Bank Application

Scenario: You are a back-end developer and need to create an application to handle new customer bank account requests.

The application should do the following:

1. Read a .csv file of names, social security numbers, account type, and initial deposit
2. Use a proper data structure to hold all these accounts
3. Both savings and checking accounts share the following properties:

deposit()

withdraw()

transfer()

showInfo()

11-digit Account Number (generated with the following process:1 or 2 depending on Savings or Checking, last two digits of SSN, unique 5-digit number, and random 3-digit number)

1. Savings Account holders are given a Safety Deposit Box, identified by 3-digit number and accessed with a 4-digit code.
2. Checking Account holders are assigned a Debit Card with a 12-digit number and 4-digit PIN
3. Both account will use an interface that determines the base interest rate.

Savings accounts will use .25 points less than the base rate

Checking accounts will use 13% of the base rate

1. The showInfo method should reveal relevant account information as well as information specific to the Checking account or Savings account.

What I learned from this project:

1. Learn to develop a robust application architecture
2. Use when to use abstract classes and abstract methods
3. Use an interface API to receive information from a developer’s application
4. Explore constructors deeper and use the super() keyword
5. Explore access modifiers and when to use public, private, or protected
6. Read data from a file and store in an appropriate data structure
7. Generate random numbers and work with String API

Outline:

The superclass will be called first, then the derived class