成都信息工程大學
Chengdu University of Information Technology

本科毕业论文（设计）

概要设计说明书

学生姓名	翁忠旭
学号	2021131028
专业	区块链工程
年级班级	2021级1班
指导教师	梁培利（讲师）
所在学院	人工智能学院（区块链产业学院）
提交日期	2025年3月30日

2025 年 3 月

成都信息工程大学 人工智能学院（区块链产业学院）

目录

1 引言

1.1 编写目的

本概要设计说明书旨在详细描述基于链上随机数和时间锁的彩票系统的设计方案，包括系统架构、功能模块、接口设计等内容。本文档主要面向开发团队、测试人员、项目经理以及系统维护人员，为系统实现提供技术指导和参考依据。

1.2 背景

1) 本系统名称为"基于链上随机数和时间锁的彩票系统"（Blockchain-based Lottery System with On-chain Randomness and Timelock，简称 BLSRT）。
2) 任务提出者：成都信息工程大学区块链产业学院；开发者：区块链工程专业翁忠旭。
3) 本系统应用于数字货币市场中的彩票玩法，为用户提供公平、透明、无需中介的彩票服务，同时适用于需要结果透明的社区应用场景。

1.3 术语

列表 1-1 术语和缩略语

术语、缩略语	解　　释
链上随机数	通过链上数据生成的随机值，能够确保公平性和不可预测性；常规方案包括 VRF。
时间锁	使用预定时间后才能解锁的机制，用于提高系统安全性，防止人举操作。
drand	分布式随机数生成网络，提供可验证的随机性
智能合约	部署在区块链上的程序，用于自动执行合同条款。

1.4 参考资料

[1] Ethereum Foundation. (n.d.). Ethereum Developer Documentation. Retrieved from https://ethereum.org/en/developers/docs/
[2] Hardhat Team. (n.d.). Hardhat Documentation. Retrieved from https://hardhat.org/docs
[3] Drand Team. (n.d.). Drand Documentation. Retrieved from https://drand.love/docs/
[4] Solidity Documentation. (2024). Solidity: Smart Contract-Oriented Programming. Retrieved from https://docs.soliditylang.org/

2 总体设计

2.1 系统体系结构

本系统采用前后端分离的架构，后端基于区块链智能合约实现核心业务逻辑，前端通过 Web3 接口与区块链交互。

本系统体系结构，见图 2-1。

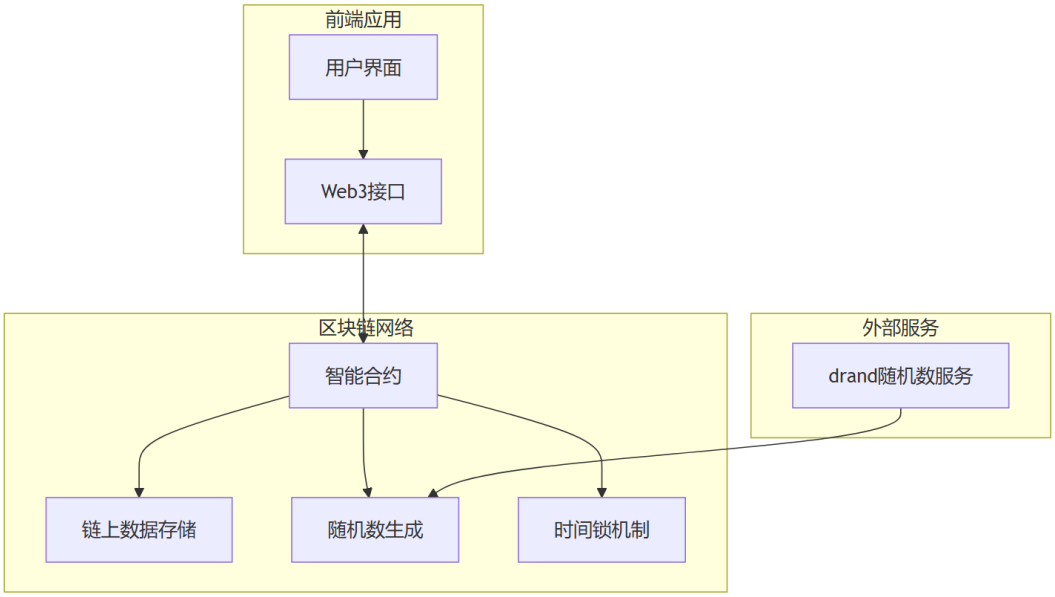

图 2-1 系统体系结构

系统运行原理：用户通过前端界面与区块链上的智能合约交互，购买彩票、查询记录等操作均通过智能合约执行。智能合约负责管理彩票发行、购买记录、开奖和派奖等核心功能。系统利用链上随机数生成机制确保开奖公平性，通过时间锁机制确保开奖时间的透明性和不可操控性。

2.2 系统总体功能结构

系统总体功能结构图见图 2-2。

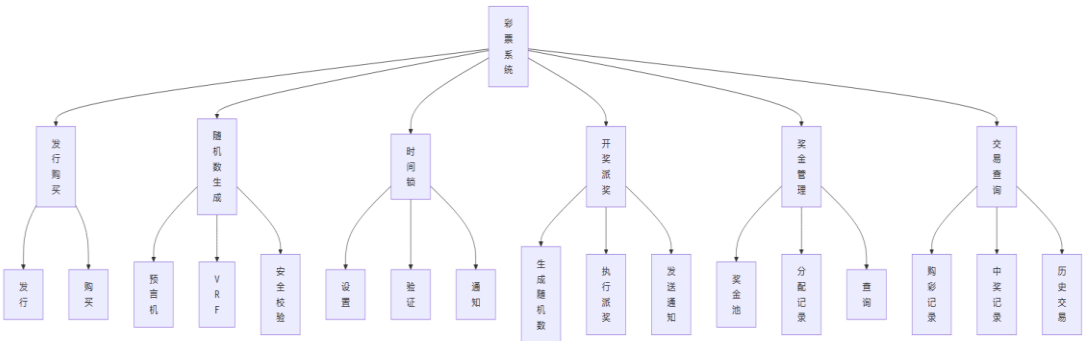

图 2-1 系统总体功能结构图

2.3 运行环境

系统的网络拓扑结构如下：

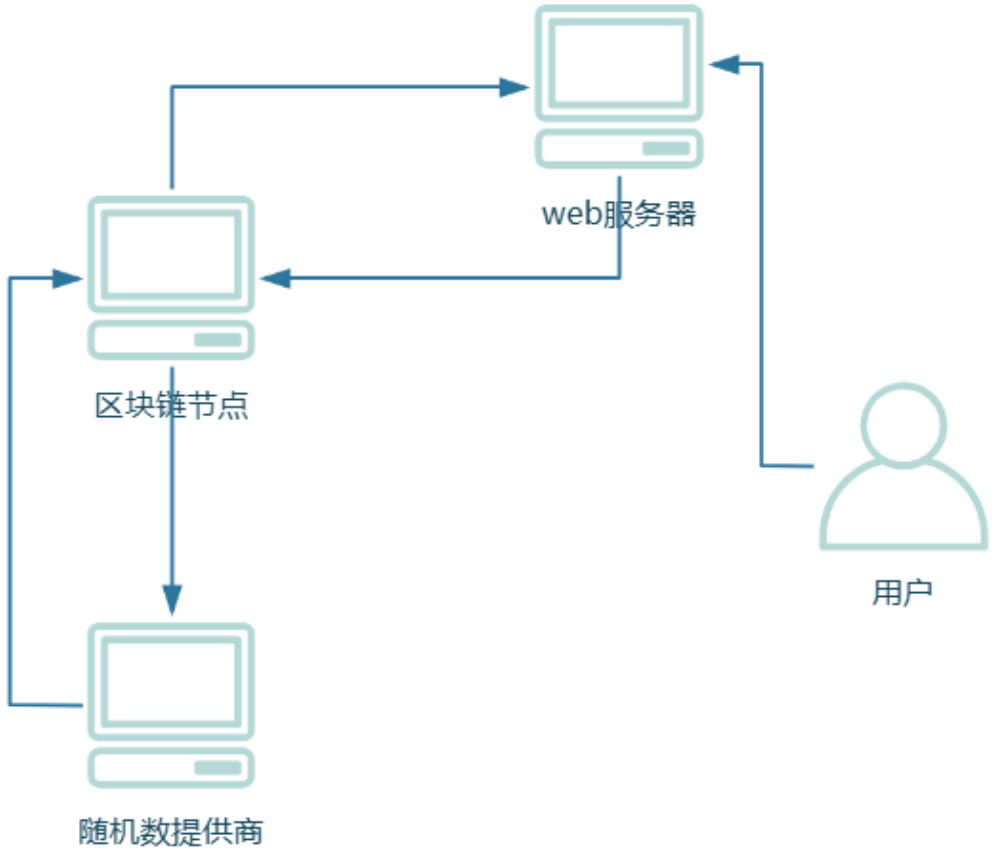

图 3-2 网络拓扑图

本系统提供本地和远程两种访问方式，用户可通过浏览器和钱包插件（如 MetaMask）访问系统。

Web 服务器：

- IP：59.73.87.106
- 网关：59.73.87.254
- 端口：8080
- 域名：http://www.***。

区块链节点：

- 网络：以太坊测试网络（Goerli/Sepolia）或兼容 EVM 的其他网络
- 节点类型：全节点
- 数量：至少 3 个节点以确保系统稳定性

2.3.1 硬件环境

1. 描述本软件运行对服务器、客户端的硬件要求：

服务器：

- 设备类型：云服务器
- 处理器：至少 8 核 CPU
- 内存：至少 16GB RAM
- 存储：至少 500GB SSD
- 数量：至少 1 台

客户端：

- 处理器：双核及以上
- 内存：4GB 及以上
- 硬盘：50GB 及以上
- 网络：支持稳定的互联网连接

2.3.2 软件环境

1. 操作系统：
   1) 服务器：Ubuntu 20.04 LTS 或更高版本
   2) 客户端：支持现代浏览器的任何操作系统
2. 区块链环境：以太坊兼容网络
3. 智能合约语言：Solidity 0.8.24
4. 开发平台及工具：
   1) 前端：React.js、Web3.js、Material-UI
   2) 智能合约开发：Hardhat、Remix
   3) 测试工具：Chai、Mocha
   4) 通信协议：HTTP/HTTPS，WebSocket，JSON-RPC

5.其他软件：

   1) 版本控制：Git
   2) CI/CD：GitHub Actions
   3) 文档工具：Markdown。

2.4 系统的关键技术

1. 区块链智能合约技术：

使用 Solidity 语言开发智能合约，实现彩票系统的核心业务逻辑

测评结果：Solidity 0.8.x 版本具有良好的安全特性，能够有效防止整数溢出等常见漏洞

2. 链上随机数生成技术：

采用 drand 分布式随机数生成服务，利用分布式随机网络共识机制确保随机数的公平性和不可预测性

测评结果：drand 服务提供的随机数具有高熵值和可验证性，满足系统对随机性的要求

时间锁机制：

利用区块链的时间戳和智能合约的时间锁功能，确保开奖过程的透明性

测评结果：基于区块时间戳的时间锁机制能够有效防止提前开奖，保证系统公平性

3. Web3 交互技术：

使用 Web3.js 库实现前端与区块链的交互

测评结果：Web3.js 提供了完善的 API，能够满足系统与区块链交互的需求

4. React 前端框架：

采用 React.js 构建用户界面，提供良好的用户体验

测评结果：React 的组件化开发模式和虚拟 DOM 技术能够提高前端性能和开发效率

3 功能模块设计说明

3.1 功能模块列表

表 3-1 功能模块列表

模块编号	模块名称	对应需求功能编号	所对应需求功能	实现优先级
DS_BLSRT01	彩票发行	SRS_SYS1.01	彩票发行	高
DS_BLSRT02	开奖	SRS_SYS2.01, SRS_SYS2.02	开奖随机数生成，派奖执行	高
DS_BLSRT03	奖金管理	SRS_SYS3.01, SRS_SYS3.02	奖金池管理，奖金分配记录	高
DS_BLSRT04	交易记录查询	SRS_SYS4.01, SRS_SYS4.02	购彩记录查询，中奖记录查询	高

3.2 彩票发行模块

3.2.1 模块编号和功能描述

- 模块编号：DS_BLSRT01
- 功能描述：管理员通过系统发布新一期的彩票，设置彩票的基本信息（编号、价格、开奖时间等），并将彩票信息记录到区块链上。

3.2.2 操作者

- 系统管理员

3.2.3 与本模块相关的码表和表

表 3-2 模块功能表

名称	中文注释	类型	作用
Lottery	彩票信息表	结构体	存储彩票的基本信息
LotteryStatus	彩票状态码表	枚举	定义彩票的不同状态

3.2.4 界面设计与说明

管理员彩票发行界面包含以下元素：

- 彩票编号输入框
- 彩票价格输入框
- 开始时间选择器
- 结束时间选择器
- 开奖时间选择器
- 奖项结构设置区域
- 发行按钮

3.2.5 输入信息

- 彩票编号：字符串，系统自动生成，格式为"BLSRT-YYYYMMDD-XXX"，其中 YYYYMMDD 为日期，XXX 为序号

- 彩票价格：数值，单位为 ETH，精度为小数点后 18 位，有效范围为 0.001-10 ETH
- 开始时间：日期时间，格式为 YYYY-MM-DD HH:mm:ss，必须大于当前时间
- 结束时间：日期时间，格式为 YYYY-MM-DD HH:mm:ss，必须大于开始时间
- 开奖时间：日期时间，格式为 YYYY-MM-DD HH:mm:ss，必须大于结束时间且与结束时间间隔至少 1 小时
- 奖项结构：JSON 格式，定义各奖项的中奖规则和奖金比例，例如：

```
[
  {"rank": 1, "percentage": 50, "description": "一等奖"},
  {"rank": 2, "percentage": 30, "description": "二等奖"},
  {"rank": 3, "percentage": 20, "description": "三等奖"}
]
```

输入方式：通过 Web 界面表单输入，数据来源为管理员手动输入。安全保密条件：管理员需通过钱包签名验证身份。。

3.2.6 输出信息

- 彩票发行成功提示：文本信息，显示"彩票发行成功"
- 彩票信息：JSON 格式，包含彩票 ID、价格、时间等信息
- 交易哈希：字符串，格式为"0x"开头的 64 位十六进制字符串，作为区块链交易凭证
- 错误信息：文本信息，当输入数据不符合要求时显示相应错误提示输出媒体：Web 界面，格式为 HTML/CSS 渲染的表单和提示信息。

3.2.7 算法

彩票发行模块主要涉及时间验证算法，确保设置的时间满足以下条件：

1. 开始时间 < 结束时间 < 开奖时间

开始时间 > 当前时间

开奖时间与结束时间的间隔满足最小锁定时间要求。

```
function validateTimes(startTime, endTime, unlockTime, currentTime, minLockDuration) {
    if (startTime <= currentTime) return false;
    if (startTime >= endTime) return false;
    if (endTime >= unlockTime) return false;
    if (unlockTime - endTime < minLockDuration) return false;
```

```
        return true;
    }
```

3.2.8 类设计

3.2.8.1 类图

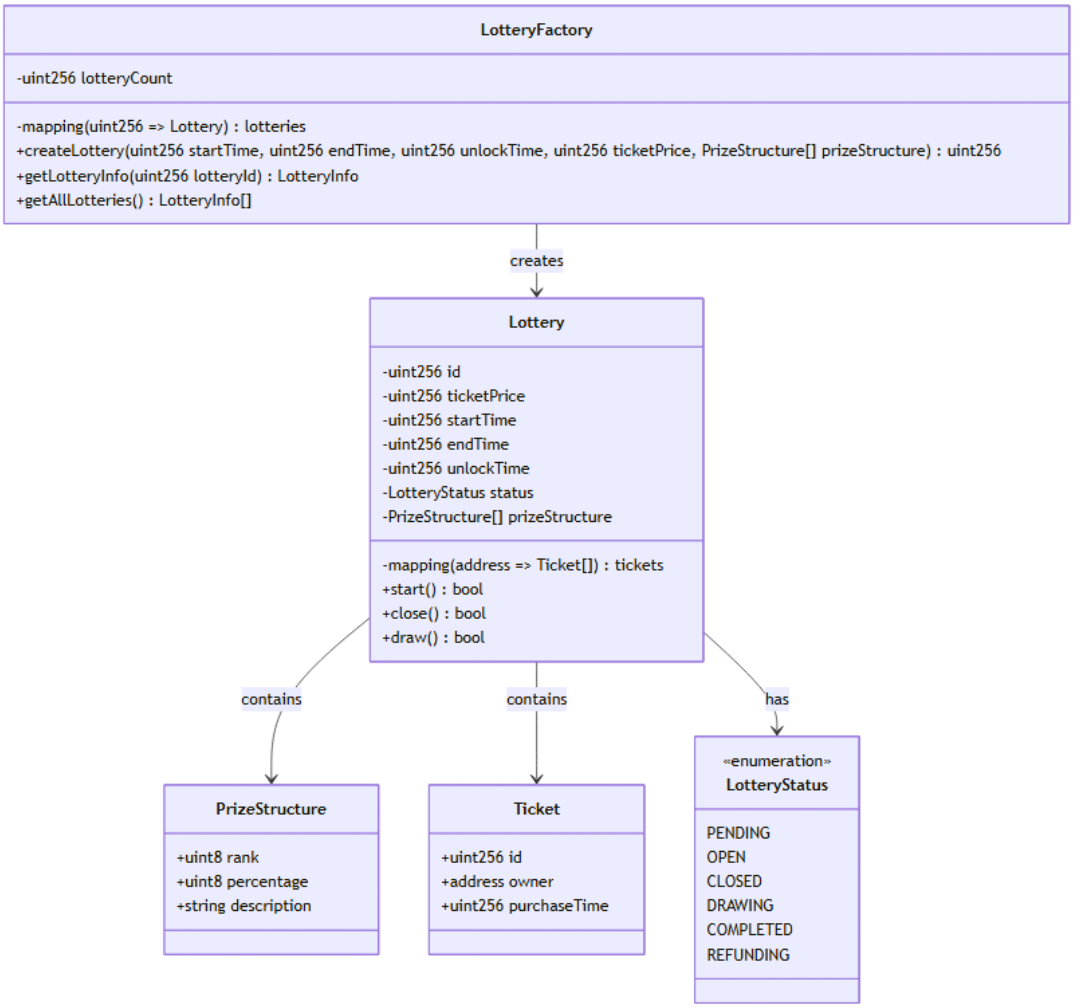

图 3-1 类图

3.2.8.2 类说明

1. LotteryFactory 类：

功能：负责创建和管理彩票实例

主要方法：

createLottery：创建新彩票，返回彩票 ID

getLotteryInfo：获取指定 ID 彩票的详细信息

getAllLotteries：获取所有彩票列表

2. Lottery 类：

功能：表示单个彩票实例，包含彩票的所有信息和状态

主要方法：

start：开始彩票销售

close：结束彩票销售

draw：执行开奖操作

3. PrizeStructure 类：

功能：定义彩票的奖项结构

属性：

rank：奖项等级

percentage：奖金比例

description：奖项描述

4. Ticket 类：

功能：表示用户购买的彩票

属性：

id：彩票 ID

owner：所有者地址

purchaseTime：购买时间

5. LotteryStatus 枚举：

功能：定义彩票的不同状态

值：

OPEN：开放购买

CALCUALTING：计算中

LOCKED：开奖中

REFUNDING：退款中。

3.3 开奖模块

3.3.1 模块编号和功能描述

- 模块编号：DS_BLSRT02

- 功能描述：系统在预设时间到达后，通过链上随机数生成机制进行开奖，确定中奖号码，并根据中奖规则分配奖金。

3.3.2 操作者

- 智能合约（自动执行）
- 系统管理员（触发开奖）

3.3.3 与本模块相关的码表和表

表 3-3 模块功能表

名称	中文注释	类型	作用
DrawResult	开奖结果表	结构体	存储开奖结果信息
WinningTicket	中奖彩票表	结构体	存储中奖彩票信息

3.3.4 界面设计与说明

开奖结果展示界面包含以下元素：
- 彩票期号显示
- 开奖时间显示
- 中奖号码展示
- 中奖名单展示
- 奖金分配情况展示

3.3.5 输入信息

- 彩票 ID：整数，由系统生成，用于标识特定彩票

随机数种子：字节数组，由 drand 服务提供，用于生成随机数
开奖时间：时间戳，表示开奖的确切时间
输入方式：通过智能合约自动获取或管理员触发。数据来源包括区块链时间戳和 drand 随机数服务。

3.3.6 输出信息

- 开奖结果：包含以下信息：

中奖号码：字符串，格式根据彩票类型定义
中奖彩票列表：数组，包含所有中奖彩票的 ID 和所有者地址
各奖项奖金金额：数值，单位为 ETH，精度为小数点后 18 位
交易哈希：字符串，格式为"0x"开头的 64 位十六进制字符串，作为开奖交易的凭证
输出媒体：Web 界面和区块链事件日志。

3.3.7 算法

开奖模块主要涉及随机数生成和中奖号码确定算法：

```
function generateWinningNumber(bytes32 randomness, uint256 lotteryId) internal pu
re returns (uint256) {
    // 使用随机数和彩票 ID 生成中奖号码
    return uint256(keccak256(abi.encodePacked(randomness, lotteryId))) % 1000000;
}
function determineWinners(uint256 winningNumber, uint256 lotteryId) internal retur
ns (address[] memory) {
    // 根据中奖号码确定中奖者
    address[] memory winners;
    // 实现中奖逻辑
    return winners;
}
```

3.3.8 类设计

3.3.8.1 类图

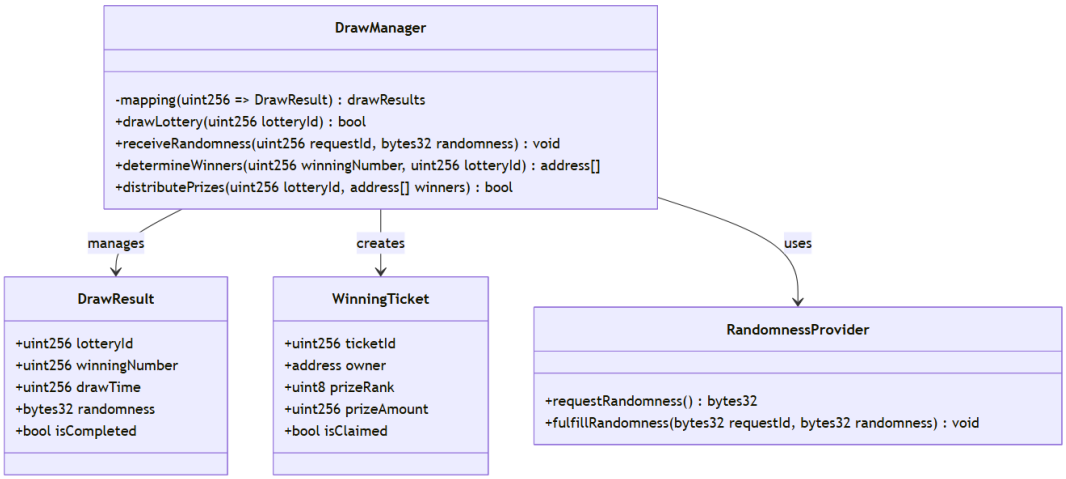

图 3-2 类图

3.3.8.2 类说明

1. DrawManager 类：

   功能：负责管理彩票开奖流程，包括请求随机数、确定中奖者和分配奖金

   主要方法：

   drawLottery：触发彩票开奖流程，返回操作是否成功

   receiveRandomness：接收随机数提供者返回的随机数，并进行后续处理

   determineWinners：根据随机数和彩票规则确定中奖者列表

   distributePrizes：向中奖者分配奖金

2. DrawResult 类：

   功能：存储彩票开奖结果的详细信息

   属性：

   lotteryId：彩票 ID，标识特定彩票

   winningNumber：中奖号码，由随机数生成

   drawTime：开奖时间，记录开奖的确切时间戳

randomness：原始随机数，用于验证开奖公平性

isCompleted：开奖是否完成的标志

3. WinningTicket 类：

功能：表示中奖的彩票，包含中奖信息和奖金领取状态

属性：

ticketId：彩票 ID，标识特定彩票

owner：所有者地址，中奖者的钱包地址

prizeRank：奖项等级，如一等奖、二等奖等

prizeAmount：奖金金额，中奖者应获得的奖金

isClaimed：奖金是否已领取的标志

4. RandomnessProvider 类：

功能：提供随机数服务，确保开奖的公平性和不可预测性

主要方法：

requestRandomness：请求生成随机数，返回请求 ID

fulfillRandomness：完成随机数请求，将随机数返回给调用者

3.4 奖金管理模块

3.4.1 模块编号和功能描述

- 模块编号：DS_BLSRT03
- 功能描述：管理系统奖金池，记录奖金流入和流出，确保奖金分配的透明性和安全性。

3.4.2 操作者

- 智能合约（自动执行）
- 用户（领取奖金）

3.4.3 与本模块相关的码表和表

表 3-4 模块功能表

名称	中文注释	类型	作用

PrizePool	奖金池表	结构体	存储奖金池信息
PrizeDistribution	奖金分配表	结构体	记录奖金分配情况

3.4.4 界面设计与说明

奖金管理界面包含以下元素：
- 当前奖金池余额显示
- 个人中奖记录
- 领取奖金按钮

3.4.5 输入信息

- 用户地址：字符串，格式为"0x"开头的 40 位十六进制字符串，表示用户的钱包地址
- 彩票 ID：整数，用于标识特定彩票
- 领取奖金请求：包含用户地址和彩票 ID 的请求输入方式：通过 Web 界面操作或智能合约自动执行。

3.4.6 输出信息

- 奖金池状态：包含以下信息：

当前余额：数值，单位为 ETH，精度为小数点后 18 位
累计流入：数值，表示历史总流入金额
累计流出：数值，表示历史总流出金额
奖金分配记录：数组，包含每次奖金分配的详细信息
个人中奖记录：数组，包含用户的所有中奖记录
输出媒体：Web 界面和区块链事件日志。

3.4.7 算法

奖金管理模块主要涉及奖金分配算法：

```
function distributePrizes(uint256 lotteryId, address[] memory winners, uint8[] memor
y prizeRanks) internal returns (bool) {
   uint256 totalPrize = getPrizePool(lotteryId);

   for (uint256 i = 0; i < winners.length; i++) {
      uint256 prizeAmount = calculatePrizeAmount(totalPrize, prizeRanks[i], lotteryId
);
      transferPrize(winners[i], prizeAmount);
```

```
        recordPrizeDistribution(lotteryId, winners[i], prizeRanks[i], prizeAmount);
    }

    return true;
}
function calculatePrizeAmount(uint256 totalPrize, uint8 prizeRank, uint256 lotteryId)
 internal view returns (uint256) {
    // 根据奖项等级和总奖金计算奖金金额
    PrizeStructure[] memory prizeStructure = getPrizeStructure(lotteryId);

    for (uint256 i = 0; i < prizeStructure.length; i++) {
        if (prizeStructure[i].rank == prizeRank) {
            return totalPrize * prizeStructure[i].percentage / 100;
        }
    }
   return 0;
}
```

3.4.8 类设计

3.4.8.1 类图

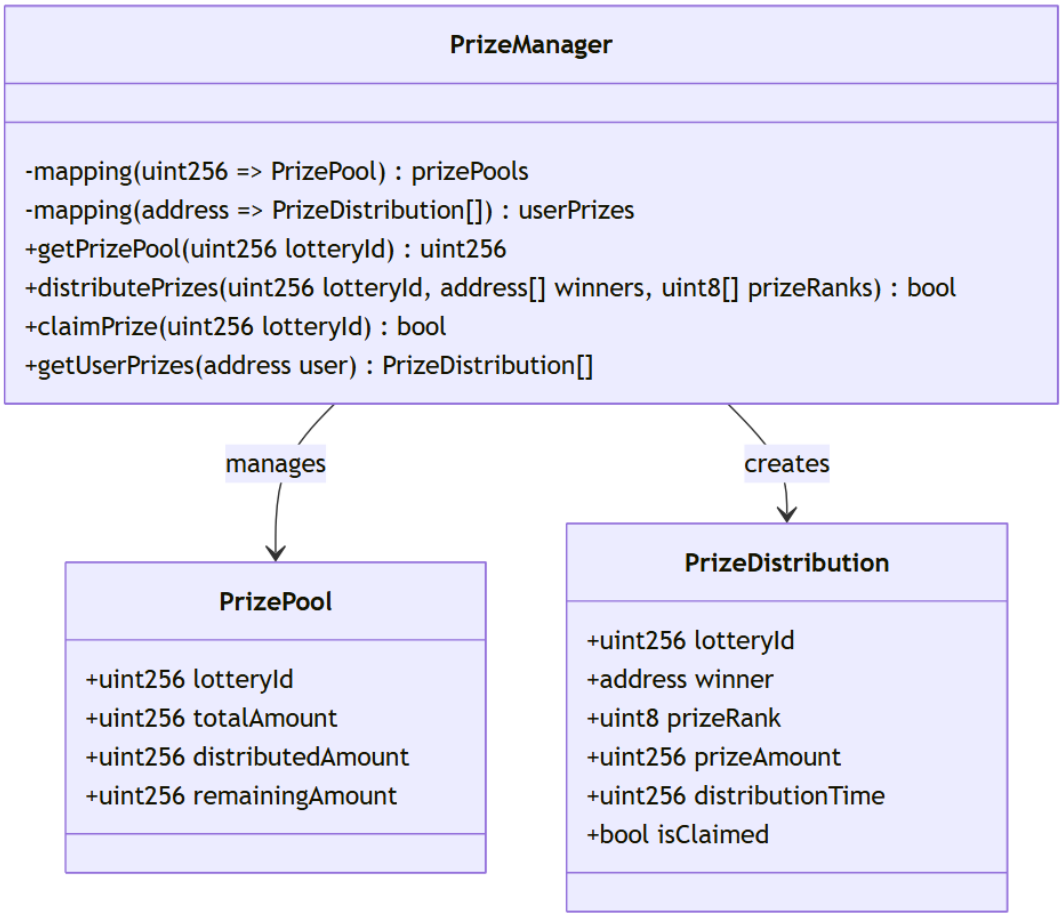

图 3-3 类图

3.4.8.2 类说明

1. PrizeManager 类：

功能：管理彩票奖金池和奖金分配，处理用户奖金领取请求

主要方法：

getPrizePool：获取指定彩票的奖金池金额

distributePrizes：根据中奖者列表和奖项等级分配奖金

claimPrize：处理用户领取奖金的请求

getUserPrizes：获取指定用户的所有中奖记录

2. PrizePool 类：

功能：表示单个彩票的奖金池，记录奖金总额和分配情况

属性：

lotteryId：彩票 ID，标识特定彩票

totalAmount：奖金池总额，所有购票金额的累计

distributedAmount：已分配奖金，已发放给中奖者的金额

remainingAmount：剩余奖金，尚未分配的奖金金额

3. PrizeDistribution 类：

功能：记录单次奖金分配的详细信息

属性：

lotteryId：彩票 ID，标识特定彩票

winner：中奖者地址，接收奖金的钱包地址

prizeRank：奖项等级，如一等奖、二等奖等

prizeAmount：奖金金额，分配给该中奖者的具体金额

distributionTime：分配时间，记录奖金分配的时间戳

isClaimed：奖金是否已领取的标志

3.5 交易记录查询

3.5.1 模块编号和功能描述

- 模块编号：DS_BLSRT04
- 功能描述：提供用户查询购彩记录、中奖记录和历史交易的功能，确保系统透明性。

3.5.2 操作者

- 用户

3.5.3 与本模块相关的码表和表

表 3-5 模块功能表

名称	中文注释	类型	作用
Transaction	交易记录表	结构体	存储交易记录信息
UserActivity	用户活动表	结构体	记录用户活动情况

3.5.4 界面设计与说明

交易记录查询界面包含以下元素：
- 查询条件选择（时间范围、交易类型等）
- 交易记录列表
- 详细信息查看按钮
- 导出功能

3.5.5 输入信息

- 用户地址：字符串，格式为"0x"开头的 40 位十六进制字符串，表示用户的钱包地址
- 查询条件：包含以下可选参数：
- 开始时间：日期时间，格式为 YYYY-MM-DD HH:mm:ss
- 结束时间：日期时间，格式为 YYYY-MM-DD HH:mm:ss
- 交易类型：枚举值，可选"购买"、"中奖"、"领奖"等
- 彩票 ID：整数，用于筛选特定彩票的记录。

输入方式：通过 Web 界面表单输入，数据来源为用户手动输入和系统自动获取的用户地址。安全保密条件：用户需通过钱包签名验证身份。

3.5.6 输出信息

- 交易记录列表：表格形式，包含以下字段：
- 交易 ID：字符串，格式为"0x"开头的 64 位十六进制字符串
- 交易类型：字符串，如"购买"、"中奖"、"领奖"等
- 交易时间：日期时间，格式为 YYYY-MM-DD HH:mm:ss
- 交易金额：数值，单位为 ETH，精度为小数点后 18 位
- 彩票 ID：整数，关联的彩票 ID
- 状态：字符串，如"成功"、"失败"、"处理中"等
- 详细信息：弹窗或新页面，显示交易的完整详情

- 导出文件：CSV 或 PDF 格式，包含查询结果的所有记录输出媒体：Web 界面和可下载文件。

3.5.7 算法

交易记录查询模块主要涉及数据过滤和排序算法：

```
function filterTransactions(transactions, filters) {
  return transactions.filter(tx => {
    // 时间范围过滤
    if (filters.startTime && new Date(tx.timestamp) < new Date(filters.startTime)) r
eturn false;
    if (filters.endTime && new Date(tx.timestamp) > new Date(filters.endTime)) ret
urn false;

    // 交易类型过滤
    if (filters.txType && tx.txType !== filters.txType) return false;

    // 彩票 ID 过滤
    if (filters.lotteryId && tx.lotteryId !== filters.lotteryId) return false;

    return true;
  });
}
function sortTransactions(transactions, sortField, sortOrder) {
  return transactions.sort((a, b) => {
    if (sortOrder === 'asc') {
      return a[sortField] > b[sortField] ? 1 : -1;
    } else {
      return a[sortField] < b[sortField] ? 1 : -1;
    }
  });
}
```

3.5.8 类设计

3.5.8.1 类图

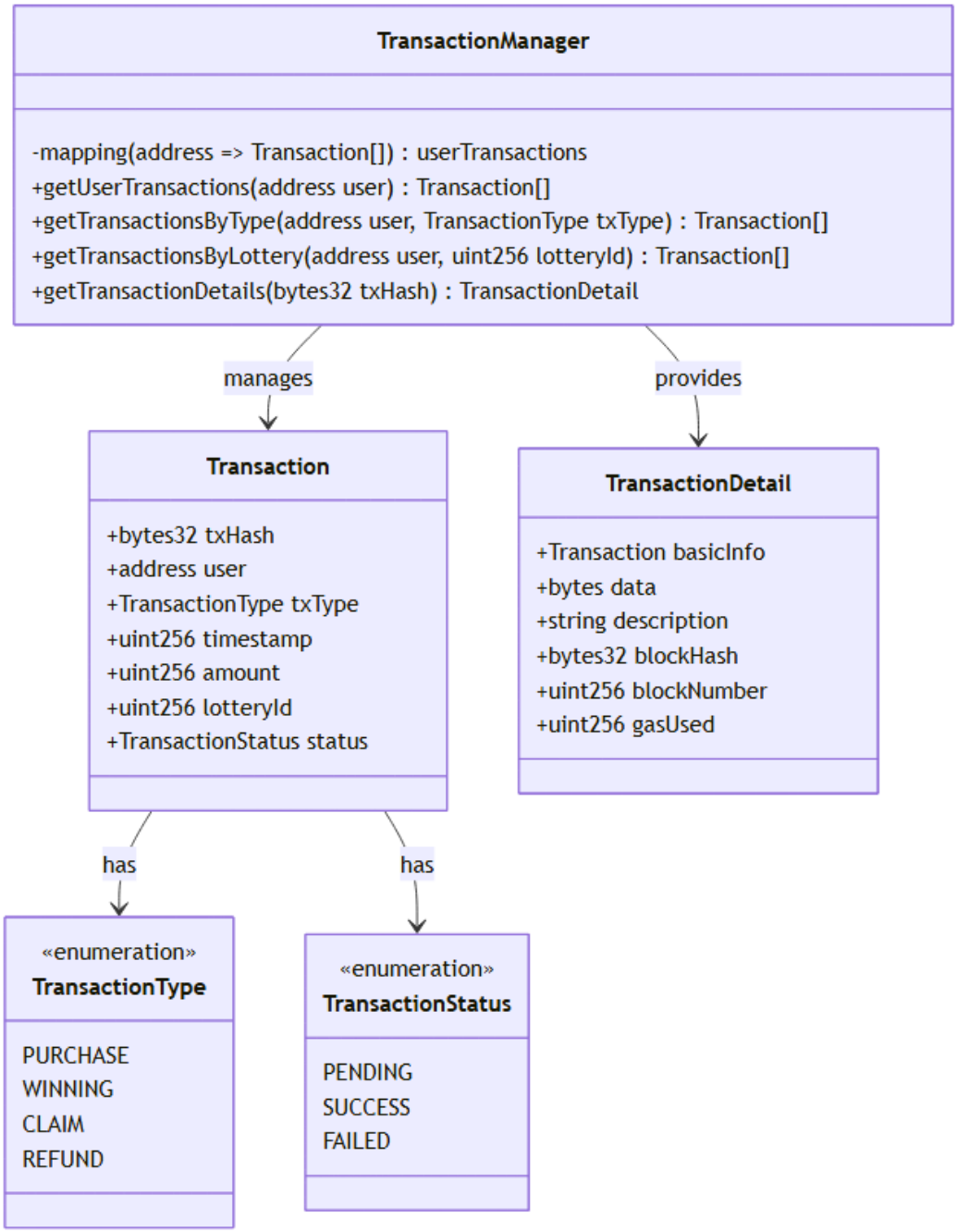

图 3-4 类图

3.5.8.2 类说明

1. TransactionManager 类：

   功能：负责管理用户的所有交易记录，提供交易信息的查询接口。

   主要方法：

   getUserTransactions(address user): T 获取指定用户的所有交易记录。

   getTransactionsByType(address user, TransactionType txType): 获取指定用户的指定类型交易记录。

   getTransactionsByLottery(address user, uint256 lotteryId): 获取用户在某一彩票中的所有交易记录。

   getTransactionDetails(bytes32 txHash):根据交易哈希获取详细的交易信息。

2. Transaction 类

   功能：表示一次具体的交易记录。

   属性：

   txHash: bytes32 交易的唯一哈希标识。

   user: address 发起该交易的用户地址。

   txType: TransactionType 交易的类型（如购票、领奖等）。

   timestamp: uint256 交易时间戳。

   amount: uint256 交易涉及的金额（单位：wei）。

   lotteryId: uint256 交易关联的彩票 ID。

   status: TransactionStatus 交易状态（如成功、失败等）。

3. TransactionDetail 类

   功能：提供更详细的交易信息，通常用于链上查询返回。

   属性：

   basicInfo: Transaction 交易的基本信息（含用户、金额等）。

   data: bytes 附加数据（例如调用合约的输入参数）。

   description: string 交易说明或注释。

   blockHash: bytes32 包含该交易的区块哈希。

   blockNumber: uint256 区块编号。

   gasUsed: uint256 该交易所消耗的 Gas 数量。

4. TransactionType 枚举

   功能：定义交易的业务类型。

属性：

OPEN: 开放购买彩票

CALCULATING: 计算中

LOCKED: 已锁定

REFUNDING: 退款

5. TransactionStatus 枚举

功能：定义交易的执行状态。

属性：

PENDING: 等待确认或处理中

SUCCESS: 成功完成

FAILED: 执行失败或被拒绝

4 视图设计

4.1 界面风格设计

本系统采用现代简约的设计风格，以提供清晰、直观的用户体验。主要设计原则如下：

1. 色彩方案：

主色调：深蓝色 (#1976D2)

辅助色：浅灰色 (#F5F5F5)

强调色：橙色 (#FF9800)

成功色：绿色 (#4CAF50)

错误色：红色 (#F44336)

2. 排版：

主标题：Roboto, 24px, 粗体

副标题：Roboto, 18px, 半粗体

正文：Roboto, 14px, 常规

按钮文字：Roboto, 14px, 半粗体

3. 组件风格：

卡片式布局，带有轻微阴影

圆角按钮 (4px 边角半径)

扁平化设计，减少过度装饰

响应式设计，适应不同屏幕尺寸

4. 交互反馈：

按钮点击时有轻微动画效果

加载状态显示进度指示器

操作成功/失败时显示提示消息

重要操作需二次确认

4.2 主界面设计

系统主界面采用标签页布局，包含以下几个主要部分：

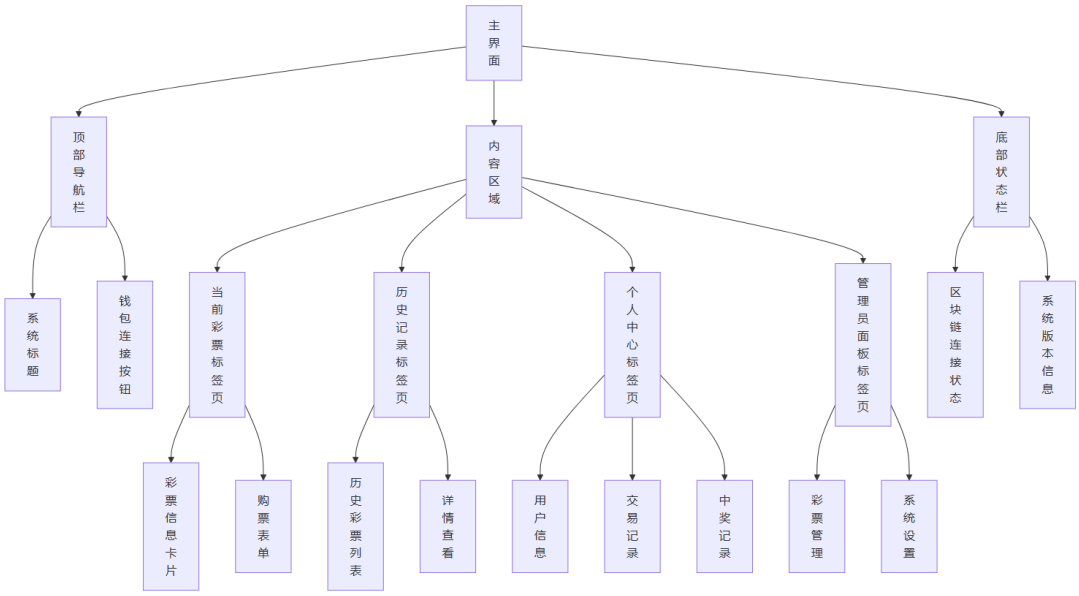

图 4-1 主界面布局图

主界面各部分功能说明：

1. 顶部导航栏：

系统标题：显示系统名称"区块链彩票系统"

钱包连接按钮：用于连接/断开用户钱包，显示钱包地址或"连接钱包"文本

2. 内容区域：

当前彩票标签页：显示当前正在进行的彩票信息和购票功能

历史记录标签页：显示历史彩票记录和详情

个人中心标签页：显示用户个人信息、交易记录和中奖记录

管理员面板标签页：仅管理员可见，用于彩票管理和系统设置

3. 底部状态栏：

区块链连接状态：显示当前连接的区块链网络

系统版本信息：显示系统版本号

5 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 5-1 构件接口列表

模块名称	接口编号	接口名称	接口类型	说明
彩票管理	IF_BLSRT_01	createLottery	内部	创建新彩票
彩票管理	IF_BLSRT_02	drawLottery	内部	触发彩票开奖
彩票购买	IF_BLSRT_03	buyTicket	内部	购买彩票
奖金管理	IF_BLSRT_04	claimPrize	内部	领取奖金
查询接口	IF_BLSRT_05	getLotteryInfo	外部	获取彩票信息
查询接口	IF_BLSRT_06	getUserTickets	外部	获取用户彩票
查询接口	IF_BLSRT_07	getUserWinningHistory	外部	获取用户中奖历史

5.1 createLottery 接口

1. 接口属性设计

表 5-2 createLottery 接口说明

接口编号	IF_BLSRT_01
接口名称	createLottery
接口说明	创建新的彩票，设置彩票的基本信息和奖项结构
数据来源	管理员输入
调用者	系统管理员
输入	startTime: uint256 - 开始时间 endTime: uint256 - 结束时间 unlockTime: uint256 - 开奖时间 ticketPrice: uint256 - 票价 prizeStructure: PrizeStructure[] - 奖项结构
输出	lotteryId: uint256 - 彩票 ID
处理流程	1. 验证输入参数 2. 检查时间设置是否合理 3. 创建彩票记录 4. 设置奖项结构 5. 返回彩票 ID

2. 接口处理流程图

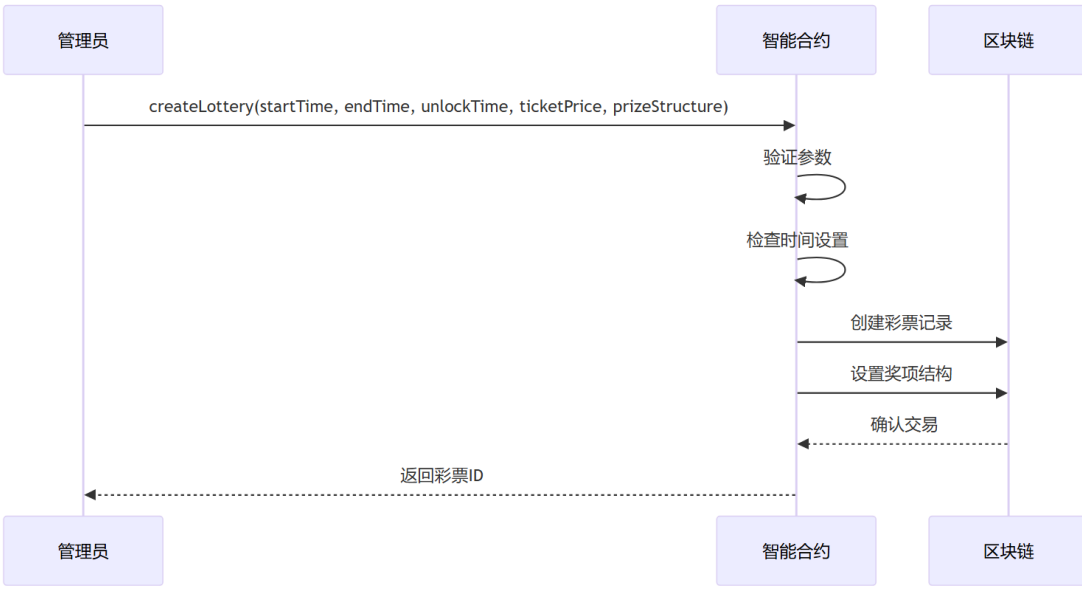

图 5-1 createLottery 接口处理流程图

5.2 drawLottery 接口

1. 接口属性设计

表 5-3 drawLottery 接口说明

接口编号	IF_BLSRT_03
接口名称	drawLottery
接口说明	触发彩票开奖流程，请求随机数并确定中奖结果
数据来源	管理员操作或系统自动触发
调用者	系统管理员或智能合约
输入	- lotteryId: uint256 - 彩票 ID
输出	- success: bool - 操作是否成功
处理流程	1. 验证彩票 ID 是否有效 2. 检查彩票当前状态是否为 OPEN 3. 检查当前时间是否已达到开奖时间 4. 更新彩票状态为 DRAWING 5. 请求随机数 6. 记录随机数请求 ID 7. 返回操作结果

2. 接口处理流程图

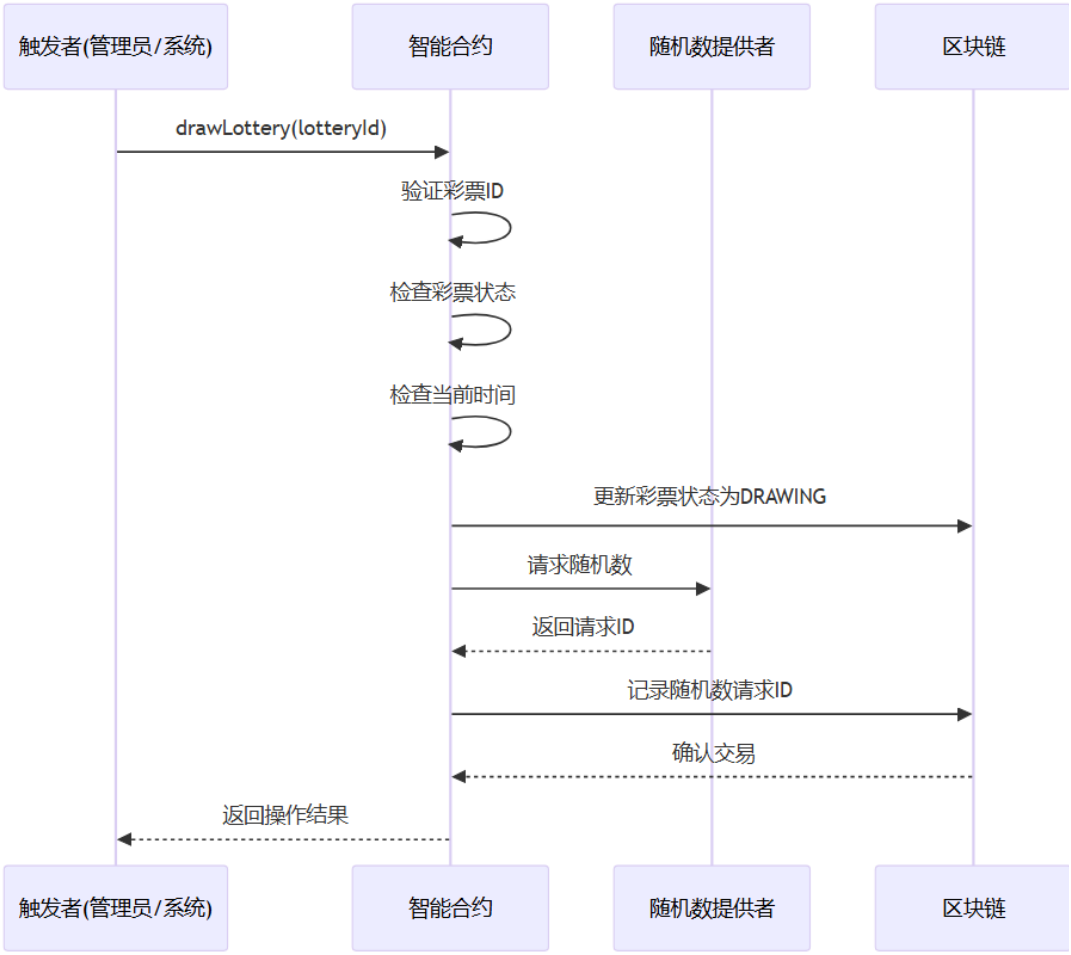

图 5-2 drawLottery 接口处理流程图

5.3 buyTicket 接口

1. 接口属性设计

表 5-4 buyTicket 接口说明

接口编号	IF_BLSRT_04
接口名称	buyTicket
接口说明	用户购买彩票，支付相应金额并获得彩票
数据来源	用户操作
调用者	系统用户
输入	- lotteryId: uint256 - 彩票 ID value: uint256 – 支付金额（通过交易 value 传入）
输出	- ticketId: uint256 - 购买的彩票 ID
处理流程	1. 验证彩票 ID 是否有效

	2. 检查彩票当前状态是否为 OPEN 3. 验证支付金额是否等于票价 4. 生成新的彩票 ID 5. 记录用户购买信息 6. 更新奖金池 7. 触发彩票购买事件 8. 返回彩票 ID

2. 接口处理流程图

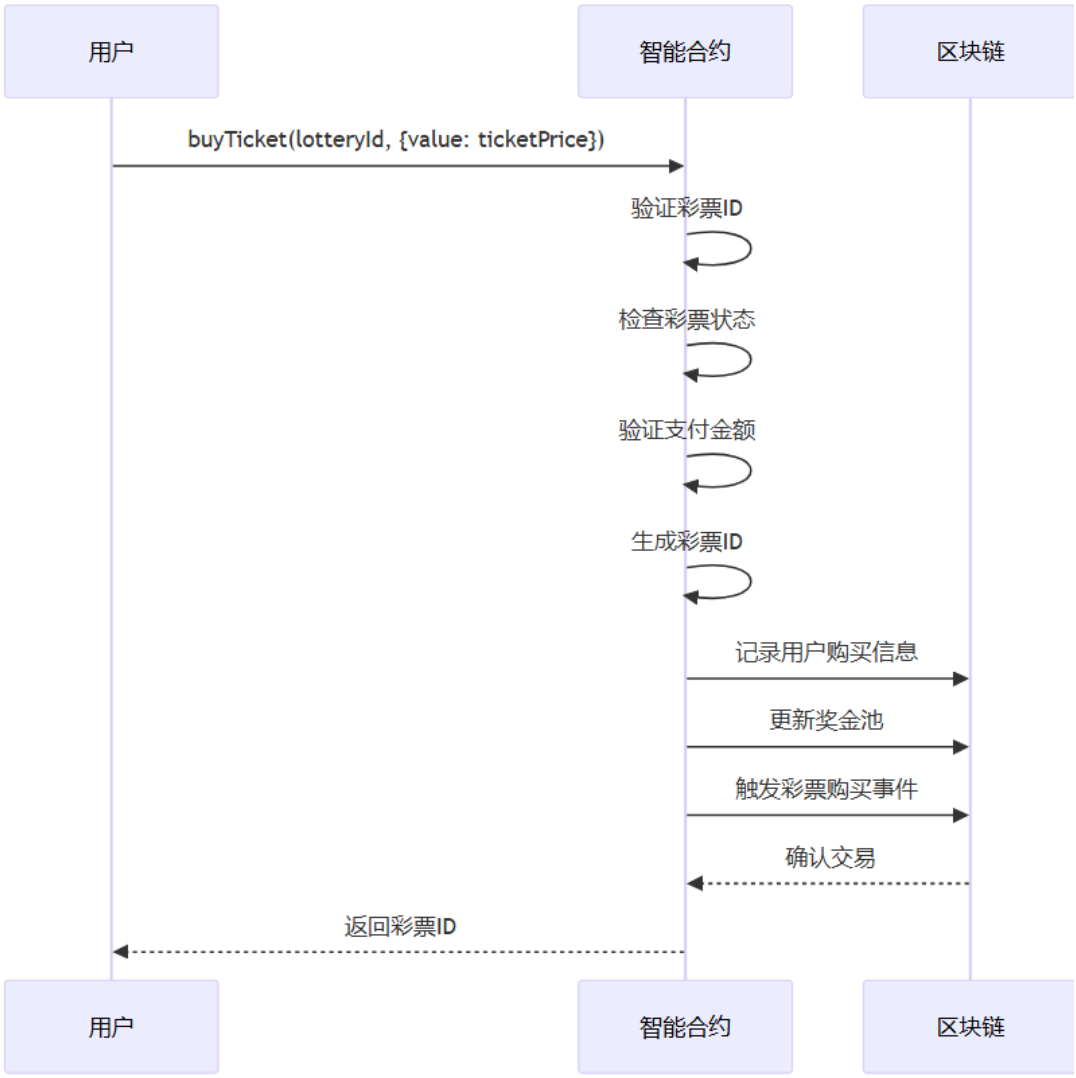

图 5-3 buyTicket 接口处理流程图

5.4 claimPrize 接口

1. 接口属性设计

表 5-5 claimPrize 接口说明

接口编号	IF_BLSRT_05
接口名称	claimPrize
接口说明	用户领取中奖奖金
数据来源	用户操作
调用者	系统用户
输入	- lotteryId: uint256 - 彩票 ID
输出	- success: bool - 操作是否成功
处理流程	1. 验证彩票 ID 是否有效 2. 检查彩票当前状态是否为 COMPLETED 3. 验证用户是否有未领取的奖金 4. 计算应领取的奖金金额 5. 将奖金转账给用户 6. 更新奖金领取状态 7. 触发奖金领取事件 8. 返回操作结果和奖金金额

2. 接口处理流程图

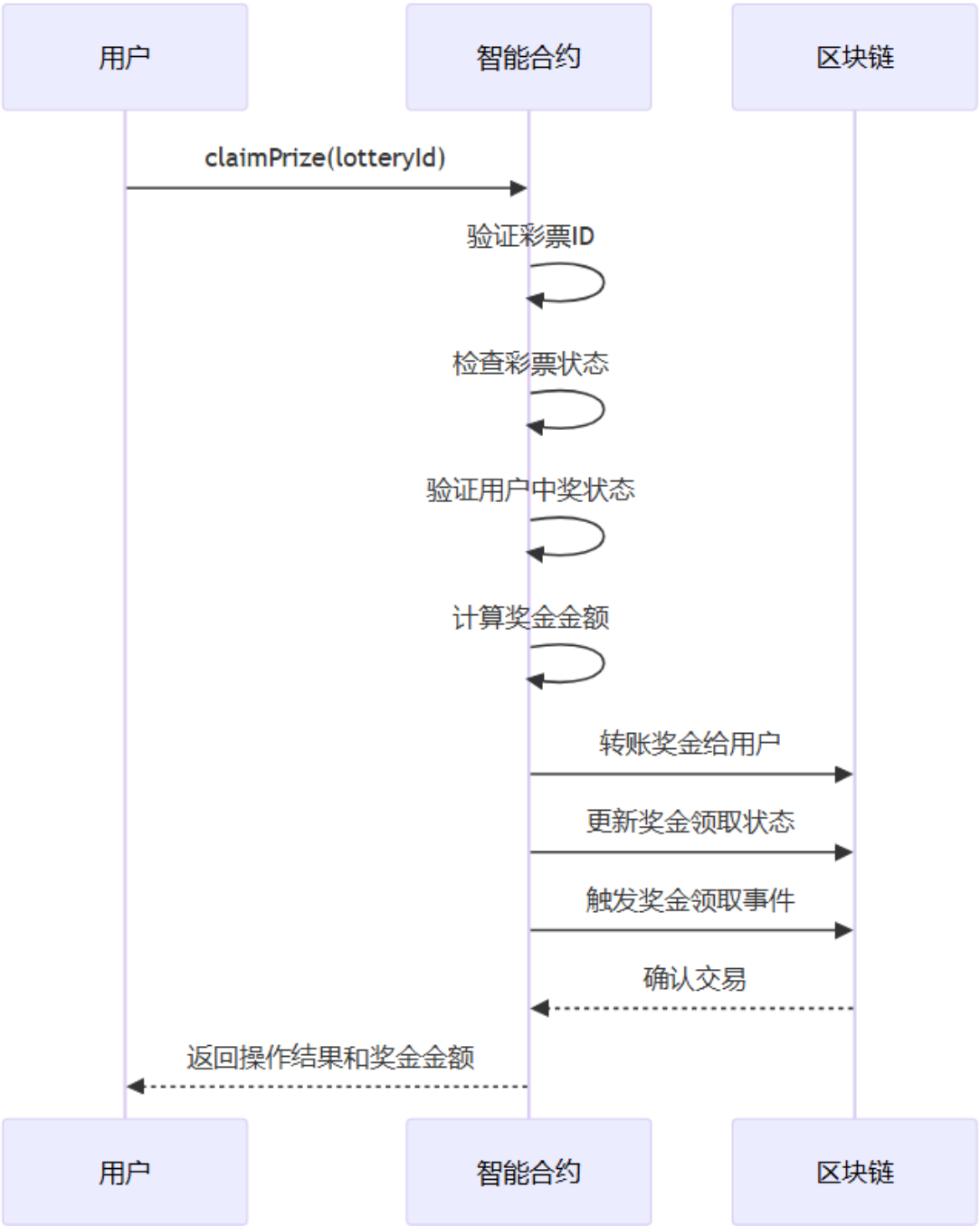

图 5-4 claimPrize 接口处理流程图

5.5 getLotteryInfo 接口

1. 接口属性设计

表 5-6 getLotteryInfo 接口说明

接口编号	IF_BLSRT_06
接口名称	getLotteryInfo

接口说明	获取指定彩票的详细信息
数据来源	区块链数据
调用者	任何用户或前端应用
输入	- lotteryId: uint256 - 彩票 ID

输出	LotteryInfo: {
	id: uint256 - 彩票 ID
	ticketPrice: uint256 - 票价
	startTime: uint256 - 开始时间
	endTime: uint256 - 结束时间
	unlockTime: uint256 - 开奖时间
	status: uint8 - 彩票状态
	prizePool: uint256 - 奖金池金额
	totalTickets: uint256 - 总票数
	winningNumber: uint256 - 中奖号码（如已开奖）
	prizeStructure: PrizeStructure[] - 奖项结构
	}

处理流程	1. 验证彩票 ID 是否有效 2. 从区块链读取彩票信息 3. 格式化并返回彩票信息

2. 接口处理流程图

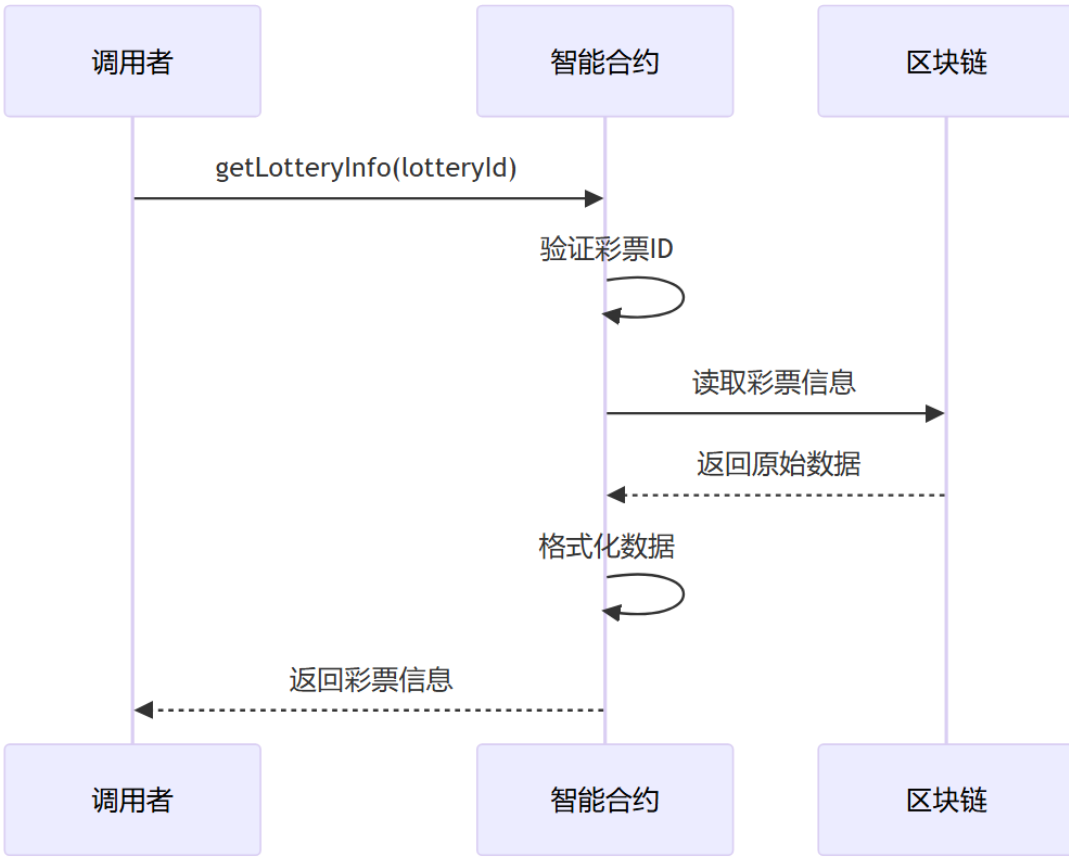

图 5-5 getLotteryInfo 接口处理流程图

5.6 getUserTickets 接口

1. 接口属性设计

表 5-7getUserTickets 接口说明

接口编号	**IF_BLSRT_07**
接口名称	getUserTickets
接口说明	获取用户在指定彩票中购买的所有彩票
数据来源	区块链数据
调用者	用户或前端应用
输入	- lotteryId: uint256 - 彩票 ID - userAddress: address - 用户地址
输出	Ticket[]: [{ id: uint256 - 彩票 ID owner: address - 所有者地址 purchaseTime: uint256 - 购买时间 }]

处理流程	1. 验证彩票 ID 是否有效 2. 验证用户地址是否有效 3. 从区块链读取用户彩票信息 4. 格式化并返回彩票列表

2. 接口处理流程图

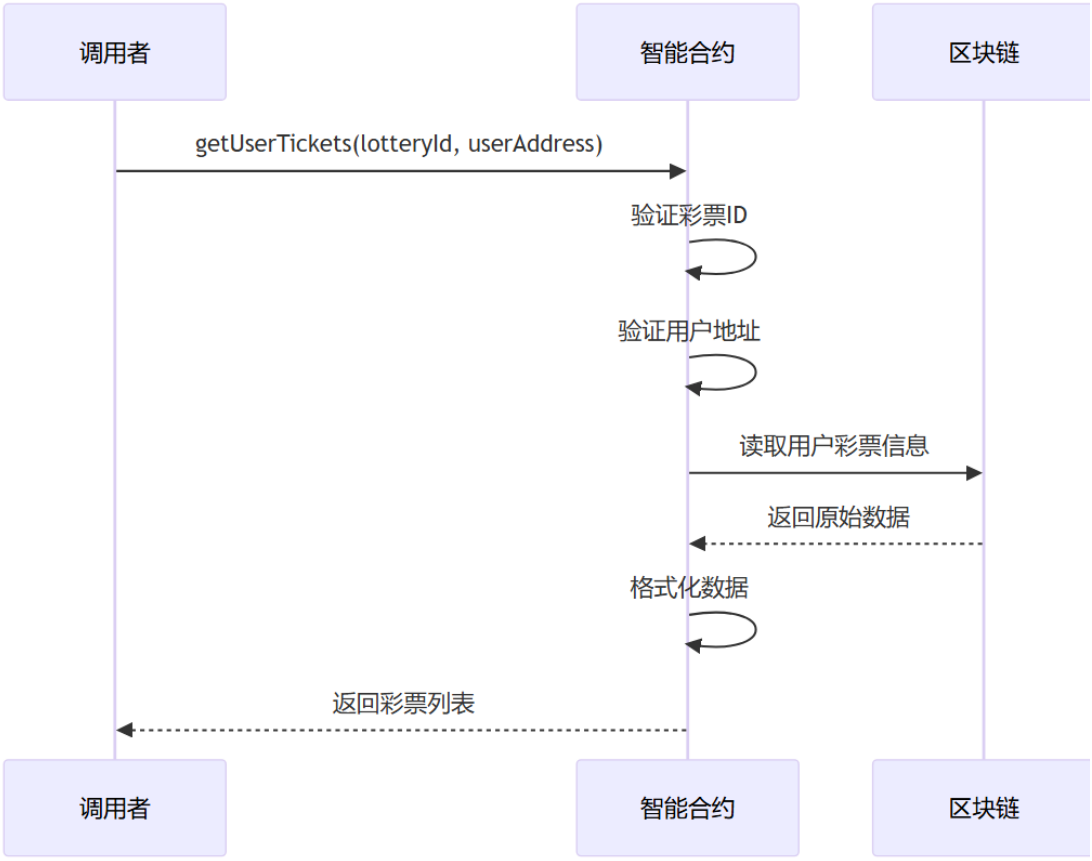

图 5-6 getUserTickets 接口处理流程图

5.7 getUserWinningHistory 接口

1. 接口属性设计

表 5-8 getUserWinningHistory 接口说明

接口编号	IF_BLSRT_08
接口名称	getUserWinningHistory
接口说明	获取用户的中奖历史记录
数据来源	区块链数据
调用者	用户或前端应用
输入	- userAddress: address - 用户地址
输出	WinningHistory: {

	lotteryIds: uint256[] - 中奖彩票 ID 列表 amounts: uint256[] - 对应的中奖金额列表 claimStatus: mapping(uint256 => bool) - 奖金领取状态 }
处理流程	1. 验证用户地址是否有效 2. 从区块链读取用户中奖记录 3. 查询每个中奖记录的领取状态 4. 格式化并返回中奖历史

2. 接口处理流程图

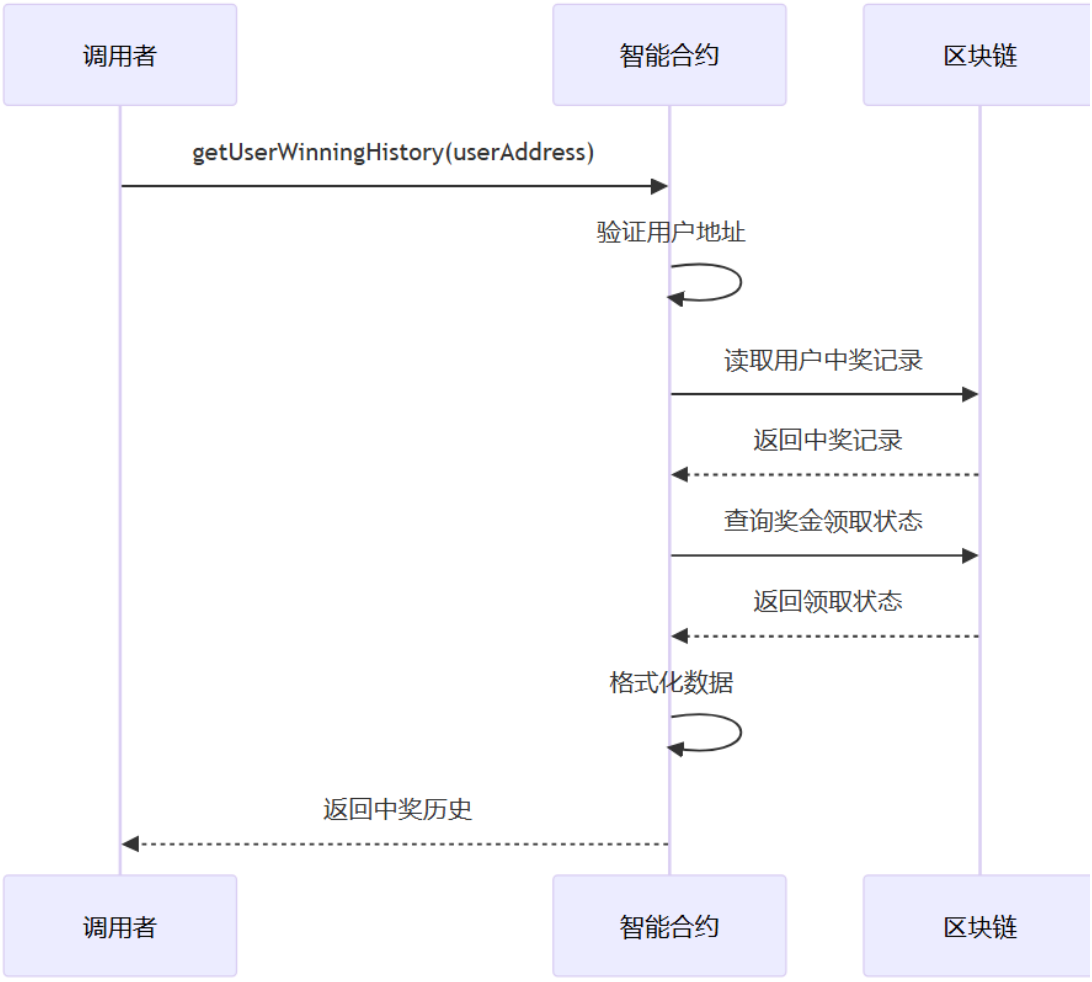

图 5-7 getUserWinningHistory 接口处理流程图

6 系统出错处理设计

6.1 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	钱包连接失败	系统弹出错误提示："钱包连接失败，请检查 MetaMask 是否安装并解锁"，用户需重新尝试连接钱包
2	彩票创建参数无效	系统弹出错误提示："彩票创建失败,请检查参数是否有效"，并标记具体的无效参数，如 "开始时间必须大于当前时间"
3	彩票购买金额不足	系统弹出错误提示："购买失败，支付金额不足"，显示所需金额和当前余额
4	彩票已关闭或未开始	系统弹出错误提示："该彩票当前不可购买"，并显示彩票的当前状态
5	开奖时间未到	系统弹出错误提示："开奖时间未到,无法执行开奖操作",并显示剩余时间
6	随机数生成失败	系统记录错误日志："随机数生成失败"，并自动尝试使用备用随机源
7	奖金领取失败	系统弹出错误提示："奖金领取失败",并显示可能的原因，如 "您没有中奖记录" 或 "奖金已领取"
8	智能合约调用异常	系统弹出错误提示："智能合约调用失败"，并显示错误代码和简要说明

9	网络连接中断	系统弹出错误提示："网络连接中断,请检查您的网络连接"，并在连接恢复后自动重新加载数据
10	权限不足	系统弹出错误提示："权限不足，无法执行该操作"，提示用户使用管理员账户

6.2 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。具体补救措施包括：

- **用户输入错误：**
- 实时表单验证，在用户输入时即时提供反馈
- 提供明确的错误提示和修正建议
- 保留用户已输入的有效数据，避免全部重新输入
- **网络连接问题：**
- 实现自动重连机制
- 本地缓存关键数据，减少网络依赖
- 提供离线模式下的基本功能
- **智能合约交互失败：**
- 实现交易重试机制
- 提供交易状态跟踪
- 记录失败交易，便于后续分析和处理
- **系统异常：**
- 完善的日志记录系统，记录所有异常情况
- 自动错误报告机制，将严重错误发送给开发团队
- 系统状态监控，及时发现并处理异常
- **数据不一致：**
- 定期数据同步机制，确保前端显示与区块链数据一致
- 提供手动刷新功能，允许用户获取最新数据
- 数据校验机制，检测并修复不一致的数据