



**CS4001NI Programming**

**30% Individual Coursework**

**2022-23 Autumn**

**Student Name: Sajin Raj Amatya**

**London Met ID:** E.g. 18053646

**College ID:** E.g. NP01MM0474747

**Group: Enter your group here**

**Assignment Due Date: Friday, January 27, 2023**

**Assignment Submission Date: Friday, January 27, 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.*

# **Introduction**

Java is a programming language that was developed by Sun Microsystems in 1995 and follows the pattern of write-once and run-anywhere (Shankar, 2023). Once a Java program is written in a particular platform we can run it on another platform too as it is platform-independent. Java is an object-oriented, cross-platform, fast, secure, and reliable programming language used for developing applications for web, mobile, desktop ,etc. (amazon, 2023).

In this individual assignment, we were assigned to implement the graphical user interface for a system that stores a Bank Card detail of it in an array list and also integrate the first part of the coursework. A new class BankingGUI was created for the implementation of a graphical user interface in the project. In BankingGUI class both Swing and AWT framework of Java were imported in order to create a user-friendly interactive interface. Labels and a Text field for each attribute of Bankcard, Debit card, and Credit card was created with a suitable button with the functionality of adding, removing, and displaying the data entered by the user in an object of debit card or credit card. Bankcard Array List is also created for storing the object of debit and credit card. Combo Box was also used for withdrawal date and expiration date.

A new object of debit or credit card with user entered data was added to the bank card array list by using for each loop to iterate thought the array list and instance of was used to distinguish between the credit and debit card object. If object with same card ID was detected, then error message would be displayed in the dialog box. Display button would display the user entered data if the object of credit card or debit card was successfully added to the bankcard array list. For withdraw, set credit limit and cancelling credit card their respective method from Debit card and Credit card class was called to perform the operation in Banking GUI class.

The tool used for completing this course work are:

1. Bluej : For coding and developing project in java BlueJ an integrated development environment for java was used (Rouse, 2017).
2. MS-Word: MS-Word was used for report writing.
3. Draw.io: Draw.io was used for creating combine relationship class diagram between four classes.

# **Class Diagram**

The class diagram is graphical representation of the classes which represent the overall structure of the system by showing all the methods, attributes and relation between different classes in the system (Visual-paradigm, 2022).

## **2.1) Class Diagram of Bankcard**

Table 1 : Class diagram of Bankcard

|  |
| --- |
| **Bankcard** |
| * balance\_Amount: int * card\_Id: int * issuer\_Bank: String * client\_Name: String * bank\_Account : String |
| +<<constructor>>Bankcard (balance\_Amount:int, card\_Id: int, Issuer\_Bank : String, bank\_Account : String)  + getBalance\_Amount() : int  + getCard\_Id() : int  + getIssuer\_Bank () : String  + getClient\_Name (): String  + getBank\_Account () : String  + setBalance\_Amount (balance\_Amount:int): void  + setClient\_Name(client\_Name: String): void  + display(): void |

## **2.2) Class diagram of Debitcard**

|  |
| --- |
| **Debitcard** |
| * pin\_Number: int * date\_Of\_Withdrawal: String * withdrawal\_Amount: int * has\_Withdrawn : boolean |
| +<<constructor>>Bankcard (balance\_Amount : int , card\_Id : int, bank\_Account : String Issuer\_Bank :  String, client\_Name:String , pin\_Number:int)  +getPin\_Number () : int  +getWithdrawal\_Amount (): int  + getDate\_Of\_Withdrawal () : String  +getHas\_Withdrawn (): boolean  +setWithdrawal\_Amount (withdrawal\_Amount: int) : void  +withdraw (withdrawal\_Amount: int, pin\_Number:int): void  + display (): void |

Table 2: Class diagram of Debitcard

## **2.3) Class diagram of Creditcard**

Table 3: Class diagram of Creditcard

|  |
| --- |
| **Creditcard** |
| * cvc\_number: int * credit\_Limit: double * interest\_Rate: double * expiration\_Date: String * grace\_Period: int * is\_Granted : boolean |
| +<<constructor>>Creditcard ( card\_Id : int, client\_Name : String, issuer\_Bank : String, bank\_Account : String, balance\_Amount: int, cvc\_Number : int, interest\_Rate : double, expiration\_Date : String) + getCvc\_Number (): int  + getCredit\_Limit (): double  + getInterest\_Rate (): double  + getExpiration\_Date (): String  + getGrace\_Period (): int  + getIs\_Granted (): Boolean  + setCredit\_Limit (new\_Creditlimit : double, new\_Gracelimit : int ) : void  + cancel\_Credit\_Card (): void  + display() : void |

## **2.4) Class diagram of BankingGUI**

|  |
| --- |
| **BankingGUI** |
| - frameMain : JFrame - frameDebitCard : JFrame - frameWithdraw: JFrame - frameCreditCard : JFrame - frameCancelCreditCard : JFrame - frameCreditlimit : JFrame - panelDebitCard : JPanel - panelCreditCard : JPanel - panelWithdraw : JPanel - panelCancelCreditCard: JPanel - panelCreditlimit : JPanel - panelcolor : Color - textfieldcolor : Color - lightBlue : Color - Montecarlo: Color - darkgreen : Color - lightgreen : Color - lightRed : Color  - DarkBlue : Color - buttonDebitCard : JButton - buttonClearDebitCard : JButton - buttonDisplayDebitCard : JButton - buttonWithdraw : JButton - buttonBackDebitCard : JButton - buttonBackWIthdraw : JButton - buttonAdDebitCard : JButton - buttonWithdrawalAmount : JButton - buttonClearWithdraw : JButton - buttonAddCreditCard  : JButton - buttonCreditCard : JButton - buttonClearCreditCard : JButton - buttonDisplayCreditCard : JButton - buttonBackCreditlimit : JButton - buttonBackCreditCard : JButton - buttonBackCancelCreditCard : JButton - buttonCancelCredit : JButton - buttonSetCreditLimit : JButton - buttonCreditLimit : JButton - textfieldIssuerBank : JTextField - textfieldClientName : JTextField - textfieldBankAccount : JTextField - textfieldBalanceAmount : JTextField - textfieldCarId : JTextField - textfieldWithdrawCardId : JTextField - textfieldWithdrawalAmount : JTextField - textfieldIssuerBankCreditCard : JTextField - textfieldClientNameCreditCard : JTextField - textfieldBankAccountCreditCard : JTextField - textfieldSetCreditlimit : JTextField - textfieldGracePeriod : JTextField - textfieldCardICreditCard : JTextField - textfieldBalanceAmountCreditCard : JTextField - textfieldCVCNumber : JTextField - textfieldInterestRate : JTextField - textfieldCancelCreditCardId : JTextField - textfieldCreditlimitCardId : JTextField - textfieldCreditlimit : JTextField - textfieldPIN : JTextField - textfieldWithdrawPIN : JTextField - BankcardArraylist : ArrayList <Bankcard> - withdrawYear : JComboBox - withdrawMonth  : JComboBox - withdrawDay  : JComboBox - expirationYear  : JComboBox - expirationMonth  : JComboBox - expirationDay  : JComboBox - balanceAmount : int  - cardID : int  - PINNumber : int  - balanceAmountCreditCard : int  - cardIDCreditCard : int - CVCNumber : int  - cardIdWithdraw : int  - PINNumberWithdraw : int  - withdrawAmount : int  - PINNumberCreditCard : int  - cardIdCancelCreditCard : int  - cardIdSetCreditlimit : int  - gracePeriod : int  - InterestRate : double  - creditlimit : double |
| + <<constructor>> BankingGUI() + actionPerformed(ActionEvent e) : void + debitCardGUI ()  : void  + clearCreditCard ()  : void  + clearDebitCard () : void  + userInputDebitCard () : void  + displayDebitCard () : void  + withdrawGUI () : void  + userInputWithdrawAmount() : void  + creditCardGUI () : void  + userInputCreditCard () : void  + displayCreditCard() : void  + cancelcreditcardGUI () : void  +  setCreditLimitGUI () : void  + userInputSetCreditLimit() : void |

Table 4: Class Diagram of BankingGUI

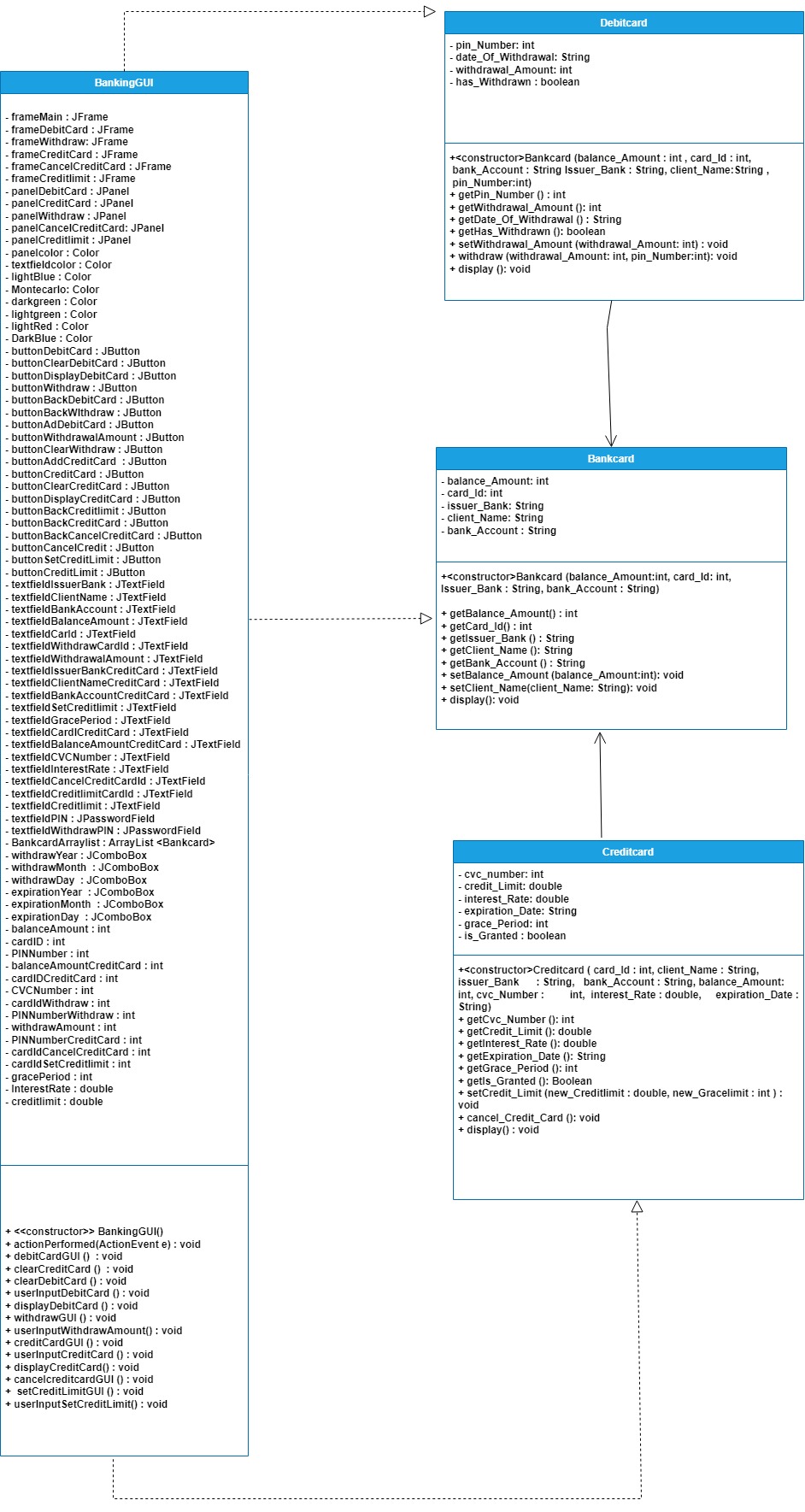
****

Figure 1 Relation Diagram

**DECLARE** frameMain, frameDebitCard,frameWithdraw,frameCreditCard,frameCancelCreditCard,

frameCreditlimit as a JFrame

**DECLARE** panelDebitCard, panelWithdraw, panelCreditCard,panelCancelCreditCard, panelCreditlimit as JPanel

**DECLARE** panelcolor, textfieldcolor, lightBlue, Montecarlo, darkgreen, lightgreen, lightRed, DarkBlue as Color

**DECLARE** buttonDebitCard,buttonClearDebitCard,buttonDisplayDebitCard,buttonWithdraw

,buttonBackDebitCard,buttonBackWithdraw,buttonAddDebitCard,buttonWithdrawalAmount, buttonClearWithdraw,buttonAddCreditCard,buttonCreditCard,buttonClearCreditCard, buttonDisplayCreditCard,buttonBackCreditlimit,buttonBackCreditCard,buttonBackCancelCreditCard, buttonCancelCreditCard,buttonCancelCredit,buttonSetCreditLimit,buttonCreditLimit as JButton

**DECLARE** textfieldIssuerBank, textfieldClientName,textfieldBankAccount,textfieldBalanceAmount,

textfieldCardId, textfieldWithdrawCardId, textfieldWithdrawalAmount, textfieldIssuerBankCreditCard, textfieldClientNameCreditCard,textfieldBankAccountCreditCard,textfieldSetCreditlimit,textfieldGracePeriod,textfieldCardIdCreditCard,textfieldBalanceAmountCreditCard,textfieldPIN,textfieldWithdrawPIN textfieldCVCNumber,textfieldInterestRate,textfieldCancelCreditCardId,textfieldCreditlimitCardId,textfieldCreditlimit as JTextfield

**DECLARE** instance variablebalanceAmount ,cardID, PINNumber, balanceAmountCreditCard, cardIDCreditCard, CVCNumber, cardIdWithdraw ,PINNumberWithdraw, withdrawAmount, PINNumberCreditCard, cardIdCancelCreditCard, cardIdSetCreditlimit, gracePeriod as int

**DECLARE** instance variable InterestRate, creditlimit as double

# **Pseudocode of BankingGUI**

**CREATE** a constructor of BankingGUI

**DO**

**DECLARE BankcardArraylist**

**DECLARE** frameMain as JFrame object

**SET** background color of frameMain to gray color

**DECLARE** labelheading1 as JLabel ("sajin bank") object

**SET** the size and position of labelheading1

**SET** the font style, font size and font family of a label heading1 to Century Gothic, bold, size 60

**SET** font color of labelheading1 to Black

**ADD** labelheading1 to frameMain

**DECLARE** label heading as JLabel ("Our Services") object

**SET** the size and position of a labelheading

**SET** the font style, font size and font family of a labelheading to Century Gothic, bold, size 30

**SET** font color of labelheading as black

**ADD** label heading to frameMain

**DECLARE** buttonDebitCard as a JButton(“Debit Card”) object

**SET** size and position of buttonDebitCard

**SET** font style, font size, and font family of buttonDebitCard to Century Gothic, bold, size 19

**SET** font color of buttonDebitCard to black

**SET** background color of buttonDebitCard

**ADD** actionlistener to buttonDebitCard

**ADD** buttonDebitCard to frameMain

**DECLARE** buttonCreditCard as a JButton object

**SET** size and position of buttonCreditCard

**SET** font style, font size, and font family of buttonCreditCard to Century Gothic, bold, size 19

**SET** font color of buttonCreditCard to black

**SET** background color of buttonCreditCard

**ADD** actionlistener to buttonCreditCard

**ADD** buttonCreditCard to frameMain

**SET** position and size of frameMain

**SET** layout of frameMain to null

**SET** visibility of a frameMain to true

**SET** default close operation to frameMain

**END DO**

**CREATE** method actionPerformed with no return type

**DO**

**IF** source of the ActionEvent is equal to buttonDebitCard **OR** buttonBackWithdraw

**CALL** debutCardGUI() method

**SET** Visibility of frameMain to false

**SET** Visibility of frameDebitCard to true

**SET** Visibility of frameWithdraw to false

**END IF**

**ELSE IF** source of ActionEvent is equal to buttonBackDebitCard OR buttonBackCreditlimit

**SET** visibility of frameMain to true

**SET** visibility of frameDebitCard to false

**SET** visibility of frameCreditCard to false

**END ELSE IF**

**ELSE IF** source of ActionEvent equal to buttonBackCancelCreditCard

**OR** buttonBackCancelCreditCard

**SET** visibility of frameCreditCard to true

**SET** visibility of frameCancelCreditCard to fase

**SET** visibility of frameCreditlimit to false

**END ELSE IF**

**ELSE IF** source of ActionEvent is equal to buttonWithdraw

**CALL** withdrawGUI() method

**SET** visibility of frame main to false

**SET** visibility of frameDebitCard to false

**SET** visibility of frameWithdraw to true

**END ELSE IF**

**ELSE IF** source of ActionEvent is equal to buttonClearDebitCard

**CALL** ClearDebitCard() method

**END ELSE IF**

**ELSE IF** source of ActionEvent is equal to buttonClearCreditCard

**CALL** ClearCreditCard() method

**END ELSE IF**

**ELSE IF** source of ActionEvent is equal to buttonSetCreditLimit

**CALL** setCreditLimitGUI() method

**SET** visibility of frameCreditlimit to true

**SET** visibility of frameMain to false

**SET** visibility of frameCreditCard to false

**END ELSE IF**

**ELSE IF** source of ActionEvent is equal to buttonClearWithdraw

**SET** value of textfieldWithdrawCardId to empty

**SET** value of textfieldWithdrawalAmount to empty

**SET** value of textfieldWithdrawPIN to empty

**END ELSE IF**

**ELSE IF** source of ActionEvent is equal to buttonCreditCard

**CALL** creditcardGUI () method

**SET** visibility of frameMain to false

**SET** visibility of frameCreditCard to true

**END ELSE IF**

**ELSE IF** source of ActionEvent is equal to buttonAddDebitCard

**CALL** UserInputDebitCard() method

**END ELSE IF**

**ELSE IF** source of ActionEvent is equal to buttonCancelCredit

**CALL** cancelcreditcardGUI() method

**SET** visibility of frameMain to false

**SET** visibility of frameCreditCard to false

**SET** visibility of frameCancelCreditCard to true

**END ELSE IF**

**ELSE IF** source of ActionEvent is equal to buttonDisplayDebitCard

**CALL** displayDebitCard() method

**END ELSE IF**

**ELSE IF** source of ActionEvent is equal to buttonAddCreditCard

**CALL** userInputCreditCard () method

**END ELSE IF**

**ELSE IF** source of ActionEvent is equal to buttonDisplayCreditCard

**CALL** displayCreditCard()method

**END ELSE IF**

**ELSE IF** source of ActionEvent is equal to buttonWithdrawalAmount

**CALL** userInputWithdrawAmount() method

**END ELSE IF**

**ELSE IF** source of ActionEvent is equal to buttonCancelCreditCard

**CALL** userInputSetCreditLimit() method

**END ELSE IF**

**END DO**

**CREATE** debitCardGUI() method with no return type

**DO**

**DECLARE** frameDebitCard as object of JFrame

**SET** background color of rameDebitCard to gray

**DECLARE** panelDebitCard as object of JPanel

**SET** size and position of panelDebitCard

**SET** panelcolor as object of Color

**SET** layout of panelDebitCard to null

**DECLARE** labelheading as a new object of JLabel ("Sajin Bank")

**SET** size and position of labelheading

**SET** font style of label headingCentury Gothic, Bold, size 60

**SET** color of labelheading to Black

**ADD** labelheading to frameDebitCard

**DECLARE** labelheading1 as a new object of JLabel(“Debit Card”)

**SET** size and position of labelheading1

**SET** font of label heading1 Century Gothic, Bold, size 30

**SET** color of labelheading1 to Black

**ADD** labelheading1 to frameDebitCard

**DECLARE** labelClientName as a new object of JLabel(“Client Name”)

**SET** size and position of a labelClientName

**SET** font of labelClientName Century Gothic, Bold, size 17

**SET** font color of labelClientNameto Black

**ADD** labelClientName to panelDebitCard

**DECLARE** labelBankAccount as a new object of JLabel(“Bank Account”)

**SET** size and position of a labelBankAccount

**SET** font of labelBankAccount to Century Gothic, Bold, size 17

**SET** font color of labelBankAccount to Black

**ADD** labelBankAccount to panelDebitCard

**DECLARE** labelBalanceAmount as a new object of JLabel(“Balance Amount”)

**SET** size and position of a labelBalanceAmount

**SET** font labelBalanceAmount to Century Gothic, Bold, size 17

**SET** font color of labelBalanceAmount to Black

**ADD** labelBalanceAmount to panelDebitCard

**DECLARE** labelIssuerBankas a new object of JLabel(“Issuer Bank”)

**SET** size and position of a labelIssuerBank

**SET** font of labelIssuerBank to Century Gothic, Bold, size 17

**SET** font color of labelIssuerBank to Black

**ADD** labelIssuerBank to panelDebitCard

**DECLARE** labelCardId as a new object of JLabel(“Card ID ”)

**SET** size and position of a labelCardId

**SET** font of labelCardId to Century Gothic, Bold, size 17

**SET** font color of labelCardId to Black

**ADD** labelCardId to panelDebitCard

**DECLARE** labelPIN as a new object of JLabel(“PIN Number ”)

**SET** size and position of a labelPIN

**SET** font of labelPIN to Century Gothic, Bold, size 17

**SET** font color of labelPIN to Black

**ADD** labelPIN to panelDebitCard

**DECLARE** textfieldcolor as new object of Color

**DECLARE** textfieldClientName as new object of JTextField()

**SET** size and position of a textfieldClientName

**SET** background color of textfieldClientName

**SET** font of textfieldClientName to Century Gothic, Bold, size 14

**ADD** textfieldClientName to panelDebitCard

**DECLARE** textfieldIssuerBank as new object of JTextField()

**SET** size and position of a textfieldIssuerBank

**SET** background color of textfieldIssuerBank

**SET** font of textfieldIssuerBank to Century Gothic, Bold, size 14

**ADD** textfieldIssuerBank to panelDebitCard

**DECLARE** textfieldBankAccount as new object of JTextField()

**SET** size and position of a textfieldBankAccount

**SET** background color of textfieldBankAccount

**SET** font of textfieldBankAccount to Century Gothic, Bold, size 14

**ADD** textfieldBankAccount to panelDebitCard

**DECLARE** textfieldBalanceAmount as new object of JTextField()

**SET** size and position of a textfieldBalanceAmount

**SET** background color of textfieldBalanceAmount

**SET** font of textfieldBalanceAmount to Century Gothic, Bold, size 14

**ADD** textfieldBalanceAmountto panelDebitCard

**DECLARE** textfieldCardId as new object of JTextField()

**SET** size and position of a textfieldCardId

**SET** background color of textfieldCardId

**SET** font of textfieldCardId to Century Gothic, Bold, size 14

**ADD** textfieldCardId to panelDebitCard

**DECLARE** textfieldPIN as new object of JPasswordField()

**SET** size and position of a textfieldPIN

**SET** background color of textfieldPIN

**SET** font of textfieldPIN to Century Gothic, Bold, size 14

**ADD** textfieldPIN to panelDebitCard

**DECLARE** lightBlue as a new object of Color

**DECLARE** buttonAddDebitCard as a new object of JButton(“Add Debit Card”)

**SET** size and position of buttonAddDebitCard

**SET** font of buttonAddDebitCard to Century Gothic, Bold, size 17

**SET** font color of buttonAddDebitCard to black

**SET** background color of buttonAddDebitCard to lightblue

**ADD** ActionListener to buttonAddDebitCard

**SET** Focus ability of buttonAddDebitCard to false

**ADD** buttonAddDebitCard to panelDebitCard

**DECLARE** darkgreen as a new object of Color

**DECLARE** buttonClearDebitCard as a new object of JButton(“Clear”)

**SET** size and position of buttonClearDebitCard

**SET** font of buttonClearDebitCard to Century Gothic, Bold, size 15

**SET** font color of buttonClearDebitCard to black

**SET** background color of buttonClearDebitCard to darkgreen

**ADD** ActionListener to buttonClearDebitCard

**SET** Focus ability of buttonClearDebitCard to false

**ADD** buttonClearDebitCard to panelDebitCard

**DECLARE** lightgreen as a new object of Color

**DECLARE** buttonDisplayDebitCard as a new object of JButton(“Display”)

**SET** size and position of buttonDisplayDebitCard

**SET** font of buttonDisplayDebitCard to Century Gothic, Bold, size 15

**SET** font color of buttonDisplayDebitCard to black

**SET** background color of buttonDisplayDebitCardto lightgreen

**ADD** ActionListener to buttonDisplayDebitCard

**SET** Focus ability buttonDisplayDebitCard to false

**ADD** buttonDisplayDebitCard to panelDebitCard

**DECLARE** Montecarlo as a new object of Color

**DECLARE** buttonWithdraw as a new object of JButton(“Withdraw Amount”)

**SET** size and position of buttonWithdraw

**SET** font of buttonWithdraw to Century Gothic, Bold, size 17

**SET** font color of buttonWithdraw to black

**SET** background color of buttonWithdraw to Montecarlo

**ADD** ActionListener to buttonWithdraw

**SET** Focus ability buttonWithdraw to false

**ADD** buttonWithdraw to panelDebitCard

**DECLARE** lightRed as a new object of Color

**DECLARE** buttonBackDebitCard as a new object of JButton(“Back”)

**SET** size and position of buttonBackDebitCard

**SET** font of buttonBackDebitCard to Century Gothic, Bold, size 15

**SET** font color of buttonBackDebitCard to black

**SET** background color of buttonBackDebitCard to lightRed

**ADD** ActionListener to buttonBackDebitCard

**SET** Focus ability buttonBackDebitCard to false

**ADD** buttonBackDebitCard to panelDebitCard

**ADD** panelDebitCard to frameDebitCard

**SET** size and position of frameDebitCard

**SET** layout of frameDebitCard to null

**SET** visibility of frameDebitCard to false

**SET** default close operation to frameDebitCard

**CREATE** ClearCreditCard() method with not return type

**DO**

**SET** text of textfieldClientNameCreditCard to empty

**SET** text of textfieldIssuerBankCreditCard to empty

**SET** text of textfieldBankAccountCreditCard to empty

**SET** text of textfieldCVCNumber to empty

**SET** text of textfieldCardIdCreditCard to empty

**SET** text of textfieldInterestRate to empty

**SET** text of textfieldBalanceAmountCreditCard to empty

**END DO**

**CREATE** ClearDebitCard() method with not return type

**DO**

**SET** text of textfieldClientName to empty

**SET** text of textfieldIssuerBank to empty

**SET** text of textfieldBankAccount to empty

**SET** text of textfieldBalanceAmount to empty

**SET** text of textfieldCardId to empty

**SET** text of textfieldPIN to empty

**END DO**

**CREATE** UserInputDebitCard() method with no return type

**DO**

**DECLARE** issuerBank as String type and **GET** text from textfieldIssuerBank

**DECLARE** clientName as String type and **GET** text from textfieldClientName

**DECLARE** bankAccount as String type and **GET** text from textfieldBankAccount

**IF** textfieldBalanceAmount .getText() is empty **OR** textfieldPIN .getText() is empty **OR** textfieldCardId .getText() is empty **OR** issuerBank is empty **OR** clientName is empty **OR** bankAccount is empty

**DISPLAY** please enter all the information on dialogbox

**END IF**

**ELSE**

**TRY**

**GET** text from textfieldBalanceAmount and convert it to integer and **ASSIGN** it to balanceAmount variable of type int

**GET** text from textfieldPIN and convert it to integer and **ASSIGN** it to PINNumber variable of type int

**GET** text from textfieldCardId and convert it to integer and **ASSIGN** it to cardID variable of type int

**CATCH** NumberFormatException

**DISPLAY** Incorrect Format in dialogbox

**END ELSE**

**FOR EACH** object Bankcard card in Bankcard Arraylist

**DO**

**IF** card is instance of Debitcard object **THEN**

**IF** cardID is equal to card\_Id of Debitcard object

**DISPLAY** ERROR, Card ID already exist in dialog box

**END IF**

**END IF**

**END DO**

**END FOR EACH**

**ADD** new Debitcard object with a parameter balanceAmount, cardID, bankAccount, IssuerBank, clientName, PINNumber to BankcardArraylist

**DISPLAY** Debitcard added successfully in dialog box

**END DO**

**CREATE** displayDebitCard() method with no return type

**DO**

**FOR EACH object Bankcard card in BankcardArraylist**

**DO**

**CREATE** Debitcard object variable debit by casting Bankcard object variable card

**DISPLAY** DebitCardDetail in dialogbox

**CALL** display() method of debit object

**END DO**

**CREATE** withdrawGUI() method with no return type

**DO**

**DECLARE** frameWithdraw as object of JFrame(“Withdraw Amount”)

**SET** background color of frameWithdraw to gray

**DECLARE** panelWithdraw as object of JPanel

**SET** size and position of panelWithdraw

**SET** background color of panelWithdraw to panelcolor

**SET** layout of panelWithdraw to null

**DECLARE** labelheading as a new object of JLabel ("Withdraw Amount ")

**SET** size and position of labelheading

**SET** font style of label heading to Century Gothic, Bold, size 30

**SET** color of labelheading to Black

**ADD** labelheading to frameWithdraw

**DECLARE** labelWithdrawCardId as a new object of JLabel(“Card ID ”)

**SET** size and position of a labelWithdrawCardId

**SET** font of labelWithdrawCardId Century Gothic, Bold, size 17

**SET** font color of labelWithdrawCardIdto Black

**ADD** labelWithdrawCardId to panelWithdraw

**DECLARE** labelWithdrawPIN as a new object of JLabel(“PIN Number ”)

**SET** size and position of a labelWithdrawPIN

**SET** font of labelWithdrawPIN to Century Gothic, Bold, size 17

**SET** font color of labelWithdrawPIN to Black

**ADD** labelWithdrawPIN to panelWithdraw

**DECLARE** labelWithdrawalAmount as a new object of JLabel(“Withdrawal Amount ”)

**SET** size and position of a labelWithdrawalAmount

**SET** font labelWithdrawalAmount to Century Gothic, Bold, size 17

**SET** font color of labelWithdrawalAmount to Black

**ADD** labelWithdrawalAmount to panelWithdraw

**DECLARE** labelDateOfWithdrawal as a new object of JLabel(“Issuer Bank”)

**SET** size and position of a labelDateOfWithdrawal

**SET** font of labelDateOfWithdrawal to Century Gothic, Bold, size 17

**SET** font color of labelDateOfWithdrawal to Black

**ADD** labelDateOfWithdrawal to panelWithdraw

**DECLARE** textfieldcolor as new object of Color

**DECLARE** textfieldClientName as new object of JTextField()

**SET** size and position of a textfieldClientName

**SET** background color of textfieldClientName

**SET** font of textfieldClientName to Century Gothic, Bold, size 14

**ADD** textfieldClientName to panelDebitCard

**DECLARE** textfieldIssuerBank as new object of JTextField()

**SET** size and position of a textfieldIssuerBank

**SET** background color of textfieldIssuerBank

**SET** font of textfieldIssuerBank to Century Gothic, Bold, size 14

**ADD** textfieldIssuerBank to panelDebitCard

**DECLARE** textfieldBankAccount as new object of JTextField()

**SET** size and position of a textfieldBankAccount

**SET** background color of textfieldBankAccount

**SET** font of textfieldBankAccount to Century Gothic, Bold, size 14

**ADD** textfieldBankAccount to panelDebitCard

**DECLARE** textfieldBalanceAmount as new object of JTextField()

**SET** size and position of a textfieldBalanceAmount

**SET** background color of textfieldBalanceAmount

**SET** font of textfieldBalanceAmount to Century Gothic, Bold, size 14

**ADD** textfieldBalanceAmountto panelDebitCard

**DECLARE** textfieldCardId as new object of JTextField()

**SET** size and position of a textfieldCardId

**SET** background color of textfieldCardId

**SET** font of textfieldCardId to Century Gothic, Bold, size 14

**ADD** textfieldCardId to panelDebitCard

**DECLARE** textfieldPIN as new object of JPasswordField()

**SET** size and position of a textfieldPIN

**SET** background color of textfieldPIN

**SET** font of textfieldPIN to Century Gothic, Bold, size 14

**ADD** textfieldPIN to panelDebitCard

**DECLARE** lightBlue as a new object of Color

**DECLARE** buttonAddDebitCard as a new object of JButton(“Add Debit Card”)

**SET** size and position of buttonAddDebitCard

**SET** font of buttonAddDebitCard to Century Gothic, Bold, size 17

**SET** font color of buttonAddDebitCard to black

**SET** background color of buttonAddDebitCard to lightblue

**ADD** ActionListener to buttonAddDebitCard

**SET** Focus ability of buttonAddDebitCard to false

**ADD** buttonAddDebitCard to panelDebitCard

**DECLARE** darkgreen as a new object of Color

**DECLARE** buttonClearDebitCard as a new object of JButton(“Clear”)

**SET** size and position of buttonClearDebitCard

**SET** font of buttonClearDebitCard to Century Gothic, Bold, size 15

**SET** font color of buttonClearDebitCard to black

**SET** background color of buttonClearDebitCard to darkgreen

**ADD** ActionListener to buttonClearDebitCard

**SET** Focus ability of buttonClearDebitCard to false

**ADD** buttonClearDebitCard to panelDebitCard

**DECLARE** lightgreen as a new object of Color

**DECLARE** buttonDisplayDebitCard as a new object of JButton(“Display”)

**SET** size and position of buttonDisplayDebitCard

**SET** font of buttonDisplayDebitCard to Century Gothic, Bold, size 15

**SET** font color of buttonDisplayDebitCard to black

**SET** background color of buttonDisplayDebitCardto lightgreen

**ADD** ActionListener to buttonDisplayDebitCard

**SET** Focus ability buttonDisplayDebitCard to false

**ADD** buttonDisplayDebitCard to panelDebitCard

**DECLARE** Montecarlo as a new object of Color

**DECLARE** buttonWithdraw as a new object of JButton(“Withdraw Amount”)

**SET** size and position of buttonWithdraw

**SET** font of buttonWithdraw to Century Gothic, Bold, size 17

**SET** font color of buttonWithdraw to black

**SET** background color of buttonWithdraw to Montecarlo

**ADD** ActionListener to buttonWithdraw

**SET** Focus ability buttonWithdraw to false

**ADD** buttonWithdraw to panelDebitCard

**DECLARE** lightRed as a new object of Color

**DECLARE** buttonBackDebitCard as a new object of JButton(“Back”)

**SET** size and position of buttonBackDebitCard

**SET** font of buttonBackDebitCard to Century Gothic, Bold, size 15

**SET** font color of buttonBackDebitCard to black

**SET** background color of buttonBackDebitCard to lightRed

**ADD** ActionListener to buttonBackDebitCard

**SET** Focus ability buttonBackDebitCard to false

**ADD** buttonBackDebitCard to panelDebitCard

**ADD** panelDebitCard to frameDebitCard

**SET** size and position of frameDebitCard

**SET** layout of frameDebitCard to null

**SET** visibility of frameDebitCard to false

**SET** default close operation to frameDebitCard

# **Method Description of all buttons**

userInputDebitCard()

When Add Debit Card button is pressed userInputDebitCard() method is called which gets value of text field in Debit Card GUI that is entered by the user then, it checks if all the text field are empty and valid format data is entered by the then , Iteration of array list is carried where the instance of object of debit card is checked then, if card ID entered by the user already exist, If the card Id entered by the user doesn’t exist, then new Debit card object with user entered data is added to array list and success message is displayed in dialog box.

displayDebitCard()

When Display button of Debit card GUI is pressed this method is called which displays all the information that is entered by the user.

displayCreditCard()

When display button of Credit card GUI is clicked this method is called which displays all the information entered by the user.

userInputWithdrawAmount()

When withdraw amount button is clicked this method is called which gets the value of text field and combo box from data entered by user in Withdrawal amount GUI, it checks if all text field are empty or not and display error message. After that iteration of array list of Bankcard object is carried out where the instance of object of debit card is checked and down casting operation in conducted. If the card id and pin number entered by the user matches the existing data, then withdraw method of debit card class is called and withdrawal amount is deducted from balance amount by showing success and withdrawal detail message in dialog box. In case the withdrawal amount is greater than balance amount the process is terminated by showing error message in dialog box.

userInputCreditCard()

When Add Credit Card button is pressed this method is called which gets value of text field in Credit Card GUI that is entered by the user then, it checks if all the text field are empty and valid format data is entered by the user then iteration of array list is carried where the instance of object of credit card is checked then, if card ID entered by the user in text field already exist, it displays error message. If the card Id entered by the user doesn’t exist, then new Credit card object with user entered data is added to array list and success message is displayed in dialog box.

clearDebitCard()

When clear button of Debit card GUI is pressed all the text field values are set to empty.

clearCreditCard()

when clear button of Credit card GUI is clicked all the text field values are set to empty.

userInputCancelCreditCard()

This method is called with the cancel credit card button is pressed then if the card ID entered by the user match the existing card ID then it cancels the credit card by showing appropriate dialog box.

userInputSetCreditLimit

This method is called with set credit limit button is click, if card ID entered by the user matches to existing card ID then setCreditLimit() method of credit card class is called with two parameter credit limit and grace period and detail of the credit limit is shown in dialog box.

# **Testing**

Testing is the process of evaluating the quality of the program by inspecting error and bugs so that it fulfil the expected result of the program (Yasar, 2022).

## **5.1) Test 1: Compiling & Running program using command prompt**

|  |  |
| --- | --- |
| Test No. | 1 |
| Objective: | To Compile and run the program using command prompt |
| Action: | * Command prompt was opened with correct file path. * Then all the java file was compiled by using the following code   javac Bankcard.java  javac Debitcard.java  javac Creditcard.java  javac BankingGUI.java   * After that BankingGUI was run using the following code   java BankingGUI |
| Expected Result: | The BankingGUI program will compile and run using command prompt |
| Actual Result: | The BankingGUI program was visible and was compiled and run using command prompt |
| Conclusion: | The test is successful. |

Table 5 Test 1 : Compiling &Running program using command prompt

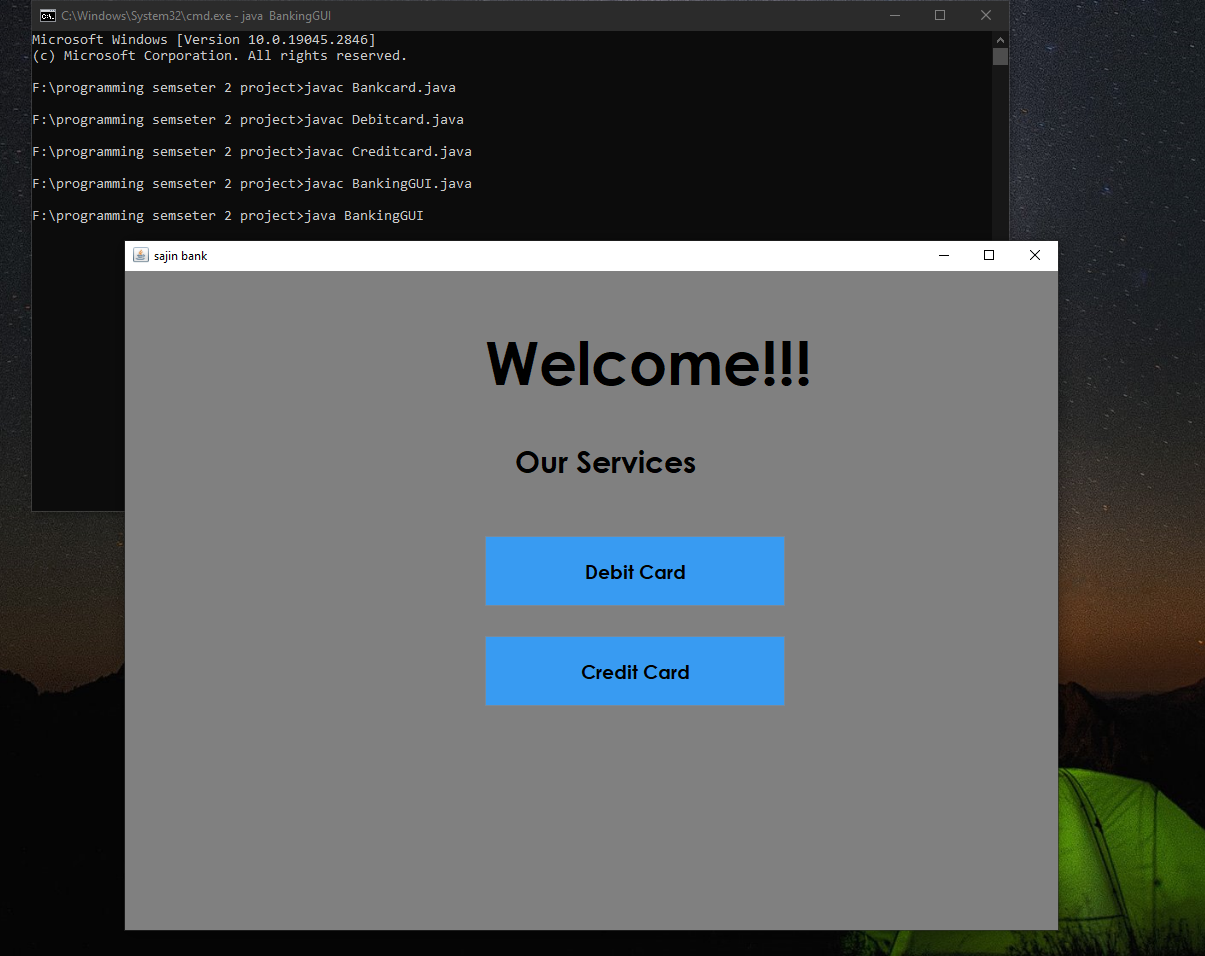


Figure 2 Test 1

## **Test 5.3): Testing Appropriate Dialog boxes when unsuitable values entered**

|  |  |
| --- | --- |
| Test No. | 3 |
| Objective: | To Test Appropriate Dialog boxes when unsuitable values is entered |
| Action: | * The text field of client name, bank account, balance amount, Issuer bank, card ID and PIN Number of debit card GUI was left empty.      * After that, Add Debit card button was clicked. * String data type was entered in text field of card ID, balance amount and PIN Number instead of integer. * Then Add Debit card button was clicked * In text field of card ID, the same ID stored in array list was added again * Then Add Debit card button was clicked |
| Expected Result: | Error message will be displayed in dialog box |
| Actual Result: | Error message was displayed in dialog box |
| Conclusion: | The test is successful. |

Table 6: Test 3 Testing Appropriate Dialog box when unsuitable value is entered

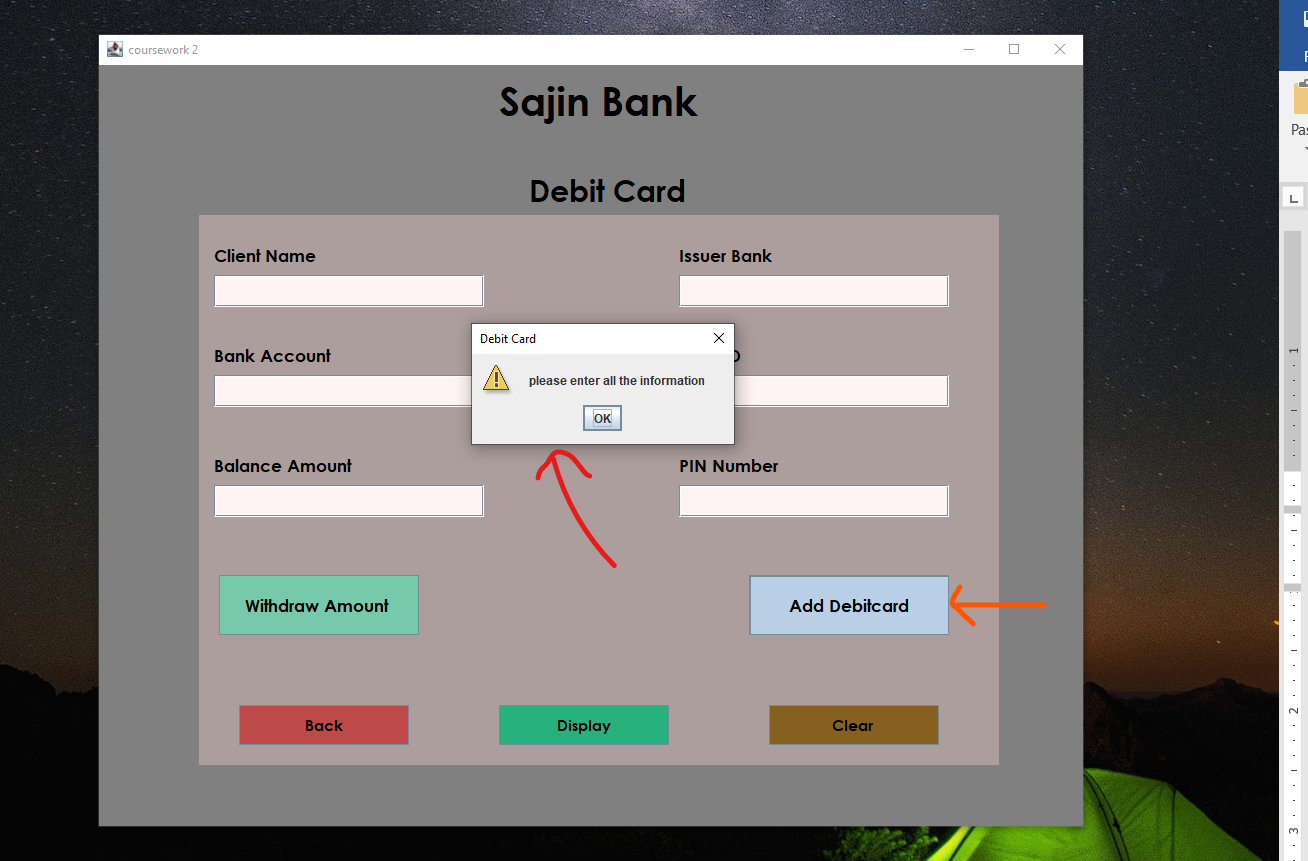
****

Figure 3: Test 3 Adding debit card with empty values in text field



Figure 4: Test 3 Adding debit card with incorrect format data in text field

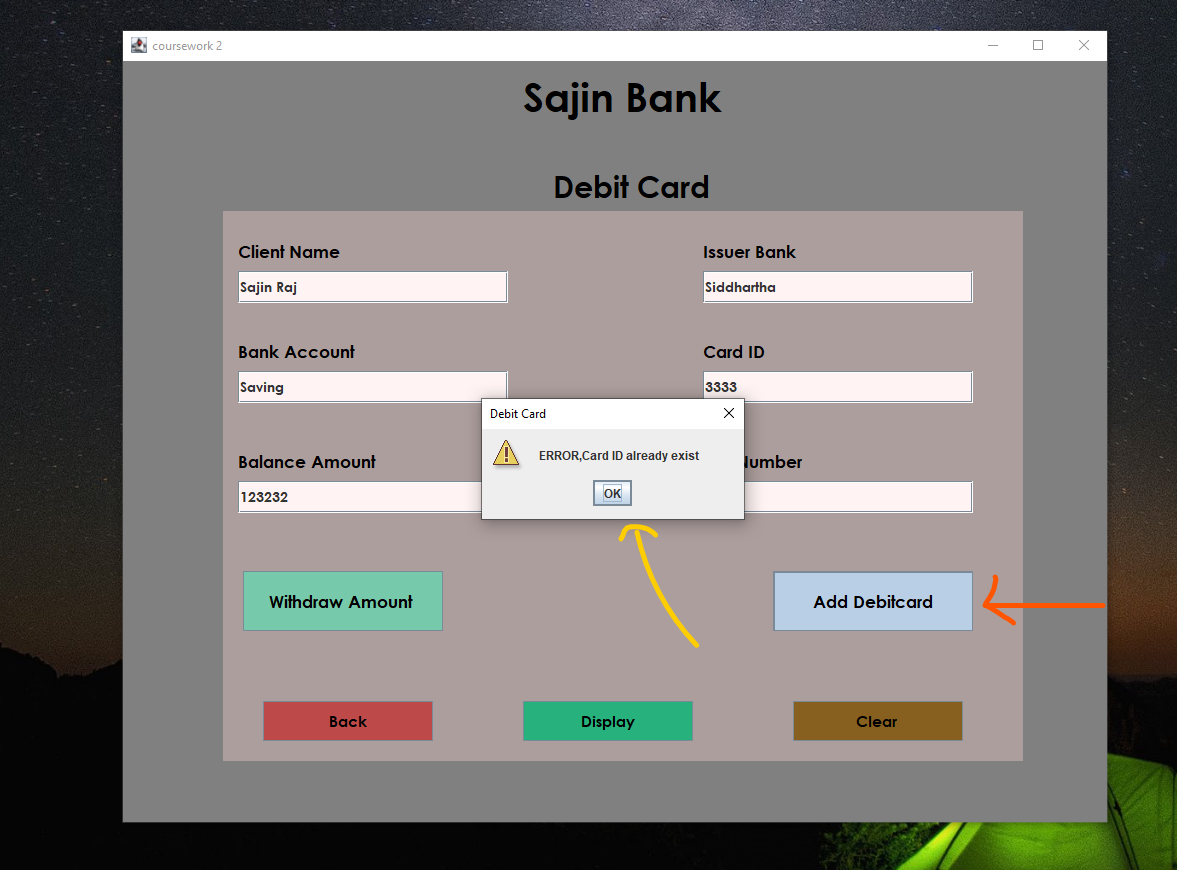


Figure 5: Test 3 Adding debit card with same Card ID

# **Error Detection and Correction**

## **6.1) Syntax Error**

A Syntax error is defined as an error which occurs in the format of a code written by the programmer which is caught by complier in the time of execution. (Rouse, 2017). Example of syntax error are missing semicolon in the end of line, use of keyword as variable, invalid expression etc.

|  |  |
| --- | --- |
| Error No. | 1 |
| Error type | Syntax error |
| Problem | While compiling the code ”Unknown type: viod” error was detected. |
| Solution | Due to the mistyping of keyword the problem was occurred, so correct keyword ‘void’ added to the code. |
| Conclusion | The syntax error was detected and the correction was made to the code. |

Table 7 : Syntax Error

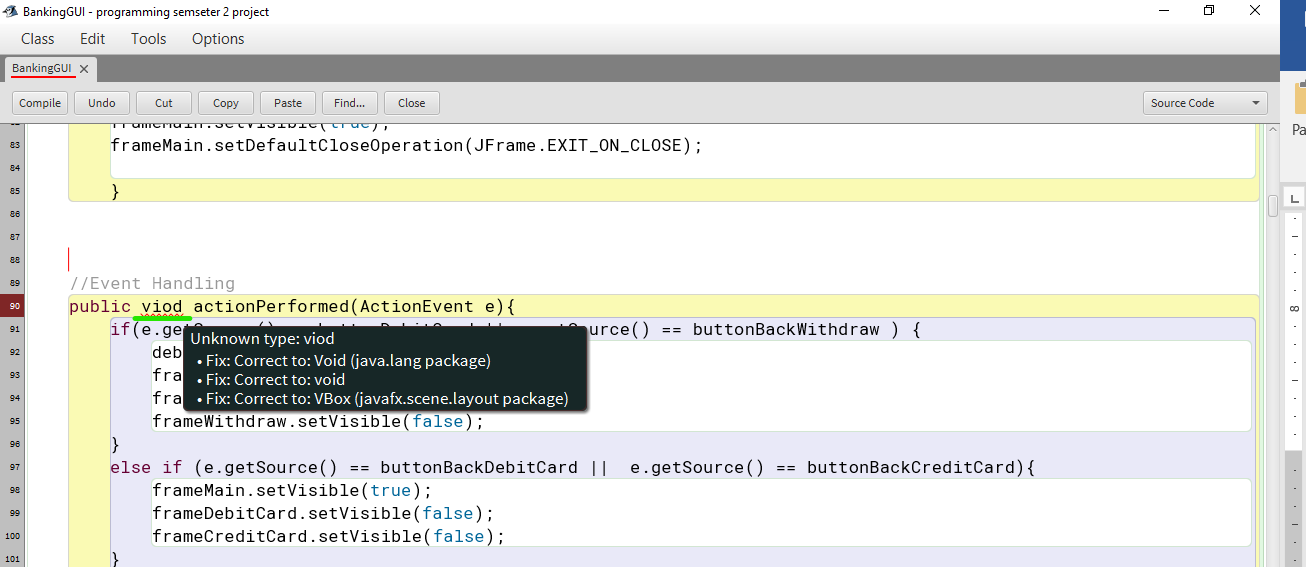


Figure 6: Screenshot of Correction of Syntax Error

Figure 7: Screenshot of Syntax error

## **6.2) Semantic Error**

A semantic error is a type of error in programming which occurs when syntax of block of code is correct but it doesn’t carry out the expected task (ALEX, 2023).

|  |  |
| --- | --- |
| Error No. | 2 |
| Error type | Semantic error |
| Problem | When withdraw button was clicked withdraw GUI fails to open due to incorrect method calling. |
| Solution | Correct method for withdraw GUI was called when withdraw button was clicked. |
| Conclusion | The semantic error was detected and the correction was made to the code. |

Table 8: Semantic Error



Figure 8: Screenshot of Semantic Error



Figure 9: Screenshot of Correction of Semantic Error

## **6.3) Logical Error**

A logical error is a type of error in programming that occurs when the program complies and runs normally without any error, but expected result is not achieved. It cannot be detected but both compiler and Java virtual machine (Singh, 2022).

|  |  |
| --- | --- |
| Error No. | 3 |
| Error type | Logical error |
| Problem | When valid data was entered in the text field of Debit card GUI, Add Debit card button was clicked but, the functionality of button was not working. |
| Solution | In button Add Debit card implementation of action listener was  Added with current object as the listener. |
| Conclusion | The syntax error was detected and the correction was made to the code. |

Table 9 : Logical Error



Figure 10: Screenshot of Logical Error

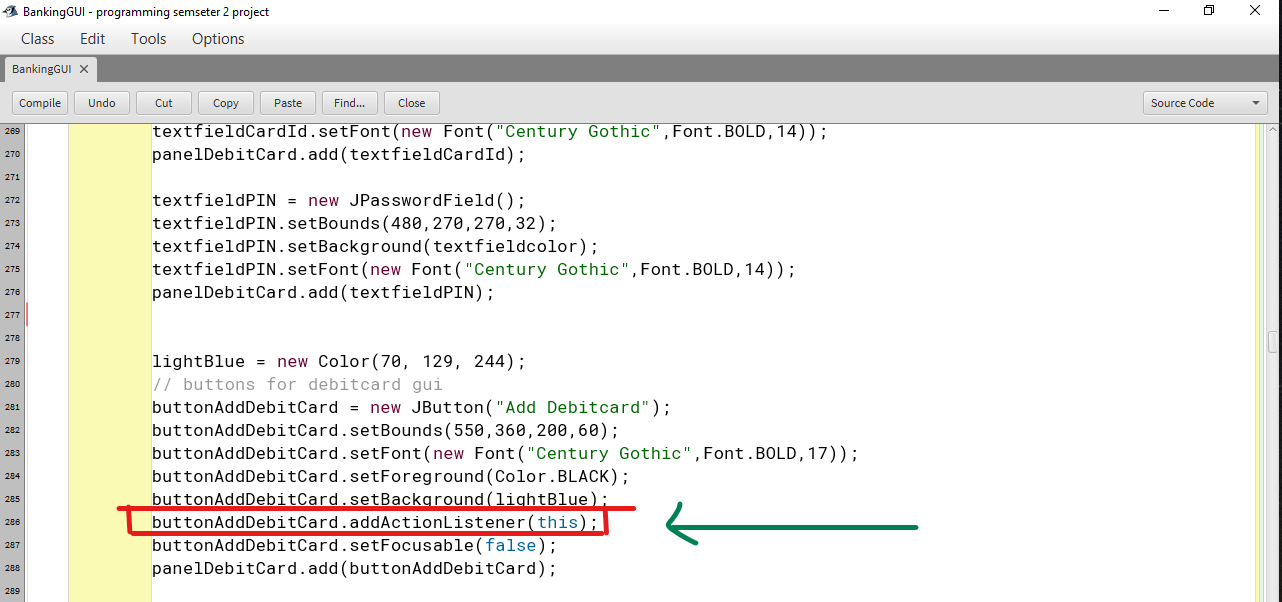


Figure 11: Screenshot of Correction of Logical Error