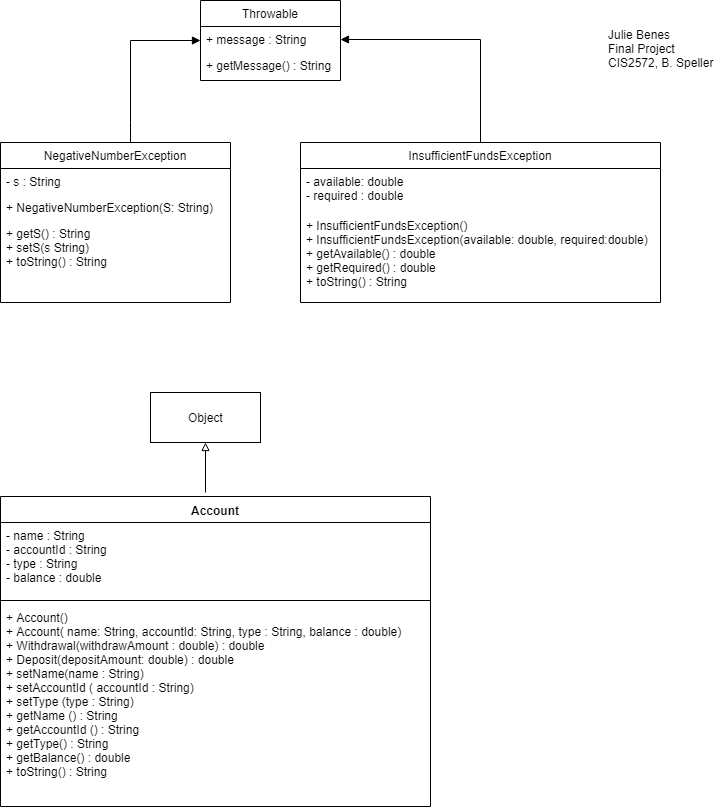
UML:



**Design:**

GUI design approach: Use labels, textFields, buttons, radio buttons and textArea.

My goal was to compartmentalize functionality.

* The upper right corner are all the fields related to Creating a new user.
* Bottom right corner has 3 menu buttons to Save and Restore data, and display summary report.
* Upper left corner is the search area with text fields below to display account data. The only field that is editable in this area is the search textfield.
* Transactions (withdrawal and deposit) are on the left side bottom. Button plus text field for input.
* A communication textArea is centered at the bottom.

The design and ending product were pretty much as I planned. Using FXML is really helpful to see options that are available and try them out.

Use string class for all text fields except those with monetary values. Use double for those.

Use radio button for choice of savings and checking which should simplify validation.

Use 4 digits for account id ranging from 5000-9999. Data is not sequential; i.e. Lot of gaps in between.

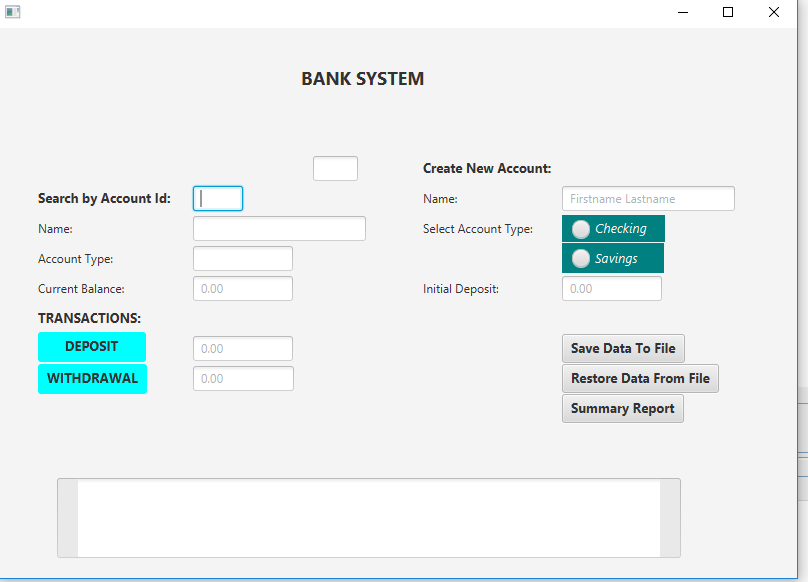
Decided to search only by Account Id to lessen typing errors with full names.

Input data: 100 accounts with random balances, and a mix of check vs savings.

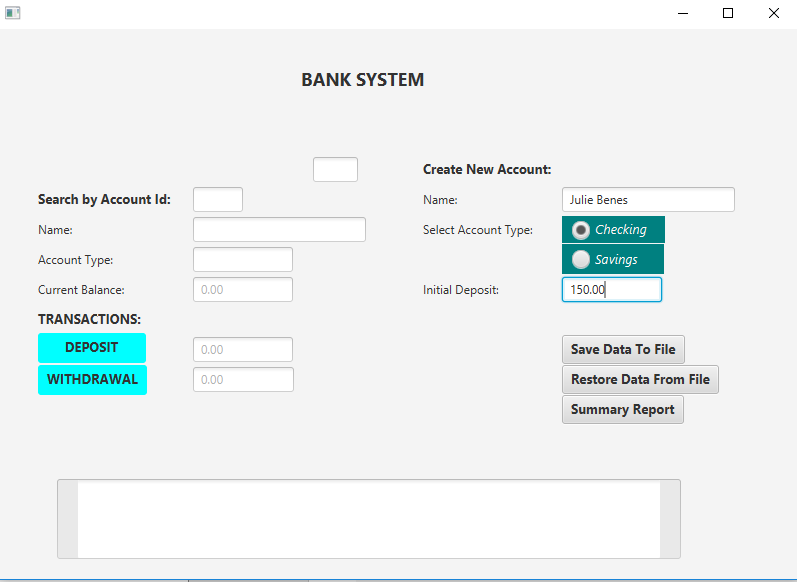
Created account class with accountId, Name (first & last), Type (checking or savings) , Balance and Withdrawal and Deposit methods.

Created NegativeNumberException class and InsufficientFundsException Classes.

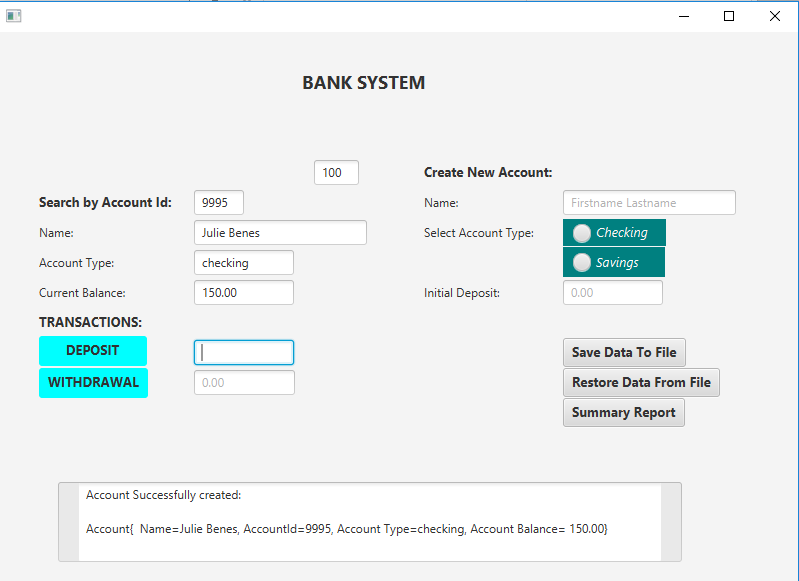
Opening screen:



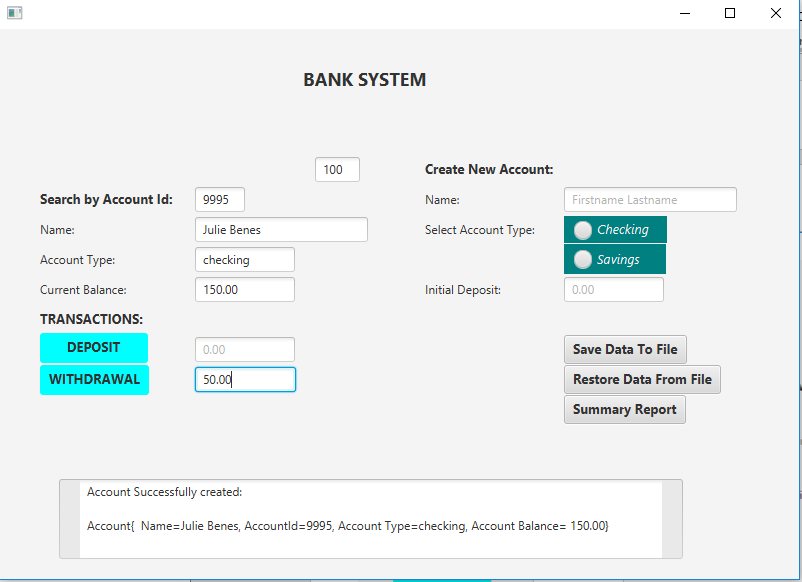
After Adding Account Info, before <enter>:



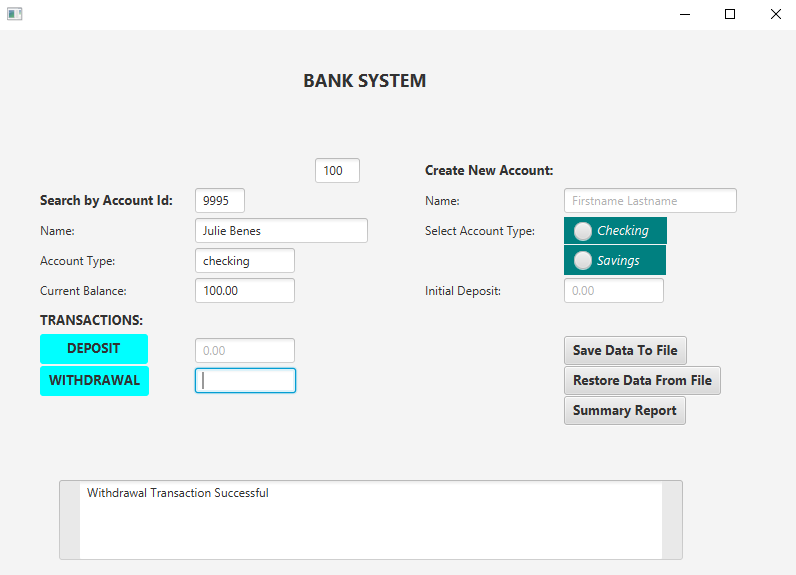
After adding account:



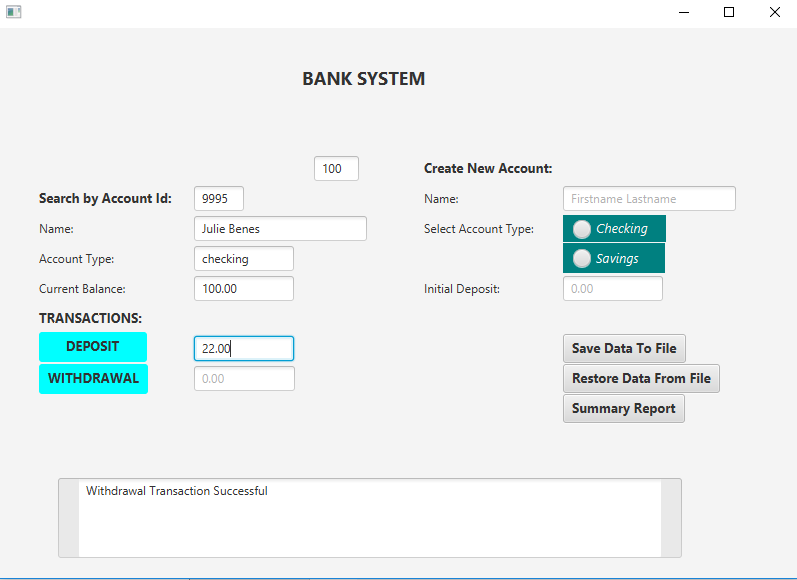
Before submitting withdrawal request:



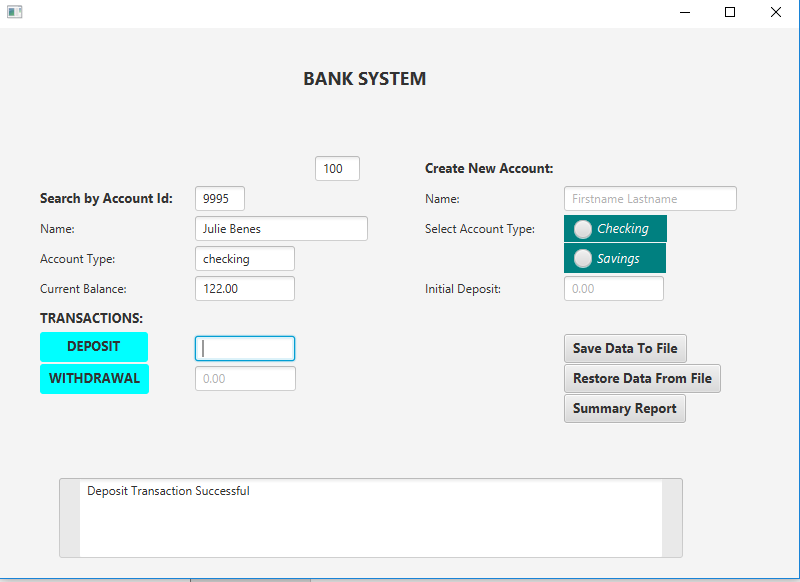
After submitting withdrawal:



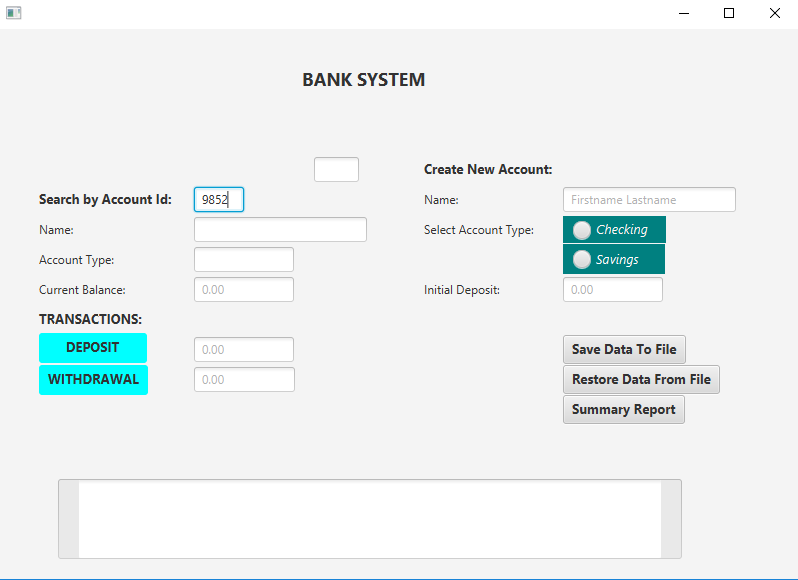
Before submitting Deposit:



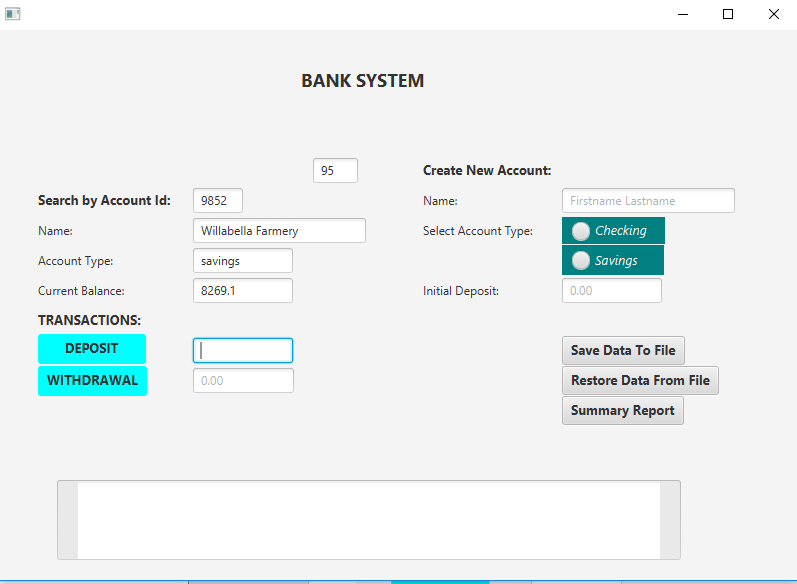
After submitting deposit request:



Before submitting search query:



After submitting query request:



Summary Report:

