

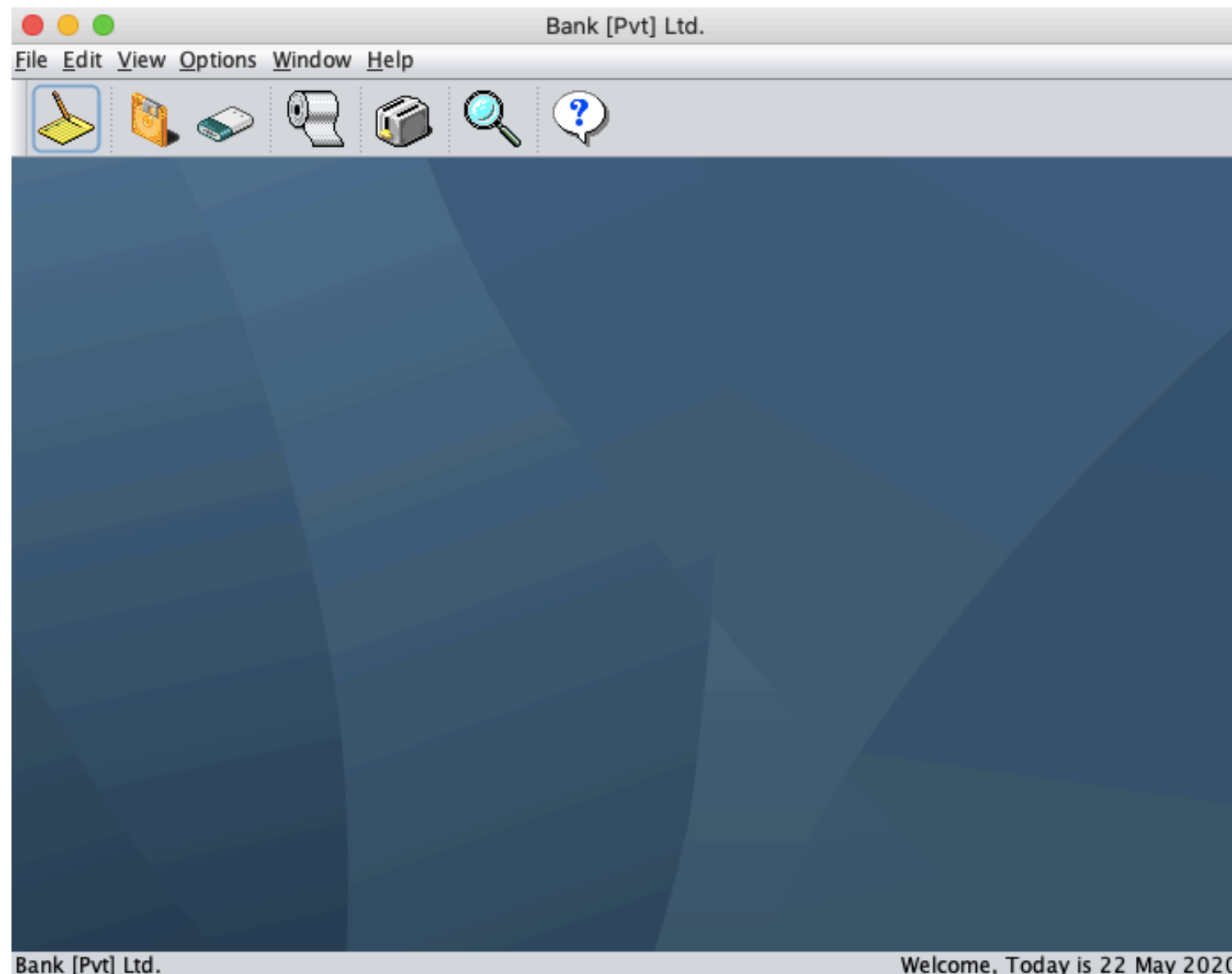
BANKING SYSTEM

JAVA (OOCp) Project

Team Member - 1: Vinay Omkumar Khilwani - 18BCP126 (Leader)

Team Member - 2: Vasu Gondaliya - 18BCP123

Team Member - 3: Umang Mehta - 18BCP120



ABSTRACT

- The broad increase in the advancement of technology has also led to technological advances in the Banking Sector as well. An efficient banking system will lead to more easiness in the Banking facilities such as Loan, Deposit, Withdrawal, Change of Address, etc. The Banking System, designed by us is an efficient, 'easy to work with' and user-friendly Banking System, which includes some of the Banking Facilities, such as "Account Creation", "Account Deletion", "Withdraw Money", "Deposit Money", "View All Accounts". The data, about all the customers is stored, moderated and retrieved from a particular separate file, using 'File Handling'.

STAGE-1

-
- We thought of the functionalities, we want in our Banking System, which include "Deposit Money", "Withdraw Money", "New Account", "Delete Account", "View Account(s)", & "View All Accounts".
 - We made the class files of these functionalities, in the Stage-1 of our project.
 - When using these functionalities, the System basically interacts with the data stored in the database File, "Bank.dat".

The image displays five Java Swing windows for a banking system interface:

- Create New Account:** Fields for Account No., Cust. Name, Date (Month, Day, Year), and Dep. Amount. Buttons: Save, Cancel.
- Delete Account Holder:** Fields for Account No., Cust. Name, Last Trans., and Account Bal. Buttons: Delete, Cancel.
- Withdraw Money:** Fields for Account No., Cust. Name, Date (Month, Day, Year), and Amount. Buttons: Save, Cancel.
- Search Customer [By No.]:** Fields for Account No., Cust. Name, Last Trans., and Balance. Buttons: Search, Cancel.
- View All Account Holders:** A table displaying account information.

Account No.	Customer Name	Opening Date	Bank Balance
1	1	January 1, 2020	1

STAGE-2

- We thought, to make the System more user-friendly, by adding the functionality to change "Theme", "Background", and/or "Layout Style" of the System.
- This will encourage the user to work with more engaging feel, on the System.
- Hence, we made class files of the "Themes", and "Layout Styles", which we thought to add, in this stage.

STAGE-3

.....

- We decided to make the Menus and Menu-Bar, more better visually, we decided to add images for some types of facilities.
- We also made class, for "Help" and "About System" features of the overall System.
- This was the last stage of the implementation, of our Project.

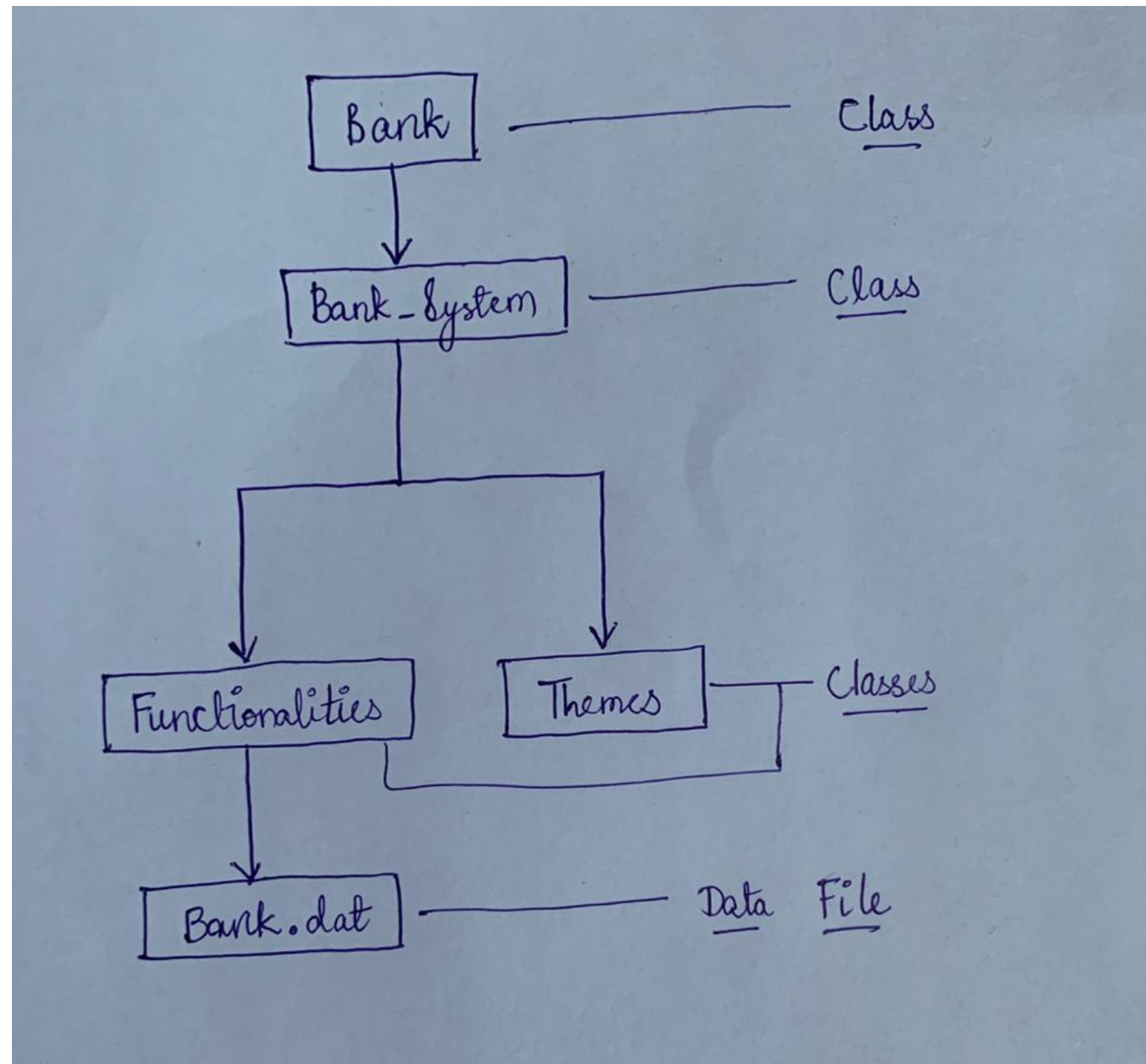


FUNCTIONALITIES OF OUR PROJECT

- Open New Account: Open New Bank Account.
- Deposit Money: Deposit Money, into a particular account.
- Withdraw Money: Withdraw Money, from a particular account.
- Search by Account No.: Search Bank Account, by Account No.
- Search by Name: Search Bank Account(s), by
- View Accounts, one by one: View all the Bank Accounts, one by one.
- View All Accounts: View all the current Bank Accounts, in a Tabular Format.
- Change Background Colour: Change Background Colour of the System.
- Change Layout Style: Change the Layout Style of the System.

FUNCTIONALITIES OF OUR PROJECT

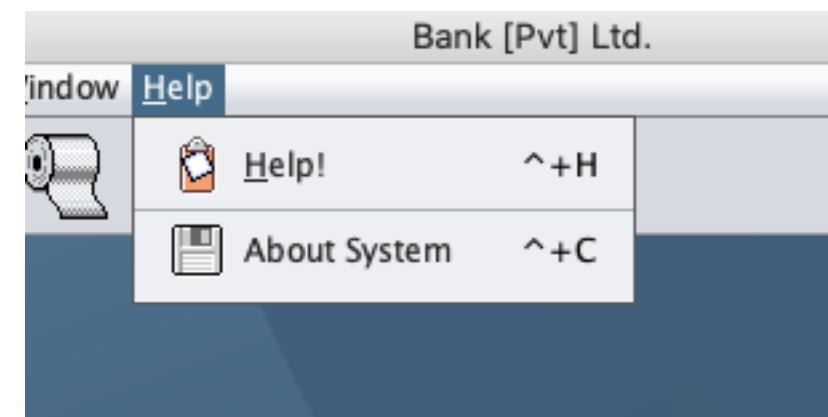
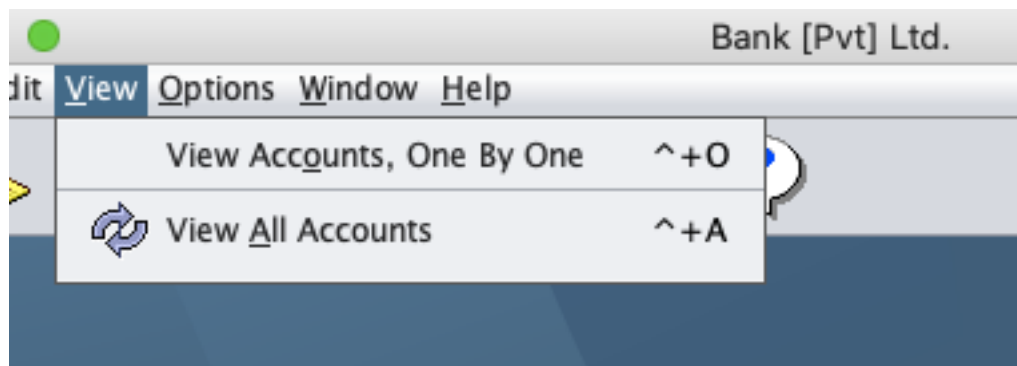
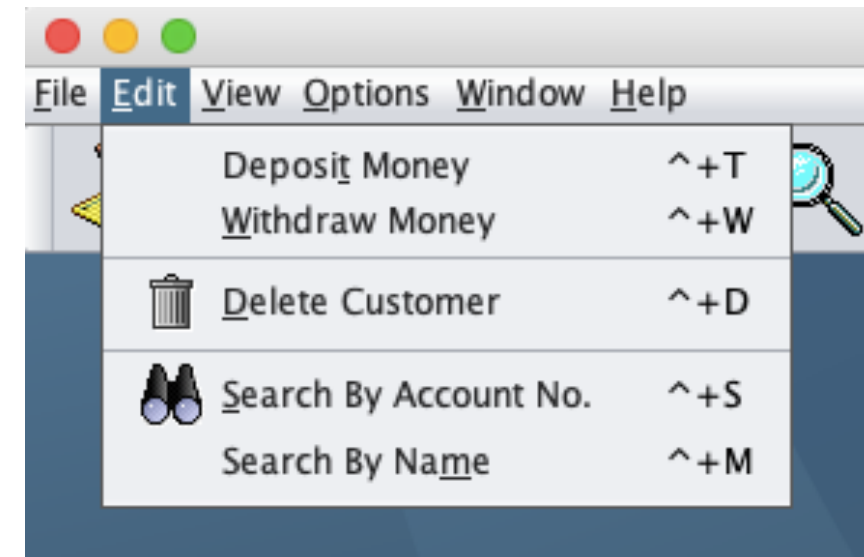
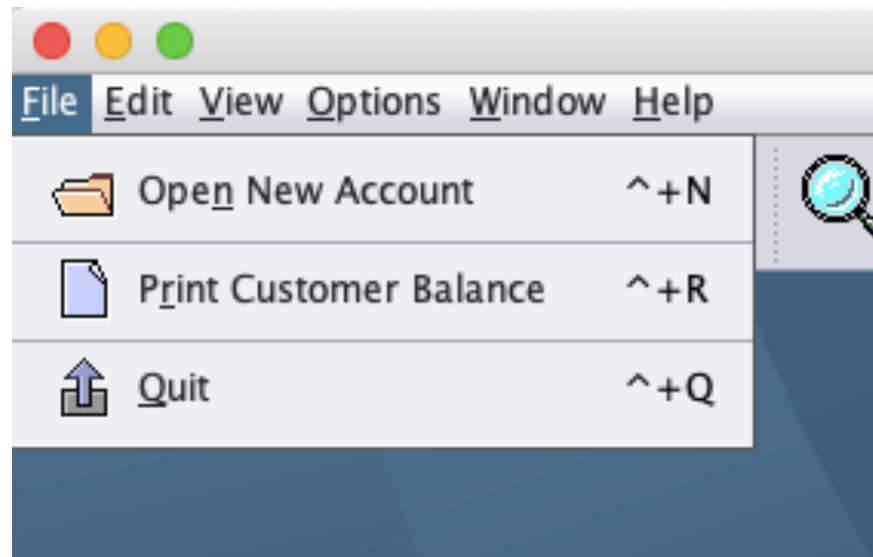
- Delete Customer: Delete a Bank Account.
- Apply Theme: Change the theme of the System.
- Close All Windows: Close all the open Windows.
- Close Active Window: Close the current open Window.
- Help: Any Help, related to the System, which redirects to a Web Page/HTML File.
- About the System: About the System (Creation and Name).
- Print Customer Balance: Directly Print "Customer Balance".
- Quit: Quit the System.



Flow-Chart of the "Project Implementation"

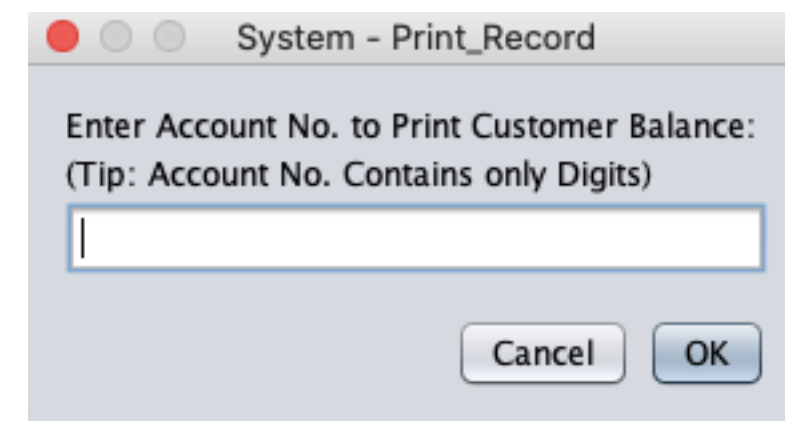
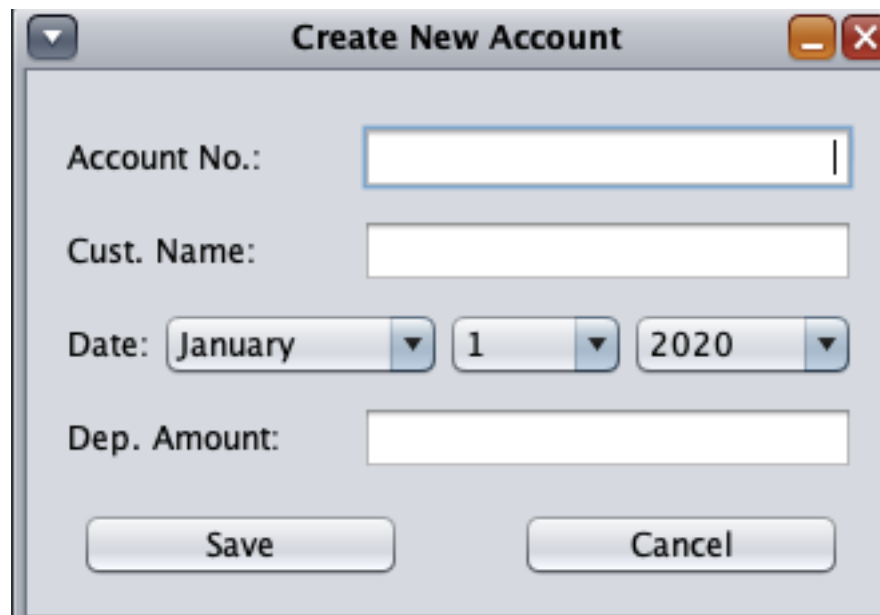
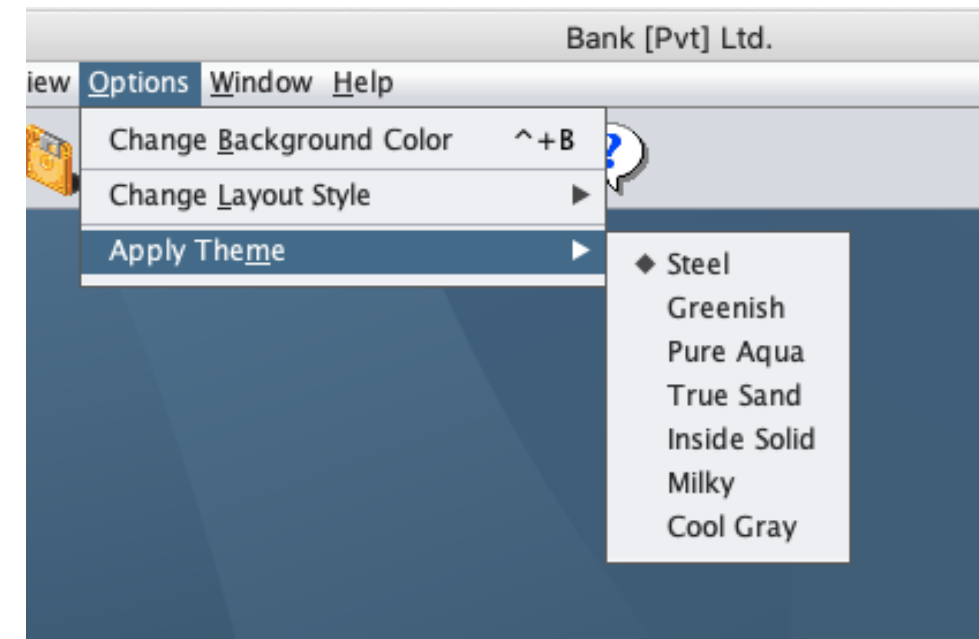
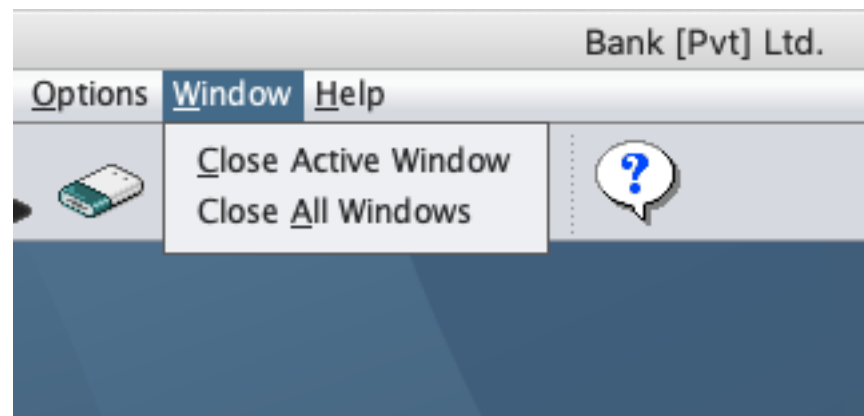
SCREENSHOTS OF THE WORKING SYSTEM

.....



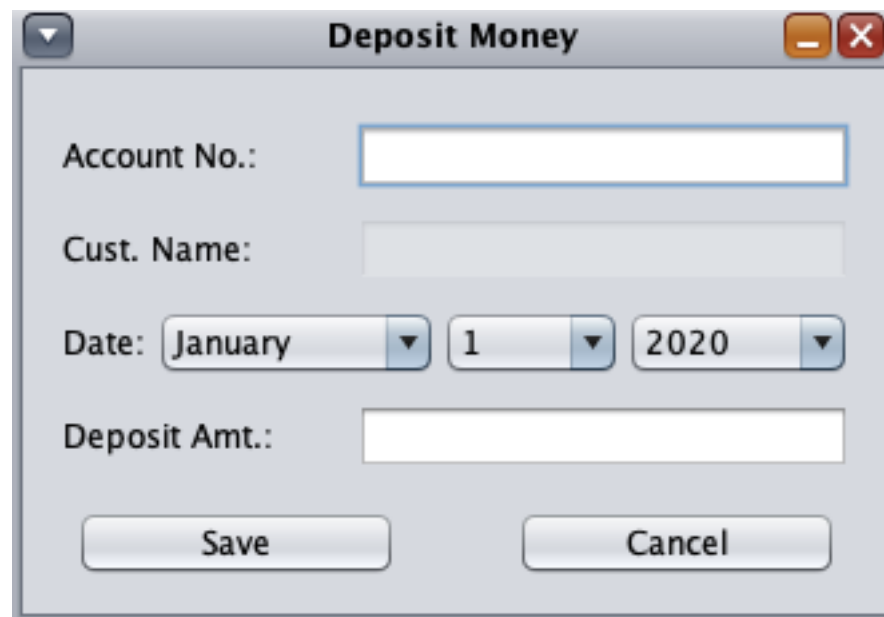
SCREENSHOTS OF THE WORKING SYSTEM

.....



SCREENSHOTS OF THE WORKING SYSTEM

.....



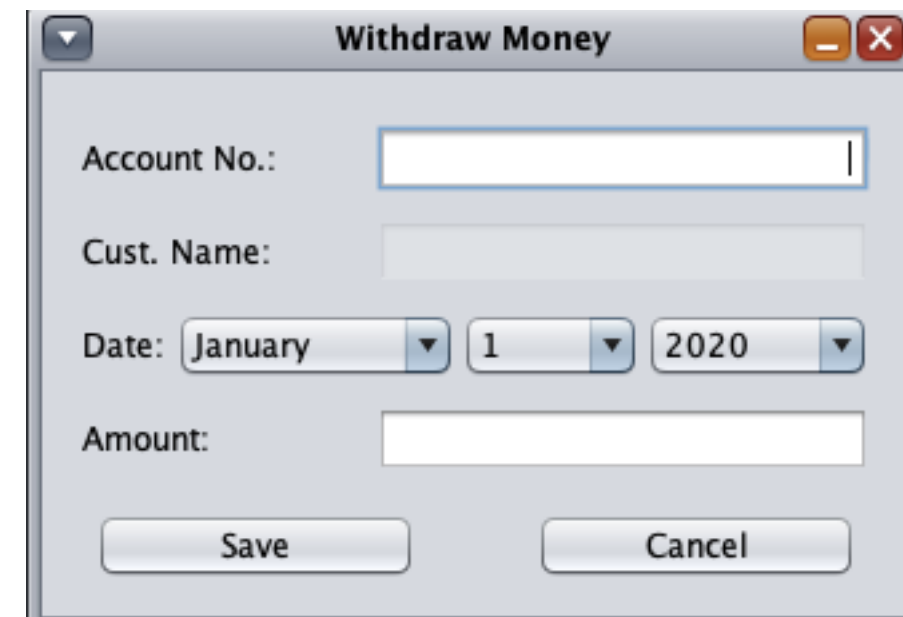
Deposit Money

Account No.:

Cust. Name:

Date:

Deposit Amt.:



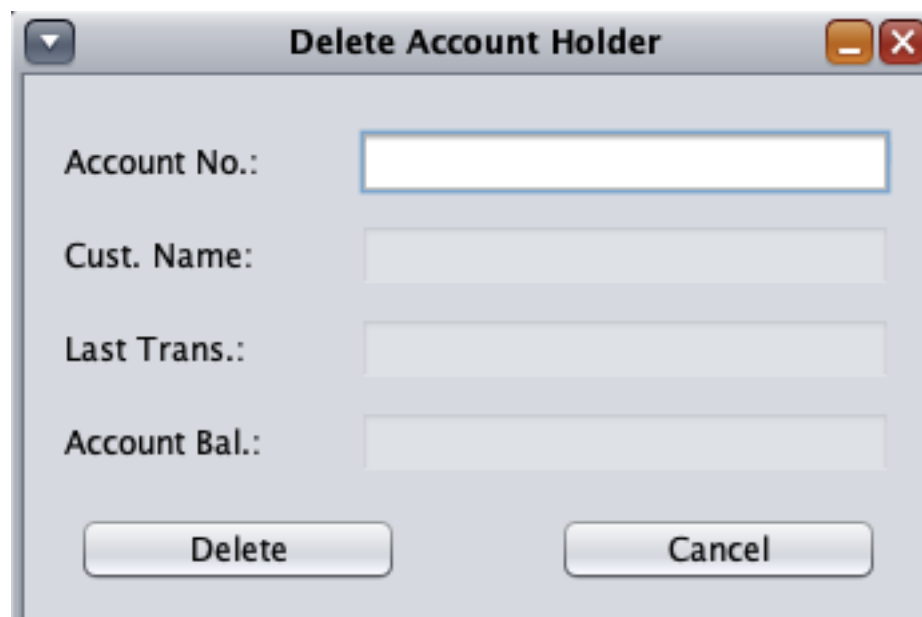
Withdraw Money

Account No.:

Cust. Name:

Date:

Amount:



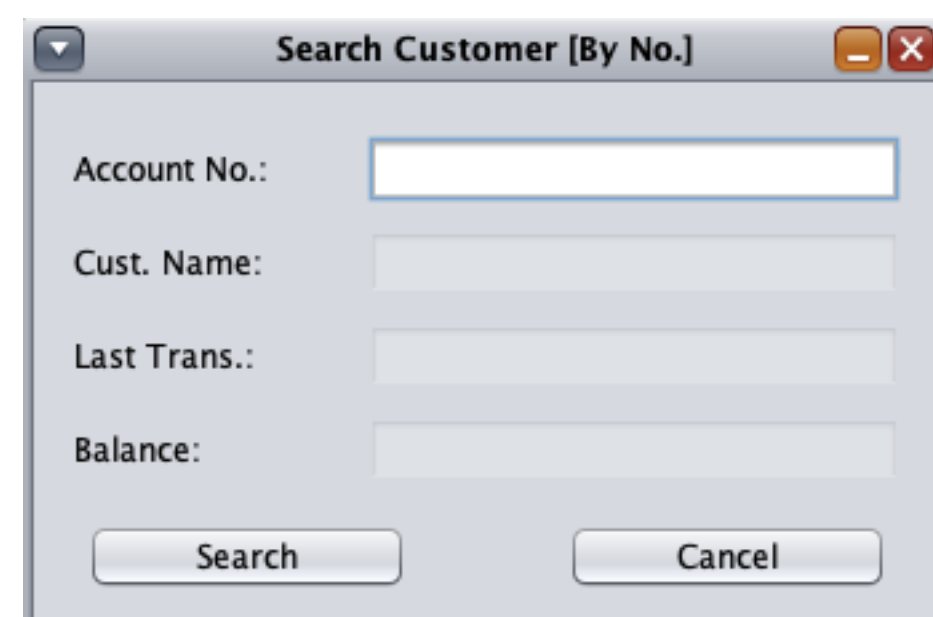
Delete Account Holder

Account No.:

Cust. Name:

Last Trans.:

Account Bal.:



Search Customer [By No.]

Account No.:

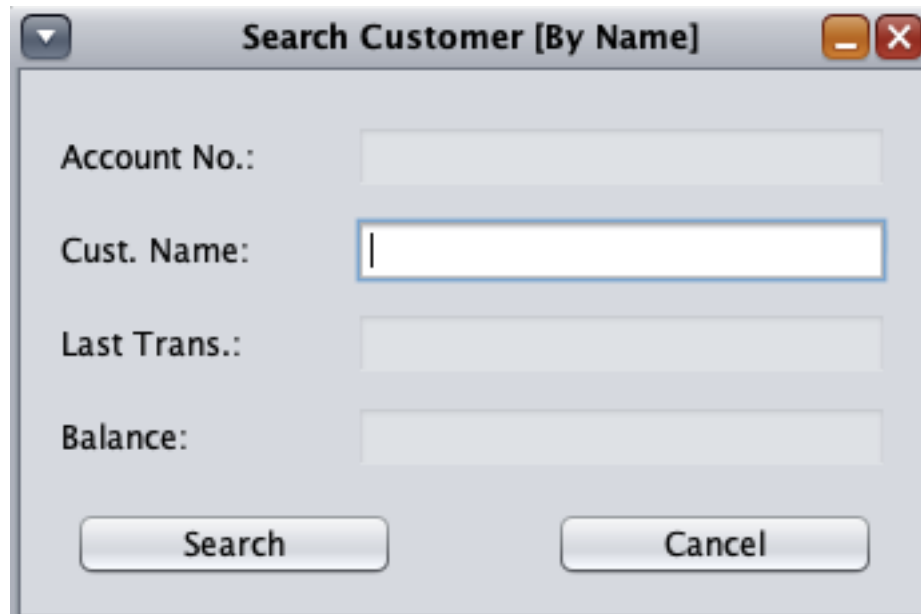
Cust. Name:

Last Trans.:

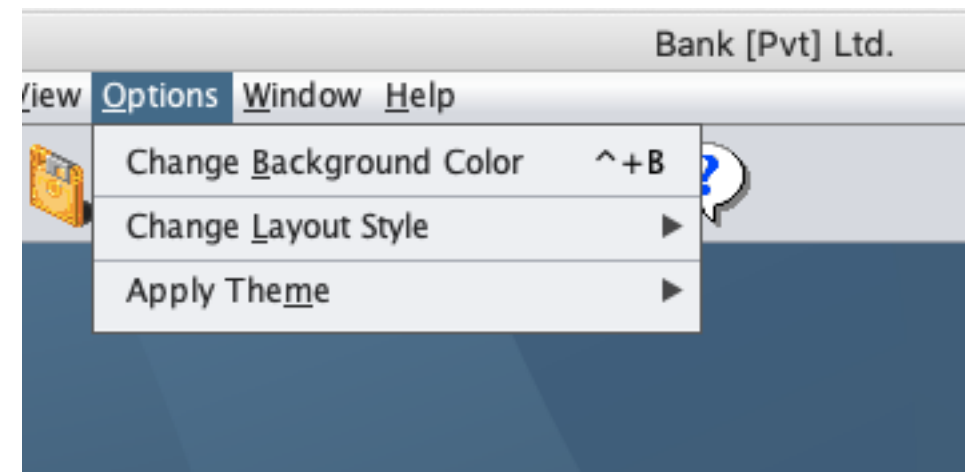
Balance:

SCREENSHOTS OF THE WORKING SYSTEM

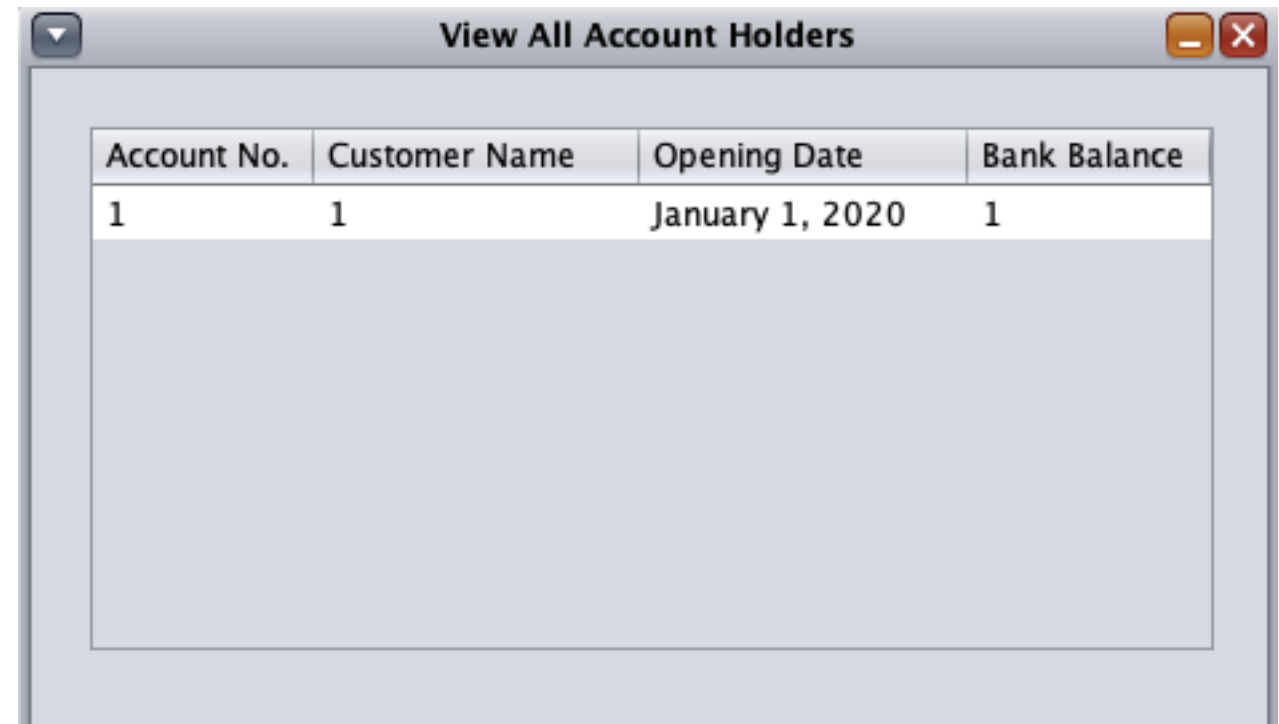
.....



A dialog box titled "Search Customer [By Name]" with a dropdown arrow on the left and standard window controls on the right. It contains four input fields: "Account No.:", "Cust. Name:" (which is currently selected), "Last Trans.:", and "Balance:". At the bottom are "Search" and "Cancel" buttons.



A dialog box titled "View Account Holders" with a dropdown arrow on the left and standard window controls on the right. It contains four input fields: "Account No.:" (with value 1), "Cust. Name:" (with value 1), "Last Trans.:" (with value January 1, 2020), and "Balance:" (with value 1). At the bottom are navigation buttons: "<<", "<", "1/1", ">", and ">>".

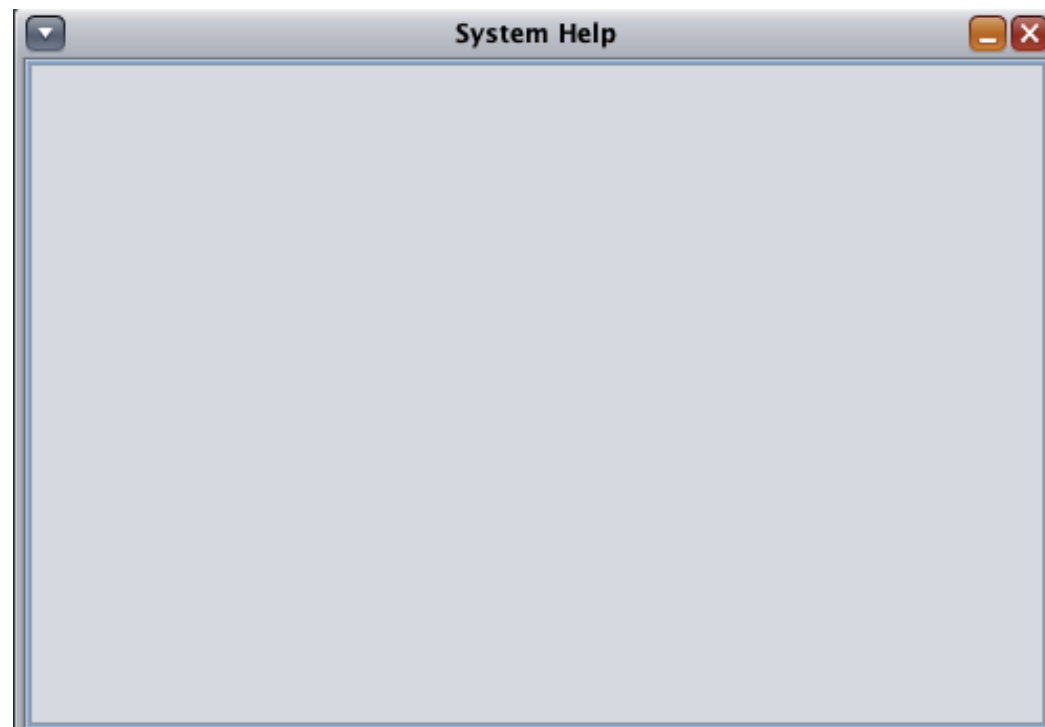
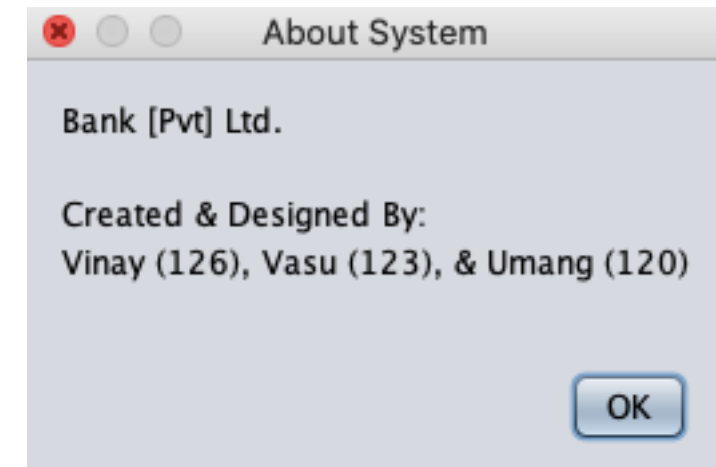
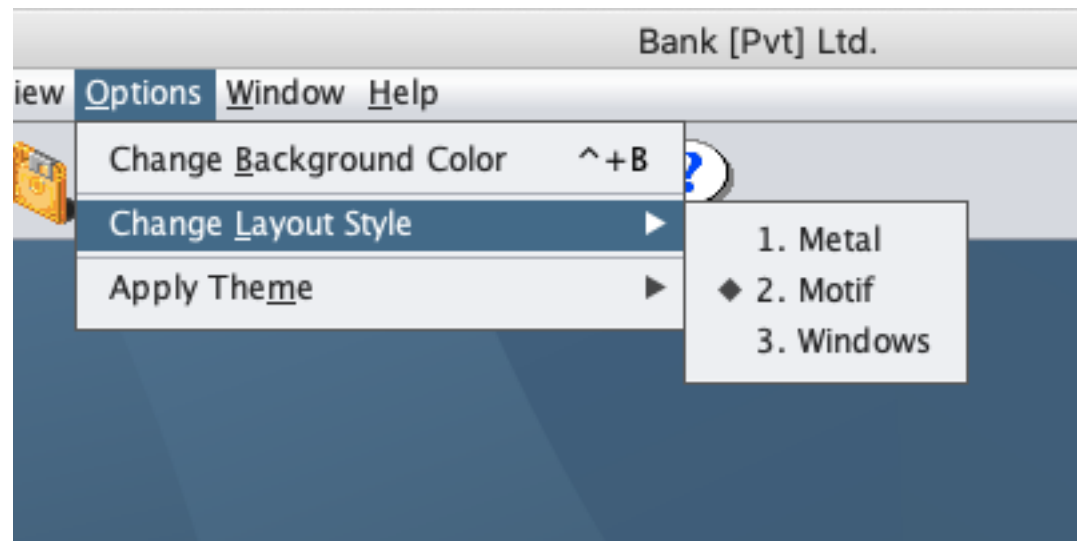


A window titled "View All Account Holders" with standard window controls. It displays a table with account information.

Account No.	Customer Name	Opening Date	Bank Balance
1	1	January 1, 2020	1

SCREENSHOTS OF THE WORKING SYSTEM

.....



EXCEPTION-HANDLING IN THE SYSTEM

.....

Type of Exception	Task, for which the Exception is Handled
IOException	Caught, while moderating and accessing the Data File, of the Bank Accounts, and also while some other actions.
MalformedURLException	It is caught, when the URL of the Web Page, linked to "Help" Section of the System, is Malformed or erroneous.
EOFException	Caught, when "Printing Customer Balance" Sheet, and also while some other actions.
Exception (Any Kind of Exception)	This exception is caught, for many actions, which includes "Closing a Window", "Quitting the System", "File Reading and Writing", etc.

IMPLEMENTATION DETAILS

➤ GUI Used: Java Swing

➤ Components of GUI, used:

- JFrame
- JDesktopPane
- JMenuBar
- JMenu
- JMenuItem
- JButton
- JLabel
- JToolBar
- JPanel
- JPopupMenu
- JScrollPane
- JInternalFrame
- JEditorPane
- JWindow
- JTextField
- JComboBox
- JTable
- JComponent
- JOptionPane

IMPLEMENTATION DETAILS

➤ Other Component(s) Used (other than 'General Ones'):

- ColorUIResource
- SwingUtilities

➤ Other Libraries Used (other than 'General Ones'):

- swing.border.*
- swing.plaf.*
- swing.plaf.metal.*
- beans.*
- text.*
- net.URL
- net.MalformedURLException
- awt.event.*
- awt.PrintJob.*

'FILE HANDLING' DETAILS

- 'FileInputStream' and 'DataInputStream': Used to retrieve the data from the file "Bank.dat", for functionalities like "View Accounts, One by One", "View all Accounts", etc.
- 'FileOutputStream' and 'DataOutputStream': Used to write the data into the file "Bank.dat", for functionalities like "New Account", "Delete Account (Customer)", etc.

DETAILS ABOUT CLASSES AND INTERFACES

- *In-Built Interfaces Used:*

- ActionListener
- PropertyChangeListener
- HyperLinkListener

- *'Class' Flow of the System:*

- Bank (Class)
 - Bank_System (Class)
 - Classes of all the Facilities, of Banking
 - All Classes for Generating the Different Themes

APPLICATIONS OF OUR PROJECT

- Can be used as an efficient System in real-world Banks and Companies.
- Automating Customer Bank Accounts.
- Maintaining human error-free records, of transactions and customers, in the Bank System.
- Maintaining centralised data, about the customers and transactions of the Bank.

REFERENCES

- <https://www.javatpoint.com/java-jcombobox>
- <https://docs.oracle.com/javase/7/docs/api/javafx/swing/plaf/ColorUIResource.html>
- <https://www.programcreek.com/java-api-examples/index.php?api=javafx.swing.plaf.ColorUIResource>
- <https://www.javatpoint.com/java-jpopupmenu>
- <https://docs.oracle.com/javase/7/docs/api/javafx/swing/JOptionPane.html>