

5 系统功能实现

5.1 用户模块实现

用户在使用该平台时需要先进行登录或注册操作，其登录注册流程图如图 5-1 所示。

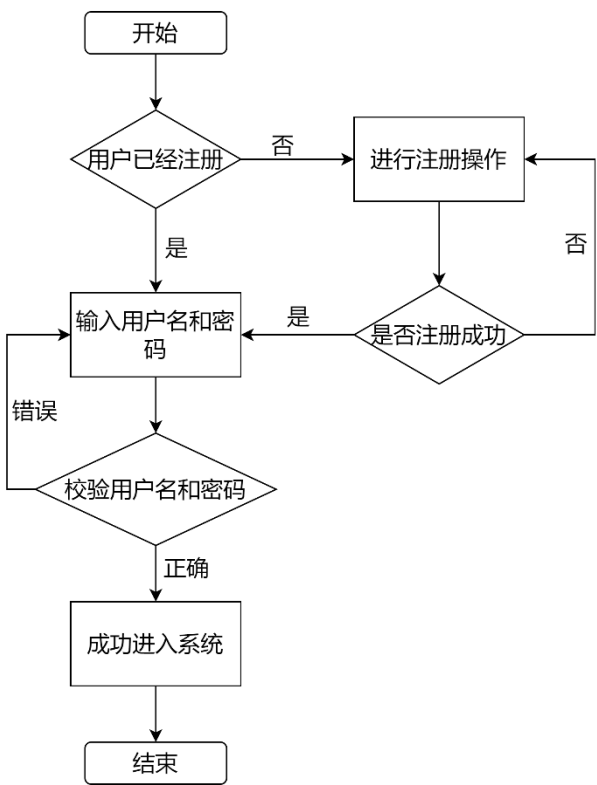



图 5-1 登录注册流程图

5.1.1 注册

用户第一次使用该平台，需要先用邮箱注册一个新账号。用户需要在注册页面上先填写用户名和邮箱，然后点击发送验证码，平台会将发送 5 分钟内有效的验证码到指定邮箱，以验证邮箱是否可用。如果邮箱已经存在，平台会提示邮箱已被注册。注册界面如图 5-2 所示。



The registration form for EASYWORK is displayed within a white box. At the top, the EASYWORK logo is shown. Below it, the title '注册' (Registration) is centered. The form includes fields for '用户名' (Username) with a hint to use a real name, '邮箱' (Email), '验证码' (Verification Code), and '密码' (Password). A '注册' (Register) button is at the bottom, along with a link to '或 登录' (or Login). A red border highlights the verification code and password fields.

注册

用户名 为了使您在工作中方便沟通, 建议使用姓名

李正航测试 5/32

邮箱

1425176577@qq.com

验证码

请输入验证码 已发送, 45秒后重新获取

请输入验证码

密码

请输入密码

请输入密码

注册

或 登录

图 5-2 注册界面

接收邮箱验证码界面如图 5-3 所示。



图 5-3 接收邮箱验证码界面

注册功能的关键代码如下所示。

1) 后端注册功能的关键代码。

```
1. public boolean register(UserRegisterDto dto) {  
2.    // 验证邮箱格式是否正确  
3.    checkEmail(dto.getEmail());  
4.    // 判断邮箱是否被注册  
5.    Integer count = this.getBaseMapper()
```

```

6.         .selectCount(Wrappers.<User>lambdaQuery().eq(User::getEmail, dto.getEmail()
        l()));
7. if (count > 0) {
8.     throw CommonException.builder().resultCode(UserErrorEnum.EMAIL_REGISTERED).b
        uild();
9. }
10. // 判断验证码是否正确
11. String codeInRedis = JedisUtil.getStringValue(dto.getEmail());
12. if (!dto.getCode().equals(codeInRedis)) {
13.     throw CommonException.builder().resultCode(UserErrorEnum.VERIFY_CODE_ERROR).
        build();
14. }
15. // 验证成功, 删除验证码
16. JedisUtil.del(dto.getEmail());
17. return true;
18. }

```

2) 发送验证码并设置有效期为 5 分钟的关键代码。

```

1. public boolean sendVerifyCode(VerifyEmail verifyEmail) {
2.     // 验证邮箱格式是否正确
3.     checkEmail(verifyEmail.getEmail());
4.     // 生成验证码
5.     String code = EmailUtil.generateCode();
6.     // 先发送邮件, 如果邮件发送过程中出现错误, 验证码也不会保存
7.     Context context = new Context();
8.     context.setVariable("code", code);
9.     try {
10.         mailService.sendMail(verifyEmail.getEmail(), Constant.PROJECT_NAME, veri
            fyEmail.getType(), context);
11.     } catch (MessagingException e) {
12.         throw CommonException.builder().resultCode(EmailErrorEnum.EMAIL_SEND_ERR
            OR).build();
13.     }
14.     // 再将验证码保存至 redis, key 为邮箱, 保存 5 分钟
15.     JedisUtil.setStringValue(verifyEmail.getEmail(), code, 300);
16.     log.info("邮箱 ==> [{}], 验证码 ==> [{}]", verifyEmail.getEmail(), code);
17.     return true;
18. }

```

5.1.2 登录

如果已经拥有平台账号, 就可以选择登录进入系统。用户需要输入邮箱和密码, 点击登录按钮时, 系统会检测邮箱和密码是否为空或者错误, 如果都正确, 系统会提示登录成功并跳转到平台首页, 如果密码错误则会提示错误。登录成功

后，系统会将该用户的基本信息返回给前端保存，并生成一个 token 保存在 Cookie，前端进行页面跳转和向后端请求接口都需要带上 token，若 token 失效或丢失，则需要重新登录。登录界面如图 5-4 所示。



图 5-4 登录界面

登录功能的关键代码如下所示，后端对邮箱和密码进行验证。

```
1. public RestResponse<UserDto> findUser(String loginName, String password) {
2.     if (StringUtils.isEmpty(loginName)) {
3.         return RestResponse.failed("请输入邮箱!");
4.     }
5.     if (StringUtils.isEmpty(password)) {
6.         return RestResponse.failed("请输入密码!");
7.     }
8.     LambdaQueryWrapper<User> wrapper = Wrappers.<User>lambdaQuery();
9.     wrapper.eq(User::getEmail, loginName);
10.    User user = this.getOne(wrapper);
11.    if (user == null) {
12.        return RestResponse.failed("邮箱不存在!");
13.    }
14.    if (password.equals(JasyptEncryptorUtils.decode(user.getPassword()))) {
15.        UserDto userDto = userParamMapper.entity2Dto(user);
16.        return RestResponse.ok(userDto);
17.    }
18.    return RestResponse.failed("密码错误!");
}
```

登录成功后，进入系统主界面，系统主界面如图 5-5 所示。

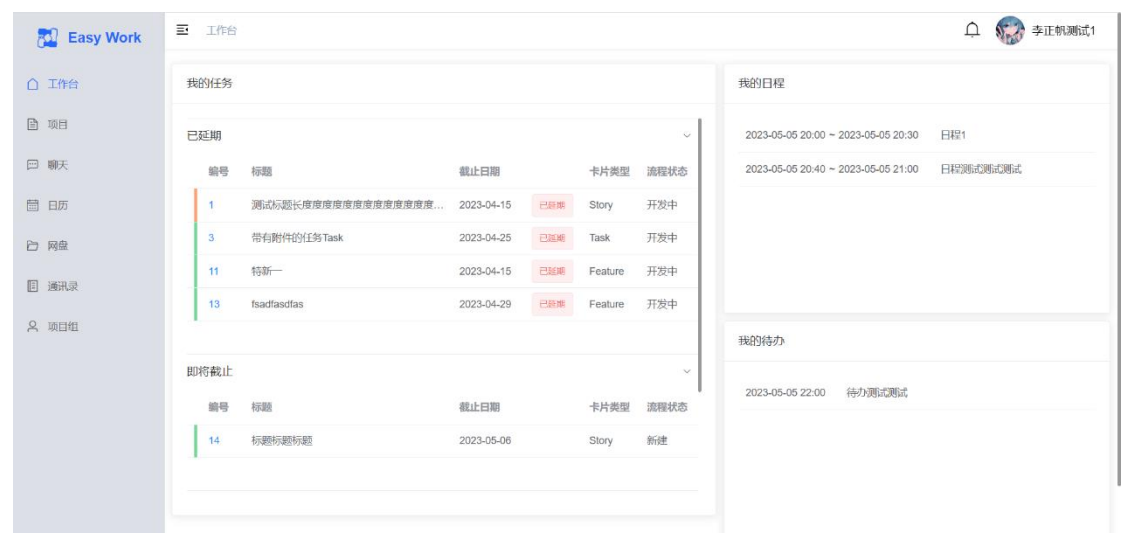


图 5-5 系统主界面

5.2 文件模块实现

文件模块的实现主要是为整个系统提供两个主要接口：上传和下载接口。文件的存储是采用阿里云的对象存储 OSS，上传接口将文件内容存储到 OSS 中，由于 OSS 中文件名称不能重复，因此保存在 OSS 中的文件采用 UUID 进行命名，上传成功后再将文件的元信息如文件原始名称、文件大小、文件的访问路径等存储到 MySQL 数据库，MySQL 存储文件元信息如图 5-6 所示。其他模块例如项目模块中，工作项的附件功能，存储的是 MySQL 中文件元信息的主键 id。之后当需要下载时，根据文件 id 请求文件模块的下载接口，即可完成下载。

对象 t_file_meta @easy_work (mysql) - 表					
开始事务 文本 筛选 排序 导入 导出					
id	file_name	original_filename	file_size	location	de
1630963882688786434	fdd5e8cd9484b86b83156f545775c5f.png	(Null)	37198	https://easywork23.oss-cn-shenzhen.aliyuncs.com/atta	(N
1631128713224048641	32ed2f60567345bf844a6ef3629c17d3.png	(Null)	179516	https://easywork23.oss-cn-shenzhen.aliyuncs.com/atta	(N
1631130363649167362	13d8fe60b48f4e82823905105efb5b65.png	mvc.png	179516	https://easywork23.oss-cn-shenzhen.aliyuncs.com/atta	(N
1631130716645986306	0860e83c50c7451182d432d3e5cf176a.png	mvc.png	179516	https://easywork23.oss-cn-shenzhen.aliyuncs.com/atta	(N
1631130893754658817	c7f8b72dabdc4d18a286308941bb23a7.png	mvc.png	179516	https://easywork23.oss-cn-shenzhen.aliyuncs.com/atta	(N
1631131678496374786	18e316021f5d4648aba8a2db6652d61f.zip	毕业设计-EasyWork.z	2707	https://easywork23.oss-cn-shenzhen.aliyuncs.com/atta	(N
1632229890837782529	7747e87cbebd4bb8ac5b3e1b45d7cd2.png	用户.png	40439	https://easywork23.oss-cn-shenzhen.aliyuncs.com/atta	(N
1632956626374279169	4c2ddb5ffb92426f810ae95c36e40b3d.png	JVM.png	37198	https://easywork23.oss-cn-shenzhen.aliyuncs.com/atta	(N
1634044901470105602	73065ab17e4748c88d19ef293f2fe03c.png	JVM.png	37198	https://easywork23.oss-cn-shenzhen.aliyuncs.com/atta	(N
1634046706467860482	bf0ae5281b174163ab634c0429bbe23b.png	file.png	46522	https://easywork23.oss-cn-shenzhen.aliyuncs.com/atta	(N
1634047181158215681	88c819982f004fc9bf920daefbbe32e1.png	file.png	57056	https://easywork23.oss-cn-shenzhen.aliyuncs.com/atta	(N
1634047516962582530	7a0e43f3d3024e1da5b6f2b4dac11a3e.png	file.png	57056	https://easywork23.oss-cn-shenzhen.aliyuncs.com/atta	(N
1634049280034074626	03c05ac3630d4424982cd699448de1f7.png	file.png	84269	https://easywork23.oss-cn-shenzhen.aliyuncs.com/atta	(N
163404953339117569	2452281b4f042a68763aef0a2638609.png	file.png	57056	https://easywork23.oss-cn-shenzhen.aliyuncs.com/atta	(N
1634049687296798721	bf182e92edc34281a231eace6ad55684.png	file.png	57056	https://easywork23.oss-cn-shenzhen.aliyuncs.com/atta	(N
1634049853802278914	14b52b25c9b144379d5da287c3870f65.png	file.png	46522	https://easywork23.oss-cn-shenzhen.aliyuncs.com/atta	(N
1634049910702206978	f858ee04afbd4cccb21d27a54ddb7e84.png	file.png	68270	https://easywork23.oss-cn-shenzhen.aliyuncs.com/atta	(N
1634050094182035458	34ab22f5306640f4b7b3e8b0603604c3.png	file.png	65138	https://easywork23.oss-cn-shenzhen.aliyuncs.com/atta	(N
1634063615779270658	c8039500d7a94e52857e37cae3a5ac03.png	file.png	59229	https://easywork23.oss-cn-shenzhen.aliyuncs.com/atta	(N
1634063857694142465	bcd9c944704345e2984fc7bd55bde2c2.png	file.png	59229	https://easywork23.oss-cn-shenzhen.aliyuncs.com/atta	(N
1634064051198357506	24e58b5534d943f7b9956fca30801f61.png	file.png	59229	https://easywork23.oss-cn-shenzhen.aliyuncs.com/atta	(N
1634064479248052225	f11c433b106454ab507f90962ece8bd.png	file.png	59229	https://easywork23.oss-cn-shenzhen.aliyuncs.com/atta	(N
1634092289719136257	2f83c2403dd345baba6d65e8b316558f.png	file.png	74475	https://easywork23.oss-cn-shenzhen.aliyuncs.com/atta	(N
1634092326406713345	d36bfd9941f4502b9008c078fea69c9.png	file.png	62867	https://easywork23.oss-cn-shenzhen.aliyuncs.com/atta	(N
1634092875034259458	91396b03364a444491b2f7fc0758977c.png	file.png	99947	https://easywork23.oss-cn-shenzhen.aliyuncs.com/atta	(N
1634092943095230466	bb3a163e103f4161908d5bcd074ea2b8.png	file.png	64454	https://easywork23.oss-cn-shenzhen.aliyuncs.com/atta	(N
1634126394343129090	db02c8745a314524b18cbf5b9e310730.png	file.png	59229	https://easywork23.oss-cn-shenzhen.aliyuncs.com/atta	(N
1635115686590803969	e540756cb72f4164b8647bae5bfb3f4d.png	file.png	247067	https://easywork23.oss-cn-shenzhen.aliyuncs.com/atta	(N
1638386928785625089	27a45da7481b4c9cad6843ee09d8d902.png	JVM.png	37198	https://easywork23.oss-cn-shenzhen.aliyuncs.com/atta	(N

图 5-6 文件元信息存储

上传和下载接口的关键代码如下所示。

1) 上传功能。

```

1. public FileMetaDto upload(MultipartFile file, String directory) {
2.     String originalFilename = file.getOriginalFilename();
3.     // 得到文件扩展名
4.     String fileExtension = originalFilename.substring(Objects.requireNonNull(fil
e.getOriginalFilename()).lastIndexOf(".") + 1);
5.     // 重新拼接文件的名字
6.     String newFileName = UUID.randomUUID().toString().replaceAll("-",
", ") + "." + fileExtension;
7.     try {
8.         // 文件内容上传到OSS
9.         PutObjectResult result = OSSUtil.upload(directory + newFileName, file.ge
tInputStream());
10.         ResponseMessage response = result.getResponse();
11.         FileMeta fileMeta = (newFileName, file, response, originalFilename);
12.         // 文件元信息保存到DB
13.         this.save(fileMeta);
14.         return fileMetaParamMapper.entity2Dto(fileMeta);
15.     } catch (IOException e) {
16.         e.printStackTrace();
17.     }

```

```
18.     return null;
19. }
```

2) 下载功能。

```
1. public void download(String id, HttpServletResponse response, String directory)
2. {
3.     // 查询文件元信息
4.     FileMeta fileMeta = this.getById(id);
5.     // 创建 ossClient
6.     OSS ossClient = OSSUtil.getOSSClient();
7.     // 判断文件是否存在
8.     boolean found = ossClient.doesObjectExist(OSSConstant.BUCKET_NAME, directory
9.         + fileMeta.getFileName());
10.    if (!found) {
11.        throw new CommonException(FileErrorEnum.FILE_NOT_FOUND.setParams(new Object[]
12.            {directory + fileMeta.getFileName()}));
13.    }
14.    // 到OSS 查询文件对象
15.    OSSObject ossObject = ossClient.getObject(OSSConstant.BUCKET_NAME, directory
16.        + fileMeta.getFileName());
17.    // 设置 content-type
18.    response.setContentType(ossObject.getObjectMetadata().getContentType());
19.    // 设置头部
20.    response.setHeader("Content-
21.        Disposition", "attachment;filename=" + URLEncoder.encode(fileMeta.getOriginalFil
22.            ename(), "UTF-8"));
23.    IOUtils.copy(ossObject.getObjectContent(), response.getOutputStream());
24. }
```

5.3 项目模块实现

5.3.1 项目组模块

用户在使用该平台进行项目开发时需要加入或创建一个项目组，项目组页面如图 5-7 所示。点击新建项目组按钮，即可在弹出框中输入项目组名称和项目组描述创建一个项目组，在创建的项目组当中会对名称进行唯一性校验。


```

8. int count = this.count(Wrappers.<Group>lambdaQuery()
9.     .eq(Group::getCreateId, currentUser.getUserId())
10.     .eq(Group::getName, groupAddParam.getName()));
11. if (count > 0) {
12.     // 当前用户已经创建过相同名字的项目组
13.     throw CommonException.builder()
14.         .resultCode(GroupErrorEnum.GROUP_NAME_EXIST.setParams(new Object[]{g
15.             roupAddParam.getName()})).build();
16. }
17. // 校验通过，保存项目组信息和成员信息
18. Group group = groupParamMapper.addParam2Entity(groupAddParam);
19. save(group);
20. // 保存用户-项目组对应关系
21. userMtmGroupService.save(getMember(group, currentUser));

```

点击项目组后面的查看组成员按钮，即可查看当前项目组的所有成员。项目组成员界面如图 5-9 所示。

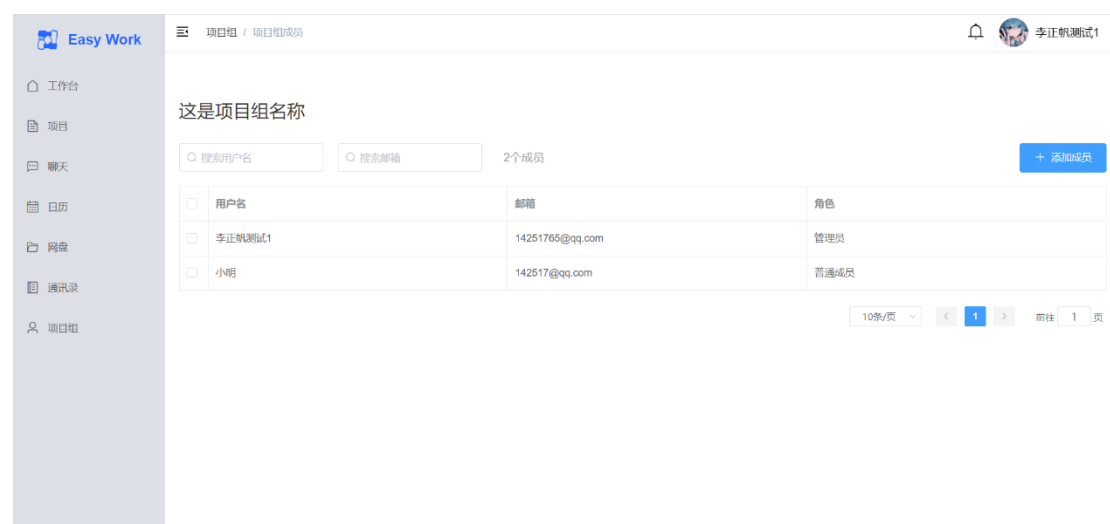


图 5-9 项目组成员界面

点击添加成员按钮，输入用户邮箱，即可向该用户发送邀请，对方需要接到通知后同意，才能成功进入该项目组。添加成员如图 5-10 所示。

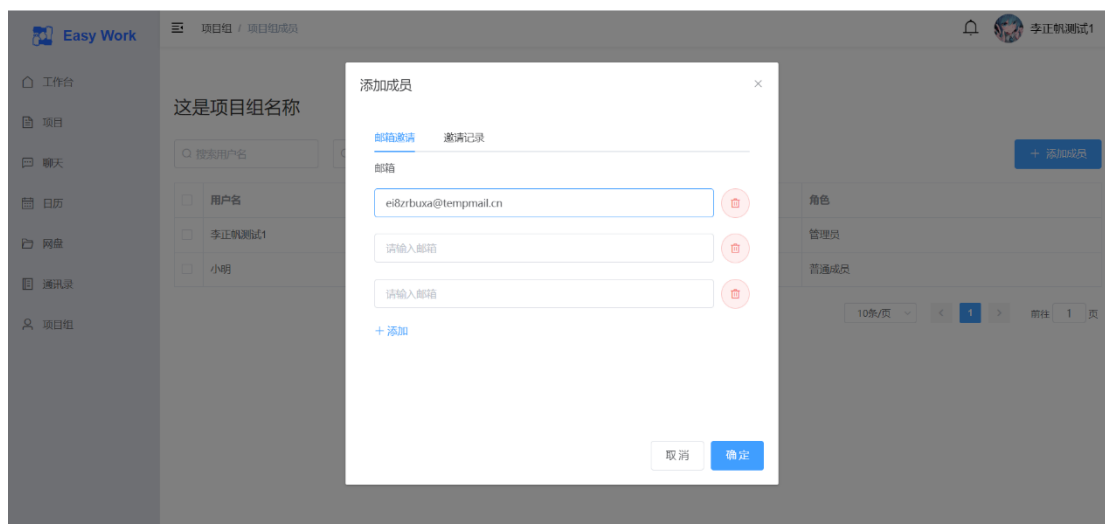


图 5-10 添加成员

在平台的右上角用户头像旁边，有一个通知按钮，点击即可查看通知，邀请进入项目组通知界面如图 5-11 所示。

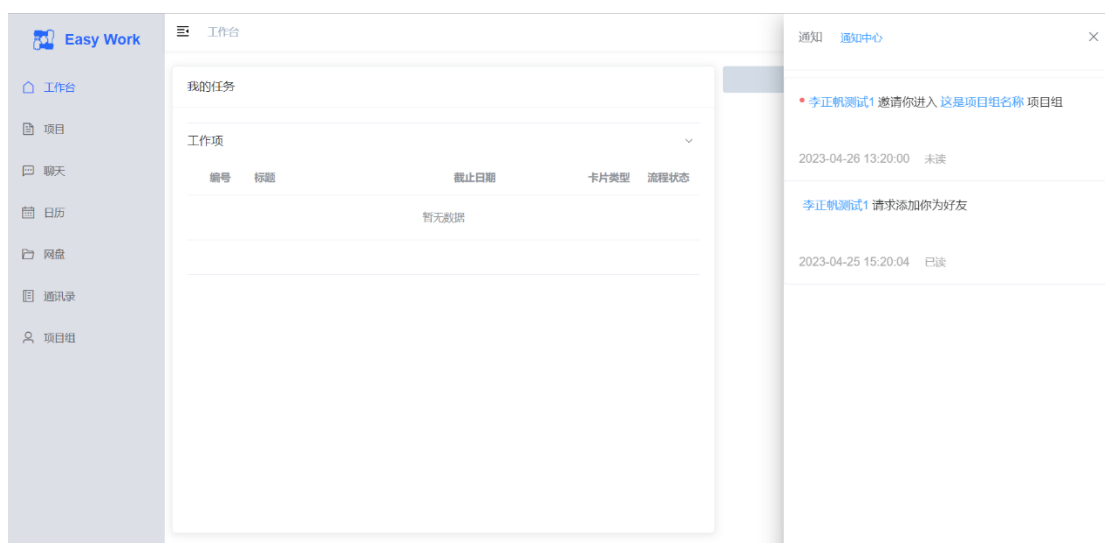


图 5-11 邀请进入项目组界面

邀请用户进入项目组的关键代码如下。

```
1. // 获取当前用户 id
2. String userId = UserUtils.getCurrentUser().getUserId();
3. // 获取邀请的邮箱集合
4. List<String> emails = groupInvitation.getEmails();
5. if (CollectionUtils.isEmpty(emails)) {
6.     for (String email : emails) {
```

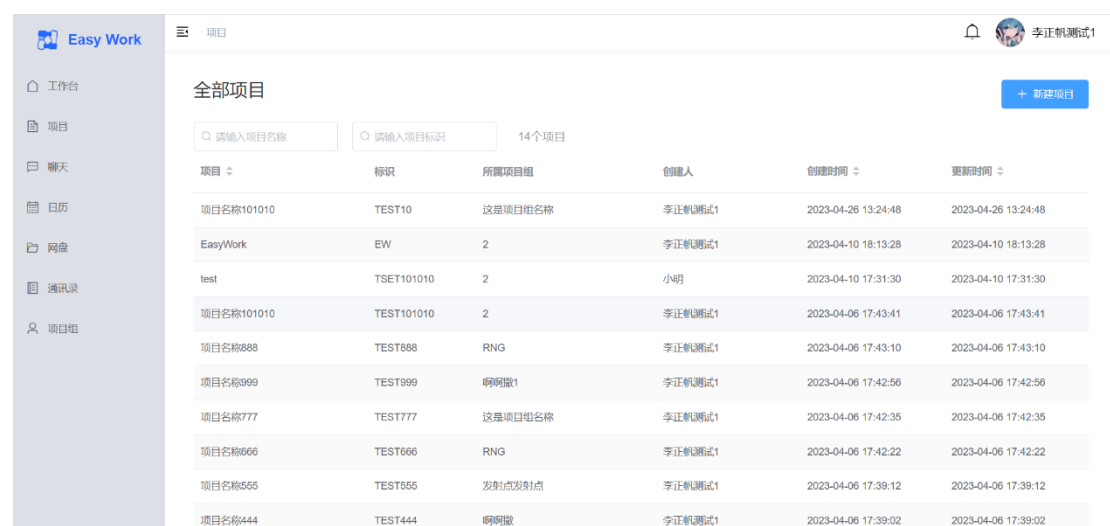
```

7.         if (StringUtils.isEmpty(email)) {
8.             // 根据用户邮箱获取用户基本信息
9.             UserDto user = serverClientService.getUserByEmail(email);
10.            if (user != null) {
11.                // 发通知
12.                Notification notification = getNotification(user);
13.                this.save(notification);
14.            }
15.        }
16.    }
17. }

```

5.3.2 项目信息模块

创建一个项目需要指定一个项目组，再输入项目名称和项目标识后会对项目标识进行唯一性校验。项目信息主界面如图 5-12 所示。



The screenshot shows the 'Easy Work' application interface. On the left is a sidebar with navigation options: 工作台 (Workbench), 项目 (Project), 聊天 (Chat), 日历 (Calendar), 网盘 (Cloud Drive), 通讯录 (Address Book), and 项目组 (Project Group). The main area is titled '全部项目' (All Projects) and shows a list of 14 projects. At the top right of the main area is a '+ 新建项目' (New Project) button. The project list has columns for 项目 (Project), 标识 (Identifier), 所属项目组 (Project Group), 创建人 (Creator), 创建时间 (Creation Time), and 更新时间 (Update Time).

项目	标识	所属项目组	创建人	创建时间	更新时间
项目名称101010	TEST10	这是项目组名称	李正航测试1	2023-04-26 13:24:48	2023-04-26 13:24:48
EasyWork	EW	2	李正航测试1	2023-04-10 18:13:28	2023-04-10 18:13:28
test	TSET101010	2	小明	2023-04-10 17:31:30	2023-04-10 17:31:30
项目名称101010	TEST101010	2	李正航测试1	2023-04-06 17:43:41	2023-04-06 17:43:41
项目名称888	TEST888	RNG	李正航测试1	2023-04-06 17:43:10	2023-04-06 17:43:10
项目名称999	TEST999	啊啊啊1	李正航测试1	2023-04-06 17:42:56	2023-04-06 17:42:56
项目名称777	TEST777	这是项目组名称	李正航测试1	2023-04-06 17:42:35	2023-04-06 17:42:35
项目名称666	TEST666	RNG	李正航测试1	2023-04-06 17:42:22	2023-04-06 17:42:22
项目名称555	TEST555	发射点发射点	李正航测试1	2023-04-06 17:39:12	2023-04-06 17:39:12
项目名称444	TEST444	啊啊啊	李正航测试1	2023-04-06 17:39:02	2023-04-06 17:39:02

图 5-12 项目信息主界面

新建项目界面如图 5-13 所示。

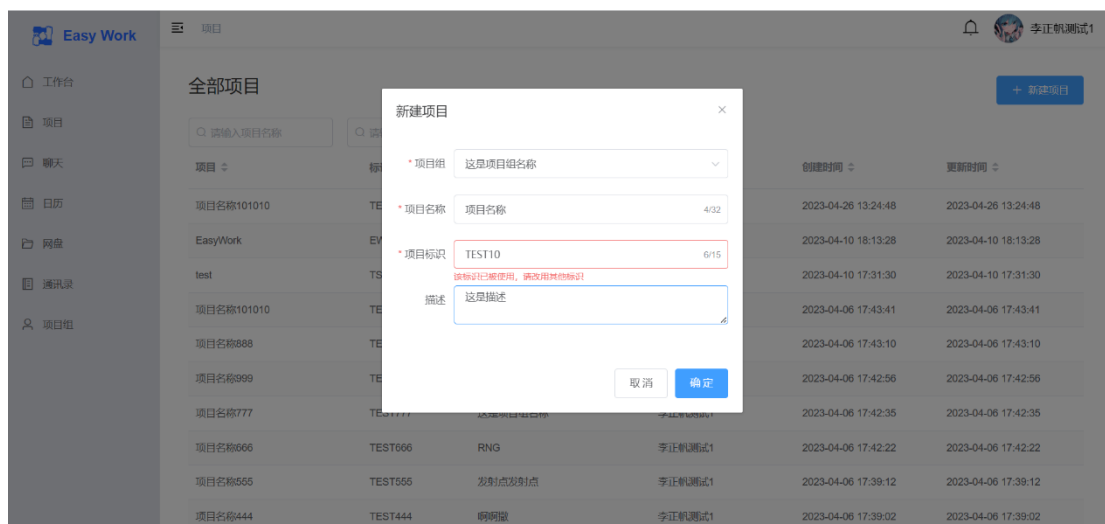


图 5-13 新建项目界面

创建项目信息的关键代码如下。

```
1. // 获取当前用户
2. SsoUser curUser = UserUtils.getCurrentUser();
3. // 校验项目标识唯一性
4. List<String> tabs = this.baseMapper.getTabsById(curUser.getUserid());
5. if (StringUtils.isEmpty(tabs)) {
6.     if (tabs.contains(addParam.getTab())) {
7.         throw CommonException.builder()
8.             .resultCode(ProjectErrorEnum.PROJECT_TAB_EXIST)
9.             .build();
10.    }
11. }
12. Project project = projectParamMapper.addParam2Entity(addParam);
13. log.info("添加项目信息====》{}", JSONObject.toJSONString(project));
14. boolean result = save(project);
15. if (result) {
16.     // 创建项目成功, 在网盘模块同时创建一个项目类型的文件夹
17.     saveNetdisk(project);
18. }
```

5.3.3 工作项模块

在项目信息界面点击一个项目即可进入对应的工作项界面，工作项主界面如图 5-14 所示。在该界面用户可以创建该项目对应的任务项进行开发，计划对应的工作项类型为 Epic，而计划集中可以包含很多计划，对工作项类型的描述见 2.2 敏捷软件开发。



图 5-16 工作项预览

新建工作项的关键代码如下。

```
1. WorkItem workItem = workItemParamMapper.addParam2Entity(addParam);
2. // 如果优先级为空，默认优先级为1
3. if (workItem.getPriority() == null || workItem.getPriority() <= 0) {
4.     workItem.setPriority(1); // 默认优先级
5. } else if (workItem.getPriority() > 5) {
6.     workItem.setPriority(5); // 最高为5
7. }
8. // 风险等级
9. if (workItem.getRisk() == null || workItem.getRisk() < 0) {
10.    workItem.setRisk(0); // 默认风险等级
11. } else if (workItem.getRisk() > 3) {
12.    workItem.setRisk(3); // 最高风险等级为3
13. }
14. // 赋值编号，如果不是Epic，则想要赋值编号
15. if (!WorkItemConstant.EPIC.equals(workItem.getWorkType())) {
16.    // 查询数据库，获得当前项目的最高值
17.    Integer maxNumber = this.getBaseMapper().getMaxNumber(workItem.getProjectId());
18.    workItem.setNumber(maxNumber == null ? 1 : maxNumber + 1);
19.    // 初始流程状态：新建
20.    workItem.setStatus("新建");
21. }
22. boolean result = save(workItem);
23. if (!result) return false;
24. // 保存文件列表
25. saveFileList(workItem.getId(), addParam.getFileList());
```

5.4 网盘模块实现

网盘模块主要分两种类型的文件，项目类型和个人类型。网盘模块主界面如图 5-17 所示。

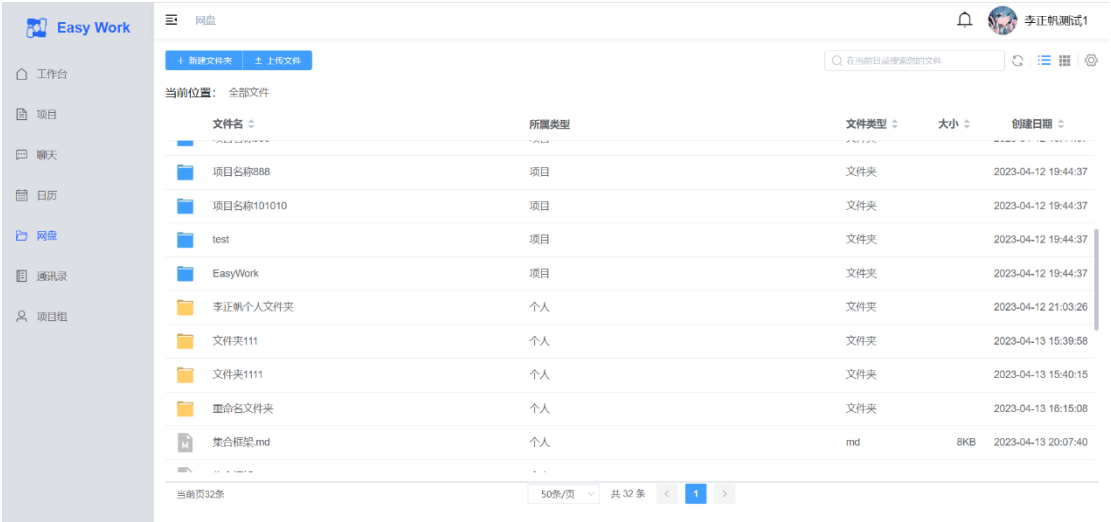


图 5-17 网盘模块主界面

在网盘根路径有一个与项目同名的文件夹，该文件夹的类型为项目类型，指在该文件夹中所有的文件均对项目中的所有成员可见，并且项目中的成员都可以对其进行操作，如图 5-18 所示，项目中的成员即为项目对应的项目组成员。个人类型的文件或文件夹即仅对用户本人可见，其他人不可见。

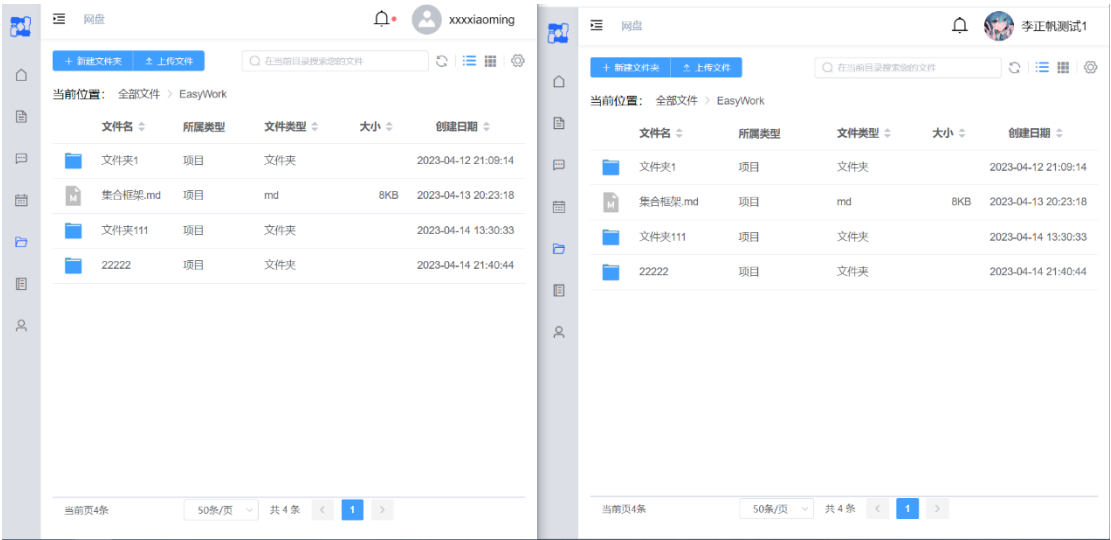


图 5-18 不同用户对同个项目类型文件夹可见

上传文件或新建文件夹的关键代码如下。

```
1. private boolean addNetDiskFile(NetDiskFileAddParam addParam, boolean isDir, boolean allowSameName) {
2.     // 设置文件或文件夹所属类型
3.     if (StringUtils.isEmpty(addParam.getDirId())) {
4.         addParam.setBelongType(NetDiskTypeEnum.PERSONAL.getCode());
5.         addParam.setBelongId(UserUtils.getCurrentUser().getUserid());
6.     } else {
7.         NetDiskFile dir = this.getById(addParam.getDirId());
8.         addParam.setBelongId(dir.getBelongId());
9.         addParam.setBelongType(dir.getBelongType());
10.    }
11.    // 判断文件路径
12.    if (StringUtils.isEmpty(addParam.getFilePath())) {
13.        // 默认为主目录路径
14.        addParam.setFilePath(NetDiskConstant.MAIN_DIR_PATH);
15.    } else if (addParam.getFilePath().indexOf(NetDiskConstant.MAIN_DIR_PATH) != 0) {
16.        // 开头不是"/"
17.        addParam.setFilePath(NetDiskConstant.MAIN_DIR_PATH + addParam.getFilePath());
18.    }
19.    NetDiskFile netDiskFile = netDiskFileParamMapper.addParam2Entity(addParam);
20.    // 如果添加的是文件
21.    if (!isDir) {
22.        // 根据文件id 获取文件基本信息
23.        FileMetaDto fileMeta = serverClientService.getFileById(netDiskFile.getFileId());
24.        String fileName = fileMeta.getName();
25.        // 赋值文件扩展名
26.        int idx = fileName.lastIndexOf('.');
27.        netDiskFile.setExtendName(fileName.substring(idx + 1));
28.        netDiskFile.setFileName(fileName.substring(0, idx)); // 赋值文件名称
29.    }
30.    // 设置类型是否为文件夹
31.    netDiskFile.setIsDir(isDir ? 1 : 0);
32.    return saveNetDiskFile(netDiskFile, allowSameName);
33. }
```

在网盘模块中对文件或文件夹的操作包括：预览、删除、复制、移动、重命名、下载和查看文件详情。

1) 预览。文件的预览支持大多数常见的文件格式。以图片预览为例，图片预览如图 5-19 所示。

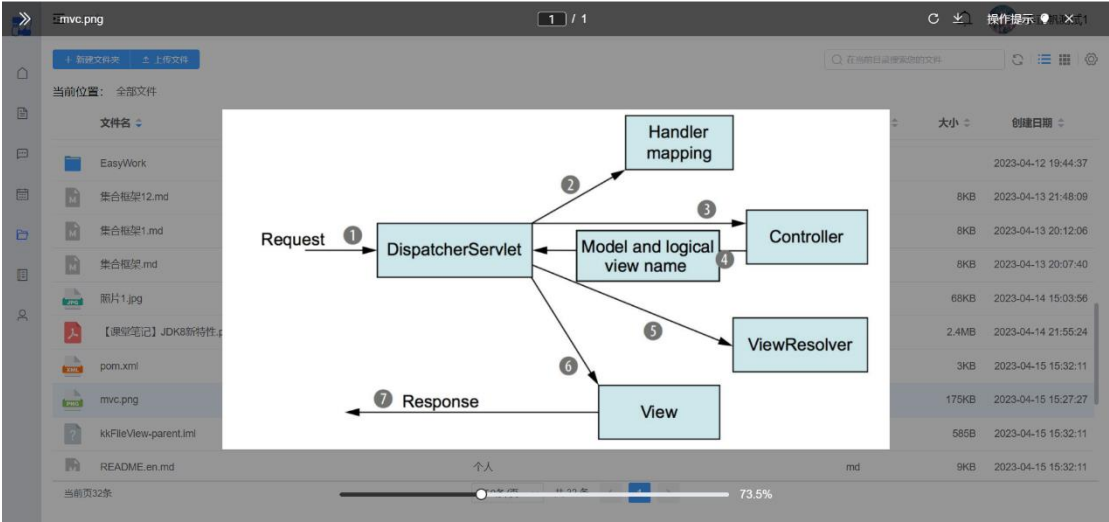


图 5-19 图片预览

2) 删除会先询问是否继续操作。删除文件如图 5-20 所示。

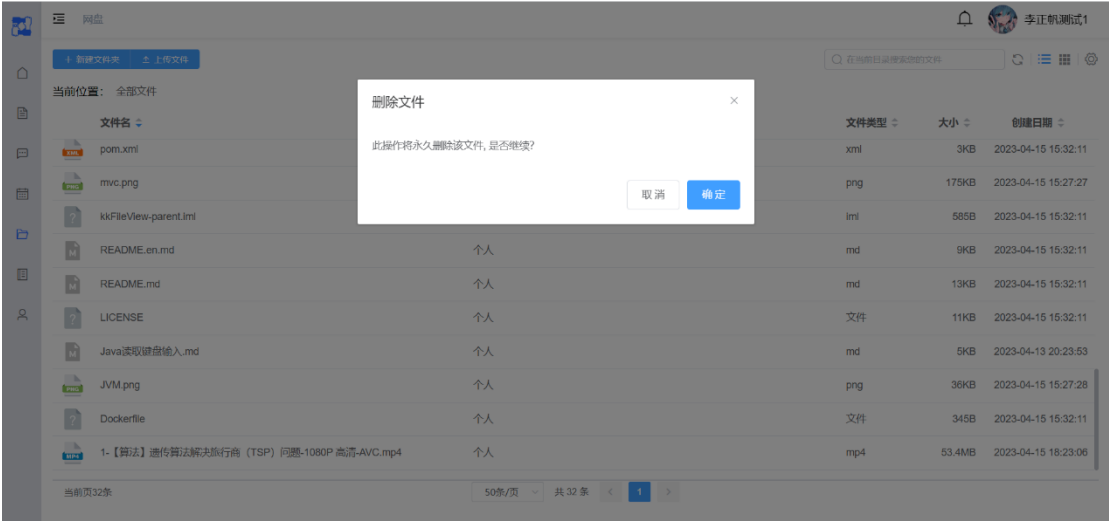


图 5-20 文件删除

3) 文件或文件夹的复制需要选择指定的路径，文件复制如图 5-21 所示。

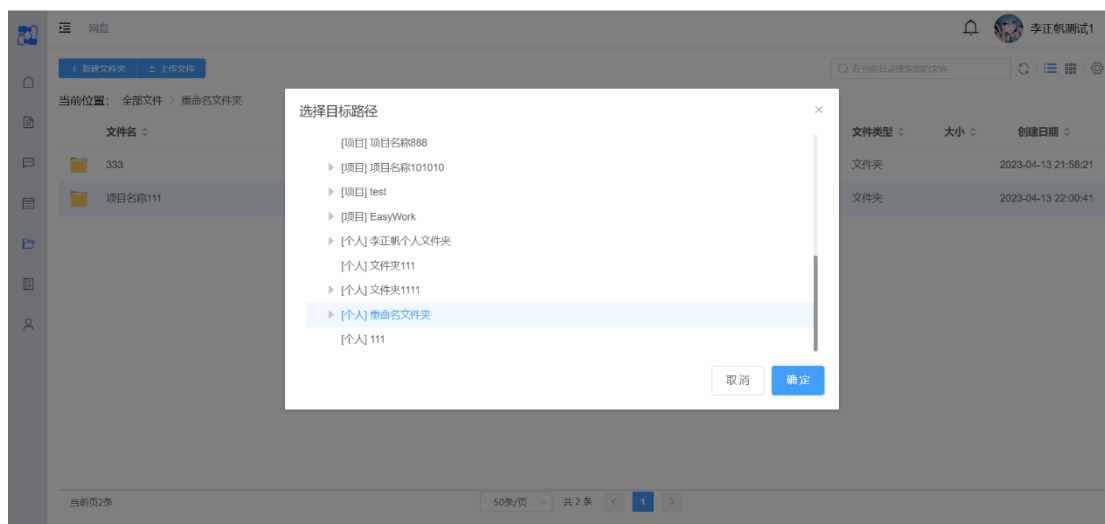


图 5-21 文件复制

如果选择的路径与当前文件的路径相同，则文件名称将自动加上“copy 副本”前缀。复制结果如图 5-22 所示。



图 5-22 复制结果

4) 文件的移动需要选择指定的路径，如果该路径上有相同名字的文件，系统则提示移动失败。文件移动如图 5-23 所示。



图 5-23 文件移动

文件复制和移动的逻辑相同，其关键代码如下。

```
1. NetDiskFile source = this.getById(editParam.getId());
2. // 如果 source 是文件夹，先获取 source 作为文件夹时，子文件的 filePath
3. String childFilePath = getFilePath(source);
4. // 获取 source 的 belongType 和 belongId
5. Integer childBelongType = source.getBelongType();
6. String childBelongId = source.getBelongId();
7. // 目标目录 id 为空，则复制到目标 filePath
8. if (StringUtils.isEmpty(editParam.getDirId())) {
9.     // 移动到主目录 "/"
10.    source.setFilePath(NetDiskConstant.MAIN_DIR_PATH);
11. } else { // 移动到目标 id 目录
12.    NetDiskFile dir = this.getById(editParam.getDirId()); // 查出目标目录
13.    source.setBelongType(dir.getBelongType());
14.    source.setBelongId(dir.getBelongId());
15.    source.setDirId(dir.getId());
16.    // 设置正确的 filePath
17.    if (NetDiskConstant.MAIN_DIR_PATH.equals(dir.getFilePath())) {
18.        source.setFilePath(dir.getFilePath() + dir.getFileName());
19.    } else {
20.        source.setFilePath(dir.getFilePath() + "/" + dir.getFileName());
21.    }
22.    if (dir.getFileNameNum() != null && dir.getFileNameNum() != 0) {
23.        source.setFilePath(source.getFilePath() + "(" + dir.getFileNameNum() + "
    ");
24.    }
25. }
26. // 判断目标目录是否有同名文件
```

```
27. int count = this.count();
28. if (count > 0) {
29.     source.setFileName("copy 副本 - " + source.getFileName());
30. }
31. // 保存
32. source.setId(null);
33. this.save(source);
```

5) 文件重命名需要判断当前路径下是否有相同名字的文件，如果存在，则系统提示重命名失败。文件重命名如图 5-24 所示。



图 5-24 文件重命名

6) 文件下载时，获取当前文件的 fileId，调用文件模块的下载接口即可完成下载。文件下载如图 5-25 所示。



图 5-25 文件下载

7) 文件详情可以查看文件的名称、类型、大小和创建人等信息。文件详情如图 5-26 所示。



图 5-26 文件详情

5.5 聊天模块实现

在该系统中，用户是作为客户端存在的，一般情况下，客户端与客户端之间无法直接建立连接，因为大多数网络环境都处于被动接受服务的状态，需要主动发起连接请求才能与其他设备进行通信。另外，客户端之间的直接连接可能涉及到网络安全和隐私等问题，需要进行一些特殊的设置和授权才能实现。在这种情况下，可以考虑使用服务器作为中转来实现客户端之间的通信。通过客户端与服务端之间的连接，可以将消息发送到服务端，服务端再将消息转发给目标客户端，从而实现客户端之间的通信。因此，在聊天模块的实现中，便是依靠 WebSocket 协议让客户端与服务端建立连接，因为 WebSocket 是一种在单个 TCP 连接上进行全双工通信的协议，客户端和服务端可以相互主动通信。WebSocket 服务端使用 Spring Boot 来创建，并开放一个端点（Endpoint），也就是 URL 路径。客户端使用 JavaScript 实现，通过 JavaScript 访问该 URL 来建立 WebSocket 连接，与服务端进行实时通信。

当用户发送消息时，本质上的将消息发送给服务端，服务端接收到消息之后

解析得到消息要发送的对象 id，服务端将消息转发给对应的对象 id，对方从而得到消息。而群聊的实现也是类似，只不过服务端要转发的并不只是一个对象，而是群聊中的所有对象。服务端处理消息的路程图如图 5-27 所示。

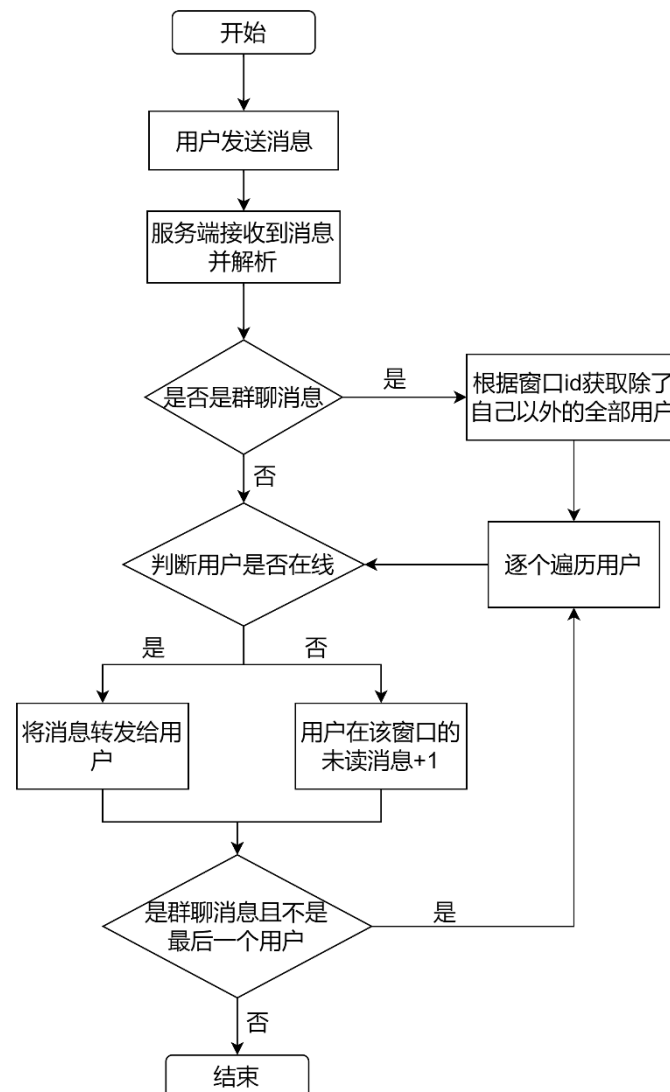


图 5-27 服务端处理消息流程图

服务端处理消息的关键代码如下。

```
1. // 根据 contactId（项目组 id）找出组成员，逐个判断是否在线。在线->转发，不在线->未读数+1
2. List<String> userIds = projectClientService.getUserIdsByGroupId(messageDto.getToContactId());
3. if (CollectionUtils.isNotEmpty(userIds)) {
4.     for (String userId : userIds) {
5.         // 不用转发给自己
```

```

6.         if (userId.equals(message.getFromUserId())) {
7.             continue;
8.         }
9.         // 逐个判断是否在线
10.        if (WsSessionUtils.isUserOnline(userId)) {
11.            Set<String> userSet = WsSessionUtils.getMultiUserSet(userId);
12.            for (String userAndTimeId : userSet) {
13.                Session targetSession = WsSessionUtils.getSession(userAndTimeId)
14.                ;
15.                // 异步转发消息
16.                targetSession.getAsyncRemote().sendText(jsonStrMsg);
17.            }
18.        } else {
19.            // 不在线，未读数量+1
20.            groupChatMemberService.addUnreadOrSave(userId, messageDto.getToContactId());
21.        }
22.    }
23.    // 将消息保存至数据库
24.    messageService.save(message);

```

聊天主界面如图 5-28 所示。

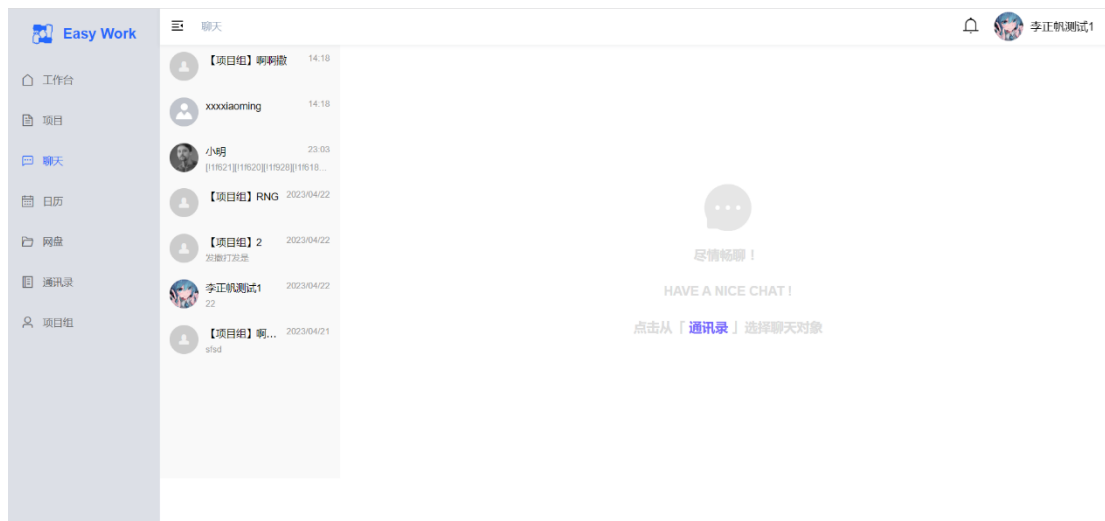


图 5-28 聊天主界面

用户刚开始进入聊天主界面的时候，如果没有其他用户对自己发送消息，界面上是没有联系人窗口的，需要去通讯录模块选择好友或选择项目组进行聊天，

通讯录界面如图 5-29 所示。如果选择一个用户，则是一对一的聊天，如果选择项目组，则是群聊。目前系统不支持自定义群聊。

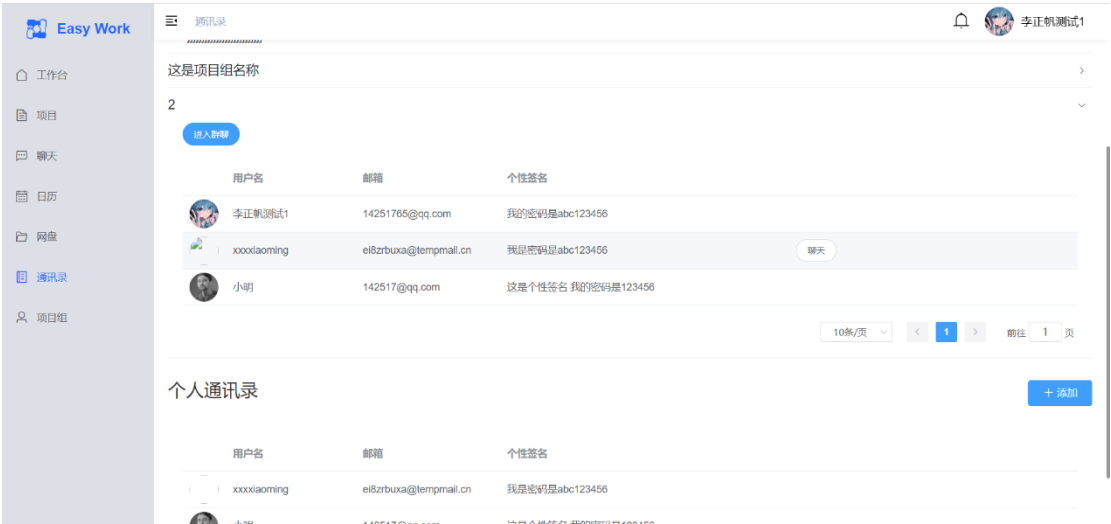


图 5-29 通讯录界面

一对一聊天效果图如图 5-30 所示。

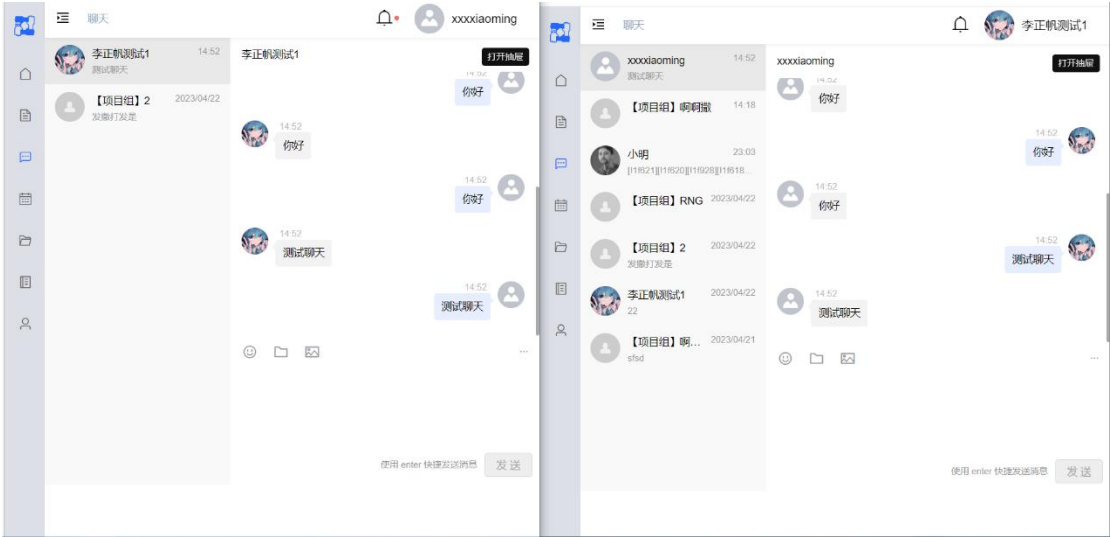


图 5-30 一对一聊天效果图

群聊效果图如图 5-31 所示。

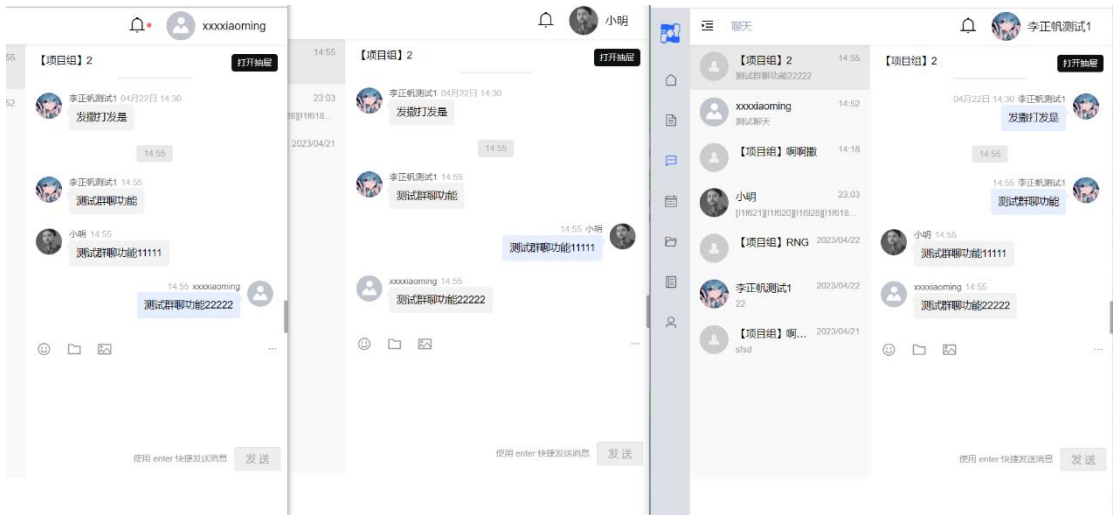


图 5-31 群聊效果图

5.6 日历模块实现

用户可以在日程模块中创建日程和待办，创建的日程均会根据时间显示在对应的日期框上，日历主界面如图 5-32 所示。

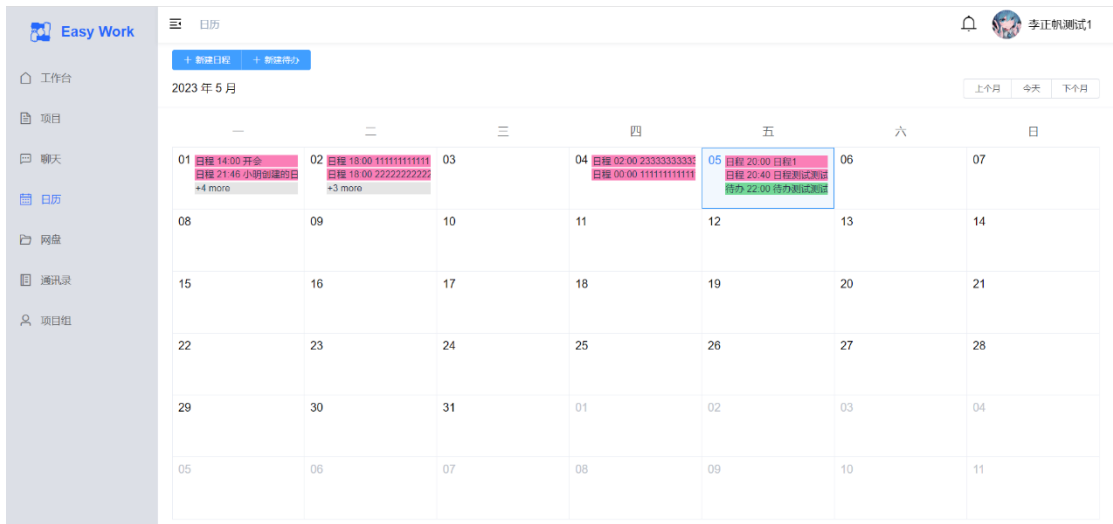


图 5-32 日历主界面

创建日程需要输入日程标题、时间段、参与人等信息，并且默认勾选邮件提醒，新建日程如图 5-33 所示。

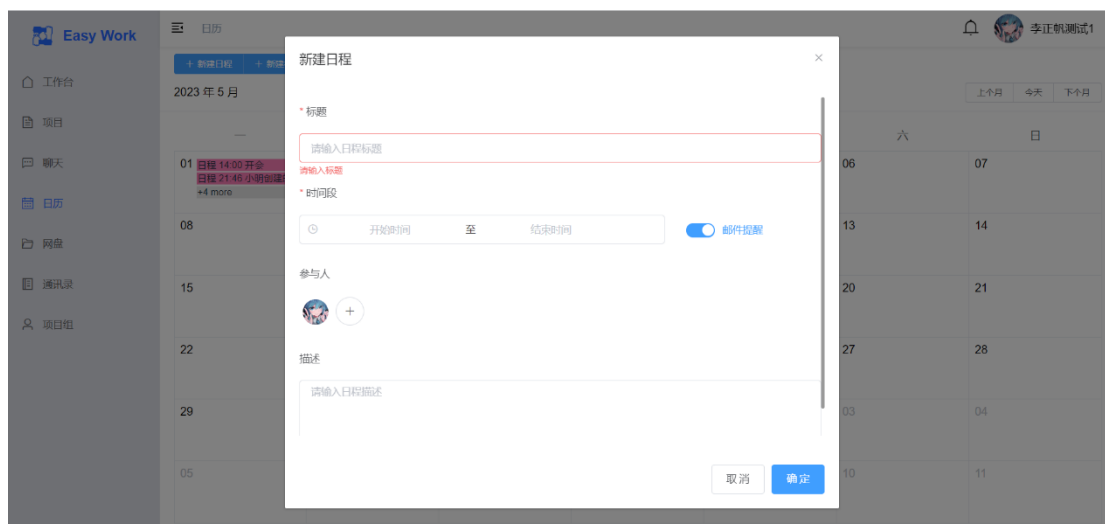


图 5-33 新建日程

如果勾选了邮件提醒，则会在日程即将开始之前发送邮件提醒参与人，日程邮件提醒如图 5-34 所示。



图 5-34 日程邮件提醒

创建待办时可以自定义在截止时间前几分钟提醒用户，新建待办如图 5-35 所示。

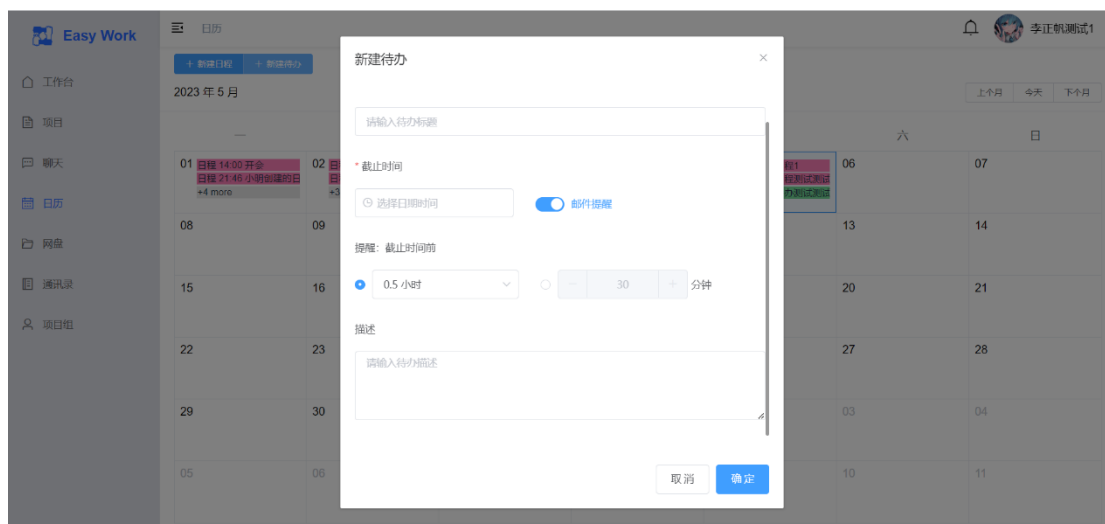


图 5-35 新建待办

如图勾选了邮件提醒，则会另外通过邮件提醒用户，待办邮件提醒如图 5-36 所示。



图 5-36 待办邮件提醒

无论是否勾选了邮件提醒，系统均会在界面上通知用户。日程和待办的系统通知如图 5-37 所示。

Easy Work

工作台

项目

聊天

日历

网盘

通讯录

项目组

日历

+ 新增日程+ 新增待办

2023 年 5 月

一	二	三	四	五
01 <div>日程 14:00 开会 日程 21:45 小明创建的日 +4 more</div>	02 <div>日程 19:00 111111111111 日程 18:00 222222222222 +3 more</div>	03	04 <div>日程 02:00 233333333333 日程 00:00 111111111111</div>	05 <div>日程 20:00 日 日程 20:40 日 待办 22:00 待</div>
08	09	10	11	12
15	16	17	18	19
22	23	24	25	26
29	30	31	01	02
05	06	07	08	09

通知 通知中心

您有一个日程 111111111111 即将开始，时间为 2023-05-02 18:00 - 2023-05-02 19:00，请及时参加

2023-05-02 17:30:00 已读

您有一个待办 3333333333333333，4 分钟后截止，截止时间为 2023-05-02 00:05，请及时完成

2023-05-02 00:01:00 已读

您有一个日程 323333 即将开始，时间为 2023-05-02 00:00 - 2023-05-05 00:00，请及时参加

2023-05-01 23:55:00 已读

您有一个日程 323333 即将开始，时间为 2023-05-02 00:00 - 2023-05-05 00:00，请及时参加

2023-05-01 23:30:00 已读

您有一个待办 dalban，16 分钟后截止，截止时间为 2023-05-01 23:16，请及时完成

图 5-37 日程和待办系统通知