

# 计算机科学系实验报告

课程名称	<u>UML 与可视化建模</u>	班级	<u>14 网一</u>		
实验名称	<u>课程管理系统</u>	指导教师	<u>曾少宁</u>		
姓名	<u>马明健</u>	学号	<u>1414080903109</u>	日期	<u>2017.03.03</u>

---

## 一、实验目的

掌握基于 UML 2.0 的建模概念与方法，掌握各种 UML 图的概念与画法，其中包括用例图、活动图、类图、顺序图、组件图和状态图等。

## 二、实验设备与环境

操作系统：Windows 7；建模工具：StarUML。

## 四、实验要求

1. 实验及实验报告以增量方式完成，每次作业都在上一次作业的基础上完成，作业提交网站不提供报告下载，所以请同学们自行保管好自己的实验报告；
2. 请将实验报告中“占位符”信息替换为自己的实验相关信息；
3. 请认真撰写实验体会，**实验课结束时**立即上传实验报告：<http://zeng.shaoning.net/uml/>。

## 四、实验内容、程序清单及运行结果

课程管理系统

### 1、实验一：需求建模 - 用例模型

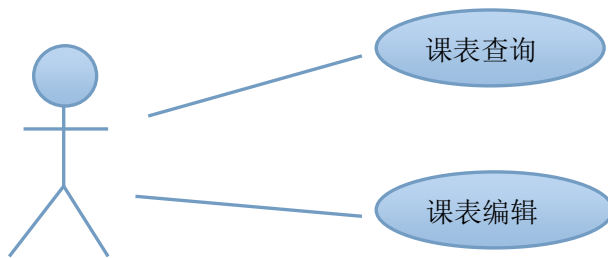


图 1: ?? 用例图

注：用例规约内容及项目可自行增加。

用例编号:	UC001
用例名称:	课表查询
用例描述:	用户可以通过该系统查询自己的课表
前置条件:	打开课程管理系统, 用户输入自己的学号进行查询
基本流程:	<ol style="list-style-type: none"> <li>1. 系统显示输入学生学号界面。</li> <li>2. 用户输入自己的学号, 点击查询。</li> <li>3. 根据用户输入的学号, 识别出该用户的年级, 系别, 专业及班级, 通过这些信息在数据库中检索出该用户的课程表, 并显示在界面上。</li> <li>4. 录入学号信息之后, 界面上会多出日期课表查询的选项。</li> <li>5. 用户点击日期查询, 输入某个日期, 点击查询。</li> <li>6. 系统会根据用户输入的日期, 检索该用户当天的课程, 并将当天的课程显示在用户界面上。</li> </ol>
扩展流程:	<ol style="list-style-type: none"> <li>2.1 如果学号输入有误, 则显示“学号有误, 请重新输入学号。”学号正确的话, 则识别该用户的年级, 系别, 专业, 班级等信息。</li> <li>6.1 如果当天有课程的话, 便将课程显示在界面上。当天没有课程的话, 则显示“当天无课程”。</li> </ol>
后置条件:	

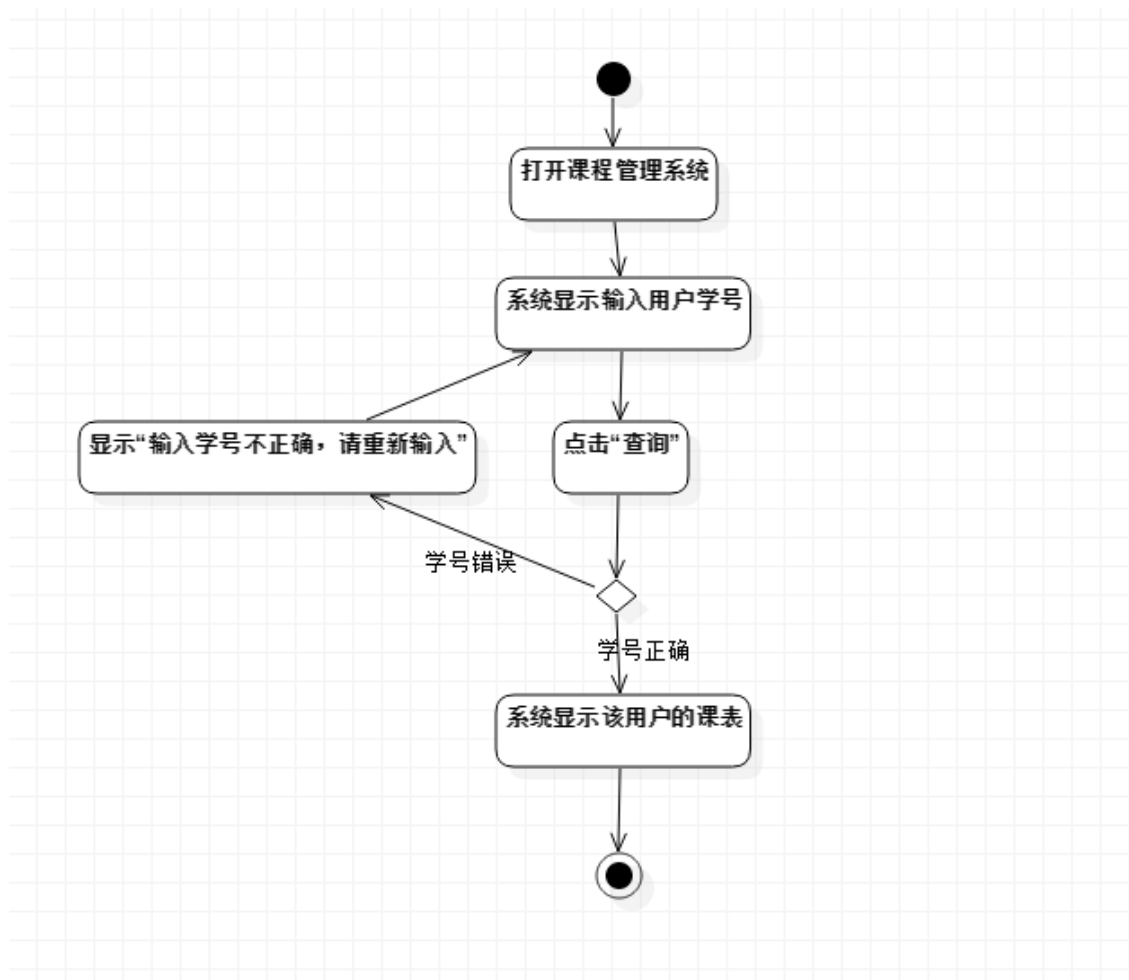
用例编号:	UC002
用例名称:	课表编辑
用例描述:	当课程发生变化时, 用户可以通过该系统来编辑自己的课表
前置条件:	打开课程管理系统, 用户录入自己的学号信息
基本流程:	<ol style="list-style-type: none"> <li>1. 系统界面显示“课表查询”, 点击课表查询。</li> <li>2. 该用户的课表会显示在系统界面上。</li> <li>3. 点击任意课程, 会出现“删除”“移动”“修改信息”“增减课时”这几个选项。</li> <li>4. 用户根据自己的需要, 对课表进行编辑。</li> <li>5. 编辑完成之后, 点击保存。</li> </ol>

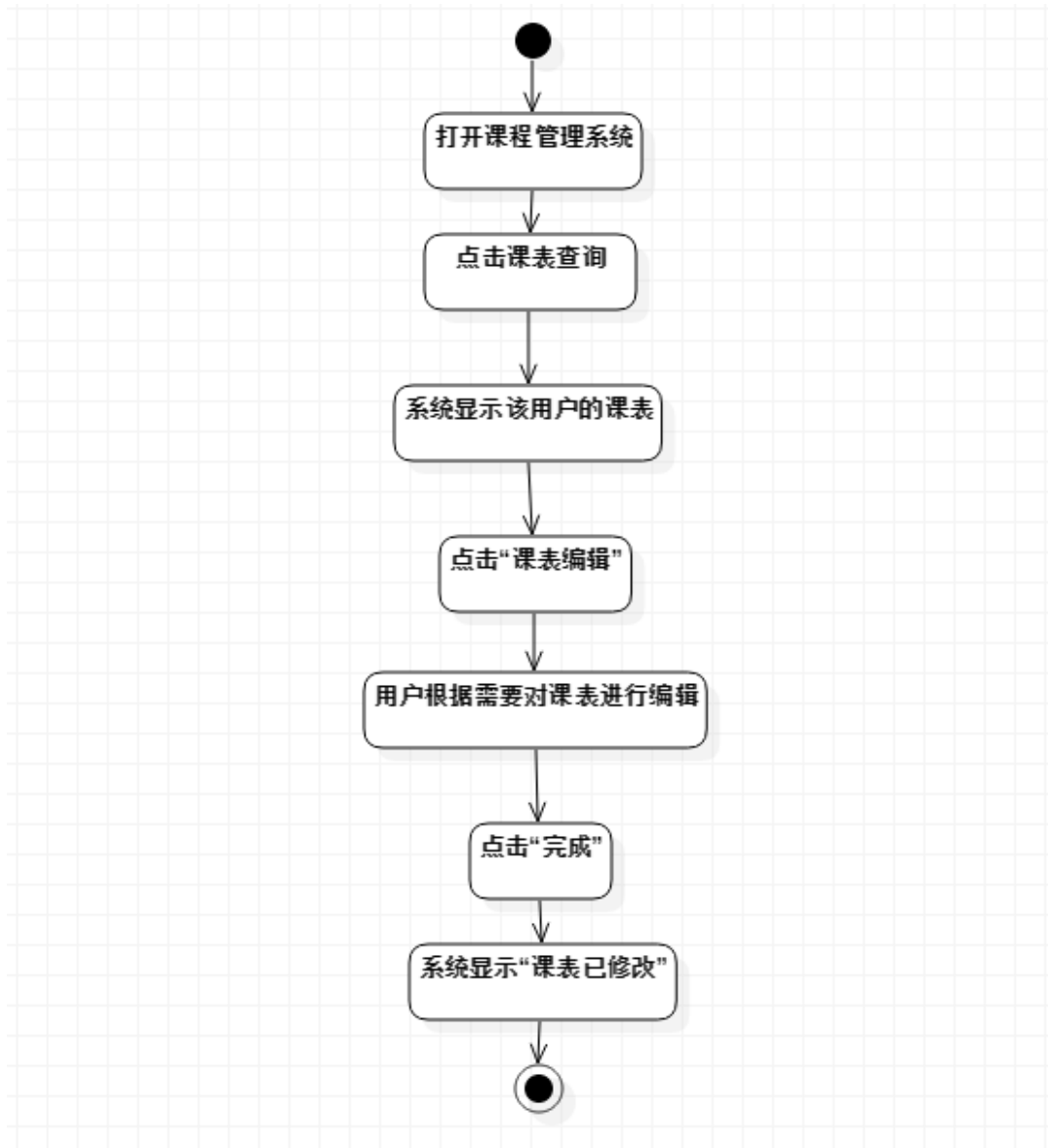
	<p>6. 界面显示“课表修改中。。”，同时系统将该用户原来数据库中的课表信息删除，并上传新的课表信息到数据库中。</p> <p>7. 界面显示“课表已完成修改”。</p>
扩展流程:	<p>3.1 如果该用户将某门课程退选之后，可点击删除，将该课程在课表上删除。</p> <p>3.2 如果该用户的某门课程发生时间变化，可点击移动，将该课程移动到另外的空白时间点上。</p> <p>3.3 如果该用户某门课程的授课老师、上课地点发生变化，可点击修改信息，将原有的课程信息进行更正。</p> <p>3.4 如果该用户某门课程的课时发生变化，可点击增减课时，得出正确的课表。</p>
后置条件:	

## 2、实验二：过程建模 – 活动模型

使用活动图描述系统的业务过程。

方法：将用例规约中的基本流程与扩展流程抽象为过程步骤（Action），画出对应的活动图。



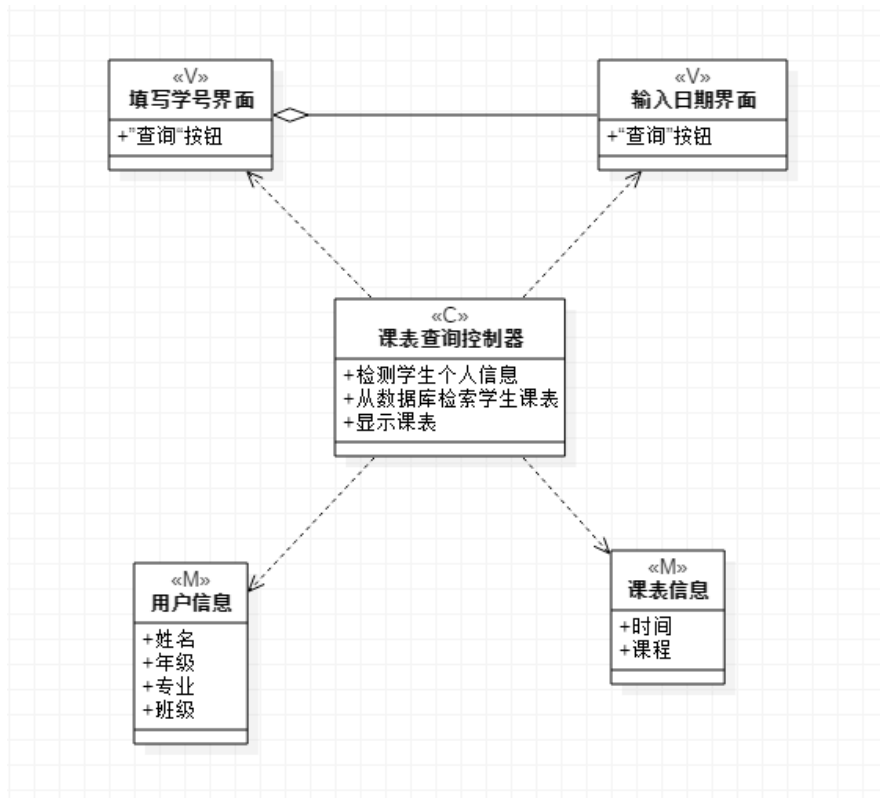


### 3、实验三：逻辑建模 – 类模型

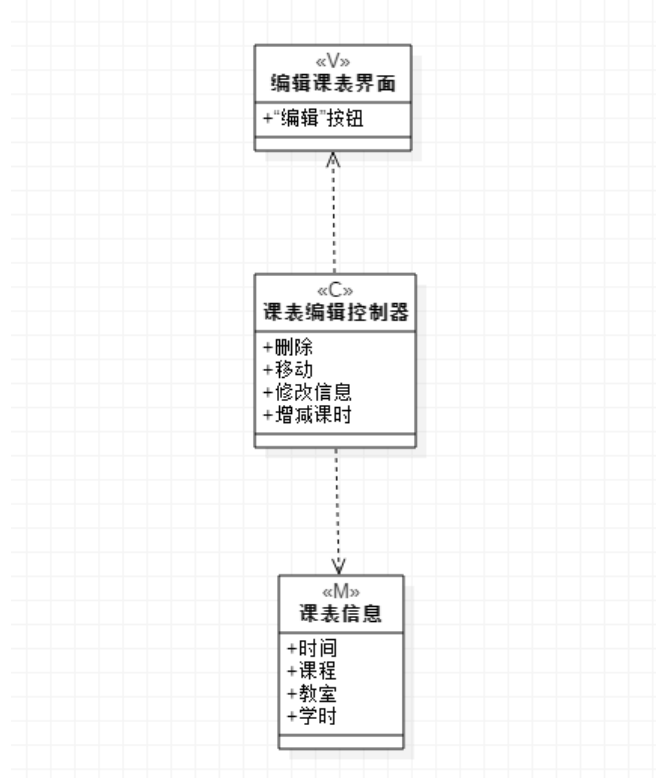
基于 MVC 设计模式找出实现用例的类。

方法：分别找出实现用例的模型（Model）、视图（View）和控制器（Controller）类，确定类之间的关系及其关键属性，画出类图。

## 课表查询类图



## 课表编辑



## 4、实验四：交互建模 – 顺序模型

创建各个类（MVC 及 Actor）的对象，并描述对象之间的交互。

方法：分别创建参与者（Actor）、界面类（View）、控制器类（Controller）和模型类（Model）的对象，描述各个对象之间的消息及其顺序，画出顺序图。

参考：讲义 P33 页 8.7.2。

## 5、实验五：状态建模 – 状态模型

对系统中最重要的对象进行状态建模。

方法：选择一种对象，定义该对象的状态，描述状态之间的切换及条件，画出状态图。

参考：讲义 P9 和 P10 页。

## 五、实验体会

实验一：

实验二：

实验三：

实验四：

实验五：