

石河子大学

本科毕业论文(设计)



不同水体识别系统设计与实现

学 生 姓 名 :	梁嘉成
学 号 :	20201108005
学 院 :	信息科学与技术学院
专 业 :	计算机科学与技术 (第二学士学位)
年 级 :	2020 级
指 导 教 师 :	尹小君

中国·新疆·石河子

2022 年 6 月

摘 要

图像识别技术最早被提出在上世纪四十年代，但自提出以来直到九十年代，人们将神经网络和支持向量机相结合，才进一步促进了图像识别技术的发展，这使图像识别技术有了更加广泛的应用空间。计算机水体识别技术，是利用计算机对图像进行处理、分析和理解，以识别各种不同模式的水体的技术，是应用深度学习算法实现图像识别的一种实践应用。

本设计旨在实现一个前后端分离的可二次开发的系统，通过用户信息处理、用户仓库处理、数据采集、不同水体提取、和算法评价等模块，实现不同水体识别系统。

通过选用合理的模型，可以完成水体的模式识别和评价工作，通过与模型低耦合的系统设计，使识别系统可以适配多种算法模型，进而实现不同水体识别系统。实现初期，选用实现简单的深度学习模型，完成系统的功能性测试，而后可根据用户反馈不断迭代。系统设计采用 Web 应用 B/S 结构实现，简化系统开发、维护和使用；前端选用 Vue 框架实现前后端分离的组件化编程，后端采用 Java 语言开发，使用 Spring Boot 框架，整合 MyBatis、JUnit 等其他开源框架，完成系统功能的实现。

关键词：水体识别、深度学习、Vue、Spring Boot

Abstract

Image recognition technology was first proposed in the 1940s, but since it was put forward until the 1990s, the combination of neural network and support vector machine has further promoted the development of image recognition technology, which makes the image recognition technology has more extensive application space. Computer water body recognition technology is a technology that uses computer to process, analyze and understand images to identify various water bodies with different patterns. It is a practical application of deep learning algorithm to realize image recognition.

This design aims at realizing a system which can be developed again by separating the front and back ends. Through the modules of user information processing, user warehouse processing, data collection, extraction of different water bodies, and algorithm evaluation, different water bodies identification system can be realized.

By selecting a reasonable model, the water body pattern recognition and evaluation can be completed. Through the system design with low coupling with the model, the recognition system can adapt to a variety of algorithm models, and then realize different water body recognition system. At the beginning of implementation, a simple deep learning model was selected to complete the functional test of the system, and then the system could be iterated continuously according to user feedback. The system design adopts Web application B/S structure to simplify the development, maintenance and use of the system. The front end uses Vue framework to realize the componentization programming of the front and back end separation. The back end uses Java language development, uses Spring Boot framework, integrates MyBatis, Junit and other open source frameworks to complete the realization of system functions.

Keywords: water body recognition, deep learning, Vue, Spring Boot

目 录

1. 绪论.....	1
1.1. 研究意义.....	1
1.1.1. 国外研究现状	1
1.1.2. 国内研究现状	3
1.2. 设计思路.....	4
1.2.1. 水体识别系统用户界面	4
1.2.2. 水体识别系统服务器端	4
1.2.3. 水体识别系统持久层端	5
2. 相关技术原理.....	6
2.1. 图像识别技术	6
2.1.1. 技术分析	6
2.1.2. OpenCV	6
2.1.3. PyTorch.....	6
2.1.4. YOLOv5.....	7
2.2. 前端技术.....	7
2.2.1. Vue 全家桶	7
2.2.2. Axios.....	8
2.3. 后端技术.....	8
2.3.1. Java EE.....	8
2.3.2. Spring 全家桶	9
2.3.3. 项目管理工具、服务器及其它框架.....	9
2.3.4. 数据库	10
3. 需求分析.....	11
3.1. 可行性分析.....	11
3.1.1. 技术可行性	11
3.1.2. 经济可行性	11
3.1.3. 操作可行性	11
3.1.4. 时间可行性	11
3.2. 系统功能性需求	12
3.2.1. 数据库需求	12
3.2.2. 前端交互	12
3.2.3. 后端业务逻辑	12
3.2.4. 图像识别处理	12
3.3. 非功能性需求	12
3.3.1. 仓库系统	12
3.3.2. 收费系统	12
4. 系统总体设计.....	13
4.1. 系统总体设计原则	13
4.2. 系统逻辑结构设计	13
4.3. 整体结构设计	14
4.4. 数据库设计.....	15
4.4.1. 概念设计阶段	15

4.4.2. 逻辑结构设计阶段	16
4.4.3. 数据库物理设计阶段	18
4.4.4. 数据库实施阶段	18
4.4.5. 数据库运行和维护阶段	18
4.5. 缓存设计	18
5. 系统实现	19
5.1. 系统框架	19
5.1.1. 前端服务	19
5.1.2. 后端服务	19
5.2. 开发环境	19
5.2.1. 开发环境	19
5.2.2. 部署环境	20
5.3. 前端实现	20
5.3.1. 登录逻辑	20
5.3.2. 登录成功主页	21
5.3.3. 关于我们	22
5.3.4. 项目业务	23
5.3.5. 新建仓库	25
5.3.6. 查看仓库	26
5.3.7. 修改密码	27
5.3.8. 联系我们	28
5.4. 后端实现	29
5.4.1. 前端控制器逻辑	29
5.4.2. 登录逻辑	29
5.4.3. 增、删、改、查逻辑	29
5.4.4. 图像处理逻辑	29
6. 系统测试	31
6.1. 接口测试	31
6.2. 单元测试	31
7. 总结与展望	32
7.1. 总结	32
7.2. 展望	32
8. 主要参考文献	34
9. 致谢	36

1. 绪论

1.1. 研究意义

传统水体识别方法中，模型结构层次较浅，计算量较小，计算时间较短，不需要以大量数据集为基础，即可完成对图像的分析处理，但此方法很难从原图像中获取高层次的语义和深度特征，由于认为以及外界因素的干扰，图像识别率一般较低，在面对大量数据分析下，无人设计，很难获取图像信息的特征。采用深度学习识别图像时，往往能够获得更深层次的图像特征，图像特征表达将更为丰富，图像特征提取更加准确，图像识别正确率相较传统方法更高。

本设计选用目前研究水体识别方法前沿领域中的深度学习模型，通过不同水体提取系统实现数据采集、特征提取、不同水体提取、和算法评价等模块，对不同算法进行全方面评价，并应用于研究区域的不同水体提取。

通过选用合理的模型，可以完成水体的模式识别和评价工作，通过与模型低耦合的系统设计，使识别系统可以适配多种算法模型，进而实现不同水体识别系统。实现初期，选用实现简单的深度学习模型，完成系统的功能性测试，而后可根据用户反馈不断迭代。

1.1.1. 国外研究现状

图像识别技术，是利用计算机对图像进行处理、分析和理解，以识别各种不同模式的目标和对象的技术，是应用深度学习算法的一种实践应用。研究该技术对推动人工智能的发展具有重要的意义。

图像识别技术最早被提出在上世纪四十年代，但自提出以来直到九十年代，人们将神经网络和支持向量机相结合，才进一步促进了图像识别技术的发展，这使图像识别技术有了更加广泛的应用空间。随着时间的推移和技术的发展，原本需要耗费大量人力对图像做预处理的工作，也逐渐由科学家们在研究中提出的深度学习模型取代，例如 DBN、DFN、CNN、RNN 等。

深度学习是图像识别领域的一种重要技术手段，其目的是通过构建一个多层网络，在这个网络上的计算机通过自动学习以得到数据隐含的内部关系，从而得到更加隐含和深刻的以数据形式表现的内容，因此深度学习在未来图像识别研究和应用中仍然是一个重要课题。

水体，水的集合体。水体是江、河、湖、海、地下水、冰川等的总称。水体识别系统的核心是通过深度学习的计算机图像识别技术对不同水体提取的相关算法和模型进行研究。目前该领域已经吸引了众多相关学者数十年的研究投入，存在众多不同的水体识别系统和算法，但大多都是针对某一特定水体或是某一特定功能的系统。

深度学习在图像识别中的模型有以下几类，如表 1-1。

表 1-1 经典网络结构对比分析

方法	主要任务	优点	缺点
DBN	特征提取	能够反映同类数据本身的相似性	对于分类问题，分类精度不高 某些学习的复杂性较高 输入数据具有平移不变性
CNN	特征提取	训练参数减少，模型的泛化能力更强 池化运算降低网络的空间维度，对输入数据的平移不变性要求不高	容易出现梯度消散问题 空间关系辨识度差 物体大幅度旋转之后识别能力低下
RNN	特征提取	可以对序列内容建模	需要训练的参数较多，容易出现梯度消散或梯度爆炸问题 不具备特征学习能力
GAN	生成对抗样本	与其他模型相比，能够产生更好的样本 可以训练任何一种生成器网络 不需要设计遵循任何种类的因式分解的模型，任何生成器网络 and 任何鉴别器都会有用 避免了复杂的马尔科夫过程，回避了近似计算棘手的概率难题	模型的收敛性较差 训练过程中，容易出现崩溃问题 由于无需预先建模，模型过于自由不可控
CapsNet	特征提取	解决了CNN模型出现的空间关系辨识度差以及物体大幅度旋转之后识别能力低下	模型的网络结构较浅，在图像识别分类上的正确率和目前流行的CNN模型还是有很大差距

采用图像识别技术识别遥感图像是当前主流的水体识别分析方法，遥感图像作为良好、可靠、稳定的数据源，为图像识别技术提供了广泛的学习模型。遥感图像分类的主要依据是地物的波谱特征。地物波谱特征是指该地物对太阳辐射的反射，散射能力随波长而变的规律，地物波谱特征与地物的组成成份，物体内部的结构关系密切。一般说来，不同地物拥有不同的地物波谱特征，据此可以将它们识别。

随着遥感技术被广泛应用于水体监测领域，水体信息提取方法成为热门研究

方向。如 Komeil 等利用 Landsat TM、ETM+和 OLI 遥感影像,模拟了 2000~2013 年伊朗乌鲁米耶湖的时空变化; Adrian 等以澳大利亚东部的 TM/ETM/OLI 影像为数据源,在比较七种水体指数的基础上,提出了一种简单精确的大范围水体自动分类方法。目前,用于水体信息提取的方法以单波段阈值法和多波段谱间关系法为主。单波段阈值法主要是利用水体与背景地物在遥感影像的某一波段反射率存在差异,能有效抑制背景地物,实现与背景地物相分离的目的,但单波段阈值法对不同相、不同区域的水体需要设置不同的阈值,具有一定的局限性;多波段谱间关系法综合利用各波段信息,通过波段之间的组合,极大的增强了水体与其它地物反射率的差异,与单波段阈值法相比不受时空的影响,多波段谱间关系法中以水体指数法最为常见,如 McFeeters 提出了归一化差异水体指数(normalized difference water index, NDWI),能够抑制植被和土壤信息,实现增强水体信息的作用。

1.1.2. 国内研究现状

徐涵秋针对 NDWI 提取市区水体不理想,在 NDWI 的基础上,提出了改进的归一化水体指数(modified normalized difference water index,MNDWI),陈文倩等基于高分一号 4 个波段并结合决策树法,提出了阴影水体指数(shade water index, SWI),能有效的剔除阴影、裸地等背景地物对水体的影响,王瑾杰等在阴影水体指数的基础上,提出了改进的阴影水体指数(modified shade water index,MSWI),进一步提高阴影与水体的分离程度,王小标等针对复杂环境下水体提取精度易受到低反射率地表影响的问题,利用 ETM+影像,构建了多波段水体指数(multi-band water index, MBWI),但此方法很难提出与水体反射率接近的地物,王琳等利用 Landsat8 影像,提出了双红外水体指数(doubleinfrared band waterindex, DIBWI),用于准确识别蓝藻湖泊水体信息。

现阶段水体研究主要利用国外卫星数据,国外卫星数据存在时间/空间分辨率低、晴空数据少等问题,难以对水体进行精准监测,而我国的高分六号卫星,是我国自主研发的低轨光学遥感卫星,与国外数据相比,具有高时间分辨率、宽覆盖等特点,有利于湖泊水库的精细化动态监测。

1.2. 设计思路

本系统总体设计采用 Web 应用 B/S¹结构实现,简化系统开发、维护和使用。

系统中需要设计实现的模块功能如下图 1-1。

系统设计遵循 MVC 模式²,其优点有职责单一、互不影响,利于各功能模块解耦合、利于组件重用和二次开发。

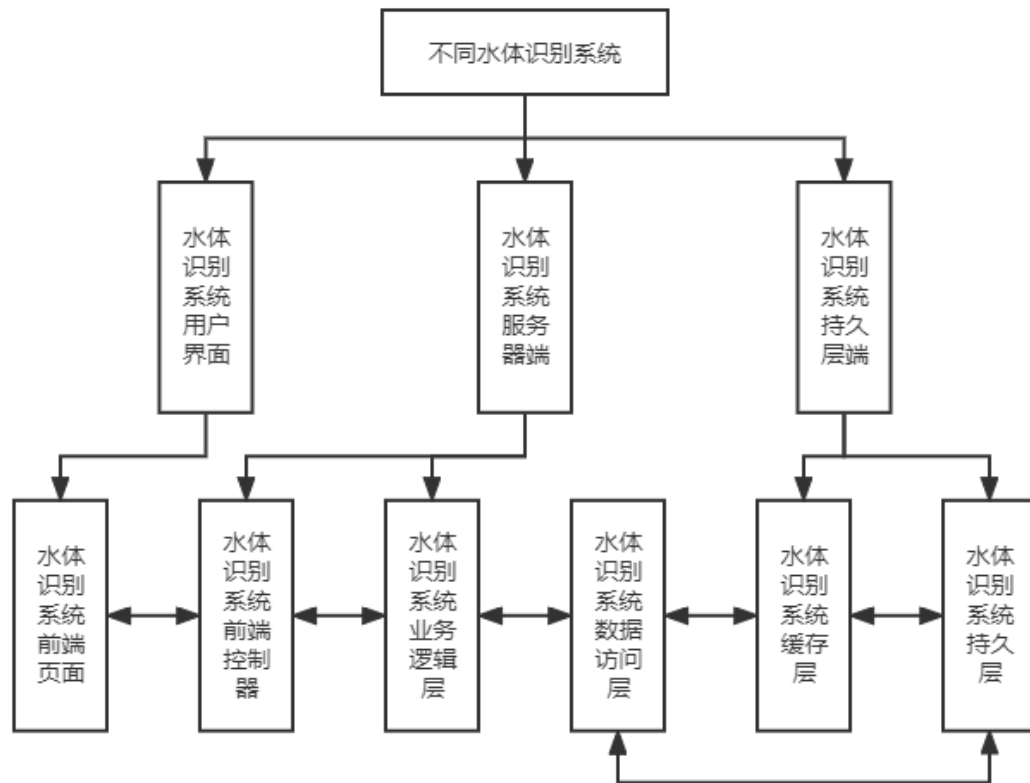


图 1-1 不同水体识别系统设计思路图

1.2.1. 水体识别系统用户界面

该模块的主要任务是实现与用户交互,完成用户的登录、注册、上传图像等交互功能。该模块设计的主要目的是实现水体识别系统前端页面的展示和构建、前端路由跳转的设计与实现、前端向后端发送请求和处理响应数据并展示的逻辑实现。

1.2.2. 水体识别系统服务器端

该模块的主要任务是用于构建业务逻辑体系。

¹ Browser/Server, 浏览器/服务器模式

² 一种分层开发的模式; M: Model 业务模型, 处理业务; V: View 视图, 界面展示; C 控制器, 处理请求, 调用模型和视图

（1）前端控制器

前端控制器主要完成与前端的信息交互，实现处理前端请求并相应数据的功能。该模块设计的主要目的是处理前端请求数据，向下交付给业务逻辑层进行处理，并以响应形式返回业务逻辑层向上交付的数据给前端。

（2）业务逻辑层

该层是实现业务逻辑的核心层，主要包含两部分，设计的目的就是实现两部分功能。

一是实现核心功能，核心功能模块是水体识别的处理逻辑，这里是主要运用深度学习模型实现水体识别的环境。

二是实现用户功能，用户功能主要包括识别用户登录逻辑、识别用户权限逻辑等功能。

（3）数据访问层

该层是用于数据库和缓存交互，实现数据的增、删、改、查基本操作，并向上为业务逻辑层提供服务，该层的主要任务是编写业务逻辑层需要请求数据库交互所需的全部接口。

1.2.3. 水体识别系统持久层端

（1）缓存层

该层主要功能是处理一些经常被访问和操作的数据，以达到降低磁盘 I/O 操作导致的 I/O 阻塞平均时长，以加快程序运行速度。该层的主要任务是实现数据访问层提供的接口。

（2）持久层

该层是将数据进行持久化操作的一层，通过合理的技术选型和建库建表等操作可以大大降低占用的外存存储空间。该层的主要任务是实现数据访问层提供的接口。

2. 相关技术原理

2.1. 图像识别技术

2.1.1. 技术分析

深度学习模型存在各自优缺点，下表 1-1 阐述了不同深度学习模型各自存在的优缺点。

由于卷积神经网络训练参数少，模型的泛化能力更强，池化运算降低网络的空间维度，对输入数据的平移不变性要求不高，根据水体信息综合以上信息考虑，为解决遥感影像分辨率高、信息量大而导致信息识别提取精度不高的问题，本项目选用卷积神经网络进行水体识别系统的设计。

水体识别系统依赖的数据源可以选用北斗卫星导航系统。北斗卫星导航系统是中国着眼于国家安全和经济社会发展需要，自主建设、独立运行的卫星导航系统，是为全球用户提供全天候、全天时、高精度的定位、导航和授时服务的国家重要空间基础设施。

系统实现选择前后端分离的开发模式，前后端分离模式可以避免在后端项目中添加前端页面，导致逻辑结构不清、维护和二次开发困难，有利于实现高并发、高性能、高可用，后端编写维护接口文档，在 API 变化时更新接口文档，后端根据接口文档进行接口开发，前端根据接口文档进行开发，开发完成后联调和提交测试。

2.1.2. OpenCV

OpenCV 是一个基于 Apache2.0 许可（开源）发行的跨平台计算机视觉和机器学习软件库。它轻量级而且高效——由一系列 C 函数和少量 C++ 类构成，同时提供了 Python、Ruby、MATLAB 等语言的接口，实现了图像处理和计算机视觉方面的很多通用算法。

应用此框架可以完成对上传图片的先行处理。

2.1.3. PyTorch

利用 PyTorch 框架可以训练一个用于图像处理的模型，采用此模型对图片进行处理。

2.1.4. YOLOv5

由于图像识别采用微服务实现，这里的神经网络预计采用流行的 YOLOv5 实现。YOLOv5 是一系列在 COCO 数据集上预先训练的对象检测架构和模型，是最著名的对象检测算法之一，发布于 2020 年 5 月 18 日，作者是 Glenn Jocher。

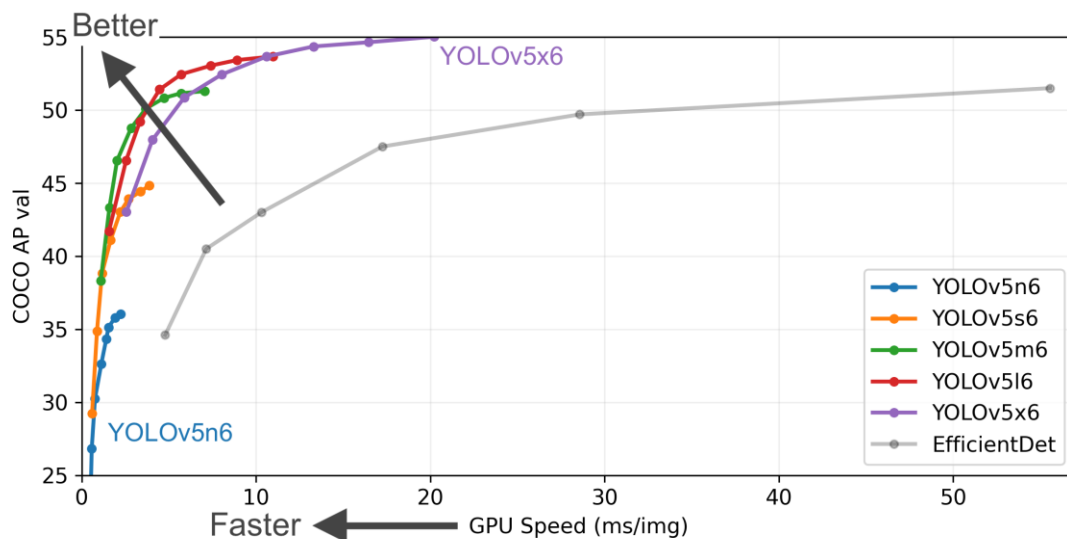


图 2-1 YOLOv5 性能图

2.2. 前端技术

2.2.1. Vue 全家桶

(1) Vue-CLI

前端选用 Vue.js 框架实现，Vue 是一套用于构建用户界面的渐进式 JavaScript 框架。与其它大型框架不同的是，Vue 被设计为可以自底向上逐层应用。Vue 的核心库只关注视图层，不仅易于上手，还便于与第三方库或既有项目整合。另一方面，当与现代化的工具链以及各种支持类库结合使用时，Vue 也完全能够为复杂的单页应用提供驱动。

Vue 的开发模式还借鉴了 MVVM¹模型使得程序员可以将注意力完全集中在数据上，用数据驱动视图，而数据和视图间的维护工作则交给 Vue 进行。借助强大的 Webpack²打包工具实现的 Vue-CLI，我们可以实现组件化模式的编程，以大大降低代码复用和模块耦合关系。

¹ Model-View-ViewModel 它本质上就是 MVC 的改进版，是将其中的 View 的状态和行为抽象化，让我们将视图 UI 和业务逻辑分开

² webpack 是一个用于现代 JavaScript 应用程序的静态模块打包工具

尽管截至 2022 年尤雨溪团队已经宣布 Vue3 成为 Vue 的默认版本，但是为了保证本项目的稳定性，本项目仍然采用 Vue2 实现前端应用。

(2) Vue-Router

采用 Vue 实现单页面应用时，需要进行的前端路由跳转可借助 Vue-Router 插件实现，这简化了路由跳转的开发难度，增加了视图的多样性，不仅可以给用户良好的体验，同时也实现了路由跳转逻辑和前端编码工作的解耦合，另外也可以借助路由守卫、JWT¹ 技术实现用户登录状态的认证，避免了采用 Session 认证的诸多弊端。

(3) Vuex

为了方便不同组件之间的消息和数据共享，可以借助 Vuex 插件实现。Vuex 可以帮助我们管理共享状态，并附带了更多的概念和框架。这是对短期和长期效益进行权衡后的选择。

2.2.2. Axios

由于 Vue 只注重视图的展示，且是单页面应用，通信时主要采用 AJAX，我们就需要一个 AJAX 框架以实现前后端的通信工作。这里我们选用 Axios 框架，Axios 是一个基于 promise 的 HTTP 库，可以用在浏览器和 node.js² 中。

2.3. 后端技术

2.3.1. Java EE

后端选用 Java 作为开发语言，其优势有如下四点。

(1) Java 是全场景开发语言。采用 Java 开发一个比较方便的事情是技术选型可以完成从前端、移动端到后端的整体解决方案，这是目前不少开发团队选择 Java 语言的一个重要原因。

(2) Java 语言的生态体系比较健全。Java 语言经过多年的发展，已经形成了一个健全的语言生态体系，这会在很大程度上降低程序开发的潜在风险，也能够很大程度上保证项目的开发周期。随着 Java 语言在云计算和大数据领域的广泛应用，未来 Java 语言在生产环境下的应用依然有广阔的前景。

¹ Java Web Token 是为了在网络应用环境间传递声明而执行的一种基于 JSON 的开放标准（RFC 7519）。该 token 被设计为紧凑且安全的，特别适用于分布式站点的单点登录（SSO）场景

² 一个基于 Chrome V8 引擎的 JavaScript 运行环境，使用了一个事件驱动、非阻塞式 I/O 模型，让 JavaScript 运行在服务端的开发平台

(3) 稳定的性能和较强的扩展性。Java 语言虽然在代码实现上相对繁琐一些（与 Python 相比），但是 Java 语言的性能却比较稳定，而且 Java 语言的扩展性也比较强，这些因素决定了大型互联网平台往往更愿意采用 Java 语言。

(4) 开发人员基数大。目前 IT 行业内掌握 Java 编程语言的程序员非常多，这使得搭建 Java 开发团队非常方便，这也是不少开发团队愿意采用 Java 语言的一个重要原因。

Java EE 是指 Java Enterprise Edition，Java 企业版，用于企业级开发，是 Java 的一套开发规范，包括 Web 开发等。目的是帮助程序员开发和部署可移植、健壮、可伸缩且安全的服务器端 Java 应用程序。

2.3.2. Spring 全家桶

(1) Spring、Spring MVC

而选用 Java 就不得不提到 Spring 框架，Spring 是一款目前主流的 Java EE 轻量级开源框架，是 Java 世界最为成功的框架之一。

(2) Spring Boot

Spring Boot 是所有基于 Spring 开发的项目的起点。Spring Boot 的设计是为了尽可能快的跑起来 Spring 应用程序并且尽可能减少配置文件。简单来说就是 Spring Boot 其实不是什么新的框架，它默认配置了很多框架的使用方式。

2.3.3. 项目管理工具、服务器及其它框架

(1) Maven

Maven 是一个项目管理工具，可以通过一小段描述信息来管理项目的构建，报告和文档的项目管理工具软件；它包含了一个项目对象模型，一组标准集合，一个项目生命周期，一个依赖管理系统，和用来运行定义在生命周期阶段中插件目标的逻辑。

本项目采用 Maven 作为项目管理工具。

(2) Tomcat

Tomcat 服务器是一个免费的开放源代码的 Web 应用服务器，属于轻量级应用服务器，在中小型系统和并发访问用户不是很多的场合下被普遍使用，是开发和调试 JSP 程序的首选。

(3) MyBatis

MyBatis 是一款持久层框架，它支持自定义 SQL、存储过程以及高级映射。MyBatis 免除了几乎所有的 JDBC 代码以及设置参数和获取结果集的工作。MyBatis 可以通过简单的 XML 或注解来配置和映射原始类型、接口和 Java POJO¹为数据库中的记录。

(4) Git

Git 是一个分布式版本控制系统，旨在快速高效地处理从小型到超大型项目的所有内容。

2.3.4. 数据库

(1) Redis

Remote Dictionary Server，即远程字典服务，是一个开源的使用 ANSI C 语言编写、支持网络、可基于内存亦可持久化的日志型、Key-Value 数据库，并提供多种语言的 API。

(2) MySQL

MySQL 是最流行的关系型数据库管理系统，在 WEB 应用方面 MySQL 是最好的 RDBMS²应用软件之一。

¹ Plain Old Java Objects，普通老式 Java 对象

² Relational Database Management System 关系数据库管理系统

3. 需求分析

3.1. 可行性分析

3.1.1. 技术可行性

图像识别技术诞生 80 年来，已经形成了非常成熟的理论和技术基础，相关文献记载的关于水体识别技术的应用也不胜枚举，本系统图像识别算法借助相关文献和开源平台资料，选用现阶段比较成熟的开源算法作为水体识别的基础，以现阶段发展成熟的 Java Web 技术作为后端开发，选用 Spring MVC 框架，以易于上手的 Vue 作为前端框架，数据存储选用开源、高效且成熟的 MySQL 库保证了系统技术上的可行性。

3.1.2. 经济可行性

本项目在研发阶段开发算法和环境均来自相关文献资料和开源社区，开发时间预计为半年以内，相关开发软件中有 IntelliJ IDEA、WebStorm 需要支付每年约 ¥2600 费用，网络方面 10Mbps 带宽网络需要每年向电信运营商支付 ¥1000 费用，服务器上，可以购买云服务器，年费用在 ¥100 左右，经济上可行。项目投入运营阶段需要申请域名等相关服务和资源，运营维护阶段可根据访问量和需求进行系统迭代和变更，但核心技术框架一般情况下无需迭代升级，经济上完全可行。

3.1.3. 操作可行性

本系统实现需要学习运用的知识面贯穿前端、后端、数据库和算法，是软件工程的具体实践，是对本科阶段所学知识的一个综合应用，操作上可以参照软件工程规范进行，可行性较高。

3.1.4. 时间可行性

通过制定充分的开发计划，结合项目管理技术的应用，项目在时间周期安排上可行。

第 1-2 周：收集阅读文献、完成开题报告和文献综述；

第 3-5 周：需求分析，确立课题设计总体规划，和各个模块的方案设计；

第 6-10 周：完成软件设计以及系统的详细设计；

第 11-12 周：系统运行及测试；

第 13-14 周：撰写论文、用户手册等文档。

3.2. 系统功能性需求

3.2.1. 数据库需求

数据库系统用以存储必要的数据库，本系统中最重要的是存储图片在本地的 URL 和图片状态等必要信息。由于系统需要识别用户和用户状态等信息，所以数据库中也要存储用户信息。

3.2.2. 前端交互

前端用于和用户直观交互，以实现一个亲和用户的平台，前端交互中，需要用户使用浏览器进行登录访问和使用服务器资源，前端交互需要实现的功能主要包含用户登录、用户资料、用户充值、用户仓库、用户上传、用户下载等核心功能。

3.2.3. 后端业务逻辑

后端用来处理登录请求、图片识别请求等逻辑业务。主要是为前端提供 Restful 风格接口，以保证前端发送的请求能够被正确的识别和反馈，后端的处理逻辑直接相关的便是如何保证用户的登录状态、如何存储和修改用户的资料、如何实现用户充值和用户仓库系统，如何保证用户的上传功能，如何实现图片的识别设计。

3.2.4. 图像识别处理

用来处理分析图像，并标记特征并返回给后端业务逻辑，进行处理。图像识别是该系统的核心业务功能，如何处理图片是本系统考虑的重要问题之一。

3.3. 非功能性需求

3.3.1. 仓库系统

用来存储用户图片的仓库。仓库系统是使使用户获得良好交感体验的功能，是非必要系统，但仓库系统可以异步、批量的处理用户的识别请求，能够使用户获得更好的体验。

3.3.2. 收费系统

实现按图片处理数量收费。收费系统是系统上线到生产环境中后需要盈利的手段，在测试环境中是非必要系统。

4. 系统总体设计

4.1. 系统总体设计原则

抽象：描述物体本质的设计技术即为抽象，它忽略了不重要的层面。抽象将一些复杂事情简化到可以分析和理解的程度。在软件工程学科中，从定义软件到开发系统经历了许多步骤。每个步骤都可以看作是软件解决方案抽象层次的细化。抽象的底层是软件的源代码。在模块化设计中，也可以有多个抽象层次。抽象级别最高的模块一般用概括的办法描述问题的解决方案，抽象级别较低的模块详细描述了抽象较高的问题的解决方案。

模块化：程序中的模块总是程序对象的集合，例如数据描述和可执行语句，或者是单独定名和寻址的元素。模块化是将一个要开发的软件分成几个简单的小部件模块，每个模块可以独立开发、测试，最后拼装成一个完整的程序。使程序结构清楚，易于阅读、理解、测试和更改，则是模块化的主要目的。

封装：封装是程序结构开发中应用的规则。每个功能的组件都封装在一个模块中，在定义每个模块时，尽可能少地暴露内部处理。封装在提升软件的易修改性、易测试性和易移植性方面起着至关重要的作用。

可扩展性原则：系统设计还应考虑到未来可能会有新的需求，如扩展新功能、接入新系统、增加新设备等，因此系统应保留第三方接口，以方便其他系统的接入和并行使用。

系统实现选择前后端分离的开发模式，前后端分离模式可以避免在后端项目中添加前端页面，导致逻辑结构不清、维护和二次开发困难，有利于实现高并发、高性能、高可用，后端编写维护接口文档，在 API 变化时更新接口文档，后端根据接口文档进行接口开发，前端根据接口文档进行开发，开发完成后联调和提交测试。

4.2. 系统逻辑结构设计

不同水体识别系统的逻辑结构设计对该系统有着至关重要的作用，本系统公司信息展示模块、用户信息处理模块、图片仓库模块、图片处理模块如下所示。

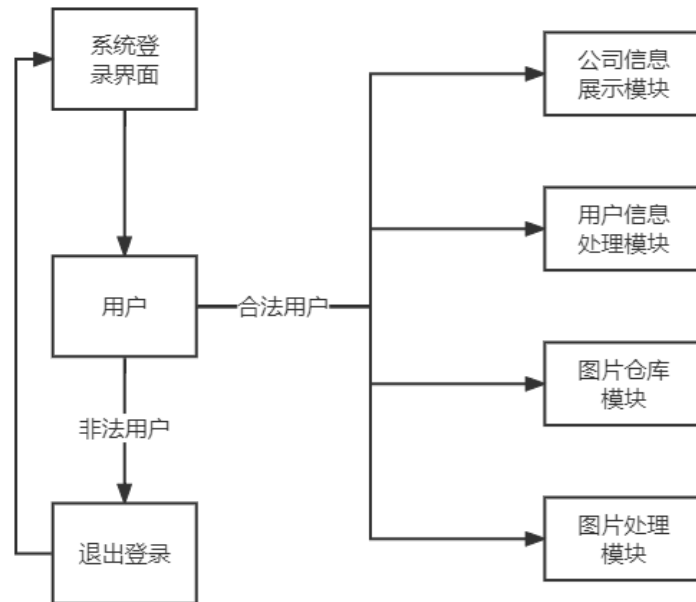


图 4-1 逻辑结构设计

4.3. 整体结构设计

系统设计采用模块化结构。各模块的设计应遵循高内聚、低耦合的思想，以改善各模块间的不一致性，尽可能减少模块间的数据耦合，最大限度地降低模块间的数据依赖性。系统总体结构设计如下图所示。

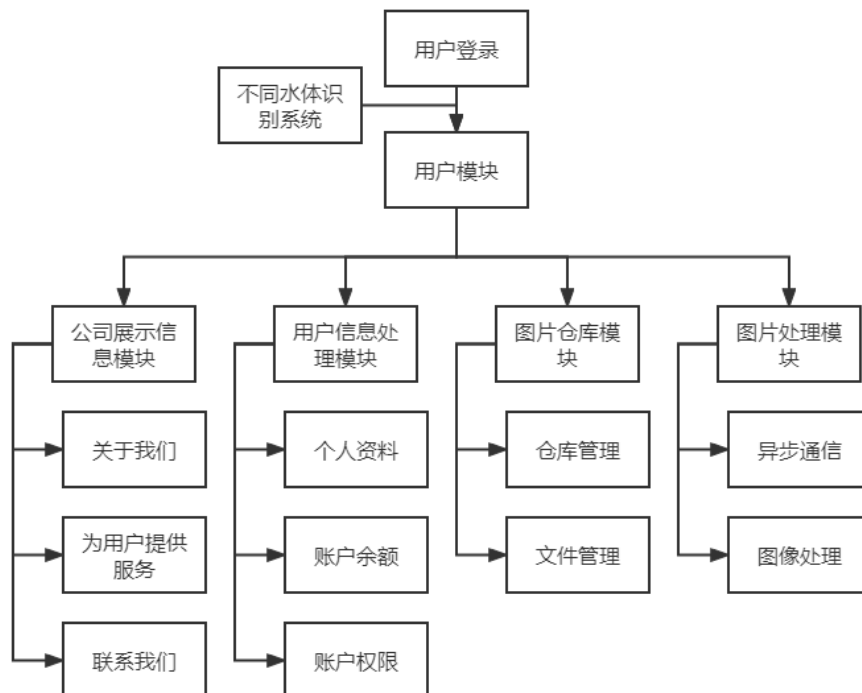


图 4-2 系统结构图

4.4. 数据库设计

4.4.1. 概念设计阶段

根据需求分析，本设计预计需要如下的三张表和属性，E-R 图如下。

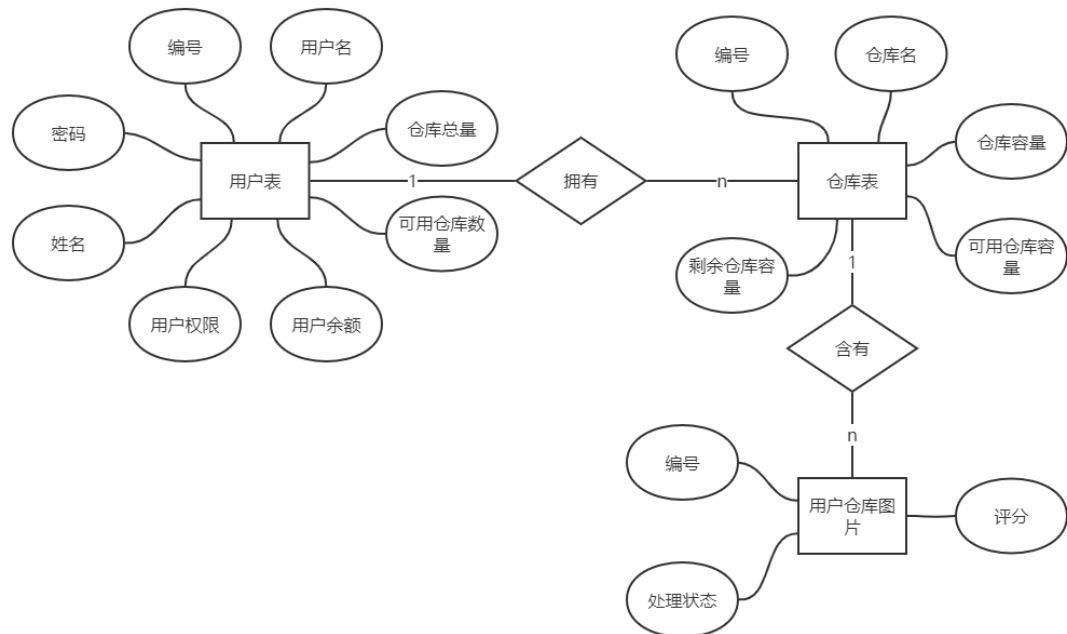


图 4-3 概念设计 E-R 图

(1) 用户表

编号：唯一编号作为该表主键；

用户名：登录的用户名；

姓名：用户姓名；

密码：用户登录密码；

用户权限：用户的使用权限；

用户余额：用户的使用余额；

仓库总量：用户仓库总量；

可用仓库数量：用户可用仓库数量。

(2) 仓库表

编号：唯一编号作为该表主键；

仓库名：仓库名称；

仓库容量：仓库总容量；

可用仓库容量：可用仓库容量；

剩余仓库容量：剩余仓库容量。

(3) 用户仓库图片

编号：唯一编号作为该表主键；

评分：该图片的评分；

处理状态：该图片的处理状态。

4.4.2. 逻辑结构设计阶段

(1) 表结构设计

表 4-1 用户表结构设计

属性名	描述	数据类型	约束
ID	编号	INT	主键约束
ACCOUNT	用户名	VARCHAR(50)	唯一约束、非空约束
NAME	姓名	VARCHAR(20)	非空约束
PASSWORD	密码	VARCHAR(20)	非空约束
PERMISSION	用户权限	ENUM('1','2','3')	非空约束
BALANCE	用户余额	FLOAT	非空约束、默认约束值为 0
AVAILABLE	仓库总量	INT	非空约束、默认约束值为 10
CAPACITY	可用仓库数量	INT	非空约束、默认约束值为 10

表 4-2 仓库表结构设计

属性名	描述	数据类型	约束
ID	编号	INT	主键约束
NAME	仓库名	VARCHAR(50)	非空约束
COUNT	仓库容量	INT	非空约束、默认约束值为 10
AVAILABLE	可用仓库容量	INT	非空约束、默认约束值为 0
REMAINING	剩余仓库容量	INT	非空约束、默认约束值为 10

表 4-3 用户仓库图片表结构设计

属性名	描述	数据类型	约束
ID	编号	VARCHAR(128)	主键约束
STATE	处理状态	ENUM('处理中','处理成功','处理失败')	非空约束
SCORE	评分	INT	非空约束

(2) 表结构设计

创建数据库

create database user2;

创建用户

create user 'user2'@'localhost' identified by '123456';

```
# 授予用户权限
grant all on user2.* to 'user2'@'localhost';
# 使用该数据库
use user2;
# 建立表
# 1. 用户表
# 用户编号
# 用户邮箱（登录名）
# 用户密码
# 用户姓名
# 用户权限
# 用户余额
# 可建仓库数量
# 仓库总量
create table userInfo(
    id int auto_increment primary key,
    account varchar(50) unique not null ,
    password varchar(20) not null ,
    name varchar(20) ,
    permission enum('1','2','3') not null ,
    balance float default 0 not null ,
    available int default 10 not null,
    capacity int default 10 not null
);
drop table userInfo;
select * from userInfo;
delete from userInfo where true;
insert into userInfo values (null,'443808626@qq.com','123456','梁嘉成','1',100,10,10);
update userInfo set balance=100 where account='443808626@qq.com';
# 2.用户仓库表
# 仓库编号
# 仓库名称
# 仓库容量
# 已用容量
# 剩余容量
# 用户编号
create table userWarehouse(
    id int auto_increment primary key,
    name varchar(50) not null ,
    count int not null default 10,
    available int not null default 0,
    remaining int not null default 10,
    userInfo_id int not null ,
    foreign key (userInfo_id) references userInfo(id)
```

```
);  
select * from userWarehouse;  
delete from userWarehouse where true;  
drop table userWarehouse;  
# 3.仓库  
# 图片编号  
# 仓库编号  
# 图片原始地址  
# 识别图片地址  
create table warehouse(  
    id varchar(128) primary key,  
    userWarehouse_id int not null ,  
    state enum('处理中','处理成功','处理失败') default '处理中' not null ,  
    score int default 0,  
    foreign key (userWarehouse_id) references userWarehouse(id)  
);  
select * from warehouse;  
drop table warehouse;
```

4.4.3. 数据库物理设计阶段

数据库物理设计采用 MySQL 关系型数据库，数据引擎采用 INODB，原因是 INODB 支持事务处理，本设计中需要用到事务。

数据库的存储结构根据存储引擎决定，不需要设计存储过程，索引只需要主键默认的聚簇索引，采用 B+树。

数据库存取路径可视生产环境决定放在默认路径下。

系统配置选用 CentOS。

4.4.4. 数据库实施阶段

数据库实施阶段，采用 JDBC 技术、和 Junit4 对数据库进行必要测试。

4.4.5. 数据库运行和维护阶段

根据需要，进行必要的维护。

4.5. 缓存设计

缓存采用 Redis 数据库，主要缓存内容是登录的 UUID 和验证码，项目上线后如遇流量高峰可加入对特定数据的缓存，缓存设计要求如有增删时，要先更新数据库，后删除缓存，最大程度保证数据一致性。

5. 系统实现

5.1. 系统框架

5.1.1. 前端服务

由于前端部分采用浏览器，实现，固本设计选用 Vue.js 框架及其全家桶系列构建前端系统，利用 AJAX 与后端实现通信。

5.1.2. 后端服务

后端采用开源的 Spring 框架实现，服务器采用 Spring Boot 集成的 Tomcat 服务器实现，具体见图 5-1。

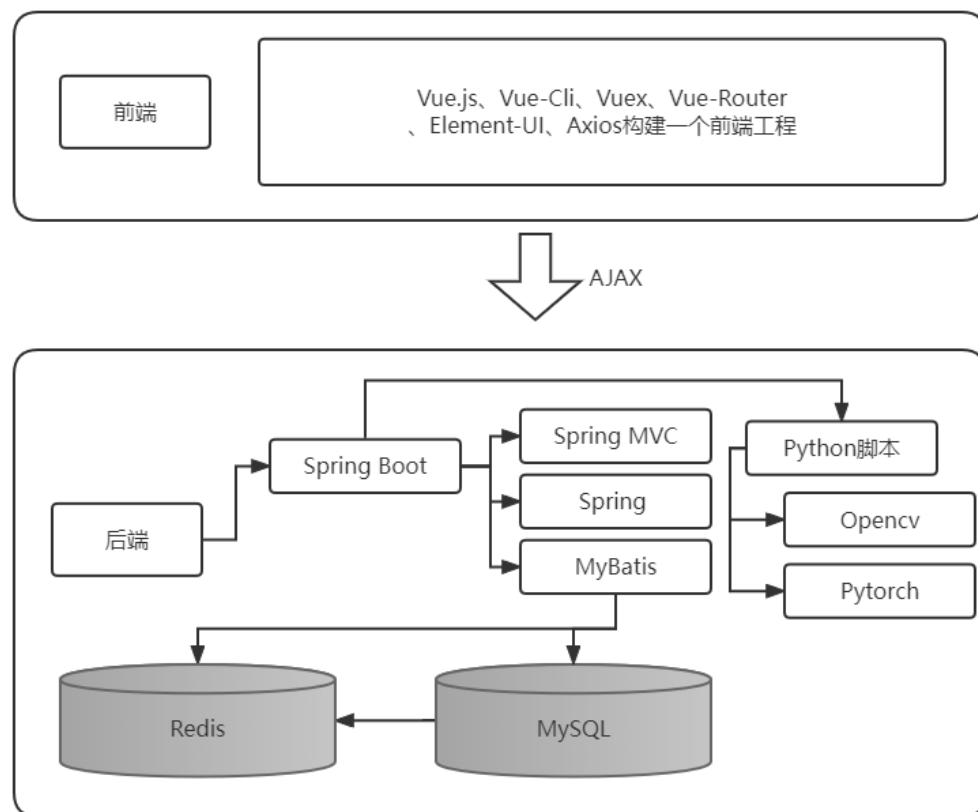


图 5-1 项目整体架构图

5.2. 开发环境

5.2.1. 开发环境

前端：WebStorm、Vue-Cli；

后端：IntelliJ Idea、Spring Boot。

5.2.2. 部署环境

可将项目部署在云服务器上，选用云服务器规格如下。

硬盘：高 I/O，40GB；

内存：2GB；

操作系统镜像：CentOS 8.2 64bit。

5.3. 前端实现

5.3.1. 登录逻辑

用户只有通过登录模块的检验后，才可访问不同水体识别系统。可以说，登录模块是整个大学生综合测评系统的门户。用户先访问登录页 login，运行效果如图 5-2、图 5-3、图 5-4 所示。在用户名栏输入系统分配的用户名，在密码栏输入与用户名匹配的密码，设定要访问的模块，然后点击登录按钮就可以进入管理页面；如果用户名、密码填写错误，系统会转到错误信息页，给出相应的错误提示。

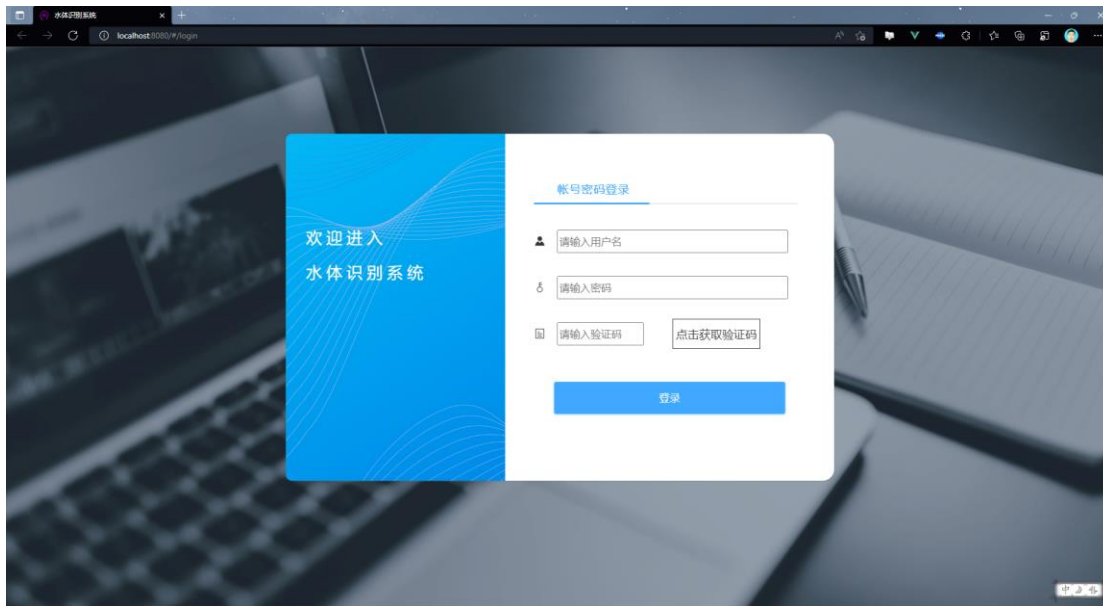


图 5-2 登录界面

图 5-3 所示为输入错误的时候所展示的用户界面。

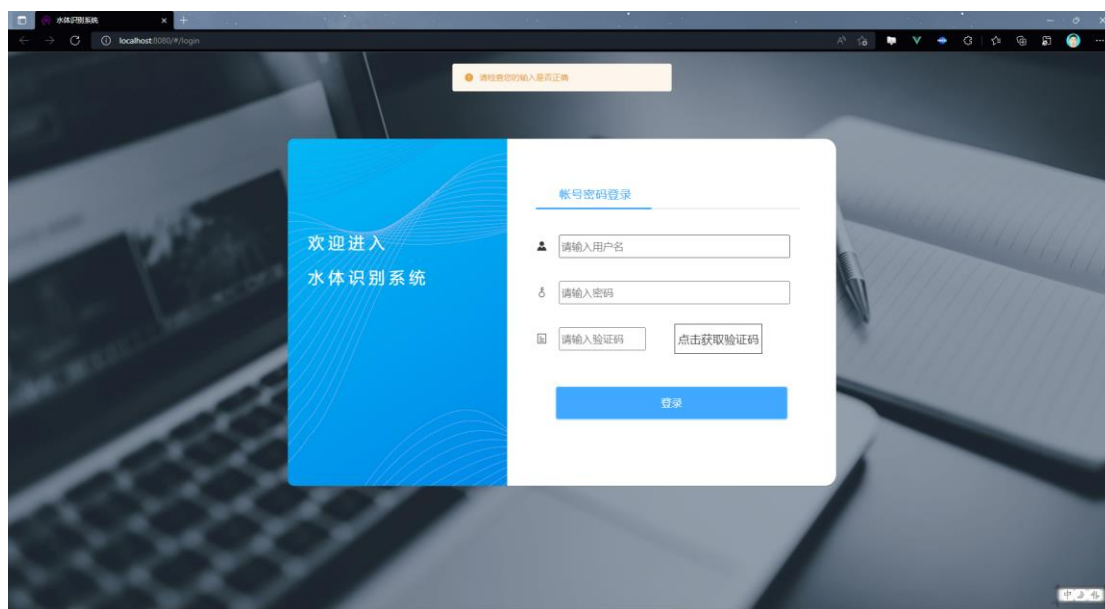


图 5-3 提醒输入提示界面

图 5-4 所示为输入错误的时候所展示的用户界面。



图 5-4 输入错误提示界面

5.3.2. 登录成功主页

登录成功界面展示为首页，上方导航栏与下方导航栏内容相同，为固定组件，内容包含首页、关于我们、项目业务、联系我们、注销用户，下侧导航栏右侧有联系信息。

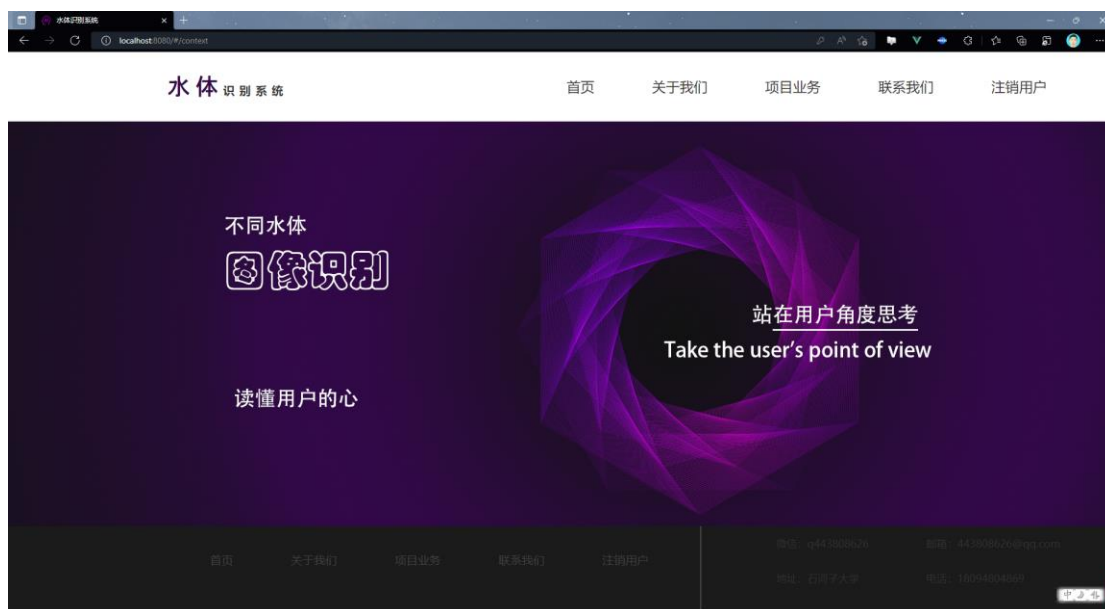


图 5-5 登录成功界面

5.3.3. 关于我们

此栏主要介绍的是该系统的介绍内容。

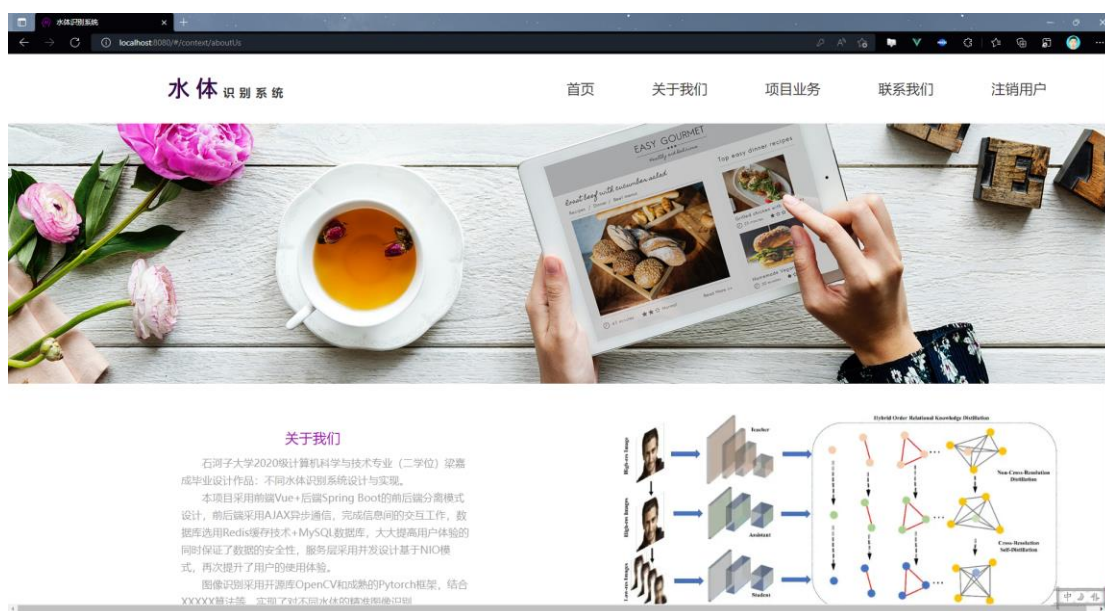


图 5-6 关于我们 1



图 5-7 关于我们 2

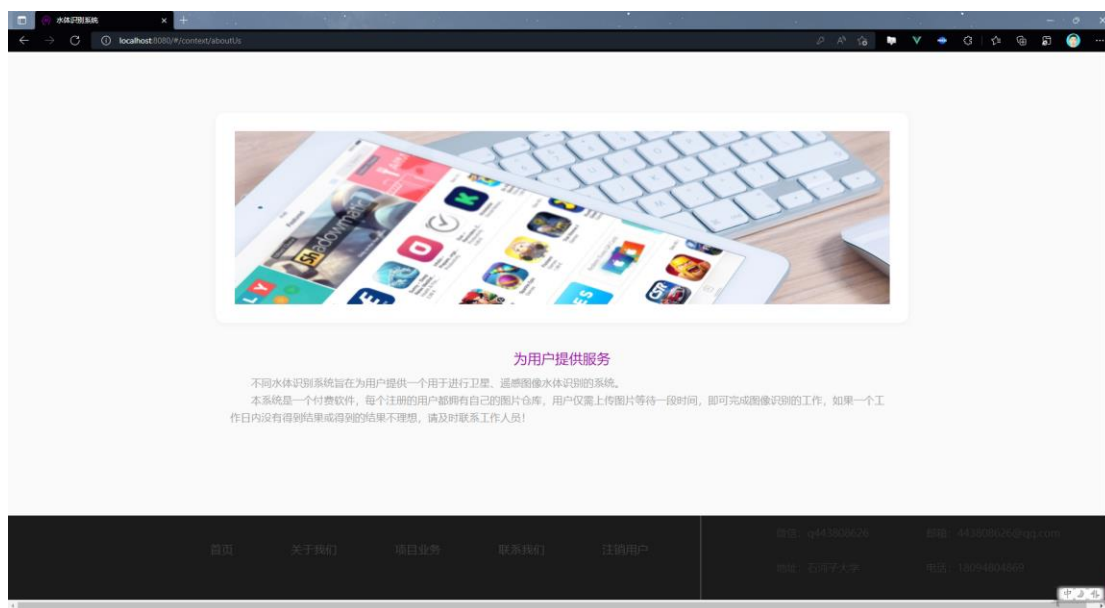


图 5-8 关于我们 3

5.3.4. 项目业务

该栏包含业务核心，左侧导航栏分为个人资料、新建仓库、查看仓库、修改密码、注销用户。

个人资料包含用户姓名、用户邮箱（登录名）、用户密码、用户权限、用户余额、可建仓库数量。可以通过刷新按钮刷新信息，也可以通过提交按钮修改用户个人资料。

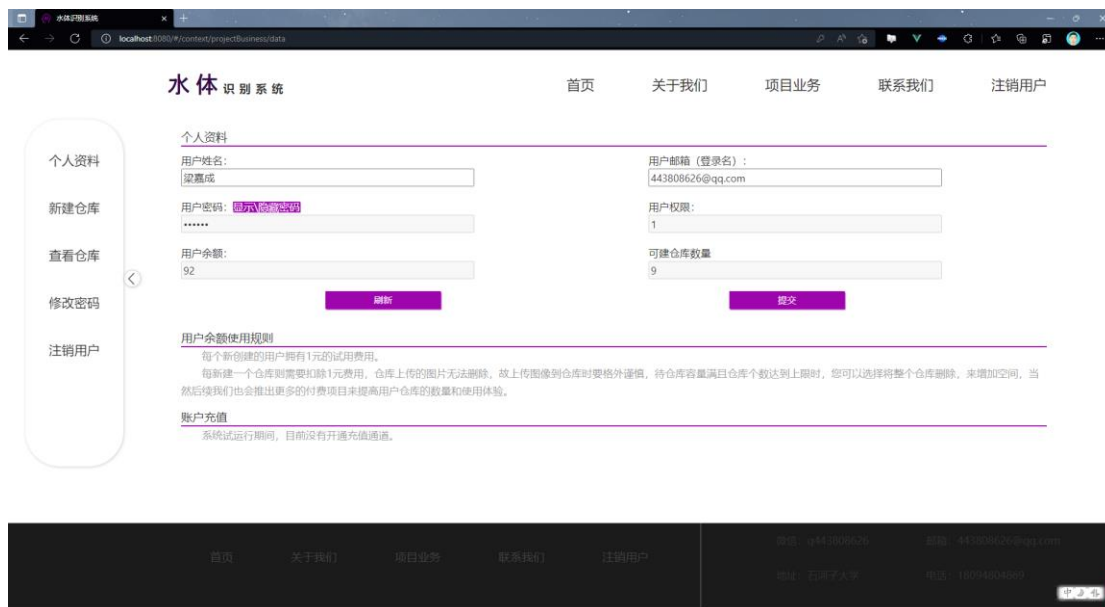


图 5-9 个人资料

图 5-10 展示的是个人资料修改确认时的界面展示。

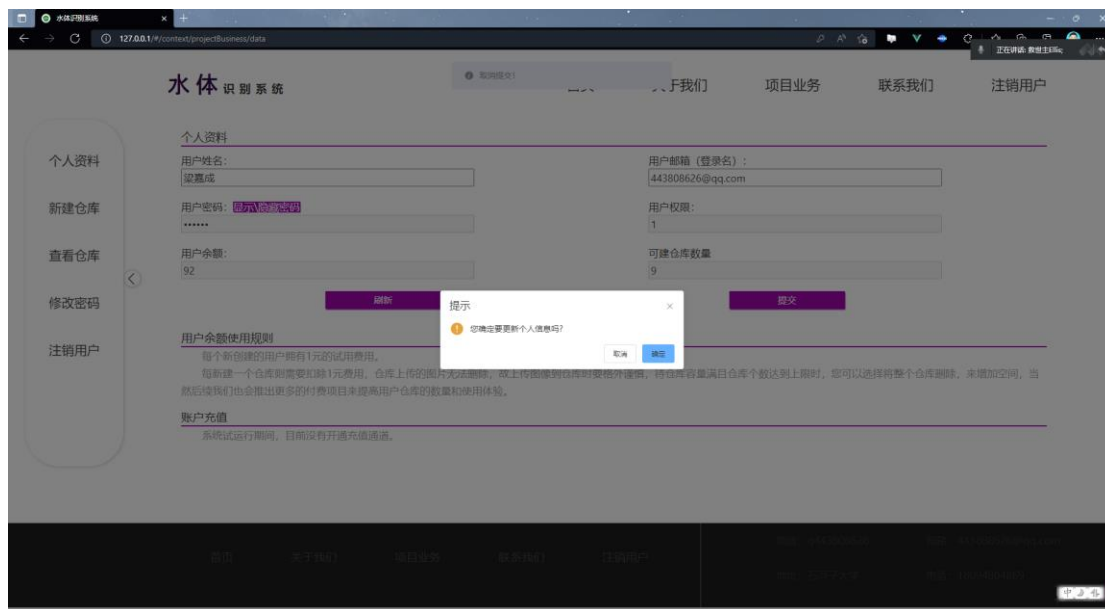


图 5-10 个人资料（修改展示）1

图 5-11 展示的是个人资料修改取消时的界面展示。

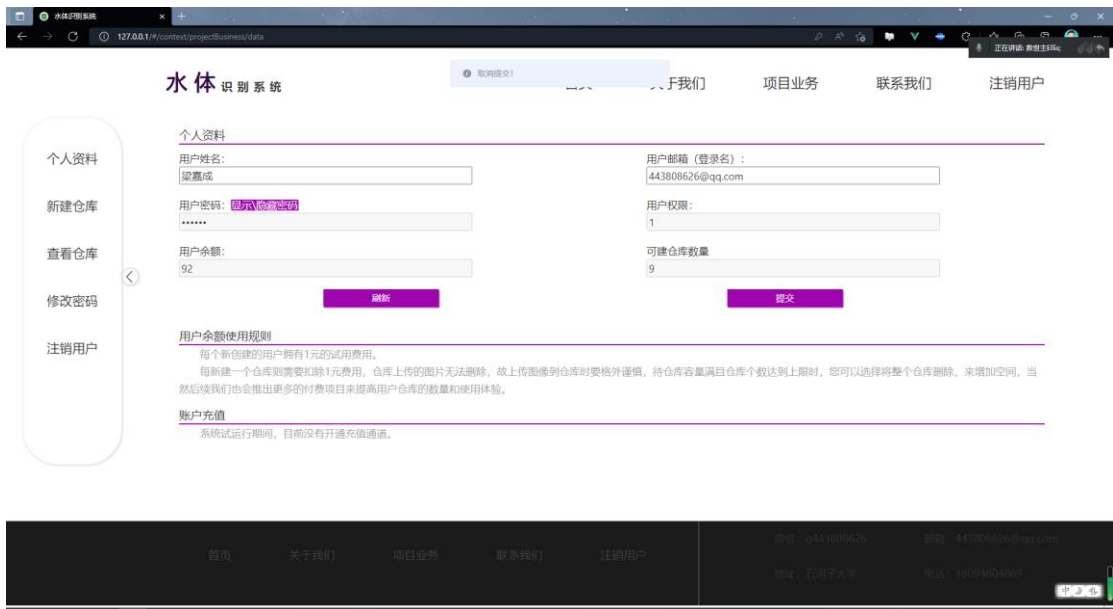


图 5-11 个人资料（修改展示）2

图 5-12 展示的是个人资料修改成功时的界面展示。

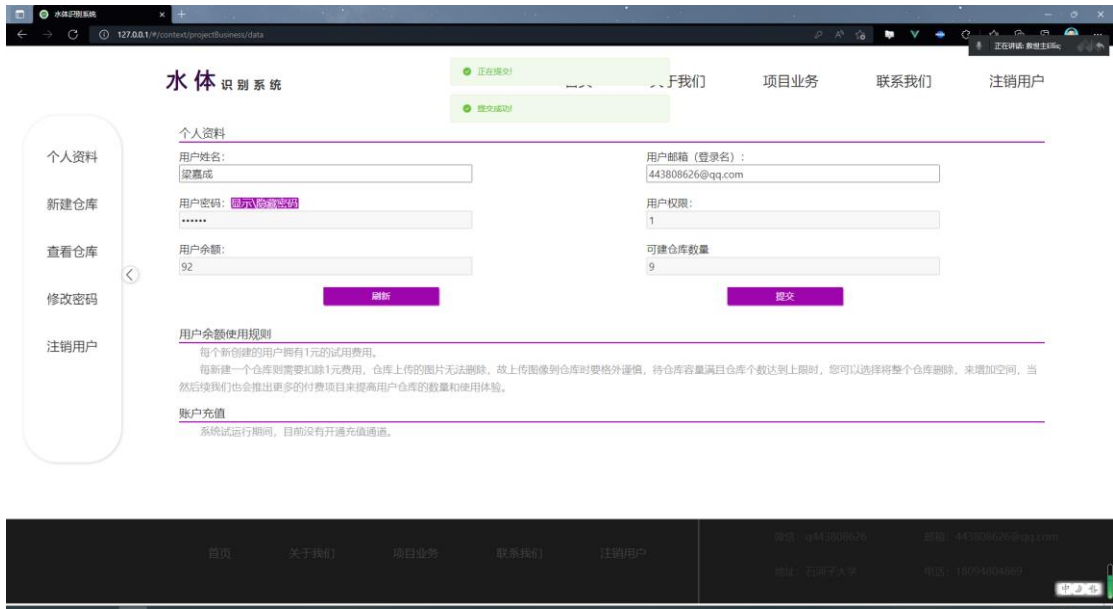


图 5-12 个人资料（修改展示）3

5.3.5. 新建仓库

图 5-13 展示的是新建仓库里包含仓库的新建功能。

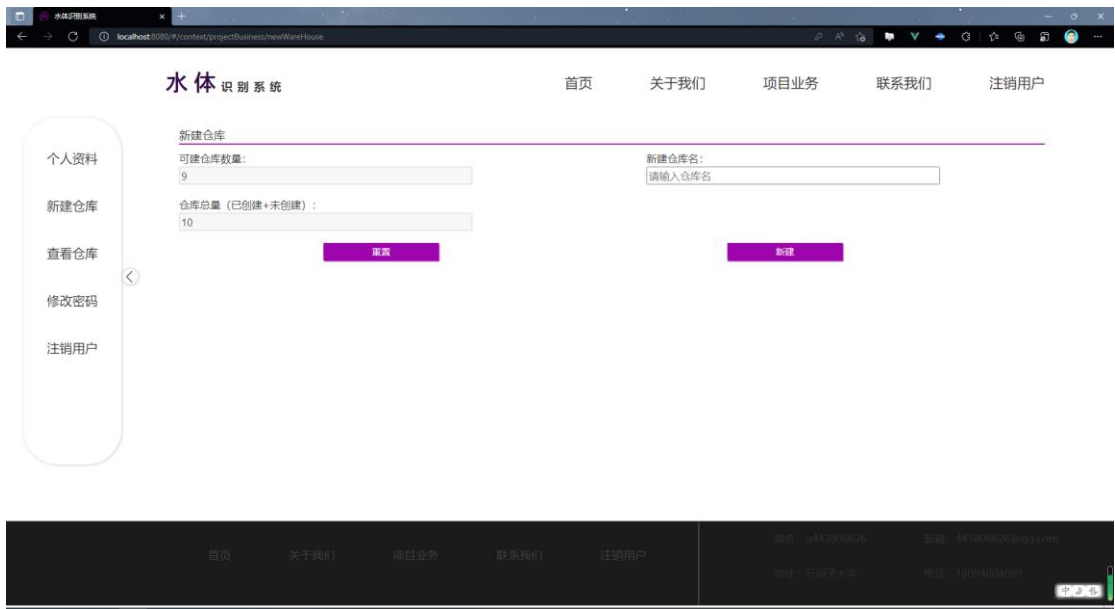


图 5-13 新建仓库

5.3.6. 查看仓库

图 5-14 展示的是查看仓库界面，可以打开和删除仓库。

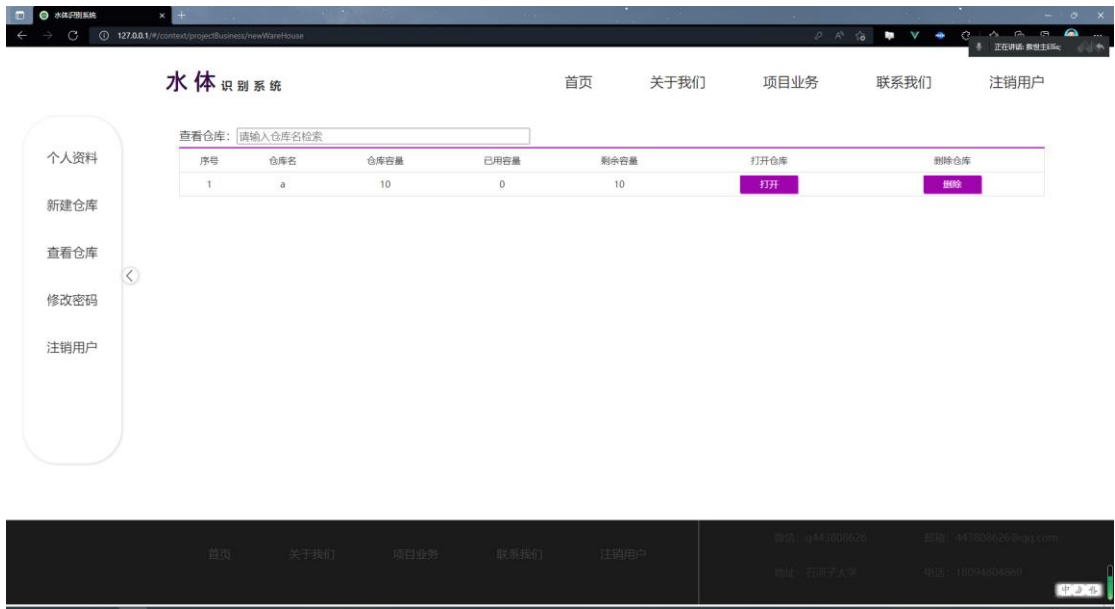


图 5-14 查看仓库

图 5-15 展示的是查看仓库内容界面，其内可以修改仓库名称，查看仓库信息，上传要进行处理图像，刷新查询结果。

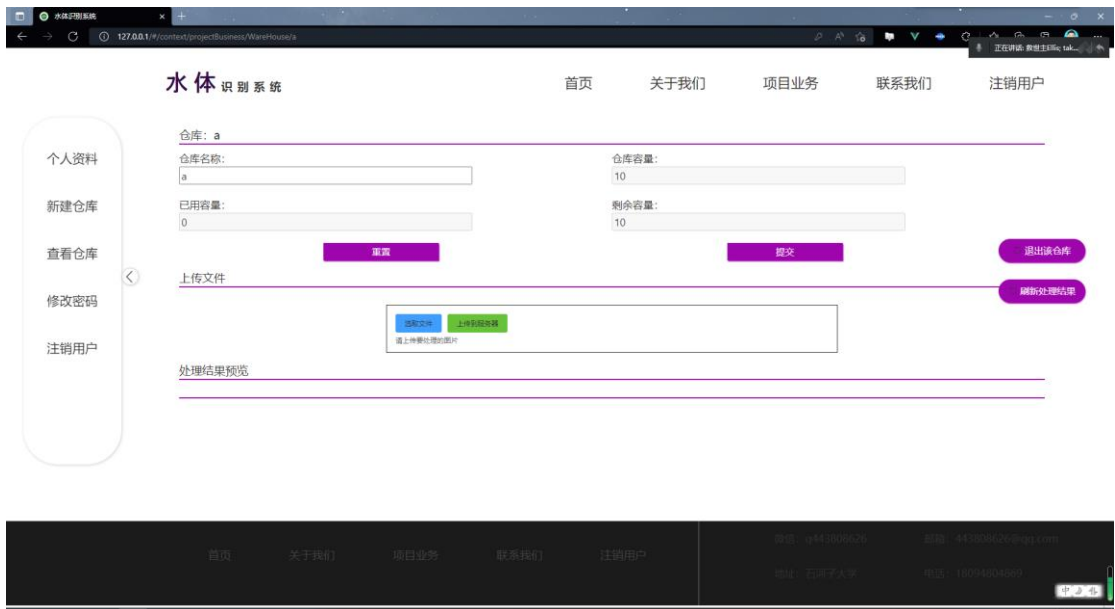


图 5-15 仓库内容

5.3.7. 修改密码

修改密码一栏专门用来修改密码，如图 5-16。

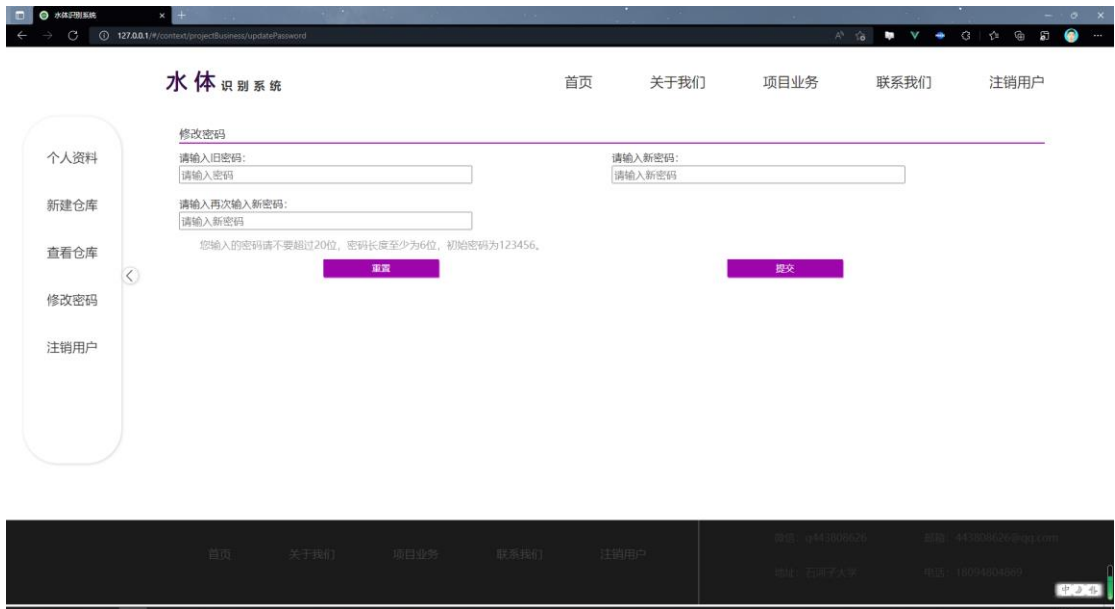


图 5-16 仓库内容

图 5-17 所示的是两次密码输入不正确的情况下所展示的界面。

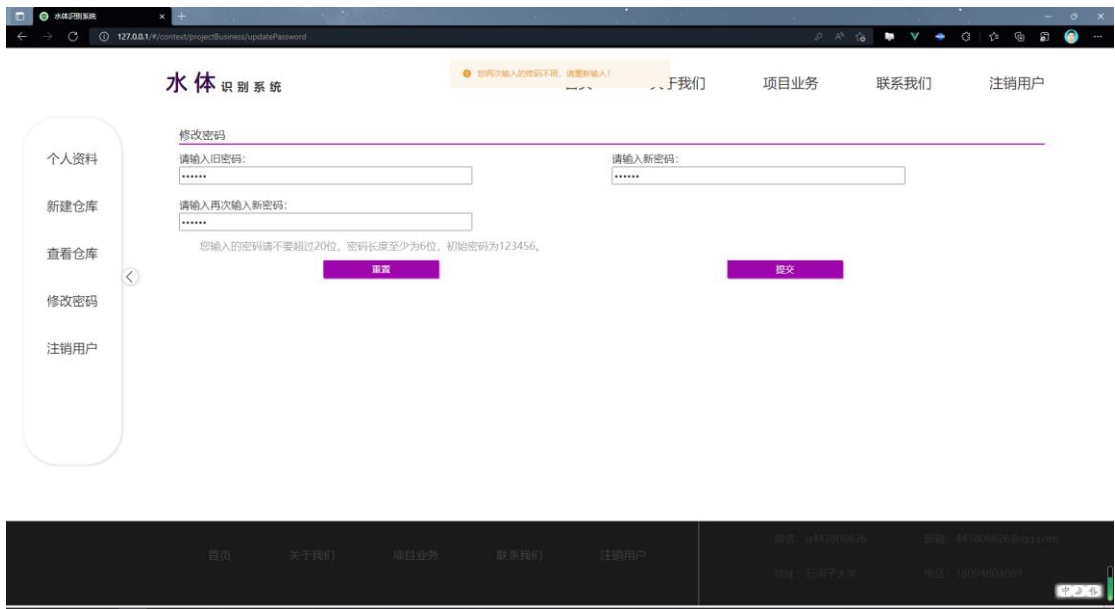


图 5-17 密码不相同时显示

图 5-18 所示，是修改密码成功时的情况。

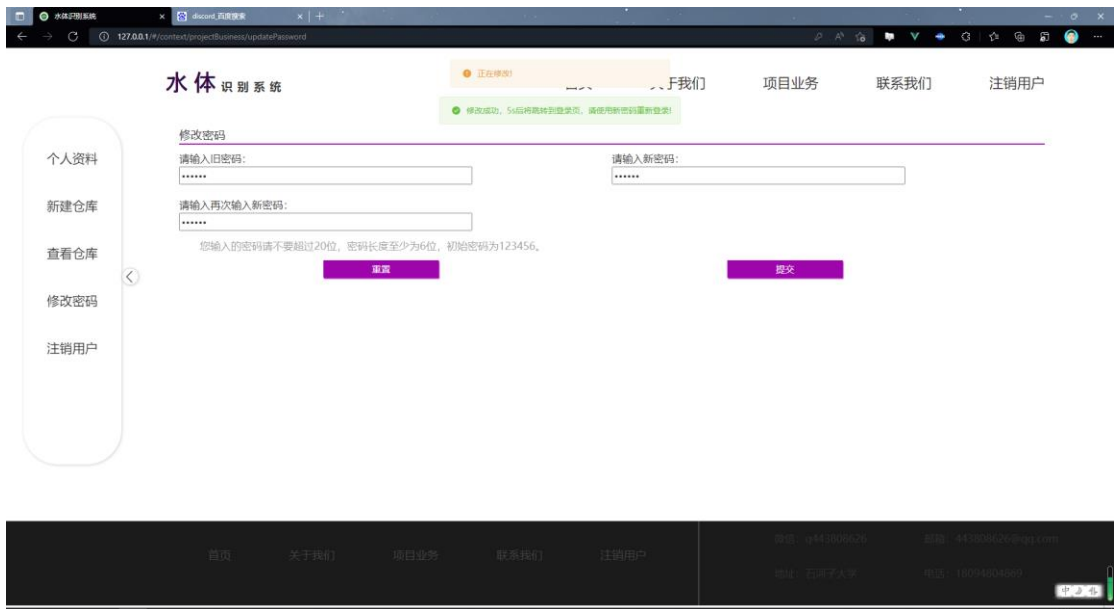


图 5-18 仓库内容

5.3.8. 联系我们

该栏目主要是本系统作者的联系方式。

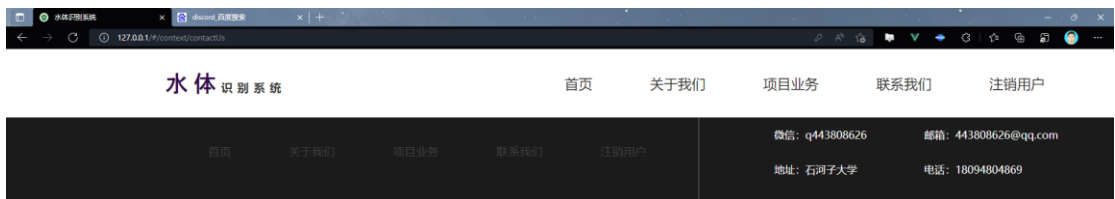


图 5-19 仓库内容

5.4. 后端实现

5.4.1. 前端控制器逻辑

由于前端采用单页面应用，则前端的增、删、改、查一律应用 AJAX 请求进行，选用 Axios 框架发起请求。

后端设计 5 个前端控制器模块分别对应登录控制及 4 个页面处理，分别接收来自前端的项目业务下的 5 个页面发出的 HTTP 请求。

5.4.2. 登录逻辑

登录逻辑使用 JWT 技术验证登录状态，其具体的处理逻辑是前端请求验证码时，向后端发送一段 UUID，后端将这段 UUID 配合向前端返回相应验证码的验证码值一起缓存如 Redis，缓存时间 3 分钟。

当登录时，用户输入用户名密码和验证码发送到后端后，后端首先验证其用验证码是否正确，用来保护 SQL 数据库的安全，验证成功后，查询数据库中的用户名和密码是否能够和前端数据匹配，匹配成功则返回成功状态，并携带 token，前端将 token 存入浏览器 localStorage，在浏览器每次进行页面跳转时验证其状态，每次请求时携带 token，以实现登录状态的记录。

5.4.3. 增、删、改、查逻辑

后端对必要数据进行验证后选用 MyBatis 框架与 SQL 数据库进行交互（如运行上线后流量压力过大，可考虑增加 Redis 缓存保护 SQL 数据库）。

5.4.4. 图像处理逻辑

图像处理采用异步方式，开设一个服务专门作为图像服务，采用 Socket 通信方式 TCP 协议进行通信。

当一个上传图片的请求发送到后端时，后端开启一个线程向图片处理服务发送请求，请求处理图片，并向前端反馈正在处理。

图片处理服务接收到请求，用事先训练好的通过 YOLOv5 训练的模型进行处理，标记不同水体，并返回到指定仓库。

用户可以调用指定仓库的内容查看处理结果,也可以为图片的处理结果进行打分。

6. 系统测试

6.1. 接口测试

利用 Postman 进行测试，主要测试后端接口是否满足需求。

后端的接口包含两部分：一是提供给前端进行 AJAX 请求的前端控制器；二是提供给其他服务的接口。

表 6-1 接口测试表

序号	类型	接口名	url	是否成功
1	loginController	login	/login	成功
2	loginController	checkCode	/checkcode	成功
3	loginController	checkToken	/checkToken	成功
4	Page1Controller	getInfo	/info	成功
5	Page1Controller	updateUserInfo	/updateInfo	成功
6	Page2Controller	getWareHouse	/wareHouse	成功
7	Page2Controller	insertWareHouse	/newWareHouse	成功
8	Page3Controller	selectWareHouseList	/viewWareHouse	成功
9	Page3Controller	deleteWareHouse	/delete	成功
10	Page3Controller	getWareHouseMessage	/getWareHouse	成功
11	Page3Controller	updateWareHouseMessage	/updateWHMessage	成功
12	Page3Controller	upload	/upload	成功
13	Page3Controller	refresh	/refresh	成功
14	Page3Controller	getImg	/getImg	成功
15	Page3Controller	getScore	/submitScore	成功
16	Page4Controller	updatePassword	/updatePassword	成功
17	图像处理微服务	端口：9600		成功

6.2. 单元测试

单元测试主要是对后端单元进行测试，利用 Junit4 进行。

7. 总结与展望

7.1. 总结

本文设计并实现了不同水体识别系统,该系统旨在为用户提供一套水体识别的系统。本文收集了相关系统构建方法、水体识别的方法,通过该系统用户可以登录、充值并进行上传图片,后端通过水体识别算法模型完成图像的处理。

本次设计采用了当前比较流行的开发技术和手段,设计了一套前后端分离系统,前端选用较易上手的 Vue.js 全家桶,后端则采用 Spring 全家桶进行开发,图片识别则采用 YOLOv5 神经网络进行水体识别。

系统的设计遵循着 MVVM 模型¹,实现了视图和模型的分离,实现了低耦合、可重用、独立开发、可测试性。前端视图由浏览器渲染,当数据变化时首先渲染虚拟 DOM²树,通过 diff 算法对比与真实 DOM 树的区别,从而减少渲染,加快前端响应速度;后端由 Spring 框架通过 IOC³来创建和管理对象,加快响应速度,图像处理为了使用户获得良好的体验,采用异步方式进行处理,通过线程池方式,创建一个合理的线程池向图像处理微服务进行通信,图像处理部分则由实现训练好的 YOLOv5 网络进行处理,这样做的好处是,图像处理部分与后端系统耦合度低,系统上线后,可根据用户评分及反馈,及时对神经网络进行更新、迭代、甚至替换,其运维代价相较传统的长时间停机维护较低。

数据库设计则采用缓存机制,启动服务时对缓存进行预热,查询时需要先查询 Redis 缓存中的数据,若没有查询到则查询 MySQL 数据库,增、删、改时为保证数据库一致性问题,这里采用先更新关系型数据库,再删除 Redis 缓存。

7.2. 展望

通过本次毕业设计,我们学习和了解了三个方向的技术,下面是对着三个技术的未来展望。

(1) 前端部分

¹ Model-View-ViewModel 它本质上就是 MVC 的改进版,是将其中的 View 的状态和行为抽象化,让我们将视图 UI 和业务逻辑分开

² Document Object Model 文档对象模型,它将网页中的各个元素都看作一个个对象,从而使网页中的元素也可以被计算机语言获取或者编辑。

³ Inversion of Control 通过控制反转,对象在被创建的时候,由一个调控系统内所有对象的外界实体将其所依赖的对象的引用传递给它。

本设计仍然使用的是 Vue2 框架，但 Vue3 以在 2022 年的 2 月 7 日成为了新的默认版本，相比 Vue2，Vue3 有了更多在性能上的提升，例如 diff 算法的优化等，并且更重视视图和业务层，分离做的更好，有了更好 TypeScript 的支持；随着前端框架的不断发展，前端也逐步发展出微服务框架以适应不断变化的用户需求。

（2）后端部分

后端部分则更应注重安全和容灾，作为毕业设计的 Demo 我们只注重了实现功能和内容，对于数据库安全、缓存雪崩、击穿、穿透等内容考虑较少，为了解决该问题。为了数据库的安全，在未来实现中我们应该考虑在数据库的集群实现、容灾备份中占有更多的份额，而在服务器的设计中应该考虑使用 Nginx¹实现对静态资源和动态资源的分离，并且要针对各种可能的攻击例如 DDOS²等作出相应的防护。

（3）深度学习

为了实现图像识别，我们采用了比较著名的卷积神经网络——YOLOv5 进行训练实现，并没有专门针对此类（水体）图像识别去专门考虑使用专项算法和模型。若实现过程中有较多的时间，我们可以考虑采用多种算法实现功能，这也体现了我们采用微服务拆分图像识别部分的优点。

¹ 是一个高性能的 HTTP 和反向代理 web 服务器

² 分布式拒绝服务攻击

8. 主要参考文献

- [1] 畅鑫,李艳斌,田淼,等. 基于一维卷积循环神经网络的深度强化学习算法[J]. 计算机测量与控制: 1-8.
- [2] 百科出版社[M]: 百科出版社.
- [3] 何海清,杜敬,陈婷,等. 结合水体指数与卷积神经网络的遥感水体提取[J]. 遥感信息, 2017, 32(5): 82-86.
- [4] 王坤峰,苟超,段艳杰,等. 生成式对抗网络 GAN 的研究进展与展望[J]. 自动化学报, 2017, 43(3): 321-332.
- [5] 杨巨成,韩书杰,毛磊,等. 胶囊网络模型综述[J]. 山东大学学报(工学版), 2019, 49(6): 1-10.
- [6] 曾子悦,许继军,王永强. 基于遥感空间信息的洪水风险识别与动态模拟研究进展[J]. 水科学进展, 2020, 31(3): 463-472.
- [7] Faghanpour Mehdi,Jahanifar Komeil,Seresht Kazem-Nik,等. The effect of using treated sewage on irrigating urban forested areas[C]//Proceedings of International Conference on Environmental Engineering and Applications (ICEEA 2011) , 2011: 106-111.
- [8] 徐涵秋. 新型 Landsat8 卫星影像的反射率和地表温度反演[J]. 地球物理学报, 2015, 58(3): 741-747.
- [9] Al-quraishi Ayad-M-F,Gaznayee Heman-A,Crespi Mattia. Drought trend analysis in a semi-arid area of Iraq based on Normalized Difference Vegetation Index, Normalized Difference Water Index and Standardized Precipitation Index[J]. Journal of Arid Land, 2021, 13(4): 413-430.
- [10] 陈文倩,丁建丽,李艳华,等. 基于国产 GF-1 遥感影像的水体提取方法[J]. 资源科学, 2015, 37(6): 1166-1172.
- [11] 王瑾杰,丁建丽,张成,等. 基于 GF-1 卫星影像的改进 SWI 水体提取方法[J]. 国土资源遥感, 2017.
- [12] 王小标,谢顺平,都金康. 水体指数构建及其在复杂环境下有效性研究[J]. 遥感学报, 2018, 22(2): 360-372.
- [13] 王琳,谢洪波,文广超,等. 基于 Landsat8 的含蓝藻湖泊水体信息提取方法研究[J]. 国土资源遥感, 2020, 32(4): 130-136.
- [14] 尹小君,祝宏辉,GAO Jerry,高军,郭丽洁,苟贞珍. 基于 Landsat 和 MODIS 数据融合的农牧区 NPP 模拟[J]. 农业机械学报, 2020, 51(08): 163-170.
- [15] 赵艳玲,丁宝亮,何厅厅,肖武,任河. 基于 Google Earth Engine 的采煤沉陷水体方向变化自动识别[J/OL]. 煤炭学报: 1-10[2021-12-18]. <http://kns.cnki.net/kcms/detail/11.2190.TD.20211105.1311.003.html>.
- [16] 尹小君,宁川,韩峰,张雅,高军. 天山北坡土壤盐渍化光谱响应特征与动态监测的研究[J]. 江苏农业科学, 2019, 47(16): 277-281. DOI:10.15889/j.issn.1002-1302.2019.16.060.
- [17] 冯春,赵南京,殷高方,甘婷婷,陈晓伟,陈敏,华卉,段静波,刘建国. 多波长透射光谱特征提取结合支持向量机的水体细菌识别方法研究[J]. 光谱学与光谱分析, 2021, 41(09): 2940-2944.
- [18] 尹小君,宁川,张永才. 加工番茄早疫病高光谱遥感识别研究[J]. 遥感信息, 2015, 30(02): 94-98.
- [19] 韩利冬. 高分 2 号遥感影像典型自然要素的自动识别方法研究[D]. 山东农业大学, 2020.
- [20] 尹小君. 基于遥感技术的新疆玛纳斯县土地利用变化的分析[J]. 石河子大学学报(自然科学版), 2008(04): 402-406. DOI:10.13880/j.cnki.65-1174/n.2008.04.033.
- [21] 杜敬. 基于深度学习的无人机遥感影像水体识别[J]. 江西科学, 2017, 35(01): 158-161+170.
- [22] 徐文健. 基于卷积神经网络的高分辨率遥感图像上的水体识别技术[D]. 浙江大学, 2018.

- [23]徐蓉,张增祥,赵春哲.湖泊水体遥感提取方法比较研究[J].遥感信息,2015,30(01):111-118.
- [24]莫伟华,孙涵,钟仕全,黄永璘,何立.MODIS 水体指数模型(CIWI)研究及其应用[J].遥感信息,2007(05):16-21+104-105.
- [25]吴赛,张秋文.基于 MODIS 遥感数据的水体提取方法及模型研究[J].计算机与数字工程,2005(07):1-4.
- [26]都金康,黄永胜,冯学智,王周龙.SPOT 卫星影像的水体提取方法及分类研究[J].遥感学报,2001(03):214-219.
- [27]王仁军,李东颖,刘宝康.基于高分六号 WFV 数据的可可西里湖泊水体识别模型研究[J/OL].自然资源遥感:1-12[2021-12-18].<http://kns.cnki.net/kcms/detail/10.1759.P.20211119.1637.002.html>.
- [28]王伟,阿里木·赛买提,马龙,葛拥晓,吉力力·阿不都外力.1986—2019 年新疆湖泊变化时空特征及趋势分析[J/OL].生态学报,2022(04):1-15[2021-12-18].<http://kns.cnki.net/kcms/detail/11.2031.Q.20211103.1335.038.html>.
- [29]顾佳艳,何国富,占玲骅,胡元树,孔维鑫,杨根森.上海市黑臭水体光谱特征分析及遥感识别模型构建[J/OL].环境科学研究:1-14[2021-12-18].<https://doi.org/10.13198/j.issn.1001-6929.2021.10.04>.
- [30]苑玉彬. 基于深度学习的寒旱区遥感影像水体识别研究[D].兰州交通大学,2021.
- [31]刘瑶. 基于深度学习的多光谱遥感影像水体识别[D].南京信息工程大学,2021.
- [32]薛祥祥,罗泽.青海湖区域水体识别系统设计[J].计算机系统应用,2018,27(09):68-73.
- [33]何海清,杜敬,陈婷,陈晓勇.结合水体指数与卷积神经网络的遥感水体提取[J].遥感信息,2017,32(05):82-86.
- [34]温爽,王桥,李云梅,朱利,吕恒,雷少华,丁潇蕾,苗松.基于高分影像的城市黑臭水体遥感识别:以南京为例[J].环境科学,2018,39(01):57-67.

9. 致谢

首先,感谢学校和学院对二学位学生的照顾,回望大半年以来从收集整理资料、思考解决方案、最终实现业务逻辑,少不了学校和学院的支持。

其次,感谢我的指导老师,在我撰写论文中给予我的帮助。

最后,感谢我的同学们在疫情期间能够帮我再学校内处理一些必要性的工作。