

迭代三需求规格说明文档

Gitming Project



2016-4-22

NANJING UNIVERSITY

|  |  |  |  |
| --- | --- | --- | --- |
| 文档类别 | 需求规格说明文档 | 文档编号 | I2\_REQUIRMENTS\_1.0 |
| 创建人 | 崔浩、袁阳阳 | 创建时间 | 2016-02-28 |
| 最后一次修改人 | 崔浩 | 最后一次修改时间 | 2016-04-22 |

目录

[1. 引言 2](#_Toc449108746)

[1.1目的 2](#_Toc449108747)

[1.2范围 3](#_Toc449108748)

[1.3参考文献 3](#_Toc449108749)

[2. 总体描述 3](#_Toc449108750)

[2.1商品前景 3](#_Toc449108751)

[2.1.1背景与机遇 3](#_Toc449108752)

[2.1.2业务需求 4](#_Toc449108753)

[2.2商品功能 4](#_Toc449108754)

[2.3用户特征 4](#_Toc449108755)

[2.4约束 4](#_Toc449108756)

[2.5假设和依赖 5](#_Toc449108757)

[3. 详细需求描述 5](#_Toc449108758)

[3.1对外接口需求 5](#_Toc449108759)

[3.1.1用户界面 5](#_Toc449108760)

[3.1.2通信接口 5](#_Toc449108761)

[3.2功能需求 6](#_Toc449108762)

[3.2.1 GitHub用户信息查询 6](#_Toc449108763)

[3.2.2 GitHub仓库信息查询 8](#_Toc449108764)

[3.2.3 GitHub用户信息统计 10](#_Toc449108765)

[3.2.4 Github仓库信息统计 11](#_Toc449108766)

[3.2.5 用户注册和登陆 12](#_Toc449108767)

[3.2.6 查看推荐项目 14](#_Toc449108768)

[3.2.7 查看关注项目 14](#_Toc449108769)

[3.3非功能需求 15](#_Toc449108770)

[3.3.1 可维护性 15](#_Toc449108771)

[3.3.2 可靠性 15](#_Toc449108772)

[3.4数据需求 15](#_Toc449108773)

[3.4.1 数据定义 15](#_Toc449108774)

[3.4.2 默认数据 16](#_Toc449108775)

## 1. 引言

### 1.1目的

本文档描述了Github数据分析系统Gitming的功能需求和非功能需求。开发小组的软件系统的实现和验证工作都以此文档为依据。

### 1.2范围

Github数据分析系统Gitming是为分析Github项目与用户信息开发的业务系统，开发的目标是帮助编程爱好者找到自己感兴趣的项目并了解该项目的一些信息，包括项目、用户信息的查询。

### 1.3参考文献

1）IEEE标准。

2）Github数据分析系统Gitming用例文档V1.0。

## 2. 总体描述

### 2.1商品前景

#### 2.1.1背景与机遇

GitHub系统是全球最大的社交编程及代码托管网站，作为开源代码库以及版本控制系统，拥有140多万开发者用户，且随着应用程序向云上的转移，GitHub已经成为了管理软件开发以及发现已有代码的首选方式。但是随着GitHub的普及，项目、仓库、用户的增多，在GitHub上找到编程爱好者感兴趣且适合的项目变得越发困难。希望能够更好更快地为开发者和编程爱好者找到合适的项目、GitHub用户，并查看项目和用户的相关信息。

GitHub数据分析系统Gitming就是为了满足GitHub网站这一需要而开发的，它通过客户端从网页获取API数据，并进行显示、分析，用户通过客户端查询需要的项目或GitHub用户信息。

#### 2.1.2业务需求

BR1：在系统开放1个月后，用户查找项目或GitHub用户的速度提升20%.

### 2.2商品功能

SF1：快速显示用户需要的项目与用户信息。

SF2：显示github用户和项目统计信息

SF3：根据用户的个人信息推荐可能感兴趣的项目和人

### 2.3用户特征

本系统由开发者或编程爱好者使用，他们在GitHub网站上查找感兴趣或者自己擅长的项目以及GitHub用户的信息，由于GitHub内容的多样性或对自己的不够了解，他们难以找到适合的项目及信息，希望系统可以帮助他们解决这些问题。

### 2.4约束

CON1：项目整体采用Java（java8）语言开发。

CON2：系统使用的是PC端的图形界面。

CON3：迭代一迭代二不使用数据库。

CON4：工程行为记录在GitLab上。

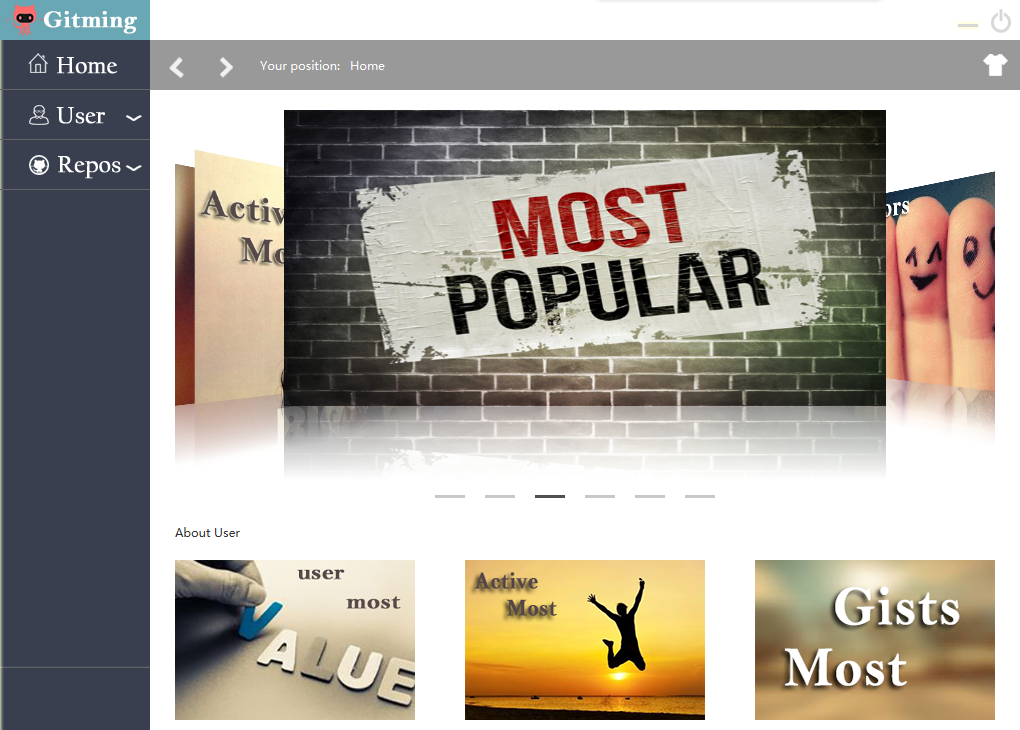
### 2.5假设和依赖

<http://www.gitmining.net/api>不会崩溃（迭代二中在数据初始化后这一依赖已经无关紧要。）

## 3. 详细需求描述

### 3.1对外接口需求

#### 3.1.1用户界面



#### 3.1.2通信接口

### 3.2功能需求

#### 3.2.1 GitHub用户信息查询

3.2.1.1特性描述

在用户使用系统时，输入关键字或者根据标签查找GitHub用户信息，完成对符合要求的用户信息的显示。

3.2.1.2刺激/响应序列

刺激：用户选择查看GitHub用户信息

响应：系统显示单页GitHub用户信息列表，列表中包括用户简略信息

刺激：用户输入关键字并确认检索GitHub用户

响应：系统显示符合关键字要求的单页GitHub用户信息列表，列表中包括用户简略信息

刺激：用户选择查看下一页（上一页）

响应：系统切换到相应页码，（如果是本来在最后一页或者第一页则不切页）

刺激：用户在github用户信息列表中选择具体的GitHub用户查看其具体信息

响应：系统显示该用户具体信息。包括展现用户能力的雷达图

刺激：用户在github用户信息列表中选择同页内零至多位用户进行对比

响应：系统显示被选择用户各个参数信息对比的表格和对比的雷达图（如果是零位显示为空，如果只是一个用户显示一个用户细信息）

3.2.1.3相关功能需求

|  |  |
| --- | --- |
| Gitusers.input  Gitusers.Input.Search | 系统要允许用户在查询时使用鼠标和键盘输入 |
| 在用户输入关键字并检索后，系统要显示符合关键字要求的GitHub用户列表，无符合要求项时显示为空 |
| Gitusers.Show | 用户选择查看GitHub用户时，系统显示所有GitHub用户信息列表 |
| Gitusers.Show.detail | 用户选择某一具体GitHub用户时，系统显示该GitHub用户具体信息，包括login name，company，blog，location，email，bio， public\_repos，public\_gists，followers，following，created\_at，updated\_at |
| Gitusers.Show.detail.Project | 用户选择查看某一GitHub用户的相关项目，系统显示该GitHub用户相关联的项目列表。 |
| Gitusers.compare | 系统允许用户选择两个或多个github用户进行对比 |
| Gitusers.compare.show | 用户选择两个github用户进行对比后，系统显示两位用户参数的对比表格和对比的能力雷达图 |
| Gitusers.follow | 用户可以选择关注该Github用户 |

#### 3.2.2 GitHub仓库信息查询

3.2.2.1 特征描述

用户可以查看项目列表，也可以选择某个具体的Github仓库查看它的详细信息。

优先级=高

3.2.2.2 刺激/响应序列

刺激：用户选择查看仓库信息

响应：系统提供单页仓库信息列表

刺激：用户输入想查看的仓库名（不需输入仓库所有者名字）

响应：系统显示符合搜索条件的仓库信息列表

刺激：用户选择查看下一页（上一页）

响应：系统切换到相应页码，（如果是本来在最后一页或者第一页则不切页）

刺激：用户在仓库信息列表中选择具体的仓库

响应：系统显示该仓库的具体信息，包括仓库各个属性的雷达图

刺激：用户选择同一页仓库信息列表内的零至多个仓库进行对比

响应：系统显示被选中仓库参数属性对比的表格和对比的雷达图（如果是零个仓库，显示空，如果一个仓库，显示一个仓库的信息）

3.2.2.3 相关功能需求

|  |  |
| --- | --- |
| Repository. input | 系统要允许用户在查询时使用键盘输入 |
| Repository. search | 系统要支持用户按仓库名称检索 |
| Repository. show | 系统要能展示仓库的基本信息，包括：项目名称、项目描述、最后更新时间、stars、forks、subscribers数量。 |
| Repository. show. basic | 系统要能展示单个项目的基本信息，包括：项目语言、所有者、项目主页、项目描述、创建时间、最后更新时间、stars、forks、contributors、subscribers、collaborators数量、项目大小，并提供查询其他详细信息的入口 |
| Repository. show. owner | 系统要能显示单个项目所有者的id、登陆名 |
| Repository. show. Relativeuser | 系统要能显示单个项目的相关用户信息 |
| Repository. show. activity | 系统要能显示单个项目自创建到最后一次更新以来的活跃度随时间变化关系 |
| Repository.compare | 系统允许用户在仓库信息列表选择两个仓库进行对比 |
| Repository.compare.show | 用户选中两个仓库进行对比后，系统显示两个仓库参数属性对比的表格和对比的雷达图。 |
| Repository.follow | 用户可以选择关注该仓库 |
| Repository.commit.month | 系统要能展示该项目的commit数量关于一个月的统计，即在一个月的每一天的commit数量 |
| Repository.commit.week | 系统要能展示该项目的commit数量关于一周的统计，即在一周的每一天的commit数量 |
| Repository.commit.day | 系统要能展示该项目的commit数量关于一天的统计，即在一天的每个小时的commit数量 |
| Repository.commit.relation | 系统要能展示该项目所有collaborator的关联度（共同修改的文件数量） |
| Repository.commit.contribution | 系统要能展示该项目的所有collaborator对该项目的贡献度（提交数量、新增代码行数和删除代码行数的统计） |

#### 3.2.3 GitHub用户信息统计

3.2.3.1 特征描述

用户使用系统时，可以查看github用户的相关统计信息

优先级 = 高

3.2.3.2 刺激/响应序列

刺激：用户选择查看github用户统计信息

响应：系统显示可供选择查看的用户信息统计项，具体项参见用例文档——用户信息统计

刺激：用户选择具体要查看的统计项

响应：系统显示该项统计信息

3.2.3.3相关功能需求

|  |  |
| --- | --- |
| UserStatistic.input | 系统允许用户在查看时使用鼠标点选 |
| UserStatistic.showKinds | 用户选择查看用户统计信息后，系统显示可供选择查看的统计项，具体项参见UserStatistic.showSpecific |
| UserStatistic.showSpecific | 用户选择具体查看的统计项后，系统显示相应具体项的用户统计信息 |
| UserStatistic.showSpecific.orgUser | 系统显示真实用户和组织的统计 |
| UserStatistic.showSpecific.comUserNum | 系统显示部分公司的github用户数量统计 |
| UserStatistic.showSpecific.email | 系统显示用户使用的电子邮件统计 |
| UserStatistic.showSpecific.createTime | 系统显示用户创建时间统计 |
| UserStatistic.showSpecific.repoNum | 系统显示用户仓库数量统计 |
| UserStatistic.showSpecific.gistNum | 系统显示用户gist数量统计 |
| UserStatistic.showSpecific.followersNum | 系统显示用户follower数量统计 |

#### 3.2.4 Github仓库信息统计

3.2.4.1 特征描述

用户使用系统时，可以查看github仓库的相关统计信息

3.2.4.2 刺激/响应序列

刺激：用户选择查看github仓库统计信息

响应：系统显示可供选择查看的仓库信息统计项，具体项参见用例文档——仓库信息统计

刺激：用户选择具体要查看的统计项

响应：系统显示该项统计信息

3.2.4.3 相关功能需求

|  |  |
| --- | --- |
| RepoStatistic.input | 系统允许用户使用鼠标点选 |
| RepoStatistic.showKinds | 用户选择查看仓库统计信息后，系统显示可供选择查看的统计项，具体项参见RepoStatistic.showSpecific |
| RepoStatistic.showSpecific | 用户选择具体查看的统计项后，系统显示相应具体项的仓库统计信息 |
| RepoStatistic.showSpecific.language | 系统显示项目使用的语言统计 |
| RepoStatistic.showSpecific.createTime | 系统显示项目创建时间的分布 |
| RepoStatistic.showSpecific.contributorNum | 系统显示项目的贡献者人数统计 |
| RepoStatistic.showSpecific.collaboratorNum | 系统显示项目的合作者人数统计 |
| RepoStatistic.showSpecific.subscriberNum | 系统显示项目的订阅者人数统计 |
| RepoStatistic.showSpecific.size | 系统显示项目大小 |
| RepoStatistic.showSpecific.forkNum | 系统显示项目的被fork次数 |
| RepoStatistic.showSpecific.starNum | 系统显示项目的被点赞数 |

#### 3.2.5 用户注册和登陆

3.2.5.1 特征描述

用户使用系统时，可以注册和登陆

3.2.5.2 刺激/响应序列

刺激：用户选择注册

响应：系统要求用户填写个人信息，包括用户名、密码、密码确认、选择关注领域标签等

刺激：用户填写完成用户信息并确认

响应：系统检查用户信息的可靠性，若可靠则同意注册

刺激：用户输入用户名和密码要求登陆

响应：系统检查用户名和密码，若匹配则同意登陆。

3.2.5.3 相关功能需求

|  |  |
| --- | --- |
| Login.regist | 系统要能允许用户注册 |
| Login.regist.input | 用户在可以用键盘输入注册信息 |
| Login.regist.validate | 系统可以验证用户输入的准确性，用户名只可以由是字母数字和下划线组成，密码必须为8-16位，包括数字、字母和特殊字符 |
| Login.login | 系统要能允许用户登陆 |
| Login.login.regist | 用户登陆时若发现未注册则要求用户注册 |
| Login.validate | 系统要能验证用户的用户名和密码信息 |
| Login.refuse | 用户名不存在或密码错误时系统要能拒绝用户的登陆请求 |
|  |  |

#### 3.2.6 查看推荐项目

3.2.6.1 特征描述

用户使用系统时，可以查看系统认为该用户可能感兴趣的项目或人

3.2.6.2 刺激/响应序列

刺激：用户选择查看推荐项目

响应：系统展示该用户可能感兴趣的项目和人

3.2.6.3 相关功能需求

|  |  |
| --- | --- |
| Recommend.show | 系统要能展示推荐信息 |
| Recommend.show.repository | 系统要能展示推荐的项目信息 |
| Recommend.show.user | 系统要能展示推荐的人的信息 |

#### 3.2.7 查看关注项目

3.2.7.1 特征描述

用户使用系统时，可以查看该用户关注的项目和人

3.2.7.2 刺激/响应序列

刺激：用户选择查看关注项目

响应：系统展示该用户关注的项目和人

3.2.7.3 相关功能需求

|  |  |
| --- | --- |
| Follow.show | 系统要能展示用户关注的项目和人 |
| Follow.show.user | 系统要能展示用户关注的人的信息 |
| Follow.show.repository | 系统要能展示用户关注的项目的信息 |
| Follow.login | 若用户未登陆，系统会要求登陆 |

### 3.3非功能需求

#### 3.3.1 可维护性

1. 如果系统的交互语言（中文、英语）要发生变化，系统要在3人日内完成修改。
2. 如果系统获取的数据格式发生变化，系统要在10人日内完成修改。

#### 3.3.2 可靠性

1、如果网络连接中断，系统不能崩溃。

2、如果网络连接中断，系统要给出提示信息。

### 3.4数据需求

#### 3.4.1 数据定义

**DR1：GIT用户（User）（owner、contributor、collaborator）**

包括：id，login，type，name，company，blog，location，email，bio， followers，following，创建时间，最后更新时间，以及相关联的[项目/仓库(repository)](#_DR2：项目/仓库（repository）)

**DR2：项目/仓库（repository）**

包括：项目语言（language）、[所有者(owner)](#_DR3：所有者（owner）)、项目主页（url）、项目描述、创建时间、最后更新时间、stars、forks、contributors数量、项目大小

以及：[contributor](#_DR1:GIT用户（User）)信息、[collaborator](#_DR1:GIT用户（User）)信息、[版本](#_DR4：版本（branch）)信息、fork[项目](#_DR2：项目/仓库（repository）)信息、[commit](#_DR4：提交（commit）)信息、[事件（issue）](#_DR5：事件（issue）)信息

**DR3：所有者（owner）**

所有者有该[用户](#_DR1:GIT用户（User）（owner、contributor、c)的信息，以及所有者类型（owner\_type：（分为User和Organization））.这是user的一个枚举属性。

**DR4：提交（commit）**

包括日期与所属仓库，以键值对的方式存储。（repositoryname->List<created\_at>）

**DR5：事件（issue）**

包括所属仓库与创建时间（created\_at），以键值对的方式存储。（repositoryname->List<created\_at>）

#### 3.4.2 默认数据

1、初始项目列表按被关注（star）度排序。

2、初始用户列表按被关注（followed）度排序。