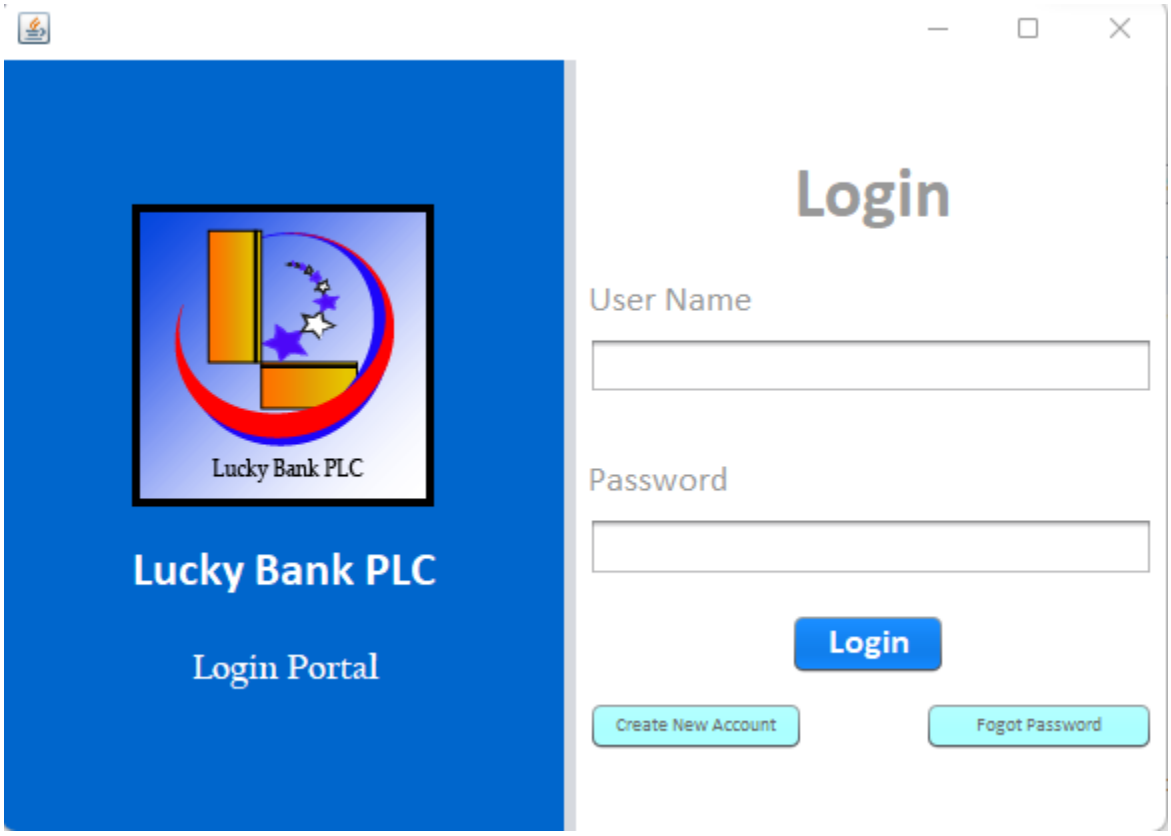
Lucky Bank Smart Banking App
(Java)
Project Specification

This is simply a document that describes what each interface in the software does and not a professional document. Hope you can get an idea about the project.

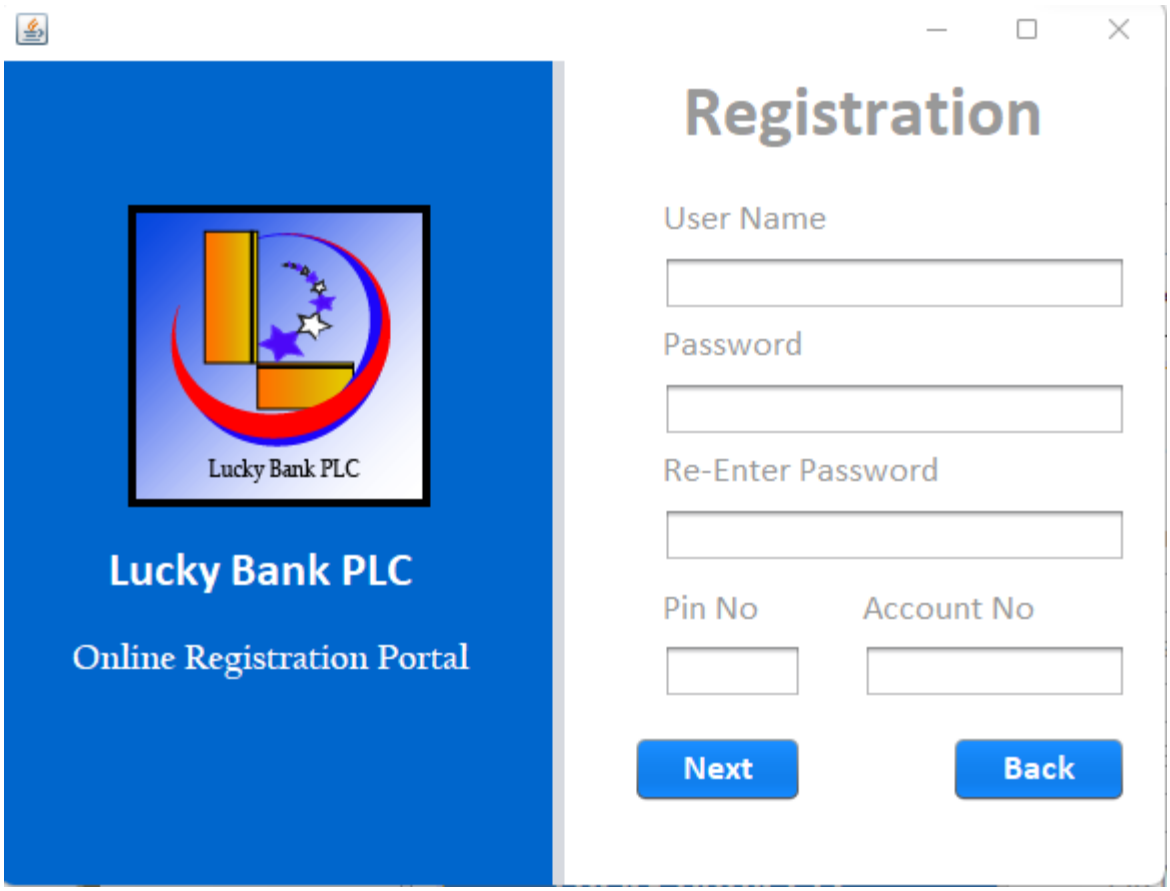
- **Interface 01 – Login.**



- Login interface is simply accepting a user name and a password as inputs.
- After the login button clicked code checks if any field is empty. If any field is empty interface outputs an error message. If all the fields are filled,
- Then the code checks if there is a username similar to the input in the customer_login_details table in the lucky_bank_db and checks if the password corresponding to that username is similar to the user input password.
- If the condition passed, code automatically clears every record in a table called temp in the lucky_bank_db and selects * in the customer_login_details table where username equals to user input and insert into it. (I've used this table to authenticate the user). Every time a new user logged in, above process repeats and table only consists only one username, password and a pin. (I've used this pin so it could use as a verification pin in password recover phase and assume it as the security pin of the credit card).

- At the same time interface navigates into Main_Menu interface.
- If the condition failed that means username and password not available in the database interface outputs a Try Again message and nothing happen to the database.
- Create New Account button navigates into Registration interface.
- Forgot Password button navigates into change_password interface.

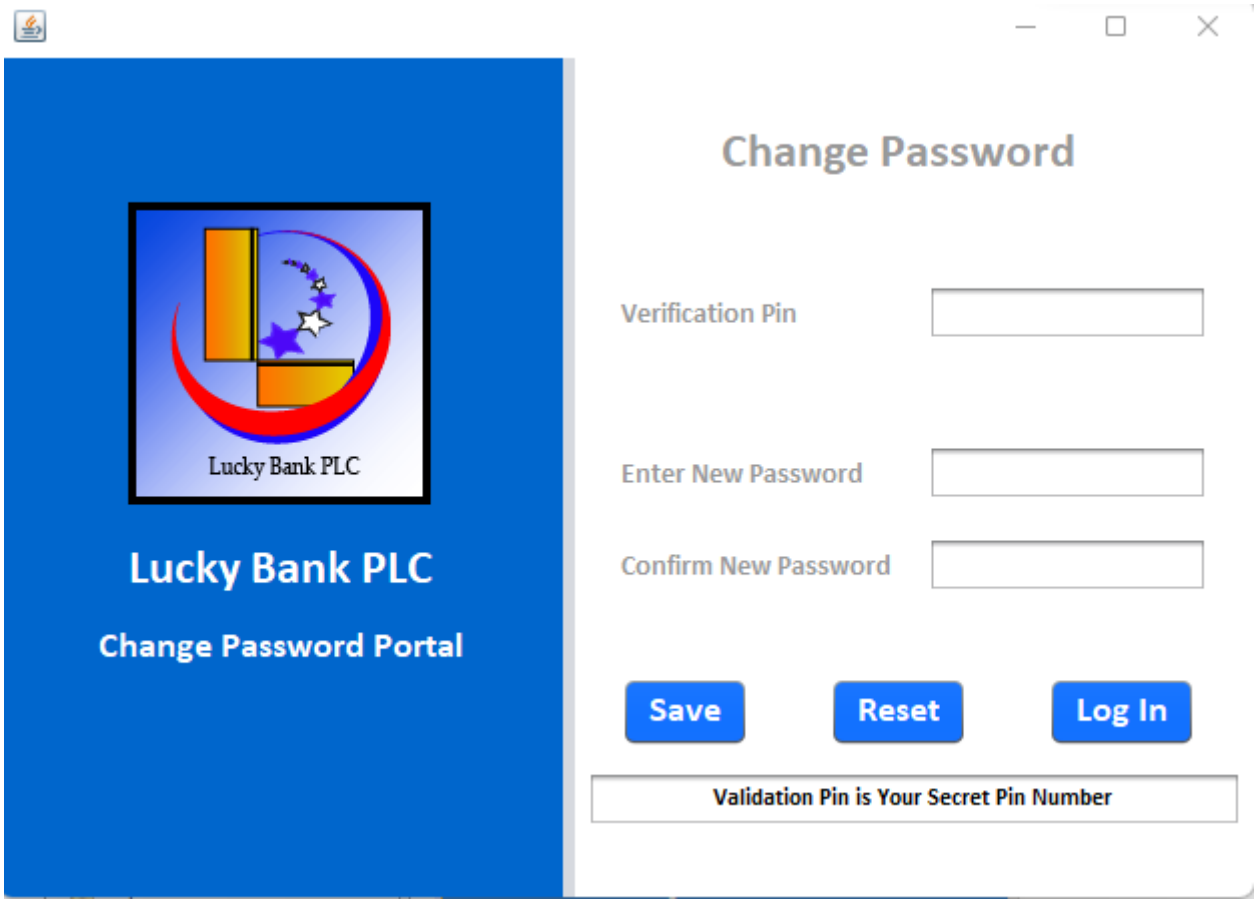
- **Interface 02 – Registration.**



- Registration interface is accepting Username, Password, Re – Enter Password, Pin No and an Account No.
- When clicked next, code checks if any fields are empty. If any field is empty interface outputs an error message. If all the fields are filled,
- Then the code checks if the username already exists in the customer_login_details table in the database. (This prevents duplicate username existence). If username exists code outputs an error message as “username already exists.”
- If the username is unique, code checks if the pin no already exists in the customer_login_details table (this prevents creating multiple accounts for a single bank account). If pin exists interface outputs an error message as “There is an existing account with this pin”.

- If pin is unique, code checks if there is a bank account similar to the user input and its pin is similar to the input pin in customer_details table. (Assume this table is created by the bank). If the condition failed interface outputs an error message as “Your Account_No and Pin_No Doesn't Match”
- If the condition passed, code checks if Password and Re-Enter Password is same. If they are different, interface outputs an error message as “Check Your Passwords”.
- If Passwords are same, code inserts user’s input data into the customer_login_details table and outputs “Account Creation Success” message box.
- After the process interface navigates into Login Page again and also clicking back button navigates into the Login Page as well.

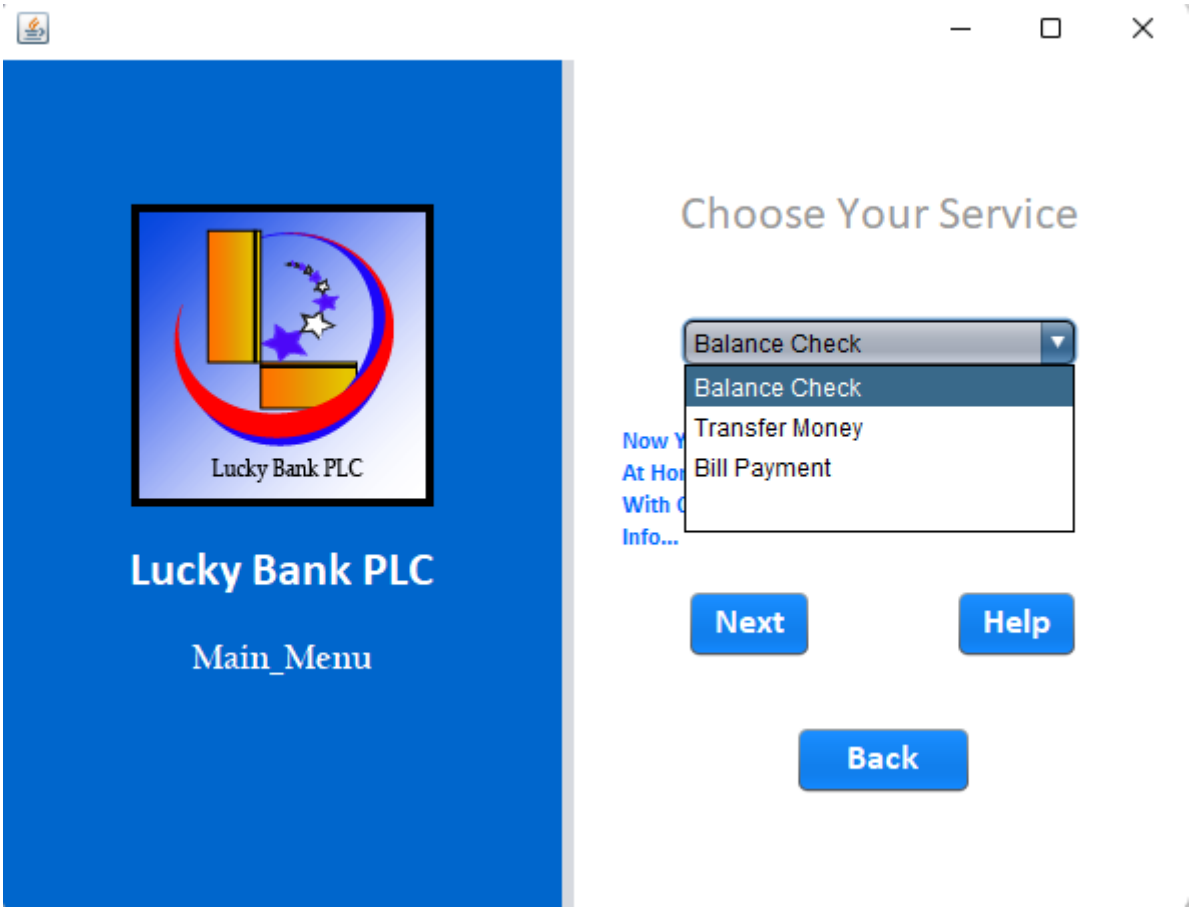
- **Interface 03 – Password Change.**



- Change Password interface accepts Verification Pin (which I assumed as the security pin in the card), New Password and Confirm New Password.
- When clicked Save, code checks if any fields are empty. If any field is empty interface outputs an error message. If all the fields are filled,
- Code checks if verification pin exists in the customer_login_details table. If it doesn’t exist, Interface outputs an error message as “Check Your Pin No”.

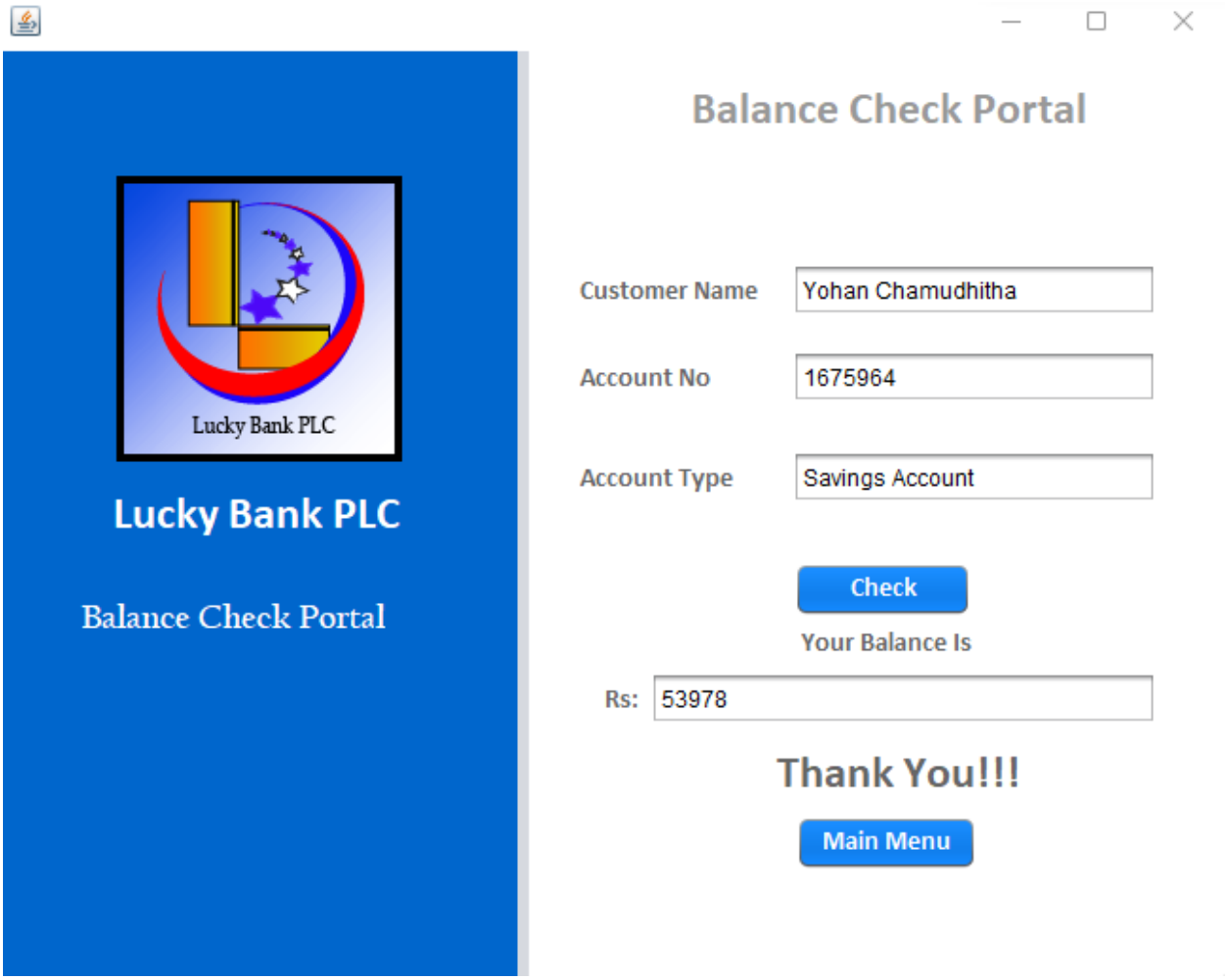
- If the condition passed customer_login_details table's record corresponding to the pin gets updated with the new password.
- Reset button resets the fields as well as Log In button navigates into Login Page.

- **Interface 04 – Main Menu**



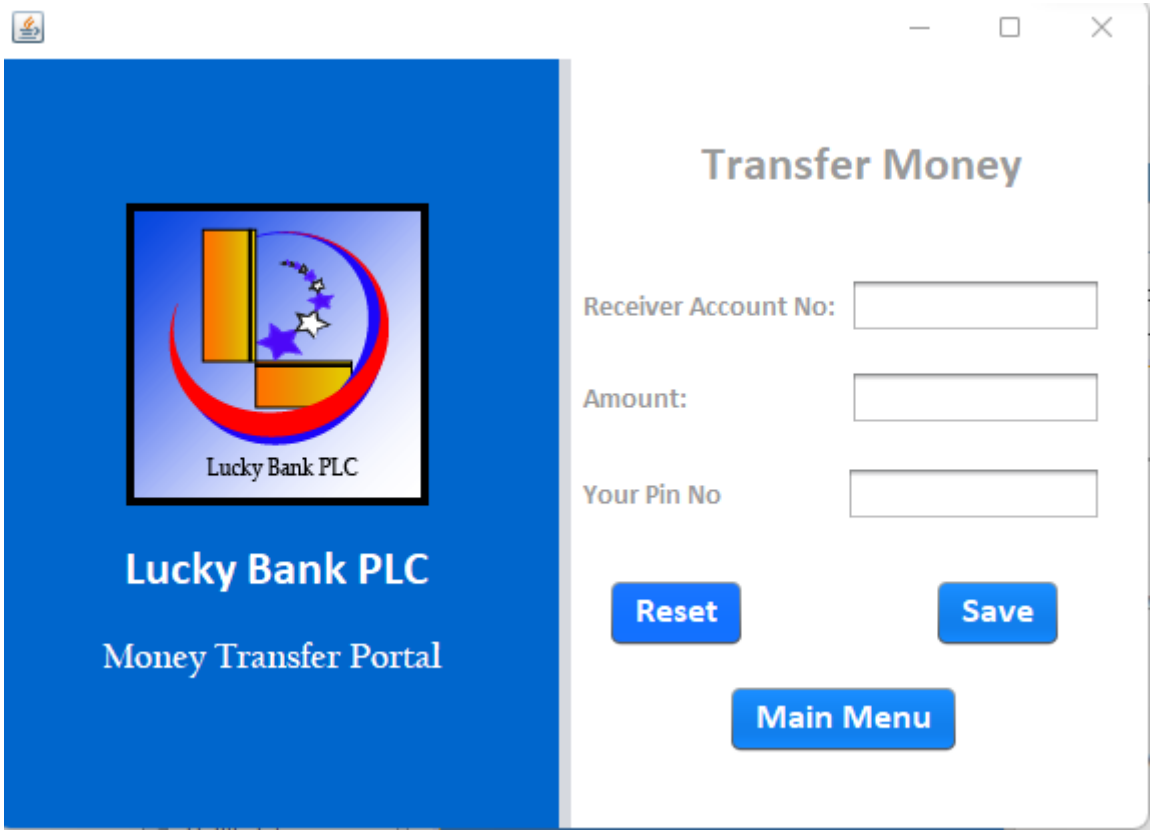
- Main Menu interface is simply a navigation page.
- There is a dropdown menu consists of three items, Balance Check, Transfer Money and Bill Payment.
- User can navigate into these interfaces by selecting a item and click next.
- Help button is navigating into help interface (still not implemented) and Back button navigate into Login Page.

- **Interface 05 – Balance Check**



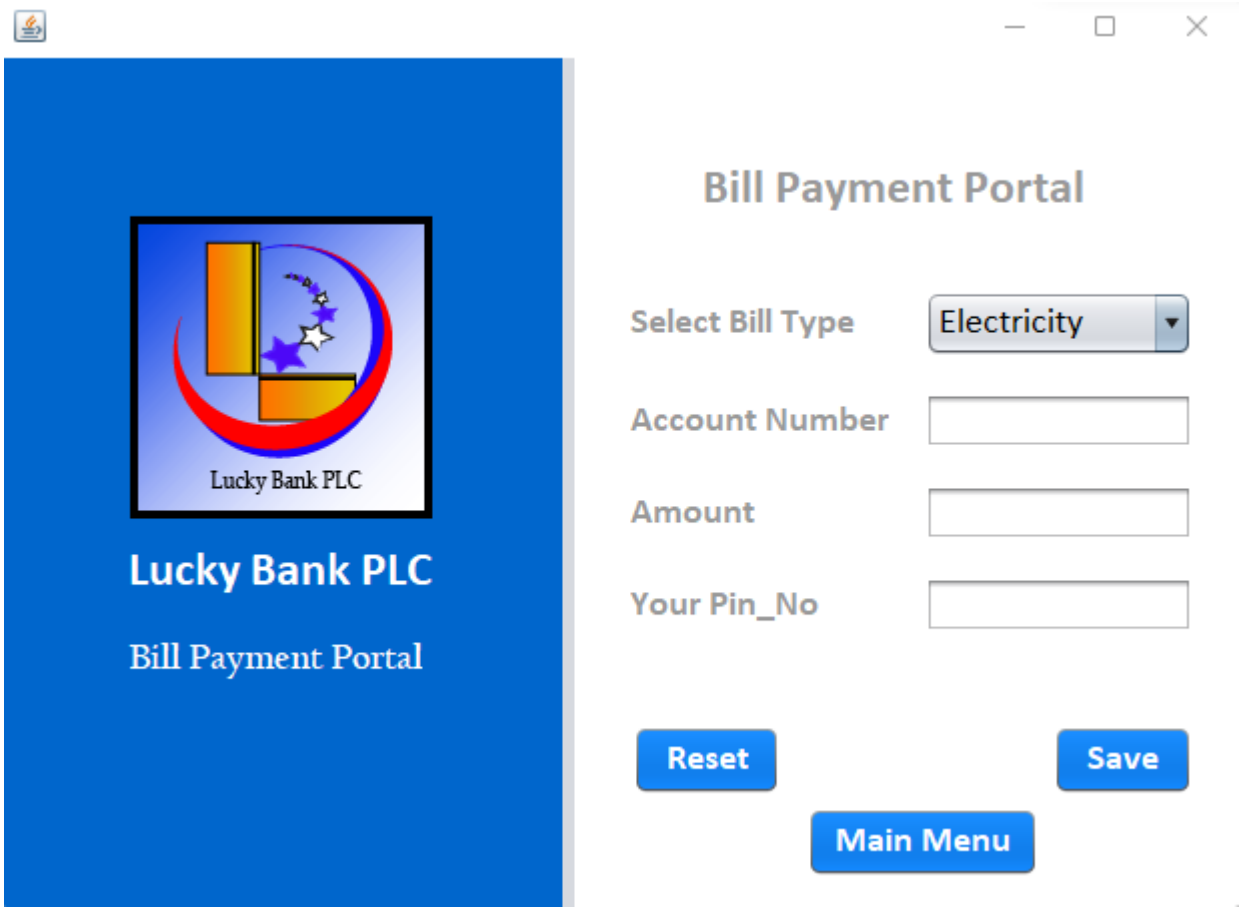
- Balance check interface is showing customer name, account number, account type and balance.
- When the user clicked check button code searches what are the details saved on the temp table on the database.
- If there is a record, code takes the pin no and selects all the details on customer_details table corresponding to that pin no and outputs those details on the corresponding fields.
- Main Menu button navigates back to main menu.

- **Interface 06 – Transfer Money**



- Transfer money interface takes three inputs from user which are Receiver's account no, Amount and Sender's pin no.
- When user clicked save button code first checks if all the fields are filled. Otherwise, interface outputs an error message.
- If all the fields are filled, code checks if there's an account corresponding to receiver's account number. If there aren't interface outputs an error message as "Check Your Receiver's Account Number".
- If there is an account, the code checks the details saved on the temp table and selects the pin_no.
- Then code selects the balance corresponding to that pin no from customer_details table.
- Then code checks, If that balance is larger than the amount user input. If not interface outputs an error message as "Sorry, Insufficient Balance".
- Otherwise, code runs two processes at the same time,
    - First one is updating the balance of receiver's account by add it the input amount.
    - Second one is updating the balance of sender's account by subtracting the input amount.
- If everything successful interface outputs a "Transfer Successful" message box.

- **Interface 07 – Bill Payment**



- Bill payment interface consists of a drop-down menu which has 4 items (Electricity, Water, SLT and Ez Cash) and three input fields Account Number (bill related), Amount and Your Pin No.
- When user selects an item from the drop down, fill the fields and clicked save button, code first checks if all the fields are filled. Otherwise, Interface outputs an error message.
- If all the fields are filled code, then selects the pin_no from temp table and checks it with input pin no. If those two are not the same interface outputs an error message as “Check Your Pin No”.
- If those two are same, code the selects the balance from customer_details table corresponding to the pin_no.
- Then code checks if that balance is greater than the input amount. If it isn’t Interface outputs an error message as “Sorry, Insufficient Balance”.
- Otherwise, code selects * from bill_payments table in electricity_board_payments database where account number is similar to user input. If there isn’t any account corresponding to that acc no, Interface outputs an error message as “Check Your Account No”.
- Otherwise, code runs two processes at the same time,
    - First one is Insert into bill_payment table Bill_No, Customer_acc_no, Payment_type, Timestamp, Amount and balance.

        - Bill No – code takes existing bill_no from the table and updates it by adding 1.
        - Customer_accc_no – Same as the user input.
        - Payment_type – Always the inserting “Online”.
        - Timestamp – Exact timestamp of the device when the transaction happens.
        - Amount – Amount that user Inputs.
        - Balance – Code takes existing balance from the table and updating it by subtracting the amount.
    - Second process is updating customer_details table of lucky_bank_db by subtracting the amount from the balance of the account corresponding to the pin _no.
- If everything happens as planned Interface outputs an error message as “Your Electricity Bill_Payment Success”.
- Above process is the same process that runs when Water and SLT items selected, except they update different table on different databases.
- For EZ_Cash option process is same as above except instead of subtracting amount it adds to the balance here and update the corresponding database.

- **Additional Details.**

    - There are five databases I’ve connected them using separate coding.

    - lucky_bank_db
        - customer_details
            - Customer_name
            - Account_Type
            - Account_No
            - Pin_No
            - ID_Type
            - ID_No
            - Phone_No
            - Balance
        - customer_login_details
            - User_Name
            - Password
            - Pin_No
        - temp
            - User Name
            - Password
            - Pin No

- electricity_board_payments
  - bill_payments
    - Bill_No
    - Customer_acc_no
    - Payment_Type
    - Timestamp
    - Amount
    - Balance
  - customer_details
    - Customer_Name
    - Customer_acc_no
    - Acc_balance
- Water_department_payments
  - bill_payments
    - Bill_No
    - Customer_acc_no
    - Payment_Type
    - Timestamp
    - Amount
    - Balance
  - customer_details
    - Customer_Name
    - Customer_acc_no
    - Acc_balance
- slt_payments
  - bill_payments
    - Bill_No
    - Customer_acc_no
    - Payment_Type
    - Timestamp
    - Amount
    - Balance
  - customer_details
    - Customer_Name
    - Customer_acc_no
    - Acc_balance

- ez_cash_balance
  - bill_payments
    - Bill_No
    - Customer_acc_no
    - Payment_Type
    - Timestamp
    - Amount
    - Balance
  - customer_details
    - Customer_Name
    - Customer_acc_no
    - Acc_balance

This project consists most basics of java programming if anyone has suggestions, please send them to ychamu2@gmail.com.

Project Creator – Yohan Chamudhitha Amarasekara.

Thank You!