Java Challenge

The purpose of this challenge is to give you a chance to use Java programming constructs, OO constructs and the Java API to create a small command line Java application. You will do this exercise in teams and each team is expected to demonstrate their application running after the challenge. Try to make the application robust by handling exceptions and including validation. When handling exceptions include useful messages for the user.

Another purpose is to have fun solving problems and exploring the Java API. Do not feel that you must finish it all, but it would be nice to see at least a good attempt.

1. Write a Java program to create a class known as "BankAccount" with an account name and balance and methods called deposit() and withdraw(). The account should have a constructor to initialise the name and balance and getters and setters as appropriate. Create a subclass called SavingsAccount that overrides the withdraw() method to prevent withdrawals if the account balance falls below one hundred. You will also need to provide a constructor for the savings account as this is not inherited.
2. Use your classes in a main method that asks the user to choose the type of account and then prompts for the account name and balance before creating the account object. You should also display the account details after they have been entered, including the account type.
3. Modify you program so that it displays a simple text-based menu that looks like this:

   Choose an option: (a)Add Account (q)Quit

4. The program should keep running until the user enters "q" to Quit.
5. Modify the program so that it stores the accounts in a list and only displays the accounts when the user chooses to do so. You will need to modify the menu:

   Choose an option: (a)Add Account (l)Display Accounts (q)Quit

6. Modify the program so that it can store all the accounts in your list in a MongoDB collection, your prompt will also need to change as shown here.

   Choose an option: (a)Add Account (l)Display Accounts (s)Save to database (q)Quit

7. Modify the program so that it loads the accounts from the database into the list when the program starts and saves them to the database when the program ends
8. Modify the program so that it enables the user to choose an account from the collection and then deposit or withdraw funds. A menu could look like this:

   Choose an option: (a)Add Account (l)Display Accounts (s)Save to database
   (d) Deposit funds (w)Withdraw funds (q)Quit