**架构设计文档**

|  |  |  |  |
| --- | --- | --- | --- |
| 角色 | 小组成员 | 班级 | 得分 |
| 组长 | 李霞芳 | 17级软件工程1班 |  |
|  | 成子谦 | 17级软件工程1班 |  |
| 郭蕴雪 | 17级软件工程1班 |  |
| 蔡泽莎 | 17级软件工程1班 |  |
|  | 武云峰 | 17级软件工程1班 |  |

**软件学院**

**2018年12月**

目录

[1 概述 2](#_Toc531536090)

[1.1 系统简述 2](#_Toc531536091)

[1.2 名词解释 2](#_Toc531536092)

[2 架构设计规约 2](#_Toc531536093)

[2.1 设计思路 2](#_Toc531536094)

[2.2 设计目标 3](#_Toc531536095)

[2.3 设计约束条件 3](#_Toc531536096)

[3 系统总体结构设计 6](#_Toc531536097)

[3.1 系统架构图 6](#_Toc531536098)

[3.2 模块分解视图 7](#_Toc531536099)

[3.3 组件连接器视图 8](#_Toc531536100)

[3.4 分配视图 9](#_Toc531536101)

# 概述

## 系统简述

（1）背景：信息技术与经济社会的交汇融合引发了数据迅猛增长，数据已成为国家基础性战略资源，大数据正日益对全球生产、流通、分配、消费活动以及经济运行机制、社会生活方式和国家治理能力产生重要影响。2015年，国务院印发了《促进大数据发展行动纲要》，纲要中提到，数据已成为国家基础性战略资源。目前，我国在大数据发展和应用方面已具备一定基础，拥有市场优势和发展潜力，但也存在政府数据开放共享不足、产业基础薄弱、缺乏顶层设计和统筹规划、法律法规建设滞后、创新应用领域不广等问题，亟待解决。

（2）系统描述：本系统旨在解决各政府部门的数据孤立问题，是一个数据共享云平台，也是一个政府统一登陆平台，方便用户管理不同的系统上面的信息

（3）系统功能：统筹各个信息登记系统的信息，为用户提供统一登记平台，免去多次登陆的繁琐，且能够在移动端使用；实现不同的数据库之间的数据交互；自动生成相关数据报表与数据分析结果，并能给出一些智能化决策建议。

## 名词解释

|  |  |  |
| --- | --- | --- |
| 序号 | 名词或术语 | 解释说明 |
|  | 软件前端 | 与用户直接交互，并把用户请求交付给后端处理的模块 |
|  | 软件后端 | 从软件前端获取用户请求并进行处理的模块 |
|  | 请求处理模块 | 负责高效率处理用户请求，并保证服务可用的模块 |
|  | 数据库 | 存放用户信息的模块 |
|  | 安全组件 | 包括系统日志、系统进程监视、系统防火墙等组件 |
|  | 服务检测组件 | 检测服务是否可用的组件 |

# 架构设计规约

## 设计思路

（1）分层：我们将整个系统分为四层：交互层、逻辑层、处理层、数据层。交互层负责与用户进行直接交互；逻辑层负责监控系统状态并传递用户请求；处理层用于处理用户请求；数据层用于存储用户数据。处理层和数据库之间用统一的中间件连接，达到统筹数据库的作用，为服务层提供统一的接口，后面每接入一个新的数据库仅需要提供一个新的连接中间件即可，极大增加了可拓展性。

（2）分隔：各个子系统依旧保留原本的功能以及数据格式，统一登陆平台只提供一个登陆功能，要进行实际的操作需要从子系统中进行，这减少了系统维护的难度，将不同的业务分割开来，由不同的团队负责

（3）安全：统一登陆平台提供唯一的识别码例如身份证以及密码，并提供手机验证码登陆或者密码找回。进入统一登陆平台后自动为此用户绑定其他子系统的账号，而对于安全性要求高的子系统则要求二次密码确认之后才进行账号绑定，第一次绑定之后后面就可以直接从统一登陆平台跳转到对应的子系统。

## 设计目标

（1）系统具有互操作性：可实现不同数据库之间信息交互与信息纠错差错能力。

（2）系统具有安全性：正确识别合法用户与非法用户，确定用户来自系统内部或外部，及时处理用户非法操作。

（3）系统具有易用性：实现各个系统数据库之间的信息交互，减少数据库的冗余与用户的冗余操作。

（4）系统架构具有可用性：要求系统在正常情况下可以使用，发生错误时有应急方案并能修复错误。

（5）平台架构具有高性能：支持高并发，提高平台访问速度和数据库查询速度。

## 设计约束条件

**（1）系统设计约束：**

A）明确系统使用的架构，并考虑架构的可扩展性。

B）系统内各个模块之间关系如何，如何通信。

C）每个模块的外部接口是什么，返回值是什么，不能有二义性。

D）如果系统较为庞大，可将系统分为子系统。须明确定义子系统关系，子系统的设计要求同系统的设计要求。

E）系统设计前，必须编写软件功能规格说明书，以明确功能需求。也可在UML中使用用例图来描述软件功能。

F）系统设计中所有的交互行为只描述到接口。

**（2）数据库设计约束：**

A）数据库设计最低要求符合BC范式。

B）必须明确数据库内表、视图、存储过程、关系。

C）表的设计必须明确字段数据类型、长度、默认值、是否非空，此类型不必是具体数据库类型。

D）存储过程必须明确参数和返回值。

E）必须明确增、删、查、改时的处理方式。

F）对于一年内记录会超出百万的数据，必须考虑性能的问题。及时备份并删除冗余数据。

**（3）接口/协议约束：**

A）一般情况下不要求模块间使用独立的接口进行通信，但必须明确模块之间的关联关系。

B）明确说明每个内部接口的信息。参数与返回值必须明确定义。

C）子系统间和层间不得通过已设计的接口以外的任何方式进行通信。

D）硬件接口必须明确说明通信的方式以及使用的协议，如不是通用协议，必须详细说明协议格式。

**（4）用户界面的约束：**

A）用户界面适合于软件的功能 ；容易理解；风格一致；及时反馈信息；出错处理；合理的布局；和谐的色彩。

B）用UML表述界面与人互动的所有接口，包含显示信息、输入信息、操作信息。所有的信息必须明确无二义。

C）制作可与用户进行动态互动的具体界面，优先使用开发语言进行设计。

D）在UML中用序列图描述各个界面与应用服务接口之间的交互关系。

**（5）安全设计：**

A）用户身份验证设计

B）用户授权设计

C）数据安全设计

D）安全审核设计

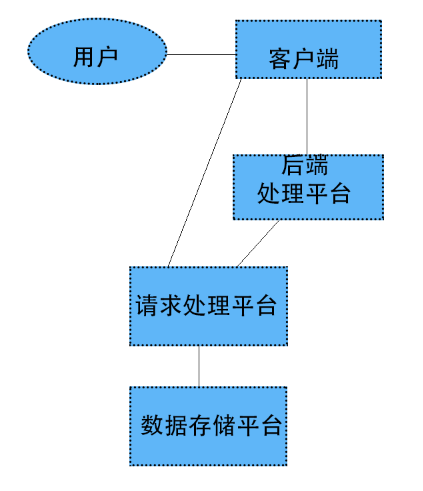
（6）软件、硬件环境（包括运行环境和开发环境）的约束：

A）软件开发环境、运行环境：Red hat linux。

B）硬件开发环境、运行环境：至少32G运行内存+E5+1024T硬盘

# 系统总体结构设计

## 系统架构图

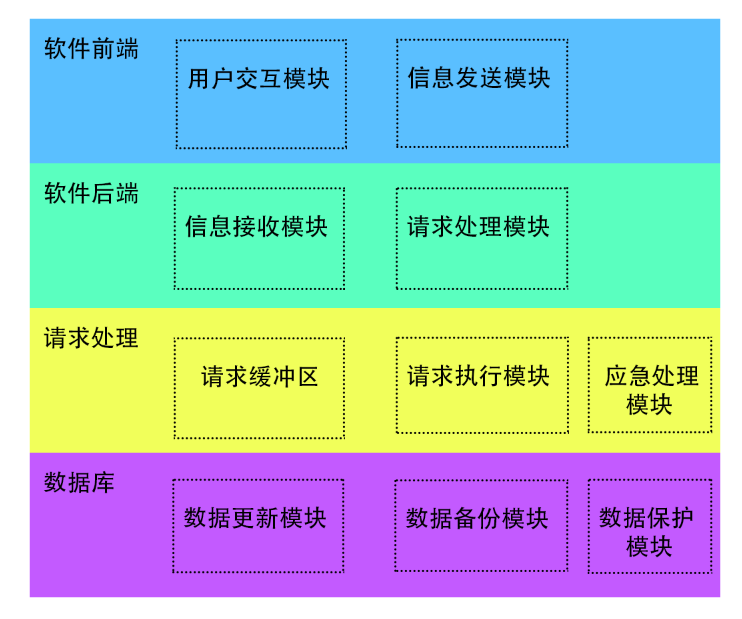


## 模块分解视图

分解所遵循的维度：

1. 第一维度：软件前后端、请求处理模块、数据库
2. 第二维度：每个模块的功能

分解所遵循的规则：高内聚、低耦合



## 组件连接器视图

## 

## 分配视图

