|  |  |
| --- | --- |
|  | |
| **《企业大数据征信系统》**  **项目概要设计**  **V1.0.1** | |
|  |

**版 本 历 史**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 版本/状态 | 作者 | 参与者 | 日期 | 备注 |
| 1.0.0 | 黄启旭 |  | 2023-08-25 | 创建 |
| 1.0.1 | 全体组员 |  | 2023-08-25 | 修订 |
| 1.0.2 | 全体组员 |  | 2023-08-30 | 完善项目概述、项目约束和项目功能概要结果部分 |
| 1.0.3 | 全体组员 |  | 2023-08-31 | 完善总体设计、运行环境和部署部分 |

目录

第一部分 引言

[一、说明 3](#_Toc_1)

[二、术语定义 3](#_Toc_9)

第二部分 项目概述

[一、项目描述 5](#_Toc326323006)

[二、项目功能描述 5](#_Toc326323007)

第三部分 设计约束

[一、需求约束 9](#_Toc304552738)

[二、隐含约束 10](#_Toc326049727)

第四部分 企业大数据征信系统业务设计

[一、安全设计 11](#_Toc326049729)

[二、企业大数据征信平台相关业务流程 11](#_Toc304552751)

[三、业务功能概要结构 12](#_Toc189556075)

[四、模块定义 18](#_Toc326049734)

第五部分 E-R实体设计

[一、E-R实体结构图 19](#_Toc326323060)

[二、实体描述 19](#_Toc326323061)

第六部分 总体设计

[一、企业大数据征信系统逻辑架构设计 22](#_Toc293583298)

[二、物理架构设计 23](#_Toc293583299)

[三、技术架构设计 23](#_Toc293583300)

第七部分 用户界面设计

[一、征信查询风格设计 25](#_Toc304552772)

[二、用户，管理员界面展示 28](#_Toc304552773)

第八部分 运行环境和部署

[一、运行环境 30](#_Toc445698284)

[二、系统性能要求 30](#_Toc304552776)

第一部分 引言

### **一、说明**

编写本说明书的目的是为了准确阐述项目具体业务需求和需求边界，本说明书的作者是爽划慢填项目组，本说明书的确认者是组长侯博凡，本说明书的读者是项目所有直接干系人。

本说明书是指导项目实施的重要指导性文件，也是用户最后进行验收的依据，说明书中内容一旦确认双方将以此为基础开展工作。如果需要变更说明书内容，必须走变更流程，变更必须得到甲乙双方书面确认，最后变更内容将作为本文的一部分，在项目实施过程中得以体现。

### **二、术语定义**

#### **1、企业征信**

**企业征信**，是指征信机构作为提供信用信息服务的企业，按一定规则合法采集企业、个人的信用信息，加工整理形成企业、个人的信用报告等征信产品，有偿提供给经济活动中的贷款方、赊销方、招标方、出租方、保险方等有合法需求的信息使用者，为其了解交易对方的信用状况提供便利。

#### **2、征信业务**

**征信业务**，是指对企业、事业单位等组织（以下统称企业）的信用信息和个人的信用信息进行采集、整理、保存、加工，并向信息使用者提供的活动。

#### **3、不良信息**

**不良信息**，是指对信息主体信用状况构成负面影响的下列信息：信息主体在借贷、赊购、担保、租赁、保险、使用信用卡等活动中未按照合同履行义务的信息，对信息主体的行政处罚信息，人民法院判决或者裁定信息主体履行义务以及强制执行的信息，以及国务院征信业监督管理部门规定的其他不良信息。

#### **4、信用编码**

11315全国企业征信系统为每一家合法注册的企事业单位都匹配独一无二的信用编码，由8位阿拉伯数字组成，终身免费归企业所有，信用编码是企业信用的门牌号码，具有唯一性，不重复，在企业经营，营销推广，赢得市场信赖等方面有广泛的用途。通过信用编码，可以查看包含对应企事业单位的全面信用信息，包括：全国各级政府职能部门公共监管信息；金融机构（包括小额贷款公司、民间借贷等）对企业的信贷评价信息； 行业协会（社团组织）的评价信息；主流媒体对企业发布评价的评价信息；企业管理评估信息及财务信息；市场反馈信息（包括消费者、交易对方、合作伙伴及员工等主体在线实名提交的评价信息）。

#### **5、信用网址**

根据国务院《征信业管理条例》，利用十多年来积累的大量数据，依法为所有企业建成网上信用档案，每个信用档案均由一组8位数字组成的统一“信用编码”与之唯一对应、永久绑定，“信用编码”+“11315.com”后缀即形成该企业信用档案的“信用网址”，登录该信用网址随时可上网查询其信用档案。

第二部分 项目概述

### 一、项目描述

企业大数据征信系统项目旨在建立一个综合性信息平台，通过整合多渠道数据如财务、税务、信用记录等，为金融机构、供应商等提供全面准确的企业信用评估，从而支持决策制定与风险管理。项目包括数据收集、存储管理、分析挖掘等环节，以提升对企业经营状况的洞察力和决策效率。

### 二、项目功能描述

用户业务主要包含查询企业、关注企业、全国企业概览以及企业对比。

查询企业包含四个主要业务：股票代码查询、信用代码查询、关键词（企业名称）查询和索引查询；

关注企业包含两个主要业务：设置关注和关注提醒；

全国企业概览包含两个主要业务：地区数据分析和行业数据分析；

企业对比包含：数据对比

#### 1、查询企业

三种方式查询返回的结果格式相同，

主要信息包括：

1概况信息：公司全称、公司简称、法人代表、挂牌日期、行业分类、注册地址、注册日期、注册资本、主营产品

2.公司高管：人员级别，人员职位，人员姓名

3.主要股东（TOP10）：股东名称、持股数、持股比例

4.财务指标：营业收入、营业利润、净利润、未分配利润、总资产、总负债、净资产、每股收益

5.征信情况：是否被纳入征信黑名单、执行案号、立案时间

1>股票代码查询

根据企业的股票代码查询相关企业

任务信息包括：企业的股票代码：\*\*\*\*\*\*（六位）

点击查询后执行。（可以增加字符匹配，检查该字段是否是六位）

2>信用代码查询

根据企业的信用代码查询相关企业

任务信息包括：企业的信用代码：\*\*\*\*\*\*\*\*（八位）

点击查询后执行。（可以增加字符匹配，检查该字段是否是八位）

3>关键字（企业名称）查询

根据企业的名称模糊查询相关企业

任务信息包括：输入的关键字（\*\*\*）

点击查询后执行。

4>（地区）索引查询

根据索引查询相关企业

任务信息包括：下拉框|输入框中的地区字段（\*\*\*）

点击查询后执行。

#### 2、全国/地区/行业企业概览

用户在主页当中点击“企业总数据分析”按钮可以进入全国企业数据分析页面。

1>企业总数据分析：

用户在主页当中点击“企业总数据分析”按钮可以进入全国企业数据分析页面，该功能需要登录。在该界面用户可以查看全国各地区企业总体情况，包括各地区企业数量占企业总体比例、全国失信企业地区分布情况、上市企业占比等，用户可以点击自己想要查看的相应数据分析界面按钮进行浏览。

输入信息：用户点击界面提供的不同按钮（按钮上是对应类型数据的名称），可跳转不同数据页面。

返回信息：

a.各地区企业数量占企业总数比例(柱状图+饼状图)

b.各地区近2个月失信企业占全国失信企业总体比例(柱状图+饼状图)

c.各地区上市企业资产总和占所有上市企业总资产比例(饼状图)

d.各地区上市企业净资产总和占所有上市企业总净资产比例(饼状图)

e.各行业上市企业资产总和占所有上市企业总资产比例(饼状图)

f.各行业上市企业净资产总和占上市企业总净资产比例(饼状图)

g.各地区上市企业净利润总和占所有上市企业总净利润比例(饼状图)

h.各行业上市企业净利润总和占所有上市企业总净利润比例(饼状图)

i.各地区上市企业负债总和占上市企业总负债比例(饼状图)

j.各行业上市企业负债总和占上市企业总负债比例(饼状图)

2>地区数据分析：

在地图中选择想要进行数据分析的省份，随后加载相对应企业的总体分析信息：

任务信息：地区字段（\*\*\*）

返回信息：a.地区企业总数及各行业企业占地区企业总数比例(柱状图)

b.地区各行业上市企业总资产占地区企业总资产比例(饼状图)

c.地区各行业上市企业营业收入占地区总营业收入比例(柱状图)

d.地区各行业上市企业总负债占地区上市企业总负债比例(饼状图)

e.地区各行业上市公司净利润总和比较(饼状图)

f.地区所有上市公司的负债占所有上市公司总资产的比例(饼状图)

3>行业数据分析：

在全国企业数据分析页面下点击对行业进行选择，，页面内显示当前选择的地区（全国或某一省份）的该行业的近期数据.

任务信息：行业字段（\*\*\*）

返回信息：参考地区数据分析参数及数据总结类型，交换地区与行业的影响主次。

第三部分 设计约束

### 一、需求约束

#### 1、本系统应当遵循的技术标准

数据命名的规则遵循《Objective-C Style Guide》中相关的规定；

#### 2、软、硬件环境标准

本系统采用C/S架构。Object-C2.0编写，数据库采用MYSQL；大数据处理采用Hadoop+Hive+sqoop；后端使用Springboot+MYbatis完成数据的查询、修改工作等

#### 3、接口/协议标准

本系统通过http协议实现客户端与服务端之间的联通。

#### 4、用户界面标准

使用VUE3+Element—plus+Echarts标准UI库，并进行界面全部美工优化。

#### 5、软件质量

1）正确性

系统请求必须能够被正确处理；

2）健壮性

系统应能够7\*24小时无故障运行；

3）效率性

系统可以支持100个终端同时发起业务，处理业务的时间不超过10秒钟；

4）易用性

界面应采用图形化操作方式，便于业务人员操作；

5）安全性

报文中的关键数据域以密文的方式传输；

6）可扩展性

应该充分考虑到将来业务的修改或增加，避免需求变更时大规模修改程序。

### 二、隐含约束

1）用户具有基本的业务技能和基本的电脑知识，对我们提供的操作界面应保证他们经过简单培训后无障碍的操作；

2）应该把有可能变动的参数存放到配置文件或数据库中，保证修改参数的灵活性。

第四部分 企业大数据征信系统业务设计

### 一、安全设计

系统采用基于阿里云的云端数据库服务，用户名密码校验在后台完成。

### 二、企业大数据征信平台相关业务流程

#### 1、征信查询流程

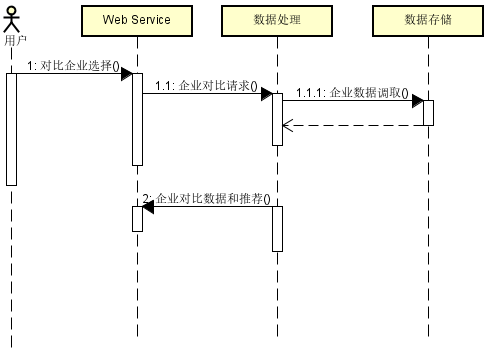
#### 

**说明：**

征信查询流程包括：

* 用户向平台输入查询信息
* 网页向后端发送相应的查询请求
* 后端从数据库将数据调出
* 前端根据调出的数据新型绘图和展示

#### 2、征信对比流程



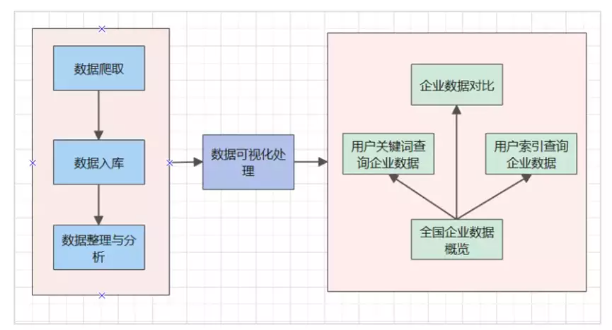
**说明：**

征信对比的主要业务流程包括：

* 用户向Web Service 告知征信对比选择的对象
* Web Service向后端发送对比请求
* 后端从数据库中调出数据，给出建议并将数据发送回Web Service进行展示

### 三、业务功能概要结构

企业大数据征信平台作为应用系统的终端，逻辑结构图如下：



#### 1、数据收集与分析模块



##### 1.1、数据爬取

1、系统从https://www.neeq.com.cn上爬取相关企业的数据，包括：1>RedWEBBaseInfo 共15个属性，依次为:

stockNumber 股票代码

fullName 公司全称

briefName 公司简称

legalRepresentative 法人代表

industryClassification 行业分类

listingDate 挂牌日期

area 所属地区

businessIncome 营业收入

businessProfit 营业利润

pureProfit 净利润

undistributedProfit 未分配利润

totalAssets 总资产

totalDebt 总负债

pureAssets 净资产

profitPerShare 每股收益

2>RedWEBExec 企业执行人，共5个属性，依次为：

creditNumber 信用代码

fullName 公司全称

level 人员级别

position 人员职位

personName 人员姓名

3>RedWEBHolder 企业股东，共6个属性，依次为：

creditNumber 信用代码

fullName 公司全称

rank 股东排名

holderName 股东名称

holdAmount 持股数

holdRate 持股比例

2.从https://www.11315.com/上爬取相关企业征信数据，包括：

1>失信人名单各列属性值：

单位名称

组织机构代码证

执行案号

关注

立案时间

2>企业补充数据：

|  |  |
| --- | --- |
| creditNumber | 信用代码 |
| registrationAddress | 注册地址 |
| registrationDate | 注册日期 |
| registeredAssets | 注册资本 |
| mainProducts | 主营产品 |

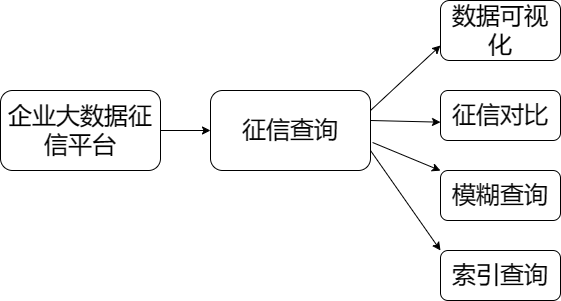
##### 1.2、数据处理

通过hive将爬取到的数据导入近hdfs，然后再通过hive对数据及表单进行处理，生成新的数据表；最后将所需要的数据表导出到云端MYSQL。

在后期处理中，将从https://www.neeq.com.cn和https://www.11315.com/爬取到的企业基本信息合并为一张表，合并后的企业基本信息情况表应为：

|  |  |  |  |
| --- | --- | --- | --- |
| **编号** | **英文名** | **中文名** | **数据类型** |
| 1 | creditNumber | 信用代码 | String |
| 2 | stockNumber | 股票代码 | String |
| 3 | fullName | 公司全称 | String |
| 4 | briefName | 公司简称 | String |
| 5 | legalRepresentative | 法人代表 | String |
| 6 | industryClassification | 行业分类 | String |
| 7 | listingDate | 挂牌日期 | String |
| 8 | area | 所属地区 | String |
| 9 | registrationAddress | 注册地址 | String |
| 10 | registrationDate | 注册日期 | String |
| 11 | registeredAssets | 注册资本 | String |
| 12 | mainProducts | 主营产品 | String |
| 13 | businessIncome | 营业收入 | Double |
| 14 | businessProfit | 营业利润 | Double |
| 15 | pureProfit | 净利润 | Double |
| 16 | undistributedProfit | 未分配利润 | Double |
| 17 | totalAssets | 总资产 | Double |
| 18 | totalDebt | 总负债 | Double |
| 19 | pureAssets | 净资产 | Double |
| 20 | profitPerShare | 每股收益 | Double |

#### 2、征信查询模块



##### 2.1、关键词查询

1、开始初始化模糊查询主界面。

2、进入主页面，进行模糊查询业务操作。

3、用户输入关键词。

4、系统向服务器请求关键词对应的数据。

5、跳转到查询结果页

6、点击查询结果跳转到征信详情

##### 2.2、索引查询

1、开始初始化模糊查询主界面。

2、进入主页面，进行索引业务操作。

3、用户输入索引，填入索引选项。

4、系统向服务器请求关键词对应的数据。

5、跳转到查询结果页

6、点击查询结果跳转到征信详情

##### 2.3、数据可视化

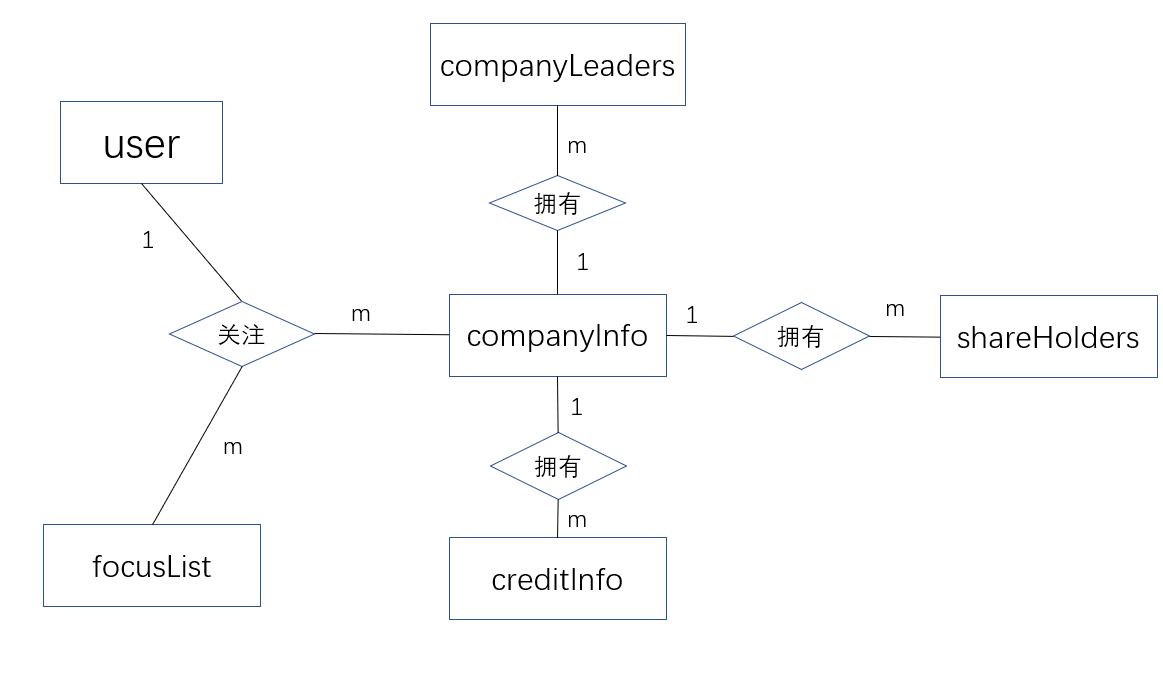
在企业征信和企业对比界面，将数据通过图表的方式展现出来

### 四、模块定义

|  |  |  |
| --- | --- | --- |
| 模块 | 组件 | 规格/型号 |
| 数据收集与分析 | 企业数据爬取 | 从网络上爬取企业的征信数据 |
| 数据处理 | 对爬取的数据进行清洗、整理与分析 |
| 征信查询单元 | 模糊查询 | 通过关键字来调取数据 |
| 索引查询 | 通过索引来调取数据 |
| 数据可视化 | 将数据通过图表方式呈现 |

第五部分 E-R实体设计

### 一、E-R实体结构图



### 二、实体描述

#### 1、companyInfo实体描述

中文名：

|  |  |  |  |
| --- | --- | --- | --- |
| **编号** | **英文名** | **中文名** | **数据类型** |
| 1 | creditNumber | 信用代码 | String |
| 2 | stockNumber | 股票代码 | String |
| 3 | fullName | 公司全称 | String |
| 4 | briefName | 公司简称 | String |
| 5 | legalRepresentative | 法人代表 | String |
| 6 | industryClassification | 行业分类 | String |
| 7 | listingDate | 挂牌日期 | String |
| 8 | area | 所属地区 | String |
| 9 | registrationAddress | 注册地址 | String |
| 10 | registrationDate | 注册日期 | String |
| 11 | registeredAssets | 注册资本 | String |
| 12 | mainProducts | 主营产品 | String |
| 13 | businessIncome | 营业收入 | Double |
| 14 | businessProfit | 营业利润 | Double |
| 15 | pureProfit | 净利润 | Double |
| 16 | undistributedProfit | 未分配利润 | Double |
| 17 | totalAssets | 总资产 | Double |
| 18 | totalDebt | 总负债 | Double |
| 19 | pureAssets | 净资产 | Double |
| 20 | profitPerShare | 每股收益 | Double |

#### 2、companyLeaders实体描述

|  |  |  |  |
| --- | --- | --- | --- |
| **编号** | **英文名** | **中文名** | **数据类型** |
| 1 | stockNumber | 股票代码 | String |
| 2 | fullName | 公司全称 | String |
| 3 | level | 人员级别 | Int |
| 4 | position | 人员职位 | String |
| 5 | personName | 人员姓名 | String |

#### 3、shareholders实体描述

|  |  |  |  |
| --- | --- | --- | --- |
| **编号** | **英文名** | **中文名** | **数据类型** |
| 1 | stockNumber | 股票代码 | String |
| 2 | fullName | 公司全称 | String |
| 3 | rank | 股东排名 | Int |
| 4 | holderName | 股东名称 | String |
| 5 | holdAmount | 持股数 | Int |
| 6 | holdRate | 持股比例 | Double |

#### 4、creditInfo实体描述

|  |  |  |  |
| --- | --- | --- | --- |
| **编号** | **英文名** | **中文名** | **数据类型** |
| 1 | creditNumber | 组织机构代码证 | String |
| 2 | fullName | 公司全称 | String |
| 3 | recordNumber | 执行案号 | String |
| 4 | attention | 关注人数 | int |
| 5 | recordTime | 立案时间 | String |

#### 5、userInfo实体描述

|  |  |  |  |
| --- | --- | --- | --- |
| **编号** | **英文名** | **中文名** | **数据类型** |
| 1 | uid | 用户编号 | Int |
| 2 | username | 用户名称 | String |
| 3 | password | 用户密码 | String |
| 4 | isManager | 是否为管理员账户 | Int |

#### 6、focusList实体描述

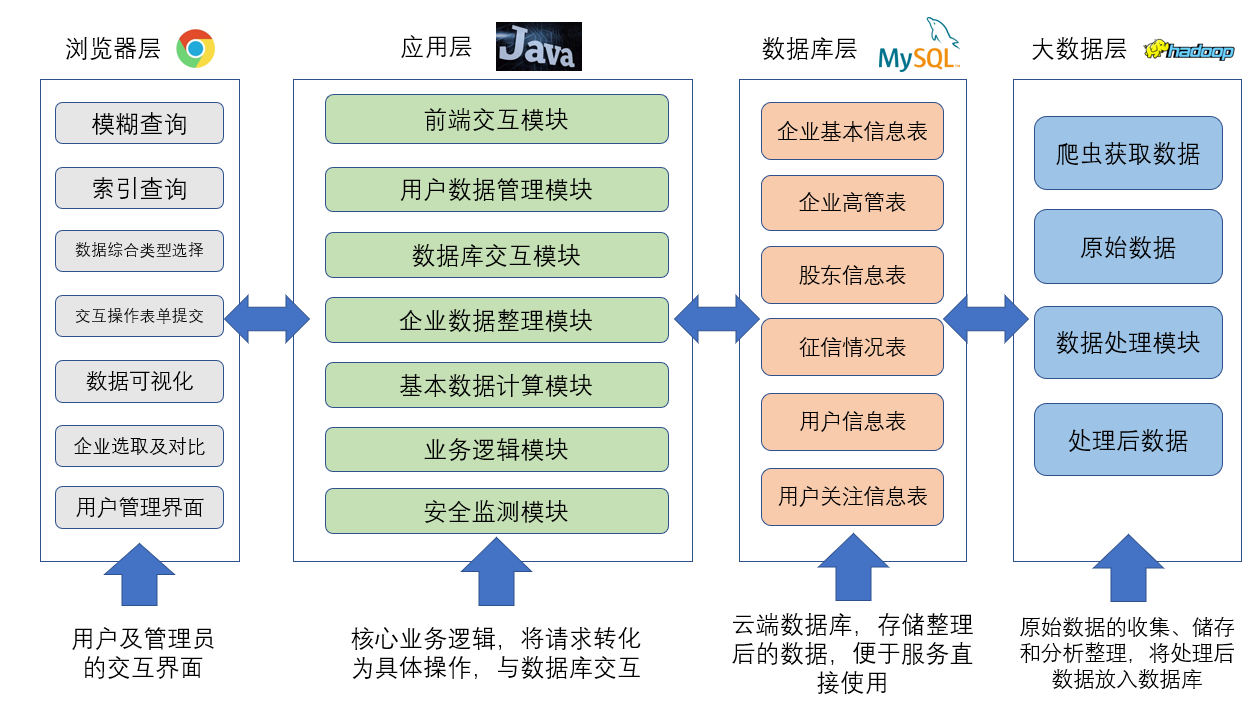
|  |  |  |  |
| --- | --- | --- | --- |
| **编号** | **英文名** | **中文名** | **数据类型** |
| 1 | uid | 用户编号 | Int |
| 2 | creditNumber | 信用代码 | String |
| 3 | fullName | 公司全称 | String |

#### 7、updateTime实体描述

|  |  |  |  |
| --- | --- | --- | --- |
| **编号** | **英文名** | **中文名** | **数据类型** |
| 1 | latestUpdateTime | 最近一次更新时间 | String |

第六部分 总体设计

### 一、企业大数据征信系统逻辑架构设计



#### 1、J2EE四层架构

从构件化与集成化的角度出发，采用J2EE四层架构的技术方案、构件化的开发模式，应用功能的实现在支撑构件的基础上完成。

1.客户端层（Presentation Layer）： 使用VUE构建前端界面，用于与用户进行交互和展示数据。VUE将用户的操作和请求转化为HTTP请求，发送到后端进行处理。

2. Web层（Web Layer）： Web层充当前后端交互的桥梁，处理来自客户端的请求，并将它们传递给后端进行处理。在这个层次，Spring Boot可以作为你的Web层框架。

控制器层：使用Spring MVC处理HTTP请求，解析请求参数，调用适当的服务层进行业务处理。

服务层：实现业务逻辑，可能涉及到数据处理、数据分析、征信计算等。

数据访问层：与数据库进行交互，执行数据的CRUD（创建、读取、更新、删除）操作。

3.业务逻辑层（Business Logic Layer）： 这个层次包含应用程序的核心业务逻辑，将用户的请求转化为具体的业务操作，可能需要与数据库和数据处理组件交互。

服务层：包含业务逻辑，负责协调不同模块的操作，可能涉及到数据处理、业务规则等。

4. 数据访问层（Data Access Layer）：数据访问层负责与数据库进行交互。这个层次涉及到与Hive和Sqoop的交互。

#### 2、采用B/S结构

B/S结构具有跨平台访问、易于维护和管理的特点。通过合理的前端和后端设计，以及与数据库和其他服务的良好集成，可以构建出一个功能丰富、安全可靠的大数据征信系统。

### 二、物理架构设计

1. 客户端层：使用Web服务器来托管前端VUE应用，处理用户的HTTP请求。

2. Web层：部署多Spring Boot应用服务器来处理Web层逻辑。

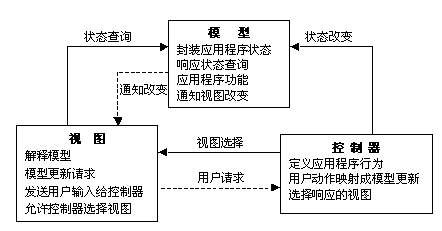
3. 数据访问层：部署数据访问服务器，与Hive和Sqoop进行交互。Hive在Hadoop集群上部署， Sqoop可以在批量数据导入导出服务器上运行。

4. 数据库层：针对Hive和Sqoop，构建Hadoop集群来存储和处理大数据。

### 三、技术架构设计

#### 1、MVC模型设计

系统实现MVC系统架构，使用SDK4.3标准库，UIView的界面层，UIViewController负责架构结合，NSUserDefaults和CoreData作为模型层进行数据的持久化。



1）Controller层

每个业务功能的主操作场景采用一个核心控制器类，利用SDK对移动终端设备的先进场景切换设计，由核心控制器控制不同业务处理视图的切换。

View层通过事件、或者异步线程模式调用控制器的函数，达到把客户响应汇集到控制器。

Model层通过Delegate托管，回调Controller中实现的协议方法把数据汇集到控制器。

2）View层

View负责数据显示与数据输入。数据的输入通过事件交给Model层处理，数据显示通过Model层的托管实现刷新。

3）Model层

Model层服务数据处理，负责向服务器请求数据，并把数据处理好传递给托管。交给View层完成。

#### 2、Model，Controller，View相互通讯的规则

1. Controller可以直接和Model通信
2. Controller也可以直接和View通信
3. Model和View永远不能直接通信
4. View和Controller的通信是透明和固定的，主要通过outlet和action实现
5. View使用Delegate接口和Controller同步信息
6. View不直接和数据通信，使用dataSource接口从Controller处获取数据
7. View的delegate和dataSource一般就是Controller
8. Controller负责为View翻译和格式化Model的数据
9. Model使用Notification & KVO的方式分发数据更新信息，Controller可以有选择的监听自己感兴趣的信息。
10. View也可以监听广播信息，但一般不是Model发出的信息

#### 3、Web Service技术

所有数据请求采用Web Service技术，在后台应用设计一个WebService服务提供，通信的数据格式采用JSON。

第七部分 用户界面设计

### 一、征信查询风格设计

#### 1、征信对比风格

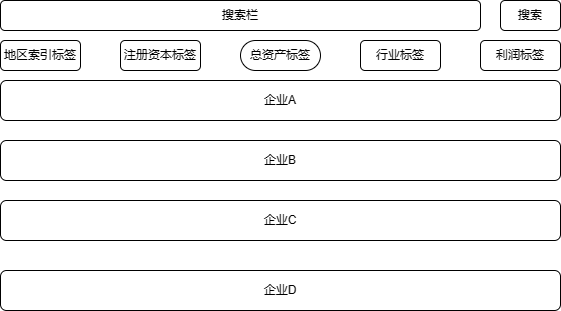


通过雷达图，柱状图，折线图反应企业经营状况及走势，并将相关的不良信息进行公示和对比

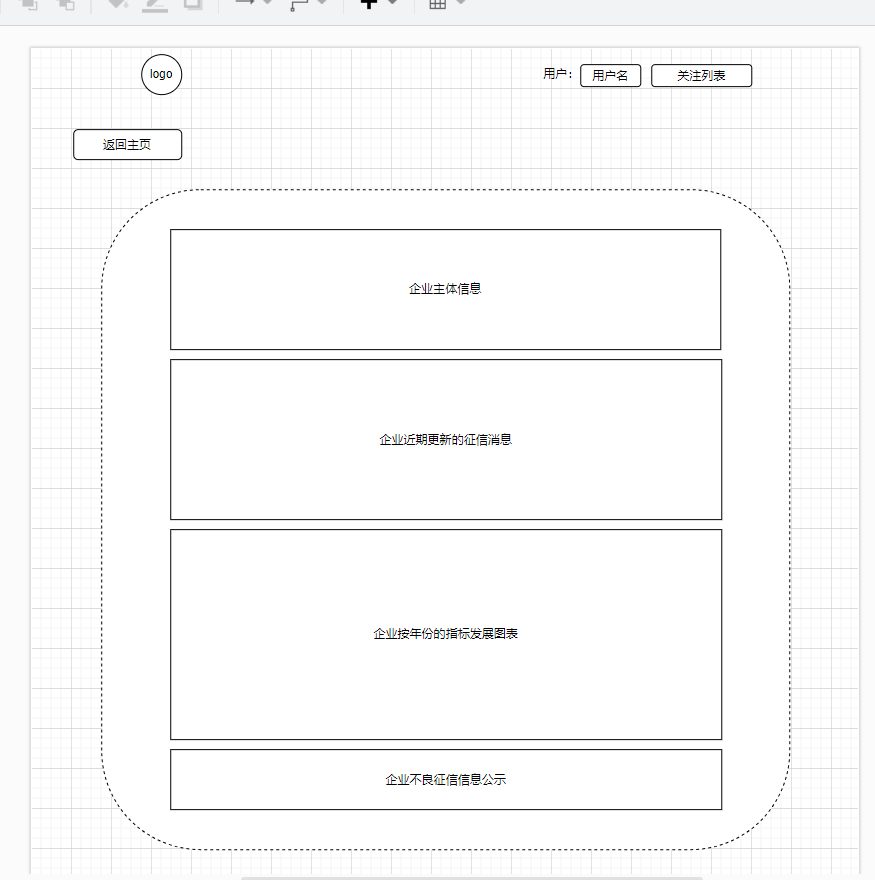
#### 2、查询风格



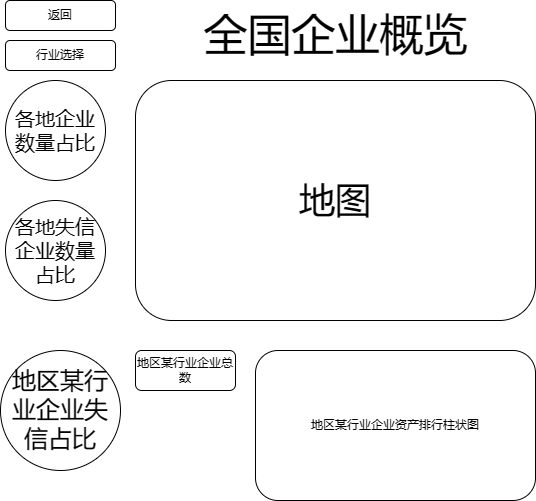
#### 3、查询结果页风格



#### 4、征信详情风格



#### 5、企业概览信息



第八部分 运行环境和部署

### 一、运行环境

#### 1、服务器环境

1.Web服务器：为了托管前端应用和处理HTTP请求，使用Web服务器。

2.应用服务器：部署Spring Boot应用程序，提供后端服务。使用Tomcat容器。

3.数据库服务器：部署Hive以及与之相关的Hadoop集群来存储和处理大数据。确保硬件和配置能够满足大数据存储和处理的需求。

#### 2、客户机器环境

1.操作系统： 用户的客户机器可以使用不同的操作系统，例如Windows、macOS、Linux。

2.浏览器兼容性：前端使用VUE，前端界面在各种主流的Web浏览器中能正常显示和运行，包括Chrome、Firefox、Safari等。

#### **3、开发环境要求**

|  |  |  |
| --- | --- | --- |
| 项目 | 名称 | 版本 |
| 开发平台 | Window | 10 |
| 开发工具 | VSCode+IDEA | — |
| 代码管理工具 | gitee | — |
| 开发环境 | JDK | 8 |

### 二、系统性能要求

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
| # | 项目 | 模块 | 级别 | 技术参数 |
| 1 | 设计实现技术指标 | 系统架构 | A | 采用B/S模式三层架构。 |
| 2 | 面向对象开发语言与框架 | A | 采用java开发语言和springboot, vue等技术。采用e-chart开源图表绘制框架 |
| 3 | 注释和文档 | A | 符合CMMI软件开发过程标准文档（至少提供：需求、概要、详细设计、测试报告、部署和环境、用户手册），代码注释量>=20%。 |
| 5 | 测试覆盖率 | A | 功能覆盖率>=100%，业务覆盖率>=100%，语言覆盖率>=100%，逻辑覆盖率>=80%。 |
| 6 | 资源利用率要求 | CPU占用率 | B | <=50%利用率（附近标准配置） |
| 7 | 内存使用率 | B | <=75%利用率（附近标准配置） |
| 8 | 响应时间要求 | 服务器 | B | <=100ms（附近标准配置） |
| 9 | 网络 | B | <=100ms（附近标准配置） |
| 10 | 客户端 | B | <=5s（附近标准配置） |
| 11 | 系统稳定性要求 | 成熟性 | A | 真实的用户，成功使用本系统 |
| 12 | 稳定性 | B | 无故障运行时间>=365天，系统恢复时间<=2小时。 |
| 13 | 先进性 | A | 采用目前体验最好、最流行的iPhone与iPad移动终端。 |
| 14 | 典型意义 | A | 案例项目要有典型意义，有推广价值。 |
|  |  |  |  |  |
| 说明：级别（A:表示非常重要必须达到的技术性能要求,B:表示重要推荐达到的技术性能要求,C：表示非重要可以弱化的技术性能要求.） | | | | |