

文档编号：《饭否》网上订单 *app*-SDS - 2.0

# 《饭否》网上订单 *app* 软件设计规格说明书

日期：2020 年 10 月 24 日

## 文档变更历史记录

序号	变更日期	变更人员	变更内容详情描述	版本
1	2020\10\13	司马晨、 刘晨晨、 蒋志文、 雷黛瑛、 田硕	撰写了软件设计规格说明书初稿	V1.0
1.1		司马晨	完成引言、软件设计约束、软件设计中的软件体系结构设计、用例设计的文字部分、类设计 和数据设计	V1.0
1.2		刘晨晨	完成用例设计的顺序图绘制	V1.0
1.3		蒋志文	完成部署设计	V1.0
1.4		雷黛瑛、 田硕	完成用户界面设计	V1.0
2.1	2020\10\20	司马晨	绘制了系统结构图，以及重新进行了类设计 (具体说明类的对象和方法)	V2.0
2.2	2020\10\20	刘晨晨	重新绘制顺序图，重新进行数据设计、以及图 的重新编号	V2.0
2.3	2020\10\20	蒋志文	完成了部署图的绘制	V2.0
2.4	2020\10\20	田硕、雷 黛瑛	完成了界面布局说明	V2.0
3.1	2020\10\24	司马晨	完善了类的具体设计（添加 Getter、setter 和构造函数），完成各种排序算法的代码	V3.0
3.2	2020\10\24	刘晨晨	完善数据库设计中的物理数据库的设计，新增 数据库和类的对应关系表	V3.0
3.3	2020\10\24	蒋志文	完成了部署图的绘制	V3.0

3.4	2020\10\24	王彬	为 3.0 版本的修改提出指导意见	V3.0
-----	------------	----	-------------------	------

# 目录

1、引言 .....	
1.1 编写目的 .....	
1.2 读者对象 .....	
1.3 软件项目概述 .....	
1.4 文档概述 .....	
1.5 定义 .....	
1.6 参考资料 .....	
2、软件设计约束 .....	
2.1 软件设计目标和原则 .....	
2.2 软件设计的约束和限制 .....	
3、软件设计 .....	
3.1 软件体系结构设计 .....	
3.2 用户界面设计 .....	
3.3 用例设计 .....	
3.4 类设计 .....	
3.5 数据设计 .....	
3.6 部署设计 .....	

# 1、引言

## 1.1 编写目的

软件设计过程包括的体系结构设计、用户界面设计、用例设计、子系统/构件设计、类设计和数据设计分别从不同的层次（从宏观到微观、从全局到局部）、不同的视角（从结构到行为、从模块到数据）对软件系统进行了设计，产生了不同的软件制品（如体系结构模型、用例实现模型、用户界面模型、子系统/构件模型、数据设计模型、部署模型等等）。在完成上述所有设计工作之后，需要将这些软件设计成果进行整合，形成一个系统、完整的软件设计方案。本文档以软件设计规格说明书的形式描述了该设计方案，方便评审人员对设计方案的正确性、合理性等方面进行评审。

## 1.2 读者对象

用户、软件设计人员、程序员、软件需求分析人员、质量保证人员、软件测试工程师、配置管理工程师。

## 1.3 软件项目概述

项目名称：《饭否》网上订单

app 项目简称：《饭否》

用户单位：各个入驻的餐厅、骑手和点餐的顾客。

开发单位：萤火科技有限公司。

功能和用途：实现快捷点餐，对商家，用户和骑手都便捷的软件，为了使人们想要点餐或者找不到吃什么的时候使用本软件。

## 1.4 文档概述

本文档由 3 个部分组成：引言、软件设计约束、软件设计。主要介绍《饭否》软件的软件设计过程的详细大纲。

## 1.5 定义

无

## 1.6 参考资料

[1]. 软件工程. 齐治昌, 谭庆平, 宁洪. 北京: 高等教育出版社, 2012

[2]. 需求分析与设计. 马素霞译. 北京: 机械工业出版社, 2009

[3]. 面向使用的软件设计. 刘正捷译. 北京: 机械工业出版社, 2011

# 2、软件设计约束

## 2.1 软件设计目标和原则

软件设计的目标是, 根据软件系统的需求(包括功能性需求和非功能性需求), 综合考虑软件开发过程中的各种制约因素(如技术、资源、进度等), 遵循软件工程设计的原则(如模块化、信息隐藏、问题分解等), 给出软件系统的实现解决方案和蓝图, 产生可指导编码实现的设计模型及文档。

软件设计活动还须遵循相关的策略和原则, 以指导软件设计人员的行为, 并对设计成果提出约束和要求。具体的, 这些设计策略和原则描述如下。

- (1) 抽象和逐步求精的原则
- (2) 模块化与高内聚度、低耦合度的原则
- (3) 信息隐藏的原则
- (4) 多视点以及关注点分离的原则
- (5) 软件重用的原则
- (6) 迭代设计的原则
- (7) 可追踪性的原则

## 2.2 软件设计的约束和限制

- 开发环境: 开发语言: Python、SQL、Java

- 开发软件：PyCharm、Android studio、MySQL
- 因为不考虑 PC 版本，所以可以在当前主流手机操作系统上使用，适配于 Android 平台。与其他软件高度兼容，可进行数据的互传。
- 硬件环境：安卓虚拟机，打印机，安卓实体机。
- 用户要求：下载安装本产品并了解产品功能使用的认证注册信息用户

## 3、软件设计

### 3.1 软件体系结构设计

本软件根据用户群体分为三类：面向顾客的“点餐系统”、面向餐厅的“菜品设计和确认订餐系统”和面向送餐员的“送餐系统”。“点餐系统”主要是用于顾客浏览餐品、订餐。“菜品设计和确认订餐系统”用于让餐厅老板进行设置菜单和价格、以及接单。“送餐系统”用于让外卖员接单以及确认送达。

- 菜品设置和确认订餐系统

该系统为餐厅老板使用，拥有修改菜单信息、接单等功能。

- 点餐系统

该系统为顾客使用，对于菜单等信息仅拥有查询功能，可以根据菜单进行选餐、下单。

- 送餐系统

该系统为骑手使用，负责接单。

系统结构图如下：

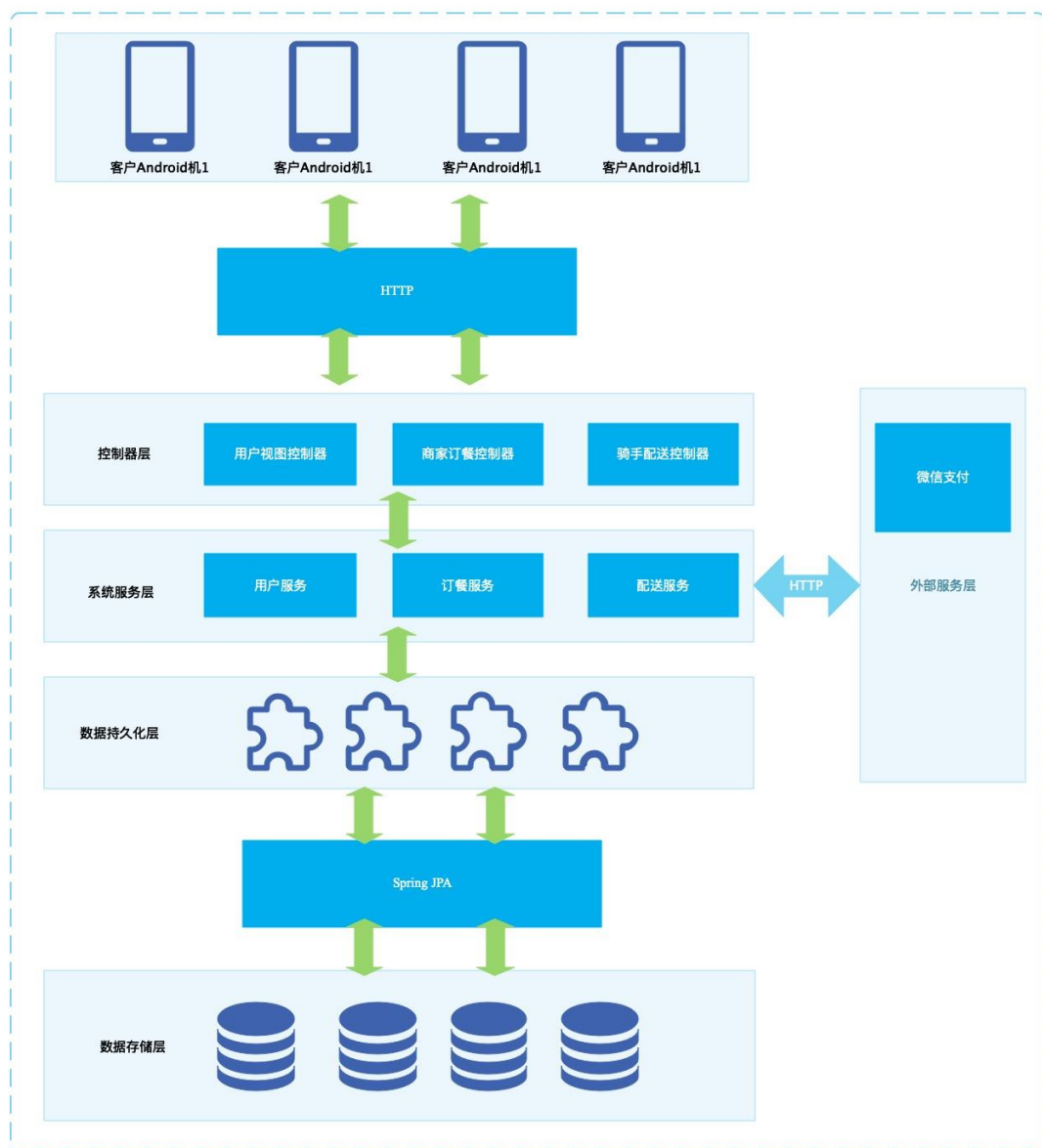


图 3.1.1 软件体系结构图

## 3.2 用户界面设计

根据“《饭否》网上订餐 app”的用例描述以及每个用例的交互图，可以发现该软件系统在手机端的 APP 需要有以下一组界面以支持用户的操作。

- 引导界面“GuidingUI”，APP 加载启动时用于展示和介绍该软件系统。
- 登录界面“LoginUI”，其职责是帮助用户输入用户信息以登录到系统之中。



- 餐品设置界面“MenuSetting”，其职责是让餐厅负责人设置菜单。
- 选餐界面“MealOrdering”，其职责是让顾客进行选餐。
- 选餐成功界面“OrderingSuccess”，当顾客选餐成功时进入该界面
- 送餐界面“MealSending”，其职责是让骑手进行送餐时的选择（确认送达等）。
- 系统设置界面“SettingUI”，其职责是配置系统。

1) 本界面为本软件的欢迎界面。



图 3.2.1 欢迎界面图

2) 本界面为用户进行登录的界面，用户需要输入注册时所选用的用户名并

输入设定的密码即可。

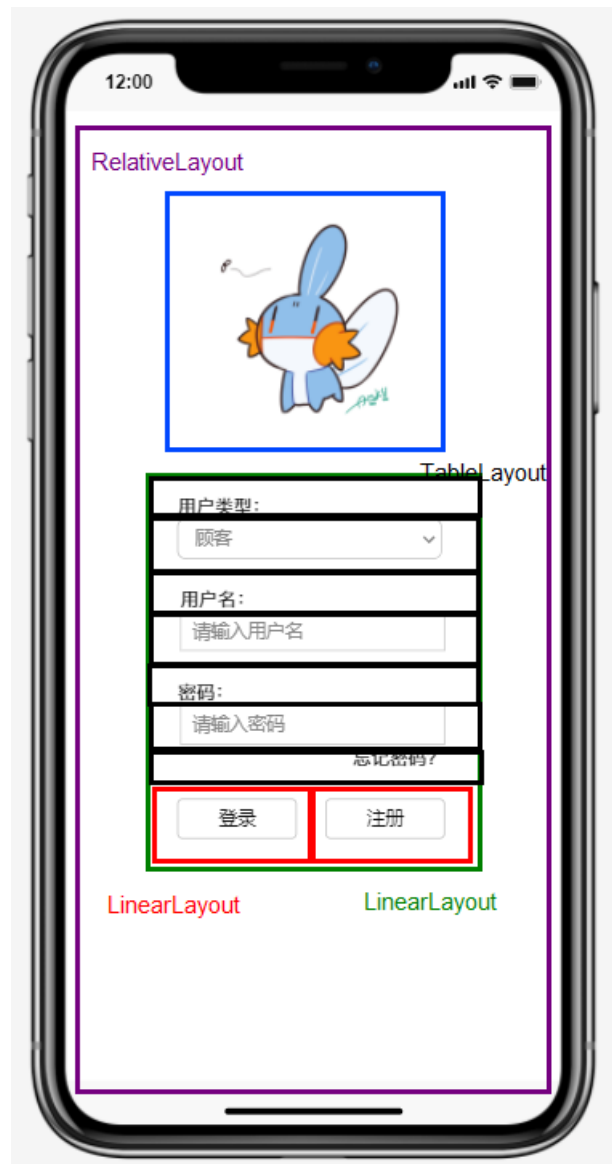


图 3.1.2 登录注册界面图

3) 下图为本软件“餐品设置”的功能界面。



图 3.1.3 餐品设置图

4) 下图为本软件“选餐”的功能界面。

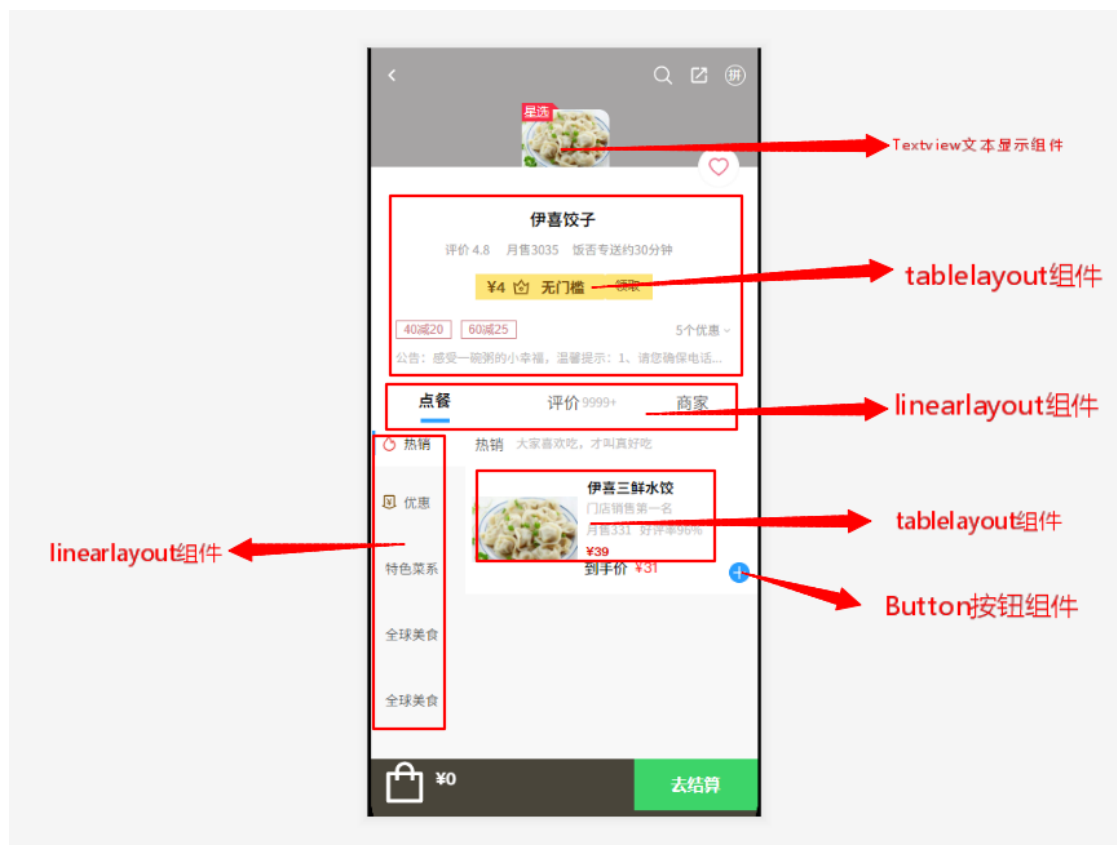


图 3.1.4 选餐功能界面图

5) 下图为本软件“选餐成功”界面



图 3.1.5 选餐成功界面图

6) 下图为本软件“送餐”功能的界面。

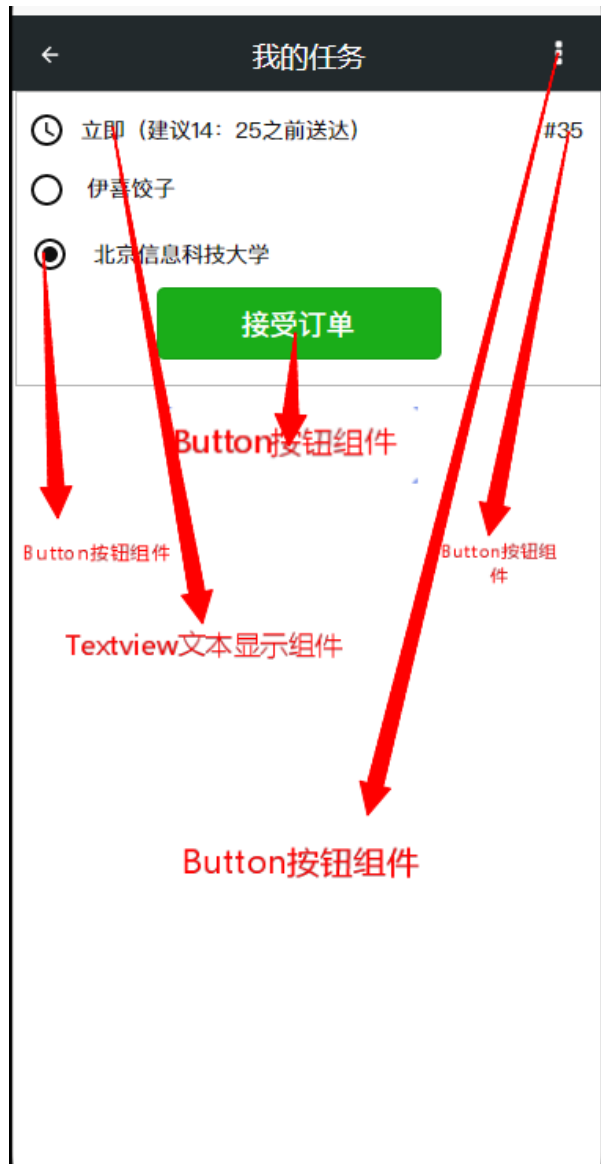


图 3.1.6 送餐功能界面图

### 3.3 用例设计

#### (1) “用户登录”用例实现的设计方案

首先，用户通过边界类“LoginUI”对象输入登录的账户和密码，随后该对象向控制类“LoginManager”对象发消息“login(account, password)”，以请求登录到系统之中。接收到消息后，“LoginManager”对象将向实体类“UserLibrary”对象发消息“verifyUserValidity(account, password)”以验证用户提交的账号和密码是否合法。“UserLibrary”对象通过自身内部的方法

“isUserValid(account, password)”来判断用户身份的合法性，并将验证的结果“VerificationResult”返回给“LoginManager”对象，“LoginManager”对象以此进一步地将登录成功与否的消息发送给“LoginUI”对象。一旦登录成功，系统将根据用户所在的用户组，将界面重定向到“MenuSettingUI”（餐厅用户）、“MealOrderingUI”（顾客用户）或“MealSendingUI”（骑手用户）主界面。

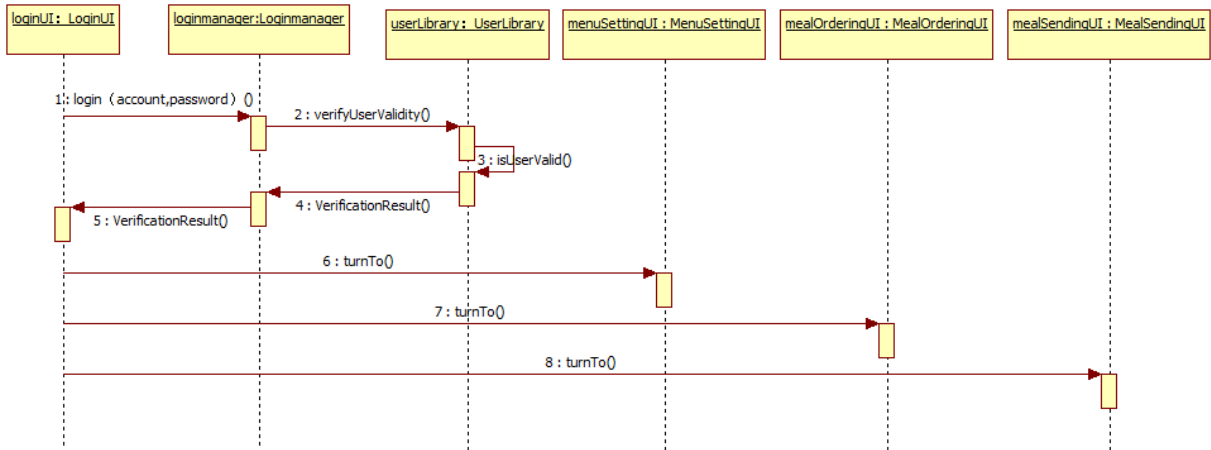


图 3.3.1 用户登录时序图

(2) “餐品设置”用例设计的实施方案

边界类“MenuSettingUI”对象基于用户输入的餐品设置信息，向控制类“MenuManager”对象发送增加、修改或删除消息：add (Meal)、set (Meal)、delete (Meal)，以请求在数据库 Menu 中进行相关操作。接受到消息后，“MenuManager”对象将对数据库 Menu 发送修改请求。修改成功后，系统将界面重定向到“MenuSettingUI”主界面。

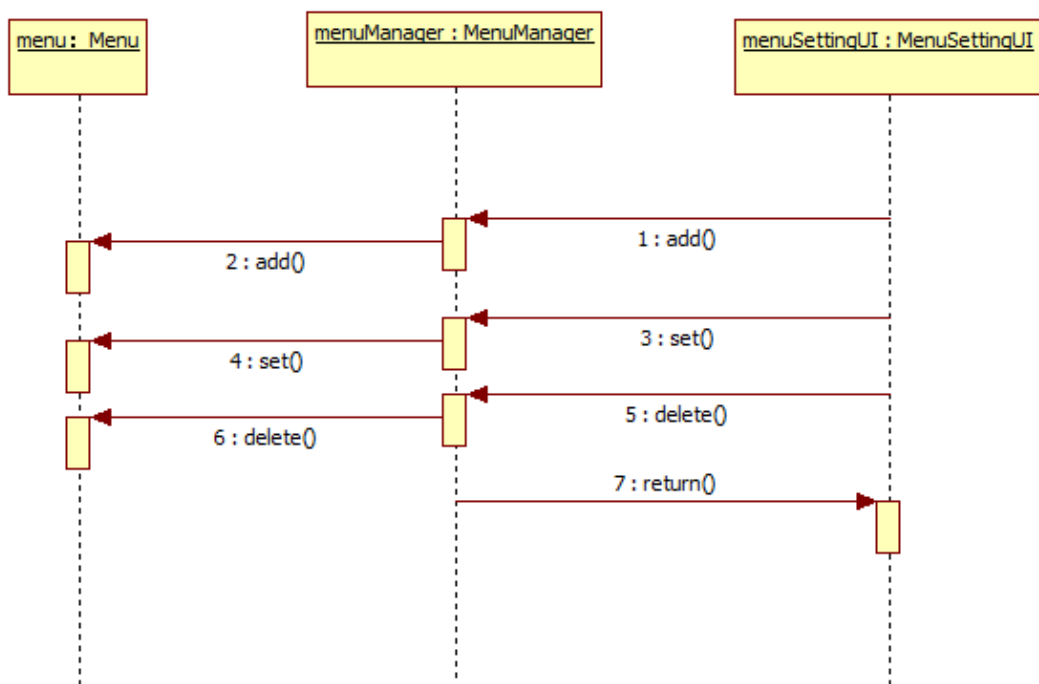


图 3.3.2 商家添加商品时序图

### (3) “点餐”用例设计的实现方案

边界类“MenuGetter”对象基于用户选择的餐厅名字在数据库 Menu 中读取相关餐品数据。边界类“MealOrderingUI”对象基于用户输入的订单信息，向控制类“MealManager”对象发送增加、修改或删除消息：add(Order)、set(Order)、delete (Order)，以请求在数据库 Order 发送修改请求。修改成功后，系统将界面重定向到“OrderingSuccess”选餐成功界面。

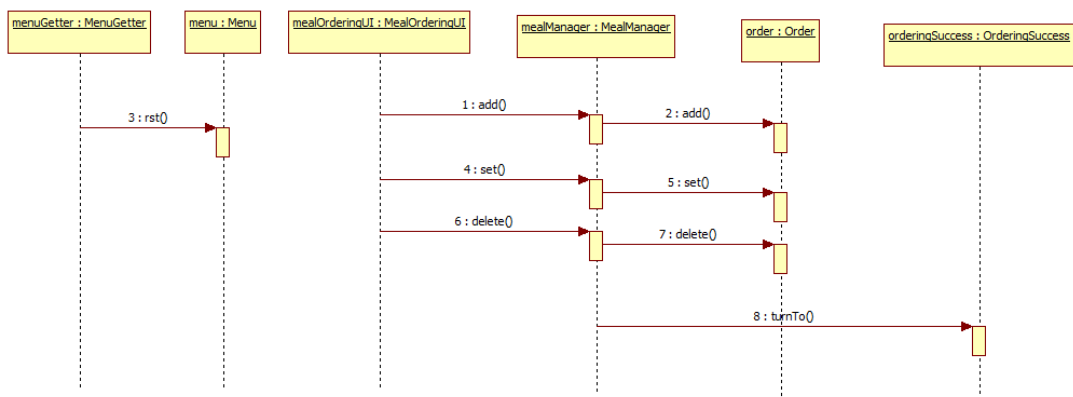


图 3.3.3 用户点餐时序图

#### (4) “送餐”用例设计的实现方案

实体类“RstCommander”根据用户所在的地理位置和当前订单计算出推荐用户可接的单:Calculate():Order。边界类“OrderGetter”对象基于“RstCommander”所计算出的结果在数据库 Order 中读取订单数据。边界类“MealSendingUI”对象基于用户输入的选单信息,向控制类“OrderManager”发送接单请求:select (Order), 以请求在数据库 Order 中将对应订单标记为“Selected”。接单成功后,系统将界面重定向到“MealSendingUI”送餐界面(可以一次多单)。

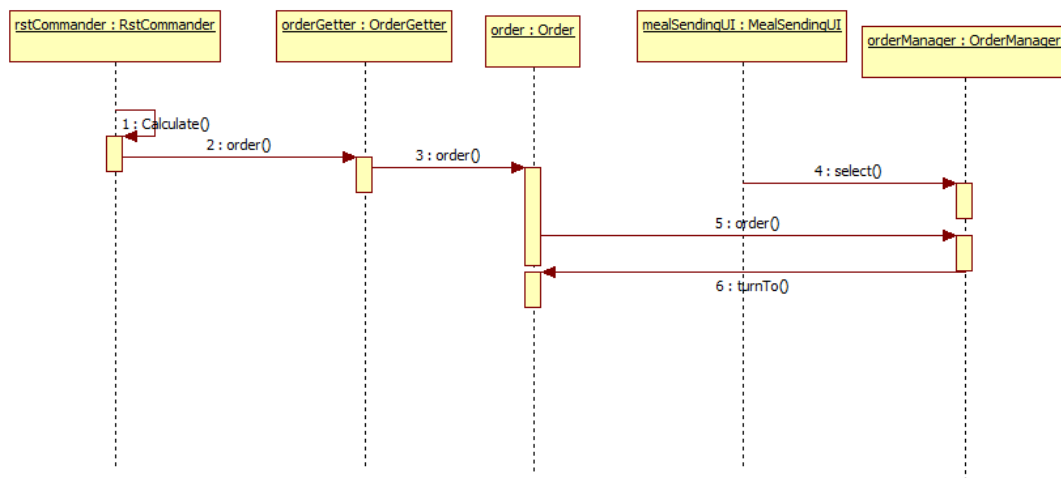


图 3.3.4 送餐时序图

### 3.4 类设

#### (1) 实体类

i. class Restaurant:

```
address:Point
name:String
id:int
foods:ArrayList<Food>
getRest():Restaurant
setRest(Restaurant restaurant):void
```



```
Restaurant() : constructor  
Restaurant(Point address, String name, int id,  
ArrayList<Food>food) : constructor
```

ii. class Food:

```
name:String  
price:double  
id:int  
getFood() :Food  
setFood(Food food):void  
Food(String name,double price,int id): constructor  
Food() :constructor
```

iii. class Order:

```
foods:ArrayList<Food>  
price:double  
rst:Restaurant  
id:int  
getOrder() :Order  
setOrder(Order order):void  
Order(ArrayList<Food>foods double price,int id): constructor  
Order() :constructor
```

iv. class UserLibrary:

```
isValid(String account,String password):boolean  
判断用户名/密码是否合法。
```

v.     class RstCommander:

        recommand(Point, ArrayList<Order>):Order

        根据坐标推荐给骑手最近的商家。

        伪代码:

```
        priority_queue<Point>q;

        q.push(rider.point);

        vector<Order>res;

        visited[rider.id]=1;

        all=0;

        while(!q.empty()){

            int temp=q.top();

            res.push_back(temp);

            all++;

            for(int i=0;i<g[temp].size();i++){

                if(!visited[g[temp][i]){

                    q.push(g[temp][i]);

                    visited[g[temp][i].id]=1;

                }

            }

            if(all>=needs)break;

        }

        return res;
```

vi.    interface MySort:

        sort():Restaurant

vii. class sortByDistance implements MySort:

sort():Restaurant。

根据距离进行排序

伪代码:

```
bool cmp(Restaurant p, Restaurant q) {  
    sqrt((p.address.x-r.x)*(p.address.x-r.x)+(p.address.y-r.y)  
*(p.address.y-r.y))<sqrt((q.address.x-r.x)*(q.address.x-r.x)  
+(q.address.y-r.y)*(q.address.y-r.y));  
}  
void sort() {  
    sort(rest.begin(), rest.end, cmp);  
}
```

viii. class sortByGrade implements MySort:

sort():Restaurant。

根据分数进行排序

伪代码:

```
bool cmp(Restaurant p, Restaurant q) {  
    p.grade>q.grade;  
}  
void sort() {  
    sort(rest.begin(), rest.end, cmp);  
}
```

ix. class sortByAl implements MySort:

sort():Restaurant。

智能排序

伪代码:

```
bool cmp(Restaurant p, Restaurant q) {  
    p.grade/sqrt ( (p.address.x-r.x ) *(p.address.x-r.x )  
+ (p.address.y-r.y)*(p.address.y-r.y))>q.grade/sqrt((q.address.x-r.x)  
*(q.address.x-r.x) +(q.address.y-r.y) *(q.address.y-r.y) );  
}  
void sort() {  
    sort(rest.begin(), rest.end, cmp);  
}
```

## (2) 边界类

i. class LoginUI: 将用户名和密码发送给 LoginManager。

sendToManager(String account, String password):void

ii. class MenuSettingUI: 根据餐厅用户提交的请求向 MenuManager 发送对数据库 Menu 修改信息。

quest():void

iii. class MealSettingUI: 根据顾客用户提交的请求向 MealManager 发送对数据库 Order 修改信息。

quest():void

iv. class MenuGetter: 根据顾客用户选择的餐厅在数据库 Menu 中读取相关信息。

getMenu(Restaurant rst):ArrayList<Food>

### (3) 控制类

- i. class MenuManager: 向数据库 Menu 申请增、删、改、查等操作。

addMenu(Food food):boolean

deleteMenu(Food food):boolean

setFood(Food oldFood, Food newFood):Boolean

getFood(int id):Food

getFoods(Restaurant rst):ArrayList<Food>

- ii. MealManager: 向数据库 Order 申请增、删、改、查等操作。

addOrder(Order order):boolean

deleteOrder(Order order):boolean

setOrder(Order oldOrder, Order newOrder):boolean

getOrder(int id):Order

- iii. class LoginManager: 向 UserLibrary 发送用户名和密码。

sendToLibrary():boolean

## 3.5 数据设计

### I 数据库 E-R 图如下所示

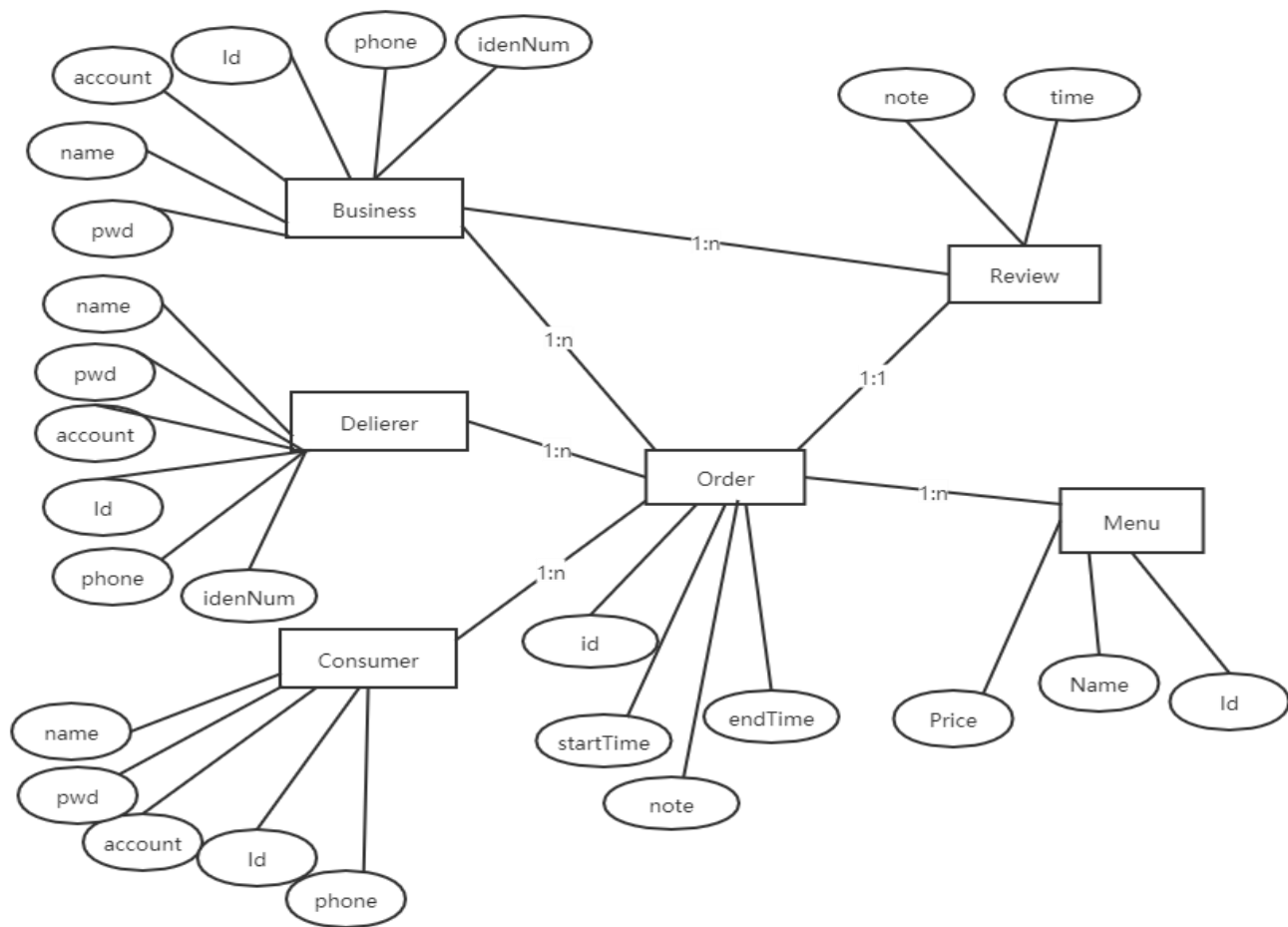


图 3.5.1 数据库 E-R 图

## II 数据库逻辑设计

### (1) 用户数据库 User

用户数据库 User 包含三个表，分别是买家表 Consumers，商家 Business，骑手 Deliverer

其中 Consumer 设计如下：

列名	数据类型	数据范围	备注
userId	Unique	8 位	用户 id，主键
account	String		用户账号
pwd	String		用户密码
name	String		用户名
phone	int	11 位	手机号

其中 Business 设计如下：

列名	数据类型	数据范围	备注
userId	Unique	8 位	商家 id, 主键
account	String		商家账号
pwd	String		商家密码
name	String		商家名
phone	Int	11 位	手机号
idenNum	String	18	身份证号

其中 Deliverer 设计如下：

列名	数据类型	数据范围	备注
userId	Unique	8 位	骑手 id, 主键
account	String		骑手账号
pwd	String		骑手密码
name	String		骑手名
phone	Int	11 位	手机号
idenNum	String	18 位	身份证号

## (2) 商品表 Menu

列名	数据类型	数据范围	备注
Name	String		商品名称
Price	Double		商品价格
Id	Unique	8 位	商品 id, 主键
orderId	Int	8 位	订单编号, 外键

## (3) 订单 Order

列名	数据类型	数据范围	备注
id	unique	8 位	订单编号
startTime	Date		开始时间
endTime	Date		结束时间
note	String		备注
consumerId	Int	8 位	用户 id, 外键
businessId	Int	8 位	商家 id, 外键
delivererId	Int	8 位	骑手 id, 外键

#### (4) 评价 Review

列名	数据类型	数据范围	备注
note	String		评价
time	date		评价时间
businessId	Int	8 位	商家 id, 外键

### III 数据库物理设计

#### a) 用户数据库 User

用户数据库 User 包含三个表，分别是买家表 Consumers，商家 Business，骑手 Deliverer

其中 Consumer 设计如下：



列名	数据类型	数据范围	备注
userId	Bigint	8 位	用户 id, 主键
account	text	8-20 位	用户账号
pwd	text	8-20 位	用户密码
name	text	8-20 位	用户名
phone	bigint	11 位	手机号

其中 Business 设计如下:

列名	数据类型	数据范围	备注
userId	Bigint	8 位	商家 id, 主键
account	text	8-20 位	商家账号
pwd	text	8-20 位	商家密码
name	text	8-20 位	商家名, 以此 建立索引
phone	bigint	11 位	手机号
idenNum	Bigint	18 位	身份证号

其中 Deliverer 设计如下:

列名	数据类型	数据范围	备注
userId	Bigint	8 位	骑手 id, 主键
account	text	8-20 位	骑手账号
pwd	text	8-20 位	骑手密码
name	text	8-20 位	骑手名
phone	bigint	11 位	手机号
idenNum	Bigint	18 位	身份证号

b) 商品表 Menu

列名	数据类型	数据范围	备注
Name	Text	8-20 位	商品名称，以此建立索引
Price	Double		商品价格
Id	Bigint	8 位	商品 id，主键
orderId	Bigint	8 位	订单编号，外键

c) 订单 Order

列名	数据类型	数据范围	备注
id	Bigint	8 位	订单编号
startTime	Date		开始时间
endTime	Date		结束时间
note	Text	100 个字符以内	备注
consumerId	Bigint	8 位	用户 id，外键
businessId	Bigint	8 位	商家 id，外键
delivererId	Bigint	8 位	骑手 id，外键

d) 评价 Review

列名	数据类型	数据范围	备注
note	Text	300 个字符	评价

		以内	
time	Date		评价时间
businessId	Bigint	8 位	商家 id, 外键

#### IV 数据库表和访问类的对应关系

类名	数据库名
LoginManager	User
MenuManager	Menu
MealManager	Order

### 3.6 部署设计

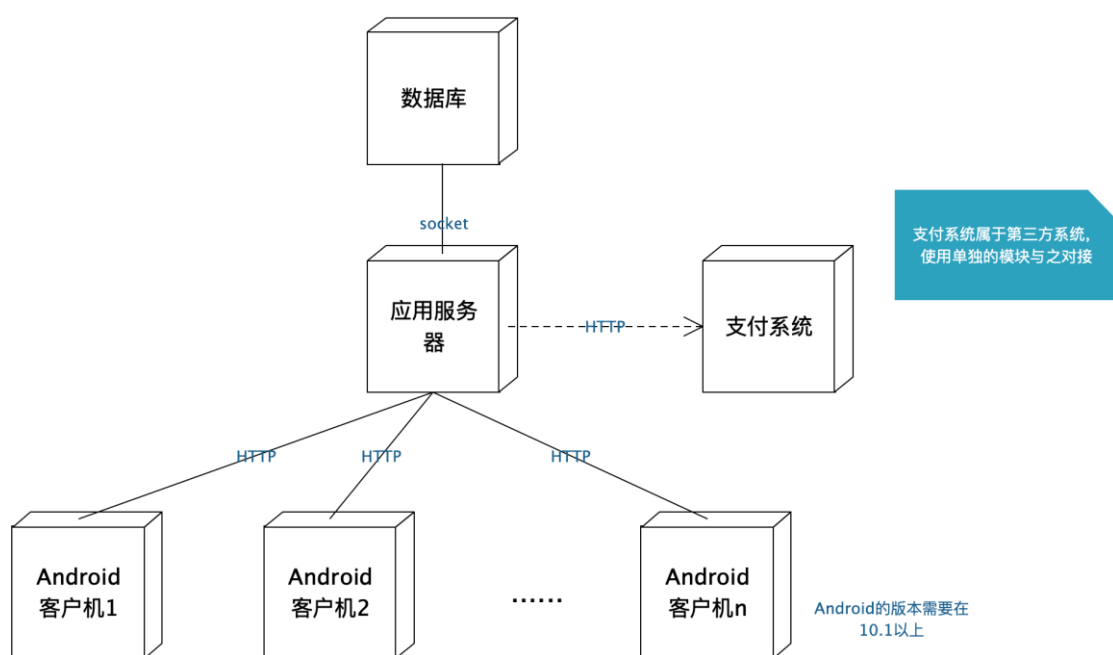


图 3.6.1 部署图

本项目的后端部署在 Centos 虚拟机上，数据库使用的是 MySQL，MySQL 服务器部署在和后端一样的服务器上，二者之间使用 socket 来进行通信。前端为 Android，将开发好的应用打包成 APK 在用户的 Android 机器上安装即可使用。

前后端之间的通信均使用 HTTP 的方式进行，交互方便。