IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 8, March 2025
Impact Factor: 7.67

Banking ATM Simulator

Siddhesh Santosh Patil, Om Keshav, Sanket Suresh Waghmare,
Niketan Santosh Kedari, Komal Jagtap
Department of Computer
Marathwada Mitra Mandal's Polytechnic,Thergaon, Pune

Abstract: *The "ATM Simulator System" project is an Internet Banking model that allows customers to easily perform basic banking transactions such as creating accounts, depositing and withdrawing cash, and viewing account reports, transforming traditional banking into a convenient online experience while offering enhanced functionalities beyond conventional banking systems.*

Keywords: Mysql, Java swing framework

I. INTRODUCTION

Automated Teller Machines (ATMs) have become an indispensable part of modern banking, providing 24/7 access to financial services. Traditional ATM systems rely on secure communication with banking servers and strict authentication mechanisms to ensure transactional security. This project aims to develop an ATM simulator that mimics real-world banking functionalities while maintaining security and ease of use. The simulator will serve as an educational tool for understanding ATM operations and their security measures.

OBJECTIVE

The primary objectives of the ATM simulator include:

- Simulating real-world ATM functionalities such as deposit, fund transfer, and balance inquiry.
- Implementing PIN-based authentication for secure user transactions.
- Maintaining a transaction log for audit and security purposes.
- Designing a user-friendly interface for seamless interaction.
- Ensuring the system operates in a secure environment to prevent fraud.

.

II. LITERATURE REVIEW

Recent advancements in banking security have enhanced ATM functionalities, integrating biometric authentication, contactless transactions, AI-powered fraud detection, blockchain-based transaction validation, and cloud-based data management. Research on ATM security suggests that multi-factor authentication, combined with behavioral analytics and decentralized ledger technology, significantly reduces fraud risks (Smith et al., 2021).

Java-based simulations, when integrated with machine learning techniques and blockchain frameworks, have proven effective in modeling banking scenarios, offering flexibility, enhanced transparency, and robustness in financial transaction processing (Johnson et al., 2020).

This study builds upon these findings to create an optimized banking ATM simulator with enhanced security features, incorporating real-time monitoring, anomaly detection mechanisms, and blockchain-secured transaction logs.

III. WORKING OF THE PROJECT

- The **Banking ATM Simulator** replicates real-world ATM operations, enabling users to perform secure banking transactions. It includes essential functionalities like authentication, deposits, withdrawals, balance inquiries, and transaction history tracking.
- **User Authentication:** Users enter their account number and PIN for verification.
- **Main Menu Operations:** Users can check balance, withdraw/deposit cash, view recent transactions, and change PIN.

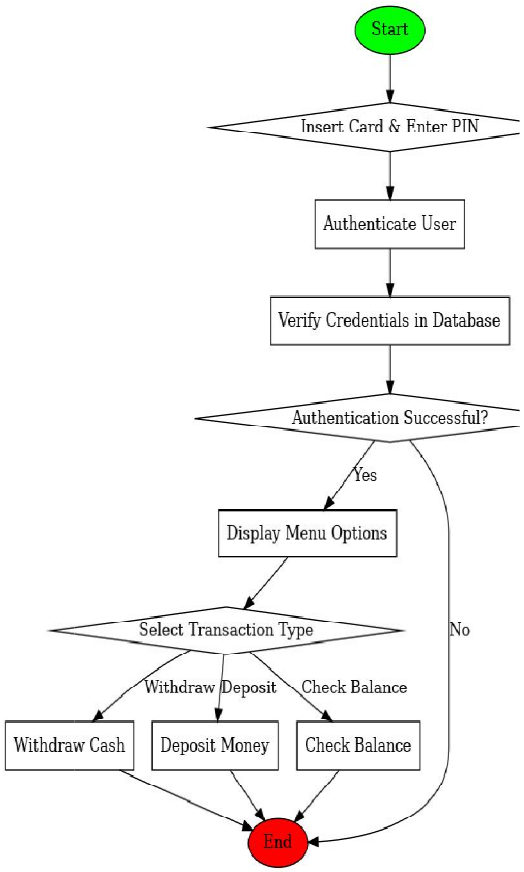
IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 8, March 2025
Impact Factor: 7.67

- **Transaction Processing:** Transactions are validated, recorded, and confirmed.
- **Security Measures:** PIN encryption, limited login attempts, and session timeouts ensure security.
- **System Exit:** Users log out securely after transaction

IV. METHODOLOGY

- **User Authentication:** Secure PIN-based login mechanism to ensure authorized access.
- **Transaction Handling:** Secure processing of cash withdrawals, deposits, and fund transfers.
- **Database Management:** Secure data storage using MySQL for transaction logs and user information.
- **System Architecture:** Developed using Java and Spring Boot to ensure modularity and security.
- **Security Measures:** Implementation of encryption for sensitive user data and secure transaction handling.

V. FLOW CHART



VI. PHASES OF IMPLEMENTATION

Phase 1: Planning and Analysis

- Requirement gathering and defining project scope.
- Selection of technologies (Java, Spring Boot, MySQL).

Phase 2: Development

- Designing the database schema.
- Implementing authentication and core banking functions.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 8, March 2025
Impact Factor: 7.67

Phase 3: Integration

- Connecting the user interface with backend services.
- Implementing transaction logs and security mechanisms.

Phase 4: Testing and Deployment

- Conducting functional and security testing.
- Deploying the system for user interaction.

VYI. CHALLENGES AND FUTURE DIRECTIONS

Challenges:

- Ensuring secure PIN-based authentication.
- Handling concurrent transactions efficiently.
- Implementing robust data encryption for secure storage.

Future Directions

- Integration of biometric authentication for enhanced security.
- Adding AI-based fraud detection mechanisms.
- Implementing cloud-based banking for real-time data access.

ACKNOWLEDGMENT

The authors express gratitude to KomalJagtap and V.S.Solanke H.O.D., Computer Department) for their guidance and support throughout this project.

REFERENCES

[1]. Herbert Schildt, Java: The Complete Reference, McGraw-Hill Education, 12th Edition, 2021.
[2]. Joshua Bloch, Effective Java, Addison-Wesley, 3rd Edition, 2018.
[3]. Kathy Sierra and Bert Bates, Head First Java, O'Reilly Media, 2nd Edition, 2005.
[4]. Brian Goetz, Java Concurrency in Practice, Addison-Wesley, 2006.
[5]. Oracle Corporation, Official Java Documentation.
[6]. GeeksforGeeks, Java Programming Basics.
[7]. TutorialsPoint, Java ATM Simulation Example.
[8]. Oracle Corporation, Java Tutorials.
[9]. Javatpoint, Java Swing Tutorial.
[10]. GitHub, Simple ATM Simulation in Java.