

本科毕业论文（设计）

概要设计说明书

|  |  |
| --- | --- |
| **学 生 姓 名** | **唐建国** |
| **学号** | **2020131040** |
| **专业** | **区块链工程** |
| **年级班级** | **2020级201班** |
| **指导教师** | **高琳（讲师）** |
| **所在学院** | **区块链产业学院** |
| **提交日期** | **2024年1月6日** |

2024 年 3 月

成都信息工程大学 区块链产业学院

目录

[本科毕业论文（设计） 1](#_Toc8767)

[概要设计说明书 1](#_Toc17080)

[1 引言 - 1 -](#_Toc19476)

[1.1 编写目的 - 1 -](#_Toc21398)

[1.2 背景 - 1 -](#_Toc26235)

[1.3 术语 - 2 -](#_Toc27024)

[1.4 参考资料 - 3 -](#_Toc2681)

[2 总体设计 - 3 -](#_Toc20242)

[2.1 系统体系结构 - 3 -](#_Toc8638)

[2.2 系统总体功能结构 - 4 -](#_Toc17667)

[2.3 运行环境 - 4 -](#_Toc8815)

[2.3.1 硬件环境 - 4 -](#_Toc24707)

[2.3.2 软件环境 - 5 -](#_Toc21355)

[2.4 系统的关键技术 - 7 -](#_Toc8263)

[3 功能模块设计说明 - 8 -](#_Toc14522)

[3.1 功能模块列表 - 8 -](#_Toc22793)

[3.2 功能模块1 - 10 -](#_Toc29122)

[3.2.1 模块编号和功能描述 - 10 -](#_Toc1751)

[3.2.2 操作者 - 11 -](#_Toc32403)

[3.2.3 界面设计与说明 - 11 -](#_Toc27586)

[3.2.4 输入信息 - 12 -](#_Toc6434)

[3.2.5 输出信息 - 13 -](#_Toc14042)

[3.2.6 算法 - 14 -](#_Toc30855)

[3.2.7 对象时序图 - 16 -](#_Toc2901)

[3.3 功能模块2 - 16 -](#_Toc5953)

[3.3.1 模块编号和功能描述 - 16 -](#_Toc27617)

[3.3.2 操作者 - 17 -](#_Toc25230)

[3.3.3 界面设计与说明 - 18 -](#_Toc23490)

[3.3.4 输入信息 - 19 -](#_Toc16277)

[3.3.5 输出信息 - 20 -](#_Toc14454)

[3.3.6 算法 - 22 -](#_Toc1946)

[3.3.7 对象时序图 - 23 -](#_Toc965)

[4 视图设计 - 24 -](#_Toc30651)

[4.1 界面风格设计 - 25 -](#_Toc5681)

[4.2 主界面设计 - 25 -](#_Toc28256)

[5 内部接口设计 - 26 -](#_Toc19603)

[6 系统出错处理设计 - 27 -](#_Toc14821)

[6.1 出错信息 - 27 -](#_Toc26011)

[6.2 补救措施 - 29 -](#_Toc9509)

# 引言

## 编写目的

本份概要设计说明书的编写目的是提供一个清晰的概要设计文档，用于指导面向高校学分认证的区块链应用系统的设计与实现工作。该文档旨在为开发团队、技术架构师、项目经理和其他相关利益相关者提供一个全面了解系统设计的参考。

预期读者包括但不限于以下人员：

- 开发团队成员：包括开发人员、测试人员等。

- 技术架构师：负责系统架构设计和技术方案选择的专业人员。

- 项目经理：负责项目管理和协调的人员。

- 利益相关者：包括业务部门代表、高校管理人员、学生等。

该文档旨在为以上人员提供一个整体了解系统设计的基础，以便更好地理解系统的功能和特性，并为后续的详细设计和开发工作提供指导。

## 背景

本节将描述面向高校学分认证的区块链应用系统的背景，包括以下内容：

1) 需开发的软件系统的名称、英文缩写和项目编号（可选）；

- 软件系统名称：高校学分认证区块链应用系统

- 英文缩写（可选）：UCAS（University Credit Authentication System）

2) 任务提出者和开发者：

- 任务提出者：本系统是由一位论文指导老师提出的项目，其论文指导老师对学生的研究方向和项目提出进行指导和支持。

- 开发者：该系统的开发者为该毕业设计学生自己。学生将负责系统的设计、开发和实施，并在论文指导老师的指导下完成系统的相关工作。

3) 软件系统应用范围和用户：

- 应用范围：该系统旨在解决高校学分认证的问题，使用区块链技术确保学分认证的透明性、安全性和可靠性。

- 用户：主要包括以下几类用户：

- 学生：申请学分认证、查看学分认证状态、管理个人学分记录。

- 高校教务部门：审核学生的学分认证申请、管理学生学分记录。

- 企业或组织：查询学生的学分认证状态和学术成绩，用于招聘和选拔。

- 第三方认证机构：参与学分认证的机构，将认证结果上链。

该系统的目标是提供一个去中心化、透明且不可篡改的学分认证平台，改善传统学分认证过程中的信任问题和操作繁琐性，为学生、高校和用人单位提供更高效、可靠的学分认证服务。同时，该系统也是毕业设计项目的一部分，旨在展示学生在区块链应用系统设计与实现方面的能力和研究成果。

## 术语

1) 区块链（Blockchain）：一种去中心化的分布式账本技术，通过分布式节点之间的共识机制，实现对交易和数据的不可篡改记录和验证。

2) 学分认证（Credit Authentication）：指高校对学生所修读的课程和取得的成绩进行确认和认可的过程。

3) 高校教务部门（University Academic Affairs Office）：负责学校教务管理和学生学籍管理的部门，负责审核和管理学生的学分认证申请和学分记录。

4) 用户界面（User Interface，UI）：系统与用户之间进行交互和信息传递的界面，包括视觉设计和用户操作方式。

5) 智能合约（Smart Contract）：一种以计算机代码形式编写的自动化合约，基于区块链技术实现交易和合约条款的自动执行和验证。

6) 去中心化（Decentralized）：指系统中的数据和决策权分散在多个节点或参与方之间，而非集中在单一机构或服务器上。

7) 透明性（Transparency）：指系统操作和数据的可见性和可追溯性，用户可以查看和验证系统的运行状态和数据记录。

8) 安全性（Security）：指系统的抵御攻击、保护数据和用户隐私的能力，包括数据加密、身份验证和防止篡改等措施。

9) 可靠性（Reliability）：指系统的稳定性和可信度，能够持续提供准确和可用的服务，并对错误和故障有恢复和容错机制。

10) 操作繁琐性（Operational Complexity）：指传统学分认证过程中，需要学生和教务部门进行大量繁复的操作和文件交换，增加了流程的复杂性和工作量。

## 参考资料

本节列出用得着的参考资料，如：

1. 本项目经核准的计划任务书或合同、上级机关的批文；
2. 属于本项目的其他已发表的文件；
3. 本文件中各处引用的文件、资料、包括所要用到的软件开发标准。
4. 行业标准和规范。
5. 列出这些文件资料的标题、文件编号、发表日期和出版单位。

# 总体设计

## 系统体系结构

体系结构决定领域问题结构设计的走向，是系统设计的蓝图。编制并描述系统的体系结构图，并说明系统的运行原理。

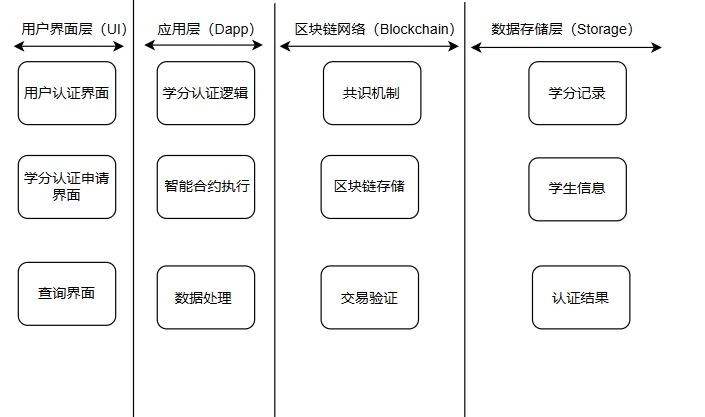


图2-1 系统分层结构

系统的运行原理：

- 学生通过用户界面层提交学分认证申请，并提供相应的证明材料和学分信息。

- 应用层接收学生的申请，对学分信息进行验证和审核，确保合法性和一致性。

- 验证通过的学分认证信息被记录在区块链上，形成一个新的区块，加入到区块链网络中。

- 高校教务部门和企业或组织可以通过用户界面层查询学生的学分认证状态和学术成绩。

- 第三方认证机构参与学分认证的机构，可以将认证结果上链，增加认证的可信度和透明度。

## 系统总体功能结构

|  |
| --- |
| 微信截图_20240107143429 |

图2‑2 系统功能结构图

## 运行环境

### 硬件环境

1. 服务器硬件要求：

- 设备名称：服务器

- 设备型号：根据实际需求选择适当的服务器型号

- 设备数量：至少一台服务器

- 处理器型号：具备足够的计算能力，建议选择多核心处理器

- 内存容量：具备足够的内存容量，以支持区块链节点和应用的运行需求

2. 客户端硬件要求：

- 内存：具备足够的内存容量，以确保良好的用户体验和系统性能

- 处理器：具备足够的处理能力，以支持客户端应用的运行需求

- 硬盘容量：具备足够的硬盘容量，以存储客户端应用和数据

3. 外围设备：

- 外存储设备：根据实际需求选择合适的外部存储设备，包括硬盘、固态硬盘等，需要具备足够的容量来存储系统所需的数据。

- 输入/输出设备：包括键盘、鼠标、显示器等，用于用户与系统进行交互。

- 数据通信设备：根据系统需求选择合适的数据通信设备，如网络交换机、路由器等，以确保系统内部和外部的数据通信。

- 其它专用硬件：根据具体需求，可能需要使用一些专用硬件设备，如加密硬件、安全模块等，以增强系统的安全性和性能。

### 软件环境

1. 操作系统：

- 服务器端：选择适用于服务器的主流操作系统，如：

- Windows Server

- Linux（如Ubuntu、CentOS等）

- 客户端设备：选择适用于客户端设备的操作系统，如：

- Windows 10

- macOS

2. 公链平台：

- 根据您的需求选择合适的公链平台，例如：

- 以太坊（Ethereum）

- EOS

- TRON

- Stellar

- Corda

3. 开发平台及工具：

- 根据您选择的公链平台和开发语言，以下是一些可能使用的开发平台和工具：

- 以太坊：

- Solidity智能合约开发语言

- Truffle开发框架

- Remix集成开发环境（IDE）

- Web3.js或ethers.js JavaScript库

- EOS：

- C++或WebAssembly（WASM）合约开发语言

- EOS Studio集成开发环境（IDE）

- eosjs JavaScript库

4. 通信协议：

- 在与公链进行通信时，可能涉及以下通信协议：

- JSON-RPC：用于与以太坊等公链进行远程过程调用（RPC）。

- HTTP/HTTPS：用于与公链节点进行通信。

- WebSocket：用于实时数据传输和事件通知。

5. 其他软件：

- 在使用公链作为数据库时，可能需要以下其他软件来支持系统功能：

- Web服务器：如Apache、Nginx等，用于承载和提供系统的Web服务。

- 数据加密工具：用于对敏感数据进行加密保护。

- 安全工具：如防火墙、入侵检测系统等，用于增加系统的安全性。

- 数据备份和恢复工具：用于定期备份和恢复数据。

- 系统监控工具：用于监控系统性能和运行状态。

6. 前端开发：

- 前端开发语言和框架：

- HTML/CSS: 用于定义网页的结构和样式。

- JavaScript: 用于实现前端交互和逻辑。

- 前端框架：如React、Angular、Vue.js等，提供了组件化开发、状态管理和路由等功能。

- 前端开发工具：

- 编辑器：如Visual Studio Code、Sublime Text、Atom等，用于编写和编辑前端代码。

- 调试工具：浏览器的开发者工具，如Chrome DevTools，用于调试和分析前端应用程序。

- 包管理工具：如npm（Node Package Manager）或Yarn，用于管理前端依赖库和模块。

- 构建工具：如Webpack、Parcel、Rollup等，用于打包和构建前端应用程序。

7. 前端通信与公链交互：

- 通过公链节点提供的API或SDK与公链进行通信，进行数据的读取和写入。在前端开发中，可以使用以下方式与公链进行交互：

- 使用Web3.js库（用于以太坊）或相应的公链SDK，通过JSON-RPC或HTTP/HTTPS协议与公链节点通信。

- 使用前端HTTP库（如Axios、Fetch等）与后端服务器建立通信，通过服务器代理与公链节点进行交互。

- 使用WebSocket与公链节点建立实时连接，获取区块链事件和通知。

## 系统的关键技术

1. 区块链平台选择：

- 关键技术：选择适合学分认证场景的区块链平台，如以太坊、EOS等。考虑平台的性能、扩展性、安全性和成熟度等因素。

- 测评结果：根据需求和实际情况，评估各个区块链平台的特性和性能，选择最适合的平台进行学分认证。

2. 智能合约开发：

- 关键技术：使用智能合约编程语言，如Solidity（适用于以太坊）或C++（适用于EOS），开发学分认证的智能合约。确保合约的安全性、可靠性和高效性。

- 测评结果：对智能合约进行安全审计和性能测试，确保合约逻辑的正确性和合约执行的高效性。

3. 前端开发：

- 关键技术：使用前端开发语言和框架，如HTML、CSS和JavaScript，以及流行的前端框架（如React、Angular、Vue.js）进行学分认证系统的用户界面开发。

- 测评结果：评估前端开发框架的易用性、性能和扩展性，并进行用户体验测试，确保界面友好、响应迅速。

4. 数据存储和加密：

- 关键技术：使用公链作为数据库存储学分认证相关数据，并进行数据加密保护，确保数据的安全性和隐私性。

- 测评结果：对公链的性能、可扩展性和安全性进行评估，并选择适合的公链平台。同时，采用合适的加密算法和安全措施对敏感数据进行加密和保护。

5. 学分认证算法：

- 关键技术：设计和实现学分认证的算法，确保学分的有效性、真实性和不可篡改性。可能涉及数字签名、哈希算法等技术。

- 测评结果：对学分认证算法进行评估和测试，验证算法的正确性、安全性和抗攻击性。

6. 用户身份认证：

- 关键技术：使用合适的身份认证技术，如数字证书、钱包地址等，确保用户身份的真实性和安全性。

- 测评结果：对身份认证技术进行评估，并验证其对用户身份的有效认证和保护。

# 功能模块设计说明

## 功能模块列表

表3‑1 功能模块列表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **模块编号** | **模块**  **名称** | **功能编号** | **需求功能** | **优先级别** |
| REG\_User01 | 用户注册 | FR1 | 用户注册 | 高 |
| STU\_SubCM02 | 学生提交认证材料 | FR2 | 学生提交认证材料 | 高 |
| STU\_SubAA03 | 学生提交学分认证申请 | FR3 | 学生提交学分认证申请 | 高 |
| AUTH\_Audit04 | 学分认证申请审核 | FR4 | 学分认证申请审核 | 高 |
| TCH\_Audit05 | 教师审核学分认证 | FR5 | 教师审核学分认证 | 高 |
| TCH\_Conf06 | 教师确认学分认证 | FR6 | 教师确认学分认证 | 高 |
| AUTH\_Conf07 | 学分认证结果确认与归档 | FR7 | 学分认证结果确认与归档 | 高 |
| AUTH\_Notify08 | 认证结果通知与归档 | FR8 | 认证结果通知与归档 | 中 |
| BC\_Storage09 | 区块链存储 | FR9 | 区块链存储 | 高 |
| Data\_Protec10 | 数据隐私保护 | FR10 | 数据隐私保护 | 高 |
| Role\_Mgmt11 | 用户角色管理 | FR11 | 用户角色管理 | 中 |
| Auth\_Auth12 | 用户认证和授权 | FR12 | 用户认证和授权 | 高 |
| Sys\_Log13 | 系统日志记录 | FR13 | 系统日志记录 | 中 |
| UI14 | 用户界面 | FR14 | 用户界面 | 中 |
| Config\_Mgmt15 | 系统配置管理 | FR15 | 系统配置管理 | 低 |
| Credit\_Query16 | 学分查询功能 | FR16 | 提供学分查询功能 | 中 |

## 功能模块1

如该功能模块下有子功能模块，请采用层级结构进一步说明。功能模块的描述，可以根据项目实际情况做裁剪和增补。一般需求明确如下内容：

### 模块编号和功能描述

模块编号：REG\_User01

功能描述：用户注册

该功能模块的主要功能是在高校学分认证的区块链应用系统中用户注册。通过使用公链进行注册，可以实现以下功能：

1. 高度去中心化：用户的注册信息将被存储在公链上的多个节点中，实现数据的高度去中心化，提高数据的可靠性和安全性。

2. 数据透明性和不可篡改性：公链上的数据是以区块的形式存储，每个区块包含前一个区块的哈希值，确保数据的透明性和不可篡改性。

3. 用户身份验证和数据隐私保护：使用公链进行注册可以提供强大的用户身份验证机制，同时保护用户的数据隐私，不需要直接暴露用户的详细信息。

具体的用户注册流程可以按照以下步骤进行：

1. 用户打开注册页面或界面，系统展示注册表单。

2. 用户填写必要的信息，如用户名、钱包地址、电子邮件地址、身份（role）等基本信息。

3. 用户提交注册请求，系统将用户注册信息以交易的形式发送到公链网络上。

4. 公链网络中的节点验证该注册交易，并将其包含在一个新的区块中。

5. 注册交易被确认并写入公链后，系统将注册成功的消息返回给用户，提示用户已成功注册。

### 操作者

注册用户（User）：注册用户是指希望加入系统并创建账户的个体。注册用户通过填写必要的信息，如用户名、钱包地址、电子邮件地址、身份（role）等，在系统中创建账户并完成注册过程。

系统管理员（Admin）：系统管理员是具有管理权限的用户，负责管理系统的运行和配置。

系统（System）：指高校学分认证的区块链应用系统。系统负责提供用户注册功能，并处理用户提交的注册请求。系统验证用户注册信息的有效性，处理注册逻辑并将注册数据存储到适当的存储介质中（公链）

### 界面设计与说明

1. 注册页面（Register Page）：

- 页面布局：注册页面应具有清晰的布局，使用户能够轻松找到所需的元素。

- 标题：在页面顶部或中心位置显示注册功能的标题，例如"用户注册"。

- 注册表单：包含必要的输入字段和提交按钮，供用户填写注册信息和提交注册请求。

- 用户名字段：用于输入用户的用户名，可以是文本输入框或单选按钮等形式。

- 电子邮件地址字段：用于输入用户的电子邮件地址，可以是文本输入框或选择器等形式。

- 其他字段：根据系统需求，可以包含其他必填或选填的字段，如姓名、联系方式等。

- 注册按钮：用户填写完所有必填字段后，点击此按钮提交注册请求。

- 错误提示：在表单下方或相应字段旁边显示错误提示信息，以指示用户填写信息时出现的任何错误或缺失。

- 注册成功提示：在注册成功后，显示注册成功的提示信息，例如"注册成功！请前往您的电子邮件进行账户确认。"。

- 登录链接：在页面底部或其他适当位置提供登录链接，以便已注册的用户可以直接登录系统。

### 输入信息

1. 用户名（Username）：

- 标识：唯一标识用户的名称。

- 数据类型和格式：字符串。

- 数据值的有效范围：通常限制长度和字符类型，如字母、数字、特殊字符等。

- 输入的方式：用户手动输入。

- 数量和频度：每个用户只能有一个用户名，注册时需要提供。

- 输入介质：通过注册页面或界面进行输入。

- 输入数据的来源和安全保密条件：用户自行提供，需要确保用户名的唯一性。

2. 电子邮件地址（Email Address）：

- 标识：用于用户与系统进行通信的电子邮件地址。

- 数据类型和格式：字符串。

- 数据值的有效范围：需要符合电子邮件地址的格式规范，如包含@符号和域名等。

- 输入的方式：用户手动输入。

- 数量和频度：每个用户只能有一个电子邮件地址，注册时需要提供。

- 输入介质：通过注册页面或界面进行输入。

- 输入数据的来源和安全保密条件：用户自行提供，需要确保电子邮件地址的有效性和可访问性。

3. 其他个人信息（如姓名、出生日期等）：

- 标识：用户的其他个人信息。

- 数据类型和格式：根据具体要求，可以是字符串、日期等。

- 数据值的有效范围：根据具体要求，如长度、日期范围等。

- 输入的方式：用户手动输入。

- 数量和频度：根据需求，可以有多个个人信息输入。

- 输入介质：通过注册页面或界面进行输入。

- 输入数据的来源和安全保密条件：用户自行提供，需要根据具体情况判断保密条件。

### 输出信息

在用户注册功能中，以下是常见的输出信息及其特性的解释：

1. 注册成功消息（Registration Success Message）：

- 数据类型：字符串。

- 媒体：通常通过系统界面或页面以文本形式呈现给用户。

- 格式：自由文本格式，可以包含文字、标点符号等。

- 数值范围：无。

- 精度：不适用。

- 示例：注册成功！欢迎加入我们的系统。

2. 状态输出（Status Output）：

- 数据类型：字符串或枚举类型。

- 媒体：通常通过系统界面或页面以文本形式呈现给用户。

- 格式：自由文本格式，可以包含文字、标点符号等。

- 数值范围：根据具体情况而定，可能包括不同的状态值。

- 精度：不适用。

- 示例：注册中断，用户名已存在。

3. 异常输出（Exception Output）：

- 数据类型：字符串或异常对象。

- 媒体：通常通过系统界面或页面以文本形式呈现给用户。

- 格式：自由文本格式，可以包含文字、标点符号等。

- 数值范围：无。

- 精度：不适用。

- 示例：注册失败！请稍后再试或联系管理员。

4. 图形或显示报告（Graphical or Display Report）：

- 数据类型：图像、图表、报告文件等。

- 媒体：通常以图像、PDF、HTML等格式呈现给用户。

- 格式：根据具体情况而定，可以是图像格式（如PNG、JPEG）、文档格式（如PDF）或网页格式（如HTML）。

- 数值范围：根据具体情况而定，可以是图像像素范围、文档页数等。

- 精度：根据具体情况而定，可以是图像的分辨率、文档的精确度等。

- 示例：注册成功后生成一份包含用户信息的PDF报告，用户可以下载查看。

### 算法

1. 输入：用户名（username）、电子邮件地址（email）、密码（password）

2. 验证用户名和电子邮件地址是否符合要求：

- 检查用户名是否唯一且符合长度和字符类型的要求。

- 检查电子邮件地址是否符合格式要求。

3. 生成加密的密码哈希值：

- 将密码进行哈希算法处理，如SHA-256，生成密码的哈希值。

4. 创建用户对象：

- 将用户名、电子邮件地址和密码哈希值存储在用户对象中。

5. 将用户对象存储到区块链网络：

- 将用户对象以交易的形式发送到区块链网络上。

6. 等待区块链网络的确认：

- 等待区块链网络中的节点验证和确认注册交易。

7. 输出注册结果：

- 如果注册交易成功被确认，则输出注册成功的消息。

- 如果注册交易被拒绝或出现错误，则输出注册失败的消息。

### 对象时序图

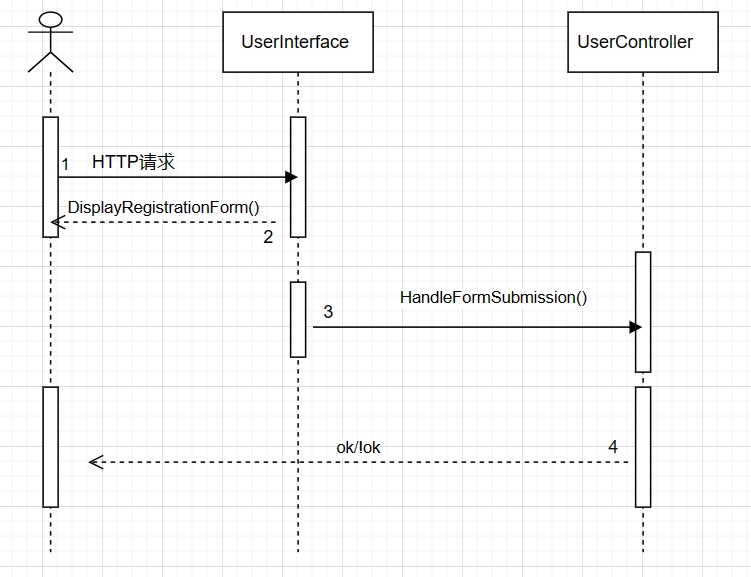


图3-2 注册的对象时序图

## 功能模块2

### 模块编号和功能描述

功能描述：学分认证

学分认证申请是学生申请认证其学分的过程，通过使用智能合约技术，可以增强学分认证的透明性、安全性和自动化程度。以下是使用智能合约的学分认证申请功能的描述：

1. 学生提交认证材料：

- 学生使用学分认证系统上传和提交认证材料，例如学习成绩单、课程证书等。

- 认证材料将被存储在区块链上，确保数据的不可篡改性和安全性。

2. 学分认证申请创建：

- 学生在学分认证系统中创建学分认证申请，填写相关信息，如个人信息、所需认证的学分信息等。

- 学分认证申请将被转化为智能合约的形式，并发布到区块链上。

3. 学分认证申请审核：

- 学分认证申请被提交到智能合约中，审核员可以访问合约进行审核。

- 审核员根据学校的认证规则和标准，检查学生提交的材料和申请的准确性和完整性。

4. 智能合约的自动化审核：

- 智能合约可以自动执行一系列的审核规则和逻辑，以验证学生的学分认证申请。

- 合约可以检查学生的学分是否满足要求，验证提交的材料是否真实有效等。

5. 学生和审核员的交互：

- 如果智能合约需要进一步核实或要求学生补充材料，学生和审核员可以通过学分认证系统进行交互，提供额外的信息或补充材料。

6. 学分认证结果确认：

- 智能合约会根据审核结果，将学分认证申请标记为通过或不通过，并记录在区块链上。

- 认证结果将通知给学生，学生可以在学分认证系统中查询和查看自己的学分认证结果。

### 操作者

以下是学分认证过程中可能涉及的操作者：

1. 学生：

- 学生是申请学分认证的主体，他们需要提交认证材料、填写学分认证申请，并与审核员和教师进行交互。

- 学生可以通过学分认证系统上传材料、填写申请表，查询认证结果等。

2. 教师：

- 教师在学分认证过程中扮演重要角色，他们审核学生的学分认证申请，并确认学分认证结果。

- 教师会仔细检查学生的学习情况和所提交的认证材料，确保符合学校的认证要求。

3. 学分认证系统管理员：

- 学分认证系统管理员负责学分认证系统的管理和维护。

- 他们负责确保系统的正常运行，处理系统的升级和维护工作，管理用户权限和数据安全等。

### 界面设计与说明

1. 学生认证申请界面：

- 学生登录后，进入学生认证申请界面。

- 学生可以填写个人信息、选择所需认证的学分、上传认证材料等。

- 提供提交按钮，学生可以点击提交按钮将申请信息发送到智能合约进行审核。

2. 认证申请列表界面：

- 教师登录后，进入认证申请列表界面。

- 教师可以查看待审核的学生认证申请列表。

- 提供搜索和筛选功能，以便教师可以根据学生姓名或其他条件查找特定的申请。

3. 学生认证申请详情界面：

- 教师在认证申请列表中选择一个申请后，进入学生认证申请详情界面。

- 教师可以查看学生的个人信息、所需认证的学分、已上传的认证材料等。

- 提供审核意见的输入框，教师可以在此处填写审核意见和认证结果。

4. 学生认证结果界面：

- 学生登录后，可以访问学生认证结果界面。

- 学生可以查看自己的学分认证结果，包括通过或不通过的认证申请。

- 提供详细的认证结果信息，包括审核意见和教师的备注。

5. 系统管理界面：

- 学分认证系统管理员登录后，可以访问系统管理界面。

- 管理员可以管理用户账户、权限设置和系统配置。

- 提供对系统进行维护和升级的功能。

### 输入信息

1. 姓名（Name）：

- 标识：Name

- 数据类型和格式：字符串（字符序列），通常是文本格式。

- 数据值的有效范围：通常要求输入真实有效的姓名，可以限制字符长度或使用正则表达式进行格式验证。

- 输入方式：通过文本框或输入字段输入。

- 数量和频度：通常是单值输入，每个用户只需要输入一次。

- 输入介质：通过键盘或其他文本输入设备输入。

- 输入数据的来源和安全保密条件：用户提供自己的姓名作为个人身份信息，应保证数据的准确性和隐私安全。

2. 学号（Student ID）：

- 标识：StudentID

- 数据类型和格式：字符串或数字，通常是学校分配的唯一标识符。

- 数据值的有效范围：根据学校的规定，可能有特定的长度要求或格式验证。

- 输入方式：通过文本框或输入字段输入。

- 数量和频度：通常是单值输入，每个学生只有一个学号。

- 输入介质：通过键盘或其他文本输入设备输入。

- 输入数据的来源和安全保密条件：学校向学生提供学号，应保证数据的准确性和隐私安全。

3. 学分课程（Course）：

- 标识：Course

- 数据类型和格式：字符串，表示课程的名称或代码。

- 数据值的有效范围：根据学校的课程列表或代码表限定有效的课程名称或代码。

- 输入方式：通过下拉菜单、选择框或文本输入框输入。

- 数量和频度：通常是多值输入，学生可以选择申请认证的多个课程。

- 输入介质：通过鼠标、键盘或触摸屏进行选择或输入。

- 输入数据的来源和安全保密条件：学校提供课程列表或代码表，用于学生选择合适的课程。

4. 上传认证材料（Upload Documents）：

- 标识：Documents

- 数据类型和格式：文件，可以是学习成绩单、证书、项目报告等。

- 数据值的有效范围：根据系统要求，可能限制文件类型（如PDF、JPEG等）和文件大小。

- 输入方式：通过文件选择对话框或拖放文件进行上传。

- 数量和频度：通常是多值输入，学生可以上传多个认证材料。

- 输入介质：通过文件选择对话框或拖放文件进行上传。

- 输入数据的来源和安全保密条件：学生准备好认证材料后，从本地计算机或其他存储设备上传到系统，应确保数据的完整性和安全性。

### 输出信息

在1. 认证结果（Certification Result）：

- 数据类型：字符串

- 媒体：文本或页面显示

- 格式：通常是人类可读的文本格式，例如“通过”或“不通过”等。

- 数值范围：结果的取值范围根据认证规则确定，通常是有限的预定义结果。

- 精度：结果表示是否达到认证要求，不涉及数值精度。

- 示例：认证结果为“通过”。

2. 审核意见（Review Comments）：

- 数据类型：字符串

- 媒体：文本或页面显示

- 格式：通常是人类可读的文本格式，用于教师提供对学生认证申请的审核意见和建议。

- 数值范围：根据教师的反馈和建议，可以是任意文本内容。

- 精度：不涉及数值精度，可以是自由文本。

- 示例：审核意见为“认证材料不完整，请提供成绩单的扫描件”。

3. 报告生成（Report Generation）：

- 数据类型：文件

- 媒体：电子文档（如PDF）或其他电子格式

- 格式：根据系统设计要求和用户需求，可以是预定义的模板或自定义格式。

- 数值范围：根据报告内容和格式的要求，可能包括表格、图表、文字描述等。

- 精度：根据报告内容和格式的要求，可能涉及数值精度、小数位数等。

- 示例：生成学生学分认证报告，包括学生个人信息、认证课程及成绩等详细内容。

4. 异常信息（Error Messages）：

- 数据类型：字符串

- 媒体：文本或页面显示

- 格式：通常是人类可读的文本格式，用于表示系统执行过程中的错误或异常情况。

- 数值范围：根据系统设计和错误类型，可能有不同的错误代码或错误描述。

- 精度：不涉及数值精度，用于指示错误类型和原因。

- 示例：显示错误信息“无法连接到数据库，请稍后再试”。

5. 图形或显示报告（Graphical or Display Reports）：

- 数据类型：图像或可视化数据

- 媒体：屏幕显示或打印输出

- 格式：可以是图表、图形、统计数据等形式，以提供可视化的认证结果或统计信息。

- 数值范围：根据报告内容和图形的要求，可能涉及不同的数值范围和数据精度。

- 精度：根据报告内容和图形的要求，可能涉及数值精度、坐标轴刻度等。

- 示例：显示学生学分认证统计图表，包括通过和不通过的认证数量。

### 算法

在学分认证模块中，可能涉及一些算法和逻辑关系来进行计算、判断和处理数据。以下是一些可能存在的算法：

1. 计算学分总和：

- 计算公式：将学生通过的课程的学分进行累加。

- 说明：系统根据学生所通过的课程及其对应的学分，对学分进行累加，得出学生的学分总和。

2. 判断课程是否符合认证要求：

- 逻辑关系：系统根据预设的认证规则和要求，对学生选择的课程进行判断。

- 说明：系统比对学生所选择的课程和相应的认证规则，判断课程是否满足认证要求，例如学分数、课程类型等。

3. 计算学生的平均学分绩点（GPA）：

- 计算公式：将每门课程的学分乘以对应的绩点，所有课程的学分绩点之和除以总学分数。

- 说明：系统根据每门课程的学分和绩点，计算学生的平均学分绩点，用于评估学生的学业成绩。

4. 身份验证：

- 逻辑关系：系统通过验证用户提供的身份信息与存储的身份信息进行比对。

- 说明：系统根据用户输入的学号和姓名等身份信息，与存储的学生信息进行比对，以验证用户的身份是否正确。

5. 申请审核流程：

- 逻辑关系：系统根据预设的审核流程和状态，对学生的认证申请进行处理和转发。

- 说明：系统根据当前的申请状态和操作者的权限，将学生的认证申请转发给相应的审核人员，按照预设的审核流程进行处理。

6. 数据查询：

- 逻辑关系：系统根据用户提供的查询条件，从数据库中检索相应的数据。

- 说明：系统根据用户提供的查询条件（如学生姓名、学号、课程名称等），执行相应的查询操作，返回符合条件的数据。

### 对象时序图

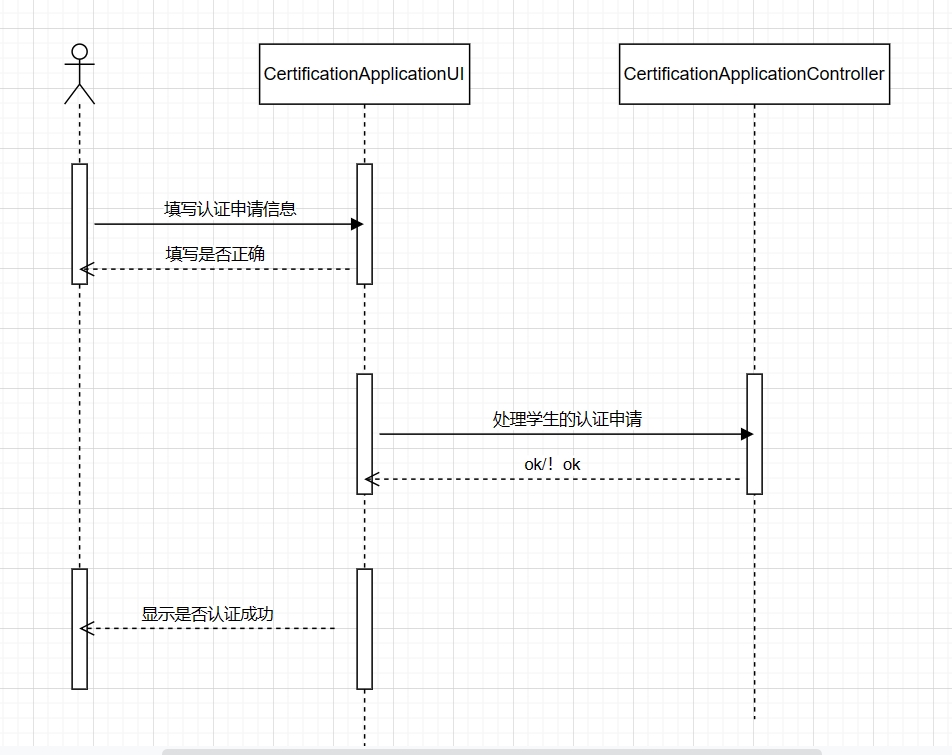


图3-3 学分认证的对象时序图

# 视图设计

以下是一个按操作流程组合人机操作界面的示例视图设计：

1. 登录界面：

- 用途：用于学生、审核人员等操作者进行身份验证和登录系统。

- 包含的元素：用户名和密码输入框、登录按钮、忘记密码链接等。

2. 学生认证申请界面：

- 用途：用于学生提交学分认证申请。

- 包含的元素：学生信息展示区域（如姓名、学号等）、课程信息输入区域、提交按钮等。

3. 审核界面：

- 用途：用于审核人员查看和处理学生的认证申请。

- 包含的元素：待审核申请列表、认证申请详情展示区域、审核意见输入框、通过和拒绝按钮等。

4. 认证结果界面：

- 用途：用于向学生展示学分认证结果。

- 包含的元素：认证结果展示区域（通过或拒绝）、认证详情展示区域、返回按钮等。

5. 查询界面：

- 用途：用于学生和其他操作者查询学分认证相关信息。

- 包含的元素：查询条件输入区域（如学号、姓名等）、查询结果展示区域、查询按钮等。

在视图设计中，需要考虑以下方面：

- 用户友好性：界面布局清晰、易于理解和操作。

- 一致性：保持界面元素、样式和交互的一致性，提供统一的用户体验。

- 可访问性：确保界面对于不同能力和使用环境的用户都易于访问和使用。

- 响应式设计：使界面能够适应不同设备和屏幕尺寸，提供良好的用户体验。

- 安全性：采取适当的安全措施，保护用户数据和系统的安全。

## 界面风格设计

选择现代、扁平化或材料设计风格作为界面风格设计的基础，根据系统的特点和要求进行个性化的调整。对于学分认证应用系统，简洁、直观和易用的界面风格能够提升用户体验，帮助用户快速完成认证申请和查询操作。

## 主界面设计

1. 导航菜单：主界面应包含清晰的导航菜单，用于导航到系统的主要功能模块。导航菜单可以使用水平或垂直布局，以确保用户可以轻松访问和切换不同的功能页面。

2. 信息展示区域：主界面应提供一个信息展示区域，用于呈现与学分认证相关的重要信息，如学生的认证状态、已认证的学分等。这样的信息展示可以帮助用户快速了解他们的学分认证情况。

3. 功能模块入口：主界面应提供快速访问各个功能模块的入口，如学生认证申请、审核、查询等。这些入口可以以按钮、图标或卡片的形式呈现，以便用户能够直观地找到所需的功能。

4. 通知和提醒：主界面可以包含通知和提醒功能，用于向用户发送重要的系统消息或提醒。这些通知可以以气泡、弹出窗口或标志等形式呈现，以便用户及时了解系统的最新动态。

5. 用户个人信息：主界面可以提供一个区域显示用户的个人信息，如头像、姓名、学号等。这样的设计可以增强用户的归属感和个性化体验。

6. 快速操作：主界面可以包含一些常用功能的快速操作按钮或快捷方式，以便用户能够快速执行常见的操作，如提交认证申请、查看认证结果等。

7. 响应式设计：确保主界面的布局和元素在不同设备和屏幕尺寸上能够自适应，以提供良好的用户体验。

# 内部接口设计

本系统内的各功能模块之间的接口。对每个模块提供的接口进行说明，需说明接口的使用者/调用者、接口的目的、内容、数据格式、读写方式、约束等。

表4‑1 构件接口列表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **模块名称** | **接口编号** | **接口名称** | **接口类型** | **说明** |
| 用户注册模块 | 001 | 注册用户 | 外部 | 用于用户注册，接收用户提供的注册信息，包括用户名、密码等 |
| 学分认证模块 | 002 | 提交认证材料 | 内部 | 学生向认证模块提交认证材料，包括身份证扫描件、学分证书等 |
| 学分认证模块 | 003 | 提交学分认证申请 | 内部 | 学生向认证模块提交学分认证申请，包括学分信息和申请内容 |
| 学分认证审核模块 | 004 | 审核学分认证申请 | 内部 | 对学分认证申请进行审核 |
| 教师审核模块 | 005 | 教师审核学分认证 | 内部 | 教师对学生的学分认证进行审核 |
| 区块链存储模块 | 006 | 存储认证信息 | 内部 | 将认证信息存储到区块链中 |
| 用户角色管理模块 | 007 | 角色授权 | 内部 | 对用户进行角色授权 |
| 学分查询模块 | 008 | 查询学分 | 外部 | 提供学生查询学分的功能 |

# 系统出错处理设计

## 出错信息

用一览表的方式说明每种可能的出错或故障情况出现时，系统输出信息的形式、含意及处理方法。

|  |  |  |
| --- | --- | --- |
| 序号 | 出错或故障情况 | 系统输出信息及处理 |
| 1 | 无法连接到区块链网络 | - 输出信息：无法连接到区块链网络，请检查网络连接或配置设置。  - 处理方法：系统应该进行网络连接和配置的检查，确保网络连接正常，并且区块链节点的配置正确。如果问题持续存在，系统可以提供建议如检查防火墙设置、重启区块链节点或联系技术支持团队。 |
| 2 | 区块链交易被拒绝或失败 | - 输出信息：区块链交易失败，请再次尝试或联系技术支持。  - 处理方法：系统应该记录失败的交易信息，并提供给用户建议如检查交易数据的正确性、确认账户余额是否足够等。如果问题持续存在，用户可以联系技术支持团队获取进一步帮助。 |
| 3 | 数据完整性验证失败 | - 输出信息：数据完整性验证失败，请联系技术支持。  - 处理方法：系统应该记录验证失败的数据信息，并进行详细的错误分析。用户应联系技术支持团队，提供相关信息以协助故障排查和修复。 |
| 4 | 用户身份验证失败 | - 输出信息：身份验证失败，请确认提供的凭据是否正确。  - 处理方法：系统应向用户提供准确的身份验证错误信息，指导用户检查提供的凭据是否正确、是否过期或与注册信息是否匹配。 |
| 5 | 区块链网络超时 | - 输出信息：网络连接超时，请稍后重试或联系技术支持。  - 处理方法：系统应记录超时错误，并提供建议如检查网络连接的稳定性、尝试使用其他网络连接等。如果问题持续存在，用户可以联系技术支持团队获取帮助。 |
| 6 | 无法访问用户数据 | - 输出信息：无法访问用户数据，请联系技术支持。  - 处理方法：系统应记录访问错误，并进行错误分析。用户应联系技术支持团队，提供相关信息以协助故障排查和修复。 |
| 7 | 无效的学分认证请求 | - 输出信息：无效的学分认证请求，请提供有效的请求信息。  - 处理方法：系统应向用户提供详细的错误信息，指导用户提供正确的学分认证请求信息，确保请求格式正确且包含必要的字段。 |
| 8 | 无法找到学生学分信息 | - 输出信息：找不到学生的学分信息，请确认学生身份信息是否正确。  - 处理方法：系统应向用户提供错误信息，指导用户确认学生身份信息的正确性，并提供适当的错误解决方法。如问题仍存在，用户可以联系技术支持团队获取帮助。 |
| 9 | 系统内部错误 | - 输出信息：系统内部发生错误，请联系技术支持。  - 处理方法：系统应记录内部错误并进行错误分析。用户应联系技术支持团队，提供相关信息以协助故障排查和修复。 |
| 10 | 服务不可用 | - 输出信息：系统当前不可用，请稍后再试或参考维护通知。  - 处理方法：系统应提供维护通知，包括服务不可用的原因和预计恢复时间。用户可以稍后再试或根据维护通知中提供的信息获取相关帮助。 |

## 补救措施

系统通过不同的界面提示信息，提示用户更正操作，让后台屏蔽异常或抛出异常状态，使得管理员在调试的过程中发现问题。

1. 通过界面提示信息：系统可以在用户界面上以弹窗、警告框等形式显示出错信息，以便用户能够立即注意到问题。这些提示信息应该清晰明了，指导用户更正操作或提供进一步的解决方案。

2. 让后台屏蔽异常或抛出异常状态：系统在后台可以检测到异常情况，并屏蔽异常或抛出异常状态。通过捕获异常，系统可以记录异常信息，例如错误类型、发生时间、相关数据等。这些信息对于管理员在调试和故障排查过程中非常有用。

3. 提供管理员调试信息：系统可以在后台记录详细的调试信息，包括错误日志、堆栈跟踪等。管理员可以使用这些信息来分析问题的根源，并进行适当的调试和修复操作。这些调试信息可以通过系统的管理界面或专门的日志查看工具进行访问。

4. 提供错误报告和日志：系统可以生成错误报告和日志文件，其中包含了出错情况的详细信息。这些报告和日志可以提供给管理员或技术支持团队，以便他们进行故障排查和修复。管理员可以使用这些报告和日志来追踪问题，并采取相应的补救措施。

5. 实时监控和警报：系统可以设置实时监控机制，对关键组件和服务进行监控，并在出现异常情况时发送警报通知给管理员或技术支持团队。这样可以及时发现问题并采取相应的补救措施，以最小化系统的故障时间和影响范围。