CS4001NI Programming

30% Individual Coursework

2023-24 Spring

Student Name: Arabinda Krishna Kansakar

London Met ID: 22068178

College ID: np01cp4a220161

Group: C3

Assignment Due Date: Tuesday, March 21, 2023

Assignment Submission Date: Tuesday, May 9, 2023

Table of Contents

List of Figures

List of Tables

1. Introduction

Introduced by Sun Microsystems in 1996, Java is a versatile programming language and computing platform suitable for various purposes. It is characterized by its platform independence, enabling it to operate across multiple operating systems. Java adopts an object-oriented methodology, emphasizes network-centric functionalities, and can serve as a platform. Renowned for its speed, security, and dependability, Java finds application in the development of diverse software, encompassing mobile applications, enterprise software, big data systems, and server-side technologies. (AWS, 2023).

1.1. Introduction to the project

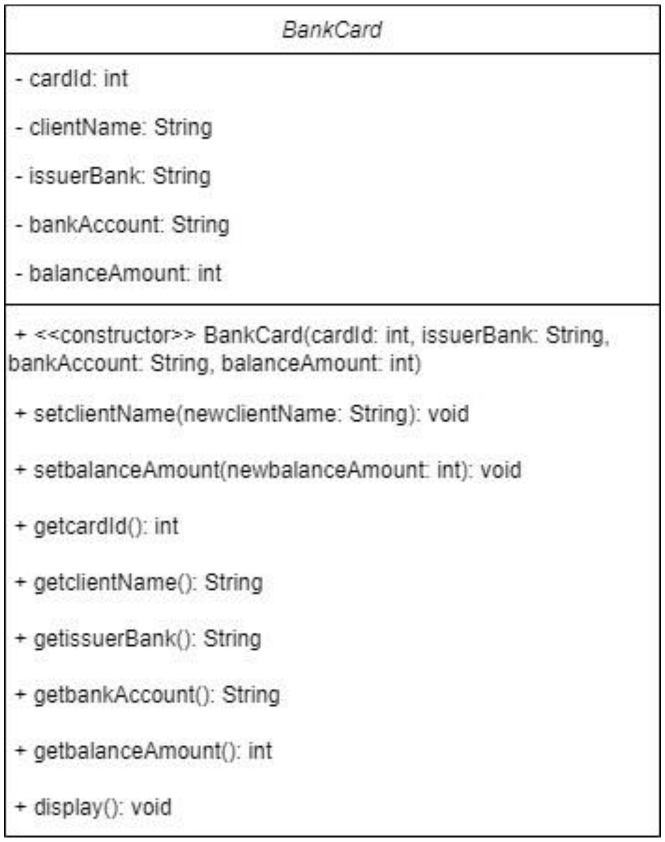
The project's objective is to apply Java's object-oriented programming principles to develop a solution for a real-world problem. The specific focus is on designing and implementing a parent class called "Bankcard" and two subclasses, namely "Debit card" and "Credit card," to represent different types of bank cards. These classes will utilize key object-oriented concepts such as inheritance, polymorphism, encapsulation, and abstraction. Additionally, the project will integrate important GUI elements by leveraging classes from the javax.swing package, functions and features from the java.awt package, event listeners, action events, and visually appealing design principles.

In the project, several tools are used to aid in its completion. These tools include BlueJ, Microsoft Word, and Draw.io. BlueJ is an integrated development environment (IDE) used for writing and compiling Java code. It helps to clearly show the relationships between different classes in the project. Microsoft Word is a popular word processing software developed by Microsoft, used to write the report for this project as it is easy to use and versatile. Draw.io is an open-source technology stack that is used to create diagrams and was used to create the class diagrams needed in the coursework report.

2. Class Diagram

Class diagrams are a type of UML (Unified Modelling Language) diagrams that are used to illustrate the structure of a system by describing the attributes, operations, and relationships between classes. UML refers to them as structure diagrams. They work based on the principles of object-orientation, which outlines how objects interact with one another. Class diagrams are an important tool in the design and development of object-oriented systems, as they provide a clear and detailed view of the system's structure, including the relationships between different classes, their attributes, and the methods they use. This helps in understanding the system's architecture and how the components interact with each other. (microTOOL, 2023).

2.1. BankCard Class



BankCard

- cardId: int
- clientName: String
- issuerBank: String
- bankAccount: String
- balanceAmount: int

+ <<constructor>> BankCard(cardId: int, issuerBank: String, bankAccount: String, balanceAmount: int)
+ setclientName(newclientName: String): void
+ setbalanceAmount(newbalanceAmount: int): void
+ getcardId(): int
+ getclientName(): String
+ getissuerBank(): String
+ getbankAccount(): String
+ getbalanceAmount(): int
+ display(): void

Figure 1 BankCard Class Diagram

2.2. DebitCard Class

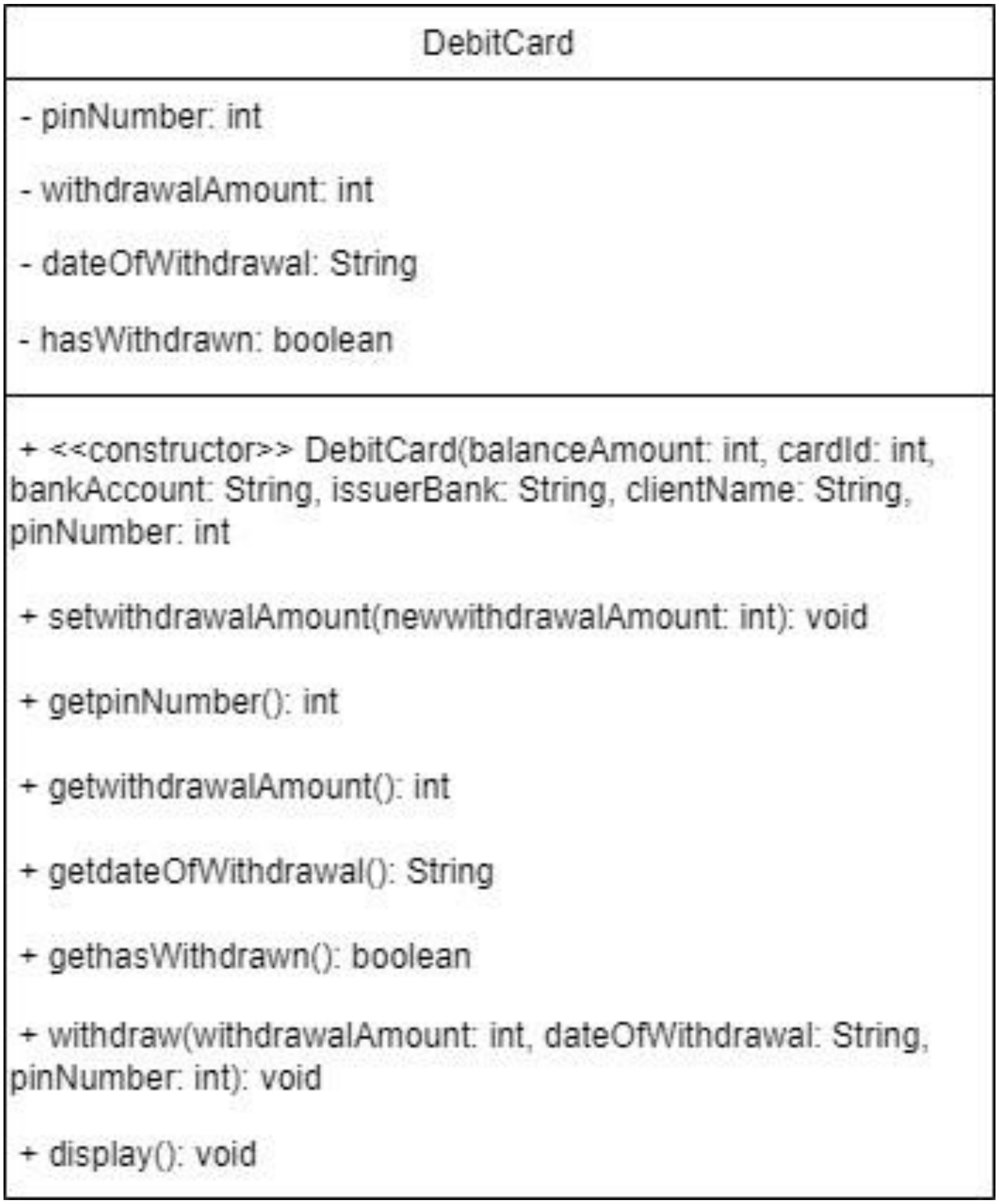

DebitCard
- pinNumber: int
- withdrawalAmount: int
- dateOfWithdrawal: String
- hasWithdrawn: boolean
+ <<constructor>> DebitCard(balanceAmount: int, cardId: int, bankAccount: String, issuerBank: String, clientName: String, pinNumber: int
+ setwithdrawalAmount(newwithdrawalAmount: int): void
+ getpinNumber(): int
+ getwithdrawalAmount(): int
+ getdateOfWithdrawal(): String
+ gethasWithdrawn(): boolean
+ withdraw(withdrawalAmount: int, dateOfWithdrawal: String, pinNumber: int): void
+ display(): void

Figure 2 DebitCard Class Diagram

2.3 CreditCard Class

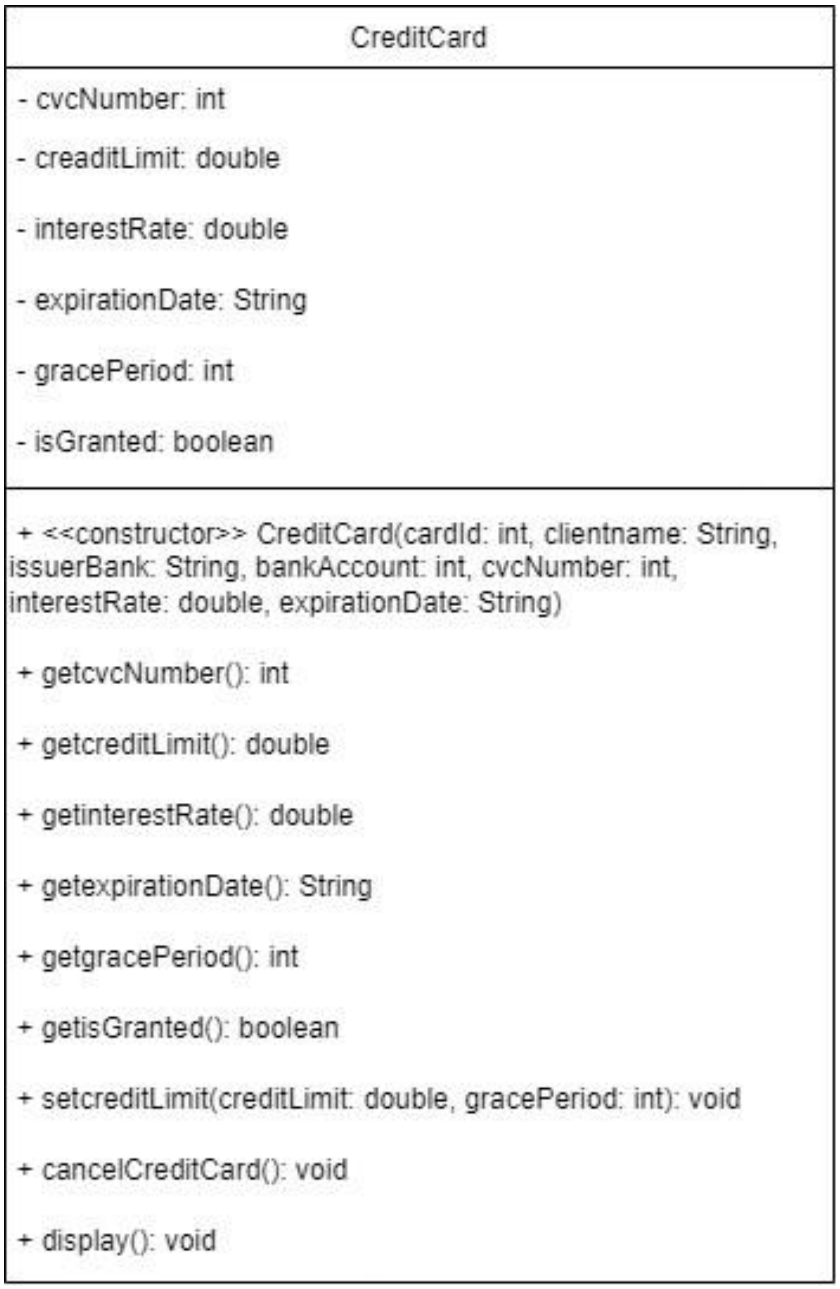
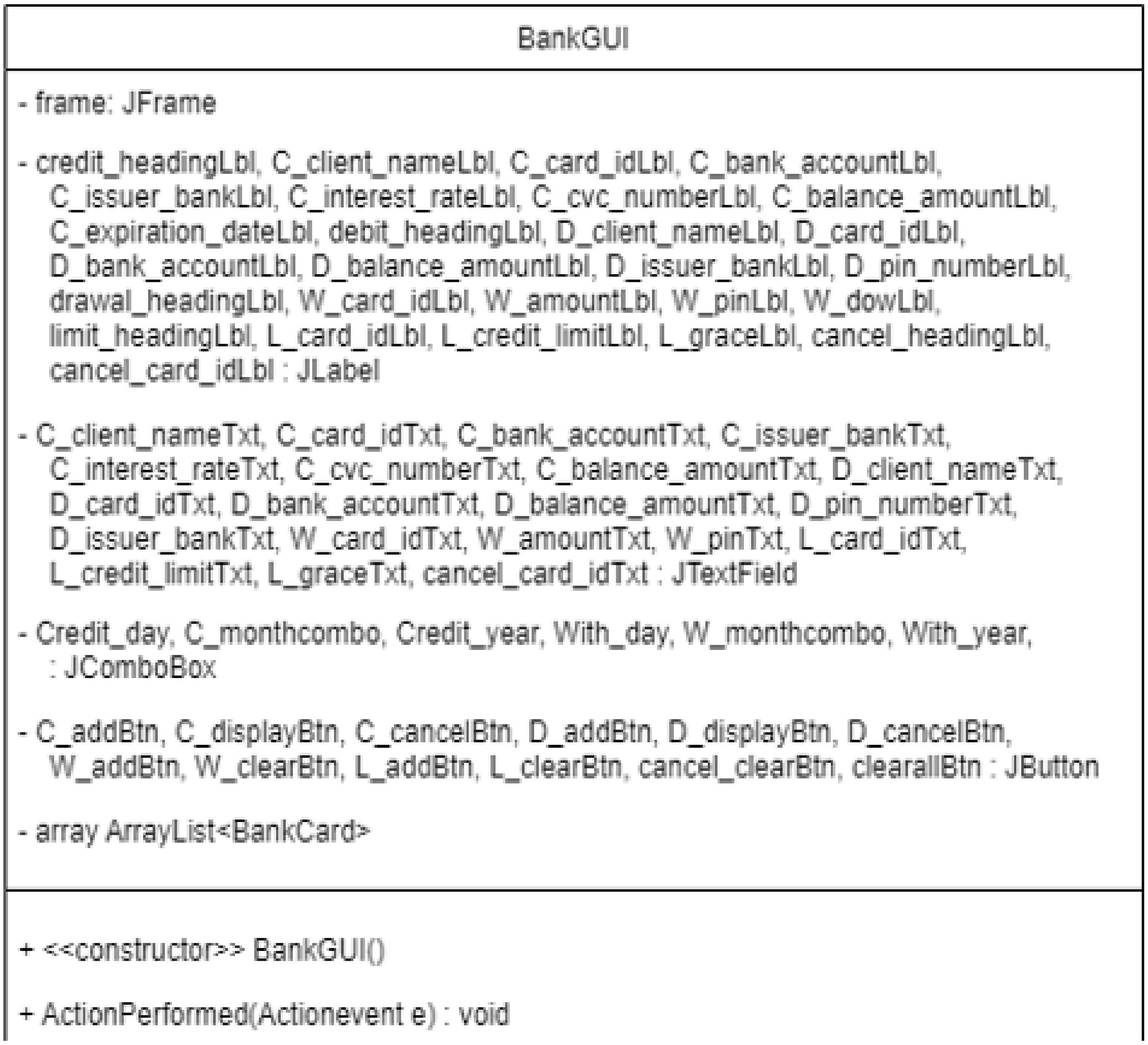

CreditCard
- cvcNumber: int
- creaditLimit: double
- interestRate: double
- expirationDate: String
- gracePeriod: int
- isGranted: boolean
+ <<constructor>> CreditCard(cardId: int, clientname: String, issuerBank: String, bankAccount: int, cvcNumber: int, interestRate: double, expirationDate: String)
+ getcvcNumber(): int
+ getcreditLimit(): double
+ getinterestRate(): double
+ getexpirationDate(): String
+ getgracePeriod(): int
+ getisGranted(): boolean
+ setcreditLimit(creditLimit: double, gracePeriod: int): void
+ cancelCreditCard(): void
+ display(): void

Figure 3 CreditCard Class Diagram

2.4 BankGUI Class



BankGUI

- frame: JFrame

- credit_headingLbl, C_client_nameLbl, C_card_idLbl, C_bank_accountLbl, C_issuer_bankLbl, C_interest_rateLbl, C_cvc_numberLbl, C_balance_amountLbl, C_expiration_dateLbl, debit_headingLbl, D_client_nameLbl, D_card_idLbl, D_bank_accountLbl, D_balance_amountLbl, D_issuer_bankLbl, D_pin_numberLbl, drawal_headingLbl, W_card_idLbl, W_amountLbl, W_pinLbl, W_dowLbl, limit_headingLbl, L_card_idLbl, L_credit_limitLbl, L_graceLbl, cancel_headingLbl, cancel_card_idLbl : JLabel

- C_client_nameTxt, C_card_idTxt, C_bank_accountTxt, C_issuer_bankTxt, C_interest_rateTxt, C_cvc_numberTxt, C_balance_amountTxt, D_client_nameTxt, D_card_idTxt, D_bank_accountTxt, D_balance_amountTxt, D_pin_numberTxt, D_issuer_bankTxt, W_card_idTxt, W_amountTxt, W_pinTxt, L_card_idTxt, L_credit_limitTxt, L_graceTxt, cancel_card_idTxt : JTextField

- Credit_day, C_monthcombo, Credit_year, With_day, W_monthcombo, With_year, : JComboBox

- C_addBtn, C_displayBtn, C_cancelBtn, D_addBtn, D_displayBtn, D_cancelBtn, W_addBtn, W_clearBtn, L_addBtn, L_clearBtn, cancel_clearBtn, clearallBtn : JButton

- array ArrayList<BankCard>

+ <<constructor>> BankGUI()

+ ActionPerformed(Actionevent e) : void

Figure 4 BankGUI Class diagram

2.5. Combined Class Diagram

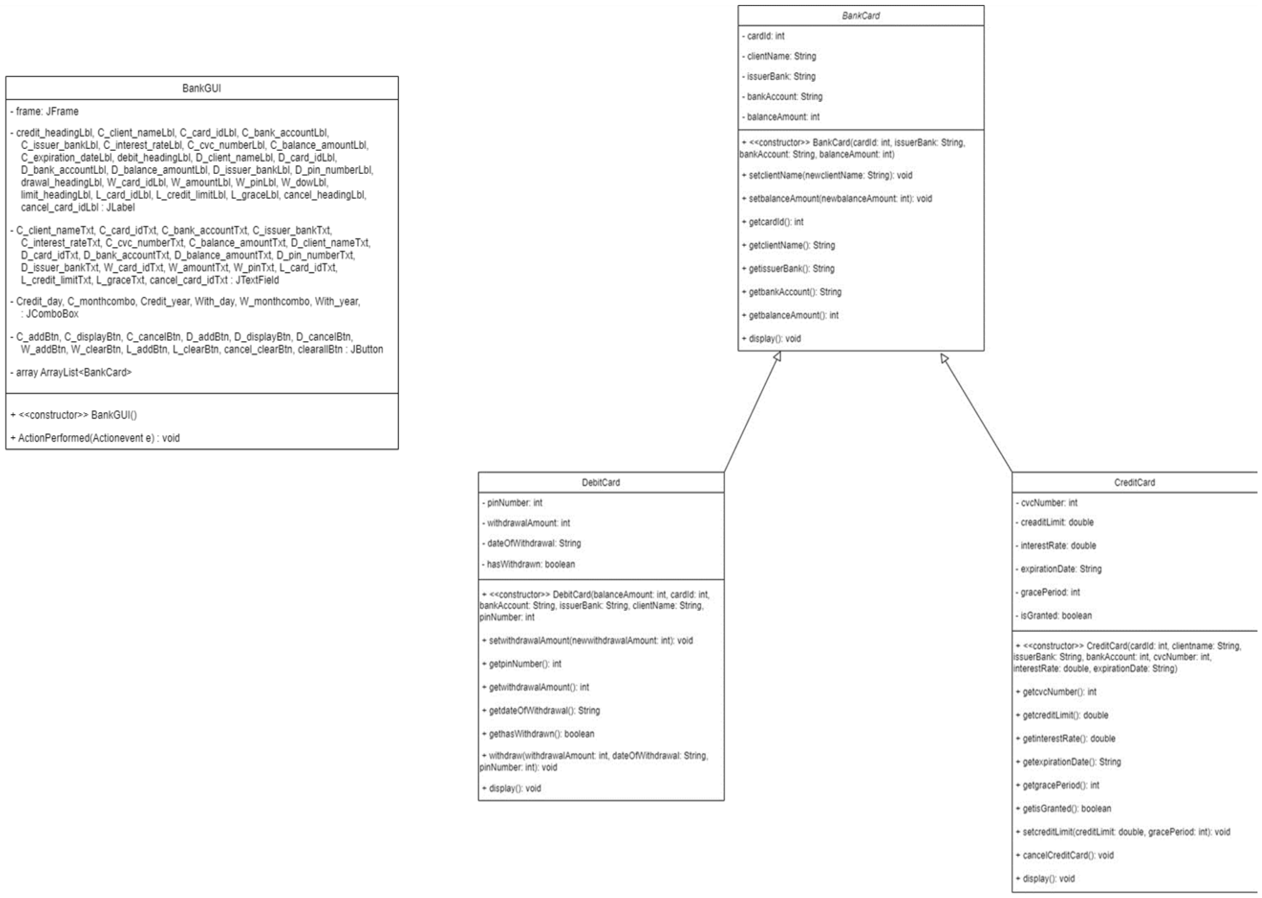

BankGUI
- frame: JFrame
- credit_headingLbl, C_client_nameLbl, C_card_idLbl, C_bank_accountLbl, C_issuer_bankLbl, C_interest_rateLbl, C_cvc_numberLbl, C_balance_amountLbl, C_expiration_dateLbl, debit_headingLbl, D_client_nameLbl, D_card_idLbl, D_bank_accountLbl, D_balance_amountLbl, D_issuer_bankLbl, D_pin_numberLbl, drawal_headingLbl, W_card_idLbl, W_amountLbl, W_pinLbl, W_dowLbl, limit_headingLbl, L_card_idLbl, L_credit_limitLbl, L_graceLbl, cancel_headingLbl, cancel_card_idLbl : JLabel
- C_client_nameTxt, C_card_idTxt, C_bank_accountTxt, C_issuer_bankTxt, C_interest_rateTxt, C_cvc_numberTxt, C_balance_amountTxt, D_client_nameTxt, D_card_idTxt, D_bank_accountTxt, D_balance_amountTxt, D_pin_numberTxt, D_issuer_bankTxt, W_card_idTxt, W_amountTxt, W_pinTxt, L_card_idTxt, L_credit_limitTxt, L_graceTxt, cancel_card_idTxt : JTextField
- Credit_day, C_monthcombo, Credit_year, With_day, W_monthcombo, With_year, : JComboBox
- C_addBtn, C_displayBtn, C_cancelBtn, D_addBtn, D_displayBtn, D_cancelBtn, W_addBtn, W_clearBtn, L_addBtn, L_clearBtn, cancel_clearBtn, clearallBtn : JButton
- array ArrayList<BankCard>
+ <<constructor>> BankGUI()
+ ActionPerformed(Actionevent e) : void
BankCard
- cardId: int
- clientName: String
- issuerBank: String
- bankAccount: String
- balanceAmount: int
+ <<constructor>> BankCard(cardId: int, issuerBank: String, bankAccount: String, balanceAmount: int)
+ setclientName(newclientName: String): void
+ setbalanceAmount(newbalanceAmount: int): void
+ getcardId(): int
+ getclientName(): String
+ getissuerBank(): String
+ getbankAccount(): String
+ getbalanceAmount(): int
+ display(): void
DebitCard
- pinNumber: int
- withdrawalAmount: int
- dateOfWithdrawal: String
- hasWithdrawn: boolean
+ <<constructor>> DebitCard(balanceAmount: int, cardId: int, bankAccount: String, issuerBank: String, clientName: String, pinNumber: int
+ setwithdrawalAmount(newwithdrawalAmount: int): void
+ getpinNumber(): int
+ getwithdrawalAmount(): int
+ getdateOfWithdrawal(): String
+ gethasWithdrawn(): boolean
+ withdraw(withdrawalAmount: int, dateOfWithdrawal: String, pinNumber: int): void
+ display(): void
CreditCard
- cvcNumber: int
- creaditLimit: double
- interestRate: double
- expirationDate: String
- gracePeriod: int
- isGranted: boolean
+ <<constructor>> CreditCard(cardId: int, clientname: String, issuerBank: String, bankAccount: int, cvcNumber: int, interestRate: double, expirationDate: String)
+ getcvcNumber(): int
+ getcreditLimit(): double
+ getinterestRate(): double
+ getexpirationDate(): String
+ getgracePeriod(): int
+ getisGranted(): boolean
+ setcreditLimit(creditLimit: double, gracePeriod: int): void
+ cancelCreditCard(): void
+ display(): void

Figure 5 Combined Class Diagram

3. Pseudocode

Pseudocode is an approach to algorithmic description that combines natural language with programming constructs. It is not an actual programming code, but rather a means of expressing the logical flow of an algorithm in a manner that is comprehensible to both humans and computers. Pseudocode is commonly employed in the planning and design stages of programming, serving as a blueprint before the actual coding commences. It can be written in any programming language or in a language-independent format. The primary objective of pseudocode is to provide clarity, conciseness, and ease of understanding, rendering it a valuable tool for programmers of all proficiency levels. (theprogrammedwords, 1957).

3.1. Pseudocode for BankGUI

```
IMPORT javax.swing.*

IMPORT java.awt.event.*

IMPORT java.awt.Color

IMPORT java.awt.Font

IMPORT java.util.ArrayList

IMPORT java.awt.Graphics

CREATE a class BankGUI which implements ActionListener

DO

    DECLARE instance variable frame as JFrame using private Access modifier
```

DECLARE instance variable colorPanel as JPanel using private modifier

DECLARE instance variable credit_headingLbl, C_client_nameLbl, C_card_idLbl, C_bank_accountLbl, C_issuer_bankLbl, C_interest_rateLbl, C_cvc_numberLbl, C_balance_amountLbl, C_expiration_dateLbl, debit_headingLbl, D_client_nameLbl, D_card_idLbl, D_bank_accountLbl, D_balance_amountLbl, D_issuer_bankLbl, D_pin_numberLbl, drawal_headingLbl, W_card_idLbl, W_amountLbl, W_pinLbl, W_dowLbl, limit_headingLbl, L_card_idLbl, L_credit_limitLbl, L_graceLbl, cancel_headingLbl, cancel_card_idLbl as JLabel using private modifier

DECLARE instance variable C_client_nameTxt, C_card_idTxt, C_bank_accountTxt, C_issuer_bankTxt, C_interest_rateTxt, C_cvc_numberTxt, C_balance_amountTxt, D_client_nameTxt, D_card_idTxt, D_bank_accountTxt, D_balance_amountTxt, D_pin_numberTxt, D_issuer_bankTxt, W_card_idTxt, W_amountTxt, W_pinTxt, L_card_idTxt, L_credit_limitTxt, L_graceTxt, cancel_card_idTxt as JTextField using private modifier

DECLARE instance variable Credit_day, C_monthcombo, Credit_year, With_day, W_monthcombo, With_year as JComboBox using private modifier

DECLARE instance variable C_addBtn, C_displayBtn, C_cancelBtn, D_addBtn, D_displayBtn, D_cancelBtn, W_addBtn, W_clearBtn, L_addBtn, L_clearBtn, cancel_clearBtn as JButton using private modifier

DECLARE instance variable gui_Card as ArrayList<BankCard>

CREATE a method named actionPerformed with public modifier, data type as

Void and pass the parameter ActionEvent

DO

```
IF the source of the event is C_addBtn THEN
    TRY to execute the following code block
        SET C_add_client_name to the text value of C_client_nameTxt
        SET C_add_card_id to the text value of C_card_idTxt
        SET C_add_bank_account to the text value of C_bank_accountTxt
        SET C_add_issuer_bank to the text value of C_issuer_bankTxt
        SET C_add_interest_rate to the text value of C_interest_rateTxt
        SET C_add_cvc_number to the text value of C_cvc_numberTxt
        SET C_add_balance_amount to the text value of C_balance_amountTxt
        SET day to the selected value of Credit_day
        SET month to the selected value of C_monthcombo
        SET year to the selected value of Credit_year
        SET fulldate to the concatenation of day, month, and year
        CONVERT C_add_card_id to an integer and assign it to card_Id
        CONVERT C_add_balance_amount to an integer and assign it to
balance_Amount
        CONVERT C_add_cvc_number to an integer and assign it to cvc_Number
        CONVERT C_add_interest_rate to a double and assign it to interest_Rate

        SET isadd to false

        FOR EACH BankCard object "bank_card" in gui_Card

          DO

            IF the card ID of "bank_card" is equal to card_Id THEN

              DISPLAY a message "You Already Added Credit Card!"

              SET isadd to true

              BREAK out of the loop

            END IF

          END DO

        END FOR EACH

        DO

          IF isadd is false THEN

            CREATE a new CreditCard object named "creditobject" with card_Id,
C_add_client_name, C_add_issuer_bank, C_add_bank_account, balance_Amount,
cvc_Number, interest_Rate, and fulldate as arguments ADD "creditobject" to the
"gui_Card" list

            DISPLAY a message "Card has been added."
```

```
            END IF

          END DO

        CATCH any NumberFormatException and execute the following code block

          DISPLAY an error message "No card has been added yet!"

        END TRY

      END IF

    END DO



    DO

      IF the source of the event is C_cancelBtn THEN

        SET the text of C_client_nameTxt to an empty string

        SET the text of C_card_idTxt to an empty string

        SET the text of C_bank_accountTxt to an empty string

        SET the text of C_issuer_bankTxt to an empty string

        SET the text of C_interest_rateTxt to an empty string

        SET the text of C_cvc_numberTxt to an empty string

        SET the text of C_balance_amountTxt to an empty string
        SET the selected index of Credit_day to 0
        SET the selected index of C_monthcombo to 0
        SET selected index of Credit_year to 0
    END IF
END DO


DO
    IF the source of the event is C_displayBtn THEN
        FOR EACH BankCard object "bank_card" in gui_Card
            DO
                IF "bank_card" is an instance of CreditCard
                    CALL the display() method on "bank_card"
                END IF
            END DO
        END FOR EACH
    END IF
END DO
```

```
DO
    IF the source of the event is D_addBtn THEN
        TRY to execute the following code block:
            SET D_add_client_name to the text value of D_client_nameTxt
            SET D_add_card_id to the text value of D_card_idTxt
            SET D_add_bank_account to the text value of D_bank_accountTxt
            SET D_add_balance_amount to the text value of D_balance_amountTxt
            SET D_add_pin_number to the text value of D_pin_numberTxt
            SET D_add_issuer_bank to the text value of D_issuer_bankTxt
            CONVERT D_add_card_id to an integer and assign it to card_Id
            CONVERT D_add_balance_amount to an integer and assign it to
balance_Amount
            CONVERT D_add_pin_number to an integer and assign it to pin_Number
            SET isadd to false
            FOR EACH BankCard object "bank_card" in gui_Card
                DO
            IF "bank_card" is an instance of DebitCard THEN
               CAST "bank_card" to a DebitCard object named "debitobject"
                  DO
                     IF the card ID of "debitobject" is equal to card_Id
                        DISPLAY a message "You Already Added A Card With This Info!"
                        SET isadd to true
                        BREAK out of the loop
                     ELSE
                        SET isadd to false
                     END IF
                  END DO
            END IF
         END DO
      END FOR EACH
      DO
         IF isadd is false THEN
```

```
                CREATE a new DebitCard object named "debitobject" with balance_Amount, card_Id, D_add_bank_account, D_add_issuer_bank, D_add_client_name, and pin_Number as arguments

                ADD "debitobject" to the "gui_Card" list

                DISPLAY a message "Card has been added."

              END IF

            END DO

          CATCH any NumberFormatException and execute the following code block

            DISPLAY an error message "No card has been added yet!"

          END TRY

        END IF

      END DO



      DO

        IF the source of the event is D_displayBtn THEN

          FOR EACH BankCard object "bank_card" in gui_Card

            DO

              IF "bank_card" is an instance of DebitCard
```

```
                **CALL** the display() method on "bank_card"

            END IF

        END DO

    END FOR EACH

  END IF

END DO



DO

  IF the source of the event is D_cancelBtn THEN

    SET the text of D_client_nameTxt to an empty string

    SET the text of D_card_idTxt to an empty string

    SET the text of D_bank_accountTxt to an empty string

    SET the text of D_balance_amountTxt to an empty string

    SET the text of D_pin_numberTxt to an empty string

    SET the text of D_issuer_bankTxt to an empty string

  END IF

END DO
```

```
DO
    IF the source of the event is W_addBtn THEN
        TRY
            SET the value from W_card_idTxt and assign it to W_add_card_id
            SET the value from W_amountTxt and assign it to W_add_amount
            SET the value from W_pinTxt and assign it to W_add_pin
            SET the selected value from With_day and assign it to day
            SET the selected value from W_monthcombo and assign it to month
            SET the selected value from With_year and assign it to year
            SET day, month, and year and assign it to fulldate
            CONVERT W_add_card_id as an integer and assign it to card_Id
            CONVERT W_add_amount as an integer and assign it to add_Amount
            CONVERT W_add_pin as an integer and assign it to pin_Number
            SET the value from D_client_nameTxt and assign it to D_add_client_name
            SET the value from D_card_idTxt and assign it to D_add_card_id
```

```
          SET the value from D_bank_accountTxt and assign it to
D_add_bank_account

          SET the value from D_balance_amountTxt and assign it to
D_add_balance_amount

          SET the value from D_pin_numberTxt and assign it to D_add_pin_number

          SET the value from D_issuer_bankTxt and assign it to D_add_issuer_bank

          CONVERT D_add_card_id as an integer and assign it to D_card_Id

          CONVERT D_add_balance_amount as an integer and assign it to
D_balance_Amount

          CONVERT D_add_pin_number as an integer and assign it to
D_pin_Number

          SET isadd to false

          FOR EACH BankCard object "bank_card" in gui_Card

            DO

              IF "bank_card" is an instance of DebitCard

              CAST debitobject to "bank_card" casted as DebitCard

                DO

                  IF debitobject's cardId is equal to card_Id THEN
```

```
                    CALL the withdraw method on debitobject passing add_Amount, fulldate, and pin_Number
                    SHOW a message dialog with the message "Withdraw Successful!"
                    SET isadd to true
                    BREAK the loop
                  END IF
                END DO
              ELSE
                SET isadd to false
              END IF
            END DO
          END FOR EACH
        CATCH NumberFormatException ex
          SHOW a message dialog with the message "No card has been added yet!" and the title "Error"
        END TRY
      END IF
```

```
END DO


DO
   IF the source of the event is W_clearBtn
      SET the text of W_card_idTxt to an empty string
      SET the text of W_amountTxt to an empty string
      SET the text of W_pinTxt to an empty string
      SET the selected index of With_day to 0
      SET the selected index of W_monthcombo to 0
      SET the selected index of With_year to 0
   END IF
END DO


DO
   IF the source of the event is L_addBtn THEN
      TRY
         SET the text from L_card_idTxt and assign it to L_add_card_id
         SET the text from L_credit_limitTxt and assign it to L_add_credit_limit

         SET the text from L_graceTxt and assign it to L_add_grace

         CONVERT L_add_card_id to an integer and assign it to card_Id

         CONVERT L_add_credit_limit to a double and assign it to credit_limit

         CONVERT L_add_grace to an integer and assign it to add_grace

         SET isadd to false

         FOR EACH bank_card in gui_Card

           DO

             IF bank_card is an instance of CreditCard

               SET limit to bank_card as a CreditCard

                 DO

                   IF limit's cardId is equal to card_Id THEN

                     CALL limit's setcreditLimit method with credit_limit and add_grace as arguments

                       DO

                         IF limit's isGranted is true

                           SHOW a message dialog with "Credit Limit Added Successfully!"
```

```
                              SET isadd to true
                              BREAK the loop
                            ELSE
                              SET isadd to false
                            END IF
                          END DO
                        END IF
                      END DO
                    END IF
                  END DO
                END FOR
          CATCH NumberFormatException
            SHOW a message dialog with "No card has been added yet!" and an error icon
          END TRY
        END IF
      END DO
```

```
DO
    IF the source of the event is L_clearBtn
        SET the text of L_card_idTxt to an empty string
        SET the text of L_credit_limitTxt to an empty string
        SET the text of L_graceTxt to an empty string
    END IF
END DO


DO
    IF the source of the event is cancel_clearBtn
        TRY
            SET the value from the cancel_card_idTxt field and assign it to
cancel_add_cancelbtn
            CONVERT the value of cancel_add_cancelbtn to an integer and assign it to
cancel_credit
            SET iscancel to false
            FOR each bank_card in gui_Card
                DO
        IF bank_card is an instance of CreditCard
          Cast bank_card to a CreditCard object and assign it to limit
            DO
              IF the cardId of limit is equal to cancel_credit
                CALL the cancelCreditCard method of limit
                SHOW a message saying "Card Canceled!"
                SET iscancel to true
              END IF
            END DO
        ELSE
          SET iscancel to false
        END IF
      END DO
    END FOR
    DO
      IF iscancel is false
        SHOW a message saying "Card Id has not been added yet"
```

```
                END IF

              END DO

            CATCH NumberFormatException

              SHOW a message saying "Please Enter Valid Card Id" with an error message dialog

            END TRY

          END IF

        END DO



FUNCTION main

    CREATE a new instance of BankGUI and assign it to obj

END FUNCTION
```

4. Method Description

In Java, methods are segments of code designed to perform specific tasks, with the option of returning a result. They offer the advantage of reusability, eliminating the need to rewrite the code. Unlike some programming languages, Java mandates that methods must be encapsulated within a class. (GeeksforGeeks, 2022).

4.1 BankGUI Class Method

Methods	Description
BankGUI()	The BankGUI() class includes a method called "main" that serves as the entry point for executing the class. This method has no parameters and does not return any value (void). It is invoked at the beginning of the BankGUI() class's execution.
actionPerformed(ActionEvent e)	The ActionListener interface is used to receive notifications when a button is clicked. These notifications are provided as ActionEvent objects. The actionPerformed() method is automatically triggered whenever you interact with the registered component, such as clicking on a button.
Add Credit Card Button	The "Add Credit Card" button incorporates the functionality to capture and store the credit card details provided by the user.
Add Debit Card Button	The "Add Debit Card" button includes the functionality to collect and save the debit card information entered by the user.
Withdraw Button	The "Withdraw" button allows the user to make a withdrawal from their debit card account if the given information is correct.
Set Credit Limit Button	The "Credit Limit" button allows users to set the credit limit and grace period for a credit card.
Cancel Credit Card	The "Cancel Credit Card" button enables users to cancel a credit card by providing the correct card ID.

Credit Card Clear Button	The "Cancel Credit Card" button is responsible for clearing all the details entered by the user for a credit card.
Debit Card Clear Button	The "Cancel Debit Card" button is responsible for clearing all the details entered by the user for a debit card.
Withdrawal Clear Button	The "Cancel Withdrawal" button is used to clear all the details entered by the user for a withdrawal transaction.
Credit Limit Clear Button	The "Cancel Credit Limit" button clears all the credit limit details entered by the user.
Credit Card Display Button	The "Display Credit Card" button is used to retrieve and display all the details entered by the user during the credit card creation process.
Debit Card Display Button	The "Display Debit Card" button is used to retrieve and display all the details entered by the user during the debit card creation process.

Table 1 BankGUI Method Description

5. Testing(Inspection)

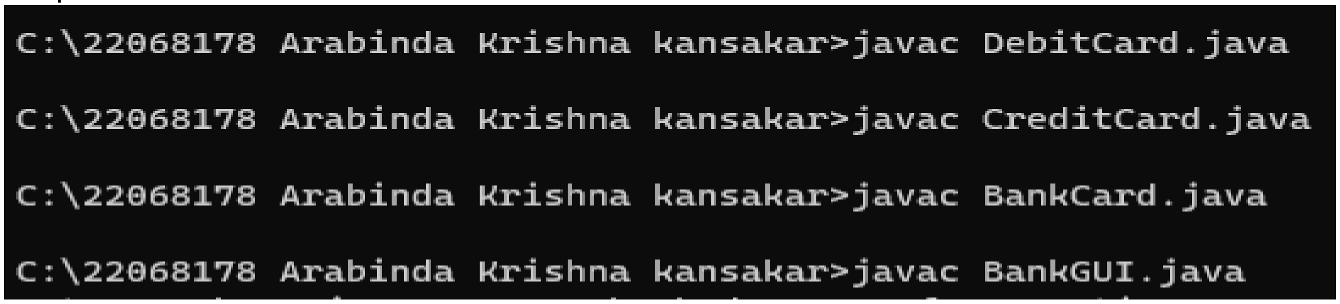
Testing is a crucial process aimed at assessing the performance and functionality of a software application. Its primary purpose is to detect and rectify any errors or bugs present in the software before it is made available to users. By conducting thorough testing, software developers can ensure that the application functions as intended, leading to enhanced performance, usability, and overall user experience. (GeeksforGeeks, 2022).

5.1. Test 1 - Compile and Running using command prompt

Test No:	1
Objective:	To compile and Run the program using command prompt
Action:	- The DebitCard Class is compiled using command prompt - The CreditCard Class is compiled using command prompt - The BankCard Class is compiled using command prompt - The BankGUI Class is compiled using command prompt - Inspect BankGUI
Expected Result:	The DebitCard class, CreditCard class, BankCard class and BankGUI class should compiled and then BankGUI should run
Actual Result:	The classes were compiled and run
Conclusion:	The Test is Successfull

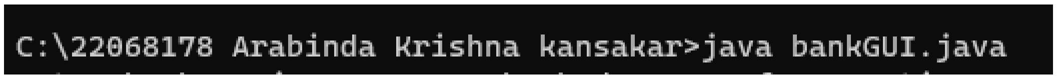
Table 2 To Compile and Run the program using command prompt

Output Result:



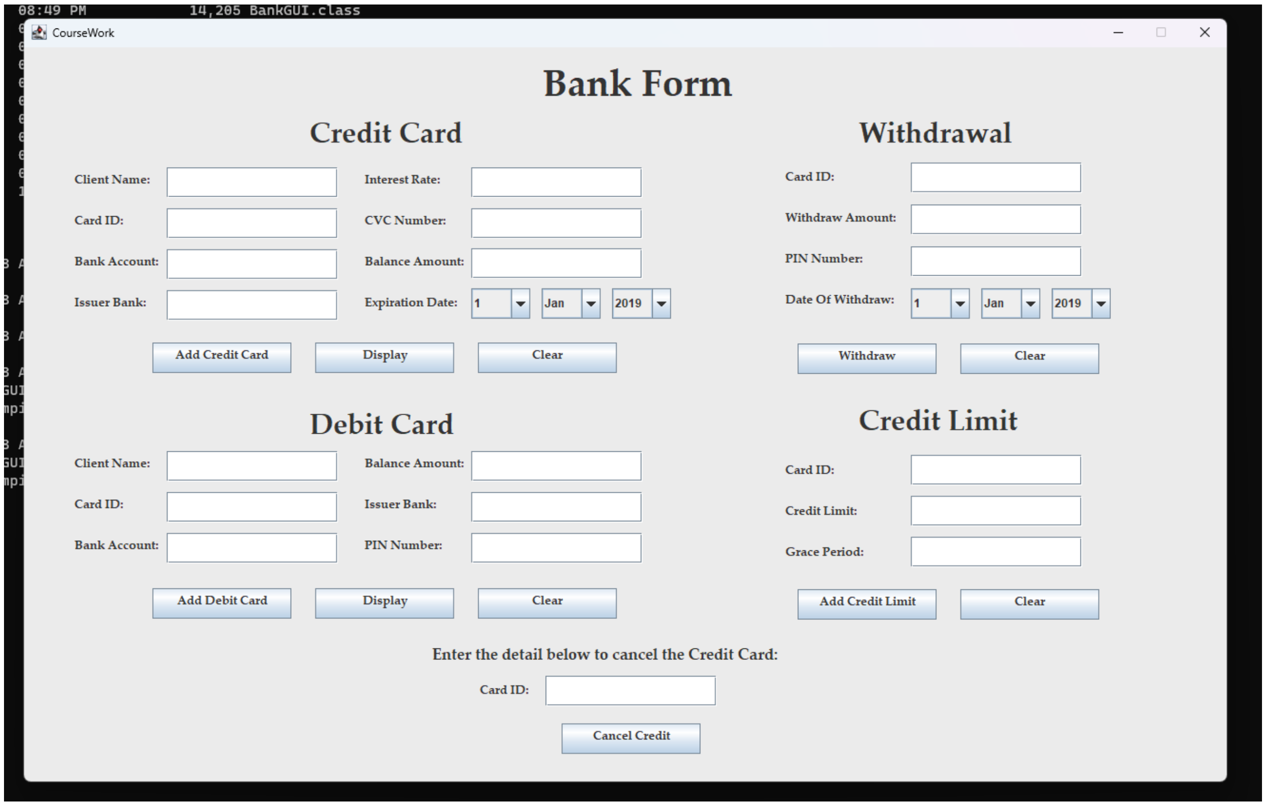
```
C:\22068178 Arabinda Krishna kansakar>javac DebitCard.java

C:\22068178 Arabinda Krishna kansakar>javac CreditCard.java

C:\22068178 Arabinda Krishna kansakar>javac BankCard.java

C:\22068178 Arabinda Krishna kansakar>javac BankGUI.java
```

Figure 6 Screenshot of classes being compiled



```
C:\22068178 Arabinda Krishna kansakar>java bankGUI.java
```

Figure 7 Screenshot of BankGuil being run



```
08:49 PM          14,205 BankGUI.class
```

Figure 8 Screenshot of BankGUI running

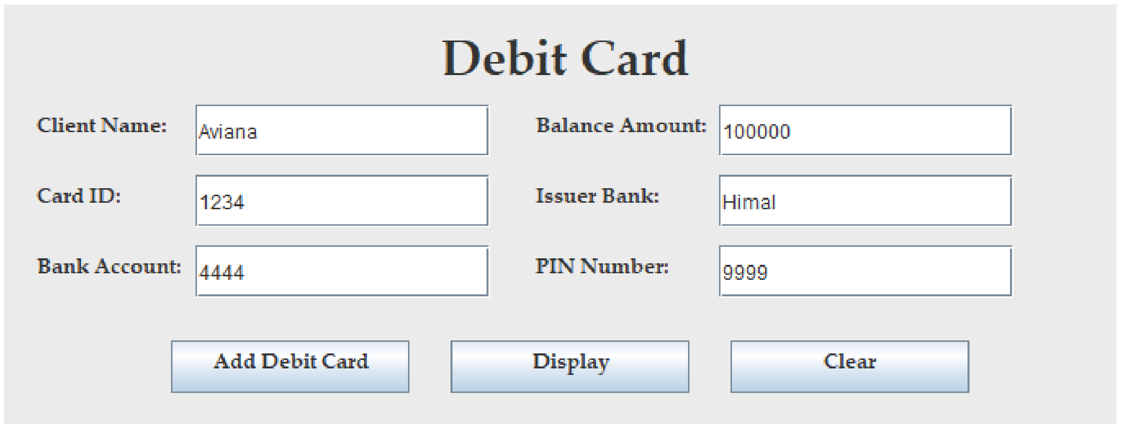
5.2 Test 2 – Evidence for the button functionalities

Test No.	2
Objective:	To show the function of Buttons

Action:	- DebitCard Add Button is inspected - Data is added If the user enters the value - Shows message in both case where the data is added or data is missing - CreditCard Add Button is inspected - Data is added If the user enters the value - Shows message in both case where the data is added or data is missing - Withdrawal Button is inspected - Withdraws the amount from debitcard - Shows message in both case where the data is added or data is missing - Set Credit Limit Button is inspected - Sets Credit Limit for Credit Card - Shows message in both case where the data is added or data is missing - Cancel CreditCard Button is inspected - Cancels the Credit Card
Expected Result:	The DebitCard Add Button, CreditCard Add Button, Withdrawal Button, Set CreditLimit Button and Cancel CreditCard Button should function
Actual Result:	The buttons functioned as expected
Conclusion:	The test is successful.

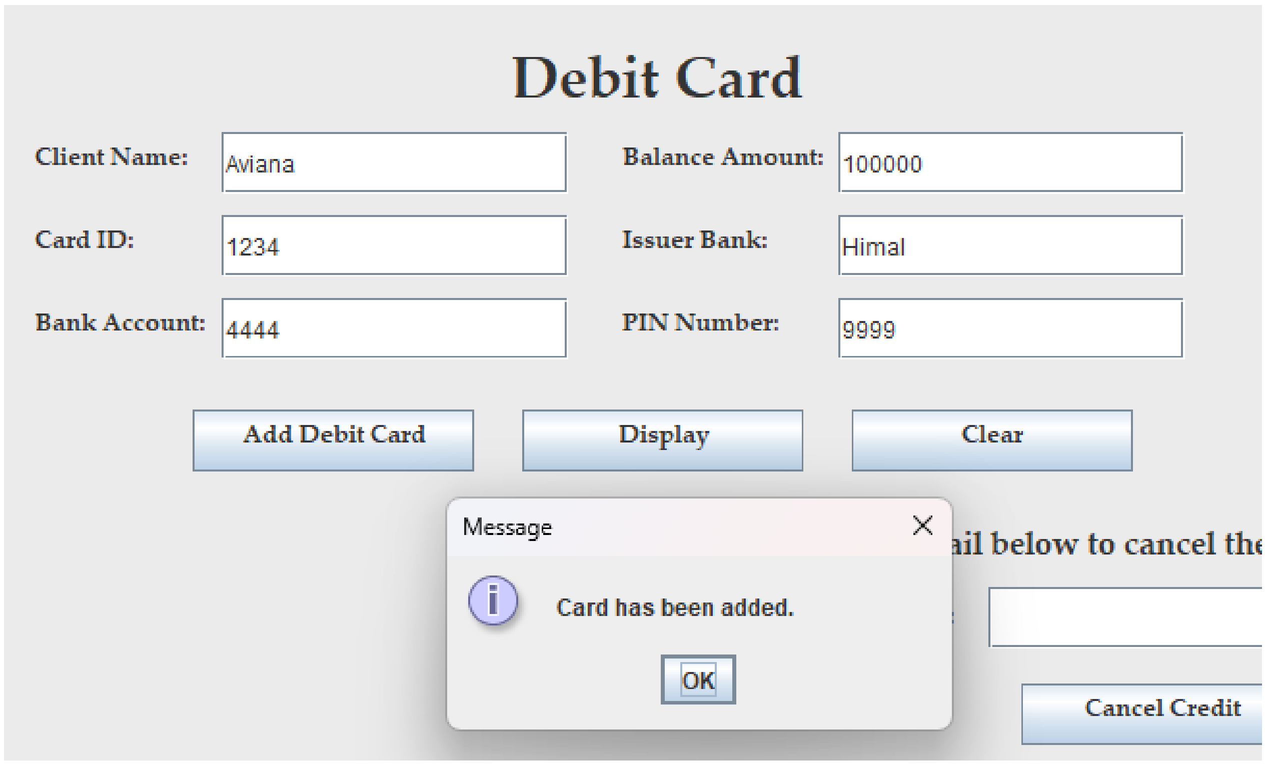
Table 3 To show evidence for the button functionalities

Output Result:



Client Name: Aviana
Balance Amount: 100000
Card ID: 1234
Issuer Bank: Himal
Bank Account: 4444
PIN Number: 9999

Figure 9 Screenshot of data insertion in DebitCard Add Button

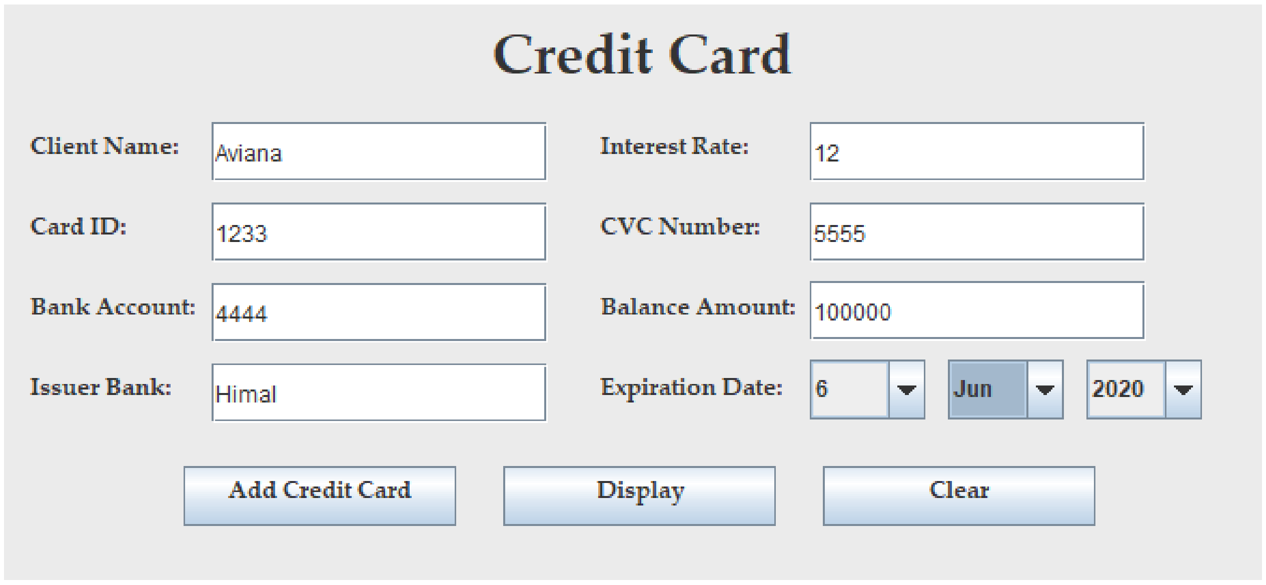


Debit Card

Client Name:	Aviana	Balance Amount:	100000
Card ID:	1234	Issuer Bank:	Himal
Bank Account:	4444	PIN Number:	9999

Card has been added.

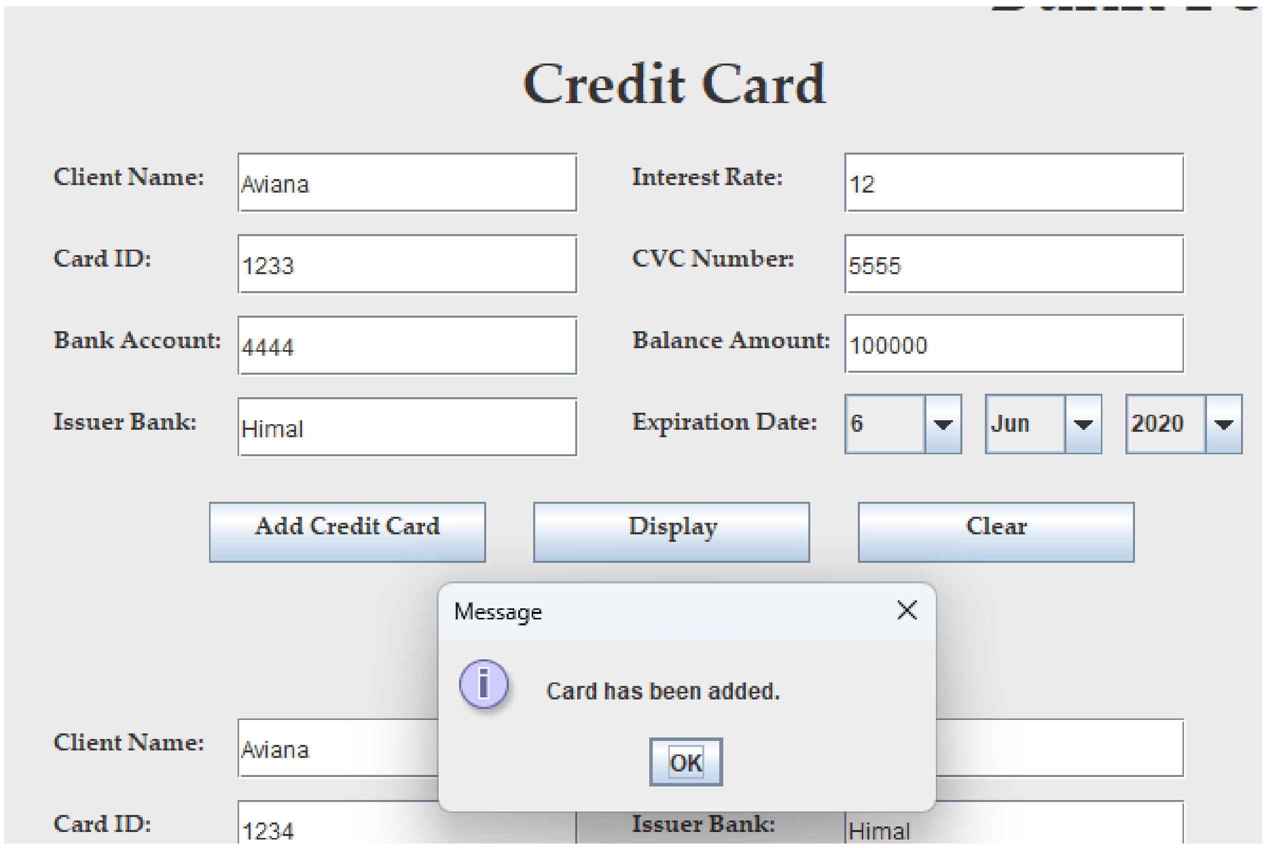
Figure 10:Screenshot of message dialogue



Credit Card

Client Name:	Aviana	Interest Rate:	12
Card ID:	1233	CVC Number:	5555
Bank Account:	4444	Balance Amount:	100000
Issuer Bank:	Himal	Expiration Date:	6 Jun 2020

Figure 11 Screenshot of data insertion in CreditCard Add Button



Credit Card

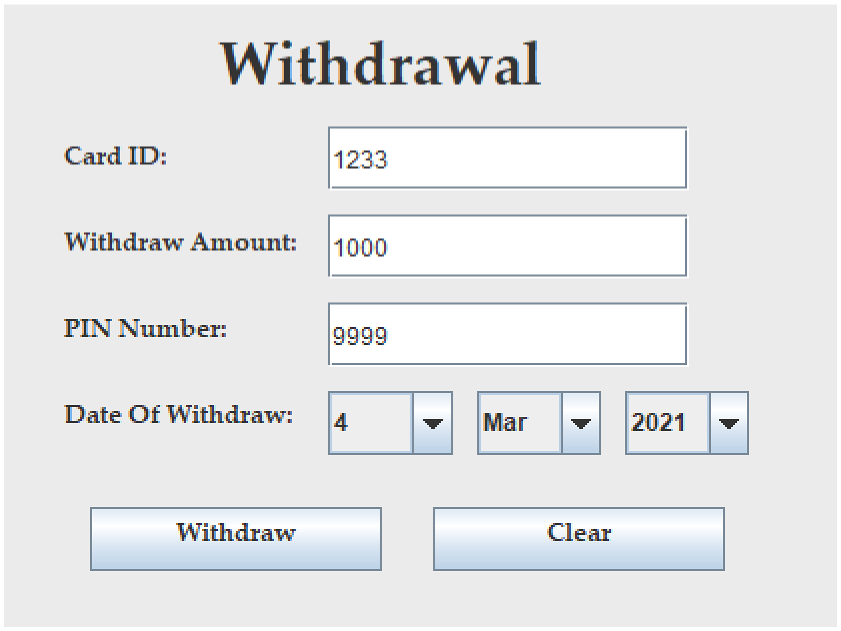
Client Name:	Aviana	Interest Rate:	12
Card ID:	1233	CVC Number:	5555
Bank Account:	4444	Balance Amount:	100000
Issuer Bank:	Himal	Expiration Date:	6 Jun 2020

Message

Card has been added.

Client Name:	Aviana		
Card ID:	1234	Issuer Bank:	Himal

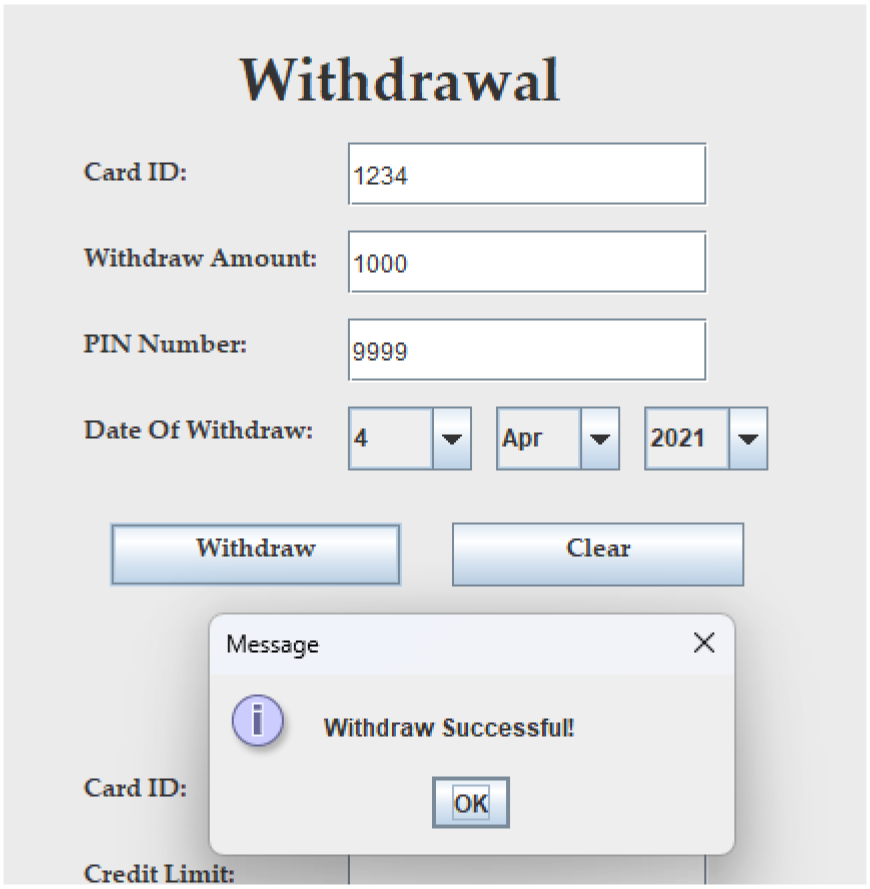
Figure 12:Screenshot of message dialogue



Withdrawal

Card ID:	1233
Withdraw Amount:	1000
PIN Number:	9999
Date Of Withdraw:	4 Mar 2021

Figure 13 Screenshot of data insertion in Withdrawal



Card ID: 1234

Withdraw Amount: 1000

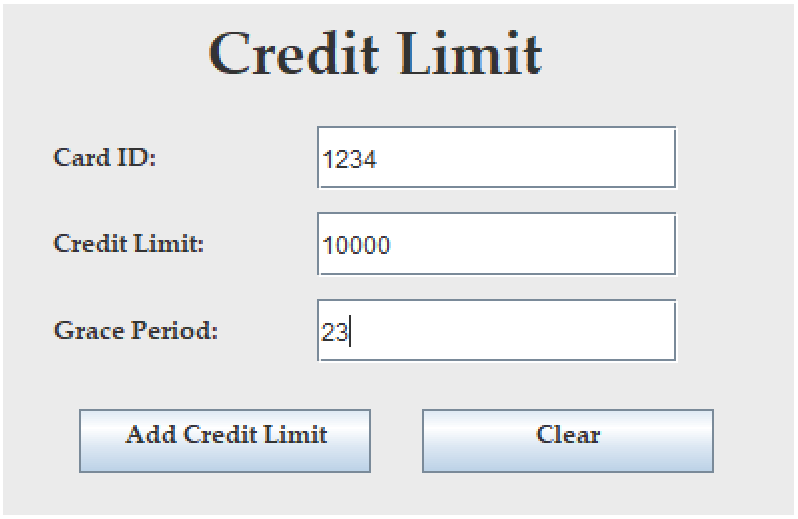
PIN Number: 9999

Date Of Withdraw: 4 Apr 2021

Message

Withdraw Successful!

Figure 14 Screenshot of message dialogue

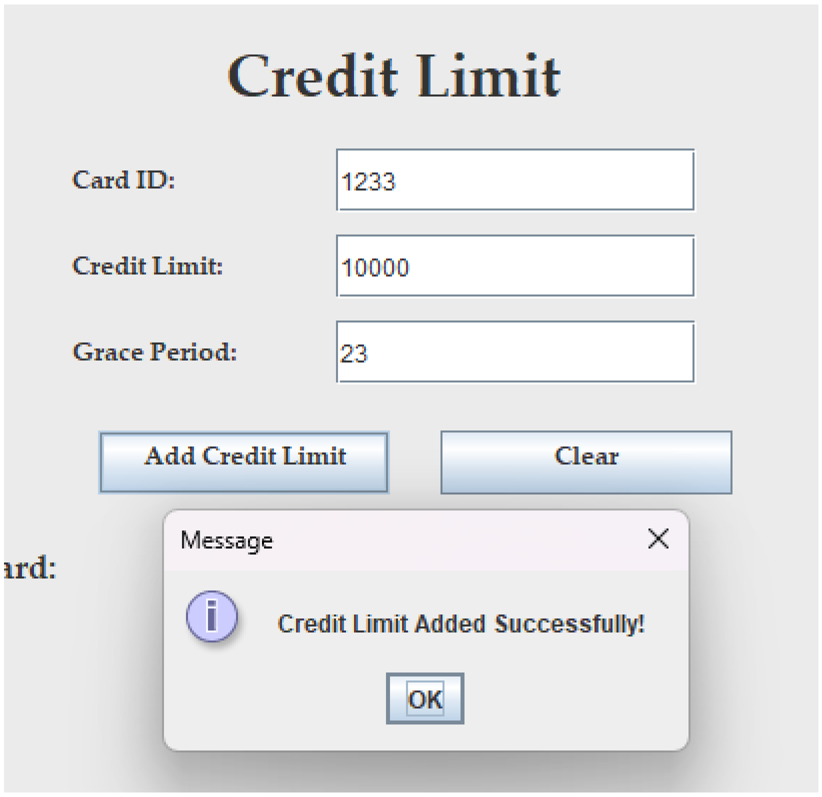


Card ID: 1234

Credit Limit: 10000

Grace Period: 23

Figure 15 Screenshot of data insertion in CreditLimit


Credit Limit

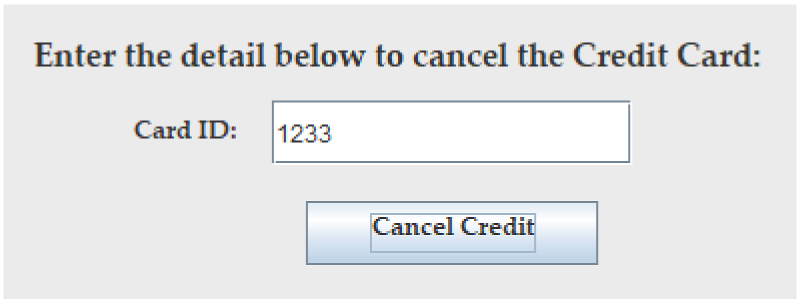
Card ID: 1233

Credit Limit: 10000

Grace Period: 23

Credit Limit Added Successfully!

Figure 16 Screenshot of message dialogue


Enter the detail below to cancel the Credit Card:

Card ID: 1233

Figure 17 Screenshot of data insertion in Cancel CreditCard

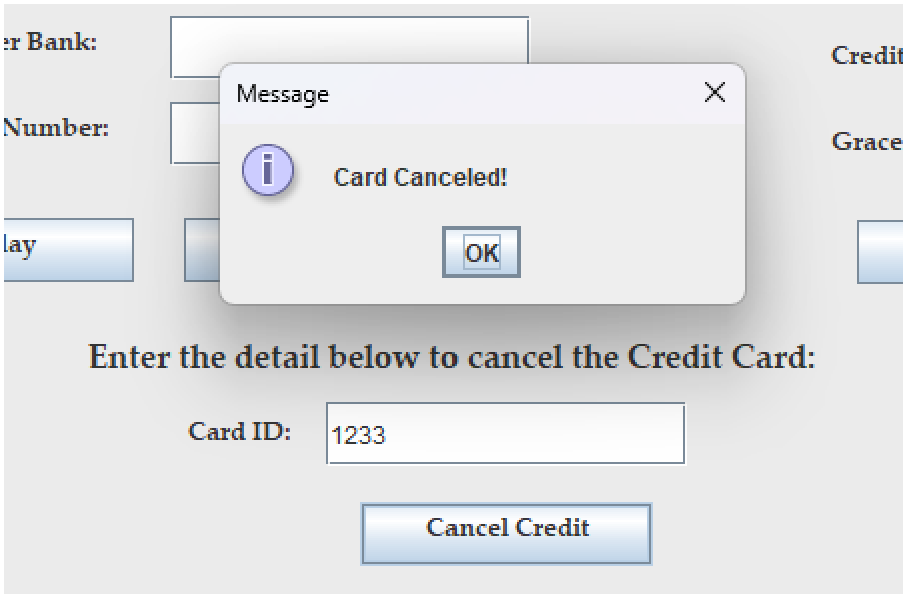


Figure 18 Screenshot of message dialogue

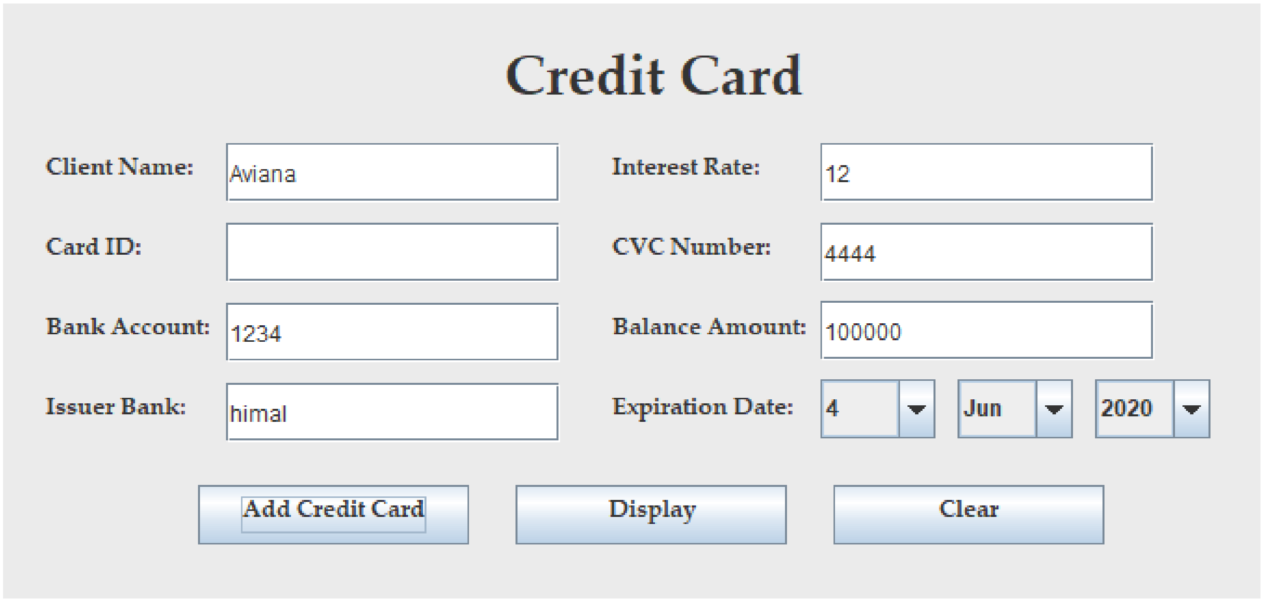
5.3. Test 3 – Check if appropriate dialog boxes appear when unsuitable values are entered

Test No.	3
Objective:	To check if appropriate dialog boxes appear when unsuitable values are entered
Action:	- DebitCard Add Button is inspected - card ID not added - Shows message no card added - CreditCard Add Button is inspected - card ID not added - Shows message no card added - Withdrawal Button is inspected - card ID not added - Shows message no card added - Set Credit Limit Button is inspected - card ID not added - Shows message no card added - Cancel CreditCard Button is inspected - Card ID not added - Shows message eanter a valid card Id

Expected Result:	Message dialogue informing the user about invalid dialogue should be displayed to the user
Actual Result:	Appropriate message dialogue is displayed
Conclusion:	The test is successful

Table 6 Inspecting CreditCard After Cancel

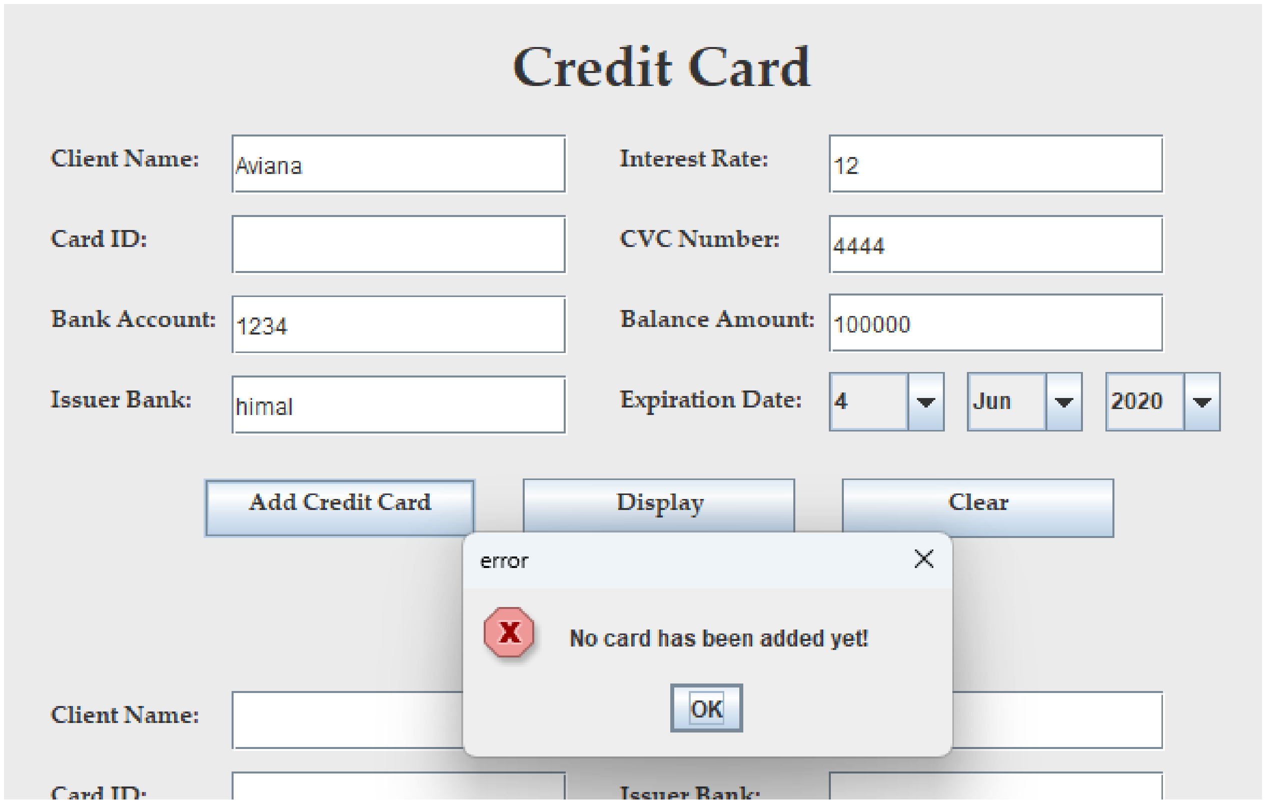
Output Result:



Credit Card

Client Name:	Aviana	Interest Rate:	12
Card ID:		CVC Number:	4444
Bank Account:	1234	Balance Amount:	100000
Issuer Bank:	himal	Expiration Date:	4 Jun 2020

Figure 19 Screenshot of not entering Card ID

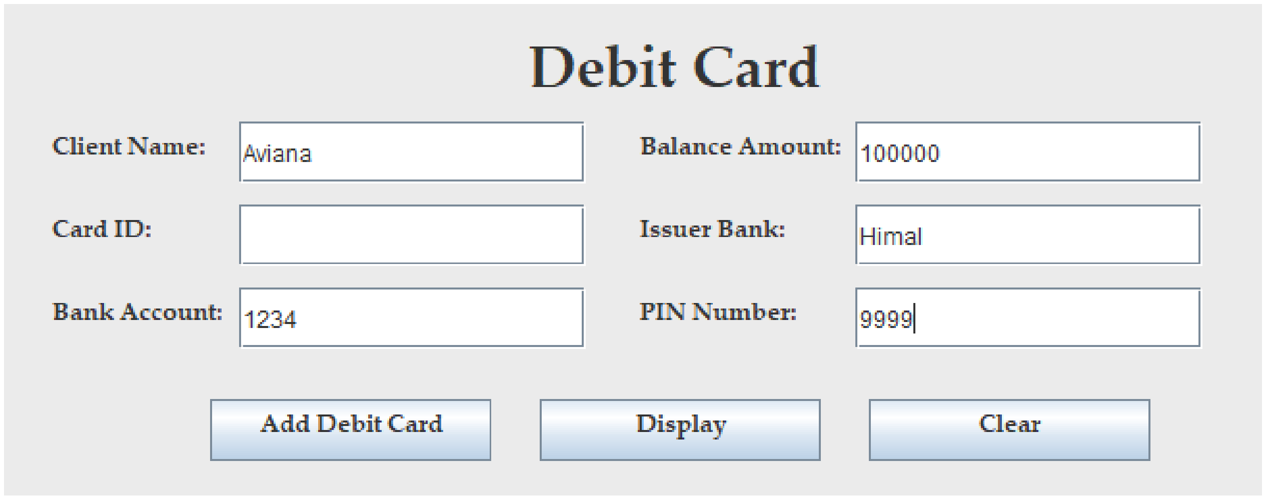

Credit Card

Client Name:	Aviana	Interest Rate:	12
Card ID:		CVC Number:	4444
Bank Account:	1234	Balance Amount:	100000
Issuer Bank:	himal	Expiration Date:	4 Jun 2020

error

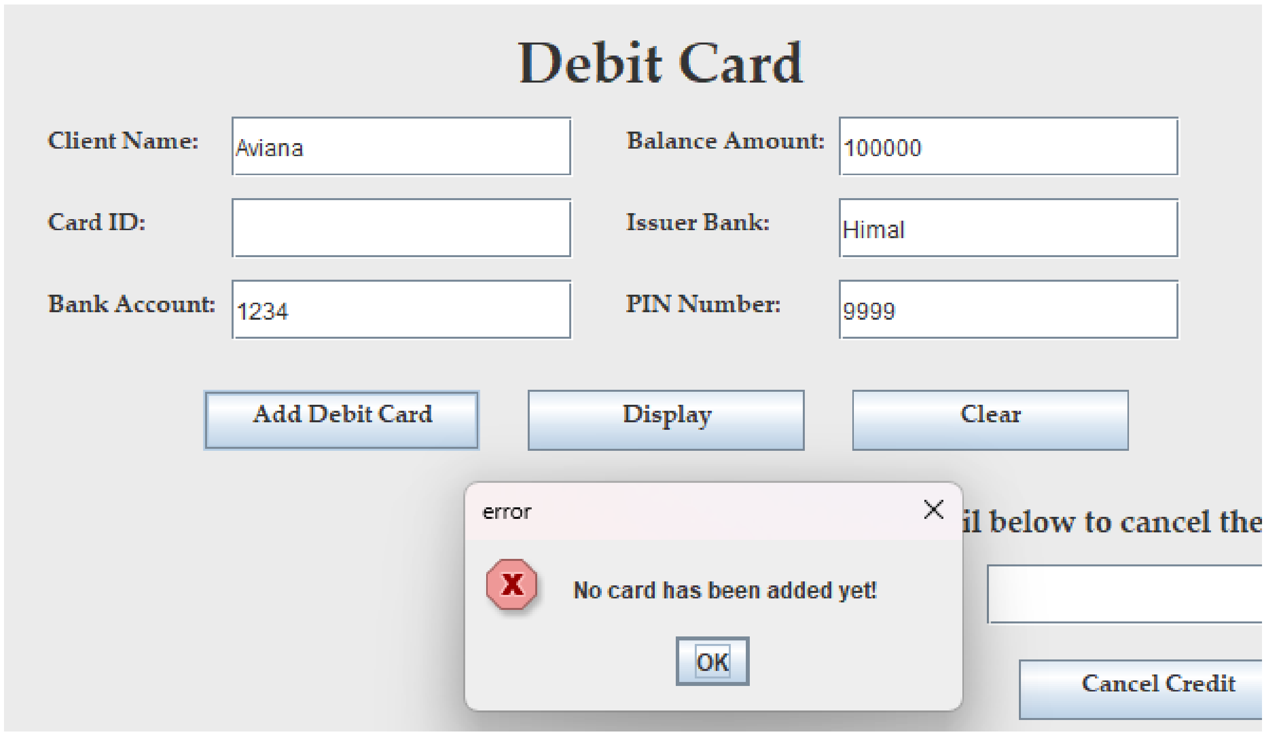
No card has been added yet!

Figure 20 Screenshot of appropriate message dialogue


Debit Card

Client Name:	Aviana	Balance Amount:	100000
Card ID:		Issuer Bank:	Himal
Bank Account:	1234	PIN Number:	9999

Figure 21 Screenshot of not entering CardID



Debit Card

Client Name: Aviana

Balance Amount: 100000

Card ID:

Issuer Bank: Himal

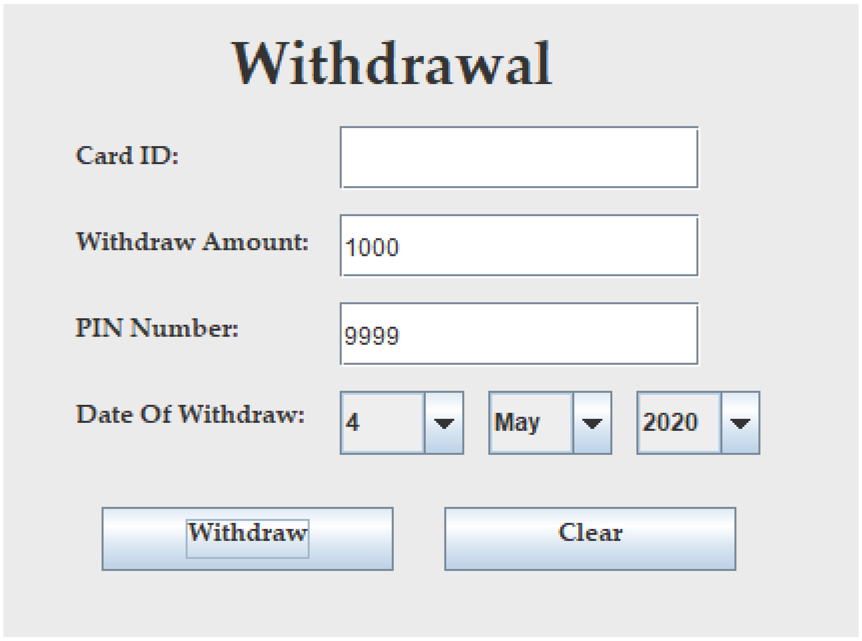
Bank Account: 1234

PIN Number: 9999

error

No card has been added yet!

Figure 22 Screenshot of appropriate message dialogue



Withdrawal

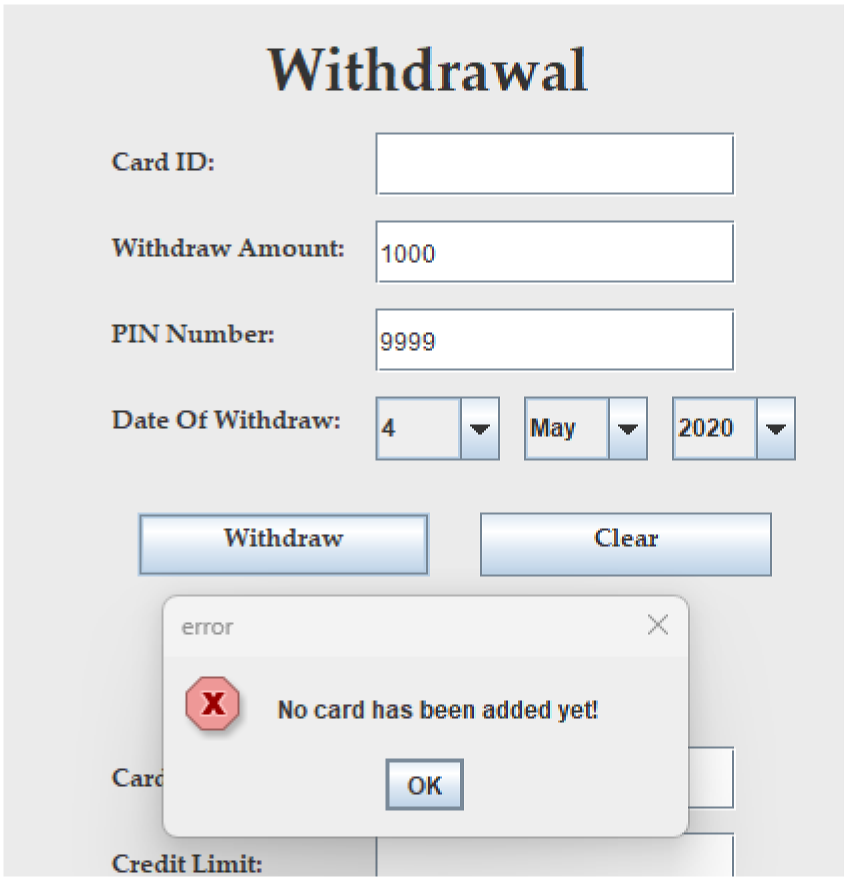
Card ID:

Withdraw Amount: 1000

PIN Number: 9999

Date Of Withdraw: 4 May 2020

Figure 23 Screenshot of not entering CardID

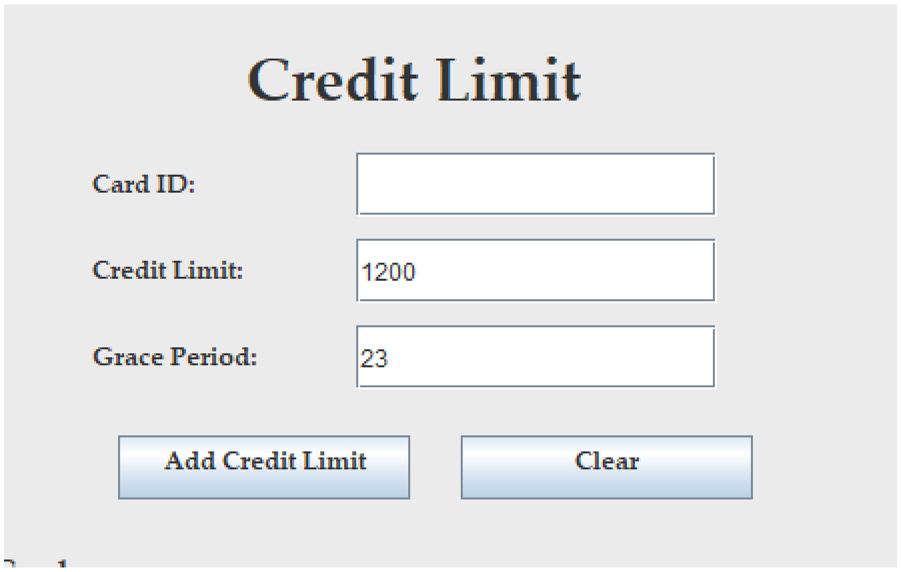

Withdrawal

Card ID:	
Withdraw Amount:	1000
PIN Number:	9999
Date Of Withdraw:	4 May 2020

error

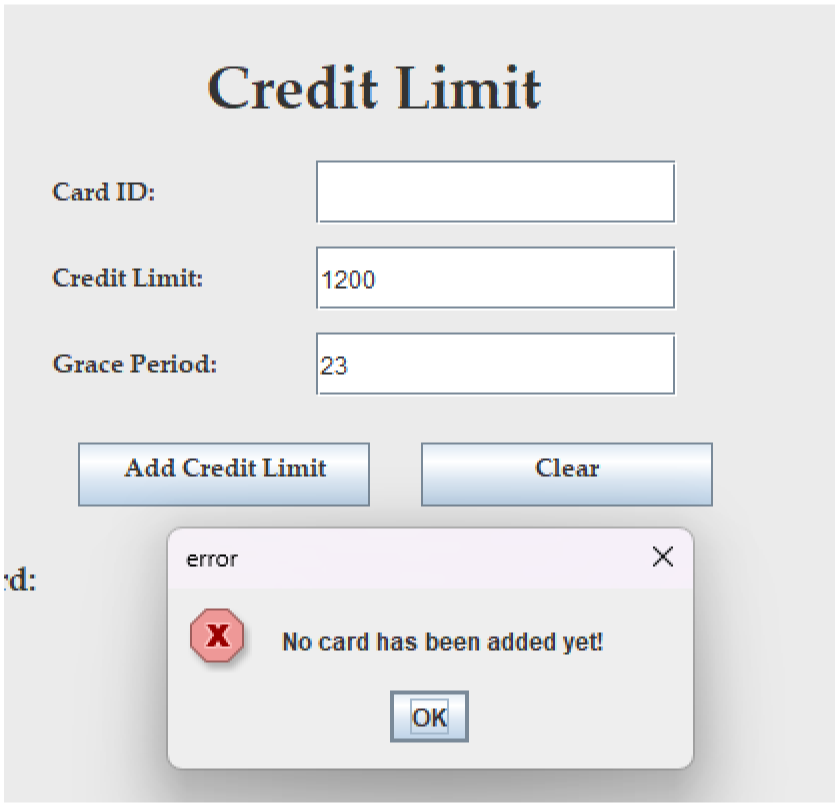
No card has been added yet!

Figure 24 Screenshot of appropriate message dialogue


Credit Limit

Card ID:	
Credit Limit:	1200
Grace Period:	23

Figure 25 Screenshot of not entering CardID



Credit Limit

Card ID:

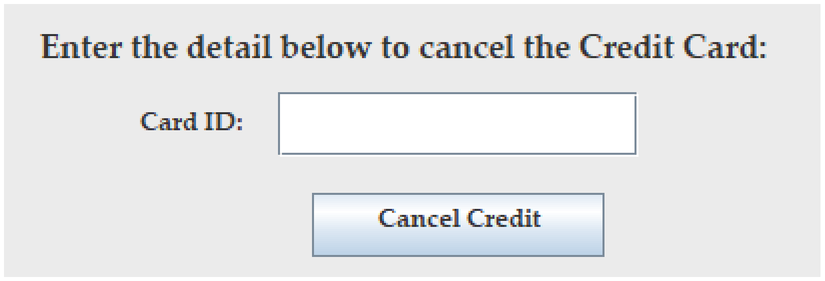
Credit Limit: 1200

Grace Period: 23

error

No card has been added yet!

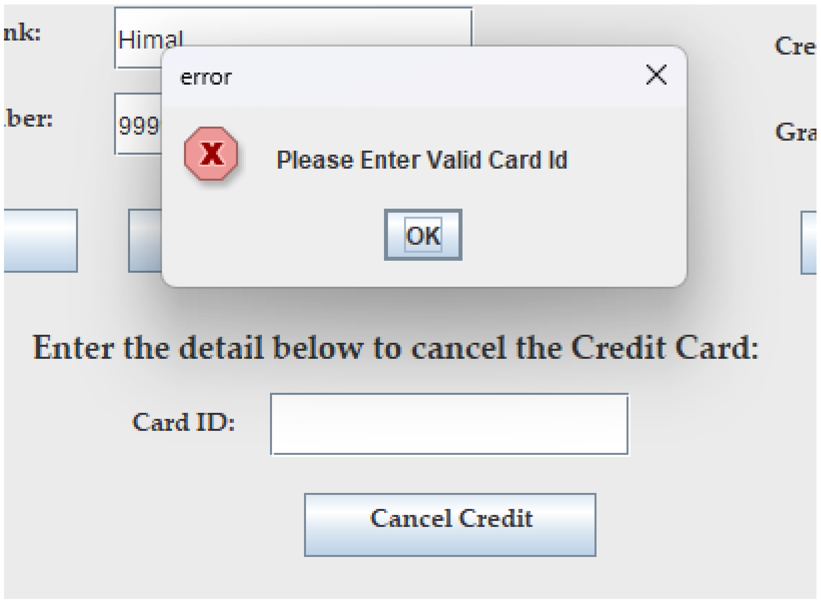
Figure 26 Screenshot of appropriate message dialogue



Enter the detail below to cancel the Credit Card:

Card ID:

Figure 27 Screenshot of not entering CardID



error

Please Enter Valid Card Id

Enter the detail below to cancel the Credit Card:

Figure 28 Screenshot of appropriate message dialogu

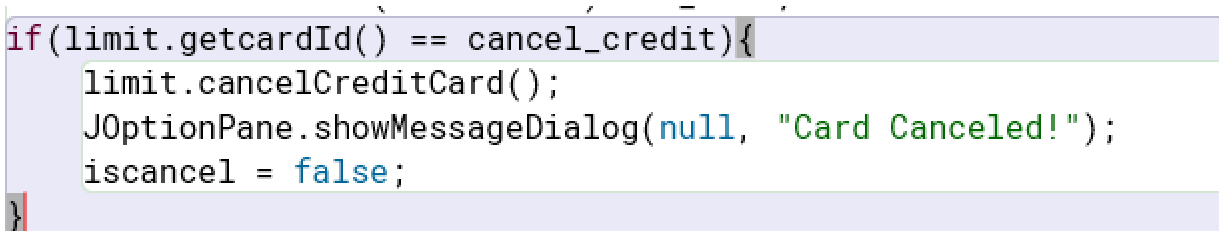
6 Error Detection and Correction

When writing code, it is common to make mistakes in the syntax, known as syntax errors. To ensure code correctness, computers use a special program called a compiler. The compiler examines the code and identifies any syntax errors, notifying you of their presence so that you can correct them. After addressing all syntax errors, you can proceed to run the program on the computer. (techopedia, 2023).

6.1 Semantic or Logical Error

A semantic error is a type of error that occurs when the code is grammatically correct but does not produce the desired or logical results. This error arises when the programmer has a misunderstanding or misinterpretation of the programs's requirements.

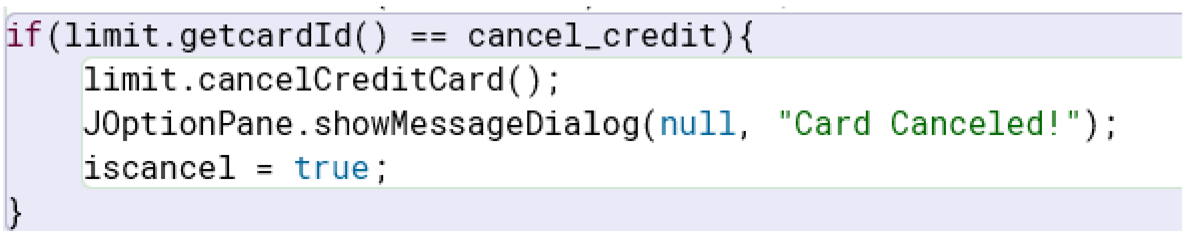
Here a code has a logical error as iscancel is set as false whereas it should be true.



```
if(limit.getcardId() == cancel_credit){
    limit.cancelCreditCard();
    JOptionPane.showMessageDialog(null, "Card Canceled!");
    iscancel = false;
}
```

Figure 29 Logical Error

The error was resolved when the iscancel button was set to true.



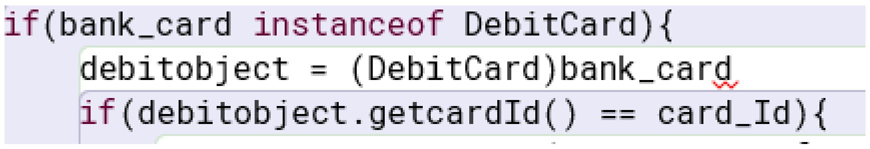
```
if(limit.getcardId() == cancel_credit){
    limit.cancelCreditCard();
    JOptionPane.showMessageDialog(null, "Card Canceled!");
    iscancel = true;
}
```

Figure 30 Correction of Logical Error

6.2 Syntax Error

Syntax errors occur when a compiler encounters code that it cannot comprehend in order to execute it. These errors commonly arise from missing punctuation or incorrectly spelled names.

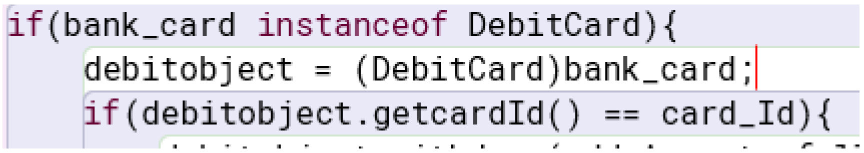
In the given code, a syntax error occurs when a semicolon is omitted from a line, resulting in an error during compilation.



```
if(bank_card instanceof DebitCard){
    debitobject = (DebitCard)bank_card
    if(debitobject.getcardId() == card_Id){
```

Figure 31 Syntax Error

The error was resolved by adding a semicolon back to the line of code.



```
if(bank_card instanceof DebitCard){
    debitobject = (DebitCard)bank_card;
    if(debitobject.getcardId() == card_Id){
```

Figure 32 Correction of Syntax Error

7. Conclusion

In this coursework, I developed a Java class using the javax.swing and java.awt packages to create a graphical user interface (GUI). The IDE BlueJ was utilized for writing and compiling the code. The coursework encompassed GUI creation, handling action events and listeners, working with ArrayLists, and utilizing concepts from the javax.swing and java.awt packages. The goal was to reinforce knowledge gained in the module.

The BankGUI class was specifically created to design a GUI for data input and utilization in previously developed classes such as BankCard, CreditCard, and DebitCard. The coursework also covered essential concepts in software development, including code writing, pseudocode, testing, and class diagrams, all crucial for building reliable software.

During the coursework, I faced challenges, particularly in researching and grasping new concepts. However, these difficulties ultimately enhanced my understanding of the module. I acquired additional knowledge on topics like different layout types, UI manager, and more, which aided me in writing GUI software with clarity and ease.

References

AWS, 2023. *What is JAVA?.* [Online]
Available at: https://aws.amazon.com/what-is/java/ [Accessed 21 1 2023].

GeeksforGeeks, 2022. *Methods in Java.* [Online]
Available at: https://www.geeksforgeeks.org/methods-in-java/ [Accessed 21 1 2023].

GeeksforGeeks, 2022. *Types of Software Testing.* [Online] Available at: https://www.geeksforgeeks.org/types-software-testing/ [Accessed 21 1 2023].

microTOOL, 2023. *What is a class Diagram?.* [Online]
Available at: https://www.microtool.de/en/knowledge-base/what-is-a-class-diagram/ [Accessed 21 1 2023].

techopedia, 2023. *What Does Syntax Error Mean?.* [Online]
Available at: https://www.techopedia.com/definition/13391/syntax-error [Accessed 23 1 2023].

theprogrammedwords, 1957. *What is pseudocode?.* [Online]
Available at: https://www.geeksforgeeks.org/how-to-write-a-pseudo-code/ [Accessed 21 1 2023].

Appendix

BankCard Class Code:

```
public class BankCard

{

   // Attributes      private int

cardId;      private String

clientName;     private String

issuerBank;     private String

bankAccount;      private int

balanceAmount;



   // Making constructor

   public BankCard(int cardId, String issuerBank, String bankAccount, int
balanceAmount)

   {

       //Assigning values of parameter to attributes

this.cardId = cardId;         this.issuerBank =

issuerBank;         this.bankAccount =

bankAccount;         this.balanceAmount =

balanceAmount;
```

```
        // Assigning empty string to clientName
this.clientName = " ";
    }

    // Mutator method for clientName     public void
setclientName(String newclientName)
    {
        clientName = newclientName;
    }

    // Mutator method for balanceAmount     public void
setbalanceAmount(int newbalanceAmount)
    {
        balanceAmount = newbalanceAmount;
    }


    // Accessor method for card_id
public int getcardId()
    {
    }

        return this.cardId;

    }


    // Accessor method for clientName

public String getclientName()

    {

        return this.clientName;

    }


    // Accessor method for issuerBank

public String getissuerBank()

    {

        return this.issuerBank;

    }


    // Accessor method for bankAccount

public String getbankAccount()

    {

        return this.bankAccount;

    }
    // Accessor method for balanceAmount
public int getbalanceAmount()
    {
       return this.balanceAmount;
    }

    // Creating a display method
public void display()
    {
       System.out.println("card_id:" +this.cardId);
       System.out.println("issuerBank:" +this.issuerBank);
       System.out.println("bankAccount:" +this.bankAccount);
       System.out.println("balanceAmount:" +this.balanceAmount);

       //Checking if clientName is empty or not
if(clientName!=" "){
         System.out.println("clientName: "+this.clientName);
       }
else{
```

```
System.out

.println("Cli

ent Name

Is Not

Assigned...

!!");


        }

    }

}
```

DebitCard Class Code:

```
public class DebitCard extends BankCard

{

    // Attributes     private int

pinNumber;      private int

withdrawalAmount;     private

String dateOfWithdrawal;

private boolean hasWithdrawn;



    // Making constructor

    public DebitCard(int balanceAmount, int cardId, String bankAccount, String
issuerBank, String clientName, int pinNumber)

    {

        // Creating a super constructor

super(cardId,issuerBank,bankAccount,balanceAmount);



        // Assigning value of parameter to PINnumber

setclientName(clientName);          this.pinNumber

= pinNumber;
```

```
        // Assigning hasWithdrawn to False
this.hasWithdrawn = false;
    }

    // Mutator method for withdrawal amount     public void
setwithdrawalAmount(int newwithdrawalAmount)
    {
        withdrawalAmount = newwithdrawalAmount;
    }

    // Accessor method for PINnumber
public int getpinNumber()
    {
        return this.pinNumber;
    }

    // Accessor method for WithdrawalAmount
public int getwithdrawalAmount()
    {
        return this.withdrawalAmount;
```

```
   }


   // Accessor method for dateOfWithdrawal
public String getdateOfWithdrawal()
   {
       return this.dateOfWithdrawal;
   }


   // Accessor method for hasWithdrawn
public boolean gethasWithdrawn()
   {
       return this.hasWithdrawn;
   }


   // Creating a method named withdraw to check if pinNumber is valid and if there is
sufficient amount     public void withdraw(int withdrawalAmount, String
dateOfWithdrawal, int pinNumber)
   {
       if(pinNumber == this.pinNumber && withdrawalAmount <= getbalanceAmount()){
super.setbalanceAmount(super.getbalanceAmount()          -          withdrawalAmount);
```

```
this.withdrawalAmount  =  withdrawalAmount;                          this.dateOfWithdrawal  =

dateOfWithdrawal;             this.hasWithdrawn = true;

        }

        else if(pinNumber != this.pinNumber){

            System.out.println("The PIN number is invalid...!!");

        }

else{

            System.out.println("Your Balance Is Insufficient...!!");

        }

    }


    // Creating a display method

public void display()

    {

        if(hasWithdrawn == true){

super.display();



            System.out.println("PINnumber: " +pinNumber);

            System.out.println("withdrawalAmount: "+withdrawalAmount);
System.out.println("dateOfWithdrawal: "+dateOfWithdrawal);
```

```
        }
else{
            System.out.println("balanceAmount: "+getbalanceAmount());
        }
    }
}
```

CreditCard Class Code:

```
public class CreditCard extends BankCard

{

    //Attributes      private int

cvcNumber;      private double

creditLimit;      private double

interestRate;     private String

expirationDate;      private int

gracePeriod;      private

boolean isGranted;



    //Creating constructor for eight parameters

    public CreditCard(int cardId,String clientName,String issuerBank,String
bankAccount,int balanceAmount,int cvcNumber,double interestRate,String
expirationDate)

    {

        //Creating a super constructor          super(cardId,

issuerBank, bankAccount, balanceAmount);

super.setclientName(clientName);



        //Assigning parameter values
    }

        this.cvcNumber = cvcNumber;
this.interestRate = interestRate;
this.expirationDate = expirationDate;

        //Asssigning isGranted to False
this.isGranted = false;
    }

    //providing accessor method for cvcNumber
public int getcvcNumber()
    {
        return this.cvcNumber;
    }

    //providing accessor method for creditLimit
public double getcreditLimit()
    {
        return this.creditLimit;
    }
    //providing accessor method for interestRate
public double getinterestRate()
    {
        return this.interestRate;
    }

    //providing accessor method for expirationDate
public String getexpirationDate()
    {
        return this.expirationDate;
    }

    //providing accessor method for gracePeriod
public int getgracePeriod()
    {
        return this.gracePeriod;
    }

    //providing accessor method for isGranted
```

```
   public boolean getisGranted()

   {

       return this.isGranted;

   }


   //Creating a method for setting credit limit      public void

setcreditLimit(double creditLimit,int gracePeriod)

   {

       if(creditLimit <= 2.5 * getbalanceAmount()){

this.creditLimit = creditLimit;             this.gracePeriod

= gracePeriod;              this.isGranted = true;

       }

else {

           System.out.println("Credit cannot be issued...!!");

       }

   }


   //Creating a method for Cancelling Credit Card public

   void cancelCreditCard()
   {        if(isGranted){
cvcNumber = 0;
creditLimit = 0;
gracePeriod = 0;
isGranted =false;
      }
   }


   //Creating a display method for the details of Credit Card
public void display()
   {
      if(isGranted == true){
super.display();
         System.out.println("cvcNumber: "+this.cvcNumber);
         System.out.println("creditLimit: "+this.creditLimit);
         System.out.println("interestRate: "+this.interestRate);
         System.out.println("ExpirationRate: "+this.expirationDate);
         System.out.println("gracePeriod: "+this.gracePeriod);
```

```
        }        else{
super.display();
            System.out.println("cvcNumber: "+this.cvcNumber);
            System.out.println("interestRate: "+this.interestRate);
            System.out.println("ExpirationRate: "+this.expirationDate);
        }
    }
}
```

BankGUI Code:

```
import javax.swing.*;

import java.awt.event.*;

import java.awt.Color;

import java.awt.Font;

import java.util.ArrayList;

import java.awt.Graphics;



public class BankGUI implements ActionListener

{

    // For Heading Bank Gui

    private JLabel formLbl;



    // Adding Background color

    private JPanel colorPanel;



    // For CreditCard

    // Declaring elements for Jlabel

    private JLabel credit_headingLbl, C_client_nameLbl, C_card_idLbl,
C_bank_accountLbl, C_issuer_bankLbl, C_interest_rateLbl,

    C_cvc_numberLbl, C_balance_amountLbl, C_expiration_dateLbl;
    // Declaring elements for JTextField

    private JTextField C_client_nameTxt, C_card_idTxt, C_bank_accountTxt,
C_issuer_bankTxt, C_interest_rateTxt, C_cvc_numberTxt,

    C_balance_amountTxt;

    // Declaring elemets for JCombobox

    private JComboBox Credit_day, C_monthcombo, Credit_year;

    // Declaring elements for JButton

    private JButton C_addBtn, C_displayBtn, C_cancelBtn;



    // For DebitCard

    // Declaring elements for Jlabel

    private JLabel debit_headingLbl, D_client_nameLbl, D_card_idLbl,
D_bank_accountLbl, D_balance_amountLbl, D_issuer_bankLbl,

    D_pin_numberLbl;

    // Declaring elements for JTextField

    private JTextField D_client_nameTxt, D_card_idTxt, D_bank_accountTxt,
D_balance_amountTxt, D_pin_numberTxt, D_issuer_bankTxt;

    // Declaring elements for JButton

    private JButton D_addBtn, D_displayBtn, D_cancelBtn;



    // For Withdrawal

    // Declaring elements for Jlabel
    private JLabel drawal_headingLbl, W_card_idLbl, W_amountLbl, W_pinLbl,
W_dowLbl;

    // Declaring elements for JTextField

    private JTextField W_card_idTxt, W_amountTxt, W_pinTxt;

    // Declaring elements for JButton

    private JButton W_addBtn, W_clearBtn;

    // Declaring elements for Jcombo Box

    private JComboBox With_day, W_monthcombo, With_year;



    // For Credit Limit

    // Declaring elements for Jlabel

    private JLabel limit_headingLbl, L_card_idLbl, L_credit_limitLbl, L_graceLbl;

    // Declaring elements for JTextField

    private JTextField L_card_idTxt, L_credit_limitTxt, L_graceTxt;

    // Declaring elements for JButton

    private JButton L_addBtn, L_clearBtn;



    // For Cancel Credit

    // Declaring elements for Jlabel

    private JLabel cancel_headingLbl, cancel_card_idLbl;

    // Declaring elements for JTextField
```

```
private JTextField cancel_card_idTxt;

// Declaring elements for JButton

private JButton cancel_clearBtn;



// Storing the variable in BankCard

ArrayList<BankCard> gui_Card = new ArrayList<BankCard>();

DebitCard debitobject;

CreditCard creditobject;



public BankGUI(){

    // Creating the Jframe

    JFrame myFrame = new JFrame("CourseWork");

    myFrame.setSize(1280, 810);

    myFrame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

    myFrame.setLayout(null);

    myFrame.setResizable(false);

    myFrame.setVisible(true);



    // Creating the components

    // Bank Form

    formLbl = new JLabel("Bank Form");
```

```
formLbl.setFont(new Font("Palatino Linotype",Font.BOLD,40));

formLbl.setBounds(545, 18, 287, 60);

myFrame.add(formLbl);


// For Credit Card

// Request Credit card Detail Heading

credit_headingLbl = new JLabel("Credit Card");

credit_headingLbl.setFont(new Font("Palatino Linotype",Font.BOLD,30));

credit_headingLbl.setBounds(300, 73, 460, 50);

myFrame.add(credit_headingLbl);


// Client Name (Credit)

C_client_nameLbl = new JLabel("Client Name:");

C_client_nameLbl.setFont(new Font("Palatino Linotype",Font.BOLD,13));

C_client_nameLbl.setBounds(53, 132, 82, 20);

myFrame.add(C_client_nameLbl);


// Text Field For Client Name

C_client_nameTxt = new JTextField();

C_client_nameTxt.setBounds(150, 126, 180, 32);

myFrame.add(C_client_nameTxt);
```

```
// Card ID (Credit)

C_card_idLbl = new JLabel("Card ID:");

C_card_idLbl.setFont(new Font("Palatino Linotype",Font.BOLD,13));

C_card_idLbl.setBounds(53, 175, 56, 20);

myFrame.add(C_card_idLbl);


// Text Field for Card ID

C_card_idTxt = new JTextField();

C_card_idTxt.setBounds(150, 169, 180, 32);

myFrame.add(C_card_idTxt);


// Bank Account (Credit)

C_bank_accountLbl = new JLabel("Bank Account:");

C_bank_accountLbl.setFont(new Font("Palatino Linotype",Font.BOLD,13));

C_bank_accountLbl.setBounds(53, 218, 93, 20);

myFrame.add(C_bank_accountLbl);


// Text Field for Bank Account

C_bank_accountTxt = new JTextField();

C_bank_accountTxt.setBounds(150, 212, 180, 32);
```

```
myFrame.add(C_bank_accountTxt);


// Issuer Bank (Credit)

C_issuer_bankLbl = new JLabel("Issuer Bank:");

C_issuer_bankLbl.setFont(new Font("Palatino Linotype",Font.BOLD,13));

C_issuer_bankLbl.setBounds(53, 261, 82, 20);

myFrame.add(C_issuer_bankLbl);


// Text Field for Issuer Bank

C_issuer_bankTxt = new JTextField();

C_issuer_bankTxt.setBounds(150, 255, 180, 32);

myFrame.add(C_issuer_bankTxt);


// Interest Rate (Credit)

C_interest_rateLbl = new JLabel("Interest Rate:");

C_interest_rateLbl.setFont(new Font("Palatino Linotype",Font.BOLD,13));

C_interest_rateLbl.setBounds(358, 132, 88, 20);

myFrame.add(C_interest_rateLbl);


// Text Field for Interest Rate

C_interest_rateTxt = new JTextField();
```

```
C_interest_rateTxt.setBounds(470, 126, 180, 32);

myFrame.add(C_interest_rateTxt);


// CVC Number (Credit)

C_cvc_numberLbl = new JLabel("CVC Number:");

C_cvc_numberLbl.setFont(new Font("Palatino Linotype",Font.BOLD,13));

C_cvc_numberLbl.setBounds(358, 175, 91, 20);

myFrame.add(C_cvc_numberLbl);


// Text Field for Interest Rate

C_cvc_numberTxt = new JTextField();

C_cvc_numberTxt.setBounds(470, 169, 180, 32);

myFrame.add(C_cvc_numberTxt);


// Balance Amount (Credit)

C_balance_amountLbl = new JLabel("Balance Amount:");

C_balance_amountLbl.setFont(new Font("Palatino Linotype",Font.BOLD,13));

C_balance_amountLbl.setBounds(358, 218, 110, 20);

myFrame.add(C_balance_amountLbl);


// Text Field for Balance Amount
```

```
        C_balance_amountTxt = new JTextField();

        C_balance_amountTxt.setBounds(470, 211, 180, 32);

        myFrame.add(C_balance_amountTxt);



        // Expiration Date (Credit)

        C_expiration_dateLbl = new JLabel("Expiration Date:");

        C_expiration_dateLbl.setFont(new Font("Palatino Linotype",Font.BOLD,13));

        C_expiration_dateLbl.setBounds(358, 261, 100, 20);

        myFrame.add(C_expiration_dateLbl);



        // Day for Expiration Date

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

        Credit_day = new JComboBox(C_day);

        Credit_day.setBounds(470, 253, 62, 32);

        myFrame.add(Credit_day);



        // Month for Expiration Date

        String[]
C_month={"Jan","Feb","Mar","Apr","May","Jun","Jul","Aug","Sept","Oct","Nov","Dec"};

        C_monthcombo = new JComboBox(C_month);
```

```
C_monthcombo.setBounds(544, 253, 62, 32);

myFrame.add(C_monthcombo);



// Loop for Year

String[] C_year = {"2019","2020","2021"};

Credit_year = new JComboBox(C_year);

Credit_year.setBounds(618, 253, 62, 32);

myFrame.add(Credit_year);



// Button for adding Credit Card

C_addBtn = new JButton("Add Credit Card");

C_addBtn.setFont(new Font("Palatino Linotype",Font.BOLD,13));

C_addBtn.setBounds(135, 310, 146, 32);

myFrame.add(C_addBtn);



// Button for displaying Credit Card

C_displayBtn = new JButton("Display");

C_displayBtn.setFont(new Font("Palatino Linotype",Font.BOLD,13));

C_displayBtn.setBounds(306, 310, 146, 32);

myFrame.add(C_displayBtn);
```

```
// Button for Cancel Credit Card

C_cancelBtn = new JButton("Clear");

C_cancelBtn.setFont(new Font("Palatino Linotype",Font.BOLD,13));

C_cancelBtn.setBounds(477, 310, 146, 32);

myFrame.add(C_cancelBtn);


// For Debit Card

// Request Debit card Detail Heading

debit_headingLbl = new JLabel("Debit Card");

debit_headingLbl.setFont(new Font("Palatino Linotype",Font.BOLD,30));

debit_headingLbl.setBounds(300, 379, 430, 50);

myFrame.add(debit_headingLbl);


// Client Name (Debit)

D_client_nameLbl = new JLabel("Client Name:");

D_client_nameLbl.setFont(new Font("Palatino Linotype",Font.BOLD,13));

D_client_nameLbl.setBounds(53, 430, 85, 20);

myFrame.add(D_client_nameLbl);


// Text Field For Client Name

D_client_nameTxt = new JTextField();
```

```
D_client_nameTxt.setBounds(150, 424, 180, 32);

myFrame.add(D_client_nameTxt);


// Card ID (Debit)

D_card_idLbl = new JLabel("Card ID:");

D_card_idLbl.setFont(new Font("Palatino Linotype",Font.BOLD,13));

D_card_idLbl.setBounds(53, 473, 54, 20);

myFrame.add(D_card_idLbl);


// Text Field for Card ID

D_card_idTxt = new JTextField();

D_card_idTxt.setBounds(150, 467, 180, 32);

myFrame.add(D_card_idTxt);


// Bank Account (Debit)

D_bank_accountLbl = new JLabel("Bank Account:");

D_bank_accountLbl.setFont(new Font("Palatino Linotype",Font.BOLD,13));

D_bank_accountLbl.setBounds(53, 516, 93, 20);

myFrame.add(D_bank_accountLbl);


// Text Field for Bank Account
```

```
D_bank_accountTxt = new JTextField();

D_bank_accountTxt.setBounds(150, 510, 180, 32);

myFrame.add(D_bank_accountTxt);


// Balance Amount (Debit)

D_balance_amountLbl = new JLabel("Balance Amount:");

D_balance_amountLbl.setFont(new Font("Palatino Linotype",Font.BOLD,13));

D_balance_amountLbl.setBounds(358, 430, 108, 20);

myFrame.add(D_balance_amountLbl);


// Text Field for Balance Amount

D_balance_amountTxt = new JTextField();

D_balance_amountTxt.setBounds(470, 424, 180, 32);

myFrame.add(D_balance_amountTxt);


// Issuer Bank (Debit)

D_issuer_bankLbl = new JLabel("Issuer Bank:");

D_issuer_bankLbl.setFont(new Font("Palatino Linotype",Font.BOLD,13));

D_issuer_bankLbl.setBounds(358, 473, 82, 20);

myFrame.add(D_issuer_bankLbl);
```

```
// Text Field for Issuer Bank

D_issuer_bankTxt = new JTextField();

D_issuer_bankTxt.setBounds(470, 467, 180, 32);

myFrame.add(D_issuer_bankTxt);


// PIN Number (Debit)

D_pin_numberLbl = new JLabel("PIN Number:");

D_pin_numberLbl.setFont(new Font("Palatino Linotype",Font.BOLD,13));

D_pin_numberLbl.setBounds(358, 516, 85, 20);

myFrame.add(D_pin_numberLbl);


// Text Field for PIN Number

D_pin_numberTxt = new JTextField();

D_pin_numberTxt.setBounds(470, 510, 180, 32);

myFrame.add(D_pin_numberTxt);


// Button for adding Debit Card

D_addBtn = new JButton("Add Debit Card");

D_addBtn.setFont(new Font("Palatino Linotype",Font.BOLD,13));

D_addBtn.setBounds(135, 568, 146, 32);

myFrame.add(D_addBtn);
```

```
// Button for displaying Debit Card

D_displayBtn = new JButton("Display");

D_displayBtn.setFont(new Font("Palatino Linotype",Font.BOLD,13));

D_displayBtn.setBounds(306, 568, 146, 32);

myFrame.add(D_displayBtn);


// Button for Cancel Debit Card

D_cancelBtn = new JButton("Clear");

D_cancelBtn.setFont(new Font("Palatino Linotype",Font.BOLD,13));

D_cancelBtn.setBounds(477, 568, 146, 32);

myFrame.add(D_cancelBtn);


// For Withdrawal

// Request Withdrawal Detail Heading

drawal_headingLbl = new JLabel("Withdrawal");

drawal_headingLbl.setFont(new Font("Palatino Linotype",Font.BOLD,30));

drawal_headingLbl.setBounds(877, 73, 520, 50);

myFrame.add(drawal_headingLbl);


// Card ID (Withdrawal)
```

```
W_card_idLbl = new JLabel("Card ID:");

W_card_idLbl.setFont(new Font("Palatino Linotype",Font.BOLD,13));

W_card_idLbl.setBounds(800, 129, 56, 20);

myFrame.add(W_card_idLbl);


// Text Field for Card ID

W_card_idTxt = new JTextField();

W_card_idTxt.setBounds(932, 121, 180, 32);

myFrame.add(W_card_idTxt);


// WithDraw Amount (Withdrawal)

W_amountLbl = new JLabel("Withdraw Amount:");

W_amountLbl.setFont(new Font("Palatino Linotype",Font.BOLD,13));

W_amountLbl.setBounds(800, 172, 128, 20);

myFrame.add(W_amountLbl);


// Text Field for WithDraw Amount

W_amountTxt = new JTextField();

W_amountTxt.setBounds(932, 165, 180, 32);

myFrame.add(W_amountTxt);
```

```
        // PIN Number (Withdrawal)

        W_pinLbl = new JLabel("PIN Number:");

        W_pinLbl.setFont(new Font("Palatino Linotype",Font.BOLD,13));

        W_pinLbl.setBounds(800, 215, 85, 20);

        myFrame.add(W_pinLbl);



        // Text Field for PIN Number

        W_pinTxt = new JTextField();

        W_pinTxt.setBounds(932, 209, 180, 32);

        myFrame.add(W_pinTxt);



        // Date Of Withdraw (WithDrawal)

        W_dowLbl = new JLabel("Date Of Withdraw:");

        W_dowLbl.setFont(new Font("Palatino Linotype",Font.BOLD,13));

        W_dowLbl.setBounds(800, 258, 124, 20);

        myFrame.add(W_dowLbl);



        // Day for Date Of Withdraw

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

        With_day = new JComboBox(W_day);
        With_day.setBounds(932, 253, 62, 32);

        myFrame.add(With_day);



        // Month for Date Of Withdraw

        String[]
W_month={"Jan","Feb","Mar","Apr","May","Jun","Jul","Aug","Sept","Oct","Nov","Dec"};

        W_monthcombo = new JComboBox(W_month);

        W_monthcombo.setBounds(1006, 253, 62, 32);

        myFrame.add(W_monthcombo);



        // Year for Expiration Date

        String[] W_year = {"2019","2020","2021"};

        With_year = new JComboBox(W_year);

        With_year.setBounds(1080, 253, 62, 32);

        myFrame.add(With_year);



        // Button for withdraw

        W_addBtn = new JButton("Withdraw");

        W_addBtn.setFont(new Font("Palatino Linotype",Font.BOLD,13));

        W_addBtn.setBounds(813, 311, 146, 32);

        myFrame.add(W_addBtn);
```

```
// Button for Clear

W_clearBtn = new JButton("Clear");

W_clearBtn.setFont(new Font("Palatino Linotype",Font.BOLD,13));

W_clearBtn.setBounds(984, 311, 146, 32);

myFrame.add(W_clearBtn);



// For Credit Limit

// Request Credit Limit Detail Heading

limit_headingLbl = new JLabel("Credit Limit");

limit_headingLbl.setFont(new Font("Palatino Linotype",Font.BOLD,30));

limit_headingLbl.setBounds(877, 375, 400, 50);

myFrame.add(limit_headingLbl);



// Card ID (Limit)

L_card_idLbl = new JLabel("Card ID:");

L_card_idLbl.setFont(new Font("Palatino Linotype",Font.BOLD,13));

L_card_idLbl.setBounds(800, 437, 56, 20);

myFrame.add(L_card_idLbl);



// Text Field for Card ID
```

```
L_card_idTxt = new JTextField();

L_card_idTxt.setBounds(932, 428, 180, 32);

myFrame.add(L_card_idTxt);


// Credit Limit (Limit)

L_credit_limitLbl = new JLabel("Credit Limit:");

L_credit_limitLbl.setFont(new Font("Palatino Linotype",Font.BOLD,13));

L_credit_limitLbl.setBounds(800, 480, 79, 20);

myFrame.add(L_credit_limitLbl);


// Text Field for Credit Limit

L_credit_limitTxt = new JTextField();

L_credit_limitTxt.setBounds(932, 471, 180, 32);

myFrame.add(L_credit_limitTxt);


// Grace Period (Limit)

L_graceLbl = new JLabel("Grace Period:");

L_graceLbl.setFont(new Font("Palatino Linotype",Font.BOLD,13));

L_graceLbl.setBounds(800, 523, 90, 20);

myFrame.add(L_graceLbl);
```

```
        // Text Field for Grace Period

        L_graceTxt = new JTextField();

        L_graceTxt.setBounds(932, 514, 180, 32);

        myFrame.add(L_graceTxt);



        // Button for Limit

        L_addBtn = new JButton("Add Credit Limit");

        L_addBtn.setFont(new Font("Palatino Linotype",Font.BOLD,13));

        L_addBtn.setBounds(813, 569, 146, 32);

        myFrame.add(L_addBtn);



        // Button for Limit

        L_clearBtn = new JButton("Clear");

        L_clearBtn.setFont(new Font("Palatino Linotype",Font.BOLD,13));

        L_clearBtn.setBounds(984, 569, 146, 32);

        myFrame.add(L_clearBtn);



        // For Cancel Credit Card

        // Request Cancel Credit Card Detail Heading

        cancel_headingLbl = new JLabel("Enter the detail below to cancel the Credit
Card:");
```

```
cancel_headingLbl.setFont(new Font("Palatino Linotype",Font.BOLD,17));

cancel_headingLbl.setBounds(429, 630, 390, 25);

myFrame.add(cancel_headingLbl);


// Card ID (cancel)

cancel_card_idLbl = new JLabel("Card ID:");

cancel_card_idLbl.setFont(new Font("Palatino Linotype",Font.BOLD,13));

cancel_card_idLbl.setBounds(479, 668, 56, 20);

myFrame.add(cancel_card_idLbl);


// Text Field for Card ID

cancel_card_idTxt = new JTextField();

cancel_card_idTxt.setBounds(548, 660, 180, 32);

myFrame.add(cancel_card_idTxt);


// Button for Cancel Credit Card

cancel_clearBtn = new JButton("Cancel Credit");

cancel_clearBtn.setFont(new Font("Palatino Linotype",Font.BOLD,13));

cancel_clearBtn.setBounds(565, 710, 146, 32);

myFrame.add(cancel_clearBtn);
```

```
// Add ActionListener For Clear

C_cancelBtn.addActionListener(this);

D_cancelBtn.addActionListener(this);

W_clearBtn.addActionListener(this);

L_clearBtn.addActionListener(this);


// Add ActionListener For Add

D_addBtn.addActionListener(this);

C_addBtn.addActionListener(this);

W_addBtn.addActionListener(this);

L_addBtn.addActionListener(this);

cancel_clearBtn.addActionListener(this);


// Add ActionListener For Display

C_displayBtn.addActionListener(this);

D_displayBtn.addActionListener(this);


// Background for myFrame

JPanel colorPanel = new JPanel();

colorPanel.setBackground(new Color(235, 235, 235));

colorPanel.setBounds(0,0,1280,810);
```

```
        myFrame.add(colorPanel);
    }

    // Action Listener For All Buttons
    public void actionPerformed(ActionEvent e){
        //--------------------- For Credit Card ---------------------//
        // Add For Credit Card
        if(e.getSource() == C_addBtn){
            try{
                String C_add_client_name = C_client_nameTxt.getText();
                String C_add_card_id = C_card_idTxt.getText();
                String C_add_bank_account = C_bank_accountTxt.getText();
                String C_add_issuer_bank = C_issuer_bankTxt.getText();
                String C_add_interest_rate = C_interest_rateTxt.getText();
                String C_add_cvc_number = C_cvc_numberTxt.getText();
                String C_add_balance_amount = C_balance_amountTxt.getText();
                String day = (String)Credit_day.getSelectedItem();
                String month = (String)C_monthcombo.getSelectedItem();
                String year = (String)Credit_year.getSelectedItem();
                String fulldate = day + month + year;
                int card_Id = Integer.parseInt(C_add_card_id);

                int balance_Amount = Integer.parseInt(C_add_balance_amount);

                int cvc_Number = Integer.parseInt(C_add_cvc_number);

                double interest_Rate = Integer.parseInt(C_add_interest_rate);



                boolean isadd = false;

                for(BankCard bank_card: gui_Card){

                    if(bank_card.getcardId() == card_Id){

                        JOptionPane.showMessageDialog(null, "You Already Added Credit
    Card!");

                        isadd = true;

                        break;

                    }

                }

                if(!isadd){

                    creditobject = new CreditCard(card_Id, C_add_client_name,
    C_add_issuer_bank, C_add_bank_account, balance_Amount, cvc_Number,
    interest_Rate, fulldate);

                    gui_Card.add(creditobject);

                    JOptionPane.showMessageDialog(null, "Card has been added.");

                }

            }catch(NumberFormatException ex){
                JOptionPane.showMessageDialog(null, "No card has been added yet!",
    "error", JOptionPane.ERROR_MESSAGE);

            }

         }



         // Cancel For Credit Card

         if(e.getSource() == C_cancelBtn){

            C_client_nameTxt.setText("");

            C_card_idTxt.setText("");

            C_bank_accountTxt.setText("");

            C_issuer_bankTxt.setText("");

            C_interest_rateTxt.setText("");

            C_cvc_numberTxt.setText("");

            C_balance_amountTxt.setText("");

            Credit_day.setSelectedIndex(0);

            C_monthcombo.setSelectedIndex(0);

            Credit_year.setSelectedIndex(0);

         }



         if(e.getSource() == C_displayBtn){

            for(BankCard bank_card: gui_Card){
```

```
        if(bank_card instanceof CreditCard){
            bank_card.display();
        }
    }
}

//--------------------- For Debit Card ---------------------//
// Add For Debit Card
if(e.getSource() == D_addBtn){
    try{
        String D_add_client_name = D_client_nameTxt.getText();
        String D_add_card_id = D_card_idTxt.getText();
        String D_add_bank_account = D_bank_accountTxt.getText();
        String D_add_balance_amount = D_balance_amountTxt.getText();
        String D_add_pin_number = D_pin_numberTxt.getText();
        String D_add_issuer_bank = D_issuer_bankTxt.getText();

        int card_Id = Integer.parseInt(D_add_card_id);
        int balance_Amount = Integer.parseInt(D_add_balance_amount);
        int pin_Number = Integer.parseInt(D_add_pin_number);
```

```
            boolean isadd = false;

            for(BankCard bank_card: gui_Card){

                if(bank_card instanceof DebitCard){

                    debitobject = (DebitCard)bank_card;

                    if(debitobject.getcardId() == card_Id){

                        JOptionPane.showMessageDialog(null, "You Already Added A Card
With This Info!");

                        isadd = true;

                        break;

                    }

                    else{

                        isadd = false;

                    }

                }

            }

            if(!isadd){

                debitobject = new DebitCard(balance_Amount, card_Id,
D_add_bank_account, D_add_issuer_bank, D_add_client_name,pin_Number);

                gui_Card.add(debitobject);

                JOptionPane.showMessageDialog(null, "Card has been added.");

            }

        }catch(NumberFormatException ex){
            JOptionPane.showMessageDialog(null, "No card has been added yet!",
"error", JOptionPane.ERROR_MESSAGE);

        }

    }



    if(e.getSource() == D_displayBtn){

        for(BankCard bank_card: gui_Card){

            if(bank_card instanceof DebitCard){

                bank_card.display();

            }

        }

    }



    // Cancel For Debit Card

    if(e.getSource() == D_cancelBtn){

        D_client_nameTxt.setText("");

        D_card_idTxt.setText("");

        D_bank_accountTxt.setText("");

        D_balance_amountTxt.setText("");

        D_pin_numberTxt.setText("");

        D_issuer_bankTxt.setText("");
```

```
}


//---------------------- For Withdrawal ----------------------//

// Add For Withdrawal

if(e.getSource() == W_addBtn){

    try{

        String W_add_card_id = W_card_idTxt.getText();

        String W_add_amount = W_amountTxt.getText();

        String W_add_pin = W_pinTxt.getText();

        String day = (String)With_day.getSelectedItem();

        String month = (String)W_monthcombo.getSelectedItem();

        String year = (String)With_year.getSelectedItem();

        String fulldate = day + month + year;


        int card_Id = Integer.parseInt(W_add_card_id);

        int add_Amount = Integer.parseInt(W_add_amount);

        int pin_Number = Integer.parseInt(W_add_pin);


        String D_add_client_name = D_client_nameTxt.getText();

        String D_add_card_id = D_card_idTxt.getText();

        String D_add_bank_account = D_bank_accountTxt.getText();
        String D_add_balance_amount = D_balance_amountTxt.getText();

        String D_add_pin_number = D_pin_numberTxt.getText();

        String D_add_issuer_bank = D_issuer_bankTxt.getText();


        int D_card_Id = Integer.parseInt(D_add_card_id);

        int D_balance_Amount = Integer.parseInt(D_add_balance_amount);

        int D_pin_Number = Integer.parseInt(D_add_pin_number);


        boolean isadd = false;

        for(BankCard bank_card: gui_Card){

            if(bank_card instanceof DebitCard){

                debitobject = (DebitCard)bank_card;

                if(debitobject.getcardId() == card_Id){

                    debitobject.withdraw(add_Amount, fulldate, pin_Number);

                    JOptionPane.showMessageDialog(null, "Withdraw Successful!");

                    isadd = true;

                    break;

                }

            }

            else{

                isadd = false;
```

```
                }

            }

        }catch(NumberFormatException ex){

            JOptionPane.showMessageDialog(null, "No card has been added yet!",
"error", JOptionPane.ERROR_MESSAGE);

        }

    }



    // Cancel For Withdrawal

    if(e.getSource() == W_clearBtn){

        W_card_idTxt.setText("");

        W_amountTxt.setText("");

        W_pinTxt.setText("");

        With_day.setSelectedIndex(0);

        W_monthcombo.setSelectedIndex(0);

        With_year.setSelectedIndex(0);

    }



    //---------------------- For Credit Limit ----------------------//

    // Add For Credit Limit

    if(e.getSource() == L_addBtn){
        try{
            String L_add_card_id = L_card_idTxt.getText();
            String L_add_credit_limit = L_credit_limitTxt.getText();
            String L_add_grace = L_graceTxt.getText();

            int card_Id = Integer.parseInt(L_add_card_id);
            double credit_limit = Integer.parseInt(L_add_credit_limit);
            int add_grace = Integer.parseInt(L_add_grace);
            boolean isadd = false;
            for(BankCard bank_card: gui_Card){
                if(bank_card instanceof CreditCard){
                    CreditCard limit = (CreditCard)bank_card;
                    if(limit.getcardId() == card_Id){
                        limit.setcreditLimit(credit_limit, add_grace);
                        if(limit.getisGranted() == true){
                            JOptionPane.showMessageDialog(null, "Credit Limit Added
Successfully!");
                            isadd = true;
                            break;
                        }
                        else{
```

```
                        isadd = false;

                    }

                }

            }

        }

    }catch(NumberFormatException ex){

        JOptionPane.showMessageDialog(null, "No card has been added yet!",
"error", JOptionPane.ERROR_MESSAGE);

    }

}



// Cancel For Credit Limit

if(e.getSource() == L_clearBtn){

    L_card_idTxt.setText("");

    L_credit_limitTxt.setText("");

    L_graceTxt.setText("");

}



// For Cancel Credit Card

if(e.getSource() == cancel_clearBtn){

    try{
        String cancel_add_cancelbtn = cancel_card_idTxt.getText();


        int cancel_credit = Integer.parseInt(cancel_add_cancelbtn);

        boolean iscancel = false;

        for(BankCard bank_card: gui_Card){

            if(bank_card instanceof CreditCard){

                CreditCard limit = (CreditCard)bank_card;

                if(limit.getcardId() == cancel_credit){

                    limit.cancelCreditCard();

                    JOptionPane.showMessageDialog(null, "Card Canceled!");

                    iscancel = true;

                }

            }else{

                iscancel = false;

            }

        }

        if(iscancel == false){

            JOptionPane.showMessageDialog(null, "Card Id has not been added yet");

        }

    }catch(NumberFormatException ex){
```

```
                JOptionPane.showMessageDialog(null, "Please Enter Valid Card Id", "error",
JOptionPane.ERROR_MESSAGE);

            }

        }

    }



    public static void main(String[] args){

        BankGUI obj = new BankGUI();

    }

}
```