概要设计报告

Project Dawn



预发布 / Pre-Publish

arvinsc@foxmail.com;511679327@qq.com

Dawn-Team



**目 录**

[1 引言 2](#_Toc497373782)

[1.1. 编写目的 2](#_Toc497373783)

[1.2. 项目背景 2](#_Toc497373784)

[1.2.1. 项目的委托单位、开发单位和主管部门 2](#_Toc497373785)

[1.2.2. 该软件系统与其他系统的关系 2](#_Toc497373786)

[1.3. 定义 2](#_Toc497373787)

[1.4. 参考资料 2](#_Toc497373788)

[2 任务概述 4](#_Toc497373789)

[2.1 目标 4](#_Toc497373790)

[2.2 运行环境 4](#_Toc497373791)

[2.3 条件与限制 4](#_Toc497373792)

[3 总体设计 4](#_Toc497373793)

[3.1 处理流程 4](#_Toc497373794)

[3.2 总体结构设计 6](#_Toc497373795)

[3.3 功能分配 6](#_Toc497373796)

[4 接口设计 7](#_Toc497373801)

[4.1 外部接口 7](#_Toc497373802)

[4.2 内部接口 7](#_Toc497373803)

[5 出错处理设计 9](#_Toc497373804)

[5.1 出错输出信息 9](#_Toc497373805)

[5.2 出错处理对策 9](#_Toc497373806)

# 引言

## 编写目的

本文档的主要任务是把需求分析得到的系统扩展用例图转换为软件结构和数据结构。为了将产品需求（详情请参见[《需求分析报告》](#_参考资料)）转化为相对应的软件结构，并为后续的详细设计以及项目开发做框架搭建，为后续项目开发提供基础，项目组成员撰写本文档。

本文档可以作为详细设计文档的撰写基础，并在产品进行开发时作为开发依据、也可作为潜在开发人员（包括产品经理、设计和实现者、质量保证人员以及产品维护人员等）进行产品测试、确认、评审和维护的重要参考资料。

## 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 | 《需求分析报告》 | 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/Outline_Design/PancakeofMountHuang/Doc/Outline%20design.docx)  Specification of PHM |
| 3 | runoob.com | 网站 | <http://www.runoob.com> |
| 4 | 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

关于整个项目的实际运行环境描述，请参见[《需求分析报告》](#_参考资料)2.2节。

## 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。

# 总体设计

## 处理流程

本产品的处理流程主要以数据为核心，围绕数据源，进行展开。

首先，用户可以通过注册账号，来登录系统。在用户登录后，可以选择数据预处理功能。

如图1所示，在数据预处理模块中，用户需要先选择需要处理的数据，如果在个人账户中不存在需要的可选数据记录，则需要上传本地文件到文件服务器，然后再进行数据预处理。在经过一段时间的计算之后，可以在个人项目中查看处理后的数据。



1. 用户操作流程

在数据预处理结束后，会在数据库中存储处理后的数据，以便向后续模块的开发使用。

如图2所示，在数据预处理过程中，系统经过数据源接口读入数据，然后进行表头抽取、反馈到界面，由客户选择待计算的列，选择完成后，系统对相对应的数据建立数据预处理任务（一次任务包括：数据清洗、附加数据运算和数据存储），而后将任务交由任务调度器执行，任务调度器在执行完成这一个任务后，该任务的生命周期即结束。



1. 数据预处理处理流程

## 总体结构设计

如下图所示，以下是本产品的完整结构组件图，从图中可以看到，本产品有四大模块，分别为：通用模块、数据预处理模块（即PMH）、数据分析模块和数据可视化模块。



1. 完整产品的程序组件图

## 功能分配

根据[《需求分析报告》](#_参考资料)中所描述的需求，我们将本产品的系统功能分为如下几种：

|  |  |  |
| --- | --- | --- |
| **通用模块** | | |
| **编号** | **功能名称** | **对应需求** |
| 1 | 用户模块 | 提供用户注册、用户删除和概要更新功能。 |
| 2 | 安全模块 | 提供用户详细信息和权限级别信息。 |
| 3 | 文件管理模块 | 用于数据源文件的输入/输出和管理 |
| 4 | 任务调度模块 | 提供添加/挂起/删除任务到优先级调度队列中的功能。 |

|  |  |  |
| --- | --- | --- |
| **数据预处理模块（PMH）** | | |
| **编号** | **功能名称** | **对应需求** |
| 1 | 数据清洗 | 异常数据计算与更新的数学过程。 |
| 2 | 数据预运算 | 额外增加的数学计算过程。 |
| 3 | 数据存储 | 用于处理数据的存储。 |

# 接口设计

## 外部接口

如下图所示，以下是本产品的完整结构组件图，从图中可以看到，本产品有四大模块，分别为：通用模块、数据预处理模块（即PMH）、数据分析模块和数据可视化模块。

在现阶段，我们的计划是首先实现通用模块和数据预处理模块的开发，并且会在开发过程中，做好预留的外部程序接口，可以提供给其他模块或外部程序来调用。



1. 完整产品的程序组件图

## 内部接口

如下图所示，以下就是本产品的主要模块——数据预处理模块，在此模块中，我们设计向外部提供一组WEB API供用客户端调用。并提供一个数据库接口，用来连接到数据库，来实现数据持久化。为了提高任务处理的效率，对系统资源进行有效的管理，我们设计了一个任务调度接口，以将预处理任务交由系统的任务调度器管理。



1. 数据预处理模块

在通用模块中，我们对所有通用子模块都设计了相应的内部接口，可以提供给其他程序内部模块所调用。



1. 通用模块提供的内部接口图

# 出错处理设计

## 出错输出信息

在用户正常使用产品的过程中，用户易在为告知的情况下触发某些非法操作，如：用户注册时两次输入的密码不同、用户给出的数据预处理数据中含有无法计算的数据内容（多媒体内容、非数字数据等）。为了在用户意外触发非法操作时能够给予用户足够的信息、提升系统稳定性，系统设计中充分考虑到了对相关非法操作的捕获和捕获信息的传达。

在系统中，对所有传递到客户端的JSON数据都进行了统一封装，其封装格式如下：

|  |
| --- |
| 文字报错格式： |
| {status:”status\_info”,code:0,data:{…}} |
| 示例： |
| {"code":0,"loginStatus":"ROLE\_ANONYMOUS","roleCode":0,"status":"OK"} |

其中status字段和code字段的规约如下：

|  |  |  |
| --- | --- | --- |
| 报错信息标准： | | |
| 报错类型 | 异常编号 | 异常解释 |
| OK | 0[0-9]\* | 所有以“0”开头的异常类型，仅能保证程序运行逻辑正常，但可能存在数据异常。 |
| Exception | 1[0-9]\* | 所有以“1”开头的异常类型，均属于系统异常。 |
|  |  |  |

## 出错处理对策

在我们设计本产品的同时，我们考虑到了很多用户在操作时可能会出现的错误，例如：由于用户点击过快而导致的显示错误以及用户未选择需要进行数据预处理的文件就进行后续操作等，我们会对这些常见的错误进行容错处理。

其中，为了处理在服务器中可能出现的非法操作，我们根据实际的业务，定义自定义异常并在出现异常情况时捕获响应的异常，通过这样的异常信息和相关的传输接口/实体，将信息传达到客户端。在客户端（浏览器），我们对用户的非法操作进行文字、视觉和逻辑上的提示和限制，如：用户注册时两次输入的密码不一致时，不仅在输入框上有红色警示，也有文字提示，同时，提交按钮是禁用状态。