# 1范围

## 1.1标识

* 标识号：SOFT\_PROJ-2025-CCMS-001
* 标题：校园卡管理系统（Campus Card Management System，CCMS）
* 缩略词语：CCMS
* 版本号：v1.0.0
* 发行号：Initial Release

## 1.2系统概述

校园卡管理系统是一个集门禁通行、餐饮消费、资源预约等多功能于一体的智能管理系统，旨在通过数字化平台提升校园管理效率和服务精准度。该系统主要解决传统人工管理模式下校园卡管理效率低、数据分散等问题，为学生和教职工提供更加便捷、高效的校园服务。

### 系统和软件的一般特性：

* 多功能性：集成门禁控制、餐饮消费、资源预约等多种功能于一体。
* 易用性：提供友好的用户界面和操作流程，方便用户快速上手。
* 可扩展性：支持用户规模的扩展和功能的升级。
* 安全性：采用数据加密和权限管理机制，确保用户数据的安全性和隐私性。

### 系统开发、运行和维护的历史（规划阶段）：

目前，校园卡管理系统正处于规划与开发阶段。项目团队已经完成了初步的需求分析和可行性研究，并制定了详细的项目计划。系统预计将在接下来的几个月内完成开发、测试和部署工作。

#### 项目相关方：

* 投资方：暂无
* 需方：学生和教职工
* 用户：学生和教职工
* 开发方：计算机专业学生团队“AAA电脑维修”
* 支持机构：无

## 1.3文档概述

本文档《校园卡管理系统软件需求规格说明（SRS）》详细描述了校园卡管理系统的软件需求，旨在为系统的设计和实现提供明确的指导。本文档涵盖了系统的功能需求、性能需求、接口需求等多个方面，确保系统的开发能够满足既定的目标和标准。

### 文档用途：

* 作为系统开发的基础文档，指导开发团队进行系统设计、编码和测试工作。
* 作为项目管理和评审的依据，帮助投资方、需方和开发方了解项目的进展和成果。
* 作为用户手册和维护手册的编写基础，为用户提供系统操作指南和运维支持。

### 文档内容：

* 系统概述：简述系统的用途、一般特性、开发历史、项目相关方以及运行现场等信息。
* 需求规格：详细描述系统的功能需求、性能需求、接口需求等，包括输入输出数据、处理流程、算法说明等。
* 合格性规定：定义一组合格性方法，确保每个需求得到满足，包括测试、演示、分析、审查等特殊合格性方法。
* 需求可追踪性：描述从SRS中的需求到系统需求（或子系统需求）的可追踪性，确保每个需求在系统设计、实现和测试阶段都能被追踪和验证。
* 其他部分：包括尚未解决的问题、注解、附录等内容，提供额外的信息和支持。

### 保密性或私密性要求：

* 本文档包含系统的详细需求和设计信息，属于敏感资料。未经授权，不得擅自复制、传播或泄露给第三方。
* 开发团队和相关人员应遵守相关的保密协议和法律法规，确保系统的安全性和隐私性。
* 在系统开发和运维过程中，应采取必要的安全措施和技术手段，保护用户数据的安全和隐私。

## 1.4基线

本SRS文档基于《FAR》文档中的需求分析结果编写，是系统设计和实现的基础。

# 2引用文件

* FAR文档：校园卡管理系统可行性分析报告

# 3需求

## 3.1 所需的状态和方式

* **运行状态**：
  + **活动状态**：系统正常运行时，用户可进行信息查询、修改等操作。
  + **维护状态**：管理员进行数据库维护或系统更新时，普通用户无法访问。
* **运行方式**：所有功能在同一GUI界面下运行，无需区分多模式。

## 3.2 需求概述

### 3.2.1 目标

* **开发意图**：开发简化版虚拟校园卡管理系统，专注基础信息管理功能。
* **主要功能**：
  + 用户信息管理（增删改查）
  + 基础信息展示
  + 简单权限管理
* **处理流程：用户请求 → 业务逻辑层 → 数据访问层 → SQLite数据库**
* **高层次系统图：**

[QT用户界面] ←→ [C++业务逻辑] ←→ [SQLite数据库]

### 3.2.2 运行环境

* **硬件**：x86架构PC（Intel i5/8GB RAM/500MB存储）
* **软件**：
  + Windows 10+/Ubuntu 20.04 LTS
  + Qt 5.15+框架
  + SQLite 3.35+

### 3.2.3 用户特点

| **用户类型** | **技能要求** | **操作范围** |
| --- | --- | --- |
| 管理员 | SQLite基础命令、QT界面操作 | 全功能CRUD及系统维护 |
| 普通用户 | 基础计算机操作 | 个人信息查询与修改 |

### 3.2.4 关键点

* 数据库单表设计（≤500条记录）
* Qt界面响应时间≤1秒
* 学号唯一性约束（8位数字）

### 3.2.5 约束条件

* **开发周期**：2个月（原型1周+核心功能4周+测试3周）
* **技术栈**：C++17/Qt 5/SQLite3
* **预算**：零成本（使用开源工具）

## 3.3 需求规格

### 3.3.1 功能结构

[校园卡管理系统] |-->[用户管理模块] |-->[信息展示模块] |-->[系统管理模块]

### 3.3.3 描述约定

* **时间格式**：24小时制（HH:MM:SS）
* **金额单位**：人民币元（保留2位小数）
* **ID规范**：8位数字学号（正则：^[0-9]{8}$）

## 3.4 CSCI能力需求

### 3.4.1 用户管理

| **要素** | **说明** |
| --- | --- |
| 输入 | 学号（8数字）、姓名（2-10汉字）、密码（SHA-256加密） |
| 处理 | 格式校验 → 数据库事务操作 → 错误回滚 |
| 输出 | JSON格式操作结果（含成功状态/错误码） |

### 3.4.2 信息展示

* ****查询性能**：单条记录检索时间≤100ms**
* ****输出格式**：**

{

"student\_id": "20250001",

"name": "张三",

"balance": 150.00

}

## 3.5 外部接口需求

* **用户接口**：Qt Widgets界面（兼容Windows/Linux）
* **软件接口**：SQLite3 C/C++ Interface

## 3.6 CSCI内部接口需求

无特殊要求，留待设计时决定。

## 3.7 CSCI内部数据需求

需要存储：

* 用户表（ID、姓名、状态等）
* 操作日志表

## 3.8 适应性需求

本系统无需特殊的安装适应性需求，所有功能均通过标准程序界面实现。系统设计时应考虑以下适应性：

* 支持不同屏幕分辨率的显示适配
* 支持主流操作系统（Windows 10/11，macOS最新版本）
* 数据库连接参数可通过配置文件修改

## 3.9 保密性需求

由于是学生课业项目，保密性需求简化如下：

* 用户密码需加密存储（如使用MD5或SHA-256哈希）
* 管理员账号需设置复杂度要求（至少8位，包含字母和数字）
* 普通用户只能查看和修改自己的信息

## 3.10 保密性和私密性需求

* 系统不存储真实敏感信息（如身份证号、银行卡号等）
* 所有用户数据仅用于教学演示目的
* 不要求实现审计日志功能

## 3.11 CSCI环境需求

系统运行环境要求：

* 操作系统：Windows 10/11或macOS 10.15+
* Cpp环境：gcc11或更高版本
* 数据库：SQLite 5.7或更高版本
* 内存：最低4GB RAM
* 存储空间：至少500MB可用空间

## 3.12 计算机资源需求

### 3.12.1 计算机硬件需求

* 处理器：Intel i5或同等性能处理器
* 内存：建议8GB RAM
* 硬盘：SSD存储设备
* 显示器：支持1024×768以上分辨率

### 3.12.2 计算机硬件资源利用需求

* CPU利用率：峰值不超过70%
* 内存占用：不超过2GB
* 存储空间：数据库不超过100MB

### 3.12.3 计算机软件需求

* 开发工具：Qt
* 数据库管理工具：DB Browser
* 版本控制：Git

### 3.12.4 计算机通信需求

* 本地网络连接（用于数据库访问）
* 不要求远程访问功能
* 无特殊通信协议要求

## 3.13 软件质量因素

| **质量属性** | **量化指标** | **实现方案** |
| --- | --- | --- |
| 可靠性 | 数据操作成功率≥99.9% | SQLite事务+WAL日志 |
| 可维护性 | 代码注释率≥30% | Doxygen文档规范 |
| 可测试性 | 单元测试覆盖率≥80% | Google Test框架 |

## 3.14 设计和实现的约束

* **架构约束**：
  + 使用单机部署的SQLite数据库，禁止依赖网络服务。
  + 采用QT框架实现GUI，禁止引入其他图形库（如GTK）。
* **标准与规范**：
  + 密码存储需使用SHA-256加密，符合《中华人民共和国网络安全法》要求。
  + 代码风格遵循Google C++ Style Guide，缩进统一为4空格。
* **灵活性要求**：
  + 数据库表结构需预留扩展字段（如reserved\_field1），以应对未来新增需求。
  + 禁止硬编码学号规则，需通过配置文件支持不同院校的学号格式。

## 3.15 数据

* **输入数据**：
  + 用户信息：学号（唯一键）、姓名、密码（加密）、邮箱等，单条记录≤1KB。
  + 操作日志：时间戳、操作类型、执行人，单条记录≤500B。
* **输出数据**：
  + 用户端：个人信息页面（≤5KB）、消费记录列表（每页≤10条）。
  + 管理端：查询结果表格（支持分页，每页≤20条）。
* **数据管理**：
  + SQLite数据库文件大小限制为≤10MB，支持手动备份/恢复（通过脚本）。

## 3.16 操作

* **常规操作**：用户登录、信息查询、管理员批量导入数据（CSV格式）。
* **特殊操作**：数据库紧急备份（通过命令行工具db\_backup.sh）。
* **初始化操作**：首次运行时自动创建数据库表结构（执行init\_db.sql）。
* **恢复操作**：提供数据回滚功能（基于备份文件，恢复至最近3天的状态）。

## 3.17 故障处理

* **软件故障**：
  + 错误信息：数据库连接失败时提示“系统繁忙，请稍后重试（错误代码：DB\_001）”。
  + 补救措施：自动重试3次后降级为本地缓存模式（仅限查询功能）。
* **硬件故障**：
  + PC断电时，通过SQLite的WAL模式保障数据完整性。
  + 读卡器异常时，提示“设备未连接，请检查硬件（错误代码：HW\_002）”。

## 3.18 算法说明

* **登录认证算法**：
  + 概况：SHA-256哈希比对（用户输入密码→加密→与数据库存储值匹配）。
  + 公式：stored\_hash = SHA256(salt + user\_input)，其中salt为随机8字节。
* **数据筛选算法**：
  + 概况：管理员端按学号前缀快速检索（B树索引优化）。
  + 公式：WHERE student\_id LIKE '2025%' LIMIT 20。

## 3.19 有关人员需求

* **用户**：500名师生，需基础计算机操作能力（如使用浏览器/输入文本）。
* **管理员**：1-2人，需掌握SQLite基础命令（如.dump备份）。
* **开发团队**：5名学生，需C++/QT培训10小时，Git协作规范。

## 3.20 有关培训需求

* **用户培训**：提供5分钟演示视频（覆盖登录、查询、修改密码）。
* **管理员培训**：包含在系统文档中，重点讲解日志审计与数据备份。

## 3.21 有关后勤需求

* **维护**：每月检查数据库文件完整性（通过sqlite3\_check工具）。
* **运输**：系统通过GitHub分发，无需物理介质。

## 3.22 其他需求

* **隐私保护**：禁止记录用户操作行为（如按键日志），符合《个人信息保护法》。
* **兼容性**：支持Windows 10+和Ubuntu 20.04 LTS。

## 3.23 包装需求

* 交付物为ZIP压缩包，内含：可执行文件（CCMS.exe）、数据库文件（ccms.db）、用户手册（README.pdf）。
* 文件命名规范：SOFT\_PROJ-2025-CCMS-001\_v1.0.0\_YYYYMMDD.zip。

## 3.24 需求的优先次序和关键程度

* **关键需求**（P0）：用户登录、学号唯一性、数据加密。
* **高优先级**（P1）：管理端CRUD、操作日志。
* **普通需求**（P2）：界面美化、多语言支持。

# 4、合格性方法

### 方法解释：

* a.演示：运行依赖于可见的功能操作的CSCI或部分CSCI,不需要使用仪器、专用测试设备或进行事后分析；
* b.测试：使用仪器或其他专用测试设备运行CSCI或部分CSCI,以便采集数据供事后分析使用；
* c.分析：对从其他合格性方法中获得的积累数据进行处理，例如测试结果的归约、解释或推断；
* d.审查：对CSCI代码、文档等进行可视化检查；
* e.特殊方法:特殊的合格性方法。任何应用到CSCI的特殊合格性方法，如：专用工具、技术、过程、设施、验收限制。

### 表格：

| **需求ID** | **需求描述** | **演示** | **测试** | **分析** | **审查** | **特殊方法** | **备注** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| REQ-3.1.1 | 系统正常运行时，用户可进行信息查询、修改等操作 |  | ✔ |  |  |  |  |
| REQ-3.1.2 | 管理员进行数据库维护或系统更新时，普通用户无法访问 |  | ✔ |  |  |  |  |
| REQ-3.1.3 | 所有功能在同一GUI界面下运行，无需区分多模式 | ✔ |  | ✔ |  |  |  |
| REQ-3.2.2 | 在x86架构PC（Intel i5/8GB RAM/500MB存储）上支持 | ✔ |  | ✔ |  |  |  |
| REQ-3.2.2 | Windows 10+/Ubuntu 20.04 LTS下均支持 | ✔ |  |  | ✔ |  |  |
| REQ-3.2.4 | 数据库单表设计 |  |  |  | ✔ |  |  |
| REQ-3.2.4 | Qt界面响应时间≤1秒 |  | ✔ |  |  |  |  |
| REQ-3.2.4 | 零成本（使用开源工具） | ✔ |  | ✔ |  |  |  |
| REQ-3.4.2 | 单条记录检索时间≤100ms |  | ✔ |  |  |  |  |
| REQ-3.6.1 | CSCI内部接口 | ✔ |  | ✔ |  |  |  |
| REQ-3.7.1 | CSCI数据要存储用户表和操作日志表等 | ✔ |  | ✔ |  |  |  |
| REQ-3.8.1 | 支持不同屏幕分辨率的显示适配 | ✔ |  | ✔ |  |  |  |
| REQ-3.8.2 | 数据库连接参数可通过配置文件修改 | ✔ |  | ✔ |  |  |  |
| REQ-3.9.1 | 用户密码需加密存储（如使用MD5或SHA-256哈希） |  |  | ✔ |  |  |  |
| REQ-3.9.2 | 管理员账号需设置复杂度要求（至少8位，包含字母和数字 |  |  | ✔ |  |  |  |
| REQ-3.9.3 | 普通用户只能查看和修改自己的信息 |  |  |  | ✔ |  |  |
| REQ-3.10.1 | 系统不存储真实敏感信息（如身份证号、银行卡号等） |  |  |  | ✔ |  |  |
| REQ-3.10.2 | 所有用户数据仅用于教学演示目的 |  |  |  |  | ✔ |  |
| REQ-3.12.2 | 硬件资源利用需求 | ✔ |  | ✔ |  |  |  |
| REQ-3.13.1 | 数据操作成功率≥99.9% | ✔ |  | ✔ |  |  |  |
| REQ-3.13.2 | 代码注释率≥30% |  |  |  | ✔ |  |  |
| REQ-3.13.3 | 单元测试覆盖率≥80% | ✔ |  | ✔ |  |  |  |
| REQ-3.14.1 | 密码存储需使用SHA-256加密，符合《中华人民共和国网络安全法》要求。 | ✔ |  | ✔ |  |  |  |
| REQ-3.14.2 | 数据库表结构需预留扩展字段（如reserved\_field1），以应对未来新增需求。 |  |  | ✔ |  |  |  |
| REQ-3.16.1 | 数据库紧急备份（通过命令行工具db\_backup.sh）。 |  | ✔ | ✔ |  |  |  |
| REQ-3.16.2 | 首次运行时自动创建数据库表结构（执行init\_db.sql） |  | ✔ | ✔ |  |  |  |
| REQ-3.16.3 | 提供数据回滚功能（基于备份文件，恢复至最近3天的状态） |  | ✔ | ✔ |  |  |  |
| REQ-3.17.1 | 软件故障提示与处理 | ✔ | ✔ | ✔ | ✔ |  |  |
| REQ-3.17.2 | 硬件故障提示与处理 | ✔ | ✔ | ✔ | ✔ |  |  |
| REQ-3.18+.+ | 人员，后勤，包装等需求 | ✔ | ✔ | ✔ |  |  |  |

# 5、需求可追踪性

### a. 从CSCI需求到系统需求的追踪性

| **CSCI需求ID** | **对应系统需求ID** | **追踪关系说明** | **注释/例外说明** |
| --- | --- | --- | --- |
| CSCI\_REQ\_001 | SYS\_REQ\_010 | 实现用户身份验证功能 | 直接追踪至系统安全需求 |
| CSCI\_REQ\_002 | SYS\_REQ\_020 | 支持校园卡余额查询 | 直接追踪至用户服务需求 |
| CSCI\_REQ\_003 | SYS\_REQ\_NA\_001 | CSCI间支付接口协议 | 设计决策需求（模块化架构设计章节X.Y） |
| CSCI\_REQ\_004 | SYS\_REQ\_030 | 交易记录存储与审计 | 直接追踪至系统合规性需求 |

### b. 从CSCI需求到系统需求的追踪性

| **系统需求ID** | **对应CSCI需求ID** | **实现方式** | **覆盖状态** |
| --- | --- | --- | --- |
| SYS\_REQ\_010 | CSCI\_REQ\_001 | 用户认证模块 | 完全覆盖 |
| SYS\_REQ\_020 | CSCI\_REQ\_002,CSCI\_REQ\_005 | 余额查询与支付服务 | 部分覆盖（需结合IRS\_001接口） |
| SYS\_REQ\_030 | CSCI\_REQ\_004 | 数据库审计功能 | 完全覆盖 |
| SYS\_REQ\_040 | IRS\_REQ\_002 | 第三方消费系统接口 | 通过IRS引用 |

# 6 尚未解决的问题

## 6.1 技术性遗留问题

1. **第三方支付接口集成**  
   由于教学项目范围限制，系统暂未实现与微信/支付宝的支付网关对接，消费充值功能采用模拟数据替代，需在后续商用版本中补充开发。
2. **大规模数据迁移方案**  
   当前仅支持手动导入500条以内的测试数据，尚未制定从旧系统（纸质记录）到新系统的自动化迁移策略，需开发专用ETL工具。
3. **高并发场景验证**  
   系统未模拟开学季万人级并发访问（如选课期间的集中查询），需在正式部署前进行分布式压力测试。

## 6.2 合规性待完善项

1. **多语言支持缺失**  
   当前界面仅实现中文显示，不符合《智慧校园总体框架》5.2.3条国际化要求。
2. **隐私政策未嵌入**  
   需补充《个人信息保护法》要求的隐私声明弹窗。

## 6.3 运营准备缺口

1. **用户培训体系**  
   管理员操作手册未编制（需包含SQLite备份/恢复指南），用户操作视频亦没有制作（预计时长15分钟）。

# 7 注解

## 7.1 术语定义

| **术语** | **定义** |
| --- | --- |
| ACID事务 | 数据库操作的原子性(Atomicity)、一致性(Consistency)、隔离性(Isolation)、持久性(Durability)特性 |
| 信号槽机制 | Qt框架中对象间通信的松耦合方式，通过connect()函数建立信号与槽的关系 |

## 7.2 缩略语清单

| **缩略语** | **全称** | **说明** |
| --- | --- | --- |
| CCMS | Campus Card Management System | 校园卡管理系统 |
| ETL | Extract, Transform, Load | 数据抽取、转换和加载工具 |
| RBAC | Role-Based Access Control | 基于角色的访问控制 |

## 7.3 技术原理说明

### SQLite并发控制

采用文件锁机制实现多进程并发访问，需注意：

* 写操作独占数据库（通过BEGIN EXCLUSIVE TRANSACTION实现）
* 读操作共享访问（默认BEGIN DEFERRED模式）

### Qt模型/视图架构

* 使用QSqlTableModel实现数据库表与GUI组件的自动同步
* 通过QDataWidgetMapper绑定界面控件与数据模型

# 附录

## 附录A 引用文件补充说明

* **FAR文档**：即《校园卡管理系统可行性分析报告》，包含项目背景、技术可行性、经济可行性等核心分析内容。
* **《智慧校园总体框架》**：教育部发布的智慧校园建设指导文件，涉及系统国际化、数据互通性等要求。
* **《中华人民共和国网络安全法》**：规范密码存储、数据传输等安全要求的法律依据。