Simple Banking System with Exception Handling

Create a Java program that simulates a simple banking system, allowing users to deposit, withdraw, and check their account balance while handling various exceptions.

Requirements:

1. Create a BankAccount class with the following attributes:
   1. accountNumber (a unique integer)
   2. balance (double) representing the account balance
   3. Implement methods in the BankAccount class for deposit, withdrawal, and checking the balance while handling exceptions such as insufficient funds, invalid input, and custom exceptions.
2. Create a custom exception called InsufficientFundsException that is thrown when a withdrawal is attempted with insufficient funds.
3. Develop a BankingApp class to interact with users.
   1. Allow users to deposit, withdraw, and check balance, and handle exceptions by providing helpful error messages.
   2. Implement a menu-driven user interface for the banking app, allowing users to perform operations repeatedly until they choose to exit.
   3. Use try-catch blocks to handle exceptions and provide detailed error messages.
4. Implement input validation to ensure that user input is in the correct format (eg, no negative deposits or withdrawals, valid account numbers).
5. Add support for multiple accounts (eg, using a ArrayList of BankAccount objects) and allow users to select an account by entering an account number.
6. Implement a feature to save account data to a file (eg, using serialization or text files) so that account information is retained between program runs.

During the program:

* Use try-catch blocks, custom exceptions, and good error messages to make the program user-friendly and robust.
* Make use of encapsulation, creating meaningful custom exceptions, and modular code design.
* Ensure you test your program with various inputs to verify the correctness of exception handling.