Programing Basics and OOP with Java

Practical Task

Write a program that simulates a banking application. At startup, the user must enter data about his user account (**myAccount**) and bank account (**bankAccount**). The IBAN account number must be generated automatically and when created, the bank account must be without available funds.

To add funds to the bank account, the user must perform a **top-up,** which should be a method in the bank account class, and enter the desired amount to deposit. The method should print the amount added to the account and the total amount in the account.

The user must select how many contacts (**contact**) he/she wants to add to the contact list and enter the data for each of them (the bankName should be random and the balance should be 0) . After entering the last contact, display a message that the contact list is complete.

When the user is done with the actions described above, the user should be presented with a menu with the following options for further actions:

1. Pay to contact
2. Refund from contact
3. View Transaction History
4. Exit the application

Each of those actions is described below:

(1) and (2): The application's main function is to perform money transactions between the user account and a selected contact from its list. The selection of contact is made by phone number.

The application must support the following types of transactions:

- **Pay** - transfers funds from the user's balance to the balance of the selected contact, after checking for sufficient availability.
- **Refund** - transfers funds from the balance of the selected contact to the balance of the user, after checking for sufficient availability.

To complete a transaction, the user must enter the following data:
transaction type (**pay** or **refund)**, *contact phone, amount*

After each transaction, the data for the transaction should be stored and the user must be presented with the menu again.

(3): View the transaction history action should present the user with all transactions that are made until the moment and give the opportunity to go back to the main menu.

(4) When the user exits the application, save the whole transaction history in a text file: *transaction type* (**pay** or **refund)**, *amount, and contact phone.*

After saving the file, display a message that the file was successfully saved and its name. The file name should consist of a user name and a timestamp in the following format: *{myAccount.lastName_yyyy-MM-dd-HH:mm:ss}.txt*.

Hint: You can use the menu class to have methods handling the print of all menu items including ones that are for the creation of a contact list, creation of the bank account, and the top-up.

Description of classes:

- **MyAccount:**
    - firstName
    - lastName
    - phone
    - bankAccount()
- **BankAccount:**
    - bankName
    - IBAN
    - balance
    - transactionHistory()
- **Contact**:
    - firstName
    - lastName
    - phone
    - bankAccount()

You can find an example of such functionality here and also you can find an example of a **console application menu here**. Do a validation of all fields for correct values.