

非遗项目信息管理系统 设计与实现

完成人：张效凯

目录

一、引言.....	3
1.1 开发背景.....	3
1.2 开发环境.....	3
1.2.1 系统环境.....	3
1.2.1 数据库.....	3
1.2.3 开发工具.....	3
1.2.4 网页开发插件.....	3
二、需求分析.....	3
2.1 功能需求.....	4
2.2 系统参与者及用例.....	4
三、系统总体设计.....	5
3.1 模块设计.....	5
3.1.1 系统设计.....	5
3.1.2 后端设计.....	6
3.1.3 前端设计.....	7
3.2 数据库设计.....	7
四、详细设计.....	10
4.1 后端详细设计.....	10
4.2 前端详细设计.....	10
五、系统代码.....	13
六、系统测试.....	13
6.1 对 UserInfoService 测试.....	13
6.2 对 ICHInfoService 测试.....	14
七、总结和展望.....	15

一、引言

1.1 开发背景

我国文化遗产蕴含着中华民族特有的精神价值、思维方式、想象力，体现着中华民族的生命力和创造力，是各民族智慧的结晶，也是全人类文明的瑰宝。

文化遗产让人感到生命的有限和情感的无限。无论人类的文化和精神强大到什么程度，时间和自然的力量终究是不可抗拒的。与永恒的时间和空间相比，人类不过是沧海一粟，渺小之极。因此，看着那些遗址和废墟，很多人会失落、空虚、惆怅。

保护文化遗产，保持民族文化的传承，是连接民族情感纽带、增进民族团结和维护国家统一及社会稳定的重要文化基础，也是维护世界文化多样性和创造性，促进人类共同发展的前提。加强文化遗产保护，是建设社会主义先进文化，贯彻落实科学发展观和构建社会主义和谐社会的必然要求。文化遗产则是使人类延续这种能力的一个重要途径。所以，人类要延续下去，必须保护文化遗产。正如我们保护环境和濒危生物是基于对大自然的敬畏一样，保护文化遗产，表达的也是对自然的敬畏。只不过，环保所牵涉的自然空间维度的自然，而文保所牵涉到的自然，是时间维度的自然，是能够击退历史的，更宏大的自然力量。

为了更好地保护、传承和发展非物质文化遗产，简化重庆市非遗项目申请流程，增强非遗信息的管理效率，促进非遗政务公开和透明，特开发重庆非遗项目信息管理系统，实现非物质文化遗产的网上宣传、非遗项目网上审批和非遗信息网上管理。

1.2 开发环境

1.2.1 系统环境

操作系统: Windows 10 家庭中文版 2020

处理器: Intel Core i7-8570H CPU @2.20GHz

1.2.1 数据库

数据库: Mysql 5.7.26

1.2.3 开发工具

IDE: IDEA (项目开发)、VSCode (网页设计)、Postman (发送 http 请求, 系统测试)、Navicat (数据库可视化操作页面)、Chrome 浏览器 (项目运行)

Web 开发集成: java 12.0.1 2019-04-06、SpringBoot 2.3.0 (web 开发框架)、thymeleaf (网页开发模板引擎)、maven (项目配置)、junit (测试)、git (版本控制) 等

1.2.4 网页开发插件

Jquery、AdminLTE-3.0.4、fontawesome、Bootstrap、Bootstrap-Table、layerui、datatables-bs4 等

二、需求分析

2.1 功能需求

题目要求实现非遗信息（非遗名称、所在类别、所属民族、所属地区、当前传承人、非遗级别、涉及器具、具体描述）的增删改查。除此之外，扩展的功能需求有安全访问和组合/模糊查询。

除题目要求外，一个完成的信息管理系统是面向不同用户的，如用户可以发起非遗申请，管理员可以审批非遗申请。还有消息通知系统和用户个人信息管理系统，也是基本不可少的。

基于以上考虑，系统应具有的功能有：

①**访问控制**：对于少数数据查询和展示页面，任何人都可访问；对于涉及到数据修改的页面，需要对其访问进行拦截，验证用户身份和权限后，才可以访问。

②**数据操作**：非遗信息和其他信息的增删改查，以及模糊查询等。只有管理员有对数据修改的权限。

③**用户个人信息修改**：如修改名字、密码等。

④**非遗申请和审批**：用户可以申请非遗项目，管理员可以审批非遗项目申请。

⑤**消息通知**：如系统公告、非遗项目申请审批结果通知等。

2.2 系统参与者及用例

整个系统的参与者，根据各自的权限和业务开展需求不同，可以分为四类账户，分别是：游客（tourist）、用户（user）、办事职员（staff）、系统管理员（manager）。系统通过显示不同的页面，控制四类用户的业务逻辑：

*游客（tourist）

此类用户只可以访问首页和非遗信息列表展示页面，没有其他的业务流程。游客除了请求两个页面之外，与系统没有其他的交互。因为未登录，所以任何访问其他页面的尝试都会被拦截，转到登录页面。

*用户（user）

此类用户要参与到用户个人信息管理系统、消息通知系统、非遗申请和审批系统（作为申请者的身份）；此外，游客可以访问的首页和非遗信息列表展示页面，这类用户也能访问。

*办事职员（staff）

此类用户要参与到用户个人信息管理系统、消息通知系统、非遗申请和审批系统（作为审批者的身份）；此外，游客可以访问的首页和非遗信息列表展示页面，这类用户也能访问。

*系统管理员（manager）

这类用户是系统的管理者，除了个人信息管理系统、消息通知系统外，系统管理员可以查看系统的所有信息，也就是所有的数据库表（将在 3.2 中详细说明）。

他们对应的用例图如下：

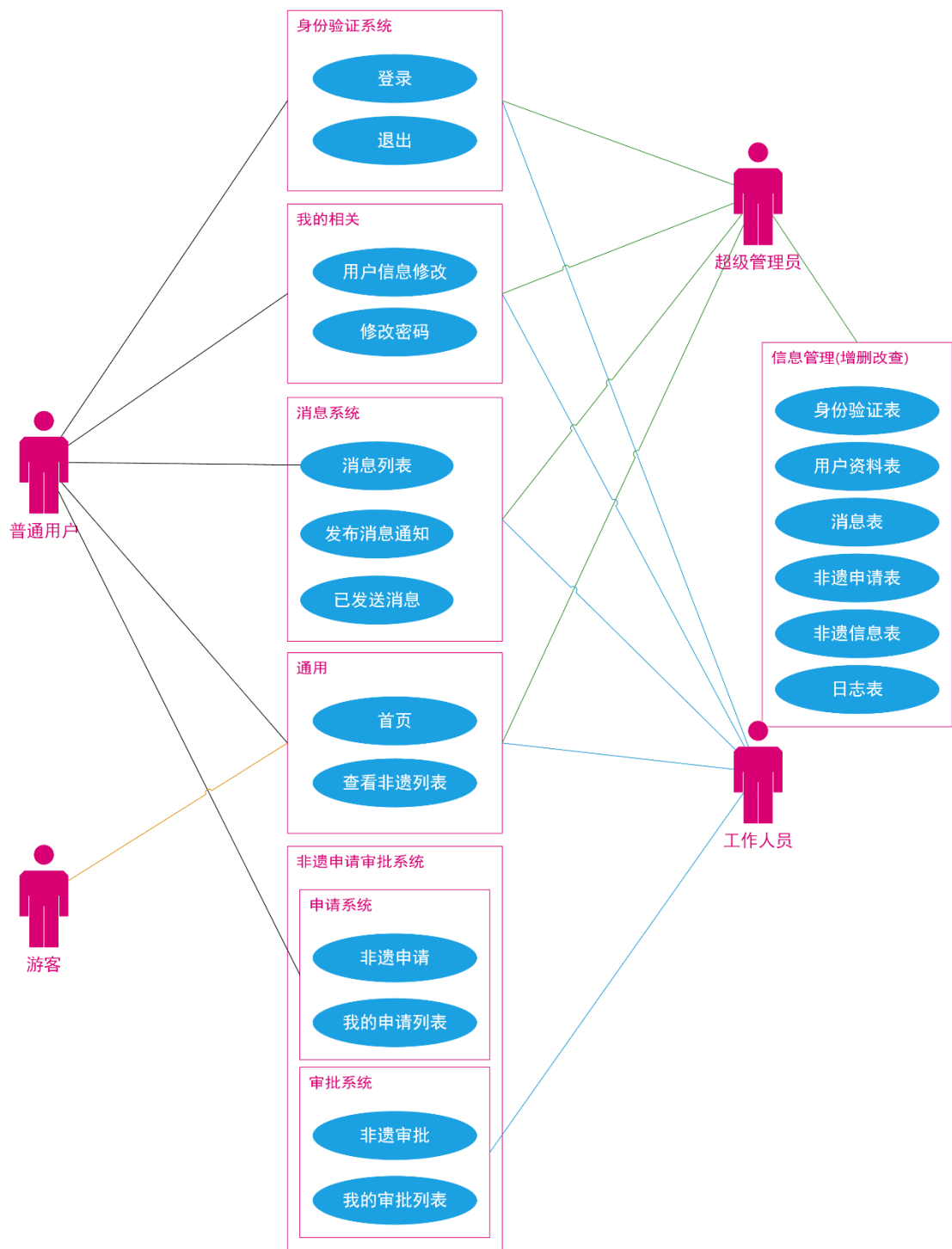


图 2-2-1 系统用例图

三、系统总体设计

3.1 模块设计

3.1.1 系统设计

在需求分析中，从功能的角度对整个系统模块做了划分。在此部分，将会从程序结构的角度入手，对整个系统的开发模块做大致的介绍。整个系统采用 MVC

的架构，业务逻辑、数据、界面显示分离，整体架构如图 3-1-1 所示：

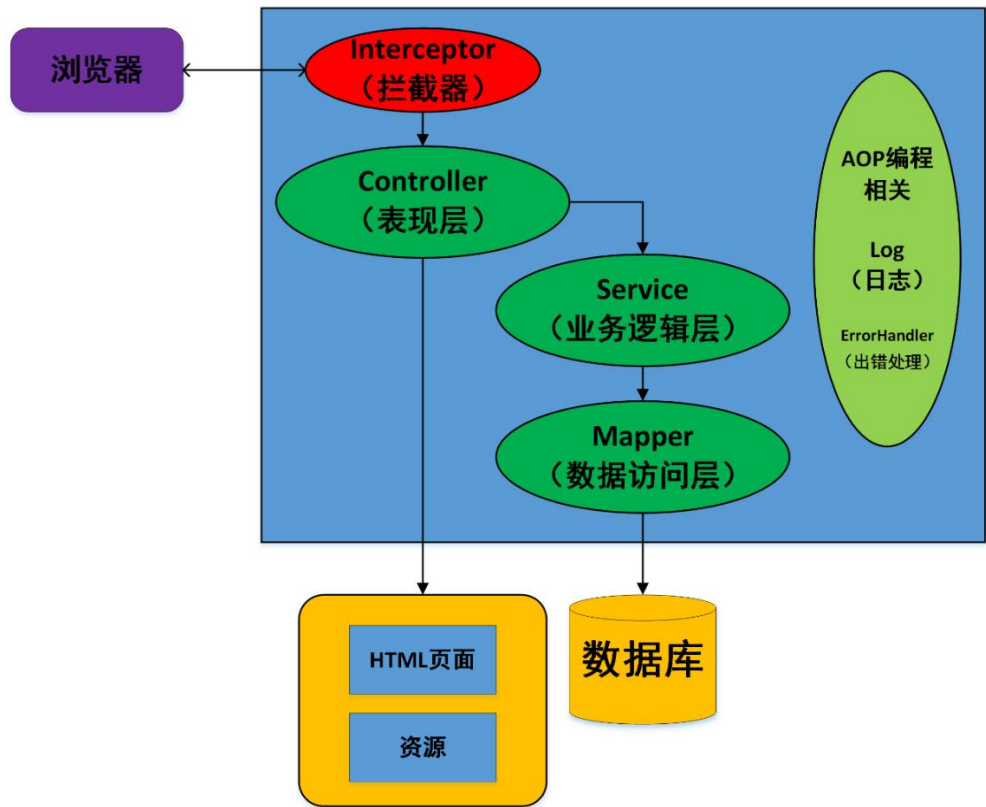


图 3-1-1 系统整体架构

3.1.2 后端设计

后端定义类的数量众多，所以在此小节隐去类的属性和方法，只对后端结构做大致的介绍，详细介绍放在下一章节。后端代码的类图为：

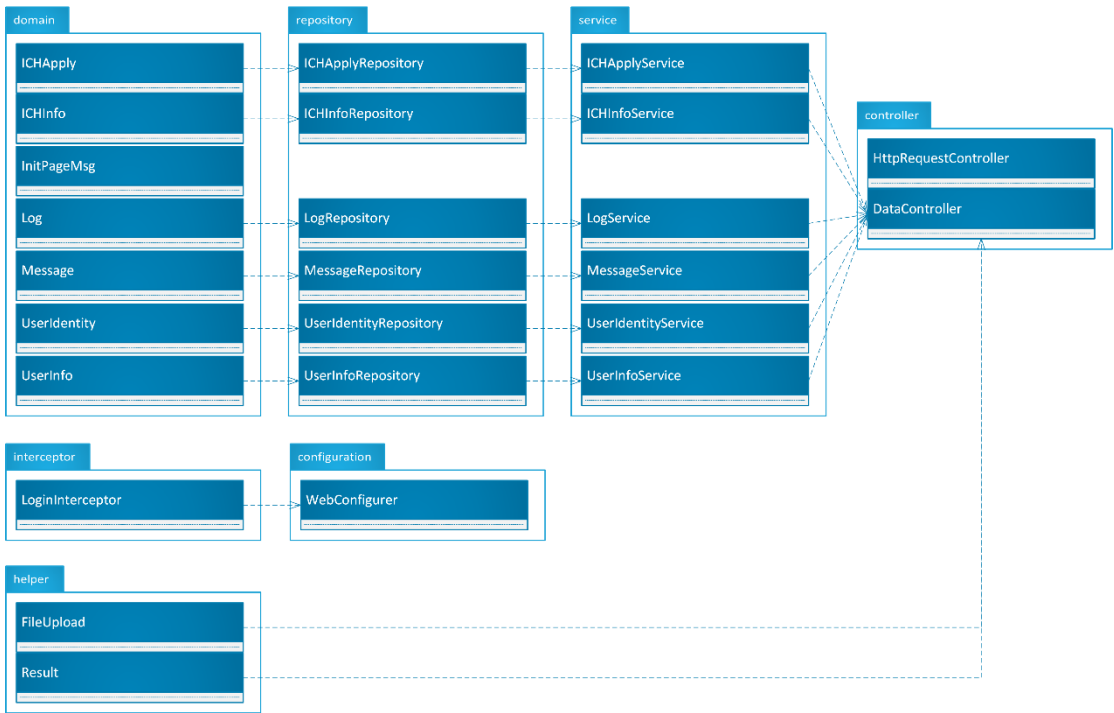


图 3-1-2 后端类图

domain 包

定义了 7 个类，对应数据库的六个表，剩余一个类是页面初始化所需的信息（如人员 ID、消息数量等）。

repository 包

对应架构的数据访问层，实现 JPA 接口，用于访问数据库。

service 包

对应架构的业务逻辑层，对 repository 查到的数据进行处理、组装等操作，最后呈现给 controller 层。

controller 包

对应架构的表现层，接受用户请求，返回相应的数据。

interceptor 包

拦截器设置，安全访问相关。

configuration 包

运行配置，如拦截器、日期与时间格式等。

helper 包

一些辅助类，如 Result（程序统一返回结果）、FileUpload（文件上传）等。

3.1.3 前端设计

前端对于四种不同的用户，显示不同的页面。系统前端总体架构的活动图为：

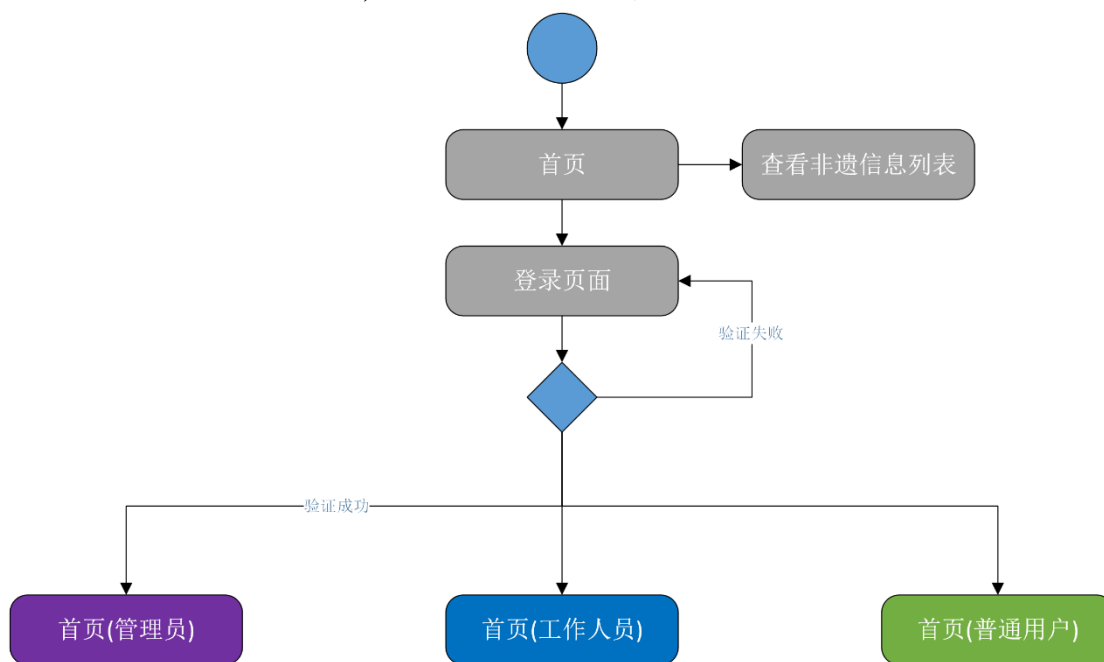


图 3-1-3 前端活动图

更详细的活动图将在详细设计中展示。

3.2 数据库设计

通过上述对用例图的分析，已经大致了解项目功能架构，本小节从数据库角度，阐述数据的组织架构方式，系统 E-R 图如图 3-2-1 所示：

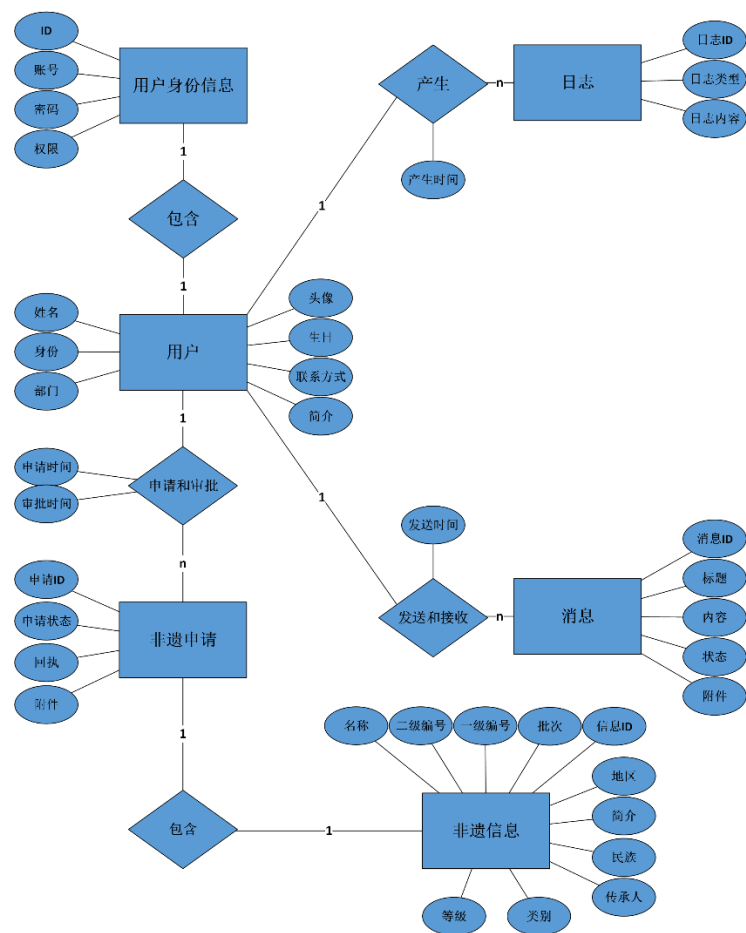


图 3-2-1 系统 E-R 图

进一步将 E-R 图转化为数据库表：

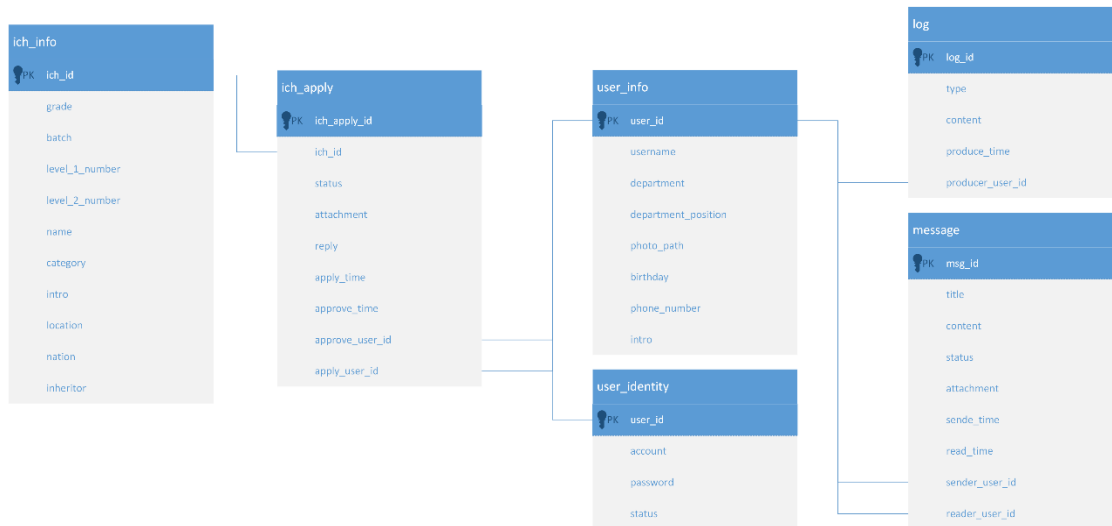


图 3-2-2 数据库表

相应的各个表的结构为：


名	类型	长度	小数点	不是 null	虚拟	键
ich_apply_id	int	7	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	 1
ich_id	int	7	0	<input type="checkbox"/>	<input type="checkbox"/>	
status	int	7	0	<input type="checkbox"/>	<input type="checkbox"/>	
attachment	varchar	255	0	<input type="checkbox"/>	<input type="checkbox"/>	
reply	varchar	255	0	<input type="checkbox"/>	<input type="checkbox"/>	
apply_time	datetime	6	0	<input type="checkbox"/>	<input type="checkbox"/>	
approve_time	datetime	6	0	<input type="checkbox"/>	<input type="checkbox"/>	
apply_user_id	int	7	0	<input type="checkbox"/>	<input type="checkbox"/>	
approve_user_id	int	7	0	<input type="checkbox"/>	<input type="checkbox"/>	

图 3-2-3 ich_apply 数据表

名	类型	长度	小数点	不是 null	虚拟	键
ich_id	int	8	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	 1
grade	varchar	32	0	<input type="checkbox"/>	<input type="checkbox"/>	
batch	varchar	32	0	<input type="checkbox"/>	<input type="checkbox"/>	
level1number	varchar	32	0	<input type="checkbox"/>	<input type="checkbox"/>	
level2number	int	8	0	<input type="checkbox"/>	<input type="checkbox"/>	
name	varchar	1024	0	<input type="checkbox"/>	<input type="checkbox"/>	
category	varchar	1024	0	<input type="checkbox"/>	<input type="checkbox"/>	
intro	varchar	2048	0	<input type="checkbox"/>	<input type="checkbox"/>	
location	varchar	1024	0	<input type="checkbox"/>	<input type="checkbox"/>	
nation	varchar	1024	0	<input type="checkbox"/>	<input type="checkbox"/>	
inheritor	varchar	15	0	<input type="checkbox"/>	<input type="checkbox"/>	

图 3-2-4 ich_info 数据表


名	类型	长度	小数点	不是 null	虚拟	键
log_id	int	7	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	 1
type	int	7	0	<input type="checkbox"/>	<input type="checkbox"/>	
content	varchar	255	0	<input type="checkbox"/>	<input type="checkbox"/>	
produce_time	datetime	6	0	<input type="checkbox"/>	<input type="checkbox"/>	
producer_user_id	int	7	0	<input type="checkbox"/>	<input type="checkbox"/>	

图 3-2-5 log 数据表


名	类型	长度	小数点	不是 null	虚拟	键
msg_id	int	7	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	 1
title	varchar	31	0	<input type="checkbox"/>	<input type="checkbox"/>	
content	varchar	1023	0	<input type="checkbox"/>	<input type="checkbox"/>	
status	int	7	0	<input type="checkbox"/>	<input type="checkbox"/>	
attachment	varchar	63	0	<input type="checkbox"/>	<input type="checkbox"/>	
send_time	datetime	6	0	<input type="checkbox"/>	<input type="checkbox"/>	
read_time	datetime	6	0	<input type="checkbox"/>	<input type="checkbox"/>	
sender_user_id	int	7	0	<input type="checkbox"/>	<input type="checkbox"/>	
reader_user_id	int	7	0	<input type="checkbox"/>	<input type="checkbox"/>	

图 3-2-6 message 数据表

名	类型	长度	小数点	不是 null	虚拟	键
user_id	int	31	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	 1
account	varchar	63	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
password	varchar	63	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
status	int	7	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	

图 3-2-7 user_identity 数据表

名	类型	长度	小数点	不是 null	虚拟	键	注
user_id	int	7	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	1	
username	varchar	31	0	<input type="checkbox"/>	<input type="checkbox"/>		
department	varchar	31	0	<input type="checkbox"/>	<input type="checkbox"/>		
department_position	varchar	31	0	<input type="checkbox"/>	<input type="checkbox"/>		
photo_path	varchar	63	0	<input type="checkbox"/>	<input type="checkbox"/>		
birthday	date	0	0	<input type="checkbox"/>	<input type="checkbox"/>		
phone_number	varchar	31	0	<input type="checkbox"/>	<input type="checkbox"/>		
intro	varchar	1023	0	<input type="checkbox"/>	<input type="checkbox"/>		

图 3-2-8 user_info 数据表

四、详细设计

4.1 后端详细设计

因为后端类数目众多，但大体上类的结构相似。整个后端的结构在前面章节已经说明，在这里我们选择“管理员-非遗信息管理”这个模块，来展示后端代码的详细设计，类图如下：

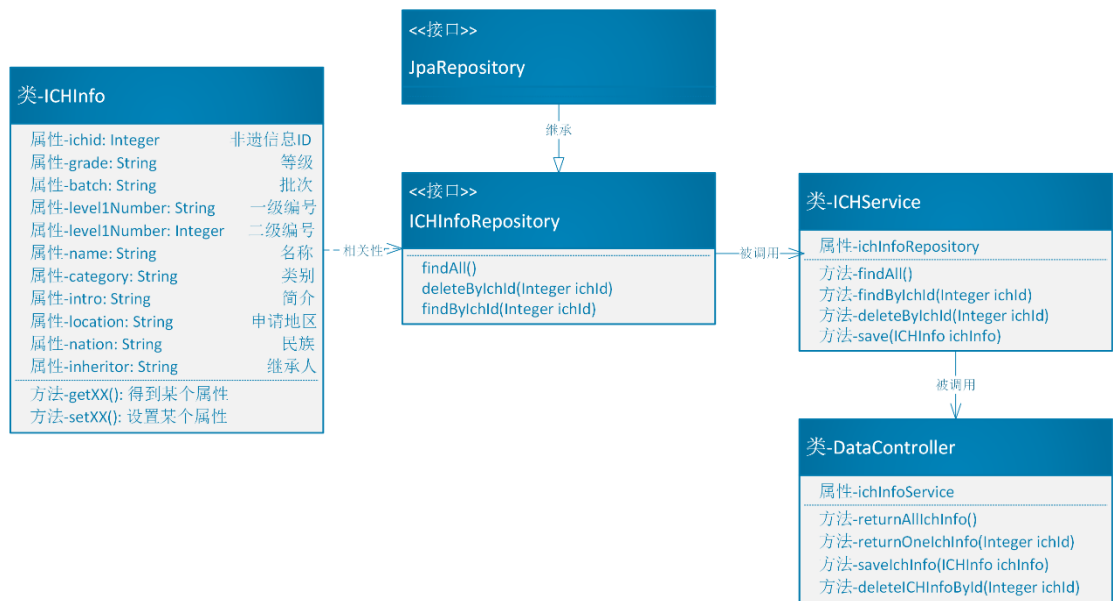


图 4-4-1 非遗信息管理子模块（后端）详细类图

4.2 前端详细设计

在 3.1.3 中，我们给出了系统前端的总体架构，在此给出前端的详细架构（活动图的形式）：

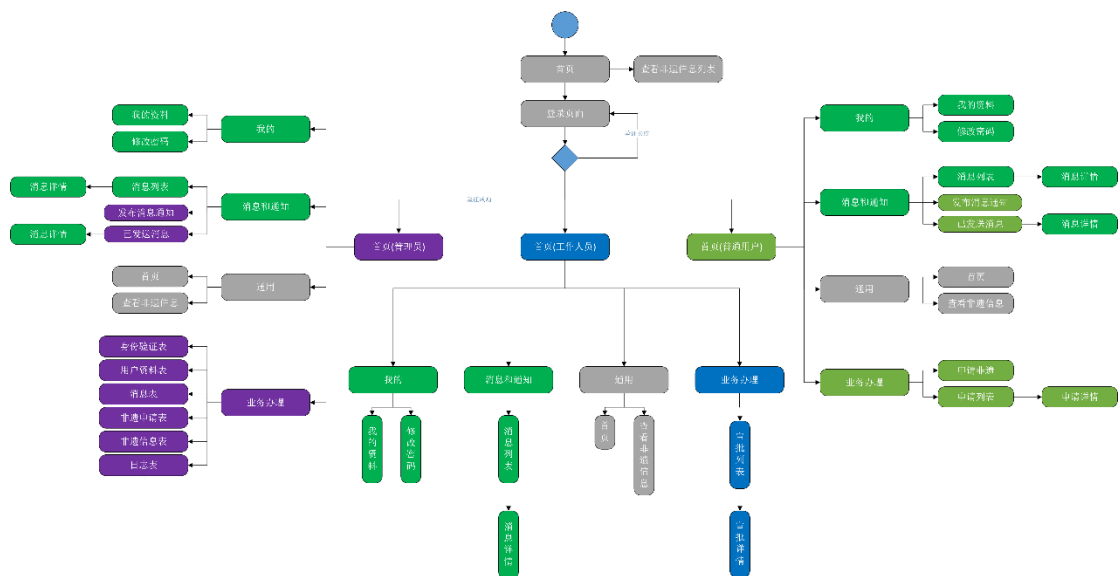
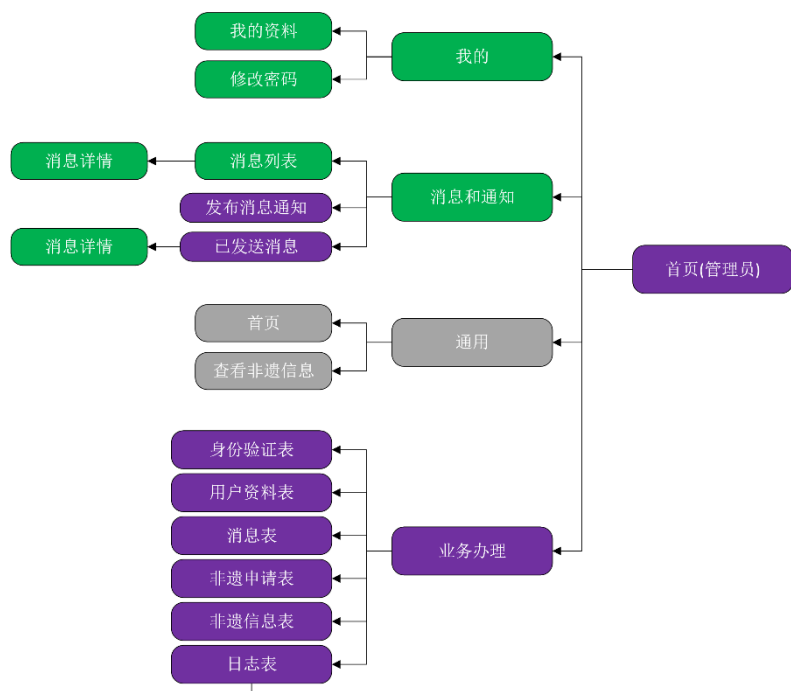


图 4-2-1 前端详细活动图



4-2-2 前端管理员活动图

我的-我的资料：查看和修改个人资料。

我的-修改密码：修改登录密码。

消息和通知-消息列表：接收到的消息。

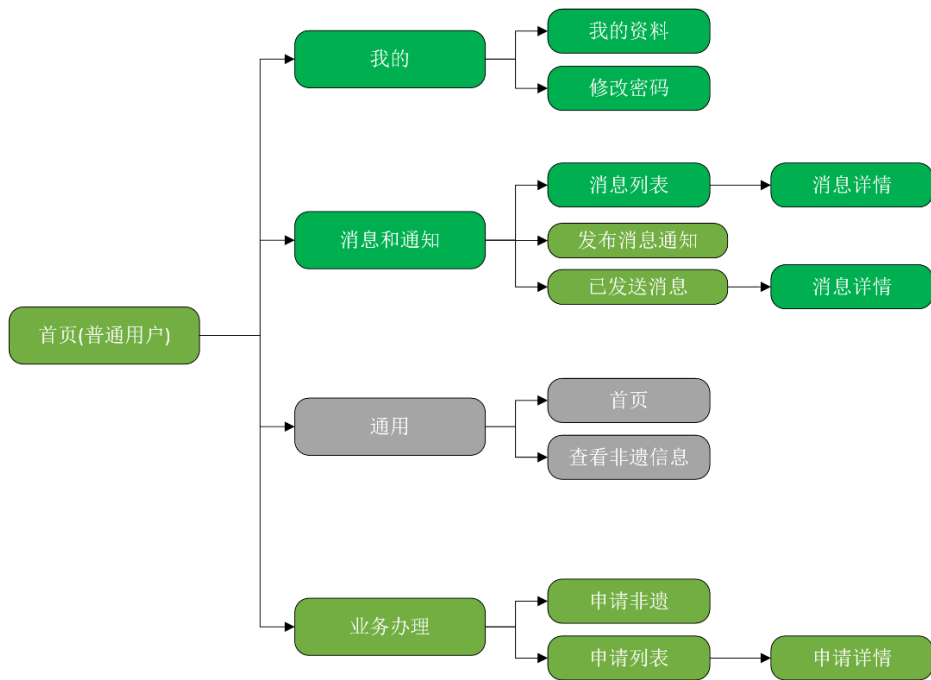
消息和通知-发布消息通知：给其他用户发送消息通知。

消息和通知-已发送通知：之前发送的消息。

通用-首页：首页。

通用-查看非遗信息：展示非遗信息列表。

业务办理-六个表：对数据库的六个表进行增删改查操作。



4-2-3 前端用户活动图

我的-我的资料：查看和修改个人资料。

我的-修改密码：修改登录密码。

消息和通知-消息列表：接收到的消息。

消息和通知-发布消息通知：给其他用户发送消息通知。

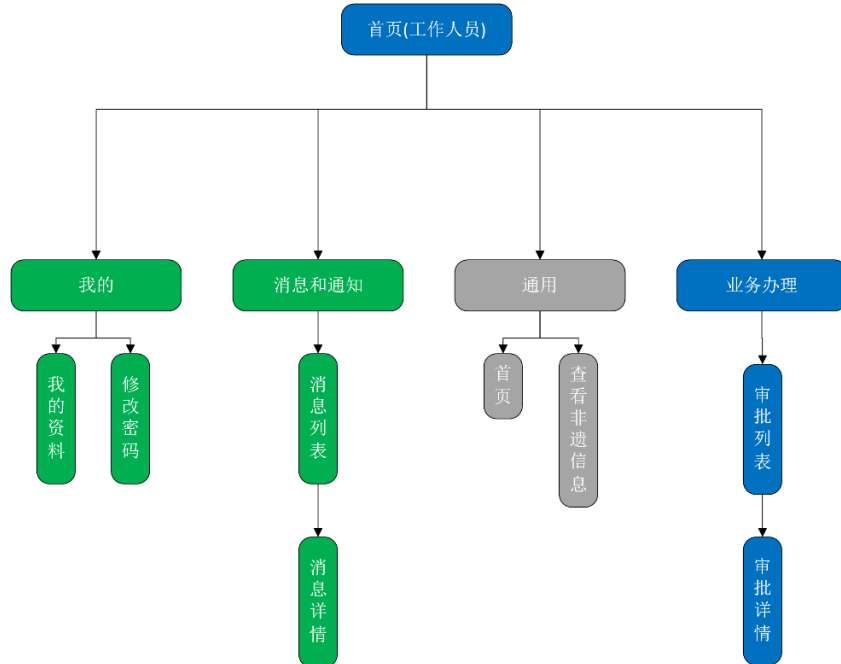
消息和通知-已发送通知：之前发送的消息。

通用-首页：首页。

通用-查看非遗信息：展示非遗信息列表。

业务办理-申请非遗：申请非遗项目。

业务办理-申请列表：查看自己已申请非遗项目。



4-2-4 前端工作人员活动图

我的-我的资料：查看和修改个人资料。

我的-修改密码：修改登录密码。

消息和通知-消息列表：接收到的消息。

通用-首页：首页。

通用-查看非遗信息：展示非遗信息列表。

业务办理-审批列表：查看和审批用户发送的非遗项目申请。

五、系统代码

代码太多，详见附件一。

六、系统测试

6.1 对 UserInfoService 测试

测试代码：

```
package code.service;

import code.DemoApplication;
import code.domain.UserInfo;
import org.junit.jupiter.api.Assertions;
import org.junit.jupiter.api.Test;
import org.junit.runner.RunWith;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.boot.test.context.SpringBootTest;
import org.springframework.test.context.junit4.SpringRunner;

@RunWith(SpringRunner.class)
@SpringBootTest(classes = DemoApplication.class, webEnvironment =
SpringBootTest.WebEnvironment.RANDOM_PORT)
class UserInfoServiceTest {
    @Autowired
    UserInfoService userInfoService;

    @Test
    void findUserInfoById() {
        Integer testUserId = 1;
        UserInfo userInfo =
userInfoService.findUserInfoById(testUserId);
        Assertions.assertEquals(userInfo.getUserId(), testUserId);
//验证是否找到用户
    }
    @Test
    void save() {
        String testName = "testName";
```

```

        UserInfo userInfo = userInfoService.findUserInfoById(1);
        userInfo.setUsername(testName); //改名
        userInfoService.save(userInfo); //保存

        userInfo = userInfoService.findUserInfoById(1);
        Assertions.assertEquals(userInfo.getUsername(), testName);
//验证是否修改
    }
}

```

6.2 对 ICHInfoService 测试

测试代码:

```

package code.service;

import code.DemoApplication;
import code.domain.ICHInfo;
import org.junit.jupiter.api.Assertions;
import org.junit.jupiter.api.Test;
import org.junit.runner.RunWith;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.boot.test.context.SpringBootTest;
import org.springframework.test.context.junit4.SpringRunner;
import java.util.List;

@RunWith(SpringRunner.class)
@SpringBootTest(classes = DemoApplication.class, webEnvironment =
SpringBootTest.WebEnvironment.RANDOM_PORT)
class ICHInfoServiceTest {
    @Autowired
    ICHInfoService ichInfoService;

    @Test
    void findAll() {
        Integer currentCount = 716;
        List<ICHInfo> ichInfo = ichInfoService.findAll();
        Assertions.assertEquals(ichInfo.size(), currentCount); //
测试是否全部取到
    }

    @Test
    void findByIchId() {
        ICHInfo ichInfo = ichInfoService.findByIchId(10);
        Assertions.assertEquals(ichInfo.getIchId(), 10); //测试是
否取到正确的信息
    }
}

```

```

        ichInfo = ichInfoService.findByIchId(100);
        Assertions.assertEquals(ichInfo.getIchId(), 100); //测试是否
        否取到正确的信息
        ichInfo = ichInfoService.findByIchId(1000);
        Assertions.assertNull(ichInfo); //测试是否取到正确的信息
    }
    @Test
    void saveThenDeleteByIchId() {
        ICHInfo ichInfo = new ICHInfo();
        ichInfo.setName("测试 name");
        Integer ichId = ichInfoService.save(ichInfo);
        Assertions.assertNotEquals(ichId, 0); //返回零, 证明存储失败
        Boolean result = ichInfoService.deleteByIchId(ichId);
        Assertions.assertTrue(result); //返回 true, 证明操作成功
    }
}

```

七、总结和展望

这次大作业本来计划选用 QT 开发, 因为自己比较熟悉。但恰巧这学期学了 Web, 可以很好的跟软件工程结合起来, 这两门科目都能学到, 所以就大作业就选择了 Web 实现。

整个项目断断续续耗时大约两周, 一开始的时候很困难, 但最后还是学到了很多: ①前期需求分析时, 复习了之前数据库学习的知识, 如 ER 图、数据库表设计等, 同时应用了软件工程学到的新知识, 如 UML 部分的用例图、活动图、类图等。前者是面向过程的思想, 后者是面向对象的思想, 两者结合可以把项目架构的很清楚。②软件开发过程中, 深入理解了 MVC 开发模式、学习了 SpringBoot 的三层模型、了解了 AOP 面向切面编程思想等。同时对于 IDEA、Postman、VSCode 这些 IDE 的使用, 更熟练了。此外 Git 版本控制这些大项目协作开发工具, 也有了初步的认识。③Web 开发的基础知识、网页的基础知识(css、js、html)、maven 项目配置管理等方面的内容, 也有了基本的了解。④后期测试时, 应用了软件测试的相关知识。

项目实现了题目要求的增删改查(普通查询、模糊查询)和安全访问, 在此基础上扩展了消息系统、审批系统、用户/身份资料系统等, 基本实现了一个具有基本功能、页面美观的网页。

因为时间有限, 考研期限临近, 况且已经远超题目的要求, 消息系统、审批系统只是做了基本的框架。自己已经学习整个开发流程, 剩下的只是一些“体力活”了, 所以就不再完善。考研结束后可以继续对系统进行开发。

学无止境, 加油加油~