

# 韩山师范学院

# 学生毕业论文

(2019 届)

题目 (中文) 公共运动场地预订系统的设计与实现

(英文) Design and implementation of public sports  
venue reservation system

学院: 计算机与信息工程学院

专业: 计算机科学与技术 (物联网) 班级: 20151152

姓名: 黄舒凯 学号: 2015115087

指导教师: 杨圣云 副教授

企行业指导教师: 杨海波 软件工程师

韩山师范学院教务处制

# 诚信声明

我声明，所呈交的毕业论文是本人在老师指导下进行的研究工作及取得的研究成果。据我查证，除了文中特别加以标注和致谢的地方外，论文中不包含其他人已经发表或撰写过的研究成果，我承诺，论文中的所有内容均真实、可信。

毕业论文作者签名：

签名日期： 年 月 日

**摘要：**随着人们日益增长的运动健身需求及国家推行全面健身的发展战略，公共运动场地在城市空间中所占的比重越来越大。传统的场地管理模式已不能满足人们的需求。基于上述背景，论文结合社会公共运动场地预订的实际需要，设计与实现了公共运动场地预订系统。

系统采用 IntelliJ IDEA 作为软件开发工具，以 JAVA 作为主要的开发语言，使用 MySQL 搭建系统的数据库，是一个使用 SpringMVC + Spring + Mybatis 三大开源框架为基础进行开发的 Java Web 应用程序。

该系统能对运动场地做出有效分配，实现了场地的预订情况信息的展示，提供在线预订功能，以及对于场地信息，用户信息及预订信息的操作管理等。通过建立公共运动场地预订系统，可以对运动场地预订做出有效分配，为人们预订运动场地提供便利。

**关键词：**SSM 框架；运动场地预订；B/S 架构；系统设计

**Abstract:** With the increasing demand for sports and fitness and the national development strategy of comprehensive fitness, the proportion of public sports venues in urban space is increasing. The traditional site management mode has been unable to meet people's needs. Based on the above background, this paper combined with the actual needs of social public sports venues booking, design a public sports venues booking system.

The system uses IntelliJ IDEA as the software development tool, JAVA as the main development language, uses MySQL to build the database of the system. It is a JAVA Web application developed on the basis of SpringMVC + Spring + Mybatis three open source frameworks.

The system can make effective allocation of sports venues, realize the display of venue booking information, provide online booking function, as well as the operation and management of venue information, user information and booking information. By establishing a reservation system for public sports venues, it is possible to effectively allocate the reservation of sports venues, and provide convenience for people to reserve sports venues.

**Key words:**SSM framework;Sports venue booking;B/S architecture;System design

# 目录

1. 绪论.....	1
1.1 研究背景与意义.....	1
1.2 国内外研究现状.....	1
1.3 论文主要研究内容.....	2
2. 系统相关理论和技术.....	3
2.1 IDEA.....	3
2.2 B/S 体系结构.....	3
2.3 SSM 框架.....	3
2.4 本章小结.....	4
3. 需求分析.....	5
3.1 功能需求分析.....	5
3.2 其他非功能性需求.....	7
3.3 本章小结.....	8
4. 系统设计.....	9
4.1 系统总体设计.....	9
4.2 详细设计.....	11
4.3 数据库设计.....	18
4.4 本章小结.....	21
5. 系统实现与测试.....	22
5.1 系统实现.....	22
5.2 系统测试.....	33
5.3 本章小结.....	35
6. 总结与展望.....	36
6.1 总结.....	36
6.2 展望.....	36
参考文献.....	37
致谢.....	38

# 公共运动场地预订系统

## 1. 绪论

### 1.1 研究背景与意义

#### 1.1.1 研究背景

为建设成为体育强国，提高国民的身体素质，国家大力发展公共体育服务，配备更完善的运动设施，建设了大量的公共运动场地，并鼓励中小型运动场馆，及各种企事业单位和学校体育场地对外开放。

大量公共运动场地出现的同时，也暴露出了部分场地因管理不善或无人管理的问题，导致了部分公共体育设施出现损坏等问题。

在传统的预订场地操作中，需要公众到达现场直接预订，且会因为对场地预订情况的不了解而出现没有场地的情况。并且在传统的场地管理中，都是由人工完成对于场地的使用情况的记录和管理，这需要工作人员能够准确规划管理好场地的使用状况，避免出现信息错误造成的场地使用冲突等情况。这种工作模式不利于工作人员进行管理，也给用户的预订造成不便。建立公共运动场地预订系统，能够有效地解决以上出现的问题。

#### 1.1.2 研究意义

通过该系统，人们可以在网站上直接看到对应场地的预订情况，并通过该系统直接进行运动场地的预约，也可在系统上看到发布的公告，提前规划好自己的时间。

场地的管理人员通过该系统可对用户、场地及订单进行管理操作，提高场地的管理效率，改善服务水准。

通过建立公共运动场地预订系统，可以对运动场地预订做出有效分配，更好地利用公共运动场地资源，改变了原有传统的预订模式，提高了运作效率，为人们预订运动场地提供便利。

## 1.2 国内外研究现状

### 1.2.1 国内现状

现如今国家大力发展公共体育服务，配备更完善的运动设施，建设了大量的公共运动场地，并鼓励中小型运动场馆，及各种企事业单位和学校体育场地对外开放。

在这样的发展趋势下，传统的运动场地预订方式不足以满足现代场地预订的需求。公共体育设施的管理方面仍然有所不足，缺乏系统的管理及便民的预订方式。

现如今在国内线上预订已经成为主流，市场上也出现了许多线上预订系统，存在于各行各业，形式正在不断地进行变革和创新，比较有代表性的有美团和百度糯米等，微信也提供了许多基于第三方的订票服务。

现如今仍有许多公共运动场地缺乏较为系统的管理，许多场地的预订仍需要到现场直接进行预订或进行电话预约等，预约信息不透明往往会造成到现场后发现没有场地可预约等情况，给用户带来诸多不便。

### 1.2.2 国外现状

发达国家在体育场馆的建设及管理方面相对我国要成熟许多。国外在运动场地的建设方面有几个特征：在场地的建设上，场地的数量及人均使用率也比我国高出很多。在场地的维护管理方面也给予足够的支持，延长场地的使用寿命。通过利用体育在经济方面的影响，吸收大量的社会资金，广泛的资金来源通道为场地的建设提供帮助。

发达国家规划社区体育中心，其中很大一部分是设置在各学校内，方便周围的居民及学生使用。同时，为提高运动场地的使用效率，居民通过线上进行场地预订以及场地使用情况的查询，来选择适合自己的场地类型，同时输入想要使用的具体时间。这样一套较为完善的系统能够方便管理人员对运动场地进行更高效的维护及管理。

## 1.3 论文主要研究内容

公共运动场地预订系统采用 B/S 结构，使用 Java 语言进行编写，使用 MySQL 搭建系统的数据库，使用 JSP 动态网页技术、CSS、JavaScript、jQuery 及 bootstrap 前端框架来实现前端页面，是一个使用 SpringMVC+Spring+Mybatis 三大开源框架为基础进行开发的 Java Web 应用程序。

该系统能对运动场地做出有效分配，实现了场地的预订情况信息的展示，提供在线预订功能，以及对于场地信息、用户信息、公告信息及预订信息的操作管理等。利用 Spring 的 ioc(控制反转)原理和 aop(面向切面编程)可以有效的减低系统的耦合度，提高系统的复用性。

该系统主要模块包括：用户注册登录及信息修改模块，场地信息管理模块，场地预订模块。用户可在该系统上进行注册登录，填写修改个人基本信息；查看对应场地的预订情况并根据自己的需求进行线上预订。系统管理人员可在该系统上进行对场地信息、用户信息、公告信息及预订信息等的查看维护管理。

## 2. 系统相关理论和技术

本章主要介绍系统开发需要运用到的理论知识和技术和所采用的开发工具。

### 2.1 IDEA

IDEA 全称 IntelliJ IDEA<sup>[1]</sup>。由 JetBrains 公司设计和开发的一款编辑器，集成了 java 开发所需的环境，对目前来说，IDEA 最适用于 Java 项目的开发。

IDEA 中的代码提示、快捷键等功能大大加快了我们的开发测试的速度，所以很受广大 Java 开发者的喜爱。

### 2.2 B/S 体系结构

该系统运用 B/S 体系结构。该体系结构由数据库管理层，逻辑处理层，用户交互层三个不同的层次组成。在 C/S 体系结构中，客户端通过浏览器运行软件，事务逻辑的处理则由服务器负责。作为一种新的软件系统开发技术，B/S 结构使用越来越先进的万维网技术，通过浏览器运行软件节省了许多开发成本。

### 2.3 SSM 框架

#### 2.3.1 Spring 介绍

Spring 是一种轻量级且开源的框架，通过容器来管理对象的生命周期，控制对象的创建方式<sup>[2]</sup>。服务器端一般采用三层体系架构的方式，其中 Spring 对每个模块都在技术上提供支持，所以它可以和每一层轻松关联。Spring 松耦合、易测试等特性让它适用于任何 Java 应用的开发。使用 AOP 即面向切面编程，使系统高度模块化，每个模块的结构更加清晰，提高了代码的复用性，使系统更易实现和更易维护。

#### 2.3.2 Spring MVC 介绍

Spring MVC 是 Spring 框架的一部分，通过一套 MVC 注解，不用实现任何接口。Spring MVC 和 Struts2 一样都是 MVC 框架，但比 Struts2 更加优秀<sup>[3]</sup>。能进行更加简洁的 web 层的开发及测试，容易和其他框架技术进行集成。

#### 2.3.3 MyBatis 介绍

Mybatis 作为一个数据持久层框架，Mybatis 与 Hibernate 相比，能够配置动态 SQL 并进行优化，所有的 SQL 语句都可以定义在 XML 中，也可以通过注解的方式在接口上实现，便于统一管理和维护，降低了程序的耦合度。



## 2.4 本章小结

本章节主要介绍了该系统在开发过程中需要使用到的技术,系统采用B/S体系结构,运用 MVC 模式和 SSM 框架技术,使用 MySQL 数据库,以 Java 作为核心开发语言,简单介绍了这些技术的内容、特点以及优势等,通过对技术的总结,为系统的开发做好准备。

### 3. 需求分析

#### 3.1 功能需求分析

系统的功能需求可以依照服务对象的不同进行划分，包括用户和管理员。其中，管理员可以进行场地管理、用户管理、预订管理和公告管理等。用户可进行个人信息管理及场地信息的查询与预订<sup>[4]</sup>。

##### 3.1.1 用户管理用例

用户管理模块可实现的具体功能是：用户信息查询、用户信息修改和用户信息删除。用户管理用例图如图 1 所示。

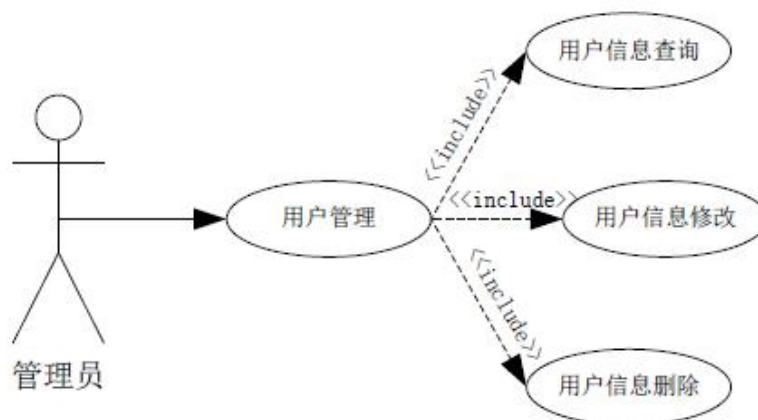


图 1 用户管理用例图

##### 3.1.2 场地管理用例

场地管理模块实现的是管理员对场地信息的查询、添加、修改和删除等操作。场地管理用例图如图 2 所示。

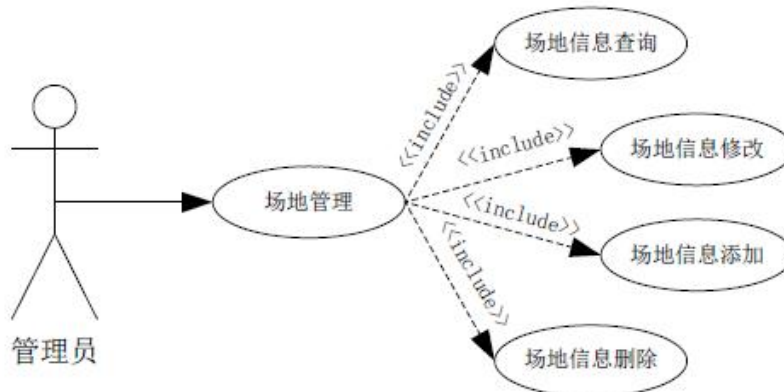


图 2 场地管理用例图

### 3.1.3 预订管理用例

预订管理模块实现的是管理员对预订信息的查询、修改和删除等操作。预订管理用例图如图 3 所示。

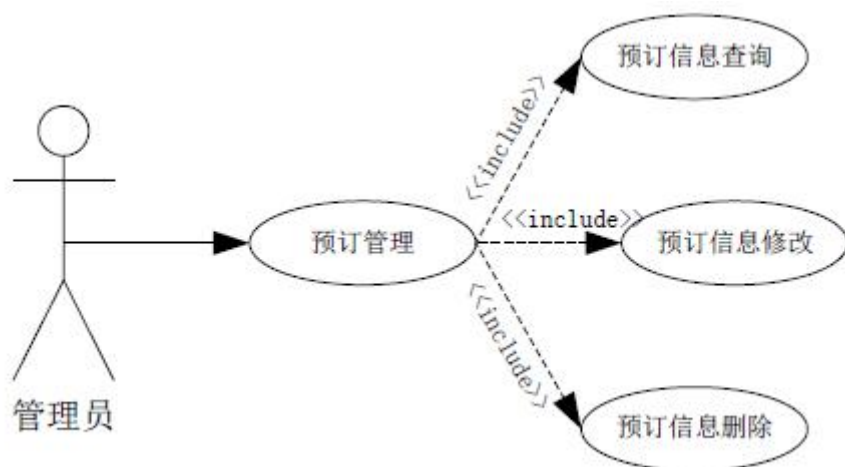


图 3 预订管理用例图

### 3.1.4 公告管理用例

公告管理模块用来发布最新的通知，其内容由管理员进行编辑并保存发布，也可对已发布的公告进行编辑修改和删除。公告管理用例图如图 4 所示。

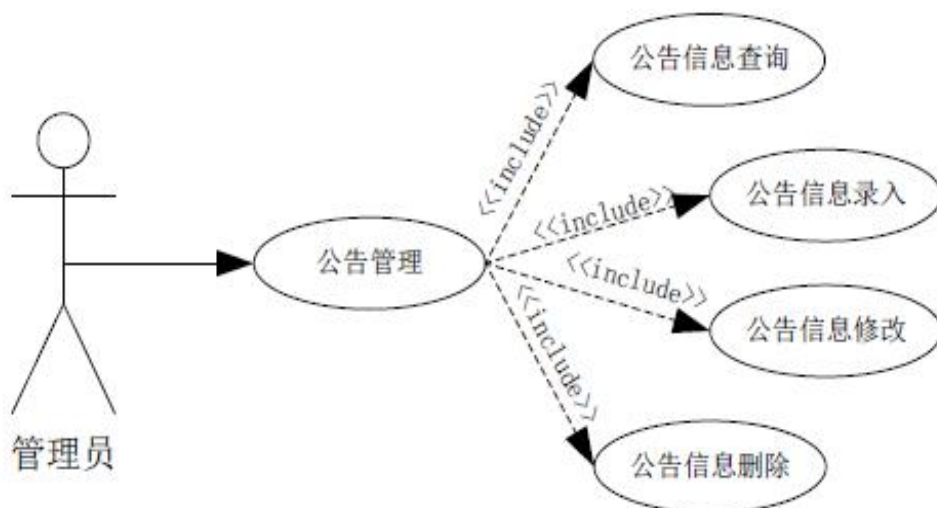


图 4 公告管理用例图

### 3.1.5 系统用例

系统主要向用户提供注册登录、场地信息查询、场地预订、历史预订信息查询和个人信息查询修改等功能。对于用户的系统用例图如图 5 所示。

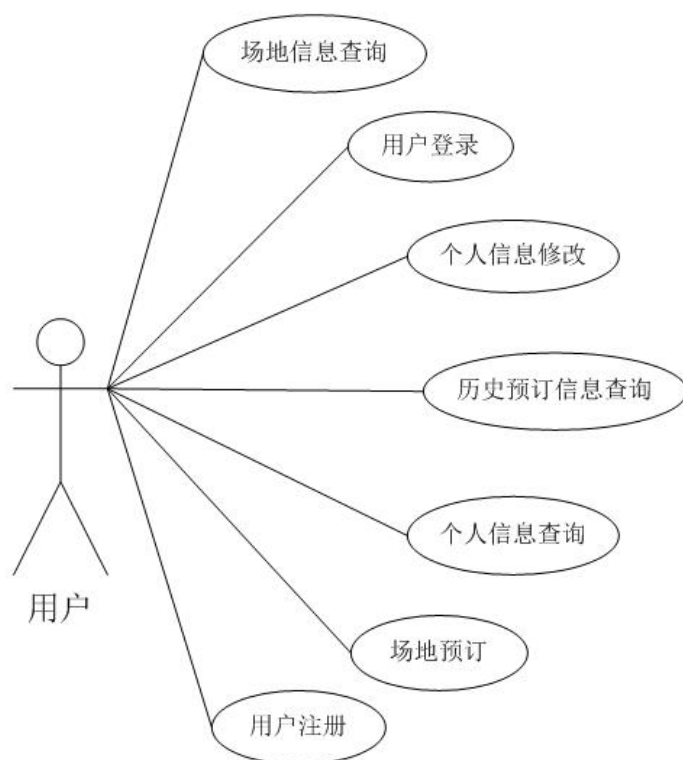


图 5 用户系统用例图

## 3.2 其他非功能性需求

### 3.2.1 性能需求

系统的网络的性能要求非复杂的查询和处理的一般业务响应时间：小于等于 3 秒。

系统平台性能要求具有良好的可扩展性，支持平滑扩容，以满足未来应用和业务系统规模的增加而不断扩展。

### 3.2.2 安全性需求

安全性对系统而言非常重要。系统的所有数据涉及用户的隐私，所以在进行系统操作时均需要进行用户的身份认证，用户在不登录状态下只能进行部分数据的查看，管理员和用户的操作权限应有效地分隔开，这样可有效地进行系统数据的安全管理。同时，管理员要定期做好数据库的备份工作，防止系统发生错误时因数据的丢失造成严重后果，在进行数据恢复的时候可防止因人为操作造成的失误及数据读写的错误。

### 3.2.3 软件质量属性

有效性：用户向系统提交信息后，系统将在 3 秒内显示确认信息。

可维护性：当系统内存不够用时，自动为系统增加 100M 内存，在硬件允许条件下可无限扩展。

可移植性：在系统指定操作系统下可移植。

### 3.3 本章小结

本章利用用例图的方式，结合系统需要解决的问题，对系统进行详细的功能需求分析，再从系统性能和安全性等角度进行系统的非功能性需求分析。为后续的系统设计提供帮助。

4. 系统设计

4.1 系统总体设计

4.1.1 系统架构设计

系统架构图如图 6 所示, 由表示层、应用层以及数据层组合<sup>[5]</sup>。通过浏览器根据用户的操作及接受客户端发送的请求向数据库服务器发送请求, 并接收数据库传递的消息, 然后将其转化成如 html 或 xml 等格式, 发送给浏览器。

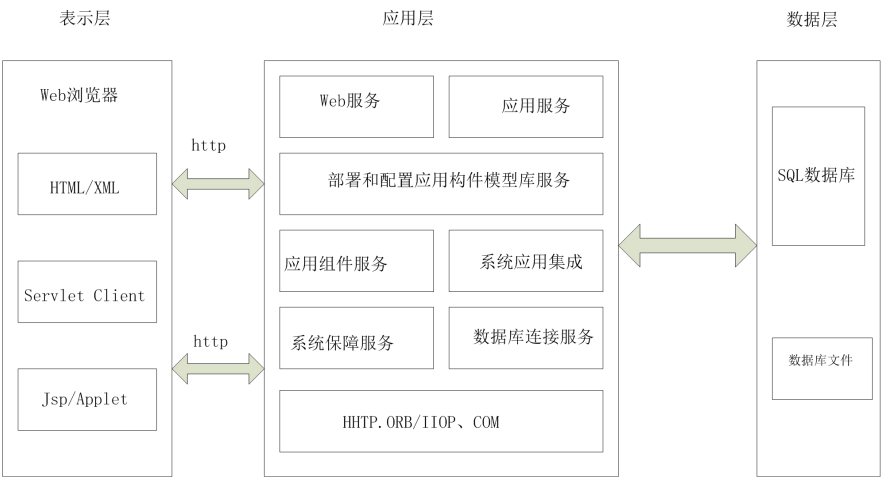


图 6 系统架构图

4.1.2 系统功能结构设计

系统的功能结构图如图 7 所示：

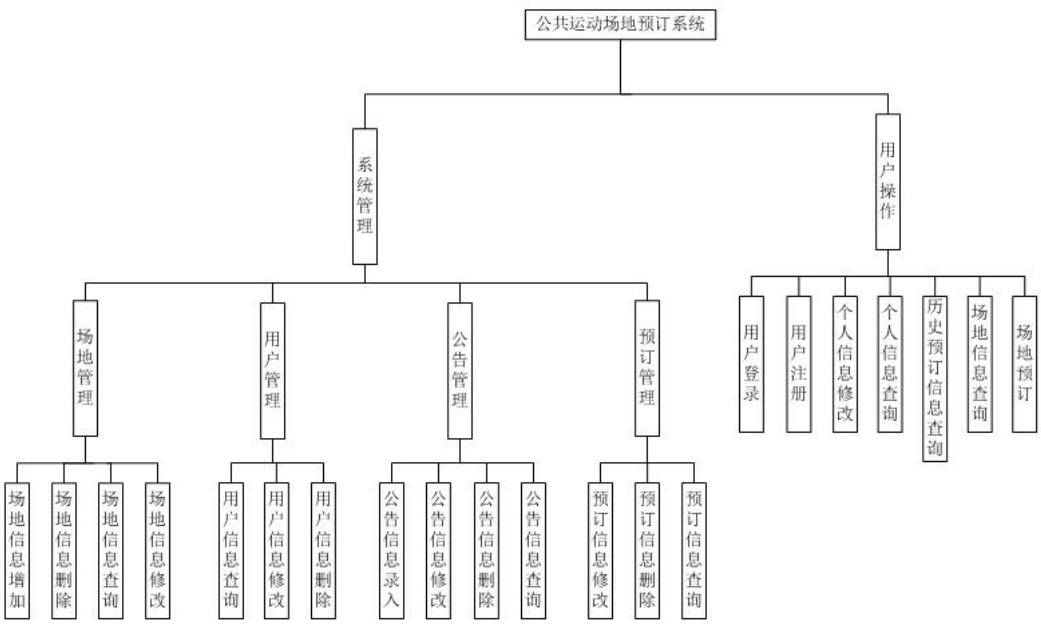


图 7 功能结构图

系统主要分为 2 个模块：用户操作和系统管理

系统管理：该模块的功能主要对于用户、场地和订单信息的管理，对信息的管理包括基本的增删查改。

用户操作：该模块的功能主要是用户的注册登录，对场地信息的查询与预订，个人信息的管理等操作。

### 4.1.3 系统功能设计

#### 4.1.3.1 用户注册

用户需在注册之后才能进行登录及进行预约操作，注册流程包括：个人基本信息填写，手机号填写，获取手机短信验证码并填写，系统进行校验后完成注册。完成注册后可在登录界面使用用户名及密码进行登录。

#### 4.1.3.2 用户登录

用户可在登录界面使用已注册的用户名和密码进行登录。

#### 4.1.3.3 用户个人信息管理

用户在登录后可查看及修改个人信息。

#### 4.1.3.4 历史预订记录查询

用户在登录后可查询个人的已预订历史记录，并按照相关条件进行筛选查询，包括订单创建起始至终止日期及预订起始至终止日期，场地类型等信息。

#### 4.1.3.5 场地预订

对于用户来说，网上预约的流程包括：账号登录，场地查询，场地预约，预约情况显示这几个主要环节。用户在登录该系统后可在首页查看系统公告，可查询各种场地类型的预约情况，根据自己的实际需求进行场地的预约，用户也可在未到达预约时间段之前自行取消已预约的订单。

#### 4.1.3.6 场地信息管理

管理员通过登录后进入系统内部的管理界面，选择对应的场地进行查看、新增、删除和修改等操作。可设置任一场地及时间段为不可预订状态。

场地信息新增：录入运动场地信息，包括场地类型，场地名称及场地位置信息等。

场地信息查看：查看系统的场地信息。

场地信息修改：对运动场地信息的变动进行修改，确保信息的准确性和实时性。

场地信息删除：对不存在或不对外开放的场地信息进行删除。

#### 4.1.3.7 公告信息管理

管理员可通过该功能发布关于场地信息的公告，包括公告的标题、公告内容及发布时间，并在系统首页展示，方便用户查看。可对已发布的公告进行修改删除操作。

#### 4.1.3.8 用户信息管理

管理员在登录系统后可对用户信息进行查询修改和删除等操作。

#### 4.1.3.9 订单信息管理

管理员可根据场地类型，订单创建起始至终止日期及预订起始至终止日期对预订信息进行查询

删除预订信息：管理员可对系统产生的预订信息进行删除。

### 4.2 详细设计

系统详细设计是设计的第二阶段，接下来是对系统的功能模块进行详细设计。

#### 4.2.1 用户注册模块

用户注册模块实现用户注册功能。用户在注册页面进行注册操作。流程为：用户输入用户名，系统判断用户名是否重复，是则提示重新输入用户名。用户设置密码后再次输入密码，系统判断密码是否一致，不则提示重新输入密码。填写完正确的手机号码后点击获取验证码，系统发送短信验证码到所填写的手机，输入对应的验证码完成注册<sup>[6]</sup>。用户注册功能流程图如图 8 所示。



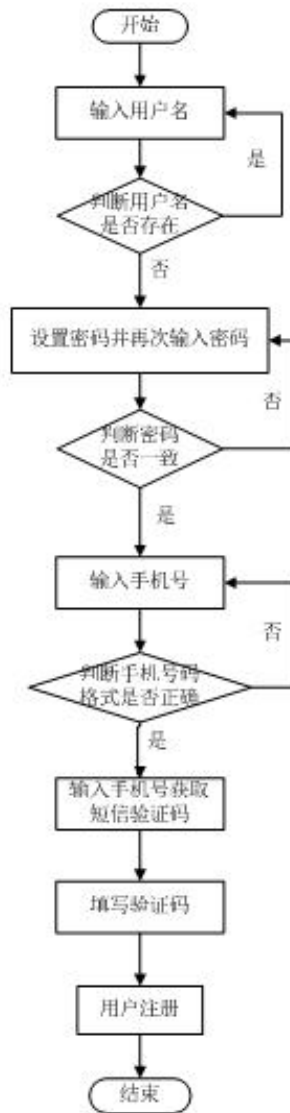


图 8 用户注册功能流程图

#### 4.2.2 用户登录模块

用户注册模块实现用户登录功能。用户在登录页面进行登录操作。流程为：用户输入用户名和密码并点击登录，系统判断用户名和密码是否与数据库的数据对应，不则提示用户重新输入，是则完成登录<sup>[7]</sup>。用户登录功能流程图如图 9 所示。

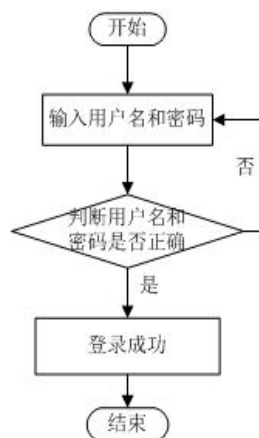


图 9 用户登录功能流程图

### 4.2.3 个人信息管理模块

个人信息管理模块实现用户对个人信息的查看和修改功能<sup>[8]</sup>。操作流程为：用户在登录页面输入用户名和密码完成登录后，点击个人信息进入个人信息页面进行查看，并可在当前页面对个人信息进行修改并提交，系统判断修改后的信息是否正确，不是则提示用户重新输入，是则显示修改后的个人信息。个人信息管理功能流程图如图 10 所示。

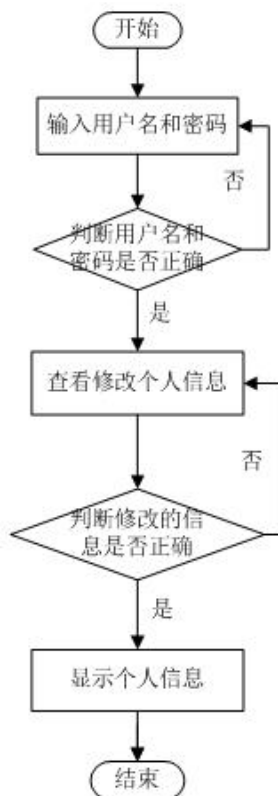


图 10 个人信息管理功能流程图

#### 4.2.4 个人历史预订查询模块

个人历史预订查询模块实现用户对预订历史的查询功能。操作流程为：用户在登录页面输入用户名和密码完成登录后，点击查询历史预订信息进入页面，系统默认显示全部预订信息，用户可在当前页面选择场地类型，订单创建起始至终止日期及预订起始至终止日期等限制条件进行条件查询。个人历史预订查询功能流程图如图 11 所示。

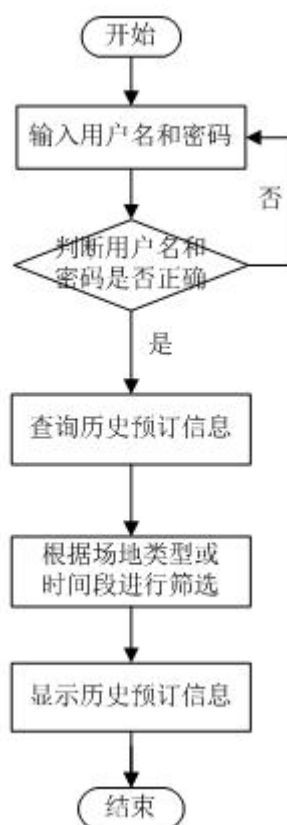


图 11 个人历史预订查询功能流程图

#### 4.2.5 场地预订模块

场地预订模块实现用户对场地的预订功能。操作流程为：用户在登录页面输入用户名和密码完成登录后，点击预订进入场地类型选择页面，选择场地类型进入对应的场地预订界面，系统会默认显示当天该场地的预订情况；用户可在日期栏中选择七天内的任一日期查看当天的预订情况；用户可选择可预订的场地和对应的时间段，点击提交进入确认订单页面，系统会显示订单信息，用户可选择取消按钮则取消预订操作，点击确认按钮则完成预订操作并显示最终的预订信息。

场地预订模块功能流程图如图 12 所示。

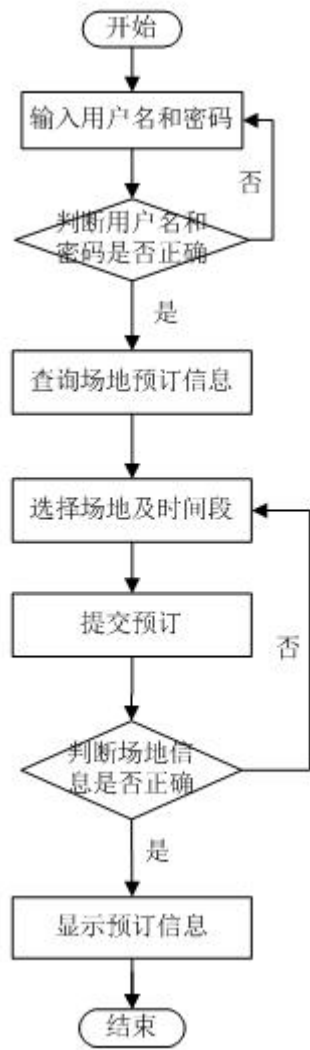


图 12 场地预订模块功能流程图

#### 4.2.6 场地管理模块

场地管理模块实现管理员对场地的管理功能。管理员登录后通过该模块对场地信息进行管理，可及时对场地信息进行修改调整，若场地因其他原因不对外开放时，在系统中可以设置对应场地及时间段为不可预订状态。场地管理模块功能流程图如图 13 所示。

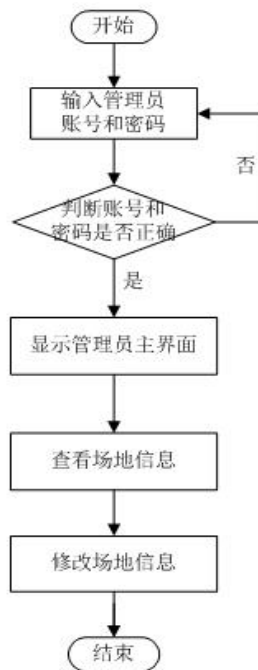


图 13 场地管理模块功能流程图

#### 4.2.7 公告管理模块

公告管理模块实现管理员对公告的管理功能。管理员登录后通过该模块对公告信息进行管理，可发布公告信息，对已发布的公告信息进行修改或删除操作<sup>[9]</sup>。公告管理模块功能流程图如图 14 所示。

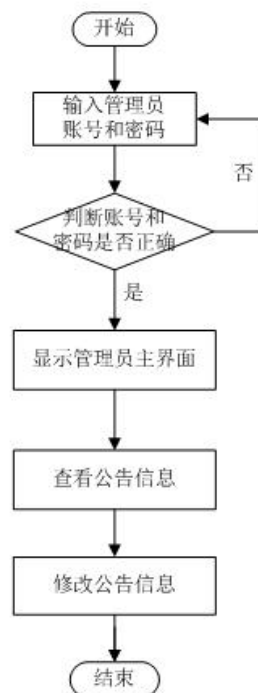


图 14 公告管理模块功能流程图

### 4.2.8 用户管理模块

用户管理模块实现对用户的管理功能。管理员登录后通过该模块对用户信息进行管理，可进行用户信息的修改或删除操作<sup>[10]</sup>。用户管理模块功能流程图如图 15 所示。

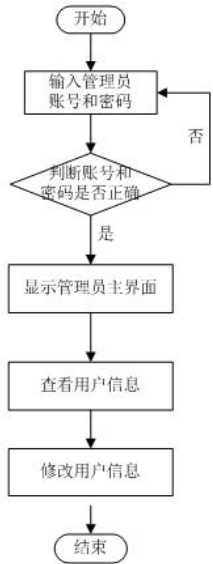


图 15 用户管理模块功能流程图

### 4.2.9 订单管理模块

订单管理模块实现对订单的管理功能。管理员登录后通过该模块对订单信息进行管理，系统默认显示全部订单信息，管理员可在当前页面选择场地类型，预订起始日期到结束日期等限制条件进行条件查询。可进行订单信息的删除操作。订单管理模块功能流程图如图 16 所示。

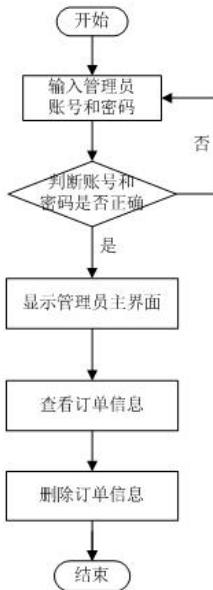


图 16 订单管理模块功能流程图

### 4.3 数据库设计

论文通过结合系统的实际开发需求，分析系统在数据库中的存储要求，转化为数据字典，建立系统实体 E-R 图，最后将 E-R 图转化为数据库的关系模型。

#### 4.3.1 概念结构设计

根据上面可设计规划出系统的实体包括用户、订单、场地、公告和管理员，各个实体具体的描述 E-R 图如下：

用户实体属性 E-R 图，如图 17 所示。

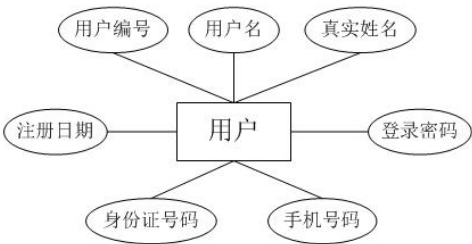


图 17 用户实体属性 E-R 图

订单实体属性 E-R 图，如图 18 所示。

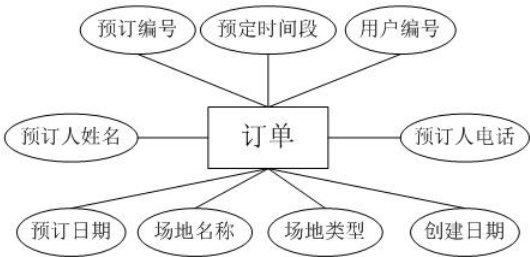


图 18 订单实体属性 E-R 图

公告实体属性 E-R 图，如图 19 所示。

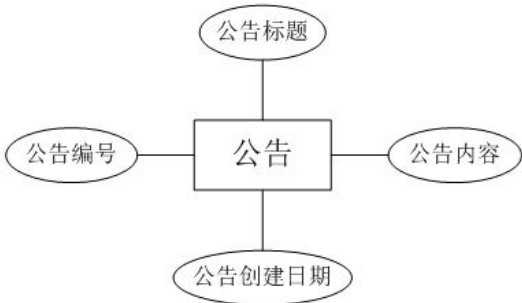


图 19 公告实体属性 E-R 图

场地实体属性 E-R 图，如图 20 所示。

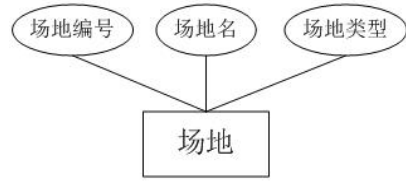


图 20 场地实体属性 E-R 图

各实体间的关系如图 21 所示。

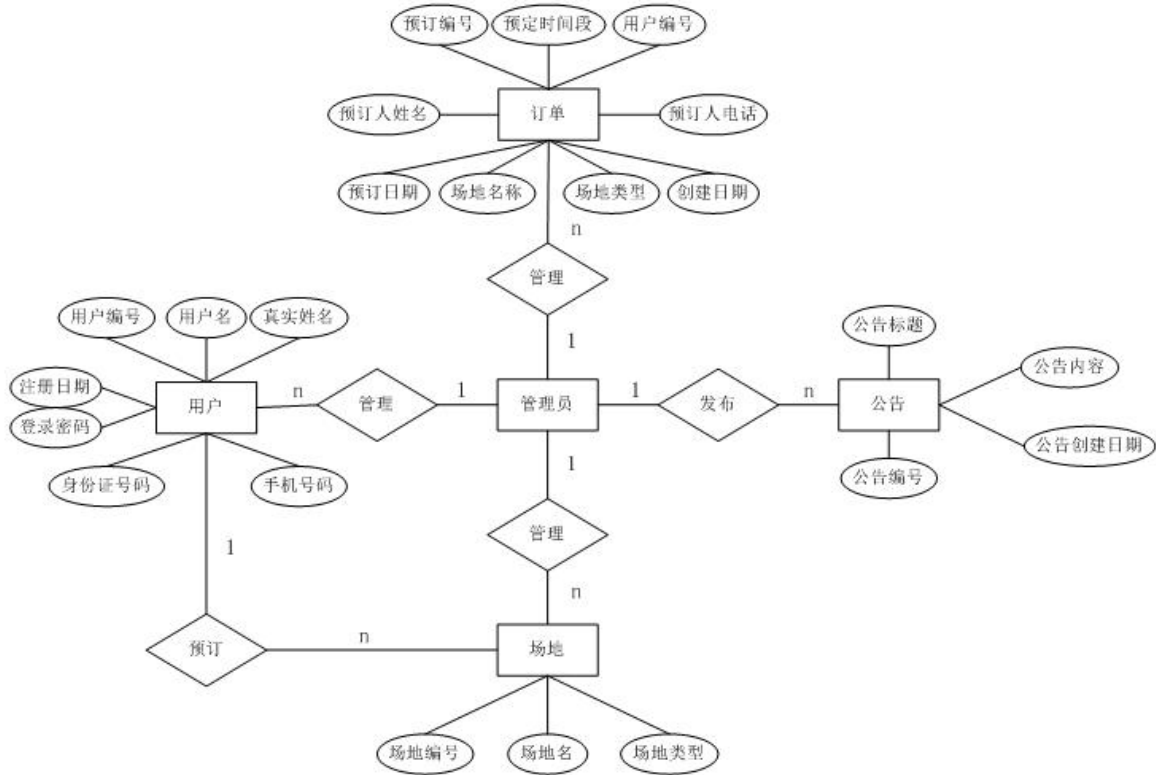


图 21 实体间关系图

### 4.3.2 逻辑结构设计

用户信息表：该表用于记录用户的基本信息，包括：用户编号，用户名，真实姓名，登录密码，手机号码，身份证号码和注册日期。表结构如表 1 所示。



表 1 用户信息表

列名	数据类型	长度	允许 null 值	是否主键	备注
user_id	varchar	255	N	Y	用户编号
username	varchar	255	N		用户名
realname	varchar	255	N		真实姓名
password	varchar	255	N		登录密码
phone	varchar	255	N		手机号码
id_num	varchar	255	N		身份证号码
regdate	date	0	N		注册日期

场地类型信息表：该表用于记录场地类型，包括：类型编号，类型名称。表结构如表 2 所示。

表 2 场地类型信息表

列名	数据类型	长度	允许 null 值	是否主键	备注
category_id	varchar	255	N	Y	类型编号
category_name	varchar	255	N		类型名称

场地预订信息表：该表用于记录场地的预订信息，包括：预订编号，用户编号，预订人姓名，预订人电话，场地类型，场地名称，创建日期，预订日期和预订时间段。表结构如表 3 所示。

表 3 场地预订信息表

列名	数据类型	长度	允许 null 值	是否主键	备注
order_id	varchar	255	N	Y	预订编号
user_id	varchar	255	N		用户编号
realname	varchar	255	N		预订人姓名
phone	varchar	255	N		预订人电话
category_name	varchar	255	N		场地类型
court_name	varchar	255	N		场地名称
create_date	date	0	N		创建日期
booking_date	date	0	N		预订日期
booking_time	varchar	255	N		预订时间段

公告信息表：该表用于记录公告信息，包括：公告编号，公告标题和公告内容。表结构如表 4 所示。

表 4 公告信息表

列名	数据类型	长度	允许 null 值	是否主键	备注
announcement_id	varchar	255	N	Y	公告编号
announcement_title	varchar	255	N		公告标题
announcement_content	varchar	255	N		公告内容

管理员信息表：该表用于记录管理员的基本信息，包括：管理员编号，管理员账号和管理员密码。表结构如表 5 所示。

表 5 管理员信息表

列名	数据类型	长度	允许 null 值	是否主键	备注
manager_id	varchar	255	N	Y	管理员编号
manager_account	varchar	255	N		管理员账号
manager_password	varchar	255	N		管理员密码

篮球场信息表：该表用于记录篮球场地的基本信息，包括：场地编号，类型编号和场地名称。表结构如表 6 所示。

表 6 篮球场信息表

列名	数据类型	长度	允许 null 值	是否主键	备注
court_id	varchar	255	N	Y	场地编号
category_id	varchar	255	N		类型编号
court_name	varchar	255	N		场地名称

网球场信息表：该表用于记录网球场地的基本信息，包括：场地编号，类型编号和场地名称。表结构如表 7 所示。

表 7 网球场信息表

列名	数据类型	长度	允许 null 值	是否主键	备注
court_id	varchar	255	N	Y	场地编号
category_id	varchar	255	N		类型编号
court_name	varchar	255	N		场地名称

羽毛球场地信息表：该表用于记录羽毛球场地的基本信息，包括：场地编号，类型编号和场地名称。表结构如表 8 所示。

表 8 羽毛球场地信息表

列名	数据类型	长度	允许 null 值	是否主键	备注
court_id	varchar	255	N	Y	场地编号
category_id	varchar	255	N		类型编号
court_name	varchar	255	N		场地名称

#### 4.4 本章小结

本章完成系统总体设计和功能模块的详细设计，完成数据库的概念结构设计和逻辑结构设计，为系统后期的实际开发提供支持。

## 5. 系统实现与测试

在上一个章节中已完成了系统的详细设计，本章主要讲述了系统的实现过程。

### 5.1 系统实现

#### 5.1.1 用户登录

用户登录模块用来进行用户登录，只有用户输入正确后才可以完成登录并进行其他操作。用户登录界面如图 22 所示。

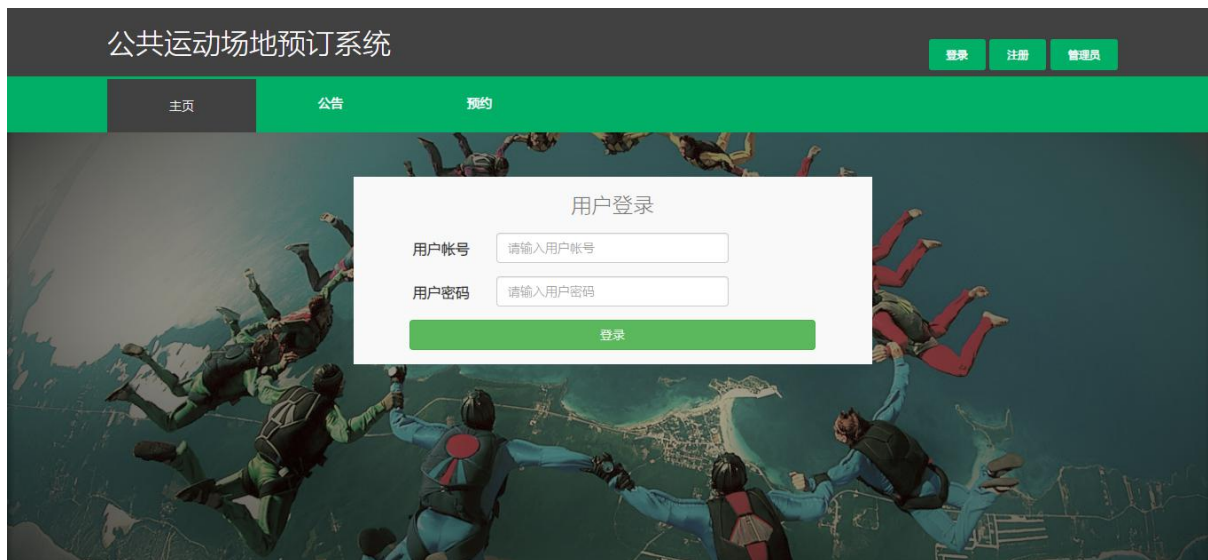


图 22 登录界面图

用户登录模块核心代码如下：

```
@RequestMapping("/login") //HandlerMapping
public String doLogin(@RequestParam String username, @RequestParam String
password, HttpSession session) throws SQLException,
UnsupportedEncodingException, NoSuchAlgorithmException {
    java.util.Map map = new HashMap();
    map.put("username", username);
    map.put("password", Utils.EncoderByMd5(password));
    User user = usi.login(map);
    if(user != null){
        session.setAttribute("user", user);
        return "../../index";
    } else{
        return "/user/loginUI";
    }
}
```

### 5.1.2 场地预约

场地预约模块包括场地查询和预约，场地预约操作实际上是判断传递的各种信息正确后，在数据库添加一条数据，场地查询的界面如图 23 所示。



图 23 场地查询界面图

场地查询核心代码如下：

```
@RequestMapping("/getVenueDeTailByDate")
@ResponseBody
public String doGetVenueDeTailByDate(String category_name, String curYear,
                                     String curMonth, String curDate,
                                     Model model) throws SQLException {
    String bookingDate = curYear + "-" + curMonth + "-" + curDate;
    System.out.println(bookingDate);
    Map map = new HashMap();
    map.put("category_name", category_name);
    map.put("bookingDate", bookingDate);
    List<Booking> list = service.findByProp(map);
    System.out.println(list);
    Map map1 = new HashMap();
    map1.put("success", true);
    map1.put("data", list);
    map1.put("status", 200);
    return JSON.toJSONString(map1);
}
```

场地预约界面如图 24、图 25 和图 26 所示。



图 24 场地预约界面图

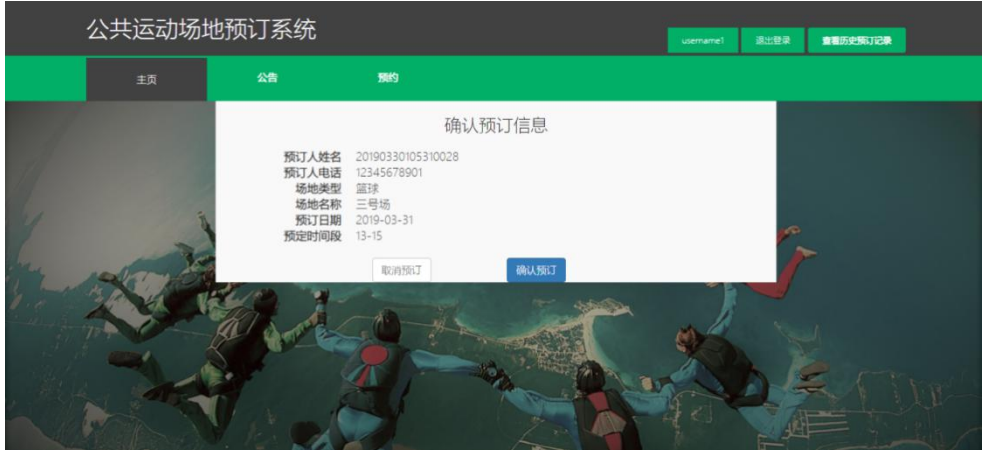


图 25 场地预约界面图

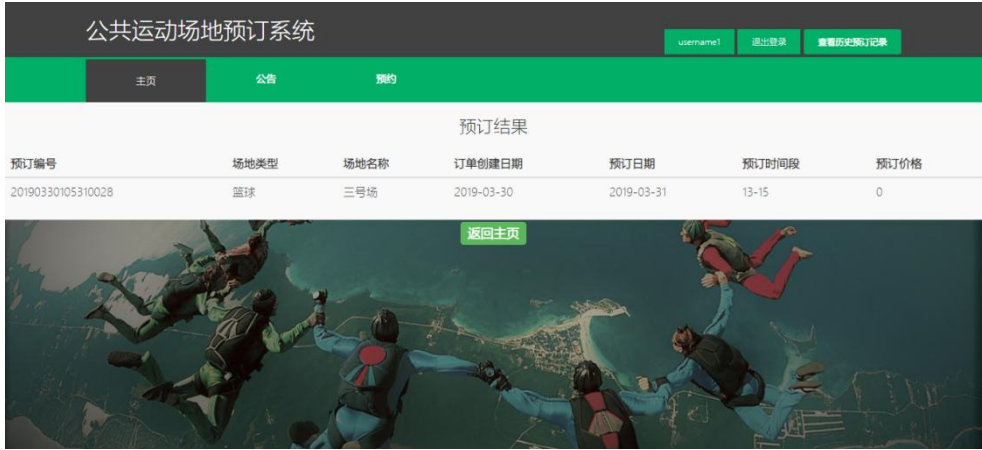


图 26 场地预约界面图

场地预约核心代码如下：

```
@RequestMapping("/add") //HandlerMapping
@ResponseBody
public String doAdd(Booking booking, HttpSession session) throws
```

```

SQLException {
    System.out.println(booking);
//    添加预订用户信息
    User user = (User) session.getAttribute("user");
    booking.setUser_id(user.getUser_id());
    booking.setRealname(user.getRealname());
    booking.setPhone(user.getPhone());
    System.out.println(booking);
//    根据对象的 id 属性的值进行判断, 如果没有, 则添加
    if (service.add(booking)) {
        return "yes";
    } else {
        return "no";
    }
}
}

```

### 5.1.3 预订记录查询

预订记录查询功能的实现界面如图所示。在界面中, 用户可以直观地看到自己的所有预订记录, 也可通过场地类型、订单创建起始至终止日期及预订起始至终止日期等条件进行筛选查询。预订记录查询界面如图 27 所示。

公共运动场地预订系统

username1 退出登录 查看历史预订记录

主页 公告 预订

所属分类: [下拉菜单] 订单创建日期: [年/月/日] 到 [年/月/日] 预订日期: [年/月/日] 到 [年/月/日] 查询

预订编号	场地类型	场地名称	订单创建日期	预订日期	预订时间段	
20180925145901072	网球	一号场	2018-09-25	2018-09-25	9-11	删除
20180925145954084	网球	三号场	2018-09-25	2018-09-28	13-15	删除
20180925150802002	网球	二号场	2018-09-25	2018-09-26	11-13	删除
20181017212451077	网球	二号场	2018-10-17	2018-10-18	15-17	删除
20181018090611090	网球	二号场	2018-10-18	2018-10-18	13-15	删除
20181018090800025	网球	三号场	2018-10-18	2018-10-18	11-13	删除
20190105193745005	网球	一号场	2019-01-05	2019-01-05	11-13	删除

第1/1页 总记录数: 7 每页显示: 10

图 27 预订记录查询界面图

### 5.1.4 用户个人信息管理

该功能用于用户个人的信息管理，用户能查看个人信息，也可对信息进行修改。用户个人信息管理功能的实现界面如图 28 所示。

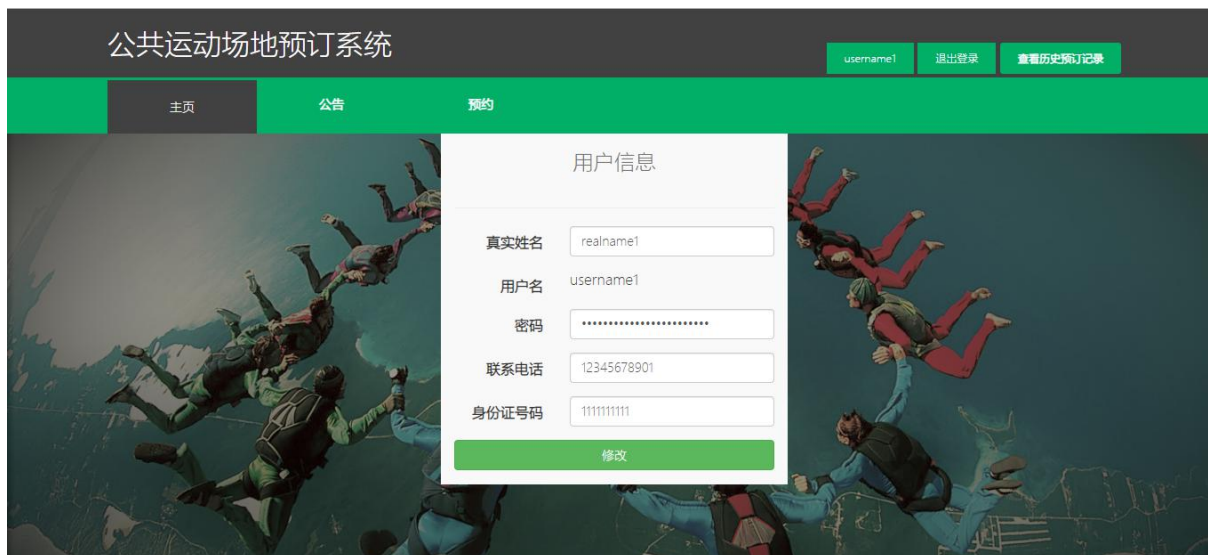


图 28 个人信息管理界面图

用户个人信息管理功能核心代码如下：

```
@RequestMapping("/getData") //HandlerMapping
@ResponseBody
public String getData() throws SQLException {
    Map queryMap = new HashMap();
    List<User> list = usi.findByProp(queryMap);
    Map map = new HashMap();
    map.put("data", list);
    return JSON.toJSONString(map);
}
```

### 5.1.5 管理员登录和权限判断

该功能用于管理员登录，只有输入正确的账号和密码才能进入系统管理界面。为了系统的安全性，该系统通过拦截器对请求进行权限判断和登录状态判断，如果管理员账号不处于登录状态，进入系统管理界面会自动跳转到管理员登录界面。

管理员登录界面如图 29 所示。



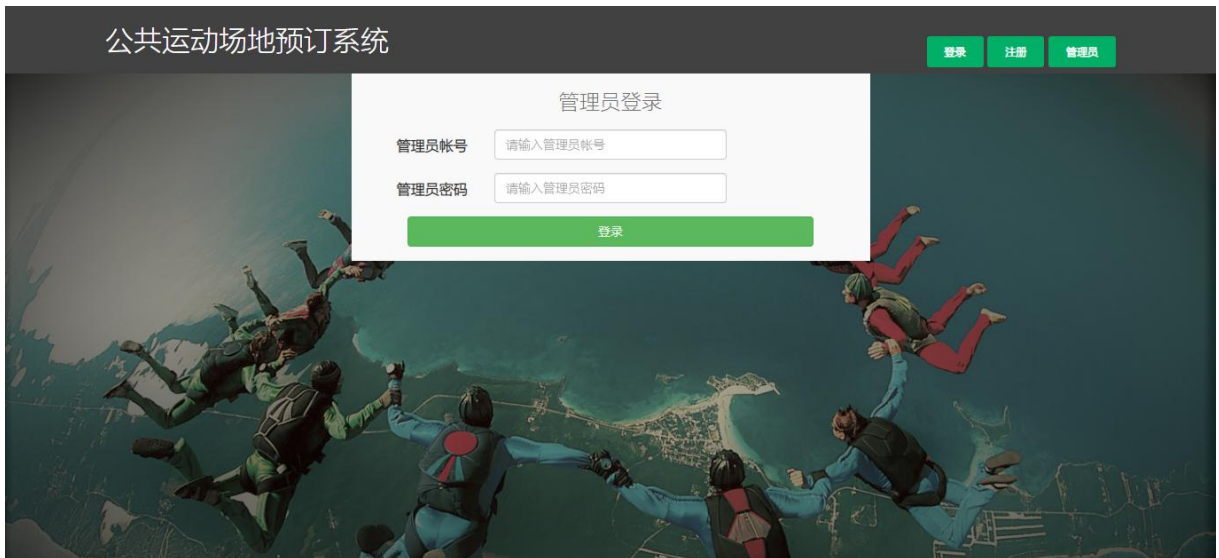


图 29 管理员登录界面图

管理员登录功能核心代码如下：

```
@RequestMapping("/mLogin") //HandlerMapping
public String doMLogin(@RequestParam String manager_account,
    @RequestParam String manager_password, HttpSession session) throws
    SQLException,                               UnsupportedEncodingException,
    NoSuchAlgorithmException {
    java.util.Map map = new HashMap();
    map.put("manager_account", manager_account);
    map.put("manager_password", manager_password);
    Manager manager = msi.login(map);
    if(manager != null){
        session.setAttribute("manager", manager);
        return "/manager1/manage_main";
    } else{
        return "/manager1/login";
    }
}
```

权限判断功能核心代码如下：

```
public class LoginInterceptor implements HandlerInterceptor{
    @Override
    public boolean preHandle(HttpServletRequest request, HttpServletResponse
response, Object handler) throws Exception {
```



```

        System.out.println(request.getServletPath());
        HttpSession session = request.getSession();
        Manager manager=(Manager)session.getAttribute("manager");
        System.out.println("manager:"+manager);
        if(manager==null){
            response.sendRedirect(request.getContextPath()+
"/manager1/toMLogin");
            return false;
        }else{
            return true;
        }
    }
}

```

### 5.1.6 用户信息管理

管理员在登录系统后可对用户信息进行操作。

用户管理界面如图 30 所示。



图 30 用户管理界面图

查询功能核心代码如下：

```

@RequestMapping("/getData") //HandlerMapping
@ResponseBody
public String getData() throws SQLException {
    Map queryMap = new HashMap();
    List<User> list = usi.findByProp(queryMap);
}

```

```

Map map = new HashMap();
map.put("data", list);
return JSON.toJSONString(map);
}

```

删除功能核心代码如下：

```

@RequestMapping("/delete/{user_id}")
public String doDelete(@PathVariable String user_id) throws SQLException {
    usi.delete(user_id);
    return "redirect:/user/manage";
}

```

修改功能核心代码如下：

```

@RequestMapping("/toEdit/{id}") //HandlerMapping
public String FindById(@PathVariable String id, Model model) throws SQLException
{
    User user = usi.findOneById(id);
    model.addAttribute("user", user);
    return "user/editUI";
}

```

### 5.1.7 场地信息管理

该功能需要管理员通过登录后才能进入系统内部的管理界面，选择对应的场地进行查看、新增、删除和修改等操作，也可设置禁止网上预约的时段。

场地信息管理界面图如图 31 所示。

图 31 场地信息管理界面图

场地信息查看：查看系统的场地信息。功能核心代码如下：

```
@RequestMapping("/manage") //HandlerMapping
public String doManage(Model model) throws SQLException {
//    查找 basketball_court 对象
    Map queryMap = new HashMap();
    List<Basketball_court> list = bsi.findByProp(queryMap);
//    查找 basketball_court 所在分类对象
    queryMap.put("category_name", "篮球");
    List<Category> list1 = csi.findByProp(queryMap);
    Map map = new HashMap();
    map.put("success", true);
    map.put("data", list);
    map.put("cdata", list1);
    map.put("status", 200);
//    将数据添加到域对象
    model.addAttribute("content", map);
    return "/basketball_court/manage";
}
```

场地信息新增及修改：录入运动场地信息，包括场地类型，场地名称及场地位置信息等。对运动场地信息的变动进行及时的修改，确保信息的准确性和实时性。功能核心代码如下：

```
@RequestMapping("/doUpdate")
public String doUpdate(Basketball_court basketball_court) throws
SQLException {
    System.out.println(basketball_court);
    if (basketball_court.getCourt_id().equals("")) {
        bsi.add(basketball_court);
    } else {
        bsi.update(basketball_court);
    }
    return "redirect:/basketball_court/manage";
}
```

场地信息删除：对不存在或不对外开放的场地信息进行删除。功能核心代码如下：

```
@RequestMapping("/delete/{court_id}")
public String doDelete(@PathVariable String court_id) throws SQLException
{
    bsi.delete(court_id);
    return "redirect:/basketball_court/manage";
}
```

### 5.1.8 预订信息管理

该功能用于管理员在登录后可根据场地类型，订单创建起始至终止日期及预订起始至终止日期对预订信息进行条件查询；也可对预订信息进行删除。

预订信息管理界面图如图 32 所示。

图 32 预订信息管理界面图

删除功能核心代码如下：

```
@RequestMapping("/deleted/{order_id}")
public String doDeleted(@PathVariable String order_id) throws SQLException
{
    service.delete(order_id);
    return "redirect:/booking/booking_history_findByProp";
}
```

### 5.1.9 公告信息管理

管理员在登录后可通过该功能发布关于场地信息的公告，包括公告的发布时间，标题及公告内容，并在系统首页展示，方便用户查看。可对已发布的公告进行修改删除操作。公告信息管理界面图如图 33 所示。



图 33 公告信息管理界面图

公告删除功能核心代码如下：

```
@RequestMapping("/delete/{announcement_id}")
public String doDelete(@PathVariable String announcement_id) throws
SQLException {
    asi.delete(announcement_id);
    return "redirect:/announcement/announcement_manage";
}
```

公告更新或添加功能核心代码如下：

```
@RequestMapping("/doUpdate")
public String doUpdate(Announcement announcement) throws SQLException {
    System.out.println(announcement);
    if (announcement.getAnnouncement_id().equals("")) {
        asi.add(announcement);
    } else {
        asi.update(announcement);
    }
}
```

```

        return "redirect:/announcement/announcement_manage";
    }

```

公告查看功能核心代码如下：

```

@RequestMapping("/getannoutcement") //HandlerMapping
@ResponseBody
public String doGetannoutcement(Model model) throws SQLException {
    java.util.Map queryMap = new HashMap();
    List<Announcement> list = asi.findByProp(queryMap);
    System.out.println(list);
    java.util.Map map = new HashMap();
    map.put("success", true);
    map.put("data", list);
    map.put("status", 200);
    return JSON.toJSONString(map);
}

```

## 5.2 系统测试

在完成开发后，需要进行一系列的测试，以确保系统在运行中不会出现重大的错误，避免系统崩溃的情况发生。测试的主要目的就是为了尽可能发现程序中存在的错误，并找到解决的方法，只有不断地测试发现问题并解决问题才能够不断地提高系统的性能。

### 5.2.1 功能性测试

系统的功能性测试一般采用黑盒测试，即关注系统的功能能否实现。针对系统功能测试，测试内容包括几个方面，如表 9 所示：

表 9

测试内容表

功能模块	测试用例	预期结果	测试结果
用户注册	填写已存在的用户名	提示用户名已存在	通过
	填写密码并在“确认密码”栏内填写不同的密码	提示密码不匹配	通过
	填写密码并在“确认密码”栏内填写相同的密码	提示密码匹配	通过
	填写正确的手机号码并点击获取验证码	系统发送短信验证码至填写的手机号	通过
	输入错误的验证码并点击注册	提示验证码错误	通过
	填写所有必填的个人信息并输入正确的验证码并点击注册	提示注册成功	通过
	填写正确的用户名和密码并点击登录	登录成功并跳转至主页	通过
用户登录	填写错误的用户名和密码并点击登录	提示登录失败	通过
用户个人信息修改	点击主页的用户名按钮	跳转至个人信息页面并显示用户个人信息	通过
	修改用户个人信息并点击提交	刷新页面显示修改后的信息	通过
查看历史预订记录	点击主页的查看历史预订记录按钮	跳转至用户预订历史信息页面并显示所有预订信息	通过
	根据条件对预订信息筛选搜索	展示符合条件的信息	通过
	点击场地类型进入相应的场地预订界面	跳转至相应的场地预订页面并显示系统当天的预订情况	通过
	在未登录状态下点击选择可选的时间段并提交	提示账号未登录	通过
场地预订	在已登录状态下不选择时间段点击提交	提示未选择任何时间段	通过
	在已登录状态下点击选择可选的时间段并提交	跳转至确认预订信息页面并显示预提交的信息	通过
	点击确认预订	跳转至用户预订信息页面并显示预订结果	通过
	点击取消预订	跳转至预订页面	通过
系统管理	在管理主页点击订单管理	跳转至订单管理页面并显示所有订单信息	通过
	根据条件对订单信息筛选搜索	显示符合条件的信息	通过

5.2.2 非功能性测试

系统主要进行浏览器兼容性测试。测试用例如表 10 所示：

表 10

测试用例表

测试用例名称	系统浏览器兼容性测试
测试目的	在 IE、Chrome、火狐、360 等主流浏览器下运行系统是否正常使用
测试用例级别	兼容性测试
测试流程	(1) 测试人员使用 IE 浏览器打开系统主页，进行登录完成场地预约操作 (2) 测试人员使用 Chrome 浏览器打开系统主页，进行登录完成场地预约操作 (3) 测试人员使用火狐浏览器打开系统主页，进行登录完成场地预约操作 (4) 测试人员使用 360 浏览器打开系统主页，进行登录完成场地预约操作
预期结果	不同的浏览器中页面布局显示正常，能够成功地完成场地预约操作
实际结果	不同的浏览器中页面显示正常，功能正常，达到兼容性测试的标准

5.3 本章小结

本章阐述了系统各个功能模块的具体实现，通过实现代码详细介绍实现过程，并展示了系统的实现界面，让每个功能模块能够愈加直观具体地展现出来，最终完成了对系统的开发。并对系统进行了功能性测试与非功能性测试，验证了系统的有效性，表明系统能够满足场地预订管理的需求，达到了课题设计的要求。



## 6. 总结与展望

### 6.1 总结

公共运动场地预订系统能够科学并规范地实现公共运动场地资源的管理和使用。该系统的实现为群众提供了便利，发挥了运动场地的应用价值，改变了原有传统的预订模式，提高了运作效率。

本文讲述了公共运动场地预订系统的设计开发过程，通过分析社会上公共运动场地的使用管理情况所存在的问题，采用实地调查的方式，分析用户对于预订运动场地的需求及需要解决的问题，对系统的需求进行分析，并通过市场上主流的预订系统，有针对性的对该系统的需求进行改进，并根据使用者进行角色划分，确定了每个角色的业务流程，再完成一系列的设计并最终完成系统开发，最后完成论文的编写工作。

### 6.2 展望

本文设计开发的基于 SSM 框架的公共运动场地预订系统已经能基本满足当前场地的预订管理要求。但由于开发时间短，需求分析不够完善的原因，在系统的具体实现技术和功能上，还有许多可以继续完善和改进的地方。包含以下几个方面：

- 1、系统的稳定性还需要加强，访问量增大到一定程度时可能会出现高并发量所带来的一些问题，系统能否可靠运行还有待检验。
- 2、随着数据量的增大，对数据的查询效率也会产生一定考验，需要在后期维护过程中对 SQL 语句进行优化来提高数据库的查询性能。
- 3、在数据传输安全性及数据库的安全性方面还有待进一步改善，对传输和存储的数据采用加密算法，以提高系统的保密性和安全性。

## 参考文献

- [1]张心雨. 基于 php 的文体中心场地预约系统设计与实现[D·硕士学位论文]. 南京大学. 2015.
- [2]王世光. 企业体育场馆管理信息系统的设计与实现[D·硕士学位论文]. 电子科技大学. 2013.
- [3]刘宗斌. 高校体育场馆管理系统的研究与分析[D·硕士学位论文]. 云南大学. 2015.
- [4]戴佩迪. 体育场馆预订管理系统设计与实现[D·硕士学位论文]. 天津大学. 2015.
- [5]马聪. 哈尔滨商业大学体育场馆管理系统设计与实现[D·硕士学位论文]. 北京工业大学. 2015.
- [6]林晓红. 高等学校体育场馆设施管理系统设计与实现[D·硕士学位论文]. 吉林大学. 2016.
- [7]龙浩, 王文乐, 刘金等. 软件工程——软件建模与文档写作[M]. 北京:人民邮电出版社, 2016. 45-46.
- [8] (美) 弗兰纳根 (Flanagan, D. ). JavaScript 权威指南 (原书第 6 版) [M]. 北京:机械工业出版社, 2012. 54-59.
- [9]Jaewon Oh, Woo Hyun Ahn, Taegong Kim. MVC Architecture Driven Restructuring to Achieve Client-Side Web Page Composition[A]. In:Proceedings of 2016 IEEE 7th International Conference on Software Engineering and Service Science (ICSESS 2016) [C].BeiJing:Institute of Electrical and Electronics Engineers ( IEEE ), 2016. 45-53.
- [10]王导利. 基于 C/S 架构的健身运动场馆预定系统设计[J]. 自动化与仪器仪表, 2017, 第 3 期 (总第 209 期): 121-123.

## 致谢

时光飞逝，四年的大学本科生涯即将结束，回首这段经历，在人生最美好的年华遇上韩师，即幸运也美妙。在此毕业论文即将完成之际，我谨向所有关心、爱护、帮助我的老师和同学们表示最真挚的感谢与最美好的祝愿！

本文是在杨圣云导师的细心指导下完成的。从毕业设计的选题到论文的撰写过程给予我很多意见，使我在整个项目进行过程中及论文的写作中都获益匪浅。在此对杨老师给予的帮助表达最真挚的感谢。感谢计算机与信息工程学院的老师们为我们提供了这么好的学习机会，能够让我的知识水平和学术水平提升到一个新的高度。他们对我们的谆谆教诲让我们感受到了学者风范和行为师表的楷模作用，我从中收获很多。

另外，也要感谢我的同学和好友，是他们让我的大学生活丰富多彩，大家真诚以待，相互帮助和学习，共同进步，让我度过了一个十分难忘、快乐和充实的大学本科生活。

最后，我要感谢我的家人，是他们在背后默默的支持着我，让我能够在遇到困难的时候能够勇敢面对，给予我勇往直前、永不放弃的动力。

黄舒凯

2019年5月1日