项目设计文档

*InSighter*

**更新历史**

|  |  |  |  |
| --- | --- | --- | --- |
| 修改人员 | 日期 | 变更原因 | 版本号 |
| 张诗琦 | 2016-2-28 | 最初草稿 | V1.0 |
| 董轶波 | 2016-2-29 | 加入userbl与userData模块描述 | V1.1 |
| 张诗琦 | 2016-2-29 | 修改repobl与repodata模块描述，插入逻辑包图 | V1.2 |

1. **引言** 
   1. **编制目的**

本文档详细完成对GitMining System数据分析系统的概要设计，达到指导开发的目的，同时实现测试人员及用户的沟通。

本文档面向开发人员、测试人员及最终用户编写，是了解系统的导航。

1. **产品描述**

参考GitMining System数据分析系统软件需求规格说明中对产品的概括描述。

1. **逻辑视角**

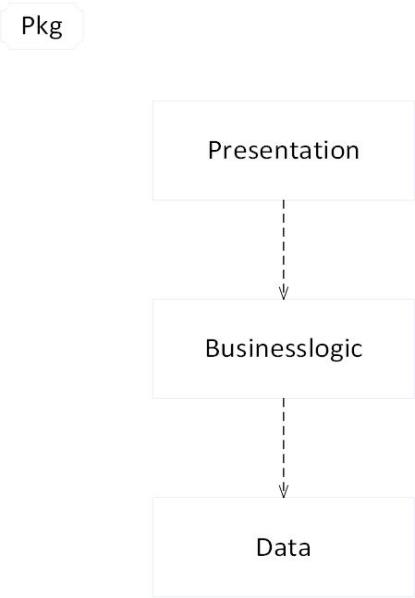
GitMining System数据分析系统中，选择了分层体系结构的风格，将系统分为3层（展示层、业务逻辑层、数据层）能够很好的示意整个高层抽象。展示层包括GUI页面的实现，业务逻辑层包含业务逻辑处理的实现，数据层负责数据的读取。分层体系结构的逻辑视角和逻辑设计方案如图1和图2所示。

图1逻辑视角

# 4.组合视角

# 4.1.开发包图

GitMining数据分析系统的最终开发包设计如图3和表1所示。

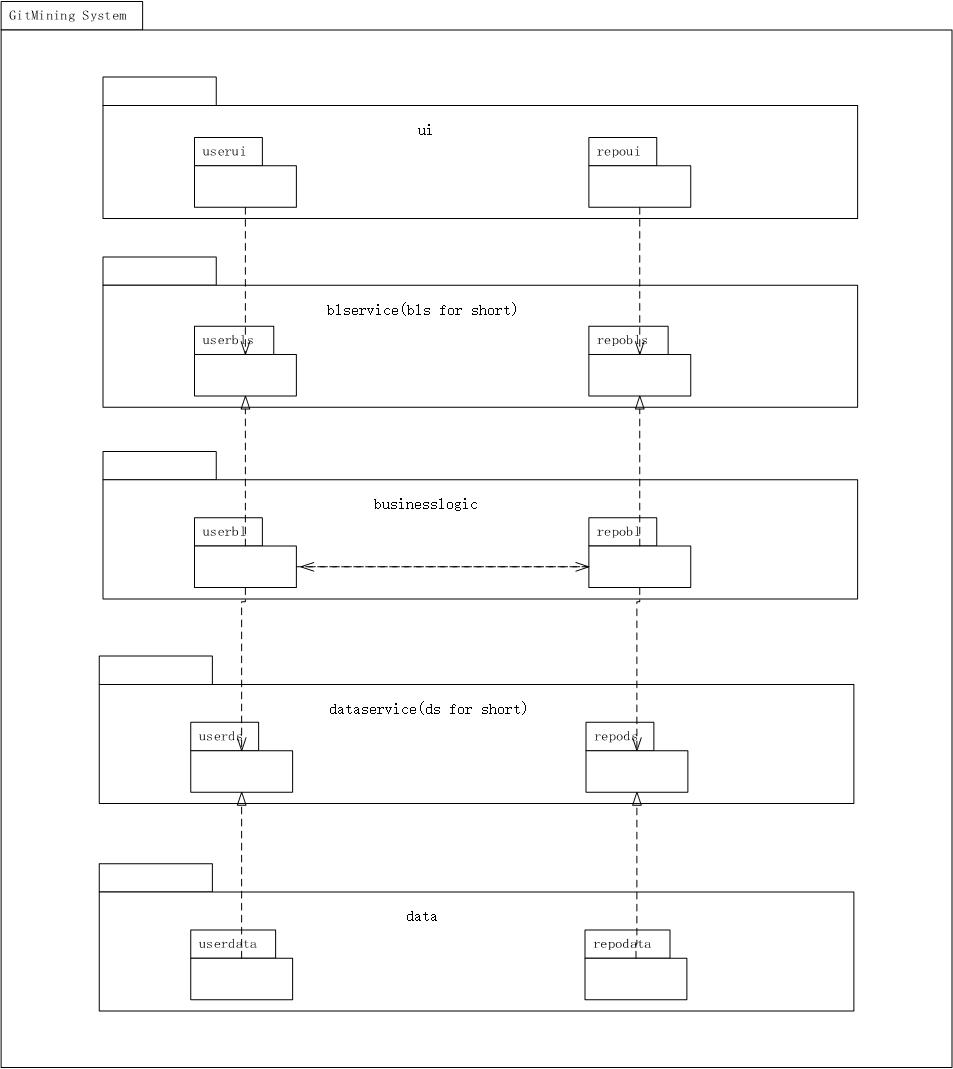


图3:开发包图

**表1层之间调用的接口**

|  |  |  |
| --- | --- | --- |
| 接口 | 服务调用方 | 服务提供方 |
| userblservice  repoblservice | 展示层 | 业务逻辑层 |
| userdataservice  repodataservice | 业务逻辑层 | 数据层 |

# 5.接口视角

## 5.1.业务逻辑层的分解

### 5.1.1业务逻辑层模块的职责

|  |  |
| --- | --- |
| 模块 | 职责 |
| userbl | 负责实现查询Github用户信息界面所需要的服务 |
| repobl | 负责实现查询Github项目信息界面所需要的服务 |

### 5.1.2业务逻辑模块的接口模范

**repobl接口规范**

|  |  |  |  |
| --- | --- | --- | --- |
| 提供的服务（供接口） | | | |
| repobl.showRepoByName | 语法 | | public RepoVO showRepoByName(string repoName) |
| 前置条件 | | 系统根据关键字查询列出相关项目列表 |
| 后置条件 | | 根据名字查找具体项目，返回项目详细信息 |
| repobl.searchVague | 语法 | | public List<String>searchVague(String keyword) |
| 前置条件 | | keywords确实存在 |
| 后置条件 | | 返回与keyword相关的项目信息列表，按照默认方式排序显示 |
| repobl.searchPrecise | 语法 | | public RepoVO searchPrecise(String rep, String user) |
| 前置条件 | | 系统处于某一用户详细信息的展示界面 |
| 后置条件 | | 返回该用户的某一具体项目的详细信息并显示 |
| repobl.sort | | 语法 | public List<String> sort(List<String> list) |
| 前置条件 | 系统已经通过模糊查询获得相关的项目信息序列 |
| 后置条件 | 根据排序类型对项目信息序列进行排序，并且返回已经排好序的项目信息序列 |
| 需要的服务（需接口） | | | |
| 服务名 | | | 服务 |
| repodata.searchVagus(String keyword) | | | 根据关键字得到相关项目列表 |
| repodata.searchPrecise(String repoAndUser) | | | 根据项目和用户名字得到项目信息 |

**userbl接口规范**

|  |  |  |  |
| --- | --- | --- | --- |
| 提供的服务（供接口） | | | |
| repobl.showUserByName | 语法 | | public UserVOshowUserByName(string userName) |
| 前置条件 | | 系统根据关键字查询列出相关用户列表 |
| 后置条件 | | 根据名字查找具体项目，返回用户详细信息 |
| repobl.searchVague | 语法 | | public List<String>searchVague(String keyword) |
| 前置条件 | | keywords确实存在 |
| 后置条件 | | 返回与keyword相关的用户信息列表，按照默认方式排序显示 |
| repobl.searchPrecise | 语法 | | public UserVO searchPrecise(String repo, String user) |
| 前置条件 | | 系统处于某一项目详细信息的展示界面 |
| 后置条件 | | 返回该项目的某一合作者的详细信息并显示 |
| repobl.sort | | 语法 | public List<string> sort(List<String> list) |
| 前置条件 | 系统已经通过模糊查询获得相关的用户信息序列 |
| 后置条件 | 根据排序类型对用户信息序列进行排序，并且返回已经排好序的用户信息序列 |
| 需要的服务（需接口） | | | |
| 服务名 | | | 服务 |
| userdata.searchVagus(String keyword) | | | 根据关键字得到相关用户列表 |
| userdata.searchPrecise(String repoAndUser) | | | 根据项目和用户名字得到项目信息 |

## 5.2.数据层的分解

数据层主要给业务逻辑层提供数据访问服务，主要是读取txt格式持久化的项目和用户数据。

### 5.2.1数据层模块的职责

数据层模块职责如下表所示

**数据层模块职责**

|  |  |
| --- | --- |
| 模块 | 职责 |
| userdata | 负责读取用户信息，包括用户基本信息，用户参与项目，用户创建项目。 |
| repodata | 负责读取项目信息，包括git项目的基本信息，使用语言，以及贡献者、合作者、fork的数量。 |

### 5.2数据层模块的接口规范

数据层模块的接口规范如下表

**数据层模块的接口规范**

repodata**接口规范**

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| repodata.searchVague | 语法 | public List<RepoPO> searchVague (String keyword) |
| 前置条件 | keyword不为null |
| 后置条件 | 根据关键字返回相关项目信息的列表，如果不存在则提示之 |
| repodata.searchPrecise | 语法 | public RepoPO searchPrecise(string repo, string user) |
| 前置条件 | Repo, user参数合法 |
| 后置条件 | 返回符合该参数的项目的的详细信息 |

userdata**接口规范**

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| userdata.searchVague | 语法 | public List<UserPO> searchVague (String keyword) |
| 前置条件 | keyword不为null |
| 后置条件 | 根据关键字返回相关用户信息的列表，如果不存在则提示之 |
| repodata.searchPrecise | 语法 | public UserPO searchPrecise(string repo, string user) |
| 前置条件 | Repo, user参数合法 |
| 后置条件 | 返回符合该参数的用户的的详细信息 |

### 