LONDON METROPOLITAN UNIVERSITY

Module Code & Module Title

CS4001NA Programming

Assessment Weightage & Type

100% Individual Coursework

Year and Semester

2019-20 Spring

Student Name: Susan Shrestha

London Met ID: 19033540

College ID: NP05CP4S200039

Assignment Due Date: 13th September 2020

Assignment Submission Date: 13th September 2020

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 mark of zero will be awarded.

Table of Contents

Table of Figures

Table of Tables

Introduction

To commence with, the final coursework for programming (CS4001NT) module is all about programming with java. An object oriented programming language designed by James Gosling and developed by Oracle Corporation (Sun Microsystems) released in 1995. Java is platform independent language that means compiled java code can run on all platform that support java without the recompiling it.

For this coursework, students are asked to write the code for three classes as mentioned in the question paper namely; MicrosoftProduct, EnterpriseEdition and ProductGUI also attributes and methods for each class. Where, MicrosoftProduct class object stores product details such as Product Name and Product Number whereas EnterpriseEdition class inherits MicrosoftProduct class and its instance stores details of those Microsoft Products related to activation key, price of product, date and client company name. These all operations are carried out by the user through a GUI (Graphical User Interface) by providing input values to the text fields and performing operations such as Add Product, Activate License, Terminate License, Set price per user, Display, Clear. These user input are stored in an arraylist of MicrosoftProduct class using instance of EnterpriseEdition class.

GUI in java requires AWT (Abstract Window Toolkit) that is an API which provides GUI interface for a java program. AWT precedes swing which is also a part of java foundation classes that provides GUI for java programs but swing was developed to provide more GUI components to programmers than AWT. So I will be using both AWT and java Swing for my GUI.

For coding purpose, I use BlueJ IDE for ease and as per the instruction of coursework question paper. BlueJ IDE was originally developed by the BlueJ Team. It is designed for the beginners and was mainly developed for educational purpose. The main screen shows the UML (Unified Modeling Language) of class structure of application under development which makes newbies to easily understand the class relation.

2. Class Diagram:

Class diagram in Unified Modeling Language (UML), can be defined as a graphical representation or diagram that defines the overall structure of program or application by presenting the classes with their methods, attributes and relationship between them.

Following the definition and guidelines of making class diagram, for this coursework I have created class diagram of three classes namely; MicrosoftProduct, EnterpriseEdition and ProductGUI and shown their relationship as created by blueJ IDE. For this particular task, I have used cloud based application called draw.io, it is very easy and reliable to use for such work.

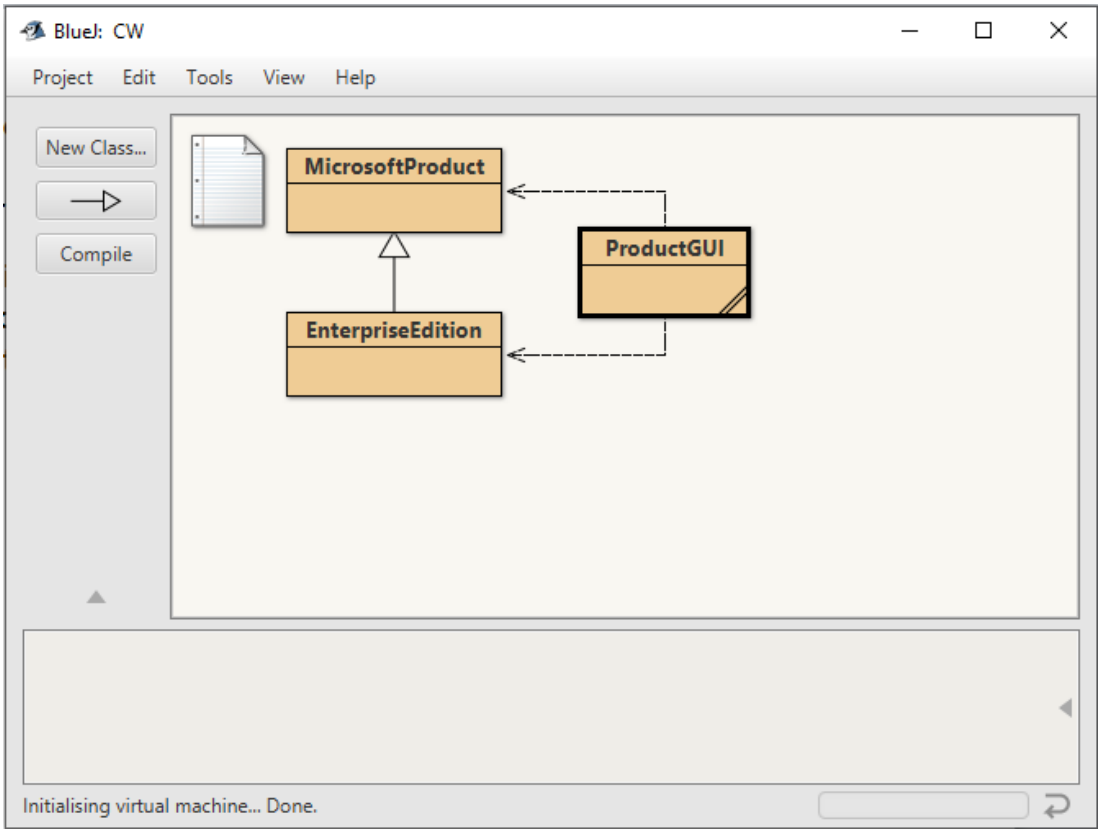


Figure 1: BlueJ class window

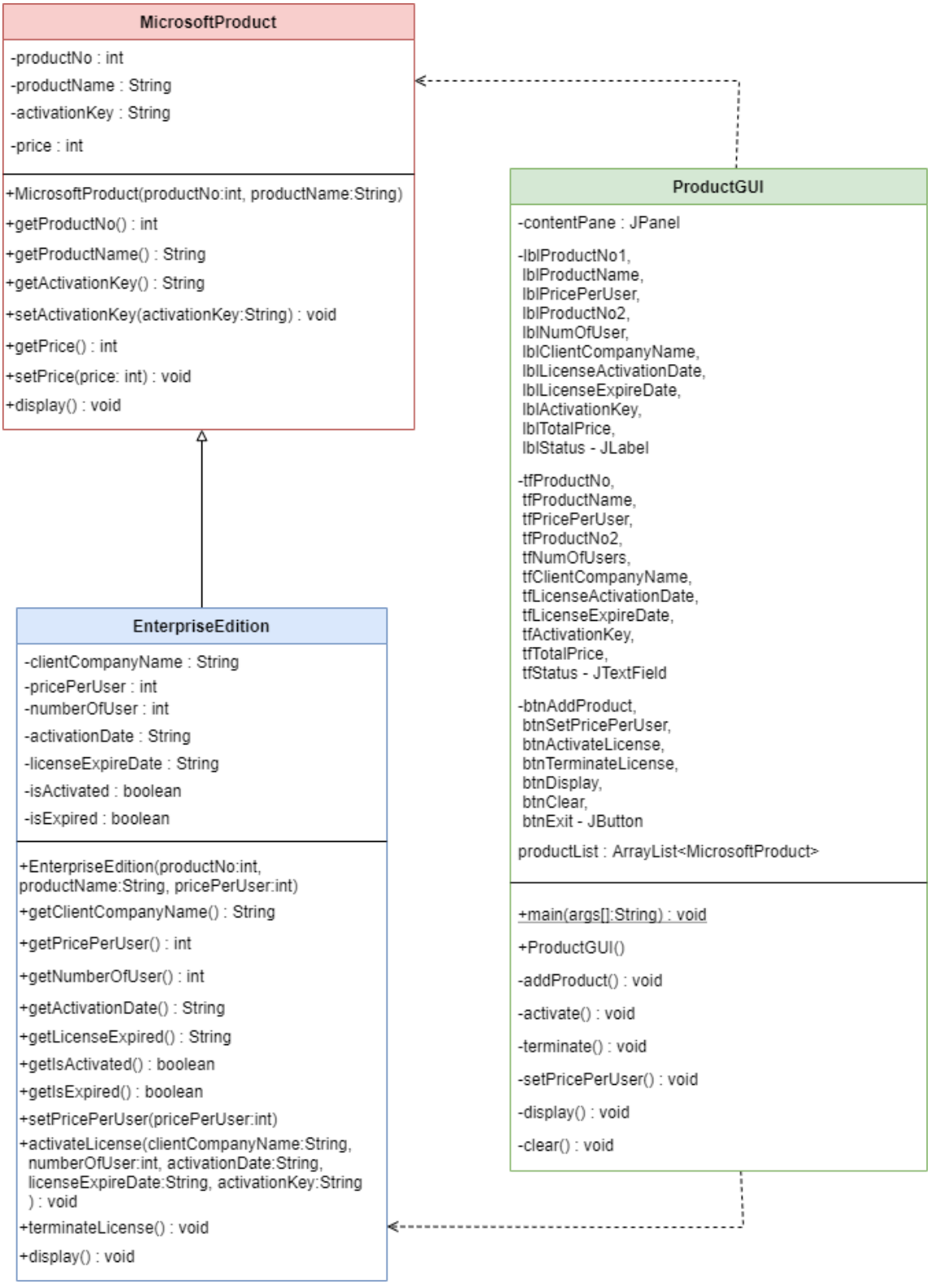


Figure 2: Class Diagram

3. Pseudocode

3.1. MicrosoftProduct

```
START
    CONSTRUCTOR MicrosoftProduct(productNo, productName):
        Store productNo, productName, activationKey and price
    METHOD getProductNo():
        RETURN productNo
    Method getProductName():
        RETURN productName
    METHOD getActivationKey():
        RETURN activationKey
    METHOD setActivationKey(activationKey):
        Store activationKey
    METHOD getPrice():
        RETURN price
    METHOD setPrice(price):
        Store price
    METHOD display():
        PRINT product number and product name
        IF activationKey and price not null
            PRINT activation key and price
STOP
```

3.2. EnterpriseEdition

START

CONSTRUCTOR EnterpriseEdition(productNo, productName, pricePerUser):

CALL super(productNo, productName)

Store pricePerUser, numberOfUser, activationDate, licenseExpireDate, isActivated, isExpired

METHOD getClientCompanyName():

RETURN clientCompanyName

METHOD getPricePerUser():

RETURN pricePerUser

METHOD getNumberOfUser():

RETURN numberOfUser

METHOD getActivationDate():

RETURN activationDate

METHOD getLicenseExpireDate():

RETURN licenseExpireDate

METHOD getIsActivated():

RETURN isActivated

METHOD getIsExpired():

RETURN isExpired

METHOD setPricePerUser(pricePerUser):

IF NOT isActivated

SET pricePerUser

PRINT " New price per user is set"

ELSE

PRINT “License is activated, cannot change price per user”

METHOD activateLicense (clientCompanyName, numberOfUser, activationDate licenseExpireDate, activationKey):

IF isActivated

PRINT “Already activated”

ELSE

SET clientCompanyName, numberOfUser, activationDate, licenseExpireDate, isActivated as true, isExpired as false

CALL super, set activationKey, price(numberOfUser * pricePerUser)

PRINT “Activation successfully done”

METHOD terminateLicense():

IF isExpired

PRINT “Not activated yet”

ELSE

SET clientCompanyName, activationDate, licenseExpireDate, numberOfUser, isActivated, isExpired to NULL

METHOD display():

CALL super.display()

IF isActivated

PRINT clientCompanyName, NumberOfUser, pricePerUser, totalPrice, activationDate, LicenseExpireDate, activationKey, activationStatus

STOP

3.3. ProductGUI

START

METHOD main():

METHOD run():

Initialize GUI Frame as ProductGUI()

Make frame Visible

CONSTRUCTOR ProductGUI():

Set window resizable as FALSE

Set title as "Microsoft Product Details"

Set window size (635,498)

ContentPane as new JPanel

Create JLabel, JTextField and JButton and set their position and bounds

For all JButton actionPerformed(actionEvent)

CALL respective Method

Create ArrayList of type MicrosoftProduct as productList

METHOD addProduct():

GET Product Number, Product Name and Price per User from user input

IF (Product Number or Product Name or Price per User equals null)

PRINT “Fill the form”

ELSE

TRY

CONVERT Product Number and Price per User to integer

CATCH NumberFormatException

PRINT “check your input format and try again”

RETURN

FOR mp as object of MicrosoftProduct:productList

IF(check mp instance of EnterpriseEdition)

Create ee as object of EnterpriseEdition storing mp of type EnterpriseEdition for class clasting

IF (ee.getProductNo() equals to Product Number)

PRINT “Product already added”

Create eeList as an object of EnterpriseEdition(Product Number,Product Name,Price per User)

ADD eeList in productList

PRINT “Product added”

METHOD activate():

GET Product Number, Number of User, Client Company Name, License Activation Date, License Expire Date, Activation Key from user input

IF (Product Number or Number of User or Client Company Name or License Activation Date or License Expire Date or Activation Key equals null)

PRINT “Fill all the form”

ELSE

TRY

CONVERT Product Number and Number of User to integer

CATCH NumberFormatException

PRINT “Invalid input format”

DECLARE flag as Boolean as FALSE

FOR i in 0 to productList.size()

Object of MicrosoftProduct stores productList.get(i)

IF (mp instance of EnterpriseEdition)

DO Class casting

IF (ee.getProductNo() is equals to Product Number)

Set flag to TRUE

IF (ee.getIsActivated() is TRUE)

PRINT “Already activated”

ELSE

CALL activateLicense(Number of User, Client Company Name, License Activation Date, License Expire Date, Activation Key) of EnterpriseEdition class

PRINT “Product Activated successfully”

BREAK

IF (flag is FALSE)

PRINT “Product not found”

METHOD terminate():

GET Product Number from user input

TRY

IF (Product Number is null)

PRINT “Enter product number”

CATCH NullPointerException

RETURN

TRY

CONVERT Product Number to integer

CATCH NumberFormatException

PRINT “Invalid input format”

DECLARE active as Boolean as FALSE

FOR (MicrosoftProduct mp: productList)

IF (mp instance of EnterpriseEdition)

DO class casting for ee to store object of mp

IF (mp.getProductNo is equals to Product Number)

Set active as TRUE

CALL terminateLicense() from EnterpriseEdition

PRINT “License Terminated”

IF (active is FALSE)

PRINT “product number not found”

METHOD setPricePerUser():

GET Product Number and Price per User form user input

TRY

IF (Product Number and Price per User equals null or Price per User equals 0)

PRINT “Enter the Product Number and Price per User correctly”

CATCH NullPointerException

RETURN

TRY

CONVERT Product Number and Price per User to integer

CATCH NumberFormatException

RETURN

Set flag as Boolean as FALSE

FOR (MicrosoftProduct mp: productList)

DO Class casting

IF (mp.getProductNo() is equal to Product Number)

Set flag to TRUE

IF (ee.getPricePerUser() is equal to Price per User)

PRINT “Same price per user, try again”

ELSE IF (ee.getIsActivated is TRUE)

PRINT “Cannot set new price per user”

ELSE

CALL setPricePerUser(Price per User) from EnterpriseEdition

PRINT “new price per user is set”

BREAK

IF (flag is FALSE)

PRINT “Product not found”

METHOD display():

Set flag as Boolean as FALSE

FOR (MicrosoftProduct mp: productList)

IF (mp instance of EnterpriseEdition)

DO class casting

Set flag as TRUE

CALL display() from EnterpriseEdition

IF (flag is FALSE)

PRINT “no product added”

METHOD clear():

ASK “Do you want to clear all fields?”

IF (Yes)

Set all TextField as null

STOP

4. Method description

- A short method description of method of all classes.

4.1. For class MicrosoftProduct (super class)

1. MicrosoftProduct(int productNo, String productName)

This is the constructor method of class MicrosoftProduct which stores object of this class and accepts input arguments to assign the values in attributes of the class. In our case we are passing productNo and productName as parameter to constructor method to store product details.

2. display()

This is a general method which prints the stored details in the object of class MicrosoftProduct.

4.2. For class EnterpriseEdition (sub class)

1. EnterpriseEdition(int productNo, String productName, int pricePerUser)

This is the constructor method of class EnterpriseEdition. This constructor method accepts input values as parameter i.e. productNo, productName and pricePerUser and calls super class using 'super' keyword and pass the values to super class.

2. setPricePerUser(int pricePerUser)

This method is used to set the price per user for a given product by user. This works only when the license of the given product is not activated.

3. activateLicense(String clientCompanyName, int numberOfUser, String activationDate, String licenseExpireDate, String activationKey)

Here, client company name, number of user, activation date, license expire date and activation key are provided by the user as input and it is passed as parameter to this method which is further used to activate license of the given product number if not

activated and stored as product details. This method sets the activation status to true and expired as false.

4. terminateLicense()

This particular method cancels the product license by setting all product activation details as null. This works for the activated product only. User has to decide whether to terminate the license or not.

5. display()

It is used to print the stored product details on terminal screen but only if the license is activated. Although it first calls the display() method of superclass (MicrosoftProduct) using 'super' keyword.

4.3. For class ProductGUI

1. public static void main(String[] args)

This is the main method of class ProductGUI from which the execution of program starts. Inside this method, I created another method called run() that will make our GUI runnable and visible to the user.

2. ProductGUI()

This is the constructor method of this class where our GUI form is designed considering the position, window size, window title, fonts and components properties.

3. addProduct()

Name itself represents this method is used for adding product to the array list we created. Here, this method takes product number, product name and price per user as input from the user and checks if the product number is already in the list or not, if the product is already in the list it will just return a message "product already in the list" else it will add those details to the array list and prints "product added successfully".

4. activate()

This method is used to activate the product in array list which are not previously activated. Here, the given product is checked if it is activated or not, if it is already activated it prints “product already activated” else it will take the user input and store those values in product list. For this, it calls activateLicense() method from the class EnterpriseEdition.

5. terminate()

This method asks the user to input the product number of which user want to terminate the license, after that it checks if the license is activated or not and product number is in the list or not, if the condition is valid then it calls the terminateLicense() method from the EnterpriseEdition class and cancels the license for given product number.

6. setPricePerUser

This particular method is used for setting the product price per user only if the license for the product is not activated. It takes product number and price per user field as input and checks if the product no exists in the list and if it does then the setPricePerUser() method from the class EnterpriseEdition is called and user input price per user value is provided as parameter in order to set new price per user.

7. display()

This method is used to print the product list in terminal window. This method calls display() method of EnterpriseEdition class.

8. clear()

This method asks user to whether clear the form or not, if yes pressed by user it clears everything in input fields.

5. Testing

Testing in software development is one of the important part to validate the final product considering its reliability, performance and speed. One can test the program by giving invalid input and see how the program responds on that. Also the programmer can identify bugs and errors on the program and code after doing some general testing that will help in debugging and improve the performance of the program.

In my coursework, I have done some several tests on my program as asked in the question paper and also to rectify errors and to fix it. So the tests are documented below.

Test: 1

Testing if my code file can be compiled and run using command prompt,

Screenshot:

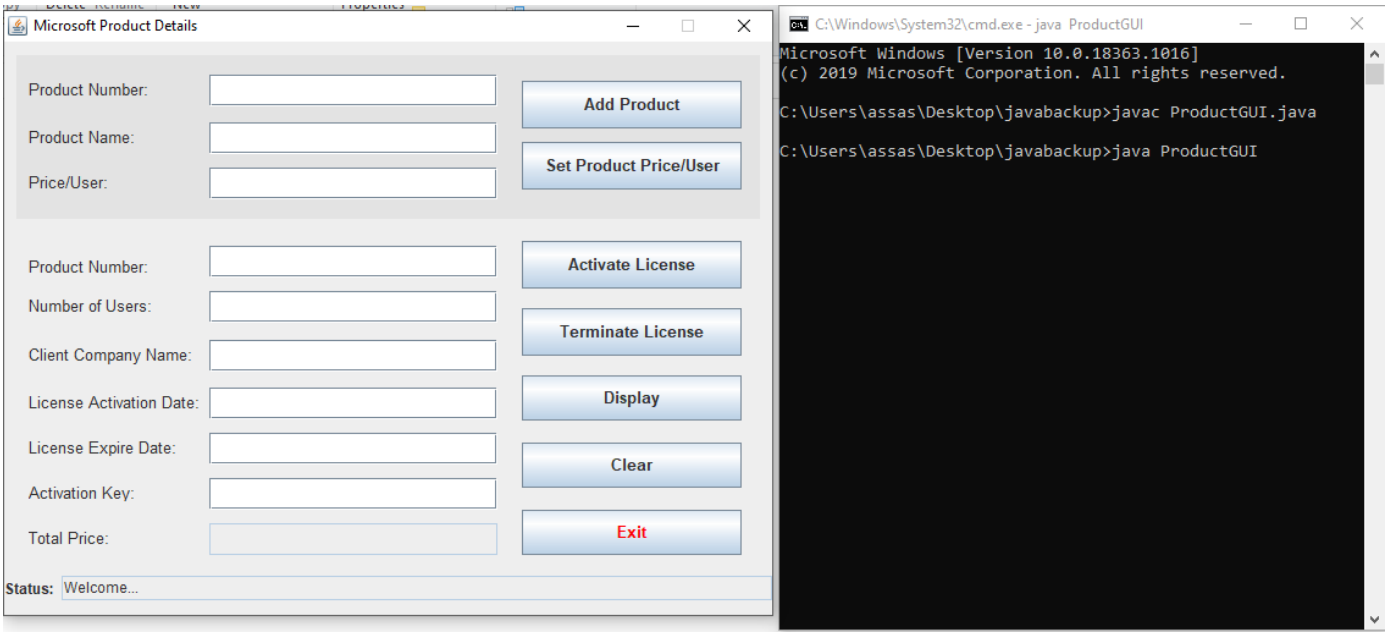


Figure 3: Compilation and running using CMD (Test 1)

As shown above, in the screenshot, I compiled and run my program using command prompt (CMD).

Test: 2

Testing my program to check if it's all functionalities are working well,

a. Add product

Table 1(test 2.a)

Objective	To add a Microsoft product
Action	Enter data into Text field as, Product Number: 102 Product Name: Windows 10 Price/User: 500 Then click on Add Product button
Expected Result	Product will be added and product info pops up
Actual Result	Product successfully added pop up message with details
Conclusion	Test succeed

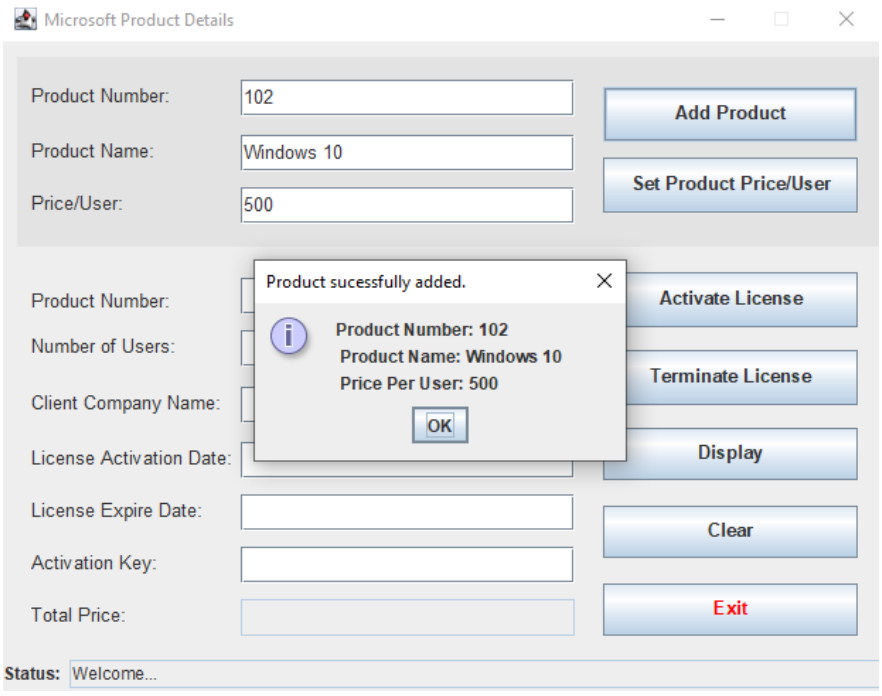


Figure 4: Adding Product (Test 2)

b. Activate the product

Table 2(Test 2.b)

Objective	To activate the given product
Action	Enter data into Text field as, Product Number: 102 Number of User: 3 Client Company Name: susZ Tech License Activation Date: 20 Jan 2020 License Expire Date: 25 Dec 2021 Activation Key: W269N-WFGWX-YVC9B-4J6C9-T83GX Then click on Activate License button
Expected Result	License will be activated and product details get updated
Actual Result	Product activated successfully pop up message
Conclusion	Test succeed

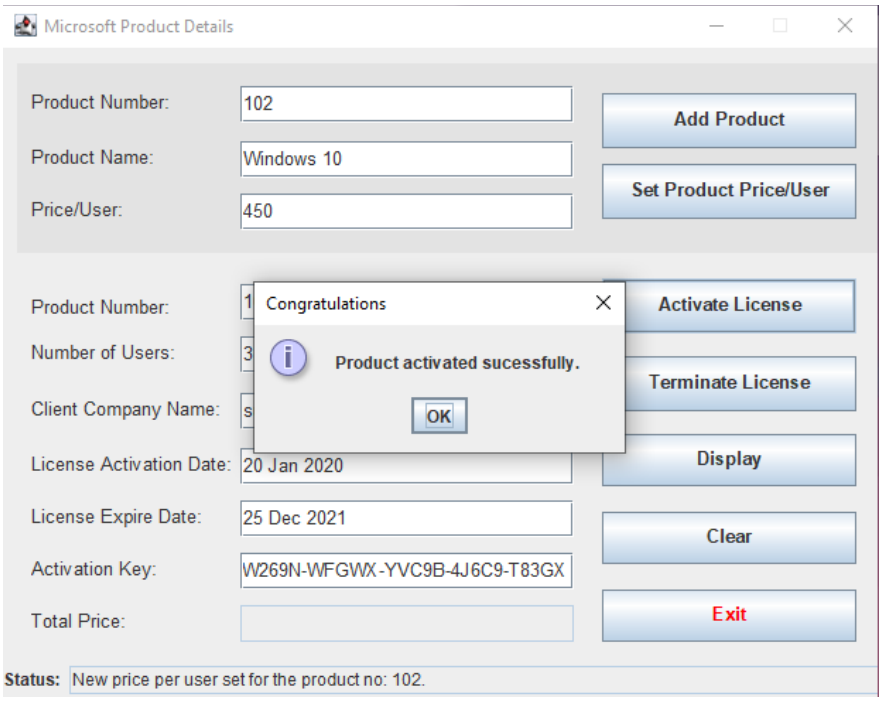


Figure 5: Activating Product (Test 2)

c. Terminate the product

Table 3(Test 2.c)

Objective	To terminate the given product
Action	Click on Terminate License button, it will ask to enter the product number to terminate license Enter data into Text field as, Product Number: 102 Then click on OK button
Expected Result	Product license will be terminated and product details get updated
Actual Result	Product license has been terminated successfully pop up message
Conclusion	Test succeed

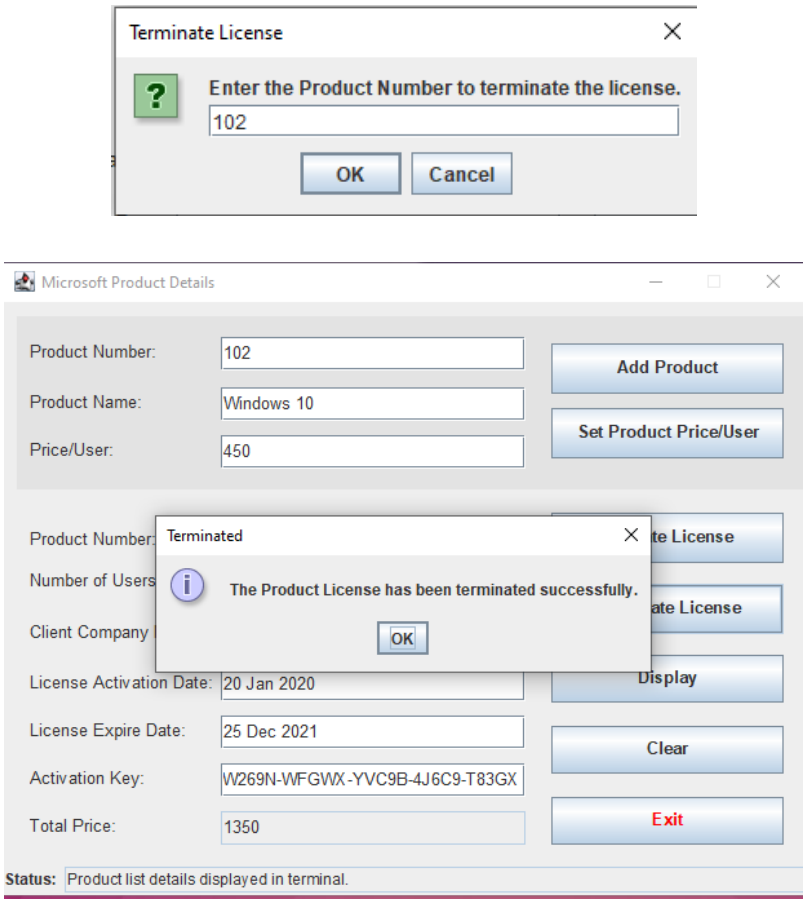


Figure 6: Terminating License (Test 2)

d. Set price per user

Table 4(Test 2.d)

Objective	To set product price per user
Action	Enter data into Text field as, Product Number: 102 Price/User: 450 Then click on Set Product Price/User button
Expected Result	Price per user will be set from 500 to 450 and list will be updated
Actual Result	New price per user set for product No. 102
Conclusion	Test succeed

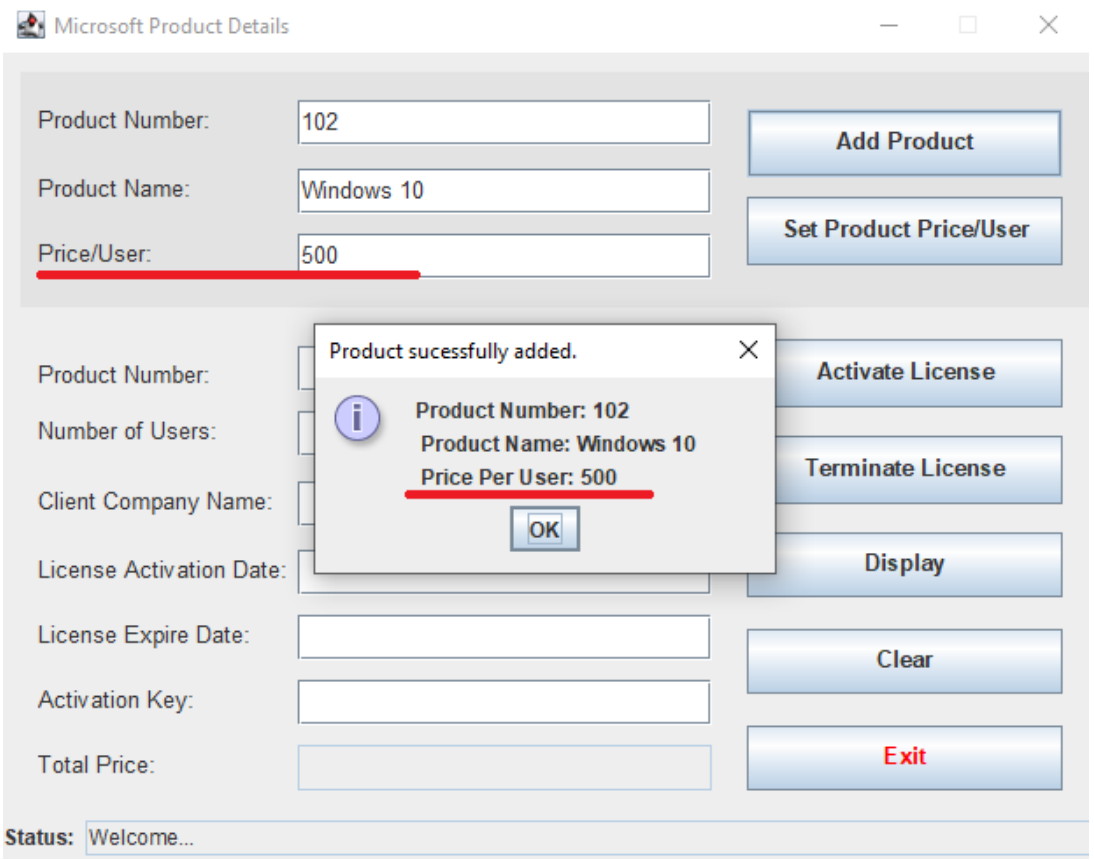
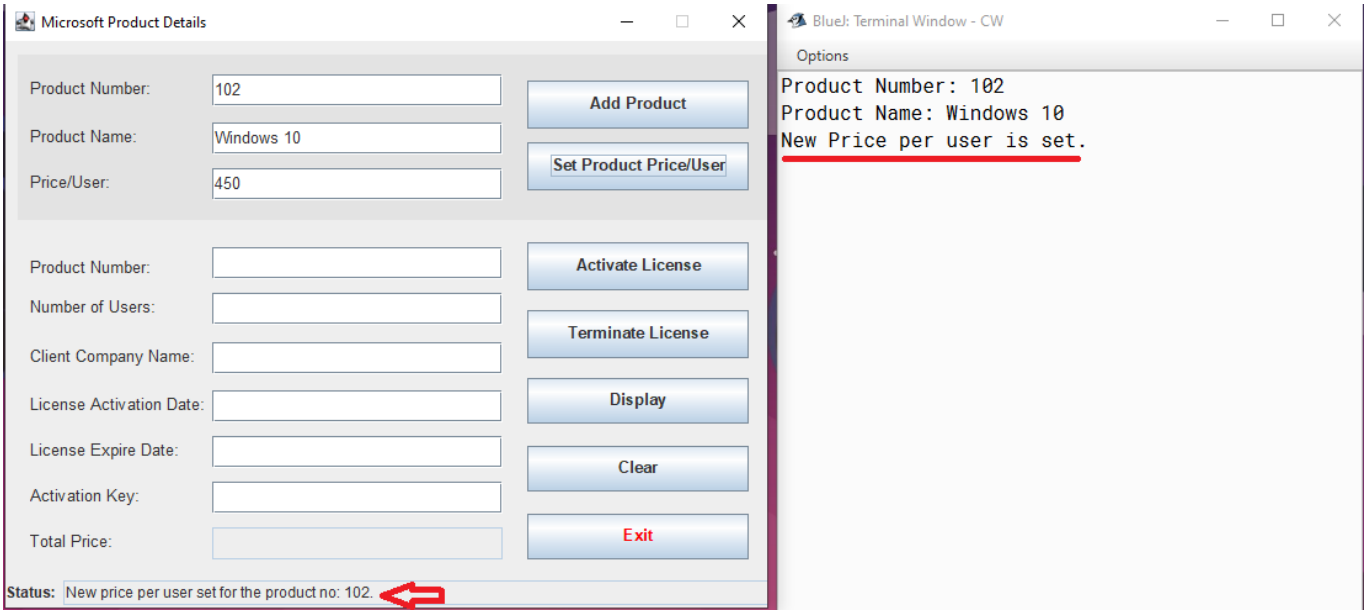


Figure 7: Setting Price Per User (Test 2)



e. Display the Product List

Table 5(Test 2.e)

Objective	To display the product list
Action	Click on display button
Expected Result	Product list will be displayed in terminal
Actual Result	Product list displayed
Conclusion	Test succeed

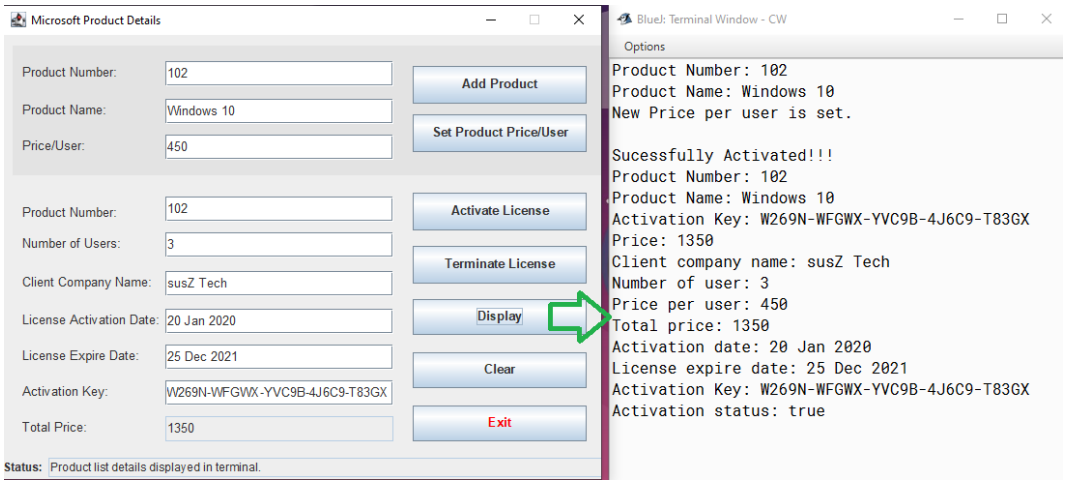


Figure 8: Displaying Product Details (Test 2)

f. Clear form

Table 6(Test 2.f)

Objective	To clear all fields on GUI
Action	Click on clear button, Click on Yes button
Expected Result	All fields will be cleared
Actual Result	Clears all text field
Conclusion	Test succeed

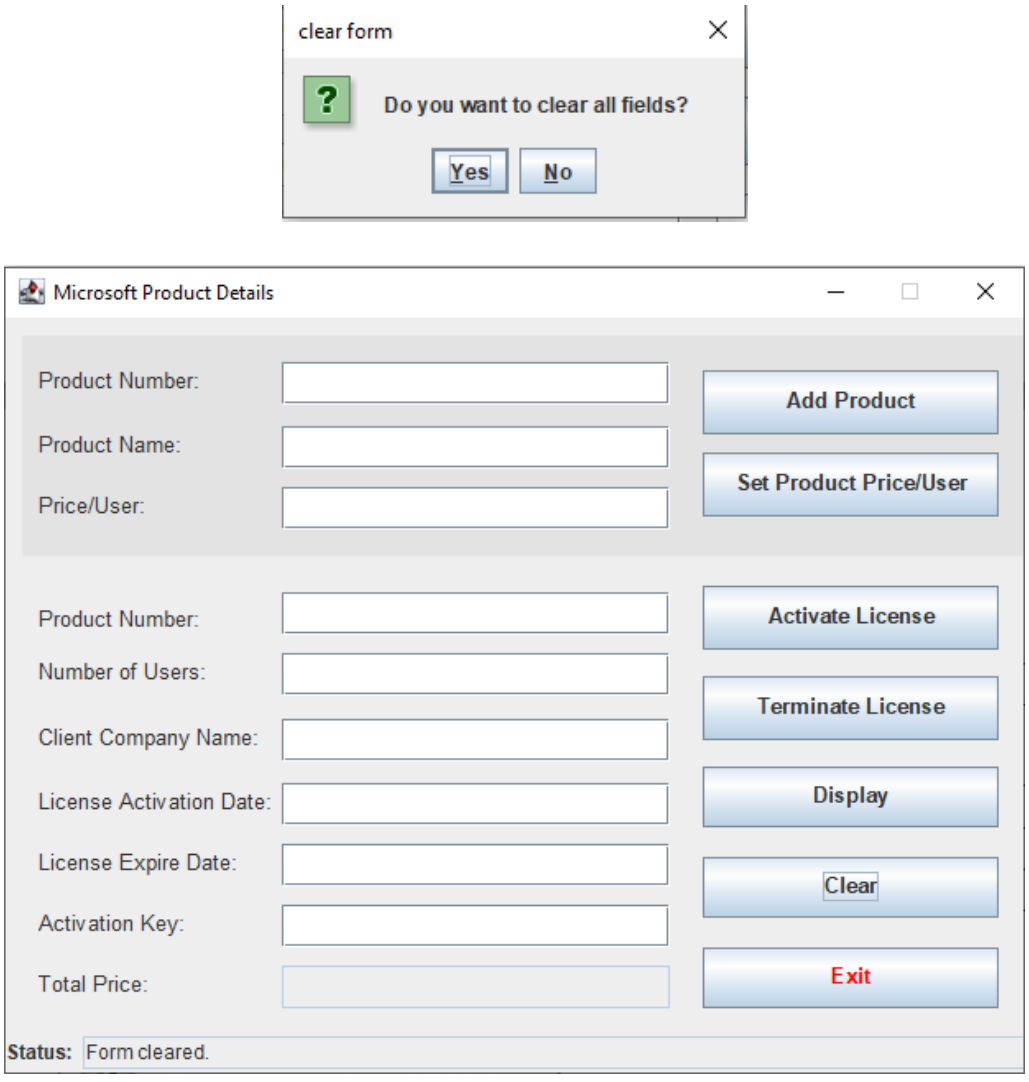


Figure 9: Clearing Form (Test 2)

Test: 3

a. Trying to add duplicate product number

Table 7(Test 3.a)

Objective	To test duplicate entry
Action	Enter the same product number that already exist in the product list Then click on Add Product button
Expected Result	given product is already added error message will be shown
Actual Result	This product is already added error pops up
Conclusion	Test succeed

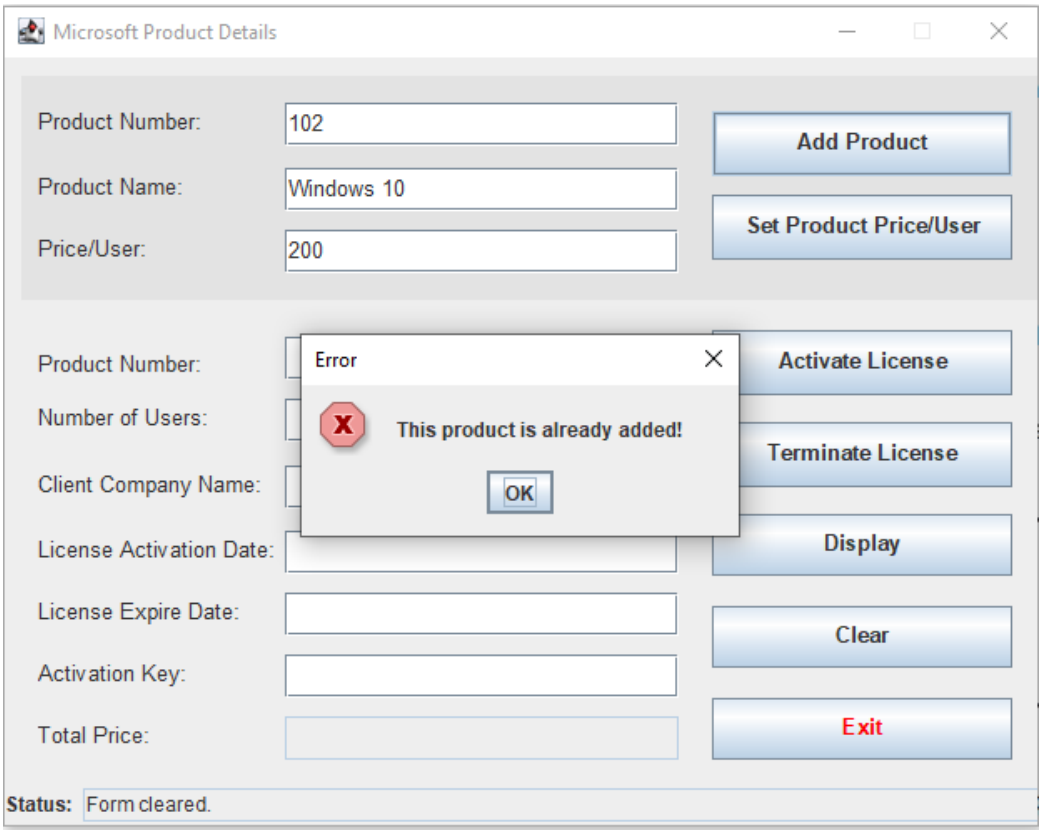


Figure 10: Duplicating product (Test 3)

b. Trying to activate already activated product

Table 8(Test 3.b)

Objective	To try activating already activated product
Action	Enter the product number that is already activated and fill other fields, Then click on Activate License button
Expected Result	Given product is already activated message will be shown
Actual Result	The product is already activated message pops up
Conclusion	Test succeed

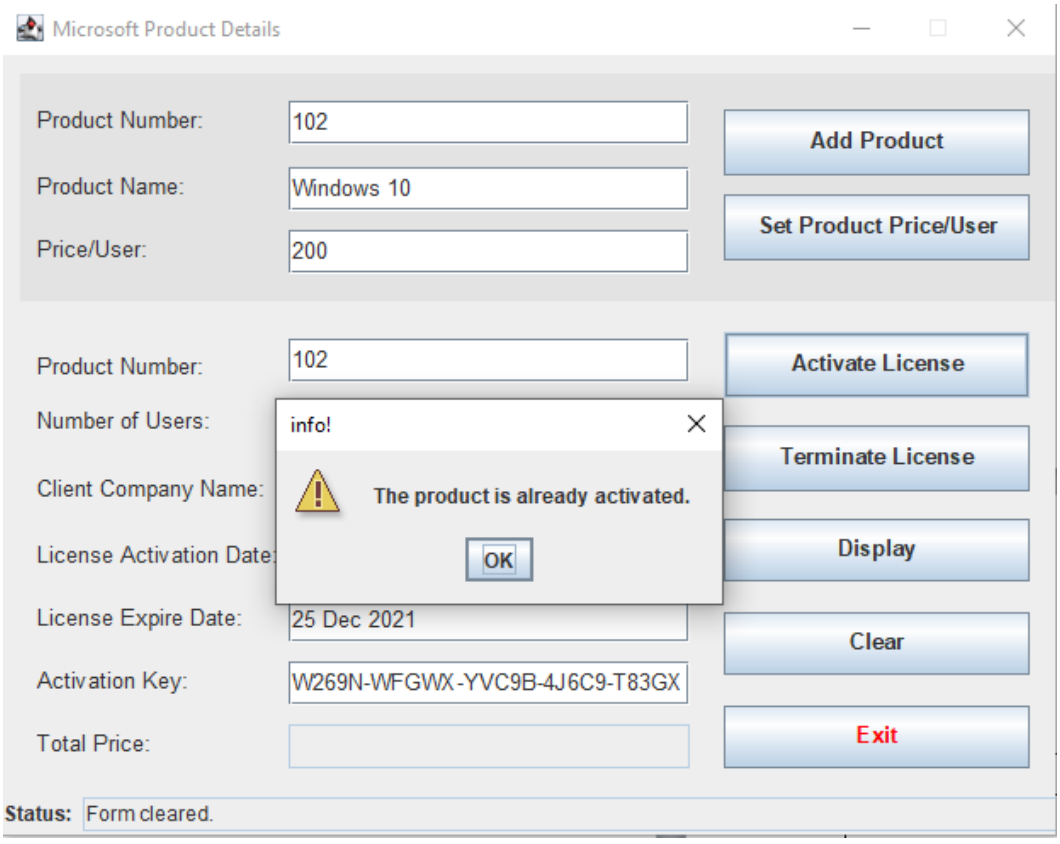


Figure 11: Activating already activated product (Test 3)

c. Trying to set new price per user for activated product

Table 9(Test 3.c)

Objective	To try setting new price per user for activated product
Action	Enter the new price per user in text field for already activated product number Then click on Set Product Price Per User button
Expected Result	Warning message: already activated product cannot change price per user
Actual Result	Warning: License is already activated for this product cannot change price per user
Conclusion	Test succeed

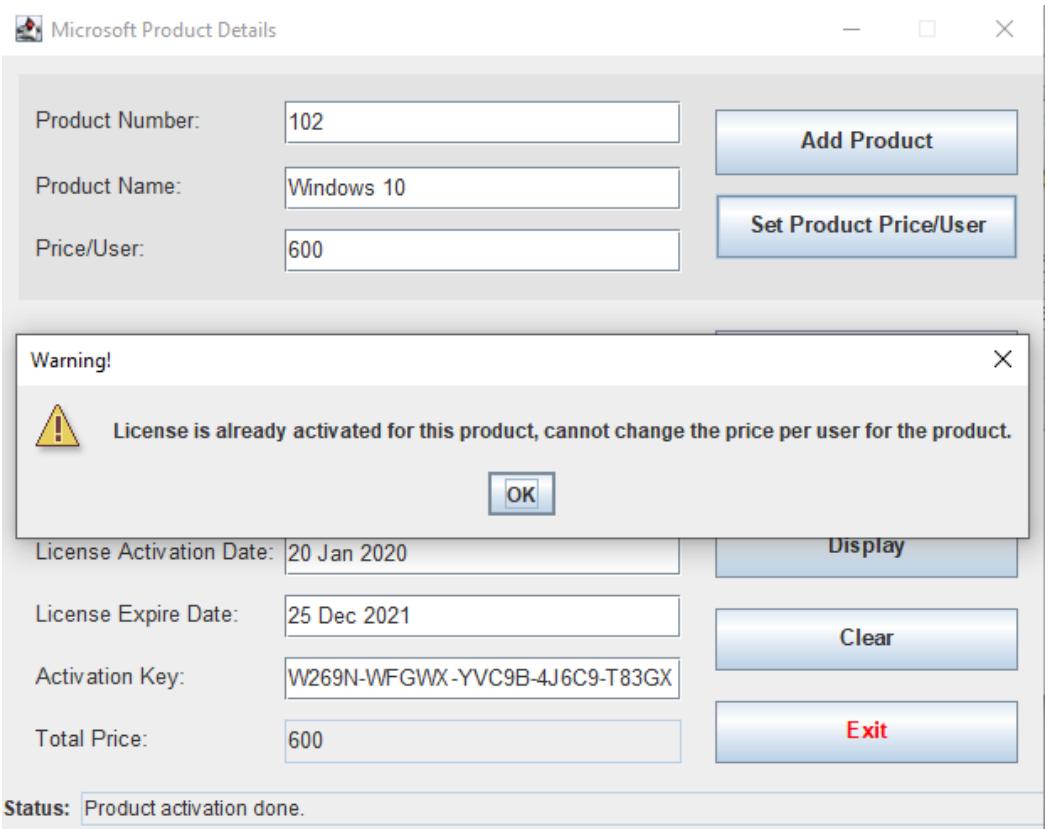


Figure 12: Setting Price Per User for activated product (Test 3)

6. Error Handling

Errors are mistakes that a programmer generally makes while coding such as missing part of code, missing syntax or it can be wrong logic in a program which leads to compile failure or unexpected behavior in a program. These errors can be detected and corrected by proper use of knowledge and skill on errors.

In general, there are 3 types of error in programming namely; syntax error, logic/semantic error and runtime error. So, I have collected some few mistakes or errors that I got during my coding part of coursework.

6.1. Syntax error

This error is most common type of mistake that every programmer probably does during coding. In this, programmer generally commits mistakes in syntax of the code such as missing terminator sign (;) after statements, missing braces in code blocks, uppercase or lowercase mistakes in keywords, etc.

In my case,

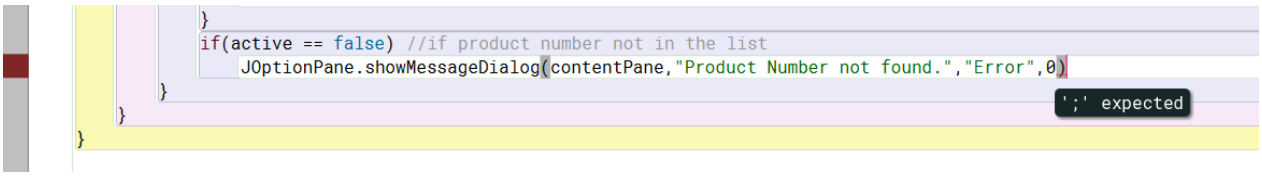


Figure 13: Syntax error

Here, while I was trying to compile my code, it didn't compiled and Error found in class message was displayed, and the mistake part of code was highlighted by a red underline and brown color on the left side of IDE in the same line, that is how I detected syntax error in my code.

To fix it,

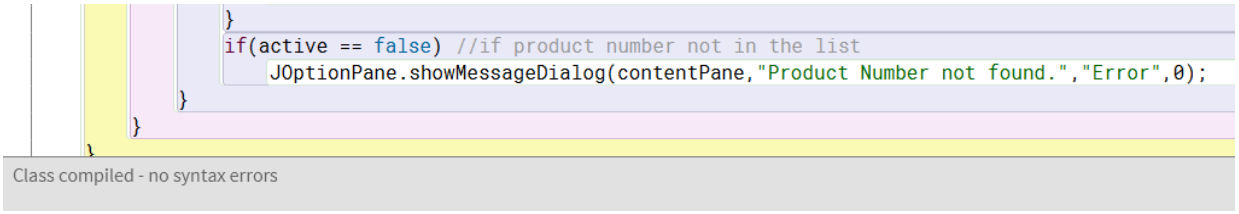


Figure 14: Fixing Syntax error

I just put the terminator (;) at the end of the statement as it is the syntax of java and also it was mentioned by the ID ';' expected, that's how I fixed it.

Also, for detecting syntax errors we can just move to the line where error occurred and then we can mouse over the red underline on the error and ToolTipsBox message hint is displayed for fixing those errors.

6.2. Logic/Semantic error

This type error can be caused by the misunderstanding of logic or confusion by the programmer. Logical errors are very difficult to detect because the program does not crash or fail but misbehaves or produces unexpected output. But logic and semantic can be different sometime as logic error produces wrong data but semantic error produces the result that are not meaningful at all.

In my code,

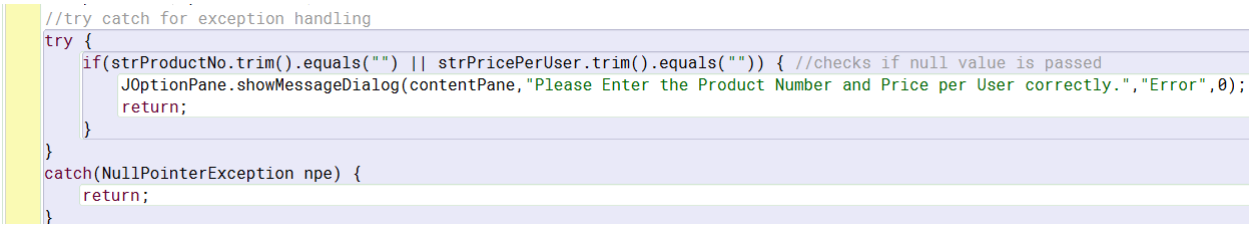
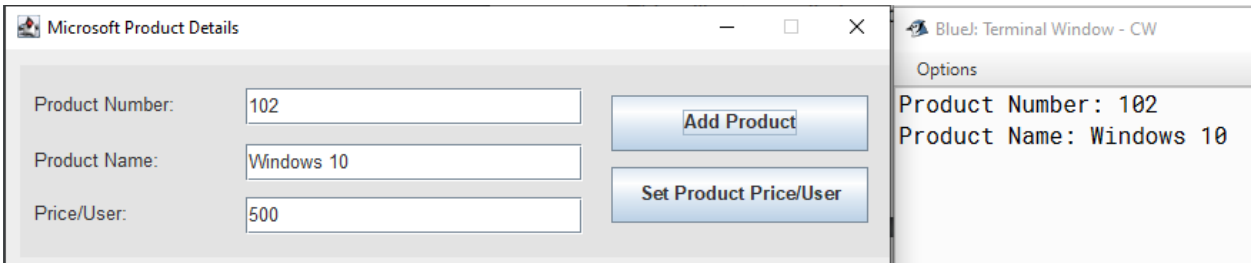


Figure 15: Logical error (code)

Here, I used try catch block for exception handling for null pointer exception during runtime but the point is I have used If clause for checking if null value is

passed then return suitable message.

So after running and entering values in program,



For product number 102, I entered price as 500 while adding it to the list.

But after adding the product to arraylist I tried to set the price as 0 which is not allowed for any product in our program which will produce wrong data later.

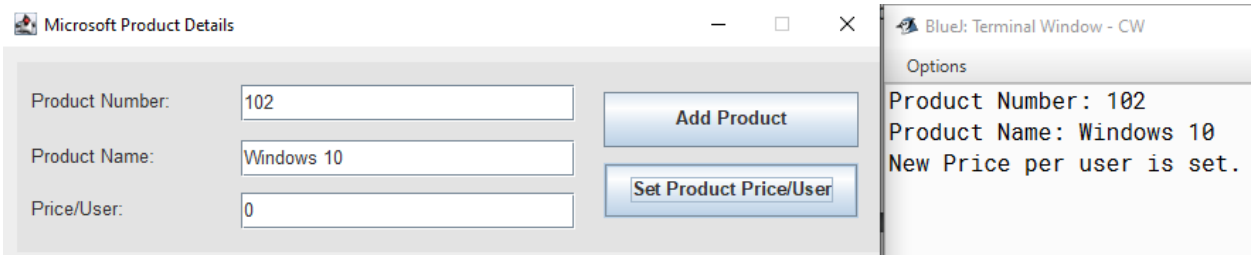


Figure 16: Program result of Logical error

The program takes 0 as input that means we are having logical error in this.

So to fix this we can just add a code to check if user should not enter 0 as input and the program does not take 0 as input.

```
try {
    if(strProductNo.trim().equals("") || strPricePerUser.trim().equals("") || strPricePerUser.trim().equals("0")) { //checks if null value is passed
        JOptionPane.showMessageDialog(contentPane,"Please Enter the Product Number and Price per User correctly.","Error",0);
        return;
    }
}
catch(NullPointerException npe) {
    return;
}
```

Figure 17: Logical error fix

Now, when user tries to enter the price per user as 0 it will return product number not entered correctly.

6.3. Runtime error

This type of error is those errors which are detected while the program is running; this is caused due to syntax error or wrong coding and also not proper use of exception handling in code.

To clarify it more,

I am removing a try catch block of my code and then try to input string where integer is required. So let's see how the program responds,

```
//try {
    //converting String to integer type
    productNo = Integer.parseInt(strProductNumber);
    pricePerUser = Integer.parseInt(strPricePerUser);
//}
//exception handling if invalid input is given by user and to continue the program instead of crashing
//catch(NumberFormatException nfe) {
   // JOptionPane.showMessageDialog(contentPane,"Please check your input format and try again" ,"Error", 0);
   // return;
//}
```

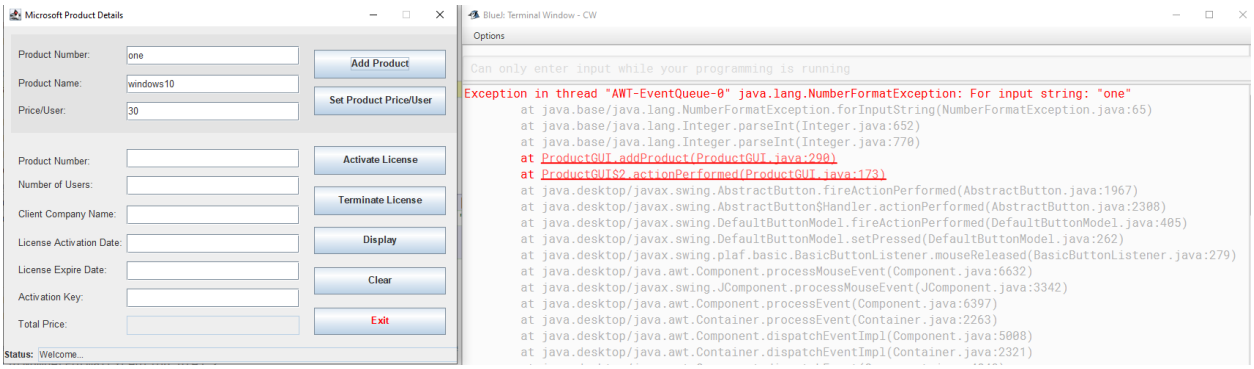


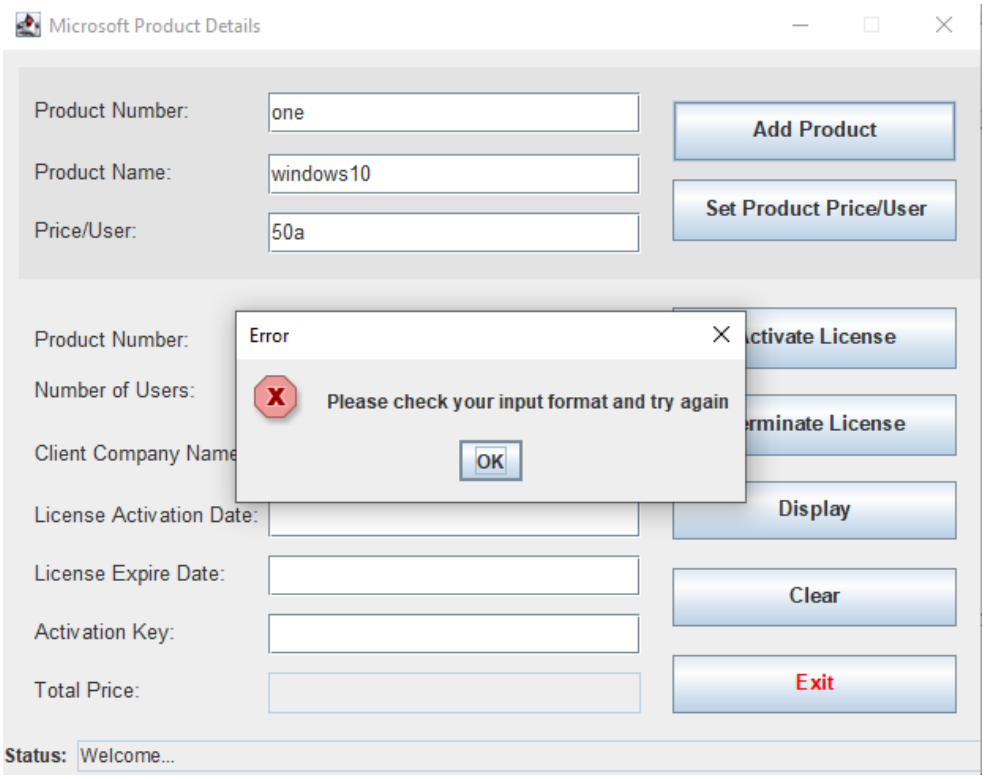
Figure 18: Runtime error

As there was no errors while compiling the code but after I run the program and try to add product by entering the product no value as 'one' and '30a' in price per user, it gives error: Number Format Exception as the program can only take integer data type for product number and price per user .

Now, I will just use try catch block there to see if how program responds after that,

```
try {
    //converting String to integer type
    productNo = Integer.parseInt(strProductNumber);
    pricePerUser = Integer.parseInt(strPricePerUser);
}
//exception handling if invalid input is given by user and to continue the program instead of crashing
catch(NumberFormatException nfe) {
    JOptionPane.showMessageDialog(contentPane,"Please check your input format and try again" ,"Error", 0);
    return;
}
```

Figure 19: Runtime error fix



After, I used exception handling it helped my program to run without any crashes instead it allows the program to respond by using suitable message to user. This is how I fixed exceptions and runtime errors in my coursework coding.

Conclusion

To conclude, this coursework was all about GUI program development using Java, where I learnt various new stuffs in java programming like AWT, java swing components, GUI designing, arraylist, class casting, exception handling using try catch and many more concepts of GUI. I also learned some concepts about documenting the java project such as making class diagram to represent program structure and relationship between classes, writing pseudocode to understand the program easily, testing the program correctness and reliability by creating test cases, writing method description and finally learned about error handling where I knew about types of error and understood how to detect the error and how to correct those errors.

Thanking to my respected teachers who were very responsive and helpful during the online classes and answered my unsolved questions and guiding me on this particular coursework. As this is individual coursework, I also got an opportunity to be self-dependent on these types of challenges. I also knew many new technical terms which arrived during my coursework by researching about them on internet, also I learned about many programming techniques to fix my problems during coding phase like class casting, exception handling and GUI form handling.

At last, this individual coursework helped me to gain good knowledge on programming with java and developed my skills. Hope I can harness my skills more in java programming and use this skill to develop some better programs in upcoming days.

Appendix

1. MicrosoftProduct (Parent Class)

```
/*
 * MicrosoftProduct is a class that has four attributes namely; productNo, productName, activationKey and price.
 * These attributes will store the input values passed from the GUI through object created and passing those
   value as parameters in the constructor.
 * Getter method are used to access the value stored in this class to other class or sub class.
 * Author: Susan Shrestha
 */
public class MicrosoftProduct {
    //declaring attributes as private access modifier
    private int productNo;
    private String productName;
    private String activationKey;
    private int price;

    //creating constructor method of MicrosoftProduct class
    public MicrosoftProduct(int productNo, String productName) {//passing parameter values to attributes of this class through constructor
        this.productNo = productNo;
        this.productName = productName;
        this.activationKey = "";
        this.price = 0;
    }
```

```
    //using getter and setter method for productNo, productName, activationKey, price;
    public int getProductNo() {
        return productNo; //returns productNo
    }

    public String getProductName() {
        return productName; //returns productName
    }

    public String getActivationKey() {
        return activationKey; //returns ActivationKey
    }

    public void setActivationKey (String activationKey) {//retrieving value using parameter
        this.activationKey = activationKey; //setting 'activationKey' attribute value of this class using 'this' keyword.
    }

    public int getPrice() {
        return price; //returns price
    }

    public void setPrice (int price) {//retrieving value using parameter
        this.price = price;//setting 'price' attribute value of this class using 'this' keyword.
    }

    //defining display method to display the product details stored in the object of
```

```
this class
    public void display() {
        System.out.println("Product Number: " + getProductNo());
        System.out.println("Product Name: " + getProductName());
        if(!activationKey.equals("") && price != 0)
        System.out.println("Activation Key: " + getActivationKey() + "\nPrice: " +
getPrice());
    }
}
```

2. EnterpriseEdition (Child Class)

```
/*
 * EnterpriseEdition is a sub-class of MicrosoftProduct class that inherits all properties of
it's super-class
  which has seven attributes namely; clientCompanyName, pricePerUser,
numberOfUser, activationDate, licenseExpireDate, isActivated and isExpired.
 * These attributes will store the input values passed from the GUI through object
created and passing those
   value as parameters in the constructor.
 * Accessor method are used to get the value stored in this class to other class or sub
class.
 * Author: Susan Shrestha
 */
public class EnterpriseEdition extends MicrosoftProduct {//inheriting MicrosoftProduct
```

class using 'extends' keyword

```
    //declaring attributes as private access modifier
    private String clientCompanyName;
    private int pricePerUser;
    private int numberOfUser;
    private String activationDate;
    private String licenseExpireDate;
    private boolean isActivated;
    private boolean isExpired;


    //creating constructor method of EnterpriseEdition class
    public EnterpriseEdition(int productNo, String productName, int pricePerUser) {//passing parameter values to super class and setting initial values for attributes.
        super(productNo, productName);
        this.pricePerUser = pricePerUser;
        this.numberOfUser = 0;
        this.activationDate = "";
        this.licenseExpireDate = "";
        this.isActivated = false;
        this.isExpired = false;
    }
```

```
    //accessor method for all attributes
    public String getClientCompanyName() {
        return clientCompanyName;
    }

    public int getPricePerUser() {
        return pricePerUser;
    }

    public int getNumberOfUser() {
        return numberOfUser;
    }

    public String getActivationDate() {
        return activationDate;
    }

    public String getLicenseExpireDate() {
        return licenseExpireDate;
```

```
}

public boolean getIsActivated() {
return isActivated;
}

public boolean getIsExpired() {
    return isExpired;
}

public void setPricePerUser(int pricePerUser) { //setter method to set the value of 'pricePerUser' by passing value through parameter
    if(isActivated == false) { //checks if isActivated status is true or false
        this.pricePerUser = pricePerUser; //sets pricePerUser
        System.out.println("New Price per user is set.\n"); //prints message
    }
    else
        System.out.println("License is already activated, Cannot change the Price per User.\n");   //prints message in else condition
}
```

```
    //method to Activate the License for the particular product

    public void activateLicense(String clientCompanyName, int numberOfUser, String activationDate, String licenseExpireDate, String activationKey) { //passing values using parameter through this method

        if(isActivated == true) //checks if license is activated

            System.out.println("License is already Activated! \nRegistered Company Name: " + clientCompanyName + "\nNo. of users: " + numberOfUser + "\n"); //print already activated

        else {//if license is not activated, activating it

            this.clientCompanyName = clientCompanyName;

            this.numberOfUser = numberOfUser;

            this.activationDate = activationDate;

            this.licenseExpireDate = licenseExpireDate;

            this.isActivated = true;

            this.isExpired = false;

            super.setActivationKey(activationKey);

            super.setPrice(numberOfUser * pricePerUser);

            System.out.println("Sucessfully Activated!!!"); //prints activated

        }

    }


    //method to Terminate the License for the particular product
```

```
    public void terminateLicense() {
        if(isExpired == true)//checks if license is expired
            System.out.println("This License is not activated.\n"); //prints not activated
        else { //if license is activated, terminating it by resetting the values to null
            this.clientCompanyName = "";
            this.activationDate = "";
            this.licenseExpireDate = "";
            this.numberOfUser = 0;
            this.isActivated = false;
            this.isExpired = true;
            System.out.println("This license has been sucessfully terminated.\n"); //prints terminated
        }
    }

    //defining display method to display the product details stored in the object of this class
    public void display() {
        super.display();
        if (isActivated == true) {
            System.out.println("Client company name: " + getClientCompanyName());
```

```
            System.out.println("Number of user: " + getNumberOfUser());
            System.out.println("Price per user: " + getPricePerUser());
            System.out.println("Total price: " + numberOfUser*pricePerUser);
            System.out.println("Activation date: " + getActivationDate());
            System.out.println("License expire date: " + getLicenseExpireDate());
            System.out.println("Activation Key: " + getActivationKey());
            System.out.println("Activation status: " + getIsActivated());
            System.out.println("\n");
        }
    }
}
```

3. ProductGUI

```
/*
* ProductGUI class is the main class to which MicrosoftProduct and EnterpriseEdition class are connected and their methods
and attributes will be used in this class to store GUI inputs to their attributes.
* This class will result a GUI frame that will be the User Interface in which user can interact with the forms and components for microsoft product details.
* Author: Susan Shrestha
*/
```

```
//importing all awt and swing components for GUI design
import java.awt.BorderLayout;
import java.awt.EventQueue;


import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.border.EmptyBorder;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JTextField;
import javax.swing.JButton;
import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;
import java.awt.Font;
import java.util.ArrayList;
import java.awt.Color;
import javax.swing.JMenuBar;
import java.awt.SystemColor;


public class ProductGUI extends JFrame { //extending JFrame for creating application
```

window frame

```
    private JPanel contentPane; //creating panel for adding components
    //creating textfields
    private JTextField tfProductNo;
    private JTextField tfProductName;
    private JTextField tfTotalPrice;
    private JTextField tfClientCompanyName;
    private JTextField tfLicenseActivationDate;
    private JTextField tfLicenseExpireDate;
    private JTextField tfActivationKey;
    private JTextField tfNumOfUsers;
    private JTextField tfPricePerUser;
    private JTextField tfStatus;
    private JTextField tfProductNo2;

    public static void main(String[] args) { //main method from where the execution starts
        //initializing frame and making GUI visible to the user
        EventQueue.invokeLater(new Runnable() {
            public void run() {
                try {
```

```
                ProductGUI frame = new ProductGUI();
                frame.setVisible(true);
            } catch (Exception e) {
                e.printStackTrace();
            }
        }
    });
}

/*
 * In this method, all the designs are made i.e components are added and structured to make
   the user easily understand and use the application.
 */
public ProductGUI() {
    setResizable(false);
    setTitle("Microsoft Product Details");
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    setBounds(100, 100, 635, 498);
    contentPane = new JPanel();
    contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
```

```
setContentPane(contentPane);

contentPane.setLayout(null);


JLabel lblProductNo1 = new JLabel("Product Number:");

lblProductNo1.setFont(new Font("Arial", Font.PLAIN, 13));

lblProductNo1.setBounds(20, 32, 148, 14);

contentPane.add(lblProductNo1);


tfProductNo = new JTextField();

tfProductNo.setFont(new Font("Arial", Font.PLAIN, 13));

tfProductNo.setBounds(168, 27, 236, 26);

contentPane.add(tfProductNo);

tfProductNo.setColumns(10);


JLabel lblProductName = new JLabel("Product Name:");

lblProductName.setFont(new Font("Arial", Font.PLAIN, 13));

lblProductName.setBounds(20, 71, 148, 14);

contentPane.add(lblProductName);


JLabel lblTotalPrice = new JLabel("Total Price:");
```

```
lblTotalPrice.setFont(new Font("Arial", Font.PLAIN, 13));
lblTotalPrice.setBounds(20, 399, 148, 14);
contentPane.add(lblTotalPrice);

tfProductName = new JTextField();
tfProductName.setFont(new Font("Arial", Font.PLAIN, 13));
tfProductName.setBounds(168, 66, 236, 26);
contentPane.add(tfProductName);
tfProductName.setColumns(10);

tfTotalPrice = new JTextField();
tfTotalPrice.setEditable(false);
tfTotalPrice.setFont(new Font("Arial", Font.PLAIN, 13));
tfTotalPrice.setBounds(168, 394, 236, 26);
contentPane.add(tfTotalPrice);
tfTotalPrice.setColumns(10);

JLabel lblClientCompanyName = new JLabel("Client Company Name:");
lblClientCompanyName.setFont(new Font("Arial", Font.PLAIN, 13));
lblClientCompanyName.setBounds(20, 246, 148, 21);
```

```
contentPane.add(lblClientCompanyName);


JLabel lblLicenseActivationDate = new JLabel("License Activation Date:");
lblLicenseActivationDate.setFont(new Font("Arial", Font.PLAIN, 13));
lblLicenseActivationDate.setBounds(20, 288, 148, 14);
contentPane.add(lblLicenseActivationDate);


JLabel lblLicenseExpireDate = new JLabel("License Expire Date:");
lblLicenseExpireDate.setFont(new Font("Arial", Font.PLAIN, 13));
lblLicenseExpireDate.setBounds(20, 325, 148, 14);
contentPane.add(lblLicenseExpireDate);


JLabel lblActivationKey = new JLabel("Activation Key:");
lblActivationKey.setFont(new Font("Arial", Font.PLAIN, 13));
lblActivationKey.setBounds(20, 362, 148, 14);
contentPane.add(lblActivationKey);


JLabel lblNumOfUser = new JLabel("Number of Users:");
lblNumOfUser.setFont(new Font("Arial", Font.PLAIN, 13));
lblNumOfUser.setBounds(20, 209, 148, 14);
```

```
contentPane.add(lblNumOfUser);


JLabel lblPricePerUser = new JLabel("Price/User:");
lblPricePerUser.setFont(new Font("Arial", Font.PLAIN, 13));
lblPricePerUser.setBounds(20, 108, 148, 14);
contentPane.add(lblPricePerUser);


tfClientCompanyName = new JTextField();
tfClientCompanyName.setFont(new Font("Arial", Font.PLAIN, 13));
tfClientCompanyName.setBounds(168, 244, 236, 26);
contentPane.add(tfClientCompanyName);
tfClientCompanyName.setColumns(10);


tfLicenseActivationDate = new JTextField();
tfLicenseActivationDate.setFont(new Font("Arial", Font.PLAIN, 13));
tfLicenseActivationDate.setBounds(168, 283, 236, 26);
contentPane.add(tfLicenseActivationDate);
tfLicenseActivationDate.setColumns(10);


tfLicenseExpireDate = new JTextField();
```

```
tfLicenseExpireDate.setFont(new Font("Arial", Font.PLAIN, 13));
tfLicenseExpireDate.setBounds(168, 320, 236, 26);
contentPane.add(tfLicenseExpireDate);
tfLicenseExpireDate.setColumns(10);

tfActivationKey = new JTextField();
tfActivationKey.setFont(new Font("Arial", Font.PLAIN, 13));
tfActivationKey.setBounds(168, 357, 236, 26);
contentPane.add(tfActivationKey);
tfActivationKey.setColumns(10);

tfNumOfUsers = new JTextField();
tfNumOfUsers.setFont(new Font("Arial", Font.PLAIN, 13));
tfNumOfUsers.setBounds(168, 204, 236, 26);
contentPane.add(tfNumOfUsers);
tfNumOfUsers.setColumns(10);

tfPricePerUser = new JTextField();
tfPricePerUser.setFont(new Font("Arial", Font.PLAIN, 13));
tfPricePerUser.setBounds(168, 103, 236, 26);
```

```
contentPane.add(tfPricePerUser);
tfPricePerUser.setColumns(10);

JButton btnAddProduct = new JButton("Add Product");
btnAddProduct.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        addProduct();
    }
});
btnAddProduct.setFont(new Font("Arial", Font.BOLD, 13));

btnAddProduct.setBounds(424, 32, 180, 39);
contentPane.add(btnAddProduct);

JButton btnActivateLicense = new JButton("Activate License");
btnActivateLicense.setFont(new Font("Arial", Font.BOLD, 13));
btnActivateLicense.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent arg0) {
            activate();
    }
```

```
});
btnActivateLicense.setBounds(424, 163, 180, 39);
contentPane.add(btnActivateLicense);

JButton btnTerminateLicense = new JButton("Terminate License");
btnTerminateLicense.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent arg0) {
        terminate();
    }
});
btnTerminateLicense.setFont(new Font("Arial", Font.BOLD, 13));
btnTerminateLicense.setBounds(424, 218, 180, 39);
contentPane.add(btnTerminateLicense);

JButton btnSetPricePerUser = new JButton("Set Product Price/User");
btnSetPricePerUser.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent arg0) {
        setPricePerUser();
    }
});
```

```
btnSetPricePerUser.setFont(new Font("Arial", Font.BOLD, 13));
btnSetPricePerUser.setBounds(424, 82, 180, 39);
contentPane.add(btnSetPricePerUser);

JButton btnDisplay = new JButton("Display");
btnDisplay.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        display();
    }
});
btnDisplay.setFont(new Font("Arial", Font.BOLD, 13));
btnDisplay.setBounds(424, 273, 180, 37);
contentPane.add(btnDisplay);

JButton btnClear = new JButton("Clear");
btnClear.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent arg0) {
        clear();
    }
});
```

```
btnClear.setFont(new Font("Arial", Font.BOLD, 13));
btnClear.setBounds(424, 328, 180, 37);
contentPane.add(btnClear);

JLabel lbStatus = new JLabel("Status:");
lbStatus.setFont(new Font("Arial", Font.BOLD, 12));
lbStatus.setBounds(0, 435, 49, 25);
contentPane.add(lbStatus);

tfStatus = new JTextField();
tfStatus.setFont(new Font("Arial", Font.PLAIN, 12));
tfStatus.setText("Welcome...");
tfStatus.setEditable(false);
tfStatus.setBounds(47, 437, 582, 20);
contentPane.add(tfStatus);
tfStatus.setColumns(10);

JButton btnExit = new JButton("Exit");
btnExit.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
```

```
            System.exit(0);
        }
    });

    btnExit.setForeground(Color.RED);
    btnExit.setFont(new Font("Arial", Font.BOLD, 13));
    btnExit.setBounds(424, 383, 180, 37);
    contentPane.add(btnExit);

    JLabel lblProductNo2 = new JLabel("Product Number:");
    lblProductNo2.setFont(new Font("Arial", Font.PLAIN, 13));
    lblProductNo2.setBounds(20, 177, 148, 14);
    contentPane.add(lblProductNo2);

    tfProductNo2 = new JTextField();
    tfProductNo2.setFont(new Font("Arial", Font.PLAIN, 13));
    tfProductNo2.setBounds(168, 167, 236, 26);
    contentPane.add(tfProductNo2);
    tfProductNo2.setColumns(10);
```

```
        JPanel panel = new JPanel();
        panel.setBackground(SystemColor.controlHighlight);
        panel.setBounds(10, 11, 609, 134);
        contentPane.add(panel);
    }

    //Creating Arraylist of MicrosoftProduct class
    ArrayList<MicrosoftProduct> productList =new ArrayList<>();

    //method to add product in the arraylist
    private void addProduct() {
        //storing user input in variables of type String
        String strProductNumber = tfProductNo.getText();
        String strProductName = tfProductName.getText();
        String strPricePerUser = tfPricePerUser.getText();
        int productNo,pricePerUser;
        //checking if field are empty or not
        if(strProductNumber.trim().equals("") || strProductName.trim().equals("") ||
strPricePerUser.trim().equals("")) {
            JOptionPane.showMessageDialog(contentPane,"Please fill all the forms","Error"
,0);
```

```
    }
    else {
        try {
            //converting String to integer type
            productNo = Integer.parseInt(strProductNumber);
            pricePerUser = Integer.parseInt(strPricePerUser);
        }
        //exception handling if invalid input is given by user and to continue the program instead of crashing
        catch(NumberFormatException nfe) {
            JOptionPane.showMessageDialog(contentPane,"Please check your input format and try again" ,"Error", 0);
            return;
        }
        //for-each loop for checking if the product entered by the user is already in the list or not
        for(MicrosoftProduct mp : productList) {
            if (mp instanceof EnterpriseEdition) {
                EnterpriseEdition ee = (EnterpriseEdition)mp; //class casting; storing object of superclass Microsoft product in subclass EnterpriseEdition
                if(ee.getProductNo() == productNo) {
                    JOptionPane.showMessageDialog(contentPane,"This product is already
```

```
added!", "Error" ,0);
                    return;
                }
            }
        }

        EnterpriseEdition eeList = new EnterpriseEdition(productNo,strProductName,pricePerUser);//passing user input values to constructor of EnterpriseEdition class
        productList.add(eeList);//adds input values to arraylist of MicrosoftProduct
        String eeDetails =
            "Product Number: " + productNo +
            "\n Product Name: " + strProductName +
            "\n Price Per User: " +pricePerUser;
        JOptionPane.showMessageDialog(contentPane, eeDetails, "Product sucessfully added." , (1)); //Displays product added
        tfStatus.setText("Product " + strProductName + " added to the list.");
    }
}

//method to activate the License of product that are stored in the arraylist
```

```
    private void activate() {
        //storing user input in variables of type String
        String productNumber = tfProductNo2.getText();
        String numOfUser = tfNumOfUsers.getText();
        String clientCompanyName = tfClientCompanyName.getText();
        String licenseActivationDate = tfLicenseActivationDate.getText();
        String licenseExpireDate = tfLicenseExpireDate.getText();
        String activationKey = tfActivationKey.getText();
        int productNo,noOfUser;
        //checking if field are empty or not
        if(productNumber.trim().equals("") || numOfUser.trim().equals("") ||
clientCompanyName.trim().equals("") || licenseActivationDate.trim().equals("")||
        licenseExpireDate.trim().equals("") || activationKey.trim().equals("")) {
            JOptionPane.showMessageDialog(contentPane,"Please fill all the
fields.","Error",0);
        }
        else {
            try {
                //converting String to integer type
                productNo = Integer.parseInt(productNumber);
                noOfUser = Integer.parseInt(numOfUser);
```

```
        }
        catch(NumberFormatException nfe) {
            JOptionPane.showMessageDialog(contentPane,"Please check your input format and try again.", "Error",0);
            return;
        }
        boolean flag = false;
        for(int i = 0; i < productList.size(); ++i) { //loop continues till the size of arraylist
            MicrosoftProduct mp = productList.get(i);
            if(mp instanceof EnterpriseEdition) {
                EnterpriseEdition ee = (EnterpriseEdition)mp; //class casting
                if(ee.getProductNo() == productNo) { //checking if the entered product number is already in the list or not
                    flag= true;
                    if(ee.getIsActivated() == true) { //checking if the entered product number is already activated or not
                        JOptionPane.showMessageDialog(contentPane,"The product is already activated.","info!",2);
                    }
                    /*if the product number is not activated, it will activate license by calling the activateLicense method of class EnterpriseEdition
                    and passing the user input details as parameter in activateLicense
```

```
method.
                    */
                    else {
                        ee.activateLicense(clientCompanyName,noOfUser,
licenseActivationDate, licenseExpireDate, activationKey);
                        JOptionPane.showMessageDialog(contentPane,"Product activated
sucessfully.","Congratulations",1);
                        int totalPrice = ee.getPrice(); //retrieving total price of the product from
class EnterpriseEdition
                        String strTP = String.valueOf(totalPrice);
                        tfTotalPrice.setText(strTP);
                        tfStatus.setText("Product activation done.");
                        break;
                    }
                }
            }
        }
        if(flag == false) { //if entered product not in the list
            JOptionPane.showMessageDialog(contentPane,"Product is not added, please
add the product first.", "info" ,1);
        }
    }
```

```
    }


    //method to terminate the product License
    private void terminate() {
        int productNo;
        //asks user to input product number to terminate the license
        String strProductNo = JOptionPane.showInputDialog(contentPane, "Enter the Product Number to terminate the license.","Terminate License",3);
        try {
            if(strProductNo.trim().equals("")) {
                JOptionPane.showMessageDialog(contentPane,"Please Enter the Product Number.","info",1);
                return;
            }
        }
        catch(NullPointerException npe) {
            return;
        }


        try{
            productNo = Integer.parseInt(strProductNo);//converting string to integer type
```

```
        }
        catch(NumberFormatException nfe) {
            JOptionPane.showMessageDialog(contentPane,"Please enter a valid product number.","Error",0);
            return;
        }
        boolean active = false;
        for(MicrosoftProduct mp : productList) {
            if(mp instanceof EnterpriseEdition) {
                EnterpriseEdition ee = (EnterpriseEdition)mp; //class casting so as to access all objects of class MicrosoftProduct and EnterpriseEdition
                if(mp.getProductNo() == productNo) { //checking the entered product number is already in the list or not
                    active = true;
                    ee.terminateLicense(); //calls the terminateLicense method from EnterpriseEdition class
                    JOptionPane.showMessageDialog(contentPane,"The Product License has been terminated successfully.","Terminated",1);
                    tfStatus.setText("Product No. " + productNo + " license terminated.");
                    break;
                }
                if(active == false) //if product number not in the list
```

```
                JOptionPane.showMessageDialog(contentPane,"Product Number not found.","Error",0);
            }
        }
    }

    //method to set the price per user of the given product
    private void setPricePerUser() {
        String strProductNo = tfProductNo.getText();
        String strPricePerUser = tfPricePerUser.getText();
        int productNo, pricePerUser;
        //try catch for exception handling
        try {
            if(strProductNo.trim().equals("") || strPricePerUser.trim().equals("") || strPricePerUser.trim().equals("0")) { //checks if null value is passed
                JOptionPane.showMessageDialog(contentPane,"Please Enter the Product Number and Price per User correctly.","Error",0);
                return;
            }
        }
        catch(NullPointerException npe) {
```

```
            return;
        }
        try {
            //converting String to integer type
            productNo = Integer.parseInt(strProductNo);
            pricePerUser = Integer.parseInt(strPricePerUser);
        }
        catch(NumberFormatException nfe) {//displays error message if valid input is not provided
            JOptionPane.showMessageDialog(contentPane,"Please enter valid Product Number and Price per User.","Error",0);
            return;
        }
        boolean flag = false;
        for(MicrosoftProduct mp : productList) { //for-each loop for accessing
            EnterpriseEdition ee = (EnterpriseEdition)mp; //class casting
            if(mp.getProductNo() == productNo) { //compares provided product number and product number in arraylist then moves to next line
                flag = true;
                if(ee.getPricePerUser() == pricePerUser) { //checks if user input same price per user
                    JOptionPane.showMessageDialog(contentPane,"You have entered the
```

```
same Price per User, please try different price per user.","Warning!",2); //if same price then displays message

            }else if(ee.getIsActivated() == true) { //checks if license is activated, if already activated displays message cannot set new price per user

                JOptionPane.showMessageDialog(contentPane,"License is already activated for this product, cannot change the price per user for the product.","Warning!",2);

            }

            else {

                ee.setPricePerUser(pricePerUser); //calls setPricePerUser() method from EnterpriseEdition class

                tfStatus.setText("New price per user set for the product no: " + productNo + ".");

                break;

            }

        }

        if(flag == false) {

            JOptionPane.showMessageDialog(contentPane,"Product not found, please try again.","error",0);

        }

    }

}
```

```
    //method to display arraylist in terminal
    private void display() {
        boolean flag = false;
        for(MicrosoftProduct mp : productList) {
            if(mp instanceof EnterpriseEdition)  {
                EnterpriseEdition ee= (EnterpriseEdition)mp; //class casting
                flag = true;
                ee.display(); //calls display() method of EnterpriseEdition class that overrides display() method of MicrosoftProduct class
                tfStatus.setText("Product list details displayed in terminal."); //arraylist displayed
            }
        }

        if(flag == false) {
            JOptionPane.showMessageDialog(contentPane,"No product added, please add the product first.","info",1);
        }
    }

    //method to clear form
```

```
    private void clear() {
        int opt = JOptionPane.showConfirmDialog(contentPane,"Do you want to clear all fields?","clear form",JOptionPane.YES_NO_OPTION);//asks user to clear the form
        if(opt == JOptionPane.YES_OPTION) { //if yes is clicked, clears form else return
            tfProductNo.setText(null);
            tfProductName.setText(null);
            tfPricePerUser.setText(null);
            tfNumOfUsers.setText(null);
            tfClientCompanyName.setText(null);
            tfLicenseActivationDate.setText(null);
            tfLicenseExpireDate.setText(null);
            tfActivationKey.setText(null);
            tfTotalPrice.setText(null);
            tfProductNo2.setText(null);
            tfStatus.setText("Form cleared.");
        }
    }
}
```