```
import java.io.*;
import java.util.*;

class BankAccount implements Serializable {
   private String accountNumber;
   private String accountHolderName;
   private double balance;

   public BankAccount(String accountNumber, String accountHolderName, double balance) {
      this.accountNumber = accountNumber;
      this.accountHolderName = accountHolderName;
      this.balance = balance;
   }

   public String getAccountNumber() {
      return accountNumber;
   }

   public String getAccountHolderName() {
      return accountHolderName;
   }

   public double getBalance() {
      return balance;
   }

   public void deposit(double amount) {
      balance += amount;
   }

   public void withdraw(double amount) {
        if (balance >= amount) {
            balance -= amount;
        } else {
            System.out.println("Insufficient balance.");
        }
    }

    @Override
    public String toString() {
        return "Account Number: " + accountNumber + "\nAccount Holder Name: " +
accountHolderName
                + "\nBalance: $" + balance;
    }
}

class BankManager {
    private ArrayList<BankAccount> bankAccounts;
    private String fileName;

    public BankManager(String fileName) {
        this.fileName = fileName;
        bankAccounts = new ArrayList<>();
        loadAccountsFromFile();
    }

    public void createAccount(String accountNumber, String accountHolderName, double balance) {
        BankAccount bankAccount = new BankAccount(accountNumber, accountHolderName, balance);
        bankAccounts.add(bankAccount);
        saveAccountsToFile();
        System.out.println("Account created successfully.");
    }
```

```java
public BankAccount getAccount(String accountNumber) {
    for (BankAccount account : bankAccounts) {
        if (account.getAccountNumber().equals(accountNumber)) {
            return account;
        }
    }
    return null;
}

public void deposit(String accountNumber, double amount) {
    BankAccount account = getAccount(accountNumber);
    if (account != null) {
        account.deposit(amount);
        saveAccountsToFile();
        System.out.println("Amount deposited successfully.");
    } else {
        System.out.println("Account not found.");
    }
}

public void withdraw(String accountNumber, double amount) {
    BankAccount account = getAccount(accountNumber);
    if (account != null) {
        account.withdraw(amount);
        saveAccountsToFile();
        System.out.println("Amount withdrawn successfully.");
    } else {
        System.out.println("Account not found.");
    }
}
```

```
public void displayAccountDetails(String accountNumber) {
    BankAccount account = getAccount(accountNumber);
    if (account != null) {
        System.out.println(account.toString());
    } else {
        System.out.println("Account not found.");
    }
}

private void saveAccountsToFile() {
    try {
        FileOutputStream fos = new FileOutputStream(fileName);
        ObjectOutputStream oos = new ObjectOutputStream(fos);
        oos.writeObject(bankAccounts);
        oos.close();
        fos.close();
    } catch (IOException e) {
        System.out.println("Failed to save accounts to file: " + e.getMessage());
    }
}

private void loadAccountsFromFile() {
    try {
        FileInputStream fis = new FileInputStream(fileName);
        ObjectInputStream ois = new ObjectInputStream(fis);
        bankAccounts = (ArrayList<BankAccount>) ois.readObject();
        ois.close();
        fis.close();
    } catch (IOException | ClassNotFoundException e) {
        // Ignore if file does not exist or is empty
        }
    }
}

public class BankingManagementSystem {
    public static void main(String[] args) {
        BankManager bankManager = new BankManager("accounts.dat"); // Initialize BankManager with the filename for data persistence

        Scanner scanner = new Scanner(System.in);

        while (true) {
            System.out.println("Banking Management System");
            System.out.println("1. Create Account");
            System.out.println("2. Deposit");
            System.out.println("3. Withdraw");
            System.out.println("4. Display Account Details");
            System.out.println("5. Exit");
            System.out.print("Enter your choice: ");
            int choice = scanner.nextInt();
            scanner.nextLine(); // Consume newline character

            switch (choice) {
                case 1:
                    System.out.print("Enter Account Number: ");
                    String accountNumber = scanner.nextLine();
                    System.out.print("Enter Account Holder Name: ");
                    String accountHolderName = scanner.nextLine();
                    System.out.print("Enter Initial Balance: ");
                    double balance = scanner.nextDouble();
                    bankManager.createAccount(accountNumber, accountHolderName, balance);
                    break;
                case 2:
                    System.out.print("Enter Account Number: ");
                    accountNumber = scanner.nextLine();
                    System.out.print("Enter Deposit Amount: ");
                    double depositAmount = scanner.nextDouble();
                    bankManager.deposit(accountNumber, depositAmount);
                    break;
                case 3:
                    System.out.print("Enter Account Number: ");
                    accountNumber = scanner.nextLine();
                    System.out.print("Enter Withdraw Amount: ");
                    double withdrawAmount = scanner.nextDouble();
                    bankManager.withdraw(accountNumber, withdrawAmount);
                    break;
                case 4:
                    System.out.print("Enter Account Number: ");
                    accountNumber = scanner.nextLine();
                    bankManager.displayAccountDetails(accountNumber);
                    break;
                case 5:
                    System.out.println("Thank you for using Banking Management System!");
                    System.exit(0);
                    break;
                default:
                    System.out.println("Invalid choice. Please try again.");
            }
        }
    }
}
```