



 slington college
(इस्लिङ्टन कलेज)

CS4001NI Programming

30% Individual Coursework

2022-23 Autumn

Student Name: Himanshu Joshi

London Met ID: 22068759

College ID: NP01CP4A220266

Group: C11

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.

Contents

Section 1: Introduction.....	4
1.1 Blue J	4
1.2 Java Development Kit (JDK)	4
1.3 MS Word	4
1.4 Draw.io	4
1.5 Moqups:.....	5
Section 2: Class Diagram.....	5
Section 3: Pseudocode	10
Section 4: Method Description of all JButtons	20
Section 5: Testing	22
5.1: Testing 1	22
Testing 2: Adding objects of Debit Card and Credit Card, withdrawing amount from debit card, setting the credit limit, and removing the credit card	23
Testing 3: Testing appropriate dialogue boxes when unsuitable values are entered.	30
Section 6: Error detection and correction	35
6.1: Syntax Error:.....	35
6.2: Logical Error:	36
6.3: Semantic Error:.....	36
Section 7: Conclusion.....	38
Section 8: Bibliography.....	39
Section 9: Appendix	40

Table of figures:

Figure 1: Testing of compilation from cmd	23
Figure 2: Debit Card add	25
Figure 3: Display after adding debit card.....	25
Figure 4: Credit Card add.....	26
Figure 5: Display after adding credit card.....	26
Figure 6: Withdraw from debit card	27
Figure 7: Withdraw method is run	27
Figure 8: Set Credit Limit.....	28
Figure 9: Print of set credit limit.....	28
Figure 10: Cancel Credit Card.....	29
Figure 11: Cancel credit card method is run.....	29
Figure 12: Display method after cancelling credit card.....	29
Figure 13: Adding string values in debit card	31
Figure 14: Invalid Card ID while withdrawing	31
Figure 15: Invalid PIN Number while withdrawing	32
Figure 16: Adding string values in credit card	32
Figure 17: Invalid Card ID while setting Credit Limit.....	33
Figure 18: Adding string values while setting the credit limit	33
Figure 19: Withdraw without adding to Debit Card	34
Figure 20: Set Credit Limit without adding to Credit Card	34
Figure 21: Giving insufficient credit limit.....	35
Figure 22: Syntax error.....	35
Figure 23: Syntax error correction	36
Figure 24: Logical error	36
Figure 25: Logical error correction	36
Figure 26: Semantic error.....	37
Figure 27: Semantic error correction	37

Section 1: Introduction

This is the coursework which is related to the java programming language. Here, we have three classes called **BankCard**, **DebitCard** and **CreditCard** which we already did in our first semester. Also, here we have a new class **BankGUI** which is basically the GUI version of Bank Card, Debit Card and Credit Card respectively. Here, we do the project of first semester practically by interacting with GUI and providing the values. I took the help of BlueJ to complete this project. Some of the tools used for the project are given below.

1.1 Blue J

Blue J is referred as a platform which is basically used for doing the Java related projects. It was a free Java environment which was started in 1999. It is a platform which also requires JDK (Java Development Kit). Not only it allows you to interact with objects, but also the tasks can be performed very conveniently (Harleenk, 2022).

1.2 Java Development Kit (JDK)

Java Development Kit is the is an environment for software development which provides the libraries and tools for the development of Java software or application. It includes the components like Java Runtime Environment (JDK), compiler and some other development tools (Xie, 2023).

1.3 MS Word

Microsoft Word is defined as a word processing software which is published by Microsoft. It allows us to write quality documents, reports, letters, etc. We can also format and edit many things like alphabets, headings, etc. It was developed in 1983 (Hope, 2021).

1.4 Draw.io

Draw.io is a software or tool designed by Seibert Media, which is useful for creating charts and diagrams. We can easily make flowcharts from here. Not only we can use the automatic layout, but also, we can make our custom layout here (Hope, 2020).

1.5 Moqups:

Moqups is a platform designed for making the outer or outline frame for any website or any graphical user interface. It has many design features and diagrams. It is used by professionals, many organizations that work in complicated projects. It provides greater impact of the information that you generate (Usmani, 2022).

Section 2: Class Diagram

Bank Card
<ul style="list-style-type: none">- cardId: int- clientName: String- issuerBank: String- bankAccount: String- balanceAmount: int
<ul style="list-style-type: none">+<<constructor>> BankCard(balanceAmount: int, cardId: int, bankAccount: String, issuerBank: String)+ getCardId(): int+ getClientName(): String+ getIssuerBank(): String+ getBankAccount(): String+ getBalanceAmount(): int+ setClientName(clientName: String): void+ setBalanceAmount(balanceAmount: int): void+ display(): void

Table 1: Bank Card class diagram

Debit Card
<ul style="list-style-type: none"> - pinNo: int - withdrawAmount: int - dateofWithdraw: String - hasWithdrawn: boolean
<p>+<<constructor>> DebitCard(balanceAmount: int, cardId: int, bankAccount: String, issuerBank: String, pinNo: int)</p> <p>+ getPinNo(): int</p> <p>+ getWithdraw(): int</p> <p>+ getDateofWithdraw(): String</p> <p>+ getHasWithdrawn(): Boolean</p> <p>+ setWithdraw(withdrawAmount: int): void</p> <p>+ withdraw(withdrawAmount: int, dateofWithdraw: String, pinNo: int): void</p> <p>+ display(): void</p>

Table 2: Debit Card class diagram

Credit Card
<ul style="list-style-type: none"> - CVCnumber: int - CreditLimit: double - IntrestRate: double - ExpirationDate: String - GracePeriod: int - isGranted: boolean
<p>+<<constructor>> CreditCard(cardId: int,clientName: String, issuerBank: String, bankAccount: String, balanceAmount: int, CVCnumber: int, IntrestRate: double,ExpirationDate: String)</p> <ul style="list-style-type: none"> + getCVCnumber(): int + getCreditLimit(): double + getIntrestRate(): double + getExpirationDate(): String + getGracePeriod(): int + getisGranted(): boolean + setCreditLimit(CreditLimit: double, GracePeriod: int): void + cancelCreditCard(): void + display(): void

Table 3: Credit Card class diagram

BankGUI
<ul style="list-style-type: none"> - al: ArrayList - jf1, jf2, jf3, jf4, jf5, jf6: JFrame - tfcdid, tfclntname, tfissbank, tfbnkacc, tfbalamnt, tfpin, tfcdtcdid, tfcdtclntname, tfcdtbnkacc, tfcdtissbank, tfcdtbalamnt, tfcdtcvcno, tfcdtint, tfcdtexp, tfwithcdid, tfwithamnt, tfwithpin, tfsetcdid, tfsetcdlmt, tfsetgrper, tfcnlcdid: JTextField - bnkcrdhead, dbtcrdhead, cdid, clntname, issbank, bnkacc, balamnt, pin, cdtrdhead, cdtcdid, cdtclntname, cdtbnkacc, cdtissbank, cdtbalamnt, cdtcvcno, cdtint, cdtexp, withhead, withcdid, withamnt, withdow, withpin: JLabel - dbtcrdgo, cdtrdgo, dbtadd, dbtwith, dbtdis, dbtclr, cdtadd, cdtnl, cdtlmt, cdt dis, cdtclr, withdraw, withdis, withclr, setcredit, setdis, setclr, cnlcdtcard, cnlcdtclr: JButton - datewith, withexpdate: JComboBox
<p>+<<constructor>>BankGUI()</p> <p>+actionPerformed(ActionEvent ae): void</p> <p>+main(String[]args): void</p>

Table 4: BankGUI class diagram

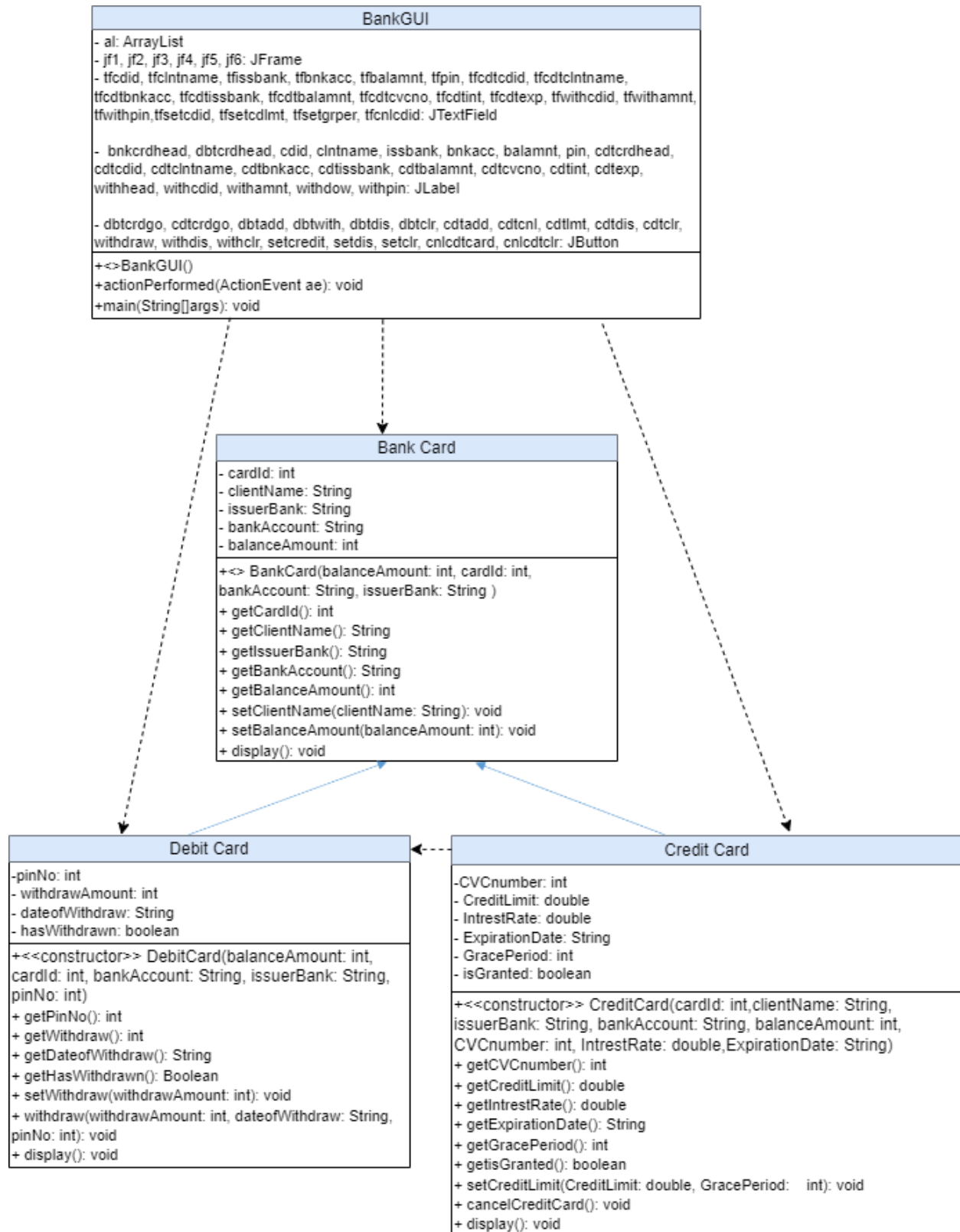


Table 5: Combined class diagram

Section 3: Pseudocode

IMPORT all from javax.swing

IMPORT Font from java.awt

IMPORT Color from java.awt

IMPORT all from java.awt.event

IMPORT ArrayList from java.util

CREATE a class called BankGUI that implements ActionListener

DO

DECLARE the ArrayList of BankCard as al

DECLARE JFrame variables as instance by adding private access modifier

DECLARE String variables as instance by adding private access modifier

DECLARE int variables as instance by adding private access modifier

DECLARE double variables as instance by adding private access modifier

DECLARE JLabel variables as instance by adding private access modifier

DECLARE JTextField variables as instance by adding private access modifier

DECLARE JButton variables as instance by adding private access modifier

DECLARE JComboBox variables as instance by adding private access modifier

END DO

CREATE a constructor as BankGUI() with no parameters

DO

INITIALIZE the JFrame variable

INITIALIZE the JLabel variables

INITIALIZE the JTextField variables

INITIALIZE the JButton variables

SET size and coordinates of JLabel variables

SET size and coordinates of JTextField variables

SET size and coordinates of JButton variables

ADD JTextField variables in JFrame

ADD JLabel variables in JFrame

ADD JButton variables in JFrame

ADD action listener to JButton variables

CREATE an object of color with a reference name as blue

SET the background color of JFrame as blue

SET the layout of JFrame as null

SET the visibility of JFrame as true

SET the resizable of JFrame as false

SET size and coordinates of JFrame

SET the default close operation of JFrame as exit on close

END DO

CREATE a method called actionPerformed() with parameter as ActionEvent ae with no return type

DO

IF the ae source is equals to dbtcrdgo or cdtdcrdgo or dbtwith or cdtlmt or cdtcnl

INITIALIZE the JFrame variable

INITIALIZE the JLabel variables

INITIALIZE theJTextField variables

INITIALIZE theJButton variables

CREATE an array of dowith of String data type

```

CREATE an array of expdate String data ype
INITIALIZE the JComboBox variable by adding dowith variable
INITIALIZE the JComboBox variable by adding exp variable
ADD action listener to JButton variables
ADD the JTextField variables in JFrame
ADD the JLabel variables in JFrame
ADD the JButton variables in JFrame
ADD the JComboBox variable in JFrame
CREATE an object of color with a reference name as blue
SET the background of JFrame as blue
SET the visibility of JFrame as true
SET the layout of JFrame as null
SET the size and coordinates of JFrame
SET the resizable of JFrame as false
SET the default close operation of JFrame as exit on close
END IF
IF the ae source is equals to dbtclr or cdtclr or withclr or setclr
    SET the TextField variables as empty text
END IF
IF source of ae is equals to dbtadd
    IF the JTextField of tfcdid or tfclntname, tfbnkacc or tfissbank or tfbalamnt
    or tfpin is empty
        SHOW an alert message as Empty field found. Please fill the fields
        properly.
    END IF
ELSE
    TRY
        ASSIGN the value of dcdid

```

```

ASSIGN the value of dpin
ASSIGN the value of dbalamnt
ASSIGN the value of clntname
ASSIGN the value of issbank
ASSIGN the value of bnkacc
ASSIGN the value of cbankacc
ASSIGN the value of cexpdate
ASSIGN the value of added to false
IF al is empty
    CREATE object of CreditCard class as ob with
parameters dbalamnt,dcdid,dbnkacc,dissbank,dclntname,dpin
    ADD that ob to al
    SHOW the information message as Data is added to
the Debit Card system.
END IF
ELSE
    FOR card in al
        IF card instance of DebitCard
            SET dbtcd to (DebitCard) card
            IF cardid already exists
                SHOW alert message as This
Card ID already exists. Please re-enter your card id.
            END IF
            ELSE
                CREATE an object of
DebitCard class as ob with parameters
dbalamnt,dcdid,dbnkacc,dissbank,dclntname,dpin
                ADD the object ob to al
                SHOW information
message as Data is added to the Debit Card system.
            END IF
        END FOR
    ASSIGN added to false

```

```

ELSE
    CREATE an object of DebitCard class
    dbalamnt, dcdid, dbnkacc, dissbank, dcIntname, dpin as parameters and object as
    ob
    ADD ob to al
    SHOW information
    message as Data is added to the Debit Card system.
    ASSIGN added to false
END IF
END FOR
END TRY
CATCH
    SHOW alert message as Number format exception
    detected. Cannot convert string into int.
END IF
IF the ae source is equals to cdtadd
    IF JTextField variables of cdtadd is empty
        SHOW alert message as Empty field found. Please enter the fields
        properly.
    END IF
ELSE
    TRY
        ASSIGN the value of ccdid
        ASSIGN the value of ccvco
        ASSIGN the value of cbalamnt
        ASSIGN the value of cint
        ASSIGN the value of ccIntname
        ASSIGN the value of cissbank
        ASSIGN the value of cbnkacc
        ASSIGN the value of cexp
        IF al is empty

```

CREATE object of creditcard class as ob with parameters ccdid, cclntname, cissbank, cbnkacc, cbalamnt ,ccvcno ,cexp

ADD the object ob1 to al

SHOW information message as Data is added to the Credit Card system.

END IF

ELSE

FOR card in al

IF card instance of CreditCard

SET cdtcard to (CreditCard) card

IF the cardid already exists

SHOW alert message as
This Card ID already exists. Please re-enter it.

END IF

ELSE

CREATE object of creditcard class as ob with parameters ccdid, cclntname, cissbank, cbnkacc, cbalamnt ,ccvcno ,cexp

ADD the object ob1 to al

SHOW information message as
Data is added to the Credit Card system.

END IF

END FOR

END TRY

CATCH

SHOW alert message as Number format exception detected. Cannot convert String into int.

END IF

IF the ae source is equals to withdraw

IF JTextField variable of withdraw is empty

SHOW alert message Empty field found. Please fill the fields properly.

```

    END IF
    ELSE
        TRY
            ASSIGN the value of withcrdid
            ASSIGN the value of withpinno
            ASSIGN the value of withwithdrawamnt
            ASSIGN the value of withdowith
            IF al is empty
                SHOW alert message as There is no Debit Card.
Please enter your debit card properly.
            END IF
            ELSE
                FOR all in al
                    IF all instance of DebitCard
                        SET with to (DebitCard) all
                        IF crdid of ddtadd and and withdraw
don't match
                            SHOW alert message The
Card ID is invalid. Please check your Card ID
                            END IF
                        ELSE
                            IF the withpinno is valid
                                IF
withwithdrawamnt is more than balamnt
                                    SHOW alert
message as You don't have enough balance. Please check your balance amount.
                                END IF
                            ELSE
                                CALL the
withdraw() method of Debit Card class with parameters withwithdrawamnt, withdowith,
withpinno
                                    SHOW information message as Money
withdrawl is successful.

```



```

ELSE
    SHOW alert
message as "Invalid Pin Number entered.Please re-check it.
END IF
END FOR
CATCH
    SHOW alert message as Number Format Exception
detected. Cannot convert String into int.
END IF
IF the ae source is equals to setcredit
    CREATE an object of BankCard class as bcd with parameters dbalamnt,
dcdid, dbnkacc, dissbank, dclntname, dpin
    SET ccd to (CreditCard) bcd
    IF JTextField variables of setcredit is empty
        SHOW alert message Empty field found. Please fill the fields
properly.
    END IF
ELSE
    TRY
        ASSIGN the value of setcrdcdid
        ASSIGN the value of setcrgprper
        ASSIGN the value of setcrdclmt
        ASSIGN the value of added to false
        IF al is empty
            SHOW alert message There is no credit card
found. Please enter your credit card properly..
        END IF
    ELSE
        FOR each in al
            IF each instance of CreditCard

```

```

don't match                                IF ccrdid of setcredit and cdtadd
                                           SHOW alert message The
Card ID is invalid. Please check your Card ID.
                                           END IF
                                           ELSE
                                           IF setcdlmt is greater than
2.5 times of balamnt
                                           SHOW alert
message as the amount is insufficient for the credit limit.
                                           END IF
                                           ELSE
                                           CALL the
setCreditLimit() method of Credit Card class with parameter setcrdlmt,setcrgpr
                                           ASSIGN the value of
added to true
                                           SHOW information
message as The Credit Limit is sucessfully set.

                                           END IF
                                           END FOR
CATCH
                                           SHOW alert message as Number format exception
detected. Cannot convert String into int.

END IF
IF source of e is equals to cnlcdtcard
    IF JTextField variable of cnlcdtcard is empty
        SHOW alert message "Empty field found, please re-check it.
    END IF
ELSE
    TRY
        ASSIGN the value of cancrdid

```

```

ASSIGN added to false
FOR all in al
    IF all instanceof CreditCard
        IF cancrdid equals to ccdid
            SET ccd to (CreditCard) all
            CALL withdraw() method of Creditcard
class
        ASSIGN added to false
    END IF
    ELSE
        SHOW alert message as The Card ID is
invalid.
    END IF
    ELSE
        SHOW alert message There is no
any credit card detected.
CATCH
    SHOW alert message Number format exception detected.
Cannot convert string into int.
END IF
IF source of e is equals to debtdis or withdis
    FOR all in al
        IF all instaceof Debitcard
            SET dis to (DebitCard) all
            CALL display() method from Debitcard
        END IF
    END IF
IF source of e is equals to cdtdis or setdis
    FOR all in al
        IF all instaceof Creditcard

```

```

        SET dis to (CreditCard) all
        CALL display() method of Creditcard
    END IF
END IF
END DO

CREATE a method main() method with parameter String[]args with no return type
DO
    CALL the constructor BankGUI()
END DO

```

Section 4: Method Description of all JButtons

There are various buttons in BankGUI class. Each of the buttons has its own functions. The buttons in the BankGUI class are given below.

The button “**dbtcrdgo**” is put in the frame of Bank Card. This button is used to take user to the Debit Card GUI frame. When this button is pressed, we directly move on to the Debit Card GUI frame where we do the operations of the Debit Card GUI.

The button “**cdtcrdgo**” is put in the frame of Bank Card. This button is used to take user to the Credit Card GUI frame. When this button is pressed, we directly move on to the Credit Card GUI frame where we do the operations of the Credit Card GUI.

The button “**dbtadd**” is put in the Debit Card GUI. This button is used to add the data of the Debit Card in array list. In this button, we check first whether any field is empty or not. If any field is empty, then we give an error message to the user and ask them to fill the fields. Also, here we check whether the Bank Card is instance of Debit Card class or not. Not only this, but we have also checked the number format exception for the values. If the given value of Card ID is in int, then we show an error message as the detection of the Number format exception.

The button “**cdtadd**” is put in the Credit Card GUI. This button is used to add the data of the Credit Card in array list. In this button, we check first whether any field is empty or

not. If any field is empty, then we give an error message to the user and ask them to fill the fields. Also, here we check whether the Bank Card is instance of Credit Card class or not. Not only this, but we have also checked the number format exception for the values. If the given value of Card ID, CVC number is in int, then we show an error message as the detection of the Number format exception.

The button “**dbtwith**” is put in the Debit Card GUI. This button is used to withdraw from the Debit Card. When we press this button, new frame is opened where it asks for some values as Card ID, PIN number, Withdrawal amount, Date of Withdrawal. If any field is empty, then it shows an error message to the user as empty text field found. It also checks for the Card ID and PIN number of the Debit Card. If they are invalid, it shows an error message to the user. After that, it checks for the balance amount and withdrawal amount. If sufficient balance amount is present, then it withdraws from the Debit Card. Also, it calls the withdraw method from the Debit Card class and prints the required values.

The button “**dbtdis**” is put in Debit Card GUI. This button is basically used to display the data of the Debit Card. When this button is pressed, firstly it checks for any empty field. If any field is empty, then it shows an error message to the user. We can only display when we add the data to the Debit Card. If the fields are filled, it firstly checks whether the Bank Card is an instance of Debit Card class or not. If it is instance of Debit Card, then it runs its function by calling the display method of Debit Card class.

The button “**cdtdis**” is put in Credit Card GUI. This button is basically used to display the data of the Credit Card. When this button is pressed, firstly it checks for any empty field. If any field is empty, then it shows an error message to the user. We can only display when we add the data to the Credit Card. If the fields are filled, it firstly checks whether the Bank Card is an instance of Credit Card class or not. If it is instance of Credit Card, then it runs its function by calling the display method of Credit Card class.

The button “**cdtcnl**” is put in Credit Card GUI. This button is basically used to cancel the Credit Card and its data. When this button is pressed, firstly it checks whether any text field is empty or not. If any text field is empty, it shows an error message to the user. If the text field is filled, it checks whether the input Card id to be cancelled is equal to the Card ID that we added in the Credit Card. If the card id is invalid, it shows an error

message to the user. If it is correct, then it cancels the Card ID by calling the `cancelCreditCard()` method of the Credit Card.

The button “**setcredit**” is put in Credit Card GUI. This button is basically used to set the credit limit in Credit Card. When this button is pressed, it checks whether any text field is empty or not. If any text field is empty, it shows an error message to the user. If the text field is filled, it checks whether the card id is valid or not. After it checks whether the `setcreditlimit` is greater than 2.5 times the balance amount. If it is not valid, it shows an error message to the user otherwise it sets the credit limit by calling the `setCreditLimit()` method from Credit Card.

Section 5: Testing

5.1: Testing 1

Test No:	1
Objective:	To compile the program and run it from command prompt.
Action:	After going to the current coursework file location, <code>cmd</code> is typed on the search bar of the file location and we get the interface of the command prompt. Now <code>javac BankGUI.java</code> is typed to compile the program and again <code>java BankGUI</code> is typed to run the program using command prompt.
Expected Result	The program would be compiled with no errors and the Bank Card GUI would be opened on the screen.
Actual Result	The program was compiled, and the GUI was opened on the screen.
Conclusion	The test was successful.

Table 6: Testing of compilation from cmd

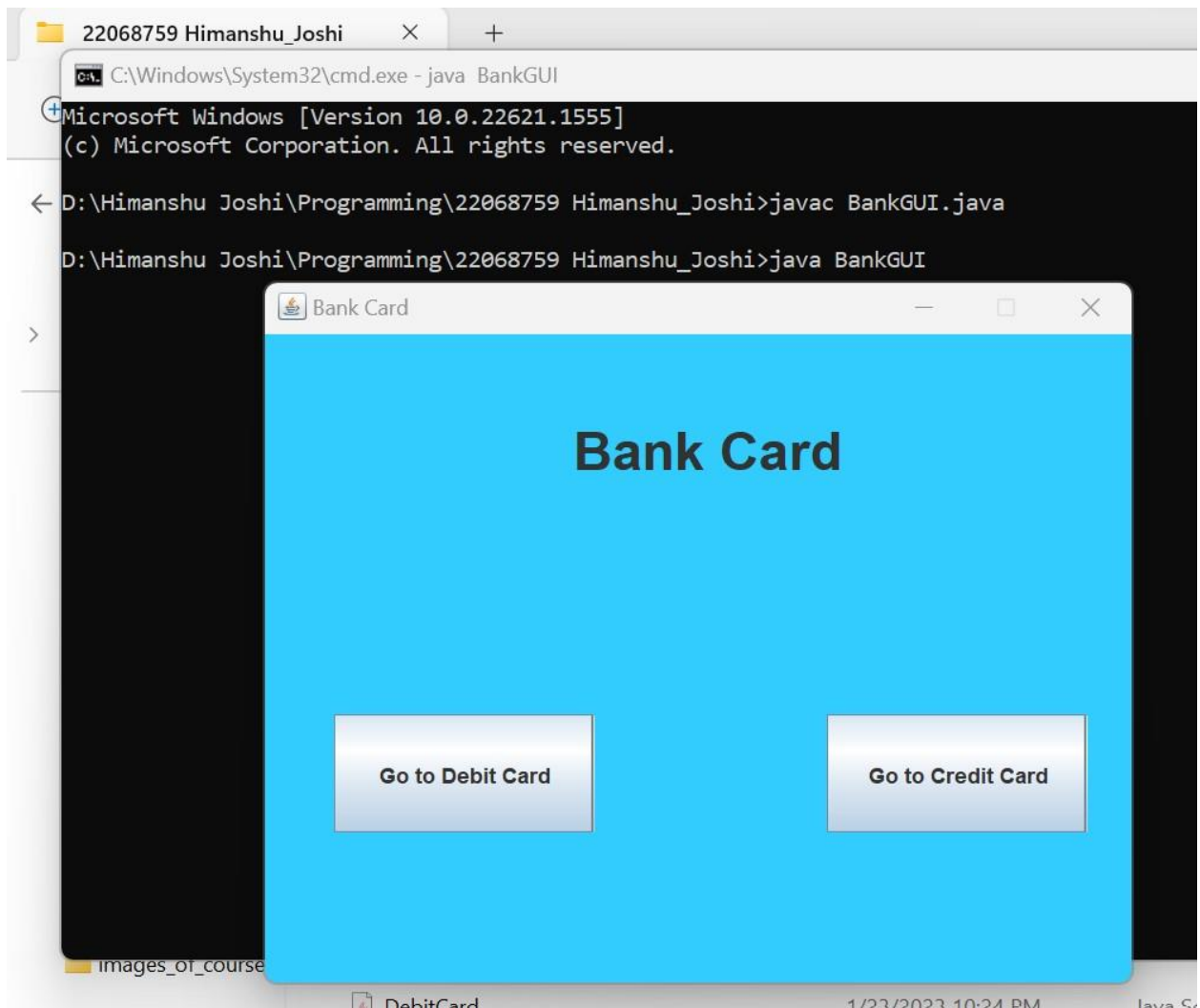


Figure 1: Testing of compilation from cmd

Testing 2: Adding objects of Debit Card and Credit Card, withdrawing amount from debit card, setting the credit limit, and removing the credit card

Test No:	2
Objective:	To add objects of debit card and credit card, withdraw amount from debit card, set the credit limit and remove the credit card
Action:	<ul style="list-style-type: none"> ➔ The text fields of Debit Card GUI are filled. ➔ Display button is clicked to observe the created object. ➔ The text fields of withdrawal are filled after clicking the “Withdraw from Debit Card” button. ➔ The withdraw amount and date of withdrawal is check after withdrawing from Debit Card. ➔ The text fields of Credit Card GUI are filled. ➔ The display button is clicked to check the data. ➔ The “Set Credit Limit” is clicked, and different fields of set credit limit is filled. ➔ The display button is clicked to check the data. ➔ The Card Id is entered after clicking the “Cancel Credit Card” button. ➔ The display button is clicked to check the data.
Expected Result	The Debit Card and Credit Card would be added, and the data would be displayed. Similarly, the credit limit would be set, and the credit card of a particular ID would be cancelled. After cancelling the credit card, some attributes like CVC number would be set to zero.
Actual Result	The Debit Card and Credit Card was added, and the data were displayed. The credit limit was set, and the data were displayed. The credit card was cancelled and the attributes like CVC number were set to zero.
Conclusion	The test was successful.

Table 7: Testing Debit and Credit Card class

Debit Card

Card ID: 1

Client Name: Himanshu

Bank Account: 12345

Issuer Bank: Nabil

Balance Amount: 1000

PIN Number: 5

Message: Data is added to the Debit Card system. OK

Add Debit Card Withdraw from Debit Card

Display Clear

Figure 2: Debit Card add

```
BlueJ: Terminal Window - 22068759 Himanshu_Joshi
Options
The client's name is: Himanshu
The card id of client is: 1
The issuer bank is: Nabil
The bank account of the client is: 12345
The balance amount is: 1000
The PIN number is: 5
The widthdrawal amount is: 0
The date of withdrawal is: null
The transaction has not been carried yet
The balance amount is: 1000
```

Figure 3: Display after adding debit card

Credit Card

Card ID: 10 CVC Number: 12345

Client Name: Ramesh Intrest Rate: 2.2

Bank Account: 67890 Expiration Date: 2024-09-09

Issuer Bank: NIC Asia

Balance Amount: 5000

Message: Data is added to the Credit Card system

Add Credit Card Cancel Credit Card

Clear

Display Set Credit Limit

Figure 4: Credit Card add

```
Options
The CVC number is: 12345
The Intrest rate is: 2.2
The expiration date is: 2024-09-09
The access granted is: false
```

Figure 5: Display after adding credit card

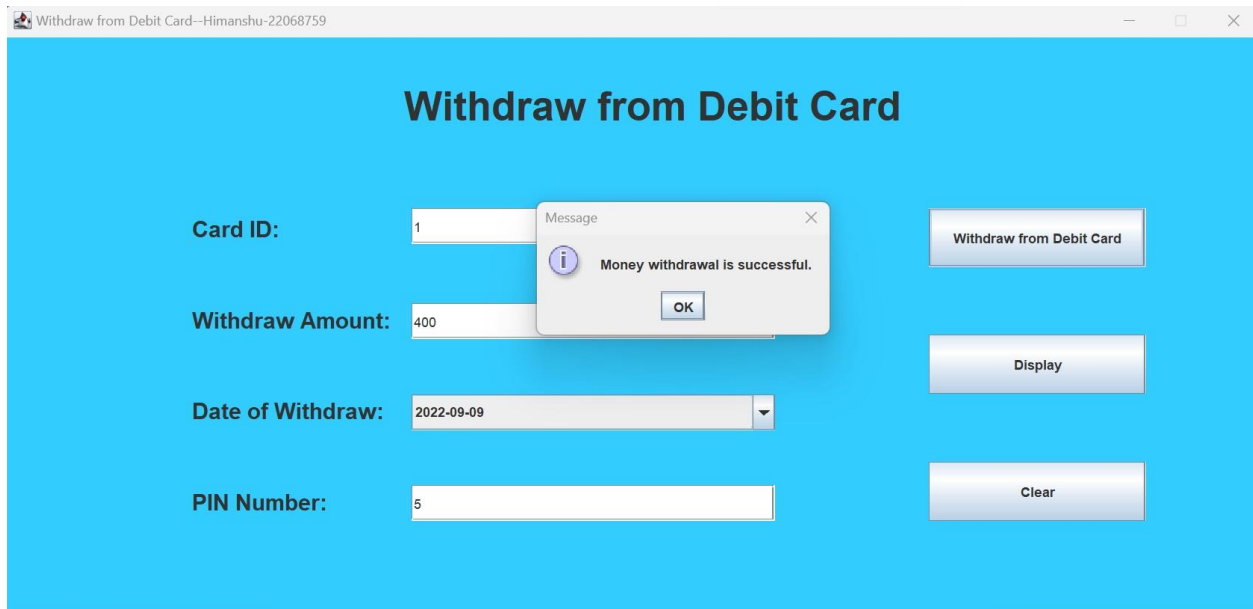


Figure 6: Withdraw from debit card

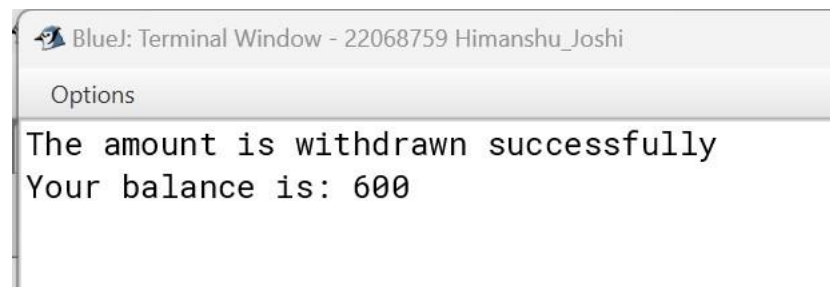


Figure 7: Withdraw method is run

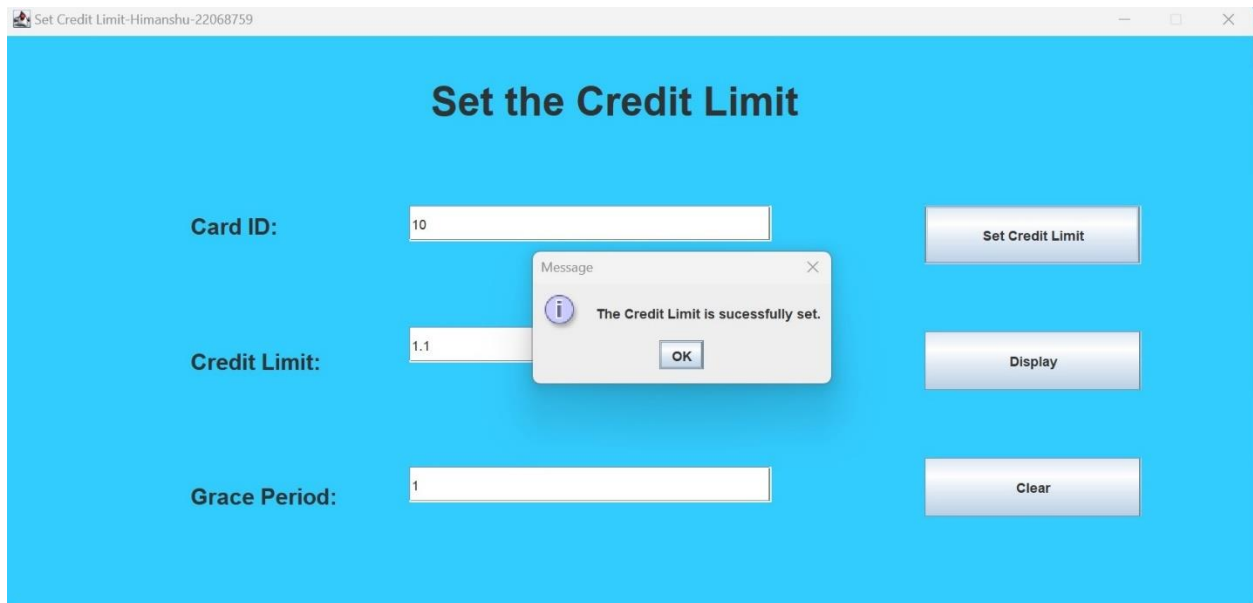


Figure 8: Set Credit Limit

```
BlueJ: Terminal Window - 22068759 Himanshu_Joshi
Options
The client's name is: Ramesh
The card id of client is: 10
The issuer bank is: NIC Asia
The bank account of the client is: 67890
The balance amount is: 5000
The Credit Limit is: 1.1
The Grance Period is: 1
The CVC number is: 12345
The Intrest rate is: 2.2
The expiration date is: 2024-09-09
The access granted is: true
```

Figure 9: Print of set credit limit

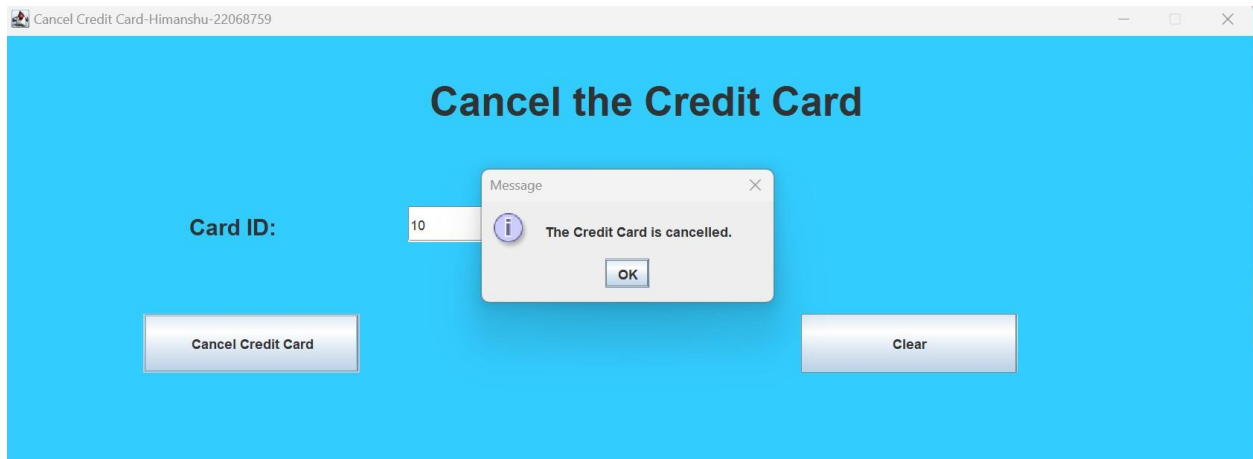


Figure 10: Cancel Credit Card



Figure 11: Cancel credit card method is run

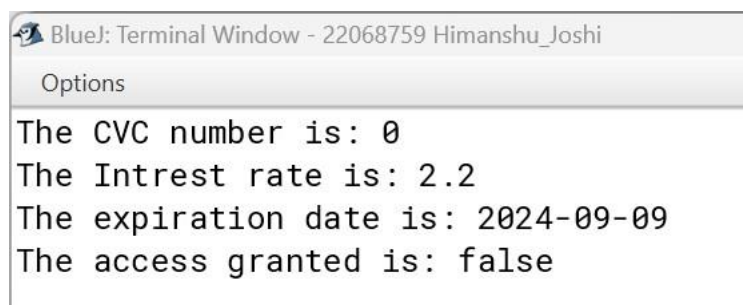


Figure 12: Display method after cancelling credit card

Testing 3: Testing appropriate dialogue boxes when unsuitable values are entered

Test No:	3
Objective:	To test appropriate dialogue boxes when unsuitable values are entered.
Action:	<ul style="list-style-type: none"> ➔ The text field of Card ID is set to “Card”, that of Balance Amount is set to “Balance”, and that of PIN number is set to “PIN”. ➔ Invalid Card ID entered while running the withdraw button from Debit Card. ➔ While running withdraw button, invalid PIN number is entered, also when debit card is not added. ➔ While adding Credit Card, the text field of Card ID is set to “Hello”, that of Balance Amount is set to “int”, that of CVC number is set to “int” and that of Interest Rate is set to “int”. ➔ While running the “Set Credit Limit” button, invalid Card ID is entered. ➔ While running the “Set Credit Limit” button, the text field of Card ID is set to “string”. ➔ While running the “Set Credit Limit” button, the value in text field of Credit Limit is set to more than 2.5 times the Balance Amount, also when credit card is not added.
Expected Result	The appropriate error messages of respective errors would be shown on the screen.
Actual Result	The appropriate error messages of respective error were shown on the screen.
Conclusion	The test was successful.

Table 8: Testing of error messages

The image shows a Java Swing window titled "Debit Card-Himanshu-22068759". The window has a light blue background and a title bar with standard Windows controls. The main content area is titled "Debit Card" in bold black text. Below the title, there are six labeled text input fields: "Card ID:" (containing "Card"), "Client Name:" (containing "Himanshu"), "Bank Account:" (containing "12345"), "Issuer Bank:" (containing "Nabil"), "Balance Amount:" (containing "Balance"), and "PIN Number:" (containing "PIN"). At the bottom of the window, there are four buttons: "Add Debit Card", "Withdraw from Debit Card", "Display", and "Clear". An "Alert" dialog box is overlaid on the "Bank Account:" field, displaying a red "X" icon and the message "Number format exception detected. Cannot convert string into int." with an "OK" button.

Figure 13: Adding string values in debit card

The image shows a Java Swing window titled "Withdraw from Debit Card--Himanshu-22068759". The window has a light blue background and a title bar with standard Windows controls. The main content area is titled "Withdraw from Debit Card" in bold black text. Below the title, there are four labeled input fields: "Card ID:" (containing "2"), "Withdraw Amount:" (containing "500"), "Date of Withdraw:" (a date picker showing "2022-09-09"), and "PIN Number:" (containing "5"). To the right of these fields are three buttons: "Withdraw from Debit Card", "Display", and "Clear". An "Alert" dialog box is overlaid on the "Card ID:" field, displaying a red "X" icon and the message "The Card ID is invalid. Please check your Card ID ." with an "OK" button.

Figure 14: Invalid Card ID while withdrawing

Withdraw from Debit Card--Himanshu-22068759

Withdraw from Debit Card

Card ID:

Withdraw Amount:

Date of Withdraw:

PIN Number:

Alert

Invalid Pin Number entered.Please re-check it.

Figure 15: Invalid PIN Number while withdrawing

Credit Card--Himanshu-22068759

Credit Card

Card ID:

CVC Number:

Client Name:

Intrest Rate:

Bank Account:

Expiration Date:

Issuer Bank:

Balance Amount:

Alert

Number format exception detected. Cannot convert string into int.

Figure 16: Adding string values in credit card

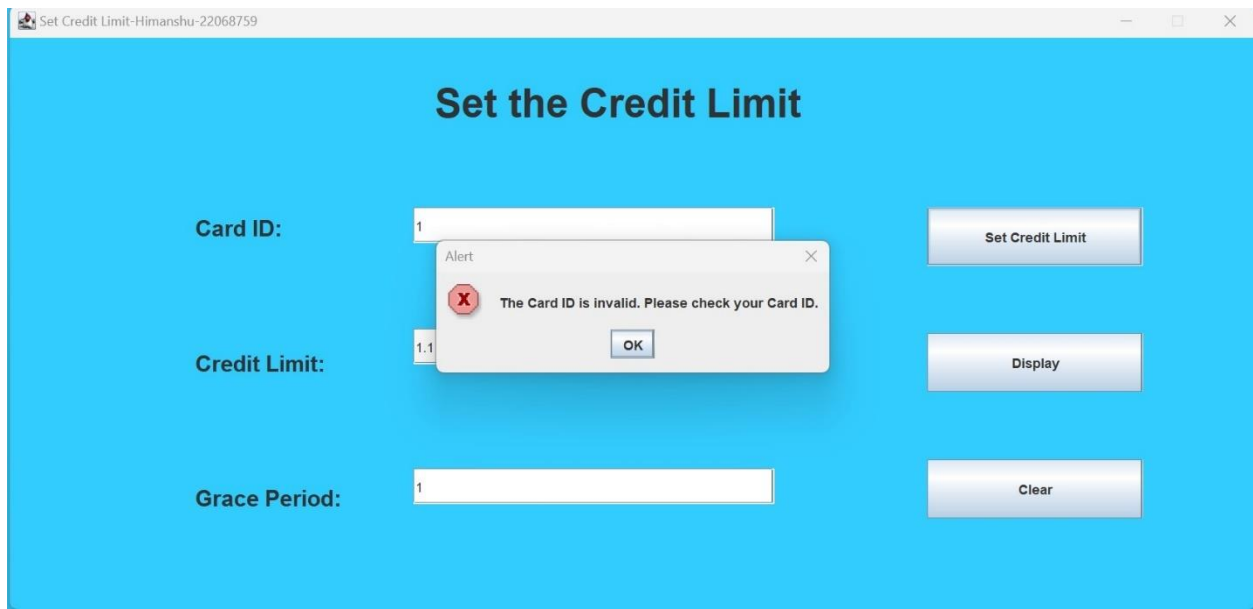


Figure 17: Invalid Card ID while setting Credit Limit

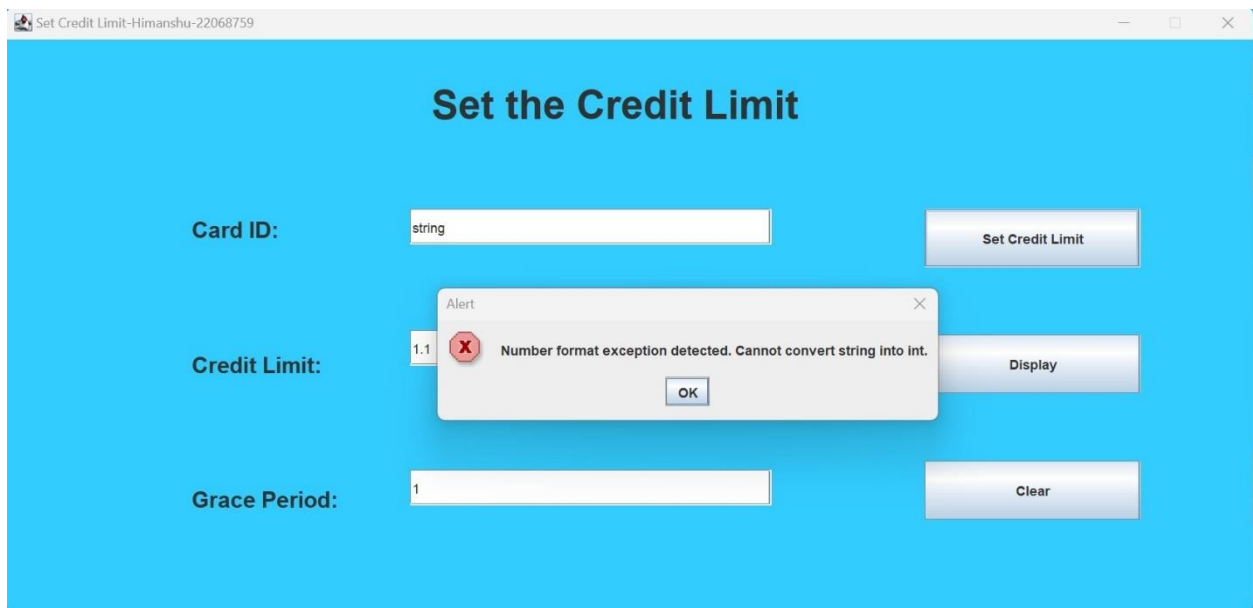


Figure 18: Adding string values while setting the credit limit

Withdraw from Debit Card-Himanshu-22068759

Withdraw from Debit Card


Card ID:

Withdraw Amount:

Date of Withdraw:

PIN Number:

Alert

 There is no Debit Card. Please enter your debit card properly.

OK

Figure 19: Withdraw without adding to Debit Card

Set Credit Limit-Himanshu-22068759


Set the Credit Limit

Card ID:

Credit Limit:

Grace Period:

Alert

 There is no Credit Card. Please enter your credit card properly.

OK

Figure 20: Set Credit Limit without adding to Credit Card

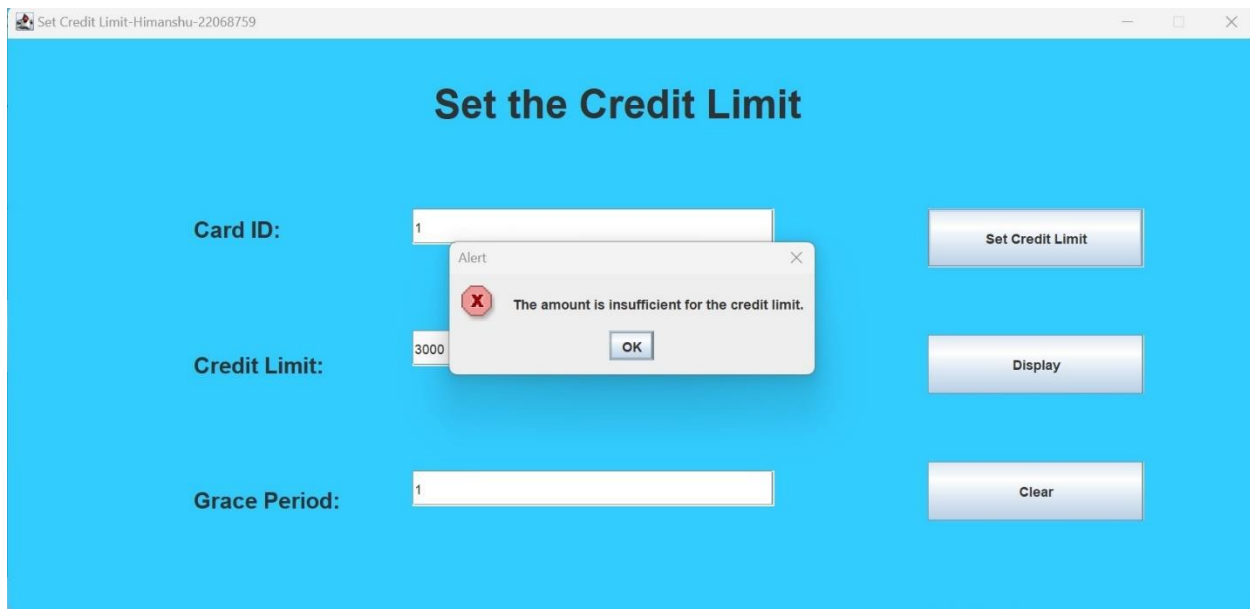


Figure 21: Giving insufficient credit limit

Section 6: Error detection and correction

6.1: Syntax Error:

A syntax error is defined as the error which is done by the programmers when they don't follow the rule of writing a line of code. Syntax error is occurred when the line of code written is in improper format. Since syntax is also known as the grammar of programming, it is also known as the grammatical error done by the user by writing a line of code (Gaudet, 2023).

```
else
{
    DebitCard bcd=(DebitCard) all
    bcd.withdraw(withwithdrawamnt,withdowith,withpinno);
    JOptionPane.showMessageDialog(jf4,"Money withdrawal is successful.");
}
```

Figure 22: Syntax error

```

else
{
    DebitCard bcd=(DebitCard) all;
    bcd.withdraw(withdrawamnt,withdowith,withpinno);
    JOptionPane.showMessageDialog(jf4,"Money withdrawal is successful.");
}

```

Figure 23: Syntax error correction

6.2: Logical Error:

Logical error is defined as the error which is done by the human due to their carelessness. The user does not get any syntax error but does not get the output that the user might have desired for. The output gives an unexpected behavior. There are different perspectives of logical error. For example, giving wrong logic inside any loop, etc. Due to logical error the program might also crash while running (Hope, 2018).

```

if(setcrdid==each.getCardId())
{
    JOptionPane.showMessageDialog(jf5,"The Card ID is invalid.
    break;
}
else
{

```

Figure 24: Logical error

```

if(setcrdid!=each.getCardId())
{
    JOptionPane.showMessageDialog(jf5,"The Card ID is invalid.
    break;
}

```

Figure 25: Logical error correction

6.3: Semantic Error:

Semantic error is the error in programming that is caused due to the error in the semantics. The program compiles without giving any compilation error but the user does not get the output. In other words, the program might crash as well. Many exceptions are also observed in the semantic error because there is also no sense of putting string value to an integer value (Narang, 2022).

```
BlueJ: Terminal Window - 22068759 Himanshu_Joshi
Options
Can only enter input while your program is running
java.lang.NullPointerException: Cannot invoke "javax.swing.JButton.setBounds(int, int, int, int)" because "this.dbtcrdgo" is null
    at BankGUI.<init>(BankGUI.java:47)
    at BankGUI.main(BankGUI.java:826)
```

Figure 26: Semantic error

```
bnkcrdhead= new JLabel("Bank Card");
dbtcrdgo= new JButton("Go to Debit Card");
cdtcrdgo= new JButton("Go to Credit Card");

//Setting Bounds for JButtons for Bank Card
bnkcrdhead.setBounds(173, 43, 160, 42);
dbtcrdgo.setBounds(39, 213, 146, 66);
cdtcrdgo.setBounds(315, 213, 146, 66);
```

Figure 27: Semantic error correction

Section 7: Conclusion

Finally, after the completion of this project, I reached to a conclusion. This is a project related to the Java programming language. Compared to the project of first semester, this project was comparatively tougher to do as we had to implement different things besides the things of the project of first semester. This project work basically teaches us about the GUI (Graphical User Interface). We had to input different packages to run the program such as event, swing, etc.

From this project I basically learned about the making of Graphical User Interface by using Java programming language. There is also another method of making GUI, but the best way was to make it using programming language. For the implementation of this coursework, I used different tools such as moqups, MS Word, Blue J, etc. We implemented this coursework by doing it in constructor and later we called the constructor in the main method of the class. The class diagram of this coursework teaches us a lot of things. The class diagram describes the relation between the four classes. We created a method called actionPerformed() which basically does the operations when we click on any GUI buttons. Different buttons performed different operations. There is also the concept of upcasting and downcasting in this project, which is also very important concept. I learned that without doing upcasting we cannot implement the downcasting concept. Different text fields and different labels were stored in different variables. Without adding buttons to action listener, we cannot perform the actions done by those buttons. So, these were the concepts that were implemented during this project.

However, this project was not an easy task to do. I faced a lot of errors and difficulties in this project. The GUI was not being implemented at first and also the buttons were not doing their operations properly. I faced different errors such as logical errors, syntax errors, etc. Also, I faced a lot of exception due to which my program was being crashed. However, I was successful to overcome those difficulties. For that, I took different references. I visited different websites and did a lot of research for it. Not only that my tutors also helped me a lot in doing this project as they cleared out all my confusions. Finally, I was successful in doing this coursework.

Section 8: Bibliography

Gaudet, H., 2023. *what-is-a-syntax-error*. [Online]

Available at: <https://www.easytechjunkie.com/what-is-a-syntax-error.htm>

[Accessed 7 May 2023].

Harleenk, 2022. *introduction-of-bluej*. [Online]

Available at: <https://www.geeksforgeeks.org/introduction-of-bluej/>

[Accessed 5 May 2023].

Hope, C., 2018. *logierro*. [Online]

Available at: <https://www.computerhope.com/jargon/l/logierro.htm>

[Accessed 7 May 2023].

Hope, C., 2020. *draw.io*. [Online]

Available at: <https://www.computerhope.com/jargon/d/drawio.htm>

[Accessed 5 May 2023].

Hope, C., 2020. *draw.io*. [Online]

Available at: <https://www.computerhope.com/jargon/d/drawio.htm>

[Accessed 6 May 2023].

Hope, C., 2021. *microsoft-word*. [Online]

Available at: <https://www.computerhope.com/jargon/m/microsoft-word.htm>

[Accessed 5 May 2023].

Narang, P., 2022. *types-of-erros-in-c*. [Online]

Available at: <https://www.scaler.com/topics/c/types-of-errors-in-c/>

[Accessed 7 May 2023].

Usmani, F., 2022. *moqups-review*. [Online]

Available at: <https://pmstudycircle.com/moqups-review/>

[Accessed 7 May 2023].

Xie, A., 2023. *what-is-the-java-development-kit-jdk*. [Online]

Available at: <https://www.educative.io/answers/what-is-the-java-development-kit-jdk>

[Accessed 5 May 2023].

Section 9: Appendix

```
import javax.swing.*;

import java.awt.Font;

import java.awt.Color;

import java.awt.event.*;

import java.util.ArrayList;


public class BankGUI implements ActionListener

{

    private BankCard bcd;

    private DebitCard dcd;

    private CreditCard ccd;

    private boolean added, withdrawl;

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

    private int
dcdid,dpin,dbalamnt,ccdid,ccvno,cbalamnt,withcrdid,withpinno,withwithdrawamnt,setcr
cdid,setcrgprper,cancrcdid;

    private String
dclntname,dissbank,dbnkacc,cclntname,cissbank,cbnkacc,cexp,withdowith;

    private double cint,setcrclmt;

    private JFrame jf1, jf2, jf3, jf4, jf5, jf6;

    private JTextField tfcdid, tfclntname, tfissbank, tfbnkacc, tfbalamnt, tfpin,
```



```

        tfcdtcdid, tfcdtclntname, tfcdtbnkacc, tfcdtissbank, tfcdtbalamnt, tfcdtcvcno,
        tfcdtint, tfcdtexp,

        tfwithcdid, tfwithamnt, tfwithpin, tfsetcdid, tfsetcdlmt, tfsetgrper, tfcnlcdid;

    private JLabel bnkcrdhead, dbtcrdhead, cdid, clntname, issbank, bnkacc, balamnt,
    pin, cdtrdhead,

        cdtcdid, cdtclntname, cdtbnkacc, cdtissbank, cdtbalamnt, cdtcvcno, cdtint,
    cdtxep,

        withhead, withcdid, withamnt, withdow, withpin,

        sethead, setcdid, setcdlmt, setgrper, cnlcdthead, cnlcdid;

    private JComboBox<String> datewith, withexpdate;

    private JButton dbtcrdgo, cdtrdgo, dbtadd, dbtwith, dbtdis, dbtclr, cdtdadd, cdtncl,
    cdtmlt, cdtdis, cdtdclr,

        withdraw, withdis, withclr, setcredit, setdis, setclr, cnlcdtcard, cnlcdtdclr;

    public BankGUI()

    {

        jf1= new JFrame("Bank Card-Himanshu-22068759");

        bnkcrdhead= new JLabel("Bank Card");

        dbtcrdgo= new JButton("Go to Debit Card");

        cdtrdgo= new JButton("Go to Credit Card");


        bnkcrdhead.setBounds(173, 43, 160, 42);

        dbtcrdgo.setBounds(39, 213, 146, 66);

        cdtrdgo.setBounds(315, 213, 146, 66);

        jf1.add(bnkcrdhead);

```

```

jf1.add(dbtcrdgo);

jf1.add(cdtcrdgo);


dbtcrdgo.addActionListener(this);

cdtcrdgo.addActionListener(this);

bnkcrdhead.setFont(new Font("SansSerif",Font.BOLD,30));


Color blue=new Color(51, 204 ,255);


jf1.getContentPane().setBackground(blue);

jf1.setLayout(null);

jf1.setVisible(true);

jf1.setResizable(false);

jf1.setSize(500, 400);

jf1.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

}


@Override

public void actionPerformed(ActionEvent ae)

{

    if(ae.getSource()==dbtcrdgo)

    {

```

```
jf2= new JFrame("Debit Card-Himanshu-22068759");
```

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

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

```
clntname= new JLabel("Client Name:");
```

```
bnkacc= new JLabel("Bank Account:");
```

```
issbank= new JLabel("Issuer Bank:");
```

```
balamnt= new JLabel("Balance Amount:");
```

```
pin= new JLabel("PIN Number:");
```

```
tfcdid= new JTextField();
```

```
tfclntname= new JTextField();
```

```
tfbnkacc= new JTextField();
```

```
tfissbank= new JTextField();
```

```
tfbalamnt= new JTextField();
```

```
tfpin= new JTextField();
```

```
dbtadd= new JButton("Add Debit Card");
```

```
dbtwith= new JButton("Withdraw from Debit Card");
```

```
dbtdis= new JButton("Display");
```

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

```
dbtcrdhead.setBounds(367, 38, 190, 50);
```

```
cdid.setBounds(167, 156, 80, 28);  
clntname.setBounds(167, 237, 140, 28);  
bnkacc.setBounds(167, 318, 140, 28);  
issbank.setBounds(167, 398, 127, 28);  
balamnt.setBounds(167, 480, 170, 28);  
pin.setBounds(167, 561, 127, 28);  
tfcdid.setBounds(363, 152, 325, 32);  
tfclntname.setBounds(363, 237, 325, 32);  
tfbnkacc.setBounds(363, 318, 325, 32);  
tfissbank.setBounds(363, 399, 325, 32);  
tfbalamnt.setBounds(363, 480, 325, 32);  
tfpin.setBounds(363, 561, 325, 32);  
dbtadd.setBounds(147, 693, 185, 47);  
dbtwith.setBounds(650, 693, 185, 47);  
dbtdis.setBounds(147, 800, 185, 47);  
dbtclr.setBounds(650, 800, 185, 47);
```

```
jf2.add(dbtcrdhead);  
jf2.add(cdid);  
jf2.add(clntname);  
jf2.add(bnkacc);  
jf2.add(issbank);  
jf2.add(balamnt);
```

```
jf2.add(pin);  
jf2.add(tfcidid);  
jf2.add(tfcIntname);  
jf2.add(tfbnkacc);  
jf2.add(tfissbank);  
jf2.add(tfbalamnt);  
jf2.add(tfpin);  
jf2.add(dbtadd);  
jf2.add(dbtwith);  
jf2.add(dbtdis);  
jf2.add(dbtclr);
```

```
dbtadd.addActionListener(this);  
dbtwith.addActionListener(this);  
dbtdis.addActionListener(this);  
dbtclr.addActionListener(this);
```

```
dbtcrdhead.setFont(new Font("SansSerif",Font.BOLD,36));  
cdid.setFont(new Font("SansSerif",Font.BOLD,20));  
clntname.setFont(new Font("SansSerif",Font.BOLD,20));  
issbank.setFont(new Font("SansSerif",Font.BOLD,20));  
bnkacc.setFont(new Font("SansSerif",Font.BOLD,20));  
balamnt.setFont(new Font("SansSerif",Font.BOLD,20));
```

```

pin.setFont(new Font("SansSerif",Font.BOLD,20));

Color blue=new Color(51, 204 ,255);

jf2.getContentPane().setBackground(blue);


jf2.setLayout(null);

jf2.setVisible(true);

jf2.setResizable(false);

jf2.setSize(905, 912);

jf2.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
}
else if(ae.getSource()==dbtclr)
{
    tfcdid.setText("");
    tfclntname.setText("");
    tfbnkacc.setText("");
    tfissbank.setText("");
    tfbalamnt.setText("");
    tfpin.setText("");
}
else if(ae.getSource()==cdtcrdgo)
{

```

```
jf3= new JFrame("Credit Card-Himanshu-22068759");
```

```
cdtcrdhead= new JLabel("Credit Card");
```

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

```
cdtclntname= new JLabel("Client Name:");
```

```
cdtbnkacc= new JLabel("Bank Account:");
```

```
cdtissbank= new JLabel("Issuer Bank:");
```

```
cdtbalamt= new JLabel("Balance Amount:");
```

```
cdtccvno= new JLabel("CVC Number:");
```

```
cdtint= new JLabel("Interest Rate:");
```

```
cdtexp= new JLabel("Expiration Date:");
```

```
tfcdtcdid= new JTextField();
```

```
tfcdtclntname= new JTextField();
```

```
tfcdtbnkacc= new JTextField();
```

```
tfcdtissbank= new JTextField();
```

```
tfcdtbalamt= new JTextField();
```

```
tfcdtccvno= new JTextField();
```

```
tfcdtint= new JTextField();
```

```
String expdate[]={ "2024-09-09", "2025-10-10", "2028-12-12", "2023-12-10",  
"2030-10-13"};
```

```
withexpdate= new JComboBox<String>(expdate);
```

```
cdtadd= new JButton("Add Credit Card");
cdtcnl= new JButton("Cancel Credit Card");
cdtlmt= new JButton("Set Credit Limit");
cdtdis= new JButton("Display");
cdtclr= new JButton("Clear");

cdtcrdhead.setBounds(560, 34, 210, 50);
cdtcdid.setBounds(167, 156, 80, 28);
cdtclntname.setBounds(167, 237, 140, 28);
cdtbnkacc.setBounds(167, 318, 140, 28);
cdtissbank.setBounds(167, 398, 127, 28);
cdtbalamnt.setBounds(167, 480, 170, 28);
cdtcvcno.setBounds(749, 147, 130, 28);
cdtint.setBounds(749, 237, 130, 28);
cdtexp.setBounds(749, 327, 160, 28);
tfcdtcdid.setBounds(363, 152, 325, 32);
tfcdtclntname.setBounds(363, 237, 325, 32);
tfcdtbnkacc.setBounds(363, 318, 325, 32);
tfcdtissbank.setBounds(363, 399, 325, 32);
tfcdtbalamnt.setBounds(363, 480, 325, 32);
tfcdtcvcno.setBounds(915, 147, 325, 32);
tfcdtint.setBounds(915, 233, 325, 32);
withexpdate.setBounds(915, 319, 325, 32);
```



```
cdtadd.setBounds(94, 583, 218, 67);  
cdtcnl.setBounds(997, 583, 218, 67);  
cdtlmt.setBounds(1009, 831, 218, 67);  
cdtdis.setBounds(94, 831, 218, 67);  
cdtclr.setBounds(565, 724, 218, 67);
```

```
jf3.add(cdtcrdhead);  
jf3.add(cdtcdid);  
jf3.add(cdtclntname);  
jf3.add(cdtbnkacc);  
jf3.add(cdtissbank);  
jf3.add(cdtbalamnt);  
jf3.add(cdtvcno);  
jf3.add(cdtint);  
jf3.add(cdtexp);  
jf3.add(tfcdtcdid);  
jf3.add(tfcdtclntname);  
jf3.add(tfcdtbnkacc);  
jf3.add(tfcdtissbank);  
jf3.add(tfcdtbalamnt);  
jf3.add(tfcdtvcno);  
jf3.add(tfcdtint);  
jf3.add(withexpdate);
```

```
jf3.add(cdtadd);
```

```
jf3.add(cdtcnl);
```

```
jf3.add(cdtlmt);
```

```
jf3.add(cdtdis);
```

```
jf3.add(cdtclr);
```

```
cdtadd.addActionListener(this);
```

```
cdtcnl.addActionListener(this);
```

```
cdtlmt.addActionListener(this);
```

```
cdtdis.addActionListener(this);
```

```
cdtclr.addActionListener(this);
```

```
cdtcrdhead.setFont(new Font("SansSerif",Font.BOLD,36));
```

```
cdtcdid.setFont(new Font("SansSerif",Font.BOLD,20));
```

```
cdtclnname.setFont(new Font("SansSerif",Font.BOLD,20));
```

```
cdtissbank.setFont(new Font("SansSerif",Font.BOLD,20));
```

```
cdtbnkacc.setFont(new Font("SansSerif",Font.BOLD,20));
```

```
cdtbalamnt.setFont(new Font("SansSerif",Font.BOLD,20));
```

```
cdtcvcno.setFont(new Font("SansSerif",Font.BOLD,20));
```

```
cdtint.setFont(new Font("SansSerif",Font.BOLD,20));
```

```
cdtexp.setFont(new Font("SansSerif",Font.BOLD,20));
```

```
Color blue=new Color(51, 204 ,255);
```

```

j3.getContentPane().setBackground(blue);

j3.setLayout(null);

j3.setVisible(true);

j3.setResizable(false);

j3.setSize(1276, 975);

j3.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
}
else if(ae.getSource()==cdtclr)
{
    tfcdtcdid.setText("");
    tfcdtclntname.setText("");
    tfcdtbnkacc.setText("");
    tfcdtissbank.setText("");
    tfcdtbalamnt.setText("");
    tfcdtcvcno.setText("");
    tfcdtint.setText("");
    tfcdtexp.setText("");
}
else if(ae.getSource()==dbtwith)
{
    jf4= new JFrame("Withdraw from Debit Card-Himanshu-22068759");

```

```
withhead= new JLabel("Withdraw from Debit Card");
```

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

```
withamnt= new JLabel("Withdraw Amount:");
```

```
withdow= new JLabel("Date of Withdraw:");
```

```
withpin= new JLabel("PIN Number:");
```

```
tfwithcdid= new JTextField();
```

```
tfwithamnt= new JTextField();
```

```
tfwithpin= new JTextField();
```

```
String dowith[]= {"2022-09-09", "2023-01-04", "2018-09-08", "2022-12-16",  
"2020-09-05", "2023-04-05"};
```

```
datewith= new JComboBox<String>(dowith);
```

```
withdraw= new JButton("Withdraw from Debit Card");
```

```
withdis= new JButton("Display");
```

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

```
withhead.setBounds(356, 35, 455, 50);
```

```
withcdid.setBounds(167, 156, 80, 28);
```

```
withamnt.setBounds(167, 237, 190, 28);
```

```
withdow.setBounds(167, 318, 190, 28);
```

```
withpin.setBounds(167, 399, 168, 28);  
tfwithcdid.setBounds(363, 152, 325, 32);  
tfwithamnt.setBounds(363, 237, 325, 32);  
datewith.setBounds(363, 318, 325, 32);  
tfwithpin.setBounds(363, 399, 325, 32);  
withdraw.setBounds(825, 152, 194, 53);  
withdis.setBounds(825, 265, 194, 53);  
withclr.setBounds(825, 378, 194, 53);
```

```
jf4.add(withhead);  
jf4.add(withcdid);  
jf4.add(withamnt);  
jf4.add(withdow);  
jf4.add(withpin);  
jf4.add(tfwithcdid);  
jf4.add(tfwithamnt);  
jf4.add(datewith);  
jf4.add(tfwithpin);  
jf4.add(withdraw);  
jf4.add(withdis);  
jf4.add(withclr);
```

```
withhead.setFont(new Font("SansSerif",Font.BOLD,36));
```

```

withcdid.setFont(new Font("SansSerif",Font.BOLD,20));
withamnt.setFont(new Font("SansSerif",Font.BOLD,20));
withdow.setFont(new Font("SansSerif",Font.BOLD,20));
withpin.setFont(new Font("SansSerif",Font.BOLD,20));

withdraw.addActionListener(this);
withdis.addActionListener(this);
withclr.addActionListener(this);

Color blue=new Color(51, 204 ,255);

jf4.getContentPane().setBackground(blue);

jf4.setLayout(null);
jf4.setVisible(true);
jf4.setResizable(false);
jf4.setSize(1135, 550);
jf4.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
}
else if(ae.getSource()==withclr)
{
    tfwithcdid.setText("");
    tfwithamnt.setText("");

```

```

        tfwithpin.setText("");
    }
    else if(ae.getSource()==cdtLmt)
    {
        jf5= new JFrame("Set Credit Limit-Himanshu-22068759");

        sethead= new JLabel("Set the Credit Limit");

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

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

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

        tfsetcdid= new JTextField();

        tfsetcdLmt= new JTextField();

        tfsetgrper= new JTextField();

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

        setdis= new JButton("Display");

        setclr= new JButton("Clear");

        sethead.setBounds(382, 32, 330, 50);

        setcdid.setBounds(167, 156, 80, 28);

        setcdLmt.setBounds(167, 277, 140, 28);

        setgrper.setBounds(167,398, 140, 28);
    }
}

```

```
tfsetcdid.setBounds(363, 152, 325, 32);  
tfsetcdlmt.setBounds(363, 261, 325, 32);  
tfsetgrper.setBounds(363, 386, 325, 32);  
setcredit.setBounds(825, 152, 194, 53);  
setdis.setBounds(825, 265, 194, 53);  
setclr.setBounds(825, 378, 194, 53);
```

```
jf5.add(sethead);  
jf5.add(setcdid);  
jf5.add(setcdlmt);  
jf5.add(setgrper);  
jf5.add(tfsetcdid);  
jf5.add(tfsetcdlmt);  
jf5.add(tfsetgrper);  
jf5.add(setcredit);  
jf5.add(setdis);  
jf5.add(setclr);
```

```
setcredit.addActionListener(this);  
setdis.addActionListener(this);  
setclr.addActionListener(this);
```

```
sethead.setFont(new Font("SansSerif",Font.BOLD,36));
```



```

setcdid.setFont(new Font("SansSerif",Font.BOLD,20));
setcdlmt.setFont(new Font("SansSerif",Font.BOLD,20));
setgrper.setFont(new Font("SansSerif",Font.BOLD,20));

Color blue=new Color(51, 204 ,255);

jf5.getContentPane().setBackground(blue);
jf5.setLayout(null);
jf5.setVisible(true);
jf5.setResizable(false);
jf5.setSize(1135, 550);
jf5.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
}
else if(ae.getSource()==setclr)
{
    tfsetcdid.setText("");
    tfsetcdlmt.setText("");
    tfsetgrper.setText("");
}
else if(ae.getSource()==cdtcnl)
{
    jf6= new JFrame("Cancel Credit Card-Himanshu-22068759");

```

```
cnlcdthead= new JLabel("Cancel the Credit Card");
```

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

```
tfcnlcdid= new JTextField();
```

```
cnlcdtcard= new JButton("Cancel Credit Card");
```

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

```
cnlcdthead.setBounds(382, 32, 390, 50);
```

```
cnlcdid.setBounds(167, 156, 80, 28);
```

```
tfcnlcdid.setBounds(363, 152, 325, 32);
```

```
cnlcdtcard.setBounds(126, 248, 194, 53);
```

```
cnlcdtclr.setBounds(715, 248, 194, 53);
```

```
jf6.add(cnlcdthead);
```

```
jf6.add(cnlcdid);
```

```
jf6.add(tfcnlcdid);
```

```
jf6.add(cnlcdtcard);
```

```
jf6.add(cnlcdtclr);
```

```
cnlcdtcard.addActionListener(this);
```

```
cnlcdtclr.addActionListener(this);
```

```
cnlcdthead.setFont(new Font("SansSerif",Font.BOLD,36));
```

```
cnlcdid.setFont(new Font("SansSerif",Font.BOLD,20));
```

```

Color blue=new Color(51, 204 ,255);

jf6.getContentPane().setBackground(blue);

jf6.setLayout(null);

jf6.setVisible(true);

jf6.setResizable(false);

jf6.setSize(1135, 423);

jf6.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
}

else if(ae.getSource()==cnlcdtclr)

{

    tfcnlcdid.setText("");

}

else if(ae.getSource()==dbtadd)

{

    if(tfcddid.getText().isEmpty() || tfclntname.getText().isEmpty() ||
tfbnkacc.getText().isEmpty() || tfissbank.getText().isEmpty() ||
tfbalamnt.getText().isEmpty() || tfpin.getText().isEmpty() )

    {

        JOptionPane.showMessageDialog(jf3,"Empty field found. Please fill the fields
properly.", "Alert",JOptionPane.ERROR_MESSAGE);

    }

else

```

```

{
    try{
        dcdid= Integer.parseInt(tfcdid.getText());
        dpin= Integer.parseInt(tfpin.getText());
        dbalamnt= Integer.parseInt(tfbalamnt.getText());
        dclntname= tfclntname.getText();
        disssbank= tfissbank.getText();
        dbnkacc= tfbnkacc.getText();
        added = false;
        if(al.isEmpty())
        {
            DebitCard ob= new
DebitCard(dbalamnt,dcdid,dbnkacc,dissbank,dclntname,dpin);
            al.add(ob);
            JOptionPane.showMessageDialog(jf3,"Data is added to the Debit Card
system.");
            added=false;
        }
        else
        {
            for(BankCard card:al)
            {
                if(card instanceof DebitCard)

```

```

        {
            DebitCard dbtcd=(DebitCard)card;

            if(card. getCardId()==dcdid)
            {
                JOptionPane.showMessageDialog(jf3,"This Card ID already
exists. Please re-enter your card id.", "Alert",JOptionPane.ERROR_MESSAGE);

                break;
            }
            else
            {
                DebitCard ob= new
DebitCard(dbalamnt,dcdid,dbnkacc,dissbank,dclntname,dpin);

                al.add(ob);

                JOptionPane.showMessageDialog(jf3,"Data is added to the Debit
Card system.");

                added = false;

                break;
            }
        }
        else
        {
            DebitCard ob= new
DebitCard(dbalamnt,dcdid,dbnkacc,dissbank,dclntname,dpin);

            al.add(ob);

```

```

        JOptionPane.showMessageDialog(jf3,"Data is added to the Debit
Card system.");

        added = false;

        break;

    }

}

}

}

catch(NumberFormatException nf)

{

    JOptionPane.showMessageDialog(jf3,"Number format exception detected.
Cannot convert string into int.", "Alert",JOptionPane.ERROR_MESSAGE);

}

}

}

else if(ae.getSource()==cdtadd)

{

    if(tfcdtcdid.getText().isEmpty() || tfcdtclntname.getText().isEmpty() ||
tfcdtbnkacc.getText().isEmpty() || tfcdtissbank.getText().isEmpty() ||
tfcdtbalamnt.getText().isEmpty() || tfcdtcvcno.getText().isEmpty() ||
tfcdtint.getText().isEmpty())

    {

```

```
JOptionPane.showMessageDialog(jf2,"Empty field found. Please enter the  
fields properly.", "Alert",JOptionPane.ERROR_MESSAGE);
```

```
}
```

```
else
```

```
{
```

```
try{
```

```
    ccdid= Integer.parseInt(tfcdtccdid.getText());
```

```
    ccvcno= Integer.parseInt(tfcdtccvcno.getText());
```

```
    cbalamnt= Integer.parseInt(tfcdtcbalamnt.getText());
```

```
    cint= Double.parseDouble(tfcdtcint.getText());
```

```
    cclntname= tfcdtcclntname.getText();
```

```
    cissbank= tfcdtissbank.getText();
```

```
    cbnkacc= tfcdtbnkacc.getText();
```

```
    cexp= withexpdate.getSelectedItemAt(0).toString();
```

```
    added = false;
```

```
    if(al.isEmpty())
```

```
    {
```

```
        CreditCard ob1= new
```

```
CreditCard(ccdid,cclntname,cissbank,cbnkacc,cbalamnt,ccvcno,cint,cexp);
```

```
        al.add(ob1);
```

```
JOptionPane.showMessageDialog(jf2,"Data is added to the Credit Card  
system");
```

```
        added = false;
```

```

    }
else
{
    for(BankCard card:al)
    {
        if(card instanceof CreditCard)
        {
            CreditCard cdtcard=(CreditCard)card;
            if(cdtcard. getCardId()==ccdid)
            {
                JOptionPane.showMessageDialog(jf2,"This Card ID already
exists. Please re-enter it.", "Alert",JOptionPane.ERROR_MESSAGE);
            }
            else
            {
                CreditCard ob1= new
CreditCard(ccdid,cclntname,cissbank,cbnkacc,cbalamnt,ccvcno,cint,cexp);

                al.add(ob1);

                JOptionPane.showMessageDialog(jf2,"Data is added to the
Credit Card system.");

                added = false;
            }
        }
    }
}

```



```

        else
        {
            CreditCard ob1= new
CreditCard(ccdid,cclntname,cissbank,cbnkacc,cbalamnt,ccvcno,cint,cexp);

            al.add(ob1);

            JOptionPane.showMessageDialog(jf2,"Data is added to the Credit
Card System.");

            added = false;

        }

    }

}

}

catch(NumberFormatException nf)

{

    JOptionPane.showMessageDialog(jf2,"Number format exception detected.
Cannot convert string into int.", "Alert",JOptionPane.ERROR_MESSAGE);

}

}

}

else if(ae.getSource()==withdraw)

{

    if(tfwithcdid.getText().isEmpty() || tfwithpin.getText().isEmpty() ||
tfwithamnt.getText().isEmpty())

    {

```

```
JOptionPane.showMessageDialog(jf4,"Empty field found. Please fill the fields properly.", "Alert",JOptionPane.ERROR_MESSAGE);
```

```
}
```

```
else
```

```
{
```

```
try{
```

```
withcrdid= Integer.parseInt(tfwithcrdid.getText());
```

```
withpinno= Integer.parseInt(tfwithpin.getText());
```

```
withwithdrawamnt= Integer.parseInt(tfwithamnt.getText());
```

```
withdowith=datewith.getSelectedItem().toString();
```

```
added=false;
```

```
if(al.isEmpty())
```

```
{
```

```
JOptionPane.showMessageDialog(jf4,"There is no Debit Card. Please enter your debit card properly.", "Alert",JOptionPane.ERROR_MESSAGE);
```

```
}
```

```
else
```

```
{
```

```
for(BankCard all: al)
```

```
{
```

```
if(all instanceof DebitCard)
```

```
{
```

```

        DebitCard with=(DebitCard)all;

        if(withcrdid!=with.getCardId())
        {
            JOptionPane.showMessageDialog(jf4,"The Card ID is invalid.
Please check your Card ID .","Alert",JOptionPane.ERROR_MESSAGE);

            break;
        }
        else
        {
            if(withpinno== with.getPinNo())
            {
                if(withwithdrawamnt> with.getBalanceAmount())
                {
                    JOptionPane.showMessageDialog(jf4,"You don't have
enough balance. Please check your balance
amount. ","Alert",JOptionPane.ERROR_MESSAGE);

                    break;
                }
                else
                {
                    DebitCard bcd=(DebitCard) all;

                    bcd.withdraw(withwithdrawamnt,withdowith,withpinno);

                    JOptionPane.showMessageDialog(jf4,"Money withdrawal is
successful.");

```

```

    }

    }

    else

    {

        JOptionPane.showMessageDialog(jf4,"Invalid Pin Number
entered.Please re-check it.", "Alert",JOptionPane.ERROR_MESSAGE);

        break;

    }

    }

    }

    }

    }

    }

    catch(NumberFormatException f)

    {

        JOptionPane.showMessageDialog(jf4,"Number format exception detected.
Cannot convert string into int.", "Alert",JOptionPane.ERROR_MESSAGE);

    }

    }

    }

    else if(ae.getSource()==setcredit)

    {

```

```

        BankCard bcd= new
CreditCard(ccdid,cclntname,cissbank,cbnkacc,cbalamnt,ccvcno,cint,cexp);

        CreditCard ccd=(CreditCard)bcd;

        if(tfsetcdid.getText().isEmpty() || tfsetcdlmt.getText().isEmpty() ||
tfsetgrper.getText().isEmpty())

        {

            JOptionPane.showMessageDialog(jf5,"Empty field found. Please fill the fields
properly.","Alert",JOptionPane.ERROR_MESSAGE);

        }

        else

        {

            try{

                setcrcdid= Integer.parseInt(tfsetcdid.getText());

                setcrgprper= Integer.parseInt(tfsetgrper.getText());

                setcrcdlmt= Double.parseDouble(tfsetcdlmt.getText());

                added=false;

                if(al.isEmpty())

                {

                    JOptionPane.showMessageDialog(jf5,"There is no Credit Card. Please
enter your credit card properly.","Alert",JOptionPane.ERROR_MESSAGE);

                }

                else

                {

                    for(BankCard each: al)

```

```

{
    if(setcrclid!=each.getCardId())
    {
        JOptionPane.showMessageDialog(jf5,"The Card ID is invalid. Please
check your Card ID.","Alert",JOptionPane.ERROR_MESSAGE);

        break;
    }
    else
    {
        if(setcrclmt>(2.5*(each.getBalanceAmount())))
        {
            JOptionPane.showMessageDialog(jf5,"The amount is insufficient for
the credit limit.","Alert",JOptionPane.ERROR_MESSAGE);

            break;
        }
        else
        {
            CreditCard aa =(CreditCard) each;

            aa.setCreditLimit(setcrclmt,setcrgprper);

            added = true;

            JOptionPane.showMessageDialog(jf5,"The Credit Limit is
sucessfully set.");
        }
    }
}

```

```

        }

    }

}

catch(NumberFormatException f)

{

    JOptionPane.showMessageDialog(jf5,"Number format exception detected.
Cannot convert string into int.", "Alert",JOptionPane.ERROR_MESSAGE);

}

}

}

else if(ae.getSource()==cnlcdtcard)

{

    if(tfcnlcdid.getText().isEmpty())

    {

        JOptionPane.showMessageDialog(jf6,"Empty field found, please re-check
it.", "Alert",JOptionPane.ERROR_MESSAGE);

    }

    else

    {

        try{

            cancrctid=Integer.parseInt(tfcnlcdid.getText());

            added= false;

```

```

for(BankCard all: al)
{
    if(all instanceof CreditCard)
    {
        if(cancrcdid==all.getCardId())
        {
            CreditCard ccd =(CreditCard)all;
            ccd.cancelCreditCard();
            added= false;
            JOptionPane.showMessageDialog(jf6,"The Credit Card is cancelled.");
        }
        else
        {
            JOptionPane.showMessageDialog(jf6,"The Card ID is
invalid.", "Alert",JOptionPane.ERROR_MESSAGE);
        }
    }
    else
    {
        JOptionPane.showMessageDialog(jf6,"There is no any credit card
detected.", "Alert",JOptionPane.ERROR_MESSAGE);
    }
}

```



```

    }

    catch(NumberFormatException f)

    {

        JOptionPane.showMessageDialog(jf5,"Number format exception detected.
Cannot convert string into int.", "Alert",JOptionPane.ERROR_MESSAGE);

    }

}

}

else if(ae.getSource()==dbtdis)

{

    if(al.isEmpty())

    {

        JOptionPane.showMessageDialog(jf2,"There is no an debit card added.
Please fill your debit card properly.", "Alert",JOptionPane.ERROR_MESSAGE);

    }

    else

    {

        for(BankCard all: al)

        {

            if(all instanceof DebitCard)

            {

                DebitCard dis= (DebitCard)all;

                dis.display();

```

```

        }
    }
}

else if(ae.getSource()==cdtdis)
{
    if(al.isEmpty())
    {
        JOptionPane.showMessageDialog(jf3,"There is no an credit card added.
Please fill your credit card properly.", "Alert",JOptionPane.ERROR_MESSAGE);
    }
    else
    {
        for(BankCard all: al)
        {
            if(all instanceof CreditCard)
            {
                CreditCard dis= (CreditCard)all;
                dis.display();
            }
        }
    }
}
}

```

```

else if(ae.getSource()==withdis)
{
    if(al.isEmpty())
    {
        JOptionPane.showMessageDialog(jf2,"There is no an debit card added.
Please fill your debit card properly.", "Alert",JOptionPane.ERROR_MESSAGE);
    }
    else
    {
        for(BankCard all: al)
        {
            if(all instanceof DebitCard)
            {
                DebitCard dis= (DebitCard)all;
                dis.display();
            }
        }
    }
}

else if(ae.getSource()==setdis)
{
    if(al.isEmpty())
    {

```

```
JOptionPane.showMessageDialog(jf3,"The field is empty. Please provide the  
fields properly.","Alert",JOptionPane.ERROR_MESSAGE);
```

```
}
```

```
else
```

```
{
```

```
    for(BankCard all: al)
```

```
    {
```

```
        if(all instanceof CreditCard)
```

```
        {
```

```
            CreditCard dis= (CreditCard)all;
```

```
            dis.display();
```

```
        }
```

```
    }
```

```
}
```

```
}
```

```
}
```

```
public static void main(String[]args)
```

```
{
```

```
    new BankGUI();
```

```
}
```

```
}
```