# Bank Account Management System - Detailed Documentation

## Introduction

The Bank Account Management System is a console-based Python application designed to allow users   
to create and manage bank accounts. The system provides functionalities for depositing,   
withdrawing money, checking balances, and viewing transaction history, with data persistence using JSON files.

## Key Features

- “Create multiple bank accounts” with unique account numbers.  
- “Deposit money”.  
- “Withdraw money” while ensuring sufficient balance.  
- “Check account balance” at any time.  
- “View past transactions” for tracking deposits and withdrawals.  
- “Data persistence” using JSON files for each account.  
- “Error handling” for invalid inputs and operations.

## How It Works

1. “Account Creation”: Users create an account by entering a unique 10-digit account number and holder name.  
2. “Deposits”: Users can deposit any valid amount into their account.  
3. “Withdrawals”: Users can withdraw money if their balance is sufficient.  
4. “Balance Inquiry”: Users can check their account balance anytime.  
5. “Transaction History”: The system stores all transactions (deposits and withdrawals) and allows users to review them.  
6. “Data Storage”: Each account’s data (account number, holder name, balance, and transactions) is stored in a JSON file inside the "accounts" folder.

## Implementation Details

The system is implemented using Object-Oriented Programming (OOP), ensuring modular and maintainable code.

## Class and Methods

### `BankAccount` Class

- “`\_\_init\_\_(self, accno, holder\_name)`”: Initializes a new bank account.  
- “`money\_deposit(self, amount)`”: Deposits money and records the transaction.  
- “`money\_withdraw(self, amount)`”: Withdraws money if the balance is sufficient.  
- “`check\_balance(self)`”: Displays the current account balance.  
- “`history(self)`”: Displays transaction history.  
- “`save\_detail(self)`”: Saves account details and transactions in a JSON file.  
- “`load\_account(self)`”: Loads previous transactions from the JSON file if available.

## User Interaction

The system provides a simple, menu-driven interface where users can select options by entering   
the corresponding number.

### Menu Options:

1. “Create New Account”: Enter account details to create a new account.  
2. “Access Existing Account”: Provides options for account management:  
 - Deposit Money  
 - Withdraw Money  
 - Check Balance  
 - View Transaction History  
 - Create New Account  
 - Exit

## Error Handling

- Prevents deposits of negative or zero amounts.  
- Ensures withdrawals do not exceed available balance.  
- Validates user inputs for numerical values.  
- Prevents creation of duplicate accounts.

## Conclusion

The Bank Account Management System provides a structured and efficient way to handle banking operations in a console-based application. By leveraging OOP principles and file handling, the system ensures data persistence and ease of use.