

武汉理工大学

毕业设计（论文）

心理测评系统的设计与实现

学院（系）： 计算机科学与技术学院

专业班级： 软件工程 zy1501 班

学生姓名： 吕世豪

指导教师： 刘传文

学位论文原创性声明

本人郑重声明：所呈交的论文是本人在导师的指导下独立进行研究所取得的研究成果。除了文中特别加以标注引用的内容外，本论文不包括任何其他个人或集体已经发表或撰写的成果作品。本人完全意识到本声明的法律后果由本人承担。

作者签名:

年 月 日

学位论文版权使用授权书

本学位论文作者完全了解学校有关保障、使用学位论文的规定，同意学校保留并向有关学位论文管理部门或机构送交论文的复印件和电子版，允许论文被查阅和借阅。本人授权省级优秀学士论文评选机构将本学位论文的全部或部分内容编入有关数据进行检索，可以采用影印、缩印或扫描等复制手段保存和汇编本学位论文。

本学位论文属于 1、保密 ☐, 在 年解密后适用本授权书
2、不保密 ☐

(请在以上相应方框内打“√”)

作者签名: 年 月 日

导师签名: 年 月 日

摘要

心理评测是一种比较先进的测试方法，它是指通过一系列手段，将人的某些心理特征数量化，来衡量个体心理因素水平和个体心理差异差异的一种科学测量方法。按评测的内容、对象特点、表现形式、目的、时间、要求等分为若干种类。主要是各机关、企业、组织等用来选拔人才、安置岗位以及对一个人进行诊断、评价、辅助咨询的一种手段，心理测评包含能力测试、人格测试和兴趣测试等。

本论文实现了一个 B/S 架构的心理评测系统，基于 Python 的 flask 框架，React 框架和 MySQL 数据库，实现了用户的注册登录、账号激活、找回密码、管理员管理试卷、管理员管理用户、计算分数、分析答案得出结论的功能。

使用系统可以进行心理评测并得出结论，每份试卷针对性的让使用者了解自己对应指标的分数，更加了解自己对应方面的优点和缺点，同时也使管理员能够根据成绩进行筛选用户，了解大众的心理趋势。

关键词：心理评测，结果分析，档案管理，目标筛选

Abstract

Psychological assessment is a relatively advanced testing method. It refers to a scientific measurement method to measure the level of individual psychological factors and individual psychological differences by quantifying some psychological characteristics of human beings through a series of means. According to the content, object characteristics, manifestation, purpose, time and requirements of the evaluation, it can be divided into several categories. It is mainly a means used by various organs, enterprises and organizations to select talents, place posts, diagnose, evaluate and assist a person in counseling. Psychological assessment includes ability test, personality test and interest test. Generally speaking, psychological assessment has great significance in education, society and work. This system completes the basic function of psychometric assessment.

This paper implements a psychological evaluation system based on B/S architecture. Based on Python flask framework, React framework and MySQL database, the functions of user registration, account activation, password retrieval, administrator management test paper, administrator management user, calculating scores, analyzing answers and drawing conclusions are realized.

The system can be used to carry out psychological evaluation and draw conclusions. Each test paper aims to let users know their corresponding index scores, better understand the advantages and disadvantages of their corresponding aspects. At the same time, administrators can screen users according to their scores and understand the psychological trends of the public.

Key Words: Psychological assessment, Result analysis, File management, Target selection

目录

摘要	I
Abstract	II
第 1 章 绪论	1
1.1 心理测评的意义	1
1.1.1 心理评测在社会中的意义	1
1.1.2 心理测评在理论研究中的意义	1
1.1.3 心理测评在教育领域中的意义	1
1.2 国内外心理评测现状分析	2
第 2 章 需求分析	1
2.1 功能性需求	1
2.2 非功能性需求	2
2.3 系统目标	2
第 3 章 系统设计	3
3.1 需求设计	3
3.1.1 功能设计	3
3.1.2 流程设计	3
3.2 项目设计	4
3.2.1 前端项目架构	4
3.2.2 后端项目架构	5
3.3 项目重要依赖	7
3.3.1 React	7
3.3.2 Flask	7
3.3.3 Typescript 和 Javascript	8
3.3.4 前端构建工具——Webpack	8
3.3.5 Pandas	10
3.4 数据库设计	10
3.4.1 一对多关系	10
3.4.2 多对多关系	12
第 4 章 系统编码	14
4.1 前后端交互模型	14
4.1.1 RESTful API	14

4.1.2	使用 Redux 存储网络请求状态	14
4.2	账号系统	15
4.2.1	密码加密	15
4.2.2	邮箱验证	16
4.3	评测与数据分析	19
4.3.1	上传试卷	19
4.3.2	数据分析	20
4.3.3	计算每个问题最多回答的答案	21
第 5 章	系统运行	22
5.1	用户模块	22
5.2	管理中心	24
5.3	评测中心	24
第 6 章	结论与展望	26
6.1	系统完成	26
6.2	未来展望	26
参考文献		27
致谢		28

第 1 章 绪论

心理评测是一种比较先进的测试方法，它是指通过一系列手段，将人的某些心理特征线性化，衡量个体心理因素水平和个体心理差异的一种科学测量方法。按评测的内容、对象特点、表现形式、目的、时间、要求等分为若干种类。主要是各机关、企业、组织等用来选拔人才、安置岗位以及对一个人进行诊断、评价、辅助咨询的一种手段，心理测评包含能力测试、人格测试和兴趣测试等。

1.1 心理测评的意义

1.1.1 心理评测在社会中的意义

(1) 描述：心理评测可以从个体的智力、能力倾向、创造力、人格、心理健康等各方面对个体进行全面的描述，说明个体的心理特性和行为。

(2) 诊断：心理评测可以对同一个人的不同心理特征间的差异进行比较，从而确定其相对优势和不足，发现行为变化的原因，为决策提供信息。

(3) 预测：心理测评可以确定个体间的差异，并由此来预测不同的个体在将来的活动中可能出现的差别，或推测个体在某个领域未来成功的可能性。

(4) 评价：可以评价个体在学习或者能力上的差异，人格的特点以及相对长处和弱点，评价儿童达到的发展阶段等。

(5) 选拔：心理评测可以客观、全面、科学、量化地选拔人才。

(6) 安置：了解个体的能力、人格、心理健康等心理特征，从而为因材施教或人尽其才提供依据。

(7) 咨询：心理测评可以为学校的升学就业咨询提供参考，帮助学生了解自己的能力倾向和人格特征，确定最有可能成功的专业或者职业。

1.1.2 心理测评在理论研究中的意义

(1) 搜集资料：心理测评是搜集有关心理学资料的一个简单易行而又可靠的方法。

(2) 建立和检验假设：从心理测评的资料中可以发现问题，建立理论假设，并且可通过测量结果进一步来检验这些假设，进而推动心理科学的发展，例如智力结构理论的提出和发展，智力测验就起了重要作用。

1.1.3 心理测评在教育领域中的意义

(1) 心理测评是教学和学习的优良工具：通过心理评测教师和学生可以了解自己的心理状态，教师对学生的心里评测结果可以进行分析，改善自己的教学方法，学生能自己的心里评测报告改善自己的生活习惯、思考方式和学习方式。老师可以做到因材施教，学生可以找到自己适合的学习方式。

(2) 心理测评创新教育的重要手段：收集学生们的心里评测结果，可以对当前教育形式做出判断，从而改善教育结构和方针，修正未来教育的路线，改进整个社会教育的结构。

1.2 国内外心理测评现状分析

国外发达国家对心理健康问题的研究起步早，发展力度大，早已形成了比较健全和成熟的心理健康服务体制。以美国为例，高校心理辅导人员配比较为充足，并且从事心理咨询工作的人员呢必须达到由 APA 和 NASP 两个专业组织制定规定的专业水平才有资格从业。并且由于在较早的年龄阶段就开始关注公民的心理健康，所以公民可以把接受心理治疗与身体治疗同等看待，这都使得线上面对面式的心理综合测评较为容易实施。

但是我国对心理问题的关注，起步晚，发展慢，国家投入与学科支撑还不足够，也没有充足的相关从业人员，各地的心理健康服务机构也并不完善。需要进行心理测评的用户分布广散，很难进行集中的测试。更为重要的是，由于目前我国在此方面的宣传与指导缺乏，许多人不愿意接受直接的心理咨询与辅导，这就使得很难有机会进行线下的心理测评。基于这些现实情况，使用计算机线上评测能够减少地域上带来的不便和隐私问题，保护用户的隐私可以让用户在做评测时候更加的真实，让测量结果更加准确。

常用的心里评测需要进行多套经典量表才能全面评估个体心理状况，存在诸多的局限。当题量大的时候，根据不完整调查，15% 的受测者会出现疲劳、烦躁等不良情绪，进行 30 分钟的时候，24% 受测者会出现不良情绪，造成测评的效率低。多套量表的设定还会出现维度重复甚至产生不需要测评的维度的问题，造成测评资源的重复和浪费。这也会影响后期数据的录入和清理，浪费大量时间和人力成本。

第 2 章 需求分析

2.1 功能性需求

整体的系统需求如图 2.1 所示

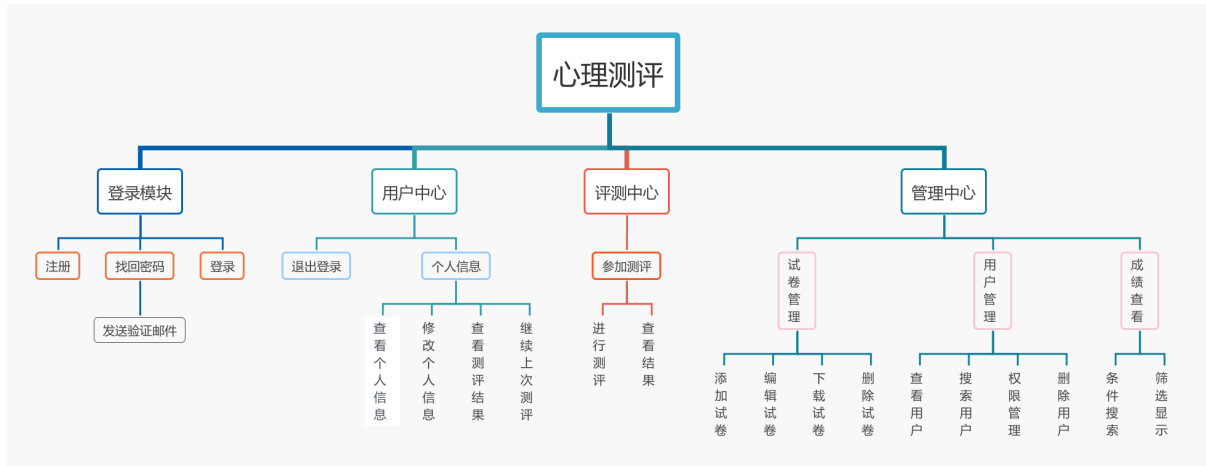


图 2.1 系统总体需求

整个系统分为四大模块：

- (1) 登陆模块：用户可以进行注册、登陆、邮箱激活、找回密码。
- (2) 用户中心：用户查看个人信息、查看自己所做试卷和未完成试卷、退出登陆。
- (3) 评测中心：用户进行评测和查看评测结果。
- (4) 管理模块：管理员对试卷进行上传、下载、编辑和删除，对用户进行添加、授权和删除，根据成绩筛选用户。

系统的用例图如图 2.2 所示

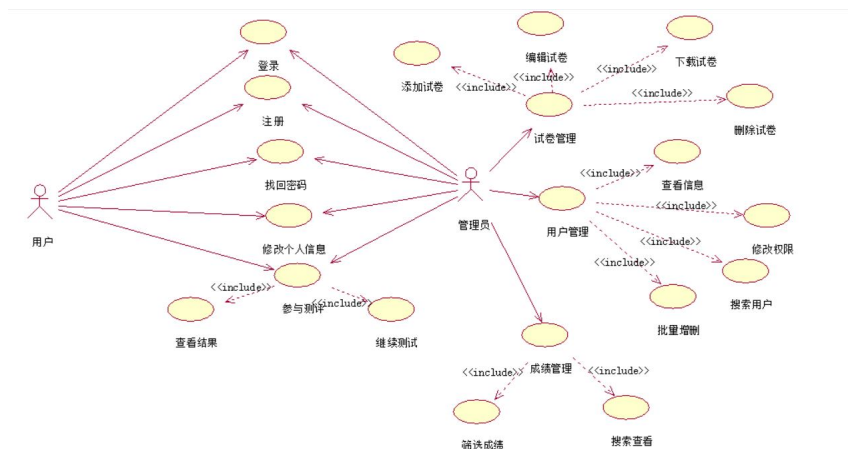


图 2.2 系统用例图

系统分为两种用户：

- (1) 普通用户：普通用户可以注册、登陆、找回密码、评测、查看评测信息。
- (2) 管理员：管理员有普通用户的所有权限，同时可以管理试卷、用户、成绩。

2.2 非功能性需求

- (1) 页面响应迅速，不会出现白屏等情况。
- (2) 对于难以表述的题目，可选择使用图片、音频等方式进行辅助。
- (3) 保护用户隐私，避免个人隐私信息的泄露。
- (4) 用户可在疲倦时暂停答题，系统将对已答部分进行保存，下次进入时可选择继续完成。
- (5) 整体设计友好温和，尽量不给受测者压力，保障测评结果尽量少受外界因素的干扰。

2.3 系统目标

目前网络上较为流行的各类网页式心理测试又存在不系统不完整、信息无法储存、无法管理、无法对比查看、试卷来源不可靠、风险性较高、测试结果不详细没有指导意义等问题。该类测试随机性较大，传播速度快，传播范围广，但是受测者经过一次测试后很难重复测试持续跟踪，也就是说测试结果很难真正的给予受测者实质性的评判依据和进一步的指导，仅仅是停留在非常表面的随手答题阶段。

基于以上客观现实和待解决问题，本次毕设希望做到，每个人可以根据我们系统的分析结果了解当前自己的心理状态，明确自己的目标，改进自己的不足，使得心理测评系统真正的产生实质性功效。

第 3 章 系统设计

3.1 需求设计

3.1.1 功能设计

心理评测系统由登陆模块、用户中心、评测中心和管理中心四大模块组成。登陆模块涉及用户的注册、登陆、邮箱验证、找回密码四个功能。用户中心可以修改个人信息、查看已经做的试卷的测评结果、查看未完成试卷的进度。评测中心可以做试卷进行评测和查看结果。管理中心是管理员对试卷、用户、成绩进行管理的模块。

3.1.2 流程设计

用户流程图如图 3.1 所示，主要流程为注册，激活邮箱后登陆然后进入评测中心进行评测，评测结果出来后查看分析结果。

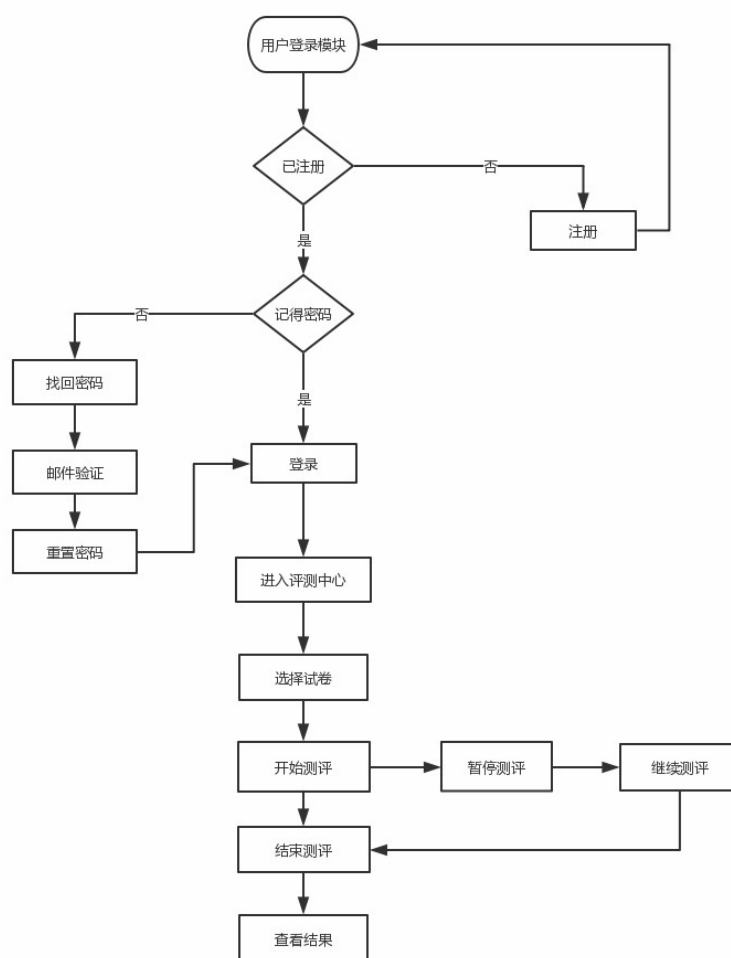


图 3.1 用户流程图

管理员流程图如图 3.2 所示，管理员有普通用户的所有流程，可以进行进入管理中心对用户、试卷、成绩进行管理。

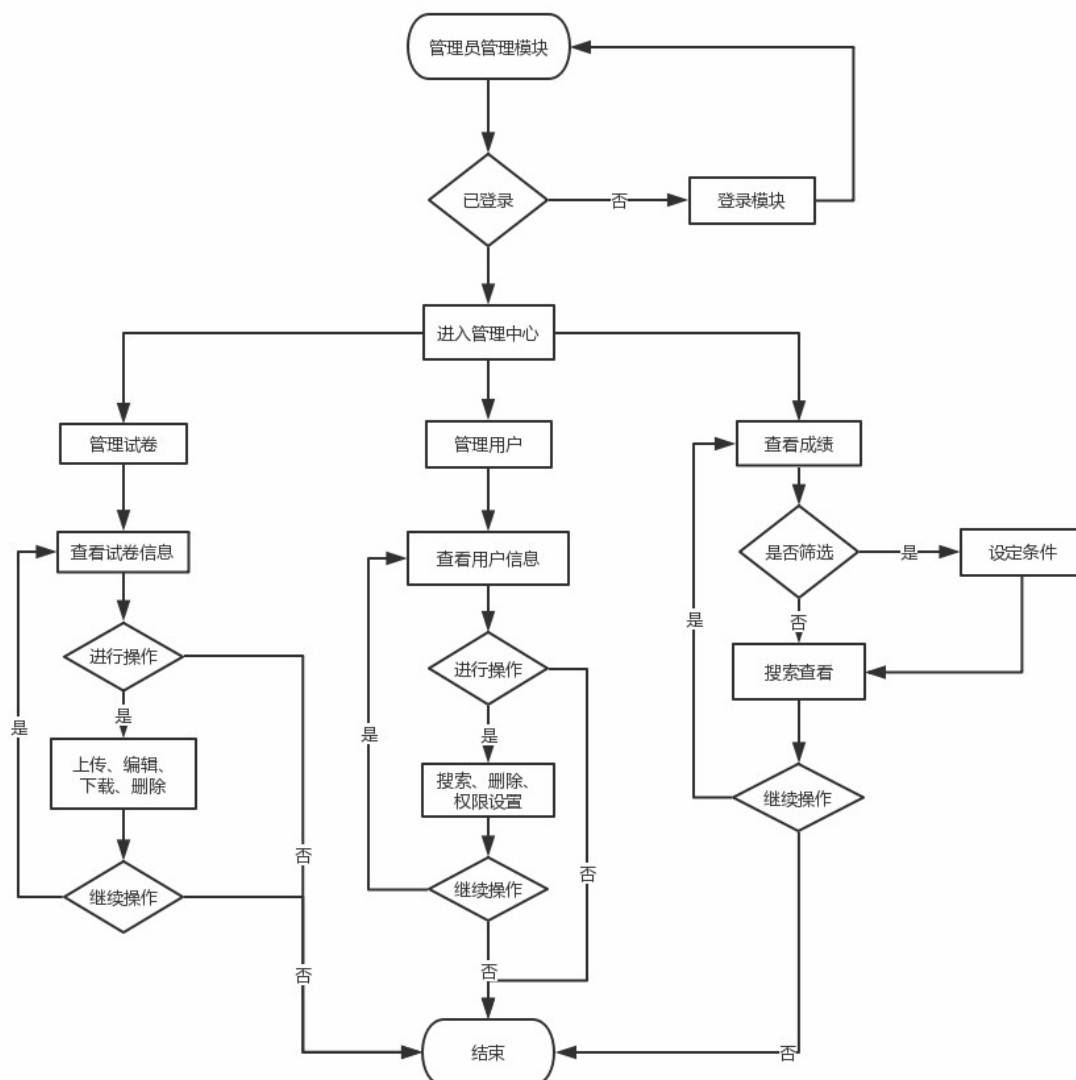


图 3.2 管理员流程图

3.2 项目设计

3.2.1 前端项目架构

前端项目架构如图 3.3 所示：

build 文件夹下存放着 webpack 的编译打包配置文件，分为两个环境，development 开发环境和 production 生产环境。

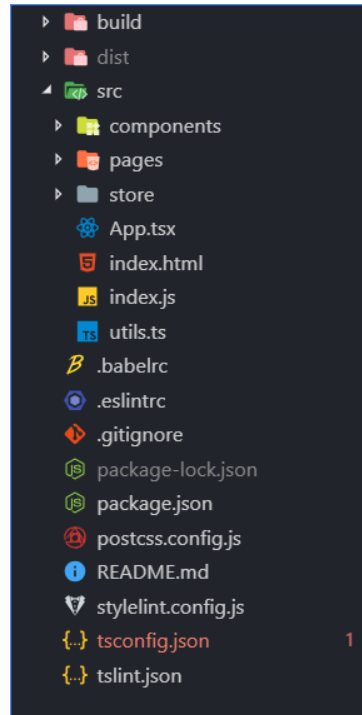


图 3.3 前端项目目录

dist 文件夹是存放着 webpack 打包后的文件，用作于 webpack-dev-server 的开发环境文件和上线后的生产环境文件。

src 文件夹是项目的根目录，里面的 components 文件夹存放着公用的组件，pages 存放着系统的各个页面文件，store 存放着项目所需要用到的 redux 状态仓库，App.tsx 是项目的主页面，index.html 是 webpack 打包注入 js 依赖的 html 文件，index.js 是项目的主入口，utils.js 是项目所需要的共用工具。

.babelrc 是 babel-loader 的配置文件，配置编译 js 的目标版本。

.eslintrc 是 eslint 代码检查的配置文件。

.gitignore 是 git 仓库 push 的时候忽略的文件。

package.json 是前端项目的架构文件，记载该前端项目所需要的依赖和依赖版本和该前端项目的基本信息。

postcss.config.js 是 postcss 的配置文件。

README.md 是描述该项目的 markdown 文件。

stylelint.config.js 是 stylelint css 代码检查的配置文件。

tsconfig.json 是 Typescript 的编译配置文件。

tslint.json 是 tslint 的配置文件。

3.2.2 后端项目架构

后端项目架构如图 3.4 所示：

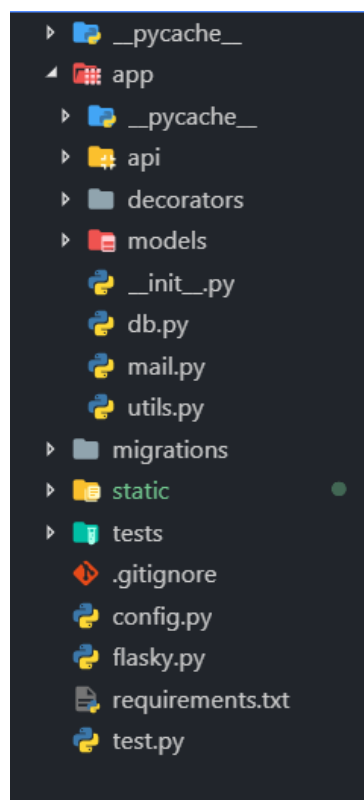


图 3.4 后端项目目录

__pycache__ 是 IDE 自动生成的 sourcemap 索引文件夹。

__init__.py 文件是该 Python 包的默认包导出文件。

app 文件夹是项目的根目录，api 文件夹存放着 Resful API 的文件，decorators 文件夹存放着项目中所用到的装饰器，models 文件夹存放着该项目所用到的所有对应数据库中的数据模型，db.py 是该项目连接数据库获得数据库实例的文件，mail.py 是该项目所使用到的邮箱模块，utils.py 是该项目的工具库。

migrations 是 Flask-Migrate 根据 Sqlalchemy ORM 数据模型定义进行数据库备份的文件夹。

static 是该后端项目的静态文件存放目录。

tests 是后端项目的测试文件存放目录。

.gitignore 是 git 仓库索要忽略文件的声明文件。

config.py 是该项目的配置文件。

flasky.py 是该项目的主入口文件，启动文件。

requirements.txt 记录着该项目的依赖和依赖版本。

3.3 项目重要依赖

3.3.1 React

React 是一个声明式，组件化的前端框架。其使用 Virtual DOM 技术把应用状态和 DOM 分离开来，搭配 Diff 算法来最小颗粒的更新 DOM，而这一切开发者都不需要关心，开发者只需要专注注意力在开发业务代码上。React 拥有极丰富的生态，本项目中还使用了 React-redux 来进行 redux 状态仓库的管理。

编写 React 组件简略代码如下：

```
class App extends Component {
  public componentDidMount() {
    ...
  }

  public render() {
    return (
      <div> this is App
        <a onClick={this.onClick}>
      </div>
    )
  }

  private onClick = (e: ClickEvent) => {
    e.preventDefault();
    ...
  }
}
```

所有 React 组件都要继承 React.Component，拥有 React 组件的生命周期，每个 React 组件需要实现一个继承父类的 render 方法，来控制组件如何进行渲染。

3.3.2 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。本项目中使用了 Flask-Migrate 来进行数据库的备份，Flask-Mail 来进行邮件的发送，Flask-SqlAlchemy 来进行更加简单的操作 SQLAlchemy。

Flask 定义视图路代码如下：

```
@app.route("/index")
def Index():
    return "<h1>Hello ,□Index</h1>"
```

传给 `app.route` 的是一个路径字符串，该方法返回一个装饰器，装饰器装饰路由函数，每个路由函数返回一个页面或者数据。

3.3.3 Typescript 和 Javascript

项目使用的前端语言并非常规的 Javascript 而是它的超集——Typescript，其作为超集在绝对兼容 Javascript 代码的前提下，加入了许多静态语言所拥有的特性，比如接口、泛型、类型判断等等。Javascript 作为脚本语言其灵活性和方便性为大家所称道，但是也由于其灵活性，前端代码的质量低下导致项目难以维护和扩展，Typescript 就是解决 Javascript 严谨性而诞生的一门语言。Java 作为世界上用途最广泛，使用人数最多的语言就是因为其严谨性得到了很多企业的偏爱，Typescript 看起来 80% 都和 Java 很相似，可以理解为在 Javascript 的基础上加了一层代码检查和提示。

3.3.4 前端构建工具——Webpack

浏览器的版本众多，浏览器与浏览器之间的内核有许多差异，这导致同一份代码可能在不同浏览器下的表现不同，通过人工的方式编写兼容代码会有很高的人力成本和时间成本。同时在以前的开发过程中，前端依赖后端导致前端开发与后端严重耦合，降低了开发效率。前端工程化时代的到来，让前端拥有自己的项目构建能力，同时还解决了不同浏览器之间的兼容问题，诞生了许多前端构件工具，Gul、Grunt、Webpack 等等。其中 Webpack 是当下活跃度最高，最火热的前端构建工具。

Webpack 最主要的两个功能是编译和打包。本项目所使用的 React 是使用 JSX 语法，不是浏览器默认可支持的语法，所以要将我们的前端代码编译成浏览器可识别的代码。Webpack 最重要的两个概念就是 loader 和 plugin，loader 负责各种类型文件（.css, .js, .ts, .png.....）的编译工作，可以将他们转化成目标浏览器的可识别文件，plugin 则负责除了编译以外的所有工作，包括打包，体积优化，分离依赖等等。

Webpack 编译 jsx 和 tsx 文件成 es5 的配置代码如下：

```
{
  test: /\.jsx?$/,
  exclude: /node_modules/,
  use: ["babel-loader"]
},
{
  test: /\.tsx?$/,
```



```

exclude: /node_modules/,
use: [{
  loader: "ts-loader",
  options: {
    transpileOnly: true,
    getCustomTransformers: () => ({
      before: [
        require("ts-import-plugin")({
          libraryName: "antd",
          libraryDirectory: "es",
          style: "css"
        })
      ]
    }),
    compilerOptions: { module: "es2015" }
  }
}]
}

```

Webpack 还提供了前端可实时预览的热更新服务器——webpack-dev-server, 该插件可以实时的检测项目文件的变化并重新编译和打包让浏览器的页面刷新。

Webpack 打包体积优化也是一个深奥的学问。

Webpack 在原始配置的情况下，因为是单页面应用，所以打包出来以后的 bundle 文件非常之大，高达 4MB，这对于首屏加载是非常不利的，过长的白屏会流失很多的用户量，所以 Webpack 的优化也是一门学问。

常用的优化有以下几点：

(1) 配置 optimization 中的 splitChunks 选项，会根据依赖图对于多次依赖的包单独打包然后进行引用。

(2) 配置 externals 选项对于第三库在 html 文件中进行 CDN 引用，打包时候忽略该库。

(3) 使用 uglifyjsPlugin 对打包文件进行压缩。

(4) 使用懒加载进行加载组件和页面，react-loadable 可以通过一个高阶组件包裹我们的异步组件使用 import 函数进行异步的加载组件，当加载成功的时候显示组件否则就加载 Loading 组件。

(5) 配置 Gzip。

进行上面的优化以后，打包以后的文件按照路由和组件分成了十几个文件，最大的体积不超过 400KB，减少了首屏加载的时间。

3.3.5 Pandas

Pandas 是一个数据科学分析第三方库，可以方便的建立矩阵以及对矩阵中的数据进行各种数学运算和变换，在解析试卷和成绩分析的时候有很多使用。同时可以对矩阵中的数据进行清理和类型的转换。

3.4 数据库设计

通过 SQLAlchemy 我们可以方便的在数据库中创建一个表并生成记录，同时也可以方便的进行数据库操作，定义数据之间的关系，搭配 Flask-Migrate 还可以方便的进行数据库备份、恢复和回滚。

这个系统的主要有三个表，用户表 (users)，试卷表 (papers)，成绩表 (grades)，数据库实体之间的关系，箭头都代表一对多关系，粗体字体为必填字段，数据库关系如图 3.5 所示。

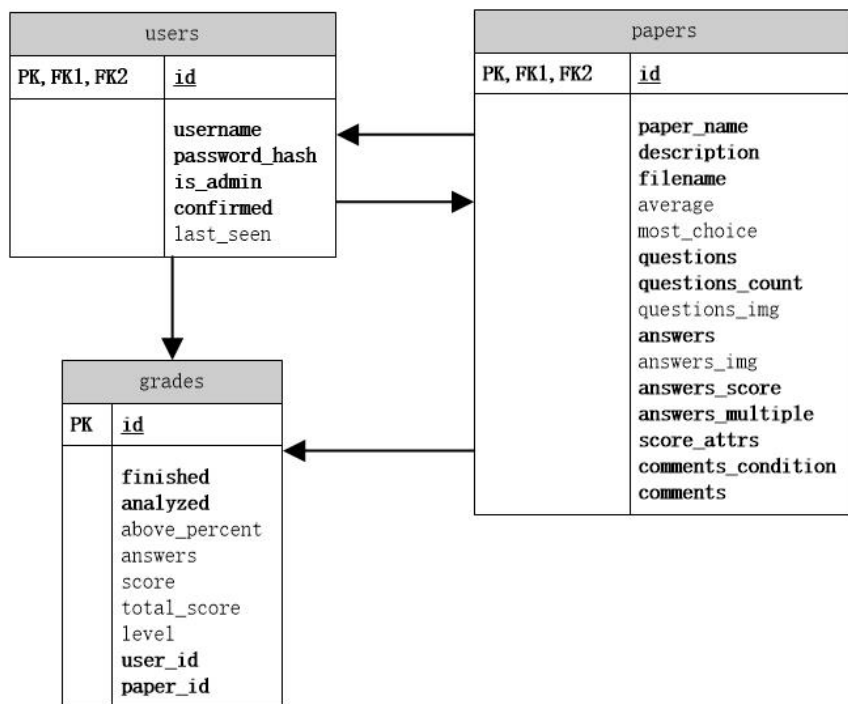


图 3.5 数据库关系

3.4.1 一对多关系

如图中，用户表和成绩表，试卷表和成绩表之间都为一对多关系。SqlAlchemy 中可以方便的通过 relationship 和 foreignKey 进行设置一对多关系。

建立用户和成绩，试卷与成绩之间的一对多关系代码如下：

```
class User(Model):
```

```

...
grades = relationship("Grade", backref="user", lazy="dynamic",
cascade="all", delete-orphan=True, passive_deletes=True)

class Paper(Model):
    ...
    grades = relationship("Grade", backref="paper", lazy="dynamic",
cascade="all", delete-orphan=True, passive_deletes=True)

class Grade(Model):
    ...
    user_id = Column(Integer, ForeignKey("users.id",
ondelete="CASCADE"))
    paper_id = Column(Integer, ForeignKey("papers.id",
ondelete="CASCADE"))

```

relationship 的这一端代表着 1 的这一端，backref 传入的是在 Grade 对象下如何访问 User 或者 Paper 对象，lazy 代表如何访问这个对象，一对多之间的关系通过外键的形式进行设置，比如 users.id 代表着是 users 这张表中的 id 作为外键。同时外键的字段设置需要和做外键的字段类型一模一样，否则会报错。

lazy 选项如下：

(1) select，默认选项，直接查找出所有对象，例如 user.grades 直接通过列表列出所有该用户下的 grade

(2) dynamic 返回的是一个查询集，例如 users.grades.filter_by(...) 返回的是一个查询集可以进行筛选操作。

(3) joined 类似于 select，但是 selectc 在查询的过程中需要查询两个表，而 joined 是建立一个新的连结表进行查询，在数额大的情况下使用 joined 能节省不少性能。

SqlAlchemy 作为最强大的 ORM 工具当然不只是定义关系，还有就是定义实体不存在的时候如何处理与其相关的数据，加入一个用户被删除了，那么于其关联的成绩也应该要全部删除，而不是作为冗余数据存在数据库中，一套试卷被删除了同理。那么 cascade 就是干这个事情的。

cascade 选项如下：

(1) save-update，默认选项，添加一条数据的时候会把其他和它相关的数据都添加到数据库中。

(2) merge，默认选项，合并一个对象的时候会将使用了 relationship 相关联的对象也进行 merge 操作。

(3) expunge，移除操作的时候，会将相关联的对象也进行移除，这个操作只是从

session 中删除，而不是从真正的数据库中删除。

(4) all，包含上面所有的选项。

(5) delete-orphan，删除相关的孤儿数据。

对应的要在多的那一端设置 ondelete 属性。

至此一对多的关系就设置好了，使用 Flask-Migrate 升级数据库的时候可以自动检测相关外键的设置然后更新数据库。

3.4.2 多对多关系

数据库多对多关系如图 3.6 所示

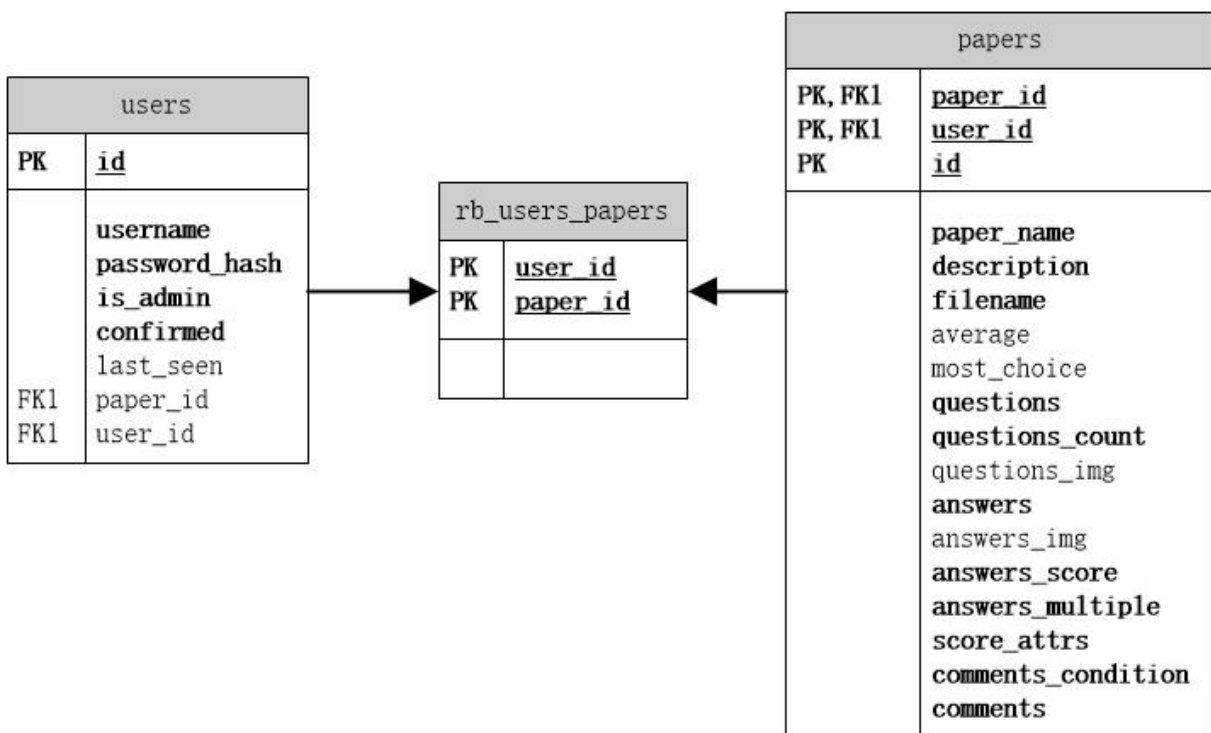


图 3.6 多对多关系

一个用户可以做多张试卷，一张试卷也可以有很多用户来做。这就是一个多对多关系。多对多关系中，需要一个额外的连接表来做查询。

多对多简略代码如下：

```
class User(Model):
    ...
    papers = relationship("Paper", secondary=rb_users_papers,
        lazy="dynamic")

class Paper(Model):
```

```
...
users = relationship("User", secondary=rb_users_papers,
lazy="dynamic")

// 连接表
rb_users_papers = Table(
    "rb_users_papers",
    Column("user_id", Integer, ForeignKey("users.id"),
primary_key=True),
    Column("paper_id", Integer, ForeignKey("papers.id"),
primary_key=True)
)
```

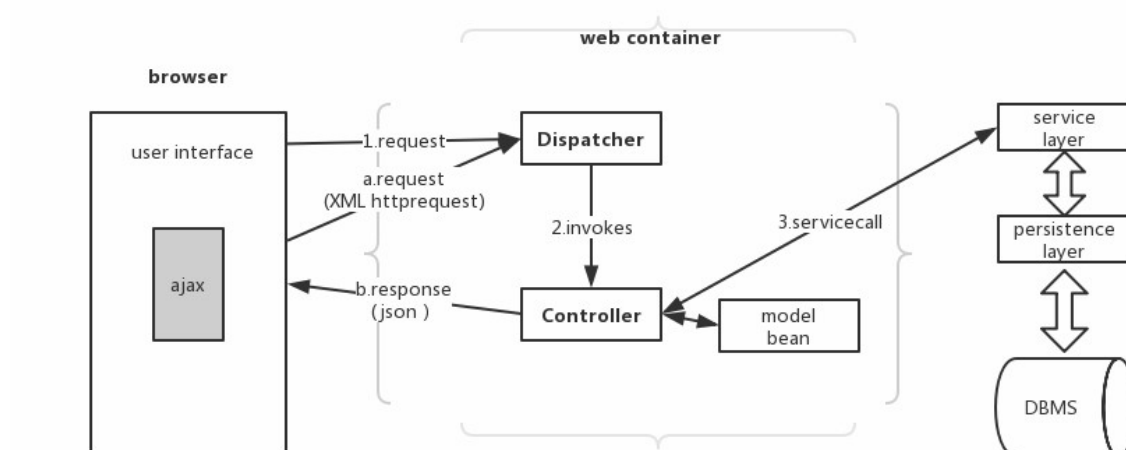
上面的代码说明了 users 表和 papers 表之间是通过一个 rb_users_papers 进行连接查询的，同时连接表中的每个字段都需要与原字段的属性相同。在原表中的 relationship 中需要设置 secondary 属性来进行使用连接表的查询。使用连接表进行连接的两个表，任何一条记录被删除以后相关的连接表记录也会被删除，当然也可以设置与其关联的记录也被删除，但在此处的情况不适合这样做，试想一下，如果一个用户记录被删除了，那么该用户做过所有的试卷记录也会被删除，别人就无法做了，这样是非常的不合适的情况。

数据库的设计是一个完善的软件系统必备也是最重要的一个阶段，一个好的数据库设计能让一个系统的性能良好、易于维护、易于拓展。有些人可能会觉得直接手写 sql 更加灵活和方便，但是如果作为一个多人维护开发的系统来说，每个人都有自己写 sql 的习惯，会让项目代码水平层次不齐难于维护，这时候就要使用 SQLAlchemy 这种工具了，通过一种 OOP 的方式来进行数据库的管理。

第 4 章 系统编码

4.1 前后端交互模型

前后端交互模型如图 4.1 所示



4.1 B/S 架构图

4.1.1 RESTful API

REST 全称是 Representational State Transfer，中文意思是表述（编者注：通常译为表征）性状态转移。

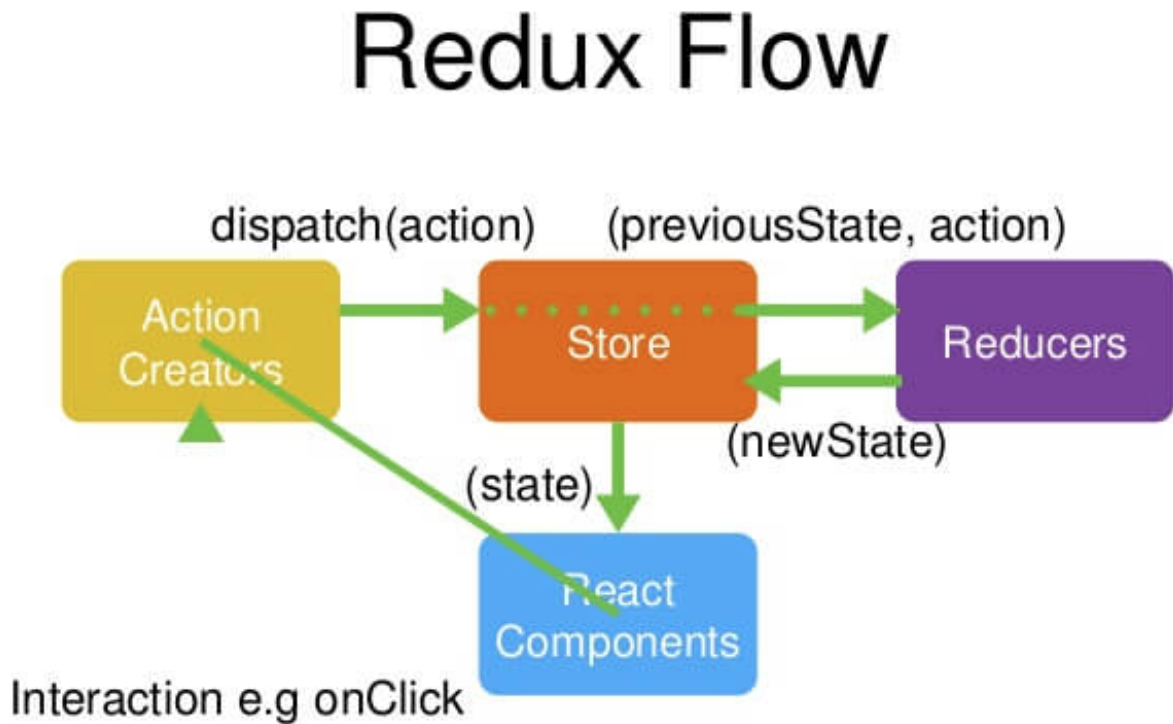
RESTful API 以下的特征：

- (1) 每一个 uri 代表一种资源。
- (2) 客户端和服务端之间，传递这种资源的某种表现层。
- (3) 客户端通过四个 HTTP 动词 (GET、POST、DELETE、PUT)，对服务器端资源进行操作，实现“表现层状态转化”(增删改查)。
- (4) URL 中通常不出现动词，只有名词。
- (5) 使用 JSON 不使用 XML。

4.1.2 使用 Redux 存储网络请求状态

Redux 是 JavaScript 状态容器，提供可预测化的状态管理。可以让你构建一致化的应用，运行于不同的环境（客户端、服务器、原生应用），并且易于测试。其使用 Immutable 数据结合 React 能让页面相应数据进行最小颗粒度的更新 DOM 结点。

redux 数据流如图 4.2 所示



4.2 redux 数据流

每一次发起 http 请求获得响应的时候触发一个 Redux Action，将服务端返回的数据存储在 store 中。这样做的好处是所有的需要使用服务端数据的 React 组件不需要影响父子组件之间的数据流直接提取公用的状态。

4.2 账号系统

4.2.1 密码加密

用户密码的创建和验证简略代码如下：

```

@property
def password(self):
    raise TypeError("cannot read a user's password")

@password.setter
def password(self, password):
    self.password_hash = generate_password_hash(password)

def verify_password(self, password):

```

```

if not password:
return False
return check_password_hash(self.password_hash, password)

```

@property 装饰器装饰一个属性的 getter，修饰的函数名为属性名，返回的值是该属性的 getter，此处该属性不能直接访问访问则会抛出异常。

@property 同时会全局创建一个 @password.setter 的装饰器装饰属性的 setter，在设置 password 的时候我们其实设置的是 password_hash 这个属性，值为 hash 加密过的 password。

处理登陆请求的时候调用 verify_password 函数判断密码是否正确。

这样做的好处就是即使是数据库管理员也不可能知道用户的密码，只能看到加密后的字符串，同时 md5 是不可逆的加密方式，不能由加密后的字符串反推出明文，保证了数据的机密性。

4.2.2 邮箱验证

用户注册和忘记密码的时候需要使用邮箱验证。在用户注册成功后会创建一个用户独有的 token，将 token 作为链接参数发送给用户邮箱中，在用户点击链接进入验证页面的时候会将 token 返回后端后端进行验证。

验证失败有两种结果，一是 token 过期，二是 token 中的用户信息不匹配当前用户。通常不匹配的情况有三种：

- (1) 用户自己改了 url 中的 token
- (2) 中间商劫持被修改了 token 信息。
- (3) 非法重定向。

无论哪种都是非法篡改 token 的行为，影响用户信息安全。

邮箱激活流程图如图 4.3 所示，用户注册以后，系统会向注册时候填写的邮箱中发送一个邮件，里面带有验证地址的连接，用户点击以后进入验证页面。有两种用户，一种是自主创建的用户，一种是管理员批量添加的用户，第二种用户在登陆的时候需要填写邮箱，然后再进行激活。如果用户篡改连接地址或者被运营商劫持验证时不会通过。因为连接中有一个参数为 token，token 是将用户信息和时间戳 hash 后的可逆加密字符串，一旦服务端验证 token 的时候信息和连接中的 username 参数对不上那就会返回失败。

找回密码流程图如图 4.4 所示，用户忘记密码后点击忘记密码，需要输入自己的用户名，随后系统会向用户名注册时候对应的邮箱发送验证邮箱，点击邮箱中的连接进入验证界面如果验证通过则会出现新密码输入框。

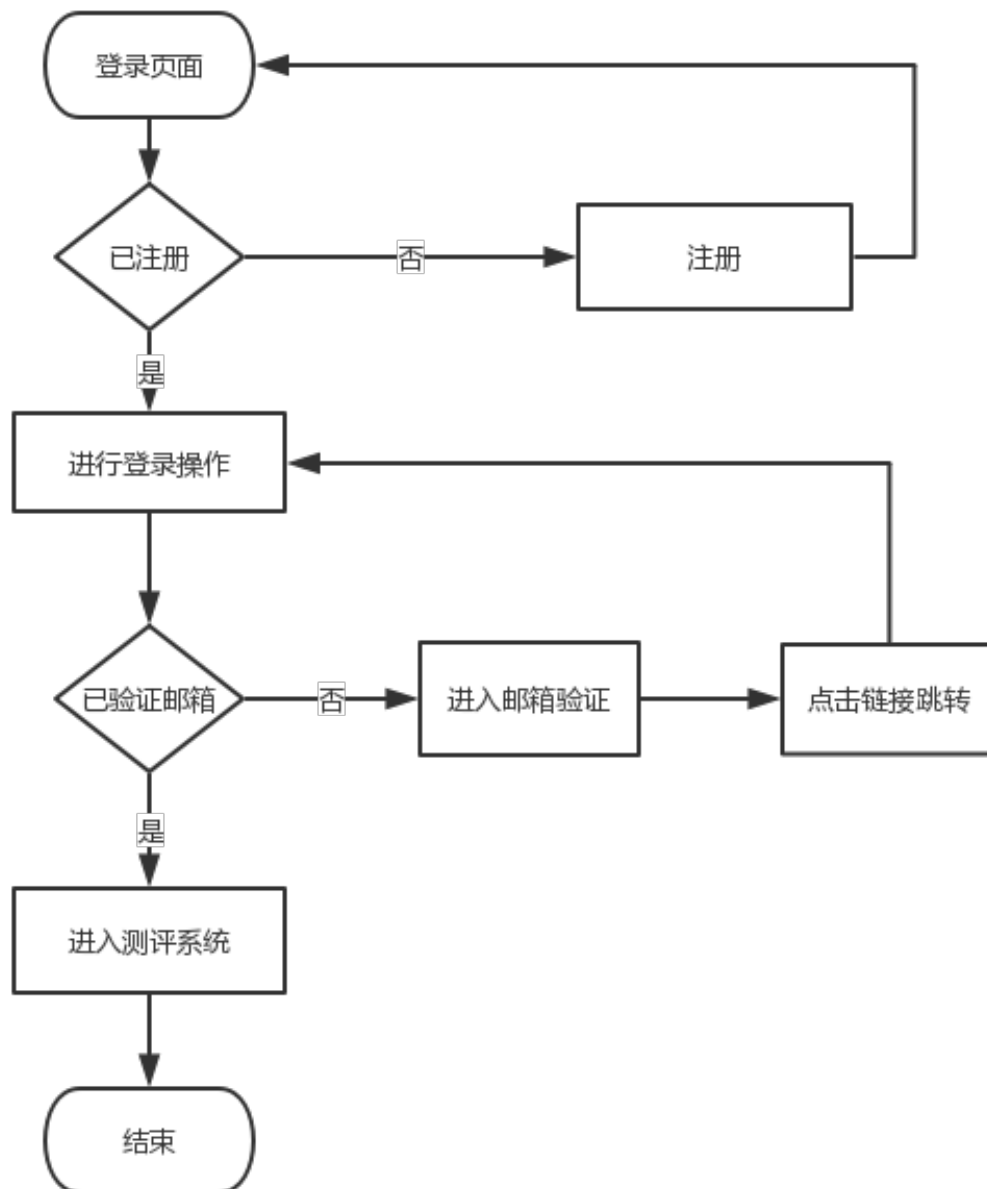


图 4.3 注册邮箱激活

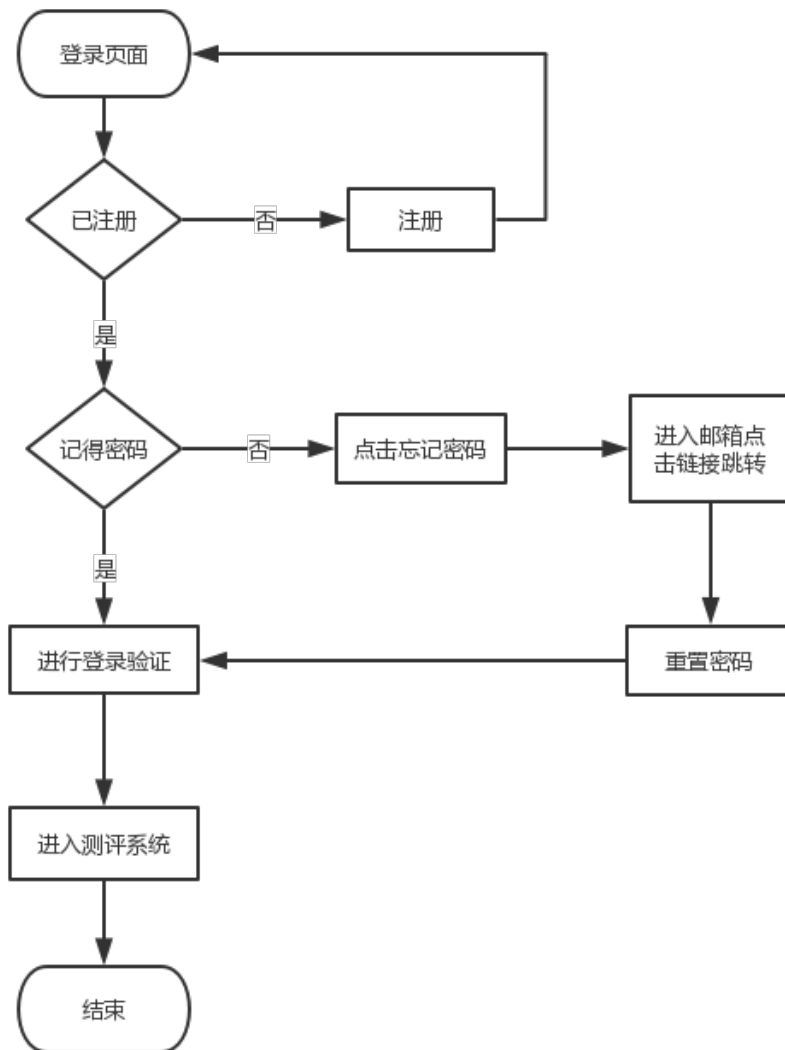


图 4.4 忘记密码邮箱验证

19

type 试卷的类型，psychology 代表室心理评测，grade 代表普通试卷，决定了试卷的算分机制

description 试卷的描述

上传文件后后端通过读取 excel 表格文件将数据插入到数据库中。

每个问题之间的数据使用”@” 来进行连接。

在评测时候后端会对表数据进行解析和构建对象发送给前端。

试卷数据结构如图 4.7 所示

```

answers: []
average: null
comment: ""
description: "这是描述"
finished_count: 0
most_choice: null
paper_name: "测试"
▼ questions: [{answers: ["甜酥糖", "牛奶糖", "跳跳糖"],...}]
  ▼ 0: {answers: ["甜酥糖", "牛奶糖", "跳跳糖"],...}
    ▶ answers: ["甜酥糖", "牛奶糖", "跳跳糖"]
    ▶ answers_img: ["https://ss2.bdstatic.com/70cFvnSh_Q1YnxGkpoWK1HF6hhy/it/u=2032379950,3484882496&fm="
      multiple: false
      question: "说到喜糖, 你觉得下面哪种糖最适合? "
      question_img: ""
    ▶ 1: {answers: ["不确定", "会", "不会"], answers_img: ["", "",...], multiple: false,...}
    ▶ 2: {answers: ["很喜欢", "不喜欢", "一般"],...}
    ▶ 3: {answers: ["希望", "不希望", "无所谓"], answers_img: ["", "", ""], multiple: false,...}
  questions_count: 4
  score: ""
  score_above: ""
  score_attrs: "责任心@爱心@激情@努力"
status: 0
    
```

图 4.7 试卷数据结构

其中 average、comment、most_choices、answers、score、score_above、score_attrs 是完成试卷以后才会有的参数，与数据分析结果的接口是一个接口。

4.3.2 数据分析

成绩数据表如图 4.8 所示

```
mysql> select * from grades;
```

id	finished	analyzed	above_percent	answers	score	total_score	level	user_id	paper_id
1	1	1	0.0000000000	1020102	0000008	8	4	1	1

图 4.8 成绩数据表

用户回答完后，答案之间以”@” 分隔开来存储。在计算得分的时候遍历答案进行指标的加减。

4.3.3 计算每个问题最多回答的答案

计算最多回答的答案的简略代码：

```
most_choice = "@".join(map(lambda x: Counter(x)
                             .most_common(1)[0][0], choices))
```

Counter 是 Python 内置的一个计数器类,可以得到每个元素出现的次数,其中`.most_common`函数可以得到列表中出现频率最频繁的元素。

第 5 章 系统运行

5.1 用户模块

登陆模块包括用户的注册、登陆、找回密码、邮箱验证等功能，登陆界面的记住我可以设置浏览器中的 cookie 的过期时间为 7 天，登陆如图 5.1 所示，注册如图 5.2 所示，找回密码如图 5.3 所示，个人中心如图 5.4 所示。



图 5.1 登陆界面



图 5.2 注册界面



图 5.3 找回密码



图 5.4 个人中心

5.2 管理中心

管理中心是管理员对试卷、用户、成绩进行筛选的模块，管理试卷如图 5.5 所示，管理用户如图 5.6 所示，可以更改用户的权限、删除用户、上传文件批量增加用户，筛选成绩如图 5.7 所示。

共有 2 张试卷

+ 添加

全部删除

<input type="checkbox"/>	试卷名称	试卷描述	操作
<input type="checkbox"/>	测试	这是描述	<div>编辑</div> <div>下载</div> <div>删除</div>
<input type="checkbox"/>	出现在	这是描述	<div>编辑</div> <div>下载</div> <div>删除</div>

图 5.5 管理试卷

共有 9 位用户

+ 添加

删除全部

请输入用户名或者邮箱

<input type="checkbox"/>	用户名	电子邮箱	权限	上次登录	操作
<input type="checkbox"/>	15007272719		管理员	Sat, 25 May 2019 18:42:42 GMT	<div>取消管理员</div> <div>删除</div>
<input type="checkbox"/>	15007272720		管理员	Sat, 25 May 2019 18:42:42 GMT	<div>取消管理员</div> <div>删除</div>
<input type="checkbox"/>	15007272721		普通用户	Sat, 25 May 2019 18:42:42 GMT	<div>设为管理员</div> <div>删除</div>

图 5.6 管理用户

共有 3 条记录

测试

选择分类

请输入用户名或者邮箱

添加搜索条件

用户名	试卷名称	心情平稳	心情有些焦虑	心情适中	心情焦虑	心情糟糕	总分
15007272717	测试	1	2	2	4	2	11
15007272719	测试	1	2	2	3	3	11
15007272720	测试	2	0	1	3	5	11

图 5.7 筛选成绩

5.3 评测中心

评测中心是用户参加评测和查看报告的模块，具体评测如图 5.9 所示，评测的结果如图 5.10 所示，可以看到用户自己的各个指标的得分、试卷的平均分、自己的分数高

于做试卷的人数的比例、自己试卷的评论和最多人选的答案。

1. 头痛

☐ 没有

☐ 较轻

☐ 中度

☐ 较重

☐ 严重

<

>

图 5.8 评测内容

<

出现在

这是描述

	责任心	爱心	激情	努力
平均分	6	8	10	16
我的得分	6	8	10	16
高于	0%	0%	0%	0%

有 1 人已完成该评测

1。2

备注：

这种标志代表最多人选的选项

这种标志代表你自己的选项

1. 说到喜糖，你觉得下面哪种糖最适合？

A. 甜酥糖

B. 牛奶糖

C. 跳跳糖

2. 如果感冒了，你一开始会先用食疗法试一试吗？

A. 不确定

B. 会

图 5.9 评测结果

25

第 6 章 结论与展望

6.1 系统完成

心理评测系统完成的功能性需求：

- (1) 用户注册、登陆、邮箱验证、找回密码
- (2) 管理员管理试卷、管理用户、筛选成绩
- (3) 用户参加评测、查看评测结果，用户根据结果了解当下的心理状态，同时可以明确以后在生活中需要做哪些调整

心理评测系统完成的非功能性需求：

- (1) 页面响应迅速，不会出现白屏等情况。
- (2) 对于难以表述的题目，使用图片进行辅助。
- (3) 用户可在疲倦时暂停答题，系统将对已答部分进行保存，下次进入时可选择继续完成。整体设计友好温和，尽量不给受测者压力，保障测评结果尽量少受外界因素的干扰

6.2 未来展望

(1) 当前心里评测系统对于试卷的管理还不够完善，只能通过上传 excel 的表格的形式，调整题目的时候只能删除当前的试卷后再进行上传。未来希望做到能够在页面上新增题目和修改题目的信息。

(2) 权限层级不够多，只设有普通用户和管理员两个权限。未来希望加入权限管理系统的权限，对不同的页面都有一个权限列表而不是单纯的区分角色。

(3) 筛选成绩得到的用户列表暂时没有什么用处。未来希望根据成绩筛选出用户以后能对用户进行一些操作，比如发送邮箱类似一样的通知，以邮件的方式通知用户报告或者一些结论性的东西。

(4) 题目之间没有关联性。未来希望题目是动态化的，后面的题目都是根据前面已做的题目计算得来的。

(5) 数据分析还很浅显，没有更加复杂层级的数据分析。未来希望更加的完善数据分析的维度和算法的复杂度。

参考文献

- [1] 未来科技. jQuery 实战从入门到精通 [M]. 水利水电出版社, 2017.
- [2] 叶维忠. Python 编程从入门到精通 [M]. 人民邮电出版社, 2017.
- [3] 杜文洁. 高等学校毕业设计（论文）指导教程——电子信息类专业 [M]. 水利水电出版社, 2015.
- [4] 刘一奇. React 与 Redux 开发实例精解 [M], 2016.
- [5] 程墨. 深入浅出 React 和 Redux[M]. 机械工业出版社, 2017.
- [6] 程墨. 深入浅出 RxJS[M]. 机械工业出版社, 2018.
- [7] 张静. 慧眼识人——心理学评测在人才管理中的应用 [M]. 辽宁科技, 2016.
- [8] 吴浩麟. 深入浅出 webpack[M]. 电子工业出版社, 2018.
- [9] Crockford. JavaScript: The Good Parts[M]. Southeast University Press, 2008.
- [10] BANKS A, PORCELLO E. Python 编程从入门到精通 [M]. China Electric Power Press, 2017.
- [11] JAWORSKI M, ZIADE T. Expert Python Programming[M]. PacktPublishing, 2016.
- [12] RAMALHO L. Fluent Python[M]. Posts and Telecommunications Press, 2017.
- [13] GORELICK M, OZSVALD I. High Performance Python[M]. Posts and Telecommunications Press, 2017.
- [14] Miguel Greenberg. Flask Web Development[M]. Posts and Telecommunications Press, 2018.
- [15] Gustave Le Bon. The crowd: a study of the popular mind[M]. Guangxi Normal University Press, 2011.

致谢

四年的大学生活即将结束同时也代表着十六年的学生生涯也即将画上句号，但这只是我人生的开始，感谢父母孜孜不倦的教导和我生活、学业、生活上的支持，也感谢这些年来老师勤勤恳恳耐心的教导。本系统设计与实现期间得到了我的导师刘传文的耐心指导，是刘老师让我明白学术的力量，营造了一种良好的学术氛围，让我的系统更加完善，论文更加严谨。历时一个学习的毕业设计，期间获得了许多同学和老师的帮助，在这里真挚的感谢你们。