<软件工程大作业>

软件需求规约

版本 <1.0>

修订历史记录

|  |  |  |  |
| --- | --- | --- | --- |
| **日期** | **版本** | **说明** | **作者** |
| <07日/06月/2025年> | <1.0> | <初始需求规约与需求> | <刘黎> |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

目录

1. 简介 4

1.1 目的 4

1.2 定义、首字母缩写词和缩略语 4

1.3 参考资料 4

2. 整体说明 4

3. 具体需求 4

3.1 功能 5

3.1.1 <Use case 图> 5

3.1.2 <Use case1 规约> 5

3.1.3 <Use case2 规约> 5

3.2 易用性 5

3.2.1 <易用性需求一> 5

3.3 可靠性 5

3.3.1 <可靠性需求一> 6

3.4 性能 6

3.4.1 <性能需求一> 6

3.5 可支持性 6

3.5.1 <可支持性需求一> 6

3.6 设计约束 6

3.6.1 <设计约束一> 6

3.7 联机用户文档和帮助系统需求 7

3.8 接口 7

3.8.1 用户界面 7

3.8.2 硬件接口 7

3.8.3 软件接口 7

3.8.4 通信接口 7

3.9 适用的标准 7

软件需求规约 (简化版)

# 简介

## 目的

本软件需求规约（SRS）文档的目的是详细描述为学生和教师提供的在线考试平台的功能性需求和非功能性需求。该平台旨在为学生提供一个便捷的在线考试和成绩查询系统，同时为教师提供一个有效的期末成绩管理工具。本系统还包括教务管理人员的附加功能

该文档将

* **功能性需求**：描述平台应具备的核心
* **非功能性需求**：如性能要求
* **设计约束**：描述
* **其他相**：包括与系统连接

本需求文档将为开发团队提供系统的全面需求说明，并为后续的设计与实现阶段提供清晰的指导。

## 定义、首字母缩写词和缩略语

为了便于理解本软件需求规约（SRS）中的相关内容，以下列出了项目中使用的主要术语及缩写词：

#### 术语

* **在线考试平台**：一个基于Web的应用程序，允许学生在线参加考试，教师管理考试和成绩，教务人员管理教师和学生信息。
* **学生端**：学生使用的系统界面，提供考试、练习、成绩查看等功能。
* **教师端**：教师使用的系统界面，提供考试管理、题库管理、成绩查询等功能。
* **教务端**：教务人员使用的系统界面，提供教师和学生信息管理等功能。
* **试卷**：由教师通过系统创建并分配给学生的考试内容，包含一组试题。
* **题库**：包含各种题目的数据库，教师可以从中选择问题生成试卷。
* **成绩查询**：学生或教师查看考试成绩的功能模块。
* **留言板**：学生用来进行互动和反馈的讨论板块，支持学生间的交流。
* **分段查询**：教师用来查看学生在某一科目或阶段的成绩分布情况。

#### 首字母缩写词和缩略语

* **SRS**：Software Requirements Specification，软件需求规约。
* **UI**：User Interface，用户界面。
* **API**：Application Programming Interface，应用程序接口。
* **DBMS**：Database Management System，数据库管理系统。
* **HTTP**：HyperText Transfer Protocol，超文本传输协议。
* **SQL**：Structured Query Language，结构化查询语言。
* **CRUD**：Create, Read, Update, Delete，增删改查四个基本操作。
* **MTTR**：Mean Time To Repair，平均修复时间。
* **MTBF**：Mean Time Between Failures，平均故障间隔时间。
* **SMTP**：Simple Mail Transfer Protocol，简单邮件传输协议。

## 参考资料

[1] 需求概要设计文档格式标准[S].GB856D-1988.

[2]Robert C. Martin.敏捷软件开发：原则、模式与实践[M].北京：清华大学出版社，2003.09

[3]窦万峰.软件工程方法与实践[M].北京：机械工业出版社，2016.10

[4]窦万峰.软件工程实验教程[M].北京：机械工业出版社，2016.11

# 整体说明

* 1. **产品总体效果**

该在线考试平台是一个独立运行的系统，专为高校设计，旨在简化考试管理流程，提高考试的效率和公正性。平台提供多角色支持，分别服务于学生、教师和教务人员。学生通过该平台参与在线考试、查看成绩并进行练习；教师负责创建考试、管理题库并批改试卷；教务人员则负责整体管理、权限控制和成绩统计等工作。

该平台自成一体，并没有与其他系统高度集成，主要依靠Web应用架构运行。平台的核心功能包括考试管理、成绩管理、题库管理和学生信息管理。所有功能都围绕着考试周期进行设计，从考试的创建、组织、批改到成绩分析，提供一体化的解决方案。用户可以通过浏览器访问平台，且平台支持主流的Web浏览器，如Chrome和Firefox。

未来，平台可能会与学校现有的学生信息系统或教务管理系统进行集成，以实现数据共享与同步。对于较大的学校或教育机构，平台还可以通过API接口与外部系统进行交互。通过这种方式，系统能够实现更高效的数据流通与管理，从而提高学校的运营效率。此外，平台的模块化设计使其在未来可以扩展更多的功能，如在线学习管理、远程教学支持等。

平台的设计目标是提供一个直观、易用且功能全面的考试管理工具，特别是在提高教师和教务人员的工作效率方面，能有效减少人工操作和繁琐的管理过程。通过实时成绩反馈和自动阅卷功能，平台还能够提升学生的学习体验和参与感，帮助学生及时发现问题并进行针对性的复习。

* 1. **产品功能**

**客户支持系统**

|  |  |
| --- | --- |
| **客户利益** | **支持特性** |
| 新的支持人员能够很快地步入正轨。 | 知识库可协助支持人员迅速地找到已知的解决方法和变通方法。 |
| 因为考虑周全而提高了客户满意度。 | 在整个解决过程中可将问题一一列出，并进行分类和跟踪。一出现老化问题就自动发出通知。 |
| 管理人员能够发现存在问题的领域并估计人员的工作量。 | 趋势及分布报告可从较高的角度来审查问题的状态。 |
| 分散的支持团队能够协同解决问题。 | 复制服务器使当前的数据库信息可以在整个企业的范围内共享 |
| 客户能够自行解决一些问题，从而降低了支持成本并缩短了答复时间。 | 可以通过 Internet 来访问知识库。包括超文本搜索功能和图形查询引擎 |

* 1. **用户特征**

该平台的主要用户群体包括以下三类：

* **学生**：有一定的计算机基础，能够通过系统完成考试、复习和成绩查询等任务。
* **教师**：具有较强的教学能力，熟悉试卷和题库的管理流程。
* **教务人员**：负责管理教师、学生的基础信息和考试安排，通常具备一定的行政管理经验。
  1. **约束**

本系统的开发需要遵守以下几个约束条件：

* **技术约束**：系统需使用Web技术（如HTML、CSS、JavaScript、数据库管理系统等）进行开发，确保跨平台兼容性。
* **时间约束**：项目需在特定时间内完成并投入使用，开发周期较短。
* **预算约束**：项目开发需在既定预算范围内完成。
  1. **假设与依赖关系**

在本项目的前景文档中，我们假设一些关键因素将影响平台的开发和运行。这些假设构成了平台设计与实现的基础，而任何这些假设的变化都可能导致前景文档的相关内容发生变化。

首先，我们假设平台将运行在学校提供的标准硬件上，并且会使用特定的操作系统（如Windows或Linux）。如果学校未能提供所需的操作系统或硬件配置，可能需要对平台的部署要求进行修改，或者在设计中引入额外的兼容性支持。例如，如果平台无法在现有的操作系统环境中运行，则可能需要对软件进行重新开发或调整其系统架构，以支持新的操作系统或硬件配置。

其次，我们假设平台将与学校现有的学生信息系统（SIS）和教务管理系统（CMS）进行一定程度的集成，以便实现数据同步和共享。这一集成假设基于平台能够访问和交互现有的学校数据库系统。如果学校的系统不兼容或者没有提供API接口支持，那么这一假设将需要更改，可能导致平台需要开发新的集成功能，或改用其他方案来处理数据管理和同步问题。

另外，我们假设所有用户（学生、教师和教务人员）将通过学校的统一身份认证系统登录平台。如果学校未能提供统一的身份认证服务，平台的用户身份管理和安全功能将需要重新设计，可能涉及到更多的权限控制和独立的用户管理系统。

最后，我们假设平台的使用不会受到极端网络环境或不稳定网络连接的影响。然而，如果学校网络环境不稳定或频繁出现网络故障，平台的设计需要增强对断线恢复、离线功能支持以及低带宽环境下的优化。这可能涉及修改平台的架构，加入更多的容错机制和数据缓存功能，以保障用户在不稳定网络条件下的正常使用体验。

* 1. **需求子集**

**2.6.1 学生端需求子集**

1. 学生登录/注册：学生可以通过账户和密码登录系统，访问自己的考试信息。
2. 参加考试：学生可以查看可参加的考试并进入考试界面。
3. 练习与复习：学生可以查看历史考试试题，并进行复习和练习。
4. 成绩查询：学生可以查看自己的考试成绩和成绩分析。
5. 留言板功能：学生可以在留言板上与同学进行互动和讨论。

**2.6.2 教师端需求子集**

1. 试卷管理：教师可以创建、编辑和发布试卷。
2. 题库管理：教师可以管理题库，添加新题目或删除旧题目。
3. 成绩管理：教师可以查看学生的成绩，并进行成绩分析与分段查询。
4. 学生管理：教师可以查看、编辑学生信息，包括添加、删除学生。

**2.6.3 教务端需求子集**

1. 教师管理：教务人员可以管理所有教师的基本信息，包括添加、删除教师等操作。
2. 学生管理：教务人员可以管理所有学生的基本信息，包括添加、删除学生等操作。
3. 考试安排管理：教务人员可以查看和管理教师设置的所有考试安排。
4. 权限管理：教务人员负责权限设置，确保不同用户（学生、教师、教务人员）的权限分配正确。

**2.6.4 系统安全和性能需求子集**

1. 用户认证：确保学生、教师和教务人员的账户安全，支持密码加密存储。
2. 数据安全：确保用户的个人信息和成绩数据的安全性，防止数据泄露。
3. 性能优化：确保系统能够支持大量用户同时访问而不影响性能，保证较低的响应时间。
4. 高可用性：确保系统的稳定性，尤其在高负载情况下能够继续正常运行。

**2.6.5 系统接口需求子集**

1. 用户界面：为用户提供简洁、直观的操作界面，确保良好的用户体验。
2. 数据库接口：系统需要与数据库管理系统进行交互，存储用户信息、考试数据等。
3. 外部接口：如果有任何第三方系统（如邮件服务或成绩导出服务），这些接口将被详细描述。

# 具体需求

## 功能

**客户支持系统**

|  |  |
| --- | --- |
| **客户利益** | **支持特性** |
| 新的支持人员能够很快地步入正轨。 | 知识库可协助支持人员迅速地找到已知的解决方法和变通方法。 |
| 因为考虑周全而提高了客户满意度。 | 在整个解决过程中可将问题一一列出，并进行分类和跟踪。一出现老化问题就自动发出通知。 |
| 管理人员能够发现存在问题的领域并估计人员的工作量。 | 趋势及分布报告可从较高的角度来审查问题的状态。 |
| 分散的支持团队能够协同解决问题。 | 复制服务器使当前的数据库信息可以在整个企业的范围内共享 |
| 客户能够自行解决一些问题，从而降低了支持成本并缩短了答复时间。 | 可以通过 Internet 来访问知识库。包括超文本搜索功能和图形查询引擎 |

### <Use case 图>

[Use case 图，并对每个actor或usecase有简要说明。如果系统比较大，则可按用户或子系统进行组织]

### <学生登陆规约>

### <Use case2 规约>

## 易用性

### <简洁直观的用户界面>

**需求说明**：

1. **界面设计简洁**：
   * 系统应采用简洁、直观的界面设计，避免过多的复杂元素，以确保学生、教师和教务人员能够迅速理解并有效操作系统。所有主要功能按钮和导航菜单应突出显示，确保用户能快速找到所需操作。
2. **一致性设计**：
   * 系统的各个页面应遵循一致的设计规范，保证用户在切换不同功能模块时能够轻松适应。页面布局、色彩、按钮样式等应统一，并遵循公认的界面设计标准，如IBM的CUA标准和Microsoft的GUI标准。
3. **支持常见的操作习惯**：
   * 界面应支持学生、教师和教务人员的常用操作习惯，例如：
     + 登录界面应设置明确的字段提示，避免用户输入错误。
     + 功能按钮应以清晰的标签（如“开始考试”、“查看成绩”、“编辑学生信息”等）命名，且应符合用户的自然操作习惯。
4. **减少学习时间**：
   * 对于普通用户（如学生），系统的学习时间应尽可能短。学生应能够在首次使用平台时，通过简洁的指导或帮助提示快速了解如何进行常见操作（如登录、查看试卷、查看成绩等）。
   * 对于高级用户（如教师和教务人员），系统应提供适当的帮助和培训文档，使其能在最短时间内掌握复杂的操作功能（如创建考试、管理题库等）。
5. **响应式设计**：
   * 系统应具备响应式设计，确保在不同屏幕尺寸下（如PC、平板、手机等）都能提供一致的用户体验。用户应在各种设备上能够顺畅操作，而不需要进行额外的调整。
6. **反馈机制**：
   * 系统应提供明确的操作反馈，确保用户在每个操作步骤中都能看到实时的反馈信息。例如，用户在提交表单时，系统应提供成功提示或错误警告，帮助用户了解操作结果。
7. **易访问性要求**：
   * 系统应支持高对比度和大字体模式，以适应视力障碍用户。
   * 提供屏幕阅读器支持，帮助视力受限的用户顺利使用平台。

## 可靠性

### <高可用性和故障恢复>

 **系统可用性**：

* 系统应具备高可用性，全年可用时间应达到**99.9%以上，意味着系统每年最多允许8.77小时**的停机时间。
* 在计划维护期间，系统应提前通知用户，且维护时间不应超过**2小时**，并且尽量安排在非高峰时段。

 **故障恢复能力**：

* 系统应具备自动恢复机制，在发生硬件故障或软件异常时，能够在不影响整体功能的情况下，迅速恢复正常服务。
* 在遇到故障时，系统应具备降级模式，确保核心功能（如学生登录、考试访问、成绩查询等）能继续运行，即使某些次要功能暂时无法使用。

 **平均故障间隔时间 (MTBF)**：

* 系统的**平均故障间隔时间（MTBF）应至少为500小时**，即系统在每500小时的运行中，出现一次故障的概率应低于1次。
* 这一指标应适用于系统的核心功能模块，如用户认证、考试管理和成绩查询等。

 **平均修复时间 (MTTR)**：

* 在发生故障时，系统应能够在**2小时**内修复问题并恢复服务。对于关键故障（如数据库崩溃、核心服务不可用等），系统的修复时间应小于**1小时**。
* 该修复时间不应超过**4小时**，并且在此期间，用户应收到定期更新通知，确保透明度。

 **错误率**：

* 系统代码的错误率（bugs/KLOC）应不超过**0.5**，即每千行代码中的错误数不应超过0.5个。这一指标有助于确保代码的高质量。
* 系统每个功能点的错误数（bugs/function-point）应控制在**0.05**以内。

 **缺陷分类和严重性定义**：

* 系统的错误或缺陷应根据影响程度进行分类，定义如下：
  + **小错误**：不影响系统功能或仅影响部分用户，修复时间不超过**24小时**。
  + **大错误**：影响部分功能或大量用户，但不影响系统的核心服务，修复时间应在**48小时**内。
  + **严重错误**：导致系统无法启动、系统崩溃或数据丢失，必须在**24小时**内修复，修复过程中系统应启用临时替代方案或降级模式，确保关键功能可用。

 **数据完整性**：

* 系统在处理用户数据（如成绩记录、考试数据等）时，应保证数据的完整性和一致性。在发生系统故障时，应确保不会丢失数据，且修复后能恢复到故障前的状态。

## 性能

### <系统响应时间和吞吐量>

 **响应时间要求**：

* **平均响应时间**：在正常负载下，系统的平均响应时间应不超过**3秒**，即用户在提交请求（如登录、查询成绩、进入考试等）后，系统应在3秒内完成响应并提供结果。
* **最长响应时间**：在高峰负载情况下，系统的最长响应时间应不超过**5秒**，以确保用户能够在合理的时间内完成关键操作，如参加考试、查看成绩等。

 **吞吐量要求**：

* 系统应能够支持每秒**至少1000次请求**的处理能力，确保在高并发情况下仍能保持较高的服务质量。
* 在使用负载均衡和集群技术的情况下，系统应能够根据需求自动扩展处理能力，支持峰值流量并保持稳定性。

 **系统容量要求**：

* **用户容量**：系统应能够支持**至少10000名并发用户**同时在线，进行登录、考试参与、成绩查询等操作，而不影响系统的性能。
* **事务容量**：系统应能够处理每秒**至少2000个事务**（如提交考试答案、成绩查询、题目生成等），并在高负载情况下保持稳定。
* **数据存储容量**：系统应能够容纳**至少1TB的数据**，包括学生信息、考试记录、成绩数据等，并保持良好的访问性能。

 **降级模式要求**：

* 在系统出现部分功能故障或超负荷时，系统应能自动启用降级模式，确保核心功能（如学生登录、考试访问、成绩查询）仍然可用。非核心功能（如系统设置、部分报告生成等）可暂时禁用。
* 降级模式应透明地通知用户，明确告知哪些功能不可用，并提供合理的操作替代或等待时间。

 **资源利用情况**：

* **内存使用**：系统应优化内存管理，确保在高并发情况下，系统的内存使用不超过可用内存的80%，避免因内存不足导致的性能下降或崩溃。
* **磁盘使用**：数据库和文件存储系统应定期清理无用数据，确保磁盘空间的有效利用，防止存储容量不足对性能的影响。
* **通信资源**：系统应优化数据传输和网络通信，确保每个请求的网络延迟不超过**200毫秒**，保证稳定的网络连接并减少通信带来的性能瓶颈。

## 可支持性

### <编码标准与命名约定>

 **编码标准**：

* 所有系统的源代码必须遵循统一的编码标准，确保代码的可读性和一致性。推荐使用[公司/团队]内部指定的编码风格指南，涵盖代码排版、注释规范、缩进方式、函数/方法的命名等内容。
* 使用现代编程语言的最佳实践，例如：
  + 在代码中使用适当的注释来解释复杂逻辑。
  + 保持函数和类的职责单一，避免过度复杂化。
  + 代码中应避免使用硬编码的值，尽量使用配置文件或环境变量进行管理。

 **命名约定**：

* 系统中的所有变量、函数、类名、文件名等应遵循统一的命名规范。例如：
  + 变量名和函数名应使用**驼峰命名法**（如getUserInfo、totalAmount）。
  + 类名应使用**Pascal命名法**（如UserAccountManager）。
  + 常量应使用**全大写字母**，并使用下划线分隔（如MAX\_USER\_LIMIT）。
* 在命名时，应确保能够清晰描述其用途，避免使用含糊不清或与其他模块功能重叠的名称。

 **类库与框架**：

* 系统应使用公认的、稳定的类库和框架，如：
  + **前端框架**：React、Vue.js 或 Angular 等，确保前端界面的响应性和性能。
  + **后端框架**：Spring Boot、Django 或 Express 等，用于提升后端开发的效率和可维护性。
* 所有第三方库或框架应保持最新版本，及时更新以解决潜在的安全漏洞或性能问题。

 **维护访问权限**：

* 系统应设置严格的访问权限控制，确保只有授权人员能够访问系统的配置文件、日志和关键服务。
* 系统的生产环境与开发环境应严格分开，并且仅授权的开发人员和运维人员可以访问生产环境。
* 系统应记录所有的管理员操作，包括访问日志、配置修改和版本更新等，确保能够追溯和审计操作记录。

 **维护实用程序**：

* 系统应提供易于使用的维护工具，如：
  + 自动化日志分析工具，能够快速定位问题，并生成可操作的故障排查报告。
  + 定期的数据库备份工具，确保数据的安全性，能够在系统发生故障时迅速恢复。
  + 系统健康监控工具，定期检查系统的运行状态、性能和安全性，并及时发出警报。

 **文档要求**：

* 所有开发的功能模块应有详细的文档，包括代码注释、功能说明文档和使用手册。
* 在功能开发过程中，开发人员应提供易于理解的文档，帮助后续的维护人员和开发人员能够迅速理解系统架构、模块功能以及常见问题的解决方案。

## 设计约束

### <技术栈与开发工具>

 **编程语言**：

* 系统的前端应使用**JavaScript**，配合现代前端框架（如**React**或**Vue.js**）来开发用户界面，确保界面的响应性和交互性。
* 系统的后端应使用**Java**（使用**Spring Boot**框架）或**Python**（使用**Django**框架）来开发RESTful API服务。选择这些语言和框架是基于其良好的性能、稳定性和广泛的社区支持。

 **数据库技术**：

* 系统应使用**关系型数据库**（如**PostgreSQL**或**MySQL**）来存储用户数据、考试记录和成绩信息。选择关系型数据库是因为其支持复杂查询和事务管理，并具有高度的数据一致性。
* 在某些高性能需求场景下（如实时数据分析），可考虑引入**NoSQL**数据库（如**MongoDB**）作为辅助存储，用于存储非结构化数据。

 **开发工具与版本控制**：

* 所有开发人员应使用**Git**作为版本控制工具，所有代码和文档应托管在**GitHub**或**GitLab**上，确保代码管理的清晰和高效。
* 开发环境应使用**IntelliJ IDEA**（针对Java开发）或**Visual Studio Code**（针对JavaScript开发）等IDE，以提高开发效率。
* 系统应实现CI/CD（持续集成/持续交付）流程，使用工具如**Jenkins**、**Travis CI**或**GitLab CI**进行自动化构建和测试。

 **架构设计**：

* 系统应采用**微服务架构**，各个服务之间通过RESTful API进行通信，确保系统的高可扩展性和模块化。微服务架构可以将系统分解为多个独立的服务模块，使得不同功能可以独立开发、测试和部署。
* 所有服务应遵循**SOLID原则**（单一职责、开放封闭、里氏替换、接口隔离、依赖倒置），确保代码的可维护性和扩展性。

 **第三方类库与构件**：

* 系统应使用行业标准的类库和框架，如**Spring Security**（用于用户认证和授权）、**Hibernate**（用于数据库交互）、**JUnit**（用于单元测试）、**Swagger**（用于API文档生成）。
* 所有第三方类库和构件应定期更新至稳定版本，确保它们不会成为系统漏洞的来源。

 **安全与合规性**：

* 系统的设计必须符合数据保护和隐私保护的法律法规，如**GDPR**（通用数据保护条例）和中国的**个人信息保护法**。所有涉及敏感数据的操作（如用户信息、成绩等）都应符合相关合规要求。
* 系统应内置权限管理和审计日志功能，确保只有授权用户能够访问敏感数据，并记录所有重要操作。

 **兼容性**：

* 系统应支持所有主流操作系统（Windows、MacOS、Linux）以及主流浏览器（Chrome、Firefox、Safari、Edge），确保用户在不同平台上都能流畅使用系统。
* 移动端版本应适配常见的手机操作系统（iOS、Android），并保证核心功能的可用性和良好的用户体验。

## 联机用户文档和帮助系统需求

本节列出了系统中涉及联机用户文档和帮助系统的需求。这些需求确保用户能够方便地获得操作指导和系统支持，帮助他们高效地使用平台功能。

**3.7.1 联机用户文档需求**

1. **用户手册**：
   * 系统应提供一份详细的用户手册，包含学生、教师和教务人员在使用平台时可能遇到的所有操作指南和功能说明。
   * 用户手册应包括以下内容：
     + 如何注册、登录和退出系统。
     + 如何进行考试、查看成绩以及复习练习。
     + 如何创建和管理试卷、题库，如何查看和管理学生信息。
     + 教务人员如何管理教师和学生账户。
     + 系统的常见问题解答（FAQ）和解决方案。
2. **在线帮助文档**：
   * 系统应内嵌在线帮助文档，提供简洁明了的操作指引。用户可以在系统界面上直接访问这些帮助文档，以便快速找到问题的解决方法。
   * 文档应支持搜索功能，用户可以通过关键字快速查找相关帮助内容。
   * 每个功能模块（如学生考试、教师管理试卷、教务人员管理账户等）都应提供相应的帮助文档，确保用户能够轻松理解和操作。

**3.7.2 帮助系统需求**

1. **帮助按钮与工具提示**：
   * 在系统的关键操作界面，应提供帮助按钮，用户可以点击帮助按钮查看与当前操作相关的帮助内容。
   * 每个输入框、按钮或交互元素应提供相应的工具提示，简要说明该功能的用途和操作方法，以帮助用户快速理解如何使用。
2. **互动帮助系统**：
   * 系统应支持通过实时聊天或人工客服提供互动帮助。用户可以通过在线客服与支持人员沟通，获取实时的技术支持或操作指导。
   * 系统应集成常见问题解答（FAQ）和人工客服服务，确保用户能够迅速得到问题的答案。
3. **常见问题解答（FAQ）**：
   * 系统应提供一个FAQ页面，列出用户在使用系统过程中可能遇到的常见问题和解答。FAQ应涵盖各类用户（学生、教师、教务人员）的常见问题。
   * FAQ页面应支持搜索功能，用户可以通过关键字快速查找相关的常见问题和解答。
4. **帮助系统可访问性要求**：
   * 帮助系统应具备良好的可访问性，确保视力障碍的用户也能够通过屏幕阅读器等辅助设备轻松访问帮助文档和系统功能。
   * 系统应提供高对比度和大字体模式，确保所有用户能够清晰阅读帮助内容。

**3.7.3 帮助内容更新与维护**

1. **定期更新**：
   * 用户文档和帮助系统的内容应定期更新，以反映系统的功能更新和版本变更。任何新增的功能或变更都应在文档和帮助系统中及时更新。
2. **版本控制**：
   * 系统文档应与软件版本进行关联，确保用户能够访问与其使用的系统版本相匹配的文档内容。

## 接口

### 用户界面

 **登录界面**：

* 系统应提供简洁、直观的登录页面，用户可以通过输入用户名和密码登录系统。登录界面应支持错误提示，指导用户重新输入正确的信息。
* 登录页面应提供“忘记密码”和“注册新账户”链接，支持学生恢复账户访问。

 **学生界面**：

* 学生的主页应显示考试信息、成绩查询入口和复习材料，界面应易于导航，突出关键操作（如开始考试、查看成绩、查看试题等）。
* 所有按钮和操作元素应按照常见的UI设计标准进行布局，确保界面响应迅速且易于理解。

 **教师界面**：

* 教师界面应提供创建试卷、管理题库、查看学生成绩和分析结果等功能。
* 界面应支持批量操作，如批量导入题目、批量上传成绩等，减少教师的工作负担。

 **教务人员界面**：

* 教务人员界面应具备管理教师和学生账户、审核考试安排和处理系统设置的功能。
* 教务人员界面应支持查询和导出考试数据和成绩报告，且界面功能应经过简化以便于高效操作。

 **辅助功能**：

* 用户界面应提供高对比度模式和大字体模式，以适应视力受限的用户。
* 支持屏幕阅读器和语音指令操作，确保系统符合易用性和无障碍标准。

 **平台兼容性**：

* 用户界面应支持主流操作系统（如Windows、MacOS、Linux）和主流浏览器（如Chrome、Firefox、Safari、Edge）以及移动端设备（iOS和Android）。

### 硬件接口

 **客户端硬件接口**：

* 系统应支持通过各种设备（如PC、平板、智能手机）进行访问。客户端设备应支持标准的Web浏览器，并能够流畅运行系统的前端界面。
* 设备应至少具备**2GB内存**和**2GHz处理器**以确保平台的流畅运行。推荐配置：**4GB内存，Intel i5处理器或更高**。

 **服务器硬件接口**：

* 系统应部署在支持**Linux**或**Windows Server**操作系统的服务器上，硬件要求包括至少**8GB内存**、**四核处理器**和**100GB硬盘空间**。
* 数据库服务器应具有冗余备份和数据恢复功能，确保数据的安全性和持久性。

### 软件接口

 **数据库接口**：

* 系统与**PostgreSQL**或**MySQL**数据库进行交互，使用标准SQL协议进行数据查询和管理。数据库应支持ACID事务，以保证数据的完整性和一致性。
* 系统通过JDBC（Java数据库连接）接口与数据库交互，所有数据库操作（如增删改查）应通过标准化的API进行封装，避免硬编码SQL。

 **身份验证接口**：

* 系统需要集成**LDAP**或**OAuth 2.0**认证机制来验证用户身份，确保用户凭证的安全管理。与外部认证服务的交互应遵循相关协议，以支持SSO（单点登录）功能。

 **第三方API集成**：

* 系统应能够与第三方工具（如邮件服务、短信验证系统等）进行接口集成。例如，通过**SMTP**协议发送确认邮件或通过**SMS API**发送短信验证码。
* 所有外部服务的API接口应提供详细的文档，并且与外部服务的通信必须通过加密协议（如HTTPS）进行。

 **文件上传/下载接口**：

* 系统应支持文件上传和下载功能，教师和学生可以上传试卷、成绩单等文件。文件接口应支持常见格式（如PDF、Excel）和最大文件大小限制（如20MB）。

### 通信接口

 **网络协议**：

* 系统与客户端之间的通信应采用**HTTPS**协议，确保数据在传输过程中进行加密，防止信息泄露。
* 后端服务之间的通信应使用标准的**RESTful API**接口，支持**JSON**格式的数据交换。

 **局域网接口**：

* 在公司内部局域网环境下，系统应支持通过**TCP/IP**协议与内部服务（如数据库、认证服务）进行通信。
* 局域网中的服务器应配置防火墙和端口控制，以确保系统的安全性。

 **远程通信接口**：

* 系统应支持通过**VPN**或**SSH**协议进行远程访问，保证系统管理人员能够安全地访问和维护系统。

 **消息队列接口**：

* 为了处理系统的异步任务（如批量发送邮件、成绩分析等），系统应集成消息队列服务（如**RabbitMQ**或**Kafka**），确保任务的高效处理和系统的高并发支持。

## 适用的标准

本节列出了适用于本系统的所有相关标准，包括法律法规、质量标准、易用性标准、互操作性标准等。这些标准确保系统符合行业规定，并提供高质量的服务。

**3.9.1 法律与合规性标准**

1. **GDPR（通用数据保护条例）**：
   * 系统必须遵循**GDPR**规定，保护用户的个人数据隐私。所有涉及个人信息收集、存储和处理的操作，必须获得用户的明确同意，并确保用户能够随时访问、修改或删除其个人数据。
2. **《个人信息保护法》**（中国）：
   * 系统应遵守\*\*《个人信息保护法》\*\*，对用户的个人信息进行保护，确保数据收集、存储、传输和使用符合相关法律要求，避免信息泄露和滥用。
3. **电子签名法**：
   * 对于涉及在线身份验证和电子签名的操作，系统应符合**电子签名法**的要求，确保用户的电子签名具有法律效力，并满足相关法规对数字证书和签名机制的标准。

**3.9.2 质量标准**

1. **ISO 9001**（质量管理体系）：
   * 系统的开发、维护和支持过程应遵循**ISO 9001**标准，确保每个阶段的质量控制和持续改进。所有开发和运维工作应根据标准化流程进行，以确保产品质量和服务稳定性。
2. **ISO/IEC 27001**（信息安全管理体系）：
   * 系统应符合**ISO/IEC 27001**信息安全管理体系的标准，确保系统的安全性和数据保护要求得到有效实施。系统应具备定期进行风险评估和安全审计的机制。

**3.9.3 易用性标准**

1. **IBM CUA（Common User Access）标准**：
   * 系统应遵循**IBM CUA标准**，确保系统的用户界面设计符合一致性、可用性和直观性的要求。这包括但不限于菜单结构、按钮布局、对话框设计等，确保用户能够高效、便捷地完成操作。
2. **WCAG 2.1（Web内容可访问性指南）**：
   * 为了提高系统的可访问性，系统应遵守**WCAG 2.1**标准，确保平台适合视力障碍、听力障碍等特殊需求的用户使用，确保信息内容的可访问性和可理解性。

**3.9.4 互操作性与国际化标准**

1. **W3C标准**（World Wide Web Consortium）：
   * 系统的前端开发应遵循**W3C标准**，确保Web页面的HTML、CSS和JavaScript符合最新的Web标准，保证跨浏览器兼容性，提升用户体验。
2. **Unicode标准**：
   * 系统应支持**Unicode标准**，确保能够处理多种语言的字符集，特别是支持中文、英语、法语、德语等多种语言的显示和输入。

**3.9.5 操作系统与硬件兼容性标准**

1. **操作系统兼容性**：
   * 系统应兼容主流操作系统，包括**Windows**、**MacOS**和**Linux**。开发时应使用跨平台技术（如Web技术、Java等）来保证系统在不同操作系统上的一致性和稳定性。
2. **硬件兼容性**：
   * 系统应支持常见的硬件设备，包括但不限于PC、智能手机、平板电脑等。所有设备必须符合设备性能要求，如支持Web浏览器和相关应用程序。

**3.9.6 网络与通信标准**

1. **IEEE 802.11**（Wi-Fi标准）：
   * 系统在设计和部署时应确保支持**IEEE 802.11**无线网络标准，确保用户能够在无线网络环境中稳定访问系统。
2. **TCP/IP协议**：
   * 系统的网络通信应遵循**TCP/IP协议**，确保不同设备和系统之间的数据传输稳定、高效。