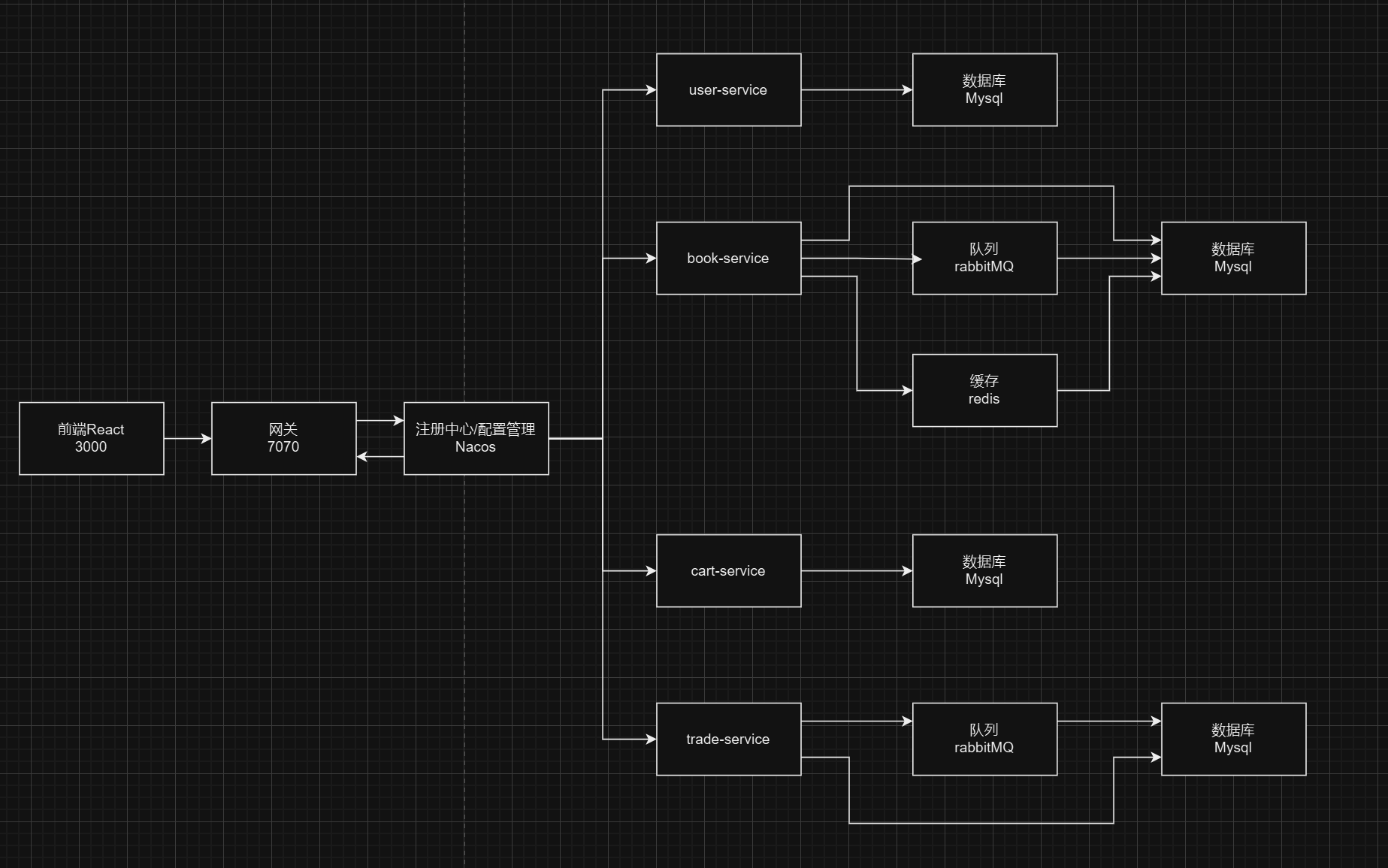
**应用文档**

1. **项目概述：**

* 项目名称：图书管理系统
* 概述：这是一个基于微服务架构的图书管理系统，允许管理员管理书籍、用户、借阅信息。允许用户浏览可借阅的书籍、借用书籍。系统实现了图书信息的增删改查，用户管理，预定模块，交易模块等核心功能。

1. **系统架构：**

* **架构图**



* **技术栈：**

1. **前端：React**
2. **后端：Spring Boot,Spring Cloud,Spring Redis 等**
3. **数据库: Mysql**
4. **缓存：Redis**
5. **消息队列：RabbitMQ**
6. **认证：headers中Authorization认证**
7. **服务注册与发现：Nacos**
8. **API网关：Spring Cloud Gateway**
9. **负载均衡：Spring Cloud Load Balancer**
10. **配置管理：Nacos**
11. **容器化：Docker**
12. **微服务说明：**

* 微服务列表：<services.pdf>

·每个微服务的详细描述：

1. User-service：

·功能描述：负责用户登录，注册，注销，管理员登录。登陆时生成基于JWT加密的token并返回前端。生成验证码。

·API端口： <Api-user.pdf>

·数据库结构： <Domain-user.pdf>

·依赖的其他服务：Trade-service

·错误处理：根据JWT校验捕获登录异常返回错误信息（用户不存在，密码错误，用户冻结，验证码错误）以及状态代码500。

1. Book-service：

·功能描述：实现对大量书籍的分页查询，根据书名，作者，题材进行模糊查询。管理员可以添加修改删除书籍以及实现还书功能。以及在用户成功借书时删除库存。系统应对高并发环境设置了redis缓存数据库分页查询数据，乐观锁防止超扣库存，rabbitMQ队列处理数据库信息。

·API端口：<Api-book.pdf>

·数据库结构：<Domain-book.pdf>

·依赖的其他服务：无

·错误处理：返回错误代码500

1. Cart-service：

·功能描述：添加了一个用户预定库，类似购物车的设计，可以修改要借阅书本数量，日期。

·API端口：<Api-cart.pdf>

·数据库结构：<Domain-cart.pdf>

·依赖的其他服务：User-service,Book-service

·错误处理：返回错误代码500

1. Trade-service：

·功能描述：负责预定书的下单，保存借阅日期，归还日期，逾期费用等关键信息，提示用户逾期金额。在逾期的书超过一定数量时将提醒用户先还书，数量特别多时冻结用户账号。为管理者提供还书功能。数据库分为两个表，一个记录订单，一个记录订单细节，逻辑外键为订单id。

·API端口：<Api-trade.pdf>

·数据库结构：<Domain-order.pdf>，<Domain-orderdetail.pdf>

·依赖的其他服务：Cart-service，Book-service，User-service

·错误处理：返回信息（库存不足，用户异常状态，无法重复下单）返回错误代码500

1. Gateway-service：

·功能描述：负责微服务的网关。开放7070端口，实现路由。内部有全局过滤器实现用户登录的判断并保存用户信息，在配置文件中处理CORS跨域和Nacos实时配置修改

·错误处理：返回错误信息（未授权，token无效，未登录），返回错误代码304，401。

1. API，Common模块

·功能描述：Api基于openfeign负责微服务之间的调用，添加了一个过滤器使得在微服务调用间保存用户信息。Common模块包含众多工具类，异常，配置类和过滤器。实现了用户密码AES加密，redis序列化，JWT加密配置，保存用户信息到当前线程，定义了泛型类用于传递信息给前端，包含其他微服务通用的依赖等。

1. **配置说明**

·CORS配置

|  |
| --- |
| spring:  cloud:  gateway:  routes:  - id: book\_to\_login  uri: lb://book-service  predicates:  - Path=/book/\*\*  - id: user\_service\_route  uri: lb://user-service  predicates:  - Path=/user/\*\*  - Path=/captcha/\*\*  - id: cart\_service\_route  uri: lb://cart-service  predicates:  - Path=/cart/\*\*  - id: trade\_service\_route  uri: lb://trade-service  predicates:  - Path=/trade/\*\*  globalcors:  cors-configurations:  '[/\*\*]':  allowedOrigins:  - "http://localhost:3000"  - "https://flip-wedding-offices-gardening.trycloudflare.com"  allowedMethods:  - GET  - POST  - PUT  - DELETE  - OPTIONS  allowedHeaders:  - Authorization  - Content-Type  allowCredentials: true |

·共享配置

|  |
| --- |
| spring:  datasource:  url: jdbc:mysql://192.168.92.211:3306/yune?useUnicode=true&characterEncoding=UTF-8&autoReconnect=true&serverTimezone=Asia/Shanghai  driver-class-name: com.mysql.cj.jdbc.Driver  username: root  password: 123  redis:  host: 192.168.92.211  database: 0  port: 6379  mybatis-plus:  configuration:  default-enum-type-handler: com.baomidou.mybatisplus.core.handlers.MybatisEnumTypeHandler  global-config:  db-config:  update-strategy: not\_null  id-type: auto  hm:  jwt:  location: classpath:keystore.jks  alias: yune  password: 123456  tokenTTL: 30m |