软件设计规格说明书（SDD）

依据 IEEE STD 1016-2009

项目名称：在线考试系统（OES）

修订日期：2025年3月5日

编写团队：苟瑞祥（项目经理）、李轩珂（系统分析师）、侯一玮（后端）、胡禹晨（全栈）、陈致江（测试）

目录

[1. 引言 3](#_Toc194485582)

[1.1 目的 3](#_Toc194485583)

[1.2 范围 3](#_Toc194485584)

[1.3 术语&定义 3](#_Toc194485585)

[2. 初步设计 3](#_Toc194485586)

[2.1 系统架构 3](#_Toc194485587)

[2.1.1 逻辑架构（组件图） 3](#_Toc194485588)

[2.1.2 物理架构 4](#_Toc194485589)

[2.2 数据架构 4](#_Toc194485590)

[2.2.1 数据流图（DFD） 4](#_Toc194485591)

[2.2.2 数据字典（部分） 5](#_Toc194485592)

[3. 详细设计 5](#_Toc194485593)

[3.1 组件设计 5](#_Toc194485594)

[3.1.1 组卷引擎 5](#_Toc194485595)

[3.1.2 自动阅卷模块 5](#_Toc194485596)

[3.2 接口设计 6](#_Toc194485597)

[3.2.1 REST API 规范 6](#_Toc194485598)

[3.2.2 WebSocket 协议 6](#_Toc194485599)

[3.3 数据设计 6](#_Toc194485600)

[3.3.1 数据库表结构 6](#_Toc194485601)

[3.3.2 缓存策略 7](#_Toc194485602)

[4. 安全设计 7](#_Toc194485603)

[4.1 认证与授权 7](#_Toc194485604)

[4.2 安全防护 7](#_Toc194485605)

[5. 部署设计 7](#_Toc194485606)

[5.1 容器化配置 7](#_Toc194485607)

[5.2 高可用方案 8](#_Toc194485608)

[6. 设计验证 8](#_Toc194485609)

[6.1 单元测试策略 8](#_Toc194485610)

[6.2 性能测试用例 8](#_Toc194485611)

[附录 8](#_Toc194485612)

# 1. 引言

## 1.1 目的

本文档依据 IEEE 1016 标准定义在线考试系统的软件设计描述，将需求转化为可执行的组件、接口与数据结构，指导开发团队完成系统实现。

## 1.2 范围

覆盖系统架构设计、数据模型、接口规范、安全机制及部署策略，满足以下需求：

30人并发考试（响应时间 ≤1.5秒）

99.9%可用性（MySQL主从架构）

Excel试题批量导入（效率提升60%）

## 1.3 术语&定义

OES 基于B/S架构的在线考试系统，支持自动组卷与阅卷

JWT JSON Web Token，用于无状态跨域认证

Shiro Apache安全框架，实现RBAC权限控制

# 2. 初步设计

## 2.1 系统架构

## 2.1.1 逻辑架构（组件图）

[前端] Vue.js SPA

│

▼

[REST API] SpringBoot 微服务

├── 认证服务（Shiro + JWT）

├── 组卷引擎（遗传算法）

├── 阅卷模块（正则匹配）

└── 监控服务（WebSocket）

│

▼

[数据层] MySQL 8.0 + Redis 7.0

## 2.1.2 物理架构

部署拓扑：

前端：Nginx静态资源服务器（CDN加速）

后端：阿里云ECS集群（Docker容器化部署）

数据库：MySQL主从同步 + Redis哨兵模式

## 2.2 数据架构

## 2.2.1 数据流图（DFD）

考生 → 提交答案 → 阅卷服务 → 成绩库

▲

│

组卷服务 → 试题池 → Redis缓存

## 2.2.2 数据字典（部分）

|  |  |  |
| --- | --- | --- |
| 数据项 | 类型 | 描述 |
| Exam.paper\_id | UUID | 试卷唯一标识符 |
| User.role | Enum | 用户角色（Admin/Teacher/Student） |

# 3. 详细设计

## 3.1 组件设计

## 3.1.1 组卷引擎

输入：知识点标签、难度系数、题型分布

算法：

java

public List<Question> generatePaper(Map<String, Integer> constraints) {

// 1. 从Redis缓存加载候选试题

// 2. 遗传算法优化权重分配（适应度函数：匹配度评分）

// 3. 返回最优试题序列

}

输出：试卷ID及试题列表（JSON格式）

## 3.1.2 自动阅卷模块

客观题评分逻辑：

sql

SELECT CASE WHEN user\_answer = standard\_answer THEN 1 ELSE 0 END

FROM question\_bank

WHERE question\_id = ?

## 3.2 接口设计

## 3.2.1 REST API 规范

|  |  |  |  |
| --- | --- | --- | --- |
| 接口 | 方法 | 请求体 | 响应码 |
| /api/exam/start | POST | { "userId": 123, "paperId": "xyz" } | 201 (Created) |
| /api/exam/submit | POST | { "answers": {"Q1": "A", ...} } | 200 (Success) |

## 3.2.2 WebSocket 协议

端点：/ws/monitor/{examId}

消息格式：

json

{

"eventType": "WINDOW\_BLUR",

"timestamp": "2025-05-01T14:30:00Z",

"userId": 123

}

## 3.3 数据设计

## 3.3.1 数据库表结构

表：exam\_record

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 类型 | 约束 | 描述 |
| record\_id | BIGINT | PRIMARY KEY, AUTO\_INCREMENT | 考试记录ID |
| user\_id | INT | FOREIGN KEY | 关联用户表 |
| start\_time | DATETIME | NOT NULL | 考试开始时间 |

## 3.3.2 缓存策略

Redis键设计：

会话令牌：session:{jwt} → {userId, expireTime}（TTL 1800秒）

试题池：questions:{examId} → List<Question>

# 4. 安全设计

## 4.1 认证与授权

认证流程：

用户提交用户名/密码

BCrypt哈希验证

签发JWT（包含角色声明）

权限模型：

role:Teacher → permission:paper\_manage

role:Student → permission:exam\_take

## 4.2 安全防护

输入验证：

XSS过滤：前端使用DOMPurify.sanitize()

SQL注入：MyBatis预编译语句

# 5. 部署设计

## 5.1 容器化配置

dockerfile

# SpringBoot服务Dockerfile

FROM openjdk:17

COPY target/oes-backend.jar /app.jar

ENTRYPOINT ["java","-jar","/app.jar"]

## 5.2 高可用方案

MySQL：

主库：读写操作

从库：实时同步 + 故障自动切换（Keepalived）

微服务：

Nginx负载均衡（轮询策略）

服务健康检查（Spring Boot Actuator）

# 6. 设计验证

## 6.1 单元测试策略

覆盖率要求：

业务逻辑层 ≥90%（JUnit 5 + Mockito）

DAO层：集成测试（Testcontainers）

## 6.2 性能测试用例

|  |  |  |
| --- | --- | --- |
| 测试场景 | 指标 | 预期结果 |
| 30并发提交试卷 | 平均响应时间 ≤1.5秒 | 服务器CPU ≤70% |
| 1000试题导入 | 执行时间 ≤30秒 | 内存峰值 ≤2GB |

## 附录

A. 参考文档

IEEE STD 1016-2009 官方标准

《OES需求规格说明书》v2.1