions as intended, various testing and monitoring tools are employed. Postman is used for testing API endpoints, ensuring proper communication between different parts of the system. WireShark is used to monitor communication between hardware components, checking data transfer and ensuring proper connectivity.

Version Control and Collaboration

Version control is handled using Git and GitHub, ensuring collaboration among multiple developers and maintaining code integrity. These tools allow developers to track changes, manage codebase versions, and streamline collaboration within the project team.

Security Features and Authentication

Before deployment, it is essential to integrate robust security features. Secure communication protocols like TLS or SSL are used to protect data in transit between devices, APIs, and servers. Additionally, biometric data is stored in an encrypted format, ensuring it remains secure even if unauthorized access is attempted. Multi-factor authentication (MFA) using voice, fingerprint, and RFID adds an additional layer of security. Access control mechanisms, such as role-based authentication (RBAC), help to ensure that only authorized users can perform sensitive operations within the system.

Deployment and Cloud Integration

The deployment of the system is optimized using Docker to containerize machine learning models and their dependencies, ensuring consistency across different environments. Kubernetes is used to scale the application, especially when handling a high volume of voice transaction requests. For large-scale deployments, the system can be integrated with cloud platforms like Google Cloud or AWS, offering scalable compute resources for model training and inference, and secure storage for biometric data.

CHAPTER 5

SYSTEM DESIGN

5.1 SYSTEM ARCHITECTURE

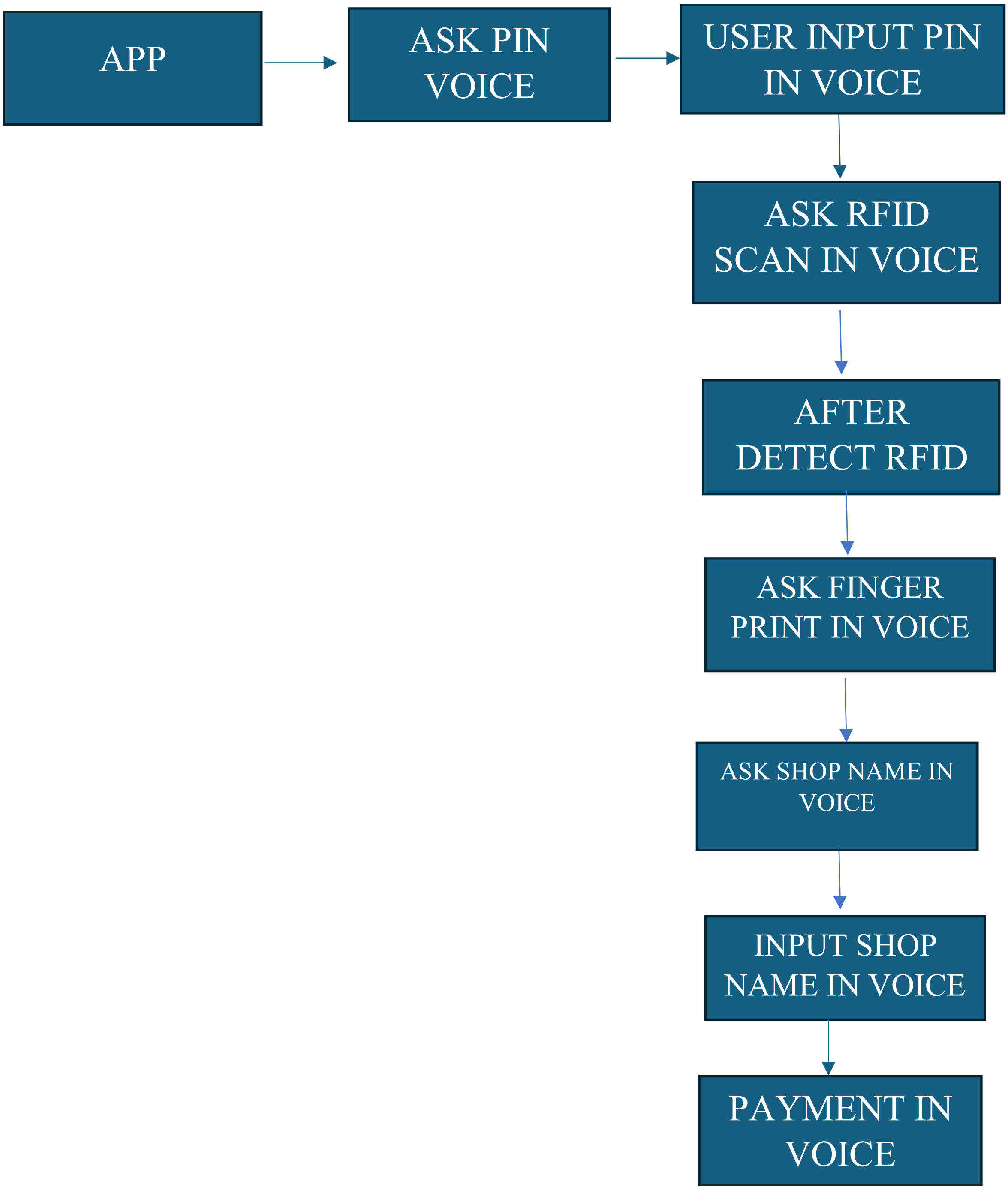


Fig. 5.1 System Architecture

Fig.5.1 represents the architecture of the Secure Voice Transaction System, detailing the step-by-step flow of the authentication and transaction process. The process begins when the user opens the application (APP) on their device. Once the app is launched, it initiates a security protocol by prompting the user to enter their Personal Identification Number (PIN). This PIN serves as the first layer of authentication to verify the user's identity.

After the user inputs the PIN, the system proceeds to the second authentication step, which involves requesting the user's RFID (Radio Frequency Identification). This could be in the form of an RFID-enabled card or tag linked to the user's account. Upon successful RFID scanning, the system transitions to the next stage, labeled "After", signifying that prior validations were successful and the user can proceed further.

The third layer of authentication involves biometric verification through fingerprint scanning. This adds an additional layer of security by ensuring that the user is physically present and matches the registered fingerprint data.

Following the authentication steps, the system shifts its focus to transaction-specific details. It prompts the user to provide the name of the shop where the transaction is to take place. After the user inputs the shop name, the system confirms the transaction parameters and prepares to execute the transaction.

Finally, the process culminates in the "Payment In" stage, where the transaction amount is processed, and the payment is made to the designated shop. This structured and layered approach ensures that the transaction is secure, reliable, and authenticated at multiple levels before payment is authorized.

5.2 DATA FLOW DIAGRAM

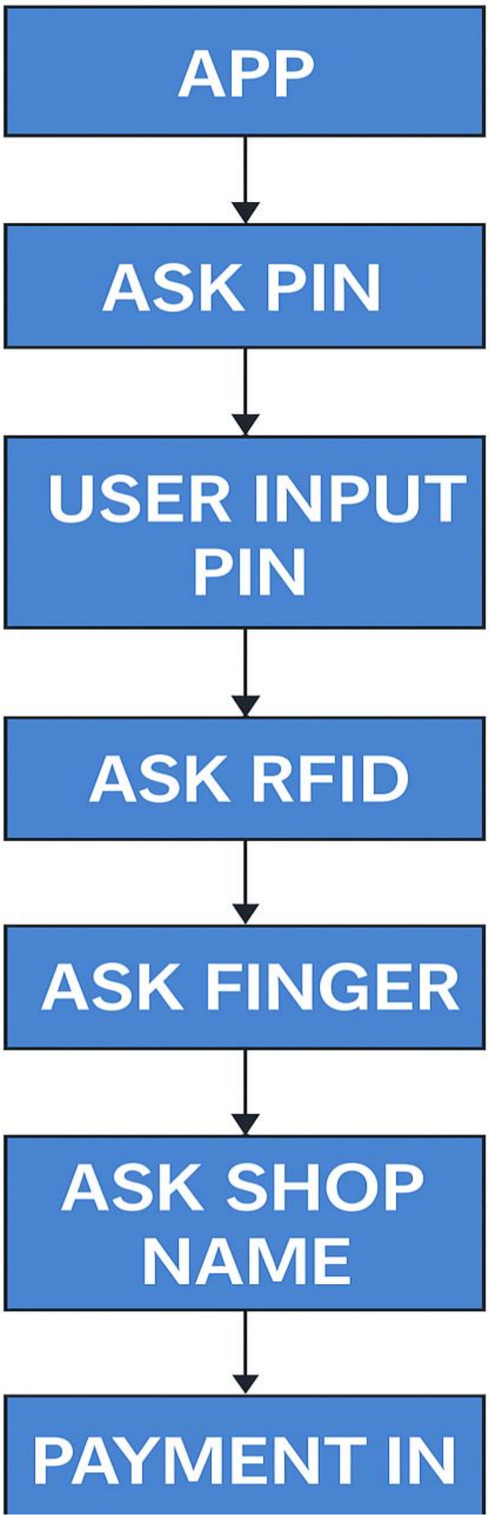


Fig. 5.2 Data Flow Diagram

Fig.5.2 presents the architectural flow of the Secure Voice Transaction System, depicting a structured, multi-layered process for initiating and completing secure transactions. The process starts with the user opening the application (APP), which serves as the gateway to the transaction system. Upon launching the app, the user is prompted to enter a PIN, which acts as the first level of authentication to confirm the identity of the user. Once the PIN is entered, the system proceeds to request RFID verification, where the user must scan an authorized RFID card or device linked to their account.

Following successful RFID validation, the flow transitions to the next phase, indicated as "AFTER", suggesting the continuation of the process upon successful preliminary checks. The user is then asked to provide a fingerprint scan, adding a strong biometric layer to the security framework. This fingerprint check ensures that only the authorized individual can proceed with the transaction.

Once the fingerprint is verified, the system prompts the user to input the shop name, allowing the transaction to be directed to the correct recipient. The user then enters the shop details in the next step. Finally, upon successful entry of all required data and authentication validations, the system proceeds to the "PAYMENT IN" stage, where the transaction is securely processed and completed. This architectural flow ensures that the system enforces strong, layered security while maintaining a user-friendly process for secure voice transactions.

5.3 USE CASE DIAGRAM

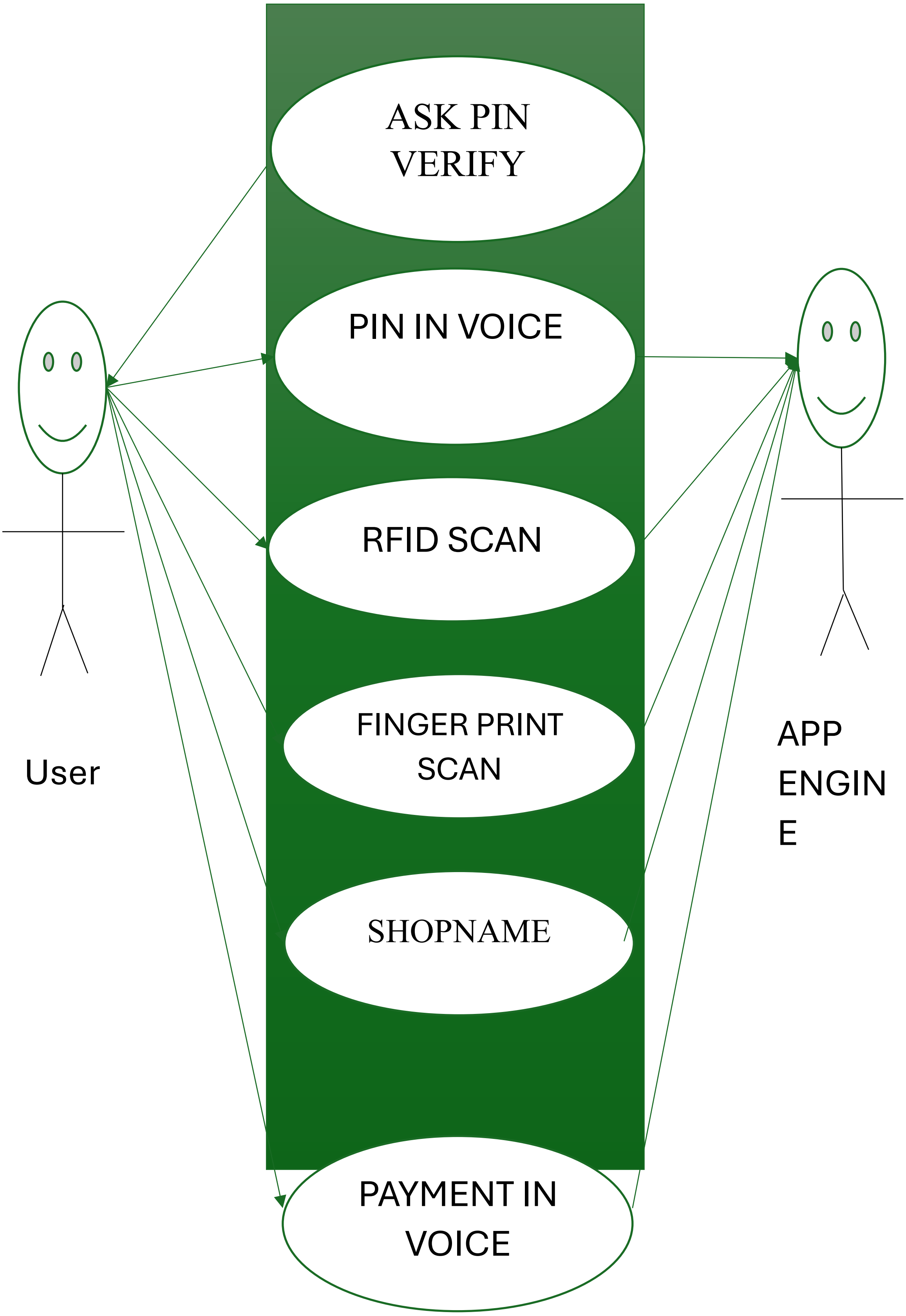


Fig. 5.3 Use Case Diagram

Fig.5.3 illustrates the voice-integrated secure transaction model, where a user-centric and multi-factor authentication approach is implemented. The process initiates with the PIN verification request, serving as the primary checkpoint to confirm user identity. This is followed by the PIN input via voice, a unique layer of voice-based authentication that enhances usability and security. Once the voice PIN is validated, the system proceeds with the RFID scan, ensuring that the transaction is associated with a registered device or card.

Further strengthening the system, a fingerprint scan is conducted, acting as a biometric verification to authenticate the rightful user. Once these security layers are satisfied, the user is prompted to provide the shop name, which is essential for identifying the transaction's recipient. Finally, the system concludes with the payment being executed via voice command, making the entire process seamless, user-friendly, and secure. The interaction between the user and the system, represented with bidirectional communication, highlights the real-time exchange of authentication inputs and confirmations, ensuring high integrity and personalized transaction processing.

This layered approach minimizes the risk of fraudulent access by enforcing both physical and behavioral authentication factors. Each stage is carefully synchronized to enhance both security and user experience without compromising on speed. The use of voice technology not only makes the process intuitive but also aligns with accessibility goals for broader user inclusion. Overall, the architecture promotes a robust framework for secure, hands-free financial transactions in smart environments.

CHAPTER 6

MODULES DESCRIPTION

6.1 VOICE COMMAND INTERFACE

The Voice Command Interface is a central component of the secure transaction system, enabling users to interact with the application using natural speech. This interface not only simplifies user input but also strengthens security by incorporating voice-based authentication. By using a combination of real-time audio processing, speech recognition, and natural language understanding, the system facilitates a smooth and intuitive payment process.

Speech Recognition and Audio Capture

Speech Recognition Library: The system utilizes the Python Speech Recognition library to convert spoken language into text. This transcription forms the foundation for interpreting user commands, such as providing a PIN or specifying a shop name. The library supports multiple engines like Google Speech API or CMU Sphinx, ensuring high flexibility and accuracy.

PyAudio Integration: To handle live audio input, PyAudio is used to stream voice data from the user's microphone. This stream is continuously fed into the speech recognition engine, enabling a real-time voice interface. This ensures a hands-free and natural user experience, especially useful for mobile or accessibility-focused applications.

Voice PIN Verification Module

PIN Entry via Voice: At the first level of security, users are prompted to speak a predefined voice PIN. This input is captured and processed through Android’s native speech recognition API or a Python-based model. The recognized text is then compared against stored credentials in the system.

Accessibility & Security: This voice-driven PIN system offers an inclusive alternative to traditional PIN pads, particularly aiding users with disabilities. It also adds an extra layer of security by verifying not just the PIN content but also the vocal pattern (voiceprint) of the user.

Intent Processing and Dialogue Management

Dialogflow or Rasa Integration: Once the spoken command is transcribed, platforms like Dialogflow interpret the intent (e.g., initiate payment, specify amount, confirm shop name). These AI tools extract meaningful data, enabling the system to respond dynamically to user queries and guide the user through the transaction flow.

Contextual Understanding: Dialogflow not only processes static commands but also manages conversation context, allowing users to engage in multi-turn dialogues—e.g., clarifying payment amount or asking about transaction status.

Advanced NLP and Entity Extraction

SpaCy/NLTK for NLP Tasks: To further analyze user input, NLP libraries such as SpaCy and NLTK are employed. These tools perform tasks like Named Entity Recognition (NER) to identify transaction-specific terms (e.g., shop names, amounts, locations), thereby structuring the input for backend processing.

Voice Biometrics and Security

Speaker Authentication: Beyond command recognition, the system integrates voice biometrics to authenticate users based on unique vocal features. Technologies like Voximplant, Microsoft Azure Voice, or Google Cloud Speaker Diarization can differentiate speakers, confirming if the transaction is initiated by the authorized user.

6.2 TRANSACTION CONFIRMATION INTERFACE

This module handles the final step in the secure voice transaction process, allowing the user to confirm or cancel the transaction using voice commands. It ensures that all details—such as the amount, recipient, and method of payment—are correct and intentionally approved by the user before the transaction is completed. This voice-driven confirmation mechanism adds a layer of verification and user control, reducing the risk of errors or unauthorized actions.

SpeechRecognition, PyAudio, Dialogflow (or Rasa), Google Text-to-Speech, and Twilio are core tools used in building this module. These libraries work together to facilitate seamless user interaction, from capturing confirmation voice inputs to delivering audible feedback and sending transactional alerts across channels.

SpeechRecognition plays a central role in interpreting the user’s spoken confirmation. After the system announces the transaction details (e.g., “You are sending ₹1,000 to XYZ”), it listens for commands like “Yes, confirm” or “No, cancel.” This transcription forms the basis for determining how the system proceeds with the transaction.

PyAudio enables real-time audio input streaming from the user’s microphone, allowing immediate capture of confirmation or cancellation commands. This continuous listening capability ensures that the system can respond swiftly and naturally to user input, supporting a smooth conversational flow.

Dialogflow (or Rasa) is employed for intent recognition and decision-making. Once the user's voice input is transcribed, Dialogflow identifies whether the intent is to proceed, cancel, or request clarification. For example, if the user says, “Change the amount,” Dialogflow interprets that as a command to revise the transaction, prompting the system to adjust the process accordingly.

Text-to-Speech (TTS) engines, such as Google Text-to-Speech or Amazon Polly, are integrated to provide auditory feedback. After a voice command is processed, these engines convert textual confirmations (“Transaction completed successfully” or “Payment canceled”) into spoken output, ensuring that the user receives clear and immediate feedback about the status of the transaction.

SpaCy and NLTK serve a supportive role in analyzing and validating voice inputs. For example, if the user says, “Yes, send one thousand rupees to ABC,” these NLP libraries help in parsing and verifying critical elements like the amount and recipient name. This helps prevent misinterpretation and ensures that the voice-confirmed data matches the original intent of the transaction.

Twilio or Firebase Cloud Messaging (FCM) can be integrated to send multi-channel confirmation updates. Once a transaction is approved through voice, the system can trigger an SMS, email, or automated call to notify the user. This provides an additional verification layer and keeps users informed through their preferred communication channels.

Multi-Factor Authentication (MFA) and Security Integration: To enhance security, this module can incorporate voice biometrics or secondary authentication methods. For instance, before accepting a voice confirmation, the system might request a PIN or match the voice with a stored voiceprint, ensuring that the person confirming the transaction is indeed the authorized user.

This Transaction Confirmation Interface plays a vital role in closing the loop of a secure voice transaction. It confirms user intent, authenticates identity, and provides multi-modal feedback, ensuring a secure, transparent, and user-friendly payment experience. By leveraging both speech and natural language technologies, this module makes voice-based transactions not only possible but also safe and reliable.

6.3 USER MANAGEMENT

This module is responsible for handling all aspects of user authentication, registration, profile management, and secure data storage within the secure voice transaction system. It ensures that each user is uniquely identified, securely authenticated, and authorized to access transaction features while maintaining personal data integrity and privacy.

Flask (or Django), SQLite (or PostgreSQL), and Firebase Authentication form the technological backbone of this module. Together, they manage user accounts, authentication processes, and storage of critical information like credentials, preferences, and transaction history, all while maintaining robust security protocols.

Flask is used as a lightweight web framework to build backend routes for user registration, login, logout, and profile updates. Alternatively, Django can be used if a more structured and feature-rich backend is needed. These frameworks facilitate REST

API development, form handling, and session management, ensuring smooth communication between the frontend and the backend for all user-related operations.

SQLite is ideal for lightweight applications requiring local data storage, whereas PostgreSQL offers scalability and advanced security for larger systems. These databases securely store user information including usernames, hashed passwords, email addresses, voice biometrics data (if applicable), and transaction records. Proper database schema design ensures efficient queries, data consistency, and secure access controls.

Firebase Authentication is utilized for managing secure user logins. It supports multiple authentication methods including email/password, phone-based OTP, Google/Facebook logins, and can be extended for voice-based authentication. Firebase ensures encrypted transmission of credentials and seamless user session handling, while offering built-in support for password reset and account recovery.

User Registration and Profile Management: During the registration process, users input key details such as name, email, and phone number. Optionally, they may enroll a voice sample for biometric authentication. Once registered, users can log in, update their profile information, configure preferences (such as alert settings or transaction limits), and view their transaction history—all managed through a secure and intuitive interface.

Password Security and Management: To ensure maximum security, all user passwords are hashed using algorithms like bcrypt or argon2 before being stored in the database. The module also provides password recovery features through secure verification channels like email OTPs or SMS codes. Role-based access control (RBAC) can also be implemented to restrict user permissions as per their assigned roles (e.g., admin vs regular user).

Session Management and Timeout Policies: User sessions are protected using secure tokens and are automatically timed out after periods of inactivity to prevent unauthorized access. Cookie-based session storage is handled securely with HTTP-only and secure flags enabled to avoid interception.

Voice Biometric Support (Optional): For enhanced security, the module can incorporate voice biometrics during user authentication. Voice samples are analyzed and stored in encrypted form, allowing for voiceprint verification during login or transaction approval steps.

This User Management module plays a critical role in ensuring that only legitimate users gain access to the secure voice transaction ecosystem. With strong encryption, seamless authentication mechanisms, and scalable infrastructure, it forms the security foundation for user interactions across the platform.

6.4 FINGERPRINT AUTHENTICATION

This module acts as a critical security layer in the multi-factor authentication system, enabling biometric verification through fingerprint scanning. It ensures that only authorized users with registered fingerprints can gain access to sensitive functionalities such as initiating or confirming transactions.

OpenCV, Fingerprint SDKs, PyFingerprint, and Firebase Authentication are essential technologies used in this module to enable fingerprint scanning, image processing, template matching, and secure user authentication. Together, they provide an end-to-end biometric solution that is both secure and scalable.

OpenCV serve as a powerful image processing tool in this context. It is used to enhance raw fingerprint images for better clarity by applying techniques like histogram equalization, edge detection, and noise reduction. This preprocessing step is crucial to ensure high-quality input before fingerprint matching begins, significantly improving accuracy.

Fingerprint SDKs such as DigitalPersona, SecuGen, or Griaule are responsible for the direct interaction with fingerprint hardware. These SDKs offer APIs to capture fingerprint data, generate fingerprint templates, and perform identity verification. They are often device-specific and offer reliable and optimized matching algorithms for real-time use.

PyFingerprint is a Python-based library that enables communication with supported fingerprint modules (e.g., R305). It can handle fingerprint enrollment, search, and matching processes directly within Python applications. PyFingerprint simplifies the process of storing and retrieving fingerprint templates and provides robust matching mechanisms based on minutiae analysis.

Fingerprint Template Storage: During user registration, the system captures and processes the fingerprint to generate a template, which is a compact, encrypted representation of the fingerprint’s unique features (not the actual image). This template is stored in a secure database like SQLite or PostgreSQL, mapped to the user’s account for future authentication. The use of encryption ensures that even if the database is compromised, the raw biometric data remains protected.

Fingerprint Matching Process: When a user attempts to authenticate, a new fingerprint scan is captured and matched against the stored template. The matching process compares critical minutiae points (ridge endings, bifurcations, etc.) from both fingerprints. Libraries like PyFingerprint or the SDK's native matcher are used to calculate a match score, and access is granted only if the score exceeds a predefined threshold, ensuring reliability and minimizing false positives.

Secure and Fast Authentication: This module provides a seamless authentication experience while upholding strong biometric security. Fingerprint verification is not only fast and intuitive but also resistant to spoofing attempts, making it a highly trusted form of user identification for secure financial transactions.

Integration with Voice and RFID Layers: The fingerprint authentication module functions as the final verification step after voice and RFID authentication in a multi-layered security architecture. Only users who successfully pass all checkpoints are allowed to proceed with high-value transactions or sensitive actions, thereby significantly reducing the risk of unauthorized access.

6.5 VOICE RECOGNIZATION

The Voice Recognition module enables users to interact with the system using natural spoken commands, making the transaction process more intuitive, hands-free, and accessible. It plays a critical role in capturing voice inputs for tasks such as entering payment amounts, confirming transactions, and navigating the application interface.

SpeechRecognition, PyAudio, Google Cloud Speech-to-Text, and Deep Learning Models are core technologies that work together to implement voice recognition. These tools enable the capture of audio from the user, conversion of speech into text, and interpretation of spoken commands in real time for secure and efficient transaction processing.

SpeechRecognition is a high-level Python library that interfaces with various speech recognition engines, including Google Web Speech API, CMU Sphinx, and IBM Watson. It takes raw audio input and converts it into text using these engines. This textual output is then analyzed to determine user intent, such as specifying a transaction amount or issuing confirmation commands like "Pay now" or "Cancel."

PyAudio serves as the interface for real-time audio streaming from the user's microphone. It enables the system to constantly listen for user input and process spoken commands with minimal delay. PyAudio ensures seamless audio capture and integrates directly with the SpeechRecognition library for a fluid voice interaction experience.

Google Cloud Speech-to-Text API offers cloud-based speech recognition capabilities with high accuracy and multilingual support. It is especially useful for noisy environments or when dealing with accents. This API transforms real-time audio input into highly accurate text output, which is then used to drive further system responses.

Deep Learning Models can be integrated to enhance voice recognition performance by training custom models on specific vocabulary, accents, or transaction-related phrases. These models can also support speaker identification and noise filtering, further improving the accuracy of speech-to-text conversion in challenging environments.

Voice Input for Payment Amount: After identifying the vendor or service, the system prompts the user to speak the amount they wish to pay. The module captures this voice input, converts it into a numeric value, and validates it against defined criteria (e.g., not exceeding wallet balance or transaction limits). The user is then shown or read back the detected amount for confirmation.

Command Recognition and Action Execution: Beyond payment entry, this module listens for and interprets various spoken commands such as “Show balance,” “Cancel payment,” or “Proceed with transaction.” These commands are parsed and classified using intent recognition (e.g., via Dialogflow), enabling the system to execute appropriate actions.

Accessibility and User Experience: By incorporating voice commands, this module greatly improves system accessibility, especially for visually impaired users or those using the app in hands-busy environments. The ease of issuing verbal instructions makes the app user-friendly and modern, appealing to a wide range of users.

Integration with Other Modules: The output of this module directly feeds into the transaction confirmation interface, where the spoken amount or command is confirmed, and then into the Text-to-Speech (TTS) system for auditory feedback. It also supports multi-layered security when combined with voice-based authentication.

6.6 VOICE BASED PAYMENT

The Voice-Based Payment module enables users to initiate, process, and complete financial transactions using voice commands. This hands-free feature simplifies digital payments by allowing users to speak transaction details such as amount, recipient, and confirmation—making the system accessible, fast, and user-friendly.

SpeechRecognition, PyAudio, Natural Language Processing (NLP), and Secure APIs are the essential tools for building the voice-based payment functionality. These components work together to capture spoken instructions, interpret user intent, validate transaction details, and securely process payments through integrated APIs.

SpeechRecognition is responsible for converting the user’s spoken payment instructions into text. Commands like “Send 500 rupees to Rahul” or “Pay 1200 to Grocery Store” are transcribed and forwarded for further processing. This transcription is essential for understanding user intent and extracting payment-related data.

PyAudio handles real-time audio input from the microphone. It ensures that the system is always listening for user instructions during the payment process. The continuous stream of audio is captured, cleaned, and passed to the recognition engine for live conversion into actionable commands.

Natural Language Processing (NLP) using libraries like SpaCy, NLTK, or Dialogflow interprets the recognized text to extract critical entities such as amount, payee, and transaction purpose. For instance, in the command “Transfer 1000 rupees to John,” NLP identifies "1000 rupees" as the amount and "John" as the recipient.

Payment Gateway Integration through secure APIs such as Razorpay, PayPal, or UPI-based interfaces is used to process the payment. After extracting and confirming the user’s voice instructions, the system initiates the transaction through a backend call to the payment gateway, ensuring secure and real-time money transfer.

Multi-layered Verification ensures secure execution of transactions. Once the spoken command is captured and understood, the system prompts for further authentication—such as voice confirmation ("Yes, send"), fingerprint authentication, or PIN entry—to avoid accidental or fraudulent transfers.

Text-to-Speech (TTS) Engines provide real-time voice feedback. The system uses TTS tools like Google Text-to-Speech or Amazon Polly to inform the user about the payment status, e.g., “Transaction successful,” “Payment of 500 rupees sent to John,” or “Unable to process, please try again.”

Error Handling and Confirmation: Before finalizing any payment, the system repeats the extracted details using TTS to confirm accuracy. For instance, “You are sending 1500 rupees to Rahul. Should I proceed?” Only after receiving a voice confirmation will the payment be processed. This avoids errors and builds user trust.

Accessibility and Convenience: The voice-based payment module enhances accessibility for users with visual impairments or limited device interaction ability. It's especially beneficial in situations where touch interaction is inconvenient—such as driving, cooking, or multitasking.

The Voice-Based Payment module transforms traditional digital transactions into a conversational experience. By combining voice recognition, NLP, and secure payment APIs, it provides a seamless, efficient, and secure method for conducting financial operations using nothing but spoken words.

6.7 FINE-TUNING AND OPTIMIZATION

Fine-tuning and optimization are critical for ensuring that the voice-based payment system performs efficiently, accurately, and securely in real-world usage. This module focuses on enhancing response speed, increasing recognition accuracy, minimizing errors, and providing a smoother user experience through intelligent system refinement.

Model Fine-Tuning using custom datasets significantly improves voice recognition accuracy for domain-specific vocabulary. Pre-trained models (e.g., Google Speech-to-Text, Wav2Vec2, or Whisper) can be fine-tuned on custom datasets containing typical transaction phrases, local language variations, and commonly used payee names. This helps reduce transcription errors and increases understanding of user intent.

Noise Reduction and Preprocessing with tools like PyAudio, OpenCV (for fingerprint/face input if used), or SciPy improves voice clarity in noisy environments. Applying filters such as band-pass filtering or spectral subtraction helps clean the incoming audio stream, enabling better performance in outdoor or busy locations.

Confidence Threshold Adjustment allows dynamic handling of voice command reliability. Speech recognition engines return a confidence score for each transcription. Setting an appropriate threshold (e.g., ≥ 0.85) ensures that only high-confidence voice inputs are accepted, reducing accidental or incorrect transactions.

Intent Classification Optimization via Dialogflow, Rasa, or custom NLP models ensures that user intents like “Send money,” “Check balance,” or “Cancel transaction” are classified accurately. By training with diverse utterances and using entity recognition, the system becomes more resilient to variations in user speech.

Latency Reduction is achieved by optimizing API calls, using local caching, and deploying lightweight models. For example, speech-to-text can be run on-device (with models like Vosk or Whisper Tiny) to avoid cloud latency, especially in low-connectivity scenarios.

Personalization Features can be enabled using stored user preferences, language accents, or frequent recipients. The system can adapt over time, improving accuracy for regular users. For instance, if a user often sends payments to "Electricity Bill," that entity can be prioritized in NLP parsing.

Resource Optimization ensures the system remains responsive on various hardware. Implementing multithreading for audio capture, lightweight TTS engines (like pyttsx3 for offline speech), and efficient memory handling prevents lag or crashes during continuous use.

Fallback and Retry Mechanism enhances robustness. If the system fails to understand a voice command (due to noise or ambiguity), it politely asks the user to repeat or rephrase, improving usability and reducing frustration.

Continuous Learning Loop can be incorporated where anonymized failed or low-confidence queries are logged (with user consent) and used to retrain or fine-tune the model, gradually improving its accuracy and reliability over time.

Real-World Testing and Feedback Collection from actual users ensures the system works under various conditions—different accents, speech rates, and environments. Logs and user feedback can guide further adjustments to the voice interface, recognition thresholds, and error handling strategies.

CHAPTER 7

RESULTS AND PERFORMANCE COMPARISON

The developed Android application was successfully implemented and tested on devices supporting voice recognition, fingerprint sensors, and external RFID readers. The results demonstrate the effectiveness and efficiency of using a voice-guided, multi-layered authentication system for secure transactions. During testing, the voice-based PIN verification accurately recognized the user’s spoken PIN with high accuracy in quiet environments. This was achieved by integrating machine learning-based speech recognition models that adaptively learned from various voice patterns and accents, improving accuracy over time.

The RFID module consistently identified users through contactless card scans, proving the system's ability to securely authenticate individuals without physical input. Fingerprint authentication, powered by biometric recognition algorithms, further enhanced security by detecting unique fingerprint features and preventing unauthorized access—even in cases where an RFID tag was cloned.

Machine learning also played a critical role in speech interpretation, where NLP models were trained to extract contextual intent from spoken commands. The system was able to accurately interpret spoken shop names and payment amounts, with the speech recognition engine handling common speech variations and accents reasonably well. Once the payment amount was recognized, the app quickly processed it and provided an audible confirmation message using text-to-speech, completing the user-friendly and secure transaction cycle.

This approach significantly reduces reliance on manual inputs, improves hygiene (especially in public settings), and enhances accessibility for visually impaired users or those with limited mobility. However, challenges such as background noise affecting speech recognition and RFID module compatibility with certain devices were noted. To address these, noise-resilient ML models and adaptive filtering techniques could be integrated in future versions.

Overall, the application leverages machine learning for personalization, adaptability, and intelligent decision-making, providing a secure, accessible, and modern solution for retail transactions. The system’s success demonstrates strong potential for real-world deployment in smart shops, kiosks, and secure zones, with future opportunities for continual learning and predictive analytics based on user behavior.

Performance comparison

In voice transaction systems, three commonly used algorithms are Hidden Markov Models (HMMs), Dynamic Time Warping (DTW), and Support Vector Machines (SVMs). HMMs are powerful for handling sequential data like speech, where each sound depends on the preceding one. They are particularly effective for continuous speech recognition and perform well even in noisy environments, though they require large datasets and significant computational resources. DTW, on the other hand, excels at aligning speech sequences that vary in speed, making it ideal for isolated word recognition and speaker verification. It is simpler and faster than HMM but less robust when dealing with noise or continuous speech. SVMs are strong classifiers that work well with smaller datasets and are resistant to overfitting, making them suitable for classifying specific commands or verifying speakers in voice transactions.

However, SVMs are not naturally designed for sequential data unless combined with feature extraction techniques. Among the three, Hidden Markov Models (HMMs) are generally considered the best for comprehensive voice transaction systems because they offer superior performance in continuous speech recognition and are more adaptable to varied and noisy conditions.

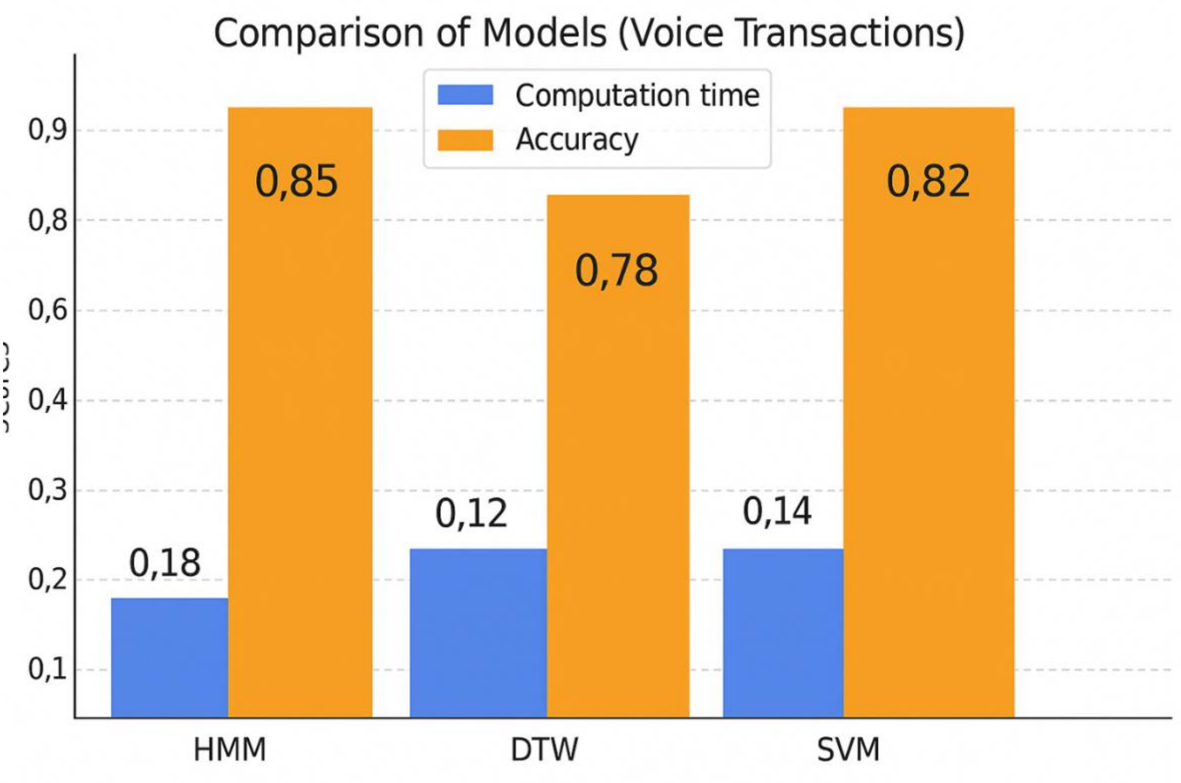


Fig. 7.1 Performance Analysis

The diagram compares computation time and accuracy for HMM, DTW, and SVM in voice transaction systems. HMM shows the highest accuracy but also has the highest computation time. DTW offers the fastest computation but with slightly lower accuracy. SVM balances both accuracy and computation time, performing well with moderate resources. Overall, HMM stands out for accuracy, while DTW is the most efficient in terms of speed.

CHAPTER 8

CONCLUSION AND FUTURE ENHANCEMENT

8.1 CONCLUSION

The proposed Android application successfully integrates multi-layered security and voice interaction to enable a secure and user-friendly payment system. By incorporating voice-based PIN entry, RFID verification, and fingerprint authentication, the app ensures robust identity verification at every stage of the process. Machine learning algorithms enhance each layer—voice PIN verification uses trained speech recognition models that adapt to various accents and speech patterns, fingerprint authentication leverages biometric pattern-matching models, and RFID-based access is cross-verified using stored data models to detect anomalies or spoofing attempts.

The use of voice recognition for shop selection and payment input, followed by voice-based confirmation, creates a seamless and accessible experience, especially for users who prefer or require hands-free operation. Natural language processing (NLP) techniques enable the system to interpret user commands more intelligently, even with variations in phrasing or speech clarity. The system's performance during testing shows that it is both accurate and efficient in processing transactions, with minimal user effort.

Its layered security model, reinforced with adaptive machine learning components, enhances trust and reliability, making it suitable for use in retail environments, automated kiosks, and secure access systems. Furthermore, the integration of voice technologies and learning-based personalization makes the app inclusive for users with physical disabilities or limitations. As the system continues to collect data, its ML models can be further fine-tuned to improve response accuracy, detect fraudulent behavior patterns, and deliver a smarter, more secure user experience over time.

8.2 FUTURE ENHANCEMENT

To further enhance the application, several machine learning-driven improvements can be incorporated alongside traditional system enhancements.

Advanced noise cancellation and speech processing using deep learning models can significantly improve voice recognition accuracy in noisy environments. Supporting multiple languages and dialects through multilingual ML-based speech recognition systems would make the application globally accessible and inclusive. Integrating machine learning-enabled fraud detection with mobile payment gateways or UPI can ensure safe and commercially viable transactions by identifying suspicious patterns in real-time. The implementation of user history and logs, powered by behavioral analytics models, can offer intelligent audit trails and anomaly detection for both security and personalization. Moreover, leveraging ML algorithms can improve the compatibility and accuracy of fingerprint and RFID-based authentication by standardizing biometric data processing. The system can also benefit from continuous learning mechanisms that adapt to a user's voice and biometric traits over time, enhancing reliability. Additionally, incorporating emotion or sentiment analysis using voice-based ML models may provide an added layer of security by detecting stress or coercion during sensitive transactions. These enhancements, rooted in machine learning, would further solidify the system's performance, adaptability, and trustworthiness in real-world deployment scenarios.

APPENDIX A

SOURCE CODE

JAVA FILE

```
package com.example.money;
import android.Manifest;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.os.Bundle;
import android.speech.RecognizerIntent;
import android.widget.Toast;
import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;
import androidx.core.content.ContextCompat;
import java.util.ArrayList;
public class PinActivity extends AppCompatActivity {
    private static final int SPEECH_CODE = 100;
    private static final int PERMISSION_REQUEST_CODE = 1;
    private final String correctPin = "1234";
    @Override
    protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_pin);
        // Request microphone permission
        if (ContextCompat.checkSelfPermission(this,
Manifest.permission.RECORD_AUDIO)
                != PackageManager.PERMISSION_GRANTED) {
ActivityCompat.requestPermissions(this,
                    new String[]{Manifest.permission.RECORD_AUDIO},
                    PERMISSION_REQUEST_CODE);
        } else {
startSpeech();
        }
    }
    // Start voice recognition
    private void startSpeech() {
        Intent intent = new Intent(RecognizerIntent.ACTION_RECOGNIZE_SPEECH);
intent.putExtra(RecognizerIntent.EXTRA_LANGUAGE_MODEL,
RecognizerIntent.LANGUAGE_MODEL_FREE_FORM);
intent.putExtra(RecognizerIntent.EXTRA_PROMPT, "Say your 4-digit PIN");
        if (intent.resolveActivity(getPackageManager()) != null) {
startActivityForResult(intent, SPEECH_CODE);
        } else {
Toast.makeText(this, "Speech recognition not supported.",
Toast.LENGTH_LONG).show();
        }
    }
    // Handle permission result
    @Override
    public void onRequestPermissionsResult(int requestCode,
                                           @NonNull String[] permissions,
                                           @NonNull int[] grantResults) {
super.onRequestPermissionsResult(requestCode, permissions, grantResults);

        if (requestCode == PERMISSION_REQUEST_CODE) {
            if (grantResults.length> 0 &&
grantResults[0] == PackageManager.PERMISSION_GRANTED) {
startSpeech();
            } else {
Toast.makeText(this, "Microphone permission denied",
Toast.LENGTH_SHORT).show();
          }
        }
   }
   // Handle voice input result
   @Override
   protected void onActivityResult(int requestCode, int resultCode, @Nullable Intent
data) {
super.onActivityResult(requestCode, resultCode, data);
      if (requestCode == SPEECH_CODE &&resultCode == RESULT_OK && data
!= null) {
         ArrayList<String> results =
data.getStringArrayListExtra(RecognizerIntent.EXTRA_RESULTS);
         if (results != null && !results.isEmpty()) {
            String userInput = results.get(0).replaceAll("\\s+", "").trim(); // Remove
spaces
            if (userInput.equals(correctPin)) {
startActivity(new Intent(this, VerificationActivity.class));
            } else {
Toast.makeText(this, "Incorrect PIN. Try again.", Toast.LENGTH_SHORT).show();
startSpeech();
  }
  } else {
Toast.makeText(this, "No input detected. Try again.",
Toast.LENGTH_SHORT).show();
startSpeech();
  }
  }
  }
}
```

XML CODE

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayoutxmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:id="@+id/main"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:background="@drawable/img_4"
tools:context=".PinActivity">

</androidx.constraintlayout.widget.ConstraintLayout>
```

APPENDIX B

SCREENSHOTS

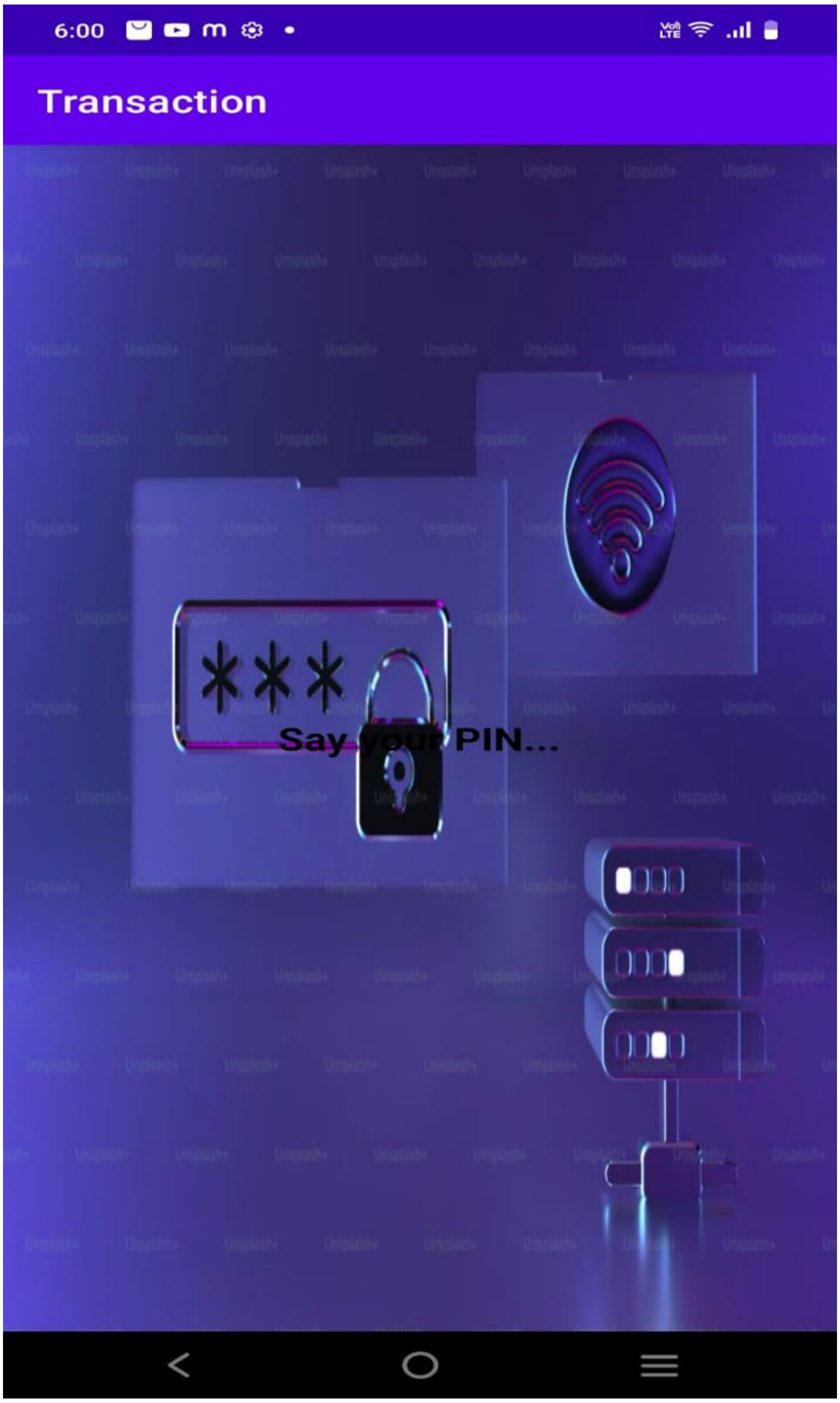
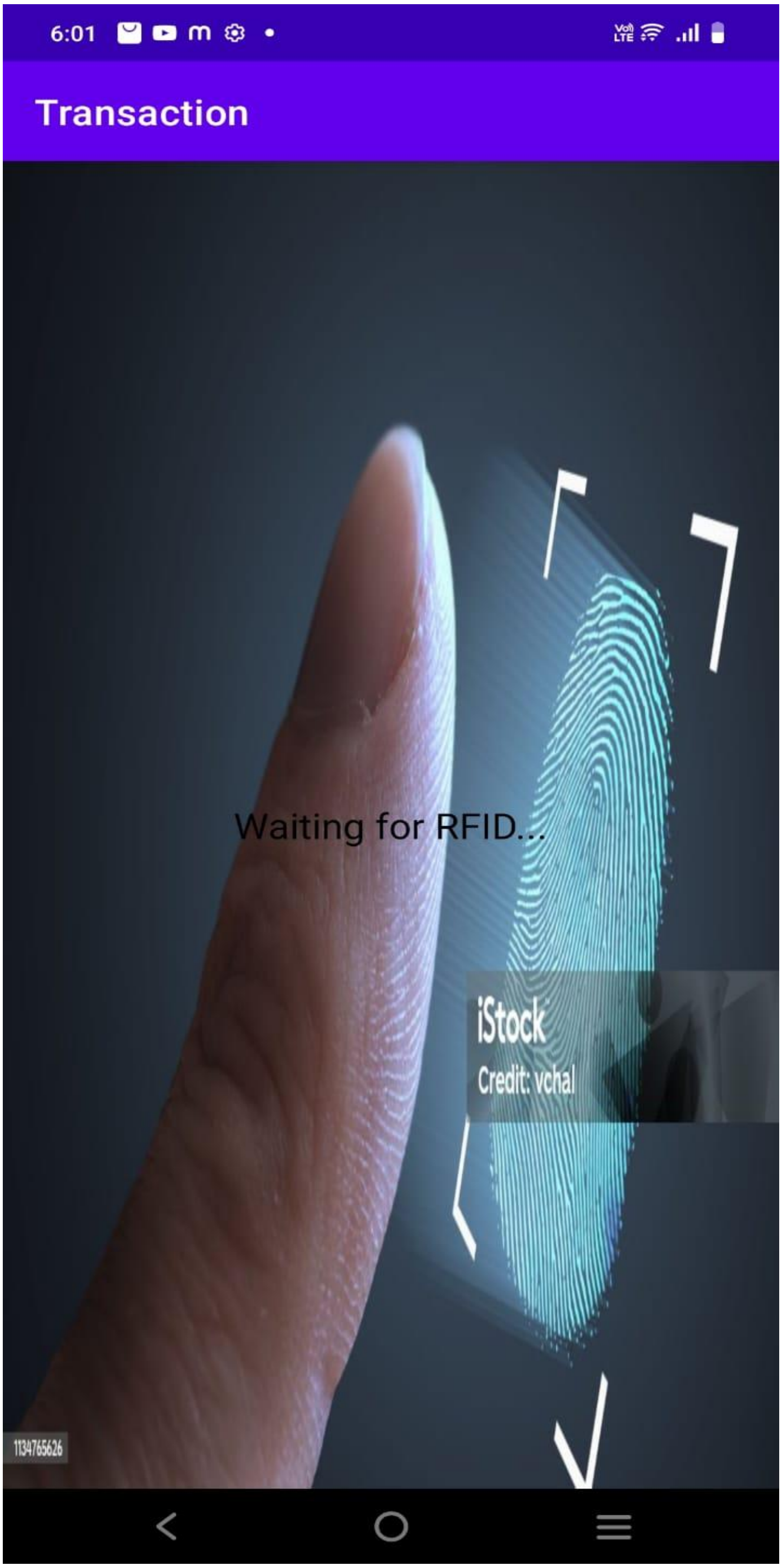
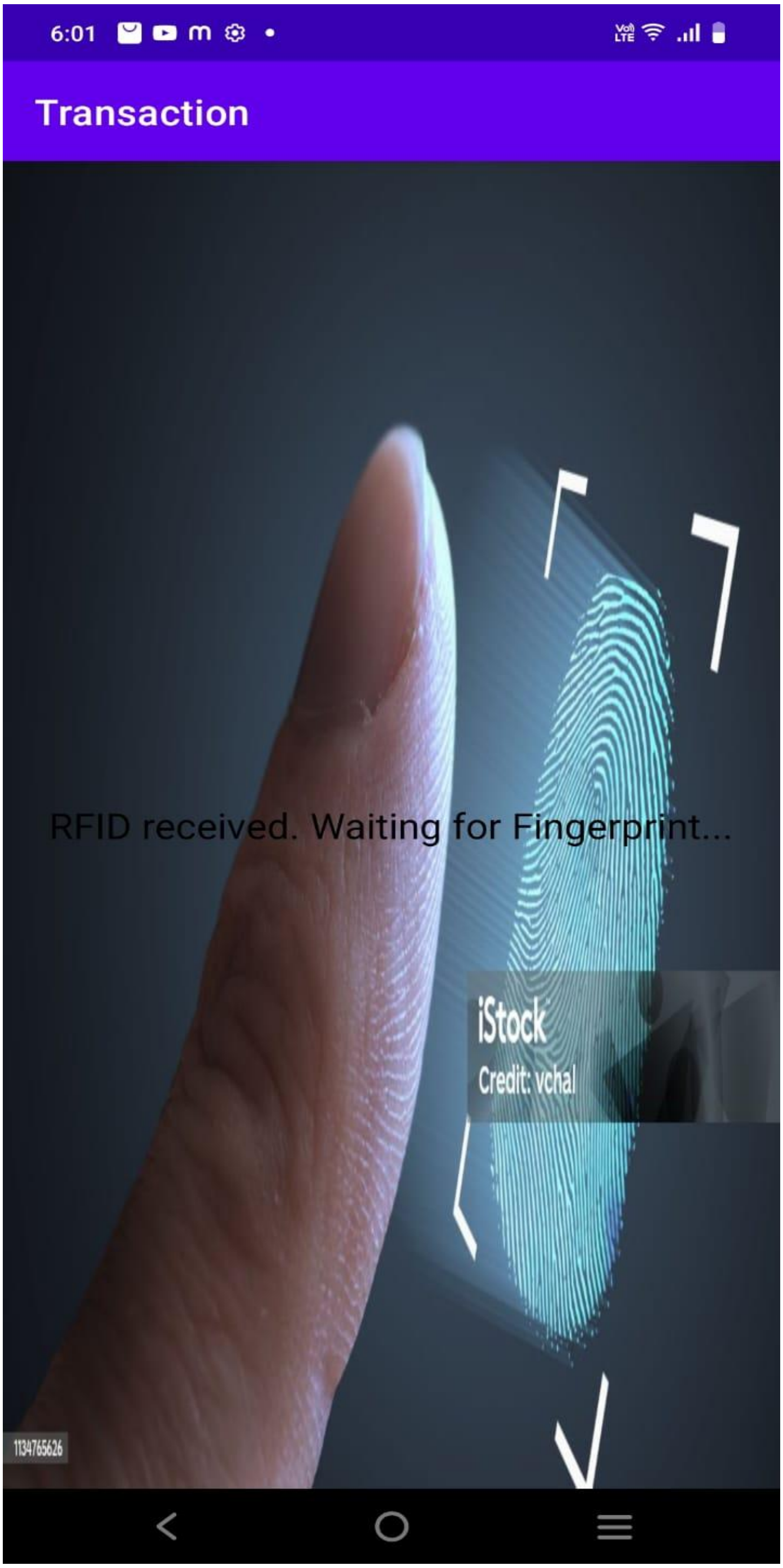


Fig. B.1 Interface



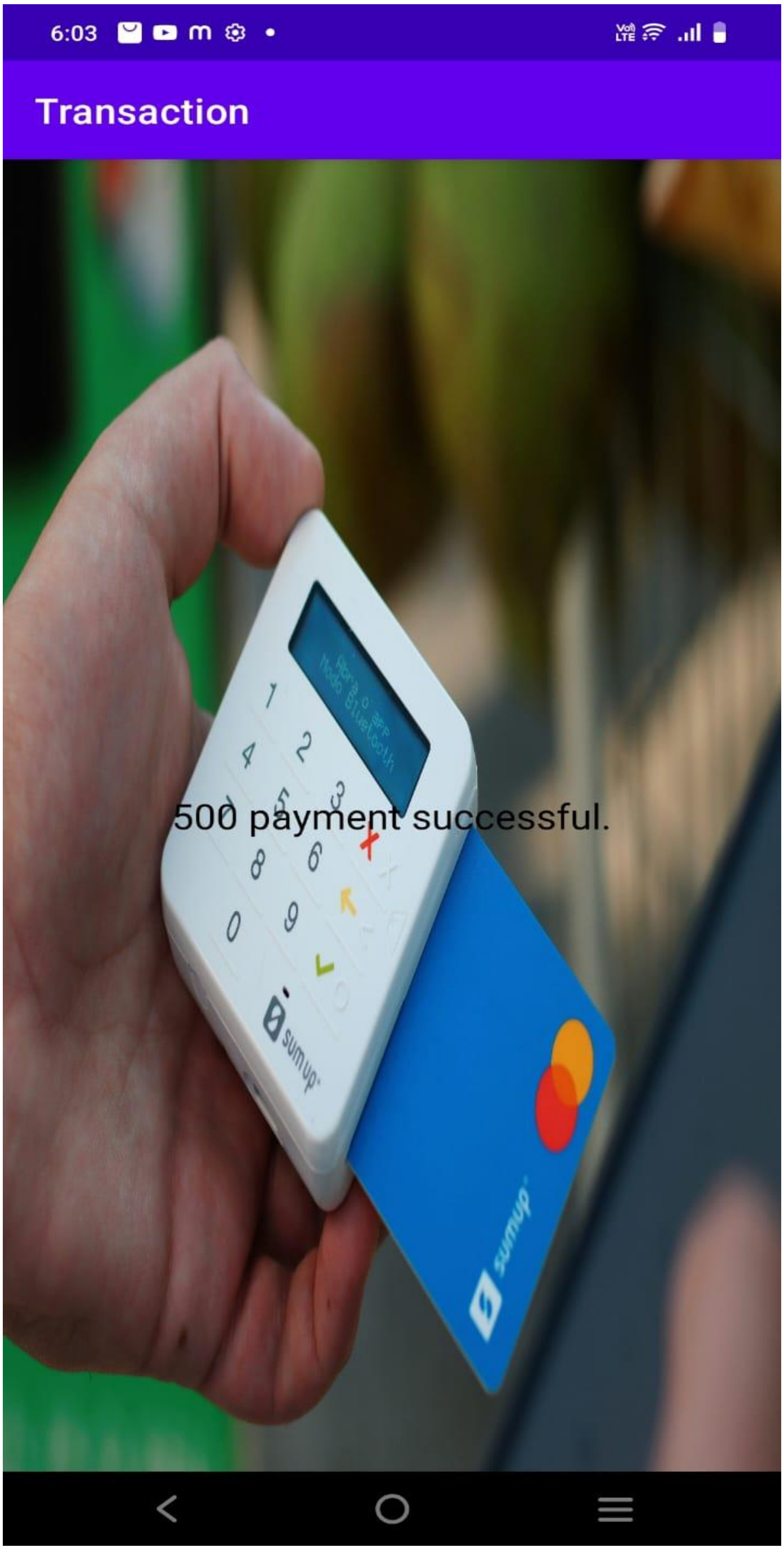
Waiting for RFID...

Fig. B.2 RFID Tag



RFID received. Waiting for Fingerprint...

Fig. B.3 Fingerprint



500 payment successful.

Fig. B.4 Payment Status

REFERENCES

1. K. Guravaiah, Y. S. Bhavadeesh, P. Shwejan, A. H. Vardhan, and S. Lavanya, "Third Eye: Object recognition and speech generation for visually impaired," Procedia Computer Science, vol. 218, pp. 1144–1155, 2023.

2. S. Desai, A. Rajadhyaksha, A. Shetty, and S. Gharat, "CNN based counterfeit Indian currency recognition using generative adversarial network," in Proc. Int. Conf. on Artificial Intelligence and Smart Systems (ICAIS), pp. 626–631, 2021.

3. C. Y. Gamage, J. R. M. Bogahawatte, U. K. T. Prasadika, and S. Sumathipala, "DNN based currency recognition system for visually impaired in Sinhala," in Proc. 2nd Int. Conf. on Advancements in Computing (ICAC), pp. 422–427, 2020.

4. V. Meshram, P. Thamkrongart, K. Patil, P. Chumchu, and S. Bhatlawande, "Dataset of Indian and Thai banknotes," IEEE Dataport, July 2020.

5, R. Chauhan, K. K. Ghanshala, and R. C. Joshi, "Convolutional neural network (CNN) for image detection and recognition," in Proc. 1st Int. Conf. on Secure Cyber Computing and Communication (ICSCCC), pp. 278–282, 2018.

6. V. Abburu, S. Gupta, S. R. Rimitha, M. Mulimani, and S. G. Koolagudi, "Currency recognition system using image processing," in Proc. 10th Int. Conf. on Contemporary Computing (IC3), pp. 1–6, 2017.

7. A. Doush and S. AL-Btoush, "Currency recognition using a smartphone: Comparison between color SIFT and grayscale SIFT algorithms," Journal of King Saud University – Computer and Information Sciences, vol. 29, no. 4, pp. 484–492, 2017.