****

**本科生毕业设计（论文）**

|  |  |
| --- | --- |
| **题 目：** | **基于Web的汽车一对一驾驶陪练** |
|  | **预约系统的设计与实现** |
| **学院（系）：** | **计算机学院** |
| **专业班级：** | **软件工程1903班** |
| **学生姓名：** | **王荣超** |
| **指导教师：** |  |
| **评 阅 人：** |  |
| **完成时间：** | **2023 年 5 月 4日** |



**毕业设计（论文）任务书**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 题 目 | 基于Web的汽车一对一驾驶陪练预约系统的设计与实现 | | | | | |
| 学生姓名 | 王荣超 | 学号 | | 201907070525 | 专业班级 | 软件工程1903 |
| 设计（论文）内容及基本要求 |  | | | | | |
| 设计（论文）起止时间 | | | 2023年 月 日 至 年 月 日 | | | |
| 设计（论文）地点 | | |  | | | |
| 指导教师签名 | | | 年 月 日 | | | |
| 系（教研室）主任签名 | | | 年 月 日 | | | |
| 学生签名 | | | 年 月 日 | | | |

基于Web的汽车一对一驾驶陪练预约系统的设计与实现

# 摘 要

随着社会的发展和人们生活水平的提高，汽车已经成为了很多家庭的必备交通工具。因此，学习驾驶并取得合法的驾照也成为了很多人迫切需要掌握的技能。目前，我国有很多正规的驾校可以提供专业的理论和实践教学，帮助学员顺利通过考试拿到驾照。然而，通过考试并不意味着就能够安全、自信地在各种复杂的道路环境中驾驶汽车。对于那些刚刚拿到驾照的新手司机来说，他们往往缺乏足够的实战经验和应变能力，面对拥挤、繁忙、多变的道路交通情况时容易感到紧张、恐惧或迷茫。这不仅会影响他们自身的安全和心理健康，也会给其他道路使用者带来潜在的危险。为了解决这一问题，我基于Java Web技术设计并实现了一个一对一驾驶陪练预约系统。

该系统采用的Spring MVC 框架，Web服务器采用Tomcat，数据库采用MySQL。系统设计四类用户分别为：新用户，普通用户，教练用户和驾校管理员用户。通过实名的用户即可在平台上预约驾校中的专业教练，教练陪同用户在真实道路上行驶，针对行驶环境传授给用户一些经验。

该系统可以根据用户的时间、偏好车辆类型，为用户推荐合适的教练和车辆。用户只需通过手机或电脑，简单几步就可以完成预约，无需排队或等待。经过测试分析，该系统可以有效地提高新手司机的预约满意度和学习效果，解决了新手司机在获得驾照初期对真实道路环境不熟悉的问题。

**关键词：**

Spring MVC；Java Web；驾驶陪练；预约系统；JSP

Design and implementation of web-based one-on-one driving sparring booking system for automobiles

# ABSTRACT

With the development of society and the improvement of people's living standards, cars have become a necessary means of transportation for many families. Therefore, learning to drive and obtaining a legal driver's license has also become a skill that many people urgently need to master. At present, there are many formal driving schools in China that can provide professional theoretical and practical teaching to help students successfully pass the test and get a driver's license. However, passing the exam does not mean that you will be able to drive a car safely and confidently in a variety of complex road environments. For novice drivers who have just obtained their licenses, they often lack sufficient practical experience and adaptability, and are prone to nervousness, fear or confusion in the face of congested, busy and changeable road traffic situations. This not only affects their own safety and mental health, but also poses a potential danger to other road users. In order to solve this problem, I designed and implemented a one-on-one driving sparring appointment system based on Java Web technology.

The system uses the Spring MVC framework, the web server uses Tomcat, and the database uses MySQL. The system design four types of users are: new users, ordinary users, instructor users and driving school administrator users. Users with real names can book a professional coach in the driving school on the platform, and the coach accompanies the user to drive on the real road, and imparts some experience to the user for the driving environment.

The system can recommend suitable coaches and vehicles for users based on their time and preferred vehicle type. Users can complete the appointment in a few simple steps through their mobile phone or computer, without queuing or waiting. After testing and analysis, the system can effectively improve the appointment satisfaction and learning effect of novice drivers, and solve the problem that novice drivers are unfamiliar with the real road environment in the early stage of obtaining a driver's license.

**Keywords：**Spring MVC； Java Web； driving sparring; Reservation system; JSP

**目 录**

[摘 要 I](#_Toc135692421)

[ABSTRACT II](#_Toc135692422)

[1 绪论 IV](#_Toc135692423)

[1.1 课题背景和意义 IV](#_Toc135692424)

[1.2 国内外现状 IV](#_Toc135692425)

[1.3 论文主要内容 IV](#_Toc135692426)

[2 开发工具及技术分析 V](#_Toc135692427)

[2.1 JAVA介绍 V](#_Toc135692428)

[2.2 IDEA介绍 V](#_Toc135692429)

[2.3 JSP介绍 V](#_Toc135692430)

[2.4 SSM框架 VI](#_Toc135692431)

[2.5 Tomcat服务器介绍 VI](#_Toc135692432)

[2.6 MySQL介绍 VI](#_Toc135692433)

[2.7 Element UI介绍 VI](#_Toc135692434)

[3 系统可行性研究 VII](#_Toc135692435)

[3.1 概述 VII](#_Toc135692436)

[3.2 社会可行性研究 VII](#_Toc135692437)

[3.3 经济可行性研究 VII](#_Toc135692438)

[3.4 技术可行性研究 VIII](#_Toc135692439)

[3.5 可行性研究结论 VIII](#_Toc135692440)

[4 系统需求分析 IX](#_Toc135692441)

[4.1 系统需求 IX](#_Toc135692442)

[4.1.1 目标概述 IX](#_Toc135692443)

[4.1.2 用户特点 IX](#_Toc135692444)

[4.2 功能需求 IX](#_Toc135692445)

[4.2.1 功能模块划分 IX](#_Toc135692446)

[4.2.2 功能概述 X](#_Toc135692447)

[4.2.3 功能用例模型 XI](#_Toc135692448)

[4.3 非功能需求 XI](#_Toc135692449)

[4.3.1 系统界面友好 XI](#_Toc135692450)

[4.3.2 系统运行状态 XI](#_Toc135692451)

[4.3.3 系统可维护性 XI](#_Toc135692452)

[4.3.4 输入输出要求 XI](#_Toc135692453)

[4.3.5 时间特性要求 XII](#_Toc135692454)

[4.4 运行环境 XII](#_Toc135692455)

[4.4.1 硬件环境 XII](#_Toc135692456)

[4.4.2 软件环境 XII](#_Toc135692457)

[5 系统概要设计 XII](#_Toc135692458)

[5.1 概要 XII](#_Toc135692459)

[5.2 模块设计 XII](#_Toc135692460)

[5.2.1 设计原则 XII](#_Toc135692461)

[5.2.2 业务流程分析 XIII](#_Toc135692462)

[5.2.3 系统的数据要求 XIV](#_Toc135692463)

[5.3 数据库设计 XV](#_Toc135692464)

[5.3.1 数据库设计概要 XV](#_Toc135692465)

[5.3.2 数据库概念设计 XV](#_Toc135692466)

[5.3.3 E-R图 XV](#_Toc135692467)

[5.3.4 补救措施 XVI](#_Toc135692468)

[5.3.5 系统数据库表的设计 XVII](#_Toc135692469)

[6 系统详细设计 XIX](#_Toc135692470)

[6.1 系统功能模块设计 XIX](#_Toc135692471)

[6.1.1 用户信息管理模块 XIX](#_Toc135692472)

[6.1.2 驾校信息管理模块 XX](#_Toc135692473)

[6.1.3 预约信息管理模块 XXI](#_Toc135692474)

[6.1.4 数据统计业务模块 XXII](#_Toc135692475)

[6.2 系统界面设计 XXIII](#_Toc135692476)

[6.2.1 概述 XXIII](#_Toc135692477)

[6.2.2 登录界面设计 XXIII](#_Toc135692478)

[6.2.3 注册界面设计 XXIV](#_Toc135692479)

[6.2.4 主页界面设计 XXIV](#_Toc135692480)

[6.2.5 预约信息界面设计 XXVI](#_Toc135692481)

[6.2.6 驾校信息管理设计 XXVII](#_Toc135692482)

[6.2.7 车辆信息管理界面设计 XXVII](#_Toc135692483)

[6.2.8 忘记密码界面设计 XXVIII](#_Toc135692484)

[7 系统实现与测试 XXVIII](#_Toc135692485)

[7.1 测试原则 XXVIII](#_Toc135692486)

[7.2 测试方法和环境 XXIX](#_Toc135692487)

[7.3 测试流程 XXIX](#_Toc135692488)

[7.4 测试用例 XXX](#_Toc135692489)

[7.4.1 非空测试 XXX](#_Toc135692490)

[7.4.2 非法输入测试 XXX](#_Toc135692491)

[总结 XXX](#_Toc135692492)

[参考文献 XXX](#_Toc135692493)

[致谢 XXX](#_Toc135692494)

1 绪论

## 课题背景和意义

随着汽车拥有量不断增高，驾照成为更多的人的刚性需求。但是，仅仅通过考试拿到驾照并不意味着就能够熟练地驾驶汽车。对于刚拿到驾照的新手，他还需要在实际的道路环境中练习和提高自己的驾驶技能和安全意识。这时候，一个专业的、有经验的、负责任的驾驶一对一陪练就显得非常重要。驾驶一对一陪练可以根据新手的具体情况，制定合适的教学计划和方法，帮助新手掌握各种交通规则和应对突发情况的技巧，提高新手的信心和水平，让新手在享受开车乐趣的同时，也能保证自己和他人的安全。同时提高了驾校的运营效率和服务质量，也可以方便了学员和教练之间的沟通和协调。该课题具有一定的创新性和实用性，随着社会对驾照需求和网络技术发展的趋势，这样一个系统有着广阔的市场前景和社会价值。

## 国内外现状

在国外，这种软件也有一些，比如英国的Midas、美国的RoadReady等。这些软件都有一些共同的特点，可以提供在线预约、支付、评价等功能提供实时路况、导航、行车记录等功能；提供个性化的课程规划和反馈；提供安全保障和保险服务。预约驾驶陪练软件已经有了稳定的发展，有一定的市场需求和用户基础。但是也面临着一些挑战，首先是法律法规的限制和监管；再者教练和陪练人员的质量和数量；最后随着自动驾驶的发展，用户对自动驾驶技术的信任度和接受度开始慢慢提升，该类软件发展受阻。

在国内，由于社会对驾照需求不断增加，而传统的驾校教学模式存在诸多问题和不足，如教学效率低下、服务质量差异大、费用不透明等，因此出现了一些新型的汽车陪练服务平台，如易通行汽车陪练、陪驾网等。这些平台通过Web或移动端提供一对一预约陪练服务，并配备专业教练和安全设备，旨在帮助新手或老司机提高自信心和技能水平。这些平台虽然还处于初级阶段，并没有完全实现Web化或智能化，但已经显示出了市场潜力和社会价值。

## 论文主要内容

该系统是一个基于Spring、Spring MVC和MyBatis框架开发的在线预约驾校练车的管理平台。

该系统分为前后端，前端主要供学员使用，后端主要供管理员和教练使用。

前端功能包括查看驾校新闻、教练信息、预约练车、在线留言、个人中心管理等。

后端功能包括对用户、管理员、教练、新闻、轮播图、系统、公告、留言等模块的管理。

该系统涉及三种不同的用户角色：管理员、教练和学员。每种角色都有自己的功能需求和操作权限。具体来说，管理员负责对学员、教练、车辆的信息进行统一管理和监督；教练负责安排自己的课程时间表、接受学员的预约请求和反馈车辆的故障问题；学员负责选择合适的教练进行预约、查看自己的学习进度和对教练进行评价等。另外教练和学员还可以完善和维护自己的个人信息。同时，对于新用户来说，新用户可以浏览驾校和教练信息从而选择成为某个驾校的学员。

2 开发工具及技术分析

## 2.1 JAVA介绍

JAVA是一种面向对象的编程语言，由Sun Microsystems公司于1995年推出。JAVA的特点是跨平台、高效、安全和易用。JAVA可以运行在不同的操作系统和硬件上，只要有JAVA虚拟机（JVM）的支持。JAVA也是一种高性能的语言，支持多线程、垃圾回收和异常处理等特性。JAVA的安全性体现在它的沙箱机制，可以防止恶意代码对系统造成破坏。JAVA的易用性体现在它的语法简洁、规范统一，以及丰富的类库和开发工具。

JAVA的优点是它可以开发出跨平台、高质量、可复用的软件，适合开发网络应用、移动应用、企业应用等。JAVA也有一些缺点，比如它的运行速度受到JVM的影响，可能不如本地编译的语言快；它的内存消耗较大，可能导致性能下降；它的语法较为繁琐，需要编写更多的代码。

## 2.2 IDEA介绍

IDEA是一款流行的集成开发环境，主要用于Java、Kotlin、Scala等语言的开发。它具有智能的代码补全和重构功能，可以提高开发效率和代码质量；并且拥有强大的调试和测试工具，可以帮助开发者快速定位和解决问题。同时还有丰富的插件和框架支持，可以扩展IDEA的功能和适应不同的开发需求。此外它还有美观和友好的用户界面，可以提升开发者的使用体验和工作愉悦度。但是IDEA也有一些缺陷，其中最主要的就是它是一款占用资源较多的开发工具，对于硬件配置较低的设备来说，可能会影响运行速度和性能。

## 2.3 JSP介绍

JSP（Java Server Pages）是一种动态网页开发技术，基于Servlet规范，可以在HTML代码中嵌入Java代码和JSP标签，实现静态和动态内容的结合。JSP页面最终会被服务器转换成Servlet类，然后编译运行，返回HTML页面给客户端浏览器。JSP技术的特点是：预编译：服务器在第一次访问JSP页面时，会对其进行编译，生成字节码文件，提高了执行效率和访问速度；业务逻辑分离：JSP页面可以将界面设计和应用程序开发分开，便于维护和更新；组件重用：JSP页面可以使用JavaBean封装业务逻辑或数据模型，实现组件的复用和共享；跨平台：JSP页面可以运行在任何支持Java的平台和服务器上，具有很强的移植性和兼容性。JSP技术一次编写，到处运行；系统的多平台支持；具有强大的可伸缩性；同时也具有多样化和功能强大的开发工具支持。但是系统的复杂性较高；内存和硬盘空间的消耗较大；调试代码时可能出现行数不对应的问题。

## 2.4 SSM框架

SSM是Spring、Spring MVC和Mybatis框架的整合，是一种标准的MVC模式。SSM框架可以实现业务逻辑层和持久层的解耦，提高了代码的可维护性和可扩展性；同时SSM框架利用了Spring的IOC和AOP特性，可以实现对象的管理和依赖注入，以及横切关注点的分离，提高了代码的复用性和安全性。另外SSM框架利用了Mybatis的灵活性和高效性，可以实现SQL语句的定制化和映射，以及动态SQL的生成，提高了数据库操作的性能和灵活性。SSM框架利用了Spring MVC的前端控制器模式，可以实现请求的统一分发和处理，以及视图的解析和渲染，提高了Web开发的效率和质量。但是SSM框架配置文件较多，需要花费一定的时间和精力进行配置和管理。另外，SSM框架对初学者来说有一定的学习难度，需要掌握多个框架的原理和使用方法。

## 2.5 Tomcat服务器介绍

Tomcat是一个开源的Java Web服务器，它可以运行Java Servlet和Java Server Pages（JSP）技术，提供了一个灵活的Web应用开发环境。Tomcat的特点有：轻量级，安装和配置简单，占用资源少，适合中小型Web应用；支持多种操作系统和Web服务器，具有良好的跨平台性和兼容性；遵循Java EE规范，支持多种Web开发框架和技术，如Spring Boot、Struts、Hibernate等；

提供了丰富的管理工具和文档，方便用户监控和维护Web应用。Tomcat性能稳定，安全性高，能够处理大量的并发请求；同时它的扩展性强，可以通过自定义组件和配置文件来实现个性化的功能和需求。

但是Tomcat功能相对单一，只能作为Web服务器或Servlet容器，不能提供完整的Java EE服务；性能相对较低，不能与专业的Web服务器相比，如Apache、Nginx等；配置文件较多，结构较复杂，需要一定的学习成本。

## 2.6 MySQL介绍

MySQL是一种开源的关系型数据库管理系统，它可以在多种操作系统上运行，支持多种编程语言和数据类型。MySQL具有高性能、高可靠性、易用性和灵活性的特点。MySQL的优点是成本低、兼容性好、扩展性强、安全性高。MySQL的缺点是不支持完整的事务处理、不支持存储过程和触发器、不支持外键约束和视图等高级功能。

## 2.7 Element UI介绍

Element UI是一个基于Vue.js的高质量UI组件库，它提供了丰富的组件，如按钮、表单、表格、对话框、导航栏等，可以满足各种开发需求。Element UI具有风格统一，符合设计规范，界面美观；组件丰富，功能强大，支持多种配置和自定义；文档完善，示例清晰，易于上手；兼容性好，支持主流浏览器和移动端的特点，同时Element UI可以提高开发效率，减少重复工作，同时可以保证UI质量，提升用户体验，另外还便于维护和扩展，遵循Vue.js的设计原则。

3 系统可行性研究

## 3.1 概述

系统可行性研究（Feasibility Study），是在系统开发初期对系统总体进行评估，包括系统的技术、经济、法律和时间等方面的可行性分析，并综合评估这些方面后得出开发系统的可行性结论。

系统可行性研究旨在确定一个项目的可行性，评估项目的优越性与开发所涉及的困难程度。系统可行性主要围绕着社会可行性、经济可行性、技术可行性等方向进行分析。

1.技术可行性：主要是对于技术方面的考虑，从技术的角度评估项目的可行性。需要分析该项目的技术设计，包括硬件和软件两个方面，是否能够实现所期望的目标，并对项目的实现已有的技术水平、软硬件设施、技术所涉及的复杂程度等多方面进行分析，保证程序的稳定性。

在对以上几个方面都进行了评估、研究和分析后，可以得出该项目的可行性结论。根据性能分析，可以对识别和调整项目的零部件、资源和流程，以确保它实现了既定的目标和要求。最终将有助于确保项目的实现线路明确，障碍清除，能够为客户带来价值和回报。

## 3.2 社会可行性研究

一对一驾驶陪练预约系统是一种利用互联网技术，为驾照考生提供个性化、灵活、高效的驾驶培训服务的系统。该系统的社会可行性主要体现在以下几个方面：

一对一驾驶陪练预约系统可以满足新手司机的多样化需求，例如时间安排、车辆类型、教练选择等，提高了学习效率和满意度。

一对一驾驶陪练预约系统可以提高驾驶培训的质量和安全性，通过专业的教练和车辆，为新手司机提供规范、科学、安全的教学内容和方法。

一对一驾驶陪练预约系统可以促进驾驶培训行业的发展和竞争，通过互联网平台，为教练和新手司机提供更多的信息和选择，增加了市场的透明度和活力。

因此，一对一驾驶陪练预约系统是一个具有社会价值和意义的创新项目，有利于提升社会公众的驾驶技能和素养，有利于改善社会交通状况。

## 3.3 经济可行性研究

一对一驾驶陪练预约系统是一种为新手司机提供个性化、灵活、高效的驾驶培训服务的系统。该系统的经济可行性主要体现在以下几个方面：

一对一驾驶陪练预约系统可以节约用户的学车成本，通过在线预约和支付，避免了传统驾校的中间环节和费用，同时可以根据用户的需求和时间安排，灵活选择合适的教练和车辆。

一对一驾驶陪练预约系统可以增加教练的收入，通过互联网平台，教练可以接触到更多的用户和订单，提高了工作效率和利用率，同时可以根据自己的能力和水平，设置合理的收费标准。

一对一驾驶陪练预约系统可以促进驾驶培训行业的发展和竞争，通过互联网平台，用户和教练可以获取更多的信息和选择，增加了市场的透明度和活力，同时也可以提高行业的服务质量和水平。

一对一驾驶陪练预约系统可以创造社会效益，通过提高用户的驾驶技能和素养，减少交通事故和违章，改善交通状况和环境质量，同时也可以培养用户的健康意识和生活方式。

因此，一对一驾驶陪练预约系统是一个具有经济价值和意义的创新项目，有利于提升社会公众的驾驶技能和素养，有利于改善社会交通状况和环境质量。

## 3.4 技术可行性研究

一对一驾驶陪练预约系统是一种基于互联网技术的在线服务系统，利用SSM框架、MySQL数据库、JSP技术等主流技术进行开发和实现。该系统的技术可行性主要体现在以下几个方面：

SSM框架是一种基于Java的Web开发框架，由Spring、Spring MVC和MyBatis三个组件组成，具有轻量级、高效率、易于扩展等优点，可以为Web应用提供稳定的支持。SSM框架可以实现前后端分离的开发模式，利用Spring MVC处理请求和响应，利用MyBatis处理数据访问和持久化，利用Spring进行依赖注入和事务管理。

MySQL数据库是一种开源的关系型数据库管理系统，具有高性能、高可靠性、易于使用等优点，可以为Web应用提供安全的数据存储和管理。MySQL数据库支持多种操作系统和编程语言，支持多种数据类型和索引类型，支持事务处理和存储过程。

JSP技术是一种基于Java的动态网页技术，具有跨平台、易于维护、高效率等优点，可以为Web应用提供动态的界面和功能。JSP技术可以嵌入Java代码和HTML标签，利用JavaBean、Servlet和EL表达式实现业务逻辑和数据交互。

因此，一对一驾驶陪练预约系统是一个技术成熟、可靠、高效的系统，有利于提升驾驶培训的质量和效率，有利于满足用户的多样化需求和偏好。

## 3.5 可行性研究结论

一对一驾驶陪练预约系统是一个具有技术可行性、经济可行性和社会效益的创新项目，可以为新手司机提供个性化、灵活、高效的驾驶培训服务，可以为教练提供更多的收入和选择，可以促进驾驶培训行业的发展和竞争，可以提高社会公众的驾驶技能和素养，可以改善社会交通状况和环境质量。因此，建议开发和实施这个系统，以满足市场的需求和潜力。

4 系统需求分析

## 4.1 系统需求

### 4.1.1 目标概述

该系统可以为想要提高驾驶技能的人提供一种方便快捷的预约方式，让他们可以自主选择合适的时间、地点和教练进行练习。同时为教练提供一个平台，让他们可以展示自己的专业水平和服务质量，吸引更多的学员，增加收入。另外还可以为驾校提供一个辅助工具，让他们可以管理和监督教练的工作情况，提高教学效率和质量，提升品牌形象。同时也能为社会提供一个有益的服务，让更多的人能够掌握安全、合规、高效的驾驶技巧，减少交通事故和拥堵，保障公共安全和环境。

### 4.1.2 用户特点

一对一驾驶陪练预约系统是一个为新手驾驶者和专业教练提供在线预约服务的平台。该系统主要有四类用户，分别是新用户，普通用户，教练和驾校管理员。

新用户：新用户是指第一次使用该系统的用户，他们可以通过注册功能创建自己的账号，或者通过登录功能使用已有的账号进入系统。注册功能需要新用户填写基本的个人信息，如姓名，手机号，邮箱等，并设置自己的密码。登录功能需要新用户输入正确的账号和密码才能进入系统。

普通用户：普通用户是指已经注册并登录过该系统的用户，他们可以通过预约功能和评价功能使用该系统的主要服务。预约功能允许普通用户根据自己的需求和偏好，选择合适的教练和时间段进行在线预约。预约成功后，普通用户可以查看预约详情，并在预约时间到达时与教练进行线下陪练。评价功能允许普通用户在陪练结束后，对教练的服务进行打分和评论，以反馈自己的满意度和建议。普通用户还可以取消预约和查看历史记录等。

教练：教练是指通过驾校提供陪练服务的专业人员，他们可以通过查看功能管理自己的预约请求。查看功能允许教练查看自己收到的所有预约请求，并根据自己的空闲时间和能力决定是否接受或拒绝。接受或拒绝后，教练可以与普通用户确认预约详情，并在预约时间到达时与普通用户进行线下陪练。

驾校管理员：驾校管理员是指负责管理该系统后台数据和资源的人员，他们可以通过管理功能对系统进行监督和调整。管理功能允许驾校管理员查看并修改系统中所有教练的信息，如姓名，手机号，邮箱，密码等，并对其进行增加，删除或禁用等操作。

## 4.2 功能需求

### 4.2.1 功能模块划分

该系统主要划分为用户信息管理、教练信息管理、驾校信息管理、车辆信息管理和数据统计模块。

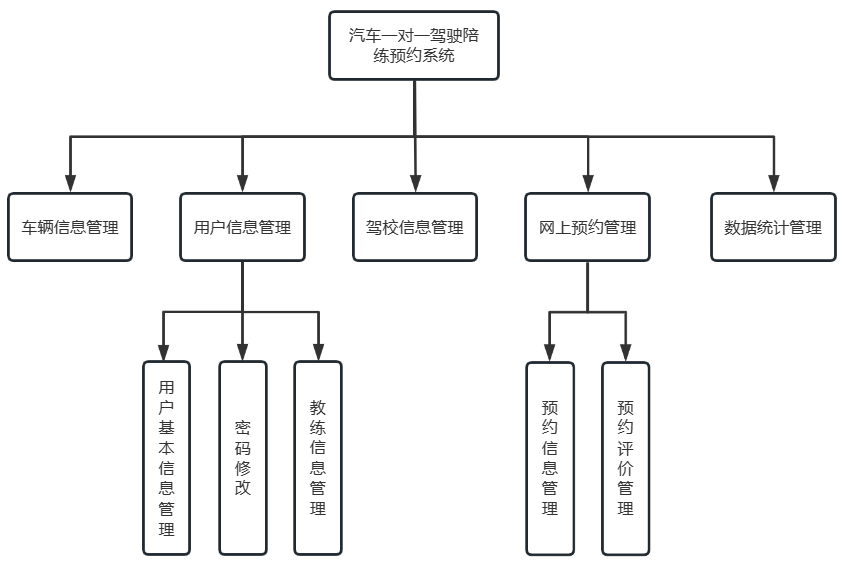


图4.1 功能模块划分

### 4.2.2 功能概述

用户信息管理模块主要涉及需要使用该系统所有用户的信息进行管理，主要内容包括用户注册登录注销操作，用户身份转换操作，用户实名验证操作。可以使用手机号注册账号成为新用户，登录账号进入该系统，进入系统后，新用户可以选择通过提交认证教练申请，经驾校管理员审核通过后成为教练用户，普通用户可以浏览驾校教练信息，通过实名认证后可以预约教练陪驾。新用户也可以通过验证内部密钥成为驾校管理员账号。

驾校信息管理主要包括新增驾校信息，更新驾校信息等。驾校信息主要由系统中的驾校管理员用户进行维护。这些信息可以被普通用户在驾校详情页看到。

车辆管理模块主要涉及驾校中车辆的管理，主要内容包括新增车辆信息，修改车辆信息，车辆报废管理。车辆信息以驾校为单位，由驾校管理员用户进行管理，驾校管理员录入车辆的相关信息后，用户可以在预约时指定车辆类型，使用中的车辆不能被其他用户预约，当车辆在使用过程中发生损坏，驾校管理员用户需要及时在系统中更新车辆状态，损坏严重的需要报废处理，损坏和报废的车辆也不可以被用户预约。

数据统计管理主要涉及教练和驾校的评价，用户预约陪驾的次数和时长，教练陪驾的次数和时长。教练的评价主要来源于其所参加的所有陪驾预约评价的平均值，而驾校的评价则为驾校中教练评价的平均值。用户和教练预约陪驾的次数和时长将会展示在个人主页上。

### 4.2.3 功能用例模型

系统在运行过程中的用例模型，如下图

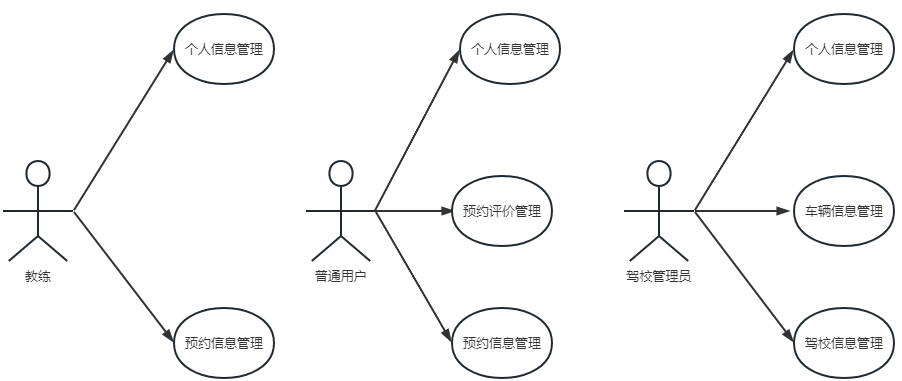


图4.2 用例模型

## 4.3 非功能需求

### 4.3.1 系统界面友好

本系统应提供简洁明了的用户界面，方便学员和教练员进行操作；本系统应支持多种浏览器和设备访问，适应不同屏幕尺寸。

### 4.3.2 系统运行状态

本系统应保证99.9%的正常运行时间，即每月故障时间不超过43分钟；本系统应具备容错机制，能够在发生异常或错误时自动恢复正常状态；本系统应采取严格的安全措施，防止未授权访问和数据泄露，保护用户隐私和信息安全。

### 4.3.3 系统可维护性

本系统应遵循良好的编码规范和文档规范，提高代码质量和可读性；本系统应提供完善的测试用例和测试报告，提高测试覆盖率和测试效率；本系统应支持模块化和组件化设计，提高代码复用率和修改灵活度；本系统应采用开放标准和通用技术，提高代码移植率和兼容率。

### 4.3.4 输入输出要求

本系统的输入主要包括用户的用户名和密码，用户选择的功能操作，用户输入的数据或信息，以及用户上传的图片等。本系统应能对输入的数据或信息进行合法性和有效性的检查，如非空检查，格式检查，范围检查等。如果输入的数据或信息不符合要求，本系统应能及时给出错误提示，并要求用户重新输入。本系统的输出主要包括系统给出的反馈信息，系统显示的数据或信息。本系统应能根据用户或管理员的选择，以合适的格式和方式输出数据或信息，如文本，表格，图表等。本系统应能保证输出的数据或信息的准确性和及时性，并避免冗余和重复。对于错误信息应该及时显示醒目的提示引导用户解决问题。

### 4.3.5 时间特性要求

本系统应能在高峰时段（每天8:00-10:00和16:00-18:00）支持至少1000名学员同时在线预约，响应时间不超过10秒；本系统应能根据学员和教练员的增加而动态扩展资源，保证服务质量。

## 4.4 运行环境

### 4.4.1 硬件环境

处理器：AMD R5-3500U（2.1GHz）

内存：8GB

存储：256GB

### 4.4.2 软件环境

操作系统：windows 10 22H2

Java版本：1.8

Web服务器：Apache Tomcat 8.5.87

数据库服务器：MySQL 8.0

5 系统概要设计

## 5.1 概要

概要设计作为软件开发的重要阶段之一，它是从需求分析到详细设计的一个过渡阶段。设计人员在概要设计阶段中需要结合需求分析的结果，综合考虑系统的性能、架构、模块划分、接口设计等方面的要素，用以定义系统的总体设计目标和方案，贯穿整个软件开发过程。

本系统概要设计的主要内容包括模块设计，数据库设计两个方面的内容。

## 5.2 模块设计

### 5.2.1 设计原则

对于这个课题的概要设计，可以采用以下设计原则：

1. 模块化原则：将系统划分成多个模块，如用户管理、陪练预约、支付管理等，以便于开发人员对每个模块进行独立的开发和维护。

2. 高内聚低耦合原则：每个模块内部的组件和逻辑应该相互关联，以实现高内聚，而不同模块之间应该尽可能减少依赖关系，实现低耦合。

3. 可扩展性原则：在概要设计中需要考虑到未来的系统扩展需求，例如不同城市的预约服务、增加不同类型车辆的陪练服务等，以提高系统的可扩展性。

4. 可维护性原则：设计时要尽可能精简和清晰，减少冗余代码，方便维护和升级，尤其数据模型设计需要友好。

5. 安全性原则：保证数据库和用户信息安全，例如使用密码加密技术，避免注入攻击等方式。

6. 可用性原则：界面设计应该符合用户的使用习惯，简单明了，操作流程要简单易懂。比如，预约服务界面需要有清晰明了的输入提示。

7. 性能优化原则：在设计时要考虑系统的性能和效率问题，例如减少文件大小、使用CDN等方式提高系统的响应速度和性能。

这些设计原则可以指导设计人员进行系统概要设计的工作，并促使设计人员在系统概要设计中更全面地考虑到实际情况，从而实现一个具备高效性、可靠性、扩展性和易维护性的系统。

### 5.2.2 业务流程分析

通过分析，本系统主要包含以下主要业务：

用户登录注册业务：登录注册业务是一个网站或应用程序中非常基础的业务之一，其主要目的是为了提供用户身份认证和信息管理功能，让用户确定自己的身份，让网站或应用程序确认用户的身份，从而保证用户可以安全地使用网站或应用程序。以下是用户登录注册的业务流程图。

图5.1 用户登录注册的业务流程图

用户预约业务：用户预约业务作为该系统的核心业务，其主要目的为了让用户简单快捷的完成驾驶陪练的预约，同时再预约的过程中，还可以根据实际修改预约信息。以下是用户预约业务流程图。

图5.2 用户预约业务流程图

用户实名认证业务：用户实名认证是为了确保用户身份信息的真实性，进一步保障系统安全。用户在实名认证时需要填写真实的身份证信息，并上传正面身份证照片，由系统进行身份证信息核验，核验成功后用户才能进行预约操作。对于未完成实名认证的用户，用户在填写预约表单前系统会弹出提示要求用户进行实名认证。新用户实名认证后会转变为该系统的普通用户。以下是用户实名认证业务流程图。

图5.3 用户实名认证业务流程图

教练身份认证业务：用户身份认证的目的是为了确保用户拥有相应的身份和权限，并给予不同的操作入口和菜单展示。教练身份认证业务中，需要选择所在驾校并填写自己在驾校的工号以便系统核验其教练资格，提交后会将申请转到驾校管理员处由驾校管理员处理该身份申请，通过申请后，用户在系统中的身份将会转变为教练。以下是教练身份认证的业务流程图。

图5.4 教练身份认证的业务流程图

### 5.2.3 系统的数据要求

在计算机系统中，数据是系统的重要组成部分，系统需要对不同类型的数据进行存储、管理、处理和调用。对于不同类型的系统，其对数据的要求也是不同的，但总体而言，系统对数据有以下要求：

1.准确性：系统需要对数据的真实性、合理性和完整性进行验证，确保数据的准确性。数据的准确性对于系统的正常运行和最终的成果具有关键性的作用。综上所数，本系统要求浮点数保留两位小数以上，对于时长，金额等可能涉及到无法用整数覆盖的场合必须用浮点型数据类型。

2.一致性：系统要求数据的一致性，即在不同时期、不同地点和不同操作上，相同数据必须保持一致。数据的不一致会影响到系统的正常运行，同时也会导致管理和决策的偏差。综上所述，要求数据库中和系统中数据类型要一致对应，对于日期类型的数据，系统和数据库中都必须保持一致，不可用字符串存储日期数据。

3.可用性：系统对数据的要求是数据必须及时、准确、可靠地提供给需要它的用户，以便于业务流程的正常运行和决策的高效实现。综上所述，在进行数据处理前要对数据的合法性进行判别，对于空值和不存在的值要求立即抛出异常，以免引起系统崩溃。

4.安全性：随着数据的存储和传输，尤其是网络时代的到来，对于系统数据的安全性要求不断提高。系统需要采取合适的技术手段和管理措施，确保数据不被非法获取、篡改或其他不良行为威胁。综上所述，要求涉及用户隐私的数据不可以明文的形式传向前端，其中包括但不限密码；对于某些个人信息，非本用户访问时，不应以明文的形式显示或应拒绝显示，包括但不限定于真实姓名，身份证号码等。

5.互操作性：尤其对于复杂的系统，需要实现系统之间的数据交换、数据共享和数据转换等功能，提高系统的相互适配性和互操作性，以保证系统在整个业务流程中有良好的互动性。综上所述，对于重复调用的数据，应该从后端存储到本地，确保数据交换和共享的效率，但同时也要保证数据更新时，本地数据也应该及时更新，这类数据包括但不限于用户身份标识、用户编号

总而言之，数据对于计算机系统的重要性是不言而喻的。数据的准确、一致、可用、安全和互操作性都是系统对数据的基本要求。优秀的数据管理和数据处理技术，是系统有效、高效、稳定运行的重要保障之一，决定着系统是否能够真正发挥其应有的作用。

## 5.3 数据库设计

### 5.3.1 数据库设计概要

在高内聚低耦合的系统开发模式下，数据库作为连接各系统和各模块的核心扮演着极其重要的角色。所有的信息都存储在数据库中，并且通过操作数据库中的数据来实现需要的功能。本系统所选用的数据库工具为 MySQL，而这个数据库设计的优劣将直接影响到整个系统的建设。为了进行良好的数据库开发，需要先进行系统分析，之后进行逻辑设计，最后进行物理实现。在逻辑设计的过程中，需要根据系统分析来设计一个关系模型，而在物理设计中，则需要选取一个最合适的应用环境物理结构来实现逻辑设计中的数据模型。这些优秀的设计决策是基于选取合适的数据库软件，而接下来则是将数据模型与数据库软件对应来创建数据库。

### 5.3.2 数据库概念设计

根据前期的需求分析，我们可以得到这个系统在开发和使用的过程中需要记录和使用以下数据：

用户信息：用户编号，用户名，真实姓名，性别，年龄，手机号码，邮箱，身份证号，密码，身份标识；

教练信息：用户编号，工号，所属驾校，评价；

驾校管理员信息：用户编号，工号，所属驾校，评价；

车辆信息：车辆编号，车牌号，车辆品牌，车辆型号，所属驾校编号，所属教练编号，状态标识，更新时间；

驾校信息：驾校编号，驾校名称，驾校地址，联系方式，评价，更新时间；

预约信息：预约编号，预约者编号，预约对象编号，预约时长，开始时间，结束时间，预约地点，预约状态，评价；

图片信息：图片编号，图片所属，存储路径。

### 5.3.3 E-R图

根据分析可以得到，系统在开发和使用的过程中，主要涉及到用户信息，教练信息，驾校（管理员）信息，预约信息和图片信息等实体，系统整体的E-R图如下：

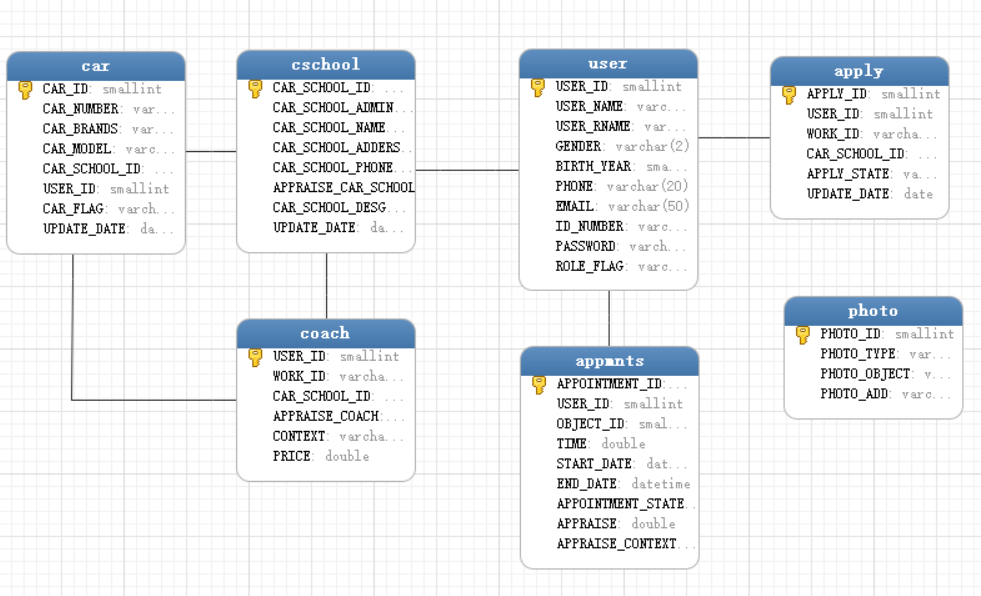


图5.1 系统E-R图

### 5.3.4 补救措施

数据库系统是现代大型系统中核心技术之一，其稳定性和可靠性对于整个系统极其重要。然而，数据库系统中仍然可能存在一些不稳定的因素，例如系统崩溃、数据丢失等。为此，本系统采取了定时备份数据库的措施，以保障数据的完整性和系统的稳定性。

首先，备份对于数据库系统而言是一种安全性措施，能够避免因数据丢失或系统崩溃而导致系统无法正常工作。在备份过程中，所有的数据将被完全保存，以便在需要时进行恢复。而定时备份数据库又能保证数据及时地更新，减少系统崩溃后数据丢失的可能性。

其次，在进行定时备份时，需要考虑到备份周期和备份策略。备份周期是指备份的时间间隔，通常可以根据数据量和重要性来确定，以保证备份的及时性；备份策略包括全量备份和增量备份。全量备份是指备份所有数据，确保备份的完整性；而增量备份则是对发生变化的数据进行备份，以减少备份的大小和时间成本。需要根据实际情况来选择相应的备份策略，以达到最佳备份效果。

另外，备份数据的存储和管理也是备份过程中需要注意的问题。备份数据需要保存在可靠的储存介质中，例如有 MySQL 的服务器、硬盘、磁带等。而备份数据的管理则涉及到备份的分类、分类的存储位置、备份数据的存储时长等问题。需要对备份数据进行分类存储，以便于查找和管理。另外，需要根据公司的相关规定制定备份数据的存储时长，例如一般情况下为 1~2 周，对于敏感数据则需要更加重视。

最后，备份的实施需要根据实际情况考虑硬件成本和维护成本等问题。需要对备份策略进行不断优化，以达到效果最佳的备份效果。而相关的监控和维护工作也需要不断地完善，以便于及时发现备份数据异常和备份过程中的问题，及时解决和修复，从而提高系统的稳定性和可靠性。

综上所述，定时备份数据库是保障数据库系统稳定性和可靠性的重要措施。备份周期、策略、存储和管理、监控和维护等环节都需要进行精心的设计和管理，以达到备份效果最佳的目的。我决定将备份周期定为一个周，采用全量备份的备份策略，存储介质选择存储速度快，成本较低的机械硬盘。

### 5.3.5 系统数据库表的设计

通过前期的系统分析和数据库概念设计，最终决定在数据库中建立用户表，教练表，车辆信息表，驾校表，预约信息表，图片信息表五张表，具体详细信息如下：

（1）用户信息表（userInfo）

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 属性 | 列名 | 数据类型 | 是否可为空 | 说明 |
| 1 | 用户编号 | USER\_ID | smallint(10) | 否 | 主键 |
| 2 | 用户名 | USER\_NAME | varchar(10) | 否 |  |
| 3 | 真实姓名 | USER\_RNAME | varchar(10) | 是 |  |
| 4 | 性别 | GENDER | varchar(2) | 是 |  |
| 5 | 出生年 | BIRTH\_YEAR | varchar(4) | 是 |  |
| 6 | 手机号码 | PHONE | varchar(20) | 否 |  |
| 7 | 邮箱 | EMAIL | varchar(50) | 是 |  |
| 8 | 身份证号 | ID\_NUMBER | varchar(20) | 是 |  |
| 9 | 密码 | PASSWORD | varchar(20) | 否 |  |
| 10 | 身份标识 | ROLE\_FLAG | varchar(2) | 否 |  |

（2）教练信息表（coachInfo）

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 属性 | 列名 | 数据类型 | 是否可为空 | 说明 |
| 1 | 用户编号 | USER\_ID | varchar(10) | 否 | 主键，外键 |
| 2 | 工号 | WORK\_ID | varchar(10) | 否 |  |
| 3 | 所属驾校编号 | CAR\_SCHOOL\_ID | varchar(10) | 否 | 外键 |
| 4 | 评价 | APPRAISE\_COACH | smallint(4) | 是 |  |

（3）驾校管理员信息表（carSchoolManagerInfo）

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 属性 | 列名 | 数据类型 | 是否可为空 | 说明 |
| 1 | 用户编号 | USER\_ID | smallint(10) | 否 | 主键，外键 |
| 2 | 工号 | WORK\_ID | varchar(10) | 否 |  |
| 3 | 所属驾校编号 | CAR\_SCHOOL\_ID | varchar(10) | 否 | 外键 |

（4）车辆信息表（carInfo）

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 属性 | 列名 | 数据类型 | 是否可为空 | 说明 |
| 1 | 车辆编号 | CAR\_ID | varchar(10) | 否 | 主键 |
| 2 | 车牌号 | CAR\_NUMBER | varchar(10) | 否 |  |
| 3 | 车辆品牌 | CAR\_BRANDS | varchar(10) | 否 |  |
| 4 | 车辆型号 | CAR\_MODEL | varchar(20) | 否 |  |
| 5 | 所属教练编号 | USER\_ID | smallint(10) | 否 | 外键 |
| 6 | 状态标识 | CAR\_FLAG | varchar(2) | 否 |  |
| 7 | 更新时间 | UPDATE\_DATE | datetime() | 否 |  |

（5）驾校信息（carSchoolInfo）

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 属性 | 列名 | 数据类型 | 是否可为空 | 说明 |
| 1 | 驾校编号 | CAR\_SCHOOL\_ID | varchar(10) | 否 | 主键 |
| 2 | 驾校名称 | CAR\_SCHOOL\_NAME | varchar(20) | 否 |  |
| 3 | 驾校地址 | CAR\_SCHOOL\_ADDERS | varchar(20) | 否 |  |
| 4 | 联系方式 | CAR\_SCHOOL\_PHONE | varchar(20) | 否 |  |
| 5 | 评价 | APPRAISE\_CAR\_SHOOL | smallint(4) | 是 |  |
| 6 | 更新时间 | UPDATE\_DATE | datetime() | 否 |  |

（6）预约信息

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 属性 | 列名 | 数据类型 | 是否可为空 | 说明 |
| 1 | 预约编号 | APPOINTMENT\_ID | varchar(10) | 否 | 主键 |
| 2 | 预约者编号 | USER\_ID | smallint(10) | 否 | 外键 |
| 3 | 预约对象编号 | OBJECT\_ID | smallint(10) | 否 | 外键 |
| 4 | 预约时长 | TIME | smallint(4) | 否 |  |
| 5 | 开始时间 | START\_DATE | datetime() | 否 |  |
| 6 | 结束时间 | END\_DATE | datetime() | 否 |  |
| 7 | 预约地点 | APPOINTMENT\_ADDERS | varchar(20) | 否 |  |
| 8 | 预约状态 | APPOINTMENT\_STATE | varchar(2) | 否 |  |
| 9 | 评价 | APPRAISE | smallint(4) | 是 |  |
| 10 | 评价内容 | APPRAISE\_CONTEXT | varchar(200) | 是 |  |

6 系统详细设计

## 6.1 系统功能模块设计

本系统主要分为四大模块，用户信息管理模块主要包含了新用户注册、用户登录、用户基本信息的维护、用户实名信息和身份信息的提交和维护的功能；驾校管理包括了驾校信息管理和驾校中的车辆信息管理。教练信息管理主要是教练的身份信息的管理以及关于教练的扩展信息的管理。预约信息管理主要包括了预约业务的流程和预约评价，驾校评价以及教练评价的业务操作。

### 6.1.1 用户信息管理模块

用户信息管理模块主要包含了用户注册和登录、用户实名认证和用户身份认证的相关业务流程。

1. 用户注册操作

业务功能：用户注册是新用户进入系统的第一步，用户注册主要收集用户的最基本信息，包括用户名，手机号，密码等，用户的核心身份信息进行收集。用户在注册后系统会给用户分配用户编号作为用户身份的唯一性标识。

操作用户：未注册的新用户。

业务场景：用户在进入系统登录页后，点击注册进入注册页面，依次填写用户名，手机号，密码和密码确认后点击提交完成注册，完成注册后会弹出弹窗告知用户编号以及提醒用户进入系统绑定邮箱的提示。该用户身份转变为该系统的新用户。

1. 用户登录操作

业务功能：用户登录是用户进入系统的最关键一步，用户登录主要是为了验证用户身份，根据不用的身份进入不同的页面，显示不同的权限菜单从而给予不同的用户权限。对于没有注册的用户或者用户编号密码不匹配的用户，系统会弹出错误提示，提示用户检查用户名和密码。对于新用户和普通用户，主要有预约功能，个人基本信息更新功能、实名认证和身份认证的功能入口。而对于教练用户，主要提供预约信息管理，个人基本信息更新功能的入口。对于驾校管理员，则有驾校信息管理，驾校车辆信息管理以及个人基本信息更新的功能入口。

操作用户：新用户。

业务场景：用户在进入系统登录页后，输入自己的用户编号和密码，账号密码不匹配则提示错误要求用户重新输入。如果账号密码匹配，用户进入系统首页并按照用户的角色渲染相对应的权限菜单。

1. 实名认证操作

业务功能：用户实名认证是为了确保用户身份信息的真实性，进一步保障系统安全。用户在实名认证时需要填写真实的身份证信息，并上传正面身份证照片，由系统进行身份证信息核验，核验成功后用户才能进行预约操作。对于未完成实名认证的用户，用户在填写预约表单前系统会弹出提示要求用户进行实名认证。新用户实名认证后会转变为该系统的普通用户。

操作用户：新用户。

业务场景：用户在进入系统首页后，点击“实名认证”菜单项进入实名入口页面。用户需要填写身份证号码、真实姓名，上传身份证正面照片。系统会将身份信息存入数据库中，并按照身份证号更新用户的出生年和性别，在此以后出生年和性别无法修改。之后用户就可以预约教练陪驾了。

1. 身份认证操作

业务功能：用户身份认证的目的是为了确保用户拥有相应的身份和权限，并给予不同的操作入口和菜单展示。申请不同的身份，身份认证流程会略有不同。对于教练身份认证，需要选择所在驾校并填写自己在驾校的工号以便系统核验其教练资格，提交后会将申请转到驾校管理员处由驾校管理员处理该身份申请，通过申请后，用户在系统中的身份将会转变为教练。对于驾校管理员身份认证，需要填写驾校名称和邀请码，并在弹出的页面填写驾校相关信息，若邀请码与驾校信息对应，则身份认证通过，该驾校的信息将会存储在数据库中，该用户在系统中的身份将会转变为驾校管理员。

操作用户：普通用户；

业务场景：

1. 教练身份认证

教练用户在登录系统后，找到身份认证入口，选择所在的驾校并填写自己在这个驾校中的工号，并上传教练证件照片。系统会进行信息核验，核验成功后教练用户就可以享有教练的相关操作入口，例如接受预约。

1. 驾校管理员身份认证

驾校管理员用户在登录系统后，找到身份认证入口，填写自己的驾校名称和邀请码，并在弹出的页面中完善驾校的相关信息。系统会进行信息核验，核验成功后管理员用户就可以享有管理员的相关操作入口，例如车辆管理、教练管理等。

### 6.1.2 驾校信息管理模块

驾校信息管理模块主要包含了驾校基本信息管理，驾校中的教练管理，驾校中的车辆管理的业务操作。

1. 驾校基本信息管理

业务功能：驾校信息管理是为了确保学员和教练的信息完备性以及更好地管理教学资源。驾校管理员可以在系统中查看、添加、修改和删除驾校的基本信息, 包括驾校名称、联系电话、地址，驾校相关照片。

操作用户：驾校管理员。

业务场景：驾校管理员登录系统后，进入驾校信息管理模块，可以查看已有驾校的基本信息列表。管理员可以通过更新功能在系统中更新驾校信息，填写驾校的名称、联系电话、地址等信息，并将其保存到系统中。还可以上传驾校有关的图片（最多可上传三张），这些图片将会展示在用户预约操作时驾校的主页上。这些功能可以方便管理员进行驾校教育资源的管理，还能方便用户更加直观的了解驾校的情况。

1. 教练相关信息管理

业务功能：教练信息管理是为了方便驾校管理员对教练个人信息和收费信息进行管理，能够更好地管理教学资源。教练的默认预约单价为30元每小时，驾校管理员修改教练的预约单价。驾校管理员还可以点击查看教练的相关信息，包括教练被预约的次数和时长，用户对教练的评价和历史评价内容，有利于驾校管理员实时掌握驾校中教练的工作状态。

操作用户：驾校管理员。

业务场景：驾校管理员登录系统后，进入教练信息管理模块，可以查看本驾校教练的基本信息列表。点击编辑管理员可以对教练的预约单价进行设置，包括各种类型的车辆的单价，驾校管理员可以根据实际情况灵活调整教练的预约单价；还可以对教练的简介进行查看和修改。教练的简介将会在用户预约时，展示在教练的信息列表中。这些功能可以方便管理员进行教学资源及教练个人信息的管理。

1. 车辆信息管理

业务功能：车辆信息管理是为了方便驾校管理员对驾校车辆的基本信息和状态进行管理。驾校管理员可以添加、查看和修改车辆的基本信息和状态。主要包括车辆的车牌号、品牌、车辆类型等，也可以修改车辆的所属以及状态，还可以上传车辆的相关图片。

操作用户：驾校管理员。

业务场景：驾校管理员登录系统后，进入车辆信息管理模块，可以查看已有车辆的基本信息列表。管理员可以点击新增按钮在系统中添加新的车辆信息，填写车牌号、品牌、车辆类型等基本信息，并上传车辆图片（最多可上传3张 ）。同时，管理员还可以点击操作列的编辑按钮，编辑车辆的所属和状态，包括将车辆分配给某位驾校教练或收回车辆，以及标记车辆状态（使用中、维修中等）。

### 6.1.3 预约信息管理模块

预约信息管理模块主要包含预约的申请和修改，预约的评价管理。

1. 预约的申请和修改

业务功能：预约的申请和修改是该系统的核心业务操作。普通用户提交预约申请，主要信息包括预约者，预约对象，预约开始时间，结束时间和时长，以及预约的状态。普通用户在填写预约信息时，日期的可选范围为当前日期到本周周天；开始时间的可选范围是所选日期的8点到18点中过滤掉已预约的时间段；结束时间的可选范围是所选日期中开始时间到18点中过滤掉已预约的时间段；时长为结束时间与开始时间的时间差，单位是小时；开始时间和结束时间只能选择完整的时间点或半小时时间点。普通用户在修改预约信息时，日期、开始时间和结束时间填写的限制条件和新增时相同，如用户在修改预约信息时已经过了预约的结束时间，预约的状态则将自动转变为已结束；预约可以在开始时间撤销。教练只能查看预约信息，无法修改预约信息。

操作用户：普通用户，教练用户。

业务场景：

1. 预约提交

普通用户在登录系统后，在主页中通过驾校信息列表进入驾校主页，可通过教练信息列表中的预约跳转到预约申请页面，填写预约日期、开始时间和结束时间，点击提交后会自动跳转到支付页面，支付金额为时长和教练的预约单价乘积，支付成功后预约成功。

1. 预约查看和修改

普通用户在登录系统后通过预约信息管理菜单项，可查看已预约的预约信息列表，点击编辑可查看预约信息也可修改，在预约开始时间之前，普通用户可以修改预约的日期、开始时间和结束时间。在预约结束时间之后，预约的状态只能为“已结束”。

教练用户在登录系统后通过预约信息管理菜单项，可查看已预约的预约信息列表。为了保证普通用户的信息安全，在展示的信息列表中，普通用户的姓名会根据姓氏和性别展示为某先生或某女士。

1. 预约撤销

普通用户在登录系统后通过预约信息管理菜单项，可查看已预约的预约信息列表，点击编辑在弹出的页面下方点击撤销按钮，弹出确认弹窗点击确认后即可撤销。当前时间超过开始时间之后，撤销按钮将会被禁用。

### 6.1.4 数据统计业务模块

数据统计业务模块主要包含驾校和用户的预约总计信息和用户的最近一次预约信息。

业务功能：数据统计模块是为了方便驾校管理员查看驾校的总预约信息，主要包括驾校中所有教练预约的总时长和总次数；驾校管理员查看驾校统计信息，主要包括驾校中的教练人数（员工数）和车辆数；教练用户和普通用户查看自己的预约总计信息，主要包括预约的次数和时长；以及教练用户和普通用户最近一次预约的具体信息，主要包括预约者（对象），开始时间，结束时间，状态；教练和驾校查看各自评分。其中参与统计的预约不包括未开始的预约和已撤销的预约。该模块可提供驾校的预约总时长，总次数，车辆数，员工数；教练和普通用户的预约总时长，总次数和最近一次预约。其中教练的评分来自于本教练所有已评价预约的评分的平均数，驾校的评分来自于该驾校中所有教练评分的平均数。

操作用户：驾校管理员，普通用户。

业务场景：

当普通用户登录系统后，在主页上显示普通用户的预约总计信息和最近一次预约信息。

当教练用户登录系统后，在主页上显示教练用户的预约总计信息、最近一次预约信息和教练的评分信息。

当驾校管理员登录系统后，在主页上显示驾校的预约总计信息和驾校统计信息、驾校的评分。

当驾校管理员登录系统后，点击驾校信息管理菜单项，弹出的页面上会显示驾校的预约总计信息和驾校统计信息和驾校的评分。

## 6.2 系统界面设计

### 6.2.1 概述

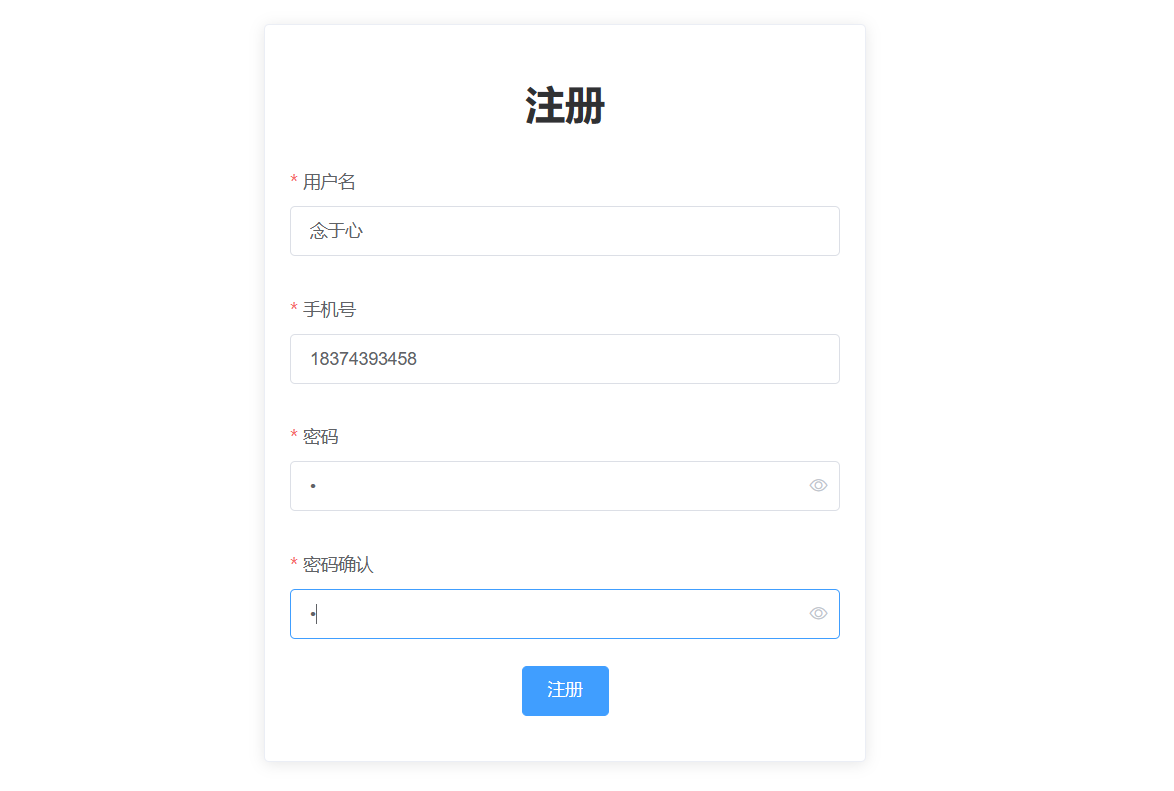
系统界面设计是指对计算机系统的人机交互界面进行设计和优化的过程。系统界面作为计算机系统中用户接口的一部分，是用户与计算机系统进行交互的载体，界面设计的好坏直接影响到用户的使用体验和工作效率，因此系统界面设计非常重要。

总之，系统界面设计是很重要的一环，一个好的界面设计能够提高系统的使用效率和用户满意度。因此在界面设计之前应充分了解用户需求，针对用户习惯、需求特点和行为习惯进行合理化的设计实现，最终以丰富、优秀的系统界面，为用户带来更好的使用感受和体验。

### 6.2.2 登录界面设计



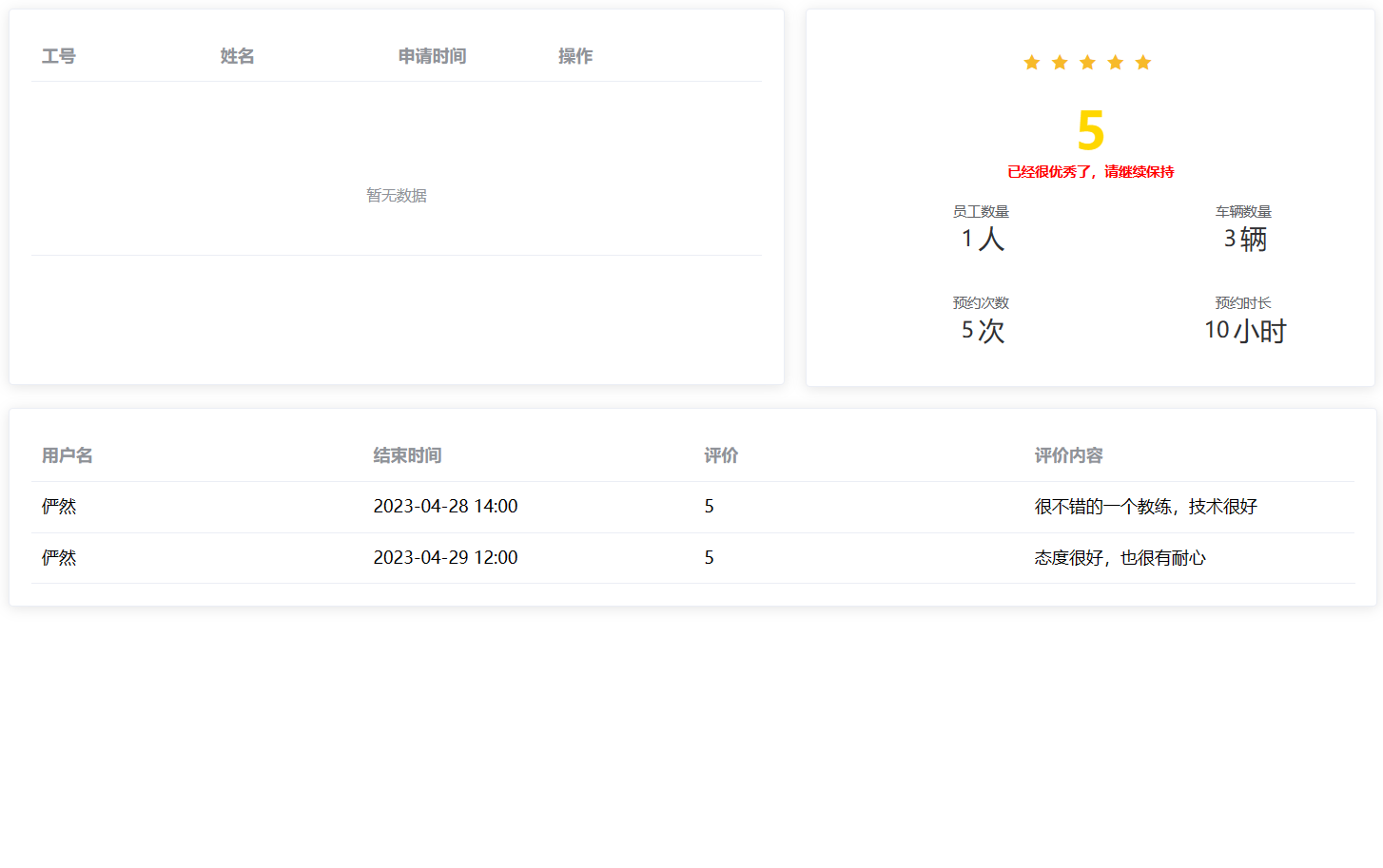
### 6.2.3 注册界面设计



### 6.2.4 主页界面设计

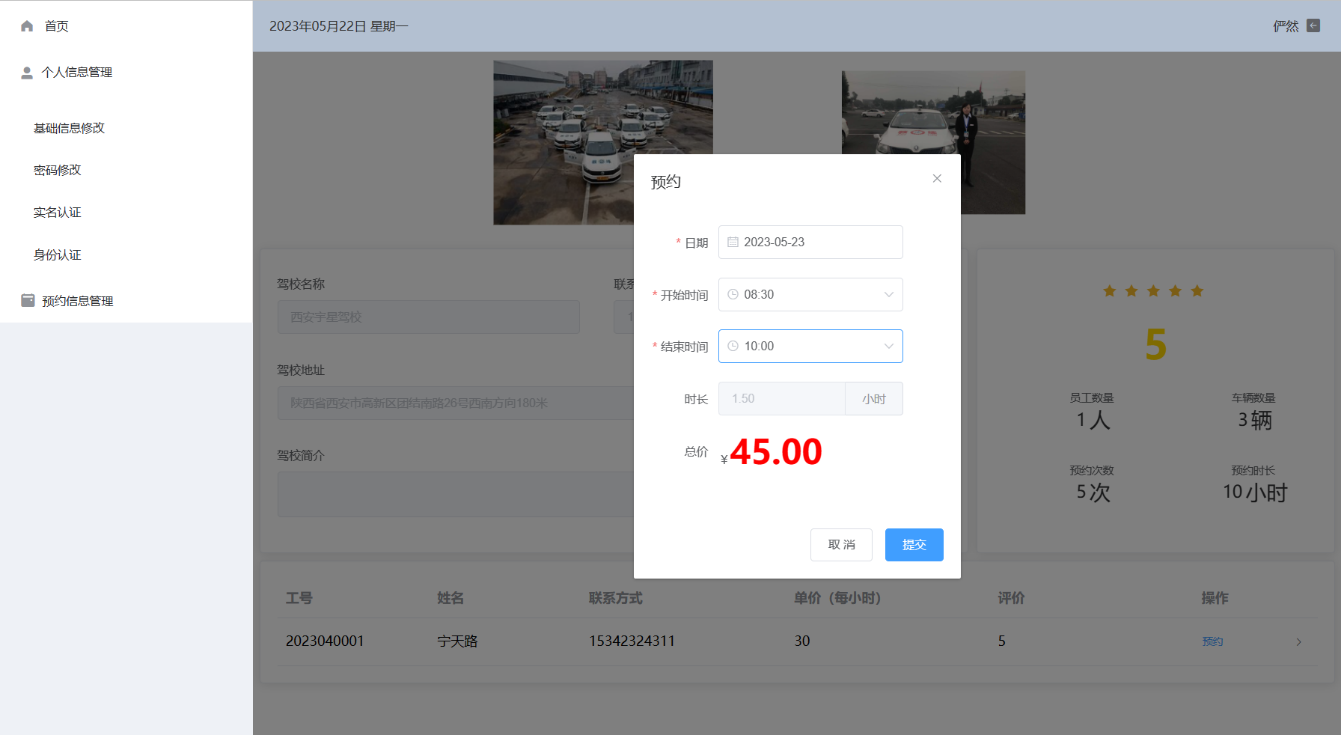




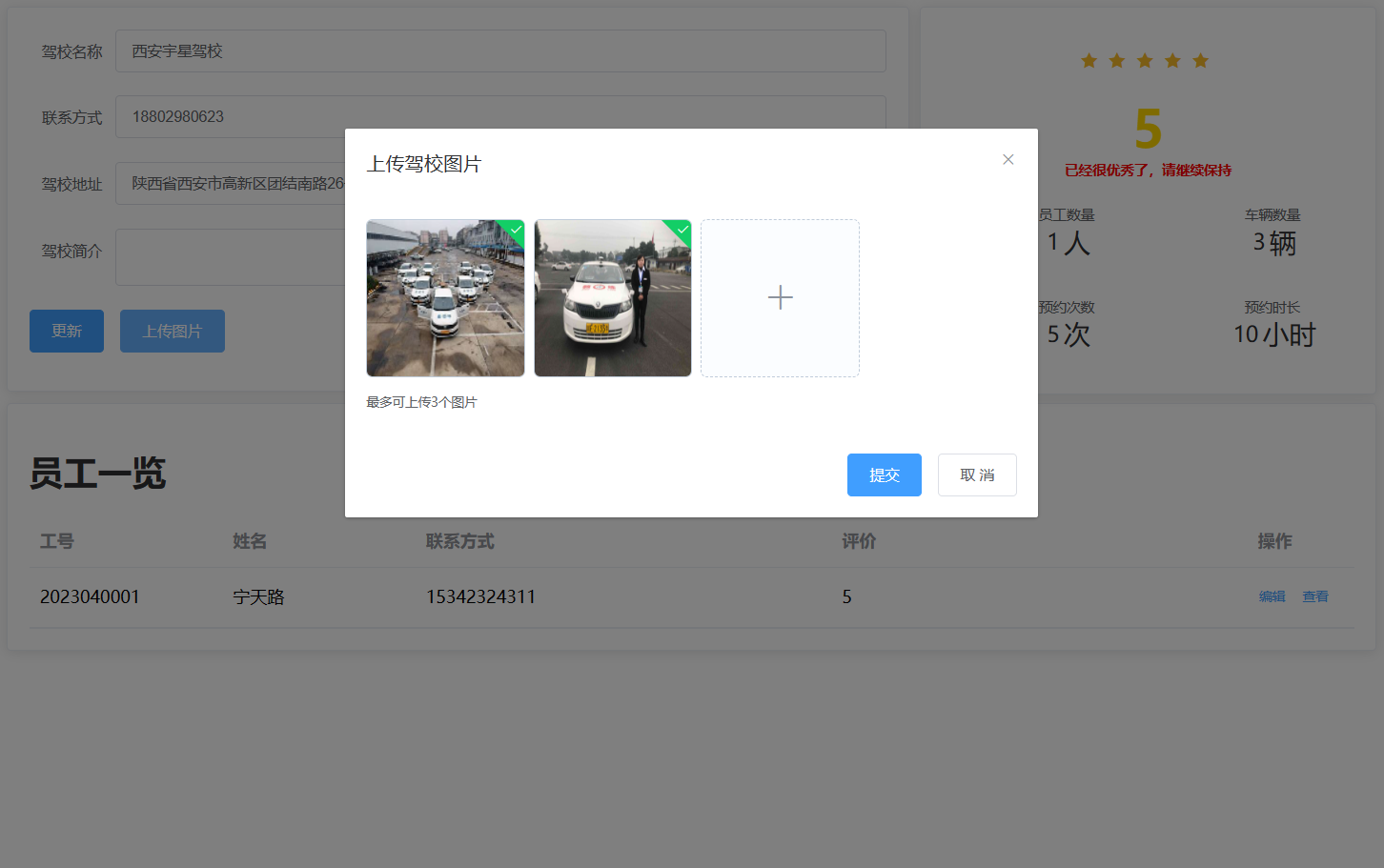




### 6.2.5 预约信息界面设计



### 6.2.6 驾校信息管理设计



### 6.2.7 车辆信息管理界面设计



### 6.2.8 忘记密码界面设计



7 系统实现与测试

## 7.1 测试原则

在该系统测试时，需要遵循以下原则：

完备性原则：要保证测试用例的覆盖范围尽可能完整，确保每个功能模块和每个组件都经过了测试，并尽可能发现系统中的各类缺陷和性能问题。

可重复性原则：测试用例应该是可重复的，可以在不同的环境和条件下进行反复测试，确保测试结果的稳定性和一致性。

有效性原则：测试用例应该是有效的，并尽可能检测到系统中的所有错误和异常情况，帮助开发人员及时发现和解决问题。

自动化原则：对于重复性高的测试任务，可以使用自动化测试工具和技术，提高测试效率、减少测试成本。

综合性原则：除了功能测试，还应该对系统的安全性、性能和可用性等方面进行综合性测试，以保证系统的整体质量水平。

可追踪性原则：测试过程中需要建立相应的测试报告和日志，记录测试过程、测试结果和问题跟踪等信息，便于追踪问题和评估测试质量。

代码覆盖率原则：对于涉及到代码的测试，需要关注代码覆盖率，尽可能覆盖到所有源代码，以提高代码质量和发现潜在问题。

以上原则可以指导测试人员在测试过程中操作，确保系统经过了全面的测试和评估，在发布和部署前，确保系统质量满足要求。

## 7.2 测试方法和环境

为了测试Java Web项目，我们通常需要准备以下测试环境和测试方法：

1. 测试环境：

（1） 数据库环境：MySQL 8.0。

（2） 应用服务器环境：Tomcat 8.5。

（3） 测试工具环境：postman。

2. 测试方法：

（1） 单元测试：对Java Web项目的每个模块进行单元测试，测试模块之间的交互和功能是否正常。

（2） 集成测试：对Java Web项目的整体集成进行测试，测试项目所需要的依赖库、API、数据库环境等是否正常。

（3） 性能测试：对Java Web项目进行性能测试，测试系统在不同负载情况下的性能指标，包括响应时间、并发数等。

（4） 安全测试：对Java Web项目进行安全测试，发现项目中可能出现的安全漏洞、SQL注入、XSS攻击等问题。

（5） 可用性测试：验证Java Web项目的易用性和用户体验，检查项目的界面设计是否符合用户的操作习惯和交互设计。

## 7.3 测试流程

系统实现与测试是指在完成系统的开发后，对系统进行全面、细致、系统的测试工作，以确保系统功能能够正常、稳定地运行。

1.测试计划：测试计划是指根据系统的设计文档或需求分析文档，对软件、硬件资源的需求进行分析，制定测试计划，具体包括测试性质和重要性、测试环境、测试策略和测试资源等。

2.测试用例的设计：测试用例是预先定义针对系统某个指定的测试点进行测试的方法和步骤，需要根据不同的测试点设计不同的测试用例，在测试用例设计时需要注意覆盖率、测试用例的有效性和可靠性等。

3.测试执行：在实施测试前，需要安装不同的测试工具和环境，并模拟真实的使用场景，对不同的测试点依次进行测试。

4.测试结果的收集和分析：需要收集测试结果，并逐一对测试的功能和对应的测试用例进行分析，查找系统中可能存在的缺陷。

5.缺陷管理：当发现系统或应用程序的缺陷时，需要对缺陷进行记录、追踪、分析，最终在下一个版本的开发过程中进行修复。

6.测试报告：在测试过程结束后，需要根据测试用例的执行情况，生产缺陷列表和测试总结报告等相关文档，为开发人员提供依据，以提高下一个版本的开发质量。

综上所述，测试是系统开发过程中不可或缺的一部分，测试的全面性和准确性对于系统可靠性和完整性具有关键性的作用。测试流程需要在系统实现和上线前充分展开，并逐一核查各个环节的质量和稳定性，只有在测试全面而有效的情况下，才能为用户提供更加优质和高效的系统使用体验。

## 7.4 测试用例

### 7.4.1 非法输入测试

### 7.4.2 图片上传测试

总结

参考文献

致谢