|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 分类号 |  | |  | | | 密 级 | |  |
| U D C |  | |  | | | 编 号 | | 10486 |
|  | | | | | | | | |
| 武汉大学logo  **硕 士 专 业 学 位 论 文** | | | | | | | | |
|  | | | | | | | | |
| 软件架构驱动的  公园票务系统的设计与实现 | | | | | | | | |
|  | | 研究生姓名 | | ： |  | |  | |
| 学号 | | ： |  | |
| 指导教师姓名、职称 | | ： |  | |
| 专业类别（领域） | | ： | 软件工程 | |
|  | |  |  | |
|  | |  |  | |
|  | | | | | | | | |
| 二〇一八年四月 | | | | | | | | |

**A Thesis Submitted to Wuhan University**

**For Degree of Master**

**Design and Implementation of**

**Park Ticketing System**

**Driven by Software Architecture**

By

XXX

School of Computer Science

Wuhan University， P.R.China

**April，2018**

论文原创性声明

本人郑重声明：所呈交的学位论文，是本人在导师指导下，独立进行研究工作所取得的研究成果。除文中已经标明引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写过的研究成果。对本文的研究做出贡献的个人和集体，均已在文中以明确方式标明。本声明的法律结果由本人承担。

学位论文作者（签名）：

年 月 日

摘要

公园的传统售票方式暴露出了不可忽视的缺点。在互联网浪潮的影响下，公园的传统售票方式急需升级，使用公园票务系统可以规范售票过程和检票过程，提高工作效率，提升管理水平，具有很大的现实意义。

本文介绍了在线票务系统的研究背景和应用意义，对国内外的票务系统现状进行分析，对此次实现公园票务系统使用到的关键技术进行详细说明。本文对软件架构的概念和重要性进行介绍，选择了一种经典的软件架构设计方法ADMEMS（Architecture Design Method has been Extended to Method System），对使用此方法进行软件架构的优势进行介绍，详细说明了此方法的步骤，为公园票务系统的软件架构设计提供指导。本文使用ADMEMS方法，按照预架构阶段、概念架构阶段、细化架构阶段的步骤，使用了多种工具，完成了公园票务系统的架构设计。最后，本文详细介绍了软件架构驱动的公园票务系统的实现，其中包括系统的数据库设计、游客子系统的实现、后台管理子系统的实现、验票子系统的实现。三个子系统共同组成了公园票务系统，满足了各方需求。

公园票务系统投入使用后，售票和检票的效率得到了提高，公园的运营数据可以实时统计，通过会员充值活动，游客得到了实惠，公园也提高了经济收益。此票务系统对公园的运营和管理有积极作用。

关键词：票务系统；软件架构；ADMEMS方法

**Abstract**

The traditional way of ticket sales in the park has revealed its shortcomings that cannot be ignored. Under the influence of the Internet tide, the park’s traditional ticketing methods need to be upgraded urgently. The use of the park ticketing system can standardize the selling process and the checking process of tickets, improve work efficiency and enhance the management level, which is of great practical significance.

This paper introduces the research background and application significance of online ticketing systems, analyzes the current situation of the ticketing system local and overseas, and explains the key technologies used to implement the park ticketing system in detail. This paper introduces the concept and importance of software architecture, selects a classic software architecture design method ADMEMS（Architecture Design Method has been Extended to Method System）, introduces the advantages of using this method for software architecture, and describes the steps of this method in detail. ADMEMS provides guidance for software architecture design of park ticketing systems. This paper uses the ADMEMS, following the steps of Pre-Architecture, Conceptual Architecture and Refined Architecture, using a variety of tools to complete the park ticketing system architecture design. Finally, this paper introduces the implementation of the park ticketing system driven by software architecture in detail, including the database design of the system, the implementation of the tourist subsystem, the implementation of the background management subsystem, and the implementation of the ticket checking subsystem. The three subsystems together constitute the park ticketing system and meet the needs of all.

After the park ticketing system was put into use, the efficiency of ticket sales and ticket examinations has been improved, and the park's operational data can be calculated timely. Through membership recharging activities, visitors have benefited and the park has also improved economic returns. This ticketing system has a positive effect on the operation and management of the park.

**Key words**: Ticketing System; Software Architecture; ADMEMS

# 目 录

[**摘要** I](#_Toc512620860)

[**Abstract** I](#_Toc512620861)I

[**目录** III](#_Toc512620862)

[**第一章 绪论** 1](#_Toc512620863)

[1.1 研究背景及意义 1](#_Toc512620864)

[1.2 国内外相关领域发展趋势及现状 2](#_Toc512620865)

[1.2.1 中国票务系统现状 2](#_Toc512620866)

[1.2.2 国外票务系统现状 3](#_Toc512620867)

[1.3 论文研究内容和结构安排 4](#_Toc512620868)

[**第二章 关键技术介绍** 6](#_Toc512620869)

[2.1 QRcode 6](#_Toc512620870)

[2.2 Bootstrap 7](#_Toc512620871)

[2.3 jQuery 7](#_Toc512620872)

[2.4 WeUI 8](#_Toc512620873)

[2.5 Spring 8](#_Toc512620874)

[2.6 Mybatis 8](#_Toc512620875)

[2.7 Log4j 9](#_Toc512620876)

[2.8 Web Service 9](#_Toc512620877)

[2.8.1 Web Service简介 9](#_Toc512620878)

[2.8.2 Web Service的技术分析 10](#_Toc512620879)

[2.8.3 Web Service在本课题中的应用 12](#_Toc512620880)

[**第三章 软件架构理论与ADMEMS方法** 13](#_Toc512620881)

[3.1 软件架构的概念 13](#_Toc512620882)

[3.2 软件架构的产生背景 14](#_Toc512620883)

[3.3 软件架构的内容和作用 14](#_Toc512620884)

[3.4 ADMEMS软件架构设计方法 15](#_Toc512620885)

[**第四章 公园票务系统的软件架构设计** 18](#_Toc512620886)

[4.1 预架构阶段 18](#_Toc512620887)

[4.2 概念架构阶段 21](#_Toc512620888)

[4.3 细化架构阶段 25](#_Toc512620889)

[**第五章 公园票务系统的实现** 30](#_Toc512620890)

[5.1 系统的数据库设计 30](#_Toc512620891)

[5.1.1 逻辑结构设计 30](#_Toc512620892)

[5.1.2 数据库软件的选择 34](#_Toc512620893)

[5.2 游客子系统的实现 35](#_Toc512620894)

[5.3 后台管理子系统的实现 41](#_Toc512620895)

[5.4 验票子系统的实现 45](#_Toc512620896)

[5.4.1 硬件设备的介绍 46](#_Toc512620897)

[5.4.2 软件部分的实现 46](#_Toc512620898)

[**第六章 结束语** 49](#_Toc512620899)

[6.1 全文总结 49](#_Toc512620900)

[6.2 展望 49](#_Toc512620901)

[**参考文献** 51](#_Toc512620902)

[**致谢** 54](#_Toc512620903)

# 第一章 绪论

本章主要介绍了本课题的研究背景和意义，说明了目前在线票务系统的研究背景和应用意义，对国内外票务系统的现状分别进行了介绍，简洁地说明了论文的研究内容和结构安排。

## 1.1 研究背景及意义

如今，互联网浪潮使得世界上各行各业都在发生巨变，所有行业都在向科技化、自动化、智能化发展，信息化管理成为了包括政府、高校、企业等在内的主流趋势。随着互联网技术不断地融入人们的日常生活，建立公园票务系统可以方便游客去公园游玩、方便公园员工进行验票、方便公园管理层实时监测公园数据，并且结合售票数据进行有针对性的运营活动，提升公园的营业额和利润，具有一定的现实意义。

在线票务系统即使用条形码或二维码为通行的电子门票，综合多种技术，其中包括条形码或二维码的生成技术、无线传输技术、加密技术、计算机网络技术等等。使用这些技术实现门票通道的控制管理功能，例如互联网的查询、售票、检票、生成报表、活动配置、门票防伪等，同时也可以进行全方位管理和实时监控，对于使用在线票务系统的单位的经济效益有积极作用，与此同时也利于提高社会效率，是现代化管理水平的正向体现。在线票务系统现在已经应用在了各行各业，其中包括滑雪场、博物馆、游乐场、体育馆、高尔夫球场等旅游休闲相关的场所。还有音乐会、大型超市、演唱会、家政清洁等城市生活的场景[1]。

在线票务系统的应用意义如下，原本票务管理是人工模式，其中的各个环节比如售票、验票、统计数据都是由人工进行操作的，其中存在着一些难以解决的问题，并不能适应好现代化的管理要求，比如人工出错率高、其他人可以印制假门票、没有办法进行复查、无法及时地统计数据和分析数据、不够环保、对人力有一定程度浪费等方面的问题[2]。为了进一步丰富游玩服务，提高服务档次和服务水平，采用在线票务系统，结合先进的互联网技术手段，通过软件和硬件的结合，实现票务的互联网化[3]。采用在线票务系统有着积极的意义，具体包括以下几个方面：

（1）规范售票过程和检票过程，据了解国家旅游局曾做过一项抽样调查，最后通过统计调查数据，得出统计结果，假如检票过程采用人工检票，检票员难免会漏掉部分逃票人员或将自己的朋友、熟人无票放入，某种程度上会助长逃票这种恶劣行径。假如对某国家级大型景区进行估算，逃票率大概在1％，门票单价为88元，平均每日游览人数6000人，那么逃票金额就有193万元，对景区来说经济损失巨大。

（2）防止假票，有些票贩子利欲熏心，制造假票进行低价贩卖，据调查有不少旅游景区、主题乐园每日清点票据都出现过假票，由于验票时人流密集，时间匆忙，人工验票用肉眼很难辨别出假票，使用电子验票可以有效防止假票，降低经济损失。

（3）提高工作效率，提升管理水平。使用电子票务系统可以实时统计日客流量、月客流量、年客流量以及相应的门票收入，为科学管理提供依据。

（4）提高形象，采用电子验票系统，可以实现规范化管理。对提升使用方的形象，提高使用方的管理水平，促进使用方的精神文明建设及数字化建设有重大意义。

本文中的公园票务系统的实施主体是武汉市中山公园。在此次的公园票务系统投入使用之前，武汉市中山公园的众多游乐项目采取的是售卖纸质门票的方式，售票员收取现金，将纸质门票卖给游客，游客拿门票去游乐项目点，经过该游乐项目的验票员核对门票进行验证后方能游玩，整个买票验票过程效率十分低下，并且购买了纸质票后游客无法退票，每天的公园的记账和清点也十分复杂，而且这种人工处理的方式难免会发生疏漏导致错误，公园管理层想实时监测公园的营收额更是不可能做到的事情。除此之外，传统的纸质门票方式也无法实现会员的管理，不便于进行会员专属活动和促销活动。

公园票务系统这个项目的主要目标是实现武汉市中山公园内游乐项目购票的互联网化。游客在武汉市中山公园游玩时，可以通过手机自主办理会员卡，对会员卡进行充值并且获得一定的赠送金额，游客还可以在手机上查询会员余额、线上浏览中山公园的游乐项目后购买游乐项目的门票、在游乐项目验票员处使用会员卡直接扣款进行游玩、在游乐项目验票员处使用电子门票验票进行游玩、线上支付、对订单进行评价、对不想要的门票申请退款、查询消费明细、查询充值记录等等。与游客行为相对应的，就是公园方的验票需求和退票审核需求。除此之外，公园的管理人员还需要对公园的整体运营情况进行宏观的把控，主要想通过系统实时监测公园的财务报表、会员人数、充值金额、零售金额等。对于中山公园来说，在线票务系统可以实现门票智能化、门票数字化的管理，使整个公园的管理和运营更加方便、更加规范、更加人性化[4]。信息化的管理对于公园领导层进行科学的决策有着难以忽视的重要意义，在线票务系统有助于推动中山公园的票务整体水平发展到一个高的层次，同时促进其经济收益和形象提升。

## 1.2 国内外相关领域发展趋势及现状

### 1.2.1 中国票务系统现状

电子票务在现在看来是属于电子商务领域的，在我国的起源时间是二十世纪九十年代。目前随着互联网技术的发展，票务系统不断进行迭代更新，已经可以基本保证信息的准确性、购票验票的及时性。由于互联网票务系统的多重优点，比如规范了售票和检票的过程、实时查询准确数据、节省人力物力等，使得在线票务必将占据票务趋势的主流[5]。

票务系统在我国的应用涉及到了各种不同行业领域，包括体育赛事、公共交通、旅游行业、文化活动、公共事业等等，票务系统相关的应用产业监管就跟不止一个行政部门有关。国家工业与信息化部门对票务系统软件和硬件的开发与生产进行监督管理；对于票务系统的应用和运作，由相应的监管部门进行管理，并与其它部门进行协调监管。如今旅游景区的门票销售有一定比例依旧是用传统的人工贩售纸质票为主[6]，传统的纸质票务在实际应用中存在很多不便之处，对使用和管理有负面影响，同时互联网票务管理系统有了一定的发展，会随着时间的推移取代传统方式的人工贩售纸质票务。

我国现在还是处在互联网票务系统和传统票证并行存在的时候，并且在某些场景下传统票证占的份额更大一些。但是目前已经可以预见，票务信息化、互联网化趋势是不可逆转的，票务系统终将逐步现代化。如今，网络带宽在不断提高，二维码或条形码的制作已经成熟，识别非常准确、容错性好。互联网票务系统中的票务管理规范且高效，同时票务自动化可以节省人力物力，实时准确的数据存储能力可以方便管理。总而言之，互联网票务系统最终将应用于相关的各个领域，传统的纸质票务已经走向衰落，会渐渐被替代[7]。互联网对票务工作的正面影响不容小觑，总体来说前景非常乐观。

### 1.2.2 国外票务系统现状

国外票务系统的现状，我们重点研究美国市场，美国市场的娱乐演出等产业相比国内更加发达，美国的在线票务系统发源较早，技术上逐步成熟，模式上比较先进，值得我们进行参考与借鉴。国外的票务系统与国内的票务系统相比，有一些不同，比如国外的票务系统更加多样化。对于中国来说，目前做得比较大的票务网站有大麦网、淘票票、猫眼购票、永乐票务等。这些票务网站基本都是从演唱会、电影等业务起家，然后向其他相关领域横向发展，并且是“商家对零售”的模式。下面，我们将选取三家比较有代表性的票务网站进行介绍，它们依次是票务大师（Ticketmaster）、史达波票务（Stubhub）和Eventbrite。

票务大师是一家主营在线零售的票务网站，建立时间比较久，主要采取的也是商家对零售的模式，卖票的形式比较丰富，其中包括线上订票、电话订票、零售网点订票等。票务大师的业务范围很广，可以说是几乎覆盖到了所有形式的商业演出以及赛事。现在，票务大师在全球已经有了接近二十个呼叫中心，接近七万个零售网点[8]，它的售票业务已经覆盖了接近三十个国家，票务大师的用户已经遍布全球。票务大师网站会与活动的主办方展开深度合作，所以在票务大师网站上会有一些带福利性质的特殊位置，还有提前预定可以享受到的特殊优惠。票务大师上的选座方式有两种，一种是系统分配座位；另外一种是交互式选座，与国内目前的在线选座比较类似。

史达波是票务转卖网站，有票需要出售的人可以在该网站上挂上详细的出售信息，有需要的人可以根据自己的需要购票，当然这些服务都不是免费提供的，卖票的用户需要给网站一定比例的酬金，根据票的类型酬金比例会有浮动，但是基本在一成左右，买票的用户同样需要支付佣金，佣金比例不会超过在一成到两成之间。史达波旗下还有一家主营体育赛事票务的网站，此网站与体育赛事主办方有深度合作，同时也与多家篮球俱乐部建立联系，是官方售票的代理商。

在Eventbrite网站上，用户可以自行在网站上创建售票活动，具体步骤包含新建活动页面、填写活动详情和售票信息，然后对此活动进行全方位的宣传。是个人对个人的售票模式，针对不同种类的卖票需求，Eventbrite网站都有完善的解决方案。

除了以上这些多样化的售票方式，国外的取票方式也很丰富，目前已经普及了用户打印电子票据的方式。国外票务网站的购票体验总体来说比较好[9]，不仅在购票前给出了座位图，也会对例如过道票等特殊位置的票做出文字提醒，让用户有更好的选票体验。国外的票务网站种类繁多，但是各个网站之间又有不同之处，都通过自己的特色建立了竞争壁垒，通过自己的核心竞争力吸引活动主办方和购票人群。

## 1.3 论文研究内容和结构安排

软件架构用来抽象地指导设计系统的整个过程。由于软件架构能带来很多好处，比如能够提前识别和减轻风险；保证方法和标准的一致性，随之产生结构良好的代码库；保证团队技术更好的协调。本文选择了软件架构驱动的方式对公园票务系统进行设计和实现。本文针对公园票务系统的设计和实现这一问题，介绍了软件架构的概念和重要性，详细介绍了ADMEMS（Architecture Design Method has been Extended to Method System）方法及其优点，选择了此软件架构方法对公园票务系统进行架构设计。本文还介绍了使用到的关键技术，最后完成对公园票务系统的实现。论文结构如下：

第一章是绪论，对本课题的研究背景和意义做了介绍，简单地介绍了在线票务系统的使用方式和应用意义，对国内外票务系统的现状分别进行了介绍，简洁地说明了论文的研究内容和结构安排。

第二章是关键技术介绍，介绍了此次公园票务系统使用到的关键技术，包括QRcode技术、Bootstrap框架、WeUI样式库、Mybatis等等，重点对Web Service技术进行了介绍，本系统通过Web Service技术，将开发工作分成服务开发与应用开发，使公园票务系统具有跨编程语言、跨运行平台的特点，很好地在系统中实现了数据共享和信息流通。

第三章是软件架构介绍，详细介绍了软件架构的概念、软件架构的产生背景、软件架构的内容和作用。最后介绍了一种经典的软件架构设计方法——ADMEMS，详细说明了此方法的三个阶段和一个贯穿原则，介绍使用此方法的优势，为公园票务系统的软件架构过程提供指导。

第四章是公园票务系统的软件架构设计，按照ADMEMS方法的步骤，依次从预架构阶段、概念架构阶段、细化架构阶段，使用了多种工具，完成了公园票务系统的架构设计。

第五章是公园票务系统的实现，其中包括系统的数据库设计、游客子系统的实现、后台管理子系统的实现、验票子系统的实现。三个子系统共同组成了公园票务系统，满足了各方需求。

第六章是结束语，进行了总结和展望。全文总结部分对软件架构驱动此公园票务系统的设计与实现的过程进行了总结，展望部分对下一步可以进行的工作进行了合理展望，主要包括将此公园票务系统全国化、增加亲情会员卡功能、无感支付、对收集到的信息进行数据挖掘进行个性化推荐。

# 第二章 关键技术介绍

本章是关于相关技术的介绍，主要给出了此次公园票务系统使用到的主要技术，包括QRcode技术、Bootstrap框架、WeUI样式库、Mybatis等等，重点对Web Service技术进行了介绍，本系统通过Web Service技术，将开发工作分成服务开发与应用开发，使公园票务系统具有跨编程语言、跨运行平台的特点，很好地在系统中实现了数据共享和信息流通。

## 2.1 QRcode

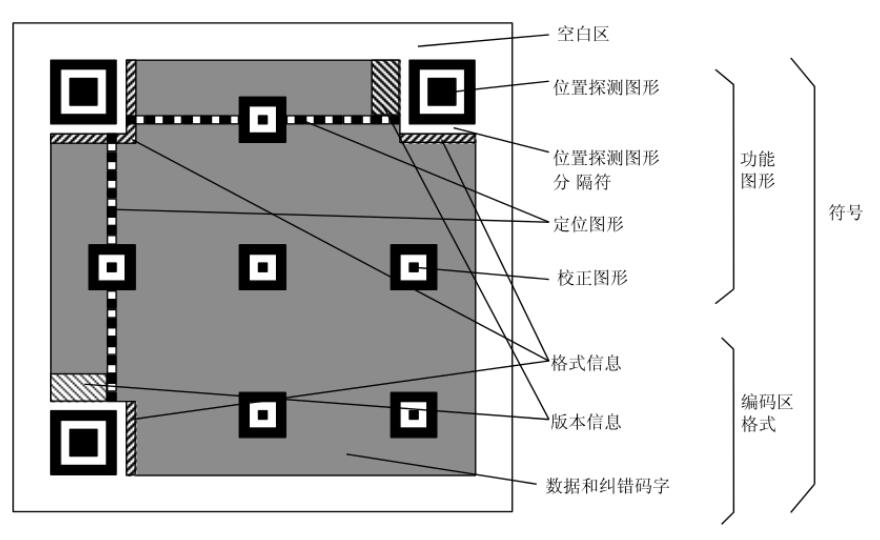
 QRCode是Quick Response Code的缩写，中文名是二维码，它由Denso公司研制出来的一种矩阵二维码符号[10]，通过特定的几何图案在二维平面上按特定规律分布来记录数据多种类型的数据信息，一般是黑白相间的。二维码的基本结构如图2.1所示：

图2.1 二维码的基本结构

其中二维码使用定位图形、位置探测图形、位置探测图形分隔符的功能来进行定位，对二维码来说，大小规格可能会有所不同[11]，但是位置是固定的；对于校正图形来说二维码规格确定后校正图形的位置和数量也确定了；格式信息用来表示二维码的纠错级别，有四个等级，由高到低分别是H级别、Q级别、M级别、L级别；版本信息即二维码的规格，二维码总共有40种规格矩阵；数据即实际保存的二维码信息；纠错码字用于修正二维码损坏或脏污带来的错误。

二维码的特点有：二维码能够为所有类型的数据提供支持，包括字母、数字、汉字、符号、二进制等；存储信息的容量大，1个二维码在仅存储数字时能够处理7089字的信息量；所占空间小，因为二维码使用横向和纵向两个方向处理数据，在信息量一样的情况下，所占空间小；具备纠错功能，部分编码破损、脏污能够恢复数据，最多可以纠错约28%，对破损和脏污的适应力强[12]；不限制读取方向，二维码不受背景样式和读取方向的影响，可以360度实现准确快速的读取[13]。

QRCode现在的应用十分广泛，已经成为了日常生活中必不可少的工具，被方便地应用在名片、宣传海报、电影票、团购优惠券等现实场景，给人们的生活和工作提供了便利。

在本次项目中，我们使用了QRCode.js这样一个用于制作QRCode的JavaScript库来进行动态的生成不同的二维码。QRCode.js具有很好的浏览器兼容性。

## 2.2 Bootstrap

Bootstrap是一套前端组件库，使用了CSS语言、HTML语言、JavaScript语言，由Twitter推出，用于开发响应式布局、移动设备优先的 WEB 项目。Bootstrap是基于HTML5和CSS3开发的，Bootstrap包是由基本结构、CSS、组件、JavaScript插件组成的[15]。使用这个组件库我们可以根据自己的需求快速地搭建出功能完备、界面美观的网站。可以说 Bootstrap 是一个简洁好用的前端开发框架，使得Web开发更简单更迅速。Bootstrap的优势有响应式开发，让一个网站可以兼容不同分辨率的设备；界面美观，给用户带来更好的用户体验；提供了丰富的组件可供选择[16]。现在许多平台默认引用Bootstrap的包，可见是一种未来的技术趋势。

## 2.3 jQuery

jQuery是一个JavaScript工具库，封装JavaScript常用的功能代码，提供一种简单方便的JavaScript设计模式，得到一整套定义好的方法。随其不断的发展jQuery已经集成了JavaScript、DOM、CSS、Ajax于一体的强大功能。jQuery的设计理念是用最少的代码量完成最多、最复杂的功能[17]。jQuery降低了跨浏览器的Web开发的难度，它在以下几方面表现出色：通过多种内建的方法，更方便快捷地迭代和遍历DOM；提供内置的、通用的、高级的使用选择器的功能，使用它在DOM中选择条目更为简单[18]；提供易于理解的插件架构，可以方便地添加自定义方法；有助于减少UI功能和导航的冗余，如选项卡、基于标记的弹出式对话框、过渡等[19]；能够兼容目前的多种主流浏览器。

## 2.4 WeUI

WeUI是一套基础样式库，这套样式库是由微信官方视觉团队所设计的，主要用于微信公众号、服务号、内网页等场景，使用WeUI可以保证公园票务系统中的游客子系统与微信原生的视觉效果一致，提升用户体验，使视觉感知统一。

## 2.5 Spring

Spring是一个开源的框架，由罗德•琼森创建，是轻量级的Java开源框架，它的核心思想是面向接口编程，达到的目的是业务逻辑层与系统的其它层松耦合[21]。Spring的主要优势在于分层架构，允许开发工程师去选择并使用某个具体的组件，将开发企业级应用程序的复杂过程很好地解决了。Spring框架有几个特征：对于开销和大小，它都是轻量的；它是非侵入式的，Spring应用中的对象并不依赖Spring的特定类；它允许通过分离系统级服务和业务逻辑开展内聚性地开发，提供了面向切面编程的巨大支持；它能够将简单的组件配置组合成繁杂的应用[22]。Spring的以上特征让开发工程师编写更简洁、更易管理的代码成为可能，同时也更易于测试工程师的工作。

## 2.6 Mybatis

MyBatis是支持自定义SQL、存储过程和高级映射的第一类持久性框架。MyBatis消除了几乎所有的JDBC代码以及手动设置参数和检索结果。MyBatis可以使用简单的XML或Annotations来配置和映射原语，Map接口和Java POJO（普通旧Java对象）到数据库记录。它的目标是让工程师在执行SQL语句的时候，对输入的数据和输出的数据进行方便、高效的管理，它的核心优势在于便捷地写出SQL语句以及便捷地获取SQL的执行结果[23]。它绕过了绝大多数的JDBC代码、手动设置参数和获取结果集。它能够通过简单的XML或者注释，配置和映射原生信息，将普通的Java对象和接口映射成数据库中的记录[24]。它的功能架构有以下三层：API接口层，一旦接到调用请求，调用数据处理层进行数据的处理，给外部使用的接口提供API以供工程师通过本地API对数据库进行操作；数据处理层，主要目标是依据调用的请求来完成数据库的操作，具体来说就是负责SQL查询、SQL解析、SQL执行和对执行结果进行映射处理[25]；基础支撑层，负责连接管理、配置加载、事务管理、缓存处理等等基础功能的支撑，将这些共用部分抽取出，成为最基础的组件，给数据处理层来提供最基础的支持。此次项目中使用MyBatis来进行数据库封装。

## 2.7 Log4j

Log4j的含义是Java日志，它是Apache的开源项目，作为一个日志操作包，它可以用来控制日志信息的输送位置，例如UNIX Syslog守护进程、文件、控制台、接口服务器、用户图像界面组件等等。它的三个主要组件分别是Layout、Logger、Appender，通过它们的相互配合协作，强大的日志记录能力成为了现实[26]。我们可以使用定义日志信息级别的方法，在细节上控制日志产生的过程，我们甚至能够对日志的输出格式进行控制。我们想要达到这些目的，并没有必要去对相关的代码进行修改，仅仅配置一个文件就可以实现。Log4j不仅仅能在Java程序中使用，它有其他的编程语言接口，所以也能在诸如C++、.Net、PHP程序中使用，具体的使用方法和语法没有区别[27-28]，最终使多种编程语言分布式系统有统一的日志组件模块，通过第三方扩展的方法可以将它集成到JINI、J2EE、SNMP应用。

## 2.8 Web Service

Web Service直接地翻译成中文意思就是网络服务，这项技术可以不用借助专门的、其他的第三方硬件或者软件，让在不同机器上运行的应用直接实现互相交换数据或者集成。也就是说，使用了Web Service的应用，不用去考虑开发应用时使用的编程语言、应用运行的平台或者内部协议的类型[29]，应用之间都能够互相地进行数据交换。简单地总结就是Web Service（网络服务）的本质是通过网络调用其他网站的资源。

### 2.8.1 Web Service简介

如今互联网在加速迅猛发展，已经成为了全球信息传递和共享的巨大的资源库。如今建立了愈来愈多的网络环境下的Web应用系统，使用各种Web技术使得在互联网环境下实现实时聊天、在线购物等多种应用成为可能。根据实际情况，这些Web应用系统很有可能分布于不同的地理位置、基于不同的操作系统平台、使用着不同的数据组织形式，如果能让这些数据资源集中起来[30]，并且加以利用，是件极有价值的事情。随着网络理念和技术的发展，Web Service（网络服务）应运而生，它可以看作是一种解决方案，利用网络进行应用集成。从简单的请求到繁杂的处理，网络服务能执行很多功能，它是Web应用程序的分支，完成部署后，另外的网络服务应用程序就能发现且去调用它部署的服务。由此可见，网络服务是构造分布式、模块化应用程序和面向服务应用集成的重要技术和今后的发展趋势[31]。

一般来说，服务就是由计算机后台程序（Daemon）所提供的功能。比如，让一个杀毒软件在后台运行，它会自动监控计算机系统，那么这种自动监控可以被称为“服务”。“服务”可以根据不同的来源分为“本地服务”和“网络服务”。本地服务不需要网络，是由同一台计算机提供的服务。网络服务必须要通过网络才可以完成，是使用另外一台计算机提供的服务。举个例子，现在有一张生活照需要进行压缩处理，这里的“服务”就是“压缩照片”。如果要使用本地服务，我们就可以在自己的电脑上使用Photoshop软件对生活照进行压缩处理，然后保存。如果要使用网络服务，我们就可以将图片上传到专门的压缩网站，网站的服务器会自动帮我们对生活照进行压缩处理，完成压缩后通过网络回传给我们。我写过一个“武汉大学美景墙”的项目，它可以动态显示关于武汉大学美景的最新照片，然而这些美丽的景色照片并没有存储在我的服务器，而是由flickr.com提供，我只是向flickr.com发出一个向我提供照片的动态请求。这就是由flickr.com提供的一种网络服务。如果我把武汉大学的美景照片存储在我本地的服务器，不调用flickr.com，这种情况就是在使用本地服务。也就是说，Web service让我的网站使用其他网站的资源成为了可能，比如在网页上使用其他网站的地图资源、天气资源等等。

简单的说，Web Service是跨操作系统平台、跨编程语言的一种远程调用技术[32]。跨编程语言是指服务端程序如果采用Java语言来编写，客户端程序使用Swift语言来编写，它们之间可以直接实现互相交换数据。远程调用，就是某台计算机上的应用能够调用其他计算机上的应用。例如：微信钱包并没有银行的信用卡账单等数据，但是它可以调用银行提供的相关接口去获取信用卡账单数据，让用户在微信界面中查看自己的信用卡欠款情况。墨迹天气APP也并不会自己去做实时的气象监测，气象局将相关的系统服务用网络服务的形式暴露出来，墨迹天气APP就可以调用这些服务进行天气预报了。

Web Service具有以下四种优势：

（1）跨编程语言，只要遵守Web Service的相关协议，就能使用Java、PHP等任意种类的编程语言，向其他网站要求网络服务。很大程度上增加了网络服务的适用性，降低了对软件开发工程师的要求。

（2）跨操作系统平台，无论使用哪种操作系统平台，都能够使用Web Service。

（3）对于网络服务的提供者而言，部署、升级和维护网络服务不用去考虑客户端的兼容问题，将这些过程简单化[33]。

（4）对于网络服务的使用者而言，能够比较简单地实现多种服务、多种数据的聚合，增加了开发过程的想象力。

### 2.8.2 Web Service的技术分析

构成网络服务平台的三大技术是XML+XSD,SOAP协议和WSDL技术。Web Service可以看作是互联网上的某种服务协议，采用XML（可扩展的标记语言）作为基本的标记语言，SOAP（简单对象访问协议）作为互操作协议，WSDL（网络服务描述语言）作为服务的描述语言，通过UDDI（统一描述、发现和集成服务）可以把服务注册到因特网便于搜索，让更多的用户能使用其服务。Web Service 结构如图2.2所示：



图 2.2 Web Service结构图

（1）XML+XSD

XML的中文意思是可扩展的标记语言，属于一个标准通用标记语言之下的子集，它是一种表示数据的基本格式，用于网络服务平台中。可扩展的标记语言有诸多优点，其中包括易于分析、易于建立、与厂商无关、跨平台。万维网协会创建了可扩展标记语言，它通过制定XML SchemaXSD来定义一套标准的数据类型，并发布了一种语言用来将这套数据类型进行扩展。XML被用来在网络服务中封装数据。

（2）SOAP协议

SOAP的全称是Simple Object Access Protocol，中文意思是简单对象访问协议，它是轻量级协议，被用来交换可扩展标记语言的编码信息。在网络服务采用XML格式封装发送的请求内容和结果内容时，增加了SOAP协议规定的特定的HTTP消息头，作用是说明HTTP消息的内容格式。

（3）WDSL语言

WSDL是人和机器都可以阅读的一种语言，用机器能阅读的方式提供出的一个正规描述文档，类似说明书的作用，描述网络服务和相关的函数、参数、返回值。通过WSDL文件，Web Service服务器端可以对供外部调用的服务进行说明。WSDL文件保存于Web服务器，经过URL地址能够对它进行访问。客户端找到某个服务的WSDL文件地址，才能调用该网络服务。

（4）UDDI服务注册

UDDI的全称是Universal Description, Discovery, and Integration。中文意思是统一描述、发现和集成，由微软等30多家公司创立。UDDI提供了一种编程模式和模型，UDDI规范中用XML来定义所有的API，包装于SOAP信封，在HTTP上传输时，采取了SOAP请求和SOAP响应的模型。

### 2.8.3 Web Service在本课题中的应用

此公园票务系统通过网络服务技术，将整个开发过程分隔开，使开发工作分成服务开发与应用开发两个部分，提高了开发团队的工作效率，简化测试工作。此项目利用了网络服务技术跨编程语言、跨运行平台的特性，结合项目中具体的功能性需求和非功能性需求，很好地在系统中实现了数据共享和信息流通。

# 第三章 软件架构理论与ADMEMS方法

本章是软件架构介绍，详细介绍了软件架构的概念、产生背景、内容和作用。最后介绍了一种经典的软件架构设计方法——ADMEMS，详细说明了此方法的三个阶段和一个贯穿原则，介绍此方法的优点，为公园票务系统的软件架构过程提供指导。

## 3.1 软件架构的概念

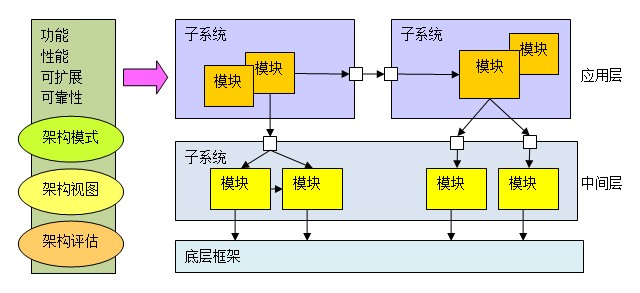
 软件架构基于特定的设计原则，对组成系统的部分从各个方面进行合理的设计，其中包含了系统的模块、模块与模块间的关系、模块的属性等，形成了系统的各类结构进而成为了架构[34]。从整体来看，软件架构设计展示了软件系统的特性和组成，对系统进行了有层次的多层决策。软件架构简单来说就是系统的草图，用来抽象地指导设计系统的整个过程，并且对构成系统的抽象的模块进行描述，同时也描述了模块与模块间的通信。在软件实现的阶段，抽象的模块就成为了真实的模块[35]，例如某个对象或者某个类。软件架构示意图如图3.1所示：

图3.1 软件架构示意图

大卫·戈尔仑和玛丽·歇尔对软件架构的看法如下，软件架构问题某种程度上是设计系统的整体结构，其中包括系统的全局控制、数据之间的访问使用什么协议、系统的性能达到哪种标准、用户界面的风格如何、模块之间怎样实现交互、对外有哪些接口、有哪些创新的突破等等。架构绝不仅仅只是结构[36]，架构是一种对系统而言最高层的概念。软件架构对系统的内部和外部都有考虑。既考虑系统内部，又对系统的开发平台及用户使用场景有整体的考虑[37]。软件系统架构的要素有：系统架构对软件系统做了最高层面的设计；模块构成了系统，对于系统的结构而言，模块的形成和互相之间的作用是与系统结构密切相关的重要内容。

## 3.2 软件架构的产生背景

软件架构的产生背景是软件危机的出现。在1960年之前，计算机技术处于起步阶段，软件的规模也很小，很大程度上依赖于汇编语言或者机器代码，没有文档资料，更没有系统化的开发方法。在1965年左右，计算机的性能有了大的提升，计算机的容量越来越大，处理速度也越来越快，这时高级语言出现，操作系统也有了新的发展，计算机的应用方式有了积极的变化导致软件开发急剧增长。在这种情况下软件的复杂度和软件系统的规模都在增长，导致软件的可靠性问题突出。原本的开发方式无法满足当前的要求了，软件的开发方式迫切的需要被改变。软件危机就在这种情况下爆发了[38]。

1967年一些计算机领域的科学家意识到了软件危机，他们聚集在一起召开了会议对软件危机进行讨论，会议为了研究和解决软件危机问题正式提出了“软件工程”，即Software Engineering。软件工程学成为了一门有实际用途的工程学科。软件工程可以研究软件生产过程的客观规律，建立与软件系统有关的原则、相关概念、方法、相关的技术和用到的工具，并且对系统的生产活动进行指导和支持，最终达到的目标有增加软件生产率、改进软件的质量、降低整个过程的生产成本[39]。软件工程学发展了软件开发的一些技术，发明了软件维护相关的科学方法，从其他的工程学科（例如硬件工程学科）中吸收先进经验，应用在了软件工程的实践过程并有很好的效果。

后来结构化设计方法作为软件工程的重点组成部分也有了快速的发展。其中的软件模块化技术在软件架构理论中比较早被提到，这项技术应用在解决复杂问题时，通过自顶向下的顺序将软件系统一层一层地划分为模块。那时候的一些架构设计原则，比如“高内聚”、“低耦合”、“提高开放性”等是目前仍然在使用的经典基本架构技术，尤其适用于开发复杂的软件系统。软件架构就从结构化设计技术起源。

## 3.3 软件架构的内容和作用

在了解软件架构之前，需要先了解应用程序架构和系统架构。对于软件开发者来说，最熟悉的应该是应用程序架构，特别是通常由单一技术编写的应用程序，比如Java网络应用程序、Windows桌面应用程序等。应用程序架构的关注点是应用程序，通常包括将应用程序解构为类和组件，确保设计模式的正确使用，构建或使用框架等等。本质上，应用程序架构谈论的是软件设计的低级别切面，通常只考虑单一的技术栈。其中结构单元主要以软件为基础，包括编程语言和结构、类库、框架、API等，它由类、组件、模块、函数、设计模式等加以描述。应用程序架构着重考虑软件和代码组织。

系统架构可以看做是更大规模的应用程序架构。大多数软件系统实际上是由横跨不同的层次和技术的多个应用程序组成。举个例子，你可能有一个软件系统，Java EE中间层使用Oracle数据库提供的数据，同时向.NET Silverlight客户端提供Web服务。每个部分都有自己的应用程序架构。要让整个软件系统工作起来，就要思考如何组合这些单独的应用程序。要有端到端软件系统在较高层次上的整体结构。另外，大多数软件系统都不是孤立的，因此系统架构还关注互操作性和与环境中其他系统的集成。结构单元就是各种软硬件，从编程语言和软件框架到服务器和基础设施。跟应用程序架构相比，系统架构描述为从组件和服务到子系统等更高层次的抽象。系统架构的定义大多数都包括了软件和硬件。毕竟，一个成功的软件系统离不开硬件，即使是云上的虚拟硬件。

应用程序和系统架构相对较容易理解，但人们对“软件架构”一词的理解不尽相同。本文将软件架构定义得尽可能简单，不去受制于各种定义的复杂性和细微差别。软件架构就是应用程序架构和系统架构的结合。换句话说，从代码结构和基础到将代码成功部署到生产环境，与一个软件系统重要元素相关的所有东西就是软件架构[40]。从开发者的角度考虑软件开发，关注点会多放在代码上。本文考虑的是有助于构架更好软件的东西，比如面向对象的原则、类、接口、控制反转、重构、自动化单元测试、代码整洁和其他不胜枚举的技术实践。

架构反映了使一个系统成型的重要设计决策，而重要性则通过改变的成本来衡量。在现实世界中，架构和设计的区别并不明显，但该定义为我们提供了一个基准，架构可能包括：系统的形态（例如，客户端—服务器、基于Web、原生移动客户端、分布式、异步），软件系统的结构（例如，组件、层、交互），技术选择（即编程语言、部署平台），框架选择（例如，Web MVC框架、持久性/ORM框架等），设计方法/模式选择(例如，针对性能、可伸缩性、可用性等的方法)。

软件架构能带来很多好处，比如让团队跟随一个清晰的愿景和路线图，无论这个愿景是一人所有还是整个团队共有；实现技术领导力和更好的协调；与团队内外部人员交流的重要内容，以便回答与重要决策、非功能需求、限制和其他横切关注点相关的问题；能够提前识别和减轻风险的框架；保证了方法和标准的一致性，随之产生结构良好的代码库；对于不同的听众，有以不同层次的抽象来交流解决方案的结构。

软件架构是在建造系统之前做好的工作，其中包含了很多重要决定，当系统开始详细设计和实现的时候，软件架构的部分就很难修正了，有时甚至没有办法修正，所以软件架构的过程一定要慎重充分考虑，是影响系统设计好坏的重大影响因素。

## 3.4 ADMEMS软件架构设计方法

2010年在我国召开了第六届软件大会，会上公布了经典的软件架构设计方法ADMEMS（Architecture Design Method has been Extended to Method System），此方法由著名的CSAI顾问团架构设计专家组公开发布。ADMEMS对比其他的软件架构设计方法有以下几个优点：

（1）本方法体系是从一线权威架构设计的实践过程中总结提取的[41]，结合了各种规模的项目实际情况，是一种实用的方法体系，具有很强的实践指导意义。

（2）本方法体系在其三大阶段中，使用了多种思考工具和表达工具，将方法与工具结合，进行高效率的优良架构设计。

（3）本方法体系阶段划分清晰，每个阶段需要的准备工作和阶段性目标非常明确，对初学者很友好，是很适合初级软件架构师入门的方法。

通过ADMEMS可以达到“输入需求，输出架构”，ADMEMS由以下3个阶段组成，如图3.2所示：



图3.2 ADMEMS方法流程图

第一个阶段是预架构阶段，也叫作PA（Pre-Architecture）阶段。此阶段是架构设计的最前期阶段，要做到对需求全面理解后使需求结构化，同时建立二维需求观，将需求间的关系弄清楚并且寻找衍生需求，所做的工作包括与客户进行深入的需求沟通，阅读该项目的需求类文档《需求规格说明书》，与项目经理进行探讨。预架构阶段的目标是通过对需求的全面理解以把握其特点，做到对整体架构设计的驱动。预架构阶段的步骤是结构化需求，了解约束影响，确定软件的质量和关键功能。预架构阶段的目标是理解需求并从整体上规划需求，确认软件架构设计的方向。预架构阶段的意义是便于从全局来把握软件需求，使软件架构设计尽早开始并且减少失败的风险，做到对整体架构设计的驱动。

第二个阶段是概念架构阶段，也叫作CA（Conceptual Architecture）阶段。此阶段是针对重大的需求、高风险需求、特色需求确定概念架构，形成稳定的高层软件结构设计结果，考虑质量、功能、约束各个方面的需求给出解决方案。所做的工作包括对项目的可行性进行讨论和分析，从软件需求、技术风险、人力风险等方面进行思考。此阶段的目标是通过业务需求、重点功能、关键质量和约束来塑造概念架构。概念架构阶段由如下三个步骤组成。

（1）初步设计：初步设计的工作是基于系统的关键功能，采用鲁棒图，使用“职责协作链”去确定职责。鲁棒图是使用在需求设计过程中的方法，此方法使用在需求设计前、需求分析后，目的是让设计人员全面地、清晰地了解需求进行软件架构分析，它关注的重点是功能需求的分析工作，向设计方案进行。

（2）高层分割：高层分割是在上一步初步设计的过程后，对系统进行划分，整个划分过程将系统看作黑盒。通常情况下对于关系众多、功能复杂的系统划分的方式有以下两种：将庞大且复杂的系统划分成为几个二级系统，将庞大且复杂的系统划分为具体子系统。分层的角度不同，可以有以下几种流派：逻辑分层（Layer）、物理分层（Tier）、按通用性进行分层、技术堆叠。对于逻辑分层而言，职责的划分被看重，职责与职责间的关系往往是上层调用下层，逻辑分层对于上层和下层能否分布于不同的机器并不重视。物理分层是指可以分布于不同机器的软件层，层与层间必定要有使用通信协议或者进行远程调用的能力，简洁地说，也就是层与层直接有跨机器进行访问的能力。按通用性进行分层强调“通用性”，严格来说算是逻辑分层的一种子类，根据通用性的不同划分不同的层，通用程度大则所处的层越靠下，举例来说比如应用层、操作系统层、硬件驱动程序层。技术堆叠依赖于其他的分层方式，综合了多种分层方式，为层的技术实现提供深层的说明。

（3）考虑非功能需求，概念架构阶段并不是简单的进行理想化架构，除了考虑客户提出的功能性需求外，也必须要考虑到决定项目质量的非功能需求。在ADMEMS软件架构方法中，通常采用“目标—场景—决策”表来进行这一过程。

第三个阶段是细化架构阶段，也叫作RA（Refined Architecture）阶段。此阶段主要通过5视图法来关注不同的视图，涉及物理架构、逻辑架构、数据架构、开发架构、运行架构，对细化架构阶段进行实践。物理架构包括对系统的软件和硬件要求，对结点的物理部署设计；逻辑架构包括对系统的分层、分区、框架提取，根据模块间的协作关系来确定模块接口；数据架构指的是对数据库的设计；开发架构指的是对开发的目录结构进行规划，确定project的数量；运行架构主要是对线程和进程的设计[42]。所做的工作有在整个软件项目的概要设计阶段，进行架构设计并制定约束与规范，主要关注对子系统的划分和接口的定义等，为后面的详细设计提供指导。

ADMEMS软件架构设计方法除了以上三个阶段以外，还有一个贯穿环节，即对非功能需求进行从始至终的关注，也就是说要贯穿软件架构设计的整个过程。当然，在具体的项目中软件架构设计的过程和内容的并非完全要与理论一致，根据项目的具体要求，有些环节是可以省略的，在进行软件架构设计的过程中要灵活运用[43]。完成了以上所有的阶段，软件架构的部分就完成了，然后就可以开始对系统进行详细设计和编码实现了。

# 第四章 公园票务系统的软件架构设计

本章是公园票务系统的软件架构设计，按照ADMEMS方法的步骤，依次从预架构阶段、概念架构阶段、细化架构阶段，使用了多种思考工具和表达工具，完成了公园票务系统的架构设计。ADMEMS是一种经典的软件架构设计方法，具有的三大优点：从一线权威架构设计的实践过程中总结提取，结合了各种规模的项目实际情况，是一种实用的方法体系，具有很强的实践指导意义；使用了多种思考工具和表达工具，将方法与工具结合，进行高效率的优良架构设计；阶段划分清晰，每个阶段需要的准备工作和阶段性目标非常明确，对初学者很友好，是很适合初级软件架构师入门。下面，我们将使用ADMEMS方法在这一章对公园票务系统完成“输入需求，输出架构”的过程。

## 4.1 预架构阶段

预架构阶段（PA阶段）对需求进行全面理解后使需求结构化，同时建立二维需求观，将需求间的关系弄清楚并且寻找衍生需求[44]，预架构阶段的目标是通过对需求的全面理解以把握其特点，做到对整体架构设计的驱动。下面将结合此公园票务系统的具体需求来进行预架构阶段。

武汉市中山公园是本公园票务系统的实施主体。在此次的公园票务系统投入使用之前，武汉市中山公园的众多游乐项目采取的是售卖纸质门票的方式，售票员收取现金，将纸质门票卖给游客，游客拿门票去游乐项目点，经过该游乐项目的验票员核对门票进行验证后方能游玩，整个买票验票过程效率十分低下，并且购买了纸质票后游客无法退票，每天的公园的记账和清点也十分复杂，而且这种人工处理的方式难免会发生疏漏导致错误，公园管理层想实时监测公园的营收额更是不可能做到的事情。除此之外，传统的纸质门票方式也无法实现会员的管理，不便于进行会员专属活动和促销活动。

公园票务系统这个项目的主要目标是实现武汉市中山公园内游乐项目购票的互联网化。游客在武汉市中山公园游玩时，可以通过手机自主办理会员卡，对会员卡进行充值并且获得一定的赠送金额，游客还可以在手机上查询会员余额、线上浏览中山公园的游乐项目后购买游乐项目的门票、在游乐项目验票员处使用会员卡直接扣款进行游玩、在游乐项目验票员处使用电子门票验票进行游玩、线上支付、对订单进行评价、对不想要的门票申请退款、查询消费明细、查询充值记录等等。与游客行为相对应的，就是公园方的验票需求和退票审核需求。除此之外，公园的管理人员还需要对公园的整体运营情况进行宏观的把控，主要想通过系统实时监测公园的财务报表、会员人数、充值金额、零售金额等。

下面，我们通过表格的形式对需求进行全面、整体的把握，功能性需求见表4-1：

表4-1 公园票务系统的功能性需求

|  |  |  |  |
| --- | --- | --- | --- |
| 功能性需求 | 说明 | 衍生需求 | 说明 |
| 注册/登录 | 公园游客注册账号登录系统进行操作 |  |  |
| 会员卡办理 | 公园游客可以在线办理此公园的会员卡 | 个人中心 | 个人中心存储会员的所有信息 |
| 会员卡充值 | 公园游客进行充值后可以使用余额进行购票 | 支付 | 所有涉及到钱款的行为都需要支付 |
| 会员卡扣款游玩 | 公园游客在游乐项目验票员处使用会员卡直接扣款进行游玩 |  |  |
| 浏览游乐项目 | 公园游客可以在手机上浏览公园的所有游乐项目及价格、详情、注意事项等 | 项目介绍 | 游乐项目的详细介绍 |
| 浏览项目购买门票 | 公园游客可以在手机上购买游乐项目的门票 | 购物车 | 公园游客购买游乐项目门票时可以任意搭配种类和数量 |
| 验证门票 | 游乐项目处的管理员验证电子门票是否合格 |  |  |
| 评价订单 | 公园游客结束游玩后可以对购买的游乐项目进行评价 |  |  |
| 申请退款 | 公园游客误买或者多买了门票以后可以申请退款 | 审核退款，退款记录 | 后台管理员对发起的退款进行审核；游客可以在手机上查看自己的退款记录 |
| 查询消费明细 | 公园游客可以查询自己的消费记录和明细 |  |  |
| 查询充值记录 | 公园游客可以查询自己的充值记录和明细 |  |  |
| 查询余额 | 公园游客可以查询自己的会员卡内的剩余金额 |  |  |

不同类型的需求以不同程度对软件架构设计产生影响，非功能性需求通过约束和性能需求来影响软件架构，功能性需求通过引出软件系统的职责来影响软件的架构设计，可以通过功能性需求之间的关系确定如何划分子系统。比如分析了以上的功能性需求和其衍生需求后，结合相应的需求说明，我们进行确定关键功能的步骤，发现此公园票务系统的功能主要分为三个部分：

（1）游客注册、登录、办理会员卡、充值会员卡、浏览游乐项目、线上购买门票、使用电子门票、在游乐项目验票员处使用会员卡扣款游玩、评价订单、查询各项使用数据等等；

（2）公园的管理人员审核退票订单，查看各项运营数据（比如充值金额、会员人数等等）；

（3）游乐项目的验票员，检验门票；

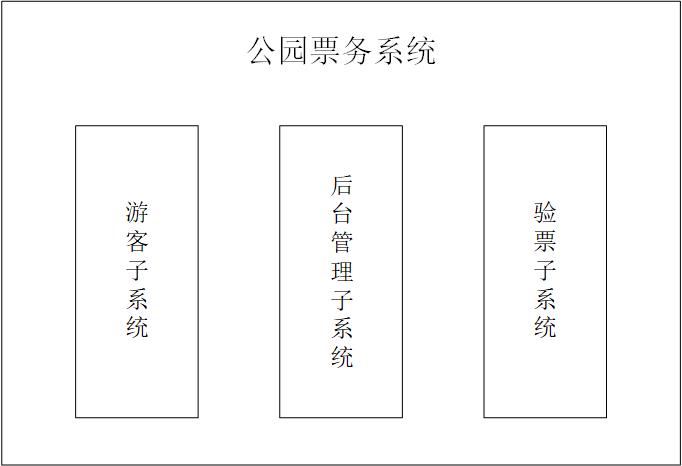
根据以上预架构阶段的分析，我们发现此公园票务系统可以分为三个子系统，分别是游客子系统、后台管理子系统、验票子系统。如图4.1所示：

图4.1 公园票务系统的子系统划分图

在确定了关键功能后，我们将使用ADMEMS方法对系统进行关键质量的确定，这个过程中有五个原则[45]，分别是分类必须合适且要进行扩充、不能仅考虑单一的受众、在整个过程中不断检查、能识别风险和解决的优先级、根据实际情况确定严格程度的原则。在此公园票务系统中，非功能性需求主要分为约束和性能需求。此公园票务系统的非功能性需求见表4-2：

表4-2 公园票务系统非功能性需求表

|  |  |  |
| --- | --- | --- |
| 需求分类 | 非功能需求 | 说明 |
| 约束 | 项目预算有限 | 公园方可提供的项目经费并不多，需要此系统在可以满足基本需求的目标上尽可能减少工作量 |
| 游客在公园使用手机操作 | 游客在公园游玩时使用手机进行在线操作，对游客端的易用性要求较高 |
| 验票在游乐项目处进行 | 游乐项目验票员需要高效、准确地验票 |
| 公园管理人员监测数据方便直观 | 公园管理人员在系统上查看运营数据时，需要简洁直观，数据要实时更新 |
| 性能需求 | 稳定性 | 公园票务系统要保持稳定运行，尤其是在节假日等交易高发期 |
| 安全性 | 公园票务系统涉及到钱款，对安全性要有保证 |
| 实用性 | 公园票务系统是根据实际需求出发开发的产品，一切立足于实际情况，具有实用性 |
| 可靠性 | 游客购票、公园验票等关键操作使公园票务系统必须具备可靠性 |
| 可扩展性 | 公园方有了新需求后，能够方便地新增功能模块 |

对于此公园票务系统的约束，一是项目预算有限，公园方作为此次项目的发起人，可提供的项目经费并不多，此公园票务系统要在满足基本需求的目标上尽可能减少工作量，综合各方面的考虑，我们决定游客子系统使用微信平台，不用去区分游客的手机是iOS系统或Android系统，不用单独开发APP，游客子系统使用微信平台可以在保证系统功能的前提下减少项目花销；二是游客在公园使用手机进行所有操作，对游客端的易用性要求较高，我们需要提高用户体验，如界面简洁易懂、显示提示信息引导游客操作等；三是验票在游乐项目处进行，游乐项目验票员需要高效、准确地验票，我们需要让门票的生成和识别都及时、可靠、准确；四是公园管理人员监测数据方便直观，在系统上查看运营数据时，需要简洁直观，数据要实时更新，我们要对后台管理子系统进行科学地设计以方便管理人员进行数据分析的工作。

对于性能需求，公园票务系统要保持稳定运行，尤其是在节假日等公园人流密集的交易高发期不能出现系统问题；公园票务系统涉及到钱款，对安全性的要求很高，公园的收入资金要确保万无一失，不能被非法人员盗用；公园票务系统是根据实际需求出发开发的产品，要立足于实际情况具有实用性，比如在游乐项目验票员处使用会员卡扣款游玩就是从实际情况出发，在不必要的时候减少游客的操作步骤；游客购票、公园验票等关键操作涉及游客的实时游玩体验和钱款，公园票务系统必须具备可靠性；公园方有了新需求后，能够方便地新增功能模块，要求系统在开发的过程中，注意可扩展性。

## 4.2 概念架构阶段

概念架构阶段（CA阶段）针对重大的需求、高风险需求、特色需求确定概念架构，形成稳定的高层软件结构设计结果，考虑质量、功能、约束各个方面的需求给出解决方案。概念架构的重点在于系统的组件、组件与组件之间的关系，概念架构并不关注细节，为了宏观地把握软件系统的组成[46]，从整体上对系统进行分解，最终产出系统的架构图，其中不包含接口设计等细节。概念架构阶段分为三个步骤，第一步是初步设计，第二步是高层分割，第三步是考虑非功能需求，下面将按照这三个步骤对公园票务系统的概念架构阶段进行详细介绍。

（1）初步设计

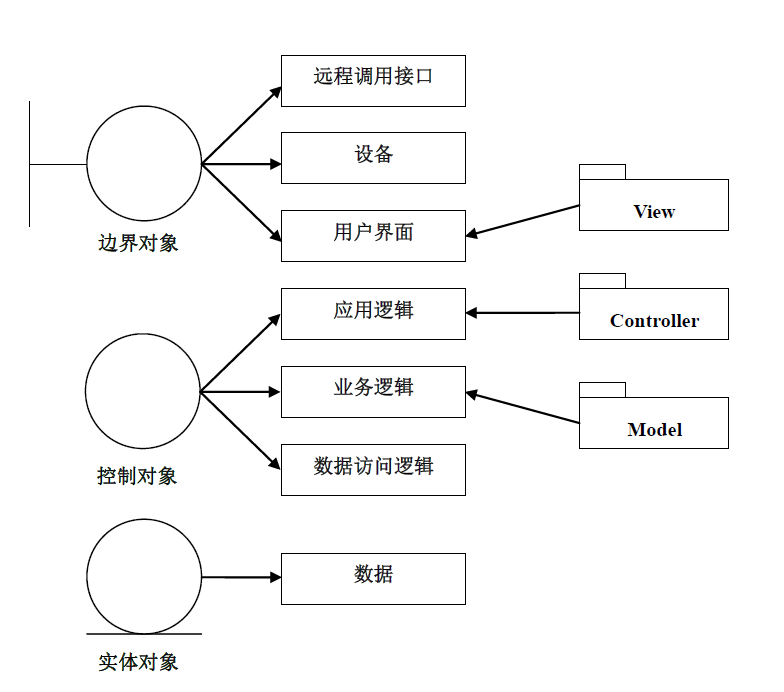
初步设计的工作是基于系统的关键功能，采用鲁棒图，使用“职责协作链”去确定职责。鲁棒图也被叫做鲁棒性分析，是使用在需求设计过程中的方法，此方法使用在需求设计前、需求分析后，目的是让设计人员全面地、清晰地了解需求进行软件架构分析，它关注的重点是功能需求的分析工作，向设计方案进行。边界对象用来接受外部的输入信息，经过内部处理后表达出对应的结果，对外部输入和软件系统间的交互活动进行建模，通俗的说就是负责交互的UI部分。实体对象负责信息，因为一般情况下实体对象来自于领域概念，所以和领域模型的对象有很大程度的对应关系，对要储存的信息实施描述行为。控制对象负责控制，通俗地说是一种业务方法，对行为封装并且说明用例里事件流的控制行为。鲁棒图中的边界对象、实体对象和控制对象对现实系统中对象的职责做了非常准确的概括，即交互、信息和控制。这些职责很好的对应了组成软件架构的抽象元素，即连接元素、数据元素和处理元素。我们将鲁棒图与传统的MVC模型做一个全面的类比，发现两者之间有着一些异同，总的来说还是有一定差异，如下图4.2所示：

图4.2鲁棒图与MVC模型的异同

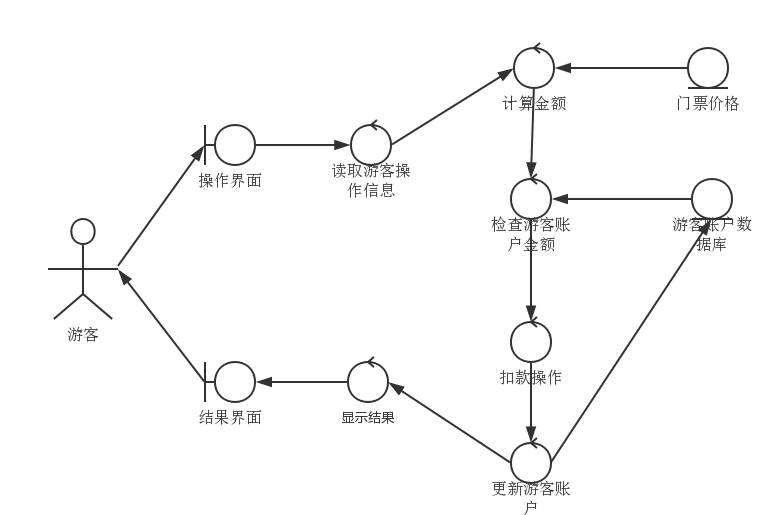
了解了鲁棒图的绘制方法和作用后，下面选择公园票务系统的核心功能——在线购票功能进行鲁棒分析，如图4.3所示：

图4.3公园票务系统——购票鲁棒图

通过此次鲁棒图的绘制，在线购票功能所需要的职责模块被清晰地展现出来，可以进行下一步高层分割的步骤。公园票务系统的其他主要功能的鲁棒图绘制原理和过程与此图类似，不再一一在论文中体现。

（2）高层分割

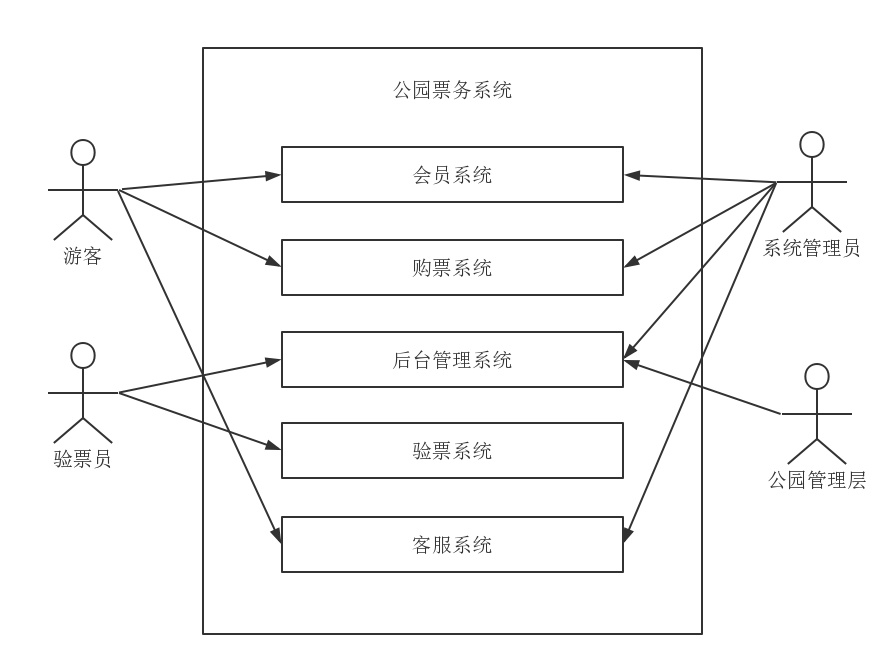
高层分割是在上一步初步设计的过程后，对公园票务系统进行划分，整个划分过程将公园票务系统看作黑盒。通常情况下对于关系众多、功能复杂的系统划分的方式有以下两种：将庞大且复杂的系统划分成为几个二级系统，将庞大且复杂的系统划分为具体子系统。对于公园票务系统，我们在这里进行高层分割的方式选择了第一种，即是将公园票务系统根据具体需求分割为多个二级系统。对于公园票务系统而言，它由会员系统、购票系统、验票系统、后台管理系统、客户服务系统组成，公园票务系统的组成图，如下图4.4所示：

图4.4公园票务系统的组成图

在完成以上划分过程后，我们将继续采用实践中一种最常用的系统切分方式，对系统继续进行分层。因为分层的角度不同，可以有以下几种流派：逻辑分层（Layer）、物理分层（Tier）、按通用性进行分层、技术堆叠。下面，我们依次简单介绍一下这几种分层流派，对于逻辑分层而言，职责的划分被看重，职责与职责间的关系往往是上层调用下层，逻辑分层对于上层和下层能否分布于不同的机器并不重视。物理分层是指可以分布于不同机器的软件层，层与层间必定要有使用通信协议或者进行远程调用的能力，简洁地说，也就是层与层直接有跨机器进行访问的能力。按通用性进行分层强调“通用性”，严格来说算是逻辑分层的一种子类，根据通用性的不同划分不同的层，通用程度大则所处的层越靠下，举例来说比如应用层、操作系统层、硬件驱动程序层。技术堆叠依赖于其他的分层方式，综合了多种分层方式，为层的技术实现提供深层的说明。综合这四种流派的特点，我们采用了物理分层，即Tier分层的模式对公园票务系统开展分层架构，分析后得到公园票务系统物理分层结构图，如下图4.5所示：

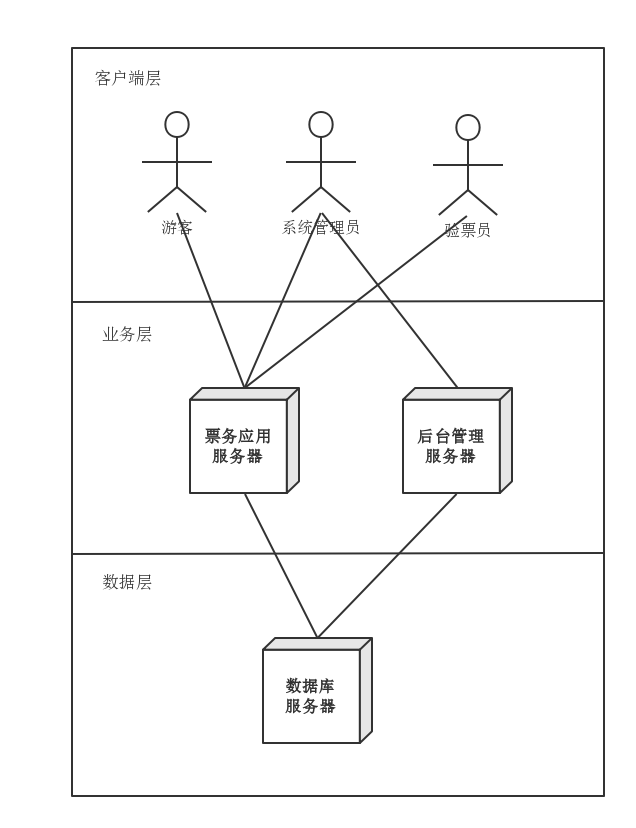


图4.5 公园票务系统物理分层结构图

在采用物理分层对公园票务系统开展分层架构后，作为软件架构师需要考虑非功能需求，原因是概念架构阶段并不是简单的进行理想化架构，除了考虑客户提出的功能性需求外，也必须要考虑到决定项目质量的非功能需求。

在ADMEMS软件架构方法中，通常采用“目标—场景—决策”表来进行这一过程。我们结合预架构阶段对于非功能需求的总结和分析，结合实际情况选取公园票务系统的重点非功能需求，建立表4-3：

表4-3 公园票务系统“目标—场景—决策”表

|  |  |  |
| --- | --- | --- |
| 目标 | 场景 | 决策 |
| 安全性 | 公园票务系统如果被攻击，涉及到钱款的数据可能被更改造成经济损失。 | 建立防火墙 |
| 可靠性 | 网络服务可能会有问题，确保公园票务系统中的数据万无一失。 | 建立数据备份中心 |

通过“目标—场景—决策”表考虑非功能需求后，公园票务系统的架构进行改进，如图4.6所示：

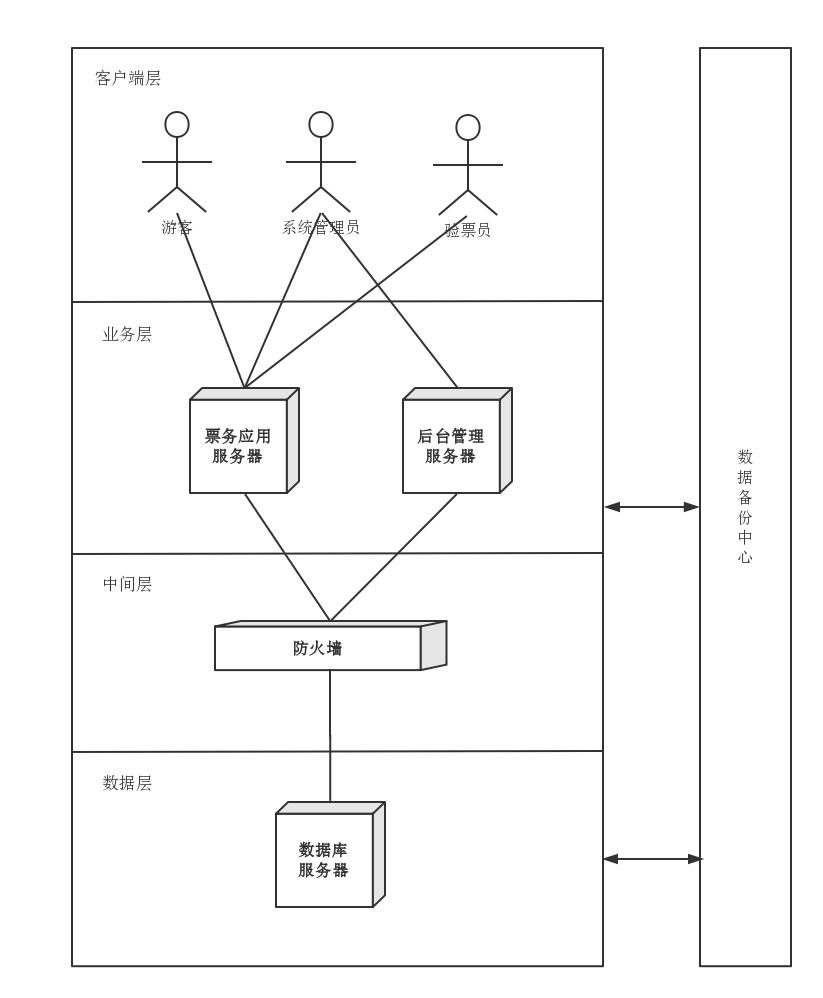


图4.6 改进后的公园票务系统架构图

## 4.3 细化架构阶段

细化架构阶段（RA阶段）通过5视图法来关注不同的视图，涉及物理架构、逻辑架构、数据架构、开发架构、运行架构，对细化架构阶段进行实践。物理架构包括对系统的软件和硬件要求，对结点的物理部署设计；逻辑架构包括对系统的分层、分区、框架提取，根据模块间的协作关系来确定模块接口；数据架构指的是对数据库的设计；开发架构指的是对开发的目录结构进行规划，确定project的数量；运行架构主要是对线程和进程的设计。所做的工作有在整个软件项目的概要设计阶段，进行架构设计并制定约束与规范，主要关注对子系统的划分和接口的定义等，为后面的详细设计提供指导。

使用5视图法进行细化架构阶段可以对公园票务系统进行几乎没有遗漏的全面分析，具有诸多对系统的架构和开发具有积极意义的优点。可以提前发现项目的风险并寻找解决方案；可以从不同的角度对该项目的软件架构进行设计和表达；可以为项目中的不同角色提供实际指导，对于测试人员而言，系统的功能在逻辑架构视图中有较为详细的描述，对于开发工程师而言，代码的风格以及软件的层次都在开发架构视图中有了体现，关键过程的设计通过运行架构视图展现，对于数据库工程师而言，数据库的设计在数据架构视图中有了展现，对于部署工程师而言，软件部署和实施的方法在物理架构视图中有了说明。有一点需要注意的是，使用5视图法进行细化架构时，要根据现实情况进行具体分析，往往并不是严格按照次序进行5视图的设计，而是对整个过程进行穿插思考，也就是说这5个视图在构建的时间上是不分先后的。对于公园票务系统而言，进行5视图的穿插思考有利于系统更完整和合理，提高了思考的全局性。

（1）逻辑架构视图的设计

逻辑架构主要通过3种方法来对公园票务系统进行功能划分和设计，这3种方法依次是细化分层、引入分区和提取机制，它们从3个各异的角度来描述逻辑架构的过程。细化分层是在公园票务系统经过初步的设计之后，对设计出来的分层再次进行细化，达到让公园票务系统的构成愈加明晰的目的，这样做对项目组成员的协同开发有积极作用。引入分区是对公园票务系统的功能设计加深深度，我们用职责链来展示公园票务系统的功能，职责链上展示的单个职责即为公园票务系统的功能模块。细化分层对公园票务系统的软件架构在广度上做出规定，引入分区是在一定深度上完成公园票务系统的某个功能。对公园票务系统进行细化架构中的逻辑架构步骤后，得到如下逻辑架构图，见图4.7：

图4.7 公园票务系统逻辑架构图

（2）物理架构视图的设计

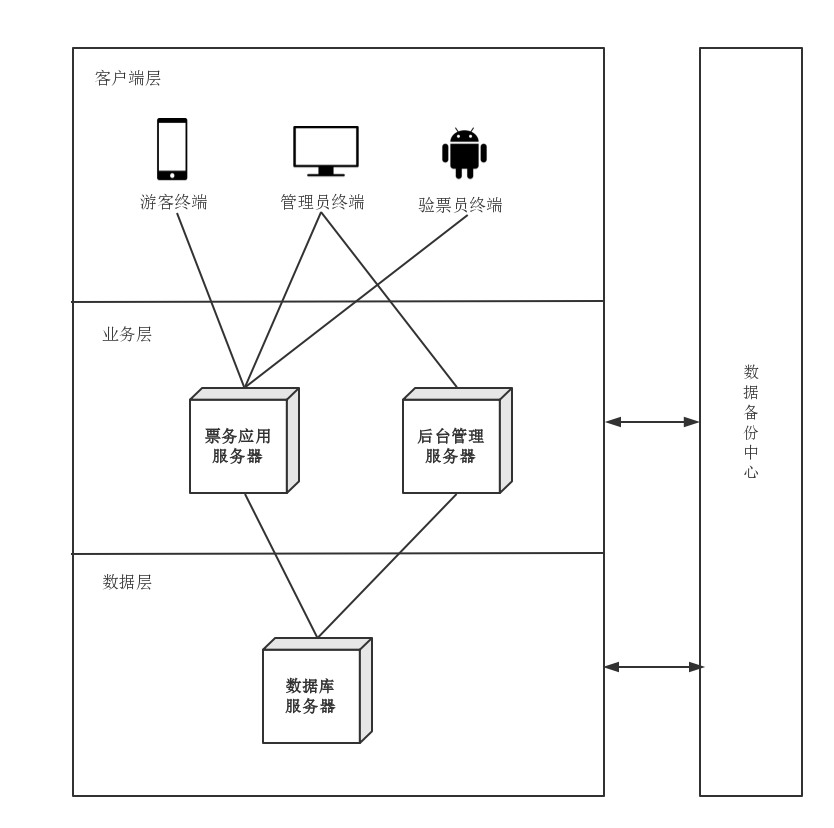
物理架构视图主要需要体现系统中的网络和硬件设备的具体情况，以及硬件平台和相应部署的软件，物理架构的合理设计使得系统能够很好地顺利运行，物理架构视图设计的同时也要考虑软件系统安全性、可靠性等非功能需求。公园票务管理系统的物理架构视图，如图4.8所示：

图4.8 公园票务系统物理架构视图

（3）运行架构视图的设计

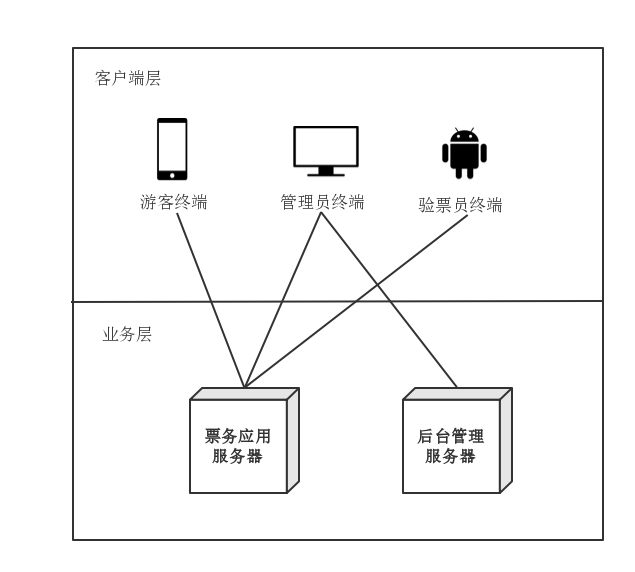
公园票务管理系统在运行时依靠不同的软件功能模块协同工作，通过合理设计运行架构视图，能够保证软件功能模块之间有序地协同工作。公园票务管理系统的运行架构视图，如图4.9所示：

图4.9 公园票务系统运行架构视图

（4）开发架构视图的设计

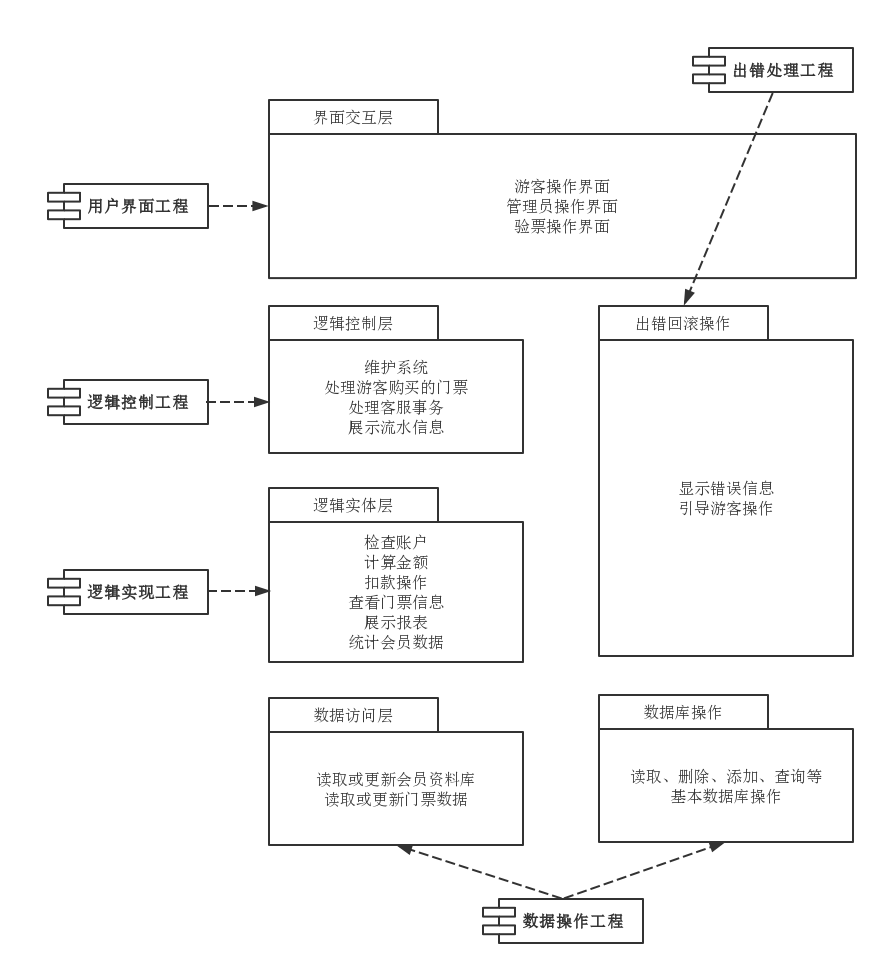
通过合理设计开发架构视图，能够对开发工程师的具体工作进行指导，开发工程师需要按照开发架构视图中的架构设计进行开发工作。开发架构设计了开发工程师感兴趣的“程序单元”等具体内容。公园票务管理系统的开发架构视图，如图4.10所示：

图4.10 公园票务系统开发架构图

（5）数据库架构视图的设计

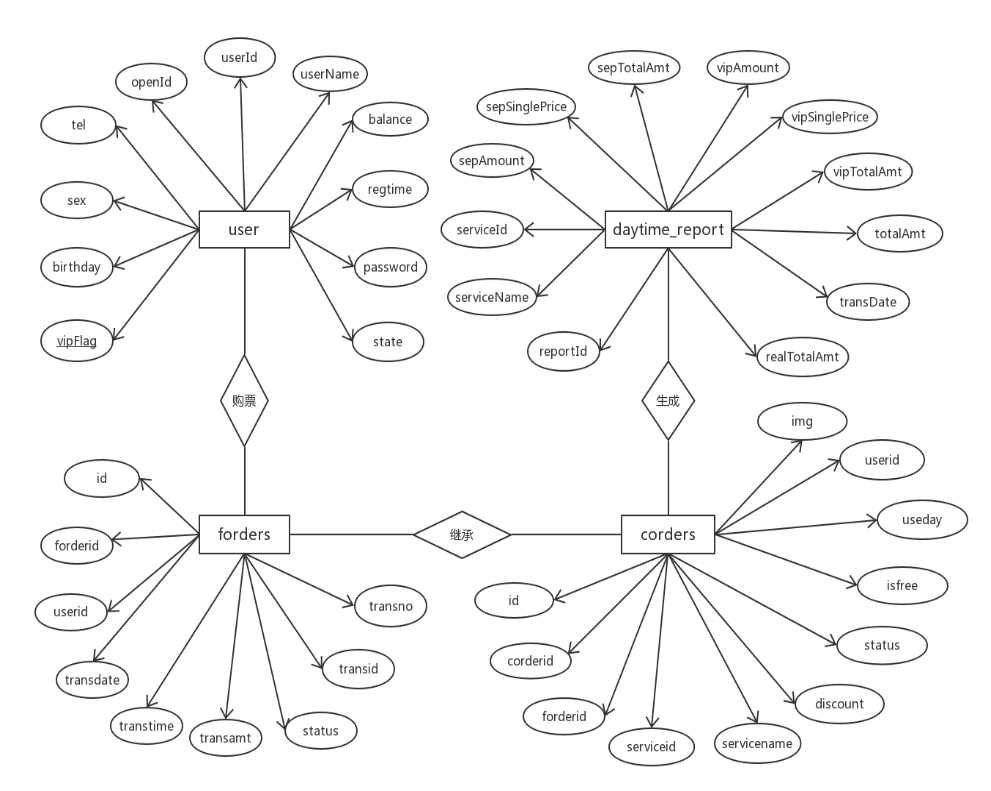
公园票务管理系统的数据库架构视图在设计过程中，遵循了ADMEMS中的数据架构方法策略，包括集中、独立、复制、分区。除了以上方法策略，进行数据库架构设计时还要注意三个应用原则：合适原则，要根据系统的实际情况和特点，选择最合适的设计；优化原则，对总体性能进行评估后尽可能再去优化；综合原则，综合实际情况，在多种不同的数据库架构中选取优点进行综合。对公园票务管理系统的进行数据架构后，我们在此选择E-R图的形式，对此次的核心数据库表间的关系做出展示，如图4.11所示：

图4.11 实体-关系图

# 第五章 公园票务系统的实现

本章介绍了公园票务系统的实现，其中包括系统的数据库设计、游客子系统的实现、后台管理子系统的实现、验票子系统的实现。游客子系统可以给游客提供办理会员卡、充值会员卡、在线购票、在验票点使用会员卡二维码购票、评价订单、申请退票等功能，为了让不同手机操作系统的游客可以方便地进行购票相关操作，此游客子系统选择了基于微信服务号开发。后台管理子系统主要满足了公园的管理层提供监测实时数据的需求，同时还可以查看各项记录、审核退票订单、管理游乐项目、导出财务报表、对账号进行角色管理和权限管理等。为了便于查看数据和管理相关内容，后台管理子系统采取了Web网页的形式。验票子系统在安卓手持扫码器上运行，主要用于扫描游乐项目的二维码门票进行验票或扫描会员卡二维码进行游乐项目扣款，验票子系统可以自动检测是否为最新版本，如果不是最新版本可以选择更新。游客子系统、后台管理子系统的实现、验票子系统三个子系统共同组成了公园票务系统，满足了各方需求。

## 5.1 系统的数据库设计

数据库设计是在对系统进行需求分析、深入理解系统的实际需求的基础上，在数据库系统中进行数据库的概念结构设计、逻辑结构设计后得到该系统数据库文件的重要过程[47-49]。数据库设计的意义在于通过设计数据对象以及它们之间的关系，使得数据被有效率地存储和使用，从而使系统的实际需求得到满足。因为数据库的设计具有一定的繁杂程度，所以整个设计的过程并不能一次性快速地完成，要经过不断的探寻和反复思考，最佳的设计方式就是从粗粒度设计到细粒度设计[50]，即先进行数据库的概念结构设计，后进行逻辑结构设计，尽力使设计出的数据库系统具备优良的性能，为整个系统提供很好的数据支持[51]。

### 5.1.1 逻辑结构设计

我们已经在本文第四章的细化架构阶段中的数据架构阶段，通过对公园票务系统进行需求分析、深入理解公园票务系统的实际需求，完成了此公园票务系统的数据库概念结构设计过程，设计出逻辑上的数据架构并用E-R图（Entity Relationship Diagram）即实体-联系图来表示这个和系统具体实现无关的过程。

下面我们根据此公园票务系统的功能性需求为背景，选取了部分比较重要的数据库表作为逻辑结构设计实例进行列举，覆盖到了公园票务系统的大多数实际应用场景。

（1）comments表用于储存公园票务系统中项目评论的相关信息，见表5-1：

表5-1 comments表

|  |  |  |  |
| --- | --- | --- | --- |
| **字段名称** | **数据类型** | **说明** | **表空值** |
| commentid | int(11) | 评论id | no |
| corderid | varchar(50) | 子订单id | no |
| serviceid | varchar(5) | 项目id | no |
| userid | varchar(5) | 用户id | no |
| commenttime | date | 评论时间 | no |
| score | int(11) | 评分 | no |
| discuss | text | 添加的评论 | no |
| isuse | int(2) | 是否显示在评论栏  0-不显示  1-显示 | no |

（2）corders表用于储存公园票务系统中子订单的相关信息，见表5-2：

表5-2 corders表

|  |  |  |  |
| --- | --- | --- | --- |
| **字段名称** | **数据类型** | **说明** | **表空值** |
| id | int(11) | 序号 | no |
| corderid | varchar(50) | 子订单 | no |
| forderid | varchar(50) | 父订单 | yes |
| serviceid | varchar(5) | 项目id | no |
| servicename | varchar(255) | 项目名称 | yes |
| discount | decimal(5,2) | 折扣 | no |
| status | int(11) | 状态  0-待付款  1-待使用  2-待评价  3-已评价  4-已退款 | no |
| isfree | int(2) | 是否为赠票  0-不是  1-是 | no |
| useday | varchar(12) | 使用日期 | yes |
| userid | varchar(20) | 用户id | no |
| img | varchar(50) | 服务图片 | no |

（3）daytime\_report表用于储存公园票务系统中每日报表的相关信息，见表5-3：

表5-3 daytime\_report表

|  |  |  |  |
| --- | --- | --- | --- |
| **字段名称** | **数据类型** | **说明** | **表空值** |
| reportId | varchar(32) | 报表编号 | no |
| serviceName | varchar(20) | 服务项目名称 | yes |
| serviceId | varchar(32) | 服务id | yes |
| sepAmount | int(11) | 零散购票次数 | yes |
| sepSinglePrice | decimal(16,2) | 零散售票单价 | yes |
| sepTotalAmt | decimal(16,2) | 零散购票总金额 | yes |
| vipAmount | int(11) | 会员购票数量 | yes |
| vipSinglePrice | decimal(16,2) | 会员购票单价 | yes |
| vipTotalAmt | decimal(16,2) | 会员购票总价 | yes |
| totalAmt | decimal(16,2) | 总金额 | yes |
| transDate | varchar(12) | 交易日期 | yes |
| realTotalAmt | decimal(16,2) | 真实的总收益 | yes |

（4）forders表用于储存公园票务系统中购票行为的相关信息，见表5-4：

表5-4 forders表

|  |  |  |  |
| --- | --- | --- | --- |
| **字段名称** | **数据类型** | **说明** | **表空值** |
| id | int(11) | 序号 | no |
| forderid | varchar(50) | 父订单号 | no |
| userid | varchar(20) | 用户id | no |
| transdate | varchar(12) | 购票日期 | no |
| transtime | varchar(12) | 购票时间 | no |
| transamt | varchar(10) | 购票金额 | no |
| status | int(11) | 0-当前待支付  1-待支付  2-支付完成  3-已使用 | no |
| transid | varchar(50) | 交易单号 | yes |
| transno | varchar(50) | 商户单号 | yes |

（5）orders表用于储存公园票务系统中交易订单的相关信息，见表5-5：

表5-5 orders表

|  |  |  |  |
| --- | --- | --- | --- |
| **字段名称** | **数据类型** | **说明** | **表空值** |
| orderId | varchar(32) | 订单号 | no |
| transDate | varchar(12) | 交易日期 | yes |
| transTime | varchar(12) | 交易时间 | yes |
| transAmt | decimal(16,2) | 交易金额 | yes |
| userId | varchar(32) | 用户id | yes |
| userName | varchar(32) | 用户名 | yes |
| tel | varchar(20) | 手机号 | yes |
| orderType | varchar(4) | 订单类型  0-账户充值  1-门票购买  2-被扫码购票  3-退款 | yes |
| channel | varchar(2) | 订单发起渠道  00-后台  01-微信端  02-手持端 | yes |
| serviceId | varchar(32) | 购买服务单号 | yes |
| orderStatus | varchar(2) | 订单状态  0-待支付  1-成功  2-处理中  3-失败  4-超时  5-退款中  6-退款待审核  7-退款成功 | yes |
| addmoney | decimal(16,2) | 充值金额 | yes |
| transid | varchar(100) | 交易单号 | yes |
| tradeno | varchar(100) | 微信流水单号 | yes |
| realmoney | decimal(16,2) | 实际金额 | yes |
| oriOrderId | varchar(32) | 原订单号用于退款 | yes |

（6）service表用于储存公园票务系统中游乐服务的相关信息，见表5-6：

表5-6 service表

|  |  |  |  |
| --- | --- | --- | --- |
| **字段名称** | **数据类型** | **说明** | **表空值** |
| serviceId | varchar(32) | 服务唯一标志 | no |
| serviceName | varchar(20) | 产品服务名称 | yes |
| desc | text | 描述 | yes |
| price | decimal(16,2) | 价格 | yes |
| amount | int(11) | 商品数量 | yes |
| img | varchar(50) | 图片 | yes |
| shopprice | decimal(10,2) | 商城价格 | yes |
| fastprice | decimal(10,2) | 秒杀价格 | yes |
| feature | varchar(200) | 特点 | no |
| level | varchar(200) | 刺激指数 | no |
| intro | varchar(200) | 简介 | no |
| fit | varchar(500) | 适合人群 | no |
| unfit | varchar(500) | 不适合人群 | no |
| remark | varchar(500) | 备注 | no |

（7）transserial表用于储存公园票务系统中交易的相关信息，见表5-7：

表5-7 transserial表

|  |  |  |  |
| --- | --- | --- | --- |
| **字段名称** | **数据类型** | **说明** | **表空值** |
| serialId | varchar(32) | 流水号 | no |
| transDate | varchar(12) | 交易日期 | yes |
| transTime | varchar(12) | 交易时间 | yes |
| transType | varchar(2) | 交易类型 | yes |
| transStatus | varchar(2) | 交易状态 | yes |
| openId | varchar(32) | 支付微信号 | yes |
| accountNo | varchar(32) | 余额账户 | yes |
| serivceId | varchar(32) | 购买服务编号 | yes |
| orderNo | varchar(32) | 订单号 | yes |
| transid | varchar(100) | 交易单号 | yes |
| tradeno | varchar(100) | 微信流水单号 | yes |

（8）user表用于储存公园票务系统中用户的相关信息，见表5-8：

表5-8 user表

|  |  |  |  |
| --- | --- | --- | --- |
| **字段名称** | **数据类型** | **说明** | **表空值** |
| userId | bigint(20) | 用户id | no |
| userName | varchar(50) | 用户名 | yes |
| openId | varchar(32) | 绑定用户的openId | yes |
| tel | varchar(20) | 手机号 | yes |
| sex | char(1) | 性别  1-男  2-女 | yes |
| birthday | varchar(32) | 生日 | yes |
| vipFlag | char(1) | 是否是会员  0-否  1-是 | yes |
| balance | decimal(16,2) | 余额 | yes |
| regtime | datetime | 注册时间 | yes |
| password | char(64) | 密码 | yes |
| state | varchar(32) | 状态 | yes |

### 5.1.2 数据库软件的选择

市场上有着种类繁多的数据库软件，我们需要选择出适合此公园票务系统的数据库软件投入使用。在挑选数据库软件和评估其优缺点的过程中，我们要综合考虑以下5个因素：公园票务系统的开发要求、公园票务系统要达到的性能要求、我们要花费的成本是多少、数据库的运行和管理是否方便高效、数据库的可升级性如何。目前市面上主流的数据库有MySQL、SQL Server、Access、Oracle、DB2等[52]。

MySQL是由MySQL AB开发、发布和支持的开源SQL数据库管理系统，在目前市面上人气很高。MySQL服务器可以支持关键任务、重负载生产系统的使用，同时也可以嵌入到一个大配置的软件中去[53]。MySQL的用户数在同类开源数据库中领先于竞品，能够适用于采用了操作系统集群的大型网络站点使用，也能够适用于规模小的新兴公司。对于数据恢复的操作，MySQL具有绝佳的表现。MySQL拥有种类繁多的插入式的存储引擎，来给使用者提供不同的所需功能，而不是默认具有全部功能。SQL Server是目前网络上使用最广泛的用于存储数据的数据库，由微软开发，已经被广泛的用于银行、物流系统、电子商务、电子政务等与数据库相关的行业，操作界面友好。操作系统的系统稳定性对数据库有着重大影响，它仅仅在Windows上运行，并行实施与共存模型还不算成熟，处理海量数据的用户数和数据卷在一定程度上有困难，伸缩性有限[54-56]。SQL Server 提供了众多的Web和电商功能，如对可扩展标记语言和互联网标准的丰富支持，通过网络对数据进行轻松安全的访问，具备灵活的、强大的、基于Web的、安全的应用程序管理等。

结合主流数据库的对比分析，在目前市场上种类繁多的数据库产品中，对于公园票务系统此次选择了My SQL数据库。My SQL作为一款开源的免费数据库管理系统，为种类繁多的编程语言提供了API，其中包括Python、C、C++、PHP、Java等，尤其对我们用到的PHP开发语言有很好的支持。而且平时的数据库操作可以使用SQLyog等图形化管理工具，非常方便和可靠。

## 5.2 游客子系统的实现

游客子系统可以给游客提供办理会员卡、充值会员卡、在线购票、在游乐项目验票点使用会员卡二维码购票、评价订单、申请退票等功能。由于游客在公园中使用手机进行以上操作，我们考虑到APP需要下载安装，安卓用户、iOS用户使用的APP需要分开进行开发，研发经费并不充足等问题，综合考虑各方面的实际情况，最终选择了以微信服务号的形式开发游客子系统。公园游客关注此微信服务号，进行注册后就可以进行所有的操作，免去了下载安装APP的麻烦，对开发者来说游客子系统的实现也无需区分手机的操作系统类型。开发游客子系统我们选择了PHP，它是一种服务器端脚本语言，作为目前流行的Web程序开发语言，易于被学习和被使用[57]。它融合了例如Java语言、C语言、Perl语言的优势特性，加入了PHP的自创新语法，具有灵活好用、便于学习、功能强大、语法简洁的特点。PHP作为一种在服务器端执行的脚本语言，嵌入HTML文档，比CGI语言或Perl语言更快速地执行动态网页，目前被大量的Web开发工程师广泛运用，用PHP做出的动态页面执行效率非常高。PHP具有强大的功能，所有的CGI或JavaScript的功能都能实现[58-59]，而且支持几乎所有主流操作系统和主流数据库，无需考虑跨平台问题。

（1）会员注册

游客使用微信客户端进入“武汉中山公园”服务号后，菜单栏有三项可以点击，包括“游玩购票”、“微官网”、“关于我们”。游客点击“游玩购票”该项菜单栏会展开，点击其中的第一项“会员中心”，如果是第一次进入此服务号，点击“会员中心”后会跳转到“会员注册”的页面，游客需要填写手机号、性别、生日这几项个人信息，然后点击“获取验证码”，等待验证码发送到填写的手机号，输入验证码点击“会员注册”按钮，就完成了会员注册的过程，如图5.1所示：



图5.1 会员注册

（2）会员中心

完成了会员注册的流程后，之后如果再次点击“会员中心”，服务号的状态是默认登录的状态，直接进入该游客的会员中心。会员中心展示了会员卡的样式，显示了该会员的手机号。可点击区域一共分为三个部分，如图5.2所示：



图5.2 会员中心

第一部分是“我的余额”和“我的卡券”，点击“我的余额”可以查看该用户的会员卡余额，点击“我的卡券”可以看见卡券包，其中包括待使用的卡券、待评价的卡券和卡券的退款记录。第二部分是“支付功能”，其中包含两种方式，第一种是“扫码支付”，即到游乐项目的验票员处出示自己的会员卡二维码，验票员用机器扫码后直接从会员卡账户中扣款，扣款成功后可以进入到该游乐项目进行游玩，如图5.3所示：

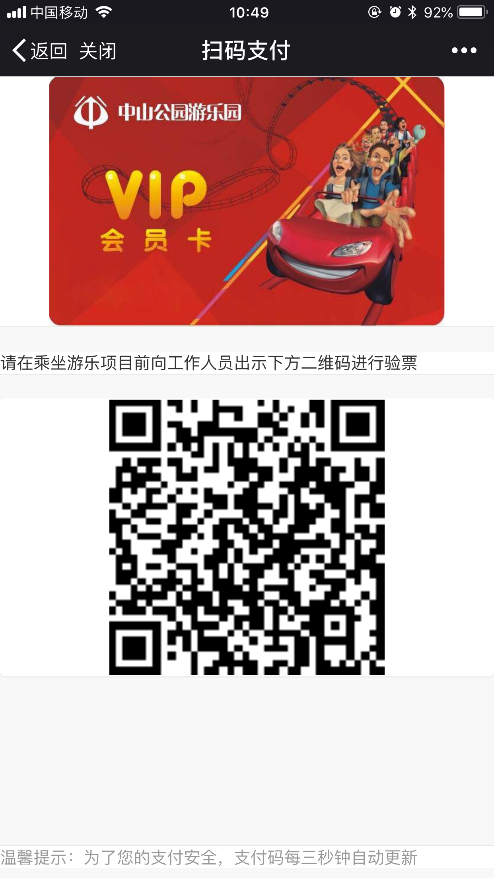


图5.3 扫码支付

扫码支付时按一定规则生成一个二维码，包含游客信息，并且会在成功被扫码收款后自动更新，且付款成功后，会向数据库中写入相关信息，核心代码如图5.4所示：



图 5.4 扫码支付核心代码

“支付功能”的第二种方式是“在线充值”，点击后进入充值页面，最上方显示“我的余额”，下方可以选择具体的充值金额并伴有充值优惠，游客选择某种充值方案后该方案会标红，点击下方按钮“立即充值”后跳转到结算页面，以充值200元为例，结算页面会显示“您的充值金额为200元，您得到的金额为250元”，点击“确认支付”按钮，弹出微信支付弹框供游客完成付款，此页面包含提示内容“点击立即充值，即表示您已同意《充值协议》”。第三部分是“查询中心”，包含“消费明细查询”和“充值记录查询”。点击“消费明细查询”即可以查看消费记录列表，其中包含日期、时间、项目名称、消费金额和消费状态，消费状态分为“已扣款”和“已退款”。

（3）优惠活动

游客在中山公园服务号的菜单栏点击“游玩购票”该项菜单栏展开，点击其中的第二项“优惠活动”，即跳转到会员卡的优惠充值活动页面，与上文中的“在线充值”是同一页面的不同入口。此充值页面最上方显示“我的余额”，下方可以选择系统提供的5种充值方案，系统提供了五种充值方案如下，“充值100元赠送20元”、“充值200元赠送50元”、“充值400元赠送100元”、“充值700元赠送300元”、“充值1000元赠送500元”，游客选择某种充值方案后该方案会标红，如图5.5所示：



图5.5 优惠活动

点击下方按钮“立即充值”后跳转到结算页面，显示充值金额（即支付金额）与实得金额（即会员卡中实际增加的金额），点击“确认支付”按钮，弹出微信支付弹框供游客付款，付款完成后提示“付款成功”。此部分核心代码如图5.6所示：

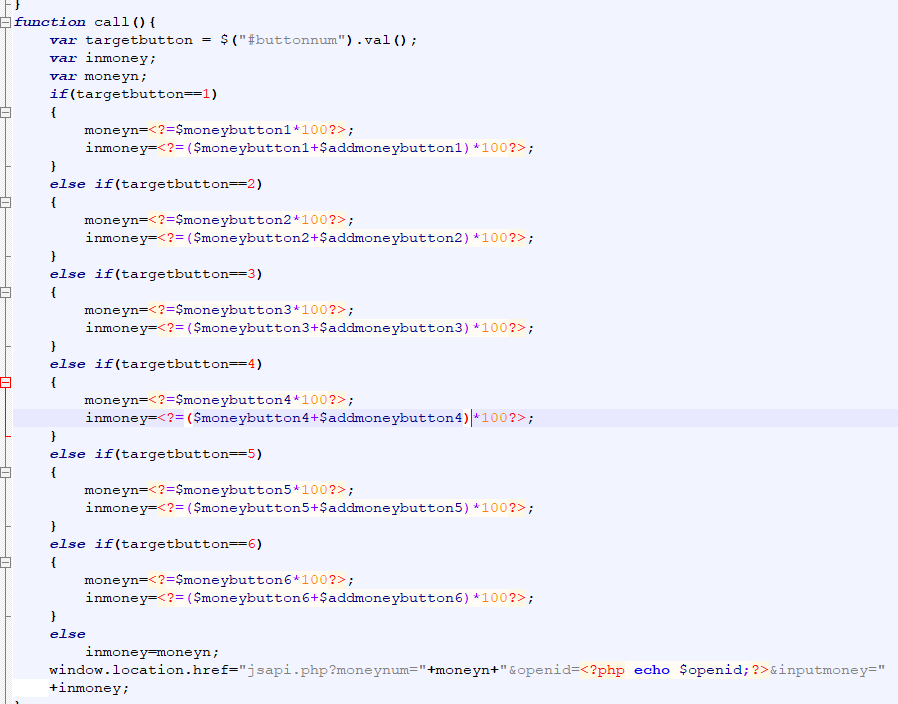


图5.6 充值核心代码

（4）单项购票

游客在中山公园服务号的菜单栏点击“游玩购票”该项菜单栏展开，点击其中的第三项“单项购票”，即跳转到门票列表页面。此页面上方有滚动展示的中山公园风景图片，让游客可以欣赏中山公园的景色；中间有搜索栏，可以根据游客输入的关键字搜索出相关的游乐项目；滑动此页面可以查看完整的游乐项目门票列表。如图5.7所示：



图5.7 单项购票

门票列表的部分在此页面上展示了门票ID、游乐项目名称、价格、游乐项目特点、“详情”按钮，游客可以选择相应的门票数量，则计入购物车，在此页面下方显示合计金额和“去结算”按钮。游客选择完门票种类和数量后，点击“去结算”按钮，跳转到付款页面，此页面再次展示游客选择的门票列表供游客进行核对，包含门票ID、游乐项目名称、游乐项目价格、购买数量、应付金额，游客核对无误后，可以点击下方的“确认支付”按钮，弹出微信支付弹框供游客付款，付款完成后提示“付款成功”。此部分核心代码如图5.8所示：



图5.8 购票页面核心代码

门票列表页面的右侧还有固定位置悬浮的“个人中心”按钮，方便游客随时可以点击进入“个人中心”，其中包含待使用的门票列表、待评价的门票列表和退款记录，在待使用的门票列表中点击“我要使用”按钮，即可快速使用门票，如图5.9所示：



图5.9 个人中心（我的订单）

（5）使用帮助

游客在中山公园服务号的菜单栏点击“游玩购票”该项菜单栏展开，点击其中的第四项“使用帮助”，即跳转到中山公园微信商城使用帮助页面。此页面简明地对使用此游客子系统的流程进行了指导，分为“用户注册”、“会员卡使用帮助”、“单项购票使用帮助”三个部分，其中“会员卡使用帮助”包含“会员卡充值”、“会员卡使用”、“游客须知”三个部分，“单项购票使用帮助”包含“购买”、“游客须知”两个部分。

（6）微官网

游客在中山公园服务号的菜单栏点击“微官网”，跳转到中山公园官网，此网站上有中山公园的介绍、公园最新活动、公园服务指南、公园景观等板块，网站有专门的工作人员运营，内容丰富、质量高，此处跳转到中山公园官网的作用主要是起到宣传作用。

（7）关于我们

游客在中山公园服务号的菜单栏点击“关于我们”，该项菜单栏会展开为“公园介绍”和“联系客服”。“公园介绍”页面对中山公园的主要信息进行了一些介绍，附有公园地址、服务监督电话和游客服务电话。“联系客服”页面有客服功能的操作说明，游客遇到问题时联系客服即可有专人解答疑问。

## 5.3 后台管理子系统的实现

后台管理子系统主要满足了公园的管理层提供监测实时数据的需求，同时还可以查看各项记录、审核退票订单、管理游乐项目、导出财务报表、对账号进行角色管理和权限管理等。为了便于查看数据和管理相关内容，后台管理子系统采取了Web网页的形式。后台管理子系统选择了Java语言进行开发，Java丢弃了C++里难以理解的概念，比如多继承、指针等，吸取C++的诸多优点，具有简便、易于使用、功能强大三个特征[60-62]。Java语言很好地对面向对象的理论进行实现，是一门面向对象的编程语言，对于开发工程师来说，在使用它的过程中不需要严格地把思考重点放在过程上，而是要注意数据和操作数据的办法，可以使用它优雅地进行复杂的编程。

（1）登录

输入网址进入到中山公园票务管理系统的后台管理子系统登录页面，此页面需要填写用户名和密码后点击“登录”按钮，可以勾选“记住我”便于下次直接登录。如图5.10所示：



图5.10 后台管理子系统登录界面

（2）概览

登录后台管理子系统后，首页展示的是概览的数据，其中包含会员总数、今日新增会员数、今日充值人数、今日充值金额、本月充值金额、本月消费金额、会员余额、实存余额等关键数据，如图5.11所示：



图5.11 后台管理子系统概览

此页面通过Web Service模式，调用PHP接口，获取跨域json数据，并相对应地展示到首页页面信息中去，核心代码如图5.12所示：

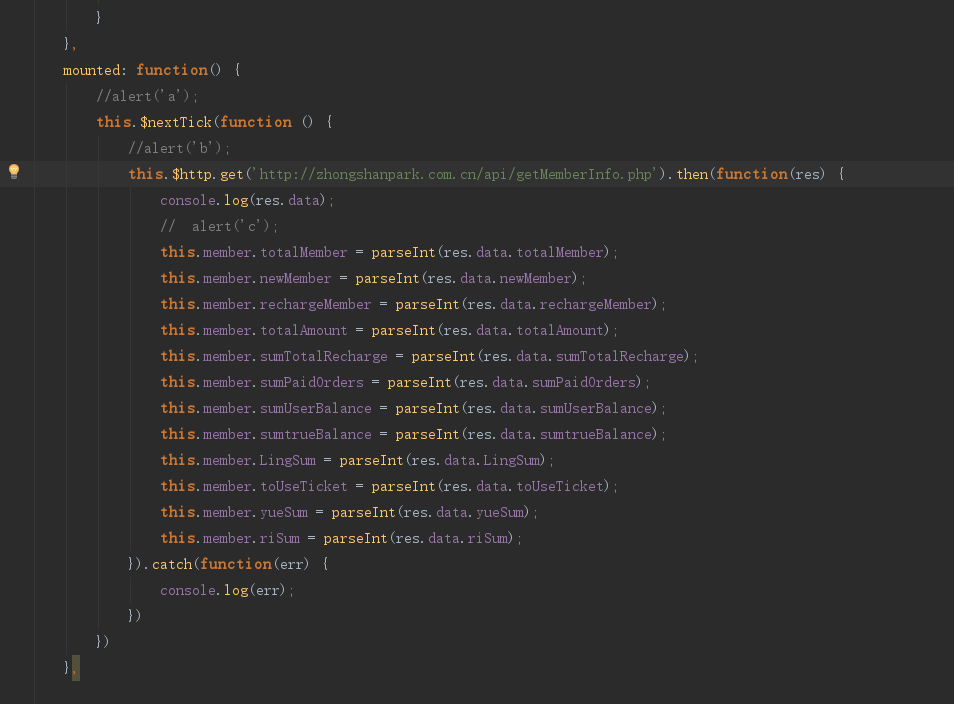


图5.12 概览核心代码

（3）游客管理

点击左侧导航栏中的“游客管理”可以展开该项，其中包含“扫码检票记录”、“会员充值记录”、“单项购票记录”、“会员退票申请”、“会员退票审核”，如图5.13所示：

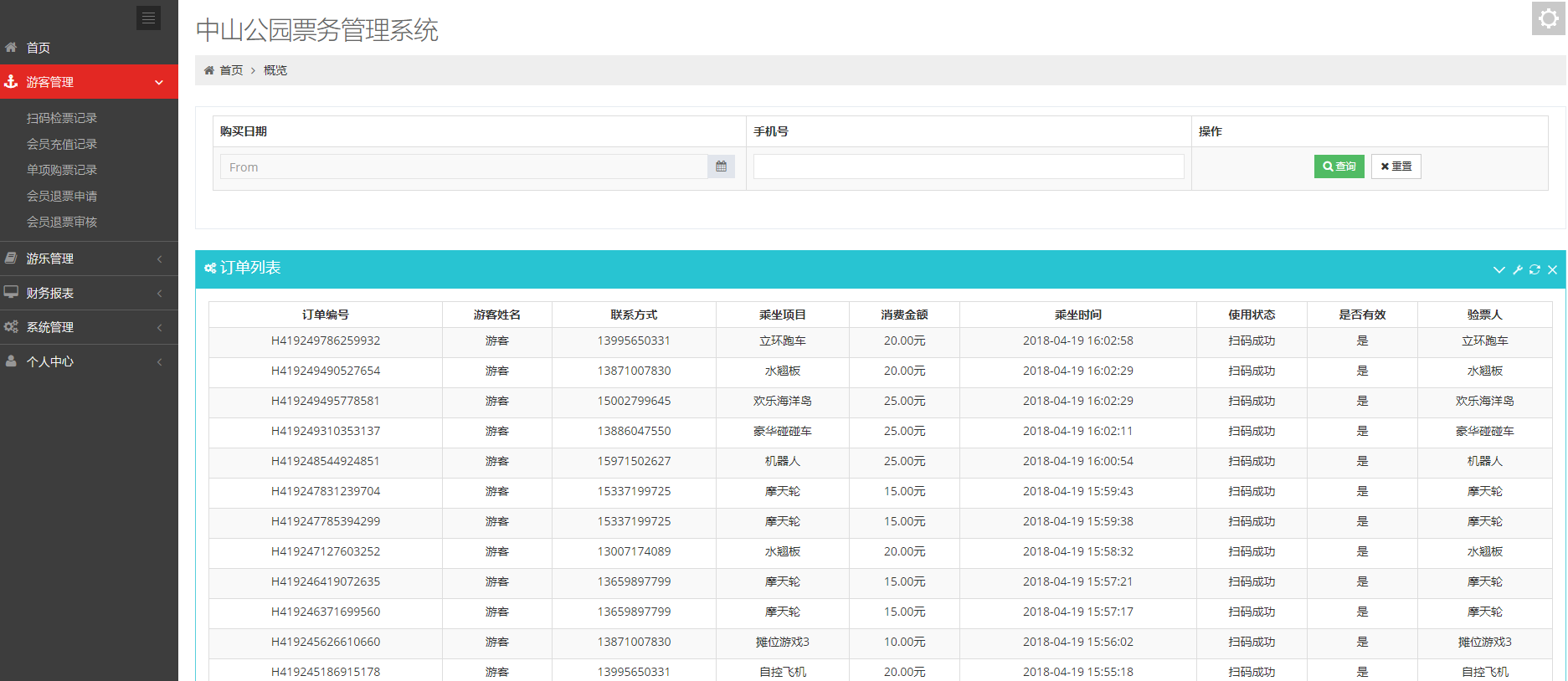


图5.13 游客管理

点击“扫码检票记录”进入相应页面，在列表中包含“订单编号”、“游客姓名”、“联系方式”、“乘坐项目”、“消费金额”、“乘坐时间”、“使用状态”、“是否有效”、“验票人”，在此页面可以通过游客手机号对相关的扫码检票记录进行查询。

点击“会员充值记录”进入相应页面，在列表中包含“微信交易号”、“游客姓名”、“联系方式”、“实际充值金额”、“赠送金额”、“充值日期”、“充值时间”、“充值状态”，在此页面可以通过游客手机号对相关的会员充值记录进行查询。

点击“单项购票记录”进入相应页面，在列表中包含“订单编号”、“游客姓名”、“联系方式”、“购买日期”、“购买项目”、“购买数量”、“购买单价”、“购买金额”、“使用状态”、“使用或退款时间”、“微信交易号”，在此页面可以通过游客手机号对单项购票记录进行查询。

点击“会员退票申请”进入相应页面，在列表中包含“订单编号”、“游客姓名”、“联系方式”、“游乐项目”、“价格”、“购买日期”、“订单状态”、“是否有效”、“验票人”、“操作”，操作中可以点击“申请退票”按钮对门票进行退票申请，此门票的状态变成“退票中”，在此页面可以通过游客手机号对会员退票申请记录进行查询。

点击“会员退票审核”进入相应页面，在列表中包含“订单编号”、“游客姓名”、“联系方式”、“游乐项目”、“价格”、“购买日期”、“订单状态”、“是否有效”、“操作”，操作中可以点击“同意退票”按钮对门票进行退票审核，点击后订单的状态变成“退票成功”，在此页面可以通过游客手机号对退票审核记录进行查询，此部分核心代码如图5.14所示：

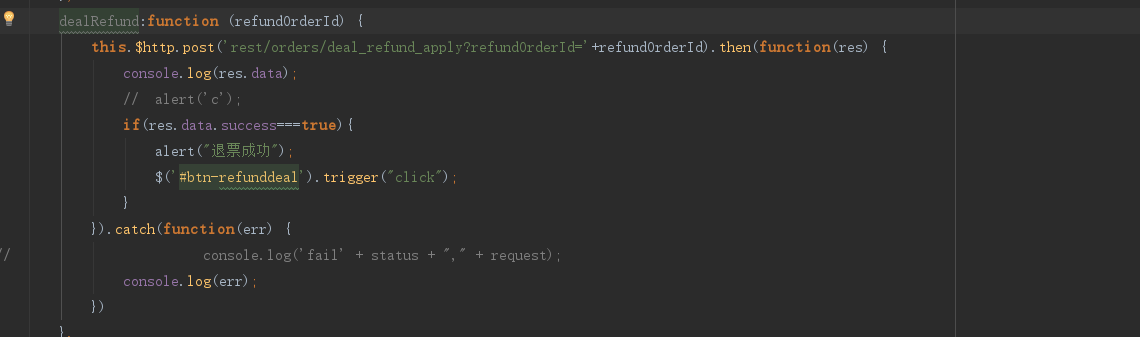


图5.14 退票审核核心代码

（4）游乐管理

点击左侧导航栏中的“游乐管理”可以展开该项，其中是公园的游乐项目列表，具体包括“项目编号”、“项目名称”、“单价”、“运行状态”、“操作”，如图5.15所示：

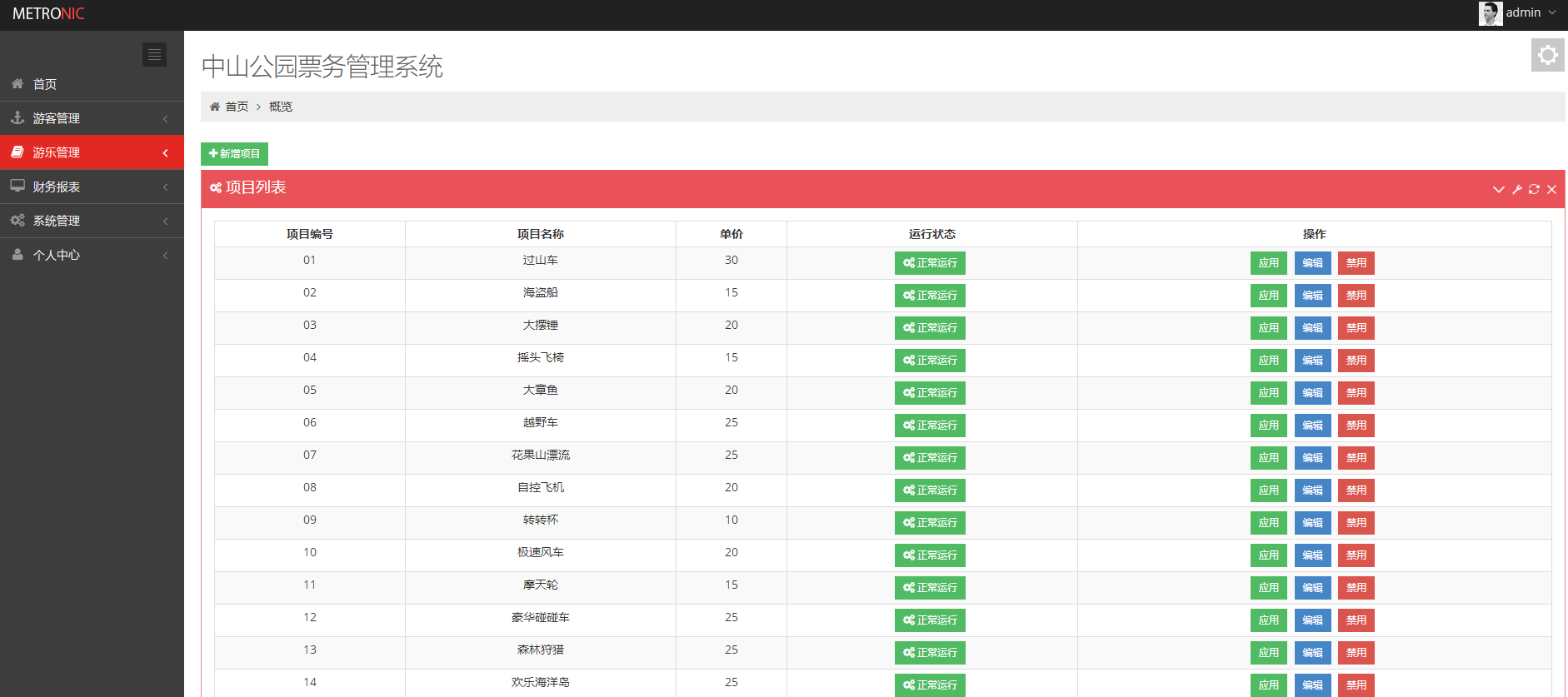


图5.15 游乐管理

（5）财务报表

点击左侧导航栏中的“财务报表”可以展开该项，其中包括“日报表”、“周报表”、“月报表”，以时间为维度对每个游乐项目的收入做出统计。

（6）系统管理

点击左侧导航栏中的“系统管理”可以展开该项，其中包括“用户管理”、“权限管理”。“用户管理”中展示了用户列表，其中包括“用户姓名”、“微信唯一码”、“电话”、“性别”、“余额”等信息。“权限管理”可以用来修改管理员的权限。

（7）个人中心

点击左侧导航栏中的“个人中心”进入该页面，在个人中心通过输入“当前密码”、“新密码”、“确认新密码”点击“保存更改”按钮，可以进行此系统中密码的修改。

## 5.4 验票子系统的实现

验票子系统在安卓手持扫码器上运行，主要用于扫描游乐项目的二维码门票进行验票或扫描会员卡二维码进行游乐项目扣款，验票子系统可以自动检测是否为最新版本，如果不是最新版本可以选择更新。验票子系统主要使用OkHttp+Retrofit和RxJava。OkHttp是一个高效的HTTP库，Retrofit是一个可以用于Android和java的网络库，两者结合使用可提升网络请求的效率以及安全性，同时也简化了对网络请求操作的工作。RxJava是一个在Java VM上使用可观测的序列来组成异步的、基于事件的程序的库。核心是Observables（被观察者，事件源）和Subscribers（观察者）。Observables发出一系列事件，Subscribers处理这些事件，对于复杂的业务逻辑可以采用链式编程方式，清晰明了。

### 5.4.1 硬件设备的介绍

验票人子系统使用到的硬件设备是安卓手持扫码器，又被称为安卓手持终端[63]，主要用于识别门票二维码，通过4G网络与服务端进行通信，具有重量轻、体积小、响应速度快、续航时间长等优点[64]，主屏幕尺寸为4.7英寸，支持蓝牙和无线功能，使用的系统是安卓4.4版本。如图5.16所示：



图5.16 安卓手持扫码器

### 5.4.2 软件部分的实现

（1）新版本更新

完成验票子系统的下载和安装后，点击打开后系统会自动检测新版本。如果不是最新版本，会弹出“发现新版本”的提示框建议下载更新，验票员可以选择“待会更新”或“现在更新”，点击“现在更新”按钮后会自动完成软件版本更新。如图5.17所示：

图5.17 新版本更新

（2）登录

由于验票子系统是内部系统，并不开放注册，提前发放相应的验票员账号，在界面中输入工号和密码，点击“登录”按钮，进入此系统。如图5.18所示：

图5.18 登录

（3）扫码验票

验票员登录进入验票子系统后，显示的界面是他管辖的游乐项目。点击“扫码验票”，安卓手持扫码器会对二维码进行识别。此处验票有两种方式，第一种是识别该游乐项目的门票二维码，扫码成功后该门票的状态变成“已使用”；第二种是识别游客的会员卡二维码，扫码成功后对会员卡进行该游乐项目相应金额的扣款。扫码验票成功后弹出提示，并且实时更新界面上的“会员消费人数”和“零散购票人数”，这两项数据每天零点清零。如图5.19所示：

图5.19 扫码验票

扫码部分的核心代码如图5.20所示：

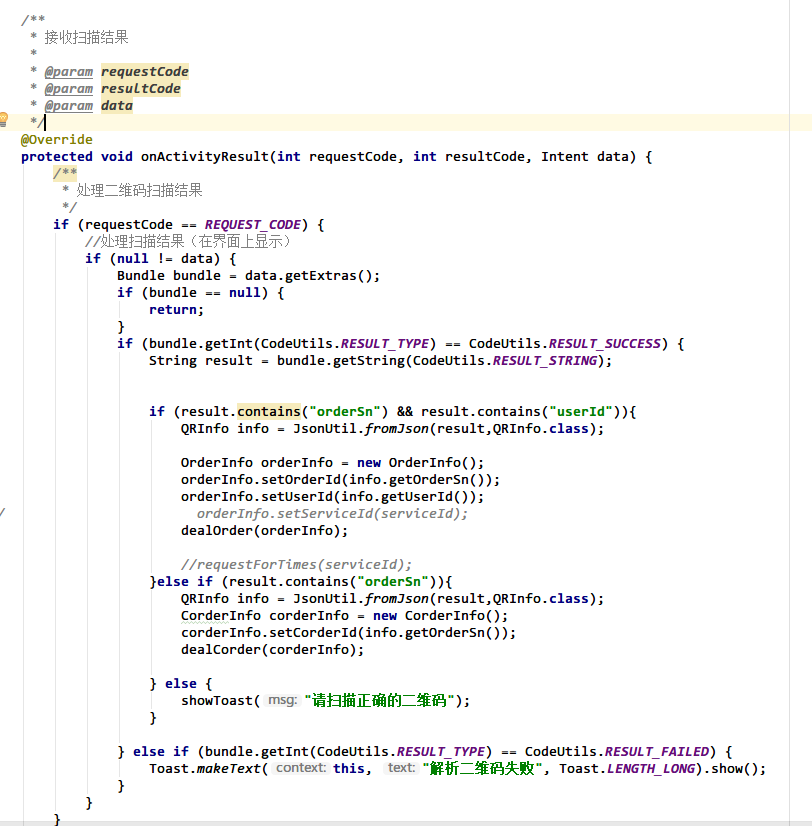


图5.20 扫码部分核心代码

# 第六章 结束语

本章是结束语，分为总结和展望两个部分。全文总结部分对软件架构驱动此公园票务系统的设计与实现的过程进行了总结，展望部分对下一步可以进行的工作进行了合理展望，主要包括将此公园票务系统全国化、增加亲情会员卡功能、无感支付、对收集到的信息进行数据挖掘进行个性化推荐。

## 6.1 全文总结

在互联网浪潮的影响下，公园的传统售票方式急需升级，建立公园票务系统可以方便游客去公园游玩、方便公园员工进行验票、方便公园管理层实时监测公园数据，具有很大的现实意义。

本文对本课题的研究背景和研究意义做了介绍，对国内外票务系统的现状进行分析，介绍了此次公园票务系统使用到的关键技术，包括QRcode技术、Bootstrap框架、Mybatis等等，重点对Web Service进行了介绍，本系统通过Web Service技术，使公园票务系统具有跨编程语言、跨运行平台的特点，很好地在系统中实现了数据共享和信息流通。由于软件架构能带来很多好处，比如提前识别和减轻风险，本文进行了软件架构驱动的公园票务系统的设计与实现。因为经典的软件架构设计方法ADMEMS具有的突出优点，本文按照ADMEMS方法的步骤，依次从预架构阶段、概念架构阶段、细化架构阶段，使用了多种思考工具和表达工具，完成了公园票务系统的架构设计。最后，本文介绍了公园票务系统的实现，其中包括系统的数据库设计、游客子系统的实现、后台管理子系统的实现、验票子系统的实现。

公园票务系统实现了武汉市中山公园票务的互联网化，使整个公园的管理和运营更加方便和规范。信息化的管理对于公园领导层进行科学的决策有着难以忽视的重要意义，在线票务系统有助于推动中山公园的票务整体水平发展到一个高的层次，同时促进其经济收益和形象提升。

## 6.2 展望

本文完成了软件架构驱动的公园票务系统的设计与实现，其中包括游客子系统、后台管理子系统、验票子系统。除了已经完成的工作，本文对下一步可以进行的工作进行展望：

（1）将此公园票务系统全国化。由于全国的中山公园运营方式和游乐项目比较类似，此公园票务系统可以经过迭代，使其适用于其他地区的中山公园或其他类似的景区，让更多公园的票务管理走向互联网化，对其经济收益产生正面影响。

（2）增加亲情会员卡功能。全家一人办卡充值，其他亲属通过绑定会员卡就可以享受会员特权和账户中的会员余额。让游客充值时没有后顾之忧，让全家更多的人可以享受在中山公园游玩的乐趣。

（3）无感支付。目前此公园票务系统主要通过二维码完成购票和验票的操作，在一定程度上给予了游客和验票员方便。但是无感支付可以更进一步，游客在进入游乐项目点时，保证手机上的无感支付开关是打开的状态，通过公园安装的感应器就可以无需手动操作，实现自动扣款、自动验票[65]。

（4）对相关信息进行数据挖掘。公园票务系统可以收集到游客的基本信息、行为数据、订单评价数据、充值数据等，可以对这些信息进行数据挖掘，通过用户画像等方式分析数据，对游客进行精准的个性化推荐。

# 参考文献

1. 林克, 罗喧, 李凌,等. 支持高并发处理的新型票务应用系统设计与实现[J]. 电信科学, 2015, 31(10):130-136.
2. Al-Ghawi S S, Hussain S A, Rahbi M A A, et al. Automatic toll e-ticketing system for transportation systems[C]// Mec International Conference on Big Data and Smart City. IEEE, 2016:1-5.
3. Tattenbaum J, Andrews D, Miglin R, et al. Ticketing systems and methods incorporating interactive seating charts[J]. 2017.
4. Qian H E. Research on the design of electronic passenger ticket system based on RFID[J]. Electronic Design Engineering, 2017.
5. 邱鹏, 万怀宇, 林友芳,等. 基于用户行为分析的在线订票系统缓存优化策略研究[J]. 计算机与现代化, 2017(5):92-98.
6. 周晓燕, 王岩, 寿璐芸,等. 电子支付方式在杭州地铁票务系统中的实现[J]. 都市快轨交通, 2017, 30(3):115-118.
7. 刘香萍, 赵雪岩, 焦宁. 基于Struts框架的网上订票系统的设计与实现[J]. 计算机工程与设计, 2007, 28(20):5060-5062.
8. Leicher A, Kuntze N, Schmidt A U. Implementation of a Trusted Ticket System[J]. Emerging Challenges for Security Privacy & Trust, 2009:152-163.
9. Finzgar L, Trebar M. Use of NFC and QR code identification in an electronic ticket system for public transport[C]// International Conference on Software, Telecommunications and Computer Networks. IEEE, 2011:1-6.
10. Liu Z, Zeng Q, Wang C, et al. Application Research of QRCode Barcode in Validation of Express Delivery[M]// Advanced Research on Electronic Commerce, Web Application, and Communication. Springer Berlin Heidelberg, 2011:346-351.
11. Xuhua W U, Xiang F U. The Design of Office Assets Management System Base on QRCODE[J]. Software Engineer, 2014.
12. Yu R, Duan X, Jiao B. Design and implement of mobile equipment management system based on QRcode[C]// 2017:012036.
13. Lin P Y. Distributed Secret Sharing Approach With Cheater Prevention Based on QR Code[J]. IEEE Transactions on Industrial Informatics, 2016, 12(1):384-392.
14. Nixon R. Learning PHP, MySQL, and JavaScript[J]. Oreilly Media, 2016.
15. Efron B. Bootstrap Methods: Another Look at the Jackknife[J]. Annals of Statistics, 1979, 7(1):1-26.
16. Johnson R W. An Introduction to the Bootstrap[J]. Teaching Statistics, 2001, 23(2):49–54.
17. Volder K D. JQuery: A Generic Code Browser with a Declarative Configuration Language[M]// Practical Aspects of Declarative Languages. Springer Berlin Heidelberg, 2006:88-102.
18. Andreasen E. Determinacy in static analysis for jQuery[C]// ACM International Conference on Object Oriented Programming Systems Languages & Applications. ACM, 2014:17-31.
19. Chaffer J, Swedberg K. Learning jquery: better interaction design and web development with simple javascript techniques[M]. Packt Publishing, 2007.
20. Bollella G, Gosling J. The Real-Time Specification for Java[J]. COMPUTER -LOS ALAMITOS-, 2000, 25(2):78-+.
21. 袁绪峰. 基于Spring框架的AOP编程[J]. 计算机与现代化, 2006(1):118-120.
22. 胡启敏, 薛锦云, 钟林辉. 基于Spring框架的轻量级J2EE架构与应用[J]. 计算机工程与应用, 2008, 44(5):115-118.
23. Ho C. Using MyBatis in Spring[M]// Pro Spring 3. 2012:397-435.
24. Reddy K S P, Reddy K S P, Reddy K S P. Working with MyBatis[M]// Beginning Spring Boot 2. 2017.
25. Zhang D, Wei Z, Yang Y. Research on Lightweight MVC Framework Based on Spring MVC and Mybatis[C]// Sixth International Symposium on Computational Intelligence and Design. IEEE, 2014:350-353.
26. Gulcu C. Short introduction to log4j[J]. Giornale Italiano Di Cardiologia, 2002, 11(12):1883-94.
27. Liu W Z, Tao Q Y, He Q, et al. Application of Log4j in E-Commerce Services[J]. Applied Mechanics & Materials, 2014, 635-637:1517-1521.
28. Surhone L M, Tennoe M T, Henssonow S F. Log4j[M]. Betascript Publishing, 2010.
29. Ankolekar A, Burstein M, Hobbs J R, et al. DAML-S: Web Service Description for the Semantic Web[C]// The, International Semantic Web Conference. 2002:348-363.
30. Liu Y, Ngu A H, Zeng L Z. QoS computation and policing in dynamic web service selection[C]// International Conference on World Wide Web - Alternate Track Papers & Posters, WWW 2004, New York, Ny, Usa, May. DBLP, 2004:66-73.
31. 程炜, 杨宗凯, 乐春晖. 基于Web Service的一种分布式体系结构[J]. 计算机应用研究, 2002, 19(3):105-107.
32. Martens A. Analyzing web service based business processes[J]. 2005, 3442:19-33.
33. Al-Masri E, Mahmoud Q H. Discovering the best web service[C]// 2007:1257-1258.
34. 王映辉. 大规模软件构架技术[M]. 科学出版社, 2003.
35. 李作, 刘宽. 基于SOA软件架构的企业应用[J]. 工程技术:文摘版:00311-00311.
36. Shaw M, Garlan D. Garlan, D.: Software Architecture: Perspectives on an Emerging Discipline. Prentice Hall[J]. For Intelligence Automating Software Design Lowry & Mccartney Editors, 1996, 24(1):129-132(4).
37. Buschmann F, Meunier R, Rohnert H, et al. Pattern-Oriented Software Architecture - Volume 1: A System of Patterns[C]// Wiley Publishing, 1996:41.
38. Bosch J. Design & Use of Software Architecture[C]// Technology of Object-Oriented Languages and Systems, 1999. Proceedings of. DBLP, 2000:404-404.
39. Medvidovic N, Taylor R N. Software architecture: foundations, theory, and practice[C]// ACM/IEEE, International Conference on Software Engineering. IEEE, 2011:471-472.
40. Jansen A, Bosch J. Software Architecture as a Set of Architectural Design Decisions[C]// Working Ieee/ifip Conference on Software Architecture. IEEE Computer Society, 2005:109-120.
41. Cao S, Sun W, Hou R, et al. The spatio- temporal information service architecture of desertification monitoring in China based on the ADMEMS: Design and verification[J]. Journal of Arid Land Resources & Environment, 2015.
42. 郭彧. ADMEMS方法在D-Partner中心软件平台架构设计中的应用实践[C]// 吉林省科学技术学术年会. 2012.
43. 巩佳知. 基于ADMEMS的架构方法在电子政务中的应用[D]. 电子科技大学, 2011.
44. 温昱. 一线架构师实践指南[M]. 电子工业出版社, 2009.
45. 马海银. 智能集群渲染管理软件分析与设计[D]. 山东大学, 2013.
46. 何少壮. 基于集中模式的新一代证券交易系统架构设计及委托子系统的实现[D]. 电子科技大学, 2011.
47. 孟小峰, 周龙骧, 王珊. 数据库技术发展趋势[J]. 软件学报, 2004, 15(12):1822-1836.
48. Christiane Fellbaum (editor, Lin D. Review of: WordNet: An Electronic Lexical Database[J]. Computational Linguistics, 1998, 25(2):292-296.
49. Allen F H. The Cambridge Structural Database: a quarter of a million crystal structures and rising[J]. Acta Crystallographica Section B-structural Science, 2002, 58(3-1):380-388.
50. Zsu M T, Valduriez P. Principles of distributed database systems[J]. Springer Berlin, 2011, 24(1):31-41.
51. Kim O S, Cho Y J, Lee K, et al. Introducing EzTaxon-e: a prokaryotic 16S rRNA gene sequence database with phylotypes that represent uncultured species.[J]. Int J Syst Evol Microbiol, 2012, 62(Pt 3):716-721.
52. 陈丽丽. 大数据时代的数据库技术应用之我见[J]. 信息系统工程, 2016(6):83-83.
53. Bell C, Kindahl M, Thalmann L. MySQL High Availability: Tools for Building Robust Data Centers[M]. O'Reilly Media, Inc. 2010.
54. 姜承尧. MySQL技术内幕:SQL编程[M]. 机械工业出版社, 2012.
55. 李现艳, 赵书俊, 初元萍. 基于MySQL的数据库服务器性能测试[J]. 核电子学与探测技术, 2011, 31(1):48-52.
56. Stanescu L, Brezovan M, Burdescu D D. Automatic mapping of MySQL databases to NoSQL MongoDB[C]// Computer Science and Information Systems. IEEE, 2016.
57. 吴庆祥. PHP书城系统用户模块的设计与实现[J]. 电脑知识与技术, 2016, 12(30):45-46.
58. 龙浩. PHP语言进阶和高级应用[M]. 清华大学出版社, 2002.
59. 孙增国, 杨柳. 基于PHP语言的操作系统教学网站的开发[J]. 长春大学学报, 2015(2):108-115.
60. Eckel B. Thinking in Java[M]. China Machine Press, 2007.
61. Downey A B, Mayfield C. Think Java: How to Think Like a Computer Scientist[M]. O'Reilly Media, Inc. 2016.
62. Martinez M, Durieux T, Sommerard R, et al. Automatic repair of real bugs in java: a large-scale experiment on the defects4j dataset[J]. Empirical Software Engineering, 2017, 22(4):1-29.
63. 李芝浩, 朱尔隆. 基于DSP技术的二维码条码扫描器的硬件设计[J]. 电子制作, 2014(11):14-15.
64. 黄宇航. 基于图像识别的移动设备无障碍QR码扫描软件的研究与实现[D]. 浙江大学, 2016.
65. 杨舒, 方莉, 刘坤. “我们的世界将因人工智能而改变”——第四届世界互联网大会嘉宾共话人工智能未来[J]. 河南科技, 2017(23):5-6.

# 致谢

两年的时间很短，这么快就又到了毕业季。在我攻读硕士学位的这两年间，衷心感谢我的导师，给我们提供了良好的实验室环境和在实践中学习的机会。感谢师兄，作为过来人给了我们很多学习和生活上的指导。感谢同学，一路上互帮互助。感谢我的父母，为我提供了良好的家庭氛围，让我自由的成长。

最后，感恩我的母校武汉大学，东湖之滨，珞珈山上，在这里的六年，遇到的人和事都很美好。