

ACADEMIC TASK

CSM216

(PYTHON PROJECT)

COMPUTER SCIENCE AND ENGINEERING

Submitted by:

Name: **VIVEK YADAV**

Registration number: **12306380**

Roll No.: **19**

Section: **K23UP**

Submitted to: Aman Kumar

PROJECT REPORT



LOVELY
PROFESSIONAL
UNIVERSITY

LOVELY PROFESSIONAL UNIVERSITY

DECLARATION

I am Vivek a student of Bachelor of Technology under CSE discipline at Lovely Professional University, Punjab, hereby declare that all the information furnished in this project report is based on my own work and is genuine

Vivek_Yadav

RegistrationNumber:12306380

Acknowledgement

I extend my heartfelt gratitude to everyone who contributed to the successful completion of this project.

First and foremost, I wish to thank my mentor, Mr. Aman Kumar, for his invaluable guidance, expertise, and support. His mentorship has been instrumental in shaping the development process and ensuring the project's objectives were met effectively. His constructive feedback and encouragement motivated me to overcome challenges and deliver a high-quality outcome.

I am deeply grateful to the faculty and staff of the School of Computer Science for creating a conducive learning environment and providing the resources necessary for this project. Their continuous encouragement and assistance were crucial to navigating the complexities involved in this endeavor.

I would also like to thank my classmates for their collaborative insights and suggestions, which brought fresh perspectives to the project. Lastly, I am immensely thankful to my family and friends for their unwavering belief in me and their constant support throughout the journey.

Table of Contents

1. Introduction	3
2. Objectives and Scope of the Project	4-6
2.1 Objectives	4-5
2.2 Scope of the Project	5-6
3. Application Tools	6-7
3.1 Python	6
3.2 Tkinter	6
3.3 SQLite Database	6
3.4 Visual Studio Code (VS Code)	7
3.5 Git and GitHub	7
4. Project Design	7-12
4.1 Core Components and Classes	7-9
4.2 Interaction Between Components	10
4.3 Achieving Project Objectives	10-11
5. Flowchart	11-12
6. Test Cases	13-15

1. Introduction

The **ATM Management System** is a Python-powered application simulating the functionalities of an Automated Teller Machine (ATM). This project demonstrates secure transactions, user account management, and financial operations. With Python's **Tkinter** for GUI and **SQLite** for database handling, the application is robust, user-friendly, and modular for future enhancements.

Key features include:

- Secure user authentication.
- Balance inquiries, withdrawals, and transfers.
- Error handling and data validation.

This system highlights efficient integration of programming principles to solve real-world banking challenges.

2. Objectives and Scope of the Project

2.1 Objectives

- 1. Develop a Secure and Intuitive Interface:**
 - Implement an easy-to-navigate GUI using **Tkinter**.
 - Provide clear feedback for user actions.
 - 2. Essential ATM Functionalities:**
 - Enable operations like account creation, balance inquiry, deposits, withdrawals, and transaction history.
 - Maintain accuracy in financial records.
 - 3. Robust Security and Validation:**
 - Implement PIN-based login.
 - Validate all inputs to prevent invalid transactions.
 - 4. Scalable Solution:**
 - Design modular code for future features like online banking, fraud detection, or biometric login.
-

2.2 Scope of the Project

This system functions as a standalone ATM application providing:

- Secure login and transaction management.
- Accurate tracking of user balances and operations.
- Modular design for potential expansions.

The project utilizes Python for backend logic, **Tkinter** for GUI, and **SQLite** for data storage. Its scalability allows for integration into more complex banking systems.

3. Application Tools

3.1 Python Programming Language

Chosen for its simplicity and strong library ecosystem, Python supports GUI development and seamless database integration.

3.2 Tkinter

Provides a user-friendly graphical interface to handle user interactions.

3.3 SQLite Database

Lightweight database for securely storing user account details and transaction logs.

3.4 Visual Studio Code (VS Code)

An efficient IDE for developing, debugging, and maintaining the project.

3.5 Git and GitHub

Used for version control, enabling easy tracking of changes and collaborative work.

4. Project Design

4.1 Core Components and Classes

1. **Main Application Class:**
 - Manages transitions between login, withdrawal, deposit, and transfer screens.
 2. **User Class:**
 - Represents user accounts with attributes like account number, PIN, and balance.
 3. **Transaction Class:**
 - Handles deposits, withdrawals, and transfers while validating inputs.
 4. **Database Class:**
 - Manages data storage and retrieval using SQLite.
 5. **Validation Class:**
 - Ensures user inputs (e.g., PINs, amounts) are correct and secure.
-

4.2 Interaction Between Components

1. **GUI Interaction:**

Users perform actions via a Tkinter-based GUI with buttons and input fields.
 2. **Database Management:**

Data is stored and retrieved securely using the Database class.
 3. **Transaction Handling:**

Validates inputs, updates balances, and logs transactions.
 4. **Error Handling:**

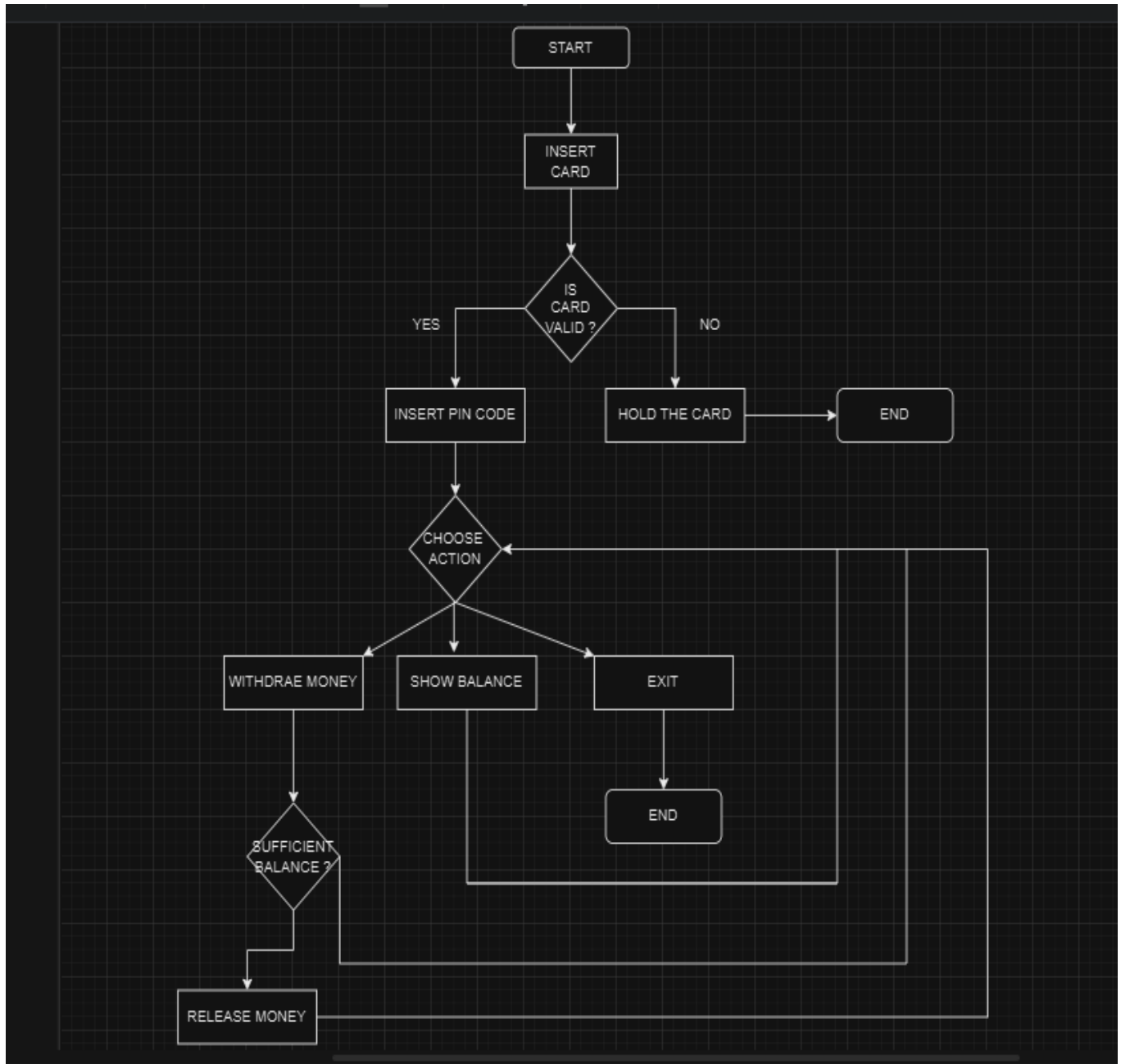
Prevents crashes by validating inputs and managing exceptions.
-

4.3 Achieving Project Objectives

- **Efficiency:** Modular architecture ensures smooth functionality.
 - **Security:** PIN protection and robust validation guard against unauthorized access.
 - **Scalability:** Design supports future enhancements like mobile banking or fraud detection.
-

5. Flowchart

The flowchart represents the sequence of user actions, validations, and responses, including login, balance inquiry, withdrawals, and transfers.



6. Test Cases

6.1 Login Functionality

Test Case ID	Description	Input	Expected Output	Status
TC-001	Valid login credentials	Card: 9534757617, PIN: 1290	Successful login	Pass
TC-002	Invalid card number	Card: 1111111111, PIN: 1234	Error: Invalid card or PIN	Pass
TC-003	Invalid PIN	Card: 9534757617, PIN: 0000	Error: Invalid card or PIN	Pass
TC-004	Non-numeric inputs	Card: abcd, PIN: ****	Error: Enter valid numbers	Pass

6.2 Balance Inquiry

Test Case ID	Description	Input	Expected Output	Status
TC-005	Check valid balance	User: 9534757617	Display correct balance	Pass
TC-006	Attempt without login	None	Redirect to login	Pass

6.3 Withdrawal Functionality

Test Case ID	Description	Input	Expected Output	Status
TC-007	Withdraw within balance limit	Amount: ₹500	Success message, updated balance	Pass
TC-008	Withdraw exceeding balance	Amount: ₹3000	Error: Insufficient funds	Pass
TC-009	Invalid withdrawal input	Amount: abc	Error: Enter valid amount	Pass

6.4 Transfer Functionality

Test Case ID	Description	Input	Expected Output	Status
TC-010	Valid transfer	To: 9306011724, Amount: ₹500	Success: Updated balances	Pass
TC-011	Transfer exceeding balance	To: 9306011724, Amount: ₹3000	Error: Insufficient balance	Pass
TC-012	Invalid account number	To: 9999999999, Amount: ₹100	Error: Account not found	Pass
TC-013	Non-numeric amount input	To: 9306011724, Amount: abc	Error: Enter valid details	Pass

6.5 Database Operations

Test Case ID	Description	Operation	Expected Output	Status
TC-014	Log balance updates	Withdrawal	Balance correctly reflected in database	Pass
TC-015	Record transfer transactions	Transfer ₹500	Transaction logged in database	Pass

Result:

The ATM system developed with Python and Tkinter successfully implements core banking functions such as login, balance inquiries, withdrawals, and money transfers. The system uses a straightforward design, relying on dictionaries to manage user data, balances, and PINs, which supports efficient transaction processing.

While the current implementation serves as a basic prototype, there are key areas for refinement before it can be considered suitable for real-world deployment. Enhancing security—such as encrypting PINs and adopting stronger authentication methods—would be critical. Additionally, introducing features like session timeouts, transaction history, and multi-account support would enhance the system's functionality and security.