需求规格说明书

PMH - Project Dawn



Revision Edition

arvinsc@foxmail.com;511679327@qq.com

Dawn-Team



目 录

[1 引言 1](#_Toc494316500)

[1.1 编写目的 1](#_Toc494316501)

[1.2 项目背景 1](#_Toc494316502)

[1.2.1 项目的委托单位、开发单位和主管部门 1](#_Toc494316503)

[1.2.2 该软件系统与其他系统的关系 1](#_Toc494316504)

[1.3 定义 1](#_Toc494316505)

[1.4 参考资料 1](#_Toc494316506)

[2 任务概述 3](#_Toc494316507)

[2.1 目标 3](#_Toc494316508)

[2.2 运行环境 3](#_Toc494316509)

[2.3 条件与限制 3](#_Toc494316510)

[3 数据描述 4](#_Toc494316511)

[3.1 静态数据 4](#_Toc494316512)

[3.2 动态数据 4](#_Toc494316513)

[3.2.1 用户信息部分 4](#_Toc494316514)

[3.3 数据库介绍 4](#_Toc494316515)

[3.3.1 MySQL Server 5](#_Toc494316516)

[3.3.2 MongoDB 5](#_Toc494316517)

[3.3.3 Redis 5](#_Toc494316518)

[3.4 数据词典 6](#_Toc494316519)

[3.5 数据采集 8](#_Toc494316520)

[4 需求描述 8](#_Toc494316521)

[4.1 功能需求 9](#_Toc494316522)

[4.2 性能需求 9](#_Toc494316523)

[4.2.1 数据精确度 9](#_Toc494316524)

[4.2.2 时间特性 9](#_Toc494316525)

[4.2.3 适应性 9](#_Toc494316526)

[4.3 运行需求 9](#_Toc494316527)

[4.3.1 用户界面 9](#_Toc494316528)

[4.3.2 软件接口 9](#_Toc494316529)

[4.3.3 故障处理 9](#_Toc494316530)

[4.4 其它需求 9](#_Toc494316531)

# 引言

## 编写目的

为明确软件的需求、向最终能用户阐述系统功能、合理安排项目规划和工作进度、有效地组织软件开发和测试工作，项目组成员撰写本文档。

在概要设计和详细设计中，分析和设计人员可以查看并根据本文档而实施其工作；在软件验收测试的过程中，本文档作为其参照的主要标准；在整个开发流程中，项目管理人员可通过本文档对整个项目组的人员、项目的进度、项目的质量进行系统化的控制。

## 项目背景

### 项目的委托单位、开发单位和主管部门

本项目为项目组的开源项目，其所有开发工作均没有委托成分。即，开发单位为Dawn-Team的所有成员，其管理工作由Dawn-Team的责任人负责。

### 该软件系统与其他系统的关系

本软件系统需要RDBMS和NoSQL数据库系统配合使用（参见《可行性分析报告》），故需要依赖于这两个外部系统。在Web实现的过程中，下述软件包库会作为实现的基础：Spring Framework，Bootstrap, Apache POI。除上述提到的外部系统外，在实际实现的过程中，可能会根据实际运行情况选择Deeplearning4J软件包作为软法支持。

## 定义

|  |  |
| --- | --- |
| **定义项** | **释义** |
| Dawn-Team | Dawn项目组成员，详细清单参见[《可行性研究报告》](#_参考资料)1.3节 |
| PMH | PancakeofMountHuang，整个项目的数据预处理部分，附加通用模块构建部分 |

## 参考资料

|  |  |  |  |
| --- | --- | --- | --- |
| **序号** | **名称** | **类型** | **备注** |
| 1 | 《可行性分析报告》 | 项目文档 | 隶属于Dawn整个项目的可行性分析报告，包括技术、人力和经济等方面。文件位置：[Github@dawn/master](https://github.com/Dawn-Team/dawn/blob/master/Doc/1.Feasibility%20Study%20Report.docx) |
| 2 | runoob.com | 网站 | <http://www.runoob.com> |
| 3 | 百度百科MongoDB | Wiki网站 | <https://baike.baidu.com/item/mongodb/60411> |
|  |  |  |  |

# 任务概述

## 目标

在限定的时间内完成对PMH的分析、设计、实现和测试，保证其功能基本实现，在能够进行数据预处理功能的基础之上，基本使用时没有运行异常，能够处理协调好多用户同时使用的场景，适当处理好并发能力，适当保证用户使用系统的安全能力。

其他关于整个项目的目标描述，参见[《可行性研究报告》](#_参考资料)第二节。

## 运行环境

|  |  |
| --- | --- |
| **项目** | **内容** |
| OS | Ubuntu 16.04 |
| Server & Container | Nginx 1.3  Spring Boot 1.5.7  Tomcat 8.5.20 |
| Language | Java SDK version 8 |
| Dependencies | Spring IO platform  Spring Boot  Spring Data  Spring Security  Spring Session |
| Database systems | MySQL Server 5.7  MongoDB 3.4.7  Redis |

## 条件与限制

1. 对于用户用于进行数据预处理的数据，需要满足一定规格的数据格式，并且，在用户上传文件之前就进行应该提供相关的提示、说明和下载。
2. 对于用户用于进行数据预处理的数据，应当遵守相关法律法规规定且系统不保证其数据安全，系统应当在用户使用系统前进行有效的提示和说明。
3. 其他未尽条件与限制内容，参见[《可行性研究报告》](#_参考资料)第二节。

# 数据描述

## 静态数据

|  |  |  |  |
| --- | --- | --- | --- |
| **序号** | **名称** | **类型** | **说明** |
| 1 | 本地化/国际化文件 | 可选 | 用于本地化支持的文件 |
| 2 | 用户使用平台和平台部分功能的使用条款、免责声明 | 可选 | 用于处理法律法规规定和责任澄清 |
|  |  |  |  |

## 动态数据

### 用户信息部分

|  |  |  |  |
| --- | --- | --- | --- |
| **序号** | **名称** | **类型** | **说明** |
| 1 | 用户信息 | 用户使用系统的必要信息 | 用户注册时输入，长期存储于数据库系统中 |
| 2 | 数据预处理数据源 | 数据预处理的目标 | 用户提供数据源（如：Excel文件），数据预处理后的数据存放于数据库系统之中以供用户下一步使用 |
| 3 |  |  |  |
| 4 |  |  |  |
| 5 |  |  |  |

## 数据库介绍

### MySQL Server

MySQL是一个关系型数据库管理系统，由瑞典MySQL AB公司开发，目前属于Oracle公司。MySQL是一种关联数据库管理系统，关联数据库将数据保存在不同的表中，而不是将所有数据放在一个大仓库内，这样就增加了速度并提高了灵活性。其特性如下：

1. MySQL是开源的，使用时不需要支付额外的费用。
2. MySQL使用标准的SQL数据语言形式。
3. MySQL可以允许于多个系统上，并且支持多种语言。这些编程语言包括C、C++、Python、Java、Perl、PHP、Eiffel、Ruby和Tcl等。
4. MySQL支持大型数据库，支持5000万条记录的数据仓库，32位系统表文件最大可支持4GB，64位系统支持最大的表文件为8TB。
5. MySQL是可以定制的，采用了GPL协议，你可以修改源码来开发自己的MySQL系统。

### MongoDB

MongoDB是一个基于分布式文件存储的数据库。由C++语言编写。旨在为WEB应用提供可扩展的高性能数据存储解决方案。

MongoDB是一个介于关系数据库和非关系数据库之间的产品，是非关系数据库当中功能最丰富，最像关系数据库的。他支持的数据结构非常松散，是类似JSON的BSON格式，因此可以存储比较复杂的数据类型。Mongo最大的特点是他支持的查询语言非常强大，其语法有点类似于面向对象的查询语言，几乎可以实现类似关系数据库单表查询的绝大部分功能，而且还支持对数据建立索引。

在本项目中，MongoDB主要提供与“数据”相关的数据的存储服务，例如对数据预处理结果的存储。

### Redis

Redis 是完全开源免费的，遵守BSD协议，是一个高性能的key-value数据库（属于NoSQL数据库）。Redis 与其他 key - value 缓存产品有以下三个特点：

1. Redis支持数据的持久化，可以将内存中的数据保存在磁盘中，重启的时候可以再次加载进行使用。
2. Redis不仅仅支持简单的key-value类型的数据，同时还提供list，set，zset，hash等数据结构的存储。
3. Redis支持数据的备份，即master-slave模式的数据备份。

在本项目中，Redis主要与Spring Session配合，实现集群场景中的Session共享功能。

## 数据词典

由于NoSQL不需要进行表结构的设计，仅是在实际业务中与部分涉及到MySQL的数据相关联，故如下的数据字典都针对于MySQL数据库。注：以下数据字典中，所有UUID字段都为32字节恒长。

|  |  |  |
| --- | --- | --- |
| **Schema Name: Users** | | |
| **Type** | **Details** | **Description** |
| Username | String | Unique, LENGTH=[6, 32] |
| Password | BASE64(SHA256(plaintext\_password+’{’+Username\_SALT+’}’)) | User-end LENGTH=[6,16] Storage-end LENGTH<=32 |
| Enabled | TRUE || FALSE | Account status |

|  |  |  |
| --- | --- | --- |
| **Schema Name: Authorities** | | |
| **Type** | **Details** | **Description** |
| Username | String | References to Schema Users |
| USER\_ROLE | ROLE\_ADMIN || ROLE\_COMMON || ROLE\_SPECIAL || Anonymous(Default) | System or Admin assigned |

|  |  |  |
| --- | --- | --- |
| **Schema Name: Users\_Profiles** | | |
| **Type** | **Details** | **Description** |
| Username | String | References to Schema Users |
| Nickname | Username(Default) | Display name of user, LENGTH=[0,64] |
| Tel | (‘+’+area code)+[0-9]{7,11} | Telephone number, LENGTH=[7,15] |
| Email | Standard email address | Email address, LENGTH=[5,32] |
| Profile\_Picture\_URL | UUID | References to Schema File\_Upload |
| Enroll\_Date | DATETIME | Date and time of enrollment |

|  |  |  |
| --- | --- | --- |
| **Schema Name: File\_Upload** | | |
| **Type** | **Details** | **Description** |
| File\_UUID | UUID | UUID for the file |
| Upload\_datetiem | DATETIME | Date and time of upload operation |
| Original\_filename | String | Original filename |
| Purpose | String | Upload purpose |
| Status | “NORMAL” || ”UPLOADING” || ”EXPIRED” | File status |
| Username | String | References to Schema Users |

|  |  |  |
| --- | --- | --- |
| **Schema Name: File\_Download** | | |
| **Type** | **Details** | **Description** |
| File\_UUID | UUID | UUID for the file to download |
| File\_Name | String | Displayed filename for download, LENGTH<=128 |
| Expire\_Date | DATETIME | Expire Date and time |
| Username | String | Reference to Schema Users |
| Public | TRUE||FLASE | Is the download public |

|  |  |  |
| --- | --- | --- |
| **Schema Name: Job\_Schedules** | | |
| **Type** | **Details** | **Description** |
| Job\_UUID | UUID | Job UUID |
| Priority | 0-9 | 0 for the superior |
| Prerequisite | UUID | Prerequisite job |
| Expected\_Resources | Positive Integer | Expected resources to run the job |
| Status | “WAITING” || ”RUNNING” || “READY” || “SUSPENDING” || “DEAD” || “ACOMPLISHED” | Job status info. |
| Information | String | Job Information, LENGTH<=128 |
| Exception\_Info | String | Exception Information,  LENGTH<=128 |
| Username | String | Reference to Schema Users |

|  |  |  |
| --- | --- | --- |
| **Schema Name: Datasource** | | |
| **Type** | **Details** | **Description** |
| DS\_UUID | UUID | UUID for DS |
| Type | “EXCEL” || “RDBMS” || “NoSQL” || “JSON” | DS Type |
| Stage | “Preprocess” || “Analysis” || “Visualization” | Stage DS used for |
| Location | String | DS Location, for EXCEL the disk url. LENGTH<=256 |
| Username | String | Reference to Schema Users |
| Status | “EXPIRED” || “INVALID” || “NORMAL” || “TIMEOUT” || “NOT\_EXIST” | DS Status |

|  |  |  |
| --- | --- | --- |
| **Schema Name: Data\_Preprocess\_records** | | |
| **Type** | **Details** | **Description** |
| Process\_UUID | UUID | UUID for the process |
| Username | String | Reference to Schema Users |
| Datasource\_UUID | UUID | Reference to Schema Datasource |
| Progress | Num[0,100] | Progress |
| Status | String | A specific step in the data preprocess procedure |

//TODO TEMPLET, DELETE BEFORE PUBLISH

|  |  |  |
| --- | --- | --- |
| **Schema Name: //TODO** | | |
| **Type** | **Details** | **Description** |
|  |  |  |
|  |  |  |

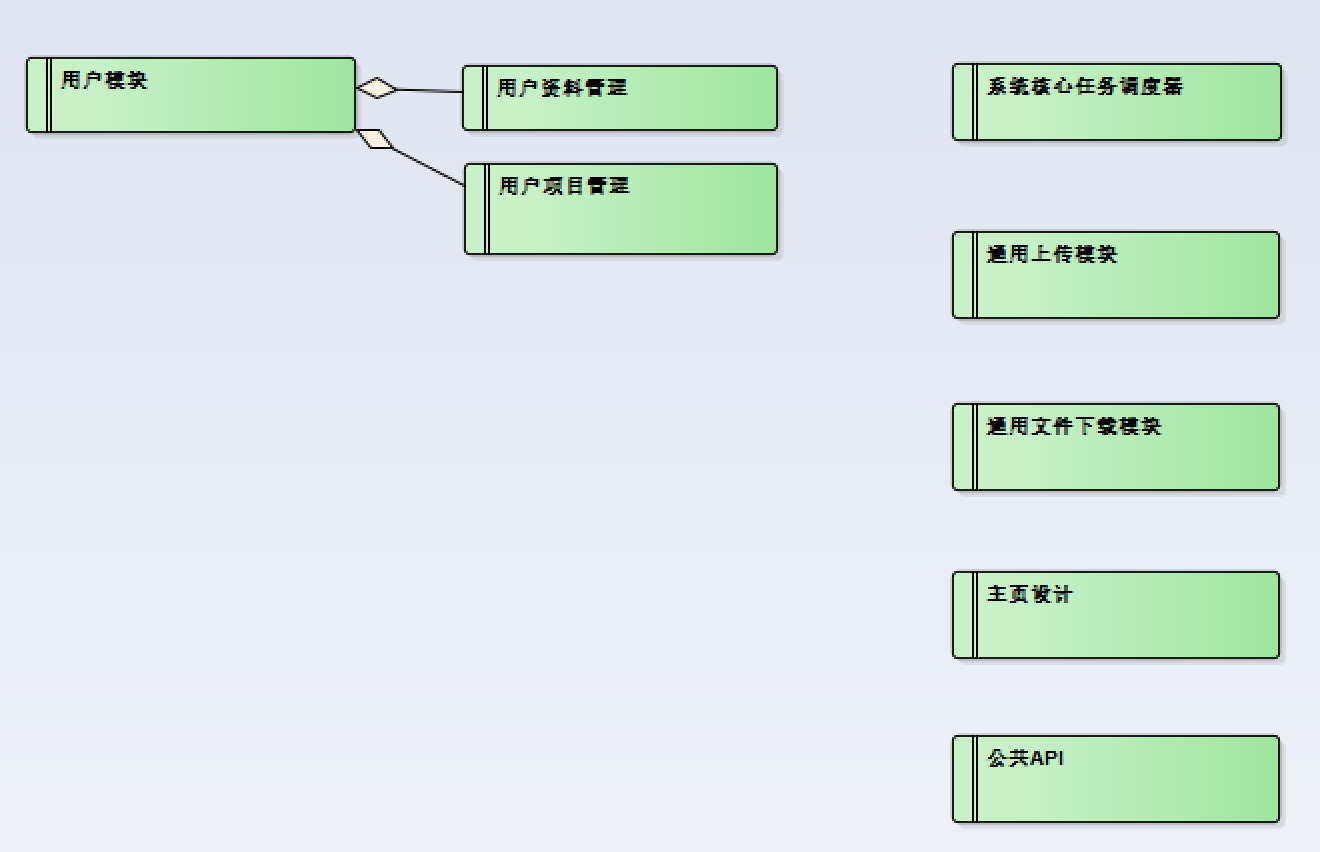
# 需求描述

## 功能需求

根据[《可行性研究报告》](#_参考资料)，当前阶段分析、设计和实现的内容包括完全包含于PMH的数据预处理内容和属于公共性的通用模块。在PMH中，功能性仅包括一个数据预处理的主需求，而这个需求又有细化和延伸出一些自需求，他们的关系参见图1。同时，对于通用模块中的功能性需求，在本阶段的工作中主要负责图2中所包含的需求。对于上述需求，他们的详细规格清单都位于本文档[需求清单](#_需求清单)一节中。



1. 主要功能性需求以及其延伸



1. 通用性功能

## 需求清单

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **REQ-ID: REQ001 - Data Pre-process Module** | | | | | |
| **Alias** | 数据预处理模块 | **Difficulty** | High | **Priority** | High |
| **Status** | Approved | **Stability** | Medium | **Type** | Functional |
| **Description** | The main requirement of PMH | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **REQ-ID: REQ002 - Users Module** | | | | | |
| **Alias** | 用户模块 | **Difficulty** | Medium | **Priority** | High |
| **Status** | Approved | **Stability** | High | **Type** | Functional |
| **Description** | Responsible for users’ register, login and logout. | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **REQ-ID: REQ003 - System Core Job Scheduler** | | | | | |
| **Alias** | 系统核心任务调度器 | **Difficulty** | High | **Priority** | High |
| **Status** | Approved | **Stability** | Medium | **Type** | Functional |
| **Description** | Responsible for resources schedule such as CPU and memory. | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **REQ-ID: REQ004 - Common Files Upload Module** | | | | | |
| **Alias** | 通用上传模块 | **Difficulty** | Low | **Priority** | Medium |
| **Status** | Approved | **Stability** | High | **Type** | Functional |
| **Description** | Responsible for file upload, should be a single sub system. | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **REQ-ID: REQ005 - Common File Download Module** | | | | | |
| **Alias** | 通用文件下载模块 | **Difficulty** | Low | **Priority** | Low |
| **Status** | Approved | **Stability** | High | **Type** | Functional |
| **Description** | Related to the Upload file, this should be also a sub system. | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **REQ-ID: REQ006 - Main UI Design** | | | | | |
| **Alias** | 主页设计 | **Difficulty** | Medium | **Priority** | High |
| **Status** | Approved | **Stability** | High | **Type** | Functional |
| **Description** | The main User Interface. | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **REQ-ID: REQ007 - Titles Extraction** | | | | | |
| **Alias** | 表头抽取 | **Difficulty** | Medium | **Priority** | High |
| **Status** | Proposed | **Stability** | High | **Type** | Functional |
| **Description** | Extracting the Excel file headers. | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **REQ-ID: REQ008 - Column Selection for Coming Computation** | | | | | |
| **Alias** | 选择待计算列 | **Difficulty** | Medium | **Priority** | High |
| **Status** | Proposed | **Stability** | High | **Type** | Functional |
| **Description** | Choose the headers to compute later. | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **REQ-ID: REQ009 - Data Cleaning** | | | | | |
| **Alias** | 数据清洗 | **Difficulty** | High | **Priority** | High |
| **Status** | Proposed | **Stability** | Medium | **Type** | Functional |
| **Description** | Math procedure of calculating and updating the abnormal data. | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **REQ-ID: REQ010 - Data Pre-Computation for Coming Analysis** | | | | | |
| **Alias** | 数据预运算 | **Difficulty** | Low | **Priority** | High |
| **Status** | Proposed | **Stability** | High | **Type** | Functional |
| **Description** | Math procedure of added computation. | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **REQ-ID: REQ011 - Data Storage** | | | | | |
| **Alias** | 数据存储 | **Difficulty** | Medium | **Priority** | High |
| **Status** | Proposed | **Stability** | High | **Type** | Functional |
| **Description** | Storage for the processed data. | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **REQ-ID: REQ012 - Users' Profiles Management** | | | | | |
| **Alias** | 用户资料管理 | **Difficulty** | Low | **Priority** | Low |
| **Status** | Proposed | **Stability** | High | **Type** | Functional |
| **Description** | Include Passwords reset and some profile updates. | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **REQ-ID: REQ013 - Users' Projects Management** | | | | | |
| **Alias** | 用户项目管理 | **Difficulty** | Medium | **Priority** | High |
| **Status** | Proposed | **Stability** | Medium | **Type** | Functional |
| **Description** | Users project management. | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **REQ-ID: REQ014 – Data Sources Interface** | | | | | |
| **Alias** | 数据源接口 | **Difficulty** | Medium | **Priority** | High |
| **Status** | Proposed | **Stability** | Medium | **Type** | Functional |
| **Description** | The interface for reading the data to be processed. | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **REQ-ID: REQ015 – User Oriented Security** | | | | | |
| **Alias** | 面向用户的安全 | **Difficulty** | Medium | **Priority** | High |
| **Status** | Proposed | **Stability** | High | **Type** | Non-Functional |
| **Description** | Information Protection. | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **REQ-ID: REQ016 – Data Persistence** | | | | | |
| **Alias** | 数据持久化 | **Difficulty** | Medium | **Priority** | High |
| **Status** | Proposed | **Stability** | High | **Type** | Non-Functional |
| **Description** | Data Persistence. | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **REQ-ID: REQ017 – Quick Witted Response** | | | | | |
| **Alias** | 快速响应 | **Difficulty** | Medium | **Priority** | Medium |
| **Status** | Proposed | **Stability** | Medium | **Type** | Non-Functional |
| **Description** | Any operation should be responded within 3 seconds. | | | | |

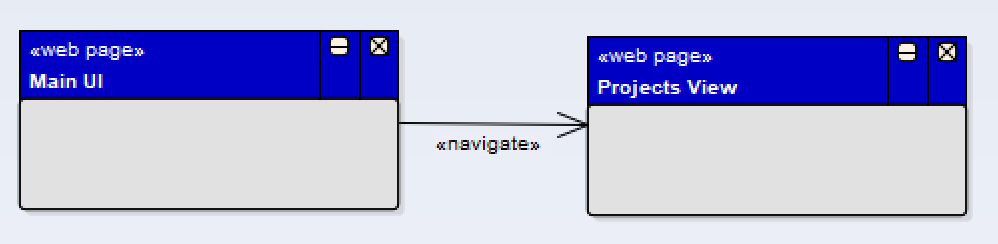
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **REQ-ID: REQ018 – Public API** | | | | | |
| **Alias** | 公共API | **Difficulty** | High | **Priority** | Low |
| **Status** | Proposed | **Stability** | High | **Type** | Non-Functional |
| **Description** | Public API to be possibly used in other systems. | | | | |

// TODO REMOVE WHEN PUBLISH

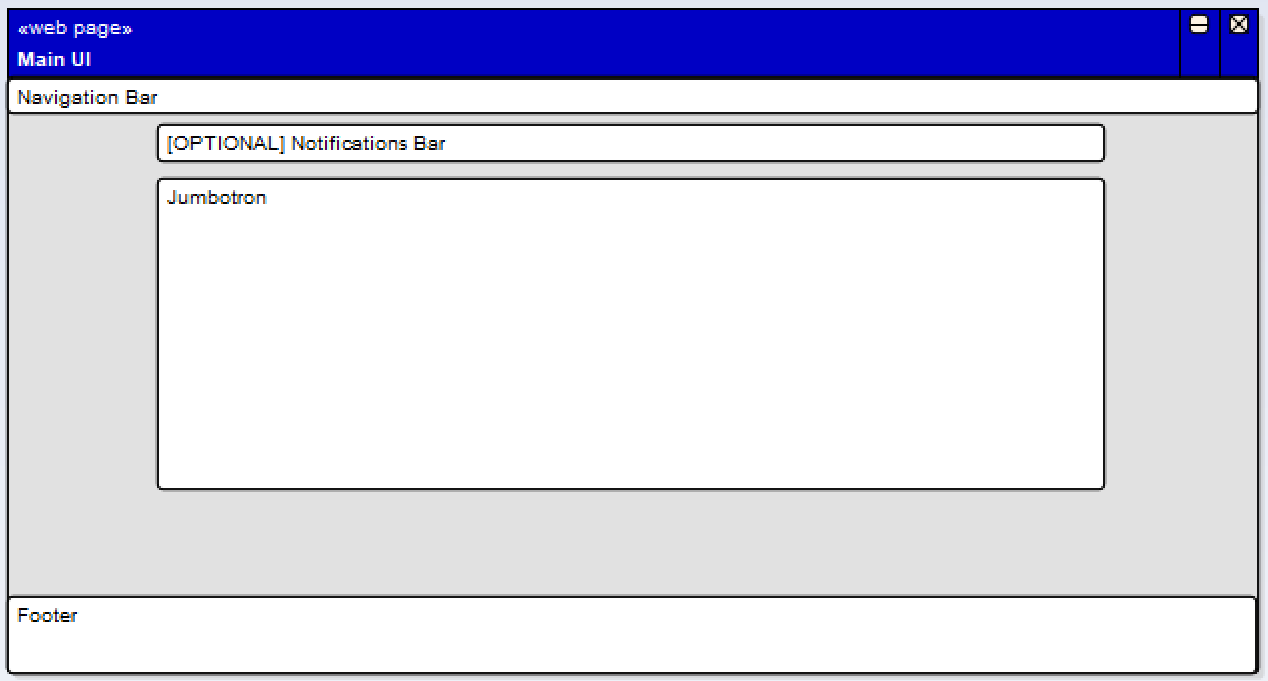
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **REQ-ID: REQ0 -** | | | | | |
| **Alias** |  | **Difficulty** | Medium | **Priority** |  |
| **Status** | Proposed | **Stability** |  | **Type** | Functional |
| **Description** |  | | | | |

## 用户界面

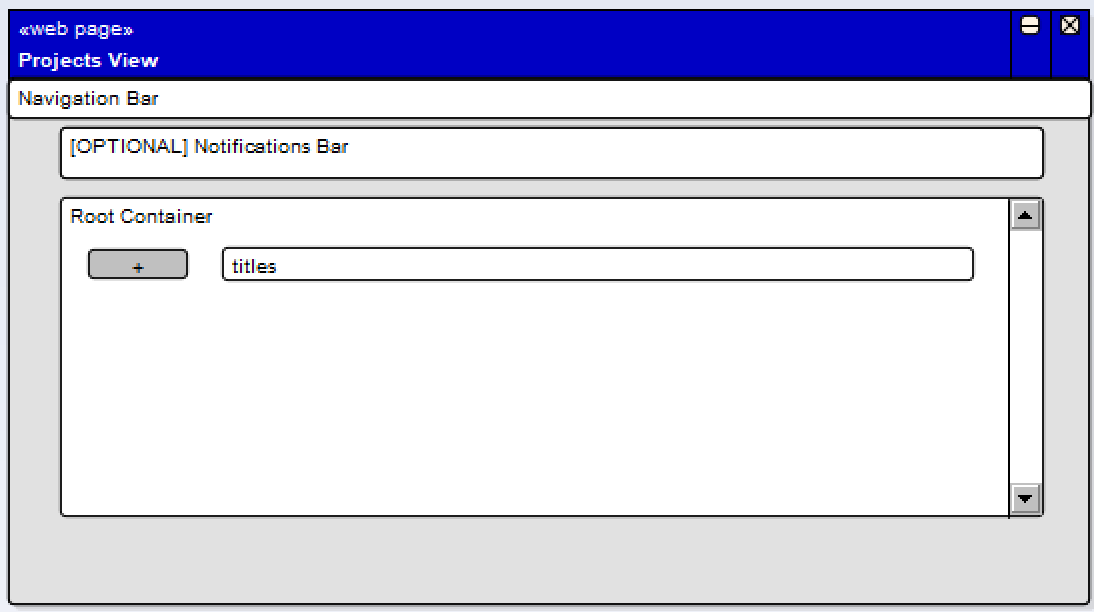
### 界面层级结构



### 主界面设计



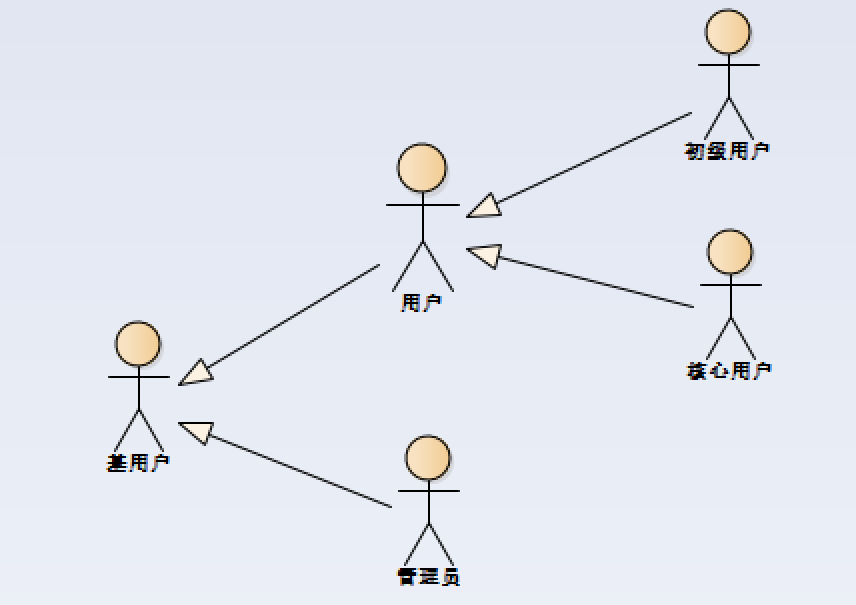
### 项目界面



# 用例

## 参与者

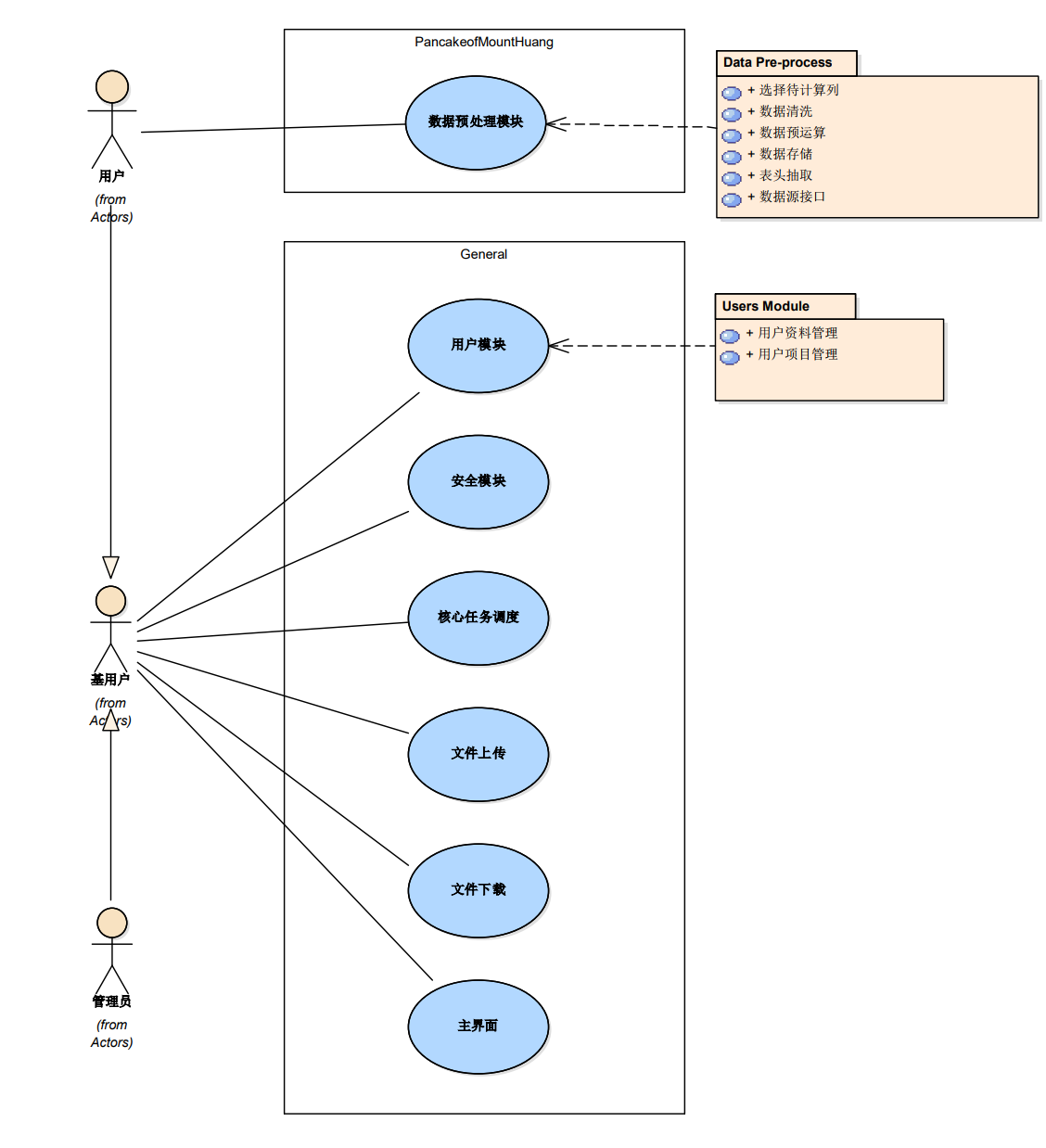
系统主要参与者分为两类，普通用户与管理员用户，均属于基用户，如图3所示。其中，管理员角色主要负责整个系统的管理；核心用户拥有初级用户的所有能力的同时，还能拥有系统提供的优先使用权，如进程的优先处理等。



1. 参与者

## 主要用例

系统主要用例如图4所示，其中根据项目任务的开发计划和本次开发的系统边界，将用例划分为PMH部分和General部分。PMH部分主要完成数据预处理的业务逻辑，而General部分主要完成系统基础设施的建设。



1. **主要用例**