系統分析與設計

系統名稱:飯店訂房系統

指導老師:陳仲儼

組員:
111403023 資管三 葉東融
111403512 資管三 李權治
111403541 資管三 李介翔
111403038 資管三 蔡東哲
111403518 資管三 吳冠毅

目錄

系統功能與介紹

這份系統主要是為飯店和旅館管理員和顧客訂房所製作, 內部流程相對簡單, 提供兩個主要選項:

1. 查詢房間:用戶輸入住退房日期以查詢可用房間。
2. 訂購房間:可以輸入顧客資訊訂購房間。

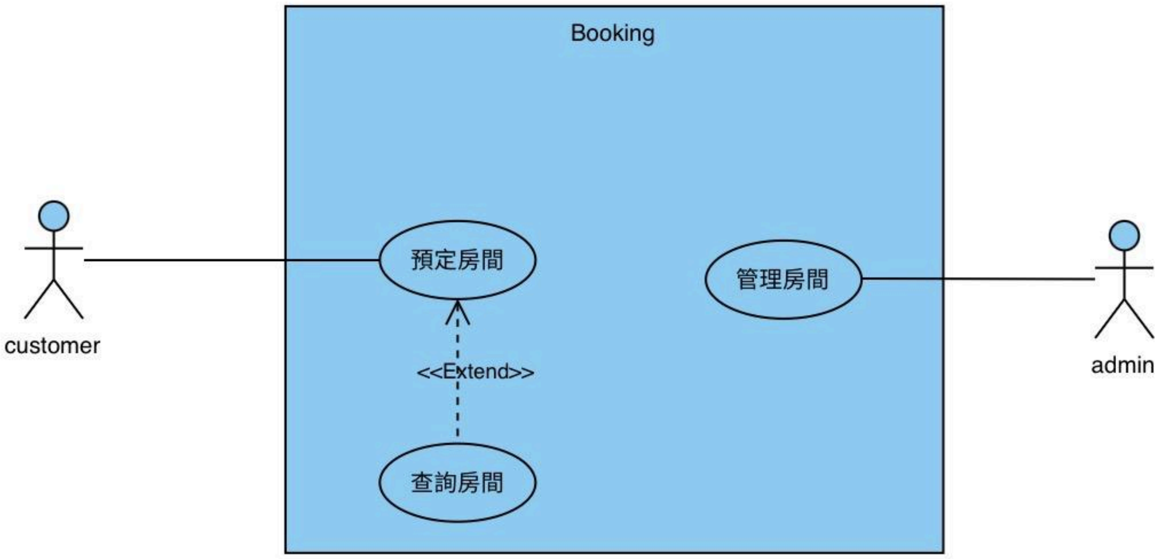
而在顧客查詢完之後可以選擇訂房, 訂房也可以同時訂好幾間房間。填寫完資料之後會回到主頁面, 並把顧客的資訊存在檔案裏, 然後房間的資訊則是存在另一個檔案內。員工在之後也可以透過查詢系統看目前所擁有的訂單。

安裝說明:
本程式使用的語言爲JAVA, 語言版本則是JAVA 19, 所需要的檔案爲

1. HotelBookingApp.java 主要程式, 開啟即可使用
2. booked_rooms.txt 已被訂購的房間的資料
3. customers.txt 顧客的資料

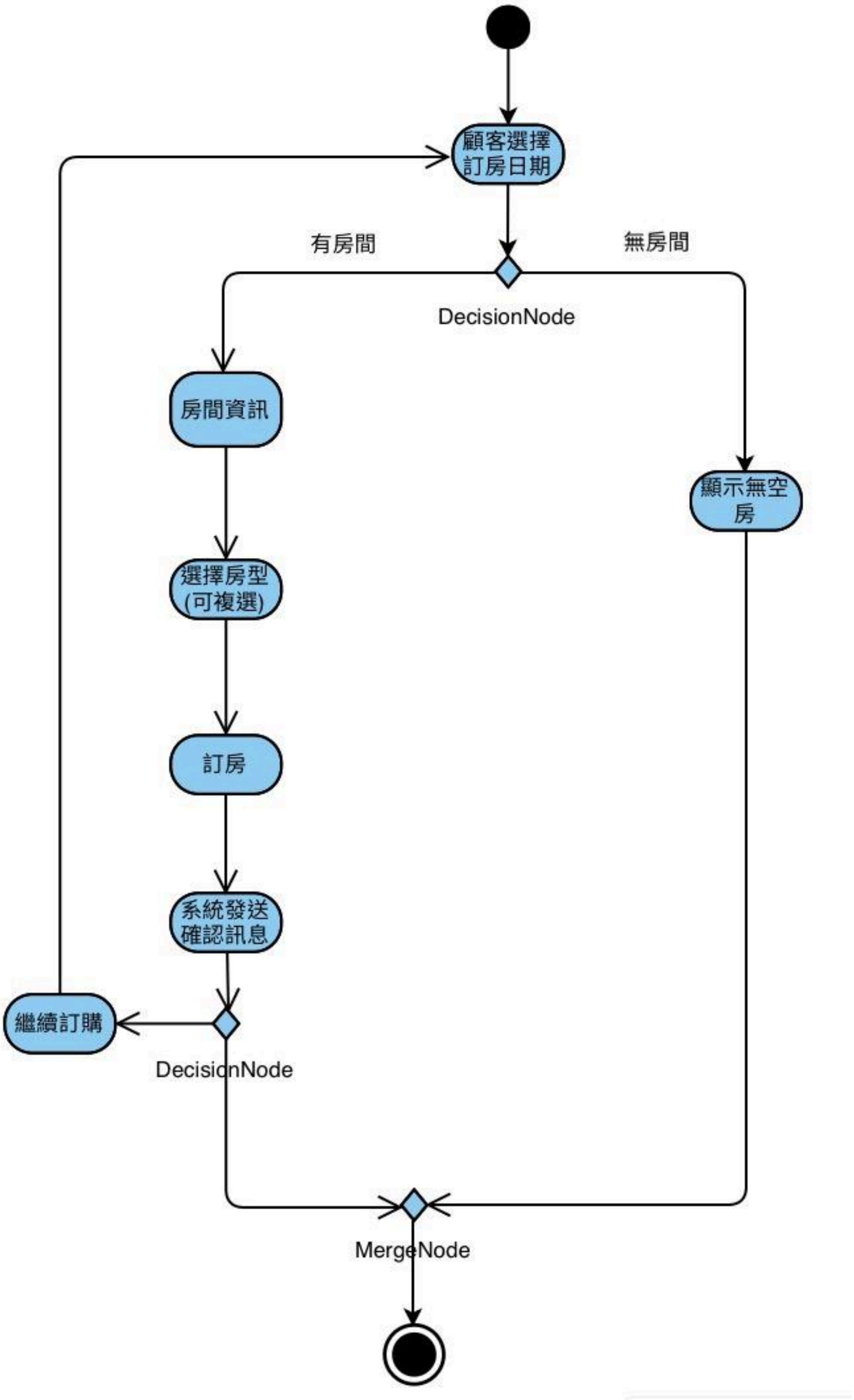
當安裝時把壓縮檔全部解壓縮後, 請把上述的txt檔放入java檔專案的根目錄, 確保執行順利。

Use case diagram

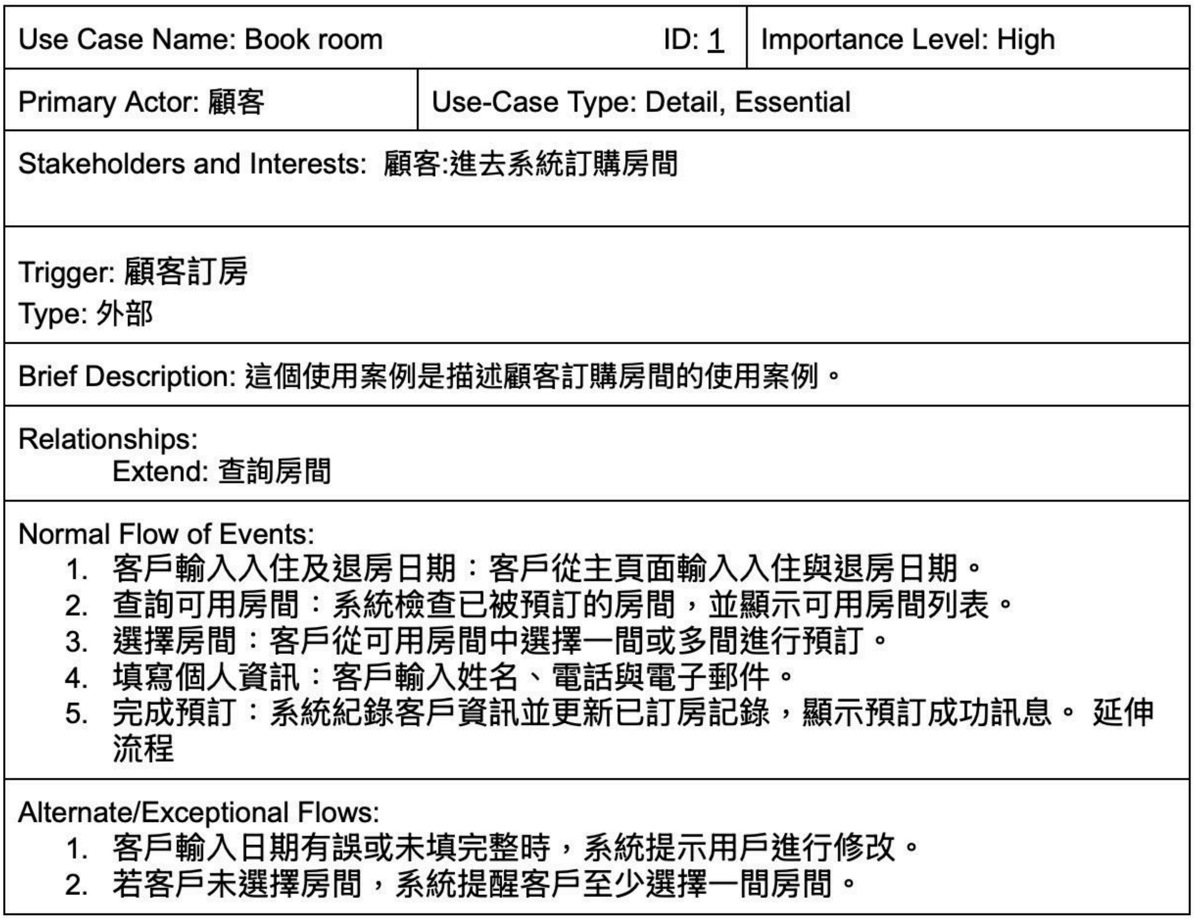


核心Use case: 預定房間

Activity diagram

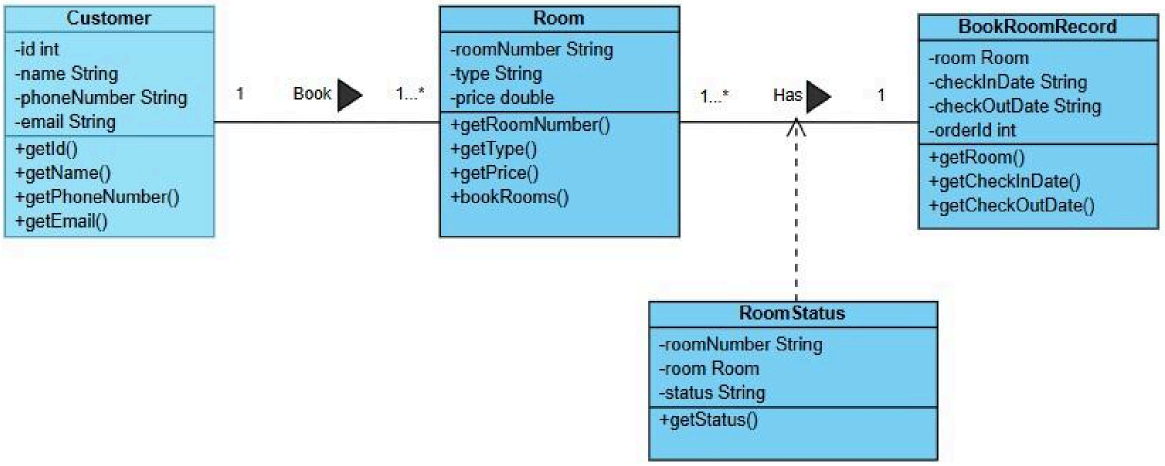


Use case description

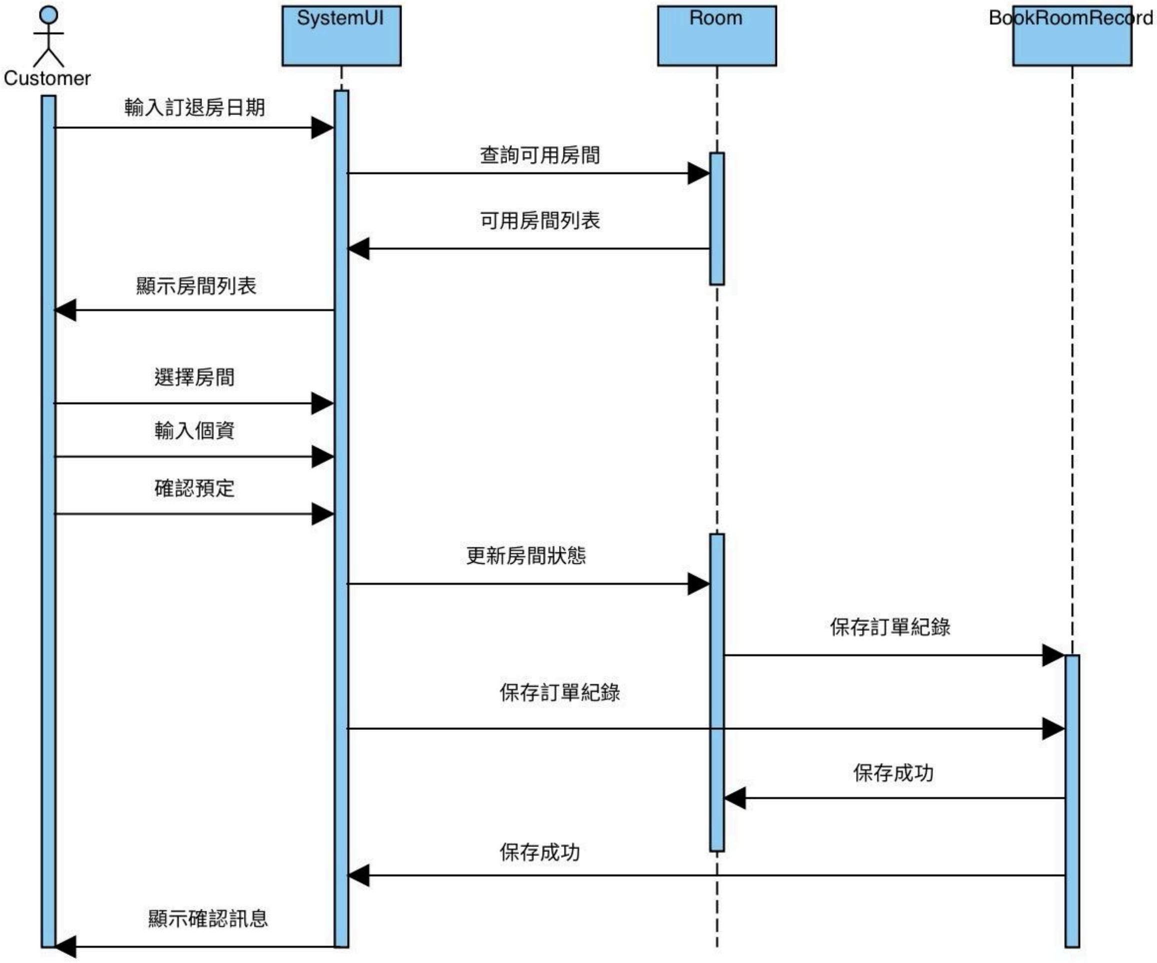


<table>
<tr><td colspan="2">Use Case Name: Book room</td><td>ID: 1</td><td>Importance Level: High</td></tr>
<tr><td>Primary Actor: 顧客</td><td colspan="3">Use-Case Type: Detail, Essential</td></tr>
<tr><td colspan="4">Stakeholders and Interests: 顧客:進去系統訂購房間</td></tr>
<tr><td colspan="4">Trigger: 顧客訂房
Type: 外部</td></tr>
<tr><td colspan="4">Brief Description: 這個使用案例是描述顧客訂購房間的使用案例。</td></tr>
<tr><td colspan="4">Relationships:
Extend: 查詢房間</td></tr>
<tr><td colspan="4">Normal Flow of Events:
1. 客戶輸入入住及退房日期：客戶從主頁面輸入入住與退房日期。
2. 查詢可用房間：系統檢查已被預訂的房間，並顯示可用房間列表。
3. 選擇房間：客戶從可用房間中選擇一間或多間進行預訂。
4. 填寫個人資訊：客戶輸入姓名、電話與電子郵件。
5. 完成預訂：系統紀錄客戶資訊並更新已訂房記錄，顯示預訂成功訊息。 延伸流程</td></tr>
<tr><td colspan="4">Alternate/Exceptional Flows:
1. 客戶輸入日期有誤或未填完整時，系統提示用戶進行修改。
2. 若客戶未選擇房間，系統提醒客戶至少選擇一間房間。</td></tr>
</table>

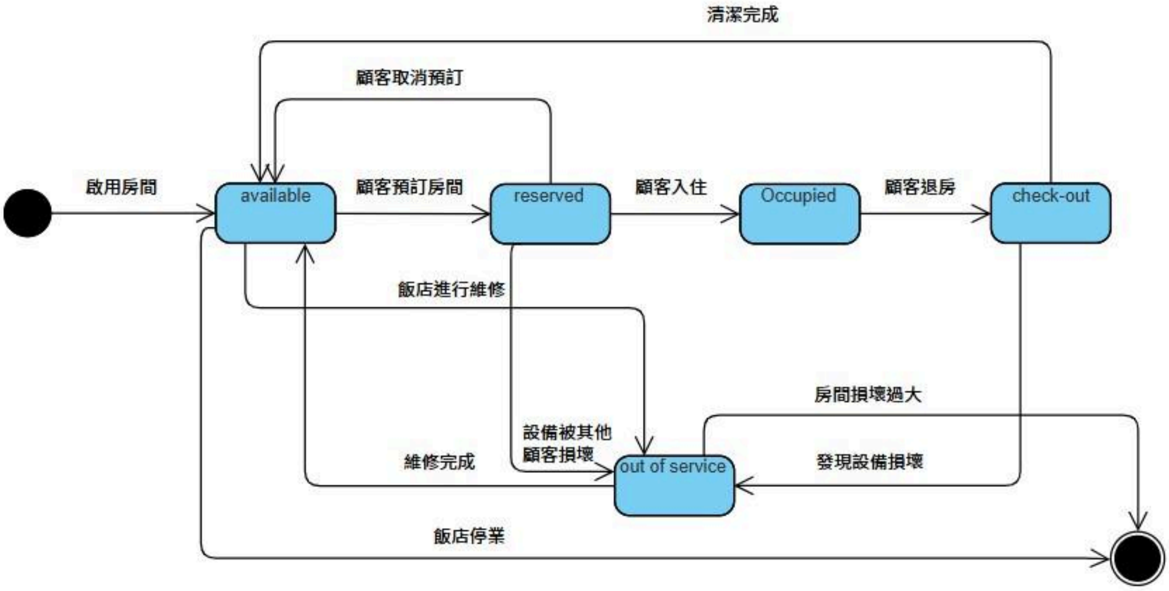
Class diagram



Sequence diagram



State machine diagram



程式碼

引用Java的Library

```
import javax.swing.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.io.*;
import java.util.ArrayList;
import java.util.List;
```

主程式:存取過去的資料

```
public class HotelBookingApp {
   private static List<Room> availableRooms = new ArrayList<>();
  public static List<BookedRoomRecord> bookedRoomRecords = new ArrayList<>();
  public static List<Customer> customers = new ArrayList<>();
  public static List<RoomStatus> roomStatusList = new ArrayList<>();
  public static void main(String[] args) {
    loadRoomData();
    loadCustomerData();
    SwingUtilities.invokeLater(() -> new WelcomePage().setVisible(true));
  }
  private static void loadRoomData() {
    // 房間主鍵為房號
    availableRooms.add(new Room("101", "單人房", 100));
    availableRooms.add(new Room("102", "單人房", 100));
    availableRooms.add(new Room("103", "單人房", 100));
    availableRooms.add(new Room("104", "雙人房", 200));
    availableRooms.add(new Room("105", "雙人房", 200));
    availableRooms.add(new Room("106", "豪華套房", 300));
    availableRooms.add(new Room("107", "總統套房", 500));
    // 從booked_rooms.txt讀取已被訂的房間
    try (BufferedReader reader = new BufferedReader(new
FileReader("booked_rooms.txt"))) {
      String line;
      while ((line = reader.readLine()) != null) {
        if (line.trim().isEmpty()) continue;
        String[] parts = line.split(",");
        if (parts.length < 5) {
          System.out.println("Invalid line in booked_rooms.txt: " + line);
          continue;
        }
        String roomNumber = parts[0];
        String type = parts[1];
        double price = Double.parseDouble(parts[2]);
        String checkInDate = parts[3];
        String checkOutDate = parts[4];
        Room room = new Room(roomNumber, type, price);
        bookedRoomRecords.add(new
BookedRoomRecord(BookedRoomRecord.idCounter, room, checkInDate, checkOutDate));
        roomStatusList.add(new RoomStatus("reserved", roomNumber,
BookedRoomRecord.idCounter));
      }
    } catch (FileNotFoundException e) {
      System.out.println("No booked rooms file found. Starting fresh.");
    } catch (IOException e) {
      e.printStackTrace();
    }
  }
  private static void loadCustomerData() {
    try (BufferedReader reader = new BufferedReader(new FileReader("customers.txt"))) {
      String line;
      while ((line = reader.readLine()) != null) {
        if (line.trim().isEmpty()) continue;
        String[] parts = line.split(",");
        if (parts.length < 4) {
          System.out.println("Invalid line in customers.txt: " + line);
          continue;
        }
        int id = Integer.parseInt(parts[0]);
        String name = parts[1];
        String phoneNumber = parts[2];
        String email = parts[3];
        customers.add(new Customer(id, name, phoneNumber, email));
      }
    } catch (FileNotFoundException e) {
      System.out.println("No customer file found. Starting fresh.");
    } catch (IOException e) {
      e.printStackTrace();
    }
  }
  public static void saveCustomerData() {
    try (BufferedWriter writer = new BufferedWriter(new FileWriter("customers.txt"))) {
      for (Customer customer : customers) {
        writer.write(customer.getId() + "," + customer.getName() + "," +
            customer.getPhoneNumber() + "," + customer.getEmail() + "\n");
      }
    } catch (IOException e) {
      e.printStackTrace();
    }
  }
  public static void saveBookedRooms() {
    try (BufferedWriter writer = new BufferedWriter(new FileWriter("booked_rooms.txt")))
{
      for (BookedRoomRecord record : bookedRoomRecords) {
        Room room = record.getRoom();
        writer.write(room.getRoomNumber() + "," + room.getType() + "," +
            room.getPrice() + "," + record.getCheckInDate() + "," +
record.getCheckOutDate() + "\n");
      }
    } catch (IOException e) {
      e.printStackTrace();
    }
  }
  public static List<Room> getAvailableRooms(String targetCheckInDate, String
targetCheckOutDate) {
    List<Room> filteredRooms = new ArrayList<>(availableRooms);
    for (RoomStatus roomStatus : roomStatusList) {
      if (!roomStatus.getStatus().equals("available")) {
        int orderId = roomStatus.getOrderId();
        // 確保 orderId 有效
        BookedRoomRecord record = bookedRoomRecords.get(orderId - 1);
        String checkInDate = record.getCheckInDate();
        String checkOutDate = record.getCheckOutDate();
        if (isDateOverlap(targetCheckInDate, targetCheckOutDate, checkInDate,
checkOutDate)) {
          filteredRooms.removeIf(room ->
room.getRoomNumber().equals(roomStatus.getRoomNumber()));
        }
      }
    }
    return filteredRooms;
  }
  private static boolean isDateOverlap(String start1, String end1, String start2, String end2) {
    return !(end1.compareTo(start2) <= 0 || start1.compareTo(end2) >= 0);
  }
}
```

Customer：存入顧客的資訊，紀錄訂房資訊

```
class Customer {
  private int id;
  private String name;
  private String phoneNumber;
  private String email;
  public Customer(int id, String name, String phoneNumber, String email) {
    this.id = id;
    this.name = name;
    this.phoneNumber = phoneNumber;
    this.email = email;
  }
  public int getId() {
    return id;
  }
  public String getName() {
    return name;
  }
  public String getPhoneNumber() {
    return phoneNumber;
  }
  public String getEmail() {
    return email;
  }

  public static void bookRooms(List<Room> rooms, String checkInDate, String
checkOutDate) {
    for (Room room : rooms) {
        HotelBooking.bookedRoomRecords.add(new
BookedRoomRecord(BookedRoomRecord.idCounter, room, checkInDate, checkOutDate));
        RoomStatus roomStatus = new RoomStatus("reserved", room.getRoomNumber(),
BookedRoomRecord.idCounter);
    }
    HotelBooking.saveBookedRooms();
  }

  @Override
  public String toString() {
    return id + " - " + name + " (" + phoneNumber + ", " + email + ")";
  }
}
```

BookedRoomRecord：紀錄訂單資訊

```
class BookedRoomRecord {
  public static int idCounter = 0;
  private int recordId;
  private Room room;
  private String checkInDate;
  private String checkOutDate;
  public BookedRoomRecord(int recordId, Room room, String checkInDate, String
checkOutDate) {
     this.recordId = idCounter;
     this.room = room;
     this.checkInDate = checkInDate;
     this.checkOutDate = checkOutDate;
     idCounter++;
  }
  public Room getRoom() {
     return room;
  }
  public String getCheckInDate() {
     return checkInDate;
  }
  public String getCheckOutDate() {
     return checkOutDate;
  }

  public int getId() {
        return recordId;
  }
}
```

Room：紀錄房間資訊

```
class Room {
  private String roomNumber;
  private String type;
  private double price;
  public Room(String roomNumber, String type, double price) {
     this.roomNumber = roomNumber;
     this.type = type;
     this.price = price;
  }
  public String getRoomNumber() {
     return roomNumber;
  }
  public String getType() {
     return type;
  }
  public double getPrice() {
     return price;
  }
  @Override
  public String toString() {
     return roomNumber + " (" + type + ", $" + price + ")";
  }
}
```

RoomStatus：紀錄每天的房間狀態

```
class RoomStatus {
private String status;
  private String roomNumber;
  private int orderId;
  public RoomStatus(String status, String roomNumber, int orderId) {
    this.status = status;
    this.roomNumber = roomNumber;
    this.orderId = orderId;
  }
  public String getStatus() {
    return status;
  }
  public String getRoomNumber() {
    return roomNumber;
  }
  public int getOrderId() {
    return orderId;
  }
}
```

首頁UI介面

```
class WelcomePage extends JFrame {
    public WelcomePage() {
        setTitle("SAD旅館");
        setSize(700, 400);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setLayout(null);
        JLabel welcomeLabel = new JLabel("歡迎來到SAD旅館，請輸入入退住日期");
        welcomeLabel.setBounds(250, 30, 300, 30);
        add(welcomeLabel);
        JLabel checkInLabel = new JLabel("入住日期 (yyyy-MM-dd):");
        checkInLabel.setBounds(200, 100, 150, 30);
        add(checkInLabel);
        JTextField checkInField = new JTextField();
        checkInField.setBounds(400, 100, 100, 30);
        add(checkInField);
        JLabel checkOutLabel = new JLabel("退房日期 (yyyy-MM-dd):");
        checkOutLabel.setBounds(200, 150, 150, 30);
        add(checkOutLabel);
        JTextField checkOutField = new JTextField();
        checkOutField.setBounds(400, 150, 100, 30);
        add(checkOutField);
        JButton searchButton = new JButton("查詢房間");
        searchButton.setBounds(150, 250, 150, 30);
        add(searchButton);
        JButton viewCustomersButton = new JButton("查看訂單");
        viewCustomersButton.setBounds(400, 250, 150, 30);
        add(viewCustomersButton);
        searchButton.addActionListener(e -> {
            String checkInDate = checkInField.getText();
            String checkOutDate = checkOutField.getText();
            String datePattern = "\\d{4}-\\d{2}-\\d{2}"; // 日期格式 yyyy-MM-dd

            if (checkInDate.isEmpty() || checkOutDate.isEmpty()) {
                JOptionPane.showMessageDialog(this, "請輸入入退住日期！", "提示", JOptionPane.WARNING_MESSAGE);
                return;
            }

            if (!checkInDate.matches(datePattern) || !checkOutDate.matches(datePattern)) {
                JOptionPane.showMessageDialog(this, "日期格式錯誤！請使用 yyyy-MM-dd 格式。", "錯誤", JOptionPane.ERROR_MESSAGE);
                return;
            }
            new RoomSelectionPage(checkInDate, checkOutDate).setVisible(true);
            dispose();
        });
        viewCustomersButton.addActionListener(e -> {
            new Order().setVisible(true);
        });
    }
}
```

選擇房間UI介面

```
class RoomSelectionPage extends JFrame {
  private List<Room> selectedRooms = new ArrayList<>();
    public RoomSelectionPage(String checkInDate, String checkOutDate) {
      setTitle("請選擇房間(可複選)");
      setSize(600, 400);
      setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
      setLayout(null);
      JLabel titleLabel = new JLabel("可用房間列表");
      titleLabel.setBounds(250, 20, 200, 30);
      add(titleLabel);
      // 根據用戶輸入的日期範圍, 獲取可用房間
      List<Room> availableRooms = HotelBookingApp.getAvailableRooms(checkInDate,
checkOutDate);
      JPanel checkboxPanel = new JPanel();
      checkboxPanel.setBounds(50, 60, 500, 200);
      checkboxPanel.setLayout(new BoxLayout(checkboxPanel, BoxLayout.Y_AXIS));
      JCheckBox[] roomCheckboxes = new JCheckBox[availableRooms.size()];
      for (int i = 0; i < roomCheckboxes.length; i++) {
        Room room = availableRooms.get(i);
        roomCheckboxes[i] = new JCheckBox(room.toString());
        checkboxPanel.add(roomCheckboxes[i]);
      }
      JScrollPane scrollPane = new JScrollPane(checkboxPanel);
      scrollPane.setBounds(50, 60, 500, 200);
      add(scrollPane);
      JButton nextButton = new JButton("下一步");
      nextButton.setBounds(250, 300, 100, 30);
      add(nextButton);
      nextButton.addActionListener(e -> {
        selectedRooms.clear();
        for (int i = 0; i < roomCheckboxes.length; i++) {
          if (roomCheckboxes[i].isSelected()) {
            selectedRooms.add(availableRooms.get(i));
          }
        }
        if (selectedRooms.isEmpty()) {
          JOptionPane.showMessageDialog(this, "請選擇至少一間房間！");
        } else {
          new CustomerInfoPage(checkInDate, checkOutDate,
selectedRooms).setVisible(true);
          dispose();
        }
      });
    }
}
```

顧客輸入資訊UI介面

```
class CustomerInfoPage extends JFrame {
    public CustomerInfoPage(String checkInDate, String checkOutDate, List<Room>
selectedRooms) {
        setTitle("請填入您的資訊");
        setSize(400, 300);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setLayout(null);
        JLabel nameLabel = new JLabel("姓名:");
        nameLabel.setBounds(50, 50, 100, 30);
        add(nameLabel);
        JTextField nameField = new JTextField();
        nameField.setBounds(150, 50, 200, 30);
        add(nameField);
        JLabel phoneLabel = new JLabel("電話:");
        phoneLabel.setBounds(50, 100, 100, 30);
        add(phoneLabel);
        JTextField phoneField = new JTextField();
        phoneField.setBounds(150, 100, 200, 30);
        add(phoneField);
        JLabel emailLabel = new JLabel("電子郵件:");
        emailLabel.setBounds(50, 150, 100, 30);
        add(emailLabel);
        JTextField emailField = new JTextField();
        emailField.setBounds(150, 150, 200, 30);
        add(emailField);
        JButton confirmButton = new JButton("確認");
        confirmButton.setBounds(150, 200, 100, 30);
        add(confirmButton);
        confirmButton.addActionListener(e -> {
            String name = nameField.getText();
            String phone = phoneField.getText();
            String email = emailField.getText();

            String phonePattern = "\\d{10}"; // 電話號碼格式為 10 位數字
            String emailPattern = "^[\\w._%+-]+@[\\w.-]+\\.[a-zA-Z]{2,}$"; // 簡單的電子郵件
格式驗證

            if (name.isEmpty() || phone.isEmpty() || email.isEmpty()) {
                JOptionPane.showMessageDialog(this, "請填寫完整的客戶資訊！", "錯誤",
JOptionPane.WARNING_MESSAGE);
                return;
            }
            if (!phone.matches(phonePattern)) {
                JOptionPane.showMessageDialog(this, "電話號碼格式錯誤！請輸入 10 位數字。
", "錯誤", JOptionPane.ERROR_MESSAGE);
                return;
            }

            if (!email.matches(emailPattern)) {
                JOptionPane.showMessageDialog(this, "電子郵件格式錯誤！請輸入有效的電子
郵件地址。", "錯誤", JOptionPane.ERROR_MESSAGE);
                return;
            }

            int newId = HotelBookingApp.customers.size() + 1;
            Customer newCustomer = new Customer(newId, name, phone, email);
            HotelBookingApp.customers.add(newCustomer);
            HotelBookingApp.saveCustomerData();

            // Book selected rooms
            HotelBookingApp.bookRooms(selectedRooms, checkInDate, checkOutDate);

            JOptionPane.showMessageDialog(this, "預訂成功！SAD旅館期待你的到來!");
            new WelcomePage().setVisible(true);
            dispose();
        });
    }
}
```

看過去歷史預定紀錄UI介面

```
class Order extends JFrame {
   public Order() {
      setTitle("訂單列表");
      setSize(600, 400);
      setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
      setLayout(null);
      JLabel titleLabel = new JLabel("已訂房記錄");
      titleLabel.setBounds(250, 20, 200, 30);
      add(titleLabel);
      JPanel orderPanel = new JPanel();
      orderPanel.setLayout(new BoxLayout(orderPanel, BoxLayout.Y_AXIS));
      for (BookedRoomRecord record : HotelBookingApp.bookedRoomRecords) {
         Room room = record.getRoom();
         String recordDetails = "房間號碼: " + room.getRoomNumber() + ", 類型: " +
room.getType() +
               ", 價格: $" + room.getPrice() + ", 入住: " + record.getCheckInDate() + ", 退房: "
+ record.getCheckOutDate();
         JLabel recordLabel = new JLabel(recordDetails);
         orderPanel.add(recordLabel);
      }
      JScrollPane scrollPane = new JScrollPane(orderPanel);
      scrollPane.setBounds(50, 60, 500, 250);
      add(scrollPane);
      JButton backButton = new JButton("返回");
      backButton.setBounds(250, 330, 100, 30);
      add(backButton);
      backButton.addActionListener(e -> {
         new WelcomePage().setVisible(true);
         dispose();
      });
   }
}
```

成員貢獻度

111403023 資管三 葉東融 25%:
製作文件檔案、修改UML、小組討論、與老師討論、程式編撰
111403512 資管三 李權治 30%:
製作文件檔案、小組討論、修改UML、程式編撰
111403541 資管三 李介翔 25%:
小組討論、修改UML、與老師討論、程式編撰
111403038 資管三 蔡東哲 10%:
小組討論、與老師討論、修改UML
111403518 資管三 吳冠毅 10%:
小組討論、與老師討論、修改UML