**春季**

16

项目设计文档

窦妍 杜天蛟 范炜 高露

目录

一、团队名称： 2

二、文档更新记录： 2

三、引言： 2

1.编写目的 2

2.对象和范围 2

4.名词和术语 3

四、系统的分层架构 3

# 一、团队名称：617team

# 二、文档更新记录：

|  |  |  |  |
| --- | --- | --- | --- |
| **版本** | **作者** | **版本描述** | **日期** |
| V1.0 | 全体成员 | 最初的设计草稿 | 2016/2/28 |
| V2.0 | 全体成员 | 修改后的迭代二设计文档 | 2016/4/14 |

# 三、引言：

## 1.编写目的

本文档提供Gitming项目迭代一的软件架构概览，采用若干架构视图描述系统的不同方面，以便表示构造系统所需要的重要架构决策。

## 2.对象和范围

本文的读者是本团队全体成员，用于指导以后的代码开发和测试工作。

3.参考文献

《软件需求规格说明书》，617Team

## 4.名词和术语

# 四、系统的分层架构

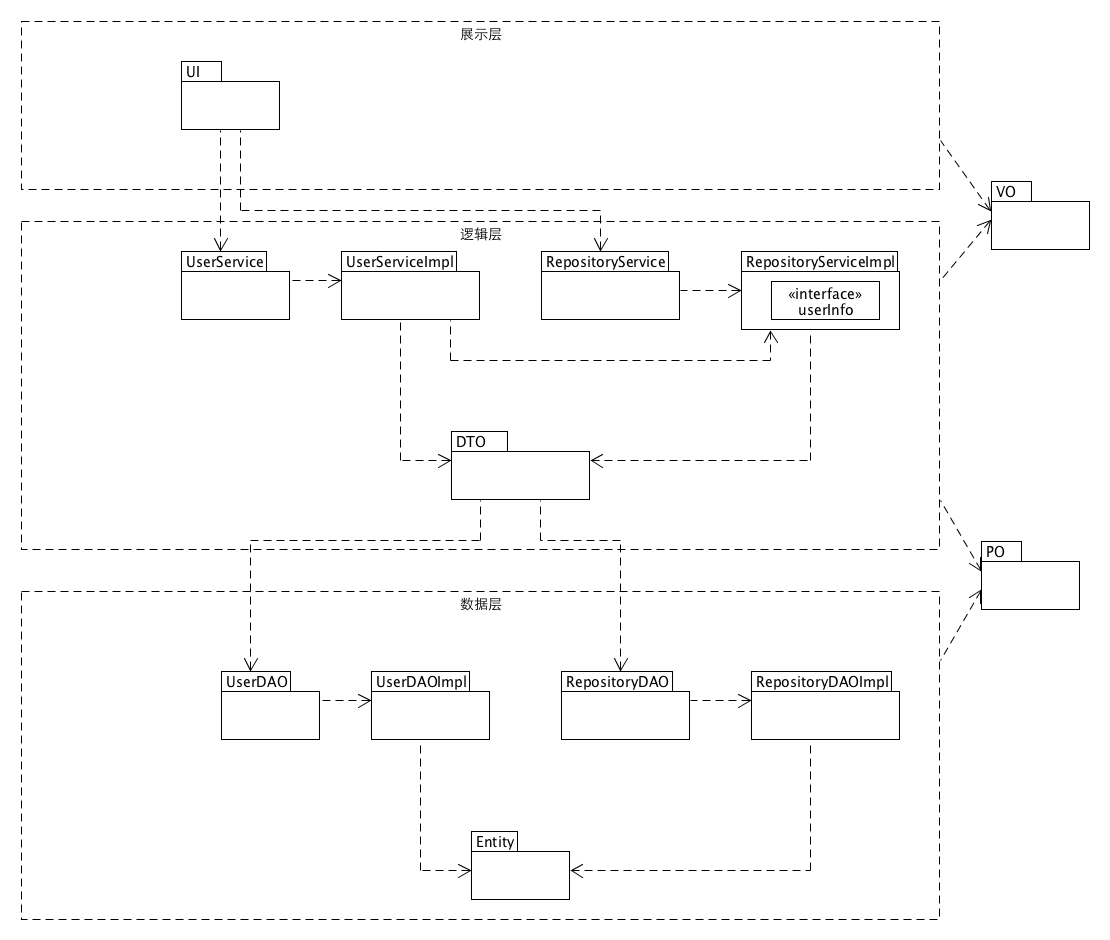
**1.**系统划分为以下4个逻辑层次。

1. 表示层：用于界面展示和配置的层次。
2. 业务层：包含业务控制和逻辑的层次。
3. 数据层：定义和存储系统中相关数据的层次。

**2.**系统的架构设计如下。

系统架构中的对象分为以下几类：

1. UI对象，负责处理系统数据的展现和用户的交互
2. Service对象，负责提供服务的抽象接口，获取从数据段组装好的数据
3. ServiceImpl对象，负责对抽象接口的实现模块
4. VO对象，为值对象，通常用于业务层之间的数据传递
5. PO对象，为持久化数据对象，可以看成是与数据库中的表相映射的java对象
6. DTO对象，负责封装从DAO获取的批量数据的接口，即完成VO与PO对象的转换
7. DAO对象，负责提供数据层服务的接口，与数据库实体交互获取数据
8. DAOImpl对象，负责对DAO的实现模块
9. Entity对象，该模块用来将从数据库中获取的数据封装成数据实体



**User模块**

|  |  |  |  |
| --- | --- | --- | --- |
| **接口ID** | **连接组件** | **接口信息** | |
| I1 | 连接UI和UserService | 语法 | Public UserVO searchUser(String id); |
| 前置条件 | 用户的输入正确 |
| 后置条件 | 根据id查找相应的用户 |
| 不变量 | 用户请求信息 |
| I2 | 连接UserServiceImpl和DTO | 语法 | Public UserVO getUser(String id); |
| 前置条件 | 传入的id正确 |
| 后置条件 | 根据id返回相应的UserVO |
| 不变量 | 无 |
| I3 | 连接DTO和UserDAO | 语法 | Public UserPO getUser(String id); |
| 前置条件 | 传入的id正确 |
| 后置条件 | 根据id返回相应的UserPO |
| 不变量 | 无 |
| I4 | 连接UserServiceImpl和RepositoryImpl | 语法 | Public ArrayList<String> getContributeRepos (String id); |
| 前置条件 | 传入的id正确 |
| 后置条件 | 根据id返回该用户参与的项目 |
| 不变量 | 无 |
| 语法 | Public ArrayList<String> getCreateRepos(String id); |
| 前置条件 | 传入的id正确 |
| 后置条件 | 根据id返回该用户创建的项目 |
| 不变量 | 无 |

**Repository模块**

|  |  |  |  |
| --- | --- | --- | --- |
| **接口ID** | **连接组件** | **接口信息** | |
| I5 | 连接UI和RepositoryService | 语法 | Public RepositoryVO searchRepository(String name); |
| 前置条件 | 项目的输入正确 |
| 后置条件 | 根据项目名查找相应的项目 |
| 不变量 | 项目请求信息 |
| I6 | 语法 | Public ArrayList<RepositoryVO> showRepositories(); |
| 前置条件 | 无 |
| 后置条件 | 返回所有项目 |
| 不变量 | 项目请求信息 |
| I7 | 语法 | Public ArrayList<RepositoryVO> sortByStar(); |
| 前置条件 | 无 |
| 后置条件 | 将项目按照star数目排序并返回列表 |
| 不变量 | 项目请求信息 |
| I8 | 语法 | Public ArrayList<RepositoryVO> sortByContributer(); |
| 前置条件 | 无 |
| 后置条件 | 将项目按照contributer数目排序并返回列表 |
| 不变量 | 项目请求信息 |
| I9 | 语法 | Public ArrayList<RepositoryVO> sortByFork(); |
| 前置条件 | 无 |
| 后置条件 | 将项目按照fork数目排序并返回列表 |
| 不变量 | 项目请求信息 |
| I10 | 连接RepositoryServiceImpl和DTO | 语法 | Public RepositoryVO getRepository(String name); |
| 前置条件 | 传入的name正确 |
| 后置条件 | 根据name返回相应的RepositoryVO |
| 不变量 | 无 |
| I11 | 语法 | Public RepositoryVO getAllRepository(); |
| 前置条件 | 无 |
| 后置条件 | 返回所有的项目 |
| 不变量 | 无 |
| I12 | 连接DTO和RepositoryDAO | 语法 | Public RepositoryPO getRepository(String name); |
| 前置条件 | 传入的name正确 |
| 后置条件 | 根据name返回相应的RepositoryPO |
| 不变量 | 无 |
| I13 | 语法 | Public RepositoryPO getAllRepository(); |
| 前置条件 | 无 |
| 后置条件 | 返回所有的项目 |
| 不变量 | 无 |