仲恺农业工程学院

毕 业 设 计

基于WEB的网络安全学习网站的设计

姓 名 叶凯佳

院（系） 信息科学与技术学院

专业班级 网络工程202

学 号 202010254227

指导教师 陈文文

职称/职务 讲师

论文答辩日期 2024年5月日

仲恺农业工程学院教务处制

**Design of a cybersecurity learning website Based on Web**

Ye Kaijia

College of Information Science and Technology  
Zhongkai University of Agriculture and Engineering  
Guangzhou,China

**Supervisor: Wenwen Chan**

**学生承诺书**

本人郑重承诺：所呈交的毕业论文（设计）是本人在导师的指导下独立进行研究所取得的成果。除了文中已用特别标志加以标记的引述内容之外，本论文（设计）不含有任何其他个人或集体已经发表或撰写的研究成果。对本文研究做出过重要贡献的个人或集体，均已在文中以明确的方式标明。若在毕业论文（设计）的各项检查、评比中被发现有抄袭、剽窃或其他的违规行为，本人愿按学校有关规定接受处理，并承担相应的法律责任。

学生（签名）：

年 月 日

摘 要

随着社会、经济和文化的数字化转型，人们对互联网的依赖性越来越强。金融、医疗、能源、交通等领域的数字化使得网络安全问题的解决变得尤为迫切，而在教育方面，网络安全教育直至近些年才逐渐被提起与重视，相关的从业人员与学生需要一个提供学习、实践与交流的学习社区环境，通过学习平台，人们能够获取有关网络攻防、网络协议、漏洞分析等方面的知识，能够实践使用网络攻防的工具，从而更好地理解和应对复杂的网络威胁。本平台使用时下流行的技术开发前后端，前端采用了Vue完成界面的搭建与设计，服务端采用了Java+SpringBoot框架,利用SSH协议和WebSocket通讯协议等技术实现、提供用户一个具有博客社区、网络攻防实践、密码加解密功能的网络安全学习平台。

关键词：网络安全学习平台 WebSocket Kali-Liunx Vue JavaSpringBoot

**Abstract**

With the digitization transformation of society, economy, and culture, people's dependence on the Internet is increasing. The digitization of sectors such as finance, healthcare, energy, and transportation has made solving cybersecurity issues particularly urgent. In the field of education, cybersecurity education has only recently gained attention, and professionals and students in this domain require a learning community environment that facilitates learning, practice, and communication. Through a learning platform, individuals can access knowledge about network attacks and defense, network protocols, vulnerability analysis, and more. This enables practical use of tools for network attack and defense, enhancing the understanding and response to complex cybersecurity threats. The platform is developed using popular technologies for both front-end and back-end. The front-end is built and designed using Vue, while the back-end utilizes Java and the SpringBoot framework. The platform leverages technologies such as SSH and WebSocket communication protocols to provide users with a cybersecurity learning platform featuring a blog community, network attack and defense practice, and cryptography study functionalities.

**Keywords:** Cybersecurity learning platform; WebSocke; Kali-Liunx; Vue; Java; SpringBoot;

目 录

[1 前言 1](#_Toc72865027)

[1.1 研究背景及意义 1](#_Toc72865028)

[1.2 国内外研究现状 2](#_Toc72865029)

[1.3 本文研究内容 2](#_Toc72865030)

[1.4 文本结构 2](#_Toc72865031)

[2 相关理论和技术介绍 3](#_Toc72865032)

[2.1 Node.js相关简介 3](#_Toc72865033)

[2.2 Express框架相关简介 3](#_Toc72865034)

[2.3 Vue.js框架相关简介 4](#_Toc72865035)

[2.4 WebSocket相关简介 4](#_Toc72865036)

[2.5 小程序相关简介 5](#_Toc72865037)

[2.6 Echarts相关简介 5](#_Toc72865038)

[2.7 Mongo相关简介 5](#_Toc72865039)

[3 系统分析 6](#_Toc72865040)

[3.1 市场调研 6](#_Toc72865041)

[3.2 可行性分析 6](#_Toc72865042)

[3.2.1 技术可行性 6](#_Toc72865043)

[3.2.2 经济可行性 6](#_Toc72865044)

[3.2.3 市场可行性 7](#_Toc72865045)

[3.3 需求分析 7](#_Toc72865046)

[3.3.1 用户人群 7](#_Toc72865047)

[3.3.2 用户需求描述 7](#_Toc72865048)

[3.3.3 竞品分析 7](#_Toc72865049)

[3.3.4 需求概述 11](#_Toc72865050)

[3.3.5 功能分析 12](#_Toc72865051)

[3.4 系统用例图 13](#_Toc72865052)

[3.5 数据流程图 14](#_Toc72865053)

[3.6 数据字典 14](#_Toc72865054)

[4 系统概要设计 19](#_Toc72865055)

[4.1 系统设计原理 19](#_Toc72865056)

[4.2 系统开发环境与工具 19](#_Toc72865057)

[4.3 系统整体架构 19](#_Toc72865058)

[4.4 系统功能结构设计 20](#_Toc72865059)

[4.5 数据库设计 21](#_Toc72865060)

[4.5.1 设计原则 21](#_Toc72865061)

[4.5.2 概念结构设计 22](#_Toc72865062)

[4.5.3 物理结构设计 24](#_Toc72865063)

[5 系统详细设计 28](#_Toc72865064)

[5.1 系统时序图 28](#_Toc72865065)

[5.2 系统流程图 29](#_Toc72865066)

[5.3 Web端前台模块 30](#_Toc72865067)

[5.3.1 首页模块 30](#_Toc72865068)

[5.3.2 图库模块 32](#_Toc72865069)

[5.3.3 预约摄影模块 34](#_Toc72865070)

[5.3.4 博客模块 36](#_Toc72865071)

[5.3.5 社交圈子模块 37](#_Toc72865072)

[5.3.6 我的模块 37](#_Toc72865073)

[5.3.7 登陆注册模块 39](#_Toc72865074)

[5.4 小程序端前台模块 40](#_Toc72865075)

[5.4.1 首页模块 40](#_Toc72865076)

[5.4.2 订单模块 43](#_Toc72865077)

[5.4.3 图册模块 44](#_Toc72865078)

[5.4.4 文章模块 45](#_Toc72865079)

[5.4.5 我的模块 47](#_Toc72865080)

[5.5 后台管理模块 49](#_Toc72865081)

[5.5.1 内容管理模块 49](#_Toc72865082)

[5.5.2 用户管理 50](#_Toc72865083)

[5.5.4 在线咨询模块 51](#_Toc72865084)

[5.5.5 统计模块 52](#_Toc72865085)

[6 系统测试 52](#_Toc72865086)

[6.1 测试需求 52](#_Toc72865087)

[6.2 测试目标 53](#_Toc72865088)

[6.3 测试用例 53](#_Toc72865089)

[6.4 测试总结 55](#_Toc72865090)

[7 部署上线 55](#_Toc72865091)

[8 总结与展望 55](#_Toc72865092)

[参 考 文 献 58](#_Toc72865093)

[致 谢 59](#_Toc72865094)

# 1 前言

## 1.1 研究背景及意义

在当前数字化转型的趋势下，人们对互联网的依赖加深，各行业也在加速数字化。然而，这种转型不仅带来便利，也引发了网络安全挑战。金融、医疗、能源、交通等领域的数字化转型为生产生活带来了更大的便利，但与此同时也带来了巨大的网络安全挑战。

互联网的普及使信息传递、存储和共享更便捷，但也伴随着恶意软件、网络攻击和数据泄露等威胁。缺乏成熟的网络安全学习平台使得解决这些问题变得更加紧迫。

在这个背景下，建立一个网络安全学习社区显得尤为重要。互联网的普及虽然带来了信息传递、存储和共享的便利，但也伴随着各种网络威胁，如恶意软件、网络攻击和数据泄露等。这些问题使得网络安全变得愈发复杂和紧迫，当前市场上尚未有一个成熟的、提供实践的网络安全学习平台，这为构建一个全面、系统的学习社区提供了契机。

在信息化时代，机遇与挑战同在，网络安全已经不仅仅关乎个人和企业，更关系到整个国家的安全和发展。习近平总书记多次提到网络安全与信息化是事关国家安全和国家发展、事关广大人民群众工作生活的重大战略问题。这表明网络安全已经逐渐上升到国家战略层面，对于构建网络强国的目标至关重要。

建设网络安全学习社区能够提供全面、系统的培训，帮助专业人才更好地应对复杂多变的网络安全威胁。通过学习社区，学生可以获取有关网络攻防、网络协议、漏洞分析等方面的知识，并实践使用网络安全工具。学习社区的建成将促进网络安全专业人才之间的合作与交流，为技术创新和职业发展提供更多机会。

最重要的是，网络安全学习社区的建设符合国家网络安全工作的重要内容。习近平总书记多次强调提升全民网络安全意识和技能，构建网络安全学习社区将为这一目标提供有力支持。通过这样的社区，不仅可以培养更多的网络安全专业人才，也能够普及网络安全知识，提高全民网络安全素养。

总的来说，构建网络安全学习社区不仅填补了市场空白，更对于网络安全的教育而言具有深远的意义。通过提供更全面、系统、实用的学习环境，学习社区将帮助初学者和专业者更好地应对日益复杂的网络安全挑战，为国家的网络安全工作做出积极贡献。网络安全学习社区不仅是一个技术平台，更可以是国家网络安全战略的一颗螺丝，为网络强国的建设贡献力量。

## 1.2 国内外研究现状

国内外和网络安全学习相关的社区有freebuf社区、HackMyVM社区、flux社区等,各个社区都有各自的聚集特点与用户对象群体。其中freebuf社区更注重教学知识的交流，HackMyVM社区更多是提供了一个虚拟机文件供用户下载安装，并交流攻破虚拟机的心得，flux社区更多是在更新一些前沿的系统/网站漏洞。

## 1.3 本文研究内容

本文研究内容是从新手学习角度出发进行需求分析，设计一个面向网络安全学习的新手人群的，拥有实践学习和理论学习功能结合的一个网络安全学习社区，结合本人的技术栈进行开发搭建具有实践性的学习环境，包括模拟攻击场景，论坛交流知识，在线加密解密等。

## 1.4 文本结构？？？

# 2 相关理论和技术介绍

## 2.1 Java语言相关简介

Java是一种跨平台、面向对象的编程语言，由Sun Microsystems（现在是Oracle Corporation）于1995年发布。它的设计目标是实现一次编写，到处运行（WORA），即编写一次代码，可以在任何支持Java的平台上运行，无需重新编译。这使得Java成为一种非常流行的语言，用于开发各种类型的应用程序，包括桌面应用程序、Web应用程序、移动应用程序和大型企业级应用程序。

它具有如下特点：

1. 跨平台性：“Write Once, Run Anywhere（一次编写，随处运行）”，Java的跨平台性是其最重要的特点之一。通过Java虚拟机（JVM），Java程序可以在不同的操作系统上运行，只需编写一次代码，就可以在Windows、Linux、macOS等系统上运行。
2. 面向对象：Java是一种纯粹的面向对象编程语言，支持类、对象、继承、多态等面向对象的特性，使得代码结构清晰、模块化程度高，易于理解和维护。
3. 高性能：Java通过即时编译技术（Just-In-Time Compilation）实现了高性能的运行效率，使得Java应用程序在执行速度上与C/C++等编程语言相媲美。
4. 安全性：Java具有严格的安全机制，包括字节码验证、安全管理器、安全沙箱等，可以有效防止恶意代码的执行，保护系统安全。

除此之外，Java本身作为一个多年的成熟的语言，其生态环境是几乎其他所有语言难以企及的，使用Java语言开发后端从很大程度上减少了本系统开发的工作量。

## 2.2 SpringBoot框架相关简介

Spring Boot是由Pivotal团队提供的一套开源框架，是基于简化spring应用的创建及部署为开发目的发行的，它提供了丰富的Spring模块化支持，通过简化配置和提供各种开箱即用的功能，可以帮助开发者更轻松快捷地构建出企业级应用。

它具有如下四个核心特性：

* 自动装配：Spring Boot通过约定大于配置的原则，大大简化了应用程序的配置过程。它提供了默认的配置选项，简单的配置甚至零配置即可运行
* 编码简单化：提供大量注解简化编码，提高效率
* 起步依赖整合：将所需的常见依赖按组聚集在一起，形成单条依赖
* 部署简单：Spring Boot提供了嵌入式的Web服务器（如Tomcat、Jetty等），可以直接打包成可执行的JAR文件，从而简化了应用程序的部署和启动过程。

本系统的后端架构采用了SpringBoot开发，结合Maven引入其他依赖，与多个数据库连接操作数据，实现了对数据的安全保护以及业务逻辑层面的增删改查。

## 2.3 Vue.js框架相关简介

Vue.js是一款用于构建用户界面的 JavaScript 框架。它基于标准 HTML、CSS 和 JavaScript 构建，并提供了一套声明式的、组件化的编程模型，帮助开发者高效地开发用户界面。

Vue.js的主要特点包括：

1. 轻量级: Vue.js相比于其他框架更轻量级，体积小，学习曲线平缓，易于上手。
2. 双向数据绑定: Vue.js采用了响应式数据绑定的机制，实现了视图和数据的双向绑定，当数据发生变化时，视图会自动更新。
3. 组件化: Vue.js将页面拆分为多个组件，每个组件具有独立的逻辑和样式，便于代码复用和维护。
4. 虚拟DOM: Vue.js通过虚拟DOM技术，将页面更新的成本降到最低，提高了页面渲染的效率。
5. 指令系统: Vue.js提供了丰富的指令系统，例如v-bind、v-if、v-for等，用于简化模板代码和实现各种交互效果。
6. 生命周期钩子: Vue.js提供了多个生命周期钩子函数，允许开发者在不同阶段添加自定义逻辑，方便进行页面初始化、数据更新等操作。
7. 社区支持: Vue.js拥有庞大的开发者社区和丰富的第三方插件，提供了丰富的资源和支持。

本系统前端方面使用了vue2作为开发框架，将各个页面抽离成一个或者多个组件组合，构建了一个相对完善的前端界面，同时结合后端做了双端访问验证，同时保证了页面和接口访问双保障。

## 2.4 WebSocket相关简介

WebSocket是HTML5的一种通信协议，与传统的HTTP通信方式相比，它具有许多优势和特点：

1. WebSocket支持全双工通信。这意味着在建立连接后，客户端和服务器可以同时向对方发送和接收数据，而无需等待对方的响应。这种特性使得WebSocket非常适合实时通讯应用，包括本系统所需要的和liunx交互等，因为它能够实现即时的数据传输和交互。
2. WebSocket也具有较低的通信延迟和数据传输量。与每次HTTP请求都需要携带完整的HTTP头信息不同，WebSocket建立连接后只需在开始时进行一次握手，之后的数据传输只需要少量的头部信息，这大大减少了通信时的数据传输量和延迟，提升了系统的性能和效率。
3. WebSocket还具有跨域通信的能力。虽然在Web开发中跨域通信通常受到限制，但是WebSocket协议可以在不同域名之间进行通信，这为跨域应用场景提供了更多的可能性。

在本系统中，利用了WebSocket来连接前端和后端，并且后端再使用WebSocket连接到Kali Linux。这种架构使得前端通过后端和Liunx连接，使整个系统具备了实时的通信能力，可以实现用户和liunx的实时操作交互和反馈。例如，在使用IP扫描的时候，前端只需要在Kali模块输入对应命令或者按钮就可以实时扫描服务器的liunx所能扫描的ip。

## 2.5 Mongo相关简介

MongoDB属于文档型数据库，数据模型结构都是树形的。在功能和操作上与关系型数据库十分相似，是功能最丰富的非关系型数据库。MongoDB有着和表相似的集合，有数据库、使用B树实现索引。MongoDB使用的是在计量级JSON上进行了扩展的BSON保存无结构数据。MongoDB有着以下优点和特性：

1. 高性能，基于文档型的数据结构，在一些数据不需要严格规定的场景下，Mongo有着更好的性能，并且它提供与关系型数据库同样强大的查询功能，并支持索引。
2. 面向集合存储，对于Web面向广大用户的项目，方便存储复杂且多变的对象类型数据。
3. 实用性高，直接存取BSON更加灵活，不受初始化格式的限制可以插入复杂的数据类型，可以无需定义。

系统为面向大众使用的产品，并且需要存储大量复杂类型的数据如对象、数组等，并且需要提供快速查询，高性能支持提高用户的使用体验。

# 3 系统分析

## 3.1 市场调研？？？

通过问卷调查的方式，可以看出无论男性或者女性，70%都对摄影有或多或少的兴趣，其中女性对摄影感兴趣和摄影师需求均比男性略高，可以看出女性群体在这个行业中存在更多潜在客户。并且有图库使用需求的人在各个职业领域均有分布，从事设计行业的人去需求尤为强烈，且大多数人支持知识版权付费。

## 3.2 可行性分析

### 3.2.1 技术可行性

本系统前端使用时下构建页面最流行的Vue.js框架，后端则采用了Java+SpringBoot的技术栈，并采用时兴的前后端分离开发的模式，在提高了开发灵活性和可维护性的同时，系统的可扩展性也大大提高，前后端增删其他功能变得相对独立与不耦合，一些功能的修改也变得非常灵活。

数据库方面，大部分数据存储在MySql数据库，一些热数据存储在Redis，文档型数据则存放在MongoDB，关系型数据库和非关系型数据库结合，极大程度上保证了数据的存放于拿取的性能保证。

版本选用方面，Vue版本采用了相对稳定的Vue2，后端考虑到JDK21发布时间，采用了SpringBoot3+JDK17开发，MySql使用了稳定的8.1.0版本，Redis使用了7.2.1版本，MongoDB使用了7.0.5版本。

综上，本系统从系统稳定流畅和开发难度方面技术性分析,选用了合适的技术栈和对于的版本进行开发，保证开发进度的正常推进。

### 3.2.2 经济可行性

本系统中使用的技术栈均为开源技术,不需要再额外付费。使用的开发工具vscode和idea均是社区版本而不是付费版本，数据库方面MySql、Redis和MongoDB也是免费可以的数据库。

本系统开发过程全由本人完成，毋须借由他人之手，因此不需额外付出其他费用，从经济角度上看本系统具有极高的经济可行性。

### 3.2.3 市场可行性

网络空间已经成为继陆、海、空、天之后的第五疆域。近年来，网络空间安全备受重视，不少高校陆续开设了网络空间安全专业。而比起传统的计算机专业，网络安全的学习社区的构建远远比不上类似于开发、运维等方向，因此一个良好的可供实践与交流的社区能够极大的降低新手入门的门槛。

也就是说，面向新手学习的用户，本系统能够为他们解决一定程度上的问题，并且几乎没有同类社区的竞争，因此本系统具备良好的市场前景，存在较大的市场可行性。3.3 需求分析

### 3.3.1 用户人群

本系统面向所有学习网络安全方向的新手提供了一个交流与实践的平台。

### 3.3.2 用户需求描述

目前国内外的相关方向的社区对大多数初学者的使用门槛相对较高，并且更多是理论上的传递与交流，几乎找不到实践和交流结合的学习社区，因此用户需要一个提供自带攻击机和靶机的网站，这对新手用户来说是大大利好他们初学者学习的。

并且网站额外携带的论坛功能能够方便他们和有经验者的交流，有助于新手进步。

### 3.3.3 竞品分析

国内外主流的网络安全相关的学习社区有如下几个。

1. 产品定位及优势对比

表1 同类产品介绍

|  |  |  |
| --- | --- | --- |
|  | 产品定位 | 特色功能 |
| FreeBuf | FreeBuf 是一个专注于网络安全领域的社区平台，致力于为网络安全爱好者提供全面的安全信息和资源。 | FreeBuf提供及时、权威的安全新闻和资讯，涵盖漏洞报告、安全事件、攻击趋势等内容，帮助用户了解安全行业的最新动态。 |
| HackMyVM | HackMyVM 是一个虚拟机（VM）平台社区，旨在为安全研究人员和渗透测试者提供实验和学习的环境。 | HackMyVM让用户能够进行真实的渗透测试和安全演练，用户可以与其他安全从业者交流经验、分享技术。 |
| OWASP | OWASP（Open Web Application Security Project）是一个开源的、非营利性的国际性组织，致力于促进Web应用程序的安全性和安全开发实践。 | 开发了各种安全工具如：OWASP ZAP、OWASP Dependency-Check等，用于辅助开发者进行安全测试和漏洞扫描。提供了丰富的安全指南和文档，包括安全开发指南、渗透测试指南等，为开发者和安全从业者提供参考和指导。 |

（2）功能对比

表2 功能对比

|  |  |  |  |
| --- | --- | --- | --- |
|  | 提供安全资讯 | 提供安全工具 | 实战环境 |
| FreeBuf | 支持 | 支持 | 不支持 |
| HackMyVM | 不支持 | 不支持 | 支持 |
| OWASP | 支持 | 支持 | 不支持 |

（5）总结：

表3 竞品总结

|  |  |  |
| --- | --- | --- |
| 产品 | 专注点 | 内容板块 |
| FreeBuf | 专注于社区属性，提供交流平台。 | 安全新闻、漏洞分析、安全工具分享 |
| HackMyVM | 专注于实践属性，提供虚拟实验环境，为安全研究人员和渗透测试者提供实践和学习的平台。 | 漏洞靶场资源提供、CTF挑战举办、学习资源分享 |
| OWASP | 专注于百科属性，提供各种安全工具分享和安全操作指南。 | OWASP Top 10（高危漏洞排行榜）、安全工具下载、安全指南分享 |

以小见大，市面上主流的安全社区几乎没有一个结合了实践与交流的例子，特别对于入行门槛高的网络安全领域，急需要一个低入门门槛和结合实践与交流的社区来改变现状。

### 3.3.4 需求概述

首先本系统是面对广大新手入门的网络安全方向的学习者，应该提供更高可用和低门槛学习的能力，因此比起传统的工具与社区，本系统需要部署好实践学习需要的工具与环境，本系统的目标用户为新手入门的初学者和乐于分享技术交流的熟手，系统应该具备管理员对网站论坛和用户群体进行管理。

用户体验上应该具备以下条件：

1. 实践环境易于入门，简便操作
2. 论坛环境交流方便，操作符合逻辑
3. 整体页面布局整齐，样式偏现代化

功能上应该满足以下条件：

1. 提供实践操作功能
2. 提供论坛交流功能
3. 提供用户管理功能（管理员）
4. 提供实践教学功能

### 3.3.5 功能分析

根据需求分析可以得出，本系统核心为交流和实践两个方向，其中交流方向通过构建论坛实现，而实践的方向则是交由时下热门的网络安全工具Kali\_Liunx和常用的Metasploitable2 靶机环境来完成，核心的两个方向的功能面向所有用户，而管理员对应的功能则是通过动态路由附加。

面向用户的包括以下功能：  
 （1）首页论坛模块：用户在登录鉴权后默认跳转到首页论坛模块，论坛默认展示目前存在的帖子，以及发布帖子、浏览帖子的功能，通过上方搜索框可以通过关键字或者标题搜索帖子，同时用户可以通过全局存在的导航栏跳转到其他模块。  
 （2）密码加解密模块: 用户点击导航栏中的密码加解密进入密码加解密模块，模块提供一个操作界面，用户可以通过选择加密或者解密功能，在输入对于的密钥，挑选对于的算法后完成对明文的加密/密文的解密。  
 （3）Kali-Liunx模块：用户点击导航栏中的“Kali-Liunx”进入Kali-Liunx模块，在本系统里，Kali-Liunx作为攻击机安装了绝大多数的工具，并且网卡做了桥接，用户可以在这里通过操作命令完成对两个靶机的侦测与攻击。  
 （4）Msf模块：用户点击导航栏中Msf可以进入Msf模块，在本系统里，特意挑选了Metasploitable2作为 靶机环境之一，作为靶机，Metasploitable2可以用于进行渗透测试训练和演练。  
 （5）个人中心模块：用户点击右上角头像的“个人信息”进入个人中心模块，可以查看个人信息和自己发布的帖子等。

（6）信息修改模块：用户点击导航栏的“修改信息”，可以进入信息修改模块，可以修改自己的昵称、邮箱、密码、手机号码等。

管理员额外附加以下功能：  
 （1）用户管理模块：可以对所有ROLE\_USER级别的用户的所有信息进行增删改查。

（2）帖子管理模块：可以对所有帖子进行增删改查。

（3）命令管理模块：可以对所有攻击机的按钮进行增删改查。

## 3.4 系统用例图

用例图通过使用小人、椭圆、连线三种元素，最简单的形式展示用户与系统的交互关系。每一个椭圆都是一个动作，每一个小人都是一种用户，因此可以描述系统种各个任务角色可以执行的动作。

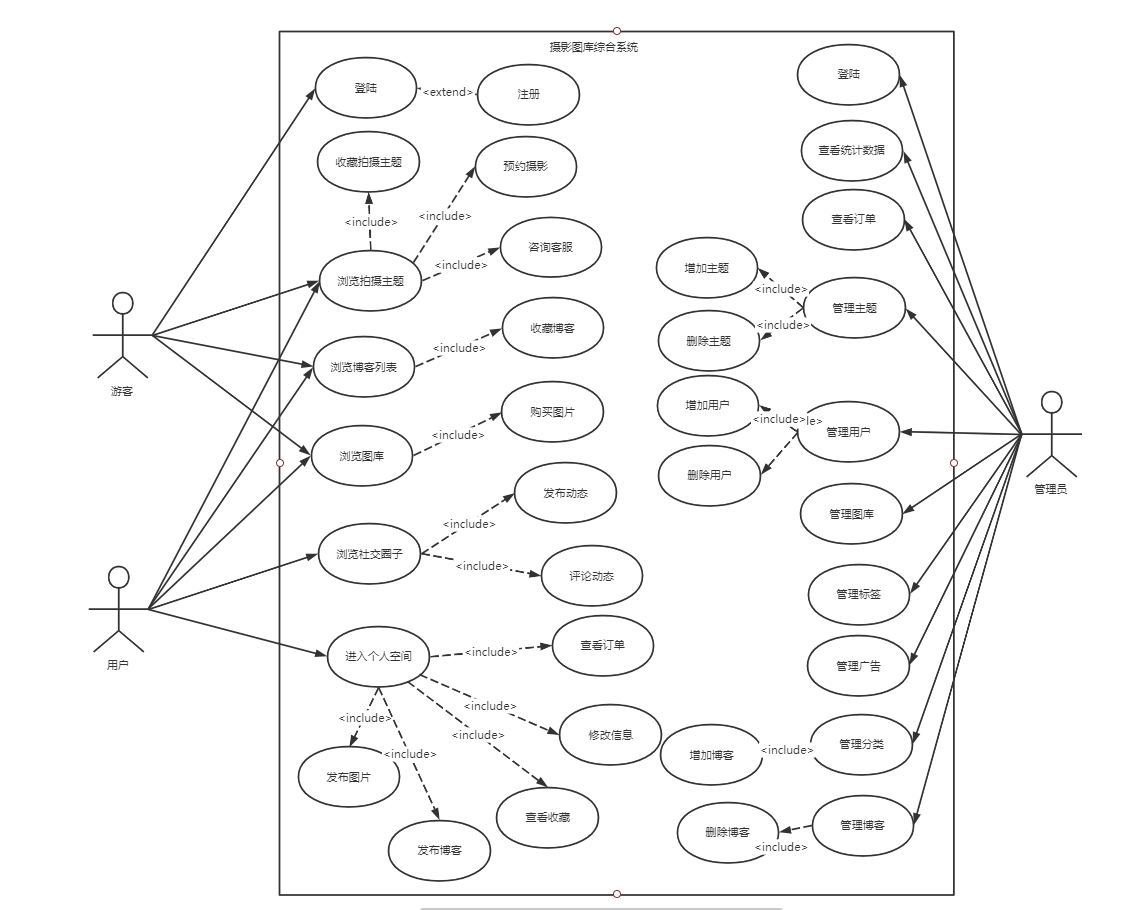


图1 系统用例图

## 3.5 数据流程图

DFD数据流图是一种用于描述系统数据流动和处理过程的图形化工具。它可以帮助人们理解系统内部数据如何流动、被处理和存储的过程。DFD图通常用于系统分析和设计阶段以及软件工程中的需求分析和系统建模。

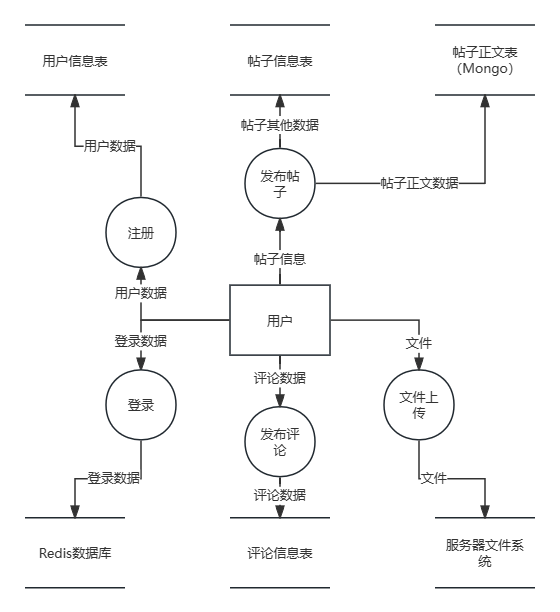


图2 顶层DFD图

## 3.6 数据字典

本系统所有数据字典如下所示：

（1）用户信息数据字典：

表4 用户信息数据字典

|  |  |
| --- | --- |
| 用户信息 | |
| 描述 | 所有用户的信息 |
| 数据结构 | 用户信息=用户ID+头像文件路径+用户名+昵称+密码+邮箱+手机号码+用户类型+用户角色 |
| 数据来源 | 任何人在本系统上进行注册 |
| 数据去处 | 用户信息表sys\_user |

（2）登录信息数据字典：

表5 登录信息数据字典

|  |  |
| --- | --- |
| 登录信息 | |
| 描述 | 所有登录系统者的信息 |
| 数据结构 | 登录系统者的信息=用户名+登录IP+凭证刷新时间+凭证过期时间 |
| 数据来源 | 用户登录 |
| 数据去处 | Redis数据库 |

（3）帖子信息数据字典：

表6 帖子信息数据字典

|  |  |
| --- | --- |
| 帖子信息 | |
| 描述 | 所有帖子的信息 |
| 数据结构 | 帖子信息=帖子ID+标题+帖子图片文件路径+帖子攥写用户ID+发帖时间 |
| 数据来源 | 用户发布帖子 |
| 数据去处 | 帖子信息表article |

（4）帖子正文信息数据字典：

表7 帖子正文数据字典

|  |  |
| --- | --- |
| 帖子正文信息 | |
| 描述 | 所有帖子正文的信息 |
| 数据结构 | 帖子正文信息=帖子正文ID+帖子ID+帖子正文 |
| 数据来源 | 用户发布帖子 |
| 数据去处 | 帖子正文信息表text(Mongo) |

（5）评论信息数据字典：

表8 评论信息数据字典

|  |  |
| --- | --- |
| 评论信息 | |
| 描述 | 所有评论信息 |
| 数据结构 | 评论信息=评论ID+用户ID+用户昵称+用户头像文件路径+被评论帖子ID+被评论评论ID+被回复用户ID+被回复用户名+评论正文+评论时间 |
| 数据来源 | 所有用户发布评论 |
| 数据去处 | 评论信息表comments |

（6）按钮信息数据字典：

表9 按钮信息数据字典

|  |  |
| --- | --- |
| 按钮信息 | |
| 描述 | 实践环境的按钮信息 |
| 数据结构 | 按钮信息=按钮ID+命令内容+创建时间+命令类型 |
| 数据来源 | 管理员创建命令按钮 |
| 数据去处 | 评论信息表comments |

# 4 系统概要设计

## 4.1 系统设计原理

本系统的目的是分析图库需求者他们想要一个拥有什么功能的系统去帮助他们，以及解决找摄影师困难的问题。主要业务为用户在登陆本系统点击进入图库模块，可以根据分类查找图片进行购买。进入预约摄影模块，查看喜欢的摄影主题进入查看详情进行填写信息提交预约。核心思想是为图库需求者、摄影需求者提供、摄影师提供优质的体验。

## 4.2 系统开发环境

（1）硬件配置

处理器:AMD Ryzen 5 4500U with Radeon Graphics 2.38 GHz

机带:RAM 8.00 GB (7.40 GB 可用)

系统类型:64 位操作系统, 基于 x64 的处理器

（2）软件配置

开发IDE:Visual Studio Code、IntelliJ IDEA

数据库: IntelliJ IDEA附带的数据库工具（MySql\MongoDB\Redis）

软件工程绘图工具：ProcessOn

## 4.3 系统整体架构

本系统可以分为三部分，分别是前端系统，后端系统，以及提供实践的虚拟机系统，整体架构如下图所示

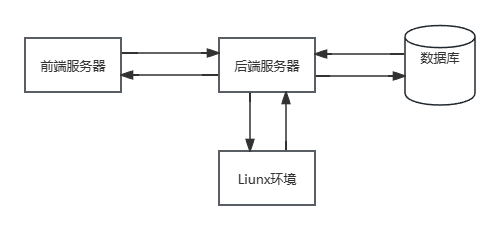


图3 系统整体架构图

## 4.4 系统功能结构设计

（1）系统模块功能设计：

用户注册登录后，可以进入论坛模块、攻击机模块、靶机模块、个人信息模块，密码加解密模块等。管理员用户还可以进入用户管理模块。其详细功能设计图如下所示。

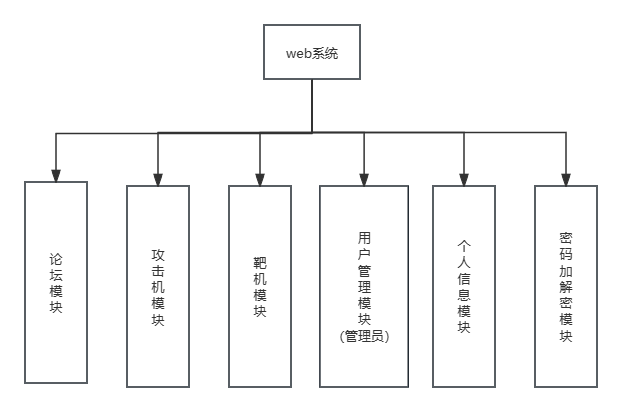


图8 Web前端功能架构

## 4.5 数据库设计

### 4.5.1 设计原则

好的数据库设计无非从数据层面、安全层面以及性能层面三个层面完成。

本系统的数据库设计从数据一致性、性能优化、安全性和易用性四个角度为目的设计，首先数据库命名风格高度统一，并且为了提高易用性按照不同数据的访问热度存放在不同数据库，比如正常信息放在MySql数据库，高并发数据存放在Mongo数据库，访问频繁的小型数据存放在Redis数据库等。

而从实际开发角度来说，本系统减少了数据库的绝大多包括外键等的约束，转而在后端层面控制数据的输入，大大提高了开发效率，减少了数据库查询和写入的开销，提升了整个系统的性能，同时利用索引降低数据冗余并提高数据库的并发处理能力。

### 4.5.2 概念结构设计

本系统采用E-R图作为描述实体之间关系的数据模型，通过各种元素的组合和连接，展示了实体之间的关系，包括一对一、一对多、多对一和多对多等不同类型的关系。本系统的数据库共有5张表，分别是sys\_user（用户表）、article（文章表）、button（命令按钮表）、comments（评论表）、text（文章正文表）。本系统部分实体和数据库E-R图如下：

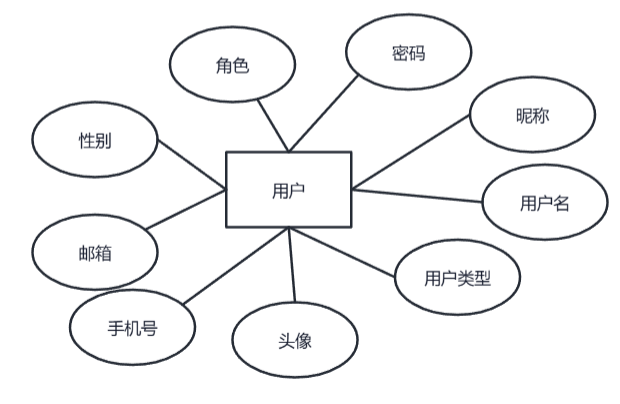


图11 用户实体

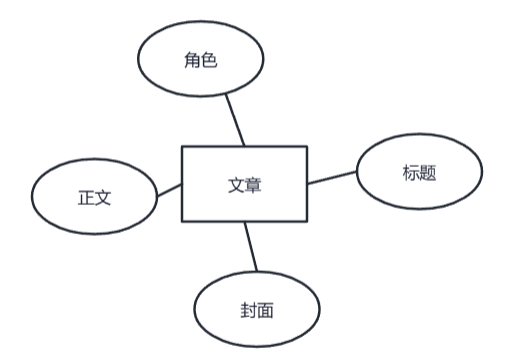


图12 文章实体

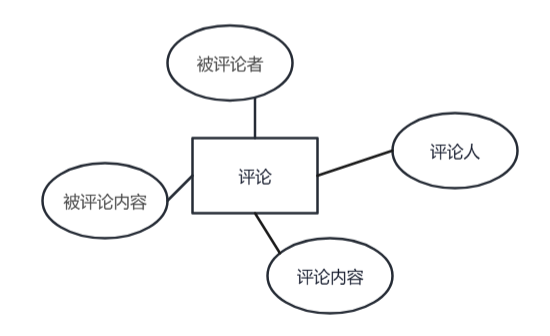


图13 评论实体

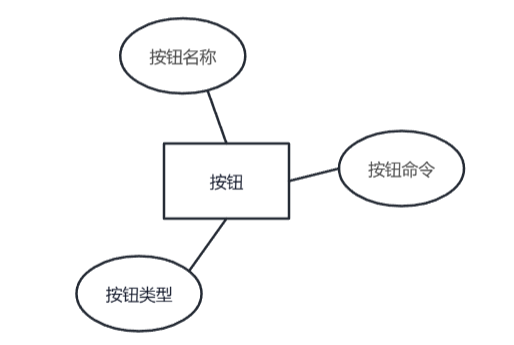


图14 拍摄主题实体

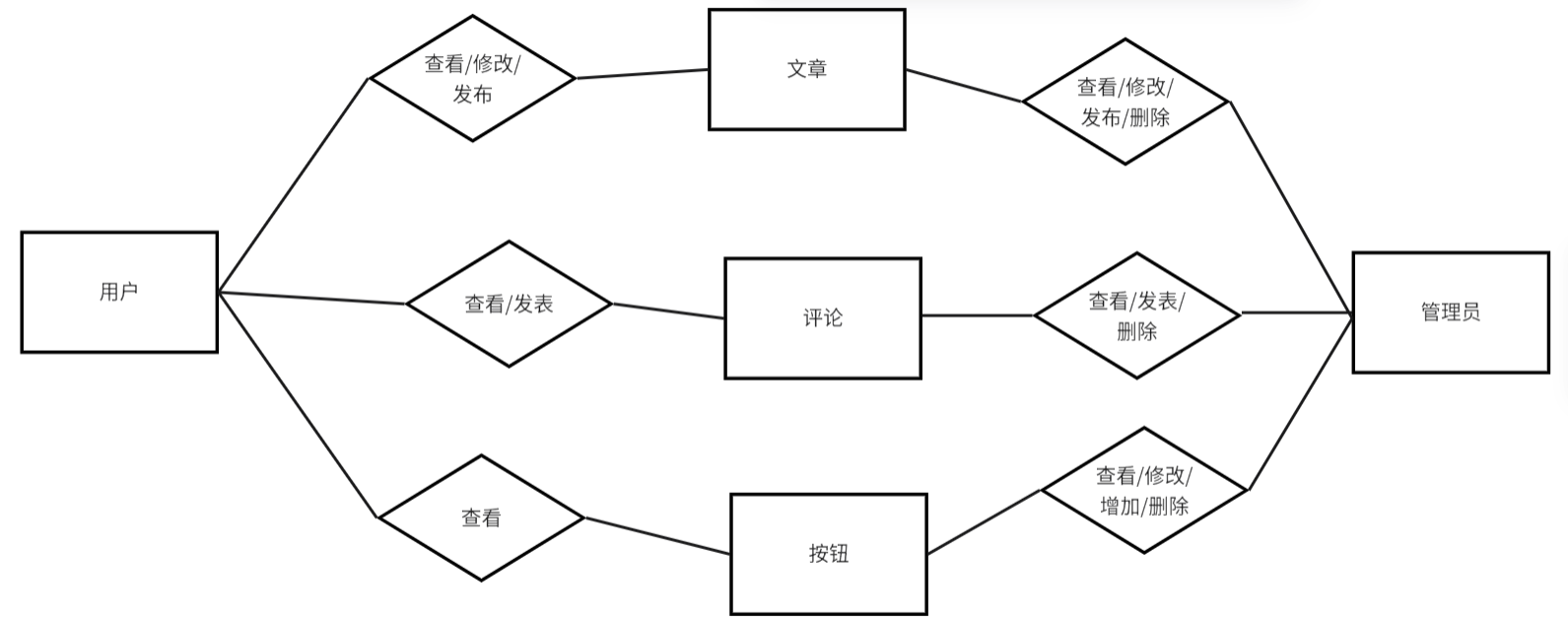


图17 系统E-R图

### 4.5.3 物理结构设计

摄影图库综合系统的表和字段如下所示：

表17 用户表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 字段 | 数据类型 | 主键 | 允许空 | 说明 |
| 1 | id | bigint | 是 | 否 | 主键 |
| 2 | user\_name | varchar(64) |  | 否 | 用户名 |
| 3 | nick\_name | varchar(64) |  | 否 | 昵称 |
| 4 | password | varchar(64) |  | 否 | 密码 |
| 5 | status | char |  | 是 | 账号状态 |
| 6 | email | varchar(64) |  | 是 | 邮箱 |
| 7 | phone\_number | varchar(32) |  | 是 | 手机号 |
| 8 | sex | char |  | 是 | 用户性别 |
| 9 | avatar | varchar(128) |  | 是 | 头像 |
| 10 | user\_type | char |  | 否 | 用户类型 |
| 11 | create\_by | bigint |  | 是 | 创建人的用户id |
| 12 | create\_time | datetime |  | 是 | 创建时间 |
| 13 | update\_by | bigint |  | 是 | 更新人 |
| 14 | update\_time | datetime |  | 是 | 更新时间 |
| 15 | del\_flag | int |  | 是 | 删除标志 |
| 16 | roles | varchar(255) |  | 是 | 角色 |

表18 文章表

| 序号 | 字段 | 数据类型 | 主键 | 允许空 | 说明 |
| --- | --- | --- | --- | --- | --- |
| 1 | id | bigint | 是 | 否 | 主键 |
| 2 | article\_title | varchar(128) |  | 是 | 文章标题 |
| 3 | article\_thumbnail\_url | varchar(128) |  | 是 | 文章缩略图地址 |
| 4 | article\_author\_id | bigint |  | 是 | 文章作者id |
| 5 | article\_view\_count | int |  | 是 | 浏览总数 |
| 6 | article\_preview\_content | varchar(256) |  | 是 | 预览内容 |
| 7 | article\_comment\_count | int |  | 是 | 评论总数 |
| 8 | created\_time | datetime |  | 是 | 创建时间 |
| 9 | updated\_time | datetime |  | 是 | 更新时间 |
| 10 | article\_status | char |  | 是 | 文章状态 |
| 11 | article\_thumbs\_up\_count | int |  | 是 | 点赞总数 |

表19 评论表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 字段 | 数据类型 | 主键 | 允许空 | 说明 |
| 1 | id | bigint | 是 | 否 | 主键 |
| 2 | content | varchar(255) |  | 是 | 评论内容 |
| 3 | author\_nickName | varchar(128) |  | 是 | 评论者昵称 |
| 4 | author\_id | varchar(128) |  | 是 | 评论者id |
| 5 | author\_avatar | varchar(128) |  | 是 | 评论者头像链接 |
| 6 | comment\_id | bigint |  | 是 | 父评论id |
| 7 | article\_id | bigint |  | 是 | 文章id |
| 8 | repaly\_id | bigint |  | 是 | 回复的评论id |
| 9 | created\_time | datetime |  | 是 | 创建时间 |
| 10 | updated\_time | datetime |  | 是 | 更新时间 |
| 11 | repaly\_name | varchar(10) |  | 是 | 回复的user |

表20 命令按钮表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 字段 | 数据类型 | 主键 | 允许空 | 说明 |
| 1 | id | bigint | 是 | 否 | 主键 |
| 2 | command | varchar(255) |  | 是 | 按键对应命令 |
| 3 | description | varchar(128) |  | 是 | 按键描述 |
| 4 | created\_time | datetime |  | 是 | 创建时间 |
| 5 | updated\_time | datetime |  | 是 | 更新时间 |
| 6 | command\_type | tinyint(1) |  | 是 | 命令类型 |

表21 文章正文表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 字段 | 数据类型 | 主键 | 允许空 | 说明 |
| 1 | id | ObjectId | 是 | 否 | MongoDB 主键类型 |
| 2 | id\_article | Long |  | 是 | 整数类型 |
| 3 | article\_content | String |  | 是 | 字符串类型 |
| 4 | article\_content\_html | String |  | 是 | 字符串类型 |
| 5 | created\_time | Date |  | 是 | 日期时间类型 |

# 5 系统详细设计

## 5.1 系统时序图

时序图描述各个系统页面或各层级之间传递消息的时间顺序，是用例图在时间轴上的交互图。用户访问网站，可以选择各个功能页面，若功能页面有权限限制则进行校验，若权限通过则发送请求到服务端请求数据进行展示，用户根据页面的提示输入信息，与系统进行交互同时发送各类请求到服务端，响应结果再给用户。

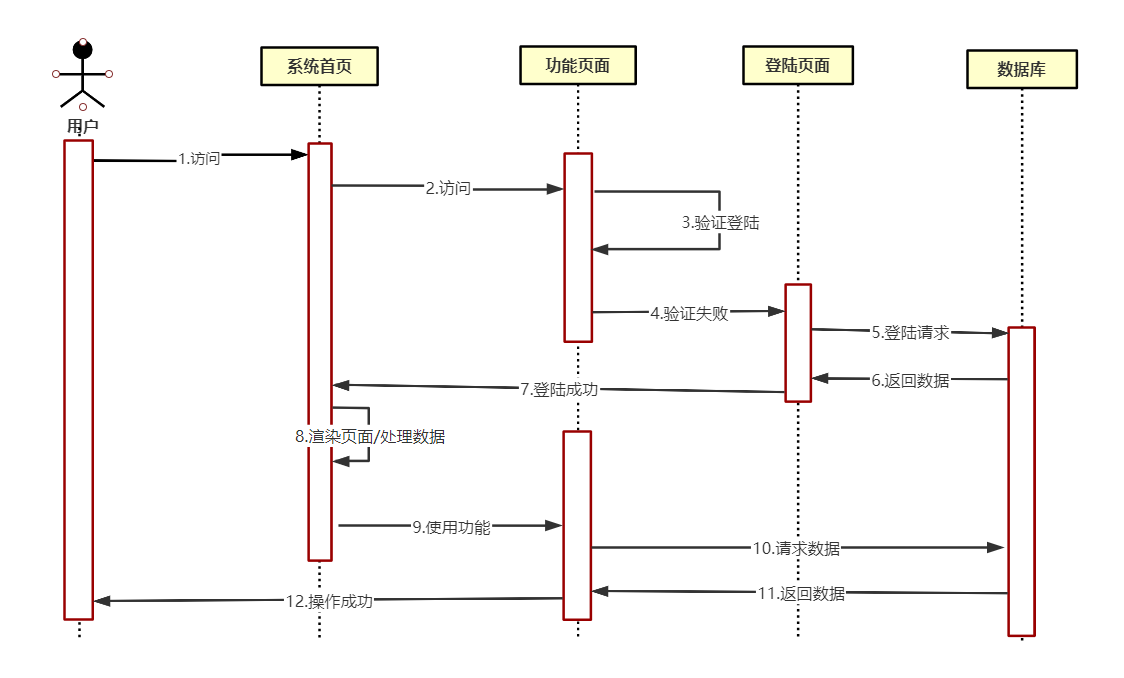


图18 系统总览时序图

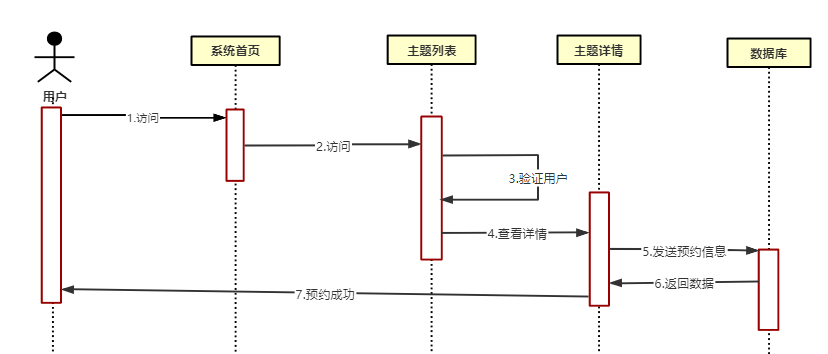


图19 预约拍摄订单时序图

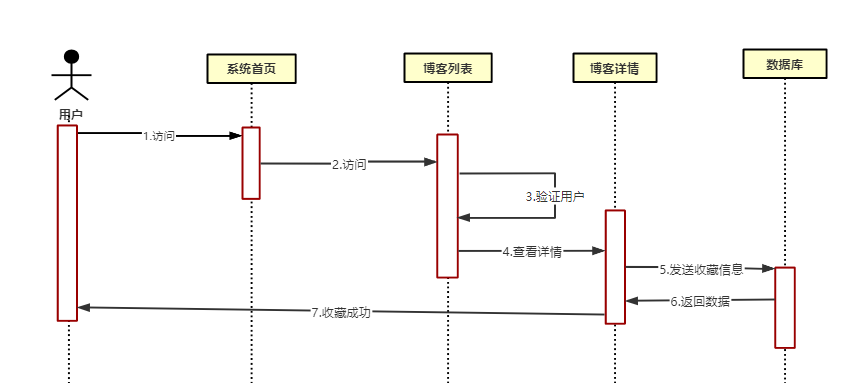


图20 收藏文章时序图

## 5.2 系统流程图

浏览器键入网址将直接鉴权，若登录将直接进入网址对应页面，否则一律重定向至登录页面，输入账号密码后正常登录跳转到首页，首页是本系统的论坛模块，通过论坛模块可以完成对帖子的查看与发布，点击任意帖子可以进入对应的正文页面，可以在帖子对正文以及正文的评论进行评论。

在登录成功后的任意页面，可以通过点击左侧导航栏进入不同的模块，其中进入MSF靶机模块可以对靶机进行操作，进入Kali攻击机模块可以使用攻击机进行操作，进入个人中心页面可以进行对个人信息修改以及退出登录等

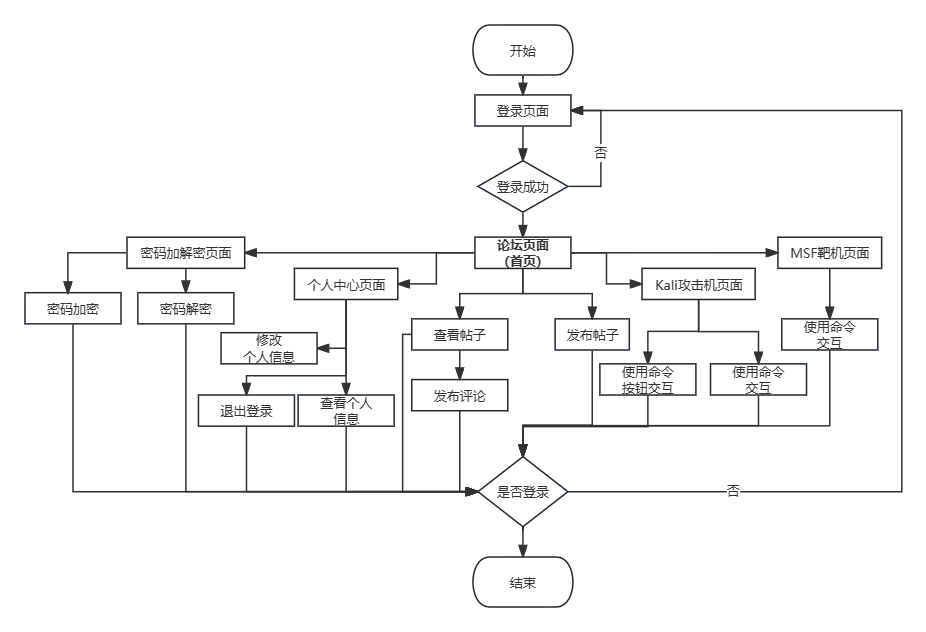


图21 系统总流程图

## 5.3 系统讲解

### 5.3.1 论坛模块与导航栏

系统前后端分离部署，前端使用端口8080，后端使用端口80，登录后默认跳转到论坛界面，论坛帖子使用了MarkDown语法展示，同时可以上传相关与预览正文，帖子正文页面可以发表与查看评论。

导航栏全局显示，其中非管理员不显示且不开放用户管理等管理员功能的路由。

涉及MongoDB的帖子正文部分的增删改查使用MongoTemplate、Query条件查询器以及Criteria条件构造器完成。

部分代码如下：

|  |
| --- |
| @Override  public ArticleContent queryByArticleId(Long id) {  //构造条件查询器  Query query = new Query();  //构造条件  Criteria criteria = new Criteria();  criteria.and("id\_article").is(id);  query.addCriteria(criteria);  //返回查询结果  return mongoTemplate.findOne(query,ArticleContent.class);  } |

涉及MySql的其他部分使用MyBatis以及MyBatis-Plus完成。

|  |
| --- |
| @Mapper  public interface CommentsMapper extends BaseMapper<Comments> {  //通过注解自定义查询语句  @Select("SELECT \* FROM comments WHERE article\_id = ${id} and comment\_id IS NULL ")  List<ComentsDto> getParentList(@Param("id") Long articleId);  @Select("SELECT \* FROM comments WHERE comment\_id = ${id}")  List<Comments> getChildrenList(@Param("id") Long commentId);  }  Service层代码：  @Service  public class CommentsServiceImpl extends ServiceImpl<CommentsMapper, Comments> implements CommentsService {  @Override  @Transactional(propagation = Propagation.REQUIRES\_NEW)  public List<ComentsDto> commentsList(Long id) {  List<ComentsDto> list = baseMapper.getParentList(id);  list=list.stream().peek(comment-> comment.setChildren(baseMapper.getChildrenList(comment.getId()))).toList();  return list;  }  } |

所有后端请求URL构建严格按照RestFul规范编写（其他模块同理不再赘述）：

|  |
| --- |
| @RestController  @RequestMapping("/Comments")  public class CommentsController {  @Autowired  private CommentsService commentsService;  @Autowired  private UserService userService;  @GetMapping("{arctileId}")  public void comments(@PathVariable("arctileId") Long arctileId, HttpServletResponse response){  try{  RestResult.responseJson(response, new RestResult<>(200, "成功！", commentsService.commentsList(arctileId)));  }catch (Exception e){  RestResult.responseJson(response, new RestResult<>(600, "失败", e.getMessage()));  }  }  @PostMapping  public void newComments(@RequestBody Comments comments,HttpServletResponse response){  try{  SysUser sysUser =userService.getById( UserUtils.getCurrentUser().getId());  comments.setAuthorId(String.valueOf(sysUser.getId()));  comments.setAuthorNickname(sysUser.getNickName());  comments.setAuthorAvatar(sysUser.getAvatar());  comments.setCreatedTime(new Date());  commentsService.save(comments);  RestResult.responseJson(response, new RestResult<>(200, "成功！", comments));  }catch (Exception e){  RestResult.responseJson(response, new RestResult<>(600, "失败", e.getMessage()));  }  }  } |

模块结构如图：

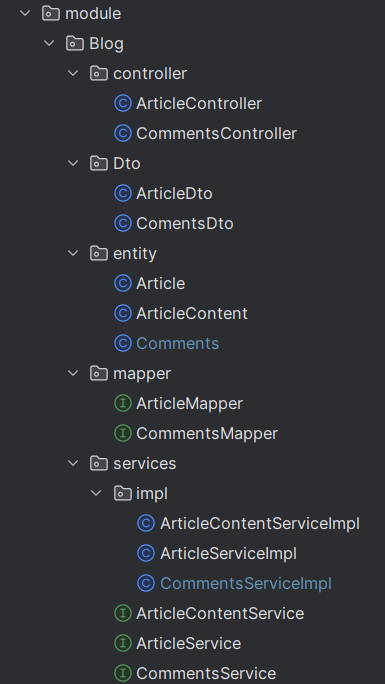


图22 论坛模块后台项目结构图

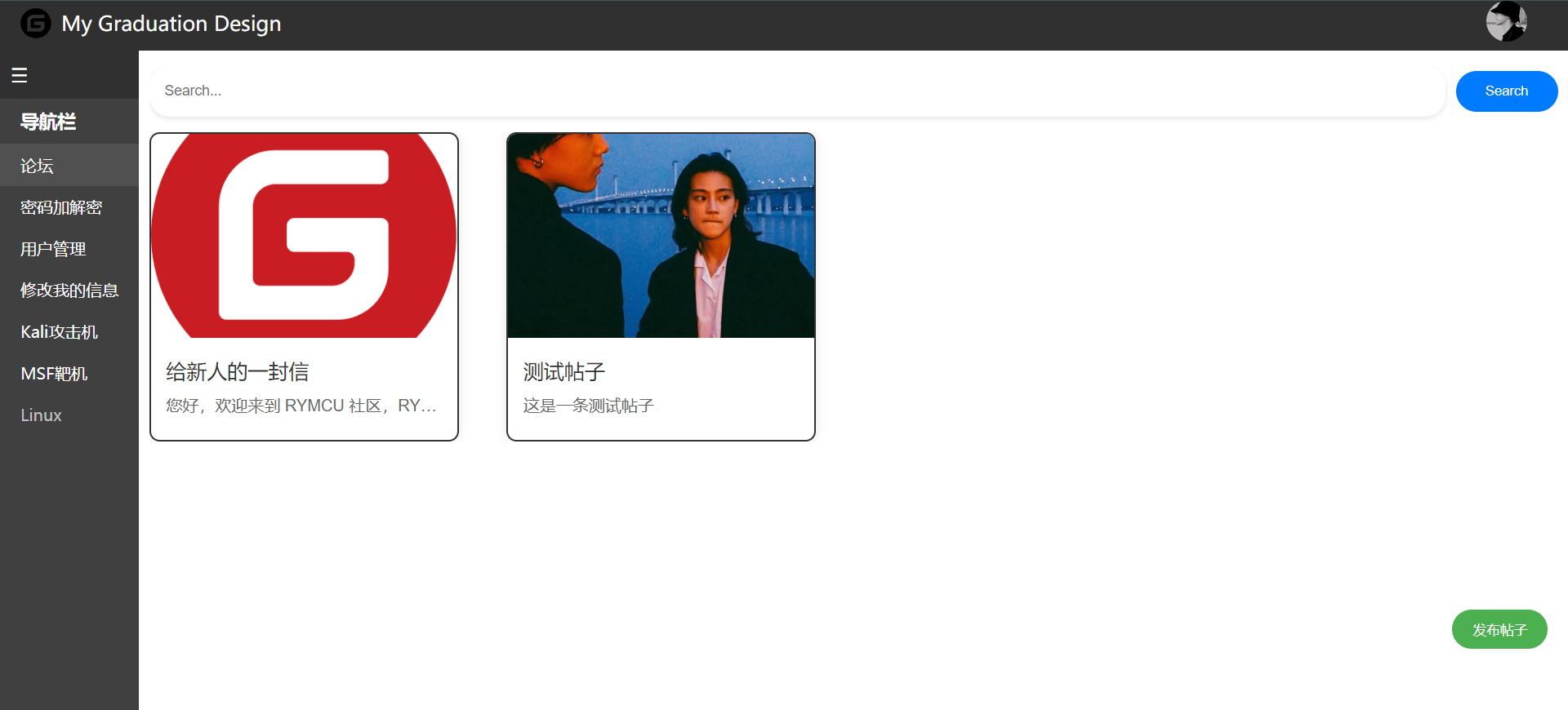


图22 论坛模块与导航栏



图22 发帖预览页面



图22 帖子页面

用户进入论坛模块的同时，前端页面会在渲染完成前向后端请求帖子数据并渲染，选中对应帖子进入对应帖子页面将向后端发送两个请求，第一个请求是向后端通过向MongoDB查询正文数据，第二个请求是向后端拿取评论数据，评论数据分为三级，文章评论、子评论以及回复评论，后端在查询评论后通过stream流完成子评论与回复评论的查询返回给前端。

### 5.3.2 密码加解密模块

通过导航栏点击“密码加解密”，可以跳转到密码加解密界面，通过选择栏可以选择加解密以及加解密的算法。

密码加解密通过前端的crypto-js库完成，部分使用体现代码如下：

|  |
| --- |
| encryptText(plainText, key, algorithm) {  switch (algorithm) {  case 'DES':  return CryptoJS.DES.encrypt(plainText, key).toString();  case '3DES':  return CryptoJS.TripleDES.encrypt(plainText, key).toString();  case 'AES128':  return CryptoJS.AES.encrypt(plainText, key).toString();  default:  return '';  }  } |

界面展示如下：

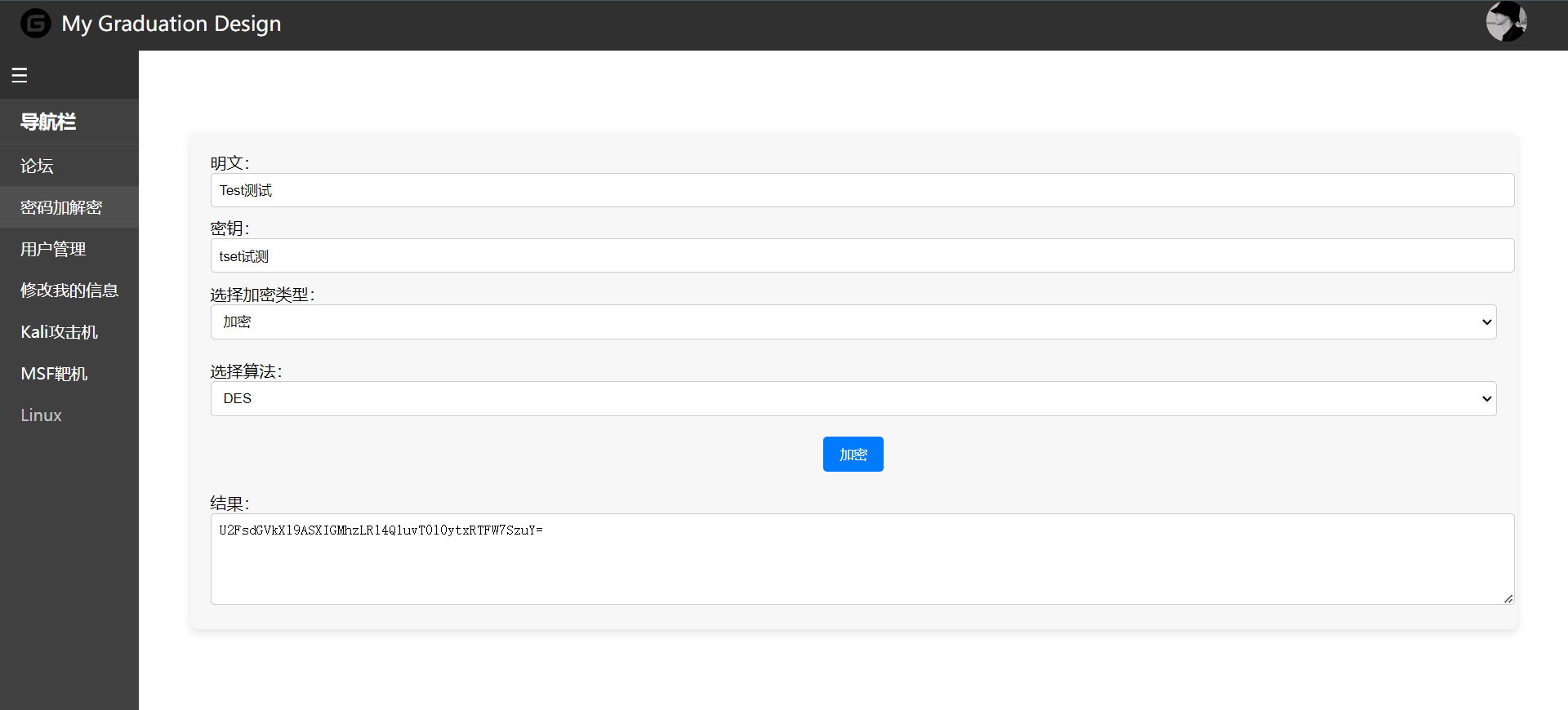


图23 密码加解密页面

### 5.3.3 用户管理模块

用户管理模块是管理员权限才有的功能，在前端登录完成后通过addRouter方法完成动态路由添加后方可访问，普通用户无法通过URL或者其他方法进入此界面。

通过请求后端，管理员（唯一）可以完成对除自己外其他用户进行增删改查。

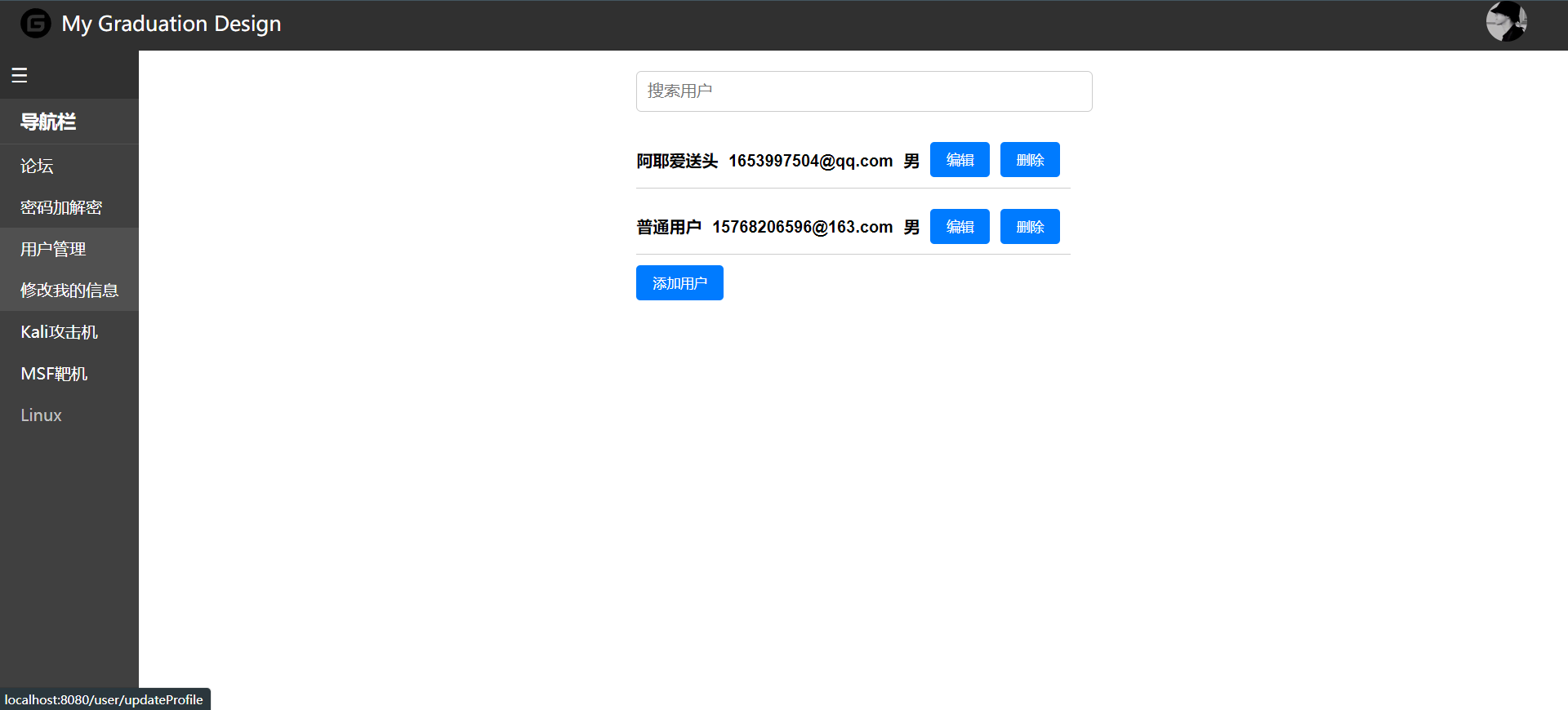


图26 用户管理模块

### 5.3.4 博客模块

系统为用户提供博客模块，用户可以查看其他用户的，以及自己编写博客，专注于相关的主题，摄影爱好者可以再平台上学习其他人的摄影技巧，也可以分享自己的游记或者拍摄经历。

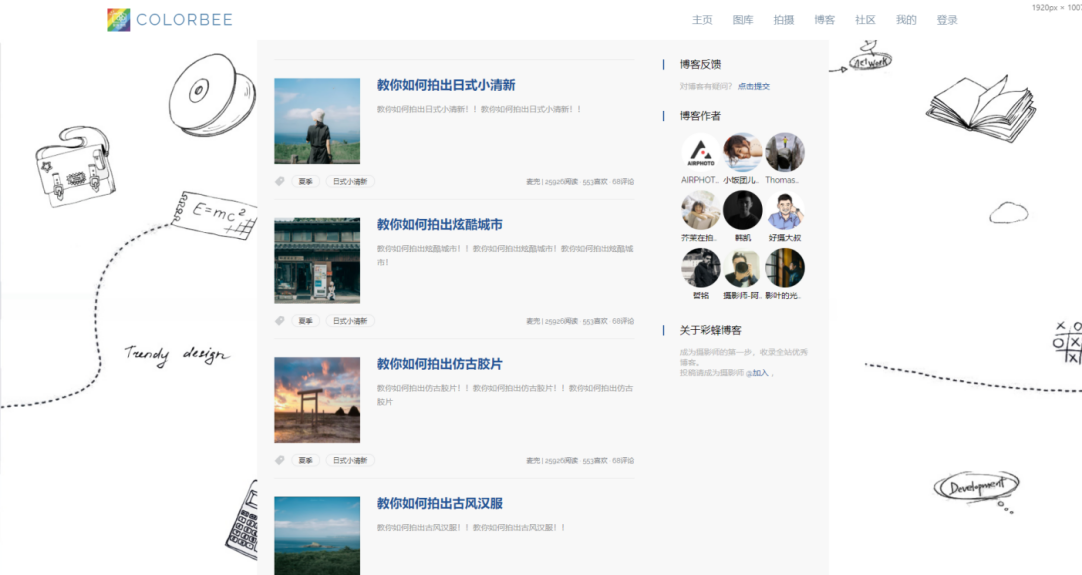


图29 博客列表

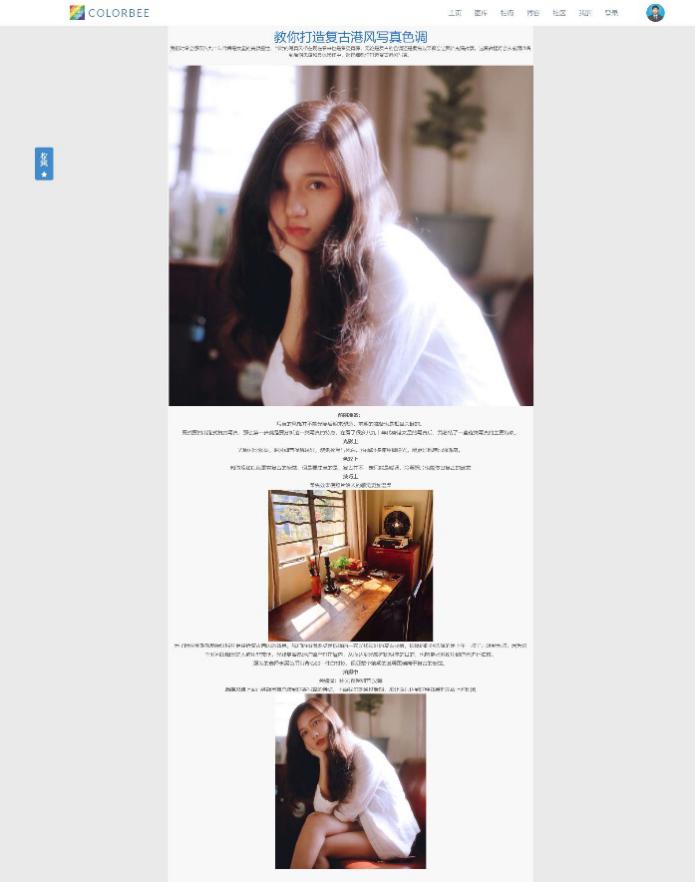


图30 博客详情

用户进入博客模块，前端请求所有博客的列表进行展示，用户可以看到博客列表，有标签等基本信息，点击任意博客进入可以查看博客的详情信息，点击收藏可以收藏博客。

### 5.3.5 社交圈子模块

社交圈子模块为用户提供类似微博一样的功能，可以发布图文动态。

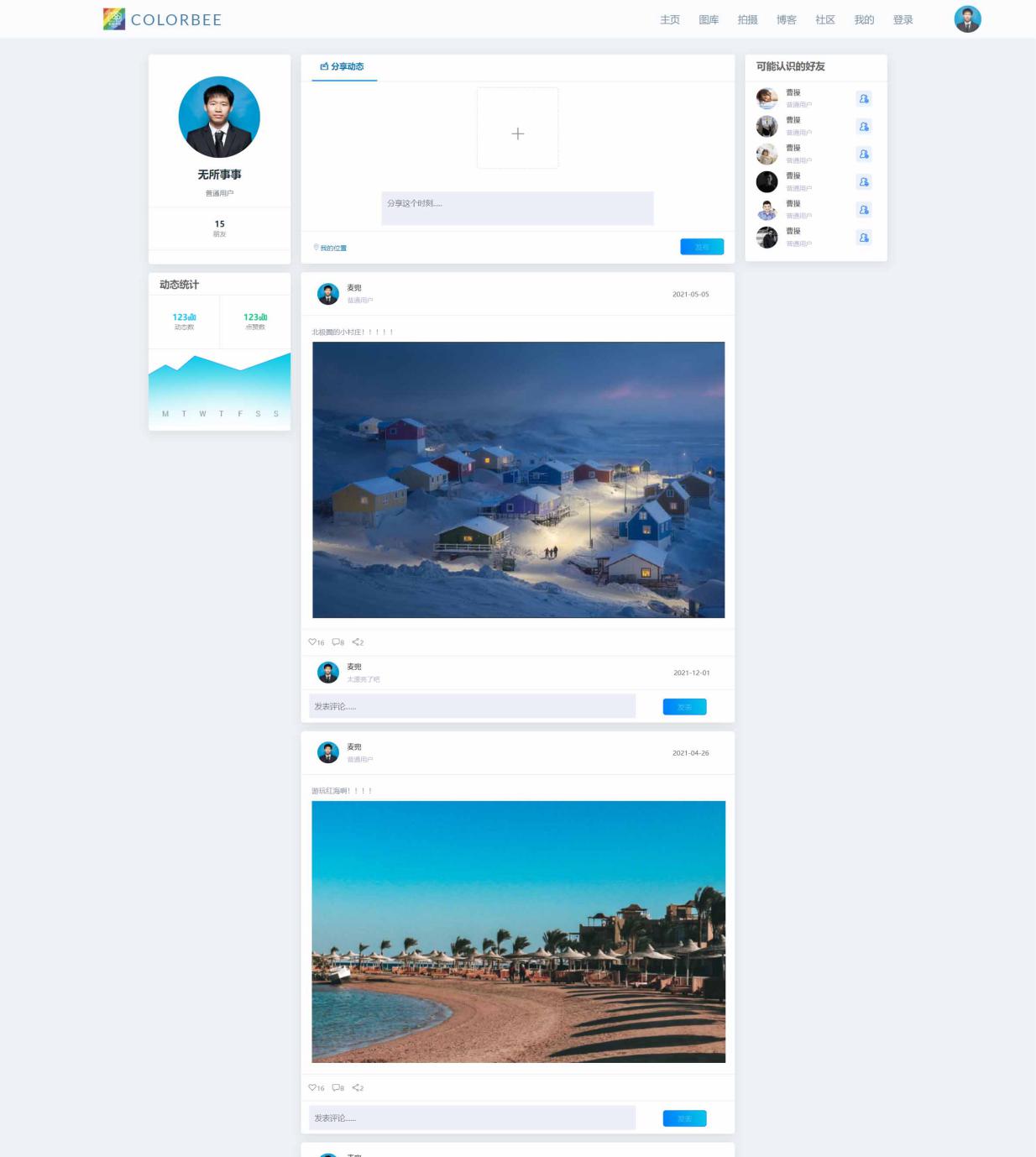


图31 社交圈子页面

用户进入社交圈子模块，前端请求所有圈子的动态，用户可以看到所有动态信息，用户可以上传图片编写文本发布动态，用户也可以在其他人的动态下面进行留言评论。

### 5.3.6 个人信息模块

个人信息模块里用户可以查看自己的个人信息，以及对个人信息进行修改，同时在右上角头像处还可以执行退出登录等操作。

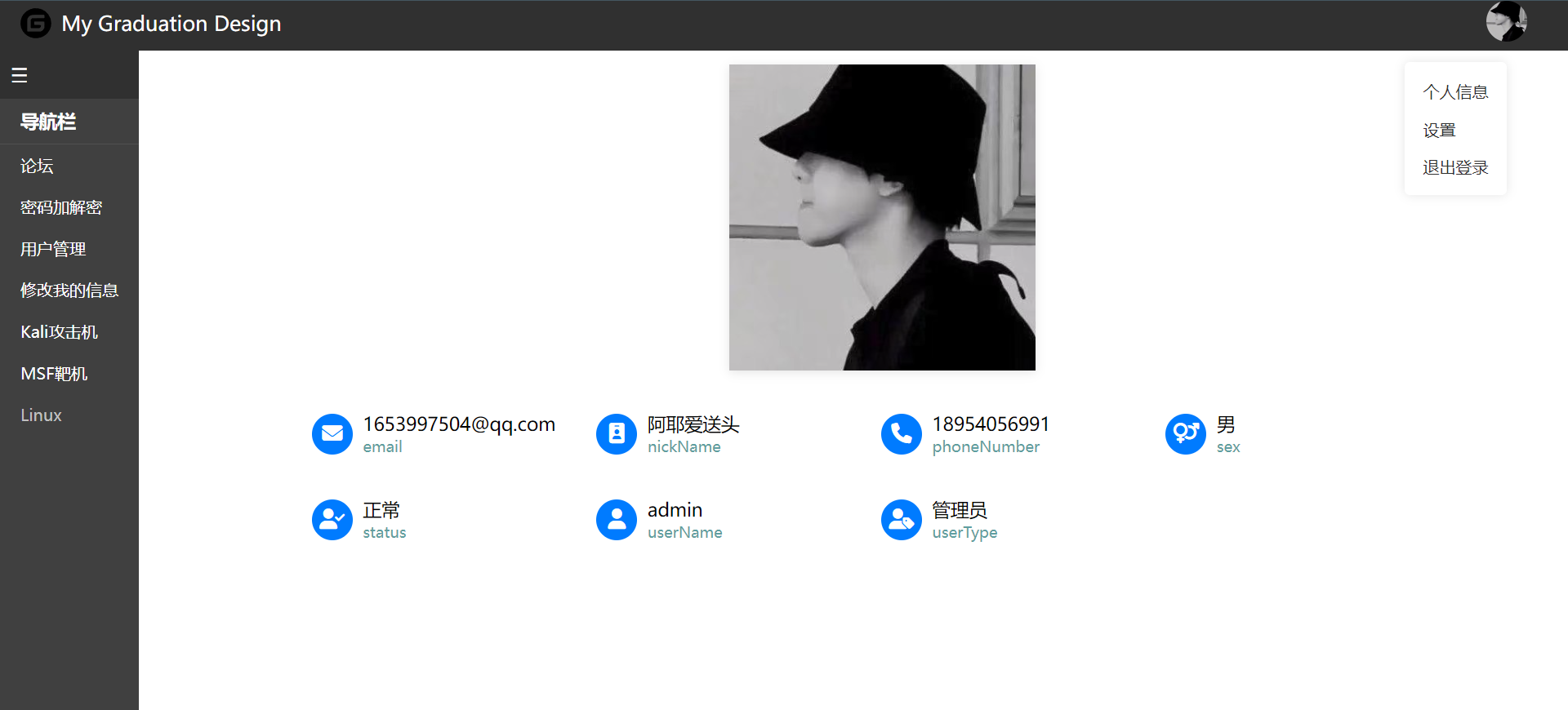


图32 预约摄影信息



图33 发布博客

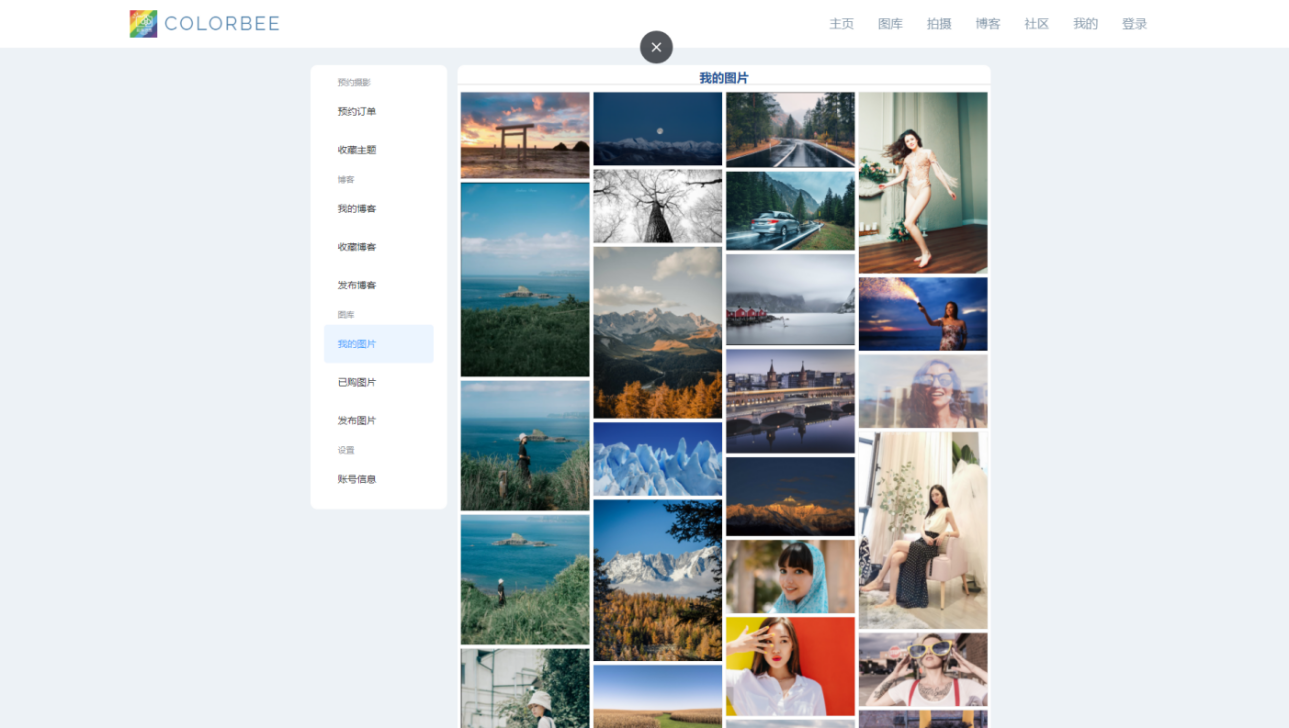


图34 我的图片

点击已购买图片或者我的图片可以查看图片大图，同时可以点击下载和删除

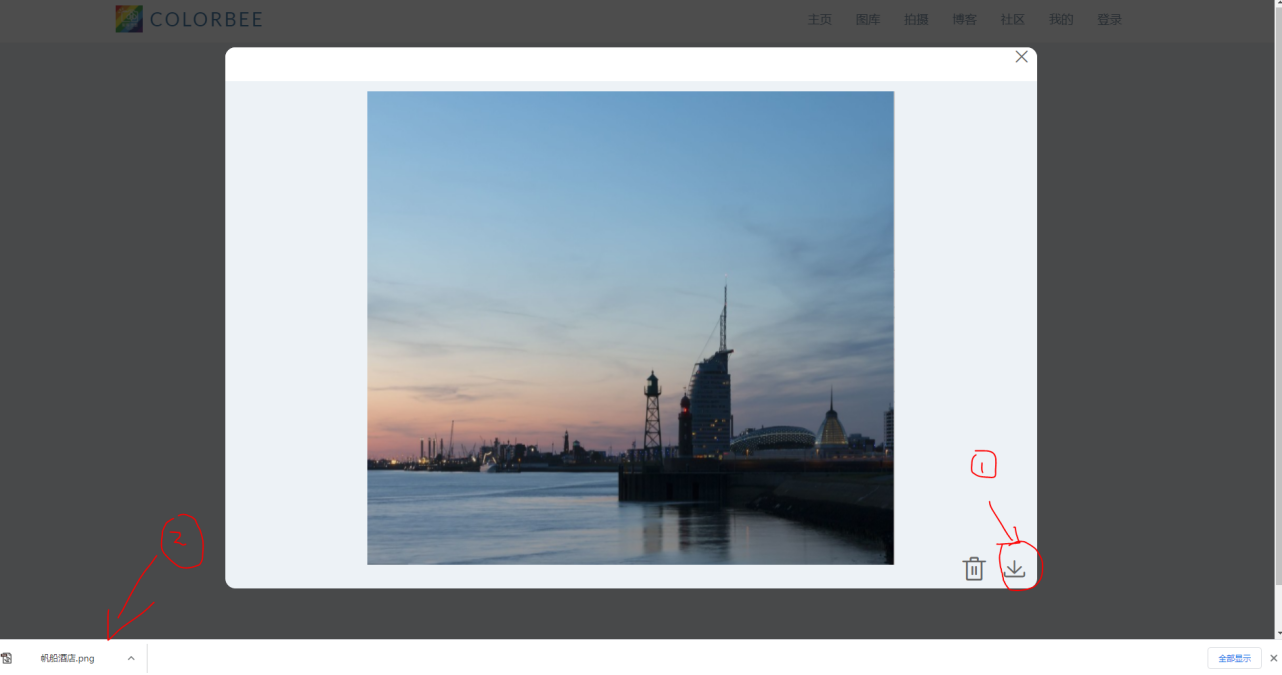


图35 下载图片

### 5.3.7 登陆注册模块

用户登陆是本系统的唯一标识，用户注册是登陆的前置功能。

用户进入登陆模块输入账号和密码进行登陆，若登陆验证成功，后端返回token并自动跳转首页。若用户尚未注册可以进入注册页面填写账号信息进行注册。

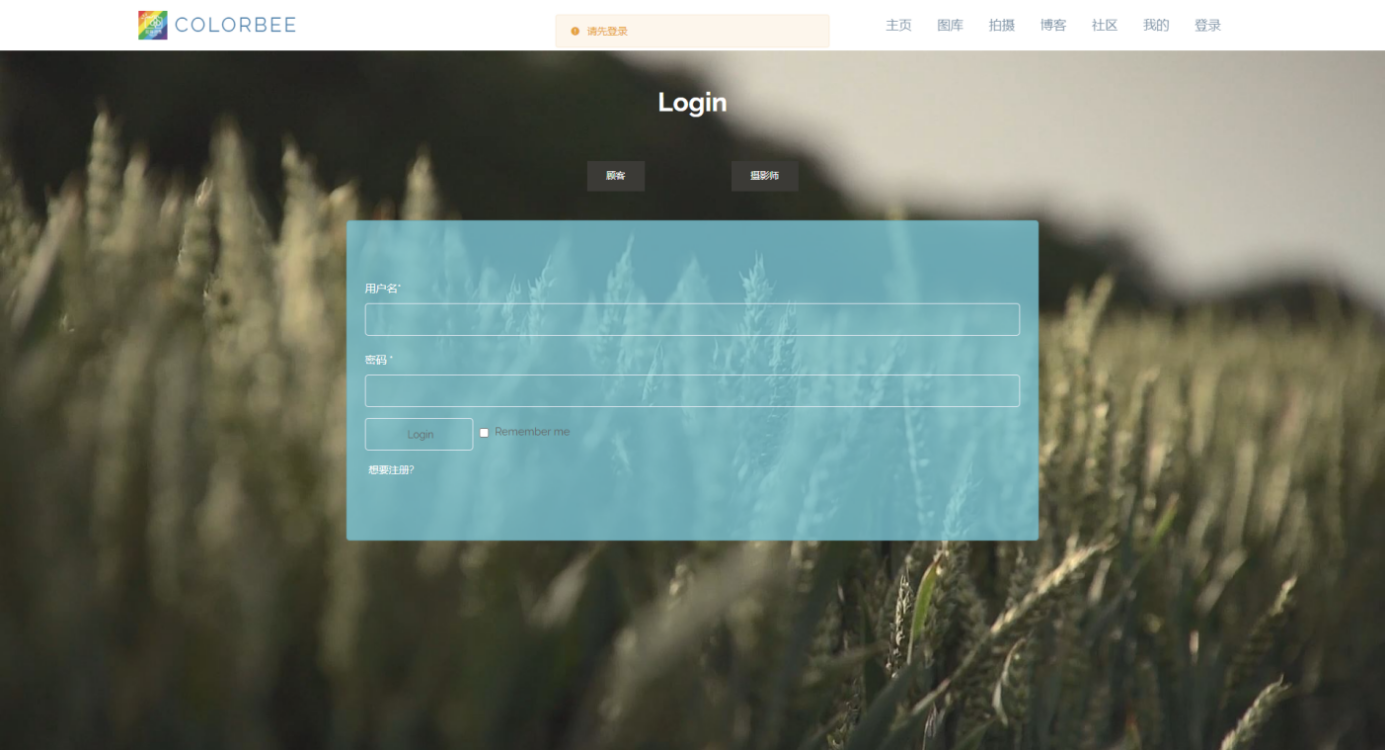


图36 登陆页面

# 6 系统测试

## 6.1 测试需求

本次测试针对开发的摄影图库综合系统进行测试，通过总结用户的使用习惯、注意用户在使用系统中容易出现的人为错误操作来进行测试用例的设置。

## 6.2 测试目标

在测试过程中发现系统的漏洞和不足并予以更正，提高系统的安全性和稳定性。

## 6.3 测试用例

表30 用户登陆测试

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 编号 | 测试项目 | 输入 | 期待输出 | 实际输出 | |
| 1 | 用户登陆  系统 | 1、输入正确的用户名和密码 | 1、登陆成功返回首页 | | 1、登陆成功返回首页 |
| 2、输入错误的账号 | 2、提示账号不存在 | | 2、提示账号不存在 |
| 3、输入正确的账号和错误的密码 | 3、提示密码错误 | | 3、提示密码错误 |
| 4、不输入账号和密码 | 4、提示请输入账号密码 | | 4、提示请输入账号密码 |

表31 用户访问受限页面测试

| 编号 | 测试项目 | 输入 | 期待输出 | 实际输出 | |
| --- | --- | --- | --- | --- | --- |
| 2 | 用户访问受限页面 | 1、用户已登陆访问主题详情页面 | 1、访问成功 | | 1、访问成功 |
| 2、用户未登陆访问主题详情页面 | 2、提示需登录自动跳转登陆页面 | | 2、提示需登录自动跳转登陆页面 |
| 3、用户已登陆访问图片详情页面 | 3、访问成功 | | 3、访问成功 |
| 4、用户未登陆访问体图片详情页面 | 4、提示需登录自动跳转登陆页面 | | 4、提示需登录自动跳转登陆页面 |
|  |  | 5、用户已登陆访问博客详情页面 | 5、访问成功 | | 5、访问成功 |
|  |  | 6、用户未登陆访问博客详情页面 | 6、提示需登录自动跳转登陆页面 | | 6、提示需登录自动跳转登陆页面 |
|  |  | 7、用户已登陆访问圈子页面 | 7、访问成功 | | 7、访问成功 |
|  |  | 8、用户未登陆访问圈子页面 | 8、提示需登录自动跳转登陆页面 | | 8、提示需登录自动跳转登陆页面 |
|  |  | 9、用户已登陆访问我的模块 | 9、访问成功 | | 9、访问成功 |
|  |  | 10、用户未登陆访问我的模块 | 8、提示需登录自动跳转登陆页面 | | 8、提示需登录自动跳转登陆页面 |

表32 管理员内容管理测试

| 编号 | 测试项目 | 输入 | 期待输出 | 实际输出 | |
| --- | --- | --- | --- | --- | --- |
| 3 | 管理员内容管理测试 | 1、登陆正确的管理员账号访问内容管理页面 | 1、访问成功，查看内容 | | 1、访问成功，查看内容 |
| 2、未登录访问内容管理页面 | 2、提示登陆跳转登陆页面 | | 2、提示登陆跳转登陆页面 |
| 3、输入完整正确主题信息进行添加主题 | 3、提示添加成功返回主题列表 | | 3、提示添加成功返回主题列表 |
| 4、输入不完整主题信息进行添加主题 | 4、提示请输入完整信息 | | 4、提示请输入完整信息 |
|  |  | 5、输入完整正确图片信息进行添加图片 | 5、提示添加成功返回图片列表 | | 5、提示添加成功返回图片列表 |
|  |  | 6、输入不完整图片信息进行添加图片 | 6、提示请输入完整信息 | | 6、提示请输入完整信息 |
|  |  | 7、输入完整正确博客信息进行添加博客 | 7、提示添加成功返回博客列表 | | 7、提示添加成功返回博客列表 |
|  |  | 8、输入不完整博客信息进行添加图片 | 8、提示请输入完整信息 | | 8、提示请输入完整信息 |

## 6.4 测试总结

通过对系统部分主要功能模块设计了测试用例执行黑河测试，并未发现有明显的业务逻辑错误漏洞，系统利用Vue中的interceptors拦截器对所有的错误信息进行拦截，并使用Vue的message进行弹框提示，具有友好的容错和交互体验。

# 7 部署上线

本项目已经部署上线，使用webpack使用webpack中的loader对设计中所用到的ES6语法代码、Sass等进行转换，使用plugin插件uglifyjs-webpack-plugin、html-webpack-plugin对js、html代码进行压缩，并分别对前台项目和后台管理项目进行模块化打包，服务端中配置静态资源托管，让服务端中可以根据不同地址分别访问前台网站和后台管理网站。租用阿里云服务器，安装node、mongo等相关依赖，使用VsCode中的ssh-remote连接服务器上传服务端代码，使用Nginx进行反向代理，解决跨域问题，使用PM2进程管理工具开启持久化Node服务，使服务端后台保持运行且无需每次手动开启，申请域名与IP地址设置映射关系，使用户可以通过域名访问。使用Certbot,申请SSL证书为网站添加安全认证，Nginx中进行配置修改，使用UFW开启防火墙443端口，使网站通过HTTPS协议进行访问。网站地址为<https://www.chakfung-ng.com/>，用户名：chakfung，密码：134679。

# 8 总结与展望

由于本人兴趣爱好是摄影，并且在大学期间担任过摄影协会的会长和作为业务摄影师去工作过，因此比较了解摄影师、摄影爱好者以及摄影需求者他们的所存在的难题和需求。常说兴趣与专业结合是最好的状态，本人很早就已经萌生了设计一个能够解决这一类群体的综合性平台，打造一个专注于与摄影相关的平台。本平台能够提供图库销售、提供摄影预约，提供摄影学习的博客等服务，能够真正有效的解决特点人群的需求。每一个存在且存活下来的产品必然是因为它能满足一部分群众的需求，目前国内做专注于摄影的平台不多，另外进行类似滴滴打车这种形式的网上预约摄影更是少见，为了方便用户系统还设计了小程序端让用户可以在微信小程序上进行使用，因此本人认为本系统是存在开发的意义。  
 本次毕业设计，从系统选题、市场调研、需求分析、系统概要设计、系统详细设计、系统测试到最后的论文撰写均为本人独立完成。实现了整个系统从0到1的过程，毕业设计是对大学四年所学到的知识的一次综合性应用，通过毕业设计不仅仅是完成任务，更是对知识的自检查。开发的过程中遇到了许多的问题，最终都通过查阅资料等方法一一解决了，期间收获了许多。整体开发流程采用了大三时候所学的软件工程知识，遵循着软件工程的开发思想，使得整体开发更加流畅，更加符合工程化开发。技术上采用了目前较为流行且新的技术，VueJS、NodeJS、Express框架、mongoDB数据库、ECharts统计图、WebSocket通讯，本人的职业方向选择了前端开发工程师，这些技术都是前端开发工程师必备的技能。通过毕业设计，对于知识是一次全方位的巩固，更加收获了许多的知识。收获包括以下但不仅以下几点：

1. 系统使用目前前端方向最流行的几门技术进行实现，Vue、Node.js、Express框架，期间遇到了许许多多的问题，通过查阅官方文档论坛、找资料、请教同学老师一个一个突破，提高了解决问题和实际开发的能力，增加了知识的广度和深度。
2. 系统采用了HTML5的WebSocket技术，利用这个单TCP连接全双工通讯协议，顺利实现了多用户在线通讯，应用在了客服咨询的功能开发上，更加贴合实际开发应用。
3. 本系统专注摄影相关的设计和理念都是本人的想法，目前市面上没有相似产品可以参考，这大大加大开发的难度，但是同时锻炼了创新性思考的能力。
4. 巩固了大学期间所学的课程知识，真正的把书本上的理论应用到实战当中去，例如从开发的初期到末期都贯穿着软件工程的思想，从需求分析、可行性分析、概要设计、E-R图、用例图设计等都用到了软件工程。其中大学中所学的Web项目开发课程也得以运用，包括计算机网络等。
5. 学会利用ECharts进行系统可视化数据展示，对系统的数据进行统计展示。
6. 把系统进行部署上线中一系列操作实施方法。

同时本摄影图库综合系统是一个庞大且具有研究价值的系统，由于目前的能力以及时间限制，无法全部实现，但是系统存在很多可以改善和拓展研究的地方：

1. 由于实际系统是面向用户的，所以性能的优化十分重要，可以在很多方面减少性能的消耗，提高网页的响应速度。
2. 后续可以添加摄影师验证的功能，让用户提交申请资料给系统进行审核通过成为注册摄影师。

未来的职业道路还很漫长，坚持学习是进步的根本。在计算机领域这片浩瀚的星辰中还有许多等待我去了解学习的知识，在以后的工作和学习中，一步一个脚印不断前行。

# 参 考 文 献

1. 子矜(摘). 中国图片产业未来前景的可能性[J]. 美术观察, 2009(3):120-120.
2. 崔丽莎. 中国网络图片库市场竞争现状分析[J]. 竞争政策研究, 2018, 000(003):P.71-79.
3. 刘树逸, 张平娟. 基于微信小程序的摄影分享软件构建与应用[J]. 福建电脑, 2019(3).
4. 刘玉佳. 微信"小程序"开发的系统实现及前景分析[J]. 信息通信, 2017.
5. 黑马程序员. 微信小程序开发实战[M]. 北京: 人民邮电出版社, 2019.
6. 朴灵. 深入浅出Node.js[M]. 北京: 人民邮电出版社, 2013.
7. 王金龙, 宋斌, 丁锐. Node.js:一种新的Web应用构建技术[J]. 现代电子技术, 2015.
8. 黄丹华. NODE.JS开发实战详解(WEB开发典藏大系)[M]. 北京: 清华大学出版社, 2014.
9. Ethan Brown. Node与Express开发[M]. 北京: 人民邮电出版社, 2015.
10. 纪尧姆.Vue项目实战[M]. 北京: 人民邮电出版社, 2019.
11. Callum Macrae. Vue.js快跑[M]. 北京: 电子工业出版社, 2018.

致 谢

时光如白驹过隙，初次踏入仲园的那一天仿如昨日，四年时间在一生当中说长不长，说短不短，大学的时光也许会是我们人生最快乐时光之一。大学的最后一个环节毕业设计也就此完结了，在这里首先要感谢指导老师梁瑜，从一开始的选题、系统的初步设计、系统设计中期、到最后的论文撰写，当中的每一环都离不开梁瑜老师的悉心指导。在此首先向梁瑜老师表示真挚的感谢。

同时非常感谢仲恺农业工程学院，提供了一个这么好的平台让我去学习成长，感谢所有曾经教导过我的老师，从对计算机这个领域一无所知到顺利地踏入这个行业离不开老师们的无私奉献和辛勤栽培，你们所教导我的点点滴滴，都将让我受益终身。另外还要感谢我的同学们，是你们让我拥有了一个积极向上而不乏味的学习氛围，让我度过了一段开心且充实的校园生活，

各奔东西之际，祝愿老师们身体健康、事事顺利，同学们前程似锦！