题目身份鉴别模块

详细设计说明书

V1.0

拟 制 人 余恩博

审 核 人

批 准 人

2023年 11月 4日

# 引言

## 项目概述

1.1.1项目目标：

该项目旨在实现一个身份鉴别模块，以满足信息系统身份鉴别的需求。通过使用Node.js和MySQL技术，开发一个具有前后端的登录系统，实现安全的身份鉴别功能。项目将采用多因素身份鉴别，结合使用Cookie、散列加密（SHA1）、验证码和时间戳等技术，保障用户身份的安全性。

1.1.2项目范围：

该项目的范围包括设计和开发一个完整的登录系统身份鉴别模块。模块将包括前端和后端组件，前端负责用户界面和交互，后端负责验证用户身份并提供相应的安全措施。模块将与现有的信息系统集成，用于用户身份验证和访问控制。

1.1.3项目重要性：

身份鉴别是保障信息系统安全的重要环节之一。合理而安全地验证用户身份对于防止未经授权的访问、数据泄露和潜在攻击至关重要。这个项目的重要性体现在以下几个方面：

1. 安全性增强：通过采用多因素身份鉴别和散列加密等技术，项目将提供更强大的安全性，减少未经授权的访问风险。
2. 数据保护：通过对数据库中的密码进行散列加密后的非明文存储，用户的敏感信息得到更好的保护，即使数据库泄露也能减少密码泄露的风险。
3. 用户体验：项目将实现验证码和时间戳等技术，增强用户体验，并提供更好的用户界面和交互，使登录过程更便捷和友好。
4. 合规要求：身份鉴别是许多法规和合规要求的基本组成部分，如GDPR（通用数据保护条例），项目的实施有助于满足这些要求。

该项目将为信息系统提供一个安全可靠的身份鉴别模块，保护用户和系统的安全，提高整体的安全性和合规性。通过结合多种安全技术，使得用户身份验证更加可靠和安全，为系统提供可信的访问控制机制。

## 文档概述

详细设计说明书是一个技术文档，旨在详细描述身份鉴别模块的设计和实现细节。它提供了一个全面的视角，包括系统架构、模块设计、数据流程、安全措施和接口等方面，以帮助开发人员理解和实施身份鉴别模块的各个部分。

1.2.1详细设计说明书的结构：

**(1)引言**

文档目的：说明详细设计说明书的目的和范围

项目概述：介绍身份鉴别模块的背景和重要性。

需求概述：概括性地描述身份鉴别模块的需求和功能。

**(2)架构设计**

系统架构：描述身份鉴别模块的整体架构，包括前端和后端组件、模块和它们之间的关系，以及系统的层次结构。

技术选择：介绍选择Node.js和MySQL作为开发技术的原因和优势。

接口设计：详细描述身份鉴别模块的接口和交互方式，包括前端与后端的接口规范和数据传输格式等。

**(3)模块设计**

用户注册模块：详细描述用户注册功能的实现，包括输入验证、数据存储和用户反馈等。

用户登录模块：详细描述用户登录功能的实现，包括身份验证、Cookie管理和错误处理等。

密码重置模块：详细描述密码重置功能的实现，包括验证身份、发送重置链接和更新密码等。

**(4)数据库设计**

4.1 数据库结构：描述数据库的结构和设计，包括用户表、存储密码的字段、其他相关字段等。

4.2 数据库操作：详细描述与数据库的交互方式，包括用户数据的插入、查询和更新等操作。

**(5)安全性设计**

密码安全性：详细描述密码的存储和加密方式，包括散列加密（如SHA1）和非明文存储等。

验证码使用：说明验证码的使用方式，以防止机器人攻击和暴力破解。

时间戳应用：描述时间戳的应用，以防止重放攻击和会话过期。

1.2.2详细设计说明书的用途：

* 提供开发人员理解和实施身份鉴别模块的指南。
* 作为团队成员之间沟通和协作的参考。
* 为后续的维护和扩展提供依据和参考。
* 作为项目审核和验收的基准。
* 提供项目文档化的基础。

# 架构设计

## 系统结构

**2.1.1 结构组件**

**前端组件：**

* 用户界面：包括登录页面和相关表单，用于用户输入身份验证信息。
* 客户端脚本：使用JavaScript和相关的前端框架（如Node.js和Express.js）开发，负责处理用户的输入、发送请求和接收响应。

**后端组件：**

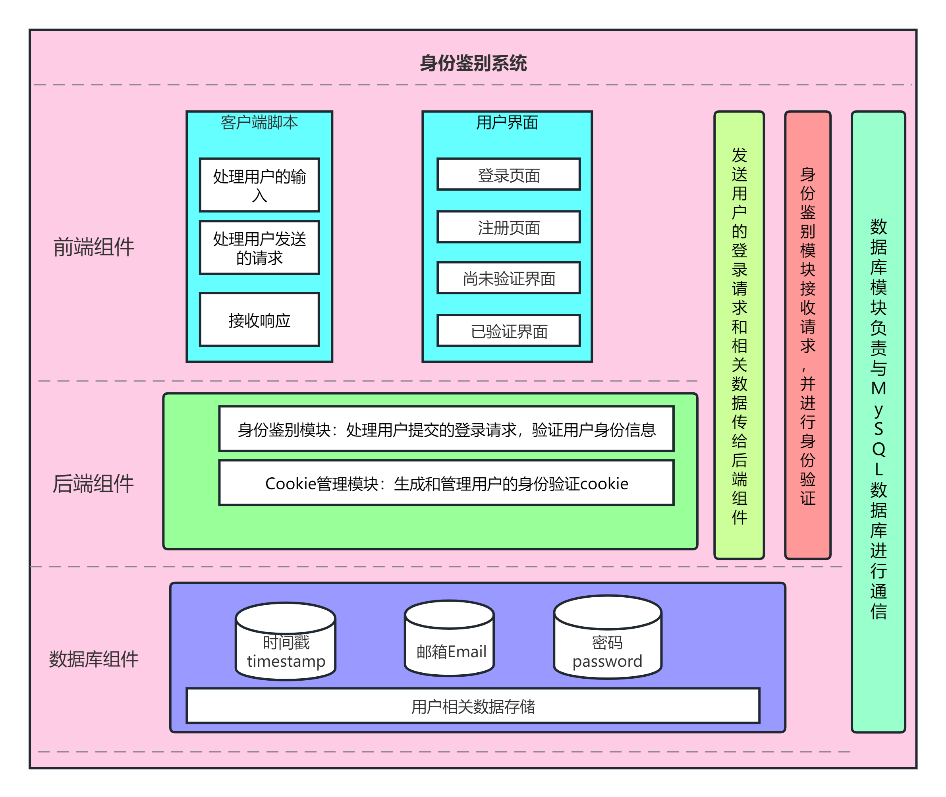
* 身份鉴别模块：处理用户提交的登录请求，验证用户提供的身份信息。
* Cookie管理模块：生成和管理用户的身份验证cookie。
* 数据库模块：包括与MySQL数据库进行通信的功能，用于存储用户的身份信息。

**数据库组件：**

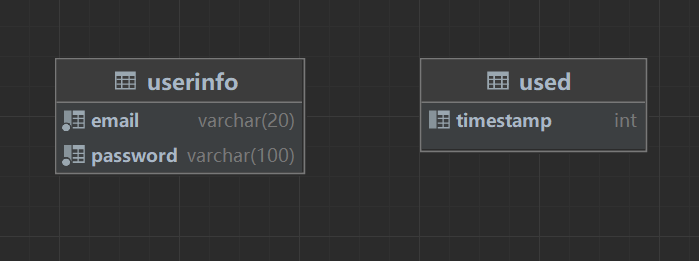
* 用户表：存储用户的身份信息，包括用户名和散列加密后的密码。

**2.1.2 模块之间的关系**

这个总体结构描述了系统的主要组件和它们之间的关系。前端组件负责用户界面和与用户的交互，后端组件负责处理身份鉴别逻辑和与数据库的交互。数据库组件存储用户的身份信息供身份鉴别模块使用。通过这些组件和模块的协作，实现了一个具有前后端的登录系统身份鉴别模块。



## 数据架构



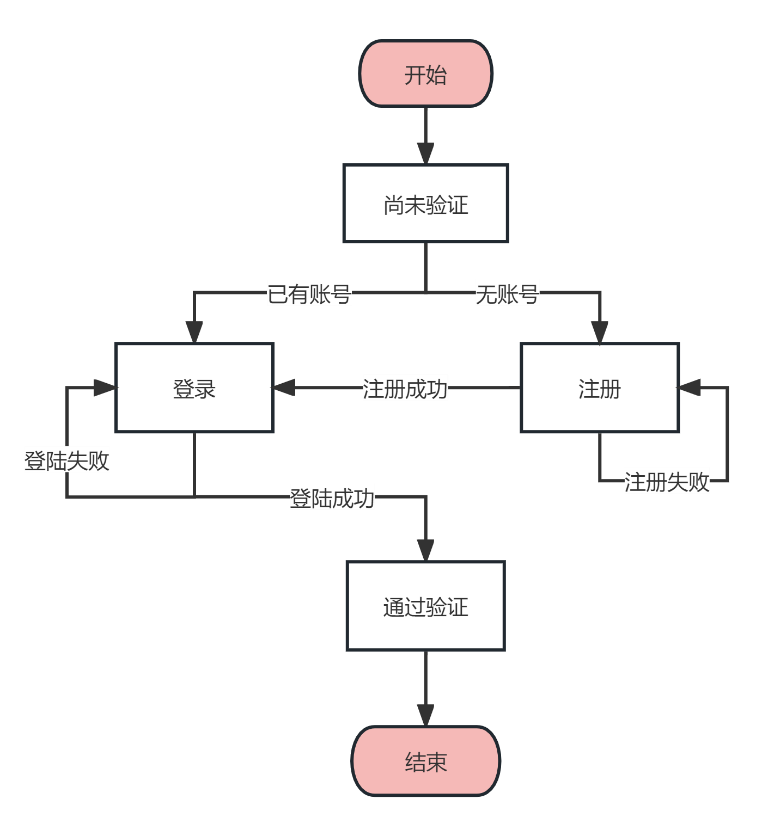
## 技术架构

|  |  |
| --- | --- |
| 所选的技术栈、开发平台和工具 |  |
| 编程语言 | JavaScript、Html |
| 前端框架 | Vue.js、bootstrap |
| 后端框架 | Node.js |
| 数据库 | MySQL |
| 版本控制系统 | Git |
| 测试框架和工具 | Selenium |
| 开发工具平台 | Pycharm professional 2023.2.3 |

# 模块设计

## 模块描述

**3.1.1 流程图**



**3.1.2 模块间关系**

* 用户界面通过前端客户端脚本与后端身份鉴别模块进行交互，发送用户的登录请求和相关数据。后端身份鉴别模块接收请求，并进行身份验证。
* 身份鉴别模块通过与数据库模块的交互，从用户表中获取存储的散列加密密码，与用户提供的密码进行比较。
* 身份鉴别模块在验证成功后，将生成一个加密的cookie，并通过Cookie管理模块进行管理。
* Cookie管理模块负责将加密的cookie发送给用户的浏览器，并管理后续的身份验证和会话管理。
* 数据库模块负责与MySQL数据库进行通信，包括查询用户表以获取用户信息和存储用户的身份信息。

## 接口描述

**用户界面：**

* 入：邮箱（varchar）、密码（varchar）
* 输出：无

**客户端脚本（使用express框架的router路由）：**

* 通信协议：HTTP或HTTPS
* 输入：用户界面输入的邮箱和密码
* 输出：HTTP响应状态码和消息

**份鉴别模块：**

* 通信协议：HTTP或HTTPS
* 输入：邮箱（varchar）、密码（varchar）
* 输出：身份验证结果

**Cookie管理模块：**

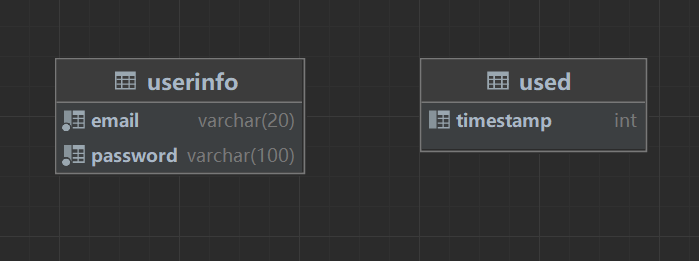
* 通信协议：HTTP（only）
* 输入：身份验证结果
* 输出：HTTP响应中的Cookie信息

**数据库模块：**

* 通信协议： MySQL的客户端/服务器协议
* 输入：身份鉴别模块的输出
* 输出：SHA-1散列加密后的身份验证结果，转化为十六进制字符串

# 数据库设计

## 数据表结构



## 数据库存储过程

存储过程是预先定义好的一组SQL语句的集合，可以通过调用存储过程来执行这些SQL语句。存储过程通常用于封装常用的、复杂的数据库操作，以提高性能和代码复用性。我创建存储过程来处理用户注册、登录等操作。

触发器是与表相关联的数据库对象，它在表上的特定操作（如插入、更新或删除）发生时自动触发执行一组SQL语句。我创建触发器来在登录、注册操作发生时执行相关的数据更新isonline参数，判断用户状态，以便实现页面的权限开放。

# 安全性设计

## 安全策略

（1）强密码策略：要求用户设置复杂的密码，包括大小写字母、数字和特殊字符，并要求定期更换密码。

（2）多因素身份鉴别：实施多因素身份鉴别，结合使用密码、验证码和时间戳等技术，以提供额外的身份验证层级。

（3）定期安全审计：定期对系统进行安全审计和漏洞扫描，以及日志记录和监视，及时响应和处理安全事件。

## 访问控制

（1）身份鉴别与授权：在用户登录时进行身份鉴别，并对用户进行授权，确保只有经过身份验证的用户能够访问系统资源。

（2）最小权限原则：将用户分配给最小权限的角色，即只授予用户必要的权限，以减少潜在的安全风险。

## 数据加密

（1）密码存储：存储用户密码时，采用散列加密算法（如SHA-1）进行单向散列加密，确保密码无法被还原。

（2）传输层加密：使用安全套接层（SSL）或传输层安全性协议（TLS）对用户与服务器之间的通信进行加密，防止数据在传输过程中被窃取或篡改。

（3）Cookie安全：使用SHA-1散列算法对身份验证的cookie进行加密，确保cookie的安全传输和存储。