

# 混合式学习的英语阅读移动学习平台的设计与开发

## 1.1 相关技术

### 1.1.1 Spring Boot

Spring Boot 是 Spring 框架的子框架之一，是 Pivotal 团队在 2013 年提供的 Java EE 编程的全新开源轻量级框架。Spring Boot 是所有基于 Spring 开发项目的起点，其设计是为了让你尽快跑起来 Spring 应用程序并尽可能地减少你的配置文件。

Spring Boot 为 Java 开发人员提供了一个很好的平台。Spring Boot 开发项目只需非常少的几个配置就可以搭建起来一个 Web 项目，为我们省去了 SSM 项目繁琐的搭建配置，以更简单的方式开发 Spring 应用程序，减少了开发时间并提高了生产力。Spring Boot 易于理解，开发部署 Spring 应用快速简单，可扩展性良好简单，与容器兼容，使用最低配置为我们提供了一种更简单的应用程序入门方式。

### 1.1.2 Spring

Spring 框架是一个开源的 Java 轻量级框架。它的出现有效解决了系统中其它各层与业务逻辑层的松耦合问题，使得 J2EE 开发更加容易。Spring 在经过不断完善和发展后，成为企业级开发的热门首选。以下为 Spring 框架的特点：

（1）轻量级容器。Spring 是轻量级的。Spring 框架包含和管理着应用中对象的配置和生命周期。而且 Spring 直接支持各种类型的框架，使得系统开发变得更加灵活。

（2）控制反转。Spring 通过控制反转（IoC）技术实现低耦合，控制对象之间的依赖关系，对象以一种被动方式传递到所需的地方，如此可降低对象之间的依赖程度。

（3）面向切面。Spring 支持面向切面的编程（AOP），将应用的业务逻辑与系统服务分离，使得影响不同对象的行为可以封装到可重用模块当中，提高了开发效率与系统质量，降低业务逻辑和系统服务的依赖程度。

(4) 对已存在的技术更加易用。Spring 框架提供了非容器依赖的测试模块 Spring Test 用作单元测试，同时也提供了对于事务管理、异常处理等更为易用的处理方法。

### 1.1.3 Spring MVC

Spring MVC 框架是 Spring 框架中提供对于 MVC 模式实现的框架，将 Web 层进行职责解耦。借助注解，Spring MVC 提供了基本是 POJO(Plain Ordinary Java Object，简单的 Java 对象)的开发模式，使得控制器的开发和测试更加简单。以下为 Spring MVC 的工作流程的简单介绍：

Spring MVC 运行原理：

- (1) 用户发送请求到 DispatcherServlet（前端控制器）。
- (2) DispatcherServlet 收到请求调用 HandlerMapping（处理器映射器）。
- (3) HandlerMapping 根据请求 URL（统一资源定位符）找到具体的处理器，生成处理器对象，并将其一并返回给 DispatcherServlet。
- (4) DispatcherServlet 通过调用 HandlerAdapter（处理器适配器）执行处理器对象（即后端控制器，Controller）。
- (5) 处理器对象执行完成之后返回 ModelAndView（包含了模型和视图的对象）。
- (6) HandlerAdapter 将执行结果 ModelAndView 返回给 DispatcherServlet。
- (7) DispatcherServlet 将 ModelAndView 传给 ViewResolver（视图解析器）。
- (8) 视图解析器解析后返回具体视图。
- (9) DispatcherServlet 对视图进行渲染，将模型数据填充至视图中，并对发送请求的用户进行响应。

### 1.1.4 MyBatis

MyBatis 是优秀的持久层框架，支持定制化的 SQL 查询，存储过程以及高级映射。MyBatis 几乎避免了所有的 JDBC 代码和参数的手工设置以及结果集的检索。MyBatis 可使用简单的 XML 或注解配置和原始映射信息，将接口和 JavaBeans（普通的 Java 对象）映射为数据库中的记录。MyBatis 最强大的特性

之一是动态 SQL 语句功能，使得开发变得更加方便简单。同时它也拥有许多优点：

（1）简单易学。掌握 SQL 语句即可快速上手使用 MyBatis，通过官方文档和源代码，可以比较容易的掌握它的设计思路和实现。

（2）简洁灵活。MyBatis 可以进行更细致的 SQL 优化，同时减少查询字段。并且它对动态 SQL 的查询支持良好。所有的 SQL 语句均存储于 XML 文件中，方便统一管理与优化。

### **1.1.5 Bootstrap**

Bootstrap 是基于 HTML、CSS、JavaScript 的响应式前端框架，它简洁灵活，使得 Web 开发更加快捷。Bootstrap 提供了优雅的 HTML 和 CSS 规范，它是由动态 CSS 语言 Less 写成。它具有以下特点：

（1）拥有丰富的组件。根据这些组件，可以快速搭建一个美丽、功能完备的网站。Bootstrap 其中包括以下组件：下拉菜单、按钮组、按钮下拉菜单、导航、分页、排版、缩略图、警告对话框、进度条等。

（2）浏览器支持。所有的主流浏览器都支持 Bootstrap。

（3）容易上手。只要具备 HTML 和 CSS 的基础知识，就可以快速地上手 Bootstrap。

（4）响应式设计。用户通过各种尺寸的设备浏览网站均获得良好视觉效果。Bootstrap 的响应式样式可自适应于 PC 机、平板电脑和手机等不同设备的屏幕。

### **1.1.6 FreeMarker**

FreeMarker 是一款用 Java 语言编写的模板引擎。它是基于模板文件生成其他文本输出的通用工具。FreeMarker 被设计用来生成 HTML 网页，配置文件，源代码等，它比较适合作为 Web 应用框架的视图组件。Java 程序准备要显示的数据，由 FreeMarker 页面通过模板显示展示的数据。

### **1.1.7 MySQL 数据库**

MySQL 是一款开源关系型数据库管理系统。MySQL 使用用于访问数据库的最常用的标准化语言：SQL 语言。由于其体积小、速度快、总体拥有成本低，

尤其是开放源码这一特点，一般中小型网站的开发都选择 MySQL 作为网站数据库。

### **1.1.8 Java**

Java 是一门功能强大和简单易用的面向对象编程语言，因为 Java 具有简单性、分布式、健壮性、安全性、可移植性等等，并且支持编写桌面应用程序、Web 应用程序、分布式系统和嵌入式系统应用程序等，Java 被广泛应用于系统开发。本系统的后台开发主要采用 Java 编程语言。

## 2 系统的分析与设计

### 2.1 需求分析与设计

#### 2.1.1 系统业务需求分析

混合式学习的英语阅读移动学习平台是一个用于师生对英语阅读学习交流的教育平台，平台主要包括文章阅读学习，课程学习，阅读测试等教育功能及后台教务管理和相关内容管理。整个平台包括两个子系统，英语阅读学习平台和后台管理系统。其中英语学习平台前台分为学生用户端和教师用户端，后台管理系统分为企业院校管理端和系统管理员端。平台功能主要功能选项如下：

表 3-1 英语阅读平台功能选项表

英 语 阅 读 学 习 平 台	教师端	学生端	企业院校 端	管理员端
功 能 选 项	课程与班级	学习课程	教师学生 管理	教师学生 管理
	作业与测试	阅读测评	文章管理	企业院校 管理
	课程学习设定	课程资源	课程管理	课程管理
	补充资源管理	学习成绩	班级管理	班级管理
	综合成绩管理	英语阅读	资讯管理	资讯管理
	班级管理	娱乐文章	院系管理	院系管理
	实训管理	课程实训	实训管理	实训管理
	通知消息	通知消息	信息管理	信息管理
	交流评价	交流评价	数据分析	文章管理
				数据分析

平台入口包括课程资源展示区、用户学习区及课程交流区三大板块，主要功能是展示课程优势、提供游客体验并导航至企业入口。栏目设计上，平台入口包括首页、英语阅读、娱乐、课程、活动、测试、博客等栏目及教师、公告等信息。企业入口是平台针对每一机构或院校的专用入口，平台用户可以从所在院校的入口直接登录，还可在院校入口查看到本院校相关的资讯、公告、学友圈等。

用户登录平台后，根据其所在的机构或院校及个人身份，自动进入对应的操作区域，操作区域以个人主页的方式展示。后台管理区域仅供平台运营管理人员使用。根据该学习平台的主要功能划分有以下几种角色：学生用户，老师用户，企业院校用户，系统管理员。以下是根据不同的角色分别阐述其需求分析：

### （一）学生用户

学生主页包含机构或院校及班级信息、个人信息，功能概要及导航信息。具体包括：

1.个人信息：格式为账号、消息，单击账号弹出菜单修改密码、退出平台，退出平台后跳转至院校主页；单击消息弹出消息列表查看；

2.导航菜单：（主页、我的课程、英语阅读文章（包括热点速递，Quora 精选，一词日历，老外看中国等栏目）、娱乐文章（包括英美剧资讯，时尚搭配，趣味职场，口语对话等栏目）、我的（我的课程，综合实训，个人任务，课程成绩，能力档案，我的通知）、我们（教师，课程，活动，新闻资讯，博客文章）、账号（个人信息，消息，修改密码，退出登录），单击进入相应栏目；

3.我的课程：显示我当前学习的课程，单击进入课程详情（课程介绍、任课教师、实训计划、教学进度，课程知识点及学习情况，知识点详情、综合练习、错题回顾等）；

4.英语阅读：单击分别进入到相应的阅读栏目（热点速递，Quora 精选，一词日历，老外看中国），对应栏目有文章列表，选择对应文章进行双语阅读；

5.娱乐文章：单击分别进入到相应的阅读栏目（英美剧资讯，时尚搭配，趣味职场，口语对话），对应栏目有文章列表，选择对应文章进行双语阅读；

6.阅读测试：单击选择英语阅读测试题，在规定时间内作答，交卷后后台批改并返回测试结果。

7.个人任务：个人任务列表，可以选择进行中及已完成两种，也可查看全部任务跳转至相应界面，列表内容包括任务名称、任务类型、开始时间、结束时间、当前状态、发布人、发布时间等；

## （二）教师用户

教师主页包含院校信息、个人信息，主要功能概要及导航信息等，具体包括：

1.个人信息：（姓名、账号），单击姓名弹出菜单修改密码、退出系统；单击消息查看消息列表；

2.导航菜单：（主页、英语阅读，娱乐，课程管理、综合实训、任务管理、能力档案、数据分析，我们（教师，通知，研究，年终奖，课程，活动，博客）），单击进入相应栏目；

3.任务管理：我发布的任务列表，可以筛选班级、状态（进行中/已完成）两种，可发起新任务，也可查看全部任务跳转至相应界面；

4.课程管理：包括创建新课程，管理的课程，学习的课程，课程任务管理，课程动态，教学资源上传等；

5.能力档案：单击分别进入到相应的能力档案业务界面；

6.数据分析:包括课程成绩录入，课程成绩分析等；

7.实训管理：包括创建新实训，管理实训，实训任务管理，资源上传等；

## （三）企业院校用户

企业教务主页包含企业信息（或院校信息）功能概要及功能菜单信息等，包括：

1.个人信息：（姓名、账号），单击姓名弹出菜单修改密码、退出系统；

2.系统管理：院系管理、班级管理、教师管理、学生管理、课程管理，成绩管理，文章管理；

3.实时数据统计界面，显示当前的相关实时数据；

4.门户管理：企业或院校定义的主页内容维护，如轮播图管理、院校资讯管理（信息发布、编辑、删除）

5.实训管理：布课程实训计划，各课程、教师、班级的实训数据汇总统计；

6.数据分析：学生成绩查询、参与度分析、实训效果分析、学生能力档案；

7.交流监管：监管教师及学生在交流圈中发布的内容，包括图片浏览/删除、敏感词设置、全文搜索、话题（含回复）锁定与删除等；

8.功能菜单：方式为左侧树状功能菜单，包括资讯管理、教学监管、数据分析、教师考核、交流监管、系统管理等。

#### （四）后台管理员用户

后台系统管理员用户具有管理整个平台后台数据的权限。我们需要有用来管理审批客户端的各种数据交互，来使得应用使用更加安全，体验更加完善。后台管理员主页包含平台功能概要及功能菜单信息等，包括：

1. 管理员信息：（姓名、账号），单击姓名弹出菜单修改密码、退出系统；
2. 系统管理：企业院校管理、管理员管理、教师管理、学生管理、院系管理、班级管理、课程管理、成绩管理，文章管理等，主要是对系统数据表的管理；
3. 数据统计界面，显示当前的相关实时数据；
4. 门户管理：企业或院校定义的主页内容维护，如轮播图管理、院校资讯管理（信息发布、编辑、删除）；
5. 实训管理：发布课程实训计划，各课程、教师、班级的实训数据汇总统计；
6. 数据分析：学生成绩查询、参与度分析、实训效果分析、学生能力档案；
7. 交流监管：监管教师及学生在交流圈中发布的内容，包括图片浏览/删除、敏感词设置、全文搜索、话题（含回复）锁定与删除等；
8. 功能菜单：方式为左侧树状功能菜单，包括资讯管理、教学监管、数据分析、教师考核、交流监管、系统管理等菜单。

### 2.1.2 功能结构分析

根据系统的业务需求分析，本系统的功能结构可以分为两个部分，一部分是应用的客户端，其中客户端有学生端和老师端，另一部分是管理员进行应用维护的后台管理端，管理端分为企业端和管理员端。各个部分功能结构图如下：

1. 英语阅读平台（学生端）功能结构图



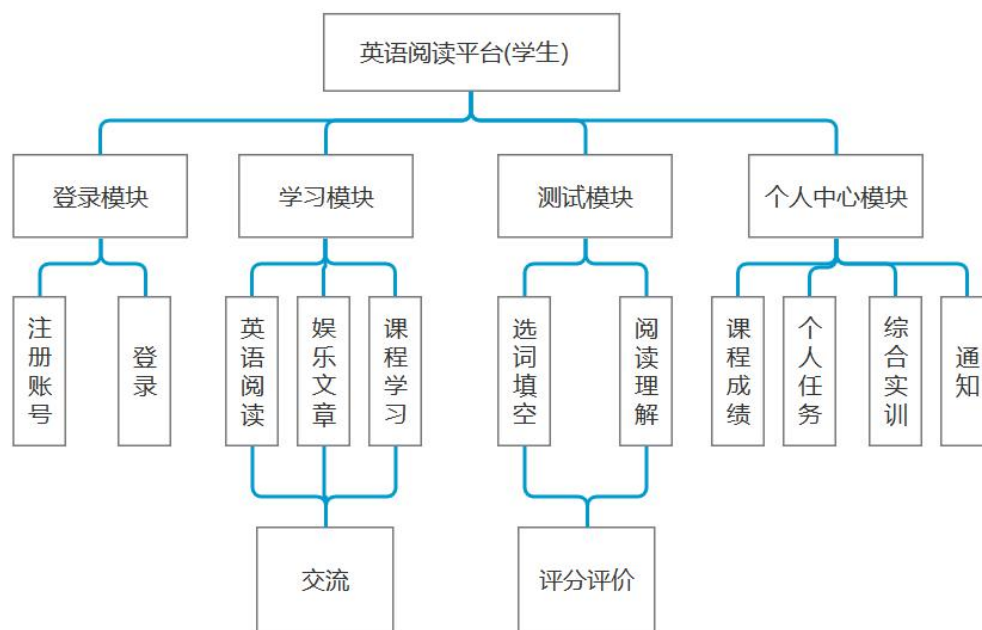


图 3-1 英语阅读平台（学生端）功能结构图

## 2. 英语阅读平台（教师端）功能结构图

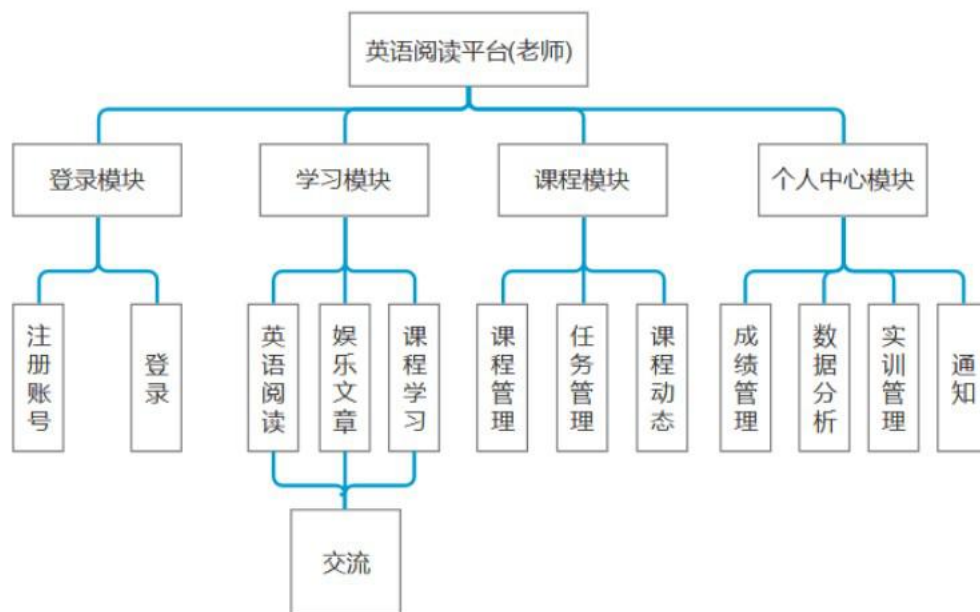


图 3-2 英语阅读平台（教师端）功能结构图

### 3. 英语阅读平台（企业院校端）功能结构图

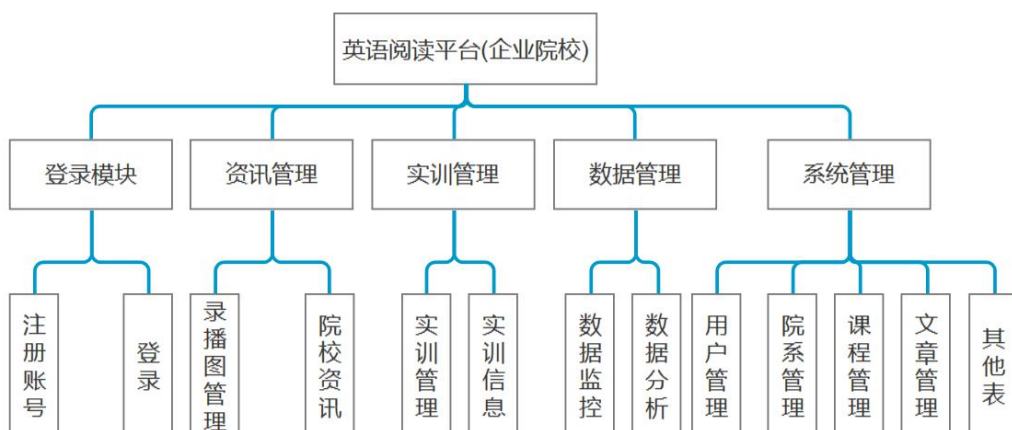


图 3-3 英语阅读平台（企业院校管理端）功能结构图

### 4. 英语阅读平台（管理员端）功能结构图

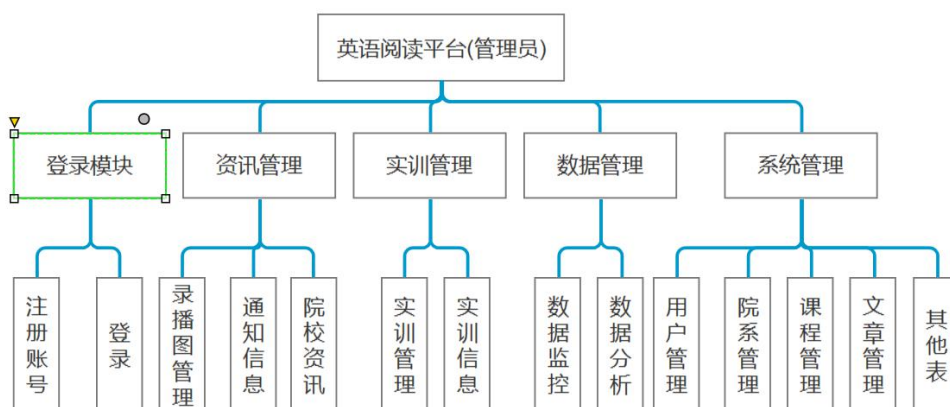


图 3-4 英语阅读平台（管理员）功能结构图

## 2.1.3 混合式课程设计

混合式课程中包含了课堂面对面学习、自主学习、协作学习和网络学习等学习方式。混合式课程教学中所涉及的学习内容，学习环境，学习资源以及学生教师的任务和评价，课程教学设计涵盖内容表如下：

表 3-2 课程教学设计涵盖内容

学习内容	学 习 环境	学 习 资源	学 生 任务	教 师 任务	评价
单元 1	1. 教 室	1.多媒 体课件	1. 接 受新知	1. 课 堂讲授	1.课前 演讲
	2.英语 阅读平台	2.英语 教材	2. 参 与课堂学 习活动	2. 设 计学习活 动	2.课堂 表现
单元 2	3.QQ 学 习 资 源 分 享 群	3.阅读 平台内英 语篇章	3. 课 后自学	3. 布 置任务	3.QQ 群参与讨 论情况
	4.微信 群聊	4.QQ 群分享资 料	4. 完 成课后作 业	4. 答 疑解惑	4.微信 群参与讨 论情况
。。。。。		微 信 群 分 享资料	5. 自 我评价	5. 提 供反馈	5.课后 作业完成 情况

## (1)教学环节设计

混合式学习课程包含课前准备阶段，课堂学习阶段以及课后巩固阶段，其设计表如下：

表 3-3 教学环节设计

教学环节	学生任务	教师任务	学习环境
课前准备	1.预习教材 2.学习平台中 内容	1.准备教案 2.设计课程学 习活动 3.设计作业任 务单	1.自主学习 环境 2.协作学习 环境

课堂学习	1. 课堂学习 2. 学习汇报 3. 完成任务	1.教授新课 2.布置任务 3.反馈、评价总结	1.课堂面授环境
课后巩固	1.复习 2.知识拓展 3.与师生沟通交流	1.辅导 2.答疑解惑	1.自主学习环境 2.协作学习环境

## (2)学习环境设计

混合式学习环境包括英语阅读平台的自主学习环境，QQ群和微信群的协作学习环境以及课堂面授环境。其学习环境设计如图所示：

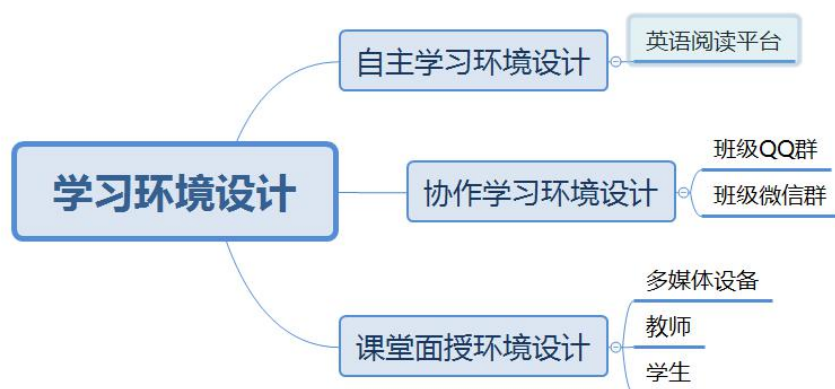


图 3-5 学习环境设计图

## (3)学习活动设计

学习活动设计是混合式课程设计的核心内容，包括课堂讲授，协作学习，讨论交流，资料分享与收集，测试以及反馈等内容。本文核心活动学习活动设计如下表所示：

表 3-4 阅读活动设计

知识类型	活动目标	活动组织	活动评级
事实性知识	知道 理解	1. 收集与布置任务主题明确的	提问 测试

		英语文章。 2. 提供相关背景信息 3. 设计问题 4. 提供参考答案	
概念性知识	知道 理解	1. 收集与布置任务主题明确的文章 2. 提供相关概念知识信息 3. 设计问题 4. 提供参考答案	提问 测试 学习汇报反馈
程序性知识	知道	5. 收集与布置任务主题明确的文章 6. 提供相关概念知识信息 7. 设计问题 提供参考答案	提问 测试

#### 2.1.4 功能模块设计

经过对系统的业务需求分析，功能结构分析，结合混合式课程设计，设计出系统的功能模块。系统分为前台客户端和后台管理端。

（一）以下是英语阅读学习平台客户端所实现的功能模块图：

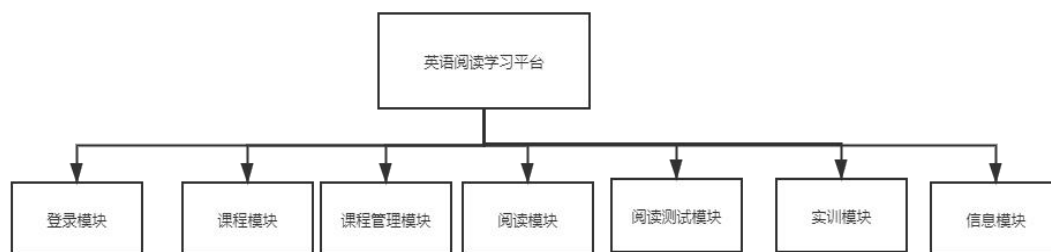


图 3-6 平台客户端功能模块图

（二）以下分别是这两个后台管理端所实现的功能模块图：

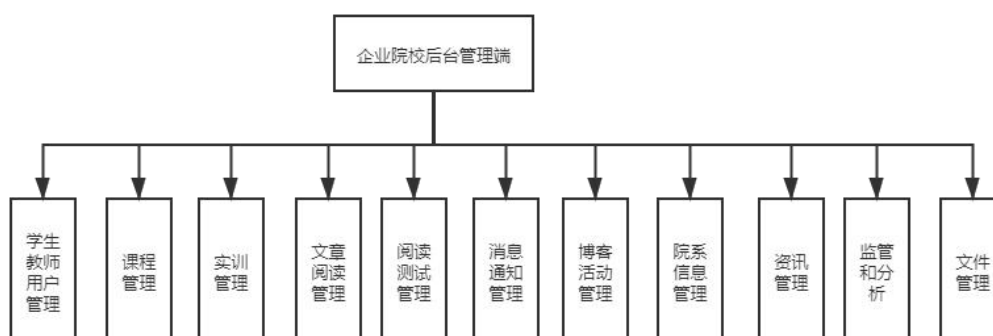
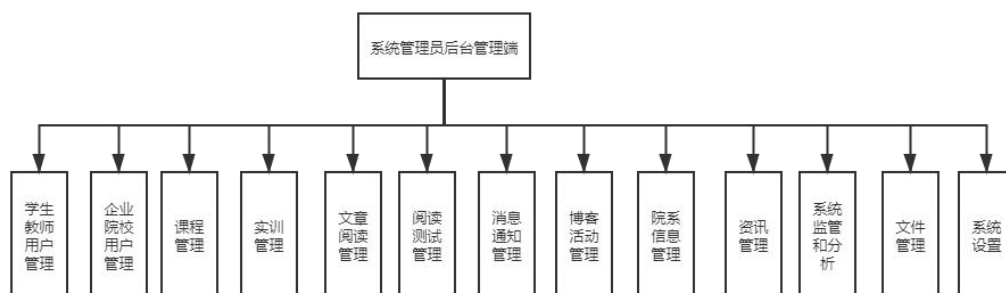


图 3-7 企业院校后台管理端功能模块图



图

3-8 系统管理员后台管理端功能模块图

1.学生端阅读学习平台主要功能模块介绍如下：

（一）学生用户登录注册

本学习平台需要学生用户注册一个账号进行登录系统，登录验证通过才可进入学生主页。

（二）阅读平台发布的英语文章

英语文章分为英语阅读文章和娱乐性文章，阅读文章分为双语阅读，中文阅读和英文阅读，更好实现中英文对照阅读，提高阅读能力。

### （三）选修课程

学生用户在主页查看各门课程及对应老师，选修学习心仪的课程。

### （四）课程学习

学生用户进入自己选修课程进行学习，课程里有课程最新动态，课程公告，课程资源，课程作业，互动交流，班级信息。学生可以查看课程动态和公告，下载教学资源，上传课程作业，课程测试，与老师或同学互动交流，查看课程班级信息。

### （五）消息通知

学生与学生、老师、企业院校与系统管理员的交流功能，学生用户可以实现收取教学交流信息、收取系统通知、回复消息等消息操作

### （六）浏览博客文章

学生用户可以浏览平台发布的博客文章，该类博客文章主要是英语学习的博客文章，帮助学生学会使用平台，养成英语阅读好习惯，掌握正确有效的英语学习方法。

### （七）查看学习平台新闻资讯，关于，研究等信息

学生用户可以查看平台发布的新闻资讯，了解英语新闻时事文章，了解学习平台及研究方向。

### （八）联系我们

学生用户可以通过联系我们填写具体信息联系院校或系统管理员。

### （九）成绩查询

学生用户通过登录选择我的成绩查看对应课程成绩。

### （十）阅读测试

点击进入阅读测试，在规定时间内答完题目后，自动提交到系统进行批改并返回测试结果。

## 2.教师端阅读学习平台主要功能模块如下：

### （一）教师用户登录注册

本学习平台需要教师用户注册一个账号进行登录系统，登录验证通过才可进

入教师主页。

## （二）阅读平台发布的英语文章

英语文章分为英语阅读文章（热点速递，Quora 精选，一词日历，老外看中国）和娱乐性文章（英美剧资讯，时尚搭配，趣味职场，口语对话），阅读文章分为双语阅读，中文阅读和英文阅读，更好实现中英文对照阅读，查看对应文章内容。

## （三）课程管理

教师用户可以创建自己的课程，进入课程管理，课程学习，课程任务管理及课程动态管理。课程管理里有课程最新动态，课程公告，教学资源，课程作业，互动交流，班级信息。教师可以发布课程动态和公告，上传教学资源，下载批改学生完成课程作业，与老师或同学互动交流，查看课程班级信息等。

## （四）综合实训管理

教师用户可以创建综合实训，进行综合实训管理。教师用户进入自己创建实训进行管理，教师可以发布实训动态和公告，上传教学资源，下载观看学生完成实训作业，与老师或同学互动交流，查看实训信息。

## （五）消息通知

老师与学生、老师、企业院校与系统管理员的交流功能，老师用户可以实现收取教学交流信息、收取系统通知、回复消息等消息操作

## （六）浏览博客文章

教师用户可以浏览平台发布的博客文章，该类博客文章主要是英语学习的博客文章，帮助学生学会使用平台，养成英语阅读好习惯，掌握正确有效的英语学习方法。

## （七）查看学习平台新闻资讯，关于，研究等信息

教师用户可以查看平台发布的新闻资讯，了解英语新闻时事文章，了解学习平台及研究方向。

## （八）课程成绩录入和统计

教师用户可以在课程末综合学生表现及平时成绩和课末测试成绩对学生的课程成绩进行录入和统计。

## （九）联系我们



教师可以通过联系我们填写具体信息联系院校或系统管理员。

## 2. 企业院校管理端主要功能模块如下：

### （一）登陆

企业院校管理员登陆后台管理端

### （二）教师管理

根据条件查询教师、新建教师、删除教师、修改教师信息

### （三）学生管理

根据条件分页或模糊查询学生信息、新建学生、删除学生、修改学生信息

### （四）课程管理

根据条件分页或模糊查询课程信息、新建课程、删除课程、修改课程信息

### （五）实训管理

根据条件分页或模糊查询实训信息、新建实训、删除实训、修改实训信息

### （六）文章管理

根据条件查询文章信息、新建文章，删除文章，修改文章信息

### （七）消息管理

根据条件查询用户的消息、审批消息内容、删除消息、发送系统消息通知

### （八）博客管理

根据条件查询博客文章信息、新建博客文章，删除博客文章，修改博客文章信息

### （九）联系我们

根据条件查询联系我们信息表、审批联系内容、删除联系消息、发送系统联系通知

### （十）院系信息管理

根据条件查询联系院系信息表、新建院系信息，删除院系信息，修改院系信息

### （十一）资讯管理

根据条件查询录播图信息和院校资讯信息表、新建录播图信息和院校资讯信息，删除录播图信息和院校资讯信息，修改录播图信息和院校资讯信息

### （十二）系统监管和数据分析

系统具备监管仪表盘模块，可以监控系统。同时根据系统的数据进行分析。

### 3. 后台管理员端主要功能模块如下：

#### （一）登陆

后台管理员登陆后台管理端

#### （二）教师管理

根据条件查询教师、新建教师、删除教师、修改教师

#### （三）学生管理

根据条件查询学生信息、新建学生、删除学生、修改学生信息

#### （四）企业院校管理

根据条件查询企业院校用户信息、新建企业院校用户、删除企业院校用户、修改企业院校用户信息

#### （五）实训管理

根据条件查询实训信息、新建实训、删除实训、修改实训信息

#### （六）文章管理

根据条件查询文章信息、新建文章，删除文章，修改文章信息

#### （七）消息管理

根据条件查询用户的消息、审批消息内容、删除消息、发送系统消息通知

#### （八）博客管理

根据条件查询博客文章信息、新建博客文章，删除博客文章，修改博客文章信息

#### （九）联系我们

根据条件查询联系我们信息表、审批联系内容、删除联系消息、发送系统联系通知

#### （十）院校系信息管理

根据条件查询联系院校系信息表、新建院校系信息，删除院校系信息，修改院校系信息。

#### （十一）资讯管理

根据条件查询录播图信息和院校资讯信息表、新建录播图信息和院校资讯信息，删除录播图信息和院校资讯信息，修改录播图信息和院校资讯信息

## （十二）系统监管和数据分析

系统具备监管仪表盘模块，可以监控系统。同时根据系统的数据进行分析。

### 2.1.5 使用流程图

本应用流程应根据角色的种类为四个部分，第一部分是学生用户学习平台使用流程，第二部分是教师用户学习平台使用流程，第三部分是企业院校用户学习平台的使用流程，第四部分是后台管理系统的使用流程。

1. 学生用户学习平台使用流程图如下：

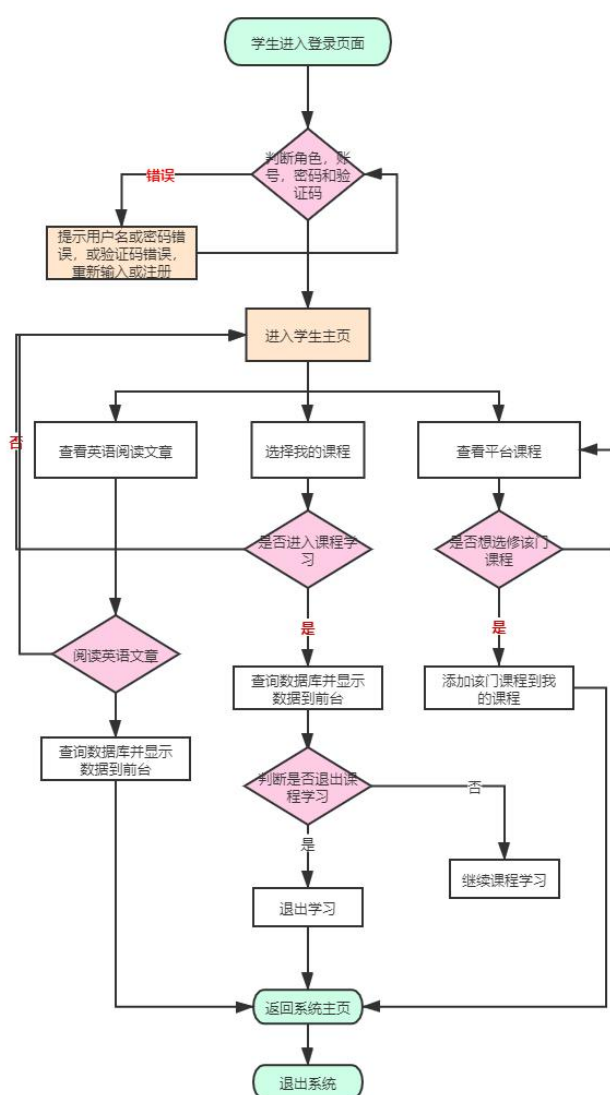


图 3-9 学生用户学习平台使用流程图

2.教师用户学习平台使用流程图如下:

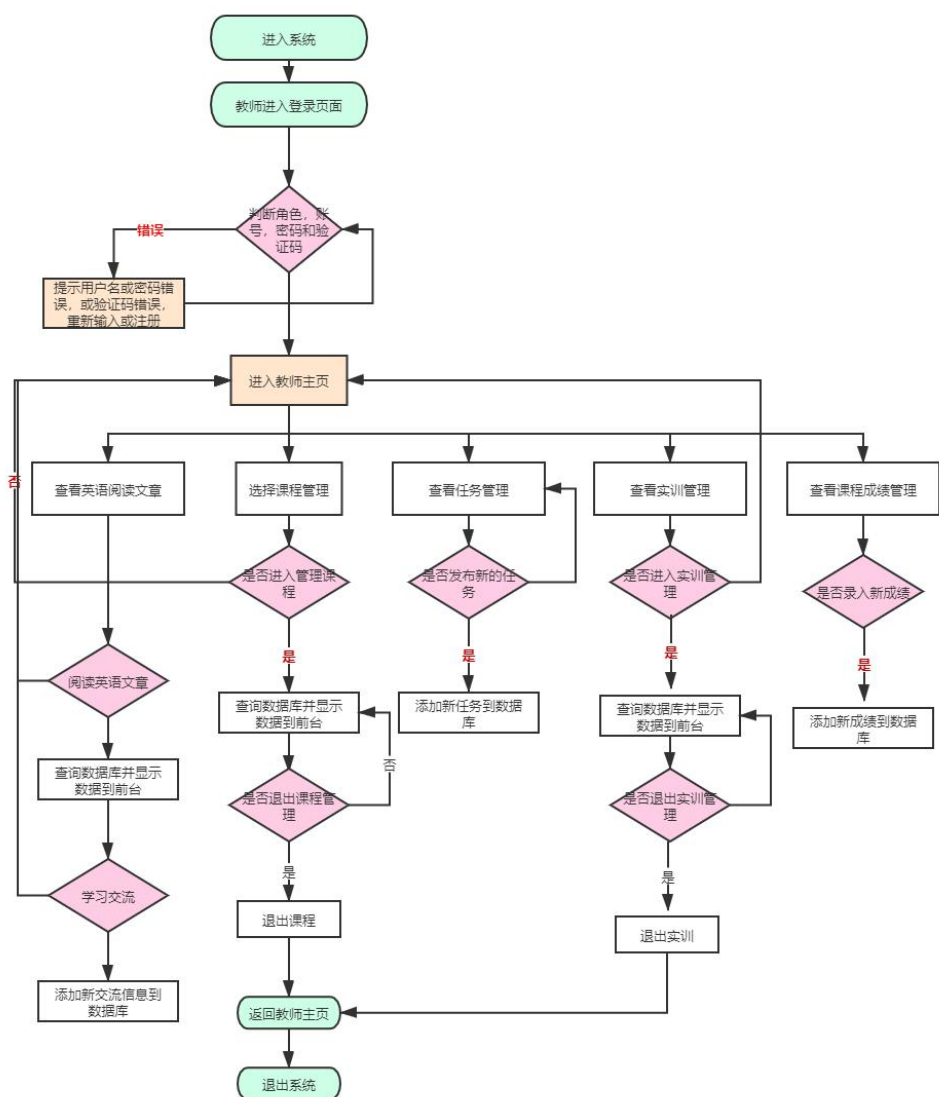


图 3-10 教师用户学习平台使用流程图

3.企业院校用户学习平台使用流程图如下:

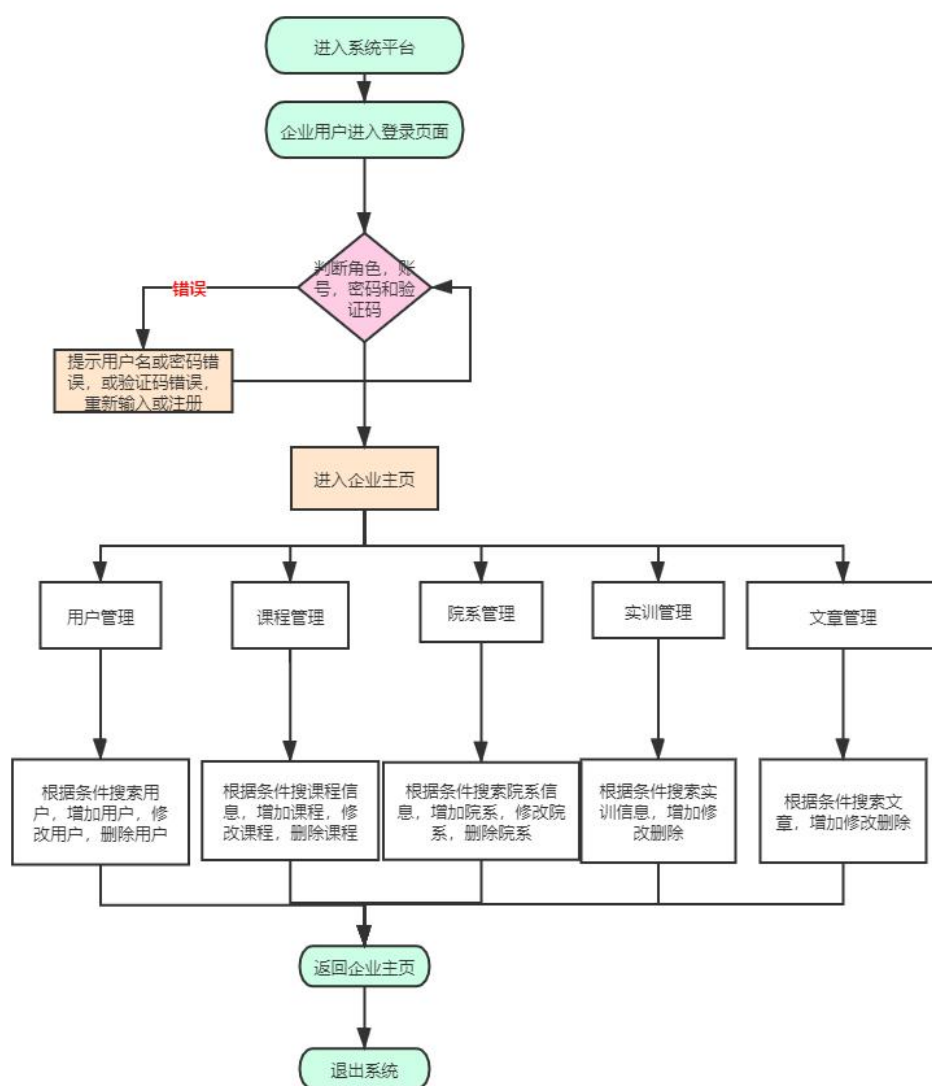
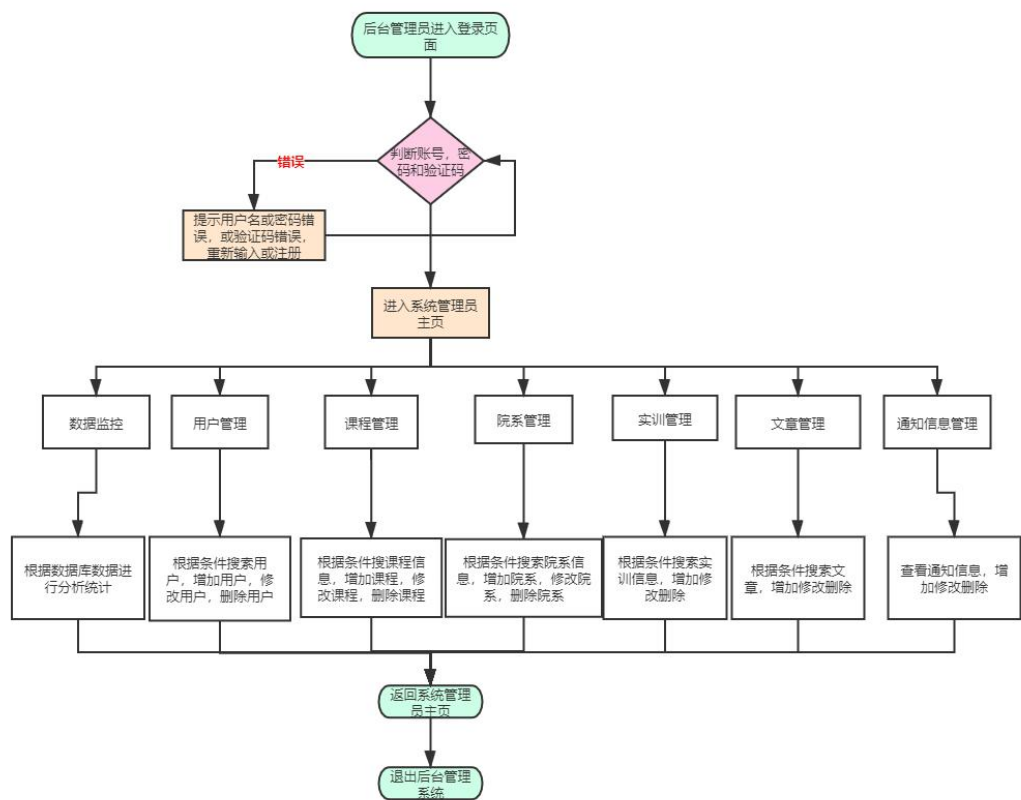


图 3-11 企业院校学习平台使用流程图

4.后台管理员用户平台使用流程图如下:



图

3-12 后台管理员使用流程图

2.1.6 模块流程图

本平台功能模块实现具体分为登录注册模块，课程模块，文章阅读模块，阅读测试模块，实训模块，消息模块，博客模块，成绩模块，活动模块等。由于功能模块实现流程的类似，以下展示核心部分模块的流程图。

(1)登陆模块流程图

登录模块是英语阅读学习平台用户的必需模块，其设计流程图如下：

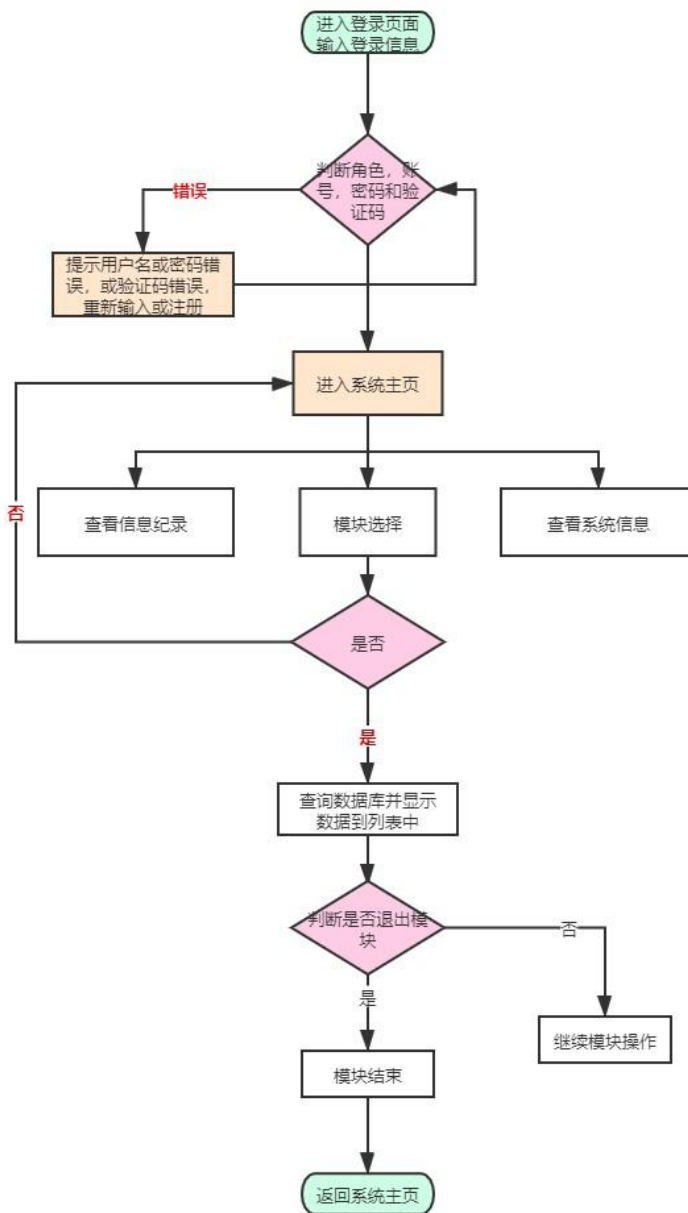


图 3-13 英语阅读移动学习平台登录模块流程图

## (2)课程模块流程图

课程模块分为学生端的课程学习，教师端的课程管理，其流程图如下：

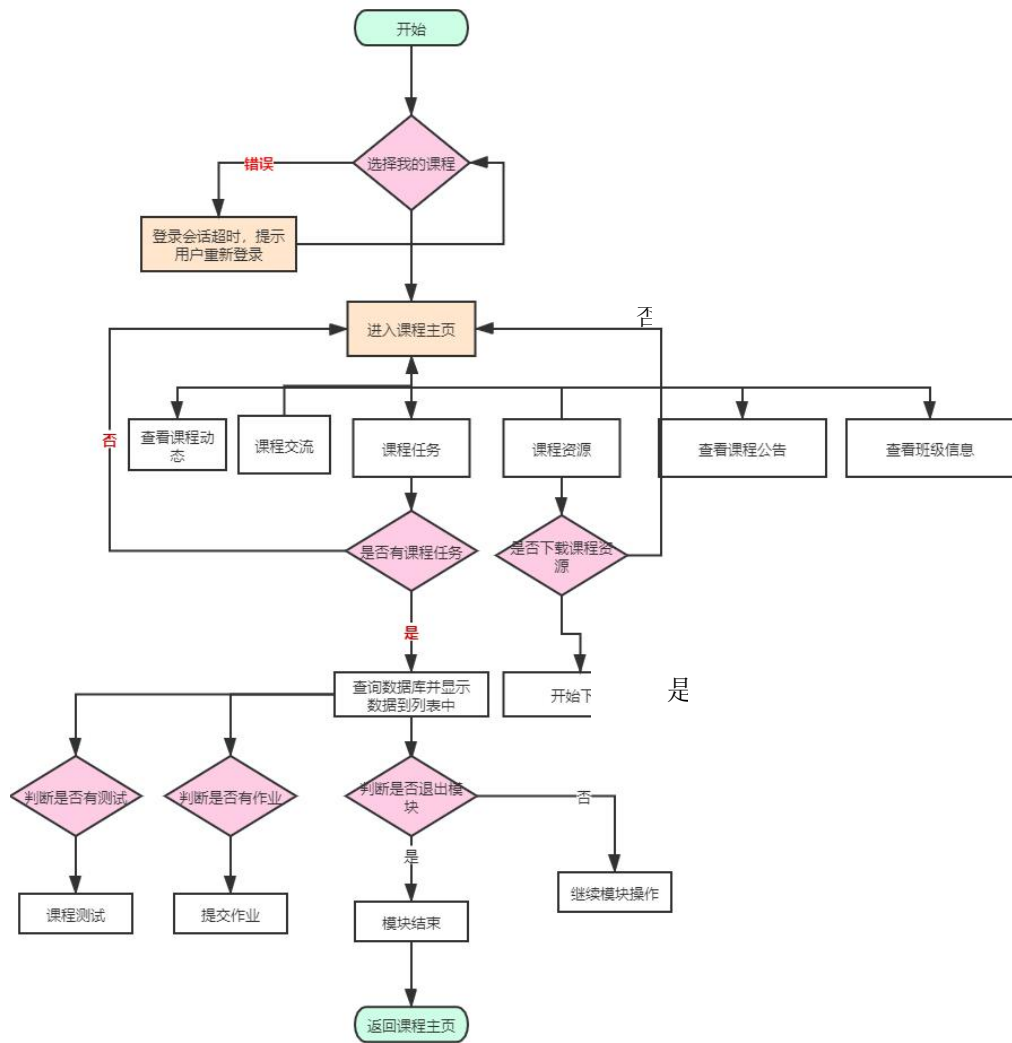


图 3-14 学生课程模块流程图



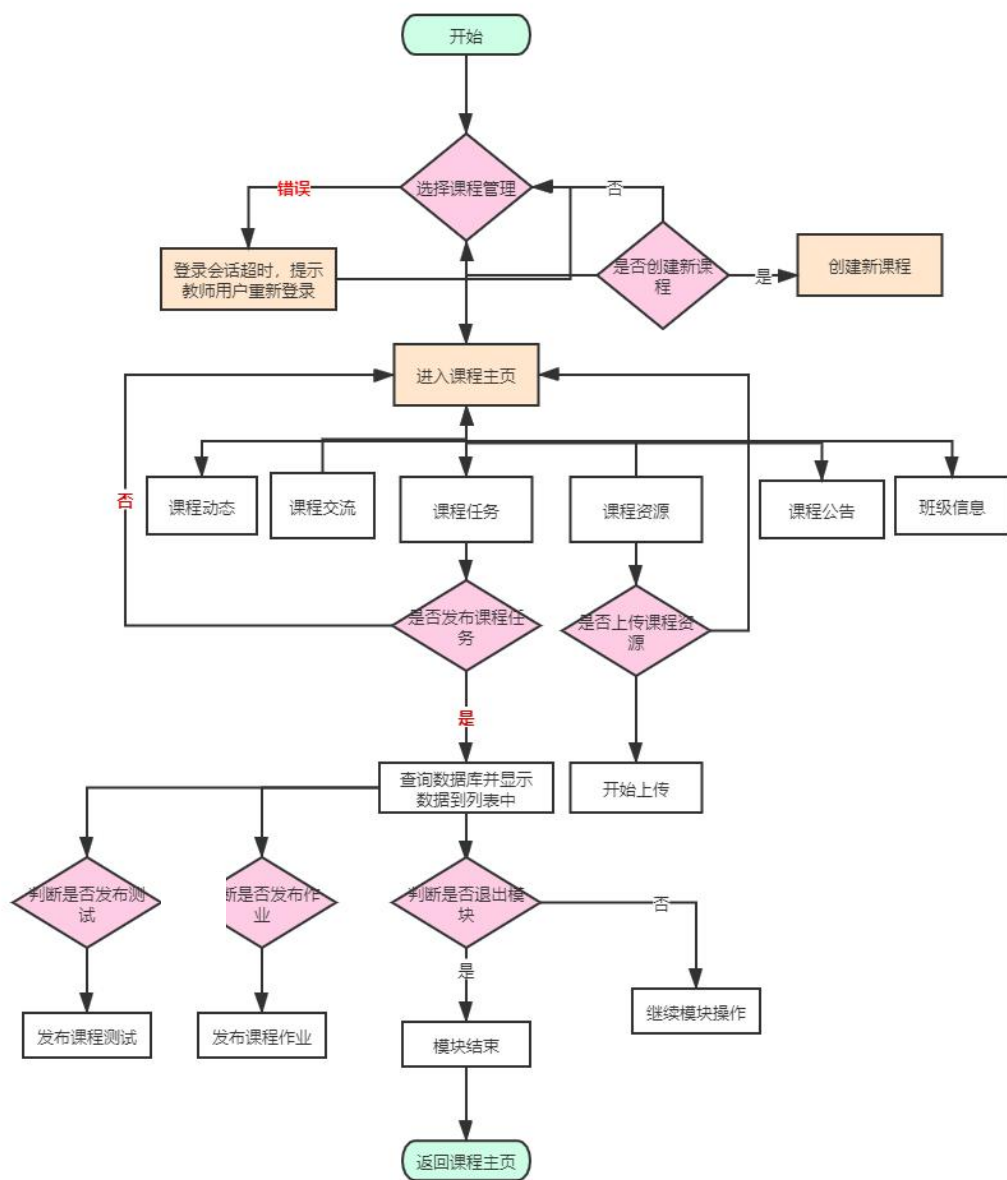


图 3-15 教师课程流程图

### (3)文章阅读流程图

文章阅读是平台的核心模块，其流程图如下：

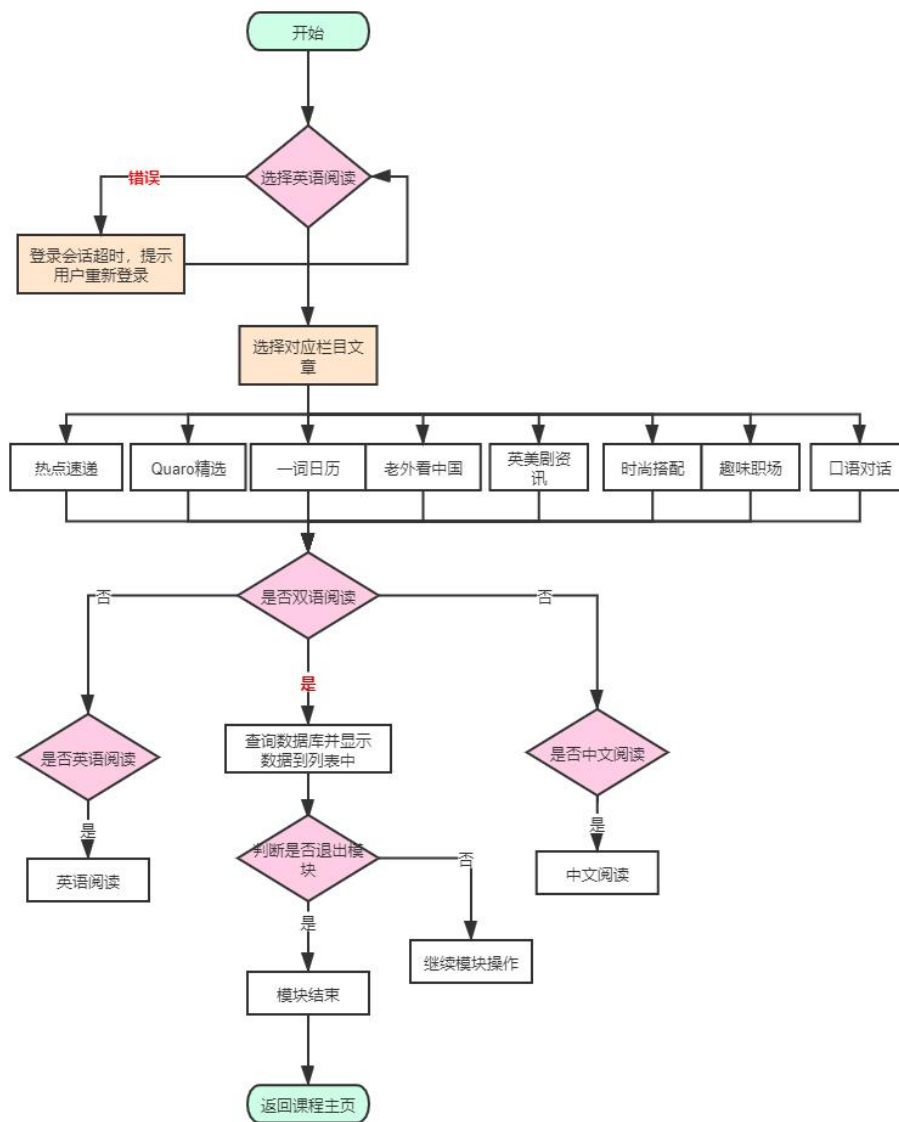


图 3-16 文章阅读模块流程图

#### (4) 阅读测试流程图

阅读测试可以检验学生的英语阅读能力，其流程图如下：

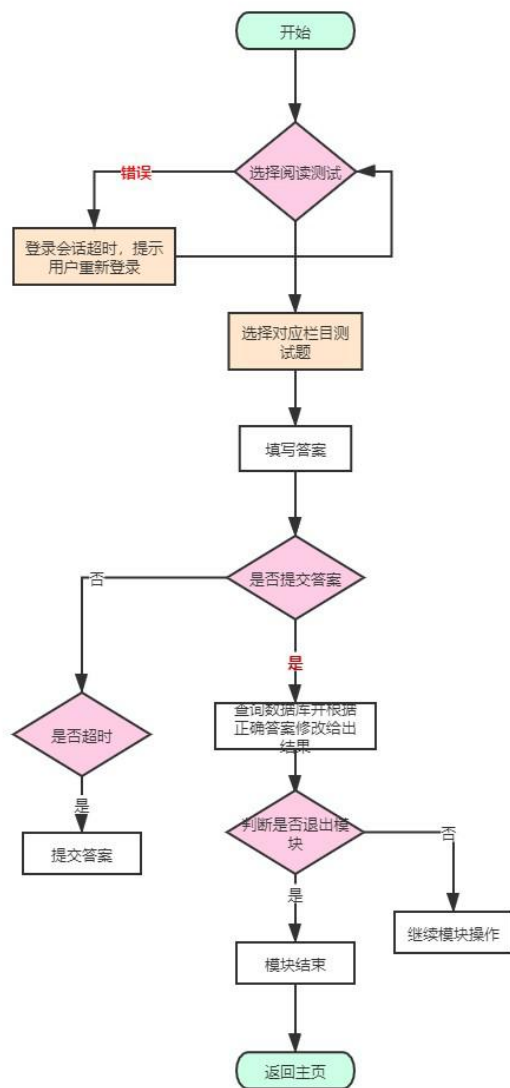


图 3-17 阅读测试流程图

#### (5)实训模块流程图

实训模块分为学生实训和老师实训管理两分模块，其流程图如下：

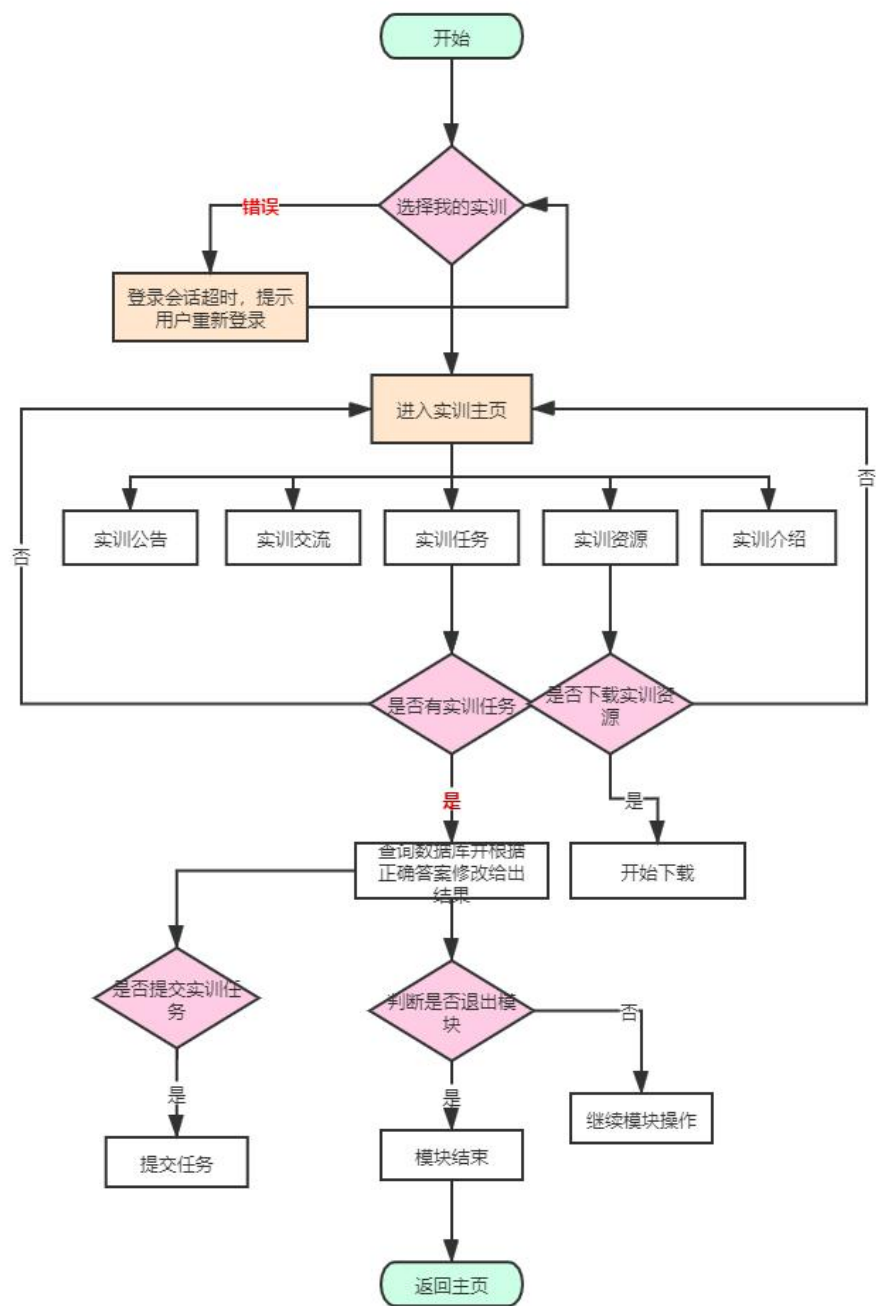


图 3-18 学生实训流程图

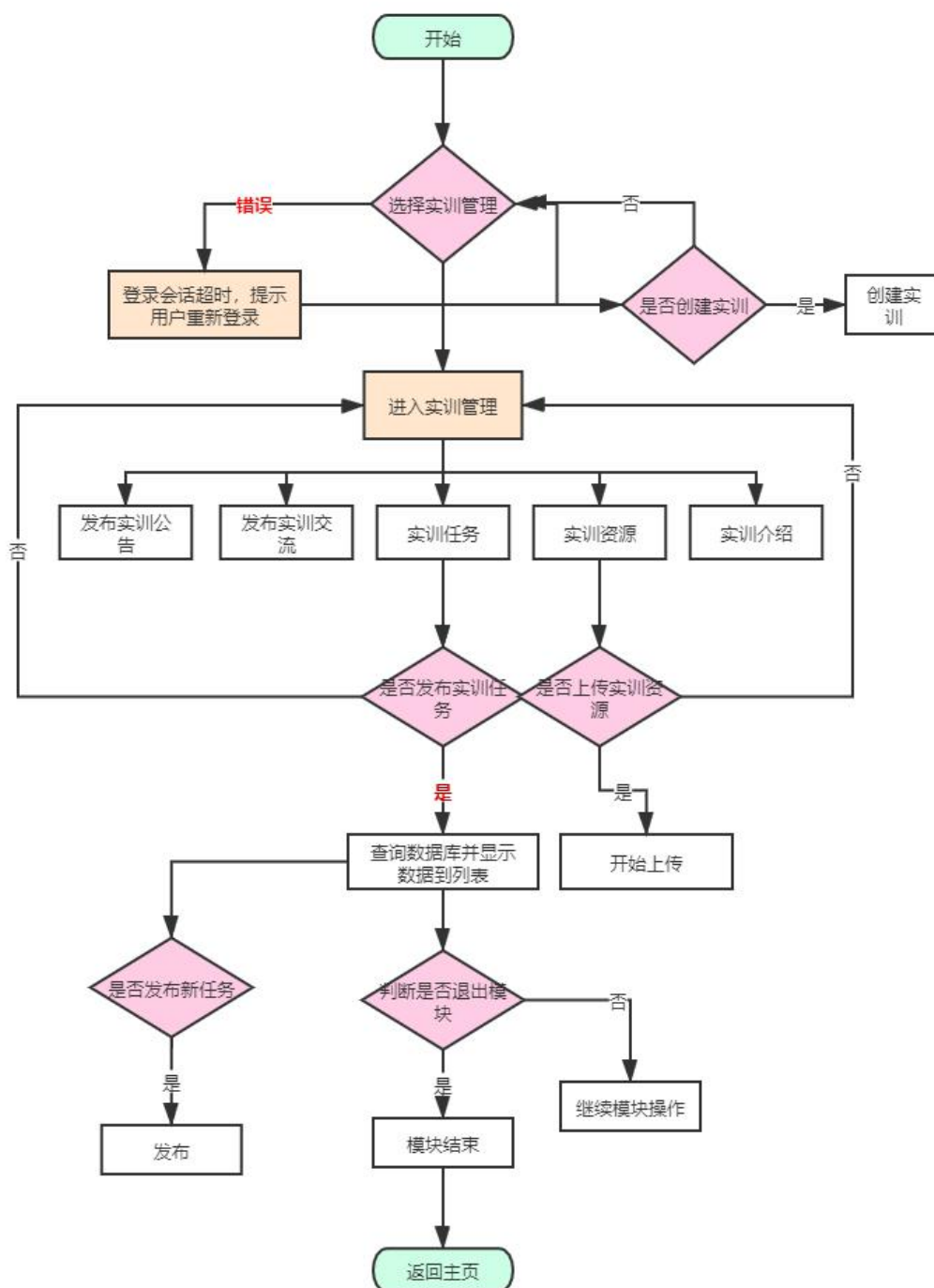


图 3-19 教师实训管理流程图

#### (6)消息模块流程图

消息模块是英语学习平台用户的交互模块，其流程图如下：

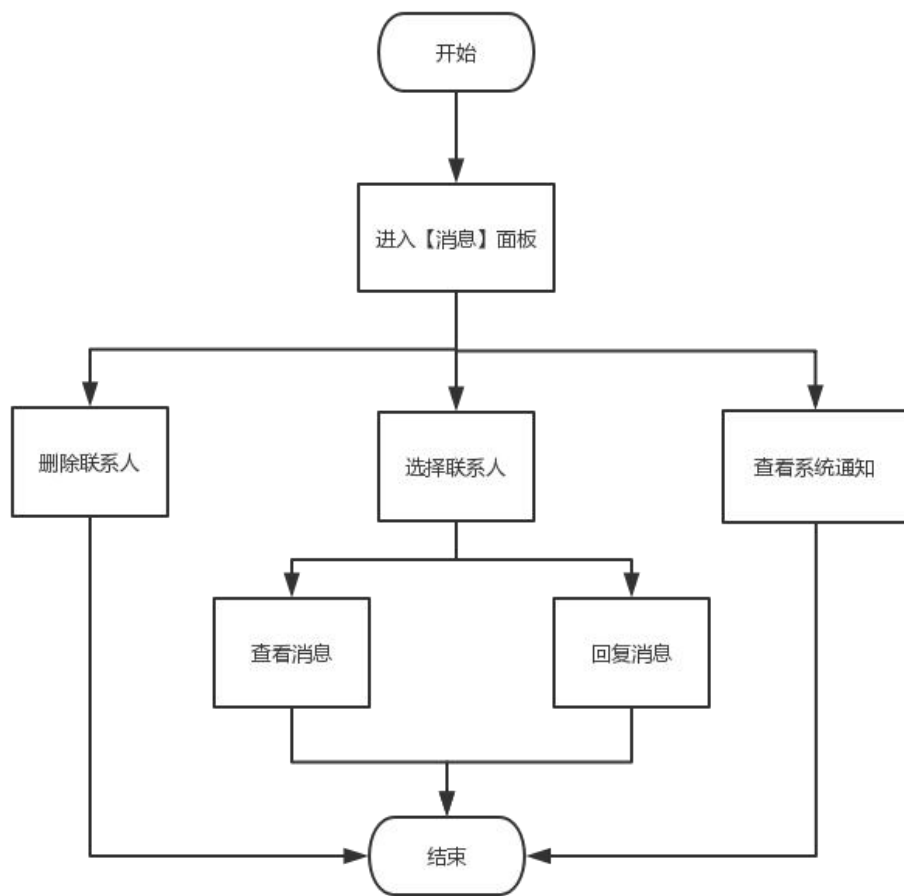


图 3-20 消息模块流程图

## 2.2 数据库概念结构设计

### 2.2.1 实体关系图

该学习平台实体关系图如下：

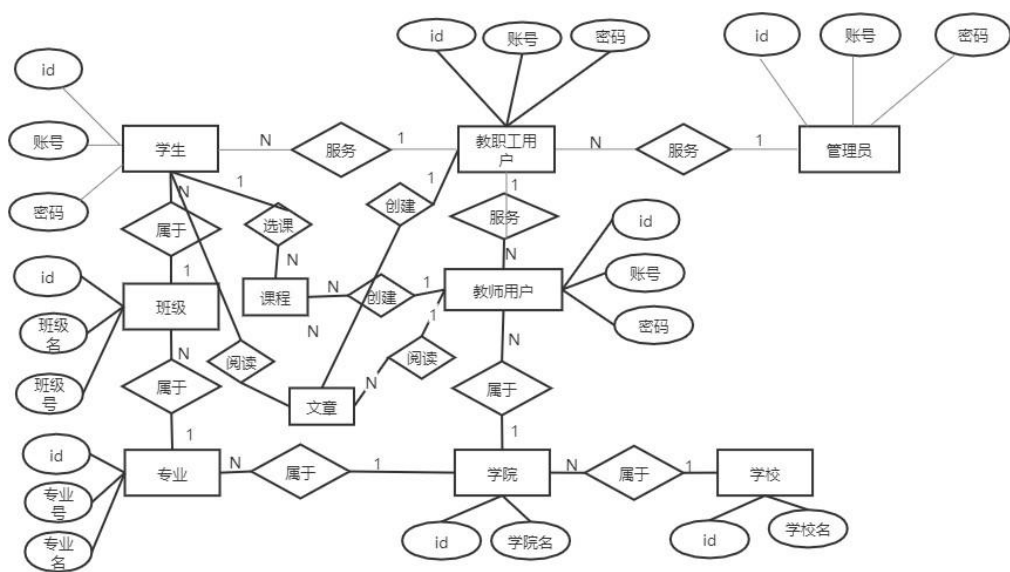


图 3-21 实体关系图

## 2.2.2 物理模型图

本图阐述了实际上各个表的具体属性（字段名称、字段类型），如下：



图 3-22 物理模型图

## 2.3 数据库设计

### 1. 管理员信息表 tb\_superuser



表 3-3-1 tb\_superuser

字段名称	字段类型	约束	说明
Sup_id	Varchar(20)	主键	账号
Sup_psw	Varchar(20)	not null	密码
Sup_name	Varchar(20)	not null	姓名
Sup_email	Varchar(30)	not null	邮箱
Sup_tel	Long	not null	手机号码
Sup_sex	Char(2)	not null, 取“男”或“女”	性别
Sup_birthday	datetime	not null	生日
Sup_authority	int		权限管理

## 2. 学生信息表 tb\_student

表 3-3-2 tb\_student

字段名称	字段类型	约束	说明
S_id	Varchar(20)	主键	账号
S_psw	Varchar(20)	not null	密码
S_name	Varchar(20)	not null	姓名
S_email	Varchar(30)	not null	邮箱
S_tel	Long	not null	手机号码
S_sex	Char(2)	not null, 取“男”或“女”	性别
S_birthday	datetime		生日
M_id	Varchar(20)	Tb_major 外键	专业号

## 3. 老师信息表 tb\_teacher

表 3-3-3 tb\_teacher

字段名称	字段类型	约束	说明
T_id	Varchar(20)	主键	老师账号
T_psw	Varchar(20)	not null	密码
T_name	Varchar(20)	not null	姓名
T_email	Varchar(30)	not null	邮箱
T_tel	Long	not null	手机号码
T_sex	Char(2)	not null	性别
T_birthday	datetime	not null	生日
T_title	Varchar(20)	not null	职称
D_id	Varchar(20)	Tb_dept 外键	院系号

## 4. 机构信息表 tb\_organization

表 3-3-4 tb\_organization

字段名称	字段类型	约束	说明
O_id	Varchar(20)	主键	机构账号
O_psw	Varchar(20)	not null	密码
O_name	Varchar(20)	not null	机构名
O_email	Varchar(30)	not null	邮箱
O_tel	Long	not null	手机号码
O_title	Varchar(20)	not null	机构称号
O_desc	text		机构介绍

## 5. 课程信息表 tb\_course

表 3-3-5 tb\_course

字段名称	字段类型	约束	说明
C_id	Varchar(20)	主键	课程号
C_name	Varchar(30)	not null	课程名
C_credit	float	not null	学分
C_class	int	not null	学时
C_desc	text	not null	课程介绍
C_plan	text		课程计划
C_content	text		课程知识点
C_test	text		课程练习
T_id	Varchar(20)	Tb_teacher 外键	老师账号

## 6. 专业信息表 tb\_major

表 3-3-6 tb\_major

字段名称	字段类型	约束	说明
M_id	Varchar(20)	主键	专业号
D_id	Varchar(20)	引用 tb_dept 外键	院系号
M_name	Varchar(20)	not null	专业名
M_assistant	Varchar(20)	not null	辅导员
M_tel	Long	not null	联系方式
M_desc	text		系介绍

## 7. 院系信息表 tb\_dept

表 3-3-7 tb\_dept

字段名称	字段类型	主外键	说明
d_id	Varchar(20)	主键	院系号

d_name	Varchar(30)	not null	系名称
d_chairman	Varchar(20)	not null	系主任
d_email	Varchar(30)	not null	系邮箱
d_tel	Long	not null	系号码
d_desc	text		系介绍
sch_id	Varchar(20)	引用 Tb_school 外键	学校号

## 8. 学校信息表 tb\_school

表 3-3-8 tb\_school

字段名称	字段类型	主外键	说明
Sch_id	Varchar(20)	主键	学校号
Sch_name	Varchar(20)	not null	校名称
Sch_chairman	Varchar(20)	not null	校长
Sch_email	Varchar(30)	not null	校邮箱
Sch_tel	Long	not null	联系方式
Sch_desc	text		学校介绍

## 9. 选课信息表 tb\_stu\_course

表 3-3-9 tb\_stu\_course

字段名称	字段类型	约束	说明
StuCourse_id	Varchar(20)	主键, 自动递增	选课编号
S_id	Varchar(20)	Tb_student 外键	学生号
C_id	Varchar(20)	Tb_course 外键	课程号
T_id	Varchar(20)	Tb_teacher 外键	教师号
grade	float		成绩
schedule	Varchar(10)		教学进度
status	Varchar(10)		学习状况

## 10. 课程控制设置表 tb\_if\_take\_course

表 3-3-10 tb\_if\_take\_course

字段名称	字段类型	约束	说明
if_take_course	Char(1)	not null, 取"0"或"1"	选课控制
If_input_grade	Char(1)	not null, 取"0"或"1"	成绩录入控制

## 11. 课程任务表 tb\_stu\_task

表 3-3-11 tb\_stu\_task

字段名称	字段类型	约束	说明
StuTask_id	Varchar(20)	主键, 自动递增	编号
C_id	Varchar(20)	Tb_course 外键	课程号

T_id	Varchar(20)	Tb_teacher 外键	教师号 发布人
T_type	Varchar(20)		任务类型
time	datetime	Not null	发布时间
title	Varchar(20)	Not null	任务标题
start	datetime		开始时间
end	datetime		结束时间
content	text		任务内容
Status	Char(1)	not null, 取"0"或"1"	当前状态

## 12. 消息表 tb\_message

表 3-3-12 tb\_message

字段名称	字段类型	约束	说明
Message_id	int	主键, 自动递增	消息编号
S_id	int	Tb_course 外键	学生号
T_id	int	Tb_teacher 外键	教师号
T_type	Varchar(20)		消息类型
time	datetime	Not null	发布时间
title	Varchar(20)	Not null	标题
content	text		内容
Status	Char(1)	not null, 取"0"或"1"	当前状态

## 13. 文章表 tb\_new

表 3-3-13 tb\_new

字段名称	字段类型	约束	说明
new_id	int	主键, 自动递增	文章编号
new_title	int	Not null	标题
new_time	varchar	Not null	时间
new_from	varchar	Not null	来源
new_writer	varchar	Not null	作者
new_img	varchar	Not null	图片
new_type	varchar	Not null	类型
new_translator	varchar	not null	翻译
new_content	varchar		双语内容
new_cn	varchar		中文
new_en	varchar		英文

## 14 附件表 tb\_attachment

表 3-3-14 tb\_attachment

字段名称	字段类型	约束	说明
attachment_id	int	主键，自动递增	文件编号
attachment_url	varchar	Not null	文件 url
filename	varchar	Not null	文件名
filesize	varchar	Not null	大小
c_id	int	Not null	课程 id
train_id	int	Not null	实训 id
s_id	int	Not null	学生 id
t_id	int	not null	老师 id
attachment_type	varchar		文件类型
attachment_time	varchar		上传时间

## 15 英语阅读问题表 tb\_problem

表 3-3-15 tb\_problem

字段名称	字段类型	约束	说明
problem_id	int	主键，自动递增	问题编号
author	varchar		作者
question	varchar	Not null	问题描述
result	varchar		答案
score	int		问题分数
answerA	varchar		A 选项
answerB	varchar		B 选项
answerC	varchar		C 选项
answerD	varchar		D 选项

## 2.4 本章小结

本章对英语阅读平台所需的系统业务进行研究，分析了系统的整体业务流程和业务需求，并根据需求分析进行系统架构设计，对该平台的混合式课程进行设计开发，划分系统的功能模块，并对各个模块在设计上进行了详细的介绍。然后通过需求分析得到的需求设计出本系统的数据库结构，并展示了系统实体与属性之间的简要联系。最后介绍数据库表的详细设计。

### 3 系统的实现

本文的英语阅读平台设计采用 Java 作为后端编程语言，基于 B/S 的 Java EE 架构，前端采用 Bootstrap 作为基础框架，后端采用 Spring Boot 整合 SSM(Spring, Spring MVC, MyBatis)框架开发。MySQL 作为后台数据库，Tomcat 作为 Servlet 容器进行开发，利用 IntelliJ IDEA 进行开发。以下为实现系统的介绍和部分代码案例。

#### 3.1 开发框架的整合

本平台采用采用 Spring Boot 整合 SSM(Spring, Spring MVC, MyBatis)框架开发，本平台实现严格按照 MVC 设计规范进行开发，根据业务需求和 MVC 设计规范，将系统架构分为四层，分别为持久层、业务层、表现层和视图层。其中 MyBatis 框架用于对数据库进行数据持久化工作，属于持久层（DAO 层）；调用持久层中提供的接口实现系统中业务模块的逻辑应用设计，属于业务层（Service 层）；系统的控制器调用业务层中提供的接口进行具体业务流程模块控制，此处使用 Spring MVC，属于表现层（Controller 层）；HTML、FTL 等页面展示，属于视图层（View 层）。Spring 作为 Bean 容器，将它们作为实体类对象管理和维护它们。其中整体代码结构如下图所示：

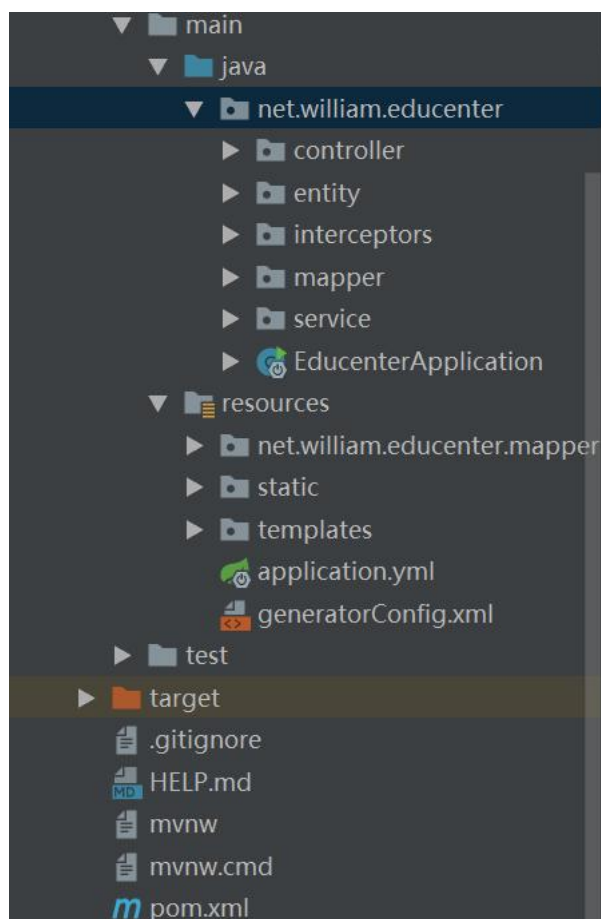


图 4-1 整体代码结构图

如图所示，系统整体代码进行前后端分离，按照 MVC 设计规范开发，持久层（Dao 层）由 entity 实体包，mapper 映射包构成。业务层（Service 层）由 service 包构成。表现层（Controller 层）由 controller 包构成。视图层（View 层）由 templates 中的 HTML 和 FTL 页面构成。generatorConfig.xml 用于配置 mybatis 逆向工程。pom.xml 作为系统依赖管理文件。application.yml 作为 Spring Boot 配置文件。

application.yml 部分核心配置代码如下：

```
<!-- Spring Boot 配置文件 -->
```

```
spring:
```

```
  datasource:
```

```
    <!-- 数据库用户名配置-->
```

```
    username: root
```

```
    <!-- 数据库驱动配置-->
```

```
    driver-class-name: com.mysql.cj.jdbc.Driver
```

```
<!-- 数据库 url 配置-->

url:

jdbc:mysql://127.0.0.1:3306/educenter?serverTimezone=UTC&useSSL=false&userU
nicode=true&characterEncoding=UTF-8

<!-- 数据库密码配置-->

password: 123456

freemarker:

<!-- freemarker 引擎页面配置-->

template-loader-path: classpath:/templates/

server:

<!-- 服务器端口配置-->

port: 8088
```

## 3.2 前台客户端的实现

经过前三章完成对系统的背景需求、功能需求和逻辑需求的分析，本节的内容将先进行实现和展示英语阅读学习平台前台客户端的详细内容与实现代码案例。

### 3.2.1 登陆注册模块

普通用户以访问英语阅读移动学习平台首页，可以浏览部分英语阅读文章，课程信息等，需要注册登录学习平台才能开放课程等功能。注册需要用户输入账号、密码，Email，手机号，姓名，注册成功后，输入账号密码，选择角色后通过验证即可登录成功。

(1)登陆注册模块效果图





The login interface features a light gray background with a central white box. At the top, the title '登录' (Login) is displayed in a large, bold, black font. Below the title, there are two input fields: the first is labeled '账号' (Account) and contains the text '1600502215'; the second is labeled '密码' (Password) and contains a series of dots. Below these fields, there are three radio buttons for user type selection: '学生' (Student), '老师' (Teacher), and '企业' (Enterprise), with '学生' being selected. To the left of the radio buttons is a checkbox labeled '记住密码' (Remember password), and to the right is a link labeled '忘记密码?' (Forgot password?). A large blue button with the text '登录' (Login) is positioned below the input fields. At the bottom of the white box, there is a link that says '云智教育新用户? 创建一个账户。' (New user? Create an account).

图 4-2 登录效果图



The registration interface has a light gray background with a central white box. The title '注册' (Registration) is at the top in a large, bold, black font. Below the title, there are five input fields: '账号' (Account) with the placeholder '新建账号' (New account), 'Email' with the placeholder 'email', '密码' (Password) with the placeholder '设置密码' (Set password), '手机号' (Mobile number) with the placeholder '手机号' (Mobile number), and '昵称' (Nickname) with the placeholder '你的昵称' (Your nickname). Below these fields, there are three radio buttons for user type selection: '学生' (Student), '老师' (Teacher), and '企业' (Enterprise), with '学生' being selected. To the left of the radio buttons is a checkbox labeled '记住密码' (Remember password), and to the right is a link labeled '忘记密码?' (Forgot password?). A large blue button with the text '注册' (Register) is positioned below the input fields. At the bottom of the white box, there is a link that says '已有账户? 直接登录。' (Already have an account? Log in directly).

图 4-3 注册效果图

## (2) 登陆注册模块实现代码案例

1. 登录模块分为学生，教师，企业登录部分和后台管理员用户登录部分。

因为这些角色登录功能相似，故以学生用户登录为例。在登录时，用户需要输入正确的账号和密码，选择正确角色才可以正常登录。输入的账号密码等信息通过 session 获取到后台，查询数据库验证身份。验证通过后，返回对应的跳转页面链接进行跳转。以下是登录的业务流程代码：

educenter\controller\StudentController.java

//登录部分后端代码:

```
@RequestMapping("/studentlogin.do")
```

```
public String login(HttpSession session) {
```

```
//登录页面获取登录信息
```

```
String useraccount = (String) session.getAttribute("useraccount");
```

```
String password = (String) session.getAttribute("password");
```

```
StudentExample studentExample = new StudentExample();
```

```
StudentExample.Criteria criteria = studentExample.createCriteria();
```

```
if (useraccount != null && useraccount.trim() != "" && password != null  
&& password.trim() != "") {
```

```
//询数据库确定账号密码是否正确
```

```
criteria.andSAccountEqualTo(useraccount);
```

```
criteria.andSPswEqualTo(password);
```

```
List<Student> students =
```

```
studentService.selectByExample(studentExample);
```

```
if (students!=null && students.size()>0) {
```

```
Student student = students.get(0);
```

```
session.setAttribute("sId", student.getId());
```

```
System.out.println("登录成功");
```

```
return "student-index";
```

```
} else {
```

```
System.out.println("账号或密码有误");
```

```
}
```

```
}
```

```
return "signin";
```

```
}
```

2. 注册模块分为学生，教师和企业部分，用户点击对应角色，输入账号，密码，邮箱，用户名和手机号码进行注册。通过 session 获取前台注册信息，查询数据库验证账户信息是否存在，验证通过即可注册新账户，插入新账户到数据库中。

educenter\controller\StudentController.java:

//注册部分后端代码

@RequestMapping("/studentregister.do")

public String register(Model model, HttpSession session) {

//前台获取信息

String newaccount = (String) session.getAttribute("newaccount");

String newpassword = (String) session.getAttribute("newpassword");

String newemail = (String) session.getAttribute("newemail");

String newusername = (String) session.getAttribute("newusername");

String newtel = (String) session.getAttribute("newtel");

StudentExample studentExample = new StudentExample();

StudentExample.Criteria criteria = studentExample.createCriteria();

if (newaccount != null && newaccount.trim() != "" && newpassword != null && newpassword.trim() != "") {

criteria.andSAccountEqualTo(newaccount);

criteria.andSPswEqualTo(newpassword);

List<Student> students =

studentService.selectByExample(studentExample);

if (students!=null && students.size()>0) {

//验证是否已存在该账户

System.out.println("已存在该账户");

} else {

Student student = new Student();

student.setsAccount(newaccount);

student.setsPsw(newpassword);

```

        student.setEmail(newemail);
        student.setName(newusername);
        student.setTel(newtel);
        studentService.insertstudent(student);    //注册新账户
        System.out.println("注册成功");
        return "/signin";    //返回登录
    }
}
return "/signup";    //验证不通过返回注册
}

```

### 3.2.2 课程模块

课程模块是学生用户和教师用户进行课程学习交流的模块，在该模块中，学生可以选择心仪课程进行学习，包括下载学习资源，阅读英语文章，发表评论交流学习。老师会管理课程，包括新建课程，上传学习资源，发布学习任务，修改作业并评分等。

#### (1)课程模块效果图

##### 1. 学生课程模块效果图：



图 4-4 我的课程模块效果图

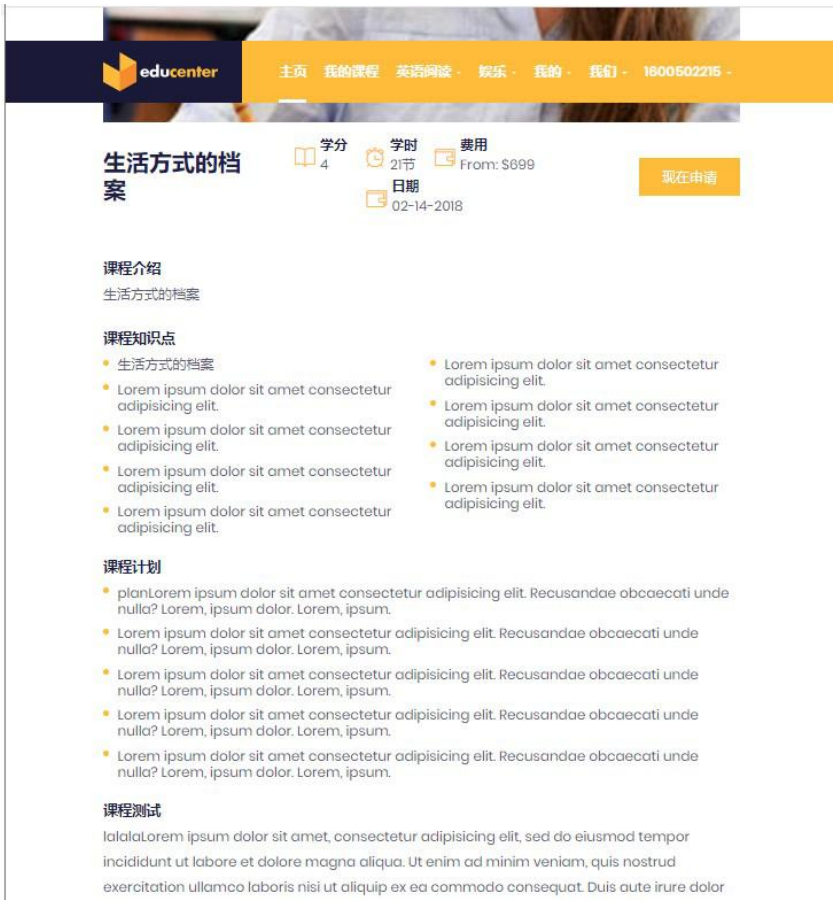


图 4-5 课程介绍效果图

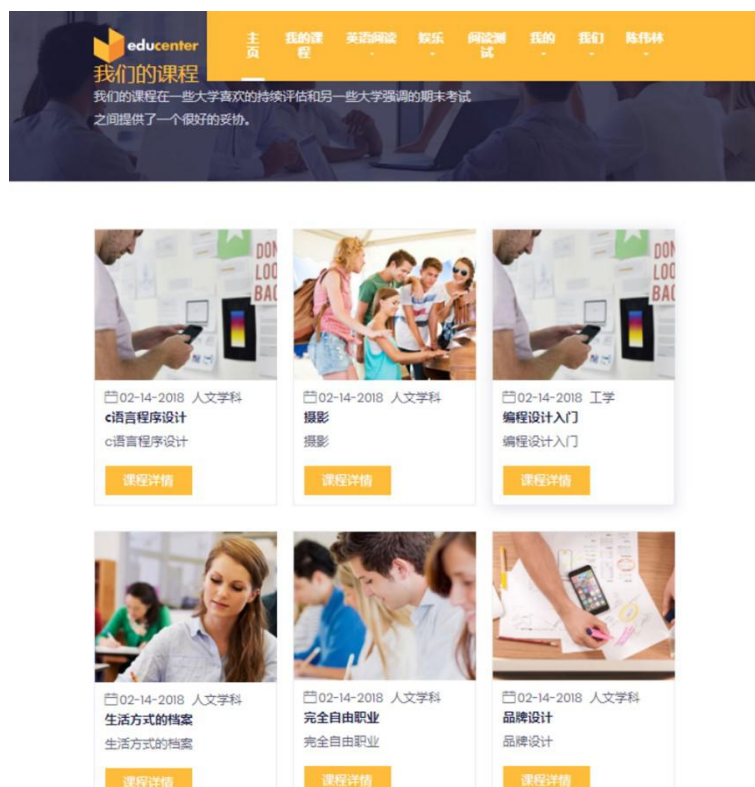


图 4-6 全部课程效果图

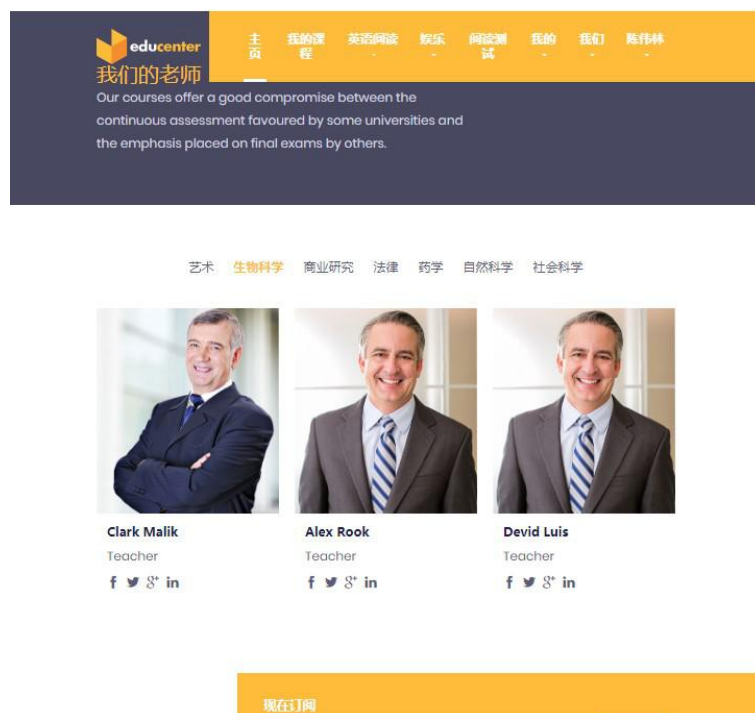


图 4-7 全部课程教师效果图

## 2. 教师课程模块效果图:

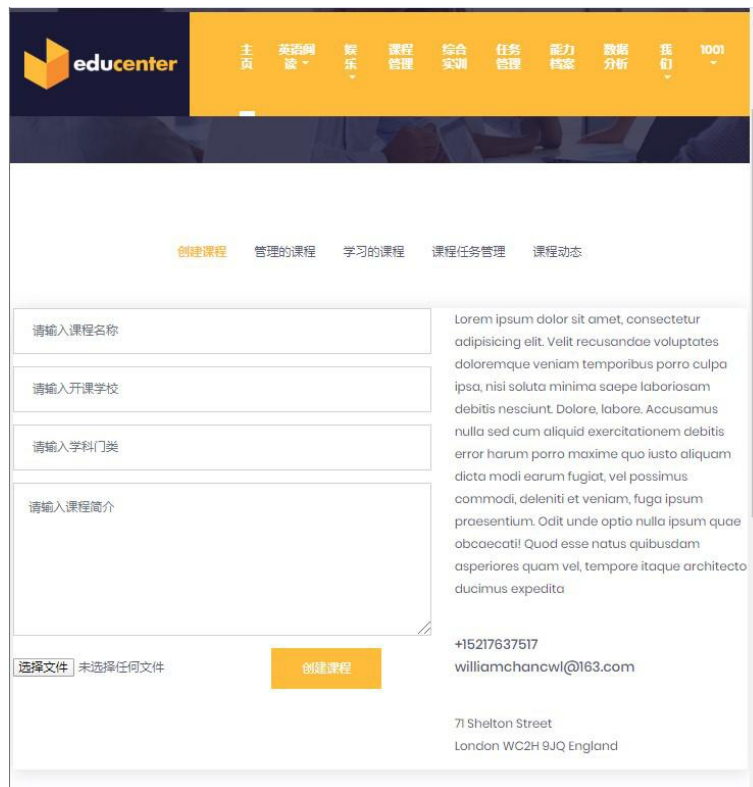


图 4-8 创建课程效果图



图 4-9 管理课程效果图

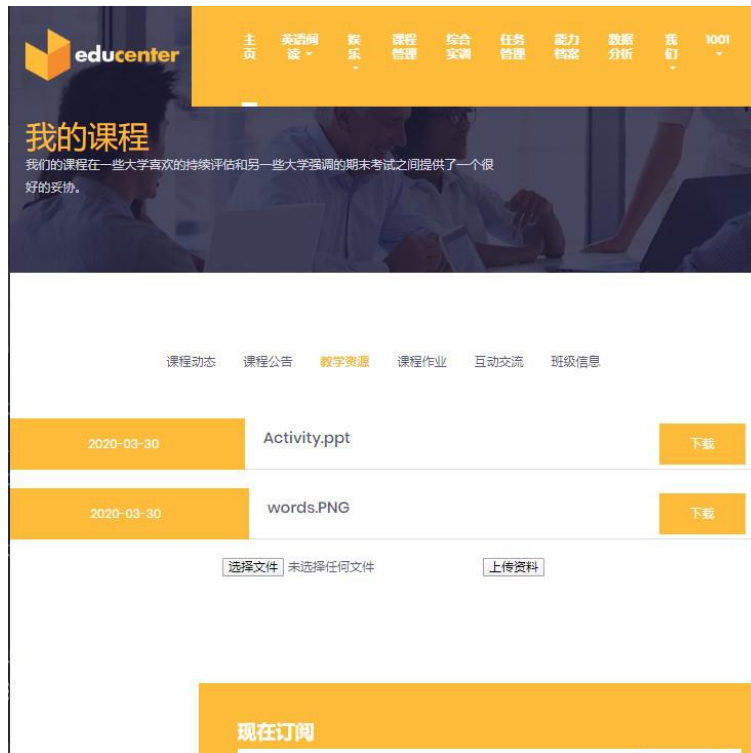


图 4-10 教学资源效果图

## (2)课程模块实现代码案例

### 1.课程展示代码案例：

后台根据登录用户 id 查询对应的 courses 置入 Model 中，前台直接查询 courses 列表即可。同理，根据 courseId 查询对应课程 course 置入 model 中，前端代码使用 freemarker 的<#list>关键词进行 Model 的 course 遍历。课程展示后端代码如下：

//后台根据课程 Id 进行查询

```
@RequestMapping("/student-course-single")

public String studentCourseSingle(Model model, Integer courseId) {

    CourseExample CourseExample = new CourseExample();

    CourseExample.Criteria criteria = CourseExample.createCriteria();

    criteria.andCIdEqualTo(courseId);

    List<Course> Courses =

courseService.selectByExample(CourseExample);

    if(Courses!=null && Courses.size()>0) {
```



```

        model.addAttribute("course", Courses.get(0));
    }

    CourseExample CourseExample1 = new CourseExample();
    CourseExample.Criteria criteria1 = CourseExample1.createCriteria();
    List<Course> Courses1 = courseService.selectByExample(CourseExample1);
    model.addAttribute("courses", Courses1);
    return "student/course-single";
}

```

### 3.2.3 文章阅读模块

英语文章阅读模块是本学习平台的主要功能模块，学生或教师用户在登录后，点击查阅主页的英语阅读文章或娱乐文章，这些文章可以进行双语阅读，或者选择英语阅读完再选择中文阅读，对照阅读提高英语阅读能力。

#### (1) 文章阅读模块效果图



图 4-11 阅读文章效果图 1



图 4-12 阅读文章效果图 2



图 4-13 阅读文章效果图 3



图 4-14 阅读文章效果图 4

## (2)文章阅读实现代码案例

### 1. 获取阅读文章详情后台代码案例:

选择文章栏目后,通过文章的类别查询到对应栏目文章列表,点击列表中的文章,通过文章 Id 查询数据库中的对应的文章传到 model,从而在前台展示出来。前台内容展示分为双语,英语,中文,使用 data-filter 划分内容, data-ref 绑定对应内容,点击对应模块展示对应语言的文章。选择文章栏目后,通过文章的类别查询到对应栏目文章列表,点击列表中的文章,通过文章 Id 查询数据库中的对应的文章传到 model,从而在前台展示出来。文章详情后台代码如下:

```
educenter\controller\NewController.java

@RequestMapping("/student-new-single")

public String studentNew(Model model, Integer newId) {

    NewExample newExample = new NewExample();

    NewExample.Criteria criteria = newExample.createCriteria();

    criteria.andNewIdEqualTo(newId);

    List<New> news = newService.selectByExample(newExample);

    if(news!=null && news.size()>0) {

        model.addAttribute("new", news.get(0));

    }

    NewExample newExample1 = new NewExample();

    List<New> news1 = newService.selectByExample(newExample1);

    model.addAttribute("news", news1);

    return "student/new-single";

}
```

### 3.2.4 阅读测试模块

英语阅读测试模块是本学习平台的学生练习模块,学生用户在登录后,点击主页的阅读测试模块,即可进入阅读测试。在规定时间内答完阅读测试题,答完后系统自动批改,得出分数显示到前台。

#### (1)阅读测试模块效果图

阅读测试效果图:

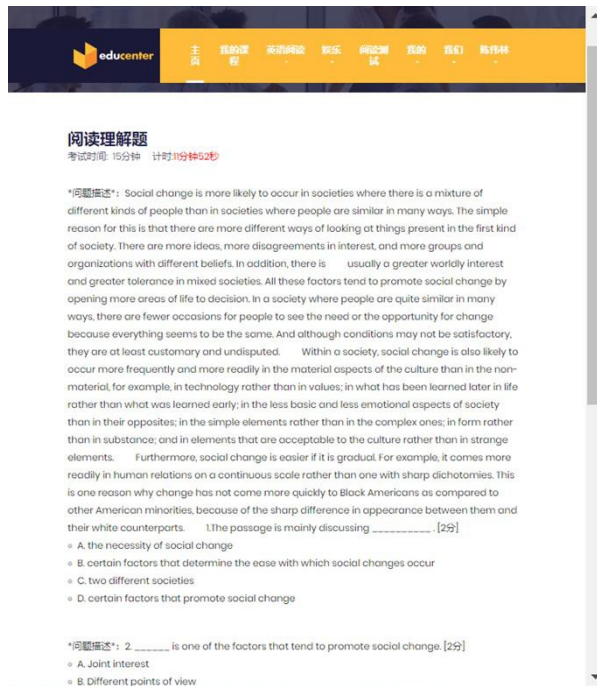


图 4-15 阅读测试模块效果图

测试结果效果图：

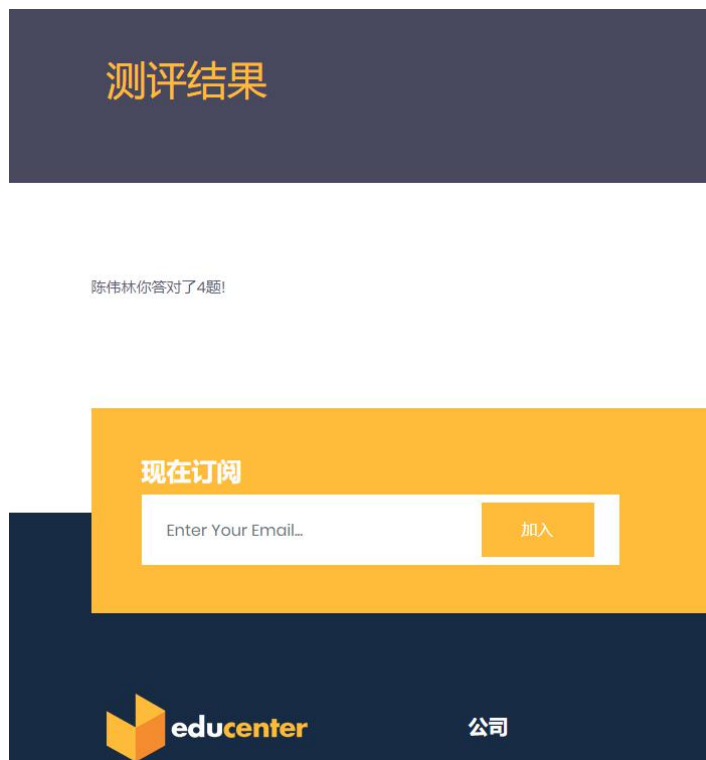


图 4-16 测试结果效果图

## (2) 阅读测试模块实现代码案例

阅读测试前端通过后端查询数据库得到的问题 model，展示阅读理解题目，

用户进入测试后，计时做题，使用 JavaScript 设置计时器。在规定时间内答完题目后提交到后端进行批改，返回测试结果到前台。阅读测试后端查询阅读测试题目并添加到 model 中，从而在前端展示题目描述。考试答题后根据题目的答案进行修改展示结果。部分阅读测试后端代码案例如下：

```
educent/controller/coursecontroller.java:

@RequestMapping("/course-test")
//查询阅读测试题目

    public String CourseTest(Model model) {

        ProblemExample problemExample = new ProblemExample();

        List<Problem>                problems                =
problemService.selectByExample(problemExample);

        model.addAttribute("problems", problems);

        return "student/course-test";

    }

    @RequestMapping("/ExamServlet")

    public String ExamServlet(Model model, HttpServletRequest req) {

        //批改题目

        ProblemExample problemExample = new ProblemExample();

        List<Problem>                problems                =
problemService.selectByExample(problemExample);

        int sum = 0;

        for(Problem problem : problems) {

            if(req.getParameter(String.valueOf(problem.getProblemId())).equals(problem.getResult())) {

                sum++;

            }

        }

        //记录答对的题目数 sum

        model.addAttribute("sum", sum);
```

```
return "student/result";  
}
```

### 3.2.5 实训模块

学生进入实训模块进行技能实际训练，包括下载实训资源，阅读相关文章，交流学习。老师会管理实训，包括新建实训，上传实训资源，发布实训任务，检验实训并评分等。

#### (1)实训模块效果图



图 4-17 学生实训效果图

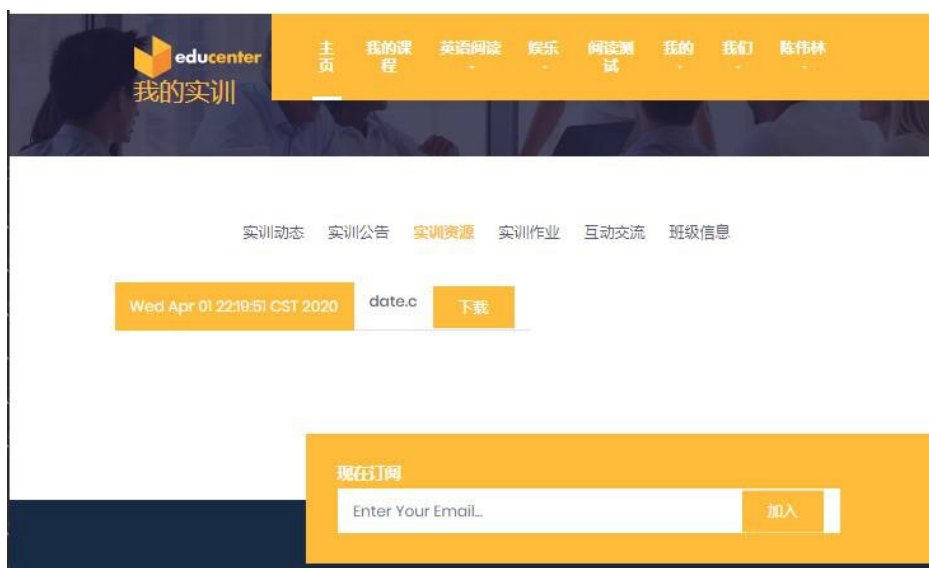


图 4-18 学生实训效果图

创建实训

管理的实训

实训任务管理

实训动态

请输入实训名称

请输入实训时长

请输入开始时间

请输入结束时间

请输入实训内容简介

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Volit recusandae voluptates doloremque veniam temporibus porro culpa ipsa, nisi soluta minima saepe laboriosam debitis nesciunt. Dolor, labore. Accusamus nulla sed cum aliquid exercitationem debitis error harum porro maximo quo iusto aliquam dicta modi earum fugiat, vel possimus commodi, deleniti et veniam, fuga ipsum praesentium. Odit unde optio nulla ipsum quae obcaecati! Quod esse natus quibusdam asperiores quam vel, tempore itaque architecto ducimus expedita

+15217637517  
williamchancwl@163.com

71 Shelton Street  
London WC2H 9JQ England

选择文件

未选择任何文件

创建实训

图 4-19 实训管理效果图 1

综合实训管理

我们的课程在一些大学喜欢的持续评估和另一些大学强调的期末考试之间提供了一个很好的妥协。

实训动态

实训公告

实训资源

实训任务

互动交流

班级信息

发布作业

现在订阅

图 4-20 实训管理效果图 2



图 4-21 实训任务效果图

## (2)实训模块代码案例

前端根据后台查询到的实训传到 model 的 trains 进行展示，使用<#list>列表循环展示，使用\${}获取实训信息。实训学习通过实训 Id 查询到实训具体信息，实训文件课件等。学生用户查询个人实训，进入实训学习，包括查看实训信息，上传下载实训文件课件等。同理，教师用户实训管理通过查询个人管理的实训，进入实训管理，发布实训任务，上传下载实训文件课件等。其中部分的实训后端代码代码如下：

```
@RequestMapping("/trainlearn")
//实训学习后台处理

public String TrainLearn(Model model, Integer trainId) {
    //根据 trainId 查询实训

    TrainExample TrainExample = new TrainExample();
    TrainExample.Criteria criteria = TrainExample.createCriteria();
    criteria.andTrainIdEqualTo(trainId);
    List<Train> Trains = trainService.selectByExample(TrainExample);
    if(Trains!=null && Trains.size()>0) {
        model.addAttribute("train", Trains.get(0));
    }

    TrainExample TrainExample1 = new TrainExample();
    TrainExample.Criteria criteria1 = TrainExample1.createCriteria();
    List<Train> Trains1 = trainService.selectByExample(TrainExample1);
    model.addAttribute("trains", Trains1);
}
```



```

//根据实训 Id 查询实训文件课件
AttachmentExample attachmentExample = new AttachmentExample();
AttachmentExample.Criteria criteria2 =
attachmentExample.createCriteria();
criteria2.andTrainIdEqualTo(trainId);
List<Attachment> attachments =
attachmentService.selectByExample(attachmentExample);
model.addAttribute("attachments", attachments);
return "student/trainlearn";
}

```

```

@RequestMapping("/train-admin")
public String TrainAdmin(Model model, Integer trainId) {
//实训管理后台处理
TrainExample TrainExample = new TrainExample();
TrainExample.Criteria criteria = TrainExample.createCriteria();
criteria.andTrainIdEqualTo(trainId);
List<Train> Trains = trainService.selectByExample(TrainExample);
if(Trains!=null && Trains.size()>0) {
model.addAttribute("train", Trains.get(0));
}
TrainExample TrainExample1 = new TrainExample();
TrainExample.Criteria criteria1 = TrainExample1.createCriteria();
List<Train> Trains1 = trainService.selectByExample(TrainExample1);
model.addAttribute("trains", Trains1);
AttachmentExample attachmentExample = new AttachmentExample();
AttachmentExample.Criteria criteria2 =
attachmentExample.createCriteria();
criteria2.andTrainIdEqualTo(trainId);
List<Attachment> attachments =

```

```
attachmentService.selectByExample(attachmentExample);

//附件文件查询

model.addAttribute("attachments", attachments);

return "teacher/train-admin";

}
```

### 3.2.6 消息模块

消息模块包括用户联系管理员的联系我们模块，系统通知模块，用户交流信息通知等信息模块。实现信息模块可以达到学生，教师，教务管理员，系统管理员的日常沟通交流，便于课程学习交流，教务管理交流及系统管理交流等。通过绑定的用户 Id 查询接收发送信息。

#### (1)消息模块效果图

**联系我们**

你还有其他问题吗?别担心, 没有什么愚蠢的问题。请填写下面的表格, 我们会尽快给您答复。

william

williamchanwl@163.com

hello

管理员您好

发送信息

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Velit recusandae voluptates doloreque veniam temporibus porro culpa ipsa, nisi soluta minima saepe laboriosam debitis nesciunt. Dolore, labore. Accusamus nulla sed cum aliquid exercitationem debitis error harum porro maxime quo iusto aliquam dicta modi earum fugiat, vel possimus commodi, deleniti et veniam, fuga ipsum praesentium. Odit unde optio nulla ipsum quae obcaecati! Quod esse natus quibusdam asperiores quam vel, tempore itaque architecto ducimus expedita

图 4-22 联系我们效果图

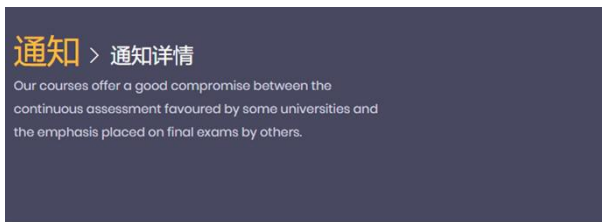


图 4-23 系统通知效果图

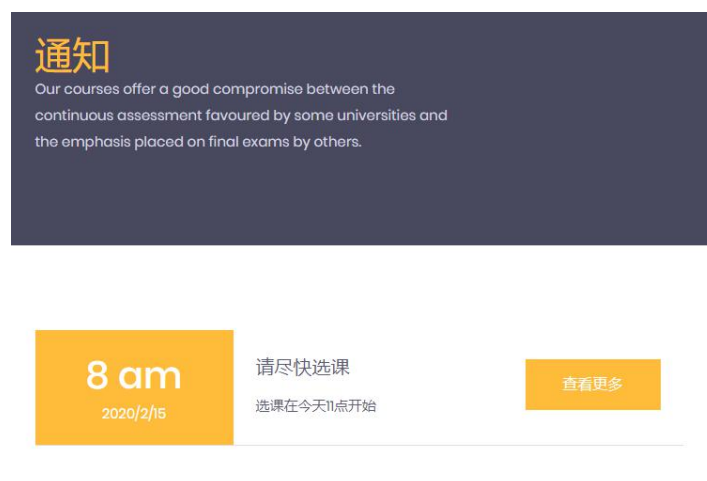


图 4-24 通知详情

## (2)消息模块代码案例

前端根据后台查询到的通知 notice 进行展示，系统通知后端代码案例展示：

```
@RequestMapping("/student-notice")
public String getNotice(Model model, HttpSession session) {
    //根据学生 ID 进行查询系统通知信息
    Integer sId = (Integer)session.getAttribute("sId");
    NoticeExample noticeExample = new NoticeExample();
    NoticeExample.Criteria criteria = noticeExample.createCriteria();
    if(sId != null) {
```

```

        criteria.andSIdEqualTo(sId);

        List<Notice>                notices                =
noticeService.selectByExample(noticeExample);

        model.addAttribute("notices", notices);
    }

    return "student/notice";
}

```

```

@RequestMapping("/student-notice-single")
public String NoticeSingle(Model model, Integer noticeId) {
    //根据通知 ID 进行查询

    NoticeExample noticeExample = new NoticeExample();
    NoticeExample.Criteria criteria = noticeExample.createCriteria();
    criteria.andNoticeIdEqualTo(noticeId);

    List<Notice>                notices                =
noticeService.selectByExample(noticeExample);

    if(notices != null && notices.size()>0) {
        model.addAttribute("notice", notices.get(0));
    }

    return "student/notice-single";
}

```

### 3.2.7 其他模块

英语阅读平台客户端除了上述的功能模块，还有其他的功能模块。如博客模块，活动模块，成绩模块，档案模块。由于这些模块实现代码案例类似于上述的功能模块，实现的代码案例就不赘述了，以下仅展示模块的效果图。

(1)博客模块效果图



图 4-25 博客模块效果图 1

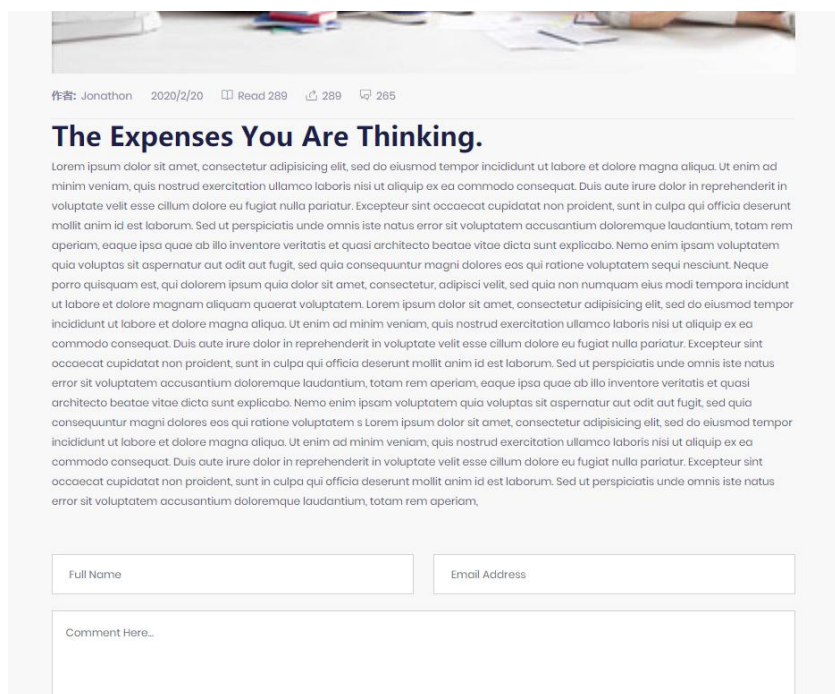


图 4-26 博客系统效果图 2

## (2)活动模块效果图



图 4-27 活动模块效果图 1

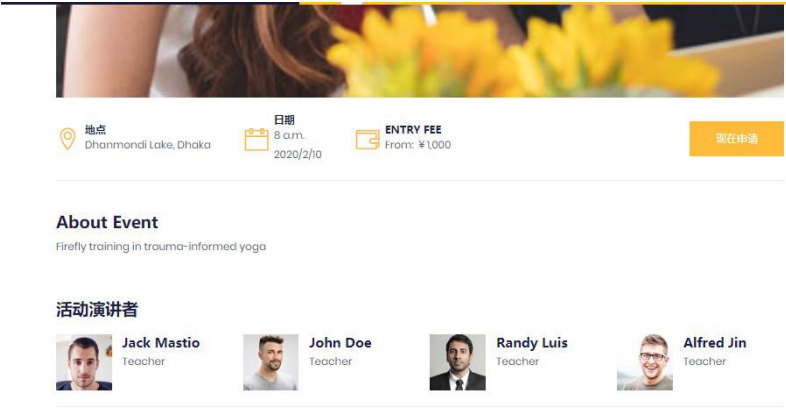


图 4-28 活动模块效果图 2

(3)成绩模块效果图



图 4-29 成绩模块效果图

### 3.3 后台管理端的实现

本系统后台管理的实现分为两大部分，一是企业院校用户使用的后台管理端，另一端就是平台管理员使用的后台管理端系统。后台管理端系统不会被学生教师用户所使用，但是其担负着整个系统的数据运营与维护的工作，本节的内容就是实现和展示这两个后台管理端系统的功能和代码。

#### 3.3.1 企业院校后台管理端实现

企业院校后台包括对学生教师管理，课程管理，实训管理，文章阅读管理，阅读测试管理，消息通知管理，博客活动管理，院系信息管理，资讯管理，文件管理，以及系统的监管和分析。

##### (1) 页面开发案例

后台管理主要是信息管理，包括增加，删除，查询，更改，文件上传下载，信息导入导出，分页和显示条目处理等操作，以学生用户管理案例为例：

##### 1. 学生用户管理

学生用户管理包括分页和模糊查询学生信息，新增学生用户，单量或批量删除学生用户，更改学生用户信息等功能，这些功能通过点击对应按钮会弹出操作框用于输入或者询问。通过 table 展示用户表信息，表的每条记录前包含标记，用于标记该记录进行删除更改等操作。其中用户管理部分前端展示信息代码如下：

studenttable.ftl:

```
<#list students as student>
```

```
<tr>
```

```
    <td><input                type="checkbox"                name="checkbox_student"
value="\${student.sId}"/></td>
```

```
    <td>\${student.sAccount}</td>
```

```
    <td>\${student.sPsw}</td>
```

```
    <td>\${student.sName}</td>
```

```
    <td>\${student.sEmail}</td>
```

```
    <td>\${student.sTel !"null"}</td>
```

```

<td>${student.sSex !"null"}</td>
<td>${student.sBirthday !"null"}</td>
<td>${student.schId !"null"}</td>
<td>${student.dId !"null"}</td>
<td>${student.mId !"null"}</td>
<td>${student.classId !"null"}</td>
</tr>
</#list>

```

## (2)后台数据交互案例

学生用户管理后台数据交互包括查询学生，增加学生用户，删除学生用户，更改学生用户信息等功能，这些数据交互通过查询数据库传到 **Model** 传到前端。点击对应按钮会弹出操作框用于输入或者询问，通过属性名获取前端数据，进而在后端处理。

```

@RequestMapping("/organization-studenttable")
public String getOrganizationStudentTable(Model model) {
    //企业院校教务展示学生页面

    StudentExample studentExample = new StudentExample();

    List<Student>                students                =
studentService.selectByExample(studentExample);

    model.addAttribute("students", students);

    return "organization/studenttable";
}

@RequestMapping("/deleteStudent")
public String deleteStudent(Integer[] sIds) {
    //根据学生 ID 进行删除

    for(Integer sid : sIds) {
        if (sid != null) {

            StudentExample StudentExample = new StudentExample();

            StudentExample.Criteria                criteria                =

```



```

StudentExample.createCriteria();

        criteria.andSIdEqualTo(sid);

        studentService.deletestudent(StudentExample);

    }

}

System.out.println("删除成功");

return "redirect:/studenttable";

}

@RequestMapping("/insertStudent")

public String InsertStudent(Model model, String newaccount, String
newpassword, String newemail, String newtel, String newusername, String newsex,
String newbirthday) {

    //添加学生用户

    StudentExample studentExample = new StudentExample();

    StudentExample.Criteria criteria = studentExample.createCriteria();

    if (newaccount != null && newaccount.trim() != "" && newpassword !=
null && newpassword.trim() != "") {

        criteria.andSAccountEqualTo(newaccount);

        criteria.andSPswEqualTo(newpassword);

        List<Student>                Students                =
studentService.selectByExample(studentExample);

        if (Students!=null && Students.size()>0) {

            System.out.println("已存在该账户");

        } else {

            Student Student = new Student();

            Student.setsAccount(newaccount);

            Student.setsPsw(newpassword);

            Student.setsEmail(newemail);

            Student.setsName(newusername);

            Student.setsTel(newtel);

```

```

        Student.setsSex(newsex);

        Student.setsBirthday(newbirthday);

        studentService.insertstudent(Student);

        System.out.println("注册成功");

    }

}

return "redirect:/studenttable";

}

@RequestMapping("/updateStudent")

public String updateStudent(Model model, String newaccount, String
newpassword, String newemail, String newtel, String newusername, String newsex,
String newbirthday) {
    //更改学生用户信息

    StudentExample studentExample = new StudentExample();

    StudentExample.Criteria criteria = studentExample.createCriteria();

    if (newaccount != null && newaccount.trim() != "" && newpassword !=
null && newpassword.trim() != "") {

        Student Student = new Student();

        Student.setsAccount(newaccount);

        Student.setsPsw(newpassword);

        Student.setsEmail(newemail);

        Student.setsName(newusername);

        Student.setsTel(newtel);

        Student.setsSex(newsex);

        Student.setsBirthday(newbirthday);

        studentService.updatestudent(Student, studentExample);

        System.out.println("更新成功");

    }

    return "/studenttable";

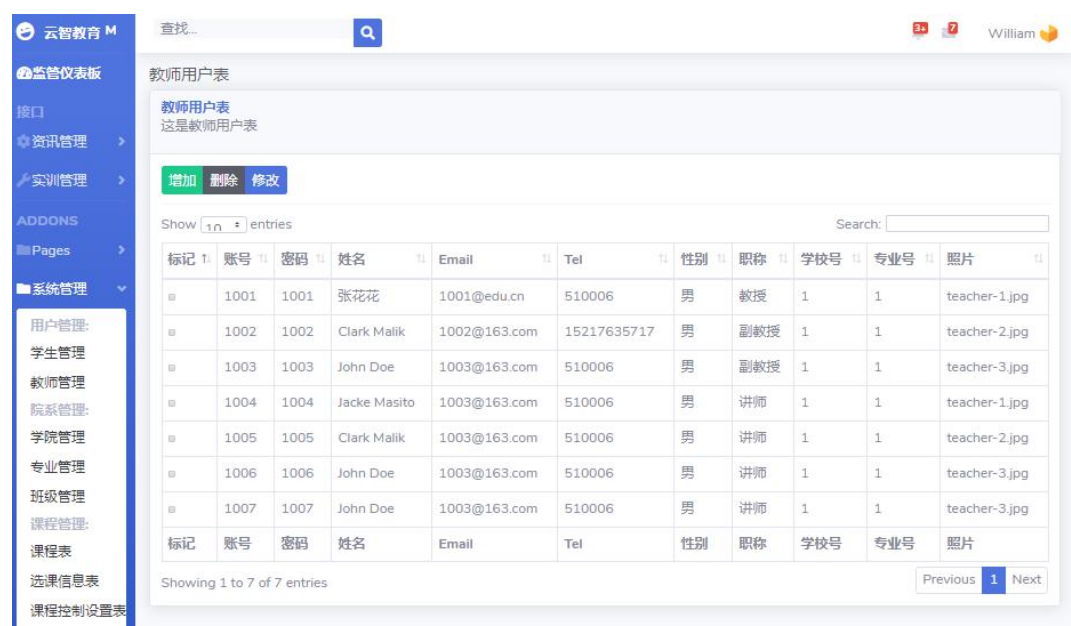
}

```

(3)页面展示



图 4-30 后台管理端页面展示图 1



4-31 后台管理端页面展示图 2

图



图 4-32 后台管理端页面展示图 3



图 4-33 后台管理端页面展示图 4

### 3.3.2 系统管理员后台管理端页面实现

系统管理员拥有英语阅读移动学习平台的最高权限，通过管理员登录页面验证后，管理员可以对平台的各数据库表进行管理，同时可以监控系统和对数据进行分析。下面以系统管理员管理企业院校用户信息为例。

#### (1) 页面开发案例

系统管理员登录到后台管理系统，进行企业院校用户管理，前端页面主要显示该类用户信息，部分代码如下：

#### 1. Organizationtable.ftl

```

<#list organizations as organization>

<tr>

    <td><input                type="checkbox"                name="checkbox_student"
value="${organization.oId}"/></td>

    <td>${organization.oAccount}</td>
    <td>${organization.oPsw}</td>
    <td>${organization.oName}</td>
    <td>${organization.oEmail}</td>
    <td>${organization.oTel !"null"}</td>
    <td>${organization.oTitle !"null"}</td>
    <td>${organization.oDesc !"null"}</td>
    <td>${organization.oLocation !"null"}</td>

</tr>

</#list>

```

## (2)后台数据交互案例

企业院校用户管理后台包括增加，删除，更改，查询等操作，以查询所有企业院校用户为例：

@RequestMapping("/organizationtable")

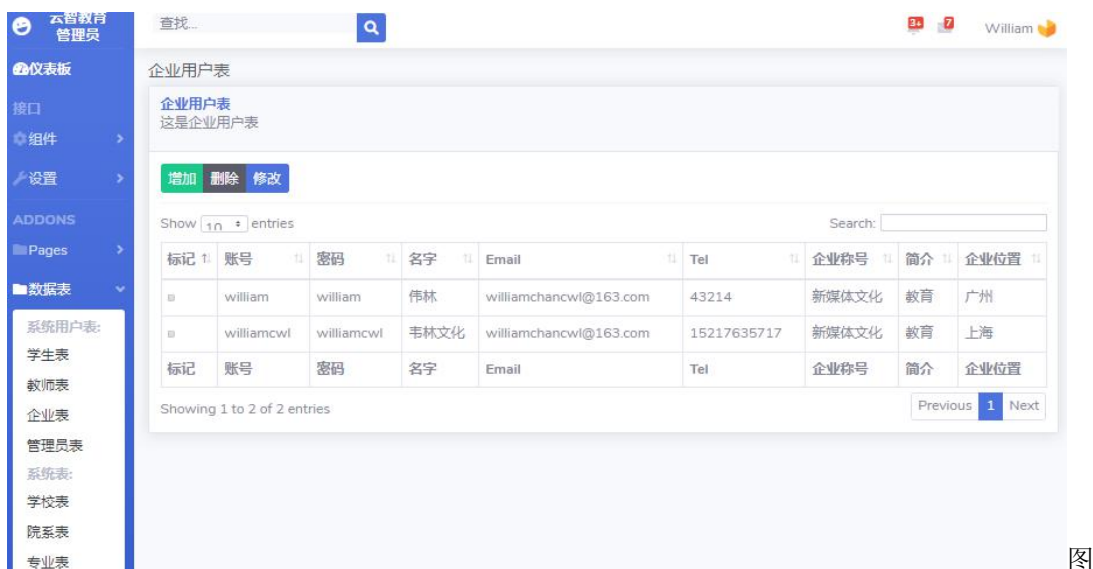
```

public String getOrganizationTable(Model model) {
    //查询企业院校用户

    OrganizationExample OrganizationExample = new OrganizationExample();
    List<Organization>                organizations                =
OrganizationService.selectByExample(OrganizationExample);
    model.addAttribute("organizations", organizations);
    return "superuser/organizationtable";
}

```

## (3)页面展示



4-34 系统管理员效果图 1



图 4-35 系统管理员效果图 2

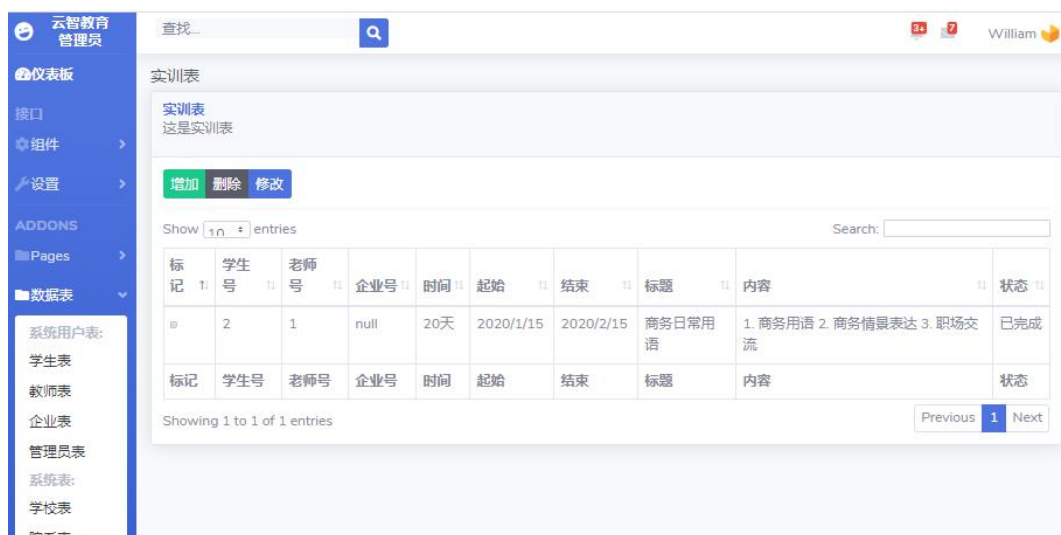


图 4-36 系统管理员效果图 3

### 3.4本章小结

本章介绍了英语阅读移动学习平台的实现过程，主要介绍了前台客户端系统各模块以及后台管理系统各模块的实现，介绍了系统整体代码架构与相关说明，给出了系统模块的流程图以及编码过程中的部分核心代码与思路分析，对系统主要功能模块的实现界面的效果展示。

## 4 系统测试

软件测试工作衡量着应用程序的编码质量过不过关，决定着该应用程序能否成功的达到需求和设计的目标。软件测试的过程是充斥着开发的整个阶段的，目的是使得软件在开发的过程中能够说明出软件的规则和程序内部结构，本章内容主要是进行软件测试中的系统测试，设计若干测试用例，使用测试用例运行系统，来得出测试结果判断软件是否存在错误。

### 4.1 测试方法

本次系统测试分为两步，第一步使用测试用例来检测系统功能，主要使用黑盒测试，也即是功能测试。黑盒测试不考虑程序的内部结构，主要的测试程序的外部特征，这样更容易发现用户操作时会发生的错误问题，如：输入表单的错误，页面跳转的错误，页面层级交互的错误等等。这样更加贴近用户使用的特点。测试方法是列出测试功能点，设计用户操作，根据所涉及的用户操作得出程序输出的结果。第二步检测客户端的兼容性，主要是测试平台的浏览器兼容性和不同屏幕大小页面兼容性。

### 4.2 单元测试

系统在开发过程中使用了 Java 的单元测试框架 Junit。单元测试在测试过程中可能可以发现该模块或功能在设计上的问题或实现上的缺陷，只要发现错误，即可马上将其修复或改进，以保证该模块在功能上的可行性和正确性。单元测试在一定程度上降低了测试和开发的成本，提高开发的效率。下面将通过简单列举部分功能模块的单元测试用例。

#### (1) 用户注册

用户注册模块主要包含了用户填写信息的校验和校验后用户信息的保存。利用单元测试对注册模块进行注册校验和信息保存的正确性，表 5-1 为用户注册的测试用例。



表 5-1 用户注册测试用例

测试用例编号	测试目的和步骤	前提条件	期望结果	实际测试结果及结论
1	<p>[目的]测试注册（符合校验条件的）</p> <p>用户账号</p> <p>[流程]</p> <p>1)填写账号 1600502215</p> <p>2)填写密码和确认密码为 1600502215</p> <p>3)填写注册邮箱 williamchancwl@163.com</p> <p>4)填写手机号 15217637517，昵称 william，选择角色“学生”，输入正确验证码</p> <p>5)获取账号与密码</p>	<p>1)账号 1600502215 为未注册账号</p> <p>2)密码和确认密码均一致</p> <p>3)邮箱手机号填写正确，选择学生角色</p> <p>4)验证码输入正确</p>	注册成功	注册成功 说明正确注册一个未注册账号成功
2	<p>[目的]测试注册已注册用户账号</p> <p>[流程]</p> <p>1)填写账号 1600502215</p> <p>2)填写密码和确认密码为 1600502215</p> <p>3)选择学生角色</p>	<p>1)账号 1600502215 为已注册账号</p> <p>2)密码和确认密码均一致</p>	提示账号已存在，终止注册操作	提示账号已存在 终止注册操作 说明已注册用户不可以再注册
3	<p>[目的]测试验证码的有效性</p> <p>[流程]</p> <p>1)填写账号 1600502215</p> <p>2)填写密码和确认密码为 1600502215</p> <p>3)输入错误验证码</p>	<p>1)验证码输入错误</p> <p>2)密码和确认密码均一致</p>	提示验证码输入错误，终止注册操作	提示验证码输入错误 说明验证码有效性

## (2) 用户登录

用户登录模块主要包含了用户信息的校验和登录状态的保存。用户登录的测试用例如下表：

表 5-2 用户登录测试用例

测试用例编号	测试目的和步骤	前提条件	期望结果	实际测试结果及结论
4	<p>[目的]测试登录（符合校验条件的）</p> <p>用户账号</p> <p>[流程]</p> <p>1)填写账号 1600502215</p> <p>2)填写密码和确认密码为 1600502215</p>	<p>1)账号 1600502215 为已注册账号</p> <p>2)密码为正确密码</p>	登录成功	<p>登录成功</p> <p>说明已注册用户登录成功</p>
5	<p>[目的]测试登录未注册用户账号</p> <p>[流程]</p> <p>1)填写账号 1600502201</p> <p>2)填写密码和确认密码为 1600502201</p>	<p>账号 1600502201 为未注册账号</p>	提示账号或密码错误，终止登录	<p>账号为未注册账号 说明未注册用户登录失败</p>
6	<p>[目的]测试登录（填写密码错误的）</p> <p>用户账号</p> <p>[流程]</p> <p>1)填写账号 1600502215</p> <p>2)填写密码和确认密码为 errorpassword</p>	<p>1)账号 1600502215 为已注册账号</p> <p>2)密码为错误密码</p>	提示账号或密码错误，终止登录	<p>密码错误 说明密码错误登录失败</p>

### （3）教师课程管理

教师登录成功后，在课程管理模块中包含创建课程，管理课程，课程资源上传，课程作业发布等功能。一下列举了创建课程，课程资源上传和课程作业发布的测试用例。利用单元测试验证课程管理模块功能的有效性与正确性。课程管理的测试用例如下：

表 5-3 课程管理测试用例

测试用例编号	测试目的和步骤	前提条件	期望结果	实际测试结果与结论
7	<p>[目的]测试新增（符合条件的）课程</p> <p>[流程]</p> <p>1)填写新增课程的相关信息</p> <p>2)提交</p>	课程为未新增课程	创建课程成功	<p>创建新课程成功</p> <p>说明创建课程成功</p>
8	<p>[目的]测试课程资源上架</p> <p>[流程]</p> <p>1)选择课程资源</p> <p>2)上传</p>	课程资源是没有上传过的资源	课程资源上传成功	<p>课程新增资源</p> <p>说明资源上传成功</p>
9	<p>[目的] 测试作业发布</p> <p>[流程]</p> <p>1)输入作业发布信息</p> <p>2)发布</p>	作业是没有发布过的	作业发布成功	<p>作业发布成功</p> <p>说明作业发布模块成功</p>
10	<p>[目的]测试新增（已创建）课程</p> <p>[流程]</p> <p>1)填写（已创建）课程信息</p> <p>2)创建</p>	课程是之前创建过的	提示课程已创建，终止课程创建	<p>已创建该课程创建失败</p> <p>说明不能再创建重复课程</p>
11	<p>[目的] 测试上传（重复名字）课件</p> <p>[流程]</p> <p>1)选择相同课件</p> <p>2)上传</p>	课件是已上传的	提示课件已上传，终止课件上传	<p>课件名重复上传失败</p> <p>说明课件名不能重复</p>

#### （4）管理员文章管理

文章管理包含增加文章，删除文章，查看文章，查询文章列表等功能，以下列举新增文章、删除文章、查看文章和查询文章列表的测试用例，利用单元测试验证文章管理模块功能的有效性与正确性。

表 5-4 为文章管理的测试用例。

表 5-4 文章管理测试用例

测试用例编号	测试目的和步骤	前提条件	期望结果	实际测试结果与结论
12	<p>[目的]测试新增（符合条件的）文章</p> <p>[流程]</p> <p>1)填写文章信息</p> <p>3)提交</p>	<p>1)用户为系统管理员</p> <p>2)文章标题是没有发布过的</p>	<p>新增文章成功</p>	<p>文章新增成功</p> <p>说明新增新文章成功</p>
13	<p>[目的]测试删除文章</p> <p>[流程]</p> <p>1)选择文章</p> <p>2)删除</p>	<p>1)用户为系统管理员</p> <p>2)该文章是可以删除的</p>	<p>删除文章成功</p>	<p>删除文章成功</p> <p>说明可删除文章</p>
14	<p>[目的]测试查看英语阅读文章</p> <p>[流程]</p> <p>1)分别选择双语，英文，中文模块</p> <p>2)内容是正确显示的</p>	<p>文章的内容是正确显示的</p>	<p>查看文章成功</p>	<p>文章显示正确</p> <p>说明文章展示模块完成</p>
15	<p>[目的] 测试查询文章列表</p> <p>[流程]</p> <p>1)选择对应的文章栏目</p> <p>2)选择列表里面的文章</p>	<p>文章是已发布的，列表里的文章是对应栏目的</p>	<p>查询文章列表成功</p>	<p>查询文章列表正确</p> <p>说明文章列表查询成功</p>

以上列举了部分模块的测试用例，大致介绍了单元测试运用的方法和测试的步骤，并给出预期达到的结果。根据设计的测试用例，经过了对系统各个模块的单元测试之后，各个模块运行正常，测试的结果也达到了预期结果。

### 4.3 集成测试

集成测试在全局层面对系统的兼容性进行测试。集成测试模拟实际使用过程中的条件，对系统模块中的每个功能进行测试。表 5-5 列举了简要的测试用例，集成了登录注册系统、前台系统、后台管理系统多个子系统进行测试。

表 5-5 集成测试用例

测试用例名称	测试步骤	期望测试结果	实际测试结果 与结论
用户注册	输入新账号，邮箱，手机号，填写密码和确认密码，昵称。	输入正确，注册成功，跳转到登录界面。	用户注册成功 注册模块完成
用户登录	输入注册的账号、密码。点击登录。	账号密码正确，登录成功，跳转至平台首页。	用户登录成功 登录模块完成
查看文章列表	进入文章阅读，选择分类，跳转分类文章。	显示选择类型的文章列表。	正确展示文章 列表 文章列表模块 完成
查阅文章	选择一篇文章，选择双语，中文或英文	文章内容为对应语言，查阅文章成功	正确展示文章 对应内容 文章展示成功
查看我的课程	进入我的课程，选择课程学习	跳转我的课程，课程查看成功	展示我的课程 我的课程模块 完成
下载课件	选择我的课程课件点击下载	下载课件到本地成功	下载对应课件 下载课件模块 正确
阅读测试	选择阅读测试，规定时间内答题	批改文章成功，返回正确测试结果	正确返回答题 结果
修改个人信息	进入编辑个人信息，点击编辑，填写需要修改的信息，点击提交。	个人信息编辑成功，可以查看到修改后的个人信息。	个人信息修改 成功
退出登录	点击退出登录。	返回到登录界面，无法访问用户其他页面	退出平台登录 成功
课程管理	使用教师账号，点击课程管理	跳转我的课程，管理课程	进行课程管理 成功
创建课程	使用教师账号，创建课程信息	课程创建成功	创建新课程成

			功
上传课程资源	选择课程资源，点击上传	上传成功	课程资源上传成功
管理员登录	输入管理员账号、密码，点击登录。	账号密码正确，登录成功，跳转至后台管理系统首页。	管理员登录成功 管理员登录模块无误
新增文章	填写文章相关信息，点击提交。	新增文章成功，刷新文章列表。	新增文章成功
编辑文章	在文章列表选择一篇文章点击编辑，修改需要修改的信息，提交。	编辑文章成功，刷新文章列表。	编辑文章成功
删除文章	使用超级管理员账号，选择一篇文章，点击删除。	文章删除成功，刷新文章类型列表。	删除文章成功
新增企业院校用户	使用超级管理员账号，进入企业管理，添加企业用户，点击提交。	新增企业号成功，刷新企业列表。	新增企业号成功
编辑企业院校	使用超级管理员账号，进入查看社区列表，选择一个企业点击编辑，修改需要修改的内容，点击提交。	编辑企业成功，刷新企业列表。	修改企业成功
新增管理员	使用超级管理员账号，进入新增管理员，点击选择用户，选择需要称谓管理员的用户，点击选择，再点击提交。	新增管理员成功，刷新管理员列表。	新增管理员成功
删除管理员	使用超级管理员账号，进入管理员列表，选择一个管理员点击删除。	删除管理员成功，刷新管理员列表。	删除管理员成功

集成测试基于系统的单元测试完成的，按照系统的概要设计说明组成模块对其业务流程进行测试。

经过了本次集成测试后，系统各模块之间的协作均正常，达到了集成测试的预期测试结果，证明系统各个模块之间功能均正常且可以顺利地协作与运行。

## 4.4 兼容性测试

英语阅读学习平台的兼容性测试包括浏览器兼容性测试和不同屏幕大小的兼容性测试

1. 浏览器兼容性测试主要测试主流浏览器的网页显示效果的兼容性，各类浏览器兼容性效果图如下：

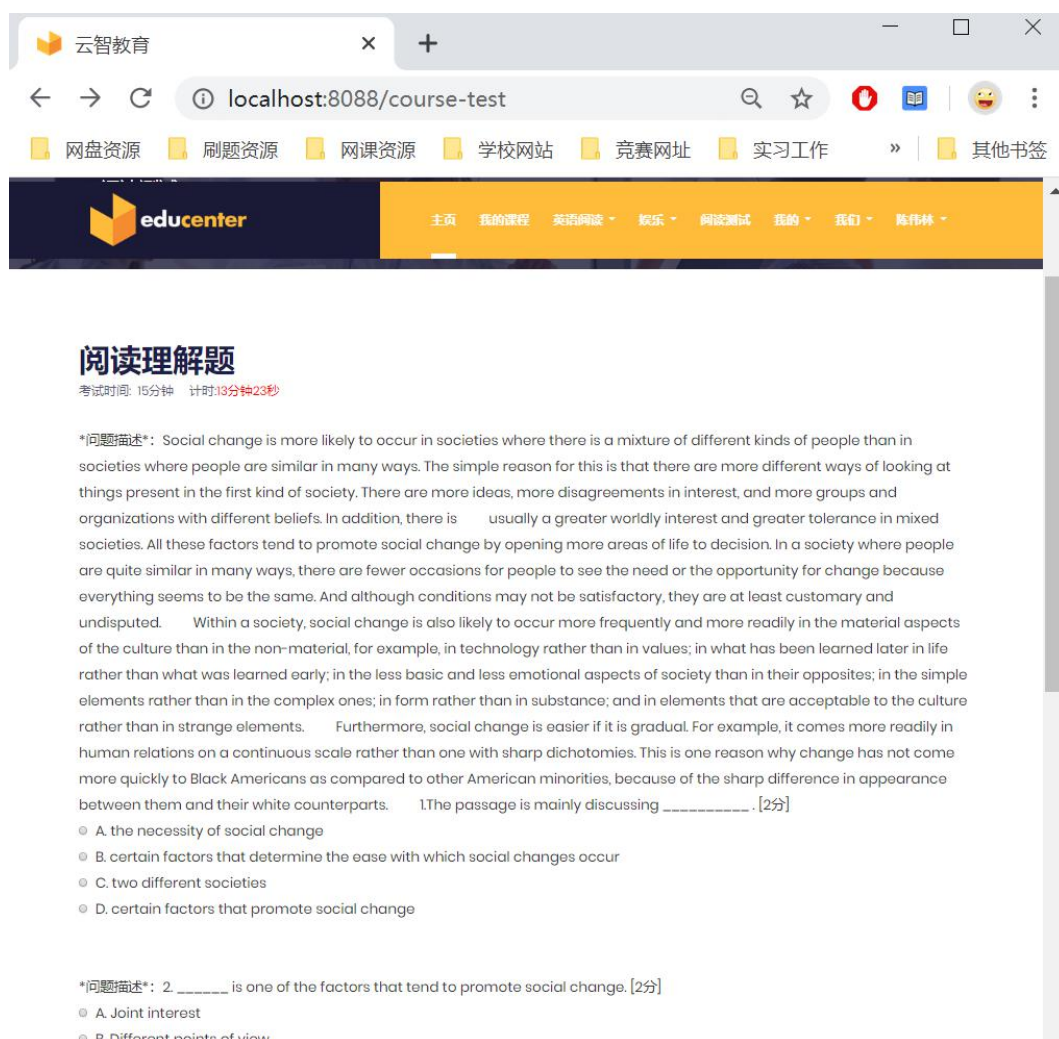


图 5-1 Chrome 浏览器效果图

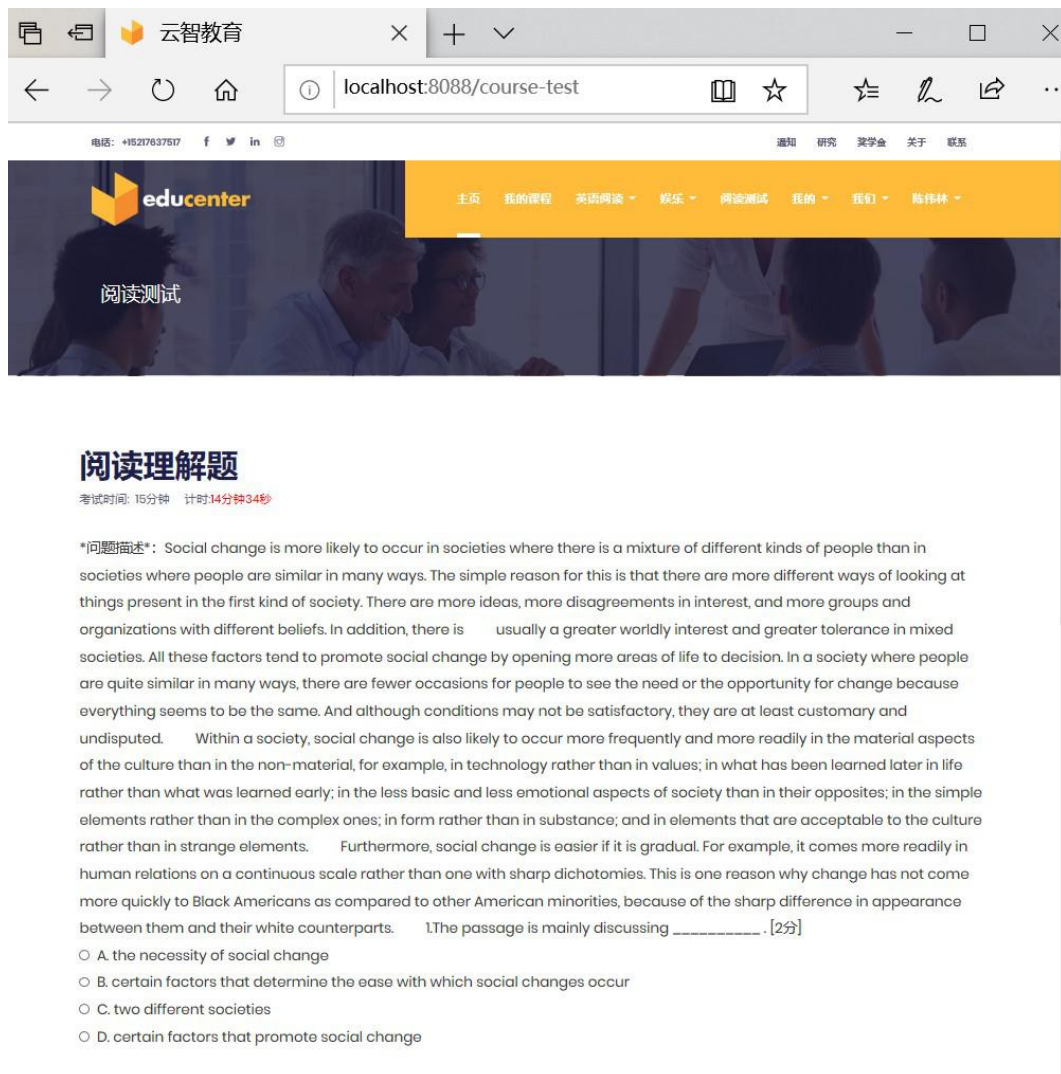


图 5-2 Microsoft Edge 浏览器效果图



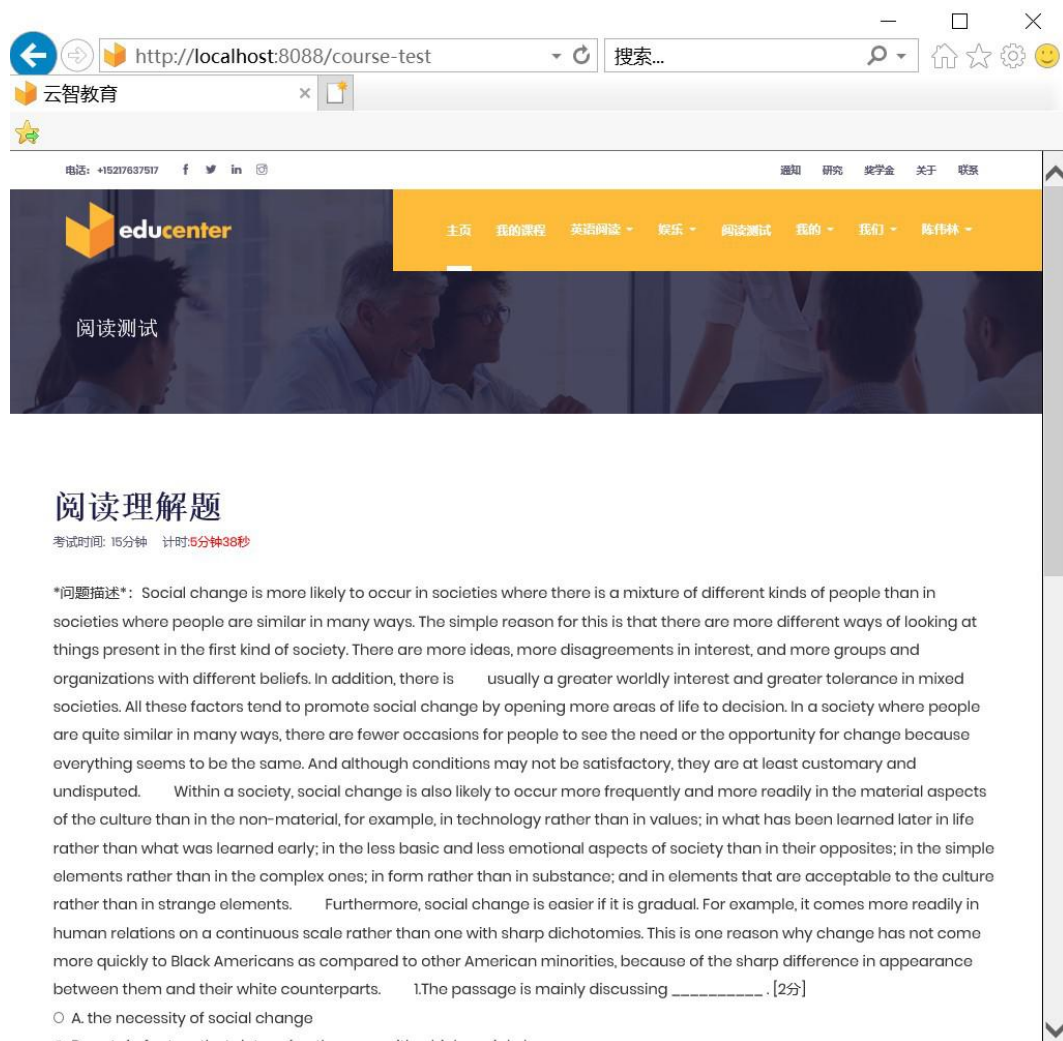


图 5-3 IE 浏览器效果图

## 2. 接下来展示部分屏幕兼容的图片：



图 5-4 大屏幕显示效果图



图 5-5 中屏幕页面效果图



图 5-6 小屏幕页面兼容性效果图 1



图 5-7 小屏幕页面兼容性效果图 2

### 3. 测试结论

英语阅读移动学习平台对主流浏览器和手机平板等设备兼容性好，页面展示效果较佳，用户体验不错。

## 4.5 本章小结

本章简要介绍了系统测试，在开发过程中应用了单元测试，并对系统进行了集成测试和性能测试，展示了简单的集成测试测试用例，随后进行了网页自适应测试，并展示了英语阅读平台的测试结果，经过一系列测试之后，基本达到了系统的需求和预期结果。

## 5 总结

### 5.1 总结

本平台为满足学生的英语阅读的需求出发，结合相关理论基础和英语阅读实际需求，开发了基于混合式学习的英语阅读移动学习平台。在系统开发上遵循整个软件开发过程。从前期的资料搜集，到系统的需求成型，再到系统的搭建与开发，到最后完成开发并对系统进行测试。

本平台可以实现学生和教师的英语阅读的学习，提供优质的教学资源，准确地向学生和教师提供数据动态，使得学生提高英语阅读能力，使教师方便展开线上课堂学习，同时教务可以监控管理学习平台。解决了学生和教师之间交互信息不足，英语阅读资源不够等问题。

平台采用 SSM 框架结合 SpringBoot 框架，MVC 框架。网站前端开发使用 Bootstrap 框架，自适应网站设计及相关开源框架与模板，采用 MySQL 数据库，IntelliJ IDEA 开发平台，使得系统更加灵活规范，且方便运营和维护。由于本次研究的项目属个人完成，在实际开发过程中不会遇到团队合作开发时遇到的配合问题。在实际开发过程中，从前端页面、数据交互，到后台搭建以及业务逻辑处理，都遇到过很多未曾发现的问题，也学到了很多，对 Java 编程语言和使用的三大开发框架也有了更为深入的了解。但在实际开发中，由于笔者开发时间和技术有限，本人无法实现如视频，直播等教学内容传输上的技术问题，这也是本次研究的不足之处。

本平台是基于教学平台的教学模型，在教师的带领下，加强师生之间在学习英语阅读过程中的互动以及各种教学资源，以创造平等，启发，共享，互动合作与和谐的教学环境，打破了传统教学模式的时空限制，完善了传统教学方法。平台实现多层次课程体系，提供一站式混合教学解决方案，打造教、学、评、测，研一站式混合教学解决方案，全方位提升学习体验和教学效果，实现多终端无缝衔接，融合不同的教学场景，随时随地，智慧学习。