

测试文档管理系统的设计与实现

数学与信息学院 计算机科学与技术专业
105032016065 李可 指导教师：林岭

【摘要】测试文档管理系统是一个收集项目测试文档的系统，在此系统上每个项目都有单独的测试文档，测试人员可以针对每个项目上传测试文档，在该系统上也可以对测试人员，测试项目所属的公司进行管理。该系统是用面向对象语言 Java 在基于 springboot 框架下在 idea 上编写的项目，并且结合了优秀的网络与人工智能技术，驱动整个系统设计合理、运作流畅，走在时代技术的前沿。

【关键词】Java web;springboot;mybatis

目录

1. 引言	3
1.1 系统设计背景	3
1.2 系统设计目标	3
1.3 系统贡献	3
1.4 开发语言和框架	3
2. 系统需求	3
2.1 需求说明	3
2.2 用例图	4
2.2.1 普通测试员用户	4
2.2.2 管理员	4
2.3 用例规约	5
2.3.1 登录用例规约	5
2.3.2 人员管理用例规约	5
2.3.3 部门管理用例规约	6
2.3.3 客户管理用例规约	6
2.3.4 项目管理用例规约	7
2.3.5 版本管理用例规约	7
2.3 功能需求	7
2.3.1 前台用户功能	7
2.3.2 管理员	8
3. 系统设计	9
3.1 概要设计	9
3.1.1 系统分析	9
3.1.2 系统流程图	9
3.1.3 系统功能模块设计	9
3.2 详细设计	10
3.2.1 系统框架	10
3.2.2 关系模式结构	11
3.2.3 ER 图设计	11
3.2.4 数据库设计	12
3.2.5 用户权限与安全登录模块设计	14
3.2.6 功能模块具体设计	14
3.2.7 文档和图片存储设计	15
4. 系统实现	15
4.1 系统开发环境	15
4.2 系统功能实现	16
4.2.1 基于 SpringSecurity 表单登录	16
4.2.2 用户权限登录	17
4.2.3 数据库连接设计	18
4.2.4 人脸识别	18
4.2.5 文件操作	19
4.2.5 业务操作	20
5. 系统测试	22
5.1 测试方法	22

5.2 测试环境.....	22
5.3 测试对象.....	22
5.4 测试用例.....	23
5.4.1 登录注册测试用例	23
5.4.2 个人信息界面	23
5.4.3 管理员权限	23
5.4.4 用户基本功能	24
5.4.5 文档功能	24
5.5 测试结论.....	25
6. 总结.....	25
致谢.....	25
参考文献.....	25

1. 引言

1.1 系统设计背景

软件测试环节在一个项目中的作用绝对是重中之重，在进度安排中，由于种种没有预料到的原因受到的影响，单元调试和系统测试所收到的牵涉会比预期的更加多。并且时间上所遇到的错误、缺陷数量会达到难以察觉的程度。理论上，缺陷的数量应该为零。但是，由于我们对未来的美好预期，通常实际出现的缺陷数量会比想象中的要多。因此，软件测试进度安排一般来说是项目安排中最不合理的部分。所以对于软件任务的进度安排是 1/3 计划，1/6 编码，剩下 1/2 时间通常安排在测试身上^{错误!未找到引用源。}。

所有的软件测试就是为程序或系统能够按预期设想运行而创立信心的过程^[2]。软件测试决定了项目是否能够符合预期需求成功交付给客户。软件开发本质上是一项在错综复杂关系下的系统工作，软件测试作为重要一环，测试文档对公司来说没有一个特定的平台进行放置显得格外的麻烦，对于开发部门和测试部门的交流都会带来不便。

1.2 系统设计目标

本系统是针对软件测试文档的存放设计的，测试部门测试完项目后对测试文档进行编写上传到该系统项目管理目录下对应的项目中，开发人员登录该系统查看测试结果然后对项目进行完善。公司通过搜索功能搜索到各个项目的测试情况。

1.3 系统贡献

本系统为公司提供了一个测试文档存放的平台，方便了测试部门和开发部门的交流，也可以在本平台中存放不同项目的测试文档，还能很直观的看到该项目的进度，项目测试人，测试文件等信息，公司所有项目测试可以通过这个方式存放。方便了开发部门和测试部门的交流。

1.4 开发语言和框架

该项目所用的后台框架是 Spring boot。Spring boot 是 2013 年推出的特别项目，使得 spring boot 比 Spring 开发框架更加简洁。在开发、配置、调试、部署工作上的作用明显增强，并且在项目内集成了非常的方便使用的基础框架^{错误!未找到引用源。}。使用的服务器为 Tomcat，并在 Gradle 中加入了相应的依赖包。数据库方面采用了 MyBatis 框架技术，MyBatis 自动生成方式可以替代了 JDBC 代码与参数的设置以及结果的搜索，它通过简洁明了的 XML 文件对原始文件映射和属性进行配置，能生成接口和 Java 对象^{错误!未找到引用源。}。

2. 系统需求

2.1 需求说明

本系统最主要实现的功能是对测试项目的管理，普通测试员登录该系统后修改自己的用户信息，比如姓名、密码、头像、电话号码等信息。根据人员管理目录可以查询到同事的联系方式，部门等信息。作为管理员可以对人员的信息进行修改，删除等操作，例如给测试员分配部门和权限，维护测试员的信息。测试员可以查看所以部门信息，管理员可以对部门信息进行新增部门，修改部门名称与描述的操作。测试员在客户管理目录下对客户信息进行添加，删除等操作，可以更好的把客户信息呈现出来，方便和客户之间的交流。测试员添加项目后，在项目的文件中对项目测试文档进行上传、删除或下载操作，达到项目文件储存的功能，方便测试员对项目的测试情况进行交流。在上传完测试文件后可以在项目文档的界面记录这次测试的信息，这条信息会生成版本信息存储在数据库中来达到记录的一个功能。

2.2 用例图

2.2.1 普通测试员用户

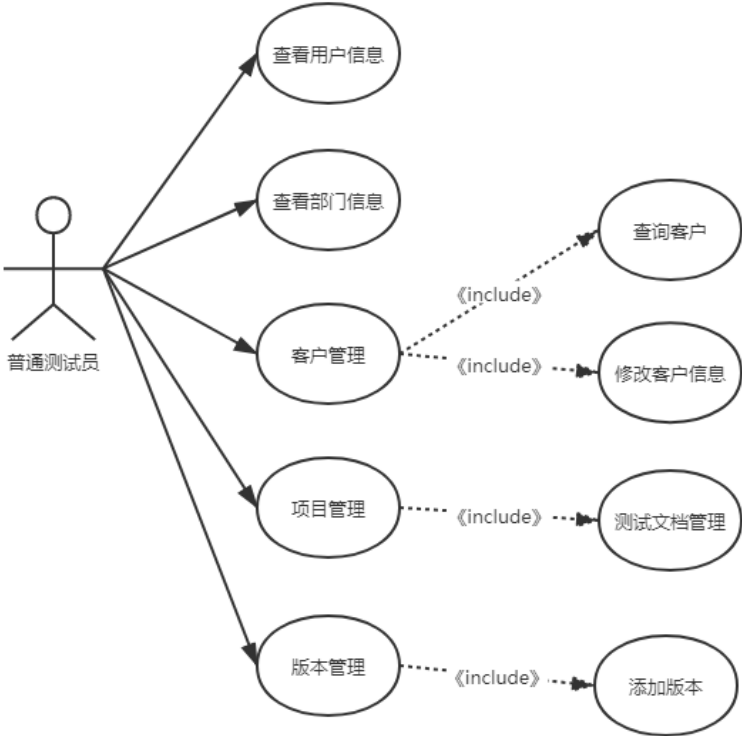


图 2-1 测试员用例图

普通测试仪用户系统需具备以下功能：

- (1) 登录的普通测试员用户登录后进入系统进行查询信息，可以查看到所有测试员，部门，项目负责人，项目，版本信息。
- (2) 普通测试员可在用户信息中对自己的信息进行修改，但是不可以修改自己的部门和权限，也不可以修改其他人的信息。
- (3) 普通测试员可以添加项目与客户，并对测试文档进行上传，删除操作。
- (4) 普通测试员在用户信息，部门信息，项目信息，版本信息中都可以进行搜索。

2.2.2 管理员

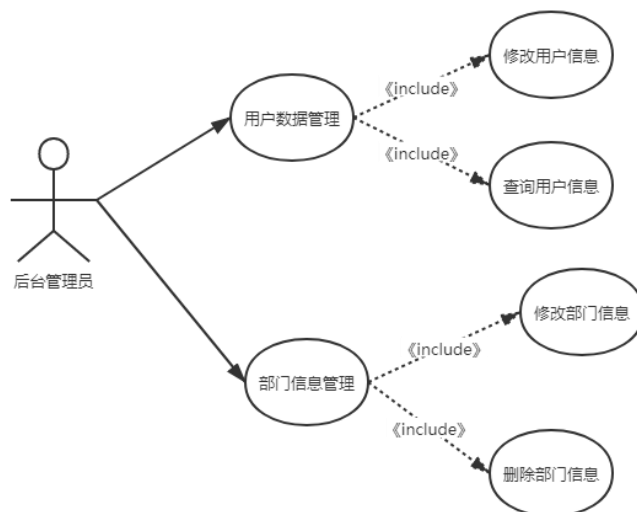


图 2-2 后台管理员用例图

管理员系统功能如下：

- (1) 管理员可以对所有用户信息进行管理，分配用户的部门，职务和权限信息。
- (2) 管理员可以对部门进行管理，修改部门的信息，新增部门或删除部门。

2.3 用例规约

2.3.1 登录用例规约

用例编号	CD-1
用例名称	登陆系统
用例描述	验证用户身份合法后允许登录
参与者	管理员、普通用户
前置条件	进入登录页面
后置条件	参与者登陆成功，进入主页面
涉众利益	1、用户登录后才能使用所有功能 2、管理员登陆后可以对人员信息和部门信息进行管理
基本路径	1、打开系统页面 2、进入登陆界面 3、键入用户名密码 4、参与者提出登录请求 5、系统检验参与者信息是否填写完整 6、系统检查参与者信息是否与数据库内信息相对应 7、系统检查参与者的权限 8、登陆成功
拓展点	1、登陆者信息不完整，提示后返回界面 2、登录者信息不正确，提示后返回界面 3、参与者为管理员则会显示拥有管理员权限的界面内容
非功能需求	页面简介满足且需求

2.3.2 人员管理用例规约

用例编号	CD-2
用例名称	人员管理

用例描述	对所有人员进行管理
参与者	管理员、普通用户
前置条件	进入人员管理界面
后置条件	点击人员信息
涉众利益	1、普通用户可以查询人员信息 2、管理员可以删除或者修改人员信息
基本路径	1、 成功登录系统 2、 点击人员管理下的人员信息 3、 管理员点击人员信息后面的删除按钮 4、 管理员点击人员信息后面的修改按钮 5、 普通用户点击查询按钮
拓展点	1、普通用户不能对人员删除或者修改，只有管理员才有该权限
非功能需求	点击人员信息即出现所有人员的信息

2.3.3 部门管理用例规约

用例编号	CD-3
用例名称	人员管理
用例描述	对所有部门进行管理
参与者	管理员、普通用户
前置条件	进入部门管理界面
后置条件	点击部门管理
涉众利益	1、用户可以查询部门信息 2、管理员可以删除或者修改部门信息
基本路径	1、 成功登录系统 2、 点击部门管理 3、 管理员点击新增部门 4、 管理员点击部门信息，每条部门信息后有删除，修改的按钮 5、 普通用户点击部门信息，只可以查看部门信息与查询部门信息按钮
拓展点	1、普通用户不能对部门删除或者修改，只有管理员才有该权限
非功能需求	点击部门信息即出现所有部门的信息

2.3.3 客户管理用例规约

用例编号	CD-4
用例名称	客户管理
用例描述	对所有客户进行管理
参与者	管理员、普通用户
前置条件	进入客户管理界面
后置条件	点击客户管理
涉众利益	1、用户查询客户信息 2、用户可以对客户的信息进行修改与删除
基本路径	1、 成功登录系统 2、 点击客户管理 3、 用户点击客户信息后面的删除，修改按钮来进行操作 4、 用户点击新增客户按钮对客户进行增加
拓展点	
非功能需求	

2.3.4 项目管理用例规约

用例编号	CD-5
用例名称	项目管理
用例描述	对所有项目进行管理
参与者	管理员、普通用户
前置条件	进入项目管理界面
后置条件	点击项目管理
涉众利益	1、用户可以查询到项目信息 2、用户可以对项目的信息进行修改与删除 3、用户可以对项目文件进行删除、上传与下载
基本路径	1、成功登录系统 2、点击项目管理 3、用户点击项目信息后面的删除，修改按钮来进行操作 4、用户点击新增项目按钮对客户进行增加 5、用户点击项目新信息后的文件对项目文件进行操作
拓展点	
非功能需求	

2.3.5 版本管理用例规约

用例编号	CD-6
用例名称	版本管理
用例描述	对版本信息进行管理
参与者	管理员、普通用户
前置条件	进入版本管理界面
后置条件	点击版本管理
涉众利益	1、用户可以查询版本信息 2、用户可以增加版本信息
基本路径	6、成功登录系统 7、点击版本管理 8、用户点击版本信息后面的删除，修改按钮来进行操作 9、用户点击新增版本按钮对版本进行增加
拓展点	
非功能需求	

2.3 功能需求

本系统最主要实现的功能是用户登陆注册功能，人脸识别功能，用户管理功能，部门管理功能，客户管理功能，项目管理功能，版本管理功能。用户可通过账号密码登录，也可以通过人脸识别登录，登陆后可有五个功能选项，分别是用户管理，部门管理，客户管理，项目管理，版本管理功能。普通测试用户登录后通过用户管理就可以看到其他测试人员的信息，部门管理可以查看部门信息，在项目管理中可以添加项目，或者在已有的项目中，添加测试文件，当项目进行迭代的时候，将迭代信息记录在版本管理当中。而后台管理则是管理测试人员的信息，可以分配测试人员的部门，管理部门信息，可以新增和删除部门。

2.3.1 前台用户功能

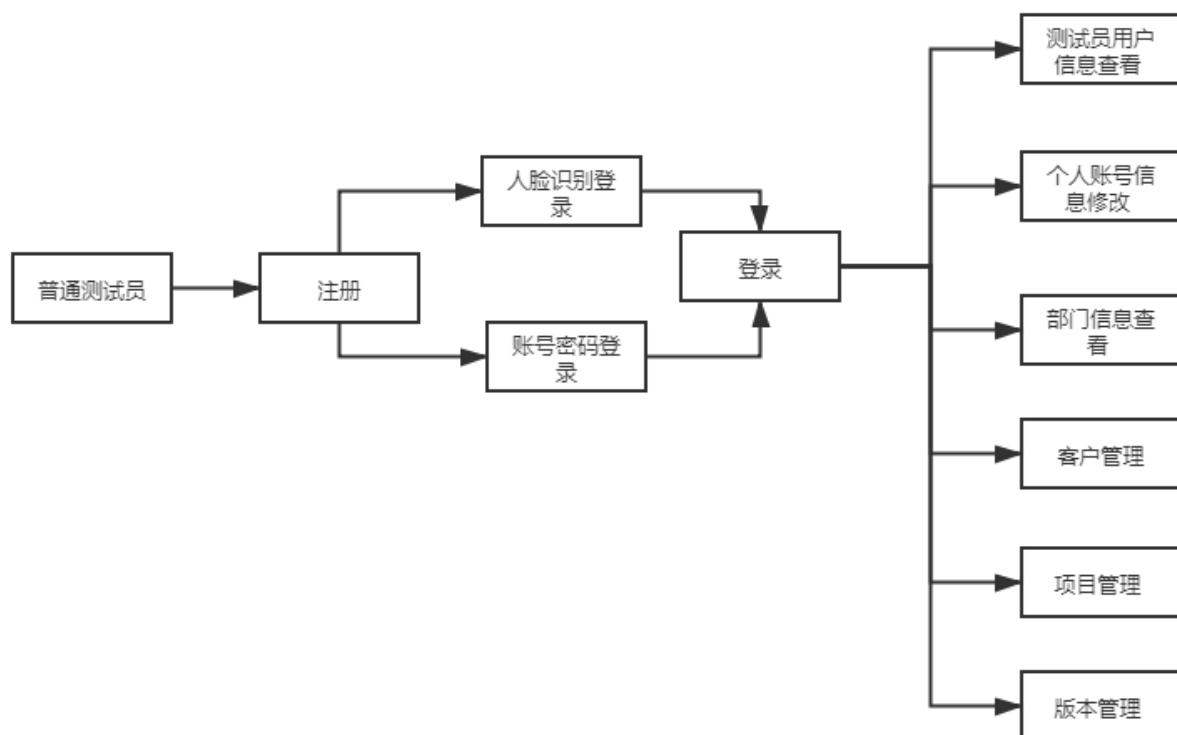


图 2-3 普通用户功能图

(1) 用户注册：测试员若想使用该系统，需先进行注册，注册信息必须符合注册要求，并且上传个人真实正脸且清晰的照片，系统会自动识别个人的照片是否符合要求。

(2) 用户登录：用户可以通过账号密码登录，也可以通过人脸识别登录。用户通过人脸识别登录时，输入人脸识别的用户账号，系统通过比对用户账号的个人照片实现登录。

(3) 测试员用户查看：普通测试员登录后可以看到其他同事的电话号码，邮箱等信息，方便同事之间的联系和交流。

(4) 个人账号信息修改：登录后可以修改个人的用户信息，比如用户名，密码，邮箱，电话，头像等信息。

(5) 部门信息查看：普通测试员登录后可以查看公司所有的部门信息。

(6) 客户管理：普通测试员登陆后可以查看所有客户的电话，邮件，所属公司等信息，也可以自行添加新的客户信息。

(7) 项目管理：测试员管理项目的测试文档，可以删除测试文档或添加测试文档。

(8) 版本管理：管理员将项目进行测试后，项目优化再测试的信息记录在版本管理中。

2.3.2 管理员

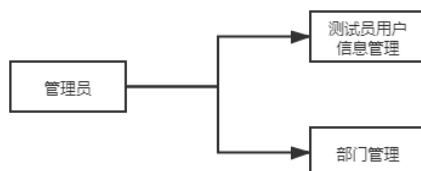


图 2-4 管理员功能图

(1) 测试员人员管理：管理员需要对测试人员的信息进行管理，比如说给测试人员分配部门，权限等操作。

(2) 部门管理：管理员对部门的信息进行管理，可以进行删除部门，新增部门，修改部门信息等操作。

3. 系统设计

3.1 概要设计

3.1.1 系统分析

本系统是一个对测试文档进行管理的网站，分为测试人员模块和管理员模块。输入网址进入本系统后进行注册后登录测试文档管理系统后可以查看同事的联系方式等信息和部门的信息，也可以对客户进行管理，方便公司对客户的联系和记录。测试员测试完项目后可以新建项目并且在项目文档中上传测试文档，并且对测试文档进行管理，修复后的项目再测试可以上传版本信息，方便维护。本系统为测试公司带来了便利，简洁的操作。

3.1.2 系统流程图

系统用户功能流程图如下：

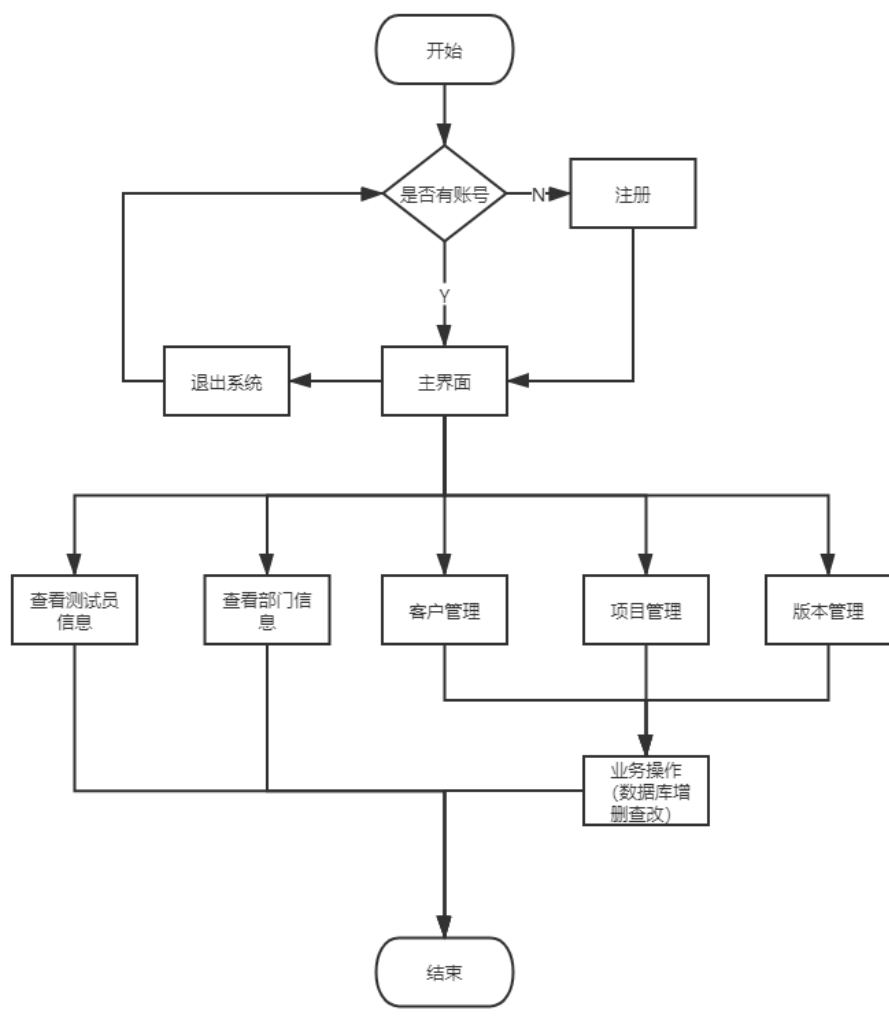


图 3-1 用户使用流程图

输入网址后，进行登录操作，如果还未有账号则进行注册操作。登录到主界面后，用户可以查看其他测试员信息，比如姓名，部门，电话号码，邮箱等信息，可以方便同事之间的交流，然后可以查看公司所有部门的信息，接着可以对客户进行管理，添加或者查找客户。测试员测试完项目后对项目文档进行管理，添加项目，删除项目，或者在项目中选择上传，删除测试文档。测试员对改版后的初代项目进行测试，把版本信息更新进版本管理中。

3.1.3 系统功能模块设计

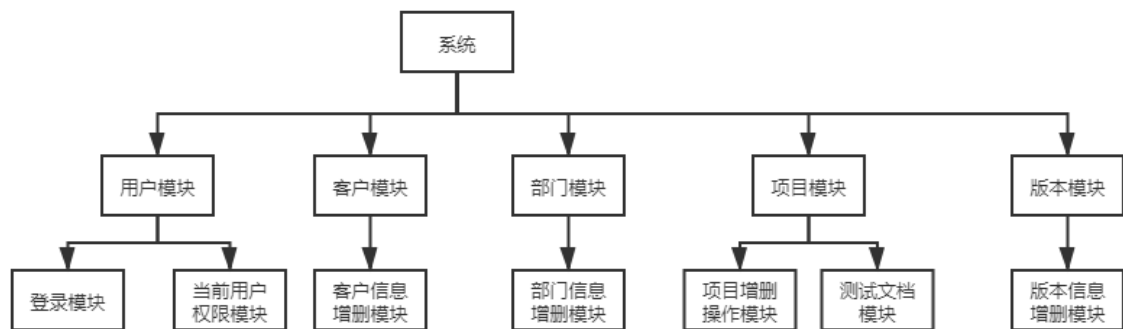


图 3-3 系统设计

本系统的基本模块分为用户模块，客户模块，部门模块，项目模块和版本模块。

用户模块有登录模块和当前用户权限模块；登录模块可以实现普通测试员用户和管理员用户登录，和测试员的注册，用户当前基本信息和密码的修改。检测当前用户的权限，用户权限有两个，user 普通用户和 ROLE_SUPERADMIN 管理员。实现用户权限是一个通过 tb_authority 数据库与 springsecurity，具体语句 `<sec:authorize access="hasRole('ROLE_SUPERADMIN')">` 来实现的。

客户模块有客户信息增删模块，客户信息增删模块是对客户进行操作，可以选择增加客户或者查询客户，具体在服务层 ClientService 中实现对储存在数据库中的客户信息进行增删查改等函数的编写。

部门模块有部门信息增删模块，部门信息模块是对部门进行操作，可以选择增加部门或者修改部门信息，具体在服务层 DepartmentService 对储存在数据库中的部门信息进行增删查改等函数的编写。

项目模块有项目增删操作模块和测试文档模块，项目增删操作模块是对项目进行操作，可以选择增加项目，修改项目内容，删除项目，具体操作在服务层的 ItemService 对储存在数据库中的项目信息进行操作。测试文档模块是项目中存储测试文档信息的模块，测试文档中包含测试计划、用例、记录、缺陷、输入等文件夹。文档储存和上传是用 MultipartFile 实现的。

版本模块有版本信息增删模块，版本信息增删模块是对版本信息进行操作，可以选择对版本信息进行添加或者修改，具体在服务层 VersionService 中对储存在数据库的版本信息进行增删查改等函数的编写。

3.2 详细设计

3.2.1 系统框架

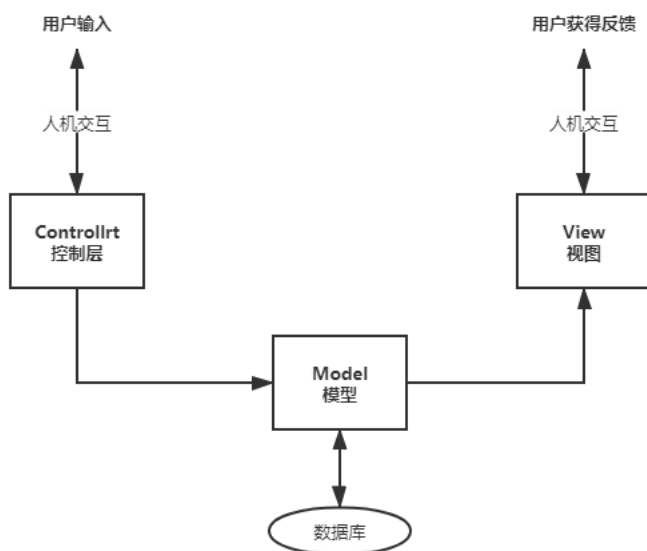


图 3-4 框架设计

本系统包含控制层 Controller， Model 层包含了业务层 service 和数据层 DAO，数据库方面使用了 MySQL 数据库，并且根据功能编写了七个实体类，具体有权限类，客户类，部门类，项目类，状态类，用户类，版本类。这种设计模式可以很好地表达用户与系统之间的交互以及与系统的交互模式。[5]^[5]。



图 3-5 框架目录

控制层：Controller 负责前后端的连接，响应用户在页面的请求后把请求传给业务层 service。Controller 控制器用于管理用户和视图之间的交互，负责用户界面和业务层之间的过程控制，即将用户的操作映射到特定的业务层和 DAO 层，完成特定的业务逻辑，然后进行处理。业务数据及时反映给用户^[6]。

视图层：代表着用户交互界面，对于 web 应用程序而言，就是网页界面^[7]。

模块：事务逻辑模块，是整个模型的核心^[8]。业务层也叫业务逻辑层，Service 层的作用是完成项目基本的功能设计，接着调用 DAO 层接口，DAO 层从数据库调取数据返回给业务层，业务层返回给控制层，最后控制层返给用户具体的页面和数据。

3.2.2 关系模式结构

- (1) 用户（账号，姓名，密码，性别，手机，邮箱，权限编号，头像，职务，部门，照片）
- (2) 权限（权限编号，权限中文名称，权限英文名称）
- (3) 部门（部门编号，部门名称，部门描述）
- (4) 客户（客户编号，邮箱，客户所属公司，客户姓名，地址，电话）
- (5) 版本（版本编号，项目编号，版本时间，修改人，版本内容）
- (6) 项目（项目编号，项目序号，名称，开始时间，结束时间，测试员账号，项目描述）
- (7) 状态（状态编号，状态名称）
- (8) 设计项目（测试员账号，项目编号）
- (9) 设计版本（测试员账号，版本编号）

3.2.3 ER 图设计

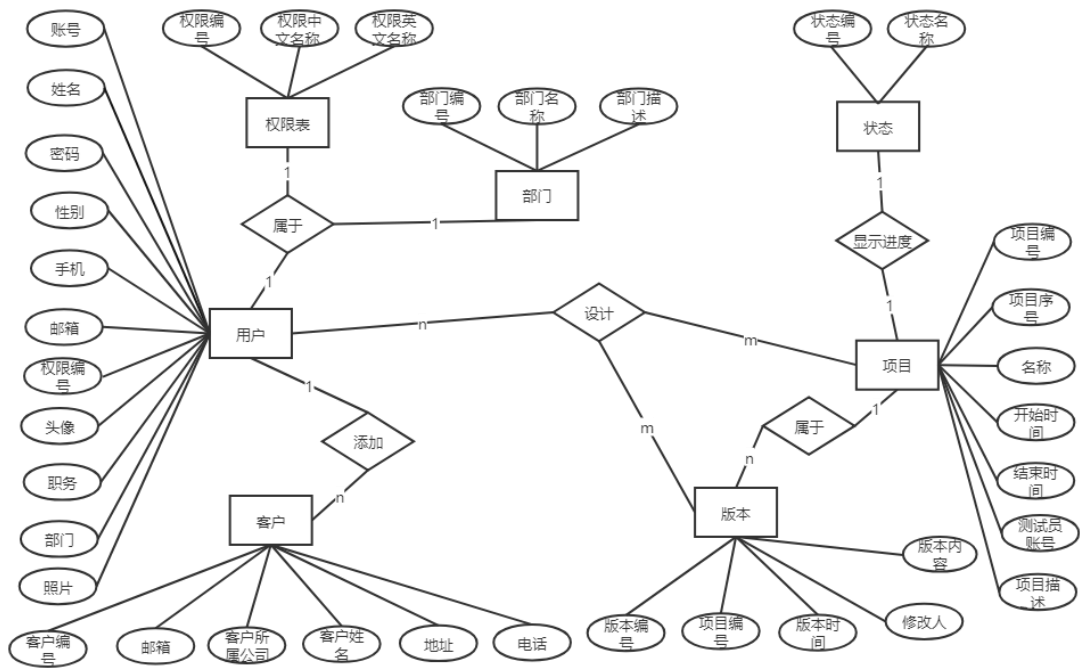


图 3-6 ER 图

3.2.4 数据库设计

用户模块：用户表实现用户注册，登录，包含用户的登录账号，用户名，登录密码，性别，电话号码，电子邮箱，权限信息，头像，用户职责，用户部门，照片。

表 3-1 用户表

字段名	数据类型	长度	是否为空	主外键	备注
account	varchar	255	否	主键	用户账号
name	varchar	255	否		用户姓名
password	varchar	255	否		用户密码
sex	varchar	2	否		用户性别
phone	varchar	11	否		用户手机号码
email	varchar	255	否		用户邮箱
authority_id	varchar	255	否		权限编号
avatar	varchar	255	是		头像
duty	varchar	255	是		职务
department	varchar	255	是		所属部门
ace_avatar	longblob	0	否		照片

用户权限信息模块：用户权限信息模块用来分辨用户的权限信息，每个用户注册后都会自动生成一个权限编号将用户表连接在一起，权限英文名识别权限，权限中文名会显示在界面中。

表 3-2 权限表

字段名	数据类型	长度	是否为空	主外键	备注
auth_id	varchar	255	否	主键	权限编号
auth_English_name	varchar	255	否		权限英文名

auth_Chinese_name	varchar	255	否		权限中文名
-------------------	---------	-----	---	--	-------

客户信息模块：包含客户编号，客户所属公司，客户姓名，联系电话，邮箱，公司地址。

表 3-3 客户表

字段名	数据类型	长度	是否为空	主外键	备注
clien_id	int	11	否	主键	客户编号
clien_company	varchar	255	否		客户所属公司
clien_name	varchar	255	否		客户姓名
cilen_phone	varchar	11	否		联系电话
cilen_email	varchar	255	否		邮箱
cilen_address	varchar	255	否		公司地址

部门模块：包含部门编号，部门称号，部门描述。

表 3-4 部门表

字段名	数据类型	长度	是否为空	主外键	备注
dep_id	varchar	255	否	主键	部门编号
dep_name	varchar	255	否		部门名称
dep_content	varchar	255	否		部门描述

项目模块：包含项目编号，项目序号，项目名称，项目状态，项目开始时间，项目结束时间，测试人账号，项目说明。

表 3-5 项目表

字段名	数据类型	长度	是否为空	主外键	备注
item_id	int	18	否	主键	项目编号
item_num	varchar	255	否		项目序号
item_name	varchar	255	否		项目名称
state	varchar	4	否		项目状态
item_startdate	date	0	否		开始时间
item_deadline	date	0	否		结束时间
account	varchar	255	否		测试员账号
item_content	varchar	255	是		项目描述

状态模块：描述项目的状态，不同的字符代表不同的状态，有如下六个状态：未测、启动、计划、用例、报告、结束。

表 3-6 状态表

字段名	数据类型	长度	是否为空	主外键	备注
state_num	varchar	255	否	主键	状态编号
state	varchar	255	否		状态名称

版本模块：包含版本编号，项目编号，版本时间，版本修改人，版本内容。

表 3-7 版本表

字段名	数据类型	长度	是否为空	主外键	备注
-----	------	----	------	-----	----

ver_id	varchar	255	否	主键	版本编号
item_id	varchar	255	否		项目编号
ver_time	varchar	255	否		版本时间
ver_people	varchar	255	否		版本修改人
ver_content	varchar	255	否		版本内容

3.2.5 用户权限与安全登录模块设计

每个用户都有独一无二的权限编号，进入系统点击注册，填写注册信息提交确认，用户表添加了一个新的用户，此时这个用户也会生成一条属于该用户的权限信息，此权限信息对应的权限 id 与权限表相连接，用户注册默认权限是普通测试员用户，没有修改部门信息与用户的职务和部门的权限。

(1) LoadUserByUsername (String username)

该函数的作用是根据登录时提供的用户名去查用户的权限 id，接着去判断权限表中该用户的权限。

(2) public Collection<? extends GrantedAuthority> getAuthorities()

该函数的作用是将 List<Authority>转成 List<SimpleGrantedAuthority>, 否则前端拿不到角色列表信息，无法判断权限。

为了实现用户的安全登录，我们需要对用户的密码进行加密，我们使用最常见的对称加密算法，在 utils 工具包下新建一个 AESUtils 类。

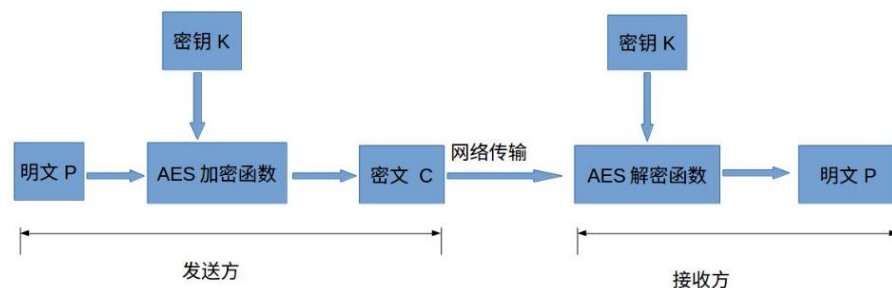


图 3-4 AES 加密

(1) public static String Encrypt(String sSrc,String sKey)

该函数的目的是使用参数 sKey 中的密钥进行加密。

(2) public static String Decrypt(String sSrc,String sKey)

该函数的目的是使用参数 sKey 中的密钥进行解密。

3.2.6 功能模块具体设计

本系统有用户，部门，客户，项目，版本五个对象，这五个对象分别对应着五个实体类，按照需求设计数据库后编写实体类并存放在 entity 包下，五个功能对应不同的控制层 Controller, 服务层 service, DAO 层 mapper 包。

具体操作都是对数据的增删查改。数据的增删查改指令被传送到 controller 层，再经过 service 层提取数据库中的信息。以下函数都在服务层，DAO 层中与数据库相连接。该部分使用了 MyBatis 框架，该框架是一个支持普通 SQL 查询。存储过程和高级映射的优秀持久框架，能够支持自己编写 sql，操作特别灵活^[9]。

(1) insert 函数

新增数据函数，可以用于注册用户，添加部门信息、客户数据、项目、版本信息。

(2) insertSelective 函数

其作用和 insert 函数相似，但是不同点在于 insert 函数会将所有字段都添加进数据库，即使没有值。但是 insertSelective 函数会对传进来的字段进行非空判断，只会将有值的字段传入数据库中。如果数据库中有 default 值，default 值就不起作用了，而 insertSelective 不会忽略 default 值

(1) deleteByPrimaryKey 函数

根据表的主键去找表的信息，来删除这条数据，可以用于注销用户，删除客户，删除部门，删除项目，删除版本。

(2) updateByPrimaryKey 函数

用于数据的更新，根据主键找到数据，然后可以用于修改用户信息，部门信息，客户信息，项目信息，版本信息。

(1) updateByPrimaryKeySelective 函数

其作用和 updateByPrimaryKey 函数相同，但是 updateByPrimaryKeySelective 函数只更新不是空的字段，而 updateByPrimaryKey 函数会将空的字段置为 NULL。

(2) selectByPrimaryKey 函数

按照唯一标识符寻找数据。

(3) List<Department> findAllDepartment 函数

以部门信息为例，其他功能实现函数类似这个，目的是在界面显示出全部的部门信息。由此可知，该类型函数可以在界面显示所有的用户信息，客户信息，项目信息。

(4) List<Department> findByDeptName 函数

以部门信息为例，该功能是按照部门名称寻找部门。由此可知，该类型的函数可以实现按照用户姓名寻找用户，客户姓名寻找客户，项目姓名寻找项目和版本。

3.2.7 文档和图片存储设计

图像的存储是前端将图像转化为 64 位的字节码，在前端编写上传图片函数，使图片 url 数据以 Blob 形式存储。

文档的存储是本系统实现完整功能的重要一环，根据需求将文档分为测试计划、用例、记录、缺陷报告和测试报告。测试员根据项目发展的不同阶段存储测试相关的文件。封装好 Utils 类，文件的操作被编写在 controller 层。

(1) private void saveFile(String path, MultipartFile file)

该函数的目的是判断文件夹是否为空，如果不为空则可以进行新建文件夹的操作。

(2) fileUpload 函数

该函数为文件上传函数，调用 springboot 中的 MultipartFile[] 来自动识别前端传来的文件，再利用文件流将文件信息保存在本地的数据库中，以进行多个文件或单个文件的上传。

(3) downloadOne 函数

文件下载操作，该函数先判断文件父目录是否存在，再将文件转化为输出流，实现单文件的下载。

(4) downloadMore 函数

文件下载操作，首先，在上传界面要指定编码类型为 multipart/form-data。只有这样，我们所要上传的文件内容才会被包含在上传的数据流之中^[10]。该函数对选中的文件用浏览器将文件流进行下载，接着将选中文件的文件流打包压缩，浏览器直接下载压缩包流，所以多文件的下载会自动打包为 zip 形式。

(5) removeFile 函数

该函数实现文件删除操作，先将文件的路径保存在集合中，再调用 windows 系统的文件删除指令可以达到统一删除的结果。

4. 系统实现

4.1 系统开发环境

开发环境：Windows 10

开发软件：Java web: IntelliJ IDEA 2019.3.1 x64、jdk1.8.0_151、mysql-8.0.15-win64、Tomcat 8.0、gradle 3.5.1

辅助算法接口：AES 算法，face++ 技术

4.2 系统功能实现

4.2.1 基于 SpringSecurity 表单登录

首先编写用户的实体类 User，使用 Springsecurity 进行表单验证登录，编写一个 SecurityConfig 配置类来继承 WebSecurityConfigurerAdapter，重写其中的 configure 方法，在里面定义如下逻辑：

```
protected void configure(HttpSecurity http) throws Exception {
    http.authorizeRequests()
        .antMatchers("/css/**",
            "/js/**",
            "/images/**",
            "/druid/**",
            "/swagger-resources/**",
            "/swagger-ui.html",
            "/security/toFaceLogin",
            "/security/toRegister",
            "/security/register",
            "/security/faceContrast*",
            "/index").permitAll() // 都可以访问
        .anyRequest().authenticated()
        .and()
        .formLogin() //基于 Form 表单登录验证
        .loginPage("/security/toLogin")
        .loginProcessingUrl("/login")
        .usernameParameter("username")
        .passwordParameter("password")
        .successForwardUrl("/security/enter")
        .failureUrl("/security/loginError") // 自定义登录界面
        .permitAll()
        .and()
        .rememberMe()
        .rememberMeServices(rememberMeServices())
        .tokenValiditySeconds(REMEMBER_ME_SECONDS)
        .key(KEY)
        .and()
        .logout()
        .logoutSuccessUrl("/security/logOut")
        .and()
        .exceptionHandling()
        .accessDeniedPage("/error")
        .and()
        .sessionManagement()
        .maximumSessions(1)
        .sessionRegistry(sessionRegistry);
    http.csrf().disable();
    http.headers().frameOptions().disable();
}
```

图 4-1 springsecurity 验证登录

http.formLogin 使用 form 表单进行登录操作，如上所示的 URL 是 toLogin 页面的数据，即登录所需要的字段数据，提交的路径为 “/login”，这样配置代表 SpringSecurity 处理表单数据，通过这个路径进

行登录验证处理。除了登录，以上还增加了记住我的功能，勾选了该功能可以将账号密码信息存储七天。用户点击退出后的路径是“/securit/logout”。

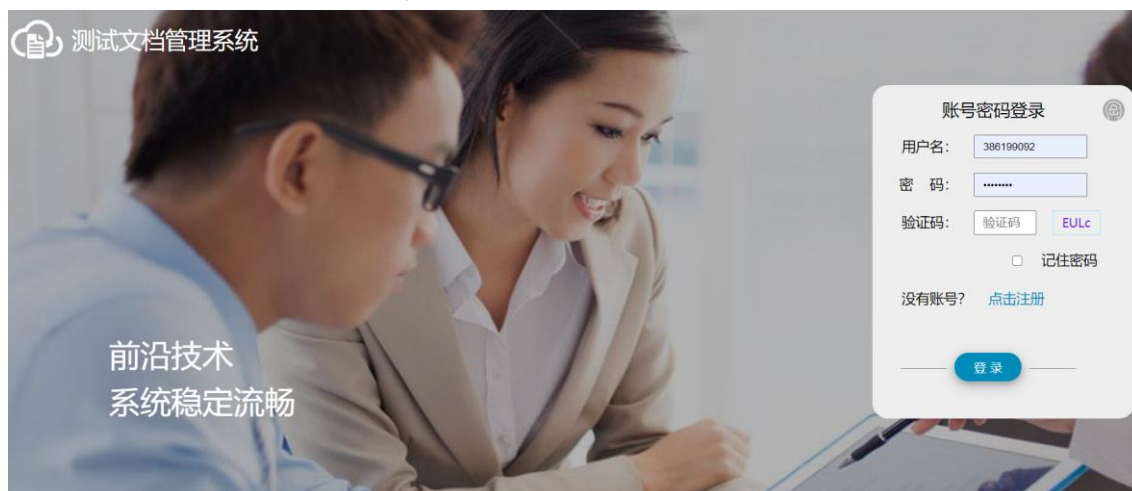


图 4-2 界面登录

4.2.2 用户权限登录

访问控制技术对任何系统来说都是非常重要的信息安全保障措施[11]^[11]。实现权限验证登录，我们在数据库中建立一个权限表，权限编号为主键，也是用户表的外键，每个用户注册后会自动生成一张权限表，默认权限为普通用户。我们从权限表中读取每个用户的角色信息来交给 SpringSecurity 去判断用户的角色。

我们编写一个 UserServiceImpl 来实现 UserDetailsService 接口，而 UserDetailsService 中提供一个 LoadUserByUsername (String username) 的方法，这个方法的作用是根据登录时提供的用户名去查用户的权限 id，接着去判断权限表中该用户的权限。User 实体类实现 UserDetails 接口使用一个类 GrantedAuthority,这个类表示已经被授权的权限然后实例化一个 SimpleGrantedAuthority 类。首先我们从数据库中读取到用户角色，用户角色为一个列表，有权限 id，权限英文名称，权限中文名称，接着 SimpleGrantedAuthority 类把权限信息交给 SpringSecurity 来验证。

```
public Collection<? extends GrantedAuthority> getAuthorities() {  
    // 需将 List<Authority> 转成 List<SimpleGrantedAuthority>, 否则前端拿不到角色列表名称  
    List<SimpleGrantedAuthority> simpleAuthorities = new ArrayList<>();  
  
    simpleAuthorities.add(new SimpleGrantedAuthority(this.authority.getAuthority()));  
  
    return simpleAuthorities;  
}
```

图 4-3 验证权限



图 4-4 界面功能

4.2.3 数据库连接设计

本系统的关系型数据库为 MySQL，连接池使用 druid，因为我们使用的 MySQL 版本是 8.0.15 所以在 application.yml 配置如下所示：

```
spring:
  datasource:
    druid:
      driver-class-name: com.mysql.cj.jdbc.Driver #
      url: jdbc:mysql://localhost:3306/likedb?serverTimezone=Asia/Shanghai
      username: root
      password: root
      initial-size: 5 # 下面为连接池的补充设置，应用到上面所有数据源中
      min-idle: 5
      max-active: 20
      max-wait: 60000 # 配置获取连接等待超时的时间
      time-between-eviction-runs-millis: 60000 # 配置间隔多久才进行一次检测，间隔多久
      min-evictable-idle-time-millis: 300000 # 配置一个连接在池中最小生存的时间
      validation-query: SELECT 1
      test-while-idle: true
      test-on-borrow: false
      test-on-return: false
      pool-prepared-statements: true # 打开PSCache，并
      max-pool-prepared-statement-per-connection-size: 20
      filters: stat # 配置监控统计拦截的filters，去掉后监控
```

图 4-5 数据库配置信息

4.2.4 人脸识别

为了实现人脸识别的功能，我们选择调用旷视平台 face++ 的 api 接口来检测人脸，其只要功能在于用户传入自己的真人照片时，检测用户的照片是否合格，例如是否只有一用户一个人，用户上传的照片正脸是否模糊不清或有遮挡。用户上传完自己真人正脸合格照片后，可以通过前端 hasUserMedia() 函数调用电脑摄像头在登录时进行人脸识别验证登录。

首先为了调用 face++ 平台，我们要申请自己独一无二的 API key，然后在系统里调用 API，face++ 利用人工智能算法识别人脸，申请的个人 API 是 HTTP API，在系统中调用后会发起 HTTP 请求，即使使用者向旷视平台的服务器发起 HTTP 请求，并加上合适的参数，服务器会对请求进行处理，得到结果会返给使用者。调用方法使用 POST 调用，需要传的参数如下：

```
public String compare(String token1, String token2) {  
  
    String compareUrl = "https://api-cn.faceplusplus.com/facepp/v3/compare";  
  
    List<BasicNameValuePair> formparams = new ArrayList<>();  
    formparams.add(new BasicNameValuePair( name: "api_key", value: "7vUA3ZGQCoQP5T1UFyV9QPBk6VtLGssE"));  
    formparams.add(new BasicNameValuePair( name: "api_secret", value: "t03oox4092nYvSMvEXrLI-dnHeTw6q8B"));  
    formparams.add(new BasicNameValuePair( name: "face_token1", token1));  
    formparams.add(new BasicNameValuePair( name: "face_token2", token2));  
    return postHttp(formparams, compareUrl);  
}
```

图 4-6 主要参数

以上传参列表可以看出，api_key 和 api_secret 是必要传的，而 face_token1 和 face_token2 是两张图片的数字信息，我们选择的方式是上传 base64 编码的图片数据，face_token1 是之前用户存在用户表中的图片信息，face_token2 是用户选择人脸识别登录时调用摄像头来得到的人脸信息参数。用户人脸识别后，java 发送 http 请求传入了 api_key, api_secret, face_token1, face_token2 四个参数，调用成功后会得到 confident 参数，即两张人脸的相似置信度，通过 face++ 的结果对比如果置信度达到了千分之一 62.327 就可以认为两张照片为一个人，但是可能性并不是特别高。

4.2.5 文件操作

springboot 用的是 MultipartFile 来进行文件上传，第一步我们先配置最大上传文件的大小和总上传文件最大的容量，并将配置信息封装在 FileUploadConfig 类中。

```
public MultipartConfigElement multipartConfigElement() {  
    MultipartConfigFactory factory = new MultipartConfigFactory();  
    //单个文件最大  
    factory.setMaxFileSize("80MB"); //KB,MB  
    /// 设置总上传数据总大小  
    factory.setMaxRequestSize("102400KB");  
    return factory.createMultipartConfig();  
}
```

图 4-7 文件配置

配置完成后如果文件超出限制会在进入 controller 层前抛出异常，但是在这个规定范围内不会受到影响。MultipartFile 接受文件是通过二进制流的方法，即利用 MultipartFile.getInputStream() 函数输入到文件流中保存文件。文件的上传下载删除等操作函数在控制层 VersionController 中实现。首先是文件上传操作功能，在 JSP 代码中必须将 POST 请求表单设置为 enctype="multipart/form-data"，并且在控制层中指定将文件位置传到该项目中去。

```

public String fileUpload(@RequestParam("file") MultipartFile[] files,
Map<String, Object> map,
    @PathVariable("verId") Integer verId,
    @PathVariable("getFileName") String getFileName,
    @PathVariable("getBiggest") Double getBiggest) {

    for(MultipartFile f:files){
        System.out.print(f.getOriginalFilename()+" ");
    }
    System.out.println();

    Utils utils = new Utils();
    Version version = versionService.selectByPrimaryKey(verId);
    version= utils.ver(files, version,getFileName);
    versionService.updateByPrimaryKeySelective(version);

    map.put("version", version);
    if(utils.theMap(version,getFileName).get("stringList")!=null)
        map.put("stringList",
            utils.theMap(version,getFileName).get("stringList"));

    map.put("getFileName", getFileName);
    map.put("getBiggest", getBiggest);
    map.put("userList", userService.findAllUser());

    return "version/version_file";
}

```

图 4-8 文件上传函数

文件下载功能实现是先判断要下载文件的父目录是否存在，在用 `response.setHeader` 的方法让服务器告诉浏览器它发送的数据属于什么文件类型，当 `Content-Type` 为要需要下载的类型时，这个信息头会传送浏览器需要下载的文件的名字与类型。文件的下载过程是将文件转化为输出流，通过浏览器将文件的输出流下载下来。

```

FileInputStream fis = null; //文件输入流
BufferedInputStream bis = null;

OutputStream os = null; //输出流

os = response.getOutputStream();
fis = new FileInputStream(file);
bis = new BufferedInputStream(fis);
int i = bis.read(buffer);
while(i != -1){
    os.write(buffer);
    i = bis.read(buffer);
}
bis.close();
fis.close();

```

图 4-9 文件输出流

文件删除操作相对比较简单，总的来说就是将文件路径先保存在集合中接着用 `java` 调用删除文件命令将文件删除即可。

4.2.5 业务操作

针对人员管理，测试部门管理，客户管理，项目管理，版本管理这些功能最重要的就是增删查改的操作，利用 `MyBatis` 框架可以很好的把增删查改的功能进行调用。在控制层中对这些数据进行更新或者

增加。后端 Api 通过 swagger 完成。通过 jsp 和 tablibs 完成，jsp 可以直接书写 html 代码并且会自动翻译成对于的 java 代码，后台人员可以在 jsp 中添加对应的业务程序。

```
@ApiOperation(value="删除操作后台所需要的值")

@ApiImplicitParam(name = "clientId", value = "客户 clientId", required =
true, dataType = "String")
@DeleteMapping(value="/remove/{clientId}")
public String remove(@PathVariable("clientId") String clientId) {

    clientService.deleteByPrimaryKey(clientId);
    return "redirect:/client/list";
}

@ApiOperation(value="进入客户修改界面")

@ApiImplicitParam(name = "clientId", value = "客户 clientId", required =
true, dataType = "String")
@GetMapping(value="/preUpdate/{clientId}")
public String preUpdate(@PathVariable("clientId") String clientId,
Map<String, Object> map){

    map.put("client", clientService.selectByPrimaryKey(clientId));
    return "client/update_client";
}

@ApiOperation(value="修改操作需要传入后台的值")
@PutMapping(value="/update")
public String update(Client client) {
    clientService.updateByPrimaryKey(client);
    return "redirect:/client/list";
}
```

图 4-8 控制层中的数据操作

在数据信息显示界面中需要用到分页功能，使用了 pageHelper 工具，该工具可以与 MyBatis 相适应，再用 java 的 map 集合将信息传送到前台显示出来。

客户管理						请输入您要搜索的客户编号	搜索
客户编号	公司名称	客户姓名	联系电话	邮箱	公司地址	操作	
01	福建师范大学	李明	15059928431	qmbar@163.com	福建师范大学旗山校区	修改	删除
02	福建师范大学	coco	15059928431	likelem@163.com	天桥底下	修改	删除

图 4-9 分页显示


```

@ApiOperation(value="获取分页列表", notes="用来获取分页列表")

@ApiImplicitParam(name = "pageNoStr", value = "页码:pageNoStr")

@RequestMapping("/list")
public String list(Map<String, Object> map,
                  @RequestParam(value="pageNo", required=false,
defaultValue="1") String pageNoStr,
                  @RequestParam(value="pageSize", required=false,
defaultValue="3") String pageSizeStr,
                  @RequestParam(value="searchClientId", required=false)
String searchClientId) {
    Integer pageNo = 1;
    Integer pageSize = 3;
    //对 pageNo 的校验
    pageNo = Integer.parseInt(pageNoStr);
    if(pageNo < 1){
        pageNo = 1;
    }
    //校验 pageSize
    pageSize = Integer.parseInt(pageSizeStr);
    if(pageSize < 1){
        pageSize = 3;
    }
    Map<String, Object> map1 = new HashMap<>();
    if(!isEmpty(searchClientId))
        map1.put("searchClientId", searchClientId);
    PageHelper.startPage(pageNo, pageSize);
    List<Client> clientList = clientService.findByClientId(map1);
    PageInfo<Client> page=new PageInfo<Client>(clientList);
    map.put("page", page);
    map.put("pageSize", pageSize);
    if(!isEmpty(searchClientId))
        map.put("searchClientId", searchClientId);
    return "client/list_client";
}

```

图 4-10 界面显示操作

5. 系统测试

5.1 测试方法

软件的黑盒测试就是将测试对象看成一个黑色的盒子，即不考虑程序内部的逻辑结构，只看功能是否符合要求。所以本系统的测试方法主要是以黑盒测试为主。首先以管理员身份登录系统，测试管理员功能权限，再以普通测试员的身份登录系统，测试测试员的功能。

5.2 测试环境

谷歌浏览器

5.3 测试对象

登录注册界面，个人信息界面， 管理员权限，用户基本功能，文档功能

5. 4 测试用例

5. 4. 1 登录注册测试用例

表 5-1 登录注册测试用例

用例编号	前置条件	操作	预期结果	测试结果
001	进入首页点击注册用户	输入符合系统规定的用户信息，上传符合规定的照片。	注册成功，跳转到主页	通过
002	进入首页点击注册用户	输入不合规定的用户信息	显示用户信息填写错误	通过
003	进入首页点击注册用户	输入符合规定的用户信息，上传不符合规定的个人照片	显示照片信息有误	通过
004	进入首页点击账号密码登录	输入普通用户的账号密码	登入进主页面	通过
005	进入登录页面,选择人脸识别登录	输入自己的账号，对电脑摄像头录入人脸	登入进主页面	通过

5. 4. 2 个人信息界面

表 5-2 个人信息界面测试

用例编号	前置条件	操作	预期结果	测试结果
001	普通用户进入个人信息界面	修改自己的职务或部门	不可修改	通过
002	普通用户进入个人信息界面	修改除了职务和部门以外的信息	修改成功	通过
003	普通用户进入个人信息界面	修改密码，输入和原密码不一样的密码	显示密码不正确	通过
004	普通用户进入个人信息界面	修改密码，输入和原密码相同的密码	修改成功	通过
005	普通用户进入个人信息界面	点击注销账号	账号被注销，返回登录界面	通过

5. 4. 3 管理员权限

表 5-3 管理员权限测试

用例编号	前置条件	操作	预期结果	测试结果
001	管理员登入主页面点击人员管理	查看人员信息	每个人员后面都有删除，修改按钮	通过
002	管理员登入主页面点击人员管理	点击人员信息后面的人员删除	人员账号被注销	通过
003	管理员登入主页面点击人员管理	点击人员信息后面的人员信息修改	不仅可以修改人员的基本信息，还可以分配人员的职位和部门	通过
004	管理员登入主页面	点击部门管理	有部门信息和新增部门	通过
005	管理员点击部门管理下的新增部门。	管理员填写部门信息，并点击确认	新增了一条部门信息，并且用户的可选择的部门新增	通过
006	管理员点击部门管理下的部	点击删除部门	部门信息被删除，并且用户可	通过

	部门信息。		选择的部门少了一个	
007	管理员点击部门管理下的部门信息。	点击修改部门，并填写内容	部门信息被修改	通过

5.4.4 用户基本功能

表 5-4 用户基本功能测试

用例编号	前置条件	操作	预期结果	测试结果
001	用户已登录，进入人员管理。	查看人员信息。	可以搜索到人员但是不能修改人员信息。	通过
002	用户已登录，点击进入测试部门管理。	查看部门信息。	部门管理下只有部门信息的选项，可以查看部门信息但是不能修改。	通过
003	用户已登录，点击客户管理。	查看客户信息。	可以对客户进行修改或删除。	通过
004	用户已登录，点击客户管理。	客户管理下有新增客户选项，填写客户基本信息后点击确认。	新增了一名客户。	通过
005	用户已登录，点击项目管理。	点击项目管理下的新增项目选项，填写基本信息后确认。	新增了一个项目。	通过
006	用户已登录，点击项目管理。	点击项目管理下的项目信息查看。	可以查询项目，并且可以看到全部项目。。	通过
007	用户已登录，点击版本管理。	点击版本管理下的新增初始版本，并填写相关信息。	版本信息增加。	通过
008	用户已登录，点击版本管理。	点击版本管理下的初始版本清单。	可以看到所有初始版本信息并进行查找。	通过

5.4.5 文档功能

表 5-5 文档功能测试

用例编号	前置条件	操作	预期结果	测试结果
001	用户已登录，点击项目管理，随意选择一个项目。	点击项目后面的文档按钮。	进入文档管理，有测试计划，测试报告，测试记录等文档。	通过
002	用户已登录，进入测试文档管理页面。	选择一个文档进行上传文件，选择大小为 80MB 以下的文件进行上传的操作。	文件上传成功。	通过
003	用户已登录，进入测试文档管理页面。	选择一个文档进行上传文件，选择大小为 80MB 以上的文件进行上传的操作。	显示文件大小不符合标准。	通过
004	用户已登录，进入测试文档管理页面。	选择一个文档进行上传操作，勾选多个 80MB 以下的文件进行上传。	文件上传成功。	通过
005	用户已登录，进入测试文档管理页面。	选择一个文档对里面的一个文件进行	下载成功。	通过

	试文档管理页面。	下载到桌面文件夹的操作。		
006	用户已登录，进入测试文档管理页面。	选择一个文档对里面的多个文件同时进行下载到桌面文件夹的操作。	下载成功。	通过
007	用户已登录，进入测试文档管理页面。	在项目页面的最后一次修改记录中记录修改信息并点击确认。	修改信息被上传，登录另一个账号可以看到其他人编写的最后一次修改信息。	通过

5.5 测试结论

经过功能测试，该系统的功能大致都符合需求分析的预期。

6. 总结

本次项目大致完成了当初的项目设定的需求和功能，系统的日志管理采用了 Log4j 技术，具有强大的日志功能，可以指定日志的输出级别与输出目的地及每条日志的输出格式等^[14]。在此之前作为一名计算机科学与技术专业的学生，学的大多都是理论知识，对实践的经验少之甚少。在此之前对 springboot 技术不太了解，也从未使用。应用了 springboot 和 MyBatis 框架在 web 系统中，对系统在性能差、复杂度高等问题方面有很大改进^[15]。经过三个多月的学习，从 springMVC 架构开始学习与平台的搭建一直到现在，几乎是每天都在不断学习。互联网这个行业日新月异，如果不保持着终身学习的态度很快就会被时代抛在身后。本系统所使用的技术几乎是两年内比较新的，本系统的主要优点就是对测试文档进行存放，方便项目人员的交流，并且在本系统中不仅可以对项目文档进行管理，还可以看到项目的版本信息，并且同时也可以对公司客户进行管理，在本系统中还可以查看到所以人员信息，方便了同事之间的交流，促进了工作效率的提高。但是本系统的不足之处仍然十分明显，比如说不能够新建文件夹，测试员只能在指定的文件夹内上传信息，并且此系统偏向人工操作功能，如果能将文档自动分类会达到更好的效果。

完成一个项目的过程更是学习的过程，这次的项目给了我一个巨大的启示，从学习的角度来说，更重要的就是勇往直前的态度，不屈不挠的精神。作为一名即将走入社会的学生，也要了解自身在复杂多样的互联网技术中了解到自己哪些知识的不足，在计算机专业技术这条道路上，保持着学习的精神，勇往直前。

致谢

在林岭老师的悉心指导之下，我完成了毕业设计和毕业论文的编写。万分感谢林岭老师对于我们毕业设计的监督和教导。其实初次面对着大量不曾学过甚至不曾听过的技术，内心是充满了抗拒的，但是在这个过程中，林岭老师给予了我巨大的动力以及正确的学习方法，林岭老师的耐心让我觉得没有理由退缩，我必须变得更加强大。在学习和设计的过程中，每当我遇到难题，我总是可以从林岭老师那里得到我需要的东西，数次的视频会议，都让我有恍然大悟的感觉，我觉得老师给了我们方向和方法，只要坚定的学习，那就能成功。林岭老师呕心沥血为我们这些应届毕业生编写了毕业论文编写指南，对待我们的毕业论文，老师甚至比我们自己还用心，几乎考虑到了所有的细节，我真的很幸运，在正确的时间遇到了对的老师。再次发自内心的感谢林岭老师为我们所做的一切，我们走在学习计算机的道路上，您就是我们身边最闪亮的灯塔，引领我们前进。此外还要感谢众位大学老师的教导，长辈的教诲还有同学的帮助和建议。

参考文献

- [1] 布鲁克斯, 汪颖. 人月神话 (32 周年中文纪念版) [M]. 清华大学出版社, 2007.
- [2] William C. Hetzel. The Complete Guide to Software Testing[M]. QED Information Sciences, 1988.
- [3] 王永和, 张劲松, 邓安明, et al. Spring Boot 研究和应用[J]. 信息通信, 2016(10):91-94.
- [4] 黄艳秀. 基于 mybatis 的面向数据库自动生成技术[J]. 河南科技, 2014(04):29-30.
- [5] 陆荣幸, 郁洲, 阮永良, et al. J2EE 平台上 MVC 设计模式的研究与实现%Study and Implementation of MVC Design Pattern on J2EE Platform[J]. 计算机应用研究, 2003, 020(003):144-146.
- [6] 张黎明. 基于 MVC 模式的 Java Web 应用设计%Designing Java Web Applications Based on MVC Mode[J]. 甘肃科技, 2006, 22(6):46-48.
- [7] 徐琼. 基于 MVC 模式的 Struts 框架的研究与实现[D]. 武汉理工大学, 2006.
- [8] 孙莹, 许俊华, 张毅, et al. Web Application of MVC Pattern and its Implementation with Java%MVC 编程模型在 Web 程序中的应用及 Java 实现 [J]. 计算机工程与应用, 2001, 037(017):160-163.
- [9] 阳小兰, 罗明. 基于 Spring+SpringMVC+MyBatis 网上论坛的设计与实现[J]. 黑龙江科技信息, 2016(36):279-280.
- [10] 王文龙, 王武魁. 利用 java 语言实现文件上传功能[J]. 微计算机信息, 2007(33):175-177.
- [11] 黄道斌. 一种基于 SPRING SECURITY 的访问控制方案[J]. 软件导刊, 2011(08):124-125.
- [12] 胡洁萍, 杨树林. 在 Struts2 中实现文件上传的动态跟踪[J]. 北京印刷学院学报(6):74-76.
- [13] 段力军, DUANLi-jun. 软件产品黑盒测试的测试用例设计 [J]. 测试技术学报, 2007, 21(2):160-162.
- [14] 徐雯, 高建华. 基于 Spring MVC 及 MyBatis 的 Web 应用框架研究 [J]. 微型电脑应用, 2012(07):5-8+14.
- [15] 谌湘倩, 狄文辉, 孙冬. 基于 SSH 框架与 AJAX 技术的 JavaWeb 应用开发[J]. 计算机工程与设计, 2009(10):256-258+262.

Design and Implementation of Test Document Management System

LIKE 105032016065 Advisor: LIN Ling

College of Mathematics and Informatics computer science and technology

【Abstract】 Test document management system is a system that collects test documents of projects. On this system, each project has its own test documents. Testers can upload test documents for each project. This system is a project written on idea based on the framework of springboot with object-oriented language Java, and it combines excellent network and artificial intelligence technology to make the whole system design reasonable and run smoothly, which is at the forefront of The Times technology.

【Key words】 Java Web; springboot; mybatis