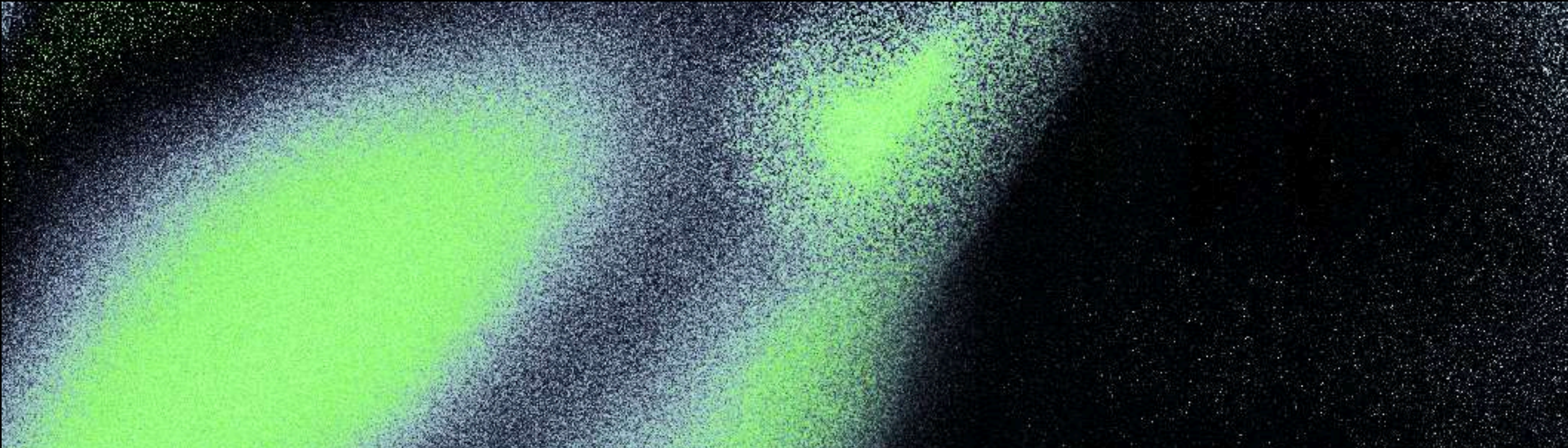
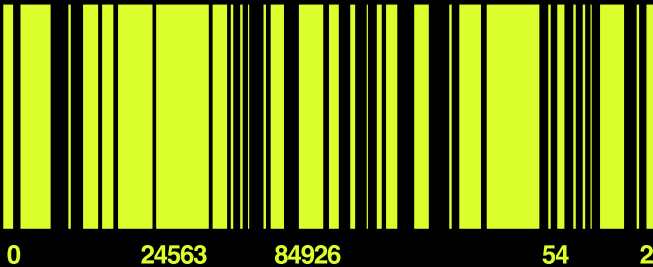
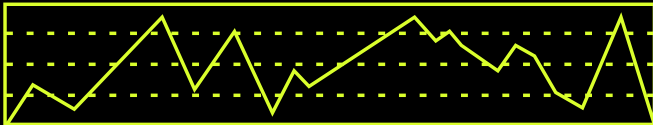
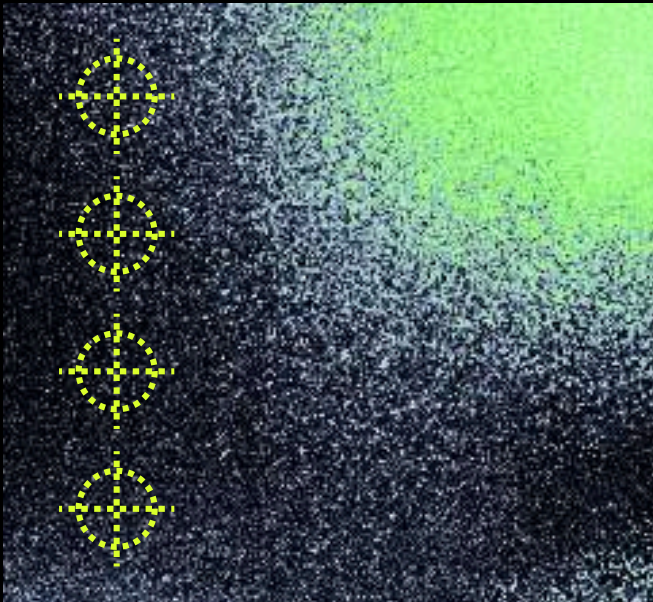
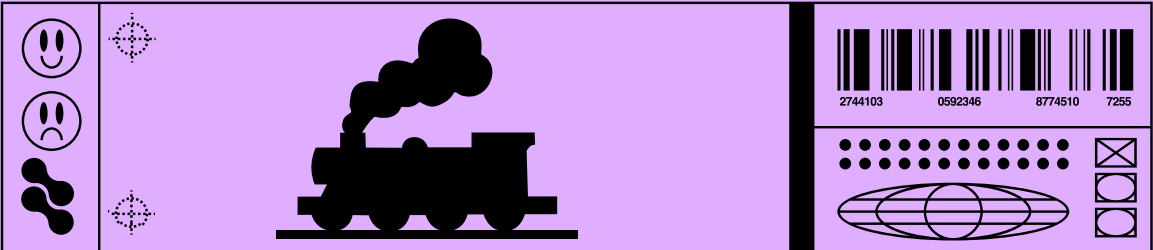


# PROJECT JAVA

## PRESENTATION

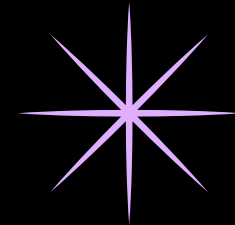
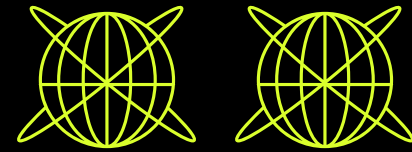


**TRAINBOOKING.COM**

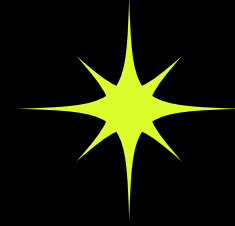




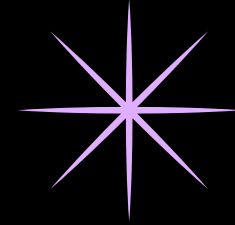
# AGENDA



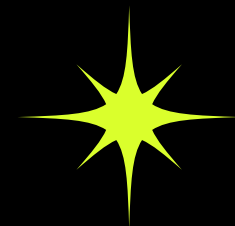
Topic 1 : Introduction



Topic 2 : User Interface

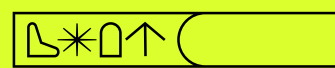
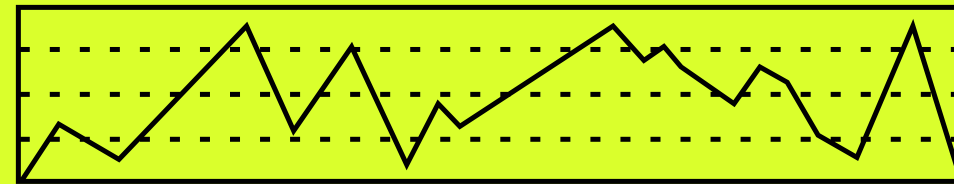


Topic 3 : Algorithm

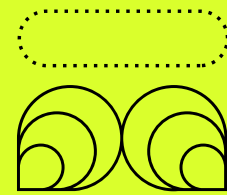
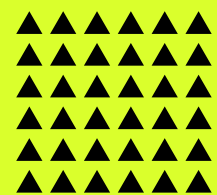


Topic 4 : Demo

## TOPICS COVERED



53°24'19.2"N 2°14'28.5"E  
53°24'19.2"S 42°21'34.7"W



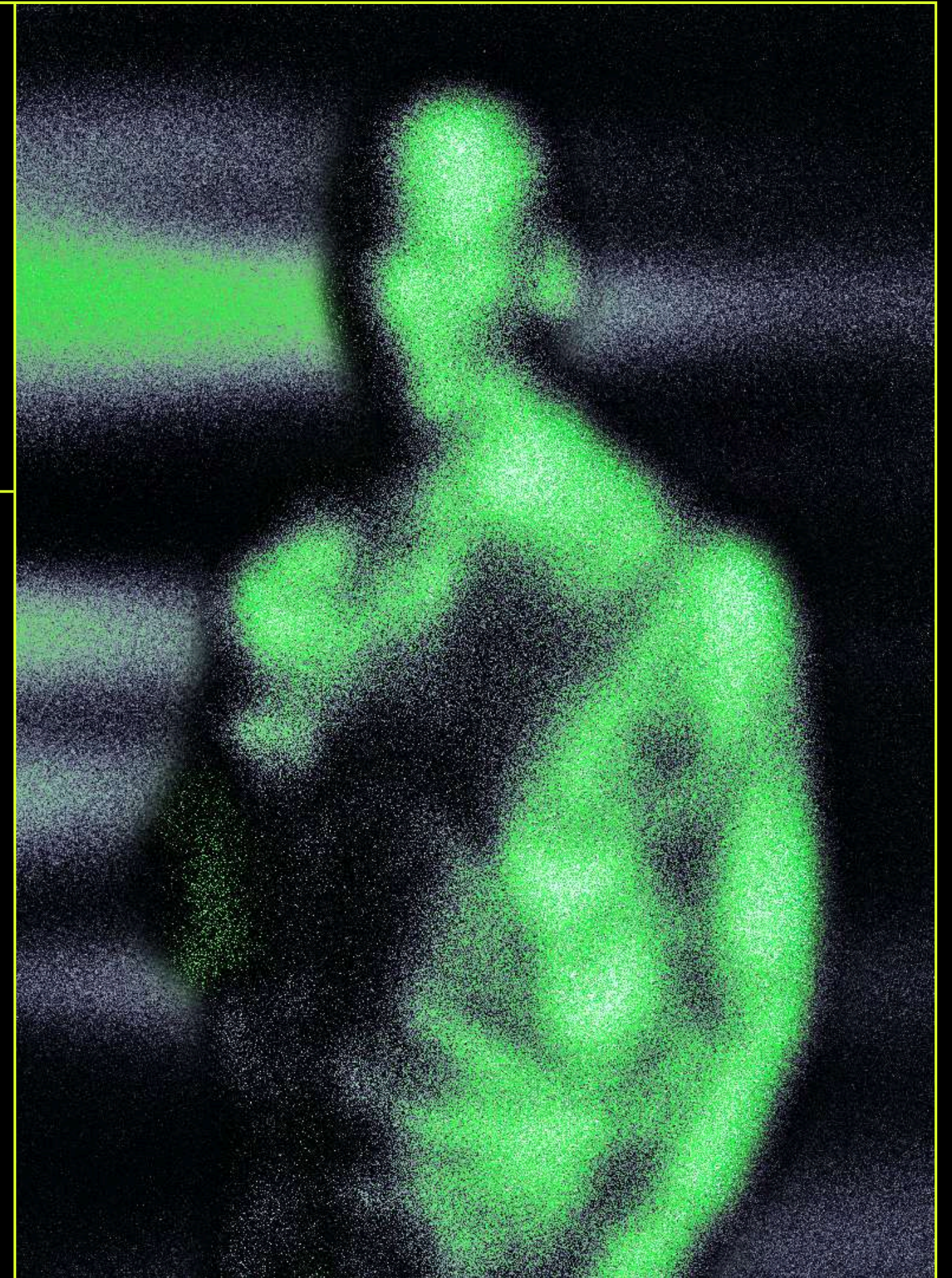
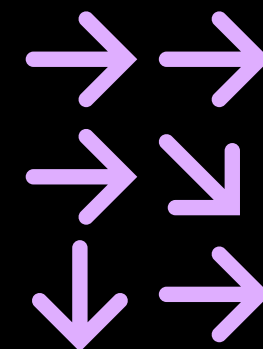
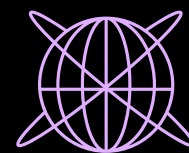
0000111010100101001011...011110101010001010





# WHY DID YOU DO THIS PROJECT?

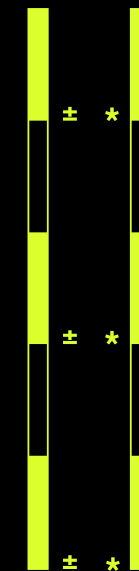
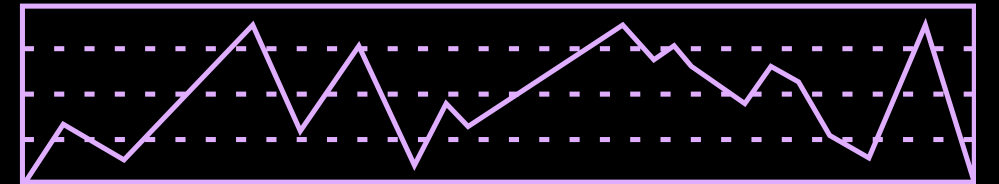
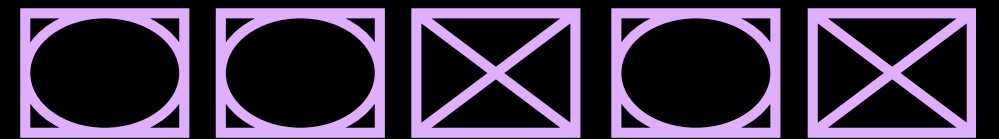
เหตุผลที่เลือกทำ Project นี้เพราะเป็นคนชอบ  
เดินทาง เป็นคนชอบเที่ยว เลยอยากจะทำ  
โปรแกรมเกี่ยวกับการท่องเที่ยว และตอนเด็ก  
เป็นคนที่ชอบรถไฟมากๆ จึงเกิดเป็นโปรเจกต์นี้  
TrainBooking.com ขึ้นมาเป็นเว็บที่รวบรวม  
จังหวัดท่องเที่ยวของประเทศไทย





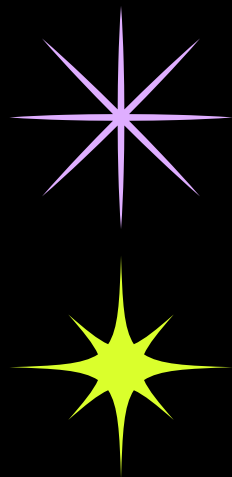
# USER INTERFACE

# FACE



# USER INTERFACE

## PROFILEPAGE



TrainBooking.com

**WELCOME TO TRAINBOOKING.COM**

**1.Username**


**2.Phone Number**

**3.Email**

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

**5.Select Destination Province Station**  
-- Selected --

**6.Price per person**

 คิดถึงตัวแพงคิดถึงเรา รวดเร็วทันใจจับใจ ตัวใครตัวมัน

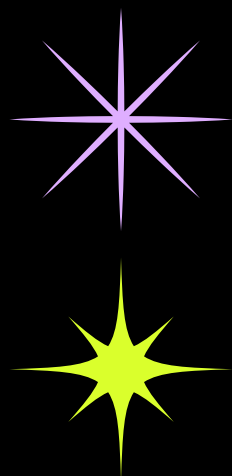
**Booking!**

หน้า ProfilePage ทำหน้าที่รับข้อมูลจากผู้ใช้งาน ชื่อ เบอร์  
โทร สถานีต้นทาง สถานีปลายทาง และแสดงราคาตัว



# USER INTERFACE

## BOOKINGPAGE



TrainBooking.com

**Name : Kanakorn Wongchaiya**  
**Phone Number : 0631911583**  
**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**

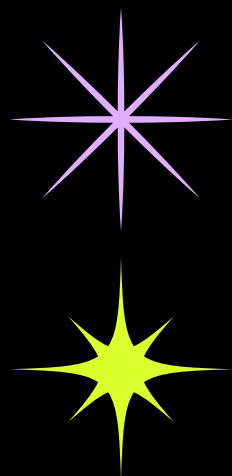


**Back** **Payment**

หน้า BookingPage ทำหน้าที่ให้ผู้ใช้จองที่นั่ง เลือกที่นั่งที่ต้องการและแสดงข้อมูลของผู้ใช้

# USER INTERFACE

## PAYMENTPAGE




TrainBooking.com

<b>Name :</b> Kanakorn Wongchaiya	<b>Departure Station :</b> Bangkok
<b>Phone Number :</b> 0631911583	<b>Destination Station :</b> Chiang Rai
<b>Email :</b> Kanakorn.w@ku.th	<b>Price :</b> 900 Baht/Person
	<b>Selected Seats :</b> [D4, C4]

<b>Total Price:</b>	<b>1800 Baht</b>
---------------------	------------------

	<b>0653129297 Kasikorn</b> <b>Name: Kanakorn Wongchaiya</b>
---	--

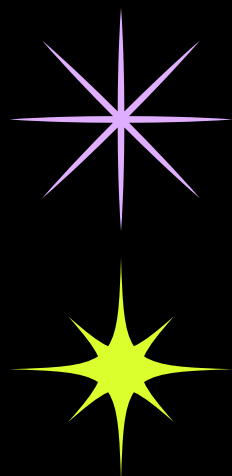
  

<b>Back</b>	<b>Receipt</b>
-------------	----------------

หน้า PAYMENTPAGE ทำหน้าที่รวบรวมข้อมูลของผู้ใช้  
ทั้งหมดและที่นั่งที่ทำการจอง และแสดงช่องทางการชำระเงิน

# USER INTERFACE

# RECEIPT



RecieptTrainBookin...

**Name : Kanakorn Wongchaiya**

**Phone Number : 0631911583**

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

**Departure Station : Bangkok**

**Destination Station : Chiang Rai**

**Price : 900 Baht/Person**

**Selected Seats : [D4, C4]**

**Total Price: 1800 Baht**

**Exit**

หน้า RECEIPT ทำหน้าที่สรุปคำสั่งซื้อทั้งหมดแสดงข้อมูล  
ออกมาเป็นใบเสร็จ/ตัว



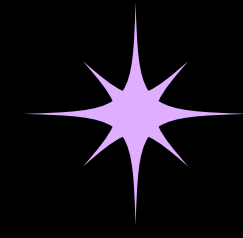
# ALGORITHM MAINAPP ✨

```
public class MainApp {  
    Run | Debug  
    public static void main(String[] args) {  
        SwingUtilities.invokeLater(new Runnable() {  
            // Event Dispatch Thread (EDT)  
            @Override  
            public void run(){  
                AppWindow app = new AppWindow();  
                app.setLocationRelativeTo(c:null);  
                app.setVisible(b:true);  
            }  
        });  
    }  
}
```

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



# ALGORITHM APPWINDOW



```
public AppWindow() {
    setTitle(title:"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();
}

public void showReceiptPage(Data data, Set<String> selectedSeats ,int totalPrice) {
    receipt = new Receipt(this, data, selectedSeats , totalPrice);
    setContentPane(receipt);
    setTitle(title:"RecieptTrainBooking.com");
    setSize(width:300,height:430);
    setLocationRelativeTo(c:null);
    setResizable(resizable:false);
    setDefaultCloseOperation(EXIT_ON_CLOSE);
    revalidate();
}

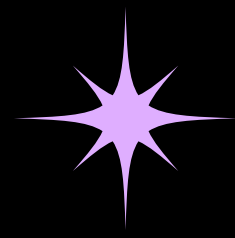
public void showProfilePage() {
    setContentPane(profilePage);
    revalidate();
}

public void showBookingPage() {
    setContentPane(bookingPage);
    revalidate();
}
```

คลาส AppWindow เป็นคลาสหลักที่ใช้สำหรับการจัดการหน้าต่างและการเปลี่ยนแปลงหน้าต่างในแอปพลิเคชัน มีการสร้างอ็อบเจกต์ รับส่งข้อมูล และประกาศเมทอดต่างที่ใช้ในการเปลี่ยนหน้าต่างของโปรแกรม การทำงานของโปรแกรม มีการสืบทอดมาจากคลาส JFrame โดยการเปลี่ยนหน้ากระดาษจะเป็นการใช้เมทอด setContentPane ในการเปลี่ยนหน้าต่าง



# ALGORITHM DATA



```
// 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;
}
```

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

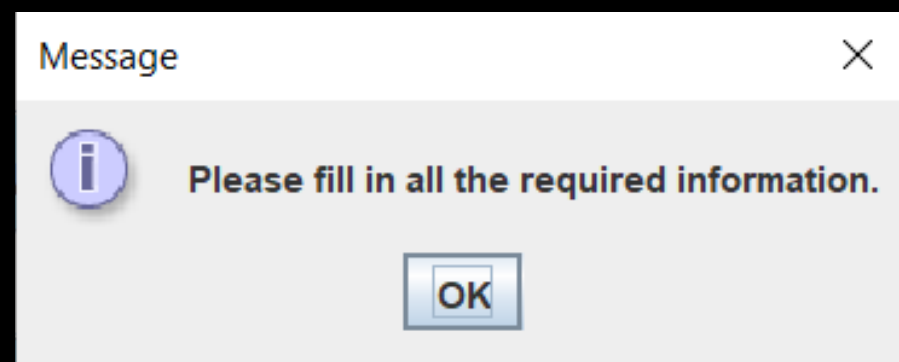


# ALGORITHM PROFILEPAGE ✨

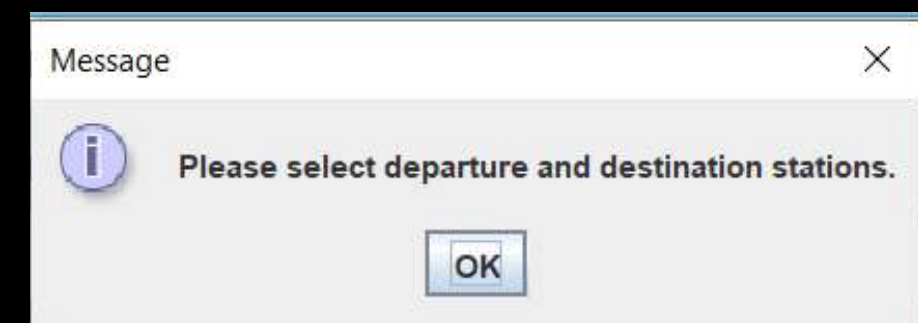
```
//สร้างเมทอดให้เช็คค่าช่องที่รับค่าเข้ามาเป็นช่องว่างหรือไม่และใน ComboBox มีการเลือกสถานีต้นทางและสถานีปลายทางหรือไม่
private boolean isValidData() {
    // ตรวจสอบว่าข้อมูลในช่อง Name, PhoneNumber, Email ถูกป้อนหรือไม่
    if (usernameTextField.getText().isEmpty() || phoneNumberTextField.getText().isEmpty() || emailTextField.getText().isEmpty()) {
        JOptionPane.showMessageDialog(app, message:"Please fill in all the required information.");
        return false;
    }
    // ตรวจสอบว่า ComboBox สถานีปลายทางและสถานีต้นทางถูกเลือกหรือไม่
    if (departureComboBox.getSelectedItem().equals(obj:"-- Selected --") || destinationComboBox.getSelectedItem().equals(obj:"-- Selected --")) {
        JOptionPane.showMessageDialog(app, message:"Please select departure and destination stations.");
        return false;
    }
    //ถ้าพบว่ามีข้อความ
    return true;
}
```

## isValidData()

เป็นเมทอดสำหรับตรวจสอบความถูกต้องของข้อมูลที่ใช้กรอกเข้ามา โดยตรวจสอบว่าช่องใดมีข้อมูลที่ถูกป้อนหรือไม่ และว่าผู้ใช้เลือกสถานีต้นทางและปลายทางหรือไม่ หากข้อมูลไม่ครบถ้วนหรือไม่ถูกต้อง โปรแกรมจะแสดงข้อความเตือนให้ผู้ใช้กรอกข้อมูลให้ครบถ้วนและเลือกสถานีที่ต้องการให้ถูกต้อง

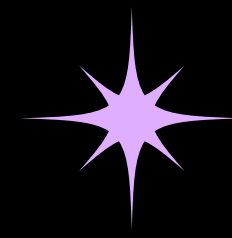


เมื่อกดปุ่ม Booking โดยไม่กรอกข้อมูลอะไรเข้าไป  
JOptionPane จะส่งข้อความว่าให้กรอกข้อมูล



เมื่อกดปุ่ม Booking โดยไม่เลือกสถานีต้นทางและ  
สถานีปลายทาง JOptionPane ก็จะแจ้งข้อความว่า  
ให้เลือกสถานีต้นทางและปลายทางด้วย

# ALGORITHM BOOKINGPAGE



```
//สร้างแผนผังที่นั่งทางซ้ายของจอ
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:0));
    for (char row : rows) {
        JButton seatButton = new JButton(row + String.valueOf(i));
        seatButton.addActionListener(new SeatButtonHandler()); // เพิ่ม ActionListener ในปุ่มที่นั่ง
        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))); // เพิ่มพื้นที่ว่างระหว่างปุ่ม
    }

    leftseat.add(rowPanel);
    leftseat.setBackground(new Color(r:144,g:238,b:144));
}
seat.add(leftseat, BorderLayout.CENTER);

//เพิ่ม Action ให้ปุ่มที่นั่ง
private class SeatButtonHandler implements ActionListener {
    @Override
    public void actionPerformed(ActionEvent e) {
        JButton button = (JButton) e.getSource(); //ดึงอ็อบเจกต์ JButton ที่ทำให้เกิด ActionEvent โดยใช้เมทอด getSource() ของอ็อบเจกต์ ActionEvent ที่ถูกส่งมา.
        String seat = button.getText(); //ดึงข้อความที่ปุ่มมีอยู่ ซึ่งในที่นี้คือตำแหน่งหมายเลขที่นั่ง โดยใช้เมทอด getText() ของอ็อบเจกต์ JButton.
        //สร้างเงื่อนไขเพื่อตรวจสอบว่าที่นั่งที่ผู้ใช้คลิกเลือกอยู่ในเซตของที่นั่งที่ถูกเลือกหรือไม่.
        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
        }
    }
}
```

มีการสร้างแผนผังที่นั่งเป็นแบบปุ่มกดและมีการ  
addActionListener ที่ถูกเพิ่มให้กับปุ่มที่นั่งที่แสดงบนหน้าจอ  
เมื่อผู้ใช้คลิกที่ปุ่มที่นั่ง ปุ่มนั้นจะเปลี่ยนเป็นสีม่วงเพื่อบอก  
สถานะว่ากำลังเลือกที่นั่งตรงนั้นอยู่ และมีการดึงข้อความที่อยู่  
บนปุ่มที่นั่งที่เลือกเพื่อนำข้อมูลไปใช้ต่อไป



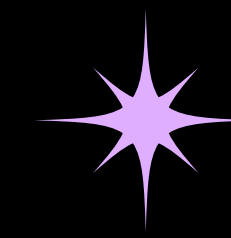
# ALGORITHM PAYMENTPAGE ✨

```
//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);
    }
}
```

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

# ALGORITHM RECEIPT



```
//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
        }
    }
}
```

เมื่อกดปุ่ม Exit จะมี JOptionPane ขึ้นมาเพื่อให้กดยืนยันอีกครั้ง



0 24563 84926 54 2

MODEL NO.  
\*\*738\*312/1

