



餐厅预订网站

软件设计文档

项目组成员：梁智昊 15331187

林 淼 15331202

廖颖泓 15331191

刘 阳 15331218

刘俊君 15331214

明友芬 15331242

林泽云 15331207

目录

1. 引言.....	1
1.1 编写目的.....	1
1.2 背景.....	1
1.3 定义.....	1
1.4 参考资料.....	2
2. 总体设计.....	2
2.1 需求规定.....	2
2.1.1 网站相关规定.....	2
2.1.2 用户功能说明.....	3
2.1.3 网站界面说明.....	3
2.2 运行环境.....	7
2.3 基本设计概念和处理流程.....	7
2.3.1 注册.....	8
2.3.2 登录.....	9
2.3.3 预订.....	11
2.3.4 取消订单.....	12
2.3.5 查看订单.....	14
2.3.6 管理顾客信息.....	15
2.4 结构.....	18
2.4.1 视图层.....	18
2.4.2 控制层.....	18
2.4.3 数据层.....	19
2.5 功能需求与程序的关系.....	20
2.6 人工处理过程.....	21
2.7 尚未解决的问题.....	21
3. 设计技术.....	21
3.1 Object-oriented Programming.....	21
3.2 Aspect-oriented Programming.....	21
3.3 Service Oriented Architecture.....	22

3.4 Design Patterns.....	22
4. 接口设计.....	22
4.1 用户接口.....	22
4.2 外部接口.....	22
4.3 内部接口.....	22
5. 模块划分.....	23
5.1 用户模块设计.....	23
5.2 商家管理员模块设计.....	24
6. 运行设计.....	24
6.1 运行模块组合.....	24
6.2 运行控制.....	24
6.3 运行时间.....	25
7. 系统出错处理设计.....	25
7.1 出错信息.....	25
7.2 补救措施.....	25
7.3 系统维护设计.....	25

1. 引言

1.1 编写目的

编写这份文档是为了在充分理解需求的基础上，为后续工作做准备的。对餐厅预订系统的结构进行概要设计，明确模块划分和界面设计。文档面向建设餐厅预订系统的系统程序员，帮助程序员安排开发计划。本手册将进行技术评审和技术的可行性检查，它是详细设计和编码的重要根据，为下一步的详细设计说明提供框架同时也是用户进行交流的文档之一。软件开发小组的每一位参与开发成员应该阅读本说明，以清楚产品在技术方面的要求和实现策略。

1.2 背景

通过对生活经验和调查结果的总结，我们发现餐厅的人流量不一定会给顾客和餐厅管理人员带来了一定程度的不便。对顾客而言，临时决定到店可能遭遇高峰期，需要花费长时间等待上菜；对餐厅管理人员而言，人流量的未知会导致食材的浪费或者不足，让顾客长时间等待则会降低服务质量。

因此，为了达到减少顾客就餐等待时间、提高餐厅服务质量和服务效率的目的，本项目组将设计和开发一个餐厅预订网站。该网站采用 MVC 架构，统一管理顾客信息和顾客订单信息。本网站设定两种角色，分别为管理员和顾客，支持管理、预订、交互功能。该网站的用户是餐厅的经理（餐厅网站负责人员）和想要在该餐厅就餐的顾客，

待开发系统的名称：餐厅预订网站；

本次项目的任务提出者：软件设计综合实验小组全体成员；

开发者：软件设计综合实验小组全体成员；

用户：使用本系统的餐厅网站负责人和顾客。

1.3 定义

表 1-1 术语表如下所示：

表 1-1 术语表

术语	含义
顾客	网站的使用者之一，可使用网站进行预订座位和预订菜品
管理员	网站的使用者之一，管理用户信息和订单信息
顾客信息条目	列表，每个条目包含手机号，密码，订单号
管理员信息条目	列表，每个条目包含手机号，密码
座位列表	已预订列表和未预订列表，管理员可对列表进行查看删除操作
座位订单	包含订单联系人信息、座位类型、订单时间
菜品订单	包含订单联系人信息、菜品类型及数量

1.4 参考资料

表 1-2 参考资料表

标题	编号	发行日期	发行单位	来源
Hypertext Transfer Protocol--HTTP/1.1	RFC 2616	1999 年 6 月	万维网协会和互联网工程任务组	http://tools.ietf.org/html/rfc2616
Information technology Document description and processing languages HyperText Markup Language (HTML)	ISO/IEC 15445:2000	2000 年 5 月	国际标准化组织和国际电工委员会	https://www.cs.tcd.ie/misc/15445/15445.HTML

2. 总体设计

2.1 需求规定

2.1.1 网站相关规定

(1) 订座时间分为 2 个时间段：中午（11:00-14:00）晚上（17:00-21:00）；

- (2) 餐厅可选座位为 40 个；
- (3) 顾客可自由选择任何座位；
- (4) 订座是订餐的前提，订餐可选；
- (5) 顾客只能当天预订座位，逾期网站将自动取消其所定座位。

2.1.2 用户功能说明

使用该网站的角色分为两类：顾客，管理员。管理员使用权限大于顾客。

顾客可使用功能如下：

(1) 注册：输入唯一标记的合法登录手机号、合法登录密码即可成功注册，用户数据存储在数据库中可供下次自动登录，并由管理员进行管理。

(2) 登录：已经注册的用户使用手机号和对应的密码进行登录，即可跳转到网站主界面。

(3) 预订：包括预订座位和预订菜品，预订座位需要选择预订时间、预订桌型，可选择确认或取消；预订菜品需要选择菜品类型，可选择确认和重置。

(4) 查看订单：可查看当前订单和历史订单，餐桌信息和已预订的菜单。

(5) 取消订单：顾客可根据实际情况取消未消费订单。

注：任何未注册的顾客不具有“预订座位”、“取消订座”、“预定菜品”和“查看订单”4 种功能。未注册顾客可以访问网页首页信息。

管理员可使用的功能如下：

(1) 登录：管理员点击网页首页“管理员”，输入员工登录用户名以及唯一的登录密码即可进入管理员模式。

(2) 查看订单：可查看任何顾客的订单信息，包括：顾客就餐时间、联系、顾客对应的餐桌信息和菜单。

(3) 修改订单：管理员可以修改顾客的订单。

(4) 取消订单：管理员可以根据具体情况强制取消用户订单，拥有最高权限。

(5) 添加订单：管理员可以根据具体情况添加顾客信息和顾客的订单，拥有最高权限。

(6) 删除用户：管理员可以删除用户，拥有最高权限。

2.1.3 网站界面说明

该网站分为以下 7 个界面：注册登录界面、首页、管理员、预订座位、预订菜品、查看订单。

(1) 登录界面

这个界面默认是登录界面，如果没有注册的话点击“没有账号，立即注册”来是注册，也可以选择管理员模式登录。

手机号

密码

登录

取消

管理员登录

没有账号，立即注册 >

图 2-1 登录界面

(2) 注册界面

如图 1-2 所示，未注册用户输入手机号码和密码后，点击最下面的“注册”按钮即可进行注册。

手机号

密码

确认密码

立即注册

返回登录

图 2-2 注册界面

(3) 管理员登录界面

如图 1-3 所示为管理员的登录界面，管理员输入账号和密码，网站认证通过后即可进入管理员界面。

管理账号

密码

管理员登录

取消

图 2-3 管理员登录界面

(4) 管理员界面

在如下图 1-4 所示的管理员界面可以查看顾客预订的产品、修改顾客信息及顾客订单信息、删除顾客订单以及删除用户等功能。管理员拥有最高的权限。



图 2-4 管理员界面

(5) 主页

如图 1-5 所示为主页，在这个界面可以查看订单和预订，在页面的右上角会显示出顾客的手机号码，并且可以选择退出当前系统。



图 2-5 主页

(6) 预订/取消座位界面

如图 1-6 所示，选择之后点击右边“选择时间”按钮选择时间，选择餐桌类型，最后点击界面下方的“确认订单”和“取消订单”两种按钮来选择订座还是取消订座。左上角的主页可以点击回到主页。



（7）订餐界面

如图 1-7 所示，用户在预定座位完成之后，点击预订菜品即可进入到如下图所示的订餐界面，顾客选择相应的菜品完成菜品的预订，在页面上会显示出所预定的菜品的总的价格。



（8）查看订单界面

如图 1-8 所示，顾客进入之后，即可查看当前订单，订单显示餐桌类型，菜单，联系人信息。页面上的删除订单按钮，点击之后可以删除当前订单。



图 2-8 查看订单界面

2.2 运行环境

客户机：

系统硬件支持环境：window、Linux 与 Mac 均可运行；

系统支持环境：VUE + NODEJS + MONGODB。

服务器机：

系统硬件支持环境：window、Linux 与 Mac 均可运行；

系统支持环境：VUE + NODEJS + MONGODB。

2.3 基本设计概念和流程

本系统的基本设计概念：MVC 架构

本系统的处理流程：

（1）顾客模式：注册 -> 登录 -> 预订/取消预订 -> 退出登录

（2）管理员模式：登录 -> 添加/删除/修改顾客或订单信息 -> 退出登录

顾客模式与管理员模式下每一个功能的具体流程与解释如下所示：

2.3.1 注册

（1）简要描述

注册即要求顾客输入登录手机号、登录密码。该网站会从后台的已有的用户名条目中检查用户输入的用户名是否存在，如果存在则拒绝用户以此手机号来注册，如果该手机号没有使用过则可成功注册。

（2）参与者

顾客

（3）场景描述

顾客打开本网站，点击“注册”按钮，进入注册页面。用户需要输入新的登录手机号、登录密码。网站使用数据库中的顾客信息条目核对用户输入的手机号是否存在，如果存在，页面弹出窗口，页面提示“手机号已存在”，如果手机号在数据库中不存在，则提示注册成功并返回首页。

（4）前置条件

顾客成功打开首页。

（5）后置条件

顾客和管理员注册失败会有窗口提示，登录成功会进入新页面。注册成功后，后台数据库存入用户信息。

(6) 事件流

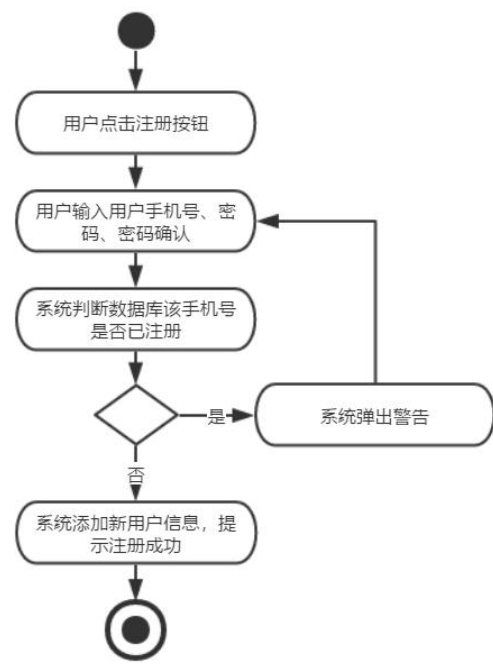


图 2-9 注册活动图

2.3.2 登录

(1) 简要描述

登录即要求顾客输入登录手机号、登录密码。管理员登录则使用特定的管理员账号。该网站会从后台的登录手机号条目中检查该手机号和密码是否匹配,如果匹配成功则登录成功,否则给出错误提示。

(2) 参与者

顾客、管理员

(3) 场景描述

顾客点击“登录”按钮,进入登录页面。输入登录手机号和登录密码。网站根据后台的顾客信息条目核对登录手机号和登录密码是否匹配,若匹配成功,页面跳转到首页,顾客可使用预订/取消、查看订单功能。若匹配失败,页面给出错误提示。

管理员点击“管理员”按钮,进入登录页面。输入唯一管理员用户名和唯一的密码、平

台使用后台的管理员条目检查管理员信息是否存在，若存在，页面进入“管理员界面”，管理员可以查看顾客订单信息和删除顾客订单，否则给出错误提示。

（4）前置条件

顾客和管理员要打开本网站，并且已经成功注册过。

（5）后置条件

顾客和管理员登录失败会有错误提示，登录成功则进入新页面。

（6）事件流

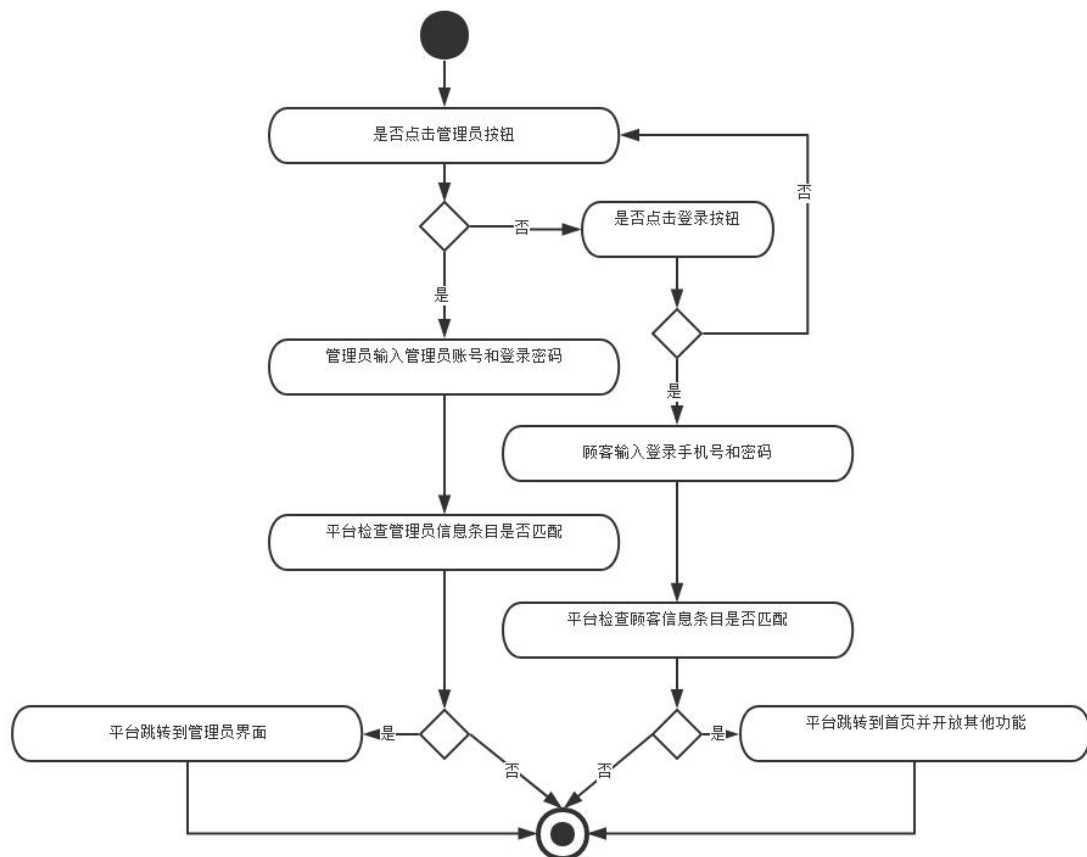


图 2-10 登录活动图

2.3.3 预订

（1）简要描述

预订座位即在顾客成功登录了本网站以后，选择尚未被预订的座位进行预订。当顾客成功预订座位后，可以使用预订菜品功能。

（2）参与者

顾客

（3）场景描述

顾客成功登录了本网站以后，在网站的主页面能够看到预定座位和预定菜品的按钮。本网站根据餐厅实际情况提供餐桌和菜品的类型信息，顾客根据需要进行选择。顾客可以根据需要选择不同座位数的桌子，即餐桌类型，并且需要选择就餐时间，提供的就餐时间选择是：中午（11:00-14:00）晚上（17:00-21:00），选择后可以确认或取消。用户成功预订座位后可进行菜品预订，菜品预订需要选择菜品类型。但是菜品预订是可选项，即可以只预订座位不预订菜品，但是预订菜品之前必须先预订座位。

（4）前置条件

顾客成功登陆、选择的座位未被预订

（5）后置条件

可预订的座位数量减少，更新数据库状态。

(6) 事件流

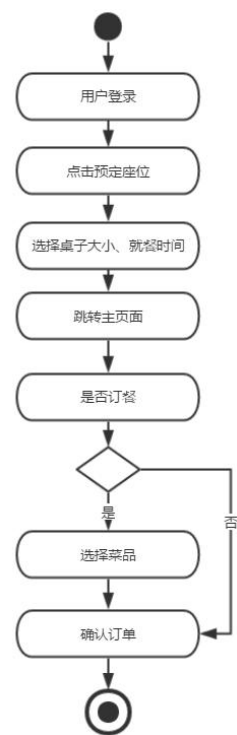


图 2-11 预订活动图

2.3.4 取消订单

(1) 简要描述

本用例描述顾客或者管理员对已选座位的取消过程：

顾客取消订位：当顾客已经成功的完成了餐桌的预定了，但是因为顾客自身的原因想要取消订位的时候就可以选择取消订位，在顾客取消之后在后台数据库和取消座位界面都会有相应的更改；

管理员取消订位：当餐厅出现特殊状况不能为顾客提供餐饮服务的时候，管理员可以取消任何顾客的订座，并且在后台数据库实时的更新对餐桌空位的更改。同时在取消座位界面也会实时的显示出来。

(2) 参与者

顾客，管理员

(3) 场景描述

取消座位分为顾客取消和管理员取消两种情形：

顾客取消：顾客已经预定好了餐桌位置，但是顾客想要取消预定。如果此时顾客已经退出了网站，那么顾客就需要重新打开平台网页以一般模式登陆进去，如果顾客没有退出登录的话就不需要重新登录。顾客点击网站首页上的“订座/取消”按钮，进入“订座/取消界面”，选中顾客已经预定的座位之后再点击“取消”，就可以在后台数据库更改顾客的订位信息，取消订位，并且返回到取消了订位后的“预定/取消座位界面”，将顾客已经取消的座位显示为空。完成操作之后顾客可以选择重新预定座位，或者退出登录。

管理员取消：管理员打开网站以管理员模式登陆进去，选中需要强行取消的顾客订单，再点击“取消”，就可以在后台数据库中取消顾客的订位，完成操作之后即可选择退出登录。

(4) 前置条件

顾客已经成功订座或者有需要管理员强行取消的订座。

(5) 后置条件

后台数据库会实时的更新顾客和管理员对订单的更改，并且“订座/取消界面”也会实时的显示出更改后的界面。

(6) 事件流

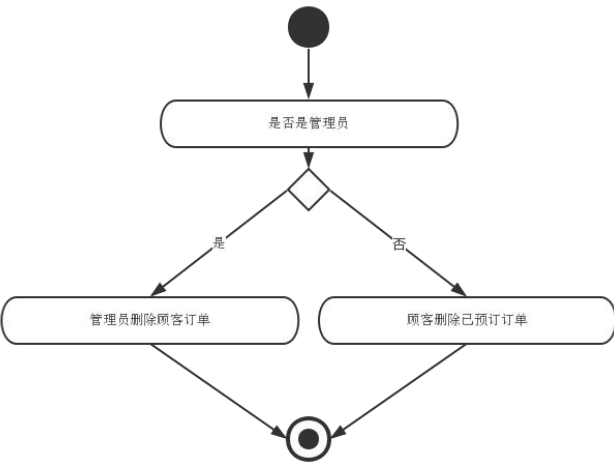


图 2-12 取消订单活动图

2.3.5 查看订单

（1）简要描述

顾客成功登录的情况下，可以查看当前订单和历史订单，订单显示餐桌类型，联系电话，用餐时间和菜单。管理员在成功登录的情况下，可以查看每个用户的订单。

（2）参与者

顾客，管理员

（3）场景描述

对于顾客，在登录的情况下可以查看自己的订单，订单情况分为两种，一种是已完成（即历史订单），一种是进行中（当前有效订单），每一条订单信息包含预订餐桌类型、用餐时间、顾客联系电话。

对于管理员，在登录的情况下可以查看每个用户的订单信息，包括顾客联系方式，用餐时间，桌型，菜单。

（4）前置条件

顾客和管理员只有在已经注册并且成功登录的情况下才能够查看订单信息。

（5）后置条件

当数据库中的餐桌预约情况发生更改时，显示的订单信息实时发生改变。

(6) 事件流

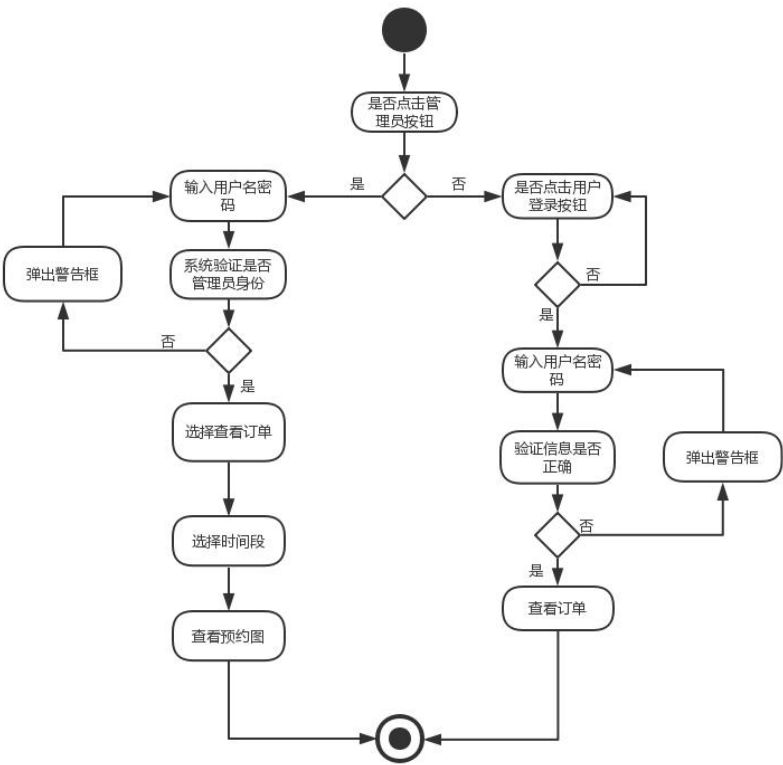


图 2-13 查看订单活动图

2.3.6 管理顾客信息

(1) 简要描述

顾客信息维护即根据系统已注册的所有用户信息，对部分用户进行必要的管理员操作。具体操作为添加未注册的新用户、删除数据库中已注册用户、更新已注册用户信息。

(2) 参与者

管理员

(3) 场景描述

管理员进入顾客管理页面，对顾客信息进行添加、删除和修改。用户登录系统，系统验证用户管理员身份。如果身份验证成功，用户以管理员身份登录成功。如果身份验证失败，用户页面弹出窗口，页面提示“非管理员无法登录”，返回登录界面。用户登录成功后，系

统列出用户表中所有注册用户用户名以及选座信息，用户选择选择所需要进行的操作，如果需要添加新用户，系统首先检查该用户是否存在，如果不存在，则系统向出具看写入新的用户信息，如果存在，则系统无法对数据库进行操作。如果需要删除列表中的旧用户，系统弹出对话框，确认是否删除，得到确认后，系统从数据库中删除已选中的用户。如果需要修改列表中的旧用户，用户输入修改后的用户信息，系统检查输入的信息是否合法，检查通过后系统将从数据库中删除旧用户的信息并添加新的用户替代该用户。

(4) 前置条件

本用例开始前用户登录餐厅管理员账号

(5) 后置条件

数据库完成对应条目的添加、删除、更新

(6) 事件流

事件流如图 2-14 所示。

2.4 结构

整个网站采用 MVC 架构，分为视图层（View）、控制层（Control）、数据层（Model）。MVC 架构设计将业务逻辑、数据和界面显示分离开来。把相应的数据模型封装起来，通过视图层与用户进行交互，并且通过控制层实现层与层之间的逻辑控制。这样可以使得模型设计、界面设计、逻辑设计分离开来，而开发人员不用在改进和个性化界面及用户交互的同时重新编写业务逻辑。该架构的使用实现了分层管理和模块化，简化了后续对程序的修改和扩展，并且使程序某一部分的重复利用成为可能，从而提高开发效率。系统架构如下图 2-15 所示。

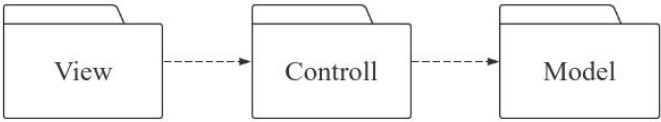


图 2-15 系统架构图

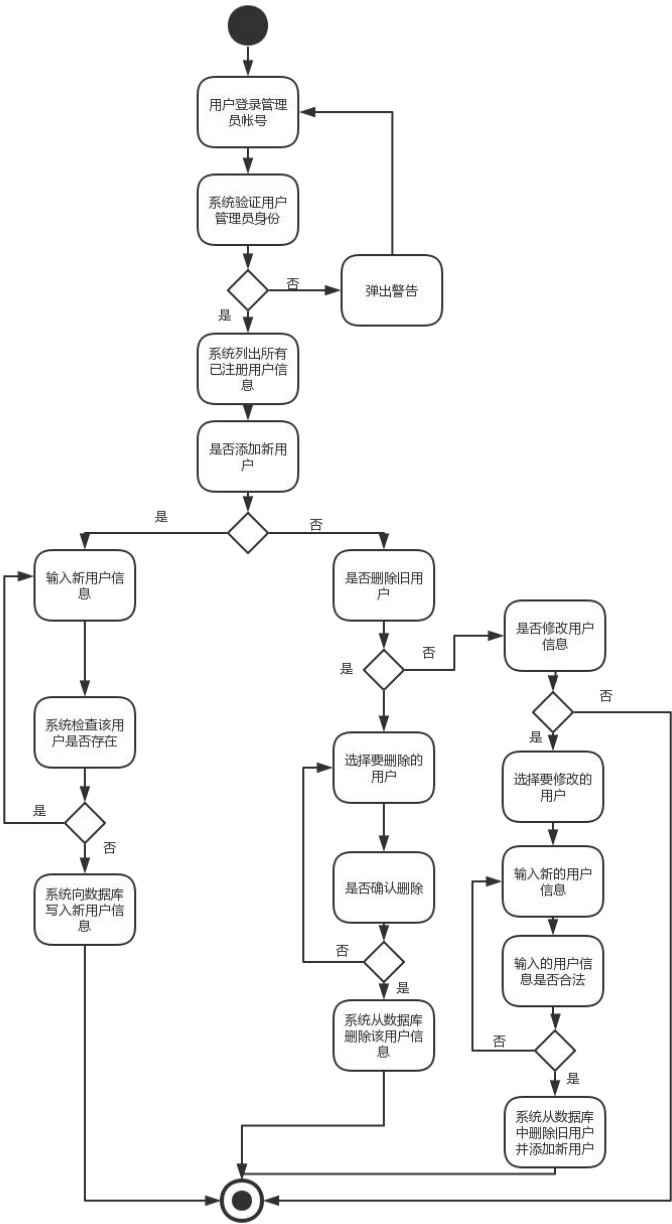


图 2-14 管理顾客信息活动图

2.4.1 视图层

视图层（View）提供用户与网站交互的可视化界面，通过获取用户的输入并将控制层的处理结果进行显示。可视化界面简洁形象，为用户操作提供直接指引。

本网站的视图层由以下几个页面视图组成：

- （1）网站主页面
- （2）注册页面
- （3）登录页面

- (4) 管理员页面
- (5) 预订座位页面
- (6) 预订菜品页面
- (7) 查看订单页面

2.4.2 控制层

本网站的控制层主要包括四个部分：

(1) 第一部分为用户注册的逻辑控制。用户注册时需要提交手机号码、用户密码。控制器从视图层的表单中获得这些信息以后，作相应的验证处理后将这些数据存储到数据库中。并且控制相应的页面跳转。

(2) 第二部分为用户登录的逻辑控制。用户登录时需要输入手机号、密码、验证码。控制器从视图层提交的表单中获取到这些信息以后，作相应的验证处理后更改数据库的状态。并且控制相应的页面跳转。

(3) 第三部分为座位预定的逻辑控制。用户在相应页面选择了心仪的座位后选择提交预订订单。控制器从视图层提交的表单中获取提交订单的用户登录名、桌型、预定时间等信息，判断该座位满足预订后生成相应的订单号并且把数据存储到数据库中。把对应的座位对应的时间段修改为不可预订以后，控制相应的页面跳转。

(4) 第四部分为用户订单取消的逻辑控制。用户在查看了自己的所有订单后以及管理员查看了某个订单后，可以选择取消对应的订单，但用户只能取消自己的未消费订单，而管理员可强制取消任意订单。控制层从视图层提交的表单中获取取消订单的用户登录名、桌型、预定时间后，把数据存储到数据库中。把对应的座位对应的时间段修改为可预订以后，控制相应的页面跳转。

2.4.3 数据层

本网站的数据模型分为三部分：

(1) 第一部分为网站的用户数据。其中包括了管理员、顾客的数据，有各自不同的数据类型。管理员数据包括了管理员登录用户名、密码，以对象数组的形式进行存储；顾客数据包括顾客登录手机号、密码，也是以对象数组的形式进行存储。

(2) 第二部分为网站的座位数据。每一个条目包括座位编号、餐桌类型、餐桌剩余数量、是否被预定信息，以对象数组的形式进行存储。

(3) 第三部分为网站的订单数据。每一个条目包括订单号、订单关联顾客用户名、订单关联座位编号及数量、订单生成时间、订单是否已完成信息，以对象数组的形式进行存储。

视图层、数据层、控制层各个元素之间的控制与被控制的关系如下图 2-16 所示：

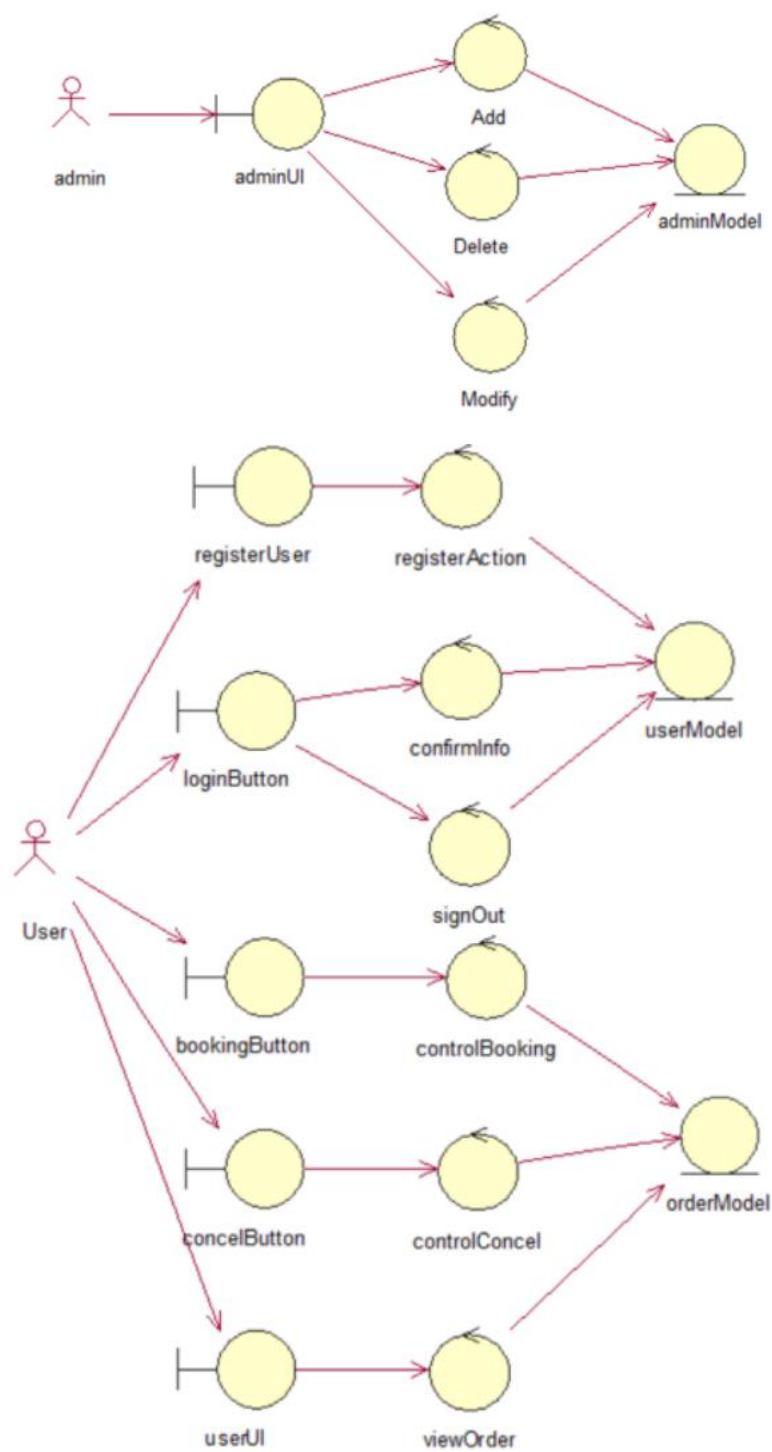


图 2-16 系统之间关系图

2.5 功能需求与程序的关系

通过对用户需求分析,可以分析出该网上购物系统大致分为:注册、顾客登录、管理员登录、预订餐桌、查看订单、取消订单六个模块;后台为:管理员模块。

本系统元素的划分如下表 2-1 所示:

表 2-1 系统元素划分表

类名	标识符	功能
注册	registerUser	注册账号信息
登录	loginButton	登录系统
预订餐桌	bookingButton	预订餐桌和餐品
查看订单	userUI	查看预订订单
取消订单	concelButton	取消顾客订单
管理用户信息	adminUI	添加、修改、删除用户和订单信息

2.6 人工处理过程

用户注册、登录时输入自己的账号和密码;

管理员登录时输入自己的账号和密码;

顾客在预订餐桌和餐品时按照页面的提示点击,选择餐桌和餐品;

顾客在取消预订时,点击的“取消预订”按钮;

顾客正在查看订单时,点击“查看订单”按钮;

管理员管理用户信息时,按照页面的提示点击选择餐桌、餐品,以及输入用户信息。

2.7 尚未解决的问题

并没有该类问题。

3. 设计技术

3.1 Object-oriented Programming

(1)技术说明

整个项目中都使用了面向对象的编程,最基本的就是对 DOM 对象进行操作。我们使用

到了继承,封装等基本的思想进行程序设计. 我们使用的是 vue 框架, 而 vue 有着数据驱动的视图概念. Vue 实例在创建时有一系列初始化步骤如编译模板, 创建必要的数据绑定等. 我们在普通 html 模板中使用特殊的语法将 DOM 绑定到底层数据. 一旦创建了绑定, DOM 将与数据保持同步, 每当修改了数据, DOM 便相应地更新. 这让我们的代码更容易撰写, 理解和维护.

(2) 源代码中的体现

整个项目

3.2 Aspect-oriented Programming

本项目比较简单, 没有使用到面向切面编程. 对于两个类之间的耦合, 我们只涉及到将头部组件在其他组件中进行调用, 由于比较简单, 所以耦合度比较低不会影响项目的质量.

3.3 Service Oriented Architecture

(1) 技术说明

在后端编程中, 按照模块对代码进行划分, 并针对抽出公用代码, 降低了项目整体的耦合度, 服务之间的关系最小化, 提高了内聚性. 在前端编程中, 采用组件式开发方式, 大大地提高了代码的复用程度.

(2) 源代码中的体现

整个项目

3.4 Design Patterns

(1) 技术说明

主要使用了简单工厂模式, customer 和 admin 都继承 user, 并且 customer 和 admin 的功能有所不同.

(2) 源代码中的体现

Admin.vue

4. 接口设计

4.1 用户接口

用户界面接口设计: 本系统为 webAPP, 只提供可视化操作方式, 不提供命令控制语句

进行输入控制，从而用户只需要使用鼠标进行命令操作，使用键盘输入系统接受的参数。总的来说，系统用户界面应做到可靠性、简单性、易学习和使用。

4.2 外部接口

软件接口：使用 nodejs 使用的 MongoDB 接口来实现对数据库的访问；

硬件接口：键盘、鼠标输入，显示器输出。

4.3 内部接口

内部接口方面，采用共有类的一些方法对外接口，供各模块使用。各模块之间采用函数调、参数传递、返回值的方式进行信息传递。结构传递的信息将是以数据结构封装的数据，以参数传递或返回值的形式在各模块间传输。

5. 模块划分

我们的餐厅预定网站主要分为两个功能独立的子系统，分别是用户模块和商家管理员模块。

5.1 用户模块设计

用户模块主要针对用户登陆，下订单，查看订单，取消订单操作。

视图层：registerUser, loginButton, bookingButton, cancelButton, userUI

控制层：registerAction, confirmInfo, signOut, controlBooking, controlCancel, viewOrder

数据层：userModel, orderModel

用户模块划分如下图 5-1 所示。

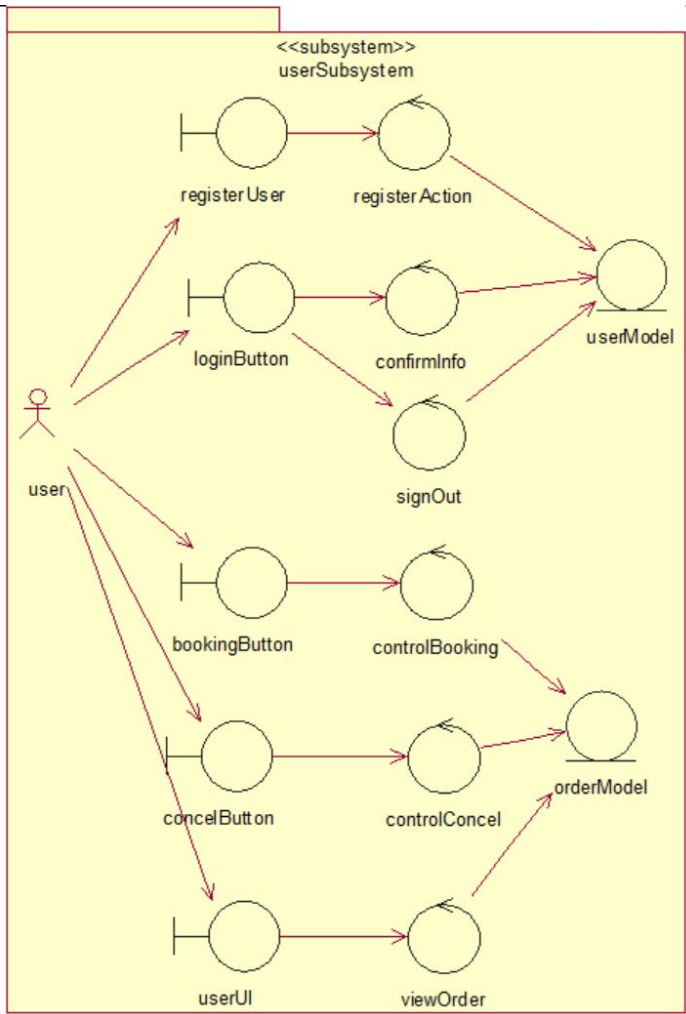


图 5-1 用户子系统划分

5.2 商家管理员模块设计

商家管理员模块主要是针对订单和用户信息的添加、删除和修改。

视图层：adminUI

控制层：Add, Delete, Modify

数据层：adminModel

管理员模块划分如下图 5-2 所示：

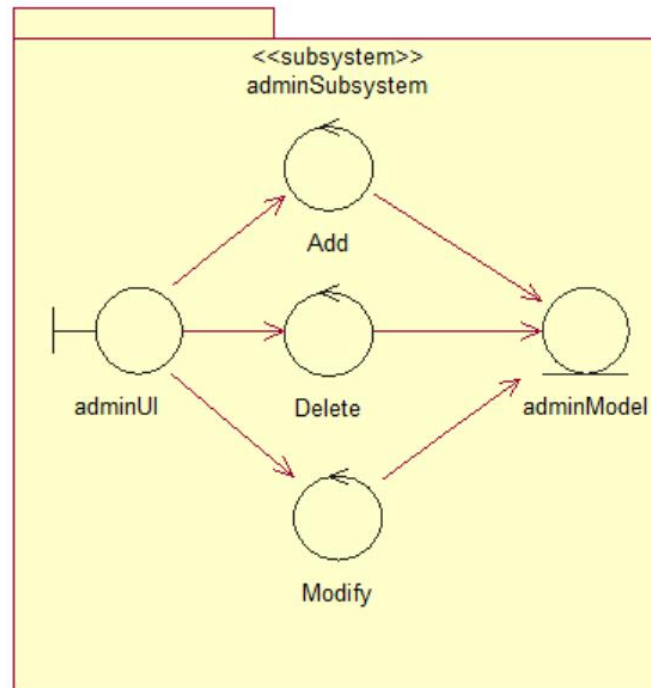


图 5-2 管理员子系统划分

6. 运行设计

6.1 运行模块组合

在有用户访问时,需要先注册并登陆后才可以点击相应的功能模块.用户点击相应功能使各个模块被调用.模块之间若存在依赖关系则可相互调用.

6.2 运行控制

运行控制严格按照各模块间函数调用关系来实现.在各事务中心模块中,需要对运行控制进行正确的判断,选择正确的运行控制路径.

客户机在发送数据后,需要等待服务器的响应,收到响应后,再次等待服务器发送回答数据,然后对数据进行确认.服务器在接收到数据后发送确认信号,在对数据处理,访问数据库后,将返回信息送回客户机,并等待确认.

6.3 运行时间

由于我们的网站比较简单,所以数据库的运行以及 http 的响应耗时较短,因而没有做出确切的要求.

7. 系统出错处理设计

7.1 出错信息

本网站可能出现的故障以及处理方法如下表所示：

表 7-1 出错及处理方法表

错误名称	输出结果	处理方法
不合法的 http 请求	404 异常或 400 异常	重新获取并提交
用户输入数据不合法	给出错误提醒	要求重新输入并判断
程序异常	网页内容异常或无法显示	重新获取并提交

7.2 补救措施

由于我们的项目内容相对比较简单, 而且我们的网站中订单的数据存储通常只需要保持两天即达到要求, 对数据的存储也不需要永久性的保存, 所以当故障出现后重启或者重新部署即可.

7.3 系统维护设计

本系统保存操作日志以及相关操作信息, 以供维护人员在故障出现时进行调试和维护.