|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **学校代号** | **10162** |  | **学 号** | **201730401002** |
|  | | | | |
| **学校图标辽宁中医药大学**   |  | | --- | | Liaoning University of Traditional Chinese Medicine |   **学士学位论文**   |  |  |  |  | | --- | --- | --- | --- | | **高校在线心理测试系统的设计与实现** | | | | | **学位申请人** | **梁兵月** | | **指 导 教 师** | **吴磊** | | **专 业 名 称** | **信息管理与信息系统** | | **申请学位类型** | **管理学** | | **论文提交日期** | **2020.05** | | | | | |

**辽宁中医药大学学位论文独创性与真实性声明**

本人郑重声明：在我呈交的学位论文中，本人经研究所得的数据、原理、结论等一切内容均真实，且经本人认真核对无误；我呈交的与学位论文相关的资料（软件、原始记录、照片、录像片、检查化验报告单等）确为完整的学位论文研究中的原始资料，本人没作任何修改；我呈交的学位论文是我本人在指导教师的指导下进行的研究工作及取得的研究成果。除文中特别加以标注和致谢的地方外，论文中没有其他人已经发表或撰写过的研究成果，其他人对我完成此论文的帮助以及对本研究所做的任何贡献均已在论文中作了明确说明并表示谢意。

我呈交的学位论文及与该论文相关的资料若有不实之处，本人承担一切相关责任。

学位论文作者签名＿＿＿＿＿

日 期： 年 月 日

**关于学位论文使用授权的声明**

本人完全了解辽宁中医药大学有关保留使用学位论文的规定，同意学校保留或向国家有关部门机构送交论文的印刷版和电子版，允许被查阅和借阅。本人授权辽宁中医药大学可以将本学位论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或其他复印手段保存和汇编本学位论文。

（保密论文在解密后应遵守此规定）

学位论文作者签名＿＿＿＿＿ 学位论文导师签名＿＿＿＿＿

日 期： 年 月

**目 录**

[1 绪论 - 10 -](#_Toc29431)

[1.1 选题背景 - 10 -](#_Toc15209)

[1.2 关键技术介绍 - 10 -](#_Toc15149)

[1.2.1 Spring Boot - 10 -](#_Toc11650)

[1.2.2 MyBatis - 11 -](#_Toc19154)

[1.2.3 Redis - 11 -](#_Toc31334)

[1.2.4 Spring Security - 11 -](#_Toc26572)

[1.2.5 Vue JS - 11 -](#_Toc13094)

[1.2.6 Element UI - 12 -](#_Toc16035)

[2 系统分析 - 12 -](#_Toc6909)

[2.1 需求分析 - 12 -](#_Toc29490)

[2.1.1 功能需求分析 - 12 -](#_Toc9654)

[2.1.2 非功能需求分析 - 13 -](#_Toc5103)

[2.2 可行性分析 - 14 -](#_Toc21096)

[2.2.1 系统管理员 - 14 -](#_Toc27713)

[2.2.2 技术可行性 - 14 -](#_Toc3294)

[2.2.3 经济可行性 - 14 -](#_Toc22207)

[2.2.4 社会可行性 - 15 -](#_Toc21345)

[2.3 业务流程分析 - 15 -](#_Toc30298)

[2.3.1 普通用户 - 15 -](#_Toc3013)

[3 概要设计 - 16 -](#_Toc13743)

[3.1 系统功能设计 - 16 -](#_Toc2532)

[3.1.1 登录注册模块 - 16 -](#_Toc9752)

[3.1.2 心理测试模块 - 16 -](#_Toc6895)

[3.1.3 用户信息模块 - 17 -](#_Toc25666)

[3.2 系统架构设计 - 18 -](#_Toc17463)

[3.2.1 系统类图 - 18 -](#_Toc5695)

[3.2.2 数据库设计 - 21 -](#_Toc6498)

[4 详细设计与实现 - 27 -](#_Toc30584)

[4.1 项目目录结构设计 - 27 -](#_Toc27944)

[4.2 登录设计 - 28 -](#_Toc18671)

[4.2.1 前端设计 - 28 -](#_Toc30983)

[4.2.2 后端设计 - 29 -](#_Toc28835)

[4.3 注册设计 - 33 -](#_Toc15254)

[4.3.1 前端设计 - 33 -](#_Toc10526)

[4.3.2 后端设计 - 33 -](#_Toc2644)

[4.4 题库设计 - 35 -](#_Toc27254)

[4.4.1 前端设计 - 35 -](#_Toc31538)

[4.4.2 后端设计 - 36 -](#_Toc16453)

[4.5 测试设计 - 38 -](#_Toc11048)

[4.5.1 前端设计 - 38 -](#_Toc7263)

[4.5.2 后端设计 - 39 -](#_Toc21127)

[4.6 个人中心设计 - 40 -](#_Toc29784)

[4.6.1 前端设计 - 40 -](#_Toc23798)

[4.6.2 后端设计 - 41 -](#_Toc10263)

[5 系统测试 - 43 -](#_Toc10808)

[5.1 登录测试 - 43 -](#_Toc25084)

[5.2 答题测试页测试 - 44 -](#_Toc1574)

[5.3 量表管理页测试 - 46 -](#_Toc25975)

[5.4 个人中心页测试 - 48 -](#_Toc6537)

[总 结 - 50 -](#_Toc9973)

[参考文献 - 51 -](#_Toc25272)

[致 谢 - 52 -](#_Toc8333)

**摘 要**

随着社会的发展，大学生择业与就业问题尤其严重，环境适应问题、学习问题、人际关系问题、恋爱问题等方面的压力也越来越大，高校学生心理问题越来越多，大学生因心理障碍自杀事件频发，所以我们应该时刻关注学生的心理健康问题，为此，我开发了一款心理测试Java Web系统，使用人群包括学生和老师，通过查询心理相关资料找寻适合当代学生三观和心理的测试题，根据不同类型、维度，对学生心理进行测试评价，判断该学生的心理状态，预防心理不健康，甚至更危险的心理疾病或行为的发生。本系统是使用IntelliJ IDEA集成开发IDE，基于Java8，使用了Spring 生态下脚手架Spring Boot2.4、Spring Security等作为基础架构进行开发，数据库使用的是MySQL。项目部署在腾讯云服务器，解析域名可以通过域名进行访问。本文主要是论述项目架构、实现过程和业务设计，并阐述了业务和技术的选型利弊。

**关键词：**心理健康；Java；Spring；MySQL；IntelliJ IDEA

**Abstract**

With the development of society, college students’ job selection and employment problems are particularly serious. The pressure on environmental adaptation problems, learning problems, interpersonal problems, love problems, etc. is also increasing. College students have more and more psychological problems, and college students are suffering from psychological barriers. Suicides occur frequently. Therefore, mental health education for college students has become a social issue that cannot be ignored. Therefore, we should always pay attention to the mental health of students. For this reason, I developed a psychological test Java Web system. The users include students and teachers. By querying psychological information, we can find test questions suitable for contemporary students’ three views and psychology. It is not necessary to test and evaluate the student's psychology by type and dimension, to judge what kind of situation the student's psychology is in, and try to avoid the occurrence of unhealthy or even more dangerous psychological diseases or behaviors. The system is developed using IntelliJ IDEA integrated development IDE, based on Java8, using the scaffolding Spring Boot 2.4, Spring Security, etc. under the Spring ecology as the infrastructure for development, and the database uses MySQL. The project is deployed on the Tencent Cloud server, and the domain name can be resolved by using the domain name for access. This article mainly discusses the process of project architecture and realization, and business design, and expounds the pros and cons of business and technology selection.

**Keywords：**Mental health;Java;Spring;MySQL;IntelliJ IDEA

**前 言**

在校大学生正处在成长转型、适应新环境、心理塑性以及三观建立的关键时期，近年来事关大学生的几起案件，大多数是由于学生心理障碍才导致了悲剧的发生。社会、学校、家庭三方对此应该重视起来。就业问题、学业压力、人际交往等情况，是造成当今大学生心理障碍的主要原因，但由于现阶段，相当多的学生虽然存在着心理困惑，但不愿与父母朋友交流沟通，大多数学生很不乐意在心理状况上倾注精力和时间，不愿走进心理咨询室，甚至认为心理咨询是羞耻的，在现实生活中将自己的内心隐藏起来。因此通过查阅相关心理方面的资料，确定人际，学习，就业，情感四个基础方面的测试量表为主要测试内容；根据不同测试类型的量表来检测学生在哪一方面存在心理困惑。大学生可以通过本系统进行自我测评，系统给出测试结果，帮助学生更快的了解自己的心理状态，提供参考依据，对自己的心理问题提前采取有效的干预措施；另外学校也可通过本系统定期监测学生的心理状况，以便及时发现学生心理障碍问题，针对问题可以提前进行适度的干预，为学生提供切实帮助，精准有效地为学生解疑答惑，尽到学校职责；且本系统可随时查阅学生的测试记录，方便学校收集数据分析，减轻繁琐的手工劳动，提高监测效率，跟踪分析学生的心理变化情况，从根本上预防危机事件的发生。本设计的主要内容有有设计用户操作界面，实现用户注册登录，可以通过 PC 机和移动终端相结合的方式访问本系统，成功给出测试结果。本系统的内容主要包括心理测试模块和衍生的题库管理、量表管理、选项管理、评判管理和维度管理，以及系统最基本的用户角色管理模块。各模块实现基础数据的增删改查，本系统的核心功能心理测试模块不仅实现答题卡的题目切换，还实现了答题时间的记录，使测试结果更加真实客观，保证结果分析的精准有效性。

**文献综述**

心理评估是一套标准化的评估技术。指对各种心理和行为问题的评估。心理测试是心理评估最为重要的技术，它借助标准化的测量工具，将人的心理现象或行为进行量化。心理测试量表是一种心理测试工具，编制时要确保量表的有效性与时效性。当今的测试技术越来越多样化，测试量表的出现使心理测试工作达到全新水平。

国外研究现状：南非国内西北大学的 Nicolene Barkhuizen 等人在研究中用它来测量人格。西开普大学的 Verona Rosemarie Solomon 研究了15FQ+在工作绩效中的预测性。新西兰国内梅西大学的 Chloe M.Hammans 用它来测量人格特质[基于B/S]。近年来国外高校比较看重学生的心理健康，针对学生身心发展规律开通不同的心理教育课程。目前，国外高校充分利用计算机网络方式，对大学生进行在线心理教育[1]。

国内研究现状：目前国内进行心理测试的方式有问卷调查方式、谈话方式和线上测试方式。其中前二种方式工作量太大，不适合对高校学生开展大规模心理测试。计算机测试方式可以自动收集数据并进行分析和评定，是目前比较主流的发展方向，有广阔的发展空间。所以国内高校官方网站一般都配有信息中心、线上心理咨询室等平台，但实际使用起来的效果并不理想。

现存问题：（1）网络自由问题、个人隐私保护问题。由于网络保密性能比较差．现有的网络技术还不能做到切实有效地保护使用者的隐私和秘密，同时网络病毒和黑客随时都有可能破坏测试者所使用的电脑和网络测试资源，使测试者在使用网络进行 测试活动时缺乏足够的安全感，也就使得测试更多地受制于网络技术条件和手段。(2)网络信息资源的管理问题。网络的广泛应用，为心理健康测试活动提供了大量的信息，成为心理健康测试的信息资源。这些信息资源对于心理健康测试活动起着积极的作用，但是，我们也要清醒地看到，网络提供的这些庞杂多样的信息，既有测试工作所需要的正确有效的信息，也有不少是与测试工作无关甚至是对测试活动起着反作用的信息。这些问题的克服与解决，都要依靠网络技术手段的提高与网络信息资源管理措施的配合[2]。

总之，近年来，互联网时代下，心理测试将通过网络这一载体广泛传播，所以心理测试技术的应用，仍是一个不断探索、研究和发展的过程。在互联网技术快速发展的基础上，心理测试应该保证实事求是树立良好的网络测试新观念，最大限度地利用网络作为心理健康测试新型载体的优势。

# 1 绪论

## 选题背景

大学时期是成长阶段最重要的过渡时期，心理特性还处在发展阶段，且大学生充满活力，年轻气盛，思想活跃，心理状况波动幅度大，情绪易受外界环境影响，学校应该密切关注其心理状况。近年来,大学生因心理健康障碍造成的悲剧时有发生，比如大连理工大学化学工程专业硕士三年级学生，因学业问题可能导致延期毕业，最终选择了终结自己的性命。

及时帮助心理存在障碍的学生恢复心理健康，为高校建立高校的心理教育平台是当前最主要的任务。然而，现阶段不少高校的心理健康教育模式仍停留在面对面心理咨询的水平上，多数学生不愿尝试面对面心理咨询，对一些隐私问题因为面谈而难以启齿[3]。利用互联网技术构建高校多元交互式心理自我测评平台可以定时跟踪学生心理状态，以便及时发现学生心理问题，进行适度干预，为学生提供切实帮助，有效地缓解这种严峻形势。大学生可以通过该功能模块进行自我测评，系统给出测试结果，帮助学生更快的认识自己并且了解自己的心理状态，为其提供参考依据，对存在的心理问题提前采取有效的干预措施，从源头上避免悲剧的发生。

## 关键技术介绍

### 1.2.1 Spring Boot

本项目是基于Java语言来编写的，作为Java语言最好的框架Spring 生态提供了很多场景开箱即用的解决方案，不必担心编写大量冗余代码，节省了大量的时间和内存。其他组件框架也都需要跟Spring进行集成，例如接下来要介绍的MyBatis，Spring Security 等等，需要使用这些框架实现项目需求，为此Spring 也面临一些问题，就是要整合很多组件、容器来搭建项目。因此Spring 生态产生了Spring Boot这一优秀的脚手架，方便使用者集成各种组件，并且摒弃上一代SSM架构的复杂XML配置，仅用非常少量的配置就可以完成项目需求。还可以内嵌Tomcat，Undertow等容器，免去配置容器、部署war包的麻烦，直接部署成jar包即可使用java -jar 命令启动项目，简单便捷。基于以上种种的优点，本项目采用Spring Boot来搭建基础架构。

### 1.2.2 MyBatis

持久层的框架也有很多，相比于JPA和Hibernete等其他纯ORM持久层框架，MyBatis是一种SQL到POJO的模型，它需要我们提供SQL、映射关系（XML或注解）和映射关系。MyBatis几乎免除了JDBC的代码配置，可以通过简单的XML或者注解的方式来映射原始类型（由于注解的方式和Java代码会混在一起，影响代码的阅读，本项目使用XML来编写SQL），无需像纯ORM框架那样来设计映射关系，只需写好SQL就可以完成业务的开发。

### 1.2.3 Redis

Redis可以说是当前互联网最流行的NoSQL数据库，也可以说是缓存数据库，具有一定的持久化的功能（AOF和RDB模式）。Redis读写数据快。因为Redis是基于内存的，采用多路I/O服用模型，并且底层是C语言编写的，采用单线程来避免多线程切换所导致的竞争CPU。Redis 虽然是Key-Value数据库，但是它支持五种数据类型分别是set，zset，list，hash，string，无数据用户在进行字符串的转化。

### 1.2.4 Spring Security

Spring Security 是一款非常强大的身份验证和访问控制框架，支持高度定制化。主要是对Spring 的项目进行安全防护，所支持的功能比较全面，包括数据加密、请求拦截等功能，相比于Shiro来说Spring Security 可以更好的与Spring进行集成，功能更为强大。缺点是上手难度较大。本项目使用它进行系统登录、权限验证以及其集成的QQ登录功能。

### 1.2.5 Vue JS

作为前端JS框架，Vue可以说是目前最流行的前端框架，Vue的核心库只关注视图层，快速掌握，可以较好地与第三方库进行整合。本项目使用Vue很好的解决了操作DOM元素的痛苦， 更好的关注与业务功能的开发，并且集成了Axios，Element UI，减少的很多的前端代码量。

### 1.2.6 Element UI

Element 作为一款CSS框架可以很好的和Vue进行兼容，功能全面，页面简约，对于定制化要求不高的前端页面来说，Element可以提供较好的解决方案，在前端展示上节约了大量的代码。

# 系统分析

## 需求分析

通过检索资料分析，本系统应包括以下功能需求和非功能需求

### 功能需求分析

根据具体分析，本系统的使用者可以分为两种用户，即学生和系统管理员。不同用户有不同的权限，具有的功能也存在差异。本系统的用户需求具体如下：

1.系统用户管理：

(1)学生：学生可以对个人基本资料进行修改，自主选择心理测试量表进行测试，查询测试结果，对测试结果进行反馈或者咨询操作。

(2)管理员：管理员对系统进行管理操作，对学生信息进行导入、修改和删除操作，可以添加、修改、删除量表信息，录入测试题库信息；可以对姓名、年龄、性别等基础资料进行维护，对数据库进行备份操作等，具体功能需求如下所示：

2.题库管理：分析本系统的题库管理，可以分为测试管理，量表管理，评判管理，选项管理，维度管理，试题管理。

1. [维度管理](https://ailuoli.cn:9528/dimension.html)：对问题所属维度能够提供简单的查询，录入等操作。
2. [测试管理](https://ailuoli.cn:9528/examination.html)：用户可以选择不同维度下的量表进行测试，测试时包括答题卡，答题计时等细节小功能；答题完成后根据量表评分规则进行分数计算，根据不同维度的分数以及总分输出对应的评判记录。
3. [量表管理](https://ailuoli.cn:9528/scale.html)：管理员可以对新增测试量表手动录入，对已存在的测试量表进行编辑修改删除等操作。
4. [评判管理](https://ailuoli.cn:9528/judge.html)：测试量表分析结果属于不同维度，管理员可以对所需要操作的评判数据进行增删改查。
5. [选项管理](https://ailuoli.cn:9528/options.html)：对心理测试量表的选项进行操作。
6. [试题管理](https://ailuoli.cn:9528/questions.html)：对测试量表试题进行增加，删除，编辑等操作。

3.用户管理：个人基本信息的修改，用户可随时查看测试量表试题和测试成绩。

1. 角色管理：管理员可以修改用户角色与资源的对应信息。
2. 用户管理：根据用户基本信息搜索符合条件的用户数据，管理员可随时对用户信息进行编辑、删除等操作。

从本系统的使用者来看，不同的用户使用不同的功能模块，不同模块之间需要保证数据的交互，当页面数据较多时，实现分页加载。

### 非功能需求分析

1.系统界面需求

因为本系统的主要面向用户是学生和教师，并不熟练对计算机的操作。所以本系统应具有操作简易，界面简洁设计原则。有利于用户快速熟练操作各系统功能。

2.系统性能需求

1. 易使用性：系统的大部分使用者是教师和学生，年龄跨度比较大，界面应设计的简洁大方，无需培训，操作简单易上手。
2. 可靠性： 系统能够满足用户的基本需求。
3. 可维护性：当系统出现异常时，可以很容易进行故障修复（最外层异常处理器），系统故障率在一定的水平线附近波动。
4. 可扩展性： 当后续增加测试量表，心理测试评分类似时，采用统一口径设计，在需要用到的地方可以进行微调即可调用。
5. 响应性能：页面跳转时间小于等于3S,精确搜索时间小于等于1S，减少用户焦虑感。

3.安全性需求

数据安全：由于本系统涉及个人隐私问题，所以将更注重数据的安全方面。系统QQ登录方式的密码经过加密，解密等来实现账户安全，邮箱注册时不仅使用验证码验证身份，而且还使用了邮箱真实性验证，通过邮箱发送验证码，验证码有效时间为5分钟，以此来保证数据的安全性。

## 可行性分析

### 系统管理员

系统管理员注册登录后可以选择修改用户信息、编辑用户权限，增删改查测试量表、选项等功能模块，更新表中数据，实现基本业务。如图2.2所示。



图2.2 系统管理员业务流程图

### 技术可行性

本系统是基于Spring Boot框架开发的心理测试系统，开发工具是IntelliJ IDEA,数据库采用MySQL8.0.21版本，数据库管理工具使用的是操作简单的Navicat15.0；持久层框架使用的是MyBatis框架，开发语言是Java,Java是一种面向对象的编程语言，并且是学校课程，大学期间也学过数据库，数据结构等基础知识。本方案所采用的技术均采用较为成熟的技术，由以上分析可知，本系统的开发在技术上是可行的。

### 经济可行性

系统的开发基于本人项目学习、实践而来，且具有基本的开发设备，参考资料使用的是图书馆电子资源，参考项目是一些开源项目，学校无需资金投入。本系统对计算机配置的要求不高，个人基础设备以及移动端都可使用。本系统的使用周期长，成本相对更低，同时减少了对纸质材料的使用，因此开发经费完全可以接受。

### 社会可行性

本系统的社会可行性主要从法律因素、用户使用可行性两方面进行研究。

1.法律因素

本系统的软件设计是在独立的环境下完成的，使用软件为开源软件，学习资料为权威出版图书，所编写代码不涉及抄袭问题，不存在盗版等版权问题。

2.用户使用可行性

本系统并无复杂操作，具有基本电脑操作知识的人员即可使用，移动终端也可为使用设备，操作简单，无需培训即可使用。

由以上分析可知，本系统具有社会可行性。

## 业务流程分析

本系统用户主要分为两大类：普通用户和系统管理员。

### 普通用户

普通用户注册登陆后，可以选择修改个人信息、心理测试、查看心理测试结果等功能，并将信息存储在数据表中。如图2.1所示。



图2.1 普通用户业务流程图

# 概要设计

## 系统功能设计

系统功能结构图就是按功能将系统划分为细小模块，明确系统主要功能结构，全面了解整个系统的核心模块。如图3.1所示。



图3.1系统功能结构图

### 登录注册模块

1. 用户注册：利用表单验证实现用户的注册功能，格式错误，密码前后两次不一致等会进行相应提示。
2. 普通登录：根据注册时的用户基本信息进行登录。
3. QQ第三方登录：QQ登录系统校验成功授权实现URI逻辑跳转，登录失败会提示用户名或密码错误，用户名密码为空会进行相应提示。

### 心理测试模块

1. [维度管理](https://ailuoli.cn:9528/dimension.html)：本测试系统共分为四个测试维度，分别为人际、生活、学习、就业四个维度，在该模块，可以根据量表大类和维度所属量表搜索框快速检索到所需要查询、修改的维度条目信息，也可以根据在维度描述搜索框根据维度描述的第一个字进行快速匹配。管理员可以进行对维度进行删除、增加等操作。
2. [测试管理](https://ailuoli.cn:9528/examination.html)：用户可以选择不同维度下的量表进行测试，测试时包括答题卡，答题计时等细节小功能；答题完成后根据量表评分规则进行分数计算，根据不同维度的分数以及总分输出对应的评判记录。
3. [量表管理](https://ailuoli.cn:9528/scale.html)：本系统在四个测试维度下又包含了八个量表，在本功能模块中，可以根据文本输入框查询所要修改的量表所属大类名称，也可以输入大类个别简短名称，输入框会自动提示大类全称，进行量表快速检索。管理员可以通过本模块进行删除，增加量表等操作。新增量表时需要添加量表所属大类和量表描述（量表名称）。
4. [评判管理](https://ailuoli.cn:9528/judge.html)：心理测试量表得评分规则的管理模块，不同测试量表有不同的评分规则，测试量表分析结果也属于不同维度，在该模块可以实现对评判规则的删除增加等操作。在该模块系统管理员可以根据量表大类、所属量表，所属维度、起始分数和结束分数等标签快速检索所需要操作的评判数据。
5. [选项管理](https://ailuoli.cn:9528/options.html)：对心理测试量表的选项进行操作。在系统的该模块下，系统管理员可以根据所属问题、选项编码，选项描述进行模糊快速匹配所需要查询、操作的选项数据。完成对选项的编辑，增加，删除等功能操作。
6. [试题管理](https://ailuoli.cn:9528/questions.html)：对测试量表试题进行查询、增加，删除，编辑等操作。在该模块下，可以根据量表大类，所属量表，所属维度，问题描述等检索框快速进行检索匹配。

### 用户信息模块

1. 用户可以在个人中心页面查询自己的答题记录列表和点击详情查看答题详情、结果。
2. 用户可以并且可以修改自己的基本的信息，头像，生日等。
3. 用户可以对账号相关信息，如密码，邮箱进行更改。

## 系统架构设计

### 系统类图

1. 实体类图，从数据需求分析中得出系统的实体属性图。图中展示了实体之间的关联关系如图3.2、3.3所示。

1

图3.2用户角色相关类图

2

图3.3主业务类图

1. 主要业务流程类图，代码业务遵循MVC的模式进行开发，从持久层到业务层再到控制层最后到前端进行代码的编写，如图3.4 所示。

3

图3.4业务流程类图

1. 系统认证与权限架构基于Spring Security框架进行代码的编写，相关的处理器与过滤连比较复杂，如图3.5所示。

4

图3.5安全认证与权限类图

### 数据库设计

1. 概念结构设计
2. 用户信息表（用户编号，密码，用户名，QQ登录标识，QQ昵称，手机号，昵称，头像，生日，邮箱，用户状态，认证类型，性别，最后登录时间，最后更改密码时间，更新时间，创建时间）
3. 测试量表信息表（量表编号，量表描述，量表所属类型，创建时间，更新时间）
4. 测试记录表（记录编号，量表编号，用时，用户编号，答案记录结果，创建时间，更新时间）
5. 测试问题表（问题编号，问题描述，问题所属维度编号，创建时间，更新时间）
6. 测试问题维度表（条目编号，维度编号，量表编号，维度描述，创建时间，更新时间）
7. 测试问题选项表（选项编号，维度编号，量表编号，选项描述，选项分数，创建时间，更新时间）
8. 测试评判表（评判编号，评判分数段开始分数，评判分数段截止分数，评判描述）
9. 维度评判关联表（关联编号，维度编号，评判编号）

从数据需求分析中得出系统总体E-R属性图。如图3.6所示。



图3.6系统总体E-R图

2、数据库表设计

数据表总览: 数据表是存放系统所需的数据的集合，根据上节的实体类图分析，可以设计出各个数据表存放的数据信息。

表3.1 数据总览表

|  |  |  |
| --- | --- | --- |
| 数据表名 | 简介 | 功能 |
| psyc\_user | 用户信息表 | 存储用户基本信息 |
| psyc\_role | 角色表 | 存储用户角色信息 |
| psyc\_permission | 资源权限表 | 存储资源权限信息 |
| psyc\_role\_permission\_rlt | 用户权限关联表 | 存储不同用户的权限信息 |
| psyc\_user\_role\_rlt | 用户角色关联表 | 存储不同用户的角色信息 |
| persistent\_logins | 记住我表 | 存储用户持久登录令牌信息 |
| psyc\_dimension\_judge\_rlt | 维度评判关联表 | 存储维度与评判的关联信息表 |
| psyc\_exam\_record | 测试记录表 | 存储用户的测试记录信息 |
| psyc\_question | 测试问题表 | 存储测试量表问题信息 |
| psyc\_question\_dimension | 测试问题维度表 | 存储量表问题所属维度信息 |
| psyc\_question\_judge | 测试问题评判表 | 存储测试量表问题信息 |
| psyc\_scale | 测试量表信息表 | 存储测试量表信息 |

1、用户管理模块

该模块需要介绍一下用户模块采用的RABC（Role-Base Access Control）基于角色的访问控制模型来设计的数据库表，系统权限不直接授予到用户，而是授予到角色，然后一个用户可以拥有哪些角色，实现了用户和权限逻辑的分离，极大的方便管理权限。此模型下用户、角色、权限都是一对多的关系，所以有了以下5个表。

（1）psyc\_user用户信息表：需要说明的是表中的qq\_open\_id字段，本项目集成了QQ登录，QQ登录所使用的是OAuth2协议，本系统在无需知道登录用户名密码的情况下获取用户的一些信息。在 QQ 第三方登录的时候，腾讯开放平台提供的信息当中，唯一能确定用户唯一身份的字段就是open\_id，所以用户使用QQ登录的时候将该字段存储在数据库，使用该字段确定用户的唯一性。如表3.2所示。

表3.2 用户信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名称 | 数据类型 | 是否为空 | 字段长度 | 字段含义 |
| user\_id | bigint | NO | 20 | id |
| password | varchar | NO | 255 | 密码 |
| username | varchar | NO | 50 | 用户名 |
| qq\_open\_id | varchar | NO | 50 | QQ登录openid，第三方登陆时识别用户信息的唯一标识 |
| qq\_nickname | varchar | NO | 255 | qq昵称 |
| phone | varchar | NO | 30 | 手机号 |
| nickname | varchar | NO | 50 | 昵称 |
| figure\_url | varchar | NO | 400 | 头像 |
| birth\_date | datetime | NO |  | 生日 |
| email | varchar | NO | 50 | 邮箱 ( 检验用户的 ) |
| user\_status | smallint | NO | 5 | 1正常 0 冻结 |
| auth\_type | int | NO | 5 | 认证类型10 邮箱，20 qq登录 |
| sex | smallint | NO | 5 | 10男 20 女 0其他 |
| last\_user\_login\_date | datetime | NO |  | 最后登录时间 |
| last\_password\_rest\_date | datetime | NO |  | 最后更改密码时间 |
| update\_date | datetime | NO |  | 更新时间 |
| create\_date | datetime | NO |  | 创建时间 |

（2）psyc\_role角色表：如表3.3所示。

表3.3 角色表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名称 | 数据类型 | 是否为空 | 字段长度 | 字段含义 |
| role\_id | int | NO | 11 | 角色id |
| role\_code | varchar | NO | 10 | 角色编码 |
| role\_status | smallint | NO | 5 | 角色状态，1 为默认 |
| remark | varchar | NO | 255 | 备注 |

（3）psyc\_role\_permission\_rlt角色权限关联表：如表3.4所示。

表3.4 角色权限关联表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名称 | 数据类型 | 是否为空 | 字段长度 | 字段含义 |
| role\_permission\_id | int | NO | 11 | id |
| permission\_id | int | NO | 11 | 权限id |
| role\_id | int | NO | 11 | 角色id |
| remark | varchar | NO | 255 | 备注 |

（4）psyc\_user\_role\_rlt用户角色关联表：如表3.5所示。

表3.5 用户角色关联表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名称 | 数据类型 | 是否为空 | 字段长度 | 字段含义 |
| user\_role\_rlt\_id | int | NO | 11 | id |
| role\_id | int | NO | 11 | 角色id |
| user\_id | bigint | NO | 20 | 用户id |
| remakr | varchar | NO | 255 | 备注 |

（5）psyc\_permission 资源权限表：如表3.6所示。

表3.6 资源权限表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名称 | 数据类型 | 是否为空 | 字段长度 | 字段含义 |
| permission\_id | int | NO | 11 | 权限id |
| permission\_type | smallint | NO | 5 | 权限类型 10 接口 20 页面 |
| permission\_url | varchar | NO | 255 | 权限路径 |
| data\_feather | varchar | NO | 255 | 导航图标 |
| group\_code | varchar | NO | 20 | 组名 |
| group\_name | varchar | NO | 255 | 组描述 |
| page\_name | varchar | NO | 255 | 页面名称 |
| remark | varchar | NO | 255 | 备注 |
| priority | int | NO | 11 | 优先级（先后顺序） |
| auth\_type | smallint | NO | 5 | 认证级别0 匿名 1 登录 |

6、persistent\_logins：记住我表，该表是记住我功能的表，用户记住我登录后会生产对应的token存储到该表中，该表并不是我们主动建立的，而是我们使用Spring Security框架项目启动时自动创建的表（在代码中配置好记住我功能）。如表3.7所示。

表3.7 记住我表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名称 | 数据类型 | 是否为空 | 字段长度 | 字段含义 |
| token | varchar | NO | 64 |  |
| last\_used | timestamp | NO |  |  |
| series | varchar | NO | 64 |  |
| username | varchar | NO | 64 |  |

2、测试管理模块

该模块是主要业务模块，主要是对心理测试的量表所有的属性进行抽象建立的表。该业务场景中主要是以最大单位量表进行心理测试，量表中的题分为不同的维度，最后的计算结果是针对每个维度的标准进行计算的评判，将答题结果存储在记录表里，所有产生了以下几个表。

7、psyc\_dimension\_judge\_rlt维度评判关联表：维度与评判是一对多的关系，一个维度对应有多个不同的评判结果。如表3.8所示。

表3.8 维度评判关联表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名称 | 数据类型 | 是否为空 | 字段长度 | 字段含义 |
| id | int | NO | 11 | id |
| dimension\_id | int | NO | 11 | 维度id |
| judge\_id | int | NO | 11 | 评判id |

8、psyc\_exam\_record测试记录表：存储每个用户的测试记录。如表3.9所示。

表3.9 测试记录表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名称 | 数据类型 | 是否为空 | 字段长度 | 字段含义 |
| record\_id | bigint | NO | 11 | id |
| scale\_id | int | NO | 11 | 量表id |
| use\_time | datetime | NO |  | 用时 |
| user\_id | bigint | NO | 20 | 用户id |
| record\_content | blob | NO | 65535 | 答题记录正文 |
| create\_date | datetime | NO |  | 创建时间 |
| update\_date | datetime | NO |  | 更新时间 |

9、psyc\_question：测试问题表：如表3.10所示。

表3.10 测试问题表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名称 | 数据类型 | 是否为空 | 字段长度 | 字段含义 |
| id | int | NO | 11 | id |
| question\_name | varchar | NO | 400 | 问题描述 |
| dimension | int | YES | 11 | 维度id |
| create\_date | datetime | NO |  | 创建时间 |
| update\_date | datetime | NO |  | 更新时间 |

10、psyc\_question\_dimension测试问题维度表：不同量表下的测试问题所属不同的维度。如表3.11所示。

表3.11 测试问题维度表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名称 | 数据类型 | 是否为空 | 字段长度 | 字段含义 |
| dimension\_id | int | NO | 11 | id |
| dimension\_code | varchar | NO | 20 | 维度编码 |
| scale\_id | int | NO | 11 | 量表id |
| dimension\_name | varchar | NO | 255 | 维度描述 |
| create\_date | datetime | YES |  | 创建时间 |
| update\_date | datetime | YES |  | 更新时间 |

11、psyc\_question\_judge 测试问题评判表：每一个测试问题都属于某一个测试维度，每一个测试维度在不同分数段又有相应的评判结果。如表3.12所示。

表3.12 测试评判表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名称 | 数据类型 | 是否为空 | 字段长度 | 字段含义 |
| judge\_id | int | NO | 11 | id |
| end\_score | int | NO | 11 | 评判分数段数 |
| start\_score | int | NO | 11 | 评判分数段数 |
| judge\_name | varchar | NO | 400 | 评判描述 |

12、psyc\_question\_options测试问题选项表：每一个问题都有可供选择的选项，选项个数不等，没有正确与否之分。如表3.13所示。

表3.13 测试问题选项表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名称 | 数据类型 | 是否为空 | 字段长度 | 字段含义 |
| option\_id | int | NO | 11 | id |
| option\_code | varchar | NO | 10 | 选项编码 |
| question\_id | int | NO |  | 所属问题id |
| option\_name | varchar | NO | 255 | 选项描述 |
| option\_score | int | NO |  | 选项分数 |
| create\_date | datetime | NO |  | 创建时间 |
| update\_date | datetime | NO |  | 更新时间 |

13、psyc\_scale 测试量表表：如表3.3所示。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名称 | 数据类型 | 是否为空 | 字段长度 | 字段含义 |
| scale\_id | int | NO |  | 量表id |
| scale\_name | varchar | NO | 255 | 量表描述 |
| scale\_type | int | NO |  | 量表类型（10人际，20学习，30就业，40情感） |
| create\_date | datetime | NO |  | 创建时间 |
| update\_date | datetime | NO |  | 更新时间 |

表3.14 测试量表表

# 详细设计与实现

在该章节，从系统架构和设计实现两方面来详细说明架构和业务之间是如何实现的以及遇到问题和解决方案。

## 项目目录结构设计

项目架构时首先需要有一个合理的目录结构，在开发时才能更好的管理项目中各类型文件。项目开发遵循现在普遍的MVC（Modle Veiw Controller）模式，所以model 文件夹存放项目实体类，包括数据库的映射类、扩展类、入参类、枚举等；controller文件夹作为控制层与页面进行交互存放controller类；service层作为业务层处理具体业务，存放一些业务接口和其实现类；mapper目录作为持久层，存放与XML映射的Mapper类；config目录存放系统架构的一些必要的配置类，包括Spring Security、Knife4j、全局异常处理、自定义加密类等；filter存放全局的过滤器；uiti存放的是开发过程中的工具类；resources文件夹下存放项目用到的静态文件，XML、前端页面、全局配置等静态文件存放在该目录下。如图4.1所示。

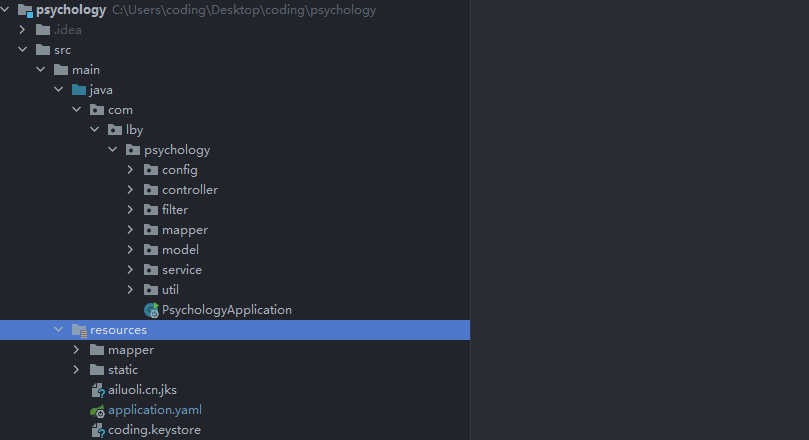


图4.1项目文件架构图

## 登录设计

### 前端设计

该页面前端主要是一个表单，对输入框校验邮箱格式（正则表达式）和密码（长度不小于6位）格式并且支持响应式，根据不同的分辨率呈现自适应的页面，该项目的登录集成了第三方QQ登录，所以在下面添加QQ登录的logo，点击会跳转到QQ登录页面。常规登录失败会进行提示，这个是用的Element UI提供的Message组件，根据后端返回不同结果进行提示。

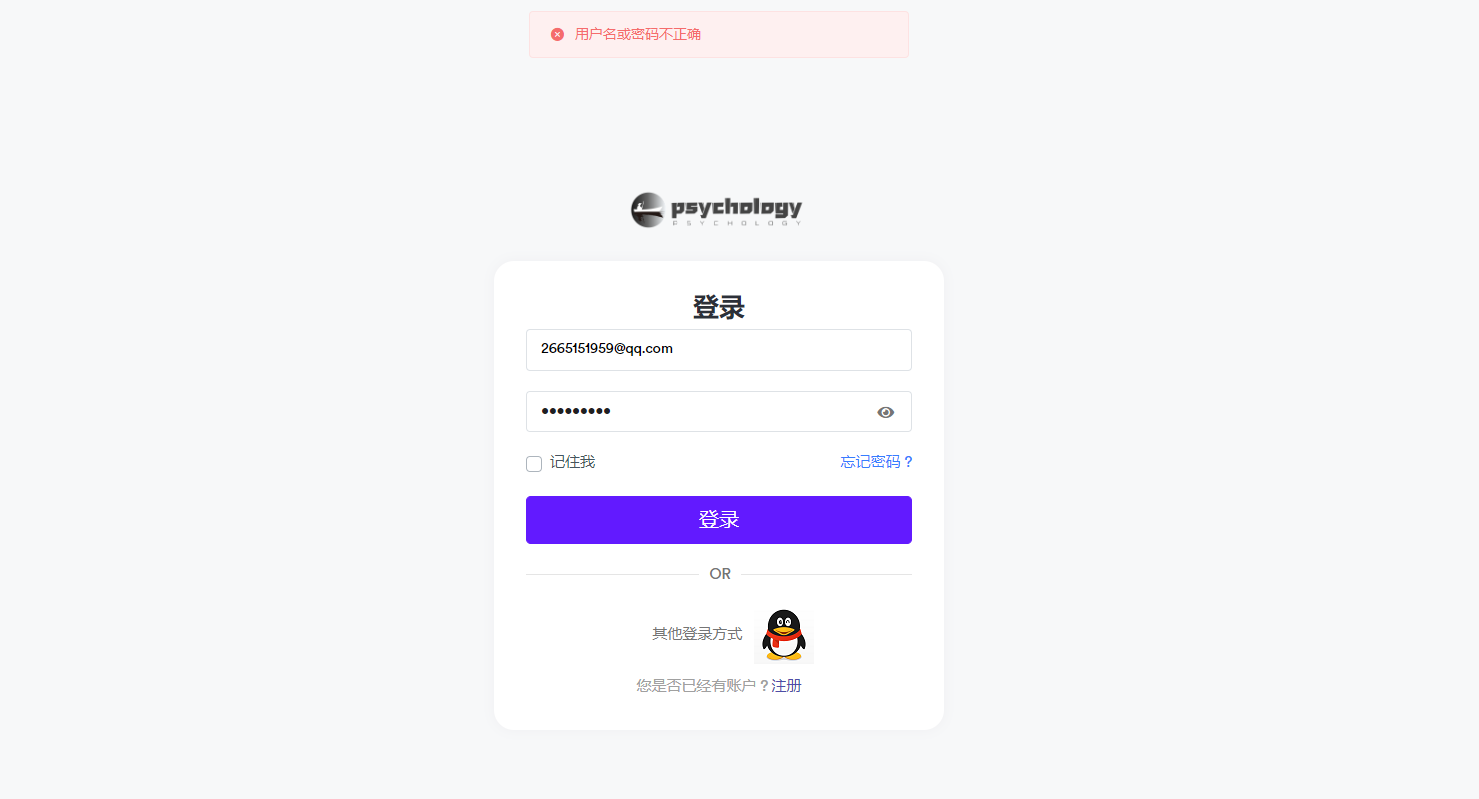


图4.2普通登录前端页面



图4.3 QQ登录前端页面

### 后端设计

1、普通用户登录

本项目是基于Spring Security 做的验证和授权，所以相关的过滤链是由Spring Security做的，主要过滤器为UsernamePasswordAuthenticationFilter ，在该过滤器中先获取前端传入的username和password 构建UsernamePasswordAuthenticationToken，调用setDetails 方法给details进行赋值，detail里存放着本次请求的remoteAddress和生成的sessionId。然后对detail进行校验，首先获取AuthenticationManager，找到对应的provider 调用authenticate方法，从Authentication对象中提取用户名调用项目自定义的查询逻辑，也就是UserServiceDetails中的loadUserByUsername方法。在本项目中为了使登录更加安全，前后端传输用户密码的时候使用了非对称加密，前端先获取publicKey，再使用publicKey进行加密，将加密结果作为密码传给后端；所以这里先使用privateKey对密码进行解密，得到真实密码后再将查询到的user对象返回（该方法返回Spring Security的user对象）。接下来由preAuthenticationChecks.check方法对用户密码进行校验（数据库里存放的是加密后的密文，该加密是由Spring Security 配置的加密算法，与前后端的非对称加密不同），如果通过完整的过滤链后会构建一个新的UsernamePasswordAuthentocationToken对象，最后登录成功后会将用户信息通过SecurityContextHolder.getContext().setAtuhentication()方法将用户信息存储到Java内存中，完成一次正常的登录流程。登录流程较复杂，过滤链较繁琐，本文不一一详细介绍，感兴趣的同学可以查阅网上资料详细了解。

如果用户在登录时勾选了记住我，那么在过滤链中还会经过SecurityContextPersistenceFilter调用RememberMeService，利用TokenRepository生成用户对应的token写入到Cookie中，同时存储到数据库中进行持久化，再次请求经过RememberMeAuthenticationFilter时验证token，这样就实现了记住我功能。

2、QQ用户登录

与普通登录的区别在于用户的一部分过滤链需要自定义。QQ登录首先需要在腾讯开放平台申请网站资格，需要填写系统登录成功的回调地址。申请通过后会生成一个APP ID 和 APP K，由于腾讯开放平台提供的第三方登录接口是基于OAuth2协议的，并且采用的授权码模式，所以APP ID 和 APP K相当于是客户端Id和Secret。该登录的流程为用户登录OAuth Server 后端，即腾讯端先进行认证，如果认证成功回调项目接口（申请资格时所填写的地址），返回一个code，因为本项目是与Spring Security进行集成，需在项目中自定义授权逻辑，因此自定义一个过滤器实现AbstractAuthenticationProcessingFilter，重写attemptAuthentication()方法，请求腾讯接口获取access\_token，成功后利用令牌获取用户信息。所以项目先获取openId，然后组建UsernamePasswordAuthenticationToken，返回结果。接下来还需要进行自定义一个类来实现AuthenticationManager，写认证成功后本系统应进行的操作。

1. 首先通过access\_token 获取用户性别、昵称、头像等基本信息。
2. 构建系统用户对象，并将用户信息存储在psyc\_user 表中，实现持久化，也算是同时登录并注册（通过openId查询，若表中有数据，就更新该条数据，openId可以确认唯一一个QQ用户）。
3. 赋予QQ用户默认的访问权限，查询数据库默认角色，插入用户角色关联表，为用户赋予默认的角色。

以上步骤结束后构建新的UsernamePasswordAuthenticationToken并返回。至此自定义结束，剩下的可以交给Spring Security过滤链处理，与普通用户登录一样。

整体登录流程图如图4.3所示。



图4.3 登录流程图

## 注册设计

### 前端设计

注册前端主要是对表单的验证，昵称不能为空，邮箱正则表达式，两次密码不能为空且大于6位，验证码不能为空。倒计时读秒按钮如图4.4所示。

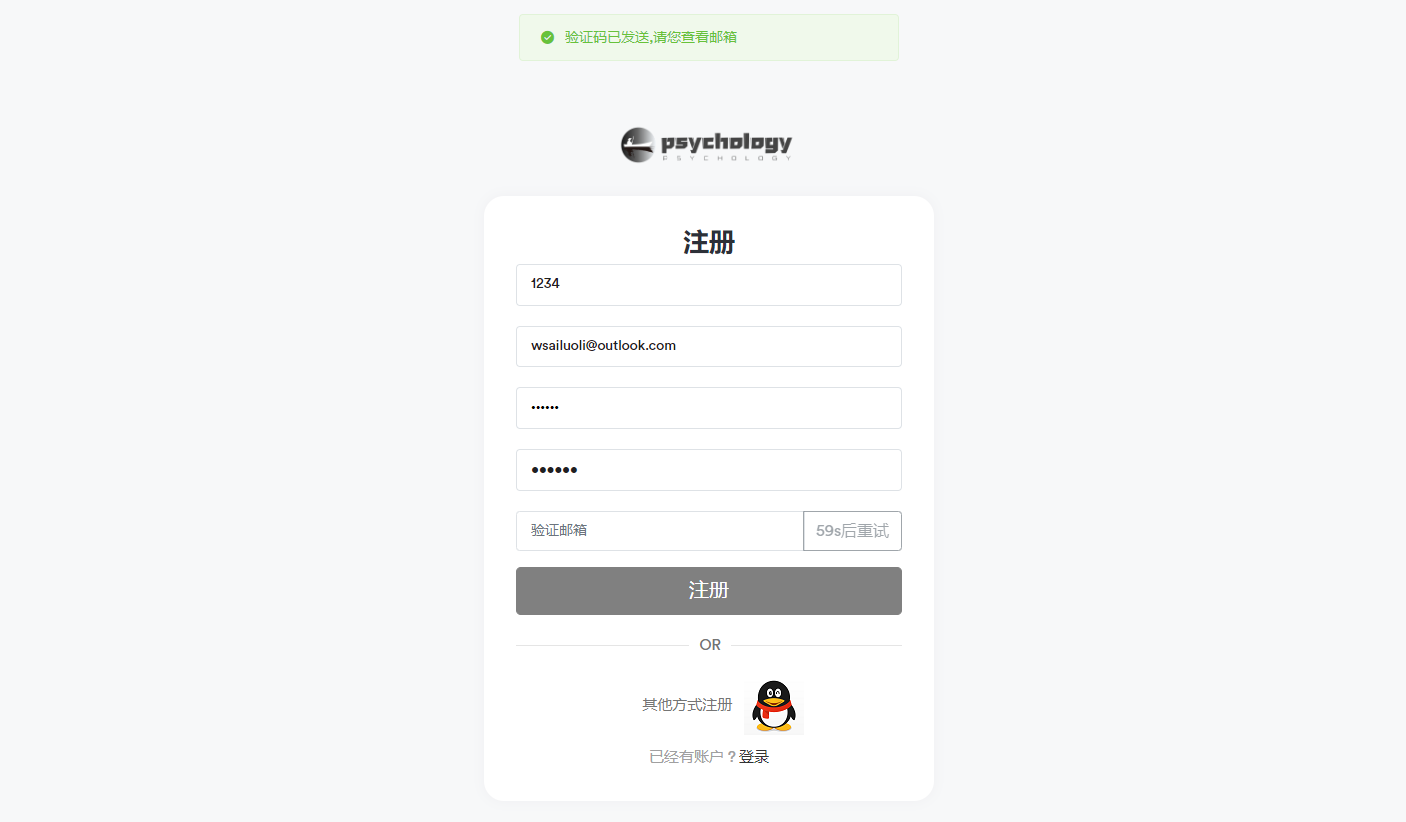


图4.4 注册前端页面

### 后端设计

注册功能后端主要是一个新增用户的过程。首先对用户邮箱进行验证码校验（这里传输用户密码依然进行了非对称加密操作，防止密码被浏览器插件窃取），验证码有效期为300s，过期自动失效，这里利用的是Redis特性：邮箱为Key，随机生产的验证码为Value放进Redis，并发送给用户邮箱（发送验证码的邮箱使用的是个人QQ邮箱，需要集成Spring Boot Mail组件）。用户收到验证码后，输入点击注册后有以下两个步骤。

1. 将密码非对称解密后用Spring Security 的BCryptPasswordEncoder对明文进行加密存储在psyc\_user表中（数据库中尽量不直接存储明文密码）。
2. 获取角色表中默认角色，插入用户角色关联表，为用户绑定默认角色。

注册流程图如图4.5所示。



图4.5 注册流程图

### 发送验证码核心代码

public void checkEmailRegistered(RegisteredUserVo registeredUserVo) {

//检验用户是否存在

if(userMapper.selectEmailCount(registeredUserVo.getEmail()) > 0){

throw new PsycException(EnumResponseType.EMAIL\_IS\_EXIST);

}

Integer code = commonUtil.getRandomNumber();

log.info("开始发送验证码：{}",code);

commonUtil.sendMailCheckCode(registeredUserVo.getEmail(),code.toString());

//将验证码设置到redis内，过期时间60s

redisUtil.set(EnumRedisPre.EMAIL\_REG\_CHECK.getCode()+registeredUserVo.getEmail(),code,expired);

}

### 注册核心代码

public boolean registeredPsycUser(RegisteredUserVo registeredUserVo) {

//用户名和邮箱相同

registeredUserVo.setUsername(registeredUserVo.getEmail());

boolean result;

//获取默认角色

List<PsycRole> authorityList = roleMapper.selectDefaultRoleList();

String code = String.valueOf(redisUtil.get(EnumRedisPre.EMAIL\_REG\_CHECK.getCode()+registeredUserVo.getEmail()));

if(!StringUtils.isEmpty(registeredUserVo.getCode()) &&!registeredUserVo.getCode().equals(code)){

throw new PsycException(EnumResponseType.VALID\_CODE\_NOT\_SAME);

}

//将秘密进行非对称解密，然后加密保存

registeredUserVo.setPassword(passwordEncoder.encode(passwordEncoder.decodeByRSA(registeredUserVo.getPassword())));

result = userMapper.insertPsycUser(registeredUserVo) > 0;

List<PsycUserRoleRlt> psycUserRoleRltList = new ArrayList<>();

for(PsycRole role : authorityList){

PsycUserRoleRlt psycUserRoleRlt = new PsycUserRoleRlt();

psycUserRoleRlt.setUserId(registeredUserVo.getUserId());

psycUserRoleRlt.setRoleId(role.getRoleId());

psycUserRoleRlt.setRemark("qq登录授予用户默认角色，用户名为"+registeredUserVo.getUsername());

psycUserRoleRltList.add(psycUserRoleRlt);

}

roleMapper.insertRoleUserRlt(psycUserRoleRltList);

return result;

}

## 题库设计

该模块涉及到较多子模块的实现，其他模块与此大同小异，因此下文仅阐述试题管理一个子模块的设计与实现。

### 前端设计

1. 列表页：根据不同条件查询试题，所以查询条件为一个表单，量表大类和所属量表，所属维度为级联下拉，问题描述支持左半模糊查询（考虑到查询效率问题，将问题描述设为索引，索引命中问题没有设置全模糊查询）。列表为一个Element的表格，下方为分页器，需要和后端同步分页字段。给每个问题做了一个下级选项详情展开。
2. 新增/编辑页，该页面较简单，为一个表单，量表大类，所属量表，所属维度为级联的下拉。
3. 删除弹窗是Element的MeseeageBox组件。如图4.6，4.7，4.8所示。

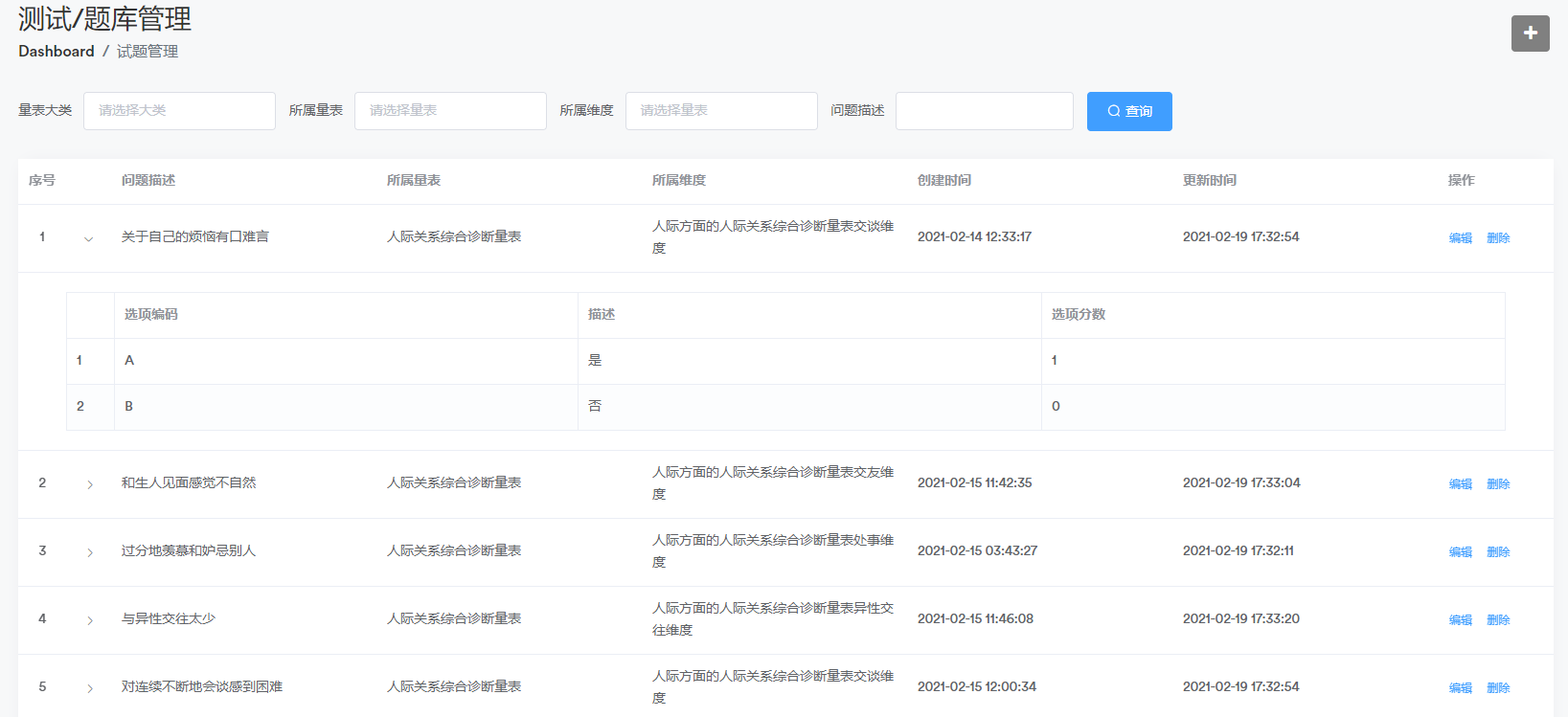


图4.6 试题管理列表页

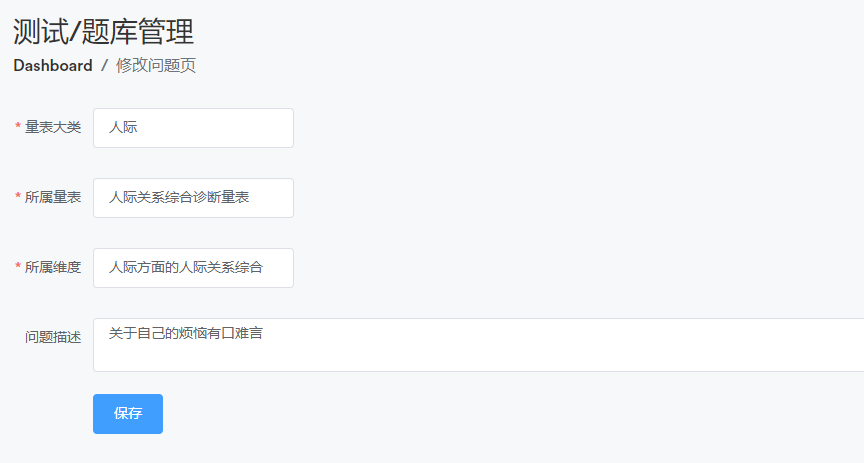


图4.7 试题编辑/新增页

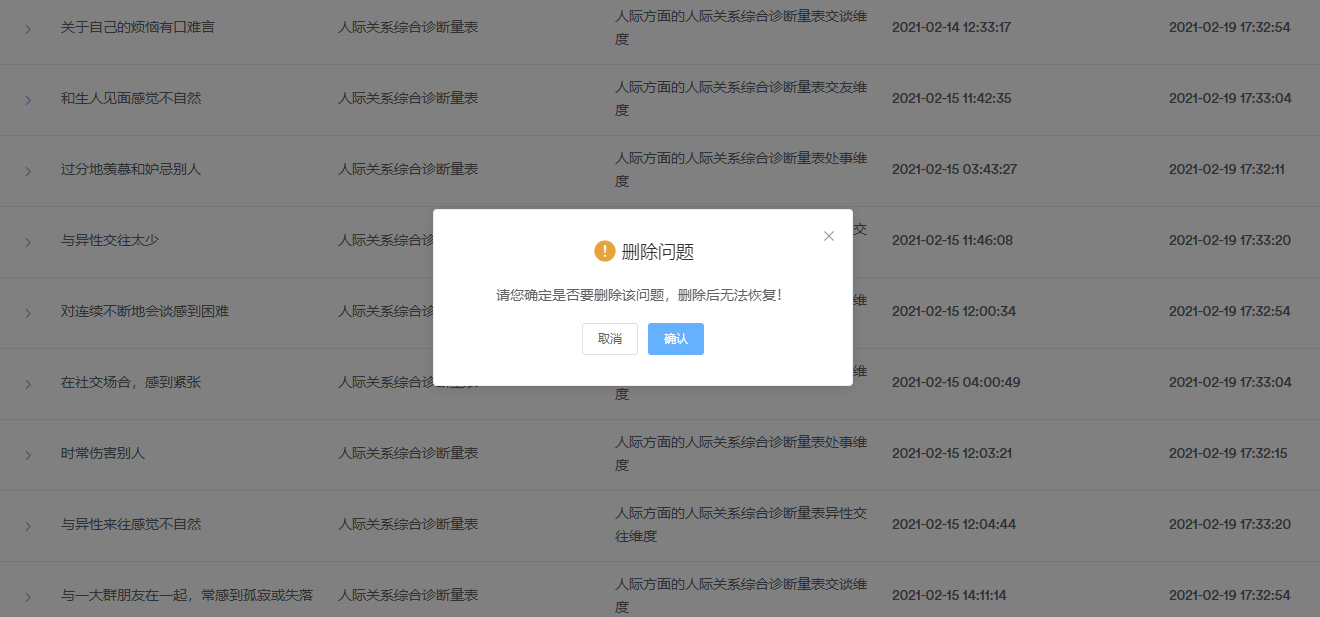


图4.8 试题删除页

### 后端设计

1. 列表接口，列表页功能主要是对分页功能的实现，后端集成的分页插件是PageHelper组件，获取到问题列表后根据查询结果的id查询对应的选项，因此在返回结果实体中还要加入选项List，再将查询到分页结果和列表数据封装为一个实体返回。因为分页需要和后端进行配合，前端同样需要pageNum,totalPages,totalSize,pageSize等分页必备数据。前端根据这些数据计算展示的问题，并且下一次传入分页信息的时候，才能知道传入第几页，每页多少等这些分页的必要信息，PageHelper组件也依赖这些信息进行分页查询。
2. 新增/编辑接口，对于问题没有参数的校验，在前端校验，编辑的时候将详情信息返回给前端并显示在输入框里，点击保存调用更新接口。
3. 删除接口，删除接口、删除问题的同时也要删除相关的选项。

后端流程图如图4.9所示。



图4.9 试题查询、删除流程图

### 题库管理核心代码

public PageResult getQuestionPageList(PsycQuestionCo co) {

PageHelper.startPage(co.getPageNum(),co.getPageSize());

List<PsycQuestionVo> list = questionMapper.selectQuestionPageList(co);

list.forEach(e ->{

//获取相关选项

e.setOptionList(optionsMapper.selectOptionByQuestionId(e.getId()));

});

PageInfo<PsycQuestionVo> pageInfo = new PageInfo<>(list);

return PageResult.getPageResult(pageInfo);

}

## 测试设计

### 前端设计

答题页设计的较简洁大气。左边为答题卡，根据问题是否作答，判断题号边界框的颜色。提交按钮，答题人，计时器，测试量表名称等信息展示在答题卡表头。右边为试题详情，问题只可单选，所有试题为一个表单，并且做了一个是否答完所有试题的校验，确保答完所有试题之后才能进行提交；答题页也是响应式的页面，可以自适应屏幕。提交答题后会有对应的弹窗展示答题结果，考虑到用户体验，将评判内容设置为较鲜艳的棕色，并在弹窗右下角添加了一个返回个人中心的链接，关闭答题弹窗后也会自动跳转到个人中心页，如图4.10，4.11所示。



图4.10 测试页

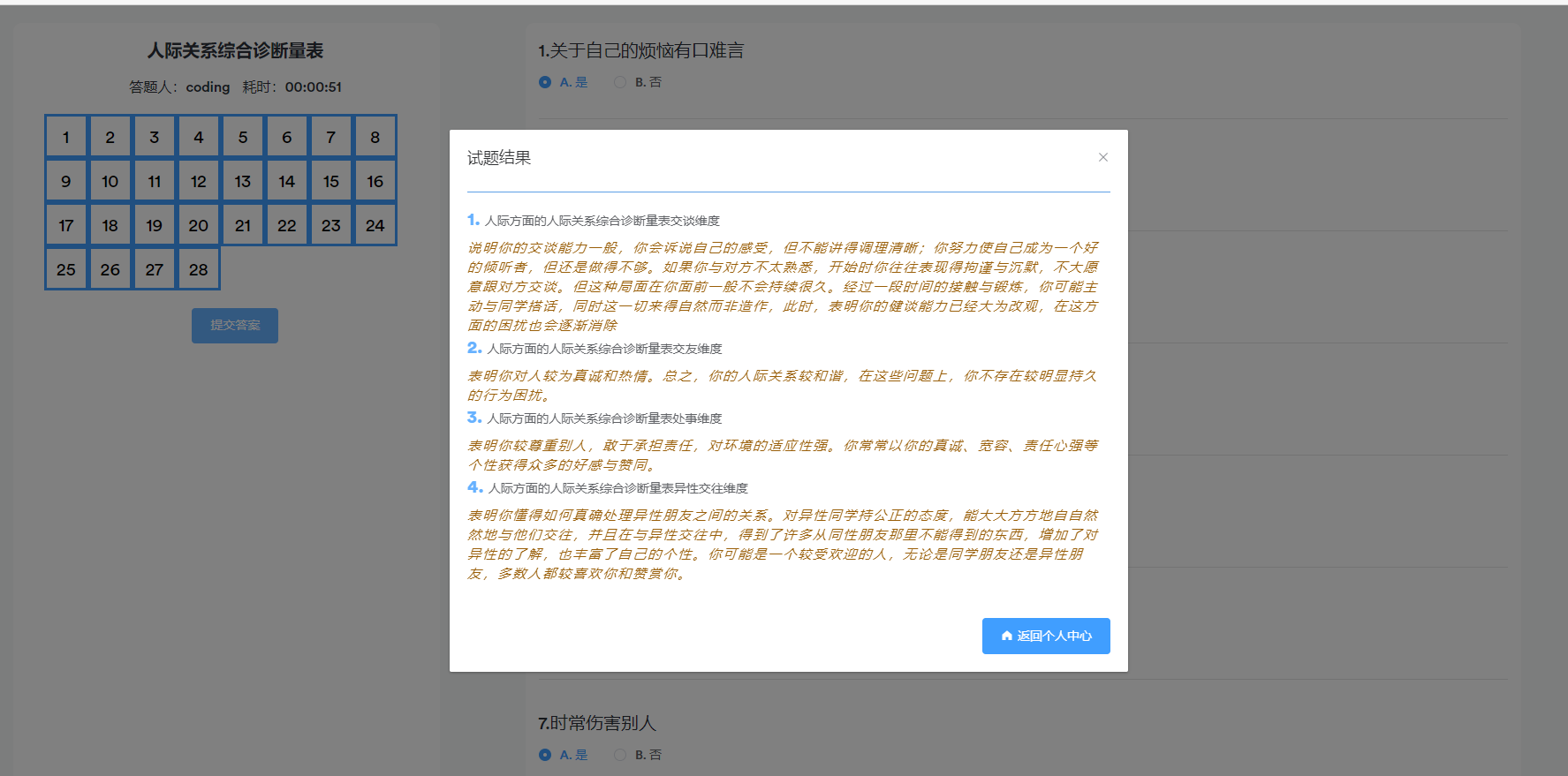


图4.11 测试结果页

### 后端设计

后端将前端表单结果按维度分组，计算出正确的评判内容并返回给前端，并将答题记录存储下来。流程图如图4.12所示。



图4.12 测试流程图

### 计算答题结果核心代码

public List<PsycJudgeVo> examScale(PsycExamCo co, Long userId) throws JsonProcessingException {

List<PsycJudgeVo> result = new ArrayList<>();

//计算答题,将list中的数据分组 将PsycQuestionVo按维度Dimension(维度ID）分组，value为一个list<PsycQuestionVo>

Map<Integer,List<PsycQuestionVo>> map = co.getQuestionList().stream().

collect(Collectors.groupingBy(PsycQuestionVo::getDimension));

//按维度计算

for(Map.Entry<Integer,List<PsycQuestionVo>> entry : map.entrySet()) {

List<PsycQuestionVo> questionVoList = entry.getValue();

//获取维度分数求和

Integer score = questionVoList.stream()

.mapToInt(e-> e.getOptionList().

stream().

filter(v->v.getOptionCode().equals(e.getOptionCode())).

collect(Collectors.toList()).get(0).getOptionScore()).

sum();

//判断该分数在哪个评判

result.add(psycQuestionJudgeMapper.selectJudgeByScore(entry.getKey(),score));

}

//将结果存储到记录表

PsycExamRecord record = new PsycExamRecord();

record.setScaleId(co.getScaleId());

record.setUserId(userId);

//writeValueAsBytes将java对象转换成json字符串

record.setJudgeContent(objectMapper.writeValueAsBytes(result));

record.setRecordContent(objectMapper.writeValueAsBytes(co.getQuestionList()));

record.setUseTime(LocalTime.parse(co.getTime(), DateTimeFormatter.ofPattern("HH:mm:ss")));

recordMapper.insertRecord(record);

return result;

}

## 个人中心设计

### 前端设计

前端页面个人中心页面，分为三个切换卡进行展示，分别展示基本信息、答题记录、账号设置。基本信息页展示用户的一些信息，默认为不可编辑的状态，点击编辑按钮，昵称、性别、生日、手机号可以进行编辑，点击下方的保存保存成功。点击上方图片编辑可以更新头像。答题记录页展示用户答题的结果记录，可以点击查看详情。账号设置页可以更改用户密码，普通用户仅可以修改自己的密码，系统管理员可以修改其他普通用户的密码。效果如图4.13、4.14、4.15所示。

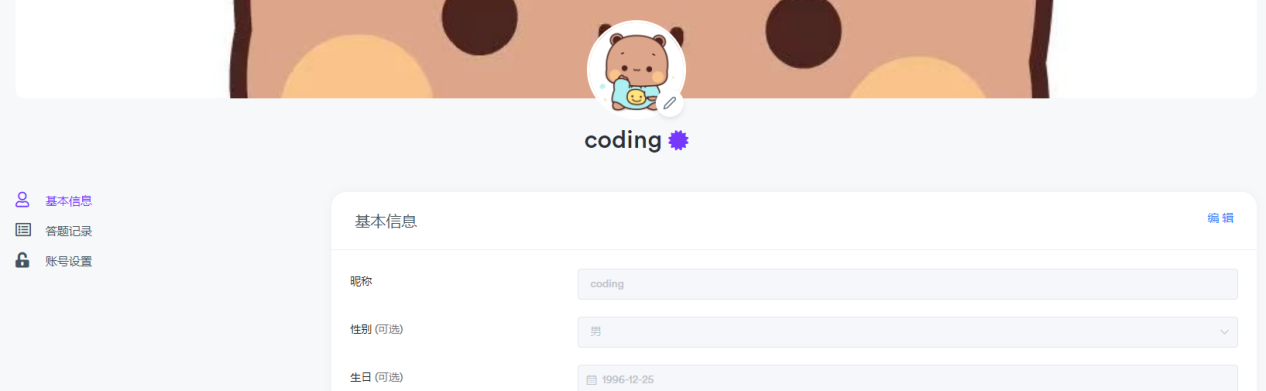


图4.13 个人中心基本信息页



图4.14 个人中心答题记录页

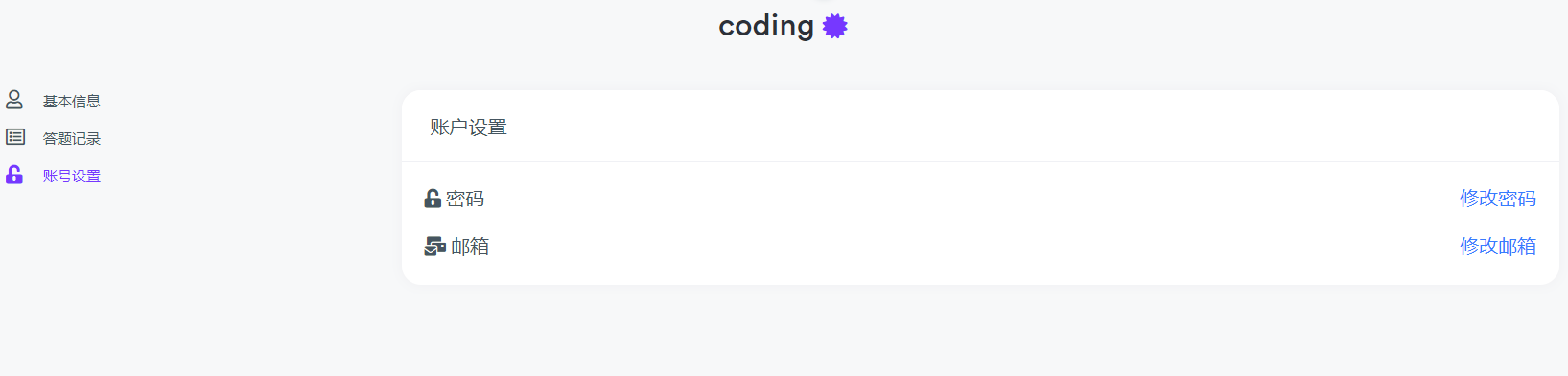


图4.15 个人中心账号设置页

### 后端设计

后端主要向前端提供三个接口，保存用户信息接口，分页查询用户答题记录接口，更改用户密码接口。

1. 保存用户信息接口：将用户信息更新到数据库，并且将更改的用户信息保存到Spring Security内存中。
2. 分页查询答题记录接口：从Spring Security 获取当前登录人的UserId，通过UserId作为条件查询答题记录返回给前端。
3. 更改用户密码接口：对非对称加密的新密码和旧密码进行解密，分别校验旧密码是否正确，新密码和旧密码不能相同，如果不通过校验则抛出对应异常，通过则对密码进行加密后更新到数据库中。流程如图4.16所示。



图4.16 更改密码流程图

# 系统测试

## 登录测试

**普通登录**：

期望结果：登录成功后跳转至重定向URI或主页，隐藏登录图标，并显示头像和昵称，该登录过期时间为半小时。登录失败提示用户名或密码错误。若再次登录则会跳转到最后操作的页面。如图5.1所示。

**“记住我”登录：**

期望结果：登录成功后跳转至重定向URI或主页，隐藏登录图标，并显示头像和昵称，登录过期时间14天。登录失败提示用户名或密码错误。如图5.2所示。

**QQ登录：**

期望结果：跳转至QQ登录，登录成功后跳转至重定向URI或主页，并显示QQ头像和昵称，登录过期时间为半小时。如图5.3所示。

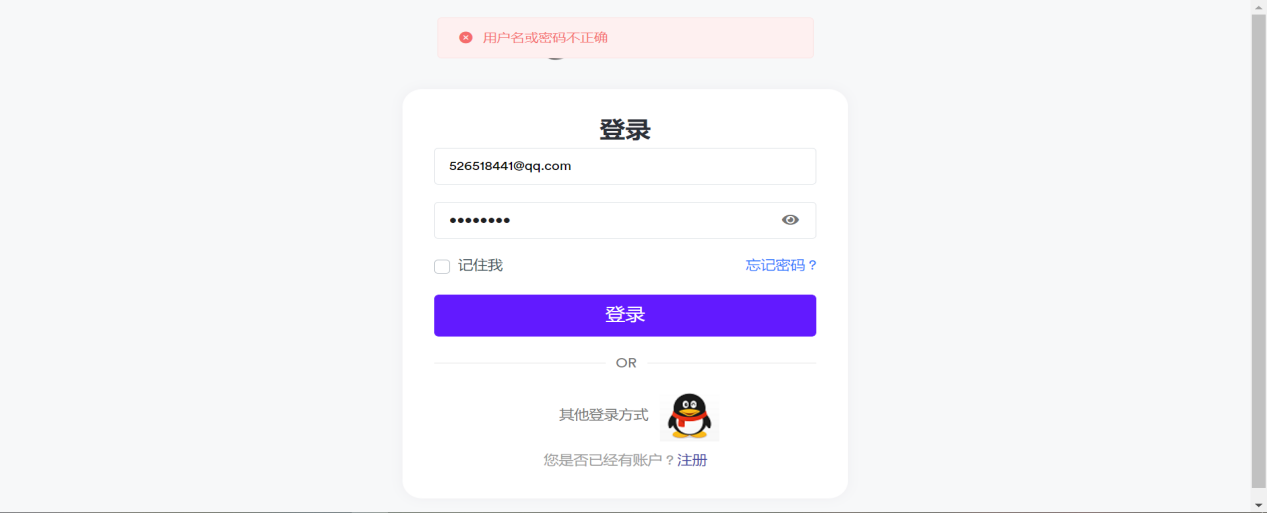


图5.1 普通登录失败显示



图5.2 记住我登录成功显示



图5.3 QQ登录授权信息显示

## 答题测试页测试

期望结果：点击测试管理标题进入文章详情页，头部显示当前所在路径，下方显示测试温馨提示语，提示语下方显示所答量表所属大类和量表名称，所属大类和量表名称两个控件为级联查询，只有先选择所属大类，量表名称才会有内容提示。量表名称下显示开始答题按钮。点击开始答题跳转至答题页面，页面左上角展示答题卡详情，答题卡上方展示所测量表名称，名称下展示用户昵称和答题计时时间，点击题目选项完成答题，答题卡随着答题情况实时变化，所有题目完成才能提交答案。测试结果图片5.4、5.5、5.6所示。

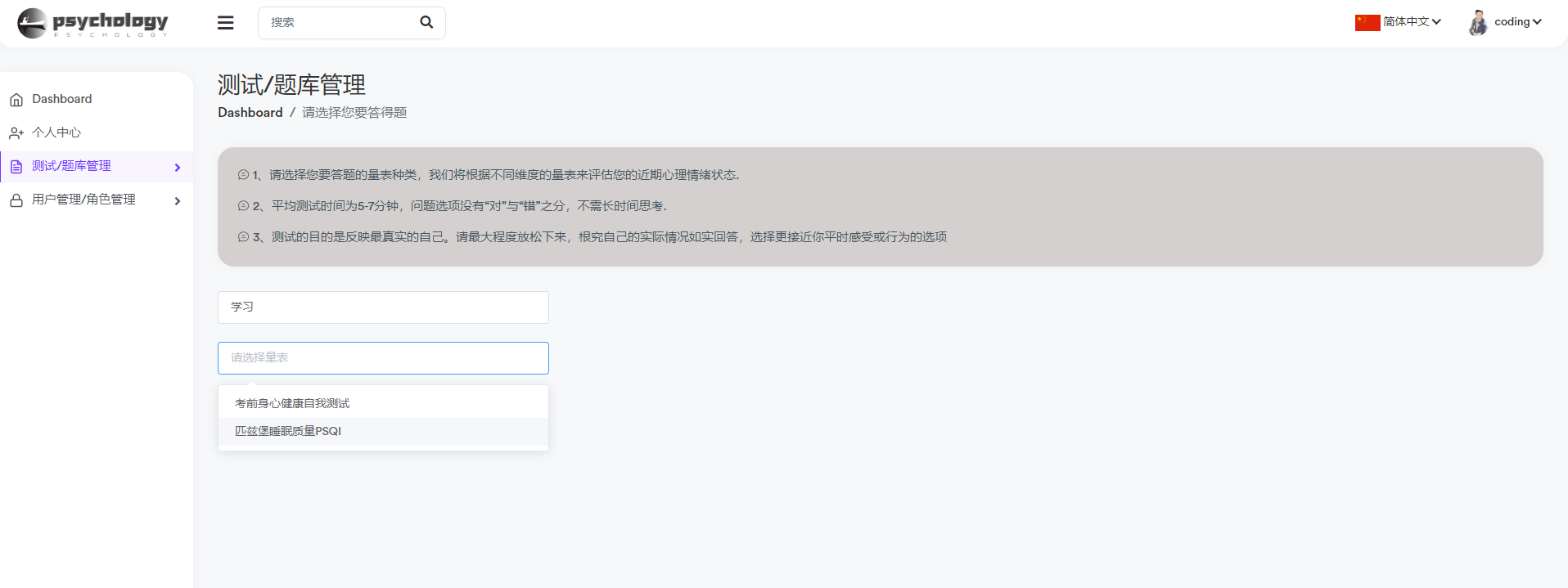


图5.4 答题测试页显示



图5.5 答题测试详情页显示

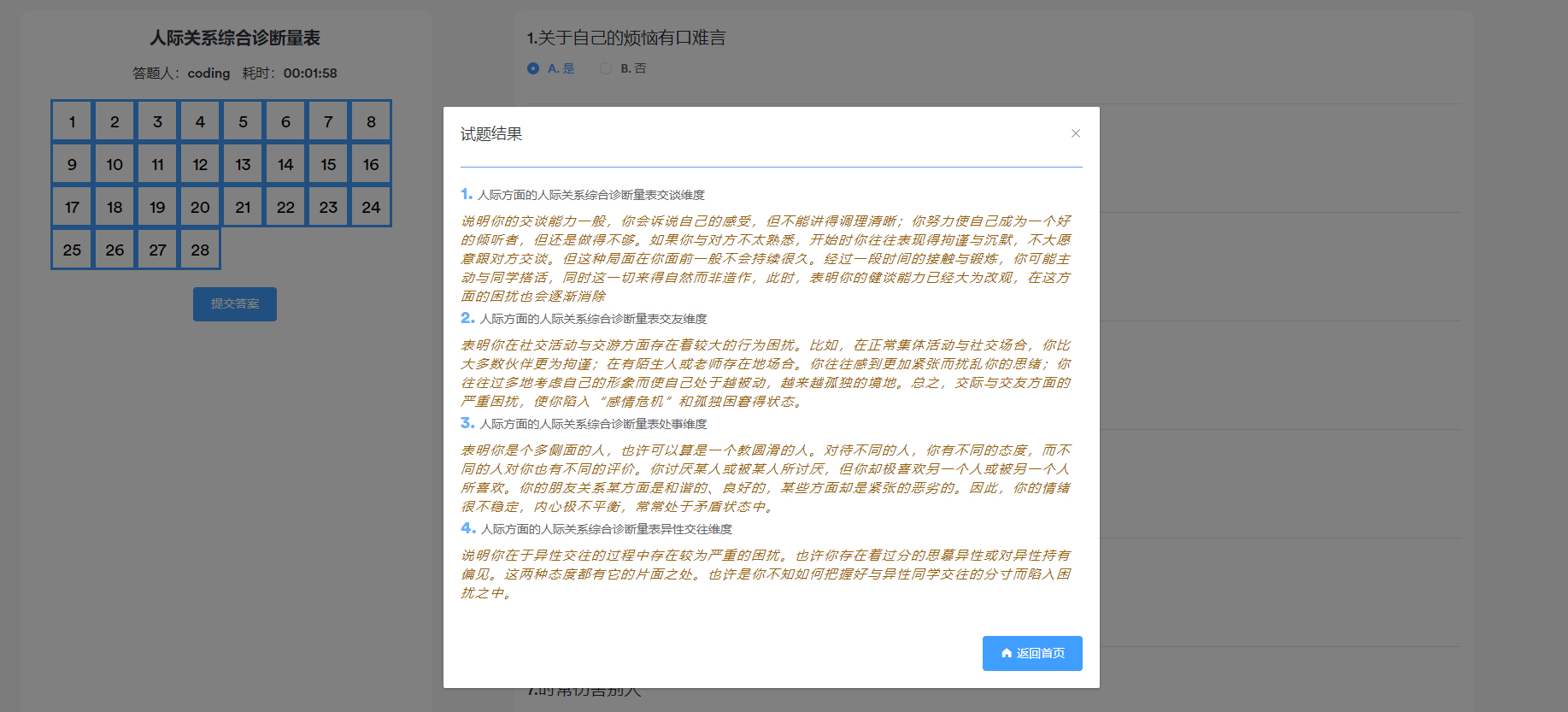


图5.6 测试结果详情页显示

## 量表管理页测试

期望结果：点击量表管理的新增按钮，跳转至量表新增页面，选择量表所属大类，输入量表名称，点击保存，弹出新增成功提示语。量表编辑和删除操作类似。测试结果如图5.7、5.8、5.9所示.



图5.7 量表新增成功显示



图5.8 量表新增失败显示

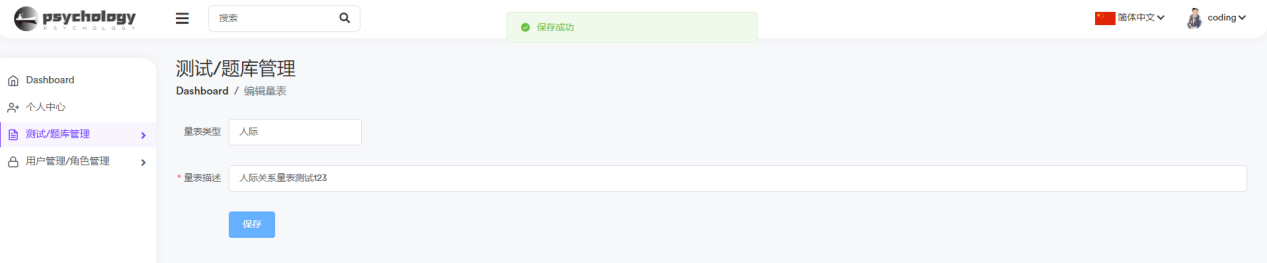


图5.9 量表编辑保存成功显示

## 个人中心页测试

期望结果：点击个人中心的基本信息编辑按钮，昵称，性别等个人基本信息由不可编辑状态转换成可编辑状态，弹出更新用户信息成功提示。测试结果如图5.6所示；在测试页面完成测试，点击答题记录按钮，数据随着测试记录实时更新，评判结果与测试页面展示结果相一致。测试结果如图5.10、5.11所示；点击修改密码，输入账户原始密码和新密码后点击提交按钮，前后两次密码不可相同。更新成功后会提示密码修改成功、Session已过期提示并跳转至登录页面，用户需要重新登录。测试结果如图5.12、5.13、5.14所示.

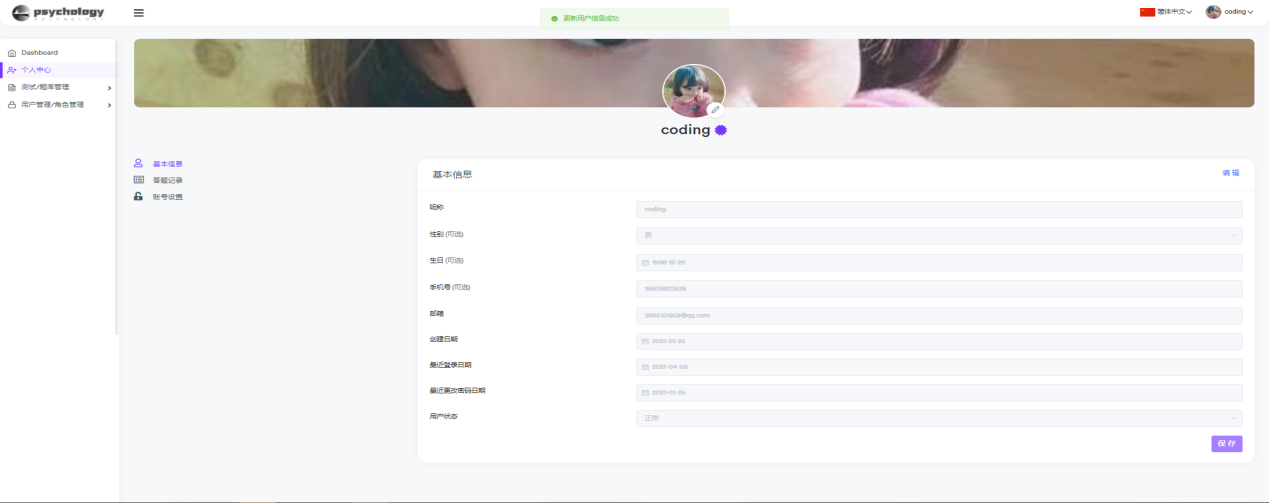


图5.10 用户信息更新成功显示



图5.11 答题测试评判结果显示



图5.12 答题记录详情页显示

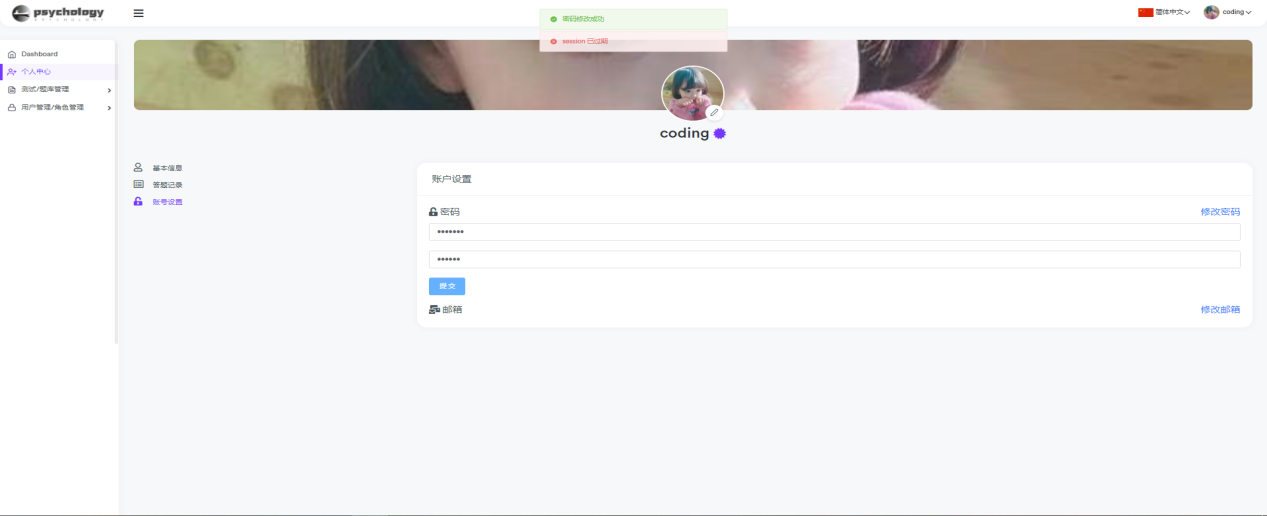


图5.13 账户密码修改成功显示

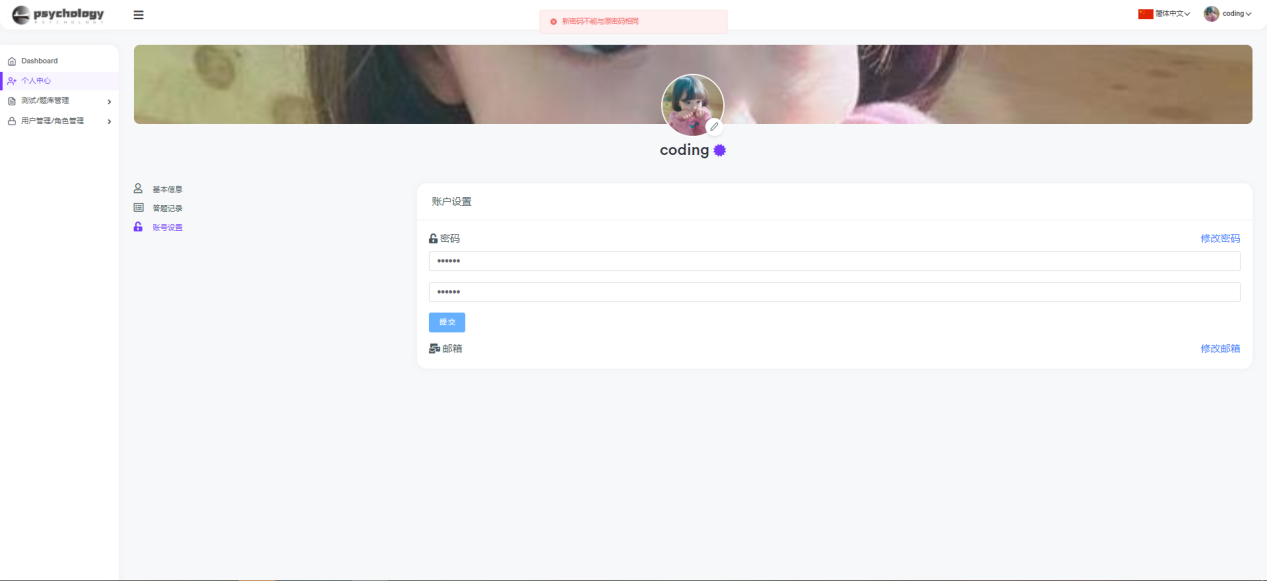


图5.14 账户密码修改失败显示

# 总 结

经过查阅资料分析，本课题是从大学生的心理健康角度出发的，对大学生心里现状进行详细分析，从开始选题到具体实现，总历时六个月。最终经过实际测试，证明系统达到了预期效果，能够正常使用。但客观的来讲，本系统所选用的心理测试量表计算口径单一，也就是这个系统的可扩展性并不是很完美，因为心理测试是一个很规范的问卷，计算方法，答题形式都需要经过严密分析才能敲定 理测试模块做精做深，做到细致入微，考虑到一些后续性的问题，实现细节功能，界面设计力求简洁干净，操作方便易使用，考虑到用户的真实体验。在做设计时，平常有了灵感就随时记录下来，等空闲下来查询资料看是否可以实现，怎么实现，需要用到哪些知识，有了需求才能有技术。这个系统是专门服务于心理测试方面的，在这个系统的使用下，心理健康测试变得高效速度，计分规则标准统一，分数统计真实，大大减少了学校心理健康教育方面的工作量。但尽管如此，这个系统仍然还有很多地方需要改善，对心理测试的评判需要更加客观真实，总体设计更加细致深入，再完善一些附带功能，仔细斟酌。总之，作为大学生使用的心理健康测试系统，还需要根据专业人士的理论成果与指导分析，采用更加先进的技术和真实的数据对系统进行完善和提高，根据社会形势的变化，与时俱进，开拓创新，建立信息化的心理健康教育及障碍预防平台，让大学生能随时自查，自测，让处在疑惑中的学生能及时得到帮助。

# 参考文献

[1]熊礼杭,罗湘明.大学生心理健康教育模式的构建[J].当代教育论坛：宏观教育研究,2007(1):59-60.

[2]刘耿冰.基于B/S结构的大学生心理测试系统的设计与实现[D].广东:广东工业大学,2016.

[3]董丹,汤玮丽. 浅谈网络技术在心理健康测试中的优势[J]. 现代企业教育,2008(6):122.

[4]成琳.设计大学生心理测试的研究[J].科技与创新,2020，（18）：54-55，62.

[5]张伟明.大学生心理健康测试及分析系统开发[D].陕西:西安理工大学,2018.

[6]马永康.个人博客体系框架的设计与实现[D].鞍山：辽宁科技大学,2019.

[7]杨开振．深入浅出Spring Boot2.x［M］．人民邮电出版社，2018.8（12）

[8]战晓苏，江凌 译.Java8官方变成教程[M].北京：清华大学出版社，2015.

[9]蒲龙,胡逸群.大学生心理健康状况与促进对策研究 ——以江苏省某高校为例[J].决策探索,2020,(20):39-40.

[10]武紫彤，田立强.当代大学生的心理问题研究[J].现代交际,2021（04）.

[11]丁雪丰 译．Spring Boot实战［M］．北京：人民邮电出版社，2016.9（8）

[12]陈漾.大学生心理健康测试系统的设计与实现[D].广东工业大学,2017.

[13]黄玉峤.某高校大学生心理教育及综合辅导管理信息系统的设计与实现[D].四川:西华大学,2018.

[14]杨雅彬.中小学在线心理健康测评系统的设计与开发[D].天津师范大学.2017.

[15]王博.地方高校青年教师心理测试系统设计与实现[J].电子测试,2014(19).

# 致 谢

异地求学四年，虽然路途辛辛，却收获满满。在这里最要感谢的是各位关照我的老师以及我的父母，你们做的每一件事都是为了让我学习知识，提高自己，是我学业路上的垫脚石，使我无后顾之忧。另外要感谢的是我的指导老师，授人以鱼不如授人以渔，您不止在设计上给了我锦囊妙药，更是我人生的指路明灯。在此诚挚感谢各位良师益友们的支持，在我遇到疑惑时积极给予帮助，为我指点迷津。最后我需要感谢以下我自己，感谢你没有放弃，今天一切努力都是为了明天更好的自己。

**个人简历**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 姓 名 |  | | 性 别 | |  | |  |
| 民 族 |  | | 政治面貌 | |  | |
| 籍 贯 |  | | | | | |
| 毕业学校 |  | | 学 历 | |  | |
| 专 业 |  | | | | | |
| 健康状况 |  | | 体重/身高 | |  | |
| 联系电话 |  | | 邮 箱 | |  | |
| 主修课程 |  | | | | | | |
| 通讯地址 |  | | | | | | |
| 教育情况 | | | | | | | |
| 时 间 | | 学校/实习单位 | | 曾任职务 | | 通过考试 | |
|  | |  | |  | |  | |
|  | |  | |  | |  | |
|  | |  | |  | |  | |
|  | |  | |  | |  | |
| 自我评价 | | | | | | | |
|  | | | | | | | |