



เรื่อง การพัฒนาโปรแกรมด้วยภาษาจาวา

จัดทำโดย

นาย คณากร วงศ์ไชยา 6630300106

เสนอ

อาจารย์กุลวดี สมบูรณ์วิวัฒน์

รายงานเล่มนี้เป็นส่วนหนึ่งของการเรียนการสอนวิชา Program Fundamental II

ภาคเรียนที่ 2 ปีการศึกษา 2566

มหาวิทยาลัยเกษตรศาสตร์ วิทยาเขตศรีราชา

## คำนำ

รายงานเล่มนี้เป็นส่วนหนึ่งของรายวิชา Program Fundamental II รหัสวิชา 256602 เพื่อศึกษาและใช้ข้อมูลที่เรียนรู้มาทำโปรเจกต์ให้เกิดเป็นชิ้นงานจริง โดยรายงานเล่มนี้มีเนื้อหาเกี่ยวกับการนำเสนอโปรเจกต์ การอธิบายโปรแกรมและการติดตั้งโปรแกรม

ผู้จัดทำหวังเป็นอย่างยิ่งว่ารายงานฉบับนี้จะทำให้ผู้อ่านรับรู้ถึงการทำงานของโปรแกรมที่จัดทำ และเข้าใจโค้ดที่ใส่ไปในโปรแกรม

ผู้จัดทำ

## สารบัญ

คำนำ.....	i
สารบัญ.....	ii
สารบัญรูปภาพ.....	iii
แนวคิดการออกแบบ.....	1
User Interface.....	1
UML Diagram.....	3
Algorithm.....	4
วิธีการติดตั้ง.....	8
วิธีการใช้งาน.....	9
ตัวอย่างการรันโปรแกรม.....	10
เอกสารอ้างอิง.....	12

## สารบัญรูปภาพ

Figure 1 ProfilePage .....	1
Figure 2 BookingPage.....	2
Figure 3 PaymentPage .....	2
Figure 4 Receipt.....	3
Figure 5 UML Diagram.....	3
Figure 6 MainApp.java .....	4
Figure 7 AppWindow.java.....	4
Figure 8 Data.java .....	5
Figure 9 Profile.java .....	5
Figure 10 BookingPage.java .....	6
Figure 11 PaymentPage.java .....	7
Figure 12 Receipt.java .....	7

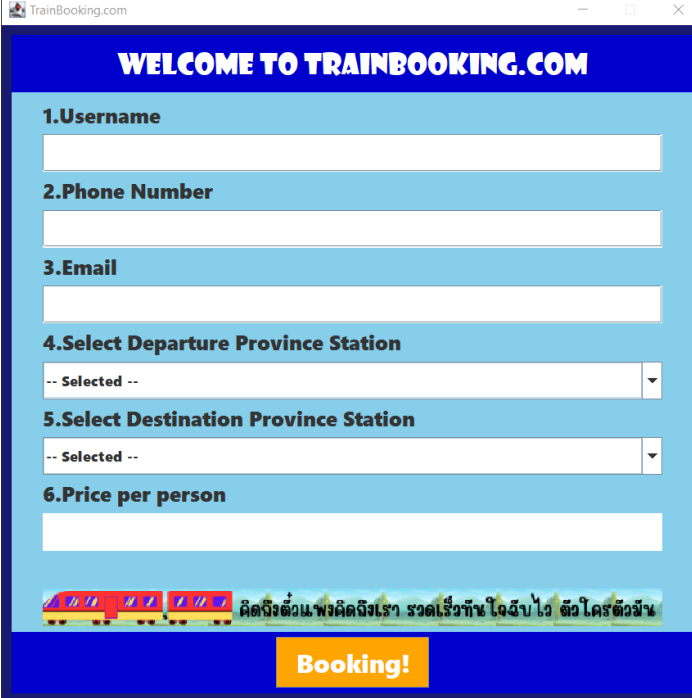
## แนวคิดการออกแบบ

เหตุผลที่เลือกทำ Project นี้เพราะเป็นคนชอบเดินทาง เป็นคนชอบเที่ยว เลยอยากจำทำโปรแกรมเกี่ยวกับการท่องเที่ยว และตอนเด็กเป็นคนที่ชอบรถไฟมากๆ จึงเกิดเป็นโปรเจกนี้ TrainBooking.com ขึ้นมาเป็นเว็บที่รวบรวมจังหวัดท่องเที่ยวของประเทศไทย

## User Interface

### 1. ProfilePage

เป็นแรกของโปรแกรมที่จะแสดงให้ผู้ใช้กรอกข้อมูล ชื่อ เบอร์โทรศัพท์ อีเมล เลือกสถานีต้นทางและสถานีปลายทาง และเมื่อผู้ใช้เลือกสถานีต้นทาง/ปลายทาง ก็จะแสดงราคาให้เลย



The screenshot shows a web browser window with the URL 'TrainBooking.com'. The page has a blue header with the text 'WELCOME TO TRAINBOOKING.COM'. Below the header, there is a light blue form area with six numbered fields: 1. Username (text input), 2. Phone Number (text input), 3. Email (text input), 4. Select Departure Province Station (dropdown menu with '-- Selected --'), 5. Select Destination Province Station (dropdown menu with '-- Selected --'), and 6. Price per person (text input). At the bottom of the form area, there is a small graphic of a train and the text 'คิดจิงตัวแพงคิดจิงเรา รวดเร็วทันใจมาไว ตัวใครตัวมัน'. Below the form area, there is a blue footer with a yellow 'Booking!' button.

Figure 1 ProfilePage

## 2. BookingPage

เป็นหน้าที่ให้ผู้ใช้เลือกที่นั่ง โดยโปรแกรมจะแสดงที่นั่งที่ว่าง ที่นั่งที่ถูกจอง และที่นั่งที่กำลังเลือกให้ผู้ใช้ได้เลือก



Figure 2 BookingPage

## 3. PaymentPage

เป็นหน้าที่ให้ผู้ใช้ทำการชำระเงิน

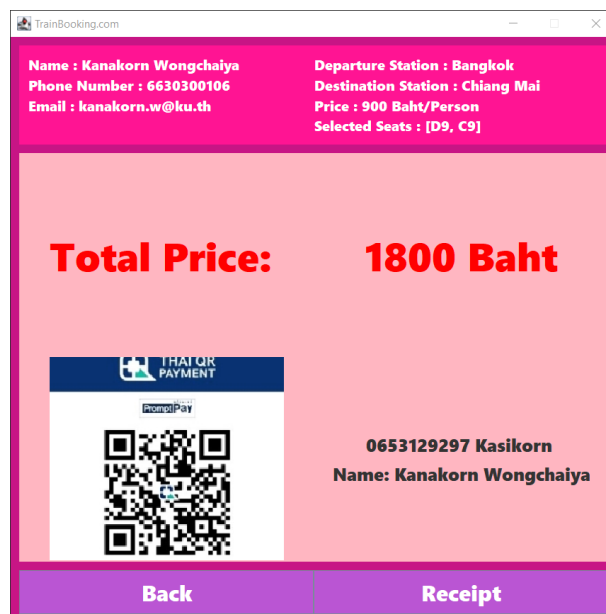


Figure 3 PaymentPage

#### 4. ReceiptPage

แสดงสรุปผลการจองให้ผู้ใช้งาน

The screenshot shows a window titled 'RecieptTrainBookin...' with a red border. Inside, a grey box contains the following text: 'Name : Kanakorn Wongchaiya', 'Phone Number : 6630300106', 'Email : kanakorn.w@ku.th', 'Departure Station : Bangkok', 'Destination Station : Chiang Mai', 'Price : 900 Baht/Person', 'Selected Seats : [D9, C9]', and 'Total Price: 1800 Baht' in red. A yellow 'Exit' button is at the bottom.

Figure 4 Receipt

#### UML Diagram

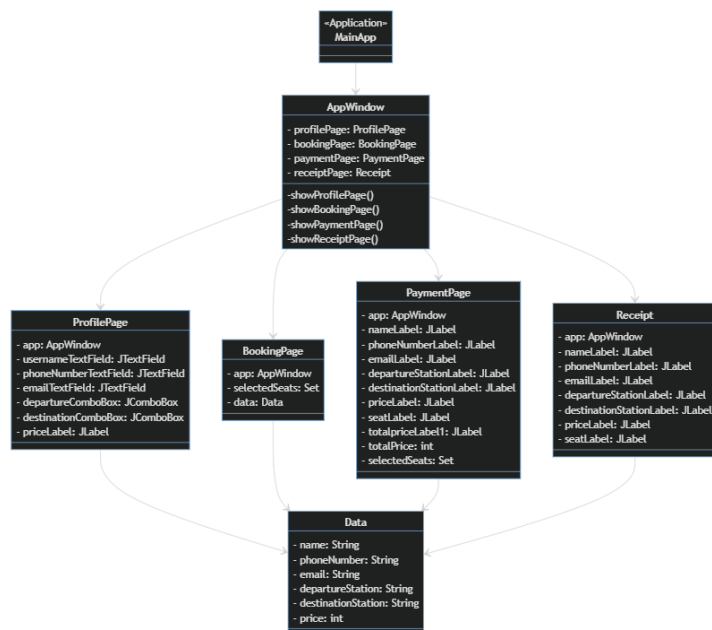


Figure 5 UML Diagram

- Class MainApp ใช้สั่งรันโปรแกรม
- Class Data ใช้เก็บข้อมูล รับ/ส่งข้อมูลทั้งหมด
- Class AppWindow เป็นคลาสที่ใช้สลับหน้าต่างของโปรแกรมทั้งหมด
- Class ProfilePage ใช้แสดงหน้าต่างโปรแกรมให้รับข้อมูลของผู้ใช้
- Class BookingPage ใช้แสดงที่นั่งให้ผู้เลือก
- Class PaymentPage ใช้แสดงช่องทางการชำระเงินและราคารวม
- Class Receipt ใช้แสดงใบเสร็จ

## Algorithm

### 1. MainApp.java

เมทอด `invokeLater()` ใช้เรียกใช้งานแอปพลิเคชันใน EDT เมื่อการทำงานของ EDT พร้อมแล้ว `Runnable` จะถูกเรียกเมื่อเรียกใช้ `invokeLater()` กับอ็อบเจกต์ `Runnable` ที่สร้างขึ้นในเมทอด `run()` จะสร้างอ็อบเจกต์ `AppWindow` และทำการแสดงอ็อบเจกต์ ด้วยการเรียกเมทอด `setVisible(true)` เพื่อแสดงหน้าต่างของแอปพลิเคชันและเรียก `setLocationRelativeTo(null)` ใช้เพื่อกำหนดตำแหน่งของหน้าต่างแอปพลิเคชันให้อยู่ตรงกลางของหน้าจอ.

```
1 package FinalProject.src;
2
3 import javax.swing.SwingUtilities;
4
5 public class MainApp {
6     public static void main(String[] args) {
7         SwingUtilities.invokeLater(new Runnable() {
8             // Event Dispatch Thread (EDT)
9             @Override
10            public void run(){
11                AppWindow app = new AppWindow();
12                app.setLocationRelativeTo(c:null);
13                app.setVisible(b:true);
14            }
15        });
16    }
17 }
```

Figure 6 MainApp.java

### 2. AppWindow.java

คลาส `AppWindow` เป็นคลาสหลักที่ใช้สำหรับการจัดการหน้าต่างและการเปลี่ยนแปลงหน้าต่างในแอปพลิเคชัน มีการสร้างอ็อบเจกต์รับส่งข้อมูล และประกาศเมทอดต่างที่ใช้ในการเปลี่ยนหน้าต่างของโปรแกรม การทำงานของโปรแกรม มีการสืบทอดมาจากคลาส `JFrame` โดยการเปลี่ยนหน้าต่างกระดาดจะเป็นการใช้เมทอด `setContentPane` ในการเปลี่ยนหน้าต่าง

```
public class AppWindow extends JFrame{
    private ProfilePage profilePage;
    private BookingPage bookingPage;
    private PaymentPage paymentPage;
    private Receipt receipt;

    public AppWindow() {
        setTitle("TrainBooking.com");
        profilePage = new ProfilePage(this);
        setContentPane(profilePage);
        setSize(width:700,height:700);
        setResizable(resizable:false);
        setDefaultCloseOperation(EXIT_ON_CLOSE);
    }

    public void bookingButtonPressed(Data data){
        bookingPage = new BookingPage(this, data);
        setContentPane(bookingPage);
        revalidate();
    }

    public void paymentButtonPressed(Data data , Set<String> selectedSeats) {
        paymentPage = new PaymentPage(this, data ,selectedSeats);
        setContentPane(paymentPage);
        revalidate();
    }
}
```

Figure 7 AppWindow.java



### 3. Data.java

มีการ Constructor Overloading เป็นการสร้าง Constructor ที่มีชื่อเหมือนกันแต่ต่างกันที่การรับพารามิเตอร์ จากในโปรแกรมในคลาส BookingPage มีการรับ Data มาจากคลาส ProfilePage ซึ่งยังไม่มีการรับพารามิเตอร์ selectedSeats จึงเรียกใช้ Constructor ตัวล่าง แต่ในคลาส PaymentPage , Reciept มีการรับพารามิเตอร์ selectedSeats แล้วจึงต้องประกาศใช้ด้วย

```
// Constructor
public Data(String name, String phoneNumber, String email, String departureStation, String destinationStation, int price, Set<String> selectedSeats) {
    this.name = name;
    this.phoneNumber = phoneNumber;
    this.email = email;
    this.departureStation = departureStation;
    this.destinationStation = destinationStation;
    this.price = price;
    this.selectedSeats = selectedSeats;
}

public Data(String name, String phoneNumber, String email, String departureStation, String destinationStation, int price) {
    this.name = name;
    this.phoneNumber = phoneNumber;
    this.email = email;
    this.departureStation = departureStation;
    this.destinationStation = destinationStation;
    this.price = price;
}
```

Figure 8 Data.java

### 4. ProfilePage.java

มีการสร้าง HashMap เพื่อเก็บข้อมูลของสถานีปลายทางตามราคาของสินค้า โดยการแสดงข้อมูลให้ผู้ใช้เลือก สถานีปลายทางจะแสดงอยู่ใน JComboBox มีการใส่ ActionListener ให้กับ JComboBox เมื่อมีการกดเลือก สถานีปลายทางที่ JComboBox จะให้แสดงข้อมูลราคาตัวจาก priceMap กล่าวคือ เมื่อผู้ใช้เลือกสถานีปลายทาง จาก ComboBox ราคาของการเดินทางที่เกี่ยวข้องจะถูกแสดงทันที

```
// สร้าง Map เก็บราคาตามจังหวัด
priceMap = new HashMap<>();
priceMap.put(key: "-- Selected --", value:0);
priceMap.put(key: "Trang", value:790);
priceMap.put(key: "Nakhon Nayok", value:200);
priceMap.put(key: "Chonburi", value:150);
priceMap.put(key: "Nakhon Ratchasima", value:500);
priceMap.put(key: "Ayutthaya", value:220);
priceMap.put(key: "Chanthaburi", value:450);
priceMap.put(key: "Chiang Mai", value:900);
priceMap.put(key: "Chiang Rai", value:900);
priceMap.put(key: "Phuket", value:900);
priceMap.put(key: "Surat Thani", value:800);

//สร้าง Action listener DestinationComboBox
private class DestinationComboBoxHandler implements ActionListener {
    @Override
    public void actionPerformed(ActionEvent e) {
        //เมื่อเลือกสถานีปลายทางจาก destinationComboBox โดยได้ getSelectedItem() ส่งคืนชื่อของสถานีปลายทาง
        //และแปลงชื่อของสถานีปลายทางเป็น String โดยได้เพิ่มเข้าไปในราคาเดิม เป็นราคาที่ถูกเลือกไปพร้อมกับ selectedDestination/
        String selectedDestination = (String) destinationComboBox.getSelectedItem();
        //ใช้ค่าของข้อมูลราคาของสถานีปลายทางตามสถานีปลายทางที่ได้ โดยได้ priceMap เป็น Map ที่เก็บราคาตามจังหวัด
        int price = priceMap.get(selectedDestination);
        //ใช้แสดงราคาที่ได้อีก priceMap โดยได้ setText() เพื่อกำหนดข้อความที่จะแสดงบน JLabel ที่ชื่อ priceLabel โดยแปลงค่า price ให้เป็น String
        priceLabel.setText(Integer.toString(price) + " Bath/person");
    }
}
```

Figure 9 Profile.java

## 5. BookingPage.java

มีการสร้างแผนผังที่นั่งเป็นแบบปุ่มกดและมีการ AddActionListener ที่ถูกเพิ่มให้กับปุ่มที่นั่งที่แสดงบนหน้าจอ เมื่อผู้ใช้คลิกที่ปุ่มที่นั่ง ปุ่มนั้นจะเปลี่ยนเป็นสีม่วงเพื่อบอกสถานะว่ากำลังเลือกที่นั่งตรงนั้นอยู่ และมีการดึงข้อความที่อยู่บนปุ่มที่นั่งที่เลือกเพื่อนำข้อมูลไปใช้ต่อไป

```
//สร้างปุ่มที่นั่งตามขนาดของ
JPanel leftseat = new JPanel();
leftseat.setLayout(new BorderLayout(leftseat, BorderLayout.PAGE_AXIS)); // เปลี่ยนเป็น BorderLayout
leftseat.setBorder(new EmptyBorder(top:10, left:10, bottom:10, right:10));
char[] rows = {'A', 'B', 'C', 'D'};
for (int i = 1; i <= 9; i++) {
    JPanel rowPanel = new JPanel();
    rowPanel.setLayout(new BorderLayout(rowPanel, BorderLayout.LINE_AXIS));
    rowPanel.setBorder(new EmptyBorder(top:5, left:5, bottom:5, right:10));
    for (char row : rows) {
        JButton seatButton = new JButton(row + String.valueOf(i));
        seatButton.addActionListener(new SeatButtonHandler()); // เพิ่ม ActionListener ให้กับปุ่ม
        seatButton.setPreferredSize(new Dimension(width:50, height:50)); // กำหนดขนาดของปุ่ม
        seatButton.setBackground(Color.WHITE);
        rowPanel.add(seatButton);
        rowPanel.setBackground(new Color(r:102,g:205,b:170));
        rowPanel.add(Box.createRigidArea(new Dimension(width:5, height:5))); // ช่องว่างระหว่างปุ่ม
    }
    leftseat.add(rowPanel);
    leftseat.setBackground(new Color(r:144,g:238,b:144));
}
seat.add(leftseat, BorderLayout.CENTER);

//เพิ่ม Action ให้กับปุ่ม
private class SeatButtonHandler implements ActionListener {
    @Override
    public void actionPerformed(ActionEvent e) {
        JButton button = (JButton) e.getSource(); //ดึงชื่อของ JButton ที่ทำให้เกิด ActionEvent โดยไม่หมด getSource() ของชื่อของ ActionEvent ที่ถูกส่งมา
        String seat = button.getText(); //ดึงข้อความที่นั่งที่เลือก ซึ่งในที่นี้คือค่าแบบนามธรรมซึ่ง โดยไม่หมด getText() ของชื่อของ JButton
        //สร้างเงื่อนไขเพื่อตรวจสอบว่าที่นั่งที่ถูกเลือกแล้วหรือไม่
        if (selectedSeats.contains(seat)) {
            selectedSeats.remove(seat); // ถ้าที่นั่งถูกเลือกแล้ว ก็ลบออกจากเซต
            button.setBackground(Color.WHITE); // ลบสีพื้นหลังที่เลือก
        } else {
            selectedSeats.add(seat); // ถ้าที่นั่งยังไม่ถูกเลือก ก็เพิ่มเข้าไปในเซต
            button.setBackground(Color.MAGENTA); // ตั้งสีพื้นหลังเป็นสีม่วงเมื่อเลือก
        }
    }
}
```

PaymentButtonHandler เป็น ActionListener ที่ถูกเพิ่มให้กับปุ่ม Payment เมื่อผู้ใช้คลิกที่ปุ่ม Payment โปรแกรมจะตรวจสอบว่ามีที่นั่งที่ถูกเลือกหรือไม่ หากไม่มี โปรแกรมจะแสดงข้อความเตือนให้ผู้ใช้เลือกที่นั่งก่อน หากมีที่นั่งที่ถูกเลือก โปรแกรมจะสร้างอ็อบเจกต์ Data จากข้อมูลที่ได้รับเข้ามา และส่งข้อมูลไปยัง AppWindow เพื่อเปิดหน้าจอ PaymentPage

```
//เพิ่ม Action ให้กับ Payment
private class PaymentButtonHandler implements ActionListener {
    @Override
    public void actionPerformed(ActionEvent e) {
        // ตรวจสอบว่ามีที่นั่งที่ถูกเลือกหรือไม่
        if (selectedSeats.isEmpty()) {
            JOptionPane.showMessageDialog(app, message:"Please select a seat before proceeding to payment.");
            return; // ไม่ทำงานและหากไม่มีที่นั่งที่ถูกเลือก
        }
        // ดึงข้อมูลจาก Label และใช้ในการสร้าง Data object
        String name = nameLabel.getText().substring(beginIndex:7); // ชื่อคำว่า "Name: " ลบ
        String phoneNumber = phoneNumberLabel.getText().substring(beginIndex:15); // ชื่อคำว่า "Phone Number: " ลบ
        String email = emailLabel.getText().substring(beginIndex:9); // ชื่อคำว่า "Email: " ลบ
        String departureStation = departureStationLabel.getText().substring(beginIndex:19); // ชื่อคำว่า "Departure Station: " ลบ
        String destinationStation = destinationStationLabel.getText().substring(beginIndex:21); // ชื่อคำว่า "Destination Station: " ลบ
        int price = Integer.parseInt(priceLabel.getText().substring(beginIndex:8, priceLabel.getText().indexOf(str: " Baht"))); // ชื่อคำว่า "Price: " และเอาเฉพาะตัวเลขมา

        // สร้าง Data object ใหม่ พร้อมกับการส่ง selectedSeats
        Data data = new Data(name, phoneNumber, email, departureStation, destinationStation, price, selectedSeats);

        // ส่งข้อมูลไปยัง AppWindow เพื่อเปิดหน้าจอ PaymentPage
        app.paymentButtonPressed(data, selectedSeats);
        app.revalidate();
    }
}
```

Figure 10 BookingPage.java

## 6. PaymentPage.java

ReceiptButtonHandler เป็น ActionListener ที่ถูกเพิ่มให้กับปุ่ม Receipt เมื่อผู้ใช้คลิกที่ปุ่ม Receipt โปรแกรมจะโปรแกรมจะสร้างอ็อบเจกต์ Data จากข้อมูลที่ได้รับเข้ามา และส่งข้อมูลไปยัง AppWindow เพื่อเปิดหน้าจอ Receipt

```
//สร้าง Action ให้ปุ่ม Receipt
private class ReceiptButtonHandler implements ActionListener {
    @Override
    public void actionPerformed(ActionEvent e) {
        String name = nameLabel.getText().substring(beginIndex:7);
        String phoneNumber = phoneNumberLabel.getText().substring(beginIndex:15);
        String email = emailLabel.getText().substring(beginIndex:8);
        String departureStation = departureStationLabel.getText().substring(beginIndex:19);
        String destinationStation = destinationStationLabel.getText().substring(beginIndex:21);
        int price = Integer.parseInt(priceLabel.getText().substring(beginIndex:8, priceLabel.getText().indexOf(str:" Baht")));
        int totalPrice = Integer.parseInt(totalpriceLabel1.getText().substring(beginIndex:0, totalpriceLabel1.getText().indexOf(str:" Baht")));

        Data data = new Data(name, phoneNumber, email, departureStation, destinationStation, price, selectedSeats);
        app.showReceiptPage(data, selectedSeats, totalPrice);
    }
}
```

Figure 11 PaymentPage.java

## 7. Receipt.java

เมื่อกดปุ่ม Exit จะมี JOptionPane ขึ้นมาเพื่อให้กดยืนยันอีกครั้ง

```
//สร้าง Action ให้ปุ่ม Exit
private class ExitButtonHandler implements ActionListener {
    @Override
    public void actionPerformed(ActionEvent e) {
        int confirmed = JOptionPane.showConfirmDialog(parentComponent:null, message:"Are you sure you want to exit?", title:"Exit Confirmation",
                                                    JOptionPane.YES_NO_OPTION);
        if (confirmed == JOptionPane.YES_OPTION) {
            app.dispose(); // ปิด AppWindow
        }
    }
}
```

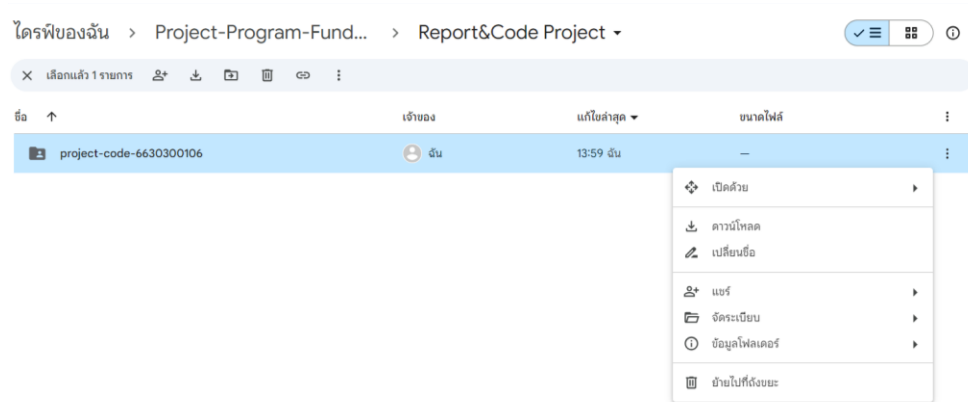
Figure 12 Receipt.java

## วิธีการติดตั้ง

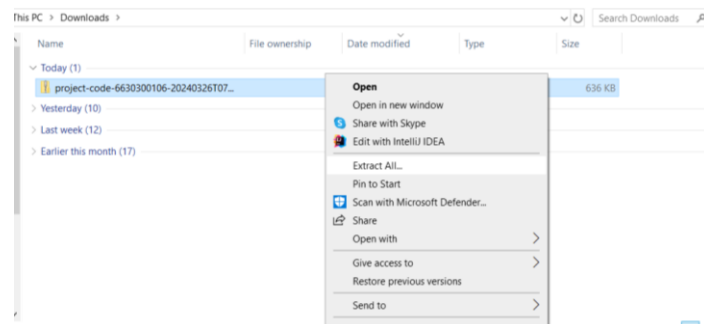
1. ไปที่ลิงค์ <https://drive.google.com/drive/folders/1opi5XD9hOf008302gWSSfAgtnewGBBX-?usp=sharing>
2. คลิกไปที่โฟลเดอร์ที่ชื่อ Report&Code Project



3. เลือก project-code-6630300106
4. กด 3 จุด เลือกดาวน์โหลด



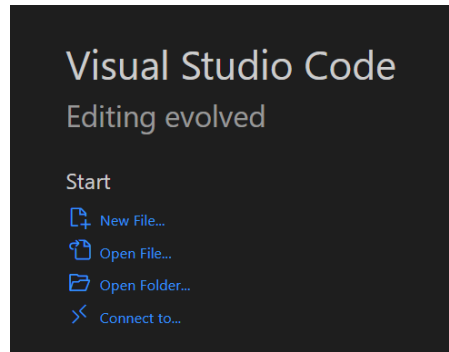
5. แยกไฟล์ ไปที่ไฟล์ที่ดาวโหลด -> คลิกขวา -> Extract All -> เลือกที่อยู่ที่ต้องการ



6. ใช้งานโปรแกรมได้เลย

## วิธีการใช้งาน

1. เปิดโปรแกรมที่ต้องการใช้ -> เลือก Open Folder



2. สั่ง Run ที่ MainApp.java และใช้งานได้เลย
3. กรอกข้อมูลให้ครบไม่งั้นไม่สามารถไปหน้าต่อไปได้
4. การเลือกที่นั่งไม่งั้นไม่สามารถไปหน้าต่อไปได้
5. ทำการจ่ายเงิน
6. ปรี้นใบเสร็จ

## ตัวอย่างการรันโปรแกรม

**\*\*อาจมีปัญหาด้านรูปภาพ\*\***

1.



**WELCOME TO TRAINBOOKING.COM**

**1.Username**  
Kanakorn Wongchaiya

**2.Phone Number**  
6630300106

**3.Email**  
Kanakorn.w@ku.th

**4.Select Departure Province Station**  
-- Selected --

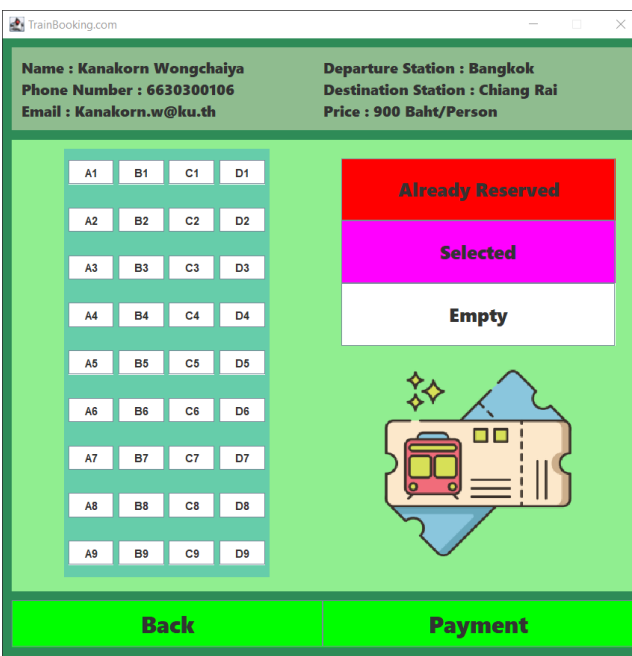
**5.Select Destination Province Station**  
Chiang Rai

**6.Price per person**  
900 Bath/person

คิดจึรตัวแพงคิดจึรเรา รวดเร้วทันใจจับไว ตัวใครตัวมัน

**Booking!**

2.



Name : Kanakorn Wongchaiya  
Phone Number : 6630300106  
Email : Kanakorn.w@ku.th

Departure Station : Bangkok  
Destination Station : Chiang Rai  
Price : 900 Baht/Person

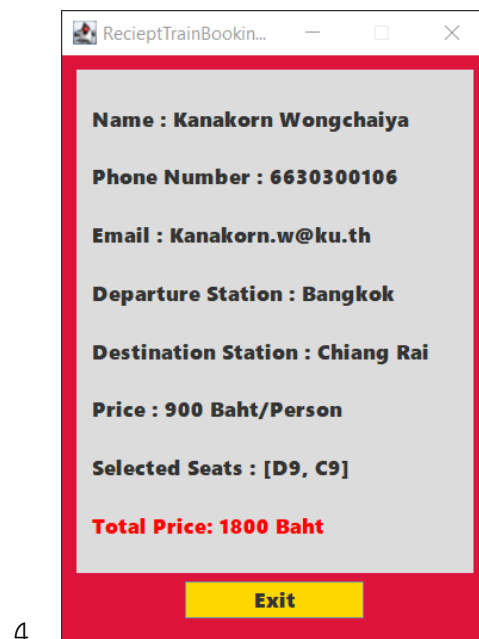
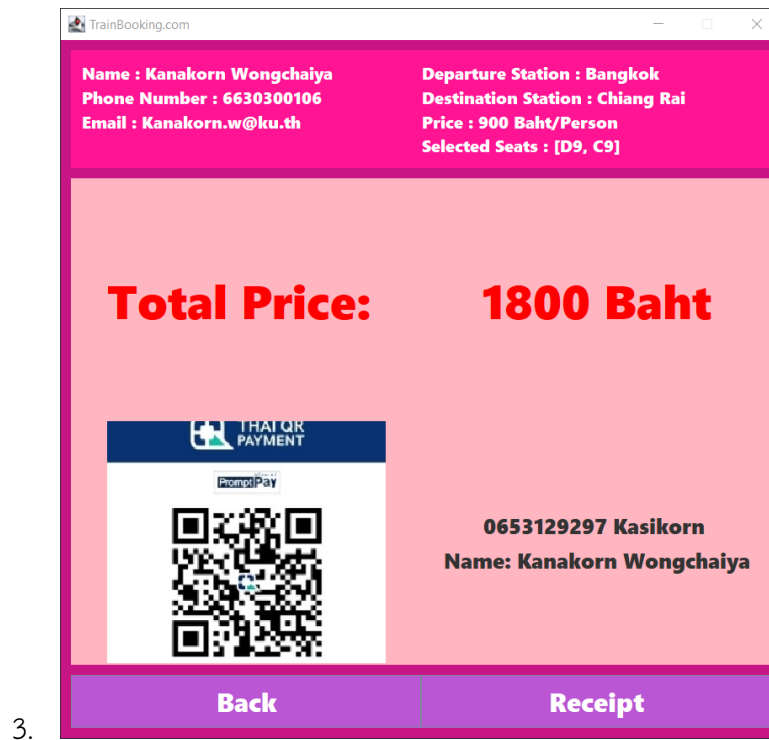
A1	B1	C1	D1
A2	B2	C2	D2
A3	B3	C3	D3
A4	B4	C4	D4
A5	B5	C5	D5
A6	B6	C6	D6
A7	B7	C7	D7
A8	B8	C8	D8
A9	B9	C9	D9

**Already Reserved**

**Selected**

**Empty**

**Back** **Payment**



## เอกสารอ้างอิง

- guidemo.zip From Aj. Kulwadee Somboonviwat

<https://o365ku.sharepoint.com/:u:/r/sites/256602-03603112ProgrammingFundamentalsII/Class%20Materials/07-SwingGUI/guidemo.zip?csf=1&web=1&e=acnxZf>