# 性能数据管理系统

详细设计文档

Version：1.1

Date：2021-09-22

|  |  |
| --- | --- |
| 项目名称 | 性能数据管理系统 |
| 版本 | V1.1 |
| 责任人 | 白雪公主和她的6个小矮人 |
| 责任部门 | 解决方案管理部 |
| 发布日期 |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Owner | Notes |
| 1.0. | 2021-09-20 | Pei.wu | 起草详细设计文档大纲 |
| 1.1. | 2021-09-22 | Pei.wu | 完成1-3节内容  （文档说明、系统描述、架构及接口设计） |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

*说明：本文档字体大小、行间距及其他格式在后续添加内容时尽量保持一致。*

目 录

[性能数据管理系统 1](#_Toc83294945)

[1 引言 1](#_Toc83294946)

[**1.1** **编写目的** 1](#_Toc83294947)

[**1.2** **范围** 1](#_Toc83294948)

[**1.3** **预期读者和阅读建议** 1](#_Toc83294949)

[**1.4** **参考资料** 1](#_Toc83294950)

[**1.5** **缩写术语** 1](#_Toc83294951)

[2 软件系统描述 1](#_Toc83294952)

[**2.1** **需求规定** 2](#_Toc83294953)

[**2.2** **运行环境** 2](#_Toc83294954)

[**2.3** **应用环境** 2](#_Toc83294955)

[**2.4** **约束条件** 2](#_Toc83294956)

[3 软件系统的总体设计 3](#_Toc83294957)

[**3.1** **软件系统总体设计方案架构** 3](#_Toc83294958)

[**3.2** **子系统之间的关系** 3](#_Toc83294959)

[3.2.1前端与后端 3](#_Toc83294960)

[3.2.2后端与数据库 3](#_Toc83294961)

[**3.3** **资源分配** 3](#_Toc83294962)

[4 后端设计方案 4](#_Toc83294963)

[**4.1** **子系统框架** 4](#_Toc83294964)

[**4.2** **子模块功能** 4](#_Toc83294965)

[**4.3** **内部接口设计方案** 4](#_Toc83294966)

[**4.4** **外部接口设计方案** 4](#_Toc83294967)

[5 前端设计方案 4](#_Toc83294968)

[**5.1** **子系统框架** 4](#_Toc83294969)

[**5.2** **UI设计** 4](#_Toc83294970)

[**5.3** **父、子、并级页面逻辑关系** 4](#_Toc83294971)

[**5.4** **内部接口设计方案** 4](#_Toc83294972)

[**5.5** **外部接口设计方案** 4](#_Toc83294973)

[6 主要性能相关设计 4](#_Toc83294974)

[7 其他非功能性设计 4](#_Toc83294975)

# 引言

性能数据管理系统是一款集一键导入\导出、在线查询、在线可视化分析于一体的PC端网站管理系统，本文旨在提供性能数据管理系统的软件详细设计。

* 1. **编写目的**

1. 使读者再进行软件开发前准确快速的了解本项目软件设计的相关工作及设计原则。
2. 提供参考资料和设计方案、说明软件平台环境。完整性的开发平台和运行环境让开发过程有据可依，精简开发过程。
   1. **范围**

性能数据管理系统软件设计开发。

* 1. **预期读者和阅读建议**

预期读者：结合Django、Mysql、Bootstrap、python3进行PC端性能数据管理网站的开发人员。

阅读建议: 收集参考资料中的文档，熟悉所用技术的功能特性，结合文档资料进行本详细设计内容的阅读。

* 1. **参考资料**

|  |  |  |  |
| --- | --- | --- | --- |
| 编号 | 文献名称 | 版本 | 出版单位 |
| 1 | Django（英文官方） |  | https://docs.djangoproject.com/en/3.2/ |
| 2 | Django(菜鸟教程) |  | https://www.runoob.com/django/django-tutorial.html |
| 3 | MySQL必知必会 |  | 哔哩哔哩、人民邮电出版社 |
| 4 | Bootstrap(已编译版) |  | https://v3.bootcss.com/ |
| 5 | Django（博客） |  | https://www.liujiangblog.com/course/django/95 |

* 1. **缩写术语**

BDFT : BigData-FT

FB：Front to Back

DS：Django & MySQL

DP：Django & Pandas

# 软件系统描述

本软件系统为PC端web管理系统，主要功能是一键导入特定模板数据，在线查询数据，按多种格式导出用户所需数据，在线生成多种可组合的可视化图表。功能分为：后端存储、后端查询、后端成图、前端展示等。

* 1. **需求规定**

|  |  |  |
| --- | --- | --- |
| 性能数据管理平台功能 | | |
| 类型 | 功能 | 描述 |
| 基本  配置 | 一键导入 | 将按照模板上传的文件录入到数据库中 |
| 在线查询 | 按需进行组合查询 |
| 一键导出 | 一键导出为多种文件格式 |
| 在线可视化 | 生成多种格式的图表进行在线分析 |

* 1. **运行环境**

硬件支持：

软件支持：

开发语言：Python、JavaScript、Mysql

开发工具：VSCode、PyCharm、Linux\Windows

编译环境:

1. 利用Linux搭建Mysql服务器
2. 利用vscode或Pycharm进行Django框架设计修改
3. 利用vscode或Pycharm进行HTML设计并调整
4. 利用命令行及chrome对软件进行调试。
   1. **应用环境**

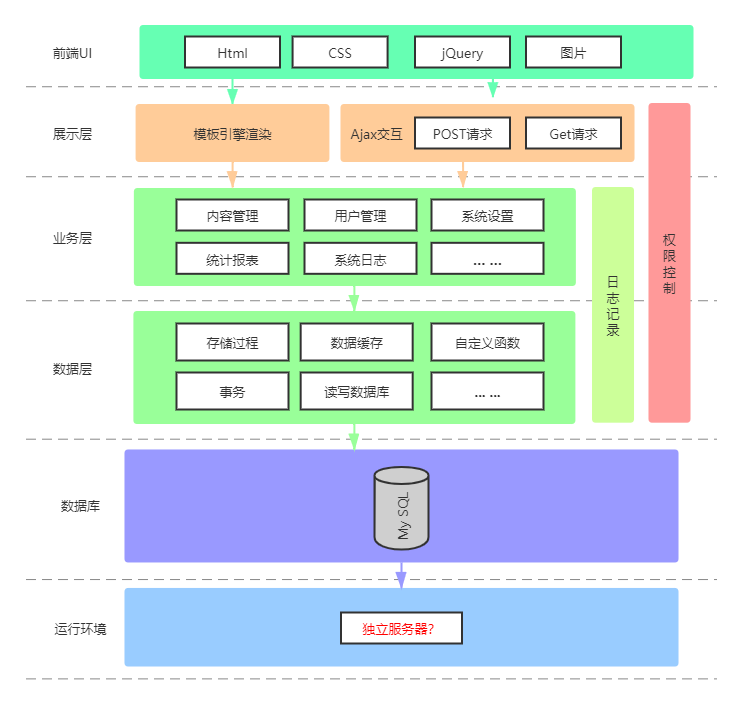
Chrome、Firefox、Edge等PC端浏览器。

* 1. **约束条件**

1. 时延：平均时延不高于公司其他网站。
2. 缓存大小：及时清理或者保留历史缓存数据？是否占用太多浏览器太多内存？
3. 事务隔离：数据库设计是否考虑到事务隔离性？

# 软件系统的总体设计

* 1. **软件系统总体设计方案架构**



**硬件（虚拟）：**

数据库：MySQL

后端（Web）框架：Django==X.x.x

前端框架：Bootstrap==3.x.x（预编译版）

**接口：**

FB：前端与后端的接口

DS：Django与数据库的接口

DP：Django与可视化脚本的接口

……

* 1. **子系统之间的关系**

3.2.1 前端与后端

**模块选择：**通过当前发起的任务（前端体现在用户点击了某个功能按钮）去进行相应的**前端和后端**模块或函数查找（该部分需要内部函数接口及外部调用接口定义明确，避免交叉错误）。

**数据传输：**前端与后端之间的数据通过接口传输，本系统设计理念是，当发起数据传输功能时，数据传输功能的优先级设置为最高，页面缓存若出现bug可以通过用户手动刷新来解决。

**IDLE与Connect:**此部分主要考虑到在线查询过程中不断进行多个查询任务切换带来的服务器压力。（暂时可以先不考虑该性能问题）

3.2.2 后端与数据库

**增：**通过**一键导入**的文件，通过关键字查询锁定主键，然后录入到相应的数据库中。

**删：**历史数据的清理（仅限管理员）

**改：**数据整理（项目整合与分支管理，仅限管理员）

**查：**通过在线查询按钮及组合方式，实现在线查询。前端需与后端保持一致，且确保后端能够在相应数据库找到该字段，即该字段为有效字段，否则，返回404page.

**PS: 删和改仅为管理员功能，因此在前端的设计上应另增一个管理员页面及其功能。**

* 1. **资源分配**

待进一步讨论……

# 后端设计方案

* 1. **子系统框架**
  2. **子模块功能**
  3. **内部接口设计方案**
  4. **外部接口设计方案**

# 前端设计方案

* 1. **子系统框架**
  2. **UI设计**
  3. **父、子、并级页面逻辑关系**
  4. **内部接口设计方案**
  5. **外部接口设计方案**

# 主要性能相关设计

# 其他非功能性设计