



Development of Online Education Platform Based on Django

Supervisor:
Dr Natasha Edwards

Author:
Changchen Wang

Date:06/10/2021

Msc Computing and IT Management
School of Computer Science and Informatics, Cardiff University

Contents

1. INTRODUCTION.....	6
1.1 ABSTRACT	6
1.2 BACKGROUND	6
1.3 RESEARCH AIMS AND OBJECTIVES.....	6
1.4 STRUCTURE OF DISSERTATION.....	7
2. RELATED WORK:.....	7
2.1 HOMEPAGE	7
2.2 LOGIN	8
2.3 REGISTRATION	10
2.4 PERSONAL CENTRE.....	11
2.5 COURSES PAGE.....	15
2.6 COURSES DETAIL PAGE.....	16
2.7 COURSE CHAPTER INFORMATION PAGE.....	17
2.8 TEACHER PAGE	17
2.9 TEACHING ORGANIZATION PAGE.....	18
2.10 ORGANISATION DETAILS PAGE	19
2.11 WHAT I WOULD LIKE TO ADD TO GMOOC	20
3. APPROACH	23
4. DATABASE MODEL.....	25
4.1 EER MODEL.....	25
4.2 DATABASE DESIGN	26
5. SYSTEM REQUIREMENT	27
5.1 REQUIREMENT DOCUMENT	27
5.2 SYSTEM DESIGN	29
6. IMPLEMENTATION.....	43
6.1 HTML PAGE DESIGN.....	43
6.2 VIEWS AND ROUTES	53
6.3 ROUTES	59
7. UNIT TESTING.....	61
7.1 HOME PAGE	61
7.2 USER LOGIN.....	65
7.3 REGISTRATION:	68
7.4 PERSONAL CENTER	71
7.5 OPEN COURSES.....	77
7.6 COURSE DETAIL PAGES.....	81
7.7 COURSE CHAPTER INFORMATION PAGE	84
7.8 ORGANIZATION DETAIL	88
7.9 BACKGROUND MANAGEMENT SYSTEM	96
8. REFLECTION.....	105
IMPLEMENTATION CONCLUSION.....	105
PERSONAL CONCLUSION.....	106
WHOLE PROJECT CONCLUSION	106
APPENDICES.....	108
APPENDIX 1.....	108

REFERENCES.....	114
------------------------	------------

List Of Figures

2.1 Home page of FutureLearn education website	5
2.2 Home page of icourse.org education website.....	6
2.3 page of FutureLearn.org Online education Platform.....	7
2.4 Login page of icourse.org Online education Platform.....	7
2.5 Registration page of icourse.org Online education Platform.....	8
2.6 Registration page of FutureLearn Online education Platform.....	9
2.7 Personal center data setting.....	10
2.8 Registration page of FutureLearn Online education Platform.....	10
2.9 The course the user is learning on the FutureLearn.....	11
2.10 The course the user is learning on the icourse.org.....	11
2.11 The school list on the icourse.org.....	12
2.12 The user message on the icourse.org.....	12
2.13 The user message on the FutureLearn.org.....	13
2.14 The course list on the icourse.org.....	13
2.15 The course detail on the FutureLearn.org.....	14
2.16 The course detail on the icourse.org.....	14
2.17 The course chapter on the FutureLearn.org.....	15
2.18 The Teacher list on the icourse.org.....	16
2.19 The school list on the icourse.org.....	17
2.20 The school detail on the FutureLearn.org.....	18
2.21 The teacher list on the FutureLearn.org.....	18
2.22 The chapter detail on the Mooc.org.....	19
2.23 The chapter detail on the icourse.org.....	19
2.24 The chapter detail on the FutureLearn.org.....	20
3.1 System development steps.....	21
4.1:EER Model for GMOOC Database.....	22
6.1:GMOOC homepage.....	41
6.2: GMOOC homepage 2.....	41
6.3: GMOOC homepage 3.....	42
6.4: Login page.....	44
6.5: Registration page.....	44
6.6: Course-list page.....	45
6.7: Course-detail page.....	46
6.8: Course-info page.....	47
6.9: Course-comment page.....	47
6.10: Course-comment page.....	48
6.11: Org-detail-homepage.....	48
6.12: Org-detail-course page.....	49
6.13: Org-detail-desc page.....	50
6.14: User center-fav-course page.....	51
6.15: User center-message page.....	52
6.16: Register View.....	53
6.17: Login windows.....	53
6.18: Login views.....	54
6.19: User center url.....	55
6.20: User information view.....	56
6.21: Course url.....	56
6.22: Paginator function.....	57
6.23: Comments view.....	58
6.24: Comments view 2.....	58
6.25: Comments view.....	59
6.26: Course resource filter.....	59

List of Tables

2.1 The result of comparison.....	23
5.1: User requirement Analysis.....	29
5.2: Use case 1 User registration.....	30
5.3: Use case 2 User Login.....	32
5.4: Use case 3 view courses.....	32
5.5: Use case 4 comments.....	34
5.6: Use case 5 download learning materials.....	34
5.7: Use case 6 Change personal information.....	35
5.8: Use case 7 Global search.....	37
5.9: Use case 8 Navigation bar.....	37
5.10: Use case 9 View the lessons the user is learning.....	38
5.11: Use case 10 Users view organization that have been bookmarked.....	40
5.12: Use case 11 View my message.....	42
5.13: Use case 12 Course sorting.....	42
5.14: Use case 13 Favorite courses.....	44
6.1: Routes and URL.....	<u>61</u>
7.1: Unit test navigation bar.....	62
7.2: Unit test global search.....	66
7.3: Unit test Login.....	66
7.4: Unit test Log out.....	68
7.5: Unit test User registration.....	69
7.6: Unit test User Incorrect Username Registration.....	70
7.7: Unit test User information change.....	72
7.8: Unit test view courses.....	74
7.9: Unit test view organizations.....	75
7.10: Unit test view own messages.....	76
7.11: Unit test change account password.....	78
7.12: Unit test courses sorting 1.....	79
7.13: Unit test paging.....	81
7.14: Unit test learn courses.....	83
7.15: Unit test favorite courses.....	83
7.16: Unit test courses detail.....	85
7.17: Unit test user comments.....	85
7.18: Unit test paging.....	87
7.19: Unit test organizations sorting.....	89
7.20: Unit test view courses information.....	90
7.21: Unit test view courses information.....	91
7.22: Unit test view teachers.....	93
7.23: Unit test change the theme.....	93
7.24: Unit test change view user's information.....	94
7.25: Unit test change view learning materials.....	95
7.26: Unit test manage organization information.....	96
7.27: Unit test results.....	104

1. Introduction

1.1 Abstract

With the widespread use of MOOCs (Massive Open Online Course) in 2013, people began to pay attention to Online distance learning as a new way of learning. Due to technological limitations, most schools and educational institutions still use traditional education methods [1]. It has been proved that the MOOC-based online education model can advance teachers' careers and improve their teaching level [2] and enable students to break through the constraints of time and space and receive education anytime and anywhere [3]. On the other hand, the online education platform still needs to be updated regarding resource integration and course component functions [4]. This dissertation will discuss the defects of the online education platform from the perspective of controversy and improve the potential weaknesses of the online education platform as much as possible through additional functions.

1.2 Background

Students need to supplement their extracurricular knowledge through online education platforms. Due to its rich teaching content and high-quality teaching resources, MOOC has gradually become an online education platform frequently used by Chinese college students. Not long ago, I was interested in data analysis, so I went to the MOOC website for college students to find teaching videos related to machine learning. But I find that if I want to review the teacher's content in class, after class, I need to continue to read a cell phone photo album to view the course the teacher showed in the process of teaching PPT screenshot. I just think if the teaching video next to can have a quick download learning materials channel quickly get PPT and document in the process of the teacher in class. Still, teaching resources to upload web applications have a background management system; management needs to download data upload information in the background management system for students to realize this idea. I'm going to use the Django framework to develop a background management system of an online education platform, by improving the page layout and function, to the integration of teaching materials, Convenient user experience.

1.3 Research aims and objectives

Aims:

Due to the COVID-19 pandemic, more and more universities and educational institutions are choosing to teach online. However, schools in some countries or regions may not be able to find suitable ways to offer online courses due to lack of funding or the poor quality of existing platforms. Therefore, the aim of this plan is to design an online education platform that meets the upper limit standard and make appropriate improvements to better solve the problem of shortage of online education platform resources caused by the epidemic, to provide more

choices of online platforms for schools or educational institutions in different countries and regions.

Objectives:

1. View and research existing online education platforms in different countries, and analyse the advantages and good functions of different online education platforms.
2. Evaluate the good functions of the existing online education platform, consider the functions that need to be preserved, and write the observed results into the functional and non-functional requirements of the project. Then see where the existing solution falls short and think about how to add new features.
3. Completed the design of online education platform, database design, apps design, model design, etc.
4. When the online education platform is completed, I will use the method of unit test to evaluate the functional and non-functional requirements of the system, platform availability and platform quality.
5. Complete the project dissertation.

1.4 Structure of dissertation

This dissertation consists of eight chapters. The first chapter introduces the background and structure of the thesis. In the second chapter, I will analyze the front-end page and functional design of the three existing online education platforms to design the requirements of this project. In the method of chapter 3, I will introduce the development tools and Web framework used in the project, the software design scheme of the project, and clarify the development process and steps of the project. In Chapter 4, I'll introduce the database model and design. In chapter 5, I will design the functional requirements of this project by referring to the existing solutions and define the development goals of this online education platform. In chapter 6 implementation, I will design front-end pages, URLs, and views according to the four core modules of this topic. In Chapter 7, I will unit test the project against the project requirements document. The eighth chapter is the summary of this subject and the personal reflection report.

2.Related work:

To complete this project, I needed to look at online education platforms in different countries. I used Google and Chinese search engines respectively to search for "MOOC" as the keyword and selected the top MOOC website for analysis. By default, the leading MOOC website in this dissertation has the most clicks, and most people have recognized the quality of online education. The following chapter will analyze the pages and functions of mooc.org, Futuerlearn.com, and Icourse.org, which ranked first in the search engine of Baidu, and then design the demand document of this online education platform GMOOC.

2.1 Homepage

By observing the home pages of Chinese college students' MOOC network (icourse.org)and FutureLearn.com, we can first find that they both have navigation bars to check the information of courses, educational institutions, and lecturers. Then we can see that they all support global search, and users can search for courses of user's interest. Finally, there are log-in and

registration functions in the upper right corner. As shown in the figure 2.1 and 2.2:



Figure 2.1 Home page of FutureLearn education website



Figure 2.2 Home page of icourse.org education website

2.2 login

Users can log in to the online education platform with an email and password, and if they forget their password, they can reset it by clicking "Forget your password." As shown in the figure 2.3 and 2.4:

Email

Password

[Forgot your password?](#)

Remember me

Sign in

Figure 2.3 page of FutureLearn.org Online education Platform

邮箱登录

手机号登录

爱课程登录



网易邮箱/常用邮箱



密码

登录



十天内免登录 | 忘记密码?

[去注册](#)

Figure 2.4 Login page of icourse.org Online education Platform

2.3 Registration

The registration function allows users to register with their email and password and first name, last name, and other personal information. Considering the portability of user operation, I think that only the mailbox entered by the user will be used as the only login account in this project. As shown in the figure 2.5 and 2.6:

1 填写帐号密码

2 验证手机号码

3 完成注册

请输入邮箱地址

请输入8-16位密码

用户勾选即代表同意 [《网易邮箱帐号服务条款》](#) 和 [《网易隐私政策》](#)

下一步

Figure 2.5 Registration page of icourse.org Online education Platform

[Sign in](#) [Register](#)

First name *

Last name or initial *

Age *

Email address *

Please check your email address to make sure it's correct.

Choose a password (min. 10 characters) *

Register

Figure 2.6 Registration page of FutureLearn Online education Platform

2.4 Personal Centre

2.4.1 Change personal information

In the personal center, users can modify personal information such as name, profile image, etc., As shown in the figure 2.7 and 2.8:

On FutureLearn courses, you can follow and join discussions with other learners. To help you recognise each other, everyone has a profile.

If you're under 18, read our advice about [staying safe online](#), before adding anything to your profile.

First name *

Changchen

Last name or initial *

Wang

Certificate name

This is the name that will appear on your digital and printed certificates. If you leave this blank, we will use the name that is shown on your profile.

Changchen Wang

Profile image (Optional)



Upload an image

Remove image

0/200

About you

Say a few things about yourself

Figure 2.7 Personal center data setting

资料设置 帐号设置 邮件设置

绝不会以任何形式向第三方透漏你的身份信息

头像

编辑头像

*昵称 ChangchenWang

*常用邮箱 734078899@qq.com 未验证 [修改邮箱](#)

用于接收课程通知与认证证书电子版。

!

验证邮件已发送到您填写的邮箱，请查收并验证。[立刻登录邮箱验证](#)

如已验证，请刷新页面；
如果没有收到验证邮件，① 确认邮箱填写正确；② 确认该邮箱已激活；③ 检查垃圾箱；
如邮件中链接失效，请点击 [再次发送验证邮件](#)

*手机帐号 17633458938 [更换手机](#)

用于优质内容推荐及活动提醒，该信息不会对外公开

Figure 2.8 Registration page of FutureLearn Online education Platform

2.4.2 Users can view the lessons they are learning

Users can check the courses they are learning by clicking the Wishlist in the personal center page, As shown in the figure 2.9 and 2.10:



Figure 2.9 The course the user is learning on the FutureLearn

Figure 2.10 The course the user is learning on the iCourse.org

2.4.3 Viewing Organization Information

Users can also query course information of different institutions in the personal center, As shown in the figure 2.11:

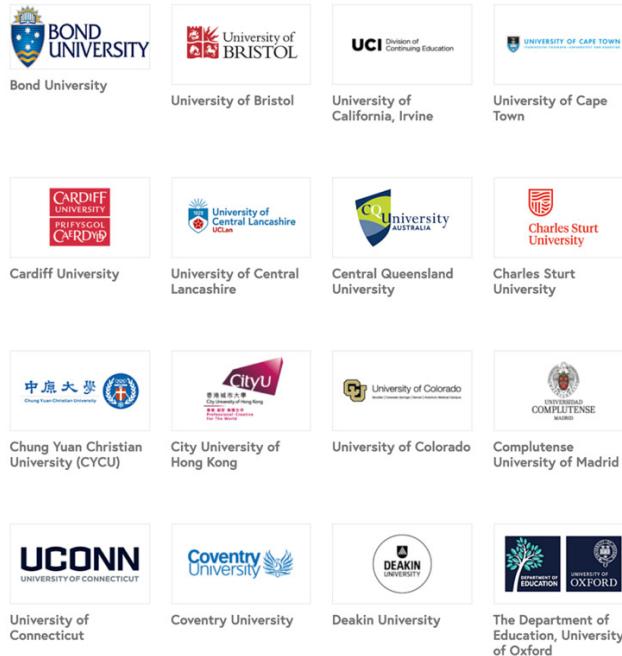


Figure 2.11 The school list on the icourse.org

2.4.4 View personal message

Users can view personal messages in the personal center, As shown in the figure 2.12 and 2.13:



Figure 2.12 The user message on the icourse.org

Notifications



Figure 2.13 The user message on the FutureLearn.org

2.5 Courses page

In the course page users can view the most popular courses and sort courses, and there is a pagination function at the bottom of the page, users can view more courses. As shown in the figure 2.14:

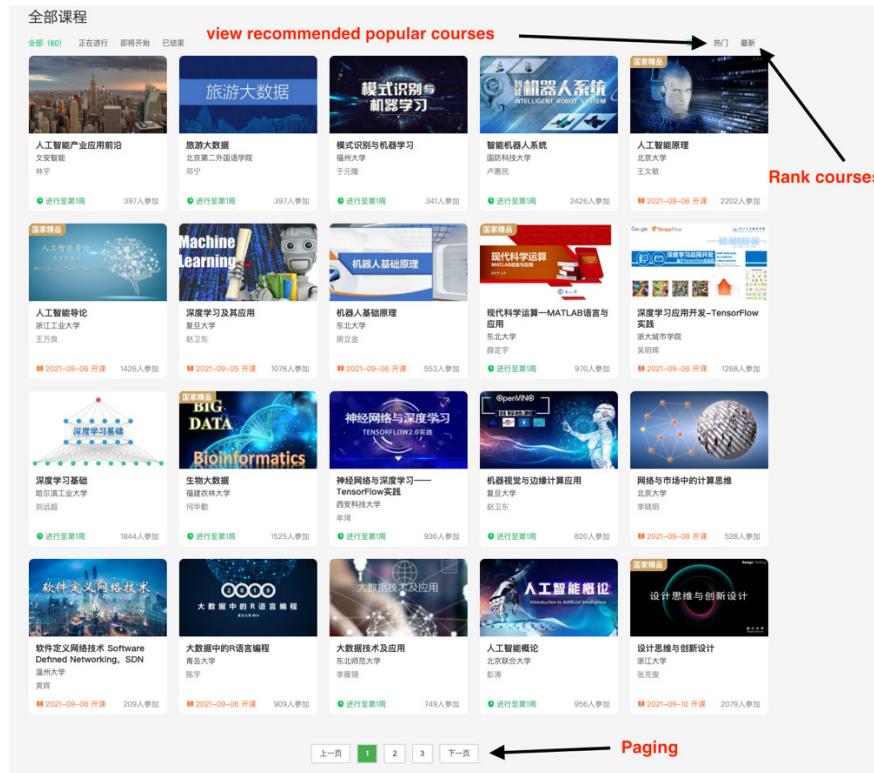
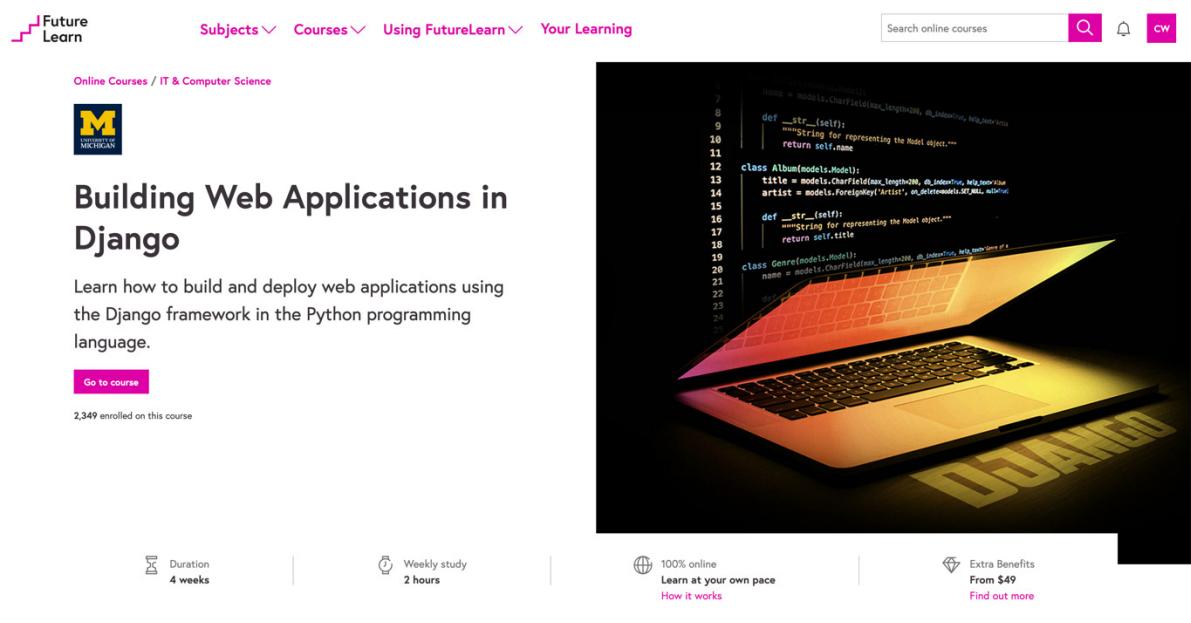


Figure 2.14 The course list on the icourse.org

2.6 Courses detail page

In the course details page, users can click to start learning the course, and they can bookmark or unbook mark the course they are interested in. In the course details page, there is a text description of this course, explaining the main content of this course. As shown in the figure 2.15 and 2.16:



The screenshot shows the course detail page for 'Building Web Applications in Django' on FutureLearn.org. The page includes the course title, a brief description, a 'Go to course' button, enrollment statistics (2,349 enrolled), and course details like duration (4 weeks) and weekly study (2 hours). A large image of a laptop displaying Python code is prominently featured. The course description text is as follows:

```
name = models.CharField(max_length=200, db_index=True, help_text='Name of the album')
def __str__(self):
    """String for representing the Model object."""
    return self.name
class Album(models.Model):
    title = models.CharField(max_length=200, db_index=True, help_text='Title of the album')
    artist = models.ForeignKey('Artist', on_delete=models.SET_NULL, null=True)
    def __str__(self):
        return self.title
class Genre(models.Model):
    name = models.CharField(max_length=200, db_index=True, help_text='Name of the genre')
    def __str__(self):
        return self.name
```

Figure 2.15 The course detail on the FutureLearn.org



The screenshot shows the course detail page for 'Machine Learning' on icourse.org. The page includes the course title, a brief description, a '立即参加' (Join Now) button, and sections for course details and faculty. The course description text is as follows:

本课程主要介绍深度学习的基本原理、卷积神经网络、循环神经网络、生成对抗网络、注意力机制、Transformer等基本方法及其典型应用领域。并借助机器学习开源框架TensorFlow实现深度学习在证券趋势预测、声音质量评价、电子推荐、目标检测、社交网络信息分析等多个典型领域的应用。

Figure 2.16 The course detail on the icourse.org

2.7 Course chapter information page

By observing the features of the FutureLearn.com course chapter details page, this article found that the functions that can be added to this project include: users can browse the course content; Users can comment on courses; Users can download learning materials is a new feature added to this project. The FutureLearn.com course chapter information page is shown below:

The screenshot shows a course chapter page with the following structure:

- Welcome to the course**
 - Who we are and what the course is about.
 - 1.1** Welcome to the course VIDEO (02:18)
 - 1.2** Who are we? ARTICLE
 - 1.3** Learning with us ARTICLE
 - 1.4** Introduce yourself DISCUSSION
- Evidence based prevention**
 - How can we prevent caries in children? We start off with two videos on advising families and what we can do, and some reading based on important evidence-based approaches. There's a quiz at the end to check your knowledge.
 - 1.5** Preventing caries in children: Giving families the best advice VIDEO (06:19)
 - 1.6** Preventing caries in children: Treatment in the chair VIDEO (03:08)
 - 1.7** Using Silver Diamine Fluoride (SDF) ARTICLE
 - 1.8** Evidence based prevention: Testing your knowledge QUIZ

Course Resource
chapter 1.1
chapter 1.2
chapter 1.3
...

Support

Figure 2.17 The course chapter on the FutureLearn.org

2.8 Teacher page

By observing icourse.org, we can find that in the teacher course page of this website, users can rank the lecturers and their corresponding courses. And users can view the list of teachers; There is also paging at the bottom of the page. As shown in the figure 2.18:

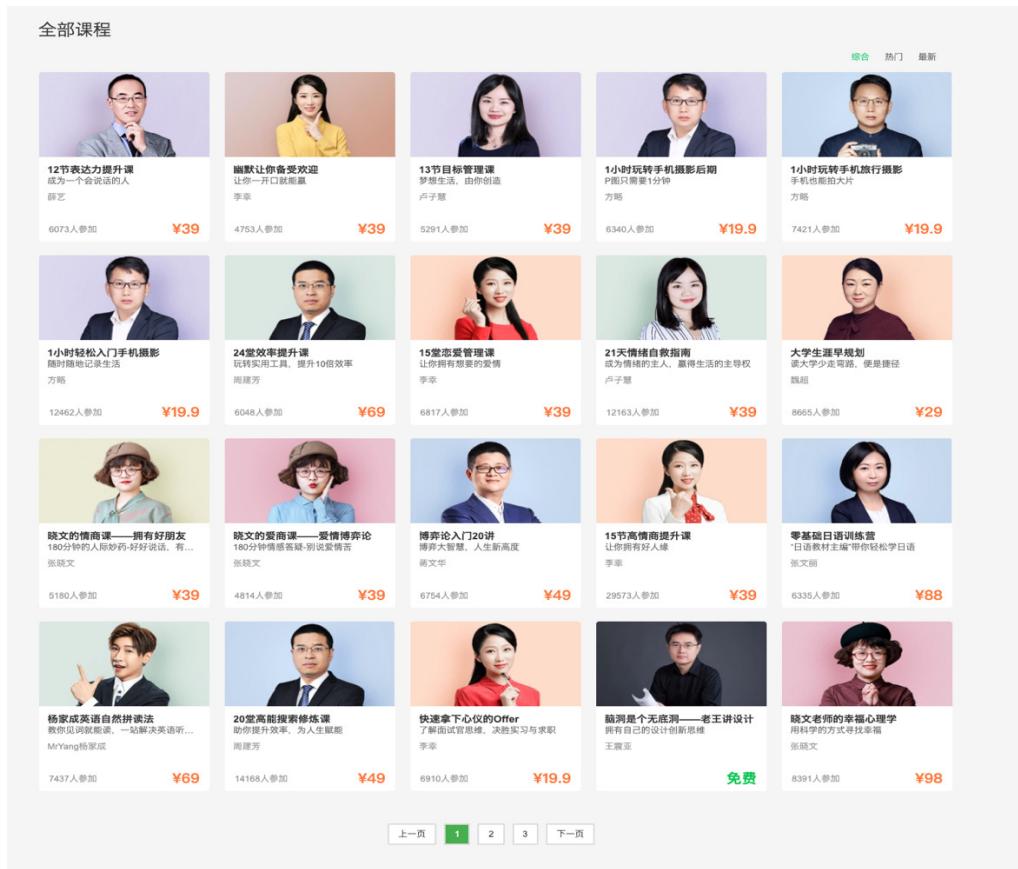


Figure 2.18 The Teacher list on the icourse.org

2.9 Teaching Organization Page

By observing the icourse.org, we can find that on the page of teaching institutions, users can filter the teaching institutions by category and region. Users can click to enter the teaching institution's detailed information page; Users can rank courses according to the number of students and the number of courses. As shown in the figure 2.19:



Figure 2.19 The school list on the icourse.org

2.10 Organisation Details Page

On FutureLearn.com's teaching institution Details page, users can view courses at a given institution, and they can view an organization profile for that institution, as well as view all lecturers within the organization. As shown in figure:

Cardiff University



About

Cardiff University is one of Britain's major teaching and research universities. It is a member of the Russell Group of the UK's leading research-intensive universities and is ranked within the top 150 universities in the world in the QS World University Rankings.



<http://www.cardiff.ac.uk/>
@cardiffuni
cardiffuni

Support

Figure 2.20 The school detail on the FutureLearn.org



Morten Andersen

Lecture at School of Earth & Ocean Sciences, Cardiff University. I am a Geologist studying the long term chemical evolution of the Earth's surface environment.



Ernest Chi Fru

Ernest does research in Geomicrobiology, Geology and Biogeochemistry. Current focus is on biogeochemical elemental cycles linked to the evolution of life through Earth history.



Marc-Alban Millet

I'm a lecturer in isotope geochemistry at Cardiff University (UK). I research how magmatic systems work and how this underpins the evolution of Earth and other planets.



Dan Hobley

I'm a lecturer in geomorphology



Claire Earlie

Claire is a Lecturer in Coastal



Wim Degruyter

I am a Lecturer in Geodynamics

Figure 2.21 The teacher list on the FutureLearn.org

2.11 What I would like to add to GMOOC

I want to add the function of downloading learning materials in GMOOC because none of the sample systems I searched in this design provided a download link for downloading the

supporting learning materials of course chapters. Through observation, it is not difficult to find that the teaching video of Mooc.org in Figure 2.22 provides download and subtitles of video files at the bottom of the video. However, there is no comment section where students can interact and discuss. Although icourse.org in Figure 2.23 has the functions of course evaluation and 310 student comments, it does not provide a functional module for downloading materials. Similarly, the instructional video page of FutureLearn.com's online education platform in Figure 2.24 does not include the ability to download materials. Therefore, I want to add this function to the GMOOC system developed this time.

Welcome to the Course

Bookmark this page

Welcome to the Course

So in this course, we're going to move from the basic shape of a Django application to more of the skills and techniques that you're going to use when you build applications, things like sessions, things like users. We're going to learn a lot about the Django way of doing things. A lot of code that you do in Django is really short and succinct, but ultimately underneath that, there's a whole bunch of stuff. A big part of the approach to

0:00 / 0:00 1.0x

视频 [下载视频文件](#)

字幕 [下载 SubRip \(.srt\) file](#) [下载 Text \(.txt\) file](#)

15% First-Time Learner Discount

6 days left

- ✓ Earn a [verified certificate](#) of completion to showcase on your resume
- ✓ Unlock your access to all course activities, including [graded assignments](#)
- ✓ [Full access](#) to course content and materials, even after the course ends
- ✓ Support our [non-profit mission](#) at edX

Upgrade for \$41.65 (\$49)

Use code **EDXWELCOME** at checkout

Figure 2.22 The chapter detail on the Mooc.org

中国大学MOOC

课程 学校 学校云 慕课堂 下载APP

搜索感兴趣的课程

登录 | 注册

首页 > 计算机 / 工学

机器学习

第3次开课

开课时间: 2021年07月06日 ~ 2021年09月12日

学时安排: 3-5

进行至第5周, 共10周

已有 9988 人参加

立即参加

课程详情 课程评价(310)

本课程重点介绍机器学习中的核心算法和理论, 使学生通过理论学习掌握机器学习中的经典理论, 了解当前最新发展, 并学会针对各自学科的具体问题设计机器学习算法。课程内容包括支持向量机、人工神经网络、深度学习、强化学习、传统的机器学习五个模块, 用丰富的理论推导和实验案例加深学生的理解。

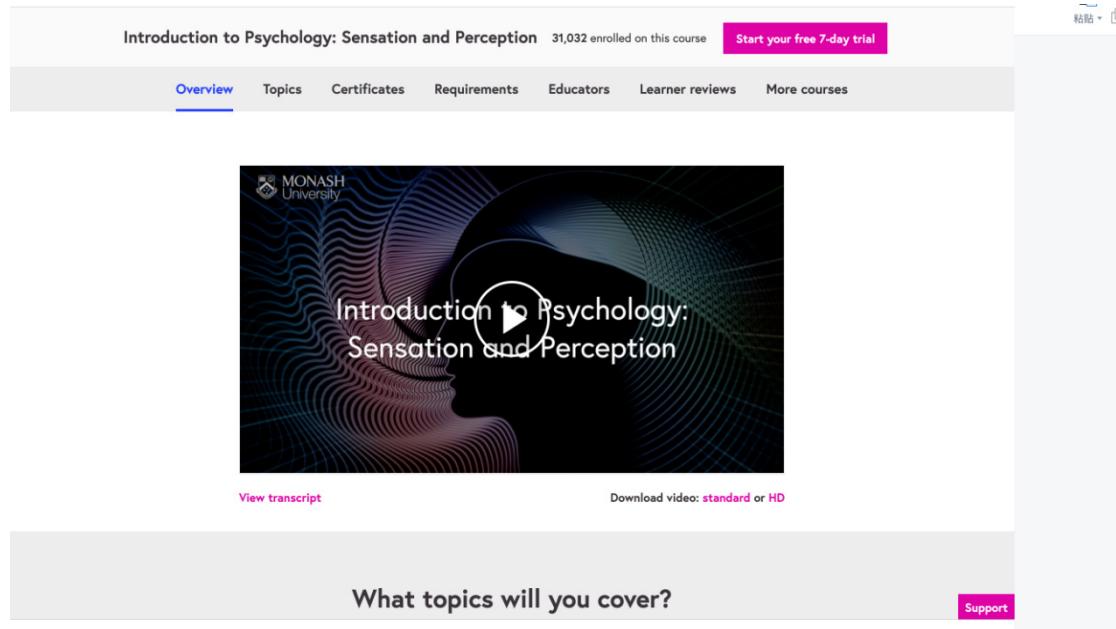
— 课程团队

浙江大学

1 位授课老师

胡浩基

Figure 2.23 The chapter detail on the icourse.org



What topics will you cover?

Support

Figure 2.24 The chapter detail on the FutureLearn.org

Here are my results based on the comparison of three online education platforms:

Page	Icourse.org	FutureLearn.org	MOOC.org
navigation bar	✓	✓	✓
Global search	✓	✓	✓
Course list	✓	✓	✓
Teacher list	✓	✓	✓
The login	✓	✓	✓
logout	✓	✓	✓
registration	✓	✓	✓
Change personal Information	✓	✓	✓
Check out the courses at the People center	✓	✓	✓
Reset password	✓	✓	✓
View personal message	✓	✓	✓
Filter the course	✓	✓	✓
Course sorting	✓	✓	✓
Course details	✓	✓	✓
Video playback	✓	✓	✓
User comments	✓	✓	✓
View the teacher's message	✓	✗	✗
List of teaching institutions	✓	✓	✓
Information about teaching institutions	✓	✓	✓
Classification of	✓	✗	✗

teaching institutions (region, individual/university)			
Download teaching resources	x	x	x
Sort of teaching institutions	✓	✓	✓
Page paging	✓	✓	✓
User favorite courses	✓	✓	✓

Table 2.1 The result of comparison

Finally, In this chapter, I observed online education platforms' page design and function design in different countries. In the fourth chapter, I will combine the content of this chapter to conceive the method of system design.

3. Approach

Since this online education platform is a case study, I need to adopt a qualitative research method. I studied every function of systems and the significance of the existence of each page by observing the existing online education platforms recognized in different countries introduced in related work. Qualitative research explores current events and discovers patterns when we have no opinion about something [5]. In the second chapter, I describe and compare online education platforms in three different regions. However, due to the limited time, I could not conduct historical and development research on them. All of the data and results in the second chapter were obtained through my observation and analysis. In addition, I may have some personal will in the selection process of some functions in the education platform. Due to limited time, I may omit unimportant functions that I cannot realize in this project, so there is a particular subjective bias.

Regarding the choice of web frameworks, Django is an advanced Web application framework that supports dynamic Web development. It is designed to facilitate rapid development and clean, functional design to build high-performance, elegant Web applications [6] quickly. Django is an excellent tool for beginners who want to learn about Web development practically [7].

Learning to use Django as a website development tool is a great choice, so I chose Django as this online education platform. Hourirh introduced detailed teaching steps in the book, which gave me a clear and quick understanding of the steps of website construction. The functional design of teaching cases is relatively comprehensive, and there are teaching codes and abundant electronic resources. The second reference book in this web development series [8] introduces Django's advantages and reasons for web development, providing theoretical support for developing a Django-based online education platform. Jeff describes the environment for different systems to build and use different types of databases. In this book, I will focus on URL design, model design, and template design. As a third reference book, I will refer to the book "Django 3 By Example: Build powerful and reliable Python web applications from

scratch" on how to use newer materials to modify technical updates during development and the authors' design recommendations, such as model design. The book has the advantage of being relatively new, with some space devoted to designing an online education platform. I can innovate the author's original ideas and finally develop this online education platform. Finally, online distance education faces virus infection; Illegal database access; Account and PIN theft; Information disclosure; Investigation and research on remote attacks, and other problems [9]. The author emphasizes that whether distance education meets the security requirements is one of the core issues for developing online education platforms. I will refer to the security protection principles to provide security protection suggestions for the online platform.

About development tools, The main reason for using PyCharm as a program development tool in this project is that PyCharm has tools that support HTML and JavaScript, database tools, SQL terminal, and data export function [10]. The main reason is that PyCharm comes with a built-in Django framework suitable for this project.

Navicat was chosen as the database development tool because Navicat can connect to up to seven databases simultaneously: MySQL, MongoDB, SQLite, etc. In this project, the database tool is MySQL. When the data is updated in MySQL, Navicat can quickly complete data transfer, data synchronization, and structure synchronization so that I can easily and quickly migrate the data at a low cost.

These are why I chose to use Django as the Web framework and Pycharm and other tools for development in this GMOOC project. Below I will compare the design planning of the system to the other part of the description. This chapter focuses on realizing the functions observed from the existing online education platform and my ideas, that is, how to make GMOOC from scratch, from 0 to 1.

I divided the development process into the following steps: 1. Select reference books and development tools. 2. System requirement analysis. 3. The software design. 4. Programming. 5. Unit test.

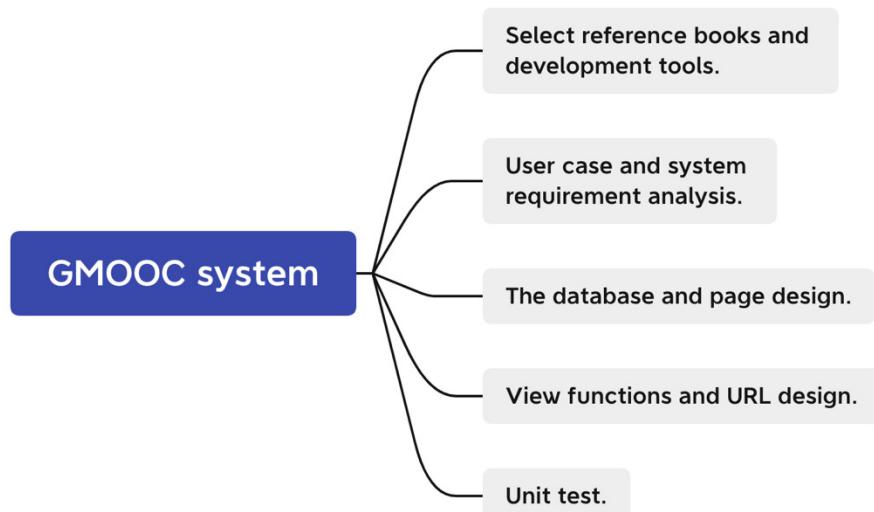


Figure 3.1 System development steps

In chapter 2 related Work, I analyzed three existing online education platforms' functions and front-end page design. Chapter 4 will conduct user case analysis and requirement document design for this online platform development project based on the summarized experience and the data download function I want to add this time.

I collectively refer to the four core modules in this project as apps: courses, Operation, organization, and Users. In a GMOOC project, all user activities are organized around these four core components. For example, I will define the course list, course details, specific chapters, user comments, and other functions in the course app.

I will consider displaying course lists, institution ranking, page paging, institution introduction, user favorites, and other functions in the organization APP. In the Users APP, I will mainly design user login, registration, user center, and other parts; Operation App is primarily responsible for recording users' Operation records. The function of this app is to record users' comments, favorites, and other behavior records. Chapter 5 will introduce the database design of the four core modules and the design results of the main HTML pages except for inheritance.

After the database and front-end page are written, I will introduce the view function and URL design on the Implementation page in chapter 6. The view function is responsible for designing the functional logic between the contents of the database and the front page, while the URL connects the two.

Finally, in Chapter 7's unit tests, I will unit test the project against the requirements document in Chapter 5. The manual test was that it took too long to prepare the system for the training of the ethics committee and to prepare the questionnaire. Another reason for adopting a unit test is that the data input by the unit test in the system is my data input, which can ensure the information leakage of others and ensure data security to a certain extent.

In the unit test, the first step is to deal with the bugs in the system and adjust the bugs until the online education platform can run smoothly. The second step is to adjust the front-end page and ensure that the page redirect is standard. The third step is to test each function of the system according to the requirements document to ensure the regular operation of functions such as login, registration, global search, user comments, and background administrator information.

4. Database Model

4.1 EER model

To enable the necessary and typical e-commerce functionality, such as storage and retrieval of

user, course, operation and organization, a database was implemented, based on the Enhanced Entity – Relationship (EER) model shown in Figure 4.1.

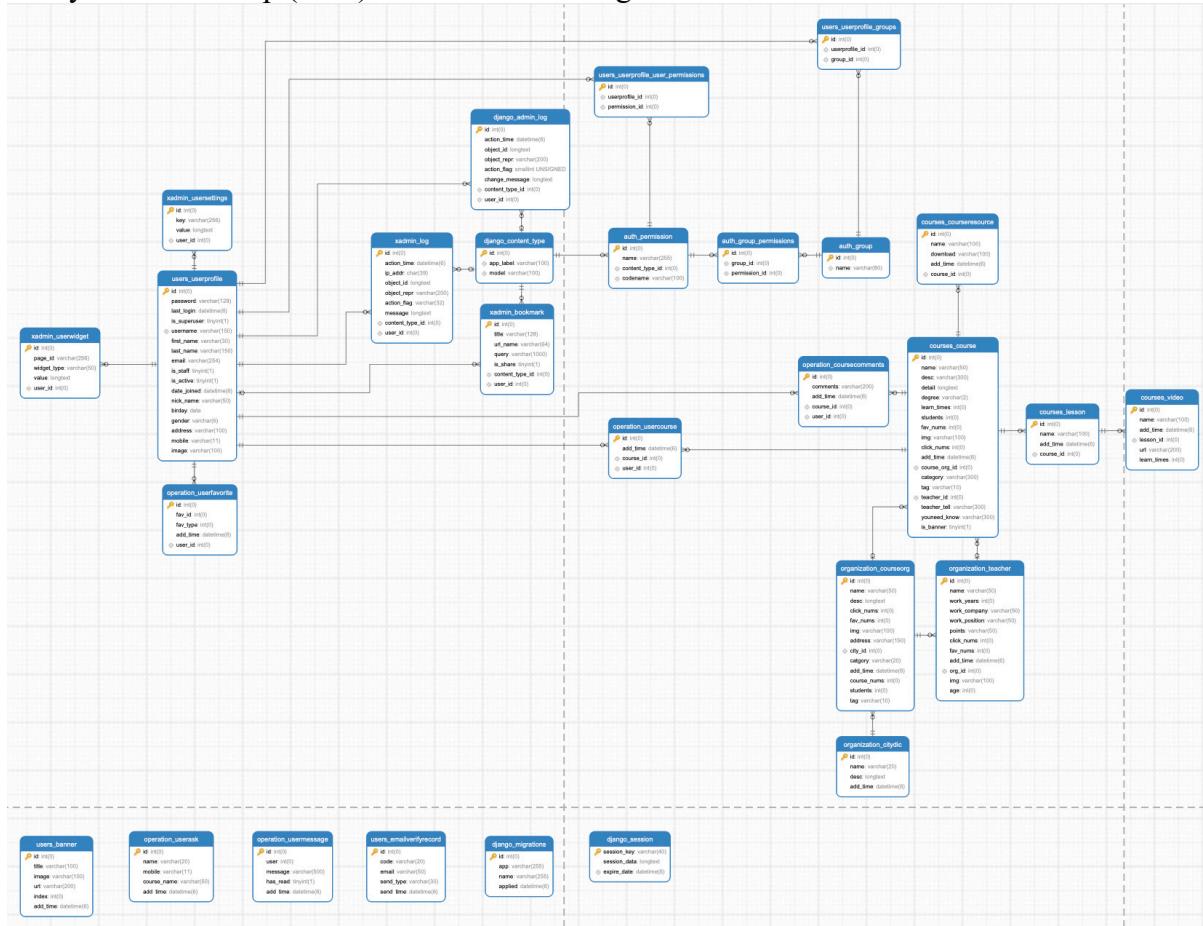


Figure 4.1:EER Model for GMOOC Database

4.2 Database design

In this chapter, I will start to design the front-end pages of the database and their corresponding functions for the four core modules of the project: courses, Operation, organization, and Users.

4.2.1 users

The user table stores user information, such as the user's nickname, address, and profile picture. These tables are used for login, registration, information modification, and saving user messages. The code is shown in Appendix 1.1

4.2.2 Courses

The course table will store course information, such as chapter information, course resources, etc. These forms are mainly used to introduce the course content in detail and provide users

with learning resources as the primary function of an online education platform. The code is shown in Appendix 1.2.

4.2.3 Operation

The Operation table stores user's behavior information, such as the messages posted by the user, the courses attended by the user, and the courses liked by the user. The code is shown in Appendix 1.3.

4.2.4 Organization

The organization table is mainly used to store information such as organization city, organization name, organization description, organization teachers, etc. The code is shown in Appendix 1.4.

5. System requirement

By observing and summarizing the functions of different online education platforms in Chapter 3 related Work, I designed the requirements document of this online education platform, which will also serve as a reference for the unit test after the completion of the system.

5.1 Requirement document

Functional Requirements

1.home page
1.1 The navigation bar allows you to quickly view the home page, open class, lecturers and teaching institutions.
1.2 Global search, you can search specific courses
2.Sign in
2.1 The user can log in
2.2 The user can log out
3.Registration:
3.1 Registration form is shown when the user arrives at the registration page
3.2 If the user uses an incorrect mailbox, an error message is displayed.
4.Personal center
4.1 Users can modify their personal information (nickname, profile picture, birthday, address, email, etc.)
4.2 Users can view the courses they are working on
4.3 Users can view the institution, the lecturer and the specific course, and can delete the contents

4.4 Users can reset their passwords
4.5 Users can view personal messages
5.Open Class Page
5.1 Users can rank courses
5.2 Users can view recommended popular courses
5.3 paging
6.Course Details Page
6.1 Users can click to start the course and enter the chapter information of the course
6.2 Users can bookmark and cancel classes
6.3 Rich text shows the details of the course, explaining the main content of the course
7.Course chapter information page
7.1 Users can view sections of the course
7.2 Users can comment on the course
7.3 Users can download the data
7.4 Users can view the instructor's tips for the course
7.5 Video chapter page for video playback
8.Lecture Lecturer page
8.1 Users can rank instructors by popularity
8.2 paging
8.3 Users can view a list of instructors
9.Teaching Institution Page
9.1 Users can filter the teaching institutions by category and region
9.2 Users can click to access the details page of the teaching institution
9.3 Users can rank courses by number of students and number of courses
9.4 Users can view the ranking of the institutions that taught them
10.Details page of the teaching institution
10.1 Users can view the entire curriculum of a specific institution
10.2 You can view the introduction of the organization
10.3 Users can view all the instructors in the organization
11.Background management system
11.1 The manager can set the theme of the background management system
11.2 User information: Administrators can manage user information, page rotation diagram
11.3 User operations: Administrators can manage and view users' courses, user messages, course comments and favorites.
11.4 Course management: administrators can add and delete courses, upload pictures and edit rich text on the new page of the course, and add chapters to the course; Filter courses according to fixed fields such as date range, course difficulty and course name; Export and import the course information in Excel,JSON and other formats
11.5 Authentication and authorization functions
11.6 Log recording function to record the modification operations of each background administrator

Non-Functional Requirements

12.Usability, presentation and Content
12.1 Ease of navigation (how easy it is to access every page from every other page)
12.2 Consistency of the design and professionalism in visual presentation

Table 5.1: User requirement Analysis

5.2 System design

Use Case Number: 1	Use Case Name: User registration	Rating: MUST
Purpose: Users must register an account to use the online education platform		
Main Actor: Users	Secondary Actor: N/A	
Pre-Conditions:		
<ul style="list-style-type: none"> ● User is on the home page 		
Trigger:		
<ol style="list-style-type: none"> 1. The User clicks the “registration” button on the homepage. 		
Description:		
<ol style="list-style-type: none"> 1. Users first need to decide whether to use the online education platform and then decide whether to register an account. 2. The user is presented with a input for their email. 3. The user is presented with a input for their username. 4. The next asks for a password and a password confirmation. 5. After the registration is successful, the login page is displayed. 		
Extension:		
<ol style="list-style-type: none"> 1. The email is already in use <ol style="list-style-type: none"> a. An error will be presented to the user and they will be asked to enter a valid email which is not already in use. 		
Related User-Cases:		
Post-Conditions:		
The user can also now log in to their account, whenever they choose.		
Author: Changchen Wang	Date: 06/09/2021	

Table 5.2: Use case 1 User registration

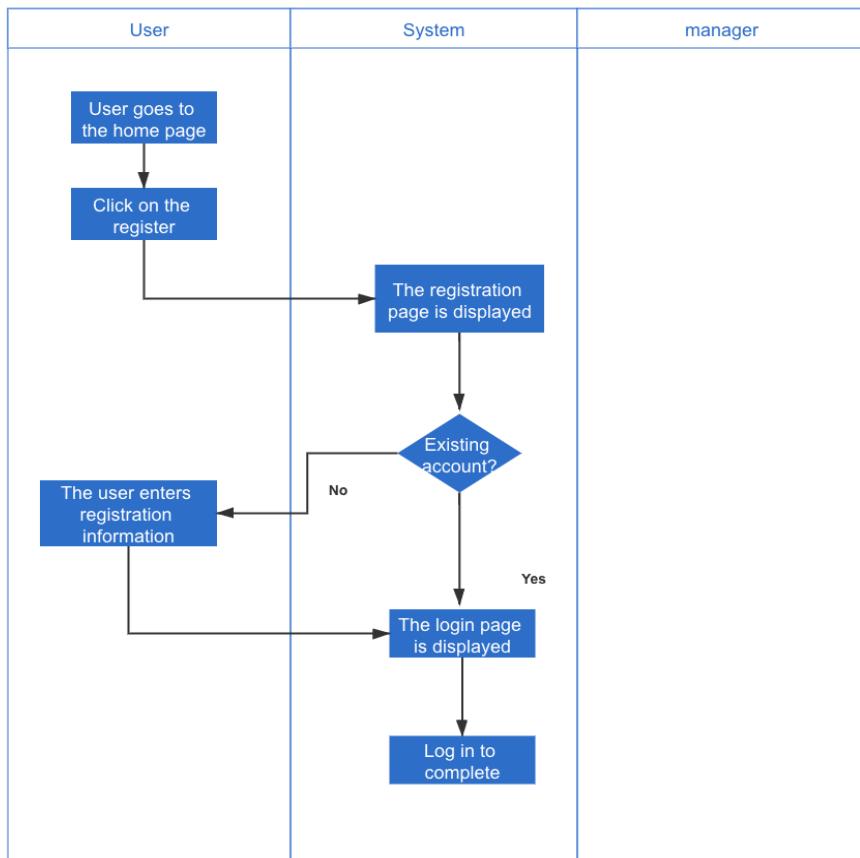


Figure 5.1 Flow chart for registration and login

Use Case Number: 2	Use Case Name: User Log in	Rating: MUST
Purpose: Users must be able to log in to their account		
Main Actor: Users Secondary Actor: N/A		
Pre-Conditions:		
<ul style="list-style-type: none"> ● Users have already started registering accounts 		
Trigger:		
2. The User clicks the “Log In” button on the homepage.		
Description:		
1. The user enters their username and password into the provided inputs and click “Log In”		
Extension:		
Related User-Cases:		
Post-Conditions:		
Users will be redirected to the home page, where they can select the courses, they want to		

take.	
Author: Changchen Wang	Date: 06/09/2021

Table 5.3: Use case 2 User Login

Use Case Number: 3	Use Case Name: View courses	Rating: MUST
Purpose: Users must be able to select the course they want to attend within the course.		
Main Actor: Users		Secondary Actor: N/A
Pre-Conditions:		
<ul style="list-style-type: none"> ● The user has registered an account. ● Users have logged in to their accounts. 		
Trigger:		
1. Click the course button in the navigation bar to enter the course details page.		
Description:		
1. On the course details page, users will see different course categories and list of courses from different educational institutions. Users can then click on one of the courses to enter a specific teaching chapter.		
Extension:		
Related User-Cases:		
Post-Conditions:		
Users who are interested in a course can save the course to their personal center.		
Author: Changchen Wang	Date: 06/09/2021	

Table 5.4: Use case 3 view courses

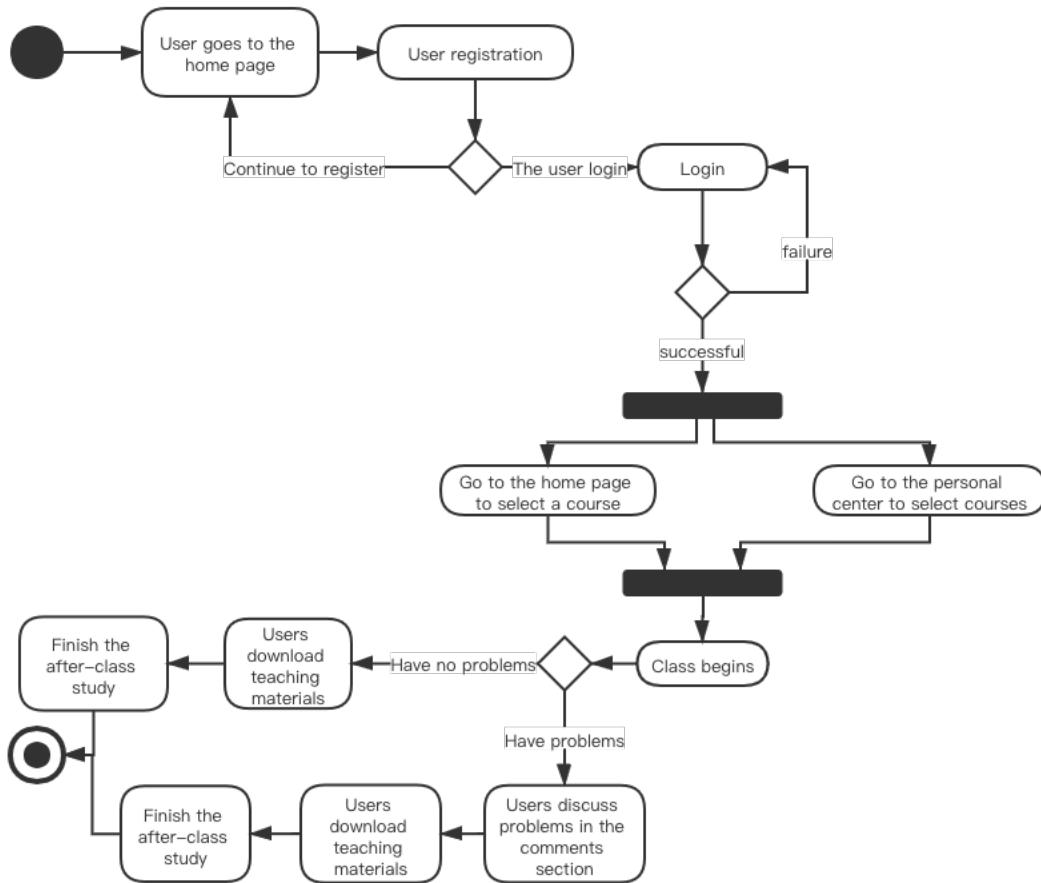


Figure 5.2 Flow chart for view courses, comments, download learning materials.

Use Case Number: 4	Use Case Name: Comments	Rating: MUST
Purpose: Users can comment at the bottom of the course if they have questions		
Main Actor: Users		Secondary Actor: N/A
Pre-Conditions: <ul style="list-style-type: none"> ● The user has registered an account ● User is logged in to their account. ● The user enters the course page and starts the class. 		
Trigger: <ol style="list-style-type: none"> 1. Users need to discuss in the comments section when they encounter problems in the course. 		
Description: <ol style="list-style-type: none"> 1. Users can input their own comments in the comment box, and also see other people's historical comments on the course chapters. 		
Extension: <ol style="list-style-type: none"> 1. Comments need to be effective <ol style="list-style-type: none"> a. Users cannot input invalid comments (e.g. empty). An error will be shown, and the user must correct this action. 		
Related User-Cases:		

Post-Conditions: After a user's comment is sent successfully, the user can view his or her own comment in the personal center.	
Author: Changchen Wang	Date: 06/09/2021

Table 5.5: Use case 4 comments

Use Case Number: 5	Use Case Name: Download learning materials	Rating: COULD
Purpose: Users can download the learning resources for this chapter next to the course chapter.		
Main Actor: Users		Secondary Actor: N/A
Pre-Conditions: <ul style="list-style-type: none"> ● The user has registered an account. ● Users have logged in to their accounts. ● The user enters the course details page and starts the class. 		
Trigger: <ol style="list-style-type: none"> 1. The user has completed the chapter teaching and needs additional learning materials 		
Description: <ol style="list-style-type: none"> 1. Users can download data by clicking the data download link on the right of the page. 		
Extension:		
Related User-Cases:		
Post-Conditions: Users who are interested in a course can save the course to their personal center.		
Author: Changchen Wang	Date: 06/09/2021	

Table 5.6: Use case 5 download learning materials

Use Case Number: 6	Use Case Name: Change personal information	Rating: MUST
Purpose: Users can modify their personal information (nickname, profile picture, birthday, address, email).		
Main Actor: Users		Secondary Actor: N/A
Pre-Conditions: <ul style="list-style-type: none"> ● The user has registered an account. ● Users have logged in to their accounts ● User is on the personal center page. 		
Trigger: <ol style="list-style-type: none"> 1. Users want to change their information. 		

2. The Users change the information and click the save button.	
<p>Description:</p> <ol style="list-style-type: none">1. The user is now presented with a modal window, showing their current personal information, including name, DOB and profile image.2. The user can either choose to change a value and click the “Update” button, saving their changes.	
<p>Extension:</p> <ol style="list-style-type: none">1. The new information is invalid, such as profile image is invalid.2. The user is shown an error.	
<p>Related User-Cases:N/A</p>	
<p>Post-Conditions:</p> <ul style="list-style-type: none">● If the update was successful, the user’s information has been updated.● Otherwise, the information remains as whatever it was before.	
Author: Changchen Wang	Date: 06/09/2021

Table 5.7: Use case 6 Change personal information

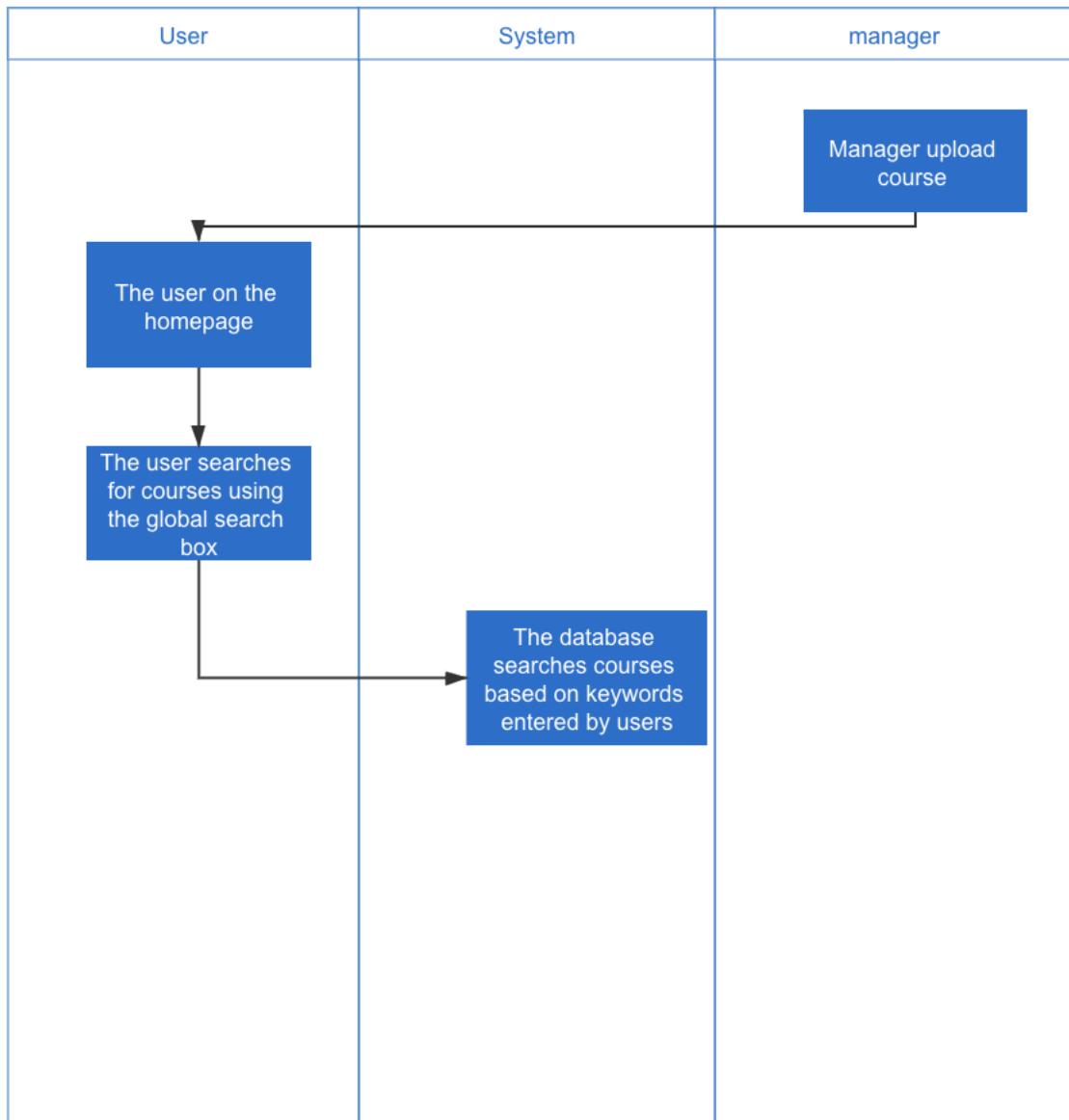


Figure 5.3 Flow chart for Global search

Use Case Number: 7	Use Case Name: Global search	Rating: MUST		
Purpose: Users can search for specific courses through the global search function				
Main Actor: Users	Secondary Actor: N/A			
Pre-Conditions:				
<ul style="list-style-type: none"> ● The user has registered an account. ● Users have logged in to their accounts ● The user clicks global search on the Home Page 				
Trigger:				
1. The User clicks the global search button.				

Description: 1. Enter a course name in the search box on the homepage.	
Extension:	
Related User-Cases: N/A	
Post-Conditions:	
<ul style="list-style-type: none"> When the user clicks the search button, the page jumps to a list of courses with the specified course name. 	
Author: Changchen Wang	Date: 06/09/2021

Table 5.8: Use case 7 Global search

Use Case Number: 8	Use Case Name: Navigation bar	Rating: SHOULD
Purpose: The navigation bar allows users to quickly view the homepage, open courses, teachers and organizations.		
Main Actor: Users		Secondary Actor: N/A
Pre-Conditions:		
<ul style="list-style-type: none"> The user has registered an account. Users have logged in to their accounts. The user is on the home page. The administrator has uploaded information about courses, teachers and organizations in the background management system. 		
Trigger:		
<ol style="list-style-type: none"> The User clicks the “Home page” button. The User clicks the “Open courses” button. The User clicks the “Teacher” button. The User clicks the “Organization” button. 		
Description:		
1. The navigation bar provides quick location.		
Extension:		
Related User-Cases: N/A		
Post-Conditions:		
1. After the user clicks the button in the navigation bar, the page will automatically jump to the specified list page, such as teacher list and organization list.		
Author: Changchen Wang	Date: 06/09/2021	

Table 5.9: Use case 8 Navigation bar

Use Case Number: 9	Use Case Name: View the lessons the user is learning	Rating: SHOULD
Purpose: users can view the courses they have started.		
Main Actor: Users		Secondary Actor: N/A
Pre-Conditions: <ul style="list-style-type: none"> ● The user has registered an account. ● Users have logged in to their accounts. ● User is on the personal page. ● The administrator uploaded the course information in the background, and the user clicked on the course details page to start learning. 		
Trigger: <ol style="list-style-type: none"> 1. The User clicks the “My courses” button. 		
Description: <ol style="list-style-type: none"> 1. The courses viewed by users on this page are the courses entered by users after clicking the "Start learning" button in the course details page. In other words, if users select a course and enter the detailed chapter page, they can query this course in the personal center. 		
Extension:		
Related User-Cases: N/A		
Post-Conditions:		
Author:	Date:	
Changchen Wang	06/09/2021	

Table 5.10: Use case 9 View the lessons the user is learning

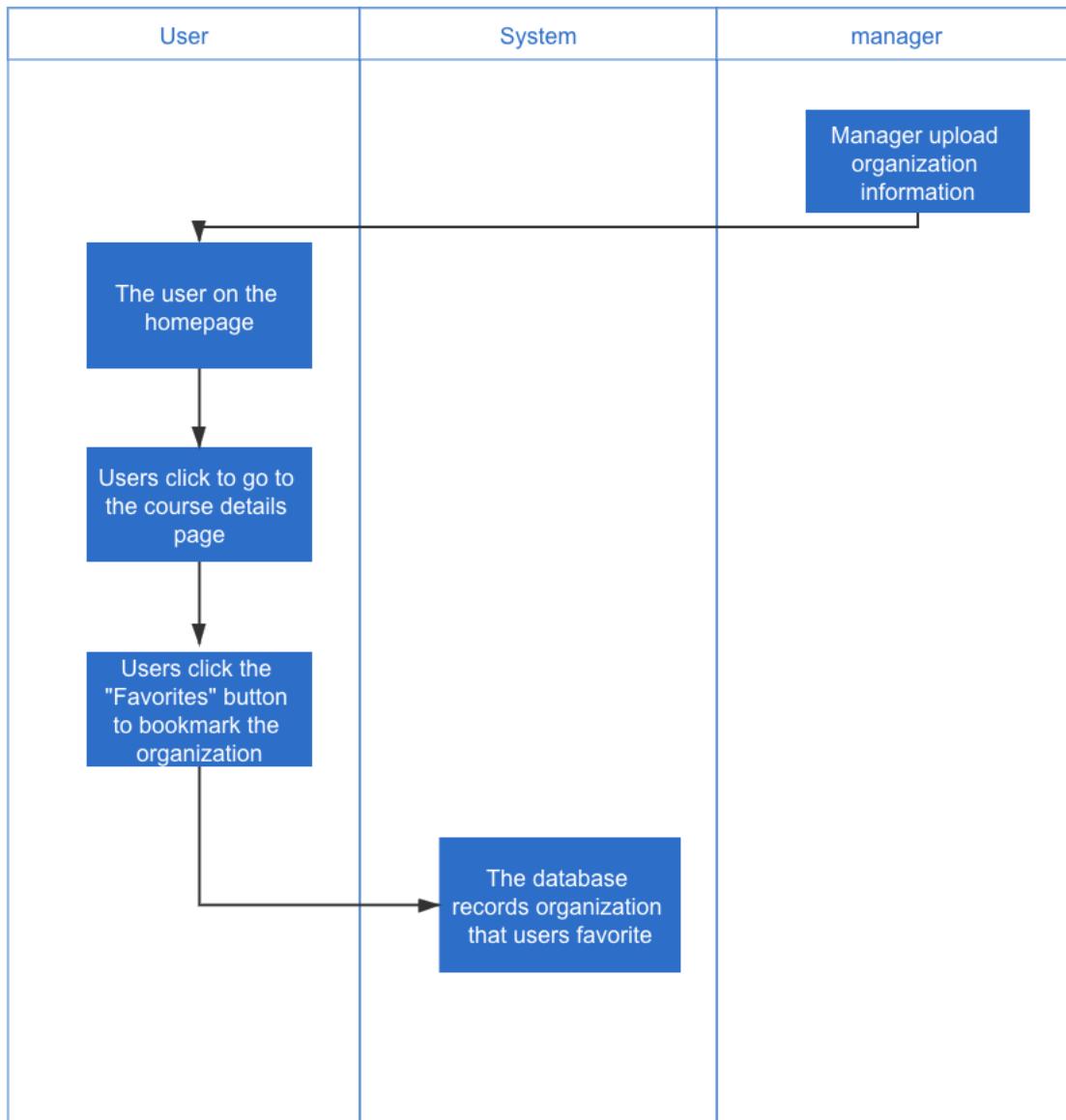


Figure 5.4 Flow chart for view organization

Use Case Number: 10	Use Case Name: Users view organization that have been bookmarked	Rating: SHOULD
Purpose: Users can view organizations they already favorite.		
Main Actor: Users	Secondary Actor: N/A	
<p>Pre-Conditions:</p> <ul style="list-style-type: none"> ● The user has registered an account ● Users have logged in to their accounts ● User is on the personal center page. ● The administrator has uploaded the course in the background, and users favorite the course. 		

Trigger: 1. The User clicks the “My collected” button.	
Description: 1. Users can see a star on the course details page, and they can choose whether to like this course or not. If the user chooses to bookmark a course, it will be displayed on the personal center page.	
Extension: 1. Users can choose to unfavorites.	
Related User-Cases: N/A	
Post-Conditions: ● If the user cancel favorites the course, the course name will also be undisplay in the personal center.	
Author: Changchen Wang	Date: 06/09/2021

Table 5.11: Use case 10 Users view organization that have been bookmarked

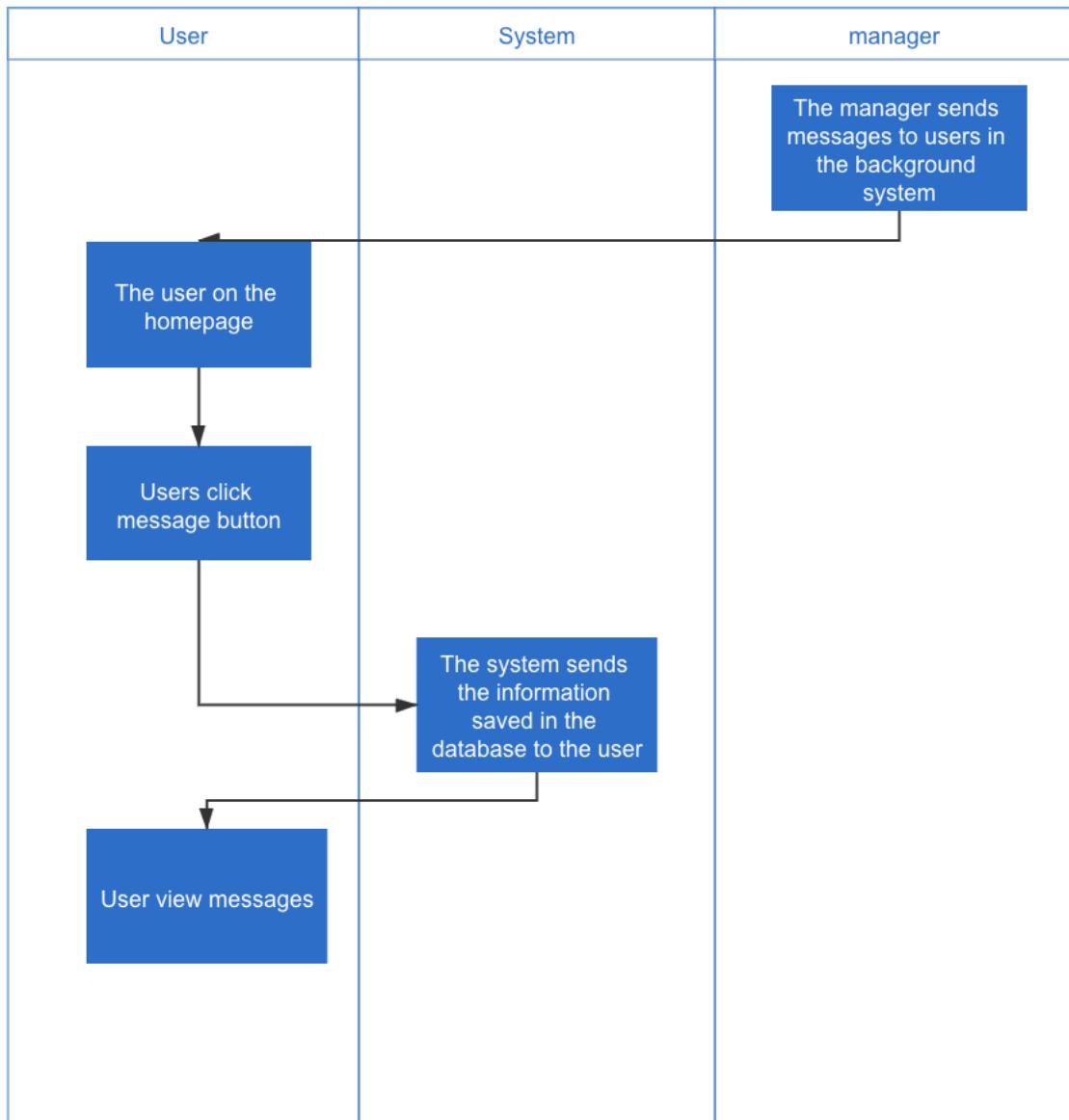


Figure 5.5 Flow chart for view message

Use Case Number: 11	Use Case Name: View my message	Rating: SHOULD		
Purpose: users can view their own messages.				
Main Actor: Users	Secondary Actor: N/A			
Pre-Conditions: <ul style="list-style-type: none"> ● An account has been registered ● Users have logged in to their accounts ● User is on the personal page. ● The administrator has sent messages to users in the background management system. 				
Trigger: <ol style="list-style-type: none"> 1. The User clicks the “My message” button. 				

Description: 1. When an administrator publishes system messages in the background management system, users can view them on the home page.	
Extension: Related User-Cases: N/A	
Post-Conditions:	
Author: Changchen Wang	Date: 06/09/2021

Table 5.12: Use case 11 View my message

Use Case Number: 12	Use Case Name: Course sorting	Rating: SHOULD
Purpose: Courses can be sorted by upload time and number of participants		
Main Actor: Users		Secondary Actor: N/A
Pre-Conditions: <ul style="list-style-type: none"> ● The user has registered an account ● Users have logged in to their accounts ● User is on the homepage. ● The background management system has uploaded the open course content and there are already people who favorite some courses. 		
Trigger: <ol style="list-style-type: none"> 1. The User clicks the “The Latest” or “The most popular”button. 		
Description: <ol style="list-style-type: none"> 1. There are two ways users can sort courses in the course details page. The first is based on the time of course uploading in the background management system, and the latest uploaded course is displayed first. The second ranking is based on the number of students who have clicked "Begin to Learn" on the course details page. 		
Extension: Related User-Cases: N/A		
Post-Conditions:		
Author: Changchen Wang	Date: 06/09/2021	

Table 5.13: Use case 12 Course sorting

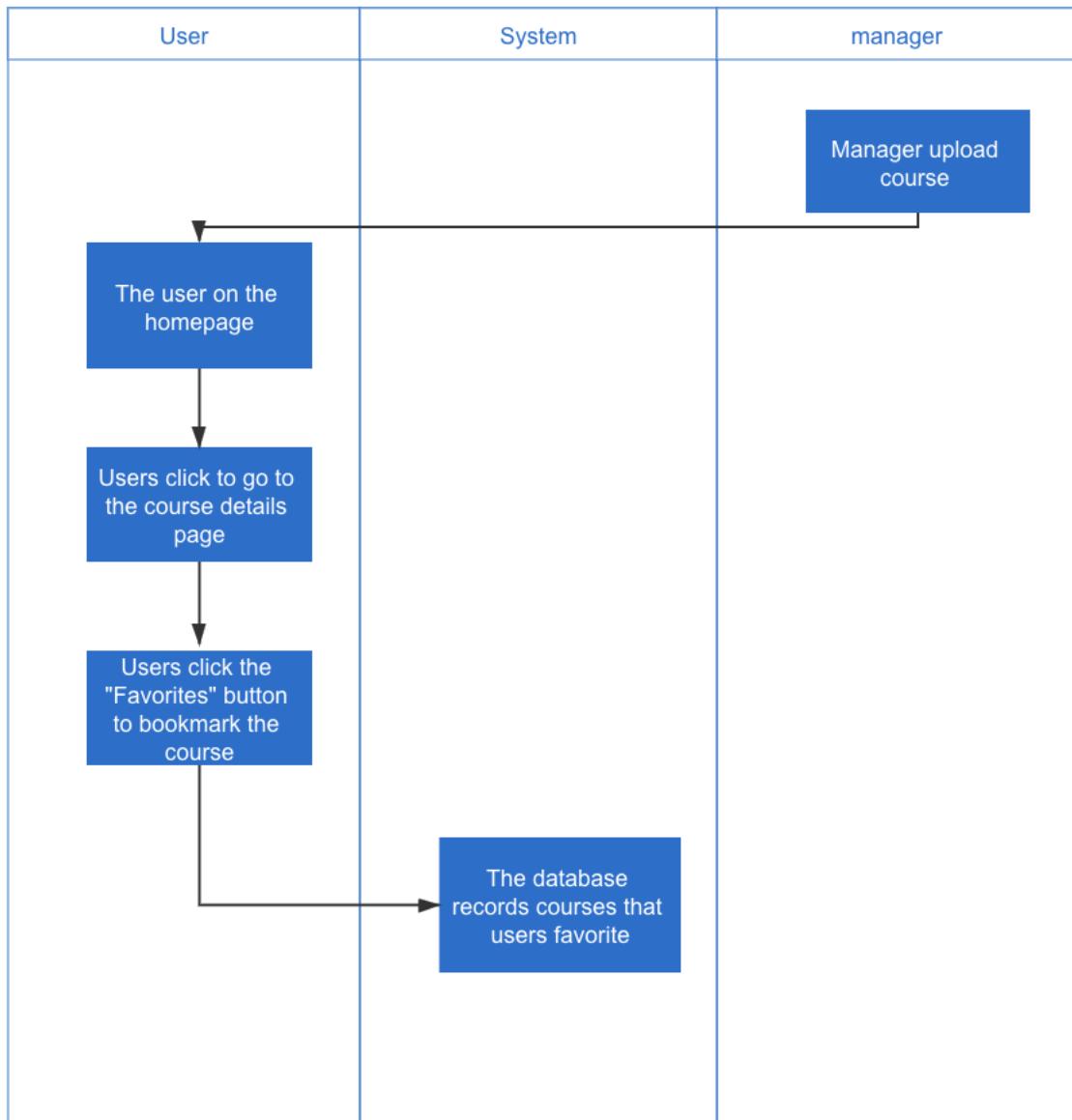


Figure 5.6 Flow chart for favorite courses

Use Case Number: 13	Use Case Name: Favorite courses	Rating: SHOULD		
Purpose: user can favorite courses.				
Main Actor: Users	Secondary Actor: N/A			
Pre-Conditions:				
<ul style="list-style-type: none"> ● The user has registered an account ● Users have logged in to their accounts ● User is on the Open courses page. ● The administrator has uploaded the course in the background system. 				
Trigger:				
<ol style="list-style-type: none"> 1. The User clicks the "Favorite" button. 				

<p>Description:</p> <ol style="list-style-type: none"> 1. Users can collect their favorite courses, and they can query the collected courses in the personal center. 	
<p>Extension:</p> <ol style="list-style-type: none"> 1. Users can choose to unbook mark courses. 	
<p>Related User-Cases:N/A</p>	
<p>Post-Conditions:</p>	
Author: Changchen Wang	Date: 06/09/2021

Table 5.14: Use case 13 Favorite courses

6. Implementation

After completing the database design of the online education platform, we need to design URL and HTML pages for different apps to ensure that the client can interact with the front-end page through these urls.

6.1 Html page design

In this project, I created 23 HTML pages to jump from page to page and redirect to ensure that the online education platform functions properly, this section provides a basic overview of 14 important page's functionality and usage.

6.1.1 Base.html

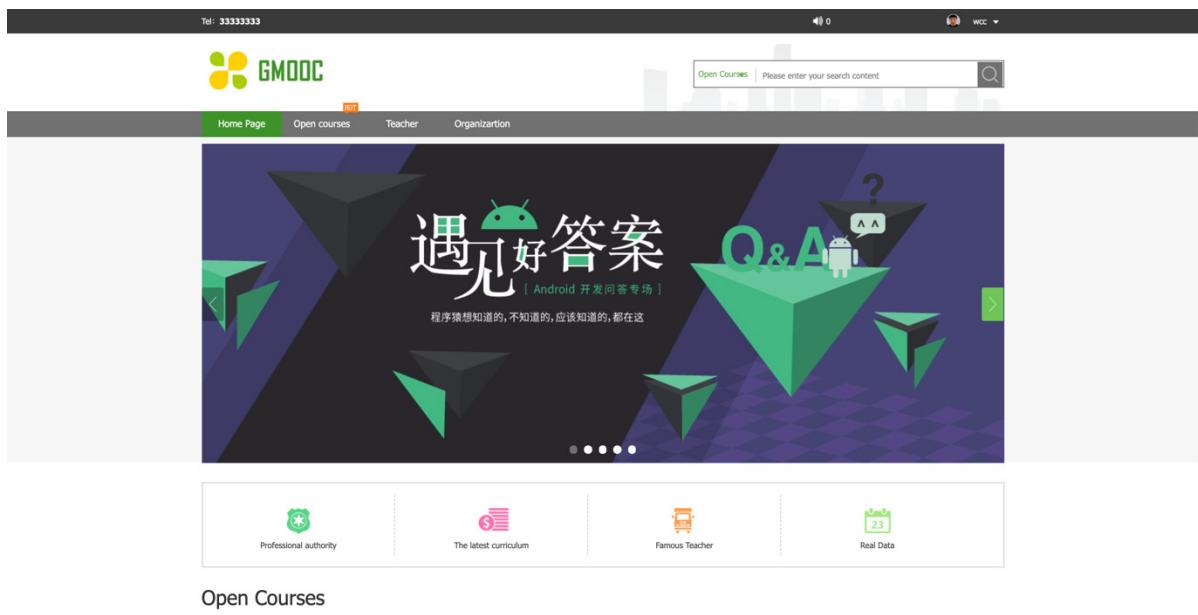


Figure 6.1:GMOOC homepage



Figure 6.2: GMOOC homepage

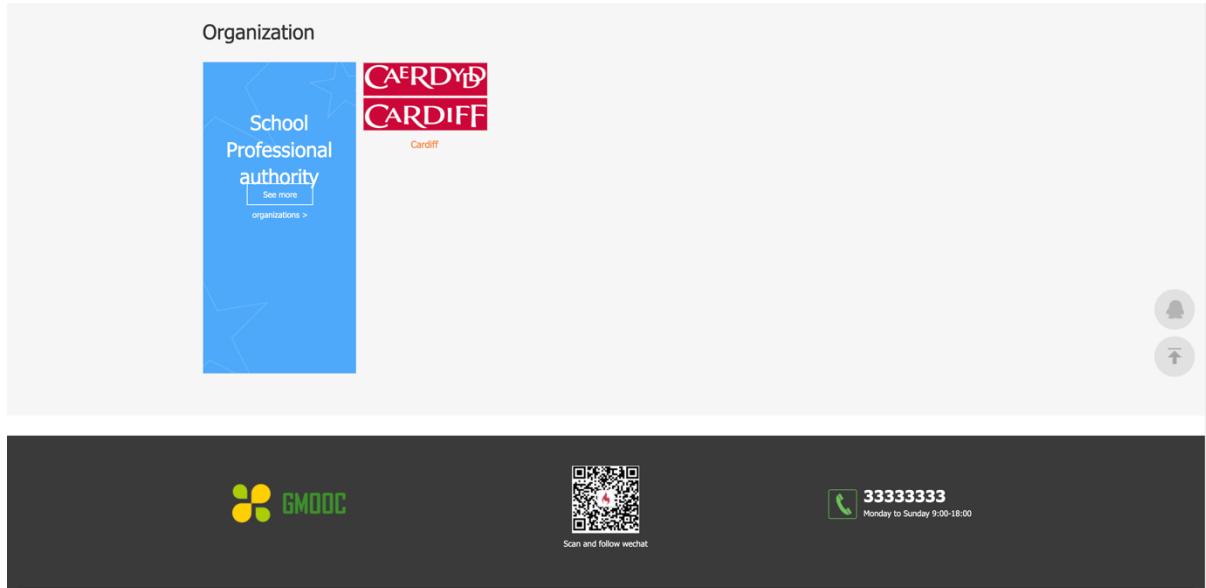


Figure 6.3 : GMOOC homepage

Base.html is the home page of this online education platform, and most of the other pages are inherited from base.html. The whole page has five parts: 1. In the Top part, there is the logo of the education platform, and the functions of login, logout, and entering the personal center are added. 2. The second part is the rote map area and course search area. Its primary function is to display the rote map added in the background and support users searching the designated course by using the course name and course organization. 3. The third part is the public course part. The primary function of this part is to display all the public course names and cover images added in the background, which is convenient for users to click on the course details page. 4. The fourth part is the Organization part, which mainly displays all types of educational organizations to connect to enter and view the detailed information

of all organizations. 5. The fifth part is at the bottom of the page. Its principal function is to display the developer information of the education platform.

6.1.2 Login.html



Figure 6.4: Login page

The Login page is used for logging in to the online education platform. The left area of the center part of the page is the rotation chart, showing some basic course information. The right side location is for users to enter login information. Users can log in by entering their registered email addresses and passwords. If the user enters an unregistered user name and password, an error message is displayed. Suppose the user has not registered their account before entering the login interface. In that case, they can click "Register Now" under the form to quickly direct to the registration page for account registration.

6.1.3 Register.html

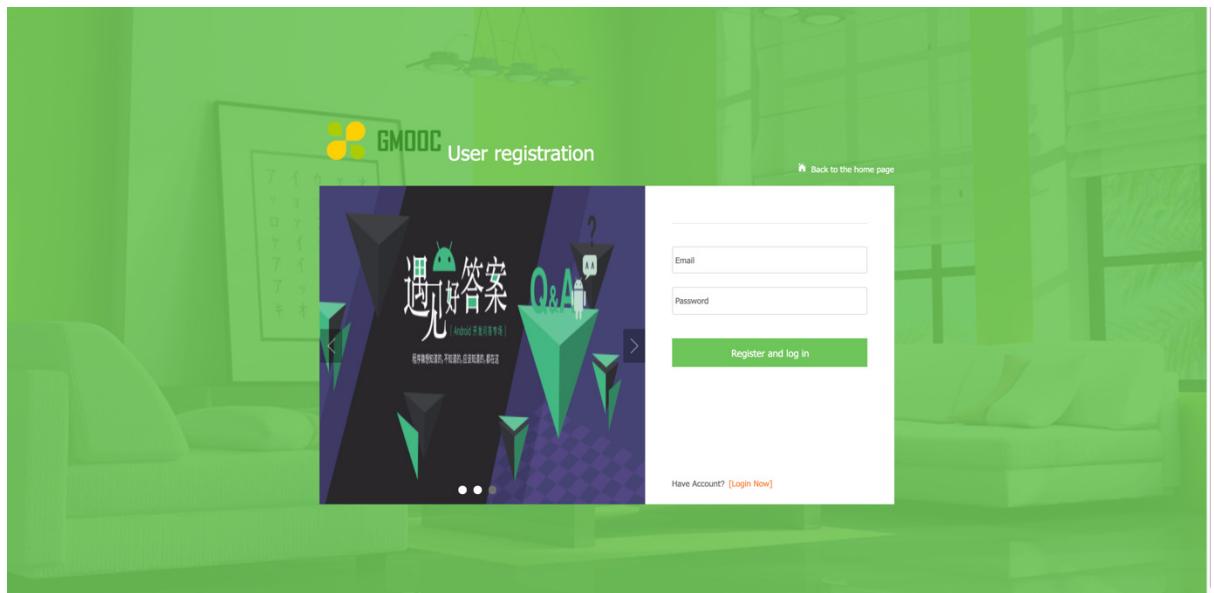


Figure 6.5: Registration page

The primary function of the Register Page is to provide a user registration service. The design logic of the login page remains the same. Users need to enter their email address and password on the page to complete registration. Note that if the mailbox is already used and stored in the database, you cannot complete registration with duplicate mailbox information and will get an error message.

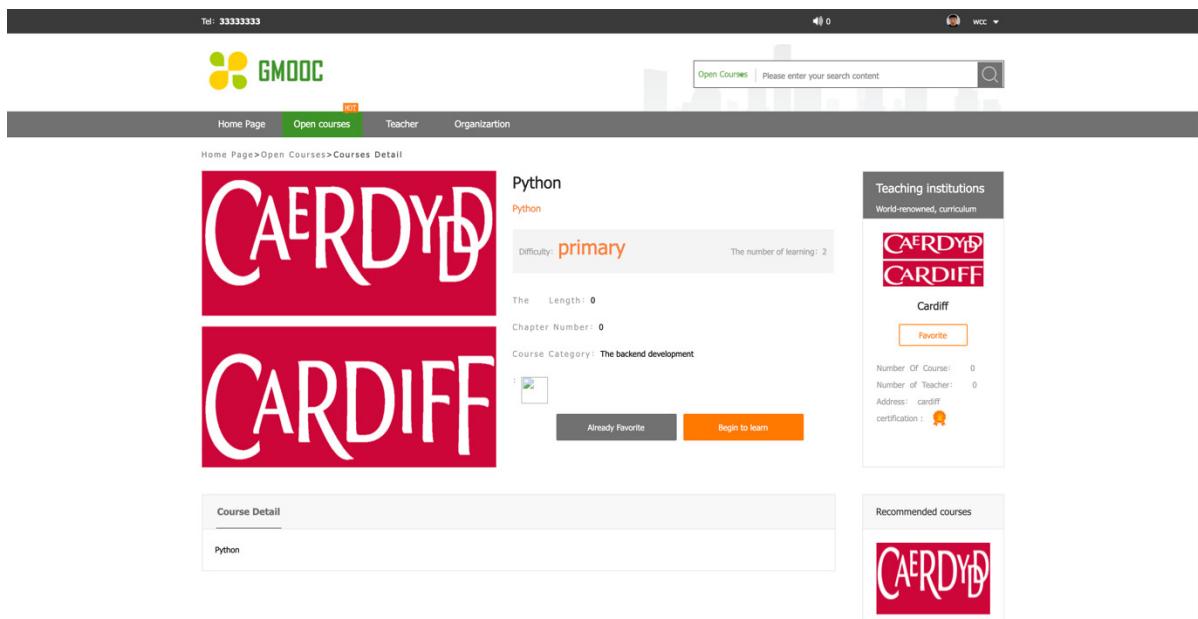
6.1.4 Course-list.html

A screenshot of the GMODC Course-list.html page. The page features a header with the GMODC logo, a search bar, and navigation links for "Home Page", "Open courses" (which is the active tab), "Teacher", and "Organization". Below the header is a breadcrumb navigation "Home Page > Courses Detail". The main content area displays a list of courses. On the left, a card for a course titled "CAERDYD CARDIFF" in red, with "Python" listed as the language, shows "Learn Times: 0" and "Course Students: 2". On the right, a similar card for the same course shows "Recommended Popular courses" and includes the text "Difficulty: primary". At the bottom of the page is a footer with the GMODC logo, a QR code, and contact information: a phone icon with the number "33333333" and the text "Monday to Sunday 9:00-18:00".

Figure 6.6: Course-list page

The Course-list page is responsible for displaying all the courses uploaded in the background management interface. It can be paginated at the bottom of the page when there are many courses.

6.1.5 Course-detail.html

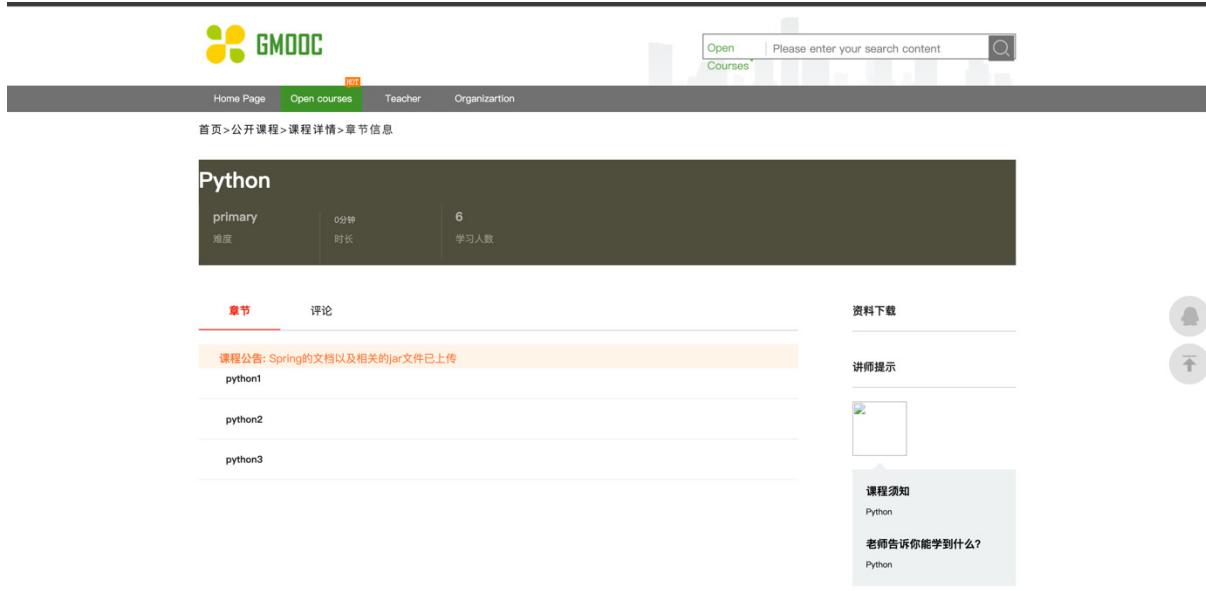


The screenshot shows a web browser displaying the 'Course-detail' page. The top navigation bar includes 'Home Page', 'Open courses' (highlighted in green), 'Teacher', and 'Organization'. A search bar with placeholder text 'Please enter your search content' is also present. The main content area features a large red banner with the text 'CAERDyD' and 'CARDIFF'. Below the banner, the course details for 'Python' are shown, including 'Difficulty: primary', 'Length: 0', 'Chapter Number: 0', and 'Course Category: The backend development'. There are buttons for 'Already Favorite' and 'Begin to learn'. To the right, a sidebar for 'Teaching institutions' shows 'CAERDyD CARDIFF' with 'Cardiff' selected, and provides statistics: 'Number Of Course: 0', 'Number Of Teacher: 0', 'Address: cardiff', and 'certification:'. A 'Favorite' button is also present. At the bottom, there are sections for 'Course Detail' (Python) and 'Recommended courses'.

Figure 6.7: Course-detail page

When the user selects a course through the Course List page and clicks on it, the web page will be directed to the course detail page. The course detail page mainly displays the primary information of the course to the user, such as course length, number of chapters, and course description. It also offers course collection and starts learning functions. The right side of the page also provides an area for viewing information about educational institutions. If users want to view specific information about educational institutions, they can click the link on the right side of the page.

6.1.6 Course-info.html



Python

primary
难度

0分钟
时长

6
学习人数

章节 评论

课程公告: Spring的文档以及相关的jar文件已上传

python1

python2

python3

资料下载

讲师提示

课程须知

Python

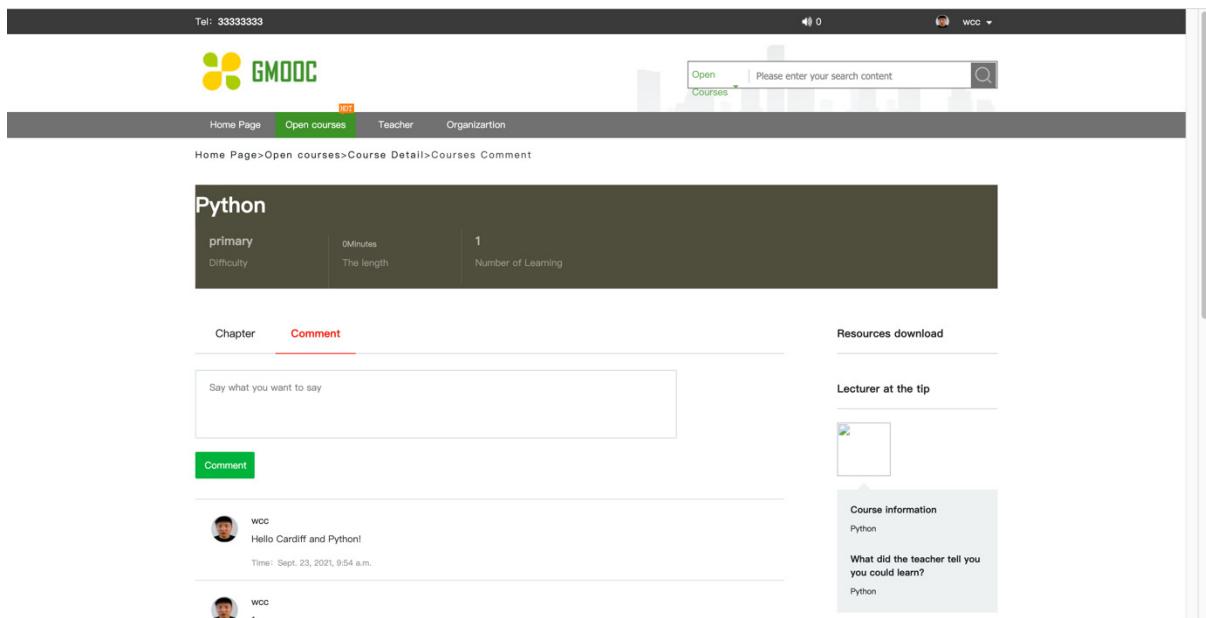
老师告诉你能学到什么?

Python

Figure 6.8: Course-info page

The Course Info page displays specific chapter information. Users can click on the chapter they want to learn. As shown in the figure, the user can choose between python1 and Python2 to learn. On the right side of the page, you can download the learning resources in the course and view some tips from the teacher for this course.

6.1.7 Course-comment.html



Python

primary
Difficulty

0Minutes
The length

1
Number of Learning

Chapter Comment

Say what you want to say

Comment

wcc Hello Cardiff and Python!

Time: Sept. 23, 2021, 9:54 a.m.

wcc 1

Resources download

Lecturer at the tip

Course information

Python

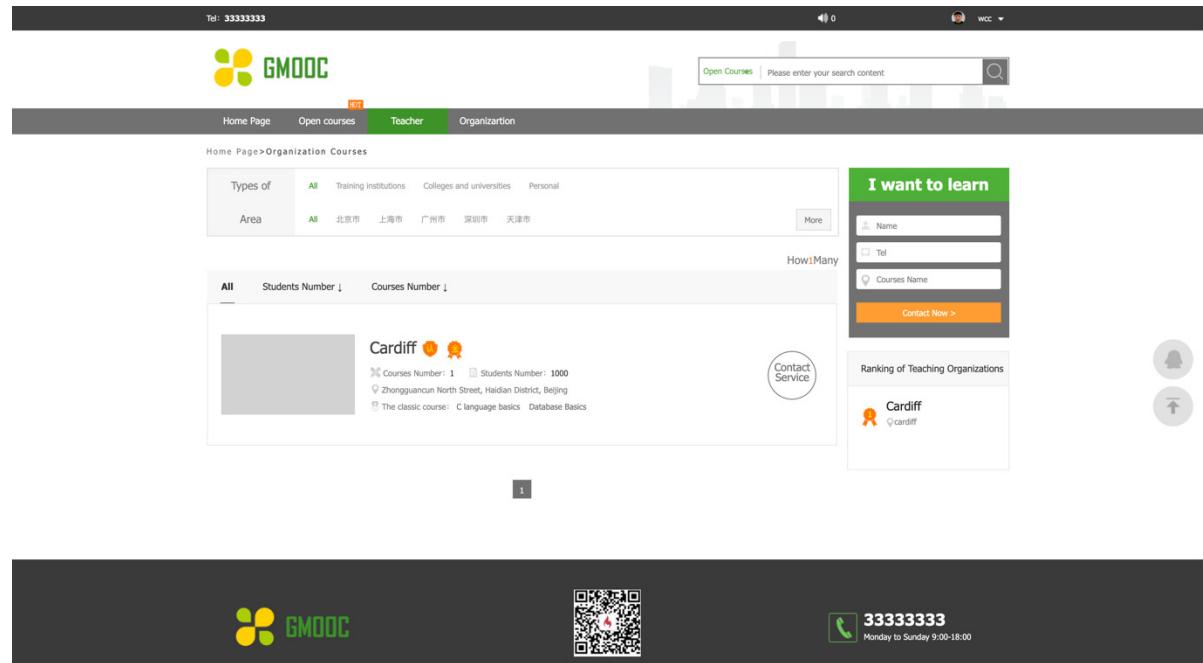
What did the teacher tell you you could learn?

Python

Figure 6.9: Course-comment page

The primary function of the Course-comment page is to allow users to comment on chapter contents in the Course. The upper part of the page inherits the navigation bar and search bar of base.html, the middle part is the primary comment function of the page, and the right-side part is the download area of course materials. If the administrator adds files in the background, users can click the link to download.

6.1.8 Teachers-list.html

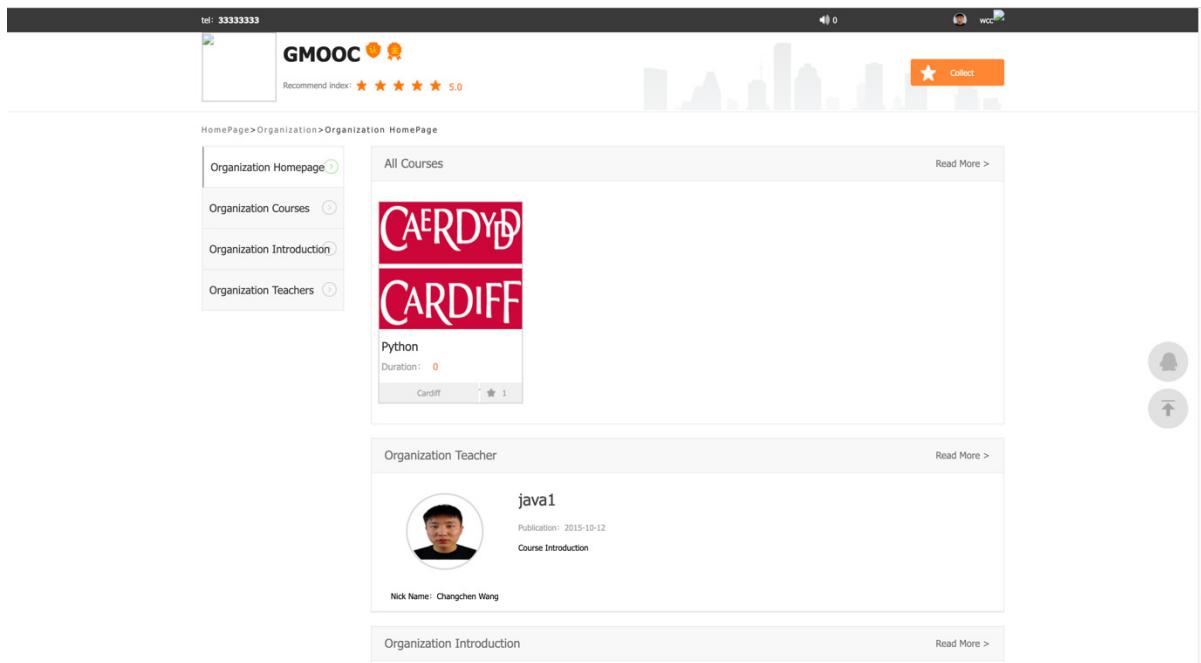


The screenshot shows the GMODC Teacher-list.html page. At the top, there is a navigation bar with links for 'Home Page', 'Open courses', 'Teacher' (which is the active tab, indicated by a red box), and 'Organization'. A search bar is also present. The main content area displays a list of teachers, with 'Cardiff' highlighted. The 'Cardiff' entry includes a profile picture, two orange status icons, and text indicating 'Courses Number: 1', 'Students Number: 1000', 'Zhongguancun North Street, Haidian District, Beijing', and 'The classic course: C language basics Database Basics'. To the right of the list is a 'Contact Service' button. On the right side of the page, there is a sidebar titled 'I want to learn' with fields for 'Name', 'Tel', and 'Courses Name', and a 'Contact Now >' button. Below this is a 'Ranking of Teaching Organizations' section showing 'Cardiff' at the top. At the bottom of the page, there is a footer with the GMODC logo, a QR code, and a phone number '33333333' with the note 'Monday to Sunday 9:00-18:00'.

Figure 6.10: Course-comment page

The teacher-list page classifies all teachers by educational institution. If the user wants to query the details of a teacher at a particular institution, the user can click to enter.

6.1.9 Org-detail-homepage.html

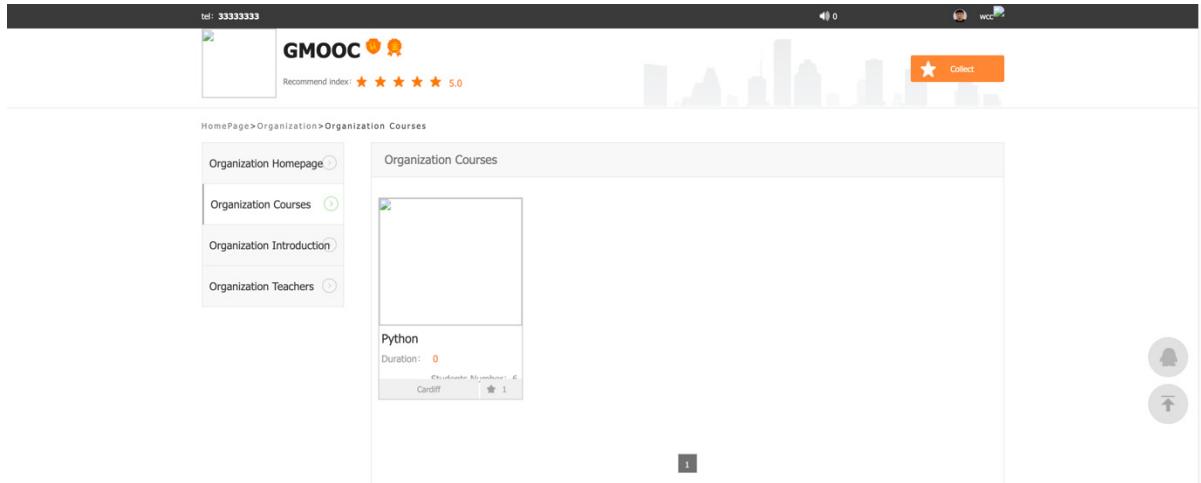


The screenshot shows a web application interface for an organization. At the top, there is a header with a phone number (tel: 33333333), a logo for 'GMOOC', a recommendation index of 5.0, and a 'Collect' button. Below the header, the URL 'HomePage>Organization>Organization HomePage' is visible. The main content area is divided into three sections: 'Organization Homepage' (with a link to 'Organization Courses'), 'Organization Courses' (listing a course titled 'Python' by 'Cardiff' with a duration of 0), and 'Organization Teacher' (listing a teacher named 'java1' with a profile picture, publication date of 2015-10-12, and course introduction). There are also 'Organization Introduction' and 'Read More >' links. On the right side, there are two circular icons: a bell and an upward arrow.

Figure 6.11: Org-detail-homepage

After the user selects an organization in the organization list, the first part of the org-detail-home page displays all the courses in this organization. The second part shows the list of all the teachers on this organization page. The last part is the description of the institution.

6.1.10 Org-detail-course.html



The screenshot shows a web application interface for an organization. At the top, there is a header with a phone number (tel: 33333333), a logo for 'GMOOC', a recommendation index of 5.0, and a 'Collect' button. Below the header, the URL 'HomePage>Organization>Organization Courses' is visible. The main content area is divided into three sections: 'Organization Homepage' (with a link to 'Organization Courses'), 'Organization Courses' (listing a course titled 'Python' by 'Cardiff' with a duration of 0), and 'Organization Teacher' (listing a teacher named 'java1' with a profile picture, publication date of 2015-10-12, and course introduction). There are also 'Organization Introduction' and 'Read More >' links. On the right side, there are two circular icons: a bell and an upward arrow.

Figure 6.12: Org-detail-course page

The org-detail-courses primary function is to display all the courses of the institution selected by the user.

6.1.11 Org-detail-desc.html

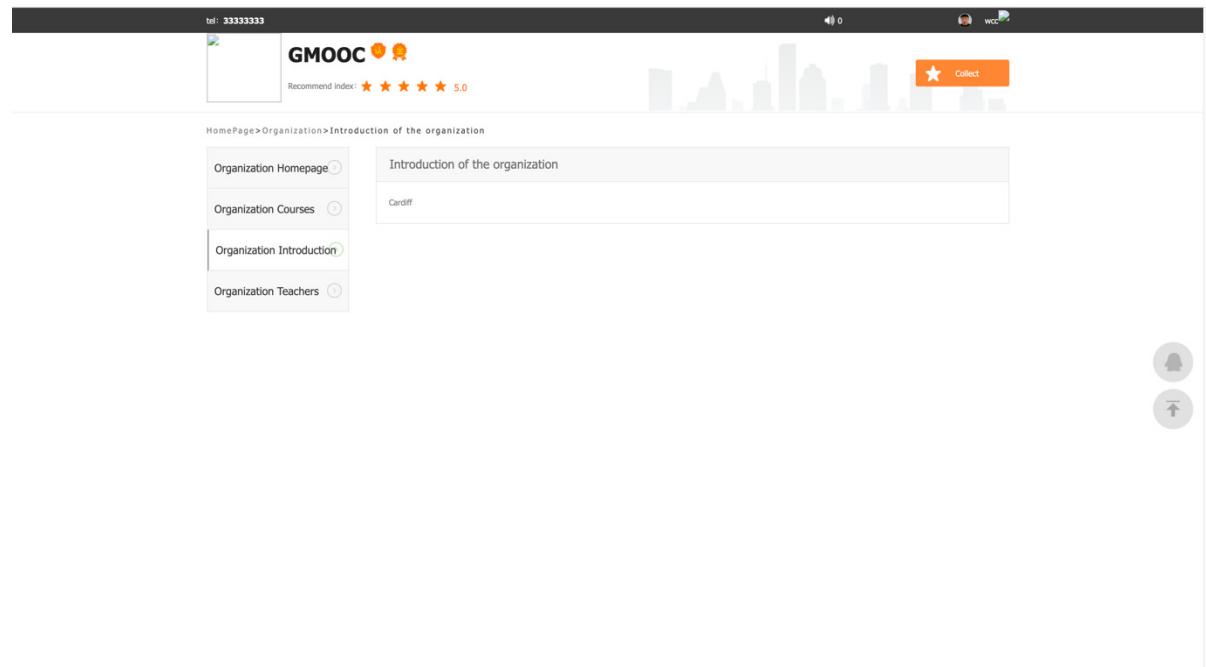


Figure 6.13: Org-detail-desc page

This page is responsible for presenting the introduction of the organization selected by the user.

6.1.12 Usercenter-info.html

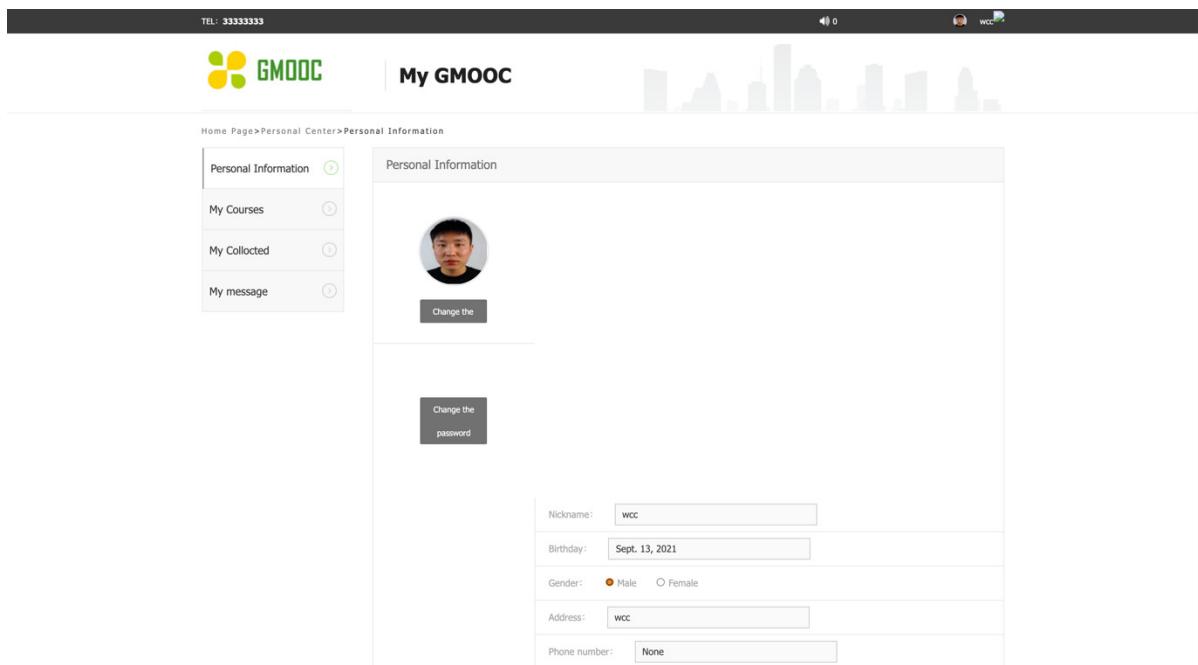


Figure 6.13: User center-info page

User center-info Page Displays basic information about the user, such as the user profile picture, nickname, date of birth, gender, address, and mobile phone number. Users can change their personal information on this page.

6.1.13 Usercenter-fav-course.html

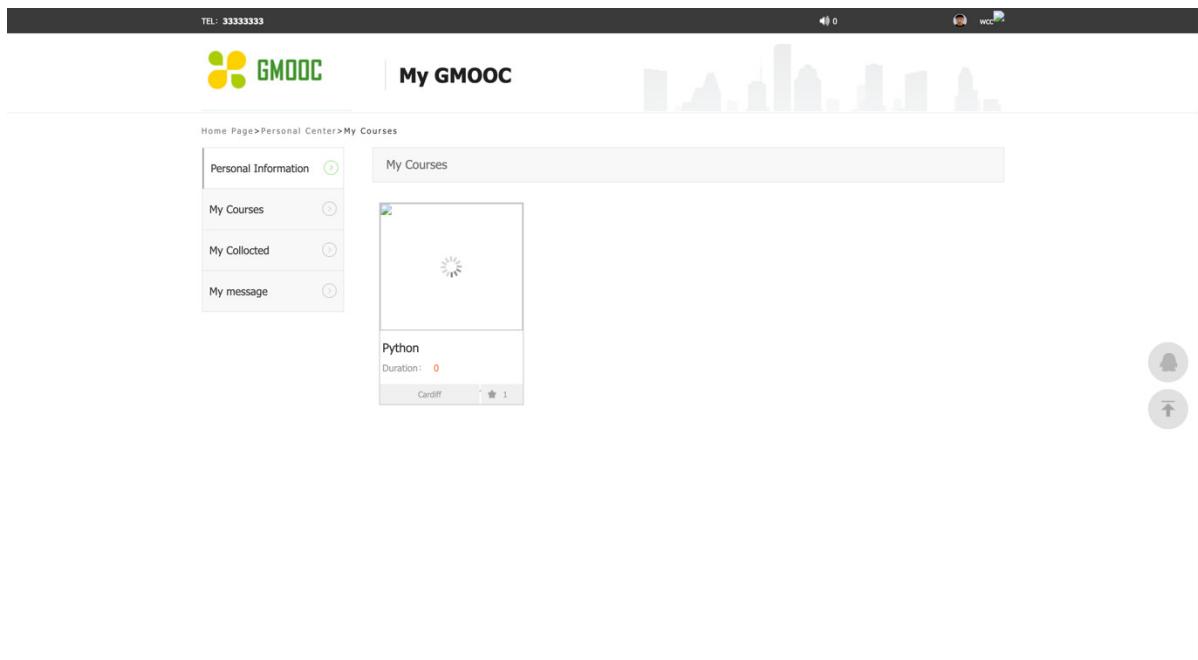


Figure 6.14: User center-fav-course page

The user center-fav-Course page allows users to view the courses they have bookmarked in the course list. If users click the Favorites button on the course page, they can look up their favorites in the user center.

6.1.14 Usercenter-message.html

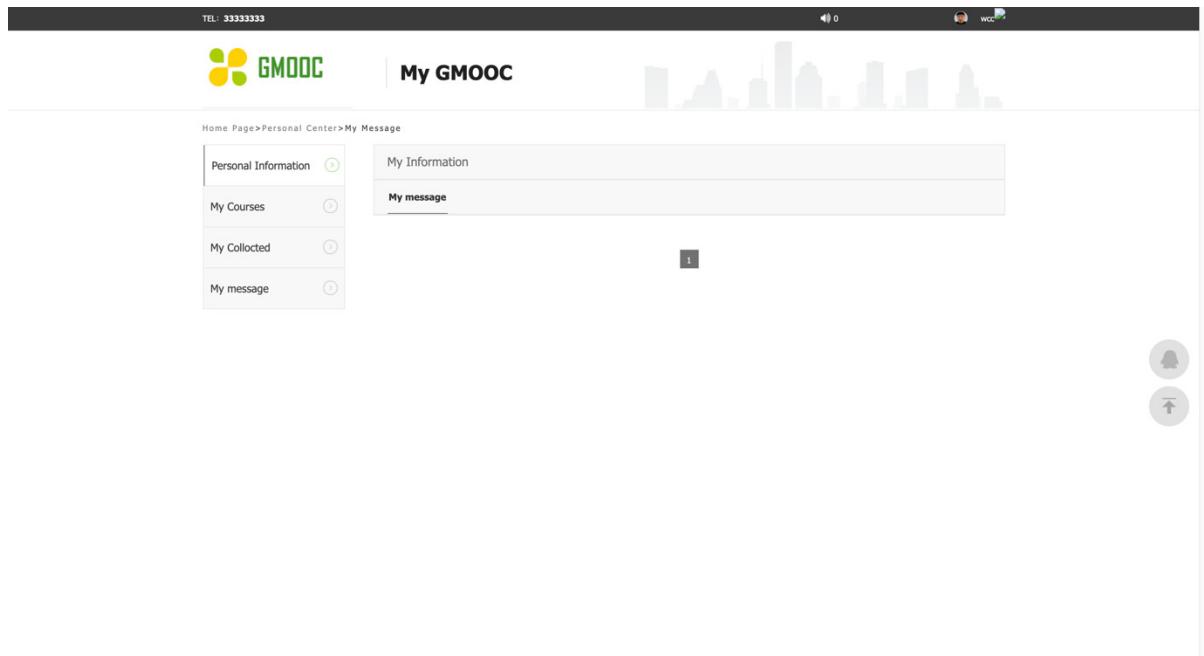


Figure 6.15: User center-message page

Users can view their published comments or received messages on the User center-message page. It should be noted here that the comments posted by users are spontaneously posted on the course chapter page, while users receive the user messages after the background management system uploads information by the administrator.

6.2 Views and Routes

6.2.1 Login/Registration

The login and registration pages are the first and second pages created in this project. We use `login/` and `register/` to route the login and registration pages in the URL configuration. After configuring the route, we need to go to the front-end page to find the location where the URL is required for the design process of the front-end page, that is, to jump to the login and registration page.

After linking the URL to the front page, the essential part is to write the background logic for user registration and login, which must be configured in the `views.py` file under the `user` folder. First, in the user registration part, we need to use the `METHOD` of `POST`; the purpose is to pass the user information instantiated into the database, But we need to determine whether the

user input is valid before it is passed in. The following is the background logic code for user registration. In this project, we use the `make_password` method in a package called Hashers to encrypt the password entered by the user.

```

class RegisterView(View):
    def get(self,request):
        register_form = RegisterForm()
        return render(request,"register.html",{"register_form":register_form})

    def post(self,request):
        register_form = RegisterForm(request.POST)
        if register_form.is_valid():
            user_name = request.POST.get("email", "")
            if UserProfile.objects.filter(email=user_name):
                return render(request, "register.html", {"register_form":register_form, "msg":"The user already exists"})
            else:
                pass_word = request.POST.get("password", "")
                user_profile = UserProfile()
                user_profile.username = user_name
                user_profile.email = user_name
                user_profile.is_active = True
                user_profile.password = make_password(pass_word)
                user_profile.save()

                # send_register_email(user_name, "register")
                return render(request, "login.html")
        else:
            return render(request, "register.html", {"register_form":register_form})

```

Figure 6.16: Register View

The user login function generally adopted is the "POST" method. Our idea is that when the user enters the account and password in the input box of the front page and clicks the Submit button, the system will send the information entered by the user to the database through POST for checking with the user information already saved in the database. Because we have a authenticate method in Django, we pass the username and password as arguments and return them to the user object. If the thing is None, we call Django's login method, passing the Request and user objects into the forbidden area. The user logs in successfully.

Account login

User name

Password

Login Now >

Don't have an online account? [\[Register Now\]](#)

Figure 6.17: Login windows

```

class LoginView(View):
    def get(self, request):
        return render(request, "login.html", {})
    def post(self, request):
        login_form = LoginForm(request.POST)
        if login_form.is_valid():
            user_name = request.POST.get("username", "")
            pass_word = request.POST.get("password", "")
            user = authenticate(username=user_name, password=pass_word)
            if user is not None:
                if user.is_active:
                    login(request, user)
                    return HttpResponseRedirect(reverse("index"))

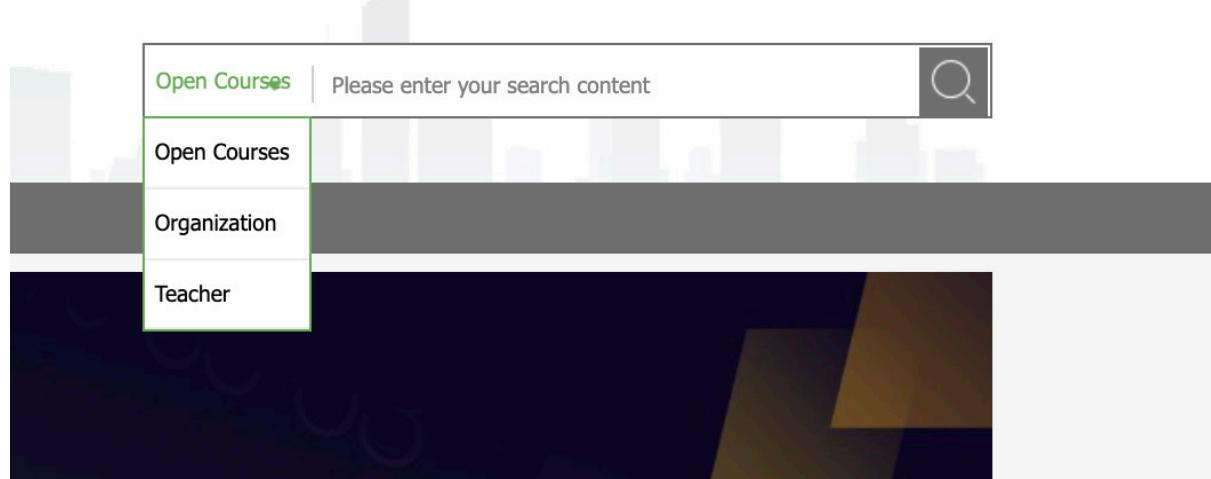
                else:
                    return render(request, "login.html", {"msg": "User not activated!"})
            else:
                return render(request, "login.html", {"msg": "The user name or password is incorrect!"})
        else:
            return render(request, "login.html", {"login_form": login_form})

```

Figure 6.18: Login views

6.2.2 Home page Global Search

At the top of the home page, we can search for specific open courses, institutions, and teachers. The design logic in this part is that we need to find the keywords we need from the database. If the keywords exist, we can search the place where the keywords appear by filtering. Here we will use the "Q" method to filter keywords. When calling the Q method, we can define multiple search contents simultaneously, and these contents will be retrieved as long as they contain the keywords, we enter in the search box.



```

class CourseListView(View):
    def get(self, request):
        all_courses = Course.objects.all().order_by("-add_time")
        hot_courses = Course.objects.all().order_by("-click_nums")[:3]

        #Search
        search_keywords = request.GET.get('keywords', "")
        if search_keywords:
            all_courses = all_courses.filter(Q(name__icontains=search_keywords)|Q(desc__icontains=search_keywords)|Q(detail__icontains=search_keywords))

```

6.2.3 User Personal Information

In the user profile function, we still design our routes in the urls under the Users folder and the view function under the Views folder. All the urls in this part are respectively to skip to

user information, user profile picture, view user courses and other functional pages, as shown in the figure:



```

urlpatterns = [
    #User info
    url('info/', UserinfoView.as_view(), name="user_info"),

    #User img
    url('image/upload/', UploadImageView.as_view(), name="image_upload"),

    #password
    url('update/pwd/', UpdatePwdView.as_view(), name="update_pwd"),

    #code
    #url('sendemail_code/', SendEmailCodeView.as_view(), name="sendemail_code"),

    #email
    url('update_email/', UpdateEmailView.as_view(), name="update_email"),

    #course
    url('mycourse/', MyCourseView.as_view(), name="mycourse"),

    #org
    url('myfav/org/', MyFavOrgView.as_view(), name="myfav_org"),

    #teacher
    url('myfav/teacher/', MyFavTeacherView.as_view(), name="myfav_teacher"),

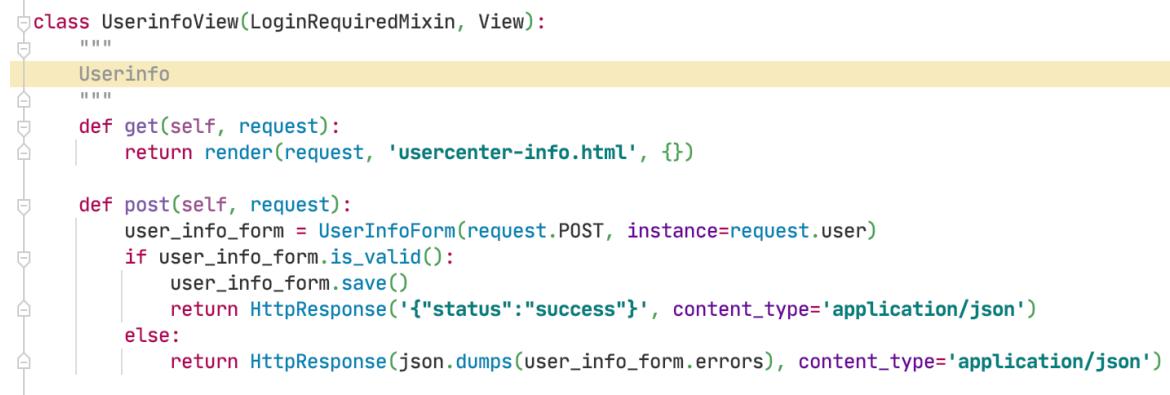
    #fav_course
    url('myfav/course/', MyFavCourseView.as_view(), name="myfav_course"),

    #message
    url('mymessage/', MymessageView.as_view(), name="mymessage"),
]

```

Figure 6.19: User center url

After requesting the user's personal information from the database via GET method, we return the information to the user center-info page. Suppose users want to modify their data (nickname, avatar, etc.) in the form. In that case, we still need to define post in the background logic to synchronize the user's modified information to the database. Similarly, we use the is_Valid() method to check whether the user has changed the information already in the database or specified the input value. If the modified data is valid, we save the new value to the database. Otherwise, we return an error message.



```

class UserinfoView(LoginRequiredMixin, View):
    """
    Userinfo
    """
    def get(self, request):
        return render(request, 'usercenter-info.html', {})

    def post(self, request):
        user_info_form = UserInfoForm(request.POST, instance=request.user)
        if user_info_form.is_valid():
            user_info_form.save()
            return JsonResponse({'status': 'success'}, content_type='application/json')
        else:
            return JsonResponse(json.dumps(user_info_form.errors), content_type='application/json')

```

Figure 6.20: User information view

6.2.4 See the course

When designing the course function, we still need to define all URLs related to the course, which are used to view the course list, details, comments, and information. As shown in the figure:



```
urlpatterns = [
    #list
    url('list/', CourseListView.as_view(), name="course_list"),

    #detail
    url('detail/(?P<course_id>\d+)/', CourseDetailView.as_view(), name="course_detail"),

    url('info/(?P<course_id>\d+)/', CourseInfoView.as_view(), name="course_info"),

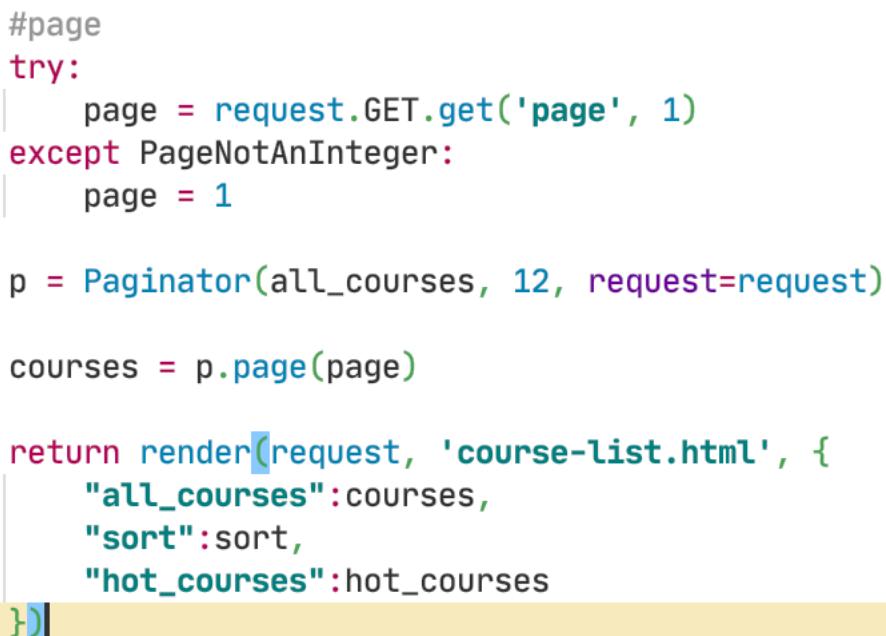
    #comment
    url('comment/(?P<course_id>\d+)/', CommentsView.as_view(), name="course_comments"),

    #add comment
    url('add_comment/', AddCommentsView.as_view(), name="add_comment"),

    url('video/(?P<video_id>\d+)/', VideoPlayView.as_view(), name="video_play"),
]
```

Figure 6.21: Course url

In the view function, we still need to use the get method to obtain course materials uploaded and saved in the database through the background management system and return the course information to the course list page by calling the model. Object to traverse all contents of the database. In addition, we implemented the Paginator function during the course design by calling the Paginator method. The specific design method and parameter design are shown in the figure below:



```
#page
try:
    page = request.GET.get('page', 1)
except PageNotAnInteger:
    page = 1

p = Paginator(all_courses, 12, request=request)

courses = p.page(page)

return render(request, 'course-list.html', {
    "all_courses":courses,
    "sort":sort,
    "hot_courses":hot_courses
})
```

Figure 6.22: Paginator function

6.2.5 Comments

In viewing the course function, we have configured the URL of the course comment in the URL file, so we need to complete the design of the view function to complete this part of the function. In the course comments section, we need to define two attempt functions. The first view function is mainly responsible for adding course comments. In this section, we need to use the post to save the comments added by the user to the database. First of all, users can check all course comments under their account to check course comments. Therefore, in the beginning, we need to determine whether the user has logged in. If not, a message will be returned telling the user that he needs to log in. After the user logs in, we need to determine whether the course ID is greater than 0 and whether the comment's content is not empty. If the course ID exists and the comment's content is not Null, the system will save the content of the comments in the database. The specific view function design logic is as follows:

```
class AddCommentsView(View):
    """
    comment
    """
    def post(self, request):
        if not request.user.is_authenticated:
            #login?
            return JsonResponse({'status': 'fail', 'msg': 'user have not login'}, content_type='application/json')

        course_id = int(request.POST.get("course_id", 0))
        comments = request.POST.get("comments", "")
        if course_id > 0 and comments:
            course_comments = CourseComments()
            course = Course.objects.get(id=int(course_id))
            course_comments.course = course
            course_comments.comments = comments
            course_comments.user = request.user
            course_comments.save()
            return JsonResponse({'status': 'success', 'msg': 'add successful'}, content_type='application/json')
        else:
            return JsonResponse({'status': 'fail', 'msg': 'add failure'}, content_type='application/json')
```

Figure 6.23: Comments view

The comments of the course are saved in the database by post on the course chapter page, so it is still necessary to obtain the information of the database through the get method, and then traverse the contents of the database through model.objects.all() to display them in the course-comment page. We need to define a view function called Comment View to retrieve user comments from the database. The view function is shown below:

```
class CommentsView(LoginRequiredMixin, View):
    def get(self, request, course_id):
        course = Course.objects.get(id=int(course_id))
        all_resources = CourseResource.objects.filter(course=course)
        all_comments = CourseComments.objects.all().order_by("-id")
        return render(request, "course-comment.html", {
            "course": course,
            "course_resources": all_resources,
            "all_comments": all_comments
        })
```

Figure 6.24: Comments view 2

6.2.6 Learning Resources download

The download of course Resources We have designed the Download field of FileField type in the Model under the course folder. When the administrator uploads course resources to the database through the background management system, users can download course resources by clicking the link of course resources.

```

class CourseResource(models.Model):
    course = models.ForeignKey(Course, verbose_name=u"Course", on_delete=models.CASCADE)
    name = models.CharField(max_length=100, verbose_name=u"name")
    download = models.FileField(upload_to="course/resource/%Y/%m", verbose_name=u"Resource file", max_length=100)
    add_time = models.DateTimeField(default=timezone.now, verbose_name=u"add time")

    class Meta:
        verbose_name = u"resource file"
        verbose_name_plural = verbose_name

```

Figure 6.25: Comments view

In the view function, we request the course resources stored in the database through get in the background management system, and get the required course resources through the model.objects.filter(), and display them in the HTML page of the course chapter. The user can download them by clicking.

```

def get(self, request, course_id):
    course = Course.objects.get(id=int(course_id))
    all_resources = CourseResource.objects.filter(course=course)

```

Figure 6.26: Course resource filter

6.3 Routes

In the route part, we allow users to click the system's route to complete the access to the front-end page. The following are the HTTP methods, paths, and usage of routes in this online education platform.

HTTP Method	Route	Usage
Get	register/	User registration
Get	active/(?P<active_code>)\.*)/	Activate the user's registration information
Post	login/	User login
Get	logout/	User logout
Get	info/	Viewing User Information
Post	image/upload/	Users modify their profile pictures
Get	mycourse/	User view courses
Get	myfav/org/	Users view favorite course organizations
Get	myfav/teacher/	Users view favorite lecturers
Get	myfav/course/	Users view favorite courses
Get	mymessage/	Users view personal messages

Get	/	Get the caroute graph
Get	list/	Get a list of course's organizations
Get	home/	Get organization home page details
Get	desc/	Get organisation details
Get	org_teacher/	Get a list of organization's faculty
Post	add_fav/	The user can favorite courses
Get	teacher/detail/(?P<teacher_id>\d+)/	Get teacher details
Get	list/	Get a list of open courses
Get	detail/(?P<course_id>\d+)/	Get course Details
Get	info/(?P<course_id>\d+)/	Get Course Information
Get	comment/(?P<course_id>\d+)/	Get user reviews
Post	add_comment/	Users post comments

Table 6.1: Routes and URL

6.4 Comparison to Existing Solutions:

6.4.1 The front page

Compared with the existing online education platform introduced in the second chapter of this dissertation, there are still many deficiencies in the functional design of the front-end page. Like on the home page, the navigation bar can redirect the page quickly. If time is enough, I will consider using different categories in the navigation bar like icourse.org. To join the quick access links, such as a bar to add a course, the user can in the home page according to oneself want to learn the course categories (such as computer class, Foreign language) for a quick visit. The advantage of this is that users do not need to click on the course details page to search, which increases the difficulty of development but improves the user experience to a certain extent.

In addition, the content of the rote chart displayed on the front page is not novel enough. The rote chart is the picture I searched on the Internet. Compared with the existing online education platform, it does not highlight the characteristics of the website designed by myself. If I have enough time, I will consider creating a picture with my logo as the content of the website's rotation chart. The benefits of doing this are to increase the personal publicity of the online education platform and generate more novel and beautifying ideas in the process of designing the website's logo.

6.4.2 Web functions

In terms of system functions, GMOOC refers to the existing online education platform, imitates the design of most existing functions, and adds the function of data download. However, unfortunately, some functions are still not realized.

Compared with existing online education platforms, GMOOC mainly refers to the following functions: user registration, user activation, login, and logout; In the personal center to provide users with personal information to view and modify; Change the avatar; View your favorite courses, favorite institutions, favorite teachers and users' messages. Educational institutions provide a list of courses to show different academic institutions and specific information; Users can choose whether to collect educational institutions according to their preferences. The system can provide users with detailed courses and chapter lists and support users to comment on courses and download course resources.

However, compared with existing online education platforms, GMOOC still lacks some functions. For example, there is no way to use third-party authentication (such as mailbox authentication) to ensure the security of users' registration. Users cannot change passwords. In the next chapter, unit Test, this dissertation will conduct the unit test on completing system functions against the function sheet.

7. Unit Testing

For testing GMOOC, I chose to perform unit testing to ensure the consistency and stability of the system across platforms. All test cases provide screenshots below the point as evidence.

7.1 Home Page

Test Case 7.1.1 – The navigation bar

Test Case #	7.1.1
Test Case Description	The navigation bar allows users to quickly view the homepage, open courses, teachers and organizations.
Test Steps	<ul style="list-style-type: none"> - Go to homepage - Click the Home page button. - Click the Open courses button. - Click the Teacher button. - Click the Organization button.
Pre-Requisites	The administrator has uploaded information about courses, teachers and organizations in the background management system.
Expected Results	Users can click a homepage, open courses, teacher and organization button in the navigation bar to access the correct contents.
Actual Results	Users can click a homepage, open courses, teacher and organization button in the navigation bar to access the correct contents.
Pass / Fail	PASS

Table 7.1: Unit test navigation bar

遇见好答案 [Android 开发问答专场]
程序猜想知道的, 不知道的, 应该知道的, 都在这

Professional authority The latest curriculum Famous Teacher Real Data

Home Page > Courses Detail

The Latest The most Popular participation

Course Name	Language	Learn Times	Course Students	From Cardiff University	Participation
CAERDYD	java	15	43	From Cardiff University	33
CAERDYD	Python	0	12	From Cardiff University	32
CARDIFF					
CARDIFF					

[Home Page](#) > [Organization Courses](#)

Types of

All

Training institutions

Colleges and universities

Personal

Area

All

Cardiff

Beijing

More

HowMany

[All](#) [Students Number ↓](#) [Courses Number ↓](#)

Cardiff University

Courses Number: 1 Students Number: 1000

Zhongguancun North Street, Haidian District, Beijing

The classic course: C language basics Database Basics

Contact Service



Peking University

Courses Number: 1 Students Number: 1000

Zhongguancun North Street, Haidian District, Beijing

The classic course: C language basics Database Basics

Contact Service



mooc1

Courses Number: 1 Students Number: 1000

Zhongguancun North Street, Haidian District, Beijing

The classic course: C language basics Database Basics

Contact Service

Home Page Open courses Teacher Organization

Home Page > Organization Courses

Types of	All	Training institutions	Colleges and universities	Personal
Area	All	Cardiff	Beijing	More

How Many

All Students Number ↓ Courses Number ↓



Cardiff University 

Courses Number: 1 Students Number: 1000

Zhongguancun North Street, Haidian District, Beijing

The classic course: C language basics Database Basics





Peking University 

Courses Number: 1 Students Number: 1000

Zhongguancun North Street, Haidian District, Beijing

The classic course: C language basics Database Basics





mooc1 

Courses Number: 1 Students Number: 1000

Zhongguancun North Street, Haidian District, Beijing

The classic course: C language basics Database Basics



Test Case 7.1.2 – Global Search

Test Case #	7.1.2
Test Case Description	Verify that users can search for specific courses through the global search on the home page.
Test Steps	<ul style="list-style-type: none"> - Go to homepage. - Click on the global search box. - Enter course name. - Click the “magnifying Glass” icon
Pre-Requisites	The administrator has uploaded the course information in the background management system.
Expected Results	Users can enter a specific course name to search for courses.

Actual Results	Users can enter a specific course name to search for courses.
Pass / Fail	PASS

Table 7.2: Unit test global search

The image consists of two screenshots of the GMDOC platform. The top screenshot shows a search bar with the text 'python' entered. The bottom screenshot shows the search results for 'python' courses, displaying one result for 'CAERDYD CARDIFF'.

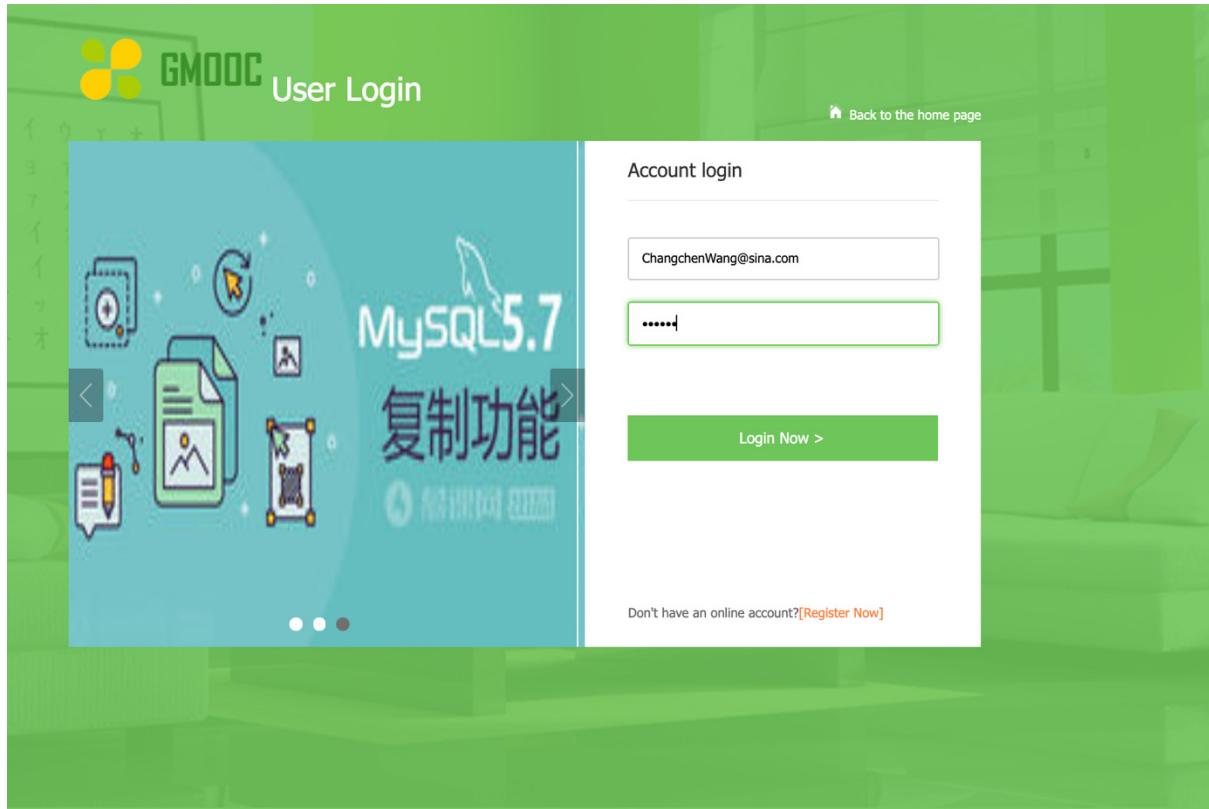
7.2 User Login

Test Case 7.2.1 – User Login

Test Case #	7.2.1
Test Case Description	Verify a user can log in with a correct email and password.
Test Steps	<ul style="list-style-type: none"> - Go to homepage - Click the login button - Enter the registered email address and password - Click 'Register and log in' button
Pre-Requisites	Users need to make sure they have a valid account.
Expected Results	The user logs in successfully and is

	automatically redirected to the home page.
Actual Results	The user logs in successfully and is automatically redirected to the home page.
Pass / Fail	PASS

Table 7.3: Unit test Login



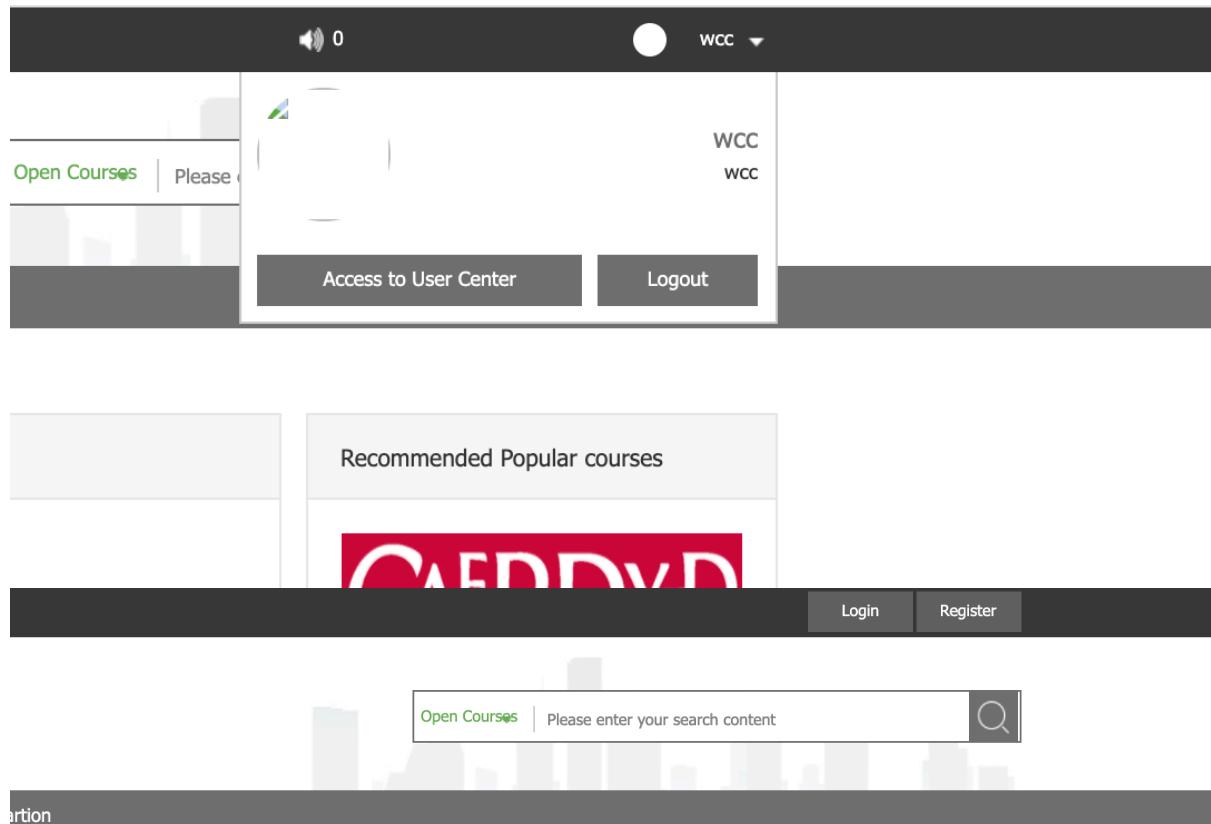
According to the user information in the upper right corner, ChangchenWang@sina.com is successfully logged in.



Test Case 7.2.2 – User Log out

Test Case #	7.2.2
Test Case Description	Verify that the user can log out
Test Steps	<ul style="list-style-type: none"> - Go to homepage - Click the log out button
Pre-Requisites	User logged in.
Expected Results	The user logs out successfully.
Actual Results	The user logs out successfully.
Pass / Fail	PASS

Table 7.4: Unit test Log out

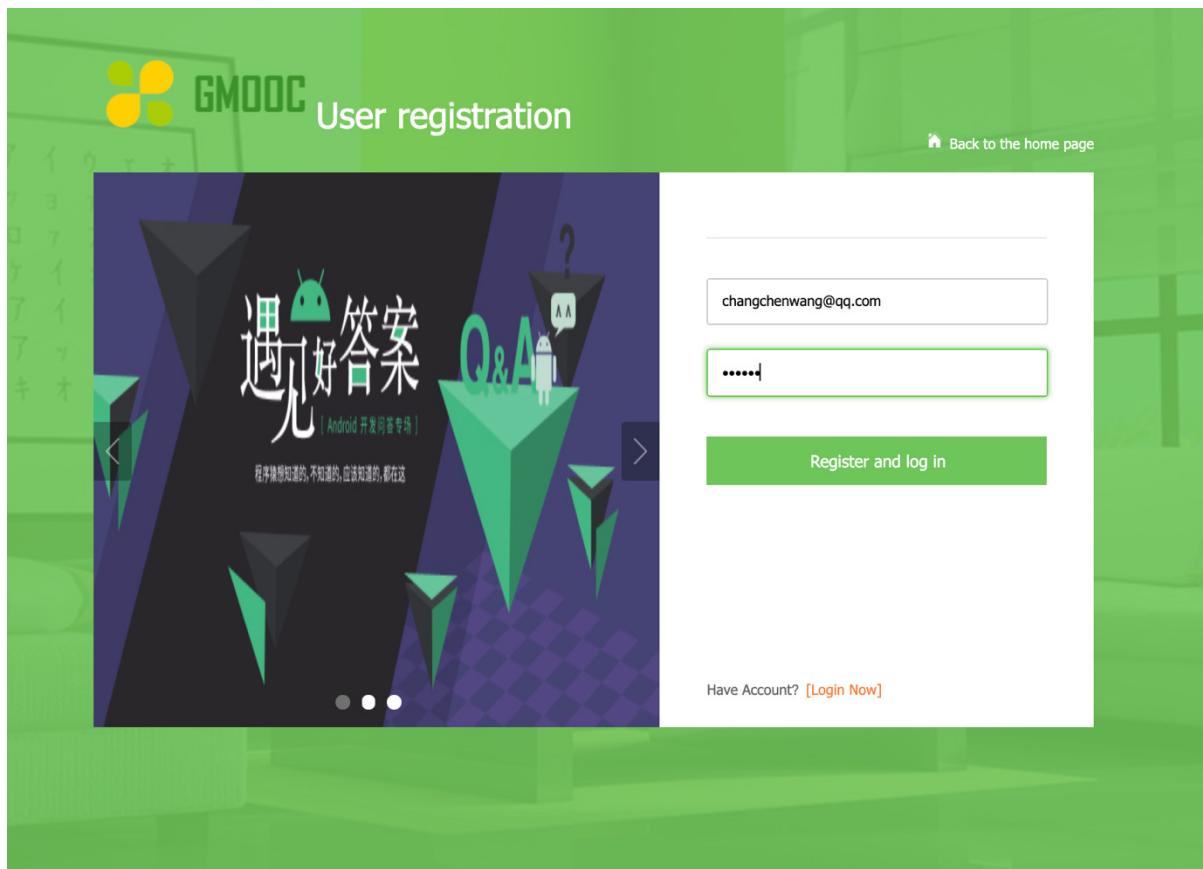


7.3 Registration:

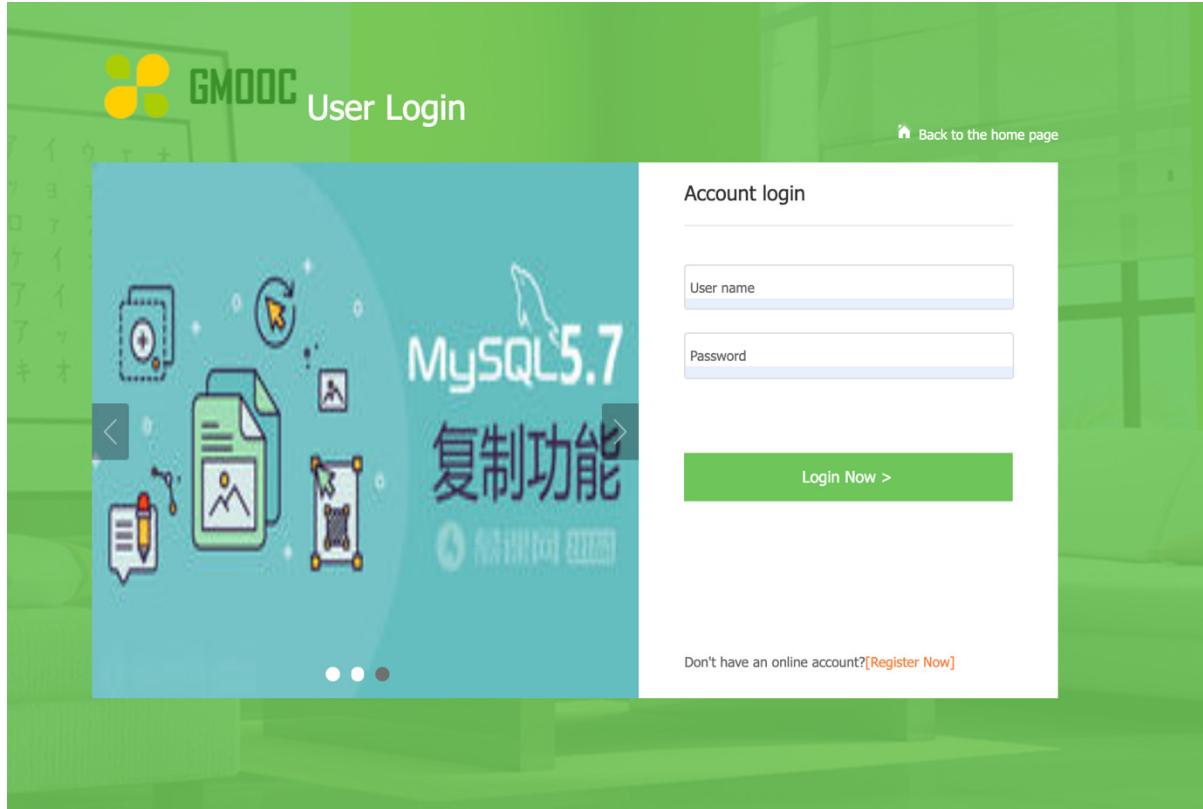
Test Case 7.3.1 – User Registration

Test Case #	7.3.1
Test Case Description	Verify that a user can register with the correct username and password
Test Steps	<ul style="list-style-type: none"> - Go to homepage - Click the register button - Enter correct username and password - Click 'Register and log in' button
Pre-Requisites	User must use a properly formatted mailbox
Expected Results	Users click 'Register and login' button to enter the user page
Actual Results	Users click 'Register and login' button to enter the user page
Pass / Fail	PASS

Table 7.5: Unit test User registration



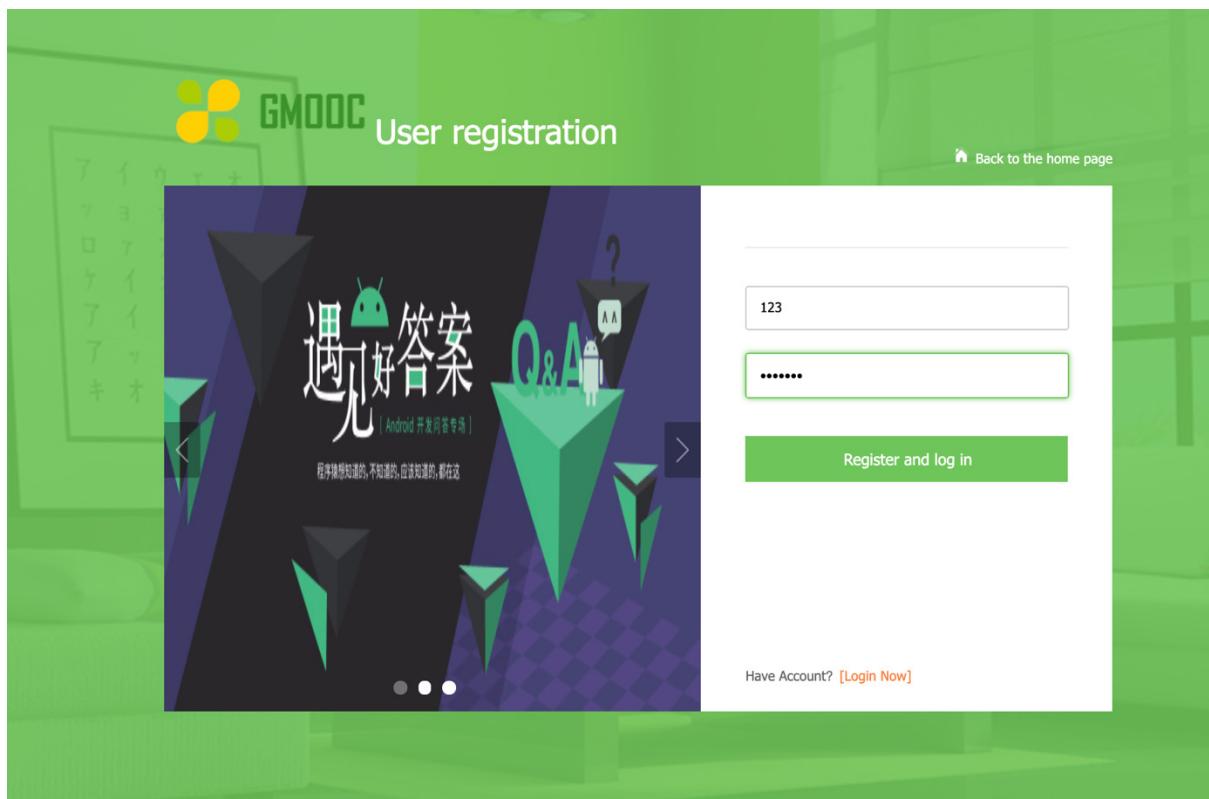
跳转成功

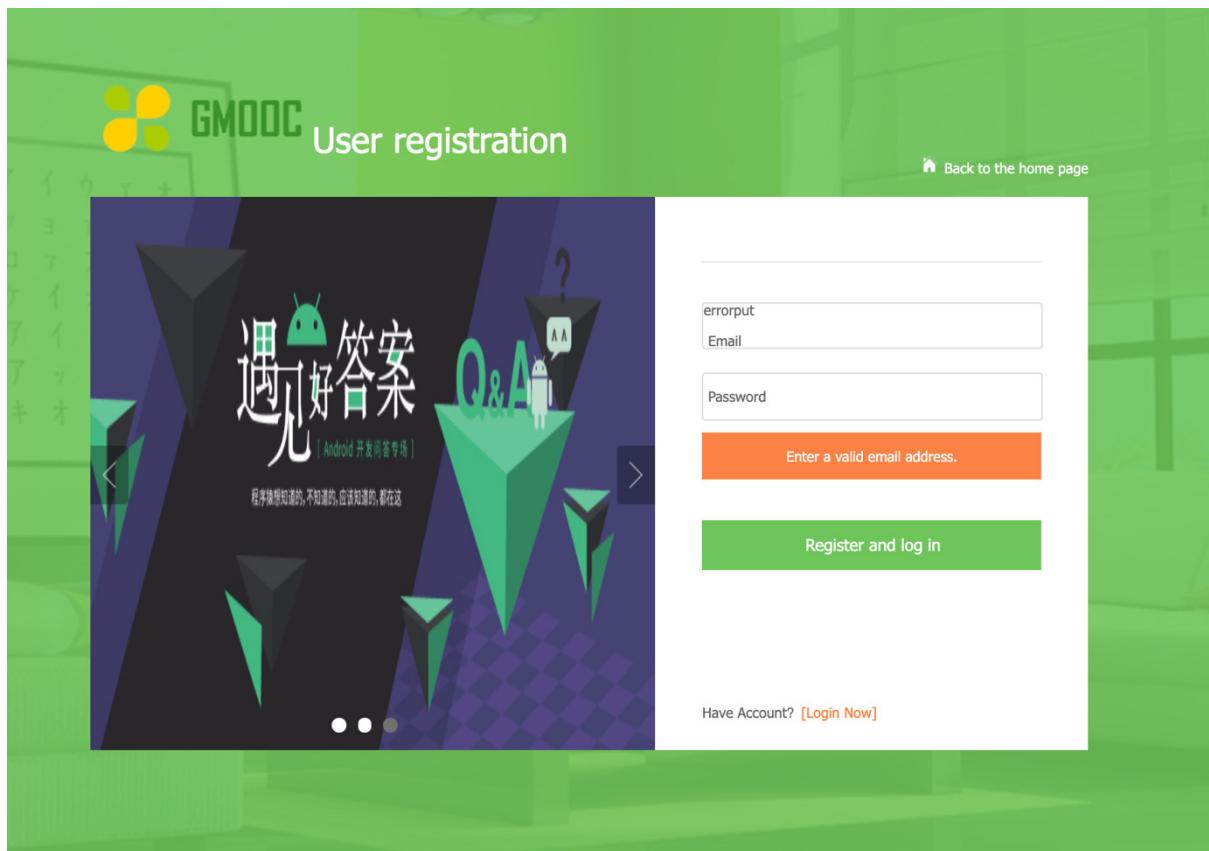


Test Case 7.3.2 – Incorrect Username Registrations

Test Case #	7.3.2
Test Case Description	Verify a user cannot register with an incorrect email.
Test Steps	<ul style="list-style-type: none"> - Go to homepage - Click the register button - Enter correct username and password - Click 'Register and log in' button
Pre-Requisites	The user registered with an incorrectly formatted mailbox
Expected Results	User is not register, and is shown an error message
Actual Results	User is not register, and is shown an error message
Pass / Fail	PASS

Table 7.6: Unit test User Incorrect Username Registration





7.4 Personal Center

Test Case 7.4.1 – Change personal information

Test Case #	7.4.1
Test Case Description	Verify a user can modify their personal information (nickname, profile picture, birthday, address, email).
Test Steps	<ul style="list-style-type: none"> - Go to homepage. - Click the Personal Center button. - Change information.
Pre-Requisites	The user has registered and logged in.
Expected Results	The user successfully modified the file
Actual Results	The user successfully modified the file
Pass / Fail	PASS

Table 7.7: Unit test User information change

Personal Information



Change the

Change the
password

Nickname:

Birthday:

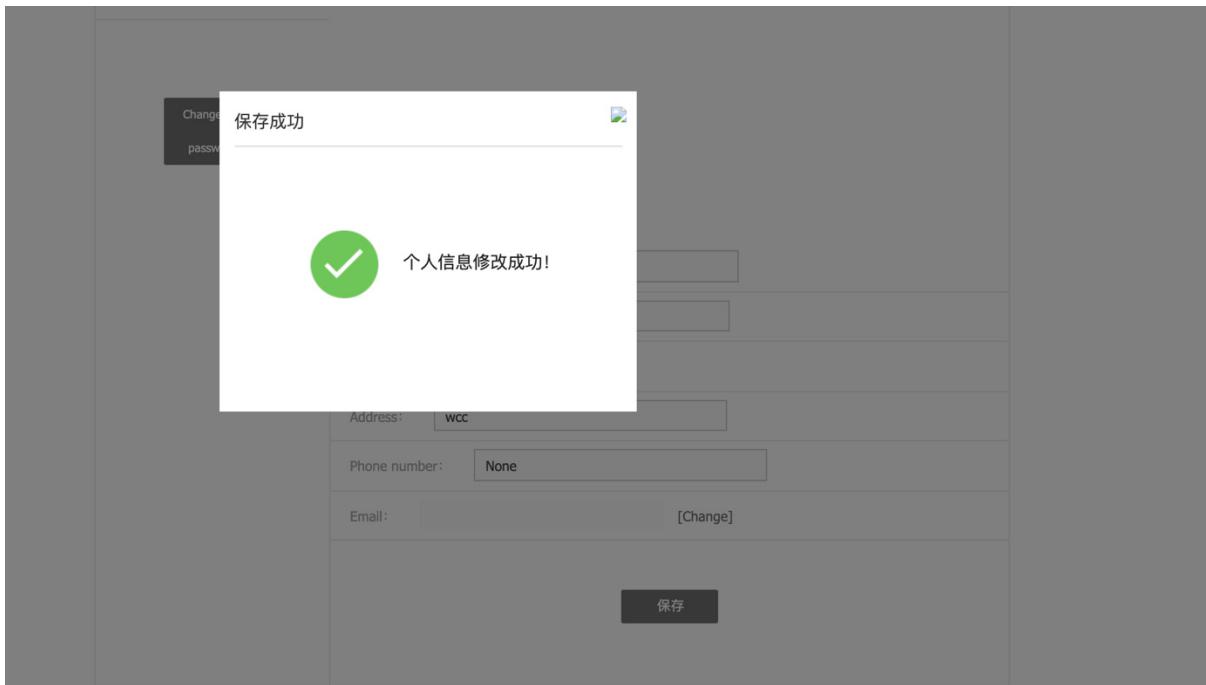
Gender: Male Female

Address:

Phone number:

Email: [Change]

Save



Test Case 7.4.2 – View the courses they are learning

Test Case #	7.4.2
Test Case Description	Verify that users can view the courses they have started.
Test Steps	<ul style="list-style-type: none">- Go to homepage.- Click the personal center button.- Click my courses button.
Pre-Requisites	The administrator uploaded the course information in the background, and the user clicked on the course details page to start learning
Expected Results	Users can view the lessons they are learning.
Actual Results	Users can view the lessons they are learning.
Pass / Fail	PASS

Table 7.8: Unit test view courses

The screenshot shows a navigation bar with 'Home Page > Personal Center > My Courses'. On the left, a sidebar lists 'Personal Information' (with a green circular icon), 'My Courses' (with a grey circular icon), 'My Collected' (with a grey circular icon), and 'My message' (with a grey circular icon). The main area is titled 'My Courses'. A course card for 'java' is displayed, featuring the 'CAERDYD CARDIFF' logo, the title 'java', a duration of '15', and a rating of '33'. Below the card, it says 'Cardiff University'.

Test Case 7.4.3 – View institutions that have been collected

Test Case #	7.4.3
Test Case Description	Verify that users can view organizations they already favorite.
Test Steps	<ul style="list-style-type: none"> - Go to homepage. - Click the User center button. - Click My collected button.
Pre-Requisites	The administrator has uploaded the course in the background, and users favorite the course.
Expected Results	Users can view organizations they already favorite.
Actual Results	Users can view organizations they already favorite.
Pass / Fail	PASS

Table 7.9: Unit test view organizations

Test Case 7.4.4 – View my messages

Test Case #	7.4.4
Test Case Description	Verify that users can view their own messages
Test Steps	<ul style="list-style-type: none"> - Go to homepage

	<ul style="list-style-type: none"> - Click the personal center button. - Click My message button.
Pre-Requisites	T The administrator has sent messages to users in the background management system.
Expected Results	Users can view their own messages.
Actual Results	Users can view their own messages.
Pass / Fail	PASS

Table 7.10: Unit test view own messages



My Information

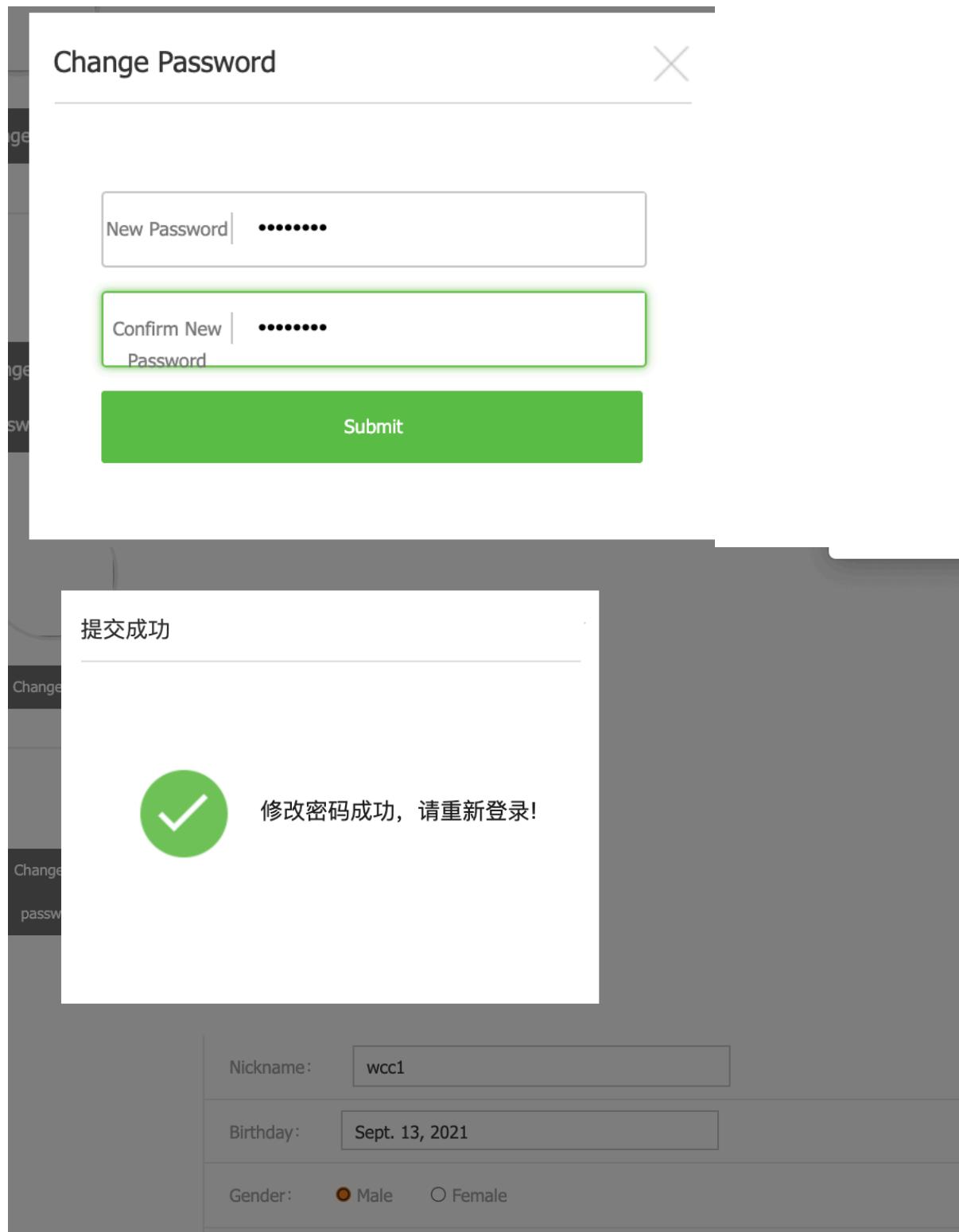
My message

Sept. 27, 2021, 10:32 a.m. HI

Test Case 7.4.5 – Change the password

Test Case #	7.4.5
Test Case Description	Authentication Users can change account passwords
Test Steps	<ul style="list-style-type: none"> - Go to homepage - Click the User center button - Click Change the password button.
Pre-Requisites	The administrator has uploaded the course resources in the background management system.
Expected Results	Users click the download link to download automatically.
Actual Results	Users click the download link to download automatically.
Pass / Fail	PASS

Table 7.11: Unit test change account password



"Password changed successfully, and please login again! "But I don't know why it is still shown here in Chinese; I am sure the content in the JS file is not Chinese.

```
Dml.fun.showTipsDialog( obj: {  
    title:'submit successfully',  
    h2:'Password changed successfully, please log in again!',  
});  
Dml.fun.winReload();  
else if(data.msa){
```

7.5 Open courses

Test Case 7.5.1 – Courses are sorted by upload time

Test Case #	7.5.1
Test Case Description	Verify that courses are sorted by upload time.
Test Steps	<ul style="list-style-type: none">- Go to homepage- Click the Open Courses button- Click ‘The Latest’ button
Pre-Requisites	The background management system has uploaded the open course content
Expected Results	Courses are sorted by the time they were uploaded, with the most recently uploaded courses at the top.
Actual Results	Courses are sorted by the time they were uploaded, with the most recently uploaded courses at the top.
Pass / Fail	PASS

Table 7.12: Unit test courses sorting 1

The Latest

The most Popular

participation

CAERDYD	CAERDYD
CARDIFF	CARDIFF
java6	java5
Learn Times: 0	Learn Times: 0
Course Students: 0	Course Students: 0
FromCardiff	FromCardiff
java4	java3
Learn Times: 0	Learn Times: 0
Course Students: 0	Course Students: 0
FromCardiff	FromCardiff
CAERDYD	

Test Case 7.5.2 – Courses are sorted by number of favorites

Test Case #	7.5.2
Test Case Description	Verify that courses are sorted by number of favorites.
Test Steps	<ul style="list-style-type: none"> - Go to homepage - Click the Open Courses button - Click ‘The most popular’ button
Pre-Requisites	The background management system has uploaded the open course content and there are already people who favorite some courses.
Expected Results	Courses are sorted according to the number of students who liked them, with the most liked courses at the top.
Actual Results	Courses are sorted according to the number

	of students who liked them, with the most liked courses at the top.
Pass / Fail	PASS

Table 7.13: Unit test courses sorting 2

The Python course has one student who likes it (star 1), so it's on top.

The screenshot shows a website interface for GMODC. At the top, there is a navigation bar with a logo, a search bar, and buttons for 'Open Courses' and 'Ple'. Below the navigation bar, there is a breadcrumb trail: 'Home Page > Courses Detail'. The main content area displays course listings. The courses are sorted by popularity, with 'The most Popular' tab selected. The courses listed are:

- CAERDYD** (Top Left): Python. Learn Times: 0, Course Students: 11. FromCardiff. 1 star.
- CAERDYD** (Top Right): Flask1. Learn Times: 0, Course Students: 0. FromCardiff. 0 stars.
- CARDIFF** (Bottom Left): django. Learn Times: 0, Course Students: 0. FromCardiff. 0 stars.
- CARDIFF** (Bottom Right): django and vue. Learn Times: 0, Course Students: 0. FromCardiff. 0 stars.

Test Case 7.5.3 – Course listings are paginated

Test Case #	7.5.3
Test Case Description	Verify that the course list page is pageable
Test Steps	<ul style="list-style-type: none"> - Go to homepage - Click the Open Courses button

	- Click 'The next page' button
Pre-Requisites	The background management system has uploaded more than 10 open course resources.(10 Open Courses / Page)
Expected Results	Users can click to go to the next page to view the content of the second page.
Actual Results	Users can click to go to the next page to view the content of the second page.
Pass / Fail	PASS

Table 7.13: Unit test paging

The Latest The most Popular participation

 <p>java6</p> <p>Learn Times: 0 Course Students: 0</p> <p>FromCardiff  0</p>	 <p>java5</p> <p>Learn Times: 0 Course Students: 0</p> <p>FromCardiff  0</p>
 <p>java4</p> <p>Learn Times: 0 Course Students: 0</p> <p>FromCardiff  0</p>	 <p>java3</p> <p>Learn Times: 0 Course Students: 0</p> <p>FromCardiff  0</p>
 <p>java2</p> <p>Learn Times: 0 Course Students: 0</p> <p>FromCardiff  0</p>	 <p>java</p> <p>Learn Times: 0 Course Students: 0</p> <p>FromCardiff  0</p>
 <p>Computational Thinking</p> <p>Learn Times: 0 Course Students: 0</p> <p>FromCardiff  0</p>	 <p>django and vue</p> <p>Learn Times: 0 Course Students: 0</p> <p>FromCardiff  0</p>
 <p>django</p> <p>Learn Times: 0 Course Students: 0</p> <p>FromCardiff  0</p>	 <p>Flask1</p> <p>Learn Times: 0 Course Students: 0</p> <p>FromCardiff  0</p>

Recommended Popular courses

 <p>Python</p> <p>Difficulty: primary</p>
 <p>Flask1</p> <p>Difficulty: primary</p>
 <p>django</p> <p>Difficulty: primary</p>

[1](#) [2](#) [The Next Page](#)

7.6 Course Detail pages

Test Case 7.6.1 – The user begin to learn

Test Case #	7.6.1
Test Case Description	Verify that the user can learn courses
Test Steps	<ul style="list-style-type: none"> - Go to homepage - Click the Open Courses button - Click Begin to learn button.
Pre-Requisites	The administrator has uploaded the course in the background system.
Expected Results	Users can begin to learn courses.
Actual Results	Users can begin to learn courses.
Pass / Fail	PASS

Table 7.14: Unit test learn courses

The screenshot shows a web interface for a course titled 'java'. The course details include: Difficulty: primary, Duration: 15Minutes, Students number: 40. The course has three chapters: chapter1, chapter2, and chapter3. A comment section is present with a message: 'Course announcement: Course documentation and study materials have been uploaded'. A 'Resource download' section lists 'Lecturer at the tip' (Dr. X, Cardiff). A 'You have to know' section lists 'java' and 'What did the teacher tell you you could learn?'. A sidebar on the right indicates that 'The students in this class also learned it'.

Test Case 7.6.2 – Favorite courses

Test Case #	7.6.2
Test Case Description	Verify that the user can favorite
Test Steps	<ul style="list-style-type: none"> - Go to homepage - Click the Open Courses button - Click Favorite button.
Pre-Requisites	The administrator has uploaded the course in the background system.
Expected Results	Users can favorite course.
Actual Results	Users can favorite course.
Pass / Fail	PASS

Table 7.15: Unit test favorite courses

Home Page Open courses Teacher Organizartion

Home Page>Open Courses>Courses Detail



java

java

Difficulty: **primary** The number of learning: 34

The Length: 15

Chapter Number: 0

Course Category: The backend development

...

Favorite **Begin to learn**

Course Detail

java

Home Page Open courses Teacher Organizartion

Home Page>Open Courses>Courses Detail



java

java

Difficulty: **primary** The number of learning: 34

The Length: 15

Chapter Number: 0

Course Category: The backend development

...

Already favorite **Begin to learn**

Course Detail

java

7.7 Courses chapter information page

Test Case 7.7.1 – Courses detail

Test Case #	7.7.1
Test Case Description	Verify that users can view course details.
Test Steps	<ul style="list-style-type: none"> - Go to homepage. - Click the Open Courses button.
Pre-Requisites	The administrator has uploaded the course information in the background management system and fill in course details e.g. number of chapters etc.
Expected Results	users can view course details.
Actual Results	users can view course details.
Pass / Fail	PASS

Table 7.16: Unit test courses detail

Home Page>Open Courses>Courses Detail

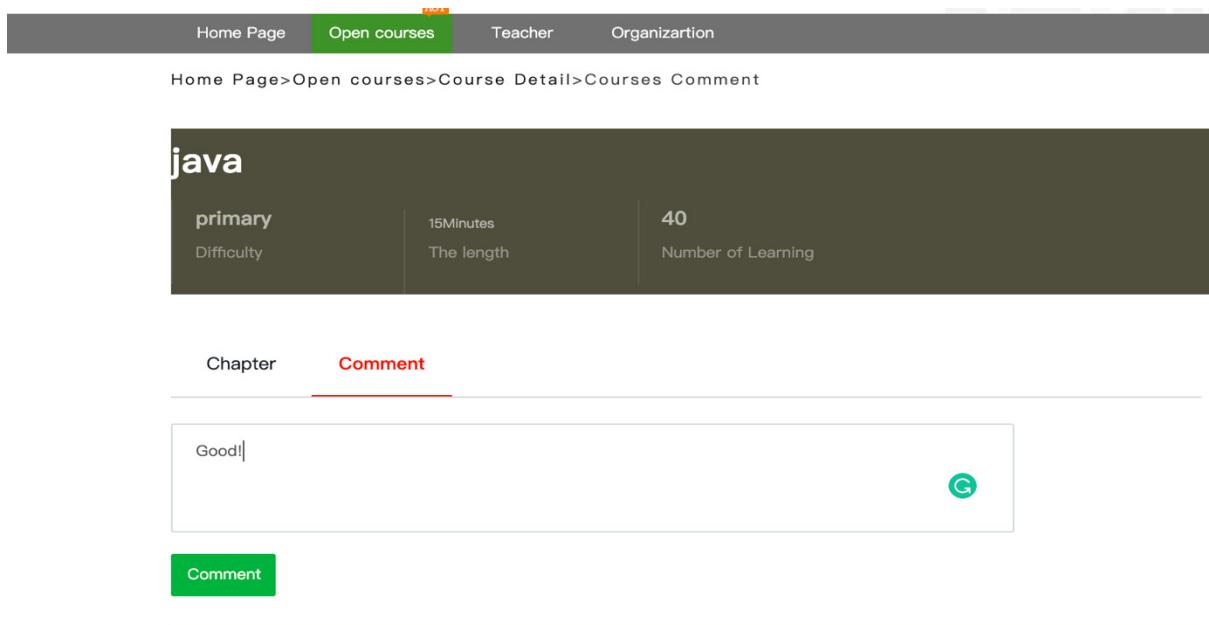


Test Case 7.7.2 – User comments

Test Case #	7.7.2
Test Case Description	Verify that users can post comments under the course section.
Test Steps	<ul style="list-style-type: none"> - Go to homepage - Click the Open Courses button - Click Begin to learn button. - Click Comment button. - Users enter comments they want to post. - Click Comment button.
Pre-Requisites	The administrator has uploaded the course

	information in the background management system and the user's comments are not empty.
Expected Results	Users can post personal comments under the course section
Actual Results	Users can post personal comments under the course section
Pass / Fail	PASS

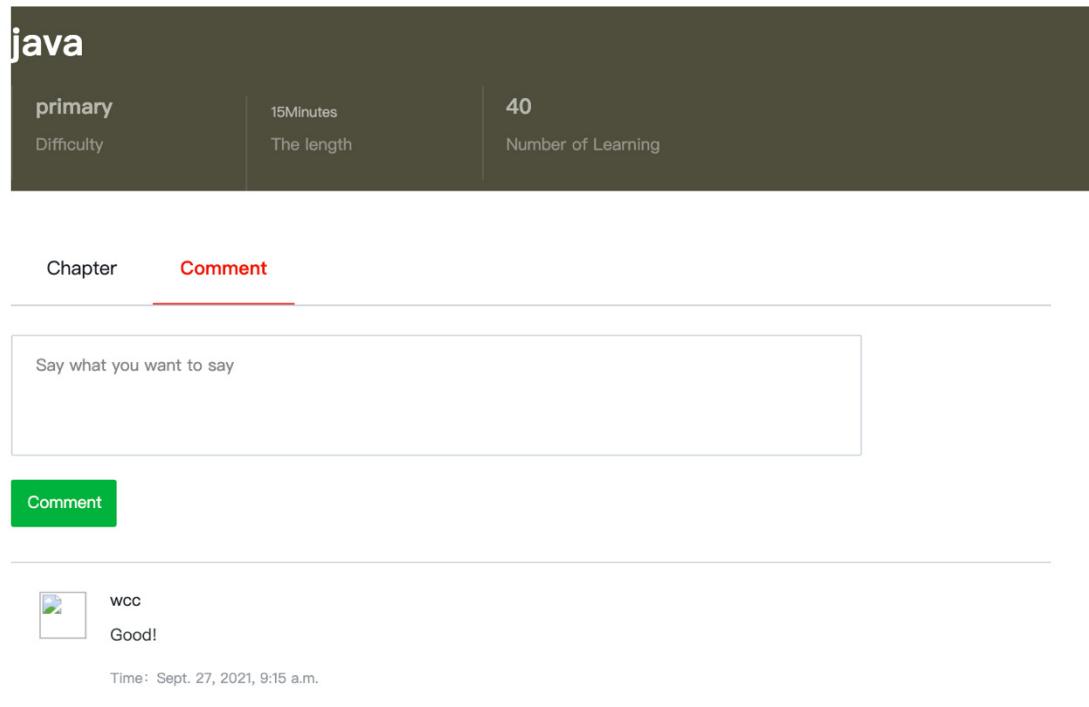
Table 7.17: Unit test user comments



The screenshot shows a course detail page for a course titled 'java'. The course details are as follows:

- Difficulty:** primary
- Length:** 15Minutes
- Number of Learning:** 40

The page includes a navigation bar with 'Home Page', 'Open courses' (highlighted in green), 'Teacher', and 'Organization'. Below the navigation, the URL is shown as 'Home Page>Open courses>Course Detail>Courses Comment'. The main content area has a header 'Comment' and a text input field containing 'Good!'. A green 'Comment' button is visible at the bottom left of the comment area.



The screenshot shows a course detail page for a Java course. The course title is "java". It has a "primary" difficulty level, a 15-minute duration, and 40 learners. Below the course details, there is a "Comment" section. The section has a header with "Chapter" and "Comment" (which is highlighted in red). A text input field contains the placeholder "Say what you want to say". A green "Comment" button is below the input field. A comment from a user named "wcc" is shown, with the text "Good!". The timestamp for this comment is "Time: Sept. 27, 2021, 9:15 a.m."

Test Case 7.7.3 – Resource download

Test Case #	7.7.3
Test Case Description	Verify that the user can download learning resources from the course chapter.
Test Steps	<ul style="list-style-type: none"> - Go to homepage - Click the Open Courses button - Click Begin to learn button. - Click on resources to download the learning resources link below.
Pre-Requisites	The administrator has uploaded the course resources in the background management system.
Expected Results	Users click the download link to download automatically.
Actual Results	Users click the download link to download automatically.
Pass / Fail	PASS

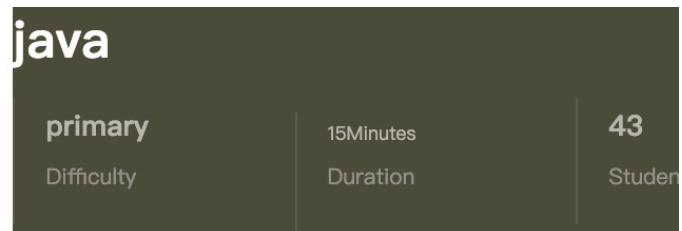
Table 7.18: Unit test download learning resources

Resource download

 java

[Download](#)

Lecturer at the tip



[Chapter](#) [Comment](#)

Course announcement: Course documentation and s
chapter1

chapter2

chapter3

127.0.0.1:8000/media/course/resouce/2021/09/截屏2021-09-27_17.15.15.png



截屏2021-09-27....png

^

7.8 Organization detail

Test Case 7.8.1 – Organization list are paginated

Test Case #	7.8.1
Test Case Description	Verify that the organization list page is pageable
Test Steps	<ul style="list-style-type: none">- Go to homepage- Click the Open Courses button- Click ‘Organization’ button
Pre-Requisites	The background management system has uploaded more than 5 organization's information. (5 organizations / Page)
Expected Results	Users can click to go to the next page to view the content of the second page.
Actual Results	Users can click to go to the next page to view the content of the second page.
Pass / Fail	PASS -

Table 7.18: Unit test paging

Types of All Training Institutions Colleges and universities Personal

Area All Cardiff Beijing

More

How11Many

All Students Number ↓ Courses Number ↓

mooc4
 Courses Number: 1 Students Number: 1000
 Zhongguancun North Street, Haidian District, Beijing
 The classic course: C language basics Database Basics

mooc5
 Courses Number: 1 Students Number: 1000
 Zhongguancun North Street, Haidian District, Beijing
 The classic course: C language basics Database Basics

mooc6
 Courses Number: 1 Students Number: 1000
 Zhongguancun North Street, Haidian District, Beijing
 The classic course: C language basics Database Basics

mooc7
 Courses Number: 1 Students Number: 1000
 Zhongguancun North Street, Haidian District, Beijing
 The classic course: C language basics Database Basics

mooc8
 Courses Number: 1 Students Number: 1000
 Zhongguancun North Street, Haidian District, Beijing
 The classic course: C language basics Database Basics

Contact Service

Contact Service

Contact Service

Contact Service

Contact Service

Contact Service

I want to learn

Name

Tel

Courses Name

Contact Now >

Ranking of Teaching Organizations

Cardiff University Cardiff University

Peking University Beijing

mooc1 beijing

The previous 1 2 3 The next page

Test Case 7.8.2 – Organizations are sorted by number of courses

Test Case #	7.8.2
Test Case Description	Verify that organizations are sorted by number of courses.
Test Steps	<ul style="list-style-type: none"> - Go to homepage - Click the Open Courses button - Click 'Courses Number' button
Pre-Requisites	A certain number of courses have been added to each educational organizations in the background management system.
Expected Results	Organizations are sorted according to the

	number of courses, with the largest number of courses is at the top.
Actual Results	Organizations are sorted according to the number of courses, with the largest number of courses is at the top.
Pass / Fail	PASS

Table 7.18: Unit test paging

The screenshot shows a list of organizations on the GMooc website. The organizations are sorted by student number, with Cardiff University at the top. Each organization entry includes its name, logo, course and student counts, address, classic course, and a 'Contact Service' button. A sidebar on the right shows a search bar and a 'I want to learn' form. The footer shows a page navigation bar with links to page 1, 2, 3, and 'The next page'.

Test Case 7.8.3 – Organizations are sorted by number of students

Test Case #	7.8.3
-------------	-------

Test Case Description	Verify that organizations are sorted by number of students.
Test Steps	<ul style="list-style-type: none"> - Go to homepage - Click the Open Courses button - Click 'Students Number' button
Pre-Requisites	The background management system has uploaded the information of the number of students in the course organization.
Expected Results	Organizations are sorted according to the number of students, with the largest number of students is at the top.
Actual Results	Organizations are sorted according to the number of students, with the largest number of students is at the top.
Pass / Fail	PASS

Table 7.18: Unit test organizations sorting

Home Page Open courses Teacher Organization

Home Page > Organization Courses

Types of All Training Institutions Colleges and universities Personal

Area All Cardiff Beijing More

How Many

All Students Number ↓ Courses Number ↓

Cardiff University  Courses Number: 1 Students Number: 1000
Zhongguancun North Street, Haidian District, Beijing
The classic course: C language basics Database Basics 

Peking University  Courses Number: 1 Students Number: 1000
Zhongguancun North Street, Haidian District, Beijing
The classic course: C language basics Database Basics 

mooc1  Courses Number: 1 Students Number: 1000
Zhongguancun North Street, Haidian District, Beijing
The classic course: C language basics Database Basics 

mooc2  Courses Number: 1 Students Number: 1000
Zhongguancun North Street, Haidian District, Beijing
The classic course: C language basics Database Basics 

mooc3  Courses Number: 1 Students Number: 1000
Zhongguancun North Street, Haidian District, Beijing
The classic course: C language basics Database Basics 

I want to learn

Name

Tel

Courses Name

Contact Now >

Ranking of Teaching Organizations

Cardiff University  Cardiff University

Peking University  Beijing

mooc1  beijing

1 2 3 The next page

Test Case 7.8.4 – View organization Details

Test Case #	7.8.4
Test Case Description	Verify that the user can view details of the course organization.
Test Steps	<ul style="list-style-type: none"> - Go to homepage - Click the Organization button - Click one organization cover page.
Pre-Requisites	In the background management system in a specific organization to add courses, teachers

	and other information.
Expected Results	Users can view all courses, teachers and introductions of an organization.
Actual Results	Users can view all courses, teachers and introductions of an organization.
Pass / Fail	PASS

Table 7.19: Unit test organizations sorting

The screenshot shows the GMOOC platform's organization page for 'CAERDYD CARDIFF'. The page header includes the logo, a 'GMOOC' badge, and a 'Collect' button. The main content area is titled 'All Courses' and lists two courses: 'Python' and 'java'. Each course card shows the title, duration (0 or 15), and the provider (Cardiff University). Below the courses is a 'Organization Teacher' section featuring 'java1' with a profile picture of a blue 'X'. The page also includes a 'Organization Introduction' section. Navigation links on the left include 'Organization Homepage', 'Organization Courses', 'Organization Introduction', and 'Organization Teachers'.

Test Case 7.8.5 – View courses information of organization

Test Case #	7.8.5
Test Case Description	Verify that the user can view all course information for an organization.
Test Steps	<ul style="list-style-type: none"> - Go to homepage - Click the Organization button - Click one organization cover page. - Click Organization courses button.
Pre-Requisites	The administrator has added courses information for the organization in the background management system.
Expected Results	Users can view all courses of an organization.

Actual Results	Users can view all courses of an organization.
Pass / Fail	PASS

Table 7.20: Unit test view courses information

Test Case 7.8.6 – View introduction of organization

Test Case #	7.8.6
Test Case Description	Verify that the user can view introduction of an organization.
Test Steps	<ul style="list-style-type: none"> - Go to homepage - Click the Organization button - Click one organization cover page. - Click Organization introduction button.
Pre-Requisites	The administrator has added introduction for the organization in the background management system.
Expected Results	Users can view introduction of an organization.
Actual Results	Users can view introduction of an organization.
Pass / Fail	PASS

Table 7.21: Unit test view courses information

HomePage>Organization>Introduction of the organization

Organization Homepage	Introduction of the organization
Organization Courses	Cardiff University
Organization Introduction	
Organization Teachers	

Test Case 7.8.7 – View teachers of organization

Test Case #	7.8.7
Test Case Description	Verify that the user can view teachers of an organization.
Test Steps	<ul style="list-style-type: none"> - Go to homepage - Click the Organization button - Click one organization cover page. - Click Organization teachers button.
Pre-Requisites	The administrator has added teacher information for the organization in the background management system.
Expected Results	Users can view introduction of an organization.
Actual Results	Users can view introduction of an organization.
Pass / Fail	PASS

Table 7.22: Unit test view teachers

HomePage>Organization>Organization Teacher

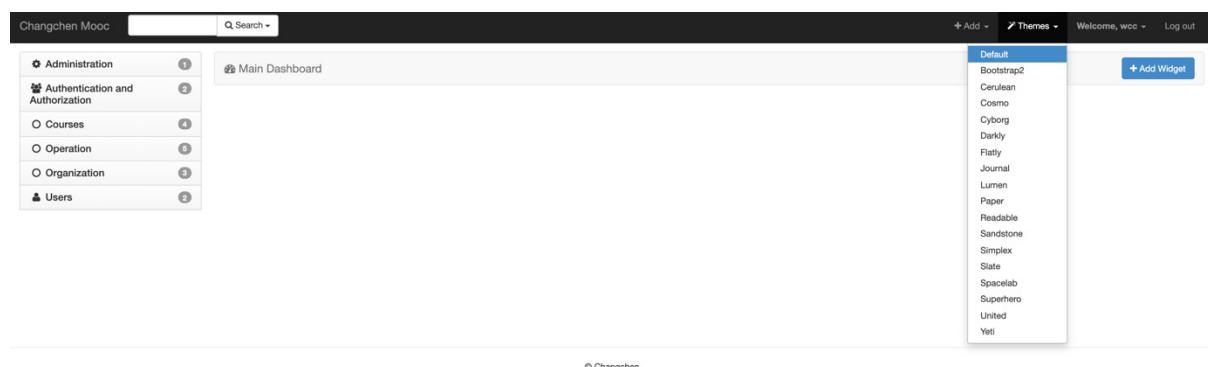
Organization Homepage	Organization Teacher
Organization Courses	Dr. X  certified
Organization Introduction	Work Years: 12年 Number of Courses: 3
Organization Teachers	

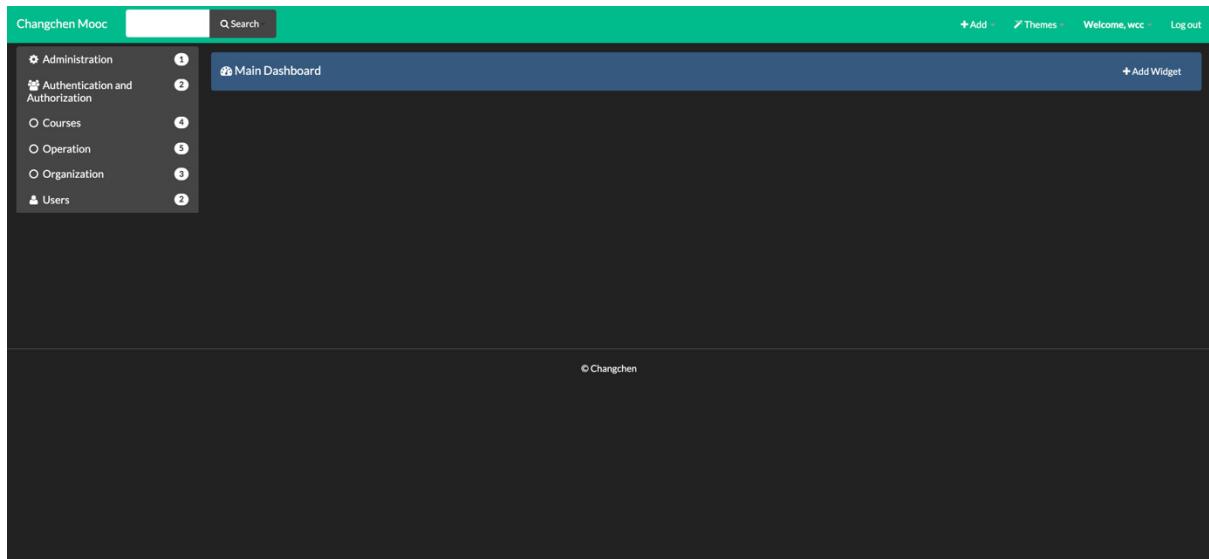
7.9 Background management system

Test Case 7.9.1 – Change the theme

Test Case #	7.9.1
Test Case Description	Verify that the administrator can modify the theme in the background administration system.
Test Steps	<ul style="list-style-type: none">- Go to xadmin.- Click the Themes button in the upper right corner.- The administrator selects the theme that you want to change.
Pre-Requisites	The administrator has been registered with the system.
Expected Results	The administrator can set the theme of the background management system.
Actual Results	The administrator can set the theme of the background management system.
Pass / Fail	PASS

Table 7.23: Unit test change the theme





Test Case 7.9.2 – View user's information

Test Case #	7.9.2
Test Case Description	Verify that an administrator can modify or view the identity of a user.
Test Steps	<ul style="list-style-type: none"> - Go to xadmin. - Click the Users button. - Click User information button.
Pre-Requisites	The administrator has been registered with the system.
Expected Results	The administrator can set the theme of the background management system.
Actual Results	The administrator can set the theme of the background management system.
Pass / Fail	PASS

Table 7.24: Unit test change view user's information

Test Case 7.9.3 – View user's operation

Test Case #	7.9.3
Test Case Description	Verify that the administrator can view users' favorite courses, users' messages, and users' comments.
Test Steps	<ul style="list-style-type: none"> - Go to xadmin. - Click the operation button.
Pre-Requisites	The administrator has been registered with the system.
Expected Results	Administrators can view users' favorite courses, users' messages, and users' comments.
Actual Results	Administrators can view users' favorite courses, users' messages, and users' comments.
Pass / Fail	PASS

Table 7.25: Unit test change view user's operation

Home / UserCourse

UserCourse Bookmark Filters Search

+ Add UserCourse

1 UserCourse

User	Course	Add_time
wcc	Course object (24)	Sept. 26, 2021, 2:23 p.m.

0 of 1 selected ▾

1 UserCourse

© Changchen

Home / UserMessage

UserMessage Bookmark Filters Search

+ Add UserMessage

1 UserMessage

user	message	has_read
1	Hi	✓

0 of 1 selected ▾

1 UserMessage

Home / CourseComments

CourseComments Bookmark Filters Search

+ Add CourseComments

1 CourseComments

user	course	comments	add_time
wcc	Course object (24)	Good!	Sept. 27, 2021, 9:15 a.m.

0 of 1 selected ▾

1 CourseComments

© Changchen

The screenshot shows a web application interface. On the left is a sidebar with a tree-like navigation menu. The menu items are: Administration (1), Authentication and Authorization (2), Courses (3), Operation (4), UserAsk, UserCourse, UserMessage, CourseComments, UserFavorite (selected, 5), Organization (3), and Users (2). The main content area has a header 'Home / UserFavorite' with a search bar 'Search UserFavorite' and a 'Filters' dropdown. Below this is a table titled 'UserFavorite' with 4 rows. The table has columns: user, fav_id, 收藏类型 (Collection Type), and add_time. The data is as follows:

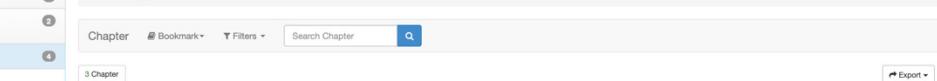
user	fav_id	收藏类型	add_time
wcc	24	course	Sept. 26, 2021, 2:19 p.m.
wcc	12	course	Sept. 24, 2021, 11:12 a.m.
wcc	10	course	Sept. 21, 2021, 7:09 a.m.
wcc	3	organization	Sept. 20, 2021, 6:24 a.m.

At the bottom left of the main content area is a green button with the text '0 of 4 selected'. At the bottom center is a footer '© Changchen'.

Test Case 7.9.4 – Upload the learning materials

Test Case #	7.9.4
Test Case Description	Verify that the administrator can upload learning materials.(Chapter learning materials, download resources.)
Test Steps	<ul style="list-style-type: none"> - Go to xadmin. - Click the courses button.
Pre-Requisites	The administrator has been registered with the system.
Expected Results	Administrator can upload learning materials.
Actual Results	Administrator can upload learning materials.
Pass / Fail	PASS

Table 7.25: Unit test change view learning materials



The screenshot shows a left sidebar with navigation options: Administration, Authentication and Authorization, Courses, Course, Chapter (selected), Video, Resource file, Operation, Organization, and Users. The main content area shows a header with 'Home / Chapter', a search bar, and a 'Add Chapter' button. Below is a table with three chapters, each with a checkbox, a name, and a timestamp. A green button at the bottom left indicates 3 chapters are selected. The bottom navigation bar shows '3 Chapter'.

	Course	Chapter	Add time
<input type="checkbox"/>	Course object (24)	chapter3	Sept. 26, 2021, 2:23 p.m.
<input type="checkbox"/>	Course object (24)	chapter2	Sept. 26, 2021, 2:23 p.m.
<input type="checkbox"/>	Course object (24)	chapter1	Sept. 26, 2021, 2:23 p.m.

3 Chapter

Home / Chapter

Chapter Bookmark Filters Search + Add Chapter

3 Chapter

Course	Chapter	Add time
Course object (24)	chapter3	Sept. 26, 2021, 2:23 p.m.
Course object (24)	chapter2	Sept. 26, 2021, 2:23 p.m.
Course object (24)	chapter1	Sept. 26, 2021, 2:23 p.m.

0 of 3 selected

3 Chapter

© Changchen

Home / resource file

resource file Bookmark Filters Search + Add resource file

2 resource file

Course	name	Resource file	add time
Course object (24)	java	course/resource/2021/09/截屏2021-09-27_17.15.15.png	Sept. 27, 2021, 9:20 a.m.
Course object (23)	java1	course/resource/2021/09/截屏2021-09-27_17.07.55.png	Sept. 27, 2021, 9:19 a.m.

0 of 2 selected

2 resource file

© Changchen

Changchen Mooc Search

Home / Chapter / Lesson object (10)

Change Lesson object (10)

Course* +

Chapter*

Add time*

Save Save and add another Save and continue editing Delete

© Changchen

Test Case 7.9.5 – Manage organization information

Test Case #	7.9.5
Test Case Description	Verify that the administrator can manage organization information, such as city, organization details, and teachers.
Test Steps	<ul style="list-style-type: none"> - Go to xadmin. - Click the organization button.
Pre-Requisites	The administrator has been registered with the system.
Expected Results	Administrator can upload learning materials.
Actual Results	Administrator can upload learning materials.
Pass / Fail	PASS

Table 7.26: Unit test manage organization information

© Changchen

© Changchen

© Changchen

Unit test results:

1.home page	Pass/Fail
1.1 The navigation bar allows you to quickly view the home page, open class, lecturers and teaching institutions	Pass
1.2 Global search, you can search specific courses	Pass
2.Sign in	
2.1 The user can log in	Pass
2.2 The user can log out	Pass

3.Registration:	
3.1 Registration form is shown when the user arrives at the registration page	Pass
3.2 If the user uses an incorrect mailbox, an error message is displayed.	Pass
4.Personal center	
4.1 Users can modify their personal information (nickname, profile picture, birthday, address, email, etc.)	Pass
4.2 Users can view the courses they are working on	Pass
4.3 Users can view the institution, the lecturer and the specific course, and can delete the contents	Pass
4.4 Users can reset their passwords	Pass
4.5 Users can view personal messages	Pass
5.Open Class Page	
5.1 Users can rank courses	Pass
5.2Users can view recommended popular courses	Pass
5.3 paging	Pass
6.Course Details Page	
6.1 Users can click to start the course and enter the chapter information of the course	Pass
6.2 Users can bookmark and cancel classes	Pass
6.3 Rich text shows the details of the course, explaining the main content of the course	Pass
7.Course chapter information page	
7.1 Users can view sections of the course	Pass
7.2 Users can comment on the course	Pass
7.3 Users can download the data	Pass
7.4 Users can view the instructor's tips for the course	Pass
7.5 Video chapter page for video playback	Fail
8.Lecture Lecturer page	
8.1 Users can rank instructors by popularity	Pass
8.2 paging	Pass
8.3 Users can view a list of instructors	Pass
9.Teaching Institution Page	
9.1 Users can filter the teaching institutions by category and region	Pass
9.2 Users can click to access the details page of the teaching institution	Pass

9.3 Users can rank courses by number of students and number of courses	Pass
9.4 Users can view the ranking of the institutions that taught them	Pass
10. Details page of the teaching institution	
10.1 Users can view the entire curriculum of a specific institution	Pass
10.2 You can view the introduction of the organization	Pass
10.3 Users can view all the instructors in the organization	Pass
11. Background management system	
11.1 The manager can set the theme of the background management system	Pass
11.2 User information: Administrators can manage user information, page rotation diagram	Pass
11.3 User operations: Administrators can manage and view users' courses, user messages, course comments and favorites.	Pass
11.4 Course management: administrators can add and delete courses, upload pictures and edit rich text on the new page of the course, and add chapters to the course; Filter courses according to fixed fields such as date range, course difficulty and course name; Export and import the course information in Excel, JSON and other formats	Pass
11.5 Authentication and authorization functions	Pass
11.6 Log recording function to record the modification operations of each background administrator	Pass

Table 7.27: Unit test results:

8. Reflection

This project has achieved the initial goal. As an online education platform with relatively complete functions and usability, GMOOC has met most of the requirements in the requirements document.

Implementation Conclusion

In general, I think the project implementation is successful. I want to introduce the overview of the whole project implementation process, challenges encountered, successful cases, and functions that I have not fully realized due to time reasons.

As an online education platform, I want to show an online education platform system that meets the basic functional requirements in this project. When users use the design for the first time, they can browse the basic page, such as the home page, course details page, etc. When users want to use more functions such as course comments, profile downloads, personal center, etc., they need to register an account and log in to use it. Regarding system interaction functions and login and registration, functions should also include introductory courses or educational institutions sorting, screening, global search, user course comments, user favorite courses, and other parts. In addition, I have added a data download function to this online education platform. This module is located on the course details page. The administrator can upload the course information in the background management system, and users can download it by clicking the link.

The web framework used in this project is Django, the background management system is Xadmin, and the database is MySQL. Having studied Flask during the semester, I am also interested in how the Django framework works. In developing an online education platform based on Django, I used Pycharm as the development software. Pycharm comes with Django framework and has py files such as Settings and URLs after installation, which is relatively convenient:

1. At the beginning of the project, I learned how to configure the database and static file paths in settings.py.
2. By confirming that the apps in this project are courses, Operation, organization, and Users to design the database model, I learned the syntax of defining the database model and how to use the manage.py file to migrate to the database synchronously.
3. I reviewed the knowledge of HTML, CSS, and JS by preparing front-end pages and template inheritance.
4. I learned how to write URLs and views functions in the Django framework by connecting and redirecting the front-end pages.

I encountered some challenges and unfinished features during the development of the project. First of all, in the course chapter, the video playing function can be imported from the Internet to my project page in the form of the video network link. However, due to the lack of

time, I did not find a suitable cloud platform to store the teaching videos. Second, because the Xadmin version is relatively backward, only Django 2.1 or the lower version can be used in the python3 environment, so there are a lot of bugs and errors that need debugging in the writing process of the project, most of which can be found on the Internet solutions. However, it took me a lot of time and energy, and bug modification and debugging became one of the unique challenges in this project. Finally, due to the version of Django used in reference books and on the Web, there will be some syntax incompatibility with Django2.1, such as the URL's syntax.

In general, in developing this online education platform, I have successfully added the function of downloading materials in the course chapter page according to my ideas in the system I designed. Therefore, I think this project has met my expectations. However, it is a pity that I did not complete the video function design. Teaching video is one of the most important factors of education method in the online education platform, and online education platform without teaching video is doomed to be functionally deficient. I will carefully consider how to supplement the functions of this part after the end of this project.

Personal Conclusion

I rated my performance as merit in this project. I learned how to use Django as a Web framework for system design. It was not easy for me to develop a system of this size in two months by myself. I did it and met all the functions of the requirements document. That's why I've given my performance merit.

However, I also have shortcomings, such as low efficiency and lack of preciseness in the project design process. I did not consider the version of the background management system Xadmin when I selected references and determined the development environment in the early stage of the project. As a result, when I prepared to integrate Xadmin with my project later, many function packages could not be used. It was September 15th at that time. I had finished the design of the front-end page and the function of URL and views and had a 10-day holiday. When I found that the whole project could not run, I suddenly realized the seriousness of the problem. At that time, I regretted that I should have built the background management system before choosing to rest for some time. I searched the Web for several solutions to Xadmin, and most of the advice on the Web was to downgrade Django. Then I went to System Settings and demoted Django from the latest version 3.6 to version 2.1. Since Django was demoted to 2.1, the model code and URLs written in Python3.8 have a lot of errors; Fortunately, they can find a solution on the Internet, and eventually, The system finally worked. I realized that I almost didn't finish the project because I wasn't rigorous enough and wasted my time. I feel terrified, so I will get rid of the bad habit of being thorough and inefficient in my future work and study.

Whole Project Conclusion

I am pleased to take this opportunity to learn to use Django to complete the development of an

online education platform. It is a significant thing, and I think the quality of my work is quite good. However, I did not complete the video playback function and did not deploy the system online due to the limited time. Although it has met most functional requirements, it is still not a deliverable system. I will consider continuing to improve this project in the future. As a graduate of Cardiff University, I will continue to work on this meaningful project at the moment of graduation.

Overall, I am happy with the development of the project and the resulting complete system with a wide range of features and a robust structure. I learned a lot from the work I did and increased my skills, which will be very useful when I start my career in computer science. I am very grateful to Dr. Natasha for her company as my supervisor. You helped me a lot and gave me a lot of help and advice on the structure and logic of the paper.

Thank you, Cardiff University, for the unforgettable memory of meeting you.

Appendices

Appendix 1

1.1

```
class Banner(models.Model):
    title = models.CharField(max_length=100, verbose_name=u"The title")
    image = models.ImageField(upload_to="banner/%Y/%m", verbose_name=u"image", max_length=100)
    url = models.URLField(max_length=200, verbose_name=u"address")
    index = models.IntegerField(default=100, verbose_name=u"index")
    add_time = models.DateTimeField(default=datetime.now, verbose_name=u"add time")

    class Meta:
        verbose_name = u"banner"
        verbose_name_plural = verbose_name
```

```

class UserProfile(AbstractUser):
    nick_name = models.CharField(max_length=50, verbose_name=u"nick name", default="")
    bday = models.DateField(verbose_name=u"birthday", null=True, blank=True)
    gender = models.CharField(max_length=6, choices=((("male",u"male"),("female",u"female"))), default="female")
    address = models.CharField(max_length=100, default=u"")
    mobile = models.CharField(max_length=11, null=True, blank=True)
    image = models.ImageField(upload_to="image/%Y/%m",default=u"image/default.png", max_length=100)

    class Meta:
        verbose_name = "User information"
        verbose_name_plural = verbose_name

    def __unicode__(self):
        return self.username

    def unread_nums(self):
        #Gets the number of unread messages from the user
        from operation.models import UserMessage
        return UserMessage.objects.filter(user=self.id, has_read=False).count()

class EmailVerifyRecord(models.Model):
    code = models.CharField(max_length=20, verbose_name=u"Verification code")
    email = models.EmailField(max_length=50, verbose_name=u"email")
    send_type = models.CharField(verbose_name=u"Verification code type",
                                  choices=((("register",u"registration"),
                                            ("forget",u"Retrieve password"),
                                            ("update_email",u"Modify email")),
                                            max_length=30))
    send_time = models.DateTimeField(verbose_name=u"Send time", default=datetime.now)

    class Meta:
        verbose_name = u"Email verification code"
        verbose_name_plural = verbose_name

    def __unicode__(self):
        return '{0}({1})'.format(self.code, self.email)

```

1.2

```

class Course(models.Model):
    course_org = models.ForeignKey(CourseOrg, verbose_name=u"组织机构", null=True, blank=True, on_delete=models.CASCADE)
    name = models.CharField(max_length=50, verbose_name=u"课程名称")
    desc = models.CharField(max_length=300, verbose_name=u"课程描述")
    detail = models.TextField(verbose_name=u"课程详情")
    teacher = models.ForeignKey(Teacher, verbose_name=u"讲师", null=True, blank=True, on_delete=models.CASCADE)
    is_banner = models.BooleanField(default=False, verbose_name=u"是否轮播")
    degree = models.CharField(choices=((("cj", "primary"), ("zj", "intermediate"), ("gj", "senior"))), max_length=2, verbose_name=u"课程难度")
    learn_times = models.IntegerField(default=0, verbose_name=u"学习时长(分钟)")
    students = models.IntegerField(default=0, verbose_name=u"学习人数")
    fav_nums = models.IntegerField(default=0, verbose_name=u"收藏人数")
    img = models.ImageField(upload_to="courses/%Y/%m", verbose_name=u"封面图", max_length=100)
    click_nums = models.IntegerField(default=0, verbose_name=u"点击量")
    category = models.CharField(default=u"后端开发", max_length=300, verbose_name=u"课程类别")
    tag = models.CharField(default="", verbose_name=u"课程标签", max_length=10)
    youneed_know = models.CharField(default="", max_length=300, verbose_name=u"课程须知")
    teacher_tell = models.CharField(default="", max_length=300, verbose_name=u"讲师介绍")
    add_time = models.DateTimeField(default=timezone.now, verbose_name=u"添加时间")

    class Meta:
        verbose_name = u"课程"
        verbose_name_plural = verbose_name

    def get_zj_nums(self):
        # 获取课程章节数
        return self.lesson_set.all().count()

    def get_learn_users(self):
        return self.usercourse_set.all()[:5]

    def get_course_lesson(self):
        # 获取课程所有章节
        return self.lesson_set.all()

    def __unicode__(self): # 错误的
        return self.name

```

```

class Lesson(models.Model):
    course = models.ForeignKey(Course, verbose_name=u"课程", on_delete=models.CASCADE)
    name = models.CharField(max_length=100, verbose_name=u"章节名称")
    add_time = models.DateTimeField(default=timezone.now, verbose_name=u"添加时间")

    class Meta:
        verbose_name = u"章节"
        verbose_name_plural = verbose_name

    def __unicode__(self):
        return self.name

    def get_lesson_video(self):
        # 获取章节视频
        return self.video_set.all()

class Video(models.Model):
    lesson = models.ForeignKey(Lesson, verbose_name=u"章节", on_delete=models.CASCADE)
    name = models.CharField(max_length=100, verbose_name=u"视频名称")
    learn_times = models.IntegerField(default=0, verbose_name=u"学习时长(分钟)")
    url = models.CharField(max_length=200, default="", verbose_name=u"视频地址")
    add_time = models.DateTimeField(default=timezone.now, verbose_name=u"添加时间")

    class Meta:
        verbose_name = u"视频"
        verbose_name_plural = verbose_name

    def __unicode__(self):
        return self.name

```

```

class CourseResource(models.Model):
    course = models.ForeignKey(Course, verbose_name=u"Course", on_delete=models.CASCADE)
    name = models.CharField(max_length=100, verbose_name=u"name")
    download = models.FileField(upload_to="course/resource/%Y/%m", verbose_name=u"Resource file", max_length=100)
    add_time = models.DateTimeField(default=timezone.now, verbose_name=u"add time")

    class Meta:
        verbose_name = u"resource file"
        verbose_name_plural = verbose_name

1.3

class UserAsk(models.Model):
    name = models.CharField(max_length=20, verbose_name=u"name")
    mobile = models.CharField(max_length=11, verbose_name=u"mobile")
    course_name = models.CharField(max_length=50, verbose_name=u"course_name")
    add_time = models.DateTimeField(default=datetime.now, verbose_name=u"add_time")

    class Meta:
        verbose_name = u"UserAsk"
        verbose_name_plural = verbose_name

class CourseComments(models.Model):
    #course comments
    user = models.ForeignKey(UserProfile, verbose_name=u"user", on_delete=models.CASCADE)
    course = models.ForeignKey(Course, verbose_name=u"course", on_delete=models.CASCADE)
    comments = models.CharField(max_length=200, verbose_name=u"comments")
    add_time = models.DateTimeField(default=datetime.now, verbose_name=u"add_time")

    class Meta:
        verbose_name = u"CourseComments"
        verbose_name_plural = verbose_name

class UserAsk(models.Model):
    name = models.CharField(max_length=20, verbose_name=u"name")
    mobile = models.CharField(max_length=11, verbose_name=u"mobile")
    course_name = models.CharField(max_length=50, verbose_name=u"course_name")
    add_time = models.DateTimeField(default=datetime.now, verbose_name=u"add_time")

    class Meta:
        verbose_name = u"UserAsk"
        verbose_name_plural = verbose_name

class CourseComments(models.Model):
    #course comments
    user = models.ForeignKey(UserProfile, verbose_name=u"user", on_delete=models.CASCADE)
    course = models.ForeignKey(Course, verbose_name=u"course", on_delete=models.CASCADE)
    comments = models.CharField(max_length=200, verbose_name=u"comments")
    add_time = models.DateTimeField(default=datetime.now, verbose_name=u"add_time")

    class Meta:
        verbose_name = u"CourseComments"
        verbose_name_plural = verbose_name

```

```
class UserFavorite(models.Model):  
    user = models.ForeignKey(UserProfile, verbose_name=u"User", on_delete=models.CASCADE)  
    fav_id = models.IntegerField(default=0, verbose_name=u"fav_id")  
    fav_type = models.IntegerField(choices=((1, "course"), (2, "organization"), (3, "lecturer")), default=1, verbose_name=u"收藏类型")  
    add_time = models.DateTimeField(default=datetime.now, verbose_name=u"add_time")  
  
    class Meta:  
        verbose_name = u"UserFavorite"  
        verbose_name_plural = verbose_name  
  
class UserMessage(models.Model):  
    user = models.IntegerField(default=0, verbose_name=u"User")  
    message = models.CharField(max_length=500, verbose_name=u"message")  
    has_read = models.BooleanField(default=False, verbose_name=u"has_read")  
    add_time = models.DateTimeField(default=datetime.now, verbose_name=u"add_time")  
  
    class Meta:  
        verbose_name = u"UserMessage"  
        verbose_name_plural = verbose_name  
  
class UserCourse(models.Model):  
    user = models.ForeignKey(UserProfile, verbose_name=u"User", on_delete=models.CASCADE)  
    course = models.ForeignKey(Course, verbose_name=u"Course", on_delete=models.CASCADE)  
    add_time = models.DateTimeField(default=datetime.now, verbose_name=u"Add_time")  
  
    class Meta:  
        verbose_name = u"UserCourse"  
        verbose_name_plural = verbose_name
```

1.4

```
class CityDic(models.Model):  
    name = models.CharField(max_length=20, verbose_name=u"name")  
    desc = models.TextField(max_length=200, verbose_name=u"desc")  
    add_time = models.DateTimeField(default=datetime.now)  
  
    class Meta:  
        verbose_name = u"city"  
        verbose_name_plural = verbose_name  
  
    def __unicode__(self):  
        return self.name
```

```

class CourseOrg(models.Model):
    name = models.CharField(max_length=50, verbose_name=u"CourseOrg")
    desc = models.TextField(verbose_name=u"desc")
    tag = models.CharField(default="tag", max_length=10, verbose_name=u"tag")
    category = models.CharField(default="pxjg", verbose_name=u"category", max_length=20,
                                 choices=((("pxjg", "organization"),
                                           ("gr", "personal"),
                                           ("gx", "University"))))
    click_nums = models.IntegerField(default=0, verbose_name=u"click_nums")
    fav_nums = models.IntegerField(default=0, verbose_name=u"fav_nums")
    img = models.ImageField(upload_to="org/%Y/%m", verbose_name=u"logo", max_length=100)
    address = models.CharField(max_length=150, verbose_name=u"address")
    city = models.ForeignKey(CityDic, verbose_name=u"city", on_delete=models.CASCADE)
    students = models.IntegerField(default=0, verbose_name=u"students")
    course_nums = models.IntegerField(default=0, verbose_name=u"course_nums")
    add_time = models.DateTimeField(default=datetime.now)

    class Meta:
        verbose_name = u"CourseOrg"
        verbose_name_plural = verbose_name

    def get_teacher_nums(self):
        #Get the number of teachers in the course institution
        return self.teacher_set.all().count()

    def __unicode__(self):
        return self.name

class Teacher(models.Model):
    org = models.ForeignKey(CourseOrg, verbose_name=u"org", on_delete=models.CASCADE)
    name = models.CharField(max_length=50, verbose_name=u"name")
    work_years = models.IntegerField(default=0, verbose_name=u"work_years")
    work_company = models.CharField(max_length=50, verbose_name=u"work_company")
    work_position = models.CharField(max_length=50, verbose_name=u"work_position")
    points = models.CharField(max_length=50, verbose_name=u"points")
    click_nums = models.IntegerField(default=0, verbose_name=u"click_nums")
    fav_nums = models.IntegerField(default=0, verbose_name=u"fav_nums")
    age = models.IntegerField(default=18, verbose_name=u"age")
    img = models.ImageField(default='', upload_to="courses/%Y/%m", verbose_name=u"img", max_length=100)
    add_time = models.DateTimeField(default=datetime.now)

    class Meta:
        verbose_name = u"teacher"
        verbose_name_plural = verbose_name

    def __unicode__(self):
        return self.name

```

References

- [1] Chen T, Peng L, Jing B, et al. 2020. The impact of the COVID-19 pandemic on user experience with online education platforms in China[J]. *Sustainability*. 12(18): 7329.
- [2] Hao, Z., Honghui, W., Haiduan, H., Jiangyan, Z. 2012. Development for distance education platform. In Advanced Technology in Teaching-Proceedings of the 2009 3rd International Conference on Teaching and Computational Science (WTCS 2009) (pp. 291-296). Springer, Berlin, Heidelberg.
- [3] CACHEIRO-GONZALEZ, M. L., Medina-Rivilla, A., Dominguez-Garrido, M. C., Medina-Dominguez, M. 2019. The learning platform in distance higher education: Student's perceptions. *Turkish Online Journal of Distance Education*, 20(1), 71-95.
- [4] Moore, M. G., & Kearsley, G. 2011. Distance education: A systems view of online learning. Cengage Learning.
- [5] Ke,M., 2016. Comparison of qualitative and quantitative methods. Available at: <https://wenku.baidu.com/view/16bb2bb7011ca300a7c3906d.html>. [Accessed: 6 October 2021].
- [6] Hourieh, A. 2008. Learning Website Development with Django. Packt Publishing Ltd.
- [7] Melé, A. 2020. Django 3 By Example: Build powerful and reliable Python web applications from scratch. Packt Publishing Ltd.
- [8] Forcier, J., Bissex, P., Chun, W. J. 2008. Python web development with Django. Addison-Wesley Professional.
- [9] Zhang, M., Dong, X. 2017. Research on the network security strategy for digital distance education platform. In MATEC Web of Conferences (Vol. 139, p. 00184). EDP Sciences.
- [10] Islam, Q. N. 2015. Mastering PyCharm. Packt Publishing Ltd.