Project Report: Python-Based Mobile Banking System

1. Introduction This project aims to develop a Python-based mobile banking system similar to Nagad or Bkash. The system allows users to create accounts, deposit and withdraw money, check balances, and view account details securely using a PIN-based authentication mechanism.

2. Objectives

- Implement a simple and secure mobile banking system using Python.
- Enable users to create accounts with unique account numbers.
- Facilitate deposit and withdrawal transactions with PIN authentication.
- Provide balance inquiry and account details retrieval features.
- Simulate a USSD-based menu system for user interaction.

3. Tools and Technologies Used

- **Programming Language:** Python
- **Libraries Used:** None (Standard Python functionalities used)
- **Security Measures:** PIN-based authentication

4. System Design The system consists of the following components:

- **User Authentication Module:** Ensures secure login using a PIN.
- **Account Management Module:** Enables account creation and storage of details.
- **Transaction Module:** Allows deposits, withdrawals, and balance checks.
- **USSD Menu System:** Provides an interactive text-based menu for user operations.

5. Implementation

- **Account Creation:**
    - Users provide a unique account number, name, PIN, and an initial deposit.
    - Duplicate account numbers are not allowed.

- **Deposit & Withdrawal:**
  - Users can deposit any amount.
  - Withdrawals require sufficient balance and PIN authentication.
- **Balance Inquiry & Account Details:**
  - Users can check their balance and retrieve account details after PIN validation.
- **USSD Menu System:**
  - Users access services by entering a predefined code (*167#).
  - The menu guides users through available banking operations.

6. Results and Discussion

- The system successfully registers users and securely stores their account details.
- Transactions, including deposits and withdrawals, function correctly with PIN verification.
- The USSD-based menu enhances user interaction and ease of access.
- The balance inquiry and account information retrieval modules work efficiently.

7. Future Enhancements

- Implementing a database (SQLite or MySQL) for persistent data storage.
- Enhancing security with encryption for PIN storage.
- Developing a GUI-based interface for better usability.
- Adding mobile payment integration with external services.

8. Conclusion This Python-based mobile banking system effectively simulates fundamental banking operations using a command-line interface. The use of PIN authentication ensures security, while the interactive menu simplifies user interaction. Future improvements can enhance functionality and user experience.