

2016-4-20

在线支付系统之预订子系统

软件工程 A3 组总体设计报告

陆宽 陈敬松 丁诗伦 钱立辉 黄卓斐

浙江大学计算机学院 13 级

目录

- 1. 引言 4
 - 1.1 标识 4
 - 1.2 编写目的 4
 - 1.2.1 编写目的说明 4
 - 1.2.2 目的 4
 - 1.2.3 预期读者 5
 - 1.3 背景 5
 - 1.4 系统概述 6
 - 1.4.1 目标 6
 - 1.4.2 用户的特点 6
 - 1.4.3 假定和约束 6
 - 1.5 定义 7
 - 1.6 基线 8
 - 1.7 参考资料 8
- 2. 总体设计 9
 - 2.1 需求规定 9
 - 2.1.1 系统功能 9
 - 2.1.2 系统性能 11
 - 2.1.3 安全性需求 11
 - 2.1.4 可用性需求 11
 - 2.1.5 可维护性需求 12

2.2 运行环境	12
2.2.1 设备	12
2.2.2 支持软件	13
3. 系统结构	13
3.1 系统功能结构	13
3.2 技术简介	15
3.3 状态图	16
3.4 类图	17
3.5 顺序流程图	18
4. 执行概念	20
4.1 信息查询与筛选	20
4.2 预订和付款	22
4.3 打分和评价	24
4.4 预订历史查询	25
4.5 信息添加与修改	27
5. 用户界面设计	31
5.1 预订酒店主界面	31
5.2 酒店详情页面	32
5.3 酒店预订界面	32
5.4 管理员卖品信息页面	33
5.5 管理员修改卖品信息界面	33
5.6 管理员订单管理页面	34

6. 数据库设计	35
6.1 E-R 图	35
6.2 机票信息(TicketsInfo)	37
6.3 机票评价(TicketsComments)	38
6.4 酒店信息(HotelInfo)	38
6.5 房间信息(RoomInfo)	39
6.6 酒店评价(HotelComments)	39
7. 接口设计	40
7.1 查询机票/酒店	40
7.2 订购机票/酒店	40
7.3 打分和评价	40
7.4 管理员操作	40
8. 系统开发难点	40
9. 故障处理	41
9.1 出错信息表	41
9.2 补救措施	41
9.3 系统维护设计	42

1. 引言

1.1 标识

文件状态	文件标识	总体设计报告
[]草稿	当前版本	1.0
[✓]正在发布	作者	软件工程 A3 组
[]正在修改	完成日期	2016 年 4 月 20 日

1.2 编写目的

1.2.1 编写目的说明

本需求说明书是为了构建在线支付系统的酒店机票预订子系统而编写,主要是面向系统开发人员及相关人员,它描述了系统的整体架构,对功能需求、性能需求和其它非功能性需求进行了详细的描述,明确表示各功能的实现过程,并阐述了适用范围及背景。另一方面,本说明书也是开发人员、维护人员、需求人员间的共同协议,为软件的设计和编码提供基础,便于软件开发过程中的控制与管理。本说明书对整个开发过程起到至关重要的影响作用,便于开发人员做足前期准备工作以及做出科学、合理的规划与预期。

1.2.2 目的

1) 实现在线酒店机票预订系统的相关业务逻辑

1. 酒店信息的查询与筛选

2. 酒店机票的预订和付款
3. 酒店机票的打分和评价
4. 酒店机票预订历史查询
5. 酒店机票信息的添加与修改

- 2) 开发用户界面，用于实现以上功能
- 3) 作为测试与验收的依据

1.2.3 预期读者

- 1) 软件用户
- 2) 项目开发人员
- 3) 软件测试人员
- 4) 软件维护人员

1.3 背景

系统名称	在线支付系统之酒店机票预订子系统
任务提出者	浙江大学软件工程课程任课老师——王章野
开发者	浙江大学 2015-2016 学年春夏学期软件工程课程 A3 组 (组长: 陆宽 组员: 黄卓斐 钱立辉 丁诗伦 陈敬松)
用户	有酒店和机票预订需求的旅客

1.4 系统概述

1.4.1 目标

在线预订系统 OBS (Online Booking System) 是作为在线支付系统 OPS (Online Payment System) 的一个重要组件, 为在线支付系统提供十分有用的 酒店机票预订的拓展功能。主要包含酒店机票的查询与筛选、酒店机票的预订和付款、酒店机票的评分、预订历史查询、管理机票和酒店信息五大功能。

1.4.2 用户的特点

在线支付系统的最终用户分为用户和管理员, 其中用户是购买酒店和机票的个体, 而管理员是提供酒店与机票信息的角色。针对这两种不同的人群我们提供了两种账户类型可供选择, 分别为个人用户和管理员用户, 个人用户通过点击预订、购买酒店, 机票。而管理员通过添加酒店、机票信息供给个人用户购买。

1.4.3 假定和约束

此次软件开发出于实验性目的, 暂无经费划拨, 希望项目开发团队能够充分利用现有资源高效地开发出符合需求的系统。

在线支付系统由 5 个组成部分构成, 每个部分完成其中一项功能, 最后进行系统集成。此项目开发期限较短, 因此各个部分务必要安排好工作进度, 保证有一定的时间进行系统集成。个人账户系统和支付系统与其他几个部分均有关联, 因此, 务必做好与其他几个部分开发团队的沟通协商工作。

此次软件开发本着实验教学的目的, 同时访问本系统的用户较少, 故对数据处理效率的能力等方面要求不高。

1.5 定义

1) **Node.js** Node.js 是一个 JavaScript 运行环境(runtime)。实际上它是对 Google V8 引擎进行了封装。V8 引擎执行 JavaScript 的速度非常快, 性能非常好。Node.js 是时下流行的 Web 应用开发平台, 它采用了单线程、异步式 I/O、事件驱动式的程序设计模型。这些特性不仅带来了巨大的性能提升, 还减少了多线程程序设计的复杂性, 进而提高了开发效率

2) **Express** 基于 Node.js 平台, 快速、开放、极简的 web 开发框架。 它有丰富的 HTTP 快捷方法和任意排列组合的 Connect 中间件, 可以使创建健壮、友好的 API 变得既快速又简单。Express 不对 Node.js 已有的特性进行二次抽象, 只是在它之上扩展了 Web 应用所需的基本功能。

3) **MySQL** 是一个开源的关系型数据库管理系统, MySQL 所使用的 SQL 语言是用于访问数据库的最常用标准化语言。MySQL 软件采用了双授权政策, 它分为社区版和商业版, 由于其体积小、速度快、总体拥有成本低, 尤其是开放源码这一特点, 一般中小型网站的开发都选择 MySQL 作为网站数据库。

4) **用例(Use Case)** 用例是一个 UML 中非常重要的概念, 在使用 UML 的整个软件开发 过程中, Use Case 处于一个中心地位。在 UML 的文档中, Use Case 的定义是: 在不展现一个系统或子系统内部结构的情况下, 对系统或子系统的某个连贯的功能单元 的定义和描述。

5) **用例图** 被称为参与者的外部用户所能观察到的系统功能的模型图, 呈现了一些参与者 和一些用例, 以及它们之间的关系, 主要用于对系统、子系统或类的功能行为进行建模。用例图展示了用例之间以及同用例参与者之间是怎样相互联系的。用例图用于对系统、子系统或类的行为进行可视化, 使用户能够理解如何使用这些元素, 并使开发者能够实现这些元素。将每个系统中的用户分出工作状态的属性和工作内容, 方便建模, 防止功能重复和

多余的类。用例图定义了系统的功能需求，它是从系统的外部看系统功能，并不描述系统内部对功能的具体实现。

6) 数据流图(Data Flow Diagram) 简称 DFD，它从数据传递和加工角度，以图形方式来表达系统的逻辑功能、数据在系统内部的逻辑流向和逻辑变换过程，是结构化系统分析方法的主要表达工具及用于表示软件模型的一种图示方法。

7) 状态图(State Diagram) 描述一个实体基于事件反应的动态行为，显示了该实体如何根据当前所处的状态对不同的时间做出反应的。通常我们创建一个 UML 状态图是为了以下的研究目的：研究类、角色、子系统、或组件的复杂行为。

8) CRC Card(Cyclical Redundancy Check) 类责任和交互，简称 CRC 卡片。在面向对象程序设计中，用来阐述类、类的行为和类的责任的一个非常好的途径

1.6 基线

[1]《在线支付系统之酒店机票预订子系统需求分析》(陆宽、丁世伦、黄卓斐、钱立辉、陈敬松, 2016)

1.7 参考资料

[2]《软件工程：时间的研究方法（原书第八版）》(S. Pressman, 2011)

[3]《参考模板_总体设计报告》(王章野, 2015)

[4]《Node.js 开发指南》(BYVoid, 2012.7)

[5]《Project Requirement Online Payment System》(王章野, 2015)

2. 总体设计

2.1 需求规定

2.1.1 系统功能

1) 酒店信息的查询与筛选

酒店信息的查询与筛选功能主要给用户提供一个能够根据自身需求, 获得自己需要的酒店和机票的相关信息, 其中酒店的信息可以通过载明以下条目进行搜索:

- 城市
- 酒店名
- 入住日期
- 离店日期

航班的信息可以通过载明以下条目进行搜索

- 出发地
- 目的地
- 出发时间
- 返程时间

2) 酒店机票的预订和付款

在获得所需的酒店或者机票信息后, 用户可以预订酒店或者机票, 通过填写预订信息, 系统会自动生成一个订单, 同时将订单信息交给支付系统引导用户付款。用户预订信息应载明

- 房间数量

- 住客姓名
- 联系手机
- Email
- 最晚抵店日期

该模块能够展示酒店剩余房间信息以便用户作为参考，每当一个预订订单生成后，剩余房间自动分配给用户，但是如果用户在 30 分钟内没有完成支付行为，将会取消订单。

3) 酒店机票的打分和评价

用户订单完成后，可以对酒店或者机票进行评价，应载明

- 星级评分（满分五星）
- 评语（用户的实际体验对酒店进行点评，打分和评价的数据会显示在酒店的详细信息中）

4) 酒店机票预订历史查询

用户可以通过在线预订系统查询自己过去的订单，通过这个功能用户可以知道自己之前住过的酒店或者买过的机票的信息。

5) 酒店机票信息的添加与修改

作为酒店或者机票的提供商，需要能够以管理员的身份登录到系统中，对自己提供的酒店与机票信息进行修改，删除，同时添加新的信息，应载明：

- 酒店名称
- 详细地址
- 联系电话
- 房间信息

- 种类 1 总共数量 1
- 种类 2 总共数量 2

2.1.2 系统性能

系统配置：2.6GHz CPU, 2.0G Memory, 7200rpm Disk

最大访问容量：通知支持 500 用户并发访问

容量：支持至少 10000 条记录存储

响应速度：单个用户在线时，网页响应时间少于 1s，信息搜索响应时间少于 2s。

500 个用户在线时，网页响应时间少于 2s，信息搜索响应时间少于 5s。

2.1.3 安全性需求

1) 保密性

对于个人信息验证功能, 应避免未被授权的用户浏览其他用户的账户密码信息。完善的访问控制要避免系统中已注册的合法用户非法访问和使用系统资源。用户密码等敏感信息需要加密储存与传输。在用户登录过程中, 应避免数据库嵌入、密码强制破解、伪造会话侵入等情况。

2) 完整性

系统中的用户信息需要保持完整性, 要避免各种形式的数据丢失。同时, 也要避免未被授权用户偶然或恶意地插入、删除、修改系统中的各种用户信息。

2.1.4 可用性需求

多浏览器支持：该系统需要能够在流行的 web 浏览器如 Firefox 和 IE 中正确的显示和运行功能。

2.1.5 可维护性需求

- 1) 系统设计时应备齐各类文档，操作人员只需阅读文档就可知道所有的操作，维护人员也可以通过文档了解系统结构、设计思路，以便快速掌握整个系统。
- 2) 程序设计时应保持良好的编程风格，结构清晰、注释明确，使调试、测试人员能快速定位各种错误。
- 3) 交付时应文档齐全，说明详尽，文档符合相关标准。

2.2 运行环境

2.2.1 设备

由于条件有限，我们不能提供专门的服务器运行系统，请开发人员利用配置较高的PC机作为服务器。

服务器端的设备要求如下：

计算机	CPU: $\geq 2.0\text{GHz}$ 内存: $\geq 2.0\text{GB}$
外围设备	无需
硬盘	容量: $\geq 100\text{GB}$ 转速: $\geq 7200\text{rpm}$
通讯设备	网线: 具有良好的数据传输能力 网卡: 100M

2.2.2 支持软件

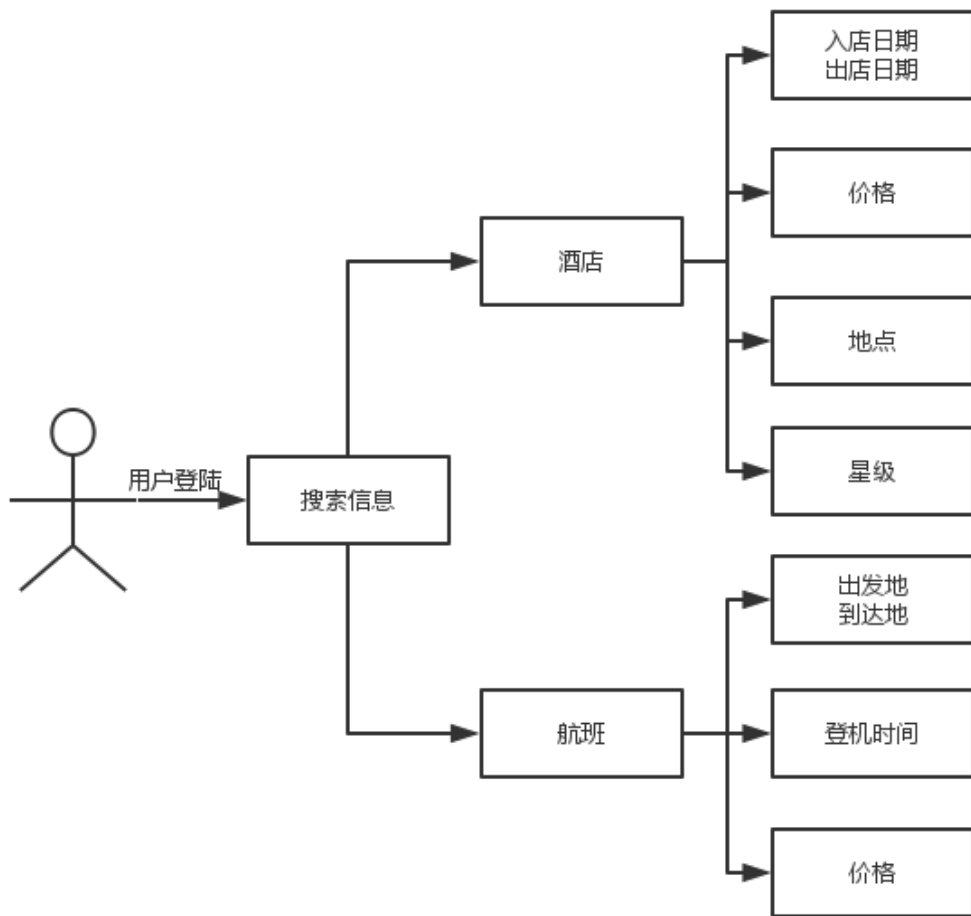
操作系统	
数据库平台	
Web 服务器	
开发工具	Atom 编辑器配合 termjs 插件
测试工具	Node-inspector
浏览器	谷歌浏览器，Safari 浏览器

3. 系统结构

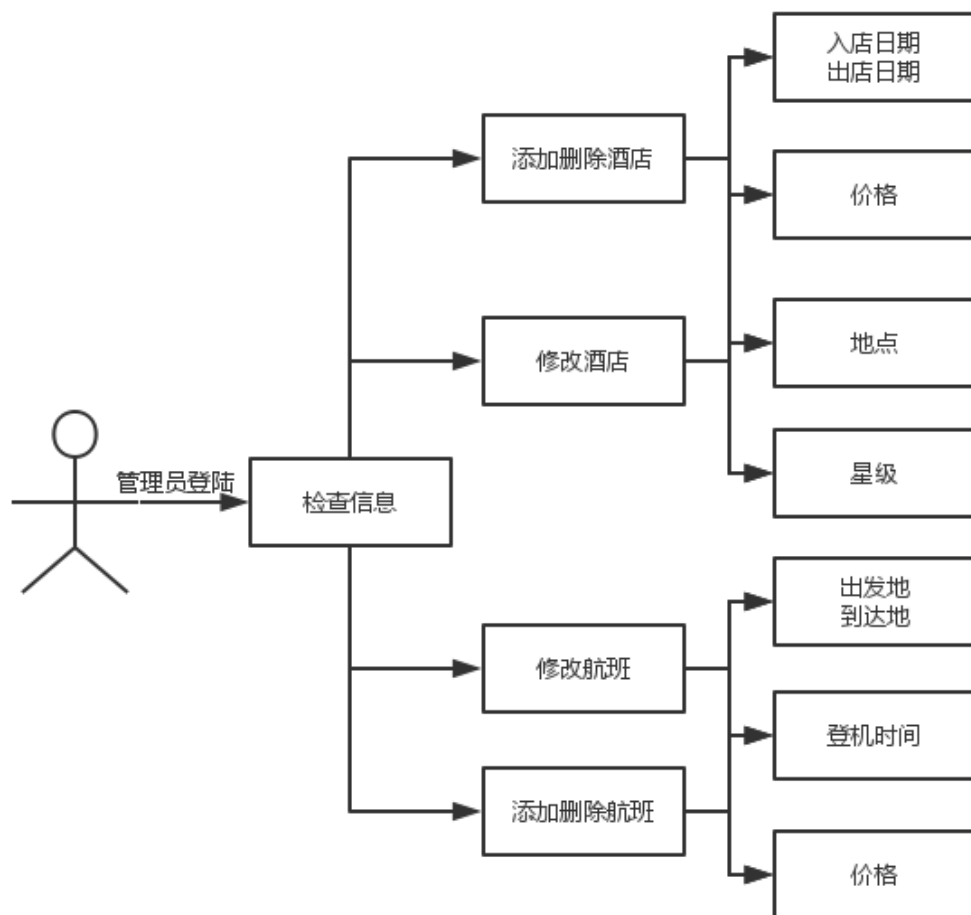
3.1 系统功能结构

在之前的需求分析中，我们组对于我们的系统进行了相应的用例分析，具体如下所示：

普通用户订酒店、机票用例图：



管理员用户添加，修改酒店机票用例图：



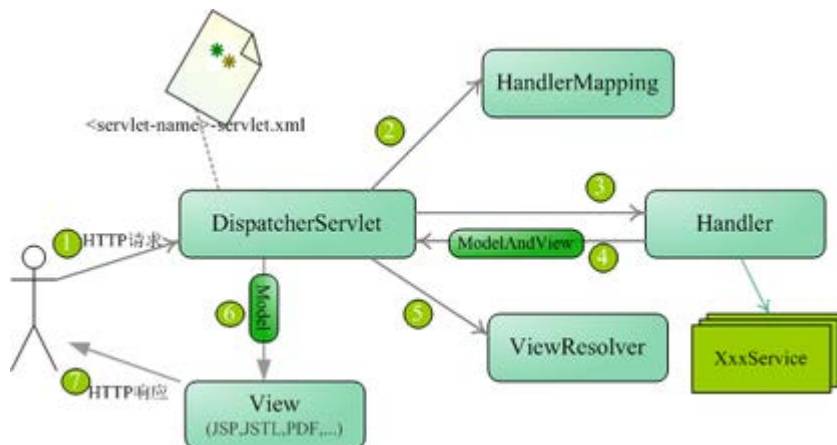
3.2 技术简介

我们机票酒店预订系统中，采用了 MVC 的模式进行设计和实现：

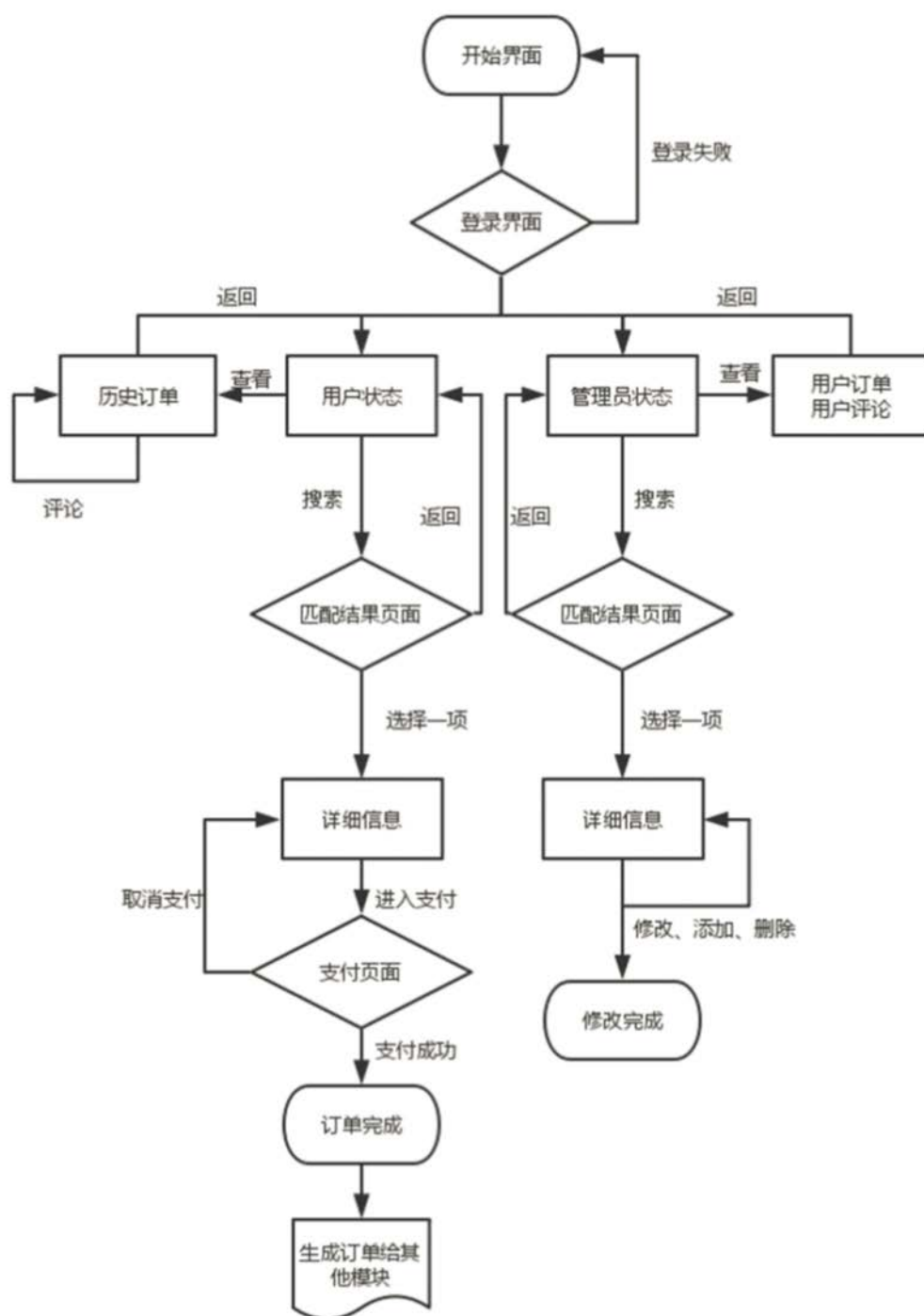
- **Model (模型)** 是应用程序中用于处理应用程序数据逻辑的部分。通常模型对象负责在数据库中存取数据。
- **View (视图)** 是应用程序中处理数据显示的部分。通常视图是依据模型数据创建的。
- **Controller (控制器)** 是应用程序中处理用户交互的部分。通常控制器负责从视图读取数据，控制用户输入，并向模型发送数据。

MVC 分层有助于管理复杂的应用程序, 这样可以在一个时间内专门关注一个方面 MVC

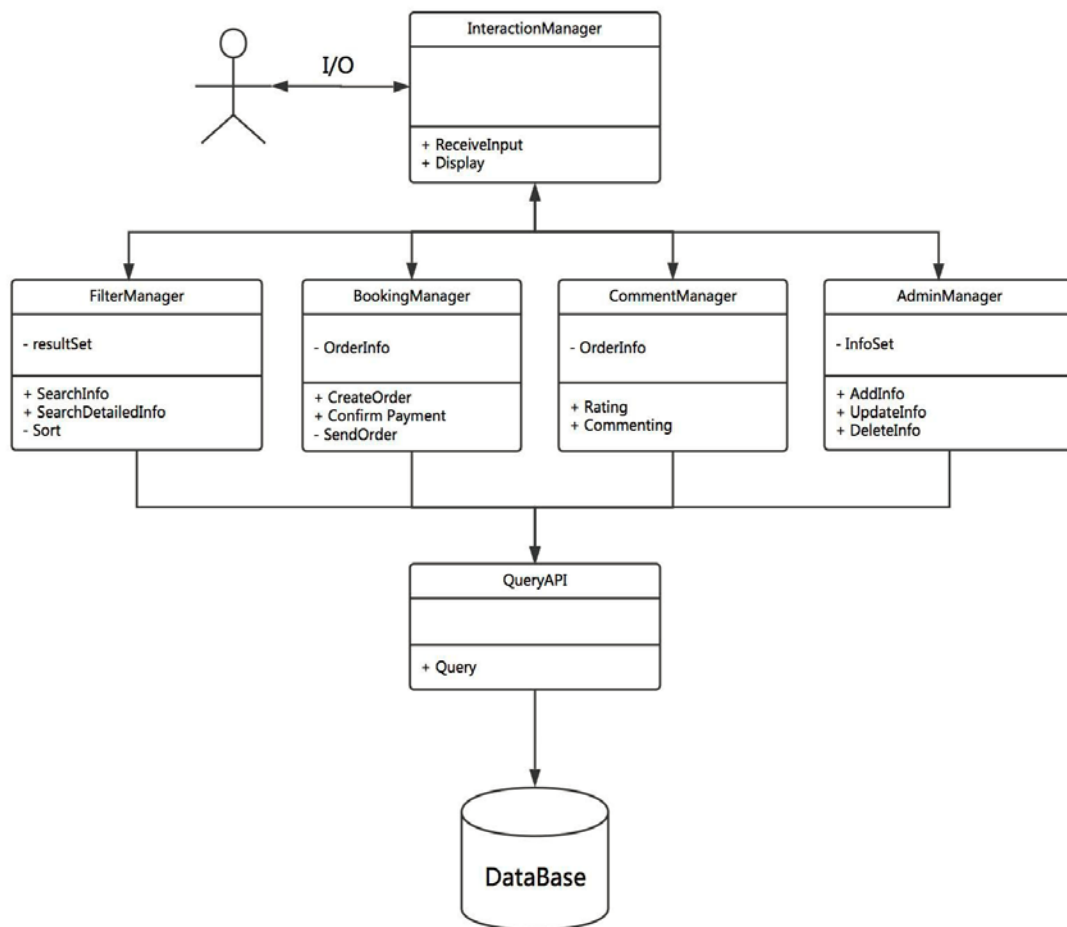
分层同时也简化了分组开发。不同的开发人员可同时开发视图、控制器逻辑和业务逻辑。



3.3 状态图



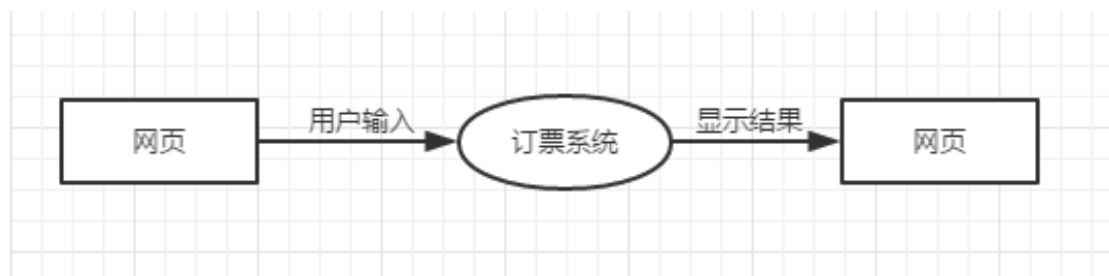
3.4 类图



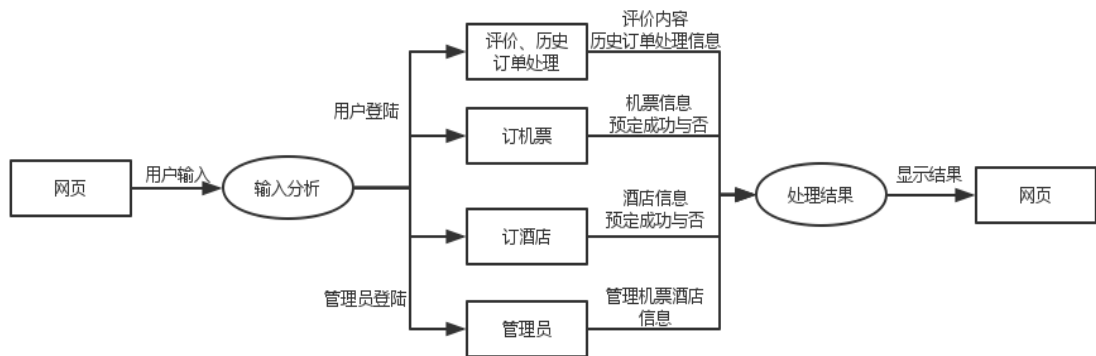
3.5 顺序流程图

此外，我们也在需求报告中基于数据流进行了相应的分析如下：

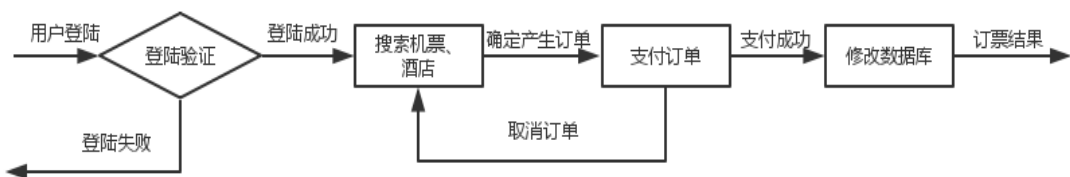
第 0 层数据流图：



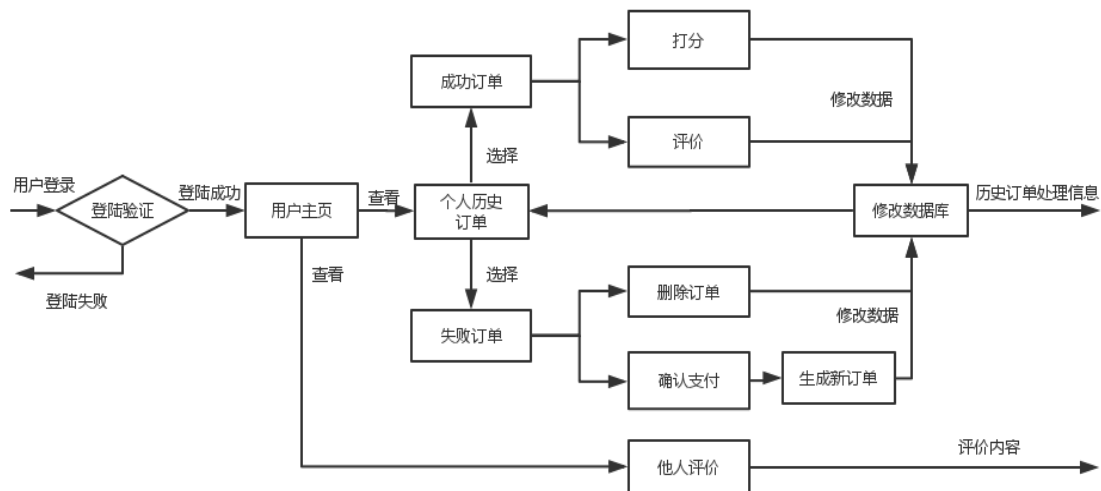
第 1 层数据流图：



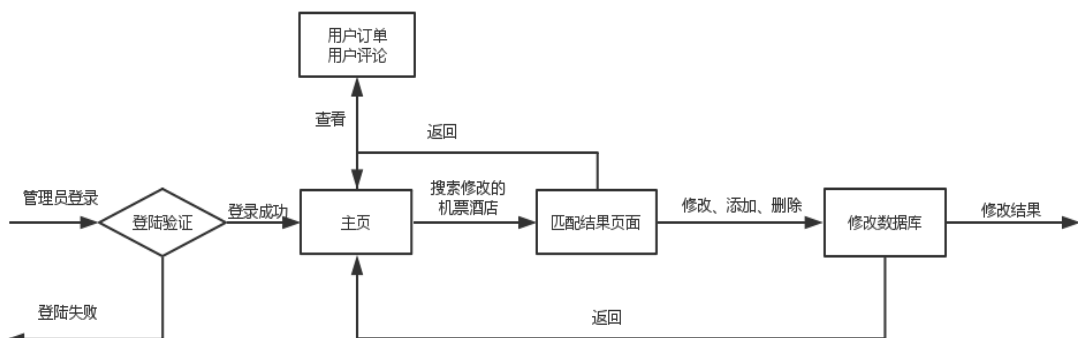
订机票、酒店子系统的第 2 层数据流图：



历史订单及评价子系统的第 2 层数据流图：



管理员管理信息子系统的第 2 层数据流图：



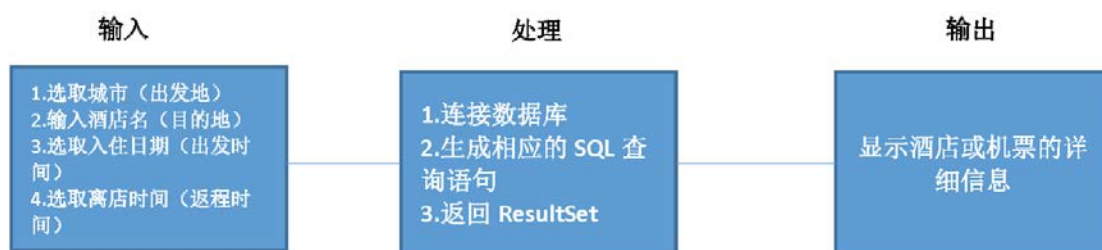
4. 执行概念

4.1 信息查询与筛选

4.1.1 模块概述

酒店信息的查询与筛选功能主要给用户提供一个能够根据自身需求，获得用户需要的酒店和机票的相关信息。

4.1.2 IPO 图



4.1.3 功能

可供用户查询的酒店的信息包括：城市，酒店名，入住日期，离店日期。

航班的信息包括：出发地，目的地，出发时间，返程时间。通过输入这些条件用户可以查找到符合自己需求的信息。

4.1.4 输入项

酒店：

名称	标志	类型和格式	输入方式
城市	city_name	varchar(20)	网页上选取
酒店名	hotel_name	varchar(20)	手工输入
入住日期	check_in_date	date	网页上选取
离店日期	check_out_date	date	网页上选取

机票：

名称	标志	类型和格式	输入方式
出发地	departure	varchar(20)	网页上选取
目的地	destination	varchar(20)	网页上选取
出发时间	setout_time	datetime	网页上选取
返程时间	return_time	datetime	网页上选取

4.1.5 输出项

名称	标志	类型和格式	输出方式
详细信息	detail	text	由脚本输出

4.1.6 设计方法

功能	视图	控制器	模型	结果视图	说明
用户查询	搜索筛选 视图	输入搜索 条件, 将条 件传送到 model 分 析, model 从数据库 获取结果 返回 View 显示	根据搜索 条件转为 相应的 SQL 语句, 向数据库 索要结果 集。	显示搜索 对象的详 细信息	查询结果 存在或不 存在, 结果 有几条记 录。

4.2 预订和付款

4.2.1 模块概述

在获得所需的酒店或者机票信息后，用户在这个模块可以预订酒店或者机票。

4.2.2 IPO 图



4.2.3 功能

通过填写预订信息,系统会自动生成一个订单,同时将订单信息交给支付系统引导用户付款, 该模块能够展示酒店剩余房间信息以便用户作为参考,每当一个预订订单生成后, 剩余房间自动分配给用户,但是如果用户在 30 分钟内没有完成支付行为,将会取消订单。

4.2.4 输入项

酒店：

名称	标志	类型和格式	输入方式
城市	city_name	varchar(20)	网页上选取
酒店名	hotel_name	varchar(20)	手工输入
入住日期	check_in_date	date	网页上选取
离店日期	check_out_date	date	网页上选取

机票：

名称	标志	类型和格式	输入方式
出发地	departure	varchar(20)	网页上选取
目的地	destination	varchar(20)	网页上选取
出发时间	setout_time	datetime	网页上选取
返程时间	return_time	datetime	网页上选取

4.2.5 输出项

名称	标志	类型和格式	输出方式
剩余房间（机票） 信息	Left_info	int	由脚本输出
订单详情	Order_info	text	由脚本输出

4.2.6 设计方法

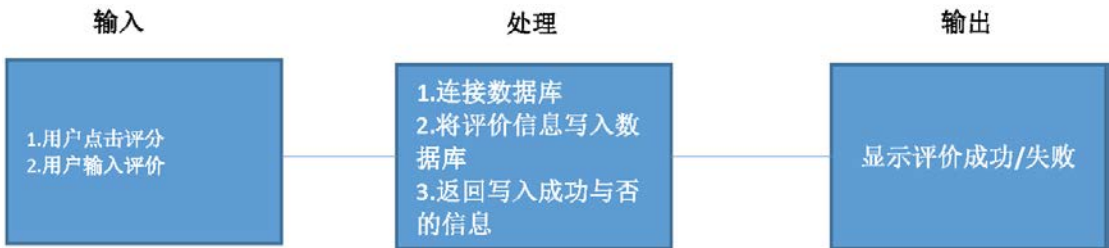
功能	视图	控制器	模型	结果视图	说明
预订付款	填写预订 信息视图	输入订单 详细信息， 交给模型 分析。	根据控制 器传来的 预订信息 生成订单， 在视图上 显示。	订单视图	返回预订 成功与否

4.3 打分和评价

4.3.1 模块概述

用户订单完成后，通过该模块对酒店或者机票进行评价。

4.3.2 IPO 图



4.3.3 功能

评价功能包括两个方面，一个是打分评价，五颗星为满分，另一个是能够撰写评语，根据用户的实际体验对酒店进行点评。打分和评价的数据会显示在酒店的详细信息中。

4.3.4 输入项

酒店：

名称	标志	类型和格式	输入方式
评分	score	float	网页上点击
评价	comments	text	手工输入

机票：

名称	标志	类型和格式	输出方式
评价结果	comment_res	bool	由脚本输出

4.3.5 输出项

名称	标志	类型和格式	输出方式
评价结果	comment_res	bool	由脚本输出

4.3.6 设计方法

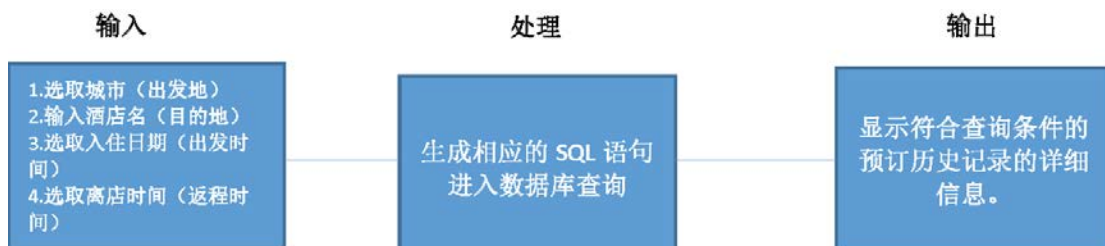
功能	视图	控制器	模型	结果视图	说明
用户评价和评分	评价与评分视图	接收用户输入的评价和评分信息，转到模型进行分析。	将评价和评分写入数据库相应表项，并把结果反馈到结果视图。	评价结果视图	显示评分或评价成功/失败。

4.4 预订历史查询

4.4.1 模块概述

用户通过该模块查询到自己的预订历史信息。

4.4.2 IPO 图



4.4.3 功能

用户可以通过在线预订系统查询自己过去的订单，通过这个功能用户可以知道自己之前住过的酒店或者买过的机票的信息，以便再次购买时能够方便的找到。

4.4.4 输入项

酒店：

名称	标志	类型和格式	输入方式
城市	city_name	varchar(20)	网页上选取
酒店名	hotel_name	varchar(20)	手工输入
入住日期	check_in_date	date	网页上选取
离店日期	check_out_date	date	网页上选取

机票：

名称	标志	类型和格式	输入方式
出发地	departure	varchar(20)	网页上选取
目的地	destination	varchar(20)	网页上选取
出发时间	setout_time	datetime	网页上选取
返程时间	return_time	datetime	网页上选取

4.4.5 输出项

名称	标志	类型和格式	输出方式
预订历史记录	history_record	text	由脚本输出

4.4.6 设计方法

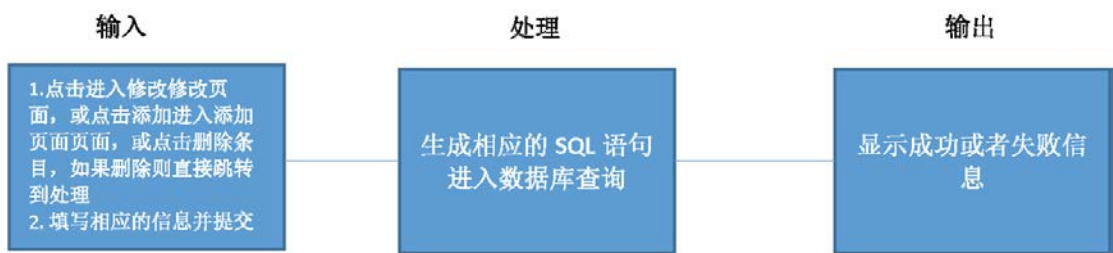
功能	视图	控制器	模型	结果视图	说明
历史记录 查询	预订历史 查询视图	接收用户 输入的预 订历史查 询条件,交 给模型生 成相应的 SQL 条件 查询语句。	连接数据 库,根据条 件查询符 合条件的 预订历史 订单,并把 相应订单 信息显示 到结果视 图中。	历史订单 记录视图	文本形式 显示符合 条件的历 史订单记 录

4.5 信息添加与修改

4.5.1 模块概述

作为酒店或者机票的提供商,需要能够以管理员的身份登录到系统中,进行相关信息的添加、删减或修改。

4.5.2 IPO 图



4.5.3 功能

该模块负责实现供应商以管理员的身份登录到系统中，对自己提供的酒店与机票信息进行修改、删除，还可以添加新的信息，该模块提供了管理员这些功能让管理员能够及时的更新信息。

4.5.4 输入项

酒店：

名称	标志	类型和格式	输入方式
酒店名	Hotel_Name	varchar(20)	手工输入
所在省	Province	varchar(20)	网页上选取
所在城市	City	varchar(20)	网页上选取
详细地址	Address	varchar(50)	手工输入
星级	Stars	int	网页上选取
介绍文本	Description	Text	手工输入
联系电话	PhoneNumber	Varchar(20)	手工输入

机票：

名称	标志	类型和格式	输入方式
出发地	Departure	verchar(50)	网页上选取
机场	Airport	verchar(50)	手工输入
目的地	Destination	verchar(50)	网页上选取
出发时间	Depart_time	datetime	网页上选取
抵达时间	Arrive_time	datetime	网页上选取

总票数	Total	int	手工输入
单价	Price	Int	手工输入

4.5.5 输出项

名称	标志	类型和格式	输出方式
成功信息	Success	text	由脚本输出
失败信息	Failure	Text	由脚本输出

4.5.6 设计方法

功能	视图	控制器	模型	结果视图	说明
添加	信息添加界面	控制器接收所有输入项, 将获得的数据输送给模型	判断合法性后在数据库中新建一条相关的条目, 并根据模型结果返回成功或失败信息到视图	显示成功或失败信息	成功会显示祝贺信息, 失败会说明具体原因
修改	信息修改界面	控制器接收所有输	判断合法性后在数	显示成功或者失败	成功会显示祝贺信

		入项, 将获得的数 据输送给模 型	据 库 中 修 改 相 应 的 条 目, 返 回 成 功 或 者 失 败 信 息 给 视 图	信 息	息, 失 败 会 说 明 具 体 原 因
删除	信 息 修 改 界 面	控 制 器 接 收 删 除 信 号	判 断 合 法 性 之 后 在 数 据 库 中 删 除 相 应 的 条 目, 返 回 成 功 或 失 败 信 息 给 视 图	显 示 成 功 或 者 失 败 信 息	成 功 会 显 示 祝 贺 信 息, 失 败 会 说 明 具 体 原 因

5. 用户界面设计

5.1 预订酒店主界面

Bookkitty

用户 zyy

在线预定

订单管理

我的账户

预订酒店

预订机票

搜索酒店

酒店名: 所在城市: 杭州市 地段: 入住日期: 2016. 4. 1 离店日期: 2016. 4. 4 价格: 最低价格 - 最高价格 元

酒店	位置	价格	评分	评价	特价推荐
	云恋情侣酒店	北环	130. 0-588. 0	4. 5	华顺旅馆 80. 0 ¥60. 0
<div>zyy: 很有情调, 好评! 2016. 3. 21 想不出昵称: 位置有点偏, 不过设施不错, 服务态度好。 2016. 3. 21 zzz: 差评, 连电视都没有!! 2016. 3. 19 哎哟不错: 还行吧…… 2016. 3. 12</div>					[玉古路]风尘客栈 210. 0 ¥188. 0
	如家快捷酒店	西湖区	181. 0-499. 0	4. 2	布丁连锁酒店 199. 0 ¥152. 0

5.2 酒店详情页面



5.3 酒店预订界面



5.4 管理员卖品信息页面

Bookkitty

卖家 zyy

卖品信息

订单管理

我的账户

酒店管理

航班管理

君庭大酒店



酒店创建日期: 1949. 10. 1
地址: 紫金港
目前定价: ¥ 200-2000

详情:
一家不知道怎么介绍的酒店。

紫金港大酒店



酒店创建日期: 1949. 10. 1
地址: 紫金港
目前定价: ¥ 200-2000

详情:
一家不知道怎么介绍的酒店。

布丁酒店



酒店创建日期: 1949. 10. 1
地址: 紫金港
目前定价: ¥ 200-2000

详情:
一家不知道怎么介绍的酒店。

5.5 管理员修改卖品信息界面

Bookkitty

卖家 zyy

卖品信息

订单管理

我的账户

< 返回酒店列表

酒店详情

编辑详情



云恋情侣酒店

商家: YunlianHotel
价格: 130. 0-588. 0
地址: 杭州市北环厦边路248号
电话: 18812345678

房间类型

新增房间

价格

已订房间/房间数量



森林主题套间 (不含早)

¥288. 0

10/20

编辑信息



热恋巧克力主题套间 (不含早)

¥588. 0

0/3

编辑信息

提交修改

网友评论

5.6 管理员订单管理页面

Bookkitty

卖家 zyy

卖品信息

订单管理

我的账户

按时间 ↑ 按状态 筛选: 所有时间 所有状态

3月26日 3月27日 3月28日 3月29日 3月30日 3月31日

订单号: SE000010325 2016. 3. 26

 ZaneXiao ¥66. 2 已付款

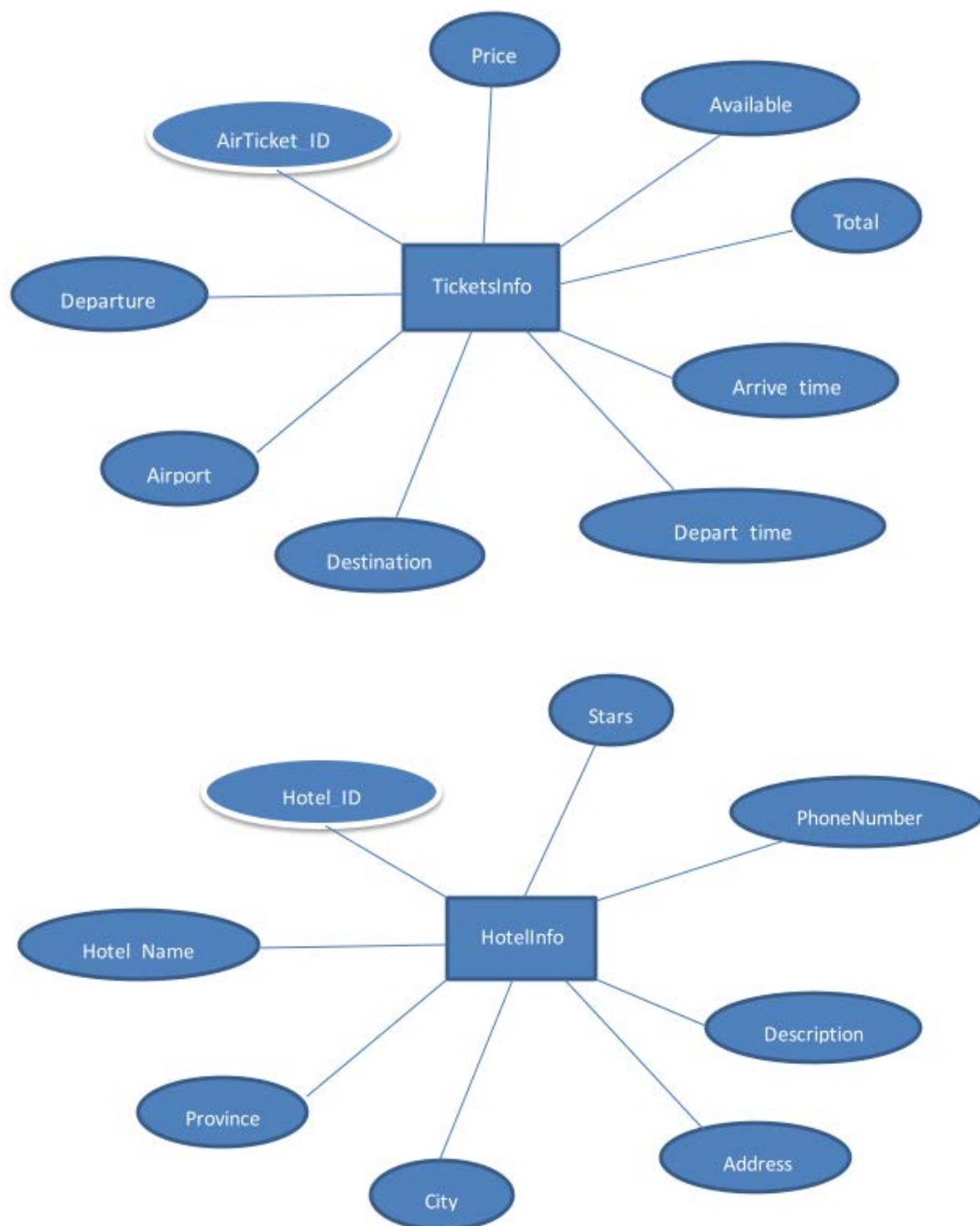
订单号: SE000010310 2016. 3. 26

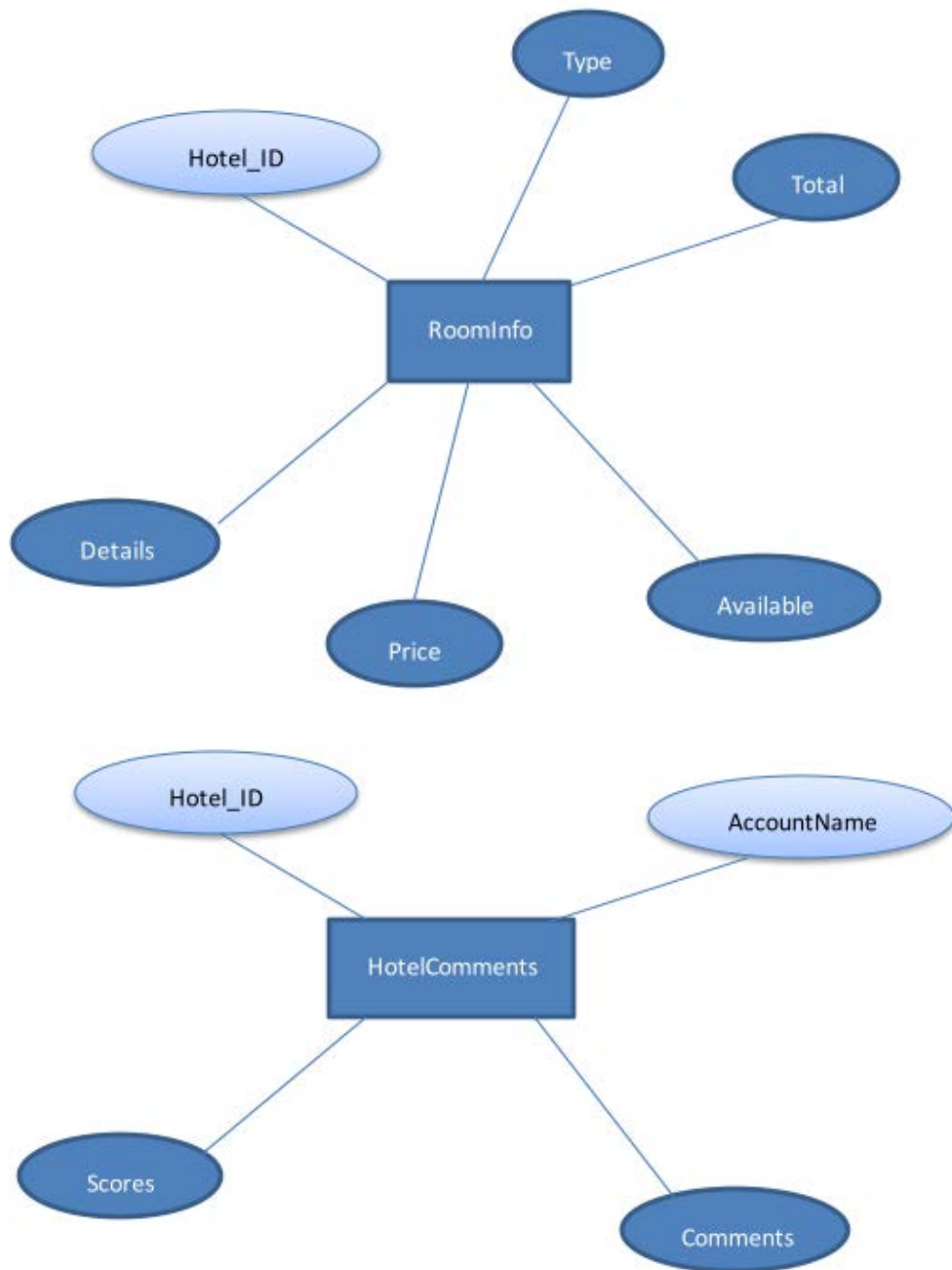
 EwinYe ¥99. 8 已发货

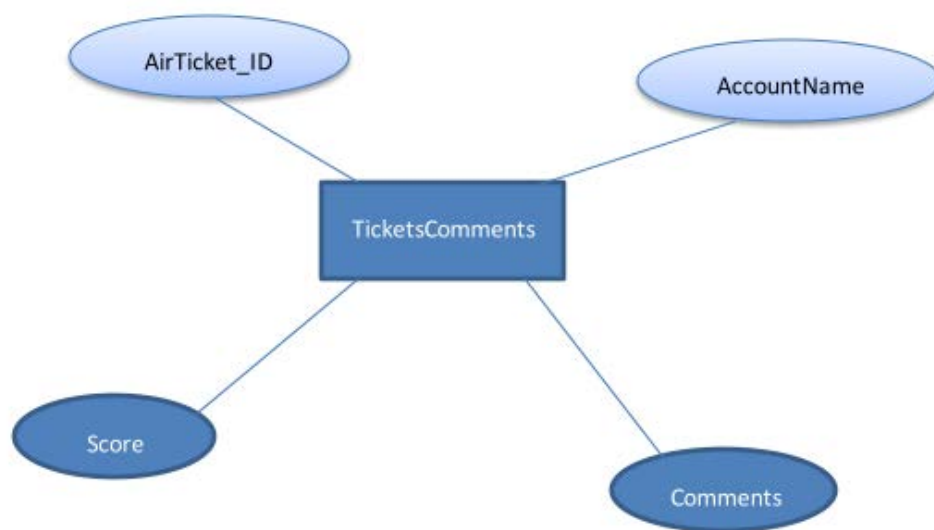
订单号: SE000010296 2016. 3. 26

6. 数据库设计

6.1 E-R 图







6.2 机票信息(TicketsInfo)

含义	属性	类型	说明	实例
机票 ID	AirTicket_ID	char(10)	Primary key	abcdefghijkl
出发地	Departure	varchar(50)		Hongzhou
机场	Airport	varchar(50)		Xiaoshan
目的地	Destination	varchar(50)		Shanghai
出发时间	Depart_time	datetime		2016-04-17- 08-00-00
抵达时间	Arrive_time	datetime		2016-04-17- 08-00-00
总票数	Total	int		100
剩余票数	Available	int		50
单价	Price	int		1000

6.3 机票评价(TicketsComments)

含义	属性	类型	说明	实例
机 票 ID	AirTicket_ID	Char(10)	Foreign key (Ticketsinfo(AAirTicket_ID))	iamaticket
评分	Score	float		4.0
用 户 ID	AccountName	Char(20)	Foreign key	YourFather
文本评 价	Comments	Text		Hahahaitsgood

6.4 酒店信息(HotelInfo)

含义	属性	类型	说明	实例
酒店 ID	Hotel_ID	Char(10)	Primary key	IgDfa40000
酒店名	Hotel_Name	varchar(20)		Huanglong
所在省	Province	varchar(20)		Zhejiang
所在城市	City	varchar(20)		Hangzhou
详细地址	Address	varchar(50)		The West Lake
星级	Stars	int		4
介绍文本	Description	Text		An incredible hotel
联系电话	PhoneNumber	Varchar(20)		12345678901

6.5 房间信息(RoomInfo)

含义	属性	类型	说明	实例
酒店 ID	Hotel_ID	Char(10)	Foreign Key	IgDfa40000
类型	Type	Char(10)		TinyBed
单价	Price	Float		500
剩余数量	Available	Int		2
详细信息	Details	Text		It' s a small room with a tiny bed
总数	Total	Int		3

6.6 酒店评价(HotelComments)

含义	属性	类型	说明	实例
酒店 ID	Hotel_ID	Char(10)	Foreign Key	IgDfa40000
评分	Scores	Float		5
用户 ID	AccountName	Char(20)	Foreign Key	YourFather
文本评价	Comments	Text		The floor was a total mess

7. 接口设计

7.1 查询机票/酒店

用户可以通过输入特定的筛选条件来选出符合需求的机票/酒店，查询出的结果来自数据库，并显示在 UI 界面上。

7.2 订购机票/酒店

用户可以订购某一特定的机票/酒店，同时生成订单发送给支付模块处理。

7.3 打分和评价

用户可以对一笔已经完成的交易进行打分和评价，结果将被保存在数据库，所打分将形成该机票/酒店的平均分。

7.4 管理员操作

上面的三个接口都是为用户准备的，这个接口是为管理员准备的，管理员可以通过这个接口进行添加、修改和删除操作。

8. 系统开发难点

本系统主要负责用户对机票和酒店的订购服务，所以难点主要有以下几点：首先，如何保证数据库的一致性，即如果当多个用户同时操作时，如何保证他们所看到的信息都是相同的；第二，当网络通讯出现状况、或者数据库出现故障时，如何处理丢失的数据；第三，如何合理地设计数据库，使得存储的信息便于增加、删除和修改，并尽量减少数据的冗余；最后，

如何设计 UI 界面才可以让用户获得更好的用户体验，使得新用户可以很快上手。

9. 故障处理

9.1 出错信息表

我们可以保证的是系统的逻辑是没有错误的，不过在运行时难免会遇到不可控错误，比如网络出错等等。所以，为了便于管理人员更好地维护系统，提高系统的服务质量，在此列出该系统可能发生的一系列错误，并给出相应的解决方案。

系统错误提示信息	解决方案
网络连接失败	检查服务器网络状况
数据库连接失败	检查驱动程序是否正常，检查数据库进程是否正常运行
服务器无响应	重启服务器或限制访问流量
UI 界面出错	检查浏览器是否正常运行

9.2 补救措施

当网络连接失败时，管理员应该立即启用备用网络，同时排查网络出错的原因，以尽快疏通网络，恢复服务。

因为我们使用的是租的云端服务器，所以当服务器无响应时，管理员应该立即联系服务器运行商的技术人员，检查出错的原因，恢复服务器。

数据库连接失败一般是因为数据库进程或驱动程序意外出错，因此，应立即检查数据库是否正常开启，同时在服务器上检查数据库是否能正常连接。

UI 界面出错一般是浏览器出错，所以应该立即检查浏览器运行是否正常。

9.3 系统维护设计

对于网络，管理员应该定期维护网络，保持良好的网络环境。对于服务器，服务器运营商应该有一套完备的服务器设备维护计划。对于数据库，主要的维护工作由程序自主完成。需要确保数据库的一致性，当本系统进行数据库的增删改操作时，禁止其他系统通过本系统获取信息，否则将会出现数据不一致的情况。