CS4001NI Programming

30% Individual Coursework

2023-24 spring

Student Name: Apil Thapa

London Met ID: 22067753

College ID: NP01Cp4a220164

Group: L1C8

Assignment Due Date: Wednesday, April 5, 2023

Assignment Submission Date: Wednesday, 10th May

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

Table of Contents

1. Introduction:

1.1 About this project:

This is our java project in which we have create a Gui (graphical user interface) of two different classes **Debitcard** and **creditcard** within a single class called Bankgui. We have created our three different classes **Bankcard, Debitcard** and **Creditcard** in our previous java project. Previously, we are providing their respective values by clicking and runs program through objects, but in this project we are providing and getting values through our gui.In our Bankgui class **Jframe**, **Jpanel**, **Jlabel**, **Jtextfield Jbutton** and **Jcombobox** are used to create gui interface. We have created two **Jframe** and two **Jpanels** for **Debitcard** and **Creditcard**. Both classes have their respective **JPanel** added to **Jframe. Debitcard Jframe** appears first in our gui, there is a button to show **Creditcard jframe** by clicking go to credit card button as per user need. ActionListener which implements ActionPerformed method is written in our project to make our gui more user interactive. Different color combinations are used in our buttons to make our gui more attractive and efficent.

1.2 Tools used:

1.2.1 Ms-word:

Microsoft Word is a word processing program that was first developed by Microsoft in 1983. Since that time, Microsoft has released an abundance of updated versions, each offering more features and incorporating better technology than the one before it. The most current web-based version of Microsoft Word is Microsoft 365, but the software version of Microsoft Office 2019 includes Word 2019.

Microsoft Word is included in all of the Microsoft 365 application suites. The most basic (and least expensive) suites also include Microsoft PowerPoint and Microsoft Excel. Additional suites exist and include other Office programs, such as Microsoft Outlook and Skype for Business. (Ballew, 2021)

1.2.2 Blue-j:

BlueJ is a development environment that allows you to develop Java programs quickly and easily. Its main features are that it is:

- **Simple** BlueJ has a deliberately smaller and simpler interface than professional environments like NetBeans or Eclipse. This allows beginners to get started more quickly, and without being overwhelmed.
- **Designed for teaching** BlueJ is deliberately designed with good pedagogy in mind. There is a popular textbook designed for teaching introductory university/college courses with BlueJ, and a site full of teaching resources.
- **Interactive** BlueJ allows you to interact with objects. You can inspect their value, call methods on them, pass them as parameters and more. You can also directly invoke Java expressions without compiling. Thus BlueJ is a powerful graphical shell/REPL for Java.
- **Portable** BlueJ runs on Windows, Mac OS X, Linux and other platforms which run Java. It can also run without installation from a USB stick.
- **Innovative** BlueJ has several features not seen before in other IDEs. Its object bench, code pad, and scope colouring were all original BlueJ features. (gosling, 20 september 2022)

1.2.3 Draw.io:

diagrams.net and our draw.io branded Atlassian integrations are the leading solution for web based sketching and diagramming functionality. You can use diagrams.net online with a number of storage platforms, and offline with a standalone desktop app.

As a security-first diagramming app for teams, we provide the diagramming functionality - you choose where to keep your diagram data. There are many different integrations with other platforms and applications, including Atlassian Confluence Cloud and Jira Cloud, Google applications, GitHub and Microsoft applications. Unofficial integrations are available for many other platforms and tools. (JGraph Ltd, 2005)

2. Class diagram:

Class diagram is is a type of uml (unified modeling language) which helps/represent structures of the system by showing relationships of classes with each other.

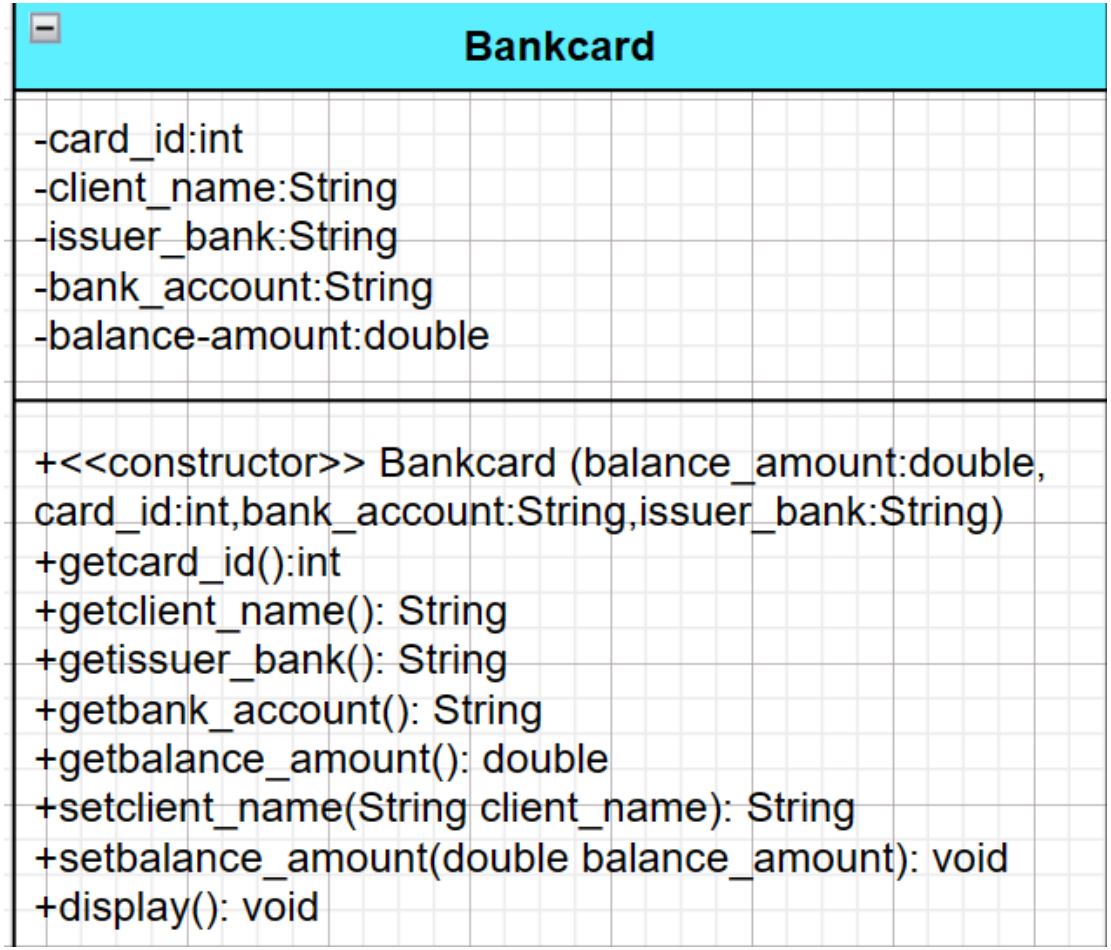

Figure 1 2.1 Bank card

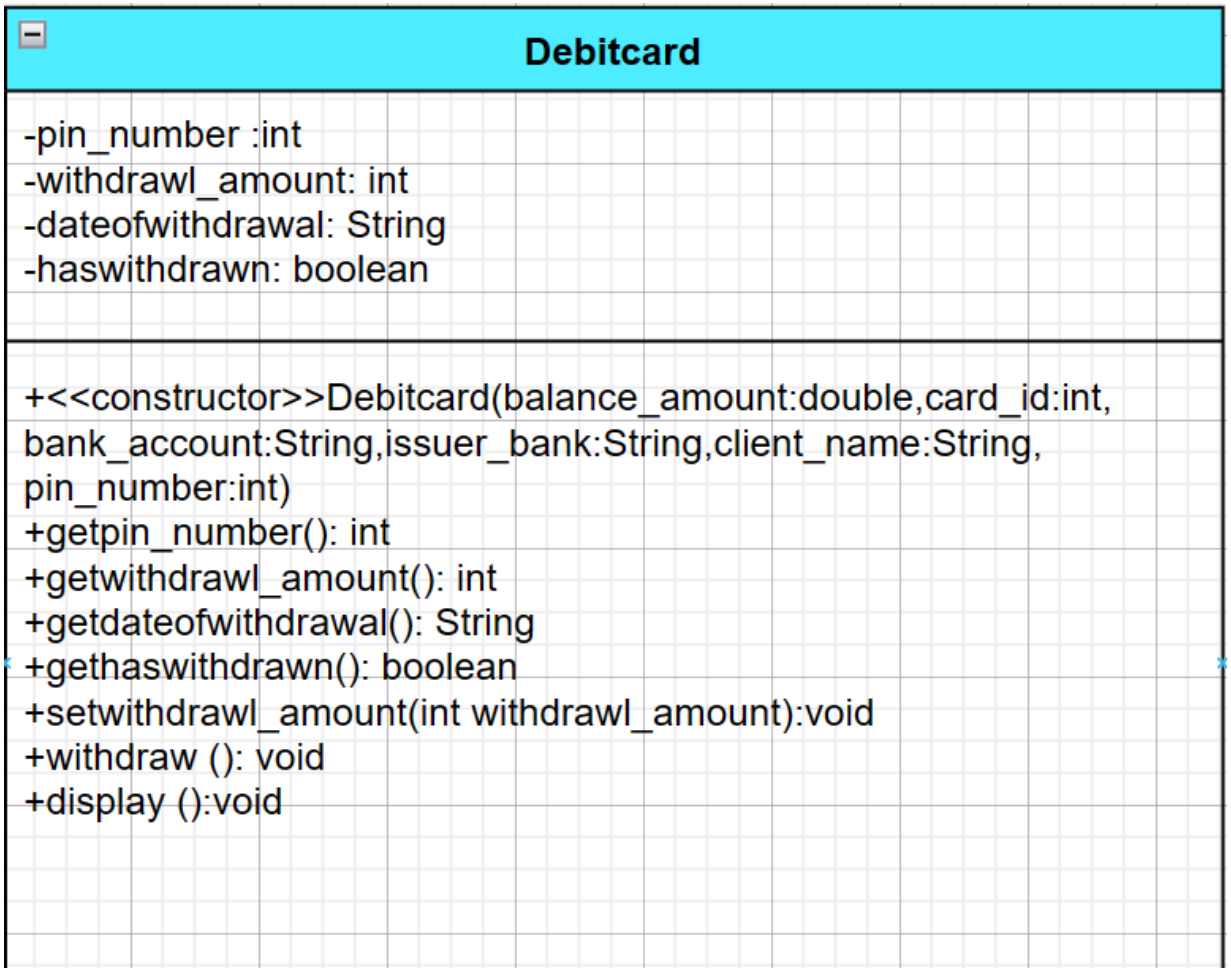

Figure 2 2.2 debit card

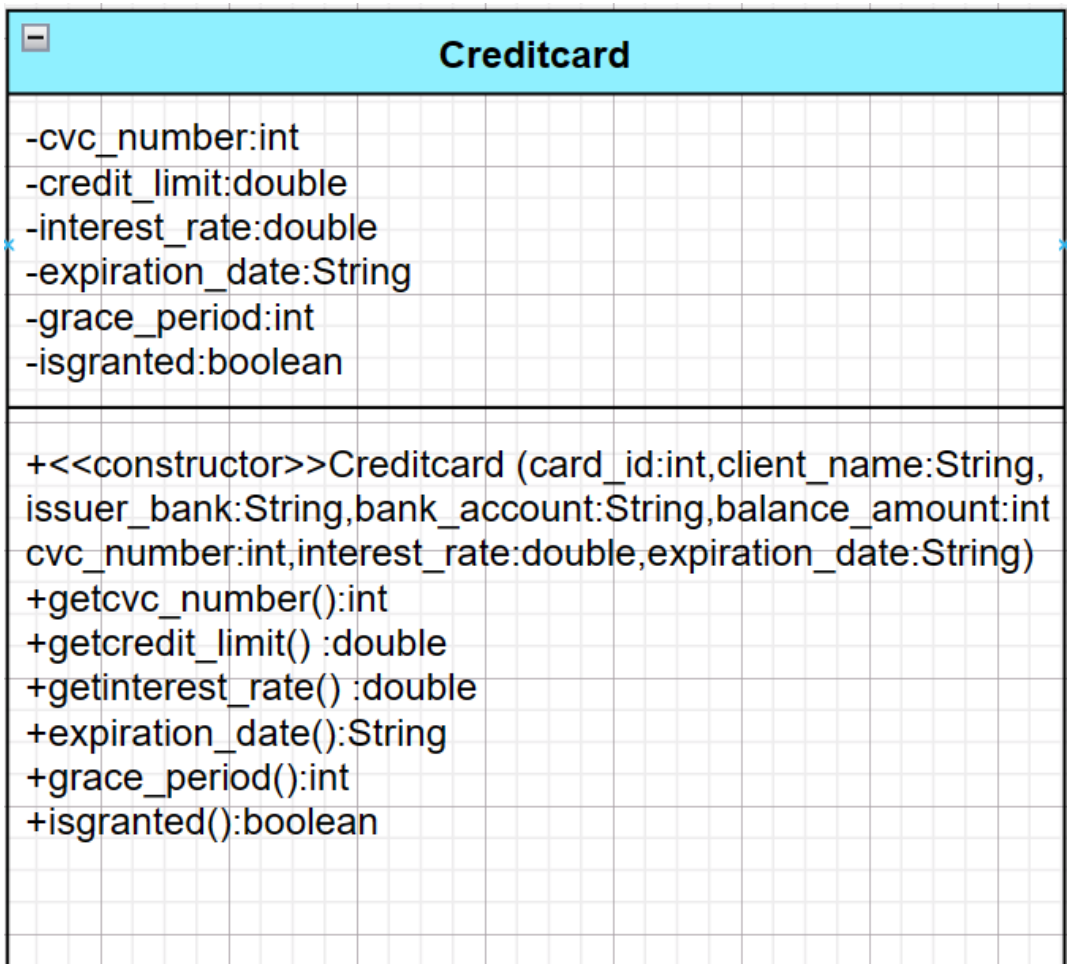

Figure 3 2.3 creditcard

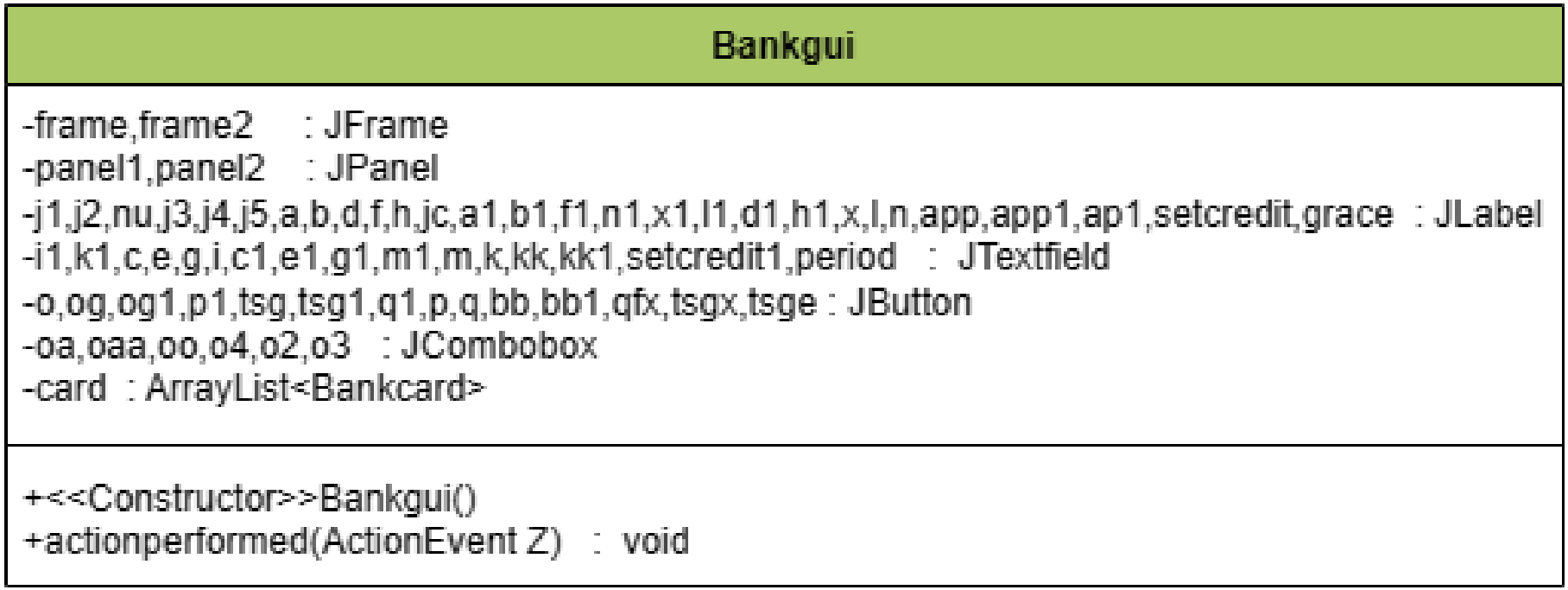

Figure 4 2.4 Bankgui

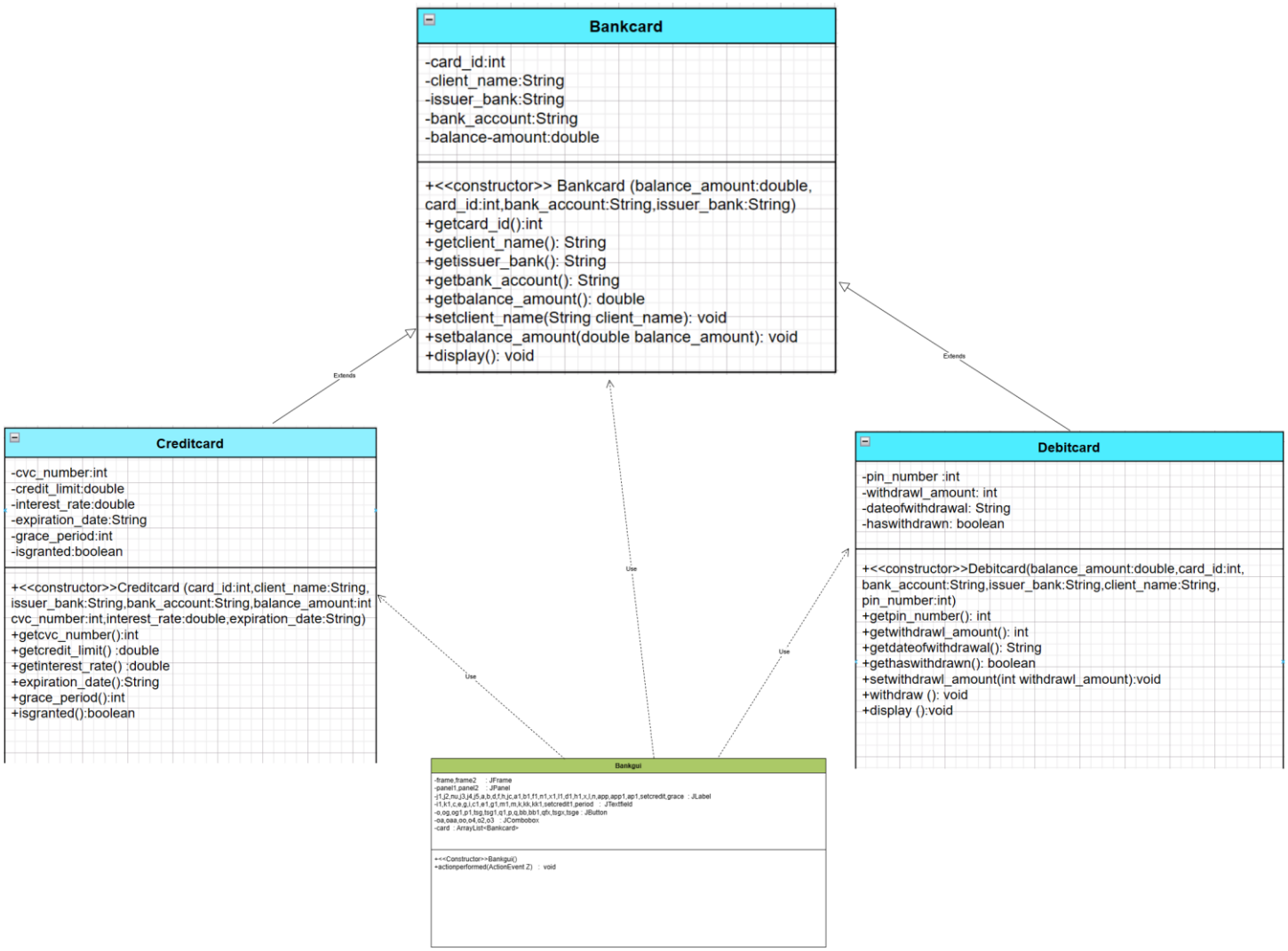

Figure 5 2.5 relation

3. Pseudocode:

DO

Declare swing components Jframe named as frame and frame2.

Declare swing components Jpanel named as panel1 and panel2.

Declare swing components Jlabel named as j1, j2 , nu, j3 ,j4 ,j5 ,a, b ,d ,f ,h ,j

,a1 ,b1 ,f1, n1 ,x1, l1 ,d1 ,h1 ,x, l, n, app, app1 ,ap1 ,setcredit and grace.

Declare swing component Jtextfield named as i1, k1, c, e, g, i, c1, e1, g1, m1, m, k, kk, kk1

, setcredit1 and period.

Declare swing component Jbutton named as o,og,og1,p1,tsg,tsg1,q1,p,q,bb,bb1,qfx,tsgx

And tsge.

Declare swing components Jcombobox as type generic named as oa, oaa, oo, o4, o2 and o3.

Create an ArrayList of type Bankcard class referenced as card.

Declare object of Debitcard and Creditcard class as obj and obj1.

END DO

Create a constructor named as Bankgui with empty parameter

DO

Initialize two swing components Jframe referenced as frame and frame2 named as Debicard and creditcard

Which is set bounded to their respective position.

Initialize two swing components Jpanels referenced as panel1 and panel2 which are set bounded to their positions and added to their respective swing components named as Jframe.

Initialize all swing component Jlabels of debitgui referenced as **a, b, d, f, h, x, app, l and n** named by Debitgui, cardid, balance amount, client name, issuer bank, bank account, withdrawal amount, Pin number and date of withdrawal which are setbounded to its position and sets its font and color.

Initialize all swing components Jlabels of creditgui named as interest rate, cvc number, card id , client name,issuer bank, bank account, setcreditlimit and grace period also expiration date which are setbounded to its position and sets their font and color.

Initialize swing components jTextfield of credit and debit GUI for every Jlabels in the gui such as cardid, balance amount, client name, issuer bank, bank account, withdrawal amount, Pin number, interest rate, cvc number, card id , client name,issuer bank, bank account, setcreditlimit and grace period also expiration date .

Create a new Jcombobox with first, second and third set of options as day, month and year to set date of withdrawal in debitcard GUI.

Create a new jCombobox with with first, second and third set of options for day, month and year for creditcard GUI.

Initialize a new Jbutton for the creditcard gui and debitcard gui as add credit card, clear, display, ,Go to debi card button, setcreditlimit, cancel creditcard and add debitcard , display, clear, go to creditcard and withdraw button also setted those color and background setbounded to their size and position.

Registered every button using this keyword which are used in event handeling.

Swing components are added to their respective panels which later on added to frames.

Frame layout is setted to null and visibility to true also its default close operation is setted to exit on close.

Buttons focusable is setted to false in order to avoid rectangular shape when user clicks on it.

END DO

Create an actionperformed method in which ActionEvent and referenced name is passed as parameter with no return type.

3.1 For debitcard clear button

DO

IF referenced name dot getsource with no parameter is equals to tsg which is clear button of creditcard

Set the text for **textfield** e1 to an empty **string**.

Set the text for **textfield** c1 to an empty **string**.

Set text for **textfield** i1 to an empty **string**.

Set text for **textfield** k1 to an empty **string**.

Set text for **textfield** g1 to an empty **string**.

Set text for **textfield** m1 to an empty **string**.

Set text for **textfield** kk1 to an empty **string**.

Set text for **textfield** setcredit1 to an empty **string**.

Set text for **textfield** period to an empty **string**.

END IF

END DO

3.2 for creditcard clear button

DO

ELSE IF referenced name dot getsource with no parameter is equals to bb which is clear button of Debitcard

Set the text for **textfield** c to an empty **string**.

Set the text for **textfield** e to an empty **string**.

Set the text for **textfield** g to an empty **string**.

Set the text for **textfield** i to an empty **string**.

Set the text for **textfield** k to an empty **string**.

Set the text for **textfield** k to an empty **string**.

Set the text for **textfield** m to an empty **string**.

Set the text for **textfield** kk to an empty **string**.

END ELSE IF

END DO

DO

ELSE IF referenced name dot getsource is equals to Go creditcard button

Set frame2 (creditcard frame) to visible.

Set frame (debitcard frame) to invisible.

END ELSE IF

END DO

DO

ELSE IF referenced name dot getsource is equals to go **Debitcard** button

Set frame2 (**Creditcard** frame) to invisible.

Set frame (**Debitcard** frame) to visible.

END ELSE IF

END DO

3.3 For creditcard add button.

DO

ELSE IF referenced name dot getsource is equals to p1 (add credit card button).

IF C1 dot get text dot **is Empty**, e1 dot **gettext** dot **is Empty** , i1 dot gettext dot **is Empty** , m1 dot

Gettext dot is Empty and l1 dot **gettext** dot **is Empty** with no parameter

Display warning message which shows textfield **is empty**

ELSE

IF referenced name of an **ArrayList** card dot **is Empty**

It is kept under try block not to face any exception during program execution if occurs catch block will execute.

Get the text value in Integer using **parseInt** method from textfield named as c1 and stored it into variable named xbn.

Get the text value in **Integer** using **parseInt** method from textfield named as e1 and stored it into variable named ynv.

Get the text value in **Integer** using **parseInt** method from textfield named as i1 and stored it into variable named xbm1.

Get the text value in **Integer** using **parseInt** method from textfield named as m1 and stored it into variable named y1.

Get the text value in **Integer** using **parseInt** method from textfield named as period and stored it into variable named gracy.

Get the **string** value using **gettext** method of textfield l1 and stored it into variable named yj1.

Get the **string** value using **gettext** method of textfield k1 and stored it into variable named x3.

Get the **string** value using gettext method of textfield g1 and stored it into variable named x2.

Get the **string** value using gettext method of textfield kk1 and stored it into variable named hh.

Get the **string** value using gettext method of textfield setcredit1 and stored it into variable named setcredit1.

Get value of day,month and year from **combobox** oo,oo and oaa using dot **getSelecteditem**() method and cast it with datatype **String** also stores it into their respective variables named as sun,jan,first and concatinate all and stores in variable named as date.

Create an object of creditcard class as obj1 with parameter as card_id, client_name, issuer_bank, bank account , balance_amount, cvc_number, interest_rate and expiration_date as getted values **xbm1,"x2","x3","hh",xbn,ynv,y1,date**.

ArrayList will stores creditcard object.

Display success message through message dialog box saying creditcard succesfully added.

Set all textfield to an empty **String**.

Run if any exception occurs and shows message that exception has ocuured.

ELSE

FOR Each Bankcard converted as its instance will iterates over an arraylist as card

IF Bankcard instance child class creditcard

Bankcard instance is converted to **Creditcard**, finally

IF Object of creditcard dot getter method card id equals

getted value of card id for each iteration it will display error message ‘duplicate card id’.

ELSE

create an object of creditcard class as obj1 with parameter as card_id, client_name, issuer_bank, bank account , balance_amount, cvc_number, interest_rate and expiration_date as getted values **xbm1,"x2","x3","hh",xbn,ynv,y1,date**

Add **obj** and store in an **ArrayList** also display succes message saying “card succesfully added”.

Set all textfield to an empty **String**.

END ELSE

END IF

END ELSE

END ELSE IF

END DO

3.4 Debitcard add button

DO

ELSE IF referenced name dot getsource is equals to p1 (add credit card button).

IF C dot get text dot **is Empty**, e dot **gettext** dot **is Empty**, i dot gettext dot **is Empty**, g dot

Gettext dot is Empty and k dot **gettext** dot **is Empty** with no parameter

Display warning message which shows textfield **is empty**

ELSE

IF referenced name of an **ArrayList** card dot **is Empty**

It is kept under try block not to face any exception during program execution if occurs catch block will execute.

Get the text value in Integer using **parseInt** method from textfield named as c and stored it into variable named x5.

Get the text value in **Integer** using **parseInt** method from textfield named as e and stored it into variable named x6.

Get the text value in **Integer** using **parseInt** method from textfield named as m and stored it into variable named y7.

Get the text value in **Integer** using **parseInt** method from textfield named as kk and stored it into variable named withdrawalamount.

Get the **string** value using **gettext** method of textfield g and stored it into variable named y5.

Get the **string** value using **gettext** method of textfield i and stored it into variable named y6.

Get the **string** value using gettext method of textfield k and stored it into variable named y7.

Get the **string** value using gettext method of textfield kk1 and stored it into variable named hh.

Get the **string** value using gettext method of textfield setcredit1 and stored it into variable named setcredit1.

Get value of day,month and year from **combobox** oo,oo and oaa using dot **getSelecteditem**() method and cast it with datatype **String** also stores it into their respective variables named as mon, feb,second and concatinate all and stores in variable named as date1.

Create an object of Debitcard class as obj with parameter as card_id, client_name, issuer_bank, bank account, balance_amount, pin number, interest_rate as getted values **x6, y7,"x7","y5","y6", x5**

ArrayList will stores Debitcard object.

Display success message through message dialog box saying Debitcard succesfully added.

Set all textfield to an empty **String**.

Run if any exception occurs and shows message that exception has ocuured.

ELSE

FOR Each Bankcard converted as its instance will iterates over an arraylist as card

IF Bankcard instance child class Debitcard

Bankcard instance is converted to **Debitcard**, finally

IF Object of Debitcard dot getter method card id equals

Getted value of card id for each iteration it will display error message ‘duplicate card id’.

ELSE

Create an object of **Debitcard** class as obj1 with parameter as card_id, client_name, issuer_bank, bank account

, balance_amount, pin number, interest_rate as getted values **x6,y7,"x7","y5","y6",x5**

Add **object** and store in an **ArrayList** also display succes message saying “card succesfully added”.

Set all textfield to an empty **String**.

END ELSE

END IF

END ELSE

END ELSE IF

END DO

3.5 Go to Creditcard button.

DO

ELSE if when the event source is ‘q’

Set the visibility of creditcard frame (frame2) to ‘true’.

Set the visibility of Debitcard frame (frame) to ‘false.

END ELSE IF

END DO

3.6 Go to debitcard button.

DO

ELSE if when the event source is ‘q1’

Set the visibility of creditcard frame (frame2) to ‘false.

Set the visibility of Debitcard frame (frame) to ‘true.

END ELSE IF

END DO

3.7 Withdraw button.

ELSE IF when the event source is ‘qfx’ (withdraw button from debicard)

IF C dot get text dot **is Empty**, e dot **gettext** dot **is Empty**, i dot gettext dot **is empty**, g dot

Gettext dot is Empty and k dot **gettext** dot **is Empty** with no parameter

Display warning message which shows textfield **is empty**

ELSE

IF referenced name of an **ArrayList** card dot **is Empty**

It is kept under try block not to face any exception during program execution if occurs catch block will execute.

Get the combobox Mon using dot getselectedItem method as String.

Get the combobox Feb using dot getselectedItem method as String.

Get the combobox second using dot getselectedItem method as String.

FOR EACH 'BANKCARD' object withdraw in 'card' list

DO

IF 'withdraw' is an instance of 'Debitcard'

IF DEBITCARD object dot card id from 'BANKCARD' class equals

To Getted textfield of card id

Cast the Bankcard object as Debitcard

Call the withdraw method from Debitcard class with

Parameter as withdrawal amount, pin number and

Date of withdrawal.

Display the succes message ‘withdrawn succefully’.

ELSE

Display an error message ‘Card id is incorrect’.

Catch the number format Exception if occurs.

Display warning message ‘Exception occurs’.

END DO

END IF

END IF

END ELSE IF

3.8 Setcreditlimit

ELSE IF when the event source is ‘qfx’ (withdraw button from debicard)

IF C1 dot get text dot **is empty**, e1 dot **gettext** dot **is Empty**, i1 dot gettext dot **is Empty**, m1 dot

Gettext dot is Empty and l1 dot **gettext** dot **is Empty** with no parameter

Display warning message which shows textfield **is empty**

ELSE

IF referenced name of an **ArrayList** card dot **is Empty**

Display message ‘empty card id found’

It is kept under try block not to face any exception during program execution if occurs catch block will execute.

Get the text value of period in Integer using parseInt method and store in variable grace period?

Get the text value of of textfield i1 in Integer using parseint method and store in variable cardd.

Get the text value of of setcredit1in Integer using parseint method and store in variable set limit.

FOR EACH ‘BANKCARD’ object setlimited in ‘card’ list

DO

IF ‘setlimited is an instance of ‘Creditcard

IF Creditcard object dot card id from ‘BANKCARD’ class equals

To Getted textfield of card id

Cast the Bankcard object as Creditcard.

Call the **setcreditlimit** from Creditcard class from object of

Bankcard with parameter as setlimit and graceperiod.

Display the success message ‘credit limit set successfully’.

Catch the number format Exception if occurs.

Display warning message ‘Exception occurs’.

END DO

END IF

END IF

END ELSE IF

3.9 Cancel credit limit

ELSE IF when the event source is ‘tsge’ (cancel creditlimit)

IF Textfield i1 dot get text is Empty

Display message ‘Creditcard not added’

ELSE

IF referenced name of an **ArrayList** card dot **is Empty**

Display message ‘empty card id found’

ELSE

It is kept under try block not to face any exception during program execution if occurs catch block will Execute.

Get the textfield i1 in integer using parse int method and store in variable cardes.

FOR EACH ‘BANKCARD’ object id in ‘card’ list

DO

IF ‘id’ is an instance of ‘Creditcard

Cast Bankcard as creditcard for object cancel.

Call the cancelcredit method from Creditcard class with no parameter.

Display success message ‘succesfully cancelled’.

Catch the number format Exception if occurs.

Display warning message ‘Exception occurs’

END IF

END DO

END IF

END ELSE IF

Y

3.10 Display button from creditcard

ELSE IF when the event source is ‘tsg1’ (display button for creditcard)

FOR EACH ‘BANKCARD’ object know in ‘card’ list

DO

IF ‘know is an instance of ‘Creditcard

Cast the Bankcard as creditcard as display1 object.

Call the display method from creditcard class using casted object.

Display success message ‘details displayed’

END IF

END DO

END ELSE

3.11 Display method from Debitcard

ELSE

FOR EACH ‘BANKCARD’ object us in ‘card’ list

DO

IF ‘us’ is an instance of ‘Debitcard

Cast the Bankcard as Debitcard as display1 object.

Call the display method from Debitcard class using casted object.

Display success message ‘details displayed’

END IF

END DO

END ELSE

4. Method Description:

4.1. Add Debit card:

This is a button of **Debitcard** jframe. When this button is pressed. All the given texfield such as **balance amount, card id, Issuer bank, bank account, client name and pin number** values are getted. New object of **Debitcard** class is created out there in which values which are getted from the textfields are added to an objects of **Debitcard class.** An **ArrayList** of **Bankcard** class is also created which stores the objects of **Debitcard** class. While adding we will check first that given textfields from gui is empty or not if yes then certain message is displayed through message dialog box.if not then it will go inside and checks again if an **ArrayList** is empty or not if an **ArrayList** is empty then it will get the value from textfield and convert into integer value using parseInt method which will directly add an object to an **ArrayList**. But in case **ArrayList** is found to be non empty then it will check if cardid existing is equals with an cardid of previous class then it will display an error message if not then it will iterates through an **ArrayList** using for each loop and cast parent class as child class then eventually it will add debitcard to an **ArrayList** providing success message through message dialog box and so on, Proper use of try/catch is implemented to avoid unnecessary exception during the execution of a program.

4.2. Add Creditcard class:

This is a button of **Creditcard** Jframe.In this button we have to add different values of textfield into creditcard objects and An **ArrayList** should be created of Bankcard class which stores **creditcard** objects.while doing this we need to first get the values of different textfields such as **card Id, client name, issuer bank, bank account, balance amount, CVC number, Interest rate also one combobox** of **expiration date.**These values are added to an objects of **Creditcard** class later which is also stored in an **ArrayList** of **Bankcard** class. While adding we will check first that given textfields from gui is empty or not if yes then certain message is displayed through message dialog box.if not then it will go inside and checks again that if an **ArrayList** is empty or not if an **ArrayList** is empty then only it will get the value from textfield and convert into integer value using pareseInt method.which will directly add an object to an **ArrayList**. But in case **ArrayList** is found to be non empty then it will check if **cardid** existing is equals with an **cardid** of previous class then it will display an error message if not then it will iterates through an **ArrayList** using for each loop and cast parent class as child class then eventually it will add Creditcard to an **ArrayList** providing success message through message dialog box and so on, Proper use of try/catch is implemented to avoid unnecessary exception during the execution of a program.

4.3. Withdraw from debitcard:

This is a withdraw Button from **Debitcard** jframe. While clicking this button values from the texfield **pin number, withdrawal amount, card id** and **date of withdrawal** are getted. An object of **Bankcard** is cast as **Debitcard** (Downcasting) in order to call child class attributes or methods from instances of parent class. Use of for each loop is implemented in this program to check existing card id from **Bankcard** class is equals to **card id** entered in the gui through textfields or not. If yes then **instance of** parent class calls the withdraw method of child class. Firstly, we will check if the textfields in the gui are empty or not if yes then certain message is displayed using message dialog box but if not then we will check if an **ArrayList** is empty or not if **ArrayList** is empty then it will sure that our **Debitcard** id is not added but if not then we will get the values from textfields such as **withdrawal amount** , **pin number** and **card id** also from the **combobox date of withdrawal.**then we will move to for each loop where it upcasts the **Bankcard** class and checks for each iteration in an **ArrayList** value changes. It will goes through nested where it checks either upcasted variable instance of **Debitcard** class or not, if yes then it will goes through another nested if statement In which it checks if getter method card_id store in **Bankcard** class equals with **card id** of gui.if yes then downcast the method as **Debitcard** class which calls the withdraw method from **Debitcard** class and displays the success message with the help of message dialog box. Proper use of try/catch is implemented to avoid unnecessary exception during the execution of a program.

4.4 Setcreditlimit.

This is a button from **Creditcard** Jframe. While clicking this button values of **card id, grace period** and **setcreditlimit** is getted. An object of **Bankcard** is cast as **Creditcard** (Downcasting) in order to call child class attributes from parent class instances. For each loop is used in this program to Iterates through an **ArrayList** to check if card id of getter method in **Bankcard** class is equals to card id of Gui through textfield. If yes then **instance of** parent class calls the **setcredit** method of child class. Firstly we will check if textfields contains empty value or not if yes then certain error message is displayed but if not then it will go inside else statement and checks id **Creditcard** is added or not if no then else statement runs which again displays some error message but if yes then values of textfields such as **grace period**, **setcreditlimit** and **card id** is getted then We **upcast** our **Bankcard** class with variable and it runs in an for each loop which checks if **Bankcard** variable which is **upcasted instanceof Creditcard** or not if yes in every iteration card id of getter method from **Bankcard** class equals to card id from textfields if yes then it will go inside and **downcast Bankcard** class as **Creditcard** and that attributes from parent class is used to call the method **setcreditlimit** of **Creditcard** class finally success message is displayed from message dialog box. Proper use of try/catch is implemented to avoid unnecessary exception during the execution of a program.

4.5 CancelCreditlimit:

This is a button from **Creditcard** jframe. While clicking this button card id is getted. An object of **Bankcard** is cast as **Creditcard** (**Downcasting**) in order to call child class method from parent class objects.for each loop is used in this program to Iterates through an **ArrayList** of **Bankcard** class to checks if **card id** of getter method is equals to **card id** of Gui from textfields. Firstly we will check if textfields of **card id** is empty or not if yes then certain error message is displayed using message dialog box if not then it will go inside another if else statement to check wheather **Creditcard** is added or not if not then again error message is displayed but if yes then **card id** of textfield is getted as an Integer using **Integer.parseint** method. In for each loop **Bankcard** class is **upcasted** and iterates to check if **upcasted** variable instanceof **Creditcard** or not if yes then it will go inside another if else statement and **Downcast Bankcard** instance as Creditcard which is used to call the method called **cancelcredit** of **Creditcard** class Finally succes message is displayed using mesagedialog box. Proper use of try/catch is implemented to avoid unnecessary exception during the execution of a program.

4.6. Display debit card

This is a button from **Debitcard** Jframe. When clicking this button details regarding that particular card is displayed.

For this button we need to call the method **display** () from **Debitcard** class. For this we have created an **instanceof Debitcard** class using that instance we call the **display** method which displays the information regarding that particular class.

4.7 Display Creditcard:

This is a button from **Creditcard** Jframe. When clicking this button details regarding that particular card is displayed.

For this button we need to call the method **display** () from **Creditcard** class. For this we have created an **instanceof**

Creditcard class using that instance we call the **display** method which displays the information regarding that particular class.

4.8 Clear:

This is button implemented in both **Debitcard** and **Creditcard** class. Generally when this button is pressed values from all textfields are cleared. For this we use **settext ()** method. It is set it to an empty string which cleared out all textfields when user clicks clear button from Gui.

5. Testing:

Table 1 to test wheather the program can be compiled and run using command prompt.

Test	1
Objective	To check wheather the program program can be compiled and run using command prompt
Action	Open command prompt in project location and type: ➡ Java –version ➡ Javac Bankgui.java (File name.java) ➡ Java Bankgui (File name)
Expected result	The program need to compile successfully also GUI Frame must appear.
Actual result	The program is compiled successfully and GUI Frame appeared.
Conclusion	The test is succesfull.

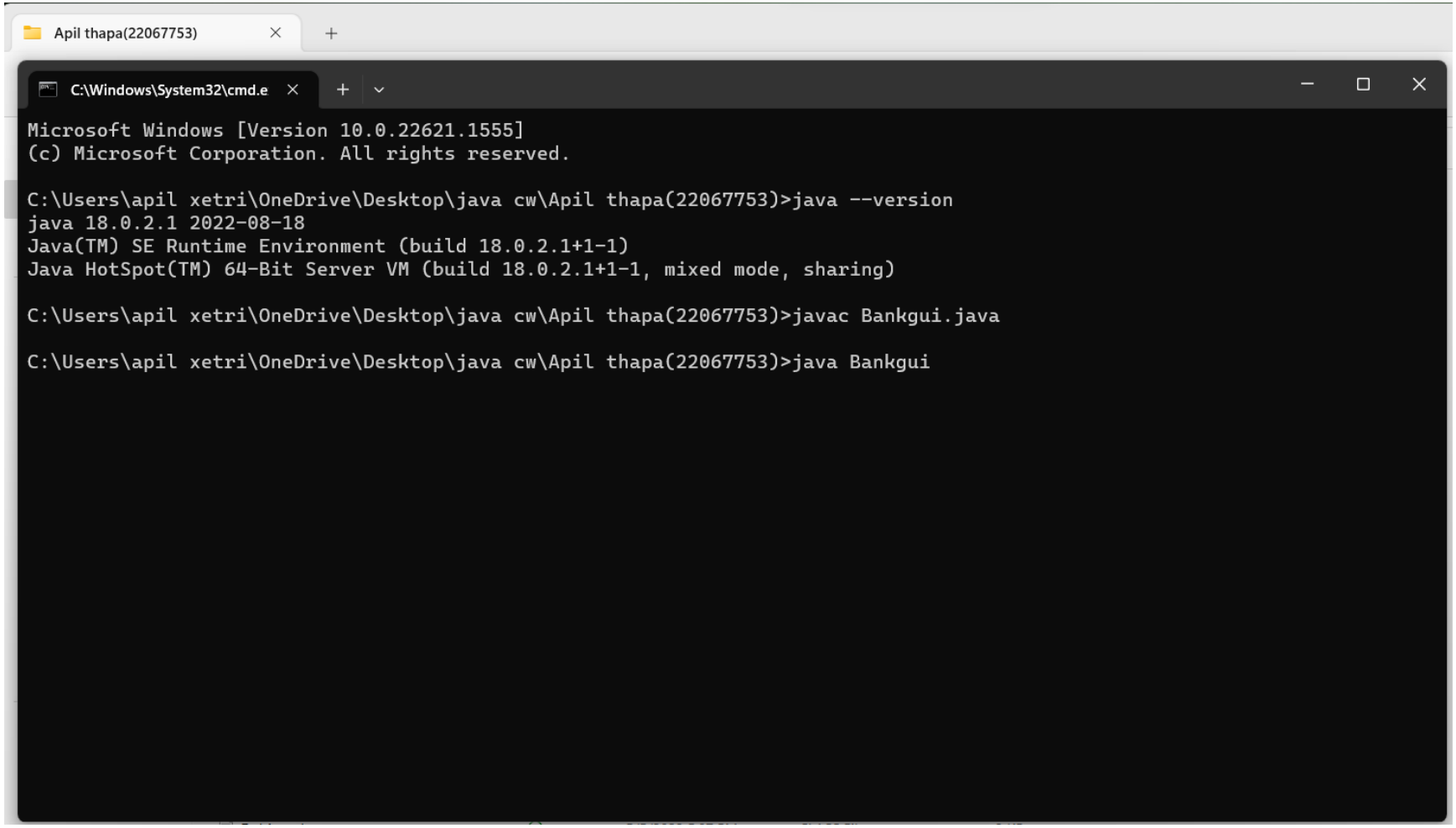

Figure 6 Screenshot of compilation with command prompt

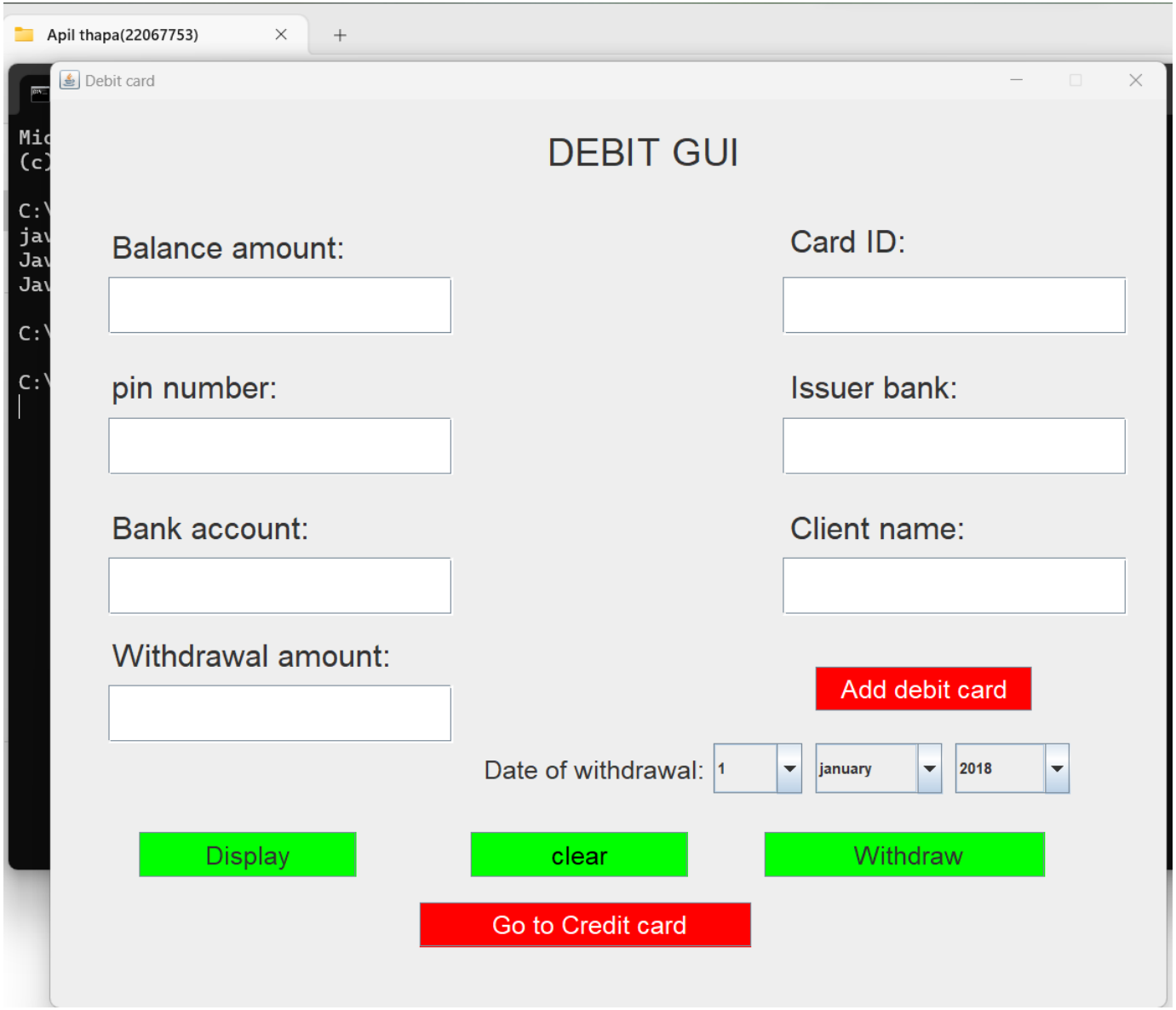

Figure 7 run after compilation

Table 2 Test-2 Evidences that should be shown

Test-2 Evidences that should be shown

Test 2.1: Add debit card

Test 2.2 Add credit card

Test 2.3 Withdraw the amount

Test 2.4 set the credit limit

Test 2.5 Cancel credit limit

Test	2
Objective	To Add debit card. To Add credit Card. Withdraw the amount. Set credit limit. Cancel the credit card.
Action	➡ Fill all the textfield of Debitcard Gui. ➡ Click to add debitcard button. ➡ Withdraw certain amount from it. ➡ Fill all the textfield of Creditcard Gui. ➡ Click on Set credit limit button.

Expected result	➡ Click on Cancel credit limit button. ➡ Click on each display button to display. Debitcard must be added. Withdrawn must be done. Creditcard must be added. Creditlimit should be setted out. Creditcard should be cancelled out.
Actual result	Debitcard is added successfully. Withdraw process is done successfully. Creditcard is added successfully. Creditlimit is setted.

	Creditcard cancelled out successfully.
Conclusion	The test is successful.

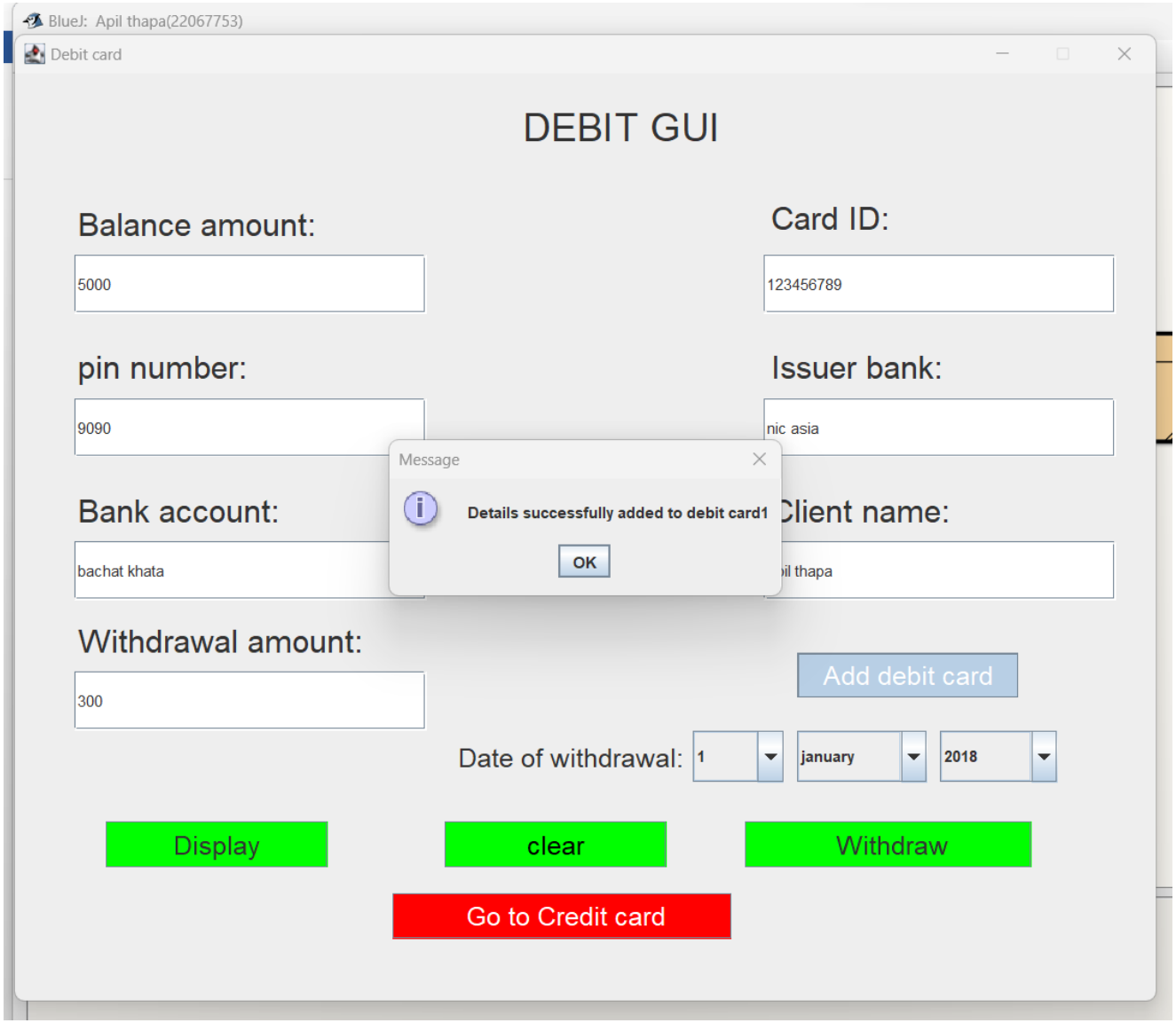

Figure 8Screenshot of Add debit card

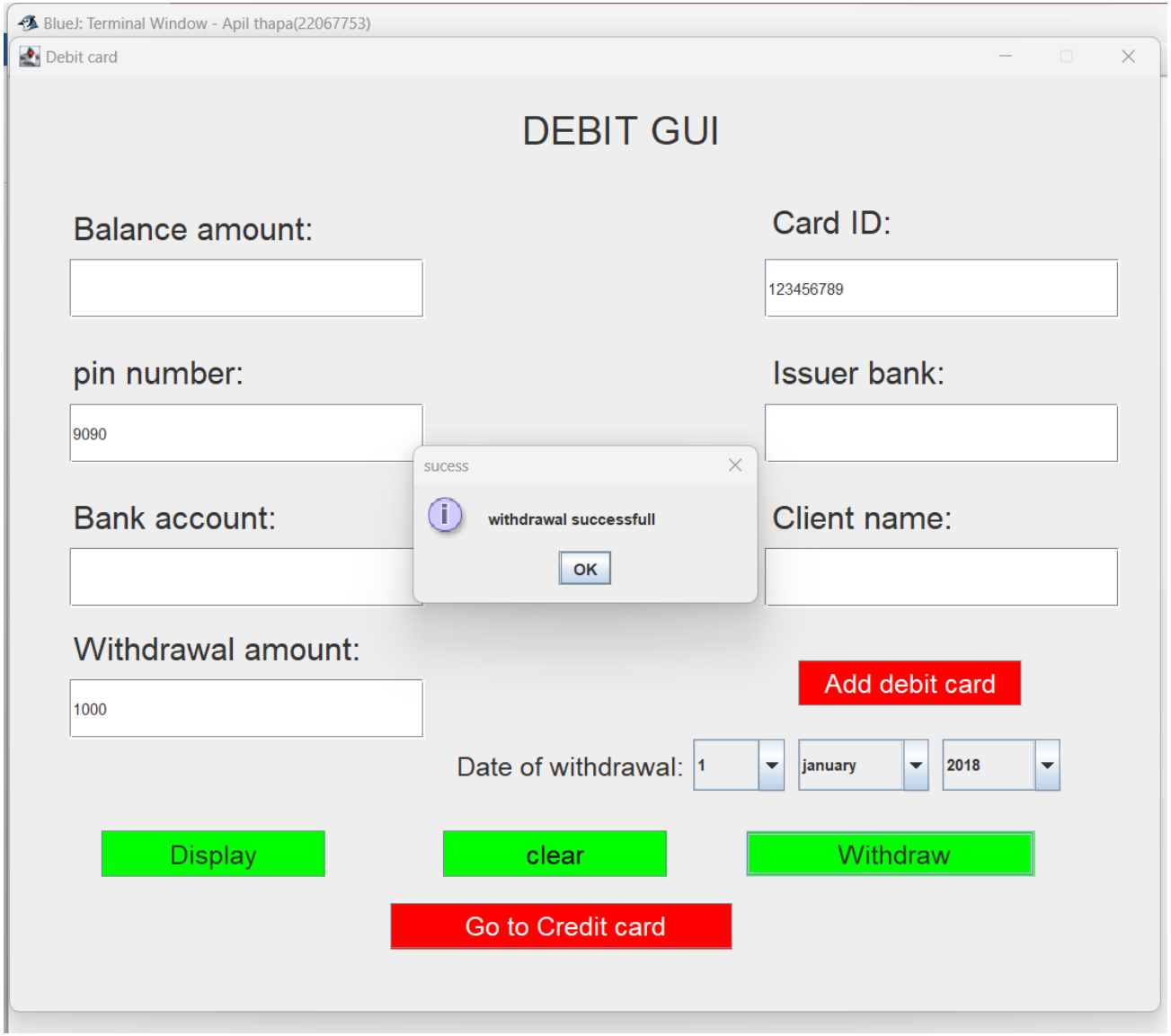

Figure 9 Screenshot of withdrawal process

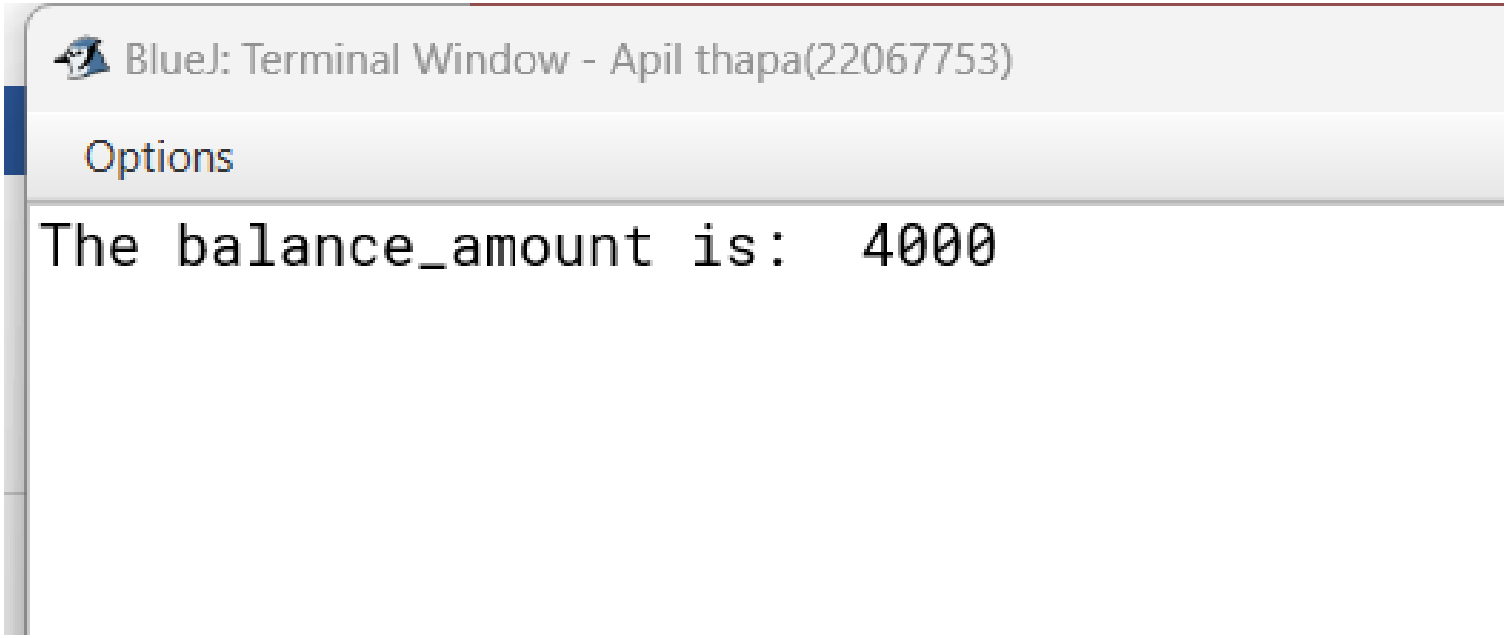

Figure 10 Screenshot of withdraw

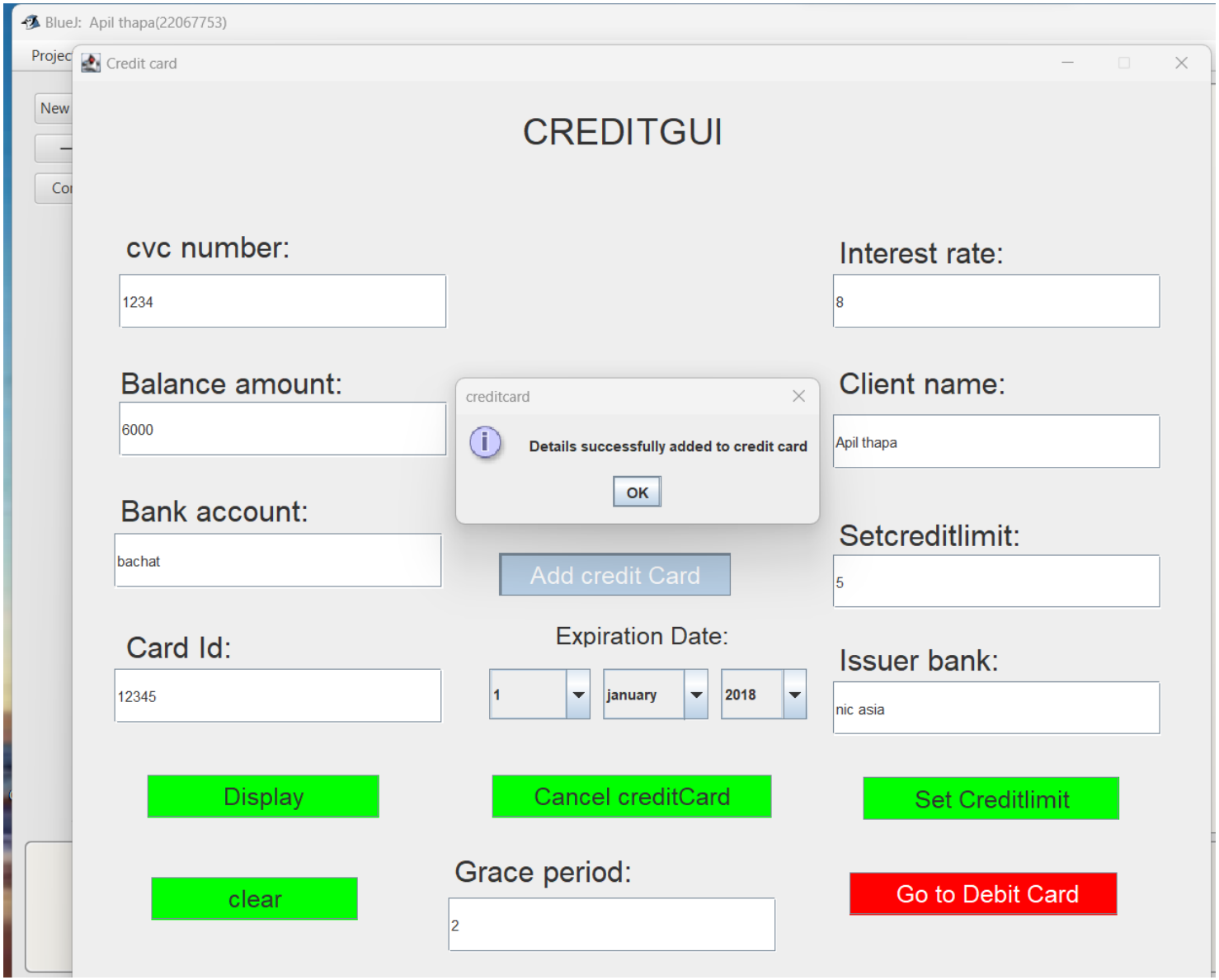

Figure 11 Screenshot of add credit card

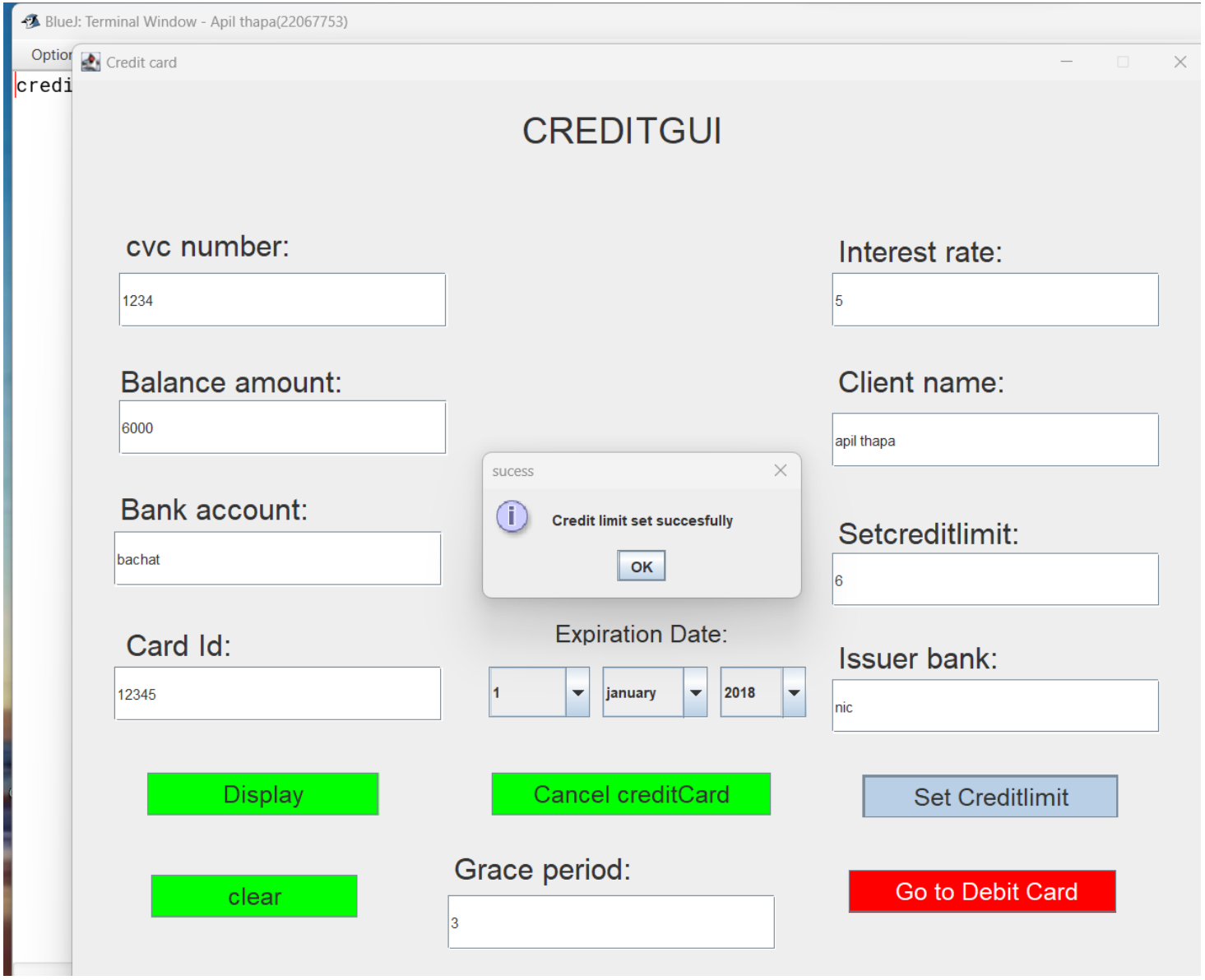

Figure 12 Screenshot of set credit limit

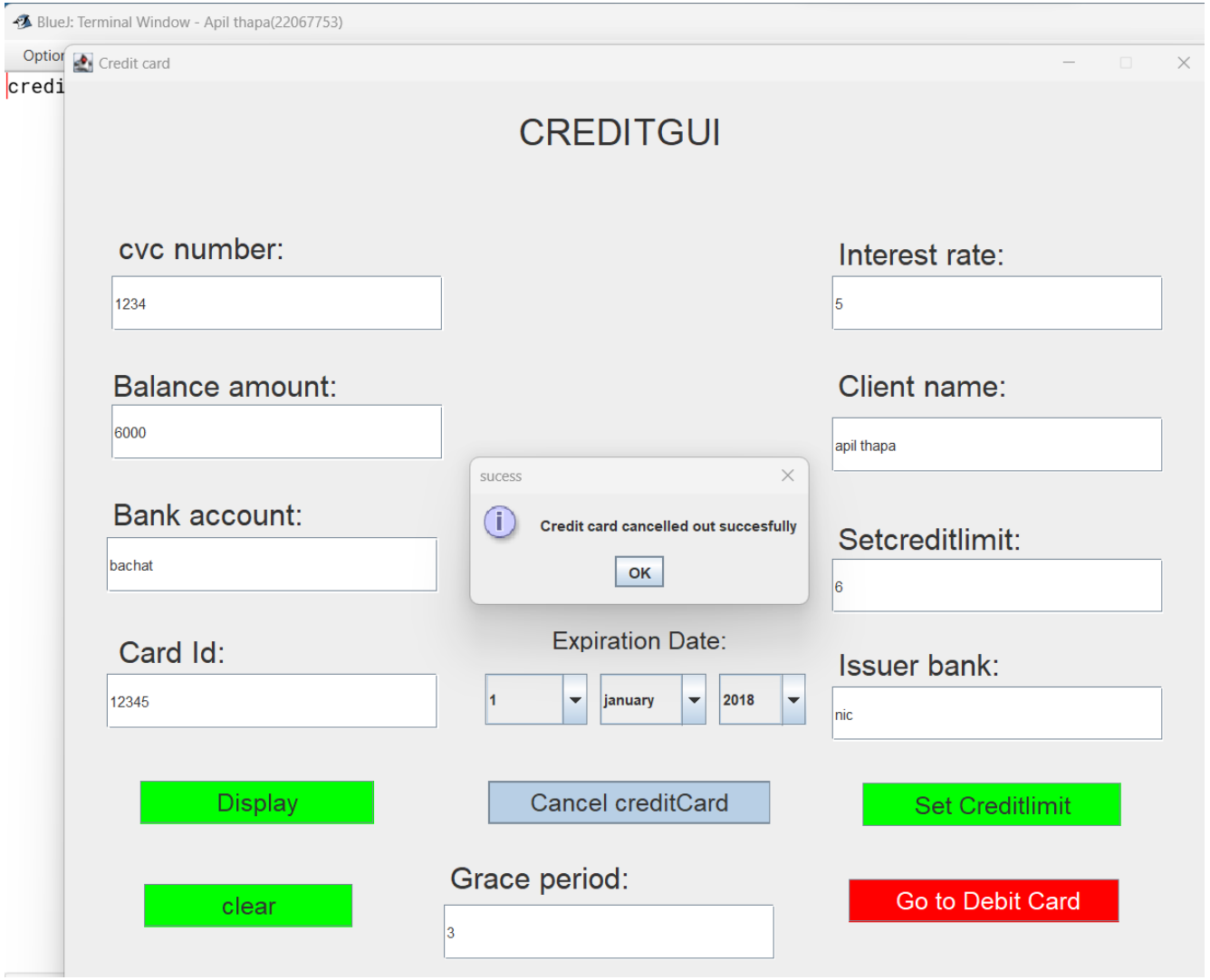

Figure 13 Screenshot of cancelled out

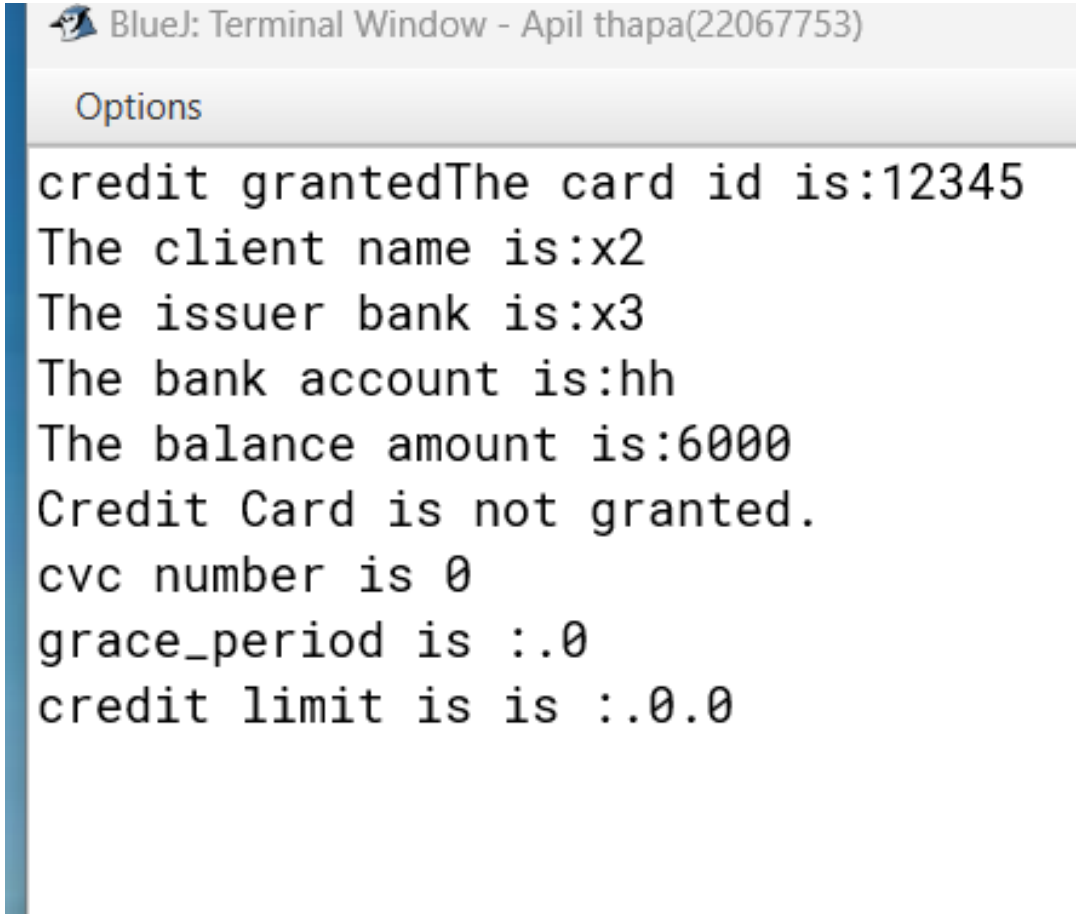

Figure 14 Screenshot of displaying details after cancelling creditcard

Table 3 Test-3: To test appropriate Dialog box when unsuitable value entered.

Test	3
Objective	To test appropriate Dialog box when unsuitable value entered.
Action	⟹ Click add debicard button with empty textfield. ⟹ Click add debitcard twice with same card id. ⟹ Click add creditcard button with empty textfield. ⟹ Click add creditcard button twice with same card id. ⟹ Click withdraw button with empty textfield. ⟹ Click withdraw button without no debit card added.

	⟹ Click withdraw button with charcter in amount textfield. ⟹ Click withdraw button with different card id from debitcard. ⟹ Click setcredit button with empty textfield. ⟹ Click setcreditbutton without credit card added. ⟹ Click setcredit button with character in grace period textfield. ⟹ Click cancel credit button with empty textfield. ⟹ Click cancel credit button without credit card added.
Expected result	They need to show their appropriate dialog boxes.
Actual result	Dialog boxes are shown.
conclusion	The test is successful.

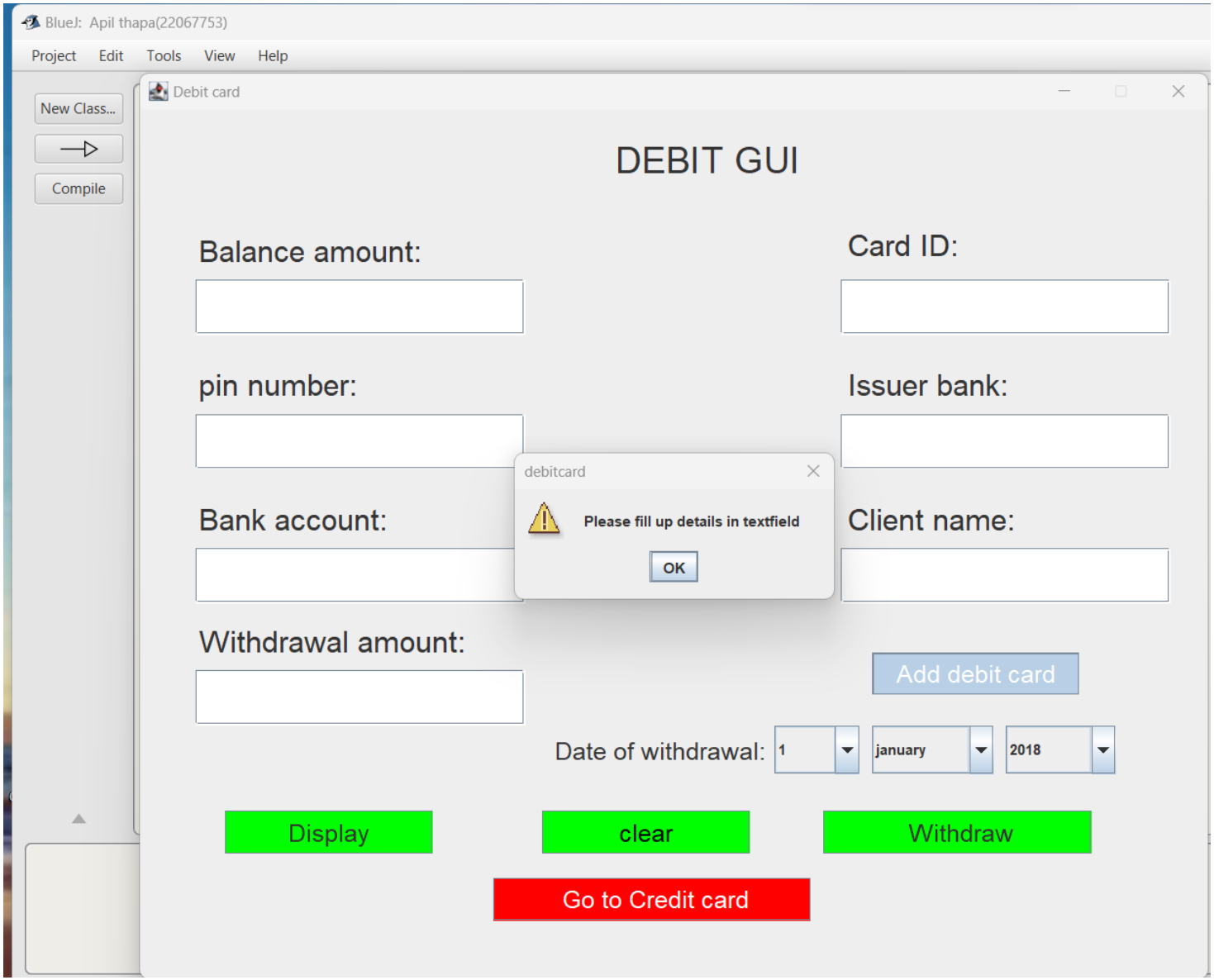

Figure 15 while adding debitcard for emptyfield

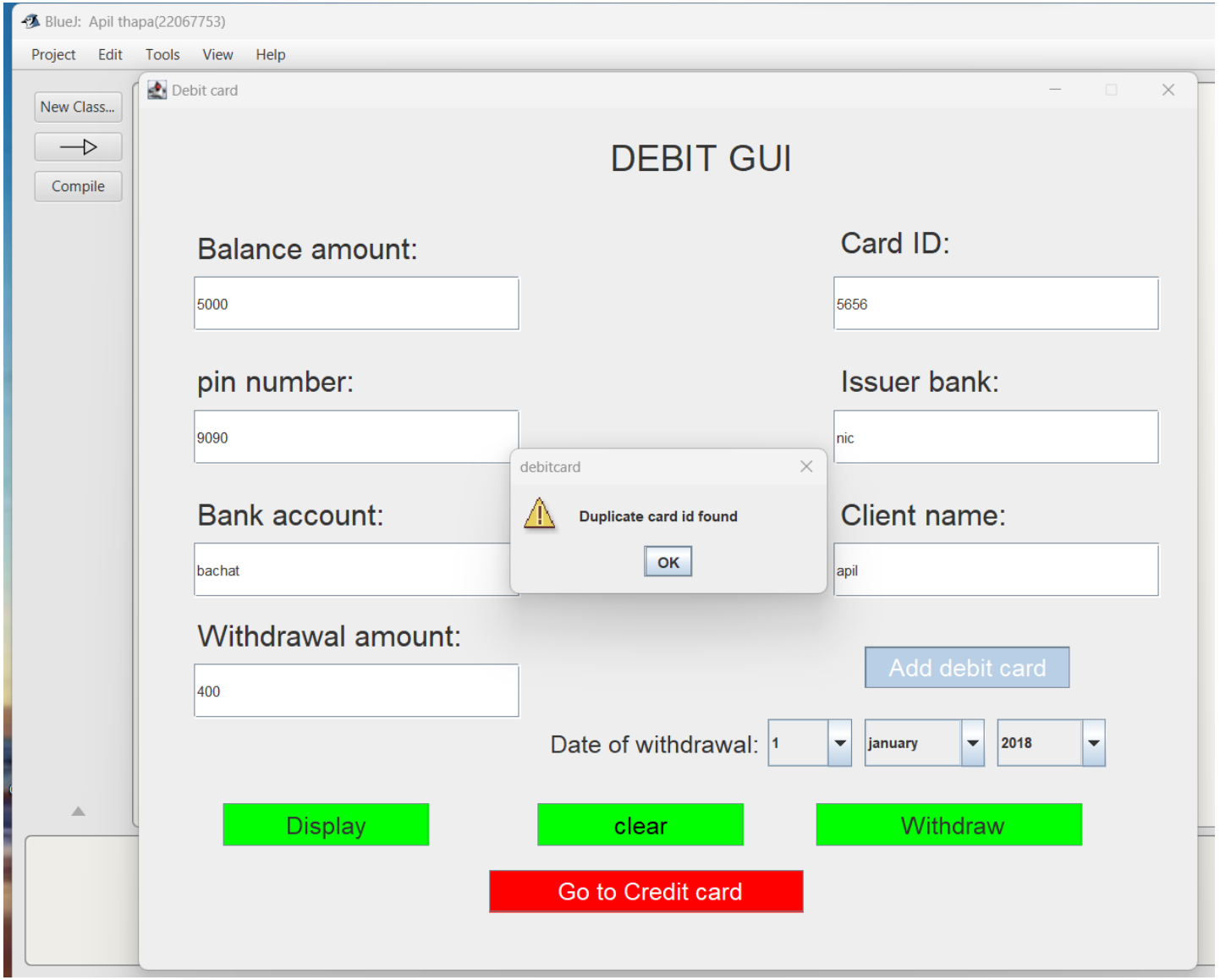

Figure 16 for same cardid debicard

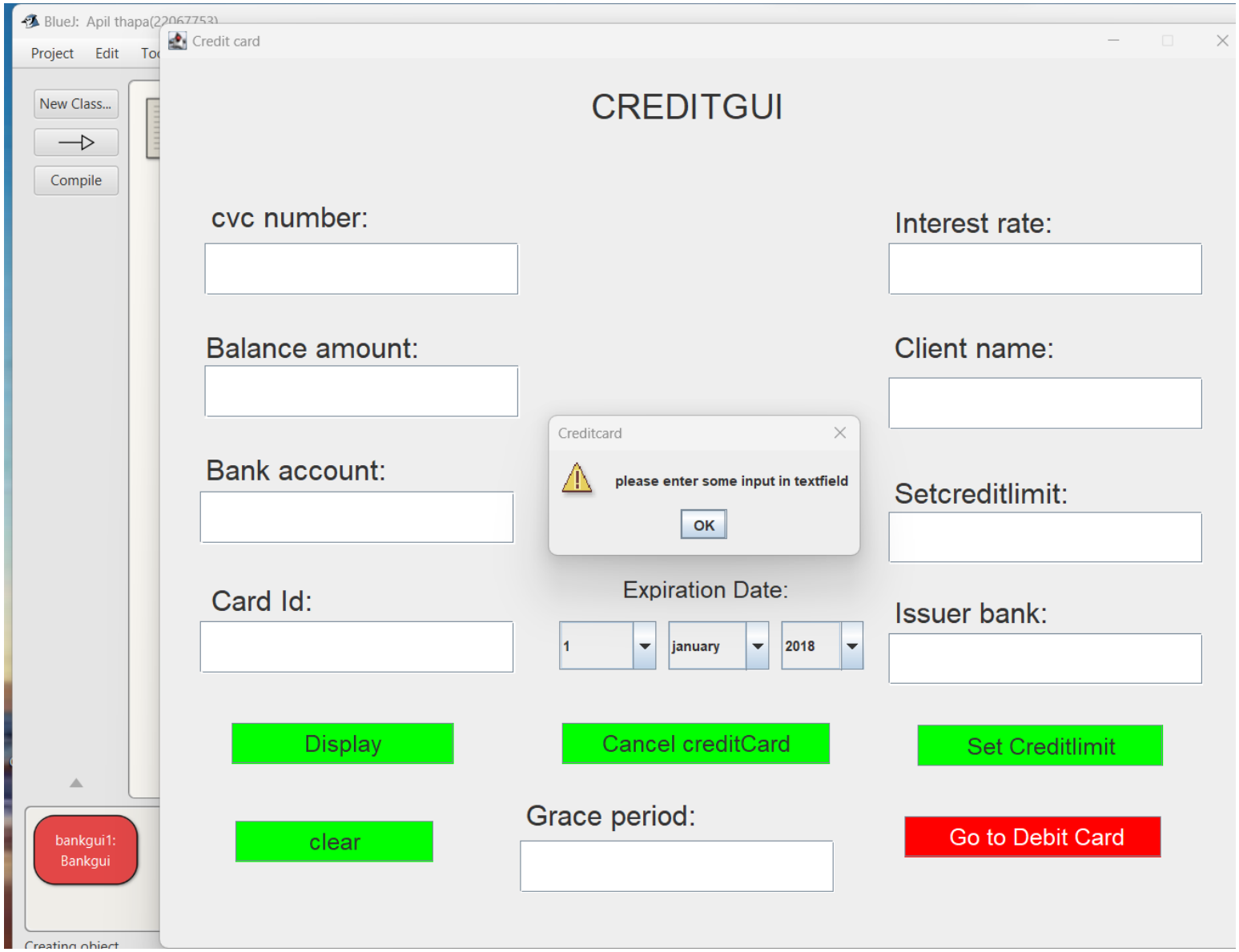

Figure 17 while adding for empty textfield creditcard

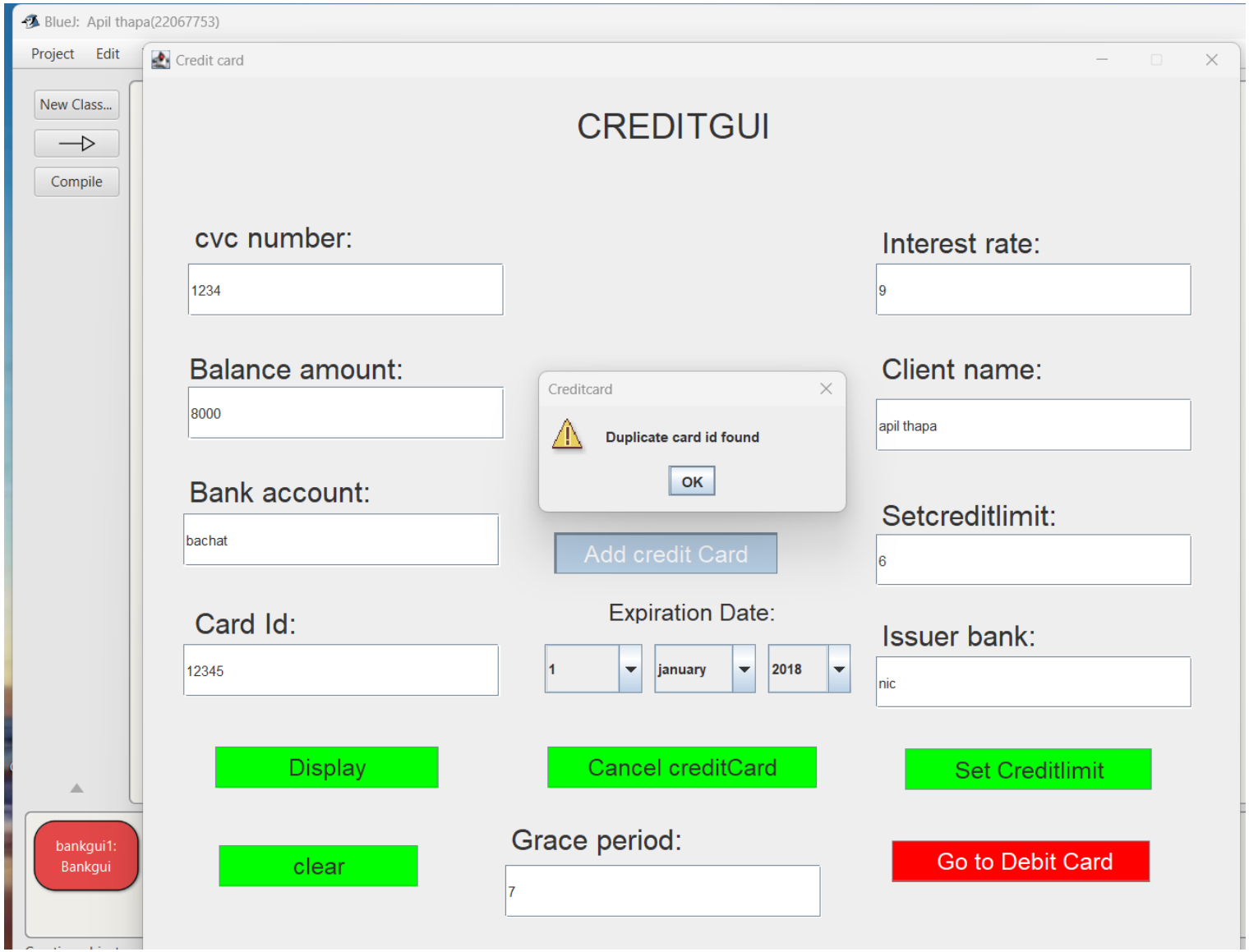

Figure 18 for same card id creditcard

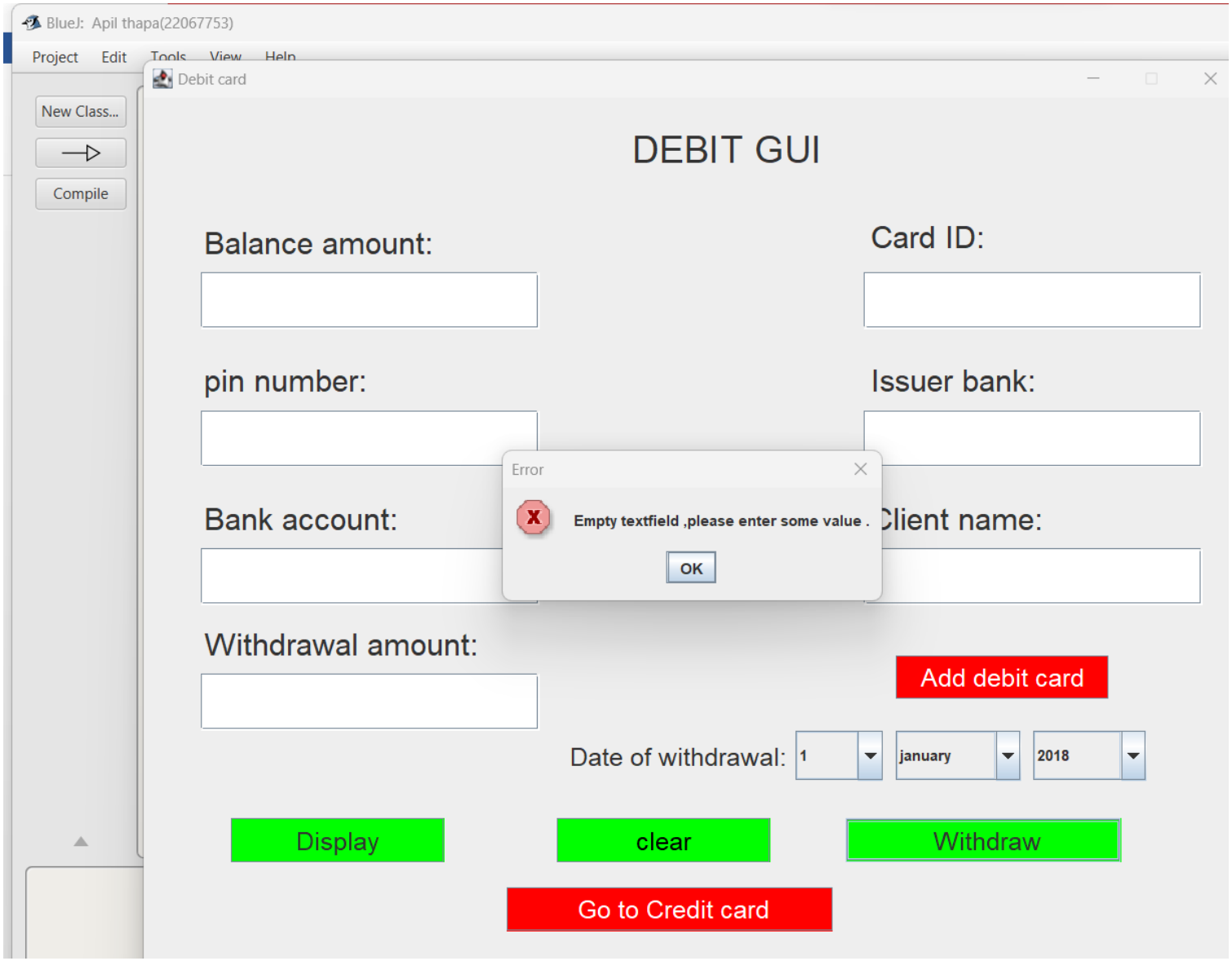

Figure 19 while withdrawing for no text value

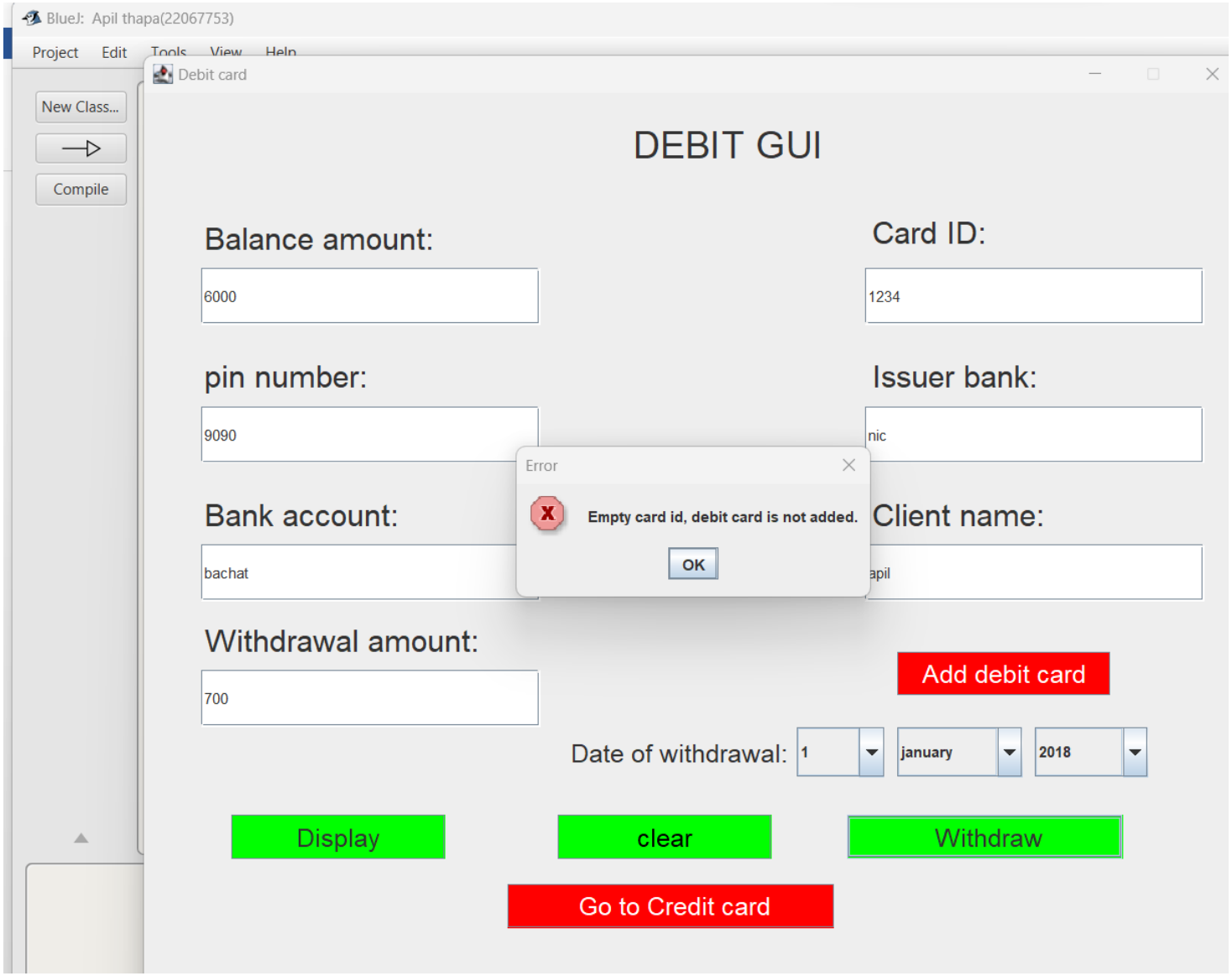

Figure 20 while withdrawing without debitcard added

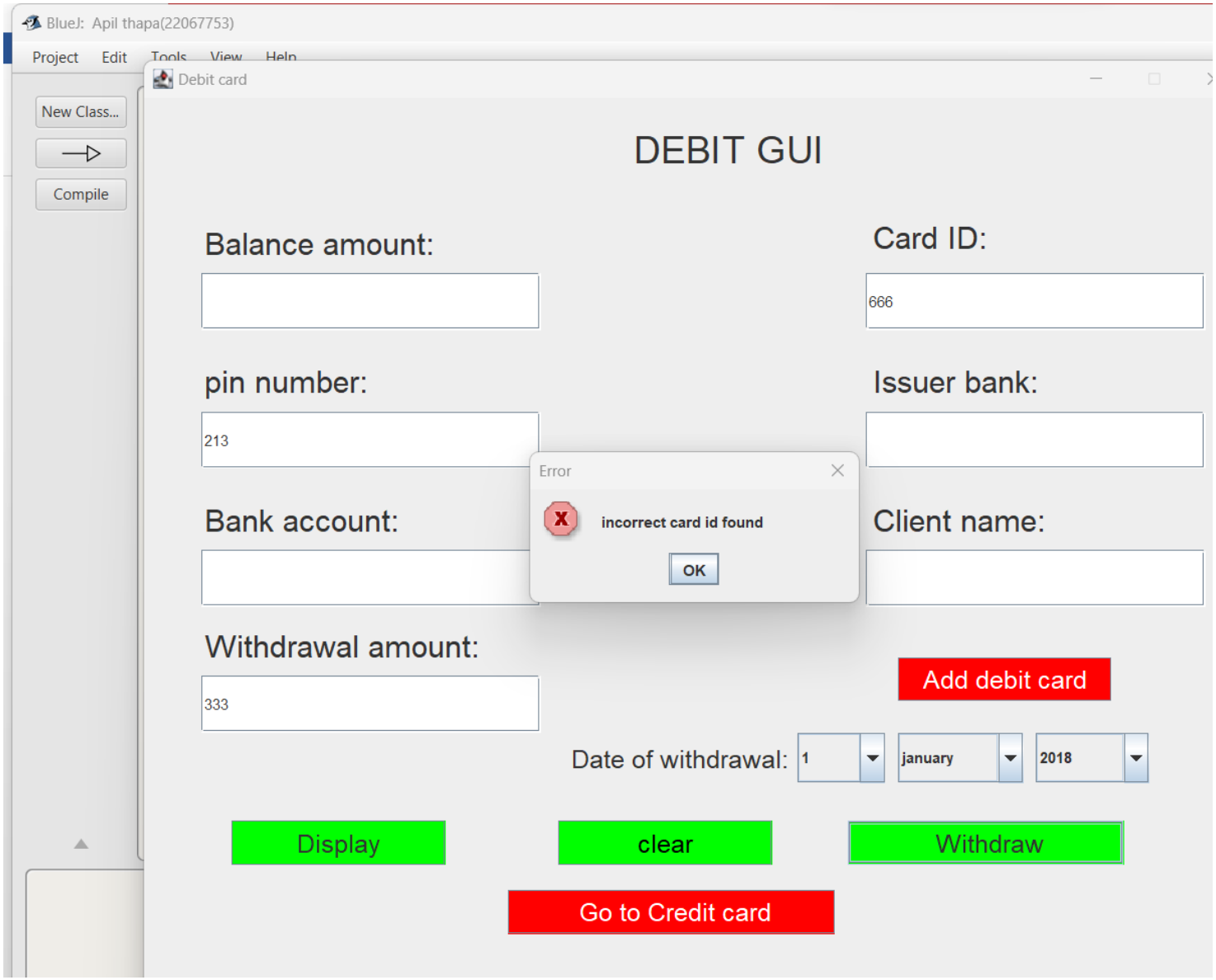

Figure 21 Card id not match case for withdraw

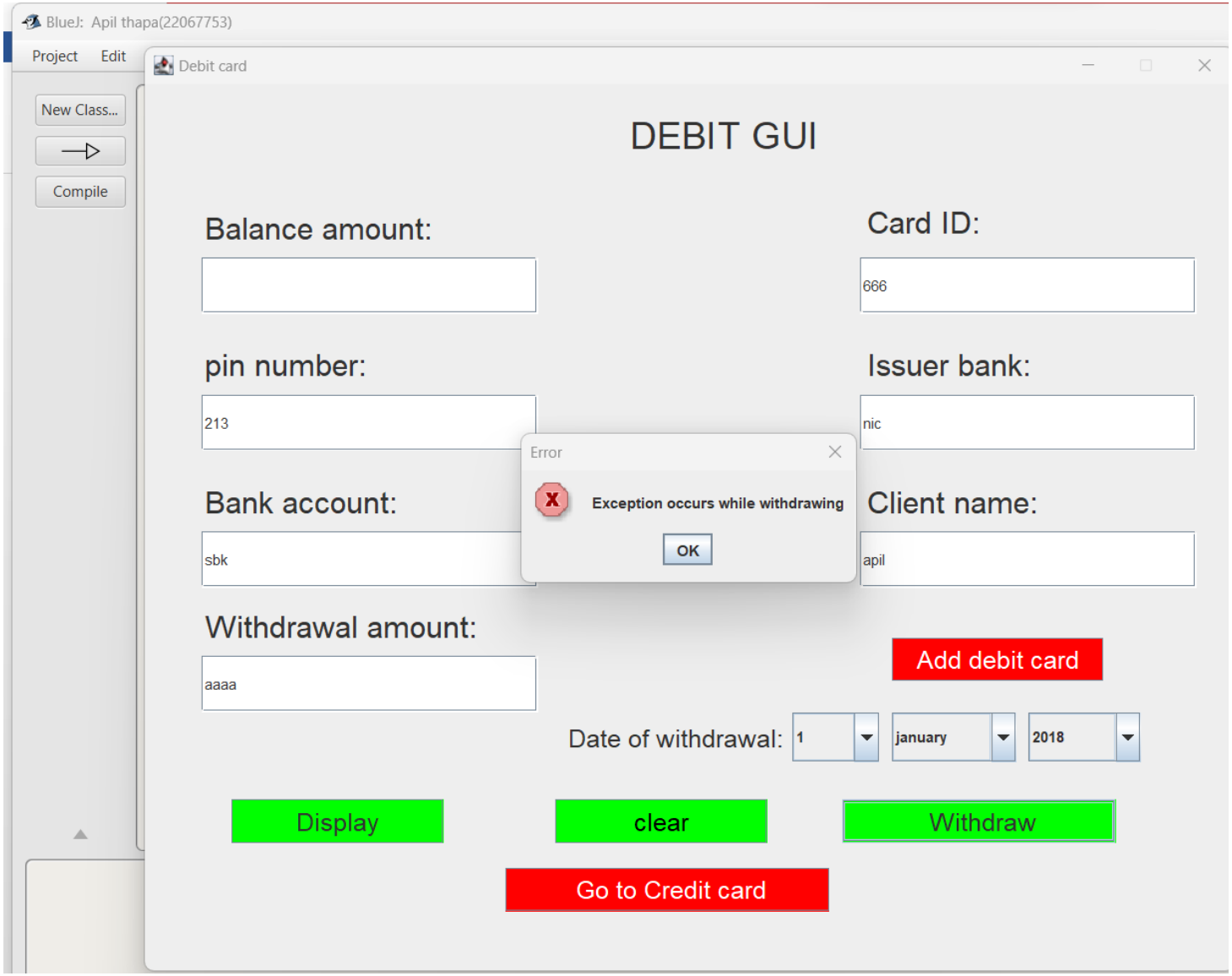

Figure 22 Exception found while withdraw

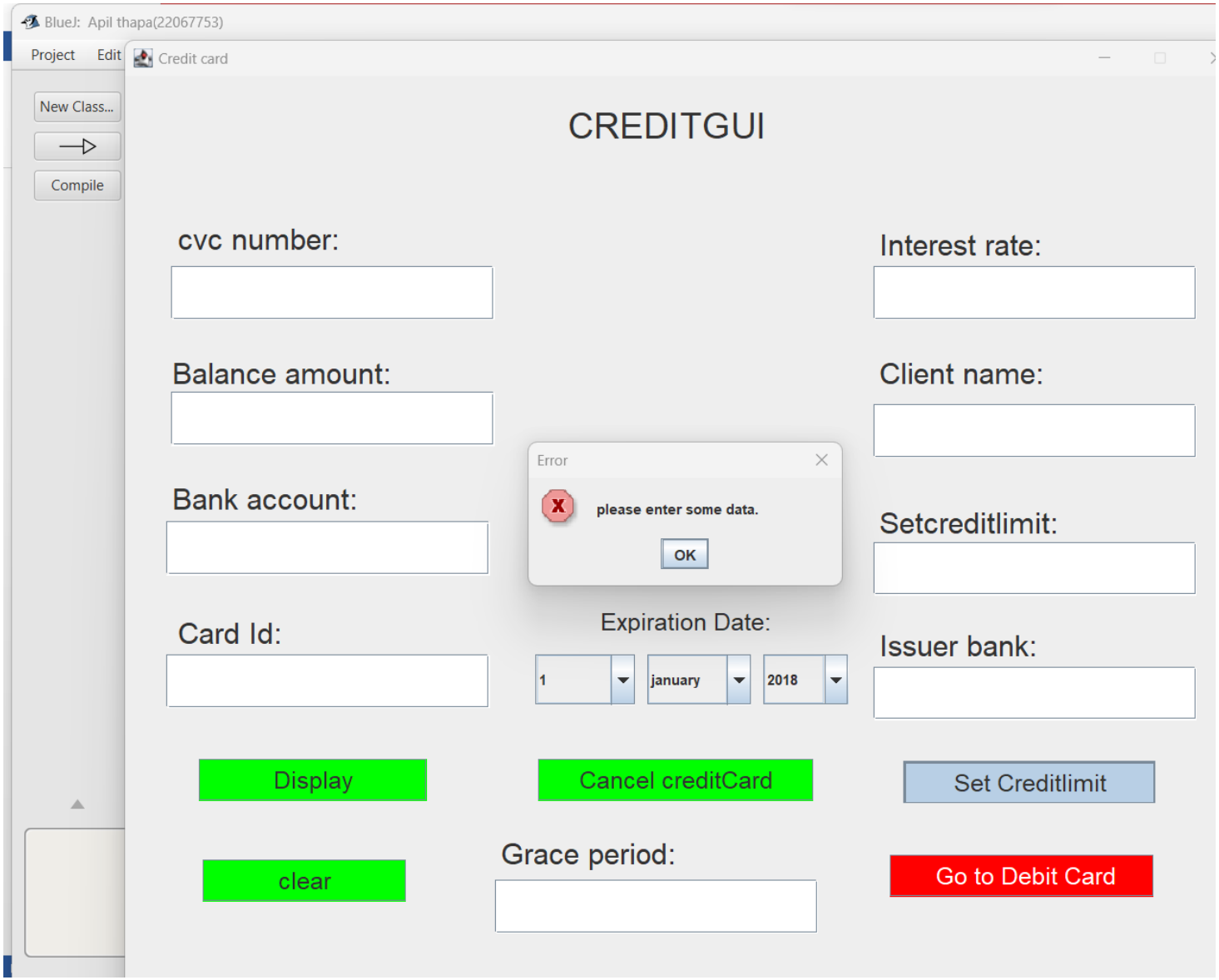

Figure 23 setcreditlimit for not textvalue

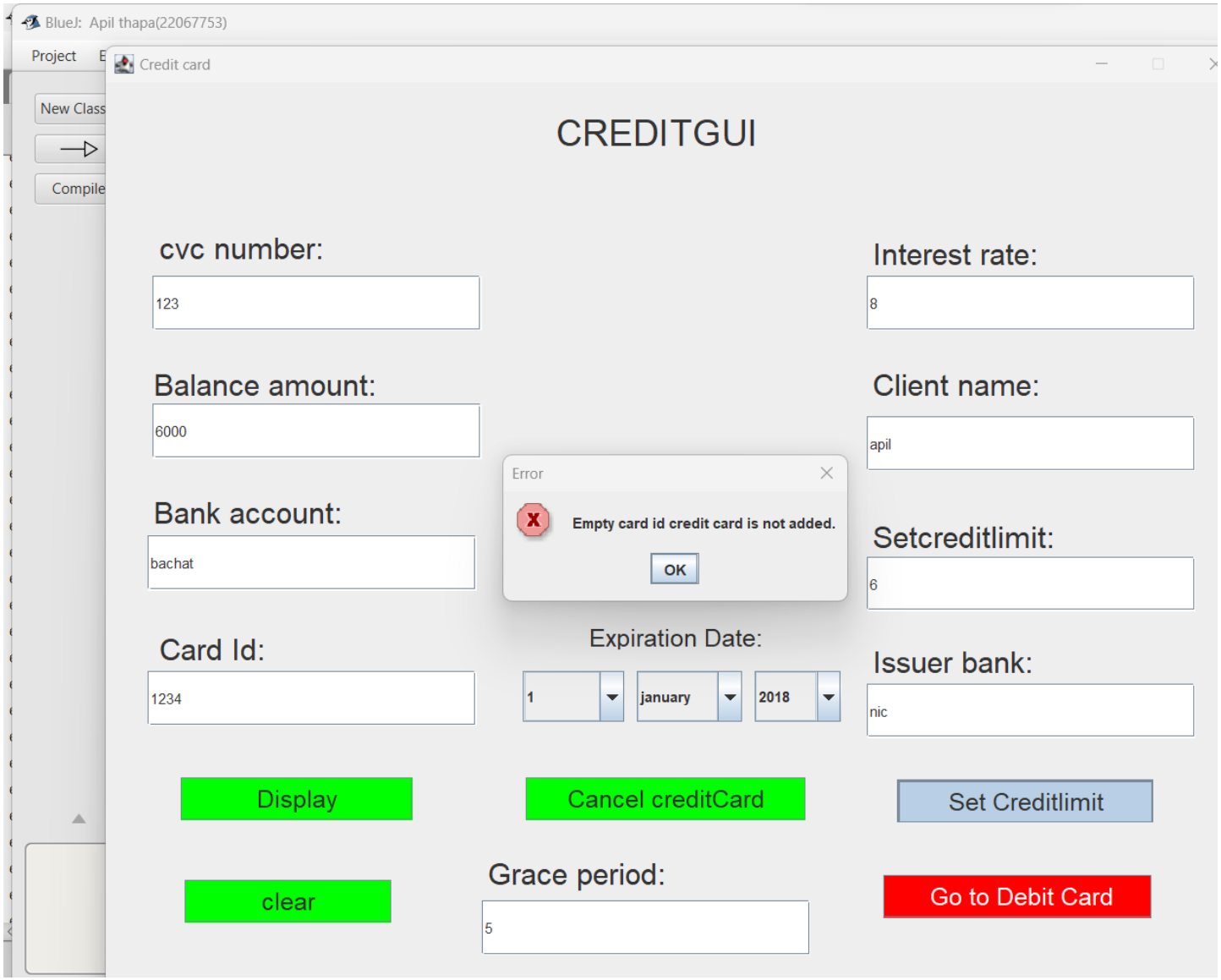

Figure 24 Setting creditlimit without adding creditcard

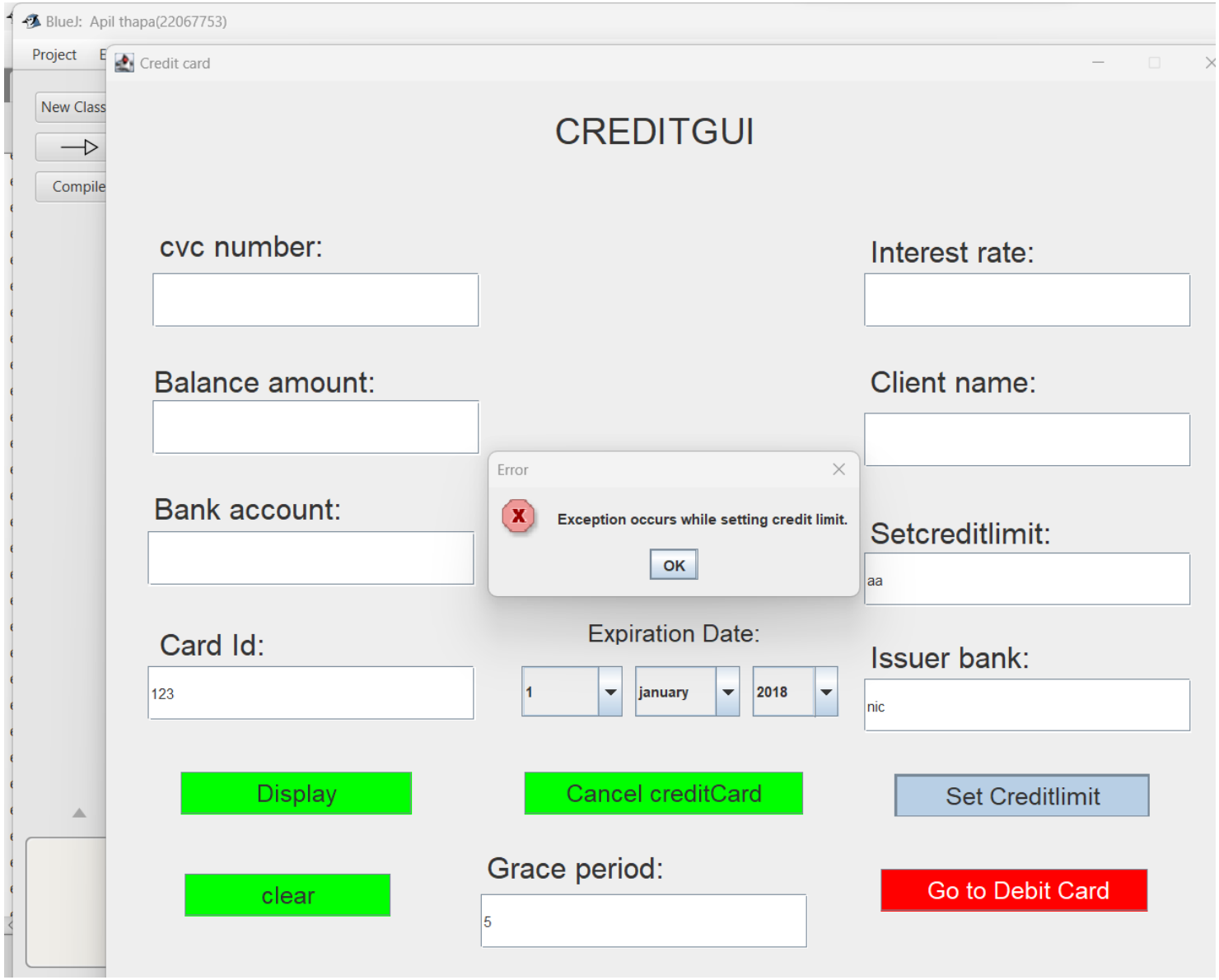

Figure 25 Exception found while setting limit

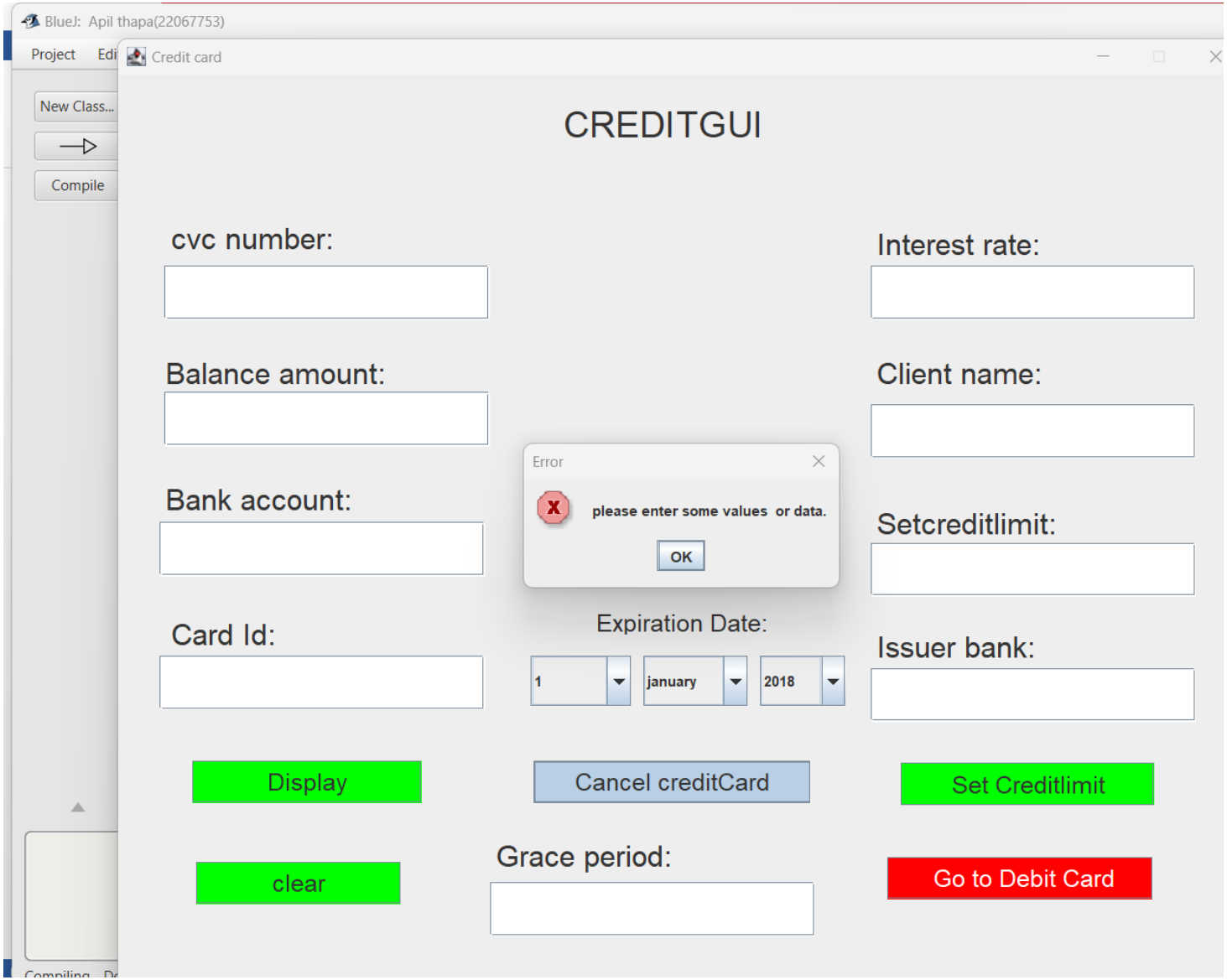

Figure 26 while cancelling for empty textfield

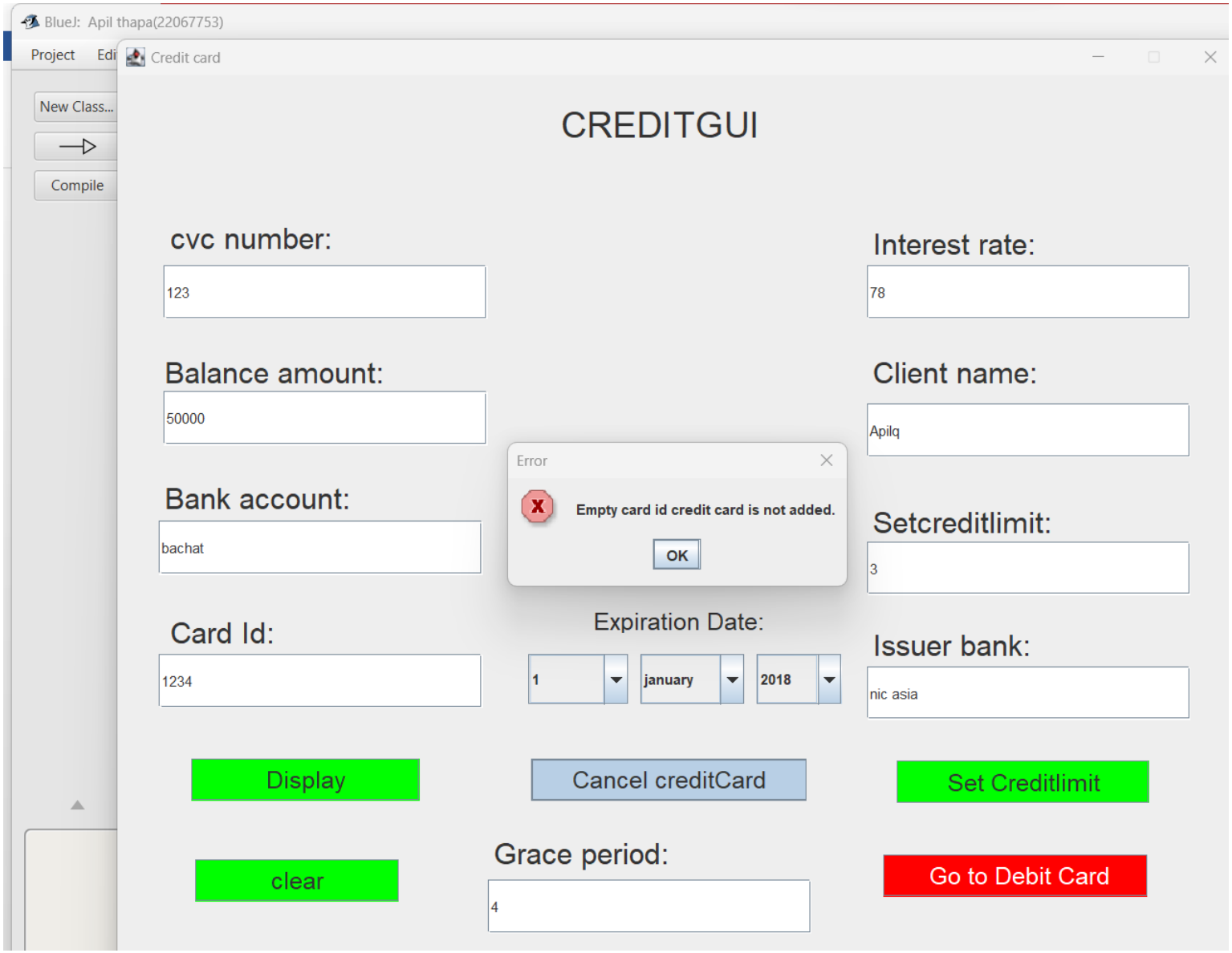

Figure 27 Cancelling creditcard without addding it

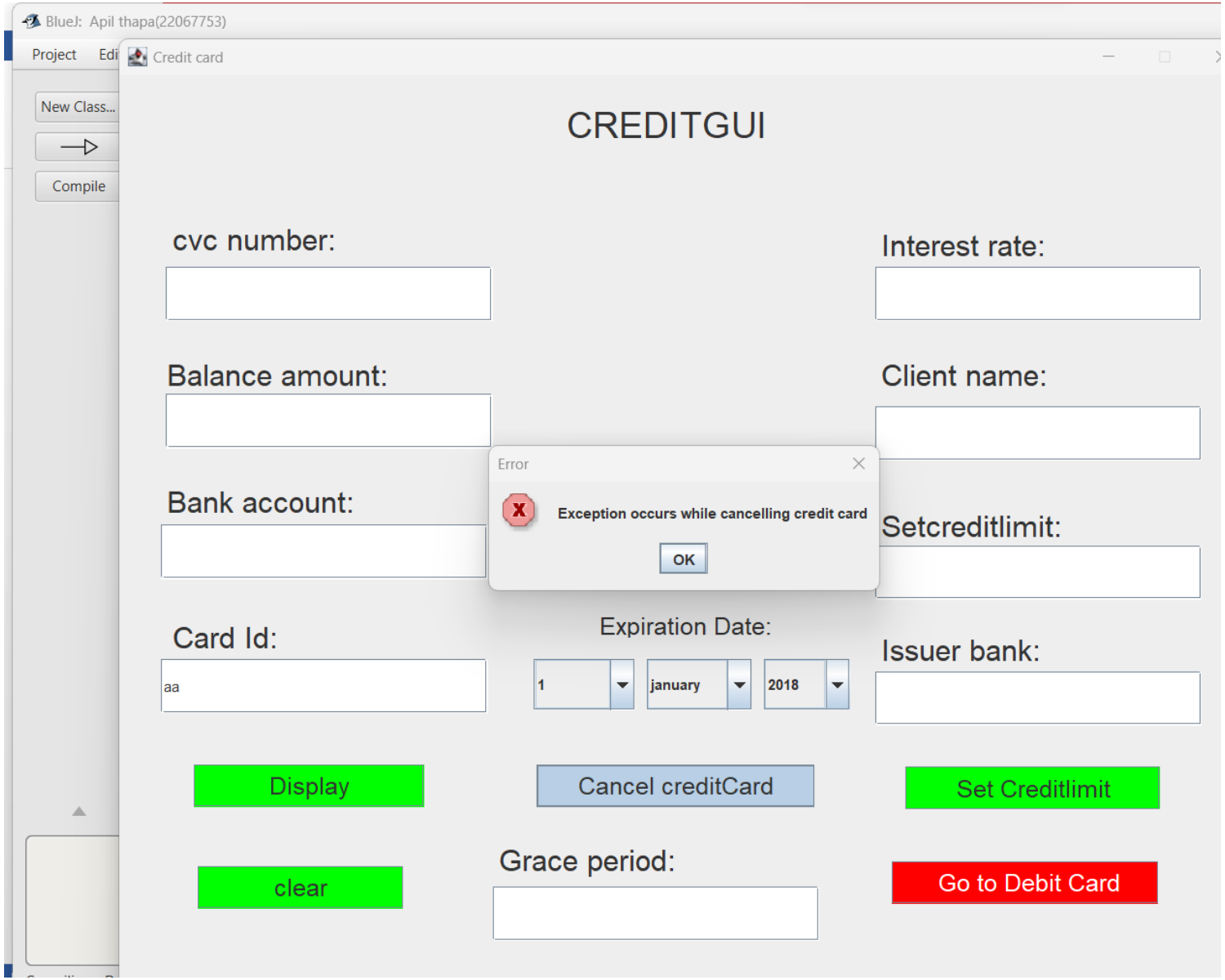

Figure 28 Exception occurs while cancellin

6. Error analysis and detection:

Simply error analysis means identifying and resolving error which occurs frequently which developing swing projects. It is one of the important part in our program to detect error and analysis to solve our problem and to generate outcomes as our expectation.generally error occurs when we code in our program we face a lot of erros if not errors then it could be exceptions and many more.error analysis and detection is an essential aspects for any field such as software development and many more. this capacity or those methods helps us to fix big issues, variety of bugs while working with big tech companies in our fucture while in other side many built in features debugging tool are used to debug those code in large aspects but still logical error can't be fix out.while working with this projects I found many errors and exceptions which occurs frequently and more frequently stilll I got to fix it out which I do. Layout issues also is my one the major problem in gui application which takes my a lot of time it generally occurs because my swing componenets are not arranged in a good order some are overlapping,some are not setvisible and soon, many errors such as syntax error which I common I had found which I solve easily later on I got to face semantic errors which takes my time a lot still I got to fix that which I do but at last not the least logical error which takes my time and my energy. logical errors generally difficult to fix out program will execute but we don't have any idea where those errors arise from and how to fix that too.also I had faced a lot of exceptions while developing this bankgui applications such as numberformat exception is common , null pointer also I had faced but which arise due to logical error in my code so I fix that but number formatexception I have handle it out using try/catch block which part exception can arise goes under try block if exception occurs it will catch by catch block and which will execute showing error message.

6.1 syntax error detection:

Syntax error generally occurs when we programmers write an code with incorrect syntax basically, this error can be detected easily by the compiler during compilation program ,while working with this swing projrct I had faced this error sometime because of my mikssing semicolon, sometime mismatched parenthesis also sometime incorrect variable names too.this error is faced by every people who code but in gui swing project this error occurs more because of lengthy code and confusing swing components referenced name. I have faced this error for my mismatched parenthesis.while doing other code I found my compiler throws error when my compiler takes at this Jlabel I found missing parameters of interest rate which shows synatx error in my program.

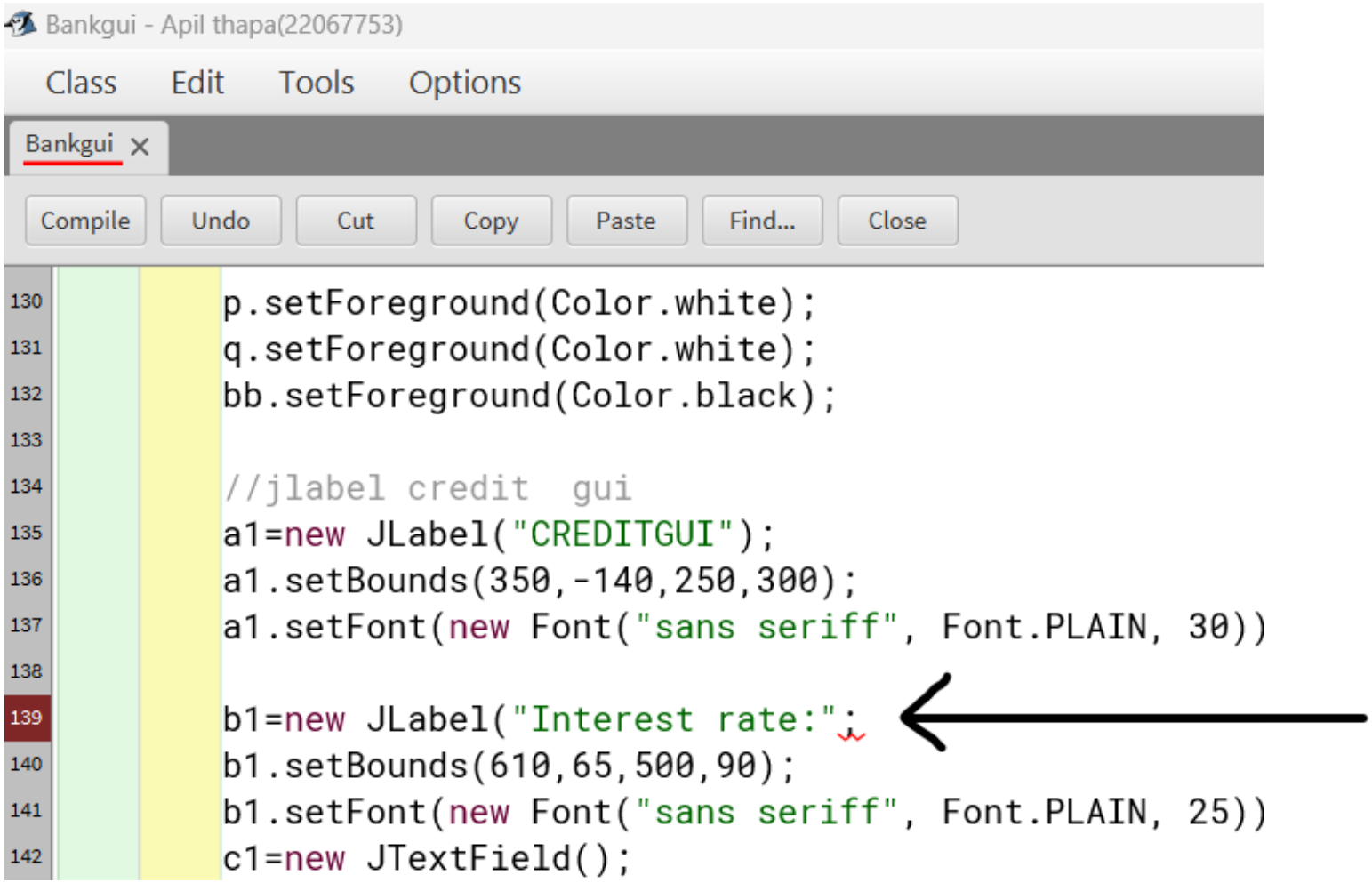

Figure 29 syntax error detection

6.1.1 Syntax error correction:

I solve this syntax error immediately after knowing by closing paranthesis of swing component Jlabels named as interest rate. Using blue j compiler it helps me to take throw immedately where error occurs from which is qute easy for me solve those errors.

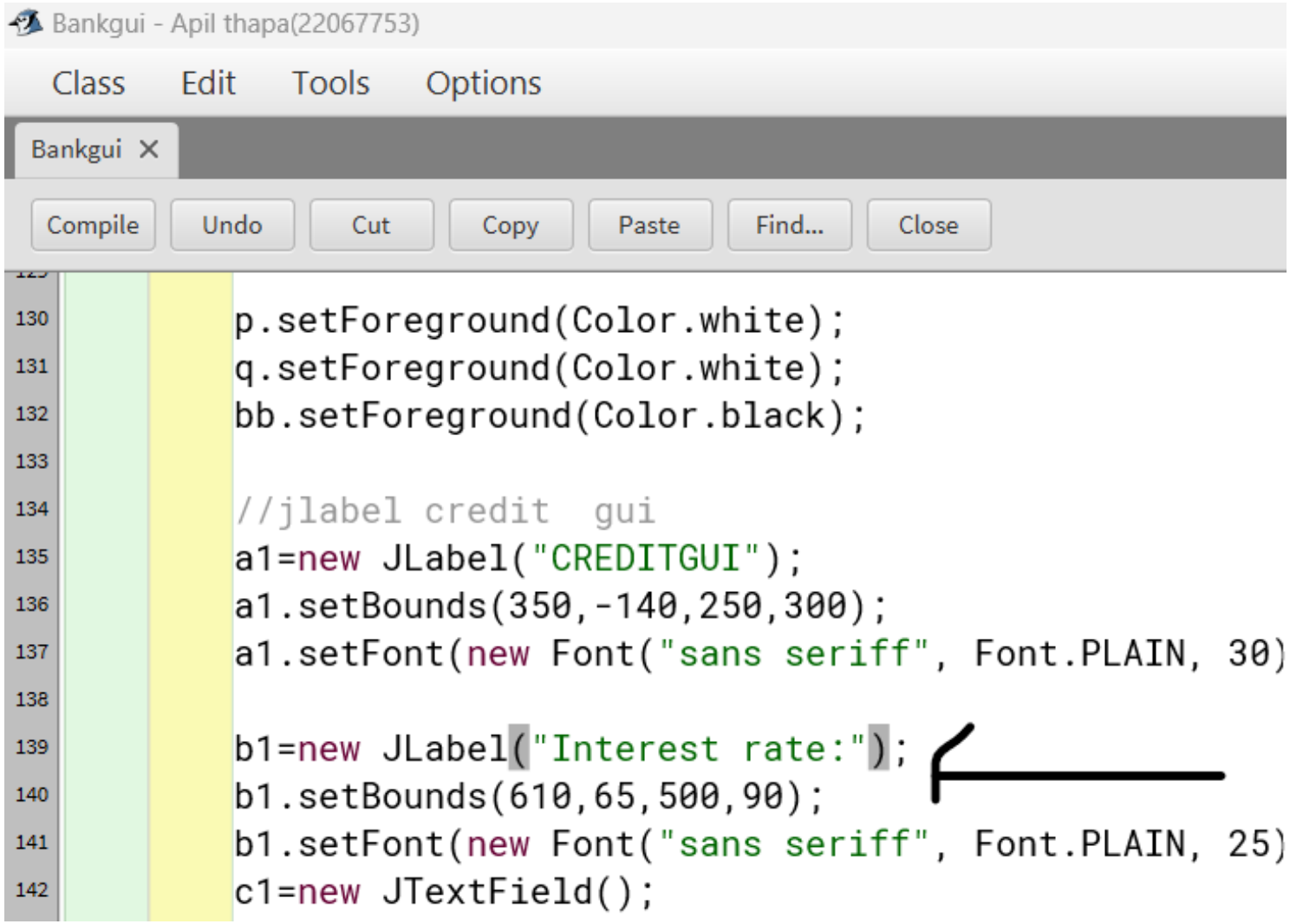

Figure 30 syntax error correction

6.2 semantic error detection:

Semantic error generally occurs when we use the wrong method, incorrectly using a variable also net setting up correct layout managers in it. Semantic error is one type of logical error also we can say program is syntactically correct but it does not make any logic In the program. I have faced this semantic error in my code in Jbutton display where I have used wrong method to set this position.each swing componenets has their own set of methods which can be used to interact with users using wrong methods can cause semantic errors instead of using setBounds I have used setLabel to set display button Jbutton position which runs in a program but it will not give output as indented which shows semantic error in the program.

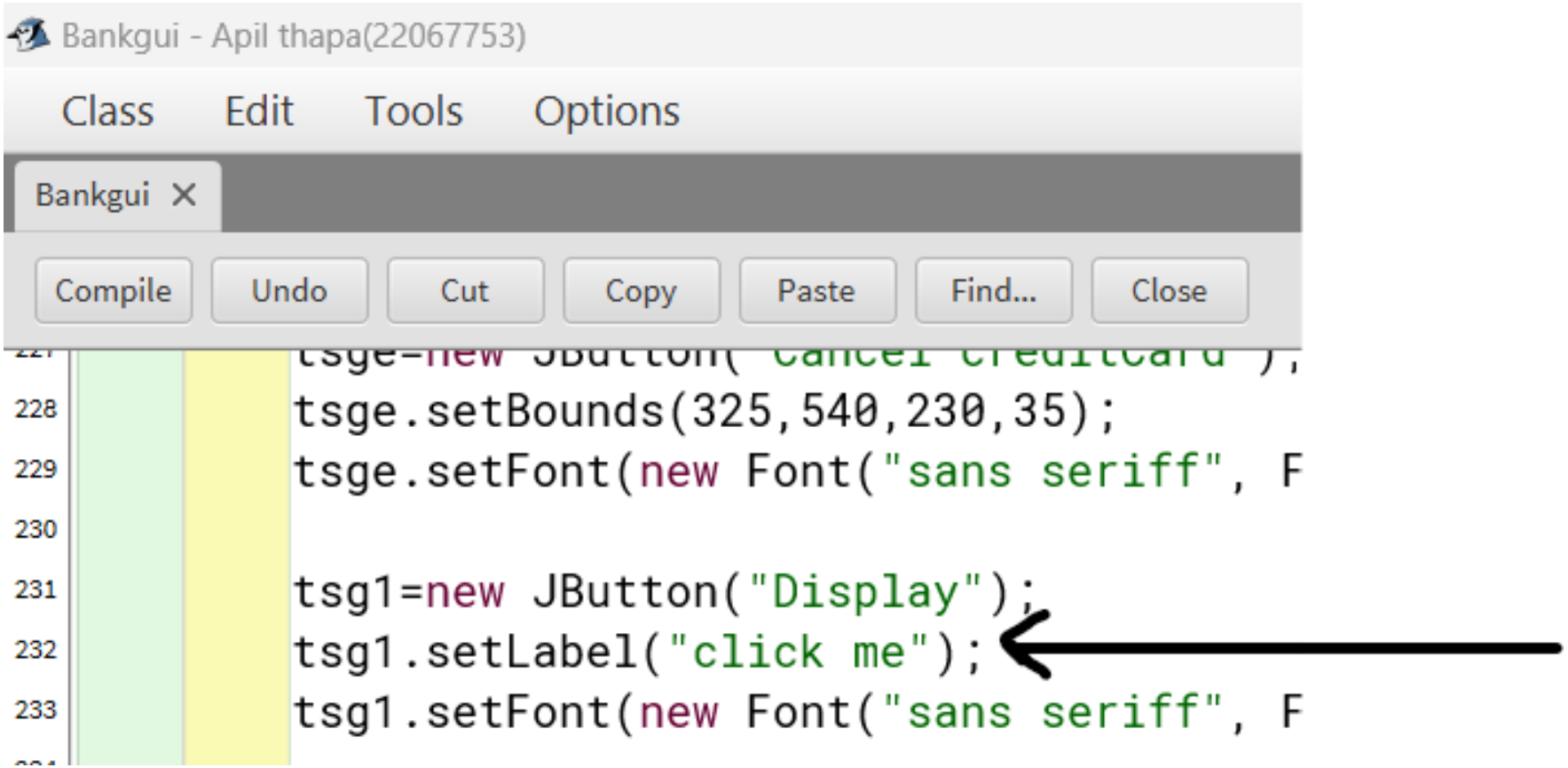

Figure 31 semantic error detection

6.2.1 Semantic error correction:

I solve this semantic error by making setBounds instead of setLabel and giving its position to it .it takes a lot of time to find out this error because this is semantic error compiler doesnot throws any error which makes more difficukt to find and solve this error.

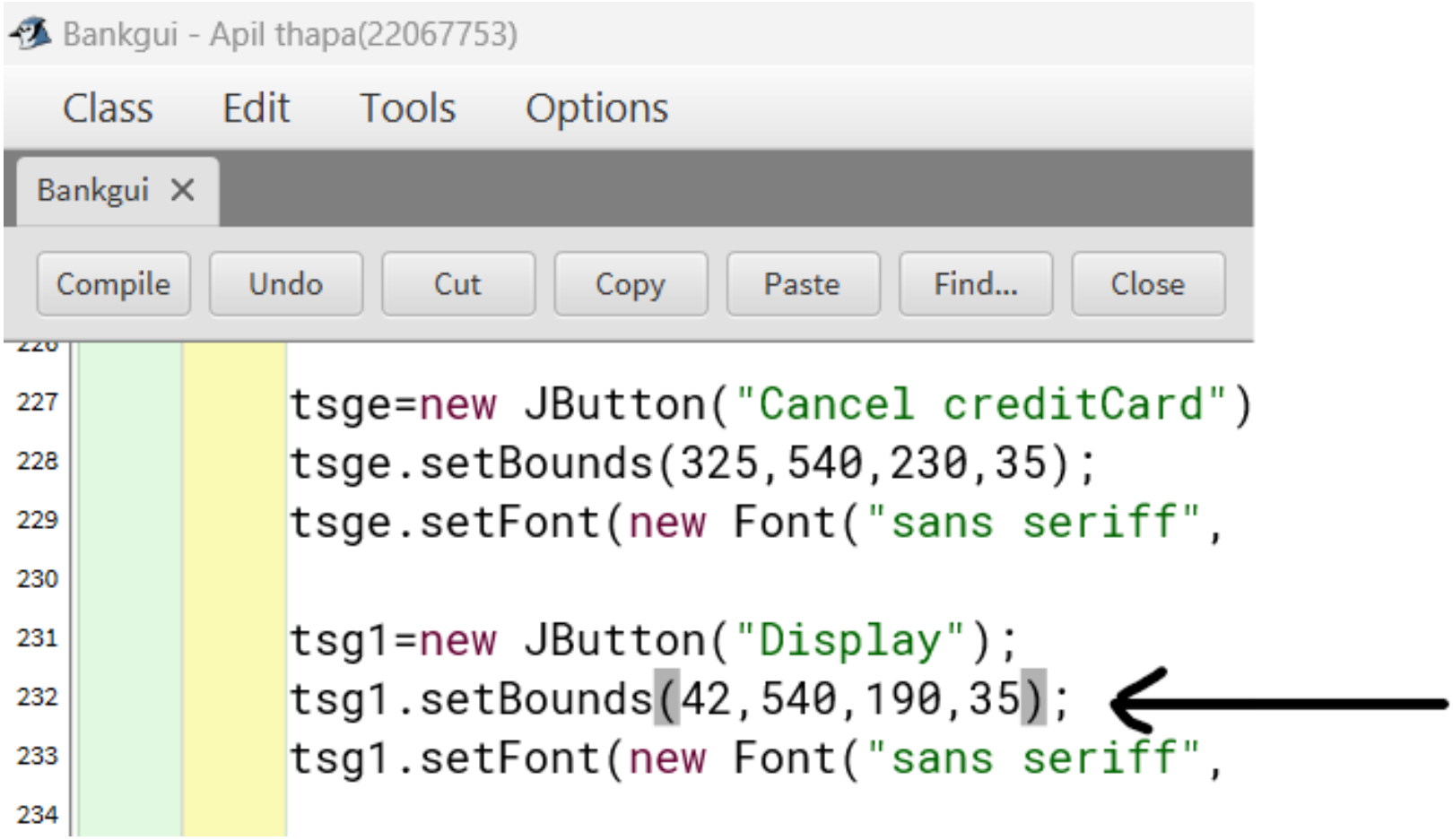

Figure 32 semantic error correction

6.3 Logical error detection:

This is a logical error in which we have to find out error by reviewing code. This is an error which is caused due to wrong logic in our code. Compiler does not support us to find out this error this error can be more diffuicukt to solve if we do not analysis the code one by one and started visualizing codes logic and algorithm.it causes unexpected results in the output .it can only be detected by running this program and observing its behaviour. Program does not behave as we expects.in our swing project this logical error arises when the withdraw button of my Bankgui class does not functions. Eventually it takes me two days to identify this error but I cant I checks all the code which I have written all inside my withdraw button but it does not functions well but my compiler runs all everything well which makes more difficult to identify this logical error in my program.

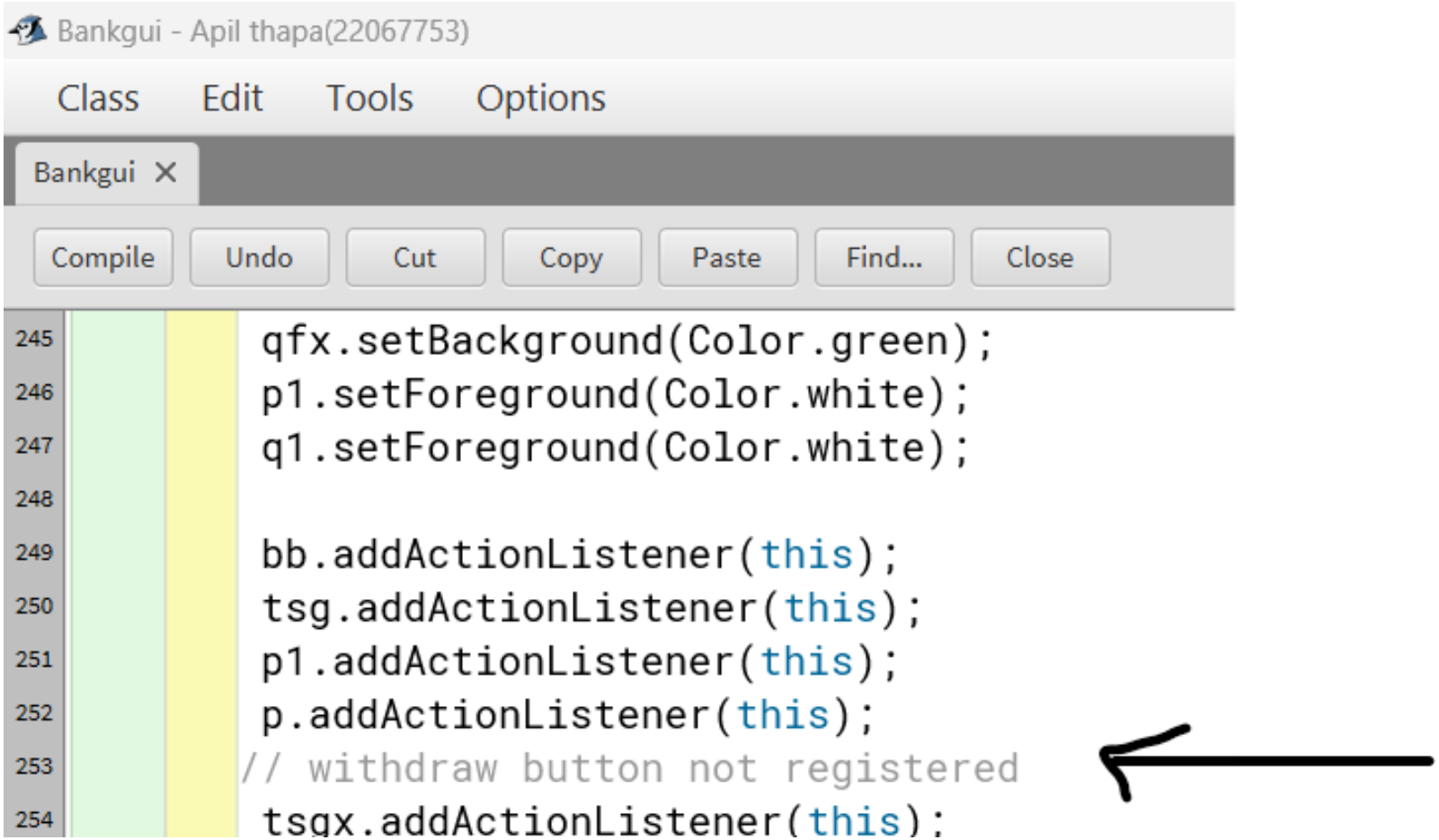

Figure 33 logical error detection

6.3.1 Logical error correction:

This error is corrected by registering the withdraw button in my code which is generally a small mistake but it effects my time as well as my whole code.

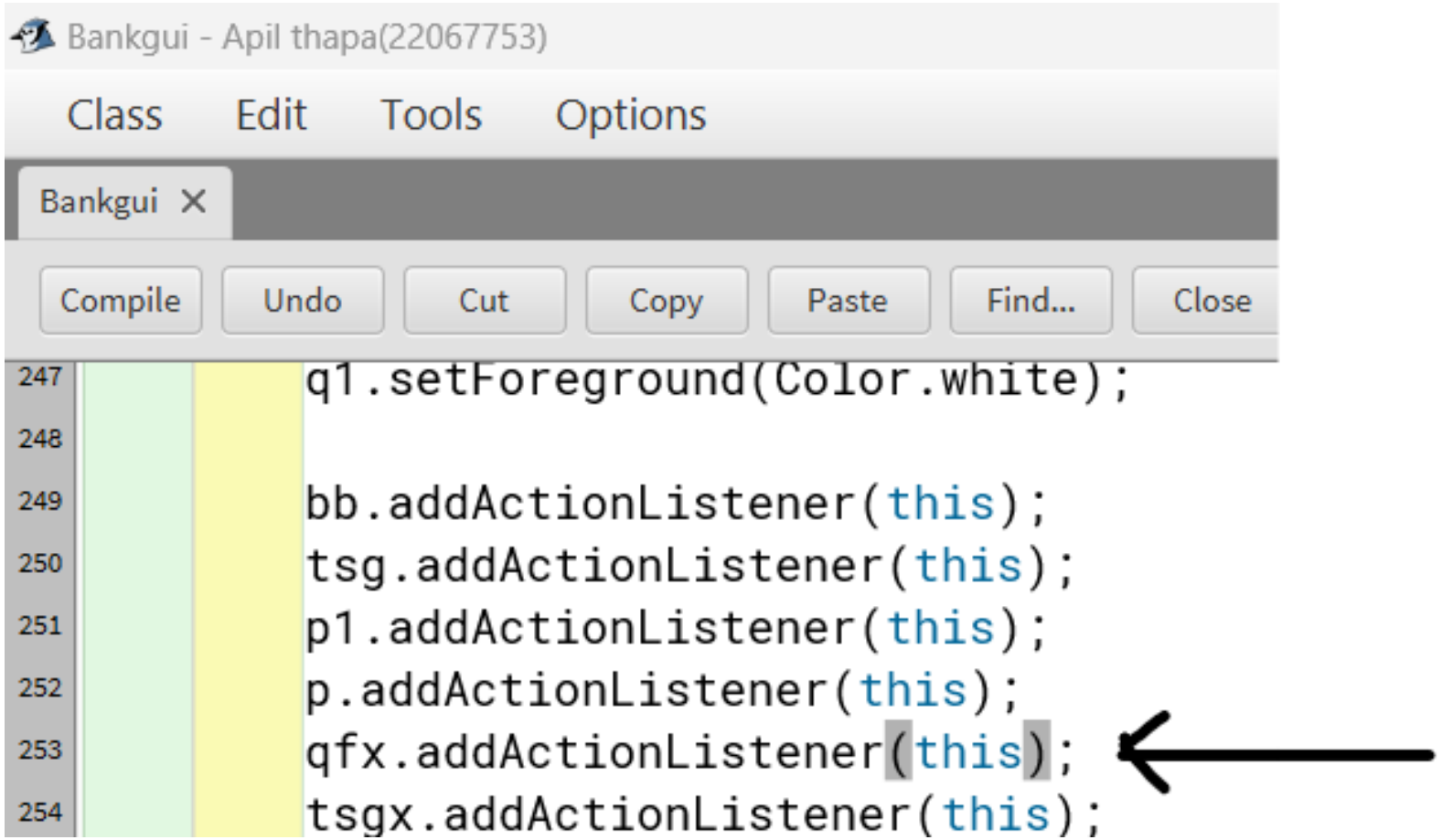

Figure 34 logical error correction

7. Conclusion:

7.1 What I have learned from this project:

First of all I have learned how the successful project looks like .how we can create an debitcard, credit card and bankcard classes using the concept of object oriented programming.how can we use inheritance , polymorphism and other oops features used in our recent projects of last semester which is in command line interface. but in this semester we have learned that how to work with GUI graphical user interface, how actually it looks like how can we creates many swing components like jLabel, jpanel, jButton, jcombobox, where we have multiple components added to panel then frame is displayed over interface with which user will interact.event handling how user will feel comfortable with GUI.i have learned how message box is displayed over screen when user press any button. mainly I have learned how to face every type of errors in program syntax error which is caused generally due to wrong syntax but symantic and logical error are difficult to identify in which compiler does not throw any error but there is error in our logic or logic of code which can only known by looking or reviewing all code with good analysis.exception handling many exception occurs while running code which will stop the execution of program.for this to avoid stopping of program we use try,catch block those code which can throw exceptions will be put under try block and in case exception occurs catch block will catch it and shows messages which will not affect program flow.

7.2 Difficulties I have faced:

while developing something unique which we haven't done before is itself a great task difficult times are at some level after that we can fix those things and develop it which we think.at first we need to have some target of developing what we want then we need to find out necessary syntax , algorithms which is essential for that program to develop.As we are developing GUI application about banking system.for which we have faced a lot of problems like facing a lot of errors dealing up with the different exception while running program.firstly while dealing up with the variety of swing components texfield overlapping jLabel invisibility button disappearing setlayout invisibility panels visibility false and many more frame problems we have encountered while developing GUI.while adding event listener else while doing event handling firstly we have encountered with syntax error and fix it after that symantic error which is more tough than syntax error but still we fix that with reviewing of code again also same logical error which were not easy to solve but still we got to their solution by careful analysis of code and reviewing of our code.faced a lot of exception like number format exception,null pointer exception which is caused due to logical errors in the code.solge those exception using try catch block try contains code which can throws some exceptions while execution of program and catch block will catch those exception and runs code inside that catch block if exception occurs which avoid unnecessary stopping of flow of program.many days I have wasted my time on event handling and to understand logical error and solution to fix it.my swing components kill my time and getting value from combobox wasted my time too

7.3 How we solved those problems:

Many difficulties I have faced while making this project I have mention at top what problems I had faced and what kind of different types of errors I had faced while doing this project. I finally solve those errors like syntax errors through debugging and compiling which were very easy to identify but for semantic and logical errors compiler does not throw any kind of errors which is a task for me to identify those errors. still we fix that with reviewing of code again also same logical error which were not easy to solve but still we got to their solution by careful analysis of code and reviewing of our code.faced a lot of exception like number format exception,null pointer exception which is caused due to logical errors in the code.solge those exception using try catch block try contains code which can throws some exceptions while execution of program and catch block will catch those exception and runs code inside that catch block if exception occurs which avoid unnecessary stopping of flow of program. while dealing up with the variety of swing components texfield overlapping jLabel invisibility button disappearing setlayout invisibility panels visibility false and many more frame problems we have encountered while developing GUI.while adding event listener else while doing event handling firstly we have encountered with syntax error and fix it after that symantic error which is more tough than syntax error but still we fix that with reviewing of code again also same logical error which were not easy to solve but still we got to their solution by careful analysis of code and reviewing of our code after great analysis of code for many times and debugging I solved those errors and for exceptions try to avoid such exceptions imnitially but if there is the any chance for the occurance of exception then try/catch block is implemented in those program.

8. References

Ballew, j., 2021. *ms-word.* [Online]
Available at: https://www.lifewire.com/microsoft-word-4159373
[Accessed 1 may 2023].

gosling, j., 20 september 2022. *blue-j.* [Online]
Available at: https://www.bluej.org/
[Accessed 18 january 2023].

JGraph Ltd, A. H. 7. Q. N. 7. N. E. C. #., 2005. *diagram.net.* [Online]
Available at: https://www.diagrams.net/doc/
[Accessed 1 may 2023].

9. Appendix.

```
import javax.swing.*;
import java.awt.Font;
import java.awt.Color;
import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;
import java.util.ArrayList;

public class Bankgui implements ActionListener
{
    private JFrame frame,frame2;
    private JPanel panel1,panel2;
```

```
    private JLabel
j1,j2,nu,j3,j4,j5,a,b,d,f,h,j,a1,b1,f1,n1,x1,l1,d1,h1,x,l,n,app,app1,ap1,setcr
edit,grace; //adding frame

    private JTextField
i1,k1,c,e,g,i,c1,e1,g1,m1,m,k,kk,kk1,setcredit1,period;//adding textfield
added into panel

    private JButton o,og,og1,p1,tsg,tsg1,q1,p,q,bb,bb1,qfx,tsgx,tsge;
/*adding button,combobox,label into panel which is added into frame*/

    private JComboBox<String> oa,oaa,oo,o4,o2,o3;

    private ArrayList<Bankcard> card=new
ArrayList<Bankcard>();//creating arraylist of bankcard class

    private Integer
xbn,ynv,xbm1,y1,x5,x6,y7,gracy,withdrawalamount;//getted variables
from action performed method

    private String x3,x2,hh,y5,y6,llll,x7,yj1,setcreditl,date1,date;

    private Debitcard obj;//debitcard obj
```

```
private Creditcard obj1;//creditcard obj

public Bankgui()
{

    frame=new JFrame("Debit card");
    frame.setBounds(400,50,890,750);

    panel1=new JPanel();
    panel1.setBounds(40,30,900,650);


    frame2=new JFrame("Credit card");
    frame2.setBounds(350,17,950,800);
```

```
panel2=new JPanel();
panel2.setBounds(20,30,900,700);

//jlabel debit gui code
a=new JLabel("DEBIT GUI");
a.setBounds(350,-140,250,300);
a.setFont(new Font("sans seriff", Font.PLAIN, 30));
Color mycolor=new Color(102, 232, 189 );

b=new JLabel("Card ID:");
b.setBounds(540,40,500,80);
b.setFont(new Font("sans seriff", Font.PLAIN, 25));
```

```
c=new JTextField();
c.setBounds(6,220,270,45);  //pin no/////////////////////////////////////

d=new JLabel("Balance amount:");
d.setBounds(8,40,270,90);
d.setFont(new Font("sans seriff", Font.PLAIN, 25));
e=new JTextField();
e.setBounds(6,110,270,45);

f=new JLabel("Client name:");
f.setBounds(540,260,150,90);
f.setFont(new Font("sans seriff", Font.PLAIN, 25));
g=new JTextField();
g.setBounds(535,220,270,45);//issuer bank
```

```
h=new JLabel("Issuer bank:");
h.setBounds(540,150,150,90);
h.setFont(new Font("sans seriff", Font.PLAIN, 25));
i=new JTextField();
i.setBounds(535,330,270,45);//client name

x=new JLabel("Bank account:");
x.setBounds(8,260,190,90);
x.setFont(new Font("sans seriff", Font.PLAIN, 25));
k=new JTextField();
k.setBounds(6,330,270,45);//bank account

app=new JLabel("Withdrawal amount:");
```

```
app.setBounds(8,360,290,90);
app.setFont(new Font("sans seriff", Font.PLAIN, 25));
kk=new JTextField();
kk.setBounds(6,430,270,45);

l=new JLabel("pin number:");
l.setBounds(8,150,290,90);
l.setFont(new Font("sans seriff", Font.PLAIN, 25));
m=new JTextField();
m.setBounds(535,110,270,45);

n=new JLabel("Date of withdrawal:");
n.setBounds(300,400,588,190);
n.setFont(new Font("sans seriff", Font.PLAIN, 20));
```

```
String day[]={"1","2","3","4","5","6","7","8","9","10","11","12","13","14","15","16","17","18","19","20","21","22","23","24","25"};
o2=new JComboBox<String>(day);
o2.setBounds(480,475,70,40);

String day1[]={"january","februrary","march","april","may","jun","july","august","september","october","november","december"};
o3=new JComboBox<String>(day1);
o3.setBounds(560,475,100,40);

String day2[]={"2018","2019","2020","2021","2022","2023","2024","2025"};
```

```
o4=new JComboBox<String>(day2);
o4.setBounds(670,475,90,40);

p=new JButton("Add debit card");
p.setBounds(560,415,170,35);
p.setFont(new Font("sans seriff", Font.PLAIN, 20));

q=new JButton(" Go to Credit card");
q.setBounds(250,600,260,35);
q.setFont(new Font("sans seriff", Font.PLAIN, 20));

qfx=new JButton(" Withdraw");
qfx.setBounds(520,545,220,35);
qfx.setFont(new Font("sans seriff", Font.PLAIN, 20));
```

```
//for debit card clear button
bb=new JButton("clear");
bb.setBounds(290,545,170,35);
bb.setFont(new Font("sans seriff", Font.PLAIN, 20));

bb1=new JButton("Display");
bb1.setBounds(30,545,170,35);
bb1.setFont(new Font("sans seriff", Font.PLAIN, 20));

p.setBackground(Color.red);
q.setBackground(Color.red);
bb.setBackground(Color.green);
bb1.setBackground(Color.green);
```

```
p.setForeground(Color.white);
q.setForeground(Color.white);
bb.setForeground(Color.black);

//jlabel credit  gui
a1=new JLabel("CREDITGUI");
a1.setBounds(350,-140,250,300);
a1.setFont(new Font("sans seriff", Font.PLAIN, 30));

b1=new JLabel("Interest rate:");
b1.setBounds(610,65,500,90);
b1.setFont(new Font("sans seriff", Font.PLAIN, 25));
c1=new JTextField();
```

```
c1.setBounds(19,234,270,45);//Balance amount

d1=new JLabel("cvc number:");
d1.setBounds(24,60,200,90);
d1.setFont(new Font("sans seriff", Font.PLAIN, 25));
e1=new JTextField();
e1.setBounds(19,129,270,45);  //cvc

f1=new JLabel("Card Id:");
f1.setBounds(24,424,300,20);
f1.setFont(new Font("sans seriff", Font.PLAIN, 25));
g1=new JTextField();
g1.setBounds(605,244,270,45);//client name
```

```
h1=new JLabel("Client name:");
h1.setBounds(610,172,250,90);
h1.setFont(new Font("sans seriff", Font.PLAIN, 25));
i1=new JTextField();
i1.setBounds(15,453,270,45);//card id

x1=new JLabel("Issuer bank:");
x1.setBounds(610,425,300,38);
x1.setFont(new Font("sans seriff", Font.PLAIN, 25));
k1=new JTextField();
k1.setBounds(605,463,270,45);

app1=new JLabel("Bank account:");
app1.setBounds(19,312,300,20);
```

```
app1.setFont(new Font("sans seriff", Font.PLAIN, 25));
kk1=new JTextField();
kk1.setBounds(15,342,270,45);

setcredit=new JLabel("Setcreditlimit:");
setcredit.setBounds(610,322,200,40);
setcredit.setFont(new Font("sans seriff", Font.PLAIN, 25));
setcredit1=new JTextField();
setcredit1.setBounds(605,359,270,45);

grace=new JLabel("Grace period:");
grace.setBounds(294,573,200,90);
grace.setFont(new Font("sans seriff", Font.PLAIN, 25));
period=new JTextField();
```

```
period.setBounds(289,641,270,45);

m1=new JTextField();
m1.setBounds(605,129,270,45);// upper interest rate

l1=new JLabel("Balance amount:");
l1.setBounds(19,207,201,20);
l1.setFont(new Font("sans seriff", Font.PLAIN, 25));

n1=new JLabel("Expiration Date:");
n1.setBounds(377,415,210,20);
n1.setFont(new Font("sans seriff", Font.PLAIN, 20));
```

```
String days1[]={"1","2","3","4","5","6","7","8","9","10","11","12","13","14","15","16","17","18","19","20","21","22","23","24","25","26","27","28","29","30"};
oo=new JComboBox<String>(days1);
oo.setBounds(323,453,83,42);

String days2[]={"january","februrary","march","april","may","jun","july","august","september","october","november","december"};
oa=new JComboBox<String>(day1);
oa.setBounds(416,453,87,42);

String days3[]={"2018","2019","2020","2021","2022","2023","2024","2025"};
oaa=new JComboBox<String>(day2);
```

```
oaa.setBounds(513,453,71,42);

p1=new JButton("Add credit Card");
p1.setBounds(331,358,190,35);
p1.setFont(new Font("sans seriff", Font.PLAIN, 20));

q1=new JButton(" Go to Debit Card");
q1.setBounds(619,620,220,35);
q1.setFont(new Font("sans seriff", Font.PLAIN, 20));

//for credit card clear button628 542
tsg=new JButton("clear");
tsg.setBounds(45,624,170,35);
tsg.setFont(new Font("sans seriff", Font.PLAIN, 20));
```

```
tsgx=new JButton("Set Creditlimit");
tsgx.setBounds(630,542,210,35);
tsgx.setFont(new Font("sans seriff", Font.PLAIN, 20));

tsge=new JButton("Cancel creditCard");
tsge.setBounds(325,540,230,35);
tsge.setFont(new Font("sans seriff", Font.PLAIN, 20));

tsg1=new JButton("Display");
tsg1.setBounds(42,540,190,35);
tsg1.setFont(new Font("sans seriff", Font.PLAIN, 20));

Color my=new Color(215, 91, 227 );
```

```
tsg1.setBackground(my);

p1.setBackground(Color.red);
q1.setBackground(Color.red);

tsg.setBackground(Color.green);
tsg1.setBackground(Color.green);
tsge.setBackground(Color.green);
tsgx.setBackground(Color.green);
qfx.setBackground(Color.green);
p1.setForeground(Color.white);
q1.setForeground(Color.white);

bb.addActionListener(this);
```

```
tsg.addActionListener(this);
p1.addActionListener(this);
p.addActionListener(this);
qfx.addActionListener(this);
tsgx.addActionListener(this);
tsge.addActionListener(this);
tsg1.addActionListener(this);
bb1.addActionListener(this);
q.addActionListener(this);
q1.addActionListener(this);

//adding j components into jpanel credit gui
panel2.add(a1);
panel2.add(b1);
```

```
panel2.add(c1);
panel2.add(d1);
panel2.add(e1);
panel2.add(f1);
panel2.add(g1);
panel2.add(h1);
panel2.add(i1);
panel2.add(x1);
panel2.add(k1);
panel2.add(l1);
panel2.add(m1);
panel2.add(n1);
panel2.add(oa);
panel2.add(oo);
```

```
panel2.add(oaa);
panel2.add(app1);
panel2.add(kk1);
panel2.add(p1);
panel2.add(q1);
panel2.add(setcredit);
panel2.add(setcredit1);
panel2.add(tsgx);
panel2.add(tsge);
panel2.add(tsg);
panel2.add(tsg1);
frame2.add(panel2);
panel2.add(grace);
panel2.add(period);
```

```
//adding j components into jpanel

//debitgui add
panel1.add(a);
panel1.add(b);
panel1.add(c);
panel1.add(d);
panel1.add(e);
panel1.add(f);
panel1.add(g);
panel1.add(h);
panel1.add(i);
panel1.add(bb);
```

```
panel1.add(k);
panel1.add(bb1);
panel1.add(p);
panel1.add(kk);
panel1.add(app);
panel1.add(q);
panel1.add(qfx);
panel1.add(x);
panel1.add(l);
panel1.add(m);
panel1.add(o2);
panel1.add(o3);
panel1.add(o4);
panel1.add(n);
```

```
frame.add(panel1);
panel1.setLayout(null);
panel2.setLayout(null);
frame.setLayout(null);
frame2.setLayout(null);
frame.setVisible(true);

frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
frame2.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
frame.setResizable(false);
frame2.setResizable(false);
tsg.setFocusable(false);
bb.setFocusable(false);
q.setFocusable(false);
```

```
    q1.setFocusable(false);
    p1.setFocusable(false);
    p.setFocusable(false);
    tsgx.setFocusable(false);
    tsge.setFocusable(false);
    tsg1.setFocusable(false);
    bb1.setFocusable(false);

}
public static void main(String[]args)
{

    new Bankgui();
```

```
}

Public void actionPerformed (ActionEvent z)
{

    if (z.getSource() == tsg)//triggering this action event when cear
button is pressed which set text to empty string for creditcard
    {
        e1.setText ("");
        c1.setText ("");
        i1.setText ("");
        k1.setText ("");
        g1.setText ("");
        m1.setText ("");
```

```
        kk1.setText ("");
        setcredit1.setText ("");
        period.setText ("");


    }

    else if (z.getSource() == bb)//triggered this action event when clear
button from debit card is pressed
    {
        c.setText("");
        e.setText("");
        g.setText("");
        i.setText("");
        k.setText("");
```

```
    m.setText("");
    kk.setText("");

}
else if(z.getSource() == q)
{
    frame2.setVisible(true);
    frame.setVisible(false);
}
else if(z.getSource() == q1)
{
    frame2.setVisible(false);
    frame.setVisible(true);
}
```

```
        else if (z.getSource() == p1) //credit card add button

        {

            if(c1.getText().isEmpty() || e1.getText().isEmpty() ||
i1.getText().isEmpty() || m1.getText().isEmpty() || l1.getText().isEmpty())

            {
                JOptionPane.showMessageDialog(frame2,"please enter some
input in textfield","Creditcard",JOptionPane.WARNING_MESSAGE);

            }

            else
```

```
{
    if (card.isEmpty())

    {
        try

        {

            xbn=Integer.parseInt(c1.getText());
            ynv=Integer.parseInt(e1.getText());
            xbm1=Integer.parseInt(i1.getText());
            y1=Integer.parseInt(m1.getText());
            gracy=Integer.parseInt(period.getText());
            yj1=l1.getText();
```

```
x3=k1.getText();
x2=g1.getText();
hh=kk1.getText();
setcreditl=setcredit1.getText();
String sun=(String)oo.getSelectedItem();
String jan=(String)oa.getSelectedItem();
String first=(String)oaa.getSelectedItem();
String date=sun+jan+first;
obj1=new
Creditcard(xbm1,"x2","x3","hh",xbn,ynv,y1,date);
card.add(obj1);
JOptionPane.showMessageDialog(frame," Details
successfully added to credit
card","creditcard",JOptionPane.INFORMATION_MESSAGE);
e1.setText("");
```

```
            c1.setText("");
            i1.setText("");
            k1.setText("");
            g1.setText("");
            m1.setText("");
            kk1.setText("");
            setcredit1.setText("");
            period.setText("");

        }//try ended
        catch (NumberFormatException e) //catch block for any
exceptions
        {
```

```
                JOptionPane.showMessageDialog(frame2, "Exception
found while adding to creditcard.", "Error",
JOptionPane.ERROR_MESSAGE);
            }


        }

        else //for not empty arraylist

        {
            boolean duplicate =false;
            for (Bankcard hp: card)
            {

                if(hp instanceof Creditcard)
```

```
    {
        Creditcard credit=(Creditcard)hp;
        if(obj1.getcard_id() == xbm1)
        {
            duplicate =true;
            break;

        }

    }
}
if (duplicate)
{
```

```
                JOptionPane.showMessageDialog(frame,"Duplicate card
id found","debitcard",JOptionPane.WARNING_MESSAGE);

            }

            else

            {

                obj1=new
Creditcard(xbm1,"x2","x3","hh",xbn,ynv,y1,date);
                card.add(obj1);
                JOptionPane.showMessageDialog(frame," non-empty
Details successfully added to Credit
card","creditcard",JOptionPane.INFORMATION_MESSAGE);
```

```
            e1.setText("");
            c1.setText("");
            i1.setText("");
            k1.setText("");
            g1.setText("");
            m1.setText("");
            kk1.setText("");
            setcredit1.setText("");
            period.setText("");

        }

    }
}
```

```
        }

        //for debit card add button

        else if (z.getSource() == p)

        {

            if(c.getText().isEmpty() || g.getText().isEmpty() ||
e.getText().isEmpty() || i.getText().isEmpty() || k.getText().isEmpty() ||
            m.getText().isEmpty())
            {

                JOptionPane.showMessageDialog(frame,"Please fill up details
in textfield","debitcard",JOptionPane.WARNING_MESSAGE);
```

```
}

else
{
    if(card.isEmpty())

    {
        try

        {

            x5=Integer.parseInt(c.getText());
            y5=g.getText();
```

```
x6=Integer.parseInt(e.getText());
y6=i.getText();
x7=k.getText();
y7=Integer.parseInt(m.getText());
withdrawalamount=Integer.parseInt(kk.getText());
String mon=(String)o2.getSelectedItem();
String feb=(String)o3.getSelectedItem();
String second=(String)o4.getSelectedItem();
String date1 = mon+feb+second;

obj=new Debitcard(x6,y7,"x7","y5","y6",x5);
card.add(obj);
JOptionPane.showMessageDialog(frame," Details
successfully added to debit card"
+JOptionPane.INFORMATION_MESSAGE);
```

```
            c.setText("");
            e.setText("");
            g.setText("");
            i.setText("");
            k.setText("");
            m.setText("");
            kk.setText("");
        }
        catch (NumberFormatException e)

        {
            JOptionPane.showMessageDialog(frame, "Exception
occurs while adding to Debitcard .", "Error",
JOptionPane.ERROR_MESSAGE);
```

```
    }

}

else

{

    for (Bankcard mac : card)
    {
        if (mac instanceof Debitcard)

        {
```

```
                    Debitcard debit=(Debitcard)mac;

                    if (obj.getcard_id() == y7)
                    {
                        JOptionPane.showMessageDialog(frame,"Duplicate
card id found","debitcard",JOptionPane.WARNING_MESSAGE);

                    }

                    else
                    {

                        obj=new Debitcard(x6,y7,"x7","y5","y6",x5);
                        card.add(obj);
```

```
                    JOptionPane.showMessageDialog(frame," cardid is
non empty,Details successfully added to debit card","debitcard",
JOptionPane.INFORMATION_MESSAGE);
                        c.setText("");
                        e.setText("");
                        g.setText("");
                        i.setText("");
                        k.setText("");
                        m.setText("");
                        kk.setText("");
                    }
                }


            }
        }
```

```
            }


        }
        else if(z.getSource() == qfx)
        {

            if(c.getText().isEmpty() || m.getText().isEmpty() ||
kk.getText().isEmpty())

            {
                JOptionPane.showMessageDialog(frame, "Empty textfield
,please enter some value .", "Error", JOptionPane.ERROR_MESSAGE);
            }
```

```
        else
        {
            if (card.isEmpty())
            {

                JOptionPane.showMessageDialog(frame, "Empty card id,
debit card is not added.", "Error", JOptionPane.ERROR_MESSAGE);

            }
            else
            {
                try
                {
```

```
String mon=(String)o2.getSelectedItem();
String feb=(String)o3.getSelectedItem();
String second=(String)o4.getSelectedItem();
String date1 = mon+feb+second;
Integer pinnumber=Integer.parseInt(c.getText());
Integer cardid=Integer.parseInt(m.getText());
Integer withdrawalamount=Integer.parseInt(kk.getText());

for (Bankcard withdraw : card)

{
    if (withdraw instanceof Debitcard)

    {
```

```
                    if(obj.getcard_id() == cardid)

                    {

                        Debitcard debit=(Debitcard)withdraw;

debit.withdraw(withdrawalamount,date1,pinnumber);
                            JOptionPane.showMessageDialog(frame,
"withdrawal successfull", "sucess",
JOptionPane.INFORMATION_MESSAGE);

                    }
                    else
                    {
```

```
                        JOptionPane.showMessageDialog(frame,
"incorrect card id found", "Error", JOptionPane.ERROR_MESSAGE);
                    }
                }
            }

        }
        catch(NumberFormatException e)

        {
            JOptionPane.showMessageDialog(frame, "Exception
occurs while withdrawing", "Error", JOptionPane.ERROR_MESSAGE);
        }

    }
```

```
            }

        }

        else if(z.getSource() == tsgx)

        {

            if(period.getText().isEmpty() || i1.getText().isEmpty() ||
setcredit1.getText().isEmpty())

            {
                JOptionPane.showMessageDialog(frame2, "please enter some
data.", "Error", JOptionPane.ERROR_MESSAGE);
```

```
        }

        else
        {
            if (card.isEmpty())
            {
                JOptionPane.showMessageDialog(frame2, "Empty card id
credit card is not added.", "Error", JOptionPane.ERROR_MESSAGE);

            }

            else
            {
                try
```

```
{
    Integer graceperiod = Integer.parseInt(period.getText());
    Integer cardd=Integer.parseInt(i1.getText());
    Integer setlimit=Integer.parseInt(setcredit1.getText());

    for (Bankcard setlimited : card)

    {

        if (setlimited instanceof Creditcard)

        {
```

```
                    if(obj1.getcard_id() == cardd)

                    {
                        Creditcard credit=(Creditcard)setlimited;
                        credit.setcreditlimit(setlimit,graceperiod);
                        // Display success message
                        JOptionPane.showMessageDialog(frame2, "Credit
limit set succesfully","sucess",JOptionPane.INFORMATION_MESSAGE
);

                    }
                }
            }
        }
```

```
                catch(NumberFormatException e)

                {
                    JOptionPane.showMessageDialog(frame2, "Exception
occurs while setting credit limit.", "Error",
JOptionPane.ERROR_MESSAGE);

                }

            }
        }

    }

    else if(z.getSource() == tsge)
```

```
        {
            if(i1.getText().isEmpty())


            {
                JOptionPane.showMessageDialog(frame2, "please enter some
values  or data.", "Error", JOptionPane.ERROR_MESSAGE);
            }
            else
            {
                if (card.isEmpty())
                {
                    JOptionPane.showMessageDialog(frame2, "Empty card id
credit card is not added.", "Error", JOptionPane.ERROR_MESSAGE);


                }
```

```
else
{
    try
    {
        Integer cardes=Integer.parseInt(i1.getText());
        for (Bankcard id : card)
        {
            if(id instanceof Creditcard)
            {
                Creditcard cancel = (Creditcard)id;
                cancel.cancelcreditcard();
                JOptionPane.showMessageDialog(frame2, "Credit
card cancelled out
succesfully","sucess",JOptionPane.INFORMATION_MESSAGE );
```

```
                }
            }


        }
        catch(NumberFormatException e)
        {
            JOptionPane.showMessageDialog(frame2, "Exception
occurs while cancelling credit card", "Error",
JOptionPane.ERROR_MESSAGE);


        }
    }
  }


}
```

```
        else if(z.getSource() == tsg1) //for creditcard
        {
            for(Bankcard know : card)
            {
                if(know instanceof Creditcard)
                {
                    Creditcard display1=(Creditcard)know;
                    display1.display();
                    JOptionPane.showMessageDialog(frame, "Details of
creditcard displays Successfully", "success",
JOptionPane.INFORMATION_MESSAGE);

                }
            }
```

```
        }
        else  //for debitcard
        {
            for(Bankcard us : card)
            {
                if(us instanceof Debitcard)
                {
                    Debitcard display2=(Debitcard)us;
                    display2.display();
                    JOptionPane.showMessageDialog(frame, "Details of
debitcard displays Successfully", "sucess",
JOptionPane.INFORMATION_MESSAGE);
                }
            }
```

```
        }

    }

}
```