

ATM Software System

Table of Contents

1. Introduction
2. Project Objectives
3. System Requirements
4. Implementation
5. Testing
 - Test Plan
 - Test Cases
 - Results
6. User Documentation
 - How to Use the System
7. Conclusion

Introduction

- Provide an overview of the ATM software system, its purpose, and its importance.
- ATMs are essential for providing banking services round-the-clock, offering features such as cash withdrawals, deposits, and balance inquiries.
- Highlight the significance of developing a secure and efficient ATM software system using Python.

- Define the boundaries of the project.
- The project covers the design, development, and testing of the ATM software system.
- Key functionalities include user authentication, transaction processing, account management, and database interaction.

Project Objectives

The project aims to implement secure user authentication, validate credentials, and safeguard against unauthorized access. It will develop functionalities for efficient transaction handling, including cash withdrawals, deposits, and balance inquiries. The system will manage user accounts, facilitating creation and maintenance. A secure, optimized backend database will be designed for user info and transactions. An intuitive user interface will be developed with clear instructions. Comprehensive security measures and detailed logging for auditing will be implemented. The system will be scalable, reliable, and capable of handling multiple users and transactions concurrently.



Features

1. Account Statement:

- Users can view their account statements.

2. Withdrawing Amount:

- Users can withdraw money, and the system updates and displays the remaining balance.

3. Depositing Amount:

- Users can deposit money, and the system calculates and shows the new total balance.

4. Change PIN:

- Users can change their PIN by entering a new code and confirming it.

System Requirements

01

Functional
Requirements

02

Non-Functional
Requirements

03

Hardware
Requirements

04

Software
Requirements:

Implementation



- Modules and Functions:
 - Describe each module in detail.
 - User Authentication Module: Handles user login and authentication.
 - Transaction Module: Manages cash withdrawals, deposits, and balance inquiries.
 - Account Management Module: Supports account creation and maintenance.
 - Database Module: Manages data storage and retrieval.
- Code Snippets:
 - Provide key sections of code demonstrating important functionalities.
 - Example: User authentication process.
 - Example: Transaction processing (withdrawal, deposit, balance inquiry).

Test Plan:

- Overview of the testing strategy.
- Types of testing to be conducted: Unit testing, integration testing, system testing, and acceptance testing.
- Test environment setup and configuration.

Testing

Results:

Summary of testing results. Issues encountered and their resolutions. Performance metrics and system reliability.

- Sample test cases with expected outcomes.
- Test Case 1: User login with valid credentials.
- Test Case 2: User login with invalid credentials.
- Test Case 3: Cash withdrawal with sufficient balance.
- Test Case 4: Cash withdrawal with insufficient balance.
- Test Case 5: Deposit transaction.
- Test Case 6: Balance inquiry.

User Documentation

How to Use the System:

Steps for user login: Enter user ID and PIN.

Performing transactions:

Cash Withdrawals: Select withdrawal option, enter amount, confirm transaction.

Deposits: Select deposit option, enter amount, insert cash, confirm transaction.

Balance Inquiries: Select balance inquiry option, view current balance and recent transactions.

Managing accounts: Access account settings, update personal information.

Logging out: Select logout option to securely exit the system.

Conclusion

- Developed a secure and efficient ATM software system.
- Implemented user authentication, transaction processing, and account management functionalities.
- Ensured data security and integrity through robust database integration.
- Developed a user-friendly interface for seamless user interaction.
- Integration with mobile banking applications.
- Implementation of advanced security features like biometric authentication.
- Expansion of transaction types and services offered.



Thank you