

**ATM Simulation**

211000606 Karim Fawzy

Ahmed Salah 211001546

Jaser Kasim 211001801

Mohamed Tamer 211000379

CSCI217

Dr. Sameh Ahmed

Eng. Yossef Fathy

May, 2024

Table Of Contents

[1. Introduction 2](#_Toc166795159)

[2. Project Overview 2](#_Toc166795160)

[3. System Requirements 2](#_Toc166795161)

[4. System Design 2](#_Toc166795162)

[5. Implementation 3](#_Toc166795163)

[6. Testing and Error Handling 6](#_Toc166795164)

[7. User Guide 6](#_Toc166795165)

[8. Conclusion 6](#_Toc166795166)

[9. Future Work 6](#_Toc166795167)

[10. Appendices 6](#_Toc166795168)

# 1. Introduction

This report details the development of an ATM Simulation System using Java. The project encompasses two main sections: admin and user sections, with each section responsible for different functionalities within the system. The project aims to create an automated ATM system that ensures secure transactions and user-friendly interaction.

# 2. Project Overview

The ATM Simulation System is designed to facilitate both administrative and user operations in a secure and efficient manner. The system allows administrators to manage user accounts and user operations to perform transactions such as deposits, withdrawals, and balance inquiries. The project utilizes Java for developing the GUI application and incorporates encryption for securing card details.

**2.1 Admin Section**

* Adding and deleting user accounts
* Updating user records
* Managing overall system settings

**2.2 User Section**

* PIN validation
* Account operations: deposit, withdrawal, balance check
* Profile management

# 3. System Requirements

* Operating System: Windows/Linux/MacOS
* Java Development Kit (JDK) 8 or higher
* Integrated Development Environment (IDE) like Eclipse or IntelliJ IDEA

# 4. System Design

**4.1 Architecture**

The system follows a client-server architecture where the client is the user interface and the server handles the backend operations, including file management and transaction processing.

**4.2 File Structure**

The system utilizes text files for storing user and transaction data.

**4.3 User Interface Design**

The UI is designed to be intuitive and user-friendly, with separate panels for admin and user operations.

**4.3.1 Admin Interface**

* Login Panel
* Account Management Panel

**A screen shot of a cash machine

Description automatically generated4.3.2 User Interface**

* Login Panel
* Dashboard
* Transaction Panels (Deposit, Withdrawal, Balance Check)
* Profile Management Panel

# 5. Implementation

**5.1 Key Classes and Methods**

Detailed explanation of key classes and methods used in the project, including code snippets.

**5.1.1 Admin Operations**

* A screen shot of a machine

  Description automatically generatedAdminLogin.java
* AddUser.java
* Delete.java
* Update.java

A screenshot of a computer

Description automatically generatedA screen shot of a machine

Description automatically generated

**5.1.2 User Operations**

* A screen shot of a machine

  Description automatically generatedUser.java
* Deposit.java
* Withdraw.java
* FastCash.java
* CheckBalance.java

**5.1.3 Encryption and Security**

* EncryptionUtil.java

A screen shot of a machine

Description automatically generatedA screen shot of a cash machine

Description automatically generated

A screen shot of a machine

Description automatically generatedA screen shot of a machine

Description automatically generated

# 6. Testing and Error Handling

**6.1 Error Handling Mechanisms**

This is an example of how we do handle the all the possible errors that may face as:-

# 

# 7. User Guide

**7.1 Admin Section**

* How to login as an admin:

We made a user name for the admin as "admin" then a password "1234" so that he can be the only one to access the admin page.

* How to add/delete user accounts:

After entering to the admin page, we will see an options for the add, delete and update in the add we add all the information that needed then we create an account. In the delete page we inter the ID of the account that we want to delete.

* How to update user records:

Here we inter the ID of the account that we want to update then add all the information that we want to update.

**7.2 User Section**

* How to login as a user:

In the first page we inter the ID and PIN to access the account that we created before.

* How to perform deposits and withdrawals:

In this page we inter the ID to access the account that we created then do the functions for the deposit and withdrawal.

* How to check account balance:

In this page we inter the ID to access the account that we created then do the functions for checking the account.

* How to update profile details:

In this page we inter the ID to access the account that we created then do the functions for updating profile.

# 8. Conclusion

The ATM Simulation System project successfully demonstrates the creation of a robust and secure ATM interface using Java. By integrating both administrative and user functionalities, the system ensures a seamless experience for managing and operating bank accounts. Key features such as PIN validation, secure transactions, and user-friendly interfaces contribute to the reliability and usability of the system. Additionally, the use of encryption ensures that sensitive data is well-protected.

Throughout the project, we focused on designing an intuitive user interface and implementing critical backend operations to handle transactions and account management efficiently. The admin section allows for effective management of user accounts and system settings, while the user section facilitates essential banking operations like deposits, withdrawals, and balance inquiries.

Our testing and error handling mechanisms were carefully planned and executed to address potential issues, ensuring the system's robustness. The detailed user guide provides clear instructions for both admins and users, making the system easy to navigate and operate.

This project lays a solid foundation for future enhancements, such as integrating more advanced security features, expanding functionality, and potentially transitioning to a database-driven system for improved scalability. The knowledge and skills gained through this project will be invaluable for tackling more complex software development challenges in the future.