**在线考试系统概要设计说明书**

**学号：**  2017012509

**作者： 王诚昱**

**专业：2017级软件工程**

**指导老师：邹向华**

目录

[1. 引言 3](#_Toc4880811)

[1.1 编写目的 3](#_Toc4880812)

[1.2 背景 3](#_Toc4880813)

[1.3 定义 3](#_Toc4880814)

[1.4 项目风险 4](#_Toc4880815)

[1.5 预期读者和阅读建议 4](#_Toc4880816)

[1.6 参考资料 4](#_Toc4880817)

[2. 总体设计 4](#_Toc4880818)

[2.1 需求规定 5](#_Toc4880819)

[2.2 运行环境 5](#_Toc4880820)

[2.3 基本设计概念和处理流程 6](#_Toc4880821)

[2.4 结构 6](#_Toc4880822)

[2.5 人工处理过程 7](#_Toc4880823)

[2.6 尚未解决的问题 8](#_Toc4880824)

[3. 接口设计 8](#_Toc4880825)

[3.1 用户接口 8](#_Toc4880826)

[3.2 外部接口 8](#_Toc4880827)

[3.3 内部接口 8](#_Toc4880828)

[4. 运行设计 8](#_Toc4880829)

[4.1 运行模块组合 9](#_Toc4880830)

[4.2 运行控制 9](#_Toc4880831)

[4.3 运行时间 9](#_Toc4880832)

[4.4 模块流程图 9](#_Toc4880833)

[5. 系统数据结构设计 10](#_Toc4880834)

[5.1 逻辑结构设计要点 10](#_Toc4880835)

[5.2 物理结构设计要点 12](#_Toc4880836)

[5.3 数据结构与程序的关系 12](#_Toc4880837)

[6. 系统出错处理设计 12](#_Toc4880838)

[6.1 出错信息 12](#_Toc4880839)

[6.2 补救措施 12](#_Toc4880840)

[6.3 系统维护设计 12](#_Toc4880841)

在线考试系统概要设计说明书

# 引言

## 编写目的

在线考试已经成为新时代考试的趋向，越来越多的考试都开始采用无纸化的考试。采用无纸化的考试不仅可以节省纸张，保护环境。同时去除了打印试题需要的时间，减少了出题的环节，节约了出题的时间。同时，在线考试系统将试题、电脑评卷、成绩查询功能集为一体，方便教师的出题和评卷以及学生的答题和成绩查询，这样也能够更加直观的表现出各学生的学习状态。

本说明书阐述了此在线考试系统的基本设计，指出开发软件所使用的方法。

## 背景

1. 本开发软件的名称为学生在线考试系统
2. 项目是面向教师对学生或企业对员工的能力测试，适用于各学校和企业。使得学校或者企业能够更将方便快捷的举行能力测试。
3. 在线考试系统越来越成为现代考试系统不可或缺的一部分，现在人口越来越多，考试规模和数量也越来越大，如果依然沿用原有的纸质考试的形式，需要消耗大量的材料和资源，同时试题的周期较长，效率低下。当今是信息化社会，开发在线考试系统，以此提高效率，减少不必要的能源和资源损耗很有必要。本软件就是本着这样的初衷而开发，同时努力将系统做的更人性化。
4. 开发人员: 王诚昱

## 定义

实现环境: 系统运行的目标软件、硬件环境

实现技术: 系统所采用的软件技术或体系结构，如: Flask、Gunicorn、B/S结构、中间件、Web技术、多层体系结构、后台数据库、ORM等

实现语言或工具: 实现系统最终采用的编程语言或工具包，如python、SQL、Html等

流程图: 由一些特定意义的图形、流程线及简要的文字说明构成，能清晰明确地

表示程序的运行过程

在线考试系统: 该系统是面向学校和企业的，它为学校、企业、教学人员、学生的学习工作提供了方便，同时方便了交流，本系统使用Django框架开发，独立完成其功能。

Flask: Flask是一个使用 Python 编写的轻量级 Web 应用框架。其 WSGI 工具箱采用 Werkzeug ，模板引擎则使用 Jinja2 。Flask使用 BSD 授权。

Flask也被称为 “microframework” ，因为它使用简单的核心，用 extension 增加其他功能。Flask没有默认使用的数据库、窗体验证工具。

## 项目风险

首要风险承担者包括:

* 任务提出者: 王诚昱
* 软件开发者: 王诚昱
* 产品试用者

## 预期读者和阅读建议

本系统概要说明书针对的各种不同的预期读者，包括:

* 程序开发员，建议阅读第2章、第3章、第5章和第6章
* 系统设计师；建议阅读所有章节
* 测试工程师；建议阅读第2章、第3章和第6章

## 参考资料

《面向对象分析与设计(UML)》---------侯爱民 欧阳骥 胡传福 编著

清华大学出版社

《软件工程导论》--------------------张海藩 牟永敏 编著

清华大学出版社

《Python编程:从入门到实践》--------[美]埃里克·马瑟斯 编著

人民邮电出版社

《Flask Web开发：基于Python的Web应用开发实战》---[美]Miguel Grinberg 编著

人民邮电出版社

《Flask Web开发实战》--------------李辉 编著

机械工业出版社

《深入理解 Flask》------------------[美]Jack Stouffer 编著

电子工业出版社

# 总体设计

## 需求规定

#### 对功能的规定

本系统通过网络访问，应该支持100-500人同时在线使用。应该包括出题和做题两种模式。同时有四种成员类型，分别为管理员、出卷人、审卷人和做题人。

#### 性能规定

1. 精度规定

数量值: 精确到小数后一位

时间值: 精确到日，并以yyyy/mm/dd

价格值: 精确到分，并以.XX的形式表示。

本系统要有输入格式检验功能，防止非法格式的输入。并拥有高精度数据的处理能力。

本系统要求有对大文件上传下载的时间提示。

本系统需对每个用户进行严格的权限管理。

本系统对于考试的必须精确到秒。

1. 时间特性要求

每添加、修改、删除一个属性的操作步数不应超过3步。每次响应时间不应超过3秒，数据量较大时不应超过10秒且系统应当及时提示。如果3秒钟系统未响应要有提示。若用户在进行删改操作时系统应有提示。

1. 灵活性

在不同浏览器的环境下必须要保证功能的正常运行。需要有对输入的字符进行判断，防止注入的出现。在不同的操作系统下不应出现乱码。在不同版本的操作系统下能够正常运行。能够迅速的将已有的数据库数据和系统数据转移到另一台设备，并保证不出现不兼容的问题。

## 运行环境

#### 对功能的规定

1. 服务器的软硬件配置

服务器的硬件配置: CPU 1.6G，硬盘256G，内存2G

服务器的操作系统: CentOs 7.4

Web服务器软件: Gunicorn和Nginx（反向代理）

数据库服务器软件: MySQL 或 Oracle

网站开发前台: VsCode

网站开发后台: PyCharm集成开发环境

1. 客户端的软硬件配置

客户端的硬件配置: Pentium 双核或以上，硬盘40G，内存1G以上

客户端操作系统: Windows XP及以上

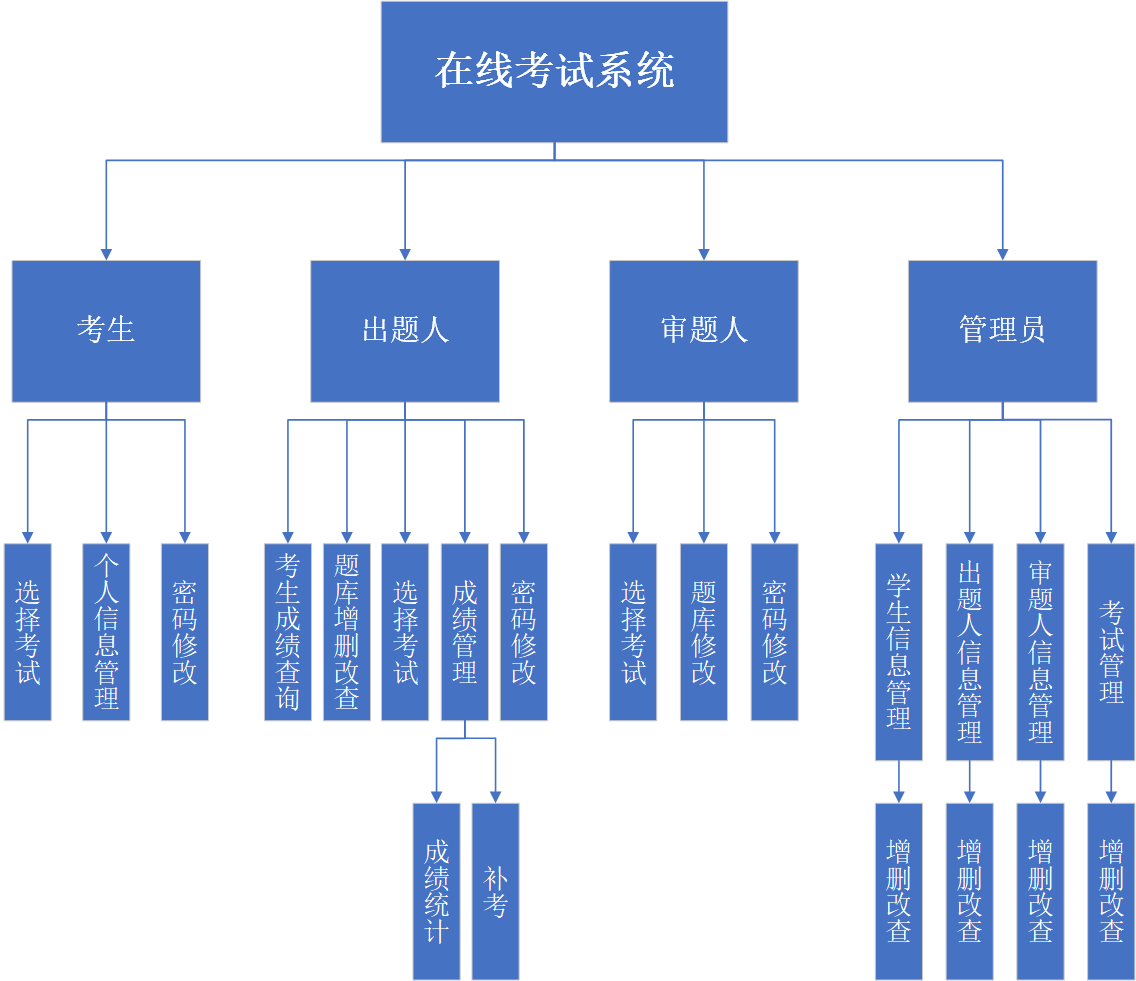
客户端浏览器软件: Google Chrome

客户端浏览器分辨率: 1980\*1080像素

#### 系统的开发工具

利用Flask(Python)、Html、Css、Js、SQL数据库等技术和工具开发设计

## 基本设计概念和处理流程



## 结构

1. 考生子系统

|  |  |  |
| --- | --- | --- |
| **模块名称** | **功能需求** | **程序ID** |
| 成绩查询 | 按照查询条件进行查询  打印成绩单 | cardID |
| 个人信息管理 | 查看个人信息 | studentID |

1. 出题人子系统

|  |  |  |
| --- | --- | --- |
| **模块名称** | **功能需求** | **程序ID** |
| 考试人员管理 | 查看参加考试人员  修改参加考试人员 | teacherID |
| 试题管理 | 发布试题  查看试题  修改试题 | testID |

1. 审题人子系统

|  |  |  |
| --- | --- | --- |
| **模块名称** | **功能需求** | **程序ID** |
| 试题管理 | 修改试题 | testID |

1. 管理员子系统

|  |  |  |
| --- | --- | --- |
| **模块名称** | **功能需求** | **程序ID** |
| 个人信息管理 | 查看个人信息  修改个人信息 | managerID |
| 公告管理 | 发布公告  修改公告 | postID |

## 人工处理过程

省略

## 尚未解决的问题

目前我们觉得最大的问题是还是安全性的问题，比如还存在一些注入和跨站的问题。另外，运行效率也有待提高，大量人员进行考试时可能系统会出现大大小小的问题。

# 接口设计

## 用户接口

本系统的用户接口采用鼠标和键盘，方便用户进行对考试系统的操作。系统的访问主要针对浏览器。对于所有用户，都需要输入正确的用户名和密码才能进入系统，否则，将一直停留在登录页面，等待用户重新输入

## 外部接口

本系统所用的数据库是 MySQL 5.7，使用内部网络和服务器和用户进行交互。可以安装在部分 Linux 和 Windows 上，必须保留 256MB 以上的硬盘空间。

本系统使用的缓存数据库是Redis，使用内部网络和服务器和用户进行交互。可以安装在部分 Linux上（Windows 对 Redis 的支持不是很完整），必须保留 256MB 以上的硬盘空间。

本系统提供以Json形式传输基本课程信息的接口，方便其他的应用调用本应用的基本信息。

## 内部接口

设计共同模块（经常调用的重复代码），提高代码复用程度。

内部是页面和数据库连接，对应的页面输入框和显示框的内容在数据库得到显示。内部接口是指系统内部元素的接口安排。

本系统采用以Json传输数据的方式将所有的数据库数据传输给前端的接口。

# 运行设计

## 运行模块组合

1. 本系统的的各个模块在服务器启动的时候完成所有模块加载工作，随时等候用户的调用。
2. 根据不同的用户权限，调用不同的模块。
3. 管理员可以管理用户对于模块的调用。

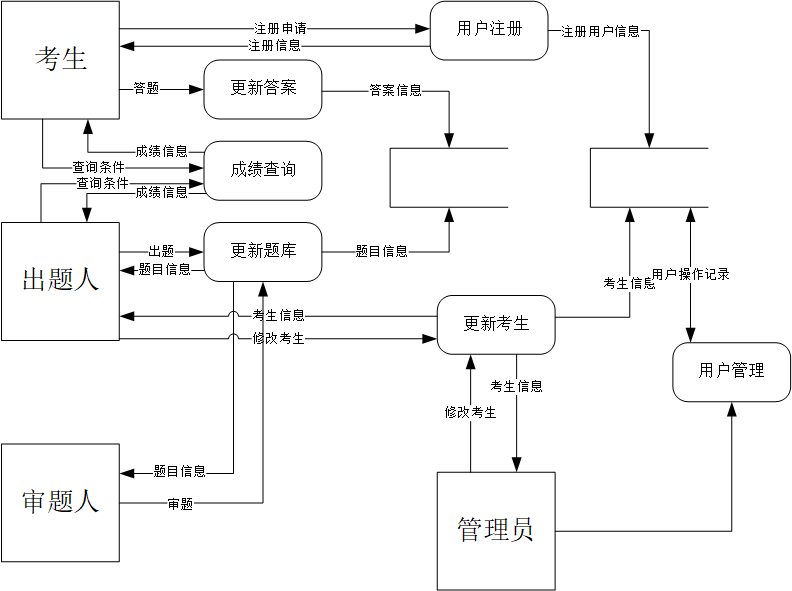
## 运行控制

1. 使用浏览器打开，可以使用鼠标进行相应的操作，如考生对客观选择题的选择。
2. 在浏览器上，可以使用键盘上的键进行一些操作（受到限制，并会经受检查），如考生对主观题的作答。

## 运行时间

1. 考生用户模块，在运行某个动作的时间需求是5秒之内
2. 管理员模块运行模块组合将占用各种资源的时间是在3秒之内
3. 教师管理模块在运行模块组合将占用各种资源的时间是3秒之内
4. 浏览器对服务器的响应的时间是3秒之内

## 模块流程图



# 系统数据结构设计

## 逻辑结构设计要点

1. 学生信息表

|  |  |  |  |
| --- | --- | --- | --- |
| **字段名称** | **数据类型** | **能否为空** | **说明** |
| **studentID** | **自动编号Int** | **NOT NULL** | **准考证（主键）** |
| **studentName** | **Varchar(50)** | **NOT NULL** | **考生姓名** |
| **sex** | **Int** | **NOT NULL** | **考生性别** |
| **cardID** | **Varchar(50)** | **NOT NULL** | **身份证号码** |
| **password** | **Varchar(50)** | **NOT NULL** | **密码** |
| **number** | **Varchar(20)** | **NOT NULL** | **学号** |

1. 出题人信息表

|  |  |  |  |
| --- | --- | --- | --- |
| **字段名称** | **数据类型** | **能否为空** | **说明** |
| **teacherID** | **自动编号Int** | **NOT NULL** | **(主键)** |
| **testID** | **Int** | **Not NULL** | **考试** |
| **teacherName** | **Varchar(50)** | **NOT NULL** | **用户名** |
| **password** | **Varchar(20)** | **NOT NULL** | **密码** |

1. 审题人信息表

|  |  |  |  |
| --- | --- | --- | --- |
| **字段名称** | **数据类型** | **能否为空** | **说明** |
| **supervisorID** | **自动编号Int** | **NOT NULL** | **(主键)** |
| **testID** | **Int** | **Not NULL** | **考试** |
| **supervisorName** | **Varchar(50)** | **NOT NULL** | **用户名** |
| **password** | **Varchar(20)** | **NOT NULL** | **密码** |

1. 管理员信息表

|  |  |  |  |
| --- | --- | --- | --- |
| **字段名称** | **数据类型** | **能否为空** | **说明** |
| **managerID** | **自动编号Int** | **NOT NULL** | **(主键)** |
| **managerName** | **Varchar(50)** | **NOT NULL** | **用户名** |
| **password** | **Varchar(20)** | **NOT NULL** | **密码** |

1. 题库信息表

|  |  |  |
| --- | --- | --- |
| **字段名称** | **数据类型** | **说明** |
| **qID（关键字）** | **自动编号Int** | **库题ID** |
| **subject** | **Varchar(50)** | **科目** |
| **chart** | **Varchar(50)** | **章节** |
| **title** | **Varchar(50)** | **题目标题** |
| **type** | **Varchar(50)** | **题目类型** |
| **ans1** | **text** | **选项一** |
| **ans2** | **text** | **选项二** |
| **ans3** | **test** | **选项三** |
| **ans4** | **test** | **选项四** |
| **key** | **答案** | **回答内容** |
| **pub** | **文本** | **试题是否发布** |

1. 成绩查询表

|  |  |  |  |
| --- | --- | --- | --- |
| **字段名称** | **数据类型** | **能否为空** | **说明** |
| **CardID** | **Varchar(50)** | **NOT NULL** | **身份证号** |
| **studentID** | **Int** | **NOT NULL** | **准考证号** |
| **studentName** | **Varchar(50)** | **NOT NULL** | **考生姓名** |
| **qID** | **Int** | **Not NULL** | **试题** |
| **objectIvesScore** | **Int** | **NOT NULL** | **客观题分数** |
| **subjectIvesScore** | **Int** | **NOT NULL** | **主观题分数** |
| **totalScore** | **Float** | **NOT NULL** | **总分数** |
| **teacherID** | **Int** | **Not NULL** | **出题人** |
| **supervisorID** | **Int** | **Not NULL** | **审题人** |
| **time** | **Date** | **Not Null** | **发布时间** |

1. 公告表

|  |  |  |  |
| --- | --- | --- | --- |
| **字段名称** | **数据类型** | **能否为空** | **说明** |
| **postID** | **Varchar(50)** | **NOT NULL** | **公告编号** |
| **title** | **Int** | **NOT NULL** | **公告标题** |
| **content** | **Varchar(50)** | **NOT NULL** | **公告内容** |
| **time** | **Date** | **Not NULL** | **公告时间** |

## 物理结构设计要点

本系统在MySQL上建立了一个物理数据库，命名为: examSystem, 包括七张表，如下: 考生信息表、出题人信息表、管理员信息表、题库信息表、成绩查询表和公告表，是根据逻辑结构要点分析做的数据库

## 数据结构与程序的关系

1. 考生系统关键联表: 学生信息表、成绩查询表、题库信息表
2. 出题人系统关键链表: 学生信息表、成绩查询表、题库信息表、出题人信息表
3. 审题人系统关键联表: 审题人信息表、题库信息表
4. 管理员系统关键链表: 管理员信息表、公告表、学生信息表、出题人信息表、审题人信息表

# 系统出错处理设计

## 出错信息

1. 提供统一的系统出错用户界面。
2. 出错用户界面应提供详细的出错信息，如: 错误页面，出错原因等。
3. 出错用户界面中应提供解决错误的提示操作步骤。

## 补救措施

定期对数据进行备份。采用硬盘做备份设备，使用MySQL 5.7i 提供的备份功能定期对数据进行备份。一旦系统遭遇到意外破坏，用该备份文件进行恢复，本系统未提供自动恢复功能，只能由系统管理员手动进行恢复。打开MySQL的binlog功能，记录下数据库操作。

## 系统维护设计

定期重启服务器: 服务器优化，包括整理系统空间和性能优化。

系统运行时，应该具有一定的容错能力（如出现错误评判结果后，可以手工评判或者选择重新评判），而且一般的非正常操作不应该影响系统的正常使用。

一旦系统崩溃或出现故障导致系统不能运行，在正常的工作日，应该能够在4个小时内恢复系统正常运行，在节假日等非工作时间，应该能够在24小时内恢复系统正常运行。

设置防火墙服务器，用来防御DDos攻击以及其他类型的攻击。

设置备份数据库，每小时都将主数据库的内容更新到备份数据库，防止数据库内容丢失。