# 

# Code Dream

## 一个编程自学网站

# 详细设计

|  |  |
| --- | --- |
| **成员** | **学号** |
| 卢彦谚 | 201630364818 |
| 吴潘安 | 201630665908 |
| 吴志祥 | 201630665939 |
| 谢旻珊 | 201630665984 |
| 谢珮爽 | 201630665991 |
| 谢哲威 | 201630666035 |
| 李林键 | 201630406969 |

目录

[Code Dream 1](#_Toc532566412)

[一个编程自学网站 1](#_Toc532566413)

[详细设计 1](#_Toc532566414)

[1概述 4](#_Toc532566415)

[1.1系统简述 4](#_Toc532566416)

[1.2软件设计目标 4](#_Toc532566417)

[1.3参考资料 4](#_Toc532566418)

[1.4修订版本记录 4](#_Toc532566419)

[1.5非功能性需求 5](#_Toc532566420)

[1.5.1性能需求 5](#_Toc532566421)

[1.5.2安全性需求 5](#_Toc532566422)

[1.5.3可靠性需求 5](#_Toc532566423)

[1.5.4兼容性需求 5](#_Toc532566424)

[1.5.5数据保密需求 5](#_Toc532566425)

[1.5.6可用性需求 6](#_Toc532566426)

[1.5.7可测试性需求 6](#_Toc532566427)

[1.5.8可维护性需求 6](#_Toc532566428)

[2术语表 6](#_Toc532566429)

[3设计概述 7](#_Toc532566430)

[3.1系统的复用计划 7](#_Toc532566431)

[3.1.1前端框架复用计划 7](#_Toc532566432)

[3.1.2后台框架复用计划 9](#_Toc532566433)

[3.2系统接口设计 10](#_Toc532566434)

[3.2.1用户界面设计 10](#_Toc532566435)

[3.2.2系统内部接口 13](#_Toc532566436)

[3.2.3系统外部接口 26](#_Toc532566437)

[3.3对象模型设计 27](#_Toc532566438)

[3.3.1对象模型 27](#_Toc532566439)

[3.3.2对象描述 28](#_Toc532566440)

[3.4系统功能实现详细设计 37](#_Toc532566441)

[3.4.1系统状态管理 37](#_Toc532566442)

[3.4.2注册 37](#_Toc532566443)

[3.4.3登录 38](#_Toc532566444)

[3.4.4注销 39](#_Toc532566445)

[3.4.5查看课程内容 39](#_Toc532566446)

[3.4.6查看个人中心 40](#_Toc532566447)

[3.4.7在线编译 40](#_Toc532566448)

[3.4.8查看帖子 41](#_Toc532566449)

[3.4.9发布帖子 41](#_Toc532566450)

[3.4.10删除帖子 42](#_Toc532566451)

[3.4.11修改帖子 42](#_Toc532566452)

[3.4.12收藏帖子 43](#_Toc532566453)

[3.4.13取消收藏 43](#_Toc532566454)

[3.4.14发布留言 44](#_Toc532566455)

[3.4.15修改留言 44](#_Toc532566456)

[3.4.16删除留言 45](#_Toc532566457)

[3.4.17课程管理 45](#_Toc532566458)

[3.4.18添加课程 46](#_Toc532566459)

[3.4.19修改课程 46](#_Toc532566460)

[3.4.20删除课程 46](#_Toc532566461)

[3.4.21添加章节 47](#_Toc532566462)

[3.4.22修改章节 47](#_Toc532566463)

[3.4.23删除章节 48](#_Toc532566464)

[3.4.24添加小节 48](#_Toc532566465)

[3.4.25修改小节 49](#_Toc532566466)

[3.4.26删除小节 49](#_Toc532566467)

[3.5系统非功能设计 49](#_Toc532566468)

[3.5.1性能设计 49](#_Toc532566469)

[3.5.2安全性设计 50](#_Toc532566470)

[3.5.3可靠性设计 50](#_Toc532566471)

[3.5.4数据保密设计 50](#_Toc532566472)

[3.5.5可用性设计 50](#_Toc532566473)

# 1概述

## 1.1系统简述

本项目的目标产品是一个前后端完整的线上编程知识自学网站，具备用户注册/登录系统、教学系统、用户信息管理系统、用户交流系统；具备用户-系统交互、用户-用户交互功能；该自学网站旨在通过线上课程为用户进行个性化学习管理；该自学网站的基本功能包括注册/登录、课程选择列表管理功能、知识图谱、实例展示、在线编译、留言板、留言板管理功能、折线图（每日在线时长管理）、学习进度管理、专注度管理、帮助功能。

## 1.2软件设计目标

互联网时代的发展，向新的一代发出新的需求。软件的地位水涨船高。编程能力成为大学生必备的一项新技能，从网上查找编程教程成为学习编程的主要方法。然而，网上的教程参差不齐，查找时间将花费我们大量的时间。

我们的编程学习网站将收集网上优秀的教程发布到网站上，给广大的编程学习者提供方便的途径。只需一个网站便能解决他们的许多问题，缩短他们的查找时间。与此同时，网站将提供一个给广大学习者交流的平台。他们通过留言留下自己的问题或者解决他人的问题，达到共同进步的目标。

## 1.3参考资料

①系统开题报告 《网页教程》开发小组

②需求规格说明书 《网页教程》开发小组

③网页低保真原型 《网页教程》开发小组

④系统概念数据模型 《网页教程》开发小组

⑤数据库系统概念》 西尔伯沙茨 机械工业出版社

⑥《UML和模式应用》（美）拉曼(Larman，C.)著李洋等译 机械工业出版社

## 1.4修订版本记录

|  |  |  |  |
| --- | --- | --- | --- |
| **修改日期** | **版本号** | **修改内容** | **修改人** |
|  |  |  |  |

## 1.5非功能性需求

### 1.5.1性能需求

①响应时间：在95％的情况下，一般时段响应时间不超过1.5秒，高峰时段不超过4秒；

②系统容量：支持3万用户，支持GB级数据且数据库表行数不超过100万行，数据库最大容量不超过1000GB，磁盘空间至少需要40G以上；

③资源使用率：CPU占用率<=50%。内存占用率<=50%。

### 1.5.2安全性需求

①保证管理者和注册用户的密码安全，数据库访问控制。

②提供运行日志管理及安全审计功能，可追踪系统的历史使用情况。

③系统要有足够的防御能力,防御普通黑客的攻击,尽量减少各种可能出现的漏洞,防止用户信息被不良分子窃取。

④严格权限访问控制，用户在经过身份认证后，只能访问其权限范围内的数据，只能进行其权限范围内的操作。

⑤能经受来自互联网的一般性恶意攻击。如病毒（包括木马）攻击、口令猜测攻击、黑客入侵等。至少99%的攻击需要在10秒内检测到。

### 1.5.3可靠性需求

①对输入有提示，数据有检查，防止数据异常；

②系统健壮性强，应该能处理系统运行过程中出现的各种异常情况，如：人为操作错误、输入非法数据、硬件设备失败等，系统应该能正确的处理，恰当的回避；

③因软件系统的失效而造成不能完成操作的概率要小于5‰；

④要求系统7x24小时运行，全年持续运行故障停运时间累计不能超过10小时。

### 1.5.4兼容性需求

系统应支持IOS，Android , windows操作系统;

### 1.5.5数据保密需求

网络传递数据应经过加密。需要保证数据在采集、传输和处理过程中不被偷窥、窃取、篡改。

### 1.5.6可用性需求

①在95%的故障中，系统最多需要20秒重启；

②提供数据备份和恢复功能，使得在由于系统的错误或其他原因引起系统的数据丢失或系统的数据被破坏时，能够及时恢复和还原数据。

### 1.5.7可测试性需求

①一个模块的最大圈复杂度不能超过15；

②交付的系统必须通过单元测试，并且是100%覆盖；

③开发活动必须使用回归测试，并允许在12小时内重新进行完整的测试。

### 1.5.8可维护性需求

从接到修改请求后，对于普通修改应在1~2天内完成；对于评估后为重大需求或设计修改应在1周内完成。及时修改软件以适应需求环境的变化。

# 2术语表

|  |  |
| --- | --- |
| **术语** | **术语解释** |
| 在线编译器 | 一个可用于编译代码并显示运行结果的网页。 |
| 示例代码 | 教程中，根据学习内容贴出的代码，用于演示当前的教学内容。 |
| 知识图谱 | 表示课程学习顺序的图，课程之间以代表学习顺序的箭头相连。 |
| 折线图 | 展示用户学习时间的变化情况，日期为横轴，学习时间为纵轴。 |
| 课程进度 | 通过课程列表下拉菜单中的课程图标或知识图谱中的课程图标设置动画，以“水位”动画展示课程进度。 |
| 专注森林 | 通过在网页右下角设置“森林”动画，通过树木生长高度、数量表示用户专注度的累积。 |
| 专注度 | 是指用户在登陆网站情况下进行的页面停留时长、课程花费时长与登陆时长的比值，用于衡量用户个人投入时间的效率。 |
| 在线时长 | 是指用户登陆网站的时间长度，从登陆开始计算。 |
| 邮件提醒 | 根据用户希望的学习时间发送学习提醒邮件 |

# 3设计概述

## 3.1系统的复用计划

### 3.1.1前端框架复用计划

#### 3.1.1.1框架介绍

前端采用复用Bootstrap的策略进行开发，前端界面布局设计、控件应用基于Bootstrap框架展开并补充。Bootstrap 提供了一个带有网格系统、链接样式、背景的基本结构。Bootstrap 包含了十几个可重用的组件，用于创建图像、下拉菜单、导航、警告框、弹出框等。Bootstrap自带了13个[jQuery](https://baike.baidu.com/item/jQuery)插件，包括：模式对话框、标签页、滚动条、弹出框等。

#### 3.1.1.2移动优先

·如何复用：

在<head>内部添加<meta name="viewport" content="width=device-width, initial-scale=1">

·简述：

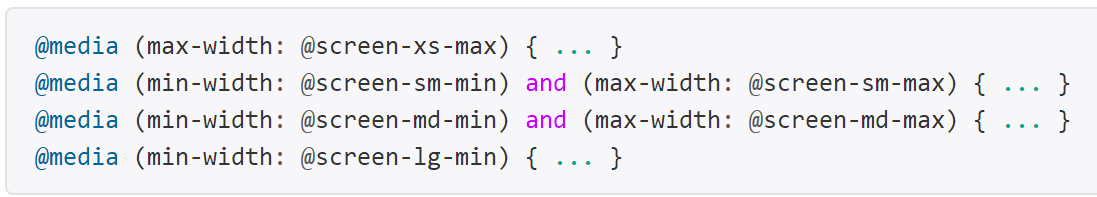
①width 属性控制设备的宽度。假设您的网站将被带有不同[屏幕分辨率](https://www.baidu.com/s?wd=%E5%B1%8F%E5%B9%95%E5%88%86%E8%BE%A8%E7%8E%87&tn=SE_PcZhidaonwhc_ngpagmjz&rsv_dl=gh_pc_zhidao)的设备浏览，那么将它设置为 device-width 可以确保它能正确呈现在不同设备上；  
②initial-scale=1.0 确保网页加载时，以 1:1 的比例呈现，不会有任何的缩放。

#### 3.1.1.3栅格系统

·如何复用：

在的div中使用col-xs-？ col-sm-? col-md-? col-lg-?布局；

使用@media对不同的设备屏幕大小设置不同的css样式。



·简述：

栅格系统中的列是通过指定1到12的值来表示其跨越的范围。例如，三个等宽的列可以使用三个 .col-xs-4 来创建。



#### 3.1.1.4Label标签

·如何复用：

将class放到特定的标签中。

·简述：

可用于教程中部分内容的高亮效果（如注意事项）



#### 3.1.1.5进度条

·如何复用：

使用progress类、progress-bar类，并调试相应属性（aria-valuenow,aria-valuemin,aria-valuemax）。

·描述：

可将进度条用于学习进度的展示。

3.1.1.6使用Font Awesome提供的图标

·如何复用：

<div> <span class="copyright">Copyright &copy; 2018 </span></div>

效果如图：

·描述：

Font Awesome提供了可缩放的矢量图标，可以使用CSS所提供的所有特性对它们进行更改，包括：大小、颜色、阴影或者其它任何支持的效果。

图标库链接：<http://fontawesome.dashgame.com/>

### 3.1.2后台框架复用计划

#### 3.1.2.1框架介绍

ThinkPHP是一个快速、兼容而且简单的轻量级国产PHP开发框架，使用[面向对象](https://baike.baidu.com/item/%E9%9D%A2%E5%90%91%E5%AF%B9%E8%B1%A1/2262089)的开发结构和MVC模式，融合了Struts的思想和TagLib（[标签库](https://baike.baidu.com/item/%E6%A0%87%E7%AD%BE%E5%BA%93/5088969)）、RoR的ORM[映射](https://baike.baidu.com/item/%E6%98%A0%E5%B0%84/410062)和ActiveRecord模式。ThinkPHP可以支持windows/[Unix](https://baike.baidu.com/item/Unix)/Linux等服务器环境，支持[MySql](https://baike.baidu.com/item/MySql/471251)、[PgSQL](https://baike.baidu.com/item/PgSQL)、[Sqlite](https://baike.baidu.com/item/Sqlite)多种数据库以及PDO扩展。其自身包含了底层架构、兼容处理、[基类](https://baike.baidu.com/item/%E5%9F%BA%E7%B1%BB)库、数据库访问层、[模板引擎](https://baike.baidu.com/item/%E6%A8%A1%E6%9D%BF%E5%BC%95%E6%93%8E/907667)、缓存机制、[插件](https://baike.baidu.com/item/%E6%8F%92%E4%BB%B6/369160)机制、角色认证、[表单](https://baike.baidu.com/item/%E8%A1%A8%E5%8D%95/5380322)处理等常用的组件，并且对于跨版本、跨平台和跨数据库移植都比较方便。

#### 3.1.2.2MVC分层结构用于模块化及代码复用

模型（M）：模型的定义由Model类来完成。

控制器（C）：应用控制器（核心控制器）和Action控制器都承担了控制器的角色，区别在于Action控制器完成业务过程，而应用控制器（App类）负责调度控制。

视图（V）：模板的实现是和框架无关的，做到了100%分离，可以独立预览和制作。

#### 3.1.2.3数据访问中的可复用：Dao对象

Thinkphp自带DataAccessObject（数据访问对象），Dao设计模式把底层的数据访问逻辑和业务逻辑分开，夹在业务层与数据层中间，用于数据库访问和操作。可以针对单表和多表操作。ThinkPHP中的Dao对象属于建立在抽象数据库访问层之上的事务性对象。

#### 3.1.2.4业务逻辑和数据访问中的可复用：VO对象

ThinkPHP包含ValueObject（数据对象），VO对象是存活在业务层的，供业务逻辑使用的。它和数据表存在一种映射关系，VO对象的每个属性对应数据表（或者视图）的一个字段，可以添加其他非数据表字段的VO对象属性，以用于业务逻辑操作，例如关联字段。VO对象有个特殊的保留属性\_info，是一个数组变量，用来保存该VO对象的一些特殊信息。和VO对象对应的概念还有PO（持久数据对象），VO和PO的概念都来自于O/RMapping（ObjectRelationalMapping，对象关系映射），PO只能用在数据层（也就是说PO对象的属性必须和数据表一一对应），VO用在业务逻辑层和表示层（可以有自己的额外属性）。各层操作属于该层自己的数据对象，这样就可以降低各层之间的耦合，便于以后系统的维护和扩展。

#### 3.1.2.5业务逻辑获取到数据对象后遍历和操纵的可复用：VoList对象

在ThinkPHP框架中，把查询后的数据集（ResultSet）转换成数据列表对象，也就是VOLIST对象，和ResultSet对象不同的是每个记录就相当于一个VO对象。在VOLIST对象中可以方便地遍历和操作VO对象。

#### 3.1.2.6应用逻辑中的可复用：Action对象

Action对象主要提供应用逻辑的控制和处理，属于ThinkPHP的控制层，由核心的App类完成模块和操作的解析后直接转交给相应的Action对象处理。应用类中的所有Action对象都继承系统的Action基类。

## 3.2系统接口设计

### 3.2.1用户界面设计

#### 3.2.1.1主页设计

##### 3.2.1.1.1导航栏

* 名称：导航栏
* 特性：响应式组件
* HTML标签：<div class="collapse navbar-collapse">
* CSS特殊属性：
* JavaScript事件：
* 备注：导航栏在移动设备的视图中是折叠的，随着可用视口宽度增加，导航栏会水平展开，复用Bootstrap框架。

##### 3.2.1.1.2布局容器

* 名称：布局容器
* 特性：固定宽度并支持响应式
* HTML标签：<div class="container">
* CSS特殊属性：
* JavaScript事件：
* 备注：

##### 3.2.1.1.3轮播幻灯片

* 名称：轮播幻灯片
* 特性：循环播放风格展示图片
* HTML标签：<ul class="slides-container text-center">
* CSS特殊属性：
* JavaScript事件：
* 备注：通过复用bootstrap框架的轮播插件实现幻灯片式画面展示。

##### 3.2.1.1.4隐藏菜单

* 名称：隐藏菜单
* 特性：具备隐藏/展开两种状态
* HTML标签：<a style="display: none;" onmousemove="over()" onmouseout="out()" >
* CSS特殊属性：style="display:none;"和style="display:block;"
* JavaScript事件：

function over(){}

function out(){}

* 备注：通过监听鼠标移入、移出事件调用JavaScript函数更改display属性从而实现菜单的隐藏和展开。

#### 3.2.1.2注册登录页设计

##### 3.2.1.2.1注册/登录面板

* 名称：注册/登录面板
* 特性：动态面板，可切换注册/登录状态
* HTML标签：

<div id="loginPage" class="l-tab-list current" onclick="changeState()">登录</div>

<div id="registerPage" class="l-tab-list" onclick="changeState()">注册</div>

* CSS特殊属性：style="display:none;"和style="display:block;"
* JavaScript事件：function changeState(){}
* 备注：通过面板容器点击鼠标事件调用JavaScript函数切换面板的“display”属性从而进行注册/登录面板可见性切换。

##### 3.2.1.2.2动态按钮

* 名称：动态按钮
* 特性：可切换可用/不可用状态
* HTML标签：
* CSS特殊属性：
* JavaScript事件：

window.onload =function(){

document.getElementById("password").onblur = function(){}

document.getElementById("registerPassword").onblur = function(){}

}

* 备注：页面加载时先设置按钮不可用，只有当用户填写完所需的注册/登录信息后，通过监听文本框失去焦点事件调用JavaScript函数，按钮切换为可用状态。

#### 3.2.1.3留言板页设计

##### 3.2.1.3.1文章容器

* 名称：文章容器
* 特性：动态面板，可切换注册/登录状态
* HTML标签：<article class="post">
* CSS特殊属性：
* JavaScript事件：
* 备注：<article> 标签定义外部的内容。

##### 3.2.1.3.2页码栏

* 名称：页码栏
* 特性：只显示当前页面前后5页页码，以及第1页和最后一页，其他页码隐藏
* HTML标签：<ul class="pagination">
* CSS特殊属性：
* JavaScript事件：
* 备注：复用Bootstrap提供的处理分页的class，通过.active指示当前页面。

#### 3.2.1.4个人中心页设计

* 名称：页面停留时长统计
* 特性：有效计算用户浏览页面时长
* HTML标签：
* CSS特殊属性：
* JavaScript事件：

window.onload()

window.onbeforeunload()

document.addEventListener('visibilitychange', function (event) {

console.log(document.hidden, document.visibilityState)

})

* 备注：在页面进入window.onload()和离开window.onbeforeunload()事件间计算用户页面开启时长，需要注意，用户浏览时长并不是用户页面开启时长，所以需要考虑页面的活动状态切换。

### 3.2.2系统内部接口

##### 3.2.2.1登陆注册模块

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 名称 | | 登陆检测 | | | |
| 接口地址 | | /login/login | | | |
| 类型 | | POST | | | |
| 描述 | | 进行登录验证，根据用户输入的用户名和密码返回是否登录成功 | | | |
| 类型 | 数据类型 | | 变量名 | 描述 |
| request | String | | mail | 用户名 |
| request | String | | password | 密码 |
| response | Number | | responseStatus | 0-普通用户，1-游客，2-管理员， -1表示异常 |
| response | String | | errorMessage | status为-1时使用 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 名称 | | 注册 | | |
| 接口地址 | | /login/register | | |
| 类型 | | POST | | |
| 描述 | | 注册接口，传入用户名，密码，昵称，返回是否注册成功 | | |
| 类型 | 数据类型 | | 变量名 | 描述 | |
| request | String | | nickname | 昵称 | |
| request | String | | password | 密码 | |
| request | String | | mail | 邮箱 | |
| response | Number | | responseStatus | 后台返回状态码，0表示成功注册，1表示用户名重复 | |

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | | 注销 | |
| 接口地址 | | /login/logout | |
| 类型 | | POST | |
| 描述 | | 注销功能 | |
| 类型 | 数据类型 | | 变量名 | | 描述 |
| response | Number | | responseStatus | | 后台返回状态，0表示注销成功 |

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | | 判断用户状态 | |
| 接口地址 | | /login/judge | |
| 类型 | | GET | |
| 描述 | | 获取当前用户的登陆状态及身份信息 | |
| 类型 | 数据类型 | | 变量名 | | 描述 |
| response | Number | | responseStatus | | 0-普通用户，1-游客，2-管理员 |

##### 3.2.2.2交流栈模块

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 名称 | | 获取帖子列表 | | |
| 接口地址 | | /blog/getBlogList | | |
| 类型 | | GET | | |
| 描述 | | 获取用户点击进入的课程板块内的帖子列表 | | |
| 类型 | 数据类型 | | 变量名 | 描述 | |
| response | Array | | blog | 帖子数组 | |
| response | String | | title | 帖子标题 | |
| response | String | | time | 帖子发表时间 | |
| response | String | | authorName | 作者名 | |
| response | Number | | commentCount | 帖子的留言数 | |
| response | String | | comment | 帖子内容第一段预览 | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 名称 | | 进入帖子 | | |
| 接口地址 | | /blog/enter | | |
| 类型 | | POST | | |
| 描述 | | 进入帖子前传给后台session 用户点进去的帖子的id | | |
| 类型 | 数据类型 | | 变量名 | 描述 | |
| request | Number | | blogID | 用户选择的帖子的id | |
| response | Number | | responseStatus | 0-成功，1-失败 | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 名称 | | 获取帖子内容 | | |
| 接口地址 | | /blog/getBlog | | |
| 类型 | | GET | | |
| 描述 | | 获取帖子内容及帖子对应的留言 | | |
| 类型 | 数据类型 | | 变量名 | 描述 | |
| response | String | | title | 帖子标题 | |
| response | String | | time | 帖子发表时间 | |
| response | String | | authorName | 帖子作者名 | |
| response | String | | content | 帖子内容 | |
| response | Number | | responseStatus | 返回状态 | |
| response | Array | | comment | 帖子评论数组 | |
| response | String | | nickName | 该评论作者名 | |
| response | String | | time | 该评论发表时间 | |
| response | String | | headLink | 该评论用户的头像 | |
| response | String | | content | 评论内容 | |
| response | Number | | replyID | 该评论回复的对象的ID | |
| response | Boolean | | editable | 后台判断这个留言是否是当前登陆的用户发的，是则返回true，则留言角落应该出现编辑选项 | |
| response | Number | | collected | 0代表用户没收藏当前帖子，1代表已收藏 | |
| response | String | | replyName | 被回复留言的作者名 | |

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | | 发表帖子 | |
| 接口地址 | | /blog/addBlog | |
| 类型 | | POST | |
| 描述 | |  | |
| 类型 | 数据类型 | | 变量名 | | 描述 |
| request | String | | content | | 帖子内容 |
| request | String | | title | | 帖子名 |
| response | Number | | responseStatus | | 返回状态，0代表正常返回 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 名称 | | 编辑帖子 | | |
| 接口地址 | | /blog/modifyBlog | | |
| 类型 | | POST | | |
| 描述 | |  | | |
| 类型 | 数据类型 | | 变量名 | 描述 | |
| request | String | | title | 修改后的帖子的标题 | |
| request | String | | content | 修改后的帖子的内容 | |
| request | Number | | blogID | 文章的ID | |
| response | Number | | responseStatus | 返回状态，0代表正常返回 | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 名称 | | 删除帖子 | | |
| 接口地址 | | /blog/deleteBlog | | |
| 类型 | | POST | | |
| 描述 | |  | | |
| 类型 | 数据类型 | | 变量名 | 描述 | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 名称 | | 回复留言 | | |
| 接口地址 | | /blog/replyComment | | |
| 类型 | | POST | | |
| 描述 | | 回复留言 | | |
| 类型 | 数据类型 | | 变量名 | 描述 | |
| request | String | | content | 留言内容 | |
| request | String | | replyID | 被回复的留言的ID，没有则为-1 | |
| response | Number | | responseStatus | 返回状态，0代表正常返回 | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 名称 | | 修改留言 | | |
| 接口地址 | | /blog/modifyComment | | |
| 类型 | | POST | | |
| 描述 | | 修改留言接口 | | |
| 类型 | 数据类型 | | 变量名 | 描述 | |
| request | String | | content | 新留言 | |
| request | Number | | commentID | 需要修改的留言的id | |
| response | Number | | responseStatus | 返回状态，0代表正常返回 | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 名称 | | 删除留言 | | |
| 接口地址 | | /blog/deleteComment | | |
| 类型 | | POST | | |
| 描述 | | 删除留言接口 | | |
| 类型 | 数据类型 | | 变量名 | 描述 | |
| request | Number | | commentID | 要删除的留言的id | |
| response | Number | | responseStatus | 返回状态，0代表正常返回 | |

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | | 收藏帖子 | |
| 接口地址 | | /blog/collect | |
| 类型 | | POST | |
| 描述 | |  | |
| 类型 | 数据类型 | | 变量名 | | 描述 |
| response | Number | | responseStatus | | 0-收藏成功，1-未登录，2-收藏失败 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 名称 | | 取消收藏 | | |
| 接口地址 | | /blog/cancelCollect | | |
| 类型 | | POST | | |
| 描述 | |  | | |
| 类型 | 数据类型 | | 变量名 | 描述 | |
| response | Number | | responseStatus | 返回状态，0代表正常返回 | |

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | | 最新帖子 | |
| 接口地址 | | /blog/getNew | |
| 类型 | | GET | |
| 描述 | | 获取整个交流栈板块最新发布的五个帖子 | |
| 类型 | 数据类型 | | 变量名 | | 描述 |
| response | Array | | blog | | 帖子数组 |
| response | String | | title | | 帖子标题 |
| response | String | | blogID | | 帖子ID |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 名称 | | 删除收藏的帖子 | | |
| 接口地址 | | /user/deleteBlog | | |
| 类型 | | POST | | |
| 描述 | |  | | |
| 类型 | 数据类型 | | 变量名 | 描述 | |
| request | String | | blogID | 被取消收藏的帖子的ID | |
| response | Number | | responseStatus | 0-成功，1-失败 | |

##### 3.2.2.3用户信息模块

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | | 修改信息 | |
| 接口地址 | | /user/modify | |
| 类型 | | POST | |
| 描述 | |  | |
| 类型 | 数据类型 | | 变量名 | | 描述 |
| request | String | | nickname | | 用户名 |
| request | String | | mail | | 邮箱 |
| response | Number | | responseStatus | | 0-成功，1-失败 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 名称 | | 获取所有信息 | | |
| 接口地址 | | /user/getAllInfo | | |
| 类型 | | GET | | |
| 描述 | | 获取用户所有信息，包括：\n"gender":性别,\n"mail":邮箱,\n"nickName":昵称,\n"course":已上课程数组，\n"learnTime":最近7天在线时间列表 | | |
| 类型 | 数据类型 | | 变量名 | 描述 |
| response | String | | nickname | 昵称 |
| response | String | | mail | 邮箱 |
| response | Number | | gender | 性别，0女1男 |
| response | Array | | course | 用户学习过的课程（有学习经历或完全学习完） |
| response | String | | courseName | 课程名 |
| response | Array | | learnTime | 用户最近7天在线时间列表 |
| response | Number | | time | 每天在线时间（分钟） |
| response | Number | | courseID | 0-html 1-css 2-js 3-php |
| response | String | | studyProgress | 1%-100%的字符串 |
| response | String | | dateString | 2018-11-09 此格式的字符串 |
| response | Array | | collectBlog | 个人收藏 |
| response | String | | author | 收藏的帖子的作者 |
| response | String | | desc | 收藏的帖子的标题 |
| response | String | | blogID | 收藏的帖子的ID |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 名称 | | 更新专注度 | | |
| 接口地址 | | /user/concentrate | | |
| 类型 | | POST | | |
| 描述 | | 用户进行切回首页、离开页面等操作，更新用户学习的专注度。此接口作用只是通知后台用户离开了，把专注度更新一下 | | |
| 类型 | 数据类型 | | 变量名 | 描述 |
| response | Number | | responseStatus | 返回状态，0代表正常返回 |

##### 3.2.2.4在线编译模块

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 名称 | | 在线编译 | | |
| 接口地址 | | /compiler/compile | | |
| 类型 | | POST | | |
| 描述 | | 在线编译功能，获取用户输入的代码，确定用户设置的语言、语言版本、编译器，调用接口，返回代码结果 | | |
| 类型 | 数据类型 | | 变量名 | 描述 |
| request | String | | language | 用户选择的编程语言 |
| request | Number | | version | 用户选择的版本号 |
| request | String | | code | 用户输入的代码 |
| response | String | | result | 运行过后的程序结果 |
| response | Number | | responseStatus | 返回的状态码（0为成功运行，1为编译错误） |

##### 3.2.2.5课程模块

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 名称 | | 获取课程列表 | | |
| 接口地址 | | /course/getCourseList | | |
| 类型 | | GET | | |
| 描述 | | 获取已有的课程的列表，课程名，id和简要介绍 | | |
| 类型 | 数据类型 | | 变量名 | 描述 | |
| response | Number | | responseStatus | 返回状态，0代表正常返回 | |
| response | Array | | courseList | 课程列表 | |
| response | String | | name | 课程名 | |
| response | String | | profile | 课程简介（用于主页） | |
| response | String | | courseID | 课程ID | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 名称 | | 获取课程内容 | | |
| 接口地址 | | /course/getCourse | | |
| 类型 | | GET | | |
| 描述 | | 获取课程名字和课程页面的长summary | | |
| 类型 | 数据类型 | | 变量名 | 描述 | |
| response | String | | name | 课程名 | |
| response | Number | | responseStatus | 返回状态，0代表正常返回 | |
| response | String | | summary | 课程长简介 | |
| response | String | | shortSummary | 就是主页的profile | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 名称 | | 获取章节小节列表 | | |
| 接口地址 | | /course/getChapter | | |
| 类型 | | GET | | |
| 描述 | | 根据课程id获取章节。返回状态码和章节数组的数组，每个章节数组包含章节名、章节简介和一个小节数组，小节数组内包含若干小节，每个小节包括一个小节名 | | |
| 类型 | 数据类型 | | 变量名 | 描述 |
| response | Array | | chapter | 章节 |
| response | String | | chapterName | 章节名 |
| response | Number | | responseStatus | 返回状态，\n0代表正常返回，\n1代表课程id有误 |
| response | Number | | chapterID | 章节ID |
| response | Array | | section | 小节名 |
| response | String | | sectionName | 小节名 |
| response | Number | | sectionID | 小节ID |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 名称 | | 选择课程板块/帖子版块 | | |
| 接口地址 | | /course/selectCourse | | |
| 类型 | | POST | | |
| 描述 | | 根据用户点击的课程留言板块传给后台session 课程id | | |
| 类型 | 数据类型 | | 变量名 | 描述 | |
| request | Number | | courseID |  | |
| response | Number | | responseStatus |  | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 名称 | | 获取小节内容 | | |
| 接口地址 | | /course/getContent | | |
| 类型 | | GET | | |
| 描述 | | 根据 course ID，chapter ID， section ID 获取具体内容 | | |
| 类型 | 数据类型 | | 变量名 | 描述 | |
| request | Number | | sectionID | 小节ID | |
| response | Number | | responseStatus | 查询状态，0代表成功查询到课程内容，1代表小节id错误 | |
| response | String | | content | markdown格式的课程内容文本，需要调用Parser.js 转成html | |
| response | String | | title | 小节标题 | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 名称 | | 获取课程进度 | | |
| 接口地址 | | /course/process | | |
| 类型 | | GET | | |
| 描述 | | 传入用户当前选择的课程，获取用户在这门课的学习进度 | | |
| 类型 | 数据类型 | | 变量名 | 描述 | |
| response | Number | | progress | 课程进度，以百分比返回 | |
| response | Number | | responseStatus | 返回状态，0代表成功 | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 名称 | | 修改课程内容 | | |
| 接口地址 | | /course/editCourse | | |
| 类型 | | POST | | |
| 描述 | | 管理员页面，用于修改课程内容 | | |
| 类型 | 数据类型 | | 变量名 | 描述 | |
| request | String | | name | 新课程名 | |
| request | String | | shortSummary | 用于主页的短介绍 | |
| request | String | | longSummary | 课程页面的长介绍 | |
| request | Number | | preCourseID | 前置课程ID | |
| response | Number | | responseStatus | 返回状态，0代表成功 | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 名称 | | 新建课程 | | |
| 接口地址 | | /course/addCourse | | |
| 类型 | | POST | | |
| 描述 | | 管理员页面，用于新建课程 | | |
| 类型 | 数据类型 | | 变量名 | 描述 | |
| request | String | | shortSummary | 用于主页的短课程介绍 | |
| request | Number | | preCourseID | 前置课程ID | |
| request | String | | courseName | 新课程名 | |
| request | String | | longSummary | 用于课程的长课程介绍 | |
| response | Number | | responseStatus | 返回状态，0代表成功 | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 名称 | | 修改章节名 | | |
| 接口地址 | | /course/editChapter | | |
| 类型 | | POST | | |
| 描述 | |  | | |
| 类型 | 数据类型 | | 变量名 | 描述 | |
| request | Number | | chapterID | 章节ID | |
| request | String | | name | 章节名 | |
| response | Number | | responseStatus | 返回状态，0代表成功 | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 名称 | | 删除章节 | | |
| 接口地址 | | /course/deleteChapter | | |
| 类型 | | POST | | |
| 描述 | | 根据课程id和章节id删除一整个章节 | | |
| 类型 | 数据类型 | | 变量名 | 描述 | |
| request | Number | | chapterID | 章节ID | |
| response | Number | | responseStatus | 返回状态，0代表成功 | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 名称 | | 修改小节内容 | | |
| 接口地址 | | /course/editSection | | |
| 类型 | | POST | | |
| 描述 | | 修改小节内容 | | |
| 类型 | 数据类型 | | 变量名 | 描述 | |
| request | Number | | sectionID | 小节ID | |
| request | Number | | name | 修改后的小节标题 | |
| request | String | | content | 修改后的小节内容 | |
| response | Number | | responseStatus | 返回状态，0代表成功 | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 名称 | | 增加章节 | | |
| 接口地址 | | course/addChapter | | |
| 类型 | | POST | | |
| 描述 | |  | | |
| 类型 | 数据类型 | | 变量名 | 描述 | |
| request | String | | chapterTitle | 章节名 | |
| response | Number | | responseStatus | 返回状态，0代表成功 | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 名称 | | 增加小节 | | |
| 接口地址 | | course/addSection | | |
| 类型 | | POST | | |
| 描述 | |  | | |
| 类型 | 数据类型 | | 变量名 | 描述 | |
| request | String | | title | 小节标题 | |
| request | String | | chapterID | 章节ID | |
| request | String | | content | 小节内容 | |
| response | Number | | responseStatus | 返回状态，0代表成功 | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 名称 | | 删除课程 | | |
| 接口地址 | | /course/deleteCourse | | |
| 类型 | | POST | | |
| 描述 | | 根据课程id删除整个课程 | | |
| 类型 | 数据类型 | | 变量名 | 描述 | |
| request | Number | | courseID | 要删除的课程ID | |
| response | Number | | responseStatus | 返回状态，0代表成功 | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 名称 | | 删除小节 | | |
| 接口地址 | | /course/deleteSection | | |
| 类型 | | POST | | |
| 描述 | | 根据小节id删除小节 | | |
| 类型 | 数据类型 | | 变量名 | 描述 | |
| request | Number | | sectionID | 要删除的小节的ID | |
| response | Number | | responseStatus | 返回状态，0代表成功 | |

### 3.2.3系统外部接口

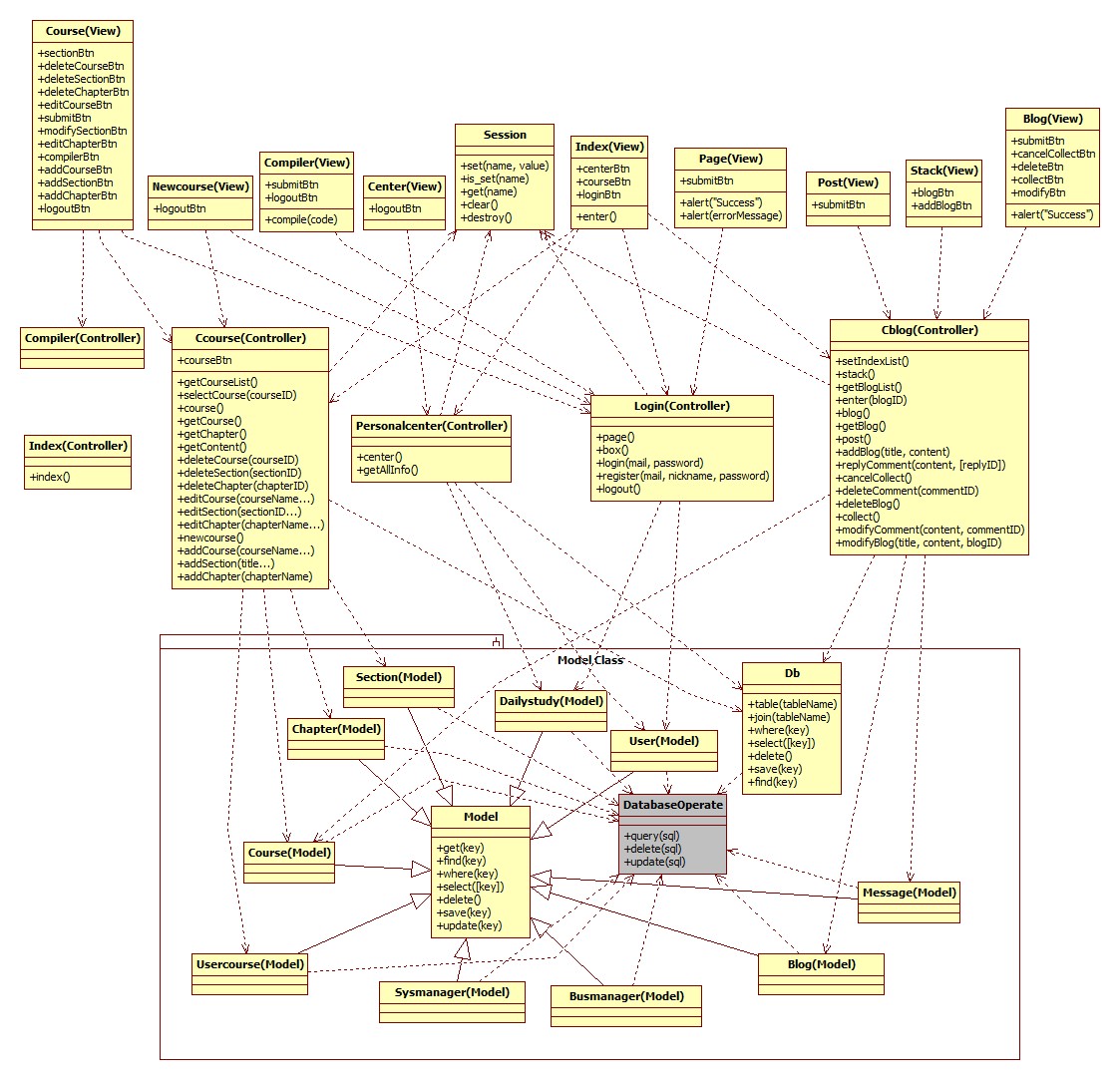
##### 3.2.3.1 QQ登陆接口

* 地址：https://openmobile.qq.com/user/get\_simple\_userinfo
* 类型：GET
* 状态码：0
* 简介：获取登录用户在QQ空间的信息，包括昵称、头像、性别及黄钻信息，降低用户的注册成本

|  |  |
| --- | --- |
| 参数 | 含义 |
| access\_token | 访问接口时的密钥 |
| oauth\_consumer\_key | 申请QQ登录成功后，分配给应用的appid |
| openid | 用户的ID，与QQ号码一一对应。 |

## 3.3对象模型设计

### 3.3.1对象模型



### 3.3.2对象描述

#### 3.3.2.1登陆用例

##### 3.3.2.1.1 Login View

|  |  |  |
| --- | --- | --- |
| id | 标签类型 | 描述 |
| unameLogin | <input> | 用户名的输入文本框 |
| pwdLogin | <input> | 密码的输入文本框 |
| loginBtn | <button></button> | 确定登陆的按钮 |

|  |  |  |
| --- | --- | --- |
| 属性名字 | 类型 | 描述 |
| username | var | 用户名 |
| password | var | 密码 |

|  |  |  |  |
| --- | --- | --- | --- |
| 方法 | 参数 | 用途 | 使用的接口 |
| login() | username,password | 数据传到login.php，验证登录信息，验证成功后在本窗口打开MainPage页面。 | ajax |

##### 3.3.2.1.3 Login.php

简述：Login.php相当于用户在登陆时触发的一个main函数，调用myDB封装好的方法验证用户登陆信息，并创建用户名、登陆时间的session，用于其他功能的实现。

|  |  |  |
| --- | --- | --- |
| 名字 | 类型 | 描述 |
| username | 字符串 | 账号 |
| password | 字符串 | 密码 |
| loginTime | 字符串 | 登陆时的时间 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **方法内部功能** | 参数 | 返回类型 | 用途 | 概要步骤 |
| 检验登录信息 | username,  password | 字符串 | 用于验证登陆信息。 | ①创建myDB类的对象  ②使用myDB类封装好的select方法访问数据库，select函数返回验证结果  ③返回验证结果 |
| 将用户名存到session中 | username | 无 | 将username存到session中。 | 使用session类封装好的set函数。 |
| 获取当前时间 | format,  time() | 字符串 |  | 使用Date(format,time()) |
| 将登录时间存储到session中 | loginTime | 无 | 将loginTime存到session中，以便登出时获取。 | 使用session类封装好的set函数。 |

#### 3.3.2.2注册用例

##### 3.3.2.2.1 Login/RegisterView

|  |  |  |
| --- | --- | --- |
| id | 标签类型 | 描述 |
| unameReg | <input></input> | 用户名的输入文本框 |
| pwdReg | <input></input> | 密码的输入文本框 |
| email | <input></input> | 邮箱的输入文本框 |
| registerBtn | <button></button> | 确定登陆的按钮 |

##### 3.3.2.2.2 Register.js

|  |  |  |
| --- | --- | --- |
| 属性名字 | 类型 | 描述 |
| username | var | 用户名 |
| password | var | 密码 |
| email | var | 邮箱 |

|  |  |  |  |
| --- | --- | --- | --- |
| 方法 | 参数 | 用途 | 使用的接口 |
| register() | username,password, email | 将数据传到  register.php，存储用户的注册信息后，向Login/RegisterView返回注册成功。 | ajax |

##### 3.3.2.2.3 Register.php

简述：Register.php相当于用户在注册时触发的一个main函数，调用myDB封装好的方法插入用户的注册信息到数据库中。

|  |  |  |
| --- | --- | --- |
| 名字 | 类型 | 描述 |
| username | 字符串 | 账号 |
| password | 字符串 | 密码 |
| email | 字符串 | 邮箱 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **方法内部功能** | 参数 | 返回类型 | 用途 | 概要步骤 |
| 插入注册信息 | ①username  ②password  ③email | 字符串 | 用于验证登陆信息。 | ①创建myDB类的对象  ②使用myDB类封装好的select方法访问数据库，插入注册信息。  ③返回插入状态 |

#### 3.3.2.3注销用例

##### 3.3.2.3.1 Logout.php

简述：Login.php相当于用户在注销时触发的一个main函数，通过session获取用户的登陆时间、鼠标悬停时间（学习时长），计算在线时长、专注度，调用myDB封装好的方法将用户的在线时长和专注度到数据库中。

|  |  |  |
| --- | --- | --- |
| 名字 | 类型 | 描述 |
| username | 字符串 | 账号 |
| loginTime | 字符串 | 登陆时间 |
| onlineTime | 字符串 | 在线时长 |
| onMouseTime | 字符串 | 鼠标停留在页面的时长 |
| concentration | 字符串 | 根据onMouseTime计算的专注度 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **方法内部功能** | 参数 | 返回类型 | 用途 | 概要步骤 |
| calculateOnlineTime  计算在线时间 | loginTime | 字符串 | 获得用户今日的在线时长 | ①使用Date(format,time())获得注销时间logoutTime  ②计算获得onlineTime |
| calculateConcentration  计算专注度 | ①onMouseTime  ②onlineTime | 字符串 | 获得用户今日的学习专注度 | 使用专注度计算公式计算用户今日的学习专注度 |
| storeTodayStudy  存储今日学习状况 | ①userName  ②loginTime  ③concentration | 字符串 | 存储用户今日的学习情况（在线时间+专注度） | ①创建myDB类的对象  ②使用myDB类封装好的select方法访问数据库，  插入onlineTime和concentration。  ③返回插入状态 |

#### 3.3.2.4选择课程用例

##### 3.3.2.4.1 MainPageView

|  |  |  |
| --- | --- | --- |
| id | 标签类型 | 描述 |
| html | <div></div> | html课程跳转框 |
| css | <div></div> | css课程跳转框 |
| js | <div></div> | js课程跳转框 |
| php | <div></div> | php课程跳转框 |
| personalBtn | <li></li> | 导航栏中的个人中心按钮 |

##### 3.3.2.4.2 CourseView

在课程选择动作做出后，动态添加的标签：

|  |  |  |
| --- | --- | --- |
| id | 标签类型 | 描述 |
| mainHead | <p></p> | 章节标题 |
| mainDes | <p></p> | 章节简述 |
| sec1\_head | <p></p> | 小节1标题 |
| sec1\_des | <p></p> | 小节1简述 |
| sec1\_tip | <p></p> | 小节1 tips |
| sec2\_head | <p></p> | 小节2标题 |
| sec2\_des | <p></p> | 小节2简述 |
| sec2\_tip | <p></p> | 小节2 tips |
| sec3\_head | <p></p> | 小节3标题 |
| sec3\_des | <p></p> | 小节3简述 |
| sec3\_tip | <p></p> | 小节3 tips |
| sec4\_head | <p></p> | 小节4标题 |
| sec4\_des | <p></p> | 小节4简述 |
| sec4\_tip | <p></p> | 小节4 tips |
| ... | ... | ... |

##### 3.3.2.4.3 selectCourse.js

|  |  |  |
| --- | --- | --- |
| 属性名字 | 类型 | 描述 |
| courseName | var | 课程名称，值可能为html、css、js、php… |
| chapterArr | 一维数组 | 章节标题和简述 |
| secArr | 二维数组 | 记录小节内容的二维数组 |

|  |  |  |  |
| --- | --- | --- | --- |
| 方法 | 参数 | 用途 | 算法 |
| chooseCourse() | self | 获取触发此次click事件的元素的id，判断被选择的课程，将courseName传到  selectCourse.php。 | ajax |
| addChapterDiv | chapterRow | 使用chapterArr数组中的内容向course.html中动态添加章节的标题和简介。 |  |
| addSecDiv | secArr | 使用secArr数组中的内容向course.html中动态添加每个小节的内容。 |  |

##### 3.3.2.4.4 SelectCourse.php

简述：selectCourse.php相当于用户在选择课程时触发的一个main函数，通过session获取用户名，从而获得与该用户相应的课程进度（章节），并获得章节内容的两个数组，传回给selecrCourse.js。

|  |  |  |
| --- | --- | --- |
| 名字 | 类型 | 描述 |
| chapterRow | 一维数组 | 存储了章节名、简述的一维数组 |
| secArray | 二维数组 | 存储了小节信息的二维数组 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **方法内部功能** | 参数 | 返回类型 | 用途 | 概要步骤 |
| getUsername  从session中获得userName | ‘userName’ | 字符串 | 获得username，用于描写sql语句的，以获得用户相应的学习进度，并展示相应章节。 | 使用session类的get() |
| getChapterRow  获得章节数组 | courseName | 一维数组 | 获得章节名、简述，以一维数组的形式。 | ①使用myDB类封装好的select方法访问数据库，select函数返回数组。 |
| getSecArray  获得小节数组 | courseName | 二维数组 | 获得每个小节的具体内容，以二维数组的形式。 | ①使用myDB类封装好的select方法访问数据库，select函数返回数组。 |

#### 3.3.2.5查看个人中心（折线图）用例

##### 3.3.2.5.1 PersonalCenterView

|  |  |  |
| --- | --- | --- |
| id | 标签类型 | 描述 |
| personInformation | <div></div> | 个人信息栏 |
| courseLearnt | <div></div> | 已学课程栏 |
| lineChartCanvas | <canvas></canvas> | 折线图的绘制画布 |

##### 3.3.2.5.2 PersonalCenter.js

|  |  |  |
| --- | --- | --- |
| 属性名字 | 类型 | 描述 |
| informationRow | 一维数组 | 个人信息数组（用户名、昵称、邮箱、性别） |
| courseNameRow | 一维数组 | 已学的课程数组 |
| learnTimeRow | 一维数组 | 近7天的在线时间数组 |

|  |  |  |
| --- | --- | --- |
| 方法 | 参数 | 用途 |
| addInformationDiv | informationRow | 使用informationRow数组中的内容向PersonalCenter.html中的个人信息div添加信息。 |
| addToCourseDiv | courseNameRow | 使用courseNameRow数组中的内容向PersonalCenter.html中的已学课程div添加信息。 |
| addToTimeChartDiv | learnTimeRow | 使用learnTimeRow数组中的内容向PersonalCenter.html中的折线图画布canvas添加信息。 |

##### 3.3.2.5.3 PersonalCenter.php

简述：PersonalCenter.php相当于用户在点击个人中心时触发的一个main函数，通过session获取用户名，并获得个人信息、已学课程、近7日学习时间的三个数组，并传回给PersonalCenter.js。

|  |  |  |
| --- | --- | --- |
| 名字 | 类型 | 描述 |
| informationRow | 一维数组 | 个人信息数组  （用户名、昵称、邮箱、性别） |
| courseNameRow | 一维数组 | 已学的课程数组 |
| learnTimeRow | 一维数组 | 近7天的在线时间数组 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **方法内部功能** | 参数 | 返回类型 | 用途 | 概要步骤 |
| getInformationRow  获得个人信息数组 | uname | 一维数组 | 获得个人信息数组 | ①使用myDB类封装好的select方法访问数据库，select函数返回数组。 |
| getCourseNameRow  获得课程名称数组 | uname | 一维数组 | 获得已学的课程数组 | ①使用myDB类封装好的select方法访问数据库，select函数返回数组。 |
| getLearnTimeRow  获得学习时间数组 | uname | 一维数组 | 获得近7天的在线时间数组 | ①使用myDB类封装好的select方法访问数据库，select函数返回数组。 |

#### 3.3.2.6鼠标悬停和鼠标移出

##### 3.3.2.6.1 public.js

|  |  |  |
| --- | --- | --- |
| 属性名字 | 类型 | 描述 |
|  |  |  |

|  |  |  |
| --- | --- | --- |
| 方法 | 参数 | 用途 |
| mouseOn() | 无 | post空信息到onMouse.php |
| mouseOut() | 无 | post空信息到offMouse.php |

##### 3.3.2.6.2 onMouse.php

|  |  |  |
| --- | --- | --- |
| 名字 | 类型 | 描述 |
| flag1 | 布尔 | flag1=is\_set(‘totalTime’)，flag1为假时，此次鼠标悬停事件为第一次鼠标悬停 |
| flag2 | 布尔 | flag2=is\_set(‘userName’)，flag2为真时，用户状态为登陆后 |
| now | 字符串 | 当前时间 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **方法内部功能** | 参数 | 返回类型 | 用途 | 概要步骤 |
| 判断是否第一次悬停 | ‘totalTime’  ‘userName’ | 布尔 | 判断此次onMouse事件是否为用户登陆后第一次悬停事件，如果是，则需要将totalTime设置为0，放入session中 | ①使用Session类的is\_set()判断此次鼠标悬停事件为第一次鼠标悬停。  ②使用Session类的is\_set()判断用户状态为登陆后 |
| 获取当前时间 | format,  time() | 字符串 | 获取当前时间，用于下一次的鼠标移出事件，以获取此阶段的专注时长 | 使用Date(format,time()) |
| 设置专注时长为0，并存入session | ‘totalTime’ | 无 | 如果此次鼠标悬停事件是用户登陆后的第一次悬停事件，则需将totalTime置为0，并存入session中 | 使用Session类的set() |
| 将当前时间存入session | ‘onMouseTme’ | 无 | 将当前时间存入session中，当下一次的鼠标移出事件发生时，可以此获取此阶段的专注时长 | 使用Session类的set() |

3.3.2.6.3 offMouse.php

|  |  |  |  |
| --- | --- | --- | --- |
| 名字 | 类型 | 描述 | |
| totalTime | 字符串 | flag1=is\_set(‘totalTime’)，flag1为假时，此次鼠标悬停事件为第一次鼠标悬停 | |
| now | 字符串 | 当前时间 | |
|  |  | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **方法内部功能** | 参数 | 返回类型 | 用途 | 概要步骤 |
| 获取当前时间 | ①format  ②time() | 字符串 | 获取当前时间，用于下一次的鼠标移出事件，以获取此阶段的专注时长 | 使用Date(format,time()) |
| 获取目前的专注时长 | ‘totalTime’ | 字符串 | 从session中获取此前的专注总时长 | 使用Session类的get() |
| 获取上一次鼠标悬停事件的时间 | ‘onMouseTime’ | 字符串 | 从session中获取上一次鼠标悬停事件的时间 | 使用Session类的get() |
| 计算新的专注总时长 | ①now  ②totalTime  ③onmouseTime | 字符串 | 更新专注总时长 | 通过上次悬停事件与当前时间的差可得此阶段的专注时长，加入到专注总时长中，可获得新的专注时长 |
| 将新的专注总时长存入session | ①totalTime  ②’totalTime’ | 无 | 在session中更新专注总时长 | 使用Session类的set() |

#### 3.3.2.7用例公用类：

##### 3.3.2.7.1 myDB

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 方法名称 | 参数 | 返回类型 | 用途 | 概要步骤 |
| Select | ①sql  ②type  ③responseStatus | 字符串或数组 | 连接数据库，执行sql语句，获得sql语句的结果，并返回。 | ①连接数据库：mysqli[ root, user, pwd, responseStatus ]  ②根据type判断此次sql操作是否需要返回信息。  ③如果需要返回数组，则调用query(sql)  和fetch\_row(),并返回数组  ④如果不需要返回数组，则调用query(sql)，并根据query结果设置responseStatus的值，并返回responseStatus |

##### 3.3.2.7.2 Session类

Session::set(name, value)：注册 session 。

Session::is\_set(name)：检查Session的值是否设置。

Session::get(name)：读取 session 。

Session::clear()：清空Session。

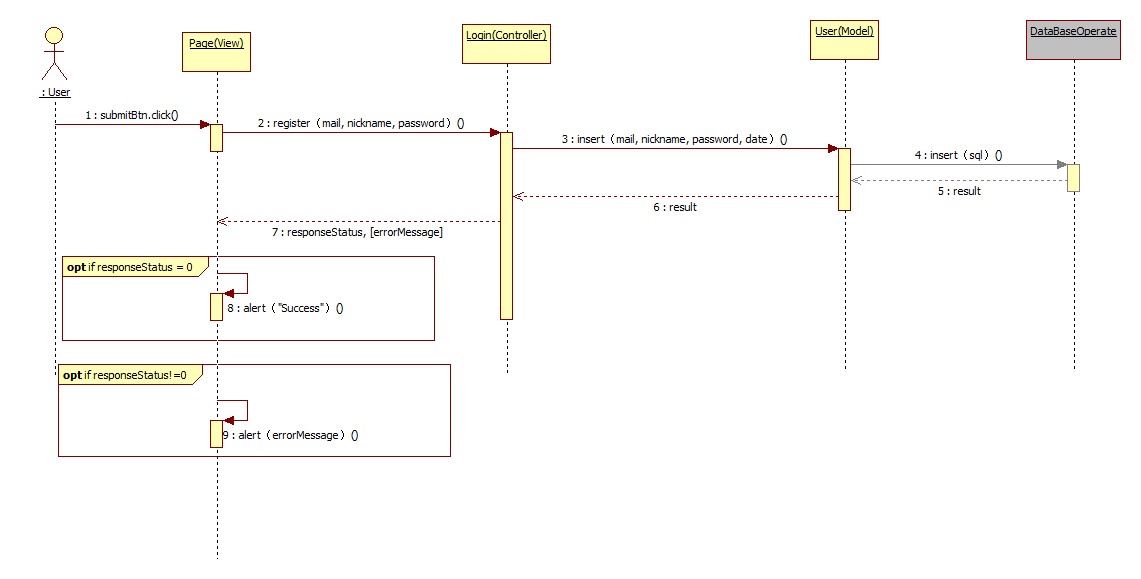
Session::destroy()：销毁 session 。

## 3.4系统功能实现详细设计

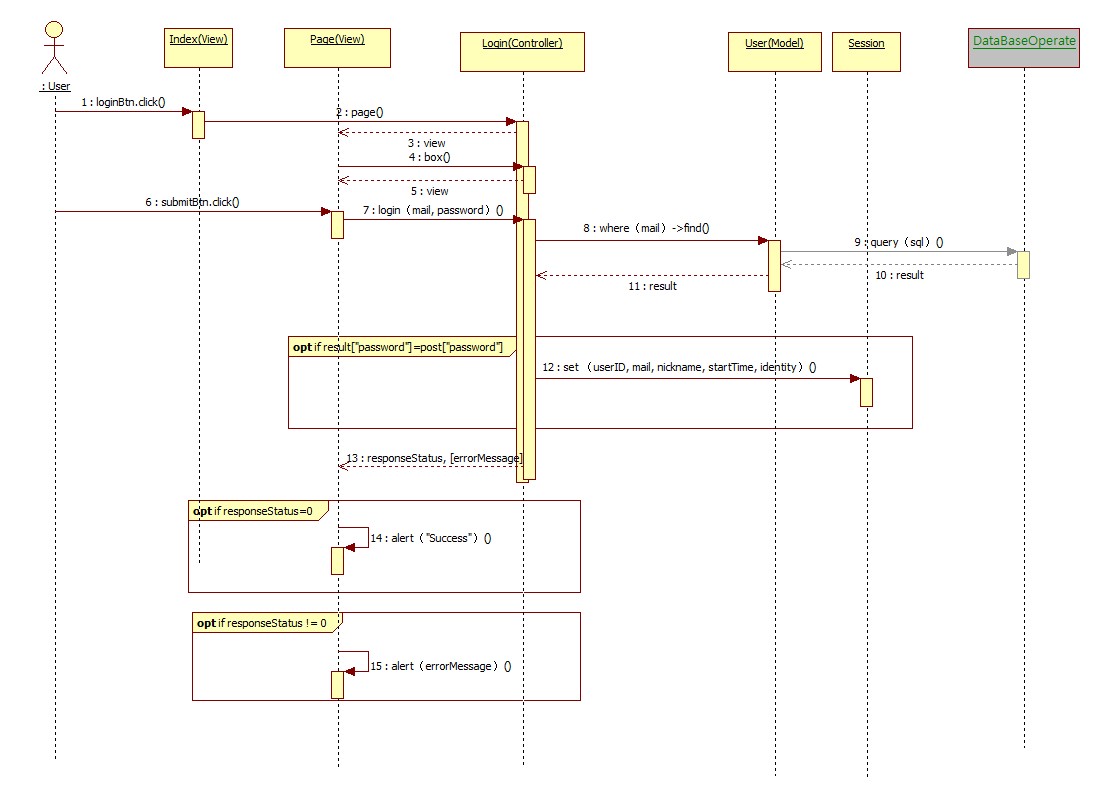
### 3.4.1系统状态管理



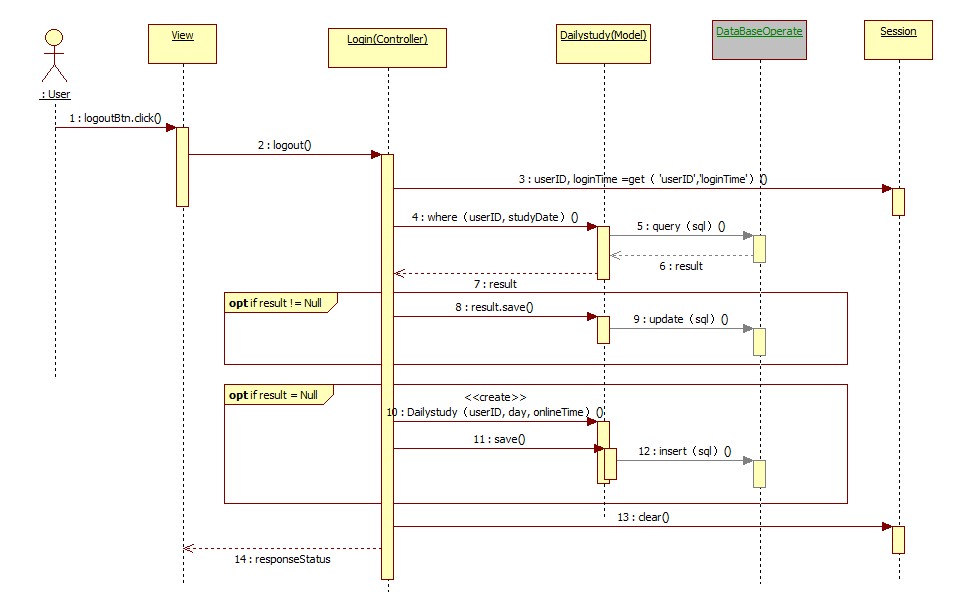
### 3.4.2注册



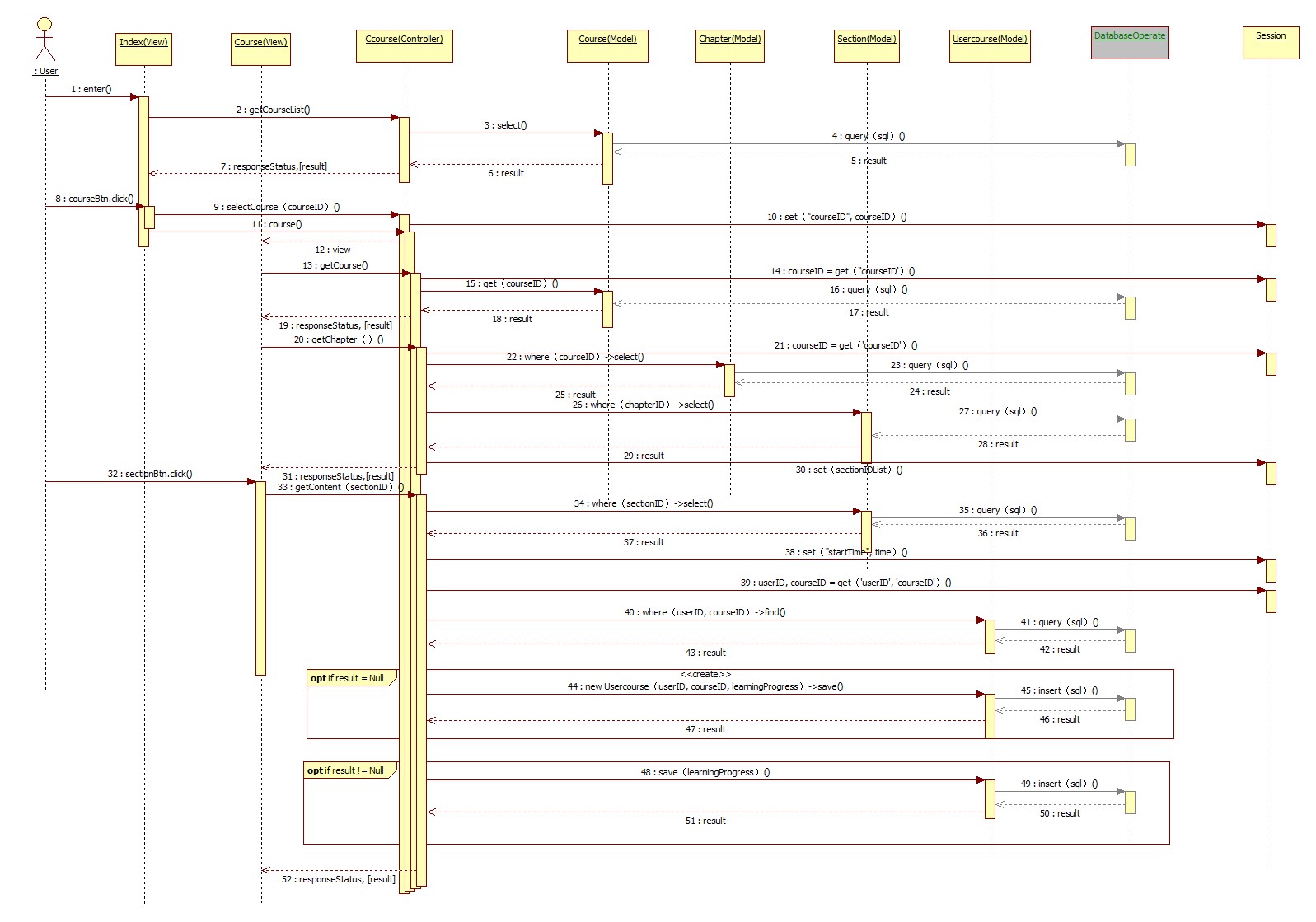
### 3.4.3登录



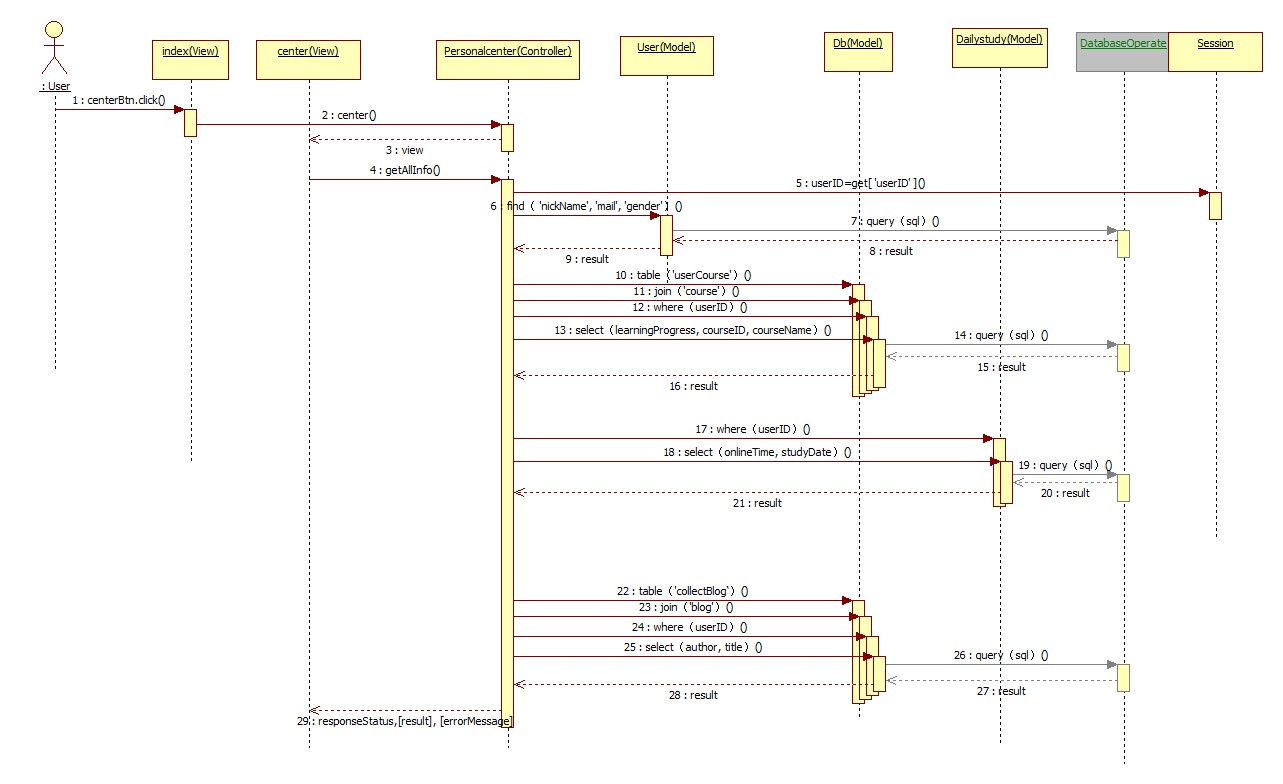
### 3.4.4注销



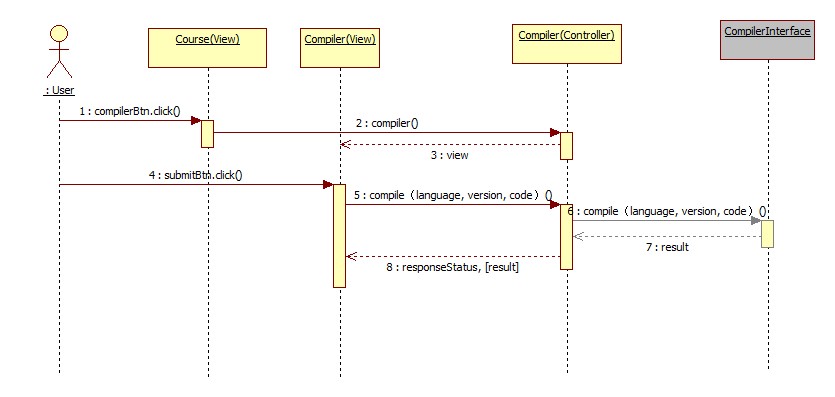
### 3.4.5查看课程内容



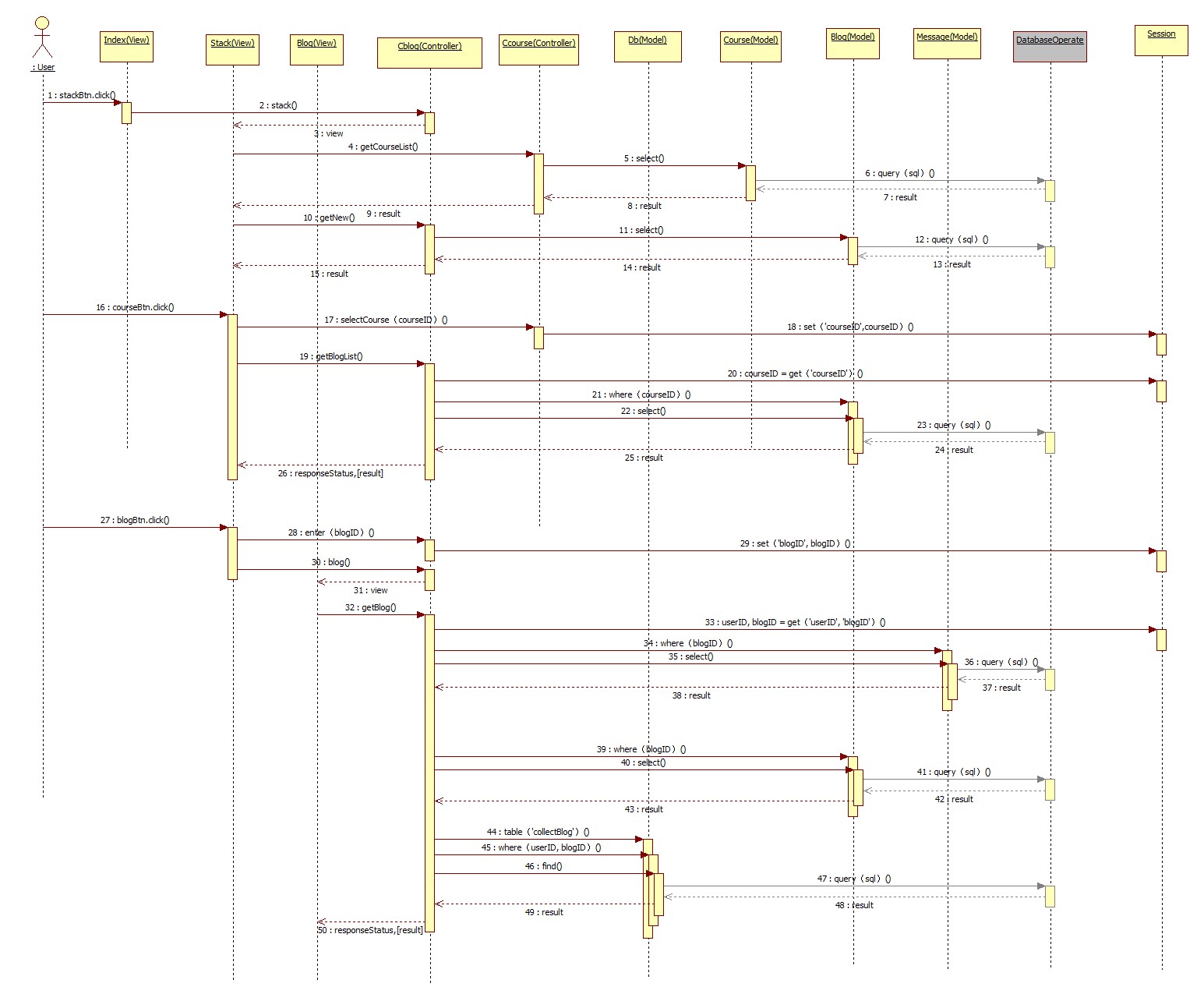
### 3.4.6查看个人中心



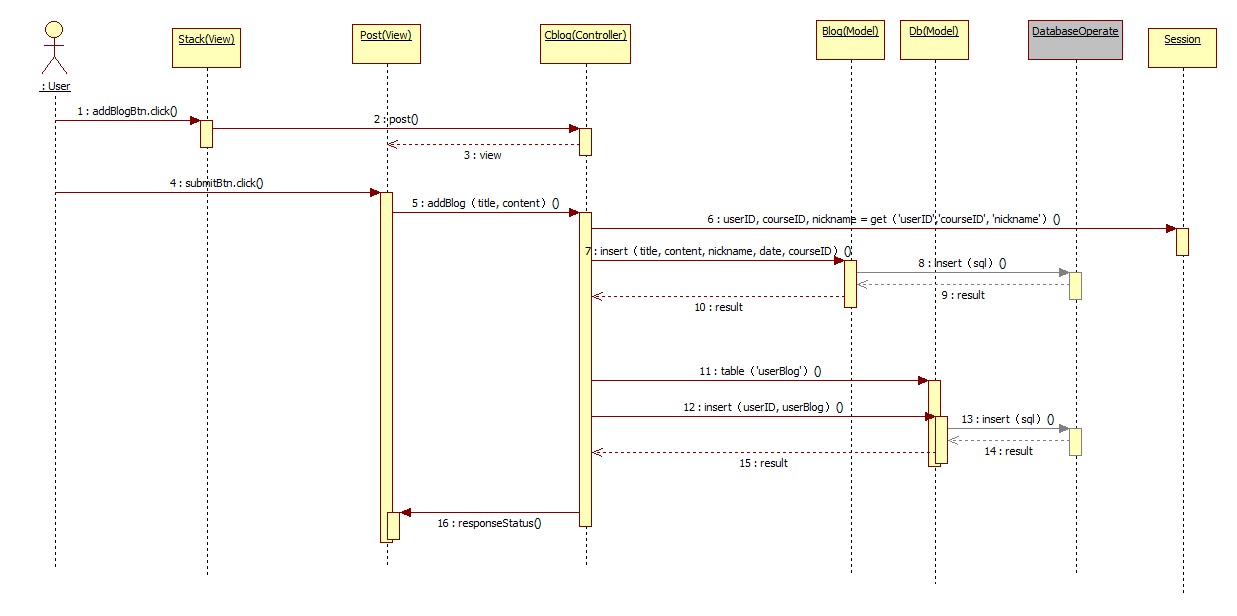
### 3.4.7在线编译



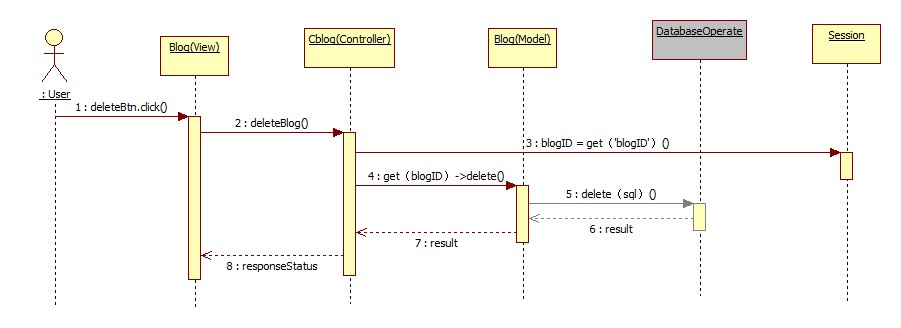
### 3.4.8查看帖子



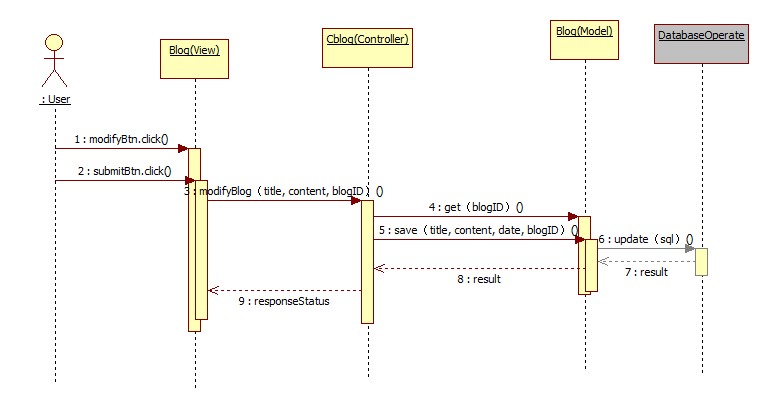
### 3.4.9发布帖子



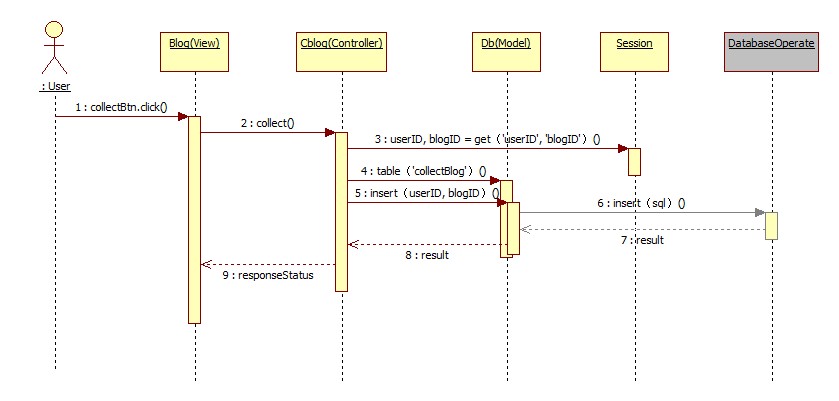
### 3.4.10删除帖子



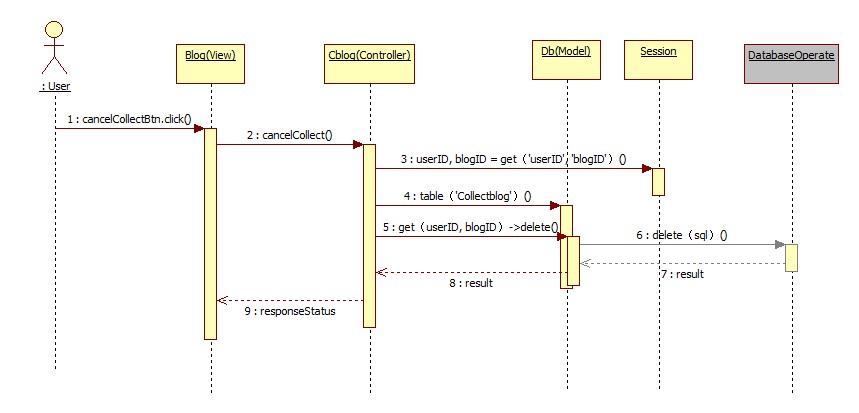
### 3.4.11修改帖子



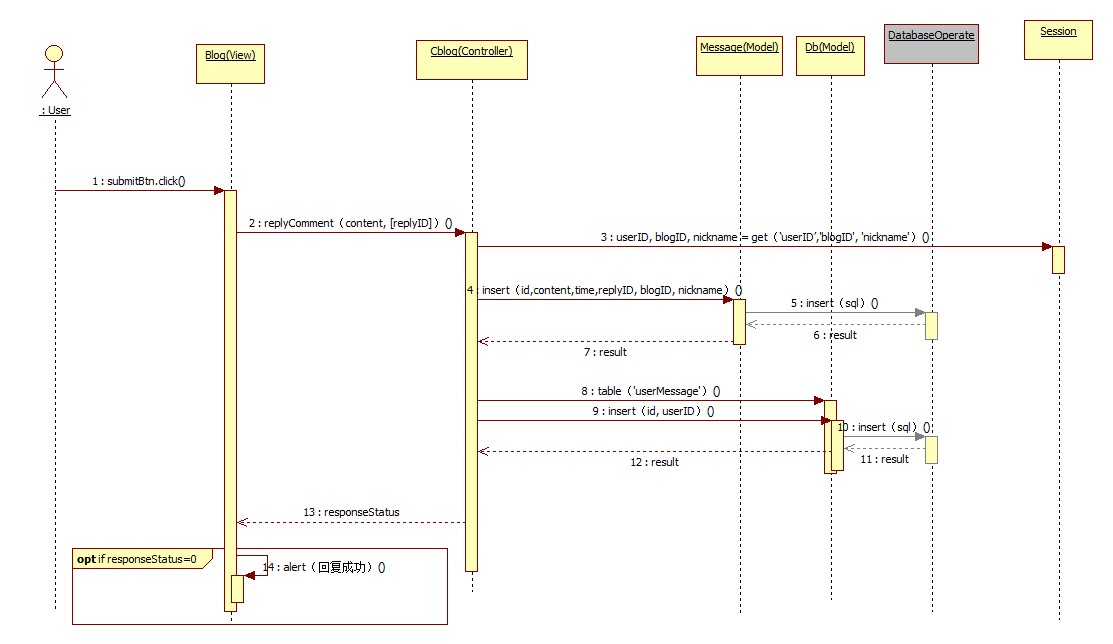
### 3.4.12收藏帖子



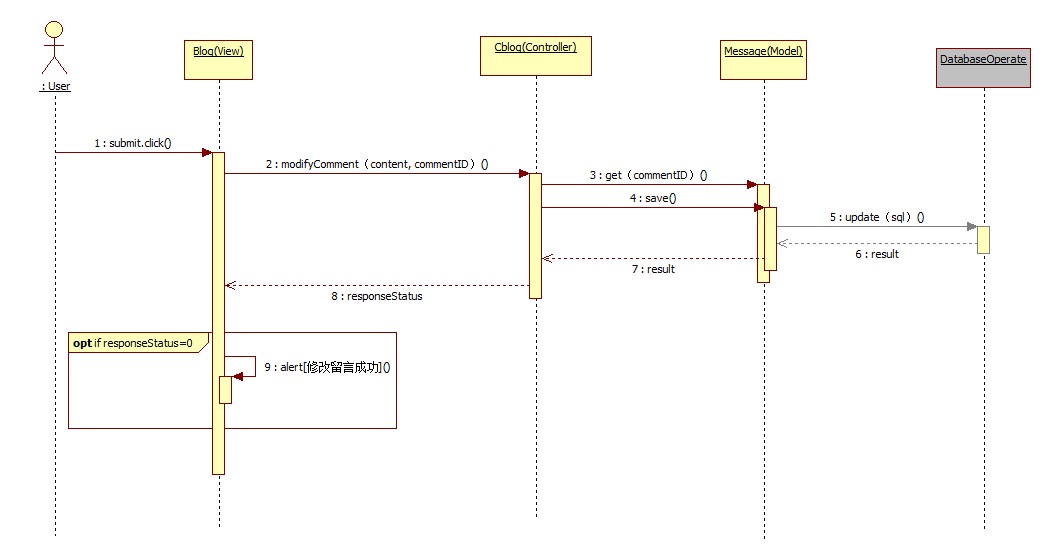
### 3.4.13取消收藏



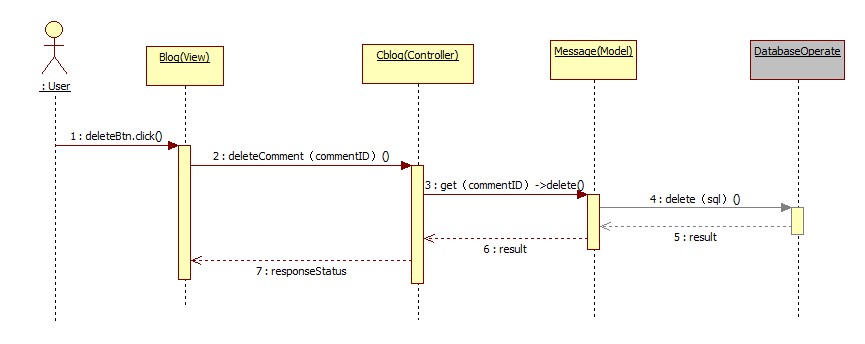
### 3.4.14发布留言



### 3.4.15修改留言



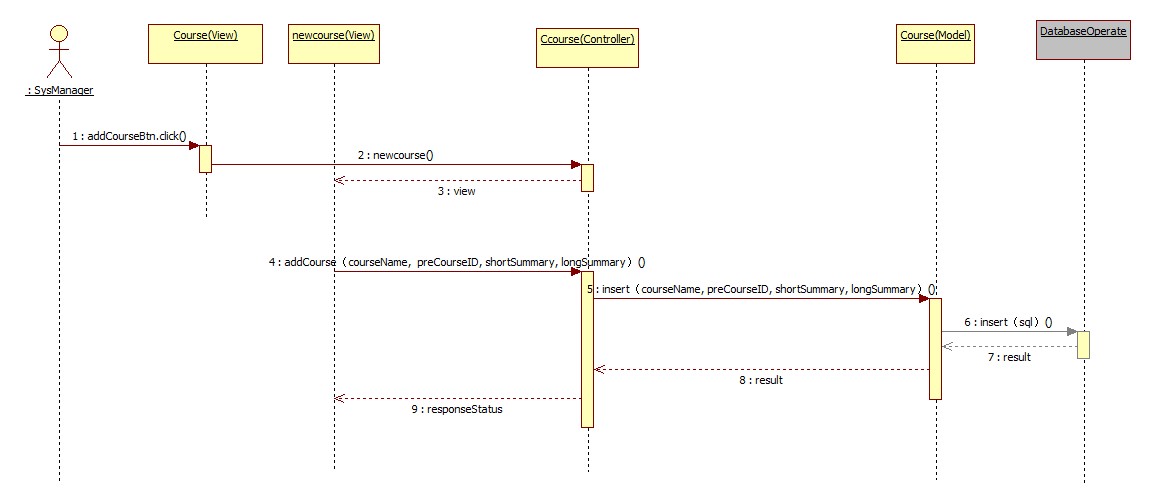
### 3.4.16删除留言



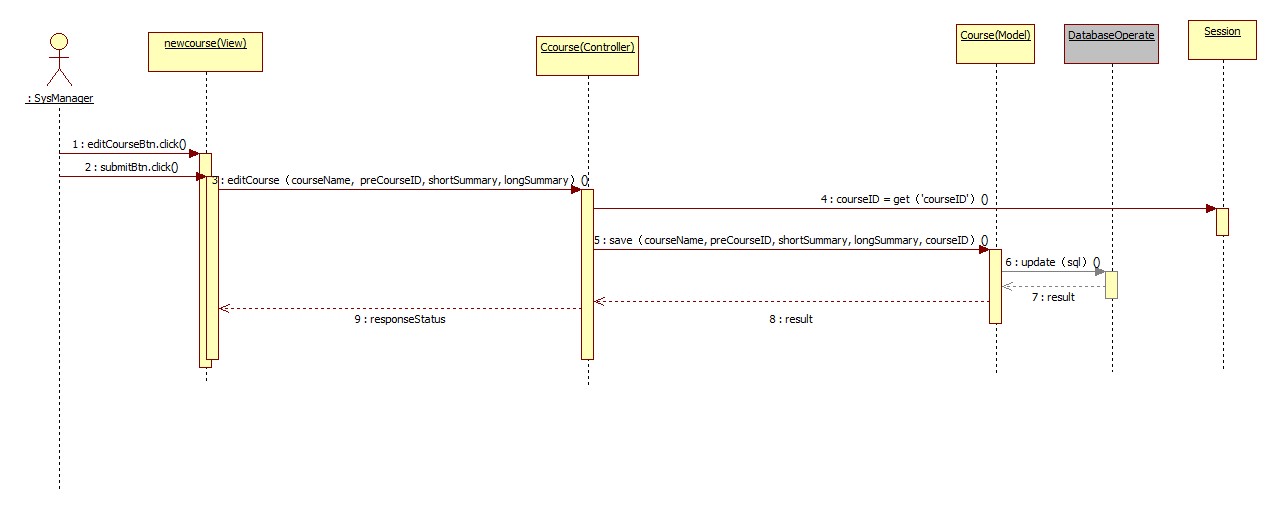
### 3.4.17课程管理



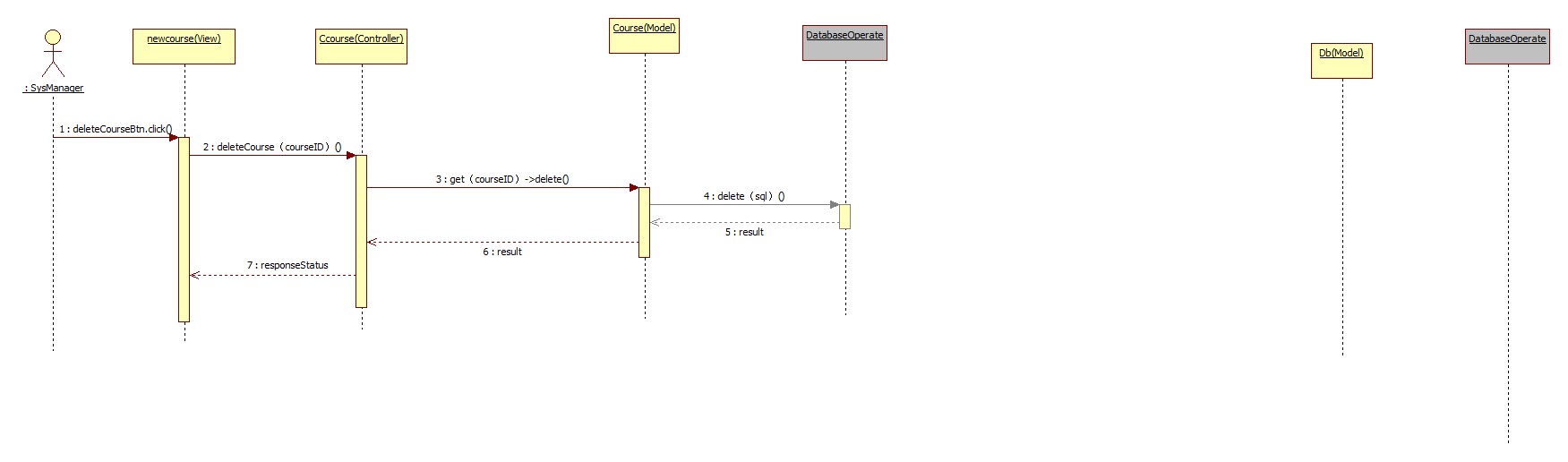
### 3.4.18添加课程



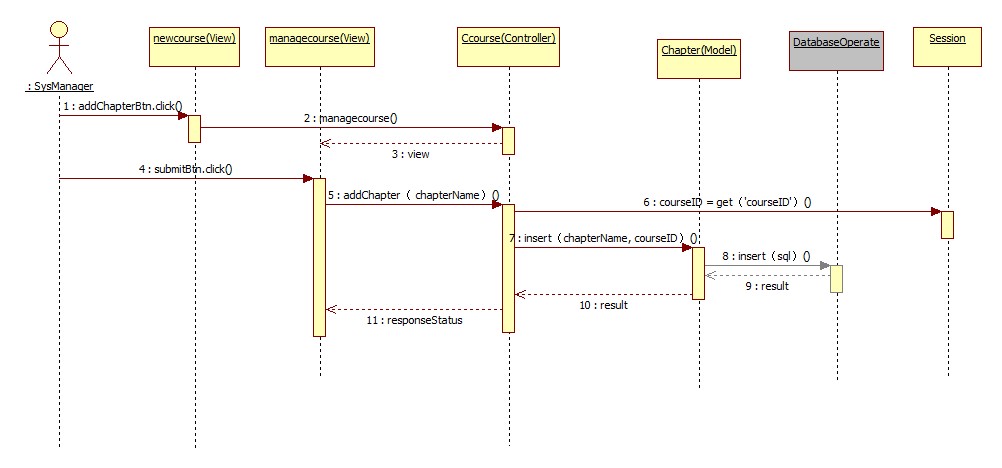
### 3.4.19修改课程



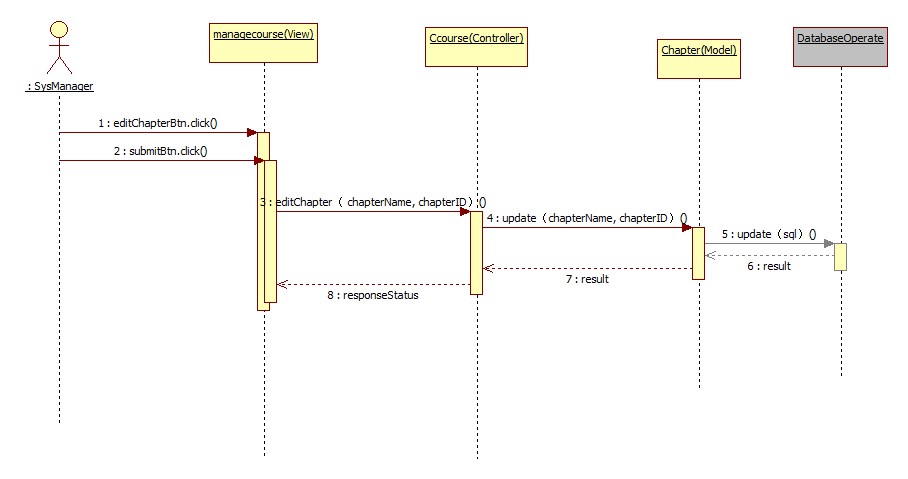
### 3.4.20删除课程



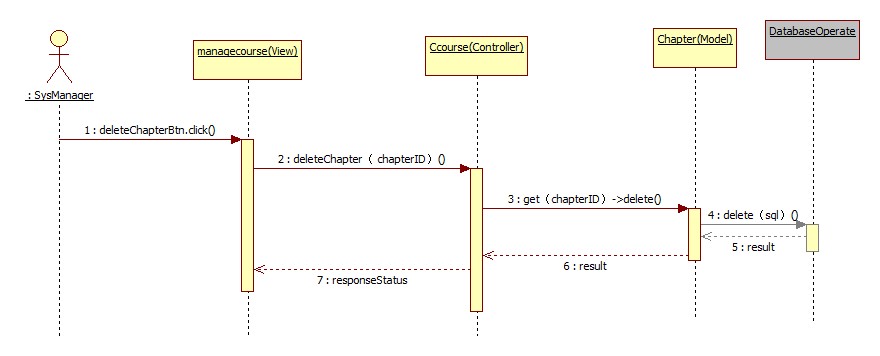
### 3.4.21添加章节



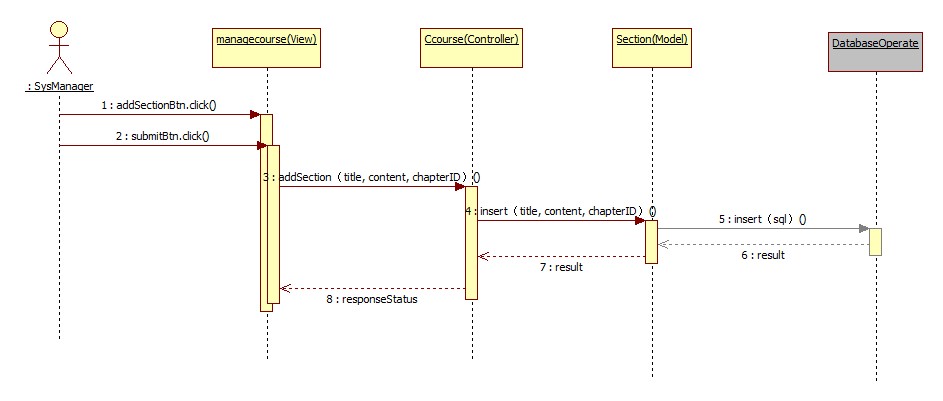
### 3.4.22修改章节



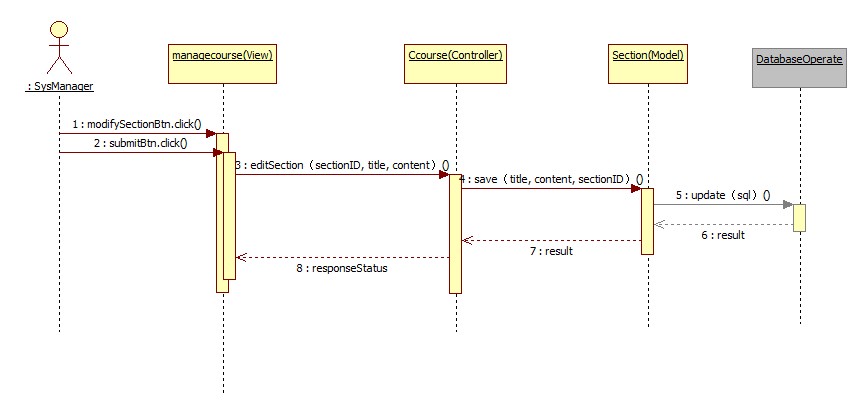
### 3.4.23删除章节



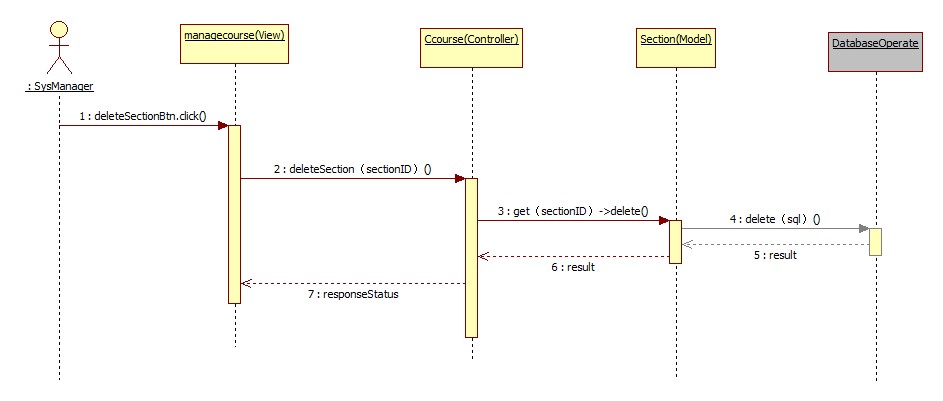
### 3.4.24添加小节



### 3.4.25修改小节



### 3.4.26删除小节



## 3.5系统非功能设计

### 3.5.1性能设计

1. 在腾讯云云服务器上租一台服务器(1核2G 50GB云磁盘 1MPs带宽以上) ，搭建LAMP环境（Linux + Apache + MySQL + PHP），依靠云服务保证网站服务器性能方面达到最大保障。
2. 可以增加带宽大小，保证多用户并发访问，防止系统因访问量过大响应迟缓。
3. 根据需求租用更高性能的服务器，提升用户访问体验。

### 3.5.2安全性设计

1. 服务器的托管选择在大的云服务器商处，以减少被攻击的可能性。

2. 服务器上系统和软件的版本尽可能选择稳定且新的版本，以减少软件或系统漏洞，关闭服务器不用的端口，及时更新安全补丁。

3. 后台开发时加强进行安全性测试，包括密码加密、使用验证码等。

4. 常做数据备份，防止数据丢失。

5. 管理员本身的密码尽可能复杂，并且通常访问数据库或后台的时候尽可能使用低权限的账户。

### 3.5.3可靠性设计

1. 在云服务部署服务器，依靠腾讯云强大的服务器支持，可以保证系统7x24小时运行，全年持续运行故障停运时间累计不超过10小时。
2. 系统管理员不定期检查系统，当系统出现问题时及时维护。
3. 设置备份服务器，避免系统因服务器问题全面瘫痪。

### 3.5.4数据保密设计

1. 监控数据库服务器的所有输出端口，控制其配置和信息，防止服务器被外部入侵。

2. 对传输的数据进行加密，防止数据在传输过程中被截取后泄露。

3. 严格把控数据库及后台管理员的账户及密码，定期更换复杂密码，防止数据库被内部破解。

### 3.5.5可用性设计

1. 所有数据，包括用户数据，课程数据，留言信息数据，全部对应建立历史备份数据库，定期将当前数据库中的数据追加到历史数据库中。对历史数据库中的数据，定期备份，一般当前数据，每月一次自动复制到历史数据库中，历史数据库保留半年后，使用光盘刻录设备，进行2份以上备份，以后清除历史数据库这部分数据，以提高系统运行效率，释放硬件资源。
2. 主数据库服务器使用磁带备份系统，对数据库中的实时数据更新备份，可以采用每日，每周，每月系统自动备份到磁带机上，用跟踪带对系统的操作进行记录。
3. 如果当前数据库遭到破坏，不能保证完整性，可以考虑采用一下方法进行恢复：

①将历史数据库中的数据追加到数据库中；

②将磁带上的数据按月备份，周备份，日备份的顺序依次恢复到数据库中；

③还可以用光盘的数据导入数据库中。