## [详细设计文档参考](https://www.cnblogs.com/hellowzl/p/11511922.html)

|  |  |  |  |
| --- | --- | --- | --- |
| 文件名称 | UMOOC教学系统平台详细设计说明书 | 文件版本 | V1.0 |
| 文档编号 |  | 总页数 | 共 9 页 |

UMOOC教学系统平台详细设计说明书

|  |  |  |  |
| --- | --- | --- | --- |
| 拟  制 | 李映霞 | 日  期 | 2020.03.16 |
| 复  核 |  | 日  期 |  |
| 质  量 |  | 日  期 |  |
| 标准化 |  | 日  期 |  |
| 批  准 |  | 日  期 |  |

                                                                   xxx有限公司

                                                                  修订记录

|  |  |  |  |
| --- | --- | --- | --- |
| 日期 | 修订版本 | 描述 | 作者 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**目    录**

1       引言... 5

1.1             背景... 5

1.2             编写目的与范围... 6

1.3             术语定义... 6

1.4             参考资料... 8

1.5             相关工具... 8

2       设计概述... 8

2.1             任务和目标... 8

2.2             需求概述... 8

2.3             运行环境概述... 9

2.3.1         硬件环境... 9

2.3.2         支持环境... 11

2.4             限制和约束... 11

2.5             详细设计方法... 11

3       总体方案确认... 12

3.1             体系结构详细设计... 12

3.2             功能模块划分... 13

3.3             程序框架设计... 13

4       系统详细设计... 15

4.1             教学信息维护... 15

4.1.1         界面... 15

5       数据库设计... 40

6       安全性设计... 41

6.1             用户权限... 41

6.2             软件授权... 41

7       设计与开发规范... 43

# 1  引言

## 1.1   背景

xxx。

产品名称：UMOOC教学系统平台

所属系统：Windows

任务提出：xxx

任务承担：xxx

产品用户：xxx

应用场景：xxx。

## 1.2   编写目的与范围

本文档为“UMOOC教学系统详细设计说明书”，主要用于为实现系统功能而进行的系统详细设计说明，具体描述了系统包含的的软件模块的实现流程、功能、接口、数据结构等内容，供项目组开发人员和软件维护人员阅读。

本文档用于软件设计阶段的详细设计阶段，其范围是：系统各组成部门的构成；各软件模块（或类）的属性、方法、事件、外部接口；用户界面设计；数据结构设计等内容。

## 1.3   术语定义

**分层：**在分解复杂的软件系统时，软件设计者用得最多的技术之一就是分层。在计算机本身的架构中，可以看到：到处都有分层的例子。不同的层从包含了操作系统调用的程序设计语言，到设备驱动程序和CPU指令集，再到芯片内部的各种逻辑门。网络互联众，FTP层架构再TCP之上，TCP架构再IP之上，IP又架构再以太网之上。

当用分层的观点来考虑系统时，可以将各个子系统想象成按照“多层蛋糕”的形式来组织，每一层都依托在其下层之上。在这种组织方式下，上层使用了下层定义的各种服务，而下层对上层一无所知。另外，每一层对自己的上层隐藏其下层的细节。因此，第4层使用第3层的服务，第3层使用第2层的服务，第4层无需知道第2层的细节。（当然，并非所有的分层架构都这么隔绝，但绝大多数是不透明的，或至少是几乎不透明的。）

[**WPF**](https://docs.microsoft.com/en-us/visualstudio/designers/introduction-to-wpf?view=vs-2019)**：**Windows Presentation Foundation（WPF）是Microsoft的一个图形子系统（类似于WinForms），用于在基于Windows的应用程序中呈现用户界面。 WPF，以前称为“Avalon”，最初于2006年作为.NET Framework 3.0的一部分发布.WPF使用DirectX，并尝试为构建应用程序提供一致的编程模型。

[**MVVM**](https://docs.microsoft.com/en-us/xamarin/xamarin-forms/enterprise-application-patterns/mvvm)**：**Model-view-viewmodel是一种软件架构模式，它有助于将应用程序的业务和表示逻辑与其用户界面（UI）完全分离。 在应用程序逻辑和UI之间保持清晰的分离有助于解决许多开发问题，并使应用程序更易于测试，维护和发展。 它还可以极大地改善代码重用机会，并允许开发人员和UI设计人员在开发应用程序的各个部分时更轻松地进行协作。

MVVM模式中有三个核心组件：模型，视图和视图模型。 每个都有不同的目的。 下图显示了三个组件之间的关系。

**EF 6：**Entity Framework 6是一个经过实践检验的.NET对象关系映射器（O / RM），具有经过多年开发的功能和稳定性。

作为ORM，EF6减少了关系和面向对象世界之间的阻抗不匹配，使开发人员能够开发这样的程序：使用.NET应用程序域中的强类型对象，与存储在关系数据库中的数据进行交互，这样就不再需要编写大量的数据访问“管道”代码。

## 1.4   参考资料

《UMOOC教学系统需求规格设计说明书》

## 1.5   相关工具

设计时采用Power Designer、Visio等工具。开发时采用Visual Studio 2019、Sql Sever Management Studio 2012等工具。

# 2  设计概述

## 2.1   任务和目标

**一.详细设计的任务**

 1.为每个模块确定采用的算法

2.确定每一模块使用的数据结构

3.确定模块接口的细节

4.要为灭一个模块设计出一组测试用例

**二.详细设计的方法**

在过程设计阶段，要决定各个模块的实现算法，并精确地表达这些算法。表达过程规格说明的工具叫做详细设计工具。它可以分为以下三类：

图形工具（程序流程图，N-S图，问题分析图，PAD）

表格工具 （判定表）判定路径是否正确

语言工具（program design language，PDL）

## 2.2   需求概述

 目的:

随着互联网技术的发展，生活方式逐渐信息化,网络化,逐渐涌现了各种网上教学系统，网上教学近几年逐渐兴盛起来， 是未来教育的一种发展趋势， 网上教学能为教师的“教”和学生的“学“提供更广阔的灵活空间， 能打破地域、资源和时间的限制，让学生真正为自己的学习做主 。本书目的主要是给客户管理者、客户核心操作员、程序员、实施人员与项目负责人使用。以此说明书为软件开发蓝本。需要版本控制。每次的改动转发给每一位相关工作人员。以保持项目内容思想的统一。

文档:<UMOOC需求规格说明书>

软件维护的参考资料:

A、龚沛曾：《Visual Basic 程序设计教程》，高等教育出版社第三版；

B、《数据库系统概论》高等教育出版社第四版；

C、《软件工程》

D、网站编程开发基础；

E、项目计划任务书、

F、国家标准《详细设计说明书》 ；

G、项目开发计划：

UMOOC教学系统开发小组： 《教育类网站总结》、《UMOOC教学网站可行性研究报告

H、需求规格说明书：《网站规划与设计三之需求规格说明书》；

I 、概要设计说明书：《网站规划与设计四之概要设计说明书》；

J、测试计划（初稿） ：使用黑盒测试和白盒测试来检验每个功能是否能正

常使用，检验产品内部动作是否按照需求规格说明书的规定正常进行；

K、用户操作手册（初稿）:《大学门户网站系统使用说明书》；

范围: 网上教学系统采用传统的 b/s 结构，教师，学生和管理员可以在浏览器上远程操作,本文档是项目的软件需求规格说明书，是技术文档。 本文档使用对象为： 项目需求人员,项目经理,高层经理,软件工程组,软件相关组成员用户, 本文档为以后的实际代码编写提供依据，为程序员的分工提供参照等。,未经项目负责人书面许可，该文档不得提供给上述规定对象以外的人员阅读或使用。

## 2.3   运行环境概述

本系统的运行环境的规定如下：

### 2.3.1       硬件环境

本系统的硬件环境包括：

a)  服务器

服务器硬件推荐选用宝德至强双路云服务器PR2510SW，其主要技术参数如下表所示：

表1 服务器主要技术参数

|  |  |
| --- | --- |
| **指标名称** | **指标参数** |
| 处理器型号 | Intel® Xeon® E5-2600V3/V4系列处理器 |
| 处理器数量 | 1/2个 |
| 尺寸 | 2U机架 |
| 内存 | 8 DIMM插槽, 支持2400MHz DDR4的RDIMM内存 ，最大支持512GB |
| 硬盘 | 支持8个3.5”/2.5” SATA/SAS/SSD硬盘，可内置两个2.5” SATA/SAS/SSD硬盘，最大支持存储容量68TB |
| 板载网络 | 集成2个Intel i210千兆GE网口；可选配外接千兆及万兆网卡 |
| PCIE扩展 | 提供6个PCI-E 扩展插槽（3 PCI-E3.0×8，1 PCI-E3.0×4（in×8），1 PCI-E 3.0 ×16， 1 PCI-E 2.0×4(in ×8)） |
| 风扇 | 3个热插拔风扇 |

### 2.3.2       支持环境

本系统软件运行环境包括：

a)  xxx系统软件所需的软件支持环境包括：

1)   数据库： SQL Server 2008 R2；

2)   操作系统：Windows Server 2012 Enterprise（如需正版软件，由用户方提供）；

3)   Web Server：IIS 7.5

4)   .Net Framework 4.5.2

## 2.4   限制和约束

 必须遵循的标准:

（1）所有的 html 页面代码将遵循 html4.0 标准；

（2）系统的编码、设计和维护遵循本开发小组发布的标准；

（3）页面显示标准遵循本开发小组编写的 css文件；

（4）页面使用 javascript 设置页面脚本

输入项不能为空，否则不能进行查询；

后台处理服务器的性能会影响查询速度和网页响应速度

其他条件与限制:经费限制，开发期限，设备条件，用户现场环境准备、

## 2.5   详细设计方法

本文采用结构化设计方法，结合数据流图进行整体功能模块的划分。针对各个具体模块的设计采用面向对象设计方法，结合相应的UML模型进行功能设计，具体包括以包图表示的软件体系结构图，以交互图表示的用例实现图，完整精美的类图、针对复杂对象的类图，用以描述流程处理过程的活动图。

# 3  总体方案确认

## 3.1   体系结构详细设计

 3.1 需求概述

自从有了互联网， 信息量及信息的传递速度大大增加。 教育，作为国家基础建设的重中之重， 各高等教育院校利用了互联网诸多优点， 建立了许多教育类网站，方便了信息的上传与下达。UMOOC教学系统开发小组根据第三方提出的开发先生为教育系统的需求，基于使用成熟技术和实用化的原则， 通过对软件,网站系统的总体结构设计、 系统运行环境设计、 处理流程设计和系统功能设计、 编码测试和维护等， 设计出一款基于网站的教育系统， 此系统是向最终用户提供方便强大的操作平台， 包括信息资讯模块和信息管理模块， 使得教育机构可以通过互联网向更多的网民宣传高校办学理念、 学校规模、 培养目标等信息； 学生通过该系统可以加强管理教育信息和优化使用，增强学生与老师的交流学习.

## 3.2   功能模块划分

 UMOOC教学系统分为： 4 个子系统：资源中心子系统、 我的课程子系统、 随堂教学子系统,个人中心子系统；

我的课程子系统模块分为：基本信息子模块、 单元学习子模块、课程活动子模块、 课程资源子模块、 学习进度子模块、课程通知子模块,答疑谈论子模块,在线测试子模块,课程作业子模块,学习统计等

个人中心子系统模块分为：下载管理子模块、 系统设置子模块、通知公告子模块、学习任务子模块

随堂教学子系统模块分为：学生签到模块

子模块。

## 3.3   程序框架设计

本系统采用分层架构模式将整个系统分为业务展示层（提供服务、显示信息）、系统服务层（业务逻辑，系统中真正的核心）、数据采集层（与数据库、消息系统及其他软件包通信）。

表现层采用MVVM（Model-View-View-Model）设计模式，结合微软 WPF 框架，可以把界面设计（用xaml语言实现）和业务逻辑开发（用C#实现）分离。

领域层设计符合面向接口编程规范，这样可以降低层与层之间的耦合，也有利于接口设计和实现的分离，但针对基本的单表的增删改查服务则是直接调用数据源层的EF API来实现，从而减少过度的接口封装。

数据源层运用ORM技术来简化对数据表结构的维护，具体运用了微软 .Net框架下的的Entity Framework 框架。

# 4  系统详细设计

## 4.1   课程信息管理

### 4.1.1   菜单

基础数据 >> 课程信息

### 4.1.2   界面

### 4.1.3   界面描述

|  |  |
| --- | --- |
| 页面元素 | 元素说明 |
| 查询条件 |  |
| 车牌号 | 文本录入，忽略大小写模糊匹配 |
| 列表项 |  |
| 车牌号 | PlateNumber |
| 操作 |  |
| 页面加载 | 默认不查询数据 |
| 查询 | 查询表“xxx”  固定条件：IsDelete != true  查询条件：根据条件录入，对应字段  排序：ID 倒序 |

### 4.1.4   新增/修改界面

### 4.1.5   功能详细说明

### 4.1 功能

### 课程信息检索

### 此功能主要是方便在校学生浏览网页， 查阅相关课程信息。 数据库会根据用户的检索单词迅速查找网站一切关于检索单词的相关内容， 使学生的查找更加方便快捷。

### ☆老师通知

### 学生可以通过软件.网站系统指定连接了解老师的通知, 考试通知、作业都会第一时间在网页上更新。

### ☆用户交互信息

### 交互信息的含义是指用户与网站制作方可以进行信息相互交流， 学生可以将对学校提一些自己的意见与建议，做好的作业上传提交,发现好的学习资料也可以进行上传， 使更多的学生获益。交互信息的概念使学生的上传与下载变的更加方便， 信息资源更加广泛。

### ☆课程管理功能

### 课程系统具体包括基础信息管理、作业管理、资源管理和系统管理等功能。

### a、基础信息管理

### 主要由课程介绍管理、教学大纲管理、教学日志管理,教师信息等个子模块组成。

### 各个模块功能如下。

### 课程介绍管理要完成学习目标,授课对象,课程类型,教材,教学方式基本信息的输入、保存、修改、删除、查询功能。

### 教学大纲管理对课程学习大纲的说明

### ③教学日志管理主要完成课程的时间安排

### b、作业管理

### 主要由作业发布,提交,批改等子模块组成。各个模块功能如下。

### 发布管理主要完成对学生作业的发布的功能。

### 提交主要完成作业提交功能。

### 批改主要完成对作业查看批改打分功能。

### c、单元学习管理

### 主要由学习项目,学习视频等子模块组成。各个模块功能如下。

### 学习项目管理主要完成对学习的安排的功能。

### d、系统管理

### 主要由操作员设置、 密码设置、 操作员级别设置、 权限管理 4 个子模块组成。各个模块功能如下。

### ①操作员设置主要完成操作员的管理工作。

### ②密码设置主要完成操作员密码的设置工作。

### ③操作员级别设置主要完成操作员级别的设置工作。

### ④权限管理主要完成操作员权限的设置工作。

### ☆课程活动管理功能

### 人事管理信息系统主要包括课程作业管理、答疑谈论、播放单元、在线测试等主要功能。

### a、课程作业管理

### 课程作业管理用于查看发布的作业

### b、答疑谈论管理

### 答疑谈论实现了对老师与学生互动的功能,老师发布课程问题,学生进行谈论回答等。

### c、在线测试

### 对各课程进行在线测试。

### d、数据库管理

### 包括数据库的备份和恢复功能。

### e、系统管理

### 登录系统时需要验证身份，只要合法的用户才可以进入管理系统，不同的用户使用不同的功能，同时也可以对密码进行修改，其主要包括注册和登录模块、用户和角色管理模块等。

### ☆老师教务信息管理功能

### 老师教务信息管理系统包括课程发布、学生管理、教师管理、 教案管理等子系统。其中，“学生管理”和“教师管理”又分别包含基本资料 、个人档案、考勤录入和考勤管理四个模块， “学生管理”和“教师管理”

### 功能相似

### A课程发布

### 课程发布子系统实现对课程、 通知的管理， 包括在线发布、 浏览等功能。

### b、学生管理

### 学生管理子系统实现对学生信息的管理，包括模糊查询、修改、添加、删除等功能。它包括以下四个模块 :

### ① 基本资料

### 此模块的功能是对学院学生的基本资料进行查询、 修改、添加、删除等操作。

### ②个人档案

### 此模块的功能是对学院学生的个人档案进行查询、 修改、添加、删除等操作。

### ③考勤录入

### 此模块的功能是对学院学生的日常考勤，比如迟到、早退、请假、旷课进行录入管理操作。

### ④考勤管理

### 此模块的功能是对学院学生日常考勤进行查询、修改、添加、删除等操作，以作为学生考核评定基础。

### c、教师管理

### 教师管理子系统与学生管理子系统的功能相似，实现对教师信息的管理，包

### 括查询、修改、添加、删除等功能。支持模糊查询、综合查询、防止误删除等功

### 能。

### d、教案管理

### 教案管理子系统实现对教案网上进行管理，包括查询、浏览、编辑、添加、

### 删除、改名等功能。其中教案是以文件的形式存放的，按系别和教师分别归类，

### 方便检索，系统也会自动列出所要查询的文件名称、文件类型、文件大小、文件

### 上次修改时间等信息。

### ☆成绩和选课管理功能

### 成绩管理子系统实现对学生每学年每门课的成绩进行管理， 包括查询、录入、

### 统计等功能。它包括四个功能模块：

### ①成绩查询模块

### 此模块功能是对学院学生每学年每门课的成绩进行查询，它支持智能查询、

### 综合查询。

### ②成绩录入模块

### 此模块功能是对学院学生每学年每门课的成绩进行录入操作，由于期中或期

### 末考试后往往是班级成绩成批录入， 所以有必要单独成为一子模块。 如果选择的

### 班级已有成绩， 则提示要么重新选择， 要么进入成绩处理模块进行所在班级的成

### 绩管理。录入完毕按“保存”则可查看已录入的记录。

### ③成绩处理模块

### 此模块的功能是对学院学生每学年各班成绩进行查询、修改、添加、删除等

### 操作，支持模糊查询、综合查询、防止误删除等功能，成绩存放在每学年的数据

### 库中。

### ④成绩统计模块

### 此模块的功能是对学院学生每学年各班成绩进行统计。包括按序号排序、按姓名排序、按成绩排序、统计及格率、统计平均分以及分数段统计。如果不选择课程，则默认对所有课程进行操作， 成绩排序按总分由高往低排列， 及格率和平均分按各课程统计。

### ☆邮箱管理功能

### 邮箱管理系统主要功能模块有：

### ①邮件发送模块：包括普通邮件正文及附件的发送、安全邮件正文及附件的送。

### 邮件接收模块： 包括普通邮件正文及附件的接收、 安全邮件的解密与接收。

### 邮件加密模块：包括对邮件正文及附件的加密与数字签名。

### ④邮件地址簿模块：包括邮件用户常用的收发件人地址及相关信息

### 性能

### 快捷性：信息检索时需快捷方便，并且搜索查询能力要强；

### 及时性：通过数遍点击链接， 网页要及时反应， 并且通知与新闻的更新要及

### 时和发布，这样才能使用户群了解更多的信息；

### 沟通性：交互的使用必须双方共同参与沟通， 此时交互的无障碍和及时反馈就显得很重要；

### 完整性：图书馆数据库系统的使用需使得数据保持完整性， 具备一般数据库所有的数据要求， 比如数据的及时更新和无冗余等， 数据库系统的功能也应该完整，用户在使用查询、更新、修改和删除等操作时，应该保证输出的数据结果是用户所想要的，数据结果必须完整全面和有价值；

### 保密性：人事管理的各项信息应该保密， 由专门人员进行处理， 并且人事信息必须及时更新和发布，当然数据也要进行备份和维护过程；

### 及时性：教务信息管理的各项功能应最及时的更新， 让用户及时得到所需信息并进行相关的反馈；

### 功能性：对于成绩和选课管理， 需要系统做到公平公正和公开原则， 在对成绩的录入、 查询和处理统计过程中， 系统的自动处理能力需要很强大， 比如自动排序、加权和统计等，而选课管理正选、补选等过程也需要系统自动处理，自动

### 校正信息；

### 4.1.6   类图

### 

### 4.1.7   主要方法

public List<xxxDto> QueryList(); //查询信息列表

### 4.1.8   数据库表

<Umooc学生学习时长统计表>表关系如下图：

### 4.1.9   UML模型

# 5  数据库设计

数据库表整体结构如下，详见《UMOOC教学系统数据库设计说明》。

# 6  安全性设计

## 6.1   用户权限

 老师可以进行学生信息的修改,课程的增删改查,上传资源,视频,批改作业等

,学生可以查询课程学习资源,学习时长,提交作业,观看视频,作业成绩等

## 6.2   软件授权

## 各大高校

# 7  设计与开发规范

## 7.1   界面设计原则

以用户为中心：Theo Mandel博士创造了人机交互的“黄金三原则”：置于用户控制之下；保持界面一致性；减轻用户的记忆负担。

拥有良好的直觉特征：以用户所熟悉的现实世界事务的抽象来给用户暗示和隐喻，来帮助用户快速地学会使用软件。

较快的响应速度。

简单且美观。

## 7.2   面向对象程序设计原则

**单一职责原则（SRP）：**就一个类而言，应该仅有一个引起它变化的原因。

**里氏替换原则（LSP）：**子类型（subtype）必须能够替换掉它们的基类型。

**依赖倒置原则（DIP）：**

a高层模块不应该依赖于低层模块，两者都应该依赖于抽象。

b抽象不应该依赖于细节，细节应该依赖于抽象。

**接口隔离原则（ISP）：**不应该强迫客户依赖于它们不要的方法。接口属于客户，不属于它所在的类层次结构。

**开闭原则（OCP）：**软件实体（类、模块、函数等）应该是可以扩展的，但是不可修改的。

**迪米特原则（LoD）：**也叫最少知识原则，即一个软件是他应当尽可能少地于其他实体发生相互作用，模块之间的交互要少。

## 7.3   编码规范

### 7.3.1   模块命名

每一个模块分一个包，包名以包名的英文单词命名。小模块依次放在所属的大模块下面，也是以模块的英文单词来命名。

### 7.3.2   类命名

数据访问类，用驼峰式命名法，首字母大写，以业务名称开头，Repository结尾，例如UserRepository.cs。

业务逻辑类，用驼峰式命名法，首字母大写，以业务名称开头，Service结尾，例如UserService.cs。

业务控制类，用驼峰式命名法，首字母大写，以业务名称开头，ViewModel结尾，例子如UserViewModel.cs。

### 7.3.3   变量命名

要以有意义的命名的方式来为变量命名，并且单词首字母小写，以驼峰式命名法。

### 7.3.4   过程函数命名

本系统的过程或函数都在Sql Server2012中运行，命名规则按驼峰式命名法，写好每一个变量、过程、函数、包的注释。变量以@开头，后接实际名称。

总之一点，不管程序代码中还是数据库的过程或函数，都要以清晰的代码结构和详细的注释，以此为标准。