OOP II Project Report

PROJECT TITLE

Bank Management System

Prepared by

Students ID	**Students Name**
221-15-5688	Rakibul Hasan Akash

Daffodil International University

Date: 26-05-2024

Table of Contents

Banking Management System 2

1. Introduction 2

  1.1 Background 2

  1.2 Objective 2

  1.3 Motivation 2

  1.4 Related Works/Review 2

  1.5 Gap Analysis 3

2. System Architecture 3

  2.1 System Overview 3

  2.2 System Components 3

  2.3 Architecture Diagram 4

3. Project Features and Interface 5

  3.1 List of Features 5

  3.2 User Interfaces 5

  3.3 Input Output Demo 5

4. System Implementation 6

  4.1 Development Tools C Technologies 6

  4.2 Implementation Plan 6

  4.3 Testing and Validation 6

5. Future Scope and Limitation 6

  5.1 Limitation 6

  5.2 Future Scope 7

  5.3 Conclusion 7

6. References 7

Banking Management System

1. Introduction

1.1 Background

The financial industry has undergone significant changes due to technological advancements, leading to the emergence of digital banking. Traditional banking systems, which often require physical visits to branches, are being replaced by more efficient and user-friendly digital solutions. This project presents the development of a custom banking application that aims to provide a secure and intuitive platform for users to manage their finances.

1.2 Objective

The primary objective of this project is to develop a user-friendly banking application that offers essential functionalities such as registration, login, balance inquiry, deposits, and withdrawals. The application ensures data security and provides a seamless user experience.

1.3 Motivation

The motivation behind this project is to address the need for a simple yet secure banking solution. Many existing banking applications are either too complex for average users or lack adequate security measures. This project aims to bridge that gap by offering a straightforward and secure banking platform.

1.4 Related Works/Review

Several banking applications exist in the market, each with its unique features. However, many of these applications either compromise on user-friendliness or security. This project takes inspiration from existing solutions but focuses on enhancing user experience and security simultaneously.

1.5 Gap Analysis

Current banking applications often suffer from usability issues and security vulnerabilities. This project identifies these gaps and aims to develop an application that is both easy to use and highly secure, providing a better alternative for users.

2. System Architecture

2.1 System Overview

The system comprises a graphical user interface (GUI) built using Tkinter, a standard Python interface to the Tk GUI toolkit. It integrates with the Python Imaging Library (PIL) for handling image-related tasks. The application supports essential banking functions such as registration, login, and financial transactions (deposits and withdrawals).

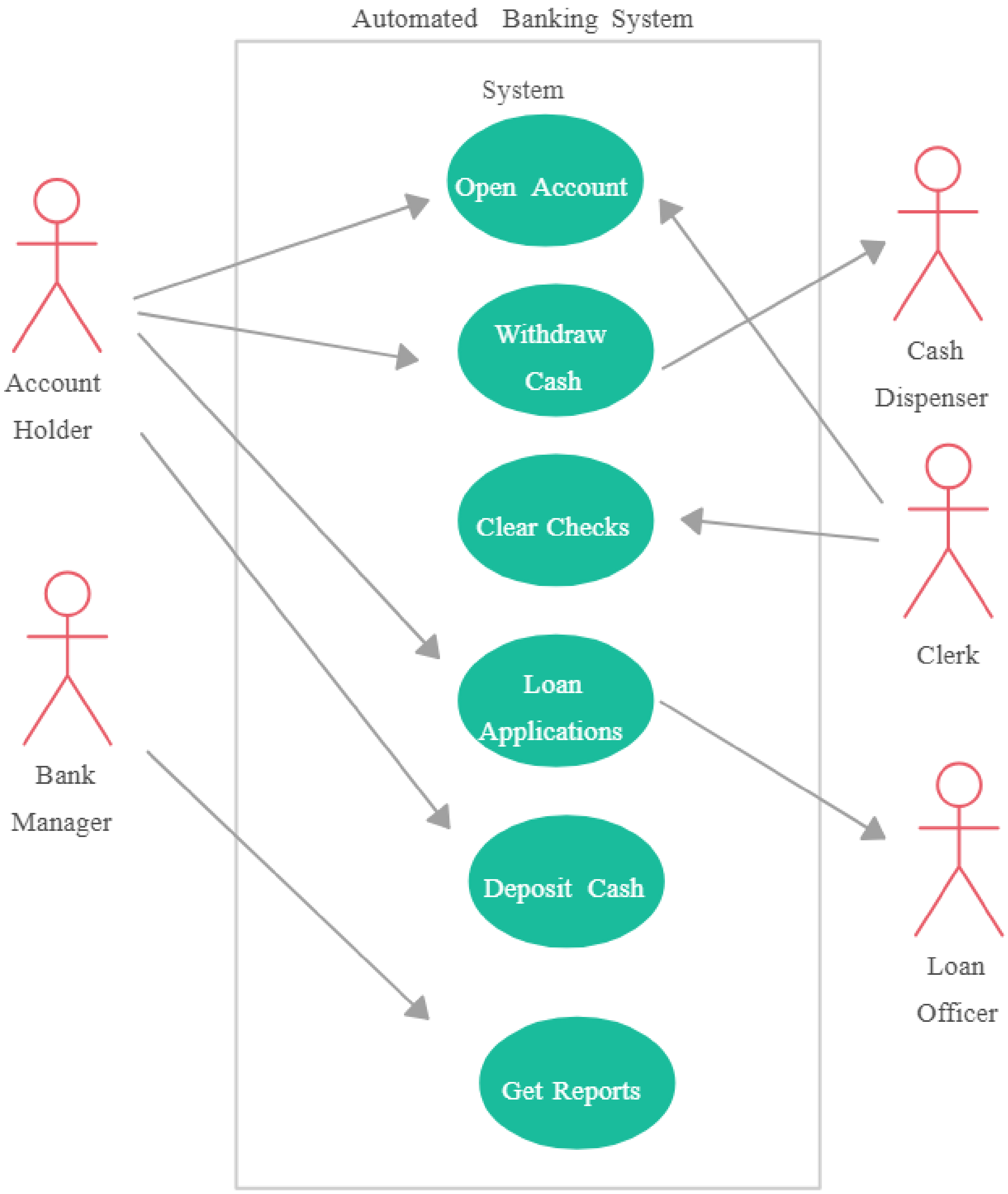
2.2 System Components

GUI: Built using Python OOP Concept, providing an intuitive interface for users.

File Handling: Utilizes Python's file handling capabilities to store and retrieve user data.

PIL: Used for image processing, enhancing the visual appeal of the application.

2.3 Architecture Diagram



3. Project Features and Interface

3.1 List of Features

User Registration: Allows new users to create an account.

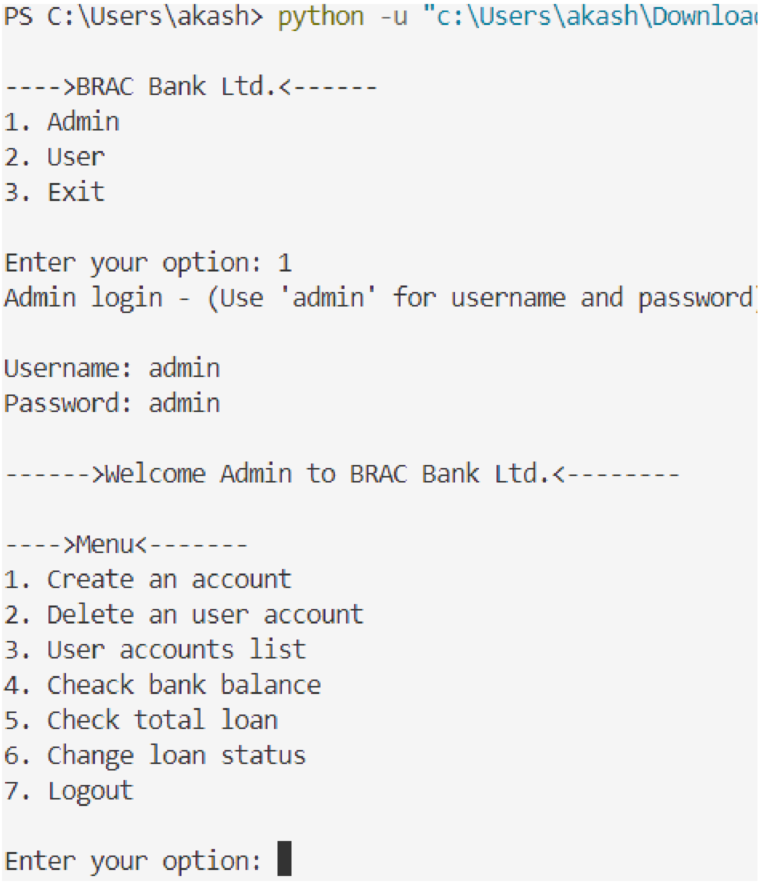
User Login: Enables existing users to access their accounts.

Deposit Funds: Allows users to deposit money into their accounts.

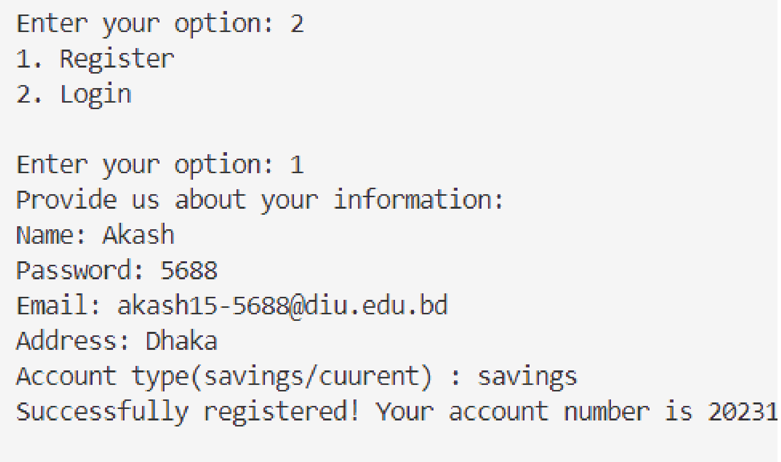
Withdraw Funds: Enables users to withdraw money from their accounts.

View Personal Details: Users can view their account details and balance.

3.2 User Interfaces



```
PS C:\Users\akash> python -u "c:\Users\akash\Downloa

---->BRAC Bank Ltd.<------
1. Admin
2. User
3. Exit

Enter your option: 1
Admin login - (Use 'admin' for username and password

Username: admin
Password: admin

------>Welcome Admin to BRAC Bank Ltd.<--------

---->Menu<-------
1. Create an account
2. Delete an user account
3. User accounts list
4. Cheack bank balance
5. Check total loan
6. Change loan status
7. Logout

Enter your option: 
```



```
Enter your option: 2
1. Register
2. Login

Enter your option: 1
Provide us about your information:
Name: Akash
Password: 5688
Email: akash15-5688@diu.edu.bd
Address: Dhaka
Account type(savings/cuurent) : savings
Successfully registered! Your account number is 20231
```

3.3 Input Output Demo

Registration: Users enter their name, age, gender, and password. The system creates a new account and stores the details in a file.

Login: Users enter their username and password. The system verifies the credentials and grants access to the account dashboard.

Deposit: Users enter the amount to be deposited. The system updates the account balance and confirms the transaction.

Withdraw: Users enter the amount to be withdrawn. The system checks the balance and processes the withdrawal if funds are sufficient.

4. System Implementation

4.1 Development Tools & Technologies

Python: The primary programming language used for developing the application.

Tkinter: Used for creating the graphical user interface.

PIL (Pillow): Used for image processing tasks.

4.2 Implementation Plan

The implementation was carried out in phases, starting with the design of the GUI using Tkinter, followed by the integration of file handling for data storage. Finally, PIL was incorporated to manage images and enhance the visual aspects of the application.

4.3 Testing and Validation

Extensive testing was conducted to ensure all functionalities work as expected. This included unit tests for individual components and integration tests to verify the overall system performance. User feedback was also gathered to identify and fix usability issues.

5. Future Scope and Limitation

5.1 Limitation

Scalability: The current file-based system may not be suitable for a large number of users.

Security: While basic security measures are in place, advanced security features such as encryption and two-factor authentication are not yet implemented.

5.2 Future Scope

Database Integration: Replacing the file-based system with a robust database solution for better scalability and security.

Enhanced Security: Implementing advanced security features to protect user data.

Mobile App: Developing a mobile version of the application to provide users with greater accessibility.

5.3 Conclusion

The custom banking application successfully demonstrates the implementation of a secure and user-friendly banking solution. While there are areas for improvement, particularly in scalability and security, the project lays a strong foundation for future enhancements.

6. References

- Tkinter Documentation

- Python Imaging Library (PIL) Documentation

- Various online resources and tutorials on Python file handling and GUI development