**餐饮排队系统**

**需求规约**

Table of Contents

[1.介绍 2](#_Toc409213445)

[1.1文档概述 2](#_Toc409213446)

[1.2背景 2](#_Toc409213447)

[1.3 目标 2](#_Toc409213448)

[2.整体描述 2](#_Toc409213449)

[2.1项目特点&特色 2](#_Toc409213450)

[2.2运行环境 3](#_Toc409213451)

[2.2.1 服务器端要求 3](#_Toc409213452)

[2.2.2 客户端要求 3](#_Toc409213453)

[3.主要的功能需求描述 3](#_Toc409213454)

[3.1文字描述 3](#_Toc409213455)

[3.1.1用户功能用例 3](#_Toc409213456)

[3.1.2合作餐馆用例 3](#_Toc409213457)

[3.2用例模型描述 4](#_Toc409213458)

[3.2.1用户功能UseCase 图 4](#_Toc409213459)

[3.2.2商家功能Usecase图 9](#_Toc409213460)

[4.非功能性需求 13](#_Toc409213461)

[4.1性能要求 13](#_Toc409213462)

[4.1.1 精度 13](#_Toc409213463)

[4.1.2 时间特性要求 13](#_Toc409213464)

[4.1.3 输入输出要求 13](#_Toc409213465)

[4.2 数据管理能力要求 13](#_Toc409213466)

[4.3 故障处理要求 13](#_Toc409213467)

[4.4 灵活性 13](#_Toc409213468)

# 1.介绍

## 1.1文档概述

该文档简易描述了手机餐饮排队系统APP的需求规约，为进一步的概要设计和和需求分析说明书做基础。

## 1.2背景

如今全球化的时代，在类似于北京、上海这样的大城市，人们已经可以尝到来自全世界各地的美食。然而令广大美食爱好者头疼的是，凡是越热门的商家就越难预订位置和取号排队。

基于此，我们决定开发一个手机平台上的在线餐饮排队系统，可以不用一家地取号而开始排队。

本项目由组长张立权、组员胡亦成、常智辉、阮康乐共通提出及技术实现。

## 1.3 目标

我们的目标，是开发一个给餐馆提供一个平台，去展示餐馆风格，能让用户直接通过移动客户端进行排队，浏览菜单等，解决高峰期客户多，就餐难得问题。

我们主要设计了安卓平台智能手机的APP（供普通消费者使用）和餐馆网页端（供餐馆管理）。用户可以使用该APP来浏览附近支持该APP的餐饮商家，从而直接通过手机进行取号排队、浏览菜单、提前点单、结账等操作，大大节约商家和顾客的时间，减少人力成本和时间成本。

手机客户端部分，用户可以实现注册管理账号、在线取号排队、浏览菜单、提前点单、结账，评分等各项功能，商家可以对自己的店铺进行登记，菜品的注册管理，叫号管理等功能。

整个系统由运行于手机的客户端，运行于PC机的数据库和网页端的商家管理系统组成。

# 2.整体描述

## 2.1项目特点&特色

本项目最主要的特点是要实现一个多商家共用的手机取号排队系统，不同于其他设计已有类似功能的产品存在，目前还没有手机APP具有类似功能。此系统可以集在线取号排队、浏览菜单、提前点单、结账等功能于一体，提供给客户十分便捷的服务。

网络数据传输用JSON，以便于扩展到其他的应用。

## 2.2运行环境

### 2.2.1 服务器端要求

* 硬件部分：处理器P4,2.5GHz；内存512M(或更高)；VGA视频显示器；标准的鼠标、键盘；兼容印机。
* 软件部分：操作系统为Windows 7及以上 ；数据库系统为MySQL;

### 2.2.2 客户端要求

拥有安卓2.3.3以上版本系统的智能手机。

# 3.主要的功能需求描述

## 3.1文字描述

### 3.1.1用户功能用例

1、注册本系统用户账户

2、用账户进行登陆，查看个人信息

3、选择查看某范围的餐馆信息（包含排队人数、估计排队时间），查看菜单列表

4、把餐馆加入“喜欢的餐馆”项

5、选择餐馆进行排队

6、就餐时，查看菜单并点菜

7、查看历史就餐列表，添加评价

### 3.1.2合作餐馆用例

1、查看当前排队人数

2、修改菜单信息

3、修改菜馆信息

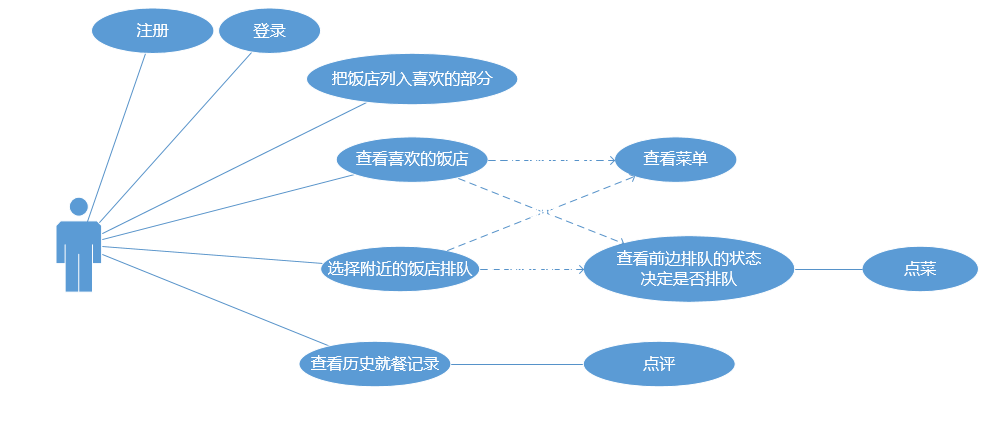
4、提醒用户就餐

5、查看用户点菜信息

6、查看用户就餐评价

## 3.2用例模型描述

### 3.2.1用户功能UseCase 图



#### UseCase#1:注册

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Use cese#1 | Register | | | |
| 描述 | 用手机号注册成为本产品用户 | | | |
| 前提 |  | | | |
| 触发条件 | 用户需要选择这个选项 | | | |
| 成功 | 创建新的用户账户，供用户使用 | | | |
| 中止 | 不创建新账户 | | | |
| 参与者 | 主要的 | | 用户 | |
|  | 从属的 | | 无 | |
| 过程 | step# | Short Name< action > | | 描述 |
|  | 1 | PromptAccount | | 提示用户输入账户和密码 |
|  | 2 | CheckAmount | | 检查用户输入的格式是否正确 |
|  | 3 | Save the account to the mysql server  Read the data | | 存放到数据库  返回用户id |
|  | 4 | Jump the account UI | | 跳转到账户信息界面 |
|  | 5 | Read the sql | | 读取用户信息 |
|  | 6 | Display the userInfo | | 显示用户信息 |
| 变更 | Step# | <action> | | 描述 |
|  |  |  | |  |
| 异常 | Step# | <action>或use case | | 描述 |
|  | 2a | BadForm | | 如果输入格式不正确 |
|  | 2b | Exit | | 重新输入账户信息 |

#### UseCase#2:登陆

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Use cese#1 | Login | | | |
| 描述 | 用户登录 | | | |
| 前提 | 有用户账户 | | | |
| 触发条件 | 用户需要选择这个选项 | | | |
| 成功 |  | | | |
| 中止 | 不登陆用户账户 | | | |
| 参与者 | 主要的 | | 用户 | |
|  | 从属的 | | 无 | |
| 过程 | step# | Short Name< action > | | 描述 |
|  | 1 | PromptAccount | | 提示用户输入账户和密码 |
|  | 2 | CheckAmount | | 检查用户输入的格式是否正确 |
|  | 3 | CheckLogin  Read the mysql | | 检测账户和密码是否对应存在,若存在则返回用户id |
|  | 4 | Jump the account UI | | 跳转到账户信息界面 |
|  | 5 | Read the sql | | 读取用户信息 |
|  | 6 | Display the userInfo | | 显示用户信息 |
| 变更 | Step# | <action> | | 描述 |
|  |  |  | |  |
| 异常 | Step# | <action>或use case | | 描述 |
|  | 1a | BadAccount | | 如果输入的账户和密码不匹配 |
|  | 1b | Exit | | 重新输入账户信息 |
|  | 2a | BadForm | | 如果输入格式不正确 |
|  | 2b | Exit | | 重新输入账户信息 |

#### Usecase3：查看附近的餐馆

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Use cese#1 | SearchRestNearby | | | |
| 描述 | 查看附近的餐馆 | | | |
| 前提 | 有用户账户 | | | |
| 触发条件 | 用户需要选择这个选项 | | | |
| 成功 | 显示附近的餐馆列表 | | | |
| 中止 |  | | | |
| 参与者 | 主要的 | | 用户 | |
|  | 从属的 | | 无 | |
| 过程 | step# | Short Name< action > | | 描述 |
|  | 1 | ViewRestNearby | | 参看附近的餐馆 |
|  | 2 | ChooseDistance | | 用户自选查看餐馆的距离 |
|  | 3 | Read the mysql | | 查询数据库，返回用户周围餐馆的信息 |
|  | 4 | Refresh UI | | 更新界面 |
| 变更 | Step# | <action> | | 描述 |
|  |  |  | |  |
| 异常 | Step# | <action>或use case | | 描述 |
|  | 1a | BadNet | | 如果网络问题 |
|  | 1b | Exit | | 提示用户网络异常 |
|  | 2a | NoRest | | 如果周围没有合作的餐馆 |
|  | 2b | Exit | | 提示用户没有餐馆 |

#### UseCase#4:选择餐馆排队

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Use cese#1 | JoinQueue | | | |
| 描述 | 选择餐馆排队 | | | |
| 前提 | 有用户账户 | | | |
| 触发条件 | 用户需要点击这个选项 | | | |
| 成功 | 进入排队状态，取消排队 | | | |
| 中止 | 取消排队 | | | |
| 参与者 | 主要的 | | 用户 | |
|  | 从属的 | | 无 | |
| 过程 | step# | Short Name< action > | | 描述 |
|  | 1 | JoinQueue | | 选择餐馆进行排队 |
|  | 2 | Save data | | 把userid和restid加入到数据库 |
|  | 4 | Refresh UI | | 刷新界面 |
| 变更 | Step# | <action> | | 描述 |
|  |  |  | |  |
| 异常 | Step# | <action>或use case | | 描述 |
|  |  |  | |  |

#### UseCase#5: 添加喜欢的餐馆

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Use cese#1 | AddLike | | | |
| 描述 | 添加喜欢的餐馆 | | | |
| 前提 | 有用户账户 | | | |
| 触发条件 | 用户需要点击这个选项 | | | |
| 成功 | 添加餐馆都喜欢列表 | | | |
| 中止 | 取消添加 | | | |
| 参与者 | 主要的 | | 用户 | |
|  | 从属的 | | 无 | |
| 过程 | step# | Short Name< action > | | 描述 |
|  | 1 | AddLike | | 用户点击添加选择的餐馆到列表按钮 |
|  | 2 | Save data | | 把userid和restid加入到数据库 |
|  | 4 | Refresh UI | | 刷新界面 |
| 变更 | Step# | <action> | | 描述 |
|  |  |  | |  |
| 异常 | Step# | <action>或use case | | 描述 |
|  |  |  | |  |

#### UseCase#5: 查看餐厅菜单

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Use cese#1 | ViewMenu | | | |
| 描述 | 查看餐厅菜单 | | | |
| 前提 |  | | | |
| 触发条件 | 用户需要选择这个选项 | | | |
| 成功 | 显示菜单列表 | | | |
| 中止 | 返回到附近的餐馆界面 | | | |
| 参与者 | 主要的 | | 用户 | |
|  | 从属的 | | 无 | |
| 过程 | step# | Short Name< action > | | 描述 |
|  | 1 | ViewMenu | | 用户点击查看菜单按钮 |
|  | 2 | SearchMenu | | 根据restid查询菜单列表 |
|  | 4 | JmpMenuUI | | 跳转到菜单列表 |
|  | 5 | DisplayMenu | | 显示菜单 |
| 变更 | Step# | <action> | | 描述 |
|  |  |  | |  |
| 异常 | Step# | <action>或use case | | 描述 |
|  | 1a | BadNet | | 如果网络问题 |
|  | 1b | Exit | | 提示用户网络异常 |

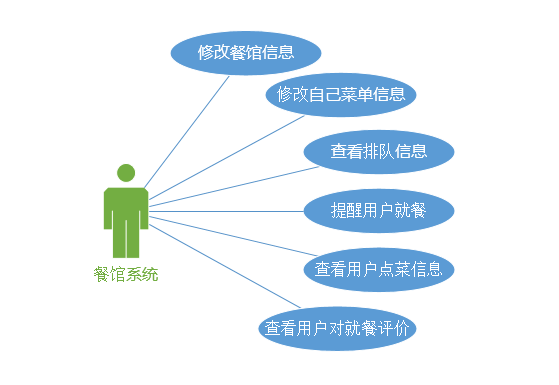
#### UseCase#6:查看历史就餐信息

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Use cese#1 | ViewPastEating | | | |
| 描述 | 查看历史就餐列表 | | | |
| 前提 |  | | | |
| 触发条件 | 用户需要选择这个选项 | | | |
| 成功 | 显示历史就餐列表 | | | |
| 中止 |  | | | |
| 参与者 | 主要的 | | 用户 | |
|  | 从属的 | | 无 | |
| 过程 | step# | Short Name< action > | | 描述 |
|  | 1 | ViewPastEating | | 用户点击查看历史就餐按钮 |
|  | 2 | SearchPastEating | | 根据userid查询历史就餐列表 |
|  | 4 | JmpPastEatingUI | | 跳转到历史就餐列表 |
|  | 5 | DisplayPastEatingInfo | | 显示过去就餐信息 |
| 变更 | Step# | <action> | | 描述 |
|  |  |  | |  |
| 异常 | Step# | <action>或use case | | 描述 |
|  | 1a | BadNet | | 如果网络问题 |
|  | 1b | Exit | | 提示用户网络异常 |

#### UseCase#6:评论

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Use cese#1 | Comments | | | |
| 描述 | 查看历史就餐列表 | | | |
| 前提 |  | | | |
| 触发条件 | 用户需要选择这个选项 | | | |
| 成功 | 对就餐进行评论 | | | |
| 中止 | 取消评论 | | | |
| 参与者 | 主要的 | | 用户 | |
|  | 从属的 | | 无 | |
| 过程 | step# | Short Name< action > | | 描述 |
|  | 1 | AddComment | | 用户输入评论内容 |
|  | 2 | Save data to mysql | | 把评论保存到数据库 |
|  | 4 | RefreshUI | | 更新界面 |
| 变更 | Step# | <action> | | 描述 |
|  |  |  | |  |
| 异常 | Step# | <action>或use case | | 描述 |
|  | 1a | BadNet | | 如果网络问题 |
|  | 1b | Exit | | 提示用户网络异常 |

### 3.2.2商家功能Usecase图



#### UseCase#1：修改餐馆信息

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Use cese#1 | ModifyRestInfo | | | |
| 描述 | 修改餐馆信息 | | | |
| 前提 |  | | | |
| 触发条件 | 用户需要选择这个选项 | | | |
| 成功 | 餐馆更新了自己信息 | | | |
| 中止 | 取消更新 | | | |
| 参与者 | 主要的 | | 餐馆人员 | |
|  | 从属的 | | 无 | |
| 过程 | step# | Short Name< action > | | 描述 |
|  | 1 | PromptRestInfo | | 提示用户输入餐馆信息 |
|  | 2 | CheckRestInfo | | 检查用户输入的格式是否合适 |
|  | 3 | Save data to sql server | | 把数据库数据更新 |
|  | 4 | RefreshUI | | 更新界面 |
| 变更 | Step# | <action> | | 描述 |
|  |  |  | |  |
| 异常 | Step# | <action>或use case | | 描述 |
|  | 2a | BadTime | | 如果输入的时间大于一周 |
|  | 2b | Exit | | 重新输入年月日 |

#### UseCase#2：修改餐馆菜单信息

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Use cese#1 | ModifyMenuInfo | | | |
| 描述 | 修改餐馆菜单信息 | | | |
| 前提 |  | | | |
| 触发条件 | 用户需要选择这个选项 | | | |
| 成功 | 餐馆更新了菜单信息 | | | |
| 中止 | 取消更新 | | | |
| 参与者 | 主要的 | | 餐馆人员 | |
|  | 从属的 | | 无 | |
| 过程 | step# | Short Name< action > | | 描述 |
|  | 1 | PromptMenuInfo | | 提示用户输入菜单信息 |
|  | 2 | CheckMenuInfo | | 检查用户输入的格式是否合适 |
|  | 3 | Save data to sql server | | 把数据库数据更新 |
|  | 4 | RefreshUI | | 更新界面 |
| 变更 | Step# | <action> | | 描述 |
|  |  |  | |  |
| 异常 | Step# | <action>或use case | | 描述 |
|  | 2a | BadTime | | 如果输入的时间大于一周 |
|  | 2b | Exit | | 重新输入年月日 |

#### UseCase#3：查看排队信息

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Use cese#1 | ViewQueueInfo | | | |
| 描述 | 查看排队信息 | | | |
| 前提 |  | | | |
| 触发条件 | 餐馆人员需要选择这个选项 | | | |
| 成功 | 查看餐馆排队信息 | | | |
| 中止 | 取消更新 | | | |
| 参与者 | 主要的 | | 餐馆人员 | |
|  | 从属的 | | 无 | |
| 过程 | step# | Short Name< action > | | 描述 |
|  | 1 | PressViewQueue | | 餐馆人员点击查看排队按钮 |
|  | 2 | Read data from mysql | | 从数据库读取数据 |
|  | 3 | RefreshUI | | 更新界面 |
| 变更 | Step# | <action> | | 描述 |
|  |  |  | |  |
| 异常 | Step# | <action>或use case | | 描述 |
|  |  |  | |  |

#### UseCase#4：提醒用户用餐

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Use cese#1 | AlertUser | | | |
| 描述 | 提醒用户用餐 | | | |
| 前提 |  | | | |
| 触发条件 | 餐馆人员需要选择这个选项 | | | |
| 成功 | 成功提醒 | | | |
| 中止 | 取消提醒 | | | |
| 参与者 | 主要的 | | 餐馆人员 | |
|  | 从属的 | | 无 | |
| 过程 | step# | Short Name< action > | | 描述 |
|  | 1 | PressAlert | | 餐馆人员点击提醒就餐按钮 |
|  | 2 | updata data to mysql | | 更新数据库数据 |
|  | 3 | RefreshUI | | 更新界面 |
| 变更 | Step# | <action> | | 描述 |
|  |  |  | |  |
| 异常 | Step# | <action>或use case | | 描述 |
|  | 2a | BadTime | | 如果输入的时间大于一周 |
|  | 2b | Exit | | 重新输入年月日 |

#### UseCase#5：查看用户点菜信息

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Use cese#1 | VieOrder | | | |
| 描述 | 查看排队信息 | | | |
| 前提 |  | | | |
| 触发条件 | 餐馆人员需要选择这个选项 | | | |
| 成功 | 查看餐馆用户点菜信息 | | | |
| 中止 | 取消更新 | | | |
| 参与者 | 主要的 | | 餐馆人员 | |
|  | 从属的 | | 用户 | |
| 过程 | step# | Short Name< action > | | 描述 |
|  | 1 | PressViewOrder | | 餐馆人员点击查看订单按钮 |
|  | 2 | Read data from mysql | | 从数据库读取数据 |
|  | 3 | RefreshUI | | 更新界面 |
| 变更 | Step# | <action> | | 描述 |
|  |  |  | |  |
| 异常 | Step# | <action>或use case | | 描述 |
|  |  |  | |  |

#### UseCase#6：查看用户评价信息

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Use cese#1 | VieComments | | | |
| 描述 | 查看排队信息 | | | |
| 前提 |  | | | |
| 触发条件 | 餐馆人员需要选择这个选项 | | | |
| 成功 | 查看餐馆用户点菜信息 | | | |
| 中止 | 取消更新 | | | |
| 参与者 | 主要的 | | 餐馆人员 | |
|  | 从属的 | | 用户 | |
| 过程 | step# | Short Name< action > | | 描述 |
|  | 1 | PressViewComments | | 餐馆人员点击查看评论按钮 |
|  | 2 | Read data from mysql | | 从数据库读取数据 |
|  | 3 | RefreshUI | | 更新界面 |
| 变更 | Step# | <action> | | 描述 |
|  |  |  | |  |
| 异常 | Step# | <action>或use case | | 描述 |
|  |  |  | |  |

# 4.非功能性需求

## 4.1性能要求

### 4.1.1 精度

该系统中没有对于较高数据精度的需要，时间单位精确到秒的整数即可，价格等系统中数字精确到0.1。数据存储和传输过程中与输入的保持一致。

### 4.1.2 时间特性要求

* 响应时间：对于用户输入的响应时间大体上决定于访问数据库的速度。
* 更新处理时间：自动更新，更新时间不应超过10秒。
* 数据的转换和传送时间：数据转换速度大体取决于用户数据库的传输速度。

### 4.1.3 输入输出要求

网络传输用JSON，便于扩展

## 4.2 数据管理能力要求

根据市场调查以及网上资料的搜集，按每家饭店有大约有50桌客人在就餐，100号在外等待排队点餐，平均每桌有10道菜，则平均每家饭店有1500-2000条需要储存并更新的记录。再计算上餐饮企业的数量，则这对于系统数据库提出了非常高的要求，一般的大型数据库如Oracle应该可以胜任。但由于这是课程设计项目，时间和经费都十分有限，所以采用了MySQL来做初步实现。

## 4.3 故障处理要求

软件故障：数据库管理系统出现故障，可能发生数据丢失，这就需要系统DBA切实做好数据备份工作，在数据库发生故障时，能够迅速的给予恢复，保证系统的正常运行。

## 4.4 灵活性

* 操作方式的变化：该系统为多模块的系统，如需改变一些操作方式如添加功能，在原有的框架下添加和修改即可，无需大量的代码重写。
* 同其他软件接口的变化：该系统为独立的系统，只要满足基本的软硬件需求，其他软硬件接口条件对该系统没有太大的影响。
* 精度和有效时限的变化：关于精度修改数据库和软件代码中变量等的定义即可。有效时限可以进行适度的加长或缩短，并不影响整体运行。
* 计划的变化或改进：如果项目出现计划变化和改进，小组成员开会协调处理。