**智能自助办证借还管理系统**

**概要设计说明书**

山东亿佰智能科技股份有限公司

2020年7月

文档修订记录

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **序号** | **版本号** | **修改时间** | **修改人** | **审核人** | **批准人** | **备注** |
| 1 | V1.0 | 2020/7/11 | 秦丕文 | 高翔 | 李刚 | 创建 |
| 2 | V1.1 | 2020/7/13 | 秦丕文 | 高翔 | 李刚 | 根据评审内容调整。 |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

目录

[1 引言 4](#_Toc12670)

[1.1 编写目的 4](#_Toc11269)

[1.2 参考资料 4](#_Toc4745)

[1.3 约束 5](#_Toc8866)

[1.4 假设、依赖及策略 5](#_Toc5526)

[2 总体设计 6](#_Toc18544)

[2.1 设计目标 6](#_Toc27764)

[2.1.1 总体原则 6](#_Toc27551)

[2.1.2 实用性和先进性 7](#_Toc2476)

[2.1.3 标准化、开放性、兼容性 7](#_Toc1351)

[2.1.4 高可靠性、稳定性 7](#_Toc22495)

[2.1.5 易用性 7](#_Toc31689)

[2.1.6 灵活性和可扩展性 7](#_Toc14631)

[2.1.7 经济性和投资保护 8](#_Toc20777)

[2.2 运行环境 8](#_Toc30338)

[1. 开发环境 8](#_Toc25936)

[2. 客户端 8](#_Toc18314)

[2.3 性能设计 9](#_Toc29533)

[1、响应时间 9](#_Toc7439)

[2、并发用户数 9](#_Toc27240)

[3 系统设计 10](#_Toc27974)

[3.1总体结构框架 10](#_Toc1374)

[3.1.1 六个层次结构 10](#_Toc30380)

[3.1.2 三大保障体系 11](#_Toc25326)

[3.2 关键技术 11](#_Toc22227)

[3.2.1 单体系统架构 11](#_Toc9623)

[3.2.2 微服务架构 12](#_Toc31911)

[3.2.3 开发框架 14](#_Toc9551)

[4 系统功能设计 15](#_Toc18283)

[4.1自助办卡 15](#_Toc5733)

[4.2 自助退款 16](#_Toc30374)

[3.2自助借还 16](#_Toc21893)

[3.2.1 自助借书 17](#_Toc17528)

[3.2.2 自助还书 18](#_Toc30108)

[3.2.3 自助续借 18](#_Toc17956)

[3.2.4 自助预借 19](#_Toc20193)

[3.2.5 借阅信息 20](#_Toc4893)

[3.3检索查询 20](#_Toc706)

[3.3.1借阅信息 20](#_Toc14261)

[3.3.2 位置查询 21](#_Toc8803)

[3.3.3 电子书 22](#_Toc17977)

[3.3.4 图书状态 22](#_Toc28102)

[5 数据结构设计 24](#_Toc16864)

[5.1 数据内容 24](#_Toc6530)

[逻辑结构设计 24](#_Toc14902)

[5.2 数据资源架构设计 24](#_Toc14077)

[5.3 设计规范 25](#_Toc25327)

[5.3.1 数据库规范 25](#_Toc31604)

[5.3.2 数据库表规范 25](#_Toc4935)

[5.3.3 表字段规范 26](#_Toc17503)

[5.3.4 索引规范 26](#_Toc12625)

[5.3.5 存储过程规范 26](#_Toc27643)

[5.3.6 视图规范 27](#_Toc28909)

[5.3.7 触发器规范 27](#_Toc22259)

[5.4 逻辑结构设计 28](#_Toc19260)

[5.5 物理结构设计 29](#_Toc11394)

[5.5.1 数据库表汇总 29](#_Toc31424)

[5.5.2 数据库列表详细 30](#_Toc29832)

[6 接口设计 38](#_Toc27031)

[6.1 接口内容 38](#_Toc29544)

[6.2 接口设计原则 38](#_Toc10860)

[6.3 接口设计分类 38](#_Toc6566)

[6.4 接口实现方式 38](#_Toc6678)

[6.4.1 本系统提供接口、外系统调用 39](#_Toc19868)

[6.4.2 外系统提供接口、本系统调用 39](#_Toc30458)

[6.4.3 外系统开放数据库、本系统调用 39](#_Toc18372)

[6.4.4 本系统提供接口、本系统调用 39](#_Toc25767)

[7 运行设计 40](#_Toc15978)

[8 维护设计 41](#_Toc17339)

[8.1 出错处理设计 41](#_Toc24589)

[8.2 安全保密设计 41](#_Toc4500)

[8.3 维护设计 42](#_Toc16892)

# 引言

## 编写目的

本阶段在系统需求分析的基础上，进行系统概要设计。主要解决了实现本系统需求的程序模块设计问题，包括如何把该系统划分成若干个模块、决定各个模块之间的接口、模块之间传递的信息，以及数据结构、模块结构的设计等。本文将对在本阶段中对系统所做的所有概要设计进行详细的说明。

在详细设计中，程序设计员可参考概要设计说明书，在概要设计对系统所做的模块结构设计的基础上，对系统进行详细设计。在软件测试以及软件维护阶段也可参考说明书，以便与了解在概要设计过程中所完成的各模块设计结构，或在修改时找出在本阶段设计的不足或错误。

## 参考资料

本项目建设的依据性材料包括：

1）《山东省人民政府办公厅关于印发山东省政务信息系统整合共享实施方案的通知》鲁政办发〔2017〕75号

2）《推进“互联网+”便捷交通 促进智能书房发展的实施方案》发改基础〔2016〕1681号

3）《推进智慧书房发展行动计划（2017-2020年）的通知》交办规划〔2017〕11号

4）《关于运用大数据加强对市场主体服务和监管的若干意见》国办发〔2015〕51号

## 约束

1、数据库及操作系统约束：本项目系统因作为数据汇聚的综合平台，采用CentOS操作系统。规划计划、安全生产、综合统计使用SQL Server数据库、windows server操作系统。

2、HTML代码将遵照HTML5标准。

## 假设、依赖及策略

本项目是否能够成功实施完成，并保障系统正常上线时间及系统功能的完整性，主要取决议以下条件：

策略：根据业主要求，全力做好配合及沟通，提前做好各类业务数据整合方案，保障项目顺利实施。

1. 为了保证项目的工期和质量，在现有业务系统数据接入过程中，需要相关业务系统开发单位的积极配合，如果配合不积极，协调不到位，会直接影响项目的进度。

策略：及时跟进与各系统开发单位的沟通，做好工作安排和辅助配合。

# 总体设计

## 设计目标

根据项目单位具体需求，结合智能自助办证借还系统最新的发展趋势、先进的技术手段和系统设计方法，充分利用已投入使用的系统，按照“抓应用、强体系、提能力、上水平”的原则，狠抓信息化成果推广和深化应用，积极适应智慧书房、图书馆工作新形势、新任务的要求，坚持“深化建设”与“深度应用”并重，以需求为导向，以应用为核心，坚持统一技术体制，充分利用现代化计算机网络技术、信息通信技术、数据整合技术，逐步建立适应智慧书房管理发展的现代化自助办证平台，实现各系统的高度集成、高效融合及各类信息的精确显示、高度共享，提升用户体验。

* + 1. **总体原则**

1、统一总体设计：遵循“统一设计、统一规划、统一实施，统一建设”的原则，加强规范化、标准化，确保各建设项目的实施过程符合总体架构设计。

2、符合标准要求：符合各项信息化建设要求，确保建设过程标准规范。

3、整合现有资源：充分整合已建设信息系统，充分利用各子系统建设成果，不浪费资源、不重复投资。

4、引用先进理念：引入云计算、大数据等先进理念，用不断创新的信息技术，提升智慧书房数字化水平，提高智能自助办卡业务能力。

* + 1. **实用性和先进性**

采用先进成熟的技术满足用户使用需求，兼顾其他相关的管理需求，使整个系统在相当一段时期内保持技术的先进性，而不至于落后，以适应现代科技和信息化技术快速发展的大趋势和大方向。同时，用户在短时间内不必再为系统的升级而开销，有效的保护了用户投资。在保证系统先进性的前提下，也要充分考虑系统的实用性，毕竟只先进不实用的系统不是用户真正需要的，最大程度的满足用户建设需要、贴合用户使用需求，才能满足用户的实用性要求。

* + 1. **标准化、开放性、兼容性**

选择标准、开放的技术和应用标准，软件协议上真正实现开放，同时基于开放式标准，坚持统一规范的原则，实现标准化、模块化，从而为未来系统的开放、兼容、发展奠定基础。

* + 1. **高可靠性、稳定性**

业务系统的运行，高可靠性是第一位的。要对系统架构进行高可靠性的设计和建设。采用冗余、分布式、集群、热备等相关的技术手段，为整个系统的稳定运行保驾护航。

* + 1. **易用性**

系统建成后是否能让用户使用人员很快上手，这直接关系到系统的使用效率。因此，在系统建设中，必须坚持系统的易用性原则。在系统的操作和控制方式上，尽可能通过技术手段，使得操作人员可以快速掌握系统的使用。

* + 1. **灵活性和可扩展性**

考虑到未来业务的调整及快速发展，系统结构要层次化、模块化，易于未来应用的扩展。现代化的系统应该是一个不断发展、与时俱进的系统，所以它必须具有良好的灵活性和可扩展性，能够根据用户不断发展的业务需要，方便灵活的进行扩展和升级，并提供技术升级、设备更新的灵活性。

* + 1. **经济性和投资保护**

应以较高的性能价格比构建系统，无论是技术的选择上还是系统的构建上，使资金的产出投入比达到最大值。能以较低的成本、较少的人员投入来维持系统的正常运转，以体现系统的高效能与高效益。并且在不影响系统改造目标的前提下，尽可能保留和延长现有系统运行，以充分利用以往资金与技术方面的投入。

## 运行环境

* + 1. **开发环境**

|  |  |
| --- | --- |
| 名称 | 描述 |
| 硬件 | D1801、D6520、HCICRF、RFID、ZKLiveFace |
| 网络 | 办公室局域网 |
| 软件 | Window10、IE10以上、Google |
| 数据库 | sql server |
| 备注 | 开发涉及的系统软件以开发人员实际使用为准 |

* + 1. **客户端**

客户端操作系统：Windows 7及以上版本。

客户端浏览器：支持用户通过谷歌、360安全浏览器、IE10以上等进行访问。

## 性能设计

在服务商提供的服务器、存储、网络等运行环境良好的情况下，满足一下要求：

* + 1. **响应时间**

系统应具有快速响应的特性，用户打开界面和提交事务的平均响应时间应低于2秒。用户进行在线实时查询业务操作的数据处理时间应低于5秒。

* + 1. **并发用户数**

高峰并发用户数50人。

# 系统设计

## 总体结构框架

智能自助办证借还管理系统设采用顶层设计思想，使用先进技术手段，按照层级规划为基础支撑层、应用支持层、数据资源层、综合应用层，以及纵向贯穿的信息安全、标准规范及运行保障体系。

* + 1. **六个层次结构**
* 基础支撑层

——计算及存储资源：依托“一片云”的提供本次系统的计算资源池和存储资源池。

——网络系统资源，依托“两张网”的电子政务外网进行数据传输。

* 应用支撑层

根据现有的数据库、中间件、操作系统、备份系统，为图书馆管理服务子系统的各项应用提供支持。

* 数据资源层

在对智能办证借还系统应用现状进行深入分析的基础上，明确信息分类、统一数据标准、确定信息来源和获取方式，按照路网静态（基础）数据、路网动态数据库、业务管理数据库、应用主题数据库等四大类别进行数据规划和数据资源建设，实现新建以及整合的业务系统数据采集和集中存储。

* 综合应用层

通过智能自助办证借还系统促进业务流程整合和简化，提升行业工作效率。

* 用户层

根据业务所属内容不同，通过权限划分，区分读者管理员。

* 信息展现层

通过电脑端，实现信息的查询和展示。

* + 1. **三大保障体系**

智能自助办证借还管理系统由3大保障体系构成：

* 信息安全管理体系

本系统提供防火墙、网络入侵检测，数据库访问控制、数据加解密，操作审计、身份认证、消息传递加解密、服务访问控制、身份认证、功能访问控制、运行日志等安全措施，保证系统资源、数据、消息、程序访问的多方面的安全保障。

* 数据资源整合标准管理体系

制定统一的数据标准规范、采集规范、业务管理规范、数据共享规范业务标准规范、技术标准规范等，制定数据资源目录。

* 建设运行保障体系

建立智能自助办证借还系统的建设与运营保障体系，落实机构、人员和资金、制定一套系统建设和长效运营机制。

## 关键技术

* + 1. **单体系统架构**

本方案采用单体应用服务架构，采用SpringBoot开源项目，其主要特点：

独立应用: 每一个 SpringBoot App 都是独立存在的个体应用，可以是一个 Jar 文件，也可以是一个war 文件，只需要基本 Java 运行环境即可满足部署要求，SpringBoot App 内部嵌入了 Tomcat，从而无需部署 Web 容器。Maven 项目:SpringBoot 项目实际上是一个Maven 项目，在第三方插件的使用上完全可以通过依赖 maven 或者 Gradle 来实现，大大简化部署量。Spring 配置自动化:基于Spring4.X，配置工作可以完全通过 Java 注释的方式来实现，从而实现自动化配置。生产就绪功能:指标、健康检查、外部配置。去 XML 化:可以完全取消配置文件。

* + 1. **微服务架构**

本方案采用微服务架构，采用SpringCloud开源项目族群，实现服务注册与发现、智能路由、配置中心、熔断等特性。微服务架构解决了传统系统庞大、强耦合的问题。

微服务设计模式是将应用分解为原子化的、互相连接的微服务。微服务一般完成某个特定的功能，比如专家库、法律法规库、监督检查等功能。每一个微服务都是微型六角形应用，都有自己的业务逻辑、数据库、缓存、权限认证和访问控制等后台资源。微服务还会发布API给其它微服务和应用客户端使用。

#### 本项目使用微服务架构总体说明

根据本项目的实际特点，综合应用系统作为整合应用的系统，支撑业务系统因逻辑清晰，采用轻量级的系统架构体系。

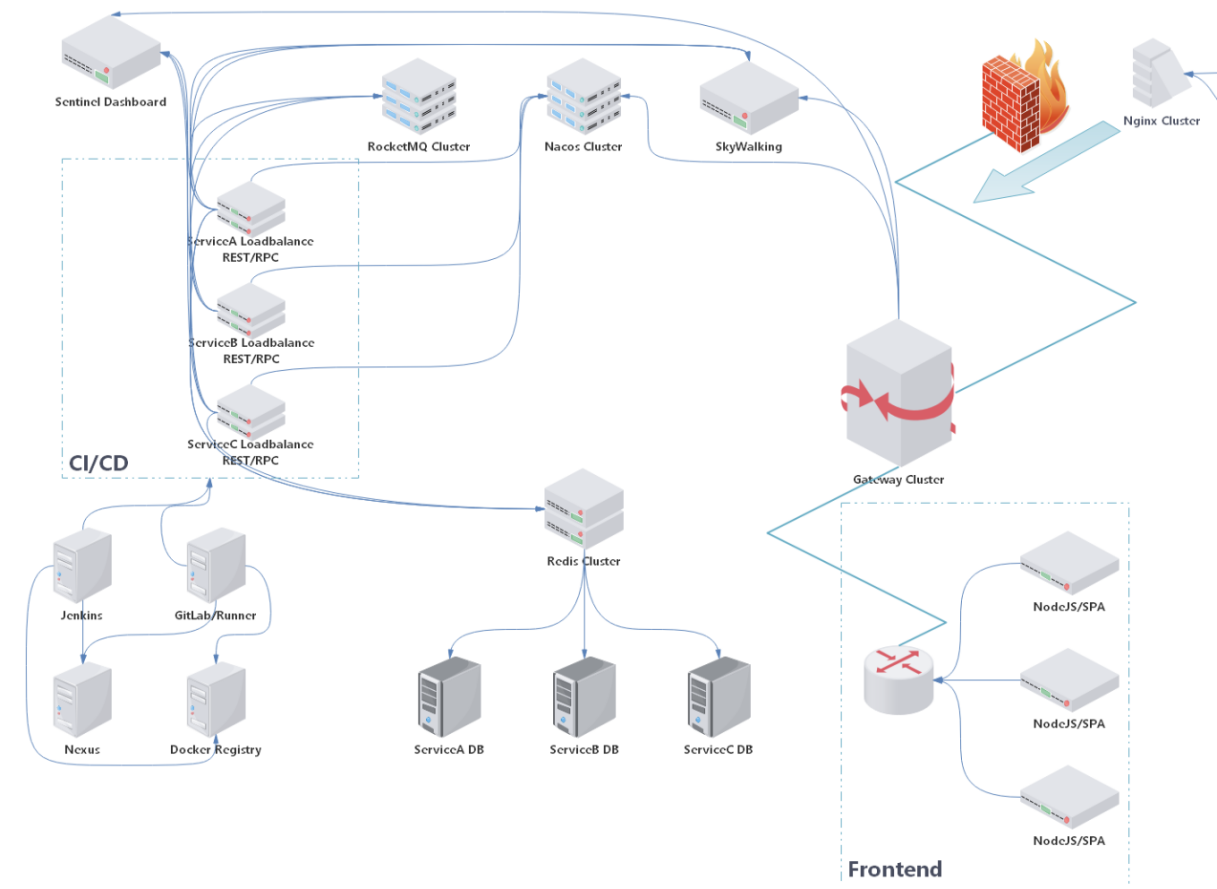
使用的后台主要技术栈：

* 核心框架：Spring Boot + Spring Cloud Alibaba
* ORM 框架：mybatis-plus 简化 MyBatis 开发
* 数据库连接池：Alibaba Druid
* 数据库缓存：Redis Sentinel
* 消息中间件：RocketMQ
* 接口文档引擎：Swagger2 RESTful 风格 API 文档生成
* 全文检索引擎：ElasticSearch
* 反向代理负载均衡：Nginx

该系统架构遵循的两个标准：

* 提高敏捷性：及时响应业务需求
* 降低成本：降低增加项目、客户或业务方案的成本

拓扑结构：



#### 服务规划

1. SpringBoot

|  |  |  |
| --- | --- | --- |
| 序号 | 服务名称 | 服务说明 |
| 1 | SkyWalking | 链路追踪 |
| 2 | RocketMQ | 消息队列 |
| 5 | Redis | 数据缓存 |
| 6 | Nginx | 反向代理/负载均衡 |

* + 1. **开发框架**

|  |  |  |
| --- | --- | --- |
| 系统名称 | 开发框架 | 备注 |
| 智能自助办证借还管理系统 | 单体服务架构、JAVA技术框架 |  |

# 系统功能设计

## 自助办证

* + 1. **自助办卡**

读者信息资料收集是系统核心的模块，为后续智能自助办证借还管理业务提供数据基础，也是同其他业务系统关联最紧密的环节。

自助办卡主要包含以下两方面：

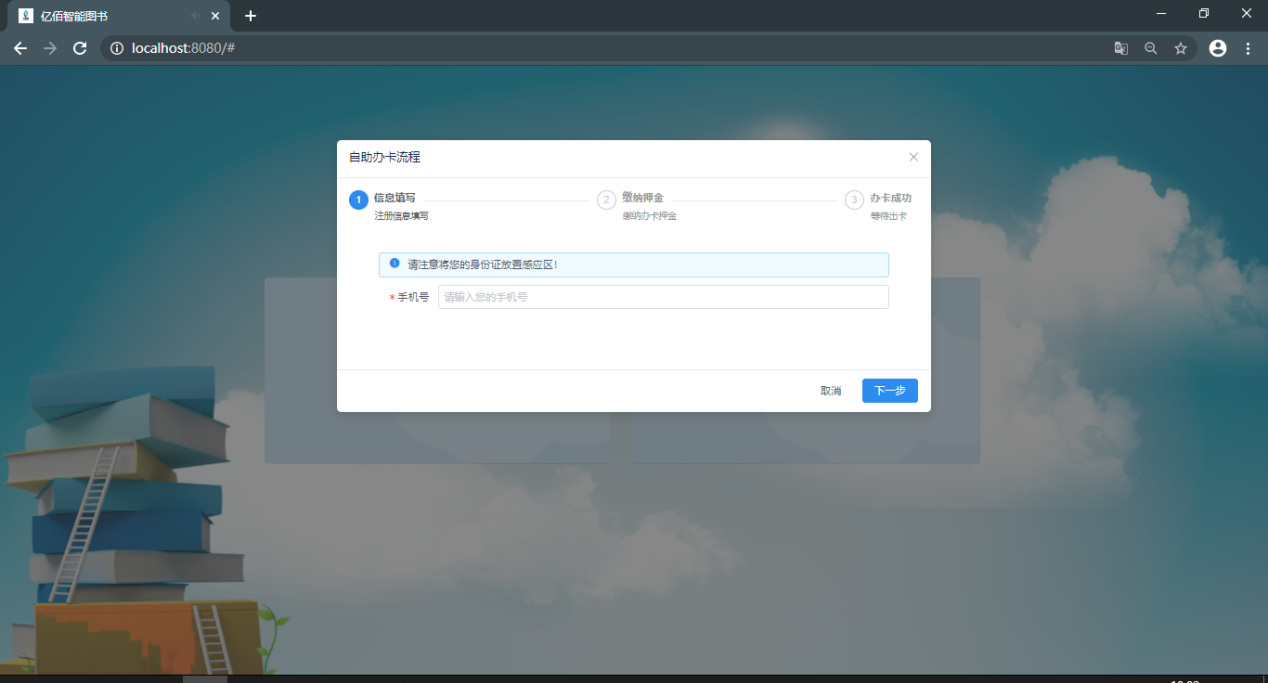
一是身份信息读取，通过身份证多合一IC卡读取，存储。

二是押金缴纳，客户可以根据是否需要缴纳押金来进行设置，支付方式为线上支付，支付宝或者是微信支付。

三办卡成功，通过D6520设备将信息写入卡中，在通过D1801设备将卡吐出。

四是同业务系统进行对接，产生的用户资料信息。基于事先建立的业务资料分类对应关系，业务系统与图书系统进行接口对接时，按照已建立完成的图书分类规则对应关系，对需要上传的用户信息进行自动分类映射转换，同时上传用户信息，完成不同系统的初步对接。

原界面模型：



* + 1. **自助退款**

当用户不需要继续使用读者卡的时候，可以通过本功能进行在线急速退款，退款方式为办卡时所支付的账号，需要的身份信息确认证件为身份证和读者卡。

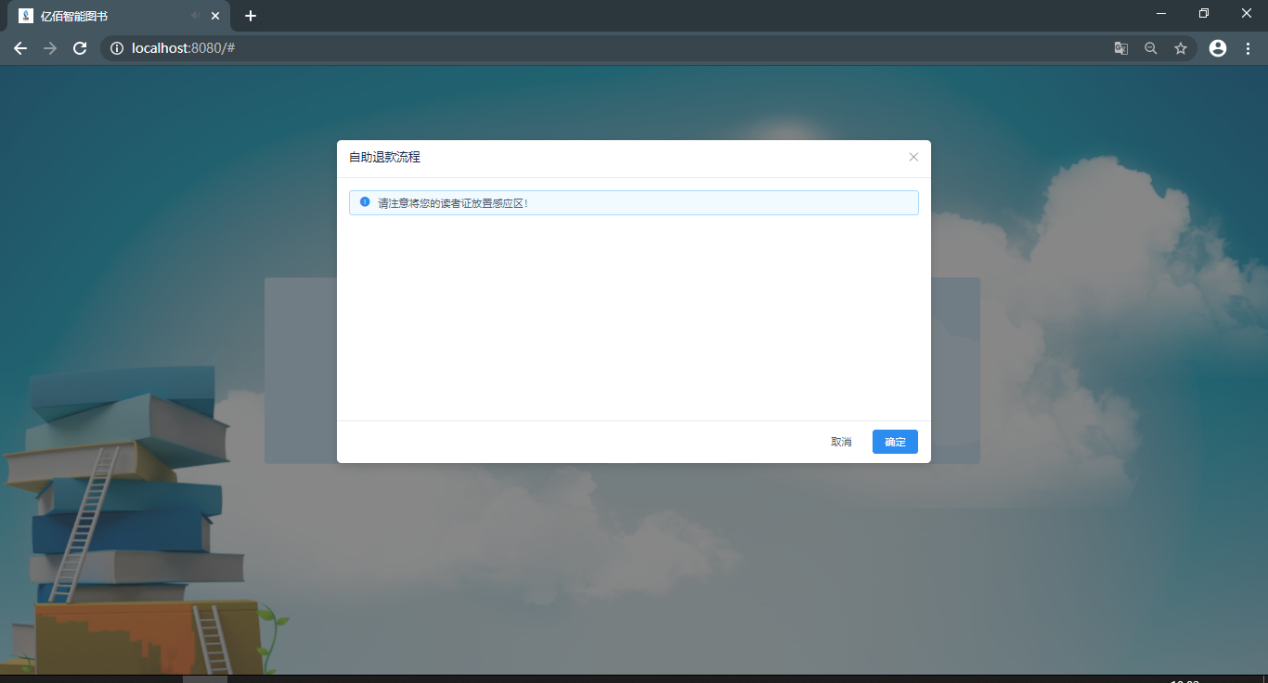
无法退款情况：

有在借阅书籍未归还

有借阅书籍存在违约，损坏，罚款未交等。

注意：本功能只支持在办卡后一年内在线急速退款，如办卡时间超出一年，可以在前台管理员哪里办理退款。

原界面模型：



## 自助借还

自助借还是图书借还系统的核心部分，将其作为一个完整的业务模块集成在自助借还系统中。自助借还的系统功能主要实现图书的借书、还书、续借和预借、借阅信息等。

* + 1. **自助借书**

用户仅需要带读者卡或者身份证通过自助借还书机将需要借阅的书籍和读者卡放置感应区既可以操作。

操作过程

系统对借阅证进行资格审查

审查是否为读者本人（非本人不得外借）；

是本人，审查读者的借阅权限。在以下情况下显示拒借：

读者证无效（即处于验证/挂失/注销/暂停状态）；

读者已过有效期；

读者有未交清罚款；

读者有过期未还文献；

如果审查未通过，则系统通知读者无权借书；

如果审查通过，则系统使用RFID读取图书的条码号；

系统读取图书的条码号后可能出现以下拒借的情况：

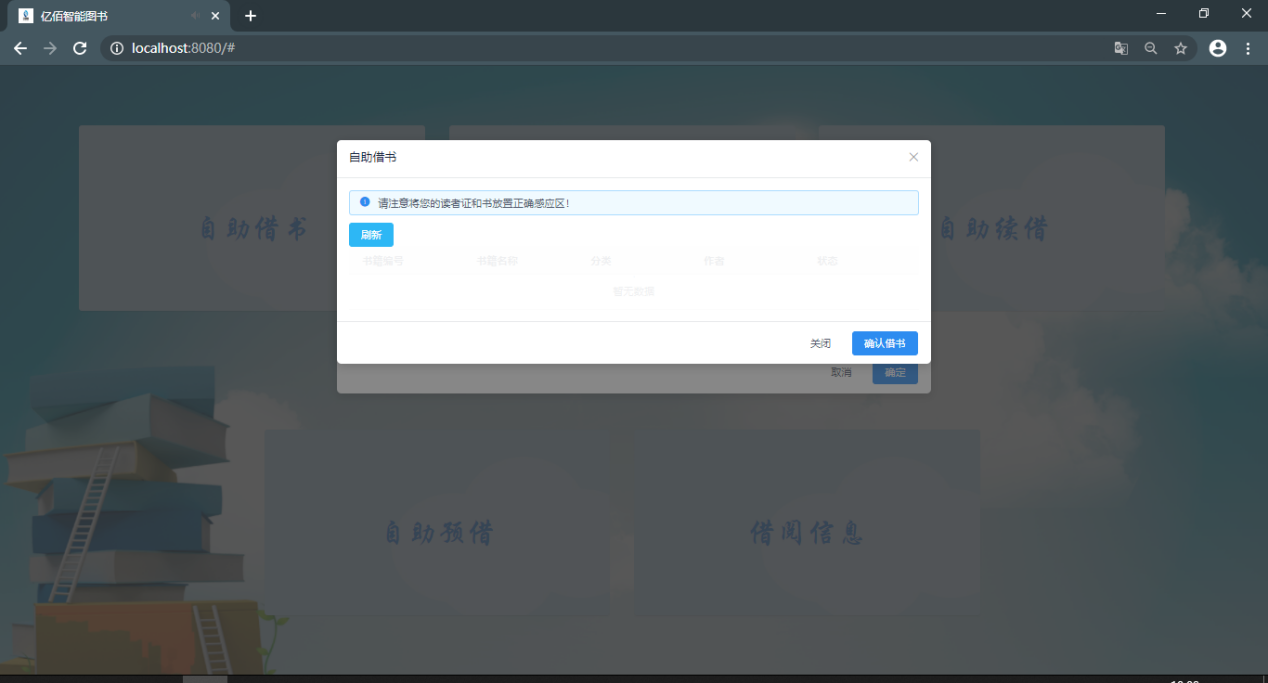
馆藏库无指定书目记录

本书是预约借书，但当前文献实际预阅者与借阅者不符；

读者总借数已满。

以上情况通过则进行具体的借书处理，最后借书成功。

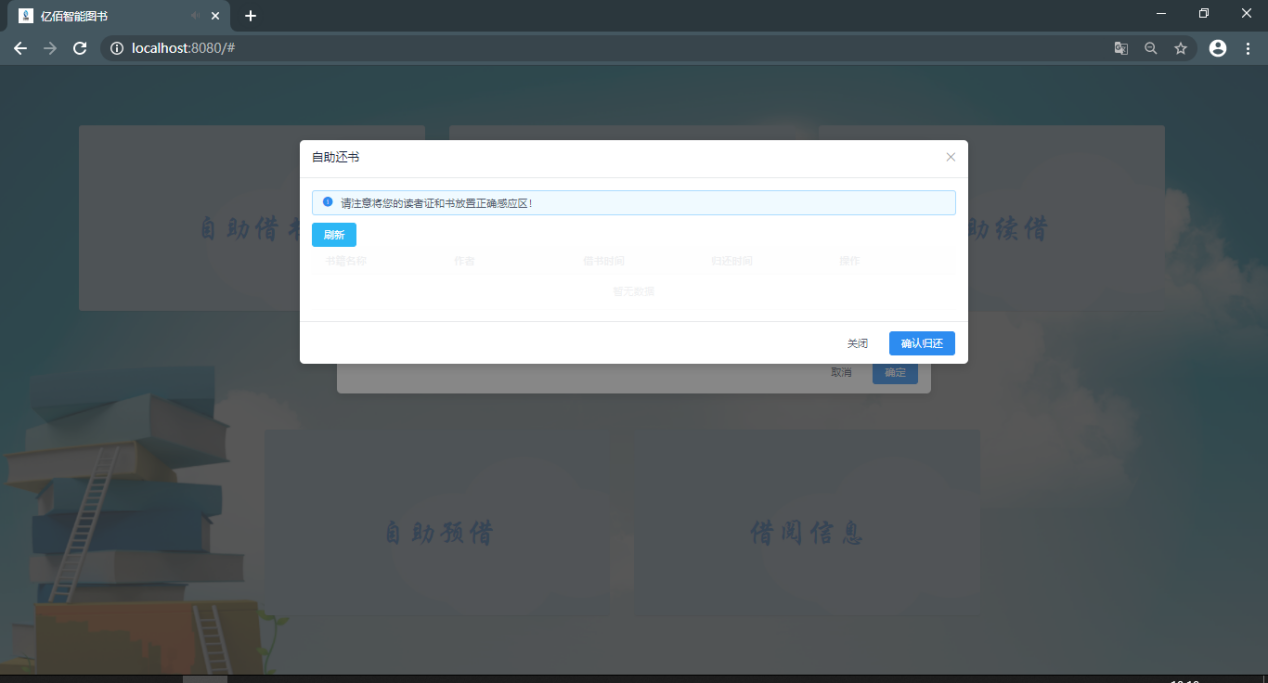
原界面模型：



* + 1. **自助还书**

用户仅需要带借阅的书籍通过自助借还书机将需要归还的书籍放置感应区按照操作步骤既可以还书完成。

原界面模型：

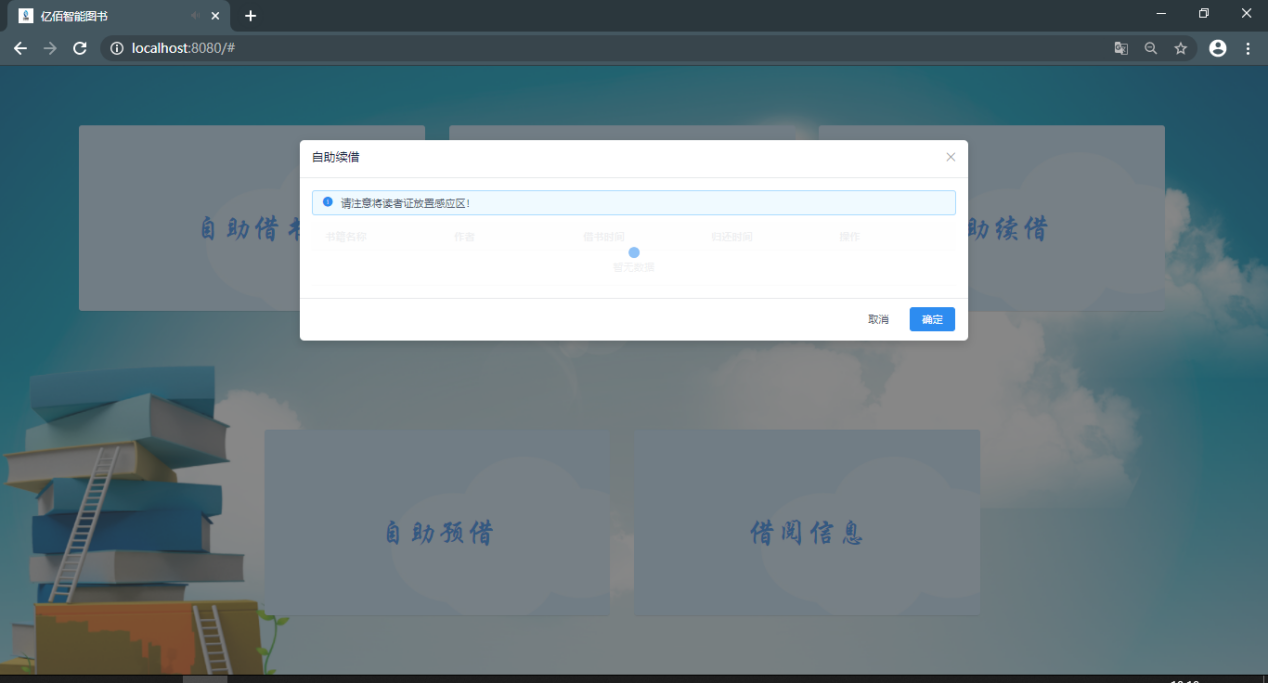


* + 1. **自助续借**

用户自助续借功能，用户仅需要带读者证即可在自助借还书机查询到正在借阅的书籍，用户可以根据自己选择借阅的时间续借。

注意：如无正在借阅的书籍，应先去借阅在来此续借。

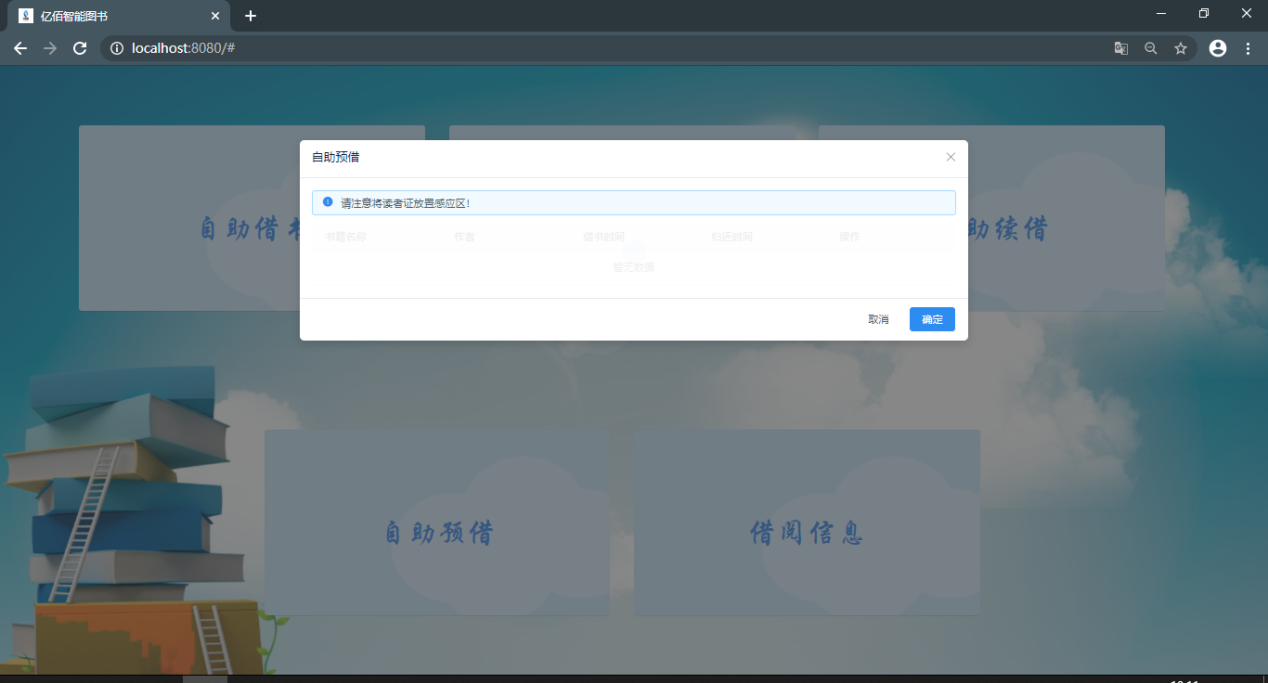
原界面模型：



* + 1. **自助预借**

为了方便读者不必亲自跑来图书馆就可以借到自己喜欢的书籍，只需登录网上图书馆，查询想要借阅的图书，预借登记后，输入手机号信息，选择要取书的地点，等待通知消息取书即可。

原界面模型：



* + 1. **借阅信息**

借阅信息是指用户通过自助借还书机查询到的自己所有借阅信息记录，可以帮助用户管理自己借阅的图书信息。

原界面模型：

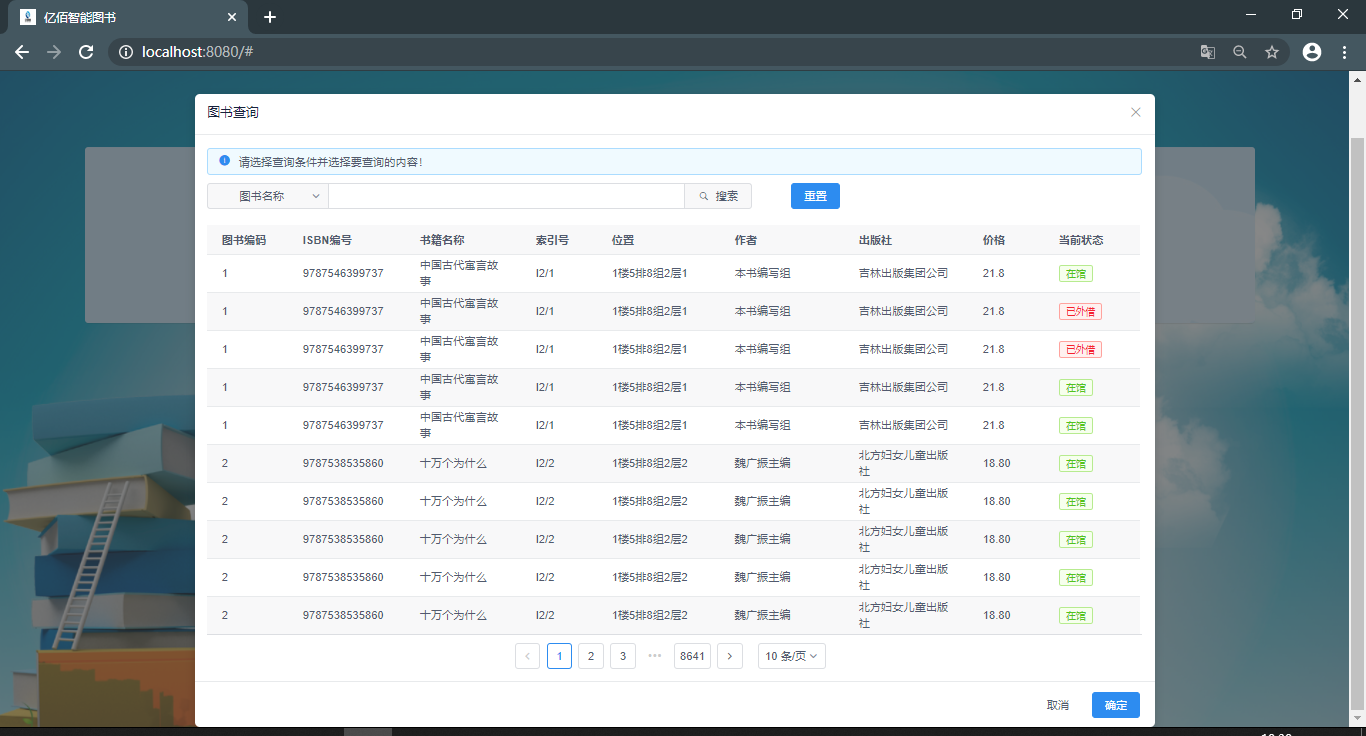


## 检索查询

* + 1. **借阅信息**

查询图书馆的所有图书信息状态，可以根绝不同的条件进行模糊查询，例如：图书名称、图书编号、作者名称、出版社、ISBN编号等。

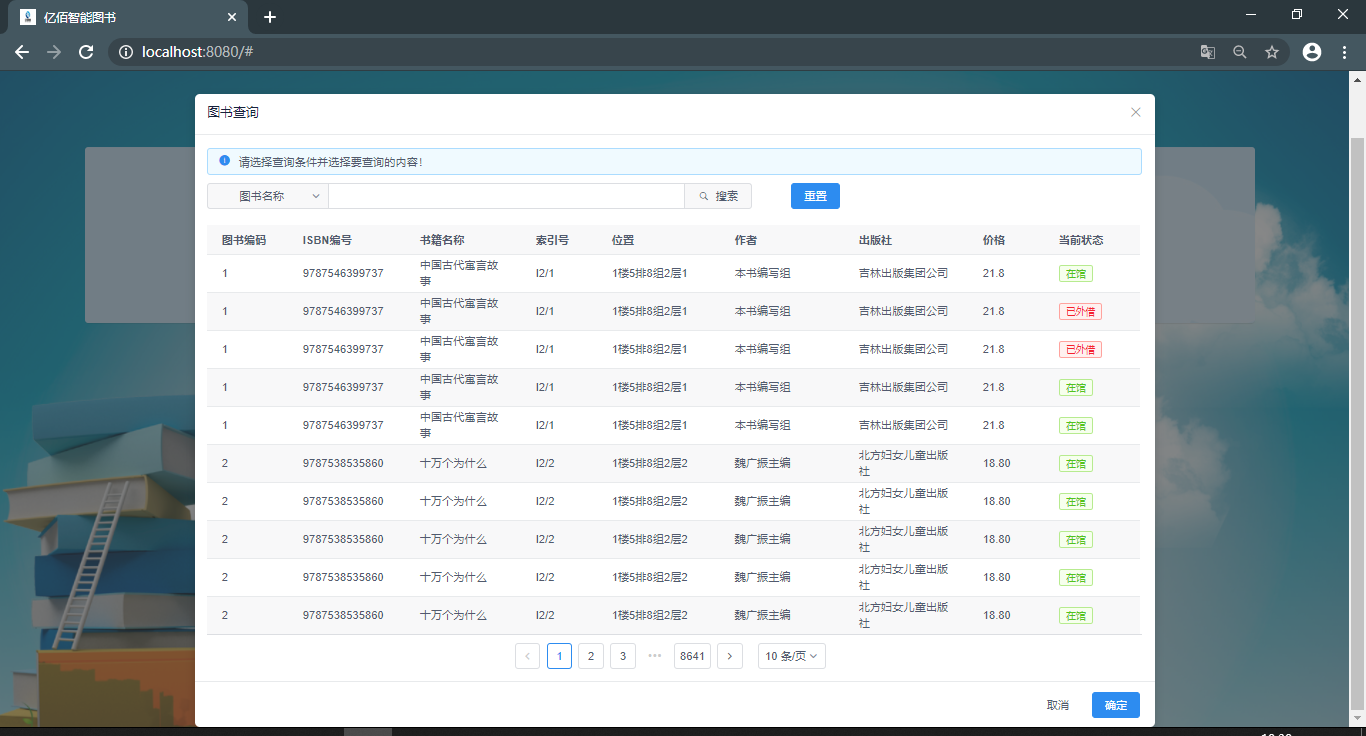
原界面模型：



* + 1. **位置查询**

通过图书的名称或者图书编号可以快速检索出图书信息，图书信息存在图书准确的位置信息。

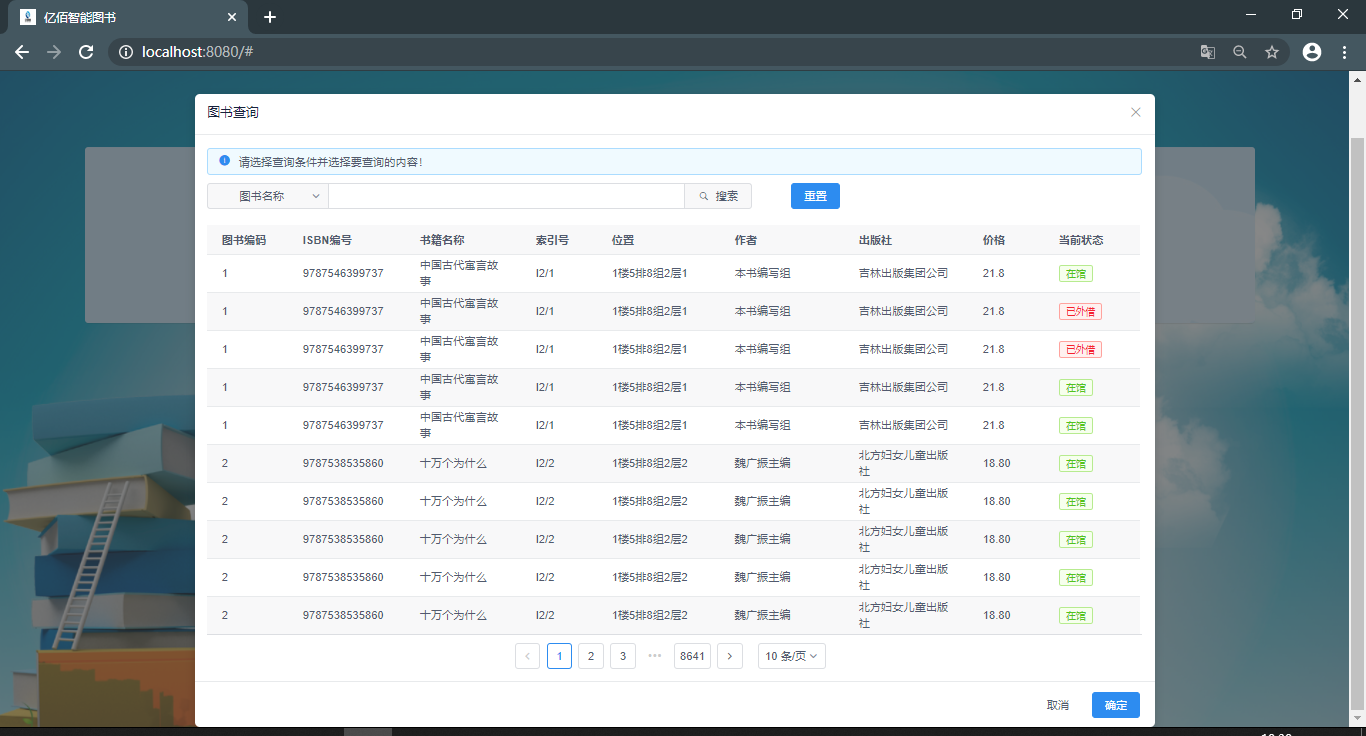
原界面模型：



* + 1. **电子书**

用户可以根据想搜索的图书名称查询电子版本的图书信息，可以支持在线阅读，无须借阅等。

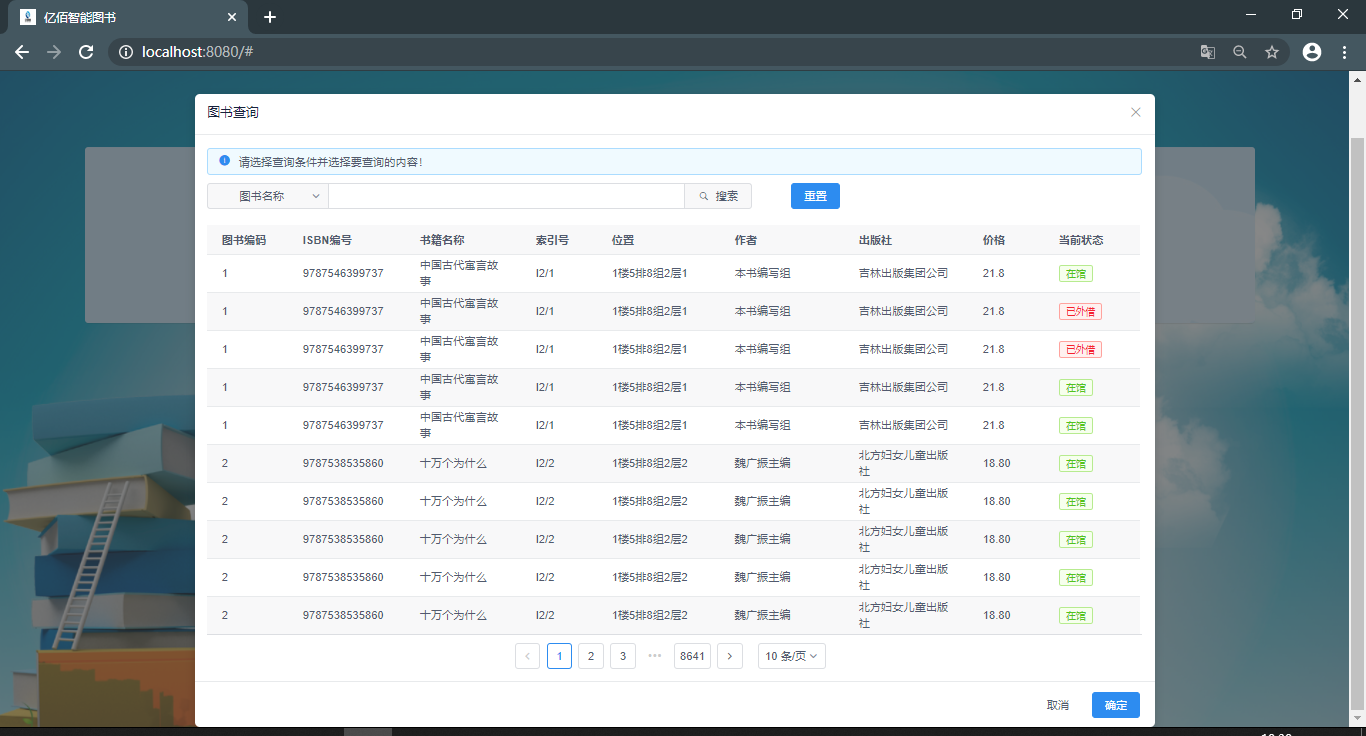
原界面模型：



* + 1. **图书状态**

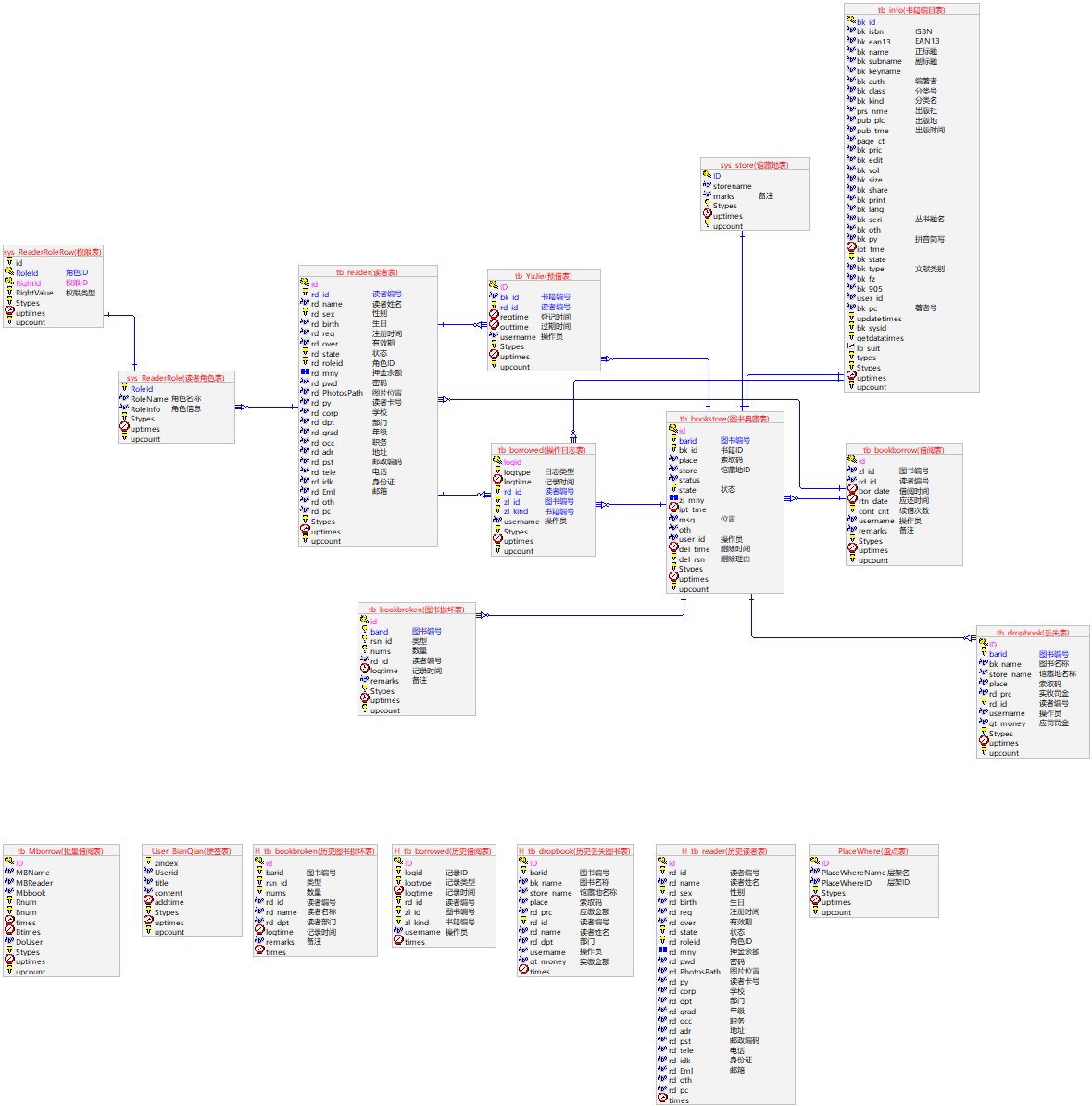
用户根据图书名称等检索条件可以查询当前所有图书在图书馆的状态信息。

原界面模型：



# 数据结构设计

## 逻辑结构设计



## 物理结构设计

系统内所使用的数据结构中的数据项，每张数据表都有自增主键，不能为空，大多数项不能为空，只有少数项可以为空，为了提高查找速度，分别对某些数据结构创建了索引文件，并提供不同的权限，对数据进行操作。

## 数据库表设计

以下是智能自助办证借还管理系统数据表的详细概要描述，根据设计需求，本平台共涉及21张表，详见《数据库设计说明书》

|  |  |
| --- | --- |
| 表名 | 中文名 |
| user\_bianQian | 用户便签表 |
| pay\_order | 支付订单表 |
| sys\_readerrole | 读者角色表 |
| sys\_readerrolerow | 读者角色权限表 |
| sys\_users | 系统管理员表 |
| sys\_config | 系统配置表 |
| sys\_grouprights | 系统角色表 |
| sys\_store | 馆藏地表 |
| sys\_trees | 系统菜单表 |
| tb\_brokenrole | 图书损坏类型表 |
| tb\_Mborrow | 批量借阅表 |
| tb\_YuJie | 图书预借表 |
| tb\_bookborrow | 图书借阅表 |
| tb\_bookbroken | 图书损坏表 |
| tb\_bookstore | 图书典藏表 |
| tb\_borrowed | 操作日志表 |
| tb\_dropbook | 图书丢失表 |
| tb\_info | 书籍编目表 |
| tb\_pub | 图书出版社表 |
| tb\_reader | 读者表 |
| tb\_type | 图书类别表 |

## 数据资源架构设计

采用“数据集中、服务集中”的设计思想，使用大数据等技术手段，数据平台包括数据集成开发、数据治理、数据存储、数据处理、数据资源、运维管理、数据访问等模块。数据总体架构如下：

数据存储，关系数据库、实现混合型的数据存储支撑能力。对于业务应用，优先选用应用适配的关系数据库；对于数据资源建设和管理，选用大数据手段，实现各类各型数据的存储支持。

数据处理提供分布式批处理计算、内存计算、流计算处理等，满足不同场景的数据分析、处理和加工要求。

数据资源，结合应用和数据管理等要求，规划基础、专题、应用等类别数据资源。基础类数据。

数据访问提供统一的数据访问支撑，包括数据库接口、数据服务接口等内容。

## 设计规范

在本系统中，数据库的设计采用Visio进行，并且采用面向对象的设计方法，首先进行对象实体的设计，最后将对象持久化到数据库中，所有的表和表之间的关联(ER图)都采用标准的Visio设计工具进行，这样能够将整个系统的设计和数据库设计有机的结合起来。

* + 1. **数据库规范**
* 数据库命名：SR\_业务(汉语拼音的大写首字母,最长4位)，如：规划计划库，命名为SR\_GHJH。
  + 1. **数据库表规范**
* 数据库表名应该有意义，并且易于理解。
* 表命名：T\_业务\_模块\_表名(汉语拼音的大写首字母，和汉字解释字数一样)，例子：规划计划库中，年度计划模块中的项目基本信息表，命名为：T\_GHJH\_NDJH\_XMJBXX。
* 表名不可以太长，最好不要超过22个字母。
  + 1. **表字段规范**
* 数据库表字段采用统一小写字母。
* 数据库表字段应该是有意义而且易于理解的，使用汉语拼音首字母缩写，对于相同的，原则使用全拼，过长适当删除字母。例子：项目名称，命名为：xmmc。
* 不要使用计算机术语，尽量使用面向业务的术语。
* 字段必须填写描述信息（注释）。
* 每个表必须包括：创建人、创建时间、更新人、更新时间。
  + 1. **索引规范**
* 索引名称和数据库列名需保持一致。
* 如果字段事实上是与其它表的关键字相关联而未设计为外键引用，需建索引。
* 如果字段与其它表的字段相关联，需建索引。
* 如果字段需做模糊查询之外的条件查询，需建索引。
  + 1. **存储过程规范**
* 存储过程的命名请遵循以下命名规范：USP\_系统模块缩写（与表前缀类似）\_功能标识代表存储过程操作的主要表名（不带前缀）或功能的英文单词或英文单词缩写。
* 存储过程设计规范

(1)目的：说明此存储过程的作用。

(2)作者：首次创建此存储过程的人的姓名。在此请使用中文全名，不允许使用英文简称。

(3)创建日期：创建存储过程时的日期。

(4)修改记录：修改记录需包含修改顺序号、修改者、修改日期、修改原因，修改时不能直接在原来的代码上修改，也不能删除原来的代码，只能先将原来的代码注释掉，再重新增加正确的代码。修改顺序号的形式为：log1，log2，log3。。。，根据修改次数顺序增加，同时在注释掉的原来的代码块和新增的正确代码块前后注明修改顺序号。

(5)对存储过程各参数及变量的中文注解。

* + 1. **视图规范**
* 视图的命名请遵循以下命名规范：UV\_+系统模块缩写（与表前缀类似\_功能标识+代表视图查询的主要表名（不带前缀）或功能的英文单词或英文单词缩写。
* 视图设计规范

(1)目的：说明此视图的作用。

(2)创建者：首次创建此视图的人的姓名。在此请使用中文全名，不允许使用英文简称。

(3)修改者、修改日期、修改原因：如果有人对此视图进行了修改，则必须在此视图的前面加注修改者姓名、修改日期及修改原因。

(4)对视图各参数及变量的中文注解。

* + 1. **触发器规范**
* 触发器的命名请遵循以下命名规范：UTR\_模块\_功能\_标识符。标识符'i'代表Insert触发器、'd'代表Delete触发器、'u'代表Update触发器。
* 触发器设计规范

(1)目的：说明此触发器的作用。

(2)创建者：首次创建此触发器的人的姓名。在此请使用中文全名，不允许使用英文简称。

(3)修改者、修改日期、修改原因：如果有人对此触发器进行了修改，则必须在此触发器的前面加注修改者姓名、修改日期及修改原因。

# 接口设计

## 接口内容

本系统根据外部接口提供的内容，定义本系统所需要的数据结构，通过连接服务，获取数据，将数据转入本系统处理，存储为本系统所需要的数据。

## 接口设计原则

* 使用简单、快捷，通用性好，可靠性高；
* 充分考虑接口所涉及的各个系统的应用扩展情况，能灵活地支撑需求变化；
* 保证接口数据在接口所涉及的各个系统间的一致性；
* 接口数据能够自动导出和导入；
* 在数据交互过程中，应具有传送和接收后的确认过程。

## 接口设计分类

从技术上，本次系统的接口主要分为如下几类：

* Http接口

通过URL格式发布的Http访问接口。

在访问过程中通过普通字符串、JSON等数据来实现参数的传入和结果的返回。

* 数据库表接口

以数据库表访问的方式实现的接口。

## 接口实现方式

根据外部系统的不同情况，在接口实现上会出现不同的方式。

* + 1. **本系统提供接口、外系统调用**

本系统为了发布信息或者希望其他系统提供数据，需要开放一些接口来实现数据的发布与获取。提供接口的方式主要是基于http协议post和get的方式。通过调用本系统提供的接口，可以实现两件事情：

外系统可以获取由本系统提供的数据。

外系统提供数据给本系统。

* + 1. **外系统提供接口、本系统调用**

外系统通过HTTP接口方式提供数据访问接口，本系统通过访问这些接口获取数据，然后对数据进行处理。

* + 1. **外系统开放数据库、本系统调用**

若其他系统没有提供接口，但是本系统需要使用这些系统中生产的数据，我们需要直接在外系统的数据库的基础上开发一些新的接口，通过在前置机上部署新的接口提供服务给本系统使用。根据数据量级别以及数据更新频率，接口采用HTTP方式来实现。

* + 1. **本系统提供接口、本系统调用**

此方式主要为内部接口。

# 运行设计

* 运行模块的组合

客户端程序在有输入时启动接收数据模块，通过各模块之间的调用，读入并对输入进行格式化。在接收数据模块的得到充分数据时，将调用网络传输模块，将数据通过网络送到服务器，并等待接收服务器返回的信息。接收到返回信息后随即调用数据输出模块，对信息进行处理，产生相应的输出。

服务器端程序的网络数据接口模块必须始终处于活动状态。接收到数据后，调用数据处理、查询模块对数据库进行访问，完成后调用网络发送模块，将信息返回客户端。

* 运行控制

首先为用户分配账户和密码，用户在经过认证后登录系统，在分配账户是进行了权限和角色的分配。

* 运行时间

在软件需求分析中，对运行时间的要求为必须对做出的操作有较快的反应。网络硬件对运行时间的影响最大。

服务器的性能将对影响对数据库的访问时间，即客户机操作的等待时间，所以必须使用高性能的服务器。

# 

# 维护设计

## 出错处理设计

* 出错信息提示

程序在运行时主要会出现两种错误：由于输入信息，或无法满足要求时产生的错误，称为软错误。由于其他问题，如网络传输超时等，产生的问题，称为硬错误。

对于软错误，须在操作成功判断及输入数据验证模块由数据进行数据分析，判断错误类型，再生成相应的错误提示语句，送到输出模块中。

对于硬错误，可以在出错的相应模块中输出简单的出错语句，并将程序重置。返回输入阶段。

* 出错处理对策

所有服务器都必须安装不间断电源，以防止停电或电压不稳定造成数据丢失的损失。若突然断电，对客户机将不会有太大的影响；对于服务器，可采用日志文件，对事务进行回滚处理，对数据进行恢复。

在网络传输方面，可考虑建立一条成本较低的备份网络，以保证当主网络断路时数据的通信。

在硬件方面要选择教可靠、稳定的服务器，保证系统运行时的可靠性。

## 安全保密设计

1. 在用户登录系统时进行身份验证，只允许合法用户进入系统。
2. 增加弱口令判断机制，用户输入简单的用户名密码，不能设置。

3、在数据接口方面，通过token进行验证，保证数据传输过程可控。

4、对于存储到数据库中的一些关键敏感信息(如密码等信息)，按要求进行加密。

5、加密存储的各项运行参数和业务参数作为商业机密数据，在未经授权时任何人员都应无法查看和存取。

6、对于加密存储到数据库的关键敏感信息，在客户端和服务器之间传递时应进行加密传输。

7、网络层面的安全，通过省电子政务云平台提供安全防护策略，保障网络应用的安全可靠。同时，做好软件应用层面的经常性检查，及时打好系统补丁和Bug修复。

## 维护设计

为了系统维护的方便，在程序内部设计了日志监控模块，监测用户登录日志、操作日志以及报错日志。

为了用户维护方便，设计了系统管理功能，用户可以调整组织机构、人员、权限。

针对服务器上的数据库数据进行维护，可使用SQL Server的数据库维护功能机制，并配备专门的维护人员定期检查并维护。