

计算机科学系实验报告

课程名称 UML 与可视化建模 班级 14 网络一班
实验名称 卡牌游戏抽卡系统 指导教师 曾少宁
姓名 陈星海 学号 1414080903104 日期 2017.5.22

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

<卡牌游戏抽卡系统>

- 1、玩家可抽取卡池（奖励池）内的卡牌；
- 2、玩家可查看每日卡牌（奖励池）更新后的卡牌列表。

1、实验一：需求建模 - 用例模型

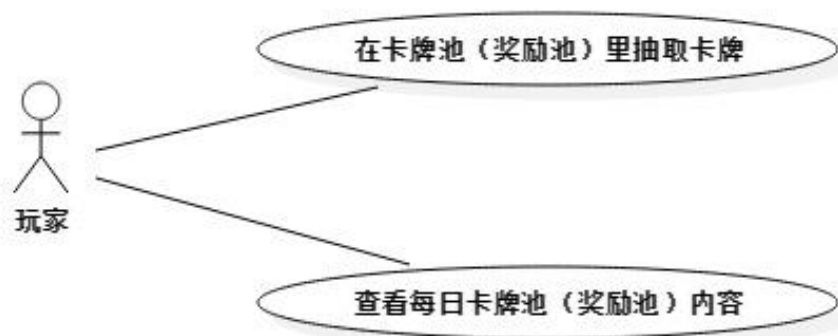


图 1：卡牌游戏抽卡系统用例图

注：用例规约内容及项目可自行增加。

用例编号:	UC001
用例名称:	在卡牌池（奖励池）里抽取卡牌
用例描述:	
前置条件:	<ol style="list-style-type: none"> 1、 存在一个总数据库，存放着 50 张卡牌的 id、名字、星级、背景故事、能力属性； 2、 存在一个名为“卡牌池”的空数据库，用于存放 20 张卡牌的 id、名字、星级； 3、 系统每日凌晨 0:00 自动更新“卡牌池”数据库内的所有数据，即系统根据算法随机在 50 个 id 号码内选取 20 个，并根据这 20 个 id 号码向总数据库发送数据请求以接受获取相应卡牌信息（卡牌的 id、名字、星级），再存入“卡牌池”数据库； 4、 处于抽卡页面。
基本流程:	<ol style="list-style-type: none"> 1、 玩家在卡牌游戏抽卡页面中点击“抽卡”按钮； 2、 系统跳转到新的页面（抽卡实现页面）； 3、 系统根据算法在“卡牌池”数据库里的 20 个 id 内随机抽取一个 id 号码； 4、 系统根据上述得到的 id 号码向总数据库发送数据请求； 5、 系统获取该卡牌的内容（卡牌的名字、星级、背景故事、攻击力、防御力、血量）； 6、 系统将获得卡牌相关信息（卡牌的名字、星级、背景故事、攻击力、防御力、血量）显示到本页面内，卡牌信息显示完毕后在列表底部弹出“返回”按钮； 7、 玩家在本页面内点击“返回”按钮； 8、 系统返回到抽卡页面。
扩展流程:	<ol style="list-style-type: none"> 1.1、 如果玩家未点击“抽卡”按钮，系统不做任何操作； 4.1、 若发送请求失败，弹出提示信息“请求发送失败”，并返回到抽卡界面； 5.1、 若获取数据失败，弹出提示信息“数据获取失败”，并返回到抽卡界面； 7.1、 如果玩家未点击“返回”按钮，系统不做任何操作。
后置条件:	

用例编号:	UC002
用例名称:	查看每日更新卡牌池（奖励池）内容

用例描述:	
前置条件:	<ol style="list-style-type: none"> 1、 存在一个总数据库，存放着 50 张卡牌的 id、名字、星级、背景故事、能力属性； 2、 存在一个名为“卡牌池”的空数据库，用于存放 20 张卡牌的 id、名字、星级； 3、 系统每日凌晨 0:00 自动更新“卡牌池”数据库内所有数据，即系统根据算法随机在 50 个 id 号码内选取 20 个，并根据这 20 个 id 号码向总数据库发送数据请求以接受获取相应卡牌信息（卡牌的 id、名字、星级），再存入“卡牌池”数据库； 4、 处于抽卡页面。
基本流程:	<ol style="list-style-type: none"> 1、 玩家点击“查看最新卡牌池”按钮； 2、 系统跳转到新的页面（查看页面）； 3、 系统向“卡牌池”数据库发送数据请求； 4、 系统获取“卡牌池”数据库的所有卡牌相应信息（卡牌的名字、星级）； 5、 系统将上述得到的卡牌信息（卡牌的名字、星级）显示到本页面内，卡牌信息显示完毕后在列表底部弹出“返回”按钮； 6、 玩家在上述页面内点击“返回”按钮； 7、 系统返回到抽卡页面。
扩展流程:	<ol style="list-style-type: none"> 1.1、 如果玩家未点击“查看最新卡牌池”按钮，系统不做任何操作； 3.1、 若发送请求失败，弹出提示信息“请求发送失败”，并返回到抽卡界面； 4.1、 若获取数据失败，弹出提示信息“数据获取失败”，并返回到抽卡界面； 6.1、 如果玩家未点击“返回”按钮，系统不做任何操作。
后置条件:	

2、实验二：过程建模 – 活动模型

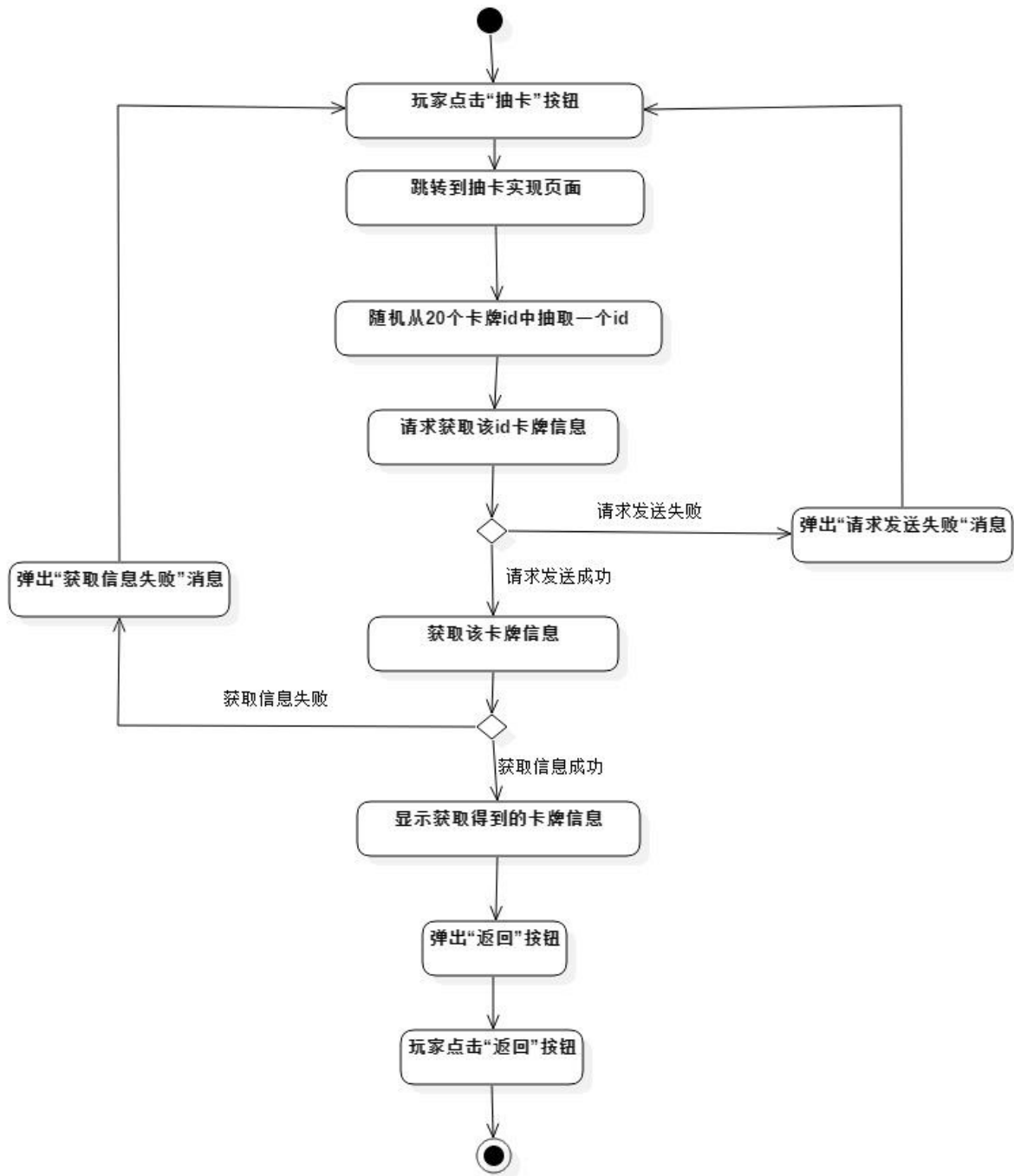


图 2.1 用例 UC001 活动图

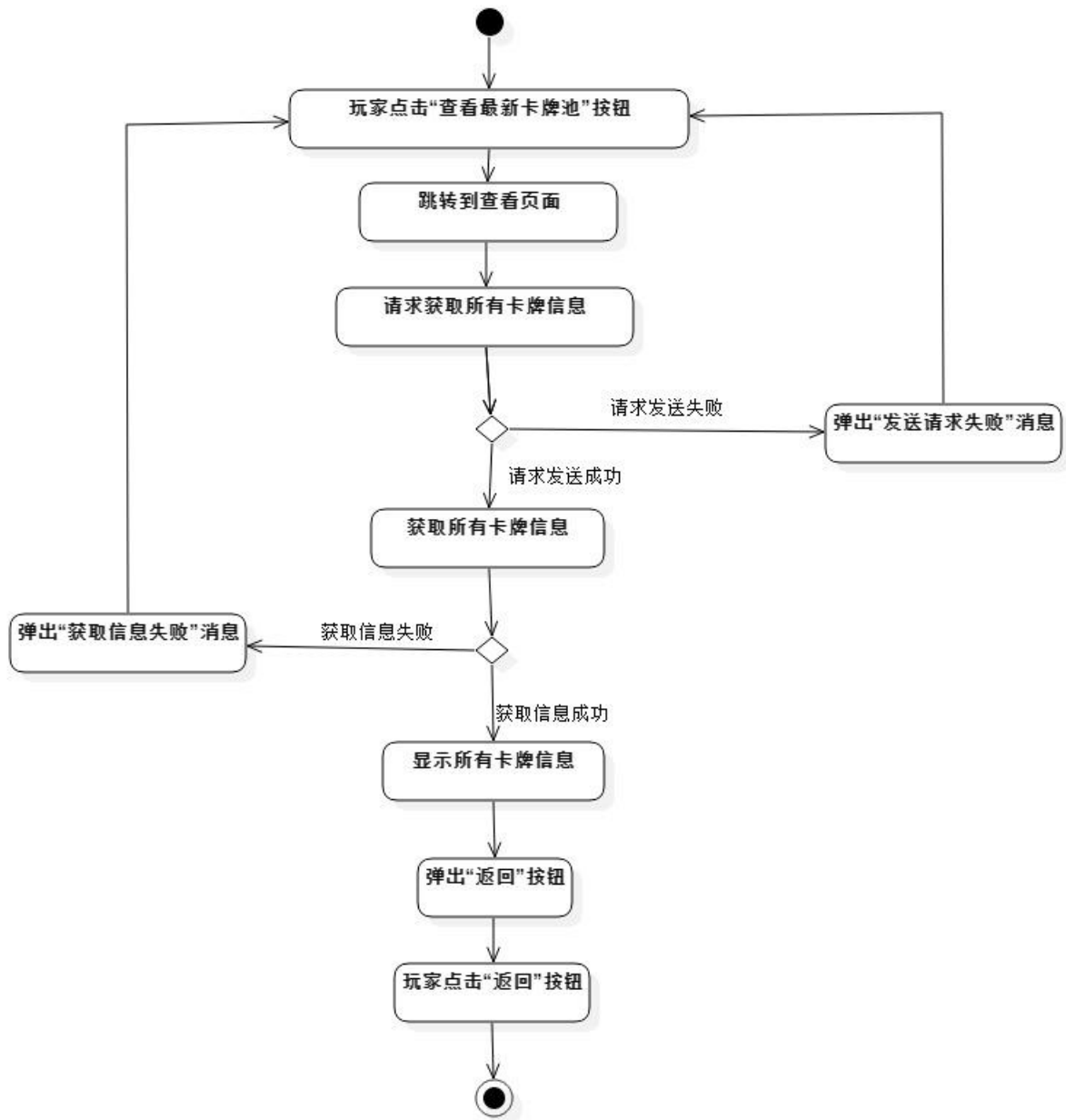


图 2.2 用例 UC002 活动图

3、实验三：逻辑建模 - 类模型

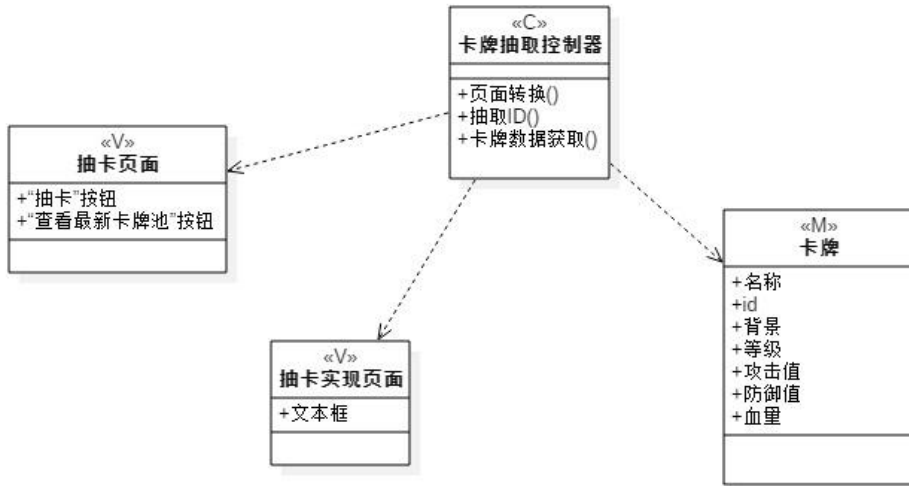


图 3.1 用例 UC001 类图

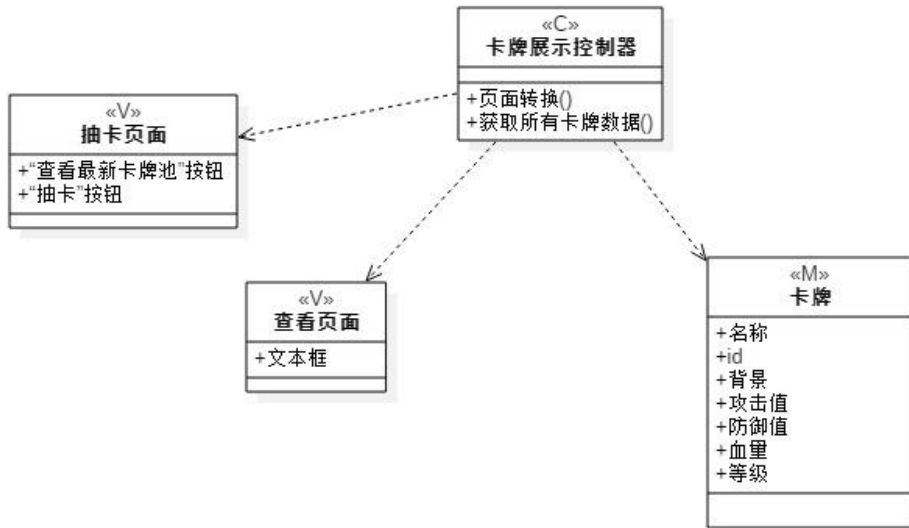


图 3.2 用例 UC002 类图

4、实验四：交互建模 – 顺序模型

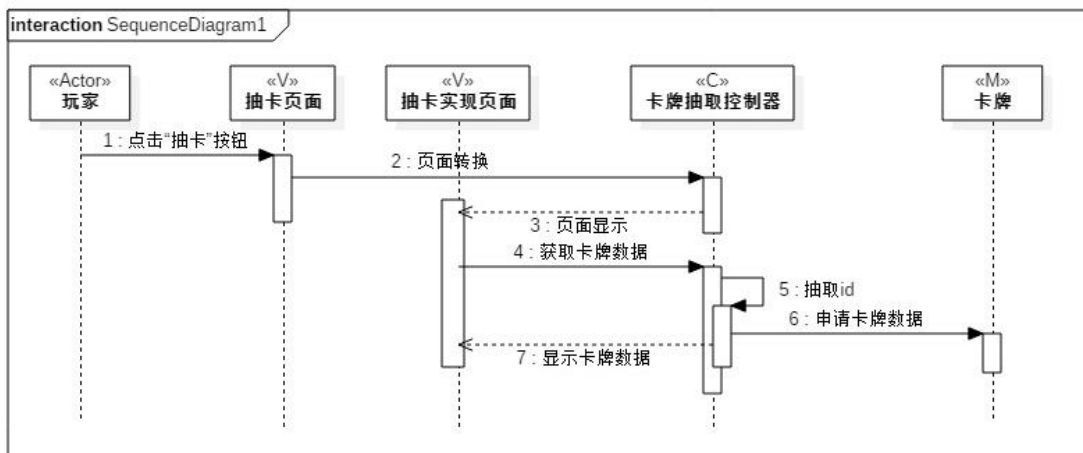


图 4.1 用例 UC001 时序图

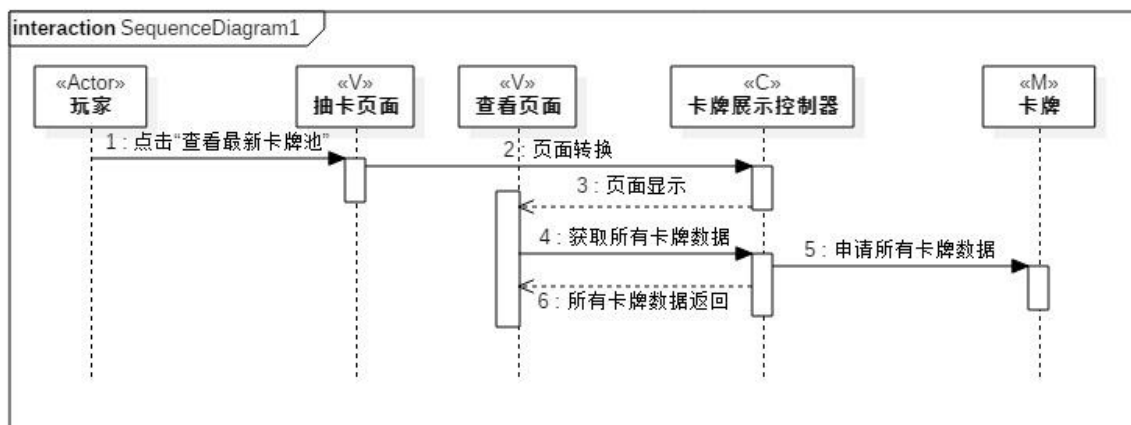


图 4.2 用例 UC002 时序图

5、实验五：状态建模 – 状态模型

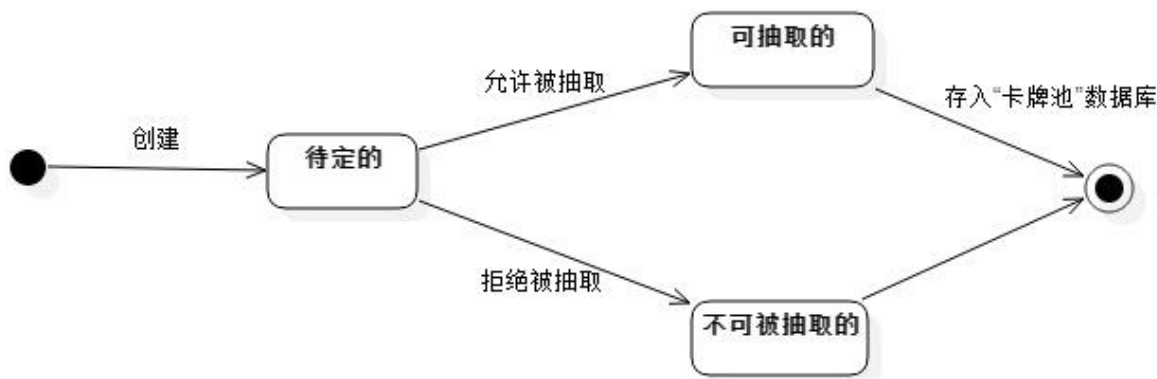


图 5.1 “卡牌”模型状态图

五、实验体会

实验一：第一次使用 uml 建模，尽管课上老师已经说清楚了用例设计的具体要求，但是在做实验工程中还是会出现不少问题。虽然自己做实验的时候已经写了很多文字，但是在建模规则里，这些只是废话，并没有以计算机的角度进行编写，以至于修改了不少次。至于现在该文档内的用例，我也不能保证是正确的，还需要老师多多指点。

实验二：活动图的设计过程中，由于考虑到在实验一中老师的评价是未说明清楚操作过程，以至于在本次活动图里面我的活动图操作名称基本就是在用例图里面粘贴复制，删了某些东西感觉就不对了，导致其内容过于复杂，因为总是怕没有说明清楚具体的操作步骤。再者就是那条粗线，因为一开始我的用例里边有两个操作是同时进行的，我在 ppt 里或者网上的资料里发现说粗线是指并列的关系，可能我没有看清楚或者理解错误了，导致直接使用出现了错误。

实验三：类图的设计过程中，虽然说基本理解了 MVC 的架构，但是在套用到自己的题目中去，发现还是不简单，一开始也是自己想复杂了，一直在考虑着“怎么连接数据库，怎么获取数据库信息”这样的步骤属于哪个模块的工作，一开始也理解错误了以为也是属于控制器模块的工作，但是跟数据库有关的工作已经属于更底层的工作了，并不包括在 MVC 当中，当然数据访问本身就包含在了 Model 模块当中了。然后我的设计思路来自于 Java Web 架构，想着在 Java Web 里，JSP 应该充当 V，Servlet 应该充当 C，而 JavaBean 应该充当 M，当然我还不知道这样是否真的正确，怕自己还是在哪些地方理解错误了，还需要老师多多指点。

实验四：本次的实验，我的困扰主要在于那个转换页面该怎么弄，本着考虑到一个用例尽可能的只使用一个控制器，但是又想到这样子设计我两个用例的控制器就会有重合的代码，不知道这样会不会有影响。另外，我感觉设计类图和时序图可以同时设计，这样在类图的时候就不会想到什么属性或者方法就填进类图里，确实上次作业里我的类图看上去就是很乱的，别人是根本不知道我想表达什么。再者，抱歉了，老师晚交作业了。

实验五：本次实验相对简单，当然主要是自己的这个“卡牌”模型并未涉及过多的状态变化，简单说明下，由于本系统只是涉及到“抽取”的过程，其实就是根据抽取到的 id 号到数据库内获取数据，其中该“卡牌”模型的数据是已经固定不变，只能在修改数据库的时候才能改变，又加上在用例设计中的前置条件里提及到“更新”，即每天能抽取的卡牌不同，所以存在可抽取和不可抽取的状态。不过主要问题还是出现在上个实验时序图中的“页面转换”，这次的想法是在“抽卡页面”内发送消息给控制器，控制器再发送一个返回值给新的页面，新的页面根据返回值再向控制器发送获取卡牌数据的消息，想是这么想，但是不知道是否正确。我参考了几份其他同学的作业，均未涉及到页面转换的过程。