

i 食堂系统

详细设计说明书

文件状态：	文件标识：	
<input type="checkbox"/> 草稿	当前版本：	1.1
<input checked="" type="checkbox"/> 正式发布	作 者：	林泰来
<input type="checkbox"/> 正在修改	完成日期：	2020.04.09

版本历史

版本/状态	作者	参与者	起止日期	备注
1.0	林泰来		2020.04.05- 2020.04.05	开始编写
1.1	陈秉宥		2020.04.09- 2020.04.09	排版，添加层次图

目录

1	引言	4
1.1	编写目的	4
1.2	背景	4
1.3	参考资料	4
1.4	定义及说明	4
2	设计概述	5
2.1	任务和目标	5
2.1.1	需求概述	5
2.1.2	运行环境概述	5
2.1.3	条件与限制	5
2.1.4	详细设计方法	6
3	系统详细需求分析	7
3.1	详细需求分析	8
4	系统详细设计	8
4.1	系统程序代码架构设计	8
4.1.1	UI (User Interface) 用户界面表示层	9
4.1.2	BLL (Business Logic Layer) 业务逻辑层	9
4.1.3	DAL (Data Access Layer) 数据访问层	9
4.1.4	Common 类库	9
4.1.5	Entity Class 实体类	9
4.2	系统结构设计及子系统划分	10

1 引言

1.1 编写目的

- 1). 本文档是“i食堂系统”的设计说明书。
- 2). 本文档供所有与此系统相关的人员使用,包括设计人员、开发人员、测试人员、维护人员。
- 3). 方便用户与设计、开发人员之间的信息交流,确定开发者的基本设计实现思路想法是否满足实现条件。
- 4). 本文档可以作为系统分析员进行系统详细设计的出发点。
- 5). 本文档可以作为系统分析员向程序员分配代码设计任务的依据。

1.2 背景

- 1). 待开发的软件系统的名称: i食堂系统
- 2). 开发者: 铁华组
- 3). 用户: 系统管理员、高校食堂窗口商家、高校教职工、高校学生
- 4). 实现该软件的计算中心或计算机网络: 计算中心
- 5). 该软件系统同其他系统或其他机构的基本的相互来往关系: 数据相互调用关系

1.3 参考资料

- 1). GB/T8567——2006《计算机软件文档编制规范》;
- 2). 《软件工程导论(第五版)》张海藩编著 清华大学出版社
- 3). 《图书馆信息管理系统——需求分析与规格说明》
- 4). 《软件工程》李浪、朱雅莉、熊江主编 华中科技大学出版社;
- 5). 《软件文档写作教程》马平、黄冬梅编著 电子工业出版社;
- 6). 《青院图书馆管理规章制度》
- 7). 《图书管理系统设计说明书》@IT学无止境著 CSDN论坛;

1.4 定义及说明

非功能性需求: 所有用户在使用本系统之前都必须通过自己的用户名和密码登录,才能进行其他操作。该子系统主要负责判断登录时判断用户名和密码的正确性。

2 设计概述

2.1 任务和目标

2.1.1 需求概述

1) 商家信息输入输出：商家编号，每个商家申请的用户账号，用于登入商家端系统；密码，使商家登录的必须前提，只有商家账号和密码都输入正确才能登陆i食堂商家端系统开始使用；商家名称，每个商家都有一个唯一的名称，它是唯一有效区分商家的方式；商家图片，每户商家的展示图，是区分商家的方法之一；经营人，地址，经营许可证，注册时间，如果商家的注册时间超过了三年，则需要商家申请更新商家信息，是区分商家是否具有经营许可的标准；已售数目，当前月份售出的数量；库存量，是一户商家每道菜品的库存数量的数量；菜品编号，它是唯一有效区分菜品的方式；菜品名称，是顾客区分菜品的有效方法之一；菜品展示图，是顾客区分菜品的有效方法之一；菜品说明，它是顾客了解菜品信息的方式之一；菜品价格，它是顾客了解菜品信息的方式之一；

2) 顾客输入信息输出：顾客编号，每个顾客都有一个唯一的编号，通常为学生学号或是教职工的教职工号，是有效区分顾客的唯一标识；密码，使用户登录的必须前提，只有顾客编号和密码都输入正确才能登陆i食堂顾客端系统开始使用；用户名，每个顾客的真实姓名，是区分用户的方式之一；性别，年龄，院系，专业，电话，地址，是用户资料的组成部分，用来完善用户资料，方便管理员进行管理；入校时间，离校时间，是顾客的有效期范围，只有在校用户才能使用该系统，用户从入学时分配顾客编号，直到用户离校此编号作废，不能再使用；

3) 订单输入输出：顾客编号，商家编号，菜品名称，菜品价格，是查看顾客所预订菜品信息是否正确的方式；下单时间，用户下单的时间；预订时间，用户预订的时间；

4) 评论输入输出：评论编号，每条都有一个唯一的编号，它是唯一有效区分评论的方式；顾客编号，商家编号，菜品编号；评价内容，评价的主要内容；发送时间，评价发送的时间；总体评价、态度评价、价格评价、美味评价，是顾客对菜品的评价分数；等待时间，顾客等待的时间；

2.1.2 运行环境概述

运行平台：Android 5.0以上

处理器：性能达到高通骁龙625或海思麒麟710及以上

内存：1GB

硬盘：不少于32GB

2.1.3 条件与限制

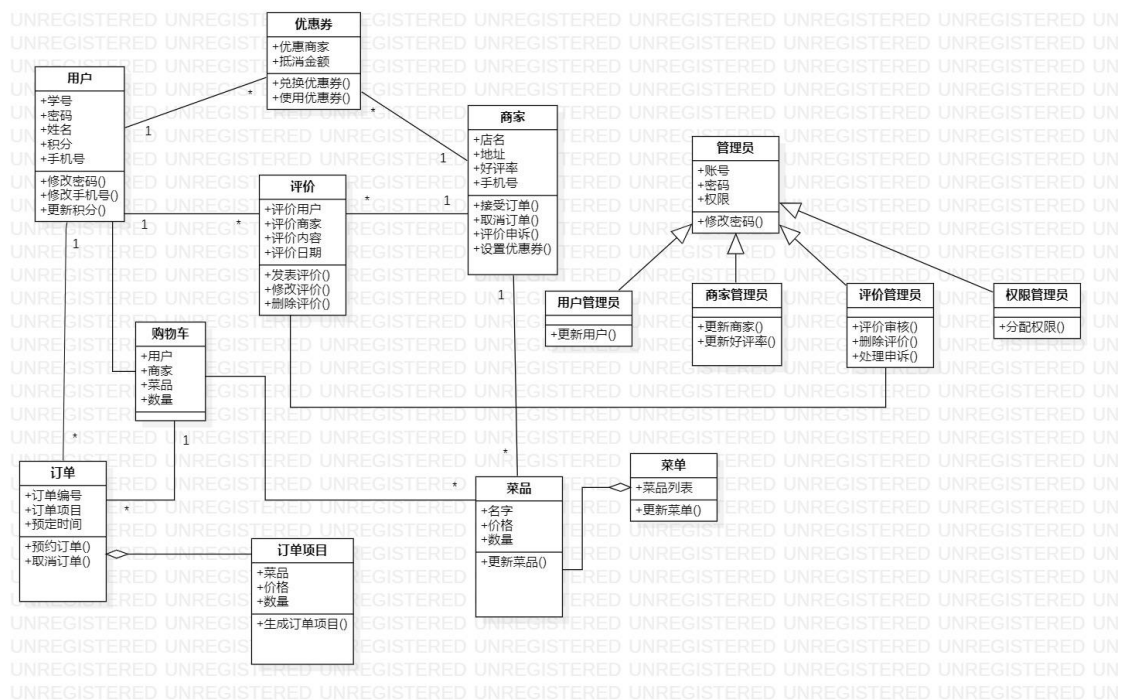
1) 内部限制：客户端系统开发过程需严格按照运行环境所要求的性能和内存要求进行开发，i食堂是一款面向全体高校学生和教职工的app，因此需要让尽可能多的用户都能流畅运行i食堂客户端的app，降低用户的使用门槛。

2) 外部限制：商家端的操作应尽可能简单，降低商家的使用成本，尽可能避免商家因使用过程过于繁琐而拒绝接入i食堂平台的情况。

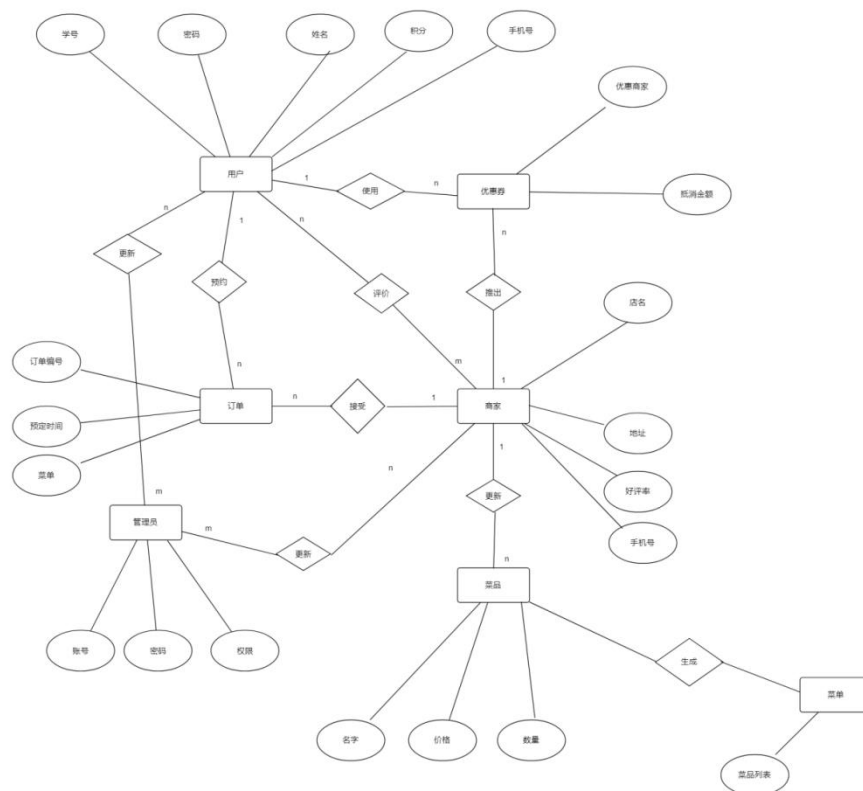
3) 进度限制：整个开发过程应在2020年6月7日前完成。

2.1.4 详细设计方法

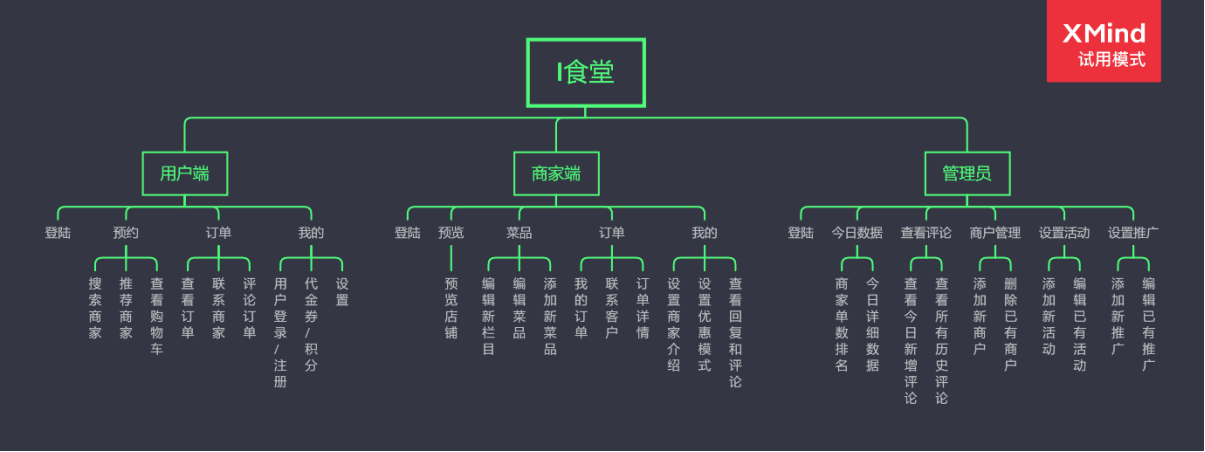
类图：



ER图：



功能模块层次图



3 系统详细需求分析

主要对系统级的需求进行分析。首先应对需求分析提出的企业需求进一步确认，并对由于情况变化而带来的需求变化进行较为详细的分析。

3.1 详细需求分析

1) 详细功能需求分析：

i: 商家在我们的平台上架自己的窗口及菜品并填上每道菜所需的制作时间，然后给予我们一定的代金券发放权，我们帮助他们进行宣传他们的窗口。

ii: 学生在我们的平台上可以查看食堂每个窗口的排队人数以及等待时间还有各食堂的口碑，然后可以更加精确地根据自己的时间、对菜品质量和价格的要求进行选择自己想要的菜品，然后进行预订。因为堂食往往菜量比外卖大且价格更低，所以学生在一般情况下更愿意堂食。在堂食结束后可以对所点的菜品口味、分量、价格、等待时间进行点评，我们制定了一套较为严谨的评分体系，可以让我们平台的评分更加真实。虽然严谨的评分体系可能会使用户评分过程变得繁琐，但我们还设置了积分、代金券奖励机制来鼓励学生评论和点赞他人的优秀评论。

2) 详细性能需求分析：

i: 易用性指标分析: i食堂平台设计与开发的目的是为了提高学生前往食堂就餐的效率，为不同的参与者用户提供实用的功能，所以系统必须贴近实际的应用流程，尽量符合用户的操作习惯，真正做到帮助用户提高活动参与和管理效率。同时还应该考虑到不同操作者的手机使用水平不同，在系统的设计时要注重易用性，使大多数的使用者都可以较好地使用系统。

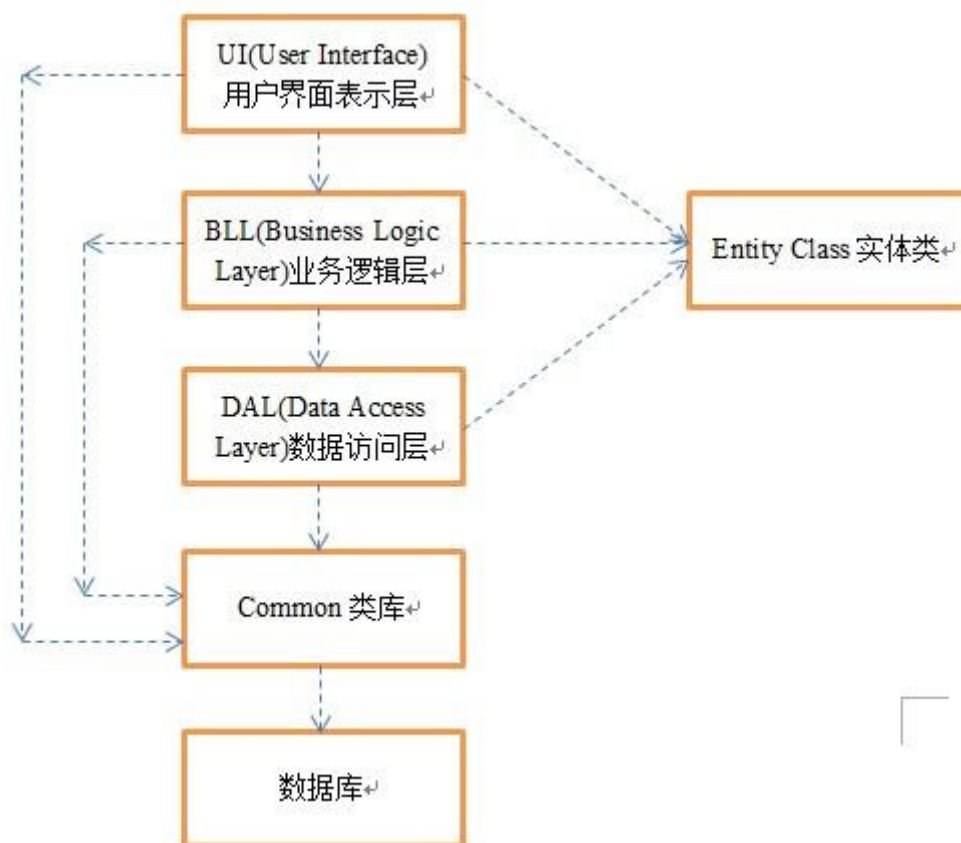
ii: 可扩展性指标分析: i食堂平台目前的设计是以当前需求为目标的，但是由于顾客需求，商家需求可能会希望使用时添加更多功能，为了更好地服务用户，必须要考虑到系统的扩展性问题，要达到增加新功能模块时不会对原有系统架构和功能模块造成太大影响的目标，同时又能够根据实际需求的变化对系统的功能进行扩展。

iii: 系统健壮性指标分析: 系统的健壮性也包括系统的稳定性，功能强大的系统如果在运行的过程中时常出现问题，也会对用户造成较大的困扰，所以必须要提高系统的稳定性。同时，用户在对系统进行操作时，由于各种原因会进行误操作，或者输入错误的数据等，系统应能够对这些情况进行处理，系统应保证个别模块出现问题不会对其他模块造成影响。

4 系统详细设计

4.1 系统程序代码架构设计

系统采用三层架构模型，将应用系统划分为用户界面表示层、业务逻辑层、数据访问层，以及 Entity Class 实体类、Common 类库组成，各层的关系如下图所示：



4.1.1 UI (User Interface) 用户界面表示层

负责与用户进行交互，显示、接受数据，与此同时，做一些简单逻辑处理，如：输入数据有效性判断、显示各种异常、处理Dataset记录集数据。它只与BLL (Business Logic Layer) 业务逻辑层、Entity Class 实体类两个项目发生关联，可能与Common类库发生关联。

4.1.2 BLL (Business Logic Layer) 业务逻辑层

是整个系统的核心，它承担了所有的逻辑判断，实现了程序的功能，它是灵活的。BLL层既是调用者，又是被调用者，因此，要适当的进行设计达到解耦的效果。BLL层只关联DAL层和实体类，可能关联Common类库。虽然BLL层被U层调用，但是BLL层无需关心UI层的情况。数据库中每个表都对应一个BLL类，为了达到解耦效果，BLL类不能直接调用其他表的DAL类，可以BLL类之间相互调用。

4.1.3 DAL (Data Access Layer) 数据访问层

提供数据访问的接口，没有任何逻辑。在接口中对数据库操作语句进行组合装配。DAL层一般关联Common类库中的最底层，最基础的数据库类（比如：链接数据库），必须关联Entity Class 实体类项目。DAL层只是数据库的管理者，但不是访问者，不直接与数据库发生关联。数据库中每个表都对应一个DAL层的接口（访问控制）类。

4.1.4 Common类库

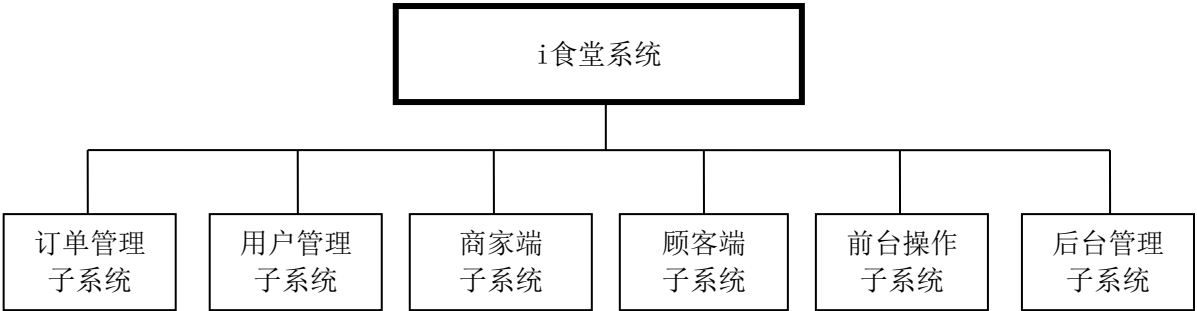
用于存放公用的类。最常用的就是数据库访问类，比如：链接字符串，数据库引擎类。它直接与数据库进行机械式的交换，无任何逻辑。

4.1.5 Entity Class 实体类

相当于加强的数据结构，实现了对数据的封装。数据库中每个表都对应一个实体类，表的字段就是实体类的属性，类型一一对应。UI、BLL、DAL这三层的交互主要就是通过实体类作为参数，并Return回信息。

4.2 系统结构设计及子系统划分

根据业务和功能，将系统的逻辑结构划分为订单管理子系统、用户管理子系等6个子系统，如下图所示：



各个子系统按照功能角度分解，划分出若干不同的功能模块，如下面各图所示：

- 1) ． 订单管理子系统：该子系统主要负责订单的创建、查询和删除功能的实现。
- 2) ． 用户管理子系统：包括顾客和商家信息的添加、查询、修改、删除等功能。
- 3) ． 商家端子系统：该子系统主要负责商家管理窗口上的菜品信息，还支持商家查询已处理和未处理的订单信息，对用户反馈的订单问题进行审核。
- 4) ． 顾客端子系统：该子系统主要负责顾客管理自己的已选菜品和预约时间，还支持顾客查询自己完成和未完成的订单信息，允许顾客对订单信息存在的问题进行反馈。
- 5) ． 前台操作子系统：该子系统主要负责提供食堂中可预约的菜品供顾客选择；如果该菜品可预约，则正常预约，提示“预约成功”；如果该菜品不可预约，则弹出窗口提示。评价时需提供详细的评分说明以及菜品图；若顾客积分达到一定数额，点击“兑换代金券”按钮即可兑换所需的代金券；
- 6) ． 后台管理子系统：该子系统主要负责添加、查询、修改、删除所有用户的信息，还支持管理员查看商家信息、顾客信息、订单信息，发放代金券和积分，调节订单矛盾等功能。