



รายงาน

เรื่อง โปรแกรม POS Restaurant

จัดทำโดย

ชื่อ นาย อัครเดช วิทยาอุฒธิรัตน์ รหัส 633021009-4

ชื่อ นางสาว รัตนาพร สวนดอกไม้ รหัส 633020559-4

ชื่อ นางสาว พิชญ์สร้อย วรงค์คำจันทร์ รหัส 633020567-5

เสนอ

รศ.ดร. ปัญญาพล หอระตะ

รายงานนี้เป็นส่วนหนึ่งของรายวิชา SC311003 การเขียนโปรแกรมเชิงวัตถุ

(Object – Oriented Programming)

สาขาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ มหาวิทยาลัยขอนแก่น

คำนำ

รายงานเล่มนี้จัดทำขึ้นเพื่อเป็นส่วนหนึ่งของรายวิชา SC311003 การเขียนโปรแกรมเชิงวัตถุ (Object – Oriented Programming) สาขาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ มหาวิทยาลัยขอนแก่น เพื่อให้ได้ศึกษาหาความรู้ในเรื่องการประยุกต์จากการใช้งานภาษา JAVA โดยใช้โปรแกรม Android Studio และได้ศึกษาอย่างเข้าใจเพื่อเป็นประโยชน์กับการเรียน

ผู้จัดทำหวังว่า รายงานเล่มนี้จะเป็นประโยชน์กับผู้อ่าน นักเรียน หรือนักศึกษา ที่กำลังหาข้อมูลเรื่องนี้ อยู่ หากมีข้อเสนอแนะหรือข้อผิดพลาดประการใด ผู้จัดทำขอน้อมรับไว้และขออภัยมา ณ ที่นี้ด้วย

คณะจัดทำ

20 เมษายน 2564

สารบัญ

หน้า

คำนำ

ก

สารบัญ

๗

บทนำ

- ที่มาและความสำคัญ

1

- วัตถุประสงค์

1

- ขอบเขตของรายงาน

1

- ผลที่คาดว่าจะได้รับ

1

ขั้นตอนการดำเนินงาน

- แผนภาพคลาส

2

- Code การทำงานของโปรแกรม

5-8

ผลการดำเนินงาน

- หน้าต่างแสดงผลการทำงานของแอปพลิเคชัน

9-10

- ตัวอย่างผลลัพธ์การทำงานของแอปพลิเคชัน

11

อ้างอิง

บทนำ

ที่มาและความสำคัญ

ในยุคปัจจุบันนี้เป็นยุคของเทคโนโลยี/ดิจิทัล คณะผู้จัดทำมีการสนใจในด้านการค้า เช่น การคิดเงินแบบแคชเชียร์ในร้านค้า สามารถพิมพ์รหัสโค้ดและสแกนบาร์โค้ด จะได้ผลลัพธ์แบบที่เราต้องการ โดยมีการใช้การเขียนโปรแกรมโดยใช้ภาษา Java และใช้โปรแกรม Netbean เป็นตัวช่วยในการเขียนเพื่อให้ได้โปรแกรมที่เราต้องการ

ดังนั้น คณะผู้จัดทำจึงได้มีการวางแผนในการเขียนการคิดเงินแบบแคชเชียร์ โดยมีการใช้การเขียนโปรแกรมโดยใช้ภาษา Java และใช้โปรแกรม Netbean เป็นตัวช่วยในการเขียนโปรแกรม

วัตถุประสงค์

1. เพื่อศึกษาเกี่ยวกับการประยุกต์การใช้งานภาษา JAVA โดยใช้โปรแกรม Netbean ในการสร้างแอปพลิเคชัน
2. เพื่อสร้างแอปพลิเคชันคิดเงินแบบแคชเชียร์ ที่เขียนด้วยภาษา Java จากโปรแกรม Netbean

ขอบเขตของรายงาน

จัดทำรายงานเล่มนี้เพื่อให้คนทั่วไปได้มาศึกษา การประยุกต์การใช้งานภาษา JAVA โดยใช้โปรแกรม Netbean เป็นตัวช่วยในการเขียนโปรแกรม

ผลที่คาดว่าจะได้รับ

1. สามารถเข้าใจการใช้งานภาษาจาวาบนโปรแกรม Netbean ได้อย่างเข้าใจ และนำไปพัฒนาในการใช้งานต่อไปในอนาคต
2. ได้แอปพลิเคชันคิดเงินแบบแคชเชียร์ ที่เขียนด้วยภาษา Java จากโปรแกรม Netbean

ขั้นตอนการดำเนินการ

แผนภาพคลาส

```

SaleUI
ArrayList<String> barcode = new ArrayList<String>();
ArrayList<String> name = new ArrayList<String>();
ArrayList<String> table = new ArrayList<String>();
ArrayList<Double> price = new ArrayList<Double>();
ArrayList<String> dateTime = new ArrayList<String>();
ArrayList<String> totalTable= new ArrayList<String>();
PreparedStatement ppsm ;
    LocalDateTime myDateObj01 = LocalDateTime.now();
ResultSet resultSetSum ;

SaleUI : constructor
MAIN
private void scanFieldKeyPressed(java.awt.event.KeyEvent evt)
private void finishButtonMouseClicked(java.awt.event.MouseEvent evt)
private void OKButtonMouseClicked(java.awt.event.MouseEvent evt)
private void AddButton01ActionPerformed(java.awt.event.ActionEvent evt)
private void removeListButtonMouseClicked(java.awt.event.MouseEvent evt)
private void Remove02ButtonMouseClicked(java.awt.event.MouseEvent evt)
private void summaryButtonMouseClicked(java.awt.event.MouseEvent evt)

```



```

JavaDatabase
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;

public void connect()

```

Code การทำงานของโปรแกรม

```

*/
public class JavaDatabase {
    Connection conn;

    public JavaDatabase(){
        try{
            conn=DriverManager.getConnection("jdbc:mysql://localhost:3306/foodmenu","root","root");
            PreparedStatement pstmt = conn.prepareStatement("SELECT * FROM FOOD_MENU");
            ResultSet resultSet = pstmt.executeQuery();

        }catch(Exception e)
        {
            e.printStackTrace();
        }
    }
    //Connection conn;
    public void connect(){
        try{
            conn=DriverManager.getConnection("jdbc:mysql://localhost:3306/foodmenu","root","root");
            PreparedStatement pstmt = conn.prepareStatement("SELECT * FROM FOOD_MENU");
            ResultSet resultSet = pstmt.executeQuery();

        }catch(Exception e)
        {
            e.printStackTrace();
        }
    }
}

```

ส่วนนี้คือส่วนของ คลาส JavaDatabase สร้างขึ้นมาเพื่อทำการเชื่อมต่อกับ Database ของ MySQL โดยมีคอนสแตนต์ ทำการเรียกข้อมูลจากฐานข้อมูลที่อยู่ในเครื่อง (localhost) โดยมีการจัดเตรียมข้อมูลโดยใช้ PreparedStatement เรียกข้อมูลจากตารางใน MySQL ที่มีชื่อว่า FOOD_MENU จากฐานข้อมูลที่มีชื่อว่า foodmenu

```

public SaleUI() throws SQLException {
    initComponents();

    DB.connect();
    ResultSet resultSet = DB.conn.prepareStatement("SELECT * FROM FOOD_MENU").executeQuery();

    while(resultSet.next()){
        // System.out.println(resultSet.getString("barcode"));
        barcode.add(resultSet.getString("barcode"));
        name.add(resultSet.getString("foodName"));
        price.add(resultSet.getDouble("price"));
    }

    LocalDateTime myDateObj = LocalDateTime.now();
    DateTimeFormatter myFormatObj = DateTimeFormatter.ofPattern("dd-MM-yyyy");
    String formattedDate = myDateObj.format(myFormatObj);
    System.out.println("After formatting: " + formattedDate);

    time.setText(formattedDate);
}

```

ส่วนนี้คือส่วนของคลาส SaleUI ซึ่งเป็นส่วนหลักของโปรแกรมนี้อยู่ โดย จากภาพคือส่วนคอนสตรัคเตอร์ ของ SaleUI ที่จะถูกเรียกใช้งานทันทีที่เริ่มรันโปรแกรม โดยข้างในจะประกอบด้วย สร้างวัตถุเพื่อเชื่อมต่อกับ ฐานข้อมูล MySQL และเรียกข้อมูลมาเตรียมใช้ โดยเรียกข้อมูลแต่ละประเภท (barcode, name, price) มาแล้วเพิ่มไปใน ArrayList เพื่อง่ายต่อการเรียกใช้ และเพิ่มวัตถุ LocalDateTime มาไว้ในหน้า UI

```

private void scanFieldKeyPressed(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
    doubleCheck = false;
    if(evt.getKeyCode()==KeyEvent.VK_ENTER){
        System.out.println(scanField.getText());
        i = 0;
        //Check Condition if order is already in table
        for(String check:table){
            if(scanField.getText().equalsIgnoreCase(check)){
                int count = (int)table1.getValueAt(table.indexOf(check), 2);
                count++;
                table1.setValueAt(count,table.indexOf(check), 2);

                Double subTotal = 0.0;
                subTotal = subTotal + count * (double)table1.getValueAt(table.indexOf(check), 3);
                table1.setValueAt(subTotal,table.indexOf(check), 4);
                total += (double)table1.getValueAt(table.indexOf(check), 3);
                doubleCheck = true;
            }
        }
        //Check Condition if order is not already in table
        if(doubleCheck == false){
            for(String check:barcode){
                //System.out.println(check);
                if(scanField.getText().equalsIgnoreCase(check)){
                    table.add(scanField.getText());
                    table1.setValueAt(scanField.getText(), row,0 );
                    table1.setValueAt(name.get(i), row, 1);
                    table1.setValueAt(qty, row, 2);
                    table1.setValueAt(price.get(i), row, 3);
                    table1.setValueAt(price.get(i), row, 4);
                    total += price.get(i);
                    row++;
                    System.out.println(name);
                    i=0;
                }
            }
        }
        else{
            //NOT FOUND PRODUCT
            i = i+1;
            check = "";
        }
    }
    for(i=0;i<table1.getRowCount();i++){
    }
    totalField.setText(total+ "");
    scanField.setText("");
}
}

```

ส่วนนี้คือการทำงานของกล่องสแกนบาร์โค้ดเพื่อนำไปประมวลผลในการเรียกใช้ข้อมูลจาก Database แล้วนำไปประมวลผลในการคิดราคาต่อไป และเพิ่มข้อมูลที่สแกนใส่ในตารางรายการ


```

private void finishButtonMouseClicked(java.awt.event.MouseEvent evt) {
    // TODO add your handling code here:

    if(totalField.getText().equalsIgnoreCase("xxx")){
    }else{
        FinishPopup.setVisible(true);
        try {
            // TODO add your handling code here:
            // "INSERT INTO `foodmenu`.`summary_table` (`DATE/TIME`, `Total`) VALUES ('sssss', '125')
            String tmp = "INSERT INTO `foodmenu`.`summary_table` (`DATE/TIME`, `TOTAL PRICE`) VALUES ('"+formattedDate01+"', '"+totalField.getText()+"')";
            PreparedStatement ppsm = DB.conn.prepareStatement(tmp);
            ppsm.executeUpdate();
        } catch (Exception e) {
            e.printStackTrace();
        }
        dispose();
        try {
            SaleUI ui = new SaleUI();
            ui.setVisible(true);
            FinishPopup.setVisible(true);
            FinishPopup.isAlwaysOnTop();
        } catch (SQLException ex) {
            Logger.getLogger(SaleUI.class.getName()).log(Level.SEVERE, null, ex);
        }
    }
}

```

ส่วนนี้คือส่วนของการกดปุ่ม (Finish) เมื่อกดแล้วจะนำข้อมูล ราคารวมของสินค้าทั้งหมด และ วันเวลา ณ ตอนนั้น ไปเก็บไว้ใน ฐานข้อมูล ในตารางที่มีชื่อว่า Summary_Table

```

private void AddButton01ActionPerformed(java.awt.event.ActionEvent evt) {
    try {
        // TODO add your handling code here:
        String tmp_check = "INSERT INTO `foodmenu`.`food_menu` (`barcode`, `foodName`, `price`) VALUES ('101010', 'Soda', '10')";
        String tmp = "INSERT INTO `foodmenu`.`food_menu` (`barcode`, `foodName`, `price`) VALUES ('"+
            + "("+BarcodeField01.getText()+"', '"+NameField01.getText()+"', '"+PriceField01.getText()+"')";
        System.out.println(tmp_check);
        PreparedStatement ppsm = DB.conn.prepareStatement(tmp);
        ppsm.executeUpdate();
        BarcodeField01.setText("");
        NameField01.setText("");
        PriceField01.setText("");
        AddMenuComp.setSize(400, 300);
        AddMenuComp.isAlwaysOnTop();
        AddMenuComp.setVisible(true);
    } catch (SQLException ex) {
        Logger.getLogger(SaleUI.class.getName()).log(Level.SEVERE, null, ex);
    }
}

```

ส่วนนี้คือส่วนหน้าต่างเพิ่มข้อมูลลงไปในฐานข้อมูล โดยสิ่งที่เพิ่มไปคือ บาร์โค้ด ชื่อเมนู ราคา โดยใช้คำสั่ง PreparedStatement เพื่อเพิ่มข้อมูลไปยัง MySQL

```
private void Remove02ButtonMouseClicked(java.awt.event.MouseEvent evt) {
    // TODO add your handling code here:
    //DELETE FROM `foodmenu`.`food_menu` WHERE (`barcode` = '7777');
    String tmp = "DELETE FROM `foodmenu`.`food_menu` WHERE (`barcode` = '"+BarcodeRemoveField.getText()+"')";

    try {
        PreparedStatement ppsm = DB.conn.prepareStatement(tmp);
        ppsm.executeUpdate();
    } catch (SQLException ex) {
        Logger.getLogger(SaleUI.class.getName()).log(Level.SEVERE, null, ex);
    }

    RemoveMenuUI.setVisible(false);
    BarcodeRemoveField.setText("");
}
}
```

ส่วนนี้คือ หน้าต่างลบข้อมูลในฐานข้อมูลของเรา โดยสิ่งที่พิมพ์เข้าไปคือ บาร์โค้ด โดยใช้คำสั่ง
PreparedStatement และใช้คำสั่งของภาษา SQL ในการลบข้อมูลในตาราง

```
private void summaryButtonMouseClicked(java.awt.event.MouseEvent evt) {
    try {
        // TODO add your handling code here:
        resultSetSum = DB.conn.prepareStatement("SELECT * FROM SUMMARY_TABLE").executeQuery();
    } catch (SQLException ex) {
        Logger.getLogger(SaleUI.class.getName()).log(Level.SEVERE, null, ex);
    }
    int i = 0;
    try {
        //ADD DATA TO SummaryTable
        while(resultSetSum.next()){
            dateTime.add(resultSetSum.getString("DATE/TIME"));
            totalTable.add(resultSetSum.getString("TOTAL PRICE"));
            System.out.println(resultSetSum.getString("DATE/TIME"));
            SummaryTable.setValueAt(resultSetSum.getString("DATE/TIME"), i, 0);
            SummaryTable.setValueAt(resultSetSum.getString("TOTAL PRICE"), i, 1);
            i=i+1;
        }
    } catch (SQLException ex) {
        Logger.getLogger(SaleUI.class.getName()).log(Level.SEVERE, null, ex);
    }
    int j=0;
    SummaryUI.setSize(800, 800);
    SummaryUI.isAlwaysOnTop();
    SummaryUI.setVisible(true);
}
}
```

ส่วนนี้คือหน้าต่างสรุปการขายของแต่ละบิล ซึ่งจะแสดงเป็นตาราง แล้วดึงข้อมูลจากฐานข้อมูลมาแสดงใน
ตารางโดยจะแสดง วันเวลาที่ขาย และยอดขายที่ได้ ณ เวลานั้น

```

    */
    public static void main(String args[]) {
        /* Set the Nimbus look and feel */
        Look and feel setting code (optional)

        /* Create and display the form */
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                try {
                    new SaleUI().setVisible(true);
                    new JavaDatabase();
                } catch (SQLException ex) {
                    Logger.getLogger(SaleUI.class.getName()).log(Level.SEVERE, null, ex);
                }
            }
        });
    }
}

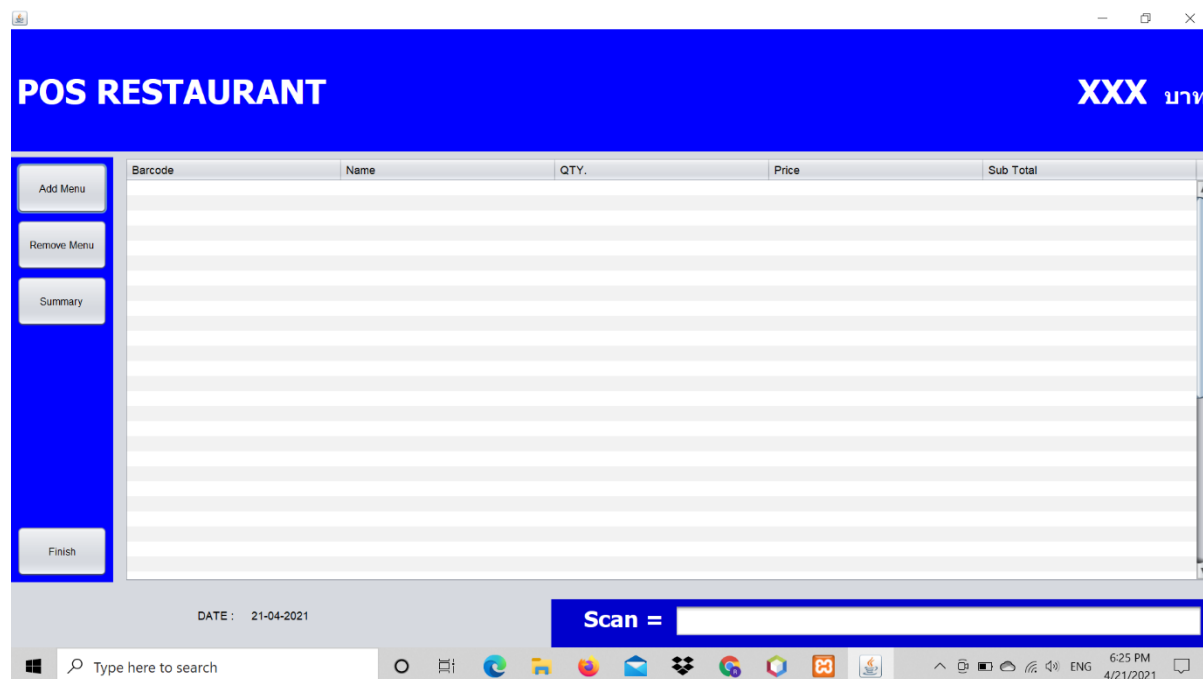
```

ส่วนนี้คือหน้า Main ในการรันโปรแกรมทั้งหมด โดยจะสร้างวัตถุจากคอนสแตคเตอร์ SaleUI แล้วโปรแกรมจะเริ่มทำงาน

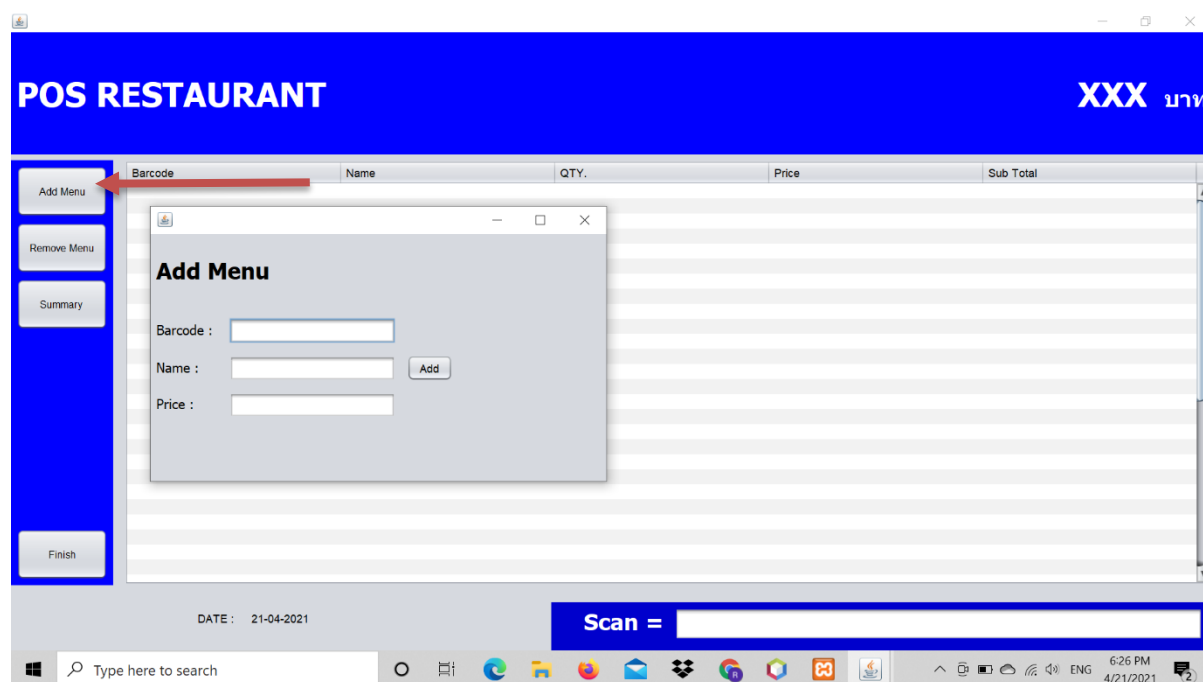
ผลการดำเนินงาน

หน้าต่างแสดงผลการทำงานของแอปพลิเคชัน

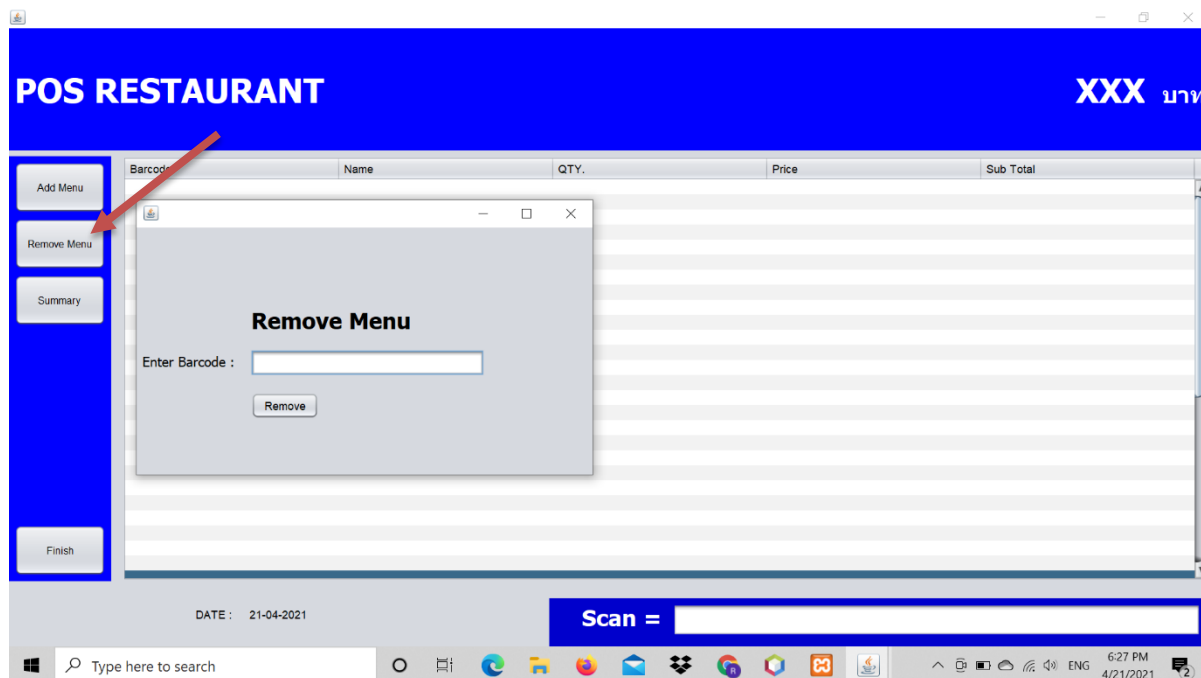
1. หน้าต่างของหน้าเริ่มต้นการทำงานของแอปพลิเคชัน



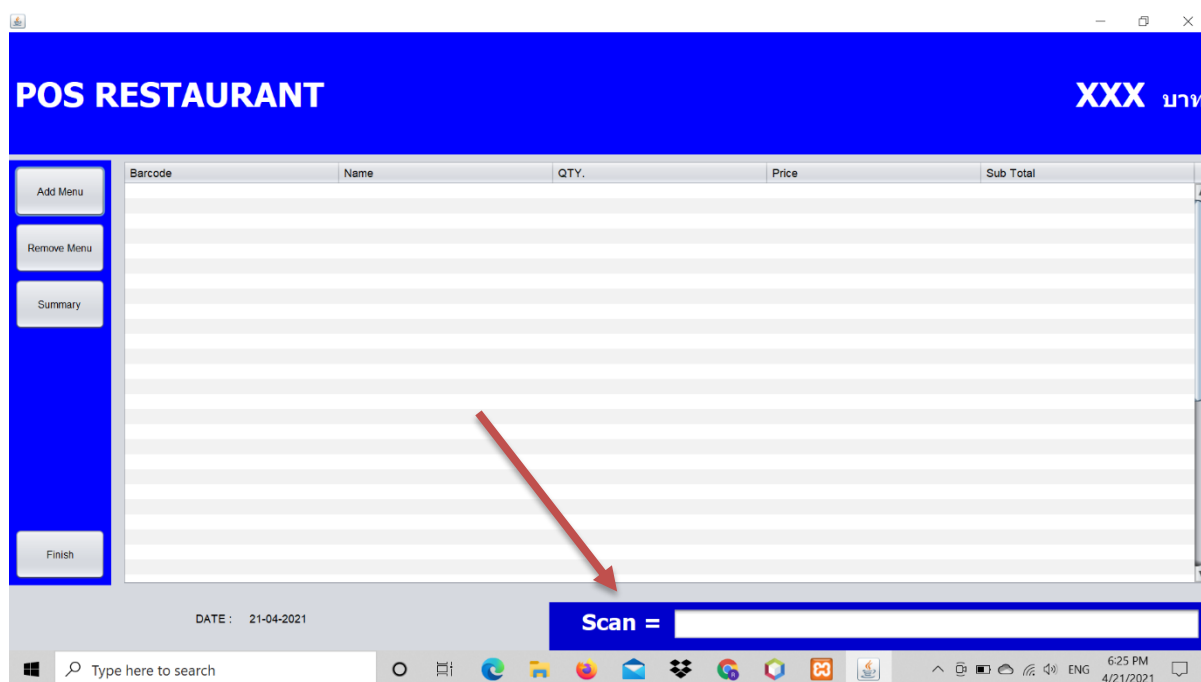
2. กด Add Menu เพื่อเพิ่ม Barcode Name และ Price กด Add สินค้าจะเพิ่มเข้าไปใน Menu



3. ถ้าต้องการลบเมนู ให้กด Remove Menu ใส่ Barcode และกด Remove เพื่อลบรายการเมนูนั้นออก

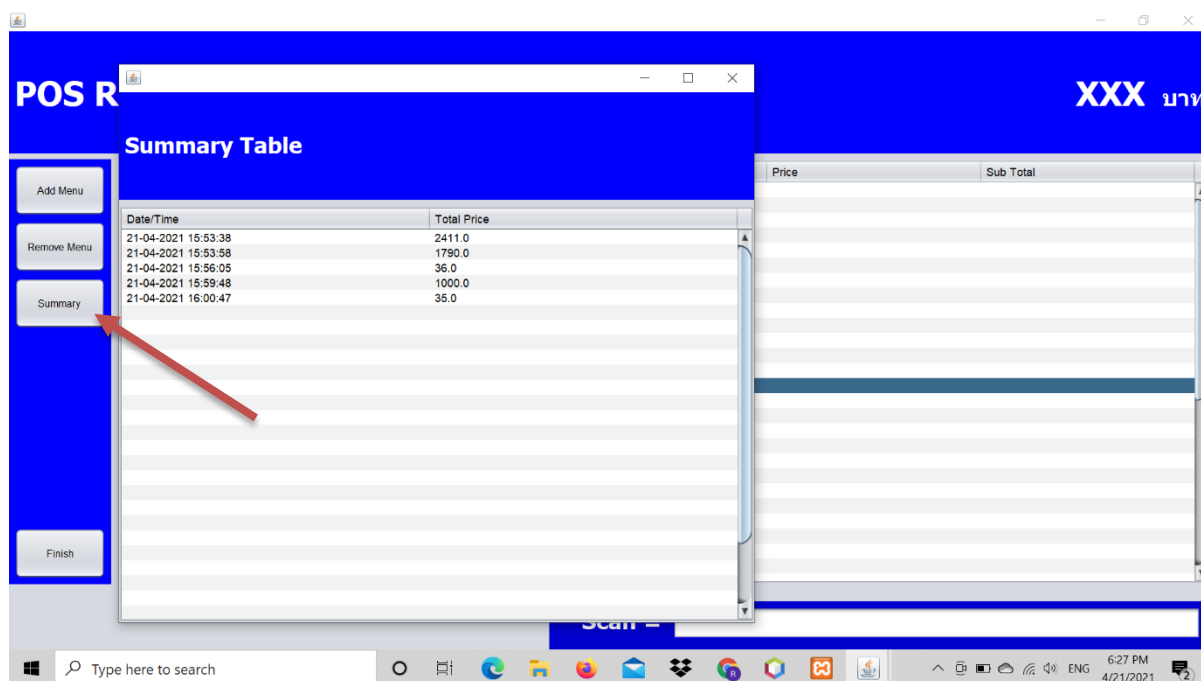


4. ใส่ Barcode ตรง Scan กด Enter เพื่อให้รายการสินค้าขึ้นการคำนวณ



5. หลังจากกรอกรายการสินค้าทุกอย่างเรียบร้อยแล้ว ให้กด Finish

6. ถ้าต้องการดูรายได้ที่ได้ทั้งหมดให้กด Summary



อ้างอิง

Netbeans : สร้าง Java GUI และการสร้าง Event Action และ Dialog ได้ตอบแบบง่าย ๆ

<https://www.thaicreate.com/java/java-gui-netbeans.html>

(วันที่สืบค้นข้อมูล 19 เมษายน 2564).

How to connect to database connection in Java

[https://stackoverflow.com/questions/21361781/how-to-connect-to-database-connection-in-java?fbclid=IwAR2usb5XsC7ysmmJ4zL41tfF8RnVTtKjgzPPqLhi-](https://stackoverflow.com/questions/21361781/how-to-connect-to-database-connection-in-java?fbclid=IwAR2usb5XsC7ysmmJ4zL41tfF8RnVTtKjgzPPqLhi-DEkJglulAYr8QT7Mdk)

DEkJglulAYr8QT7Mdk. (วันที่สืบค้นข้อมูล 19 เมษายน 2564).

สอน SQL รูปแบบคำสั่ง SQL และการใช้งานภาษา SQL ในรูปแบบต่าง ๆ

<https://www.thaicreate.com/tutorial/sql.html> (วันที่ สืบค้นข้อมูล 19 เมษายน 2564).