****

**软 件 学 院**

**软件工程报告**

**项目名称：**  在线考试系统的分析与设计

**专 业：** 软件工程（互联网应用开发与优化）

**班 级：** RB软工互161

**学 号：** 201619160132

**学生姓名：** 杨皓

**2018年 12 月 16 日**

# 在线考试系统的分析与设计

## 项目背景

在当今信息时代,计算机技术与网络技术越来越广地应用于各个领域,改变着人们的学习、工作、生活乃至思维方式,也引起了教育领域的重大变革。将计算机与网络技术应用于现代高等教育中,是现代高等教育发展的需要,也是改革教育模式,提高学校教学效果和教学效率、提高科研和管理水平的必要手段。目前的一个发展趋势是采用大规模试题库的计算机网络考试模式 随着计算机网络在生产生活、科技教育中的普及，传统试的考试出卷、答卷方式以及学绩管理正发生着巨大的变革，因此,如何使考试过程变得方便、高效、快捷、公正,是现代教育的一个重要课题。网络考试系统是传统考场的延伸,它可以利用网络的无限广阔空间,随时随地对学生进行考试,加上数据库技术的利用,大大简化了传统考试的过程。因此网络考试系统是电子化教学不可缺少的一个重要环节。有趋势表明，考试的标准化、计算机网络化已当今考试的发展方向.

目录

[在线考试系统的分析与设计 1](#_Toc532758379)

[项目背景 1](#_Toc532758380)

[第1章需求分析 3](#_Toc532758381)

[1.1会议记录 3](#_Toc532758382)

[【场景】 3](#_Toc532758383)

[【人物】 4](#_Toc532758384)

[【对话】 4](#_Toc532758385)

[1.2同类产品分析 5](#_Toc532758386)

[1.3需求总结 8](#_Toc532758387)

[1.3.1项目中的利益相关者 8](#_Toc532758388)

[1.3.2.不同利益相关者的需求 8](#_Toc532758389)

[1.4项目中的主要功能 9](#_Toc532758390)

[1.5系统用例图 10](#_Toc532758391)

[1.6用例规约 11](#_Toc532758392)

[1.6.1用例：创建题库 11](#_Toc532758393)

[1.6.2用例：防断网闪退 13](#_Toc532758394)

[1.6.3用例：防作弊 14](#_Toc532758395)

[1.7数据库设计 18](#_Toc532758396)

[1.7.1数据库E-R图 18](#_Toc532758397)

[1.7.2数据库关系模式 19](#_Toc532758398)

[1.7.3数据库表 20](#_Toc532758399)

[1.8用户界面 23](#_Toc532758400)

[第二章 测试 25](#_Toc532758401)

[2.1测试用例 25](#_Toc532758402)

[第三章 总结 27](#_Toc532758403)

# 第1章需求分析

## 1.1会议记录

【场景】中原工学院软件学院三号宿舍楼103室，在软件工程课需求建模部分结束后。

【人物】高通、杨皓、张继云，在线考试系统的分析与设计项目组成员。

### 【对话】

**高通：**经过上节课的讨论，我觉得咱们的挂号系统的设计与分析不太可行，问题和漏洞太多了，还有很多不确定性的东西，无法合理解决，而且同学们提出的疑问确实句句见血，非吾等弱辈所能完成，由于此等种种，咱们换个项目研究吧，比如在线考试系统的设计与分析就比较切合实际，其中的问题，相对于前一个项目应该易于解决，你们说呢？

**杨皓：**我觉得你说的可行，问问张继云吧，看看他意下如何。

**高通：**云哥，我俩商量了一下觉得在线考试系统比较可行，你感觉怎么样？

**张继云：**要我看，这个可行，毕竟那个挂号经讨论之后我越来越感觉有点扯了。不过既然，选择解决在线考试系统的分析与设计的问题，那首先得分析与这个系统相关的一类人有哪些需求。还要站在不同的人的角度上考虑在他的情景里都有哪些需求。怎么才能够保证系统里的各项功能非常的人性化，让人有种“爱不释手”的感觉，我认为一件产品出来之后，它的用户能够对它产生依赖，是这个产品产生的意义所在，当然了，这仅仅是在产品本身的角度的看，暂不考虑对用户的不良影响。你们觉得呢？

**杨皓：**那既然这是一个在线考试的系统，最基本的也是没必要一提的就是注册、登录的功能了吧。此项功能的存在是针对于供应商的，这应该是供应商以此盈利的入口吧，获取用户的数据，就算是开发者也是无法抗拒的吧。但是很多网站都有注册、登录，但是一般很少有人真正去注册，很多“游客”都不愿意注册，不愿意留下痕迹，所以，让绝大多数“游客”们留下他们的数据才是灵魂所在，感觉设计时，首先注册、登录模块就要精心考虑一番，比如增加一个“游客登录”的入口，让有需求的“游客”体验一把。

**高通：**你说的那个是面向的受众比较广的一类考试系统，我觉得目前在校阶段，我们就应弄一个专门针对广大师生的系统，主要的参与者就包括教师、学生和管理员。其中学生最基本的，选择考试，查看成绩包括历次成绩，教师无非就是组卷，发布考试，阅卷，至于管理员，作为系统的管理者肯定是拥有包括教师和学生的所有权限。

**张继云：**既然是深入分析和设计，那系统的各个参与者的功能就该细细考究了，学生看完成绩，也应该有正确答案的解析，以便于深入学习，根据其错题的类型推荐相应题型的练习；教师也可以先建立题库，录入不同类型的考题，便于丰富试卷类型和最大程度的覆盖考点。对于练习题，教师也应该根据不同题型，录入相应的习题，而且习题数量不能少，一定要多，更有利于学生提高自己；对于，试卷的批改，那肯定是考试结束后实时出分的，最好，也能在每道题后面智能地显示其考点。

**杨皓：**教师组卷的时候，可以根据自己的需求，选择自动组卷或者是手动组卷了，以便于控制考试范围；至于管理员，作为系统的最高管理者，这些权限都具有，他所关注的应该是学生的信息的真实性和准确性还诚信考试方面，对于学生个人素养的考验。作弊这种事情自古以来就无可避免，所以我们就不得已加上防作弊的功能，实时监测，学生的答题情况，这个功能在现在这浮躁的大环境下将会非常的实用，让那些投机取巧者无计可施。另外，还有个最重要的问题，就是异常登出的问题，学生在考试过程中由于系统或其他非正常因素的原因有可能会遇到，诸如“闪退”、“网络异常”等问题导致的异常退出考试的情况，对此，我们可以加入对异常登出的检测和处理，当真的出现异常登出的情况时，保存考生或其他参与者的操作状态，等系统恢复正常后，继续未完成的操作。

**张继云：**暂时也就想到这么多方面的东西，有不足的，就等待后期的发现和完善吧，你们感觉如何？

**高通：**可以，毕竟没有经验，后期慢慢补充完善吧，就先这样吧。

## 1.2同类产品分析

下面对同类产品“考试星”做一下分析：

首先是“考试星”的首页，如下图1-1所示：



图1-1

如上图所示，“考试星”的网站首页从菜单可以看到有功能介绍、团队介绍以及使用文档。首页提供了用户注册的渠道，而且注册方式简单，仅需提供手机号就可以实现注册。在注册的下方则是提供了使用“考试星”的方法以及使用场景。

接下来是“考试星”的功能分析，如下图1-2所示，是“考试星”的功能页面：



图1-2

如上图所示，为“考试星”的功能展示页面，可以看到在该页面上详细描述了“考试星”对用户开放的功能。总结其功能如下：

1. 支持多种题型：单选、多选、判断、填空、简答题等
2. 支持选题+组卷，可以自由选择题目进行组卷，也可以使用系统生成试卷
3. 无需人工批改，由系统自动批改。
4. 考后自动发放相关证书。
5. 使用练习模式，边练习边反馈，在练习的同时，及时给出正确答案及解析，及时反馈，及时修正，使学习更高效。
6. 错题本进行知识积累，按时复习。在平台上所做错的所有题目都会记录下来以供学生复习使用。
7. 易错知识点总结，错题率统计，方便用户了解知识难点和重点。
8. 支持考生答卷导出。
9. 支持上传各种格式的课件，实现在线学习在线考试。
10. 防止作弊，考前人脸识别，限制答题次数、限制切屏次数，“考场”摄像监控，随机拍照
11. 防止断网闪退，断网闪退能够自动恢复到考试状态，恢复已经进行的考试内容以及已经填写的答案。
12. 支持查看系统使用文档。

最后看看“考试星”的使用文档，如下图1-3所示：

图1-3

在页面的左侧提供一些常见的问题，用户可以根据自己的问题查找相关的文档，在页面中展示出来。同样在“考试星”的页面上我们随处可见的是在线咨询服务，提供了电话咨询和聊天工具的咨询。

## 1.3需求总结

### 1.3.1项目中的利益相关者

在在线考试系统中，使用者主要是学生和老师，网站管理员则负责维护网站的数据库和页面内容等。

### 1.3.2.不同利益相关者的需求

学生：

站在学生的角度上，我需要能够登录网站并进行考试，在考试前可以练习考试的题目，在考试结束后可以查看自己的成绩。如果我在做题时遇到了突发情况，比如电脑突然死机关机或者网络突然不通顺等导致考试无法正常进行，我需要一个机制能够继续考试或者重新考试。

老师：

站在老师的角度上，我需要能够发布一个考试，在发布考试时，我可以选择自己出一套卷子或者由系统为我选题随机组卷。发布考试后我需要知道考试的情况，可以通过视频监测和学生的一些实时操作判断是否有学生作弊。在考试结束后，可以由系统自动批改一些选择题、判断题等答案确定的题目，对于辨析题、简答题等题，我可以在线批改，并且可以给学生添加一些批语。通过该系统我还能查看班级的成绩，查看班级的成绩分析，包括最近几次考试的平均分走势图、成绩分布图等。我还能在网站上上传自己觉得很好的题目，建立自己的题库。在系统中可以发布每次考试的答案解析，并根据一些错题发布一些练习题。

网站管理员：

站在网站管理员的角度上，我需要能够对自己的题库进行更新，对网站防作弊的方法进行升级，通过后台可以实时记录每次考试的情况并保存一段时间。

## 1.4项目中的主要功能

#### 1）组卷发布考试

组卷分为自定义试卷和随机题库组卷，可以由老师自己出试卷，也可以由系统随机从题库中选择题目组卷，若由系统选择，可以选择所有同学做一套试卷，也可以做不同的试卷

#### 2）注册登录

用户需要实名注册，通过账号老师可以添加学生到自己的网络课堂。

#### 3）批改试卷

老师通过系统能够正确批改试卷，发布成绩。对于非主观题，系统能够自动批改。

#### 4）查看成绩

学生可以查看自己的考试成绩，同样老师也可以查看学生的成绩以及班级的成绩。

#### **5）防作弊**

老师通过随机组卷、限制切屏次数以及视频监控来防止考生作弊。

#### 6）找回密码

通过手机信息验证等方法用户可以找回自己的账号，重置自己账号的密码。

## 1.5系统用例图



图1-1

在整个系统中，有三个参与者，分别是学生、老师和管理员。学生可以通过登录注册用例登录系统，通过参加考试用例参加一场考试，在考试过程中，老师通过管理员建立的防作弊系统进行监控、判断学生是否作弊。考试结束后，学生可以查看成绩以及试卷的答案解析。老师可以批改试卷中的开放性题目，对于具有标准答案的题目，系统可以自动批改。每次考试发布前，老师可以手动组卷也可以选择由系统随机组卷。管理员和老师都有建立题库的权限。

## 1.6用例规约

### 1.6.1用例：创建题库

**主要参与者**：系统管理员和老师

**情景目标**：系统管理员和老师可以通过网络在线创建题库。

**前提条件**：系统管理员或老师需要先登录网站并提供正确的账号和密码。

**触发器**：系统管理员或老师决定更新题库的题目或创建新的题库。

**场景**：

1. 系统管理员或老师登录在线考试系统产品网站。
2. 系统管理员或老师输入他的账号。
3. 系统管理员或老师输入两个密码。
4. 系统显示所有功能的按钮。
5. 系统管理员或老师选择创建题库。
6. 系统显示创建题库的界面。
7. 系统管理员或老师填写题库名称、分类并上传试题或在线编写试题，最后提交内容。
8. 系统保存试题并提示创建成功。

**异常处理**：

1. 账号或密码不正确或不被确认（参考用例找回账号）
2. 上传试题无法识别，提示重新上传
3. 题库已经存在，提示是否更新题库或更改题库名称

**后置条件**：

丰富题库，让试卷内容更丰富。

**活动图：**

****

图1-2

如图1-2，开始后，老师需登录网站，通过身份验证，若密码输入有误提示重新输入，多次输入无效后，在一段时间内禁止该ip登录。若正常通过身份验证，则老师选择创建题库功能，填写题库的名称，所属的分类以及相应要上传的试题。提交后，系统需判断识别题目是否有效，有效则保存试题，无效则需重新上传或编写。

**时序图：**

****

图1-3

如上图1-3，老师通过登录进入功能页，身份验证失败则重新登录，进入功能页后，选择创建题库功能，填写相应的题库信息，最后提交表单由系统验证试题是否有效，有效则保存试题，无效则提示重新上传。

### 1.6.2用例：防断网闪退

**主要参与者：**系统管理员

**情景目标：**系统管理员可以通过网站查看哪些考生有断网、闪退现象，并可以查看断网、闪退的原因。断网、闪退时，在限定的时间内将考试时间暂停并保存已经填写的答案。

**前提条件：**系统管理员登录网站并提供正确的账号和密码，考生发生断网或闪退。

**触发器：**考生闪退或断网

**场景**：

1. 考生登录在线考试系统
2. 考生输入他的账号
3. 考生输入他的密码
4. 系统确认账号和密码并作出登录提示
5. 考生参加考试
6. 系统实时监测考生的网络状况和页面打开状况
7. 考生发生断网、闪退
8. 系统暂停考试时间并保存已经填写的答案
9. 考生重新登录网站
10. 系统重启考试时间

**异常处理**：

1. 账号或密码不正确或不被确认（参考用例找回账号）

2. 考生无法再次重连时，保存试卷并记录剩余考试时间。

**后置条件**：

考生正常进行后续考试

**时序图：**

****

图1-5

如上图1-5所示，考生首先登录进入首页，选择考试功能后正常进行考试，此时发生网络等问题

### 1.6.3用例：防作弊

**主要参与者**：系统管理员和老师

**情景目标**：系统管理员或老师可以通过网络视频监控考生，还能通过限制切屏以及每次显示的题目数量来防止作弊

**前提条件**：考生正确登录网站参加考试并开启电脑摄像头

**触发器**：老师想要远程查看考试时的实时情况

**场景**：

1. 系统管理员或老师登录网站
2. 系统验证登录信息
3. 系统管理员或老师选择查看考试现场
4. 系统显示考试监控的缩略图
5. 系统管理员或老师选择某一个缩略图
6. 系统放大该缩略图
7. 系统管理员或老师开启切屏锁，禁止切屏
8. 系统禁止考生切屏
9. 系统管理员或老师设置每次显示的题目
10. 系统显示对应数目的题目
11. 系统管理员或老师记录考试作弊情况并提交
12. 系统保存作弊情况
13. 系统管理员或老师考试后查看作弊情况

**异常处理**：

1. 账号或密码不正确或不被确认（参考用例找回账号）

2. 考生摄像头突然关闭，系统应暂停该考生考试。

**后置条件**：

为考试的作弊处理提供依据，为考试公平公正公开提供保障

**活动图**

****

图1-4

如图1-4，在正确登录系统后，老师可以查看“考场”的监控，并根据意愿放大某个监控视频。如登录信息错误则提示重新登录，当重新登录达到一定次数时，在一定时间内限制该ip登录网站。保护账号安全。

**时序图：**

****

图1-6

如上图所示，监考人通过页面登录打开监考页面，在监考页面上显示每个正在考试的考生的监控视频缩略图，当监考人选择某个视频缩略图后，放大该视频。若监考人在登录时未通过身份验证，需要重新登录。

**系统类图：**

****

图1-7

如图1-7所示，系统中的类包括：

1. 学生类：学号、姓名、年龄、性别
2. 管理员类：管理员姓名、管理员编号
3. 教师类：教师姓名、教师编号
4. 用户类：账户名、密码
5. 试卷类：分数、试卷编号、考试编号
6. 考试类：考试编号、监考老师编号
7. 监考情况类：监考老师编号、考试编号、考试情况
8. 问题类：所属课程编号、问题内容、所属试卷编号
9. 课程类：课程编号、课程名称、所属专业
10. 单选题类：题目、所属课程、问题编号
11. 填空题类：题目、所属课程、问题编号
12. 主观题类：题目、所属课程、问题编号
13. 多选题类：题目、所属课程、问题编号
14. 答案类：所属问题编号、答案（选项）
15. 题库类：题库编号、创建人、问题编号、题库名称

其中对于各种题目来说，题目都有答案与问题。每个问题又可以归类为不同的课程或者放在不同的试卷中，同理，将问题全部放在一起可以组成题库。对于学生、管理员、老师等都属于用户类，他们都有独立的网站账号和密码。而对于他们本身又具有一些自己特有的属性，比如学生的学号姓名，老师的工号姓名，管理员的编号和姓名等。老师每次发布一场考试都要有相关的考试记录包括监考老师，试卷，考试的监考情况。

**题库类图：**

****

图1-8

如上图1-8所示，对于不同的题目都对应着相应的答案，而对于每一个题目也同样有一个问题。再将问题所属的课程归类，同时也可以将不同的问题进行组卷。对于问题又可以组成题库，题库有创建人，以及题库的别称。

## 1.7数据库设计

本小节介绍本系统的数据库设计，包括数据库E-R图，数据库表，关系模式等内容。

### 1.7.1数据库E-R图

首先，由系统类图能够抽象出以下实体：老师、学生、管理员、问题、题库、课程、试卷、考试等。

对于老师来说，有教师编号、教师姓名的属性；

对于学生来说，有学生学号、学生姓名、学生年龄、学生性别的属性；

对于管理员来说，有管理员编号、管理员姓名的属性；

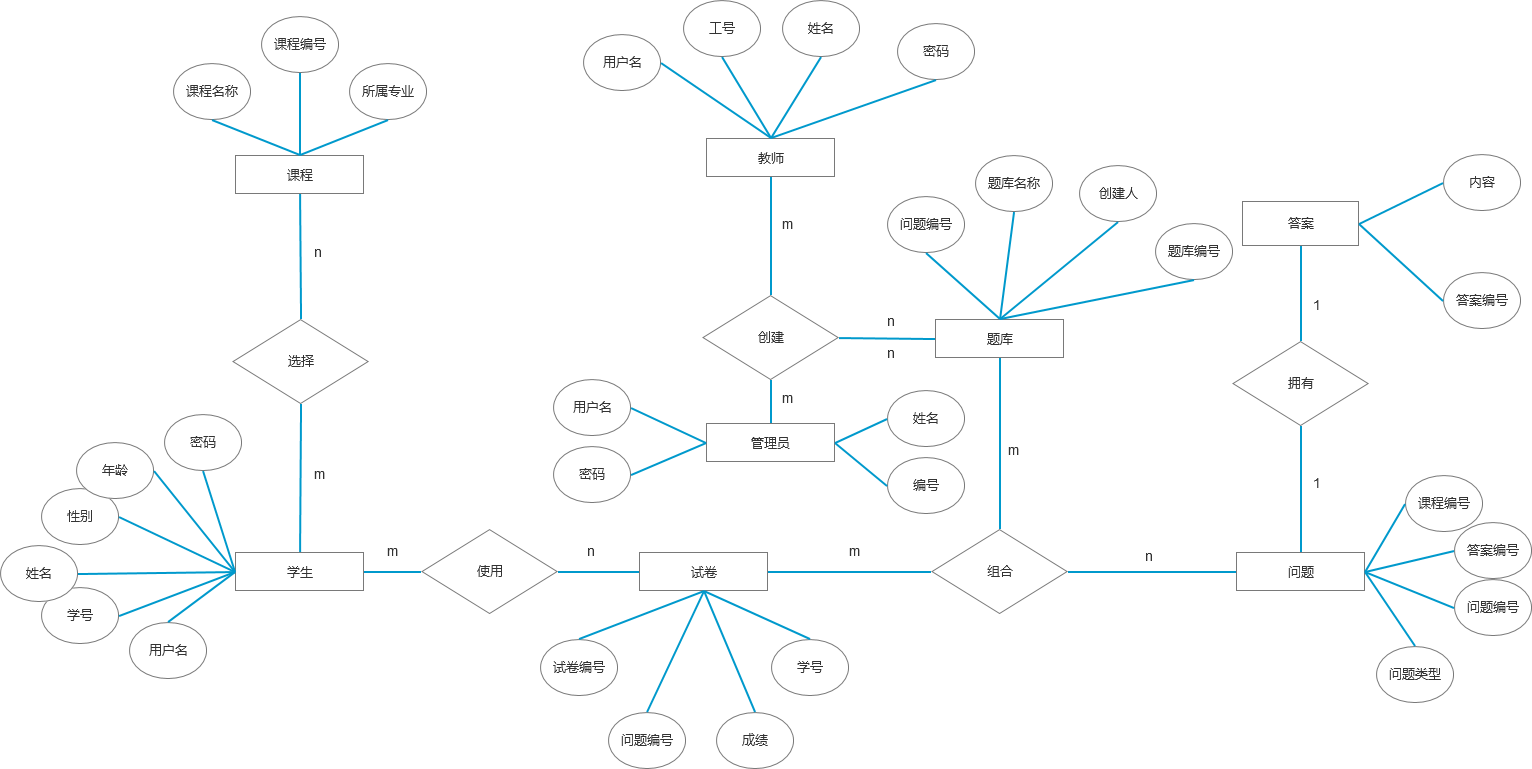
对于问题来说，有问题类型、问题编号、答案编号、所属课程的属性；

对于题库来说，有创建人、题库编号、题库名称、问题编号的属性；

对于课程来说，有课程名称、课程编号、所属专业的属性；

对于试卷来说，有试卷编号、成绩、问题编号、考生学号的属性；

据此数据库E-R图如下图1-9所示：

图1-9

其中，学生可以选择课程，学生与课程之间是多对多的关系；学生还可以参加考试使用试卷，多个学生可以使用不同试卷，不同的试卷也可以由不同的学生使用，因此学生与试卷之间也是多对多的关系；试卷与问题之间也是多对多的关系，试卷可以由不同的问题组合在一起，不同的问题组合也可以产生不同的试卷；管理员、教师都与题库建立多对多的关系，每一位管理员或老师都可以创建不同的题库，而不同的题库也可以由不同的人来创建。对于问题和答案只能是一对一的关系，一个问题只能对应一个标准答案。

## 1.7.2数据库关系模式

学生表（学生学号、学生姓名、学生年龄、学生性别）

教师表（教师编号、教师姓名）

管理员表（管理员编号、管理员姓名）

问题表（问题编号、问题类型、答案编号、所属课程）

题库表（题库编号、创建人、题库名称、问题编号）

课程表（课程编号、所属专业、课程名称）

试卷表（试卷编号、成绩、问题编号、考生学号）

用户表（用户名、密码）

如上给出的关系模式，其中带下划线的属性为主键，带波浪线的属性为外键。

## 1.7.3数据库表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型 | 描述 | 是否主键 |
| student\_id | int | 学生学号 | 主键 |
| student\_name | nvarchar(50) | 学生姓名 |  |
| student\_age | int | 学生年龄 |  |
| student\_sex | char(5) | 学生性别 |  |

表1-1学生表

如上表1-1学生表所示，学号使用整型表示为该表的主键，学生姓名自动识别字符，姓名合法性在页面中使用脚本语言控制。学生年龄使用整型表示，学生性别使用字符表示。

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型 | 描述 | 是否主键 |
| teacher\_id | int | 教师编号 | 主键 |
| teacher\_name | nvarchar(50) | 教师姓名 |  |

表1-2教师表

如上表1-2教师表所示，教师编号使用整型变量表示，置为该表的主键，教师姓名自动识别字符，合法性同样在页面中使用脚本语言控制。

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型 | 描述 | 是否主键 |
| admin\_id | int | 管理员编号 | 主键 |
| admin \_name | nvarchar(50) | 管理员姓名 |  |

表1-3管理员表

如上表1-3管理员表所示，管理员编号使用整型变量表示，置为该表的主键，管理员姓名自动识别字符，合法性同样在页面中使用脚本语言控制。

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型 | 描述 | 是否主键 |
| question\_id | int | 问题编号 | 主键 |
| question\_type | nvarchar(50) | 问题类型 |  |
| answer\_id | int | 答案编号 | 外键 |
| class\_id | int | 课程编号 | 外键 |

表1-4问题表

如上表1-4问题表所示，问题编号question\_id作为该表主键，使用整型作为其数据类型；问题类型为单选、多选、填空、主观等选项，数据由页面控制其合法性。答案编号以及课程编号作为该表的外键，保证参照完整性。

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型 | 描述 | 是否主键 |
| test\_lib\_id | int | 题库编号 |  |
| test\_lib\_creator | nvarchar(50) | 创建人 |  |
| test\_lib\_name | nvarchar(50) | 题库名称 | 主键 |
| question\_id | int | 问题编号 | 外键 |

表1-5题库表

如上表1-5题库表所示，题库表中题库名称作为题库表的主键，使用question\_id作为题库中的问题并将其置为外键，保证每个编号对应相应的问题。

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型 | 描述 | 是否主键 |
| class\_id | int | 课程编号 | 主键 |
| major | nvarchar(50) | 所属专业 |  |
| class\_name | nvarchar(50) | 课程名称 |  |

表1-6课程表

如上表1-6课程表所示，课程表包括课程编号置为主键、所属专业以及课程名称。

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型 | 描述 | 是否主键 |
| test\_id | int | 试卷编号 | 主键 |
| test\_score | nvarchar(50) | 试卷成绩 |  |
| question\_id | int | 问题编号 | 外键 |
| student\_id | int | 考生编号 |  |

表1-7试卷表

试卷表中试卷编号作为该表的主键，question\_id作为该表的外键，确保试卷上的题目都有对应的问题。考生编号为参加考试的学生的编号。

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型 | 描述 | 是否主键 |
| user\_id | nvarchar(50) | 用户名 | 主键 |
| user\_pwd | nvarchar(50) | 密码 |  |

表1-8用户表

用户表保存用户的用户名和密码，包括学生、教师、管理员三者的用户名和密码。

## 1.8用户界面

主页面布局如下图所示：

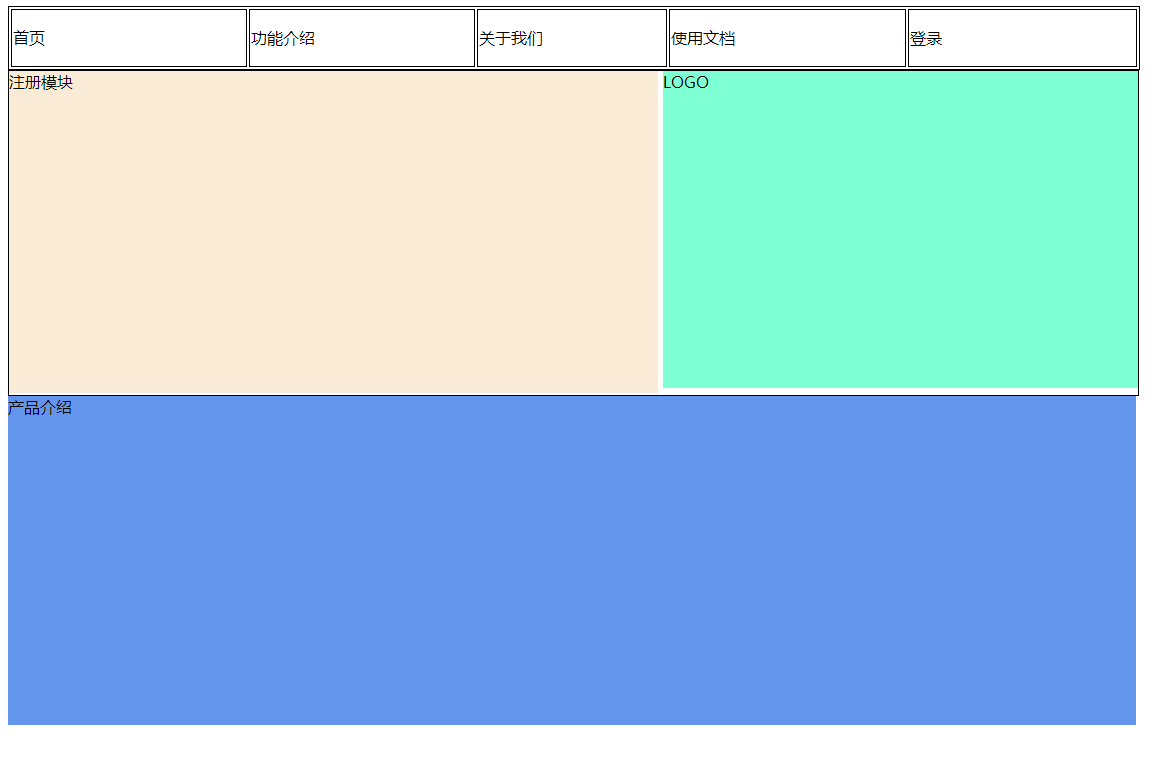


图1-10

如上图所示，最上方是导航栏，包括首页、功能介绍、关于我们、产品的使用文档以及登录入口。

黄色部分为注册模块，用户可以在首页方便快捷的进行注册。

绿色部分放置产品的LOGO图片，给每一个使用软件的人加深印象。美化界面。

下方的蓝色部分，主要简单介绍产品的功能以及产品的优势，告知用户使用该产品能为用户带来哪些便利。

如下图1-11为创建题库的页面大致布局：

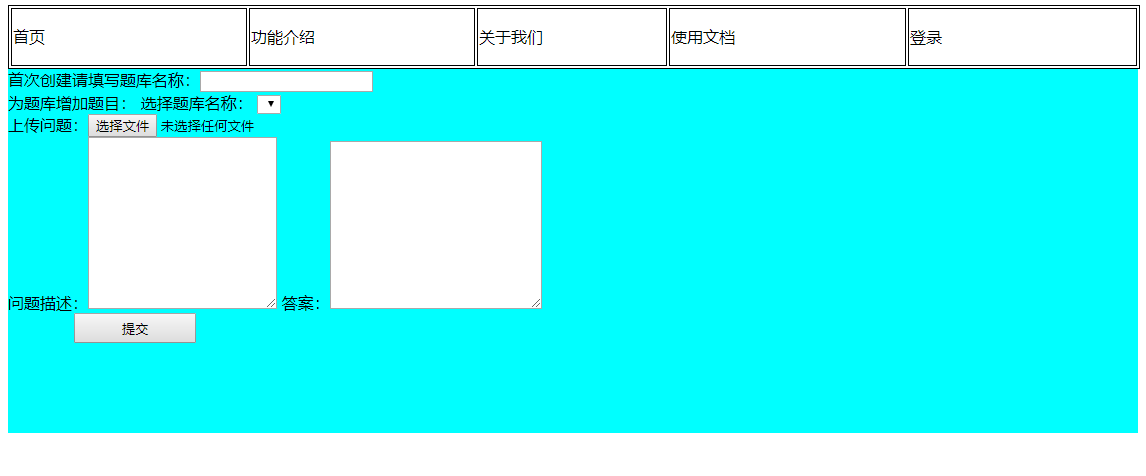


图1-11

在创建题库时，若首次创建题库，则需要先创建新的题库名称。若想完善以前的题库，需要拍照上传题目或手动添加问题描述和答案。单击提交按钮提交并验证题库名称是否重复以及填写的内容是否正确识别。

# 第二章 测试

## 2.1测试用例

如下表2-1所示，为测试题库用例：

|  |  |  |
| --- | --- | --- |
| 测试编号 | 测试对象 | 测试数据 |
| 01 | 创建题库用例 | 正确数据：  第一组数据：测试题库（题库名称），张三（创建人），100（问题编号）；  第二组数据：测试题库、张三、99  第三组数据：测试题库、张三、98  第四组数据：测试题库2、李四、50  …… |
| 错误数据：  第一组数据：测试题库，张三，100  第二组数据：测试题库，李四，100  …… |
| 预期结果： | 正确结果：正确题库名称：“测试题库”为一个题库，而“测试题库2”为另外一个名称。创建人名称可以不同，可以相同。  备注：问题编号由用户选择时确认，由数据库约束其实体完整性 | |
| 错误结果：题库重名错误，同一个题库不能由不同的人创建。 | |

表2-1

解决方案：若出现错误结果，即题库名重复错误，则提示用户题库名重复，需要用户重新填写题库名称。

如下表2-2所示，为测试防作弊用例：

|  |  |  |
| --- | --- | --- |
| 测试编号 | 测试对象 | 测试数据 |
| 02 | 防作弊用例 | 第一组数据：切屏限制为3次，测试切屏10次，快速完成；  第二组数据：切屏限制为5次，测试切屏10次，快速完成；  第三组数据：切屏限制为7次，测试切屏15次，快速完成；  …… |
| 预期结果： | 正确结果：切屏三次（或多次）后，锁定切屏。三次切屏后，切屏无效。 | |
| 错误结果：切屏三次（或多次）后，仍然切屏成功，限制失效。 | |

表2-2

解决方案：每次切屏更改题目的顺序，每次只显示一个题目。

如下表2-3所示，为测试防断网、闪退用例：

|  |  |  |
| --- | --- | --- |
| 测试编号 | 测试对象 | 测试数据 |
| 03 | 防断网、闪退用例 | 第一组数据：正常进行考试时，关闭电脑  第二组数据：正常进行考试时，关闭网络  …… |
| 预期结果： | 正确结果：关闭电脑或关闭网络后，重新启动系统，系统自动还原考试剩余时间，考试题目以及已经完成的题目，还原考试现场。 | |
| 错误结果：考试实时现场丢失。 | |

表2-3

解决方案：在考试时每隔几分钟进行一次现场保存，在触发回档现场时，按时间顺序进行回档，优先尝试距离当前时刻最近的一次现场保存。

# 第三章 总结

学习软件工程这门课已经一个学期了，课程也步入尾声。通过这一个学期的学习，让我对软件的开发过程有了更清晰的认识。从需求的获取到设计分析阶段，再到编码实现阶段以及后期的测试维护阶段。每一个阶段需要的工作我都有了一个大致的轮廓，在细节方面我还需要继续学习。

软件工程就像一个框架一样。指导我们如何正确、高效地开发软件，以及在遇到突发情况时如何及时处理避免造成更大损失。在实际应用时，软件工程中的工具、过程以及方法使得我们能够高效、高质量地构建复杂系统，让杂乱无章的工作变得井然有序。

身为一个学习软件的大学生，最苦恼的莫过于遇到问题无从下手。软件工程这门课告诉我要设计制作一个软件需要从需求开始下手，从基本需求中建立起对这个软件的基本理解，通过uml模型来直观展示出这个软件。在学习这门课程之前，我们做课题时总是东拼西凑式的去做，在编码前并没有通过详细的建模语言以及需求分析来表现出软件的特点。有了题目后就去编码，这种坏习惯导致在编码中遇到各种各样的问题。有些是页面设计问题，有些是业务的逻辑问题，还有些是数据库的问题。总之各种问题层出不穷，此时再去更改设计，再去商讨解决方案已经为时已晚，很多时候都要推翻已经做得工作重新来，这不仅浪费时间而且做出来的软件也很粗糙，会出现各种bug。软件工程让我学会了分析软件的系统，然后基于分析出来的内容去设计系统。在分析的过程中，发现各种各样可以预知的问题，在设计前就将这些问题都解决，在测试时再去发现那些无法预知的问题。进而根据问题拿出解决方案。从需求中获得用例，由用例得到用例规约，建立活动图，时序图最后得到类图。再将各个用例的类图组合在一起，形成系统类图，进而将类图反映在实际的数据库设计中，整个过程紧紧相扣，缺一不可。如果说学习编程语言是教会我如何去当一个合格的程序员，那么学习软件工程就是在教我如何更进一步成为一个合格的管理人员，以放眼整个项目的眼光去看待项目，贯穿整个项目的生命周期。