**咖啡厅叫餐系统**

**需求说明书**

**组名：天码行空**

**编写人：陈岐望何强林坚谢宇山袁泓玮**

**编写日期：年月7日**

### 1 引言

**1.1 项目目的**

为明确项目要求，合理分配组内成员任务，确保本次项目按质按量完成，编写本文。

**1.2 项目背景**

a. 本项目来自java程序设计实践课程。

b. 需求分析由天码行空团队共同完成。其中陈岐望负责整理，及第一节、第二节的编写，谢宇山完成第三节编写，林坚，袁泓玮完成第四、五节的编写，何强完成第六节、第七节、第八节的编写。整理由陈岐望完成。

c. 本项目将由天码行空团队共同开发。其中陈岐望负责项目总负责人，何强为前端组负责人，谢宇山为后端组负责人。

**1.3名词解释**

前端：指用户直接接触的功能，如交互、数据呈现等功能。

后端：指对用户完全透明的数据库查询、表单处理和各种请求等。

**1.4 参考资料**

无

### 2 任务概述

**2.1目标**

为某咖啡馆开发一套集成用户在线点餐、工作人员在线管理等功能的点餐系统。

**2.2假定与约束**

本次项目开始日期为2020年6月22日，结项日期为 2020年7月23日，历时约5周。项目还分为两个阶段：第一阶段为系统基础功能开发阶段，提交日期为2020年7月8日，历时约2周；第二阶段为用户功能开发阶段，提交日期为2020年7月23日，历时约3周。

无经费预算。

### 3 数据描述

数据分为静态数据和动态数据。所谓静态数据，指在运行过程中主要作为参考的数据，它们在很长一段时间内不会变化，一般也不会随着运行而改变，所谓动态数据，包括所有在运行中要发生变化的数据，以及在运行中要输入、输出的数据。

**3.1静态数据**

列出所有作为控制或参考用的静态数据，并给出名称。

1. 用户信息，包括用户名、密码、电话、邮箱、性别等；
2. 管理员信息，包括职员号、职位、权限等；
3. 餐品静态信息，包括餐品名称、类型（饮料、小吃、主食）、实物展示图等；

**3.2动态数据**

列出所有动态数据，并给出名称。

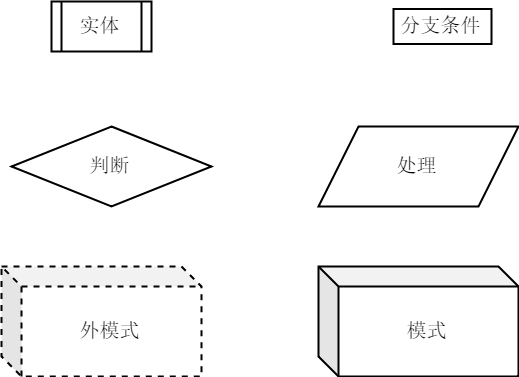
1. 餐品动态信息，包括餐品库存数量、定价、成本价等；
2. 订单数据，包括订单产生时间、餐品名、单品数量、订单总额、订单状态、订单处理人等；
3. 用户购物车数据，包括餐品名，餐品数量，总金额；
4. 营业数据，包括餐点选购频率统计、下单时间统计、客户满意度统计等；
5. 管理员动态信息，包括权限更改、操作记录、订单处理记录等。

### 功能需求

本节是我们分析需求的重点章节，包括流程图、功能划分、功能描述、数据与功能的对应关系等。

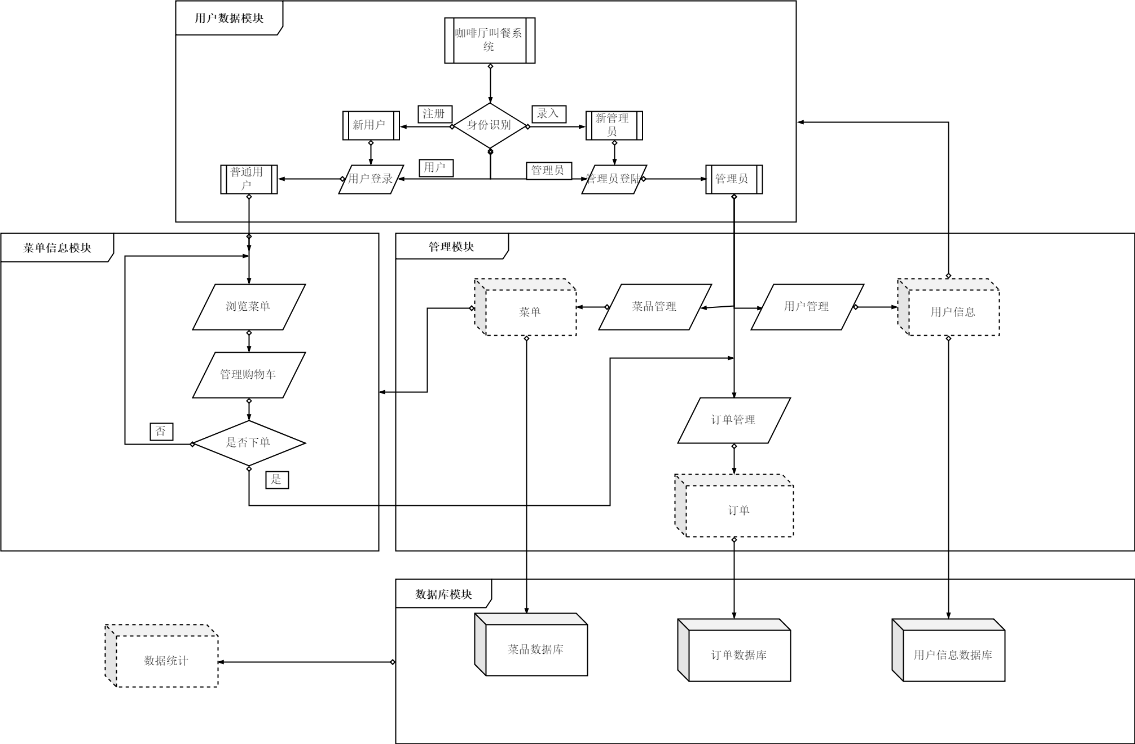
**4.1流程图**

**4.1.1 符号说明**



图一 流程图图例

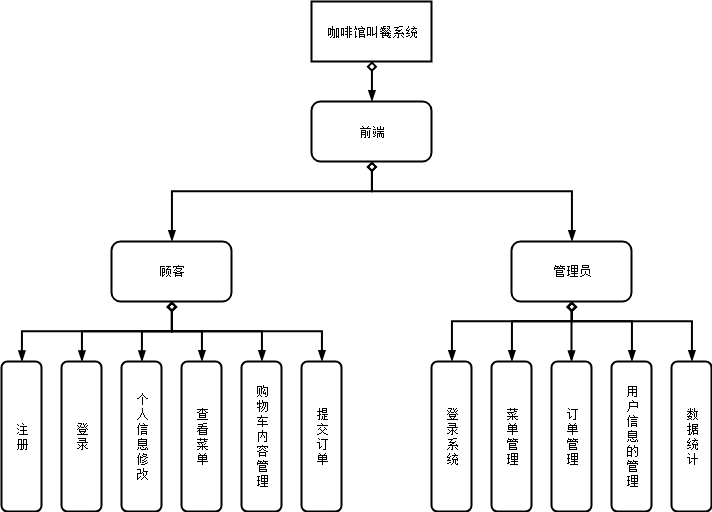
**4.1.2 系统总流程图**



图二 总流程图

**4.2功能划分**

本节是我们对系统功能的分析。



图三 系统信息图

a. 注册：用户输入用户名（不重复）和密码，输入电话，生日信息等。

b. 登录：顾客根据注册时的用户名和密码登录系统。

c. 个人信息修改：用户可以选择修改预留在店里的个人信息。

d. 查看菜单：可以查看现阶段店里现有菜品的信息（包括图片和价格，已被购买的份数），以及现阶段菜品受欢迎程度排行。

e. 购物车内容管理：用户在最终提交订单之前，都可以对购物车里的内容做修改和查看。

f. 提交订单：点击确认按钮后，根据购物车里商品的情况生成订单，提交到后端。

g. 菜单管理：根据商品数据库里商品的数量来更新菜单，已售罄的商品在菜单上删除或者标记已售罄。

h. 订单管理：每当一个顾客提交订单，添加该订单到存放订单的数据库，也可以对该数据库中内容进行查看和删除。

i. 用户信息的管理：当用户注册时，添加用户信息到存放用户的数据库，当用户修改时，数据库中内容同步修改，可以查看用户信息

j. 数据统计：可以根据历史订单中商品数量，对于商品受欢迎程度排序，将排行榜展示给顾客。

**4.3功能描述**

对最底层的功能所要完成的功能进行详细描述，填入下表中：

|  |  |  |
| --- | --- | --- |
| 功能名称 | 功能标识符 | 功能详细描述 |
| 用户注册 | User\_Register | 根据前端信息，在用户信息数据库中新增用户信息 |
| 用户登录 | User\_Login | 用户输入信息，进行登录 |
| 显示菜单信息 | Display\_Menu | 登陆成功后返回用户序号，并展示菜单 |
| 接受购物单 | Accept\_List | 接受前端发来的购物单，并处理购物单的食物种类和数量；并处理用户取消订单的请求 |
| 更新商品数据库 | Update\_ItemDB | 根据购物单的信息，修改商品数据库的库存数据 |
| 更改用户信息 | Change\_Userinfo | 根据前端信息，修改用户信息 |
| 历史订单存储 | Order\_Storage | 记录前端传来并且未被取消的订单，保存在历史信息数据库中 |
| 更新菜单 | Update\_Menu | 根据前端信息，修改菜单的种类、名称等信息 |
| 查看用户信息 | Check\_Userinfo | 根据前端信息，返回餐厅管理员要求查看的用户信息 |
| 管理用户 | Administer\_Uesr | 根据前端信息，执行餐厅管理员对用户的删除和信息修改 |
| 管理订单 | Administer\_Order | 向管理员发送新创建的订单；并据前端信息，执行餐厅管理员对用户订单的操作（更改食物信息，删除订单等） |

表一 底层功能对照

**4.4数据与功能的对应关系**

我们用一张矩阵图说明功能描述中的各个功能与数据描述中的静态数据、动态数据之间的对应关系。

|  |  |  |
| --- | --- | --- |
| 功能标识符 | 输入 | 输出 |
| User\_Register | 注册时需要填写的信息 | 注册成功的账户；并记录于后台数据库中 |
| User\_Login | 用户名和密码 | 向前端返回登录状态，后端记录userID |
| Display\_Menu | 用户名和密码 | 当前食物菜单 |
| Accept\_List | 前端发来的购物单 | 用户名，商品种类，商品数量 |
| Update\_ItemDB | 购物单中的用户名，商品种类，商品数量 | 在后端商品数据库中进行修改 |
| Change\_Userinfo | 前端发来，某用户想要改变的字段修改后的值 | 在后端用户信息数据库中进行对应修正，向前端返回修改成功的信息 |

（接上表）

|  |  |  |
| --- | --- | --- |
| Order\_Storage | 某一个下单成功的订单 | 把该订单增加至历史订单数据库 |
| Update\_Menu | 想要修改的菜单中的商品的字段值 | 在后端的菜单数据库中完成对应的修改 |
| Check\_Userinfo | 输入要查看信息的用户的用户名或序列号 | 返回该用户的注册信息 |
| Administer\_User | 管理员希望进行的用户操作（如删除，增加） | 返回操作对应的结果 |
| Administer\_Order | 管理员希望进行的订单操作（如修改，查看） | 返回操作对应的结果 |

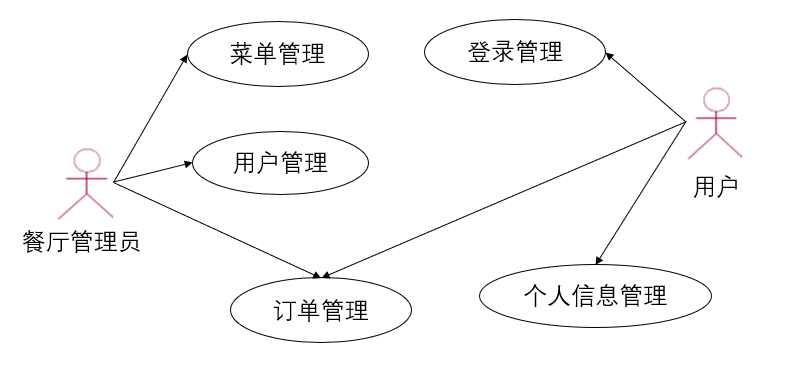
表二 数据功能对照

### 5 用例模型

本节是根据上节的功能需求，编制出的用例图和用例规约。

**5.1 用例简图**

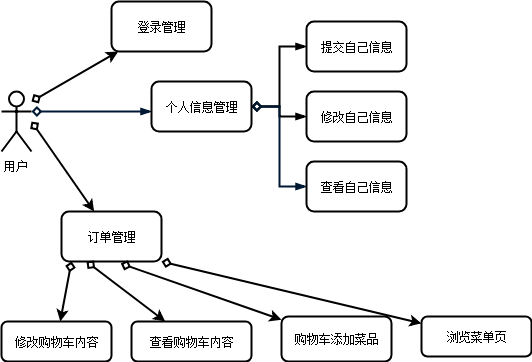
图四主要用于说明用户和餐厅管理员之间的数据使用关系。



图四 用例模型简图

**5.2 用户用例详图**

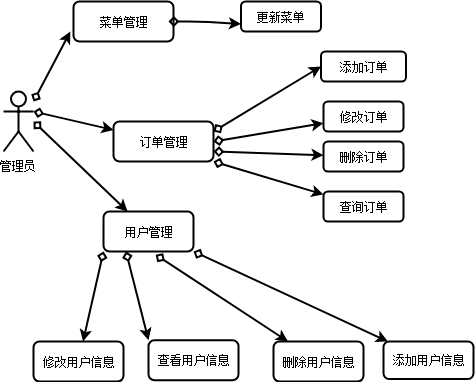
图五用于说明用户用例的具体功能，补充了关于个人信息、订单管理的具体内容，分别包括提交、修改、查看个人信息，购物车内容更改和历史订单浏览的功能。



图五 用户用例详图

**5.3 管理员用例详图**

图六用于说明管理员用例的具体功能，补充了订单管理、菜单管理、用户管理的具体内容。其中订单管理包含订单的添加、修改、删除、查询等操作，用户管理包括用户信息的修改、查询、删除、添加等功能。



图六管理员用例详图

### 性能需求

性能需求主要包括时间要求、适应性、数据精确度、完整性等，下面我们分别进行叙述。

**6.1时间要求**

* 1. 数据更新处理时间不大于5秒；
  2. 页面刷新时间不大于2秒；
  3. 数据转换和传送时间：按传送的数据量而定；
  4. 不受网速限制下，进入页面时间不大于1秒；
  5. 运行时间没有限制。

**6.2适应性**

网页浏览要求有较强的适应性，要求系统能用键盘和鼠标操作，能在Android，IOS等系统下正常加载、显示。

通过各种网页开发工具提供的不同分辨率预览，即可以简单验证网页设计的合理和正确。

**6.3数据精确度**

要求数据的小数点能按实际需求定位，字符形变量精确无误。可以对合法和非合法的输入值是否符合试剂要求进行响应并做出规定。

每一个需求的源流必须清晰，在进一步产生和改变文件编制时，可以方便地引证每一个需求。

**6.4完整性**

系统支持全部有意义的需求，无论是关系到功能的、性能的、设计约束的、还是关系到外部接口方面的需求，对所有可能出现的输入数据的响应予以定义，填写全部插图、表、图示标记等；定义全部术语和度量单位。

### 运行环境描述

**7.1支持软件**

* 1. 操作系统：无具体要求，支持IE9以上内核的浏览器即可
  2. 浏览器：IE9以上内核或其它主流浏览器。
  3. Web Server：Tomcat 8.5.55
  4. 数据库：MySql 8.0.17

**7.2接口**

**7.2.1硬件接口**

无特殊的硬件接口；

**7.2.2软件接口**

操作系统、数据库、工具、函数库、集成的商业组件等。

**7.2.3通讯接口**

1. WEB浏览器；
2. 网络通讯标准或者协议，要求安装TCP/IP协议；
3. 数据交互用电子表格；
4. 存在安全加密等方式保证通讯数据信息安全或加密问题；
5. 数据传输速率较高；
6. 具有同步或异步通讯机制；

**7.3控制**

**7.3.1输入/输出控制：**

对必输项和输入的类型进行提示，提交时要求重新输入，保证数据的正确性；列出输入数据(用户输入、来自外部接口的输入或者其它输入)并且定义针对这些输入数据的处理(计算方法，以及相应地输出数据，描述对应区别:输入数据和输出数据)。

当有大量数据需要描述时，也可以分类描述数据并且注明各项数据的输入、输出属性。对于每一项数据，均需要描述:a. 数据名称；b. 实际含义；c. 数据类型；d. 数据格式；e. 数据约束。

**7.3.2复杂计算处理**

对于复杂的处理方法，仅仅给出算法原理是不够的，必须描述详细的计算过程，并且列出每一步具体使用的实际算式;如果计算过程中涉及查表、判断、迭代等处理方法，应该给出处理依据和相关数据。如果计算方法很简单，也可以将其从略，不加描述。

**7.3.3异常处理**

对出现网络中断、系统崩溃等现象时，数据不会被破坏。

**7.3.4应用日志处理：**

1. 应用日志必须在现有基础上进行精简，对于重复日志记录、无用日志记录进行刪减；
2. 应用日志必须按日保存，每个文件保存应用一天的完整日志；
3. 历史应用日志必须压缩后按日保存，以减小历史日志文件对硬盘空间的占用；
4. 历史应用日志必须保留一个月(30天)，以保证问题出现后，日志可以得到较好的保存。

**7.4用户界面**

1. 将要采用的图形用户界面（GUI）标准或者产品系列的风格应该接近大部分用户所能接受的范围；
2. 存在对有关页面布局或者解决方案的限制的控制方法或设备；
3. 将要使用在每一个页面（图形用户界面）的软件组件，可能包括：选单、标准按钮、导航链接、消息栏、各种功能组件等；
4. 具有各种显示格式的规定，可能包括：不同情况下文字的对齐方式、不同情况下数字的表现格式与对齐方式、日期的表现方法与格式、计时方法与时间格式等；
5. 具有统一的错误信息显示标准；
6. 用户界面要求友好、简明，层次清晰，用网页实现分级菜单。

### 其他需求

其他需求主要包括可用性、安全保密性、可维护性、可跨平台性等，下面我们分别进行叙述。

**8.1可用性**

1. 要求系统能有效恢复和再启动等，可以对系统出错点进行检查；
2. 软件对用户的误操作可以进行准确清晰的提示并确认是否继续执行；
3. 系统软件对重要数据的删除给与确认提示并确认是否执行；
4. 软件是否能判断数据的有效性，屏蔽用户的错误输入，识别错误值，并有相对应的错误提示；
5. 操作直观简便并且合理

**8.2安全保密性**

这里指的是保护整个软件系统的要素，以防止各种非法的访问、使用，修改、破坏或者泄密。其具体需求必须包括：

1. 利用可靠的密码技术
2. 文档对主要功能和关键操作提供使用示例
3. 使用日志和阅文记录来掌握特定的记录或历史数据集
4. 文档是否齐全，包含使用所需的信息和所有的功能模块
5. 给不同的模块分配不同的角色
6. 限定资源库中信息在某些人之间的通信

**8.3可维护性**

这里规定若干需求以确保系统软件是可维护的。系统软件的可维护性体现在：

1. 知识库分类和论坛分类的可维护
2. 用户信息的可维护
3. 系统功能出现故障，存在相应的解决方案
4. 其它相关字典可维护
5. 操作者使用不方便，不影响正常功能的实现
6. 提供网页模板，直接文本输入就可产生网页

**8.4可跨平台性**

要求系统用JavaScript作为前端脚本，便于把系统从一种环境移植到另一种环境。

系统软件可支持多种操作系统包括Windows、Linux、Android、IOS等，并正常运行各种功能。