Software Design & Architecture

Assignment#2

Architectural Patterns and Design Drivers

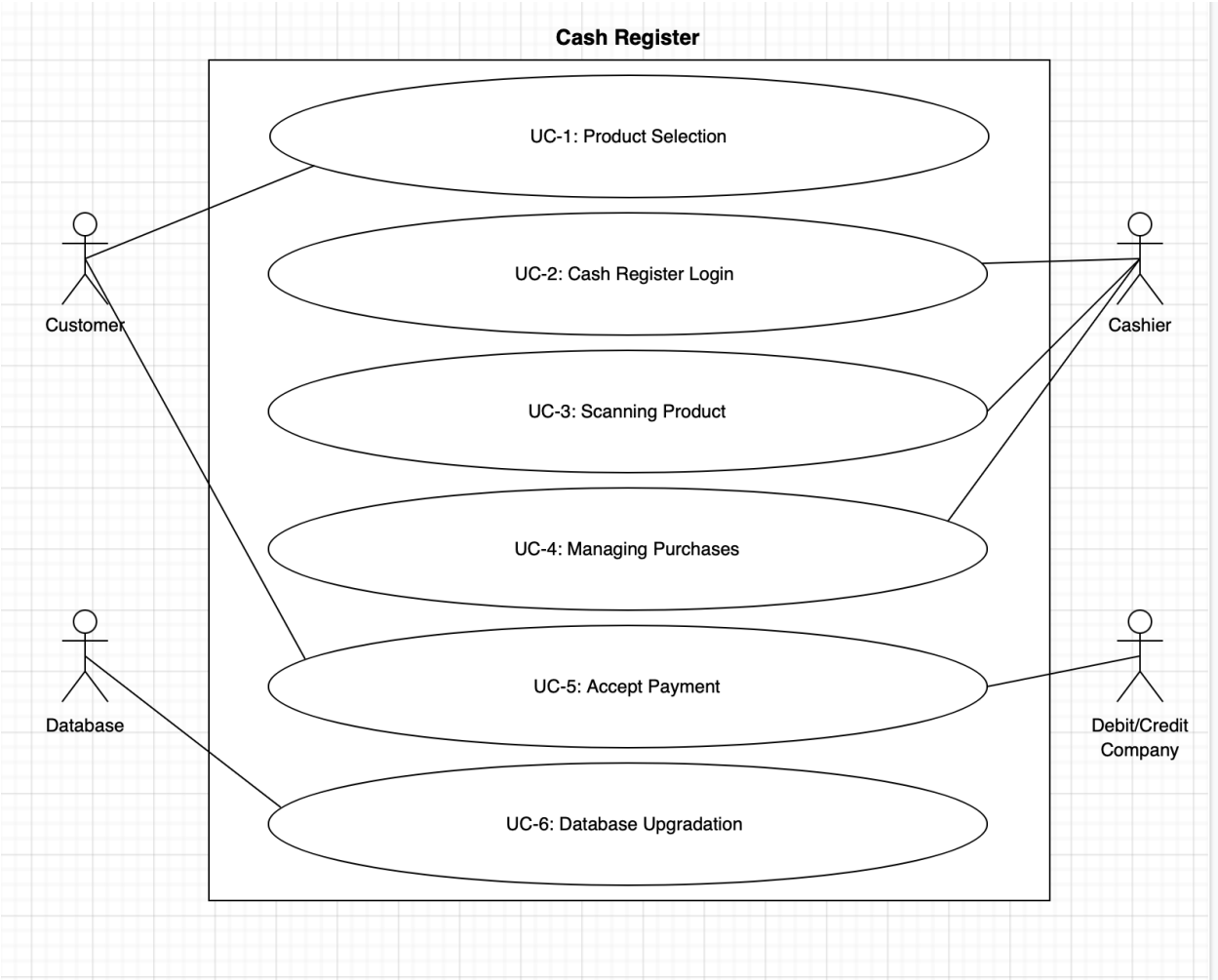
Due - October 26, 2022

Group 30

Aqsa Asif
Meet Patel
Shahab Zafar

Exercise Part 1

Use Case Diagram



Use Case

ID	Description
UC-1: Product Selection	Customer is able to select any product of his/her choice and bring it to the counter to be scanned by the cashier.
UC-2: Cash Register Login	Prior to any items being scanned by the barcode scanner, a Cashier initiates a product purchase session using the keyboard.
UC-3: Scanning Product	The cashier now has all of the items ready to be scanned by the barcode reader in order to get product information such as name and price from the database and issue a ticket for the customer.
UC-4: Managing Purchases	A product's name and price are displayed on a cash register screen once it has been recognised using a barcode scanner. If a product's barcode cannot be read by the barcode scanner, the message "Unknown product" will appear, and the barcode can be typed using the Cashier's keyboard.
UC-5: Accept Payment	When all of the products have been read, the Cashier can choose a payment method using the keyboard, such as cash, debit, or credit card. (For Debit and Credit Card payments, a third-party payment system is used to execute the transaction.)
UC-6: Database Upgradation	Cash register contains a local database for products. Everytime a product is purchased or the price is changed, the information gets upgraded in the database.

Quality Attributes

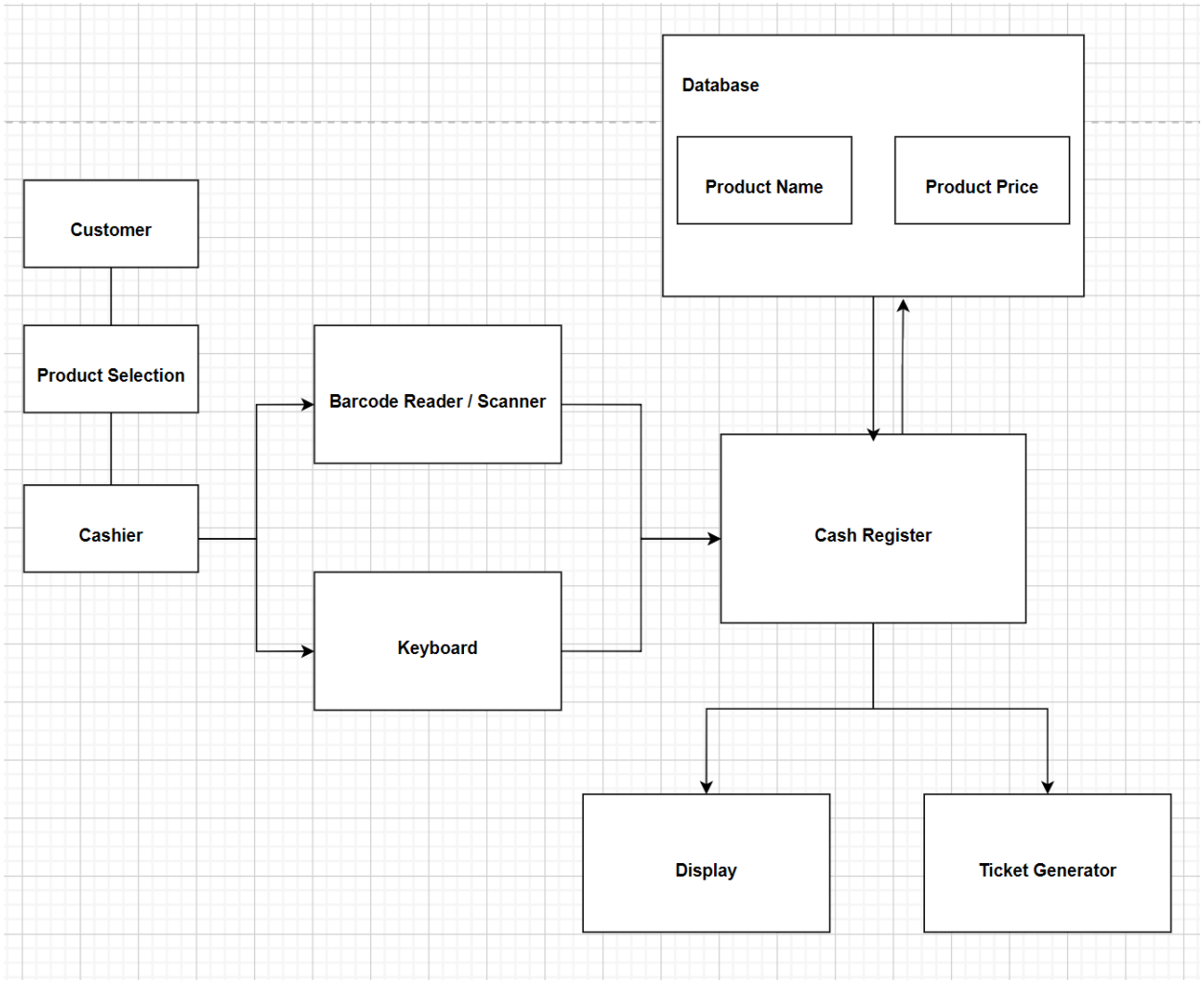
ID	Quality Attributes	Scenario	Associated Use Case
QA-1	Functionality	The system is fast and straightforward right from when the cashier is logged into the system.	All Cases
QA-2	User Friendliness/ Simplicity	The register will be easy to use and operate for all cashiers including new starters.	UC-2 UC-3 UC-4
QA-3	Connectivity	The system will always have a strong connection with the database and the cash register, if in any case the system gets disconnected the register will still be able to withhold all the data locally to finish that particular session.	UC-2 UC-6
QA-4	Accessibility	The system will display all the options possible inorder to deal with the product purchasing process. In case of any lag or glitch there will be a help navigation bar.	All Cases
QA-5	Security	The database is secured tightly to ensure that there is no malfunction associated with the product details. It also ensures that there is no leakage of customer payment	All Cases

		info whatsoever.	

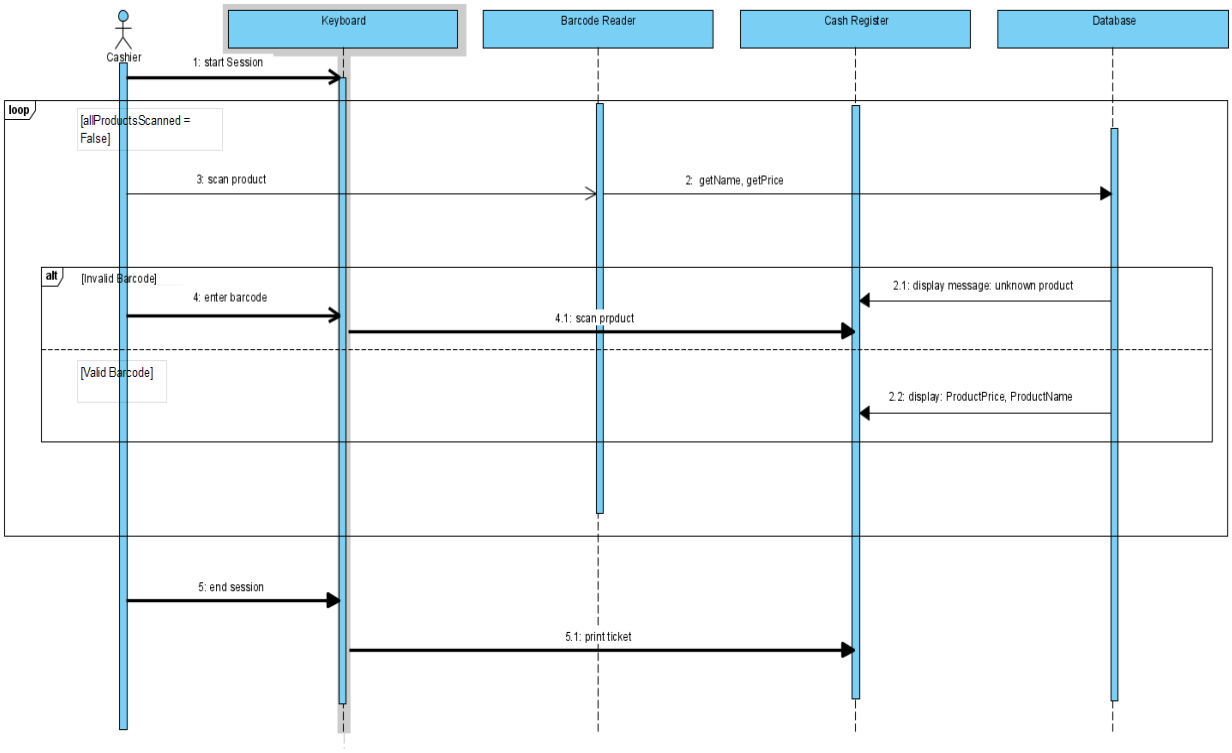
System Constraints

ID	Constraints
CON-1	Only one customer can be part of the purchase at one time.
CON-2	Barcode reader may not be able to read the barcode in situations where barcode on the product is even slightly messed up, there's low lighting etc
CON-3	Managing large volumes of data input may cause glitching in the cash register system, however the data should not be lost in any case.
CON-4	The network connection between the system and database needs to be reliable all the time in order for the transactions to go smoothly.
CON-5	There must be a backup server for all the information

Exercise Part 2



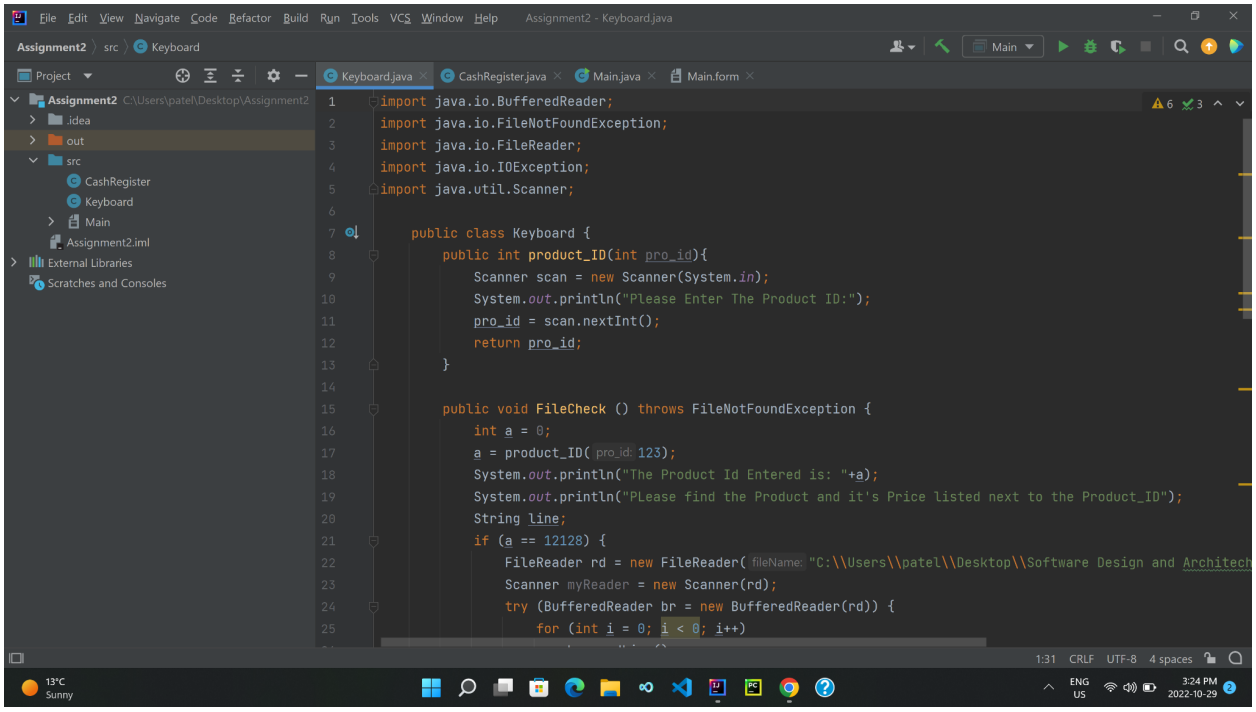
Exercise Part 3



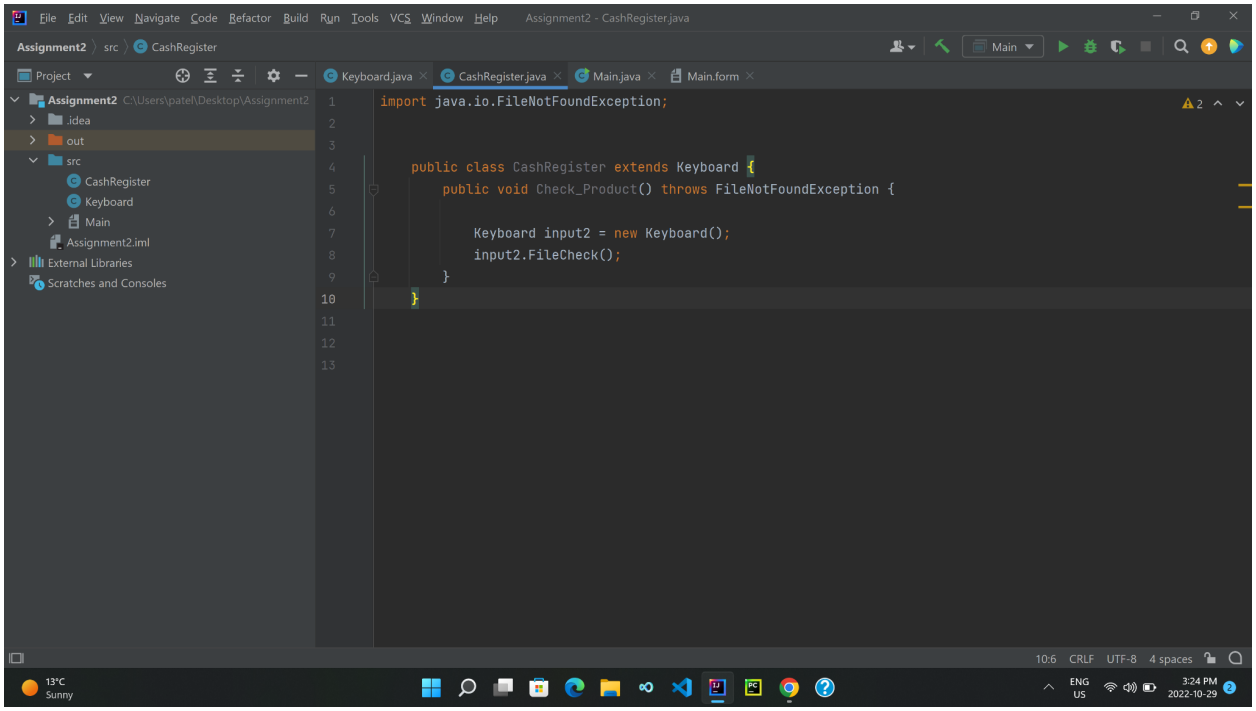
(Please Scroll Below)

Exercise Part 4.1

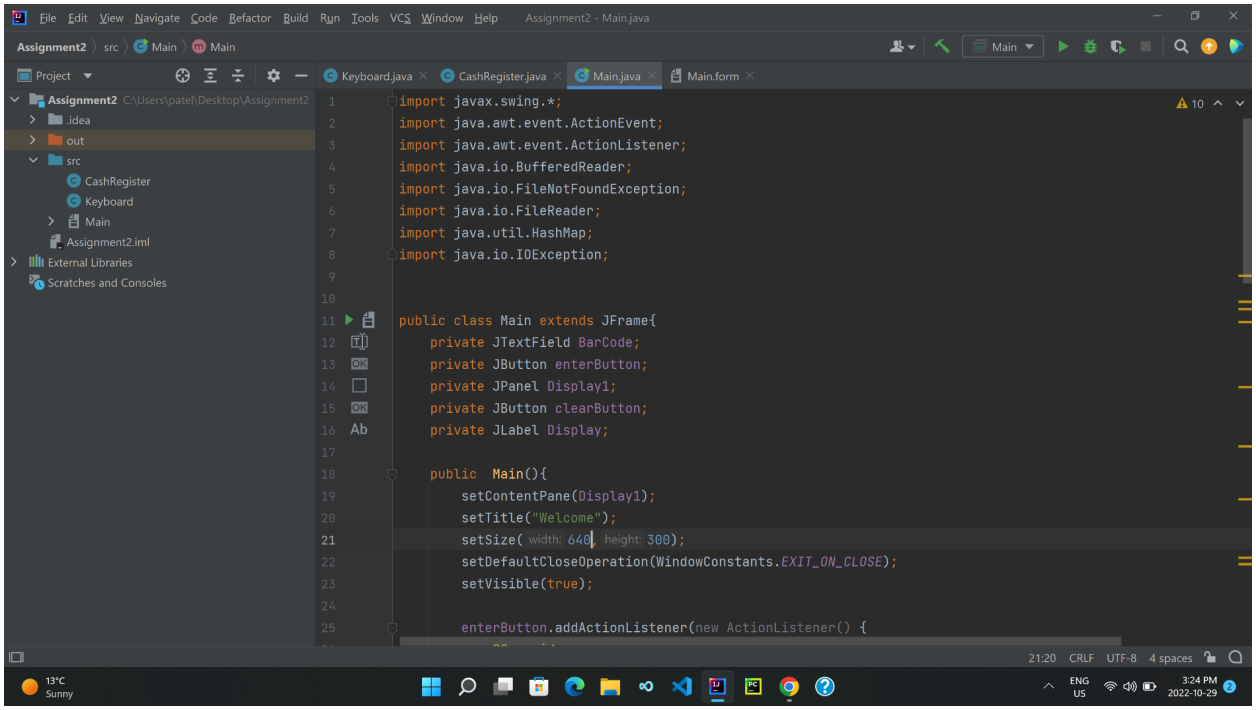
Screenshots for the Code Attached Below:



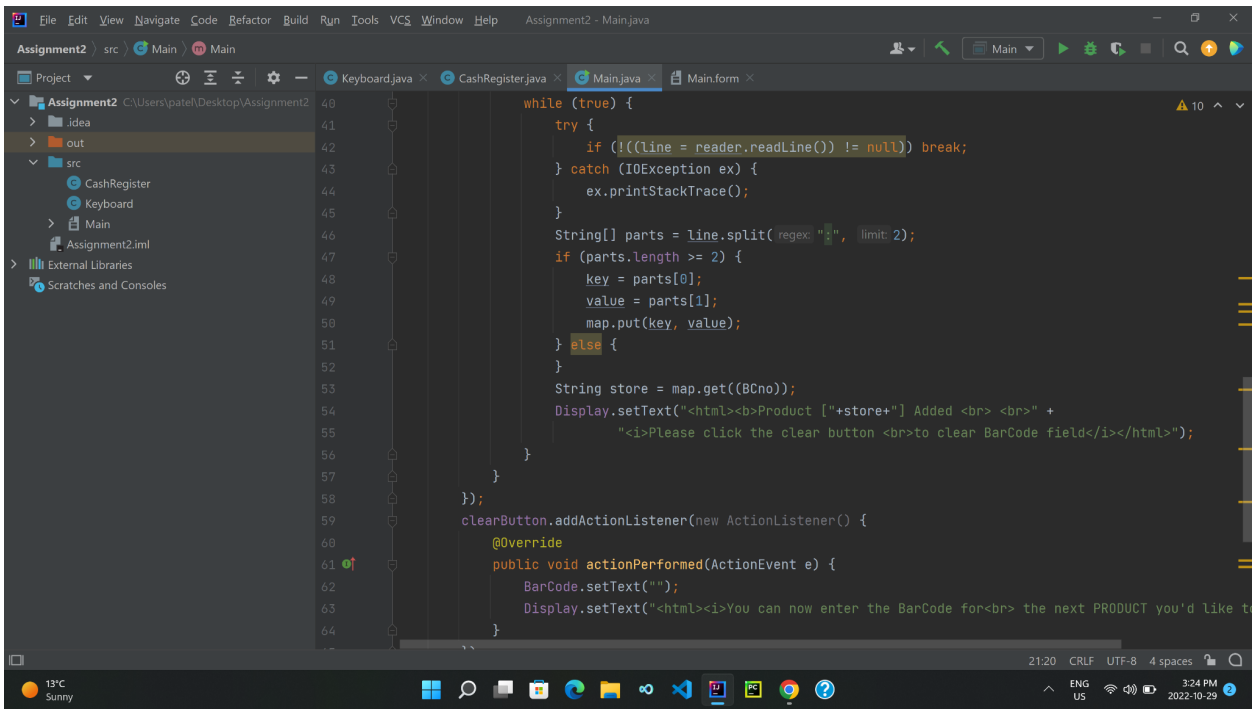
```
import java.io.BufferedReader;
import java.io.FileNotFoundException;
import java.io.FileReader;
import java.io.IOException;
import java.util.Scanner;

    public class Keyboard {
        public int product_ID(int pro_id){
            Scanner scan = new Scanner(System.in);
            System.out.println("Please Enter The Product ID:");
            pro_id = scan.nextInt();
            return pro_id;
        }

        public void FileCheck () throws FileNotFoundException {
            int a = 0;
            a = product_ID( pro_id: 123);
            System.out.println("The Product Id Entered is: "+a);
            System.out.println("PLease find the Product and it's Price listed next to the Product_ID");
            String line;
            if (a == 12128) {
                FileReader rd = new FileReader( fileName: "C:\\Users\\patel\\Desktop\\Software Design and Architech
                Scanner myReader = new Scanner(rd);
                try (BufferedReader br = new BufferedReader(rd)) {
                    for (int i = 0; i < 0; i++)
```



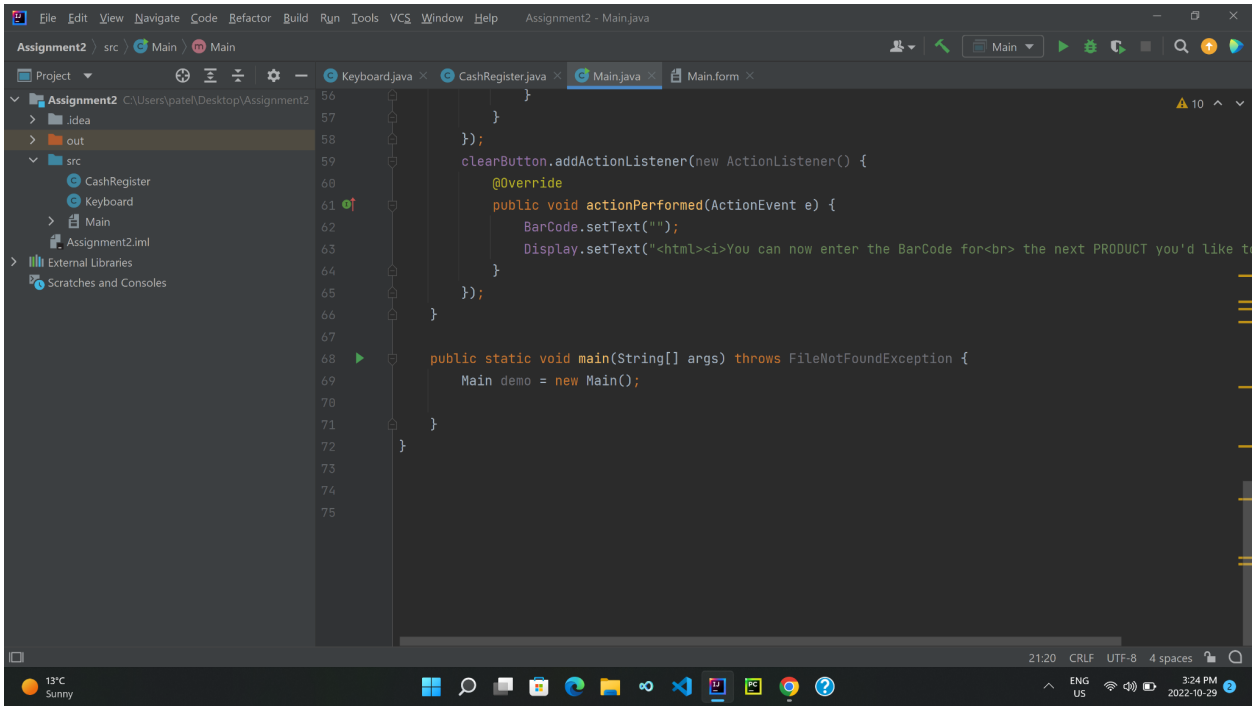
```
import java.io.FileNotFoundException;


    public class CashRegister extends Keyboard {
        public void Check_Product() throws FileNotFoundException {

            Keyboard input2 = new Keyboard();
            input2.FileCheck();
        }
    }
```

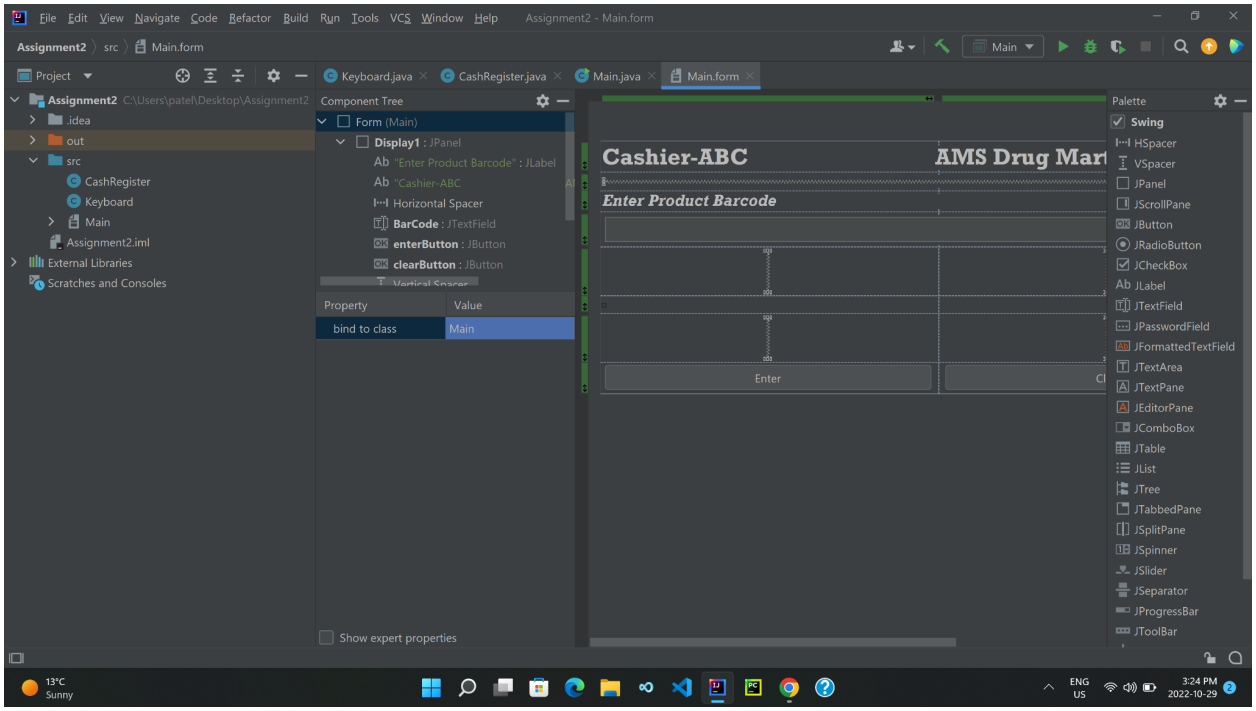


```
import javax.swing.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.io.BufferedReader;
import java.io.FileNotFoundException;
import java.io.FileReader;
import java.util.HashMap;
import java.io.IOException;


public class Main extends JFrame{
    private JTextField BarCode;
    private JButton enterButton;
    private JPanel Display1;
    private JButton clearButton;
    private JLabel Display;

    public  Main(){
        setContentPane(Display1);
        setTitle("Welcome");
        setSize( width: 640, height: 300);
        setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
        setVisible(true);

        enterButton.addActionListener(new ActionListener() {
```

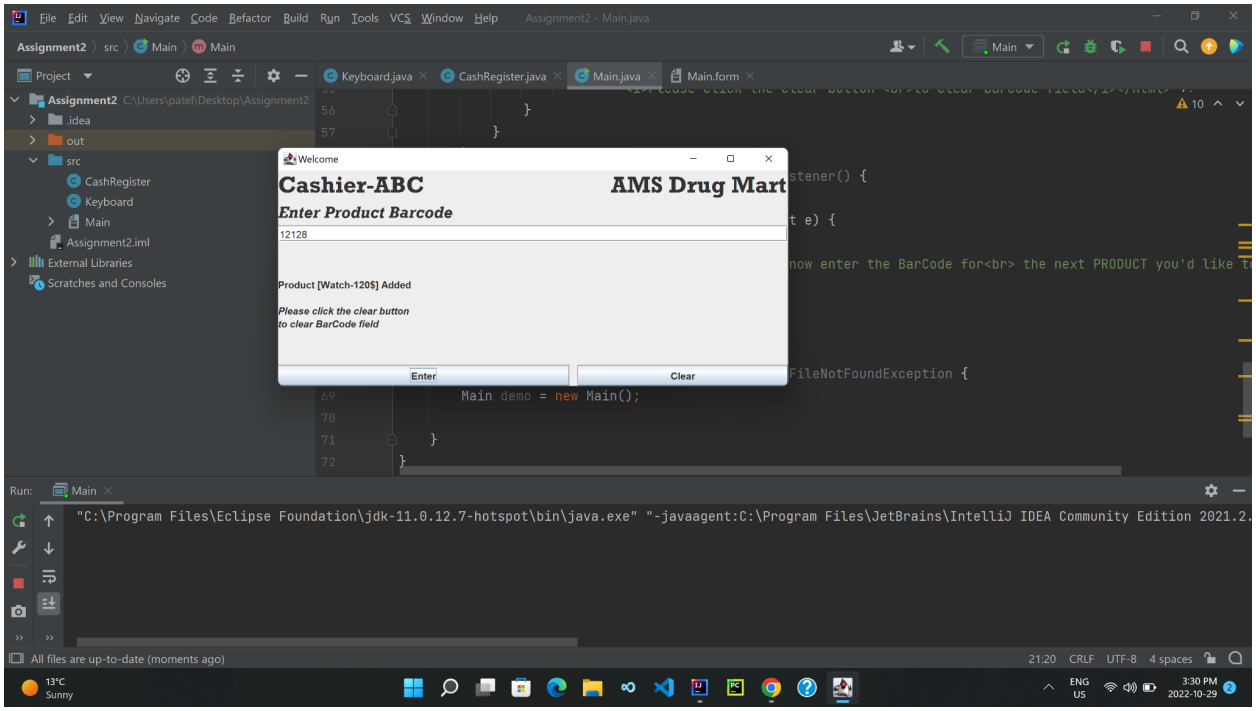


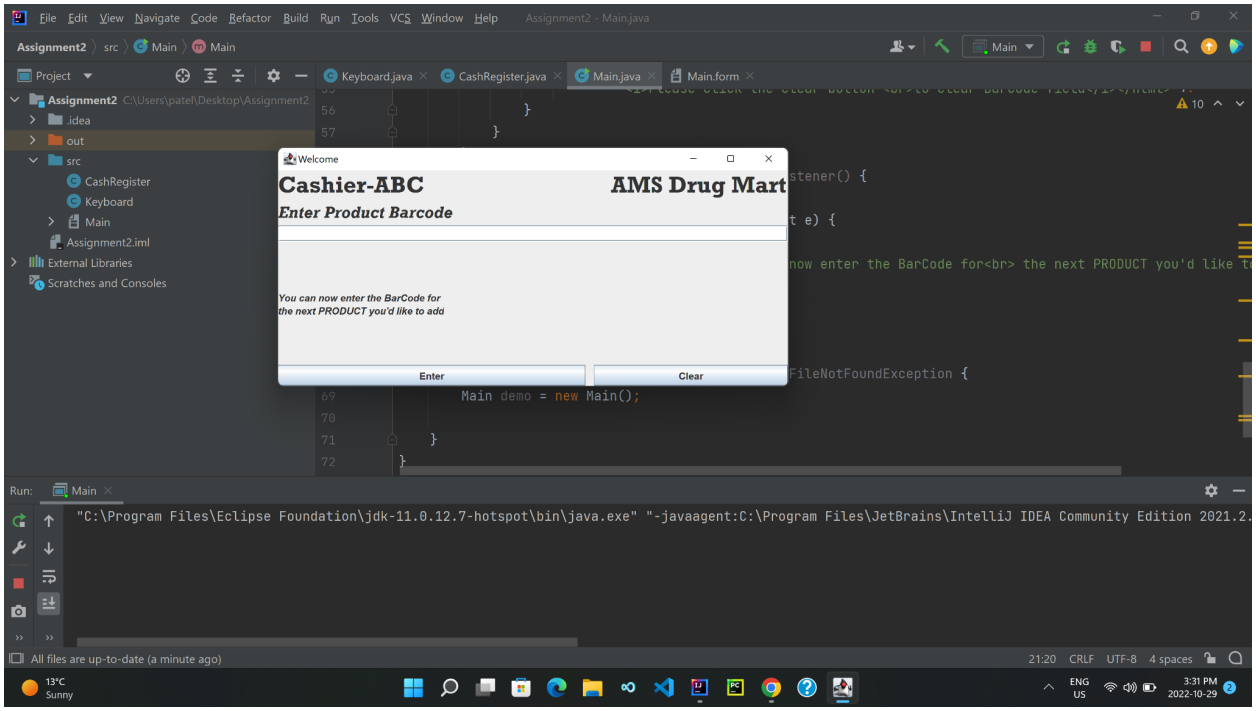
```
                while (true) {
                    try {
                        if (!((line = reader.readLine()) != null)) break;
                    } catch (IOException ex) {
                        ex.printStackTrace();
                    }
                    String[] parts = line.split( regex: ":", limit: 2);
                    if (parts.length >= 2) {
                        key = parts[0];
                        value = parts[1];
                        map.put(key, value);
                    } else {
                    }
                    String store = map.get((BCno));
                    Display.setText("<html><b>Product ["+store+"] Added <br> <br>" +
                            "<i>Please click the clear button <br>to clear BarCode field</i></html>");
                }
            }
        });
        clearButton.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                BarCode.setText("");
                Display.setText("<html><i>You can now enter the BarCode for<br> the next PRODUCT you'd like t
            }
```

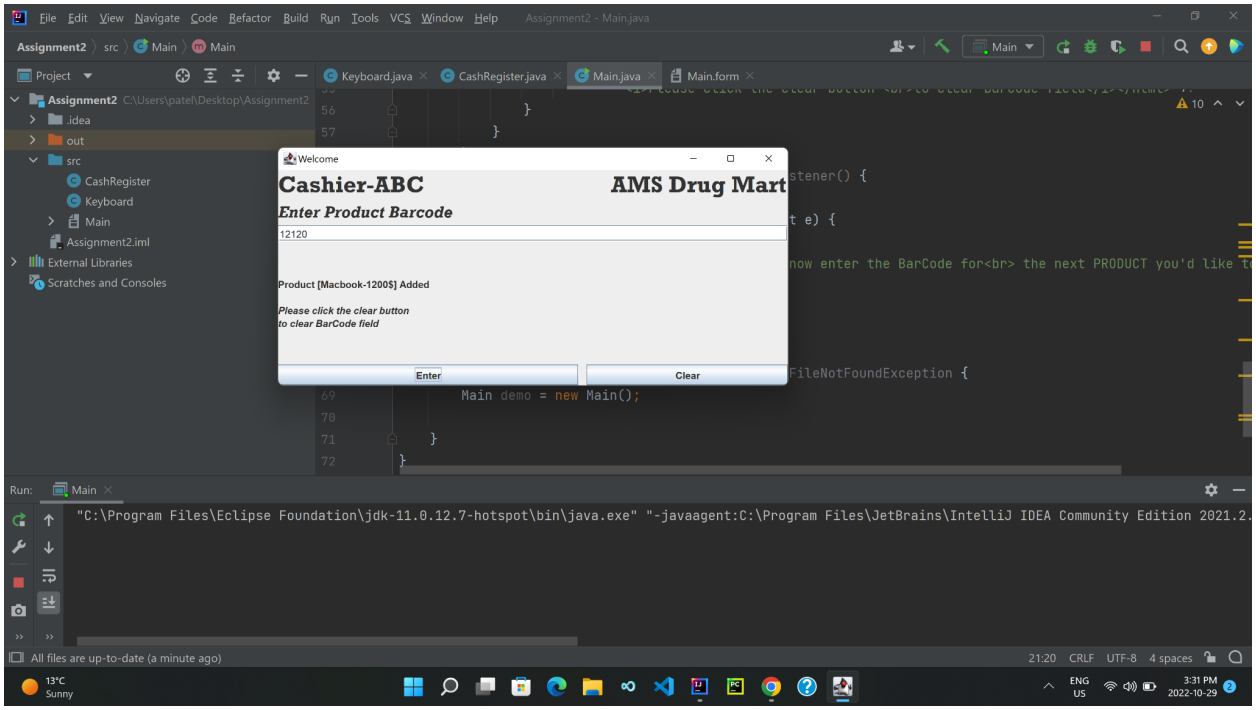

File Edit View Navigate Code Refactor Build Run Tools VCS Window Help
Assignment2 - Main.java
Assignment2 src Main Main
Main
Project
Keyboard.java
CashRegister.java
Main.java
Main.form
Assignment2 C:\Users\patel\Desktop\Assignment2
.idea
out
src
CashRegister
Keyboard
Main
Assignment2.iml
External Libraries
Scratches and Consoles
clearButton.addActionListener(new ActionListener() {
@Override
public void actionPerformed(ActionEvent e) {
BarCode.setText("");
Display.setText("<html><i>You can now enter the BarCode for
 the next PRODUCT you'd like t
public static void main(String[] args) throws FileNotFoundException {
Main demo = new Main();
21:20 CRLF UTF-8 4 spaces
13°C Sunny
ENG US
3:24 PM 2022-10-29


File Edit View Navigate Code Refactor Build Run Tools VCS Window Help
Assignment2 - Main.form
Assignment2 src Main.form
Main
Project
Keyboard.java
CashRegister.java
Main.java
Main.form
Component Tree
Form (Main)
Display1 : JPanel
"Enter Product Barcode" : JLabel
"Cashier-ABC"
Horizontal Spacer
BarCode : JTextField
enterButton : JButton
clearButton : JButton
Property
Value
bind to class
Main
Show expert properties
Cashier-ABC
AMS Drug Mart
Enter Product Barcode
Enter
Palette
Swing
HSpacer
VSpacer
JPanel
JScrollPane
JButton
JRadioButton
JCheckBox
JLabel
JTextField
JPasswordField
JFormattedTextField
JTextArea
JTextPane
JEditorPane
JComboBox
JTable
JList
JTree
JTabbedPane
JSplitPane
JSpinner
JSlider
JSeparator
JProgressBar
JToolBar
13°C Sunny
ENG US
3:24 PM 2022-10-29

Screenshots for the GUI:







Exercise Part 4.2

Instructions on how to execute GUI and how it works:

When you run the programme, the GUI panel appears; in this example the use case selected is “the cashier is already logged into his account of operation and ready to begin a session with a customer”. The top of the panel also indicates which cashier is logged into the system to keep track of who completed a specific session.

There is a prompt on the panel to "Enter product barcode," which instructs the cashier to enter the product's barcode number in order to add it to the customer's bill and generate a ticket for the total purchase. Once the barcode is entered and the cashier clicks enter, the panel prompts them with the product name and price associated with the barcode and states that it was successfully added. They can now click the clear button to clear the barcode filled and the panel is ready to accept the new barcode for the next product that needs to be added.

Also, the GUI was designed so that once launched, it runs indefinitely, allowing the cashier to enter any number of products without having to restart the programme after each addition.