**3 需求分析**

## 3.1 功能需求

1.用户注册与登录：

用户可以通过注册账号并登录系统，管理员可以为公司内部员工创建账号，用户输入用户名和密码登录系统。

1. 数据采集：

用户可以通过系统导入高质量的钢材表面图片，或直接拍摄材料照片作为检测数据。支持多种数据格式（如JPG、PNG等）的导入。提供拍照功能，确保检测数据的实时性和准确性。

1. 图像预处理：

系统对上传的图像进行预处理，以提高检测准确性。（考虑以下可能性：去噪：减少图像中的噪声干扰。增强：提高图像的对比度和清晰度。裁剪：截取图像中的关键区域，减少不必要的计算。

1. 缺陷检测：

系统利用深度学习算法对钢材表面图片进行缺陷检测。自动识别和定位钢材表面的缺陷。

1. 缺陷分类：

系统对检测到的缺陷进行分类。根据缺陷的类型（夹杂物、补丁、划痕、 其他缺陷）自动分类并标记。

1. 数据存储与管理：

系统支持检测数据的查询、存储、备份与恢复。存储检测结果，包括缺陷类型、位置、数量等信息。支持历史检测记录的查询和导出。提供数据备份与恢复功能，确保数据安全。

1. 报告生成：

系统自动生成检测报告。生成详细的检测报告，包括缺陷类型、位置、数量等信息。支持报告导出为多种格式（如PDF、Excel），便于进一步分析和存档。

1. 个人信息管理：

用户可以修改自己的个人基本信息，以及修改密码。

## 3.2 可靠性和可用性需求

1.数据安全性:系统应采用加密技术保护用户数据的安全，防止数据泄露或被改

2.系统稳定性:系统应具备高可用性，确保在高峰时段或网络波动时仍能正常运行

3.故障恢复:系统应建立完善的故障恢复机制，当发生故障时能迅速恢复服务

4.用户友好性:系统界面应简洁明了，操作流程应直观易懂，确保用户能够轻松上手

## 3.3 接口需求

3.3.1 用户接口

在本车辆缺陷系统中，会有相应用户、管理员操作界面。各级用户角色通过各自的账户密码，登录到各自职能负责的用户界面中，在这些界面中各级用户可以进行其职能权限的操作来完成他们的工作。

3.3.2 软件接口·

idea与MYSQL数据库的连接，在idea操作界面的工具栏中将项目连接到数据库。

3.3.3 通信接口

本系统既使用了内部的局部网络，支持FTP协议，也使用了Internet连接到了网络中，支持TCP/IP等协议。

## 3.4 系统流程图

系统流程图用图形化的符号来记录整个系统和各个模块之间的关系和结构，描述了数据、相关文件和各子系统之间的关系。方便全面的了解系统业务处理的概况，能够作为进一步分析系统的依据。系统流程图分对用户、管理员两种角色如下图所示：

用户：

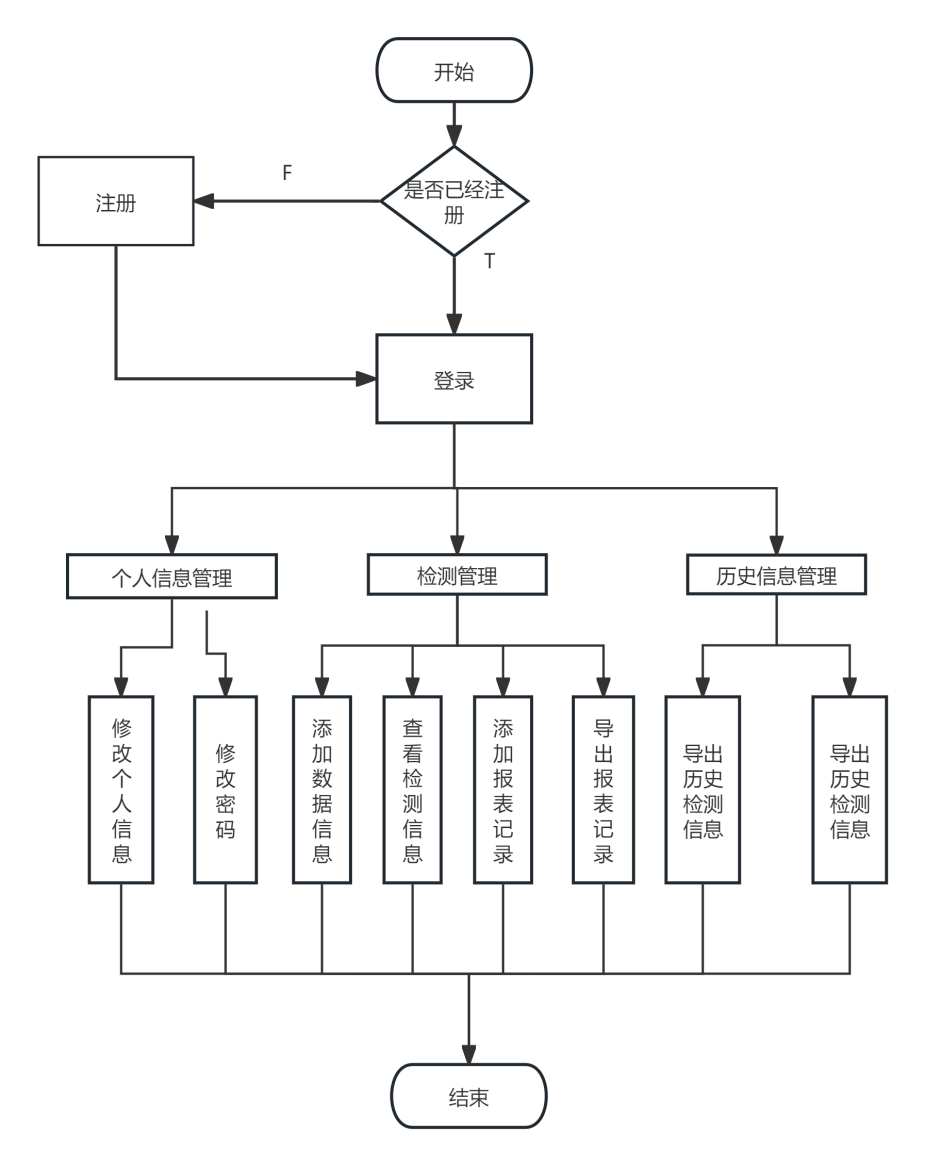


图3-1 用户系统流程图

管理员：

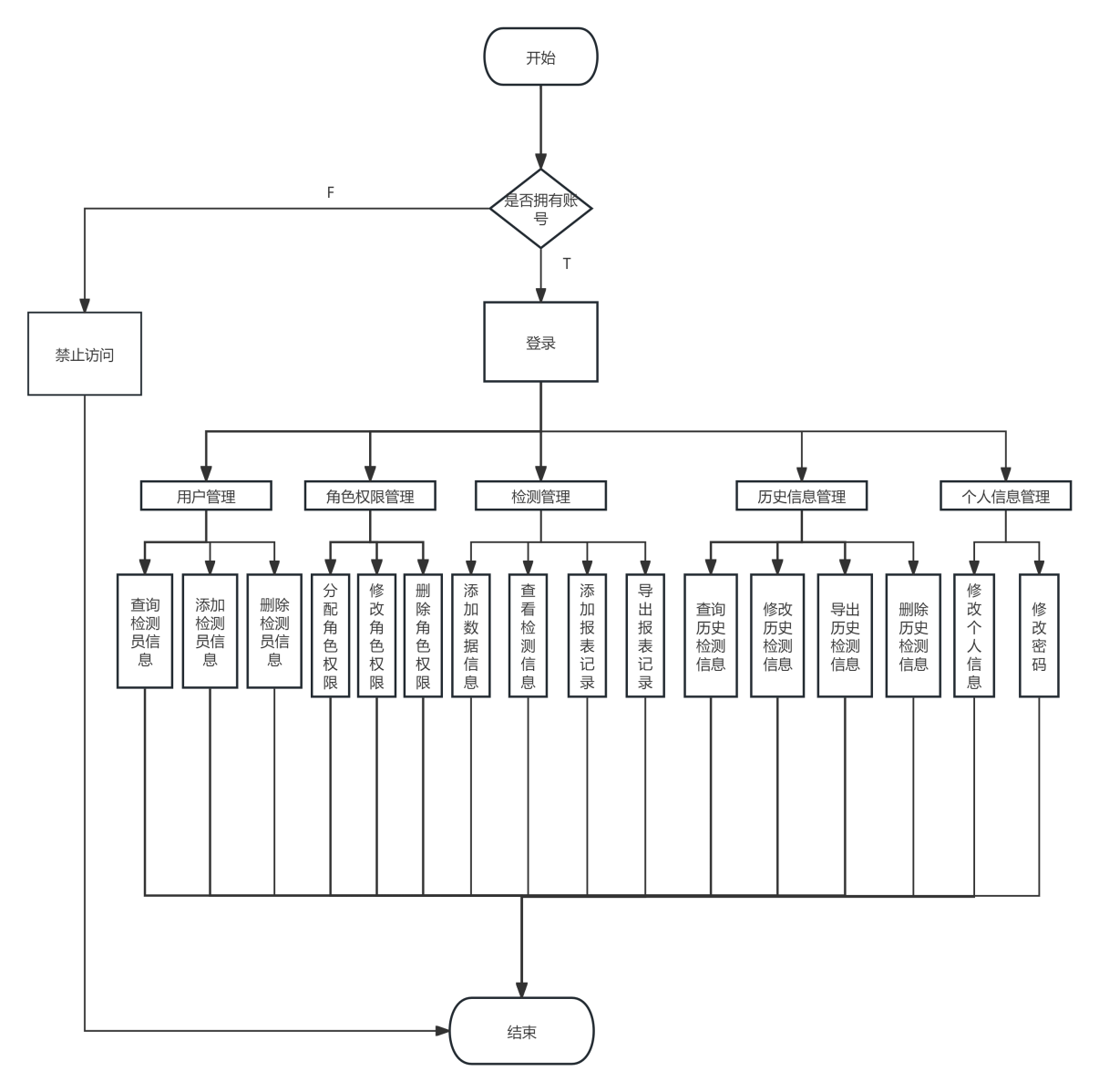


图3-2 管理员系统流程图

**4 总体设计**

## 4.1 基本设计概念和处理流程

### 4.1.1 基本设计概念

本设计是实现一个面向工厂的进行车辆零部件的表面缺陷视觉检测系统，该应用的用户群体是需要进行工厂零件检测的工人，和老板。本设计的开发目的是打造一个更方便的进行零部件的缺陷检测，以及对其检测结果生成报表等功能，能够大大提高员工的工作效率，方便老板直线观察，用户可以直观的检查到问题的存在，做出合理的规划。

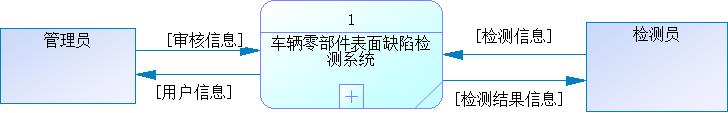


图4-1顶层数据流图

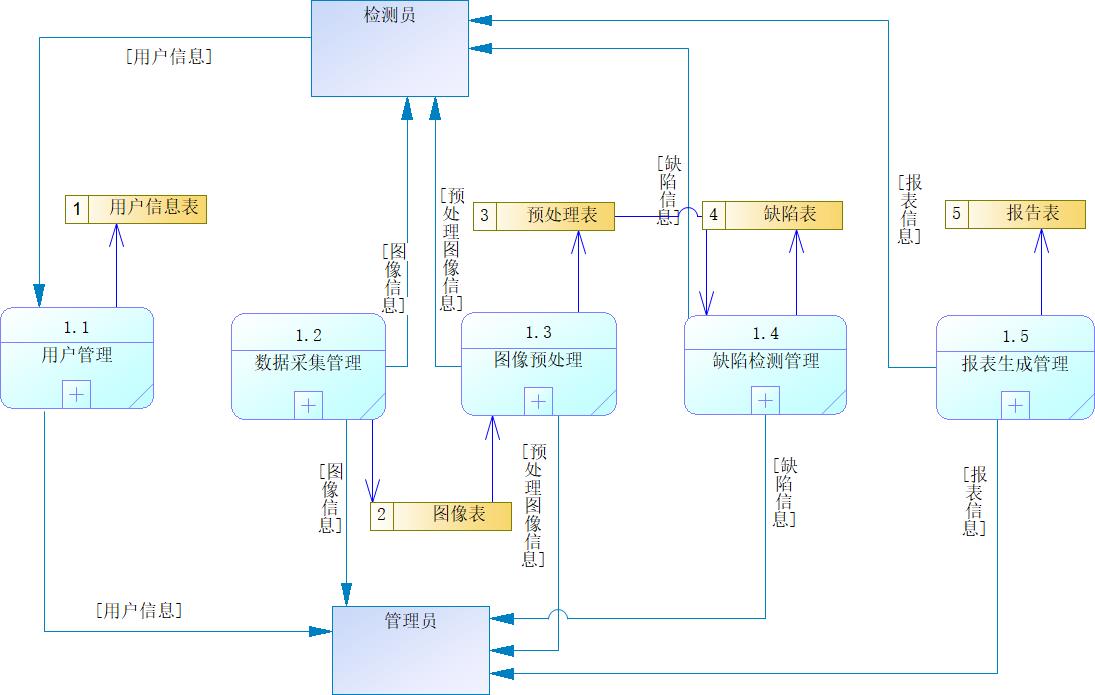


图4-2 0层数据流图

用户管理细分数据流图如下图所示：

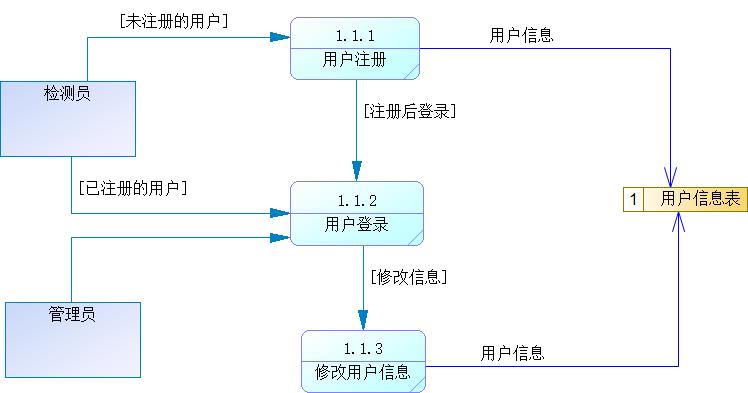


图4-3 用户管理细分数据流图

数据采集管理细分数据流图如下图所示：

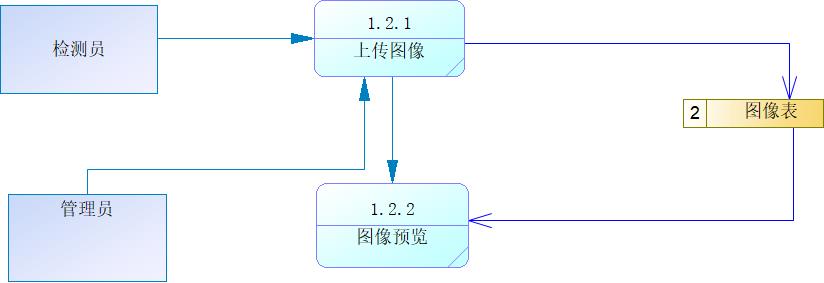


图4-4 数据采集管理细分数据流图

预处理图像细分数据流图如下图所示：

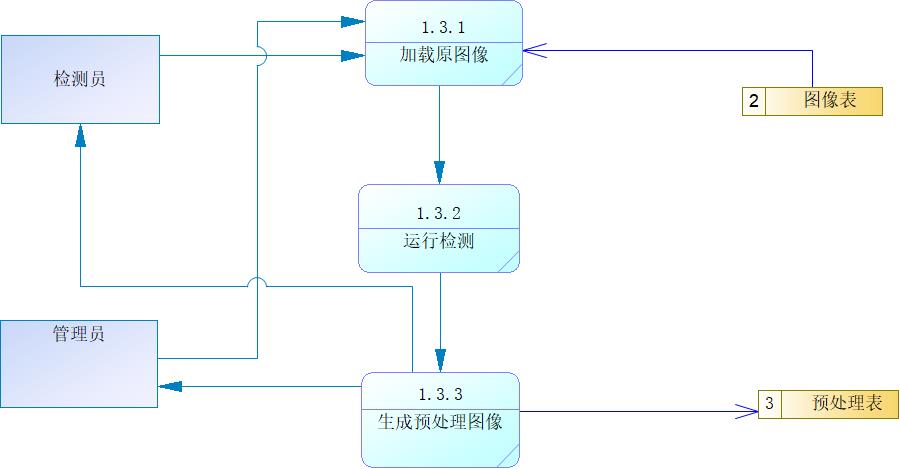


图4-5 预处理图像细分数据流图

缺陷检测管理细分数据流图如下图所示：

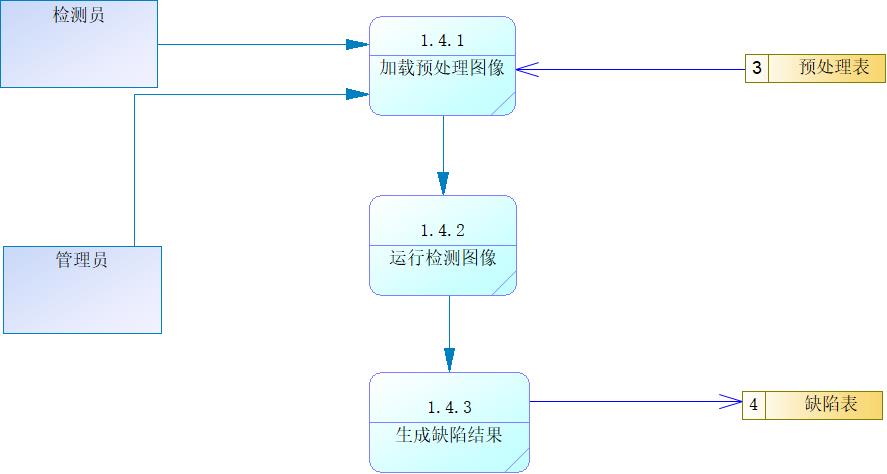


图4-6缺陷检测管理细分数据流图

报表生成管理细分数据流图如下图所示：

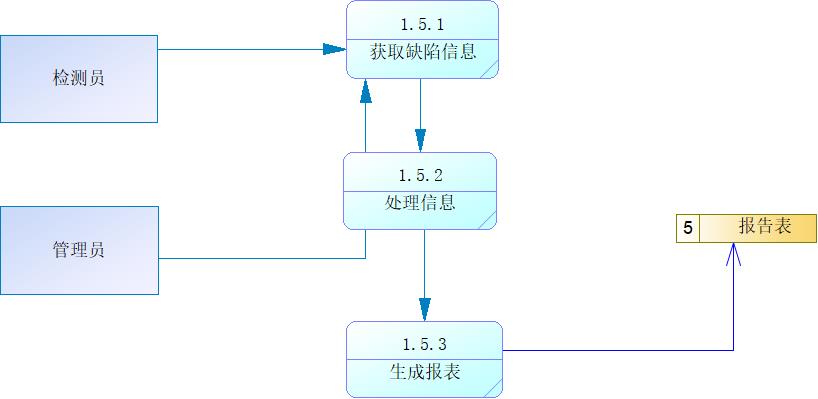


图4-7产生报表管理细分数据流图

## 4.2 软件功能总体结构

系统整体由用户管理模块、数据采集模块、预处理模块、缺陷检测模块、数据查询模块、报告管理模块七个模块构成。每个模块阐述着相应的功能，基本每个功能都实现着相应板块的增删改查。

功能结构图如下图所示：

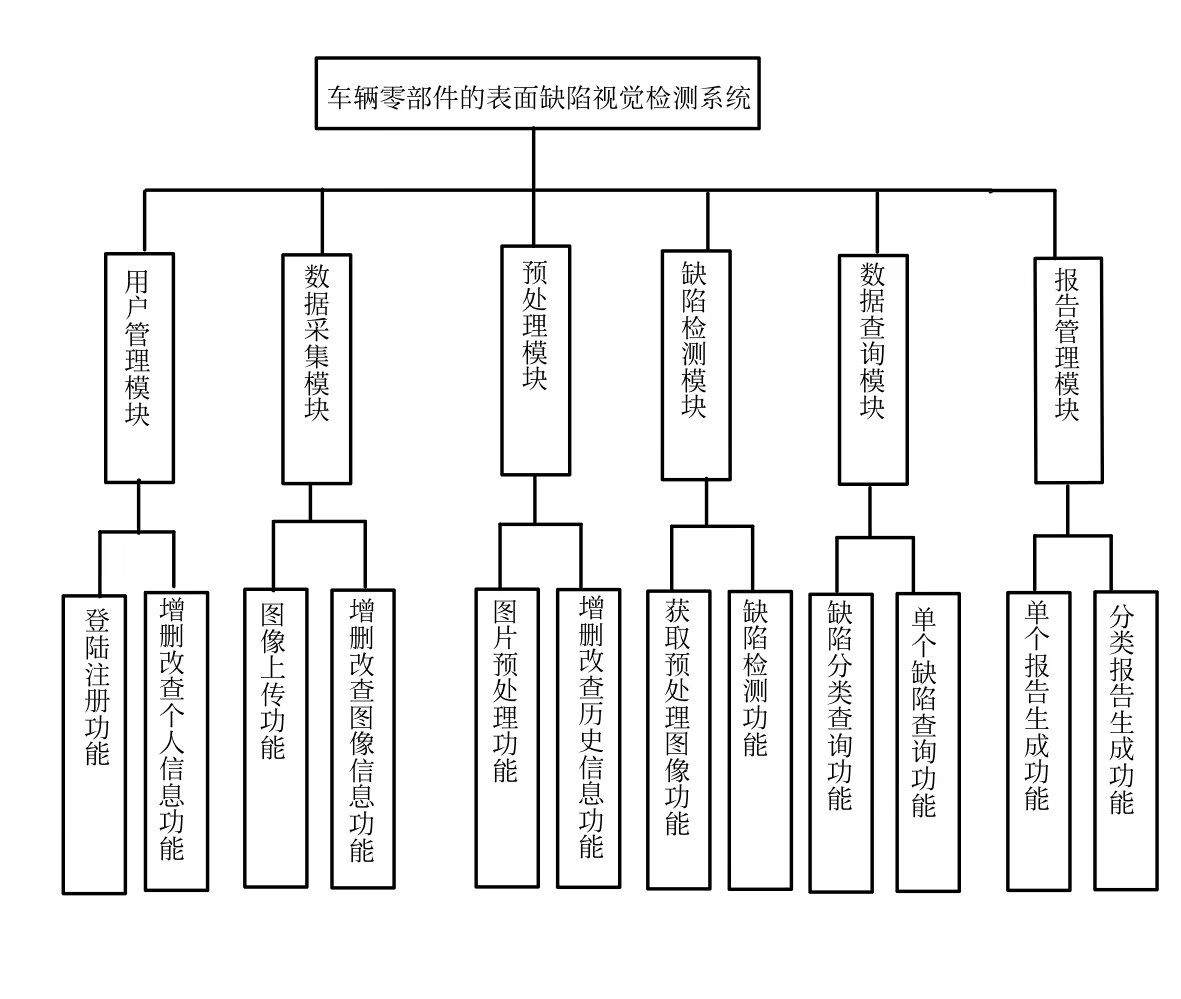


图4-8功能结构图

## 4.3 功能需求与程序的关系

系统的功能需求与程序的关系表如下表所示：

表4-1功能需求与程序的关系表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | 查询 | 新增 | 编辑 | 删除 | 登录 | 注册 |
| 登录 |  |  |  |  | √ |  |
| 注册 |  |  |  |  |  | √ |
| 个人信息 | √ | √ | √ | √ |  |  |
| 图像信息 | √ | √ | √ | √ |  |  |
| 预处理信息 | √ | √ | √ | √ |  |  |
| 缺陷信息 | √ | √ | √ | √ |  |  |
| 报告信息 | √ | √ | √ | √ |  |  |

## 4.4 运行设计

### 4.4.1 运行模块组合

（1）用户使用使用用户名、密码、第三方渠道登录账号，对个人用户名个人邮箱、密码等信息进行查询修改都会运行用户模块。

（2）用户和管理员对图像和预处理图像信息进行增加上传、修改、删除都会运行数据采集模块。

（3）用户和管理员对缺陷信息和报告信息进行增加上传、修改、删除都会运行检测管理模块。

（4）管理员和用户对图像信息、预处理图像信息、缺陷信息、报告信息进行查看和删除会运行历史信息管理模块。

### 4.4.2 运行控制

表4-2运行控制表

|  |  |
| --- | --- |
| 运行控制 | 控制方法 |
| 用户注册 | 用户填写自己的用户名和设置的密码，才能够建立新账户 |
| 用户登录 | 用户填用户名和用户密码，必须填写正确的用户名和密码才能进行登录 |
| 修改个人信息 | 用户必须已经处于已经登录状态 |
| 数据采集信息 | 用户已登录；可以选择图像信息上传 |
| 预处理图像信息 | 用户已登录且已经完成了数据采集工作，才能进行图像的预处理 |
| 查看个人主页 | 用户已登录；查询自己的个人主页可以对个人信息管理 |
| 缺陷信息 | 用户已登录且已经完成了数据采集和预处理，才能进行缺陷检测 |
| 报告信息 | 用户已登录且已经完成了数据采集、预处理和缺陷检测，才能进行报告信息的生成 |

## 4.5 概念结构设计 （看数据库设计书）

### 4.5.1 实体属性图

实体属性图如下图所示：

|  |  |
| --- | --- |
| 3a3e8542eb2cd55b57cde881ca271707  图4-8用户实体属性图 | 636414f02d0f61935aacf9798da51392  图4-9图像实体属性图 |

|  |  |
| --- | --- |
| cb8aba4c92fd71ed4dd04dbcfde88450  图4-10实预处理体属性图 | 50c5a613035e4aa0f0d50fcd46157d88  图4-11报告实体属性图 |

|  |  |
| --- | --- |
| 9a774b60189693d3dd6938408e58f4df  图4-12缺陷实体属性图 | 7c04cc646a6adca8f07c8f70be3e5f43  图4-13检测结果实体属性图 |

总体E-R图：



图4-14总体E-R图

### 4.5.2 Mysql数据库表

Mysql数据库表如下表所示：

表4-3用户表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 名称 | 别名 | 类型 | 长度 | 约束 |
| user\_id | 用户编号 | int |  | 主键，非空 |
| username | 用户名 | varchar | 50 | 非空 |
| password | 密码 | varchar | 255 | 非空 |
| email | 邮箱 | varchar | 100 | 非空 |
| role | 角色 | varchar | 50 | 非空 |
| created\_at | 创建时间 | timestamp |  | 默认为空 |
| updated\_at | 更新时间 | timestamp |  | 默认为空 |

表4-4图像表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 名称 | 别名 | 类型 | 长度 | 约束 |
| image\_id | 图像编号 | int |  | 主键，非空 |
| user\_id | 用户编号 | int |  | 默认为空 |
| image\_path | 图像路径 | varchar | 255 | 默认为空 |
| uploaded\_at | 创建时间 | timestamp |  | 默认为空 |
| image\_format | 图像格式 | varchar | 10 | 默认为空 |
| preprocessed | 状态 | tinyint | 1 | 默认为空 |

表4-5缺陷表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 名称 | 别名 | 类型 | 长度 | 约束 |
| defect\_id | 缺陷编号 | int |  | 主键，非空 |
| image\_id | 图像编号 | int |  | 默认为空 |
| defect\_type | 缺陷类型 | varchar | 50 | 默认为空 |
| defect\_location | 缺陷位置 | varchar | 100 | 默认为空 |
| defect\_confidence | 置信度 | varchar | 100 | 默认为空 |
| defect\_size | 缺陷大小 | varchar | 50 | 默认为空 |
| detected\_at | 创建时间 | timestamp |  | 默认为空 |

表4-6检测结果表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 名称 | 别名 | 类型 | 长度 | 约束 |
| result\_id | 结果编号 | int |  | 主键，非空 |
| image\_id | 图像编号 | int |  | 默认为空 |
| result\_path | 结果路径 | varchar | 255 | 默认为空 |
| detection\_time | 创建时间 | datetime |  | 默认为空 |

表4-7预处理表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 名称 | 别名 | 类型 | 长度 | 约束 |
| preprocessed\_image\_id | 预处理编号 | int |  | 主键，非空 |
| image\_id | 图像编号 | int |  | 默认为空 |
| preprocessed\_path | 预处理路径 | varchar | 255 | 默认为空 |
| preprocessed\_at | 创建时间 | timestamp |  | 默认为空 |

表4-8报告表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 名称 | 别名 | 类型 | 长度 | 约束 |
| report\_id | 报告编号 | int |  | 主键，非空 |
| user\_id | 用户编号 | int |  | 默认为空 |
| report\_name | 报告名 | varchar | 100 | 非空 |
| generated\_at | 创建时间 | timestamp |  | 非空 |
| report\_path | 报告路径 | varchar | 255 | 默认为空 |
| report\_format | 报告格式 | varchar | 10 | 默认为空 |

### 4.5.3 概念模型和物理模型

概念模型（CDM）如下图所示：

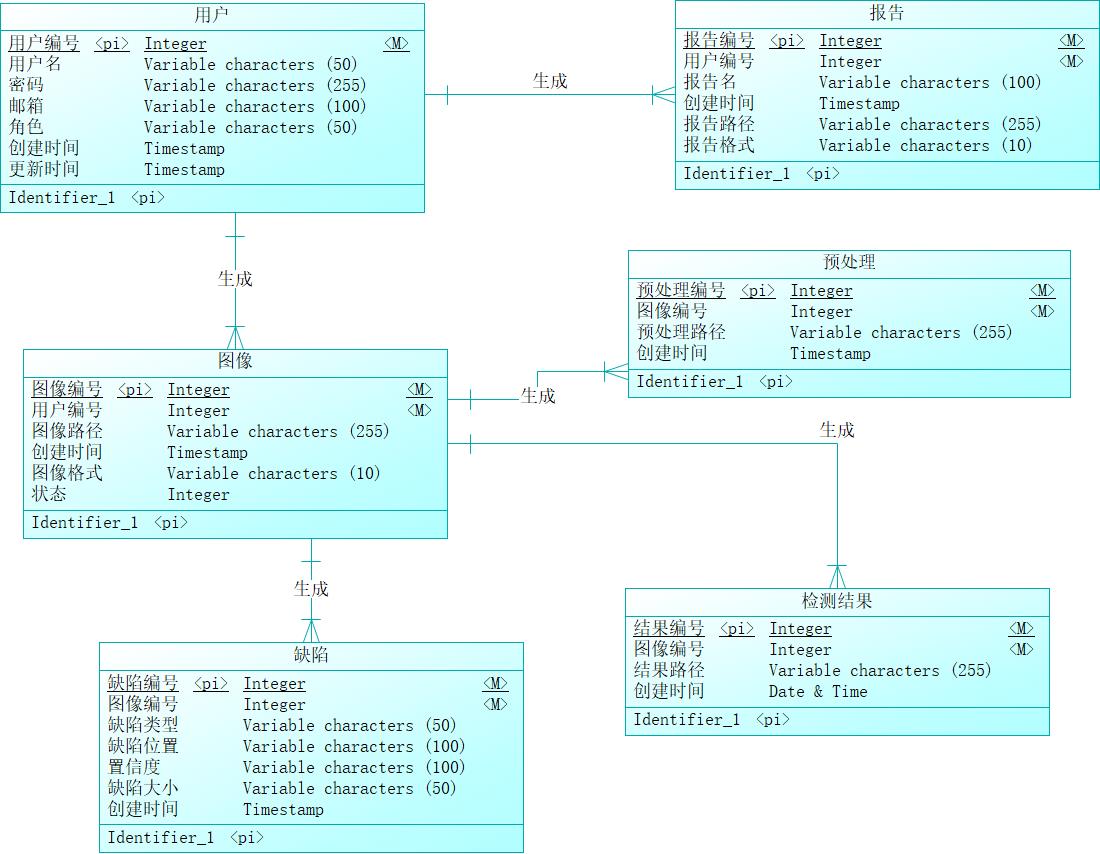


图4-15 CDM图

物理模型（PDM）如下图所示：



图4-16 PDM图

## 4.6 系统出错处理设计

### 4.6.1 出错信息

（1）用户登录失败：提示用户账号、密码可能输入错误，无法进入首页。

（2）获取信息失败：提示检查网络连接，获取数据失败。

### 4.6.2 补救措施

（1）后备：使用附加存储设备备份数据[23]。备份频率为每日一次。需手动备份。

（2）恢复及再启动：如果数据造成丢失，可使用备份数据还原。

**5 详细设计**

## 5.1 程序的系统结构

对于程序的系统结构从底至上分别为运行环境、持久层、基础应用模块层、业务层、接口层、应用层

## 5.2 用户管理模块

### 5.2.1 需求描述

对于车辆零部件的表面缺陷视觉检测系统管理系统来说，主要是实现用户的登录注册和个人信息的修改。

用户：

1. 注册登录用户；
2. 修改个人信息和密码；
3. 退出登录；

### 5.2.2 功能拆分

1. 用户可以看见自己的个人信息的情况
2. 用户可以修改个人密码

### 5.2.3 接口设计

用户管理模块核心接口设计如表5-1至表5-6所示：

表5-1 登录接口设计表

|  |  |
| --- | --- |
| 描述 | 该接口用于登录 |
| 地址 | /auth/login |
| 请求方式 | POST |

续表5-1 登录接口设计表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 入参 | 参数 | 名称 | 类型 | 说明 |
| username | 用户名 | String | 必填 |
| password | 密码 | String | 必填 |
| 出参 | 参数 | 名称 | 类型 | 说明 |
| code | 响应码 | number | 0-成功,1-失败 |
| msg | 提示信息 | String |  |
| data | 返回数据 | String |  |

表5-2 注册接口设计表

|  |  |
| --- | --- |
| 描述 | 该接口用于注册 |
| 地址 | /auth/register |
| 请求方式 | POST |

续表5-2 注册接口设计表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 入参 | 参数 | 名称 | 类型 | 说明 |
| username | 用户名 | String | 必填 |
| password | 密码 | String | 必填 |
| email | 邮箱 | String | 必填 |
| 出参 | 参数 | 名称 | 类型 | 说明 |
| code | 响应码 | number | 0-成功,1-失败 |
| msg | 提示信息 | String |  |
| data | 返回数据 | String |  |

表5-3 退出接口设计表

|  |  |
| --- | --- |
| 描述 | 该接口用于退出 |
| 地址 | /auth/lagout |
| 请求方式 | POST |

续表5-3 退出接口设计表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 入参 | 参数 | 名称 | 类型 | 说明 |
| username | 用户名 | String | 必填 |
| password | 密码 | String | 必填 |
| 出参 | 参数 | 名称 | 类型 | 说明 |
| code | 响应码 | number | 0-成功,1-失败 |
| msg | 提示信息 | String |  |
| data | 返回数据 | String |  |

表5-4 获取用户信息接口设计表

|  |  |
| --- | --- |
| 描述 | 该接口用于获取用户信息 |
| 地址 | /api/users/userInfo |
| 请求方式 | GET |

续表5-4 获取用户信息接口设计表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 出参 | 参数 | 名称 | 类型 | 说明 |
| username | 用户名 | number |  |
| password | 密码 | String |  |
| role | 角色 | String |  |
| email | 邮箱 | String |  |

表5-5 更新用户信息接口设计表

|  |  |
| --- | --- |
| 描述 | 该接口用于获取用户信息 |
| 地址 | /api/users/updateInfo |
| 请求方式 | PUT |

续表5-5 更新用户信息接口设计表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 入参 | 参数 | 名称 | 类型 | 说明 |
| username | 用户名 | String | 必填，默认 |
| role | 角色 | String | 必填，默认 |
| email | 邮箱 | String | 必填 |
| 出参 | 参数 | 名称 | 类型 | 说明 |
| code | 响应码 | number | 0-成功,1-失败 |
| msg | 提示信息 | String |  |
| data | 返回数据 | String |  |

表5-6 重置密码接口设计表

|  |  |
| --- | --- |
| 描述 | 该接口用于获取用户信息 |
| 地址 | /api/users/${id}/password |
| 请求方式 | PUT |

续表5-6 重置密码接口设计表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 入参 | 参数 | 名称 | 类型 | 说明 |
| username | 用户名 | String | 必填，默认 |
| password | 密码 | String | 必填 |
| newpassword | 新密码 | String | 必填 |
| repassword | 确认密码 | String | 必填 |
| 出参 | 参数 | 名称 | 类型 | 说明 |
| code | 响应码 | number | 0-成功,1-失败 |
| msg | 提示信息 | String |  |
| data | 返回数据 | String |  |

### 5.2.4 页面设计

5.2.4.1 页面展示

登录界面如5-1图所示：



图5-1 登录界面

注册界面如图5-2所示：



图5-2 注册界面

个人信息界面如图5-3所示

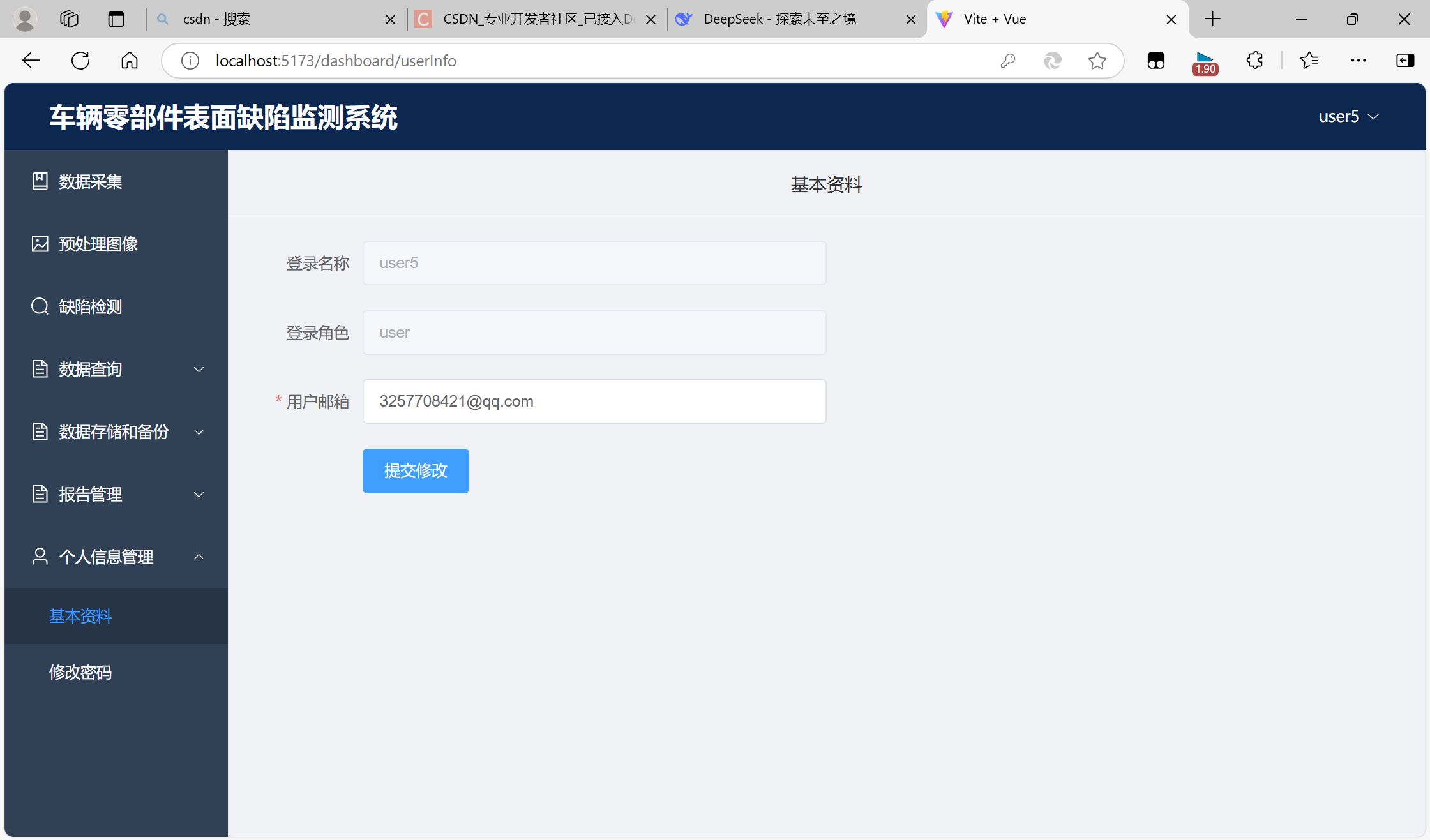


图5-3 个人信息界面

重置密码界面如图5-4所示



图5-4 重置密码界面

5.2.4.2 处理逻辑

1.输入已注册的用户名和密码，如果未注册，显示用户不存在。

2.注册界面邮箱必须是正确格式，注册成功后进行登录。

3.登录成功进入注册界面，点击基本资料，可以修改自己的个人信息

4.点击修改密码，可以实现密码重置。

## 5.3 数据采集管理模块设计

5.3.1 需求描述

对于本系统来说，最主要的就是进行数据采集，将图像上传到我们的系统进行检测和预处理。

### 5.3.2 功能拆分

1.上传图像信息

2.图像信息浏览

### 5.3.3 接口设计

数据检测模块核心接口设计如表5-7至表5-10所示。

表5-7上传图像信息接口设计表

|  |  |
| --- | --- |
| 描述 | 上传图像信息 |
| 地址 | /images |
| 请求方式 | POST |

续表5-7上传图像信息接口设计表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 入参 | 参数 | 名称 | 类型 | 说明 |
| username | 用户名 | String | 用户名 |
| user\_id | 用户编号 | Integer | 用户编号 |
| image\_path | 图像路径 | String | 图像路径 |
| image\_format | 图像格式 | String | 图像格式 |
| 出参 | 参数 | 名称 | 类型 | 说明 |
| code | 响应码 | number | 0-成功,1-失败 |
| msg | 提示信息 | String |  |
| data | 返回数据 | String |  |

表5-8 获取图像信息接口设计表

|  |  |
| --- | --- |
| 描述 | 获取图像信息 |
| 地址 | /images |
| 请求方式 | GET |

续表5-8获取图像信息接口设计表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 入参 | 参数 | 名称 | 类型 | 说明 |
| pageNum | 行数 | Integer | 行数 |
| pageSize | 页数 | Integer | 页数 |
| 出参 | 参数 | 名称 | 类型 | 说明 |
| user\_id | 用户编号 | Integer |  |
| preprocessed | 状态 | String |  |
| imagePath | 图像路径 | String |  |
| uploadAt | 创建时间 | timestamp |  |

### 5.3.4 页面设计

5.3.4.1 页面展示

上传图像页面展示



图5-5上传界面

获取列表界面和删除展示

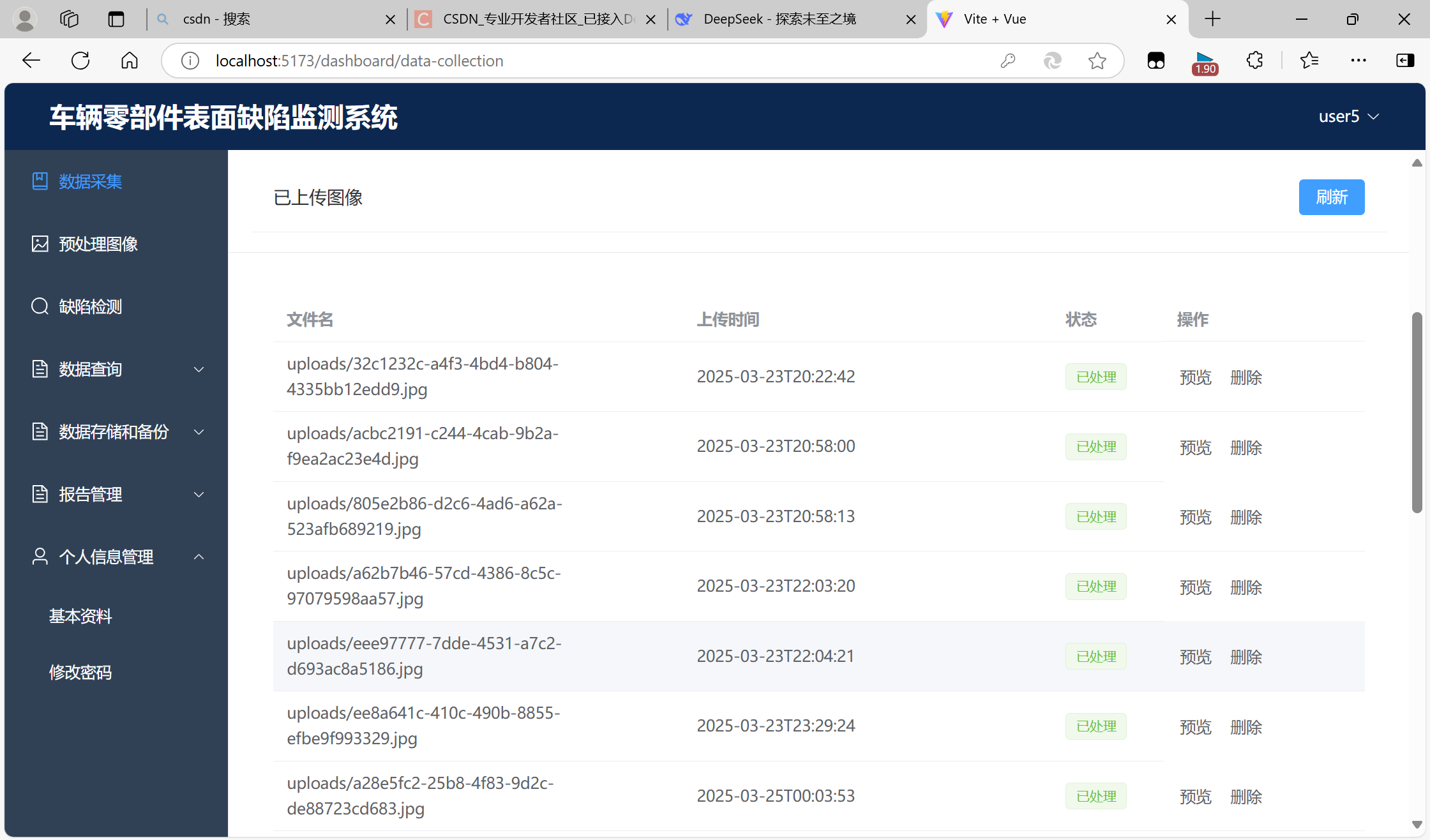


图5-6获取列表界面

预览按钮界面展示

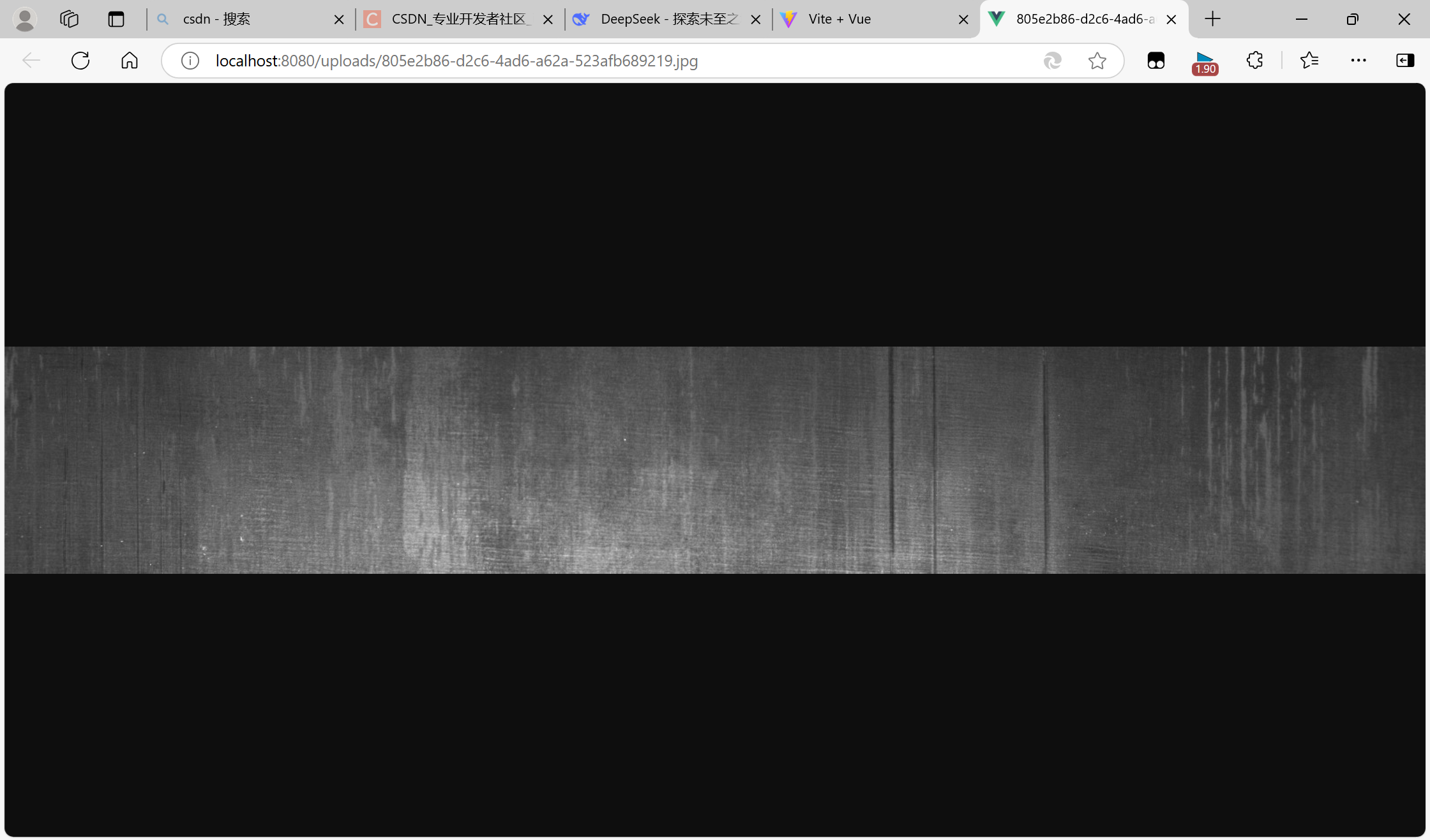


图5-7预览按钮界面

删除按钮界面展示

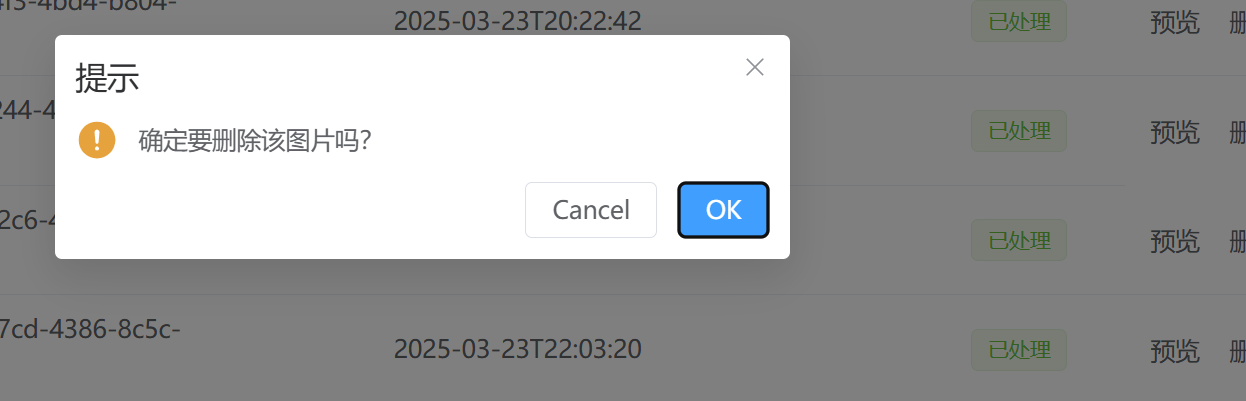


图5-8删除按钮界面

5.3.5.1 处理逻辑

用户上传图像文件，可以单个上传，也可以多个上传，在预处理界面，处理上传的图像，为我们之后的缺陷检测模块做好准备。

## 5.3 预处理模块设计

5.3.1 需求描述

对于本系统来说，进行必要功能就是预处理图像，通过将上传的图像进行预处理。

### 5.3.2 功能拆分

1.图像预处理

2.获取预处理图像列表

3.预览预处理图像

4.删除预处理图像

### 5.3.3 接口设计

预处理管理模块核心接口设计如表5-9至表5-10所示。

表5-9 获取预处理图像信息接口设计表

|  |  |
| --- | --- |
| 描述 | 获取预处理图像信息 |
| 地址 | /images/${id}/getpreprocessedImages |
| 请求方式 | GET |

续表5-9获取预处理图像信息接口设计表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 入参 | 参数 | 名称 | 类型 | 说明 |
| pageNum | 行数 | Integer | 行数 |
| pageSize | 页数 | Integer | 页数 |
| 出参 | 参数 | 名称 | 类型 | 说明 |
| preprocessed\_image\_id | 预处理编号 | Integer |  |
| preprocessed\_path | 预处理路径 | String |  |
| preprocessed\_at | 预处理时间 | timestamp |  |
| image\_id | 图像编号 | Integer |  |

表5-10 预处理图像信息接口设计表

|  |  |
| --- | --- |
| 描述 | 预处理图像信息 |
| 地址 | /images/${id}/preprocess |
| 请求方式 | POST |

续表5-10预处理图像信息接口设计表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 入参 | 参数 | 名称 | 类型 | 说明 |
| imageId | 图像编号 | Integer |  |
| 出参 | 参数 | 名称 | 类型 | 说明 |
| preprocessed\_path | 预处理路径 | String |  |

### 5.3.4 页面设计

5.3.4.1 页面展示

预处理界面展示

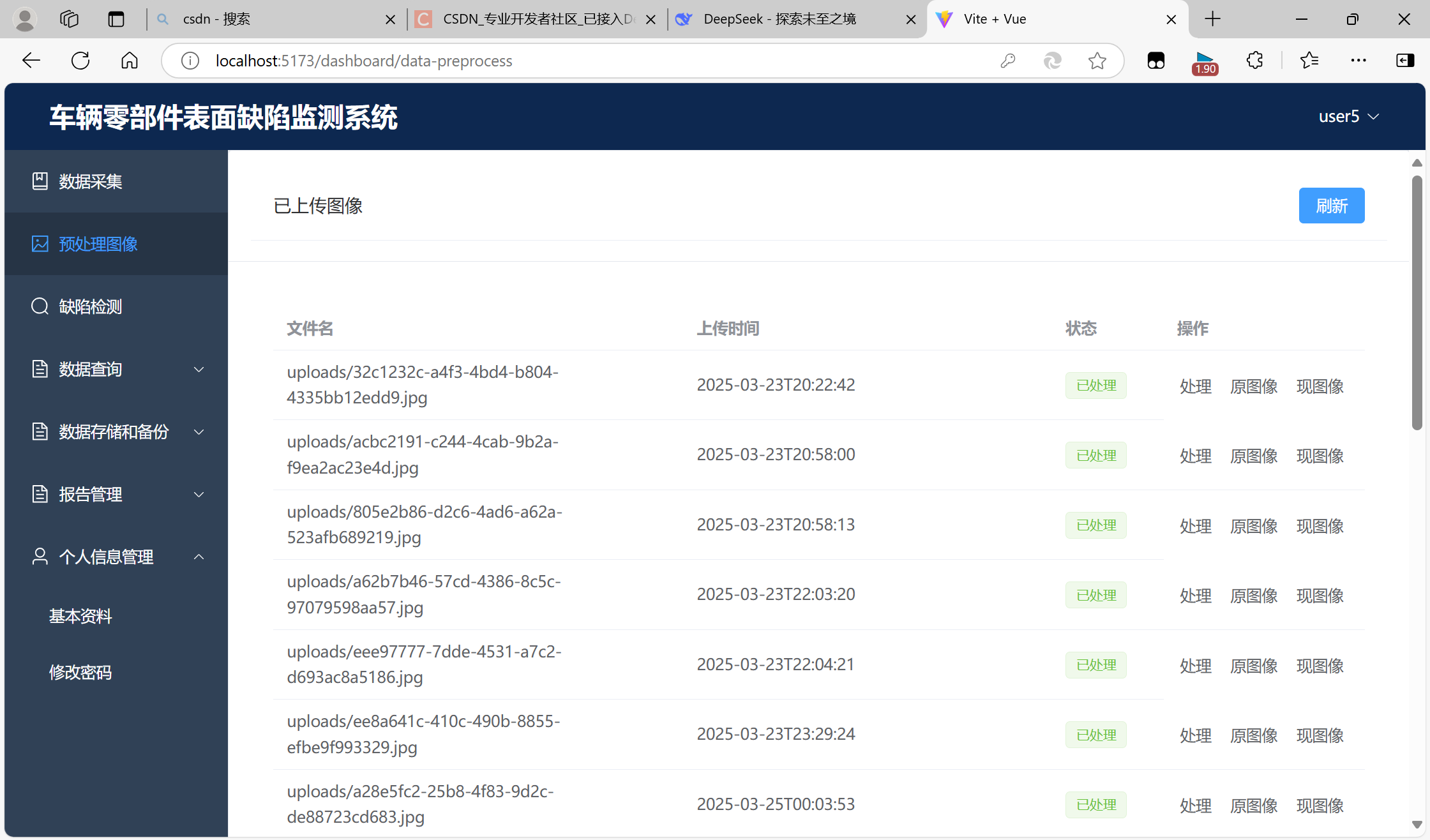


图5-9预处理界面

5.3.5.1 处理逻辑

将上传的图像进行预处理，实现图像更加清晰，方便后续的缺陷检测。

## 5.4 缺陷检测管理模块设计

5.4.1 需求描述

对于本系统来说，最主要的就是在进行数据采集后，将预处理后的图像进行缺陷检测，将结果检测出来并展示出来。

### 5.4.2 功能拆分

1.获取预处理图像

2.开始检测

3.检测结果信息浏览

### 5.4.3 接口设计

缺陷检测管理模块核心接口设计如表5-11至表5-10所示。

表5-11缺陷检测接口设计表

|  |  |
| --- | --- |
| 描述 | 缺陷检测信息 |
| 地址 | /images/${id}/detect |
| 请求方式 | POST |

续表5-11缺陷检测接口设计表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 入参 | 参数 | 名称 | 类型 | 说明 |
| imageId | 图像编号 | Integer |  |
| 出参 | 参数 | 名称 | 类型 | 说明 |
| result\_path | 结果路径 | String |  |

表5-12 获取缺陷检测接口设计表

|  |  |
| --- | --- |
| 描述 | 获取缺陷检测 |
| 地址 | /images/${imageId}/defects |
| 请求方式 | GET |

续表5-12获取缺陷检测接口设计表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 入参 | 参数 | 名称 | 类型 | 说明 |
| imageId | 图像编号 | Integer |  |
| 出参 | 参数 | 名称 | 类型 | 说明 |
| result\_id | 结果编号 | Integer |  |
| result\_path | 结果路径 | String |  |
| detection\_time | 创建时间 | timestamp |  |

### 5.4.4 页面设计

5.4.4.1 页面展示

缺陷检测页面展示

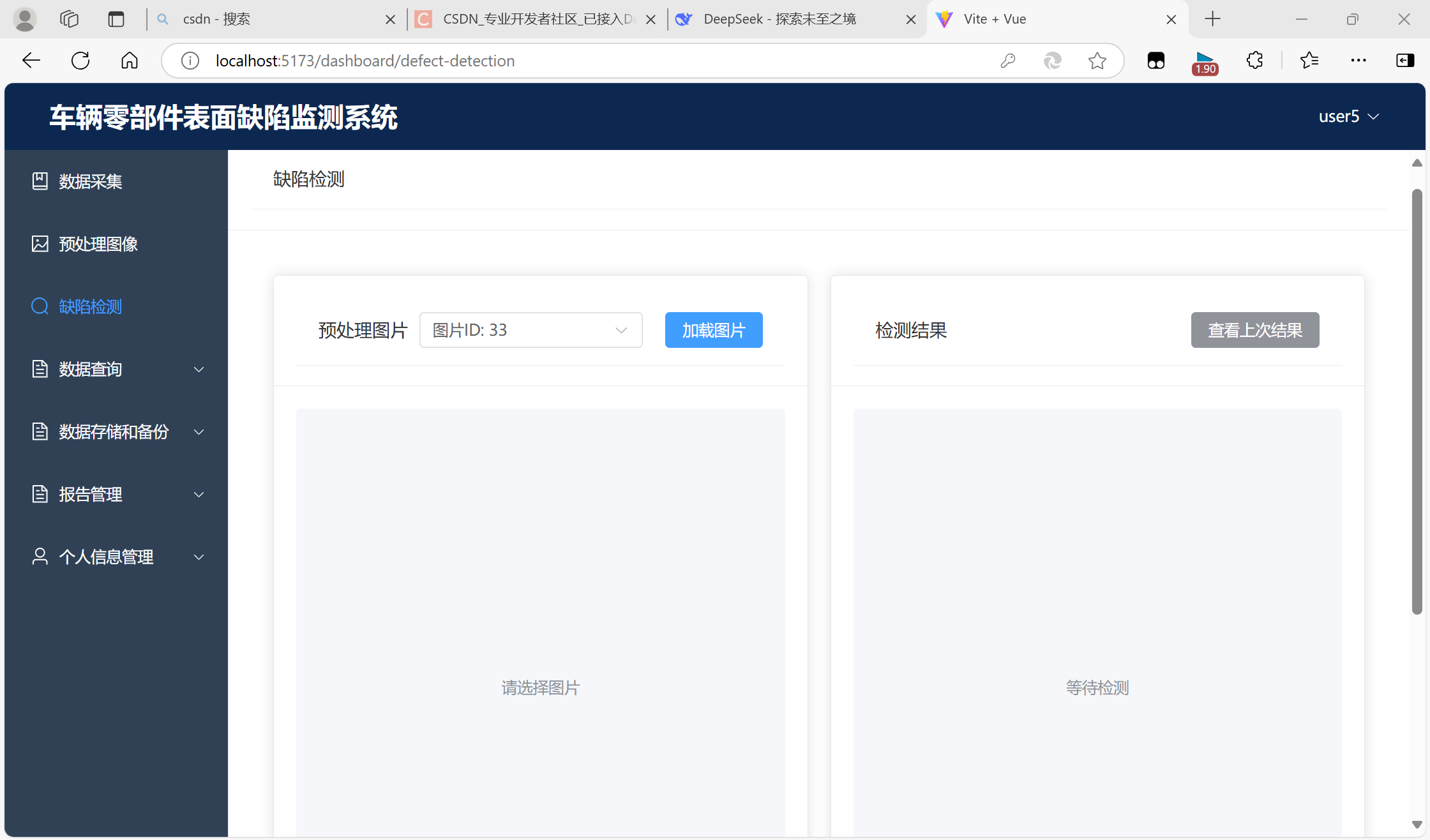


图5-10缺陷检测界面

5.4.5.2 处理逻辑

用户通过预处理后的图像，在缺陷检测界面进行缺陷检测，调用后缺陷检测功能，最后返回一张检测后的图像，和缺陷信息。

## 5.5 数据查询管理模块设计

5.5.1 需求描述

在我们完成任务后，需要对我们对我们之前的操作进行查看，可以通过数据查询实现

### 5.5.2 功能拆分

1.缺陷分类查询

2.缺陷详细查询

### 5.5.4 页面设计

5.5.4.1 页面展示

缺陷检测页面展示

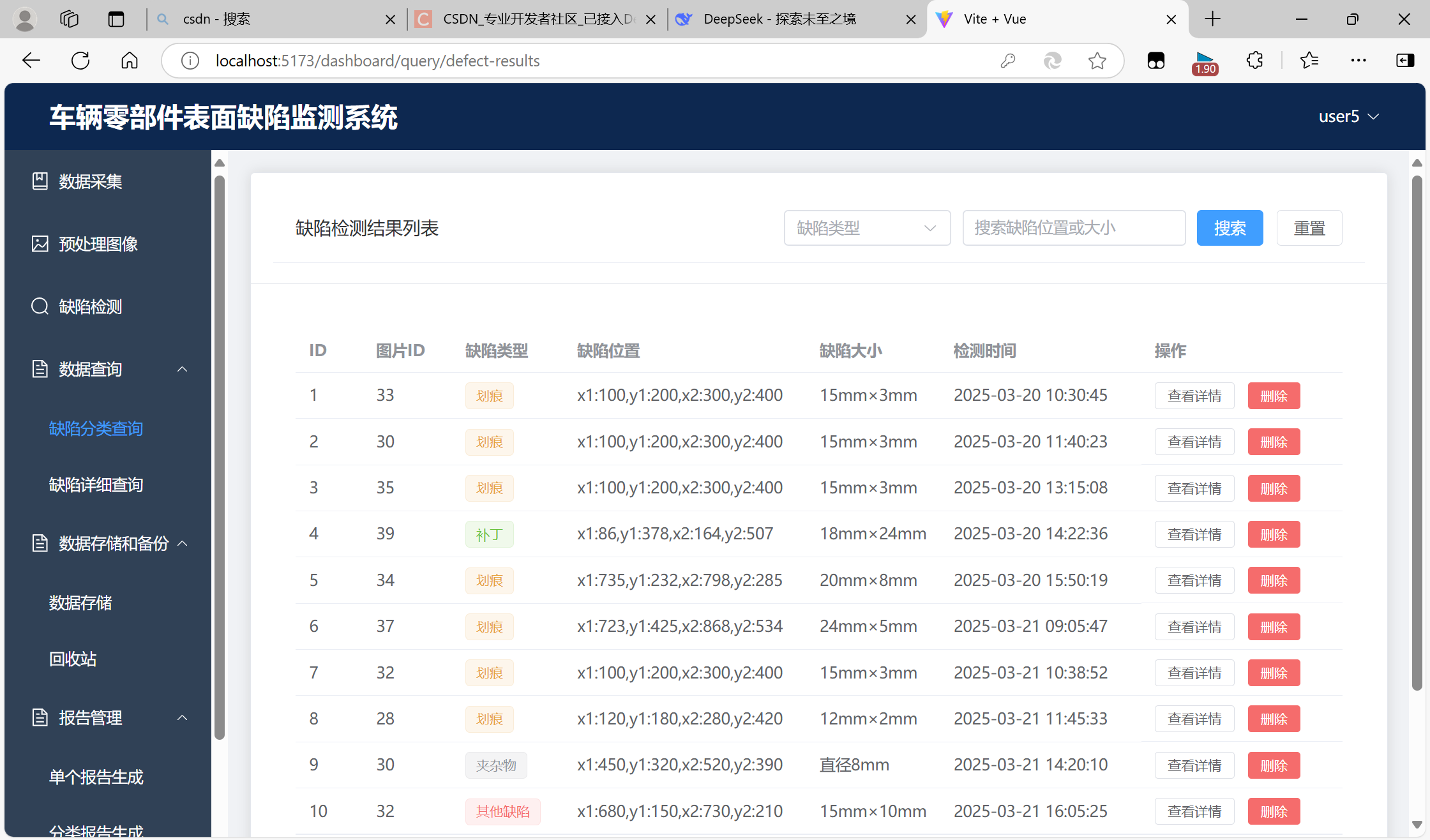


图5-8缺陷检测界面



图5-8缺陷检测界面

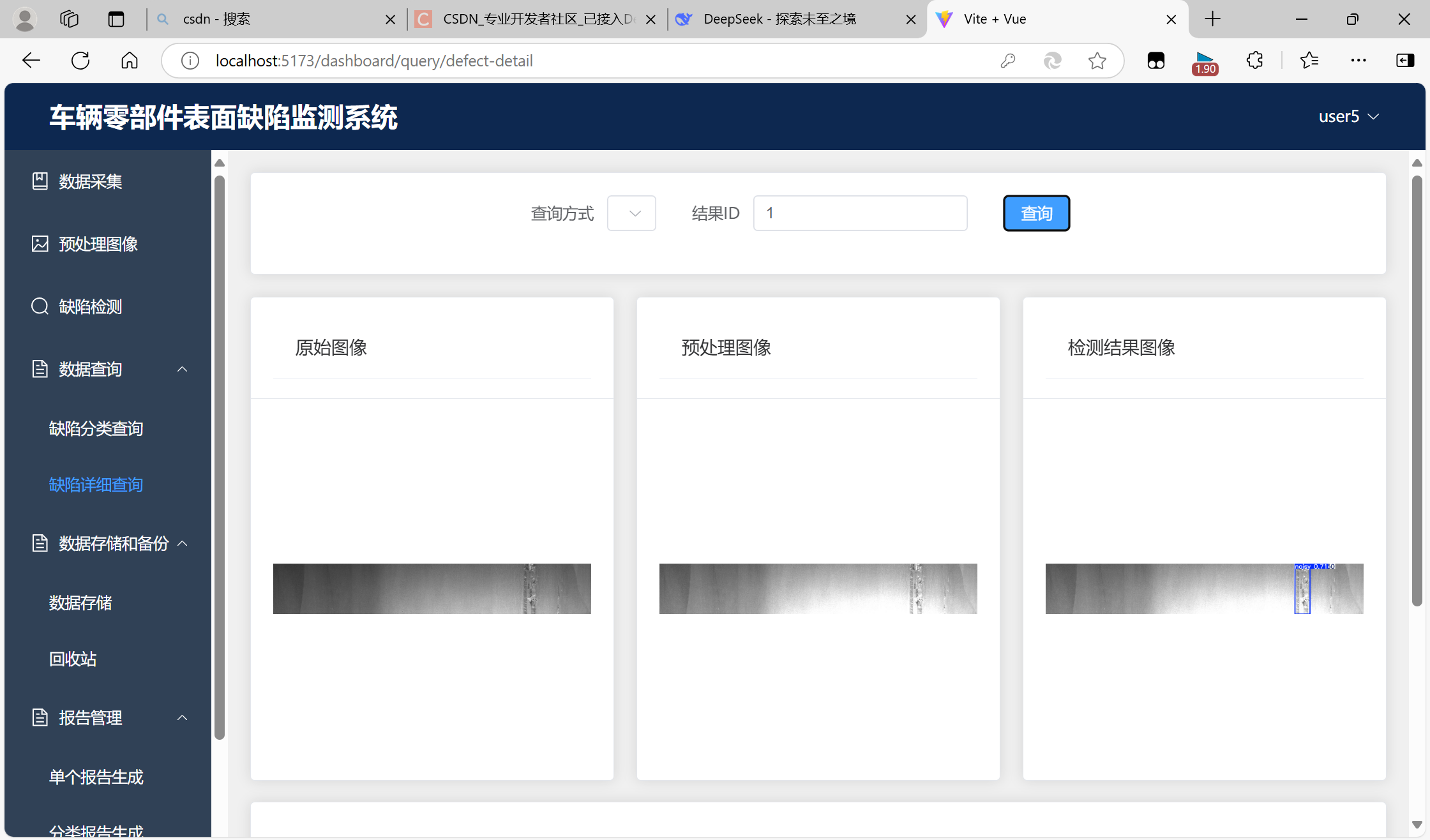


图5-8缺陷检测界面

5.5.5.2 处理逻辑

通过获取之前的处理结果，然后实现查询功能。

## 5.5 数据存储和备份管理模块设计

5.5.1 需求描述

在我们完成所有工作后，需要对我们的数据进行备份和存储。

### 5.5.2 功能拆分

1.数据存储

2.回收站

### 5.5.4 页面设计

5.5.4.1 页面展示

缺陷检测页面展示

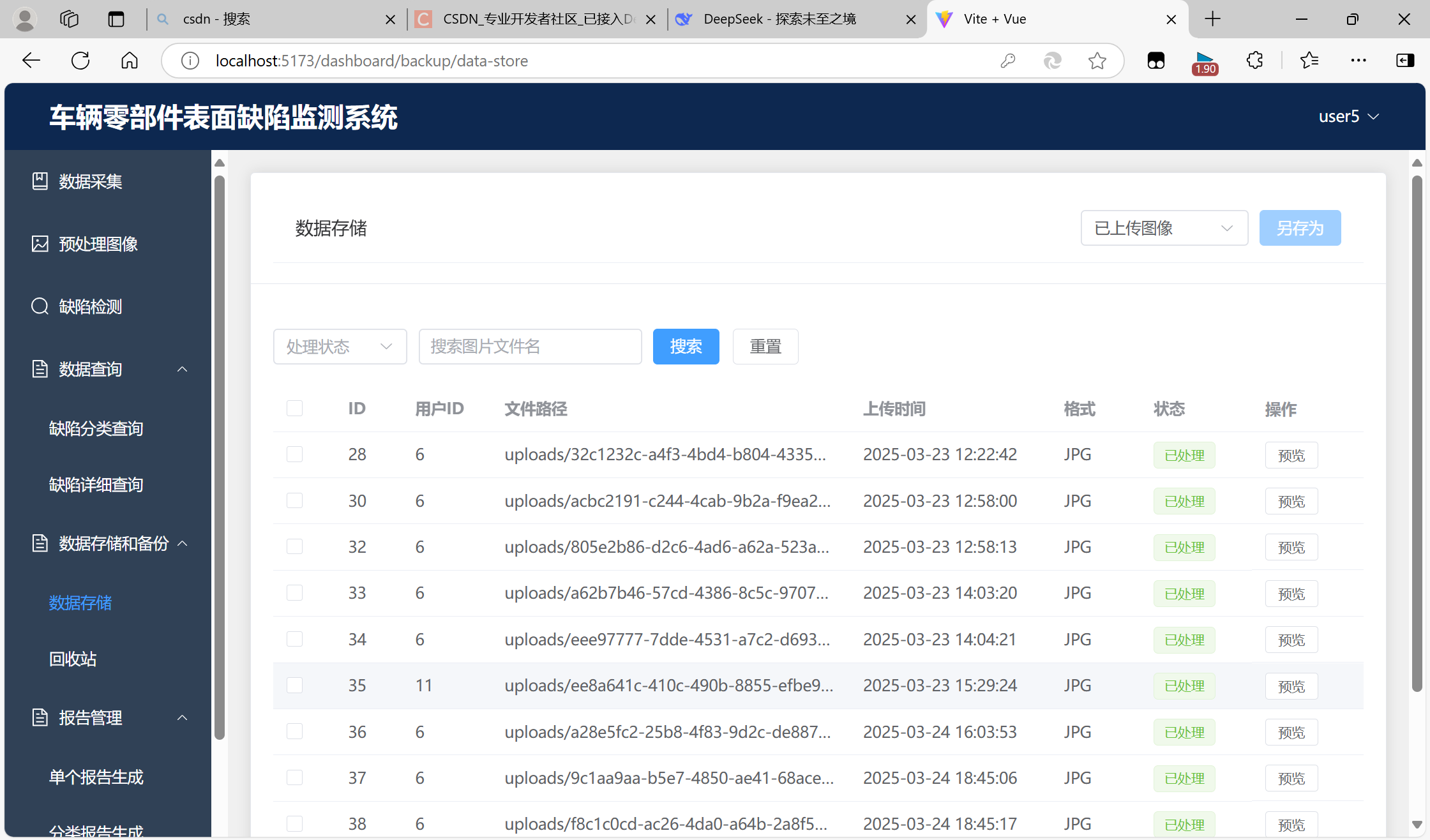


图5-8数据存储界面

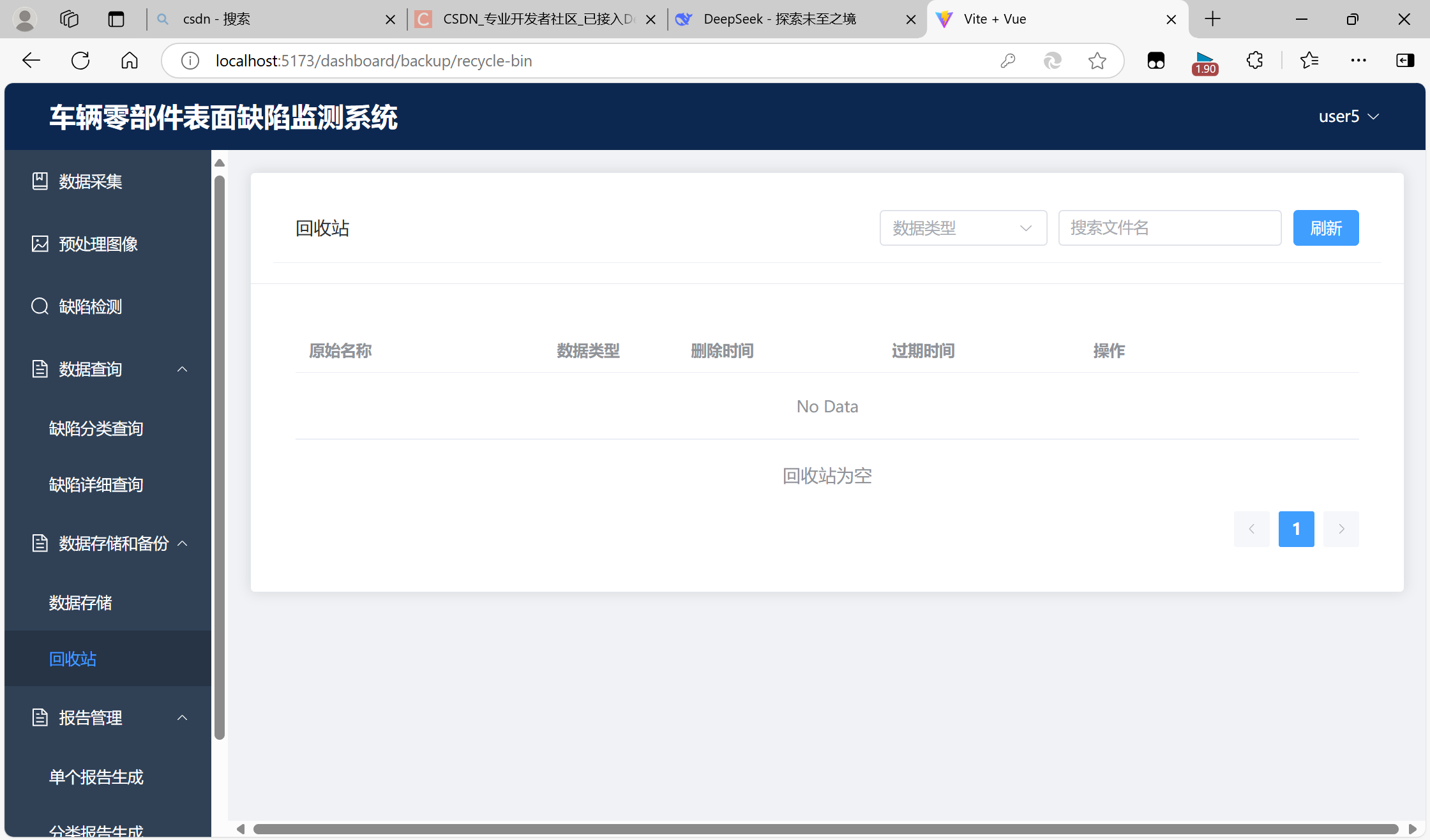


图5-8回收站界面

5.5.5.2 处理逻辑

在我们进行之前的操作时，将数据随时存储到我们的后台，通过数据存储与备份模块进行查询。

## 5.4 报告管理模块设计

5.4.1 需求描述

在我们进行完工作后需要将我们检测信息全部导出可视化，报告管理模块即可实现我们的需求

### 5.4.2 功能拆分

1.获取报告列表

2.生成报告

### 5.4.3 接口设计

报告管理模块核心接口设计如表5-13至表5-15所示。

表5-13 生成报告接口设计表

|  |  |
| --- | --- |
| 描述 | 生成报告 |
| 地址 | /reports |
| 请求方式 | POST |

续表5-13生成报告接口设计表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 入参 | 参数 | 名称 | 类型 | 说明 |
| reportName | 报告名 | String |  |
| reportFormat | 报告格式 | String |  |
| defectIds | 缺陷编号 | Integer |  |
| 出参 | 参数 | 名称 | 类型 | 说明 |
| reportName | 报告名 | String |  |
| reportFormat | 报告格式 | String |  |
| reportPath | 报告路径 | String |  |
| generated\_at | 创建时间 | timestamp |  |
| user\_id | 用户编号 | Integer |  |

表5-14 删除报告接口设计表

|  |  |
| --- | --- |
| 描述 | 删除报告 |
| 地址 | /reports/reportsList |
| 请求方式 | get |

续表5-14删除报告接口设计表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 入参 | 参数 | 名称 | 类型 | 说明 |
| report\_id | 报告编号 | Integer |  |
| 出参 | 参数 | 名称 | 类型 | 说明 |
| code | 响应码 | number | 0-成功,1-失败 |
| msg | 提示信息 | String |  |
| data | 返回数据 | String |  |

表5-15 下载报告接口设计表

|  |  |
| --- | --- |
| 描述 | 下载报告 |
| 地址 | /reports/download |
| 请求方式 | get |

续表5-15删除报告接口设计表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 入参 | 参数 | 名称 | 类型 | 说明 |
| report\_path | 报告路径 | String |  |
| 出参 | 参数 | 名称 | 类型 | 说明 |
| URL | 文件路径 | String |  |

### 5.4.4 页面设计

5.4.4.1 页面展示

报告生成页面展示

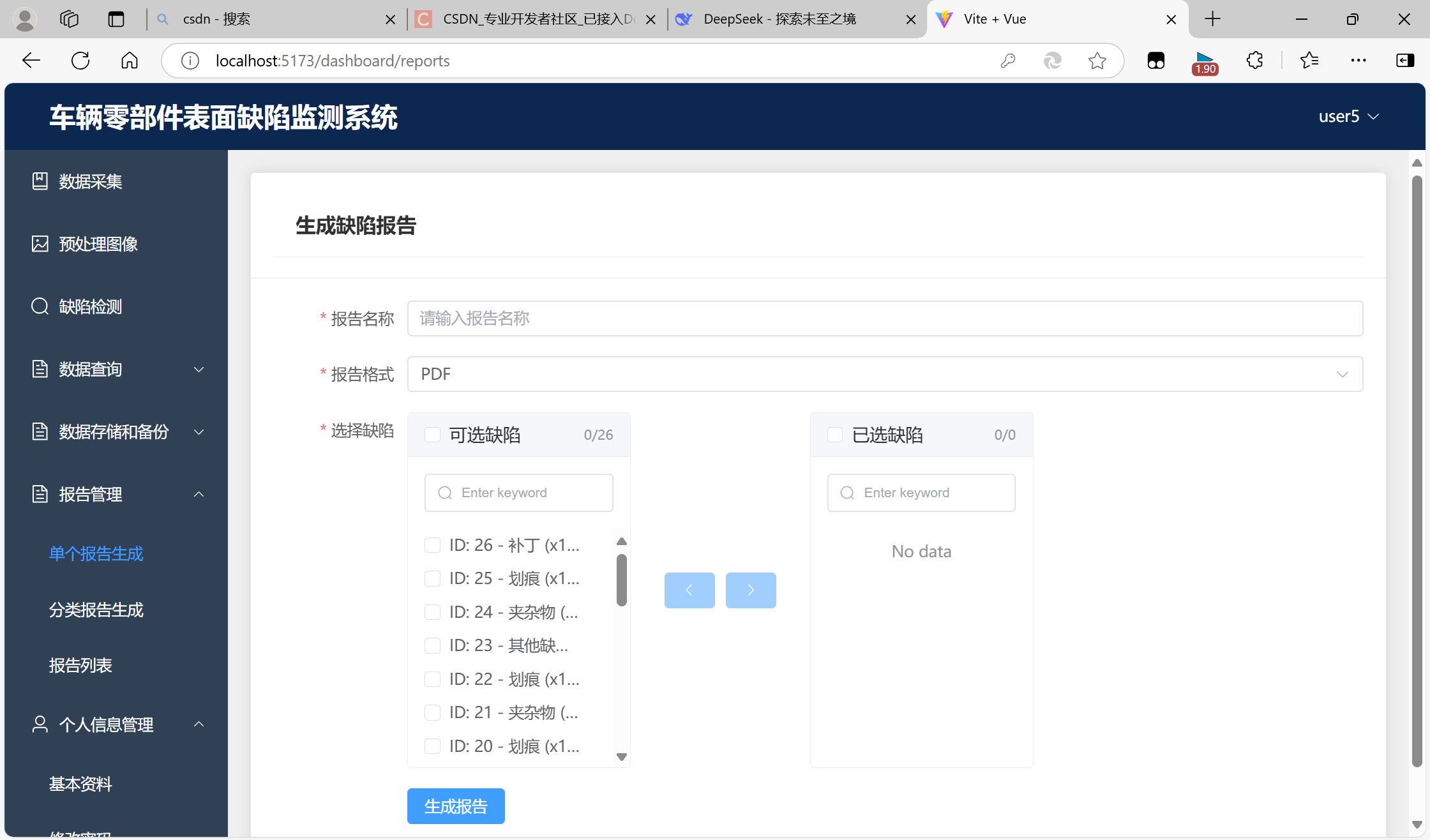


图5-9单个报告生成界面

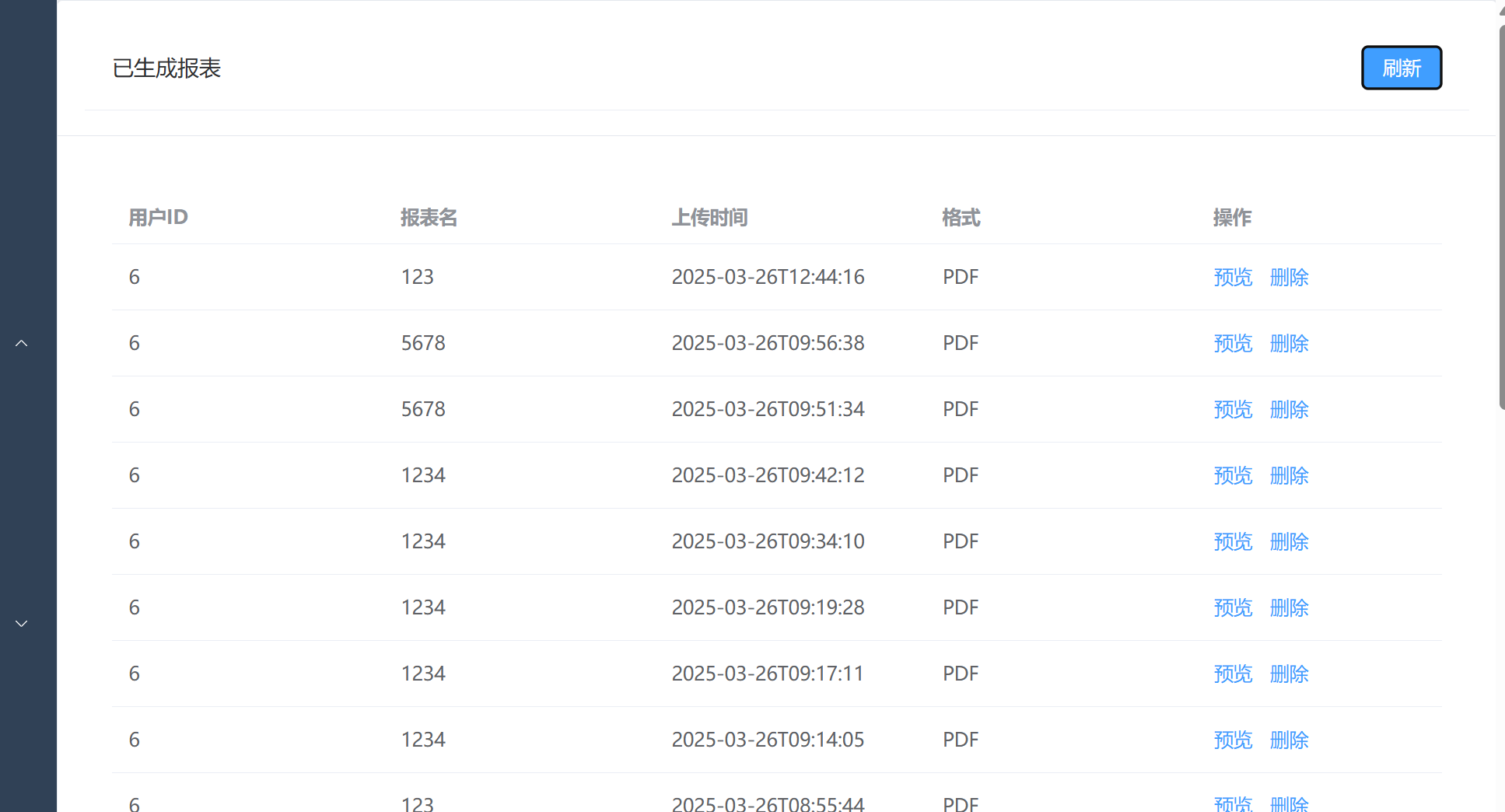


图5-9报告列表界面

5.4.5.2 处理逻辑

将缺陷检测的信息，预处理图像信息，原图像信息，展示到我们的报告中。

**6 测试**

## 6.1 测试系统图

系统的测试将会对用户模块、房间模块、预定模块、系统管理模块的核心功能进行测试，测试系统图如下图所示：

## 6.2 测试的方法

测试采用黑盒测试的等价划分类方法[30]，等价划分类是指某个输入域的子集合。在该子集合中，各个输入数据对于揭露程序中的错误都是等效的，测试代表值就等价于这一类其他值的测试。测试用例包括测试数据，待测试功能，测试结果。并且以下测试除登录模块外都是基于当前用户已经登录的情况下进行测试的，如果用户为登录，该用户的将不能够进入到操作其他模块的页面。

## 6.3 用户模块测试

### 6.3.1 账号密码登录

密码登录的有效等价类和无效等价类如表6-1所示：

表6-1 账号密码登录等价类表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 输入条件 | 有效等价类 | 编号 | 无效等价类 | 编号 |
| 用户名 | 已经注册过的用户名 | E1 | 未注册的用户名 | E3 |
| 密码 | 已经注册过且正确的密码 | E2 | 不正确不匹配的密码 | E4 |

由此等价类表导出的测试用例如下表6-2所示：

表6-2 用户名密码登录测试用例表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编号 | 输入 | | 覆盖等价类 | 预期输出 |
| 用户名 | 密码 |
| U1 | user5 | 1234567 | E1、E2 | 登录成功 |
| U2 | jjyzm | jjyzm | E1、E2 | 登录成功 |
| U3 | zhangsan | zhangsan | E1、E2 | 登录成功 |
| U4 | qaz1 | 1234567 | E2、E3 | 登录失败 |
| U5 | user5 | 123 | E1、E4 | 登录失败 |
| U6 | qaz1 | 123 | E3、E4 | 登陆失败 |

测试用例U1-U4都是正确的登录方式，能够成功登录，它的执行情况如图6-1至6-3所示。

测试用例U5-U7均为不正确的登录方式，均会有相应弹窗提示用户操作的错误，它的执行情况如图6-4所示，并且登录失败，不会进入首页。经过测试，登录的测试用例U1~U7均符合预期。



图6-1用户密码登录测试用例U1-U3登录成功图



图6-3用户密码登录测试用例U4-U6登录失败图

### 6.3.2 个人信息填写

填写个人资料的有效等价类和无效等价类如表6-3至6-4所示：

表6-3个人资料填写等价类表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 输入条件 | 有效等价类 | 编号 | 无效等价类 | 编号 |
| 邮箱 | 正确字符加上@符号加上正确字符加上.再加上正确字符 | E1 | 输入没有@符号 | E2 |
| 输入的字符中包含不符合规定的字符 | E3 |
| 输入没有.符号 | E4 |

由此等价类表导出的测试用例如下表所示：

表6-4验证输入个人信息测试用例表

|  |  |  |  |
| --- | --- | --- | --- |
| 编号 | 输入 | 覆盖等价类 | 预期输出 |
| 邮箱 |
| U1 | 3257708441@qq.com | E1 | 登录成功 |
| U2 | 3257708441qq.com | E2 | 格式错误 |
| U3 | 3257708441(qq.com | E3 | 格式错误 |
| U4 | 3257708441@qqcom | E4 | 格式错误 |

测试用例U1的执行情况如图6-4所示，能够正确输入个人信息且不报错。

测试用例U2~U4均为不正确的个人信息输入方式，均会在输入时报错，经过测试，登录的测试用例U1~U9均符预期。



图6-4填写个人资料成功图



图6-5填写个人资料失败图

### 6.3.3 重置密码功能

表6-5 重置密码功能等价类表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 输入条件 | 有效等价类 | 编号 | 无效等价类 | 编号 |
| 原密码 | 正确的原密码 | E1 | 未填写 | E4 |
| 新密码 | 非空 | E2 | 未填写 | E5 |
| 格式不对 | E6 |
| 再次输入新密码 | 与新密码一致 | E3 | 与新密码不一致 | E7 |
| 未填写 | E8 |
| 格式不对 | E9 |

表6-6 房间预约功能测试用例表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 编号 | 原密码 | 新密码 | 再次输入新密码 | 覆盖等价类 | 预期输出 |
| U1 | 1234567 | 123456 | 123456 | E1、E2、E3 | 修改成功 |
| U2 | 无 | 无 | 无 | E4、E5、E8 | 请先填写密码 |
| U3 | 1234567 | 123 | 123 | E6、E9 | 请填写5-16位密码 |
| U4 | 1234567 | 123456 | 12345 | E7 | 与新密码不一致 |

测试用例U1的执行情况如图6-11所示，能够正确重置密码。

测试用例U2-U4均为错误的修改密码方式，执行情况如图6-7至6-10所示。经过测试，重置密码测试用例U1~U4均符合预期。

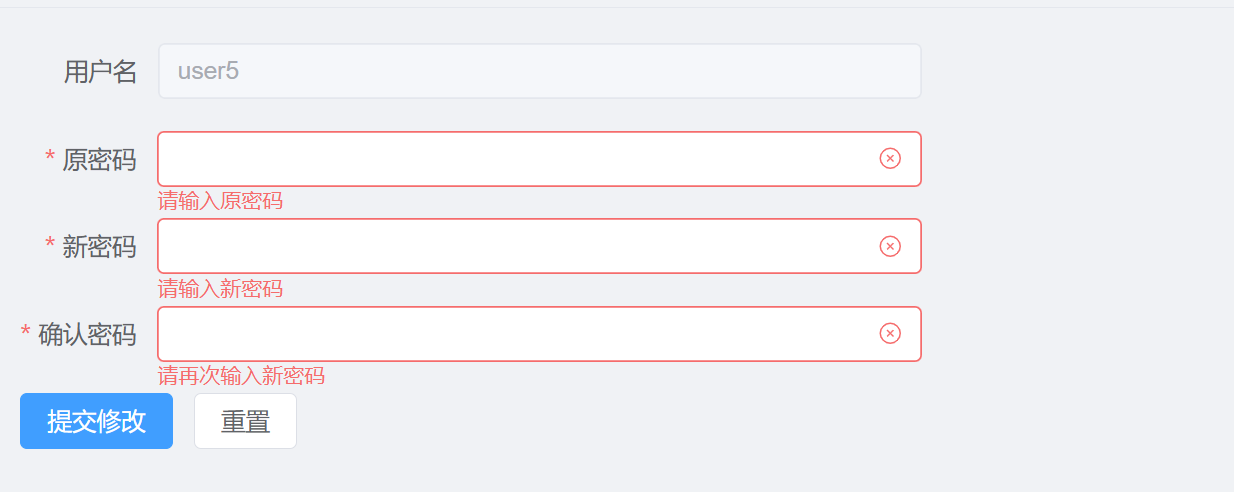


图6-7修改密码失败图（未填写）

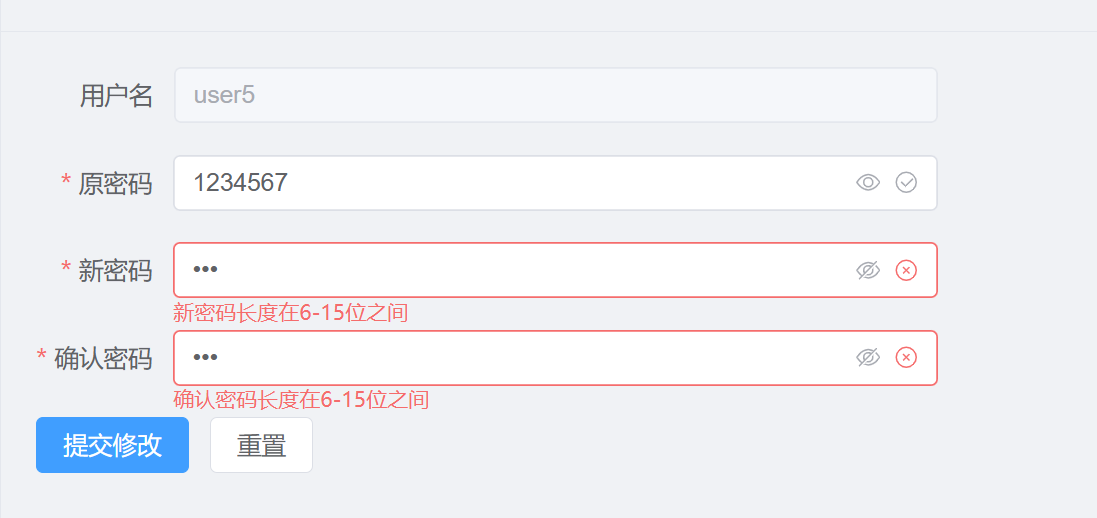


图6-8修改密码失败图（填写格式不对）

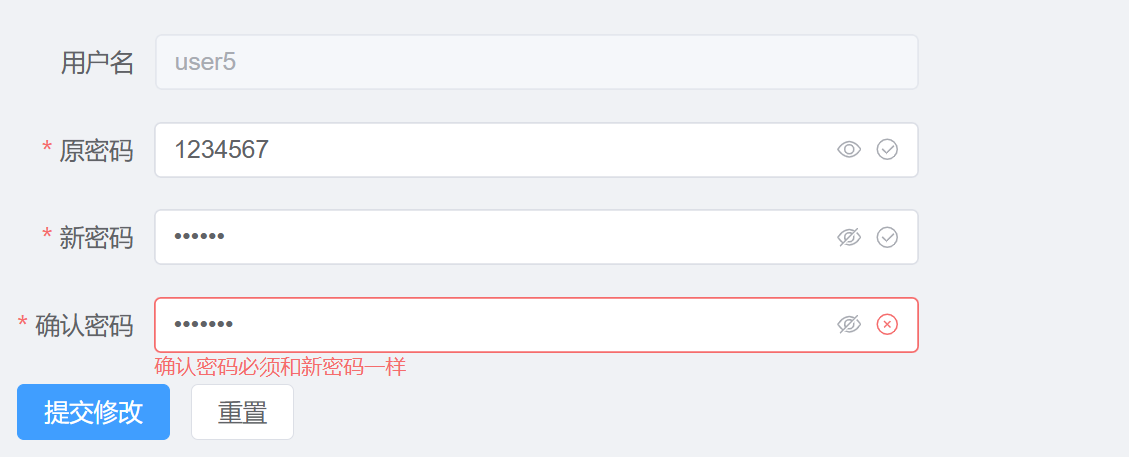


图6-9修改密码失败图（填写格式不对）



图6-10修改密码失败图



图6-11修改密码成功图

### 6.3.4 更改密码功能

表6-7 用户修改密码功能等价类表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 输入条件 | 有效等价类 | 编号 | 无效等价类 | 编号 |
| 用户名 | 未注册过的用户 | E1 | 注册过的用户 | E4 |
| 邮箱 | 正确格式 | E2 | 错误格式 | E5 |
| 新密码 | 正确的新密码 | E3 | 未输入新密码 | E6 |

表6-8 用户修改密码功能测试用例表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 编号 | 用户名 | 邮箱 | 新密码 | 覆盖等价类 | 预期输出 |
| U1 | 未注册过的用户 | 正确格式 | 正确的新密码 | E1、E2、E3 | 注册成功 |
| U2 | 注册过的用户 | 错误格式 | 未输入新密码 | E5、E7、E9 | 注册不成功 |

测试用例U1的执行情况如图6-12所示，用户能够成功注册用户，使用系统。

测试用例U2为错误的预约房间方式，相应提示操作错误执行情况如图6-13至6-14所示。经过测试，注册功能测试用例U1~U2均符合预期。



图6-12注册成功图



图6-13注册不成功图



图6-14注册不成功图

### 6.3.5 上传图像信息

表6-9 用户修改密码功能等价类表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 输入条件 | 有效等价类 | 编号 | 无效等价类 | 编号 |
| 上传图片 | 正确格式 | E1 | 错误格式 | E3 |
| 正确数据量 | E2 | 错误数据量 | E4 |

表6-10 用户上传图片功能测试用例表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编号 | 上传图片 | | 覆盖等价类 | 预期输出 |
| U1 | 正确格式 | 正确数据量 | E1、E2 | 上传成功 |
| U2 | 错误格式 | 错误数据量 | E3、E4 | 上传不成功 |

测试用例U1的执行情况如图6-15所示，用户能够成功上传图片。

测试用例U2为错误的上传图片方式，相应提示操作错误执行情况如图6-16至6-17所示。经过测试，上传图片功能测试用例U1~U2均符合预期。



图6-15 上传成功图



图6-16 上传不成功图

### 6.3.6 缺陷详细查询

表6-11 用户查询缺陷分类功能等价类表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 输入条件 | 有效等价类 | 编号 | 无效等价类 | 编号 |
| 结果或图片ID | 正确ID | E1 | 错误ID | E2 |

表6-12 用户查询缺陷详细功能测试用例表

|  |  |  |  |
| --- | --- | --- | --- |
| 编号 | 结果或图片ID | 覆盖等价类 | 预期输出 |
| U1 | 正确ID | E1 | 查询成功 |
| U2 | 错误ID | E2 | 查询不成功 |

测试用例U1的执行情况如图6-18所示，用户能够成功查询缺陷详细内容。

测试用例U2为错误的查询方式，相应提示操作错误执行情况如图6-19所示。经过测试，查询缺陷详细功能测试用例U1~U2均符合预期。

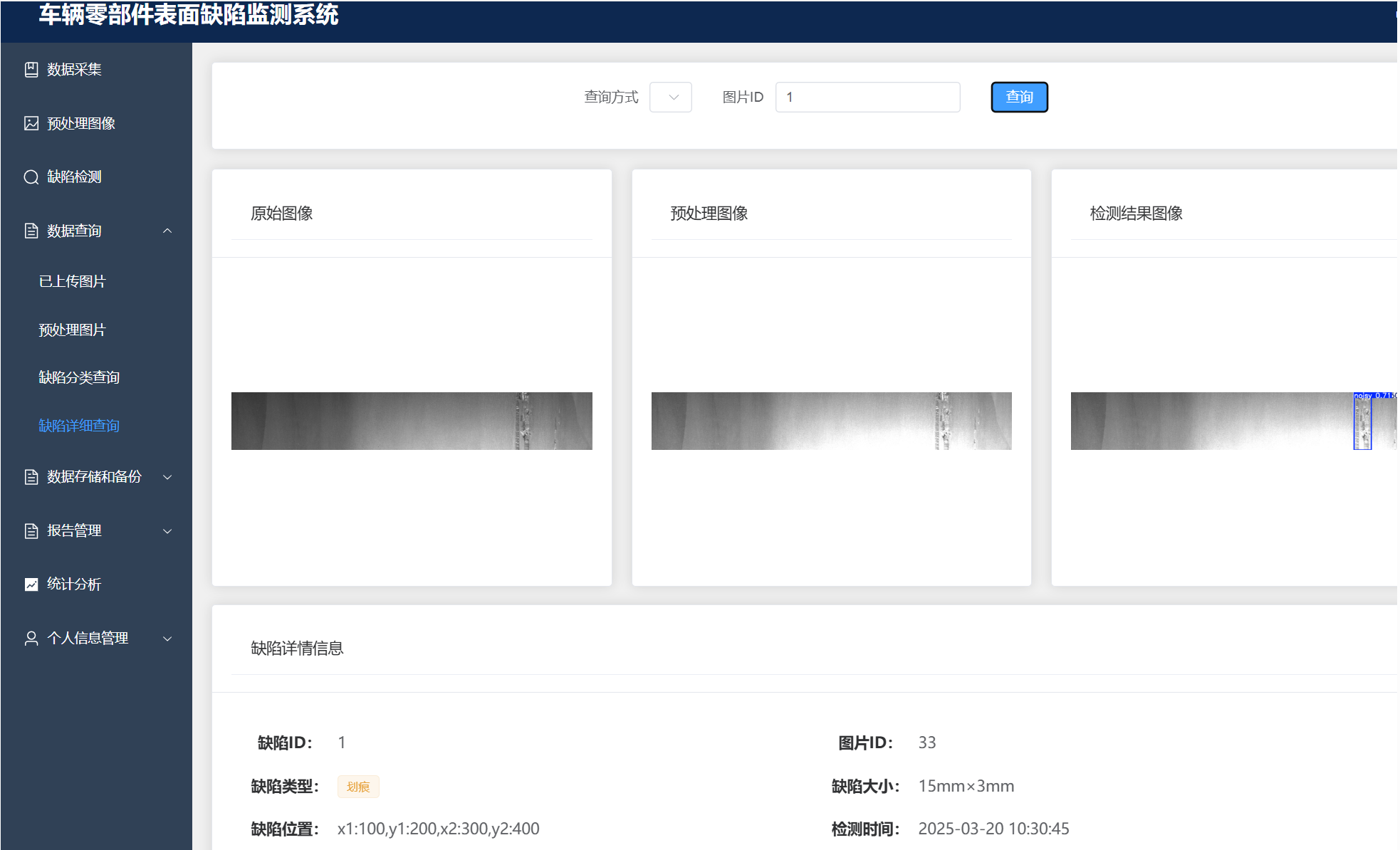


图6-18 成功查询图片



图6-19 不成功查询图片

## 6.7 测试结果分析

通过对酒店预定入住管理的黑盒测试，根据各功能模块所确定的等价划分类，对由此得出的各个测试用例的测试，未发现明显问题。