CS4001NI Programming

30% Individual Coursework

2023-24 Spring

Student Name: Rounak Joshi

London Met ID: 22067875

College ID: NP01CP4A220426

Group: C14

Assignment Due Date: Wednesday, May 10, 2023

Assignment Submission Date: Wednesday, May 10, 2023

TABLE OF CONTENTS

TABLE OF FIGURES

Table of table

1.INTRODUCTION

The following coursework has asked to create a class within the project that we had made as a first part of the coursework. In this class we are required to create a GUI (Graphical User Interface) for the system that stores details of bank card in an ArrayList. In this GUI, there are two main frames (Debit card and Credit card) in which the card details and user information are added and the card is added through JButton in the arraylist. There are also two other frames which are Withdraw frame in which the user can withdraw money from their card which is added in the arraylist. Then we have set limit frame in which we can add credit limit and grace period to the credit card. This all action is done through adding information to the various textfileds and the process is performed through JButton which is added in actionLIstener. The frames are all added to different Jpanels. The required Jpanel is set to true while other are set to false and can be changed through JButton. There are also display button which displays the information of given user which is done by calling the display method of the required class. There is also clear button that clears the textfield by setting text to null. The GUI is connected all three classes as it is called through the arraylist of the bank card class. There have been various tools used during the process of doing this coursework. The tools that have been used are as follows-

DRAW.IO

Draw.io is proprietary software for making various types of diagrams and charts. This software lets you choose from an automatic layout function and create a custom layout. (Computer Hope, 06/02/2020)

BLUEJ

BlueJ is an Integrated Development Environment that allows you to develop Java programs quickly and easily. This software is simple, interactive, portable, designed for teaching and innovative and helps to provide a more precise interface for creating projects and coding in Java. It is a port of Blue to java which was released in 1999 by Michael Kolling. (Rouse, What Does BlueJ Mean?, 10 October, 2013)

MS WORD

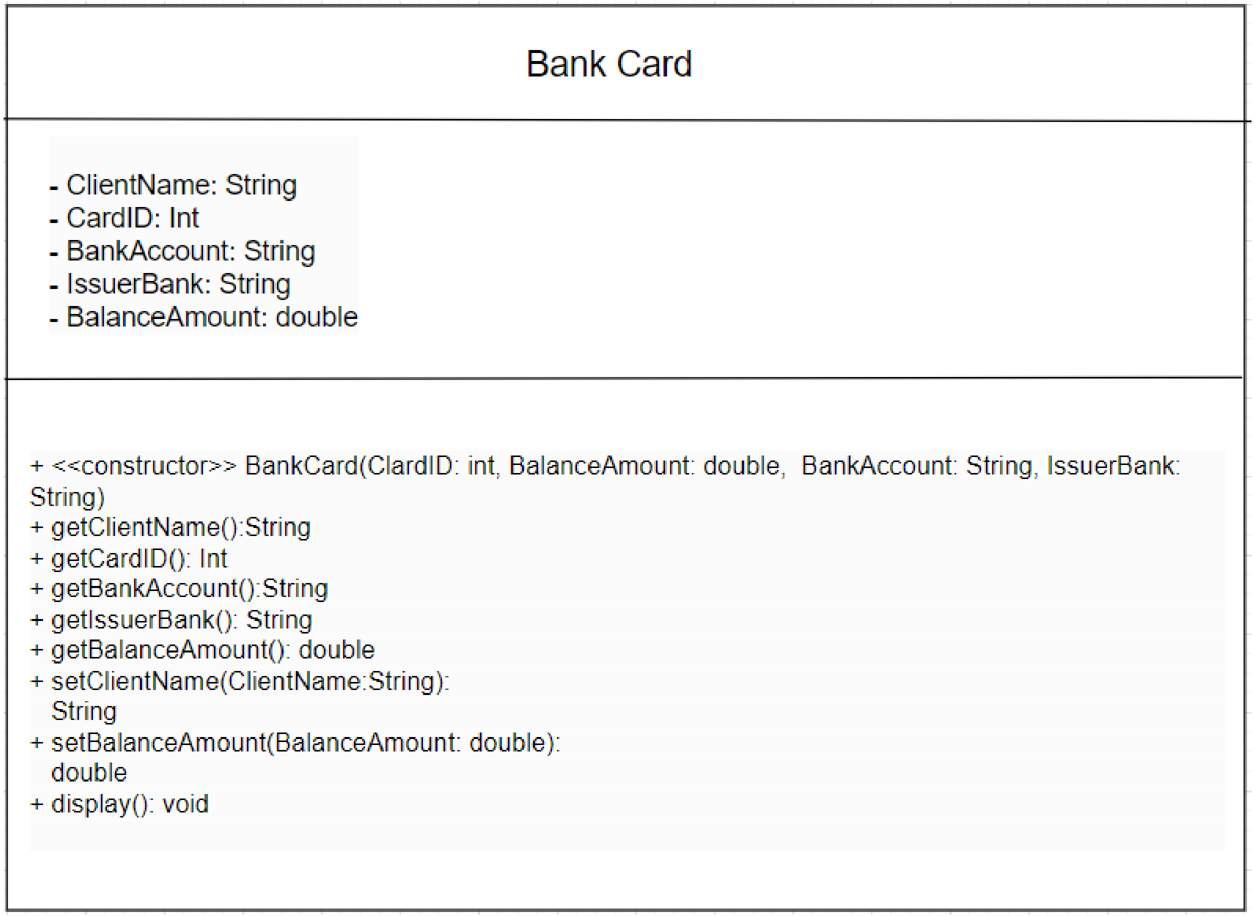
Microsoft Word is a widely used commercial word processor designed by Microsoft. Microsoft Word is a component of the Microsoft Office suite of productivity software, but can also be purchased as a stand-alone product. (Rouse, What Does Microsoft Word Mean, 10 August, 2022)

MOQUPS

Moqups is a web-based design tool that can help you build wireframes, mockups and prototypes all within one environment. (Pietroluongo, September 19, 2022)

2.CLASS DIAGRAM

2.1. CLASS DIAGRAM OF BANK CARD



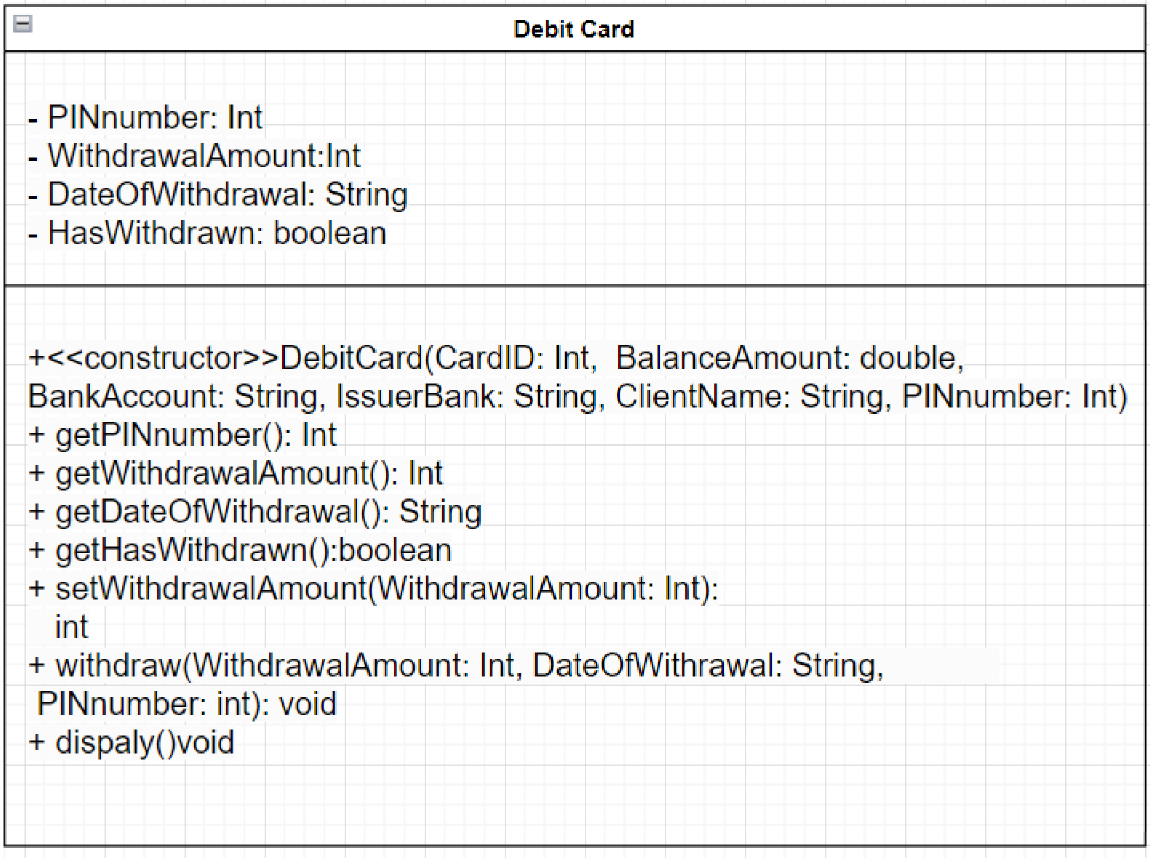
Bank Card

- ClientName: String
- CardID: Int
- BankAccount: String
- IssuerBank: String
- BalanceAmount: double

```
+ <<constructor>> BankCard(ClardID: int, BalanceAmount: double,  BankAccount: String, IssuerBank:
String)
+ getClientName():String
+ getCardID(): Int
+ getBankAccount():String
+ getIssuerBank(): String
+ getBalanceAmount(): double
+ setClientName(ClientName:String):
  String
+ setBalanceAmount(BalanceAmount: double):
  double
+ display(): void
```

FIGURE 1 CLASS DIAGRAM OF BANK CARD

2.2. CLASS DIAGRAM OF DEBIT CARD



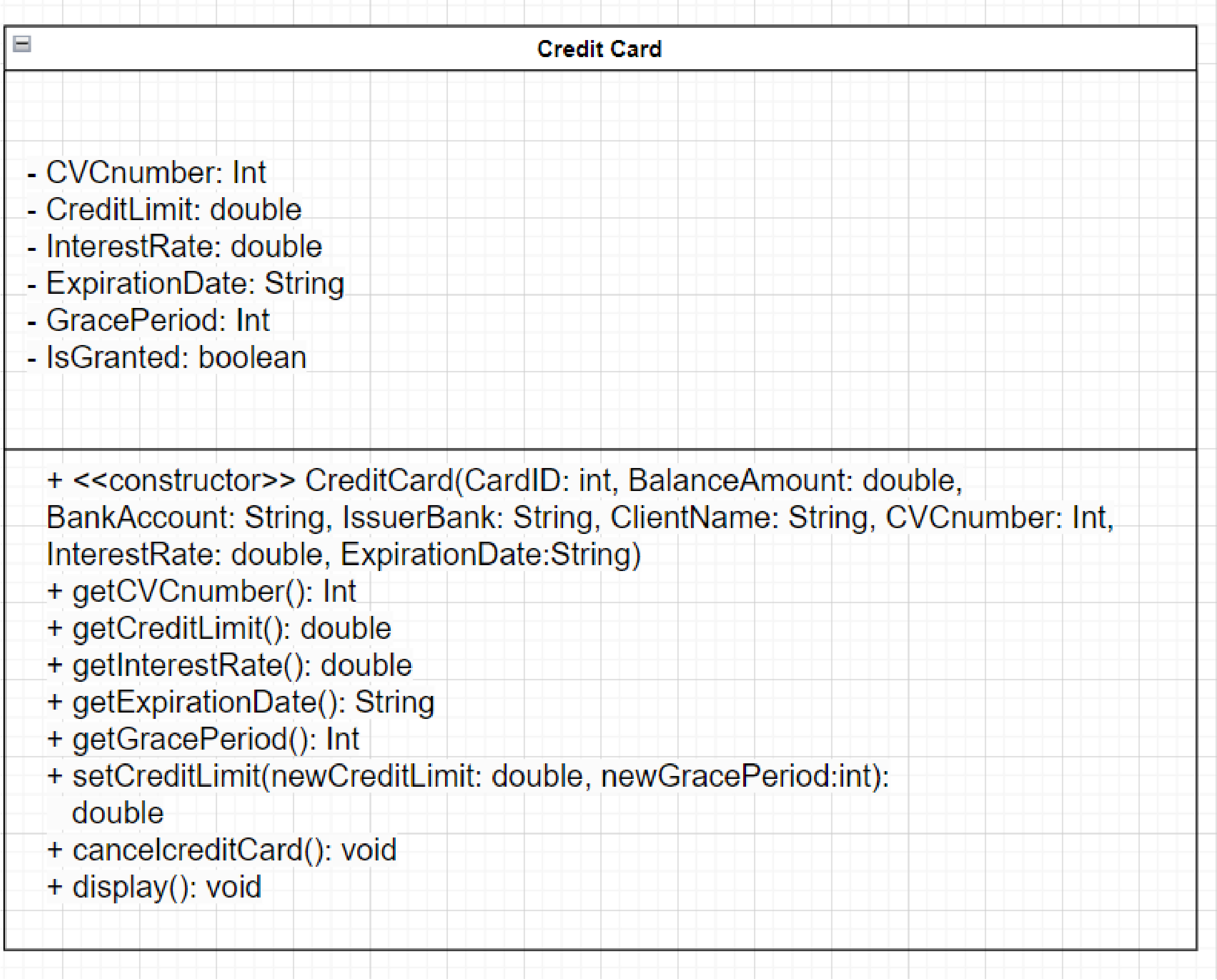
Debit Card

- PINnumber: Int
- WithdrawalAmount:Int
- DateOfWithdrawal: String
- HasWithdrawn: boolean

+<<constructor>>DebitCard(CardID: Int, BalanceAmount: double, BankAccount: String, IssuerBank: String, ClientName: String, PINnumber: Int)
+ getPINnumber(): Int
+ getWithdrawalAmount(): Int
+ getDateOfWithdrawal(): String
+ getHasWithdrawn():boolean
+ setWithdrawalAmount(WithdrawalAmount: Int): int
+ withdraw(WithdrawalAmount: Int, DateOfWithrawal: String, PINnumber: int): void
+ dispaly()void

FIGURE 2 CLASS DIAGRAM OF DEBIT CARD

2.3. CLASS DIAGRAM OF CREDIT CARD



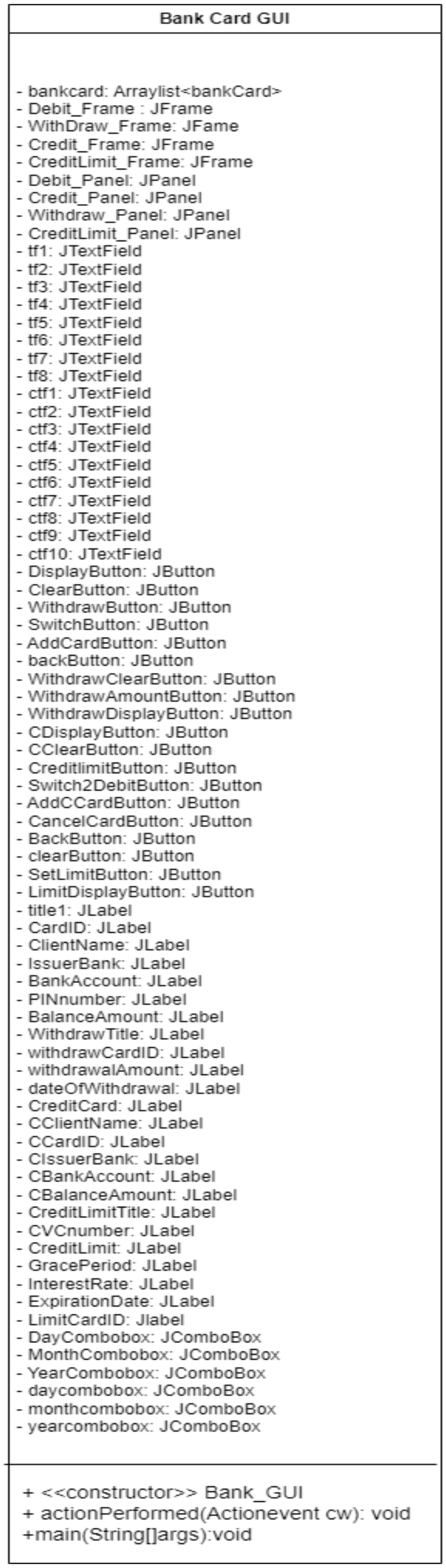
Credit Card

- CVCnumber: Int
- CreditLimit: double
- InterestRate: double
- ExpirationDate: String
- GracePeriod: Int
- IsGranted: boolean

+ <<constructor>> CreditCard(CardID: int, BalanceAmount: double, BankAccount: String, IssuerBank: String, ClientName: String, CVCnumber: Int, InterestRate: double, ExpirationDate:String)
+ getCVCnumber(): Int
+ getCreditLimit(): double
+ getInterestRate(): double
+ getExpirationDate(): String
+ getGracePeriod(): Int
+ setCreditLimit(newCreditLimit: double, newGracePeriod:int): double
+ cancelcreditCard(): void
+ display(): void

FIGURE 3CLASS DIAGRAM OF CREDIT CARD

2.4. CLASS DIAGRAM OF BANK CARD GUI



Bank Card GUI

- bankcard: Arraylist<bankCard>
- Debit_Frame : JFrame
- WithDraw_Frame: JFame
- Credit_Frame: JFrame
- CreditLimit_Frame: JFrame
- Debit_Panel: JPanel
- Credit_Panel: JPanel
- Withdraw_Panel: JPanel
- CreditLimit_Panel: JPanel
- tf1: JTextField
- tf2: JTextField
- tf3: JTextField
- tf4: JTextField
- tf5: JTextField
- tf6: JTextField
- tf7: JTextField
- tf8: JTextField
- ctf1: JTextField
- ctf2: JTextField
- ctf3: JTextField
- ctf4: JTextField
- ctf5: JTextField
- ctf6: JTextField
- ctf7: JTextField
- ctf8: JTextField
- ctf9: JTextField
- ctf10: JTextField
- DisplayButton: JButton
- ClearButton: JButton
- WithdrawButton: JButton
- SwitchButton: JButton
- AddCardButton: JButton
- backButton: JButton
- WithdrawClearButton: JButton
- WithdrawAmountButton: JButton
- WithdrawDisplayButton: JButton
- CDisplayButton: JButton
- CClearButton: JButton
- CreditlimitButton: JButton
- Switch2DebitButton: JButton
- AddCCardButton: JButton
- CancelCardButton: JButton
- BackButton: JButton
- clearButton: JButton
- SetLimitButton: JButton
- LimitDisplayButton: JButton
- title1: JLabel
- CardID: JLabel
- ClientName: JLabel
- IssuerBank: JLabel
- BankAccount: JLabel
- PINnumber: JLabel
- BalanceAmount: JLabel
- WithdrawTitle: JLabel
- withdrawCardID: JLabel
- withdrawalAmount: JLabel
- dateOfWithdrawal: JLabel
- CreditCard: JLabel
- CClientName: JLabel
- CCardID: JLabel
- CIssuerBank: JLabel
- CBankAccount: JLabel
- CBalanceAmount: JLabel
- CreditLimitTitle: JLabel
- CVCnumber: JLabel
- CreditLimit: JLabel
- GracePeriod: JLabel
- InterestRate: JLabel
- ExpirationDate: JLabel
- LimitCardID: Jlabel
- DayCombobox: JComboBox
- MonthCombobox: JComboBox
- YearCombobox: JComboBox
- daycombobox: JComboBox
- monthcombobox: JComboBox
- yearcombobox: JComboBox

+ <<constructor>> Bank_GUI
+ actionPerformed(Actionevent cw): void
+main(String[]args):void

FIGURE 4CLASS DIAGRAM OF BANK CARD GUI

2.5. CLASS DIAGRAM OF ALL CLASS CONNECTION

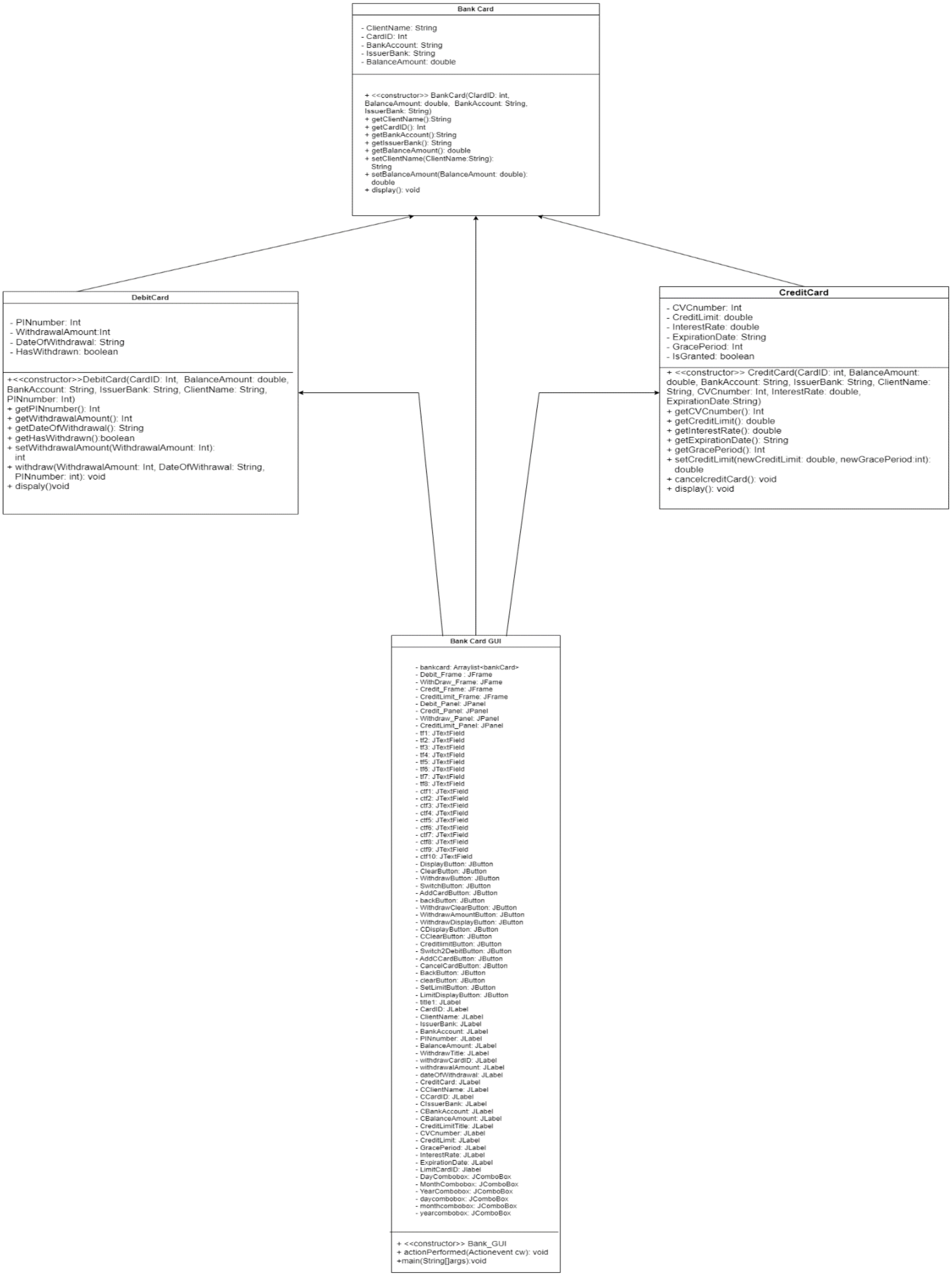

Bank Card
- ClientName: String
- CardID: Int
- BankAccount: String
- IssuerBank: String
- BalanceAmount: double
+ <<constructor>> BankCard(ClardID: int, BalanceAmount: double, BankAccount: String, IssuerBank: String)
+ getClientName():String
+ getCardID(): Int
+ getBankAccount():String
+ getIssuerBank(): String
+ getBalanceAmount(): double
+ setClientName(ClientName:String): String
+ setBalanceAmount(BalanceAmount: double): double
+ display(): void
DebitCard
- PINnumber: Int
- WithdrawalAmount:Int
- DateOfWithdrawal: String
- HasWithdrawn: boolean
+<<constructor>>DebitCard(CardID: Int, BalanceAmount: double, BankAccount: String, IssuerBank: String, ClientName: String, PINnumber: Int)
+ getPINnumber(): Int
+ getWithdrawalAmount(): Int
+ getDateOfWithdrawal(): String
+ getHasWithdrawn():boolean
+ setWithdrawalAmount(WithdrawalAmount: Int): int
+ withdraw(WithdrawalAmount: Int, DateOfWithrawal: String, PINnumber: int): void
+ dispaly()void
CreditCard
- CVCnumber: Int
- CreditLimit: double
- InterestRate: double
- ExpirationDate: String
- GracePeriod: Int
- IsGranted: boolean
+ <<constructor>> CreditCard(CardID: int, BalanceAmount: double, BankAccount: String, IssuerBank: String, ClientName: String, CVCnumber: Int, InterestRate: double, ExpirationDate:String)
+ getCVCnumber(): Int
+ getCreditLimit(): double
+ getInterestRate(): double
+ getExpirationDate(): String
+ getGracePeriod(): Int
+ setCreditLimit(newCreditLimit: double, newGracePeriod:int): double
+ cancelcreditCard(): void
+ display(): void
Bank Card GUI
- bankcard: Arraylist<bankCard>
- Debit_Frame : JFrame
- WithDraw_Frame: JFame
- Credit_Frame: JFrame
- CreditLimit_Frame: JFrame
- Debit_Panel: JPanel
- Credit_Panel: JPanel
- Withdraw_Panel: JPanel
- CreditLimit_Panel: JPanel
- tf1: JTextField
- tf2: JTextField
- tf3: JTextField
- tf4: JTextField
- tf5: JTextField
- tf6: JTextField
- tf7: JTextField
- tf8: JTextField
- ctf1: JTextField
- ctf2: JTextField
- ctf3: JTextField
- ctf4: JTextField
- ctf5: JTextField
- ctf6: JTextField
- ctf7: JTextField
- ctf8: JTextField
- ctf9: JTextField
- ctf10: JTextField
- DisplayButton: JButton
- ClearButton: JButton
- WithdrawButton: JButton
- SwitchButton: JButton
- AddCardButton: JButton
- backButton: JButton
- WithdrawClearButton: JButton
- WithdrawAmountButton: JButton
- WithdrawDisplayButton: JButton
- CDisplayButton: JButton
- CClearButton: JButton
- CreditlimitButton: JButton
- Switch2DebitButton: JButton
- AddCCardButton: JButton
- CancelCardButton: JButton
- BackButton: JButton
- clearButton: JButton
- SetLimitButton: JButton
- LimitDisplayButton: JButton
- title1: JLabel
- CardID: JLabel
- ClientName: JLabel
- IssuerBank: JLabel
- BankAccount: JLabel
- PINnumber: JLabel
- BalanceAmount: JLabel
- WithdrawTitle: JLabel
- withdrawCardID: JLabel
- withdrawalAmount: JLabel
- dateOfWithdrawal: JLabel
- CreditCard: JLabel
- CClientName: JLabel
- CCardID: JLabel
- CIssuerBank: JLabel
- CBankAccount: JLabel
- CBalanceAmount: JLabel
- CreditLimitTitle: JLabel
- CVCnumber: JLabel
- CreditLimit: JLabel
- GracePeriod: JLabel
- InterestRate: JLabel
- ExpirationDate: JLabel
- LimitCardID: Jlabel
- DayCombobox: JComboBox
- MonthCombobox: JComboBox
- YearCombobox: JComboBox
- daycombobox: JComboBox
- monthcombobox: JComboBox
- yearcombobox: JComboBox
+ <<constructor>> Bank_GUI
+ actionPerformed(Actionevent cw): void
+main(String[]args):void

FIGURE 5 CLASS DIAGRAM OF ALL CLASS CONNECTIONS

3.PSEUDOCODE

3.1. PSEUDOCODE OF BANK CARD GUI

```
CREATE a class Bank_GUI which implements ActionListener
DO
    DECLARE bankcard as Arraylist of BankCard as private
    DECLARE Debit_Frame, WithDraw_Frame as JFrame as private
    DECLARE Credit_Frame, CreditLimit_Frame as JFrame as private
    DECLARE Debit_Panel, CreditPanel as JPanel as private
    DECLARE Withdraw_Panel, Credit_Panel as JPanel as private
    DECLARE tf1, tf2, tf3, tf4, tf5, tf6, tf7, tf6 as JTextField as private
    DECLARE ctf1, ctf2, ctf3, ctf4, ctf5, ctf6, ctf7, ctf8, ctf9, ctf10 as JTextfield as private
    DECLARE DisplayButton, ClearButton, WithdrawButton, SwitchButton,
            AddCardbutton, backbutton, WithdrawClearButton, WithdrawAmountButton.
            WithdrawDisplaybutton as JButton as private
    DECLARE CDisplayButton,CClearButton ,CreditlimitButton , Switch2DebitButton,
            AddCCardButton, CancelCardButton, BackButton, clearButton, SetLimitButton,
            LimitDisplayButton as JButton as private
    DECLARE title1, CardID, ClientName, IssuerBank, BankAccount, PINnumber,
             BalanceAmount,WithdrawTitle,WithdrawCardId, withdrawalAmount,
             dateOfWithdrawal as JLabel as private
    DECLARE CreditCard,CClientName , CCardID, CIssuerBank,CBankAccount ,
             CBalanceAmount, CreditLimitTitle, CVCnumber, CreditLimit, GracePeriod,
             InterestRate ,ExpirationDate , LimitCardId as JLabel as private
```

```
DECLARE DayCombobox, MonthCombobox, YearCombobox as JComboBox as private
DECLARE daycombobox, monthcombobox, yearcombobox, as JcomboBox as private

CREATE a constructor bank_GUI
DO
    CREATE new JFrame (“Rounak_Joshi_22067875”) named Debit_frame
    CREATE new JPanel named Debit_Panel

    CREATE new JTextfield named tf1
    CREATE new JTextfield named tf2
    CREATE new JTextfield named tf3
    CREATE new JTextfield named tf4
    CREATE new JTextfield named tf5
    CREATE new JTextfield named tf6

    CREATE new JButton(“Display”) named DisplayButton
    CREATE new JButton(“Clear”) named ClearButton
    CREATE new JButton (“Withdraw Amount”) named Withdrawbutton
    CREATE new JButton (“Switch to Credit Card”) named SwitchButton
    CREATE new JButton (“Add Debit Card”) named AddCardButton

    CREATE new JLabel (“Debit Card”) named title1
    CREATE new JLabel (“Card ID”) named CardID
    CREATE new JLabel (“ClientName”) named ClientName
    CREATE new JLabel (“Issuer Bank”) named IssuerBank
    CREATE new JLabel (“Bank account”) named BankAccount
    CREATE new JLabel (“PIN Number”) named PINnumber
    CREATE new JLabel (“Balance Amount”) named BalanceAmount

    SET bounds for the JLabel title1
    SET bounds for the JLabel CardID
    SET bounds for the JLabel ClientName
    SET bounds for the JLabel PINnumber
    SET bounds for the JLabel BalanceAmount
    SET bounds for the JLabel BankAccount
    SET bounds for the JLabel IssuerBank

    SET bounds for the JTextfield tf1
    SET bounds for the JTextfield tf2
    SET bounds for the JTextfield tf3
    SET bounds for the JTextfield tf4
    SET bounds for the JTextfield tf5
    SET bounds for the JTextfield tf6

    SET bounds for the JButton DisplayButton
    SET bounds for the JButton ClearButton
    SET bounds for the JButton WithdrawaButton
    SET bounds for the JButton SwitchButton
    SET bounds for the JButton AddCardButton

    ADD JLabel title1 to Debit_Panel
    ADD JLabel CardID to Debit_Panel
    ADD JLabel ClientName to Debit_Panel
    ADD JLabel IssuerBank to Debit_Panel
    ADD JLabel BankAccount to Debit_Panel
    ADD JLabel PINnumber to Debit_Panel
    ADD JLabel BalanceAmount to Debit_Panel

    ADD JTextfield tf1 to Debit_Panel
    ADD JTextfield tf2 to Debit_Panel
    ADD JTextfield tf3 to Debit_Panel
    ADD JTextfield tf4 to Debit_Panel
    ADD JTextfield tf5 to Debit_Panel
    ADD JTextfield tf6 to Debit_Panel

    ADD JButton DisplayButton to Debit_Panel
    ADD JButton ClearButton to Debit_Panel
    ADD JButton WithdrawButton to Debit_Panel
    ADD JButton SwitchButton to Debit_Panel
    ADD JButton AddCardButton to Debit_Panel

    ADD JButton DisplayButton to ActionListener
    ADD JButton ClearButton to ActionListener
    ADD JButton WithdrawButton to ActionListener
    ADD JButton SwitchButton to ActionListener
    ADD JButton AddCardButton to ActionListener

    SET font of Label title1 to new Font (“Arial Black”)
    SET background of Debit_Panel to new color
    SET background of DisplayButton to new color
    SET background of ClearButton to new color
    SET background of WithdrawButton to new color
    SET background of SwitchButton to new color
    SET background of AddCardbutton to new color

    ADD Debit_Frame to the Debit_Panel
    SET size of Debit_Frame (width=1000, height=400)
    SET layout of DebitPanel to null
    SET visible of Debit_Frame to true
    SET resizable of Debit_Frame to false
    SET Debit_Frame to DefaultCloseOperation (Debit_Frame.EXIT_ON_CLOSE)

    CREATE new JFrame (“Rounak_Joshi_22067875”) named WithDaw_Frame
    CREATE new JPanel named Withdraw-Panel

    CREATE new JLabel (“Withdraw Amount”) named WithdrawTitle
    CREATE new JLabel (“Card ID”) named WithdrawCardID
    CREATE new JLabel (“Balance Amount”) named WithdrawalAmount
    CREATE new JLabel (“Date of Withdrawal”) named dateOfWithdrawal

    CREATE new JTextfield named tf7
    CREATE new JTextfield named tf8

    CREATE new JButton(“back”) named backButton
    CREATE new JButton(“Clear”) named WithdrawClearButton
    CREATE new JButton (“Withdraw Amount”) named WithdrawAmountbutton
    CREATE new JButton (“Display”) named WithdrawDisplayButton
    CREATE array of string named DD with elements ("1","2","3","4","5","6","7",
            "8","9","10","11","12","13","14","15","16","17","18","19","20","21",
            "22","23","24","25","26","27","28","29","30","31)
    CREATE new JComboBox named DayCombobox with “DD” as a parameter
    CREATE array of string named MM with elements ("January","february","March",
            "April","May","June","July","August","September","October",
            "November","December")
    CREATE new JComboBox named MonthCombobox with “MM” as a parameter
    CREATE array of string named YYYY with element (2023”)
    CREATE new JComboBox named YearCombobox with “YYYY” as a parameter


    SET bounds for the JLabel WithdrawTitle
    SET bounds for the JLabel WithdrawCardID
    SET bounds for the JLabel withdrawalAmount
    SET bounds for the JLabel DateOfWithdrawal

    SET bounds for the JTextfield tf7
    SET bounds for the JTextfield tf8

    SET bounds for the JButton backButton
    SET bounds for the JButton WithdrawClearButton
    SET bounds for the JButton WithdrawAmountButton
    SET bounds for the JButton WithdrawDisplayButton

    SET bounds for the JComboBox DayCombobox
    SET bounds for the JComboBox MonthCombobox
    SET bounds for the JComboBox YearCombobox

    ADD JLabel WithdrawTitle to Withdraw_Panel
    ADD JLabel WithdrawCardID to Withdraw_Panel
    ADD JLabel withdrawAmount to Withdraw_Panel
    ADD JLabel dateOfWithdrawal to Withdraw_Panel

    ADD JButton backButton to Withdraw_Panel
    ADD JButton WithdrawClearButton to Withdraw_Panel
    ADD JButton WithdrawAmountButton to Withdraw_Panel
    ADD JButton WithdrawDisplaybutton to Withdraw_Panel

    ADD JComboBox Dayombobox to Withdraw_Panel
    ADD JComboBox MonthCombobox to Withdraw_Panel
    ADD JComboBox YearCombobox to Withdraw_Panel

    ADD JTextField tf7 to Withdraw_Panel
    ADD JTextField tf8 to Withdraw_Panel

    ADD JButton backButton to ActionListener
    ADD JButton WithdrawClearButton to ActionListener
    ADD JButton WithdrawAmountButton to ActionListener
    ADD JButton WithdrawDisplayButton to ActionListener

    SET font of JLabel WithdrawTitle to new Font (“Arial Black”)
    SET background of Withdraw_Panel to new color
    SET background of backButton to new color
    SET background of WithdrawClearButton to new color
    SET background of WithdrawAmountButton to new color
    SET background of WithdrawDisplayButton to new color

    ADD WithDraw_Frame to the Withdraw_Panel
    SET size of WithDraw_Frame (width=500, height=400)
    SET layout of Withdraw_Panel to null
    SET visible of WithDraw_Frame to false
    SET resizable of WithDraw_Frame to false
    SET Withdraw_Frame to DefaultCloseOperation
    (WitDraw_Frame.EXIT_ON_CLOSE)

    CREATE new JFrame (“Rounak_Joshi_22067875”) named Credit_Frame
    CREATE new JPanel named Credit_Panel

    CREATE new JTextfield named ctf1
    CREATE new JTextfield named ctf2
    CREATE new JTextfield named ctf3
    CREATE new JTextfield named ctf4
    CREATE new JTextfield named ctf5
    CREATE new JTextfield named ctf6
    CREATE new JTextfield named ctf7


    CREATE new JButton(“Display”) named CDisplayButton
    CREATE new JButton(“Clear”) named CClearButton
    CREATE new JButton (“Set Credit Limit”) named CreditLimitButton
    CREATE new JButton (“Switch to debit Card”) named Switch2DebitButton
    CREATE new JButton (“Add Credit Card”) named AddCCardButton
    CREATE new JButton (“Cancel Credit Card”) named CancelCardButton


    CREATE new JLabel (“Credit Card”) named CreditCard
    CREATE new JLabel (“Client name”) named CClientname
    CREATE new JLabel (“Card ID”) named CCardID
    CREATE new JLabel (“Issuer Bank”) named CIssuerBank
    CREATE new JLabel (“Bank account”) named CBankAccount
    CREATE new JLabel (“Balance Amount”) named CBalanceAmount
    CREATE new JLabel (“CVC Number”) named CVCnumber
    CREATE new JLabel (“Interest Rate”) named InterestRate
    CREATE new JLabel (“Expiration Date”) named ExpirationDate


    CREATE array of string named Day with elements ("1","2","3","4","5","6","7",
            "8","9","10","11","12","13","14","15","16","17","18","19","20","21",
            "22","23","24","25","26","27","28","29","30","31)
    CREATE new JCombobox named daycombobox with “Day” as a parameter
    CREATE array of string named Month with elements ("January","february",
             "March", "April","May","June","July","August","September",
             "October","November","December")
```

```
CREATE new JComboBox named monthcombobox with “Month” as a parameter
CREATE array of string named “Year” with elements (“2023”, "2024","2025",
        "2026","2027")
CREATE new JComboBox named yearcombobox with “Year” as a parameter

SET bounds for the JLabel CreditCard
SET bounds for the JLabel CClientName
SET bounds for the JLabel CCardID
SET bounds for the JLabel CIssuerBank
SET bounds for the JLabel InterestRate
SET bounds for the JLabel BankAccount
SET bounds for the JLabel CVCnumber
SET bounds for the JLabel BalanceAmount
SET bounds for the JLabel ExpirationDate

SET bounds for the JTextfield ctf1
SET bounds for the JTextfield ctf2
SET bounds for the JTextfield ctf3
SET bounds for the JTextfield ctf4
SET bounds for the JTextfield ctf5
SET bounds for the JTextfield ctf6
SET bounds for the JTextfield ctf7

SET bounds for the JButton CDisplayButton
SET bounds for the JButton CClearButton
SET bounds for the JButton CreditLimitButton
SET bounds for the JButton Switch2DebitButton
SET bounds for the JButton AddCCardButton

SET bounds for the JComboBox daycombobox
SET bounds for the JComboBox monthcombobox
SET bounds for the JComboBox yearcombobox

ADD JLabel CreditCard to Credit_Panel
ADD JLabel CClientName to Credit _Panel
ADD JLabel CCardID to Credit_Panel
ADD JLabel CIssuerBank to Credit_Panel
ADD JLabel CBankAccount to Credit_Panel
ADD JLabel CVCnumber to Credit_Panel
ADD JLabel CBalanceAmount to Credit_Panel
ADD JLabel InterestRate to Credit_Panel
ADD JLabel ExpirationDate to Credit_Panel
ADD JTextfield ctf1 to Credit_Panel
ADD JTextfield ctf2 to Credit_Panel
ADD JTextfield ctf3 to Credit_Panel
ADD JTextfield ctf4 to Credit_Panel
ADD JTextfield ctf5 to Credit_Panel
ADD JTextfield ctf6 to Credit_Panel
ADD JTextfield ctf7 to Credit_Panel
ADD JButton CDisplayButton to Credit_Panel
ADD JButton CClearButton to Credit_Panel
ADD JButton CreditlimitButton to Credit_Panel
ADD JButton Switch2DebitButton to Credit_Panel
ADD JButton Switch2DebitButton to Credit_Panel
ADD JButton CancelCardButton to Credit_Panel
ADD JComboBox daycomboboxto Credit_Panel
ADD JComboBox monthcombobox to Credit_Panel
ADD JComboBox yearcombobox to Credit_Panel

ADD JButton CDisplayButton to ActionListener
ADD JButton CClearButton to ActionListener
ADD JButton CreditlimitButton to ActionListener
ADD JButton Switch2DebitButton to ActionListener
ADD JButton AddCCardButton to ActionListener
ADD JButton CancelCardButton to ActionListener

SET font of JLabel CreditCard to new Font (“Arial Black”)
SET background of Credit_Panel to new color
SET background of CDisplayButton to new color
SET background of CClearButton to new color
SET background of CreditlimitButton to new color
SET background of Switch2DebitButton to new color
SET background of AddCCardbutton to new color
SET background of CancelCardButton to new color

ADD Credit_Frame to the Credit_Panel
SET size of Credit_Frame (width=1000, height=500)
SET layout of Credit_Panel to null
SET visible of Credit_Frame to false
SET resizable of Credit_Frame to false
SET Credit_Frame to DefaultCloseOperation (Credit_Frame.EXIT_ON_CLOSE)


CREATE new JFrame ("Rounak_Joshi_22067875") named CreditLimit_Frame
CREATE new JPanel named CreditLimit_Panel

CREATE new JLabel ("Set Credit limit") named CreditLimitTitle
CREATE new JLabel ("Credit Limit") named CreditLimit
CREATE new JLabel ("Grace Period") named GracePeriod
CREATE new JLabel ("Card ID") named LimitCardID


CREATE new JTextfield named ctf8
CREATE new JTextfield named ctf9
CREATE new JTextfield named ctf10

CREATE new JButton("Back") named BackButton
CREATE new JButton("Clear") named clearButton
CREATE new JButton ("Set Credit Limit") named SetLimitButton
CREATE new JButton ("Display") named LimitDisplayButton

SET bounds for the JLabel CreditLimitTitle
SET bounds for the JLabel CreditLimit
SET bounds for the JLabel GracePeriod
SET bounds for the JLabel LimitCardID

SET bounds for the JTextfield ctf8
SET bounds for the JTextfield ctf9
SET bounds for the JTextfield ctf10

SET bounds for the JButton BackButton
SET bounds for the JButton clearButton
SET bounds for the JButton SetLimitButton
SET bounds for the JButton LimitDisplayButton

ADD JLabel CreditLimitTitle to CreditLimit_Panel
ADD JLabel CreditLimit to CreditLimit_Panel
ADD JLabel GracePeriod to CreditLimit_Panel
ADD JLabel GracePeriod to CreditLimit_Panel

ADD JTextfield ctf8 to CreditLimit_Panel
ADD JTextfield ctf9 to CreditLimit_Panel
ADD JTextfield ctf10 to CreditLimit_Panel

ADD JButton BackButtonto CreditLimit_Panel
ADD JButton clearButtonto CreditLimit_Panel
ADD JButton SetLimitButtonto CreditLimit_Panel
ADD JButton LimitDisplayButtonto CreditLimit_Panel

ADD JButton BackButton to ActionListener
```

```
        ADD JButton clearButton to ActionListener
        ADD JButton SetLimitButton to ActionListener
        ADD JButton LimitDisplayButton to ActionListener

        SET font of JLabel CreditLimitTitle to new Font (“Arial Black”)
        SET background of CreditLimit_Panel to new color
        SET background of BackButton to new color
        SET background of clearButton to new color
        SET background of SetLimitButton to new color
        SET background of LimitDisplayButton to new color

        ADD CreditLimit_Frame to the CreditLimit_Panel
        SET size of CreditLimit_Frame (width=500, height=400)
        SET layout of CreditLimit_Panel to null
        SET visible of CreditLimit_Frame to false
        SET resizable of CreditLimit_Frame to false
        SET CreditLimit_Frame to DefaultCloseOperation
        (CreditLimit_Frame.EXIT_ON_CLOSE)
    END DO

    CREATE a method actionPerformed with the parameter with the parameter ActionEvent cw
    DO
        IF DisplayButton is clicked
        DO
            FOR each bankcard in Bankcard list
            DO
                IF bankCard is an instance of DebitCard
                DO
                    CONVERT bankCard to DebitCard object
                    CALL display method of the debitcard object
                END DO
            END DO
        END DO

        IF AddCardButton is clicked
        DO
            TRY
            DO
                ASSIGN CardID to text of tf1 which is converted to integer
                ASSIGN ClientName to text of tf2
                ASSIGN PINnumber to text of tf3 which is converted integer
                ASSIGN BalanceAmount to text of tf4 which is converted to double
                ASSIGN BankAccount to text of tf5
                ASSIGN IssuerBank to text of tf6
                IF CardID is less than or equal to 0, or ClientName is an empty string,
                or IssuerBank is an empty string, or BankAccount is an empty string,
                or PINnumber is less than or equal to 0, or BalanceAmount is less
                than 0
                DO
                    DISPLAY an error message
                END DO
```

```
                ELSE
                DO
                    FOR each bankCard in bankcard
                    DO
                        IF bankCard is instance of DebitCard and its CardID
                        matches CardID
                        DO
                            DISPLAY an error message
                            RETURN
                        END DO
                    END DO
                    CREATE new DebitCard object with CardID, BalanceAmount,
                    BankAccount, IssuerBank, ClientName and PINnumber as its
                    Parameters
                    ADD new DebitCard object to bankcard
                    DISPLAY a successful message
                END DO
            END DO
            CATCH number format exception
            DO
                DISPLAY an error message
            END DO
        END DO

        IF ClearButton is clicked
        DO
        SET text of tf1 to empty
        SET text of tf2 to empty
        SET text of tf3 to empty
        SET text of tf4 to empty
        SET text of tf5 to empty
        SET text of tf6 to empty
END DO

IF WithdrawButton is clicked
DO
        SET visible of WithDraw-Frame to true
        SET visible of Debit_frame to false
END DO

IF WithdrawAmountButton is clicked
DO
        TRY
        DO
                ASSIGN CardID to text of tf1 which is converted to integer
                ASSIGN withdrawalAmount to text of tf8 which is converted to
                integer
                ASSIGN BalanceAmount to text of tf4 which is converted to double
                ASSIGN dateOfWithdrawal as a string which is a combination of Day,
                Month and Year combobox
                IF pinnumber matches the card PINnumber
                SET cardFound as Boolean as false
```

```
DO
    FOR each bankCard in bankcard
    DO
        IF bankCard is an instance of DebitCard
        DO
            CONVERT bankCard to DebitCard object
            IF Debit cards Card ID matches CardId
            DO
                SET cardFound to true
                ASSIGN the pinnumber as an integer and
                display input dialog box
                IF pin number matches debit cards pin
                Number
                DO
                    IF withdrawal amount is greater than
                    current BalanceAmount
                    DO
                        DISPLAY an error message
                        RETURN
                    END DO
                WITHDRAW amount using DebitCard's
                withdraw method
                DISPLAY a successful message
                END DO
                ELSE
                DO
```

```
                            DISPLAY an error message
                        END DO
                    END DO
                END DO
            END DO
            IF there is no card found
            DO
                DISPLAY an error message
            END DO
        END DO
        CATCH number format exception
        DO
            DISPLAY an error message
        END DO
    END DO
END DO

IF backButton is clicked
DO
    SET visible of WithDraw_Frame to false
    SET visible of Debit_Frame to true
END DO

IF WithdrawClearButton is clicked
DO
    SET text of tf7 to empty
```

```
        SET text of tf8 to empty
        SET selected index of DayCombobox to 0
        SET selected index of MonthCombobox to 0
        SET selected index of YearCombobox to 0
    END DO
    IF WithdrawDisplayButton is clicked
    DO
        FOR each bankcard in Bankcard list
        DO
            IF bankCard is an instance of DebitCard
            DO
                CONVERT bankCard to DebitCard object
                CALL display method of the debitcard object
            END DO
        END DO
    END DO

    IF SwitchButton is clicked
    DO
        SET visible of Credit_Frame to true
        SET visible of Debit_Frame to false
    END DO

    IF CDisplayButton is clicked
    DO
        FOR each bankcard in Bankcard list
        DO
            IF bankCard is an instance of CreditCard
            DO
                CONVERT bankCard to CreditCard object
                CALL display method of the creditcard object
            END DO
        END DO
    END DO

    IF AddCCardButton is clicked
    DO
        TRY
        DO
            ASSIGN CClientName to text of ctf1
            ASSIGN CCardID to text of ctf2 which is converted to integer
            ASSIGN CIssuerBank to text of ctf3
            ASSIGN InterestRate to text of ctf4 which is converted to double
            ASSIGN CBankAccount to text of ctf5
            ASSIGN CVCnumber to text of ctf6 which is converted to integer
            ASSIGN CBalanceAmount to text of ctf7 which is converted to double
            ASSIGN ExpirationDate as a string which is a combination of day,
            month and year combobox
            IF CCardID is less than or equal to 0, or CClientName is an empty
            string, or CIssuerBank is an empty string, or CBankAccount is an
            empty string, or is less than or equal to 0, or CBalanceAmount is less
            than 0, InterestRate is less than or equal to 0, or CVCnumber is less
            than or equal to 0
```

```
                DO
                    DISPLAY an error message
                END DO
                ELSE
                DO
                    FOR each bankCard in bankcard
                    DO
                        IF bankCard is instance of cREDbitCard and its CardID
                        matches CCardID
                        DO
                            DISPLAY an error message
                            RETURN
                        END DO
                    END DO
                    CREATE new CreditCard object with CCardID,
                    CBalanceAmount,CBankAccount, CIssuerBank, CClientName ,
                    InterestRate and ExpirationRate as its Parameters
                    ADD new CreditCard object to bankcard
                    DISPLAY a successful message
                END DO
            END DO
            CATCH number format exception
            DO
                DISPLAY an error message
            END DO
        END DO
```

```
IF CClearButton is clicked
DO

    SET text of tf1 to empty
    SET text of ctf2 to empty
    SET text of ctf3 to empty
    SET text of ctf4 to empty
    SET text of ctf5 to empty
    SET text of ctf6 to empty
    SET text of ctf7 to empty
    SET selected index of daycombobox to 0
    SET selected index of monthcombobox to 0
    SET selected index of yearcombobox to 0
END DO

IF Switch2DebitButton is clicked
DO
    SET visible of Credit_Frame to false
    SET visible of Debit_Frame to true
END DO

IF CreditkimitButton is clicked
DO
    SET Credit_Frame to false
    SET Creditlimit_Frame to true
```

```
END DO


IF SetLimitButton is clicked
DO
    TRY
    DO
        ASSIGN CCardID to text of ctf8
        ASSIGN CreditLimit to text of ctf9 which is converted to double
        ASSIGN GracePeriod to text of ctf10 which is converted to integer
        SET IsGranted as Boolean to false
        FOR each bankCard in bankcard
        DO
            IF bankCard is an instance of CreditCard
            DO
                CONVERT bankCard to CreditCard object
                IF CardId matches Credit_Card's CardID
                DO
                    SET Credit limit and grace period to creditcard
                    using CreditCards Set_Creditlimit method
                    SET isGranted to true
                    IF CreditLimit amount is greater than current
                    DISPLAY an error message
                    RETURN
                END DO
                ELSE
                DO
```

```
                    DISPLAY a successful message
                END DO
            END DO
        END DO
    END DO
    IF the card id does not exist
    DO
        DISPLAY an error message
    END DO
    CATCH number format exception
    DO
        DISPLAY an error message
    END DO
END DO


IF CancelCardButton is clicked
DO
    ASSIGN CCardID to text of ctf2 which is converted to integer
    FOR each bankCard in bankcard
    DO
        IF bankCard is an instance of CreditCard
        DO
            CONVERT bankCard to CreditCard object
            IF CardId matches Credit_Card's CardID
            DO
```

```
                    CANCEL credit card using the CreditCard’s
                    cancelCreditCard method
                    DISPLAY a successful message
                END DO
            END DO
        END DO
    END DO

    IF clearButton is clicked
    DO
        SET text of ctf8 to empty
        SET text of ctf9 to empty
    END DO

    IF BackButton is clicked
    DO
        SET visible of Credit_Frame to true
        SET visible of CreditLimit_Frame to false
    END DO

    IF LimitDisplayButton is clicked
    DO
        FOR each bankcard in Bankcard list
        DO
            IF bankCard is an instance of CreditCard
            DO
```

```
                    CONVERT bankCard to CreditCard object
                    CALL display method of the creditcard object
                END DO
            END DO
        END DO
    END DO
    CREATE a main method
    DO
        CREATE a new instance of Bank_GUI class
    END DO
END DO
```

4. METHOD DESCRIPTION

4.1. METHOD DESCRIPTION OF BANK_GUI

public class Bank_GUI implements ActionListener-

The code is a java class named Bank_GUI that implements the ActionListener. Within this class contains a constructor that is used to create a graphical user interface GUI for the Debit card and credit card of the bank. Within this class various variables such as JFrame, JPanel, JLabel, JTextfield, JComboBox have been declared as private and a new Arraylist has been created for the bankcard class. The class also contains actionperformed method which is used in order for Sthe various JButtons added and created to work in the GUI.

public Bank_GUI ()-

The following construction is created and is used to initialize the GUI components such as JTextfields, JFrame, JPanel, JComboBox etc. Within the constructor 4 different frames are created with each consisting its own panel. The frames contain various labels, textfield and buttons which have been assorted using set bounds in the frame. After that all components are added to their own respective panels. Set background color is used to assign various background color to the frame as well as the button. Set font is also used for assigning various labels different fonts and text size. The buttons have then been added to the actionlistener in order for them to work. The frame is then added to the panel along with setting the size of the frame along with panel layout being set to null. The frame visibility is set either to true or false as per the requirement and the frame resizable has been set to false. At last frames have been set to default close operation.

public void actionPerformed(ActionEvent cw)-

The method is used for the buttons to be able to react as per their requirement. The method allows the buttons that have been added to the actionperformer to perform their actions as per their needed to.

The method description for each button is given below

1. **Display button**

The following button handles the event when the button is clicked. First the method loops through the list of the bank card and for each debit card added to the list, if the bankcard is an instance of Debit card it calls the DebitCard object in which the display method is called from the debit card class.

2. AddCardButton

the following button handles the event when the button is clicked. First the method creates a try case. Within the case the method converts the user input to the appropriate type such into integer, double or string as mentioned. Then if CardID or PINnumber or balanceamount is less than or equal to 0 or clientname or issuerbank or bankaccount ie empty, it shows an error message through joptionpane. Else it loops through the list of the bank card and for each debit card added to the list, if the bankcard is an instance of Debit card and the bankcard's cardid matches cardid it shows an error message through joptionpane and returns to the beginning, else a new debitcard object is created and then add to the bankcard Arraylist and shows a successful message through joptionpane. If there is number format exception, it shows an error message through joptionpane and it found through catch

3. ClearButton

The following button handles the event when the button is clicked. When the button is clicked it clears the textfields of tf1, tf2, tf3, tf4, tf5, tf6, tf7 and sets the text to empty

4. WithdrawButton

The following button handles the event when the button is clicked. when the button is clicked, it sets the visibility of WithDraw frame to (true) and the visibility of debit frame to (false).

5. WithdrawAmountButton

The following button handles the event when the button is clicked. First the method creates a try case. Within the case the method converts the user input to the appropriate type such into integer, double or string as mentioned as well as gets combobox as called and sets card found to false. Then the method loops through the list of the bank card and for each debit card added to the list, if the bankcard is an instance of Debit card it calls the DebitCard object. If the debit cards CardID matches the CardID, it sets cardfound to true and assigns pinnumber to integer ad displays and input dialog box. Then If the pin number matches the debitcard pin number, within that, if the withdrawal amount is greater than the current balance amount, it shows and error message through joptionpane and returns to the beginning but else it calls the withdraw method from the debitcard class and shows as successful message through joptionpane. Otherwise it shows an error message through joptionpane. If the card is not found it displays an error message through joptionpane. If there is number format exception, it shows an error message through joptionpane and it found through catch. if the debitcard's pin number matches the pin number

6. **backButton**

The following button handles the event when the button is clicked. when the button is clicked, it sets the visibility of debit frame to (true) and the visibility of WithDraw frame to (false).

7. **WithdrawClearButton**

The following button handles the event when the button is clicked. When the button is clicked it clears the textfields of tf7 and tf8 and sets the text to empty as well as sets the index of DayCombobox, MonthCombobox and YearCombobox to 0

8. **WithdrawDisplayButton**

The following button handles the event when the button is clicked. First the method loops through the list of the bank card and for each debit card added to the list, if the bankcard is an instance of Debit card it calls the DebitCard object in which the display method is called from the debit card class.

9. **SwitchButton**

The following button handles the event when the button is clicked. when the button is clicked, it sets the visibility of Credit frame to (true) and the visibility of Debit frame to (false).

10. **CDisplayButton**

The following button handles the event when the button is clicked. First the method loops through the list of the bank card and for each credit card added to the list, if the bankcard is an instance of Credit card it calls the Credit card object in which the display method is called from the credit card class

11. **AddCCardButton**

the following button handles the event when the button is clicked. First the method creates a try case. Within the case the method converts the user input to the appropriate type such into integer, double or string as mentioned as well as gets combobox as called. Then if CardID or InterestRate or CVCnumber or balanceamount is less than or equal to 0 or clientname or issuerbank or bankaccount ie empty, it shows an error message through joptionpane. Else it loops through the list of the bank card and for each credit card added to the list, if the bankcard is an instance of Credit card and the bankcard's cardid matches ccardid it shows an error message through joptionpane and returns to the beginning, else a new credit card object is created and then add to the bankcard Arraylist and shows a successful message through joptionpane. If there is number format exception, it shows an error message through joptionpane and it found through catch

12. CClearButton

The following button handles the event when the button is clicked. When the button is clicked it clears the textfields of ctf1, ctf2, ctf3, ctf4, ctf5, tf6 and tf7 and sets the text to empty as well as sets the index of daycombobox, monthcombobox and yearcombobox to 0

13. Switch2DebitButton

The following button handles the event when the button is clicked. when the button is clicked, it sets the visibility of Debit frame to (true) and the visibility of Credit frame to (false).

14. CreditlimitButton

The following button handles the event when the button is clicked. when the button is clicked, it sets the visibility of CreditLimit frame to (true) and the visibility of Credit frame to (false).

15. SetLimitButton

The following button handles the event when the button is clicked. First the method creates a try case. Within the case the method converts the user input to the appropriate type such into integer, double or string as mentioned and sets is granted to false. Then the method loops through the list of the bank card and for each credit card added to the list, if the bankcard is an instance of Credit card it calls the CreditCard object. If the credit cards CardID matches the CardID it calls the set_CreditLimit method from the Crditcard class and sets isgranted to true. If the credit limit amount is greater than the current balance amount, it shows and error message through joptionpane and returns to the beginning otherwise shows as successful message through joptionpane.If the given card is not there it displays an error message through joptionpane. If there is number format exception, it shows an error message through joptionpane and it found through catch.

16. CancelCardButton

The following button handles the event when the button is clicked. First the method converts the user CCardID input into integer as mentioned. Then the method loops through the list of the bank card and for each credit card added to the list, if the bankcard is an instance of Credit card it calls the CreditCard object. if the credit cards CardID matches the CardID, it calls the cancelCreditLimit method from the Crditcard class and shows as successful message through joptionpane.

17. clearButton

The following button handles the event when the button is clicked. When the button is clicked it clears the textfields of ctf8 and ctf9 and sets the text to empty

18. BackButton

The following button handles the event when the button is clicked. when the button is clicked, it sets the visibility of Credit frame to (true) and the visibility of CreditLimit frame to (false).

19. LimitDisplayButton

The following button handles the event when the button is clicked. First the method loops through the list of the bank card and for each credit card added to the list, if the bankcard is an instance of Credit card it calls the Credit card object in which the display method is called from the credit card class

public static void main(String[]args)-

the main method is used to create a new instance of Bank_GUI class.

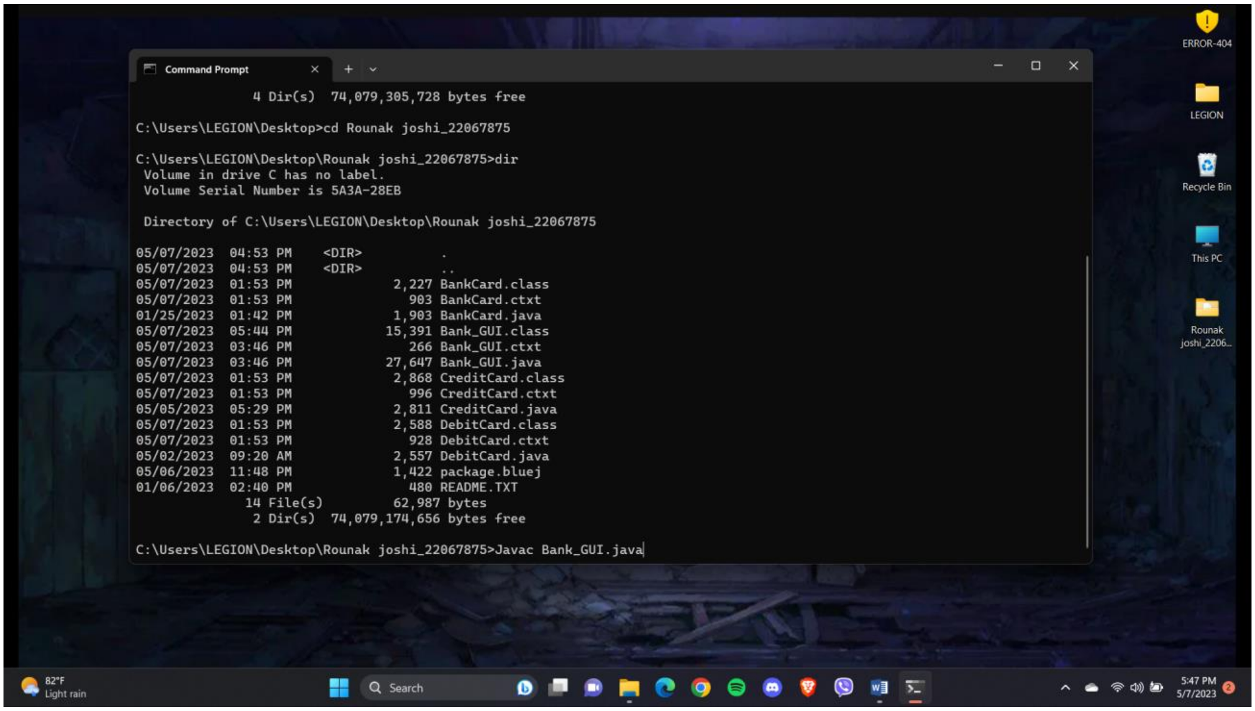
5. TESTING

5.1. TEST CASE 1 – Compiling & Running using command prompt

Test No	1
Objective	To Compile & Run the code using command prompt
Actions	• Open command prompt • Use (dir) for directory of the java file • Use cd to open the java file location • The following argument to be done to compile the code- Javac Bank_GUI • The following argument to be done to run the code • Java Bank_GUI • Insert values to the text fields
Expected result	Code To Compile & Run using command prompt
Actual result	Code compiled and run successfully in command prompt
Conclusion	The test is successful

TABLE 1 COMPILING & RUNNING USING COMMAND PROMPT

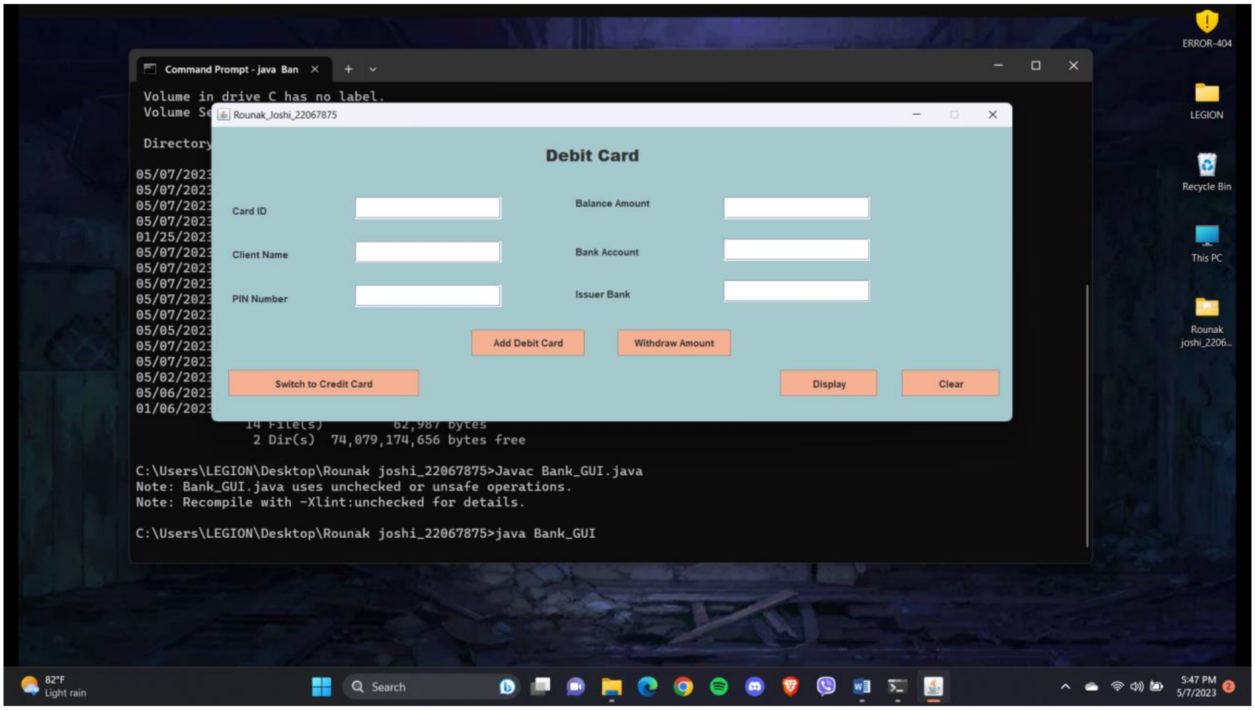
OUTPUT RESULT-



```
             4 Dir(s)  74,079,305,728 bytes free

C:\Users\LEGION\Desktop>cd Rounak joshi_22067875

C:\Users\LEGION\Desktop\Rounak joshi_22067875>dir
 Volume in drive C has no label.
 Volume Serial Number is 5A3A-28EB

 Directory of C:\Users\LEGION\Desktop\Rounak joshi_22067875

05/07/2023  04:53 PM    <DIR>          .
05/07/2023  04:53 PM    <DIR>          ..
05/07/2023  01:53 PM             2,227 BankCard.class
05/07/2023  01:53 PM               903 BankCard.ctxt
01/25/2023  01:42 PM             1,903 BankCard.java
05/07/2023  05:44 PM            15,391 Bank_GUI.class
05/07/2023  03:46 PM               266 Bank_GUI.ctxt
05/07/2023  03:46 PM            27,647 Bank_GUI.java
05/07/2023  01:53 PM             2,868 CreditCard.class
05/07/2023  01:53 PM               996 CreditCard.ctxt
05/05/2023  05:29 PM             2,811 CreditCard.java
05/07/2023  01:53 PM             2,588 DebitCard.class
05/07/2023  01:53 PM               928 DebitCard.ctxt
05/02/2023  09:20 AM             2,557 DebitCard.java
05/06/2023  11:48 PM             1,422 package.bluej
01/06/2023  02:40 PM               480 README.TXT
              14 File(s)         62,987 bytes
               2 Dir(s)  74,079,174,656 bytes free

C:\Users\LEGION\Desktop\Rounak joshi_22067875>Javac Bank_GUI.java
```

FIGURE 6 SCREENSHOT OF COMPILING JAVA FILE

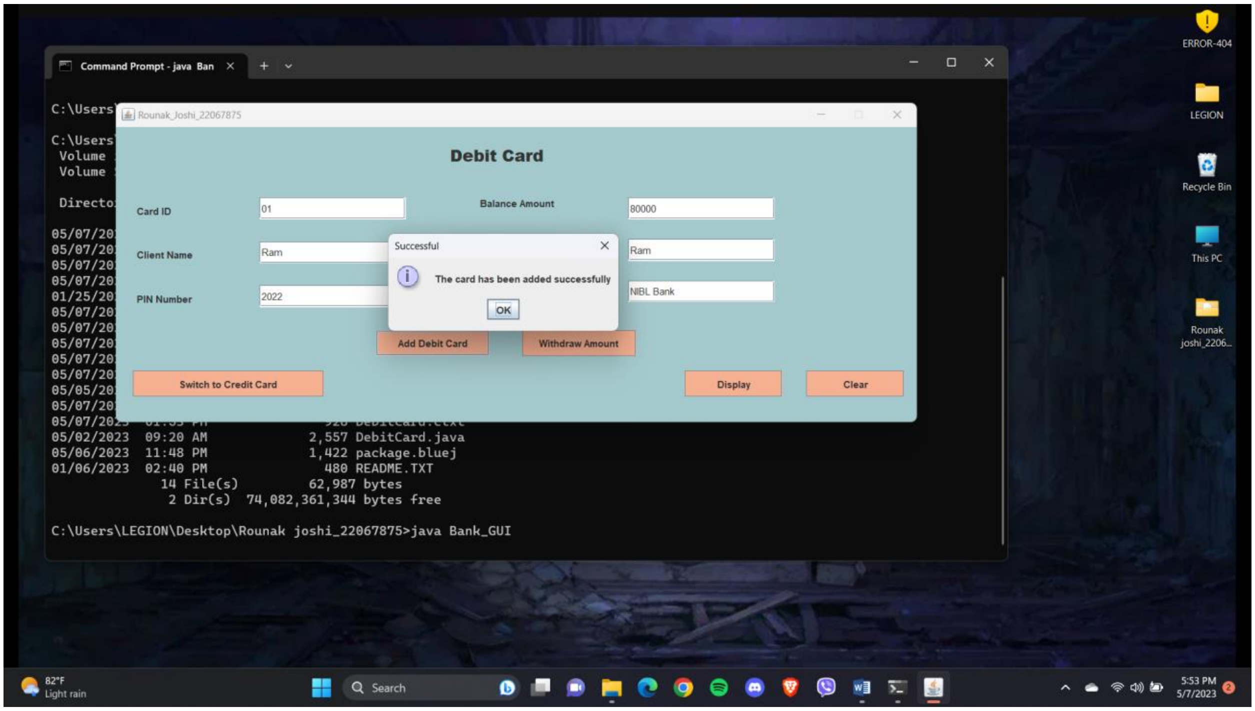
Image description- The above image shows the process of compiling the code through typing javac and the name of the java file



```
 Volume in drive C has no label.
 Volume Se

 Directory

05/07/202
05/07/202
05/07/202
05/07/202
01/25/202
05/07/202
05/07/202
05/07/202
05/07/202
05/07/202
05/05/202
05/07/202
05/07/202
05/02/202
05/06/202
01/06/202
              14 File(s)         62,987 bytes
               2 Dir(s)  74,079,174,656 bytes free

C:\Users\LEGION\Desktop\Rounak joshi_22067875>Javac Bank_GUI.java
Note: Bank_GUI.java uses unchecked or unsafe operations.
Note: Recompile with -Xlint:unchecked for details.

C:\Users\LEGION\Desktop\Rounak joshi_22067875>java Bank_GUI
```

FIGURE 7 SCREENSHOT OF JAVA FILE SUCCESSFULLY OPENING THROUGH COMMAND PROMPT

Image description- The above image shows java file successfully opening through command prompt after typing java and java file name.



```
C:\Users

C:\Users
 Volume
 Volume

 Directo

05/07/20
05/07/20
05/07/20
05/07/20
01/25/20
05/07/20
05/07/20
05/07/20
05/07/20
05/07/20
05/05/20
05/07/20
05/07/20
05/02/2023  09:20 AM             2,557 DebitCard.java
05/06/2023  11:48 PM             1,422 package.bluej
01/06/2023  02:40 PM               480 README.TXT
              14 File(s)         62,987 bytes
               2 Dir(s)  74,082,361,344 bytes free

C:\Users\LEGION\Desktop\Rounak joshi_22067875>java Bank_GUI
```

Rounak_Joshi_22067875

Debit Card

Card ID: 01

Client Name: Ram

PIN Number: 2022

Balance Amount: 80000

Ram

NIBL Bank

Successful

The card has been added successfully

FIGURE 8 SCREENSHOT OF USING GUI THROUGH COMMAND PROMPT

Image description- the above image shows the GUI successfully working through command prompt

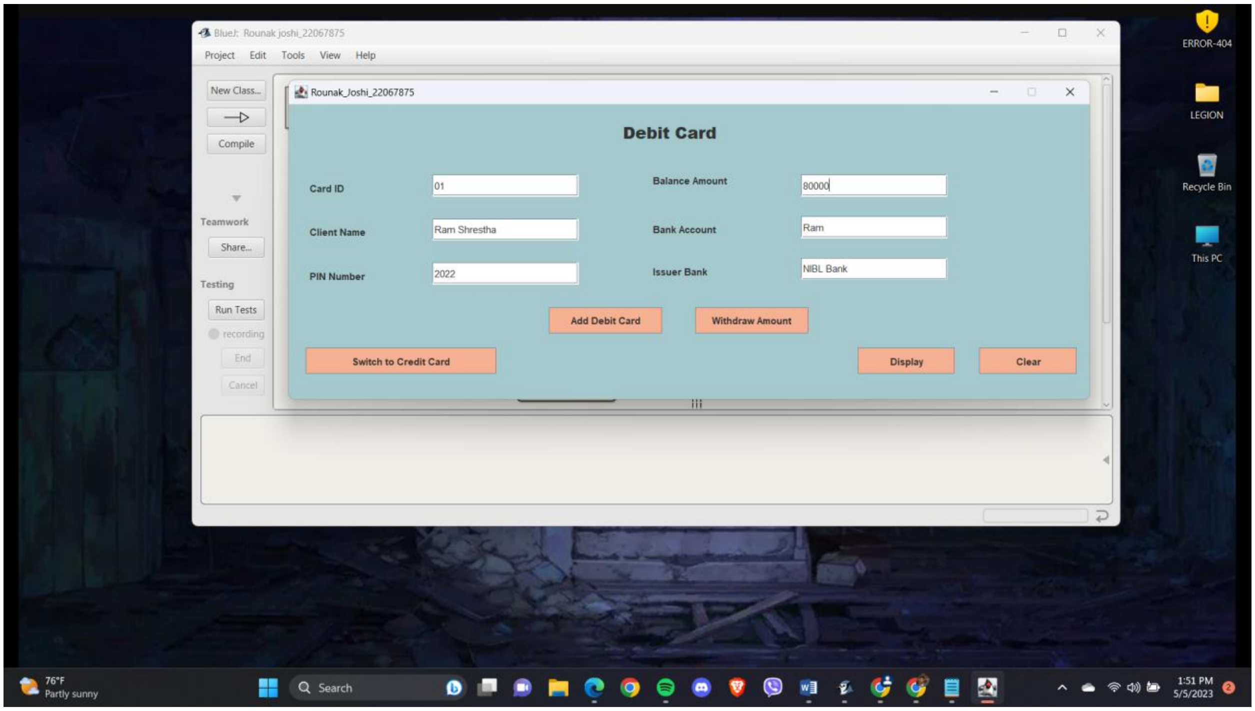
5.2. TEST CASE 2 - Adding objects of DebitCard and CreditCard, withdrawing amount from debit card, setting the credit limit and removing the credit card

Test Case 5.2.1 – Add objects to the Debit Card and Withdraw Amount Form Debit Card

Test No.	2.1
Objective	To add objects of DebitCard and Withdraw Amount From debit card
Action	• The Following values are entered to the required textFields CardID: 01 ClientName: Ram Shrestha PIN Number: 2022 Balance Amount: 80000 Bank Account: Ram Issuer Bank NIBL Bank • Add the card to the arraylist by using Add debit card button • Use Display Button to display the information added to the list • Use Withdraw Amount Button to open the withdraw frame to withdraw open • The following Information are added to the textfield PIN Number: 2022 Withdraw Amount: 10000 Date of Withdrawal 7- June- 2023 • Use Withdraw Amount Button to withdraw the amount from the Account • Use Display to display the Information of the Card
Expected Result	The debit card to be added to the arraylist and the given amount to be withdrawn from the user account
Actual result	The card was added to the arraylist and the amount was withdrawn from the user account
Conclusion	The test is successful

TABLE 2 ADDING OBJECTS TO OF DEBIT CARD AND WITHDRAWING AMOUNT FROM THE DEBIT CARD

Output Result:



Debit Card

Card ID: 01

Client Name: Ram Shrestha

PIN Number: 2022

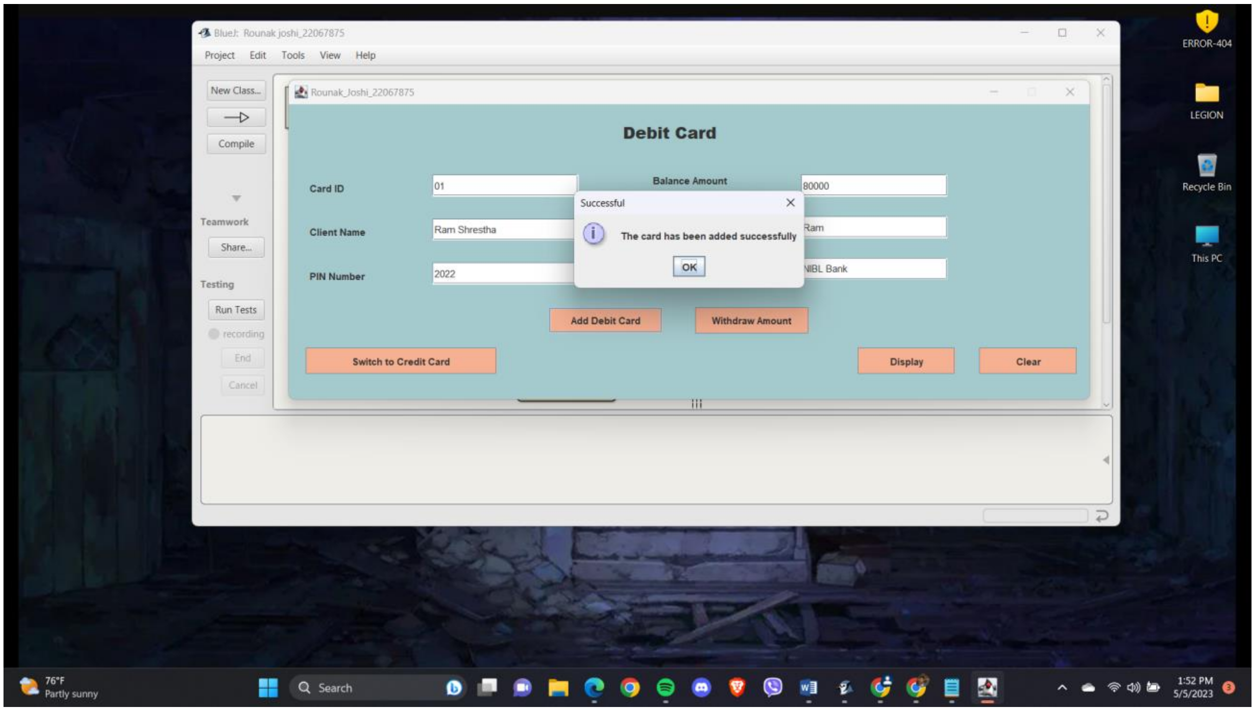
Balance Amount: 80000

Bank Account: Ram

Issuer Bank: NIBL Bank

FIGURE 9 INSERTING THE OBJECTS TO THE DEBIT CARD

Image description - (The above Image Shows inserting the required values to the textfileds in the debit card)



Debit Card

Card ID: 01

Client Name: Ram Shrestha

PIN Number: 2022

Balance Amount: 80000

Bank Account: Ram

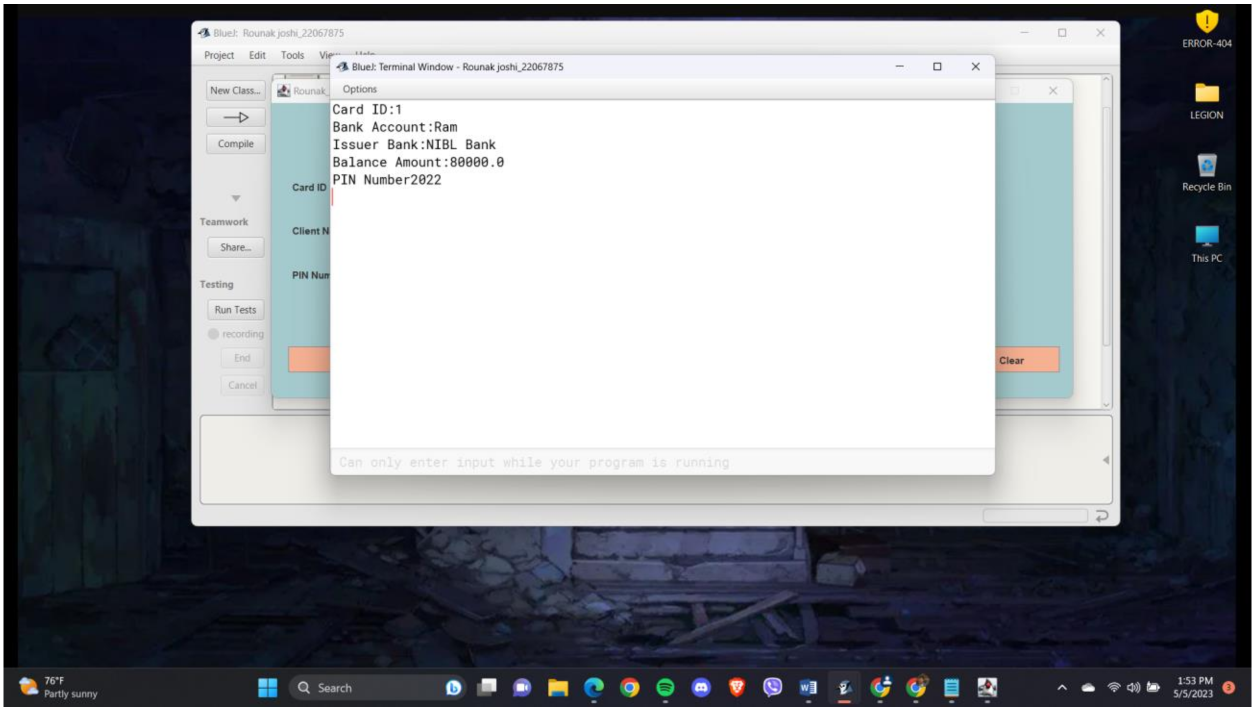
Issuer Bank: NIBL Bank

Successful

The card has been added successfully

FIGURE 10ADDING THE OBJECTS OF DEBIT CARD TO THE ARRAYLIST

Image description-(The above image shows the message dialog being displayed after successfully adding the card to the arraylist of bank card)

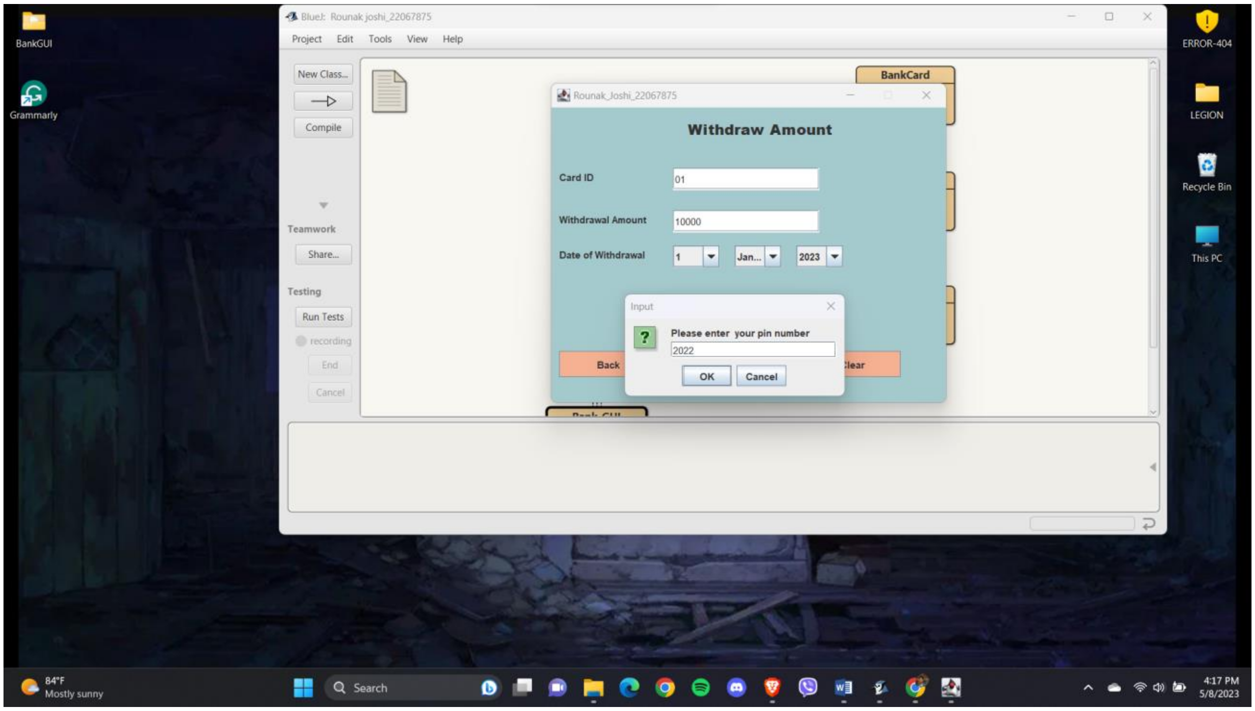


BlueJ: Rounak joshi_22067875
Project Edit Tools View Help
New Class...
Compile
Teamwork
Share...
Testing
Run Tests
recording
End
Cancel
BlueJ: Terminal Window - Rounak joshi_22067875
Options
Card ID
Client N
PIN Num
Clear
Can only enter input while your program is running

```
Card ID:1
Bank Account:Ram
Issuer Bank:NIBL Bank
Balance Amount:80000.0
PIN Number2022
```

FIGURE 11 DISPLAYING THE USER INFORMATION OF THE DEBIT CARD ADDED IN THE ARRAYLIST

Image description- (the above image shows the use of display button which shows the information of the debit card which has been added to the arraylist)



Withdraw Amount

Card ID: 01

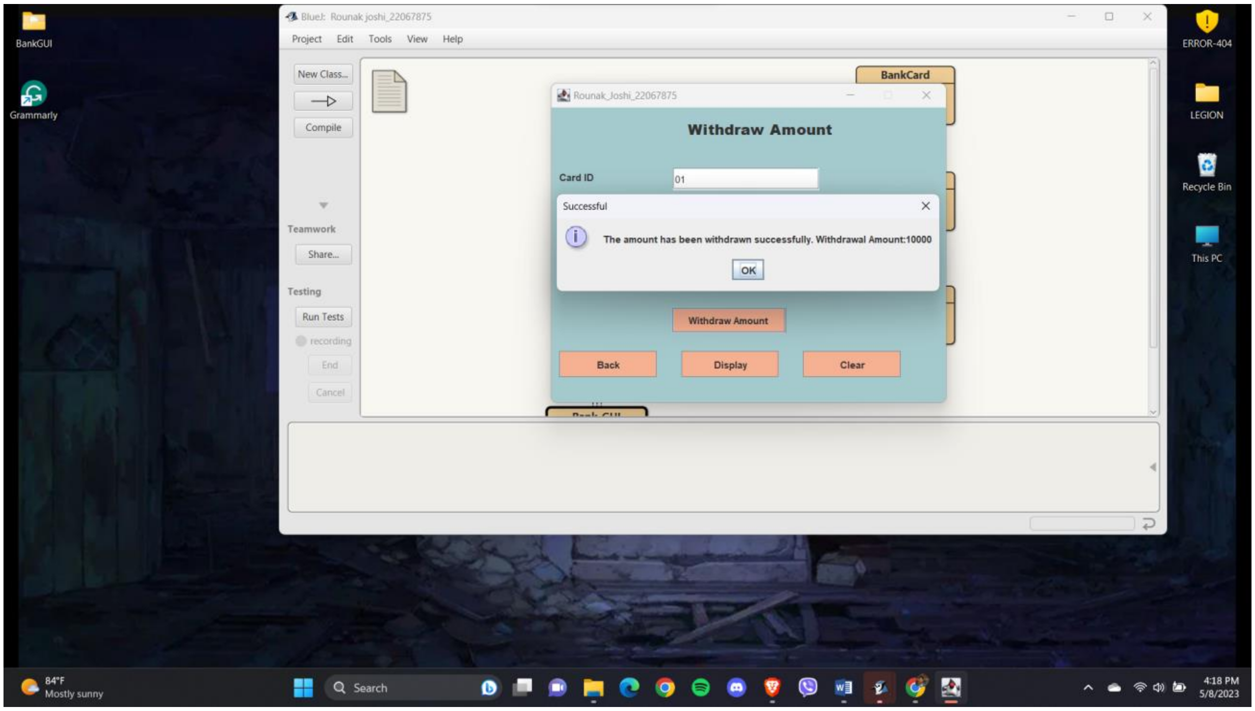
Withdrawal Amount: 10000

Date of Withdrawal: 1 Jan... 2023

Please enter your pin number: 2022

FIGURE 12ADDING WITHDRAWAL AMOUNT AND GRACE PERIOD TO WITHDRAW AMOUNT FROM THE USER ACCOUNT

Image description- (the above image shows adding the withdrawal amount and date of withdrawal to the textfileds in order to withdraw amount)

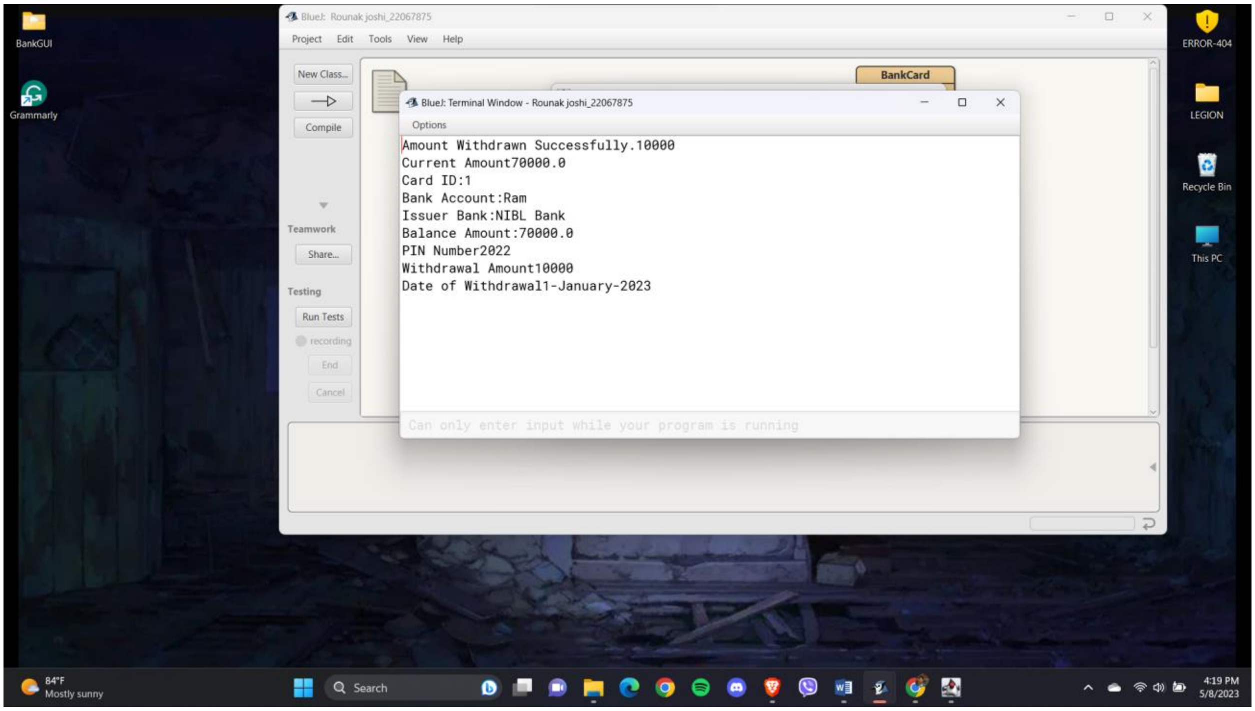


Card ID 01

The amount has been withdrawn successfully. Withdrawal Amount:10000

FIGURE 13MESSAGE SHOWN AFTER SUCCESSFULLY WITHDRAWING THE AMOUNT

Image description- (the message dialog is shown after successfully withdrawing the amount from the user account by using withdraw amount button)



```
Amount Withdrawn Successfully.10000
Current Amount70000.0
Card ID:1
Bank Account:Ram
Issuer Bank:NIBL Bank
Balance Amount:70000.0
PIN Number2022
Withdrawal Amount10000
Date of Withdrawal1-January-2023
```

FIGURE 14 DISPLAYING THE INFORMATION OF THE USER AFTER WITHDRAWING AMOUNT FROM THE USER ACCOUNT

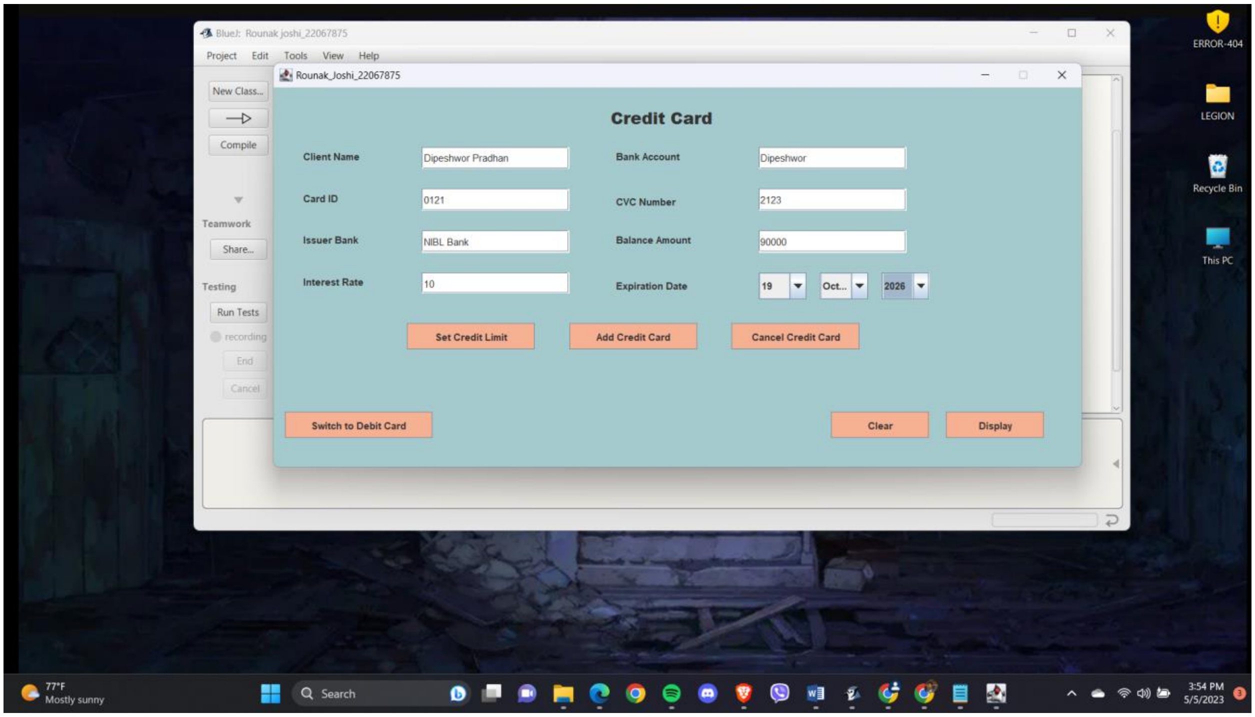
Image description -(the above image shows the use of display button displaying the information after withdrawing amount from the account and assigning grace period)

Test Case 5.2.2 Adding objects to the Credit Card, setting the credit limit and removing the credit card

Test No	2.2
Objective	To add objects to the Credit Card, set the credit limit and remove the credit card
Action	• The Following values are entered to the required textfieds: Client Name: Dipeshwor Pradhan CardID: 0121 Issuer Bank: NIBL Bank Interest Rate 10 Bank Account: Dipeshwor CVC Number: 2123 Balance Amount: 90000 Expiration date: 19-october-2026 • Use Add Credit Card to add the objects to the arraylist • Use of Display button to display the information of user of Credit card added to the arraylist • Use of Set Limit Button to open the Set credit limit panel • To set limit • The following information are added to the required textfileds: Credit Limit: 50000 Grace Period: 31 • Use of set credit limit button to set the limit and assign grace period to the credit card of the user • Use of display button to display the information of the user • Use of back button to return to credit card panel • Use of cancel credit card button to cancel the credit card • Use of display button to display information after cancelling credit card
Expected result	objects to be added the Credit Card, to set the credit limit and credit card to be removed
Actual Result	objects added to the Credit Card , credit limit set as required and credit card removed
Conclusion	The test is successful

TABLE 3 ADDING OBJECTS TO THE CREDIT CARD, SETTING THE CREDIT LIMIT AND REMOVING THE CREDIT CARD

Output Result:

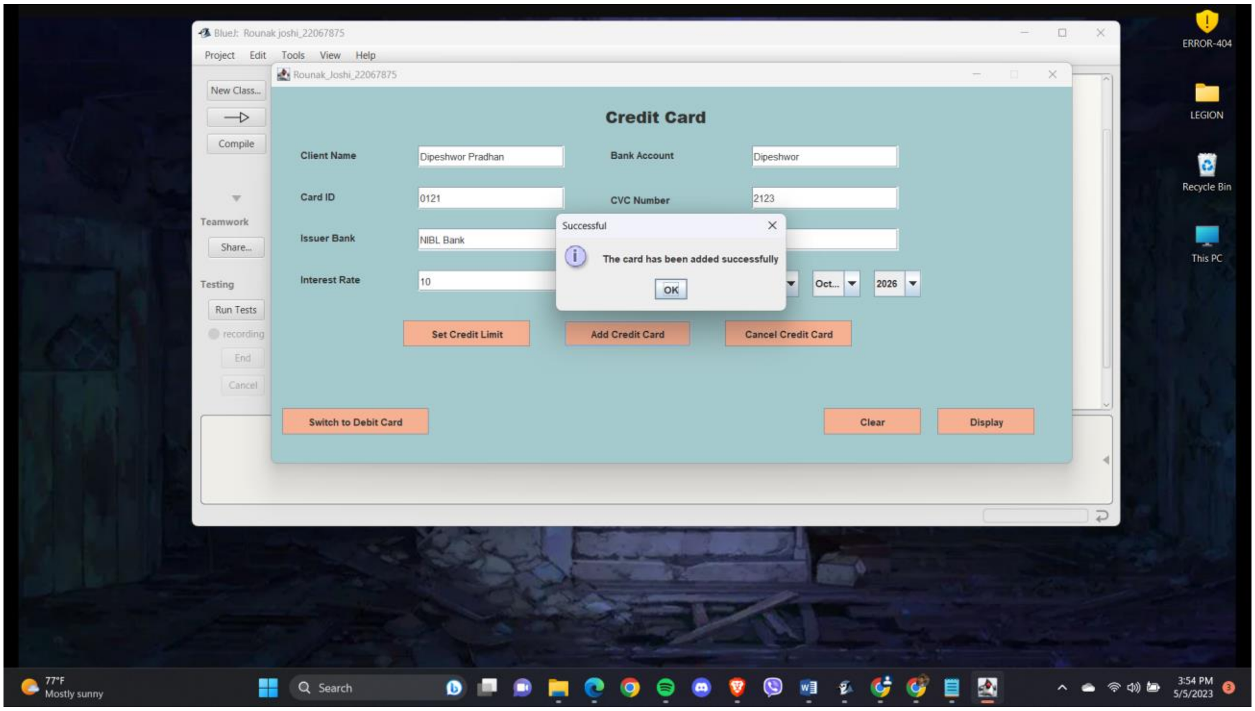


Credit Card

Client Name	Dipeshwor Pradhan	Bank Account	Dipeshwor
Card ID	0121	CVC Number	2123
Issuer Bank	NIBL Bank	Balance Amount	90000
Interest Rate	10	Expiration Date	19 Oct... 2026

FIGURE 15ADDING THE OBJECTS TO THE TEXTFILEDS OF THE CREDIT CARD

Image Description- (The above image shows inserting required values to the textfields of the credit card)



Credit Card

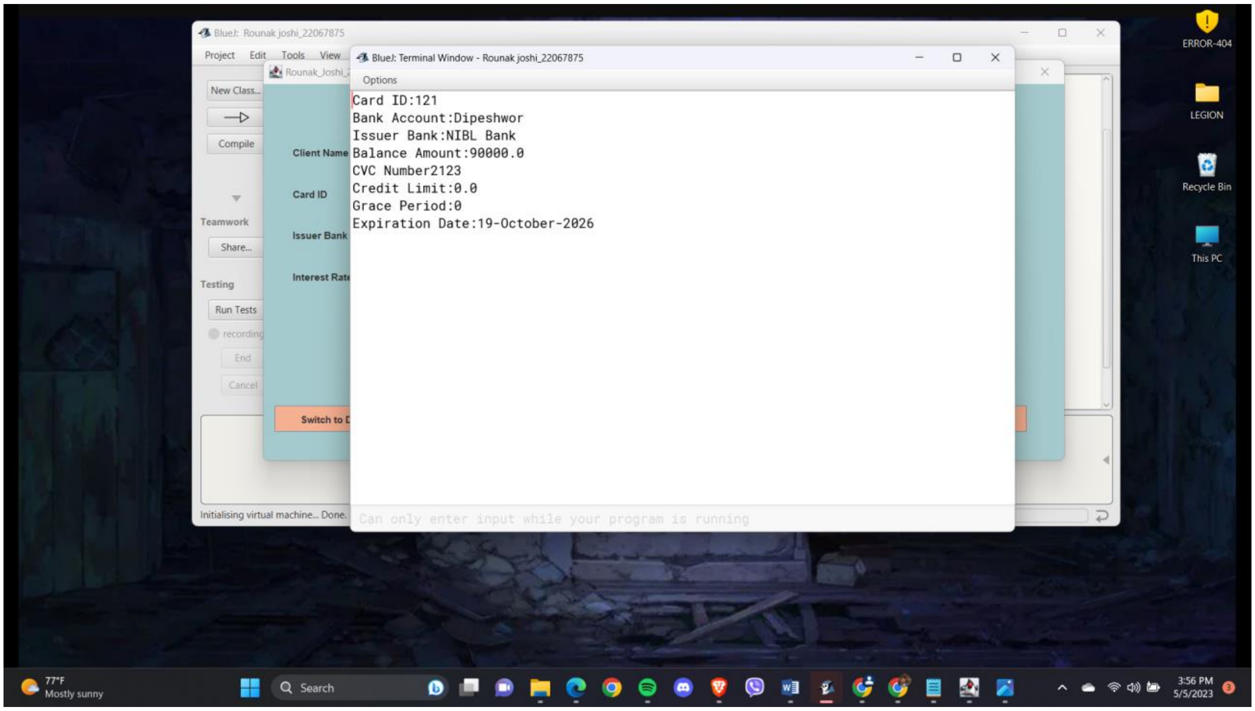
Client Name	Dipeshwor Pradhan	Bank Account	Dipeshwor
Card ID	0121	CVC Number	2123
Issuer Bank	NIBL Bank		
Interest Rate	10		Oct... 2026

Successful

The card has been added successfully

FIGURE 16 MESSAGE SHOWN OF SUCCESSFULLY ADDING THE CARD TO THE ARRAYLIST

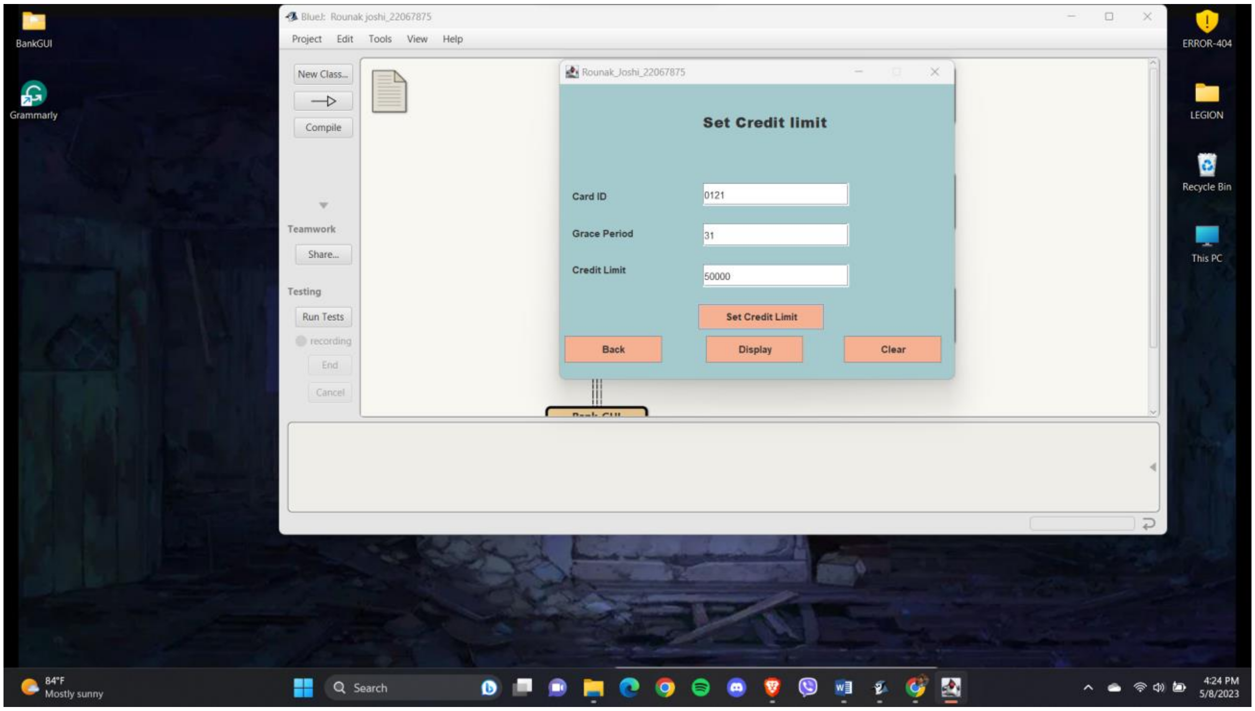
Image Description- (the above image shows the message dialog of successfully adding the objects of credit card to the arraylist)


BlueJ: Terminal Window - Rounak joshi_22067875
Options
Can only enter input while your program is running

```
Card ID:121
Bank Account:Dipeshwor
Issuer Bank:NIBL Bank
Balance Amount:90000.0
CVC Number2123
Credit Limit:0.0
Grace Period:0
Expiration Date:19-October-2026
```

FIGURE 17 DISPLAY OF THE USER INFORMATION ADDED TO THE ARRAYLIST

Image Description- (The above image shows the use of display button which shows the user information after being added to the arraylist)



Set Credit limit

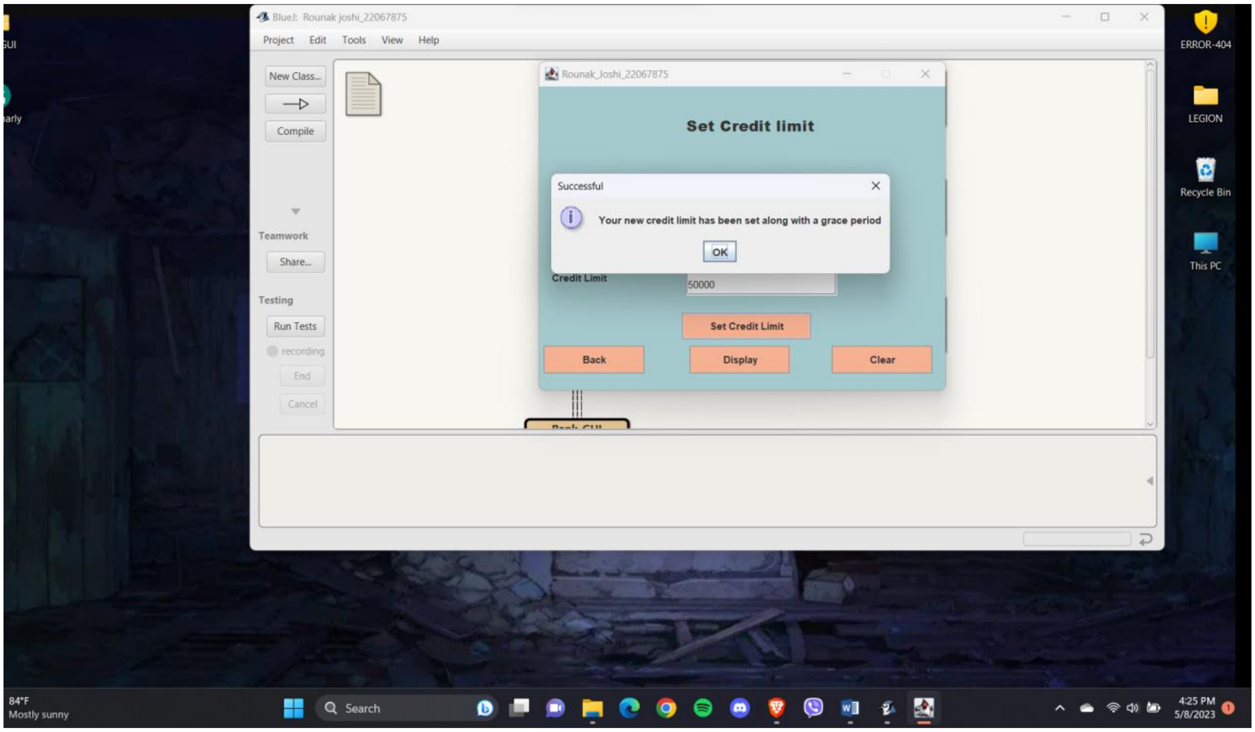
Card ID: 0121

Grace Period: 31

Credit Limit: 50000

FIGURE 18 ADDING CREDIT LIMIT AND GRACE PERIOD OF THE USER

Image Description- (The above image shows adding credit limit and grace period of the credit card of the user)



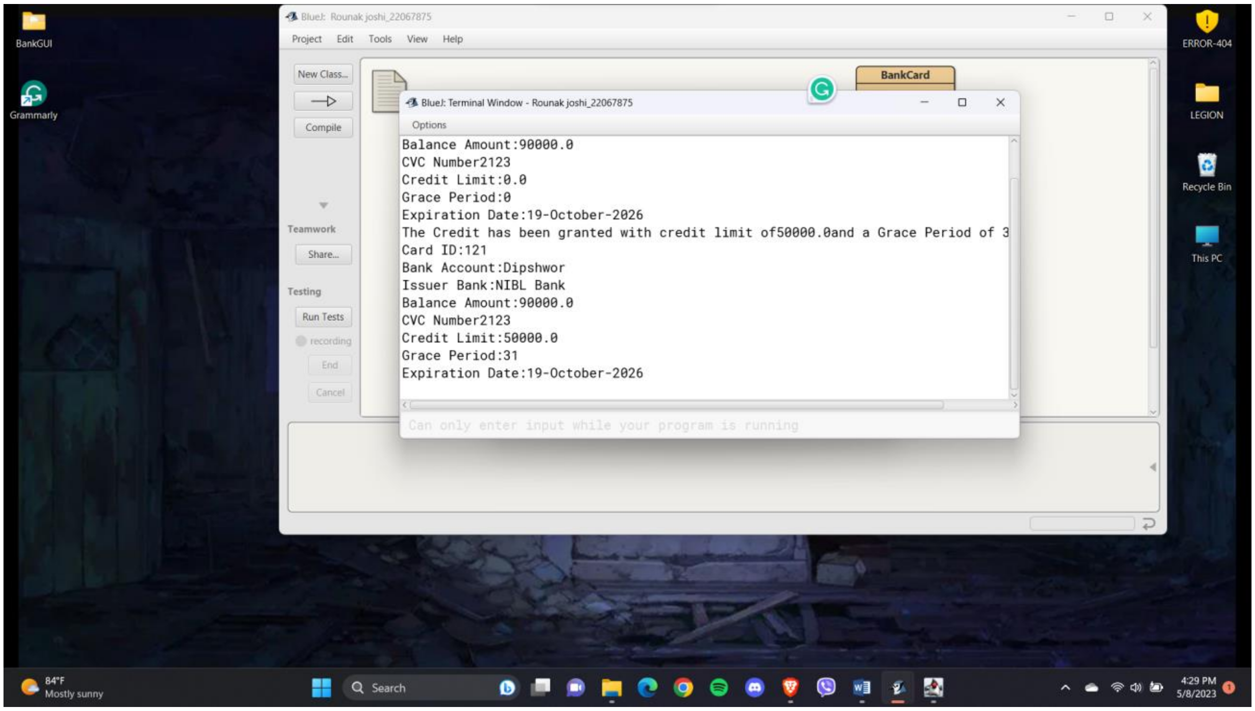
Successful

Your new credit limit has been set along with a grace period

OK

FIGURE 19 MESSAGE SHOWN AFTER SUCCESSFULLY ADDING GRACE PERIOD AND CREDIT LIMIT

Image Description- (The above image shows dialog message being shown after successfully adding credit limit and grace period to the card of the user)

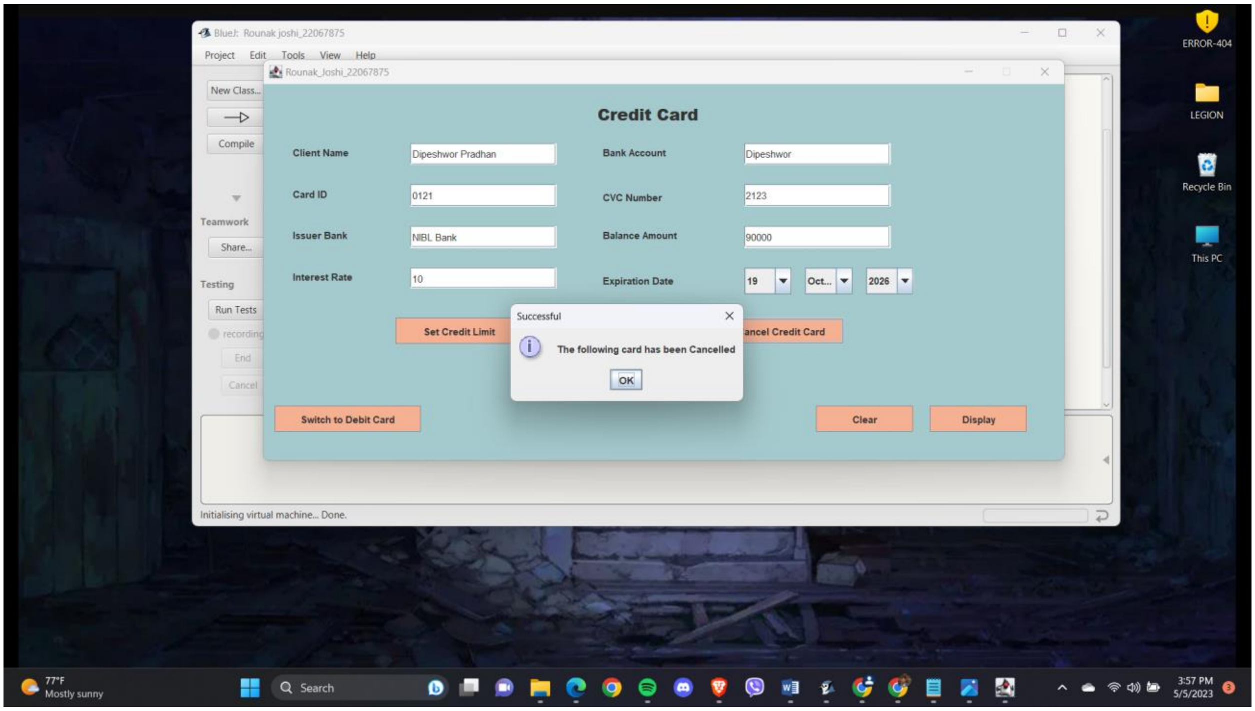


```
Balance Amount:90000.0
CVC Number2123
Credit Limit:0.0
Grace Period:0
Expiration Date:19-October-2026
The Credit has been granted with credit limit of50000.0and a Grace Period of 3
Card ID:121
Bank Account:Dipshwor
Issuer Bank:NIBL Bank
Balance Amount:90000.0
CVC Number2123
Credit Limit:50000.0
Grace Period:31
Expiration Date:19-October-2026
```

Can only enter input while your program is running

FIGURE 20 DISPLAYING THE INFORMATION OF USER AFTER ADDING CREDIT LIMIT AND GRACE PERIOD

Image Description- (The above image shows the use of display button displaying information of the user after adding grace period and credit limit to the card)



Credit Card

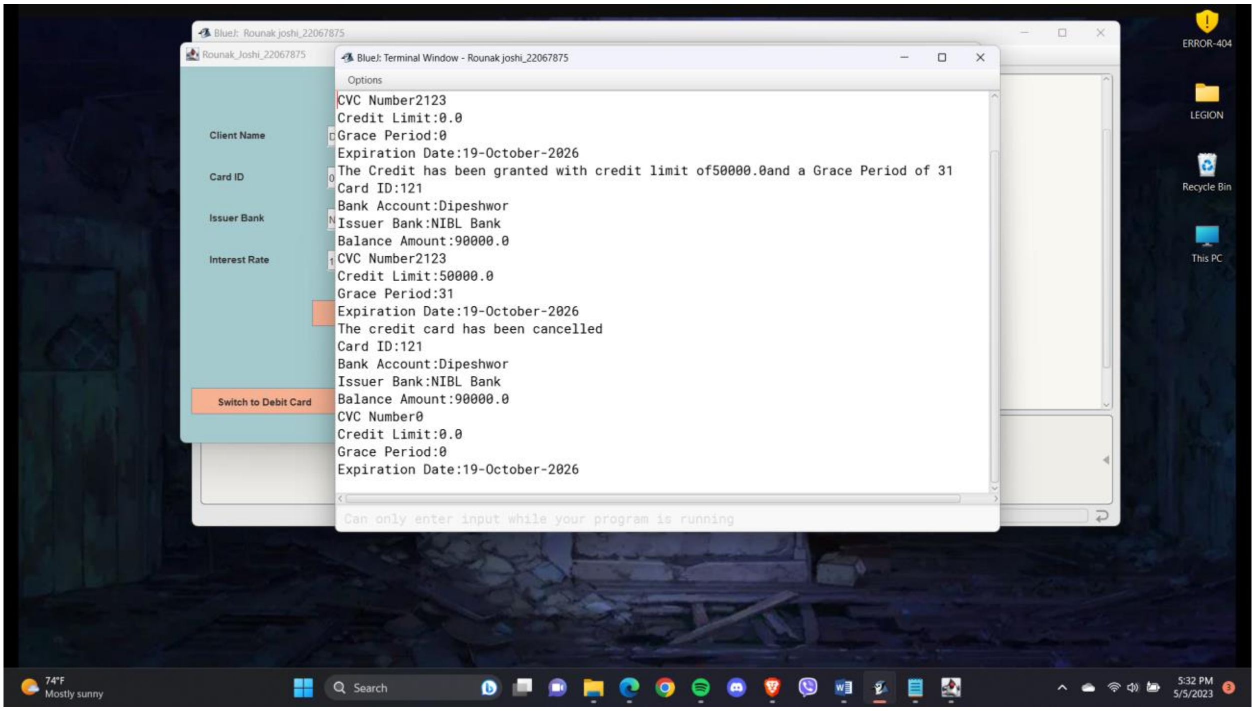
Client Name: Dipeshwor Pradhan
Bank Account: Dipeshwor
Card ID: 0121
CVC Number: 2123
Issuer Bank: NIBL Bank
Balance Amount: 90000
Interest Rate: 10
Expiration Date: 19 Oct... 2026

Successful

The following card has been Cancelled

FIGURE 21 CANCELING THE CREDIT CARD USING CANCEL BUTTON

Image Description- (the above image shows use cancel credit card button which cancels the credit card)



```
CVC Number2123
Credit Limit:0.0
Grace Period:0
Expiration Date:19-October-2026
The Credit has been granted with credit limit of50000.0and a Grace Period of 31
Card ID:121
Bank Account:Dipeshwor
Issuer Bank:NIBL Bank
Balance Amount:90000.0
CVC Number2123
Credit Limit:50000.0
Grace Period:31
Expiration Date:19-October-2026
The credit card has been cancelled
Card ID:121
Bank Account:Dipeshwor
Issuer Bank:NIBL Bank
Balance Amount:90000.0
CVC Number0
Credit Limit:0.0
Grace Period:0
Expiration Date:19-October-2026
```

FIGURE 22DISPLAYIN INFORMATION AFTER CANCELLING THE CREDIT CARD

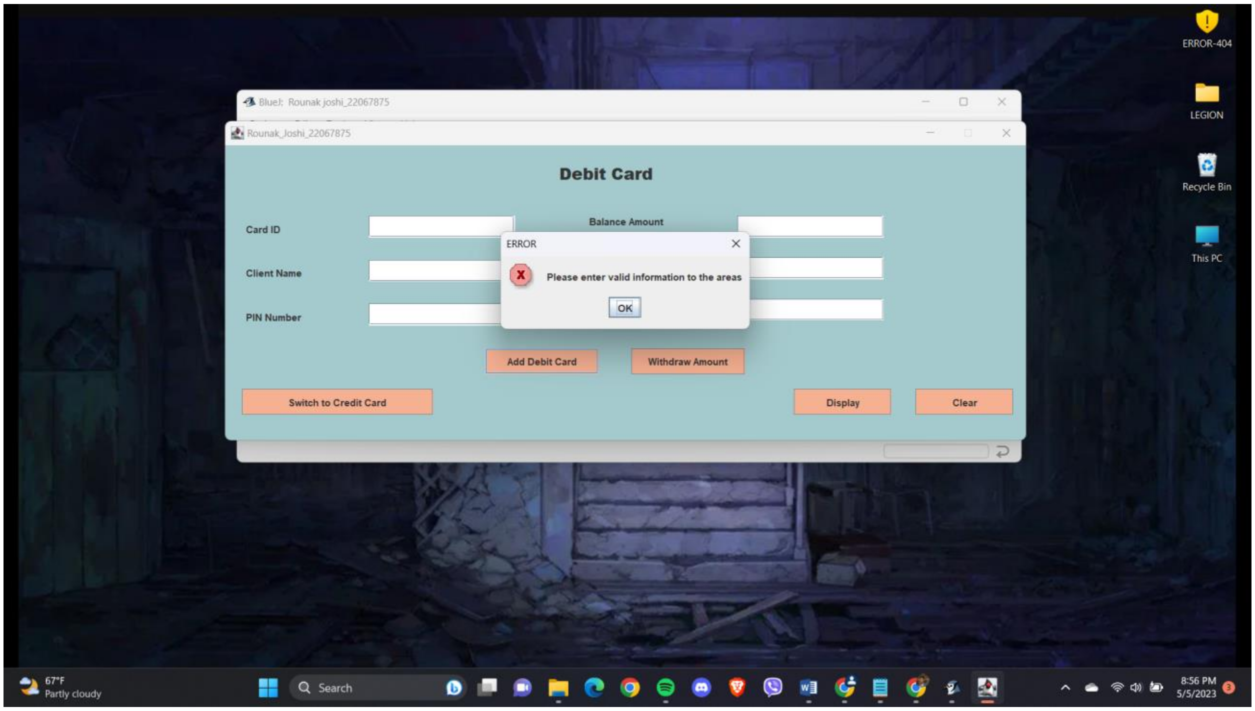
Image Description- ((The above image shows use of display button after cancelling credit card)

5.3. TEST CASE 3 –
Testing Appropriate Dialog boxes when unsuitable values entered

Test No	3
Objective	To Test Appropriate Dialog boxes when unsuitable values entered
Action	1. When textfields are left empty • Use add card button without entering values to the textField 2. When String is entered instead of integer • Use of string instead of integer in the required textfields causing number format exception 3. When invalid pin number is entered during withdrawing amount in debit card • Writing a different pin number when withdrawing amount from the user account 4. When using same CardID for adding two different cards • Using same CardID for adding two different cards in the arrayList 5. When withdraw amount is more than users Balance amount • Adding of value in withdraw textfield more than balance amount textfield 6. When credit limit is higher than the balance amount 7. • Adding of value in credit limit textfield more than balance amount textfield
Expected Result	To Show ERROR message dialog when buttons used
Actual Result	ERROR message dialog shown using buttons when invalid input entered
Conclusion	The test is successful.

Table 4 Testing Appropriate Dialog boxes when unsuitable values entered

OUTPUT RESULT:



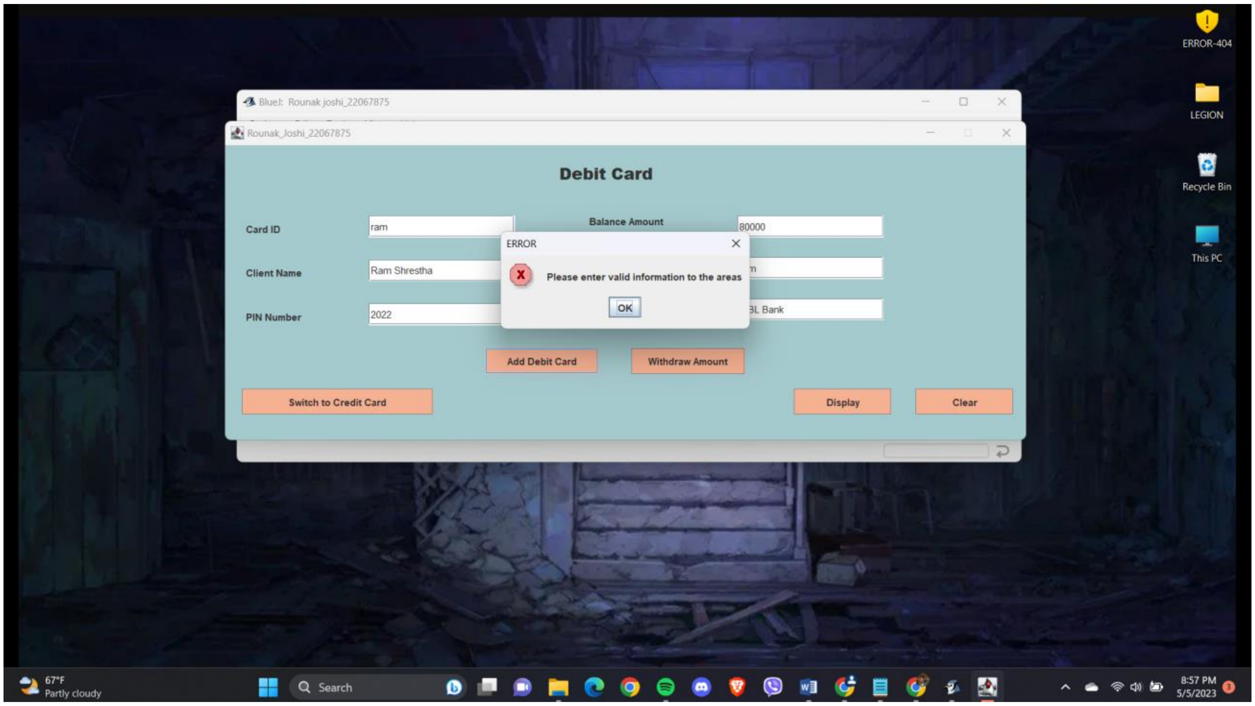
ERROR

Please enter valid information to the areas

OK

FIGURE 23 MESSAGE SHOWN FOR ADDING CARD WITH EMPTY TEXTFIELDS

Image Description- (the above image shows the message dialog when we try enter the card to arraylist with empty textfields)



Card ID: ram

Client Name: Ram Shrestha

PIN Number: 2022

Balance Amount: 80000

m

3L Bank

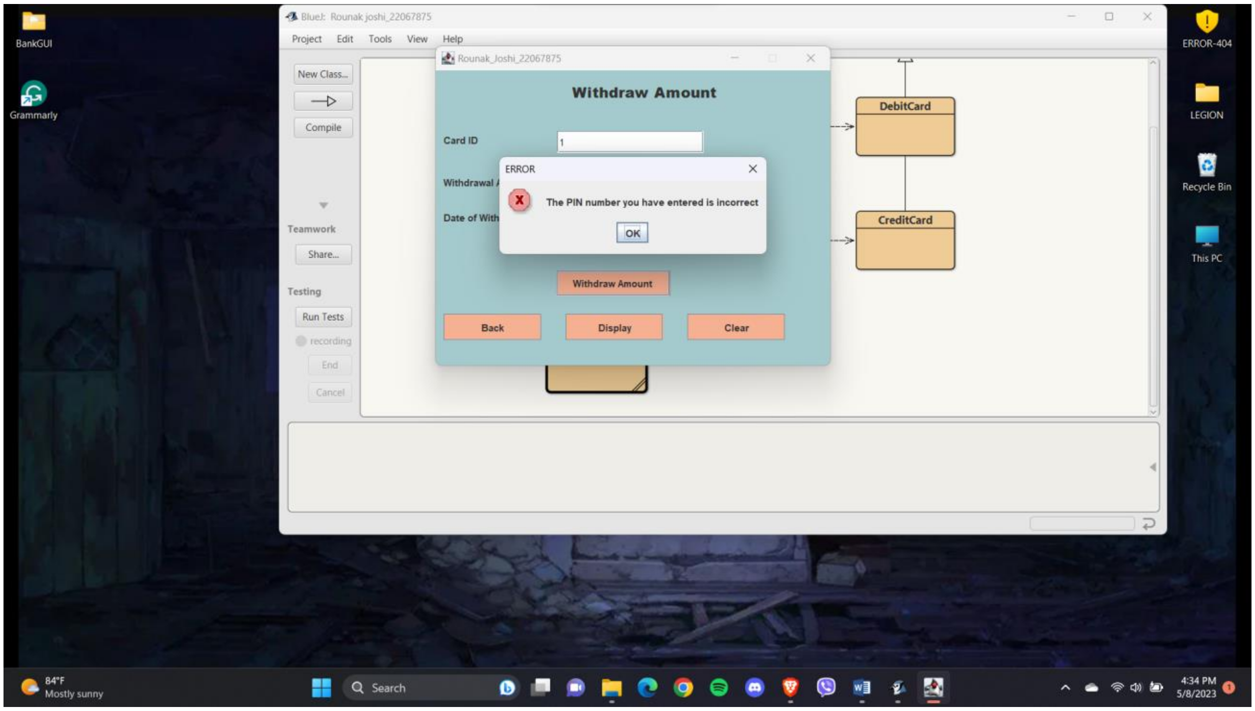
ERROR

Please enter valid information to the areas

OK

FIGURE 24 MESSAGE SHOWN WHEN INSERTING STRING IN INTEGER TEXTFIELDS

Image Description- (the above image shows message dialog when string is entered in integer textfield creating numberformatexception to occur)



Withdraw Amount

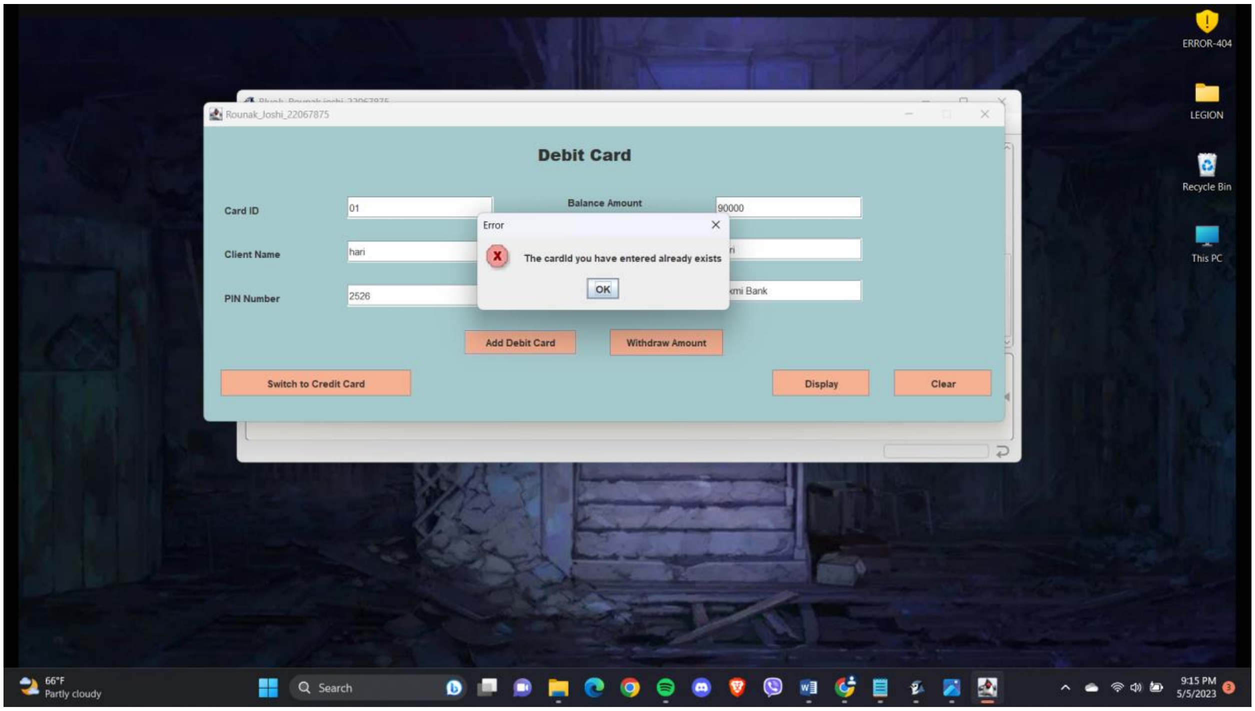
Card ID 1

ERROR

The PIN number you have entered is incorrect

FIGURE 25MESSAGE SHOWN WHEN INCORRECT PIN NUMBER IS WRITTEN

Image Description- (the above image shows message dialog when the pin number is not same to debit card pin number when trying to withdraw amount from the account)



Debit Card

Card ID 01

Client Name hari

PIN Number 2526

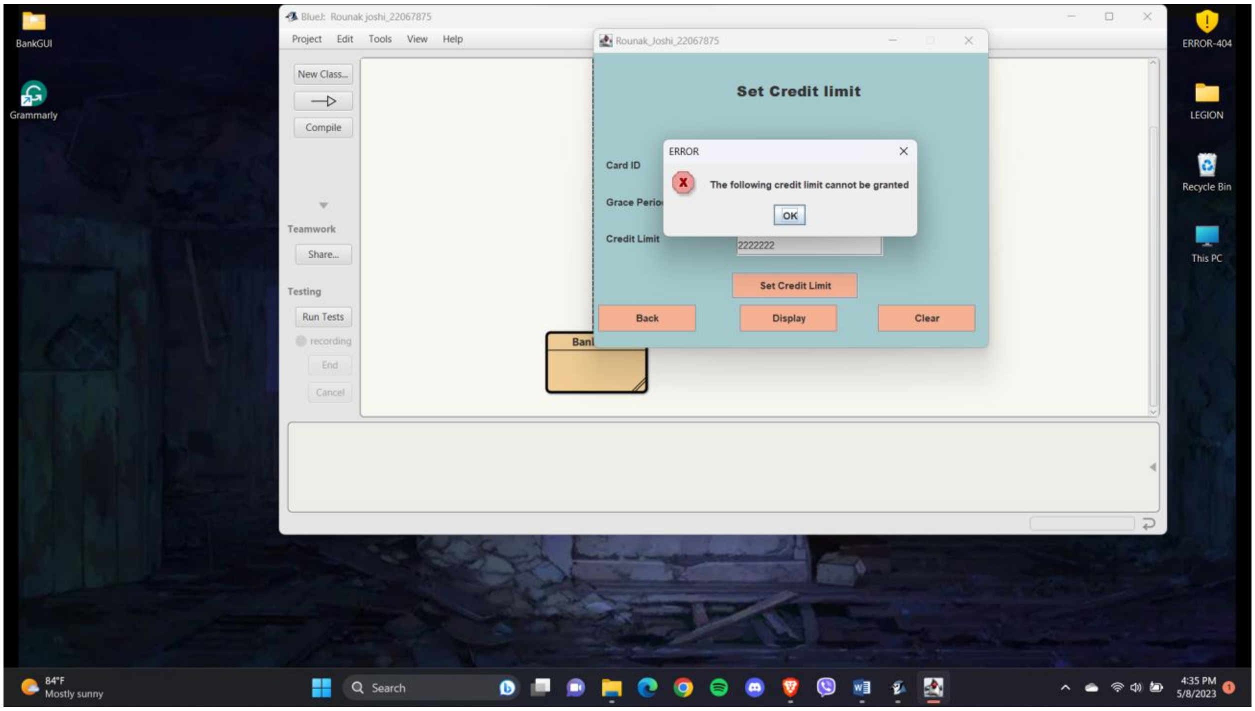
Balance Amount 90000

Error

The cardId you have entered already exists

FIGURE 26 MESSAGE SHOWN WHEN TWO DIFFERENT CARDS HAVE SAME CARDID

Image Description- (the above image shows message dialog when two different cards have the same cardid entered in the input)

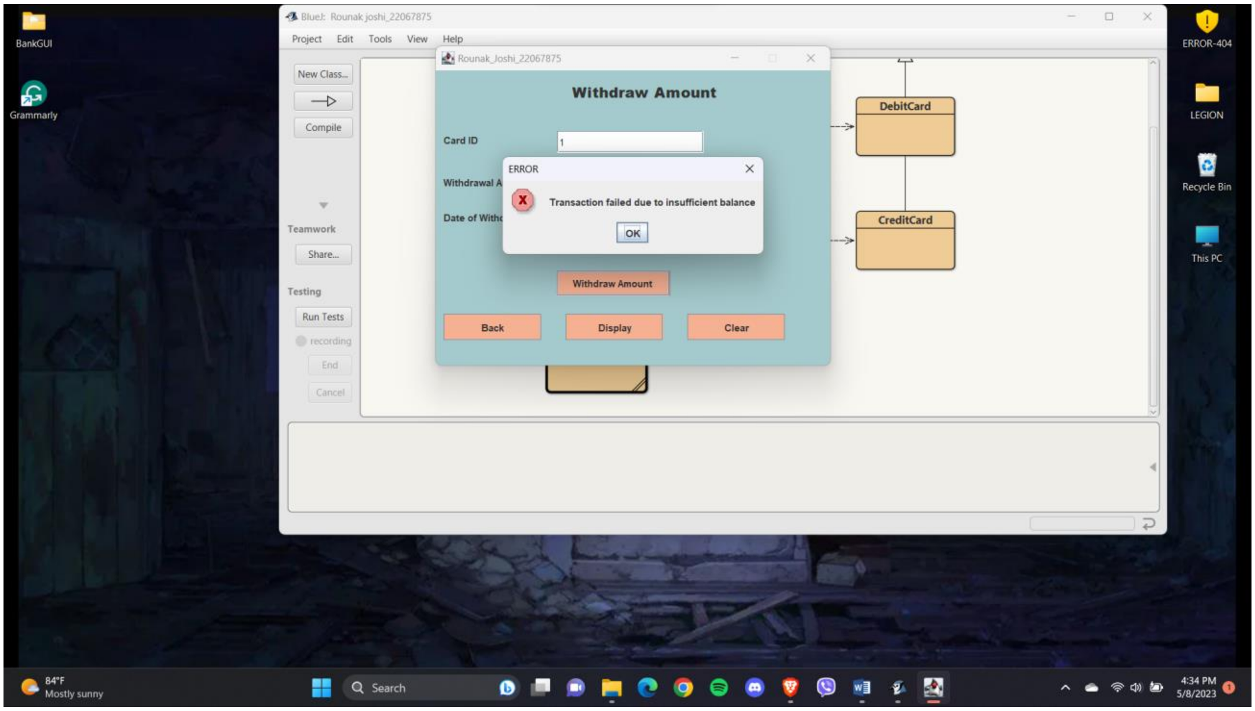


The following credit limit cannot be granted

2222222

FIGURE 27 MESSAGE SHOWN WHEN CREDIT LIMIT IS MORE THAN THE BALANCE AMOUNT

Image Description- (the above image shows message dialog when a person tries enter the value credit limit more than the value of the balance amount of the user)



Transaction failed due to insufficient balance

FIGURE 28 MESSAGE SHOWN DUE TO INSUFFICIENT BALANCE

Image Description- (the above image shows message dialog when user tries to withdraw more amount than the balance amount of the user)

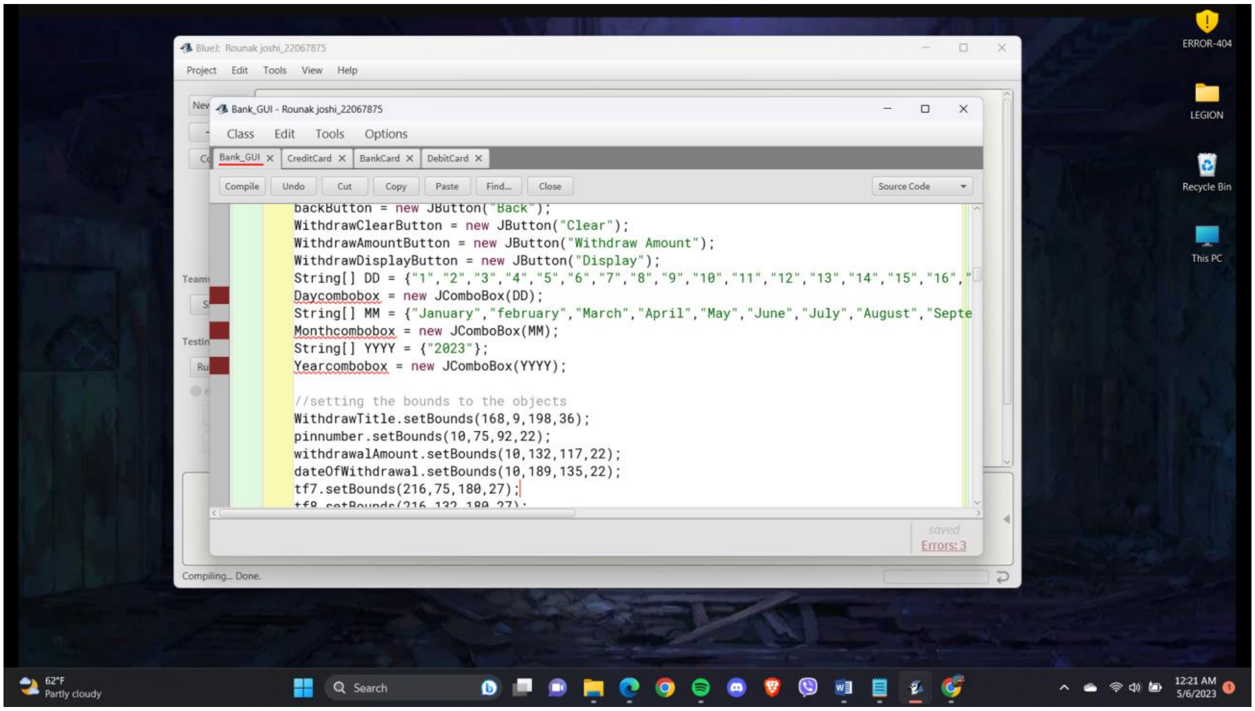
6.ERROR DETECTION AND CORRECTION

6.1. ERROR 1 SYNTAX ERROR

Error No	1
Type of error	Syntax Error
Error Occurred	Error shown due to using letters in lowercase instead of using uppercase letter as declared above
Solution	Writing the information proper letter case as required
Conclusion	Error Solved

TABLE 5 TABLE OF SYNTAX ERROR

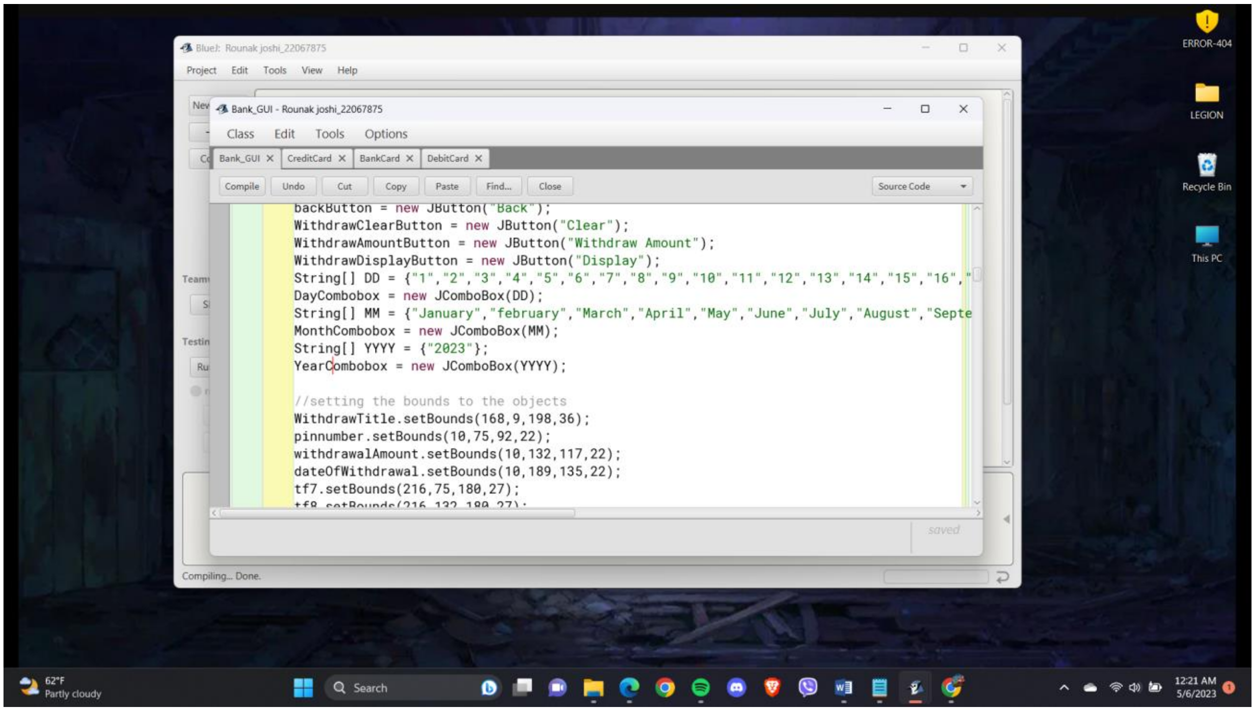
Evidence of the error:



```
backButton = new JButton("Back");
WithdrawClearButton = new JButton("Clear");
WithdrawAmountButton = new JButton("Withdraw Amount");
WithdrawDisplayButton = new JButton("Display");
String[] DD = {"1","2","3","4","5","6","7","8","9","10","11","12","13","14","15","16","
Daycombobox = new JComboBox(DD);
String[] MM = {"January","february","March","April","May","June","July","August","Septe
Monthcombobox = new JComboBox(MM);
String[] YYYY = {"2023"};
Yearcombobox = new JComboBox(YYYY);

//setting the bounds to the objects
WithdrawTitle.setBounds(168,9,198,36);
pinnumber.setBounds(10,75,92,22);
withdrawalAmount.setBounds(10,132,117,22);
dateOfWithdrawal.setBounds(10,189,135,22);
tf7.setBounds(216,75,180,27);
tf8.setBounds(216,132,180,27);
```

FIGURE 29 SCREENSHOT OF SYNTAX ERROR OCCURRED DUE TO WRONG LETTER CASING

Image Description- The above image shows error occurred in the code sue to writing the objects in lower case instead of upper case as it is written and declared up above.


BlueJ: Rounak joshi_22067875
Project Edit Tools View Help
Bank_GUI - Rounak joshi_22067875
Class Edit Tools Options
Bank_GUI CreditCard BankCard DebitCard
Compile Undo Cut Copy Paste Find... Close
Source Code
saved
Compiling... Done.
ERROR-404
LEGION
Recycle Bin
This PC
62°F
Partly cloudy
Search
12:21 AM
5/6/2023

```
backButton = new JButton("Back");
WithdrawClearButton = new JButton("Clear");
WithdrawAmountButton = new JButton("Withdraw Amount");
WithdrawDisplayButton = new JButton("Display");
String[] DD = {"1","2","3","4","5","6","7","8","9","10","11","12","13","14","15","16","
DayCombobox = new JComboBox(DD);
String[] MM = {"January","february","March","April","May","June","July","August","Septe
MonthCombobox = new JComboBox(MM);
String[] YYYY = {"2023"};
YearCombobox = new JComboBox(YYYY);

//setting the bounds to the objects
WithdrawTitle.setBounds(168,9,198,36);
pinnumber.setBounds(10,75,92,22);
withdrawalAmount.setBounds(10,132,117,22);
dateOfWithdrawal.setBounds(10,189,135,22);
tf7.setBounds(216,75,180,27);
tf8.setBounds(216,132,180,27);
```

FIGURE 30 SCREEN SHOT OF SYNTAX ERROR BEING SOLVED

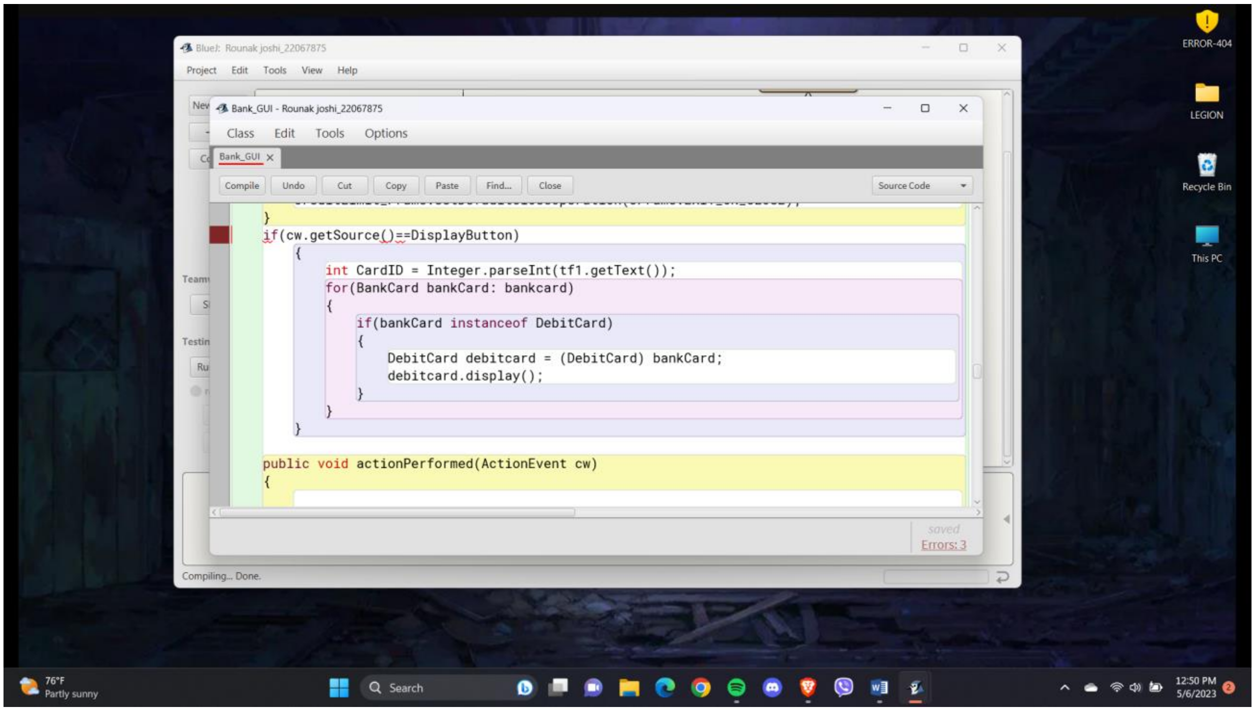
Image Description- the above image shows the error being solved as the objects are written properly in proper letter casing as written above.

6.2. ERROR 2 SEMANTIC ERROR

Error No	2
Type of error	Semantic error
Error Occurred	Calling and using getsource of button outside actionlistener method
Solution	Adding the button method within the actionlistener Method in order to use the button
Conclusion	Error solved

TABLE 6 TABLE OF SEMANTIC ERROR

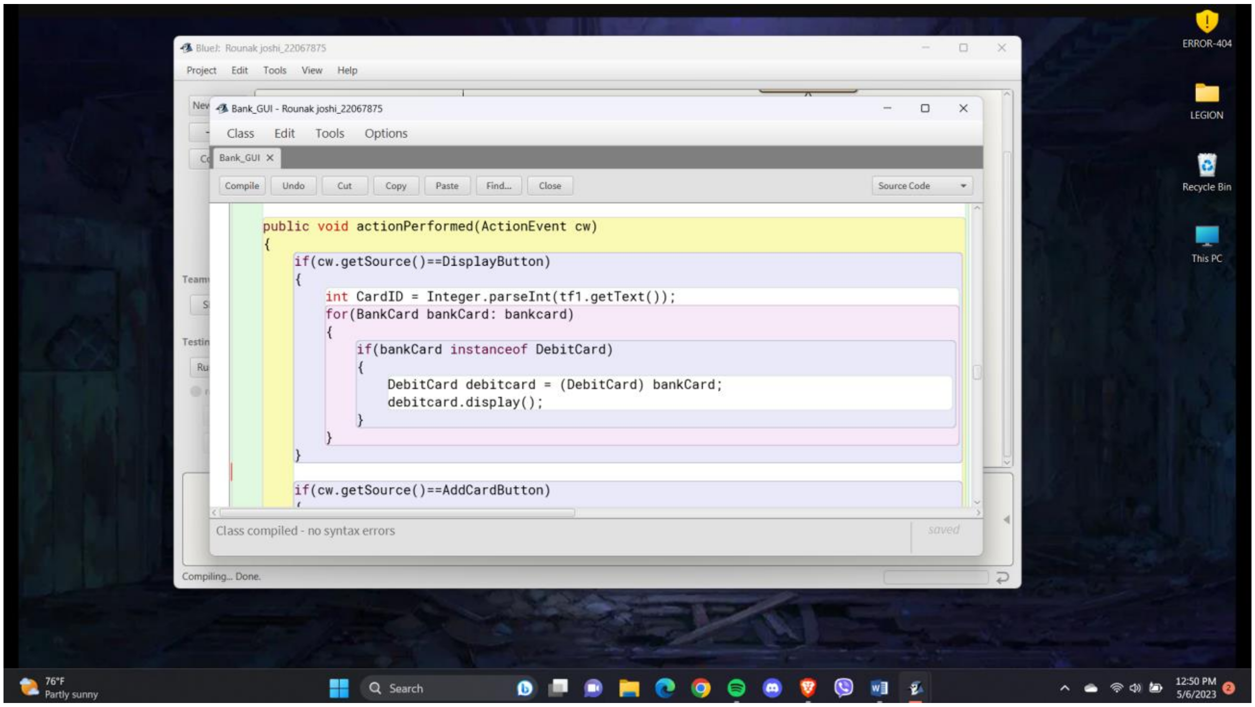
Evidence of the Error:



```
}
if(cw.getSource()==DisplayButton)
    {
        int CardID = Integer.parseInt(tf1.getText());
        for(BankCard bankCard: bankcard)
        {
            if(bankCard instanceof DebitCard)
            {
                DebitCard debitcard = (DebitCard) bankCard;
                debitcard.display();
            }
        }
    }

public void actionPerformed(ActionEvent cw)
{
```

FIGURE 31 SCREENSHOT OF SEMANTIC ERROR OCCURRING

Image description- the above image shows Semantic error being shown as the button is called outside the actionlistener method


BlueJ: Rounak joshi_22067875
Project Edit Tools View Help
Bank_GUI - Rounak joshi_22067875
Class Edit Tools Options
Bank_GUI
Compile Undo Cut Copy Paste Find... Close
Source Code
saved
ERROR-404
LEGION
Recycle Bin
This PC
76°F
Partly sunny
Search
12:50 PM
5/6/2023

```
public void actionPerformed(ActionEvent cw)
{
    if(cw.getSource()==DisplayButton)
    {
        int CardID = Integer.parseInt(tf1.getText());
        for(BankCard bankCard: bankcard)
        {
            if(bankCard instanceof DebitCard)
            {
                DebitCard debitcard = (DebitCard) bankCard;
                debitcard.display();
            }
        }
    }

    if(cw.getSource()==AddCardButton)
```

Class compiled - no syntax errors

Compiling... Done.

FIGURE 32 SCREENSHOT OF SEMANTIC ERROR BEING SOLVED

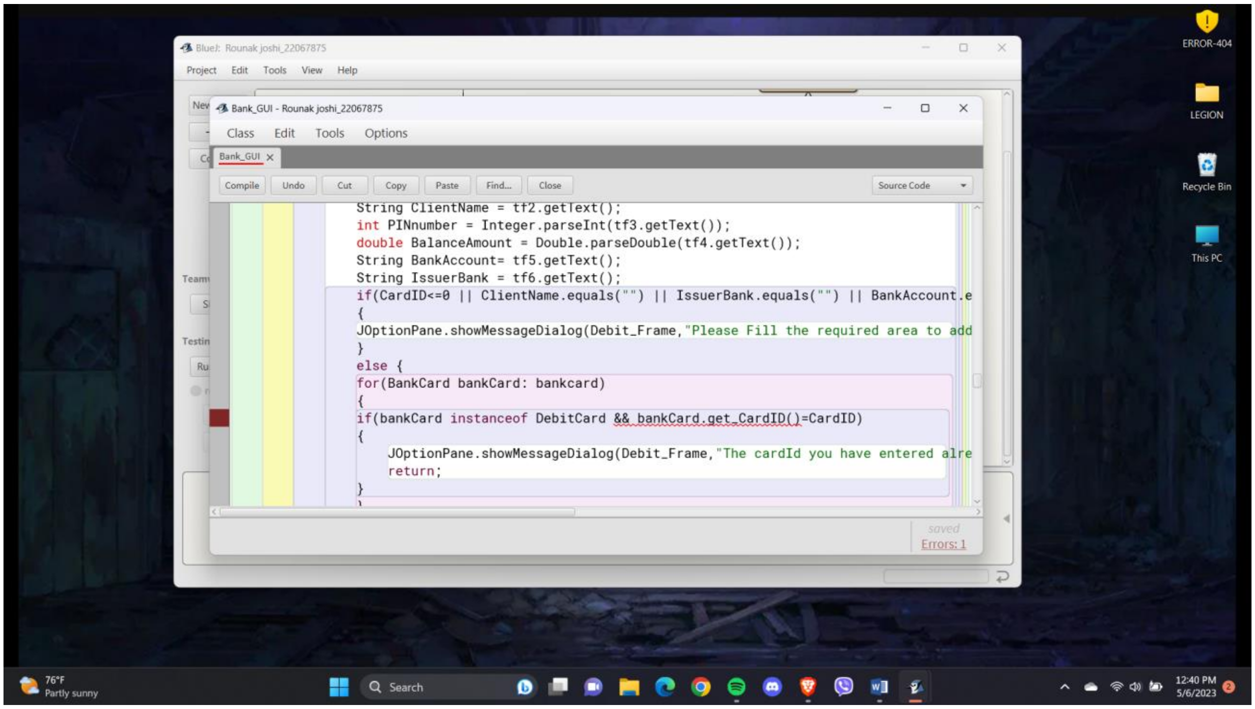
Image description – The above image shows the semantic error being solved as the buttons method is called and added inside the actionlistener method making the button to work successfully.

6.3. ERROR 3 LOGICAL ERROR

Error No	3
Type of error	Logical error
Error Occurred	Program showing error as use of only one = instead of two causing error to be shown
Solution	Use of two = sign in the needed area for the code to run
Conclusion	Error solved

TABLE 7TABLE OF LOGICAL ERROR

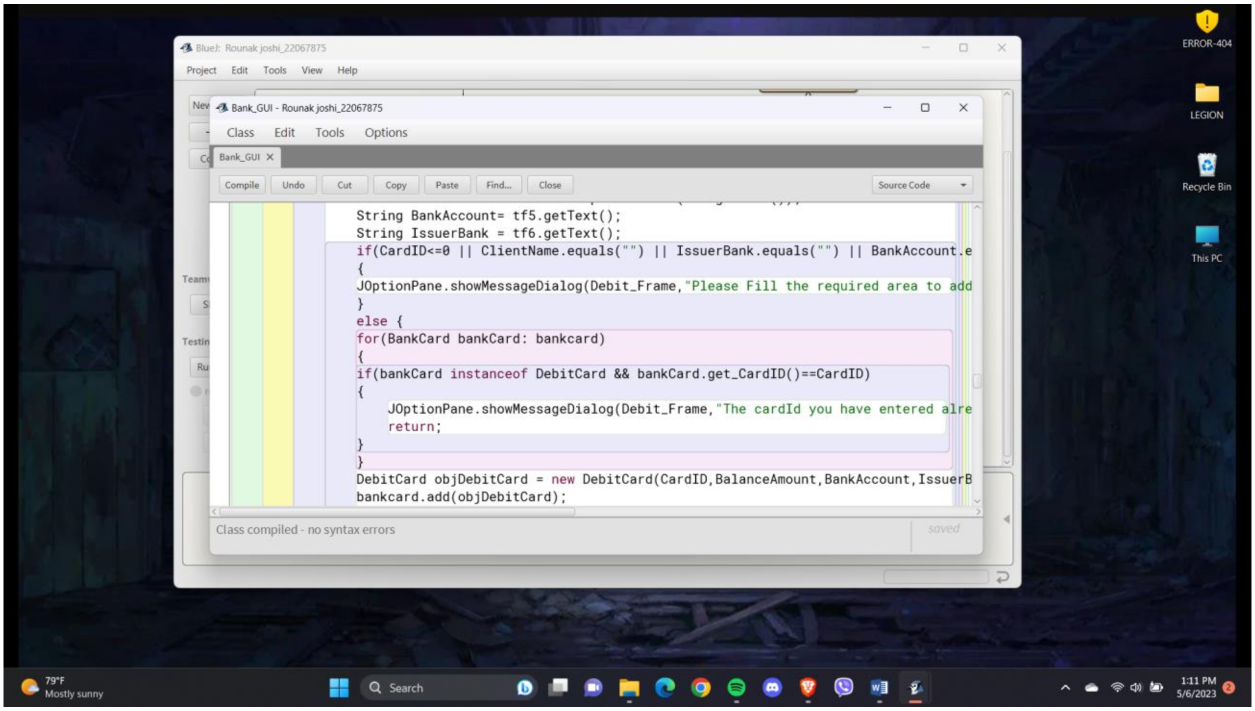
Evidence of error:


BlueJ: Rounak joshi_22067875
Project Edit Tools View Help
Bank_GUI - Rounak joshi_22067875
Class Edit Tools Options
Bank_GUI
Compile Undo Cut Copy Paste Find... Close
Source Code
saved
Errors: 1

```
String ClientName = tf2.getText();
int PINnumber = Integer.parseInt(tf3.getText());
double BalanceAmount = Double.parseDouble(tf4.getText());
String BankAccount= tf5.getText();
String IssuerBank = tf6.getText();
if(CardID<=0 || ClientName.equals("") || IssuerBank.equals("") || BankAccount.e
{
JOptionPane.showMessageDialog(Debit_Frame,"Please Fill the required area to add
}
else {
for(BankCard bankCard: bankcard)
{
if(bankCard instanceof DebitCard && bankCard.get_CardID()=CardID)
{
    JOptionPane.showMessageDialog(Debit_Frame,"The cardId you have entered alre
    return;
}
```

FIGURE 33 SCREENSHOT OF LOGICAL ERROR OCCURRING

Image Description- The above image shows logical error occurring as only one = symbol was used instead of two which causes to replace the value rather than checking if both values are equal.



```
        String BankAccount= tf5.getText();
        String IssuerBank = tf6.getText();
        if(CardID<=0 || ClientName.equals("") || IssuerBank.equals("") || BankAccount.e
        {
        JOptionPane.showMessageDialog(Debit_Frame,"Please Fill the required area to add
        }
        else {
        for(BankCard bankCard: bankcard)
        {
        if(bankCard instanceof DebitCard && bankCard.get_CardID()==CardID)
        {
            JOptionPane.showMessageDialog(Debit_Frame,"The cardId you have entered alre
            return;
        }
        }
        DebitCard objDebitCard = new DebitCard(CardID,BalanceAmount,BankAccount,IssuerB
        bankcard.add(objDebitCard);
```

Class compiled - no syntax errors

FIGURE 34 *SCREENSHOT OF LOGICAL ERROR BEING SOLVED*

Image description – the above image shows logical error being solved using two equals to sign(=) instead of only one .

7. CONCLUSION

The following coursework has been very interesting as well confusing at some part and hard. The coursework has helped me to learn about a lot of things such as using of Arraylist, creating a GUI for a system, use of upcasting as well as downcasting, about event in java etc. Learning using of various tools such as draw.io for creating class diagrams, moqups for creating wireframes of the GUI have been very helpful and can help a lot in the future as well. The project has been very interesting as we knowing about using of various panels was very confusing at the beginning. Through the project we were able to learn how actionlistener works how the buttons are added and used through it, how to use Arraylist and add objects to the Arraylist.

There were also some difficulties and errors that I faced during doing this coursework. Such as when the code had compiled yet was not working and the object was not being added to the Arraylist which after a thoroughly examing was caused due an extra bracket creating the if case to working even when it was wrong. There were also difficulties during trying to run the code through command prompt as the code and jave file to not open which was solved after reinstalling the jdk file. There always times when the code haad to be changed the set bounds had to be changed and adjusted which took a lot of time. There were always many syntax errors which had to be done and the code had to be done carefully and slowly as a letter casing was very important for the GUI to work properly.

In the end the coursework taught a lot about programming and java. It taught a lot about how a single bracket can damage the whole code and the coding should be done slowly and properly.

This was my conclusion on this coursework.

8.REFERENCES

Computer Hope. (06/02/2020). *Draw.io.* Salt lake City, Utah: https://www.computerhope.com/jargon/d/drawio.htm.

Pietroluongo, L. (September 19, 2022). *Moqups.* San Francisco, California: September 19, 2022.

Rouse, M. (10 August, 2022). *What Does Microsoft Word Mean.* 25 Old Broad Street London: https://www.techopedia.com/.

Rouse, M. (10 October, 2013). *What Does BlueJ Mean?* 25 Old Broad Street London : https://www.techopedia.com/definition/29530/bluej.

9. APPENDIX

9.2. CODE OF BANK CARD GUI

```
import javax.swing.*;

import java.awt.*;

import java.awt.event.*;

import java.awt.Color;

import java.util.ArrayList;


public class Bank_GUI implements ActionListener

{

    // creating and calling arraylist for Bank card class

    private ArrayList<BankCard> bankcard = new ArrayList<BankCard>();


    //declaring variables

    private JFrame Debit_Frame, WithDraw_Frame;

    private JFrame Credit_Frame, CreditLimit_Frame;


    private JPanel Debit_Panel, Credit_Panel;

    private JPanel Withdraw_Panel, CreditLimit_Panel;


    private JTextField tf1, tf2, tf3, tf4, tf5, tf6, tf7, tf8;

    private JTextField ctf1, ctf2, ctf3, ctf4, ctf5, ctf6, ctf7, ctf8, ctf9, ctf10;


    private JButton DisplayButton, ClearButton, WithdrawButton, SwitchButton, AddCardButton,
backButton, WithdrawClearButton, WithdrawAmountButton,WithdrawDisplayButton;

    private JButton CDisplayButton,CClearButton ,CreditlimitButton , Switch2DebitButton,
AddCCardButton, CancelCardButton, BackButton, clearButton, SetLimitButton,
LimitDisplayButton;
```

```
    private JLabel title1, CardID, ClientName, IssuerBank, BankAccount, PINnumber,
BalanceAmount,WithdrawTitle, withdrawalAmount, dateOfWithdrawal,WithdrawCardID;
    private JLabel CreditCard,CClientName , CCardID, CIssuerBank,CBankAccount ,
CBalanceAmount, CreditLimitTitle, CVCnumber, CreditLimit, GracePeriod, InterestRate
,ExpirationDate, LimitCardID;

    private JComboBox DayCombobox,MonthCombobox,YearCombobox;
    private JComboBox daycombobox,monthcombobox,yearcombobox;

    public Bank_GUI()
    {

        Debit_Frame = new JFrame("Rounak_Joshi_22067875");
        Debit_Panel = new JPanel();

        tf1 = new JTextField();
        tf2 = new JTextField();
        tf3 = new JTextField();
        tf4 = new JTextField();
        tf5 = new JTextField();
        tf6 = new JTextField();

        DisplayButton = new JButton("Display");
        ClearButton = new JButton("Clear");
        WithdrawButton = new JButton("Withdraw Amount");
        SwitchButton = new JButton("Switch to Credit Card");
        AddCardButton = new JButton("Add Debit Card");

        title1 = new JLabel("Debit Card");
```

```
CardID = new JLabel("Card ID");

ClientName = new JLabel("Client Name");

IssuerBank = new JLabel("Issuer Bank");

BankAccount = new JLabel("Bank Account");

PINnumber = new JLabel("PIN Number");

BalanceAmount = new JLabel("Balance Amount");


//setting bounds to the the objects

title1.setBounds(411,17,124,36);

CardID.setBounds(26,92,55,22);

ClientName.setBounds(26,147,88,21);

PINnumber.setBounds(26,200,92,22);

BalanceAmount.setBounds(448,83,117,22);

BankAccount.setBounds(448,143,98,22);

IssuerBank.setBounds(448,195,84,22);


tf1.setBounds(177,87,180,27);

tf2.setBounds(177,141,180,27);

tf3.setBounds(177,195,180,27);

tf4.setBounds(631,87,180,27);

tf5.setBounds(631,138,180,27);

tf6.setBounds(631,189,180,27);


DisplayButton.setBounds(700,300,120,32);

ClearButton.setBounds(850,300,120,32);

WithdrawButton.setBounds(500,250,140,32);

SwitchButton.setBounds(21,300,235,32);

AddCardButton.setBounds(320,250,140,32);
```

```
//adding the objects to the panel
Debit_Panel.add(title1);
Debit_Panel.add(CardID);
Debit_Panel.add(ClientName);
Debit_Panel.add(IssuerBank);
Debit_Panel.add(BankAccount);
Debit_Panel.add(PINnumber);
Debit_Panel.add(BalanceAmount);

Debit_Panel.add(tf1);
Debit_Panel.add(tf2);
Debit_Panel.add(tf3);
Debit_Panel.add(tf4);
Debit_Panel.add(tf5);
Debit_Panel.add(tf6);

Debit_Panel.add(DisplayButton);
Debit_Panel.add(ClearButton);
Debit_Panel.add(WithdrawButton);
Debit_Panel.add(SwitchButton);
Debit_Panel.add(AddCardButton);

//Adding actionlistener to the buttons
DisplayButton.addActionListener(this);
ClearButton.addActionListener(this);
WithdrawButton.addActionListener(this);
SwitchButton.addActionListener(this);
AddCardButton.addActionListener(this);
```

```
//seeting the background to the frame and the buttons

title1.setFont(new Font("Arial Black",Font.PLAIN,20));

Debit_Panel.setBackground(new Color(165,202,205));

DisplayButton.setBackground(new Color(246,176,146));

ClearButton.setBackground(new Color(246,176,146));

WithdrawButton.setBackground(new Color(246,176,146));

SwitchButton.setBackground(new Color(246,176,146));

AddCardButton.setBackground(new Color(246,176,146));


Debit_Frame.add(Debit_Panel);

Debit_Frame.setSize(1000,400);

Debit_Panel.setLayout(null);

Debit_Frame.setVisible(true);

Debit_Frame.setResizable(false);

Debit_Frame.setDefaultCloseOperation(Debit_Frame.EXIT_ON_CLOSE);


//creating a new frame, objects and panel for the withdrawing amount

WithDraw_Frame = new JFrame("Rounak_Joshi_22067875");

Withdraw_Panel = new JPanel();


WithdrawTitle = new JLabel("Withdraw Amount");

withdrawalAmount = new JLabel("Withdrawal Amount");

dateOfWithdrawal = new JLabel("Date of Withdrawal");

WithdrawCardID = new JLabel("Card ID");


tf7 = new JTextField();

tf8 = new JTextField();


backButton = new JButton("Back");
```

```
        WithdrawClearButton = new JButton("Clear");

        WithdrawAmountButton = new JButton("Withdraw Amount");

        WithdrawDisplayButton = new JButton("Display");


        String[] DD =
{"1","2","3","4","5","6","7","8","9","10","11","12","13","14","15","16","17","18","19","20","21"
,"22","23","24","25","26","27","28","29","30","31"};

        DayCombobox = new JComboBox(DD);

        String[] MM =
{"January","february","March","April","May","June","July","August","September","October","N
ovember","December"};

        MonthCombobox = new JComboBox(MM);

        String[] YYYY = {"2023"};

        YearCombobox = new JComboBox(YYYY);


        //setting the bounds to the objects

        WithdrawTitle.setBounds(168,9,198,36);

        WithdrawCardID.setBounds(10,75,130,22);

        withdrawalAmount.setBounds(10,127,130,22);

        dateOfWithdrawal.setBounds(10,170,130,22);


        tf7.setBounds(150,75,180,27);

        tf8.setBounds(150,127,180,27);


        DayCombobox.setBounds(150,170,57,27);

        MonthCombobox.setBounds(226,170,57,27);

        YearCombobox.setBounds(302,170,57,27);


        WithdrawAmountButton.setBounds(148,246,141,32);

        backButton.setBounds(10,300,120,32);
```

```
WithdrawDisplayButton.setBounds(160,300,120,32);
WithdrawClearButton.setBounds(310,300,120,32);

//adding the objects to the withdraw panel
Withdraw_Panel.add(WithdrawTitle);
Withdraw_Panel.add(WithdrawCardID);
Withdraw_Panel.add(withdrawalAmount);
Withdraw_Panel.add(dateOfWithdrawal);

Withdraw_Panel.add(backButton);
Withdraw_Panel.add(WithdrawClearButton);
Withdraw_Panel.add(WithdrawAmountButton);
Withdraw_Panel.add(WithdrawDisplayButton);

Withdraw_Panel.add(DayCombobox);
Withdraw_Panel.add(MonthCombobox);
Withdraw_Panel.add(YearCombobox);

Withdraw_Panel.add(tf7);
Withdraw_Panel.add(tf8);



//adding the actionlistener to the buttons
backButton.addActionListener(this);
WithdrawClearButton.addActionListener(this);
WithdrawAmountButton.addActionListener(this);
WithdrawDisplayButton.addActionListener(this);

//seetting the background color of the panel and button
```

```
Withdraw_Panel.setBackground(new Color(165,202,205));
WithdrawTitle.setFont(new Font("Arial Black",Font.PLAIN,18));
backButton.setBackground(new Color(246,176,146));
WithdrawClearButton.setBackground(new Color(246,176,146));
WithdrawAmountButton.setBackground(new Color(246,176,146));
WithdrawDisplayButton.setBackground(new Color(246,176,146));

WithDraw_Frame.add(Withdraw_Panel);
WithDraw_Frame.setSize(500,400);
Withdraw_Panel.setLayout(null);
WithDraw_Frame.setVisible(false);
WithDraw_Frame.setResizable(false);
WithDraw_Frame.setDefaultCloseOperation(WithDraw_Frame.EXIT_ON_CLOSE);

//creating a new frame, objects and panel for the credit card
Credit_Frame = new JFrame("Rounak_Joshi_22067875");
Credit_Panel = new JPanel();

ctf1 = new JTextField();
ctf2 = new JTextField();
ctf3 = new JTextField();
ctf4 = new JTextField();
ctf5 = new JTextField();
ctf6 = new JTextField();
ctf7 = new JTextField();

CDisplayButton = new JButton("Display");
CClearButton = new JButton("Clear");
CreditlimitButton = new JButton("Set Credit Limit");
```

```
    Switch2DebitButton = new JButton("Switch to Debit Card");
    AddCCardButton = new JButton("Add Credit Card");
    CancelCardButton = new JButton("Cancel Credit Card");

    CreditCard = new JLabel("Credit Card");
    CClientName = new JLabel("Client Name");
    CCardID = new JLabel("Card ID");
    CIssuerBank = new JLabel("Issuer Bank");
    CBankAccount = new JLabel("Bank Account");
    CBalanceAmount = new JLabel("Balance Amount");
    CVCnumber = new JLabel("CVC Number");
    InterestRate = new JLabel("Interest Rate");
    ExpirationDate = new JLabel("Expiration Date");

    String[] Day = {"1","2","3","4","5","6","7","8","9","10","11","12","13","14","15","16","17","18","19","20","21","22","23","24","25","26","27","28","29","30","31"};
    daycombobox = new JComboBox(Day);
    String[] Month = {"January","february","March","April","May","June","July","August","September","October","November","December"};
    monthcombobox = new JComboBox(Month);
    String[] Year = {"2023","2024","2025","2026","2027"};
    yearcombobox = new JComboBox(Year);

    //setting the bounds to the objects
    CreditCard.setBounds(411,19,133,36);
    CClientName.setBounds(36,73,88,22);
    CCardID.setBounds(36,124,55,22);
    CIssuerBank.setBounds(36,175,84,22);
```

```
InterestRate.setBounds(36,226,92,23);
CBankAccount .setBounds(418,73,98,22);
CVCnumber.setBounds(418,128,88,22);
CBalanceAmount.setBounds(418,175,117,22);
ExpirationDate.setBounds(418,231,109,23);

ctf1.setBounds(181,73,180,27);
ctf2.setBounds(181,124,180,27);
ctf3.setBounds(181,175,180,27);
ctf4.setBounds(181,226,180,26);
ctf5.setBounds(592,73,180,27);
ctf6.setBounds(592,124,180,27);
ctf7.setBounds(592,175,180,27);

CDisplayButton.setBounds(820,396,120,32);
CClearButton.setBounds(680,396,120,32);
CreditlimitButton.setBounds(163,288,156,32);
Switch2DebitButton.setBounds(14,396,180,32);
AddCCardButton.setBounds(361,288,156,32);
CancelCardButton.setBounds(559,288,156,32);

daycombobox.setBounds(592,226,58,33);
monthcombobox.setBounds(667,226,58,33);
yearcombobox.setBounds(742,226,58,33);

//adding the objects to the panel
Credit_Panel.add(CreditCard);
Credit_Panel.add(CClientName);
Credit_Panel.add(CCardID);
Credit_Panel.add(CIssuerBank);
Credit_Panel.add(CBankAccount );
Credit_Panel.add(CVCnumber);
Credit_Panel.add(CBalanceAmount);
Credit_Panel.add(InterestRate);
Credit_Panel.add(ExpirationDate);
Credit_Panel.add(ctf1);
Credit_Panel.add(ctf2);
Credit_Panel.add(ctf3);
Credit_Panel.add(ctf4);
Credit_Panel.add(ctf5);
Credit_Panel.add(ctf6);
Credit_Panel.add(ctf7);
Credit_Panel.add(CDisplayButton);
Credit_Panel.add(CClearButton);
Credit_Panel.add(CreditlimitButton);
Credit_Panel.add(Switch2DebitButton);
Credit_Panel.add(AddCCardButton);
Credit_Panel.add(CancelCardButton);
Credit_Panel.add(daycombobox);
Credit_Panel.add(monthcombobox);
Credit_Panel.add(yearcombobox);

//adding actionlistener to the buttons
CDisplayButton.addActionListener(this);
CClearButton.addActionListener(this);
CreditlimitButton.addActionListener(this);
Switch2DebitButton.addActionListener(this);
AddCCardButton.addActionListener(this);
```

```
CancelCardButton.addActionListener(this);


//setting the background of the panel and the buttons
Credit_Panel.setBackground(new Color(165,202,205));
CreditCard.setFont(new Font("Arial Black",Font.PLAIN,20));
CDisplayButton.setBackground(new Color(246,176,146));
CClearButton.setBackground(new Color(246,176,146));
CreditlimitButton.setBackground(new Color(246,176,146));
Switch2DebitButton.setBackground(new Color(246,176,146));
AddCCardButton.setBackground(new Color(246,176,146));
CancelCardButton.setBackground(new Color(246,176,146));


Credit_Frame.add(Credit_Panel);
Credit_Frame.setSize(1000,500);
Credit_Panel.setLayout(null);
Credit_Frame.setVisible(false);
Credit_Frame.setResizable(false);
Credit_Frame.setDefaultCloseOperation(Credit_Frame.EXIT_ON_CLOSE);


//creating a new frame, objects and panel for the credit limit
CreditLimit_Frame = new JFrame("Rounak_Joshi_22067875");
CreditLimit_Panel = new JPanel();


CreditLimitTitle = new JLabel("Set Credit limit");
CreditLimit = new JLabel("Credit Limit");
GracePeriod = new JLabel("Grace Period");
LimitCardID = new JLabel("Card ID");


ctf8= new JTextField();
```

```
ctf9 = new JTextField();

ctf10 = new JTextField();




BackButton = new JButton("Back");

clearButton= new JButton("Clear");

SetLimitButton= new JButton("Set Credit Limit");

LimitDisplayButton = new JButton("Display");


//setting the bounds of the objects

CreditLimitTitle.setBounds(176,29,204,37);


LimitCardID.setBounds(15,127,81,23);

CreditLimit.setBounds(15,217,76,23);

GracePeriod.setBounds(15,172,94,23);


ctf8.setBounds(176,122,180,28);

ctf9.setBounds(176,172,180,28);

ctf10.setBounds(176,222,180,28);


BackButton.setBounds(6,310,120,33);

clearButton.setBounds(350,310,120,33);

SetLimitButton.setBounds(170,270,156,33);

LimitDisplayButton.setBounds(180,310,120,33);


//adding the objects to credit limit panel

CreditLimit_Panel.add(CreditLimitTitle);

CreditLimit_Panel.add(CreditLimit);

CreditLimit_Panel.add(GracePeriod);
```

```
        CreditLimit_Panel.add(LimitCardID);

        CreditLimit_Panel.add(ctf8);
        CreditLimit_Panel.add(ctf9);
        CreditLimit_Panel.add(ctf10);

        CreditLimit_Panel.add(BackButton);
        CreditLimit_Panel.add(clearButton);
        CreditLimit_Panel.add(SetLimitButton);
        CreditLimit_Panel.add(LimitDisplayButton);

        //adding actionlistener to the button
        BackButton.addActionListener(this);
        clearButton.addActionListener(this);
        SetLimitButton.addActionListener(this);
        LimitDisplayButton.addActionListener(this);

        //setting the background of the panel and buttons
        CreditLimit_Panel.setBackground(new Color(165,202,205));
        CreditLimitTitle.setFont(new Font("Arial Black",Font.PLAIN,18));
        BackButton.setBackground(new Color(246,176,146));
        clearButton.setBackground(new Color(246,176,146));
        SetLimitButton.setBackground(new Color(246,176,146));
        LimitDisplayButton.setBackground(new Color(246,176,146));

        CreditLimit_Frame.add(CreditLimit_Panel);
        CreditLimit_Frame.setSize(500,400);
        CreditLimit_Panel.setLayout(null);
        CreditLimit_Frame.setVisible(false);
        CreditLimit_Frame.setResizable(false);
        CreditLimit_Frame.setDefaultCloseOperation(CreditLimit_Frame.EXIT_ON_CLOSE);
    }

    public void actionPerformed(ActionEvent cw)
    {
        if(cw.getSource()==DisplayButton)
        {
            for(BankCard bankCard: bankcard)
            {
                if(bankCard instanceof DebitCard)
                {
                    DebitCard debitcard = (DebitCard) bankCard;
                    debitcard.display();
                }
            }
        }

        if(cw.getSource()==AddCardButton)
        {
            try
            {
                int CardID = Integer.parseInt(tf1.getText());
                String ClientName = tf2.getText();
                int PINnumber = Integer.parseInt(tf3.getText());
                double BalanceAmount = Double.parseDouble(tf4.getText());
                String BankAccount= tf5.getText();
                String IssuerBank = tf6.getText();
                if(CardID<=0 || ClientName.equals("") || IssuerBank.equals("") ||
        BankAccount.equals("") || PINnumber<=0 || BalanceAmount<0)
                {
                    JOptionPane.showMessageDialog(Debit_Frame,"Please Fill the required area to add
        the card","ERROR",JOptionPane.ERROR_MESSAGE);
                }
                else
                {
                    for(BankCard bankCard: bankcard)
                    {
                      if(bankCard instanceof DebitCard && bankCard.get_CardID()==CardID)
                      {
                        JOptionPane.showMessageDialog(Debit_Frame,"The cardId you have entered
        already exists","Error",JOptionPane.ERROR_MESSAGE);
                        return;
                      }
                    }
                    DebitCard objDebitCard = new
        DebitCard(CardID,BalanceAmount,BankAccount,IssuerBank,ClientName,PINnumber);
                    bankcard.add(objDebitCard);
                    JOptionPane.showMessageDialog(Debit_Frame,"The card has been added
        successfully","Successful",JOptionPane.INFORMATION_MESSAGE);
                }
            }
            catch(NumberFormatException cw1)
            {
                JOptionPane.showMessageDialog(Debit_Frame,"Please enter valid information to the
        areas","ERROR",JOptionPane.ERROR_MESSAGE);
            }
        }
        if(cw.getSource()==ClearButton)

        {

            tf1.setText("");

            tf2.setText("");

            tf3.setText("");

            tf4.setText("");

            tf5.setText("");

            tf6.setText("");

        }


        if(cw.getSource()==WithdrawButton)

        {

            WithDraw_Frame.setVisible(true);

            Debit_Frame.setVisible(false);

        }

        //

        if(cw.getSource()==WithdrawAmountButton)

        {

            try

            {

                int CardID = Integer.parseInt(tf1.getText());

                int withdrawalAmount = Integer.parseInt(tf8.getText());

                double BalanceAmount = Double.parseDouble(tf4.getText());

                String dateOfWithdrawal = DayCombobox.getSelectedItem().toString()+"-"+
        MonthCombobox.getSelectedItem().toString()+"-"+YearCombobox.getSelectedItem().toString();

                boolean cardFound = false;

                for(BankCard bankCard: bankcard)

                {

                    if(bankCard instanceof DebitCard)
                    {
                      DebitCard debitcard = (DebitCard) bankCard;
                      if(debitcard.get_CardID()== CardID)
                      {
                        cardFound = true;
                        int pinNumber = Integer.parseInt(JOptionPane.showInputDialog("Please enter
        your pin number"));
                        if(pinNumber == debitcard.get_PINnumber())
                        {
                          if(withdrawalAmount>debitcard.get_BalanceAmount())
                          {
                            JOptionPane.showMessageDialog(WithDraw_Frame,"Transaction failed
        due to insufficient balance","ERROR",JOptionPane.ERROR_MESSAGE);
                            return;
                          }
                          debitcard.withdraw(withdrawalAmount,dateOfWithdrawal,pinNumber);
                          JOptionPane.showMessageDialog(WithDraw_Frame,"The amount has been
        withdrawn successfully. Withdrawal
        Amount:"+withdrawalAmount,"Successful",JOptionPane.INFORMATION_MESSAGE);
                        }
                        else
                        {
                          JOptionPane.showMessageDialog(WithDraw_Frame,"The PIN number you
        have entered is incorrect","ERROR",JOptionPane.ERROR_MESSAGE);
                        }
                      }
                    }
                }
                if(!cardFound)
                {
                JOptionPane.showMessageDialog(WithDraw_Frame,"Invalid card ID 
entered","ERROR",JOptionPane.ERROR_MESSAGE);
            }
        }
        catch(NumberFormatException cw2)
        {
            JOptionPane.showMessageDialog(WithDraw_Frame,"Please enter valid information to 
the areas","ERROR",JOptionPane.ERROR_MESSAGE);
        }
    }



    if(cw.getSource()==backButton)
    {
        WithDraw_Frame.setVisible(false);
        Debit_Frame.setVisible(true);
    }
    //
    if(cw.getSource()==WithdrawClearButton)
    {
        tf7.setText("");
        tf8.setText("");
        DayCombobox.setSelectedIndex(0);
        MonthCombobox.setSelectedIndex(0);
        YearCombobox.setSelectedIndex(0);
    }

    if(cw.getSource()==WithdrawDisplayButton)
    {
```

```
    for(BankCard bankCard: bankcard)
    {
        if(bankCard instanceof DebitCard)
        {
            DebitCard debitcard = (DebitCard) bankCard;
            debitcard.display();
        }
    }
}

if(cw.getSource()==SwitchButton)
{
    Credit_Frame.setVisible(true);
    Debit_Frame.setVisible(false);
}
//
if(cw.getSource()==CDisplayButton)
{
    for(BankCard bankCard: bankcard)
    {
        if(bankCard instanceof CreditCard)
        {
            CreditCard creditcard = (CreditCard) bankCard;
            creditcard.display();
        }
    }
}

if(cw.getSource()==AddCCardButton)
```

```
    {
        try
        {
            String CClientName = ctf1.getText();
            int CCardID =  Integer.parseInt(ctf2.getText());
            String CIssuerBank = ctf3.getText();
            double InterestRate = Double.parseDouble(ctf4.getText());
            String CBankAccount = ctf5.getText();
            int CVCnumber = Integer.parseInt(ctf6.getText());
            double CBalanceAmount = Double.parseDouble(ctf7.getText());
            String ExpirationDate = daycombobox.getSelectedItem().toString()+"-"+ monthcombobox.getSelectedItem().toString()+"-"+yearcombobox.getSelectedItem();
            if(CCardID<=0 || CClientName.equals("") || CIssuerBank.equals("") || CBankAccount.equals("") || CBalanceAmount<0 || InterestRate<=0|| CVCnumber<=0)
            {
                JOptionPane.showMessageDialog(Credit_Frame,"Please Fill the required area to add the card","ERROR",JOptionPane.ERROR_MESSAGE);
            }
            else
            {
                for(BankCard bankCard: bankcard)
                {
                    if(bankCard instanceof CreditCard && bankCard.get_CardID()== CCardID)
                    {
                        JOptionPane.showMessageDialog(Credit_Frame,"The cardId you have entered already exists","Error",JOptionPane.ERROR_MESSAGE);
                        return;
                    }
                }
```

```
          CreditCard objCreditCard = new
CreditCard(CCardID,CBalanceAmount,CBankAccount,CIssuerBank,CClientName,CVCnumber,Int
erestRate,ExpirationDate);

          bankcard.add(objCreditCard);

          JOptionPane.showMessageDialog(Credit_Frame,"The card has been added
successfully","Successful",JOptionPane.INFORMATION_MESSAGE);

        }

      }


      catch(NumberFormatException cw3)

      {

        JOptionPane.showMessageDialog(Credit_Frame,"Please enter valid information to the
areas","ERROR",JOptionPane.ERROR_MESSAGE);

      }

    }


    if(cw.getSource()==CClearButton)

    {

      ctf1.setText("");

      ctf2.setText("");

      ctf3.setText("");

      ctf4.setText("");

      ctf5.setText("");

      ctf6.setText("");

      ctf7.setText("");

      daycombobox.setSelectedIndex(0);

      monthcombobox.setSelectedIndex(0);

      yearcombobox.setSelectedIndex(0);

    }

    //
```

```
if(cw.getSource()==Switch2DebitButton)
{
   Credit_Frame.setVisible(false);
   Debit_Frame.setVisible(true);
}
//
if(cw.getSource()==CreditlimitButton)
{
   Credit_Frame.setVisible(false);
   CreditLimit_Frame.setVisible(true);
}
//
 if(cw.getSource()==SetLimitButton)
{
try
{
   int CCardID = Integer.parseInt(ctf8.getText());
   double CreditLimit = Double.parseDouble(ctf10.getText());
   int GracePeriod = Integer.parseInt(ctf9.getText());
   boolean IsGranted = false;
   for(BankCard bankCard: bankcard)
   {
      if(bankCard instanceof CreditCard)
      {
         CreditCard creditcard = (CreditCard) bankCard;
         if(creditcard.get_CardID() == CCardID)
         {
            creditcard.set_CreditLimit(CreditLimit, GracePeriod);
            IsGranted = true;
            if(CreditLimit > creditcard.get_BalanceAmount()) {
              JOptionPane.showMessageDialog(CreditLimit_Frame, "The following credit limit cannot be granted", "ERROR", JOptionPane.ERROR_MESSAGE);
              return;
            }
            else
            {
              JOptionPane.showMessageDialog(CreditLimit_Frame, "Your new credit limit has been set along with a grace period", "Successful", JOptionPane.INFORMATION_MESSAGE);
            }
          }
        }
      }
      if(!IsGranted)
      {
        JOptionPane.showMessageDialog(CreditLimit_Frame, "The provided CardID does not exist", "ERROR", JOptionPane.ERROR_MESSAGE);
      }
    }
    catch(NumberFormatException cw1)
    {
      JOptionPane.showMessageDialog(CreditLimit_Frame, "Please enter valid information to the areas", "ERROR", JOptionPane.ERROR_MESSAGE);
    }
    }

    if(cw.getSource()==CancelCardButton)
    {
      int CCardID =  Integer.parseInt(ctf2.getText());
```

```
        for(BankCard bankCard: bankcard)
        {
            if(bankCard instanceof CreditCard)
            {
                CreditCard creditcard = (CreditCard) bankCard;
                if(creditcard.get_CardID()==CCardID)
                {
                    creditcard.cancelCreditCard();
                    JOptionPane.showMessageDialog(Credit_Frame,"The following card has been
Cancelled","Successful",JOptionPane.INFORMATION_MESSAGE);
                }
            }
        }
    }

    if(cw.getSource()== clearButton)
    {
        ctf8.setText("");
        ctf9.setText("");
    }
    //
    if(cw.getSource()==BackButton)
    {
        Credit_Frame.setVisible(true);
        CreditLimit_Frame.setVisible(false);
    }

    if(cw.getSource()==LimitDisplayButton)
    {
        for(BankCard bankCard: bankcard)
        {
            if(bankCard instanceof CreditCard)
            {
                CreditCard creditcard = (CreditCard) bankCard;
                creditcard.display();
            }
        }
    }
}

public static void main(String[]args)
{
    new Bank_GUI();
}
}
```