Module Code & Module Title

CS4001NI Programming

Assessment Weightage & Type 30%

Individual Coursework 2

Year and Semester

2021 - 22 Spring - 1

Student Name: Sharams Kunwar

London Met ID: 21049701

College ID: NP01NT4A210112

Assignment Due Date: 5th August, 2022

Assignment Submission Date: 5th August, 2022

I confirm that I understand my coursework needs to be submitted online via Google Classroom 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

List of Figures

List of Tables

1. Introduction

The project is based on a real-world-problem scenario. Object-Oriented Programming (OOP) Java Programming has been used in order to simplify the complexity of the problem. Similarly, a basic Graphical-User-Interface (GUI) based application has been developed to store details of the instruments and the date they were sold, rented or returned by the customer to ease the data management of SarangiSansar store. At first, I've created a super-class called "Instrument" which can also be called a parent class. Then, the two subclasses or child classes called "InstrumentToRent" and "InstrumentToSell" have been created. A superclass or Parent class helps in building multiple subclasses. The features of the parent class are inherited by the child classes. The process is also referred to as inheritance in Object-oriented Programming. Inheritance is a method of creating a class based on pre-existing class. The child class inherits features from the old class and can also have additional features and functionality on its own.

The software I had taken in use for the completion of the project is BlueJ. It is a Java integrated development environment. It was designed primarily for educational purposes but is also taken in application for small-scale software development. It runs with the assist of Java Development Kit (JDK). Also, an online software called draw.io was used for the making of class diagram. Draw.io is a proprietary program for creation of diagrams and charts. The software was created by Seibert Media. One can use the software's layout feature or can build layout on their own. It consists of large number of shapes and numerous visual elements to help one create a one-of-a-kind diagram. The feature to Drag-and-Drop assists in making a diagram applicable for professional use an easy task.

Also, Graphical User Interfaces has been taken in use which enables users to interact with the program in multiple ways. It eases the environment for the user as the users can alter the items of GUI in various ways when they interact with the interface.

Objectives

The main Objectives of this coursework are mentioned below:

- To extend far down into the knowledge of Java packages like awt and Swing.
- To enable implementation of the Interface Action Listener.
- To store object as values in the array list.
- To be able to create a user friendly and interactive GUI.

2. Class Diagram

A class diagram is a structure diagram which represents the system's structure by representing system's classes, attributes, methods (or operations), objects and relationship among objects. Class Diagram is made for analyzing static view of the program. Also, it outlines the responsibilities of the system and can be used for data modeling. The parent and child classes including SarangiSansar which contains codes of GUI are pasted below:

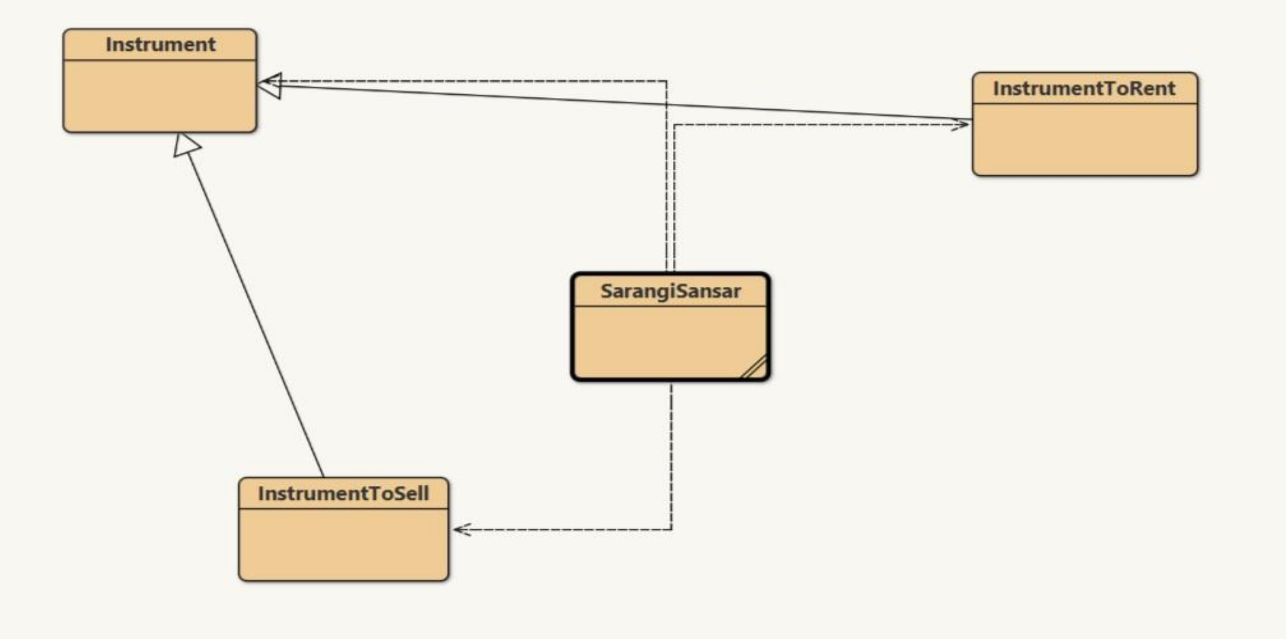
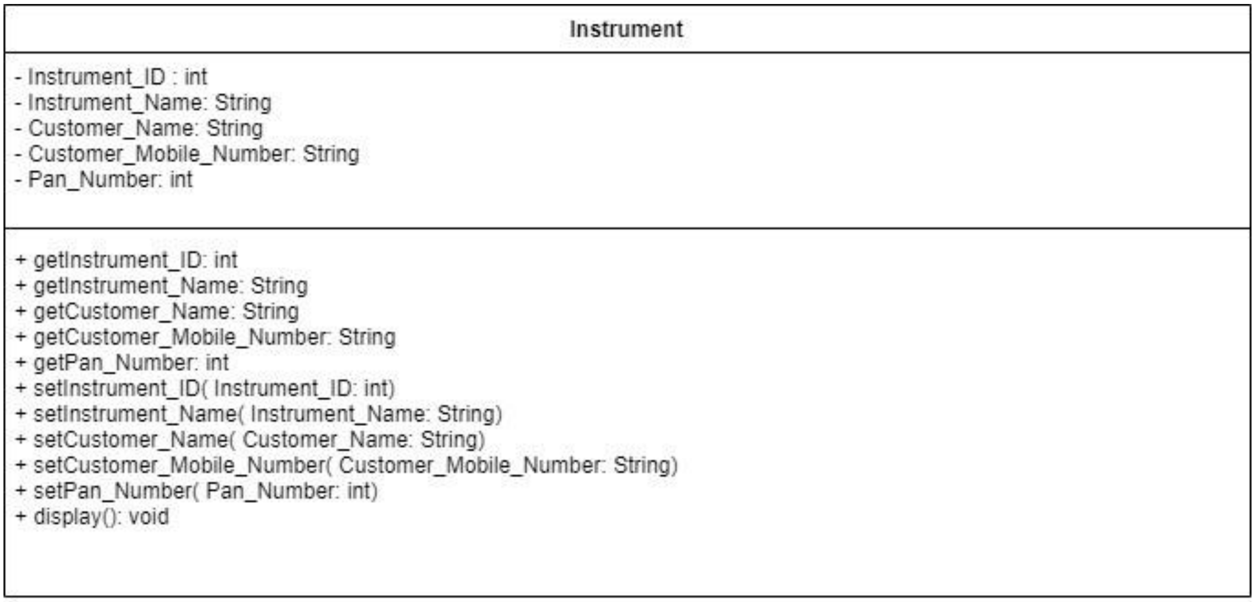


Figure 1: Class Diagram in BlueJ

2.1 Instrument



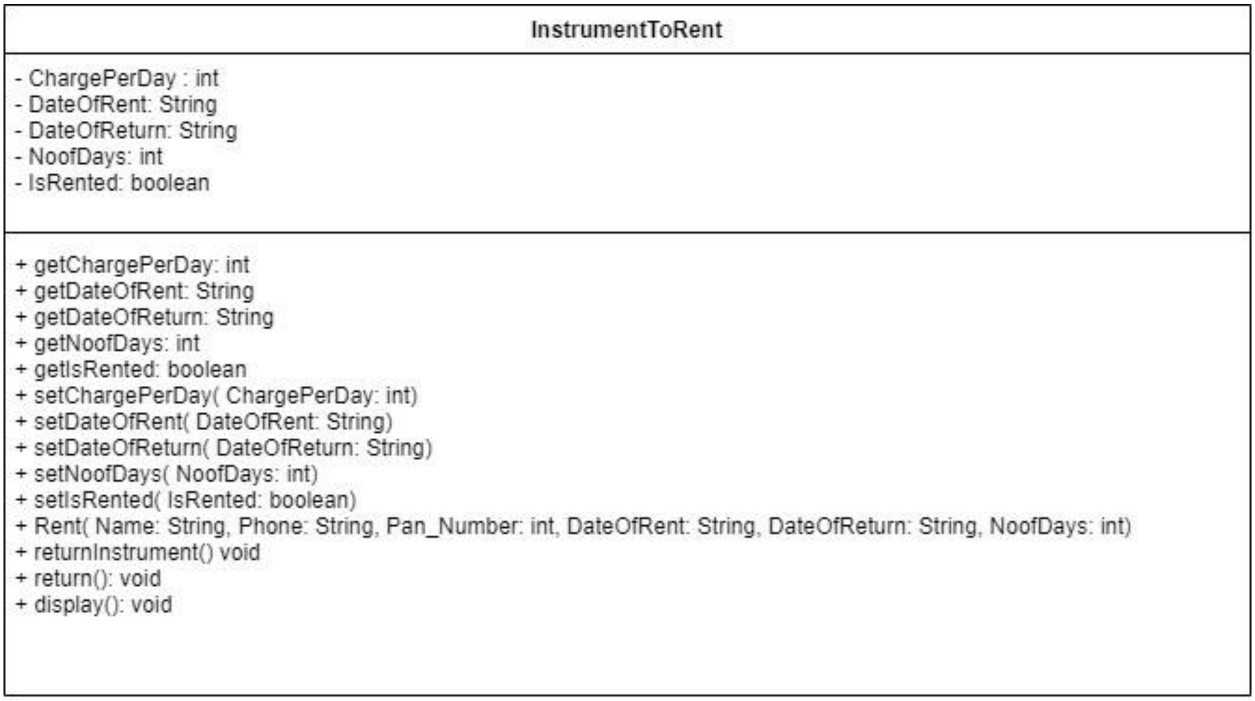
Instrument

- Instrument_ID : int
- Instrument_Name: String
- Customer_Name: String
- Customer_Mobile_Number: String
- Pan_Number: int

+ getInstrument_ID: int
+ getInstrument_Name: String
+ getCustomer_Name: String
+ getCustomer_Mobile_Number: String
+ getPan_Number: int
+ setInstrument_ID(Instrument_ID: int)
+ setInstrument_Name(Instrument_Name: String)
+ setCustomer_Name(Customer_Name: String)
+ setCustomer_Mobile_Number(Customer_Mobile_Number: String)
+ setPan_Number(Pan_Number: int)
+ display(): void

Figure 2: Class Diagram of Instrument Class

2.2 Instrument to Rent



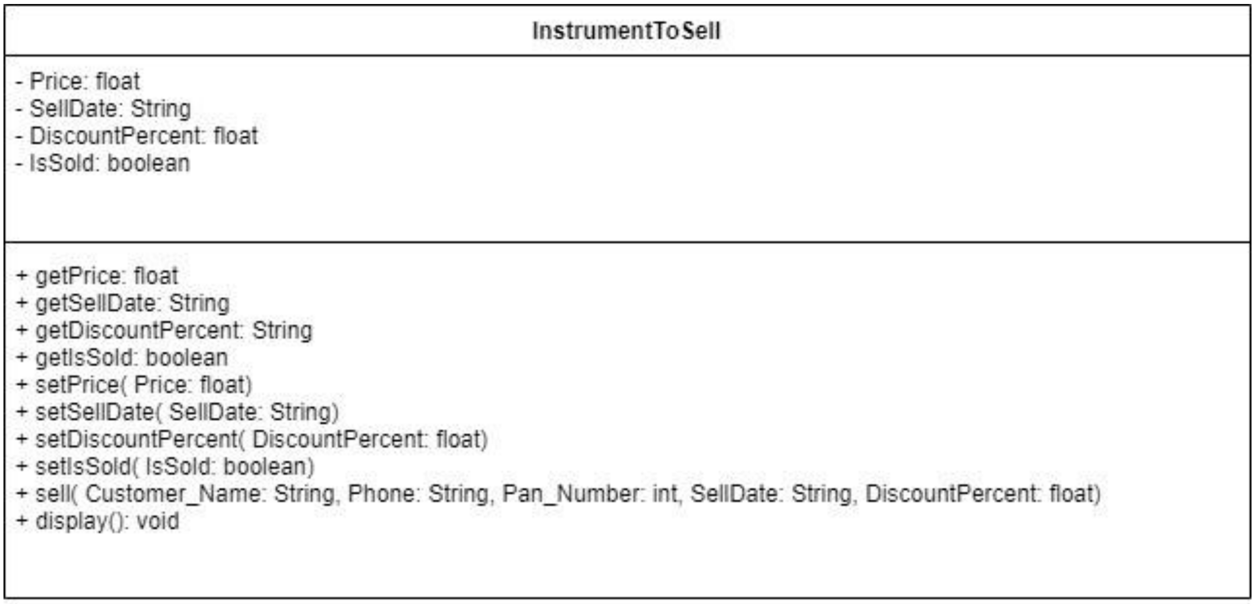
InstrumentToRent

- ChargePerDay : int
- DateOfRent: String
- DateOfReturn: String
- NoofDays: int
- IsRented: boolean

+ getChargePerDay: int
+ getDateOfRent: String
+ getDateOfReturn: String
+ getNoofDays: int
+ getIsRented: boolean
+ setChargePerDay(ChargePerDay: int)
+ setDateOfRent(DateOfRent: String)
+ setDateOfReturn(DateOfReturn: String)
+ setNoofDays(NoofDays: int)
+ setIsRented(IsRented: boolean)
+ Rent(Name: String, Phone: String, Pan_Number: int, DateOfRent: String, DateOfReturn: String, NoofDays: int)
+ returnInstrument() void
+ return(): void
+ display(): void

Figure 3: Class Diagram of Instrument to Rent Class

2.3 Instrument to Sell



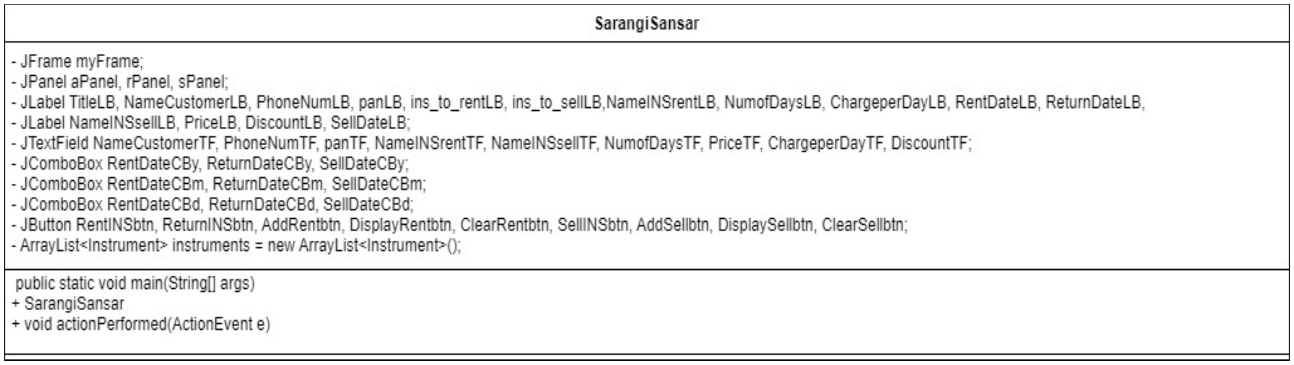
InstrumentToSell

- Price: float
- SellDate: String
- DiscountPercent: float
- IsSold: boolean

+ getPrice: float
+ getSellDate: String
+ getDiscountPercent: String
+ getIsSold: boolean
+ setPrice(Price: float)
+ setSellDate(SellDate: String)
+ setDiscountPercent(DiscountPercent: float)
+ setIsSold(IsSold: boolean)
+ sell(Customer_Name: String, Phone: String, Pan_Number: int, SellDate: String, DiscountPercent: float)
+ display(): void

Figure 4: Class Diagram of Instrument to Sell Class

2.4 Sarangi Sansar



SarangiSansar

- JFrame myFrame;
- JPanel aPanel, rPanel, sPanel;
- JLabel TitleLB, NameCustomerLB, PhoneNumLB, panLB, ins_to_rentLB, ins_to_sellLB,NameINSrentLB, NumofDaysLB, ChargeperDayLB, RentDateLB, ReturnDateLB,
- JLabel NameINSsellLB, PriceLB, DiscountLB, SellDateLB;
- JTextField NameCustomerTF, PhoneNumTF, panTF, NameINSrentTF, NameINSsellTF, NumofDaysTF, PriceTF, ChargeperDayTF, DiscountTF;
- JComboBox RentDateCBy, ReturnDateCBy, SellDateCBy;
- JComboBox RentDateCBm, ReturnDateCBm, SellDateCBm;
- JComboBox RentDateCBd, ReturnDateCBd, SellDateCBd;
- JButton RentINSbtn, ReturnINSbtn, AddRentbtn, DisplayRentbtn, ClearRentbtn, SellINSbtn, AddSellbtn, DisplaySellbtn, ClearSellbtn;
- ArrayList<Instrument> instruments = new ArrayList<Instrument>();

public static void main(String[] args)
+ SarangiSansar
+ void actionPerformed(ActionEvent e)

Figure 5: Class Diagram of Sarangi Sansar Class

2.5 Parents and Child class

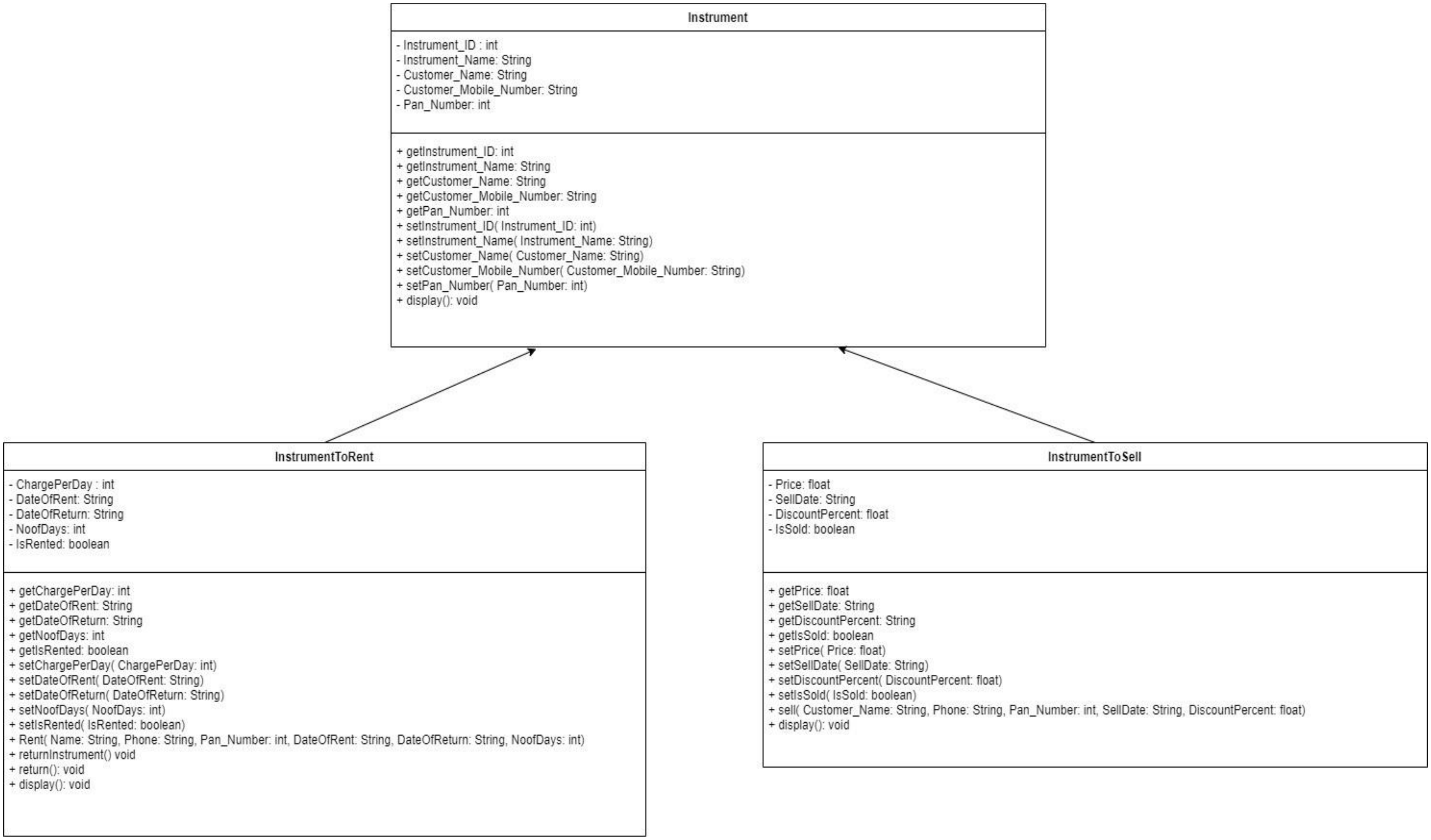

Instrument
- Instrument_ID : int
- Instrument_Name: String
- Customer_Name: String
- Customer_Mobile_Number: String
- Pan_Number: int
+ getInstrument_ID: int
+ getInstrument_Name: String
+ getCustomer_Name: String
+ getCustomer_Mobile_Number: String
+ getPan_Number: int
+ setInstrument_ID(Instrument_ID: int)
+ setInstrument_Name(Instrument_Name: String)
+ setCustomer_Name(Customer_Name: String)
+ setCustomer_Mobile_Number(Customer_Mobile_Number: String)
+ setPan_Number(Pan_Number: int)
+ display(): void
InstrumentToRent
- ChargePerDay : int
- DateOfRent: String
- DateOfReturn: String
- NoofDays: int
- IsRented: boolean
+ getChargePerDay: int
+ getDateOfRent: String
+ getDateOfReturn: String
+ getNoofDays: int
+ getIsRented: boolean
+ setChargePerDay(ChargePerDay: int)
+ setDateOfRent(DateOfRent: String)
+ setDateOfReturn(DateOfReturn: String)
+ setNoofDays(NoofDays: int)
+ setIsRented(IsRented: boolean)
+ Rent(Name: String, Phone: String, Pan_Number: int, DateOfRent: String, DateOfReturn: String, NoofDays: int)
+ returnInstrument() void
+ return(): void
+ display(): void
InstrumentToSell
- Price: float
- SellDate: String
- DiscountPercent: float
- IsSold: boolean
+ getPrice: float
+ getSellDate: String
+ getDiscountPercent: String
+ getIsSold: boolean
+ setPrice(Price: float)
+ setSellDate(SellDate: String)
+ setDiscountPercent(DiscountPercent: float)
+ setIsSold(IsSold: boolean)
+ sell(Customer_Name: String, Phone: String, Pan_Number: int, SellDate: String, DiscountPercent: float)
+ display(): void

Figure 6: Class Diagram of Parents and Child classes

3. Pseudocode

3.1 Sarangi Sansar Class Pseudocode

```
import javax.swing.*;
import java.util.*;
import java.awt.*;
import java.awt.event.*;
import java.awt.color.*;
          **CREATE** class SarangiSansar IMPLEMENTS ActionListner
          **DECLARE** Jframe as myFrame;
          **DECLARE** JPanel as aPanel, rPanel, sPanel;
          **DECLARE** JLabel as TitleLB, NameCustomerLB, PhoneNumLB, panLB, ins_to_rentLB,
                              ins_to_sellLB,NameINSrentLB, NumofDaysLB, ChargeperDayLB,
                              RentDateLB,
                              ReturnDateLB, NameINSsellLB, PriceLB, DiscountLB, SellDateLB;
          **DECLARE** JTextField as NameCustomerTF, PhoneNumTF, panTF, NameINSrentTF,
                              NameINSsellTF, NumofDaysTF, PriceTF, ChargeperDayTF, DiscountTF;
          **DECLARE** JComboBox as RentDateCBy, ReturnDateCBy, SellDateCBy;
          **DECLARE** JComboBox as RentDateCBm, ReturnDateCBm, SellDateCBm;
          **DECLARE** JComboBox as RentDateCBd, ReturnDateCBd, SellDateCBd;
          **DECLARE** JButton as JButton RentINSbtn, ReturnINSbtn, AddRentbtn, DisplayRentbtn,
                              ClearRentbtn, SellINSbtn, AddSellbtn, DisplaySellbtn, ClearSellbtn;
          **CREATE** Instrument: arraylist<Instrument>;
          **CREATE** constructor SarangiSansar()
          **INITIALIZE** JFrame;
          **INITIALIZE** JPanel for aPanel, rPanel, sPanel;
          **INITIALIZE** JLabel for TitleLB, NameCustomerLB, PhoneNumLB, panLB ;
          **INTIALIZE** JLabel for ins_to_rentLB, NameINSrentLB, NumofDaysLB, ChargeperDayLB,
                              RentDateLB, ReturnDateLB;
          **INITIALIZE** JLabel for ins_to_sellLB,NameINSsellLB, PriceLB, DiscountLB, SellDateLB;
          **INITIALIZE** JTextField for NameCustomerTF,PhoneNumTF, panTF;
          **INITIALIZE** JTextField for NameINSsellTF, NumofDaysTF,ChargeperDayTF;
          **INITIALIZE** JTextField for NameINSrentTF, PriceTF, DiscountTF;

          **CREATE** ArrayList and add years, months and days;
          **INITIALIZE** JComboBox as RentDateCBy, RentDateCBm, RentDateCBd;
          **INITIALIZE** JComboBox as ReturnDateCBy, ReturnDateCBm, ReturnDateCBd;
          **INITIALIZE** JComboBox as SellDateCBy, SellDateCBm, SellDateCBd;

          **INITIALIZE** JButton as RentINSbtn, ReturnINSbtn, AddRentbtn, DisplayRentbtn
                              ClearRentbtn;
```

```
INITIALIZE JButton as SellINSbtn, AddSellbtn, DisplayRentbtn, ClearSellbtn;

SET Border Color for myFrame;


SET Bounds for aPanel;
SET Background Color for aPanel;
SET Border Color for aPanel;

SET Bounds for TitleLB;
SET Foreground Color for TitleLB;
SET Background Color for TitleLB;
SET Fonts for TitleLB;
SET Opaque to true for TitleLB;

SET Bounds for NameCustomerLB;
SET Fonts for NameCustomerLB;

SET Bounds for NameCustomerTF;

SET Bounds for PhoneNumLB;
SET Fonts for PhoneNumLB;

SET Bounds for PhoneNumTF;

SET Bounds for panLB;
SET Fonts for panLB;

SET Bounds for panTF;

SET Bounds for rPanel;
SET Background Color for rPanel;
SET Border for rPanel;

SET Bounds for ins_to_rentLB;
SET Foreground Color for ins_to_rentLB;
SET Background Color for ins_to_rentLB;
SET Font for ins_to_rentLB;
SET Opaque to true for ins_to_rentLB;

SET Bounds for NameINSrentLB;
SET Font for NameINSrentLB;
SET Bounds for NumofDaysLB;
SET Font for NumofDaysLB;

SET Bounds for ChargeperDayLB;
SET Font for ChargeperDayLB;

SET Bounds for RentDateLB;
SET Font for RentDateLB;


SET Bounds for ReturnDateLB;
SET Font for ReturnDateLB;

SET Bounds for NameINSrentTF;

SET Bounds for NumofDaysTF;

SET Bounds for ChargeperDayTF;

SET Bounds for RentINSbtn;
SET Foreground Color for RentINSbtn;
SET Background Color for RentINSbtn;
SET Font for RentINSbtn;
SET Opaque to true for RentINSbtn;

SET Bounds for ReturnINSbtn;
SET Foreground Color for ReturnINSbtn;
SET Background Color for ReturnINSbtn;
SET Font for ReturnINSbtn;
SET Opaque to true for ReturnINSbtn;

SET Bounds for AddRentbtn;
SET Foreground Color for AddRentbtn;
SET Background Color for AddRentbtn;
SET Font for AddRenttn;
SET Opaque to true for AddRentbtn;

SET Bounds for DisplayRentbtn;
SET Foreground Color for DisplayRentbtn;
SET Background Color for DisplayRentbtn;
SET Font for DisplayRentbtn;
SET Opaque to true for DisplayRentbtn;
SET Bounds for ClearRentbtn;
SET Foreground Color for ClearRentbtn;
SET Background Color for ClearRentbtn;
SET Font for ClearRentbtn;
SET Opaque to true for ClearRentbtn;

SET Bounds for RentDateCBy;
SET Bounds for RentDateCBm;
SET Bounds for RentDateCBy;

SET Bounds for ReturnDateCBy;
SET Bounds for ReturnDateCBm;
SET Bounds for ReturnDateCBd;

SET Bounds for sPanel;
SET Background Color for sPanel;
SET Border for sPanel;

SET Bounds for ins_to_sellLB;
SET Foreground Color for ins_to_sellLB;
SET Background Color for ins_to_sellLB;
SET Font for ins_to_sellLB;
SET Opaque to true for ins_to_sellLB;

SET Bounds for NameINSsellLB;
SET Font for NameINSsellLB;

SET Bounds for PriceLB;
SET Font for PriceLB;

SET Bounds for DiscountLB;
SET Font for DiscountLB;

SET Bounds for SellDateLB;
SET Font for SellDateLB;

SET Bounds for NameINSsellTF;

SET Bounds for PriceTF;

SET Bounds for DiscountTF;
SET Bounds for SellINSbtn;
SET Foreground Color for SellINSbtn;
SET Background Color for SellINSbtn;
SET Font for SellINSbtn;
SET Opaque to true for SellINSbtn;

SET Bounds for AddSellbtn;
SET Foreground Color for AddSellbtn;
SET Background Color for AddSellbtn;
SET Font for AddSellbtn;
SET Opaque to true for AddSellbtn;

SET Bounds for DisplaySellbtn;
SET Foreground Color for DisplaySellbtn;
SET Background Color for DisplaySellbtn;
SET Font for DisplaySellbtn;
SET Opaque to true for DisplaySellbtn;

SET Bounds for ClearSellbtn;
SET Foreground Color for ClearSellbtn;
SET Background Color for ClearSellbtn;
SET Font for ClearSellbtn;
SET Opaque to true for ClearSellbtn;

SET Bounds for SellDateCBy;
SET Bounds for SellDateCBm;
SET Bounds for SellDateCBy;

ADD aPanel to frame;
ADD rPanel to frame;
ADD sPanel to frame;

ADD TitleLB to aPanel;
ADD NameCustomerLB to aPanel;
ADD PhoneNumLB to aPanel;
ADD panLB to aPanel;

ADD NameCustomerTF to aPanel;
ADD PhoneNumTF to aPanel;
ADD panTF to aPanel;
ADD ins_to_rentLB to rPanel;
ADD NameINSrentLB to rPanel;
ADD NumofDaysLB to rPanel;
ADD ChargeperDayLB to rPanel;
ADD RentDateLB to rPanel;
ADD ReturnDateLB to rPanel;

ADD NameINSrentTF to rPanel;
ADD NumofDaysTF to rPanel;
ADD ChargeperDayTF to rPanel;

ADD RentDateCBy to rPanel;
ADD RentDateCBm to rPanel;
ADD RentDateCBd to rPanel;
ADD ReturnDateCBy to rPanel;
ADD ReturnDateCBm to rPanel;
ADD ReturnDateCBd to rPanel;

ADD RentINSbtn to rPanel;
ADD ReturnINSbtn to rPanel;
ADD AddRentbtn to rPanel;
ADD DisplayRentbtn to rPanel;
ADD ClearRentbtn to rPanel;

ADD ins_to_sellLB to sPanel;
ADD NameINSsellLB to sPanel;
ADD PriceLB to sPanel;
ADD DiscountLB to sPanel;
ADD SellDateLB to sPanel;

ADD NameINSsellTF to sPanel;
ADD PriceTF to sPanel;
ADD DiscountTF to sPanel;

ADD SellDateCBy to sPanel;
ADD SellDateCBm to sPanel;
ADD SellDateCBd to sPanel;

ADD SellINSbtn to sPanel;
ADD AddSellbtn to sPanel;
ADD DisplaySellbtn to sPanel;
ADD ClearSellbtn to sPanel;
ADD ActionListener to RentINSbtn;
ADD ActionListener to ReturnINSbtn;
ADD ActionListener to AddRentbtn;
ADD ActionListener to DisplayRentbtn;
ADD ActionListener to ClearRentbtn;
ADD ActionListener to SellINSbtn;
ADD ActionListener to AddSellbtn;
ADD ActionListener to DisplaySellbtn;
ADD ActionListener to ClearSellbtn;

SET Size of myFrame;
SET Layout of myFrame;
SET Layout of aPanel;
SET Layout of rPanel;
SET Layout of sPanel;

SET DefaultCloseOperation of myFrame;
SET Visibility of myFrame;

CREATE main method
    CREATE a new object obj;

CREATE a method actionPerformed with ActionEvent e as the parameter
    IF the source of event is AddRentbtn
        IF the NameINSrentTF and ChargeperDayTF are empty
            DISPLAY message;
        END IF
        ELSE
            TRY
            CREATE the String INSname variable and store the value
                entered in NameINSrentTF;
            CREATE the String ChargePday variable and store the value
                entered in ChargeperDayTF;
            CREATE the Integer Charge variable and convert the value
                of ChargePday to integer and store the value;
            CREATE the Boolean variable and set its value to false;
            IF ArrayList instruments is empty
                CREATE object INSrent and send INSname, Charge as the
                parameter;
                IF INSrent is instanceof InstrumentToRent
                    ADD INSrent in instruments ArrayList;
                    DISPLAY message;
                    END IF
            END IF
            ELSE
                FOR each loop in instruments
                    CHECK if instrument's object's been set in
                    InstrumentToRent
                IF isAvailable is true
                    DISPLAY message;
                END IF
                ELSE
                    CREATE object INSrent and pass INSname, Charge
                    as parameters.
                    IF INSrent is instance of InstrumentToRent
                    ADD INSrent to instruments;
                    DISPLAY message;
                    END IF
            CATCH NumberFormatException except
                DISPLAY message;

    IF the source of event is AddSellbtn
        IF the NameINSSellTF and PriceTF are empty
            DISPLAY message;
        END IF
        ELSE
            TRY
            CREATE the String INSname variable and store the value
                entered in NameINSsellTF;
            CREATE the String Prices variable and store the value
            entered in PriceTF;
            CREATE the Float pr variable and convert the value
            of Prices to integer and store the value;
            CREATE the Boolean variable and set its value to false;
            IF ArrayList instruments is empty
                CREATE object INSsell and send pr, INSname as the
                parameter;
                IF INSsell is instanceof InstrumentToRent
                    ADD INSsell in instruments ArrayList;
                    DISPLAY message;
                    END IF
            END IF
            ELSE
                FOR each loop in instruments
                    CHECK if instrument's object's been set in
                    InstrumentToSell
                IF isAvailable is true
                    DISPLAY message;
                END IF
                ELSE
                    CREATE object INSsell and pass pr, INSname as
                    parameters.
                    IF INSsell is instance of InstrumentToSell
                    ADD INSsell to instruments;
                    DISPLAY message;
                    END IF
            CATCH NumberFormatException except
                DISPLAY message;


    IF the source of the button is RentINSbtn
        IF NameCustomerTF, PhoneNumTF, panTF, NameINSrentTF,
        ChargeperDayTF is empty
            DISPLAY message;
        END IF
        ELSE
            TRY
            CREATE the String CName variable and store the value
            entered in NameCustomerTF;
            CREATE the String Pnum variable and store the value
            entered in PhoneNumTF;
            CREATE the String panNo variable and store the value
            entered in panTF;
            CREATE the String INSname variable and store the value
            entered in NameINSrentTF;
            CREATE the String ChargePDay variable and store the value
            entered in ChargeperDayTF;
            CREATE the String DaysNo variable and store the value
            entered in NumofDaysTF;
            CREATE the String DateReturn variable and convert the value
            entered in ReturnDateCBy, ReturnDateCBm, ReturnDateCBd to
            String and store values;
            CREATE the String DateRent variable and convert the value
            entered in RentDateCBy, RentDateCBm, RentDateCBd to String
            and store values;
            CREATE the Integer PAN variable and convert the value
            entered in panNo to Integer and store values;
            CREATE the Integer Charge variable and convert the value
            entered in ChargePday to Integer and store values;
```

```
                CREATE the Integer NOdays variable and convert the value
                entered in DaysNo to Integer and store values;

                IF instruments is empty
                    DISPLAY message;
                END IF
                ELSE
                    FOR each loop in instruments
                        IF instrument is instanceof InstrumentToRent
                        CREATE new object and cast
                            IF newly created object is available in INSname
                                IF the object is already rented
                                    DISPLAY message;
                                    CALL Rent method and pass
                                    CName, PNum, PAN, DateRent,
                                    DateReturn, NOdays;
                                END IF
                                ELSE
                                    DISPLAY message;
                                    CALL Rent method and pass
                                    CName, PNum, PAN, DateRent,
                                    DateReturn, NOdays;
                                    SET setter method setIsRented()
                                    as Boolean value true
                        END IF
            CATCH NumberFormatException except
                DISPLAY message;


        IF the source of the button is SellINSbtn
            IF NameCustomerTF, PhoneNumTF, panTF, NameINSsellTF, PriceTF is
            empty
                DISPLAY message;
            END IF
            ELSE
                TRY
                CREATE the String CName variable and store the value
                entered in NameCustomerTF;
                CREATE the String PNum variable and store the value
                entered in PhoneNumTF;
                CREATE the String panNo variable and store the value
                entered in panTF;
                CREATE the String INSname variable and store the value
                entered in NameINSsellTF;
                CREATE the String Discount variable and store the value
                entered in DiscountTF;
                CREATE the String Price variable and store the value
                entered in PriceTF;
                CREATE the String DateSell variable and convert the value entered
                in SellDateCBy, SellDateCBm, SellDateCBd to String and store
                values;
                CREATE the Integer PAN variable and convert the value
                entered in panNo to Integer and store values;
                CREATE the Float Disc variable and convert the value
                entered in Discount to Float and store values;
                CREATE the Float Pr variable and convert the value
                entered in Price to Float and store values;

                IF instruments is empty
                    DISPLAY message;
                END IF
                ELSE
                    FOR each loop in instruments
                        IF instrument is instanceof InstrumentToSell
                        CREATE new object and cast
                            IF newly created object is available in INSname
                                IF the object is already sold
                                    DISPLAY message;
                                    CALL sell method and pass
                                    CName, PNum, PAN, DateSell,
                                    Disc;
                                END IF
                                ELSE
                                    DISPLAY message;
                                    CALL Sell method and pass
                                    CName, PNum, PAN, DateSell,
                                    Disc;
                                    SET setter method setisSold() as
                                    Boolean value true
                        END IF
            CATCH NumberFormatException except
                DISPLAY message;
        IF the source of the button is ReturnINSbtn
            IF NameINSrentTF is empty
                DISPLAY message;
            END IF
            ELSE
                TRY
                CREATE the String INSname variable and store the value
                entered in NameINSrentTF;
                IF instruments is empty
                    DISPLAY message;
                END IF
                ELSE
                    FOR each loop in instruments
                        IF instrument is instanceof InstrumentToRent
                        CREATE new object and cast
                            IF newly created object is available in INSname
                                IF the object is already rented
                                    DISPLAY message;
                                    SET IsRented as Boolean false;
                                END IF
                                ELSE
                                    DISPLAY message;
                        END IF


    IF the source of the button is DisplayRentbtn
            IF NameCustomerTF and PhoneNumTF and panTF and NameINSrentTF
            and ChargeperDayTF is empty
                DISPLAY message;
            END IF
            ELSE
                FOR each loop in instruments
                    IF inst is instance of InstrumentToRent
                        CREATE a new object rents of InstrumentToRent
                        taking reference variable as inst;
                        CALL display method for object rent;
                    END IF
    END IF
```

```
IF the source of the button is DisplaySellbtn
        IF NameCustomerTF and PhoneNumTF and panTF and NameINSsellTF
        and PriceTF and DiscountTF is empty
            DISPLAY message;
        END IF
        ELSE
            FOR each loop in instruments
                IF inst is instance of InstrumentToRent
                    CREATE a new object sells of InstrumentToSell taking
                    reference variable as inst;
                    CALL display method for object sells;
                END IF
END IF

IF the source of the button is ClearRentbtn
        IF NameCustomerTF and PhoneNumTF and panTF and NameINSrentTF
        and ChargeperDayTF is empty
            DISPLAY message;
        END IF
        ELSE
            EMPTY all textfields;
            DISPLAY message;

IF the source of the button is ClearSellbtn
        IF NameCustomerTF and PhoneNumTF and panTF and NameINSsellTF
        and PriceTF and DiscountTF is empty
            DISPLAY message;
        END IF
        ELSE
            EMPTY all textfields;
            DISPLAY message;
```

4. Methods description

4.1 Method description of Sarangi Sansar

Name of the Method	**Method Description**
actionPerformed	The getActionCommand method of the actionevent object is taken in use to ascertain the triggering the elements in the program depending upon the buttons pressed.
AddRentbtn	On pressing this button, an object of InstrumentToRent is created using the inputted value in the text field of ‘Instrument Name’ and ‘Charge Per Day’ which is later added to the array list in Instrument class.
RentINSBtn	On pressing this button, the inputted value in the text fields of ‘Instrument Name’ is compared to the existing ones to check the validity of the instrument’s name and the appropriate instrument is rented from the list. Also, the inputted value of ‘Customer Name’, ‘Phone Number’, ‘Customer PAN’, ‘Rent Date’, ‘Return Date’ and ‘Number of Days’ are taken in use for calling the method rent from InstrumentToRent class.
AddSellBtn	On pressing this button, an object of InstrumentToSell is created using the inputted value in the text field of ‘Instrument Name’ and ‘Price’ which is later added to the array list in Instrument class.
SellINSBtn	On pressing this button, the inputted value in the text fields of ‘Instrument Name’ is compared to the existing ones to check the validity of the instrument’s name and the appropriate instrument is rented from the list. Also, the inputted value of ‘Customer Name’, ‘Phone Number’, ‘Customer PAN’, ‘Sell Date’, and ‘Discount Percent’ are taken in use for calling the method sell from InstrumentToSell class.

ReturnINSBtn	On pressing this button, the inputted value in the text fields of ‘Instrument Name’ is compared to the existing ones to check if it is rented and the validity of the name and then returnInstrument and InstrumentToRent is called.
ClearRentBtn	When this button is pressed the values of text fields are cleared from rpanel.
ClearSellBtn	When this button is pressed the values of text fields are cleared from spanel.
DisplayRentBtn	On pressing this button, the information related to rent is displayed and the display method from InstrumentToRent is called.
DisplaySellBtn	On pressing this button, the information related to sell is displayed and the display method from InstrumentToSell is called.

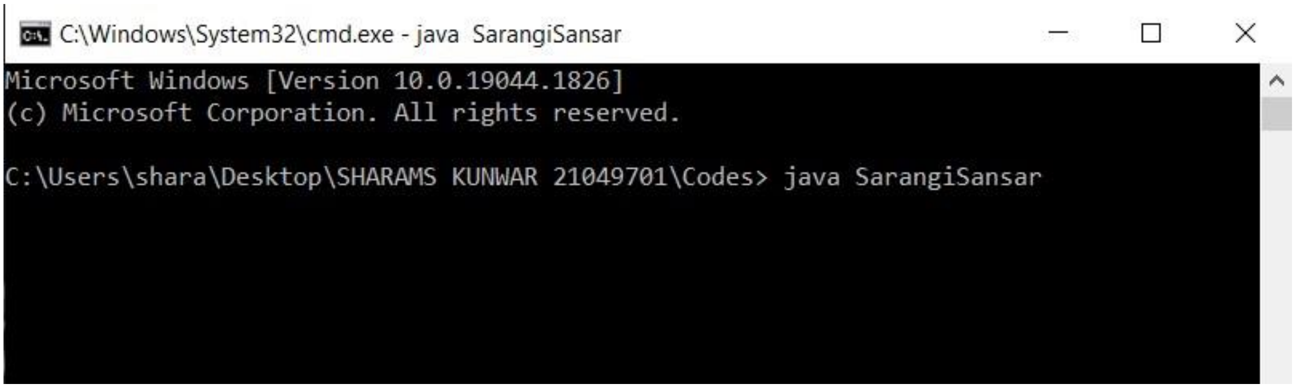
5. Testing

In order to create an error-free and much reliable code, testing of the code is a must. The codes are split and then accordingly tested to ensure an efficient manner rather than discovering the error in an entirety of the program. Testing helps in detecting the bugs and ensure that the program runs as it is supposed to and function accordingly.

5.1 Test 1: Program can be compiled and run using the command prompt.

Test No	1
Objective	**To show that the program can be compiled and run using the command prompt.**
Action	Command Prompt was opened and SarangiSansar class was compiled and run
Expected Result	The class SarangiSansar would compile and run displaying GUI.
Actual Result	The program complied and ran using command prompt.
Conclusion	The test was successful.

Table 1: Test Table 1



```
Microsoft Windows [Version 10.0.19044.1826]
(c) Microsoft Corporation. All rights reserved.

C:\Users\shara\Desktop\SHARAMS KUNWAR 21049701\Codes> java SarangiSansar
```

Figure 7: Screenshot of cmd prompt

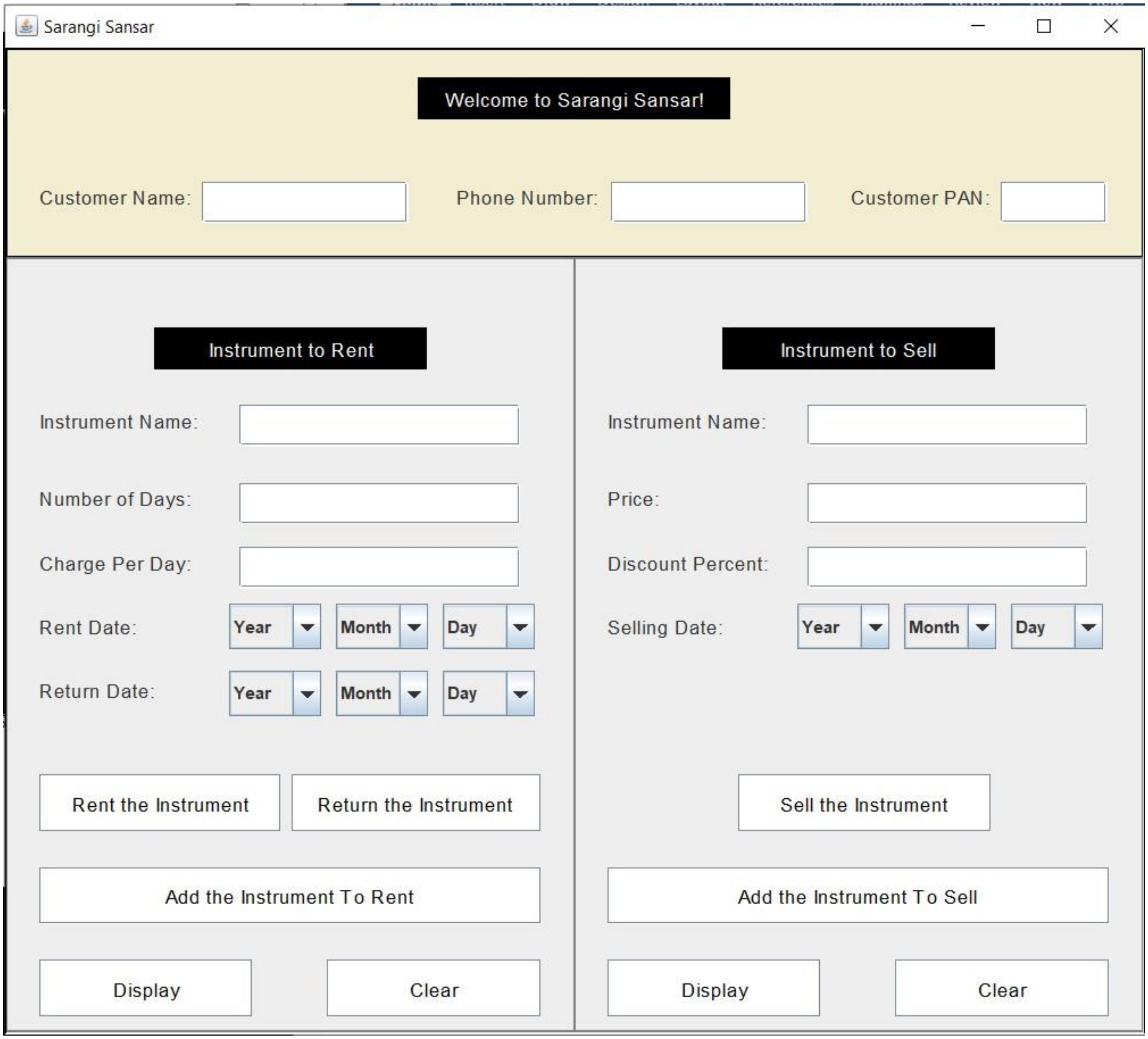


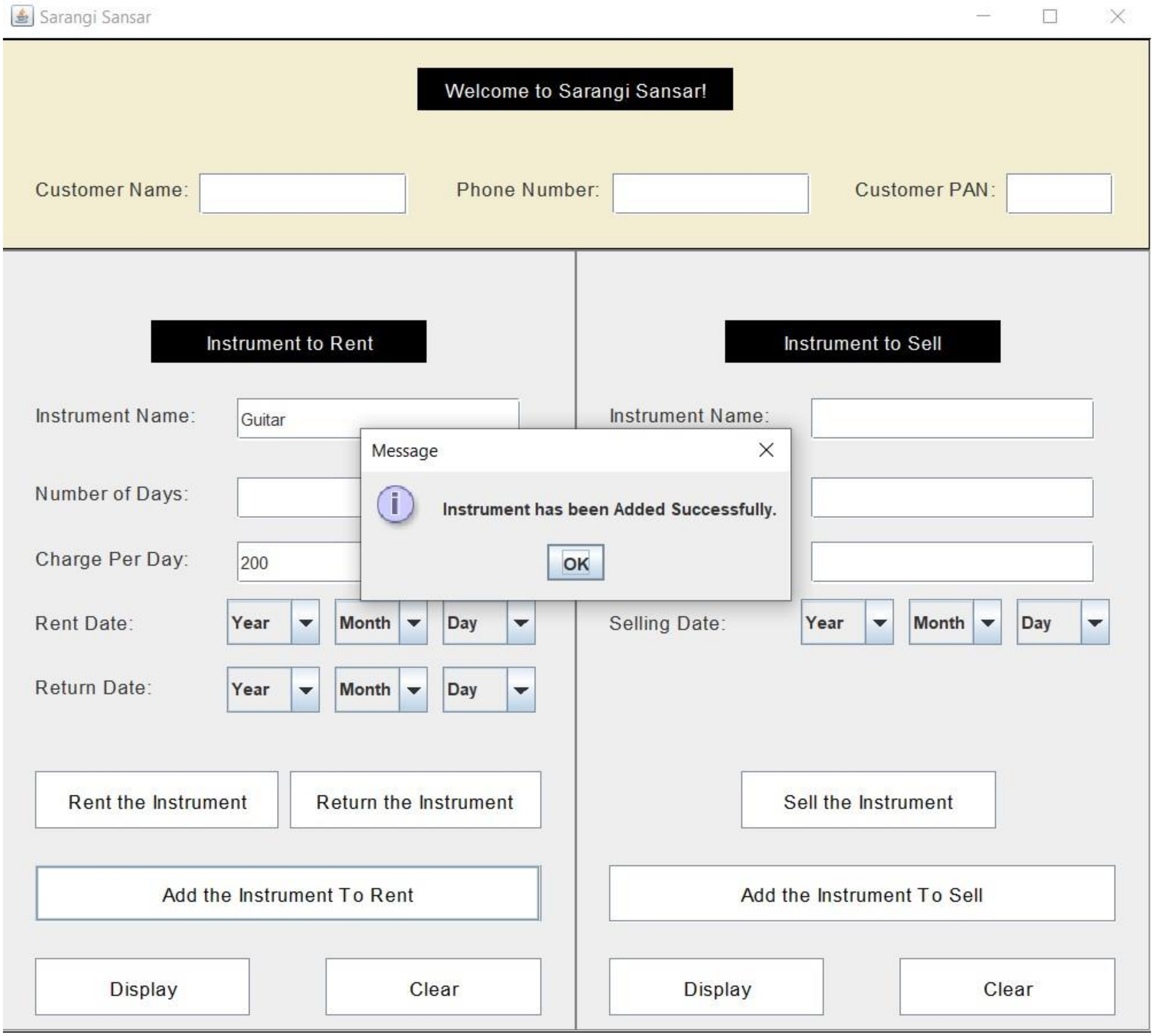
Figure 8: Screenshot of GUI opened using cmd

5.2 Test 2: Adding objects:

a. Test that the program can be Add instrument for rent.

Test No	**A**
Objective	**To test that the program can be Add instrument for rent.**
Action	Values were inserted in the respective text fields to add the Instrument for rent.
Expected Result	Instrument would be added for rent.
Actual Result	Instrument was added to rent and pop-up was displayed.
Conclusion	The test was successful.

Table 2: Test Table 2 (a)



Instrument Name: Guitar

Charge Per Day: 200

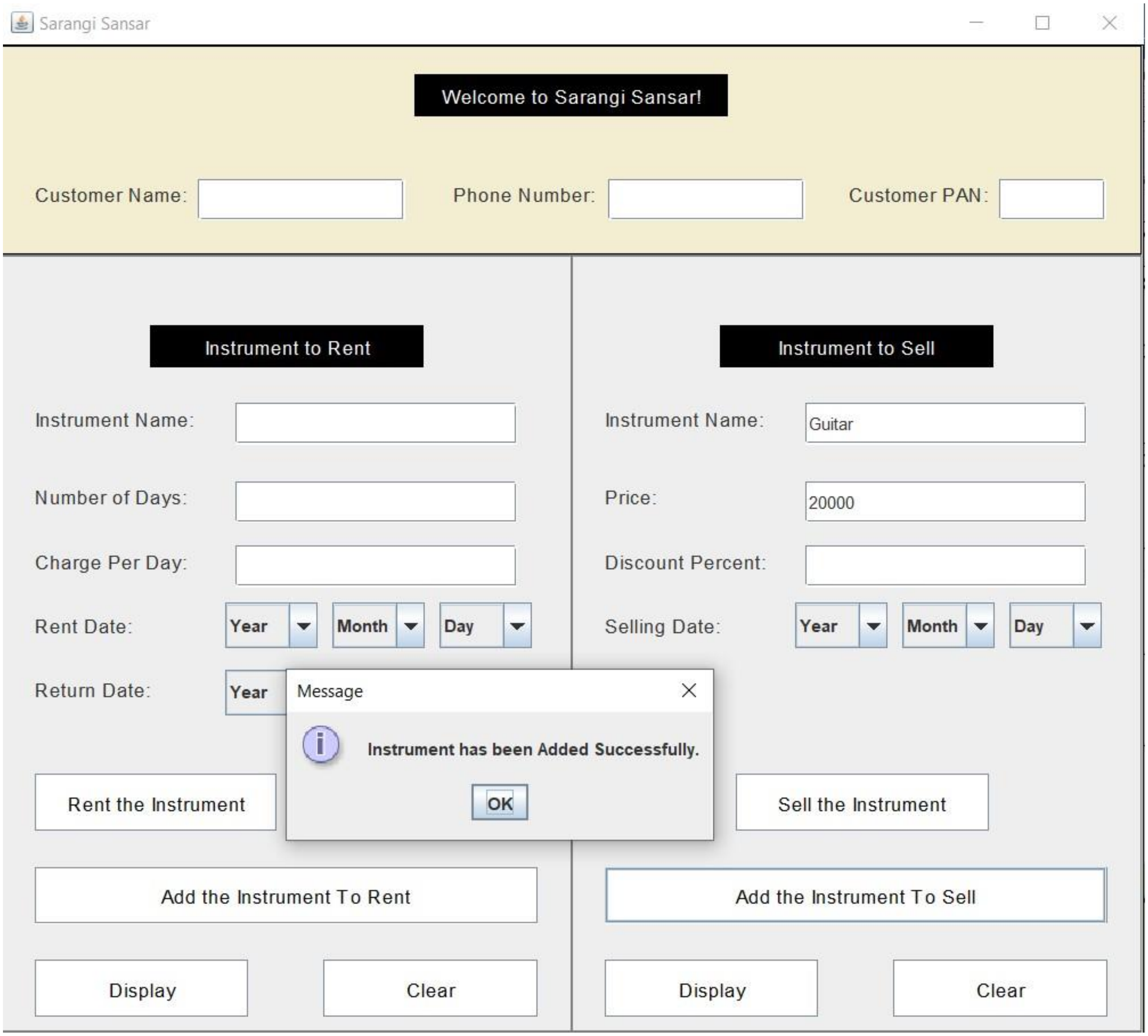
Instrument has been Added Successfully.

Figure 9: Screenshot of Adding Instrument for Rent

b. Test that program can be Add Instrument for Sell:

Test No	**B**
Objective	**To test that the program can be Add instrument for Sell**
Action	Values were inserted in the respective text fields to add the Instrument for sell.
Expected Result	Instrument would be added for sell.
Actual Result	The instrument was added for sell and pop-up was displayed.
Conclusion	The test was successful.

Table 3: Test Table 2(b)



Instrument Name: Guitar

Price: 20000

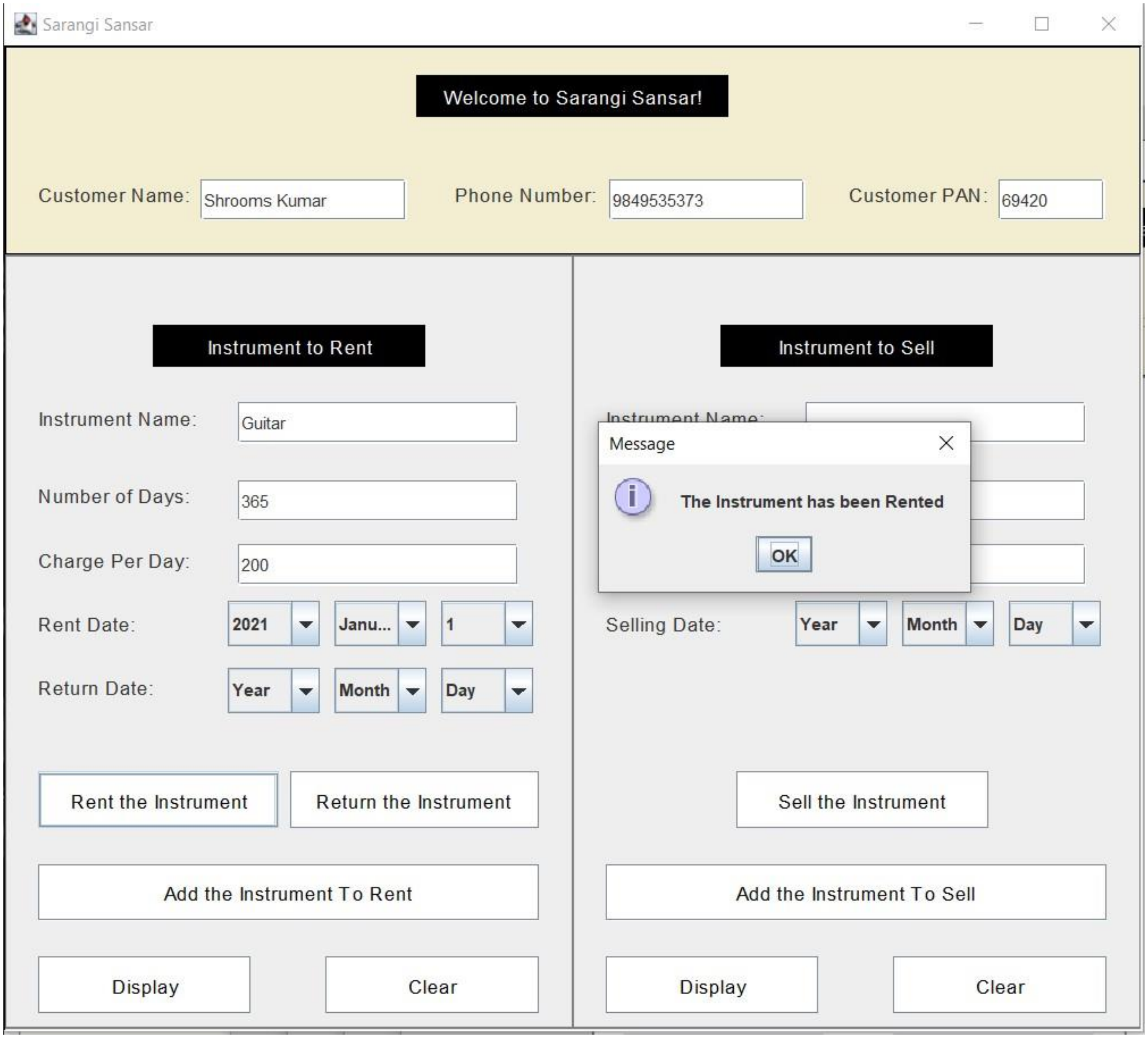
Instrument has been Added Successfully.

Figure 10: Screenshot of Adding Instrument for Sell

c. To test that program that can Rent the Instrument.

Test No	**C**
Objective	**To test that program that can Rent the Instrument.**
Action	Values were inserted in the respective text fields to Rent the Instrument.
Expected Result	Instrument was Rented.
Actual Result	The instrument was rented and pop-up was displayed.
Conclusion	The test is successful.

Table 4: Test Table 2(c)



Customer Name: Shrooms Kumar

Phone Number: 9849535373

Customer PAN: 69420

Instrument Name: Guitar

Number of Days: 365

Charge Per Day: 200

Rent Date: 2021 Janu... 1

Message

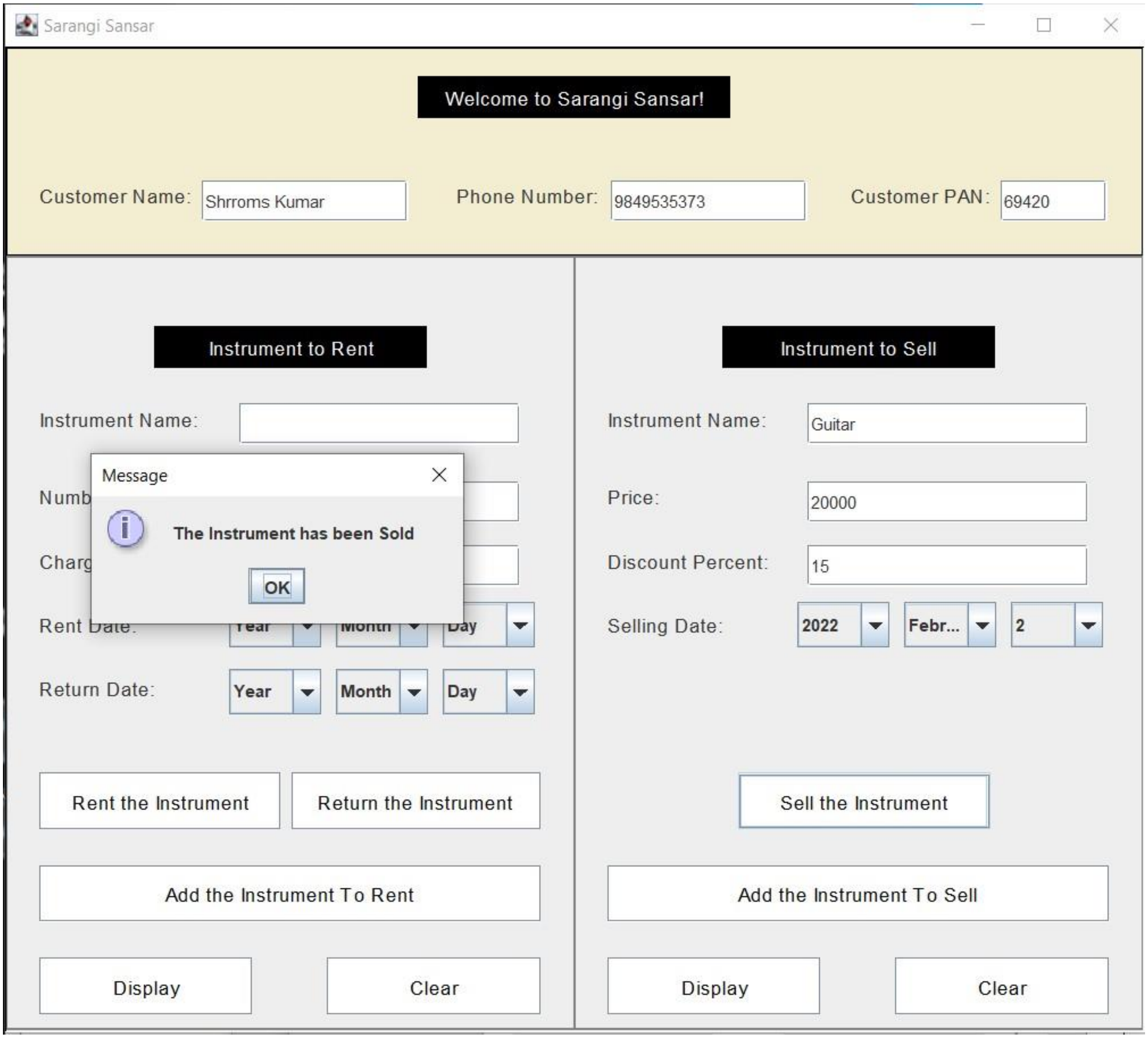
The Instrument has been Rented

Figure 11: Screenshot of Renting the Instrument

d. To test that program that can Sell the Instrument.

Test No	D
Objective	**To test that program that can Sell the Instrument.**
Action	Values were inserted in the respective text fields to Sell the Instrument.
Expected Result	Instrument was Sold.
Actual Result	The instrument was sold and pop-up was displayed.
Conclusion	The test is successful.

Table 5: Test Table 2(d)



Sarangi Sansar

Welcome to Sarangi Sansar!

Customer Name: Shrroms Kumar

Phone Number: 9849535373

Customer PAN: 69420

Instrument to Sell

Instrument Name: Guitar

Price: 20000

Discount Percent: 15

Selling Date: 2022 Febr... 2

Message

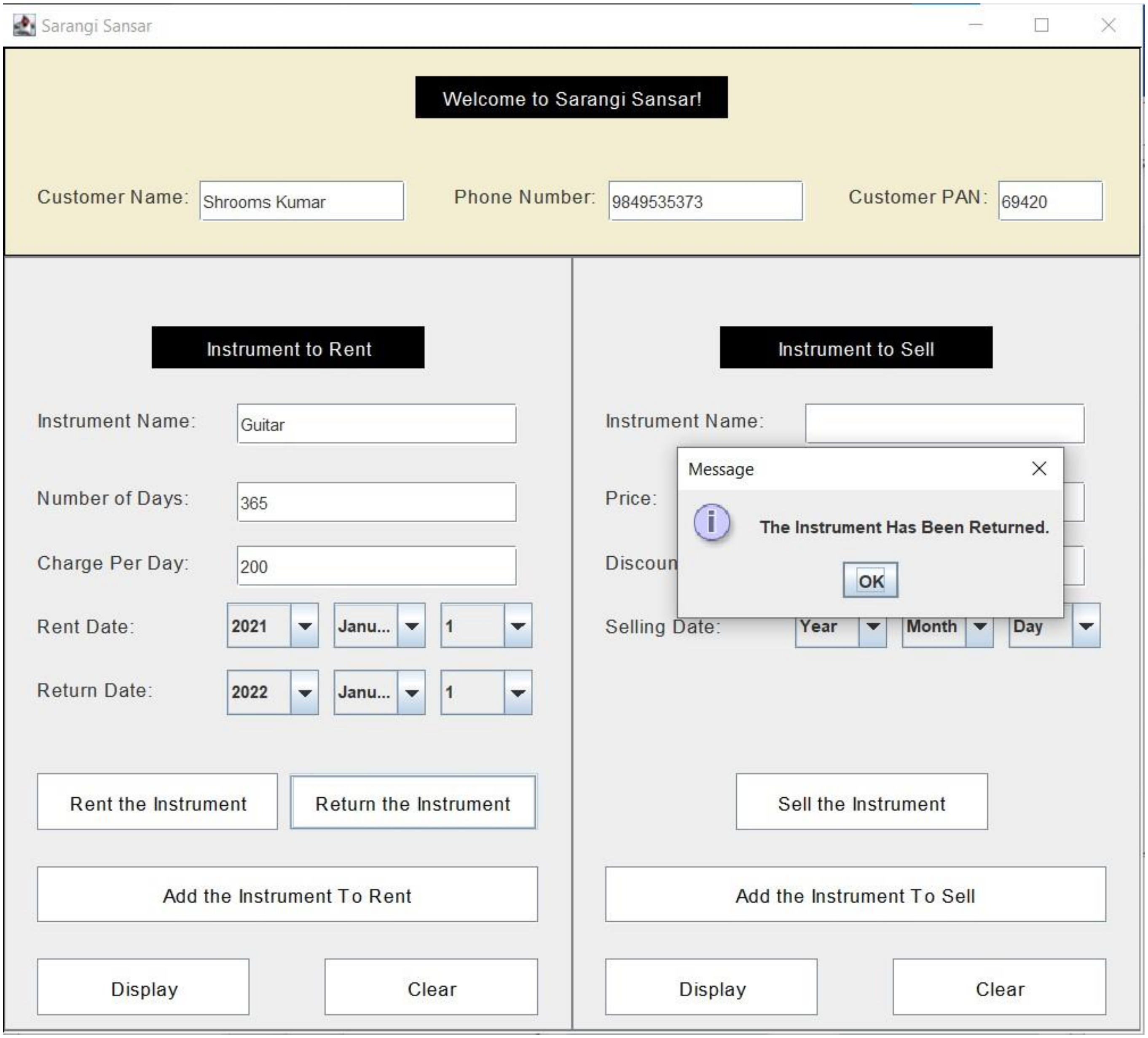
The Instrument has been Sold

Figure 12: Screenshot of Selling the Instrument

e. To test that program that can Return the Instrument.

Test No	**E**
Objective	**To test that program that can Return the Instrument.**
Action	Values were inserted in the respective text fields to Return the Instrument.
Expected Result	Instrument was Returned.
Actual Result	The instrument was returned successfully and pop-up was displayed.
Conclusion	The test is successful.

Table 6: Test Table 2(e)



Customer Name: Shrooms Kumar

Phone Number: 9849535373

Customer PAN: 69420

Instrument Name: Guitar

Number of Days: 365

Charge Per Day: 200

Rent Date: 2021 Janu... 1

Return Date: 2022 Janu... 1

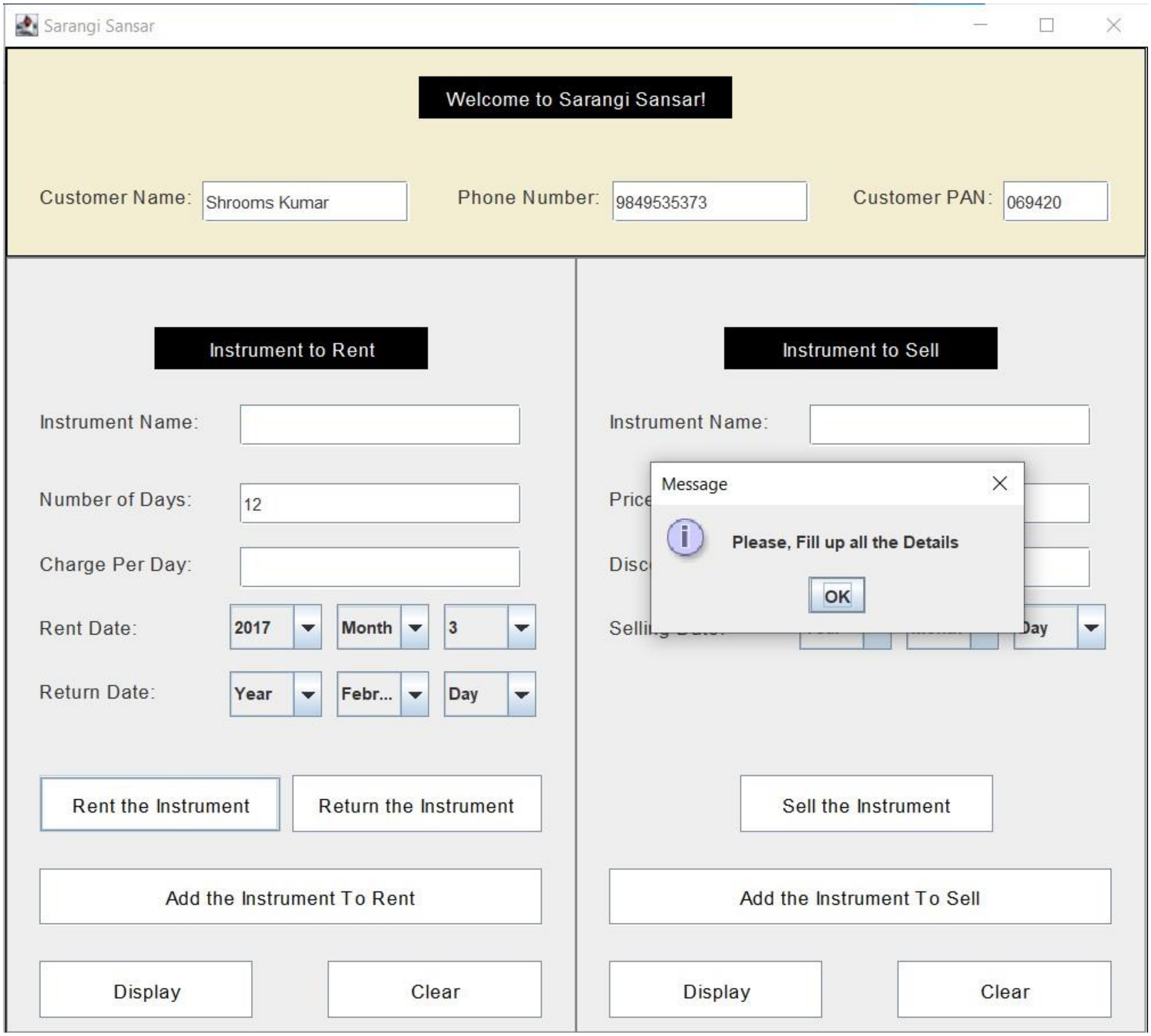
The Instrument Has Been Returned.

Figure 13: Screenshot of Returning the Instrument

5.3 Test 3: Test that appropriate dialog boxes appear when required fields are left unfilled.

Test No	**3**
Objective	**Test that appropriate dialog boxes appear when required fields are left unfilled**.
Action	Few text fields were left unfilled to rent instrument.
Expected Result	Error message would pop up.
Actual Result	Pop-up message was displayed asking user to fill all the details.
Conclusion	The test is successful.

Table 7: Test Table 3



Customer Name: Shrooms Kumar

Phone Number: 9849535373

Customer PAN: 069420

Number of Days: 12

Rent Date: 2017 Month 3

Return Date: Year Febr... Day

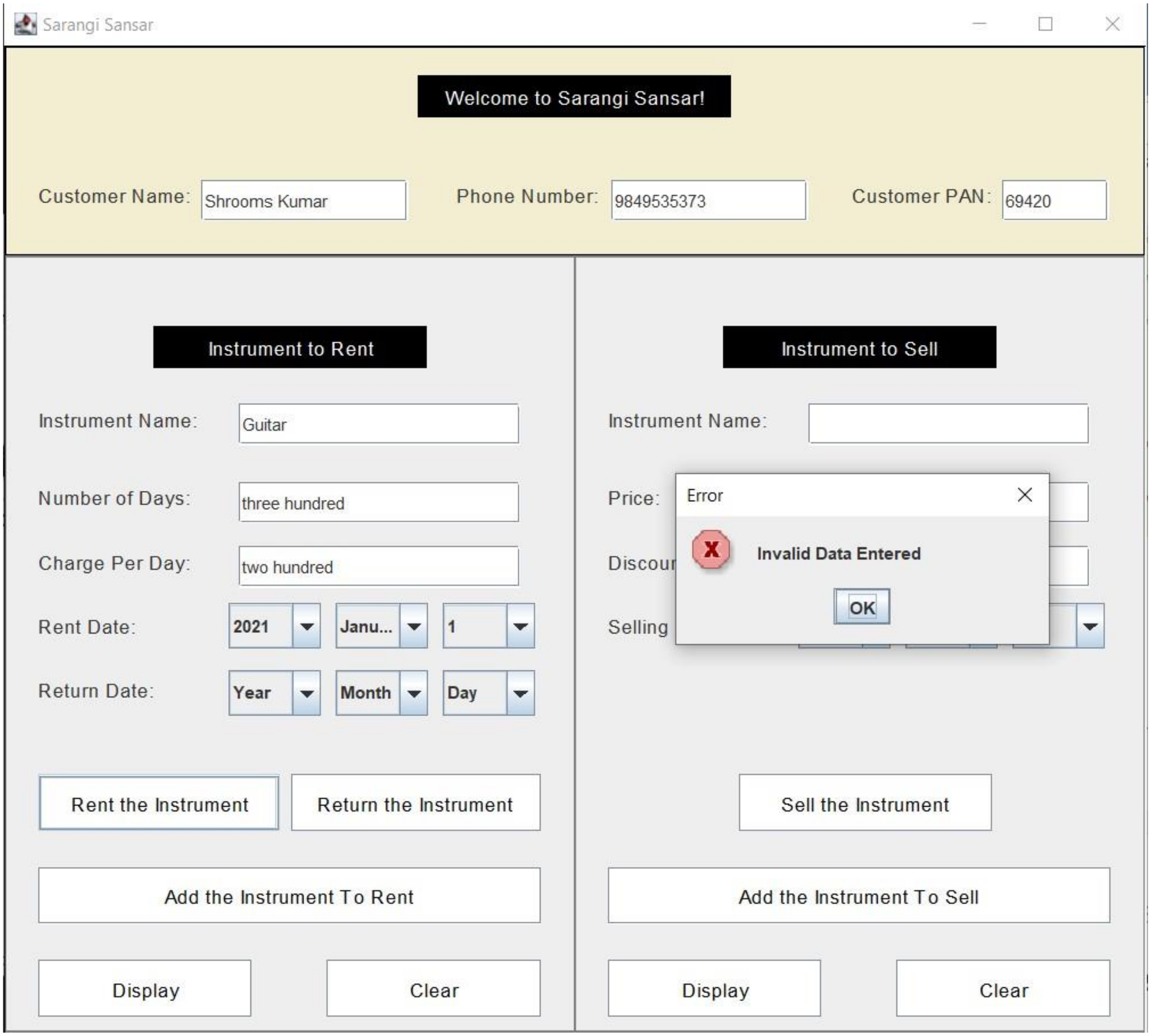
Please, Fill up all the Details

Figure 14: Testing of appropriate dialogue box when text fields are left unifiled

5.4 Test 4: Test that appropriate dialog boxes appear when invalid values are entered.

Test No	**4**
Objective	**Test that appropriate dialog boxes appear when invalid values are entered.**
Action	String values were entered in price and number of days to rent instrument.
Expected Result	Error message would pop up saying invalid input.
Actual Result	Pop-up message was displayed saying invalid input.
Conclusion	The test is successful.

Table 8: Test Table 4



Customer Name: Shrooms Kumar

Phone Number: 9849535373

Customer PAN: 69420

Instrument Name: Guitar

Number of Days: three hundred

Charge Per Day: two hundred

Rent Date: 2021 Janu... 1

Return Date: Year Month Day

Instrument Name:

Error

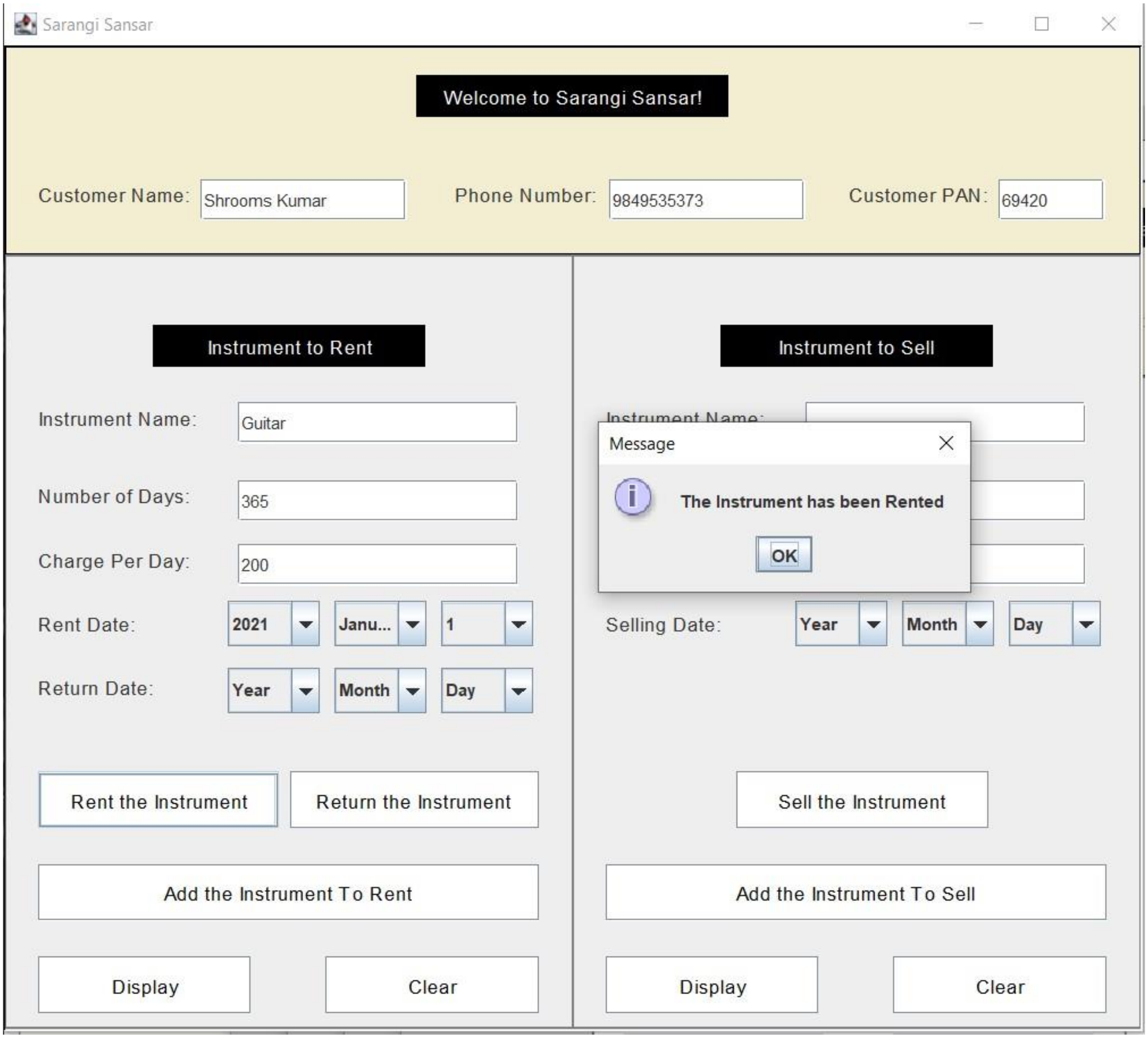
Invalid Data Entered

Figure 15: Screenshot of appropriate dialog box in case of an Invalid input

5.5 Test 5: Test that appropriate dialog boxes appear when unsuitable values are entered for the Instrument name.

Test No	**5**
Objective	**Test that appropriate dialog boxes appear when unsuitable values are entered for the Instrument name.**
Action	Unsuitable instrument name was entered to rent instrument.
Expected Result	Error message would pop up.
Actual Result	A pop-up message was displayed saying the instrument failed to be rented.
Conclusion	The test is successful.

Table 9: Test Table 5



Sarangi Sansar

Welcome to Sarangi Sansar!

Customer Name: Shrooms Kumar

Phone Number: 9849535373

Customer PAN: 69420

Instrument to Rent

Instrument Name: Guitar

Number of Days: 365

Charge Per Day: 200

Rent Date: 2021 Janu... 1

Return Date: Year Month Day

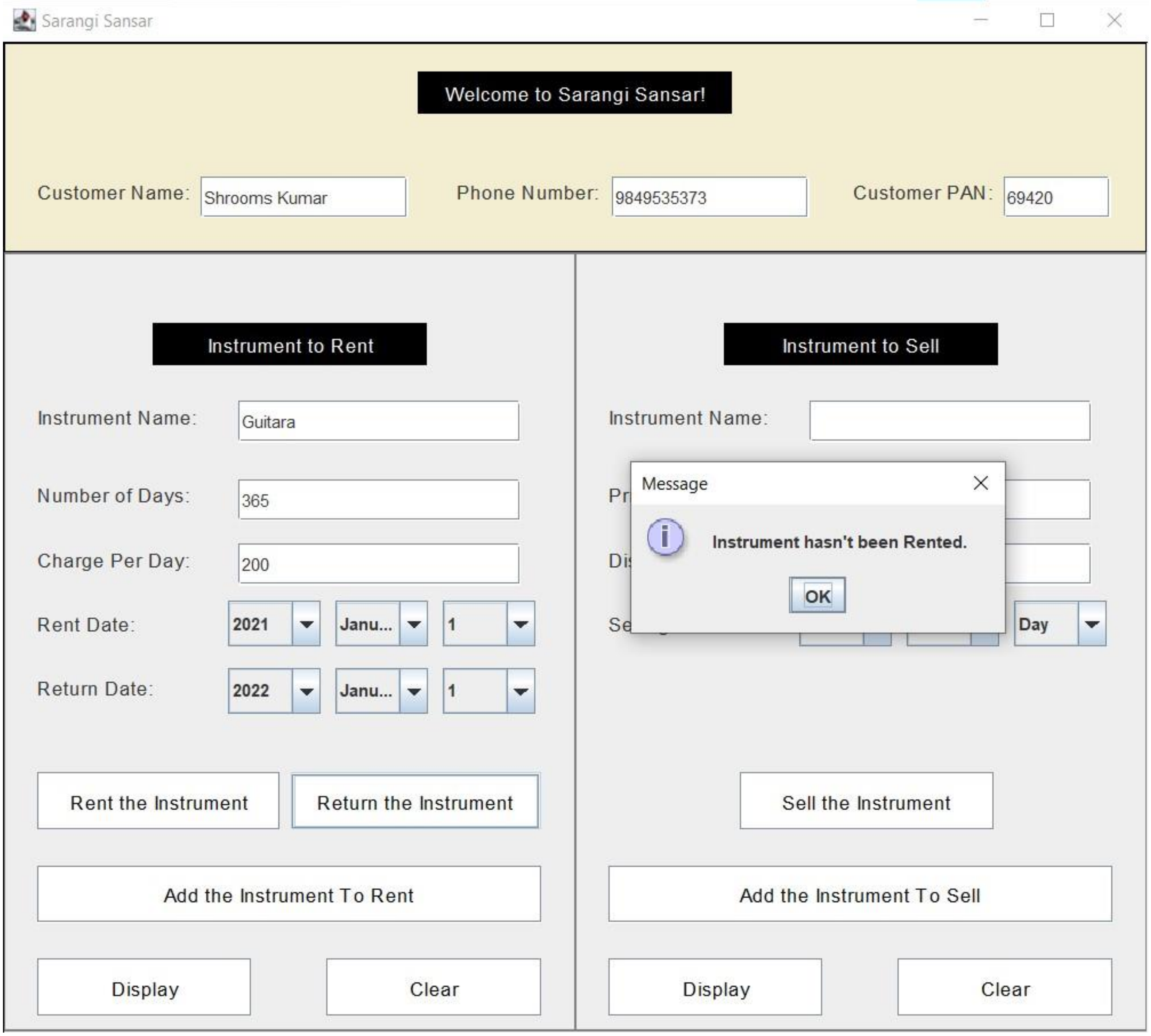
Instrument to Sell

Instrument Name:

Selling Date: Year Month Day

Message

The Instrument has been Rented



Sarangi Sansar

Welcome to Sarangi Sansar!

Customer Name: Shrooms Kumar

Phone Number: 9849535373

Customer PAN: 69420

Instrument to Rent

Instrument Name: Guitara

Number of Days: 365

Charge Per Day: 200

Rent Date: 2021 Janu... 1

Return Date: 2022 Janu... 1

Instrument to Sell

Instrument Name:

Message

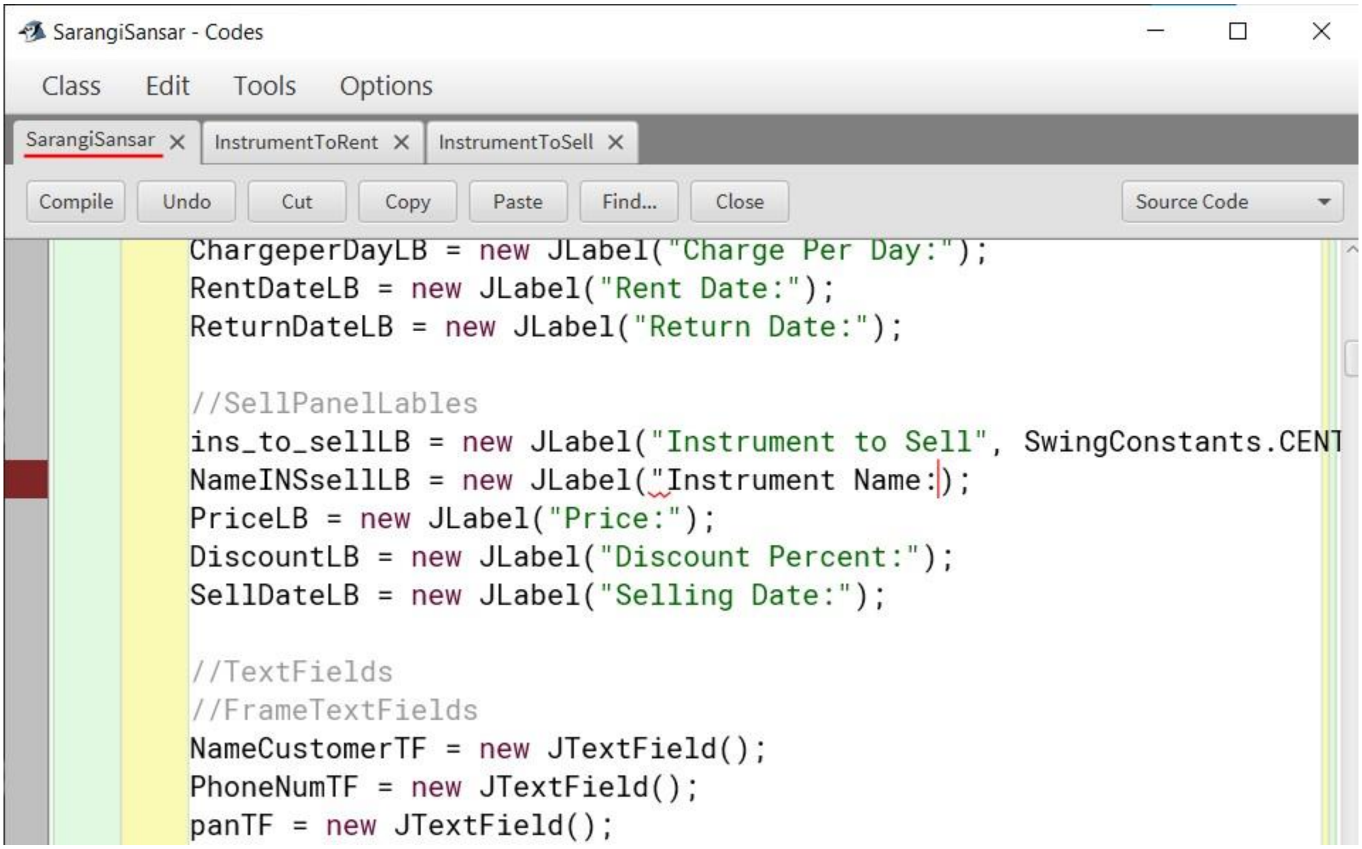
Instrument hasn't been Rented.

Figure 16: Screenshot of appropriate dialog box when unsuitable Instrument Name is entered

6. Error Detection and Correction:

6.1 Syntax Error

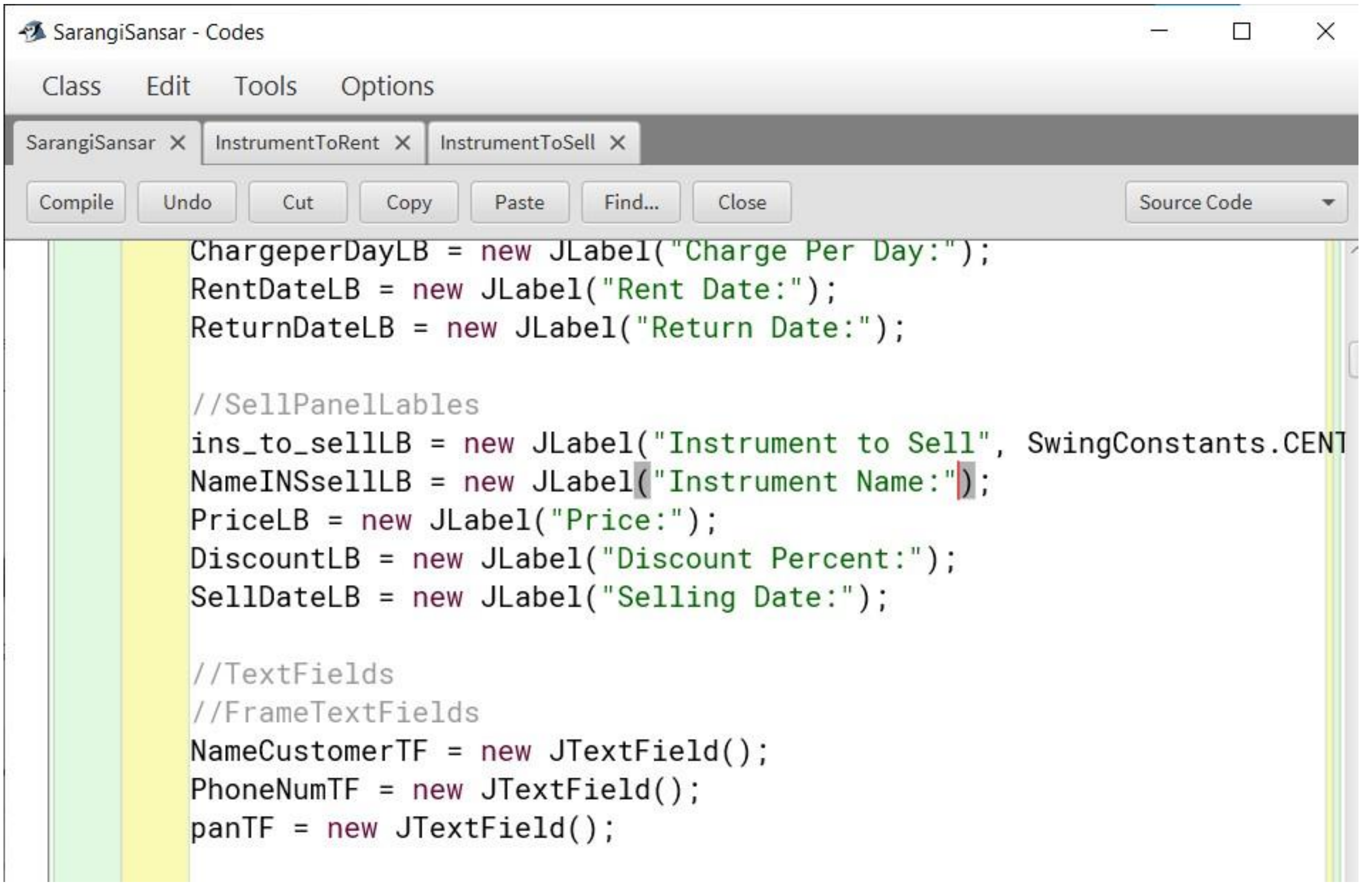
These types of errors are caused due to missing a semicolon at the phrases' end (;), missed braces, class not found, etc. During Compilation, the compiler in java finds the errors and displays the mistakes on the screen. Screenshots of Syntax errors I faced in my project are as follows:



```
ChargeperDayLB = new JLabel("Charge Per Day:");
RentDateLB = new JLabel("Rent Date:");
ReturnDateLB = new JLabel("Return Date:");

//SellPanelLables
ins_to_sellLB = new JLabel("Instrument to Sell", SwingConstants.CENT
NameINSsellLB = new JLabel("Instrument Name:);
PriceLB = new JLabel("Price:");
DiscountLB = new JLabel("Discount Percent:");
SellDateLB = new JLabel("Selling Date:");

//TextFields
//FrameTextFields
NameCustomerTF = new JTextField();
PhoneNumTF = new JTextField();
panTF = new JTextField();
```

Figure 17: Screenshot of Syntax Error

I had missed quotation marks at the end of the phrases and the program didn't compile. The red marks on the screen showed errors and I was able to solve the error. The screenshot after solving error is pasted below:



```
ChargeperDayLB = new JLabel("Charge Per Day:");
RentDateLB = new JLabel("Rent Date:");
ReturnDateLB = new JLabel("Return Date:");

//SellPanelLables
ins_to_sellLB = new JLabel("Instrument to Sell", SwingConstants.CENT
NameINSsellLB = new JLabel("Instrument Name:");
PriceLB = new JLabel("Price:");
DiscountLB = new JLabel("Discount Percent:");
SellDateLB = new JLabel("Selling Date:");

//TextFields
//FrameTextFields
NameCustomerTF = new JTextField();
PhoneNumTF = new JTextField();
panTF = new JTextField();
```

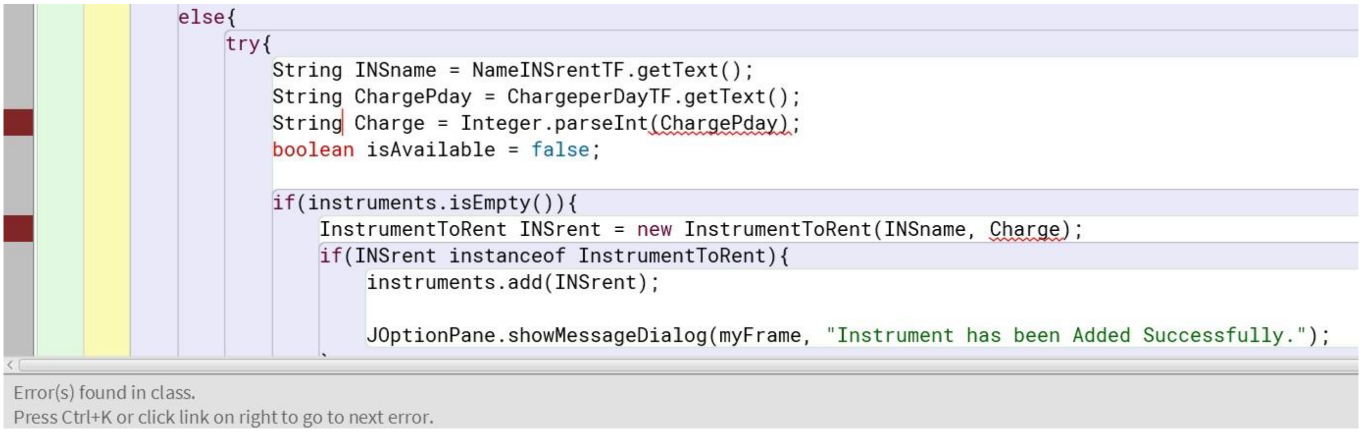
Figure 18: Screenshot of Correction of Syntax Error

The error had been fixed by adding a quotation mark (“Instrument Name”). Then, the program compiled without any errors.

6.2 Semantic Error

These types of errors arise when a statement is syntactically accurate but the function intended by the programmer isn’t accomplished. Usually, the compiler detects such errors themselves.

Screenshots of Semantic errors I faced in my project are as follows:

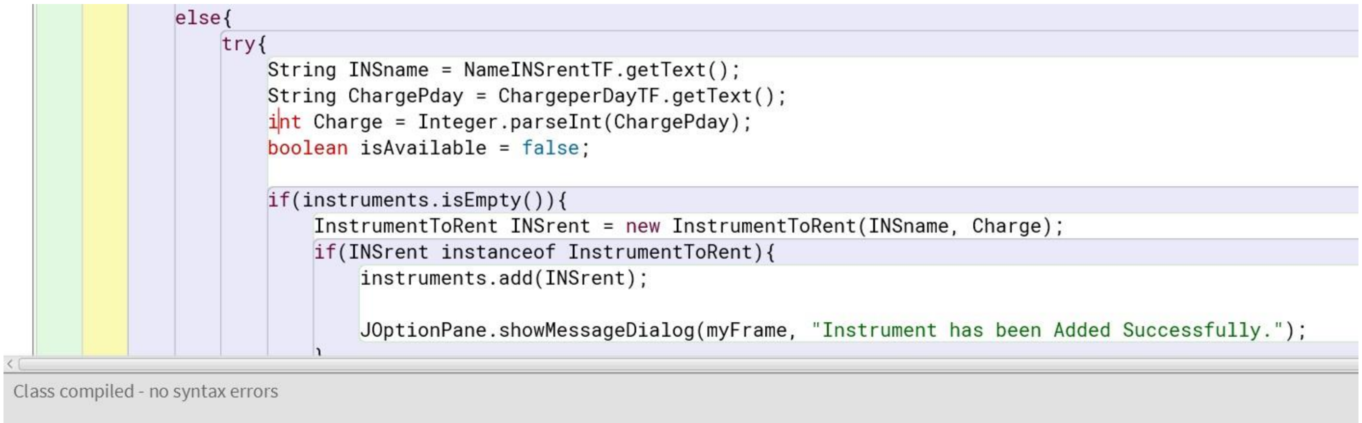


```
else{
    try{
        String INSname = NameINSrentTF.getText();
        String ChargePday = ChargeperDayTF.getText();
        String Charge = Integer.parseInt(ChargePday);
        boolean isAvailable = false;

        if(instruments.isEmpty()){
            InstrumentToRent INSrent = new InstrumentToRent(INSname, Charge);
            if(INSrent instanceof InstrumentToRent){
                instruments.add(INSrent);

                JOptionPane.showMessageDialog(myFrame, "Instrument has been Added Successfully.");
```

Error(s) found in class.
Press Ctrl+K or click link on right to go to next error.

Figure 19: Semantic Error

The error occurred because I used String in place of An Integer Data Type and the error occurred.



```
else{
    try{
        String INSname = NameINSrentTF.getText();
        String ChargePday = ChargeperDayTF.getText();
        int Charge = Integer.parseInt(ChargePday);
        boolean isAvailable = false;

        if(instruments.isEmpty()){
            InstrumentToRent INSrent = new InstrumentToRent(INSname, Charge);
            if(INSrent instanceof InstrumentToRent){
                instruments.add(INSrent);

                JOptionPane.showMessageDialog(myFrame, "Instrument has been Added Successfully.");
```

Class compiled - no syntax errors

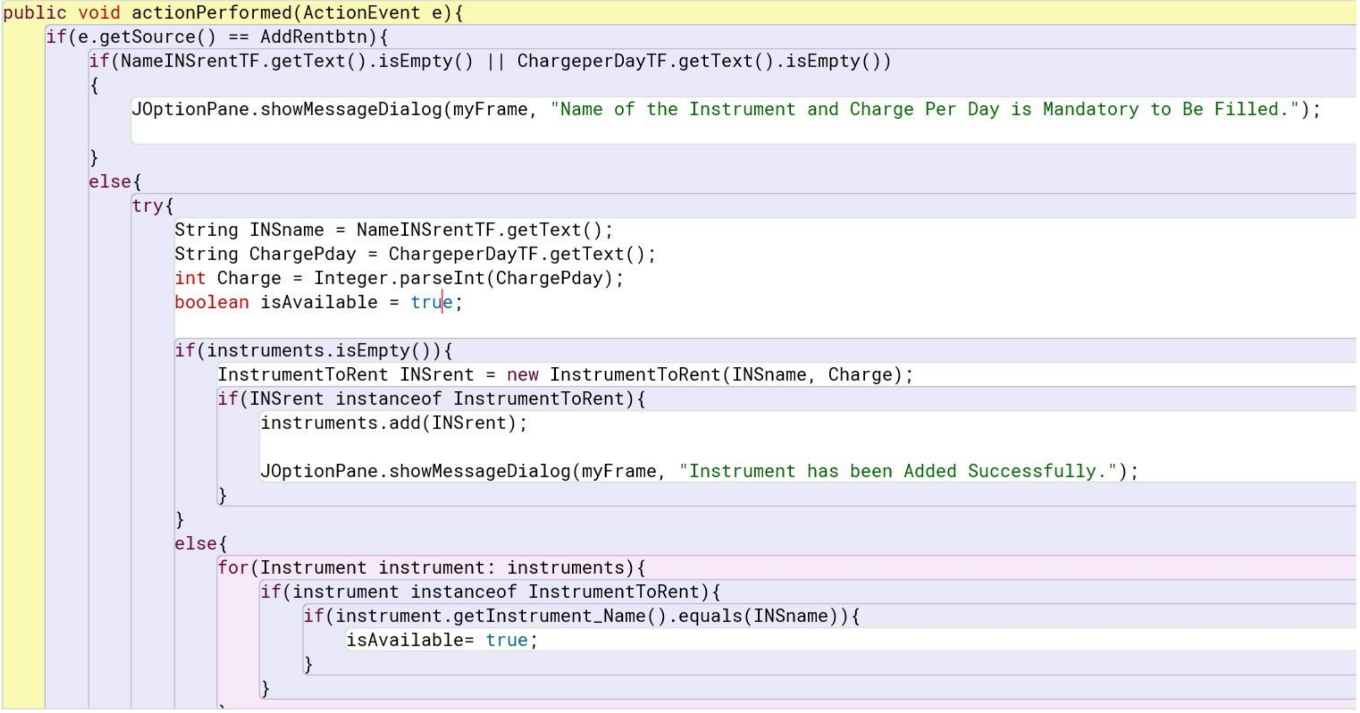
Figure 20 Semantic Error Solved

The error was solved by replacing String with int and the error was solved and program was compiled eventually.

6.3 Logical Error

These problems occur due to programming error. It usually remains undetected by a compiler or JVM. During coding, errors may occur when a coder uses the incorrect definition or concept.

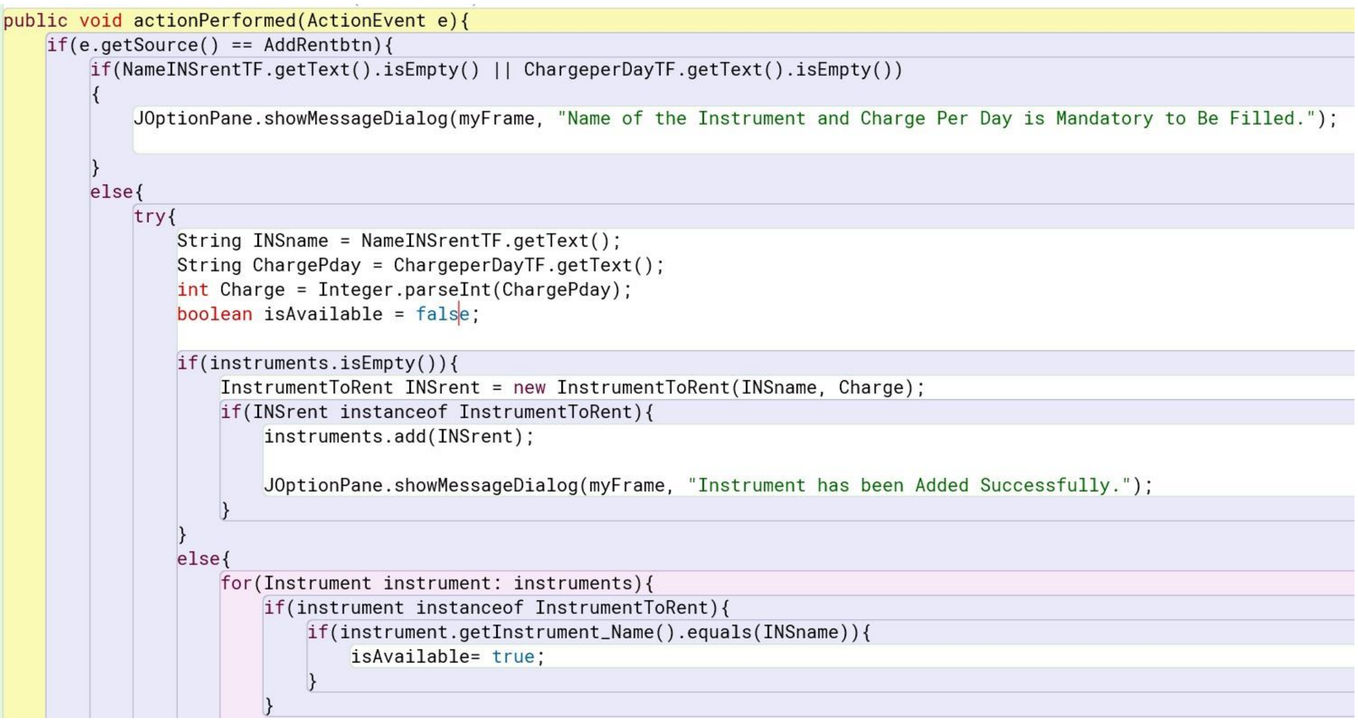
Screenshots of Logical errors I faced in my project are as follows:



```
public void actionPerformed(ActionEvent e){
    if(e.getSource() == AddRentbtn){
        if(NameINSrentTF.getText().isEmpty() || ChargeperDayTF.getText().isEmpty())
        {
            JOptionPane.showMessageDialog(myFrame, "Name of the Instrument and Charge Per Day is Mandatory to Be Filled.");

        }
        else{
            try{
                String INSname = NameINSrentTF.getText();
                String ChargePday = ChargeperDayTF.getText();
                int Charge = Integer.parseInt(ChargePday);
                boolean isAvailable = true;

                if(instruments.isEmpty()){
                    InstrumentToRent INSrent = new InstrumentToRent(INSname, Charge);
                    if(INSrent instanceof InstrumentToRent){
                        instruments.add(INSrent);

                        JOptionPane.showMessageDialog(myFrame, "Instrument has been Added Successfully.");
                    }
                }
                else{
                    for(Instrument instrument: instruments){
                        if(instrument instanceof InstrumentToRent){
                            if(instrument.getInstrument_Name().equals(INSname)){
                                isAvailable= true;
                            }
                        }
```

Figure 21: Screenshot of Logical error

I detected a logical error while working on my software, I kept true instead of false. Though, while compiling it didn’t show any error. The problem was solved by changing the Boolean isAvailable to false.



```
public void actionPerformed(ActionEvent e){
    if(e.getSource() == AddRentbtn){
        if(NameINSrentTF.getText().isEmpty() || ChargeperDayTF.getText().isEmpty())
        {
            JOptionPane.showMessageDialog(myFrame, "Name of the Instrument and Charge Per Day is Mandatory to Be Filled.");

        }
        else{
            try{
                String INSname = NameINSrentTF.getText();
                String ChargePday = ChargeperDayTF.getText();
                int Charge = Integer.parseInt(ChargePday);
                boolean isAvailable = false;

                if(instruments.isEmpty()){
                    InstrumentToRent INSrent = new InstrumentToRent(INSname, Charge);
                    if(INSrent instanceof InstrumentToRent){
                        instruments.add(INSrent);

                        JOptionPane.showMessageDialog(myFrame, "Instrument has been Added Successfully.");
                    }
                }
                else{
                    for(Instrument instrument: instruments){
                        if(instrument instanceof InstrumentToRent){
                            if(instrument.getInstrument_Name().equals(INSname)){
                                isAvailable= true;
                            }
                        }
```

Figure 22: Screenshot of Logical Error Correction

Conclusion

The overall goal of the coursework is to solve a real-world-problem scenario via using Object Oriented Programming (OOP) and create a Graphical User Interface (GUI) for users to interact with. My project enables users to solve several problems in a fast and efficient manner. Multiple data can be sent in a rapid manner after the development of the application which is written in Java.

During the course of the project, I faced many semantic and logical errors which were a headache to solve. Logical errors specially were hard to spot and the program would simply compile putting a smile on my face but at the time of running, due to the logical errors I would simply get a headache. Also, creating a class diagram was made easy due to draw.io but navigating through the software was also another headache. I learnt a lot about GUI and various techniques to play around within it to increase the efficiency of the functionality of the project.

The logical errors were hardest to solve. But going through the program several times, I was finally able to figure out the problems and solve the errors. Syntax errors were the easier ones to solve but were any in number. I was able to easily navigate through them and solve them.

Bibliography

Anon., 2021. *geeks for geeks.* [Online]
Available at: https://www.geeksforgeeks.org/introduction-of-bluej/#:~:text=BlueJ%20is%20a%20windows%20based,or%20more%20before%20installing%20BlueJ.

Anon., 2021. *Oracle.* [Online]
Available at: https://www.oracle.com/database/what-is-database/
[Accessed 09 01 2022].

Anon., 2022. *Stack Overflow.* [Online]
Available at: https://stackoverflow.com/questions/73245619/how-to-update-database-from-jtable

Anon., n.d. *blueJ.* [Online]
Available at: https://www.bluej.org/
[Accessed 2022].

Anon., n.d. *paraphraser.* [Online]
Available at: https://www.paraphraser.io/
[Accessed 2022].

Anon., n.d. *TEC.* [Online]
Available at: https://www3.technologyevaluation.com/solutions/53717/drawio
[Accessed 2 august 2022].

Bates, K. S. &. B., 2005. *Head First Java.* 2nd ed. s.l.:Shroff/O'Reilly.

Bing, J., 2022. *SourceCodeEster.* [Online]
Available at: https://www.sourcecodester.com/tags/java-gui

Bloch, J., 2018. *Effective Java.* 3rd edition ed. s.l.:Addison Wesley.

Goetz, B., 2006. *Java Concurrency in Practice.* 1st edition ed. s.l.:Addison-Wesley Professional.

guercio, k., april 5, 2022. *webopedia.* [Online]
Available at: https://www.webopedia.com/definitions/word-processor/
[Accessed 02 August 2022].

Horstmann, C. S., 2020. In: *Core Java Volume 1.* s.l.:Prentice Hall.

Pedamkar, P., 2021. *educba.* [Online]
Available at: https://www.educba.com/java-gui-framework/

Appendix

Class: Instrument

```
public class
Instrument
{
   private static int Instrument_ID;
private String Instrument_Name;
private String Customer_Name;
private String
Customer_Mobile_Number;     private int
Pan_Number;

   public Instrument(String Instrument_Name)
   {
      this.Instrument_Name =
Instrument_Name;        Instrument_ID ++;
this.Customer_Name ="";
this.Customer_Mobile_Number ="";
this.Pan_Number = 0;


   }
      public int getInstrument_ID()
   {
      return Instrument_ID;
   }
   public String getInstrument_Name()
   {
      return this.Instrument_Name;
   }
   public String getCustomer_Name()
   {
      return this.Customer_Name;
   }
   public String getCustomer_Mobile_Number()
   {
      return this.Customer_Mobile_Number;
   }
   public int getPan_Number()
   {
      return this.Pan_Number;
   }

   public void setInstrument_ID(int Instrument_ID)
   {
      Instrument.Instrument_ID = Instrument_ID;
   }
   public void setInstrument_Name(String Instrument_Name)
   {
      this.Instrument_Name = Instrument_Name;
   }
   public void setCustomer_Name(String Customer_Name)
   {
      this.Customer_Name = Customer_Name;
   }
   public void setCustomer_Mobile_Number(String Customer_Mobile_Number)
   {
      this.Customer_Mobile_Number =Customer_Mobile_Number;
   }
   public void setPan_Number(int Pan_Number)
   {
      this.Pan_Number =Pan_Number;
   }
   public void display()
   {
      System.out.println("Instrument ID: " + Instrument_ID);
      System.out.println("Instrument Name: "+ this.Instrument_Name);

      if(Customer_Name !=""&& Customer_Mobile_Number!=""&& Pan_Number!=0)
      {
          System.out.println("Customer_Name:" + this.Customer_Name);
          System.out.println("Customer_Mobile_Number:"
+this.Customer_Mobile_Number);
          System.out.println("Pan_Number:" +this.Pan_Number);
      }
   }
}
```

Class: InstrumentToRent

```
public class InstrumentToRent extends
Instrument
{
   /*creating variables*/     private int ChargePerDay;     private
String DateOfRent;     private String DateOfReturn;     private int
NoofDays;     private boolean IsRented      public
InstrumentToRent(String Instrument_Name,int ChargePerDay)
   {
      super(Instrument_Name);
this.ChargePerDay =ChargePerDay;
this.DateOfRent ="";
this.DateOfReturn = "";
this.NoofDays =0;         this.IsRented
=false;
   }

   public int getChargePerDay()
   {
      return this.ChargePerDay;
   }
   public String getDateOfRent()
   {
      return this.DateOfRent;
   }
   public String getDateOfReturn()
   {
      return this.DateOfReturn;
   }
   public int getNoofDays()
   {
      return this.NoofDays;
   }
   public boolean getIsRented()
   {
      return this.IsRented;
   }

   public void setChargePerDay(int ChargePerDay)
   {
      this.ChargePerDay = ChargePerDay;
   }
   public void setDateOfRent(String DateOfRent)
   {
      this.DateOfRent = DateOfRent;
   }
   public void setDateOfReturn(String DateOfReturn)
   {
      this.DateOfReturn = DateOfReturn;
   }
   public void setNoofDays(int noOfDays)
   {
      this.NoofDays = NoofDays;
   }
   public void setIsRented(boolean IsRented)
   {
      this.IsRented = IsRented;
   }

   public void Rent(String Name, String Phone, int Pan_Number, String DateOfRent,
String DateOfReturn, int NoofDays)
   {
       if(IsRented ==true)
       {
          System.out.println("No instrument is available.");
       }
else
       {
          super.setCustomer_Name(Name);
super.setCustomer_Mobile_Number(Phone);
super.setPan_Number(Pan_Number);
this.DateOfRent = DateOfRent;
this.DateOfReturn = DateOfReturn;
this.NoofDays = NoofDays;            IsRented
=true;            int TotalCharge;
          TotalCharge = NoofDays * this.ChargePerDay;
          System.out.println("Name of the Customer is " + Name);
          System.out.println("Mobile Number of Customer is " +Phone);
          System.out.println("Pan_Number is " + Pan_Number);
          System.out.println("DateOfRent is " + DateOfRent);
          System.out.println("DateOfReturn is " + DateOfReturn);
          System.out.println("NoofDays is " + NoofDays);
          System.out.println("The Total Charge is " + TotalCharge);


       }
    }
    public void returnInstrument()
   {
      if(IsRented == false)
      {
         System.out.println("The Instrument Has Not Been Rented");
      }
else
```

```
       {
           this.setCustomer_Name("");
this.setCustomer_Mobile_Number("");
this.setDateOfReturn("");
this.setDateOfRent("");
this.setNoofDays(0);
this.setPan_Number(0);
this.setIsRented(false);
       }
   }
    public void Return()
    {
        if(IsRented ==false)
        {
            System.out.print("The instrument has not been rented");
        }
else
        {
            super.display();
this.setCustomer_Name("");
this.setCustomer_Mobile_Number("");
this.setDateOfReturn("");
this.setDateOfRent("");
this.setNoofDays(0);
this.setPan_Number(0);
this.setIsRented(false);

        }
    }
    public void display()
    {
        super.display();
if(IsRented ==true)
        {
            System.out.println("customer's name: "+getCustomer_Name() );
            System.out.println("DateOfRent: " + DateOfRent);
            System.out.println("DateOfReturn: " + DateOfReturn);
            System.out.println("Pan Number: "+getPan_Number());
            System.out.println("Phone Number: "+ getCustomer_Mobile_Number());
        }
    }
}
```

Class: InstrumentToSell

```
public class InstrumentToSell extends
Instrument
{
    private float Price;
private String SellDate;
private float
DiscountPercent;     private
boolean isSold;


    public InstrumentToSell(float Price, String Name)
    {
        super(Name);
this.Price = Price;
this.SellDate ="";
this.DiscountPercent =0.0f;
this.isSold =false;
    }

    public float getPrice()
    {        return
this.Price;
    }
    public String getSellDate()
    {
        return this.SellDate;
    }
    public float getDiscountPercent()
    {
        return this.DiscountPercent;
```

```
   }
   public boolean getisSold()
   {
      return this.isSold;
   }

   public void setPrice(float Price)
   {   if(isSold ==false)
      {
this.Price = Price;
      }
else
      {
         System.out.println("The Instrument Has Been Already Sold");
      }
   }
   public void setSellDate(String SellDate)
   {
      this.SellDate = SellDate;
   }
   public void setDiscountPercent(float DiscountPercent)
   {
      this.DiscountPercent = DiscountPercent;
   }
   public void setisSold(boolean isSold)
   {
      this.isSold = isSold;
   }
   public void sell(String Customer_Name, String Phone, int Pan_Number, String
SellDate, float DiscountPercent)
   {
        if(isSold==true)
        {
            System.out.println("The Instrument Has Been Sold.");
            System.out.println("Customer Name: " +Customer_Name);
            System.out.println("Customer Mobile Number: " + Phone);
            System.out.println("Customer Pan Number: "+ Pan_Number);
        }
else
        {
            this.SellDate =SellDate;
this.DiscountPercent=DiscountPercent;
isSold =true;
            super.setCustomer_Name(Customer_Name);
super.setCustomer_Mobile_Number(Phone);
super.setPan_Number(Pan_Number);
            Price =Price -((DiscountPercent/100) * Price);
        }
    }
    public void display()
    {
        super.display();
        System.out.println("The Price of The Instrument is: " +Price);
if(isSold == true)
        {
            System.out.println("Name of Customer: " +getCustomer_Name());
System.out.println("Phone Number of the Customer: "+ getCustomer_Mobile_Number());
            System.out.println("Pan Number of the Customer: "+ getPan_Number());
System.out.println("Sold Date: "+SellDate);
        }
    }
}
```

Class: SarangiSansar

```
/**
 * Write a description of class SarangiSansar
here.
 *
 * @author (Sharams Kunwar)
 * @version (a version number or a date)
 */
import javax.swing.*;
import java.util.*;
import java.awt.*;
import java.awt.event.*;
import java.awt.color.*;

public class SarangiSansar implements
ActionListener
{
   private JFrame myFrame;
   private JPanel aPanel, rPanel, sPanel;
   private JLabel TitleLB, NameCustomerLB,
PhoneNumLB, panLB, ins_to_rentLB,
   ins_to_sellLB,NameINSrentLB,
NumofDaysLB, ChargeperDayLB,
RentDateLB,
   ReturnDateLB, NameINSsellLB, PriceLB,
DiscountLB, SellDateLB;

   private JTextField NameCustomerTF,
PhoneNumTF, panTF, NameINSrentTF,
    NameINSsellTF, NumofDaysTF, PriceTF,
ChargeperDayTF, DiscountTF;

    private JComboBox RentDateCBy,
ReturnDateCBy, SellDateCBy;
    private JComboBox RentDateCBm,
ReturnDateCBm, SellDateCBm;
    private JComboBox RentDateCBd,
ReturnDateCBd, SellDateCBd;

    private JButton RentINSbtn, ReturnINSbtn,
AddRentbtn, DisplayRentbtn, ClearRentbtn,
    SellINSbtn, AddSellbtn, DisplaySellbtn,
ClearSellbtn;

    ArrayList<Instrument> instruments = new
ArrayList<Instrument>();
    public SarangiSansar(){
        //Main Frame
        myFrame = new JFrame("Sarangi
Sansar");

        //Creating
        //Panels
        aPanel = new JPanel();
        rPanel = new JPanel();
        sPanel = new JPanel();

        //labels

        //FrameLabels
        TitleLB = new JLabel("Welcome to
Sarangi Sansar!", SwingConstants.CENTER);
        NameCustomerLB = new
JLabel("Customer Name:");
        PhoneNumLB = new JLabel("Phone
Number:");
        panLB = new JLabel("Customer PAN:");

        //RentPanelLabels
        ins_to_rentLB = new JLabel("Instrument
to Rent", SwingConstants.CENTER);
        NameINSrentLB = new
JLabel("Instrument Name:");
        NumofDaysLB = new JLabel("Number of
Days:");
        ChargeperDayLB = new JLabel("Charge
Per Day:");
        RentDateLB = new JLabel("Rent Date:");
        ReturnDateLB = new JLabel("Return
Date:");

        //SellPanelLables
        ins_to_sellLB = new JLabel("Instrument
to Sell", SwingConstants.CENTER);
        NameINSsellLB = new
JLabel("Instrument Name:");
        PriceLB = new JLabel("Price:");
        DiscountLB = new JLabel("Discount
Percent:");
        SellDateLB = new JLabel("Selling
Date:");
```

```
        //TextFields
        //FrameTextFields
        NameCustomerTF = new JTextField();
        PhoneNumTF = new JTextField();
        panTF = new JTextField();

        //RentPanel TextFields
        NameINSsellTF = new JTextField();
        NumofDaysTF = new JTextField();
        ChargeperDayTF = new JTextField();

        //SellPanel TesxtFields
        NameINSrentTF = new JTextField();
        PriceTF = new JTextField();
        DiscountTF = new JTextField();

        //ComboBox
        //array
        String
y[]={"Year","2017","2018","2019","2020","202
1","2022"};
        String
m[]={"Month","January","February","March","
April","May","June", "July"
            , "August", "September", "October",
"November", "December"};
        String
d[]={"Day","1","2","3","4","5","6","7","8","9","10
","11","12","13",
"14","15","16","17","18","19","20","21","22","2
3","24",
            "25","26","27","28","29","30","31"};

        //Rent ComboBox
        RentDateCBy = new JComboBox(y);
        RentDateCBm = new JComboBox(m);
        RentDateCBd = new JComboBox(d);

        //Return ComboBox
        ReturnDateCBy = new JComboBox(y);
        ReturnDateCBm = new JComboBox(m);
        ReturnDateCBd = new JComboBox(d);

        //Sell ComboBox
        SellDateCBy = new JComboBox(y);
        SellDateCBm = new JComboBox(m);
        SellDateCBd = new JComboBox(d);

        //Buttons

        //RentPanel Buttons
        RentINSbtn = new JButton("Rent the
Instrument");
        ReturnINSbtn = new JButton("Return the
Instrument");
        AddRentbtn = new JButton("Add the
Instrument To Rent");
        DisplayRentbtn = new
JButton("Display");
```

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

        //SellPanel Buttons
        SellINSbtn = new JButton("Sell the
Instrument");
        AddSellbtn = new JButton("Add the
Instrument To Sell");
        DisplaySellbtn = new JButton("Display");
        ClearSellbtn = new JButton("Clear");

        //Setting Bounds

        //main frame

myFrame.getRootPane().setBorder(BorderFa
ctory.createLineBorder(Color.black));

        //aPanel
        aPanel.setBounds(0, 0, 800, 147);
        aPanel.setBackground(new
Color(243,237,210));

aPanel.setBorder(BorderFactory.createLineB
order(Color.black));

        //1.Title
        TitleLB.setBounds(290, 20, 220, 30);
        TitleLB.setForeground(Color.WHITE);
        TitleLB.setBackground(Color.BLACK);
        TitleLB.setFont(new Font("Helvetica",
Font.PLAIN, 14));
```

```
        TitleLB.setOpaque(true);

        //2.Customer Name
        //Label

NameCustomerLB.setBounds(23,94,110,20);
        NameCustomerLB.setFont(new
Font("Helvetica", Font.PLAIN, 14));

        //TextField
        NameCustomerTF.setBounds(138, 94,
145, 29);

        //3.Phone Number
        //Label
        PhoneNumLB.setBounds(317, 94, 110,
20);
        PhoneNumLB.setFont(new
Font("Helvetica", Font.PLAIN, 14));

        //TextField
        PhoneNumTF.setBounds(426, 94,
138,29);

        //4.PAN no
        //label
        panLB.setBounds(594, 94, 130, 20);
        panLB.setFont(new Font("Helvetica",
Font.PLAIN, 14));

        //textfield
        panTF.setBounds(700, 94, 75, 29);

        //end of aPanel

        //rPanel
        rPanel.setBounds(0,147,400,545);
        rPanel.setBackground(new
Color(238,238,238));

rPanel.setBorder(BorderFactory.createLineBo
rder(Color.gray));

        //1.Title
        ins_to_rentLB.setBounds(104, 49, 192,
30);

ins_to_rentLB.setForeground(Color.WHITE);

ins_to_rentLB.setBackground(Color.BLACK);
        ins_to_rentLB.setFont(new
Font("Helvetica", Font.PLAIN, 14));
        ins_to_rentLB.setOpaque(true);

        //2.Labels
        //ins Name
        NameINSrentLB.setBounds(23, 104,
115, 20);
        NameINSrentLB.setFont(new
Font("Helvetica", Font.PLAIN, 14));

        //numofdays
        NumofDaysLB.setBounds(23, 159, 110,
20);
        NumofDaysLB.setFont(new
Font("Helvetica", Font.PLAIN, 14));

        //chargeperday
        ChargeperDayLB.setBounds(23, 204,
110, 20);
        ChargeperDayLB.setFont(new
Font("Helvetica", Font.PLAIN, 14));

        //rentdate
        RentDateLB.setBounds(23, 249, 110,
20);
        RentDateLB.setFont(new
Font("Helvetica", Font.PLAIN, 14));

        //returndate
        ReturnDateLB.setBounds(23, 294, 110,
20);
        ReturnDateLB.setFont(new
Font("Helvetica", Font.PLAIN, 14));

        //3.textfields
        //ins name
        NameINSrentTF.setBounds(164, 104,
198, 29);

        //numofdays
        NumofDaysTF.setBounds(164, 159,
198, 29);
```

```
        //chargeperday
        ChargeperDayTF.setBounds(164, 204,
198, 29);

        //4. buttons
        //rentinstrumentbtn
        RentINSbtn.setBounds(23,364,170,40);

RentINSbtn.setForeground(Color.BLACK);

RentINSbtn.setBackground(Color.WHITE);
        RentINSbtn.setFont(new
Font("Helvetica", Font.PLAIN, 14));
        //RentINSbtn.setBorder(new
RoundBtn(15));
        RentINSbtn.setOpaque(true);

        //returninsbtn

ReturnINSbtn.setBounds(201,364,175,40);

ReturnINSbtn.setForeground(Color.BLACK);

ReturnINSbtn.setBackground(Color.WHITE);
        ReturnINSbtn.setFont(new
Font("Helvetica", Font.PLAIN, 14));
        ReturnINSbtn.setOpaque(true);

        //addtorentbtn
        AddRentbtn.setBounds(23,429,353,40);
AddRentbtn.setForeground(Color.BLACK);

AddRentbtn.setBackground(Color.WHITE);
        AddRentbtn.setFont(new
Font("Helvetica", Font.PLAIN, 14));
        AddRentbtn.setOpaque(true);

        //displaybtn

DisplayRentbtn.setBounds(23,494,150,40);

DisplayRentbtn.setForeground(Color.BLACK)
;

DisplayRentbtn.setBackground(Color.WHITE)
;
        DisplayRentbtn.setFont(new
Font("Helvetica", Font.PLAIN, 14));
        DisplayRentbtn.setOpaque(true);

        //clearbtn

ClearRentbtn.setBounds(226,494,150,40);

ClearRentbtn.setForeground(Color.BLACK);

ClearRentbtn.setBackground(Color.WHITE);
        ClearRentbtn.setFont(new
Font("Helvetica", Font.PLAIN, 14));
        ClearRentbtn.setOpaque(true);
        //5. Combobox
        //rentdate
        RentDateCBy.setBounds(157, 244, 65,
32);
        RentDateCBm.setBounds(232, 244, 65,
32);
        RentDateCBd.setBounds(307, 244, 65,
32);

        //returndate
        ReturnDateCBy.setBounds(157, 291,
65, 32);
        ReturnDateCBm.setBounds(232, 291,
65, 32);
        ReturnDateCBd.setBounds(307, 291,
65, 32);

        //array

        //end of rPanel
        //sPanel
        sPanel.setBounds(400,147,400,545);
        sPanel.setBackground(new
Color(238,238,238));

sPanel.setBorder(BorderFactory.createLineB
order(Color.gray));

        //1.Title
        ins_to_sellLB.setBounds(104, 49, 192,
30);

ins_to_sellLB.setForeground(Color.WHITE);

ins_to_sellLB.setBackground(Color.BLACK);
        ins_to_sellLB.setFont(new
Font("Helvetica", Font.PLAIN, 14));
        ins_to_sellLB.setOpaque(true);

        //2.labels
        //insname
        NameINSsellLB.setBounds(23, 104,
115, 20);
        NameINSsellLB.setFont(new
Font("Helvetica", Font.PLAIN, 14));

        //price
        PriceLB.setBounds(23, 159, 110, 20);
        PriceLB.setFont(new Font("Helvetica",
Font.PLAIN, 14));

        //discount
        DiscountLB.setBounds(23, 204, 115,
20);
        DiscountLB.setFont(new
Font("Helvetica", Font.PLAIN, 14));

        //sellingdate
        SellDateLB.setBounds(23, 249, 110,
20);
```

```
        SellDateLB.setFont(new
Font("Helvetica", Font.PLAIN, 14));

        //3.textfields
        //insname
        NameINSsellTF.setBounds(164, 104,
198, 29);

        //numofdays
        PriceTF.setBounds(164, 159, 198, 29);

        //chargeperday
        DiscountTF.setBounds(164, 204, 198,
29);

        //4.buttons
        //sellbtn
        SellINSbtn.setBounds(115,364,178,40);

SellINSbtn.setForeground(Color.BLACK);

SellINSbtn.setBackground(Color.WHITE);
        SellINSbtn.setFont(new
Font("Helvetica", Font.PLAIN, 14));
        SellINSbtn.setOpaque(true);

        //addtosellbtn
        AddSellbtn.setBounds(23,429,353,40);

AddSellbtn.setForeground(Color.BLACK);
AddSellbtn.setBackground(Color.WHITE);
        AddSellbtn.setFont(new
Font("Helvetica", Font.PLAIN, 14));
        AddSellbtn.setOpaque(true);

        //displaybtn

DisplaySellbtn.setBounds(23,494,150,40);

DisplaySellbtn.setForeground(Color.BLACK);

DisplaySellbtn.setBackground(Color.WHITE);
        DisplaySellbtn.setFont(new
Font("Helvetica", Font.PLAIN, 14));
        DisplaySellbtn.setOpaque(true);

        //clearbtn

ClearSellbtn.setBounds(226,494,150,40);

ClearSellbtn.setForeground(Color.BLACK);

ClearSellbtn.setBackground(Color.WHITE);
        ClearSellbtn.setFont(new
Font("Helvetica", Font.PLAIN, 14));
        ClearSellbtn.setOpaque(true);

        //combobox
        SellDateCBy.setBounds(157, 244, 65,
32);
```

```
        SellDateCBm.setBounds(232, 244, 65,
32);
        SellDateCBd.setBounds(307, 244, 65,
32);




        //Adding to Frame

        //Panels
        myFrame.add(aPanel);
        myFrame.add(rPanel);
        myFrame.add(sPanel);

        //components of aPanel
        //1. Labels
        aPanel.add(TitleLB);
        aPanel.add(NameCustomerLB);
        aPanel.add(PhoneNumLB);
        aPanel.add(panLB);

        //2.TextFields
        aPanel.add(NameCustomerTF);
        aPanel.add(PhoneNumTF);
        aPanel.add(panTF);

        //components of Rent Panel
        //1.Labels
        rPanel.add(ins_to_rentLB);
        rPanel.add(NameINSrentLB);
        rPanel.add(NumofDaysLB);
        rPanel.add(ChargeperDayLB);
        rPanel.add(RentDateLB);
        rPanel.add(ReturnDateLB);

        //2.TextFields
        rPanel.add(NameINSrentTF);
        rPanel.add(NumofDaysTF);
        rPanel.add(ChargeperDayTF);

        //3.ComboBox
        rPanel.add(RentDateCBy);
        rPanel.add(RentDateCBm);
        rPanel.add(RentDateCBd);
        rPanel.add(ReturnDateCBy);
        rPanel.add(ReturnDateCBm);
        rPanel.add(ReturnDateCBd);

        //4.Buttons
        rPanel.add(RentINSbtn);
        rPanel.add(ReturnINSbtn);
        rPanel.add(AddRentbtn);
        rPanel.add(DisplayRentbtn);
        rPanel.add(ClearRentbtn);

        //components of sell panel
        //1.Labels
        sPanel.add(ins_to_sellLB);
        sPanel.add(NameINSsellLB);
        sPanel.add(PriceLB);
        sPanel.add(DiscountLB);
        sPanel.add(SellDateLB);

        //2.Textfields
        sPanel.add(NameINSsellTF);
        sPanel.add(PriceTF);
        sPanel.add(DiscountTF);

        //3.ComboBox
        sPanel.add(SellDateCBy);
        sPanel.add(SellDateCBm);
        sPanel.add(SellDateCBd);

        //4.Buttons
        sPanel.add(SellINSbtn);
        sPanel.add(AddSellbtn);
        sPanel.add(DisplaySellbtn);
        sPanel.add(ClearSellbtn);

        //ActionListener
        SellINSbtn.addActionListener(this);
        AddSellbtn.addActionListener(this);
        DisplaySellbtn.addActionListener(this);
        ClearSellbtn.addActionListener(this);
        RentINSbtn.addActionListener(this);
        ReturnINSbtn.addActionListener(this);
        AddRentbtn.addActionListener(this);
        DisplayRentbtn.addActionListener(this);
        ClearRentbtn.addActionListener(this);

        //setting
        myFrame.setSize(817,732);
```

```
        myFrame.setLayout(null);
        aPanel.setLayout(null);
        rPanel.setLayout(null);
        sPanel.setLayout(null);


myFrame.setDefaultCloseOperation(JFrame.
EXIT_ON_CLOSE);
        myFrame.setVisible(true);

    }

    public static void main(String[]args)
    {
        SarangiSansar obj = new
SarangiSansar();

    }

    //Add Instrument for Rent
Button(RentINSbtn)
    public void actionPerformed(ActionEvent
e){
        if(e.getSource() == AddRentbtn){
            if(NameINSrentTF.getText().isEmpty()
|| ChargeperDayTF.getText().isEmpty())
            {

JOptionPane.showMessageDialog(myFrame,
"Name of the Instrument and Charge Per Day
is Mandatory to Be Filled.");
```

```
            }
            else{
                try{
                    String INSname = 
NameINSrentTF.getText();
                    String ChargePday = 
ChargeperDayTF.getText();
                    int Charge = 
Integer.parseInt(ChargePday);
                    boolean isAvailable = false;

                    if(instruments.isEmpty()){
                        InstrumentToRent INSrent = 
new InstrumentToRent(INSname, Charge);
                        if(INSrent instanceof 
InstrumentToRent){
                            instruments.add(INSrent);


JOptionPane.showMessageDialog(myFrame, 
"Instrument has been Added Successfully.");
                        }
                    }
                    else{
                        for(Instrument instrument: 
instruments){
                            if(instrument instanceof 
InstrumentToRent){
```

```
if(instrument.getInstrument_Name().equals(IN
Sname)){
                                    isAvailable= true;
                                }
                            }
                        }
                        if(isAvailable == true){

JOptionPane.showMessageDialog(myFrame,
"Instrument is Already Available.");
                        }
                        else{
                            InstrumentToRent INSrent =
new InstrumentToRent(INSname, Charge);
                            if(INSrent instanceof
InstrumentToRent){
                                instruments.add(INSrent);



JOptionPane.showMessageDialog(myFrame,
"Instrument has been Added Successfully.");
                            }
                        }
                    }
                }
                catch(NumberFormatException
except){

JOptionPane.showMessageDialog(rPanel,
"Invalid Data Entered","Error",
JOptionPane.ERROR_MESSAGE);

            }
         }
      }

      //Add Instrument for Sell Button
      if(e.getSource() == AddSellbtn){
         if(NameINSsellTF.getText().isEmpty()
|| PriceTF.getText().isEmpty())
         {

JOptionPane.showMessageDialog(myFrame,
"Name of the Instrument and Price is
Mandatory to Be Filled.");

         }
         else{
            try{
               String INSname =
NameINSsellTF.getText();
               String Prices =
PriceTF.getText();
               float pr =
Float.parseFloat(Prices);
               boolean isAvailable = false;

               if(instruments.isEmpty()){
                  InstrumentToSell INSsell =
new InstrumentToSell(pr,INSname);
```

```
                        if(INSsell instanceof
InstrumentToSell){
                            instruments.add(INSsell);



JOptionPane.showMessageDialog(myFrame,
"Instrument has been Added Successfully.");
                        }
                    }
                    else{
                        for(Instrument instrument:
instruments){
                            if(instrument instanceof
InstrumentToSell){

if(instrument.getInstrument_Name().equals(IN
Sname)){
                                    isAvailable= true;
                                }
                            }
                        }
                        if(isAvailable == true){

JOptionPane.showMessageDialog(myFrame,
"Instrument is Already Available.");
                        }
                        else{
                            InstrumentToSell INSsell =
new InstrumentToSell(pr, INSname);
                            if(INSsell instanceof
InstrumentToSell){
```

```
                                instruments.add(INSsell);



JOptionPane.showMessageDialog(myFrame,
"Instrument has been Added Successfully.");
                            }
                        }
                    }
                }
                catch(NumberFormatException
except){

JOptionPane.showMessageDialog(myFrame,
"Invalid Data Entered","Error",
JOptionPane.ERROR_MESSAGE);

                }
            }
        }

        //Rent the instrument button
        if(e.getSource()==RentINSbtn){

if(NameCustomerTF.getText().isEmpty() ||
PhoneNumTF.getText().isEmpty() ||
panTF.getText().isEmpty() ||
NameINSrentTF.getText().isEmpty()
            ||
ChargeperDayTF.getText().isEmpty()){
```

```
JOptionPane.showMessageDialog(myFrame,
"Please, Fill up all the Details");

            }
            else{
                try{
                    String CName =
NameCustomerTF.getText();
                    String PNum =
PhoneNumTF.getText();
                    String panNo = panTF.getText();
                    String INSname =
NameINSrentTF.getText();
                    String ChargePday =
ChargeperDayTF.getText();
                    String DaysNo =
NumofDaysTF.getText();
                    String DateReturn =
ReturnDateCBy.getSelectedItem().toString()
+
ReturnDateCBm.getSelectedItem().toString()
+
ReturnDateCBd.getSelectedItem().toString();
                    String DateRent =
RentDateCBy.getSelectedItem().toString() +
RentDateCBm.getSelectedItem().toString() +
RentDateCBd.getSelectedItem().toString();
                    int PAN =
Integer.parseInt(panNo);
```

```
                    int Charge =
Integer.parseInt(ChargePday);
                    int NOdays =
Integer.parseInt(DaysNo);

                    if(instruments.isEmpty()){

JOptionPane.showMessageDialog(myFrame,
"Instrument hasn't been Added For Rent");
                    }
                    else{
                        for(Instrument instrument:
instruments){
                            if(instrument instanceof
InstrumentToRent){
                                InstrumentToRent rent =
(InstrumentToRent)instrument;

if(rent.getInstrument_Name().equals(INSnam
e)){


if(rent.getIsRented()==true){

JOptionPane.showMessageDialog(myFrame,
"The Instrument has been Already Rented");

rent.Rent(CName,PNum, PAN, DateRent,
DateReturn, NOdays);
                                    }
                                    else{
```

```
JOptionPane.showMessageDialog(myFrame,
"The Instrument has been Rented");

rent.Rent(CName,PNum, PAN, DateRent,
DateReturn, NOdays);

rent.setIsRented(true);
                                    }
                                }
                            }
                        }
                    }
                }
                catch(NumberFormatException
except){

JOptionPane.showMessageDialog(myFrame,
"Invalid Data Entered","Error",
JOptionPane.ERROR_MESSAGE);

                }
            }
        }

        //Sell the Instrument Button
        if(e.getSource()==SellINSbtn){

if(NameCustomerTF.getText().isEmpty() ||
PhoneNumTF.getText().isEmpty() ||
panTF.getText().isEmpty() ||
NameINSsellTF.getText().isEmpty()
            || PriceTF.getText().isEmpty()){

JOptionPane.showMessageDialog(myFrame,
"Please, Fill up all the Details");

            }
            else{
                try{
                    String CName =
NameCustomerTF.getText();
                    String PNum =
PhoneNumTF.getText();
                    String panNo = panTF.getText();
                    String INSname =
NameINSsellTF.getText();
                    String Discount =
DiscountTF.getText();
                    String Price = PriceTF.getText();
                    String DateSell =
SellDateCBy.getSelectedItem().toString() +
SellDateCBm.getSelectedItem().toString() +
SellDateCBd.getSelectedItem().toString();
                    int PAN =
Integer.parseInt(panNo);
                    float Disc =
Float.parseFloat(Discount);
                    float Pr = Float.parseFloat(Price);

                    if(instruments.isEmpty()){
```

```
JOptionPane.showMessageDialog(myFrame,
"Instrument hasn't been Added For Sell");
          }
          else{
            for(Instrument instrument:
instruments){
              if(instrument instanceof
InstrumentToSell){
                InstrumentToSell Sells =
(InstrumentToSell)instrument;

if(Sells.getInstrument_Name().equals(INSna
me)){


if(Sells.getisSold()==true){

JOptionPane.showMessageDialog(myFrame,
"The Instrument has been Already Sold");

Sells.sell(CName,PNum, PAN, DateSell,
Disc);
                  }
                  else{

JOptionPane.showMessageDialog(myFrame,
"The Instrument has been Rented");

Sells.sell(CName,PNum, PAN, DateSell,
Disc);
```

```
                              Sells.setisSold(true);
                           }
                        }
                     }
                  }
               }
            }
            catch(NumberFormatException
except){

JOptionPane.showMessageDialog(myFrame,
"Invalid Data Entered","Error",
JOptionPane.ERROR_MESSAGE);

            }
         }
      }

      //Return the Instrument Button
      if(e.getSource()==ReturnINSbtn){
         if(
NameINSrentTF.getText().isEmpty()){

JOptionPane.showMessageDialog(myFrame,
"Please, Fill up the 'Instrument Name'");
         }
         else{
            String INSname =
NameINSrentTF.getText();
            if(instruments.isEmpty()){
```

```
JOptionPane.showMessageDialog(myFrame,
"Instrument hasn't been Rented.");
                }
                else{
                    for(Instrument instrument :
instruments){
                        if(instrument instanceof
InstrumentToRent){

if(instrument.getInstrument_Name().equals(IN
Sname)){
                                InstrumentToRent rent =
(InstrumentToRent)instrument;

if(rent.getIsRented()==true){

JOptionPane.showMessageDialog(myFrame,
"The Instrument Has Been Returned.");
                                    rent.setIsRented(false);
                                }
                                else{

JOptionPane.showMessageDialog(myFrame,
"Instrument Hasn't Been Rented. ");
                                }
                            }
                        }
                    }
                }
            }
```

```
        }

        //Display Rent Button
        if(e.getSource()== DisplayRentbtn){

if(NameCustomerTF.getText().isEmpty() &&
PhoneNumTF.getText().isEmpty() &&
panTF.getText().isEmpty() &&
NameINSrentTF.getText().isEmpty()
            &&
ChargeperDayTF.getText().isEmpty()){

JOptionPane.showMessageDialog(myFrame,
"NOTHING TO DISPLAY!! :(");
            }else{
                for(Instrument inst : instruments){
                    if(inst instanceof
InstrumentToRent){
                        InstrumentToRent rents =
(InstrumentToRent)inst;
                        rents.display();
                    }
                }
            }
        }

        //Display Sell Button
        if(e.getSource()== DisplaySellbtn){

if(NameCustomerTF.getText().isEmpty() &&
PhoneNumTF.getText().isEmpty() &&
panTF.getText().isEmpty() &&
NameINSsellTF.getText().isEmpty()
            && PriceTF.getText().isEmpty() &&
DiscountTF.getText().isEmpty()){

JOptionPane.showMessageDialog(myFrame,
"NOTHING TO DISPLAY!! :(");
            }else{
                for(Instrument inst : instruments){
                    if(inst instanceof
InstrumentToSell){
                        InstrumentToSell sells =
(InstrumentToSell)inst;
                        sells.display();
                    }
                }
            }
        }

        //Clear Rent Button
        if(e.getSource()== ClearRentbtn){

if(NameCustomerTF.getText().isEmpty() &&
PhoneNumTF.getText().isEmpty() &&
panTF.getText().isEmpty() &&
NameINSrentTF.getText().isEmpty()
            &&
ChargeperDayTF.getText().isEmpty()){

JOptionPane.showMessageDialog(myFrame,
"NOTHING TO CLEAR!! :(");
```

```
            }else{
                NameCustomerTF.setText("");
                PhoneNumTF.setText("");
                panTF.setText("");
                NameINSrentTF.setText("");
                ChargeperDayTF.setText("");
                NumofDaysTF.setText("");

RentDateCBy.setSelectedItem("Year");

RentDateCBm.setSelectedItem("Month");

RentDateCBd.setSelectedItem("Day");

ReturnDateCBy.setSelectedItem("Year");

ReturnDateCBm.setSelectedItem("Month");

ReturnDateCBd.setSelectedItem("Day");

JOptionPane.showMessageDialog(myFrame,
"Successfully cleared!! :)");
            }
        }

        //Clear Sell Button
        if(e.getSource()== ClearSellbtn){

if(NameCustomerTF.getText().isEmpty() &&
PhoneNumTF.getText().isEmpty() &&
panTF.getText().isEmpty() &&
NameINSsellTF.getText().isEmpty()
            && PriceTF.getText().isEmpty() &&
DiscountTF.getText().isEmpty()){

JOptionPane.showMessageDialog(myFrame,
"NOTHING TO CLEAR!! :(");

            }else{
                NameCustomerTF.setText("");
                PhoneNumTF.setText("");
                panTF.setText("");
                NameINSsellTF.setText("");
                PriceTF.setText("");
                DiscountTF.setText("");

SellDateCBy.setSelectedItem("Year");

SellDateCBm.setSelectedItem("Month");

SellDateCBd.setSelectedItem("Day");

JOptionPane.showMessageDialog(myFrame,
"Successfully cleared!! :)");
            }
        }
    }
}
```