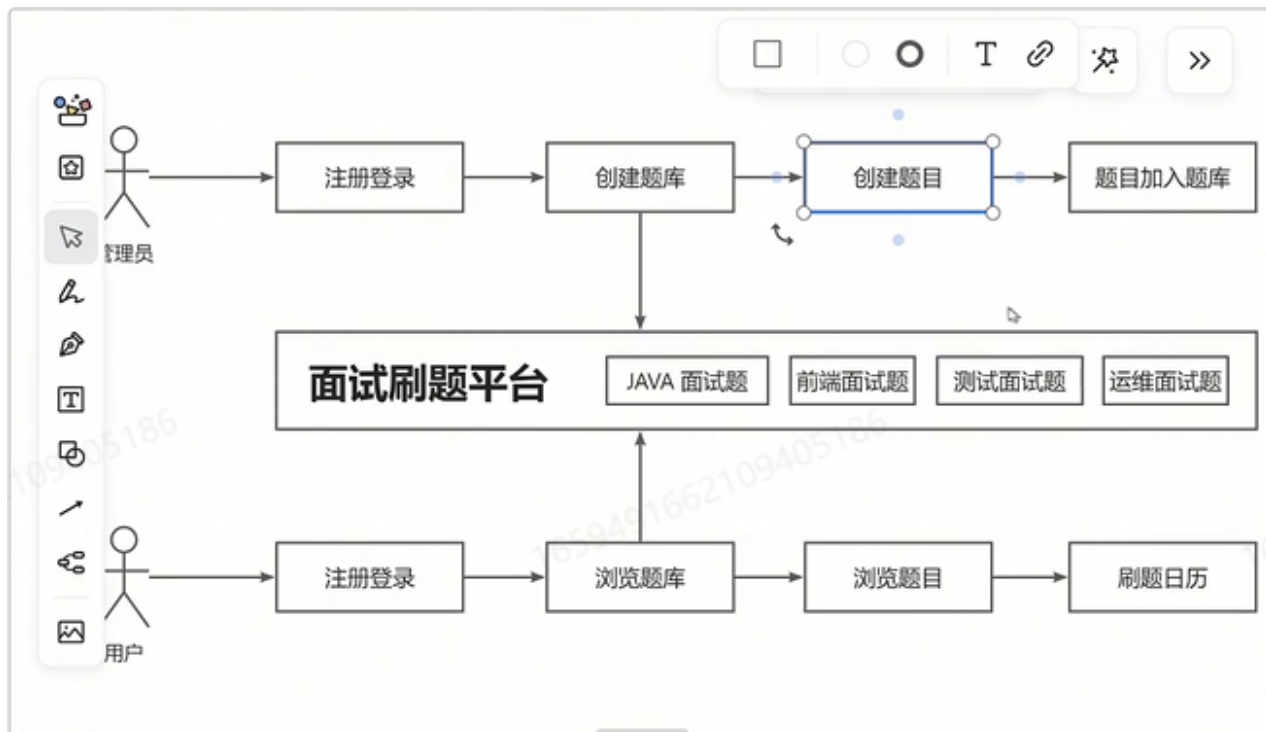


1. 开发前，要想好核心业务流程，画个流程图即最低做完多少功能这个项目才能上线



2. sa-token可以解决同端登录冲突检测

3. draw.io或者语雀画图

4. 核心业务流程分析结束后，要进行具体功能分析与梳理，梳理基础功能。

5. 高级功能

- 题目批量管理
 - 【管理员】批量向题库添加题目
 - 【管理员】批量从题库移除题目
 - 【管理员】批量删除题目
- 分词题目搜索
- 用户刷题记录日历图
- 自动缓存热门题目
- 网站流量控制和熔断
- 动态 IP 黑白名单过滤
- 同端登录冲突检测
- 分级题目反爬虫策略

6. 要有需求优先级的排期。

7. 技术选型

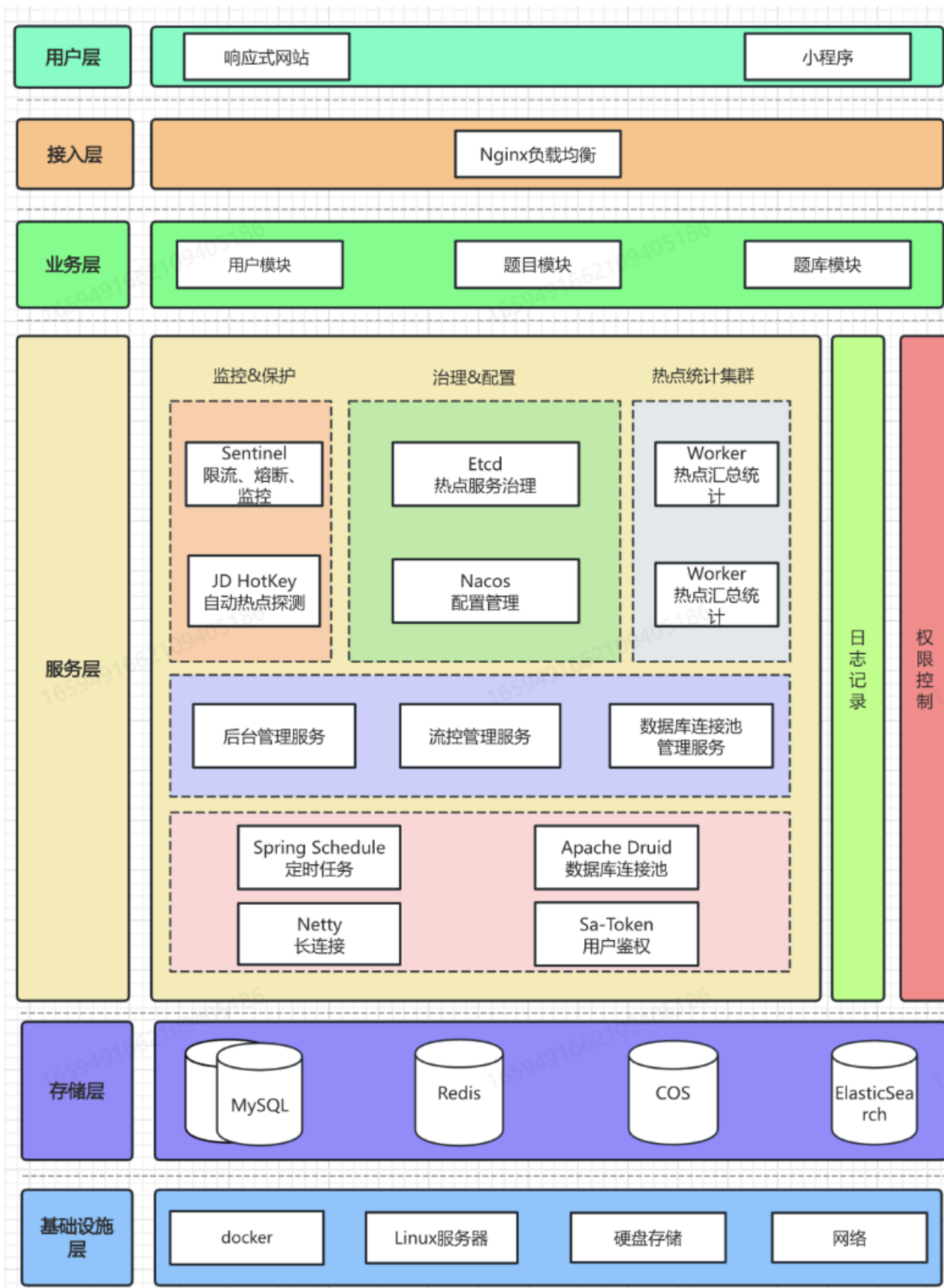
前端

- React 18 框架
- ★ Next.js 服务端渲染
- ★ Redux 状态管理
- Ant Design 组件库
- 富文本编辑器组件
- ★ 前端工程化: ESLint + Prettier + TypeScript
- ★ OpenAPI 前端代码生成

后端

- Java Spring Boot 框架 + Maven 多模块构建
- MySQL 数据库 + MyBatis-Plus 框架 + MyBatis X
- Redis 分布式缓存 + Caffeine 本地缓存
- Redisson 分布式锁 + BitMap + BloomFilter
- ★ Elasticsearch 搜索引擎
- ★ Druid 数据库连接池 + 并发编程
- ★ Sa-Token 权限控制
- ★ HotKey 热点探测
- ★ Sentinel 流量控制
- ★ Nacos 配置中心
- ★ 多角度项目优化: 性能、安全性、可用性

8. 架构设计



1. 用户层：最直接让用户去使用的层。
2. 接入层：负责接收请求，把所有用户发来的请求做一个处理，比如将某个用户发来的请求转发到某个服务器上，比如对请求做一个过滤，比如加一个web防火墙，比如加一个限流。
3. 业务层：划分为多少个模块。
4. 服务层：支撑业务的模块、服务。相比业务层，更注重**复用**。