|  |  |
| --- | --- |
| **P-No.** | EX2019.001 |

|  |
| --- |
| **在线考试系统**  **结构设计书** |

第1.0版

|  |  |  |
| --- | --- | --- |
| 作成 |  |  |
| 翻訳 |  |  |
| 承認 |  |  |

**南京富士通南大軟件技術有限公司技術開発部**

All Rights Reserved, Copyright ©FNST 2005

|  |  |  |  |
| --- | --- | --- | --- |
| **修正履歴** | | **ページ** | 1 |
| **仕様書名** | 在线考试系统结构设计书 | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **版数** | **修正日** | **修正概要** | **修正者** | **承認者** |
| 0.1 | 2019-07-16 | 1. 新规作成 |  |  |
| 0.2 | 2019-07-17 | 1. [1.1]背景，[4.3.4]分析文件，[4.4]外部接口设计的内容追加 2. [图2-1]关系描述修正 3. [2.2.1]，[3.2]流程图修正 4. [3.3] lpstrFlolder类型描述修正 5. [3.4] 函数filemain描述错误修正 6. [4.3.1]删除“使用opsrch函数时，只要将该表的首地址Optab以及表中元素个数作为第二第三参数传入opsrch函数即可。” |  |  |
| 1.0 | 2019-07-19 | 1. [图3-3] 函数调用修正 2. 板式调整 |  |  |

**目 录**

[1. 功能概要 4](#_Toc14672540)

[**1.1 功能概述** 4](#_Toc14672541)

[**1.2 功能点** 4](#_Toc14672542)

[**1.3 基本模型** 4](#_Toc14672543)

[2. 系统架构图 6](#_Toc14672544)

[**2.1** **项目代码结构** 6](#_Toc14672545)

[**2.2** **Springboot搭建SSM框架** 6](#_Toc14672546)

[**2.3** **SSM框架** 6](#_Toc14672547)

[**2.4** **数据库** 7](#_Toc14672548)

[3. 系统详细功能构成 7](#_Toc14672549)

[**3.1** **界面的设计** 7](#_Toc14672550)

[**3.2 事件触发流程图** 10](#_Toc14672551)

[**3.2.1登录界面流程图** 11](#_Toc14672552)

[**3.2.2 笔试界面一览流程图** 12](#_Toc14672553)

# **功能概要**

## **1.1 功能概述**

在公司的招新活动中，传统的现场笔试效率低下，筛选力一般，线上笔试能够克服传统方法的不足。本期项目将开发一个在线笔试系统，用于公司的招新工作。面试官在系统中录入笔试题，笔试者进入系统时将会看到自己笔试题的URL链接，点击后进入笔试，面试官能够实时的观看笔试者的答题过程，答题完毕后也可回放该过程。

## **1.2 功能点**

基于以上的基本功能描述，可以得到如下的功能点一览表：

|  |  |  |
| --- | --- | --- |
| 序号 | 功能点描述 | 备注 |
| 1 | 系统登入身份判断 |  |
| 2 | 笔试题信息的录入 |  |
| 3 | 考生状态的判断 |  |
| 4 | 考生获取URL链接进入笔试 |  |
| 5 | 笔试过程的实时观看与回放 |  |

表1-1 功能点一览表

## **1.3 基本模型**

依据以上四个功能点，可以将整个软件分为以下模块：

1. 界面

界面分为面试官界面、考生界面和登入界面。面试官界面可即时查看各登入考生的状态，给等待笔试的考生提供考题的URL链接，实时观看正在进行笔试的考生答题过程，查看笔试结束的考生的答题回放；考生界面在没有收到URL链接时即为等待笔试，收到链接后可进入笔试界面，开始答题后开始计时。登入界面根据登入信息决定登入的是考生界面还是面试官界面。

1. 笔试题信息的录入

用于新建笔试题，在面试官界面操作，录入的题目将存入数据库

1. 考生状态判断

考生状态有三种，等待笔试、笔试进行、笔试结束，即时显示在面试官界面

1. 考生获取URL链接进入笔试

考生收到来自面试官提供的URL链接，通过链接进入笔试界面

1. 笔试过程的实时观看与回放

笔试过程中，面试官可进行实时观看，笔试结束后，面试官可以回看考生的笔试过程

各模块之间之间的关系如下图所示：

登入界面

考生界面

考生状态判断

获取URL进入笔试

笔试过程

实时观看与回放

面试官界面

试题录入

图 1-1 在线笔试系统的基本模型

对于开发者而言，系统则可以分为前端界面和后端控制系统。

# **系统架构图**

## **项目代码结构**

## **Springboot搭建SSM框架**

Spring Boot是由Pivotal团队提供的全新框架，其设计目的是用来简化新Spring应用的初始搭建以及开发过程。该框架使用了特定的方式来进行配置，从而使开发人员不再需要定义样板化的配置。通过这种方式，SpringBoot致力于在蓬勃发展的快速应用开发领域(rapid application development)成为领导者。

从最根本上来讲，Spring Boot就是一些库的集合，它能够被任意项目的构建系统所使用。

进行打包和分发的工程会依赖于像Maven或Gradle这样的构建系统，为了简化依赖图，Boot的功能采用模块化的，通过导入Boot所谓的“starter”模块，可以将许多的依赖添加到工程之中。为了更容易地管理依赖版本和使用默认配置，框架提供了一个parent POM，工程可以继承它。

## **SSM框架**

SSM（Spring+SpringMVC+MyBatis）框架集由Spring、MyBatis两个开源框架整合而成（SpringMVC是Spring中的部分内容）。常作为数据源较简单的web项目的框架。

**Spring**

Spring就像是整个项目中装配bean的大工厂，在配置文件中可以指定使用特定的参数去调用实体类的构造方法来实例化对象。也可以称之为项目中的粘合剂。

Spring的核心思想是IoC（控制反转），即不再需要程序员去显式地`new`一个对象，而是让Spring框架帮你来完成这一切。

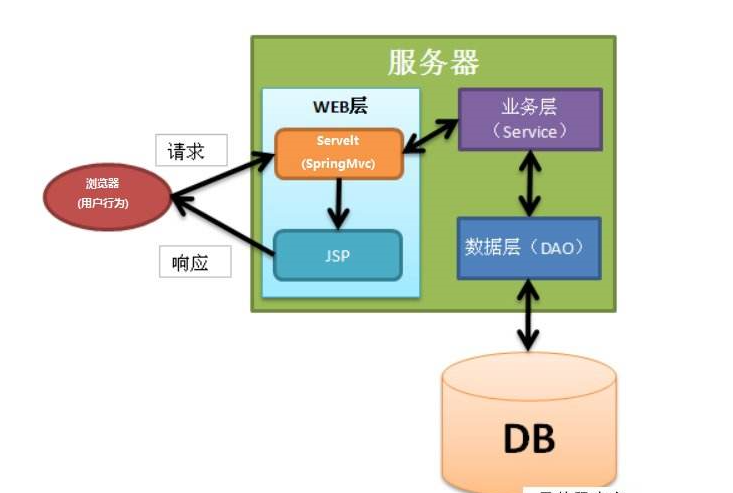
**SpringMVC**

SpringMVC在项目中拦截用户请求，它的核心Servlet即DispatcherServlet承担中介或是前台这样的职责，将用户请求通过HandlerMapping去匹配Controller，Controller就是具体对应请求所执行的操作。SpringMVC相当于SSH框架中struts。

**mybatis**

mybatis是对jdbc的封装，它让数据库底层操作变的透明。mybatis的操作都是围绕一个sqlSessionFactory实例展开的。mybatis通过配置文件关联到各实体类的Mapper文件，Mapper文件中配置了每个类对数据库所需进行的sql语句映射。在每次与数据库交互时，通过sqlSessionFactory拿到一个sqlSession，再执行sql命令。

页面发送请求给控制器，控制器调用业务层处理逻辑，逻辑层向持久层发送请求，持久层与数据库交互，后将结果返回给业务层，业务层将处理逻辑发送给控制器，控制器再调用视图展现数据。项目架构图如图2-1.

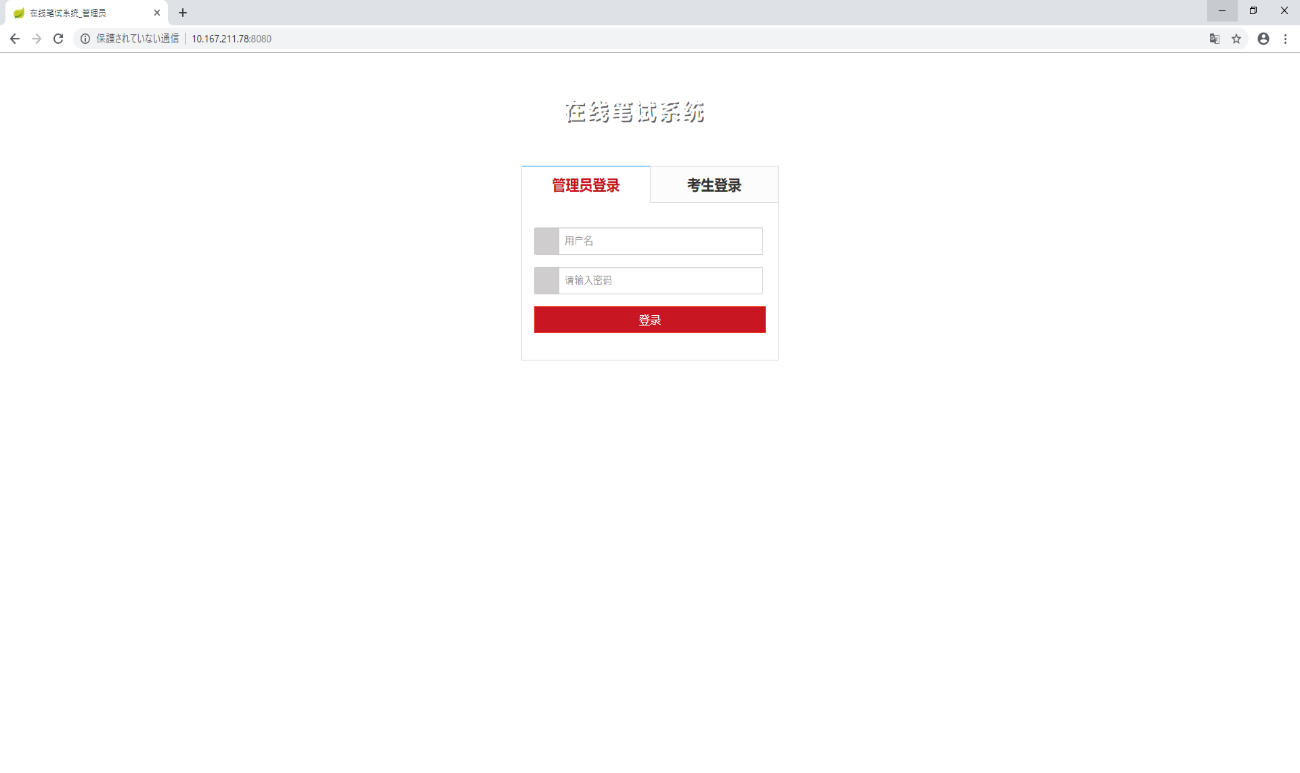
图 2-1 SSM架构图

## **数据库**

数据库采用mysql进行搭建。建表如图2-2.

# **系统详细功能构成**

## **界面的设计**

图 3-1 在线笔试系统登录界面

用户选择不同的权限进入网站，管理者进入后台管理界面，考生进入考试界面进行考试。

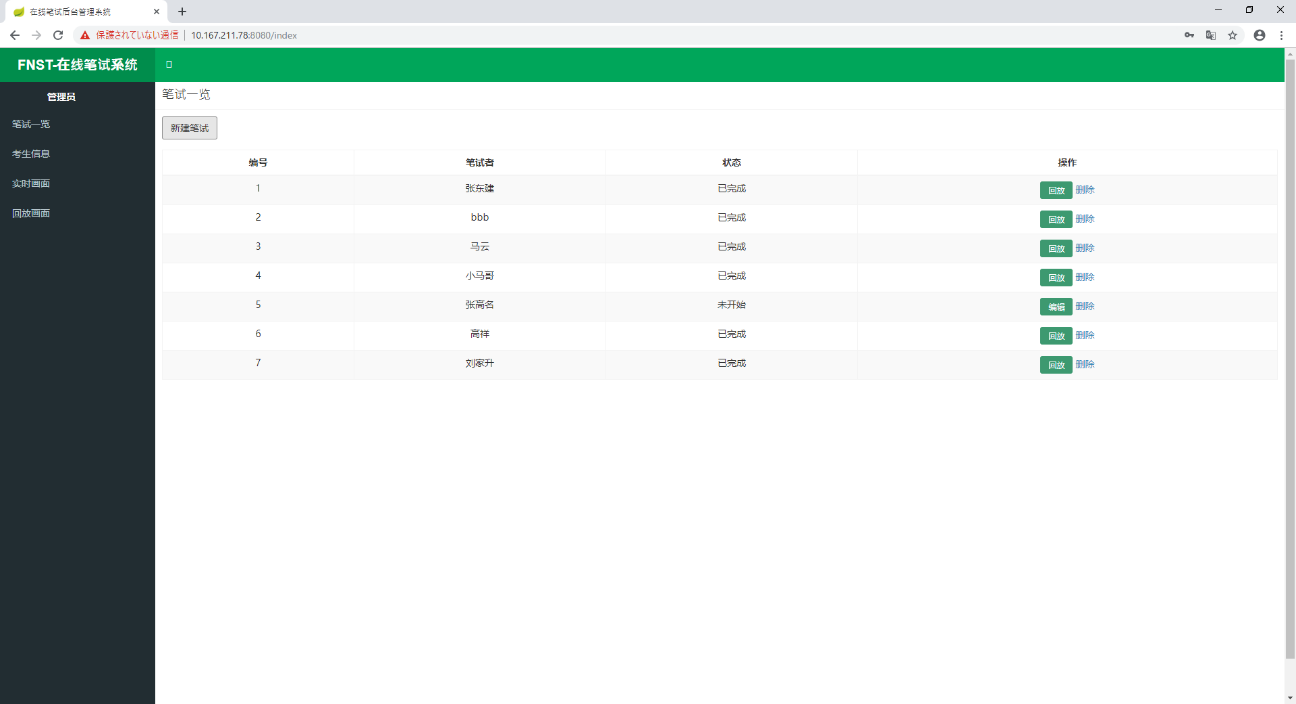


图 3-2 管理员界面-笔试一览模块

管理员在笔试一览界面可以查看考生的考试情况。未开考前，可以点击“编辑”修改考生题目；考试时，可以“直播”实时查看考生考试的电脑界面；考试结束后，还可以点击“回放”查看已经结束的考试信息。

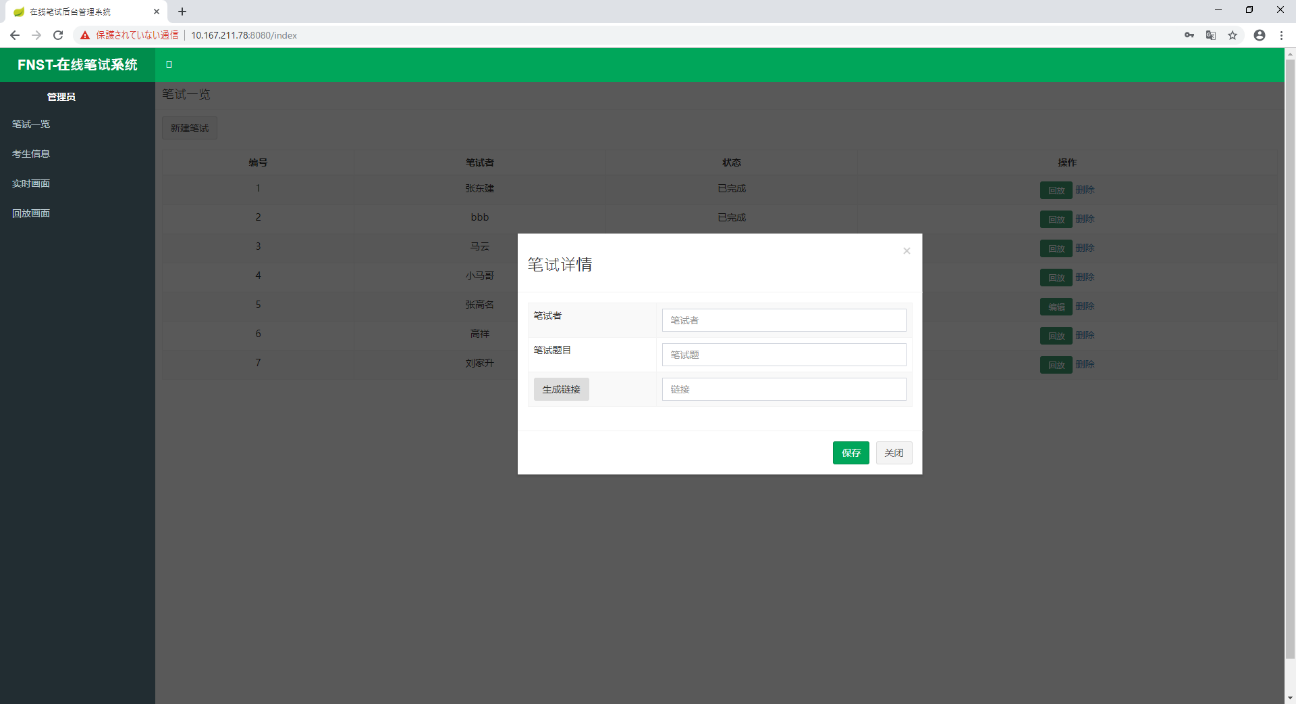


图 3-3 新建笔试界面

管理员可以点击“新建笔试”按钮来将考生与试题进行绑定，即建立考试，生成url考试链接。

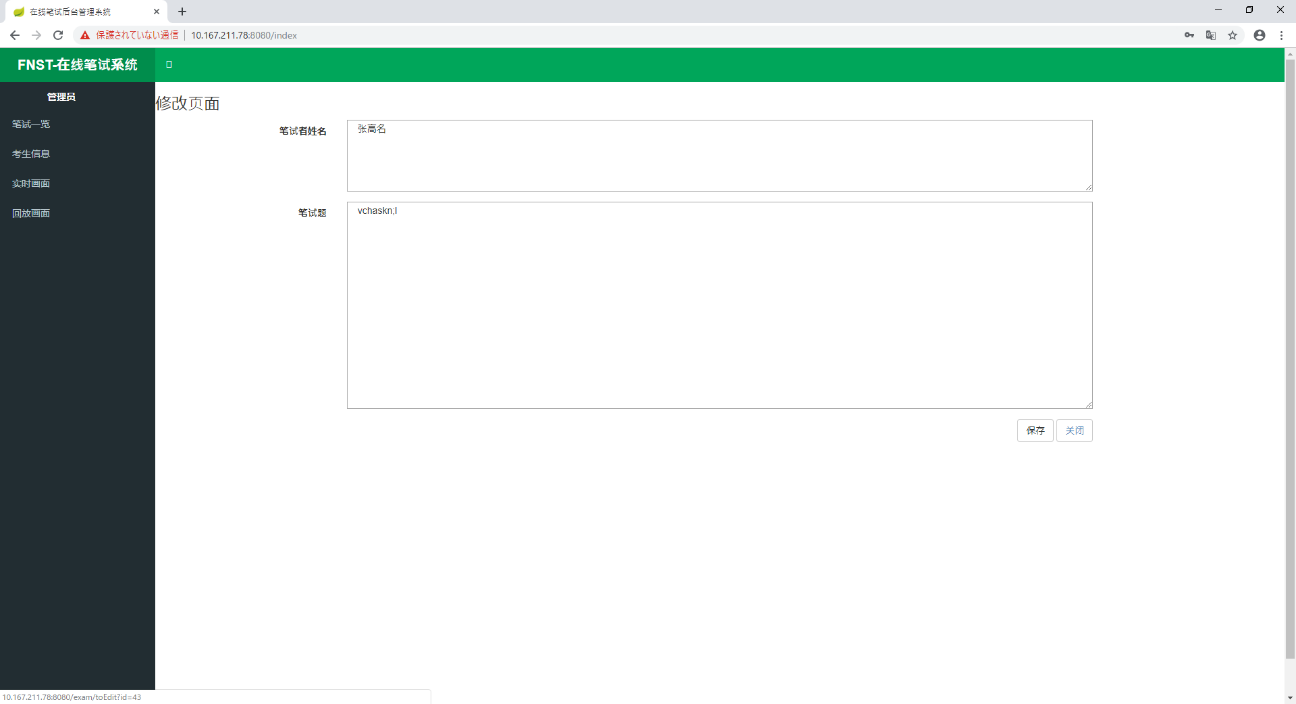


图 3-4 修改笔试界面

在考试未开始前，管理员可以点击“编辑”按钮，对已考试试题进行修改。

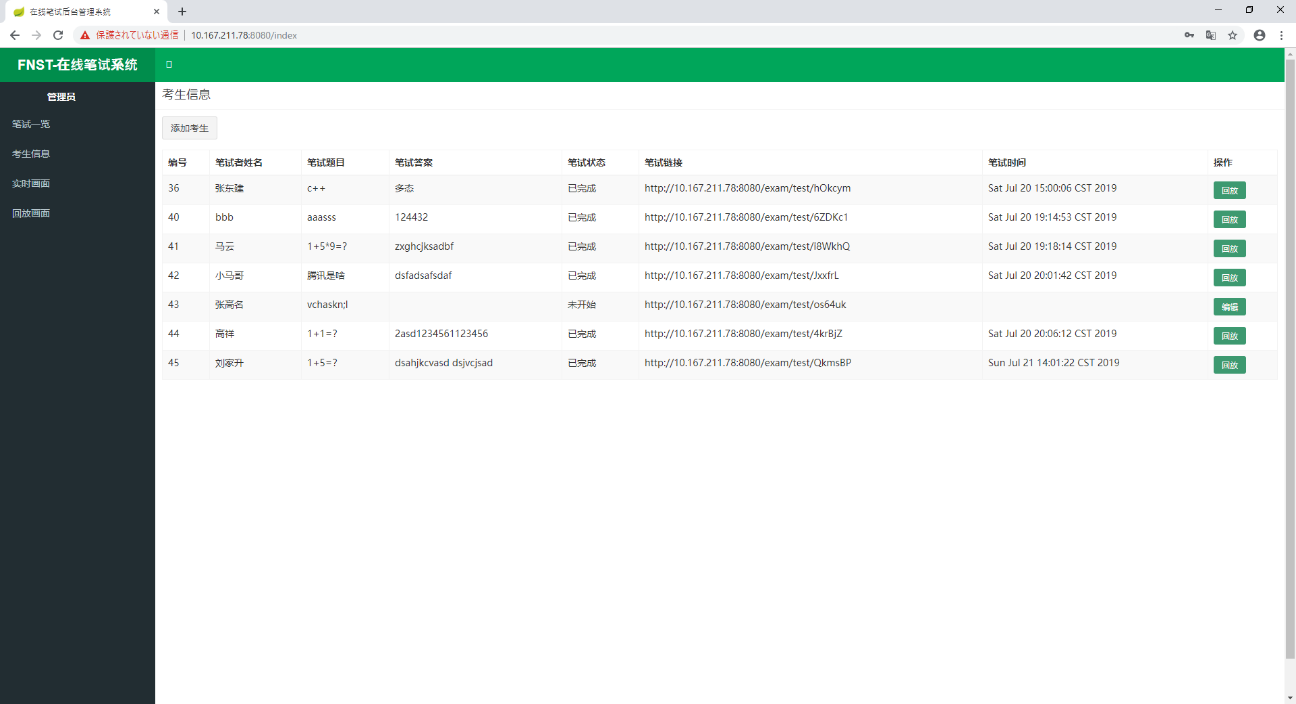


图 3-5考生信息界面

管理员也可以在考生信息界面查看考生的详细信息，包括：考生姓名，试题，答案，考试状态，url链接，考试完成时间以及查看考试视频。

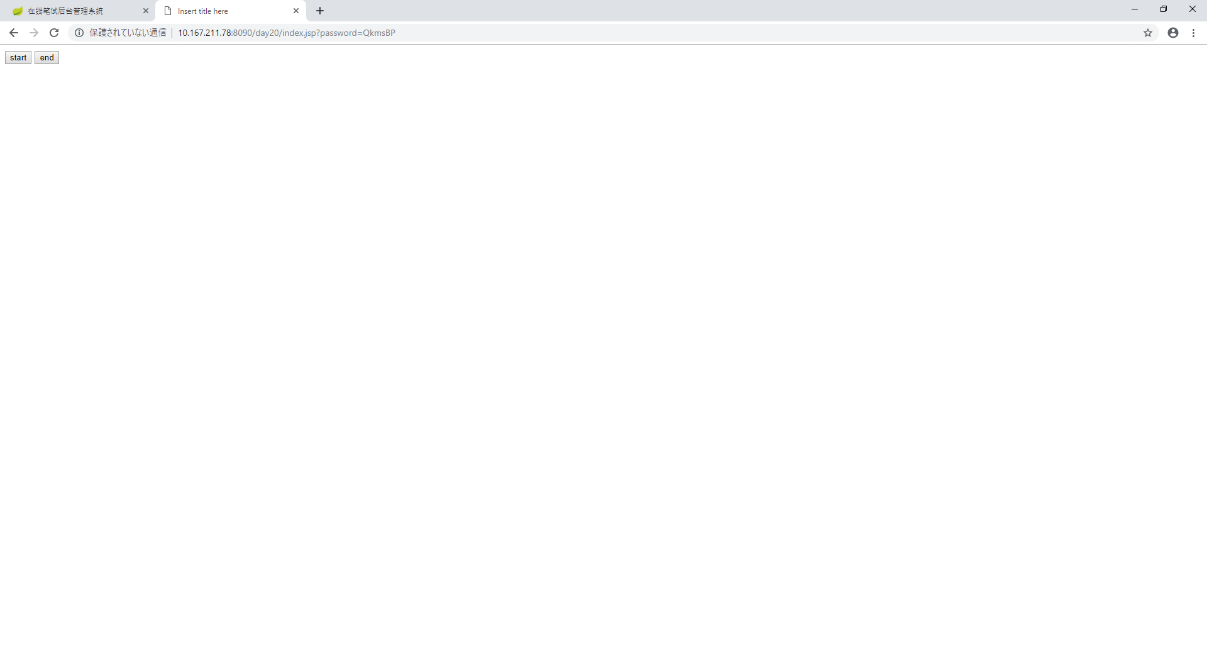


图 3-6考生开始界面

考生点击“start”按钮，进入答题界面，答题结束后，回到此界面，点击“end”结束考试和录屏。

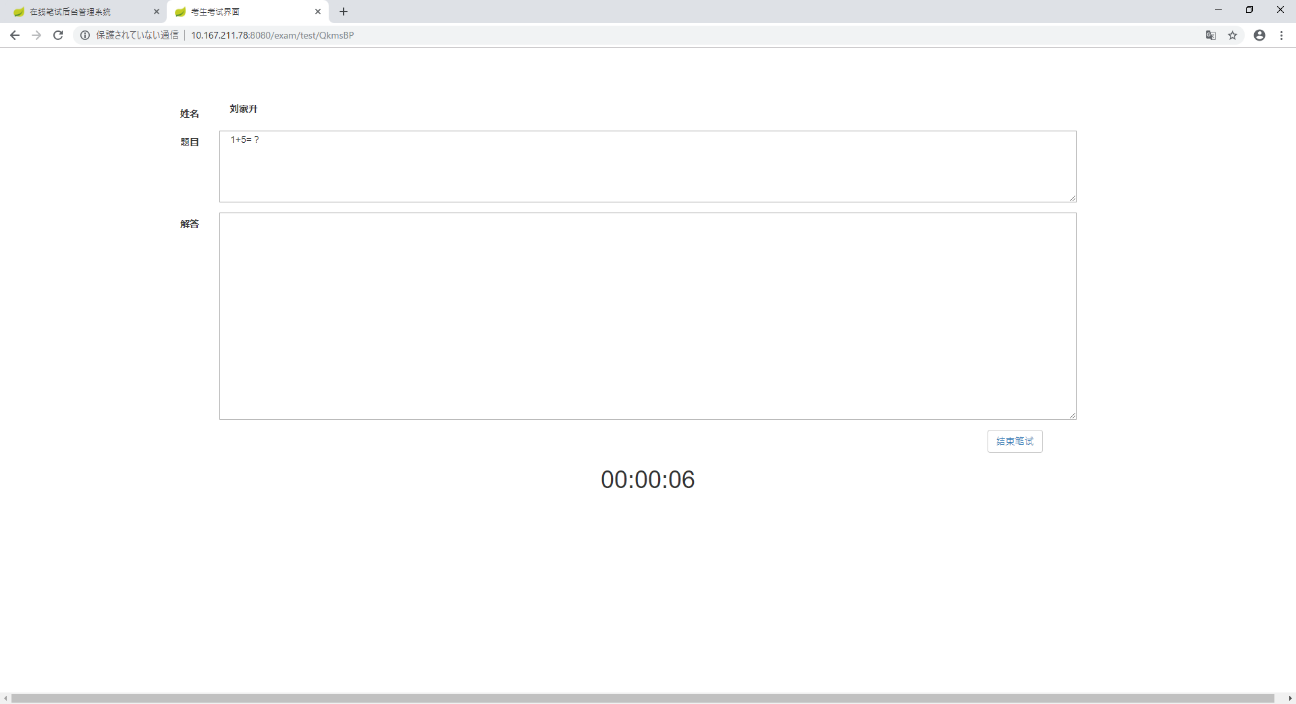


图 3-7考生考试界面

考生进入考试界面，开始考试，页面下方有正计时器，同时录屏开始。考试结束后，点击“结束笔试” 按钮，回到图3-6界面，点击“end”按钮，结束录屏和考试，退出系统。

## **3.2 事件触发流程图**

本模块由事件驱动，由用户点击按钮后触发各种事件的发生，以下将详细介绍各个按钮的功能以及基本流程。

### **3.2.1登录界面流程图**

-

图3-8 登录流程图

在这个事件触发过程中，功能的实现主要在“点击登录”一部分，在该部分中，使用click(function (){})事件处理函数，将输入框得到的账户与密码与数据库保存的考生信息进行比对，若存在，即可进入考试界面，否则无法登录。

|  |
| --- |
| $(document).ready(function () {  $("#login").click(function () {  var stuName = $("#username").val();  var password = $("#password").val();  $.ajax({  url: "/exam/loginStu",  data: {stuName:stuName,password:password},  success: function (data) {  alert(data)  if(data=="登陆成功"){  window.location.href="http://10.167.211.78:8090/day20/index.jsp?"+"password="+password;  }else{  }  }  });  });  }); |

表3-1 登录验证代码

### **3.2.2 笔试界面一览流程图**

“笔试一览”界面是在线考试系统后端的最关键部分，实现的功能包括：新建笔试、修改笔试、查看考生考试进度以及删除考生信息的功能。以下我们主要介绍一下管理员查看考试进度的具体流程图：



图3-9 管理员后台功能流程图

在这一部分最重要的是功能是考试页面的直播，录屏和回放。其原理是利用java的robot类来实现电脑页面的定时截取，并将图片保存到本地，再按照图片的保存顺序按时播放，即实现了考试的录屏和回放，直播则是定位的刚刚截下来的图片开始播放，有一小段的时延。录屏的核心代码为表3-2，回放的核心代码为表3-3，直播则是表3-4。

|  |
| --- |
|  |

表3-2 录屏的核心代码

|  |
| --- |
|  |

表3-3 回放的核心代码

|  |
| --- |
|  |

表3-4 直播的核心代码