

# 迭代三： 项目设计文档

---

## 1. 引言

### 1.1 编制目的

### 1.2 参考资料

## 2. 产品概述

## 3. 体系结构模型

### 3.1 整体架构描述

### 3.2 User模块的分解

#### 3.2.1 User模块的职责

#### 3.2.2 User模块的接口规范

### 3.3 Report模块的分解

#### 3.3.1 Report模块的职责

#### 3.3.2 Report模块的接口规范

### 3.4 Work模块的分解

#### 3.4.1 Work模块的职责

#### 3.4.2 Work模块的接口规范

### 3.5 Task模块的分解

#### 3.5.1 Task模块的职责

#### 3.5.2 Task模块的接口规范

### 3.6 其他模块

#### 3.6.1 TaskMapperWrapper

##### 3.6.1.1 TaskMapperWrapper模块的职责

#### 3.6.2 FileUtil

##### 3.6.2.1 FileUtil模块的职责

#### 3.6.3 TokenAspectJ

##### 3.6.3.1 TokenAspectJ模块的职责

## 4. 模型之间的映射

### 4.1 调用关系映射

#### 4.1.1 Controller到Service

- 4.1.1.1 UserController
- 4.1.1.2 TaskController
- 4.1.1.3 WorkController
- 4.1.1.4 ReportController

#### 4.1.2 Service到Mapper

- 4.1.2.1 UserService
- 4.1.2.2 TaskService
- 4.1.2.3 WorkService
- 4.1.2.4 ReportService

### 4.2 数据模型

#### 4.2.1 VO

- 4.2.1.1 UserVO
- 4.2.1.2 TaskVO
- 4.2.1.3 WorkVO
- 4.2.1.4 ReportVO
- 4.2.1.5 ResponseVO
- 4.2.1.6 CommentVO
- 4.2.1.7 CoworkersVO
- 4.2.1.8 RuleVO
- 4.2.1.9 RuleVO
- 4.2.1.10 AttributeVO
- 4.2.1.11 ClusterVO

#### 4.2.2 DTO

- 4.2.2.1 UserDTO
- 4.2.2.2 TaskDTO
- 4.2.2.3 WorkDTO
- 4.2.2.4 ReportDTO
- 4.2.2.5 AttributeDTO
- 4.2.2.6 CommentDTO
- 4.2.2.7 RuleDTO

#### 4.2.3 PO

- 4.2.3.1 User

4.2.3.2 Task

4.2.3.3 Work

4.2.3.4 Report

4.2.3.5 Code

4.2.3.6 ReportImage

4.2.3.7 Attribute

4.2.3.8 Comment

4.2.3.9 Coworkers

4.2.3.10 ReportCluster

## 5. 系统体系结构设计思路

# 1. 引言

## 1.1 编制目的

本报告详细完成对众包系统的详细设计，达到知道后续软件构造的目的，同时实现和测试人员及用户的沟通。本报告面向开发人员，测试人员及最终用户而编写，是了解系统的导航。

## 1.2 参考资料

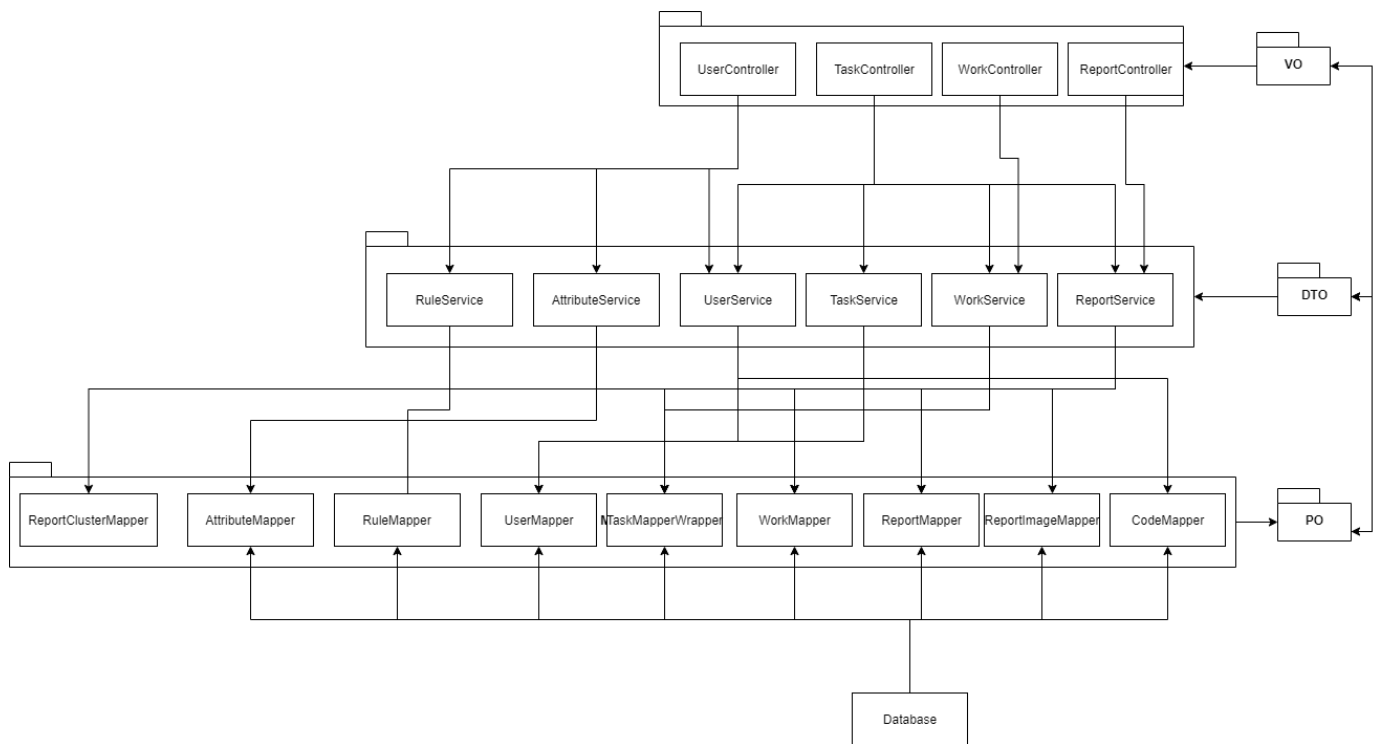
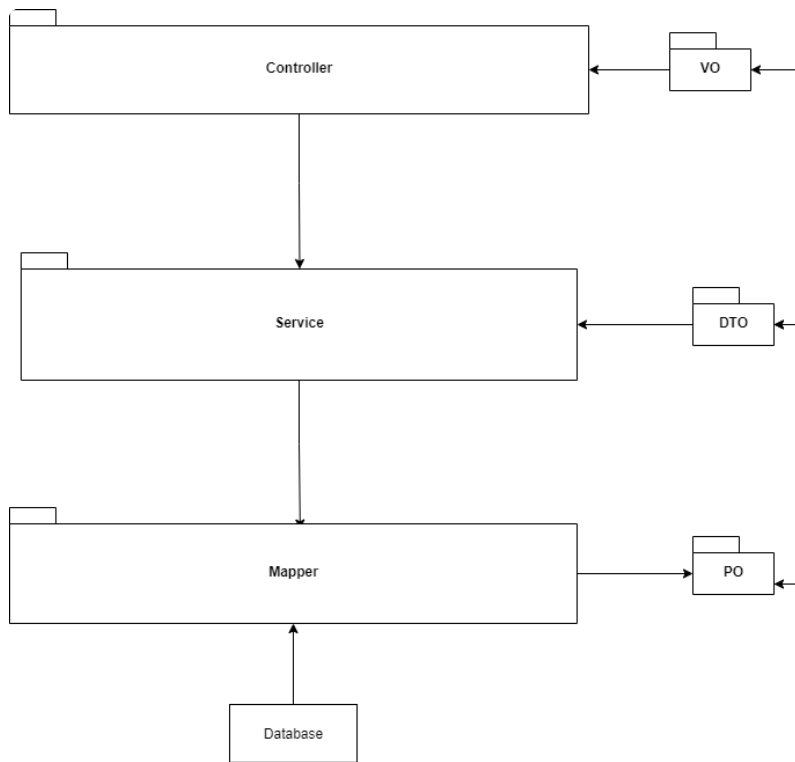
1. IEEE标准
2. 众包测试系统需求规格说明描述文档
3. 软工三课本

# 2. 产品概述

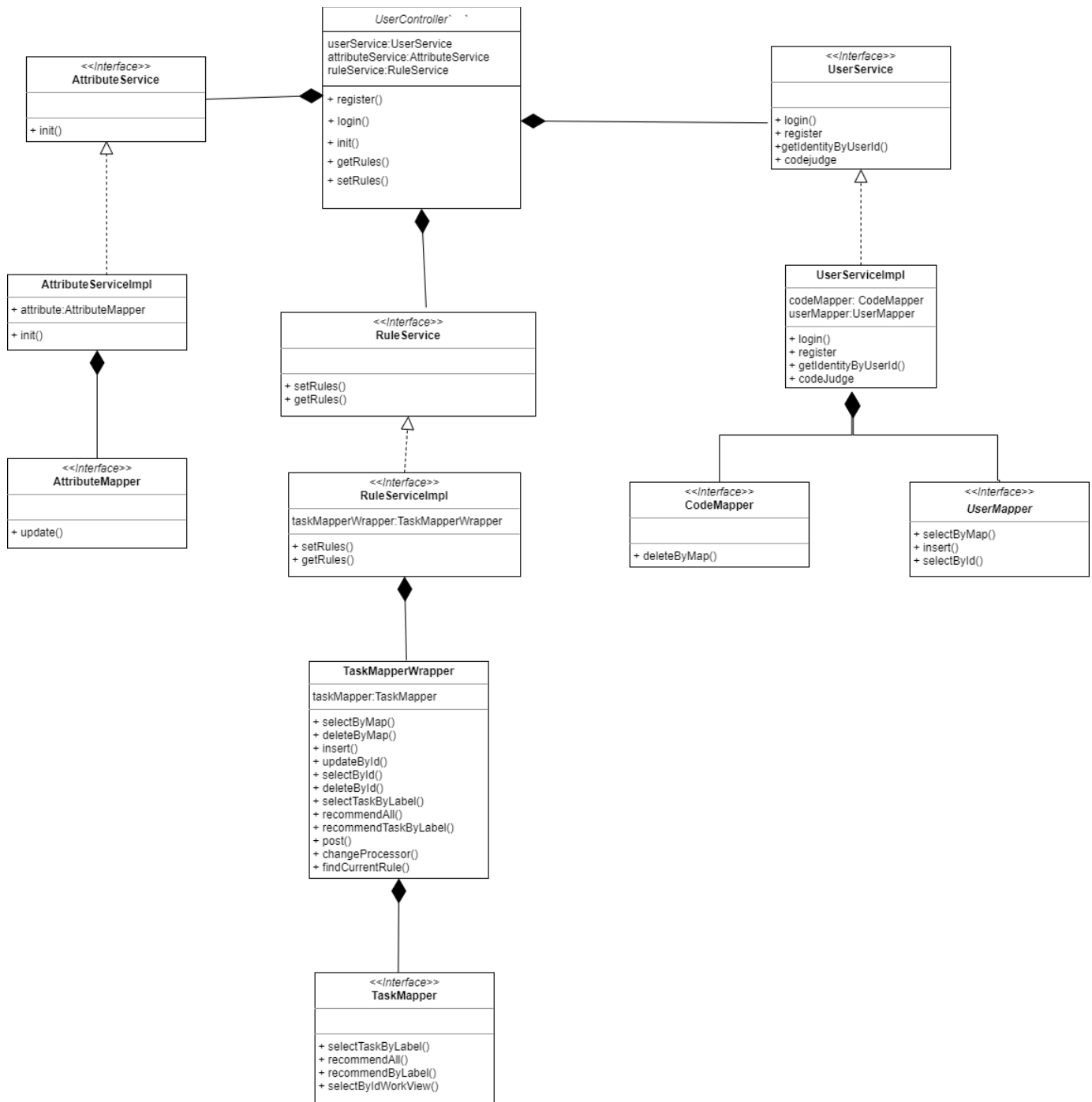
详情参考众包测试系统需求规格说明文档中对项目的概括描述。

# 3. 体系结构模型

## 3.1 整体架构描述



## 3.2 User模块的分解



类	方法	描述
UserController	login()	用户登录
	register()	用户注册
	init()	初始化用户属性
	getRules()	获得所有推荐算法
	setRules()	设置推荐算法

UserServiceImpl	login()	用户登录的实现方法
	register()	用户注册的实现方法
	getIdentityByUserId()	通过UserId获得用户身份的实现方法
	codeJudge()	判断验证码是否正确的实现方法
UserService	login()	用户登录的接口
	register()	用户注册的接口
	getIdentityByUserId()	通过UserId获得用户身份的接口
	codeJudge()	判断验证码是否正确的接口
AttributeService	init()	初始化用户属性
AttributeServiceImpl	init()	初始化用户属性
RuleService	getRules()	获得所有推荐算法
	setRules()	设置推荐算法
RuleServiceImpl	getRules()	获得所有推荐算法
	setRules()	设置推荐算法
AttributeMapper	update()	更新用户属性
CodeMapper	deleteByMap()	根据传入的hashmap对数据库进行删除
UserMapper	selectByMap()	根据传入的hashmap筛选数据库的数据并返回List<User>
	insert()	将数据拆入到数据库
	selectById	根据传入的UserId筛选数据库的数据并返回User
	selectByMap()	选择任务
	deleteByMap()	删除任务
	insert()	插入任务
	updateById()	根据id修改任务

TaskMapperWrapper	selectById()	根据id选择任务
	deleteById()	根据id删除任务
	selectTaskByLabel()	根据标签选择任务
	recommendAll()	在所有任务中推荐任务
	recommendTaskByLabel()	根据所选标签先筛选再推荐任务
	post()	推荐算法后置方法
	changeProcessor()	修改推荐算法
	findCurrentRule()	返回当前推荐算法

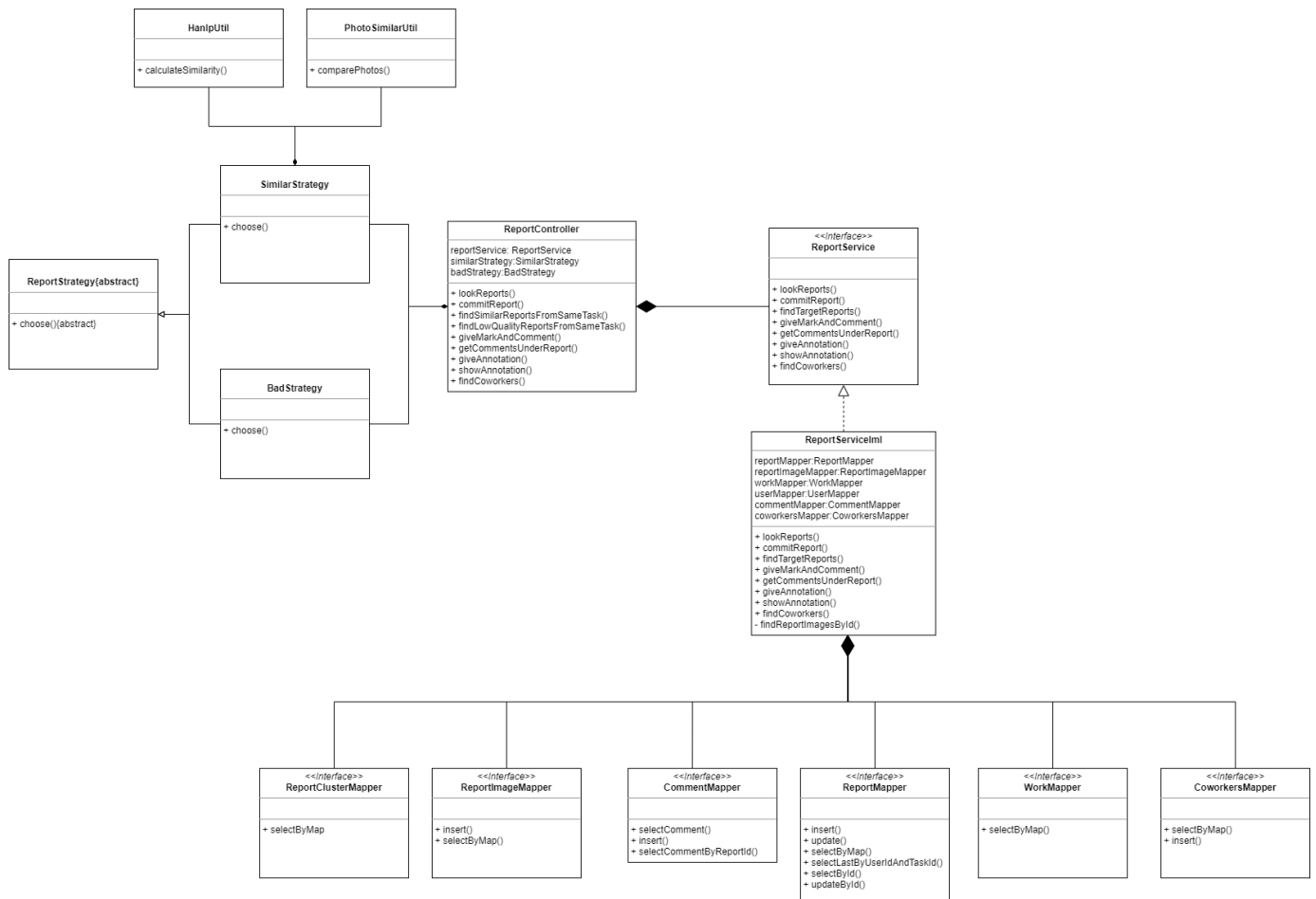
### 3.2.1 User模块的职责

提供了所有用户的登录与注册功能。

### 3.2.2 User模块的接口规范

接口名	参数	前端需传入参数	功能
login()	UserVo	email (String) password (String)	根据传入的参数判断登陆是否成功
register()	UserVo	emai (String) password (String) identity (Integer) code (String)	并根据identity的值判断是否需要验证code，完成注册操作
getRules()			
setRules()	RuleVOS	rulevos (list)	返回修改后的算法配置

## 3.3 Report模块的分解



类	方法	描述
ReportController	lookReports()	查找任务下所有工人提交报告的最后版本列表
	findSimilarReportsFromSameTask()	查找来自该报告任务的与该报告最相似的9个报告
	commitReport()	用户提交与更新报告
	findLowQualityReportsFromSameTask()	查找来自该报告任务的质量最低的9个报告
	giveMarkAndComment()	给对应报告添加打分和评价
	getCommentsUnderReport()	获取该报告的得分和评论
	giveAnnotation()	给该报告添加批注
	showAnnotation()	返回该报告的批注
	findCoworkers()	返回该报告的协作者



ReportService	lookReports()	查找任务下所有工人提交报告的最后版本列表
	findTargetReports()	查找与该报告有关的目标报告列表
	commitReport()	用户提交与更新报告的接口
	giveMarkAndComment()	获取该报告的得分和评论
	getCommentsUnderReport()	获取该报告的得分和评论
	giveAnnotation()	给该报告添加批注
	showAnnotation()	返回该报告的批注
	findCoworkers()	返回该报告的协作者
ReportServiceImpl	lookReports()	查找任务下所有工人提交报告的最后版本列表
	findTargetReports()	查找与该报告有关的目标报告列表
	commitReport()	用户提交与更新报告的接口
	giveMarkAndComment()	获取该报告的得分和评论
	getCommentsUnderReport()	获取该报告的得分和评论
	giveAnnotation()	给该报告添加批注
	showAnnotation()	返回该报告的批注
	findCoworkers()	返回该报告的协作者
ReportImageMapper	insert()	像数据库中插入数据
	selectByMap()	根据传入的hashmap筛选数据库中的数据并返回 List<Report>
CommentMapper	selectComment()	通过userId和reportId来查找评论
	insert()	插入一条评论
	selectCommentByReportId()	通过reportId来查找评论

ReportMapper	insert()	像数据库拆入数据
	updateById()	根据reportId修改数据库中的内容
	selectByMap()	根据传入的hashmap筛选数据库并返回list<report>
	update()	修改数据库中的内容
	selectById()	根据传入的reportId得到对应的报告
	selectLastByUserIdAndTaskId()	查找对应用户和任务的最新报告
WorkMapper	selectByMap()	根据传入的hashmap筛选数据库并返回list<work>
CoworkersMapper	insert()	插入一条协作关系
	selectByMap()	根据传入的hashmap筛选数据库并返回list<coworker>
ReportClusterMapper	selectBuMap()	查找相应任务的报告聚类结果

### 3.3.1 Report模块的职责

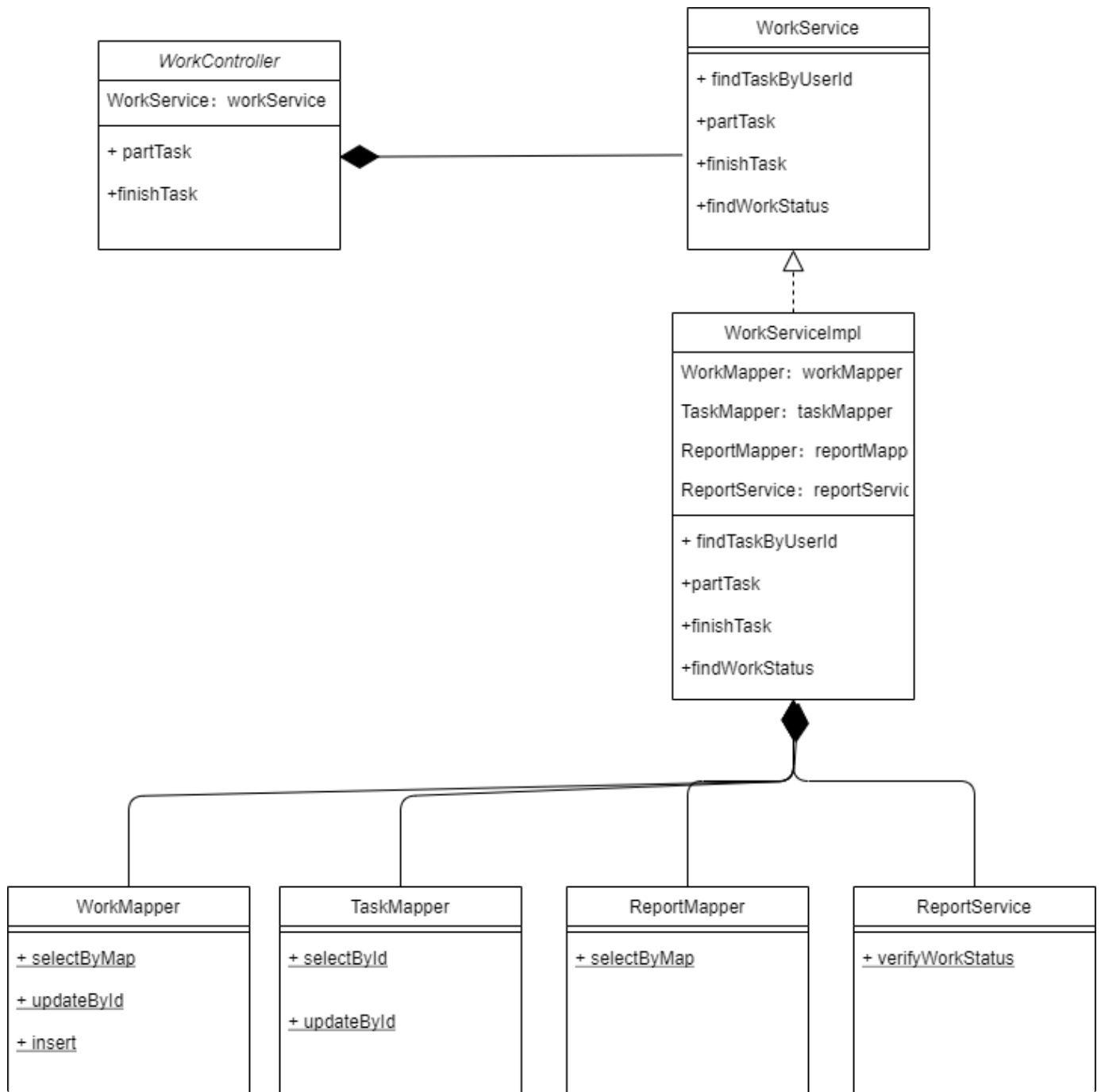
包括了任务报告相关的功能，有提交、更新、查找功能。

### 3.3.2 Report模块的接口规范

接口名	参数	前端需传入参数	功能
lookReports()	taskId(Integer)	taskId(Integer)	获得该任务的所有报告内容
findSimilarReportsFromSameTask()	reportId(Integer)	reportId(Integer)	查找来自该报告任务的与该报告最相似的9个报告
commitReport()	reportVO	userId(Integer) taskId(Integer) images(List<MultiPar	用户提交报告

		tFile>) note(String) steps(String) device(String)	
findLowQualityReportsFromSameTask()	reportId(Integer)	reportId(Integer)	查找来自该报告任务的质量最低的9个报告
giveMarkAndComment()	commentVO	reportId(Integer) userId(Integer) mark(Double) comment(String) name(String) email(String)	给对应报告添加打分和评价
getCommentsUnderReport()	reportId(Integer)	reportId(Integer)	获取该报告的得分和评论
giveAnnotation()	reportVO	userId(Integer) taskId(Integer) images(List<MultiPartFile>) note(String) steps(String) device(String)	给该报告添加批注
showAnnotation()	taskId(Integer) userId(Integer)	taskId(Integer) userId(Integer)	返回该报告的批注
findCoworkers()	taskId(Integer) userId(Integer)	taskId(Integer) userId(Integer)	返回该报告的协作者

### 3.4 Work模块的分解



类	方法	职责
WorkController	partTask	为工人分配任务
	finishTask	设置工人任务状态为完成
WorkService	findTaskByUserId	众包工人根据工人id和任务的状态获得他领取的任务id
	partTask	为工人分配任务
	finishTask	设置工人任务状态为完成
	findWorkStatus	查询工作状态
WorkServiceImpl	findTaskByUserId	众包工人根据工人id和任务的状态获得他领取的任务id
	partTask	为工人分配任务
	finishTask	设置工人任务状态为完成
	findWorkStatus	查询工作状态
WorkMapper	selectByMap	通过用户id查询任务
	updateById	更新工作状态
	insert	插入一条任务领取记录
TaskMapper	selectById	查询任务
	updateById	更新任务
ReportMapper	selectByMap	查询报告情况
ReportService	verifyWorkStatus	验证工作状态

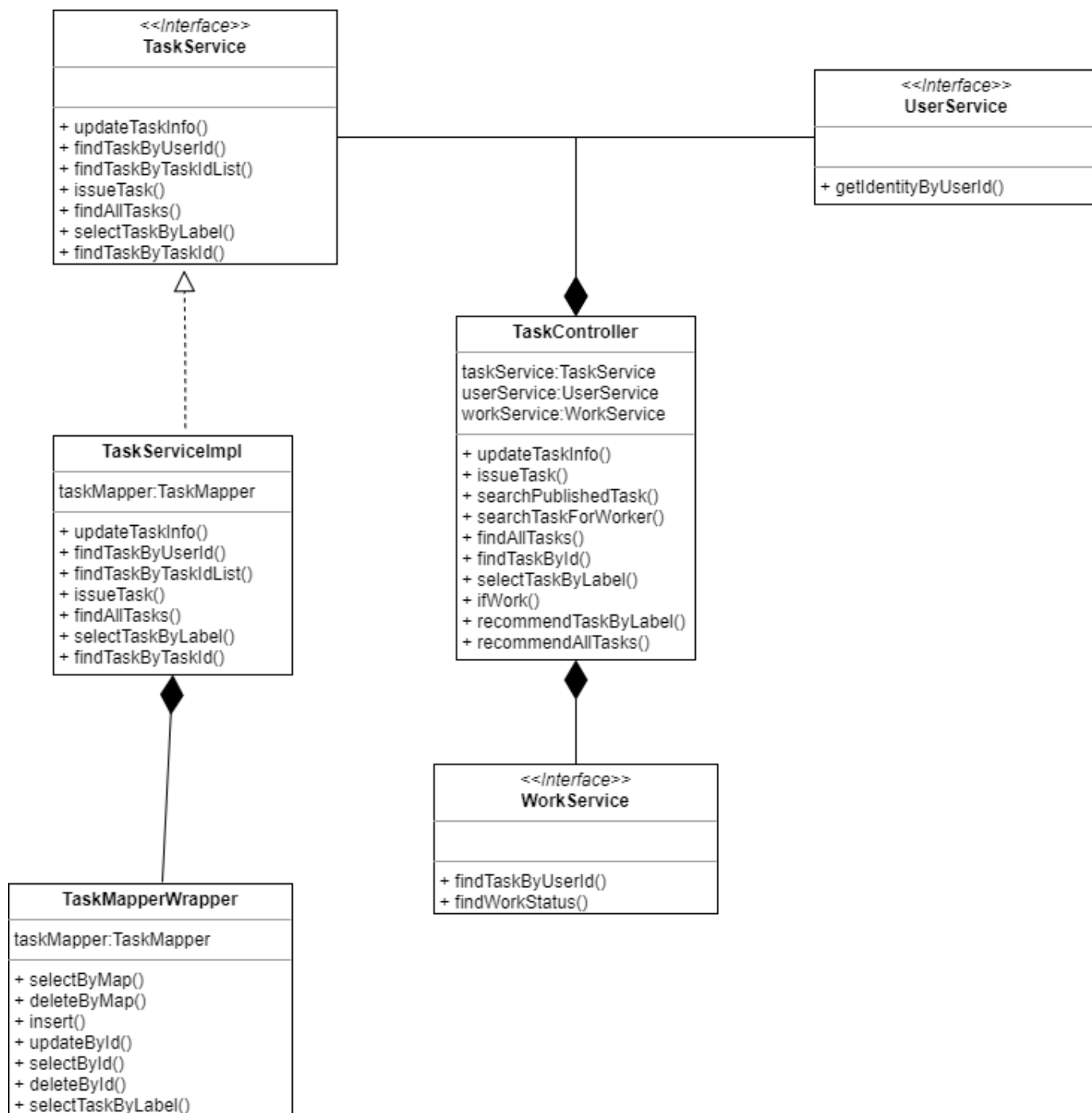
### 3.4.1 Work模块的职责

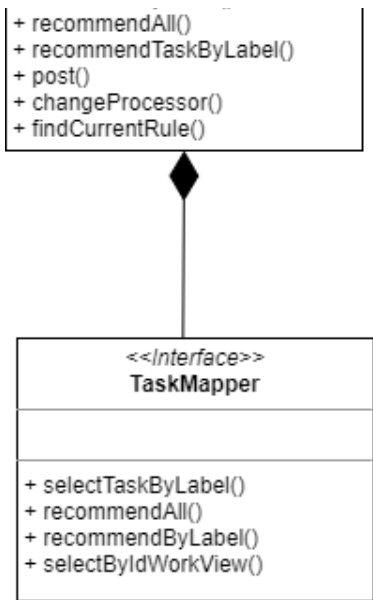
包括了任务领取相关的功能，有领取任务、更新任务状态、查找任务状态功能。

### 3.4.2 Work模块的接口规范

接口名	参数	前端需传入参数	功能
partTask	WorkVO	userId(Integer) taskId(Integer)	为工人分配任务
finishTask	WorkVO	userId(Integer) taskId(Integer)	设置工人已完成任务

### 3.5 Task模块的分解





类	方法	描述
TaskController	updateTaskInfo()	更新任务信息
	issueTask()	发布任务
	searchPublishedTask()	发包方查看自己发布的任务
	searchTaskForWorker()	众包工人查看自己参加的任务
	findAllTasks()	查找所有的任务
	findTaskById()	根据taskId获得任务详情
	selectTaskByLabel()	根据标签与名称查找任务
	ifWork()	众包工人查看任务详情，判断是否参加任务
	recommendAllTasks()	在所有任务中推荐任务
	recommendTaskByLabel()	先根据标签筛选再推荐任务
TaskService	updateTaskInfo()	更新任务的接口
	findTaskByUserId()	根据用户ID查找任务的接口
	findTaskByTaskIdList()	根据任务id列表获得任务内容的接口
	issueTask()	发布任务的接口
	findAllTasks()	获得所有任务的接口

	TaskService	发布任务接口
	selectTaskByLabel()	根据标签与名称查找任务的接口
	findTaskByTaskId()	根据taskId查找任务的接口
TaskServiceImpl	updateTaskInfo()	更新任务的实现
	findTaskByUserId()	根据用户ID查找任务的实现
	findTaskByTaskIdList()	根据任务id列表获得任务内容的实现
	issueTask()	发布任务的实现
	findAllTasks()	获得所有任务的实现
	selectTaskByLabel()	根据标签与名称查找任务的实现
	findTaskByTaskId()	根据taskId查找任务的实现
TaskMapperWrapper	selectByMap()	选择任务
	deleteByMap()	删除任务
	insert()	插入任务
	updateById()	根据id修改任务
	selectById()	根据id选择任务
	deleteById()	根据id删除任务
	selectTaskByLabel()	根据标签选择任务
	recommendAll()	在所有任务中推荐任务
	recommendTaskByLabel()	根据所选标签先筛选再推荐任务
	post()	推荐算法后置方法
	changeProcessor()	修改推荐算法

### 3.5.1 Task模块的职责

包括了任务相关的功能，有发布任务、更新任务、查找任务等功能。

### 3.5.2 Task模块的接口规范

接口名	参数	前端需传入参数	功能
-----	----	---------	----

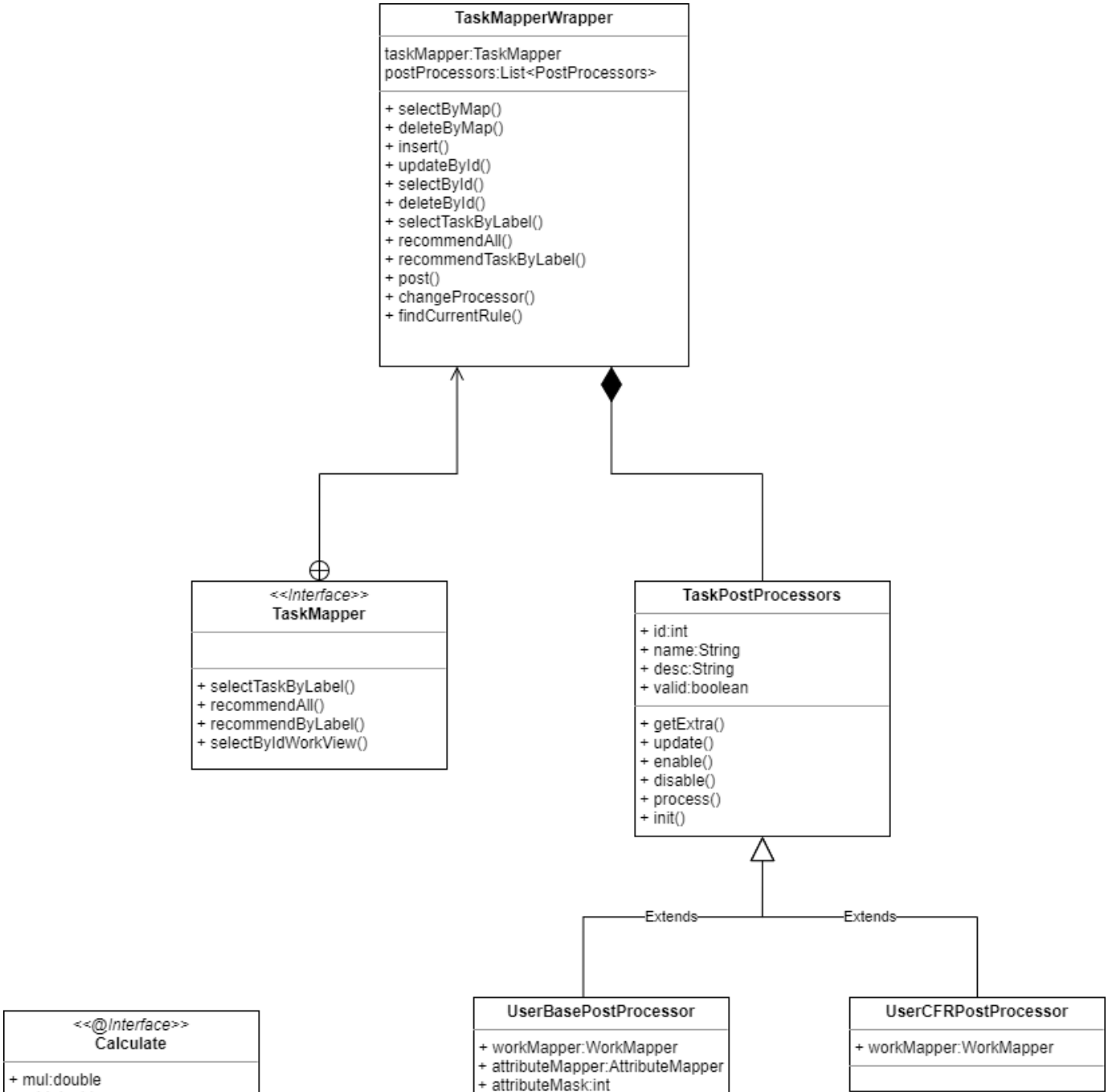


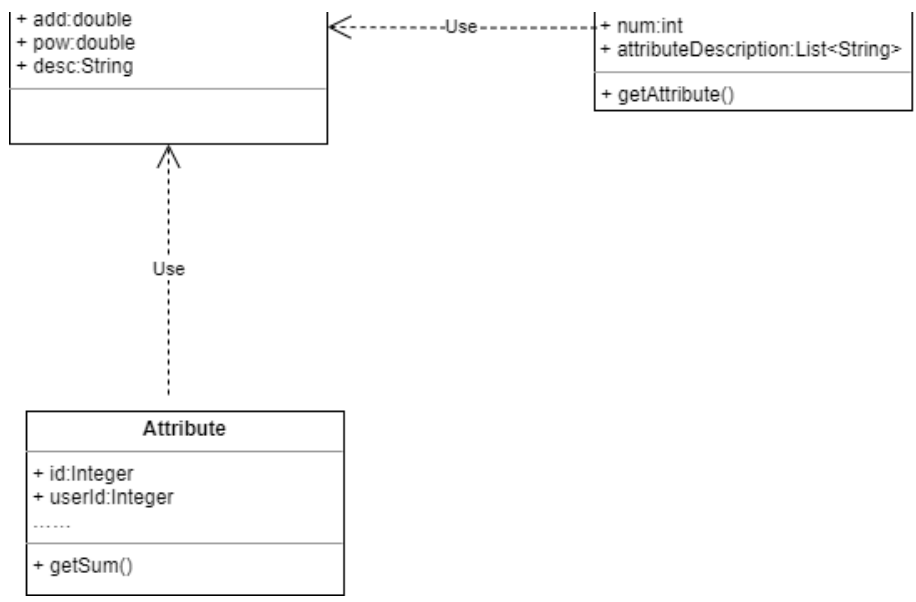
updateTaskInfo()	TaskVO	userId(Integer) name(String) number(Integer) tag(Integer) date(Long) introduction(String) apk(MultiFile) pdf(MultiFile) deleteApk(Boolean) deletePdf(Boolean)	更新任务信息
issueTask()	TaskVO	userId(Integer) name(String) number(Integer) tag(Integer) date(Long) introduction(String) apk(MultiFile) pdf(MultiFile)	创建任务
searchPublishedTask() ( )	userId(Integer)	userId(Integer)	发包方看到自己的发布任务
searchTaskForWorker() r()	userId(Integer)	userId(Integer)	众包工人查看自己的工作
findAllTasks()	无	无	返回所有的任务
findTaskById()	userId(Integer) taskId(Integer)	userId(Integer) taskId(Integer)	查看任务详情，如果 userId = task的 userId，则返回全部信息
selectTaskByLabel()	tag(Integer) if_finished(Integer) name(String)	tag(Integer) if_finished(Integer) name(String)	根据标签与名称查找任务
ifWork()	userId(Integer) taskId(Integer)	userId(Integer) taskId(Integer)	众包工人查看任务详情，判断是否参加任务
recommendByLabel()	tag(Integer)	tag(Integer)	根据筛选条件推荐任

	if_finished(Integer) userId(Integer)	if_finished(Integer)	务，其中user_id为后端自动注入
recommendAllTasks()	userId(Integer)		在所有任务中推荐任务，其中user_id为后端自动注入

### 3.6 其他模块

#### 3.6.1 TaskMapperWrapper





TaskMapperWrapper	selectByMap()	选择任务
	deleteByMap()	删除任务
	insert()	插入任务
	updateById()	根据id修改任务
	selectById()	根据id选择任务
	deleteById()	根据id删除任务
	selectTaskByLabel()	根据标签选择任务
	recommendAll()	在所有任务中推荐任务
	recommendTaskByLabel()	根据所选标签先筛选再推荐任务
	post()	推荐算法后置方法
	changeProcessor()	修改推荐算法
TaskPostProcessors	getExtra()	获取额外属性
	update()	更新推荐算法
	enable()	启动推荐算法
	disable()	关闭推荐算法
	process()	进行推荐运算
	init()	初始化
UserBasePost	getAttribute()	获取用户相关属性

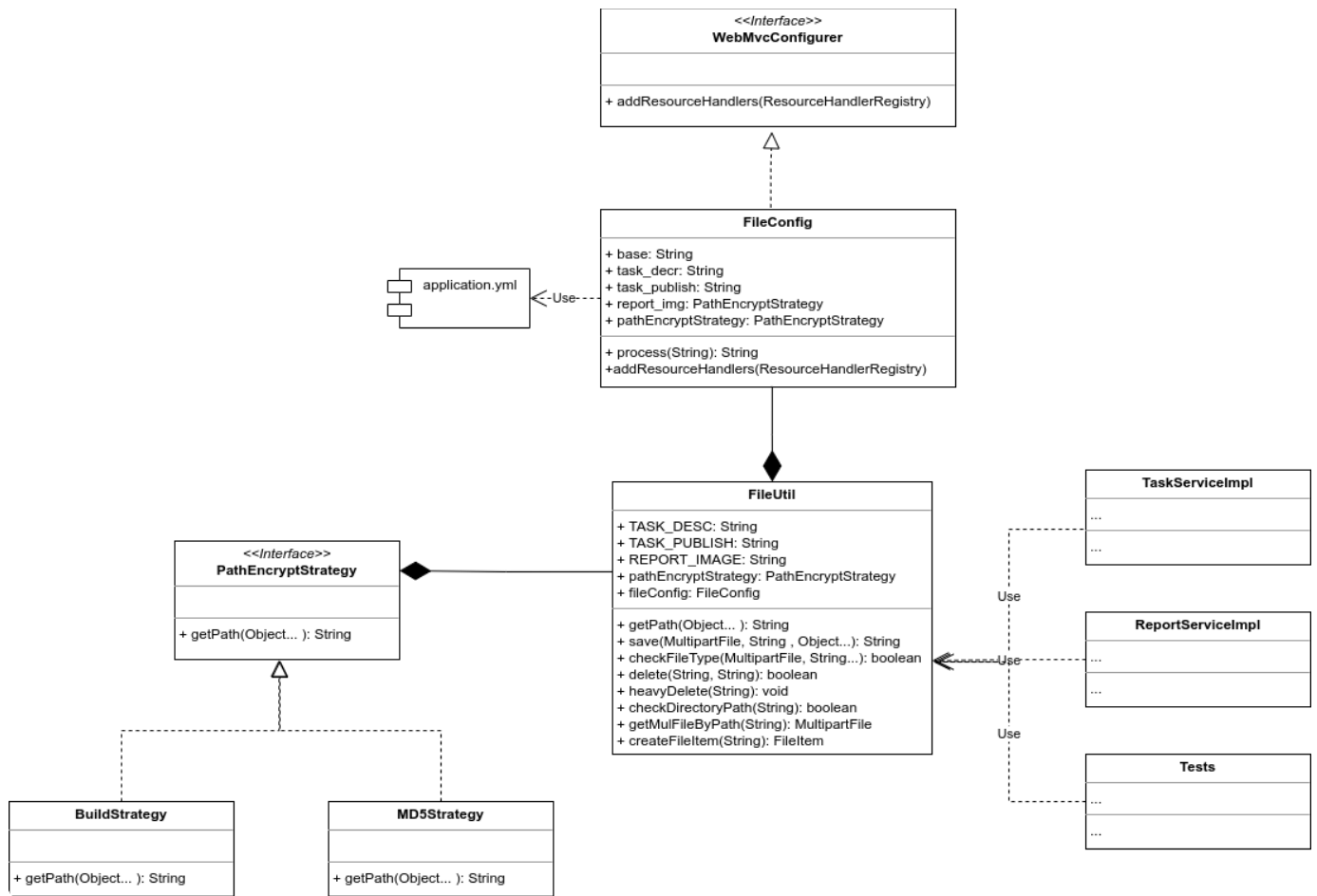
### 3.6.1.1 TaskMapperWrapper模块的职责

通过组合的方式实现装饰器模式，对TaskMapper进行封装，为后续推荐返回任务提供了便利。

其中TaskPostProcessors为推荐算法父类，其子类实现了具体的推荐算法。

Calculate为自定义注解，用于attribute类中属性，定义了该属性在UserBasePostProcessor中的权重。

### 3.6.2 FileUtil



类	方法	描述
FileUtil	TASK_DESC	任务描述附件存储地址
	TASK_PUBLISH	任务测试文件存储地址
	REPORT_IMAGE	报告附件存储地址
	pathEncryptStrategy	存储地址加密策略
	fileConfig	配置类
	save()	存储文件
	checkFileType()	检查文件类型
	delete()	删除文件
	heavyDelete()	完全删除文件夹及其下文件
	checkDirectoryPath()	检查该路径在系统中是否存在， 若不存在则创建
	getMulFileByPath()	将文件转化为MultipartFile格式
	createFileItem()	将文件转化为FileItem

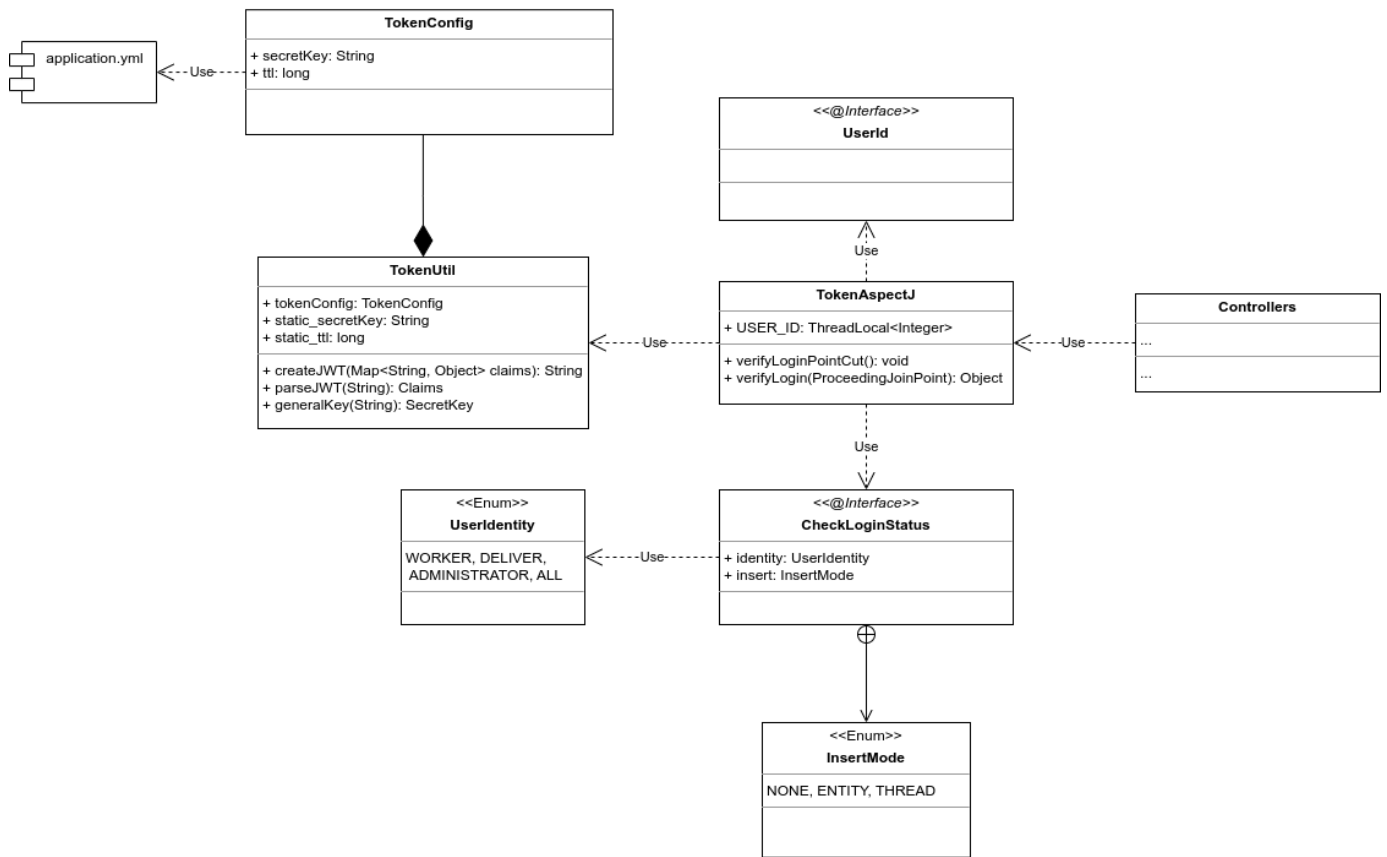
类	方法	描述
PathEncryptedStrategy	getPath()	将路径拼接为加密路径

类	方法	描述
FileConfig	base	文件存储根目录
	task_decr	任务描述附件存储地址
	task_publish	任务测试文件存储地址
	report_img	报告附件存储地址
	pathEncryptStrategy	存储地址加密策略
	getPath()	预处理path，防止path不符合命名规则
	addResourceHandlers()	将开放路径映射到准备的文件夹

### 3.6.2.1 FileUtil模块的职责

为其他模块提供文件相关的增删改查功能，使用组合配置类的方式使用yml配置其中参数

### 3.6.3 TokenAspectJ



类	方法	描述
TokenAspectJ	USER_ID	ThreadLocal 存储 userId
	verifyLoginPointCut()	切点
	verifyLogin()	检查登录状况



类	方法	描述
TokenUtil	staticKey	加密密钥
	staticTTI	token过期时间
	tokenConfig	配置类
	createJWT()	生成JWT
	parseJWT()	解析JWT
	generalKey()	由字符串生成加密key

类	方法	描述
TokenConfig	secretKey	加密密钥
	ttd	token过期时间

类	方法	描述
@CheckLoginStatus	identity()	检验用户身份
	insert()	决定用户userId注入方式
InsertMode	NONE	不注入
	ENTITY	注入controller层参数
	THREAD	注入ThreadLocal
UserIdentity	WORKER	众包工人
	DELIVER	发包方
	ADMINISTRATOR	管理员
	ALL	全部

类	方法	描述
@UserId		规定userId注入参数位置

### 3.6.3.1 TokenAspectJ模块的职责

为controller层提供用户鉴权功能，使用AOP降低耦合

## 4. 模型之间的映射

### 4.1 调用关系映射

#### 4.1.1 Controller到Service

##### 4.1.1.1 UserController

UserController调用了UserService的codeJudge、register、login方法。

##### 4.1.1.2 TaskController

TaskController调用了TaskService的updateTaskInfo、issueTask、findTaskById、findAllTasks、findTaskById、selectTaskByLabel方法，调用了UserService的getIdentityById方法，调用了WorkService的findTaskById、findWorkStatus方法。

##### 4.1.1.3 WorkController

WorkController调用了WorkService的partTask、finishTask方法。

##### 4.1.1.4 ReportController

ReportController调用了ReportService的lookReports、findReport、commitReport、updateReport方法。

#### 4.1.2 Service到Mapper

##### 4.1.2.1 UserService

UserService调用了UserMapper的selectById、insert、selectById方法，调用了CodeMapper的deleteById方法。

##### 4.1.2.2 TaskService

TaskService调用了TaskMapperWrapper的insert、updateById、selectById、selectById、selectTaskByLabel方法。

##### 4.1.2.3 WorkService

WorkService调用了WorkMapper的selectById、updateById、insert方法，taskMapper的selectById、updateById方法，ReportMapper的selectById方法。

##### 4.1.2.4 ReportService

ReportService调用了WorkMapper的selectById、updateById方法，ReportMapper的insert、updateById、selectById方法，ReportImageMapper的selectById、deleteById、insert方法，

TaskMapper的selectById方法。

## 4.2 数据模型

### 4.2.1 VO

#### 4.2.1.1 UserVO

```
1      Integer id;    //用户id
2
3      String email; //用户邮箱
4
5      String passwd; //用户密码
6
7      String code;   //注册时需要的激活码
8
9      UserIdentity userIdentity; //用户身份
10
11     String name;   //用户名
12
13     private Long lastLogin; //用户上次登录时间
14
15     private String ip;   //用户ip
```

#### 4.2.1.2 TaskVO

```
1 Integer id; //任务id
2
3 Integer userId; //发包方id
4
5 String name; //任务名称
6
7 Integer number; //任务所需测试人数
8
9 Integer remain; //任务当前还需要测试人数
10
11 Integer tag; //任务类型
12
13 Long date; //任务截止日期
14
15 String aurl; //apk的url
16
17 String purl; //pdf的url
18
19 String introduction; //任务的详细介绍
20
21 MultipartFile apk; //任务的apk
22
23 MultipartFile pdf; //任务的pdf
24
25 Boolean deleteApk; //判断apk是否要删除
26
27 Boolean deletePdf; //判断pdf是否要删除
28
29 WorkStatus workStatus; //用户的任务状态
30
31 Integer reportNum; //报告数量
```

#### 4.2.1.3 WorkVO

```
1 private Integer userId; //用户id
2
3 private Integer taskId; //任务id
4
5 private WorkStatus finish; //任务状态
```

#### 4.2.1.4 ReportVO

▼ Plain Text | 复制代码

```
1      Integer id; //报告id
2
3      Integer userId; //众包工人id
4
5      Integer taskId; //任务id
6
7      List<MultipartFile> images;    //图片文件
8
9      List<String> paths; //图片文件路径
10
11     List<String> toKeep;  //要保留的图片路径
12
13     String note;  //报错情况说明
14
15     String steps; //报错步骤
16
17     String device; //调试设备
```

#### 4.2.1.5 ResponseVO

▼ java Java | 复制代码

```
1      private Integer code;    //错误码
2
3      private String msg; //报错信息
4
5      private List<V0> data;  //传回前端的数据
```

#### 4.2.1.6 CommentVO

```
1 Integer id; //评论id
2
3 Integer reportId; //报告id
4
5 Integer userId; //评论的用户的id
6
7 Double mark; //用户评分
8
9 String comment; //用户评论
10
11 String name; //评论的用户的用户名
12
13 String email; //评论的用户的邮箱
```

#### 4.2.1.7 CoworkersVO

```
1 List<UserVO> coworkers; //参加协作的用户列表
```

#### 4.2.1.8 RuleVO

```
1 Integer id; //规则id
2
3 String hint; // 规则说明
4
5 String name; // 规则名称
6
7 Boolean valid; // 规则是否启用
8
9 Map<String, Object> extra; // 规则其他属性
```

#### 4.2.1.9 RuleVO

```
1 List<RuleVO> ruleVOS; //考虑到spring传递列表困难, 设置的对rulevo包装
```

#### 4.2.1.10 AttributeVO

```
1 Integer id; //属性id
2
3 @UserId
4 Integer userId; //对应的用户id
5
6 List<Integer> device; //传入需要初始化的设备
7
8 List<Integer> preferTask; //传入需要初始化的任务类型
```

#### 4.2.1.11 ClusterVO

```
1 List<List<Integer>> result; //聚类结果
2
3 List<String> path; //词云图路径
```

### 4.2.2 DTO

#### 4.2.2.1 UserDTO

```
1 int id;
2
3 String email;
4
5 String passwd;
6
7 UserIdentity userIdentity;
```

#### 4.2.2.2 TaskDTO

▼ Plain Text 复制代码

```
1      Integer id;  
2  
3      Integer userId;  
4  
5      String name;  
6  
7      Integer number;  
8  
9      Integer remain;  
10  
11     Integer tag;  
12  
13     Long date;  
14  
15     String aurl;  
16  
17     String purl;  
18  
19     MultipartFile apk;  
20  
21     MultipartFile pdf;  
22  
23     Boolean deleteApk;  
24  
25     Boolean deletePdf;  
26  
27     String introduction;
```

#### 4.2.2.3 WorkDTO

▼ Plain Text 复制代码

```
1      private Integer userId;  
2  
3      private Integer taskId;  
4  
5      private WorkStatus finish;
```

#### 4.2.2.4 ReportDTO



```
1      Integer id;  
2  
3      Integer userId;  
4  
5      Integer taskId;  
6  
7      List<MultipartFile> images;  
8  
9      List<String> paths;  
10  
11     List<String> toKeep;  
12  
13     String note;  
14  
15     String steps;  
16  
17     WorkStatus workStatus;
```

#### 4.2.2.5 AttributeDTO

```
1      Integer id;  
2  
3      Integer userId;  
4  
5      List<Integer> device;  
6  
7      List<Integer> preferTask;
```

#### 4.2.2.6 CommentDTO

```
1      Integer id;
2
3      Integer reportId;
4
5      Integer userId;
6
7      Double mark;
8
9      String comment;
10
11     String name;
12
13     String email;
```

#### 4.2.2.7 RuleDTO

```
1      Integer id;
2      String name;
3      String hint; // 规则说明
```

### 4.2.3 PO

#### 4.2.3.1 User

```
1      @TableId("id")
2      private Integer id;
3
4      @TableField("userId")
5      private Integer userId;
6
7      @TableField("taskId")
8      private Integer taskId;
9
10     @TableField("finish")
11     private WorkStatus finish;
```

#### 4.2.3.2 Task

Plain Text

复制代码

```
1      @TableId(value = "id", type = IdType.AUTO)
2      private Integer id;
3
4      @TableField("userId")
5      private Integer userId;
6
7      @TableField("name")
8      private String name;
9
10     @TableField("number")
11     private Integer number;
12
13     @TableField("remain")
14     private Integer remain;
15
16     @TableField("tag")
17     private Integer tag;
18
19     @TableField("date")
20     private Long date;
21
22     @TableField("aurl")
23     private String aurl;
24
25     @TableField("purl")
26     private String purl;
27
28     @TableField("introduction")
29     private String introduction;
```

#### 4.2.3.3 Work



Plain Text

复制代码

```
1      @TableId("id")
2      private Integer id;
3
4      @TableField("userId")
5      private Integer userId;
6
7      @TableField("taskId")
8      private Integer taskId;
9
10     @TableField("finish")
11     private WorkStatus finish;
```

#### 4.2.3.4 Report



Plain Text


复制代码

```
1      @TableId(value = "id", type = IdType.AUTO)
2      private Integer id;
3
4      @TableField("userId")
5      private Integer userId;
6
7      @TableField("taskId")
8      private Integer taskId;
9
10     @TableField("note")
11     private String note;
12
13     @TableField("steps")
14     private String steps;
15
16     @TableField("device")
17     private String device;
```

#### 4.2.3.5 Code



Plain Text


 复制代码

```
1      @TableId(value = "id", type = IdType.AUTO)
2      private Integer id;
3
4      @TableField("code")
5      private String code;
```

#### 4.2.3.6 ReportImage



Plain Text

 复制代码

```
1      @TableField("reportId")
2      private Integer reportId;
3
4      @TableField("image")
5      private String image;
```

#### 4.2.3.7 Attribute

```
1      @TableId(value = "id", type = IdType.AUTO)
2      private Integer id;
3
4      @TableField("userId")
5      private Integer userId;
6
7      @Calculate(desc = "windows平台参与")
8      @TableField("windows")
9      private Integer windows;
10
11     @Calculate(desc = "mac平台参与")
12     @TableField("mac")
13     private Integer mac;
14
15     @Calculate(desc = "android平台参与")
16     @TableField("android")
17     private Integer android;
18
19     @Calculate(desc = "ios平台参与")
20     @TableField("ios")
21     private Integer ios;
22
23     @Calculate(desc = "harmony平台参与")
24     @TableField("harmony")
25     private Integer harmonyos;
26
27     @Calculate(desc = "linux平台参与")
28     @TableField("linux")
29     private Integer linux;
30
31     @Calculate(desc = "性能任务参与")
32     @TableField("performance")
33     private Integer performance;
34
35     @Calculate(desc = "功能任务参与")
36     @TableField("functional")
37     private Integer functional;
38
39     @Calculate(desc = "稳定性任务参与")
40     @TableField("bug")
41     private Integer bug;
42
43     @Calculate(mul = 0.2, desc = "性能任务能力")
44     @TableField("performanceAbility")
45     private Double performanceAbility;
```

```
46
47     @Calculate(mul = 0.2, desc = "稳定性任务能力")
48     @TableField("bugAbility")
49     private Double bugAbility;
50
51     @Calculate(mul = 0.2, desc = "功能任务能力")
52     @TableField("functionalAbility")
53     private Double functionalAbility;
54
55     @Calculate(desc = "用户信誉")
56     @TableField("credibility")
57     private Double credibility;
```

#### 4.2.3.8 Comment

Java | 复制代码

```
1     @TableId(value = "id", type = IdType.AUTO)
2     Integer id;
3
4     @TableField("reportId")
5     Integer reportId;
6
7     @TableField("userId")
8     Integer userId;
9
10    @TableField("mark")
11    Double mark;
12
13    @TableField("comment")
14    String comment;
15
16    String name;
17
18    String email;
```

#### 4.2.3.9 Coworkers

```
1
2     @TableId(value = "id", type = IdType.AUTO)
3     private Integer id;
4
5     @TableField("workId")
6     private Integer workId;
7
8     @TableField("coworkerId")
9     private Integer coworkerId;
```

#### 4.2.3.10 ReportCluster

```
1
2     @TableField("taskId")
3     private Integer taskId;
4
5     @TableField("cluster_set")
6     private String clusterSet;
```

## 5. 系统体系结构设计思路

该系统服务端使用了依赖倒置原则、开闭原则、面向接口原则等原则，以及策略模式。服务端分为了Controller层、Service层、Mapper层，并且各层编程面向接口。



