

GIT SEARCH

项目架构详细设计文档



南京大学软件学院

吉特 Fan 小组

目录

| | |
|-----------------------------------|---|
| GITSEARCH | 1 |
| 项目架构详细设计文档 | 1 |
| 文档更新记录 | 3 |
| 引言 | 3 |
| 词汇表 | 4 |
| 1. 产品概述 | 4 |
| 2. 系统的分层架构 | 5 |
| 3. 用户界面层的分解 | 7 |
| 4. 系统的架构设计 | 8 |
| 5. 系统的类图设计 | 9 |

| | |
|---------------------------|----|
| 6. 系统顺序图 | 11 |
| 7. 数据层的分解 | 12 |
| 8. 接口规范设计 | 14 |

文档更新记录

| 版本 | 作者 | 描述 | 日期 |
|------|----|----|--------------|
| V1.0 | 万兴 | 初稿 | 2015 - 03-15 |
| V1.1 | 万兴 | 终稿 | 2015-04-12 |

引言

| 内容和说明 | 项目 |
|------------------------------|---|
| 1.编写的目的（说明整个文档所有达到的目标） | 本文档提供 GitSearch 系统的软件架构概览，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.对象与范围（说明整个文档的内容范围和针对的读者对象） | |
| 3.参考文献（说明文档中主要的所需引用） | 1.《软件需求规格说明书》，本团队；《软件架构文档模版》，Rational Software Corporation，2002； 2.骆斌。软件工程与计算（卷三） 3.丁二玉，刘钦。软件工程与计算（卷二）[M] |

| | |
|-----------------------------|-------------------------------------|
| | 机械工业出版社，2012：134—182 |
| 4.名词与术语(说明文档中常用的技术缩略和先相关词条) | GitSearch 表示 gitHub 项目数据统计与分析系统的代名称 |

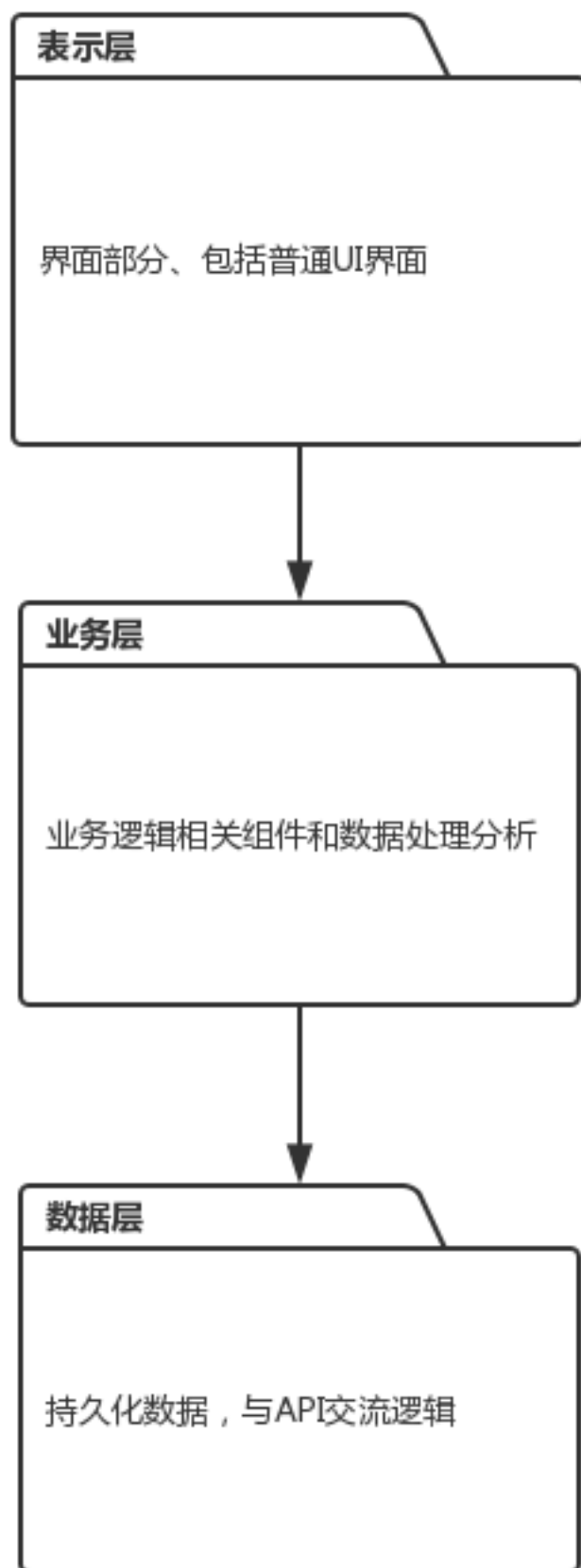
词汇表

| 词汇名称 | 词汇含义 | 备注 |
|-----------|---|----|
| _ui | 表示某展示层 | |
| _bl | 表示某逻辑层 | |
| _data | 表示某数据层 | |
| GitSearch |  itHub 项目数据统计与分析系统 | |

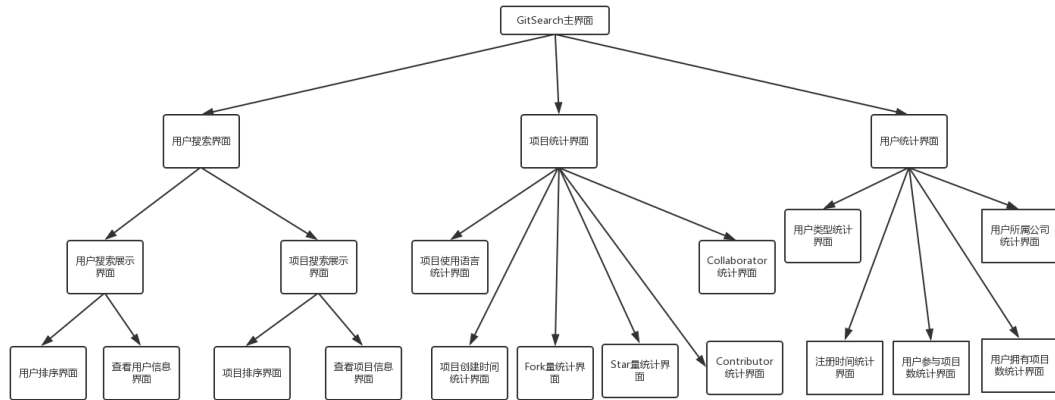
1. 产品概述

GitSearch 旨在通过对 github 网站项目和用户数据的分析展示 ,来帮助广大编程爱好者查找到自己感兴趣的项目 ,了解相关项目的一些基本信息。 主要由 Java 语言开发 ,利用数据库相关功能配合简单和复杂的数理逻辑计算 , 为用户提供一个多角度全方位访问 Github 项目数据信息的窗口。良好的用户交互体验和准确全面的数据分析 ,使用户能找到自己感兴趣的项目。

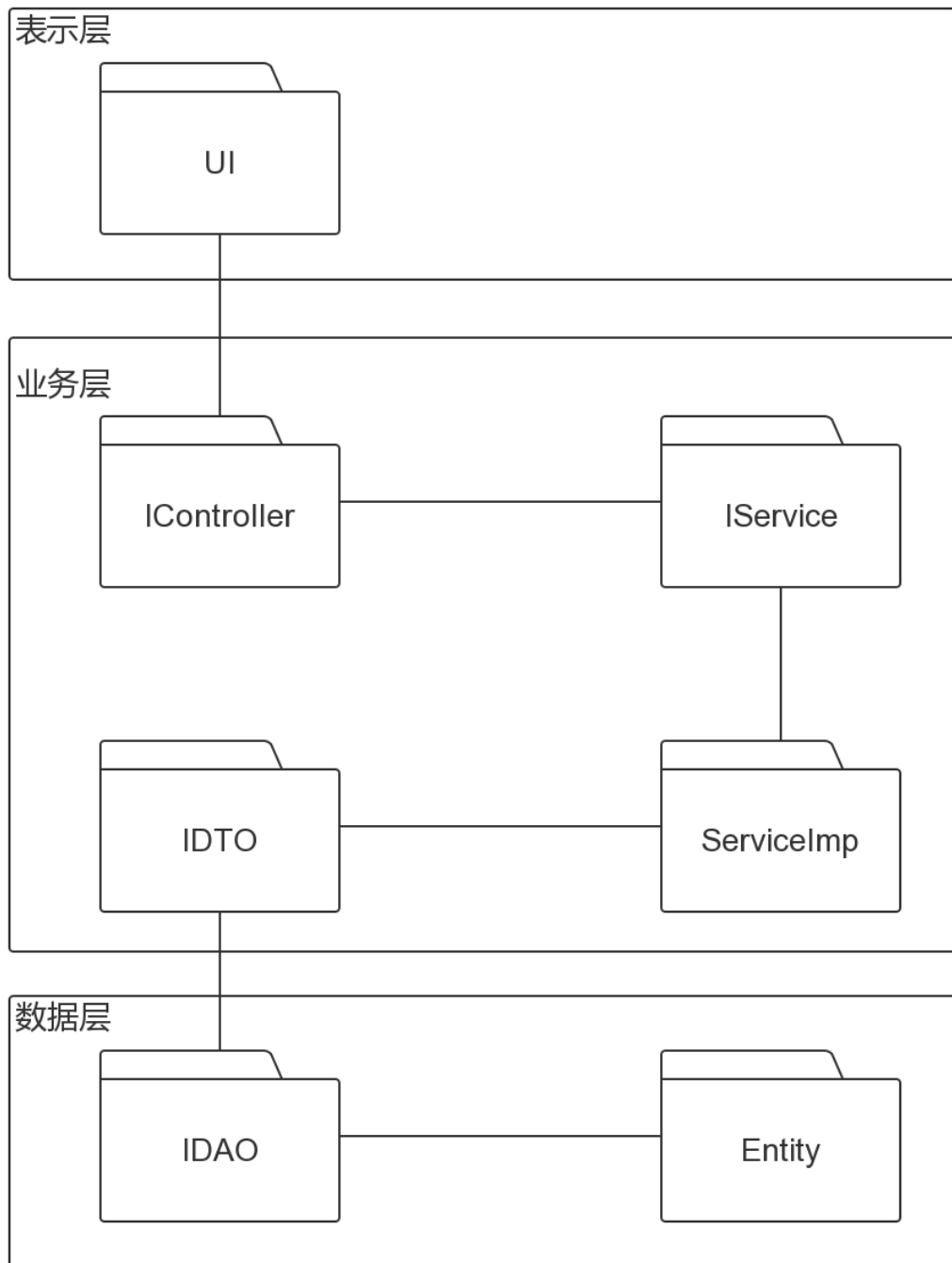
2. 系统的分层架构



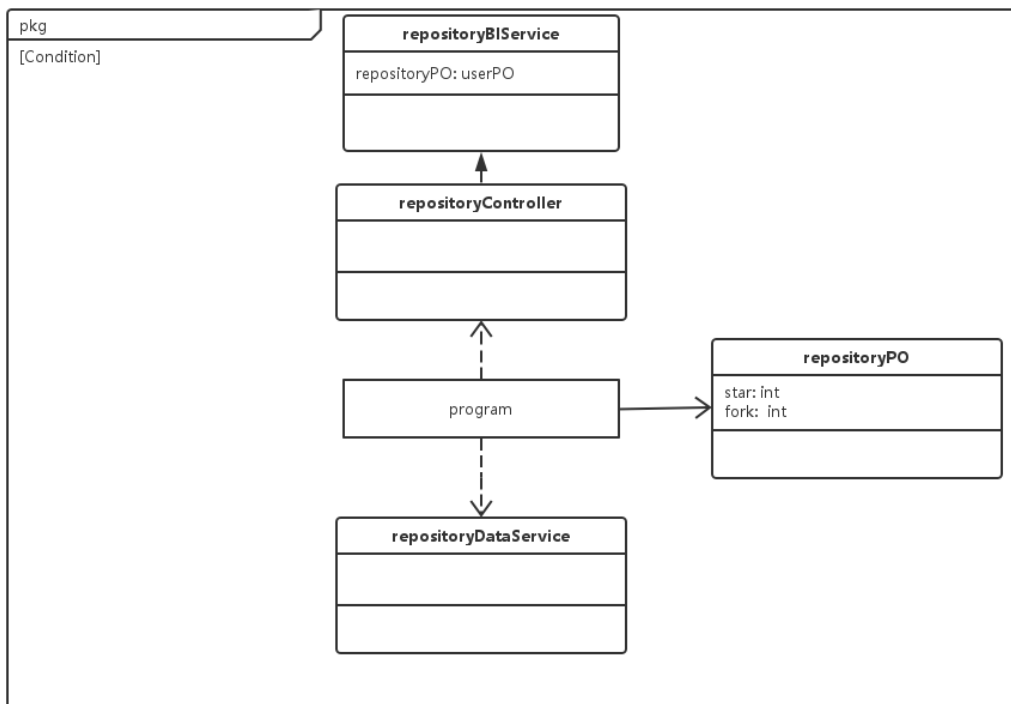
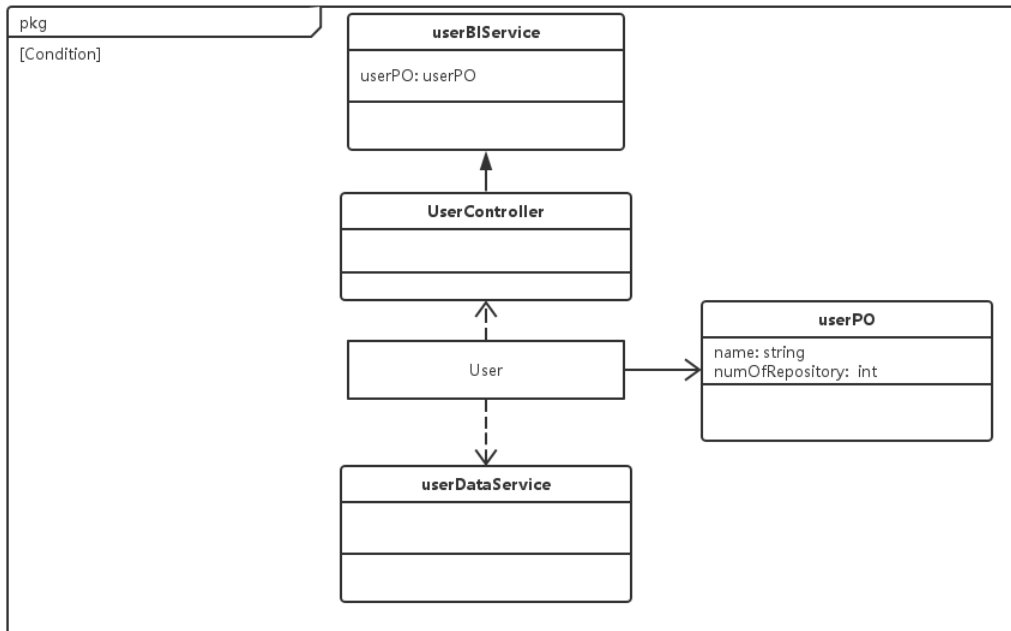
3. 用户界面层的分解

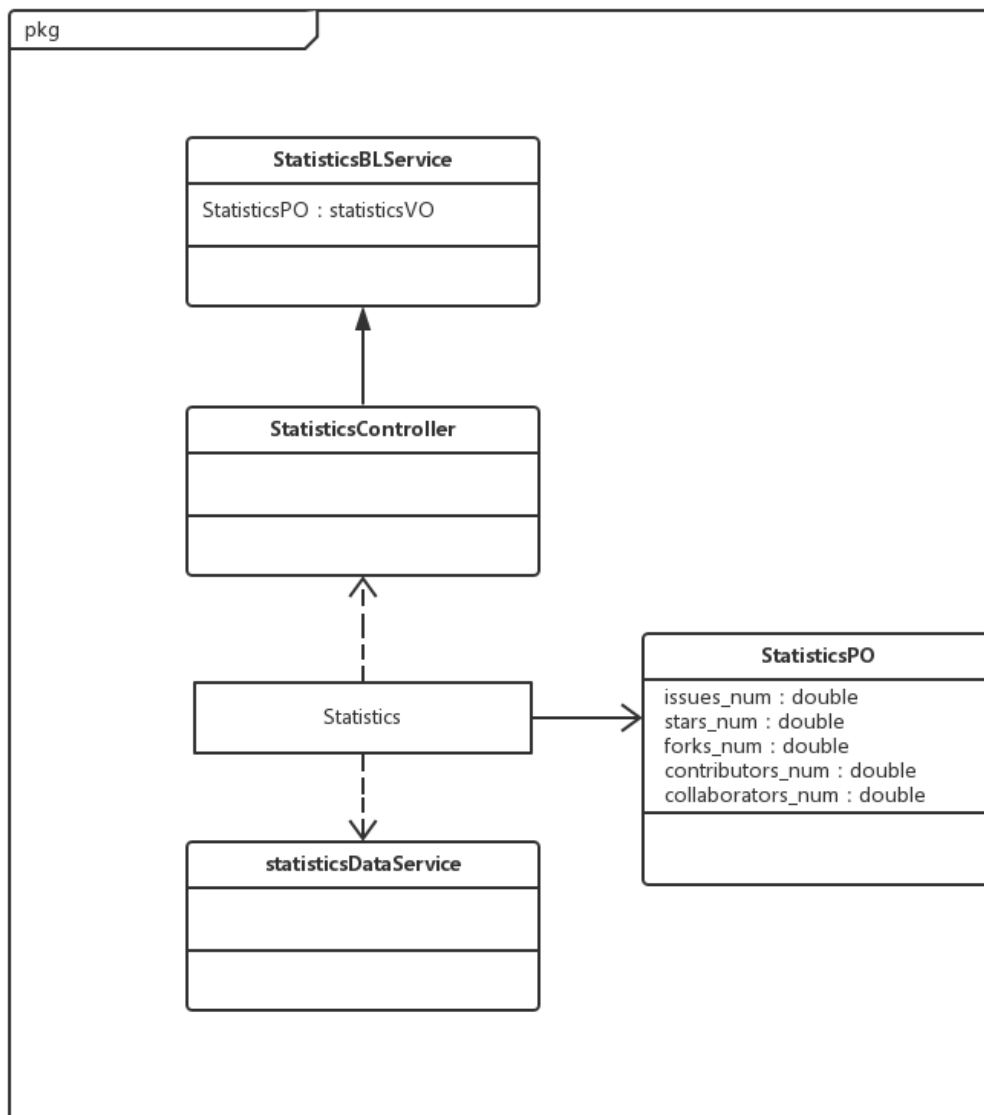


4. 系统的架构设计

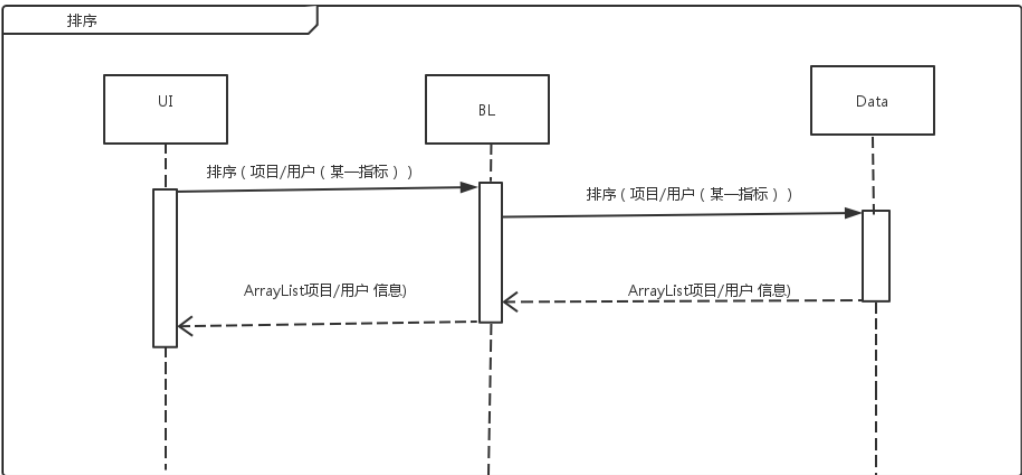
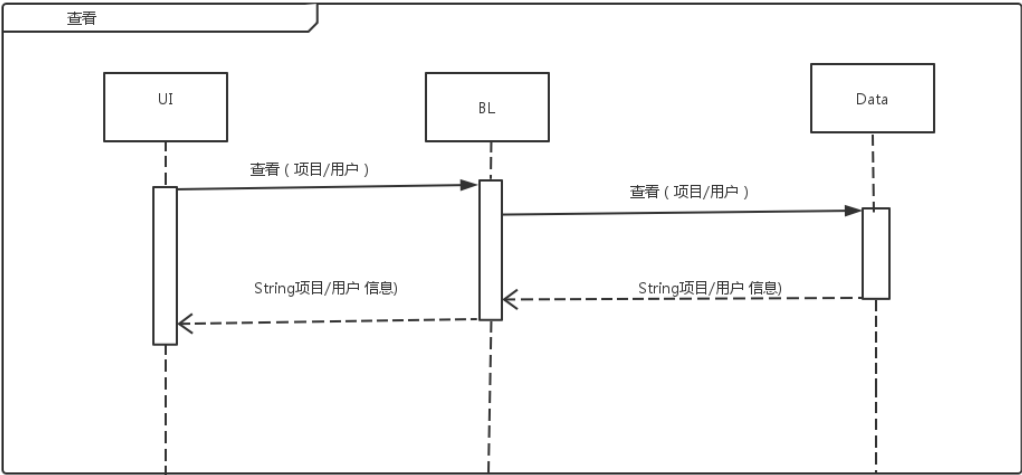


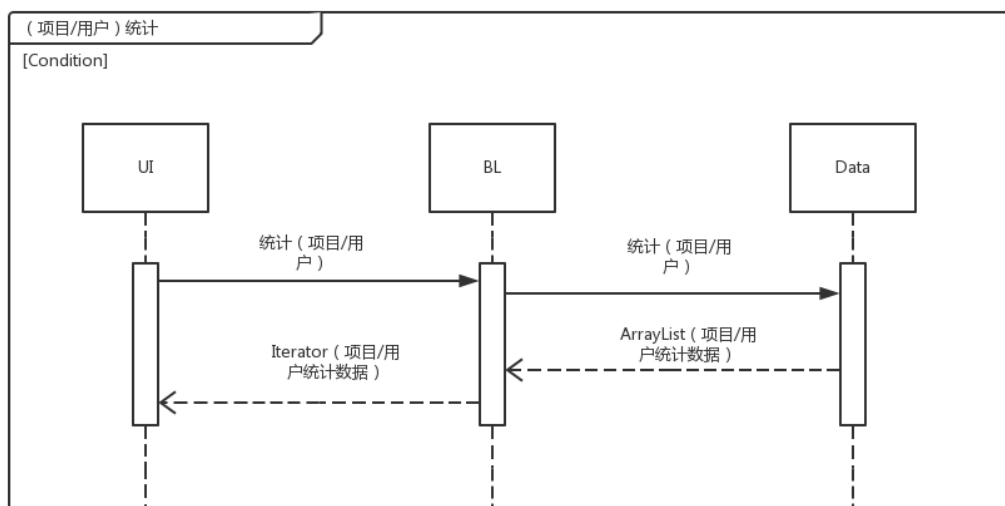
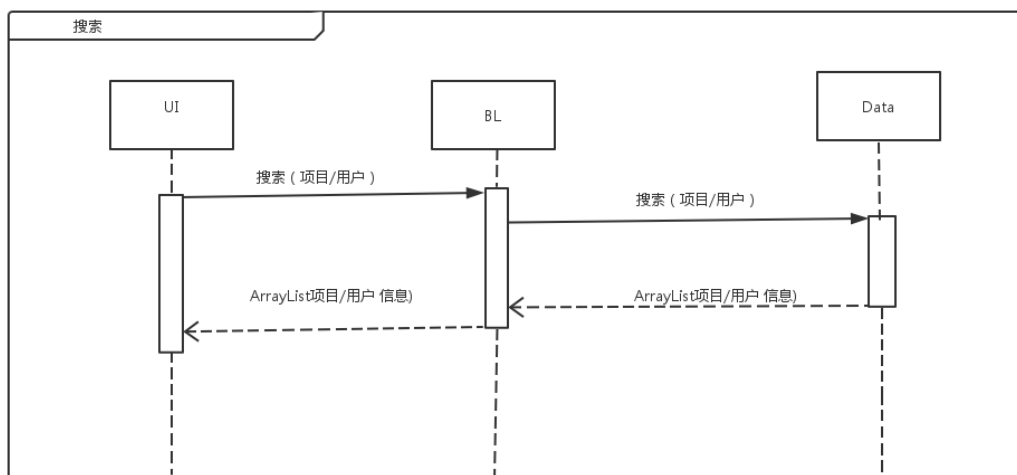
5. 系统的类图设计





6. 系统顺序图





7. 数据层的分解

数据层主要给业务逻辑层提供数据服务，包括对于持久化数据的增、删、改、查。业务逻辑需要的服务 DataService 接口提供。由于持久化数据的保存可能存在多种形式：Txt 文件、序列化文件、数据库等，所示抽象了数据服务。数据层模块的具体描述如图：

接口组件设计结构如下表格：

| 接口 ID | 连接组件 | 接口信息 | |
|-------|---------------------------|------|-------------------------------------|
| I1 | 连接 UI 与 IController | 语法 | Return(Response) Interface(Request) |
| | | 前置条件 | 用户的输入正确 |
| | | 后置条件 | 处理控制组件处理请求并且响应 |
| | | 不变量 | 用户请求信息 |
| I2 | 连接 IContraller 与 IService | 语法 | Return(result) Interface() |
| | | 前置条件 | 无 |
| | | 后置条件 | 对应的 IService 执行对应的业务逻辑 |
| | | 不变量 | 无 |
| I3 | 连接 IService 与 IDTO | 语法 | Return(data Set) Interface(command) |
| | | 前置条件 | 无 |
| | | 后置条件 | 对应的 IDTO 组件调用特定 IDAO 类获 |

| | | | |
|----|------------------|------|--|
| | | | 取数据层数据,并返回数据集 无 |
| | | 不变量 | 无 |
| I4 | 连接 IDAO 与 Entity | 语法 | Return(data) Interface(criteria) |
| | | 前置条件 | 数据库连接正常 |
| | | 后置条件 | IDAO 中的类 将 Entity 对象写入数据库或从数据库中返回 Entity 对象 |
| | | 不变量 | 无 |

8. 接口规范设计

Repository_BL 的接口规范：

| | | |
|-------------------------------|----|---|
| 提供的服务 | | |
| Repository_BL.getRepositories | 语法 | Iterator<RepositoryVO> getRepository (); |

| | | |
|--|------|--|
| | 前置条件 | 已创建一个 Repository_BL 领域对象 |
| | 后置条件 | 返回所有项目详情列表 |
| Repository_BL. getRepositoriesNames | 语法 | Iterator<String> getRepositoriesNames(); |
| | 前置条件 | 已创建一个 Repository_BL 领域对象 |
| | 后置条件 | 返回项目名称列表 |
| Repository_BL. languagesOf Repository | 语法 | Map<String,Integer> languagesOfProgram(String userName, String reponame); |
| | 前置条件 | 已创建一个 Repository_BL 领域对象 |
| | 后置条件 | 返回项目的语言使用情况 |
| Repository_BL. checkRepository | 语法 | RepositoryVO checkRepository (String userName, String reponame); |
| | 前置条件 | 已创建一个 Repository_BL 领域对象 |
| | 后置条件 | 返回单个项目的信息 |
| Repository_BL.screen | 语法 | Iterator<RepositoryVO> screen(ScreenVO vo) ; |
| | 前置条件 | 已创建一个 Repository_BL 领域对象 |
| | 后置条件 | 返回多个项目列表 |

| | | |
|-------------------------------|------|--|
| Repository_BL.Search | 语法 | Iterator<RepositoryVO> Search(String name) |
| | 前置条件 | 已创建一个 Repository_BL 领域对象 |
| | 后置条件 | 返回多个项目列表 |
| Repository_BL.sort | 语法 | Iterator<RepositoryVO> sort |
| | 前置条件 | 已创建一个 Repository_BL 领域对象 |
| | 后置条件 | 返回多个项目列表 |
| Repository_BL.getContributors | 语法 | Iterator<UserVO> getContributors(String userName, String reponame); |
| | 前置条件 | 已创建一个 Repository_BL 领域对象 |
| | 后置条件 | 返回项目所有参与者的信息 |
| Repository_BL.getStar | 语法 | StaStrVO getStar() ; |
| | 前置条件 | 已创建一个 Repository_BL 对象 |
| | 后置条件 | 返回 Star 量信息 |
| Repository_BL.getScores | 语法 | StatisticsVO getScores(RepositoryVO vo) ; |
| | 前置条件 | 已创建一个 Repository_BL 对象 |
| | 后置条件 | 返回 Score 量信息 |

| | | |
|-------------------------------|------|--------------------------------|
| Repository_BL. getForks | 语法 | StaStrVO getForks() ; |
| | 前置条件 | 已创建一个 Repository_BL 对象 |
| | 后置条件 | 返回 Fork 量信息 |
| Repository_BL. getRepoCreated | 语法 | StaStrVO getRepoCreated() ; |
| | 前置条件 | 已创建一个 Repository_BL 对象 |
| | 后置条件 | 返回项目创建信息 |
| Repository_BL. getLanguage | 语法 | StaStrVO getLanguage() ; |
| | 前置条件 | 已创建一个 Repository_BL 对象 |
| | 后置条件 | 返回 Language 信息 |
| Repository_BL. getUserCreated | 语法 | StaStrVO getUserCreated() ; |
| | 前置条件 | 已创建一个 Repository_BL 对象 |
| | 后置条件 | 返回 UserCreated 信息 |
| Repository_BL. getUserType | 语法 | StaStrVO getUserType() ; |
| | 前置条件 | 已创建一个 Repository_BL 对象 |
| | 后置条件 | 返回 UserType 信息 |
| Repository_BL. getUserHas | 语法 | StaIntVO getUserHas() ; |

| | | |
|----------------------------------|------|----------------------------------|
| | 前置条件 | 已创建一个 Repository_BL 对象 |
| | 后置条件 | 返回拥有项目数信息 |
| Repository_BL. getUserRelated | 语法 | StaIntVO getUserRelated() ; |
| | 前置条件 | 已创建一个 Repository_BL 对象 |
| | 后置条件 | 返回参与项目数信息 |
| Repository_BL. getCompany | 语法 | StaStrVO getCompany() ; |
| | 前置条件 | 已创建一个 Repository_BL 对象 |
| | 后置条件 | 返回 Company 信息 |
| Repository_BL. getHasNum | 语法 | Integer getHasNum(String name) ; |
| | 前置条件 | 已创建一个 Repository_BL 对象 |
| | 后置条件 | 返回创建项目数信息 |
| Repository_BL. getCollaborator | 语法 | StaStrVO getCollaborator() ; |
| | 前置条件 | 已创建一个 Repository_BL 对象 |
| | 后置条件 | 返回 Collaborator 信息 |
| Repository_BL. getContributor | 语法 | StaStrVO getContributor(); |
| | 前置条件 | 已创建一个 Repository_BL |

| | | |
|---------------------------------------|------|---|
| | | 对象 |
| | 后置条件 | 返回 Contributor 信息 |
| Repository_BL. NamesOfContributors | 语法 | <pre> Iterator<String> NamesOfContributors(String userName, String reponame); </pre> |
| | 前置条件 | 已创建一个 Repository_BL 领域对象 |
| | 后置条件 | 返回所有项目参与者的用户 名 |
| Repository_BL. getBranches | 语法 | <pre> Iterator<ProgramVO> getBranches(String userName, String reponame); </pre> |
| | 前置条件 | 已创建一个 Repository_BL 领域对象 |
| | 后置条件 | 返回项目的所有版本 |
| Repository_BL. NamesOfBranches | 语法 | <pre> Iterator<String> NamesOfBranches(String userName, String reponame); </pre> |
| | 前置条件 | 已创建一个 Repository_BL 领域对象 |
| | 后置条件 | 返回项目的所有版本的名字 |
| Repository_BL. checkBranch | 语法 | <pre> ProgramVO checkBranch(String userName, String reponame); </pre> |
| | 前置条件 | 已创建一个 Repository_BL |

| | | |
|-----------------------------|------|---|
| | | 领域对象 |
| | 后置条件 | 查看某一个版本的信息 |
| Repository_BL. BranchInfo | 语法 | String BranchInfo(String userName, String reponame, BranchInfo info); |
| | 前置条件 | 已创建一个 Repository_BL 领域对象 |
| | 后置条件 | 返回某个版本的某项信息 |
| Repository_BL. getForks | 语法 | Iterator<ProgramVO> getForks(String userName, String reponame); |
| | 前置条件 | 已创建一个 Repository_BL 领域对象 |
| | 后置条件 | 返回一个项目的所有 Fork 项 目 |
| Repository_BL. NamesOfForks | 语法 | Iterator<String> NamesOfForks(String userName, String reponame); |
| | 前置条件 | 已创建一个 Repository_BL 领域对象 |
| | 后置条件 | 返回单个项目的所有 fork 项 目的全称 |
| Repository_BL. checkFork | 语法 | ProgramVO checkFork(String userName, String reponame); |
| | 前置条件 | 已创建一个 Repository_BL 领域对象 |

| | | |
|-------------------------|------|--|
| | 后置条件 | 返回某个 fork 项目的信息 |
| Repository_BL. ForkInfo | 语法 | <code>String ForkInfo(String userName, String reponame, ForkInfo info);</code> |
| | 前置条件 | 已创建一个 Repository_BL 领域对象 |
| | 后置条件 | 返回某个 fork 项目的某项信息 |
| Repository_BL. getPulls | 语法 | <code>Iterator<String> getPulls(String userName, String reponame);</code> |
| | 前置条件 | 已创建一个 Repository_BL 领域对象 |
| | 后置条件 | 返回单个项目的 Pull Request 内容 (前 50 个) |
| Repository_BL. PullInfo | 语法 | <code>String PullInfo(String userName, String reponame, String num, PullInfo info);</code> |
| | 前置条件 | 已创建一个 Repository_BL 领域对象 |
| | 后置条件 | 返回某个 Pull Request 的某项信息 |
| Repository_BL. getPull | 语法 | <code>String getPull(String userName, String reponame, int number s);</code> |
| | 前置条件 | 已创建一个 Repository_BL |

| | | |
|---------------------------|------|---|
| | | 领域对象 |
| | 后置条件 | 返回某个 Pull Request 的某项信息 |
| Repository_BL.getPullInfo | 语法 | String getPullInfo(String userName, String reponame, int number, Item item); |
| | 前置条件 | 已创建一个 Repository_BL 领域对象 |
| | 后置条件 | 返回单个项目的 Issues 内容,一页 50 个 |
| Repository_BL.getIssues | 语法 | Iterator<String> getIssues(String userName, String reponame); |
| | 前置条件 | 已创建一个 Repository_BL 领域对象 |
| | 后置条件 | 返回单个项目的 Issue 编号, 一页 50 个 |
| Repository_BL.getIssueNum | 语法 | Iterator<Integer> getIssueNum(String userName, String reponame); |
| | 前置条件 | 已创建一个 Repository_BL 领域对象 |

| | | |
|-----------------------------|------|---|
| | 后置条件 | 返回单个项目的某个 issue |
| Repository_BL. getIssue | 语法 | String getIssue(String userName, String reponame,int issue Num); |
| | 前置条件 | 已创建一个 Repository_BL 领域对象 |
| | 后置条件 | 返回某个 Issue 的某项信息 |
| Repository_BL. getIssueInfo | 语法 | String getIssueInfo(String userName, String reponame,int num,IssueInfo info); |
| | 前置条件 | 已创建一个 Repository_BL 领域对象 |
| | 后置条件 | 返回单个项目的 Commits 内容,一页 50 个 |
| Repository_BL. getCommits | 语法 | Iterator<String> getCommits(String userName, String reponame); |
| | 前置条件 | 已创建一个 Repository_BL 领域对象 |
| | 后置条件 | 返回单个项目的 ommit 编号,一页 50 个 |

| | | |
|-----------------------------|------|---|
| Repository_BL. getForks | 语法 | Iterator<ProgramVO> getForks(String userName, String reponame); |
| | 前置条件 | 已创建一个 Repository_BL 领域对象 |
| | 后置条件 | 返回一个项目的所有 Fork 项 目 |
| Repository_BL. NamesOfForks | 语法 | Iterator<String> NamesOfForks(String userName, String reponame); |
| | 前置条件 | 已创建一个 Repository_BL 领域对象 |
| | 后置条件 | 返回单个项目的所有 fork 项 目的全称 |
| Repository_BL. checkFork | 语法 | ProgramVO checkFork(String userName, String reponame); |
| | 前置条件 | 已创建一个 Repository_BL 领域对象 |
| | 后置条件 | 返回某个 fork 项目的信息 |
| Repository_BL. ForkInfo | 语法 | String ForkInfo(String userName, String reponame, ForkInfo info); |
| | 前置条件 | 已创建一个 Repository_BL 领域对象 |
| | 后置条件 | 返回某个 fork 项目的某项信 息 |

| | | |
|----------------------------|------|--|
| Repository_BL. getPulls | 语法 | Iterator<String> getPulls(String userName, String reponame); |
| | 前置条件 | 已创建一个 Repository_BL 领域对象 |
| | 后置条件 | 返回单个项目的 Pull Request 内容 (前 50 个) |
| Repository_BL. PullInfo | 语法 | String PullInfo(String userName, String reponame, String num, PullInfo info); |
| | 前置条件 | 已创建一个 Repository_BL 领域对象 |
| | 后置条件 | 返回某个 Pull Request 的某 项信息 |
| Repository_BL. Getpull | 语法 | String getPull(Strin g userName, String r eponame,int number s); |
| | 前置条件 | 已创建一个 Repository_BL 领域对象 |
| | 后置条件 | 返回某个 Pull Request 的 某项信息 |
| Repository_BL. getPullInfo | 语法 | String getPullInfo (String userName, St ring reponame,int nu mber,Item item); |
| | 前置条件 | 已创建一个 Repository_BL |

| | | |
|----------------------------|------|---|
| | | 领域对象 |
| | 后置条件 | 返回单个项目的 Issues 内容,一页 50 个 |
| Repository_BL. getIssues | 语法 | <pre>Iterator<String> getIssues(String userName, String repositoryName);</pre> |
| | 前置条件 | 已创建一个 Repository_BL 领域对象 |
| | 后置条件 | 返回单个项目的 Issue 编号, 一页 50 个 |
| Repository_BL. getIssueNum | 语法 | <pre>Iterator<Integer> getIssueNum(String userName, String repositoryName);</pre> |
| | 前置条件 | 已创建一个 Repository_BL 领域对象 |
| | 后置条件 | 返回单个项目的某个 issue |
| Repository_BL. getIssue | 语法 | <pre>String getIssue(String userName, String repositoryName, int issueNum);</pre> |
| | 前置条件 | 已创建一个 Repository_BL 领域对象 |
| | 后置条件 | 返回某个 Issue 的某项信息 |

| | | |
|----------------------------|------|---|
| | | |
| Repository_BL.getIssueInfo | 语法 | String getIssueInfo(String userName, String reponame,int num,IssueInfo info); |
| | 前置条件 | 已创建一个 Repository_BL 领域对象 |
| | 后置条件 | 返回单个项目的 Commits 内容,一页 50 个 |
| Repository_BL.getCommits | 语法 | Iterator<String> getCommits(String userName, String reponame); |
| | 前置条件 | 已创建一个 Repository_BL 领域对象 |
| | 后置条件 | 返回单个项目的 ommit 编号,一页 50 个 |
| Repository_BL.getCommitNum | 语法 | Iterator<Integer> getCommitNum(String userName, String reponame); |
| | 前置条件 | 已创建一个 Repository_BL 领域对象 |
| | 后置条件 | 返回单个项目的某个 commit |
| Repository_BL.getCommit | 语法 | String getCommit(String userName, String reponame,int sha); |

| | | |
|--------------------------------|------|---|
| | 前置条件 | 已创建一个 Repository_BL 领域对象 |
| | 后置条件 | 返回某个 commit 的某项信息 |
| Repository_BL.getCommitInfo | 语法 | String getCommitInfo(String userName, String reponame,int sha,CommitInfo info); |
| | 前置条件 | 已创建一个 Repository_BL 领域对象 |
| | 后置条件 | 返回单个项目的某个 issue 的评论 |
| Repository_BL.getIssueComment | 语法 | String getIssueComment(String userName, String reponame,int num); |
| | 前置条件 | 已创建一个 Repository_BL 领域对象 |
| | 后置条件 | 返回某个 commit 的评论 |
| Repository_BL.getCommitComment | 语法 | String getCommitComment(String userName, String reponame,int num) |
| | 前置条件 | 已创建一个 Repository_BL 领域对象 |
| | 后置条件 | 返回某个 commit 的评论 |

User_BL 的接口规范：

| | | |
|-------------------|------|--|
| User_BL.checkUser | 语法 | UserVO checkUser(String name); |
| | 前置条件 | 已创建一个 User_BL. 领域对象 |
| | 后置条件 | 返回某个 user 所有信息 |
| User_BL.UserInfo | 语法 | String UserInfo(String user, UserInfo info); |

| | | |
|-------------------------------|------|---|
| | 前置条件 | 已创建一个 User_BL. 领域对象 |
| | 后置条件 | 返回 user 某个信息 |
| User_BL. getStargazers | 语法 | Iterator<String> getStargazers(String userName, String reponame); |
| | 前置条件 | 已创建一个 User_BL. 领域对象 |
| | 后置条件 | 返回单个项目的点赞者, 一页 50 个 |
| User_BL. getStargazerNames | 语法 | Iterator<String> getStargazerNames(String userName, String reponame; |
| | 前置条件 | 已创建一个 User_BL. 领域对象 |
| | 后置条件 | 返回单个项目的点赞者 登录名, 一页 50 个 |
| User_BL. getSubscribers | 语法 | Iterator<UserVO> getSubscribers(String userName, String reponame); |
| | 前置条件 | 已创建一个 User_BL. 领域对象 |
| | 后置条件 | |
| User_BL. NamesOfSubscriber | 语法 | Iterator<String> NamesOfSubscriber(String userName, String reponame); |
| | 前置条件 | 已创建一个 User_BL. 领域对象 |
| | 后置条件 | 返回单个项目的订阅者, 一页 50 个 |
| User_BL. sortUser | 语法 | Iterator<UserVO> sortUser(User_Sort sort); |
| | 前置条件 | 已创建一个 User_BL. 领域对象 |
| | 后置条件 | 返回符合某项信息的用户, 一页 50 个 |
| User_BL. screenTime | 语法 | Iterator<UserVO> screenTime(String time) |

| | | |
|--|------|---------------------|
| | 前置条件 | 已创建一个 User_BL. 领域对象 |
| | 后置条件 | 返回符合某项信息的用户，一页 50 个 |