

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 计 2</u> |
| 实验名称 | <u>记账理财应用 app</u> | 指导教师 | <u>曾少宁</u> |
| 姓名 | <u>廖东海</u> | 学号 | <u>1414080901213</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

<记账理财应用 app>

(1) 用户添加收入信息：允许用户添加自己的收入信息，包括收入金额的大小，收入的时间，收入来源，收入类别和备注信息；

(2) 用户添加支出信息：允许用户添加自己的支出信息，包括支出金额的大小，支出的时间，支出用途，支出类别和备注信息

(2) 用户查询收支情况：可以让用户按月账目的明细情况，每一笔收支分开列表，以方便用户查询，让用户把我当月收支情况

1、实验一：需求建模 - 用例模型

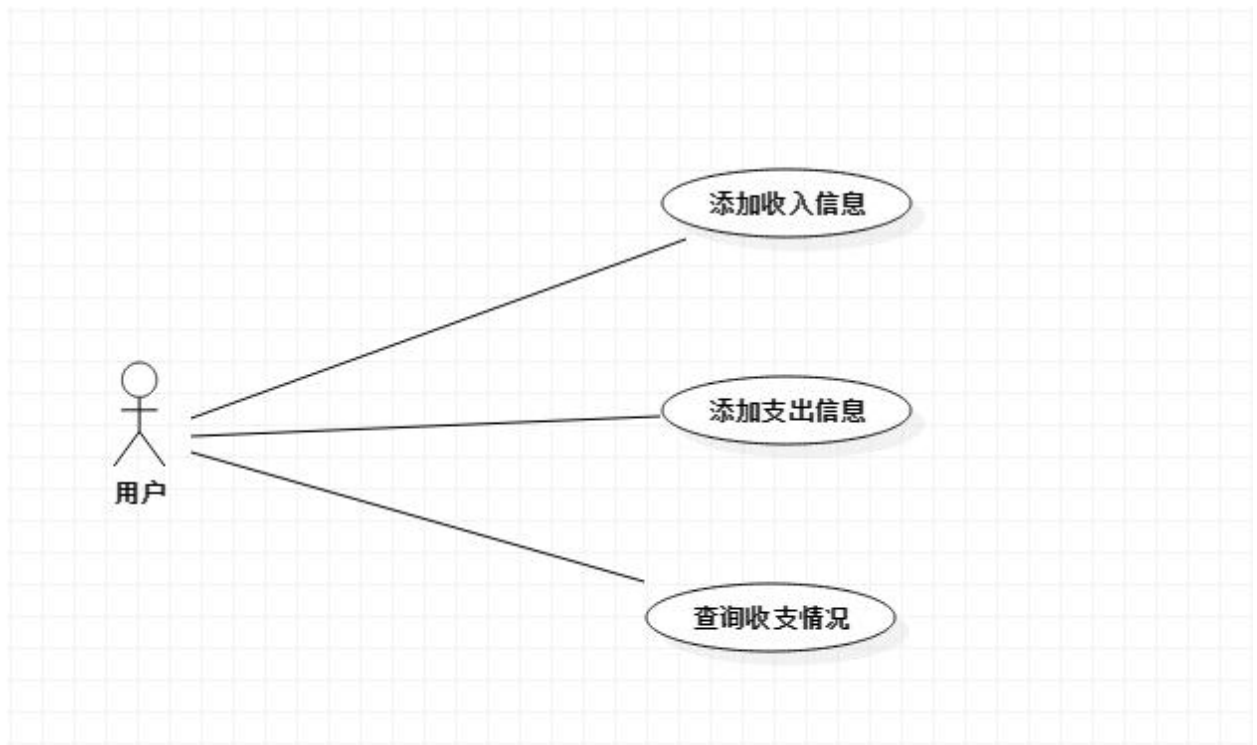


图 1：记账理财应用 APP 用例图

注：用例规约内容及项目可自行增加。

| | |
|-------|---|
| 用例编号： | UC001 |
| 用例名称： | 添加收入信息 |
| 用例描述： | 该用例允许用户添加自己的收入信息 |
| 前置条件： | 用户已登录进入系统 |
| 基本流程： | <ol style="list-style-type: none"> 1、 用户点击添加收入； 2、 用户添加收入信息，包括收入金额大小，收入的具体时间，收入的来源，收入的类别和备注信息，其中备注信息为选填部分 3、 用户点击确定按钮； 4、 系统将收入信息存入数据库； 5、 系统提示用户添加成功。 |
| 扩展流程： | 2.1 如果用户没有将完整的收入信息填入，系统将提示“添加失败”，用户可点击返回按钮重新添加信息。 |
| 后置条件： | 用户输入信息后，后台数据库及时更新 |

| | |
|-------|------------------|
| 用例编号： | UC002 |
| 用例名称： | 添加支出信息 |
| 用例描述： | 该用例允许用户添加自己的支出信息 |

| | |
|-------|---|
| 前置条件: | 用户已登录进入系统 |
| 基本流程: | 1、用户点击添加支出信息； 2、用户添加支出信息，其中包括添加的支出金额大小，支出的具体时间和支出的目的； 3、用户点击确定按钮； 4、系统将支出信息存入数据库； 5、系统提示用户添加成功。 |
| 扩展流程: | 2.1 如果用户没有将完整的支出信息填入，将提示“添加失败”，用户可点击返回键重新添加支出信息 |
| 后置条件: | 用户输入信息后，后台数据库及时更新 |

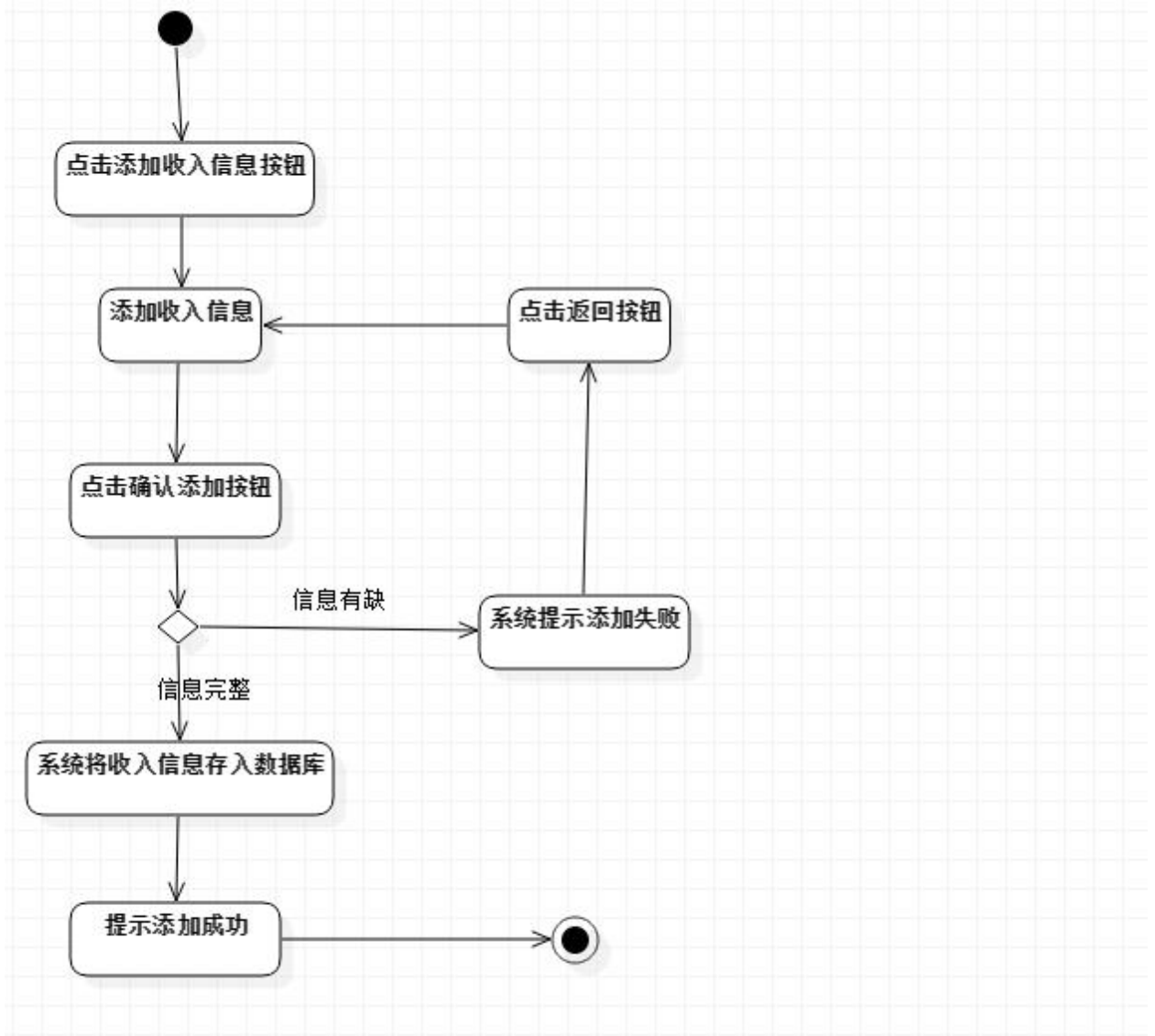
| | |
|-------|---|
| 用例编号: | UC003 |
| 用例名称: | 用户查询收支 |
| 用例描述: | 该用例允许用户查询自己当月的收支情况 |
| 前置条件: | 用户已登录进入系统 |
| 基本流程: | 1、用户点击查询按钮； 2、用户选择想查询收支信息的时间段； 3、系统将该时间段的收支信息显示出来给用户 4、系统提示用户查询收支信息成功。 |
| 扩展流程: | 2.1 如果用户在该时间段内没有添加到任何收入和支出信息，收支情况将无法查询，并提示用户“收支信息无法查询” |
| 后置条件: | 用户选择退出时，可选择返回主界面或直接退出系统 |

2、实验二：过程建模 – 活动模型

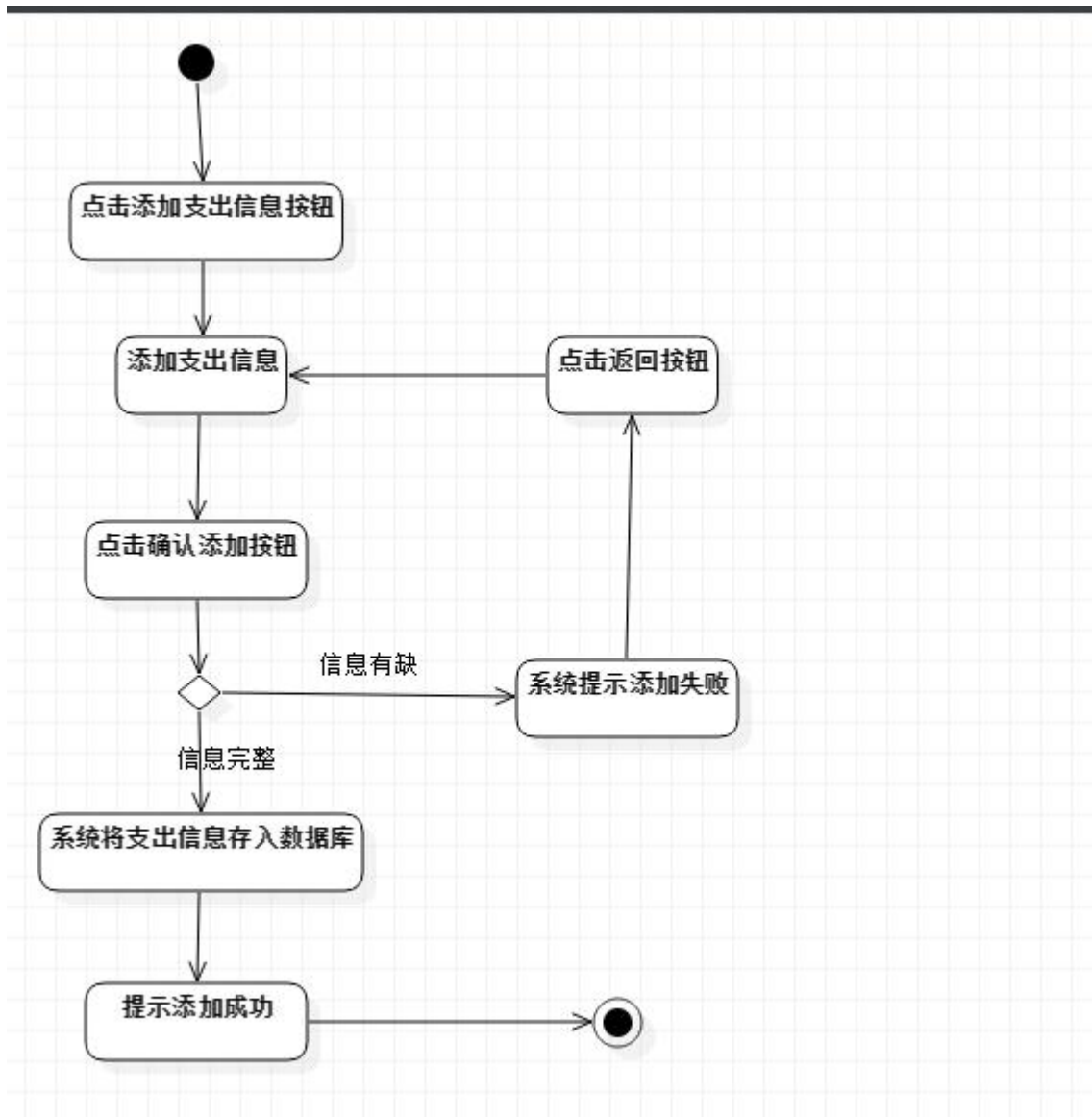
使用活动图描述系统的业务过程。

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。

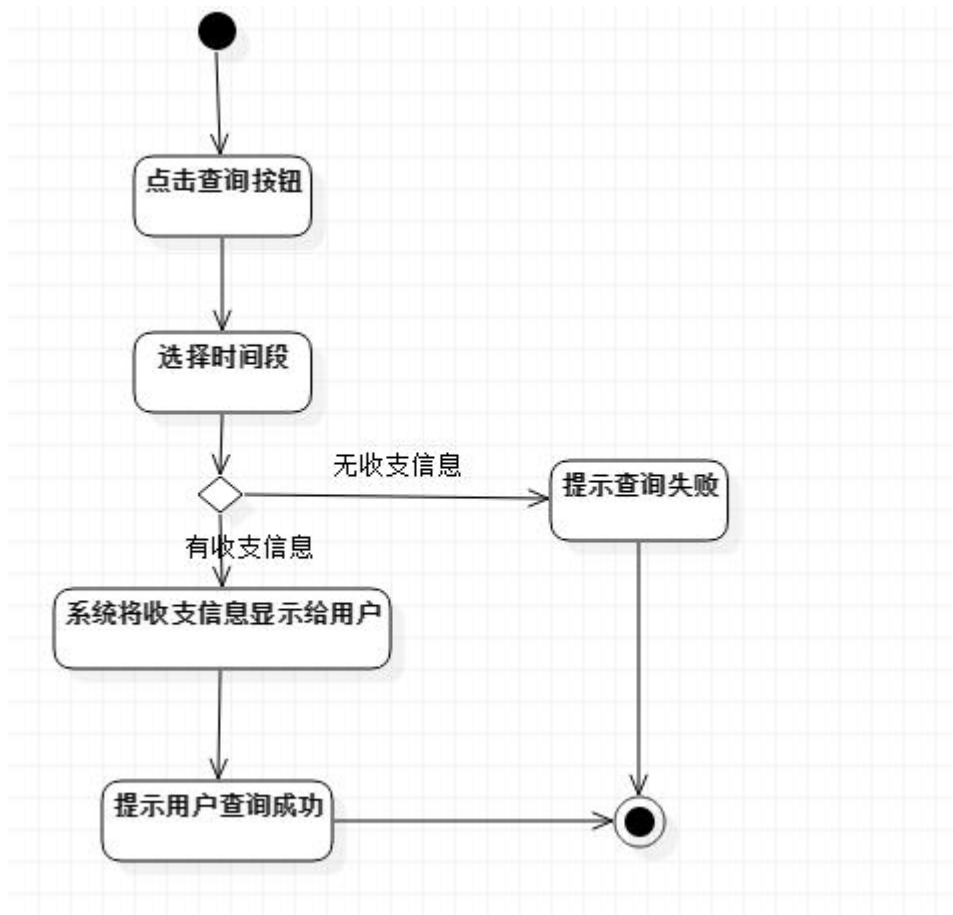
UC001 活动图



UC002 活动图



UC003 活动图

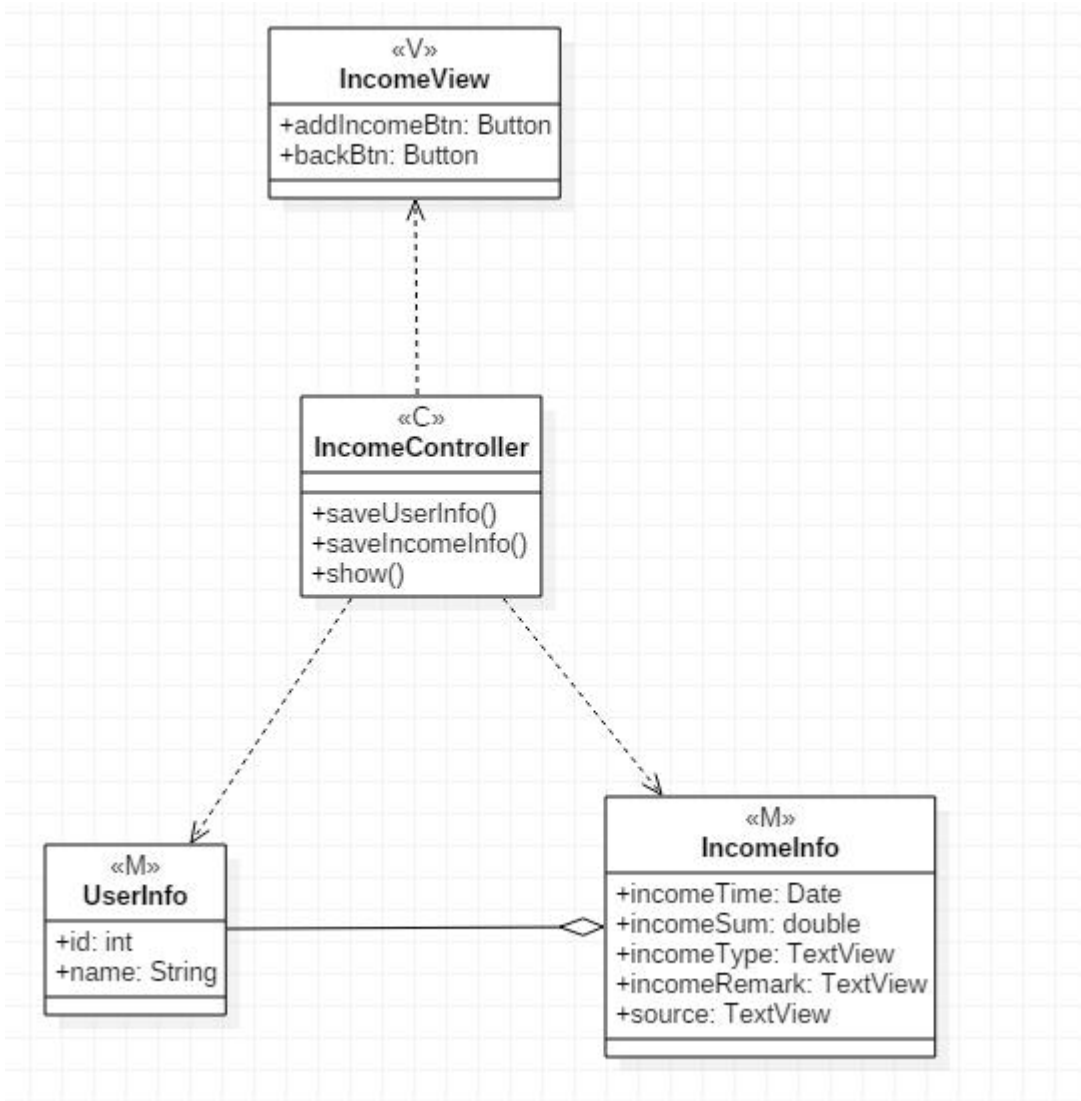


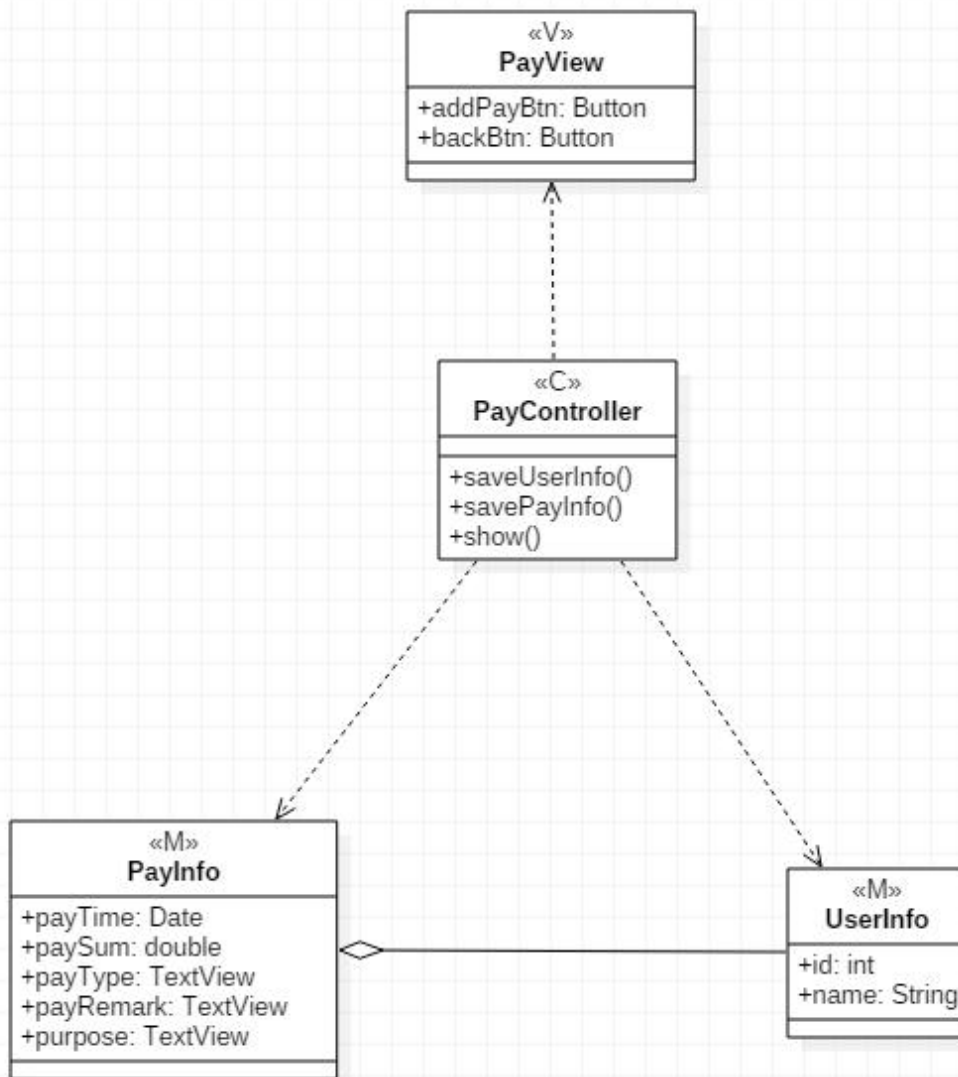
3、实验三：逻辑建模 – 类模型

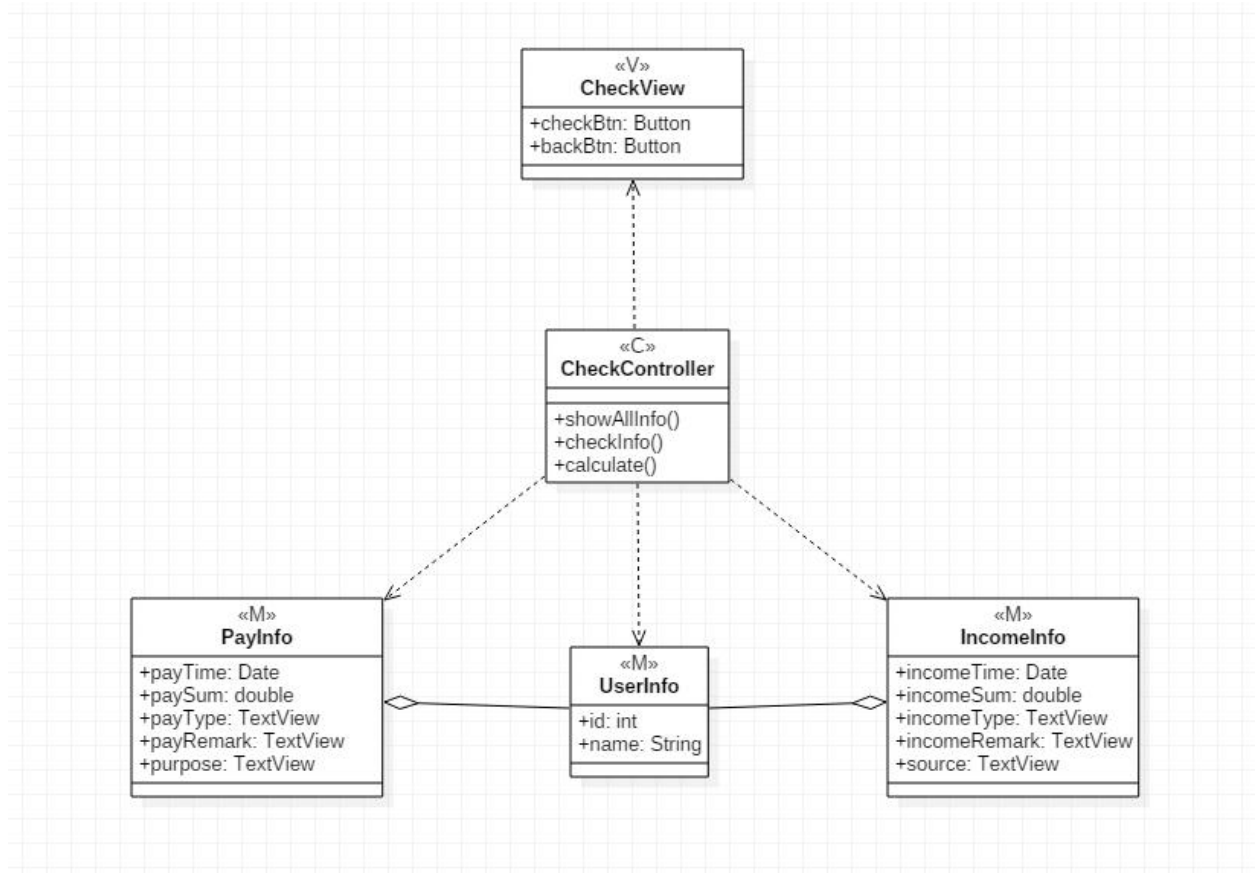
基于 MVC 设计模式找出实现用例的类。

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。

参考：讲义 P26 页。





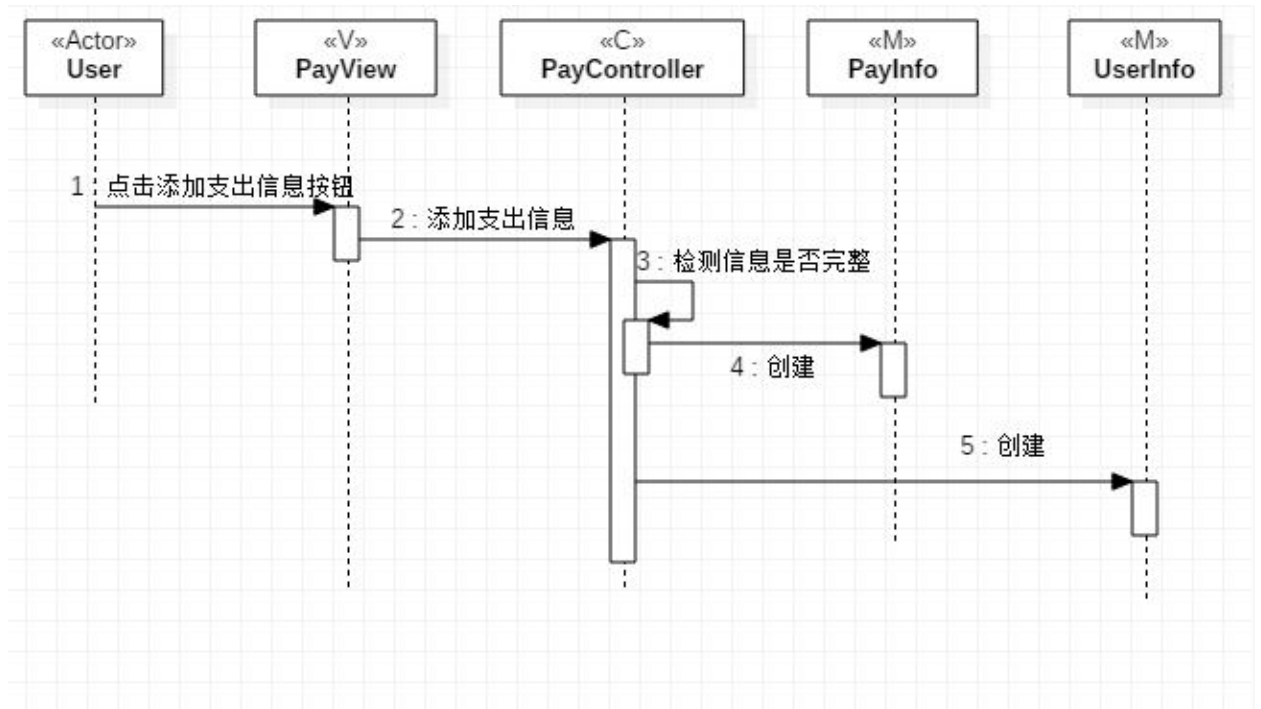
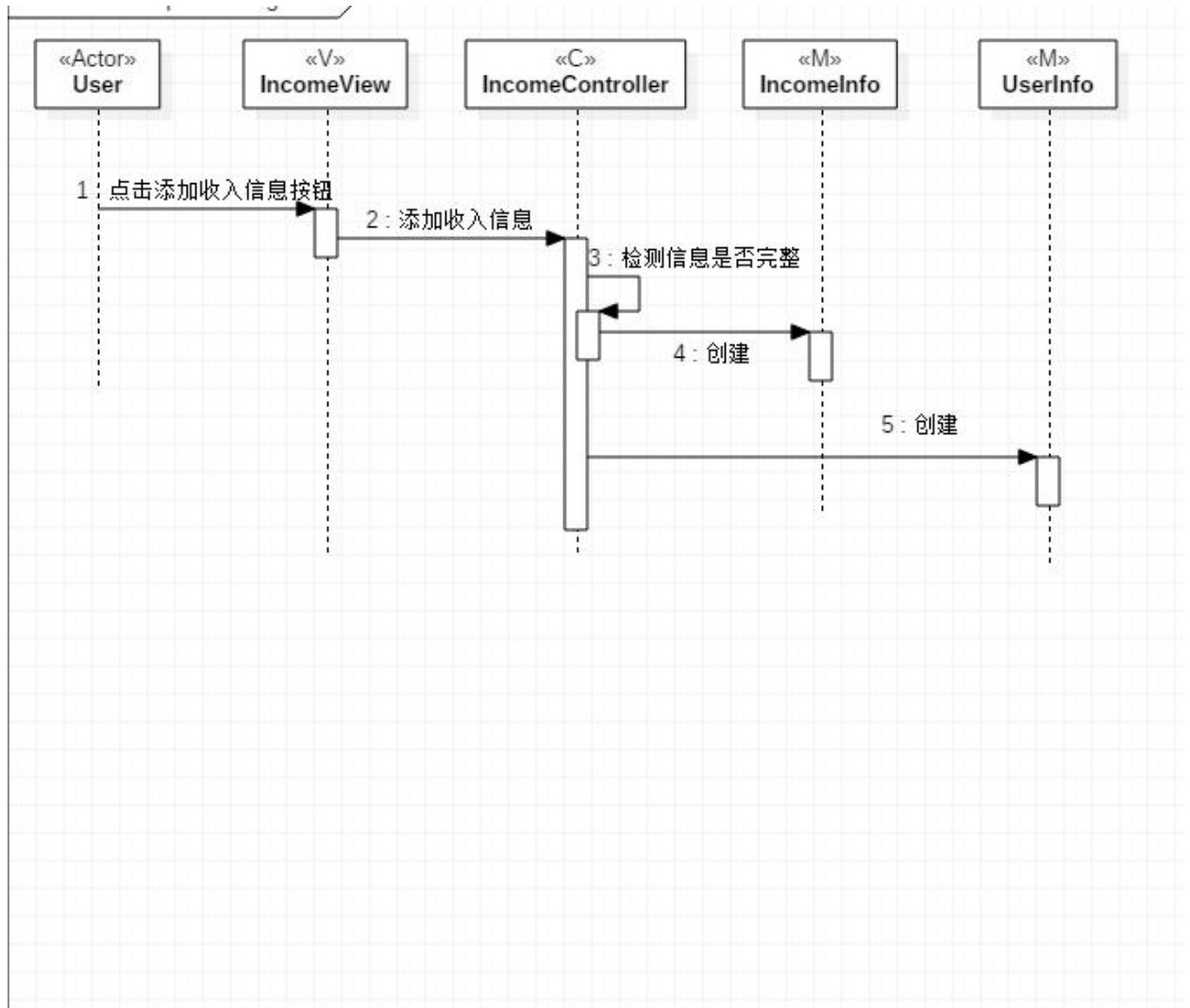


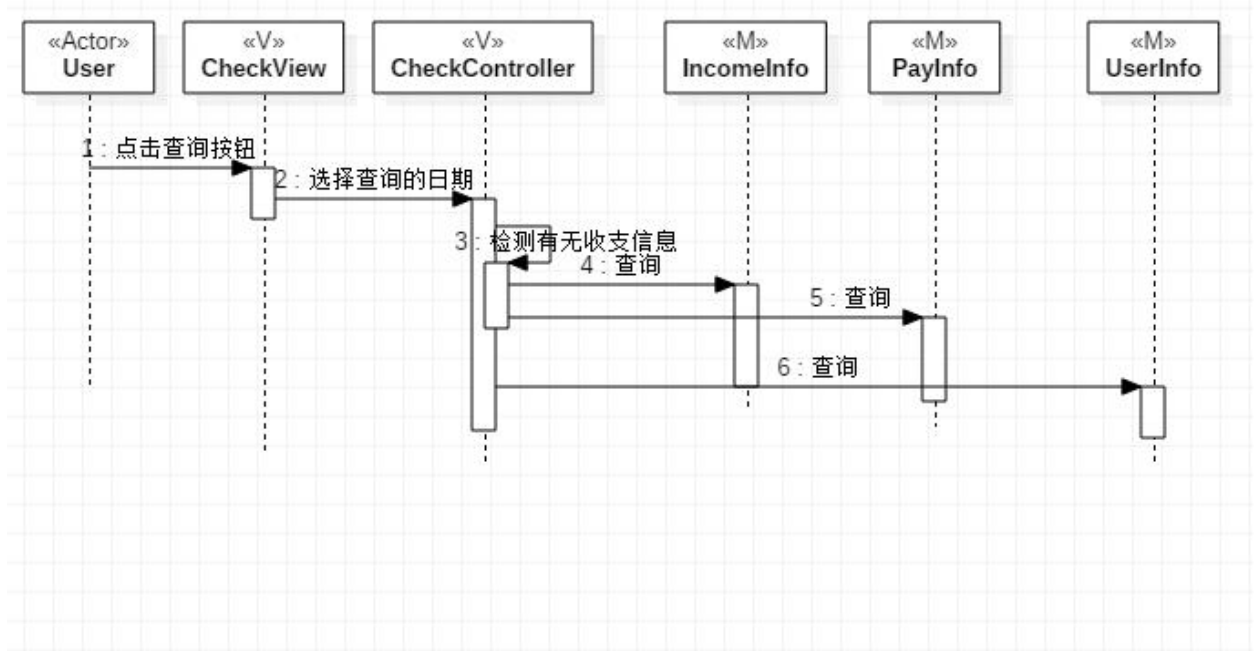
4、实验四：交互建模 – 顺序模型

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。





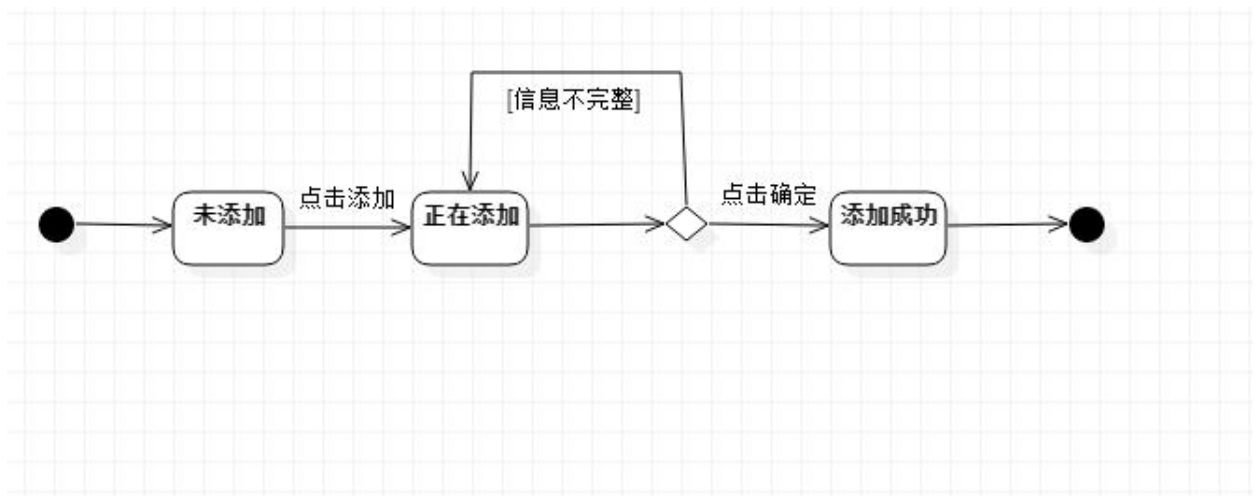
5、实验五：状态建模 – 状态模型

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。

以 IncomeInfo 类创建出的对象为例



五、实验体会

实验一：第一个实验不难，就是自己听课不够专注，然后出了问题，让老师批评了一顿，主要是画出用例图，写出用例的基本流程，需要注意的是写好用例的拓展流程里面的内容，拓展流程里面要写好流程里面容易违规的步骤，而且在写用例的基本流程的过程中不要用那些模棱两可的词去表达你想表达的东西，程序设计是一个很严谨的过程，是就是，不是就不是，用那些模棱两可的词会影响程序的设计。

实验二：实验二就是画活动图，将实验一里面的流程用活动图的形式表达出来，活动图里面的元素要和用例的基本流程对应，每一步动作要表达清楚，符号也要按照标准来画好。

实验三：实验三有点难度，主要是画类图，类图主要包括类的名称，变量的名称，方法（函数）的名称，在画的过程中也了解到了在程序语言里面的一些命名习惯，比如类名首字母大写，如果由多个单词组成，每个单词的首字母也要大写，变量名和方法名首字母小写，如果由多个单词组成，每个单词的首字母也要大写，在类与类的关系里，有继承，实现，依赖，关联，聚合，在建模的过程中，每种关系的表达也各不相同，而用的比较多的是依赖和聚合。

实验四：实验四主要是画顺序图，主要是根据前几次实验的内容来做出这次的内容，画出每一个用例里面的对象之间是如何进行交互的，明白参与者与 MVC 中的类是怎样进行交互的，然后找到每一个动作的先后执行顺序。

实验五：实验五不难，主要是画出对象的状态图，了解到状态转换之间的条件，以及对象所需要有的状态，主要是有闲置状态，正在处理的状态和处理完成的状态，但其中又可以细分几种状态，而用户通过不同的操作来完成状态的转换。