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 |
| 陈之威 | 2016-4-10 | 加入logbl与logdata模块描述 | V1.3 |
| 顾晗 | 2016-4-11 | 加入statisticsbl模块，更新repobl和userbl模块 | V1.4 |
| 顾晗 | 2016-4-16 | 更新开发包图，改错 | V1.4 |
| 顾晗 | 2016-6-7 | 转成web项目，接口发生了更改 | V1.5 |

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**

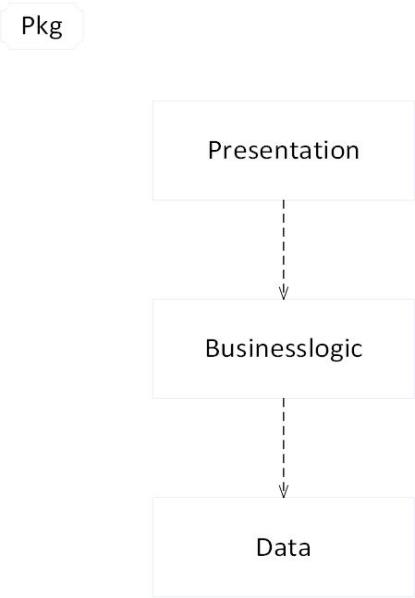
GitMining System数据分析系统中，选择了分层体系结构的风格，将系统分为3层（展示层、业务逻辑层、数据层）能够很好的示意整个高层抽象。展示层包括GUI页面的实现，业务逻辑层包含业务逻辑处理的实现，数据层负责数据的读取。分层体系结构的逻辑视角和逻辑设计方案如图1和图2所示。

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。

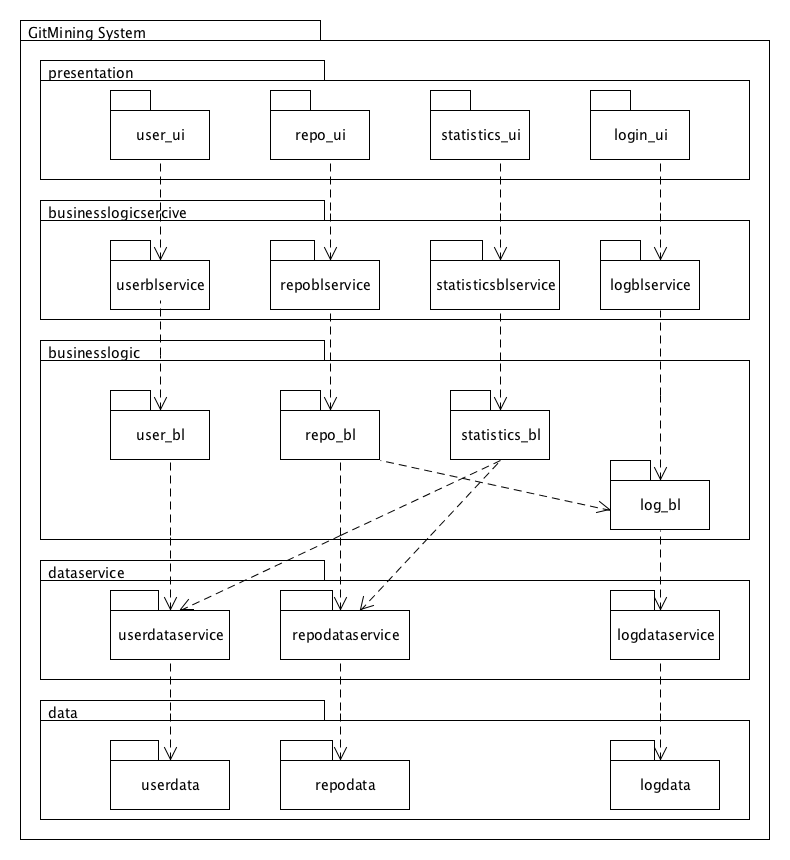


图3:开发包图

**表1层之间调用的接口**

|  |  |  |
| --- | --- | --- |
| 接口 | 服务调用方 | 服务提供方 |
| userblservice  repoblservice  logblservice  statisticblservice | 展示层 | 业务逻辑层 |
| userdataservice  repodataservice  logdataservice | 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 |
| statisticsbl | 负责实现界面展示统计图表所需要的服务 |
| logbl | 负责实现为当前用户提供个性化的服务 |

### 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.searchVague | 语法 | public void searchVague(String keyword) |
| 前置条件 | keywords确实存在 |
| 后置条件 | 在RepoBL中设置searchlist |
| repobl.searchPrecise | 语法 | public RepoVO searchPrecise(String repo, 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<RepoListBean> sort(Comparator<RepoListBean> sortBehavior) |
| 前置条件 | 无 |
| 后置条件 | 返回排好序的项目列表 |
| repobl.queryRecommendLine | 语法 | public String queryRecommendLine(String repo\_name, boolean initial) |
| 前置条件 | 无 |
| 后置条件 | 返回用户对该项目的评分 |
| repobl.startLearning | 语法 | public boolean startLearning() |
| 前置条件 | 用户确认开始学习该项目 |
| 后置条件 | 将数据层该项目状态设置为开始学习 |
| repobl.stopLearning | 语法 | public boolean stopLearning() |
| 前置条件 | 用户确认终止学习该项目 |
| 后置条件 | 将数据层该项目状态设置为终止学习 |
| repobl.finishLearning | 语法 | public boolean finishLearning() |
| 前置条件 | 用户确认已学完该项目 |
| 后置条件 | 将数据层该项目状态设置为已完成学习 |
| 需要的服务（需接口） | | |
| 服务名 | | 服务 |
| repodataservice.queryRecommendLine () | | 返回用户对该项目的评分 |
| repodataservice.searchVague(String keyword) | | 根据关键字得到相关项目列表 |
| repodataservice.searchPrecise(String repo, String User) | | 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） | | |
| userbl.searchVague | 语法 | public ArrayList<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 |
| userbl.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 |
| 需要的服务（需接口） | | |
| 服务名 | | 服务 |
| userdataservice.searchVague(String keyword) | | 根据关键字得到相关用户列表 |
| userdataservice.searchPrecise(String repo, String User) | | 根据项目和用户名字得到项目信息 |

Statisticsbl 接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| userstatisticsbl.getScore | 语法 | public UserAssessBean getScore(String name) |
| 前置条件 | 用户名存在 |
| 后置条件 | 返回该用户的评分数据 |
| userstatisticsbl.getUserActivity | 语法 | public Map<String, Integer> getUserActivity(String name) |
| 前置条件 | 用户名存在 |
| 后置条件 | 根据日期返回每天用户activity次数 |
| userstatisticsbl.getOwnerType | 语法 | public Map<OwnerType, Integer> getOwnerType() |
| 前置条件 | 无 |
| 后置条件 | 根据用户类型返回统计数据 |
| userstatisticsbl.getCreated\_at | 语法 | public HashMap<Integer, Integer> getCreated\_at() |
| 前置条件 | 无 |
| 后置条件 | 返回用户创建年份的统计数据 |
| userstatisticsbl.getRepository | 语法 | public HashMap<Integer, Integer> getRepository(int interval) |
| 前置条件 | interval>0 |
| 后置条件 | 返回以interval为间隔的用户参与项目数的统计数据 |
| repostatisticsbl.getRadar | 语法 | public RepoRadarBean getRadar(String full\_name) |
| 前置条件 | 项目名已存在 |
| 后置条件 | 返回该项目的雷达图数据 |
| repostatisticsbl.getRepoActivity | 语法 | public Map<String, Integer> getRepoActivity(String repo) |
| 前置条件 | 项目名已存在 |
| 后置条件 | 返回该项目按日期的commit统计数据 |
| repostatisticsbl.getLanguage | 语法 | public Map<Language, Integer> getLanguage() |
| 前置条件 | 无 |
| 后置条件 | 返回项目所属语言统计数据 |
| repostatisticsbl.getLanguageByName | 语法 | public Map<Integer, Integer> getLanguage(Language language) |
| 前置条件 | language已存在 |
| 后置条件 | 返回改language按年份增长的项目统计数据 |
| repostatisticsbl.getCreateTime | 语法 | public Map<Integer, Integer> getCreateTime() |
| 前置条件 | 无 |
| 后置条件 | 按创建年份返回项目的统计数据 |
| repostatisticsbl.getRepoFork | 语法 | public Map<Integer, Integer> getRepoFork(int interval) |
| 前置条件 | interval>0 |
| 后置条件 | 返回以interval为间隔的项目fork数的统计数据 |
| repostatisticsbl.getRepoStar | 语法 | public Map<Integer, Integer> getRepoStar(int interval) |
| 前置条件 | interval>0 |
| 后置条件 | 返回以interval为间隔的项目star数的统计数据 |
| 需要的服务（需接口） | | |
| 服务名 | | 服务 |
| userdata.staticsOwnerType | | 获取用户类型的统计数据 |
| userdata.staticsRepo | | 获取用户参与项目的统计数据 |
| userdata.staticsCreated\_at | | 获取用户创建时间的统计数据 |
| userdata.queryEventOfUser | | 获取用户活动的统计数据 |
| userdata.queryRankOfUser | | 获取用户评分数据 |
| repodata.queryCommitOfRepo | | 获取项目commit次数的统计数据 |
| repodata.staticsCreated\_at | | 获取项目创建时间的统计数据 |
| repodata.staticsForks\_Count | | 获取项目fork数的统计数据 |
| repodata.staticsStargazers\_Count | | 获取项目star数的统计数据 |
| repodata.staticsLanguage | | 获取项目language的统计数据 |
| repodata.queryRankOfRepo | | 获取项目评分数据 |

Logbl接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| logbl.login | 语法 | public boolean login(String login, String password) |
| 前置条件 | 系统初始化完成并已连接网络 |
| 后置条件 | 以用户身份访问系统 |
| logbl.logout | 语法 | public void logout() |
| 前置条件 | 系统已经登录 |
| 后置条件 | 退出登陆，保存操作信息 |
| logbl.show | 语法 | public LogBean show() |
| 前置条件 | 用户已登录 |
| 后置条件 | 返回用户详细信息 |
| logbl.deleteHistory | 语法 | public void deleteHistory(String input) |
| 前置条件 | input的记录确实存在 |
| 后置条件 | 从数据库中删除浏览历史记录 |
| logbl.clearHistory | 语法 | public void clearHistory() |
| 前置条件 | 用户已登录 |
| 后置条件 | 清除所有历史浏览记录 |
| logbl.queryHistory | 语法 | public List<HistoryBean> queryHistory() |
| 前置条件 | 用户已登录 |
| 后置条件 | 返回所有历史记录 |
| logbl.queryLearningList | 语法 | public List<HistoryBean> queryLearningList(int complex) |
| 前置条件 | complex=1,2,3 |
| 后置条件 | 返回该难度的学习列表 |
| logbl.queryRecommendRepos | 语法 | public List<RepoListBean> queryRecommendRepos() |
| 前置条件 | 系统已登录 |
| 后置条件 | 根据当前使用者操作记录返回个性化项目推荐列表 |
| 需要的服务（需接口） | | |
| 服务名 | | 服务 |
| logdata.queryRecommendRepos | | 返回该客户的推荐学习的项目 |
| logdata.queryHistory | | 返回浏览记录 |
| logdata.deleteHistory | | 从数据库中删除该条浏览记录 |
| logdata.clearHistory | | 清除所有浏览记录 |
| logdata.show | | 返回用户详细信息所需数据 |
| logdataservice.login(String login, String password) | | 获取用户登录权限 |
| logdataservice.logout() | | 保存操作信息 |

## 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。 |
| logdata | 负责读取使用者操作信息记录，包括用户名，密码，访问次数较多项目。 |

### 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<RepoBean> searchVague(String keyword, String year, Language language,  CategoryOfRepo category) |
| 前置条件 | 无 |
| 后置条件 | 返回项目名列表 |
| repodata.searchPrecise | 语法 | public RepoBean searchPrecise(String fullName) |
| 前置条件 | fullName参数合法 |
| 后置条件 | 返回符合该参数的项目的的详细信息 |
| repodata.queryCommitOfRepo | 语法 | public Map<String, Integer> queryCommitOfRepo(String fullName) |
| 前置条件 | fullName参数合法 |
| 后置条件 | 按日期返回该项目的commit次数 |
| repodata.staticsCreated\_at | 语法 | public Map<Integer, Integer> staticsCreated\_at() |
| 前置条件 | 无 |
| 后置条件 | 按创建日期返回所有项目的个数 |
| repodata.staticsForks\_Count | 语法 | public Map<Integer, Integer> staticsForks\_Count () |
| 前置条件 | 无 |
| 后置条件 | 按fork的数量返回统计的项目数 |
| repodata.staticsStargazers\_Count | 语法 | public Map<Integer, Integer> staticsStargaers\_Count () |
| 前置条件 | 无 |
| 后置条件 | 按stargazers的数量返回统计的项目数 |
| repodata.staticsLanguage | 语法 | public Map<Language, List<Pair<Integer, Integer>>> staticsLanguage() |
| 前置条件 | 无 |
| 后置条件 |  |
| repodata.queryRankOfRepo | 语法 | public RepoRadarBean queryRankOfRepo(String repo\_name) |
| 前置条件 | repo\_name参数合法 |
| 后置条件 | 返回该项目雷达图所需数据 |
| repodata.queryRecommendLine\_Init | 语法 | public String queryRecommendLine\_Init(String repo\_name) |
| 前置条件 | repo\_name参数合法 |
| 后置条件 | 返回用户对该项目的评分初始值字符串 |
| repodata.queryRecommendLine | 语法 | public String queryRecommendLine(String repo\_name) |
| 前置条件 | repo\_name合法 |
| 后置条件 | 返回用户对该项目的评分字符串 |

userdata**接口规范**

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| userdata.queryEventOfUser | 语法 | public Map<String, Integer> queryEventOfUser(String user\_Name) |
| 前置条件 | user\_Name参数合法 |
| 后置条件 | 按日期返回该用户活动的次数 |
| userdata.queryRankOfUser | 语法 | public UserAssessBean queryRankOfUser(String user\_name) |
| 前置条件 | user\_name参数合法 |
| 后置条件 | 返回该用户的评分所需数据 |
| userdata.staticsOwnerType | 语法 | public Map<OwnerType, Integer> staticsOwnerType() |
| 前置条件 | 无 |
| 后置条件 | 按用户类型返回对应的用户数 |
| userdata.staticsCreated\_at | 语法 | public Map<Integer, Integer> staticsCreated\_at() |
| 前置条件 | 无 |
| 后置条件 | 按创建账号的年份返回统计的用户数 |
| userdata.staticsRepo | 语法 | public Map<Integer, Integer> staticsRepo() |
| 前置条件 | 无 |
| 后置条件 | 按用户所参与的项目数返回统计的用户数 |
| userdata.searchVague | 语法 | public ArrayList<UserBean> 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 |
| userdata.searchPrecise | 语法 | public UserBean searchPrecise(String fullName) |
| 前置条件 | fullName参数合法 |
| 后置条件 | 返回符合该参数的用户的的详细信息 |

logdata**接口规范**

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| logdata.startLearning | 语法 | public boolean startLearning(RepoBean repo) |
| 前置条件 | 无 |
| 后置条件 | 将开始学习的项目添加到数据库中 |
| logdata.stopLearning | 语法 | public boolean stopLearning(String repo) |
| 前置条件 | repo参数对应的学习项目已存在 |
| 后置条件 | 将该项目的状态置为abort |
| logdata.finishLearning | 语法 | public boolean finishLearning(String repo) |
| 前置条件 | repo参数对应的学习项目已存在 |
| 后置条件 | 将该项目的状态置为finish |
| logdata.updateHistory | 语法 | public updateHistory(RepoBean repo) |
| 前置条件 | 无 |
| 后置条件 | 将浏览项目添加到数据库中 |
| logdata.queryRecommendRepos | 语法 | public List<RepoBean> queryRecommendList() |
| 前置条件 | 无 |
| 后置条件 | 返回该客户的推荐学习的项目 |
| logdata.queryLearningList | 语法 | public List<HistoryBean> queryLearnigList(int complex) |
| 前置条件 | complex=1,2,3 |
| 后置条件 | 返回对应难度的学习项目列表 |
| logdata.queryHistory | 语法 | public List<HistoryBean> queryHistory() |
| 前置条件 | 无 |
| 后置条件 | 返回浏览记录 |
| logdata.getLearningState | 语法 | public int getLearningState(RepoBean repo) |
| 前置条件 | repo对应的项目已经在学习列表中 |
| 后置条件 | 返回该项目的学习状态 |
| logdata.save | 语法 | public void save() |
| 前置条件 | 用户已登录 |
| 后置条件 | 保存当前用户信息 |
| logdata.deleteHistory | 语法 | public void deleteHistory(String input) |
| 前置条件 | input的项目名必须已经在浏览记录中 |
| 后置条件 | 从数据库中删除该条浏览记录 |
| logdata.clearHistory | 语法 | public void clearHistory() |
| 前置条件 | 无 |
| 后置条件 | 清除所有浏览记录 |
| logdata.queryLogRank | 语法 | public List<LogBean> queryLogRank() |
| 前置条件 | 无 |
| 后置条件 | 返回所有用户的排名 |
| logdata.show | 语法 | public LogBean show() |
| 前置条件 | 无 |
| 后置条件 | 返回当前登录的用户详细信息 |
| logdata.setInfo | 语法 | public void setInfo(int aptitute, Language language) |
| 前置条件 | 无 |
| 后置条件 | 设置系统使用者的编程经验和偏好语言 |
| logdata.sortRepos | 语法 | public List<RepoBean> sortRepos(List<RepoBean> repos) |
| 前置条件 | 无 |
| 后置条件 | 返回一个排好序的项目列表 |
| logdata.login | 语法 | public boolean login(String login, String password) |
| 前置条件 | 系统已连接网络且账号密码输入正确 |
| 后置条件 | 赋予使用者用户操作权限 |
| logdata.logout | 语法 | public void logout() |
| 前置条件 | 使用者已登录 |
| 后置条件 | 系统收回权限 |

### 