



实验室预约管理系统

——结构化分析方法

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	问题描述.....	4
2	问题分析.....	4
2.1	风险分析.....	4
2.1.1	需求风险.....	4
2.1.2	技术风险.....	4
2.2	需求分析.....	5
2.2.1	功能需求.....	5
2.2.2	非功能需求.....	5
3	建立数据流图.....	6
3.1	顶层数据流图.....	6
3.2	底层数据流图.....	7
3.2.1	登录.....	8
3.2.2	预约.....	8
3.2.3	取消预约.....	9
3.2.4	审批.....	9
3.2.5	预约管理.....	10
3.2.6	实验室管理.....	11
3.2.7	账号管理.....	11
3.3	数据词典.....	12
3.3.1	数据流和数据项（重复不再赘述）.....	12
3.3.2	文件.....	14
3.4	重点加工说明.....	15
3.4.1	登录.....	15
3.4.2	预约.....	16
3.4.3	取消预约.....	16
3.4.4	审批.....	17
3.4.5	预约管理.....	17
3.4.6	实验室管理.....	17
3.4.7	账号管理.....	18
4	软件设计.....	19
4.1	整体模块结构.....	19
4.2	各模块介绍.....	19
4.2.1	登录模块.....	19
4.2.2	预约模块.....	20
4.2.3	审批模块.....	20
4.2.4	数据管理模块.....	20
4.2.5	系统管理模块.....	21
5	测试用例设计.....	22
5.1	黑盒测试.....	22
5.1.1	学生登录模块测试.....	22
5.1.2	教师登录模块测试.....	24
5.1.3	管理员登录模块测试.....	27

5.1.4	预约模块测试.....	29
5.1.5	审批模块测试.....	32
5.1.6	添加学生账号模块测试.....	34
5.1.7	添加教师账号模块测试.....	34
5.1.8	添加实验室模块测试.....	35
5.2	白盒测试.....	36
5.2.1	学生登录模块测试.....	36
5.2.2	教师登录模块测试.....	38
5.2.3	管理员登录模块测试.....	40
5.2.4	预约模块测试.....	42
5.2.5	审批模块测试.....	46
5.2.6	添加学生账号模块测试.....	49
5.2.7	添加教师账号模块测试.....	51
5.2.8	添加实验室模块测试.....	53
6	项目总结.....	54

1 问题描述

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 问题分析

2.1 风险分析

2.1.1 需求风险

需求风险指项目需求的不确定性所带来的风险。其原因包括对产品的认识不够，需求分析做的不够细致，不断变更的需求，以及缺少有效的需求变更管理方法。需求分析阶段的风险主要是需求分析的不全面不细致导致的，软件项目前期需求分析遗留的问题会随着软件项目的推进逐渐增大，甚至会导致项目中止。不断变更的需求主要是由客户引起的，客户对于自身的需求不够清楚，导致需求不断变更，影响项目进度，解决方法是通过快速原型法让用户清楚自己的需求。最后是缺乏有效的需求变更管理方法，有效的需求管理会大大减少需求变更对项目的负面影响，因此在项目开始前应该制定需求变更的处理方法。

本系统的需求比较简单，核心的功能需求是实验室资源管理，具体功能在于预约、查询和审批，因而，需求风险不大。

2.1.2 技术风险

技术风险主要指软件开发人员的经验缺乏，对工具，方法，框架等理解的不够深刻，使得软件开发进度由于技术问题推迟。对于技术风险的管理办法是提前对开发人员进行培训，提升开发人员的技术水平和应用领域的经验。

此外，新的技术的应用也会带来风险。由于新的技术存在较大不确定性，再加上开发人员对于新技术的理解不够深刻，因此可能会产生很多技术问题导致项目延期。

本系统是由个人完成，所以技术风险主要是软件开发人员的开发经验的缺乏。但本系统一方面使用个人较为熟悉的 C++ 编写，另一方面比较简单可控，所以开发起来也比较容易，技术风险也是可控的。

2.2 需求分析

2.2.1 功能需求

1) 基本信息维护功能

能够维护实验室信息、教师信息、学生信息以及管理员信息，系统管理员可以对这些信息进行修改更新。

2) 实验室预约安排功能

能够提供学生在课余时间预约、取消预约实验室的功能。

3) 实验室预约审核功能

实验室教师能够审核学生的预约请求，可以审核通过也可以审核不通过

4) 信息查询功能

学生可以查询到自己的预约信息，老师可以查询所有的预约信息，系统管理员可以查询到所有预约信息。

5) 信息管理功能

系统管理员可以对系统的信息进行管理，添加学生和教师账号，添加多个实验室，以及清空所有预约。

2.2.2 非功能需求

1) 系统可用性需求

系统能够正常运行，按照系统需求学生能够进行实验室预约，教师能够审核预约信息，管理员能够管理系统信息。

2) 系统安全性需求

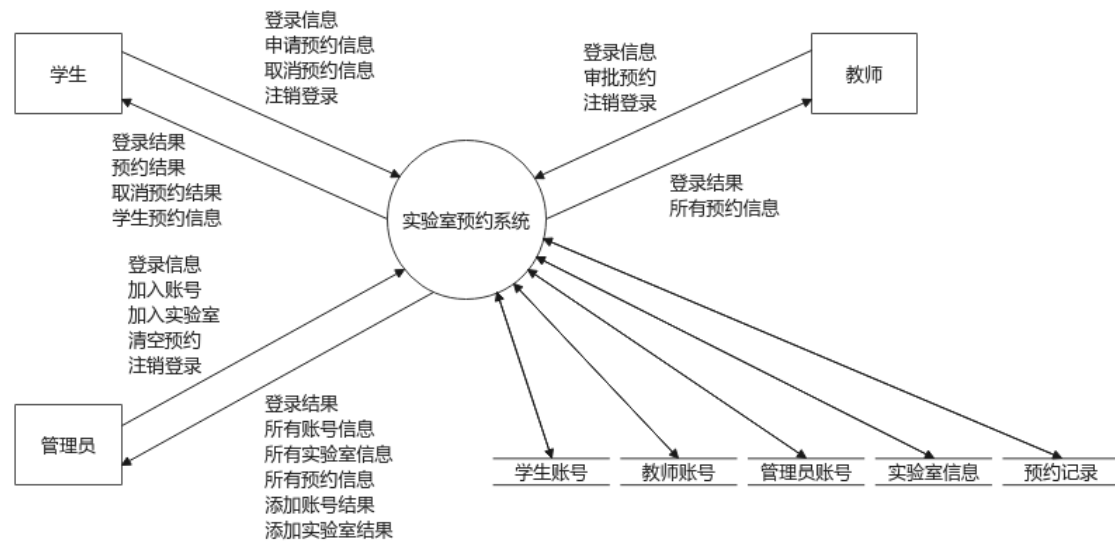
学生和老师账号只能通过接口进行预约、取消预约以及审核的操作，不可以修改其他账户信息、修改预约信息等。

3) 系统效率性需求

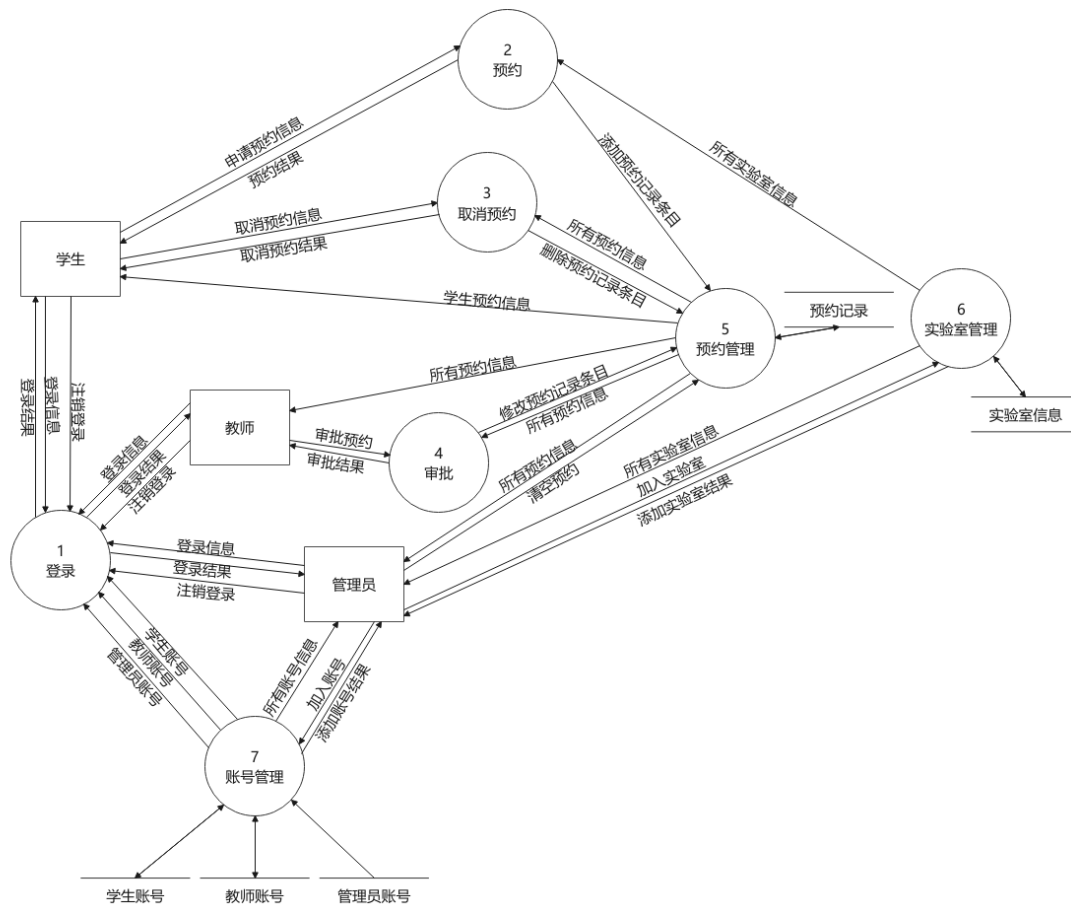
要求系统能够快速工作，快速响应用户的操作，学生预约后，教师可以立刻查询到记录，教师审批后，学生可以立刻得到反馈。

3 建立数据流图

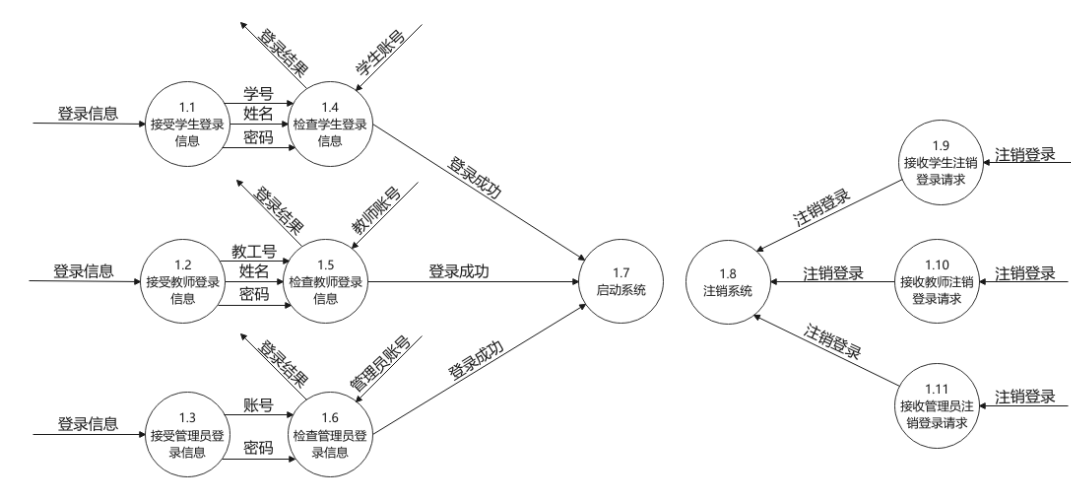
3.1 顶层数据流图



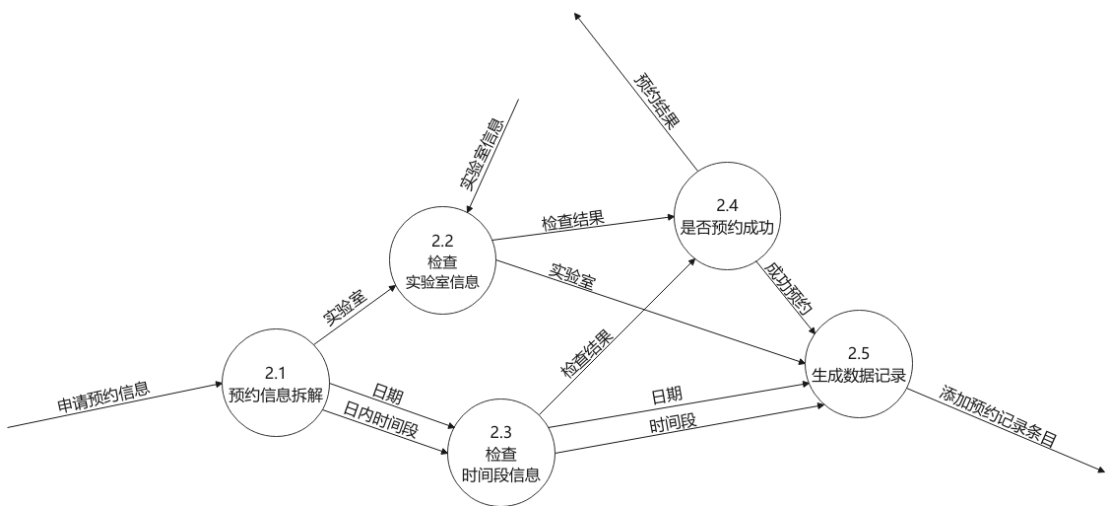
3.2 底层数据流图



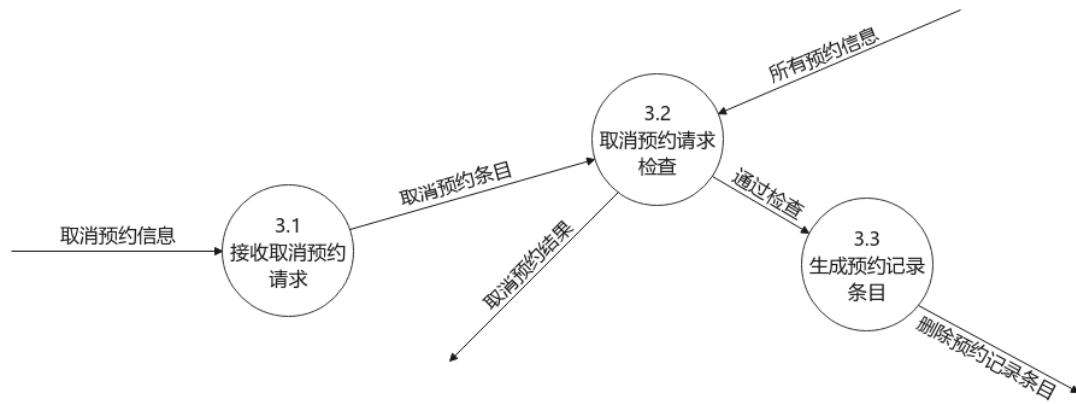
3.2.1 登录



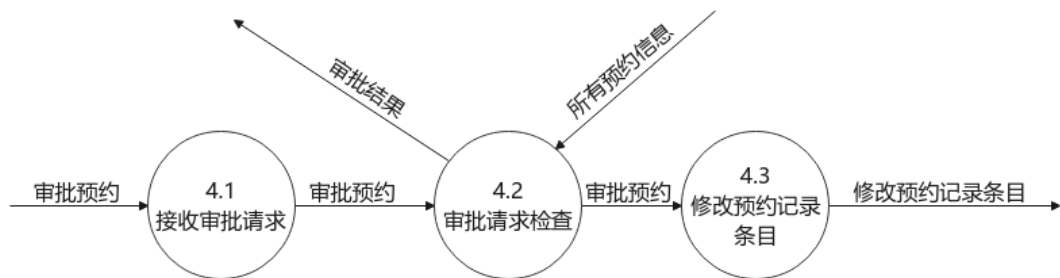
3.2.2 预约



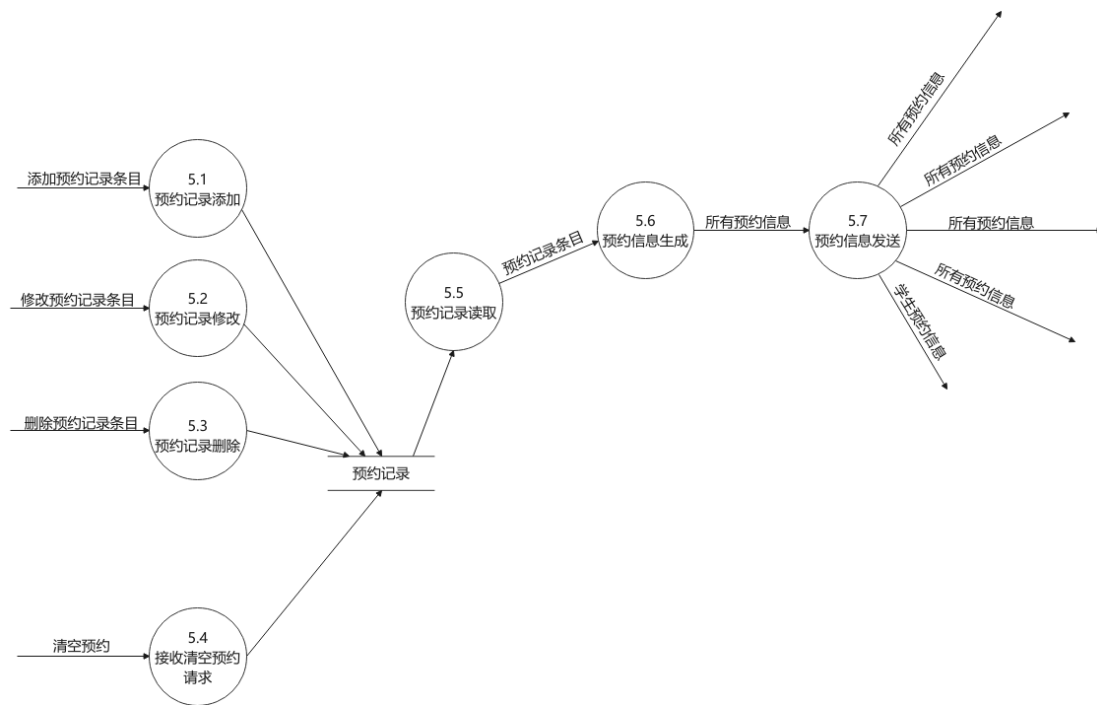
3.2.3 取消预约



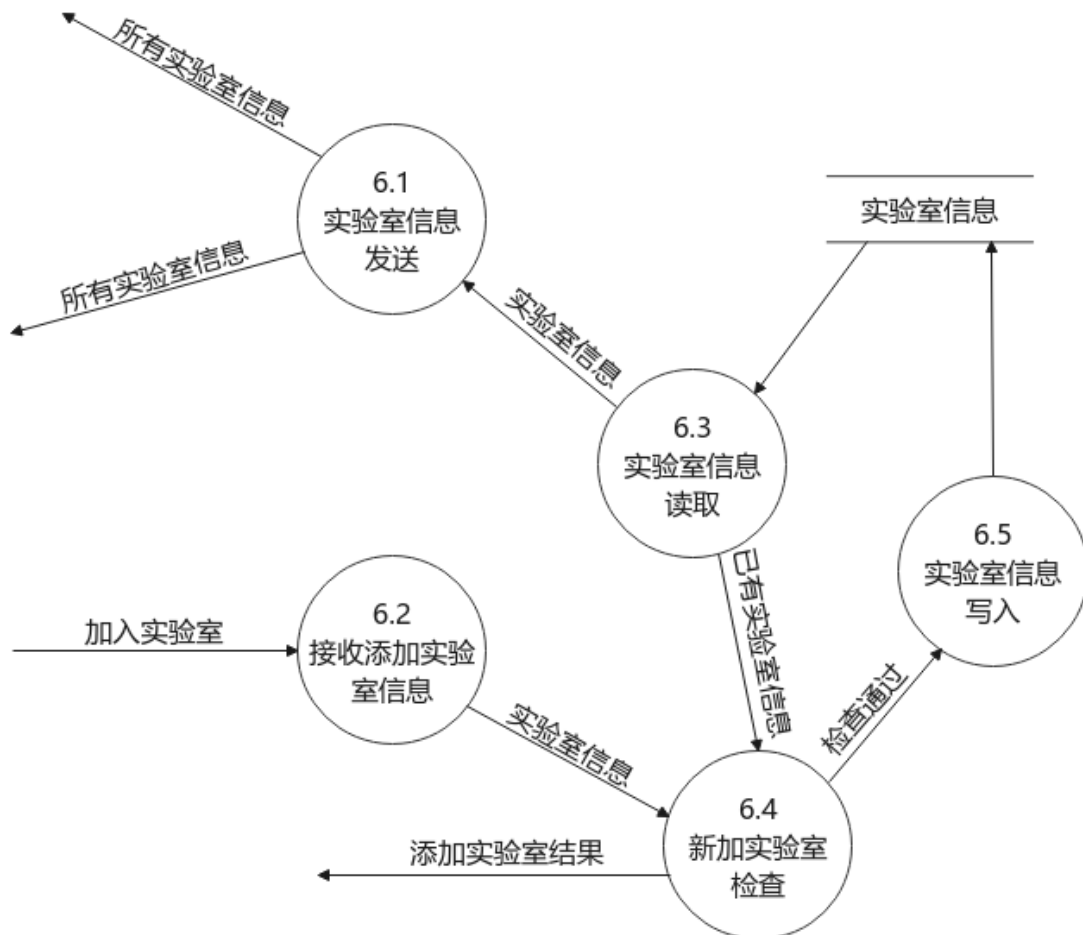
3.2.4 审批



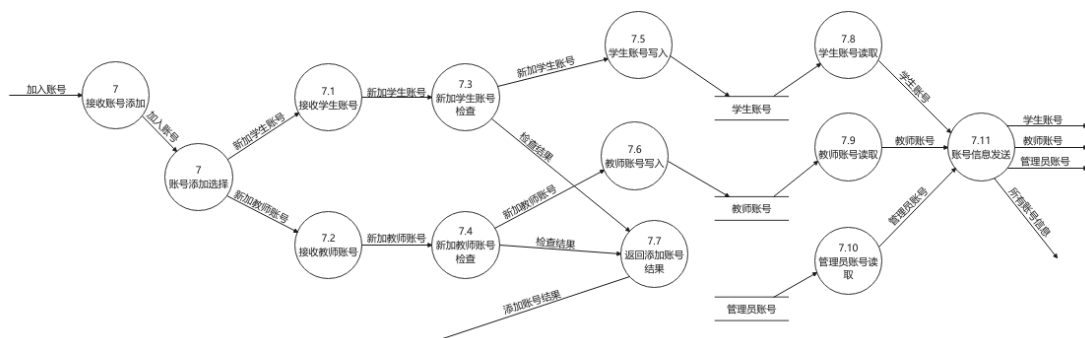
3.2.5 预约管理



3.2.6 实验室管理



3.2.7 账号管理



3.3 数据词典

3.3.1 数据流和数据项（重复不再赘述）

图表	数据流	数据项	数据类型
顶层 DFD	登录信息	ID	INT
		用户名	STRING
		密码	STRING
	登录结果	结果码	CHAR
	注销登录	注销登录请求	CHAR
	申请预约信息	预约人 ID	INT
		预约人姓名	STRING
		预约实验室 ID	INT
		预约时间	INT
		预约时间段	CHAR
	取消预约信息	预约条目 ID	INT
	预约结果	结果码	CHAR
	取消预约结果	结果码	CHAR
	学生预约信息	预约人 ID	INT
		预约人姓名	STRING
		预约实验室 ID	INT
		预约时间	INT
		预约时间段	CHAR
		预约状态码	CHAR
	审批预约	预约条目 ID	INT
		审批状态码	CHAR
	所有预约信息	预约条目 ID	INT
		预约人 ID	INT
		预约人姓名	STRING
		预约实验室 ID	INT
		预约时间	INT
		预约时间段	CHAR
		预约状态码	CHAR
	加入账号	ID	INT
		用户名	STRING
		密码	STRING
	加入实验室	实验室 ID	INT
		实验室名称	STRING
	清空预约	情况预约请求	CHAR
	所有账号信息	ID	INT
		用户名	STRING

		密码	STRING
	所有实验室信息	实验室 ID	INT
		实验室名称	STRING
	添加账号结果	结果码	CHAR
	添加实验室结果	结果码	CHAR
一层 DFD	学生账号	学号	INT
		姓名	STRING
		密码	STRING
	教师账号	教工号	INT
		姓名	STRING
		密码	STRING
	管理员账号	账户名	STRING
		密码	STRING
	添加预约记录条目	添加预约条目 ID	INT
		预约人 ID	INT
		预约人姓名	STRING
		预约实验室 ID	INT
		预约时间	INT
		预约时间段	CHAR
		预约状态码	CHAR
	删除预约条目	删除预约条目 ID	INT
	修改预约条目	修改预约条目 ID	INT
		修改后预约状态码	CHAR
1 登录	学号	号码	INT
	教工号	号码	INT
	姓名	姓名字符串	STRING
	密码	密码字符串	STRING
	登录成功	成功状态码	CHAR
2 预约	实验室	实验室 ID	INT
	日期	周一到周五	INT
	日内时间段	上午或下午	CHAR
	检查结果	结果码	CHAR
	成功预约	成功预约码	CHAR
3 取消预约	取消预约条目	条目 ID	INT
	通过检查	结果码	CHAR
4 审批			
5 预约管理	预约记录条目	预约条目 ID	INT
		预约人 ID	INT
		预约人姓名	STRING
		预约实验室 ID	INT
		预约时间	INT
		预约时间段	CHAR

		预约状态码	CHAR
6 实验室管理	实验室信息	实验室 ID	INT
		实验室名称	STRING
	已有实验室信息	实验室 ID	INT
		实验室名称	STRING
	检查通过	结果码	CHAR
7 账号管理	新加学生账号	学号	INT
		姓名	STRING
		密码	STRING
	新加教师账号	教工号	INT
		姓名	STRING
		密码	STRING
	检查结果	结果码	CHAR

3.3.2 文件

文件	数据项	数据类型
学生账号	学号	INT
	姓名	STRING
	密码	STRING
教师账号	教工号	INT
	姓名	STRING
管理员账号	账户名	STRING
	密码	STRING
实验室信息	实验室 ID	INT
	实验室名称	STRING
预约记录	预约条目 ID	INT
	预约人 ID	INT
	预约人姓名	STRING
	预约实验室 ID	INT
	预约时间	INT
	预约时间段	CHAR
	预约状态码	CHAR

3.4 重点加工说明

3.4.1 登录

1.4 检查学生登录信息

```
FIND=FLASE
FOR IT IN 所有学生账号信息
    IF 学号==IT.学号 AND 姓名==IT.姓名 AND 密码==IT.密码
        FIND=TRUE
    ENDIF
ENDFOR
IF FIND
    登录结果=0
ELSE
    登录结果=1
ENDIF
```

1.5 检查教师登录信息

```
FIND=FLASE
FOR IT IN 所有教师账号信息
    IF 教工号==IT.教工号 AND 姓名==IT.姓名 AND 密码==IT.密码
        FIND=TRUE
    ENDIF
ENDFOR
IF FIND
    登录结果=0
ELSE
    登录结果=1
ENDIF
```

1.6 检查管理员登录信息

```
FIND=FLASE
FOR IT IN 所有管理员账号信息
    IF 用户名==IT.用户名 AND 密码==IT.密码
        FIND=TRUE
    ENDIF
ENDFOR
IF FIND
    登录结果=0
ELSE
    登录结果=1
ENDIF
```

3.4.2 预约

2.2 检查实验室信息

```
FIND=FLASE
FOR IT IN 所有实验室信息
    IF 实验室==IT. ID
        FIND=TRUE
    ENDIF
ENDFOR
IF FIND
    检查结果=0
ELSE
    检查结果=1
ENDIF
```

2.3 检查时间段信息

```
IF 日期>=1 AND 日期<=5 AND (时间段==" M" OR 时间段==" A" )
    检查结果=0
ELSE
    检查结果=1
ENDIF
```

3.4.3 取消预约

3.2 取消预约请求检查

```
FIND=FLASE
FOR IT IN 所有预约条目
    IF 取消预约条目 ID==IT. ID
        FIND=TRUE
    ENDIF
ENDFOR
IF FIND
    通过检查=0
ELSE
    通过检查=1
ENDIF
```


3.4.4 审批

4.2 审批请求检查

```
FIND=FLASE
FOR IT IN 所有预约条目
    IF 审批预约条目 ID==IT. ID AND IT. 状态==待审批
        FIND=TRUE
    ENDIF
ENDFOR
IF FIND
    审批预约=0
ELSE
    审批预约=1
ENDIF
```

3.4.5 预约管理

无

3.4.6 实验室管理

6.4 新加实验室检查

```
FIND=FLASE
FOR IT IN 所有实验室信息
    IF 新加实验室 ID==IT. ID
        FIND=TRUE
    ENDIF
ENDFOR
IF FIND
    检查通过=1
ELSE
    检查通过=0
ENDIF
```

3.4.7 账号管理

7.3 新加学生账号检查

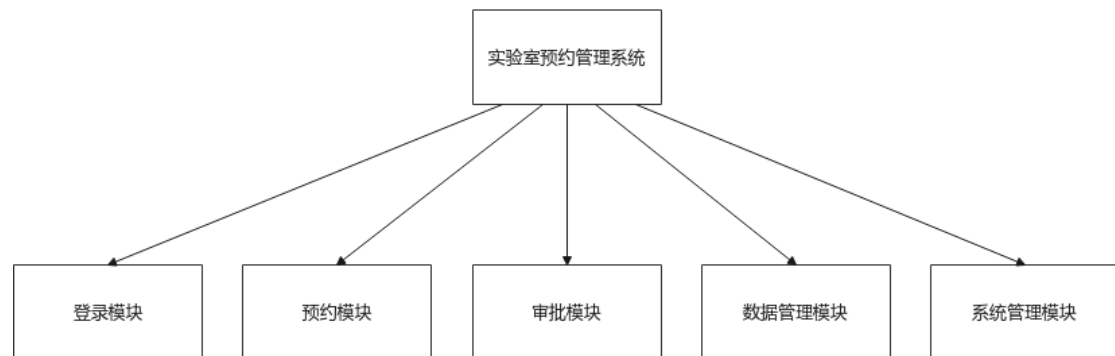
```
FIND=FLASE
FOR IT IN 所有学生账号
    IF 新加学生账号 ID==IT. ID
        FIND=TRUE
    ENDIF
ENDFOR
IF FIND
    检查结果=1
ELSE
    检查结果=0
ENDIF
```

7.4 新加教师账号检查

```
FIND=FLASE
FOR IT IN 所有教师账号
    IF 新加教师账号 ID==IT. ID
        FIND=TRUE
    ENDIF
ENDFOR
IF FIND
    检查结果=1
ELSE
    检查结果=0
ENDIF
```

4 软件设计

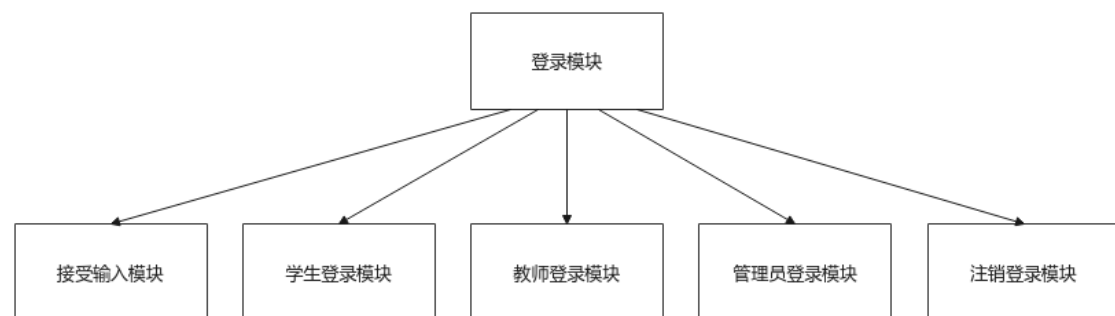
4.1 整体模块结构



4.2 各模块介绍

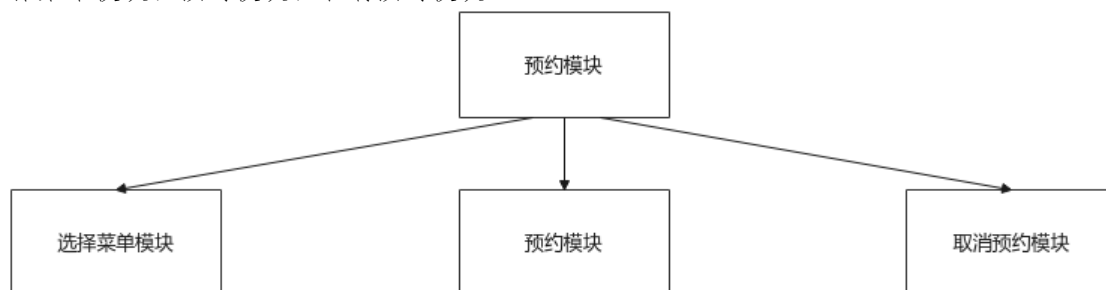
4.2.1 登录模块

登录模块是系统与用户交互的第一个模块，用户在登录模块输入登录信息，只有登录成功才能进入系统，使用系统的其他功能。登录模块还分为接受输入模块、学生登录模块、教师登录模块、管理员登录模块、注销登录模块。



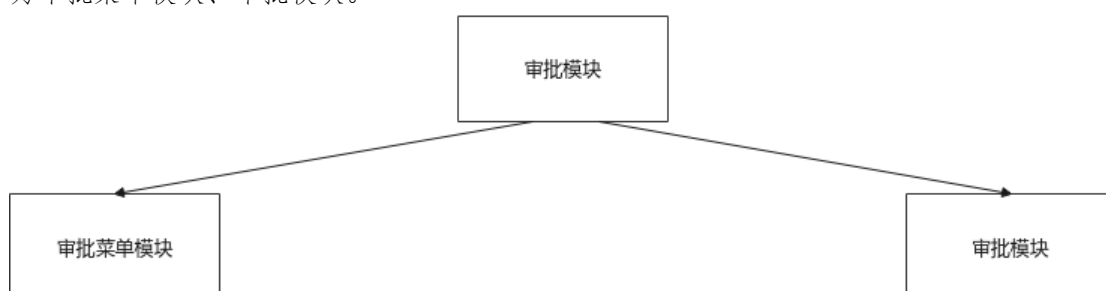
4.2.2 预约模块

预约模块是面向学生用户，为学生用户提供预约实验室的相关功能，预约模块又分为选择菜单模块、预约模块、取消预约模块。



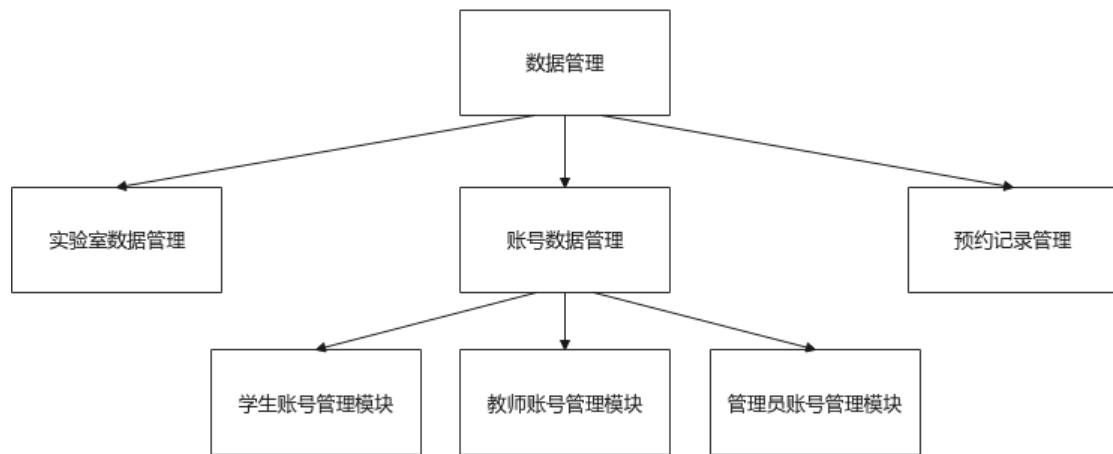
4.2.3 审批模块

审批模块是面向教师用户，为教师用户提供审批实验室预约的相关功能，审批模块又分为审批菜单模块、审批模块。



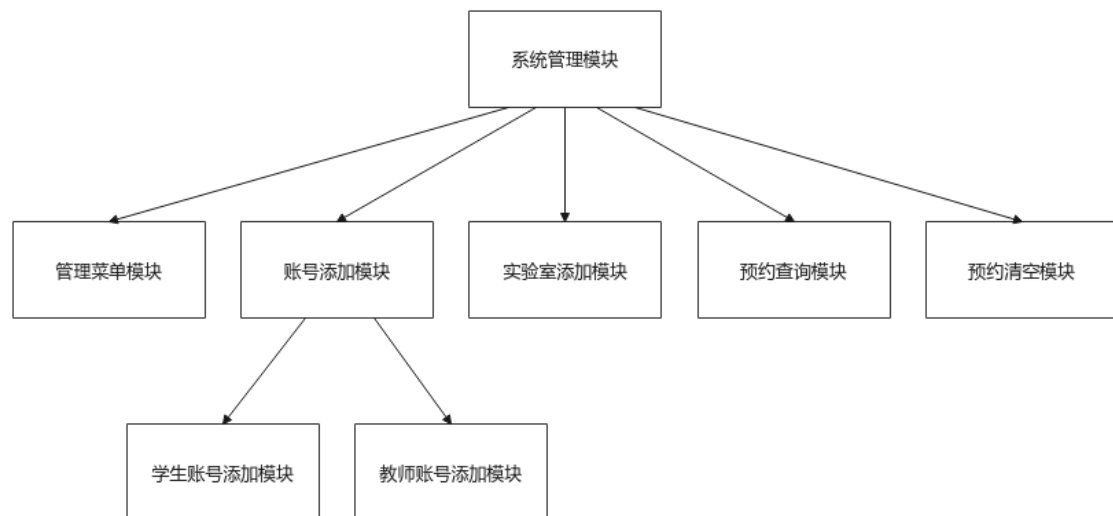
4.2.4 数据管理模块

数据管理模块主要功能是管理系统底层数据信息，向外提供对数据读写的接口，管理数据包括各种账号信息，即学生账号、教师账号、管理员账号，实验室信息，预约条目记录。



4.2.5 系统管理模块

系统管理模块面向系统管理员，为系统管理员提供一些维护功能，主要包括管理菜单模块、账号添加模块、实验室添加模块、预约查询模块、预约清空模块。



5 测试用例设计

5.1 黑盒测试

5.1.1 学生登录模块测试

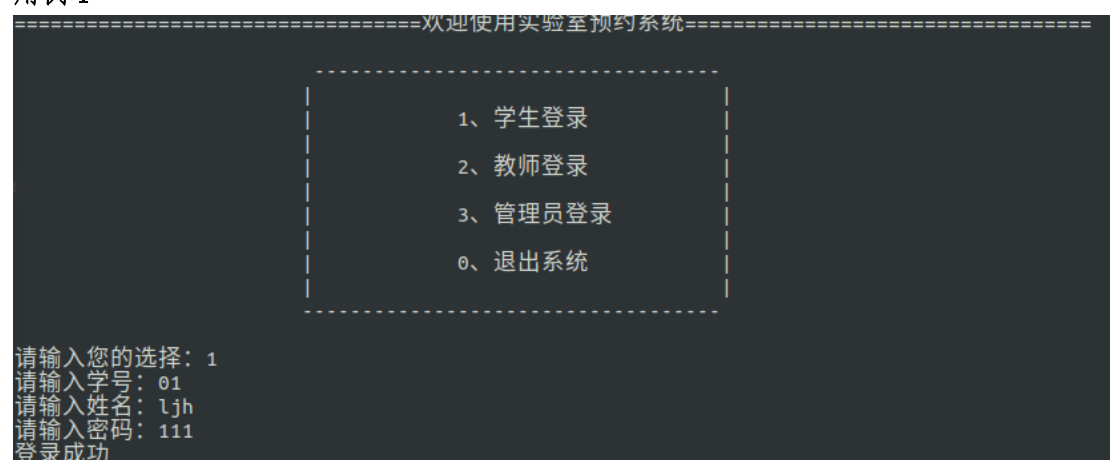
输入数据	有效等价类	无效等价类
学号	1 位于账号信息内	2 不位于账号信息内
姓名	3 与学号信息对应	4 不位于账号信息内 5 位于账号信息内但与学号不对应
密码	6 与学号信息对应	7 不位于账号信息内 8 位于账号信息内但与学号不对应

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

```
=====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：00
请输入姓名：ljh
请输入密码：111
登录失败！
```

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

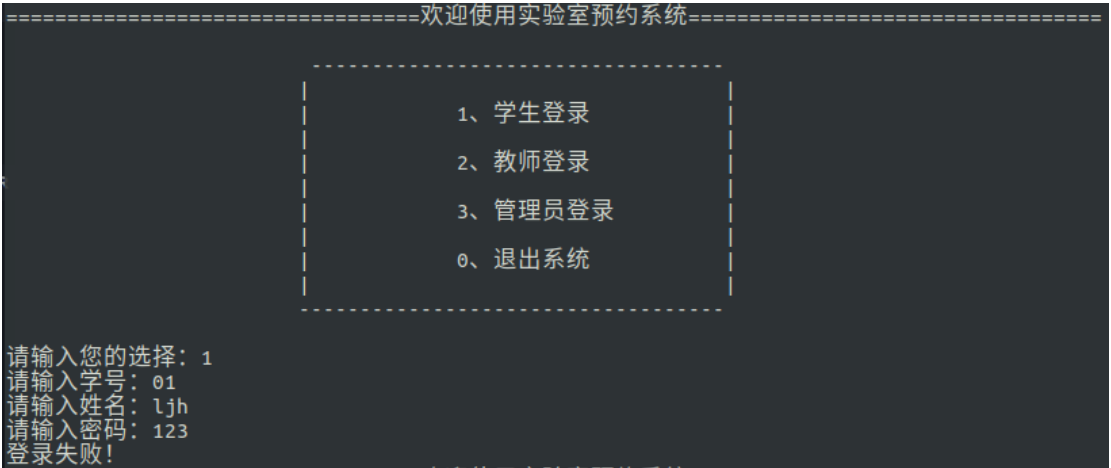
请输入您的选择：1
请输入学号：01
请输入姓名：lll
请输入密码：111
登录失败！
```

用例 4

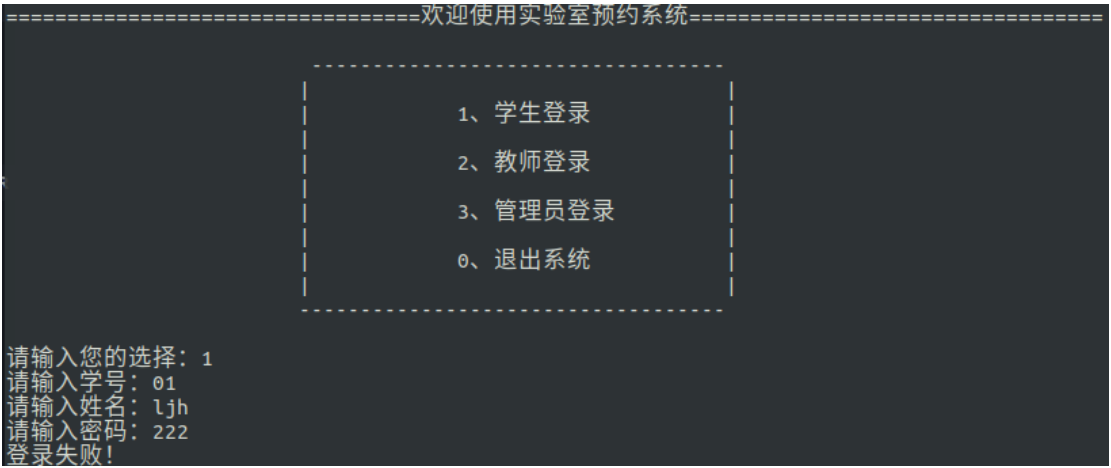
```
=====欢迎使用实验室预约系统=====
          1、学生登录
          2、教师登录
          3、管理员登录
          0、退出系统

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！
```

用例 5



用例 6



5.1.2 教师登录模块测试

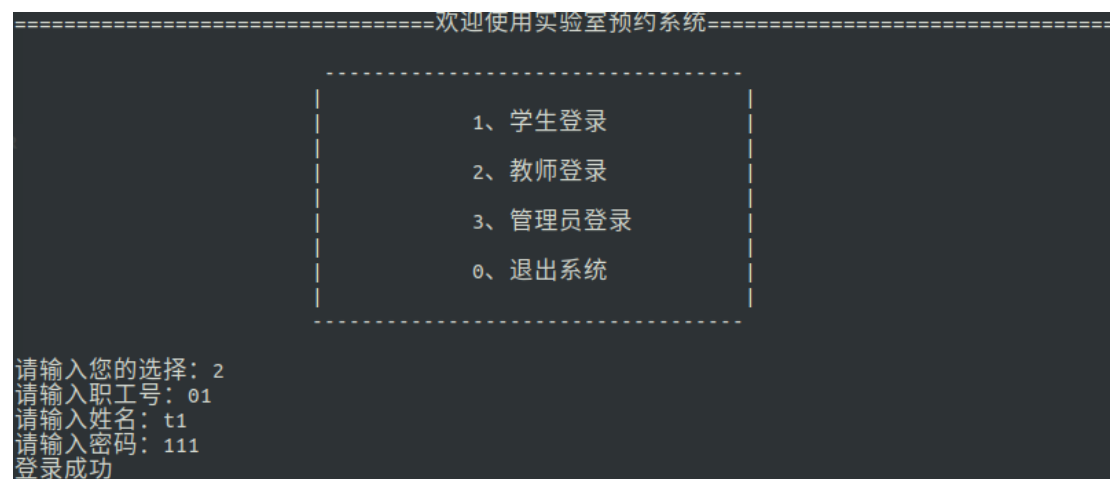
输入数据	有效等价类	无效等价类
教工号	1 位于账号信息内	2 不位于账号信息内
姓名	3 与教工号信息对应	4 不位于账号信息内 5 位于账号信息内但与教工号不对应
密码	6 与教工号信息对应	7 不位于账号信息内 8 位于账号信息内但与教工号不对应

设计用例

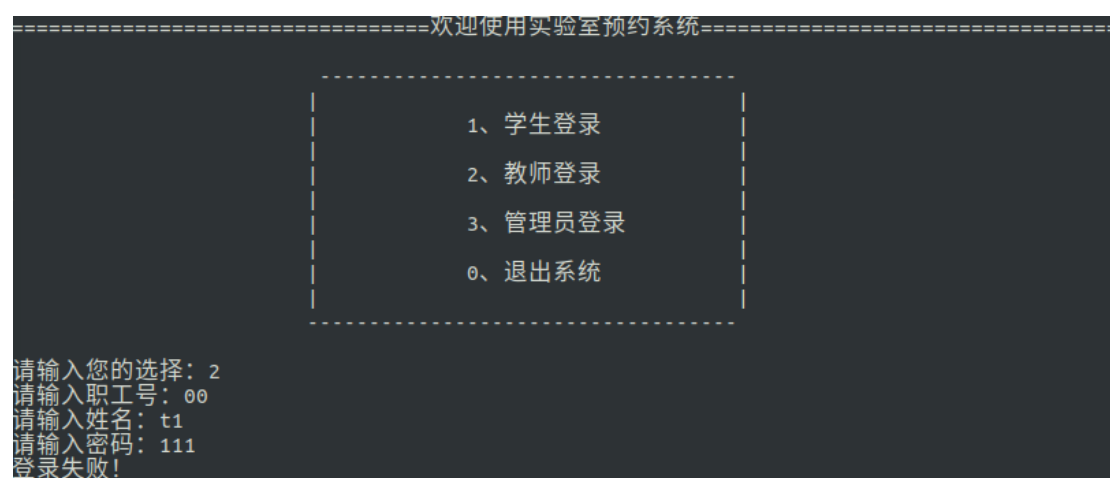
用例编号	用例	期望结果	覆盖条件	实际结果
1	(01, t1, 111)	有效	136	有效
2	(00, t1, 111)	无效	2	无效
3	(01, tt, 111)	无效	4	无效
4	(01, t2, 111)	无效	5	无效
5	(01, t1, 123)	无效	7	无效
6	(01, t1, 222)	无效	8	无效

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：2
请输入职工号：01
请输入姓名：tt
请输入密码：111
登录失败!
```

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!
```

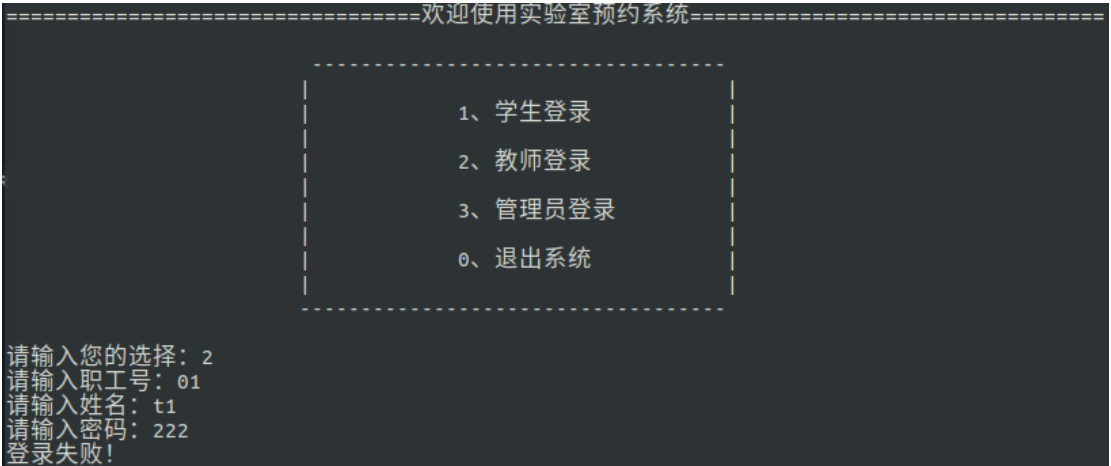
用例 5

```
=====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



5.1.3 管理员登录模块测试

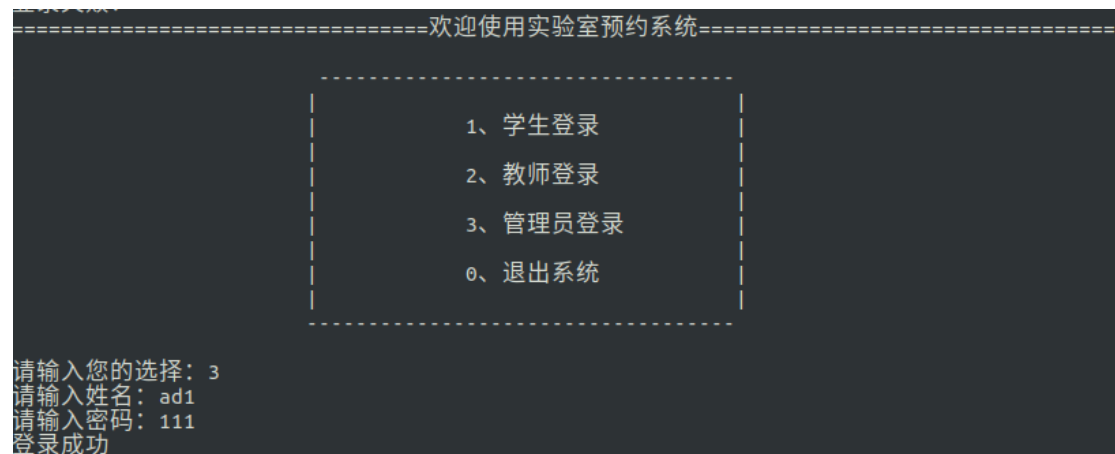
输入数据	有效等价类	无效等价类
账号	1 位于账号信息内	2 不位于账号信息内
密码	3 与账号信息对应	4 不位于账号信息内 5 位于账号信息内但与账号不对应

设计用例

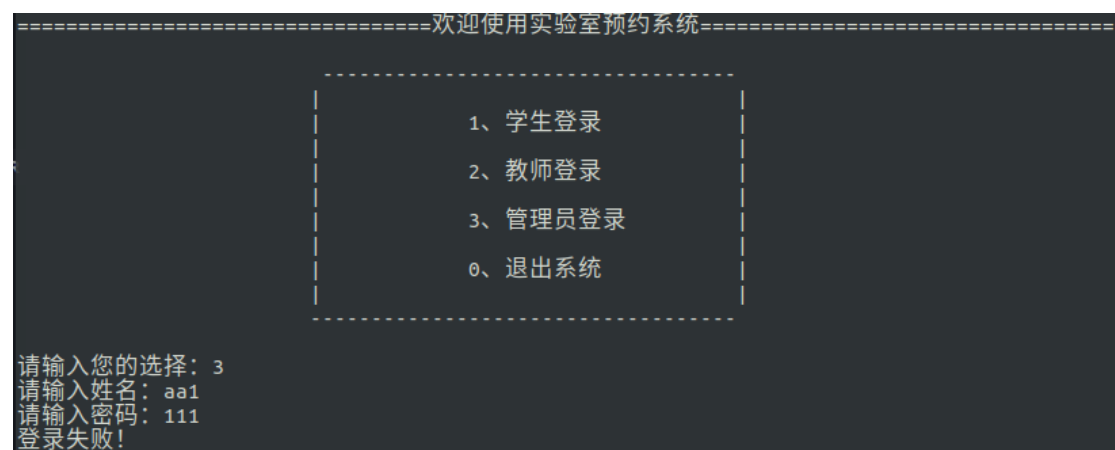
用例编号	用例	期望结果	覆盖条件	实际结果
1	(ad1, 111)	有效	13	有效
2	(aa1, 111)	无效	2	无效
3	(ad1, 123)	无效	4	无效
4	(ad1, 222)	无效	5	无效

测试用例结果截图

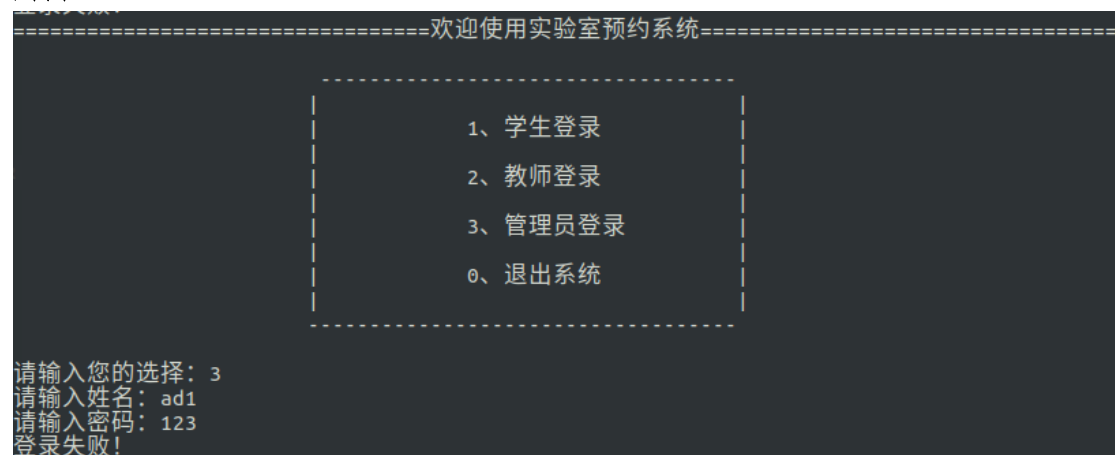
用例 1



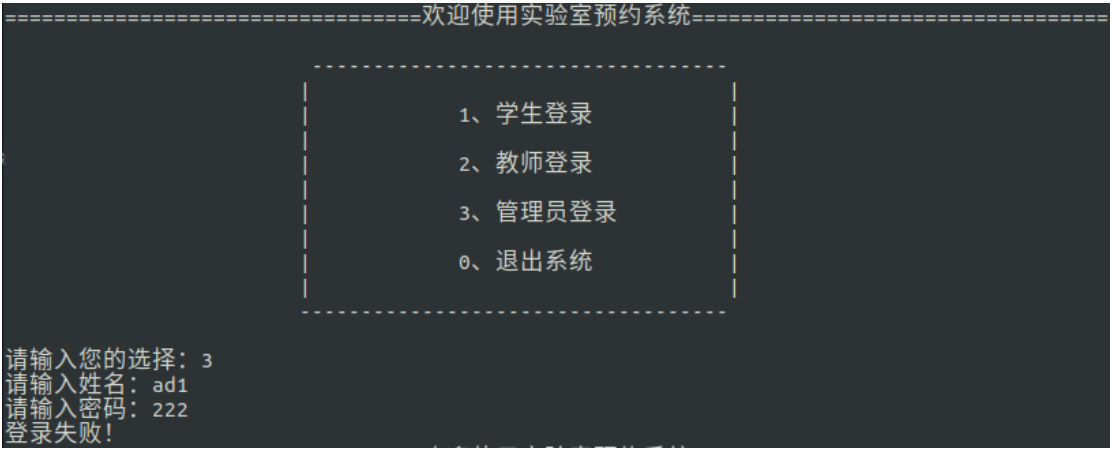
用例 2



用例 3



用例 4



5.1.4 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
```

5.1.5 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

5.1.6 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
```

5.1.7 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
```

5.1.8 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
```

5.2 白盒测试

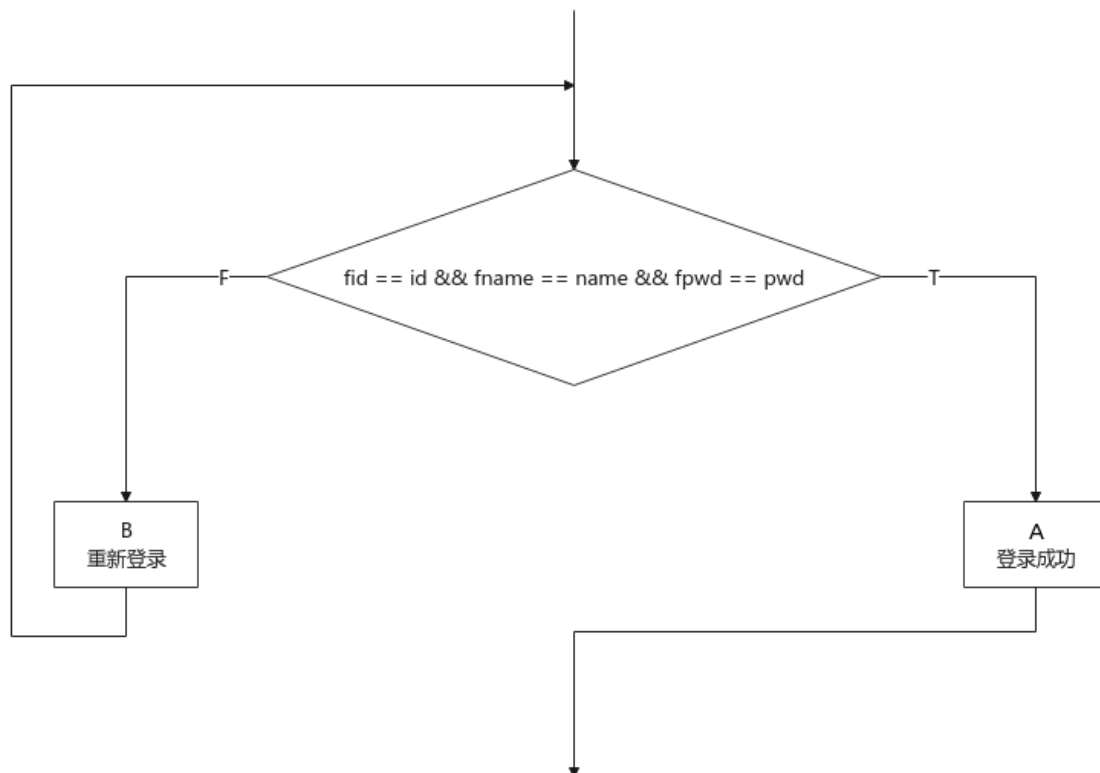
5.2.1 学生登录模块测试

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

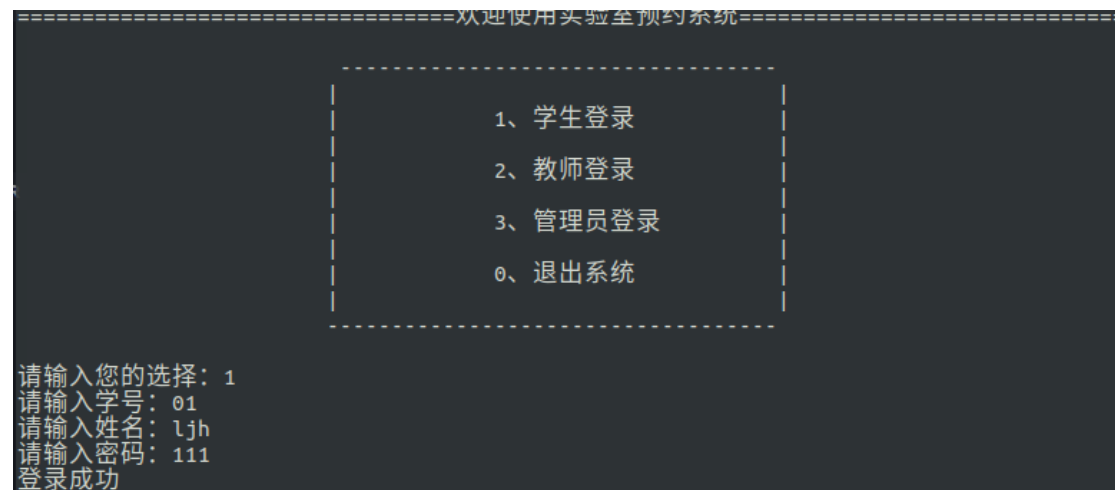


语句覆盖测试

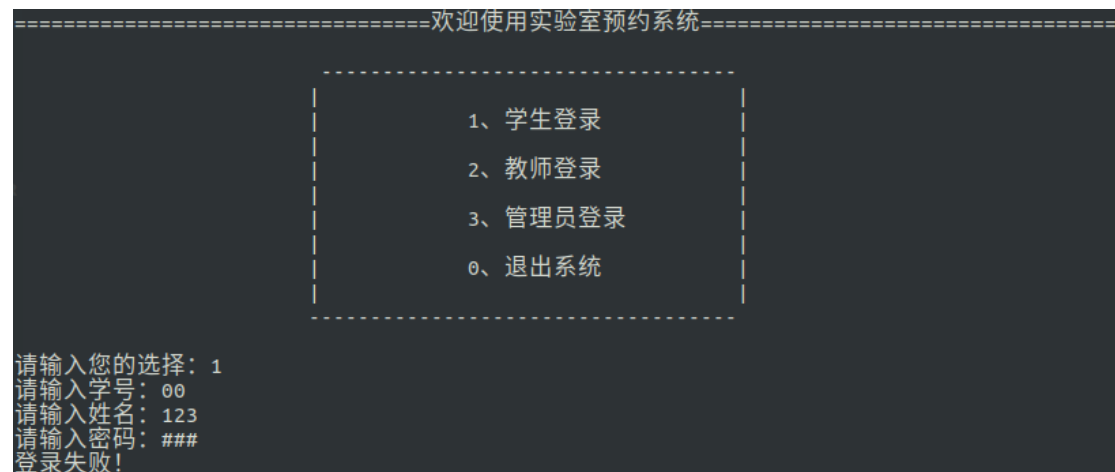
用例编号	用例	期望语句	实际语句
1	(01, ljh, 111)	A	A
2	(00, 123, ###)	B	B

测试用例结果截图

用例 1



用例 2



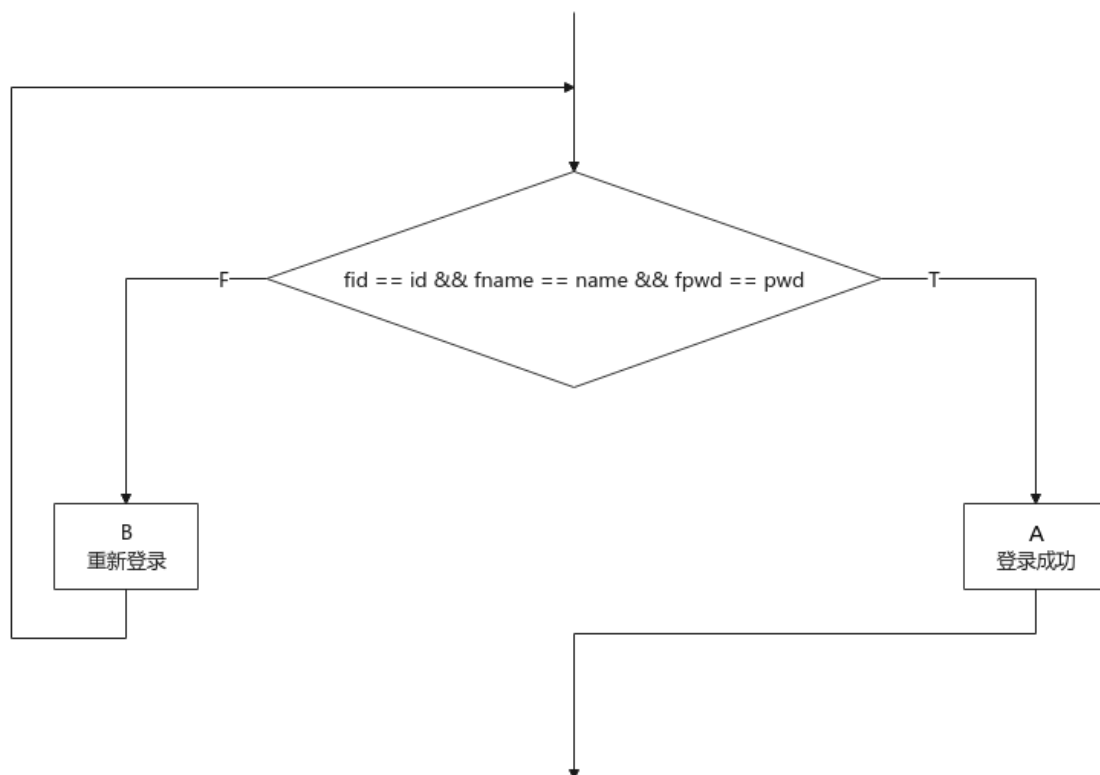
5.2.2 教师登录模块测试

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

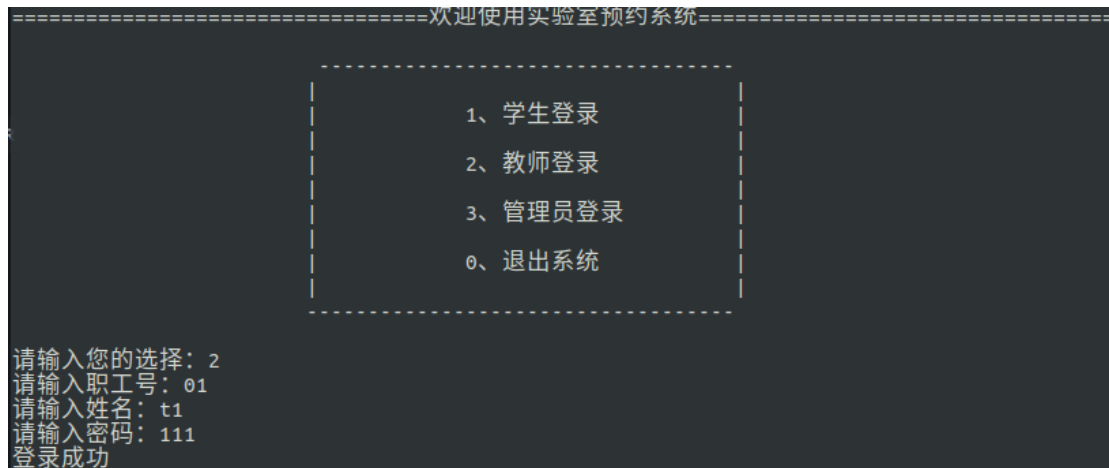


语句覆盖测试

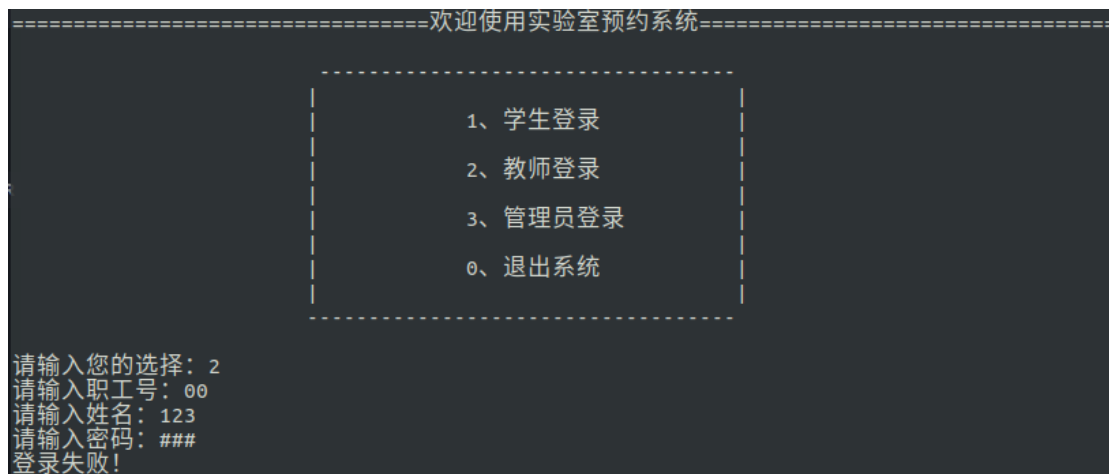
用例编号	用例	期望语句	实际语句
1	(01, t1, 111)	A	A
2	(00, 123, ###)	B	B

测试用例结果截图

用例 1



用例 2



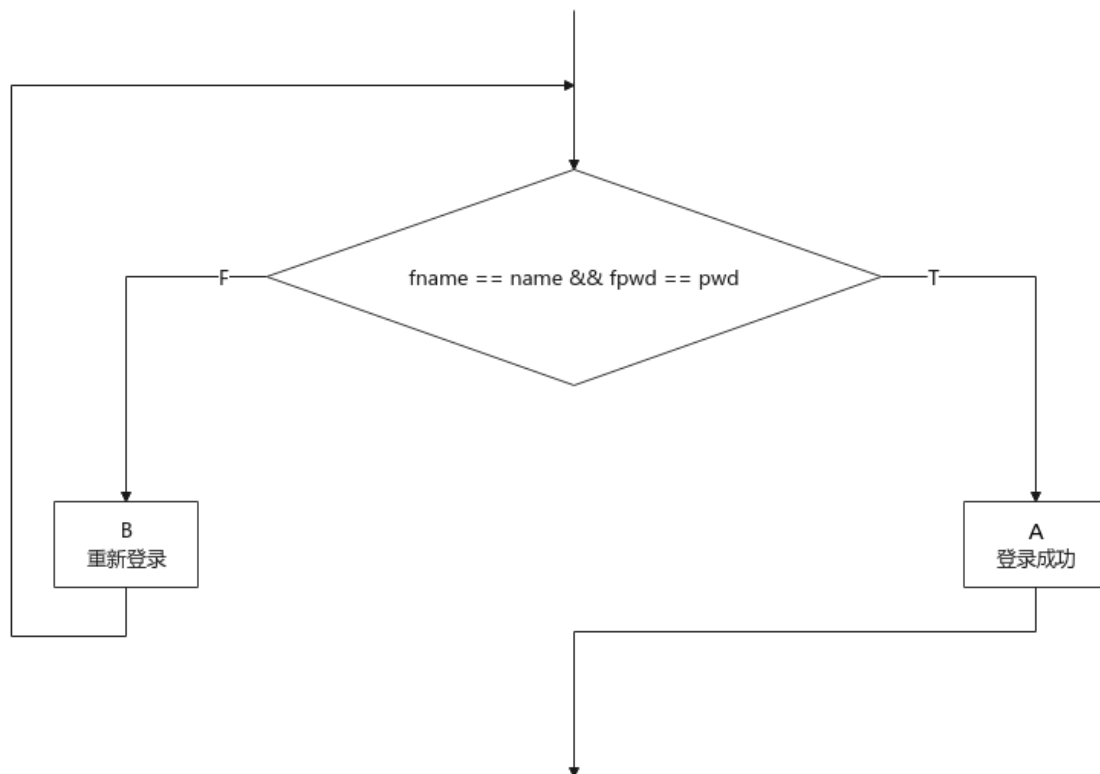
5.2.3 管理员登录模块测试

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

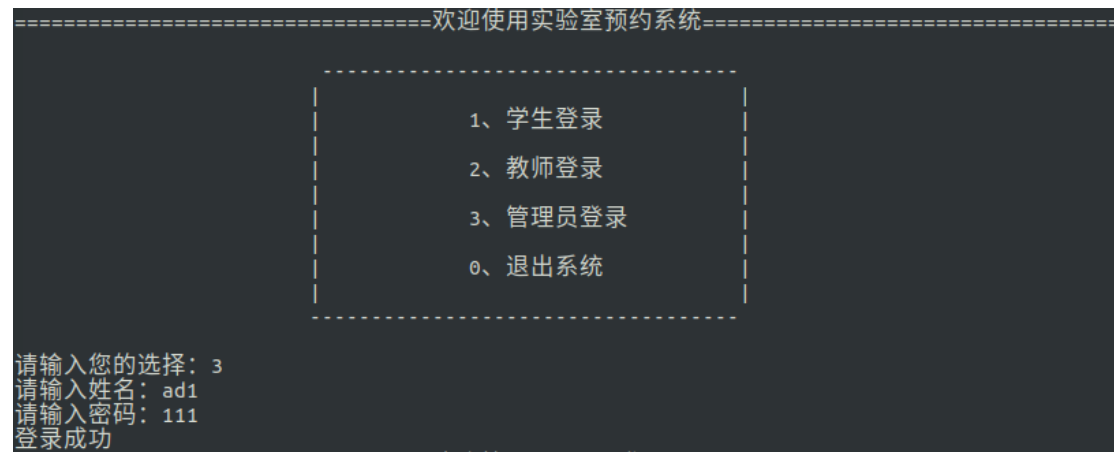


语句覆盖测试

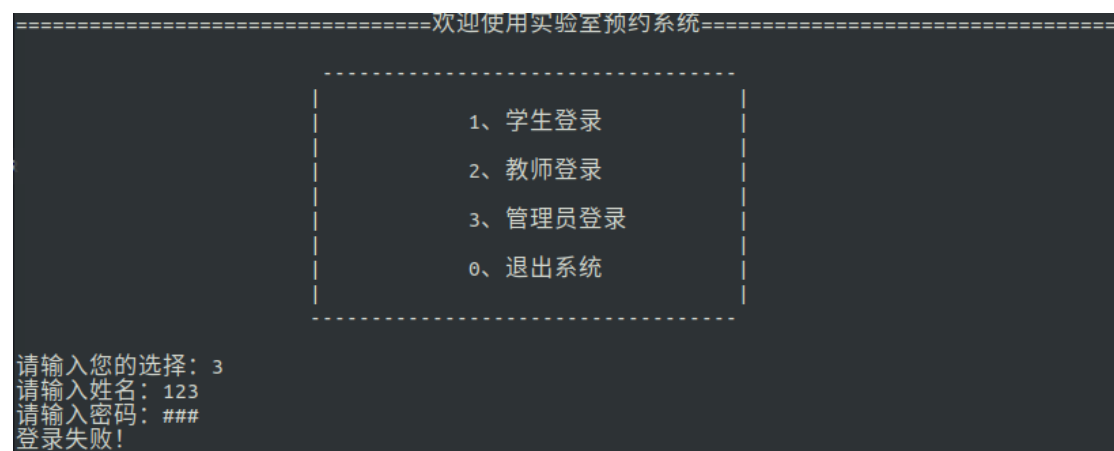
用例编号	用例	期望语句	实际语句
1	(ad1, 111)	A	A
2	(123, ###)	B	B

测试用例结果截图

用例 1



用例 2

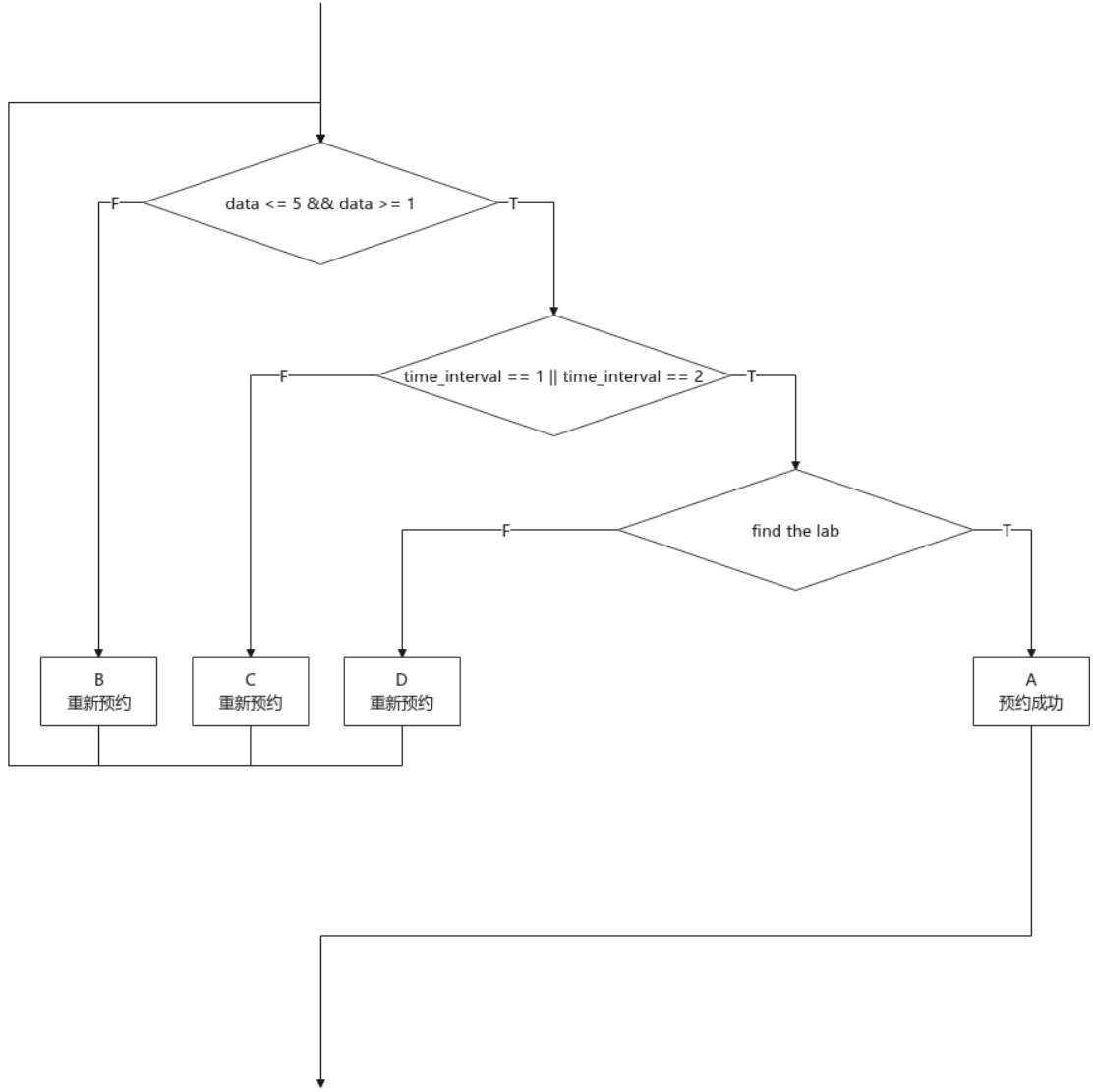


5.2.4 预约模块测试

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

5.2.5 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;
}

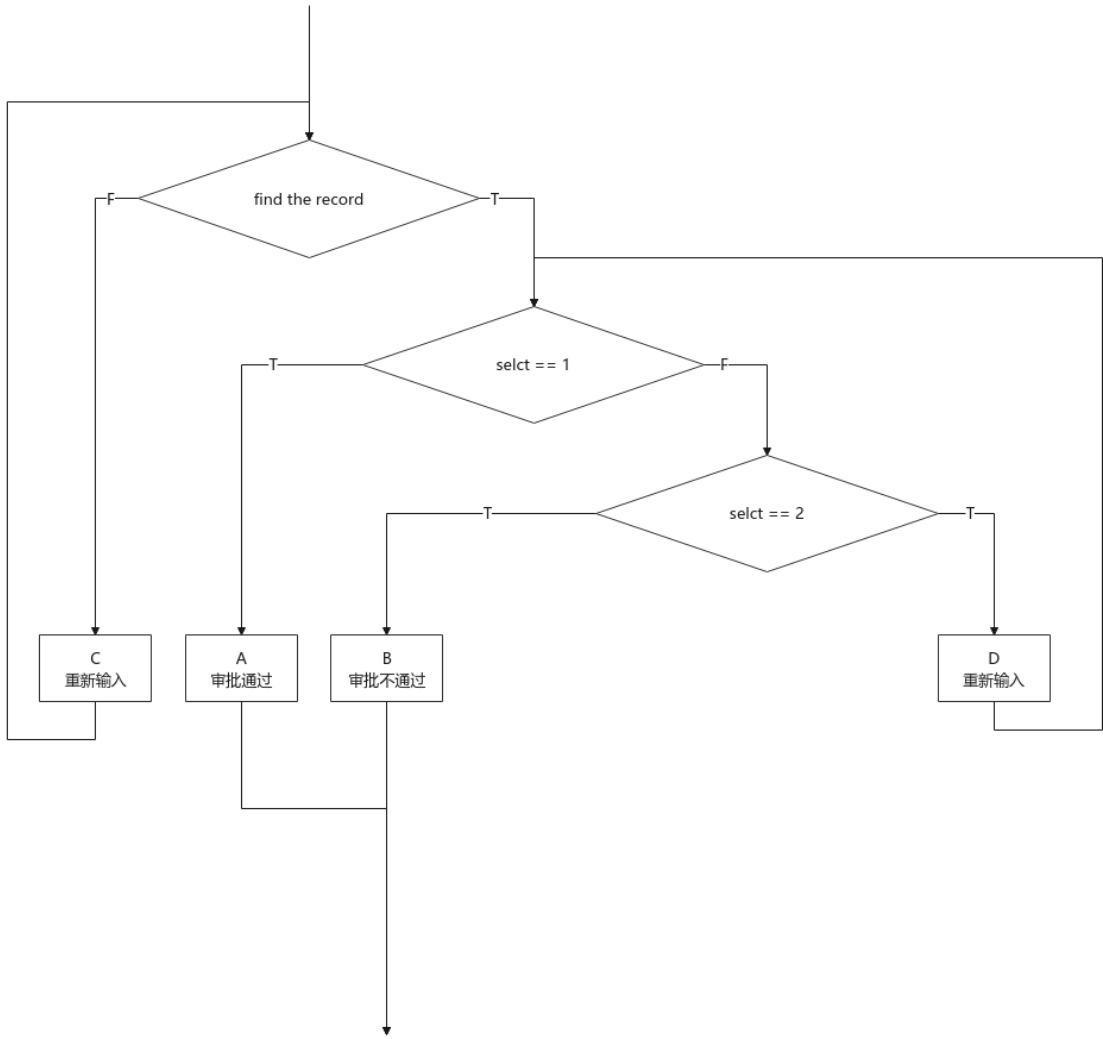
cout << "    1、通过    2、不通过    " << endl;

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

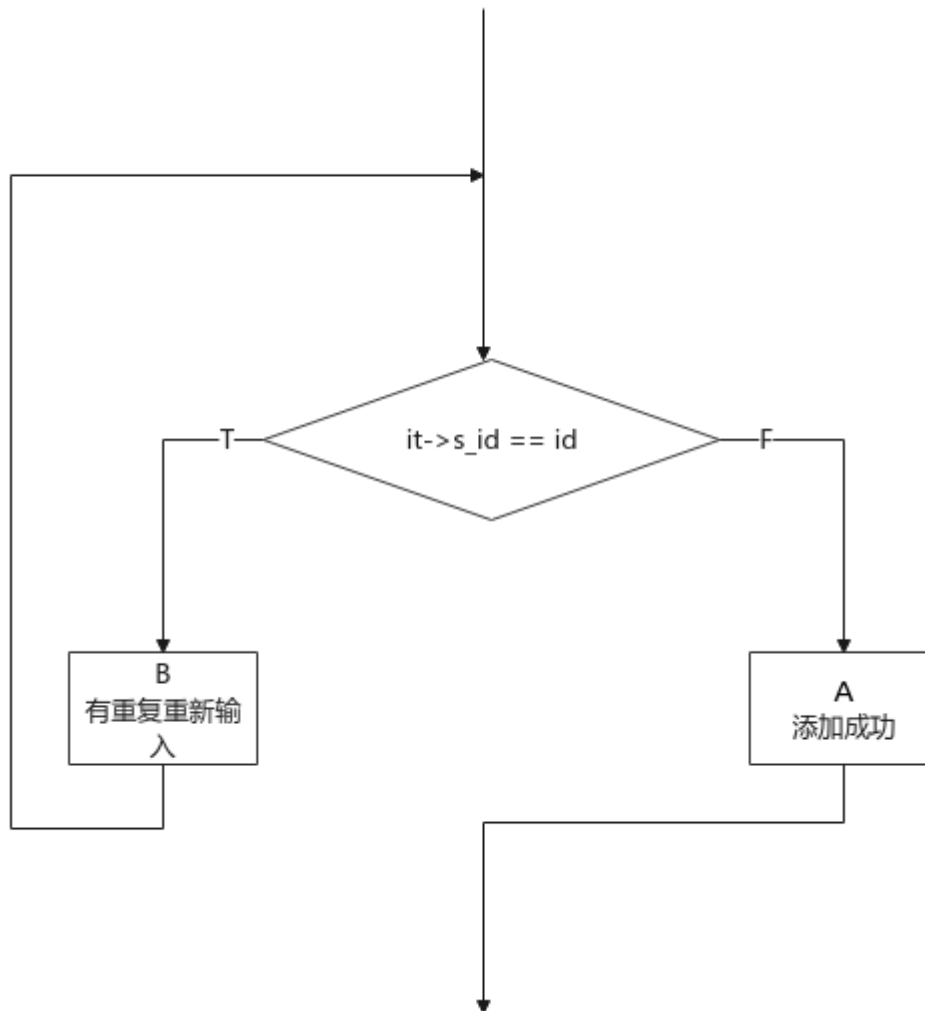
输入有误，请重新选择

5.2.6 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

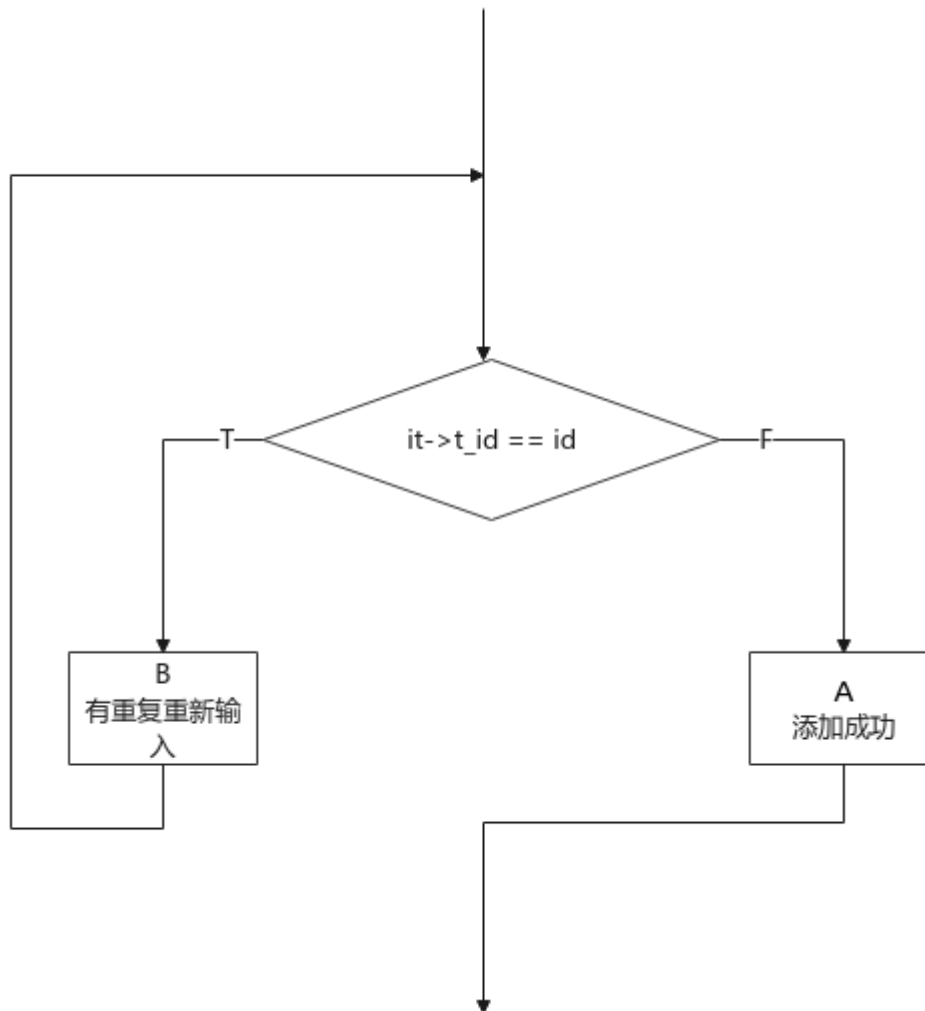
```
请输入选项: 1
请输入学号: 01
学号重复,请重新输入
```

5.2.7 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

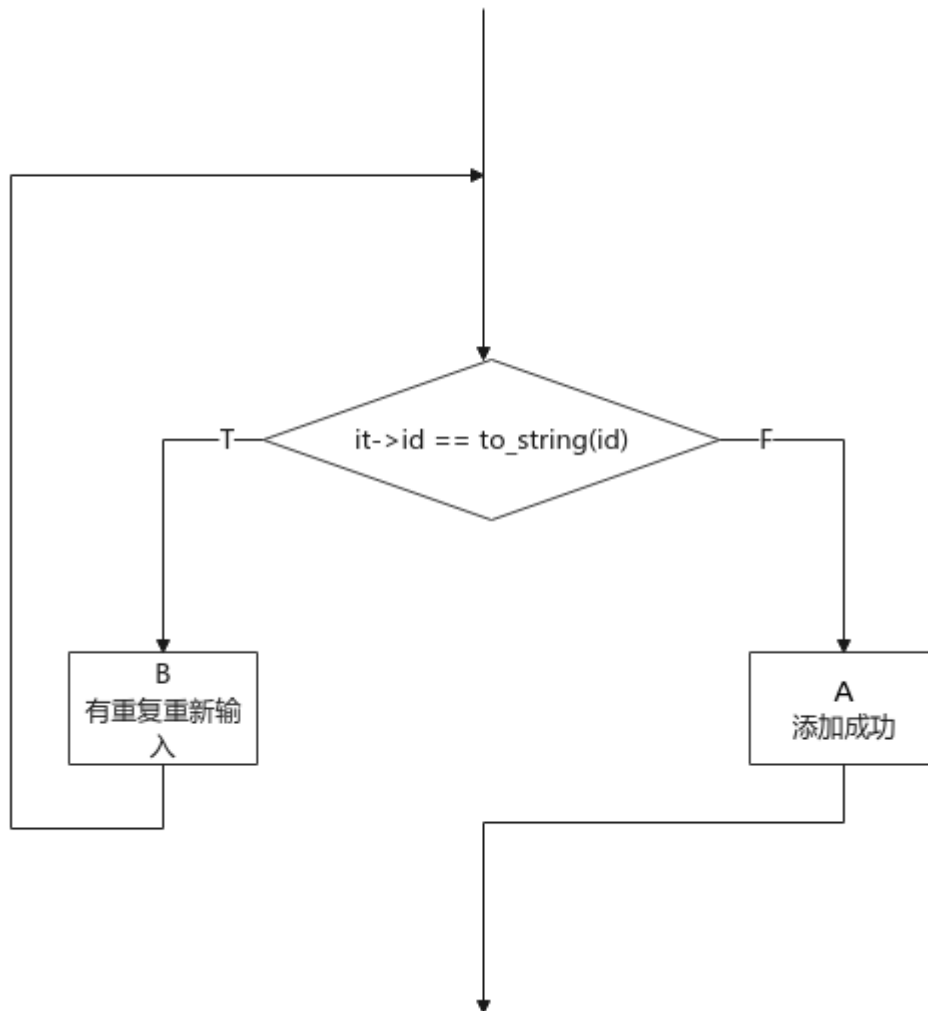
```
请输入职工号：01
职工号重复，请重新输入
请输入职工号：01
```

5.2.8 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
```

6 项目总结

本次项目使用结构化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，我对结构化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。