



南開大學
Nankai University

网络空间安全学院
软件工程团队作业

英语六级考试报考系统开发文档

小组成员 1：杨 锐 2011189

小组成员 2：李锦源 2012545

小组成员 3：曹 烨 2012287

小组成员 4：马静雯 2011968

小组成员 5：杨博宇 2011370

2023 年 6 月 2 日

目录

| | |
|--------------------|-----------|
| 1 项目概述 | 2 |
| 1.1 开发背景 | 2 |
| 1.2 项目目标 | 2 |
| 1.3 开发环境 | 2 |
| 1.4 可行性分析 | 3 |
| 1.5 项目计划 | 3 |
| 2 需求分析与系统设计 | 4 |
| 2.1 需求分析 | 4 |
| 2.1.1 业务描述 | 4 |
| 2.1.2 数据需求 | 6 |
| 2.1.3 功能需求 | 9 |
| 2.1.4 性能需求 | 9 |
| 2.1.5 系统运行要求 | 10 |
| 2.2 系统设计 | 10 |
| 2.3 详细设计 | 10 |
| 2.4 数据库设计 | 13 |
| 2.5 UI 设计 | 14 |
| 3 系统测试 | 17 |
| 3.1 测试环境 | 17 |
| 3.2 功能测试 | 17 |
| 3.3 性能测试 | 17 |
| 4 项目管理 | 18 |
| 4.1 参与人员及分工 | 18 |
| 4.2 项目进展记录 | 18 |
| 4.3 项目管理工具 | 19 |
| 5 用户手册 | 19 |

1 项目概述

1.1 开发背景

英语六级考试是全国性的英语水平考试，是许多考生展示和提高英语能力的重要途径。为了更好地管理考生信息，方便教师和考试管理人员进行考务管理、成绩统计等工作，本项目旨在开发一个基于 Web 应用的英语六级考试报考系统。

该系统主要面向三类用户：考生、教师和考试管理人员。考生可以通过系统完成信息注册、试题答题、报名缴费、查询考试结果等操作；教师和考试管理人员可以通过系统进行试题录入、自动阅卷和人工阅卷、考试结果查询等操作。

本项目基于 Django 框架进行开发，采用 Python 语言编写，实现了上述功能，并提供了良好的用户体验和数据安全保障。

1.2 项目目标

本项目旨在开发一个完善的英语六级考试报考系统，为广大考生提供方便快捷的服务，实现以下目标：

1. 实现考生信息的注册、管理和查询，提供在线报名和缴费功能，方便考生进行英语六级考试的报名和参加。
2. 实现英语六级考试的试题录入，包括单选题、多选题和填空题等，保证试题难度适度、覆盖面广，以及防范作弊等方面的要求。
3. 实现在线考场答题，支持多种题型和考试方式，提供自动阅卷和教师人工阅卷两种评分方式，保证考试评分准确和公正。
4. 实现考试结果的查询和统计，包括成绩查询、排名统计和历史记录等，方便考生、教师和考试管理人员进行考务管理和成绩分析。
5. 提供良好的用户体验和数据安全保障，包括网站界面友好、操作流程简单、数据传输加密、信息备份等方面的要求，保证系统的可靠性和稳定性。

通过实现以上目标，本项目将为英语六级考试的考生、教师和考试管理人员提供优质的服务，促进英语水平的提高和考试质量的提升。

1.3 开发环境

本项目开发环境如下：

1. 操作系统：Windows 10
2. 开发工具：PyCharm 2021.1.3
3. 编程语言：Python 3.8.5
4. Web 框架：Django 3.2.4
5. 数据库：MySQL 8.0.25、Navicat 15

6. 浏览器：Microsoft Edge 版本 113.0.1774.57 (正式版本) (64 位)
7. 前端技术：HTML5、CSS3、JavaScript、jQuery、Bootstrap 4
8. 版本控制：Git 2.31.1
9. 代码托管平台：GitHub

以上环境经过测试基本保障了开发过程的高效性和稳定性，同时也满足了 Web 应用开发的基本需求。特别是采用了 Django 框架和 MySQL 数据库，实现了 Web 应用的快速开发和数据存储等关键功能，提高了开发效率和代码可维护性，同时，鉴于小组的协作性，我们采用 github 平台及时对个人所做的代码的修改进行上传，便于小组的整体代码管理以及项目的推进。

1.4 可行性分析

本项目的可行性分析主要从技术、经济、市场等方面进行分析，结论如下：

1. 技术可行性：本项目基于 Python 语言和 Django 框架开发，同时采用了 MySQL 数据库和前端技术，这些技术符合 Web 应用开发的基本要求，并且具有较好的成熟度和稳定性。同时，由于 Python 和 Django 具有开发效率高、代码可维护性好等优点，所以本项目在技术上是可行的。
2. 经济可行性：本项目的实现需要购买服务器、域名、SSL 证书等相关费用，并需要支付开发人员的工资和维护费用等。同时，由于本项目具有一定的商业价值和社会意义，可以通过收取考生缴费、销售广告位等方式获得一定的经济收益。因此，在经济上本项目也是可行的。
3. 市场可行性：英语六级考试是全国性的英语水平考试，每年都有大量的考生参加考试。同时，随着互联网技术的普及，越来越多的人开始倾向于在线考试和线上学习。因此，本项目具有很好的市场前景和用户需求，有望在未来获得广泛的应用和推广。

综上所述，本项目具有技术可行性、经济可行性和市场可行性，可以开展后续的实施工作。

1.5 项目计划

根据项目的目标 and 需求，小组成员制定如下的项目计划：

1. 需求分析和规划：

首先，根据所要求的完成目标“英语六级考试报考系统”，梳理该项目需求，明确所需要实现功能和性能，即考生信息注册以及增删改查、试题录入、报名缴费、线上考场进行答题、自动阅卷和教师阅卷、考试结果查询等，之后制定详细的项目计划和开发计划，明确主次。同时，我们需要参考现有的六级报名相关网站等进行分析，确保项目基础走向以及后期实现的可行性和有效性。

2. 系统设计和开发：

在这个阶段，我们将进行系统架构设计，选定 Django 框架进行开发、完成数据库设计、UI 设计、编码和测试等工作，确保系统的功能和性能等达到预期要求。同时，我们还将进行代码的版本控制和文档编写等工作，方便后续的开发和维护。

3. 测试和调试：

在这个阶段，我们将进行系统测试和调试工作，确保系统的质量和稳定性等符合预期。同时，我们还将进行用户反馈和 BUG 修复等工作，保证系统的用户体验和可靠性。

综上所述，本项目计划将分成 3 个阶段完成。在每个阶段结束时，我们将进行阶段性评估和总结，小组进行讨论并不断完善，集思广益以确保项目进展顺利并达到预期目标。

2 需求分析与系统设计

2.1 需求分析

对整个项目的需求分析 [1] 将会围绕如下的几个部分展开：

1. 开发背景及目标
2. 业务描述
3. 数据需求（数据流程图及数据字典）
4. 功能需求
5. 性能需求
6. 系统运行要求

上述模块中，开发背景及描述在本文档的第 1 部分已做说明，此处不再重复。

2.1.1 业务描述

整个系统的业务流程图将参照如下进行设计：

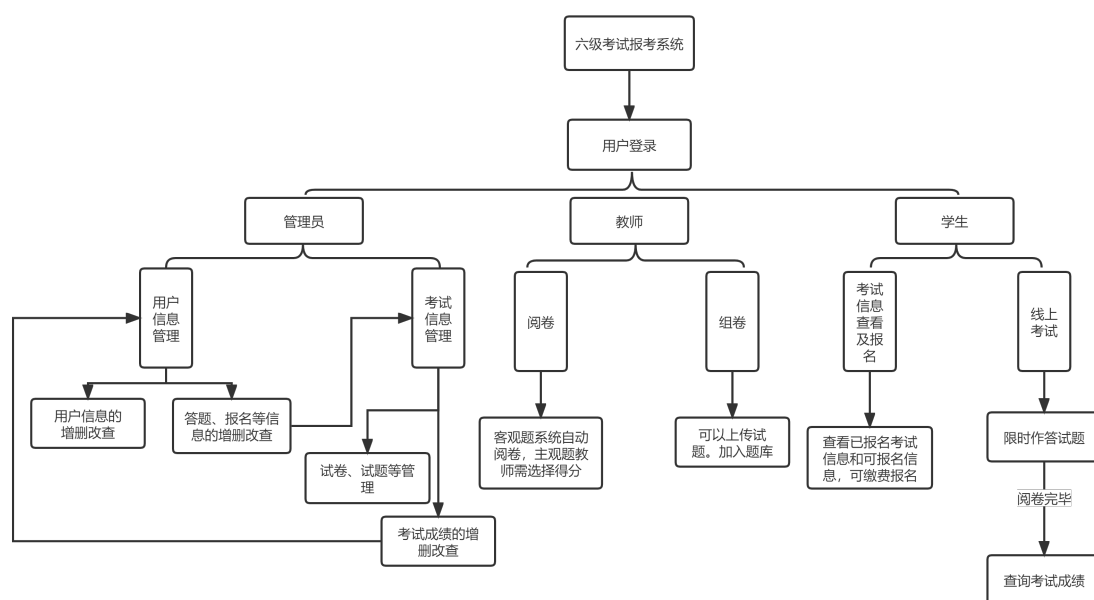


图 2.1: 系统整体业务流程图

系统的主要总体业务流程为：进入主界面 → 选择身份登录 → 进入相应的界面。然后根据不同的身份可以进行不同的业务。下面将按照系统的两个主要用户：管理员和学生，分别给出详细的子业务流程。

管理员相关业务的流程图如下：

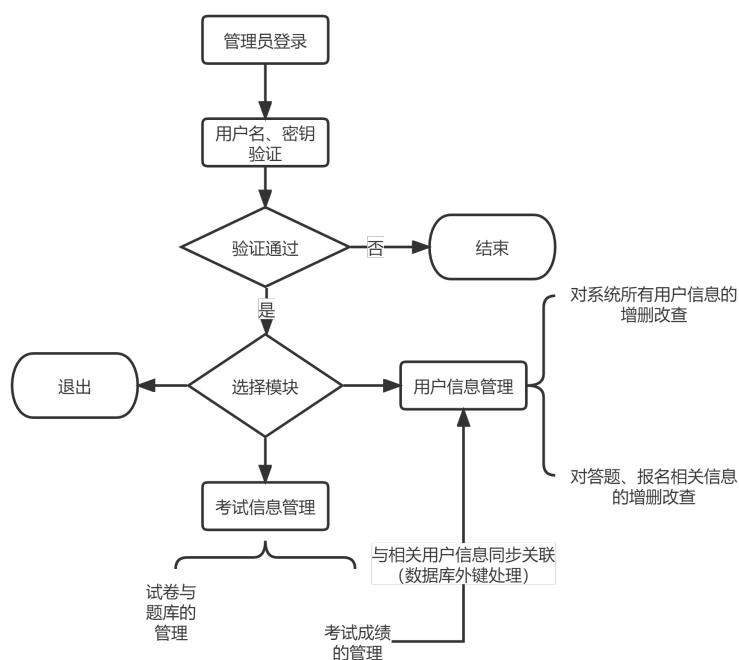


图 2.2: 管理员子业务流程图

管理员具备系统的最高级别权限，经身份验证通过后，管理员可以对系统所有的数据库进行增删改查。即管理员可以对所有用户信息进行管理（当然，为了保障用户数据的安全性，我们对用户的密码进行了加密处理，因此管理员用户在数据库中也不会看到用户密码的明文），也可以对考试、试题相关的数据进行管理。此外，如果需要对考试相关信息进行修改，结合数据库外键的相关要求，我们会对用户信息中涉及到的内容进行同步处理。

学生相关业务的流程图如下：

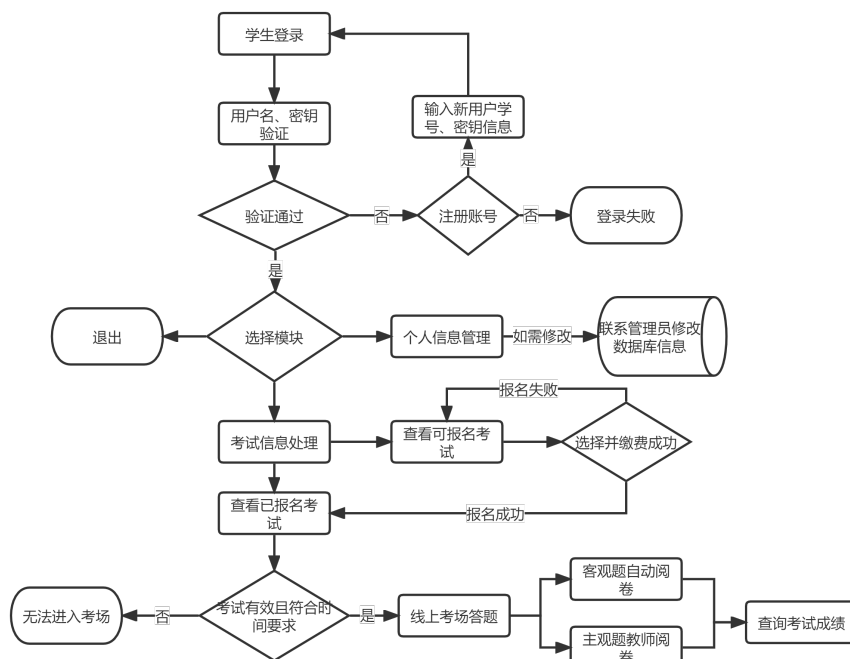


图 2.3: 学生子业务流程图

学生是该系统的最主要的目标用户，需要首先注册账号，通过身份验证后即可进入功能模块。在功能模块中，学生可以对自己个人的信息进行管理。也可以选择报名考试，完成缴费后即成功报名。成功报名后，在系统规定的时间内线上参与答题，完成提交后，系统将会自动对客观题进行阅卷，主观题则需要等待教师用户进行判分。全部判分结束后，学生登录系统可以查看到自己本次的考试成绩。

2.1.2 数据需求

首先是系统的顶层数据流图：

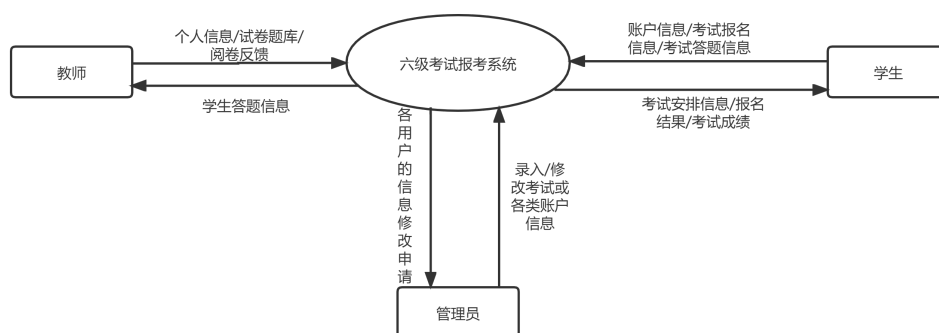


图 2.4: 系统顶层数据流图

从功能角度划分，主要包含了用户信息管理、考试报名、线上答题等功能模块，下面是本系统核心功能：考试报考的 0 层数据流图：

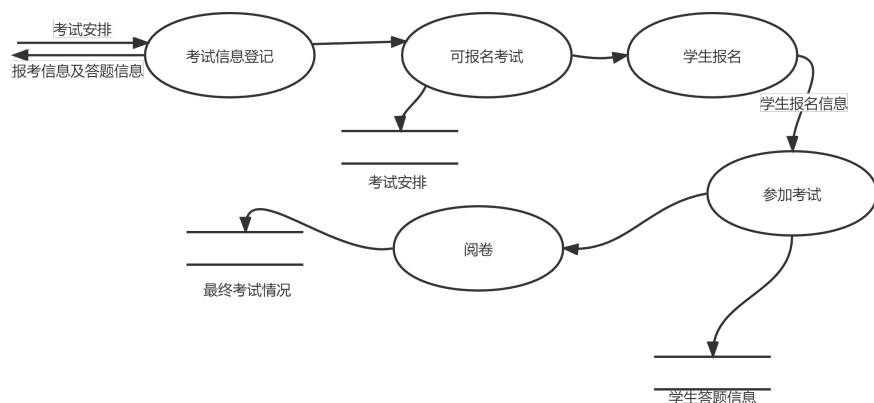


图 2.5: 报考功能 0 层数据流图

若继续划分，也可以从用户信息管理、试卷录入等功能继续画出 1 层数据流图，下面以试卷录入为例，画出 1 层数据流图：

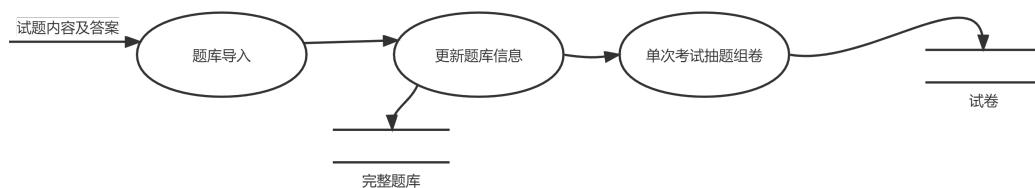


图 2.6: 试卷录入功能 1 层数据流图

而针对本系统相关的数据字典，大致信息如下（数据库设计类图在文档的 2.4 部分给出）：

学生表 (*Student*)

- **sid** (学号) [Primary Key]: 字符串，长度为 12，表示学生的学号
- **pwd** (密码): 字符串，长度为 20，表示学生的密码

教师表 (*Teacher*)

- **tid** (职工号) [Primary Key]: 字符串，长度为 12，表示教师的职工号
- **name** (姓名): 字符串，长度为 20，表示教师的姓名
- **pwd** (密码): 字符串，长度为 20，表示教师的密码

试卷表 (*TestPaper*)

- **id** (序号) [Primary Key]: 自增整数，表示试卷的序号
- **title** (名称): 字符串，长度为 40，表示试卷的名称
- **time** (考试时长): 整数，表示考试的时长（单位为分钟）
- **examtime** (考试时间): 日期时间，表示考试的具体时间

- **fee** (费用): 整数, 表示考试的费用, 可以为 null

题目表 (*QuestionBank*)

- **id** (序号) [Primary Key]: 自增整数, 表示题目的序号
- **paper** (试卷): 外键, 关联到试卷表的 id 字段, 表示题目所属的试卷
- **title** (题目): 文本, 表示题目的具体内容
- **qtype** (题目类型): 字符串, 长度为 40, 表示题目的类型, 可选值为"单选"或"主观"
- **a** (A 选项): 字符串, 长度为 40, 表示单选题的选项 A
- **b** (B 选项): 字符串, 长度为 40, 表示单选题的选项 B
- **c** (C 选项): 字符串, 长度为 40, 表示单选题的选项 C
- **d** (D 选项): 字符串, 长度为 40, 表示单选题的选项 D
- **answer** (正确答案): 文本, 表示题目的正确答案
- **difficulty** (难度): 字符串, 长度为 10, 表示题目的难度, 可选值为"easy"、"middle"或"difficult"
- **score** (分值): 整数, 表示题目的分值

考试预订表 (*Book*)

- **id** (序号) [Primary Key]: 自增整数, 表示预订信息的序号
- **sid** (学号): 外键, 关联到学生表的 sid 字段, 表示预订考试的学生学号
- **tid** (序号): 外键, 关联到试卷表的 id 字段, 表示预订考试的试卷序号
- **paid** (是否付款): 布尔值, 表示是否已付款, 可选值为 0 (否) 或 1 (是)

答题记录和评分表 (*Answer_Score*)

- **id** (序号) [Primary Key]: 自增整数, 表示答题记录和评分的序号
- **sid** (学生): 外键, 关联到学生表的 sid 字段, 表示学生的学号
- **tid** (试卷): 外键, 关联到试卷表的 id 字段, 表示答题的试卷序号
- **qid** (题目): 外键, 关联到题目表的 id 字段, 表示答题的题目序号
- **title** (题目): 文本, 表示答题的题目内容
- **answer** (答案): 文本, 表示学生的答案
- **score** (得分): 整数, 表示题目的得分, 默认为 1000 (注意: 该字段有可能为 None)

学生成绩表 (*Record*)

- **id** (序号) [Primary Key]: 自增整数, 表示学生成绩的序号
- **sid** (学号): 外键, 关联到学生表的 sid 字段, 表示学生成绩的学生学号
- **tid** (考试): 外键, 关联到试卷表的 id 字段, 表示学生成绩的考试序号
- **grade** (成绩): 浮点数, 表示学生的成绩

2.1.3 功能需求

该系统的主要功能包含了：

- 账户信息管理
- 试卷及考试信息管理
- 学生六级考试报名
- 学生在线考试
- 客观题自动阅卷 + 主观题教师阅卷
- 考试成绩反馈

不同的功能针对着不同的用户类型，详细的流程在文档的 2.1.1 部分已经说明，不再重复。

2.1.4 性能需求

如果需要实际大规模应用，系统应当具备如下的性能需求：

响应时间：

- 系统登录页面的响应时间应在 2 秒以内，避免用户长时间等待的情况出现。
- 由于考试需要计时，系统试卷加载的响应时间应在 3 秒以内，尽可能避免因系统延迟而对考试造成影响。

并发性能：

- 系统应支持大量名学生同时在线进行考试，以满足大规模考试的需求。
- 在高并发情况下，系统应保持稳定，不出现明显的性能下降或系统崩溃，数据库错乱等问题。
- 系统应具备合理的并发控制机制，防止恶意攻击或异常行为对系统性能的影响。

可扩展性：

- 系统应支持灵活的扩展，以适应未来用户量的增长和功能的扩展需求。
- 数据库设计应考虑到数据量的增加，以保证系统在数据量增长时的性能稳定。
- 系统的功能架构应具备良好的可拓展性，以备未来新的功能的加入和修改。

资源利用：

- 系统应合理利用服务器资源，尽量减少内存和处理器的占用率。
- 数据库查询和更新操作应进行性能优化，以提高系统的响应速度和资源利用率。
- 系统应定期进行性能监测和优化，确保资源的有效利用和系统的稳定性。

实际上，由于本次设计仅用作课程作业，并不涉及到商用推广，我们的系统搭建于本机服务器，在性能方面主要指进行了针对代码可拓展性和代码规范的优化，也包含了一定的数据库并发控制和触发器。由于仅用于本地，因此也不会存在响应过慢和大量人员并发等问题，系统具备良好的性能。

2.1.5 系统运行要求

若将本次设计的作品进行发布，其需要的系统配置大致如下：

- 硬件：需要具备连接互联网的电脑；
- 操作系统：计算机应搭载 Windows XP 及以上版本或 mac OS 10.0 及以上版本；（为了保障系统的安全性，我们不建议在已经被官方停止维护的系统上使用该系统）。

在实际开发阶段，我们的要求为：

- 电脑已经安装好 python3.7 及以上版本的环境；
- 配置好 mysql 数据库环境及 pymysql

2.2 系统设计

对整个项目的系统设计 [2] 的说明将会以 uml 图的形式展开，主要包括：

- 类图
- 活动图
- 用例图
- 顺序图及协作图
- 状态图
- 构件图
- 部署图

在本文档中我们将会选取最为关键的用户图、类图、顺序图进行介绍，对整个系统的设计作说明。

2.3 详细设计

本系统的用例图 [3] 如下：

表 2: 报名缴费描述

| | |
|-------|---|
| 用例名称 | 考生报名缴费 |
| 用例标识号 | 103 |
| 参与者 | 考生 |
| 简要说明 | 考生选择可报名考试, 进行报名缴费 |
| 前置条件 | 考生已经成功登录系统且该考试目前允许报名 |
| 基本事件流 | 1. 考试选择想要报名的考试, 点击“提交” 2. 系统校验, 完成响应 3. 校验通过, 提示考生进行缴费 4. 考试完成缴费, 报名成功, 用例结束 |
| 其他事件流 | 考试选择暂时不缴费, 系统给出一定的等待时间 |
| 异常事件流 | 考生超时未缴费, 报名失败。 |
| 后置条件 | 考生完成报名, 可在相应时间线上参加考试 |

表 3: 考试答题描述

| | |
|-------|--|
| 用例名称 | 在线考试答题 |
| 用例标识号 | 104 |
| 参与者 | 考生 |
| 简要说明 | 考生在考试时间内线上答题完成考试 |
| 前置条件 | 考生已经成功报名该考试且在该考试允许的时间内 |
| 基本事件流 | 1. 考试选择想要进行的考试, 点击“开始考试” 2. 系统校验并从题库组卷, 完成响应 3. 考生在规定时间内进行答题 4. 考试时间到或用户完成所有的提交, 用例结束 |
| 其他事件流 | 无 |
| 异常事件流 | 考生超时未提交, 系统自动交卷。 |
| 后置条件 | 考生完成答题, 等待阅卷结束后可以查询成绩 |

表 4: 系统管理描述

| | |
|-------|---|
| 用例名称 | 系统管理 |
| 用例标识号 | 302 |
| 参与者 | 系统管理员 |
| 简要说明 | 管理员对系统的各项数据进行管理 |
| 前置条件 | 已经成功登录管理员账号 |
| 基本事件流 | 1. 管理员选择需要修改的数据库 2. 系统校验, 完成响应 3. 管理员完成数据的增删改查并提交 4. 系统存储并保存修改记录, 用例结束 |
| 其他事件流 | 无 |
| 异常事件流 | 修改不符合数据库要求, 修改失败。 |
| 后置条件 | 数据库完成更新 |

结合考生这一系统的主体, 其整体操作的顺序图如下所示:

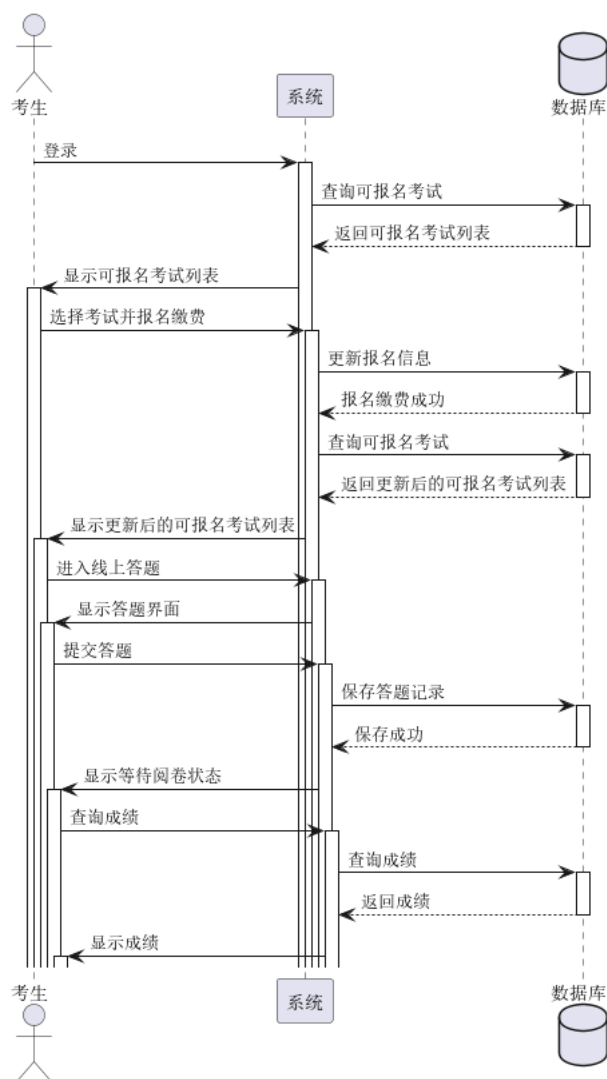


图 2.8: 考试操作顺序图

2.4 数据库设计

数据库的整体设计在需求分析部分已经给出了对数据字典和数据流图的详细说明，此处仅展示类图：

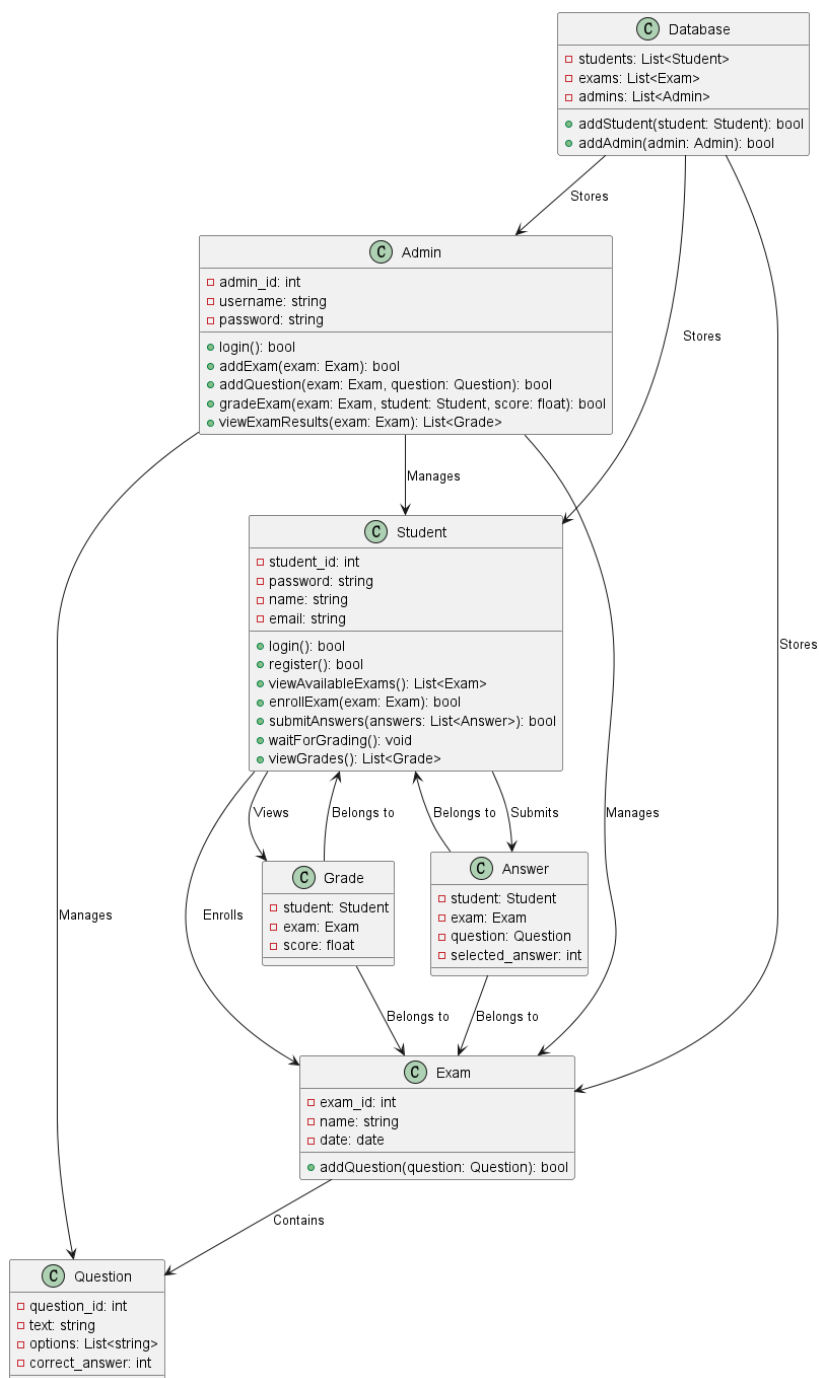


图 2.9: 数据库类图

2.5 UI 设计

系统的首页界面基本参照了四六级考试的官网进行 UI 设计，界面如下：



图 2.10: 首页

为了改善用户的使用体验，我们也增加了鼠标移动的交互动画，例如下图：



图 2.11: 首页交互动画样例

其它界面的设计则基本以简洁、清晰为原则。例如管理员操作界面大致展示如下：

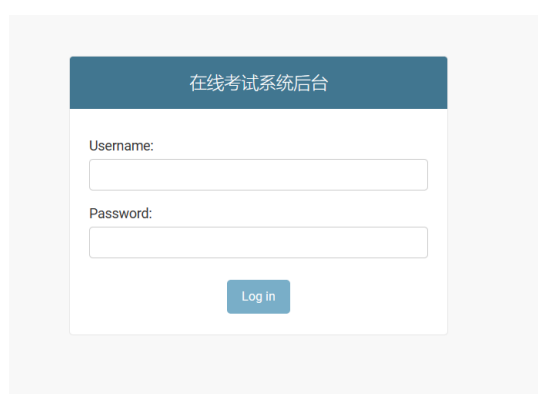


图 2.12: 管理员登录界面

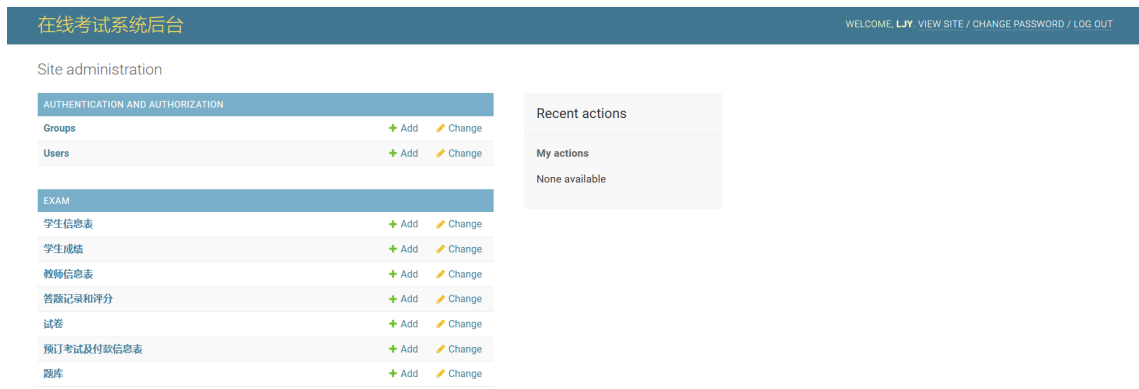


图 2.13: 管理员后台界面

对于考生操作的界面，大致如下：

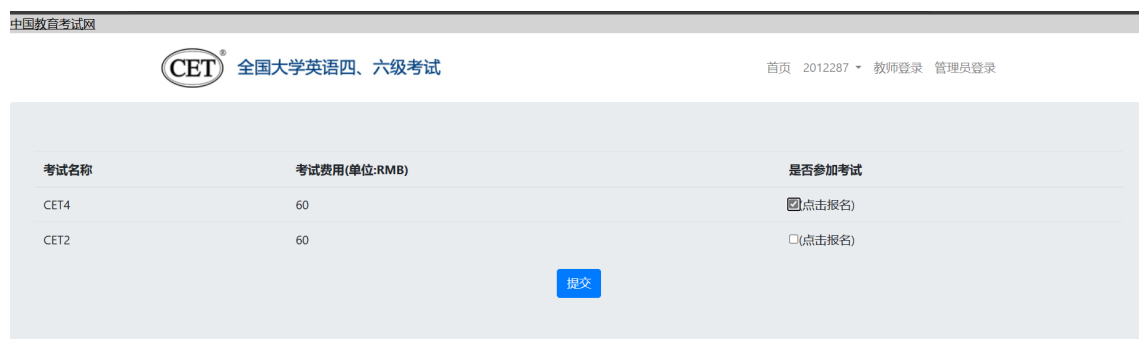


图 2.14: 考生报名界面



图 2.15: 考生答题界面



图 2.16: 报名缴费界面

3 系统测试

3.1 测试环境

- 服务器运行环境：使用 Pycharm 基于 Django 框架运行本地服务器
- 数据库：mysql、Navicat 15
- 浏览器：Microsoft Edge 版本 113.0.1774.57 (正式版本) (64 位)
- Python 版本：3.7 及以上

3.2 功能测试

我们对系统的功能测试主要基于本地服务器进行，大致流程如下：

1. 本地运行服务器，访问 <http://127.0.0.1:8000/index> 进入网页（测试结果：成功运行）
2. 测试主页 UI 交互（测试结果：交互正常）
3. 测试各用户登录（测试结果：通过）
4. 测试学生报名、缴费相关功能（测试结果：通过）
5. 测试学生答题、计时相关功能（测试结果：可以正确完成答题、提交，计时结束可以自动提交）
6. 测试阅卷、试题上传等功能（测试结果：可以正确运行）

3.3 性能测试

结合在需求分析阶段的性能需求分析，我们对数据库的并发控制和系统延迟控制进行分析。经测试，我们通过触发器等设置可以正确进行数据库的管理，并在系统按键点击后，响应时间小于 1 秒，符合性能要求。

4 项目管理

4.1 参与人员及分工

我们小组首先在集体讨论后明确整体开发框架和目标，并在此后分工合作完成开发，主要分工如下：

- 杨锐 2011189：参与框架的学习与搭建，完成系统的需求分析和系统设计，进行系统的功能和性能测试。
- 李锦源 2012545：参与框架的学习与搭建，并主要负责系统的前端开发工作及改进工作。
- 曹烨 2012287：参与框架的学习，设计系统的数据库结构，实现系统所需的数据表，并主要负责系统的后端开发及改进工作。
- 马静雯 2011968：参与项目的需求分析和框架学习，设计了用户手册，并主要负责项目的展示汇报工作。
- 杨博宇 2011370：参与框架学习，并主要负责系统的前端开发及改进工作。

4.2 项目进展记录

项目主要借助 git 进行进展记录，相关记录如下图所示：

【TODO】缺少 git 提交记录，此处暂空

项目整体的开发流程参照如下的软件工程一般开发步骤：

1. 可行性分析及需求分析：

- 明确任务总体目标，定义系统的目标和范围。
- 分析、明确用户的需求和期望。
- 结合可行性分析、业务描述、数据需求、功能需求、性能需求及系统运行要求进行分析。

2. 系统设计及框架搭建：

- 定义系统的整体结构和组件之间的关系。
- 明确使用 Bootstrap+Django 的框架进行开发，并搭建所需环境。
- 设计数据库结构和类图、用例图。
- 明确界面 UI 设计等详细设计。

3. 模块设计与编码实现：

- 将系统拆分为多个模块或组件，为每个模块定义接口和功能。
- 明确前后端开发任务与对接方式。
- 进行前、后端编码开发，注重代码的可拓展性与规范性。

4. 系统测试与改进：

- 结合前期需求分析与系统设计文档，对编码效果进行测试。

- 测试基本功能是否得以满足。
- 测试系统的性能是否良好。
- 结合测试结果进行进一步改进与完善。

5. 系统维护（计划）：

- 仅为针对实际发布系统的必要流程。
- 定期进行系统维护和升级。
- 收集用户反馈和需求，进行改进和优化。
- 本次大作业不涉及实际推广使用，因此这一步忽略。

4.3 项目管理工具

项目管理主要使用了 github 群组进行开发管理，并结合了微信群进行交流探讨。

5 用户手册

- 针对学生：TODO 等到项目整体做完了再写
- 针对教师
- 针对管理员

参考文献

- [1] Leszeka. Maciaszek. 需求分析与系统设计. 需求分析与系统设计, 2003.
- [2] 何方 and 马武刚. 面向对象设计 uml 技术在考试系统设计中的实践. 安阳师范学院学报, (5):3, 2006.
- [3] 张[国]. 以 uml 用例图为基础的系统需求分析. 现代计算机 (专业版), 2002.