

计算机科学系实验报告

课程名称	UML 与可视化建模		班级	14 计科 1 班	
实验名称	日程管理系统		指导教师	曾少宁	
姓名	施丽珊	学号	1414080901126	日期	2017.6.2

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

日程管理系统

用例：1. 创建日程

2. 查看日程

• 实验一：需求建模 - 用例模型

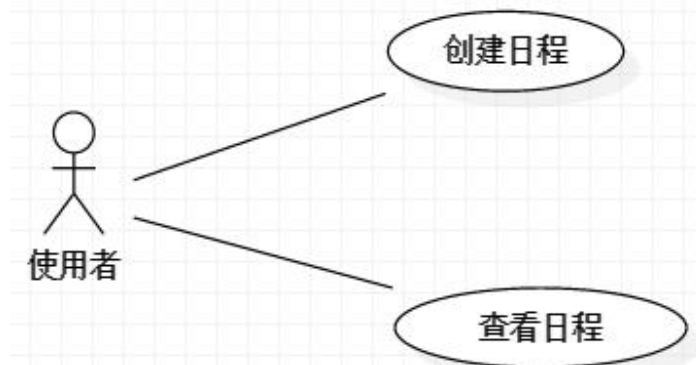


图 1：日程管理系统用例图

注：用例规约内容及项目可自行增加。

用例编号：	UC001
用例名称：	创建日程

用例描述:	
前置条件:	使用者已登陆账号
基本流程:	<ol style="list-style-type: none"> 1. 使用者点击“创建”按钮。 2. 系统显示“空日程”界面。 3. 使用者填写日程信息：时间（格式要求：2017年3月17日16:00）、事件描述、提醒时间（提前半个小时或自定义）、提醒方式（铃声提醒、震动提醒、不需要提醒），点击“提交”按钮。 4. 系统检查日程信息完整，将日程信息保存到数据库，显示“成功创建”信息。
扩展流程:	4.1 系统检查发现日程信息不完整，显示“所填写的日程信息不完整”信息。
后置条件:	

用例编号:	UC002
用例名称:	查看日程
用例描述:	
前置条件:	使用者已登陆账号
基本流程:	<ol style="list-style-type: none"> 1. 使用者点击“显示”按钮。 2. 系统成功查询到所有的日程信息，显示“所有的日程信息记录”界面。 3. 使用者点击某个日程信息。 4. 系统显示使用者选择的“日程信息”详细界面。
扩展流程:	2.1 系统检查发现使用者没有日程安排，提示“该用户日程为空，请创建日程”信息。
后置条件:	

• **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。

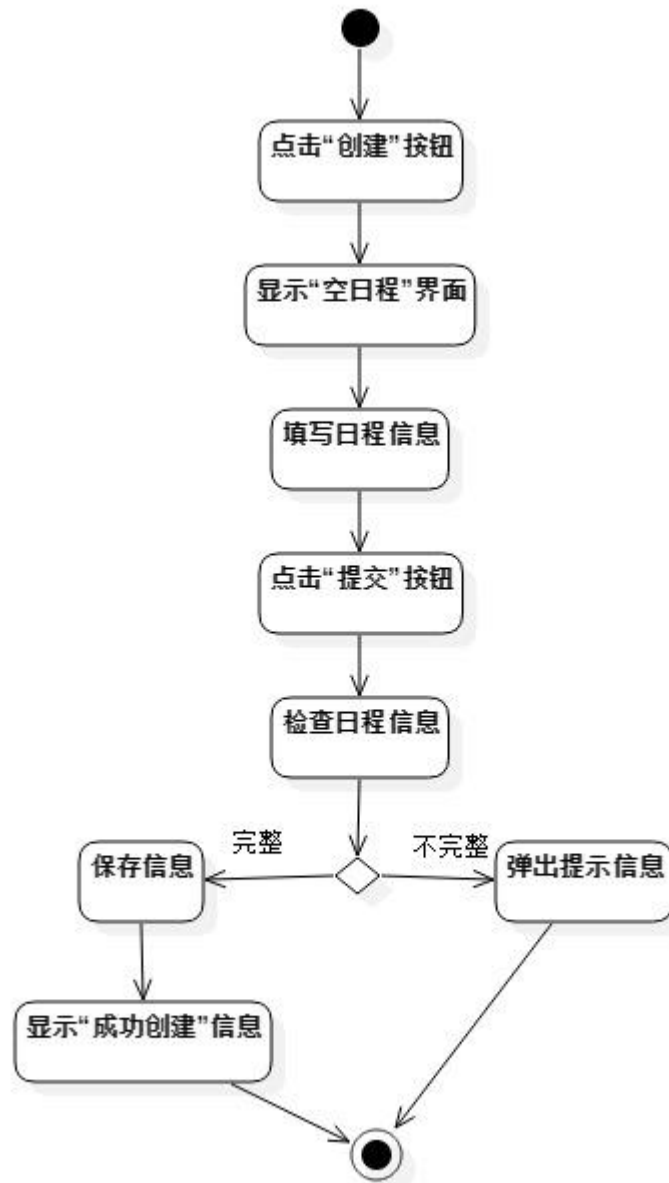


图 2-1：创建日程-活动图

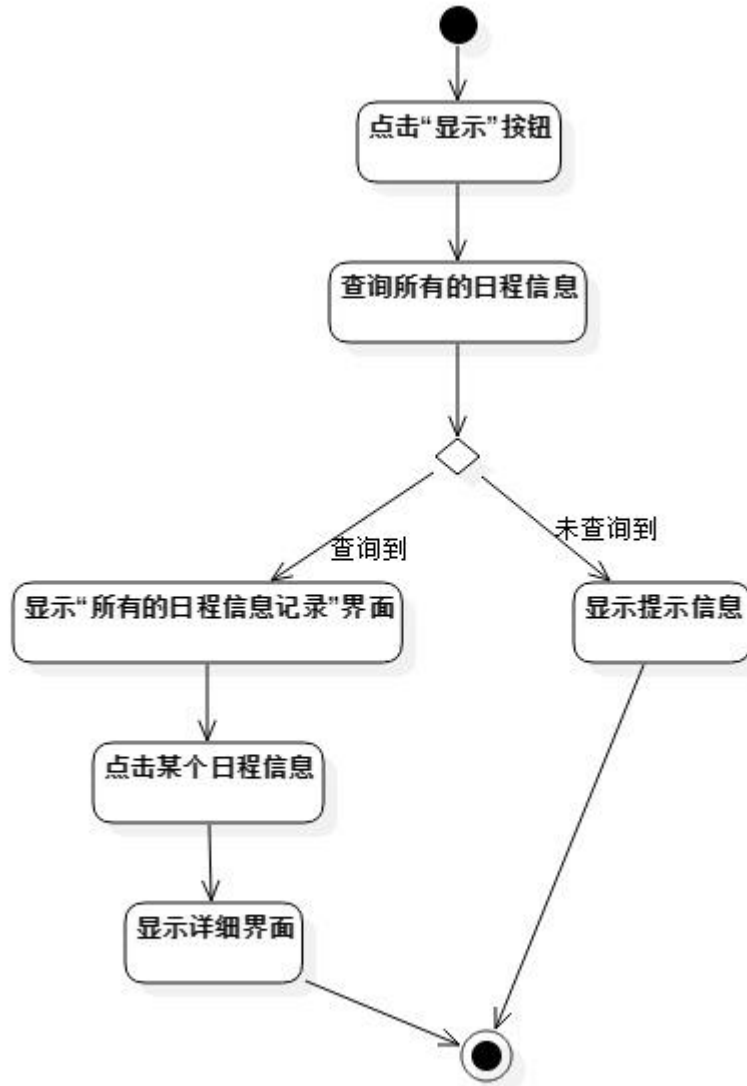


图 2-2：查看日程-活动图

• **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。

参考：讲义 P26 页。

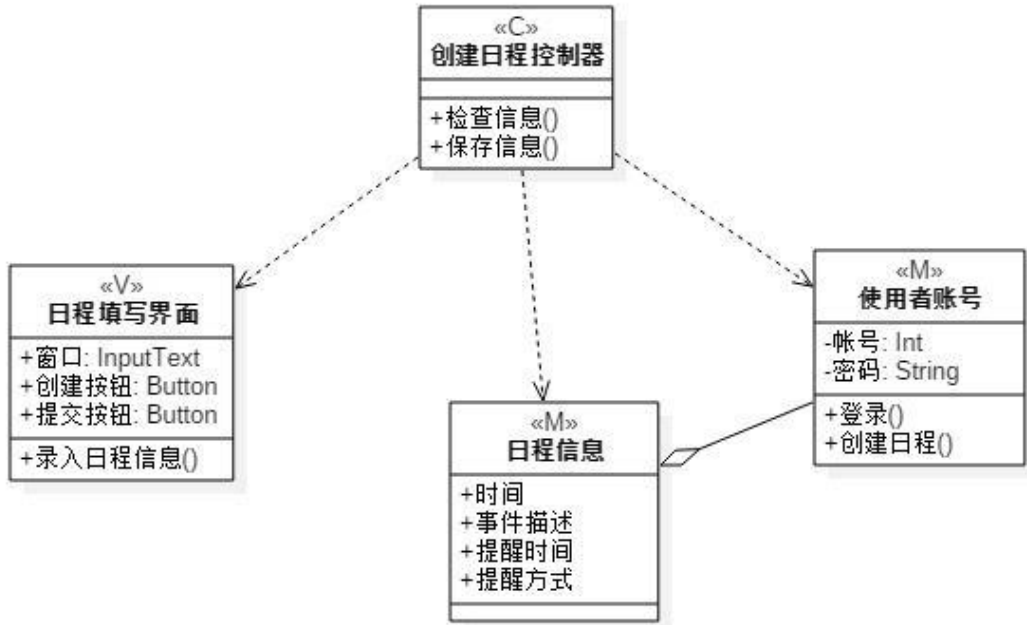


图 3-1: 创建日程-类图

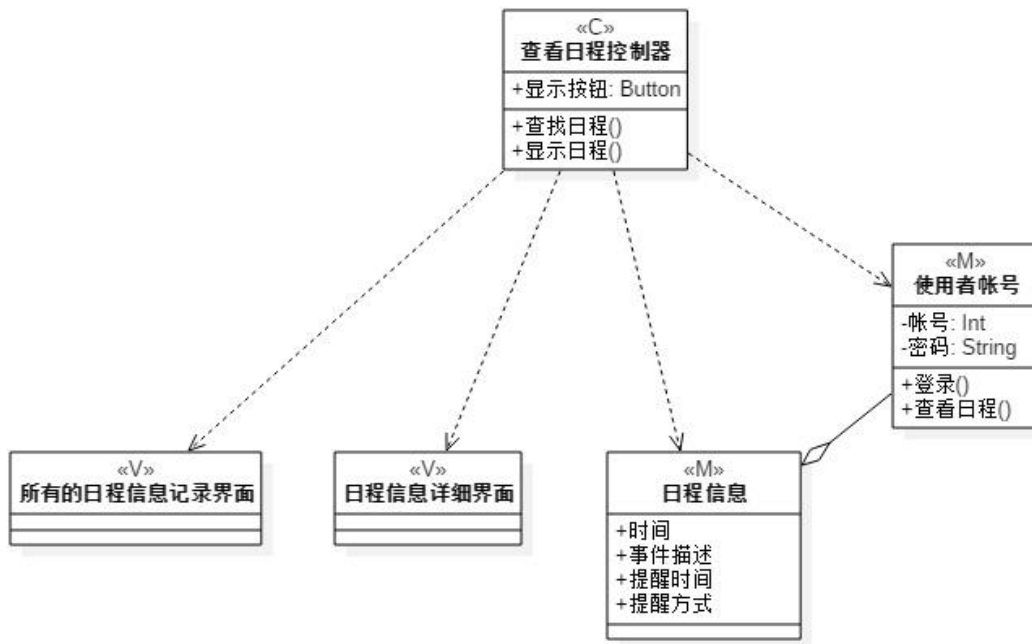


图 3-2: 查看日程-类图

• 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。

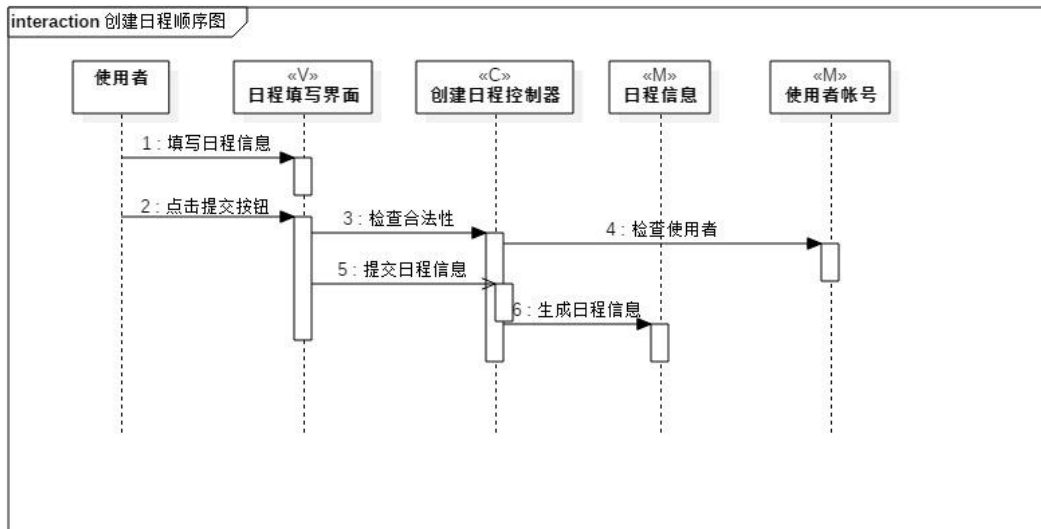


图 4-1：创建日程顺序图

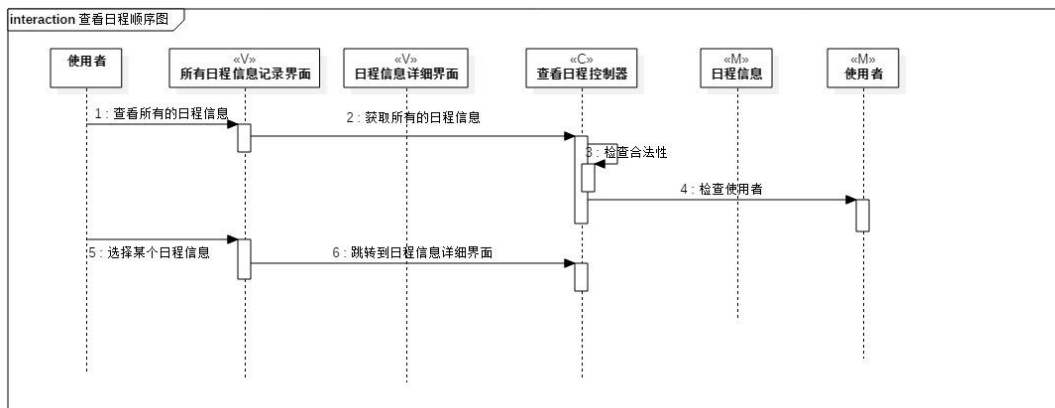


图 4-2：查看日程顺序图

• 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。

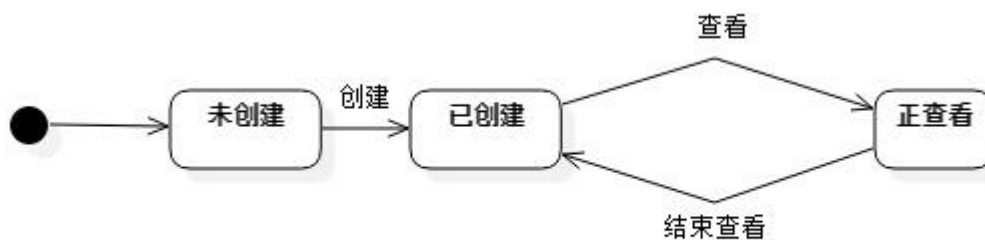


图 5-1：“日程信息”状态图

五、实验体会

实验一：

第一次实验是确定题目，由于一直想设计一个日程管理系统，不仅仅在于提醒别人日程，还能在提醒时候在锁屏上显示提示信息的时候添加一个背景图片，给予别人新鲜感和愉悦感。在设计用例时候很快根据需求得到三个用例，然后在下一次实验课时候，在确定用例规约内容时候，发现有点难以确定某些用例的流程，后面经过多次修改，改变了一开始的用例，确定为创建日程和查看日程，再去填写用例规约内容，并且使用了 StarUML 工具设计出日程管理系统用例图。

实验二：

在实验一下设计活动图，由用例规约内容的流程去确定活动图，活动图步骤与基本流程一致。设计过程中也是再一次的去分析实验一的用户规约内容是否有误，用例一创建日程中，第四步是系统检查发现日程信息是否完整，由于日程信息包含很多内容，其中某一条信息的不完整，系统会相应显示提示信息，然而再设计活动图时候，发现这样很累赘，最后直接设计为系统检查发现日程信息不完整，显示“所填写的日程信息不完整”信息，并同时又一次修改了用例规约内容。

实验三：

相对上两次实验，这次实验有点棘手，采用 MVC 模式构成整个系统的类，如果能够准确找到类，以及类的关系，将很快使用工具去设计出类图，然而在实验过程中，由于对类属性与对象等知识的不透彻，导致实验过程中进展很慢，通过询问同学完成实验。

实验四：

本次实验是做顺序图，一般它的顺序是用例参与者-页面-控制器-模型，通过顺序图进一步的检验上几次实验，在做第二个用例的顺序图时候，发现缺少了日程信息，进一步修改了上一次实验的类图。

实验五：

本次实验是对日程信息的状态建模，绘制其状态转换图，一开始日程信息设置“未创建”，当使用者去创建日程信息时，状态由“未创建”到“已创建”。然后使用者去查看日程信息，它的状态从“已创建”变成了“正查看”中，看完又恢复“已创建”的状态。我这个日程管理系统比较简单，所以它的状态图不是很复杂。