

**论文（设计）题目:**

**基于HTML5在线酒店预订和点菜平台**

姓 名 **周臣文**

学 号 **201200131157\_\_**

学 院 **山东大学计算机科学与技术学院**

专 业 **计算机科学与技术**

年 级 **2012级**

指导教师 **何伟**

2016年 5月 1日

**目录**

[摘 要 3](#_Toc14191)

[ABSTRACT 4](#_Toc31374)

[第1章 绪论 5](#_Toc16088)

[1.1在线酒店预订和点菜平台开发背景 5](#_Toc14199)

[1.2 国内外研究现状 5](#_Toc7162)

[1.3 解决的主要问题 6](#_Toc24830)

[1.4 本文的主要工作 7](#_Toc6662)

[1.5 论文的总体结构 7](#_Toc19938)

[第2章 在线酒店预订系统和点菜平台需求分析 8](#_Toc28643)

[2.1在线酒店预订和点菜平台概述 8](#_Toc466)

[2.1.1在线酒店预订和点菜平台项目背景 8](#_Toc4639)

[2.1.2 在线酒店预订和点菜平台项目说明 8](#_Toc17696)

[2.1.3 在线酒店预订和点菜系统概述 9](#_Toc7636)

[2.2 在线酒店预订和点菜平台主要解决的问题 9](#_Toc9797)

[2.3 在线酒店预订和点菜平台需求分析 10](#_Toc24120)

[2.3.1 在线酒店预订和点菜平台的业务流程 10](#_Toc26554)

[2.3.2在线酒店预订和点菜系统需求分析 11](#_Toc1943)

基于HTML5的在线酒店预订和点菜平台

# 摘 要

随着移动互联网浪潮的来临，团购已经成为了人们日常生活中不可或缺的部分，一大批的团购网站，App如雨后春笋般出现在公众视野中。本文在分析了现有的团购网站之后，结合高端酒店预订的流程，详细分析了在线酒店预订和点菜平台的项目需求，设计和实现了酒店房间预订和点菜的系统。

首先，本文在考虑了在线酒店预订和点菜系统的项目背景和开发过程中可能面临的问题的基础之上，进行了详细的功能性需求分析和非功能性需求分析，并且对需求进行了详细的说明，包括文字说明和流程图说明。

在进行了详细的需求分析之后，开始进行在线酒店预订和点菜平台的架构设计，进一步开始了在线酒店预订和点菜平台的详细设计。针对目前原生APP面临的开发成本高，占用系统内存过多等问题，使用HTML5实现的WebApp具有跨平台，轻应用的特点。最后，本文分析了基于HTML5进行移动开发的优势，以及移动开发的趋势。

**关键字**：HTML5；移动开发；酒店预订；协同过滤算法

# ABSTRACT

# 第1章 绪论

## 1.1在线酒店预订和点菜平台开发背景

最近几年来，随着移动互联网的兴起，团购类的网站和APP如雨后春笋般出现在大众视野中。美团、糯米、大众点评等等应用在人们生活中开始扮演越来越重要的角色，人们在外出逛商场，买东西，定酒店，或者吃饭的时候，都会习惯性的打开手机，通过团购网站来获取优惠或者推荐。而随着团购应用的增多，各个应用之间的竞争也是愈演愈烈，从线下大面积的推广，到线上无休止的“烧钱”，还有不断更新的APP，都只是想让自己的APP能够在国内市场占有一定的份额。然而，对于企业而言，目前市场上主流的两类操作系统：Android和IOS，都具有大量的用户，所以，不能放弃任何一方的用户。这就导致企业不得不为两种操作系统开发两套功能几乎一样的原生APP，从而使得开发成本大大提高；与此同时，对于用户而言，想要获取更多的优惠或者获取更好的体验，那就不得不安装多个APP，在不同的应用之间来回切换，从而得到更大的实惠。然而，手机的内存是有限的，如果安装过多的软件，势必会让手机变得很卡顿，极大地降低了用户体验；同时，用户想要获取最新的功能，还不得不忍受来自APP随时的更新提示......原生应用的缺陷正在随着移动互联网的不断发展而逐渐显现出来，而对应的解决方案便是由HTML5的出现而提出的。

## 1.2 国内外研究现状

HTML5是万维网的核心语言，标准通用标记语言HTML的第五次重大修改，在2014年10月29日，HTML5的标准规范定制才被正式宣布完成。HTML5作为当下最火热的Web技术，已经在互联网行业中取得了大量的应用。一些基于HTML5创建了web应用，向用户提供了一种完全不一样体验：观看视频再也不需要安装flash这样的插件，动画，音频，视频以及所有同PC端的交互都将被标准化，同时，HTML5的出现，改变了以往多版本开发，多终端适配的开发模式，让“一次开发，处处运行”成为可能。随着HTML5技术的不断成熟和应用，世界上几个主要的浏览器厂商和操作系统厂商也逐渐表态，宣布他们将逐步的在自己的浏览器产品中支持HTML5的新特性。例如，Google和苹果两家科技巨头已经宣布支持HTML5的发展，同时将HTML5的应用领域重点放置在移动领域；同时，云计算平台的出现也让大量web应用同时运行成为可能，通过基于云计算的虚拟化，分布式，和集群技术，能够在云端支持大量的web应用同时运行。对于开发者来说，许多针对于HTML5的应用开发工具也应运而生，例如jQuery Mobile是在jQuery的基础之上，对移动端的触屏操作进行了优化， Bootstrap让不同终端的适配更加简便...这些工具的出现，会让HTML5的开发门槛越来越低，大大提高应用开发的效率。

## 1.3 解决的主要问题

在线酒店预订和点菜平台是为了解决顾客在高端酒店用餐时需要等待的问题。一般情况下用户在酒店用餐的流程：用户进入酒店，选择用餐房间，之后开始点餐，等到菜品上齐之后才能开始用餐。但是这其间会发生很多问题导致用户体验下降：首先，用户到了酒店再去选择用餐包间，很有可能在用餐高峰期餐厅已经满了；其二，用户到了酒店才开始点餐，那么需要等待的时间会很长；其三，对于新用户而言，对于该酒店的菜品和特色都不太了解，通过点餐平台可以了解到其他用户在此的消费体验。其次，对于开发者来说，基于HTML5的在线酒店预订和点餐平台还能够解决跨平台运行的问题，基于HTML5开发的应用可以实现跨终端运行，能够很好地兼容各种不同大小的终端设备。传统的网站一般只能够在PC端有比较好的适配，但是随着移动互联网的到来，用户通过移动端连接网络的次数越来越多，而且，移动端的屏幕大小参差不齐，传统的网页是无法在移动端很好展示的。而对于HTML5应用来说，对移动端极其友好的支持是其最大的优势。目前，市场上主流的操作系统有两类，Android和IOS，两类操作系统所占有的用户市场都非常大，对于企业来说，任何一方的用户都不能够被舍弃，所以，一款产品一般都要开发两种适配，开发成本提高，维护成本也会提高。而对于HTML5来说，由于基于HTML5的产品是可以跨平台运行的，因此，企业想推出一款产品，只需要开发一次即可，可以实现“一次开发，处处运行”的美好愿景。

## 1.4 本文的主要工作

本文主要工作是实现一个在线酒店预订系统，根据酒店预订的基本需求，进行网站的详细设计，之后经过代码实现和系统测试，最终完成整个系统。

首先，本文分析了在线酒店预订和点菜平台的基本需求。系统需要记录用户的基本信息，所以需要实现登录和注册的功能；之后，用户可以浏览站内所有酒店的基本信息，包括酒店名称，酒店介绍，酒店图片，酒店位置，以及评价星级等等，这些需要在首页进行展示；当用户点击进入其中一家酒店进行详情了解时，需要展示酒店更加详细的信息，包括多张酒店图片的浏览，优惠信息展示，其他用户的评价，用餐包间的介绍，特色菜品展示等等；用户可以在此页面进行点菜和预订房间，预订成功之后会有对应的提示信息；之后用户进入酒店可以凭借预订成功的信息直接进入包间消费；消费完成以后，可以对酒店的房间，菜品，和整体印象（卫生，服务等）进行评价。

接着，论文介绍了在线酒店预订和点菜平台的实现过程。网站建设使用HTML+CSS+JavaScript作为前台开发语言，后端使用PHP开发，使用Mysql作为数据持久化的工具。

最后，对整个系统进行测试。通过模拟用户在线预订酒店的过程，在系统上进行一系列的操作，最后查看用户是否能够完成预订的功能。

## 1.5 论文的总体结构

# 在线酒店预订系统和点菜平台需求分析

## 2.1在线酒店预订和点菜平台概述

### 2.1.1在线酒店预订和点菜平台项目背景

在线酒店预订和点菜系统是面向中高端大型酒店所提供的一套预订平台。该平台旨在解决用户进入酒店用餐消费需要等待的问题，提前订好房间和菜品，到达酒店即可用餐。目前大多数的中高端型酒店都有自己的预订电话或者联系方式，但是并没有一个相对完善的平台来整合全部酒店的资源，在酒店和用户之间没有有效的信息传递平台，导致用户无法获取酒店的联系方式，酒店也无法向用户去推荐自己的特色消费。在线酒店预订和点菜平台就是在用户和酒店之间搭建了一座信息流通的桥梁，对于用户来说，通过平台预订，节省了用餐时间，获取了最新优惠，了解了更多的酒店特色；对于酒店来说，可以凭借良好的服务，特色的菜品在平台积累好评和人气，能够在平台上得到更多的用户。

### 2.1.2 在线酒店预订和点菜平台项目说明

在线酒店预订和点菜系统是为了便于让信息在用户和酒店之间流转的更加便捷。对于酒店来说，极其需要在用户群中增加自身良好的口碑和企业形象，通过在平台上积累人气，和会员之间线下的口口相传，可以对酒店的形象起到很大的提升作用；对于消费者而言，非常渴望能够找到一家服务，菜品，环境都很好地酒店用餐，因为他们往往是基于一种商务目的或者是交友目的来请别人用餐，此时，酒店的口碑是消费者最为看重的。通过在线酒店预订和点菜平台，消费者可以很方便的获取其他用户在该酒店的消费感受，可以通过评价星级对这家酒店产生一个基本的判断，因此，对于消费者来说，选错酒店的可能性就会减少很多；不仅如此，通过在平台上提前点菜和预订房间，可以为客户节约大量的用餐等待时间。

### 2.1.3 在线酒店预订和点菜系统概述

在线酒店预订和点菜平台主要是针对中高端型酒店而设计的，旨在解决用户用餐等待时间过长的问题。整个系统采用的是B/S架构，对于用户来说，去除下载安装APP的麻烦。系统前端使用HTML5+CSS3+JavaScript技术，创建可适配移动端屏幕大小的页面，后端使用PHP语言设计实现网站的动态化；采用了PHP基础框架CodeIgniter，CodeIgniter是一款基于PHP语言实现的web框架，采用MVC的设计思想，将模型，控制器，视图三者分开，模型用来处理与数据库的交互，通过模型将网站内的数据持久化到数据库中；视图用来完成页面的渲染，控制器用来控制整个系统的逻辑，包括路由处理，表单处理等。控制器是连接模型和视图的中间者，从模型获取到了数据，通过控制器传递到视图中，然后在通过视图部分的渲染展示在页面上。后端的数据库采用Mysql，存储引擎使用的是InnoDB。服务器使用Apache搭建。

## 2.2 在线酒店预订和点菜平台主要解决的问题

目前，在中高端类型的酒店和用户之间缺少一种信息流动的平台，用户到这些酒店进行消费的时候经常会需要等待，这对于用户来说，增加等待时间，降低用户体验；对于酒店来说，缺少一个展示酒店形象和特色的平台，无法在用户群中快速建立自己的良好形象和优秀的口碑。基于以上两点困境，在线酒店预订和点菜平台主要为用户和酒店解决以下几点问题：

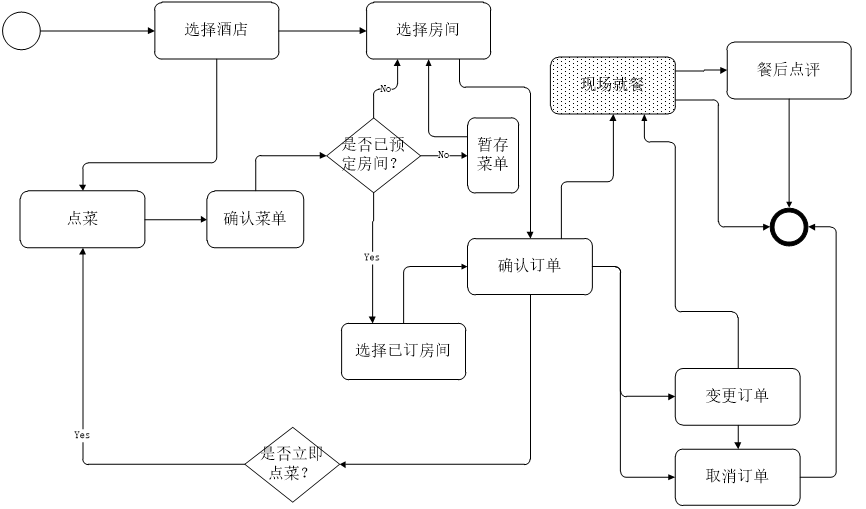
1. 用户无法获符合自己口味的酒店。在线酒店预订和点菜平台可以给用户提供大量的酒店信息，在平台内，用户可以通过不同的搜索条件，搜索符合自己口味的酒店，浏览酒店图片，酒店介绍，最重要的是可以获取其他用户在该酒店的消费评价，从而形成自己对该酒店的第一印象。
2. 用户在平台预订之后，直接去酒店消费，无需等待。目前，在酒店用餐的一般流程都是先去选择用餐包间，然后在点菜。这期间会有很多问题导致用户体验下降。例如：用户到了酒店，却被告知酒店包间已经满了，无法提供包间；又或者，点完菜之后，由于来用餐的人太多，导致菜品迟迟不能够上。这一系列的等待都会让用户对酒店的映像大打折扣。
3. 在线酒店预订和点菜平台是一款基于HTML5实现的WEB应用，因此，对于用户来说，他们不需要下载安装新的APP，只需要打开我们的网站，在浏览器中即可完成预订的功能。
4. 酒店可以在平台内展示自己的信息，吸引更多的用户前来消费。
5. 基于HTML5实现的WebApp，可以适配不同的移动设备，不需要像开发传统应用一样，为每一款系统开发对应的产品。这样可以大大节约开发成本和维护成本。

## 2.3 在线酒店预订和点菜平台需求分析

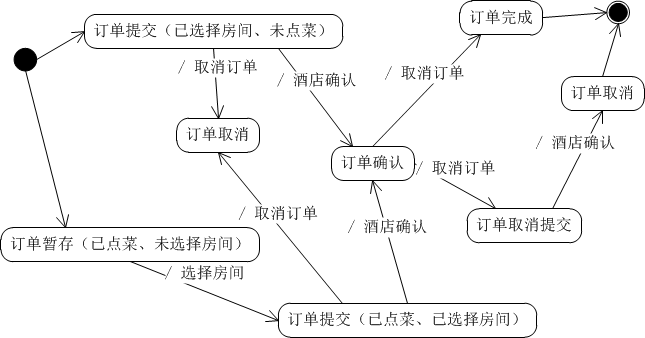
### 2.3.1 在线酒店预订和点菜平台的业务流程

用户进入在线酒店预订和点菜平台后，首先出现的是登录页面，在登陆页面，用户填写手机号和登陆密码进入系统，如果用户暂时没有网站账户，可以点击页面中的注册按钮进入注册页面，在注册页面中，用户只需要填入手机号和密码就可以完成注册。进入系统后，在首页，向用户展示系统推荐的酒店，排名的顺序根据距离用户的距离，口味，价格等维度排序；用户选择某个酒店，可以进入查看酒店详细信息，之后可以完成点菜，和预订用餐房间，形成自己的订单，在预订房间或者点菜的时候，用户可以先预订房间，然后直接点菜或者选择到了酒店再去点菜，也可以先点菜，之后到了酒店再去选择用餐房间；预订成功之后，会形成用户自己的订单，在个人信息中可以查看，用户可在订单中心取消订单或者继续预定房间和点菜；消费完成之后，用户可以在系统内完成相关评价。

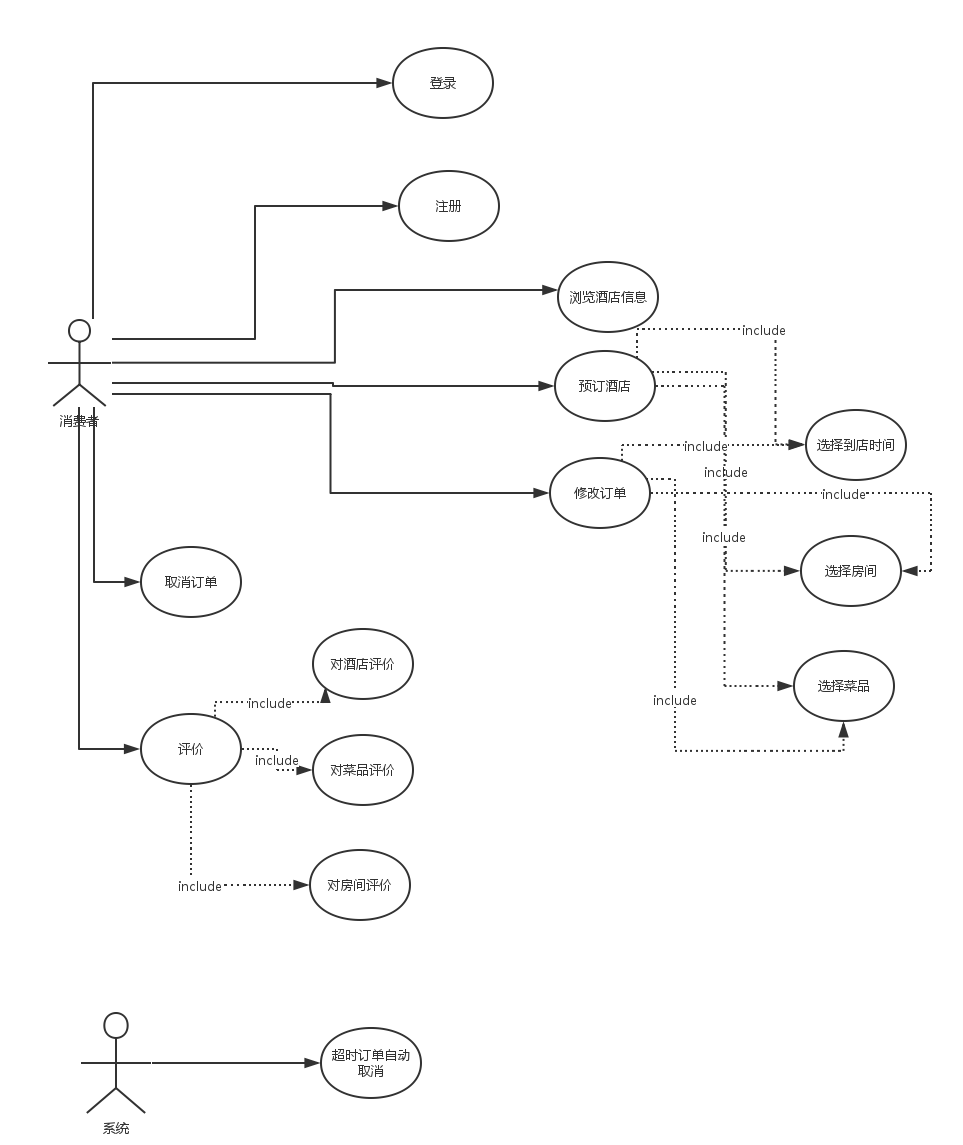
以下是平台的功能流程图：



订单状态的转换图：



在线酒店预订和点菜平台用例图：

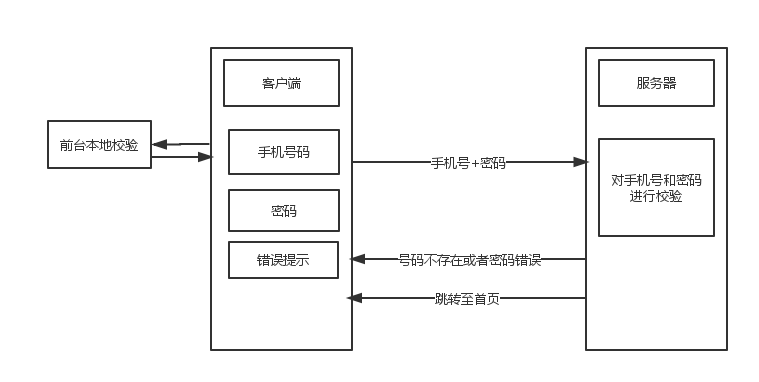


### 2.3.2在线酒店预订和点菜系统功能性需求分析

1. 登录和注册

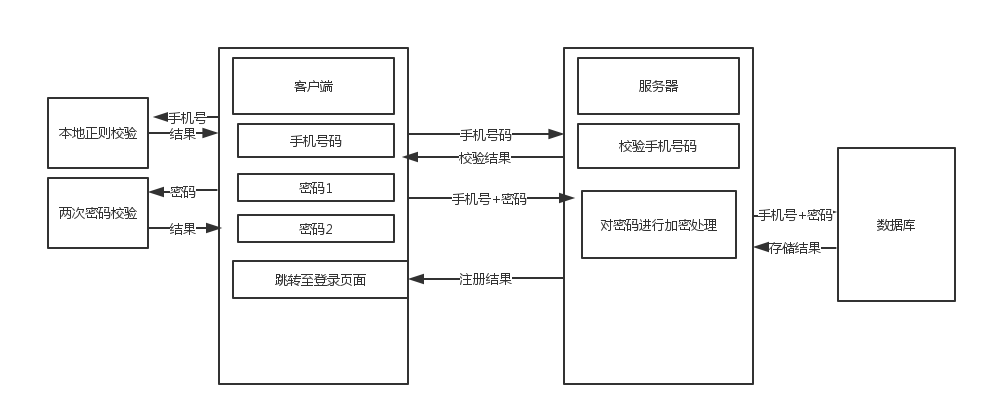
登录：用户通过输入手机号码，登录密码进行登录。当用户输入完手机号码，光标离开输入框的时候，前台需要通过JS进行基本的校验工作，校验用户的输入是否合法。用户只能够输入11位数字，不是数字或者位数不正确都要被提示输入不合法的信息，并且无法提交表单。用户正确输入用户名，密码之后，使用Ajax技术异步访问服务器，Ajax技术的最大特点是能够不用刷新页面就能够与服务器完成数据交互，或者说完成了页面也服务器的局部数据的交互，这样做不仅减小了带宽压力，同时提高了用户体验。获取服务器返回的提示信息，进行判断，如果用户名不存在，说明该手机号还没有被注册，提示用户先去注册；如果用户密码错误，则清空密码框，提示用户重新输入密码；当用户名输入正确，前台不需要任何的提示消息，直接进入首页即可。

登录过程交互情况如下：



注册：输入手机号码之后，当光标离开输入框的时候，需要进行两步操作：第一步，前台通过JS校验用户输入字符的合法性，这里主要通过正则表达式来处理，用户输入的字符必须是11位的数字，位数多了或者少了，或者含有不是数字的字符都将被判定不合法，然后直接在页面上显示出来不合法的提示信息；第二步，如果输入的字符合法，使用Ajax技术异步访问服务器，查看当前号码是否已经被注册，如果该号码已经被注册，则在页面上显示相应的提示信息；如果该号码没有被注册，则向用户显示出该号码可以使用的提示信息；之后用户输入密码，密码需要输入两次，两次密码必须保证一致，这一步也是在客户端进行校验。填写完成后，将数据提交至服务器，服务器对密码进行校验后存入数据库，向客户端发送注册成功的信息，然后客户端直接切换回登录页面，要求用户重新登录。

注册过程交互情况如下：



2.首页浏览功能

在首页中，用户可以浏览平台内所有酒店的信息。

①酒店排列顺序按照三种维度来定义：距离，价格，口碑。用户可以选择其中任意一种来进行排序，例如，如果选择按照距离来排序的话，则系统将会把距离用户最近的酒店推荐在前面，按照距离的远近进行排序展示；如果用户选择价格，则按照酒店的平均消费水平对所有酒店按照从高到低（或者从从低到高）的顺序进行排序；如果按照口碑进行排序，则按照酒店的评价星级对酒店从高到低进行排序展示；

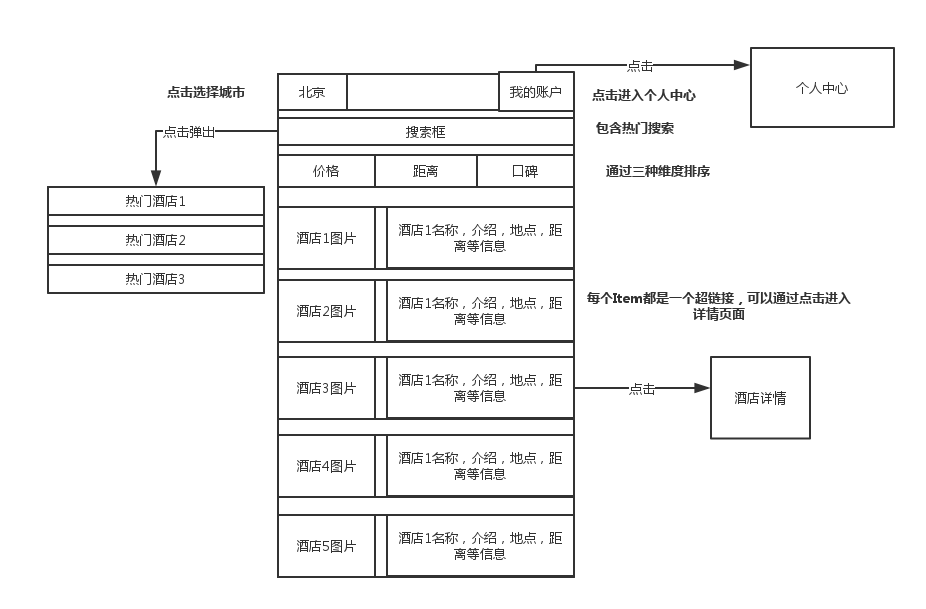
②首页需要提供搜索功能，用户输入酒店名字可以模糊检索出相关联的酒店进行展示，同时，当用户点击搜索框的时候，也会在搜索框下面展示出当前的热门搜索酒店，为用户提供最热门的推荐；

③酒店的展示信息按照列表的形式进行展现。每一个酒店的信息作为一个Item放在列表中。Item里面应该包含酒店的一张图片，酒店名称，酒店简单介绍，酒店地理位置和酒店距离用户的距离。这个Item应该是可以点击的，用户点击该Item会将页面跳转到该酒店的详情页面，在详情页面会有更加详细的描述。

④首页顶部的左边显示当前用户所在的城市，可以点击进行城市切换。切换之后，页面中的酒店列表需要根据新的城市名称重新渲染。首页顶部的右边是我的账户信息。点击我的账户，会跳转到个人中心页面。

⑤网站整体需要适配移动端的所有类型的屏幕，因此，页面内各种元素的长度都是以百分比的形式给出的，当用户通过不同终端进行浏览的时候，都会有一个比较好的页面效果。网站的页面是会随着用户屏幕的大小而进行适当的缩放，但是并不会改变整个网页的布局，目前设置的最小宽度是320px，也就是说，当访问终端的屏幕宽度小于320px的时候，网站是不会继续根据屏幕进行缩小适配的，320px也是经过查找资料之后得到的目前市场上最小的手机屏幕的宽度。

以下是首页的原型设计图：

3.详情页面功能需求

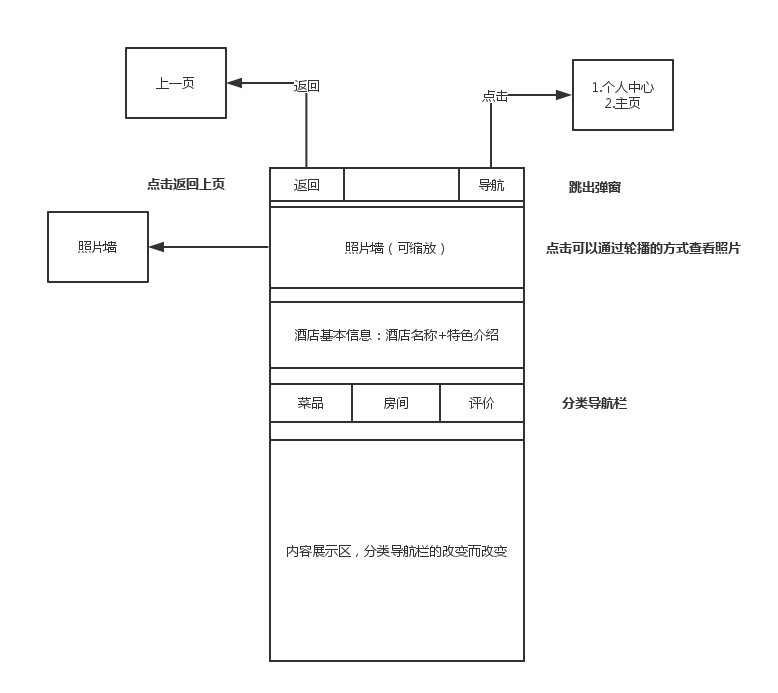
①页面顶部左侧应该是一个返回按钮，可以返回到上一个页面，右侧是一个导航图标，点击导航图标会有一个弹窗，在弹窗内可以选择返回主页或者进入个人中心。页面顶部的中间显示“商家详情”；

②详情页面中需要在显示酒店照片，在导航栏下方就是一个全宽度的照片展示区域，该照片展示区域只展现出一张酒店照片，该区域会随着访问终端屏幕大小的不同发生缩放。点击照片展示区域，会跳转到一个全屏展示照片的页面。

③照片展示区域的下方是酒店的基本信息展示栏，在基本信息展示栏中，会有酒店的名称，酒店的特色介绍等信息。

④在酒店基本信息展示栏下方是一个分类导航栏。分为三种类型：菜品，房间，评价。点击菜品，下方的div展示的是该酒店的菜品信息，同样是以列表的形式展现出来，每一个菜品是列表中的一个Item；菜品的展示方式跟首页中酒店的展示形式类似；每个Item中左侧是菜品的照片，右侧是菜品的基本信息，包括菜名，菜品简介，售价，销售量，评价数，做右下角是一个点菜的按钮，点击之后，对应的菜品会被加入订单中；分类导航栏中点击房间，下方的菜品信息被隐藏，随着代替的是酒店的房间信息。房间信息的展示是以列表形式展现出来的。每个房间的信息是其中的一个Item，Item的左侧是房间的照片，右侧是房间的名称，简单介绍，还有大小和预订按钮。房间如果已经被预定，预订按钮将会被置灰，无法点击；分类导航栏中点击评价按钮，下方的房间信息会被隐藏，随着显示的是评价列表。评价列表中显示的信息是每个在该酒店消费的用户对该酒店的评价信息，用户可以根据他人的提供的用餐体验来决定是否在此酒店进行用餐。

以下是酒店详情页面的原型设计图：



1. 订单详情页面

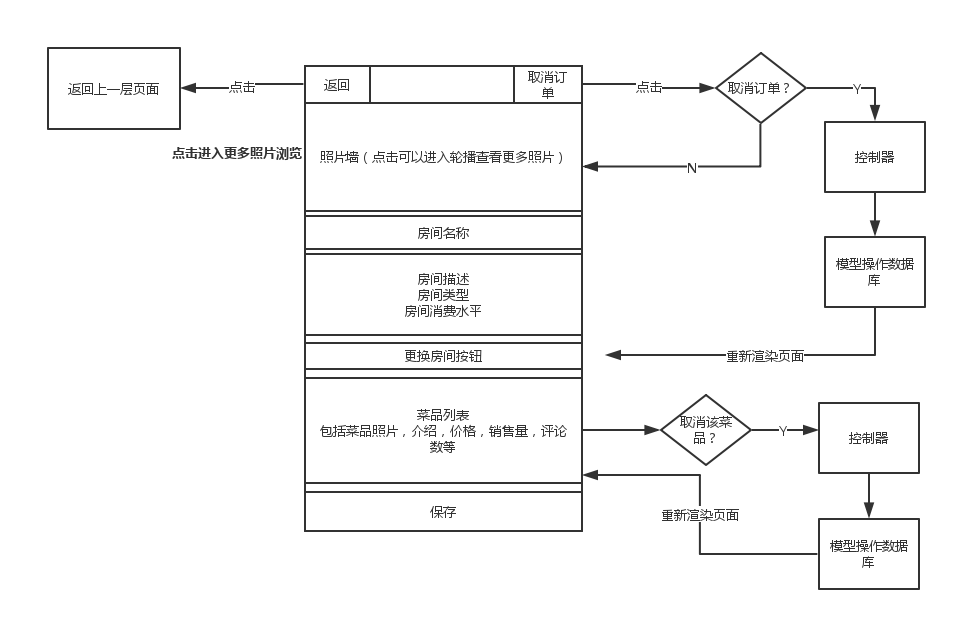
①页面顶部左侧是返回按钮，点击返回上一层页面；页面顶部右侧是取消该订单按钮，点击后可以取消当前订单；

②顶部栏下方是酒店的照片，用于展示酒店形象。点击图片可以进入大图，查看酒店更多照片。

③照片墙下方是当前用户已经预定的该酒店的房间，如果用户的这个订单还没有预定过房间，则不需要显示房间的具体信息，只需要显示是否立即预定房间即可，如果用户已经预定了房间，则可以显示出当前房间的基本信息，包括房间大小，房间类型（豪华型，普通型，经济型等），房间最低消费水平等信息，下方会有一个换一间房间的按钮，用户点击按钮后，就会跳转至酒店详情页面的房间信息列表中，可以更换房间；

④在已定房间信息下方是当前已点菜品的信息，以列表的形式展现出来，每个Item显示的是一个菜品的基本信息，包括菜品的照片，菜品的照片，菜品名称，菜品介绍，菜品的销售量，评价数，以及菜品的价格等。在每个Item的右下角还会有一个退订的图标，用户点击退订，则可以在自己的订单中取消这道菜。

以下是订单详情页面的原型图：

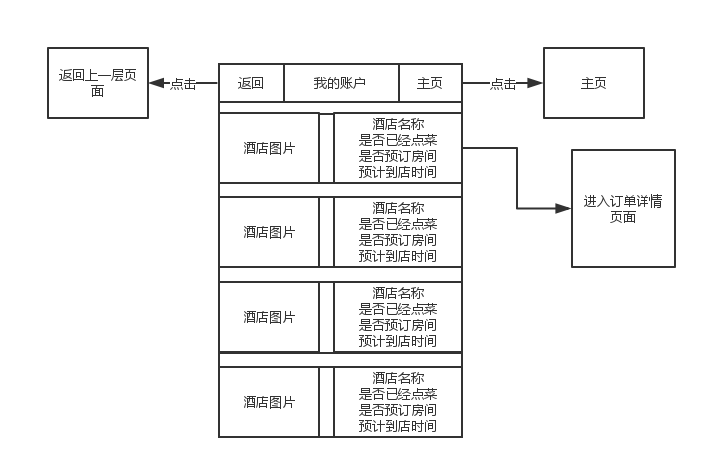


1. 个人中心页面功能需求：

个人中心页面主要展示的歌用户的个人基本信息，包括头像，当前登录名等。可以提供的功能包括修改密码，查看当前订单，修改订单等。

①查看当前订单：订单以列表的形式给出，用户在页面上可以查看所有的订单，包括已消费和未消费的订单，未消费的订单将会排在前面展示，已经消费的订单按照消费时间排序。

以下是订单列表原型：



### 2.3.3在线酒店预订和点菜平台非功能性需求分析

在线酒店预订和点菜平台的非功能性需求主要有以下几点：

①界面友好：网站的页面设计需要能够抓住大众的眼球，需要能够突出站点内酒店的特色和优点，需要留住用户，吸引用户下订单；

②操作友好：用户在浏览酒店信息，下订单，或者修改订单的时候，需要直接可操作，对于用户来说，使用门槛要尽量低，把重点操作放在显眼的位置；

③交互友好：部分交互采用Ajax技术实现，例如在登陆注册时候对于用户名的检测，通过局部刷新页面，提升用户的体验；对于站点内大量的图片加载，为了减少用户等待图片加载的时间，对于已经加载到本地的图片我们采用HTML5缓存的方式将图片缓存在本地，减少页面刷新时间；

④性能：为了减轻大量图片加载和存储对服务器和带宽造成的压力，将图片储存在七牛，使用七牛提供的CDN技术还可以加快图片的下载速度；

# 在线酒店预订和点菜系统概要设计

# 第4章 在线酒店预订和点菜系统详细设计

经过需求分析和架构设计，本章主要提出的是系统的详细设计。

## 4.1 轻量级框架CodeIgniter

CodeIgniter 是一个简单快速的PHP MVC框架。EllisLab 的工作人员发布了 CodeIgniter。许多企业尝试体验过所有 PHP MVC 框架之后，CodeIgniter 都成为赢家，主要是由于它为组织提供了足够的自由支持，允许开发人员更迅速地工作。

自由意味着使用 CodeIgniter 时，不必以某种方式命名数据库表，也不必根据表命名模型。这使 CodeIgniter 成为重构遗留 PHP 应用程序的理想选择，在此类遗留应用程序中，可能存在需要移植的所有奇怪的结构。

CodeIgniter 不需要大量代码（1.6.2 版本仅为 2.8 MB，其中的 1.3 MB 是可以删除的用户文档），也不会要求您插入类似于 PEAR 的庞大的库。它在 PHP 4 和 PHP 5 中表现同样良好，允许您创建可移植的应用程序。最后，您不必使用模板引擎来创建视图 — 只需沿用旧式的 HTML 和 PHP 即可。

CodeIgniter 是一套给 PHP 网站开发者使用的应用程序开发框架和工具包。它提供一套丰富的标准库以及简单的接口和逻辑结构，其目的是使开发人员更快速地进行项目开发。使用 CodeIgniter可以减少代码的编写量，并将你的精力投入到项目的创造性开发上。

CodeIgniter是属于视图-控制器-模型这种设计模式的，MVC模式是一种将应用程序的逻辑层和表现层分离的方法。在实践过程中，由于表现层从php脚本中分离了出来，所以他允许你的网站还有少量的脚本文件。

模型(Model)代表你的数据结构。通常来说，你的模型类将包含取出、插入、更新你的数据库资料这些功能。 视图(View)是展示给用户的信息。一个视图通常是一个网页，但是在 CodeIgniter 中，一个视图也可以是一个页面片段，如页头、[页尾](http://baike.baidu.com/view/1328990.htm" \t "http://baike.baidu.com/_blank)。它还可以是一个 RSS 页面，或任何其它类型的“页面”。 控制器(Controller)是模型、视图以及其他任何处理 HTTP 请求所必须的资源之间的中介，并生成网页。 CodeIgniter 在 MVC 使用上非常宽松，因此模型不是必需的。如果你不需要使用这种分离方式，或是发觉维护模型比你想象中的复杂很多，你可以不用理会它们而创建自己的应用程序，并最少化使用控制器和视图。CodeIgniter 也可以和你现有的脚本合并使用，或者允许自行开发此系统的核心库，其目的是，使你可以以最适合你的方式工作。

## 4.2数据库详细设计

### 4.2.1 数据库概要

数据库主要用来将网站内的所有数据进行持久化管理，通过吧数据存储在数据库中，用户随时访问站点，都能够从数据库中挑选出相应的数据进行展示。数据库分为关系型数据库和非关系型数据库，本系统采用的是开发中最普遍的关系型数据库。关系型数据库是建立在关系模型上的数据库，借助于集合代数等数学概念和方法来处理数据库中的数据。现实世界中的实体以及实体之间的关系均用关系模型来表示。关系数据库的操作主要通过结构化查询语言（Structured Query Language）来实现。通过SQL语句，我们可以对数据库实现增删改查等基本操作。常用的关系型数据库软件有Mysql，Oracle，DB2等，本系统我们采用的数据库软件是Mysql。本章主要通过分析数据库的设计目标和原则，系统中实体-关系图（E-R图），和数据表详细描述图来介绍数据库的设计。

### 4.2.2 数据库的设计目标和原则

本次数据库的设计应该本着具有良好的结构化数据，数据独立性高和数据颗粒度小的目标，从而满足在线酒店预订和点菜平台的项目需求。同时在设计数据库时，应该注意一下几个原则：

① 数据库中各个表的命名，数据表中各个字段的命名，都应该遵循基本的命名规范，可以使用驼峰命名法，或者全小写加下划线的命名方法，字段的命名需要简短而且具有明确的命名意义，不可以过于冗长或者让开发者难以理解；

② 数据库设计应该需要考虑数据安全性，对于某一些重要的数据应该做到加密处理。例如，对于用户的密码需要做加密处理，防止数据库被入侵用户的站内安全受到威胁。

③ 数据库的设计需要考虑到未来网站会员大量增加时对于数据库的压力，避免数据库中出现冗余字段，减轻存储压力；

④数据库性能合理，表内结构定义完整，没有遗漏，同时，数据表之间的关联性合理，能够保证各个数据表之间的关联字段对应一致。

### 4.2.2 实体-联系图（E-R图）

E-R图也称实体-联系图(Entity Relationship Diagram)，提供了表示实体类型、属性和联系的方法，用来描述现实世界的概念模型。它是描述现实世界概念结构模型的有效方法。是表示概念模型的一种方式，用矩形表示实体型，矩形框内写明实体名；用椭圆表示实体的属性，并用无向边将其与相应的实体型连接起来；用菱形表示实体型之间的联系，在菱形框内写明联系名，并用无向边分别与有关实体型连接起来，同时在无向边旁标上联系的类型。

通过之前的功能性需求分析和非功能性需求分析，得出在在线酒店预订和点菜平台中需要的基本实体如下：

① 用户信息：（唯一编号，用户密码）；

② 酒店信息（酒店唯一编号，酒店名称，酒店地址，酒店基本描述，酒店被评价数量）

③ 酒店照片信息（酒店编号，酒店照片url）照片存放在第三方托管平台，通过url来获取酒店照片

④ 酒店评价信息（评价唯一编号，被评价酒店编号，评价内容，评价级别，评价人编号）

⑤ 菜品信息（菜品编码，菜品描述，菜品照片，菜品名称，菜品价格，菜品折扣，菜品销售量，菜品评价数，菜品状态，所属酒店编号）

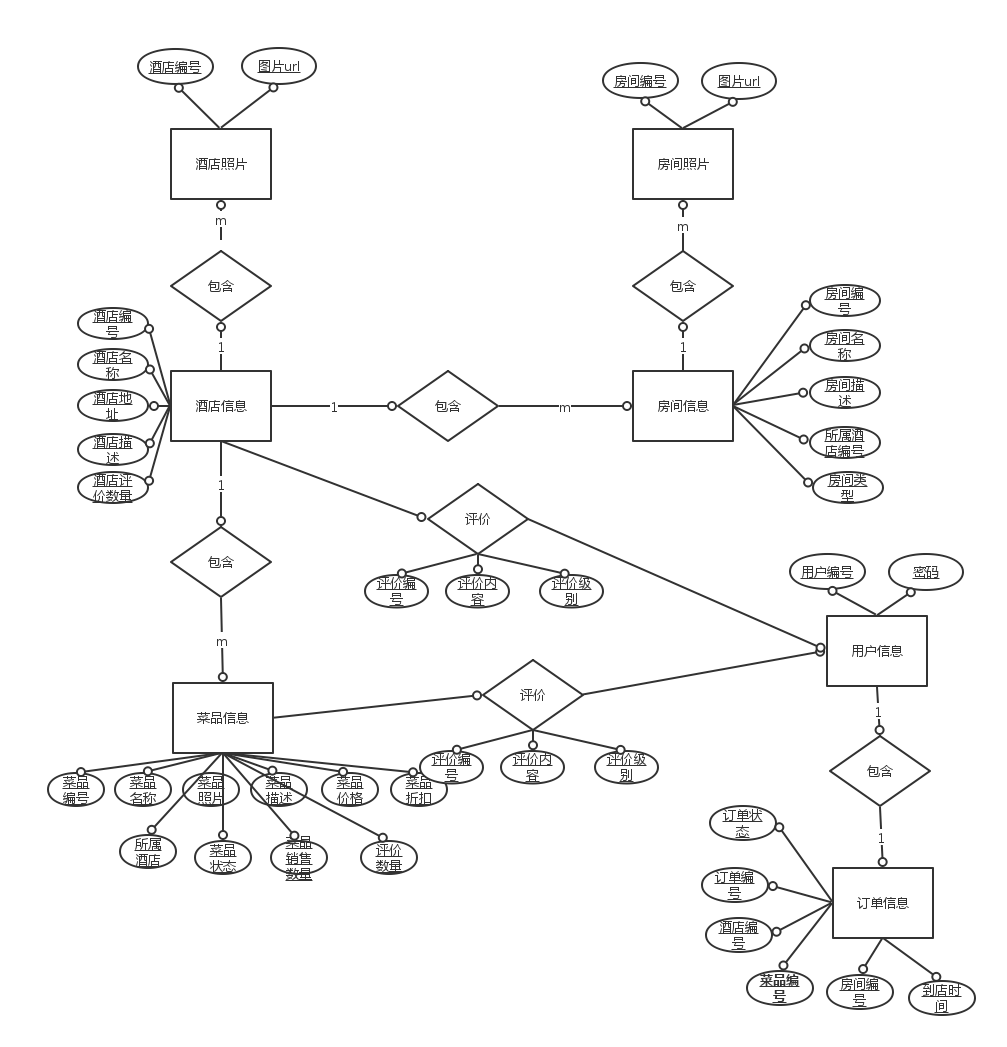
⑥ 菜品评价信息（评价编号，评价内容，被评价菜品编号，评价级别，评价人编号）

⑦ 房间信息（房间编号，房间名称，所属酒店编号，房间状态，房间类型，房间描述）

⑧房间照片信息（房间编号，图片URL）

⑨订单信息表（订单编号，酒店编号，用户编号，菜品编号，房间编号，到店时间，订单状态）

E-R图如下所示：



### 4.2.3 数据表的设计

根据上文的功能需求分析，非功能性需求分析以及数据库的详细设计，在线酒店预订和点菜平台的需要的数据表共有九张，分别是用户信息表（User），酒店信息表（h\_info）,酒店图片表（h\_pic）,酒店评价表（h\_evaluation）,房间信息表（r\_info）,房间照片表（r\_room），菜品信息表（f\_info）,菜品照片表（f\_pic），菜品评价表（f\_evaluation），订单表（order）。

在数据表的设计过程中，需要考虑到各个字段的类型和限制，本系统各个表中的各个字段都结合了系统的实际使用情况进行了相关的设计。

表4.1 用户信息表

Tab4.1 User Infomation Table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名称 | 类型长度 | 是否主键 | 是否外键 | 允许空 | 描述 |
| u\_id | Varchar(11) | 是 | 否 | 否 | 用户主键 |
| password | Varchar(20) | 否 | 否 | 否 | 用户密码 |

用户编号：用来唯一标示一位用户，本系统通过手机号来唯一标示用户，因此，id项其实就是用户的手机号，需要保持用户唯一，所以，一个手机号只能对应一位用户，不可以重复注册。

密码：用户登录站点的密码，注册后使用加密的方式存储。

表4.2 酒店信息表

Tab4.2 Hotel Information Table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名称 | 类型长度 | 是否主键 | 是否外键 | 允许空 | 描述 |
| h\_id | Varchar(20) | 是 | 否 | 否 | 酒店编号 |
| h\_name | Varchar（20） | 否 | 否 | 否 | 酒店名称 |
| h\_address | Varchar(20) | 否 | 否 | 否 | 酒店地址 |
| h\_desc | Text | 否 | 否 | 否 | 酒店描述 |
| h\_eval\_num | Int | 否 | 否 | 是 | 评价数量 |
| x | Double | 否 | 否 | 否 | 横坐标 |
| Y | Double | 否 | 否 | 否 | 纵坐标 |

酒店编号：在系统内用来唯一标示酒店，不能为空而且不能重复，一般在商家注册的时候，通过注册日期的时间戳加上随机数来唯一标示酒店；在其他表中作为外键标示该酒店；

酒店名称：酒店的真实名称；

酒店地址：酒店的真实地址，在商家注册的时候要求提供，同时会在浏览页面中显示出来；

酒店描述：在商家注册的时候，需要提供一部分文字来简单的描述酒店的特色，用来完善酒店信息，同时，也能让消费者对于酒店更加熟悉；

评价数量：酒店收到的评价数量，用户在消费完成之后，可以对酒店进行评价，评价成功，此处的评价数量增加，用来描述酒店的受欢迎程度；

横坐标/纵坐标：标示酒店的具体位置，主要用来确定哪家酒店离用户最近。

表4.3 酒店照片信息表

Table4.3 hotel\_picture Information tables

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名称 | 类型长度 | 是否主键 | 是否外键 | 允许空 | 描述 |
| h\_pic\_id | Varchar（20） | 是 | 否 | 否 | 主键 |
| h\_id | Varchar（20） | 否 | 是 | 否 | 酒店编号 |
| p\_url | varchar(100) | 否 | 否 | 否 | 图片URl |

主键：用来唯一标示此表中的一条记录

酒店编号：酒店的编号，作为外键，将酒店与酒店图片联系起来，

图片url：图片存在第三方平台上的url

表4.4 酒店评价信息表

Table4.4 hotel-evaluation table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名称 | 类型长度 | 是否主键 | 是否外键 | 允许空 | 描述 |
| pk\_eval\_h | Varchar（20） | 是 | 否 | 否 | 主键 |
| Content | Text | 否 | 否 | 否 | 评价内容 |
| h\_id | Varchar(20) | 否 | 是 | 否 | 酒店编号 |
| eval\_cat | Int | 否 | 否 | 否 | 评价等级 |
| u\_id | Varchar(20) | 否 | 是 | 否 | 用户编号 |

主键：用来唯一标示系统内的这一条评价

评价内容：用户提交评价的时候，需要填写评价的内容，主要展示给其他用户浏览

用户编号：标示哪一位用户评价了此酒店，作为外键以提供用户的基本信息；

评价等级：每一位用户在评价的时候，都要提交一份评价等级，表示自己对该酒店整体的映像是怎么样的，满分5分（最低1分）

表4.5 房间信息表

Table4.5 Room\_Information table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名称 | 类型长度 | 是否主键 | 是否外键 | 允许空 | 描述 |
| r\_id | Varchar（20） | 是 | 否 | 否 | 房间编号 |
| r\_name | Varchar（40） | 否 | 否 | 否 | 房间名称 |
| r\_status | Int | 否 | 否 | 否 | 房间状态 |
| r\_hotel | Varchar（20） | 否 | 是 | 否 | 所属酒店 |
| r\_cat | Int | 否 | 否 | 否 | 房间等级 |
| r\_desc | Text | 否 | 否 | 否 | 房间描述 |

房间编号：房间的唯一编号用来表示该房间，不可以重复，一般使用时间戳加随机数来唯一标示；

房间名称：酒店在注册信息的时候填写的房间名称

房间状态：当前房间被是否可以使用

房间所属酒店：每一个房间都是依赖于酒店而存在的，所以必须有一个酒店编号的字段将房间跟酒店挂接起来

房间等级：房间的等级，不同的等级代表不同的消费水平

房间描述：每一个房间应该有对应的简单描述。

表4.6 房间照片信息表

Table4.6 Room\_picture Information table

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 字段名称 | 类型长度 | 是否主键 | | 是否外键 | 允许空 | 描述 |
| r\_pic\_id | Varchar（20） | 是 | 否 | | 否 | 主键 |
| r\_id | Varchar（20） | 否 | 否 | | 否 | 房间编号 |
| p\_url | Varchar(100) | 否 | 否 | | 否 | 图片url |

主键：用来唯一标示此条记录，不可以重复

房间编号：用来表示房间的编号，将房间与房间照片关联起来，

图片url：图片的url

表4.7 菜品信息表

Table 4.7 food\_information table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名称 | 类型长度 | 是否主键 | 是否外键 | 允许空 | 描述 |
| f\_id | Varchar（20） | 是 | 否 | 否 | 菜品编号 |
| f\_name | Varchar（40） | 否 | 否 | 否 | 菜品名称 |
| f\_price | Double | 否 | 否 | 否 | 菜品价格 |
| f\_discount | Double | 否 | 否 | 否 | 菜品折扣 |
| f\_desc | Text | 否 | 否 | 否 | 菜品描述 |
| f\_sale\_num | Int | 否 | 否 | 否 | 销售数量 |
| f\_eval\_num | Int | 否 | 否 | 否 | 评价数量 |
| f\_status | Int | 否 | 否 | 否 | 菜品状态 |
| f\_hotel | Varchar（20） | 否 | 是 | 否 | 所属酒店 |
| f\_pic | varchar(40) | 否 | 否 | 否 | 菜品图片 |

菜品编号：唯一标示该菜品,不可重复，录入信息的时候通过时间戳加随机数唯一标示该菜品；

菜品名称：菜品的名称

菜品价格：这道菜的售价

菜品折扣：酒店可以对谋道菜品打折销售；

菜品描述：菜品的详细描述，主要用于对客户的解释；

销售数量：该菜品的销售数量

评价数量：该菜品的收到的评价数量

菜品状态：酒店的某一道菜可能暂时是不可以供给的，因此菜品必须有个状态的字段，用来表示当前这道菜是不是可以提供

所属酒店：每道菜品的都属于一个酒店的，所以需要通过酒店主键将二者关联起来

菜品照片：系统默认菜品的照片只有一张，所以可以直接在菜品信息表中加上这个字段；

表4.8 菜品评价信息表

Table 4.8 food\_evaluation information table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名称 | 类型长度 | 是否主键 | 是否外键 | 允许空 | 描述 |
| pk\_eval\_f | Varchar（20） | 是 | 否 | 否 | 主键 |
| Content | Text | 否 | 否 | 否 | 评价内容 |
| f\_id | Varchar（20） | 否 | 是 | 否 | 菜品主键 |
| eval\_cat | Int | 否 | 否 | 否 | 评价等级 |
| u\_id | Varchar（20） | 否 | 是 | 否 | 用户编号 |

主键：用来唯一标示系统中对食物的评价，不可以重复，不可以为空

评价内容：主要包括用户对该道菜的评价内容，可以展示给其他用户浏览

菜品编号：将该评论关联到一道菜品下

评价等级：对菜品的整体印象打分（5分最高，1分最低）

用户编号：标示此条评论是哪位用户给出的

表4.9 订单信息表

Table4.9 Order\_Information table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名称 | 类型长度 | 是否主键 | 是否外键 | 允许空 | 描述 |
| order\_id | Varchar（20） | 是 | 否 | 否 | 订单主键 |
| u\_id | Varchar（20） | 否 | 否 | 否 | 用户编码 |
| h\_id | varchar(20) | 否 | 否 | 否 | 酒店编号 |
| f\_ids | Text | 否 | 否 | 是 | 所点菜品 |
| Time | Date | 否 | 否 | 是 | 到店时间 |
| r\_id | varchar(20) | 否 | 否 | 否 | 房间编号 |
| Status | Int | 否 | 否 | 否 | 订单状态 |

订单编号：用来唯一标示订单的编号，不可重复，不可为空

用户编号：下订单的用户标示，不可为空

酒店编号：用户在哪家酒店消费，不可为空

房间编号：用户在那个房间里面消费，可以为空，用户可以到酒店在去选择

所点菜品：这里把用户所点的菜品的编号序列化之后存储，这样是为了将多个菜品存放在出数据库中而且还不会造成冗余，如果将用户所点的一道菜当做一个字段存储，然而，用户所点的菜品数量是不确定的，因此会造成冗余。

时间：到店时间，超过时间订单作废

状态：标示当前的订单是已经使用还是未使用