CS4001NI Programming

30% Individual Coursework

2022-23 Autumn

Student Name: Bishal Prasain

London Met ID: 22067995

College ID: NP01CP4A220297

Group: L1C1

Assignment Due Date: Wednesday, May 10, 2023

Assignment Submission Date: Wednesday, May 10, 2023

I confirm that I understand my coursework needs to be submitted online via MySecondTeacher under the relevant module page before the deadline in order for my assignment to be accepted and marked. I am fully aware that late submissions will be treated as non-submission and a marks of zero will be awarded.

Table Of Contents

Introduction 7

Class Diagram:.. 8

1)BankCard Class:.. 8

2) DebitCard Class: 9

3) CreditCard Class: 10

4)Combined Class Diagram:.. 11

5)BankGUI.. 12

Pseudocode: 13

1)BankGUI.. 13

Method Descriptions:.. 28

1)BankGUI:.. 28

2)actionPerformed : 28

3)Add Debit Card:.. 28

4)Add Credit Card:.. 28

5)Withdraw from Debit Card: 29

6)Set the Credit Limit:.. 29

7) Cancel Credit Card:.. 30

8)Display DebitCard:.. 30

9)Display CreditCard: 30

10)Clear:.. 31

Testing.. 32

Test-1.. 32

1)Testing the execution of program using command prompt 32

Test-2 35

1)Testing the functionality of Add DebitCard Button 35

2)Testing the functionality of Add CreditCard Button. 38

3)Testing the functionality of withdraw Button 41

4)Testing the functionality of Set Credit Limit button 45

5)Testing the functionality of Cancel CreditCard Button 50

Test-3 54

1) Testing the functionality of add Debit Card button after entering unsuitable .value for Card Id. 54

2) Testing the functionality of add Credit Card button after entering unsuitable .....value for Card Id. 57

3) Testing the functionality of Add Button of Debit Card after leaving some of the text field empty 60

4) Testing the functionality of withdraw button after entering invalid Card ID 63

Error detection and correction: 67

1)Syntax Error: 67

2)Semantic Error: 69

3)Logical Error: 74

Conclusion 87

References 88

Appendix 89

1.BankCard 89

2.DebitCard 92

3.CreditCard 94

4.BankGUI 97

Table of Figure

Figure 1: Running BankGUI.java using command prompt 33

Figure 2:A GUI after compiling program using command prompt................................. 34

Figure 3: Entering values in the respective text fields for adding Debit Card 36

Figure 4: Figure of showing dialog box after pressing add button of Debit Card 37

Figure 5: Entering values in the respective text fields for adding Credit Card 39

Figure 6:Figure of showing dialog box after pressing add button of Credit Card 40

Figure 7:Figure of showing dialog box after pressing add button of Debit Card 43

Figure 8:Figure of showing dialog box after withdrawing amount by entering respective values in the text field. 44

Figure 9:Figure of showing dialog box after pressing add button of Credit Card 47

Figure 10:Figure of showing dialog box after pressing add button of Credit Card 51

Figure 11: Figure of showing dialog box after setting Credit Limit 52

Figure 12: Figure of showing dialog box after cancelling the Credit Card. 53

Figure 13:Figure of showing dialog box after pressing add button of Credit Card 55

Figure 14:Figure of showing error message after entering unsuitable value of card ID in the text field. 56

Figure 15:Figure of entering values in the respective text fields for adding Credit Card 58

Figure 16: Figure of showing error message after entering unsuitable value of card ID in the text field while adding Credit Card 59

Figure 17:Figure of leaving text field empty while adding Credit Card 61

Figure 18:Figure of showing error message after leaving text field empty while adding Credit Card 62

Figure 19: Figure of adding Debit Card 64

Figure 20:Figure of entering values in the text field for withdrawing amount................. 65

Figure 21: Figure of showing error message after entering wrong Card ID 66

Figure 22:Figure of not closing the codition of if statement 67

Figure 23:Figure of fixing syntax error by closing the condition of if statement 68

Figure 24: Figure of addressing semantic error.. 69

Figure 25:Figure of the outcome created by semantic error.. 70

Figure 26: Correction of semantic error 72
Figure 27:Figure of outcomes after fixing semantic error 73
Figure 28: Screeenshots of block of code before fixing logical error. 74
Figure 29: Figure of setting the Credit Limit 75
Figure 30:Figure of cancelling the Credit Limit 76
Figure 31:Figure of adding credit card again 77
Figure 32: Figure of showing error message after setting Credit Card 78
Figure 33:Figure of successful message all of sudden after having error message 79
Figure 34: Screenshots of Block of code after fixing logical error 81
Figure 35: Figure of setting credit limit 82
Figure 36: Figure of cancelling Credit Card 83
Figure 37: Figure of adding Credit Card. 84
Figure 38:Figure of setting credit limit 85

Table of Tables

Table 1: Bank Card Classdiagram 8
Table 2: DebitCard ClassDiagram 9
Table 3:CreditCard Classdiagram 10
Table 4: Combined Class diagram 11
Table 5: BankGUI Classdiagram 13
Table 6: To check the functionality of Add Debit Card Button. 35
Table 7:To check the functionality of add Credit Card button........ 38
Table 8: To check the functionality of withdraw button........ 42
Table 9:To check the functionality of set Credit Card Button 46
Table 10: To check the functionality of Cancel Credit Card Button 50
Table 11:To check the functionality of add Debit Card button after entering unsuitable values........ 54
Table 12:To check the functionality of add Credit Card button after entering unsuitable values 57
Table 13: To check the functionaliaty of add Debit Card Button after leaving texfields empty 60
Table 14: Checking the functionality of withdraw button after enterong invalid Card ID 63

Introduction

This project basically focuses on creating GUI(Graphical User Interface) that supports in managing system of Bank in a more organized and efficient manner. Banking activities like adding Debit Card,Credit Card,withdrawing amount,setting credit limit, cancelling credit Card,and displaying informations about the cards are showcased precisely .It demostrates the actual working system of Banks in real world.

BlueJ is a software application that provides convinient workspace for creating,maintaining, and accessing software projects in java. Because it is user-friendly,it is very easy to write and edit computer programs using java as programming language.

Java is one of the most popular Object-Oriented Programming languages and Softwares.It was developed by James Gosling in 1911.It is best known for its portibility.It is easy to move a code from one platform to another. (What is Java?, 2023).

Microsoft word is a word processor developed by microsoft. It is a software that has advanced features and enables users to format and edit files and documents in a precise manner.It is one of the most commonly used progams for creating documents,resumes,contracts,etc. (What is MS Word?- Basics, Uses, Features & Questions, 2023).

Snipping tools is used to take screenshots for testing and evidence. It is a system provided software helps user to take screenshots of desired dimension. It is easily to use as it has a advanced feature to take a moment of pictures.

Moqups is a cloud-based visual collaboration software designed with the intention of providing assistance to users to create and verify working sample versions of website ormobile application designs. (Moqups Pricing, Features, Reviews & Alternatives, 2023)
.

Class Diagram:

Class diagram is the structural diagram that explains the structure of a system by showcasing the system’s classes, attributes, operations and the relation between the objects which is often used in Object Oriented Programming.

1)BankCard Class:

BankCard
-cardId : int -clientName : String -issuerBank : String -bankAccount : String -balanceAmount : double
+<<constructor>> BankCard (cardId : int,issuerBank : String, bankAccount : String,balanceAmount : double) +getClientName() : String +setClientName(clientName : String) : String +getBalanceAmount() : double +setBalanceAmount(balanceAmount : double) : double +display() : void

Table 1: Bank Card Classdiagram

2) DebitCard Class:

DebitCard
-pinNumber : int -withdrawalAmount : int -dateofWithdrawal : String -haswithdrawn : boolean
+<<constructor>>DebitCard(cardId : int, issuerBank : String,bankAccount : String,balanceAmount : double, clientName : String, pinNumber : int) +getPinNumber(): int +getWithdrawalAmount() : int +getDateofWithdrawal() : String +getHasWithdrawn() : boolean +setWithdrawalAmount (withdrawalAmount : int) : int +display() : void

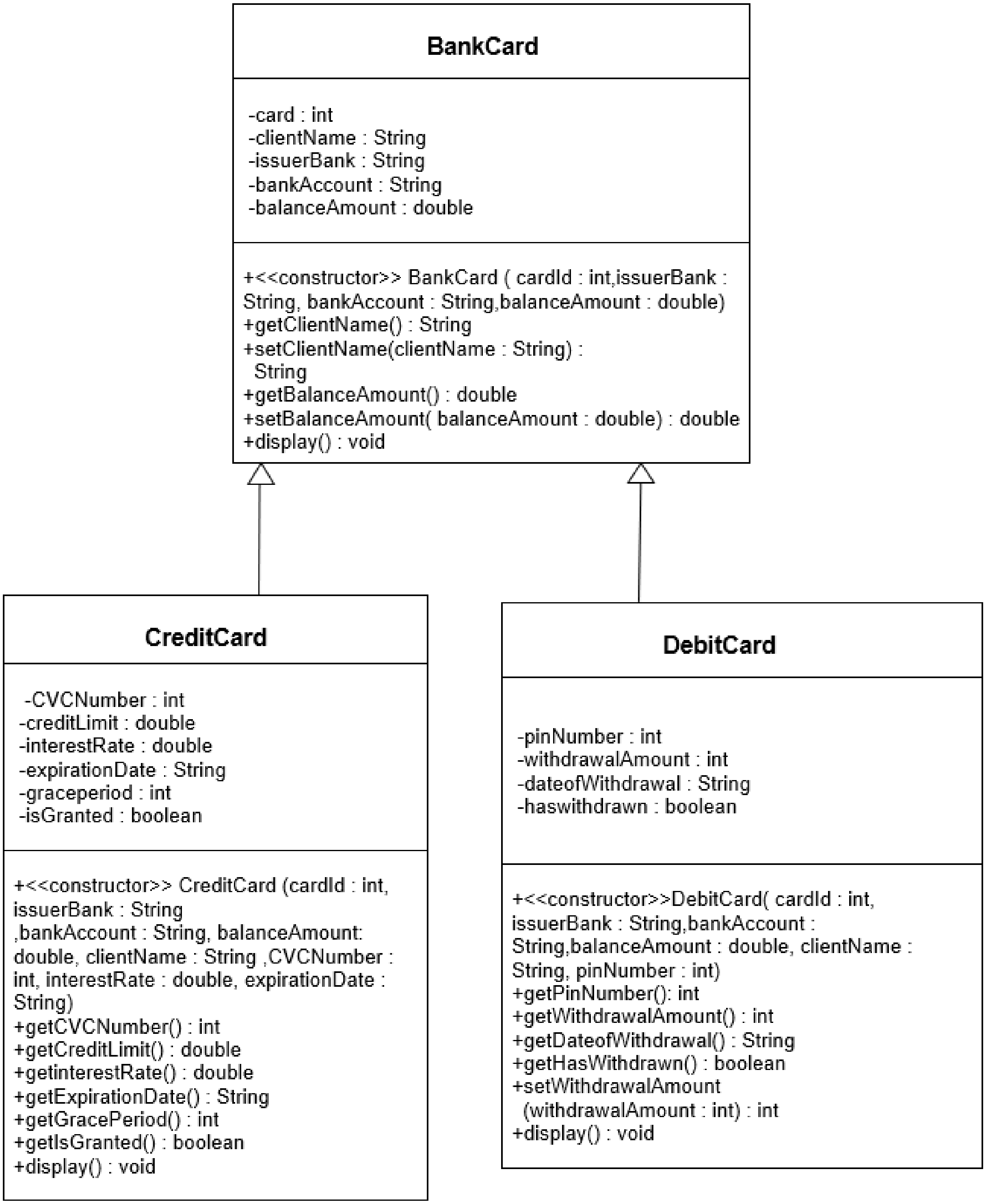
Table 2: DebitCard ClassDiagram

3) CreditCard Class:

CreditCard
-CVCNumber : int -creditLimit : double -interestRate : double -expirationDate : String -graceperiod : int -isGranted : boolean
+<<constructor>> CreditCard (cardId : int, issuerBank : String ,bankAccount : String, balanceAmount: double, clientName : String ,CVCNumber : int, interestRate : double, expirationDate : String) +getCVCNumber() : int +getCreditLimit() : double +getInterestRate() : double +getExpirationDate() : String +getGracePeriod() : int +getIsGranted() : boolean +display() : void

Table 3:CreditCard Classdiagram

4)Combined Class Diagram:



BankCard

-card : int
-clientName : String
-issuerBank : String
-bankAccount : String
-balanceAmount : double

+<<constructor>> BankCard (cardId : int,issuerBank : String, bankAccount : String,balanceAmount : double)
+getClientName() : String
+setClientName(clientName : String) : String
+getBalanceAmount() : double
+setBalanceAmount(balanceAmount : double) : double
+display() : void

CreditCard

-CVCNumber : int
-creditLimit : double
-interestRate : double
-expirationDate : String
-graceperiod : int
-isGranted : boolean

+<<constructor>> CreditCard (cardId : int, issuerBank : String ,bankAccount : String, balanceAmount: double, clientName : String ,CVCNumber : int, interestRate : double, expirationDate : String)
+getCVCNumber() : int
+getCreditLimit() : double
+getinterestRate() : double
+getExpirationDate() : String
+getGracePeriod() : int
+getIsGranted() : boolean
+display() : void

DebitCard

-pinNumber : int
-withdrawalAmount : int
-dateofWithdrawal : String
-haswithdrawn : boolean

+<<constructor>>DebitCard(cardId : int, issuerBank : String,bankAccount : String,balanceAmount : double, clientName : String, pinNumber : int)
+getPinNumber(): int
+getWithdrawalAmount() : int
+getDateofWithdrawal() : String
+getHasWithdrawn() : boolean
+setWithdrawalAmount (withdrawalAmount : int) : int
+display() : void

Table 4: Combined Class diagram

5)BankGUI

BankGUI
- (clientnameLabel,issuerbankLabel,bankaccountLabel,cardidLabel_0,balanceamtLabel,pinnumberLabel,cardidLabel_1,withdrawalamtLabel,withdrawaldateLabel,titleLabel_0,titleLabel_1,titleLabel_2, cvcnumberLabel, interestrateLabel, ExpirationdateLabel, cardidLabel_2, creditlimitLabel, graceperiodLabel) : JLabel - (clientnameTxt, issuerbankTxt, bankaccountTxt, cardidTxt_0, balanceamtTxt, pinnumberTxt, cardidTxt_1, withdrawalamtTxt, cvcnumberTxt, interestrateTxt, cardidTxt_2, creditlimitTxt, graceperiodTxt) : JtextField - (addBtn_0, displayBtn_0, displayBtn_1, addBtn_1, withdrawBtn) : Jbutton - (dayCombo_0, monthCombo_0,yearCombo_0,dayCombo_1,monthCombo_1, yearCombo_1): JComboBox
+(clientName,issuerBank,bankAccount,days_1,months_1,years_1days_0,months_0,years_0,expirationDate, dateOfWithdrawal): String +(BalanceAmount, interestRate, initialAmount, remainingAmount, newCreditLimit): double +(cardId, newgracePeriod, CVCNumber, pinNumber: int +(toAdd): boolean

+<constructor>> BankGUI() +actionPerformed(): void +main(): void

Pseudocode:

1)BankGUI

```
IMPORT required packages and classes
CREATE BankGUI class IMPLEMENTED TO ActionListener
    DECLARE frame as JFRAME
    DECLARE adequate numbers of JLabels, JTextFields, JButtons, JcomboBox for BankGUI
    CREATE an ArrayList ArrayList<BankCard> bankCardsArraylists
    CREATE a constructor BankGUI with no return type

        INITIALIZE frame
        ADD actionListener on all the buttons
        SET bounds to (850,950) and Visibility to true
        SET default close operation to "Exit on close"  and resizable to false
        SET layout to null
        SET background color to grayish-green
        INITIALIZE all the components of JLabel, JTextField, JButtons, JcomboBox of BankGUI
        SET bounds to all components of JLabel, JTextField, JButtons, JcomboBox of BankGUI
```

Table 5: BankGUI Classdiagram

```
        ADD all components of JLabel, JTextField, JButtons,
        JcomboBox on frame

    CREATE a method actionPerformed which takes an ActionEvent object as a
    parameter with no return type

        IF event is generated on clicking Add button of DebitCard
        DO

            IF any of the text fields is empty

                DISPLAY error message
            END IF
            ELSE
              TRY

                GET TEXT as  balanceamtTxt, cardidTxt_0,
                bankaccountTxt,  issuerbankTxt, clientnameTxt,
                pinnumberTxt
                INITIALIZE toADD with boolean datatype to True
                INITIALIZE object named debitCard
                IF bankCardsArraylists is empty
                    CREATE a new DebitCard object with arguments
                    BalanceAmount, cardId, bankAccount, issuerBank,
                    clientName, pinNumber
                    ADD object to an ArrayList
                    DISPLAY successful message
                END IF
                ELSE
```

```
FOR each card in bankCardsArraylists)

    IF card is an object of DebitCard

        TREAT the card as a debitCard
         IFCard ID of debitCard object equals
        entered  Card ID

            DiSPLAY error message
            SET toADD to false
            Break the loop
        END IF
    END IF
        ELSE

            DISPLAY error message
            SET toADD to false
            BREAK the loop

END FOR

IF toADD equals true

    CREATE a new DebitCard object
    with arguments
    BalanceAmount, cardId,
    bankAccount, issuerBank,
    clientName, pinNumber
    DISPLAY  success message
```

```
            END TRY
            CATCH(Number FormatException ex)

                DISPLAY message

        END DO
        END IF



        IF  event is generated on clicking Add button of CreditCard

        DO
             IF any of the text fields is empty

                 DISPLAY error message
             END IF
             ELSE
                 TRY

                   GET TEXT as  balanceamtTxt, cardidTxt_0,
                 bankaccountTxt,  issuerbankTxt, clientnameTxt,
                 pinnumberTxt, cvcnumberTxt, interestRate, expirationDate
                  INITIALIZE toADD with boolean datatype to false
                  INITIALIZE object named creditcard


                 IF bankCardsArraylists is empty
                CREATE a new CrediCard object with arguments
                BalanceAmount, cardId, bankAccount, issuerBank,
                clientName, CVCNumber, interestRate,expirationDate
                ADD object to an ArrayList
                DISPLAY successful message
            ELSE
                FOR each card in bankCardsArraylists
                    IF card is an object of CreditCard
                        TREAT the card as a creditCard
                        IF Card ID of creditCard object equals
                        entered CardID
                            DISPLAY error message
                            SET toADD to false
                            Break the loop
                        END IF
                        ELSE
                          DISPLAY error message
                          SET to add to false
                           BREAK the loop
                        IF(equals true
                            CREATE a new CrediCard object
                            with arguments
                            BalanceAmount, cardId,
                            bankAccount, issuerBank,
                            clientName, CVCNumber,
                            interestRate,expirationDate
                            DISPLAY  success message
                END FOR
                IF toADD equals true


                    CREATE a new CrediCard object with
                    arguments ,BalanceAmount, cardId,
                    bankAccount, issuerBank, clientName,
                    CVCNumber, interestRate,expirationDate
                    ADD object to an ArrayList
          END TRY
          CATCH(Number FormatException ex)

            DISPLAY message

      END DO
      END IF
```

IF event is generated on clicking withdraw button

```
DO
    IF any of the text fields is empty
        DISPLAY error message
    ELSE
        TRY
            GET TEXT as cardidTxt_1, withdrawalamtTxt, pinnumberTxt,
            dateOfWithdrawal
            INITIALIZE availableCard with boolean datatype to false
            IF bankCardsArraylists is empty
                DISPLAY error message
            END IF
            ELSE
                FOR each card in bankCardsArraylists

                    IF( card is an object of DebitCard)
                        TREAT the card as a debitCard
                        IF(Card ID and PIN No. of debitCard object
                        equals entered CardID and PIN No.

                            ASSIGN BalanceAmount with initialAmount
                            with number datatype
                           Call withdraw method of DebitCard Class

                    SET remainingAmount  with double datatype
                    as a difference of initialAmount and
                    withDrawalAmount
                    IF withDrawalAmount is greater than zero

                        IF withDrawalAmount is less or equal to
                        initialAmount
```

```
                            DISPLAY success message
                        END IF
                        ELSE

                            DISPLAY error message


                    END IF
                    ELSE

                        DISPLAY error message


                            DISPLAY error message
                    END ELSE
                    SET availableCard to true
                    BREAK the loop
                END IF
            END IF
          END FOR
          IF availableCard is true

            DISPLAY error message
          END IF


 END TRY
 CATCH(Number FormatException ex)


        DISPLAY message
END IF
```

```
IF event is generated on clicking display button for DebitCard
DO
    TRY
        GETTEXT as clientnameTxt, cardidTxt_0, withdrawalamtTxt,
        bankaccountTxt, pinnumberTxt, issuerBank, BalanceAmount,
        dateOfWithdrawal
        IF bankCardsArraylists is empty
            DISPLAY error message
        END IF
        ELSE
            FOR each card in bankCardsArraylists

                IF( card is an object of DebitCard)

                    TREAT the card as a debitCard
                    CALL display method of DebitCard Class
                    DISPLAY information message
                END IF
                ELSE

                    DISPLAY error message
            END FOR
        END ELSE
    END TRY
    CATCH(NumberFormatException)
        DISLPAY error message
END DO
END IF
```

```
IF event is generated on clicking display button for CreditCard
DO
    TRY
        GETTEXT as clientnameTxt, cardidTxt_2,  bankaccountTxt,
        newCreditLimit, newgracePeriod, CVCNumber, interestRate, issuerBank,
        BalanceAmount, expirationDate
        IF bankCardsArraylists is empty
            DISPLAY error message
        END IF
        ELSE
            FOR each card in bankCardsArraylists

                IF card is an object of CreditCard
```

```
                    TREAT the card as a creditCard
                    CALL display method of CreditCard Class
                    DISPLAY information message
                END IF
                ELSE

                    DISPLAY error message
            END FOR

    END TRY
    CATCH(NumberFormatException)
        DISLPAY error message
END IF
```

```
IF event is generated on clicking set button of Credit Card
DO
    TRY

        GETTEXT as cardidTxt_2, creditlimitTxt, graceperiodTxt
         INITIALIZE availableCard with boolean datatype to false

        IF bankCardsArraylists is empty
            DISPLAY error message
```

```
END IF
ELSE
        FOR each card in bankCardsArraylists

                IF( card is an object of CreditCard)

                        TREAT the card as a creditCard
                         IF cardID of creditCard equals entered cardID

                                SET availableCard to true
                                CALL setCreditLimit method from CreditCard
                                Class
                                IF  newcreditLimit is less or equal to 2.5 times
                                the BalanceAmount of creditCard

                                        DISPLAY information message
                                END IF
                                ELSE

                                        DISPLAY error message

                                BREAK the loop
                        END IF
                END IF
         END FOR
        IF availableCard is true

                DISPLAY error message
        END IF
```

```
            END TRY
           CATCH(Number FormatException ex)

                DISPLAY message
END IF
```

```
IF event is generated on clicking cancel button for CreditCard
DO
    TRY

        GETTEXT as cardId, newCreditLimit, newgracePeriod, CVCNumber
        IF bankCardsArraylists is empty
            DISPLAY error message
        END IF
        ELSE
             FOR each card in bankCardsArraylists

                 IF card is an object of CreditCard
```

```
                    TREAT the card as a creditCard
                    IF cardID of creditCard equals entered cardID

                        CALL cancelCreditcard method from
                        CreditCard Class
                        SET creditLimitTxt  empty
                        SET graceperiodTxt  empty
                        SET cvcnumberTxt empty
                        DISPLAY success message
                    END IF
                    ELSE

                        DISPLAY error message
                END IF
                ELSE

                    DISPLAY error message
            END FOR


    END TRY
    CATCH(NumberFormatException)
        DISPLAY error message
END DO
END IF



IF event is generated on clicking clear button
```

```
DO
    SET balanceamtTxt empty
    SET bankaccountTxt empty
    SET issuerbankTxt empty
    SET clientnameTxt empty
    SET pinnumberTxt empty
    SET cardidTxt_0 empty
    SET cardidTxt_1 empty
    SET cardidTxt_2 empty
    SET clientnameTxt empty
    SET cvcnumberTxt empty
    SET graceperiodTxt empty
    SET interestrateTxt empty
    SET withdrawalamtTxt empty
END DO
END IF
```

Method Descriptions:

1)BankGUI:

Constructor method BanGUI is a special kind of method which initializes and sets properties of different kind of components of GUI like :Button,Label,Textfield,and ComboBox.It sets the layout of those components making it user-friendly.

2)actionPerformed :

Action Performed method is event listener methods that helps to add functionality in the components of GUI such as buttons.It implements the event listener for the Buttons like: Add, Display, Withdraw, Set,Cancel, Clear etc.

3)Add Debit Card:

It is the implementation of event listener for a certain action event(addBtn_0). It adds new Debit Card and handles exceptional ourtcomes . It retrieves input values from the text fields.It checks if the arraylist is empty.If yes, it creates a new DebitCard object with retrieved values and adds the object to the arraylist.If not ,it will loop through arraylist and checks if the object of arraylist is of debit card .if yes,it cast it to DebitCard type and checks if cardID already exists. If so,it displays an error message.If not,it adds new DebitCard to the arrayList and throws a popup as success message.It displays an error message if there comes any exception case.

4)Add Credit Card:

It is similar to the implementation of event listener for add debit card button. It implements event listener for an action event called (addBtn_1). It adds new Credit Card and handles exceptional ourtcomes . It retrieves input values from the text fields.It checks if the arraylist is empty.If yes, it creates a new CreditCard object with retrieved values and adds the object to the arraylist .If not ,it will loop through arraylist and checks if the object of arraylist is an instance of credit card .if yes, it cast it to DebitCard type and checks if cardID already exists. If so,it displays an error message.If not,it adds new Credit to the arrayList and throws a popup as success message.It displays an error message if there comes any exception case.

5)Withdraw from Debit Card:

It implements event listener for withdraw button(withdrawBtn). It retrieves input from text field.It checks validity for array List.If it is empty, it displays an error.If not, it iterates through array list.It checks if the object of an array list is an instance of DebitCard.If so,it is downcasted to DebitCard. It the compares the Pin Number and Card ID with an the entered values.It then retrives the initial amount.It then calls withdraw method from Debit Card Class. If withdrawn amount is less or equal to an initial amount,it displays a success message with remaining amount and other details of transaction.if not,it will show an error message.If the object of an array list is not an instance of DebitCard,it shows error message.It also handles NumberFormatException.

6)Set the Credit Limit:

This method handles event when button (setBtn) is clicked. It retrieves the value entered by the user in the text fields. If array list is empty, it throws an error message.If not,it iterates through array list.It then checks if the object of the array list is an instance of Credit Card.If so, it downcasts to CreditCard type.If not, it shows error message.It checks if the entered Card ID is equal to the Card ID of Credit Card.If so, It calls setCreditLimit method from Credit Card Class.If not, it displays an error message.If the entered Credit Limit is less than or equals to 2.5

times the initial Balance amount, a success message is showcased. If not, it throws error information.It also handles NumberFormatException.

7) Cancel Credit Card:

This method implements event listener for an action event called (cancelBtn). . If array list is empty, it throws an error message.If not,it iterates through array list.It then checks if the object of the array list is an instance of Credit Card.If so, it downcasts to CreditCard type.If not, it shows error message.It checks if the entered Card ID is equal to the Card ID of Credit Card.If so, It calls cancelCreditcardmethod from Credit Card Class. It then set Credit Limit, grace period ,and CVC number to 0. If not, it displays an error message. It handles exception accordingly

8)Display DebitCard:

It displays every details of DebitCard and handles exception accordingly. It retrieves input from the text field. If array list is empty, it throws an error message.If not,it itterates through array list.It then checks if the object of the array list is an instance of Debit Card.If so, it downcasts to DebitCard type. It then calls display method of DebitCard Class. It shows every values entered for DebitCard.If not,it shows error message.

9)Display CreditCard:

It displays every details of CreditCard and handles exception accordingly. It implements event listener for an action event (displayBtn_1). It retrieves input from the text field. If array list is empty, it throws an error message.If not,it iterates through array list.It then checks if the object of the array list is an instance of Credit Card.If so, it downcasts to CreditCard type. It then calls display method of Credit Class. It shows every values entered for Credit Card.If its not an instance of Credit Card,it shows an error message.

10)Clear:

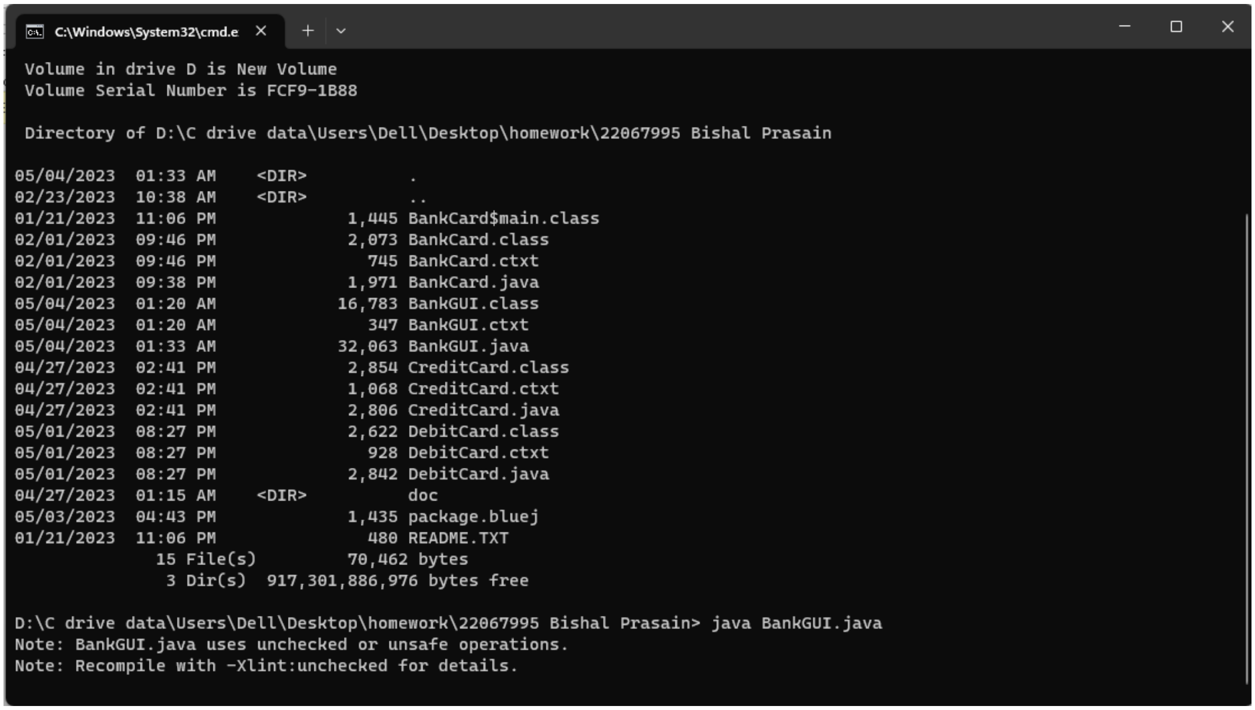
It implements event listener for an action event(clearBtn).It retrieves all input from the text field and set them to an empty string.

Testing

Test-1

1)Testing the execution of program using command prompt

Test No.	1
Objective:	To check the execution of program using command prompt.
Action:	Code was written and program was run using command prompt.
Expected Result:	Program would run successfully.
Actual Result:	Program was run successfully.
Conclusion:	The test is successful.



```
 Volume in drive D is New Volume
 Volume Serial Number is FCF9-1B88

 Directory of D:\C drive data\Users\Dell\Desktop\homework\22067995 Bishal Prasain

05/04/2023  01:33 AM    <DIR>          .
02/23/2023  10:38 AM    <DIR>          ..
01/21/2023  11:06 PM             1,445 BankCard$main.class
02/01/2023  09:46 PM             2,073 BankCard.class
02/01/2023  09:46 PM               745 BankCard.ctxt
02/01/2023  09:38 PM             1,971 BankCard.java
05/04/2023  01:20 AM            16,783 BankGUI.class
05/04/2023  01:20 AM               347 BankGUI.ctxt
05/04/2023  01:33 AM            32,063 BankGUI.java
04/27/2023  02:41 PM             2,854 CreditCard.class
04/27/2023  02:41 PM             1,068 CreditCard.ctxt
04/27/2023  02:41 PM             2,806 CreditCard.java
05/01/2023  08:27 PM             2,622 DebitCard.class
05/01/2023  08:27 PM               928 DebitCard.ctxt
05/01/2023  08:27 PM             2,842 DebitCard.java
04/27/2023  01:15 AM    <DIR>          doc
05/03/2023  04:43 PM             1,435 package.bluej
01/21/2023  11:06 PM               480 README.TXT
              15 File(s)         70,462 bytes
               3 Dir(s)  917,301,886,976 bytes free

D:\C drive data\Users\Dell\Desktop\homework\22067995 Bishal Prasain> java BankGUI.java
Note: BankGUI.java uses unchecked or unsafe operations.
Note: Recompile with -Xlint:unchecked for details.
```

Figure 1: Running BankGUI.java using command prompt

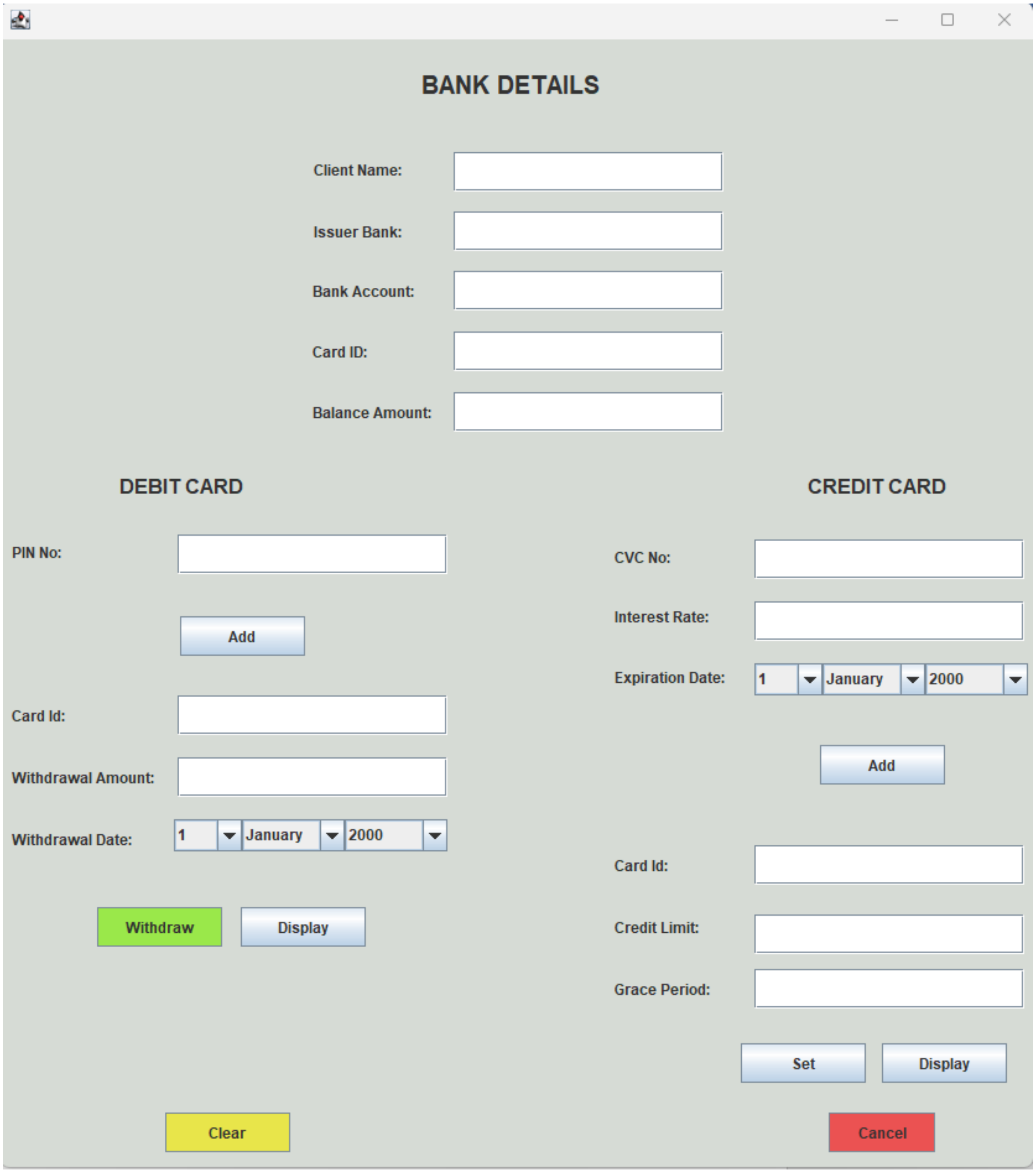


Figure 2:A GUI after compiling program using command prompt

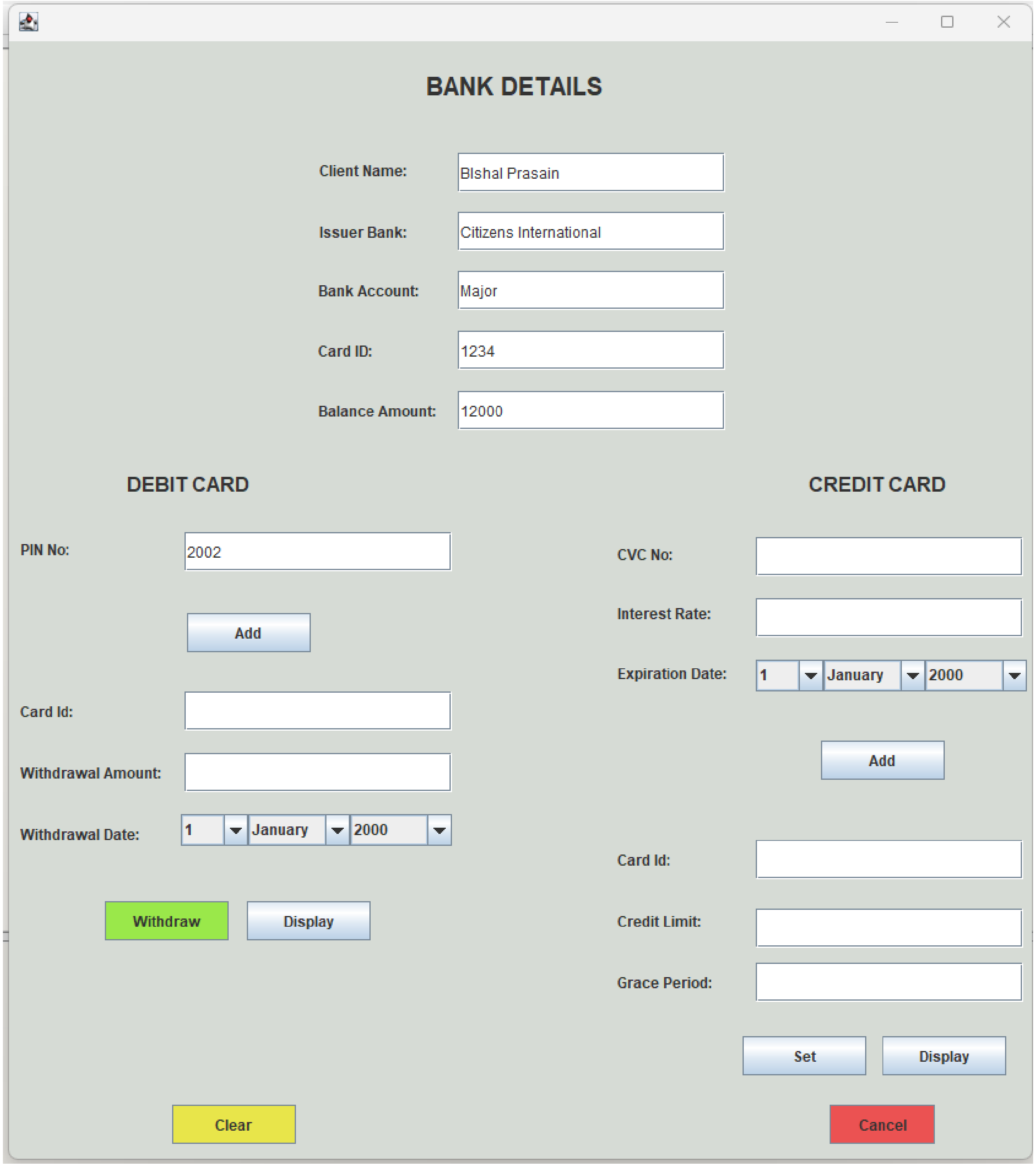
Test-2

1)Testing the functionality of Add DebitCard Button

Test No.	2.1
Objective:	To check the functionality of add button of Debit Card
Action:	Add DebitCard button was pressed by entering following values in a text field: Client Name: Bishal Prasain Issuer Bank: Citizens International Bank Account: Major Card ID: 1234 Balance Amount: 12000 Pin No.: 2002
Expected Result:	An information dialog would pop up informing that the debit card has been successfully added.
Actual Result:	An information dialog was popped up informing that the Debit Card has been Successfully added.
Conclusion:	The test is successful.

Table 6: To check the functionality of Add Debit Card Button.

Output Result:



BANK DETAILS

Client Name: BIshal Prasain

Issuer Bank: Citizens International

Bank Account: Major

Card ID: 1234

Balance Amount: 12000

DEBIT CARD

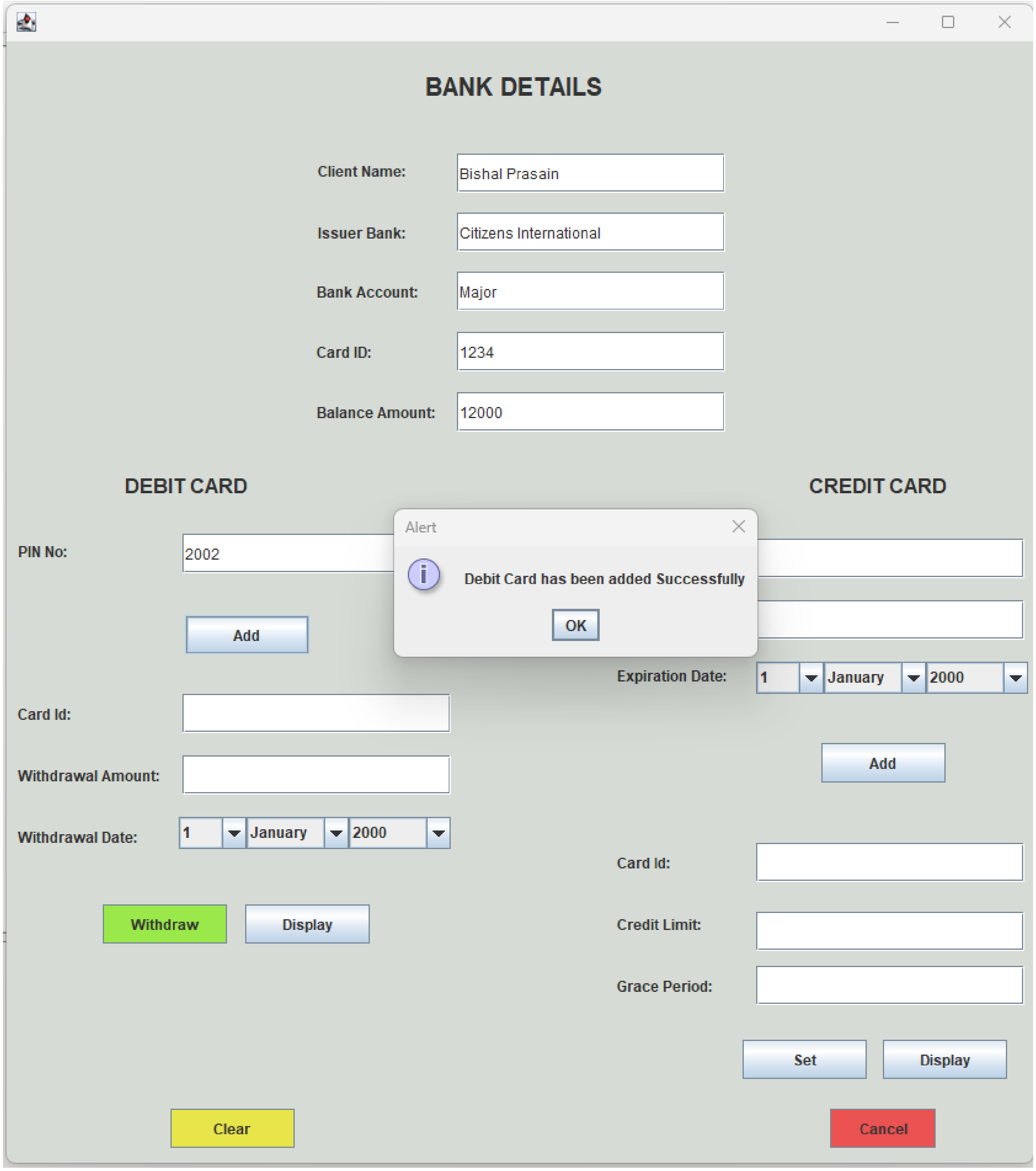
PIN No: 2002

Withdrawal Date: 1 January 2000

CREDIT CARD

Expiration Date: 1 January 2000

Figure 3: Entering values in the respective text fields for adding Debit Card



BANK DETAILS

Client Name: Bishal Prasain

Issuer Bank: Citizens International

Bank Account: Major

Card ID: 1234

Balance Amount: 12000

DEBIT CARD

PIN No: 2002

Alert

Debit Card has been added Successfully

Withdrawal Date: 1 January 2000

CREDIT CARD

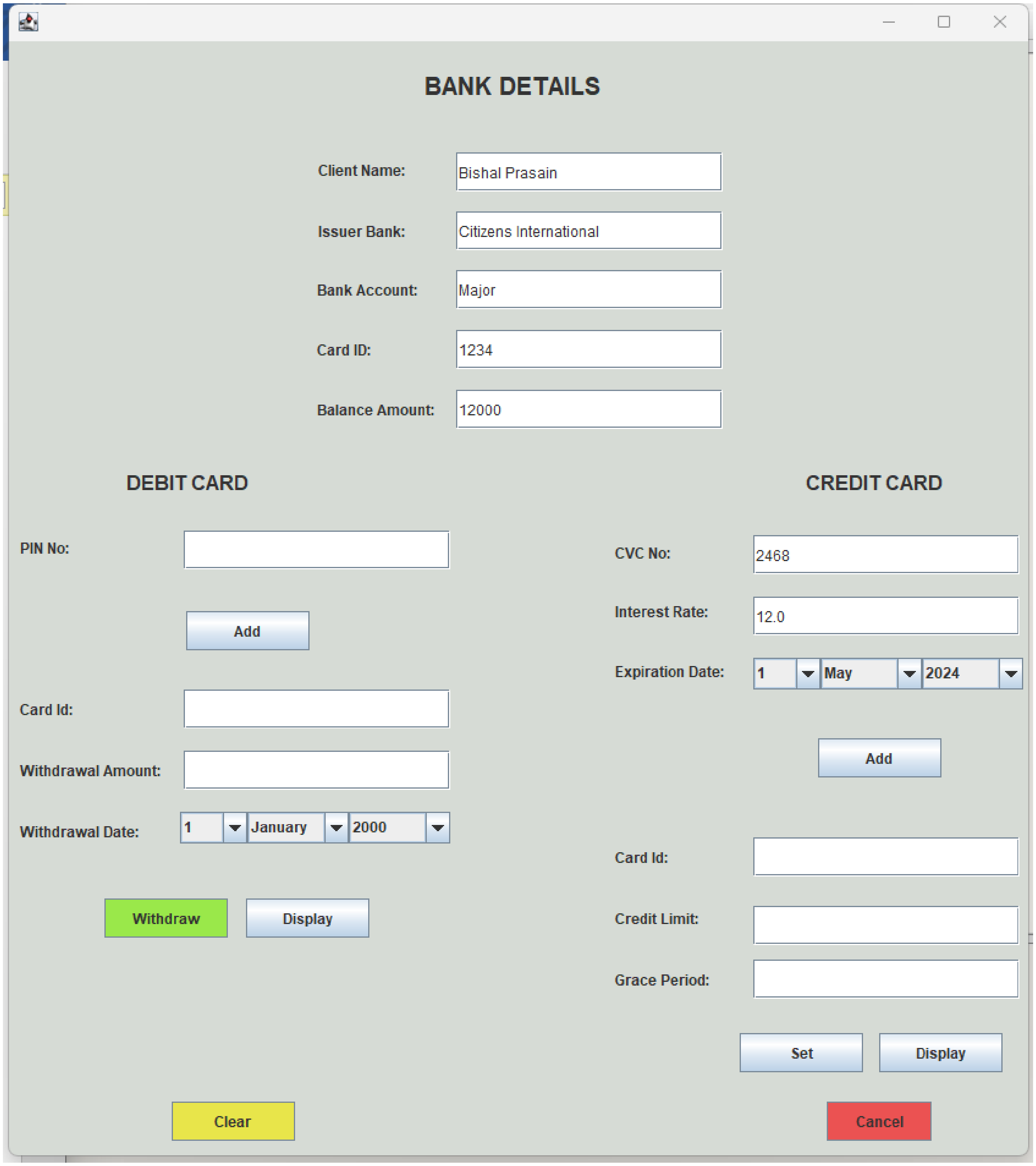
Expiration Date: 1 January 2000

Figure 4: Figure of showing dialog box after pressing add button of Debit Card

2)Testing the functionality of Add CreditCard Button

Test No.	2.2
Objective:	To check the functionality of add button of Credit Card
Action:	Add CreditCard button was pressed by entering following values in a text field: Client Name: Bishal Prasain Issuer Bank: Citizens International Bank Account: Major Card ID: 1234 Balance Amount: 12000 CVC No.: 2468 Interest Rate: 12.0 Expiration Date: 1 May 2024
Expected Result:	An information dialog would pop up informing that the Credit Card has been added successfully.
Actual Result:	An information dialog was popped up informing that the Credit Card has been added successfully.
Conclusion:	The test is successful.

Table 7:To check the functionality of add Credit Card button



BANK DETAILS

Client Name: Bishal Prasain

Issuer Bank: Citizens International

Bank Account: Major

Card ID: 1234

Balance Amount: 12000

DEBIT CARD

PIN No:

Card Id:

Withdrawal Amount:

Withdrawal Date: 1 January 2000

CREDIT CARD

CVC No: 2468

Interest Rate: 12.0

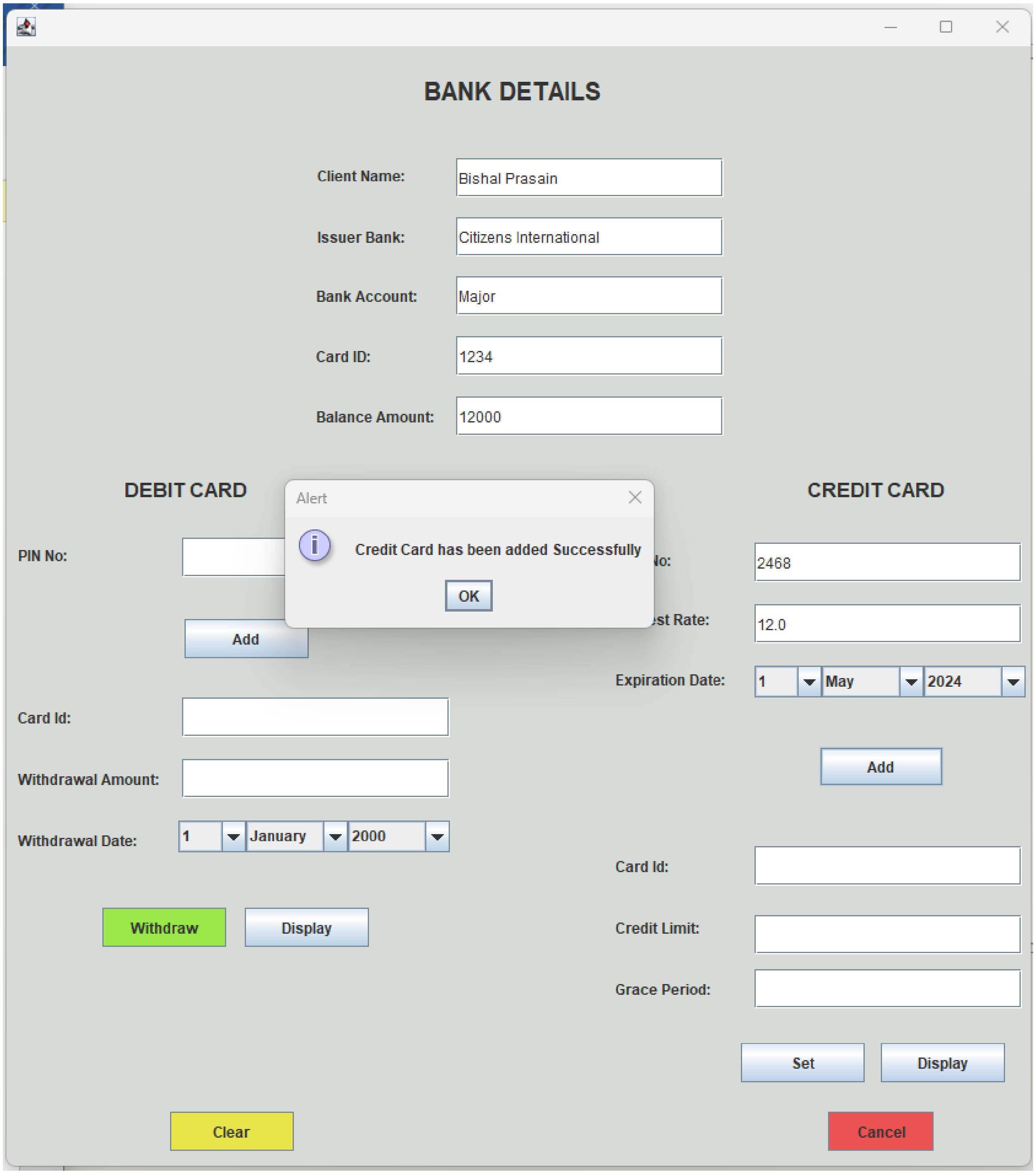
Expiration Date: 1 May 2024

Card Id:

Credit Limit:

Grace Period:

Figure 5: Entering values in the respective text fields for adding Credit Card



BANK DETAILS

Client Name: Bishal Prasain

Issuer Bank: Citizens International

Bank Account: Major

Card ID: 1234

Balance Amount: 12000

Alert

Credit Card has been added Successfully

No: 2468

est Rate: 12.0

Expiration Date: 1 May 2024

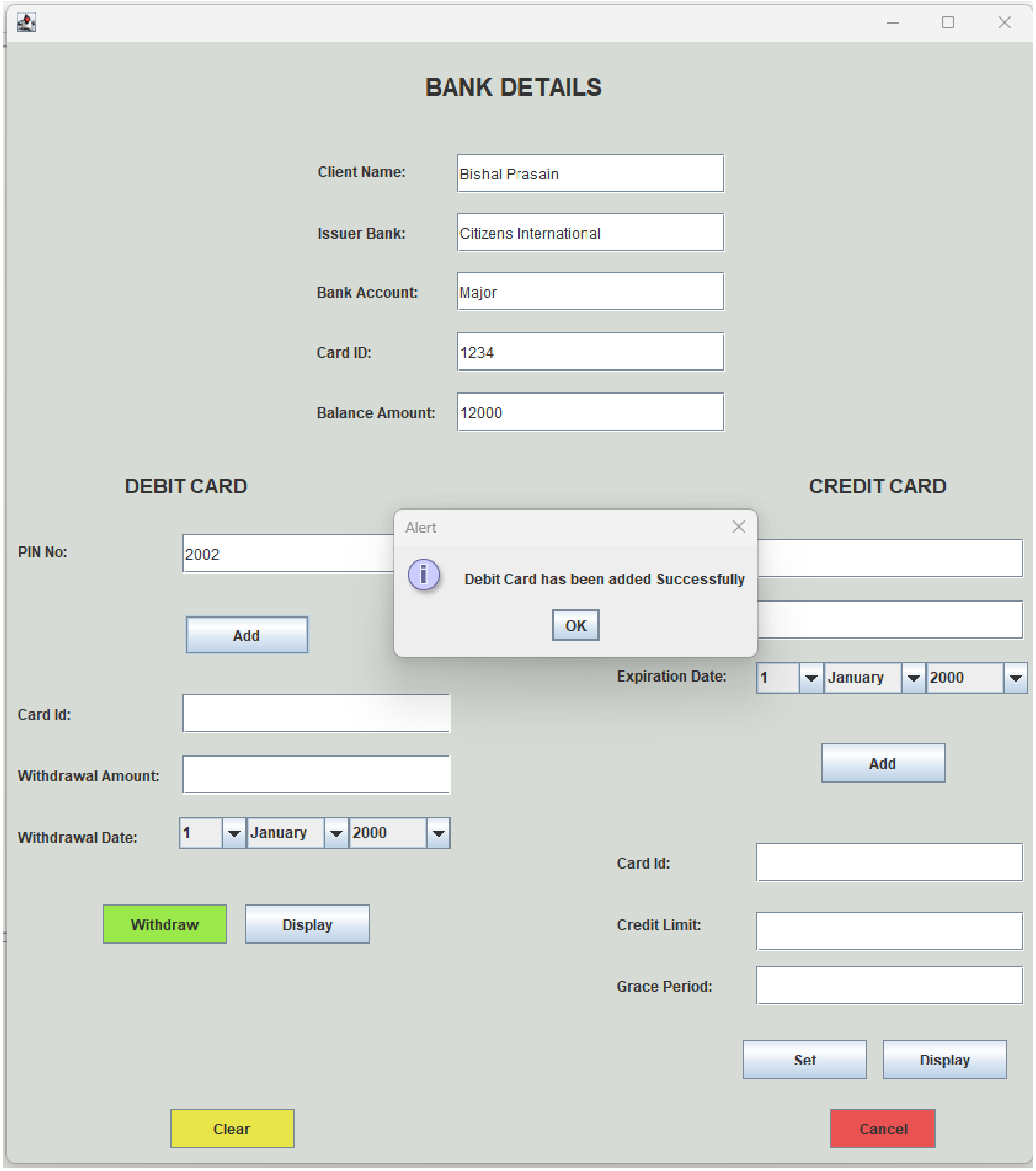
Figure 6:Figure of showing dialog box after pressing add button of Credit Card

3)Testing the functionality of withdraw Button

Test No.	2.3
Objective:	To withdraw amount from Debit Card using withdraw button
Action:	Add DebitCard button is clicked first by entering following values in a text field: Client Name: Bishal Prasain Issuer Bank: Citizens International Bank Account: Major Card ID: 1234 Balance Amount: 12000 PIN No: 2002 Then withdraw button is clicked by entering following values in a text field. Card ID: 1234 Withdrawal Amount: 2000 Withdrawal Date: 1 May 2023
Expected Result:	An information dialog would pop up showing that amount has been withdrawn successfully with all data that have been entered.

Actual Result:	An information dialog was pop up showing that amount has been withdrawn successfully with remaining amount.
Conclusion:	The test is successful.

Table 8: To check the functionality of withdraw button



BANK DETAILS

Client Name: Bishal Prasain

Issuer Bank: Citizens International

Bank Account: Major

Card ID: 1234

Balance Amount: 12000

DEBIT CARD

PIN No: 2002

Card Id:

Withdrawal Amount:

Withdrawal Date: 1 January 2000

CREDIT CARD

Expiration Date: 1 January 2000

Card Id:

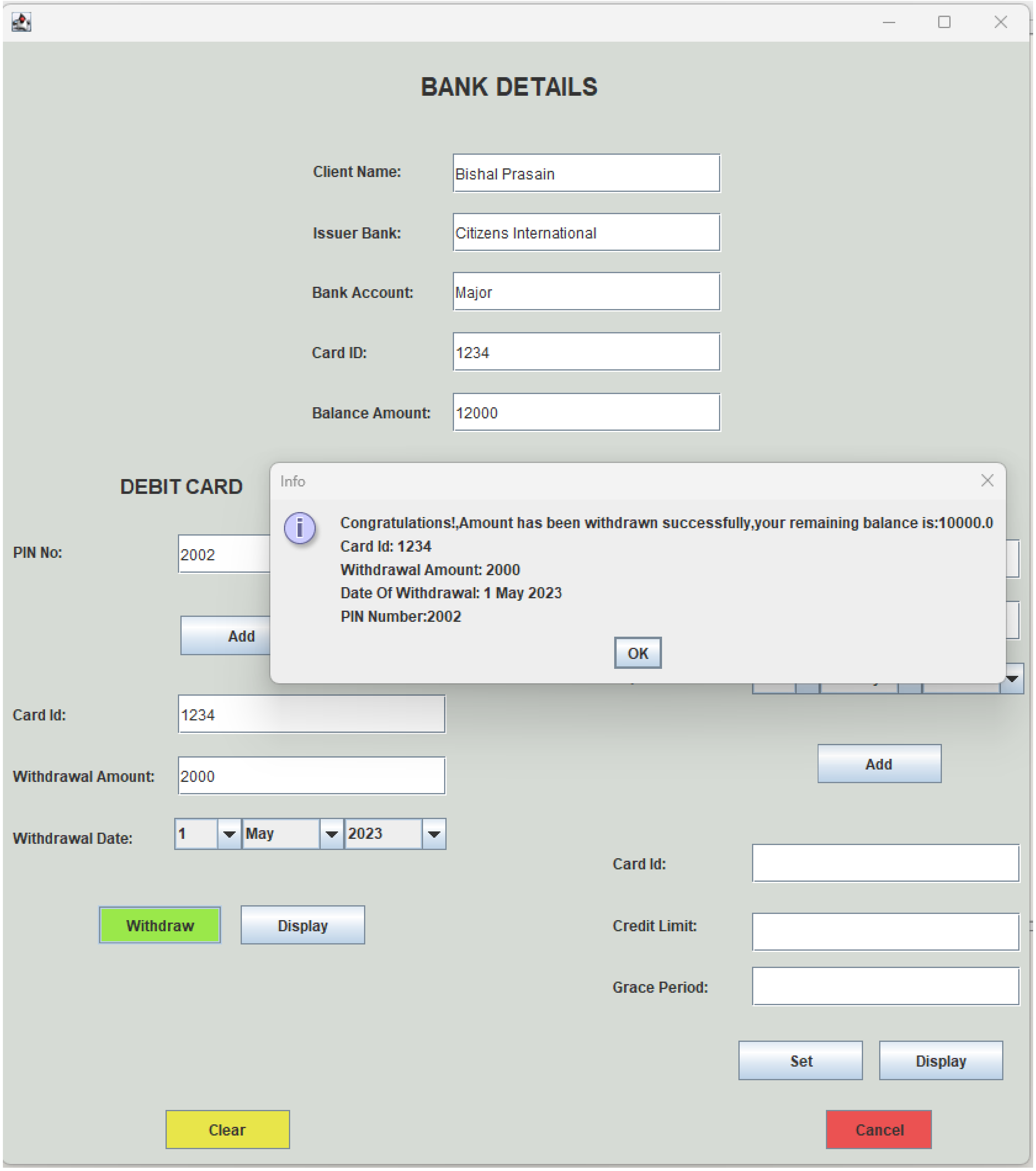
Credit Limit:

Grace Period:

Alert

Debit Card has been added Successfully

Figure 7:Figure of showing dialog box after pressing add button of Debit Card



BANK DETAILS

Client Name: Bishal Prasain

Issuer Bank: Citizens International

Bank Account: Major

Card ID: 1234

Balance Amount: 12000

PIN No: 2002

Card Id: 1234

Withdrawal Amount: 2000

Withdrawal Date: 1 May 2023

Info

Congratulations!,Amount has been withdrawn successfully,your remaining balance is:10000.0
Card Id: 1234
Withdrawal Amount: 2000
Date Of Withdrawal: 1 May 2023
PIN Number:2002

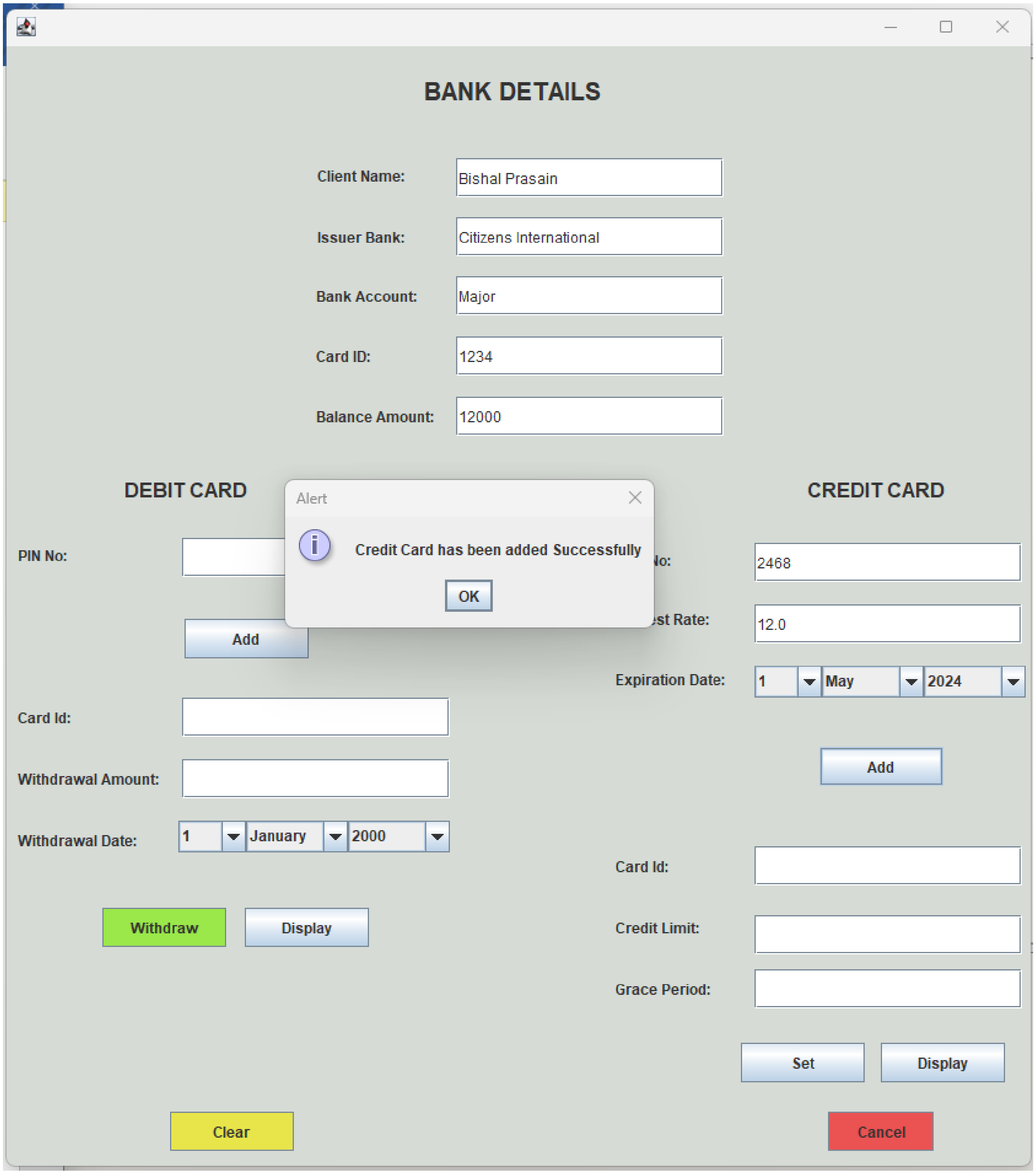
Figure 8:Figure of showing dialog box after withdrawing amount by entering respective values in the text field.

4)Testing the functionality of Set Credit Limit button

Test No.	2.4
Objective:	To set the Credit Limit using Set button
Action:	Add CreditCard button is pressed first by entering following values in a text fields: Client Name: Bishal Prasain Issuer Bank: Citizens International Bank Account: Major Card ID: 1234 Balance Amount: 10000 CVC No.: 2468 Interest Rate: 12.0 Expiration Date: 1 May 2024 Then Set button is clicked after entering following values in a text fields: Card ID: 1234 Credit Limit: 10 Grace Period: 15
Expected Result:	An information dialog would pop up saying Credit Limit has been set displaying Credit Limit and Grace Period.

Actual Result:	An information dialog was popped up saying Credit Limit has been set displaying Credit Limit and Grace Period.
Conclusion:	The test is successful.

Table 9:To check the functionality of set Credit Card Button



BANK DETAILS

Client Name: Bishal Prasain

Issuer Bank: Citizens International

Bank Account: Major

Card ID: 1234

Balance Amount: 12000

Alert

Credit Card has been added Successfully

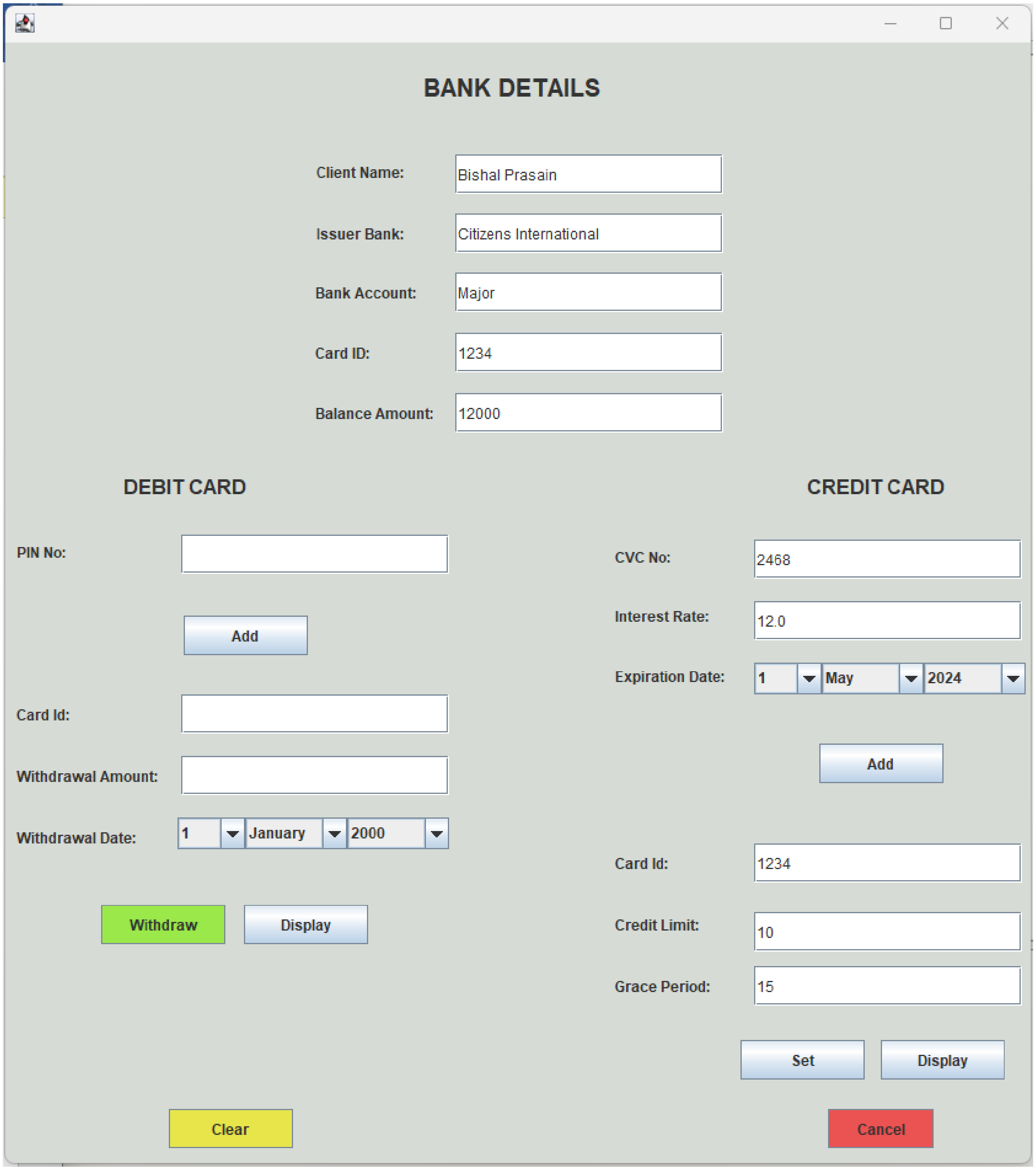
o: 2468

st Rate: 12.0

Expiration Date: 1 May 2024

Withdrawal Date: 1 January 2000

Figure 9:Figure of showing dialog box after pressing add button of Credit Card



BANK DETAILS

Client Name: Bishal Prasain

Issuer Bank: Citizens International

Bank Account: Major

Card ID: 1234

Balance Amount: 12000

DEBIT CARD

PIN No:

Card Id:

Withdrawal Amount:

Withdrawal Date: 1 January 2000

CREDIT CARD

CVC No: 2468

Interest Rate: 12.0

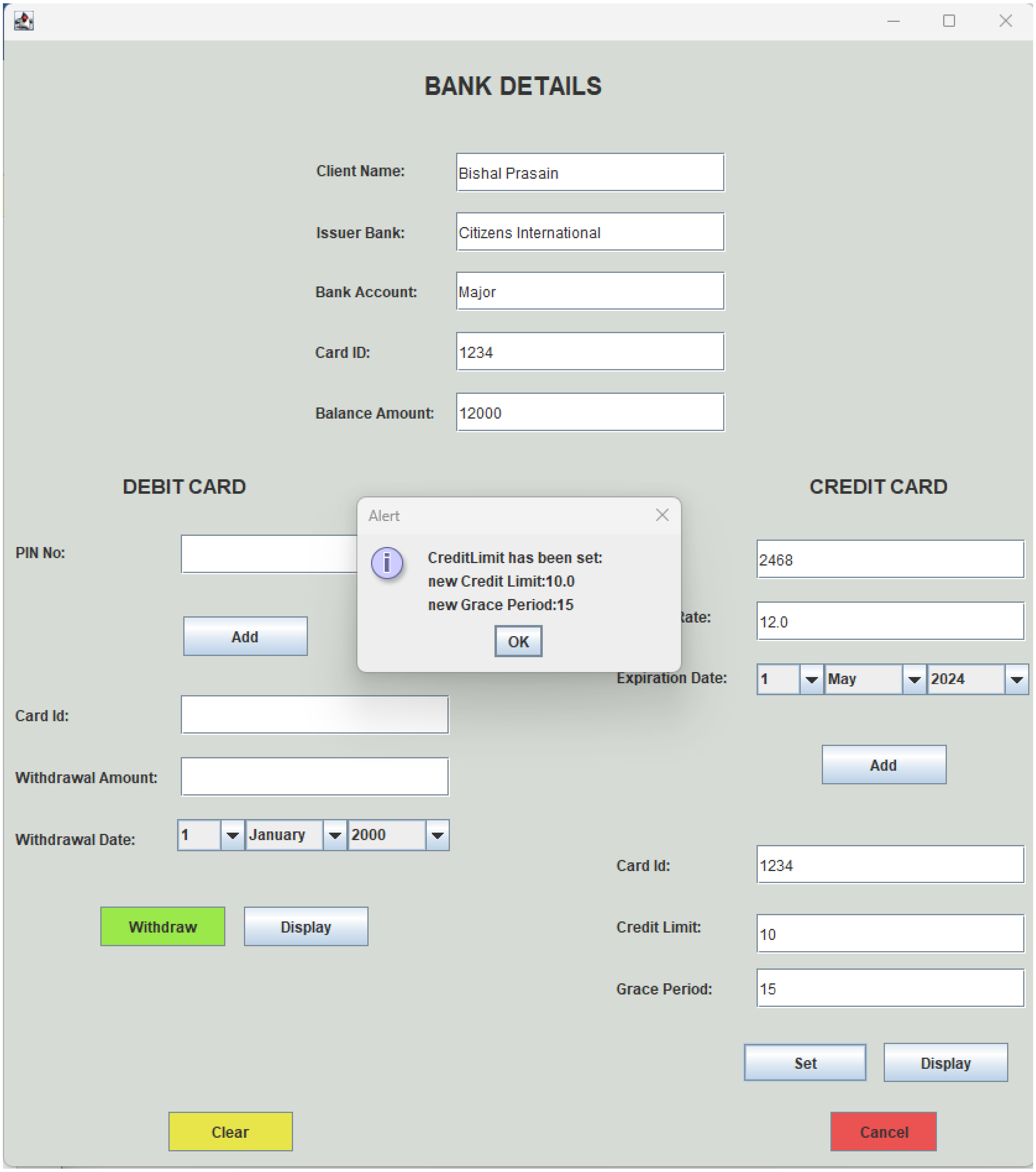
Expiration Date: 1 May 2024

Card Id: 1234

Credit Limit: 10

Grace Period: 15

Figure 10: Figure of entering text field for setting Credit Limit



BANK DETAILS

Client Name: Bishal Prasain

Issuer Bank: Citizens International

Bank Account: Major

Card ID: 1234

Balance Amount: 12000

DEBIT CARD

PIN No:

Card Id:

Withdrawal Amount:

Withdrawal Date: 1 January 2000

CREDIT CARD

2468

Rate: 12.0

Expiration Date: 1 May 2024

Card Id: 1234

Credit Limit: 10

Grace Period: 15

Alert

CreditLimit has been set:
new Credit Limit:10.0
new Grace Period:15

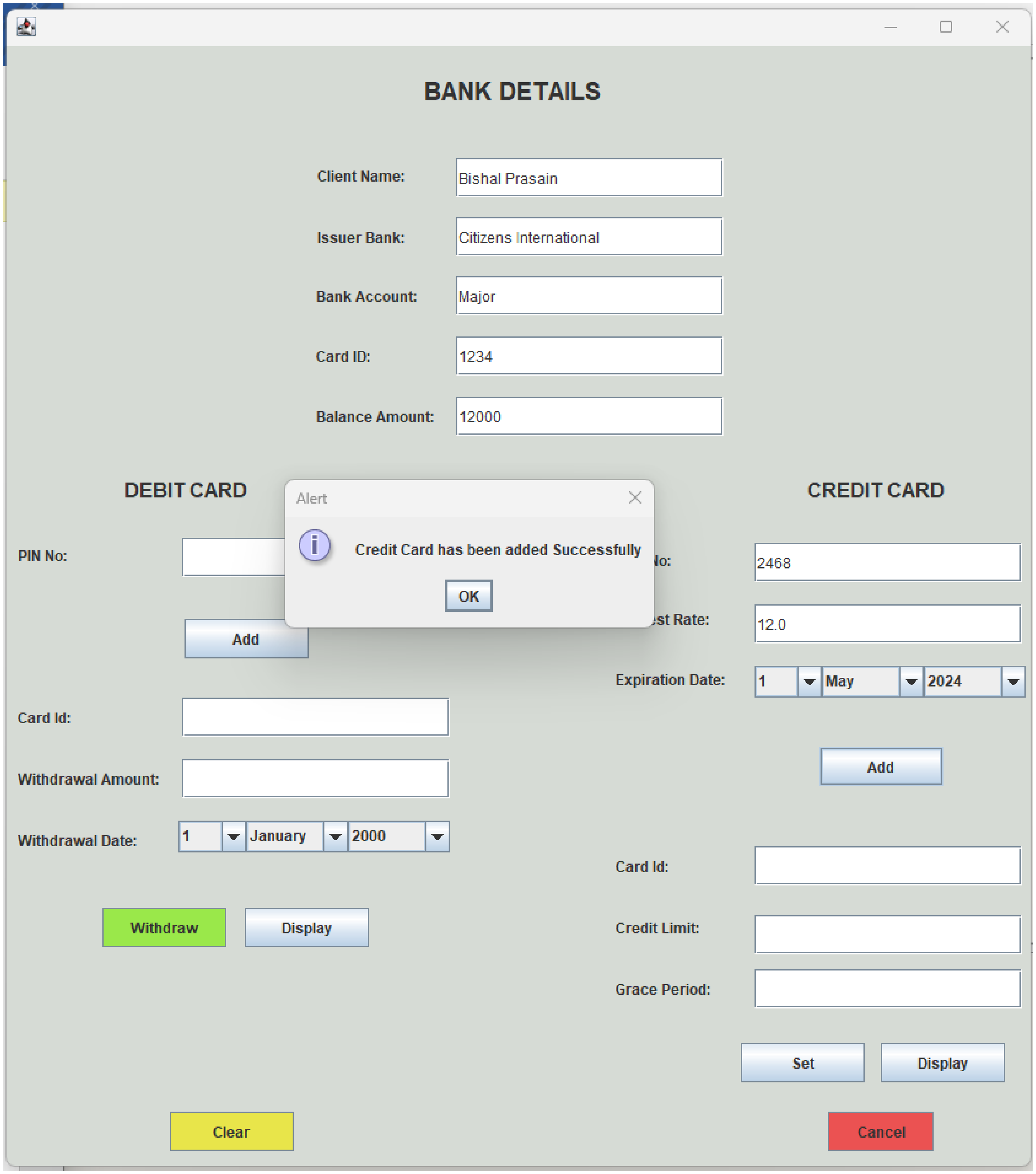
Figure 11: Figure of showing dialog box after pressing add button of Credit Card

5)Testing the functionality of Cancel CreditCard Button

Test No.	2.5
Objective:	To Cancel the Credit Limit using Cancel button
Action:	Add CreditCard button was pressed first by entering following values in a text fields: Client Name: Bishal Prasain Issuer Bank: Citizens International Bank Account: Major Card ID: 1234 Balance Amount: 10000 CVC No.: 2468 Interest Rate: 12.0 Expiration Date: 1 May 2024 Then Cancel button is clicked after entering following value in a text field: Card ID: 1234
Expected Result:	An information dialog would pop up saying Credit Card has been cancelled setting Credit Limit, Grace Period, and CVC number to empty text field.
Actual Result:	An information dialog was poped up saying Credit Card has been cancelled setting Credit Limit, Grace Period, and CVC number to empty text field.
Conclusion:	The test is successful.

Table 10: To check the functionality of Cancel Credit Card Button

Output:



BANK DETAILS

Client Name: Bishal Prasain

Issuer Bank: Citizens International

Bank Account: Major

Card ID: 1234

Balance Amount: 12000

DEBIT CARD

PIN No:

Card Id:

Withdrawal Amount:

Withdrawal Date: 1 January 2000

CREDIT CARD

No: 2468

est Rate: 12.0

Expiration Date: 1 May 2024

Card Id:

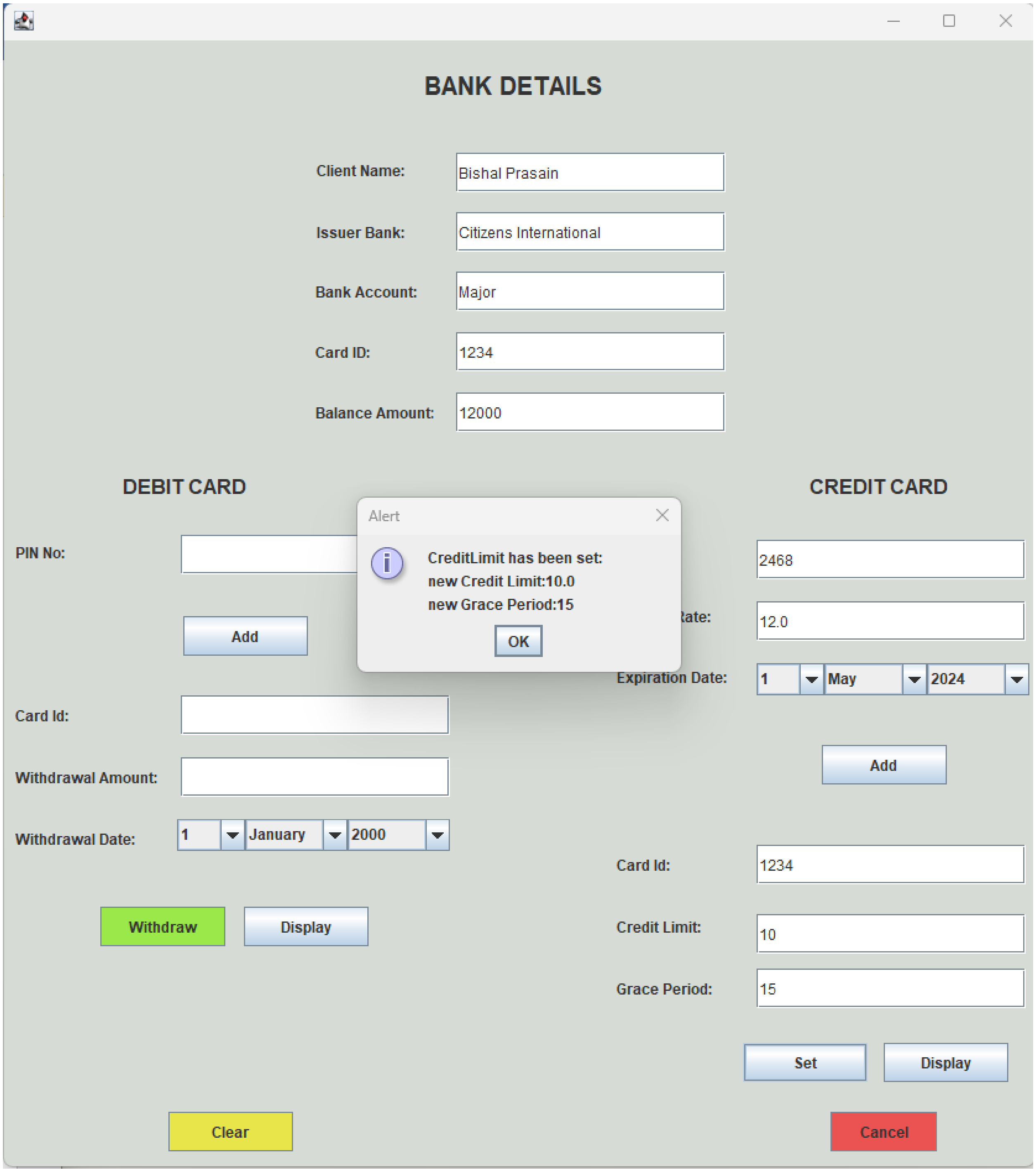
Credit Limit:

Grace Period:

Alert

Credit Card has been added Successfully

Figure 10:Figure of showing dialog box after pressing add button of Credit Card



BANK DETAILS

Client Name: Bishal Prasain

Issuer Bank: Citizens International

Bank Account: Major

Card ID: 1234

Balance Amount: 12000

PIN No:

Card Id:

Withdrawal Amount:

Withdrawal Date: 1 January 2000

2468

Rate: 12.0

Expiration Date: 1 May 2024

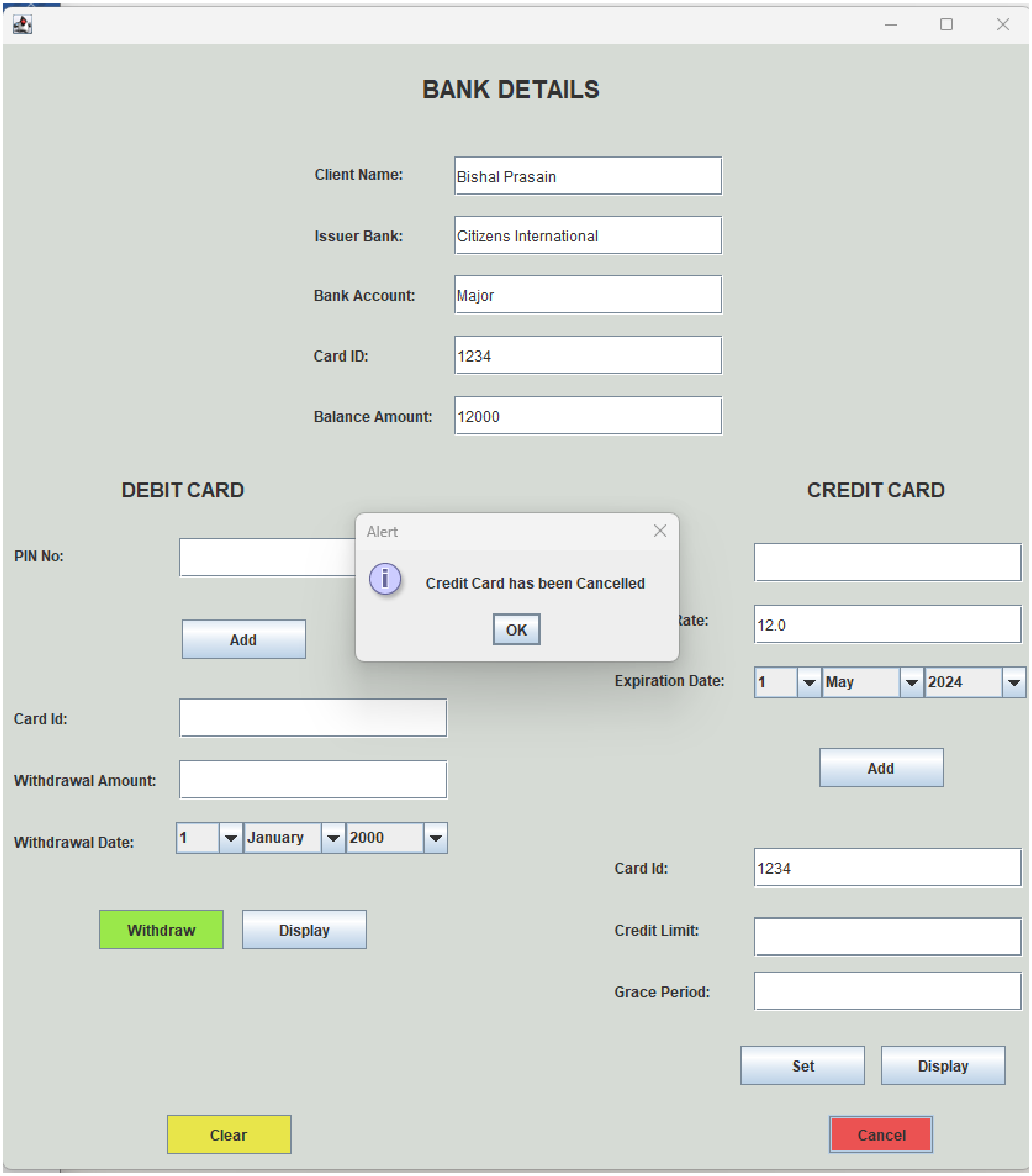
Card Id: 1234

Credit Limit: 10

Grace Period: 15

CreditLimit has been set:
new Credit Limit:10.0
new Grace Period:15

Figure 11: Figure of showing dialog box after setting Credit Limit


BANK DETAILS

Client Name: Bishal Prasain
Issuer Bank: Citizens International
Bank Account: Major
Card ID: 1234
Balance Amount: 12000

Alert

Credit Card has been Cancelled

ate: 12.0
Expiration Date: 1 May 2024
Card Id: 1234
Withdrawal Date: 1 January 2000

Figure 12: Figure of showing dialog box after cancelling the Credit Card.

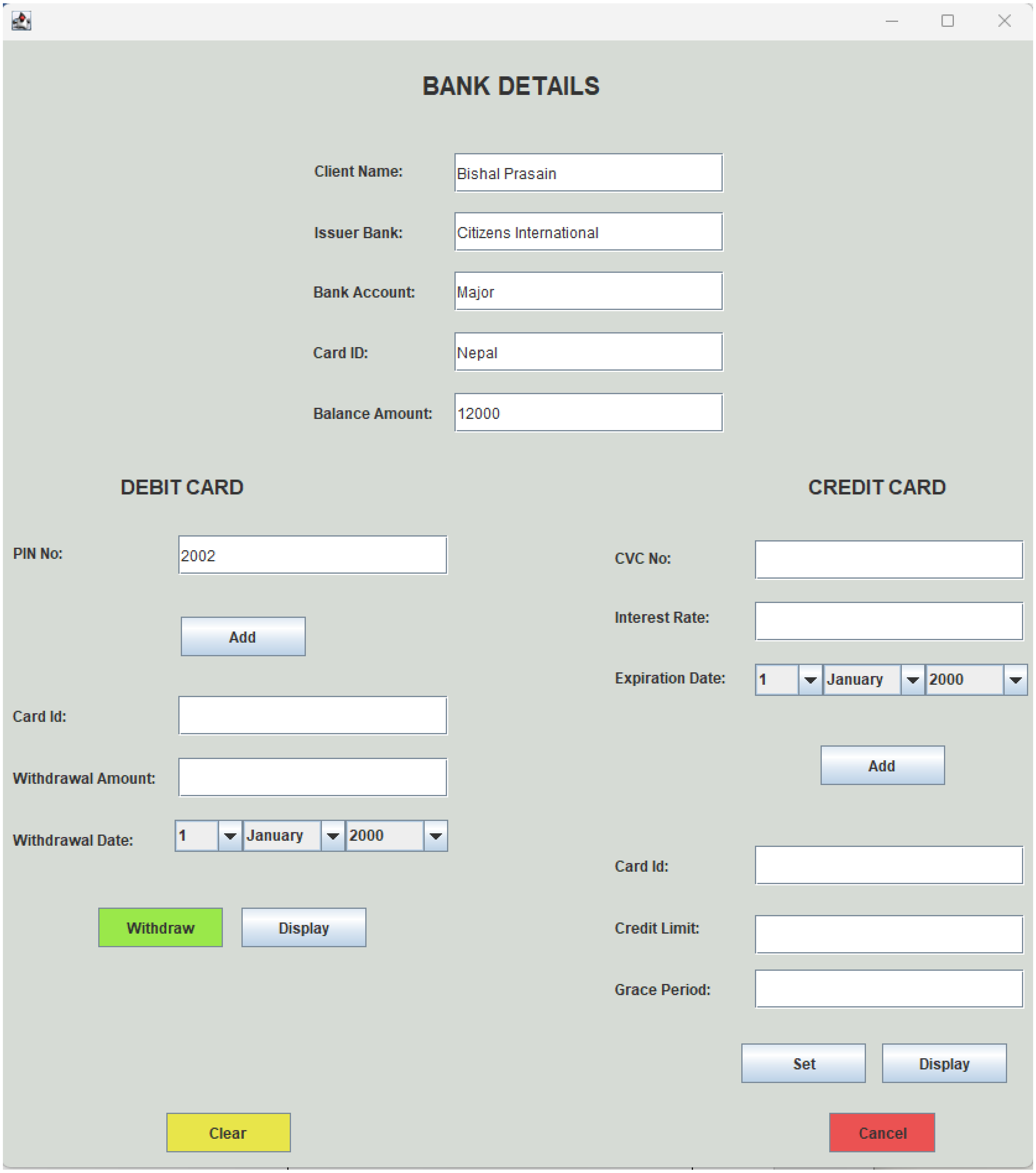
Test-3

1) **Testing the functionality of add Debit Card button after entering unsuitable value for Card Id.**

Test No.	3.1
Objective:	To check the functionality of add button of Debit Card after entering unsuitable value for Card ID
Action:	Add DebitCard button was pressed by entering following values in a text field: Client Name: Bishal Prasain Issuer Bank: Citizens International Bank Account: Major Card ID: Nepal Balance Amount: 12000 Pin No.: 2002
Expected Result:	An error dialog would pop up saying invalid format
Actual Result:	An error dialog was popped up saying the invalid format
Conclusion:	The test is successful.

Table 11:To check the functionality of add Debit Card button after entering unsuitable values

Output:



BANK DETAILS

Client Name: Bishal Prasain

Issuer Bank: Citizens International

Bank Account: Major

Card ID: Nepal

Balance Amount: 12000

DEBIT CARD

PIN No: 2002

Card Id:

Withdrawal Amount:

Withdrawal Date: 1 January 2000

CREDIT CARD

CVC No:

Interest Rate:

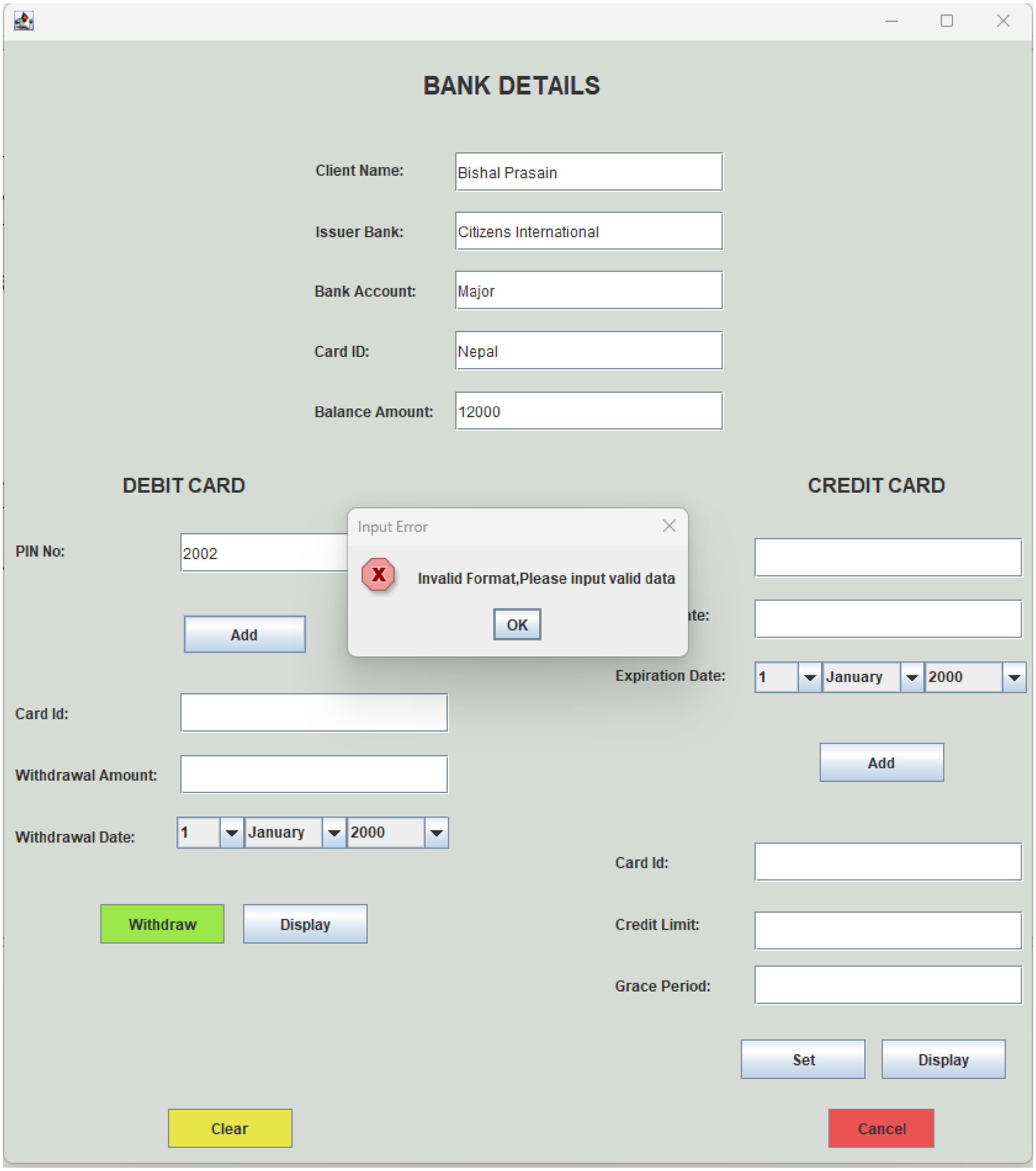
Expiration Date: 1 January 2000

Card Id:

Credit Limit:

Grace Period:

Figure 13:Figure of showing dialog box after pressing add button of Credit Card



BANK DETAILS

Client Name: Bishal Prasain

Issuer Bank: Citizens International

Bank Account: Major

Card ID: Nepal

Balance Amount: 12000

DEBIT CARD

PIN No: 2002

Card Id:

Withdrawal Amount:

Withdrawal Date: 1 January 2000

CREDIT CARD

ıte:

Expiration Date: 1 January 2000

Card Id:

Credit Limit:

Grace Period:

Input Error

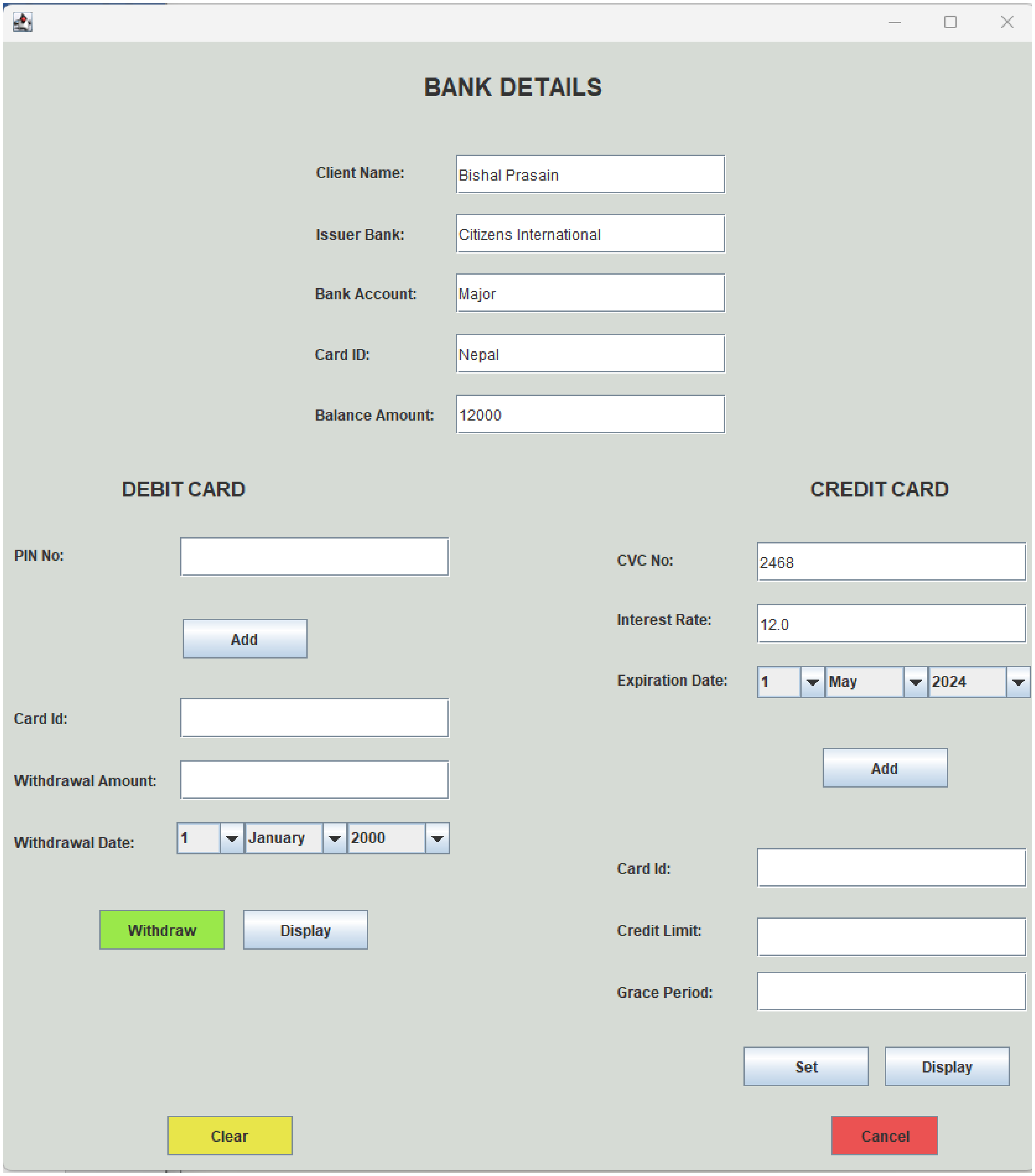
Invalid Format,Please input valid data

Figure 14:Figure of showing error message after entering unsuitable value of card ID in the text field.

2) Testing the functionality of add Credit Card button after entering unsuitable value for Card Id.

Test No.	3.2
Objective:	To check the functionality of add button of Debit Card after entering unsuitable value for Card ID
Action:	Add CreditCard button was pressed by entering following values in a text field: Client Name: Bishal Prasain Issuer Bank: Citizens International Bank Account: Major Card ID: Nepal Balance Amount: 12000 CVC No.: 2468 Interest Rate: 12.0 Expiration Date: 1 May 2024
Expected Result:	An error dialog would pop up saying invalid format
Actual Result:	An error dialog was popped up saying the invalid format
Conclusion:	The test is successful.

Table 12:To check the functionality of add Credit Card button after entering unsuitable values



BANK DETAILS

Client Name: Bishal Prasain

Issuer Bank: Citizens International

Bank Account: Major

Card ID: Nepal

Balance Amount: 12000

DEBIT CARD

PIN No:

Card Id:

Withdrawal Amount:

Withdrawal Date: 1 January 2000

CREDIT CARD

CVC No: 2468

Interest Rate: 12.0

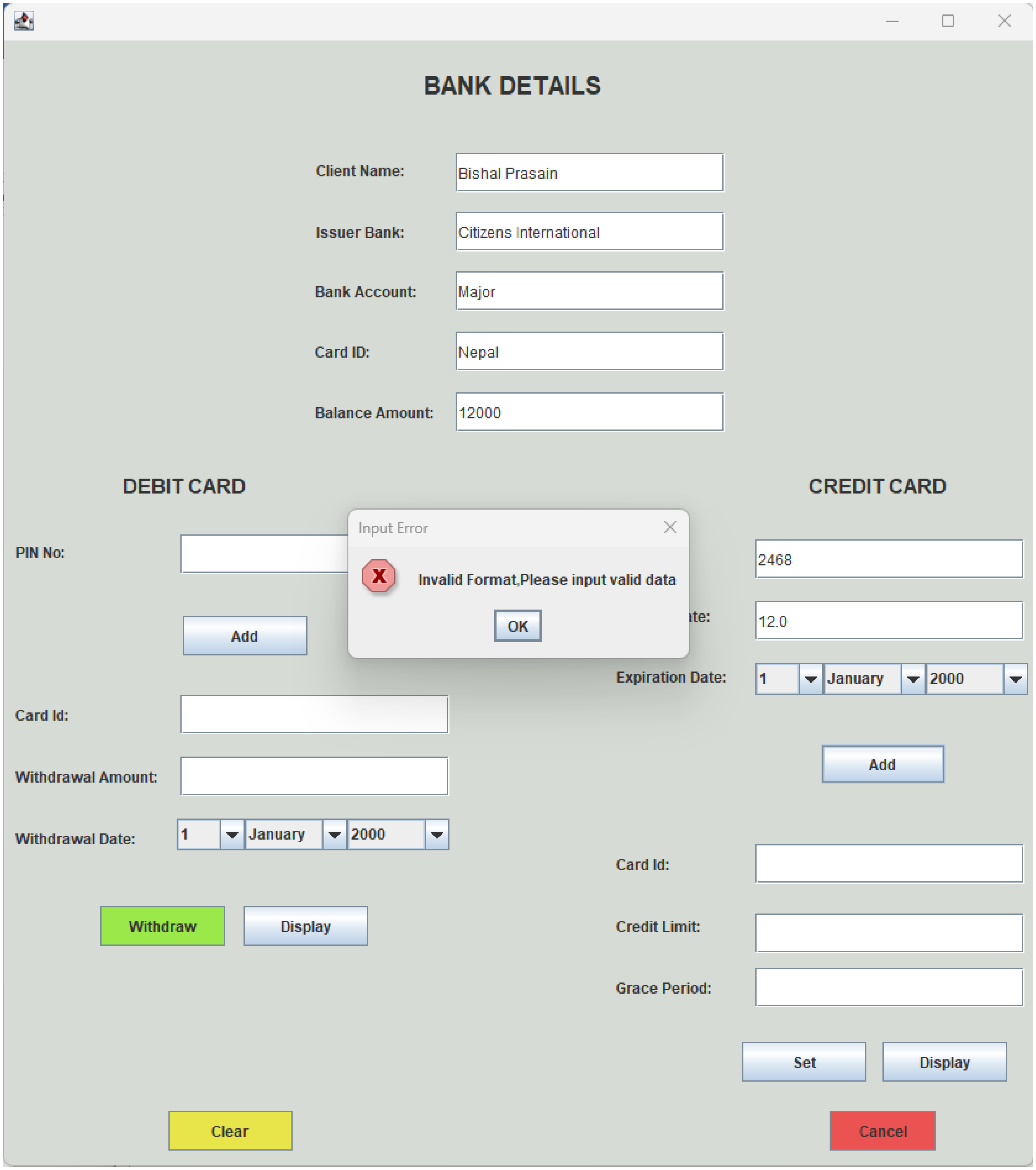
Expiration Date: 1 May 2024

Card Id:

Credit Limit:

Grace Period:

Figure 15:Figure of entering values in the respective text fields for adding Credit Card



BANK DETAILS

Client Name: Bishal Prasain

Issuer Bank: Citizens International

Bank Account: Major

Card ID: Nepal

Balance Amount: 12000

DEBIT CARD

PIN No:

Card Id:

Withdrawal Amount:

Withdrawal Date: 1 January 2000

CREDIT CARD

2468

ate: 12.0

Expiration Date: 1 January 2000

Card Id:

Credit Limit:

Grace Period:

Input Error

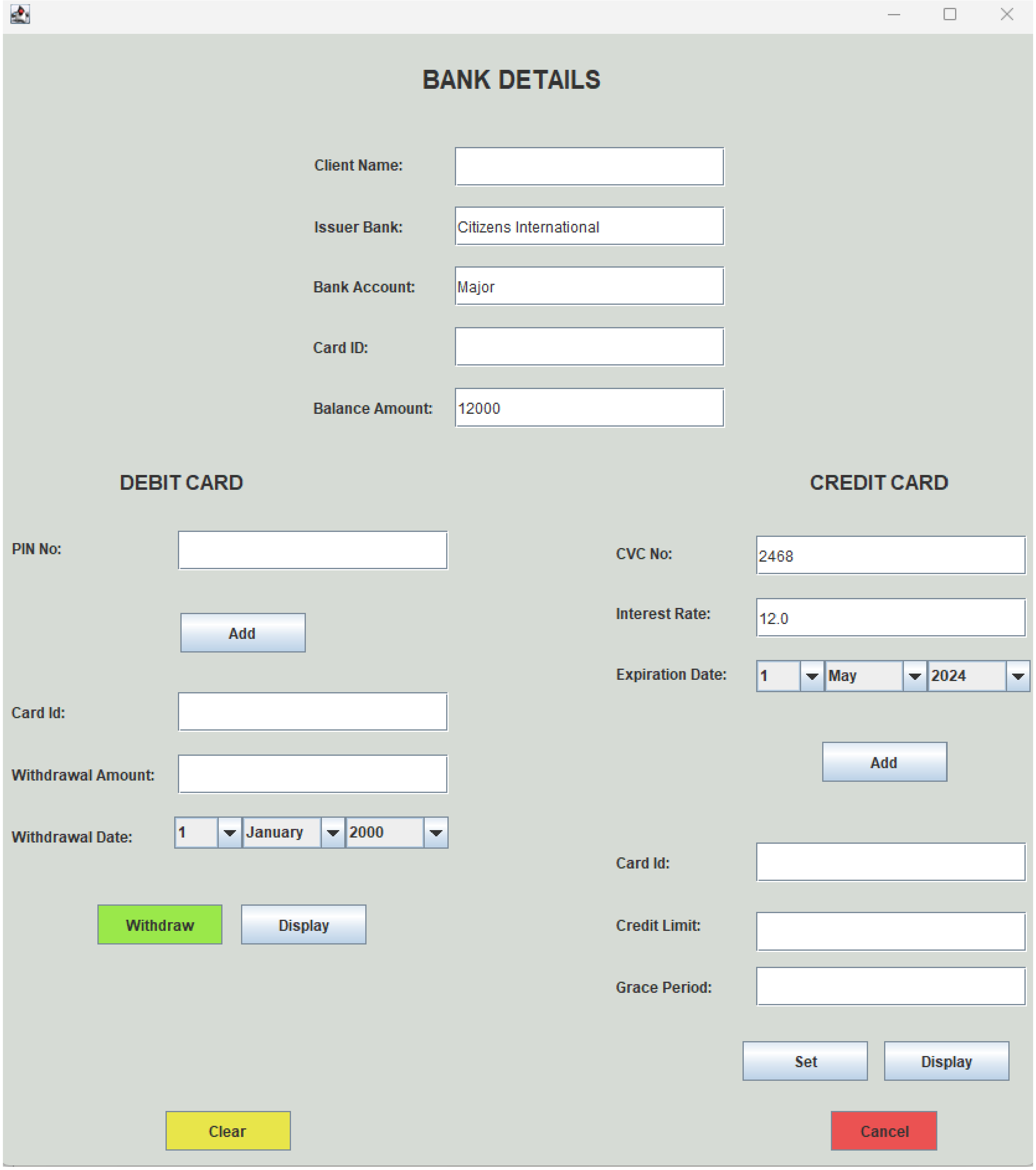
Invalid Format,Please input valid data

Figure 16: Figure of showing error message after entering unsuitable value of card ID in the text field while adding Credit Card

3) Testing the functionality of Add Button of Debit Card after leaving some of the text field empty

Test No.	3.3
Objective:	To check the functionality of add button of Debit Card after entering unsuitable value for Card ID
Action:	Add CreditCard button as pressed by entering following values in a text field: Client Name: “”(keeping empty) Issuer Bank: Citizens International Bank Account: Major Card ID: “”(keeping empty) Balance Amount: 12000 CVC No.: 2468 Interest Rate: 12.0 Expiration Date: 1 May 2024
Expected Result:	An error dialog would pop up saying text field cannot be empty
Actual Result:	An error dialog was popped up saying saying text field cannot be empty
Conclusion:	The test is successful.

Table 13: To check the functionaliaty of add Debit Card Button after leaving texfields empty



BANK DETAILS

Client Name:

Issuer Bank: Citizens International

Bank Account: Major

Card ID:

Balance Amount: 12000

DEBIT CARD

PIN No:

Card Id:

Withdrawal Amount:

Withdrawal Date: 1 January 2000

CREDIT CARD

CVC No: 2468

Interest Rate: 12.0

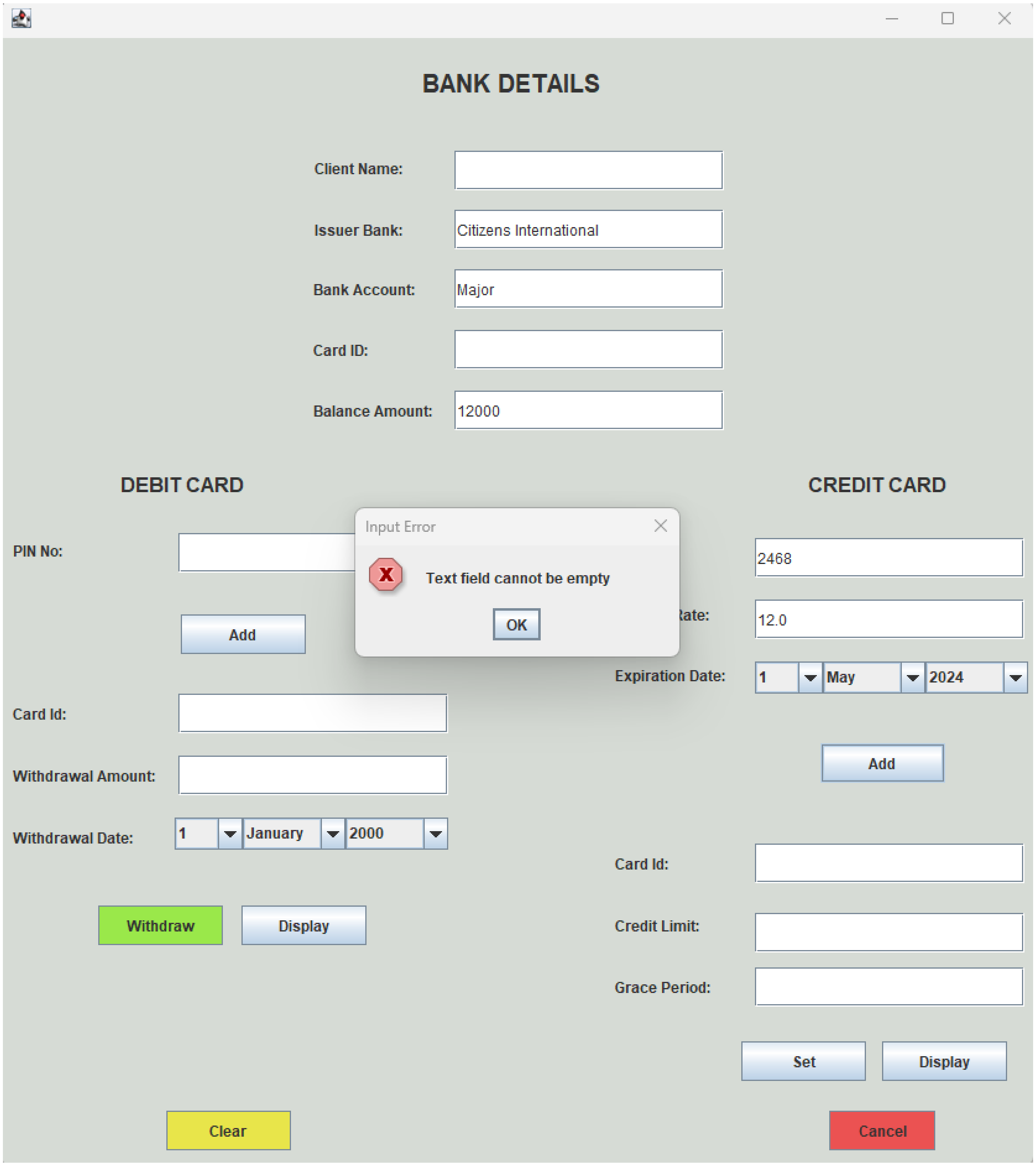
Expiration Date: 1 May 2024

Card Id:

Credit Limit:

Grace Period:

Figure 17:Figure of leaving text field empty while adding Credit Card



BANK DETAILS

Client Name:

Issuer Bank: Citizens International

Bank Account: Major

Card ID:

Balance Amount: 12000

DEBIT CARD

PIN No:

Card Id:

Withdrawal Amount:

Withdrawal Date: 1 January 2000

CREDIT CARD

2468

ate: 12.0

Expiration Date: 1 May 2024

Card Id:

Credit Limit:

Grace Period:

Input Error

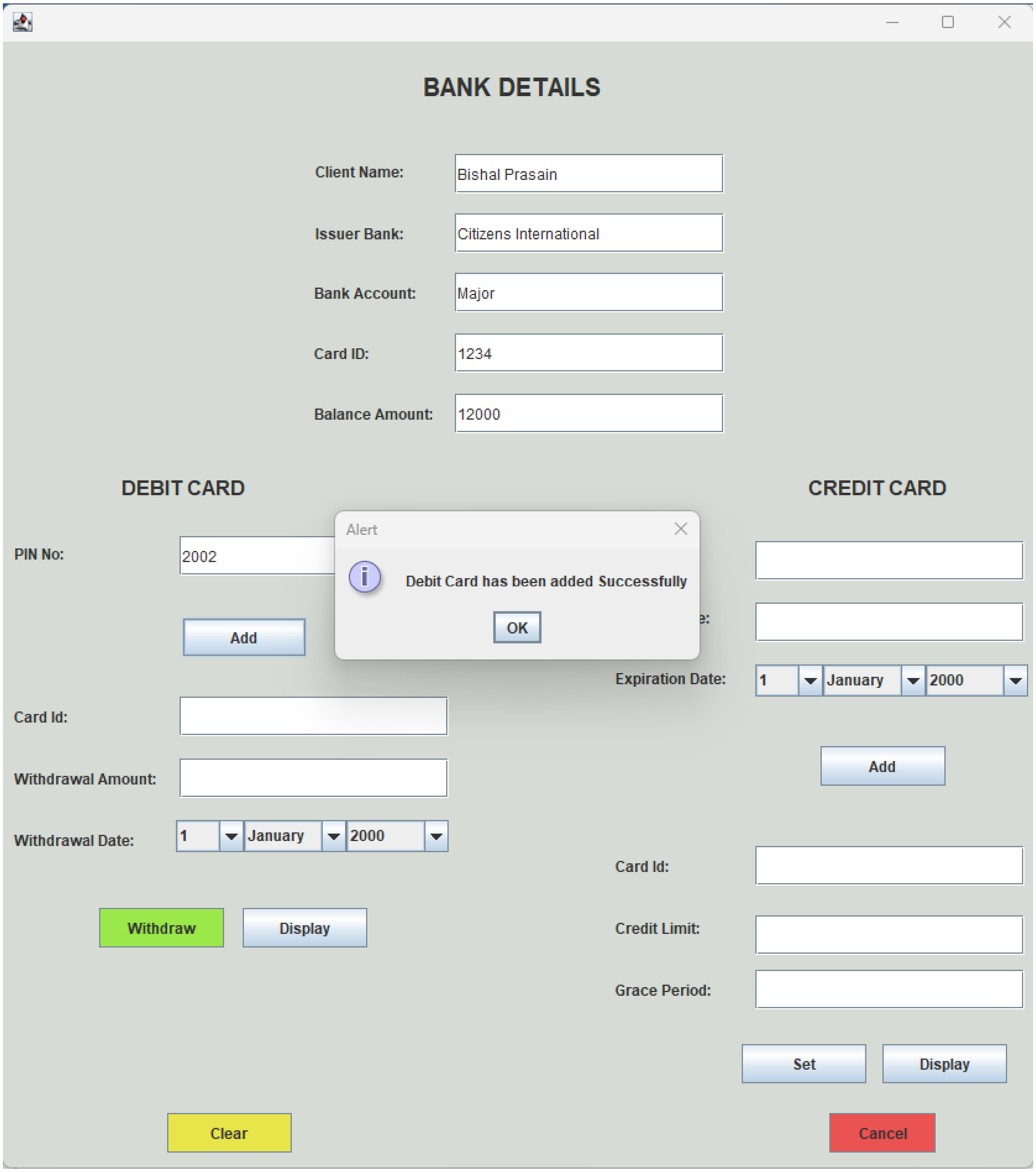
Text field cannot be empty

Figure 18:Figure of showing error message after leaving text field empty while adding Credit Card

4) Testing the functionality of withdraw button after entering invalid Card ID

Test No.	3.4
Objective:	To check the functionality of withdraw button after entering wrong card ID
Action:	Add DebitCard button was clicked first by entering following values in a text field: Client Name: Bishal Prasain Issuer Bank: Citizens International Bank Account: Major Card ID: 1234 Balance Amount: 12000 PIN No: 2002 Then withdraw button was clicked by entering following values in a text field. Card ID: 12345 Withdrawal Amount: 2000 Withdrawal Date: 1 May 2023
Expected Result:	An information dialog would pop up saying incorrect Card ID or PinNumber.
Actual Result:	An information dialog was popped up saying incorrect CardID or PinNumber.
Conclusion:	The test is successful.

Table 14: Checking the functionality of withdraw button after enterong invalid Card ID



BANK DETAILS

Client Name: Bishal Prasain

Issuer Bank: Citizens International

Bank Account: Major

Card ID: 1234

Balance Amount: 12000

DEBIT CARD

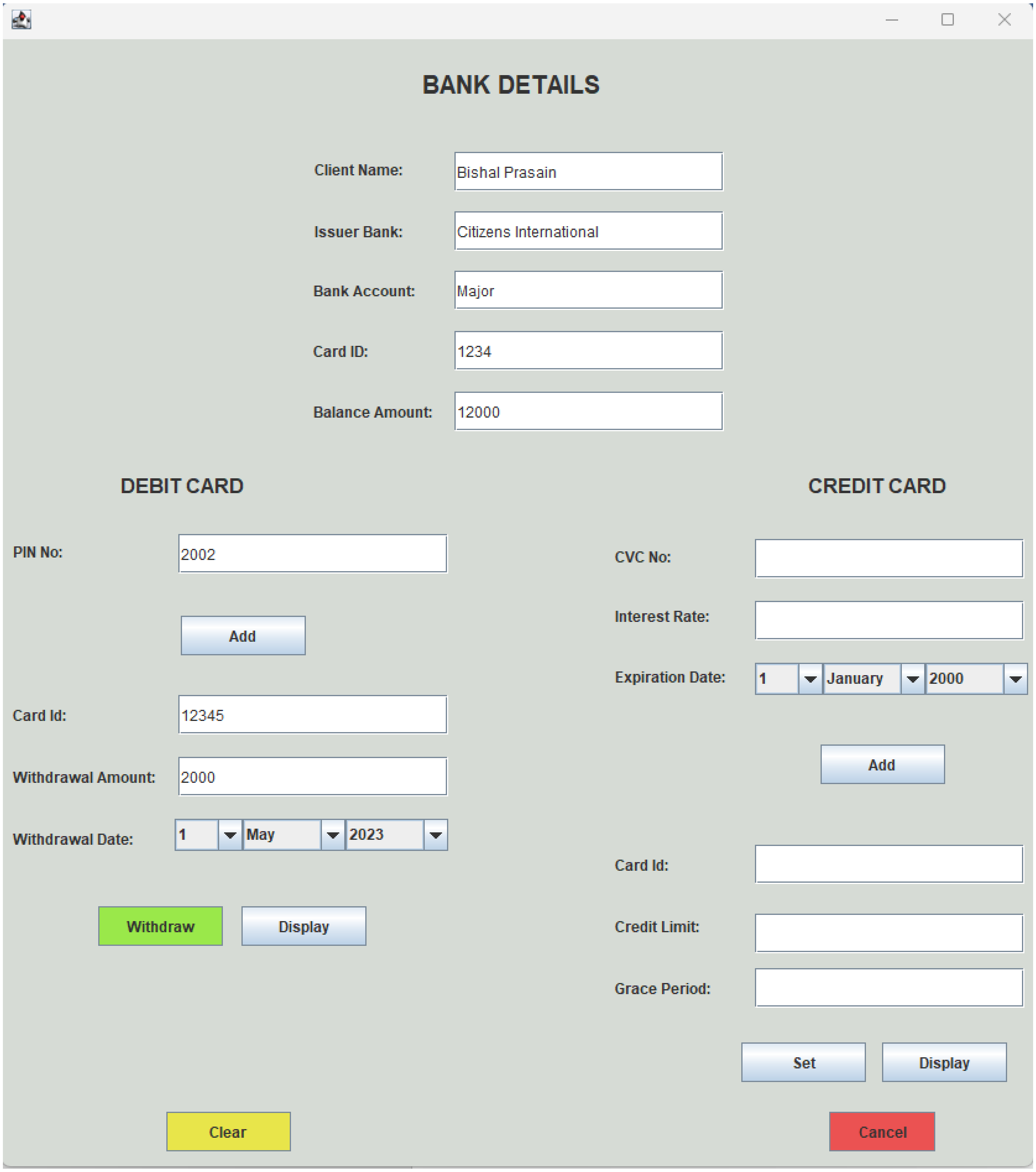
PIN No: 2002

CREDIT CARD

Alert

Debit Card has been added Successfully

Figure 19: Figure of adding Debit Card



BANK DETAILS

Client Name: Bishal Prasain

Issuer Bank: Citizens International

Bank Account: Major

Card ID: 1234

Balance Amount: 12000

DEBIT CARD

PIN No: 2002

Card Id: 12345

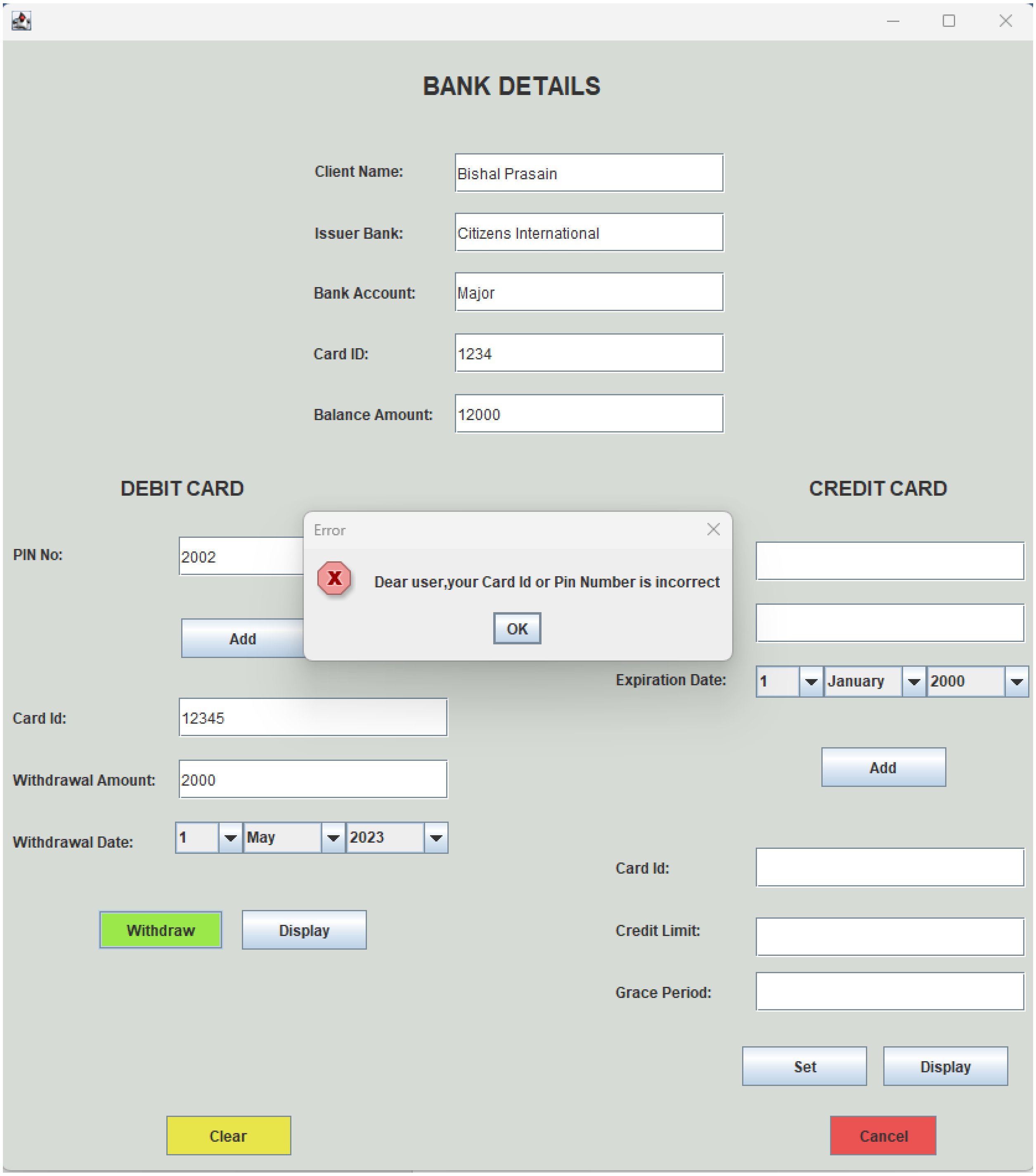
Withdrawal Amount: 2000

Withdrawal Date: 1 May 2023

CREDIT CARD

Expiration Date: 1 January 2000

Figure 20:Figure of entering values in the text field for withdrawing amount



BANK DETAILS

Client Name: Bishal Prasain

Issuer Bank: Citizens International

Bank Account: Major

Card ID: 1234

Balance Amount: 12000

DEBIT CARD

PIN No: 2002

Card Id: 12345

Withdrawal Amount: 2000

Withdrawal Date: 1 May 2023

CREDIT CARD

Expiration Date: 1 January 2000

Card Id:

Credit Limit:

Grace Period:

Error

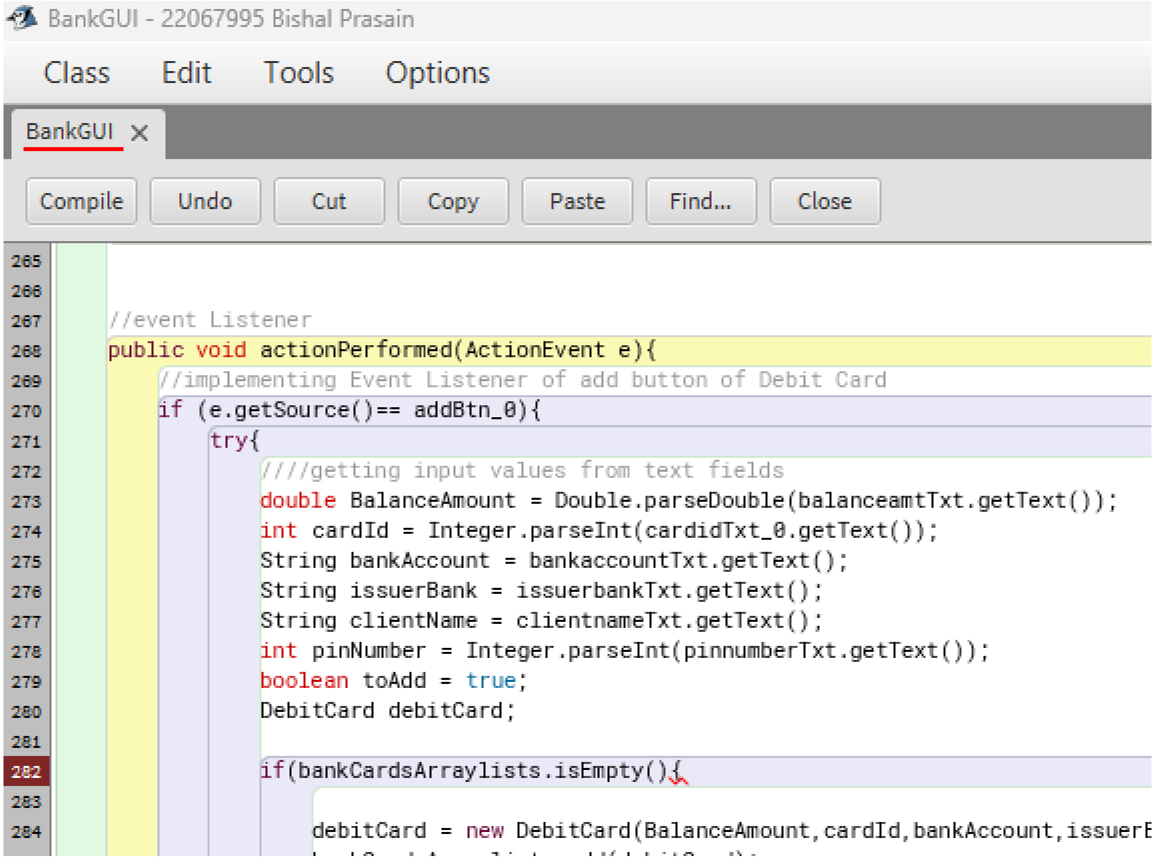
Dear user,your Card Id or Pin Number is incorrect

Figure 21: Figure of showing error message after entering wrong Card ID

Error detection and correction:

1) Syntax Error:

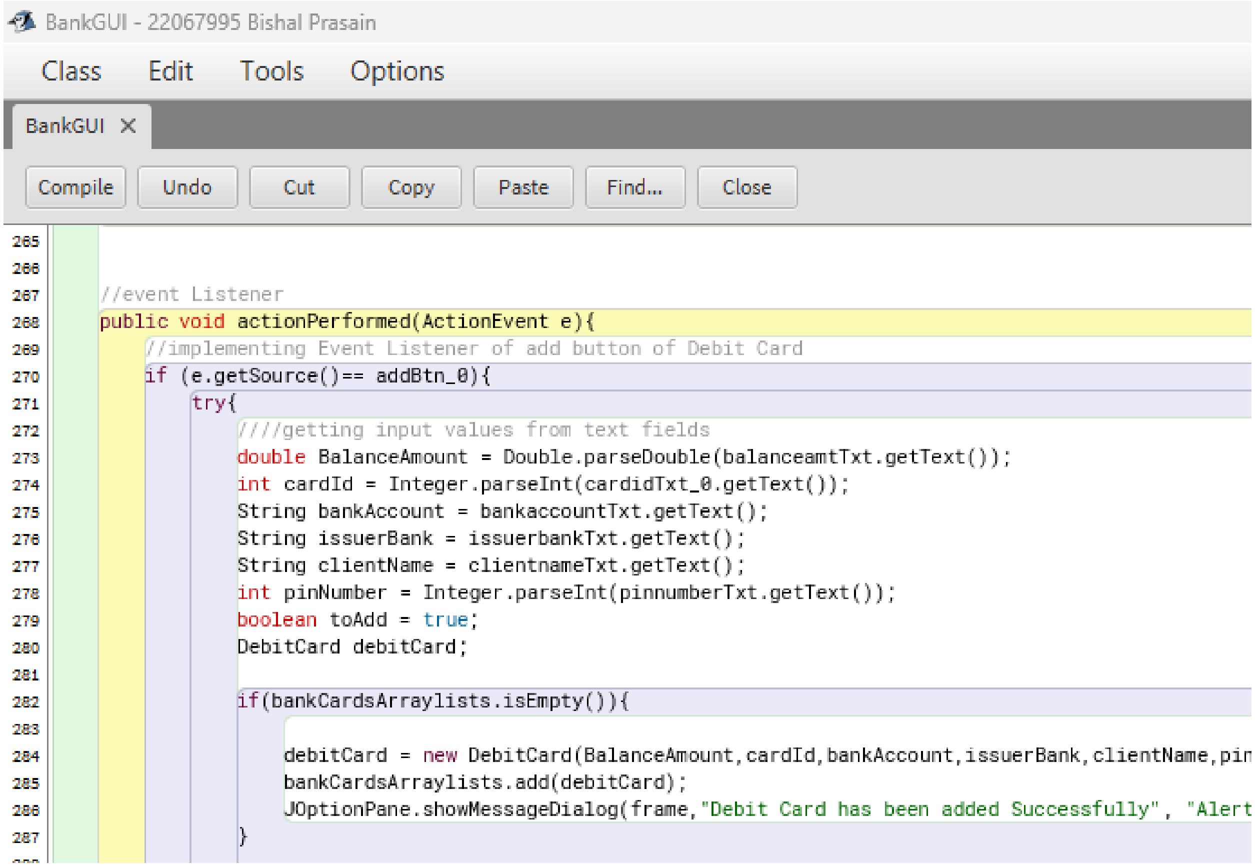
As simple as the word itself, Syntax error is the error in syntax/format of the programming language. (What is Syntax Error - Definition from Techopedia, 2023)It occurs when the written code does not align with the structure of the code. For example: forgetting punctuations, misspellings etc.


BankGUI - 22067995 Bishal Prasain
Class Edit Tools Options
BankGUI ✕
Compile Undo Cut Copy Paste Find... Close

```
265
266
267 //event Listener
268 public void actionPerformed(ActionEvent e){
269     //implementing Event Listener of add button of Debit Card
270     if (e.getSource()== addBtn_0){
271         try{
272             ////getting input values from text fields
273             double BalanceAmount = Double.parseDouble(balanceamtTxt.getText());
274             int cardId = Integer.parseInt(cardidTxt_0.getText());
275             String bankAccount = bankaccountTxt.getText();
276             String issuerBank = issuerbankTxt.getText();
277             String clientName = clientnameTxt.getText();
278             int pinNumber = Integer.parseInt(pinnumberTxt.getText());
279             boolean toAdd = true;
280             DebitCard debitCard;
281
282             if(bankCardsArraylists.isEmpty(){
283
284                 debitCard = new DebitCard(BalanceAmount,cardId,bankAccount,issuerE
```

Figure 22:Figure of not closing the codition of if statement

Error Detection: The condition of if statement is not closed at the code line 282.



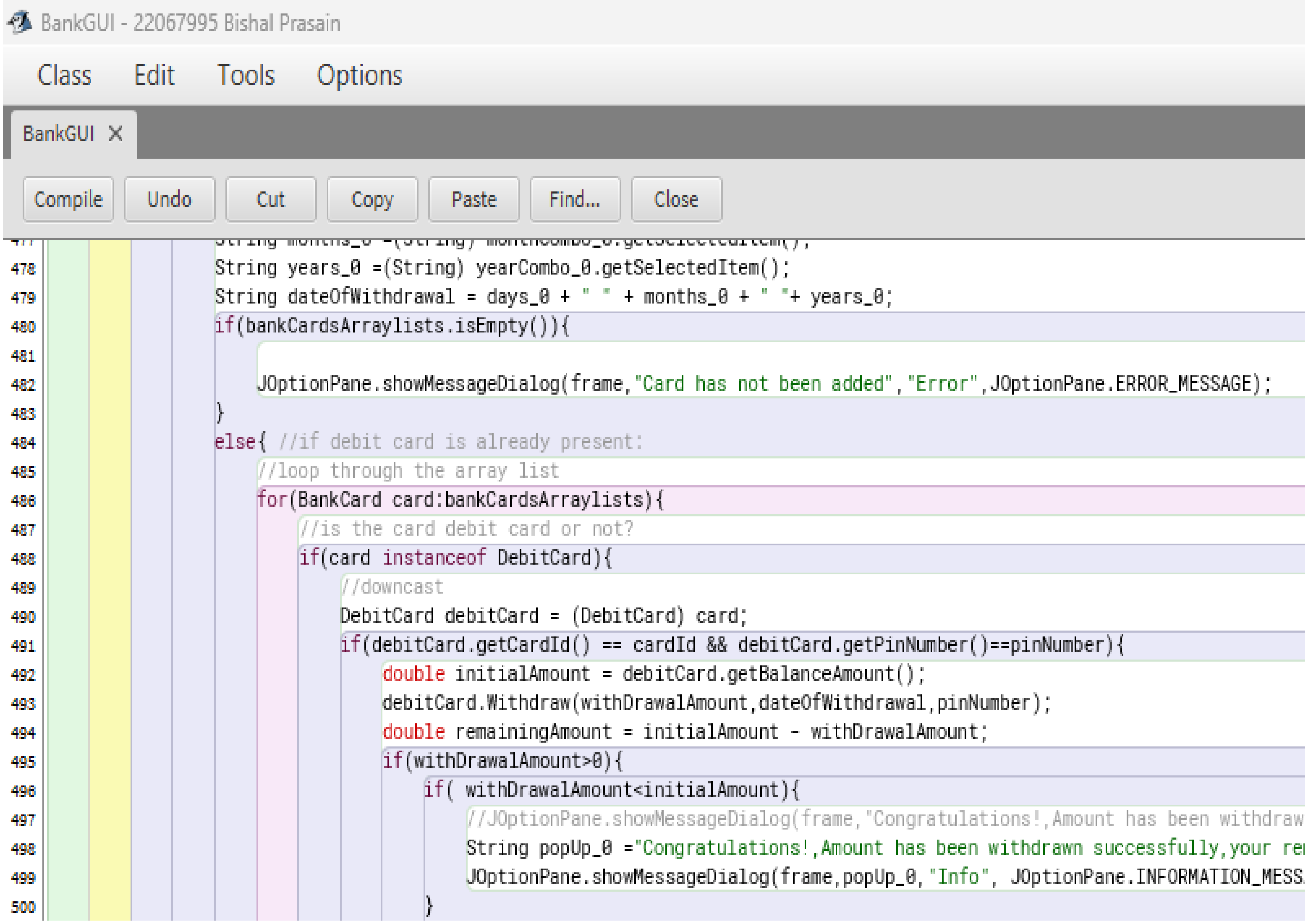
```


//event Listener
public void actionPerformed(ActionEvent e){
    //implementing Event Listener of add button of Debit Card
    if (e.getSource()== addBtn_0){
        try{
            ////getting input values from text fields
            double BalanceAmount = Double.parseDouble(balanceamtTxt.getText());
            int cardId = Integer.parseInt(cardidTxt_0.getText());
            String bankAccount = bankaccountTxt.getText();
            String issuerBank = issuerbankTxt.getText();
            String clientName = clientnameTxt.getText();
            int pinNumber = Integer.parseInt(pinnumberTxt.getText());
            boolean toAdd = true;
            DebitCard debitCard;

            if(bankCardsArraylists.isEmpty()){

                debitCard = new DebitCard(BalanceAmount,cardId,bankAccount,issuerBank,clientName,pin
                bankCardsArraylists.add(debitCard);
                JOptionPane.showMessageDialog(frame,"Debit Card has been added Successfully", "Alert
            }
```

Figure 23:Figure of fixing syntax error by closing the condition of if statement

Error solved: The condition of if statement is closed resulting in no compilation error

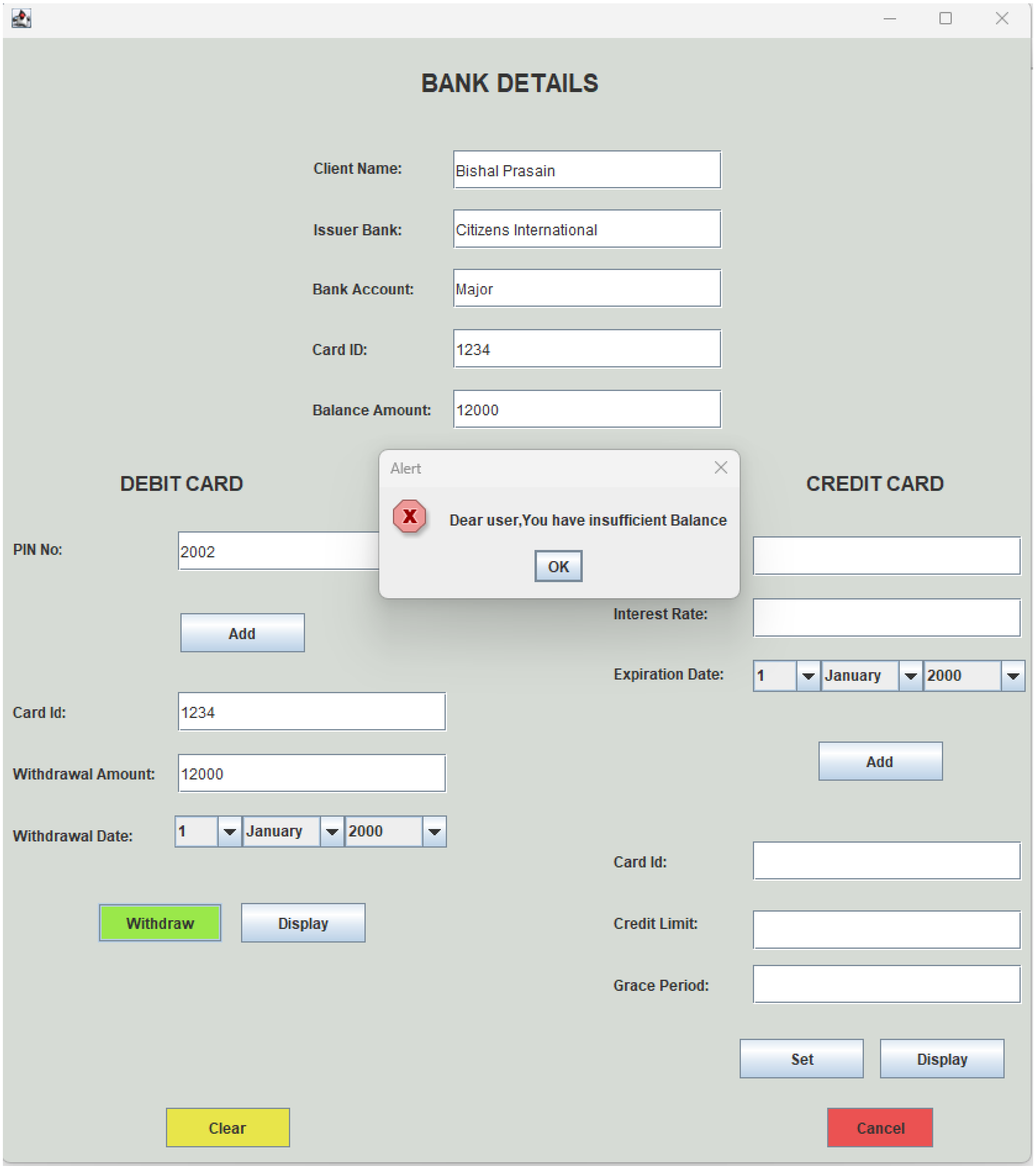
2)Semantic Error:

If the output of a certain program comes different from what was I ntended, then it can be termed as semantic errors. It can be witnessed when wrong input, wrong operators are used. Since it throws no compilation error, it is hard to address them.



```
String months_0 =(String) monthCombo_0.getSelectedItem();
String years_0 =(String) yearCombo_0.getSelectedItem();
String dateOfWithdrawal = days_0 + " " + months_0 + " "+ years_0;
if(bankCardsArraylists.isEmpty()){

    JOptionPane.showMessageDialog(frame,"Card has not been added","Error",JOptionPane.ERROR_MESSAGE);
}
else{ //if debit card is already present:
    //loop through the array list
    for(BankCard card:bankCardsArraylists){
        //is the card debit card or not?
        if(card instanceof DebitCard){
            //downcast
            DebitCard debitCard = (DebitCard) card;
            if(debitCard.getCardId() == cardId && debitCard.getPinNumber()==pinNumber){
                double initialAmount = debitCard.getBalanceAmount();
                debitCard.Withdraw(withDrawalAmount,dateOfWithdrawal,pinNumber);
                double remainingAmount = initialAmount - withDrawalAmount;
                if(withDrawalAmount>0){
                    if( withDrawalAmount<initialAmount){
                        //JOptionPane.showMessageDialog(frame,"Congratulations!,Amount has been withdraw
                        String popUp_0 ="Congratulations!,Amount has been withdrawn successfully,your re
                        JOptionPane.showMessageDialog(frame,popUp_0,"Info", JOptionPane.INFORMATION_MESS
                    }
```

Figure 24: Figure of addressing semantic error



BANK DETAILS

Client Name: Bishal Prasain

Issuer Bank: Citizens International

Bank Account: Major

Card ID: 1234

Balance Amount: 12000

DEBIT CARD

PIN No: 2002

Card Id: 1234

Withdrawal Amount: 12000

Withdrawal Date: 1 January 2000

CREDIT CARD

Interest Rate:

Expiration Date: 1 January 2000

Card Id:

Credit Limit:

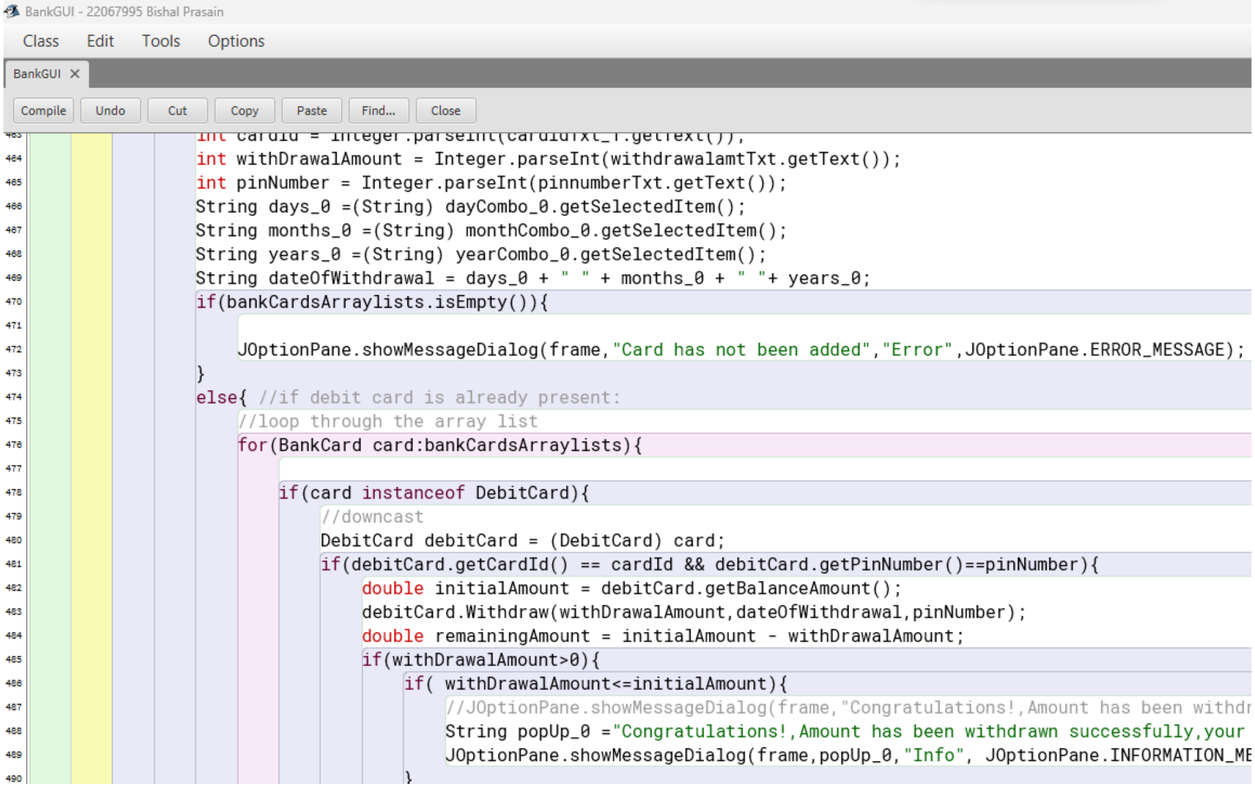
Grace Period:

Alert

Dear user,You have insufficient Balance

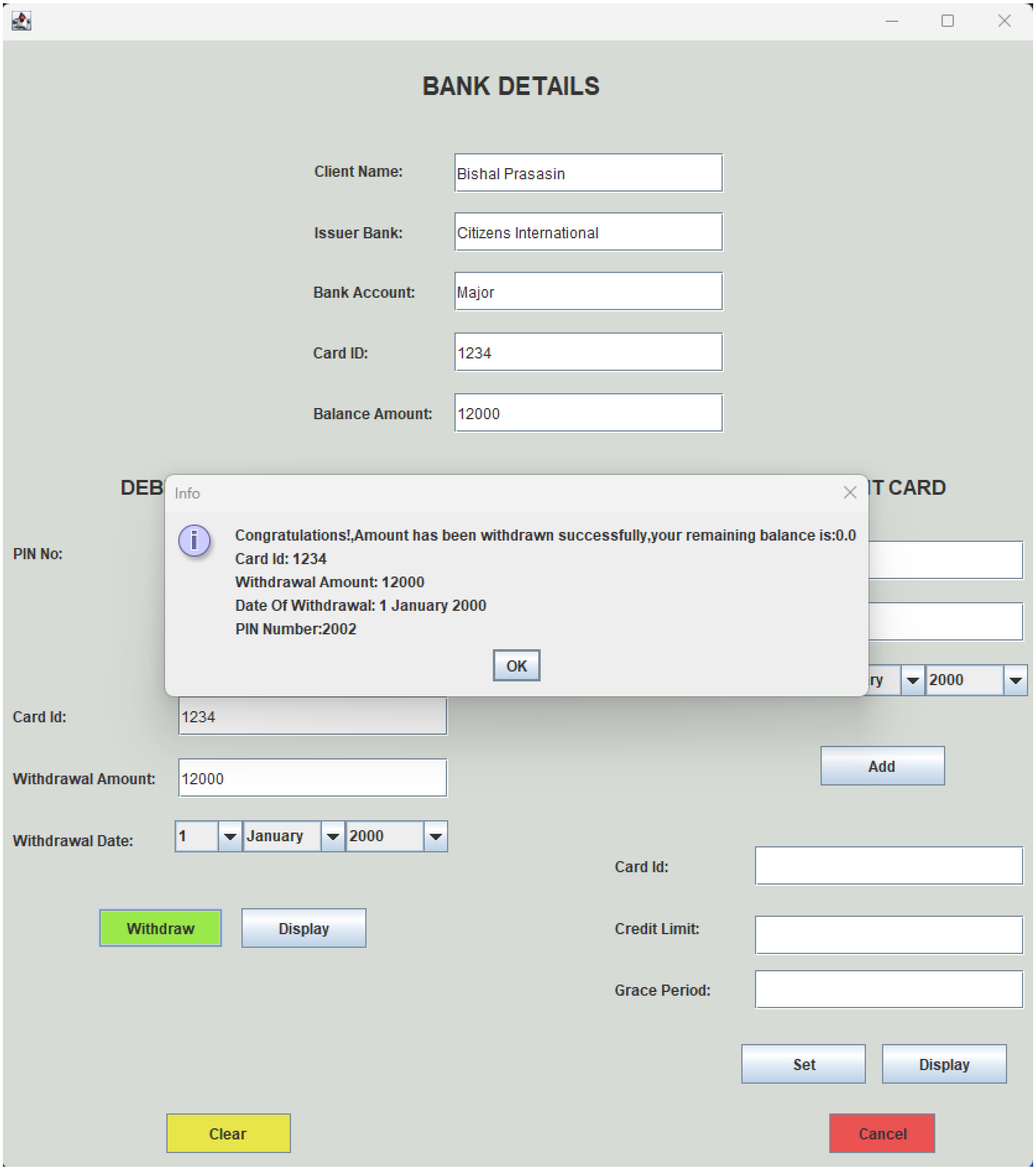
Figure 25:Figure of the outcome created by semantic error

Error detection: In fig 24 , line number 496,while validating withdrawal amout and initial amount,condition is checked using less than ‘<’ sign. After compiling and running the program, the output is somewhat unexpected.As above GUI clearly depicts, Balance Amount is 12000 and withdrawal amount is also 1200.with such values if the withdrawn process is carried out, the message is popped up saying insufficient balance which is precisely a semantic error.



```
int cardId = Integer.parseInt(cardidTxt_1.getText());
int withDrawalAmount = Integer.parseInt(withdrawalamtTxt.getText());
int pinNumber = Integer.parseInt(pinnumberTxt.getText());
String days_0 =(String) dayCombo_0.getSelectedItem();
String months_0 =(String) monthCombo_0.getSelectedItem();
String years_0 =(String) yearCombo_0.getSelectedItem();
String dateOfWithdrawal = days_0 + " " + months_0 + " "+ years_0;
if(bankCardsArraylists.isEmpty()){

    JOptionPane.showMessageDialog(frame,"Card has not been added","Error",JOptionPane.ERROR_MESSAGE);
}
else{ //if debit card is already present:
    //loop through the array list
    for(BankCard card:bankCardsArraylists){

        if(card instanceof DebitCard){
            //downcast
            DebitCard debitCard = (DebitCard) card;
            if(debitCard.getCardId() == cardId && debitCard.getPinNumber()==pinNumber){
                double initialAmount = debitCard.getBalanceAmount();
                debitCard.Withdraw(withDrawalAmount,dateOfWithdrawal,pinNumber);
                double remainingAmount = initialAmount - withDrawalAmount;
                if(withDrawalAmount>0){
                    if( withDrawalAmount<=initialAmount){
                        //JOptionPane.showMessageDialog(frame,"Congratulations!,Amount has been withdr
                        String popUp_0 ="Congratulations!,Amount has been withdrawn successfully,your
                        JOptionPane.showMessageDialog(frame,popUp_0,"Info", JOptionPane.INFORMATION_ME
                    }
```

Figure 26: Correction of semantic error



Client Name: Bishal Prasasin

Issuer Bank: Citizens International

Bank Account: Major

Card ID: 1234

Balance Amount: 12000

PIN No:

Card Id: 1234

Withdrawal Amount: 12000

Withdrawal Date: 1 January 2000

Card Id:

Credit Limit:

Grace Period:

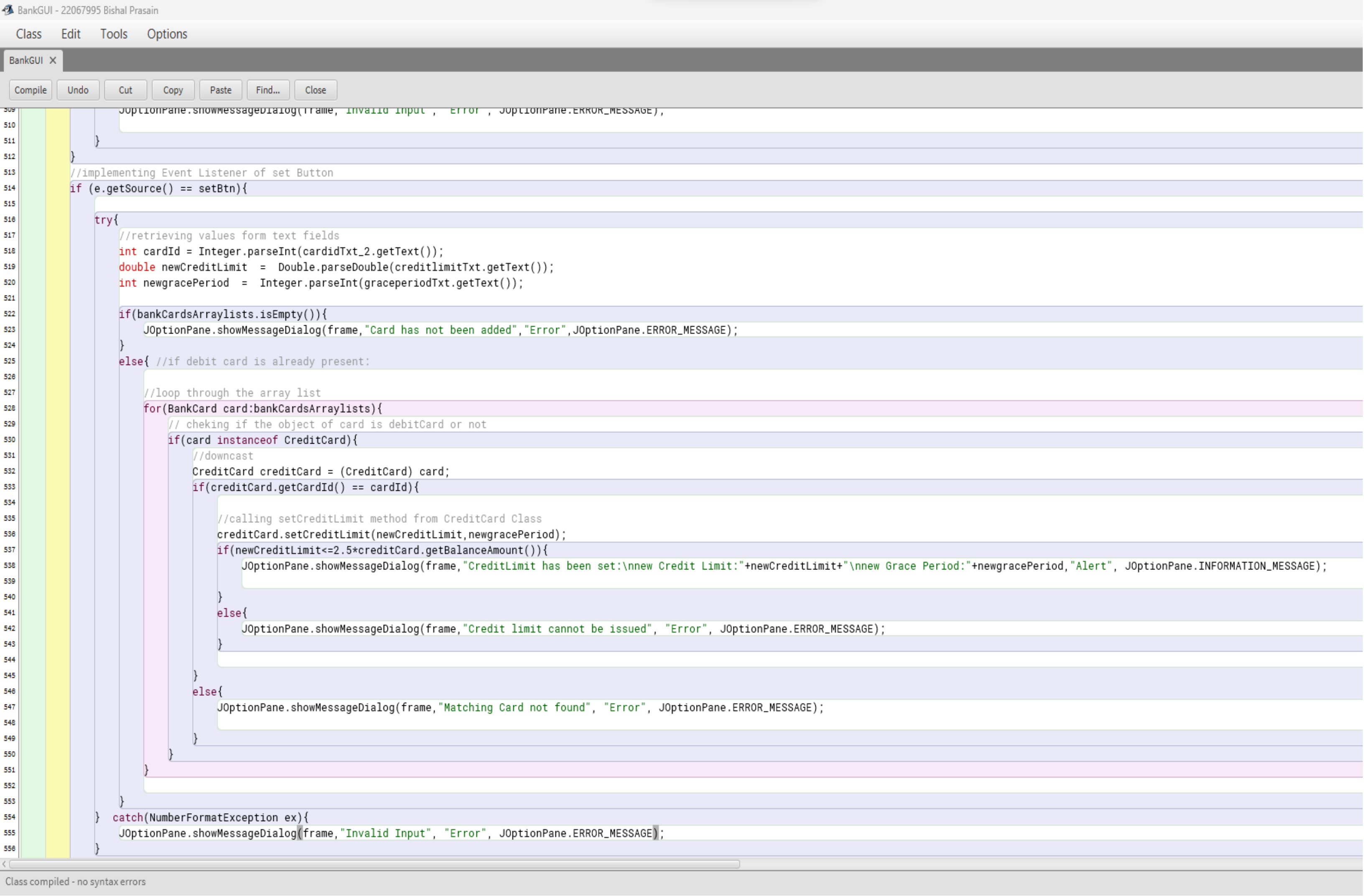
```
Congratulations!,Amount has been withdrawn successfully,your remaining balance is:0.0
Card Id: 1234
Withdrawal Amount: 12000
Date Of Withdrawal: 1 January 2000
PIN Number:2002
```

Figure 27:Figure of outcomes after fixing semantic error

Error Correcton: After using “<=” while validating withdrawal amount and initial amount, The outcome was as expected.

3)Logical Error:

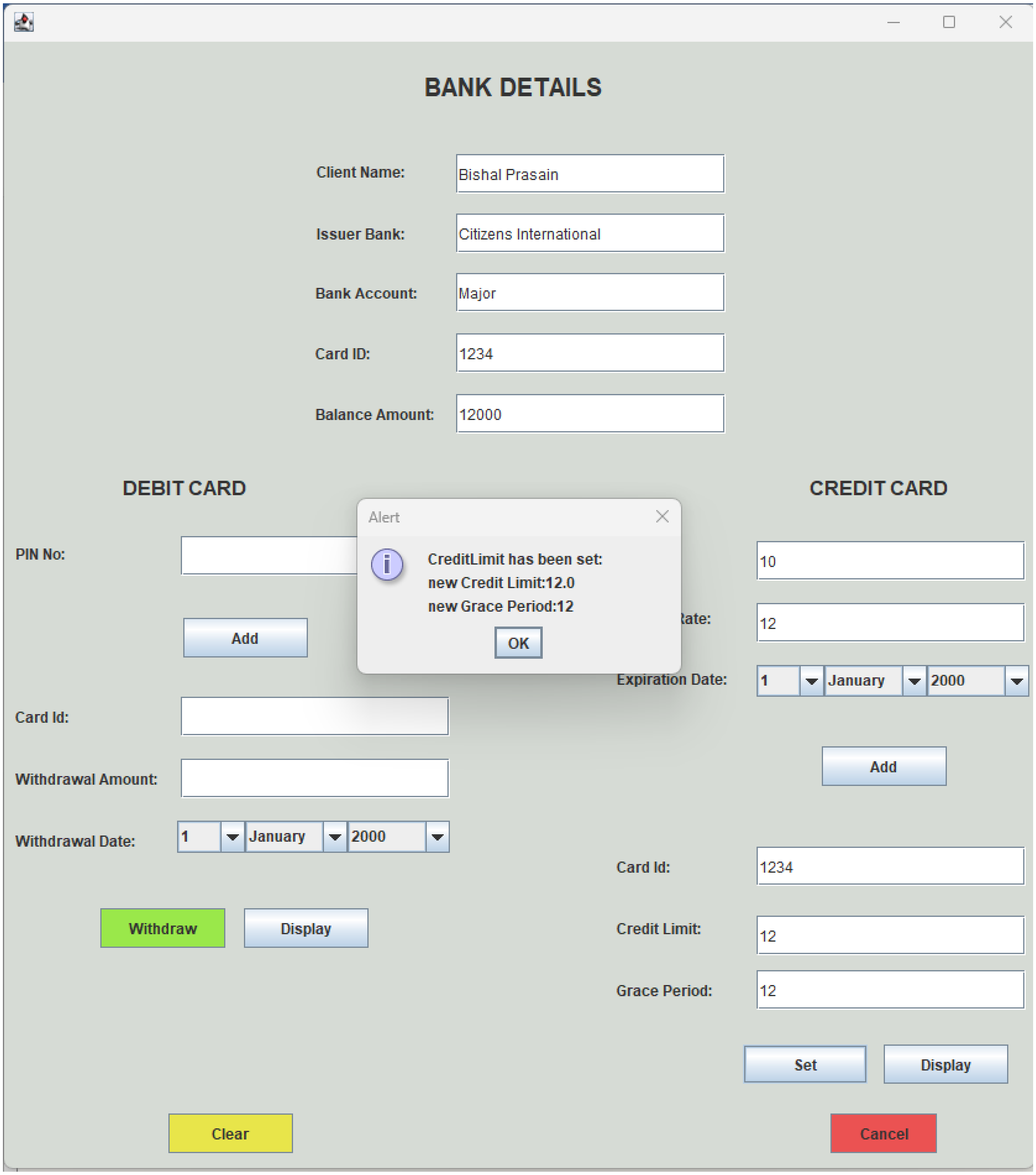
Logical errors are the errors that occurs when there is problems in logic of the particular program. It is hard to address logical errros as program can be executed even if there is logical errors.



BankGUI - 22067995 Bishal Prasain
Class Edit Tools Options
BankGUI ✕
Compile Undo Cut Copy Paste Find... Close

```
        JOptionPane.showMessageDialog(frame, "Invalid Input", "Error", JOptionPane.ERROR_MESSAGE);

    }
}
//implementing Event Listener of set Button
if (e.getSource() == setBtn){

    try{
        //retrieving values form text fields
        int cardId = Integer.parseInt(cardidTxt_2.getText());
        double newCreditLimit  =  Double.parseDouble(creditlimitTxt.getText());
        int newgracePeriod  =  Integer.parseInt(graceperiodTxt.getText());

        if(bankCardsArraylists.isEmpty()){
            JOptionPane.showMessageDialog(frame,"Card has not been added","Error",JOptionPane.ERROR_MESSAGE);
        }
        else{ //if debit card is already present:

            //loop through the array list
            for(BankCard card:bankCardsArraylists){
                // cheking if the object of card is debitCard or not
                if(card instanceof CreditCard){
                    //downcast
                    CreditCard creditCard = (CreditCard) card;
                    if(creditCard.getCardId() == cardId){

                        //calling setCreditLimit method from CreditCard Class
                        creditCard.setCreditLimit(newCreditLimit,newgracePeriod);
                        if(newCreditLimit<=2.5*creditCard.getBalanceAmount()){
                            JOptionPane.showMessageDialog(frame,"CreditLimit has been set:\nnew Credit Limit:"+newCreditLimit+"\nnew Grace Period:"+newgracePeriod,"Alert", JOptionPane.INFORMATION_MESSAGE);

                        }
                        else{
                            JOptionPane.showMessageDialog(frame,"Credit limit cannot be issued", "Error", JOptionPane.ERROR_MESSAGE);
                        }

                    }
                    else{
                        JOptionPane.showMessageDialog(frame,"Matching Card not found", "Error", JOptionPane.ERROR_MESSAGE);

                    }
                }
            }

        }
    }  catch(NumberFormatException ex){
        JOptionPane.showMessageDialog(frame,"Invalid Input", "Error", JOptionPane.ERROR_MESSAGE);
    }
```

Figure 28: Screeenshots of block of code before fixing logical error.



BANK DETAILS

Client Name: Bishal Prasain

Issuer Bank: Citizens International

Bank Account: Major

Card ID: 1234

Balance Amount: 12000

10

Rate: 12

Expiration Date: 1 January 2000

Card Id: 1234

Credit Limit: 12

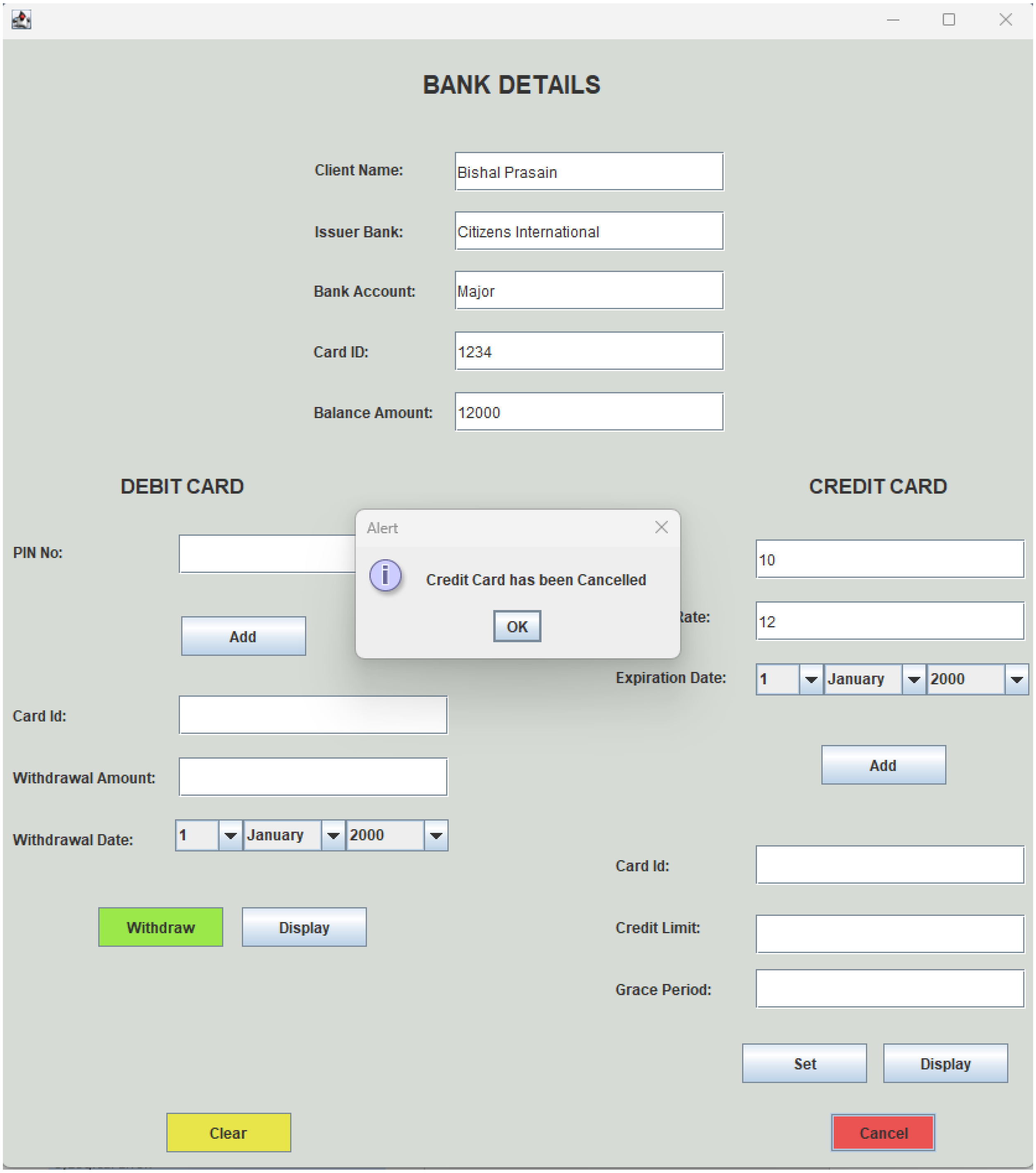
Grace Period: 12

Alert

CreditLimit has been set:
new Credit Limit:12.0
new Grace Period:12

OK

Figure 29: Figure of setting the Credit Limit



BANK DETAILS

Client Name: Bishal Prasain

Issuer Bank: Citizens International

Bank Account: Major

Card ID: 1234

Balance Amount: 12000

DEBIT CARD

PIN No:

Card Id:

Withdrawal Amount:

Withdrawal Date: 1 January 2000

CREDIT CARD

10

ate: 12

Expiration Date: 1 January 2000

Card Id:

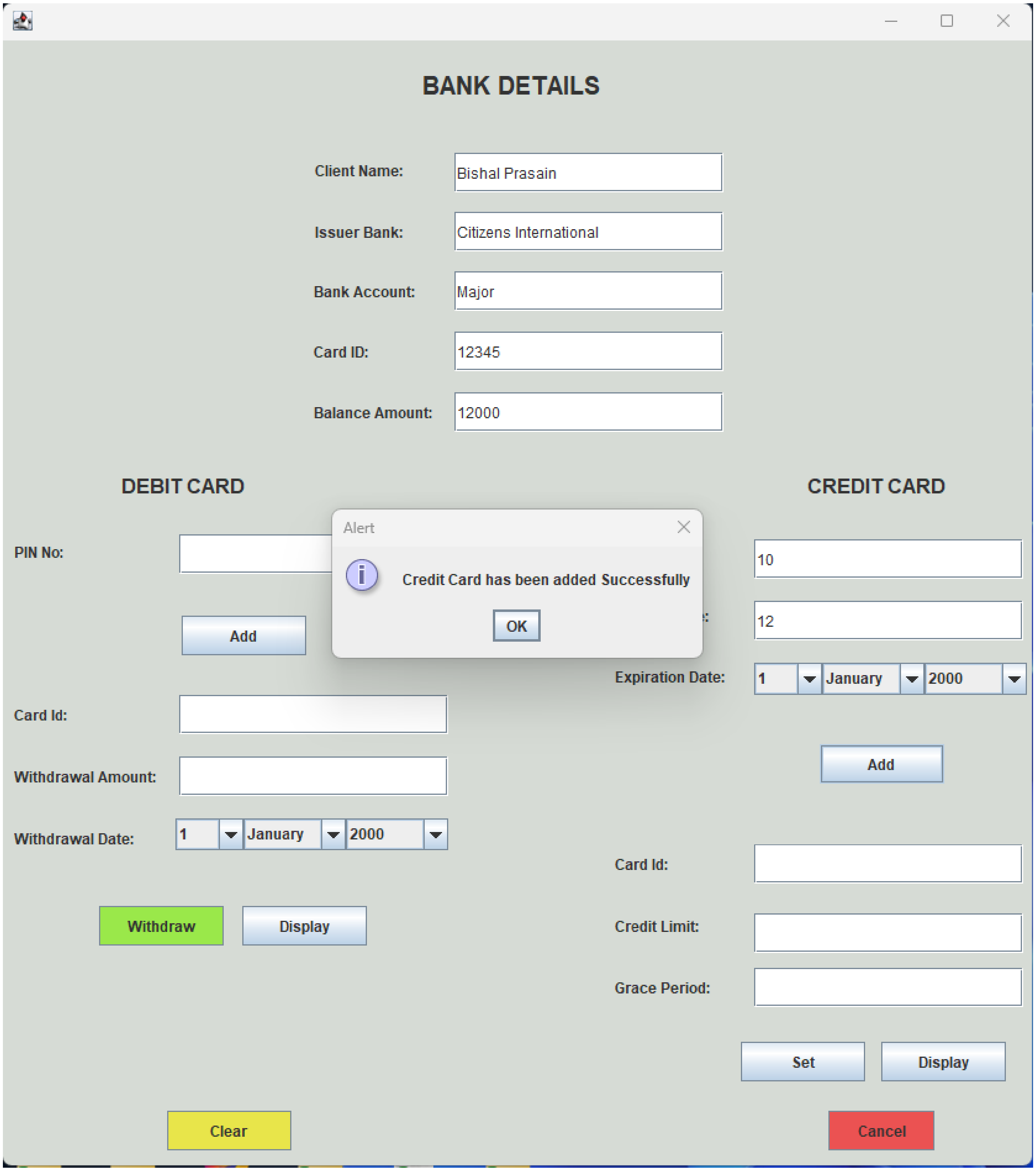
Credit Limit:

Grace Period:

Alert

Credit Card has been Cancelled

Figure 30:Figure of cancelling the Credit Limit



BANK DETAILS

Client Name: Bishal Prasain

Issuer Bank: Citizens International

Bank Account: Major

Card ID: 12345

Balance Amount: 12000

Alert

Credit Card has been added Successfully

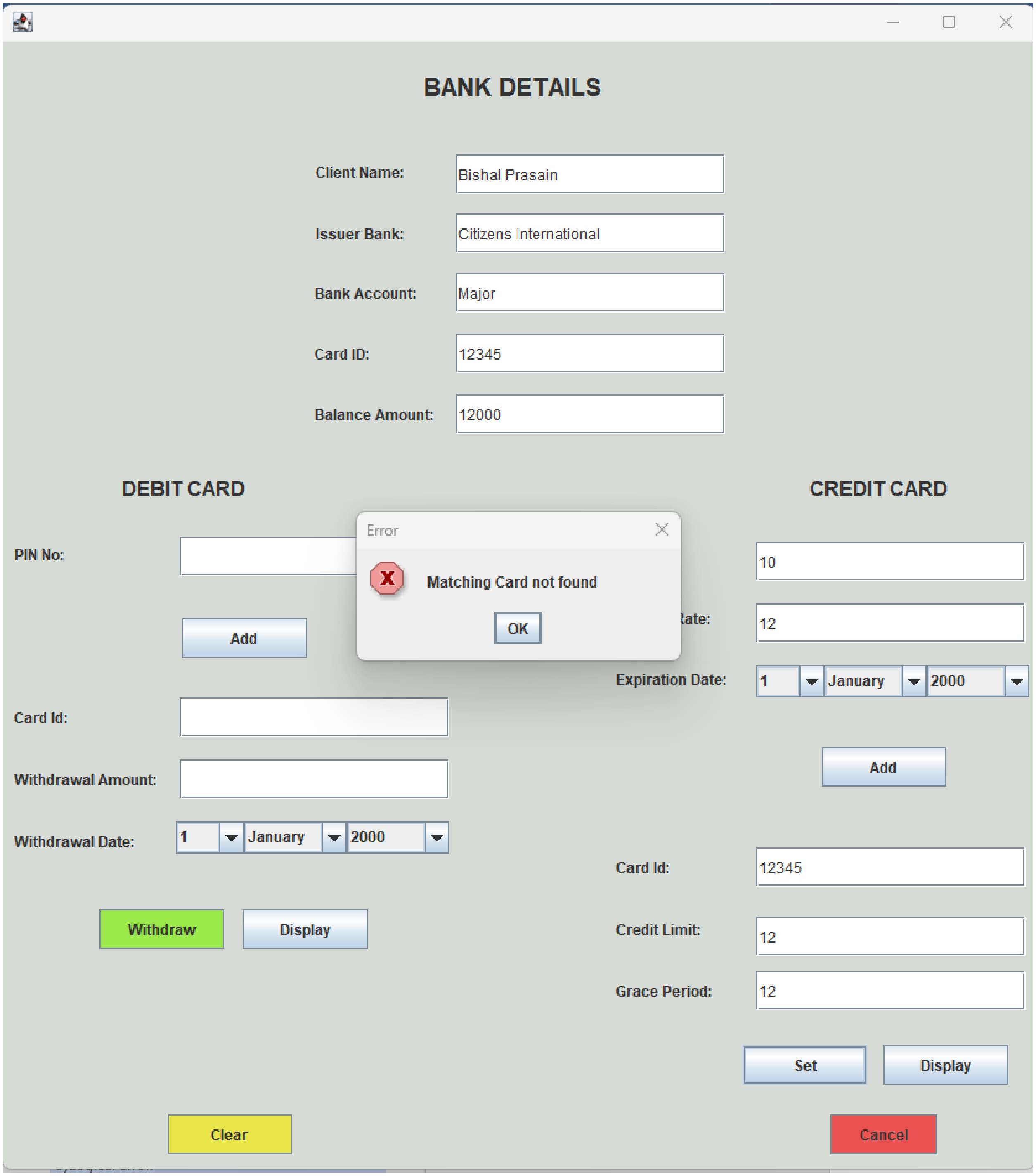
10

12

Expiration Date: 1 January 2000

Withdrawal Date: 1 January 2000

Figure 31:Figure of adding credit card again



BANK DETAILS

Client Name: Bishal Prasain

Issuer Bank: Citizens International

Bank Account: Major

Card ID: 12345

Balance Amount: 12000

DEBIT CARD

PIN No:

Card Id:

Withdrawal Amount:

Withdrawal Date: 1 January 2000

CREDIT CARD

10

ate: 12

Expiration Date: 1 January 2000

Card Id: 12345

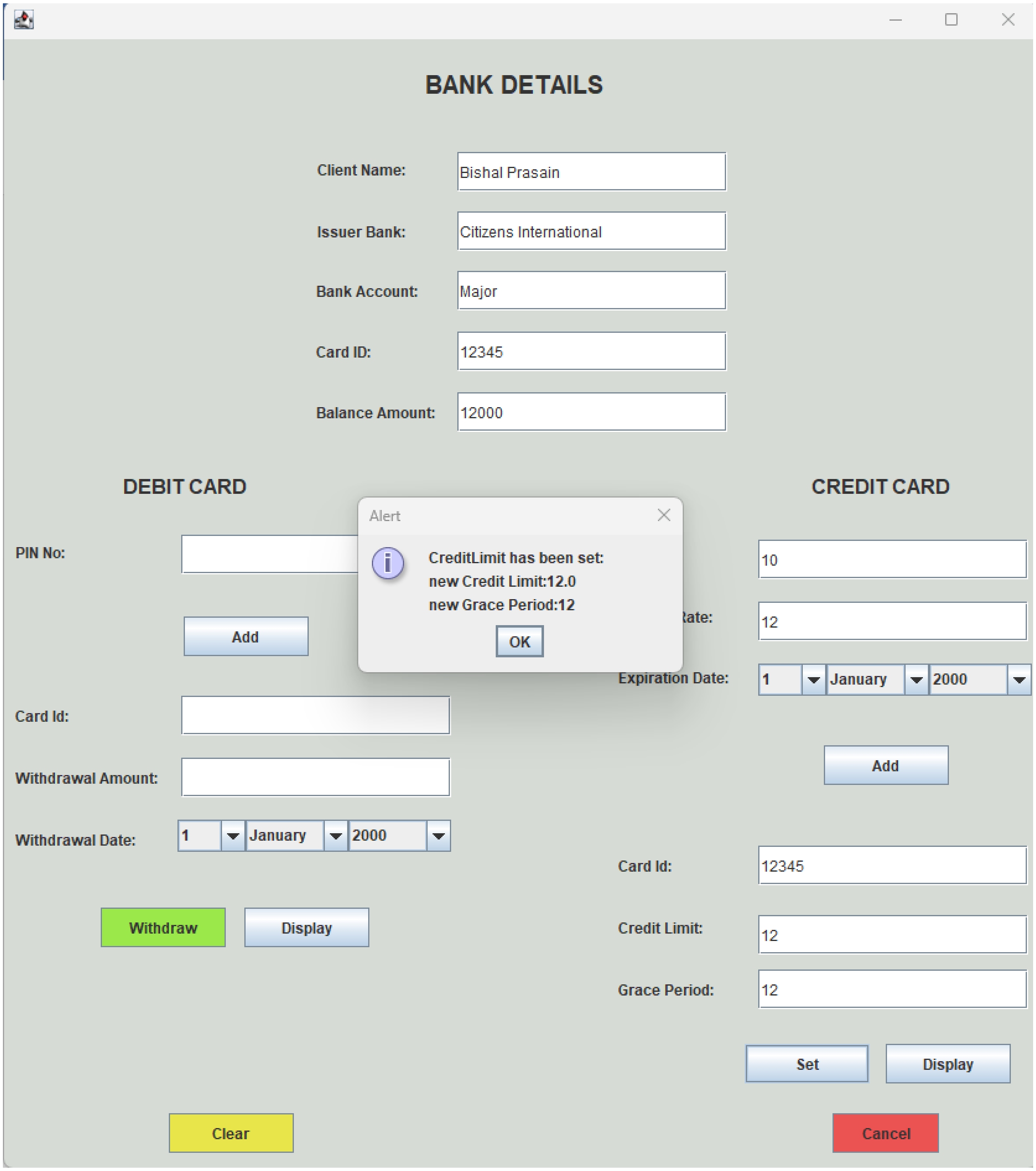
Credit Limit: 12

Grace Period: 12

Error

Matching Card not found

Figure 32: Figure of showing error message after setting Credit Card



BANK DETAILS

Client Name: Bishal Prasain

Issuer Bank: Citizens International

Bank Account: Major

Card ID: 12345

Balance Amount: 12000

DEBIT CARD

PIN No:

Card Id:

Withdrawal Amount:

Withdrawal Date: 1 January 2000

CREDIT CARD

10

ate: 12

Expiration Date: 1 January 2000

Card Id: 12345

Credit Limit: 12

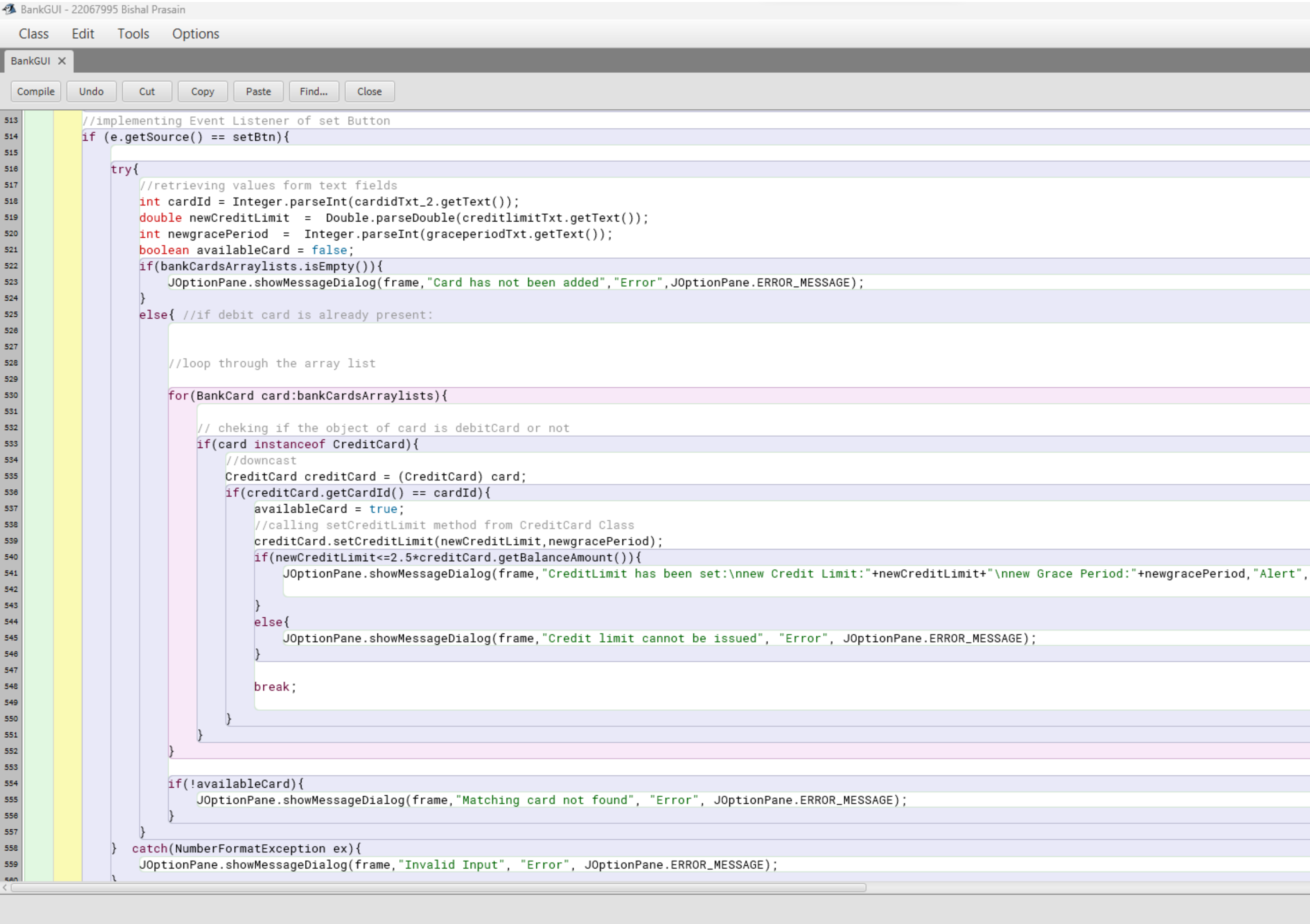
Grace Period: 12

Alert

CreditLimit has been set:
new Credit Limit:12.0
new Grace Period:12

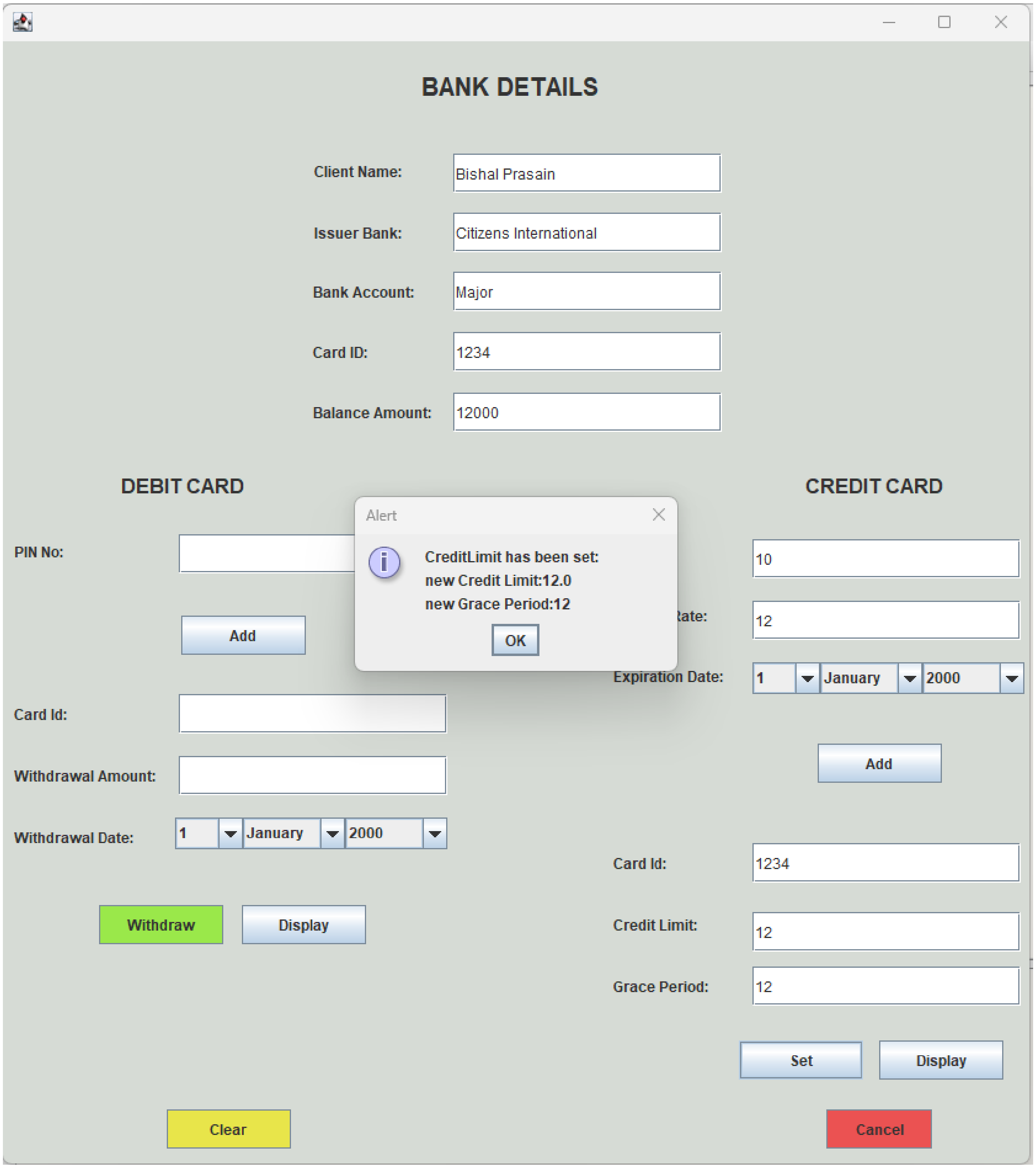
Figure 33:Figure of successful message all of sudden after having error message

Error detection: There is shere logical error in above provided code in fig no.28 . After adding Credit Card. The Credit Limit is set(fig no. 29). After that, CreditCard is cancelled(fig 30). Additionally new Credit Card is added.(fig 31) Then after entering the values for setting Credit Limit, even if Card Id is written correctly, it shows error message of credit limit not being set.(fig 32). But at a same time it shows the successful message.(fig 33).



```java
//implementing Event Listener of set Button
if (e.getSource() == setBtn){

    try{
        //retrieving values form text fields
        int cardId = Integer.parseInt(cardidTxt_2.getText());
        double newCreditLimit  =  Double.parseDouble(creditlimitTxt.getText());
        int newgracePeriod  =  Integer.parseInt(graceperiodTxt.getText());
        boolean availableCard = false;
        if(bankCardsArraylists.isEmpty()){
            JOptionPane.showMessageDialog(frame,"Card has not been added","Error",JOptionPane.ERROR_MESSAGE);
        }
        else{ //if debit card is already present:


            //loop through the array list

            for(BankCard card:bankCardsArraylists){

                // cheking if the object of card is debitCard or not
                if(card instanceof CreditCard){
                    //downcast
                    CreditCard creditCard = (CreditCard) card;
                    if(creditCard.getCardId() == cardId){
                        availableCard = true;
                        //calling setCreditLimit method from CreditCard Class
                        creditCard.setCreditLimit(newCreditLimit,newgracePeriod);
                        if(newCreditLimit<=2.5*creditCard.getBalanceAmount()){
                            JOptionPane.showMessageDialog(frame,"CreditLimit has been set:\nnew Credit Limit:"+newCreditLimit+"\nnew Grace Period:"+newgracePeriod,"Alert",

                        }
                        else{
                            JOptionPane.showMessageDialog(frame,"Credit limit cannot be issued", "Error", JOptionPane.ERROR_MESSAGE);
                        }

                        break;

                    }
                }
            }

            if(!availableCard){
                JOptionPane.showMessageDialog(frame,"Matching card not found", "Error", JOptionPane.ERROR_MESSAGE);
            }
        }
    }  catch(NumberFormatException ex){
        JOptionPane.showMessageDialog(frame,"Invalid Input", "Error", JOptionPane.ERROR_MESSAGE);
    }
```

Figure 34: Screenshots of Block of code after fixing logical error



BANK DETAILS

Client Name: Bishal Prasain

Issuer Bank: Citizens International

Bank Account: Major

Card ID: 1234

Balance Amount: 12000

DEBIT CARD

PIN No:

Card Id:

Withdrawal Amount:

Withdrawal Date: 1 January 2000

CREDIT CARD

10

...ate: 12

Expiration Date: 1 January 2000

Card Id: 1234

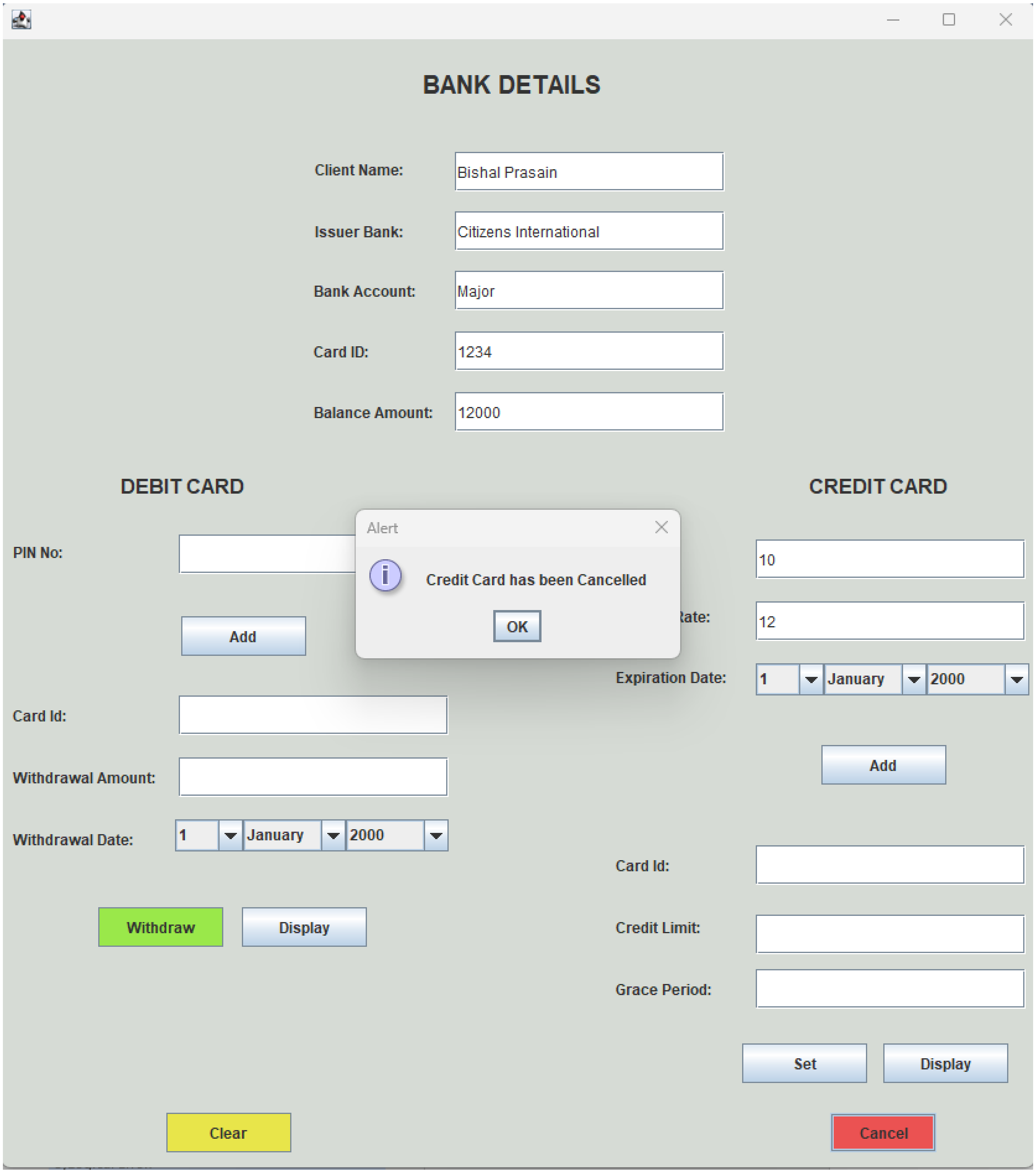
Credit Limit: 12

Grace Period: 12

Alert

CreditLimit has been set:
new Credit Limit:12.0
new Grace Period:12

Figure 35: Figure of setting credit limit



Client Name: Bishal Prasain

Issuer Bank: Citizens International

Bank Account: Major

Card ID: 1234

Balance Amount: 12000

10

ate: 12

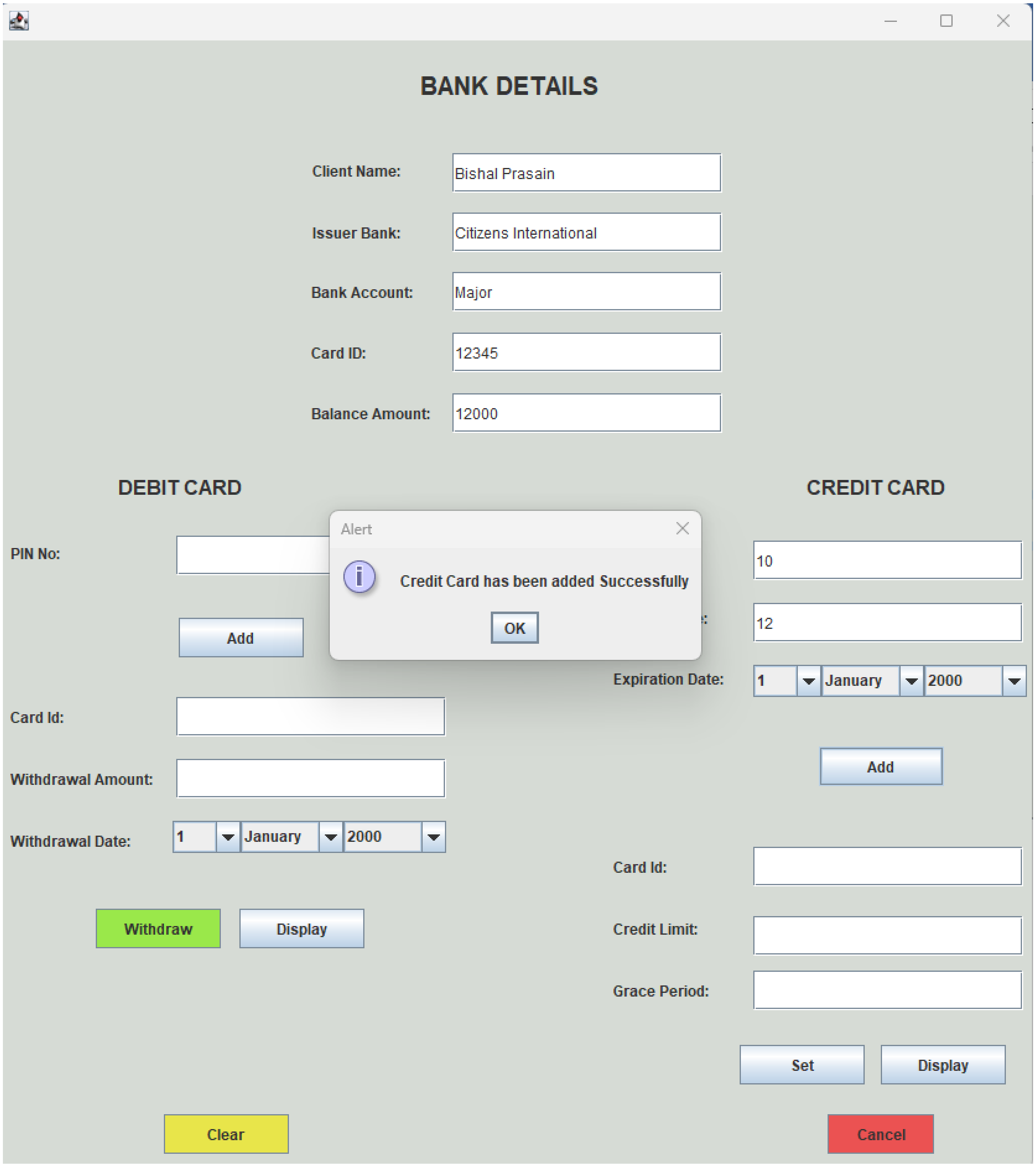
Expiration Date: 1 January 2000

Withdrawal Date: 1 January 2000

Alert

Credit Card has been Cancelled

Figure 36: Figure of cancelling Credit Card



BANK DETAILS

Client Name: Bishal Prasain

Issuer Bank: Citizens International

Bank Account: Major

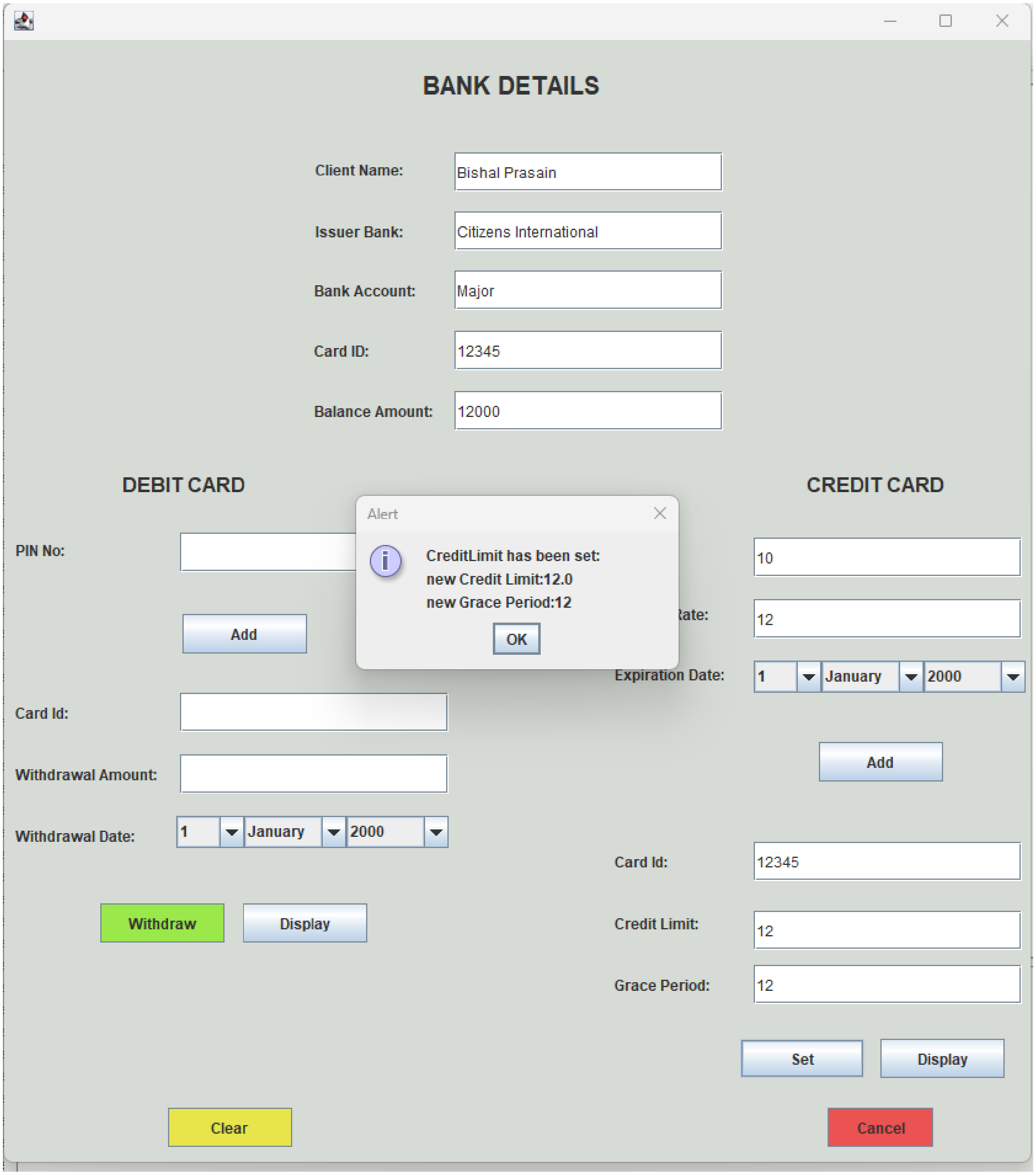
Card ID: 12345

Balance Amount: 12000

Alert

Credit Card has been added Successfully

Figure 37: Figure of adding Credit Card.



BANK DETAILS

Client Name: Bishal Prasain

Issuer Bank: Citizens International

Bank Account: Major

Card ID: 12345

Balance Amount: 12000

DEBIT CARD

PIN No:

Card Id:

Withdrawal Amount:

Withdrawal Date: 1 January 2000

CREDIT CARD

10

ate: 12

Expiration Date: 1 January 2000

Card Id: 12345

Credit Limit: 12

Grace Period: 12

Alert

CreditLimit has been set:
new Credit Limit:12.0
new Grace Period:12

Figure 38:Figure of setting credit limit

Error Correction: This logical errror was fixed after using the flag just to examine if any CreditCard object with particular Card has been found in the arrayList(fig no. 35) and perform operations accordingly.

Conclusion

The GUI(Graphical User Interface) based program follows an OOPs approach by applying different properties like polymorphism as well as inheritance which helps in a proper modulation and reuse of the program. This simple illustration of how Banking operations are being carried out in real world helps to conceptualize and visualize the actual working mechanism. This project has covered an optimum sector like: GUI based approach, implementing various properties of OOPs etc.

Most significantly, the tactic to design a user friendly interface for a Banking field can be acquired from this project. Similarly, this project shows a good instance of how an invalid data can be handled and fixed accordingly. The vital role of Object Oriented Programming in managing a system as a whole is illustrated in this project.

Reviewing every lines of code, handling every exception, upgrading the functionality of user interface,addressing runtime errors, it was quite difficult to cope with every predicament. It was difficult to make the user interface more flexible and dynamic because of the errors faced time and again.

By consistently working on this project, evaluating and analysing the required criteria for the successful completion of it, all major difficulties were addressed and sorted.

References

What is Java? (2023, April 27). Retrieved from ibm.com: https://www.ibm.com/topics/java

Moqups Pricing, Features, Reviews & Alternatives. (2023). Retrieved from getapp.com: https://www.getapp.com/collaboration-software/a/moqups/

What is a syntax Error? - Definition from Techopedia. (2023). Retrieved from techopedia.com: https://www.techopedia.com/definition/13391/syntax-error

What is MS Word?- Basics, Uses, Features & Questions. (2023). Retrieved from byjus.com: https://byjus.com/govt-exams/microsoft-word/

What is Syntax Error - Definition from Techopedia. (2023). Retrieved from techopedia.com: https://www.techopedia.com/definition/13391/syntax-error

Appendix

1.BankCard

```
/**
 * Write a description of class BankCard here.
 *
 * @author (22067995 Bishal Prasain)
 * @version (1.0.0)
 */
public class BankCard
{
   //Attribute for BankCard class
   private int cardId;
   private String clientName;
   private String issuerBank;
   private String bankAccount;
   private double BalanceAmount;
   //constructor
   public BankCard(double BalanceAmount,int cardId,String bankAccount,String issuerBank)
   {
      //Assigning attributes with parameter values
      this.clientName=""; //initializing to an empty String;
      this.BalanceAmount=BalanceAmount;
      this.cardId=cardId;
      this.bankAccount=bankAccount;
      this.issuerBank=issuerBank;
```

```
}

 public int getCardId(){
        return cardId;
}

  public String getClientName(){
        return clientName;
}

public void setClientName(String clientName){
         this.clientName=clientName;
    }

public double getBalanceAmount(){
        return BalanceAmount;
    }

public void setBalanceAmount(double BalanceAmount){
        this.BalanceAmount=BalanceAmount;
    }

public void display(){
        System.out.println("Card ID=" +this.cardId);
        System.out.println("Issuer Bank=: " +this.issuerBank);
        System.out.println("Bank Account=: " +this.bankAccount);
        System.out.println("Balance Amount=: " +this.BalanceAmount);
        if(clientName==""){//Checking if clientName equals null value
```

```
            System.out.println("oops!! you must have client name assigned");
         }
         else{
            System.out.println("Client Name:"+this.clientName);
         }
      }
   }
```

2.DebitCard

```
/**
 * Write a description of class DebitCard here.
 *
 * @author (Bishal Prasain)
 * @version (a version number or a date)
 */
public class DebitCard extends BankCard
{   //Attributes
    private int pinNumber;
    private int withDrawalAmount;
    private String dateOfWithdrawal;
    private boolean hasWithdrawn;


    public DebitCard(double BalanceAmount,int cardId,String bankAccount,String
issuerBank,
     String clientName,int pinNumber){

     super(BalanceAmount,cardId,bankAccount,issuerBank);//Inheriting
constructor from parent class
     super.setClientName(clientName);//Inheriting mutator
setClientName(clientName) from parent class
     //Assigning attributes with parameter values
     this.pinNumber = pinNumber;
     this.hasWithdrawn = false;
     }


    public int getPinNumber(){
        return pinNumber;
    }
    public int getWithDrawalAmount(){
        return withDrawalAmount;
    }

    public String getDateOfWithDrawal(){
        return dateOfWithdrawal;
    }

    public boolean getHasWithdrawn(){
        return hasWithdrawn;
    }


    public void setWithDrawalAmount(int withDrawalAmount){
        this.withDrawalAmount=withDrawalAmount;
    }

    public void Withdraw(int withDrawalAmount,String dateOfWithdrawal,int
pinNumber){
        if (this.pinNumber == pinNumber){//Checking if the pin number passed as
parameters to Withdraw method equalizes the pin number of the current class
            if(withDrawalAmount <= super.getBalanceAmount()){//Checking if
withDrawalAmount is less or equal to BalanceAmount of Parent class
            this.hasWithdrawn = true;
            super.setBalanceAmount(super.getBalanceAmount()-
withDrawalAmount);//Setting new balance amount with the help of
mutator(overiding)
            this.withDrawalAmount = withDrawalAmount;
            this.dateOfWithdrawal = dateOfWithdrawal;

            System.out.println("Successfully withdrawn,Remaining Amount:"
+super.getBalanceAmount());
            }
                else{
                    System.out.println("You have insufficient Balance");
                }
            } else{
                System.out.println("You have entered invalid pinnumber");
            }
        }
    //method
    public void display(){
        super.display();//inheriting display method from parent class
```

```
            System.out.println("Pin Number:"+this.pinNumber);
            if(hasWithdrawn == true){//Verifying if withdrawn process has been carried
    out.
               System.out.println("Withdrawal Amount:"+this.withDrawalAmount);
               System.out.println("Date Of Withdrawal:"+this.dateOfWithdrawal);
               }
               else{
                   System.out.println("No any withdrawal has been made so far,Balance
    Amount:"+super.getBalanceAmount());
               }
       }
    }
```

3.CreditCard

```
/**
 * Write a description of class CreditCard here.
 *
 * @author (22067995 Bishal Prasain )
 * @version (a version number or a date)
 */
public class  CreditCard extends BankCard
{
   //Attribute
   private int CVCNumber;
   private double creditLimit;
   private double interestRate;
   private String expirationDate;
   private int gracePeriod;
   private boolean isGranted;
   //constructor
```

```
    public CreditCard(double BalanceAmount,int cardId,String bankAccount,String issuerBank,String clientName,int CVCNumber,double interestRate,String
    expirationDate){
        super(BalanceAmount,cardId,bankAccount,issuerBank);//Inheriting Constructor from parent clas
        super.setClientName(clientName);//Calling settermethod from parent class
        //Assigning attributes with parameter values
        this.CVCNumber=CVCNumber;
        this.interestRate=interestRate;
        this.expirationDate=expirationDate;
        this.isGranted=false;
        }

    //Accessor methods
    public int getCVcNUmber(){
        return CVCNumber;
    }

    public double getCreditLimit(){
        return creditLimit ;
    }

    public double getInterestRate(){
        return interestRate;
    }

    public String getExpirationDate(){
        return expirationDate;
    }
    public int getGracePeriod(){
        return gracePeriod;
    }

    public boolean getIsGranted(){
        return isGranted;
    }

    public void setCreditLimit(double newcreditLimit,int newgracePeriod){
        if(newcreditLimit<=(2.5*super.getBalanceAmount())){//Checking if local variable
newcreditlimit is less or equal to 2.5 times the current Balanceamount called from
parent class
            this.creditLimit=newcreditLimit;
            this.gracePeriod=newgracePeriod;
            this.isGranted=true;
            System.out.println("The credit has been issued:"+newcreditLimit);
        }else{
            System.out.println("The credit amount cannot be granted to the client");
        }
    }

    public void cancelCreditcard(){
        this.CVCNumber=0;
        this.creditLimit=0;
        this.gracePeriod=0;
        this.isGranted=false;
    }
```

```
   public void display(){
       super.display();//Calling method display of super class BankCard
       if(isGranted==true){//verifying if isGranted is equal to "true"
           System.out.println("Credit Limit:"+this.creditLimit);
           System.out.println("Grace Period:"+this.gracePeriod);
           System.out.println("Expiration Date:"+this.expirationDate);
           System.out.println("Interest Rate:"+this.interestRate);
           System.out.println("CVC Number:"+this.CVCNumber);
       }
       else{
           System.out.println("Credit limit cannot be granted");
       }
   }

}
```

4.BankGUI

```
import java.util.*;
import javax.swing.*;
import java.awt.event.*;
import java.awt.*;

/**
 * Write a description of class BankGUI here.
 *
 * @author (22067995 Bishal Prasain)
 * @version (a version number or a date)
 */
public class BankGUI implements ActionListener
{
    private JFrame frame;
    //scooping ComboBox
    private JComboBox
dayCombo_0,monthCombo_0,yearCombo_0,dayCombo_1,monthCombo_1,year
Combo_1;

    //scooping common components
    private JLabel
clientnameLabel,issuerbankLabel,bankaccountLabel,cardidLabel_0,balanceamtL
abel;
    private JTextField
clientnameTxt,issuerbankTxt,bankaccountTxt,cardidTxt_0,balanceamtTxt;

    //scooping components of DebitCard
    private JLabel
pinnumberLabel,cardidLabel_1,withdrawalamtLabel,withdrawaldateLabel;
    private JTextField pinnumberTxt,cardidTxt_1,withdrawalamtTxt;

    //scooping components of CreditCard
    private JLabel titleLabel_0,titleLabel_1,titleLabel_2,
cvcnumberLabel,interestrateLabel,ExpirationdateLabel,cardidLabel_2,creditlimitL
abel,graceperiodLabel;
    private JTextField
cvcnumberTxt,interestrateTxt,cardidTxt_2,creditlimitTxt,graceperiodTxt;

    //scooping Buttons
    private JButton
addBtn_0,displayBtn_0,displayBtn_1,addBtn_1,withdrawBtn,clearBtn,cancelBtn,s
etBtn;

    //ArrayList of BankCard
    ArrayList<BankCard> bankCardsArraylists = new ArrayList<BankCard>();

    public BankGUI(){
        //setting label,Textfield, and Buttons

        frame = new JFrame();
        titleLabel_0 = new JLabel("BANK DETAILS");
        titleLabel_1 = new JLabel("DEBIT CARD");
        titleLabel_2 = new JLabel("CREDIT CARD");

        //Setting Font Size of Titles
        Font font_0 = new Font("Arial",Font.BOLD,20);
        Font font_1 = new Font("Arial",Font.BOLD,16);
        Font font_2 = new Font("Arial",Font.BOLD,16);

        clientnameLabel = new JLabel("Client Name:");
        clientnameTxt = new JTextField();

        issuerbankLabel = new JLabel("Issuer Bank:");
        issuerbankTxt = new JTextField();

        bankaccountLabel = new JLabel("Bank Account:");
        bankaccountTxt = new JTextField();

        cardidLabel_0 = new JLabel("Card ID:");
        cardidTxt_0 = new JTextField();

        balanceamtLabel = new JLabel("Balance Amount:");
        balanceamtTxt = new JTextField();

        pinnumberLabel = new JLabel("PIN No:");
        pinnumberTxt = new JTextField();

        cardidLabel_1 = new JLabel("Card Id:");
        cardidTxt_1 = new JTextField();

        withdrawalamtLabel = new JLabel("Withdrawal Amount:");
        withdrawalamtTxt = new JTextField();

        cvcnumberLabel = new JLabel("CVC No:");
        cvcnumberTxt = new JTextField();

        interestrateLabel = new JLabel("Interest Rate:");
        interestrateTxt = new JTextField();

        cardidLabel_2 = new JLabel("Card Id:");
        cardidTxt_2 = new JTextField();

        creditlimitLabel = new JLabel("Credit Limit:");
        creditlimitTxt = new JTextField();

        graceperiodLabel = new JLabel("Grace Period:");
        graceperiodTxt = new JTextField();

        withdrawaldateLabel = new JLabel("Withdrawal Date:");
```

```
        String[] days_0 =
{"1","2","3","4","5","6","7","8","9","10","12","13","14","15","16","17","18","19","20","
21","22","23","24","25","26","27","28","29","30","31","32"};
        String[] months_0 = {"January", "february", "March", "April", "May", "June",
"July", "August", "September", "October", "November", "December"};
        String[] years_0 =
{"2000","2001","2002","2003","2004","2005","2006","2007","2008","2009","2010",
"2011","2012","2013","2014","2015","2016","2017","2018","2019","2020","2021","
2022","2023","2024"};

        //ComboBox for withdrawDate
        dayCombo_0 = new JComboBox(days_0);
        monthCombo_0 = new JComboBox(months_0);
        yearCombo_0 = new JComboBox(years_0);

        JLabel expirationdateLabel = new JLabel("Expiration Date:");
        String[] days_1 =
{"1","2","3","4","5","6","7","8","9","10","12","13","14","15","16","17","18","19","20","
21","22","23","24","25","26","27","28","29","30","31","32"};
        String[] months_1 = {"January", "february", "March", "April", "May", "June",
"July", "August", "September", "October", "November", "December"};
        String[] years_1 = {"2024","2025","2026","2027","2028","2029","2030"};

        dayCombo_1 = new JComboBox(days_1);
        monthCombo_1 = new JComboBox(months_1);
        yearCombo_1 = new JComboBox(years_1);

        displayBtn_0 = new JButton("Display");
        displayBtn_1 = new JButton("Display");
        addBtn_0 = new JButton("Add");
        addBtn_1 = new JButton("Add");

        withdrawBtn = new JButton("Withdraw");
        withdrawBtn.setBackground(new Color(154, 232, 74));
        clearBtn = new JButton("Clear");
        clearBtn.setBackground(new Color(232, 229, 74));

        cancelBtn = new JButton("Cancel");
        cancelBtn.setBackground(new Color(235, 82, 82));
        setBtn = new JButton("Set");

        titleLabel_0.setBounds(340,27,200,20);
        titleLabel_0.setFont(font_0);
```

```
titleLabel_1.setBounds(96,351,200,20);
titleLabel_1.setFont(font_1);
titleLabel_2.setBounds(652,351,200,20);
titleLabel_2.setFont(font_2);

clientnameLabel.setBounds(253,95,81,20);
clientnameTxt.setBounds(366,91,218,32);

issuerbankLabel.setBounds(253,145,78,20);
issuerbankTxt.setBounds(366,139,218,32);

bankaccountLabel.setBounds(252,193,90,20);
bankaccountTxt.setBounds(366,187,218,32);

cardidLabel_0.setBounds(252,242,50,20);
cardidTxt_0.setBounds(366,236,218,32);

balanceamtLabel.setBounds(252,291,106,20);
balanceamtTxt.setBounds(366,285,218,32);

pinnumberLabel.setBounds(9,404,80,20);
pinnumberTxt.setBounds(143,400,218,32);

cardidLabel_1.setBounds(9,536,50,20);
cardidTxt_1.setBounds(143,530,218,32);

withdrawalamtLabel.setBounds(9,586,124,20);
withdrawalamtTxt.setBounds(143,580,218,32);

cvcnumberLabel.setBounds(496,408,59,20);
cvcnumberTxt.setBounds(609,404,218,32);

interestrateLabel.setBounds(496,456,84,20);
interestrateTxt.setBounds(609,454,218,32);

expirationdateLabel.setBounds(496,503,101,24);

cardidLabel_2.setBounds(496,657,50,20);
cardidTxt_2.setBounds(609,651,218,32);

creditlimitLabel.setBounds(496,707,75,20);
creditlimitTxt.setBounds(609,707,218,32);
```

```
graceperiodLabel.setBounds(496,757,86,20);
graceperiodTxt.setBounds(609,751,218,32);

withdrawaldateLabel.setBounds(9,636,105,20);

dayCombo_0.setBounds(140,630,55,26);
monthCombo_0.setBounds(195,630,83,26);
yearCombo_0.setBounds(278,630,83,26);

dayCombo_1.setBounds(609,504,55,26);
monthCombo_1.setBounds(664,504,83,26);
yearCombo_1.setBounds(747,504,83,26);

displayBtn_0.setBounds(194,701,101,32);
displayBtn_1.setBounds(712,811,101,32);
addBtn_0.setBounds(145,466,101,32);
addBtn_1.setBounds(662,570,101,32);
withdrawBtn.setBounds(78,701,101,32);
clearBtn.setBounds(133,867,101,32);
cancelBtn.setBounds(669,867,86,32);
setBtn.setBounds(598,811,101,32);

displayBtn_0.addActionListener(this);
displayBtn_1.addActionListener(this);
addBtn_0.addActionListener(this);
addBtn_1.addActionListener(this);
withdrawBtn.addActionListener(this);
clearBtn.addActionListener(this);
cancelBtn.addActionListener(this);
setBtn.addActionListener(this);

frame.add(titleLabel_0);
frame.add(titleLabel_1);
frame.add(titleLabel_2);
frame.add(clientnameLabel);
frame.add(clientnameTxt);
frame.add(issuerbankLabel);
frame.add(issuerbankTxt);
frame.add(bankaccountLabel);
frame.add(bankaccountTxt);
frame.add(cardidLabel_0);
frame.add(cardidTxt_0);
frame.add(balanceamtLabel);
```

```
        frame.add(balanceamtTxt);
        frame.add(pinnumberLabel);
        frame.add(pinnumberTxt);
        frame.add(cardidLabel_1);
        frame.add(cardidTxt_1);
        frame.add(withdrawalamtLabel);
        frame.add(withdrawalamtTxt);

        frame.add(cvcnumberLabel);
        frame.add(cvcnumberTxt);
        frame.add(interestrateLabel);
        frame.add(interestrateTxt);
        frame.add(cardidLabel_2);
        frame.add(cardidTxt_2);
        frame.add(creditlimitLabel);
        frame.add(creditlimitTxt);
        frame.add(graceperiodLabel);
        frame.add(graceperiodTxt);

        frame.add(withdrawaldateLabel);

        frame.add(dayCombo_0);
        frame.add(monthCombo_0);
        frame.add(yearCombo_0);

        frame.add(expirationdateLabel);

        frame.add(dayCombo_1);
        frame.add(monthCombo_1);
        frame.add(yearCombo_1);

        frame.add(addBtn_0);
        frame.add(addBtn_1);
        frame.add(displayBtn_0);
        frame.add(displayBtn_1);
        frame.add(withdrawBtn);
        frame.add(clearBtn);
        frame.add(cancelBtn);
        frame.add(setBtn);

        frame.setSize(850,950);
        frame.getContentPane().setBackground(new Color(214, 219, 213));
        frame.setLayout(null);
        frame.setVisible(true);
        frame.setResizable(false);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

    }

    //main method for creating an object of class BankGUI
    public static void main(String[] args){
        BankGUI obj = new BankGUI();
    }

    //event Listener
    public void actionPerformed(ActionEvent e){
        //implementing Event Listener of add button of Debit Card
        if (e.getSource()== addBtn_0){

            if(balanceamtTxt.getText().isEmpty() ||
cardidTxt_0.getText().isEmpty()||bankaccountTxt.getText().isEmpty()||
issuerbankTxt.getText().isEmpty()|| clientnameTxt.getText().isEmpty()||
pinnumberTxt.getText().isEmpty()){
                JOptionPane.showMessageDialog(frame,"Text field cannot be empty",
"Input Error", JOptionPane.ERROR_MESSAGE);
            }
            else{
                try{
                    ////getting input values from text fields
                    double BalanceAmount =
Double.parseDouble(balanceamtTxt.getText());
                    int cardId = Integer.parseInt(cardidTxt_0.getText());
                    String bankAccount = bankaccountTxt.getText();
                    String issuerBank = issuerbankTxt.getText();
                    String clientName = clientnameTxt.getText();
                    int pinNumber = Integer.parseInt(pinnumberTxt.getText());
                    //adding flag to keep track of an object
                    boolean toAdd = true;
                    DebitCard debitCard;

                    if(bankCardsArraylists.isEmpty()){
                        debitCard = new
DebitCard(BalanceAmount,cardId,bankAccount,issuerBank,clientName,pinNumb
er);
                        bankCardsArraylists.add(debitCard);
                    JOptionPane.showMessageDialog(frame,"Debit Card has been
added Successfully", "Alert", JOptionPane.INFORMATION_MESSAGE);
                }

                else{
                    for(BankCard card:bankCardsArraylists){

                        if(card instanceof DebitCard){
                            debitCard = (DebitCard) card;

                            if(debitCard.getCardId()==cardId){
                                JOptionPane.showMessageDialog(frame,"DebitCard
cannot be added", "Input Error", JOptionPane.ERROR_MESSAGE);
                                toAdd = false;
                                break;
                            }
                        }
                        else{
                            JOptionPane.showMessageDialog(frame,"Debit Card cannot
be added", "Input Error", JOptionPane.ERROR_MESSAGE);
                            toAdd = false;
                            break;
                        }
                    }

                    if(toAdd == true){
                        debitCard = new
DebitCard(BalanceAmount,cardId,bankAccount,issuerBank,clientName,pinNumb
er);
                        bankCardsArraylists.add(debitCard);
                        JOptionPane.showMessageDialog(frame,"Debit Card has been
added Successfully", "Alert", JOptionPane.INFORMATION_MESSAGE);
                    }

                }

            }
            catch(NumberFormatException ex){
                JOptionPane.showMessageDialog(frame,"Invalid Input", "Input
Error", JOptionPane.ERROR_MESSAGE);
            }
        }
    }
    //implementing Event Listener of add Button of credit Card
    if (e.getSource()== addBtn_1){

if(cardidTxt_0.getText().isEmpty()||clientnameTxt.getText().isEmpty()||issuerbank
Txt.getText().isEmpty()||bankaccountTxt.getText().isEmpty()||balanceamtTxt.getT
ext().isEmpty()||cvcnumberTxt.getText().isEmpty()||interestrateTxt.getText().isEm
pty()){
        JOptionPane.showMessageDialog(frame,"Text field cannot be empty",
"Input Error", JOptionPane.ERROR_MESSAGE);
      }
      else{
        try{
          //getting input values from text fields
          int cardId = Integer.parseInt(cardidTxt_0.getText());
          String clientName = clientnameTxt.getText();
          String issuerBank = issuerbankTxt.getText();
          String bankAccount = bankaccountTxt.getText();
          double BalanceAmount =
Double.parseDouble(balanceamtTxt.getText());
          int CVCNumber = Integer.parseInt(cvcnumberTxt.getText());
          double interestRate = Double.parseDouble(interestrateTxt.getText());
          String days_1 = (String) dayCombo_1 .getSelectedItem();
          String months_1 = (String) monthCombo_1.getSelectedItem();
          String years_1 = (String) yearCombo_1.getSelectedItem();
          String expirationDate = days_1 + " " + months_1 + " " + years_1;
          boolean toAdd = true;
          CreditCard creditCard;

          if(bankCardsArraylists.isEmpty()){

            creditCard = new
CreditCard(BalanceAmount,cardId,bankAccount,issuerBank, clientName,
CVCNumber, interestRate,expirationDate);
            bankCardsArraylists.add(creditCard);
            JOptionPane.showMessageDialog(frame,"Credit Card has been
added Successfully", "Alert", JOptionPane.INFORMATION_MESSAGE);
          }
          else{
            for(BankCard card:bankCardsArraylists){

              if(card instanceof CreditCard){
                creditCard = (CreditCard) card;
```

```
                    if(creditCard.getCardId()==cardId){
                        JOptionPane.showMessageDialog(frame,"Credit Card
cannot be added", "Input Error", JOptionPane.ERROR_MESSAGE);
                        toAdd = false;
                        break;
                    }
                }
                else{
                    JOptionPane.showMessageDialog(frame,"Card  not found",
"Input Error", JOptionPane.ERROR_MESSAGE);
                    toAdd = false;
                    break;
                }
            }

            if(toAdd == true){
                creditCard = new
CreditCard(BalanceAmount,cardId,bankAccount,issuerBank, clientName,
CVCNumber, interestRate,expirationDate);
                bankCardsArraylists.add(creditCard);
                JOptionPane.showMessageDialog(frame,"Credit Card has
been added Successfully", "Alert", JOptionPane.INFORMATION_MESSAGE);
            }

        }

    }

    catch(NumberFormatException ex){
        JOptionPane.showMessageDialog(frame,"Invalid Input", "Input
Error", JOptionPane.ERROR_MESSAGE);
    }
  }
}

//impementing Event Listener of display button for DebitCard
if (e.getSource()== displayBtn_0){

    //retrieving values from text fields
    String days_0 =(String) dayCombo_0.getSelectedItem();
    String months_0 =(String) monthCombo_0.getSelectedItem();
    String years_0 =(String) yearCombo_0.getSelectedItem();
    String dateOfWithdrawal = days_0 + " " + months_0 + " "+ years_0;
```

```
          if (bankCardsArraylists.isEmpty()){
             JOptionPane.showMessageDialog(frame,"Card has not been
added,nothing to display","Error",JOptionPane.ERROR_MESSAGE);
          }
          else{
             for(BankCard card:bankCardsArraylists){

                if(card instanceof DebitCard){
                   DebitCard debitCard = (DebitCard) card;
                   //calling display method from DebitCard class
                   debitCard.display();

                   String popUp_1 = "Card Id: " + cardidTxt_0.getText() + "\nIssuer
Bank:" +issuerbankTxt.getText() + "\nBank Account" +bankaccountTxt.getText()+
"\nBalance Amount" +balanceamtTxt.getText()+ "\nClient
Name:"+clientnameTxt.getText()+ "\nPinNumer:" +pinnumberTxt.getText()+
"\nWithdrawal Amount:" +withdrawalamtTxt.getText()+ "\nDate Of
Withdrawal:"+dateOfWithdrawal;

JOptionPane.showMessageDialog(frame,popUp_1,"Info",JOptionPane.INFORM
ATION_MESSAGE);

                }
                else{
                   JOptionPane.showMessageDialog(frame,"Debit Card not Found",
"Error", JOptionPane.ERROR_MESSAGE);
                }
             }
          }
       }

       //implementing Action Listener of display button for credit card
       if (e.getSource() == displayBtn_1){

          //retrieving values from text fields
          String days_1 =(String) dayCombo_0.getSelectedItem();
          String months_1 =(String) monthCombo_0.getSelectedItem();
          String years_1 =(String) yearCombo_0.getSelectedItem();
          String expirationDate = days_1 + " " + months_1 + " "+ years_1;

          if (bankCardsArraylists.isEmpty()){
            JOptionPane.showMessageDialog(frame,"Card has not been
added,nothing to display","Error",JOptionPane.ERROR_MESSAGE);
        }
        else{
            for(BankCard card:bankCardsArraylists){

                if(card instanceof CreditCard){
                    CreditCard creditCard = (CreditCard) card;
                    //calling display method from DebitCard class
                    creditCard.display();

                    String popUp_2 = ("Client Name:"+clientnameTxt.getText()+
"\nCard Id: " + cardidTxt_2.getText() + "\nIssuer Bank:" +issuerbankTxt.getText()
+ "\nBank Account" +bankaccountTxt.getText()+ "\nBalance Amount"
+balanceamtTxt.getText()+ "\nCVC NO.:" +cvcnumberTxt.getText()+
"\nInterestRate:"+interestrateTxt.getText()+"\nCredit
Limit:"+creditlimitTxt.getText()+ "\nGrace Period:" +graceperiodTxt.getText()+
"\nExpiration Date:" +expirationDate);

JOptionPane.showMessageDialog(frame,popUp_2,"Info",JOptionPane.INFORM
ATION_MESSAGE);

                }
                else{
                    JOptionPane.showMessageDialog(frame,"Debit Card not Found",
"Error", JOptionPane.ERROR_MESSAGE);
                }
            }
        }
    }

    //implementing Event Listener of withdraw button
    if (e.getSource()==withdrawBtn){

if(cardidTxt_1.getText().isEmpty()||withdrawalamtTxt.getText().isEmpty()||pinnum
berTxt.getText().isEmpty()){
            JOptionPane.showMessageDialog(frame,"Text field cannot be empty",
"Input Error", JOptionPane.ERROR_MESSAGE);
        }
        else{
            try{
                //retrieving values from text fields
                int cardId = Integer.parseInt(cardidTxt_1.getText());
```

```
          int withDrawalAmount =
Integer.parseInt(withdrawalamtTxt.getText());
          int pinNumber = Integer.parseInt(pinnumberTxt.getText());
          String days_0 =(String) dayCombo_0.getSelectedItem();
          String months_0 =(String) monthCombo_0.getSelectedItem();
          String years_0 =(String) yearCombo_0.getSelectedItem();
          String dateOfWithdrawal = days_0 + " " + months_0 + " "+ years_0;
          boolean availableCard = false;
          if(bankCardsArraylists.isEmpty()){

            JOptionPane.showMessageDialog(frame,"Card has not been
added","Error",JOptionPane.ERROR_MESSAGE);
          }
          else{ //if debit card is already present:

            for(BankCard card:bankCardsArraylists){
              if(card instanceof DebitCard){
                //downcast
                DebitCard debitCard = (DebitCard) card;
                if(debitCard.getCardId() == cardId &&
debitCard.getPinNumber()==pinNumber){
                  double initialAmount = debitCard.getBalanceAmount();

debitCard.Withdraw(withDrawalAmount,dateOfWithdrawal,pinNumber);
                  double remainingAmount = initialAmount -
withDrawalAmount;
                  if(withDrawalAmount > 0){
                    if( withDrawalAmount<=initialAmount){

                      String popUp_0 ="Amount has been withdrawn
successfully,your remaining balance is:"+ remainingAmount+ "\nCard Id: " +
cardId + "\nWithdrawal Amount: " +withDrawalAmount + "\nDate Of Withdrawal: "
+dateOfWithdrawal + "\nPIN Number:" +pinNumber;

JOptionPane.showMessageDialog(frame,popUp_0,"Info",
JOptionPane.INFORMATION_MESSAGE);
                    }
                    else{
                      JOptionPane.showMessageDialog(frame,"Dear
user,You have insufficient Balance", "Alert", JOptionPane.ERROR_MESSAGE);

                    }
                  }
```

```
                                    else{
                                        JOptionPane.showMessageDialog(frame,"Please input
valid number", "Error", JOptionPane.ERROR_MESSAGE);

                                    }
                                    availableCard = true;
                                    break;

                                }
                            }
                        }
                        if(!availableCard){
                            JOptionPane.showMessageDialog(frame,"Your CardId or Pin
Number is incorrect", "Error", JOptionPane.ERROR_MESSAGE);

                        }
                    }

                }     catch(NumberFormatException ex){
                    JOptionPane.showMessageDialog(frame,"Invalid Input", "Error",
JOptionPane.ERROR_MESSAGE);

                }
            }
        }
        //implementing Event Listener of set Button
        if (e.getSource() == setBtn){

            try{
                //retrieving values form text fields
                int cardId = Integer.parseInt(cardidTxt_2.getText());
                double newCreditLimit  =  Double.parseDouble(creditlimitTxt.getText());
                int newgracePeriod  =  Integer.parseInt(graceperiodTxt.getText());
                boolean availableCard = false;
                if(bankCardsArraylists.isEmpty()){
                    JOptionPane.showMessageDialog(frame,"Card has not been
added","Error",JOptionPane.ERROR_MESSAGE);
                }
                else{ //if debit card is already present:

                    //loop through the array list
                    for(BankCard card:bankCardsArraylists){
```

```
                  // cheking if the object of card is debitCard or not
                  if(card instanceof CreditCard){
                     //downcast
                     CreditCard creditCard = (CreditCard) card;
                     if(creditCard.getCardId() == cardId){
                        availableCard = true;
                        //calling setCreditLimit method from CreditCard Class
                        creditCard.setCreditLimit(newCreditLimit,newgracePeriod);
                        if(newCreditLimit<=2.5*creditCard.getBalanceAmount()){
                           JOptionPane.showMessageDialog(frame,"CreditLimit has
been set:\nnew Credit Limit:"+newCreditLimit+"\nnew Grace
Period:"+newgracePeriod,"Alert", JOptionPane.INFORMATION_MESSAGE);

                        }
                        else{
                           JOptionPane.showMessageDialog(frame,"Credit limit
cannot be issued", "Error", JOptionPane.ERROR_MESSAGE);
                        }
                        //breaking the loop after finding valid card ID
                        break;

                     }
                  }
               }

               if(!availableCard){
                  JOptionPane.showMessageDialog(frame,"Matching card not
found", "Error", JOptionPane.ERROR_MESSAGE);
               }
            }
         }  catch(NumberFormatException ex){
            JOptionPane.showMessageDialog(frame,"Invalid Input", "Error",
JOptionPane.ERROR_MESSAGE);
         }
      }

      if (e.getSource() == cancelBtn){
         try{
            int cardId = Integer.parseInt(cardidTxt_2.getText());
            double newCreditLimit  =  Double.parseDouble(creditlimitTxt.getText());
            int newgracePeriod  =  Integer.parseInt(graceperiodTxt.getText());

            if(bankCardsArraylists.isEmpty()){
                    JOptionPane.showMessageDialog(frame,"Card has not been 
added","Error",JOptionPane.ERROR_MESSAGE);
                }
                else{
                    //loop through the array list

                    for(BankCard card:bankCardsArraylists){
                        //is the card debit card or not?
                        if(card instanceof CreditCard){
                            //downcast
                            CreditCard creditCard = (CreditCard) card;
                            if(creditCard.getCardId() == cardId){

                                //calling cancelCreditcard method from CreditCard Class
                                creditCard.cancelCreditcard();
                                creditlimitTxt.setText("");
                                graceperiodTxt.setText("");
                                cardidTxt_2.setText("");

                                JOptionPane.showMessageDialog(frame,"Credit Card has 
been Cancelled","Alert", JOptionPane.INFORMATION_MESSAGE);
                                break;
                            }
                            else{
                                JOptionPane.showMessageDialog(frame,"Dear user,your 
Card Id is incorrect", "Error", JOptionPane.ERROR_MESSAGE);

                            }
                        }
                        else{
                            JOptionPane.showMessageDialog(frame,"Card not found", 
"Error", JOptionPane.ERROR_MESSAGE);

                        }

                    }
                }
            }

            catch(NumberFormatException ex){
                JOptionPane.showMessageDialog(frame,"Invalid Input", "Error", 
JOptionPane.ERROR_MESSAGE);
            }
        }

        //implementing Action Listener of display button for clearBt
        if(e.getSource() ==clearBtn){
            //Setting every textfield to empty
            balanceamtTxt.setText("");
            bankaccountTxt.setText("");
            issuerbankTxt.setText("");
            clientnameTxt.setText("");
            pinnumberTxt.setText("");
            cardidTxt_0.setText("");
            cardidTxt_1.setText("");
            cardidTxt_2.setText("");
            clientnameTxt.setText("");
            cvcnumberTxt.setText("");
            graceperiodTxt.setText("");
            interestrateTxt.setText("");
            withdrawalamtTxt.setText("");
        }
    }
}
```