**酒店订餐推荐系统**

《需求说明书》

|  |  |
| --- | --- |
| 组长 | 阳欣佩 |
| 组员 | 邓凯中 |
| 组员 | 郭展鹏 |
| 组员 | 杨律明 |

|  |  |
| --- | --- |
| 专业 | 大数据 |
| 班级 | 56班 |
| 教师 | 张影 |

# 绪论

**（算法图片示例在最后）**

## 传统点餐方式

有酒店便有菜谱，菜名、套餐、图片、介绍等内容仔细设计、排布、装订，便有了印刷精美的菜谱册。纸质菜谱，顾名思义，将菜谱打印在纸质材料上进行展示，满足展示菜品的需求。可是当菜品上新时，又得重新打印新菜谱。

传统点餐方式，无法记录历史数据。这就不能为菜品的改进以及推荐决策做出指导。久而久之，会使企业在竞争中处于下锋。

由此可见餐饮行业迫切需要一个智能点餐推荐系统。

## 餐饮菜品推荐

在智能餐饮条件下，可以更加方便地展示商家的“招牌菜”，“特价菜”，“最新菜品”等营销策略，从而提升商家的利润。顾客也可以进行个性化点餐，顾客可以根据自己的预算和实际情况自主选择，这些都可以通过智能点餐以及推荐算法进行控制和实现。

由此可见，在数据就是价值的时代，智能点餐系统如果能和服务人员配合完成更加人性化的点餐服务，一方面可以帮助就餐者快速而准确地找到感兴趣的菜品，节省点餐时间，使用户的体验更佳；另一方面还可以提高餐饮的点餐效率，增加销量，降低服务人员成本，大幅提高企业的盈利。

## 1.3目的

制作一款性能良好，功能完善，界面友好的智能点餐系统。主要功能是根据客户的不同需求，智能推荐出宴席菜单中用户喜欢的菜，方便用户选菜。

# 系统综述

## 2.1系统功能用例文档

### 2.1.1 用例图

|  |  |
| --- | --- |
| 类型 | 用例图 |
| 用户 | IMG_256 |
| 管理员 |  |

### 2.1.2 用例描述

|  |  |  |  |
| --- | --- | --- | --- |
| 用例编号 | 用例名 | 用例概述 | 主要参与者 |
| Case1 | 浏览菜品 | 客户浏览所有的菜品 | 客户 |
| Case2 | 分类浏览 | 客户浏览根据类别浏览菜品 |
| Case3 | 查看菜品详情 | 客户浏览菜品细节信息 |
| Case4 | 选择套餐 | 客户可以选择固定套餐 |
| Case5 | 查看套餐详情 | 客户浏览套餐信息 |
| Case6 | 下单 | 客户订购自己所选的套餐 |
| Case7 | 登录 | 管理员点击登录按钮登录 | 管理员 |
| Case8 | 查看所有菜品和套餐 | 管理员查看所有菜品和套餐 |
| Case9 | 修改菜品数据 | 管理员可以增加或者删除菜品 |
| Case10 | 修改套餐数据 | 管理员可以增加或者删除套餐 |

## 2.2 数据库设计

### 2.2.1 表结构

1.菜品

|  |  |
| --- | --- |
| 表单栏目 | 描述 |
| 菜品编号ID | 主键、必填、自增 |
| 菜品名称 | 必填 |
| 菜品类别 | 必填 |
| 菜品价格 | 选填 |
| 菜品图片 | 选填 |
| 菜品简介 | 选填 |
| 菜品订购量 | 选填 |
| 菜品现价 | 选填 |

1. 菜品-订单

|  |  |
| --- | --- |
| 表单栏目 | 描述 |
| 订单ID | 外键、必填 |
| 菜品ID | 外键、必填 |

1. 订单

|  |  |
| --- | --- |
| 表单栏目 | 描述 |
| 菜品编号ID | 外键、必填 |
| 用户名 | 外键、必填 |
| 套餐ID | 外键、必填 |
| 价格 | 选填 |
| 实价 | 选填 |

1. 套餐

|  |  |
| --- | --- |
| 表单栏目 | 描述 |
| 套餐ID | 主键、必填、自增 |
| 套餐名 | 选填 |
| 点单量 | 选填 |
| 价格 | 选填 |
| 实价 | 选填 |

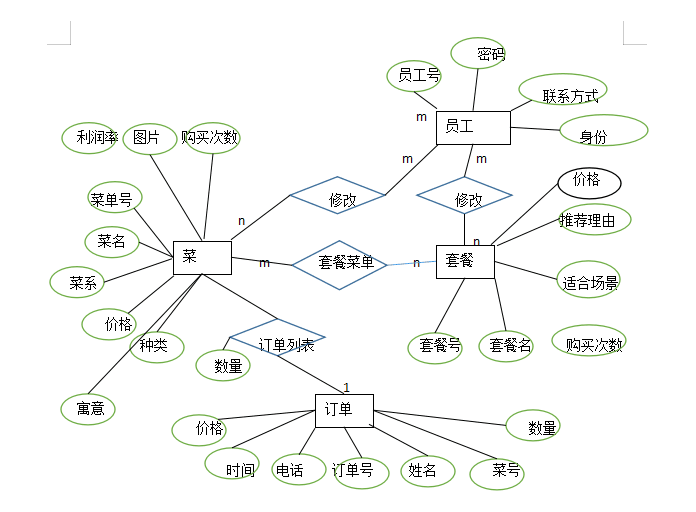
1. 菜品-套餐

|  |  |
| --- | --- |
| 表单栏目 | 描述 |
| 套餐ID | 外键、必填 |
| 菜品ID | 外键、必填 |

1. 管理员

|  |  |
| --- | --- |
| 表单栏目 | 描述 |
| 编号ID | 主键、必填、自增 |
| 用户名 | 必填 |
| 密码 | 必填 |

## 2.2.2 E-R图



# 性能要求

## 3.1 可用性

* 系统客户端

系统客户端采用电脑和手机Web浏览器，不需要安装额外的软件。

* 系统易用性

符合约定俗成的使用风格与习惯，即使是不熟练电脑操作的人也能使用该系统。

## 3.2 可靠性

* 工作时间

系统每天24小时，每周七天运行。

* 系统严重错误发生的时间间隔

系统发生严重错误小于3%。

## 3.3 性能要求

* 最大系统响应时间

系统的所有操作0.1秒内响应；

* 最大事务处理时间

本系统用户事务的最大处理时间不大于30秒，如果超过这个时间系统应该自动结束用户事务处理。

## 3.4 系统运转性能要求

* 数据冗余少，共享程度高；
* 具有可扩展性和可修改性；
* 能够充分描述数据间的内在联系。
* 系统的[用户接口](http://baike.baidu.com/view/1309799.htm)简单，用户容易掌握，使用方便；
* 数据具有独立性。数据和程序相互独立提高开发的速度
* 能够保证系统可靠运行，有良好的健壮性，非权限用户无法访问，可以防止错误数据的产生，一旦产生也可以及时发现及修正。

## 3.5 运行环境

* 服务器类型：Tomcat 9.0
* 网络类型：公网

存储容量：内存容量：4G或以上内存；外存容量：30G以上

* 操作系统：Windows 7 Professional及以上、linux操作系统
* 数据库：MySQL
* Web浏览器

Chrome、Firefox等主流浏览器

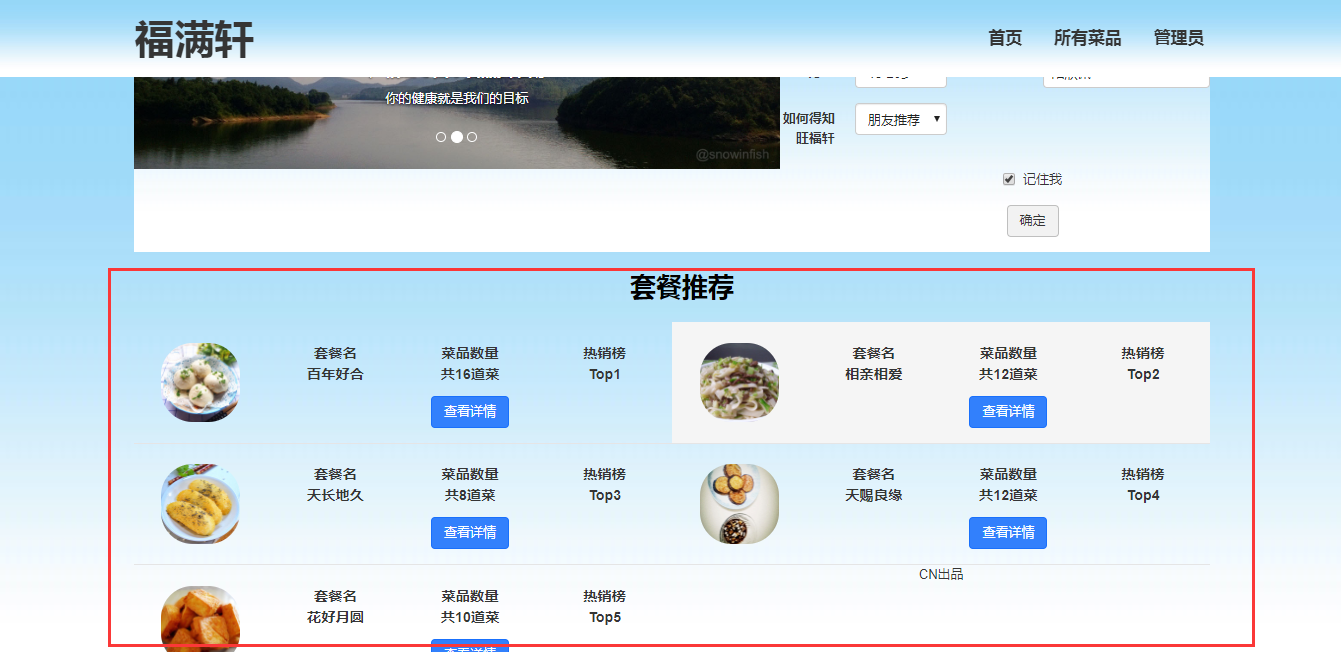
* 运行软件：eclipse等java编译环境

### 3.6 运用技术

* 前端技术：JS、JQuery等前端框架
* 后端技术：java EE、python等
* 数据库技术：MySQL，redis数据库

### 3.7 运用算法

首页：套餐推荐-根据所选场景推荐销量最高的套餐（婚宴、满月酒。。）



所有菜品

1. 基于用户协同过滤
2. 关联规则分析（redis存储规则）
3. 购物车相近套餐推荐





管理员

1.智能定价（KNN-回归-----根据输入菜品成分和以往订单销量最好的菜品成分，切分分类定价）



2.预测是否会下单（KNN-分类----根据用户首页所填信息）



1. 销量数据分析（pyecharts 图表分析）



