**Project Report: Banking System**

1. Title Page

Project Title: Banking System

Submitted By: Bavya Sarda, 23SCSE1420066

Ayush Raj Singh, 23SCSE1420007

Utkarsh Singh, 23SCSE1420105

Submitted To: Mr. Shashikant Sharma

Submission Date: 18 Jan’ 2025

Course: Data Base Management System

Section: DS-1

2. Abstract

This project aims to design and develop a Banking System using Python integrated with a Database Management System (DBMS). It provides functionalities for managing customers, accounts, and transactions. The project demonstrates the practical application of Python programming in the domain of financial systems, emphasizing user-friendliness and efficient data handling. Key features include customer registration, account management, transaction handling, and a user-friendly graphical interface.

3. Introduction

The Banking System project serves as a simulation of the operations of a real-world banking system. With increasing digitalization, such systems play a critical role in securely managing and processing financial data. Our project uses Python for coding the backend logic and a DBMS for data storage and management, providing a structured approach to implementing core banking functionalities.

4. Objectives

To implement a user-friendly banking system for basic operations.

To explore the integration of Python with DBMS for data handling.

To ensure data security and reliability in financial operations.

To demonstrate modular coding practices and efficient query handling.

5. Features of the System

1. Customer Management

Add, update, delete, or view customer records.

Store customer details like name, address, contact number, etc.

2. Account Management

Open new accounts and manage account types (savings, current, etc.).

View and update account details.

3. Transaction Management

Process deposits and withdrawals.

Generate transaction logs.

Maintain account balances.

4. Exit Functionality

Provide an option to terminate the application safely.

6. System Architecture

Frontend: User interface designed using Python libraries like Tkinter.

Backend: Business logic written in Python.

Database: [Mention the database used, e.g., SQLite, MySQL], containing tables for customers, accounts, and transactions.

7. Technology Stack

Programming Language: Python

Database: banking.db

Tkinter: For GUI design.

SQLite3/MySQL: For database integration.

Other Python modules

8. Methodology

1. Planning:

Define the scope of the project and core functionalities.

Identify the database schema requirements.

2. Development:

Implement a GUI using Tkinter.

Code backend functionalities like customer and transaction management.

Develop database tables and integrate with Python.

3. Testing:

Test each module for functionality and accuracy.

Validate database interactions.

4. Deployment:

Package the project for execution on different platforms.

9. Database Design

Tables:

Customers Table:

Columns: Customer ID, Name, Address, Contact Number, etc.

Accounts Table:

Columns: Account ID, Customer ID (Foreign Key), Account Type, Balance, etc.

Transactions Table:

Columns: Transaction ID, Account ID, Transaction Type (Deposit/Withdrawal), Amount, Date, etc.

10. Results and Discussion

The system successfully performs the following operations:

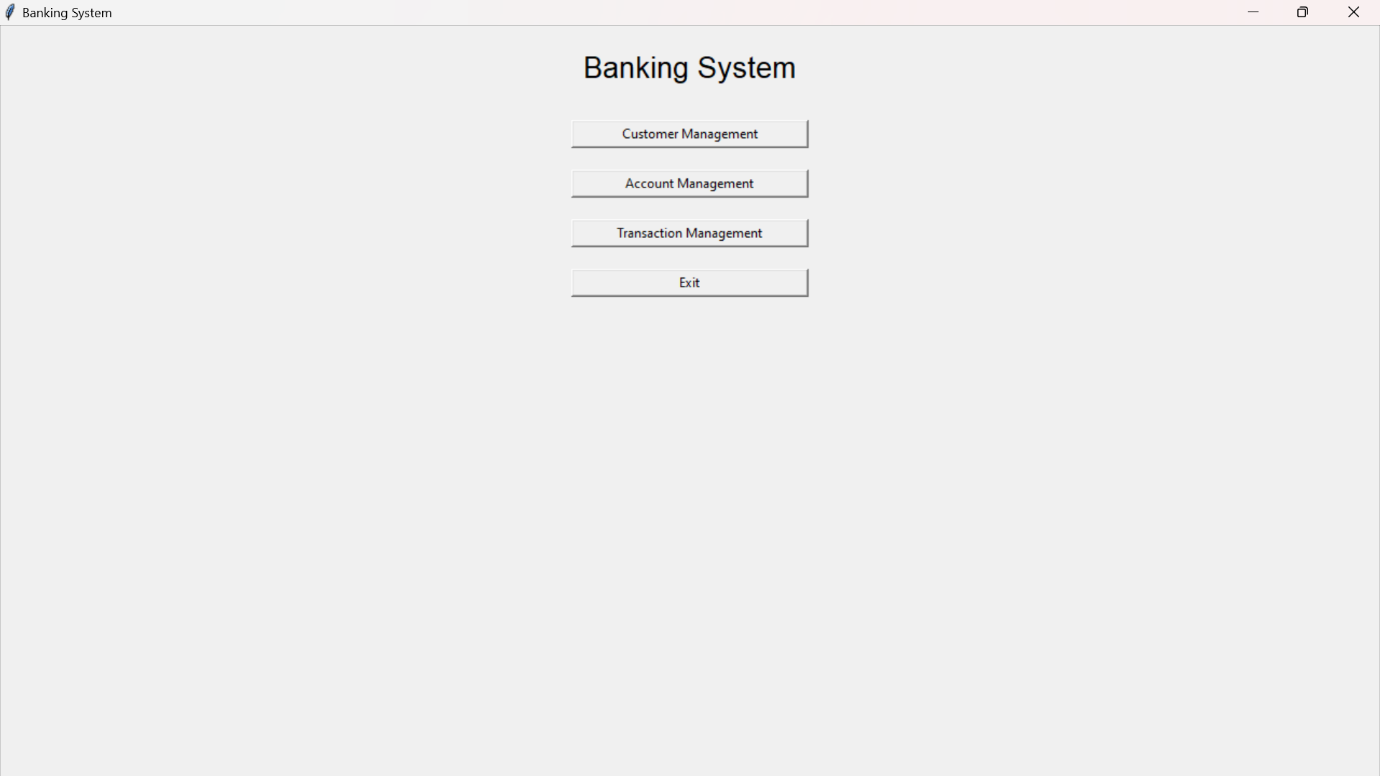
Adding and managing customer records.

Handling account creation and updates.

Processing transactions efficiently.

Discuss the system's accuracy and reliability.

Highlight any challenges faced during development and how they were resolved.



11. Limitations

The project is designed for learning purposes and may not include advanced security features.

Limited to basic banking functionalities without online transaction integration.

12. Future Scope

Adding advanced features like loan management and account analytics.

Implementing multi-user access with role-based permissions.

Enhancing the security of the database and application.

13. Conclusion

The Banking System project demonstrates how Python can be effectively integrated with a database to build robust applications. It provides an excellent foundation for understanding banking operations and implementing core functionalities using programming and database skills.

14. References

Python Documentation (https://docs.python.org)

Database Documentation ([Link to DBMS documentation])

Additional resources used for development.

15. Appendix  
  
  
