



# 实验室预约管理系统

## ——面向对象分析方法

学院：智能与计算学部

专业：计算机科学与技术（新工科试验班）

年级：2019 级

班级：2 班

姓名：**胡轶然**

学号：3019244355

## 目录

1	需求描述 .....	5
2	为问题域建立概念模型 .....	5
2.1	备选类 .....	5
2.2	决定候选类 .....	5
2.3	类之间的关联描述 .....	6
2.4	为类添加责任 .....	7
2.5	构建类图 .....	8
3	用例建模 .....	9
3.1	需求捕获 .....	9
3.2	识别参与者 .....	9
3.3	合并需求获得用例 .....	10
3.4	细化用例描述 .....	11
3.5	用例图 .....	14
4	鲁棒分析 .....	15
4.1	鲁棒分析原理 .....	15
4.2	各个事件流鲁棒分析 .....	15
4.2.1	用户登录事件流 .....	15
4.2.2	学生预约事件流 .....	16
4.2.3	学生取消预约事件流 .....	17
4.2.4	教师审核预约事件流 .....	18
4.2.5	管理员添加账号事件流 .....	19
4.2.6	管理员添加实验室事件流 .....	20
4.2.7	管理员清空预约事件流 .....	21
4.2.8	用户查看预约事件流 .....	21
5	交互建模 .....	22
5.1	用户登录时序图 .....	22
5.2	学生预约时序图 .....	23
5.3	学生取消预约时序图 .....	24
5.4	教师审核预约时序图 .....	25
5.5	管理员添加账号时序图 .....	26

5.6	管理员添加实验室时序图 .....	27
5.7	管理员清空预约时序图 .....	28
5.8	用户查看预约时序图 .....	28
6	构件与活动图 .....	29
6.1	构件图 .....	29
6.2	活动图 .....	30
6.2.1	UC01 活动图 .....	30
6.2.2	UC02 活动图 .....	31
6.2.3	UC03 活动图 .....	32
6.2.4	UC04 活动图 .....	33
6.2.5	UC05 活动图 .....	34
6.2.6	UC06 活动图 .....	35
6.2.7	UC07 活动图 .....	36
6.2.8	UC08 活动图 .....	37
7	软件测试 .....	37
7.1	黑盒测试 .....	37
7.1.1	学生登录模块测试 .....	37
7.1.2	教师登录模块测试 .....	40
7.1.3	管理员登录模块测试 .....	43
7.1.4	预约模块测试 .....	44
7.1.5	审批模块测试 .....	47
7.1.6	添加学生账号模块测试 .....	49
7.1.7	添加教师账号模块测试 .....	49
7.1.8	添加实验室模块测试 .....	50
7.2	白盒测试 .....	51
7.2.1	学生登录模块测试 .....	51
7.2.2	教师登录模块测试 .....	53
7.2.3	管理员登录模块测试 .....	55
7.2.4	预约模块测试 .....	57
7.2.5	审批模块测试 .....	61
7.2.6	添加学生账号模块测试 .....	64

7.2.7	添加教师账号模块测试 .....	66
7.2.8	添加实验室模块测试 .....	68
8	项目总结 .....	69

# 1 需求描述

在大学某些专业的学习中需要经常使用实验室的资源，学生不一定是在上课时间集体进行实验，也有可能在课余时间使用实验室，因此一套实验室预约系统就显得尤为重要。系统可以进行如下功能：学生可以在课余时间预约实验室，并查看自己的预约信息以及取消预约；实验室老师可以查看所有的预约信息，并且需要对学生的预约信息进行审核；系统管理员有权限添加系统中的实验室信息、老师信息和学生信息，并且可以在必要的时候清空所有预约记录。

## 2 为问题域建立概念模型

### 2.1 备选类

识别名词：

大学、专业、实验室、资源、学生、实验、实验室预约系统、系统、功能、预约信息、实验室老师、系统管理员、实验室信息、老师信息、学生信息、预约记录

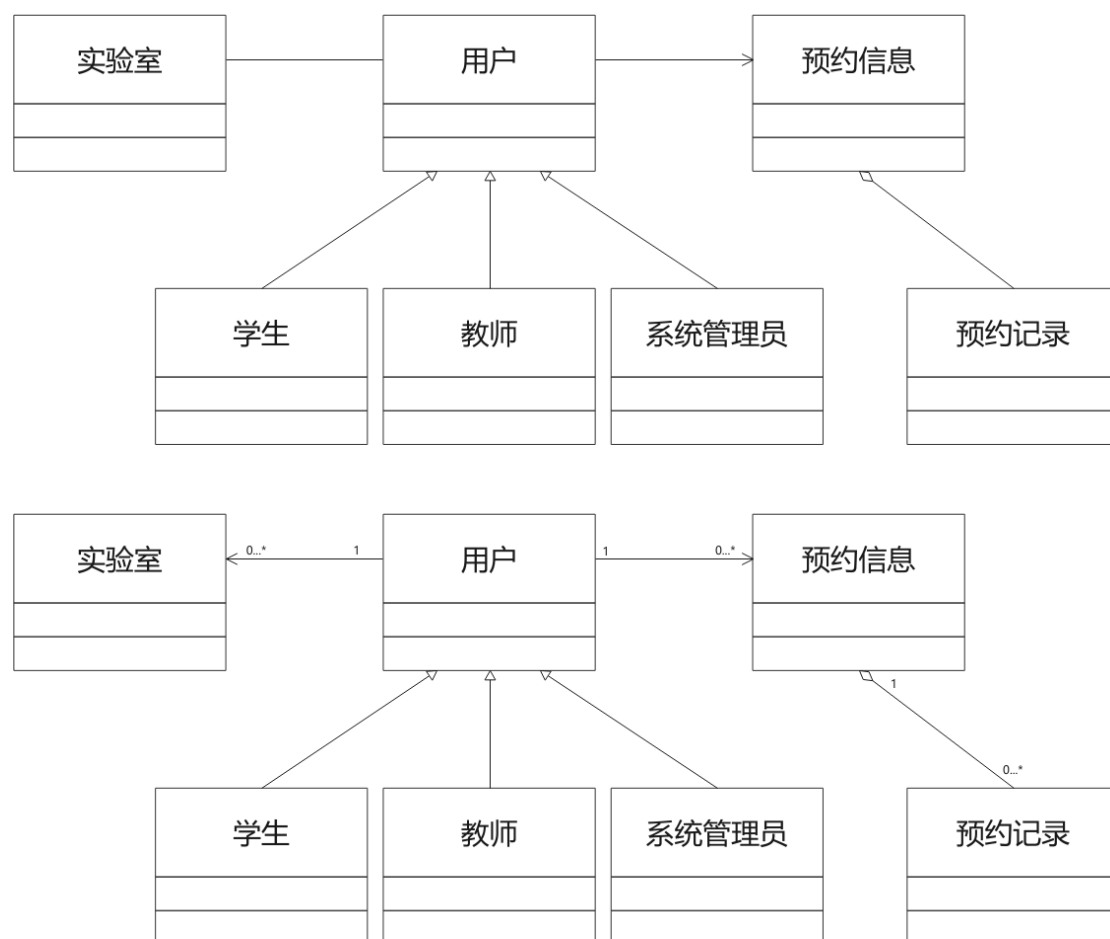
### 2.2 决定候选类

名词	是否保留	理由
大学、专业、实验	否	系统外的概念，无须建模
实验室预约系统、系统、功能	否	将要开发的系统和功能本身，无须建模
实验室、资源	是	代表实验室类，即被预约的主体
实验室信息	否	实验室的基本信息，可以作为实验室属性
学生	是	一个很重要的类，使用预约功能的主体，作为用户的子类
学生信息	否	学生的基本信息，可以作为学生属性
实验室老师	是	一个很重要的类，使用审批功能的主体，作为用户的子类
老师信息	否	教师的基本信息，可以作为教师属性
系统管理员	是	代表系统的管理人员，作为用户的子类
预约信息	是	代表预约的记录日志
预约记录	是	代表日志的一条记录

候选类列表

实验室
用户
学生
教师
系统管理员
预约信息
预约记录

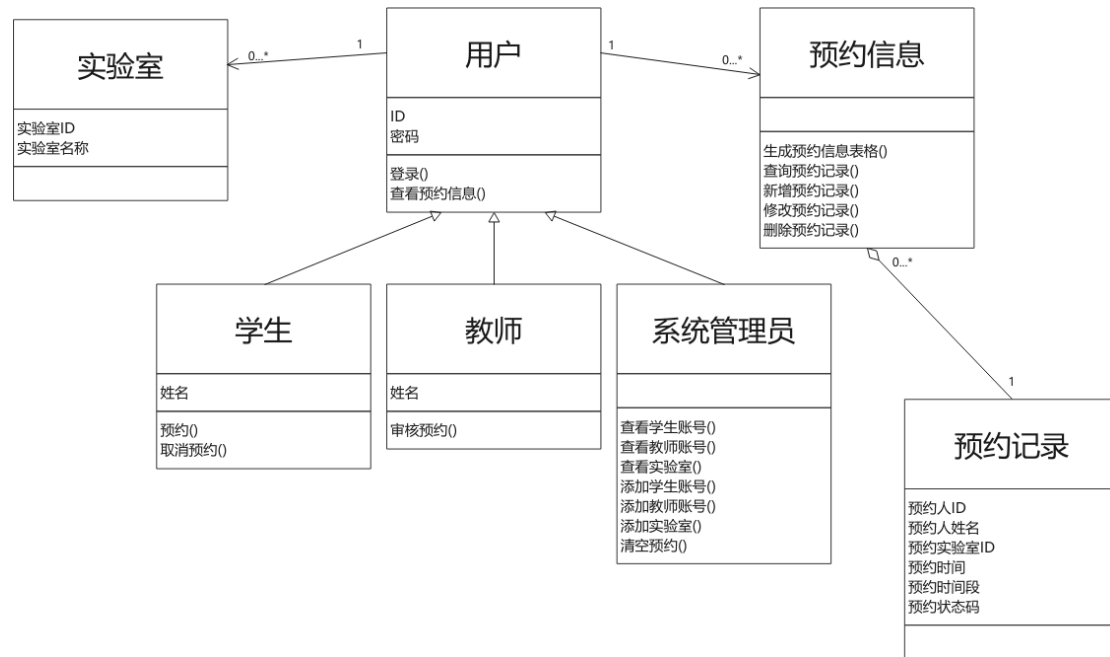
## 2.3 类之间的关联描述



## 2.4 为类添加责任

用户	成员变量	ID
		密码
	成员方法	登录
		查看预约信息
学生	成员变量	姓名
	成员方法	预约
		取消预约
教师	成员变量	姓名
	成员方法	审核预约
系统管理员	成员变量	
	成员方法	查看学生账号
		查看教师账号
		查看实验室
		添加学生账号
		添加教师账号
		添加实验室
		清空预约
实验室	成员变量	实验室 ID
		实验室名称
	成员方法	
预约记录	成员变量	预约人 ID
		预约人姓名
		预约实验室 ID
		预约时间
		预约时间段
		预约状态码
预约信息	成员变量	
	成员方法	生成预约信息表格
		查询预约记录
		新增预约记录
		修改预约记录
		删除预约记录

## 2.5 构建类图





## 3 用例建模

### 3.1 需求捕获

功能特性列表
FEAT01. 通过账号密码登录系统
FEAT02. 账号密码错误有提示
FEAT03. 预约实验室
FEAT04. 生成预约记录进行存储
FEAT05. 不可以预约不开放的实验室和时间段
FEAT06. 审批实验室预约申请
FEAT07. 只能审批待审批的申请
FEAT08. 审批通过或者不通过
FEAT09. 审批后修改预约记录
FEAT10. 添加账号
FEAT11. 账号重复有提示
FEAT12. 添加实验室
FEAT13. 添加实验室重复有提示
FEAT14. 清空预约
FEAT15. 清空预约需要确认
FEAT16. 查看预约信息
FEAT17. 学生只能查看自己的预约信息
FEAT18. 学生取消预约

### 3.2 识别参与者

参与者分析
学生
教师
系统管理员

### 3.3 合并需求获得用例

特征	用例
FEAT01. 通过账号密码登录系统 FEAT02. 账号密码错误有提示	UC01. 登录
FEAT03. 预约实验室 FEAT04. 生成预约记录进行存储 FEAT05. 不可以预约不开放的实验室和时间段	UC02. 预约实验室
FEAT18. 学生取消预约	UC03. 取消预约实验室
FEAT06. 审批实验室预约申请 FEAT07. 只能审批待审批的申请 FEAT08. 审批通过或者不通过 FEAT09. 审批后修改预约记录	UC04. 审批预约申请
FEAT10. 添加账号 FEAT11. 账号重复有提示	UC05. 添加账号
FEAT12. 添加实验室 FEAT13. 添加实验室重复有提示	UC06. 添加实验室
FEAT14. 清空预约 FEAT15. 清空预约需要确认	UC07. 清空预约
FEAT16. 查看预约信息 FEAT17. 学生只能查看自己的预约信息	UC08. 查看预约信息

### 3.4 细化用例描述

用例名称		UC01. 登录
简要说明		用户通过账号密码登录进入系统，才能使用其他功能
事件流	基本	1) 系统展示登录界面 2) 用户选择登录账户类型 3) 用户输入账号密码 4) 系统检查账号密码 5) 进入系统
	拓展	4a) 系统检查账号密码输入是否正确，如果不正确则要求用户重新登录
非功能需求		无
前置条件		无
后置条件		用户登录进入系统
拓展点		无
优先级		最高

用例名称		UC02. 预约实验室
简要说明		学生预约实验室，系统记录下这条预约记录
事件流	基本	1) 学生发出预约实验室的请求 2) 系统展示能够预约的时间 3) 学生选择时间 4) 系统确认时间并展示一天内的时间段 5) 学生选择时间段 6) 系统确认时间段并展示可以选择的实验室 7) 学生选择实验室 8) 系统确认实验室 9) 系统记录下预约记录
	拓展	4a) 系统检查时间输入是否正确，如果不正确则要求学生重新输入 6a) 系统检查时间段输入是否正确，如果不正确则要求学生重新输入 8a) 系统检查实验室输入是否正确，如果不正确则要求学生重新输入
非功能需求		无
前置条件		学生已经登录
后置条件		预约记录写入到数据文件中
拓展点		无
优先级		最高

用例名称		UC03. 取消预约实验室
简要说明		学生对已经预约的记录进行取消
事件流	基本	1) 学生发出取消预约的申请 2) 系统展示学生的预约记录 3) 学生选择一条预约记录进行取消 4) 系统检查预约记录 5) 系统修改预约记录文件，删除相应的条目
	拓展	4a) 系统检查学生输入的条目是否能够取消，如果不能则取消失败
非功能需求		无
前置条件		学生已经登录
后置条件		被取消的预约记录从数据文件中删除
拓展点		无
优先级		最高

用例名称		UC04. 审批预约申请
简要说明		教师对学生待审批的预约记录进行审批
事件流	基本	1) 教师发出审批预约的申请 2) 系统展示待审批的预约条目 3) 教师选择一条待审批的条目 4) 系统检查教师输入的条目 5) 教师选择审批通过或者不通过 6) 系统修改对应的预约条目
	拓展	4a) 系统检查教师输入的条目是否是待审批的，如果不是则要求重新输入
非功能需求		无
前置条件		教师已经登录
后置条件		预约记录的预约状态被修改为通过或者未通过
拓展点		无
优先级		最高

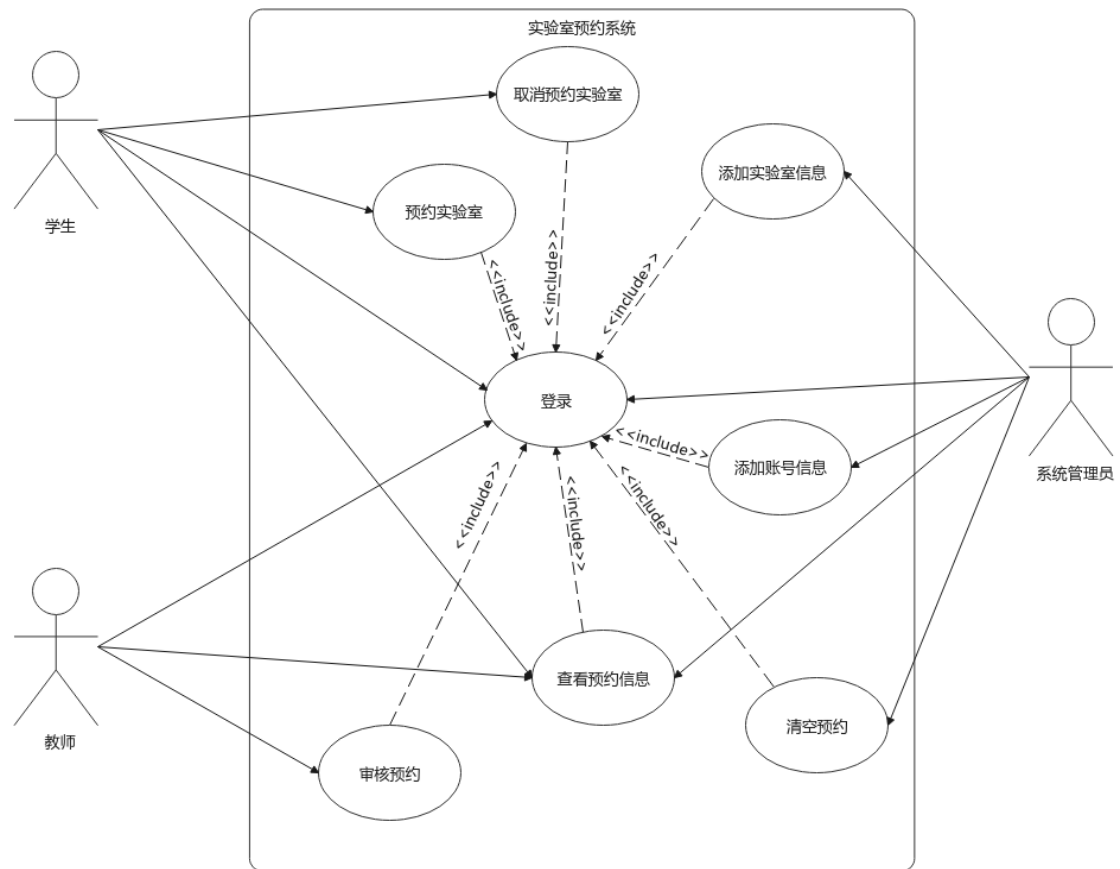
用例名称		UC05. 添加账号
简要说明		系统管理员添加学生和老师的账号信息
事件流	基本	1) 管理员选择添加学生账号还是教师账号 2) 管理员输入 ID、姓名和密码 3) 系统检查 ID 是否有重复 4) 系统将新加的账号加入到相应的账号信息文件中
	拓展	3a) 系统检查加入的账号是否有重复，如果有则添加失败
非功能需求		无
前置条件		管理员已经登录
后置条件		新的账号信息被加入到相应的数据文件中
拓展点		无
优先级		最高

用例名称		UC06. 添加实验室
简要说明		系统管理员添加实验室信息
事件流	基本	1) 系统管理员发出添加实验室申请 2) 管理员输入实验室 ID 和实验室名称 3) 系统检查实验室 ID 是否有重复 4) 系统将新加的实验室信息加入到实验室信息文件中
	拓展	5a) 系统检查加入的实验室 ID 是否有重复, 如果有则添加失败
非功能需求		无
前置条件		管理员已经登录
后置条件		新的实验室信息被加入实验室信息数据文件中
拓展点		无
优先级		最高

用例名称		UC07. 清空预约
简要说明		系统管理员清空所有的预约记录
事件流	基本	1) 系统管理员提出清空预约记录的申请 2) 系统向管理员确认 3) 管理员确认 4) 系统清空所有预约记录
	拓展	无
非功能需求		无
前置条件		管理员已登录
后置条件		所有预约记录被删除
拓展点		无
优先级		最高

用例名称		UC08. 查看预约信息
简要说明		用户查看预约信息
事件流	基本	1) 用户提出查看预约信息的请求 2) 系统展示预约信息
	拓展	无
非功能需求		无
前置条件		用户已登录
后置条件		无
拓展点		无
优先级		最高

### 3.5 用例图



## 4 鲁棒分析

### 4.1 鲁棒分析原理

**正确性检查：**鲁棒分析将通过比顺序图更简单、更有效的图形来描绘用例中的传递过程，从而确保用例是正确的。

**完整性检查：**通过鲁棒分析，你可以很容易地找到用例描述中所有必须的扩展路径。

**持续发现对象：**在做鲁棒分析时，你将开始思考具体的细节，从而很容易发现在域建模时遗失的对象，而这些对象在绘制顺序图时不容易被重新找到，而只会让你感觉根本无法绘制出相应的顺序图。

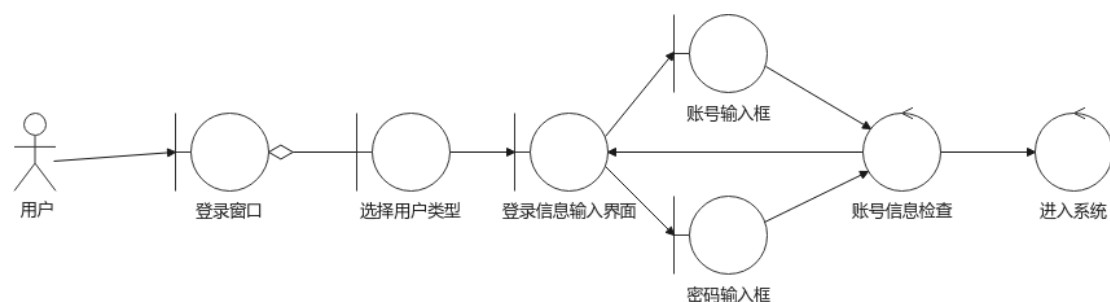
**对象确定：**类似的，我们可以发现对象，也可以确认对象，也就是能够很容易地发现不正确的类。

**初步设计：**正如后面会提到的，鲁棒分析将类分成了实体对象类、控制对象、边界对象三种，这种观点正好与 MVC 体系结构不谋而合，这样可以保证设计出更加稳健的系统。

### 4.2 各个事件流鲁棒分析

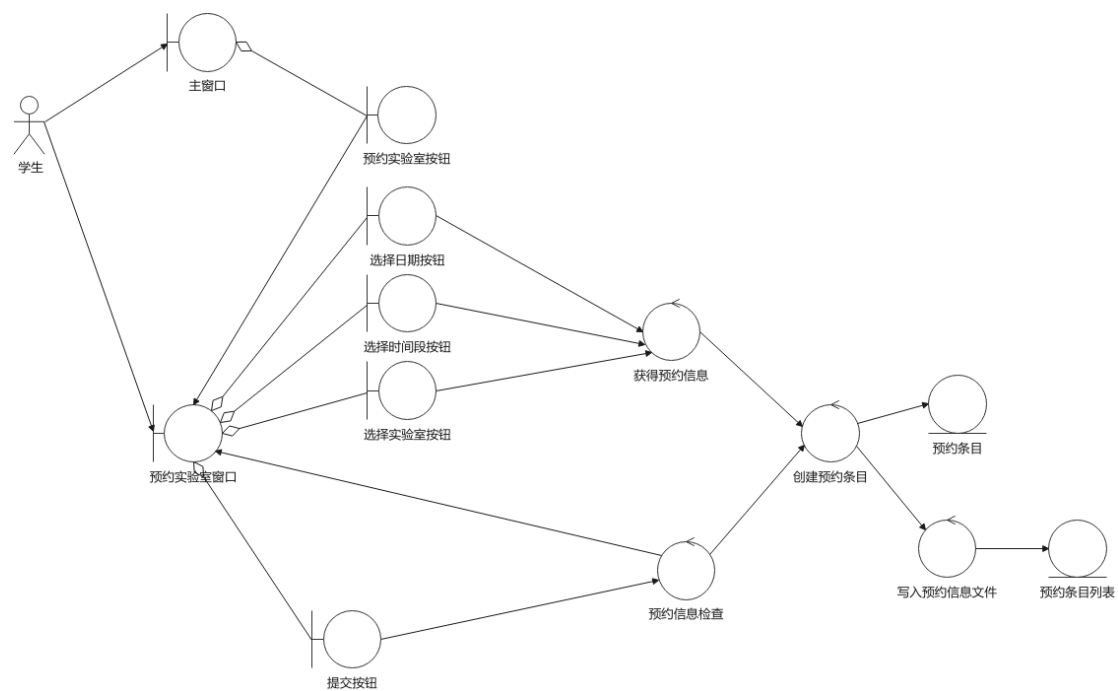
#### 4.2.1 用户登录事件流

事件流	基本	6) 系统展示登录界面 7) 用户选择登录账户类型 8) 用户输入账号密码 9) 系统检查账号密码 10) 进入系统
	拓展	4a) 系统检查账号密码输入是否正确，如果不正确则要求用户重新登录



## 4.2.2 学生预约事件流

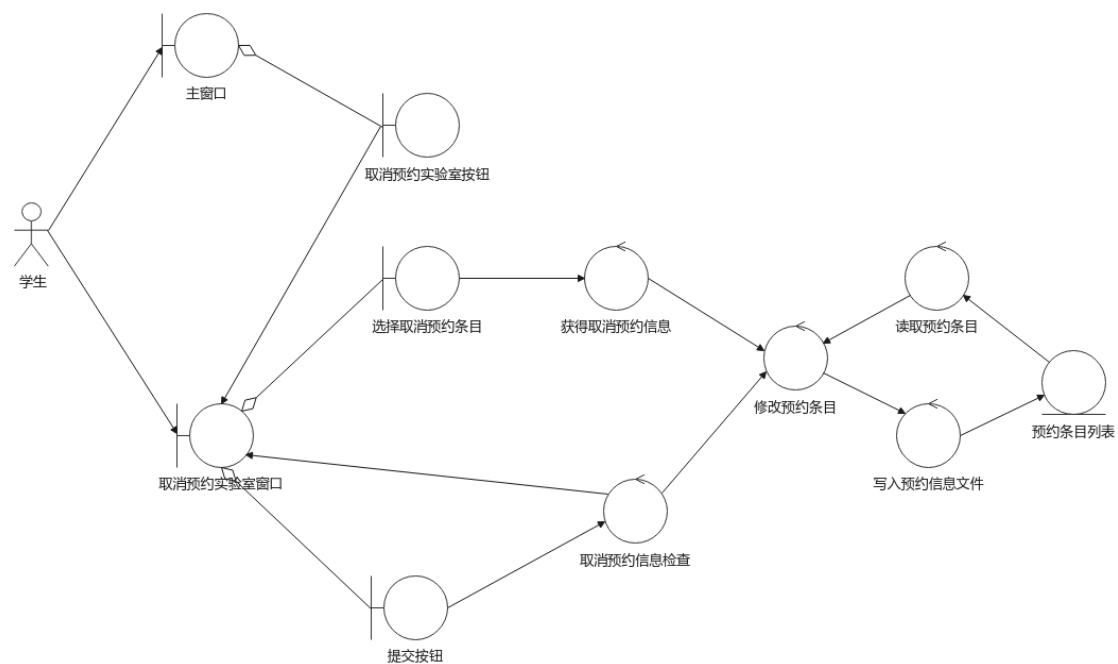
事件流	基本	10) 学生发出预约实验室的请求 11) 系统展示能够预约的时间 12) 学生选择时间 13) 系统确认时间并展示一天内的时间段 14) 学生选择时间段 15) 系统确认时间段并展示可以选择的实验室 16) 学生选择实验室 17) 系统确认实验室 18) 系统记录下预约记录
	拓展	4a) 系统检查时间输入是否正确，如果不正确则要求学生重新输入 6a) 系统检查时间段输入是否正确，如果不正确则要求学生重新输入 8a) 系统检查实验室输入是否正确，如果不正确则要求学生重新输入





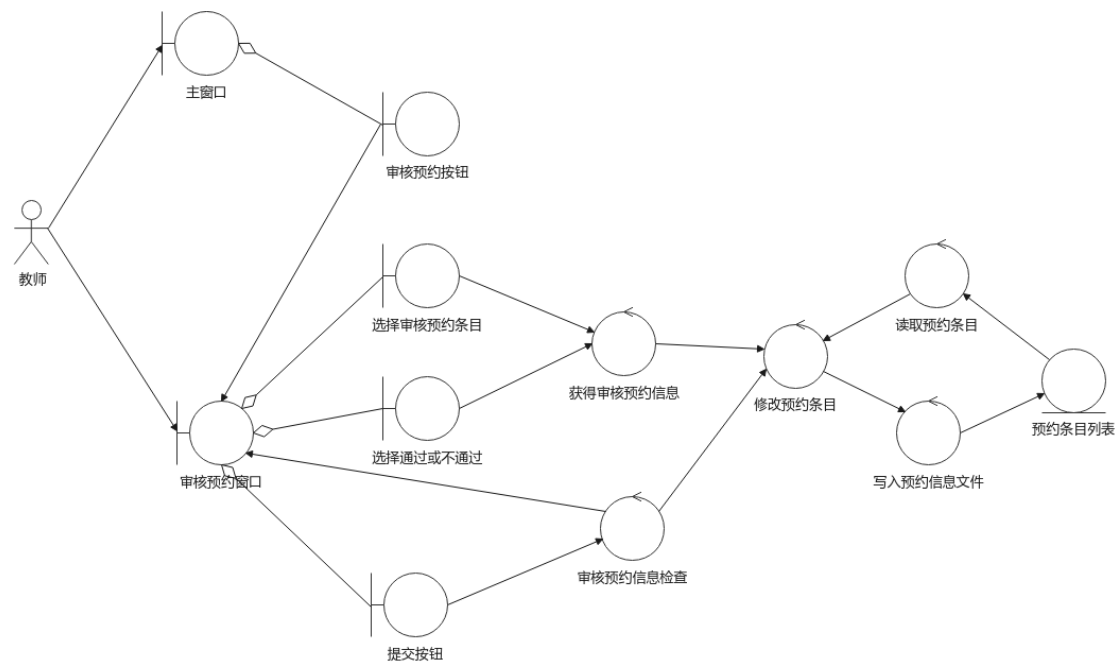
### 4.2.3 学生取消预约事件流

事件流	基本	6) 学生发出取消预约的申请 7) 系统展示学生的预约记录 8) 学生选择一条预约记录进行取消 9) 系统检查预约记录 10) 系统修改预约记录文件，删除相应的条目
	拓展	4a) 系统检查学生输入的条目是否能够取消，如果不能则取消失败



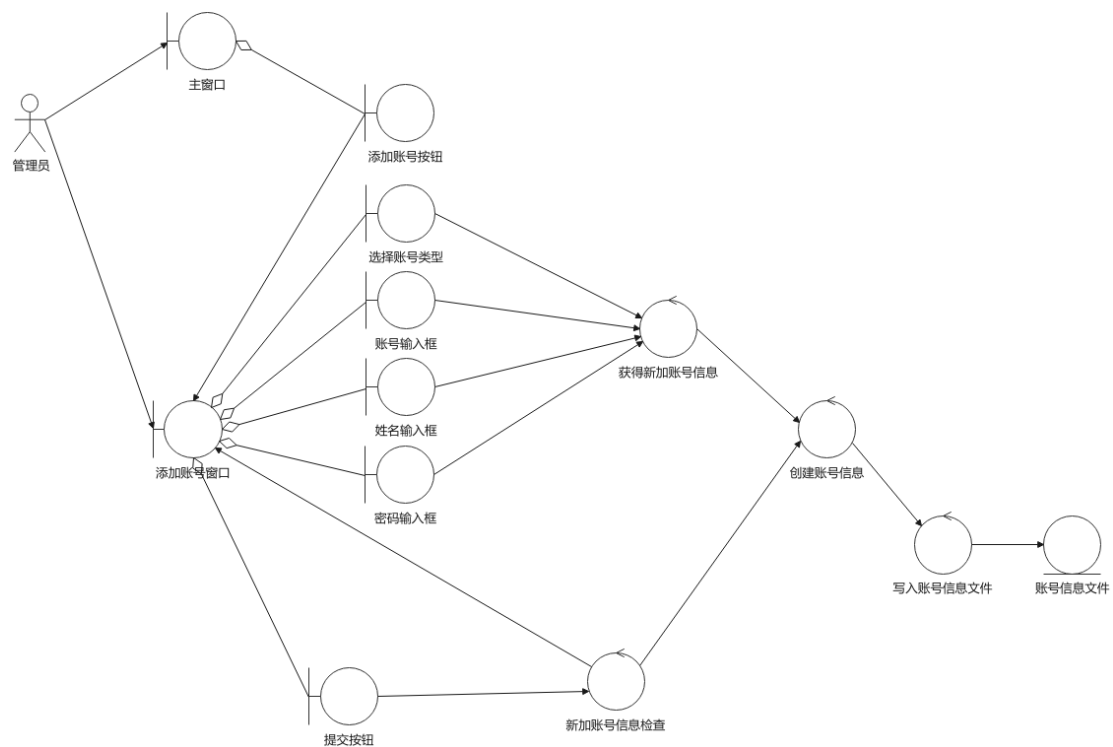
## 4.2.4 教师审核预约事件流

事件流	基本	7) 教师发出审批预约的申请 8) 系统展示待审批的预约条目 9) 教师选择一条待审批的条目 10) 系统检查教师输入的条目 11) 教师选择审批通过或者不通过 12) 系统修改对应的预约条目
	拓展	4a) 系统检查教师输入的条目是否是待审批的，如果不是则要求重新输入



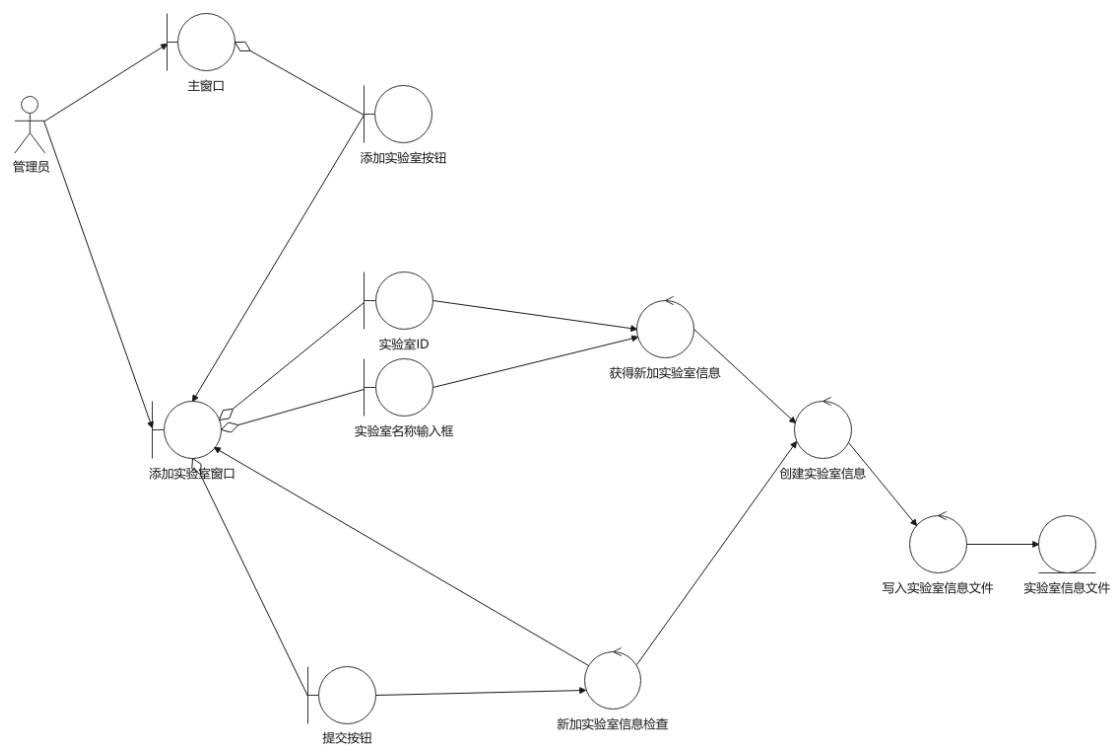
## 4.2.5 管理员添加账号事件流

事件流	基本	5) 系统管理员发出添加账号申请 6) 系统展示菜单 7) 管理员选择添加学生账号还是教师账号 8) 管理员输入 ID、姓名和密码 9) 系统检查 ID 是否有重复 10) 系统将新加的账号加入到相应的账号信息文件中
	拓展	5a) 系统检查加入的账号是否有重复，如果有则添加失败



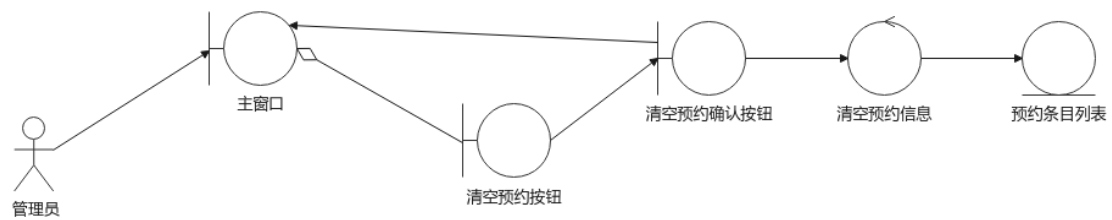
## 4.2.6 管理员添加实验室事件流

事件流	基本	5) 系统管理员发出添加实验室申请 6) 管理员输入实验室 ID 和实验室名称 7) 系统检查实验室 ID 是否有重复 8) 系统将新加的实验室信息加入到实验室信息文件中
	拓展	5a) 系统检查加入的实验室 ID 是否有重复，如果有则添加失败



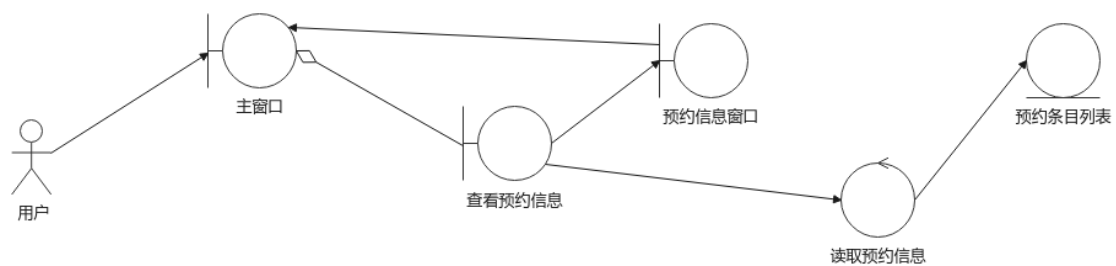
#### 4.2.7 管理员清空预约事件流

事件流	基本	5) 系统管理员提出清空预约记录的申请 6) 系统向管理员确认 7) 管理员确认 8) 系统清空所有预约记录
	拓展	无



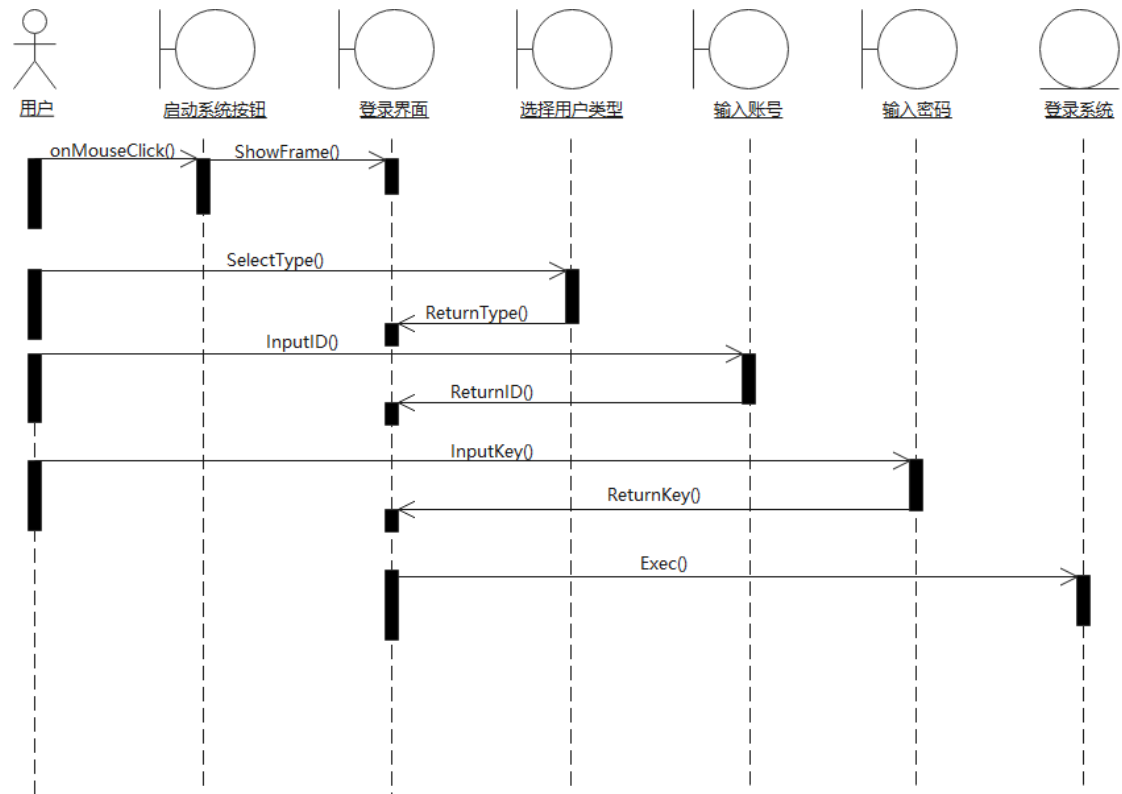
#### 4.2.8 用户查看预约事件流

事件流	基本	3) 用户提出查看预约信息的请求 4) 系统展示预约信息
	拓展	无

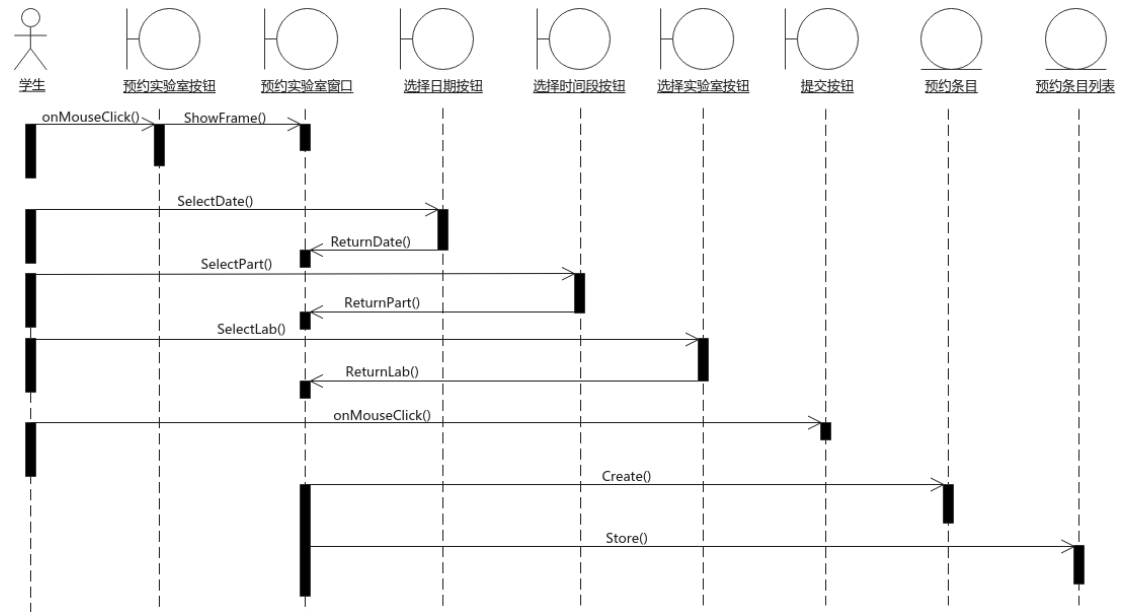


## 5 交互建模

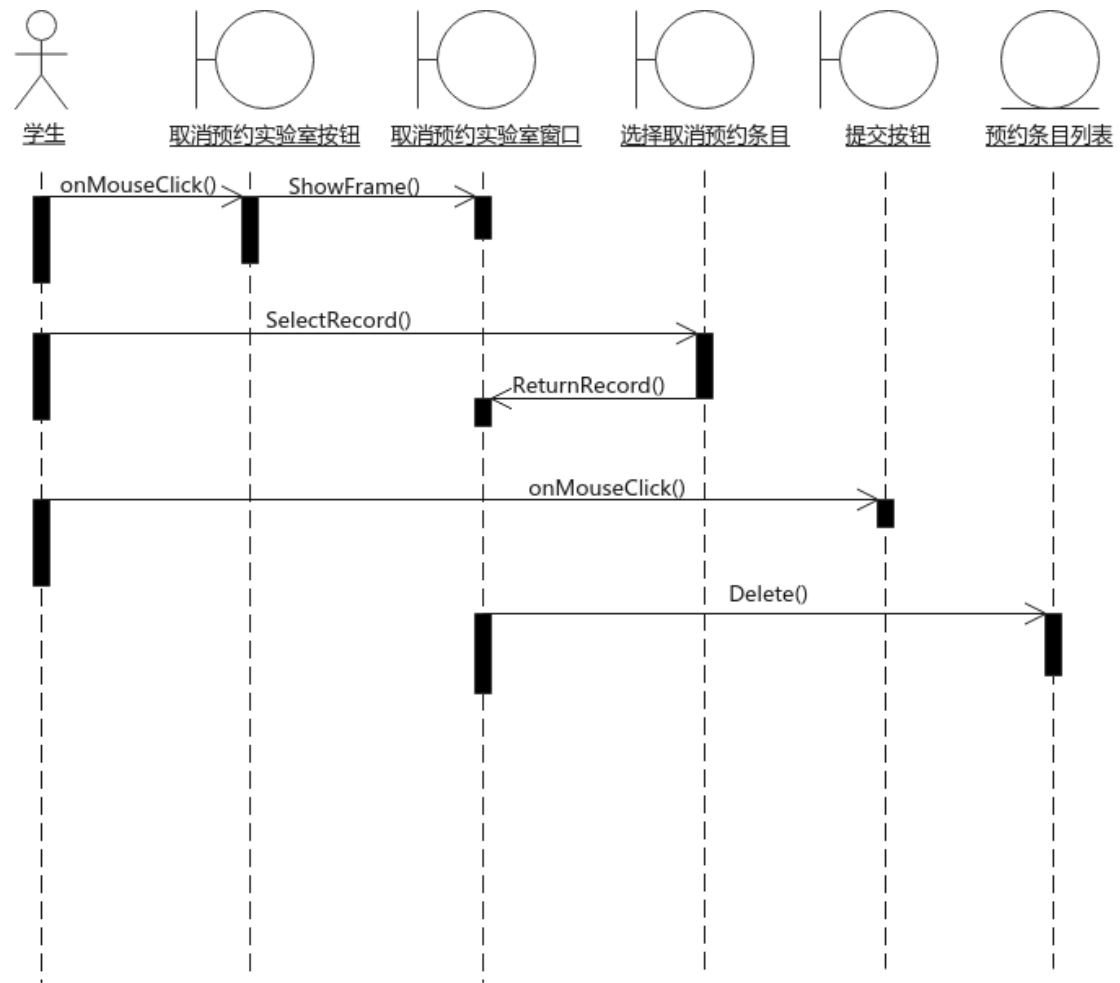
### 5.1 用户登录时序图



## 5.2 学生预约时序图

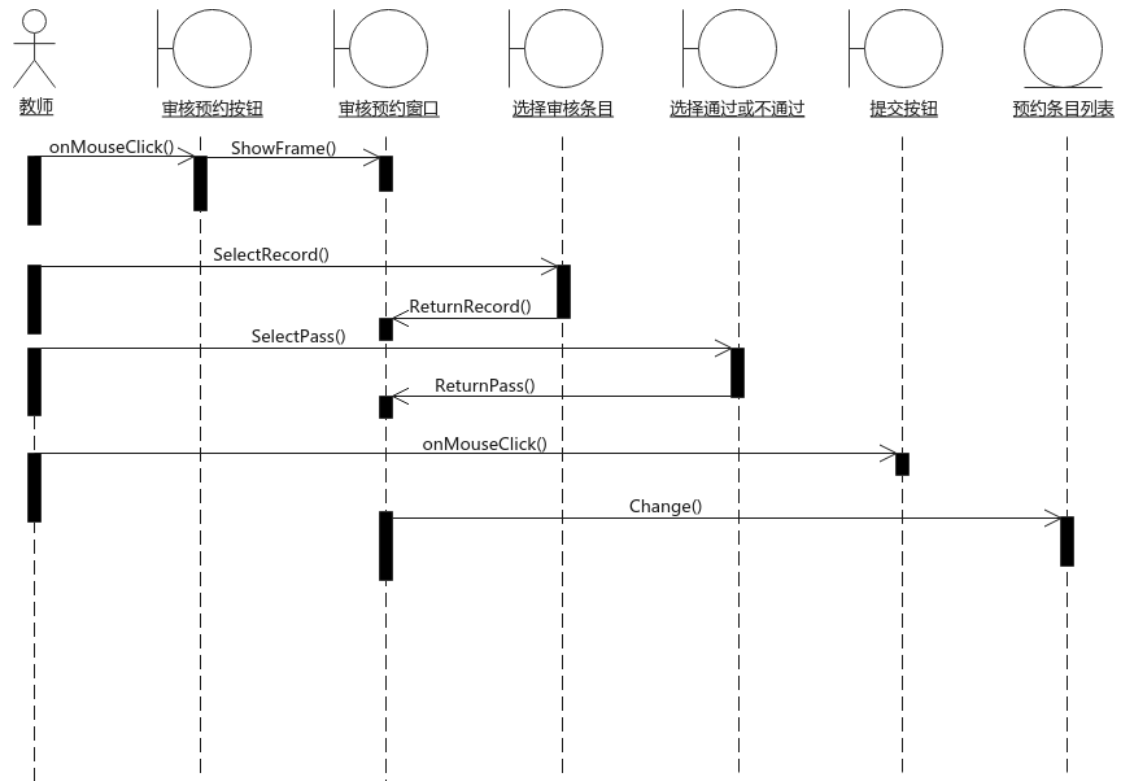


### 5.3 学生取消预约时序图

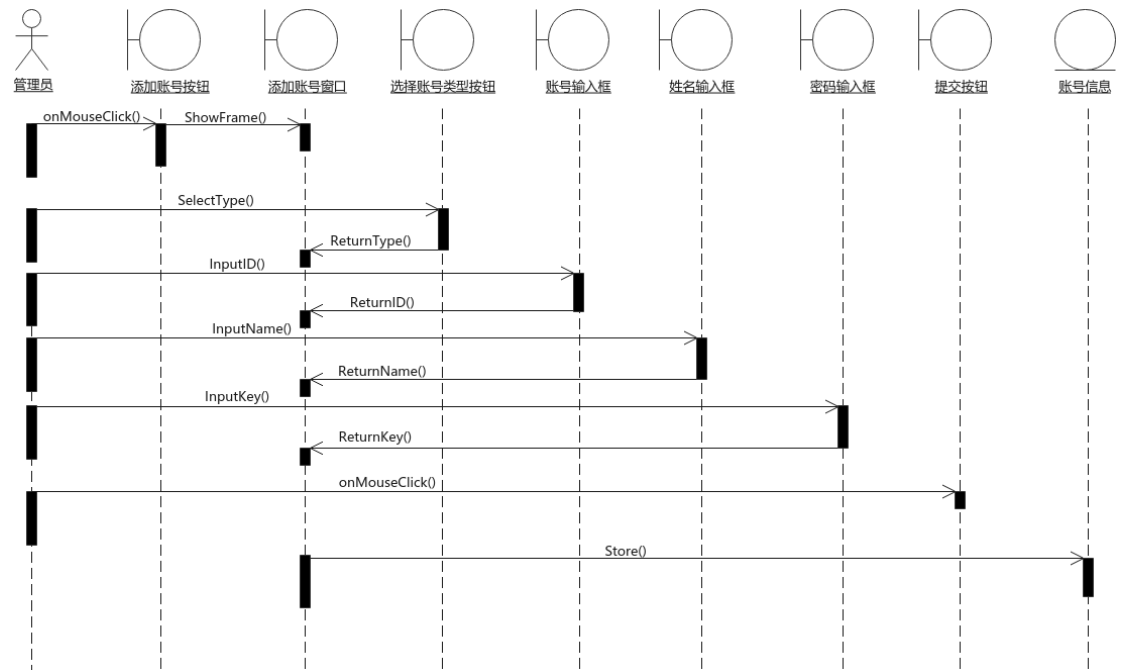




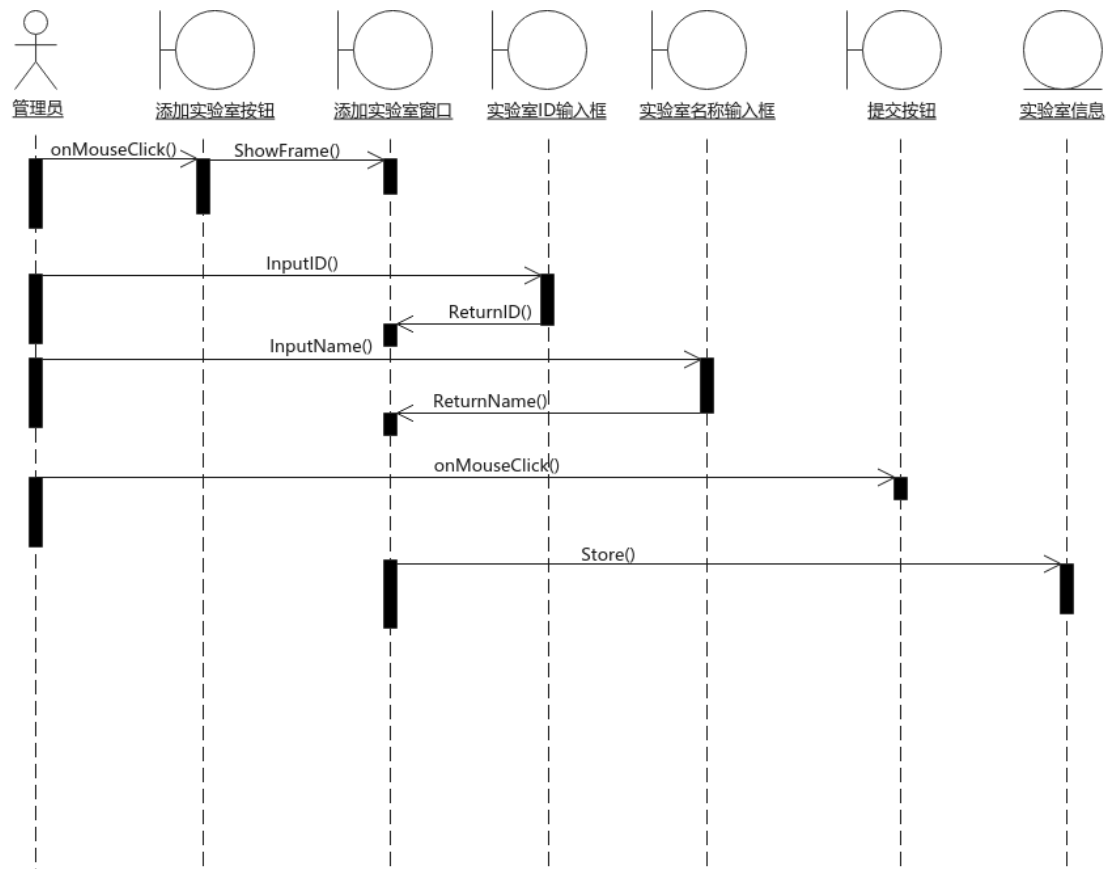
## 5.4 教师审核预约时序图



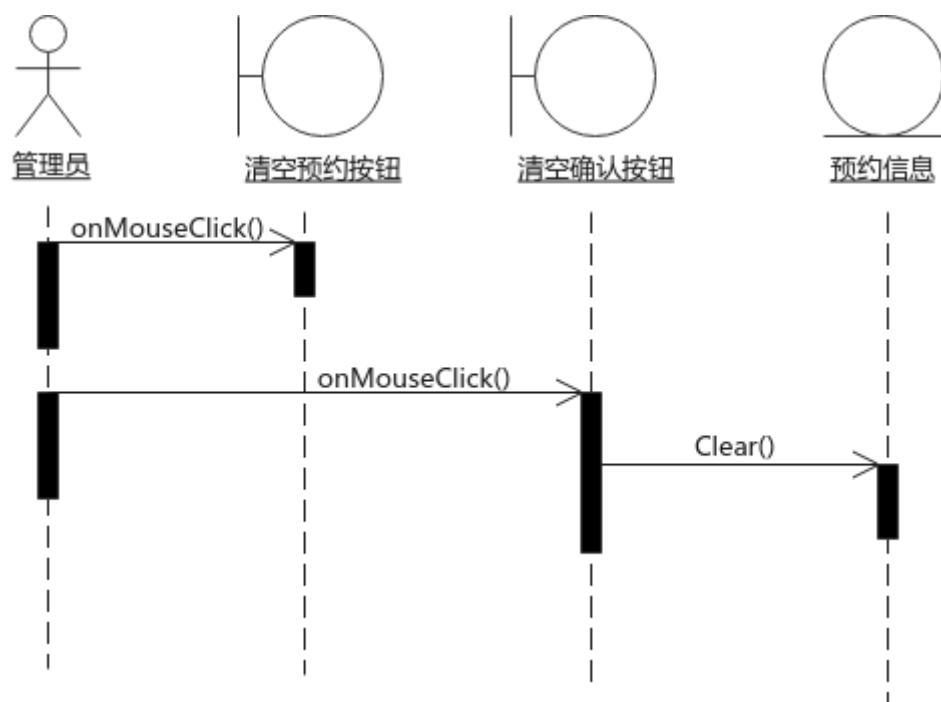
## 5.5 管理员添加账号时序图



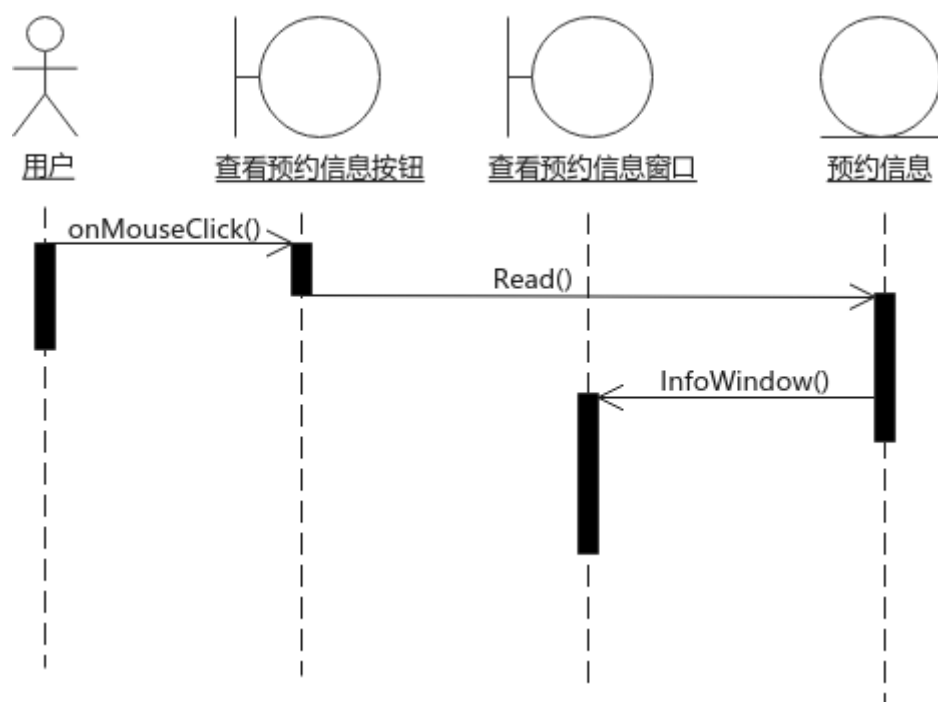
## 5.6 管理员添加实验室时序图



## 5.7 管理员清空预约时序图

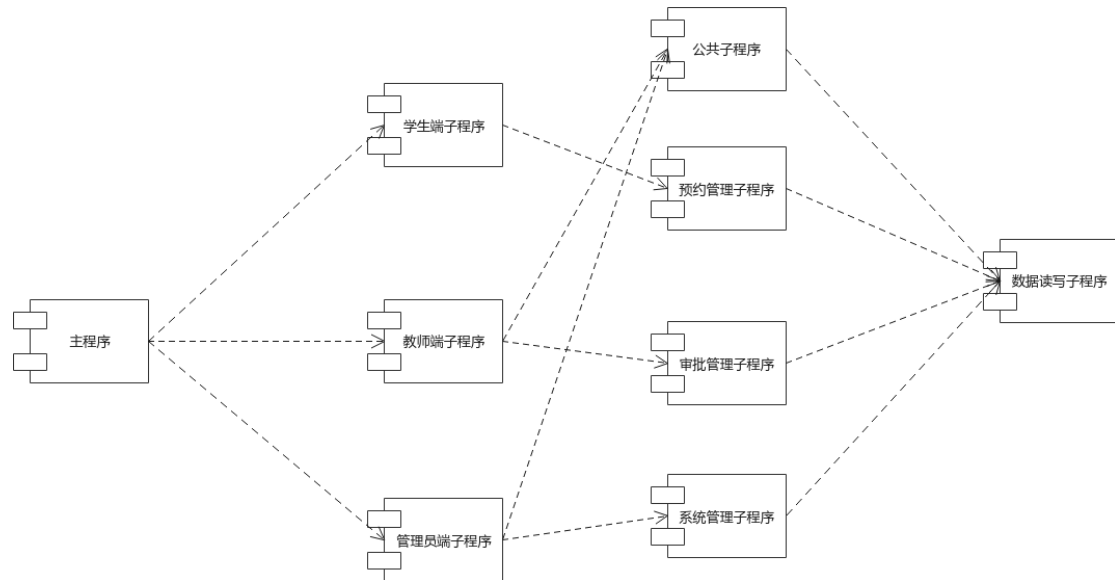


## 5.8 用户查看预约时序图



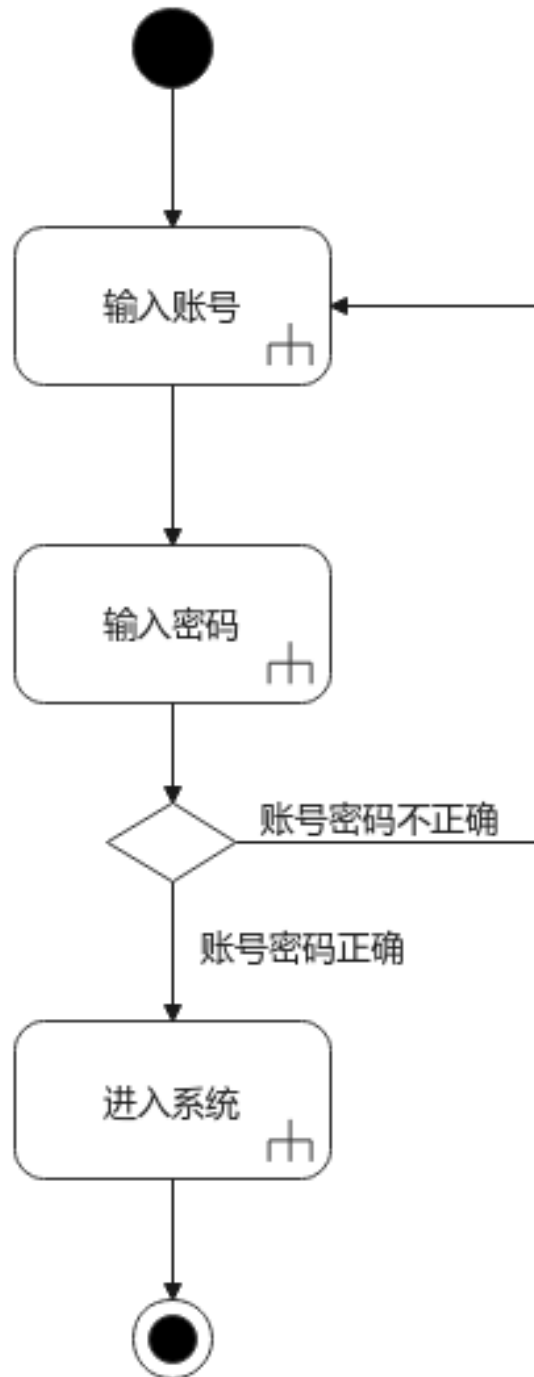
## 6 构件与活动图

### 6.1 构件图

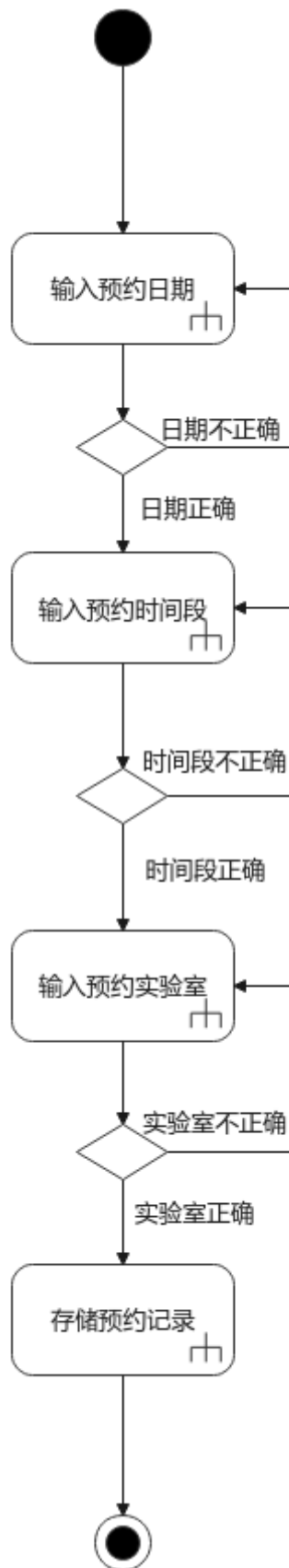


## 6.2 活动图

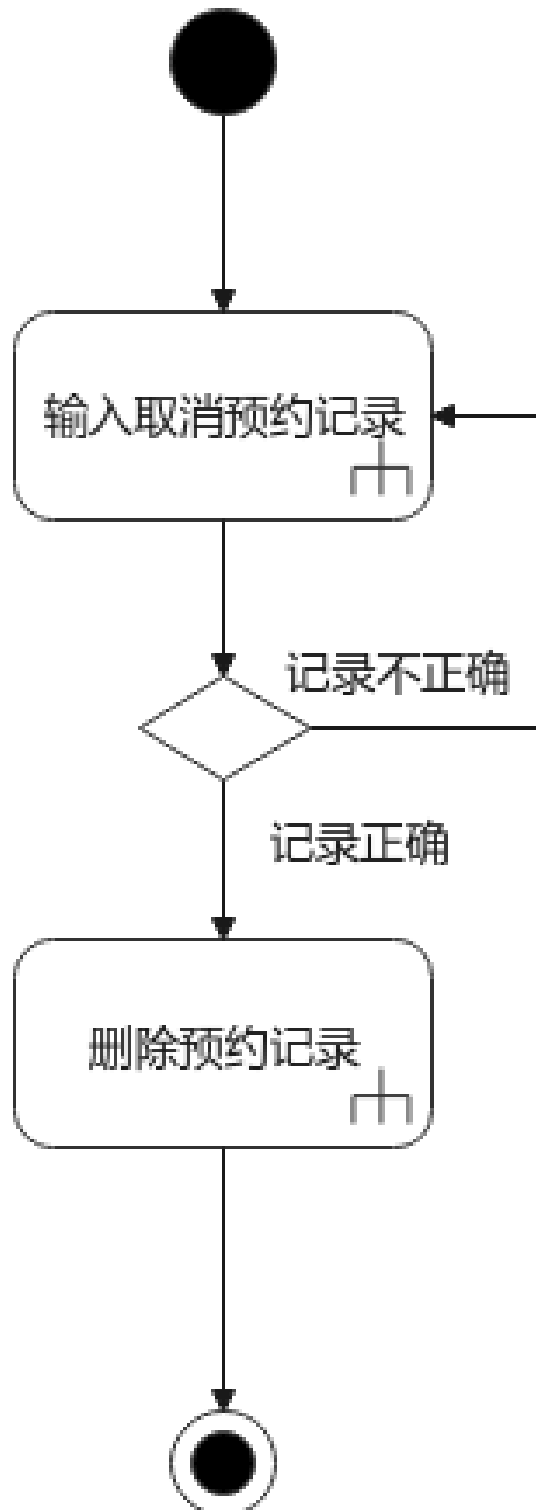
### 6.2.1 UC01 活动图



## 6.2.2UC02 活动图

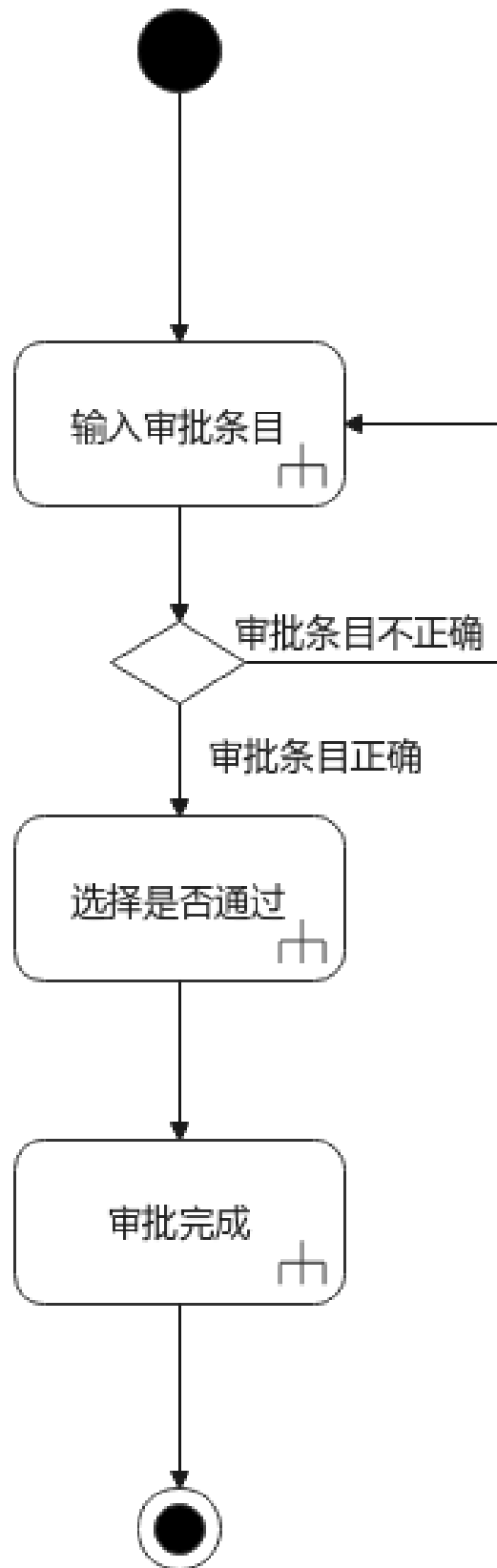


### 6.2.3 UC03 活动图

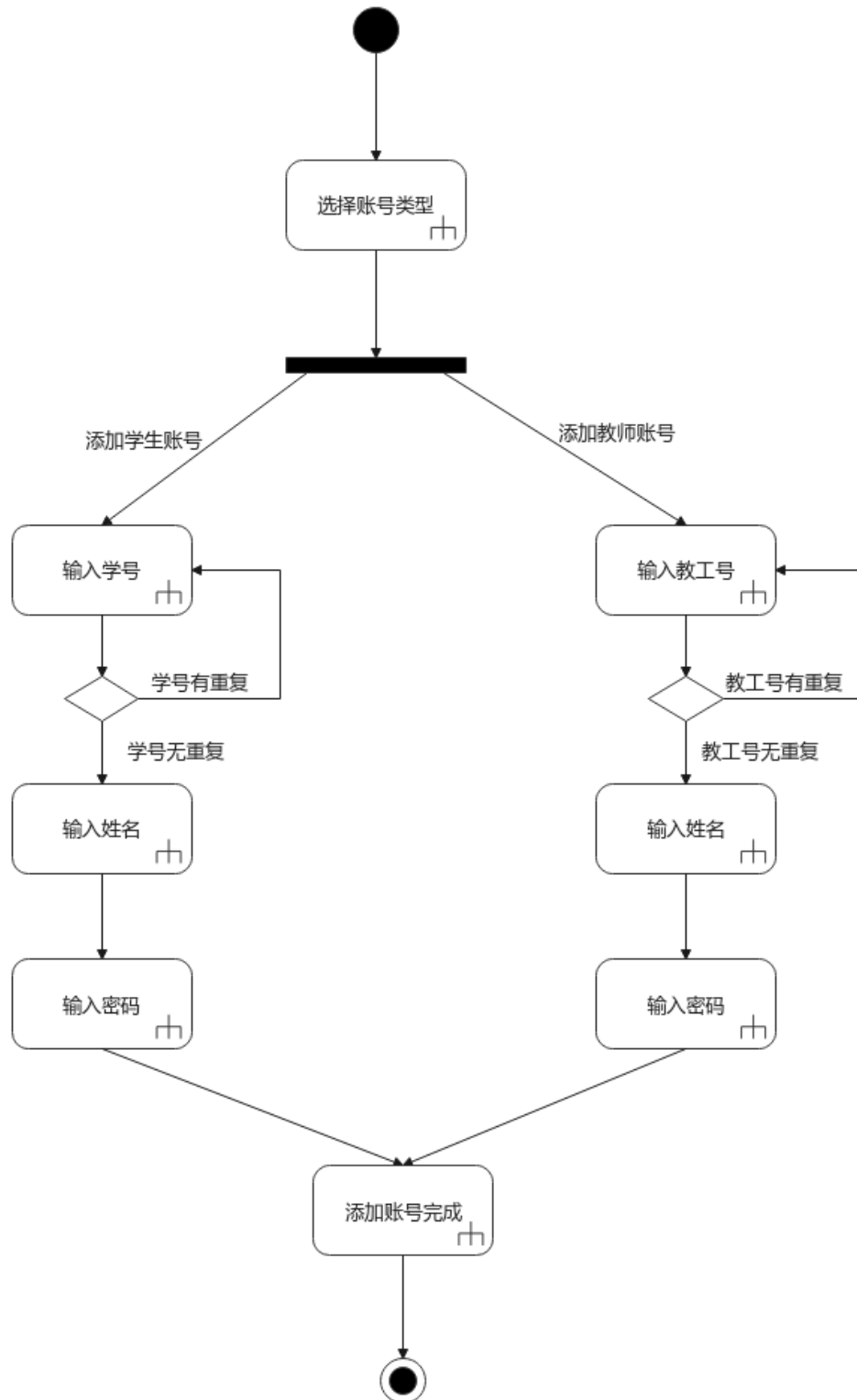




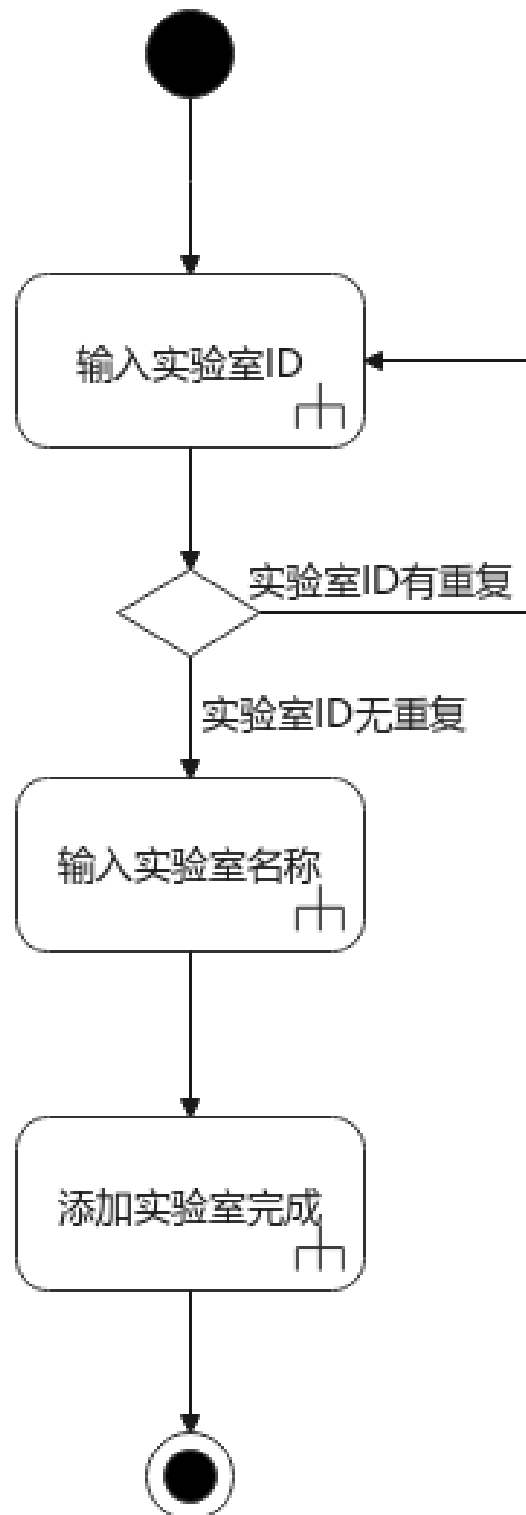
#### 6.2.4UC04 活动图



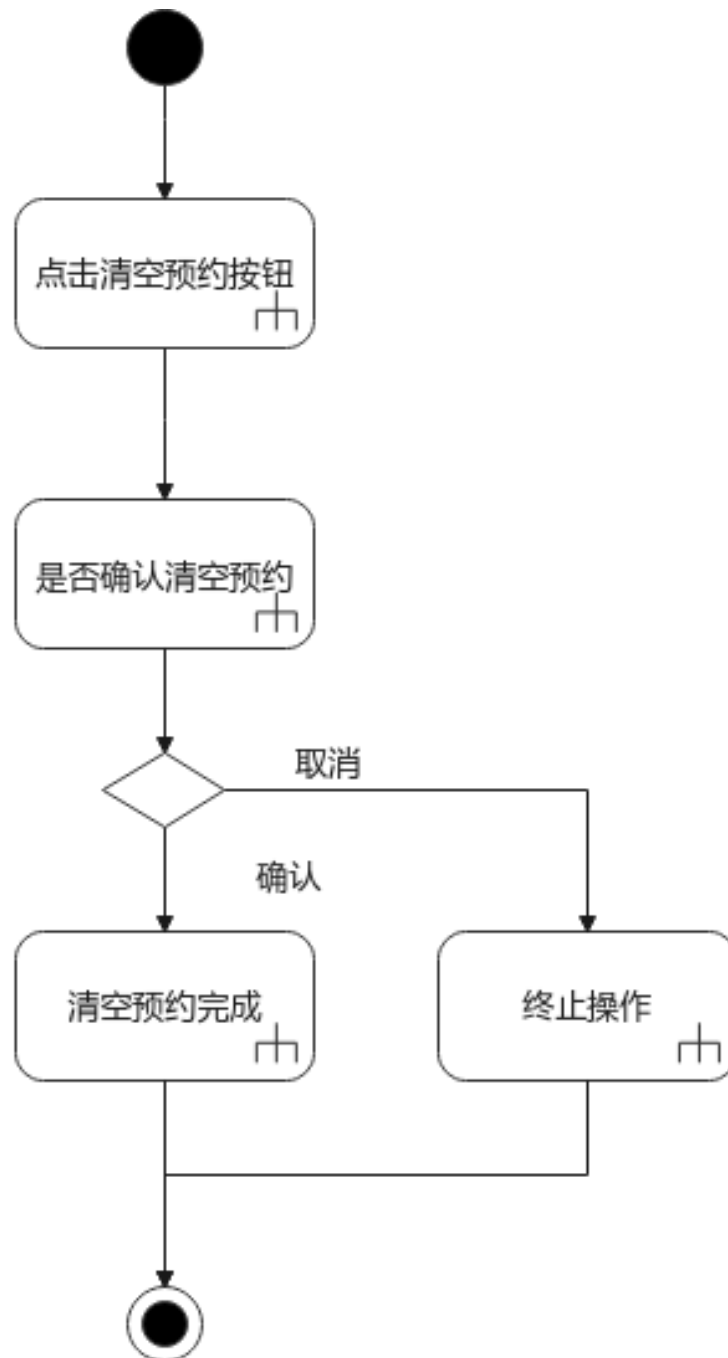
## 6.2.5UC05 活动图



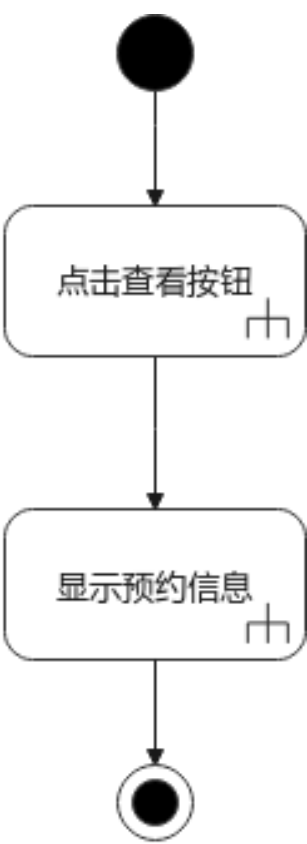
### 6.2.6 UC06 活动图



## 6.2.7UC07 活动图



6.2.8UC08 活动图



7 软件测试

7.1 黑盒测试

7.1.1 学生登录模块测试

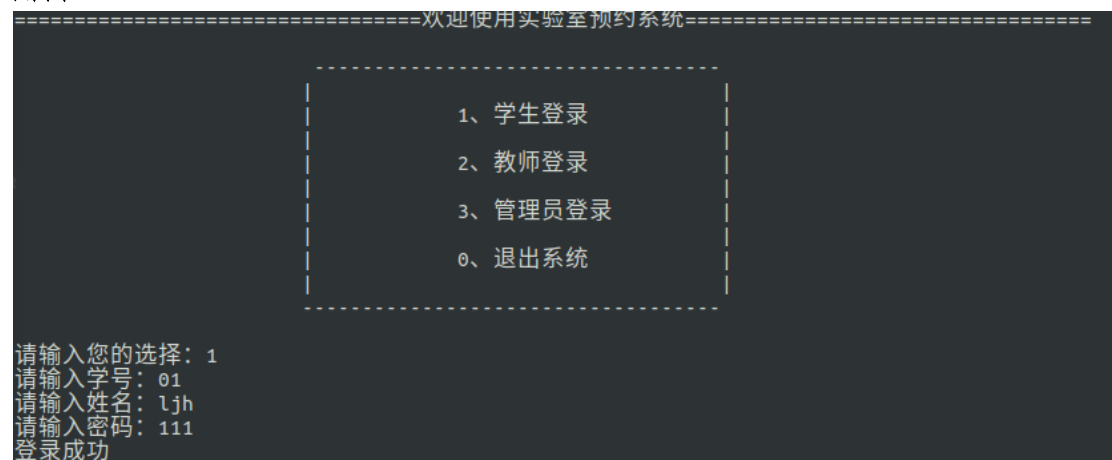
输入数据	有效等价类	无效等价类
学号	1 位于账号信息内	2 不位于账号信息内
姓名	3 与学号信息对应	4 不位于账号信息内 5 位于账号信息内但与学号不对应
密码	6 与学号信息对应	7 不位于账号信息内 8 位于账号信息内但与学号不对应

## 设计用例

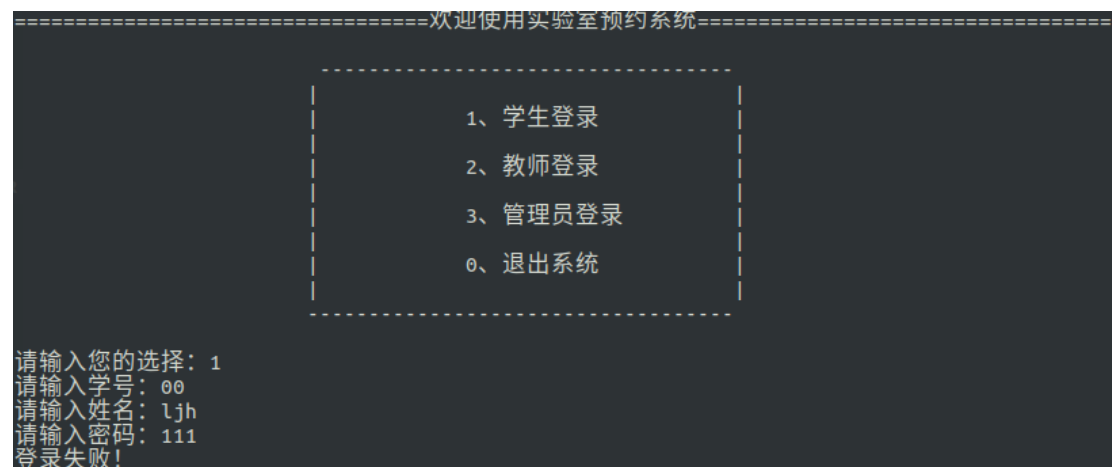
用例编号	用例	期望结果	覆盖条件	实际结果
1	(01, ljh, 111)	有效	136	有效
2	(00, ljh, 111)	无效	2	无效
3	(01, 111, 111)	无效	4	无效
4	(01, lfy, 111)	无效	5	无效
5	(01, ljh, 123)	无效	7	无效
6	(01, ljh, 222)	无效	8	无效

## 测试用例结果截图

### 用例 1



### 用例 2



用例 3

```
=====欢迎使用实验室预约系统=====

      1、学生登录
      2、教师登录
      3、管理员登录
      0、退出系统

请输入您的选择：1
请输入学号：01
请输入姓名：lll
请输入密码：111
登录失败！

=====欢迎使用实验室预约系统=====
```

用例 4

```
=====欢迎使用实验室预约系统=====

      1、学生登录
      2、教师登录
      3、管理员登录
      0、退出系统

请输入您的选择：1
请输入学号：01
请输入姓名：lfy
请输入密码：111
登录失败！

=====欢迎使用实验室预约系统=====
```

用例 5

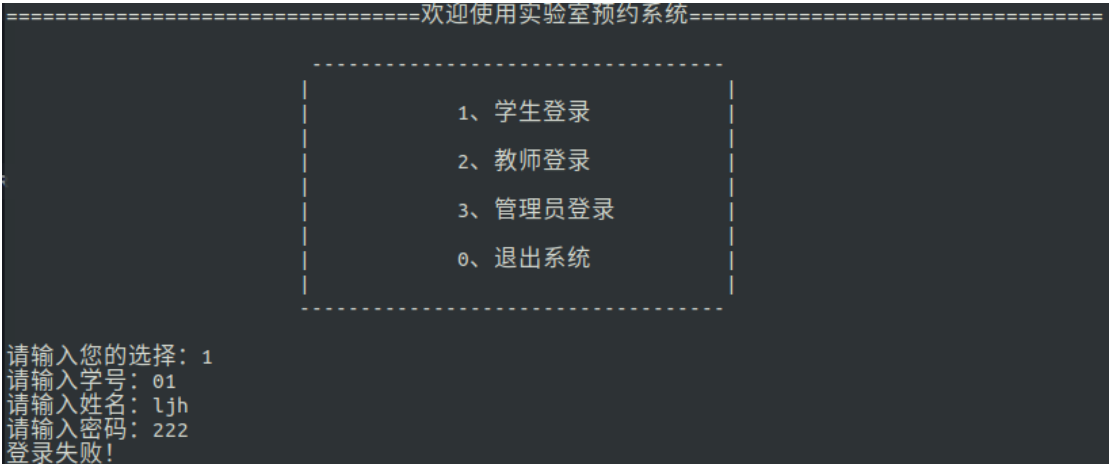
```
=====欢迎使用实验室预约系统=====

      1、学生登录
      2、教师登录
      3、管理员登录
      0、退出系统

请输入您的选择：1
请输入学号：01
请输入姓名：ljh
请输入密码：123
登录失败！

=====欢迎使用实验室预约系统=====
```

用例 6



7.1.2 教师登录模块测试

输入数据	有效等价类	无效等价类
教工号	1 位于账号信息内	2 不位于账号信息内
姓名	3 与教工号信息对应	4 不位于账号信息内 5 位于账号信息内但与教工号不对应
密码	6 与教工号信息对应	7 不位于账号信息内 8 位于账号信息内但与教工号不对应

设计用例

用例编号	用例	期望结果	覆盖条件	实际结果
1	(01, t1, 111)	有效	136	有效
2	(00, t1, 111)	无效	2	无效
3	(01, tt, 111)	无效	4	无效
4	(01, t2, 111)	无效	5	无效
5	(01, t1, 123)	无效	7	无效
6	(01, t1, 222)	无效	8	无效



### 用例 1

用例 4

```
=====欢迎使用实验室预约系统=====
1、学生登录
2、教师登录
3、管理员登录
0、退出系统

请输入您的选择：2
请输入职工号：01
请输入姓名：t2
请输入密码：111
登录失败!
```

用例 5

```
=====欢迎使用实验室预约系统=====
1、学生登录
2、教师登录
3、管理员登录
0、退出系统

请输入您的选择：2
请输入职工号：01
请输入姓名：t1
请输入密码：123
登录失败!
```

用例 6

```
=====欢迎使用实验室预约系统=====
1、学生登录
2、教师登录
3、管理员登录
0、退出系统

请输入您的选择：2
请输入职工号：01
请输入姓名：t1
请输入密码：222
登录失败!
```

### 7.1.3 管理员登录模块测试

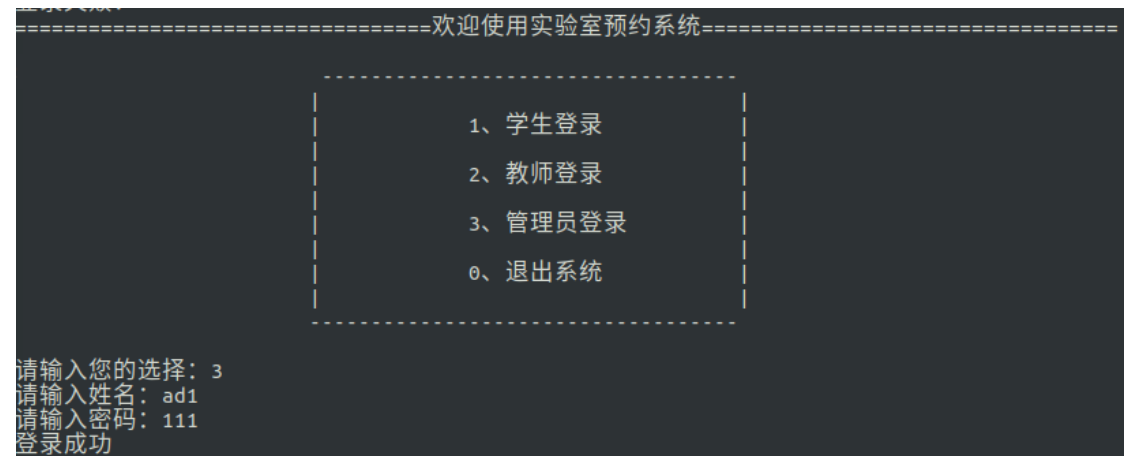
输入数据	有效等价类	无效等价类
账号	1 位于账号信息内	2 不位于账号信息内
密码	3 与账号信息对应	4 不位于账号信息内 5 位于账号信息内但与账号不对应

#### 设计用例

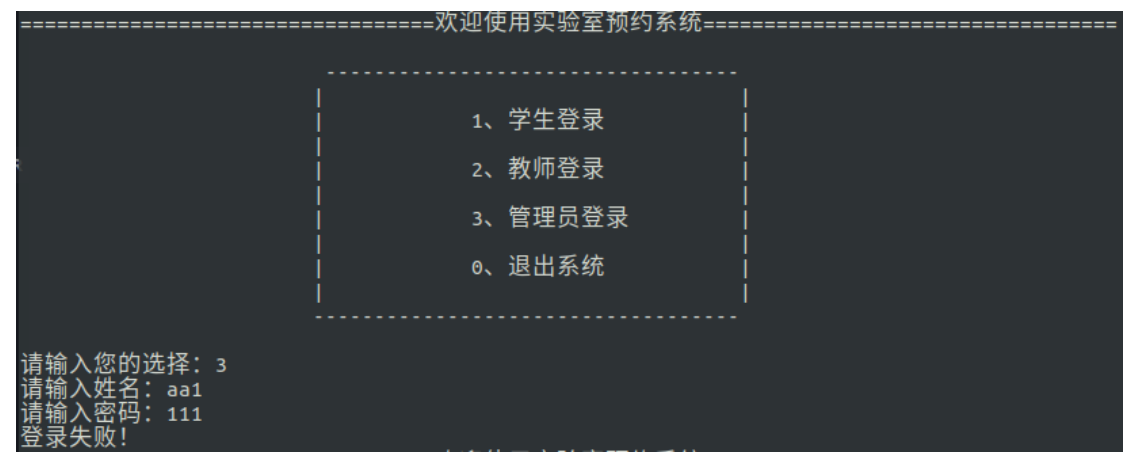
用例编号	用例	期望结果	覆盖条件	实际结果
1	(ad1, 111)	有效	13	有效
2	(aa1, 111)	无效	2	无效
3	(ad1, 123)	无效	4	无效
4	(ad1, 222)	无效	5	无效

#### 测试用例结果截图

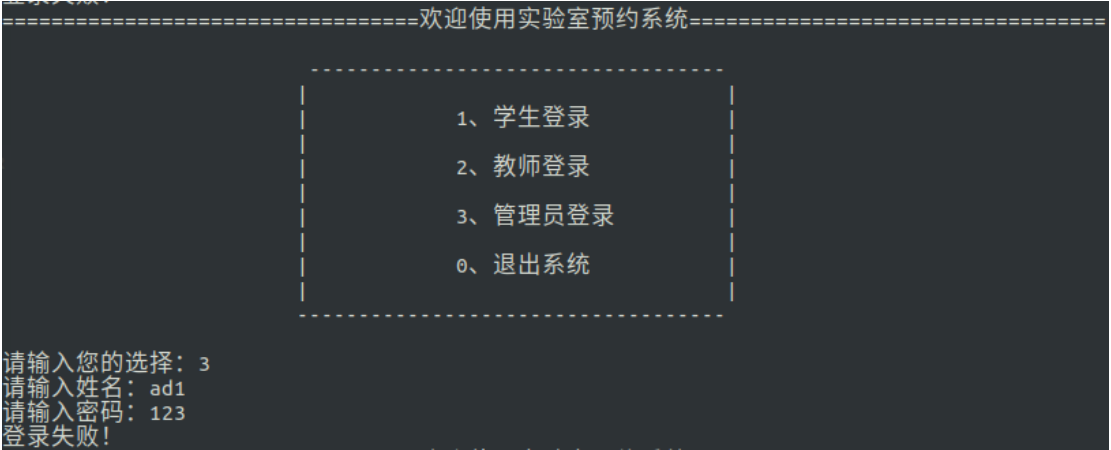
##### 用例 1



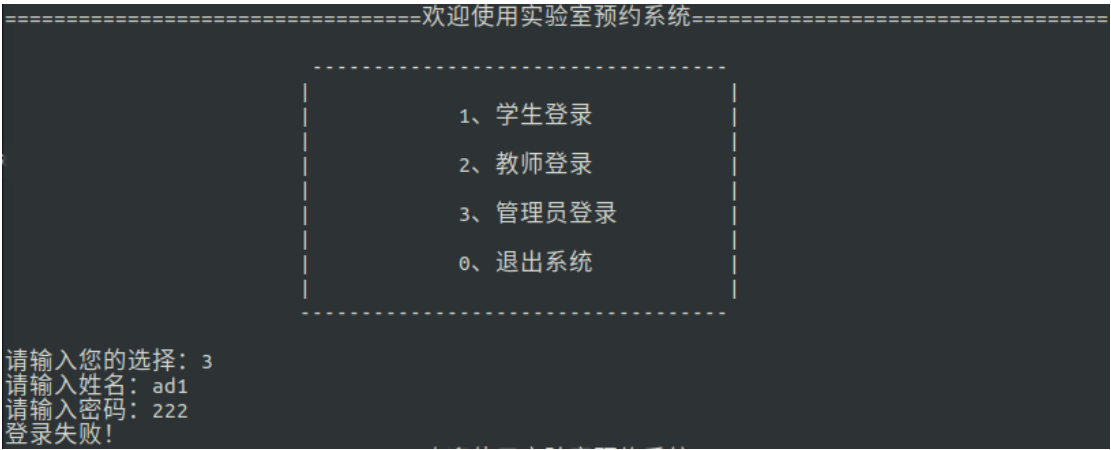
##### 用例 2



用例 3



用例 4



7.1.4 预约模块测试

输入数据	有效等价类	无效等价类
日期	1 [1, 5]	2 <1 3 >5
时间段	4 1 或 2	5 <1 6 >2
实验室	7 位于实验室信息内	8 不位于实验室信息内

## 设计用例

用例编号	用例	期望结果	覆盖条件	实际结果
1	(3, 1, 3)	有效	147	有效
2	(0, 1, 3)	无效	2	无效
3	(6, 1, 3)	无效	3	无效
4	(3, 0, 3)	无效	5	无效
5	(3, 3, 3)	无效	6	无效
6	(3, 1, 10)	无效	8	无效

## 测试用例结果截图

### 用例 1

```

实验室开放时间为每周一到周五
1、周一
2、周二
3、周三
4、周四
5、周五
您想预约到周几？
3
您预约的时间段？
1、上午
2、下午
1
实验室信息如下：

实验室编号      实验室名称
1                Physics_lab
2                Chemistry_lab
3                Biology_lab
4                cs_lab

您预约的实验室？
3
申请成功，审核中！

```

### 用例 2

```

实验室开放时间为每周一到周五
1、周一
2、周二
3、周三
4、周四
5、周五
您想预约到周几？
0
请重新选择

```

### 用例 3

实验室开放时间为每周一到周五

- 1、周一
- 2、周二
- 3、周三
- 4、周四
- 5、周五

您想预约到周几？

6

请重新选择

### 用例 4

实验室开放时间为每周一到周五

- 1、周一
- 2、周二
- 3、周三
- 4、周四
- 5、周五

您想预约到周几？

3

您预约的时间段？

- 1、上午
- 2、下午

0

请重新选择

### 用例 5

实验室开放时间为每周一到周五

- 1、周一
- 2、周二
- 3、周三
- 4、周四
- 5、周五

您想预约到周几？

3

您预约的时间段？

- 1、上午
- 2、下午

3

请重新选择

用例 6

```
实验室开放时间为每周一到周五
1、周一
2、周二
3、周三
4、周四
5、周五
您想预约到周几？
3
您预约的时间段？
1、上午
2、下午
1
实验室信息如下：
实验室编号      实验室名称
1                Physics_lab
2                Chemistry_lab
3                Biology_lab
4                cs_lab

您预约的实验室？
10
请重新选择
```

7.1.5 审批模块测试

输入数据	有效等价类	无效等价类
审核预约条目 ID	1 位于待预约条目内	2 不位于待预约条目内
审核后状态	3 1 或 2	4 <1 5 >2

设计用例

用例编号	用例	期望结果	覆盖条件	实际结果
1	(4, 1)	有效	13	有效
2	(10, 1)	无效	2	无效
3	(8, 0)	无效	4	无效
4	(8, 3)	无效	5	无效

测试用例结果截图

用例 1

您需要审核的预约如下：

序号	日期	时间段	实验室	学号	姓名	审核	状态
4	周一	上午	物理实验室	1	ljh	审核中	
6	周一	上午	物理实验室	1	ljh	审核中	
7	周三	上午	物理实验室	1	ljh	审核中	
8	周三	下午	化学实验室	1	ljh	审核中	
9	周三	下午	物理实验室	1	ljh	审核中	

请选择要审核预约的序号：4

4

1、通过      2、不通过

1

审核成功

用例 2

您需要审核的预约如下：

序号	日期	时间段	实验室	学号	姓名	审核	状态
6	周一	上午	物理实验室	1	ljh	审核中	
7	周三	上午	物理实验室	1	ljh	审核中	
8	周三	下午	化学实验室	1	ljh	审核中	
9	周三	下午	物理实验室	1	ljh	审核中	

请选择要审核预约的序号：10

输入有误，请重新选择

用例 3

您需要审核的预约如下：

序号	日期	时间段	实验室	学号	姓名	审核	状态
6	周一	上午	物理实验室	1	ljh	审核中	
8	周三	下午	化学实验室	1	ljh	审核中	
9	周三	下午	物理实验室	1	ljh	审核中	

请选择要审核预约的序号：8

8

1、通过      2、不通过

0

输入有误，请重新选择

用例 4

您需要审核的预约如下：

序号	日期	时间段	实验室	学号	姓名	审核	状态
6	周一	上午	物理实验室	1	ljh	审核中	
8	周三	下午	化学实验室	1	ljh	审核中	
9	周三	下午	物理实验室	1	ljh	审核中	

请选择要审核预约的序号：8

8

1、通过      2、不通过

3

输入有误，请重新选择



### 7.1.6 添加学生账号模块测试

输入数据	有效等价类	无效等价类
学号	1 不重复	2 重复
姓名	3 任何均有效	
密码	4 任何均有效	

#### 设计用例

用例编号	用例	期望结果	覆盖条件	实际结果
1	(11, 111, 555)	有效	134	有效
2	(01, 111, 555)	无效	2	无效

#### 测试用例结果截图

##### 用例 1

```
请输入选项: 1
请输入学号: 11
请输入姓名: lll
请输入密码: 555
添加成功
```

##### 用例 2

```
请输入选项: 1
请输入学号: 01
学号重复,请重新输入
```

### 7.1.7 添加教师账号模块测试

输入数据	有效等价类	无效等价类
学号	1 不重复	2 重复
姓名	3 任何均有效	
密码	4 任何均有效	

#### 设计用例

用例编号	用例	期望结果	覆盖条件	实际结果
1	(11, t3, 555)	有效	134	有效
2	(01, t3, 555)	无效	2	无效

#### 测试用例结果截图

用例 1

```
请输入职工号： 11
请输入姓名： t3
请输入密码： 555
添加成功
```

用例 2

```
请输入职工号： 01
职工号重复，请重新输入
```

7.1.8 添加实验室模块测试

输入数据	有效等价类	无效等价类
实验室编号	1 不重复	2 重复
实验室名称	3 任何均有效	

设计用例

用例编号	用例	期望结果	覆盖条件	实际结果
1	(11, lab11)	有效	13	有效
2	(1, lab11)	无效	2	无效

测试用例结果截图

用例 1

```
请输入实验室编号： 11
请输入实验室名称： lab11
添加成功
```

用例 2

```
请输入实验室编号： 1
实验室编号重复，请重新输入
```

## 7.2 白盒测试

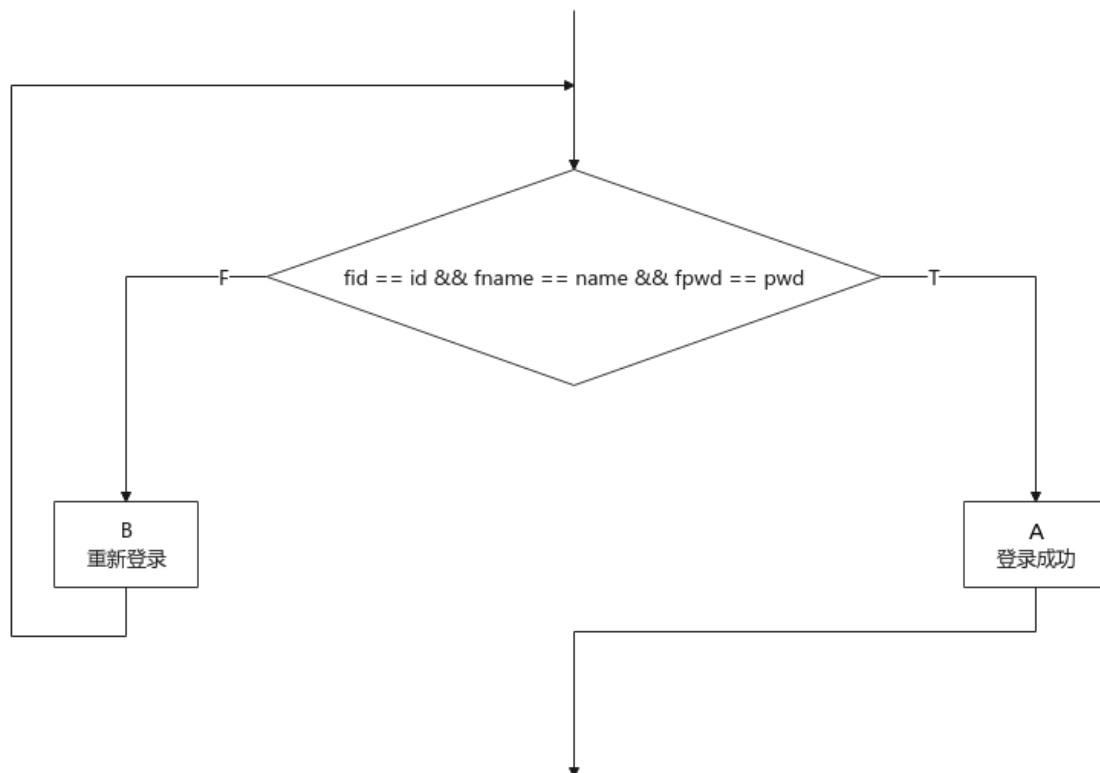
### 7.2.1 学生登录模块测试

代码块

```
if (fid == id && fname == name && fpwd == pwd)
{
    father = new Student(id, name, pwd);
    cout << "登录成功" << endl;
    ifs.close();

    this->student_work(father);
    return;
}
```

流程图

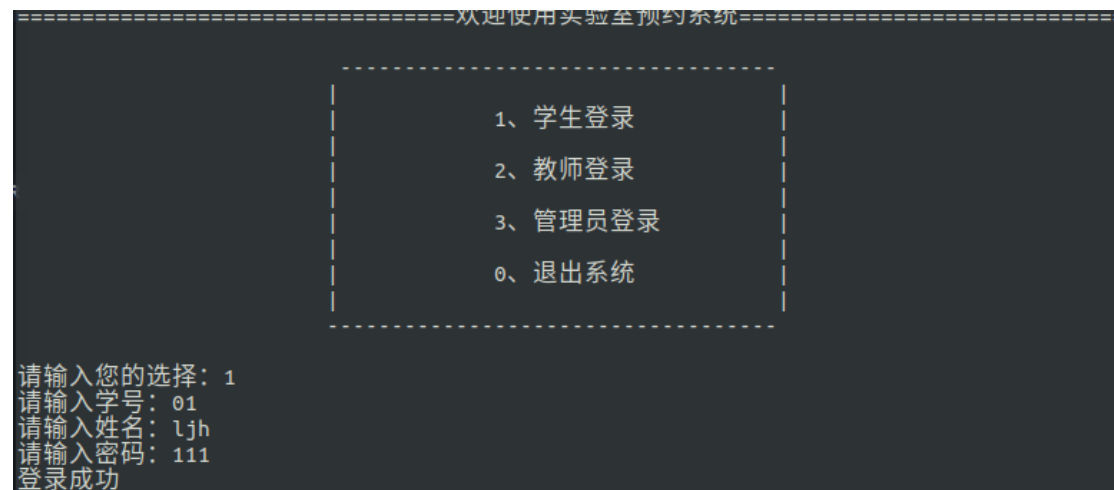


### 语句覆盖测试

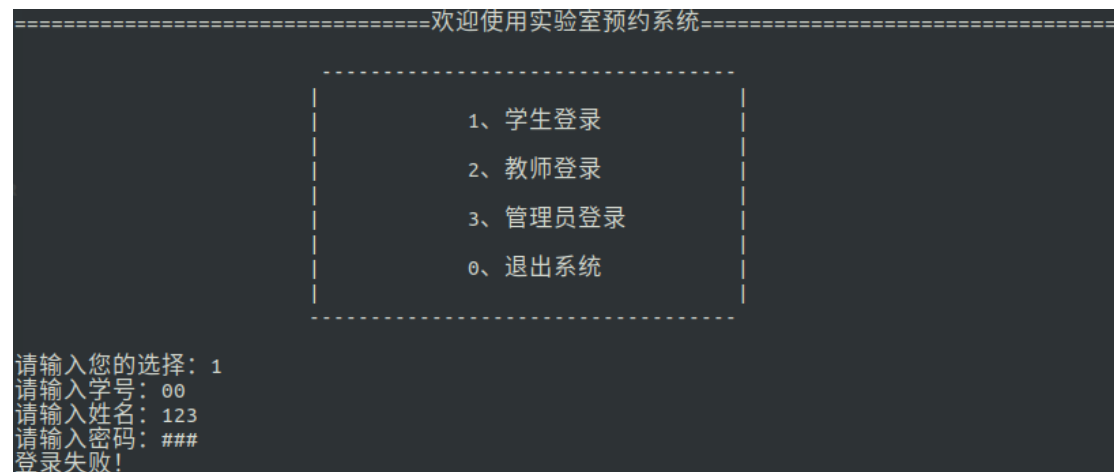
用例编号	用例	期望语句	实际语句
1	(01, ljh, 111)	A	A
2	(00, 123, ###)	B	B

### 测试用例结果截图

#### 用例 1



#### 用例 2



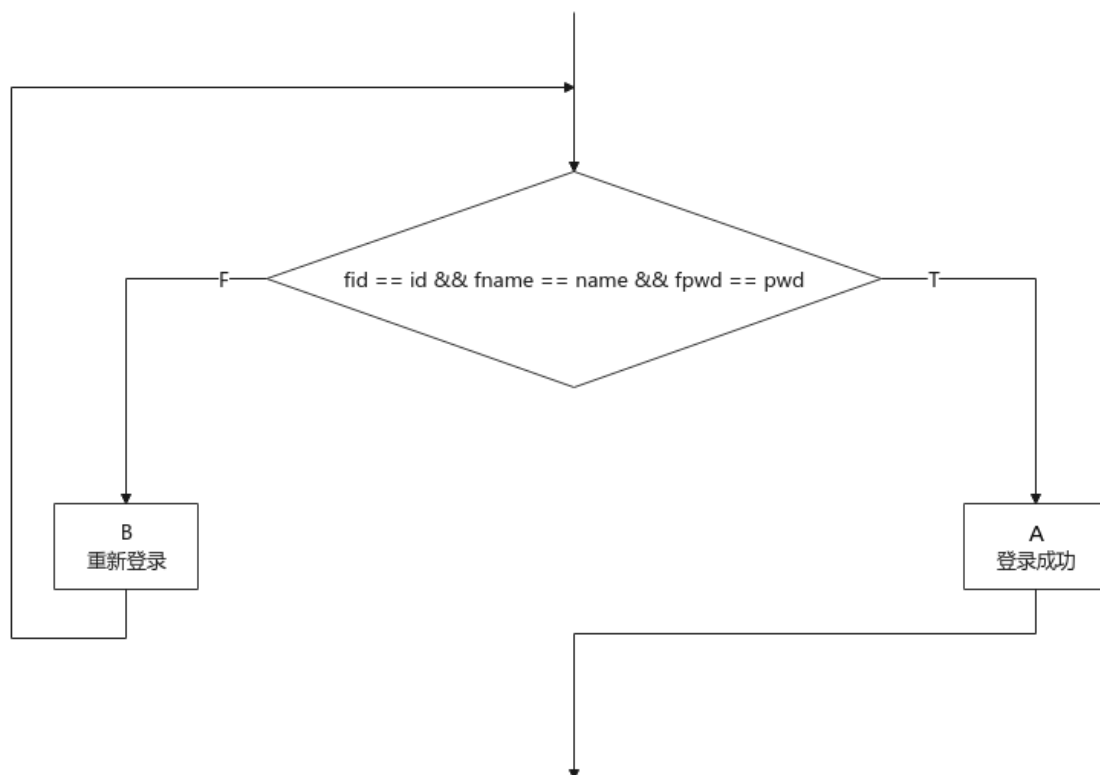
## 7.2.2 教师登录模块测试

代码块

```
if (fid == id && fname == name && fpwd == pwd)
{
    father = new Teacher(id, name, pwd);
    cout << "登录成功" << endl;
    ifs.close();

    this->teacher_work(father);
    return;
}
```

流程图

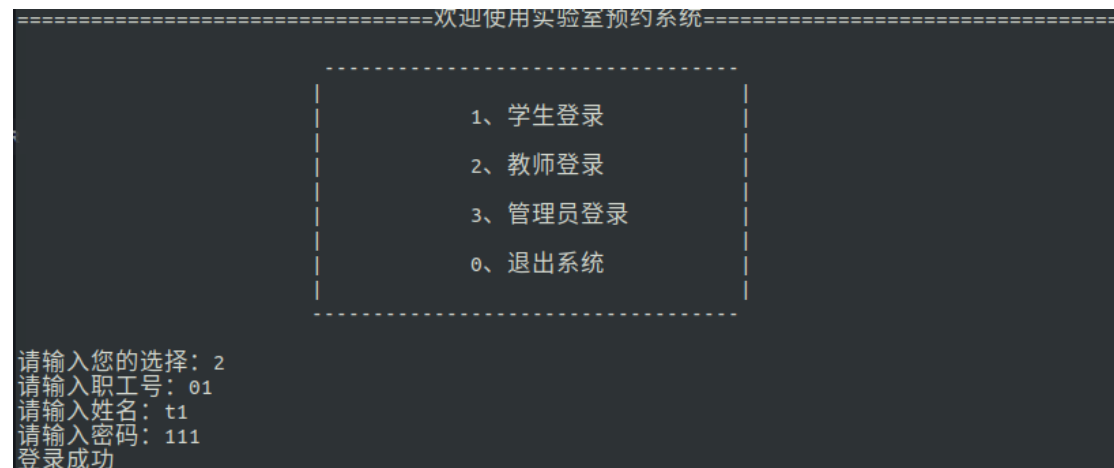


### 语句覆盖测试

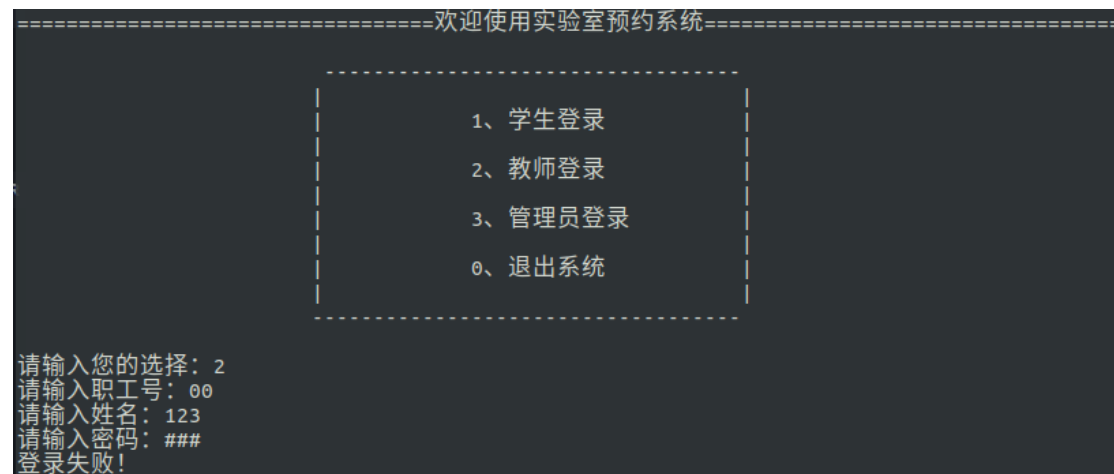
用例编号	用例	期望语句	实际语句
1	(01, t1, 111)	A	A
2	(00, 123, ###)	B	B

### 测试用例结果截图

#### 用例 1



#### 用例 2



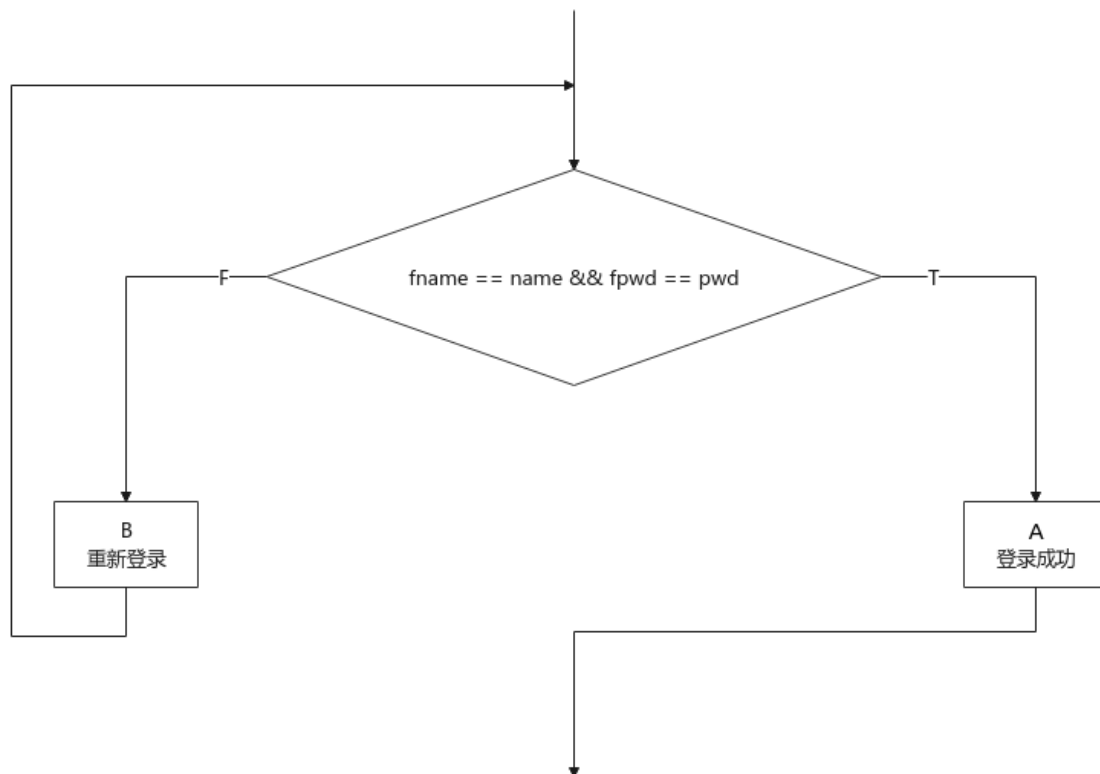
### 7.2.3 管理员登录模块测试

代码块

```
if (fname == name && fpwd == pwd)
{
    father = new Manager(name, pwd);
    cout << "登录成功" << endl;
    ifs.close();

    this->manager_work(father);
    return;
}
```

流程图

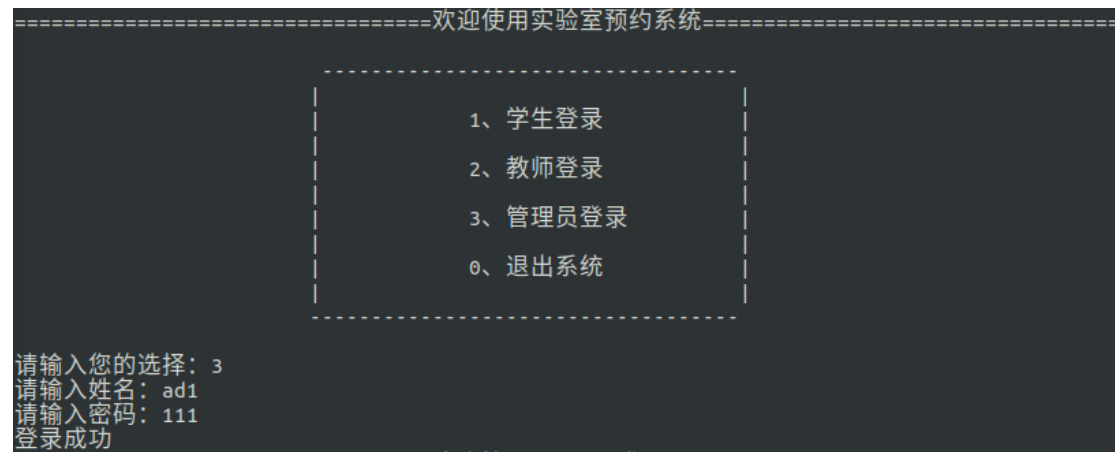


## 语句覆盖测试

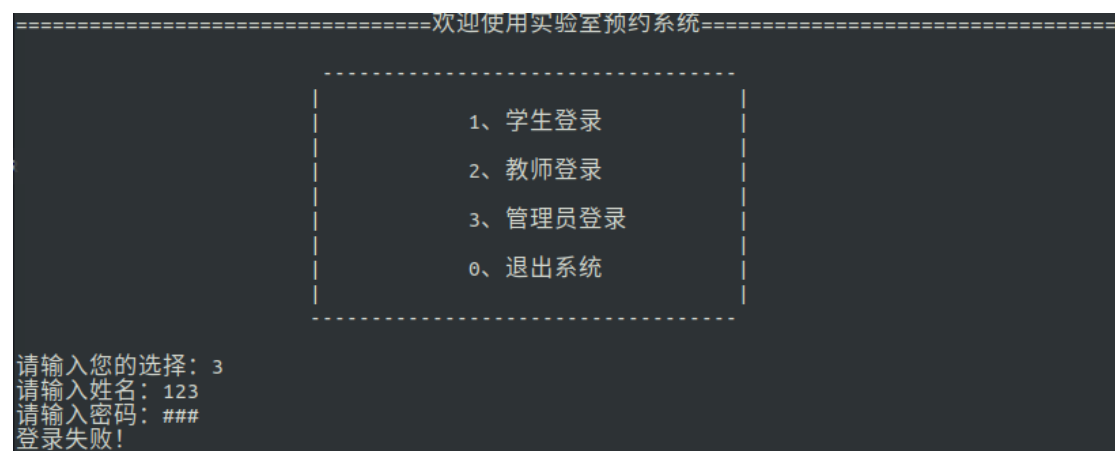
用例编号	用例	期望语句	实际语句
1	(ad1, 111)	A	A
2	(123, ###)	B	B

## 测试用例结果截图

### 用例 1



### 用例 2



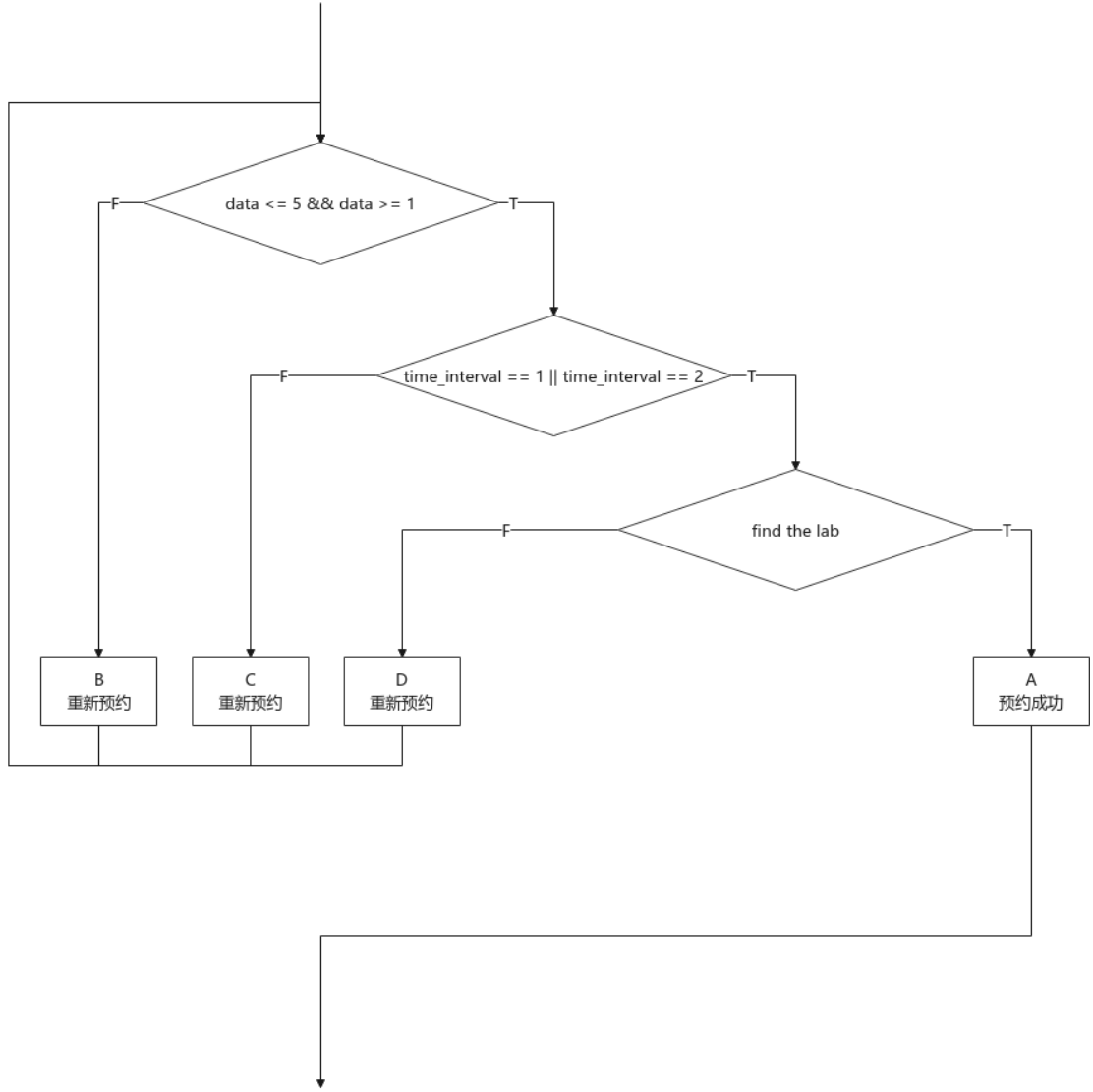


## 7.2.4 预约模块测试

代码块

```
while (1)
{
    cout << "您想预约到周几? " << endl;
    cin >> data;
    if (data <= 5 && data >= 1)
    {
        break;
    }
    cout << "请重新选择" << endl;
}
cout << "您预约的时间段? " << endl;
cout << "1、上午" << endl;
cout << "2、下午" << endl;
while (1)
{
    cin >> time_interval;
    if (time_interval == 1 || time_interval == 2)
    {
        break;
    }
    cout << "请重新选择" << endl;
}
cout << "实验室信息如下: " << endl << endl;
cout << "实验室编号 " << "\t" << "实验室名称" << endl;
for (vector<Lab>::iterator it = vl.begin(); it != vl.end(); it++)
{
    cout << "    " << it->l_id << "\t\t" << it->l_name << endl;
}
cout << endl;
while (1)
{
    cout << "您预约的实验室? " << endl;
    cin >> lab_id;
    int find = 0;
    for (vector<Lab>::iterator it = vl.begin(); it != vl.end(); it++)
    {
        if (lab_id == it->l_id) {
            find = 1;
        }
    }
    if (find) {
        break;
    }
    cout << "请重新选择" << endl;
}
```

流程图



语句覆盖测试

用例编号	用例	期望语句	实际语句
1	(3, 1, 3)	A	A
2	(0, 1, 3)	B	B
3	(3, 0, 3)	C	C
4	(3, 1, 10)	D	D

## 测试用例结果截图

### 用例 1

实验室开放时间为每周一到周五

1、周一

2、周二

3、周三

4、周四

5、周五

您想预约到周几？

3

您预约的时间段？

1、上午

2、下午

1

实验室信息如下：

实验室编号	实验室名称
1	Physics_lab
2	Chemistry_lab
3	Biology_lab
4	cs_lab
11	lab11

您预约的实验室？

3

申请成功，审核中！

### 用例 2

实验室开放时间为每周一到周五

1、周一

2、周二

3、周三

4、周四

5、周五

您想预约到周几？

0

请重新选择

### 用例 3

实验室开放时间为每周一到周五

1、周一

2、周二

3、周三

4、周四

5、周五

您想预约到周几？

3

您预约的时间段？

1、上午

2、下午

0

请重新选择

### 用例 4

实验室开放时间为每周一到周五

1、周一

2、周二

3、周三

4、周四

5、周五

您想预约到周几？

3

您预约的时间段？

1、上午

2、下午

1

实验室信息如下：

实验室编号

实验室名称

1 Physics\_lab

2 Chemistry\_lab

3 Biology\_lab

4 cs\_lab

11 lab11

您预约的实验室？

10

请重新选择

## 7.2.5 审批模块测试

代码块

```
for (; vit != vc.end(); vit++)
{
    if (*vit == opt)
    {
        break;
    }
}
if (vit == vc.end())
{
    cout << "输入有误, 请重新选择" << endl;

    continue;
}

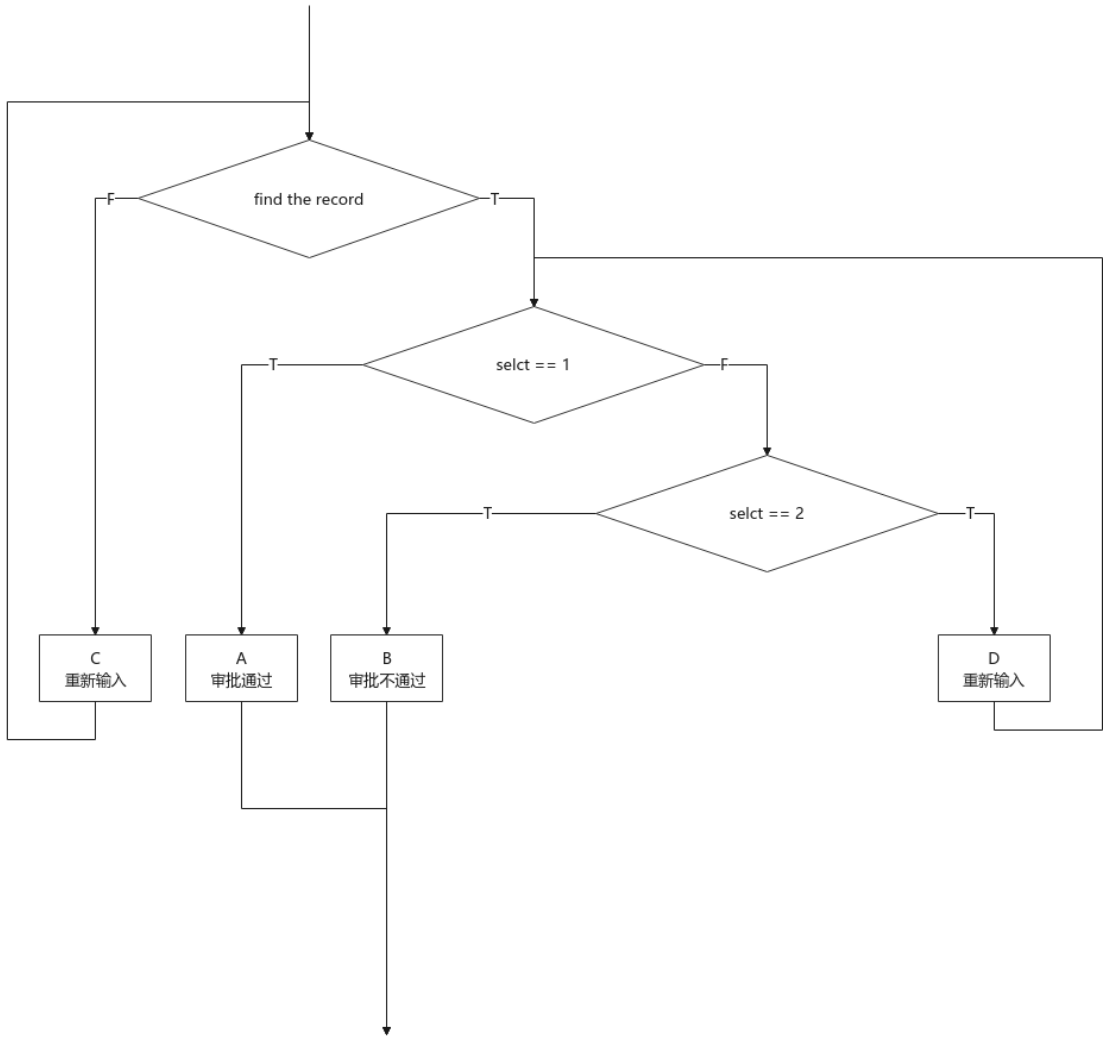
cout << "    1、通过    2、不通过    " << endl;

cin >> selct;
map<int, map<string, string>>::iterator it = o->m.find(opt);
if (selct == 1)
{
    it->second["status"] = "2";
    o->updata();
}
else if (selct == 2)
{
    it->second["status"] = "3";
    o->updata();
}
else
{
    cout << "输入有误, 请重新选择" << endl;

    continue;
}
cout << "审核成功" << endl;
cout << endl;

cout << "    1、继续审核    0、离开    " << endl;
cin >> selct;
if (selct != 1)
{
    break;
}
```

流程图



语句覆盖测试

用例编号	用例	期望语句	实际语句
1	(4, 1)	A	A
2	(8, 2)	B	B
3	(10, 1)	C	C
4	(8, 3)	D	D

测试用例结果截图

用例 1

您需要审核的预约如下：

序号	日期	时间段	实验室	学号	姓名	状态
4	周一	上午	物理实验室	1	ljh	审核中
6	周一	上午	物理实验室	1	ljh	审核中
7	周三	上午	物理实验室	1	ljh	审核中
8	周三	下午	化学实验室	1	ljh	审核中
9	周三	下午	物理实验室	1	ljh	审核中

请选择要审核预约的序号：4

4

1、通过      2、不通过

1

审核成功

用例 2

您需要审核的预约如下：

序号	日期	时间段	实验室	学号	姓名	状态
8	周三	下午	化学实验室	1	ljh	审核中
9	周三	下午	物理实验室	1	ljh	审核中
10	周三	上午	生物实验室	1	ljh	审核中
11	周三	上午	生物实验室	1	ljh	审核中
12	周三	上午	生物实验室	1	ljh	审核中
13	周三	下午	物理实验室	1	ljh	审核中
14	周三	上午	物理实验室	1	ljh	审核中

请选择要审核预约的序号：8

8

1、通过      2、不通过

2

审核成功

用例 3

您需要审核的预约如下：

序号	日期	时间段	实验室	学号	姓名	状态
6	周一	上午	物理实验室	1	ljh	审核中
7	周三	上午	物理实验室	1	ljh	审核中
8	周三	下午	化学实验室	1	ljh	审核中
9	周三	下午	物理实验室	1	ljh	审核中

请选择要审核预约的序号：10

输入有误，请重新选择

用例 4

您需要审核的预约如下：

序号	日期	时间段	实验室	学号	姓名	状态
6	周一	上午	物理实验室	1	ljh	审核中
8	周三	下午	化学实验室	1	ljh	审核中
9	周三	下午	物理实验室	1	ljh	审核中

请选择要审核预约的序号：8

8

1、通过      2、不通过

3

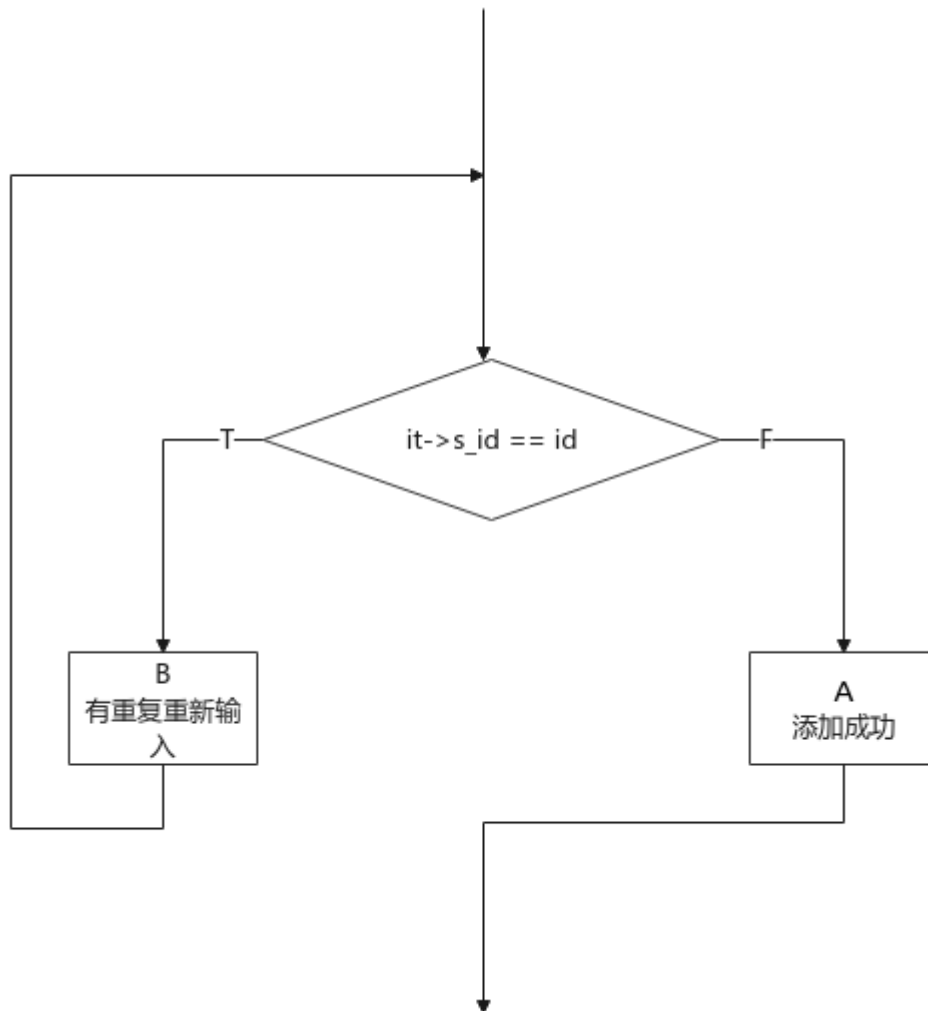
输入有误，请重新选择

### 7.2.6 添加学生账号模块测试

代码块

```
if (it->s_id == id)
{
    cout << "学号重复,请重新输入" << endl;
    return true;
}
```

流程图





### 语句覆盖测试

用例编号	用例	期望语句	实际语句
1	(11, 111, 555)	A	A
2	(01, 111, 555)	B	B

### 测试用例结果截图

#### 用例 1

```
请输入选项: 1
请输入学号: 11
请输入姓名: 111
请输入密码: 555
添加成功
```

#### 用例 2

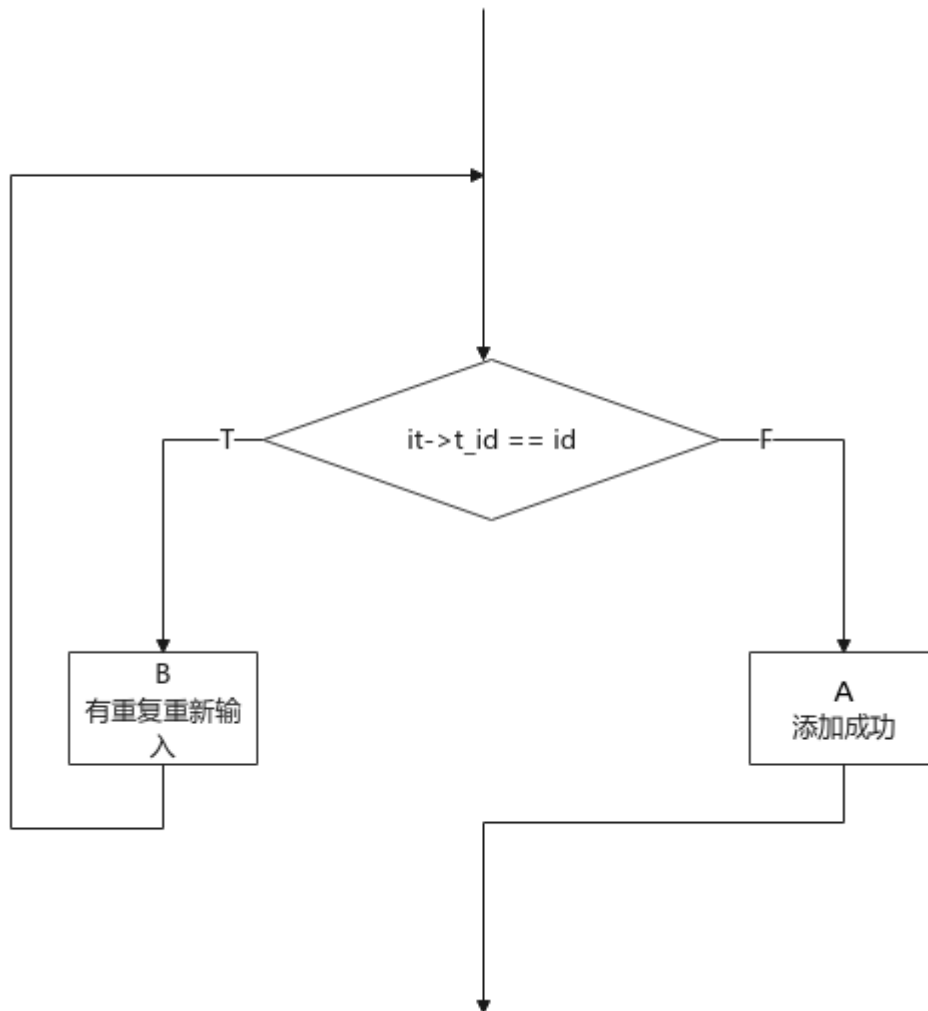
```
请输入选项: 1
请输入学号: 01
学号重复,请重新输入
```

### 7.2.7 添加教师账号模块测试

代码块

```
if (it->t_id == id)
{
    cout << "职工号重复, 请重新输入" << endl;
    return true;
}
```

流程图



### 语句覆盖测试

用例编号	用例	期望语句	实际语句
1	(11, t3, 555)	A	A
2	(01, t3, 555)	B	B

### 测试用例结果截图

#### 用例 1



请输入职工号：11  
请输入姓名：t3  
请输入密码：555  
添加成功

#### 用例 2



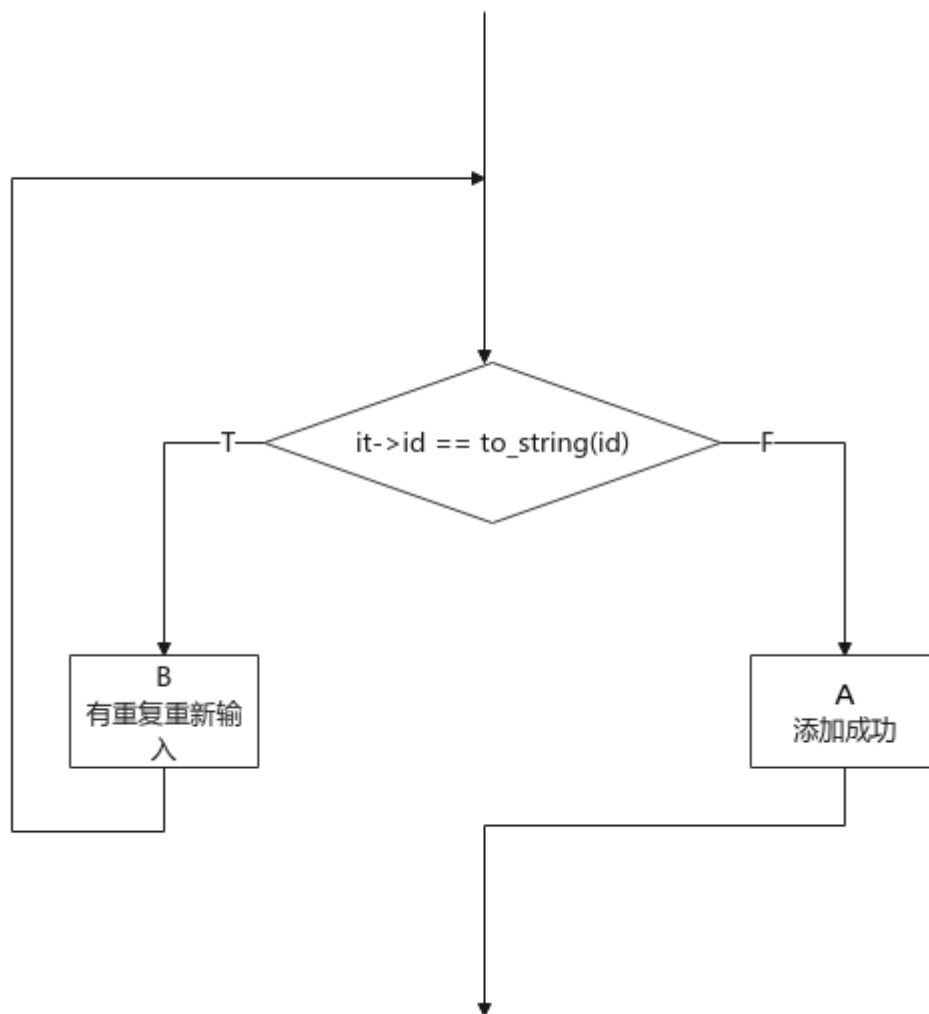
请输入职工号：01  
职工号重复，请重新输入  
请输入职工号：01

### 7.2.8 添加实验室模块测试

代码块

```
if (it->id == to_string(id))  
{  
    cout << "实验室编号重复，请重新输入" << endl;  
    return true;  
}
```

流程图



### 语句覆盖测试

用例编号	用例	期望语句	实际语句
1	(11, lab11)	A	A
2	(1, lab11)	B	B

### 测试用例结果截图

#### 用例 1

```
请输入实验室编号: 11
请输入实验室名称: lab11
添加成功
```

#### 用例 2

```
请输入实验室编号: 1
实验室编号重复, 请重新输入
```

## 8 项目总结

本次项目使用面向对象的分析方法分析了一个实验室管理预约系统。经过规范的分析、设计、测试，整体项目完成度较高。通过本次项目，我对面向对象的分析方法有了更加透彻的认识，能够在日后的软件开发中加以运用。