迭代一：测试文档

1. 简介

1.1 项目背景

本项目是一个线上众包测试平台，项目背景详情见需求规格文档。

1.2 测试目的

1. 前端页面渲染
2. 后端数据库交互
3. 后端接口完成度
4. 后端接口正确性
5. 后端接口之间的交互
6. 前端与后端的交互

1.3 测试环境

- springboot版本：2.6.3
- mybatis-plus版本：3.5.1
- mysql版本：8.0.28
- vue版本：2.X
- junit版本：5.8.2

1.4 测试范围

- 测试主要为集成测试，后端包括Mapper层单元测试，Service层集成测试，Controller层集成测试。前端测试包括与后端的交互。

2. 测试计划

2.1 后端测试

1. 单元测试：主要对后端的Mapper层进行单元测试。
2. 集成测试：调用后端的Service层和Controller层，以及前端进行层层集成测试。

3. 后端测试

3.1 Mapper层单元测试

3.1.1 测试方向

测试mapper包下的每一个被调用过的方法并调用数据库进行验证。

3.1.2 测试准备

1. 测试方法使用@Transactional注解保证任务原子性，并且在运行完测试用例后将涉及文件操作的文件夹中残留文件删除，这样不容易受外部数据干扰，也不会影响后续的测试进行。
2. 关闭正在运行的后端避免测试途中外部传入新数据影响测试结果。

3.1.3 测试用例

3.1.3.1 UserMapper

测试编号：0

测试方法：userMapper.selectByMap

Plain Text 复制代码

```
    void testSelect(){
        Map<String,Object> map=new HashMap<>();
        map.put("email","123");
        assert 0==userMapper.selectByMap(map).size();
    }
```

测试编号：1

测试方法：userMapper.insert

Java 复制代码

```java
    void testInsert(){
        User user=new User("123","123",UserIdentity.DELIVER);
        assert 1==userMapper.insert(user);
        Map<String,Object> map=new HashMap<>();
        map.put("email","123");
        userMapper.deleteByMap(map);
    }
```

3.1.3.2 CodeMapper

测试编号：0

测试方法：codeMapper.selectByMap

Java 复制代码

```java
    void testSelect(){
        Map<String,Object> map=new HashMap<>();
        map.put("code","123");
        assert codeMapper.selectByMap(map).size()==0;
    }
```

测试编号：1

测试方法：codeMapper.deleteByMap

Java 复制代码

```java
    void testDelete(){
        codeMapper.insert(new Code("123456"));
        Map<String,Object> map=new HashMap<>();
        map.put("code","123456");
        assert codeMapper.deleteByMap(map)==1;
    }
```

3.1.3.3 TaskMapper

测试编号：0

测试方法：taskMapper.insert

Java 复制代码

```java
    void testInsert(){
        User user=new User("23","32312", UserIdentity.DELIVER);
        userMapper.insert(user);
        Task task=new Task(user.getId(),"213d",123,123,0,0L,"dsajdo","fdejiodfj","ejwi");
        assert 1==taskMapper.insert(task);
        taskMapper.deleteByMap(null);
        userMapper.deleteByMap(null);
    }
```

测试编号1：

测试方法：taskMapper.updateById

Java 复制代码

```java
    void testUpdateById(){
        User user=new User("23","32312", UserIdentity.DELIVER);
        userMapper.insert(user);
        Task task=new Task(user.getId(),"213d",123,123,0,0L,"dsajdo","fdejiodfj","dhi");
        taskMapper.insert(task);
        int t=task.getId();
        task.setId(234325423);
        assert 0==taskMapper.updateById(task);
        task.setId(t);
        task.setAurl("67");
        assert 1==taskMapper.updateById(task);
        assert "67".equals(taskMapper.selectById(task.getId()).getAurl());
        taskMapper.deleteByMap(null);
        userMapper.deleteByMap(null);
    }
```

测试编号：2

测试方法：taskMapper.selectByMap

Java 复制代码

```java
void testSelectByMap(){
        User user=new User("23","32312", UserIdentity.DELIVER);
        userMapper.insert(user);
        Task task=new Task(user.getId(),"213d",123,123,0,0L,"dsajdo","fdejiodfj","wdioj");
        taskMapper.insert(task);
        Map<String, Object> select = new HashMap<>();
        select.put("id", task.getId());
        select.put("userId", task.getUserId());
        assert taskMapper.selectByMap(select).size()==1;
        select.remove("id");
        Task task2=new Task(user.getId(),"213d",123,123,0,0L,"dsajdo","fdejiodfj","fejo");
        taskMapper.insert(task2);
        assert taskMapper.selectByMap(select).size()==2;
        Task task3=new Task(user.getId(),"213d",123,123,0,0L,"dsajdo","fdejiodfj","efjoi");
        taskMapper.insert(task3);
        assert taskMapper.selectByMap(null).size()==3;
        taskMapper.deleteByMap(null);
        userMapper.deleteByMap(null);
    }
```

测试编号：3

测试方法：taskMapper.selectTaskByLabel

Java 复制代码

```java
void testSelectTaskByLabel(){
        User user=new User("23","32312", UserIdentity.DELIVER);
        userMapper.insert(user);
        long time=System.currentTimeMillis()*2;
        Task task=new Task(user.getId(),"SMTC",123,123,0,time,"dsajdo","fdejiodfj","wo");
        Task task1=new Task(user.getId(),"MT",123,123,1,time/4,"dsajdo","fdejiodfj","feji");
        taskMapper.insert(task);
        taskMapper.insert(task1);
        Integer tag=null;
        Integer if_finished=null;
        long now_time=System.currentTimeMillis();
        assert taskMapper.selectTaskByLabel(tag,if_finished,null,now_time).size()==2;
        if_finished=0;
        assert taskMapper.selectTaskByLabel(tag,if_finished,null,now_time).size()==1;
        if_finished=1;
        assert taskMapper.selectTaskByLabel(tag,if_finished,null,now_time).size()==1;
        tag=0;
        assert taskMapper.selectTaskByLabel(tag,if_finished,null,now_time).size()==0;
        tag=1;
        assert taskMapper.selectTaskByLabel(tag,if_finished,null,now_time).size()==1;
        if_finished=null;
        tag=null;
        String name="MT";
        assert taskMapper.selectTaskByLabel(tag,if_finished,name,now_time).size()==2;
        name="MTC";
        assert taskMapper.selectTaskByLabel(tag,if_finished,name,now_time).size()==1;
        taskMapper.deleteByMap(null);
        userMapper.deleteByMap(null);
    }
```

3.1.3.4 WorkMapper

测试编号：0

测试方法：workMapper.insert

Java 复制代码

```java
    public void testInsert(){
        User user0=new User("666","jdjwdfnm", UserIdentity.DELIVER);
        User user1=new User("waoij","fejiojf",UserIdentity.WORKER);
        User user2=new User("ooj","foiwjf",UserIdentity.WORKER);
        userMapper.insert(user0);
        userMapper.insert(user2);
        userMapper.insert(user1);
        Task task=new Task(user0.getId(),"jeiw",10,10,1,0L,"dea","ejijfw","djo");
        taskMapper.insert(task);
        assert 1==workMapper.insert(new Work(null,user1.getId(),task.getId(), WorkStatus.TO_FINISH));
        workMapper.deleteByMap(null);
        taskMapper.deleteByMap(null);
        userMapper.deleteByMap(null);
    }
```

测试编号：1

测试方法：workMapper.selectByMap

Java 复制代码

```java
public void testSelectByMap(){
        User user0=new User("666","jdjwdfnm", UserIdentity.DELIVER);
        User user1=new User("waoij","fejiojf",UserIdentity.WORKER);
        User user2=new User("ooj","foiwjf",UserIdentity.WORKER);
        userMapper.insert(user0);
        userMapper.insert(user2);
        userMapper.insert(user1);
        Task task=new
Task(user0.getId(),"jeiw",10,10,1,0L,"dea","ejijfw","feji");
        Task task2=new
Task(user0.getId(),"jeiw",10,10,1,0L,"dea","ejijfw","fejo");
        taskMapper.insert(task);
        taskMapper.insert(task2);
        workMapper.insert(new Work(null,user1.getId(),task.getId(),
WorkStatus.TO_FINISH));
        workMapper.insert(new Work(null,user1.getId(),task2.getId(),
WorkStatus.TO_FINISH));
        workMapper.insert(new Work(null,user2.getId(),task.getId(),
WorkStatus.TO_FINISH));
        workMapper.insert(new Work(null,user2.getId(),task2.getId(),
WorkStatus.TO_FINISH));
        Map<String,Object> map=new HashMap<>();
        map.put("userId",user1.getId());
        assert 2==workMapper.selectByMap(map).size();
        map.put("taskId",task.getId());
        assert 1==workMapper.selectByMap(map).size();
        workMapper.deleteByMap(null);
        taskMapper.deleteByMap(null);
        userMapper.deleteByMap(null);
    }
```

3.1.3.5 ReportMapper

测试编号：0

测试方法：reportMapper.selectByMap

```java
public void testSelectByTaskId(){
        User user0=new User("342","321", UserIdentity.DELIVER);
        User user1=new User("34","321", UserIdentity.WORKER);
        User user2=new User("34333","321", UserIdentity.WORKER);
        userMapper.insert(user0);
        userMapper.insert(user1);
        userMapper.insert(user2);
        Task task=new Task(user0.getId(),"软工三作业检查",10,10,0,0L,"a","b","dfs");
        taskMapper.insert(task);
        Report report1=new Report(user1.getId(),task.getId(),"67大帝","czy","6+");
        reportMapper.insert(report1);
        Report report2=new Report(user2.getId(),task.getId(),"67大帝","czy","6+");
        reportMapper.insert(report2);
        Map<String, Object> map = new HashMap<>();
        map.put("taskId", task.getId());
        assert reportMapper.selectByMap(map).size()==2;
        reportMapper.deleteByMap(null);
        taskMapper.deleteById(task.getId());
        userMapper.deleteById(user0.getId());
        userMapper.deleteById(user1.getId());
        userMapper.deleteById(user2.getId());
    }
```

3.1.3.6 ReportImageMapper

测试编号：0

测试方法：reportImageMapper.deleteByMap

Java 复制代码

```java
    public void testDelete(){
        User user0=new User("123","321", UserIdentity.DELIVER);
        User user1=new User("1","321", UserIdentity.WORKER);
        userMapper.insert(user0);
        userMapper.insert(user1);
        Task task=new Task(user0.getId(),"软工三作业检查",10,10,0,0L,"a","b","cc");
        taskMapper.insert(task);
        Report report1=new Report(user1.getId(),task.getId(),"好垃圾啊","点进去就闪退","赢麻了的Mate40 pro plus 12+512");
        reportMapper.insert(report1);
        List<String> list= Arrays.asList("123","3423","3213");
        for (String s : list) {
            reportImageMapper.insert(new ReportImage(report1.getId(),s));
        }
        Map<String, Object> map = new HashMap<>();
        map.put("reportId", report1.getId());
        assert reportImageMapper.selectByMap(map).size()==3;
        for (String s:list){
            map=new HashMap<>();
            map.put("image",s);
            reportImageMapper.deleteByMap(map);
        }
        assert reportImageMapper.selectByMap(map).size()==0;
        reportImageMapper.deleteByMap(null);
        reportMapper.deleteByMap(null);
        taskMapper.deleteById(task.getId());
        userMapper.deleteById(user0.getId());
        userMapper.deleteById(user1.getId());
    }
```

测试编号：1

测试方法：reportImageMapper.insert

Java | 复制代码

```java
public void testInsert(){
        User user0=new User("123","321", UserIdentity.DELIVER);
        User user1=new User("1","321", UserIdentity.WORKER);
        userMapper.insert(user0);
        userMapper.insert(user1);
        Task task=new Task(user0.getId(),"软工三作业检查",10,10,0,0L,"a","b","dfs");
        taskMapper.insert(task);
        Report report1=new Report(user1.getId(),task.getId(),"好垃圾啊","点进去就闪退","赢麻了的Mate40 pro plus 12+512");
        reportMapper.insert(report1);
        List<String> list= Arrays.asList("123","3423","3213");
        for (String s : list) {
            reportImageMapper.insert(new ReportImage(report1.getId(),s));
        }
        Map<String, Object> map = new HashMap<>();
        map.put("reportId", report1.getId());
        assert reportImageMapper.selectByMap(map).size()==3;
        reportImageMapper.deleteByMap(null);
        reportMapper.deleteByMap(null);
        taskMapper.deleteById(task.getId());
        userMapper.deleteById(user0.getId());
        userMapper.deleteById(user1.getId());
    }
```

测试编号：2

测试方法：reportImageMapper.selectByMap

Java 复制代码

```java
 public void testSelectByMap(){
        User user0=new User("342","321", UserIdentity.DELIVER);
        User user1=new User("34","321", UserIdentity.WORKER);
        userMapper.insert(user0);
        userMapper.insert(user1);
        Task task=new Task(user0.getId(),"软工三作业检查",10,10,0,0L,"a","b","ddji");
        taskMapper.insert(task);
        Report report1=new Report(user1.getId(),task.getId(),"67大帝","czy","6+");
        reportMapper.insert(report1);
        List<String> list= Arrays.asList("123","3423","3213");
        for (String s : list) {
            reportImageMapper.insert(new ReportImage(report1.getId(),s));
        }
        Map<String, Object> map = new HashMap<>();
        map.put("reportId", report1.getId());
        assert reportImageMapper.selectByMap(map).size()==3;
        reportImageMapper.deleteByMap(null);
        reportMapper.deleteByMap(null);
        taskMapper.deleteById(task.getId());
        userMapper.deleteById(user0.getId());
        userMapper.deleteById(user1.getId());
    }
```

3.2 Service层集成测试

3.2.1 测试方向

通过调用方法以及assert断言测试Service层中各方法的准确性。

3.2.2 测试准备

1. 测试方法使用@Transactional注解保证任务原子性，并且在运行完测试用例后将涉及文件操作的文件夹中残留文件删除，保证下一次测试不受影响。
2. 关闭正在运行的后端避免测试途中外部传入新数据影响测试结果。

3.2.3 测试用例

3.2.3.1 UserService

测试编号：0

测试方法：userService.register

```java
    void testRegister(){
        UserDTO userDTO=new UserDTO("1123","123456",UserIdentity.DELIVER);
        assert userService.register(userDTO).getCode()==4000;
        assert userService.register(userDTO).getCode()==4001;
    }
```

测试编号：1

测试方法：userService.login

```java
    void testLogin(){
        UserDTO userDTO=new UserDTO("1123","123456",UserIdentity.DELIVER);
        assert userService.login(userDTO).getCode()==4002;
        userService.register(userDTO);
        UserDTO userDTO2=new UserDTO("1123","123457",UserIdentity.DELIVER);
        assert userService.login(userDTO2).getCode()==4003;
        assert userService.login(userDTO).getCode()==4000;
    }
```

测试编号：2

测试方法：userService.codeJudge

```java
    void testCode() {
        assert !userService.codeJudge("abc");
        codeMapper.insert(new Code("abc"));
        assert userService.codeJudge("abc");
    }
```

测试编号：3

测试方法：userService.getIdentityByUserId

```java
    void testGetIdentityByUserId(){
        UserDTO userDTO=new UserDTO("tzh19850355091@163.com","123456", UserIdentity.DELIVER);
        userService.register(userDTO);
        Map<String,Object> map=new HashMap<>();
        map.put("email","tzh19850355091@163.com");
        assert UserIdentity.DELIVER==userService.getIdentityByUserId(
                userMapper.selectByMap(map).get(0).getId());
    }
```

3.2.3.2 WorkService

测试编号：0

测试方法：workService.findTaskByUserId

```java
    public void testFindTaskByUserId() {
        User user0 = new User("666", "jdjwdfnm", UserIdentity.DELIVER);
        User user1 = new User("waoij", "fejiojf", UserIdentity.WORKER);
        userMapper.insert(user0);
        userMapper.insert(user1);
        Task task = new Task(user0.getId(), "jeiw", 10, 10, 1, System.currentTimeMillis() + 100000, "dea", "ejijfw", "wo");
        taskMapper.insert(task);
        workMapper.insert(new Work(null, user1.getId(), task.getId(), WorkStatus.TO_FINISH));
        assert workService.findTaskByUserId(user1.getId(), WorkStatus.FINISH).getData().size() == 0;
        assert workService.findTaskByUserId(user1.getId(), WorkStatus.TO_FINISH).getData().size() == 1;
    }
```

测试编号：1

测试方法：workService.partTask

Java 复制代码

```java
    public void testPartTask() {
        User user0 = new User("666", "jdjwdfnm", UserIdentity.DELIVER);
        User user1 = new User("waoij", "fejiojf", UserIdentity.WORKER);
        User user2 = new User("ooj", "foiwjf", UserIdentity.WORKER);
        userMapper.insert(user0);
        userMapper.insert(user2);
        userMapper.insert(user1);
        Task task = new Task(user0.getId(), "jeiw", 10, 10, 1, 0L, "dea", "ejijfw", "fej");
        Task task2 = new Task(user0.getId(), "jeiw", 10, 10, 1, System.currentTimeMillis() + 2743892784327L, "dea", "ejijfw", "wo");
        taskMapper.insert(task);
        taskMapper.insert(task2);
        assert workService.partTask(new WorkDTO(user1.getId(), task.getId(), WorkStatus.FINISH)).getCode() == 6002;
        assert workService.partTask(new WorkDTO(user1.getId(), task2.getId(), WorkStatus.TO_FINISH)).getCode() == 4000;
        assert workService.partTask(new WorkDTO(user1.getId(), task2.getId(), WorkStatus.TO_FINISH)).getCode() == 6001;
    }
```

3.2.3.3 TaskService

测试编号：0

测试方法：taskService.issueTask

Java 复制代码

测试编号：1

测试方法：taskService.issueTask

Java 复制代码

测试编号：2

测试方法：taskService.findTaskByTaskIdList

Java 复制代码

测试编号：3

测试方法：taskService.findAllTasks

Java 复制代码

测试编号：4

测试方法：taskService.selectTaskByLabel

Java 复制代码

3.2.3.4 ReportService

测试编号：0

测试方法：reportService.commitReport，reportService.updateReport

Java 复制代码

```java
public void testCommitAndUpdate() {
        User user0 = new User("1", "321", UserIdentity.DELIVER);
        User user1 = new User("12", "321", UserIdentity.WORKER);
        userMapper.insert(user0);
        userMapper.insert(user1);
        Task task = new Task(user0.getId(), "软工三作业检查", 10, 10, 0, System.currentTimeMillis() + 100000, "a", "b", "fej");
        taskMapper.insert(task);
        MultipartFile multipartFile = FileUtil.getMulFileByPath("src/test/java/com/collect/img/-4248c8a4e6883998.jpg");
        List<MultipartFile> images = new LinkedList<>();
        images.add(multipartFile);
        workMapper.insert(new Work(null, user1.getId(), task.getId(), WorkStatus.TO_FINISH));
        ReportDTO reportDTO = new ReportDTO(user1.getId(), task.getId(), images, "干得不错", "直接不错", "iPhone13 pro max 远峰蓝 1TB");
        reportService.commitReport(reportDTO);
        ReportVO reportVO1 = (ReportVO) reportService.lookReports(task.getId()).getData().get(0);
        int uid = user1.getId();
        int tid = task.getId();
        assert reportVO1.toString().equals("ReportVO{userId=" + uid + ", taskId=" + tid + ", images=null, note='干得不错', steps='直接不错', device='iPhone13 pro max 远峰蓝 1TB'}");
        assert reportVO1.getPaths().size() == 1;
        reportDTO.setImages(new LinkedList<>());
        reportDTO.setId(reportVO1.getId());
        reportDTO.setNote("什么垃圾");
        reportDTO.setSteps("上来就闪退");
        reportService.updateReport(reportDTO);
        reportVO1 = (ReportVO) reportService.lookReports(task.getId()).getData().get(0);
        assert reportVO1.toString().equals("ReportVO{userId=" + uid + ", taskId=" + tid + ", images=null, note='什么垃圾', steps='上来就闪退', device='iPhone13 pro max 远峰蓝 1TB'}");
        assert reportVO1.getPaths().size() == 0;
        reportImageMapper.deleteByMap(null);
        reportMapper.deleteByMap(null);
        taskMapper.deleteById(task.getId());
        userMapper.deleteById(user0.getId());
        userMapper.deleteById(user1.getId());
    }
```

测试编号：1

测试方法：reportService.lookReports

Java 复制代码

```java
public void testLookReports() {
        User user0 = new User("342", "321", UserIdentity.DELIVER);
        User user1 = new User("34", "321", UserIdentity.WORKER);
        User user2 = new User("34333", "321", UserIdentity.WORKER);
        userMapper.insert(user0);
        userMapper.insert(user1);
        userMapper.insert(user2);
        Task task = new Task(user0.getId(), "软工三作业检查", 10, 10, 0, 0L, "a", "b", "dfji");
        taskMapper.insert(task);
        workMapper.insert(new Work(null, user1.getId(), task.getId(), null));
        Report report1 = new Report(user1.getId(), task.getId(), "67大帝", "czy", "6+");
        reportMapper.insert(report1);
        workMapper.insert(new Work(null, user2.getId(), task.getId(), null));
        Report report2 = new Report(user2.getId(), task.getId(), "67大帝", "czy", "6+");
        reportMapper.insert(report2);
        List<String> list = Arrays.asList("123", "3423", "3213");
        for (String s : list) {
            reportImageMapper.insert(new ReportImage(report1.getId(), s));
        }
        List<String> list2 = Arrays.asList("123", "nimabi");
        for (String s : list2) {
            reportImageMapper.insert(new ReportImage(report2.getId(), s));
        }
        assert reportService.lookReports(task.getId()).getData().size() == 2;
        ReportVO reportVO = (ReportVO) reportService.lookReports(task.getId()).getData().get(0);
        assert reportVO.getPaths().size() == 3;
        reportVO = (ReportVO) reportService.lookReports(task.getId()).getData().get(1);
        assert reportVO.getPaths().size() == 2;
        reportImageMapper.deleteByMap(null);
        reportMapper.deleteByMap(null);
        taskMapper.deleteById(task.getId());
        userMapper.deleteById(user0.getId());
        userMapper.deleteById(user1.getId());
        userMapper.deleteById(user2.getId());
    }
```

测试编号：2

测试方法：reportService.lookReports

Java 复制代码

```java
    public void testLookReportsNotFound() {
        assert reportService.lookReports(2318290).getCode() == 4000;
    }
```

3.3 Controller层集成测试

3.3.1 测试方向

通过调用方法以及assert断言测试controller层中各方法的准确性。

3.3.2 测试准备

1. 测试方法使用@Transactional注解保证任务原子性，并且在运行完测试用例后将涉及文件操作的文件夹中残留文件删除，保证下一次测试不受影响。
2. 关闭正在运行的后端避免测试途中外部传入新数据影响测试结果。

3.3.3 测试用例

3.3.3.1 UserController

测试编号：0

测试方法：userController.register

Java 复制代码

测试编号：1

测试方法：userController.register

Java 复制代码

测试编号：2

测试方法：userController.login

Java 复制代码

3.3.3.2 ReportController

测试编号：0

测试方法：reportController.commitReport、reportController.updateReport

Java 复制代码

测试编号：1

测试方法：reportController.lookReports

Java 复制代码

3.3.3.3 TaskController

测试编号：2

测试方法：taskController.issueTask、taskController.searchPublishedTask

Java 复制代码

测试编号：3

测试方法：taskController.searchTaskForWorker

Java 复制代码

测试编号：4

测试方法：taskController.searchTaskForWorker

Java 复制代码

测试编号：5

测试方法：taskController.updateTaskInfo

Java 复制代码

测试编号：6

测试方法：taskController.issueTask

Java 复制代码

测试编号：7

测试方法：taskController.findAllTasks

Java 复制代码

测试编号：8
测试方法：taskController.selectTaskByLabel

Java 复制代码

3.3.3.4 WorkController

测试编号：0
测试方法：workController.partTask

Java 复制代码