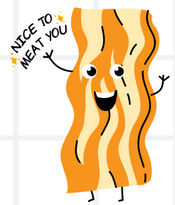
Reservation and Leftover Food

第七組

統計二 林承佑、林哲宇、陳柏翰

地政四 胡詠琪



CONTENTS

01 主題介紹與動機

02 ESG理念落實SDGs

03 系統設計與實施

04 與課堂學習內容結合

05 重要功能展示

06 總結與問題思考


YOU MAKE MISO HAPPY

YOU'RE THE
ONLY BUN FOR ME

DONUT
GIVE UP!

LAUGH S'MORE!
WORRY LESS

I U THIS MATCHA!

WANNA TACO 'BOUT IT?

ROLLING ON THE
FLOOR LAUGHING

01. 主題介紹與動機


KEEP CALM
AND CURRY ON

AVO-CUDDLE

THE ONE PHO ME

YOU ARE
BERRY SPECIAL

SO RAMEN-TIC

LEGEN-DAIRY
MILK

NICE TO
MEAT YOU

主題介紹

餐餐不剩


YOU MAKE MISO HAPPY
YOU'RE THE ONLY BUN FOR ME
LAUGH S'MORE!
WORRY LESS
餐餐不剩
Reservation and Leftover Food
帳號：

動機

餐廳	用餐位置供不應求 （買飯時間過久）	用餐時間不足 壓縮上課時間
演講活動 (交流版)	剩食無法及時消耗	浪費食物

理念

- 優化資源利用
- 提升學生用餐體驗
- 整合剩食管理
- 建立可靠的用戶資料庫
- 提高用餐效率和便利性

02. ESG理念落實SDGs

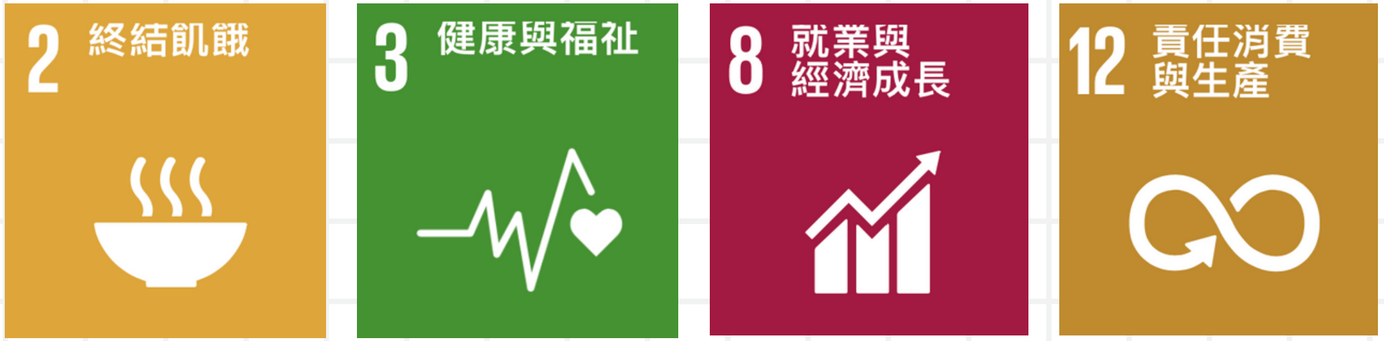
理念 → 落實SDGs

減少食物浪費

提升用餐效率

提升餐廳效能

提升資源利用效率



問題描述與解決方案

問題描述：

1. 供需失衡
2. 時間管理困難
3. 食物浪費嚴重
4. 預訂餐點後未取餐

解決方案：

1. 提升預訂系統效率
2. 完善剩食管理機制
3. 建立可靠的用戶資料庫
4. 次數累計機制


YOU MAKE MISO HAPPY


YOU'RE THE
ONLY BUN FOR ME


DONUT
GIVE UP!


LAUGH S'MORE!
WORRY LESS


I U THIS MATCHA!


WANNA TACO 'BOUT IT?


ROLLING ON THE
FLOOR LAUGHING

03. 系統設計與實施


KEEP CALM
AND CURRY ON


AVO-CUDDLE


THE ONE PHO ME

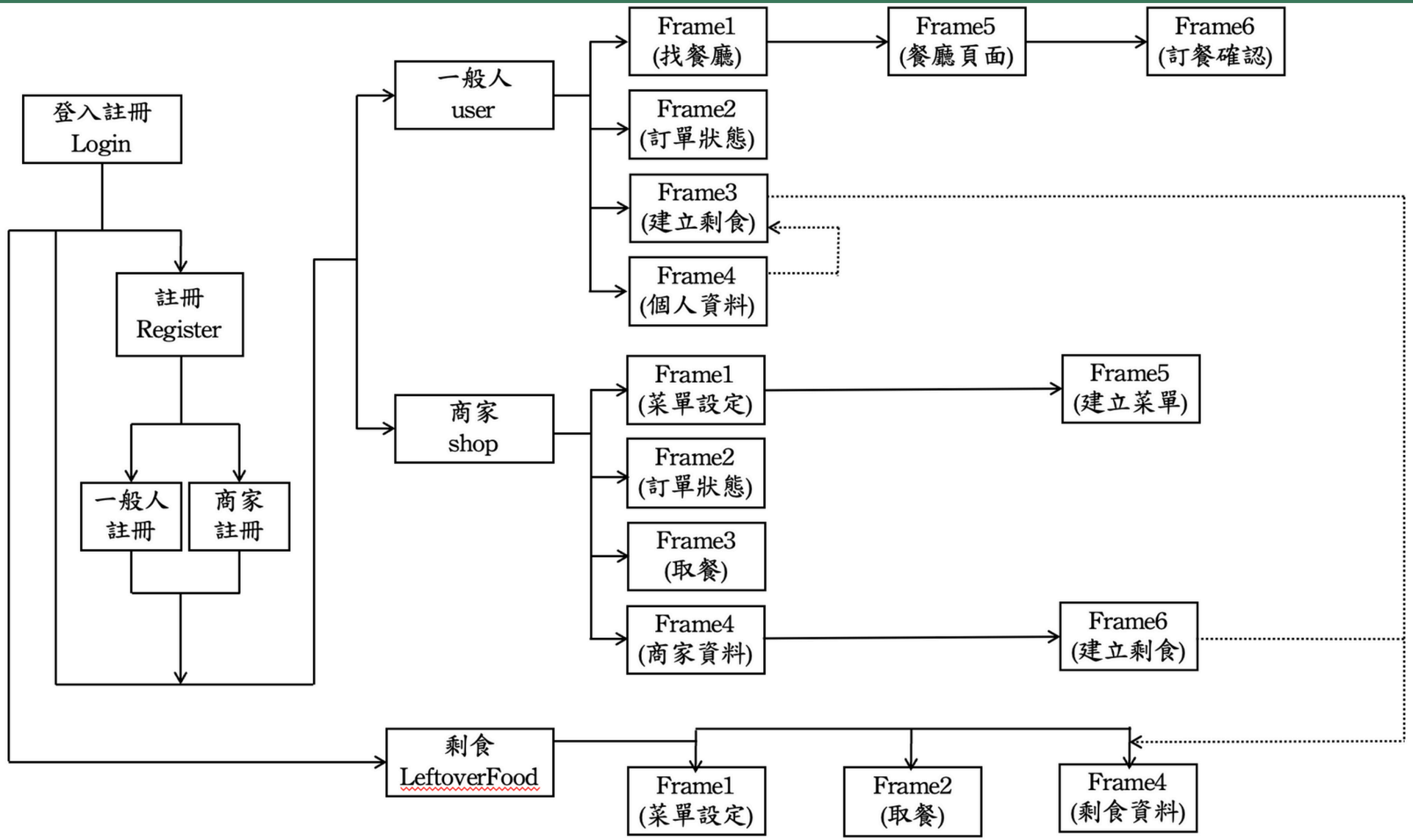

YOU ARE
BERRY SPECIAL


SO RAMEN-TIC!


LEGEN-DAIRY
MILK


NICE TO
MEAT YOU

前端設計


登入註冊
Login
註冊
Register
一般人
註冊
商家
註冊
一般人
user
Frame1
(找餐廳)
Frame5
(餐廳頁面)
Frame6
(訂餐確認)
Frame2
(訂單狀態)
Frame3
(建立剩食)
Frame4
(個人資料)
商家
shop
Frame1
(菜單設定)
Frame5
(建立菜單)
Frame2
(訂單狀態)
Frame3
(取餐)
Frame4
(商家資料)
Frame6
(建立剩食)
剩食
LeftoverFood
Frame1
(菜單設定)
Frame2
(取餐)
Frame4
(剩食資料)

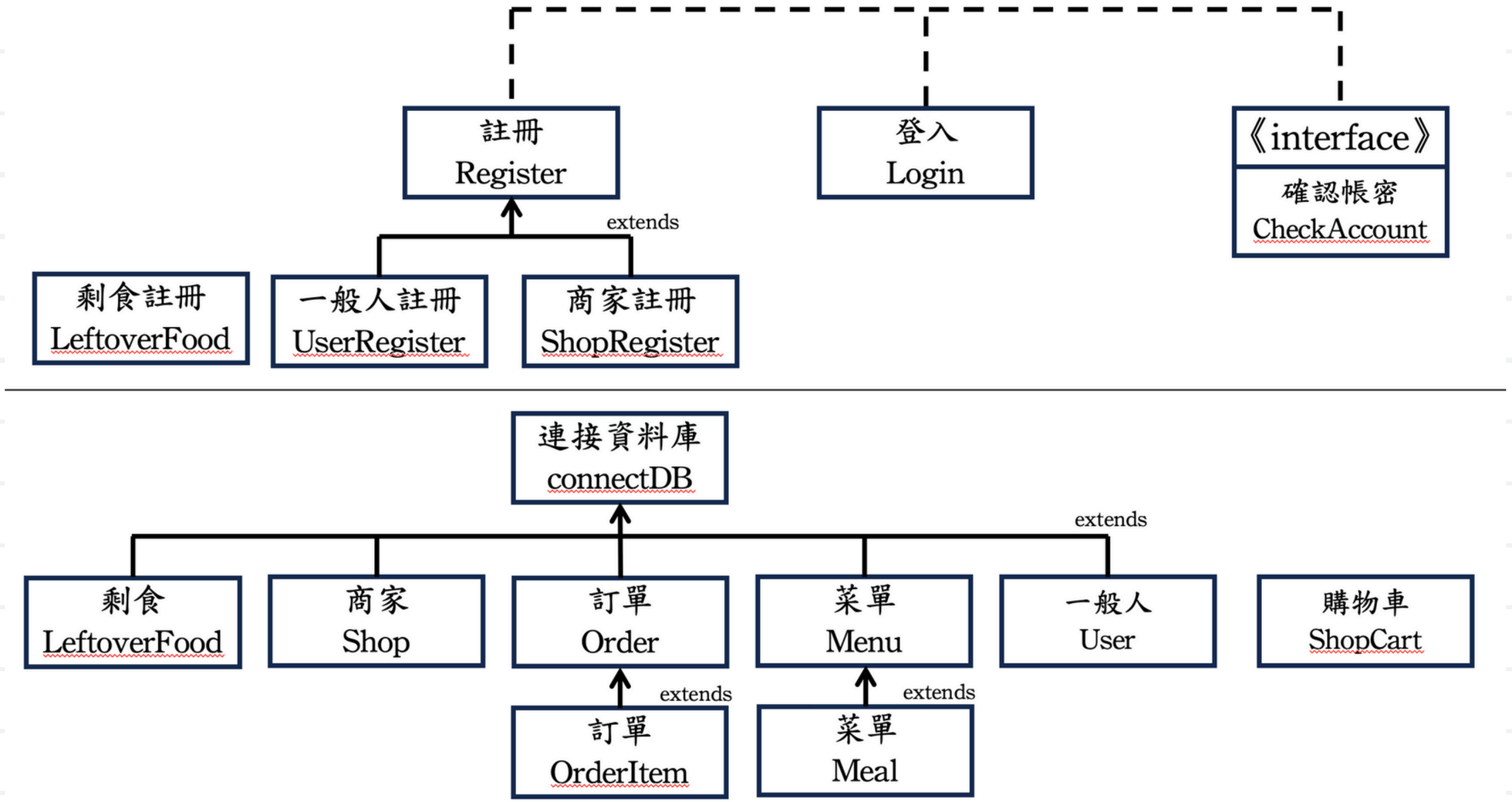
前端設計 #剩食


餐餐不剩
Reservation and Leftover Food
帳號：
密碼：
註冊
登入
快速
建立剩食


剩食名稱
數量
地點
立即預約
雞排
100
商學院
立即預約
商店名稱
營業時間
地點
訂餐
小吃部
10:00～18:00
指南路二段三百號
立即訂餐
找餐廳
訂單狀態
建立剩食
個人資料


尚未被預約數量：
97
現場領取：
0
確認
取餐編號
總數量
狀態
60300009
總數量: 3
修改內容
菜單設定
取餐
剩食資料

後段架構


註冊
Register
登入
Login
《interface》
確認帳密
CheckAccount
extends
剩食註冊
LeftoverFood
一般人註冊
UserRegister
商家註冊
ShopRegister
連接資料庫
connectDB
extends
剩食
LeftoverFood
商家
Shop
訂單
Order
菜單
Menu
一般人
User
購物車
ShopCart
extends
extends
訂單
OrderItem
菜單
Meal

各項資料表

帳號Account		
帳號ID ID	密碼 Password	種類 Type

商家基本資訊ShopInformation							
商家帳號ID ID	店名 Shop	地址 Address	電話 Telephone	信箱 Email	營業時間 WorkingTime	開放預約時間 OpeningHours	未取餐次數小於多少 NotPickUpTimes

一般人基本資訊UserInformation					
一般人帳號ID ID	系所 Department	姓名 Name	手機 Cellphone	信箱 Email	未取餐次數 NotPickUpTime

訂單Order							
一般人 ID	商家. 剩食 ID	取餐編號 PickUpNumber	餐點名稱 MealName	數量 Quantity	價格TotalPrice (該項總額)	備註Note （一般人）	取餐時間 PickUpTime

菜單Menu						
商家或剩食 帳號ID	菜名 MealName	開放預約數量 AppointmentNumber	量詞 Unit	每單位價格 UnitPrice	備註（商家） ShopNote	品項狀態 State

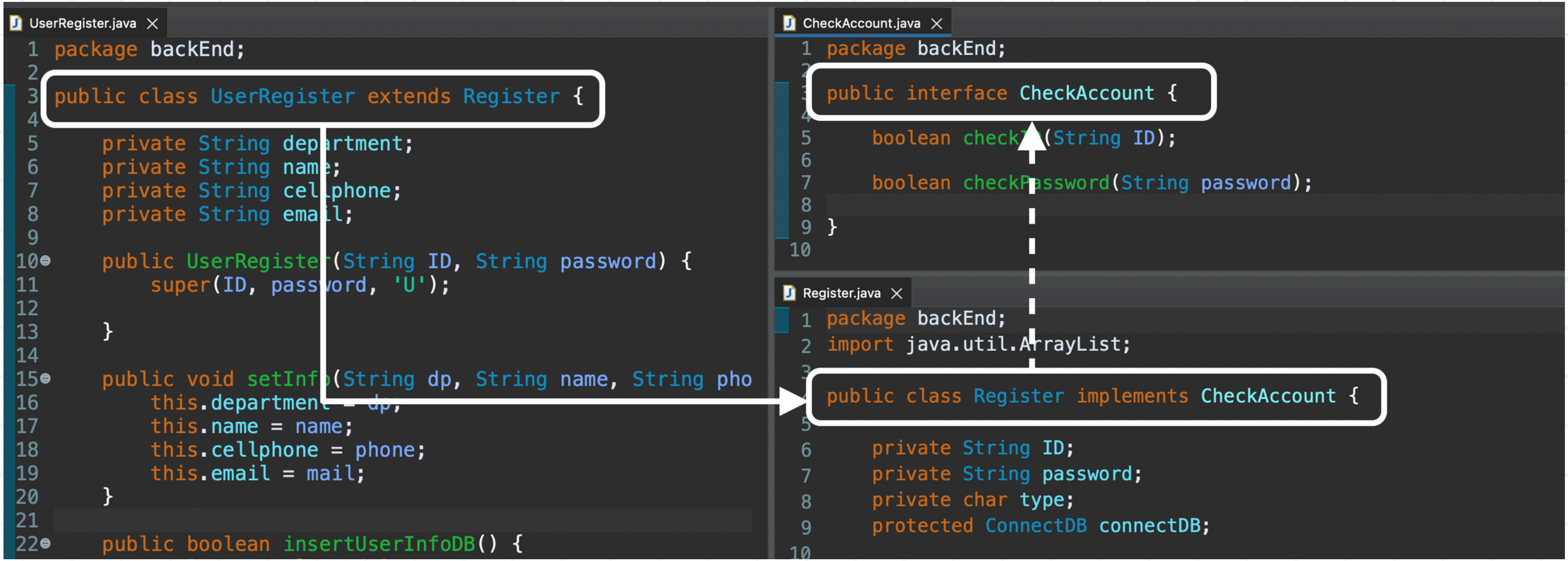
剩食LeftoverFood				
剩食帳號 ID	菜名MealName （名稱）	數量 Quantiy	地點 Address	建立時間 SetupTime

04. 與課堂學習內容結合

專案中用到的觀念

- inheritance & interface
- connect database
- overloading
- object & down casting &ArrayList
- ActionListner & Event & Anonymous class
- 其他基本邏輯運算，try- catch……等

inheritance & interface



UserRegister.java

```java
package backEnd;

public class UserRegister extends Register {

    private String department;
    private String name;
    private String cellphone;
    private String email;

    public UserRegister(String ID, String password) {
        super(ID, password, 'U');

    }

    public void setInfo(String dp, String name, String pho
        this.department = dp;
        this.name = name;
        this.cellphone = phone;
        this.email = mail;
    }

    public boolean insertUserInfoDB() {
```

CheckAccount.java

```java
package backEnd;

public interface CheckAccount {

    boolean checkID(String ID);

    boolean checkPassword(String password);

}
```

Register.java

```java
package backEnd;
import java.util.ArrayList;

public class Register implements CheckAccount {

    private String ID;
    private String password;
    private char type;
    protected ConnectDB connectDB;
```

connect database

```java
import java.sql.*;

public class ConnectDB {
    private String server = "jdbc:mysql://140.119.19.73:3315/";
    private String database = "111304019"; // change to your own databas
    protected String url = server + database + "?useSSL=false";
    private String username = "111304019"; // change to your own usernam
    private String password = "bovpk"; // change to your own password

    public ConnectDB() {}

    public ArrayList<Object> connectDB(String query, String condition) {
        ArrayList<Object> result = new ArrayList<>();
        try (Connection conn = DriverManager.getConnection(url, username,
             Statement stat = conn.createStatement()) {
            System.out.println("DB Connected");
```

```java
public class User extends ConnectDB {
    private String ID, depart, name, email, cellphone;
    private int notPickUpTime;
    private String[] infoName = { "Department", "Name", "Cellphon

    public User(String ID) {
        this.ID = ID;
    }

    public ArrayList<Shop> getAllShop(){
        ArrayList<Shop> allShop = new ArrayList<>();

        String query = "SELECT ID FROM `ShopInformation`";
        System.out.println("Executing query: " + query);
        ArrayList<Object> allID = super.connectDB(query, "ID");

```

overloading

```
// 刷新資料使用
public Meal(String ID, String mealName, int appointmentQuantity, St
        String state) {
    this.ID = ID;
    this.mealName = mealName;
    this.appointmentQuantity = appointmentQuantity;
    this.unit = unit;
    this.unitPrice = unitPrice;
    this.shopNote = shopNote;
    this.state = state;
}

// 商家新增菜單
public Meal(String ID, String mealName, int appointmentQuantity, St
    this.ID = ID;
    this.mealName = mealName;
    this.appointmentQuantity = appointmentQuantity;
    this.unit = unit;
    this.unitPrice = unitPrice;
    this.shopNote = shopNote;

    super.insertMenuDB(ID, mealName, appointmentQuantity, unit, uni
}

// 剩食新增菜單
public Meal(String ID, String mealName, int appointmentQuantity) {
    this.ID = ID;
    this.mealName = mealName;
    this.appointmentQuantity = appointmentQuantity;
    this.unit = "份";
    this.unitPrice = 0;
```

object & down casting & ArrayList

```
public ArrayList<OrderItem> refreshInfo(String shopID) {
    ArrayList<OrderItem> allOrder = new ArrayList<>();

    try {
        String query = "SELECT * FROM Orders WHERE ShopID = '" + s
        System.out.println("Executing query: " + query); // 输出查询
        ArrayList<ArrayList<Object>> result = super.fetchAll(query

        for (ArrayList<Object> row : result) {
            String userID = (String) row.get(0);
            String shopID2 = (String) row.get(1);
            String pickUpNum = String.valueOf(row.get(2));
            String mealName = (String) row.get(3);
            String quantity = (String) row.get(4);
            int totalPrice = (int) row.get(5);
            String userNote = (String) row.get(6);
            String pickUpTime = (String) row.get(7);
            String state = (String) row.get(8);

            OrderItem order = new OrderItem("LF", userID, shopID2,
                    pickUpTime, state);
            allOrder.add(order);
        }
```

ActionListner & Event & Anonymous class

```
quanTable.setCellFactory(col -> new TableCell<Meal, Void>() {
    private final Label quantityLabel = new Label();
    private final Button incrementButton = new Button("+");
    private final Button decrementButton = new Button("-");
    private final HBox hbox = new HBox(5, decrementButton, quantityLabe

    {
        incrementButton.setOnAction(event -> {
            int quantity = Integer.parseInt(quantityLabel.getText());
            quantity++;
            quantityLabel.setText(String.valueOf(quantity));
            updateMealQuantity(getIndex(), quantity);
        });

        decrementButton.setOnAction(event -> {
            int quantity = Integer.parseInt(quantityLabel.getText());
            if (quantity > 0) {
                quantity--;
                quantityLabel.setText(String.valueOf(quantity));
                updateMealQuantity(getIndex(), quantity);
            }
        });

        hbox.setAlignment(Pos.CENTER);
    }

    @Override
    protected void updateItem(Void item, boolean empty) {
        super.updateItem(item, empty);
        if (empty) {
            setGraphic(null);
        } else {
            Meal meal = getTableView().getItems().get(getIndex());
            quantityLabel.setText(String.valueOf(meal.getQuantity()));
            setGraphic(hbox);
        }
    }

    private void updateMealQuantity(int index, int quantity) {
        Meal meal = getTableView().getItems().get(index);
        meal.setQuantity(quantity);
        meal.setTotalPrice(meal.getUnitPrice() * quantity);
        tableView.refresh();
    }
});
```


YOU MAKE MISO HAPPY


YOU'RE THE
ONLY BUN FOR ME


DONUT
GIVE UP!


LAUGH S'MORE!
WORRY LESS


I U THIS MATCHA!


WANNA TACO 'BOUT IT?


ROLLING ON THE
FLOOR LAUGHING

05.重要功能展示


KEEP CALM
AND CURRY ON


AVO-CUDDLE


THE ONE PHO ME

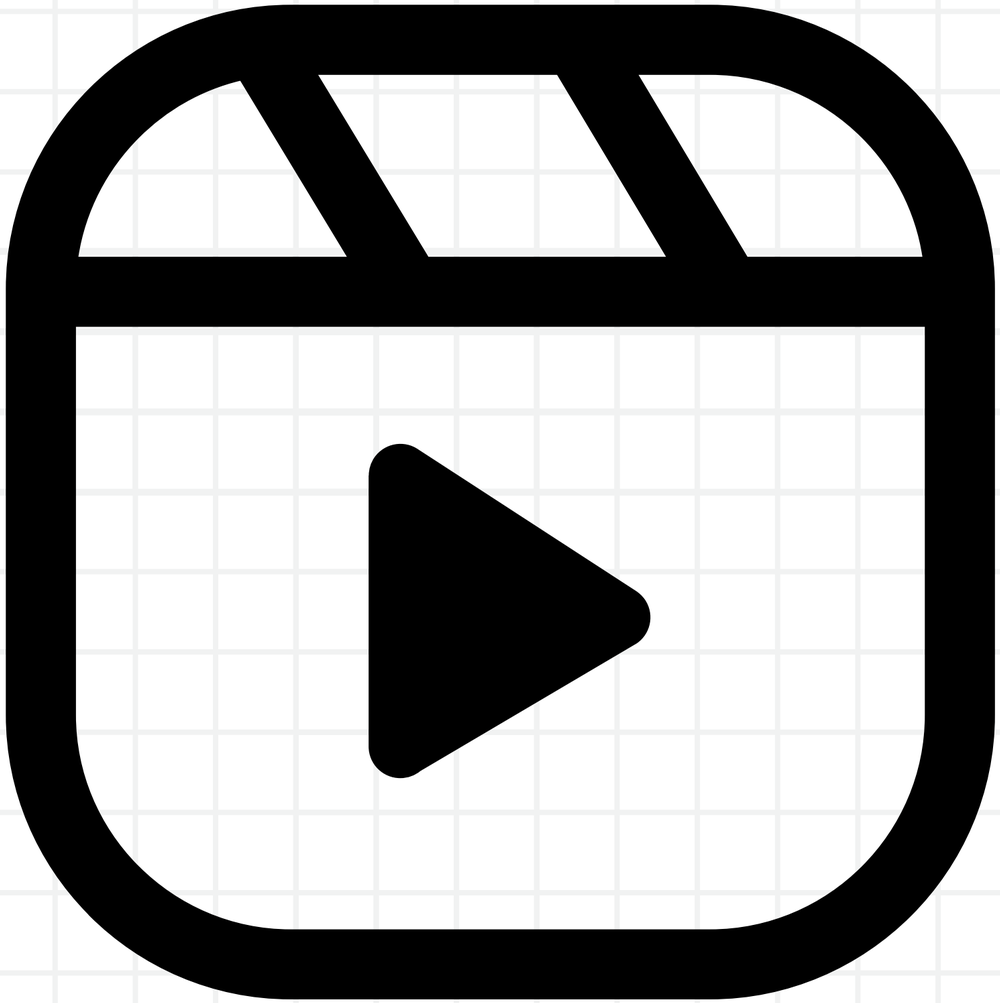

YOU ARE
BERRY SPECIAL


SO RAMEN-TIC!


LEGEN-DAIRY
MILK


NICE TO
MEAT YOU

實作影片



重要功能 #1

結構式取餐代碼

日期	當天訂單數量(順序)	隨機生成兩位數
0603	XXX	XX

重要功能 #2

使用者數量 = 系統刷新速度

登入帳號	→	刷新所有「剩食」，超過時間刪除
查看訂單	→	昨日未完成訂單刪除，並增加該未取餐次數
查看取餐	→	已完成訂單移除畫面，同步刪除資料表該筆項目
建立剩食 (刷新剩食)	→	刷新剩食，超過時間刪除 刪除剩食資料、菜單資料、未取餐資料

重要功能 #3

雙向奔赴的設計

使用者（系統端）

預訂餐點後沒有取餐

系統會對該帳號做次數累計

商家

自行設定次數上限

篩選可預訂餐點的使用者


YOU MAKE MISO HAPPY


YOU'RE THE
ONLY BUN FOR ME


DONUT
GIVE UP!


LAUGH S'MORE!
WORRY LESS


I U THIS MATCHA!


WANNA TACO 'BOUT IT?


ROLLING ON THE
FLOOR LAUGHING

06.總結與問題思考


KEEP CALM
AND CURRY ON


AVO-CUDDLE


THE ONE PHO ME


YOU ARE
BERRY SPECIAL


SO RAMEN-TIC


LEGEN-DAIRY
MILK


NICE TO
MEAT YOU

程式設計面向

- 大量使用 Class 與 polymorphism 概念

  ➜提升程式碼的重複使用率

- 使用 FXML (javaFX) 結合 controller

  ➜平台畫面整齊統一，ActionListener清楚明白

程式設計面向

- 程式碼整體複雜過高，不夠清晰易修正與使用
- 後端開發缺乏系統性建立
- 介面連結發現前後思索有斷層
- 目前學到的方法有限，學過的不夠了解使用方式

應用程式面向

- 有效整合同時大量剩食問題
- 增加訂餐功能提升平台使用率
- 雙向奔赴設計，提升平台可信度
- 免使用服務費且介面使用簡潔直覺

應用程式面向

- 大量依靠連結資料庫，使用速度切換慢
- 部分使用者體驗不夠完善
- 平台獨立性質，可能無法替代現有交流版

Group7. 餐餐不剩

The End