

# 迭代一： 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

## 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

### 4.2.2 DTO

4.2.2.1 UserDTO

4.2.2.2 TaskDTO

4.2.2.3 WorkDTO

4.2.2.4 ReportDTO

### 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

## 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

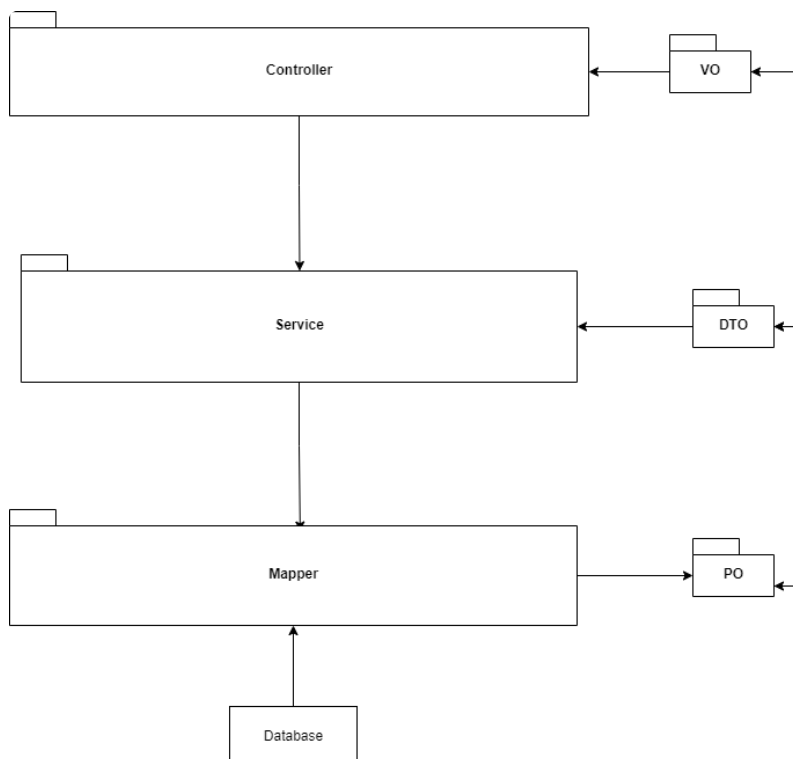
1. IEEE标准
2. 众包测试系统需求规格说明描述文档

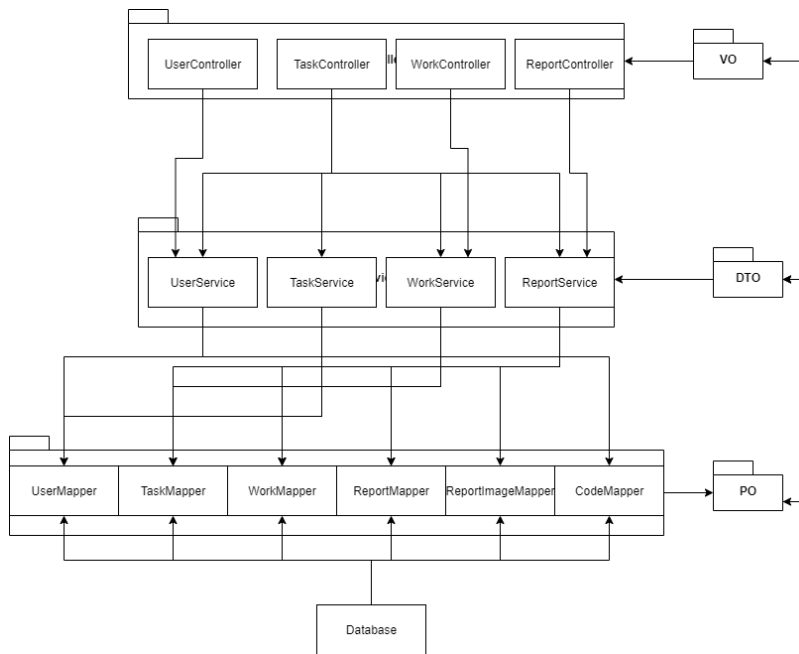
## 2. 产品概述

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。

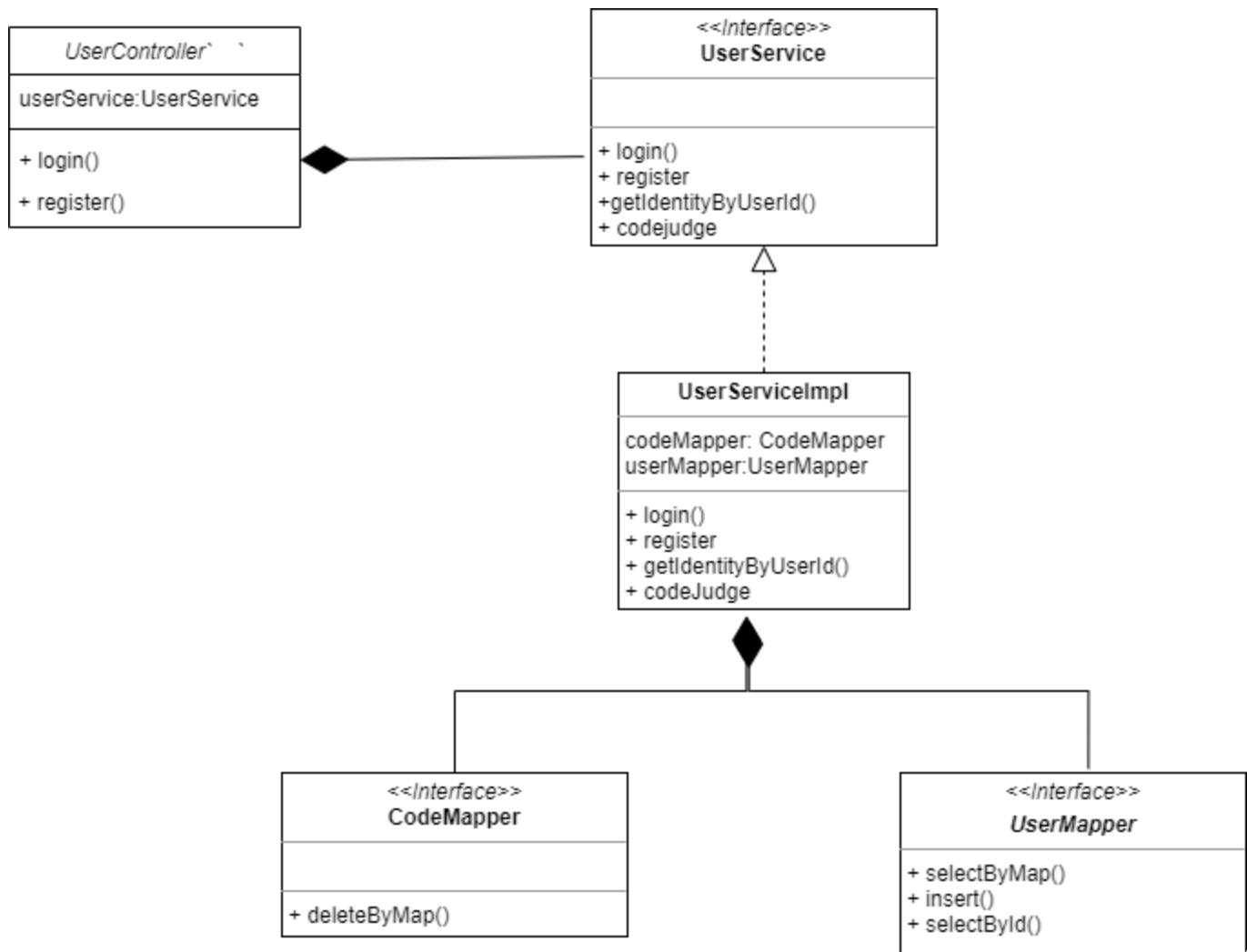
## 3. 体系结构模型

### 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
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
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

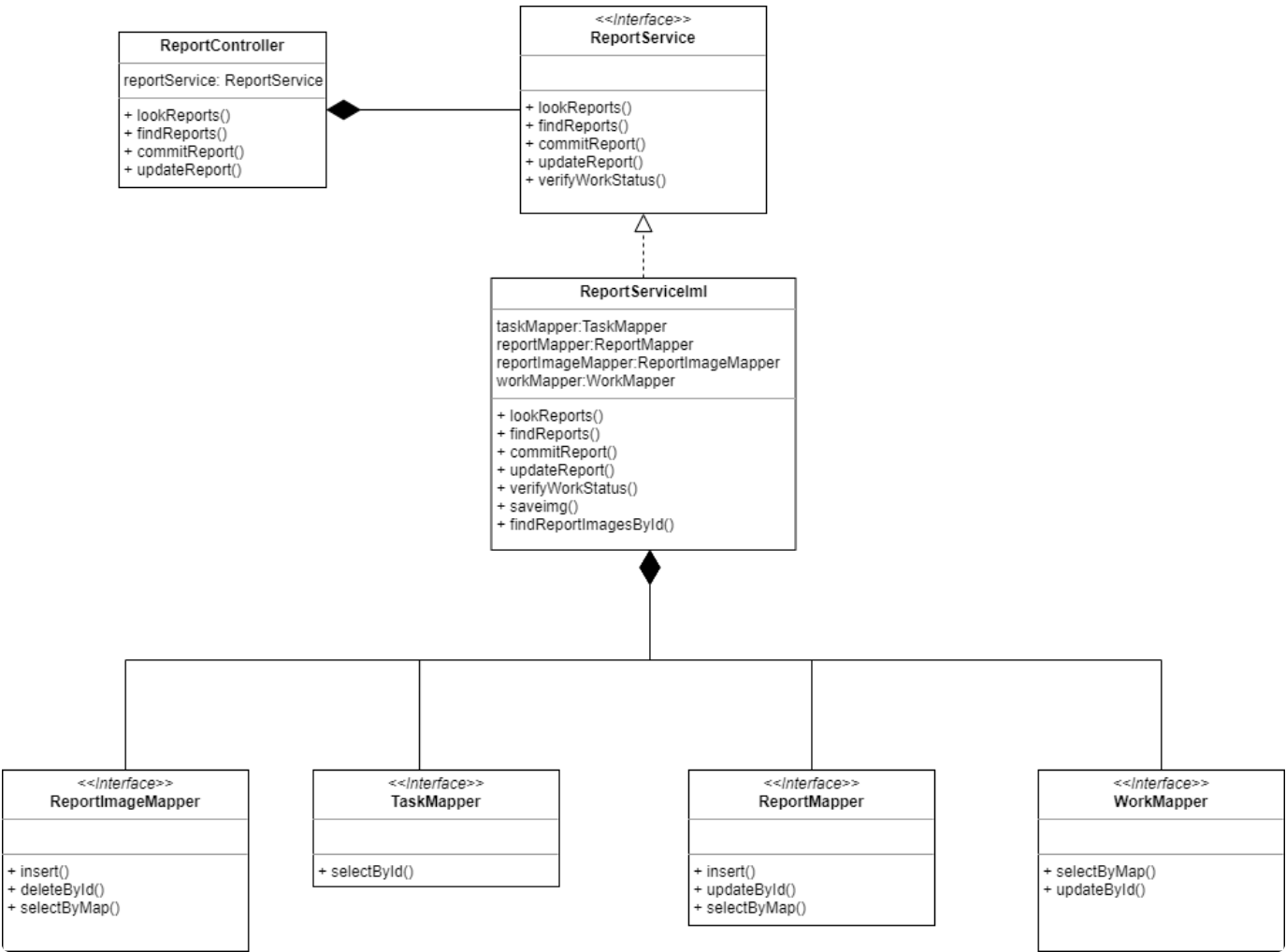
### 3.2.1 User模块的职责

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

### 3.3 Report模块的分解



类	方法	描述
	lookReports()	根据taskId返回其对应任务的所 .....

ReportController		有报告
	findReports()	根据userId和taskId返回对应任务中该用户提交的报告
	commitReport()	用户提交报告
	updateReport()	用户更新报告
ReportService	lookReports()	返回任务内所有报告的接口
	findReports()	查找用户在某任务中提交的报告的接口
	commitReport()	用户提交报告的接口
	updateReport()	用户更新报告的接口
	verifyWorkStatus()	验证工人是否接受任务以及任务是否过期的接口
ReportServiceImpl	lookReports()	返回任务内所有报告的实现方法
	findReports()	查找用户在某任务中提交的报告的实现方法
	commitReport()	用户提交报告的实现方法
	updateReport()	用户更新报告的实现方法
	verifyWorkStatus()	验证工人是否接受任务以及任务是否过期的实现方法
	findReportImagesById()	根据reportId获得该报告内的所有图片名
	saveImg()	根据传入的file将其保存
ReportImageMapper	insert()	像数据库中插入数据
	deleteById()	根据reportId删除数据库中的数据
	selectByMap()	根据传入的hashmap筛选数据库中的数据并返回 List<Report>
TaskMapper	selectById()	根据taskId筛选数据库中的数据



		并返回task
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>
WorkMapper	selectByMap()	根据传入的hashmap筛选数据库并返回list<work>
	updateById()	根据workId对数据库进行更新

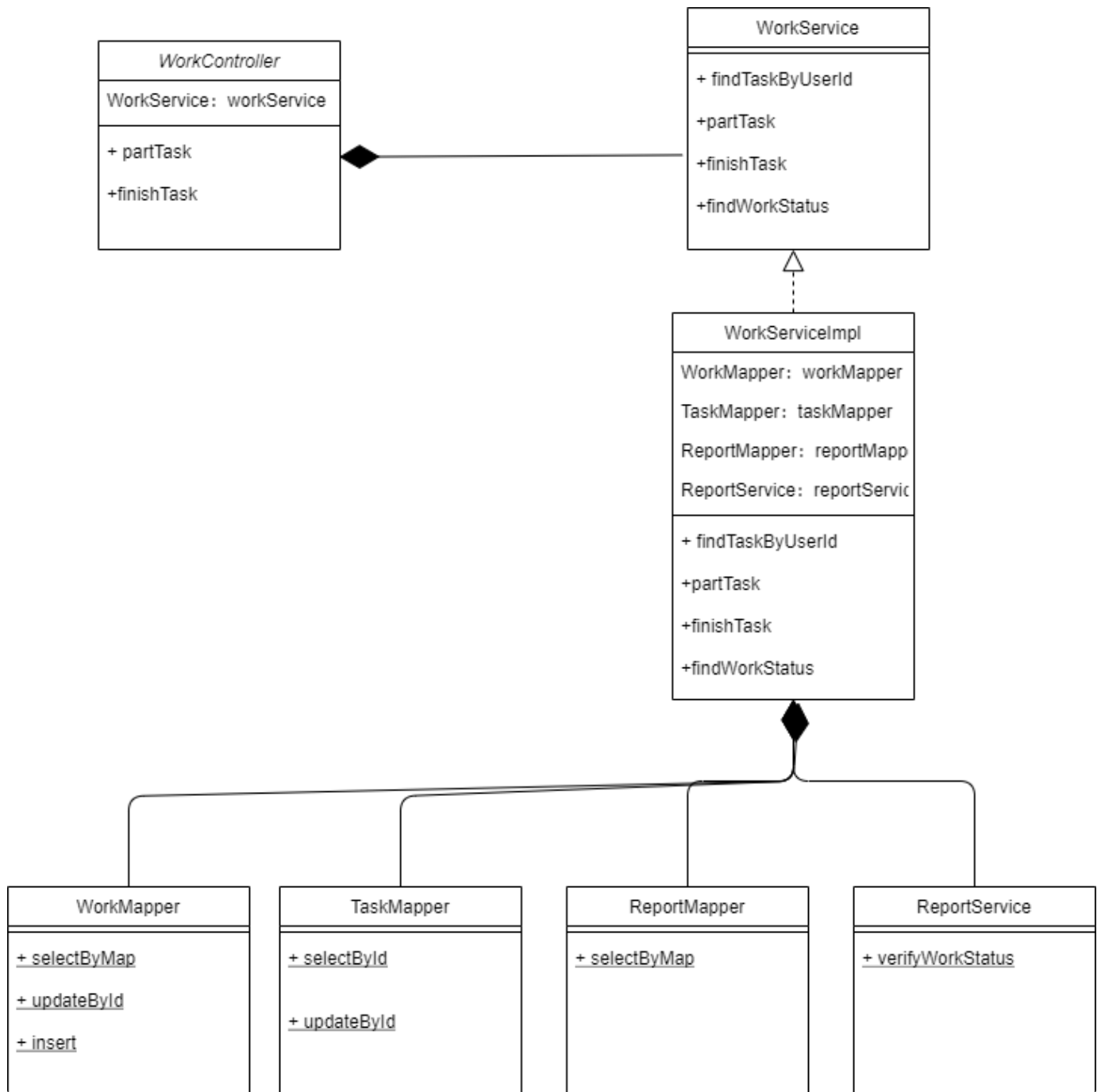
### 3.3.1 Report模块的职责

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
findReports()	userId(Integer) taskId(Integer)	userId(Integer) taskId(Integer)	获得该用户在对应任务中提交的报告
commitReport()	reportVO	userId(Integer) taskId(Integer) images(List<MultiPartFile>) note(String) steps(String) device(String)	用户提交报告
updateReport()	reportVO	userId(Integer) taskId(Integer) images(List<MultiPartFile>) toKeep(List<String>) note(String) steps(String) device(String)	用户修改报告

### 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

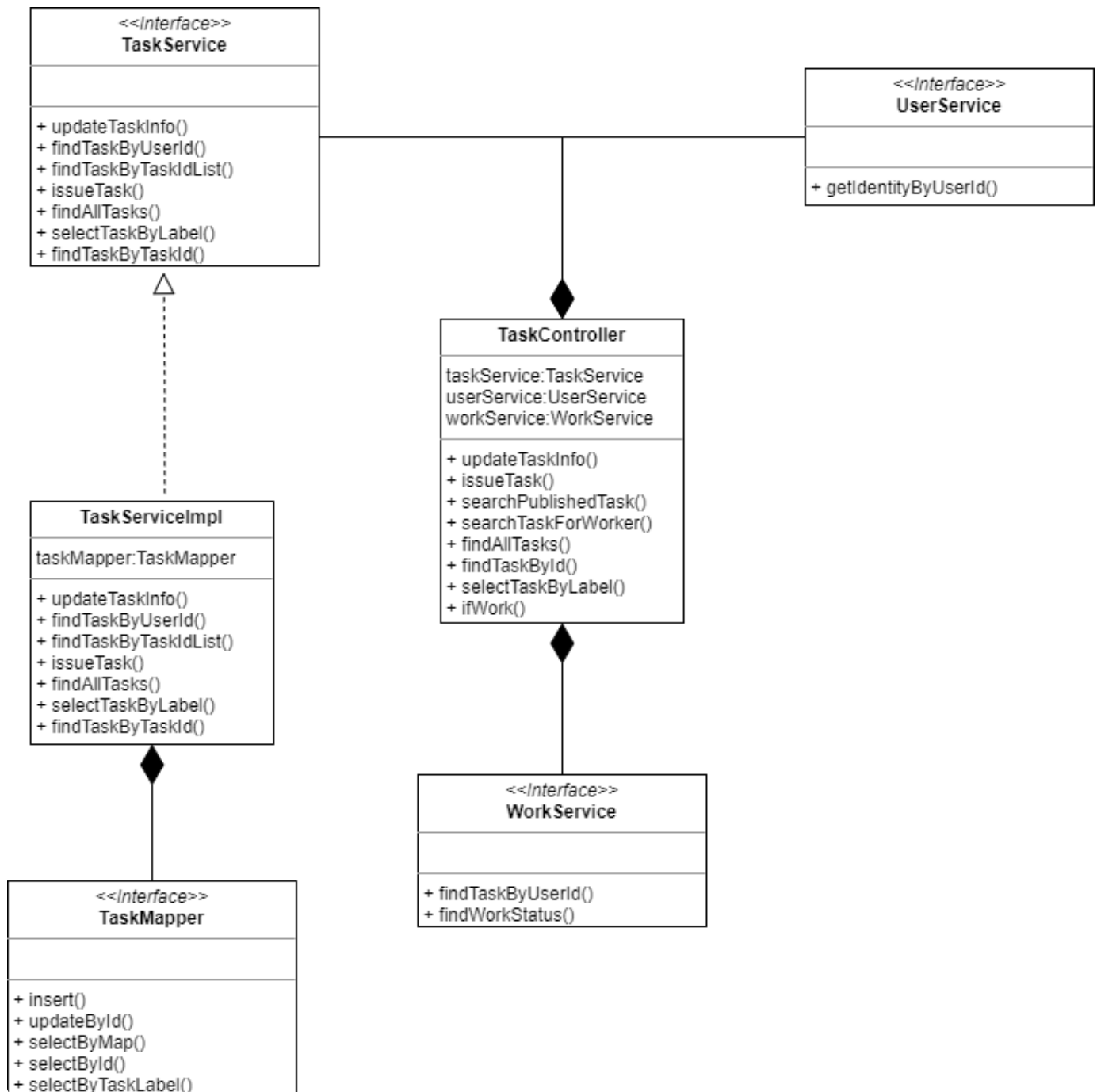
### 3.4.1 Work模块的职责

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
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
	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
TaskMapper	insert()	将数据插入数据库
	updateById()	根据Id更新数据库
	selectByMap()	根据传入的hashmap筛选数据库的数据并返回list<task>
	selectById()	根据taskId筛选数据库并返回task
	selectByTaskLabel()	根据标签与名称查找数据库并返回list<task>

### 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

## 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、findTaskByTaskId、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selectByMap、insert、selectById方法，调用了CodeMapper的deleteByMap方法。

##### 4.1.2.2 TaskService

TaskService调用了TaskMapper的insert、updateById、selectByMap、selectById、selectTaskByLabel方法。

##### 4.1.2.3 WorkService

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

##### 4.1.2.4 ReportService

ReportService调用了WorkMapper的selectByMap、updateById方法，ReportMapper的insert、updateById、selectByMap方法，ReportImageMapper的selectByMap、deleteByMap、insert方法，TaskMapper的selectById方法。


## 4.2 数据模型

### 4.2.1 VO

#### 4.2.1.1 UserVO



Plain Text

 复制代码

```
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
```

#### 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

▼ Plain Text 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
```

### 4.2.2 DTO

#### 4.2.2.1 UserDTO

▼

Plain Text | 复制代码

```
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



Plain Text

 复制代码

```
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;
```

### 4.2.3 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



```
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

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

#### 4.2.3.6 ReportImage

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

## 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。

