

本科实验报告

课程名称: B/S 体系软件设计

姓 名: 姚熙源

学 院: 计算机科学与技术学院

系: 软件工程

专 业: 软件工程 1902

学 号: 3190300677

指导教师: 胡晓军

2021 年 10 月 30 日

浙江大学实验报告

课程名称: B/S 体系软件设计 实验类型: 综合型

实验项目名称: 图像数据标注网站

学生姓名: 姚熙源 专业: 软件工程 学号: 3190300677

同组学生姓名: 无 指导老师: 胡晓军老师

实验地点: 玉泉校区 实验日期: 2021 年 10 月 30 日

图像数据标注网站-设计报告

目录

1. 项目背景	4
2. 系统需求分析.....	4
2.1 功能性需求分析	4
2.2 非功能需求分析	4
2.2.1 性能需求.....	4
2.2.2 输入输出需求.....	4
2.2.3 数据管理需求.....	4
3. 系统技术与架构设计.....	5
3.1 项目技术选型	5
3.2 项目主要技术介绍	5
3.2.1 前端技术框架.....	5
3.2.2 后端技术框架.....	5
3.3 系统总体架构图	5
3.4 前端框架构图	6
3.5 后端框架构图	6
3.6 系统运行环境	7
3.6.1 软件环境.....	7
3.6.2 硬件环境.....	7
4. 数据库设计与 ER 图.....	7
4.1 数据库表设计	7

4.1.1 user 表.....	7
4.1.2 image 表.....	7
4.2 ER 图	8
5. 系统接口设计.....	8
5.1 用户信息接口设计	8
5.1.1 用户登录.....	8
5.1.2 用户注册.....	8
5.1.3 修改密码.....	8
5.1.4 修改邮箱.....	8
5.1.5 修改手机号.....	9
5.2 任务信息接口设计	9
5.2.1 发布任务.....	9
5.2.2 修改任务信息.....	9
5.2.3 领取任务.....	9
5.2.1 查看任务列表.....	9
6. 系统界面模型.....	9
6.1 登录界面	10
6.2 注册界面	10
6.3 首页界面	10
6.4 个人信息界面	11
6.5 任务界面	11
6.6 发布任务界面	11
7. 附录	11
7.1 项目进度安排	11

1. 项目背景

本项目是《B/S 体系软件设计》的课程项目，是由我个人单独开发，项目是设计一个用于机器学习的图像数据标注网站。本项目提供给用户的功能有登录、注册、更改个人信息、发布任务、领取任务等。本网站提供友好的交互界面，给予用户良好的体验，样式适配手机端，可以在手机浏览器中提供友好的界面显示。同时，开发者也会提供本项目的各种开发文档（设计、测试、用户手册等），使整个开发工程与流程都得到完善的记录。

本文是项目的设计文档，包含了系统的需求分析、架构设计、数据库设计、接口设计已经界面设计等，由本文介绍该系统的具体设计方案。

2. 系统需求分析

2.1 功能性需求分析

本项目是开发一个 B/S 架构的 Web 应用，对用户需要实现的功能需求：

- 用户注册（用户名、密码、邮箱、手机号等）
- 用户登录
- 用户信息修改
- 提供批量图片、视频上传功能
- 上传的视频能够转换为多张图片，提供界面选择有效图片或按照一定规则（如间隔若干帧）自动提取图片
- 任务发布者选择若干图片创建标注任务，其他人可领取任务进行数据标注
- 标注结果支持导出多种数据集的格式

本项目的用户分为两种，一种是普通用户，另一种则是管理员，普通用户可发布和领取任务，管理员则可审核提交的任务确保数据标注正确。

项目的功能需求可分两个模块，一个是用户信息模块，另一个是任务管理模块。其余功能模块还可分为若干子功能模块，提供更多用户需求。

2.2 非功能需求分析

2.2.1 性能需求

- 系统运行稳定，避免出现崩溃
- 大多数浏览器能正常访问本系统
- 系统应能保证至少 100 人同时访问
- 用户登录后执行的操作，系统应及时反馈，反馈时间在 1 秒以内
- 页面加载一般在 1 秒内完成，多人访问则在 5 秒内加载完毕
- 系统保证一周内不超过一次维护或重启。
- 系统应及时检测异常情况。

2.2.2 输入输出需求

- 用户输入账号或密码时，对数据输入进行有效性检测，同时确保密码安全性
- 任务界面显示数据库里所有的发布任务提供用户领取。
- 由系统控制出错几率，减少系统因人为的错误引起系统的出错或破坏，尽量对系统可能发生的问题考虑周全，使出错率降低。

2.2.3 数据管理需求

- 系统既要与其他系统有接口，又必须保证本系统的独立性与完整性。即应防止未经授权的各类人员对本系统进行设置和修改或访问系统内部数据。
- 系统服务器软件必须提供可靠的数据备份和恢复手段，在服务器软件或硬件出现严重

故障时，能够根据备份的数据和账户信息迅速恢复正常运行环境。

- 系统应具备加密登录、数据加密传输等安全方面的保障，保证数据在不用系统间传输过程中的保密性与安全性。

3. 系统技术与架构设计

3.1 项目技术选型

本项目的开发技术是采用前后端分离的 Web 开发技术，后端采用 RESTful API 的架构规范来提高前端调用，前端则采用 React 的 Ant Design 组件库设计友好的界面让用户体验良好。整个项目所采用的技术如下：

- 前端：React+Ant Design 组件库，npm 包管理工具
- 后端：Java Spring Boot 框架+Maven
- 数据库：MySQL

3.2 项目主要技术介绍

3.2.1 前端技术框架

本项目采用的前端技术是 React+Ant Design 组件库以及 npm 包作为管理工具。

React 是由 Facebook 公司（现 Metaverse）的软件工程师 Jordan Walke 开发的 Web 开发框架，是一个为数据提供渲染为 HTML 视图的开源 JavaScript 库来设计界面。React 视图通常采用包含以自定义 HTML 标记规定的其他组件的组件渲染。React 为程序员提供了一种子组件不能直接影响外层组件（“data flows down”）的模型，数据改变时对 HTML 文档的有效更新，和现代单页应用中组件之间干净的分隔，为程序员提供灵活的开发应用。

Ant Design 是由蚂蚁集团所开发的企业级产品系统框架，提供了大量的功能性组件库。

npm（Node Package Manager，即“node 包管理器”）是 Node.js 默认的、用 JavaScript 编写的软件包管理系统。

3.2.2 后端技术框架

本项目采用的后端技术是 Java Spring Boot 框架+Maven，同时使用 MySQL 数据库作为关系数据库以连接数据。Maven 是 Java 工程项目中的包管理工具。

Spring Boot 是由 Pivotal 团队提供的 Java 开源框架，其设计目的是用来简化新 Spring 应用的初始搭建以及开发过程。该框架使用了特定的方式来进行配置，从而使开发人员不再需要定义样板化的配置。通过这种方式，Spring Boot 致力于在蓬勃发展的快速应用开发领域成为领导者。

Apache Maven 是一个软件（特别是 Java 软件）项目管理及自动构建工具，由 Apache 软件基金会所提供。基于项目对象模型（缩写：POM）概念，Maven 利用一个中央信息片段能管理一个项目的构建、报告和文档等步骤。

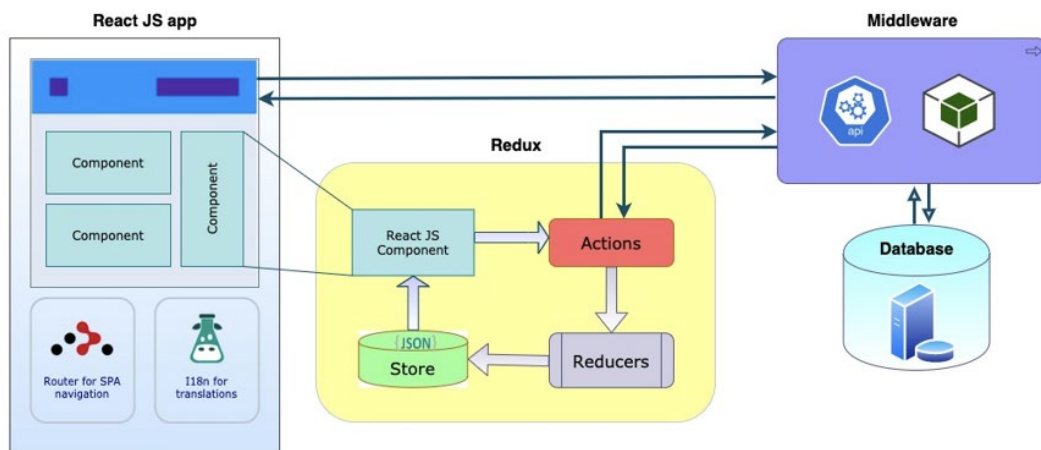
3.3 系统总体架构图

系统的总体框架如图所示，客户端支持电脑端和手机端的访问，服务端则有后端的服务器（web）以及数据库 MySQL 对数据进行交互。



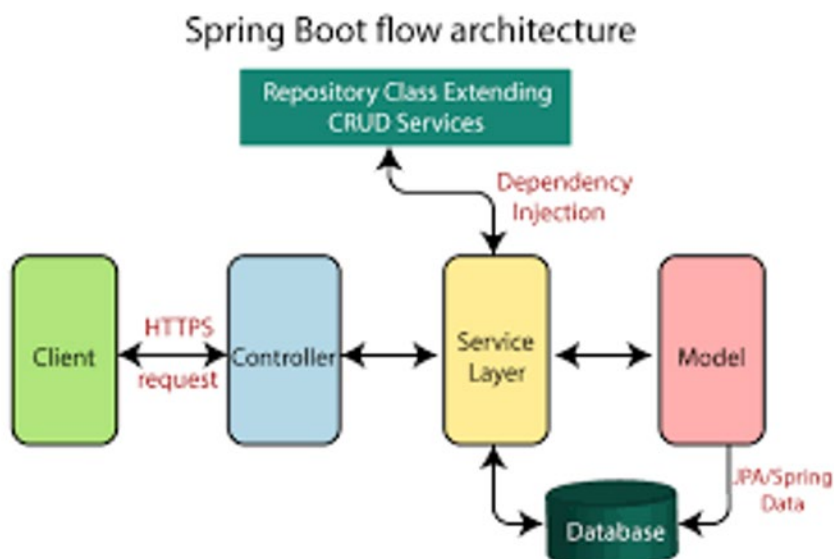
3.4 前端框架图

前端采用的是基于 Node.js 的 React 框架，并使用蚂蚁集团的开源项目 Ant Design 组件库作为 UI 组件框架，对本项目进行前端开发，设计友好的视图界面。前端大致框架可以参考以下的图来表示：



3.5 后端框架图

后端采用 Spring boot 框架编写，以下架构图所示由客户端传送图片数据到数据库以及从数据库提取数据显示到网站界面的过程。



3.6 系统运行环境

3.6.1 软件环境

环境	类型	版本
操作系统	Windows	7 及以上
数据库类型	MySQL	8.0 及以上
浏览器	Chrome	任何版本

3.6.2 硬件环境

项目	名称
操作系统	CPU CORE i5 及以上
	内存：2G 及以上
	硬盘：256G 及以上
数据库服务器 应用服务器	内存：512M 及以上
	硬盘：50G 及以上
网络通讯设备	网络：具有良好的数据传输能力

4. 数据库设计与 ER 图

4.1 数据库表设计

本项目一共

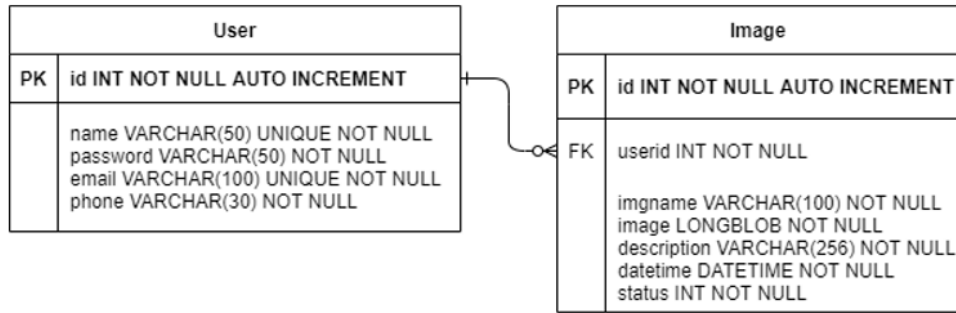
4.1.1 user 表

字段名	类型	描述	备注
id	int	用户 id	主键、非空、自增
name	varchar(50)	用户名	非空、唯一
password	varchar(50)	用户密码	非空
email	varchar(100)	用户邮箱	非空、唯一
phone	varchar(30)	用户手机号	非空

4.1.2 image 表

字段名	类型	描述	备注
id	int	图片 id	主键、非空、自增
imgname	varchar(100)	图片名	非空
image	longblob	图片	非空
userid	int	用户 id	非空、外键
receiveuserid	int	领取任务用户 id	空
description	varchar(256)	图片描述	非空
datetime	datetime	上传日期	非空
status	int	图片标注状态	非空、默认 0、1 为领取、2 为审核

4.2 ER 图



5. 系统接口设计

5.1 用户信息接口设计

主要是使用 user 表的数据，并且对 user 表的信息进行更改或添加，提供的接口都是对相关用户进行对接，具体信息可参考以下：

5.1.1 用户登录

接口 URL	/user/login
主要参数	Map<String, Object> form 登录界面提交表单中的信息，用户名、密码。
返回值类型	Map 类型，其中“code”属性为 1 表示成功，0 表示用户不存在，2 表示用户名或密码出错，“token”包含服务器生成的 token。
接口描述	用于网站的登录认证，登录成功生成 token 并返回给用户保存在浏览器。

5.1.2 用户注册

接口 URL	/user/register
主要参数	Map<String, Object> form 注册界面提交的表单信息，用户名、密码、邮箱、手机号等，邮箱和手机的输入正确性以及密码安全性在前端已审核。
返回值类型	整型，1 表示注册成功，0 表示注册失败。
接口描述	用于用户注册新账号，在数据库中添加新的用户信息

5.1.3 修改密码

接口 URL	/user/change/password
主要参数	Map<String, Object> form 新密码和旧密码
返回值类型	整型，1 表示修改成功，0 表示修改失败
接口描述	用户修改密码

5.1.4 修改邮箱

接口 URL	/user/change/email
主要参数	Map<String, Object> form 新邮箱和旧邮箱
返回值类型	整型，1 表示修改成功，0 表示修改失败
接口描述	用户修改邮箱

5.1.5 修改手机号

接口 URL	/user/change/phone
主要参数	Map<String, Object> form 新手机号和旧手机号
返回值类型	整型，1 表示修改成功，0 表示修改失败
接口描述	用户修改手机号

5.2 任务信息接口设计

主要是使用 image 表的数据，并且对 user 表的信息进行更改或添加，提供的接口都是对相关用户进行对接，具体信息可参考以下：

5.2.1 发布任务

接口 URL	/task/post/{#userid}
主要参数	Map 类型，存储需要发布的任务信息
返回值类型	整型，1 表示成功，0 表示失败
接口描述	发布任务存储任务信息

5.2.2 修改任务信息

接口 URL	/task/change/{#taskid}
主要参数	Map 类型，存储需要修改的任务信息。
返回值类型	整型，1 表示成功，0 表示失败
接口描述	修改发布的任务信息

5.2.3 领取任务

接口 URL	/task/receive/{#taskid}
主要参数	String taskid, 使用 taskid 查找对应的任务 id, 然后添加用户的 id 到 image 表中
返回值类型	整型，1 表示成功，0 表示失败
接口描述	用户领取任务，更新数据库信息。

5.2.1 查看任务列表

接口 URL	/task/all
主要参数	Map<String, Object>存储了任务信息
返回值类型	整型，1 表示成功，0 表示失败
接口描述	查询所有发布的任务

6. 系统界面模型

系统的界面设计如下所示

*由于设计报告是在中期阶段完成的，界面模型有可能与最后的界面设计模型有点不同，因此如有差异，还是按照最后的成品为依据，以下界面只是一个最初概念。

6.1 登录界面



6.2 注册界面



6.3 首页界面

*由于后端的数据交互还没连接起来（还不完善），而前端的登录界面已经完善，跳转需要用到数据库里的数据来登录，所以进入不了登录后的界面，首页界面将会在完成成品后再在设计报告上传界面截图。（此界面已经基本完成，基本的html代码已写完，界面截图在提交整个大程时会补上）

```
render(){
  if(!localStorage.getItem("token")){
    console.log("hi");
    return(
      <Redirect to={{pathname: "/login"}}/>
    )
  }
  return(
    <div>
      <Layout style={{ minHeight: '100vh' }}>
        <Sider>
          <MainMenu />
        </Sider>
        <Layout>
          <Header>
            <HeaderTitle/>
          </Header>
          <Content>
            <ContentList/>
          </Content>
          <Footer>
            <MainFooter/>
          </Footer>
        </Layout>
      </Layout>
    </div>
  );
}
```

Html 代码基本完成。

6.4 个人信息界面

*同 6.3

6.5 任务界面

*同 6.3

6.6 发布任务界面

*同 6.3

7. 附录

备注：由于设计报告在项目开发完成前编写，设计报告所描述的界面、功能等有可能会与最终成品有所差异，所以最终的结果都按照最终的成品所展示。

7.1 项目进度安排

时间阶段	计划事项
2021.10	完成系统的前端设计
2021.11	完成系统的后端代码开发
2021.12 下旬	系统集成和测试，编写文档