PROJECTS/BANK MANAGEMENT SYSTEM

```
class BankAccount:
    def __init__(self, account_number, account_holder, password, balance=0):
        self.account_number = account_number
        self.account_holder = account_holder
        self.password = password  # Added password for login
        self.balance = balance
        self.transactions = []  # List to store transactions

    def deposit(self, amount):
        if amount > 0:
            self.balance += amount
            self.transactions.append(f"Deposited ${amount}")
            print(f"Deposited ${amount}. New balance is ${self.balance}.")
        else:
            print("Invalid amount. Deposit must be greater than zero.")

    def withdraw(self, amount):
        if amount > 0 and amount <= self.balance:
            self.balance -= amount
            self.transactions.append(f"Withdrew ${amount}")
            print(f"Withdrew ${amount}. New balance is ${self.balance}.")
        elif amount > self.balance:
            print("Insufficient funds.")
        else:
            print("Invalid amount. Withdrawal must be greater than zero.")

    def transfer(self, other_account, amount):
        if amount > 0 and amount <= self.balance:
            self.balance -= amount
            other_account.balance += amount
            self.transactions.append(f"Transferred ${amount} to Account {other_account.account_number}")
            other_account.transactions.append(f"Received ${amount} from Account {self.account_number}")
            print(f"Transferred ${amount} to Account {other_account.account_number}.")
            print(f"New balance is ${self.balance}.")
        elif amount > self.balance:
            print("Insufficient funds.")
        else:
            print("Invalid amount. Transfer must be greater than zero.")

    def get_balance(self):
        return self.balance

    def display_mini_statement(self):
        print("\nMini Statement:")
        if not self.transactions:
            print("No transactions yet.")
        else:
            # Show last 5 transactions (or fewer if less than 5)
            for transaction in self.transactions[-5:]:
```

```
                print(transaction)

    def __str__(self):
        return f"Account Number: {self.account_number}, Account Holder: {self.account_holder}, Balance: ${self.balance}"


class BankSystem:
    def __init__(self):
        self.accounts = {}
        self.logged_in_account = None  # To keep track of the currently logged-in account

    def create_account(self, account_number, account_holder, password):
        if account_number in self.accounts:
            print("Account with this number already exists.")
        else:
            new_account = BankAccount(account_number, account_holder, password)
            self.accounts[account_number] = new_account
            print(f"Account for {account_holder} created successfully.")

    def get_account(self, account_number):
        return self.accounts.get(account_number, None)

    def login(self, account_number, password):
        account = self.get_account(account_number)
        if account and account.password == password:
            self.logged_in_account = account
            print(f"Login successful. Welcome {account.account_holder}.")
            return True
        else:
            print("Invalid account number or password.")
            return False

    def logout(self):
        self.logged_in_account = None
        print("You have been logged out.")

    def is_logged_in(self):
        return self.logged_in_account is not None

    def display_accounts(self):
        if self.accounts:
            print("All Accounts:")
            for account in self.accounts.values():
                print(account)
        else:
            print("No accounts found.")


def main():
    bank = BankSystem()

    while True:
```

```
        print("\nBank Management System")
        if bank.is_logged_in():
            print("1. Deposit")
            print("2. Withdraw")
            print("3. Check Balance")
            print("4. Display Mini Statement")
            print("5. Logout")
            choice = input("Enter your choice: ")

            if choice == "1":
                amount = float(input("Enter amount to deposit: "))
                bank.logged_in_account.deposit(amount)

            elif choice == "2":
                amount = float(input("Enter amount to withdraw: "))
                bank.logged_in_account.withdraw(amount)

            elif choice == "3":
                print(f"Balance: ${bank.logged_in_account.get_balance()}")

            elif choice == "4":
                bank.logged_in_account.display_mini_statement()

            elif choice == "5":
                bank.logout()

            else:
                print("Invalid choice. Please try again.")

        else:
            print("1. Create Account")
            print("2. Login")
            print("3. Exit")
            choice = input("Enter your choice: ")

            if choice == "1":
                account_number = input("Enter Account Number: ")
                account_holder = input("Enter Account Holder Name: ")
                password = input("Enter Password: ")
                bank.create_account(account_number, account_holder, password)

            elif choice == "2":
                account_number = input("Enter Account Number: ")
                password = input("Enter Password: ")
                if bank.login(account_number, password):
                    continue  # After successful login, return to the user menu.

            elif choice == "3":
                print("Exiting Bank System.")
                break

            else:
                print("Invalid choice. Please try again.")

```

```
156
157 if __name__ == "__main__":
158     main()
159
```