**其他需求**

**补充性规格说明**

**概述**

本部分主要记录了学生选课系统所有未在用例中描述的需求，主要包括跨越多个用例未被提到的功能性需求、可用性、可靠性、性能和可支持性。

**功能性**

1.日志和错误处理

在持久性存储中记录所有错误，在日志中记录每条记录。

2.安全性

任何使用都需要经过用户认证

**可用性**

1. 学生应该花费较少的时间成本选到自己想选的课程。因此：

l 必选课和可选课要进行分区，且分区要明确。

l 选课按钮用与背景不同色，能迅速察觉。

l 课程重要信息做加粗处理，如课程名称。

l 无需学习成本，所设界面安排、功能点安排符合习惯认知。

2. 便捷、无错的退选课操作处理及其重要，因为学生希望能尽快无误的选到自己的心仪课程，否则会带来很差的使用体验。

**可靠性**

1. 选课系统在选课开放时间段内，在每周七天，每天二十四小时内都应该是可以使用的，宕机的时间应小于10%。

2. 如果系统在任何时候出现错误，为了信息的一致性，要保证所有选退课事件都能从场景的任何一步中完全恢复。具体恢复原理要进行深入分析...

3. 系统在学生选课系统使用的高峰期（选课系统刚开放的时候）能够应对相应的数据吞吐量。

**性能**

1. 在任意既定时刻，系统最多可支持2000名用户同时登陆系统并进行退选课操作。

2. 系统必须能够在2分钟内完成所有事务的80%。

**可支持性**

1. 课程信息应该符合培养方案要求。

2. 排课时间、地点符合学校教学安排，不能出现冲突。

**安全性**

1. 系统必须能防止学生修改他人的课程表以及教师修改其他教师所开设的课程。

2. 只有教师才能输入学生的成绩。

3. 只有管理员才能更改学生的信息、教师的信息。

4. 至于教师是否有权限修改自己所开设的课，有待进一步讨论...

**设计约束**

1. 该系统应继承现有系统的课程数据库、学生信息和教师信息等内容，在现有系统数据基础上进行相应更改。

2. 系统需要支持手机端和电脑端访问。

**接口**

1. 软件接口：提供与之相关的数据库的访问接口。

2. 无需硬件接口和通信接口。

**设想**

**概述**

为简单的执行概要文件，用于快速了解项目的主要思想。我们设想该学生选课系统为现有系统的改进版，具有灵活性以支持学生的退选课操作。

**定位**

1. 问题综述

现有的选课系统功能分区不符合用户使用习惯；并发量较差，系统经常崩溃；且系统经常出现莫名其妙的Bug，影响正常的退选课操作，学生与教师使用体验极差。以上这些问题使得现有系统不再适合选课需要，因此需要新选课系统来改变这些情形。

2. 系统定位综述

该系统开发意愿是为了给本校师生使用，后续功能完备，获得市场认可，不排除在市面上进行销售推广的可能。

**涉众描述**

本系统并无特殊的高阶目标。

用户级目标

用户要求系统实现一下目标：

学生：退选课、查看课程信息、查看课表、查看成绩 ……

教师：查看所授课程信息、查看教学班信息、录入成绩 ……

管理员：管理学生信息、管理教师信息、管理课程信息、维护安全性 ……

**产品概览**

1. 系统展望

该系统开发意愿是为了给本校师生使用，后续功能完备，获得市场认可，不排除在市面上进行销售推广的可能。

2. 优点概述

项较原系统，有较好的差错控制措施，能支持更多的并发量。

**词汇表（数据字典）**

|  |  |  |
| --- | --- | --- |
| 术语 | 定义和信息 | 格式 |
| 公选课 | 是全校自由选课，一般是一学期期末或者是开学的第一周开始选。 |  |
| 专业选修课 | 与专业相关的一类选修科的统称，一般由学校根据该校学生的知识水平能力和兴趣确定。 |  |
| 课容量 | 每门课规定的能容纳学生的最大数量。 | int类数值 |
| 培养方案 | 为每个学校为各个专业设定的每个专业需要学习的课程和选够的学分，选课系统的课程安排必须严格按照培养方案而定。 |  |
| 学分 | 是计算学生学习量的一种计量单位，按学期计算，每门课程及实践环节的具体学分以专业教学计划的规定为准。 |  |
| 课程号 | 每所学校每门课都有独一无二的课程号用于标示。 | 20位数字代码（部分学校课总数多，可适当增加） |

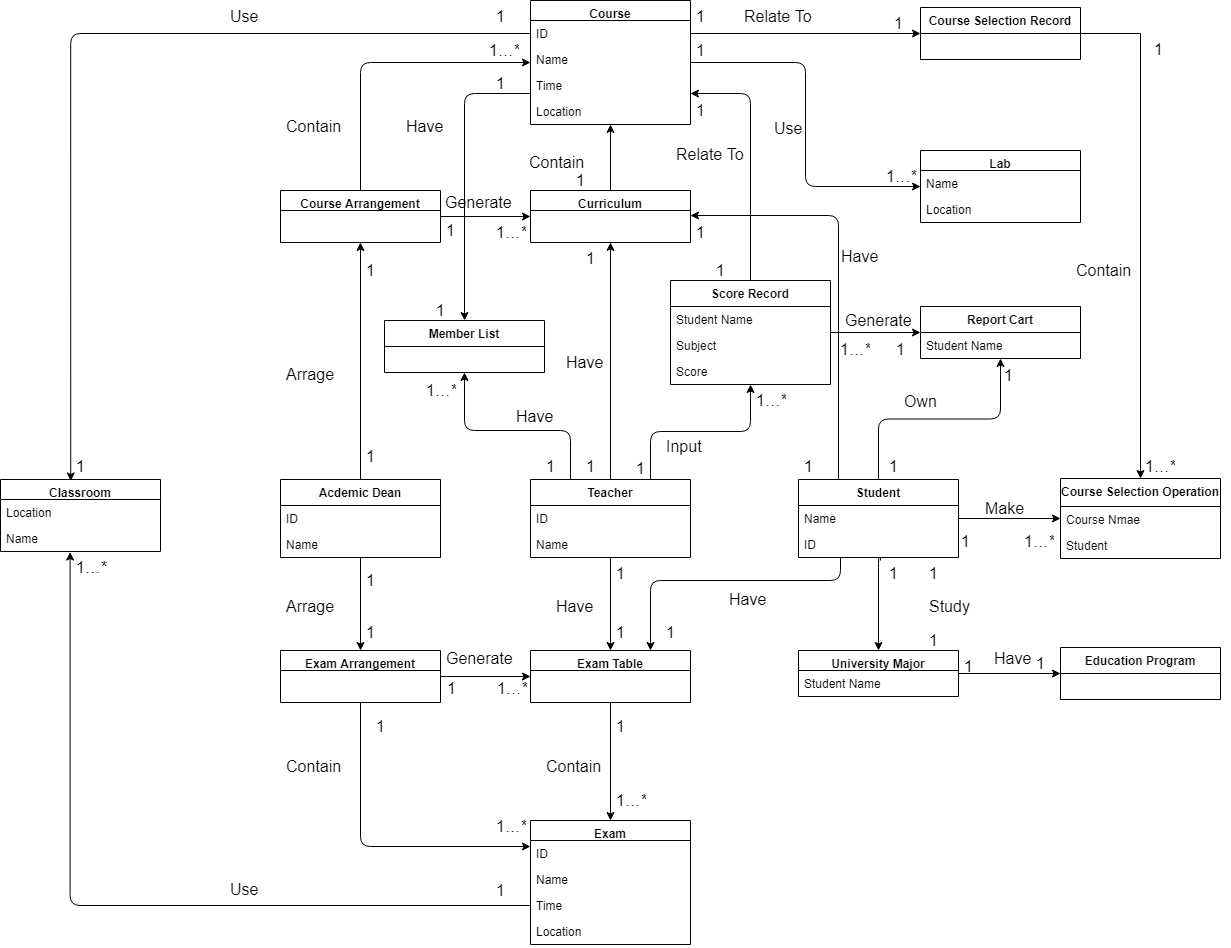
**领域规则**

|  |  |  |  |
| --- | --- | --- | --- |
| ID | 规则 | 可变性 | 来源 |
| 规则1 | 学生显示的选修课需要满足培养方案上的要求 | 高，学校每年都可能根据实际情况调整某专业的培养方案 | 课程管理实际要求 |
| 规则2 | 学生四年的学分数应该满足毕业要求 | 低 | 学生管理实际要求 |
| 规则3 | 需根据同一时段的课程，为每个班安排空闲教师，注意不能冲突。 | 低，为正常教学开展必须满足的要求 | 正常教学开展要求 |
| 规则4 | 为每个专业不同的课安排教学时间，注意必选课不能在时间上有任何冲突，专业选修课不能有大规模的冲突。 | 低，正常教学开展必须满足的要求 | 正常教学开展要求 |
| 规则5 | 教师一学年的教时有一定标准 | 可能每年学校会根据现实情况和老师职称的变化对教时区间作出新的调整 | 老师管理实际要求 |
| 规则6 | 每门课的课容量要结合教学效果、教师数量和本专业人数谨慎安排 | 高，每一年专业人数会发生变化，且也有教师数量增加等等实际情况 | 正常教学开展要求 |

**领域模型设计**

### 图示

**领域模型设计图如下所示：**



### 概念类分类列表

|  |  |  |  |
| --- | --- | --- | --- |
| 概念类 | 中文名 | 从属 | 解释 |
| Student | 学生 | 人员部分 | 储存有学生的基本信息 |
| Teacher | 教师 | 人员部分 | 储存有教师的基本信息 |
| Academic Dean | 教务员 | 人员部分 | 储存有教务员的基本信息 |
| Course | 课程 | 课程安排部分 | 指代某一具体课程，记录有课程的名字，上课的时间与地点 |
| Course Arrangement | 课程安排 | 课程安排部分 | 指代全校整体课程安排，由教务员进行统一的编排与修改 |
| Curriculum | 课表 | 课程安排部分 | 指代某一具体学生活老师的课表，上面记录有该用户相关的课程 |
| Member List | 人员名单 | 课程安排部分 | 指代某一具体课程的与课学生名单 |
| Score Record | 评分记录 | 成绩部分 | 指代某一条评分记录，记录有评分课程、相关教师、被评分学生，成绩 |
| Report Cart | 成绩单 | 成绩部分 | 指代包含某一学生的所有课程成绩的成绩单 |
| Course Selection Operation | 选课操作 | 选课部分 | 指代某一条的具体的选课操作，包括有选课恶学生，所选的课程名称，退课则通过删除选课记录实现 |
| Course Selection Record | 选课记录 | 选课部分 | 指代所有选课操作的记录信息 |
| Exam Arrangement | 考试安排 | 考试部分 | 指代全校整体考试安排，由教务员进行统一的编排与修改 |
| Exam Table | 考表 | 考试部分 | 指代某一具体学生活老师的考试有关信息项 |
| Exam | 考试 | 考试部分 | 指代某一具体考试，记录有考试的名字、时间与地点 |
| University Major | 专业 | 学籍部分 | 储存相关专业的信息，包含有专业的名称 |
| Education Program | 培养方案 | 学籍部分 | 储存相关专业的培养方案内容 |
| Classroom | 教室 | 其他 | 指代某一具体的教室 |
| Lab | 实验室 | 其他 | 指代某一具体的实验室 |

**通信图及顺序图设计**

### 学生退、补、选课

1. 学生登录账号，进入学生选课系统

2. 点击选课按钮进入选课界面

3. 系统按公选课、专业选修课、体育课分类，并逐行显示课程。每个课程条目标明任课老师，上课时间，上课类型，学分

4. 条目右侧有选择课程按钮，点击选择按钮选择该课程

5. 点击后会有提示信息：选课成功或者选课失败及原因。点击确认后，系统刷新页面，更新页面信息

6. 在页面下方，会显示已经选择的课程，课程条目右侧有退课按钮，点击退课按钮退出该课程

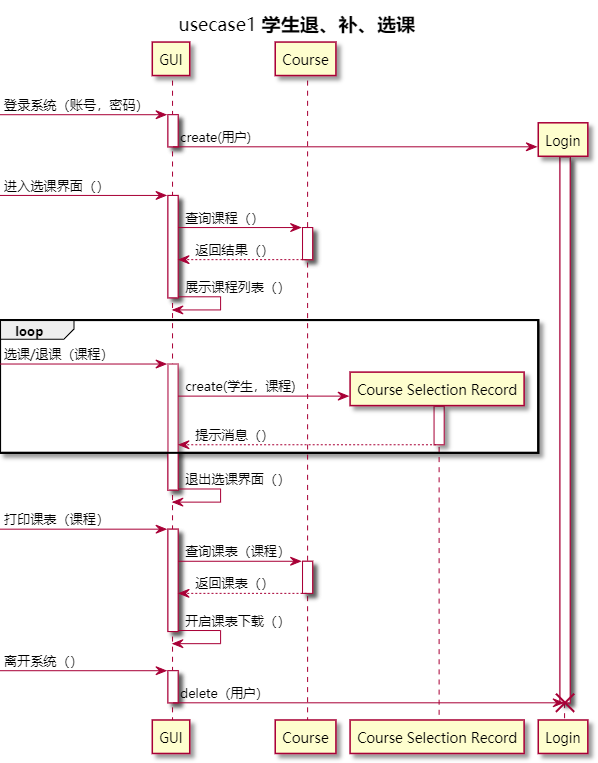
7. 点击后会有提示信息：退课成功或者退课失败及原因。点击确认后，系统刷新页面，更新页面信息

重复3-5步直至选课结束

系统根据操作结果执行5步和6步

8. 点击退出选课按钮安全退出选课界面

9. 退出后点击课表，可以看到已经选择的课程及课表。

10. 点击打印课表，下载成功后离开系统

### 学生查看选课结果

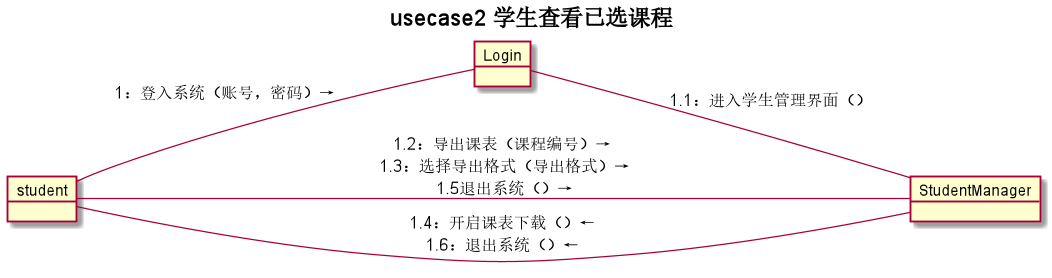
1. 学生进入教务系统

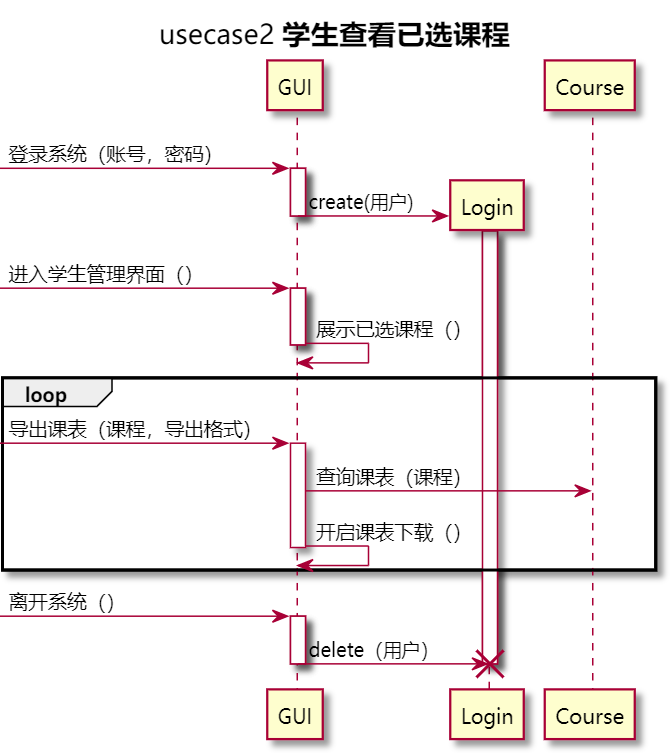
2. 学生进入学生管理部分

3. 学生点击课表，查看已选课程

4. 学生可以以多种格式导出课表

5. 学生退出系统





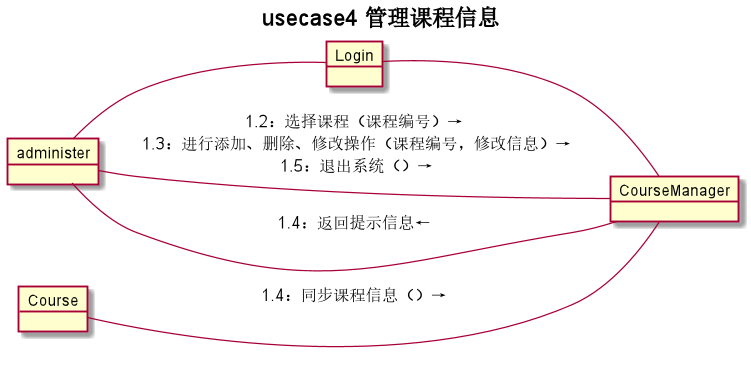
### 管理员添加、修改、删除课程

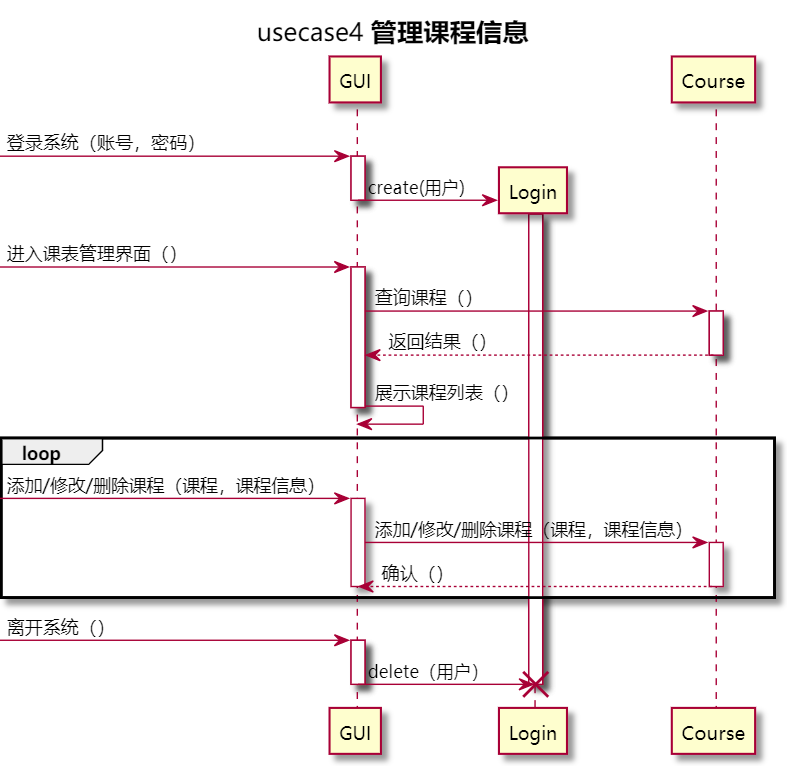
1. 管理员登录系统

2. 进行添加、删除、修改课程信息

3. 重复过程2，3直至完成工作

4. 退出系统





### 管理员安排课程

1. 管理员登录系统

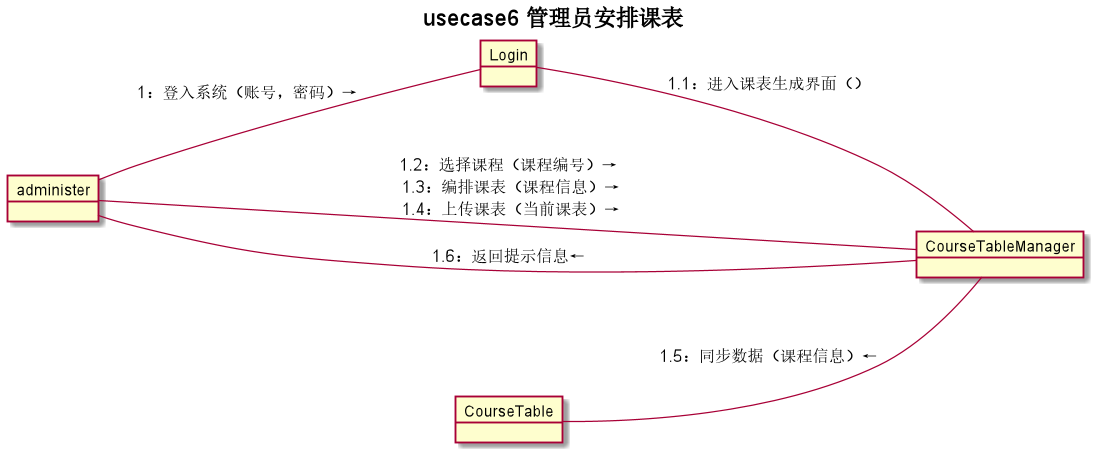
2. 选取课程

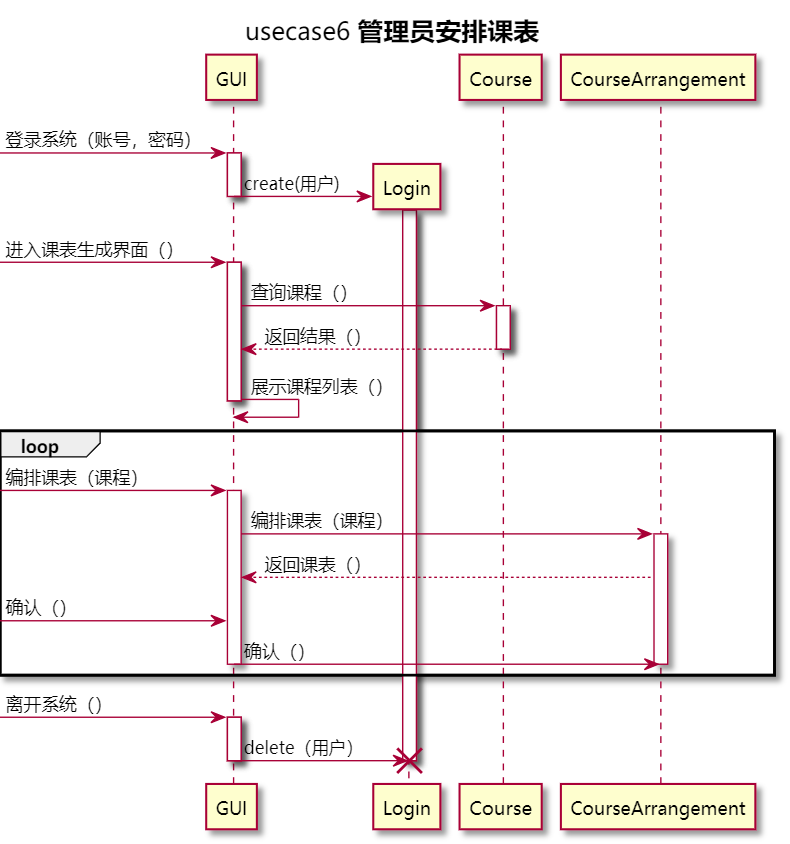
3. 编排课表

4. 上传课表

5. 重复2-4

6. 退出系统





### 学生（教师）导出课表

1. 学生登录账号，进入学生选课系统

2. 学生进入课表界面

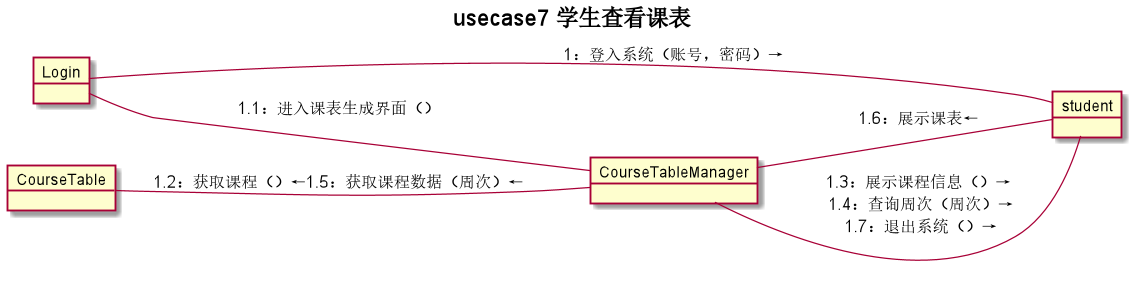
3. 学生选择周次

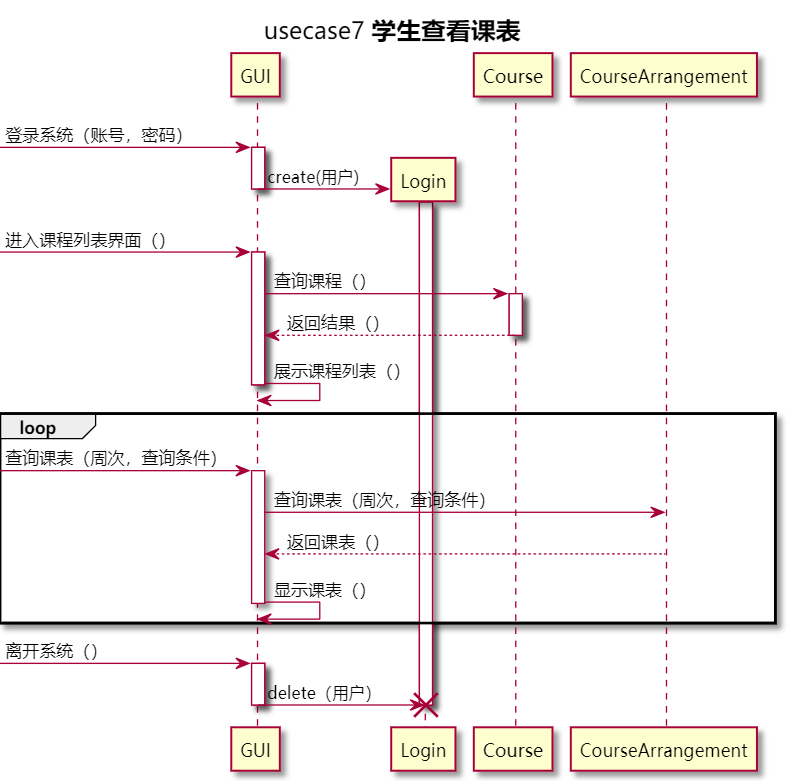
4. 学生进行课表查询

5. 系统根据数据库信息生成课表

6. 学生查看课表

7. 学生退出课表界面





### 管理员修改课表

1. 管理员登录账号，进入学生选课系统

2. 管理员进入课表界面

3. 管理员选择学期、周次、课程

4. 管理员进行课表查询

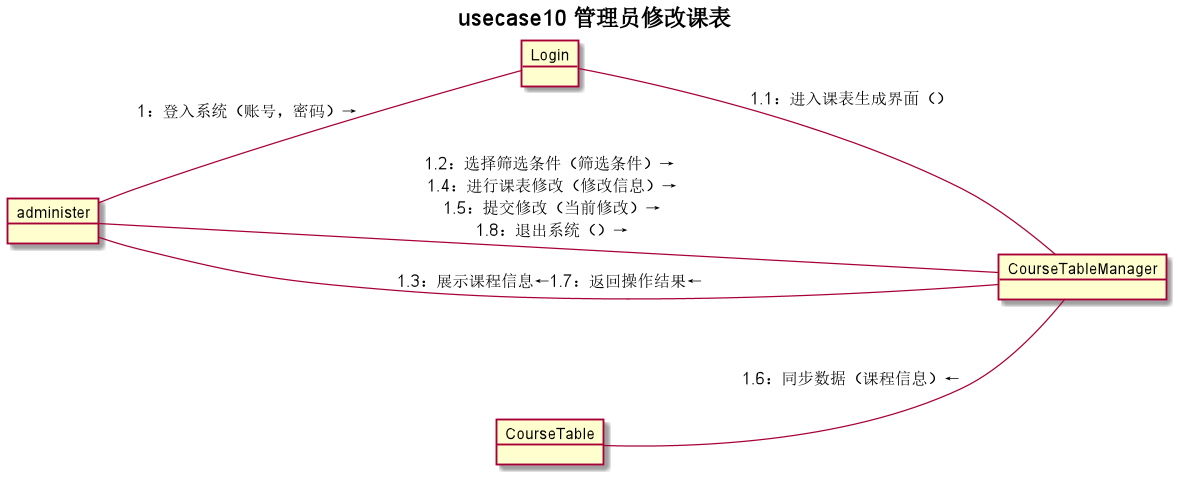
5. 系统根据数据库信息生成课表

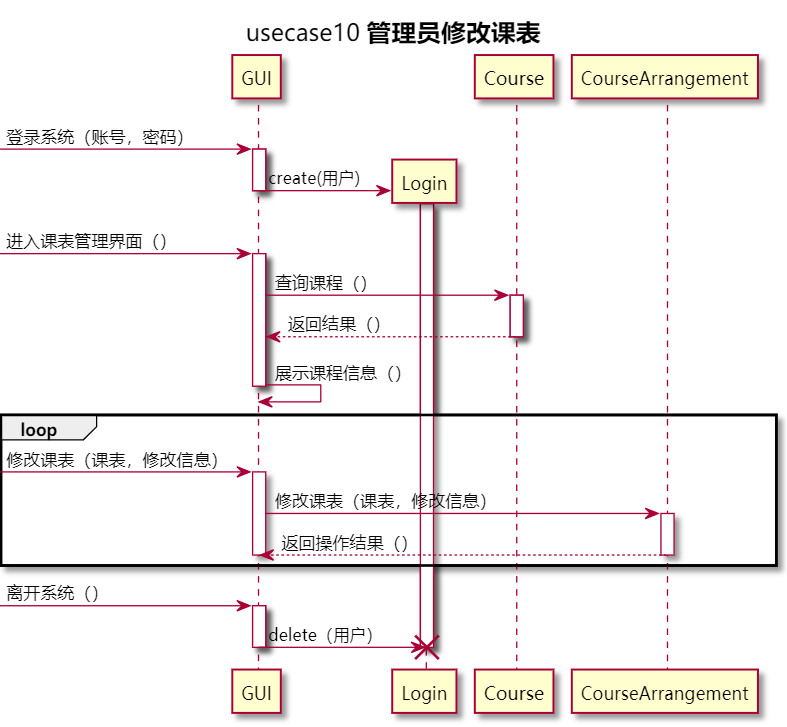
6. 管理员查看课表

7. 管理员对课表进行修改

8. 系统向数据库写回修改

9. 管理员退出选课系统





教师导出学生名册

1. 教师登录账号，进入选课系统

2. 教师进入课程列表

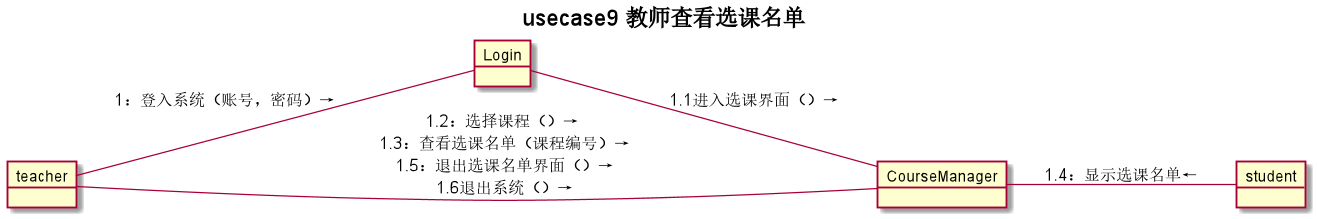
3. 教师选择课程

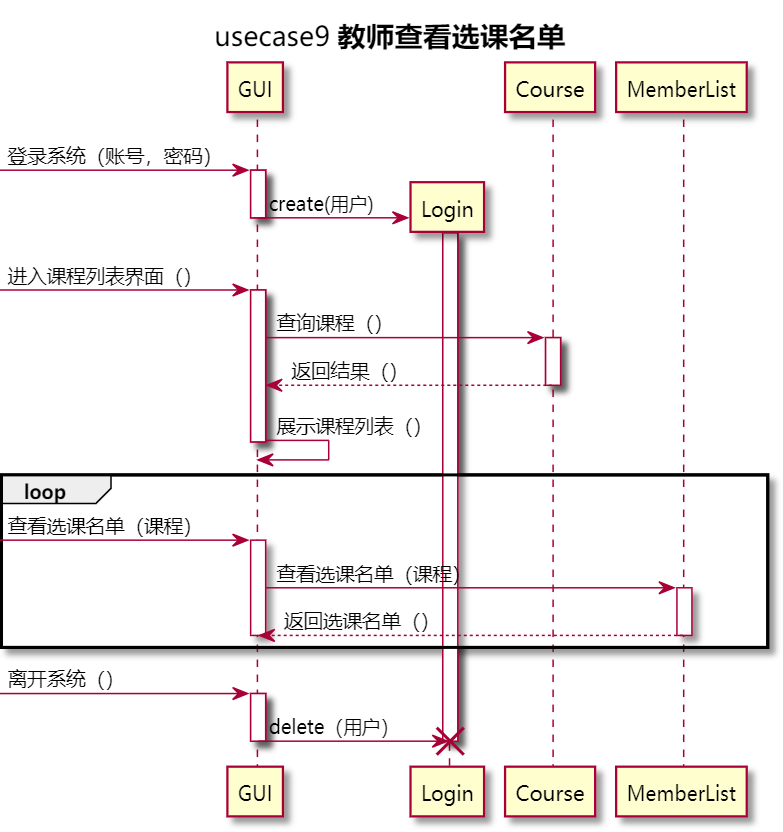
4. 教师查看选课名单

5. 系统根据数据库信息生成选课名单

6. 教师查看选课名单

7. 教师退出课程界面





### 教师使用上课教室、实验室

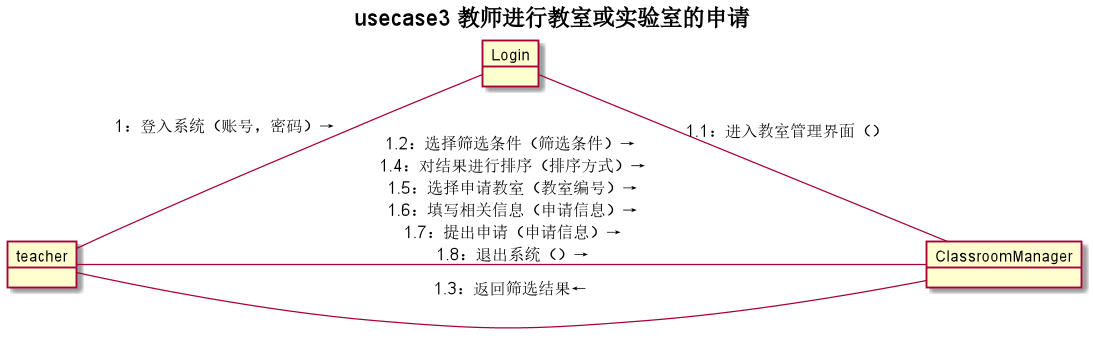
1. 教师进入教务系统

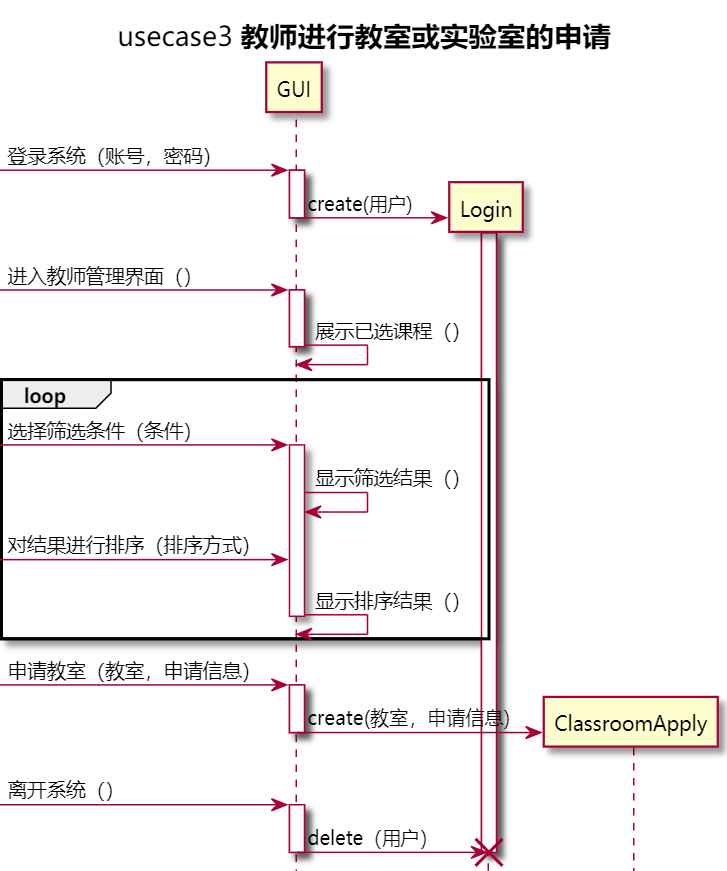
2. 教师进入教室管理部分

3. 教师点击教室页面

4. 教师选择筛选条件，查看筛选结果

5. 教师对结果进行排序





### 教师提交开课申请

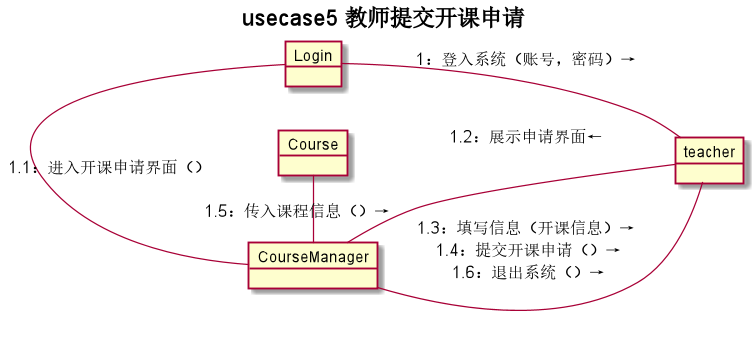
1. 教师登录系统

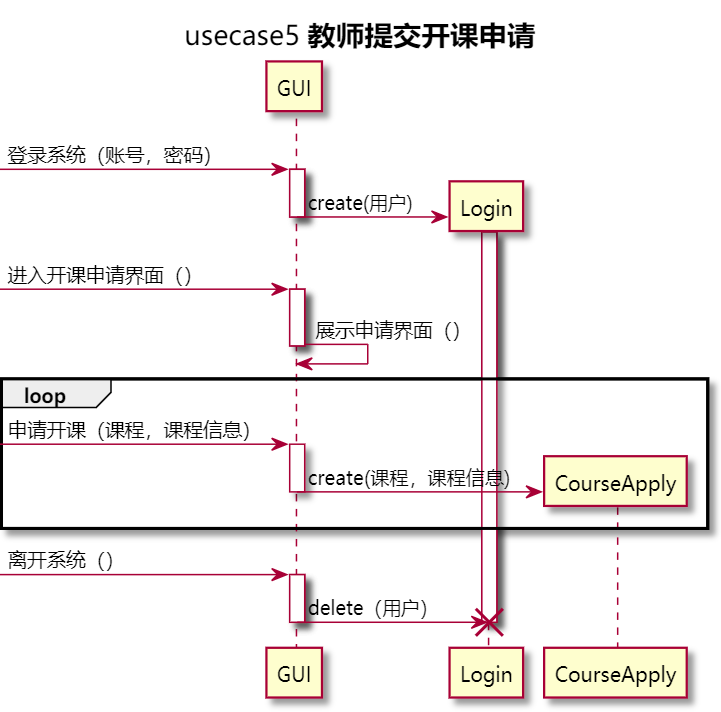
2. 提交课程名，课时安排，课程课件，参考教材等信息

3. 提交开课申请申请

4. 重复2-3

5. 退出系统





### 学生导出考表

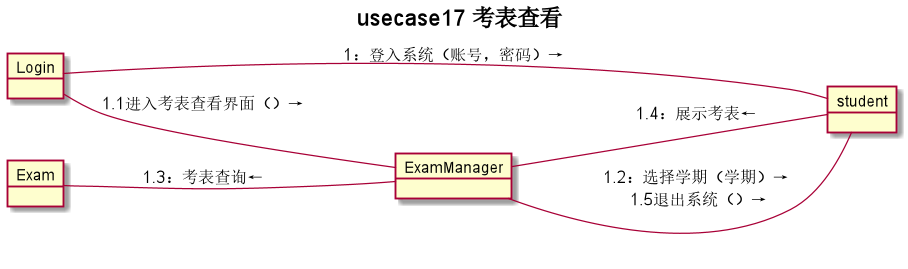
1. 教师或学生登录系统

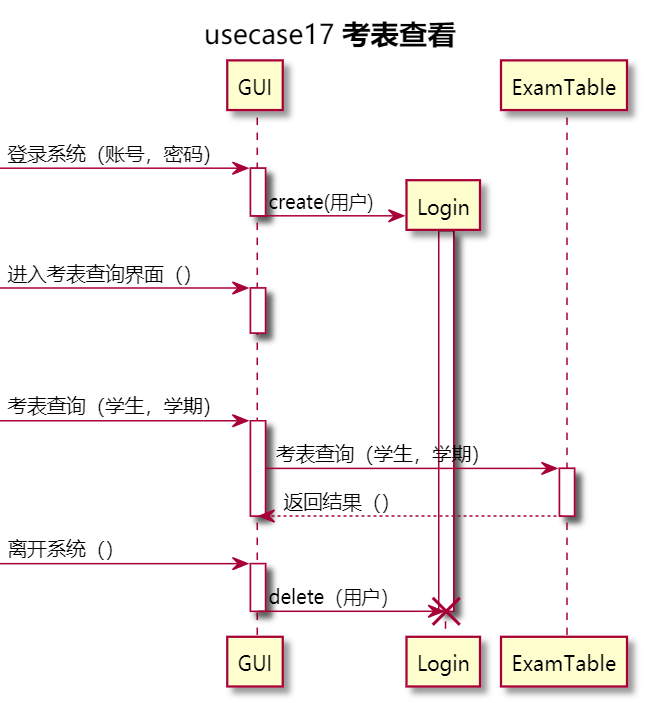
2. 教师或学生点击“考试安排查询”按钮

3. 教师或学生选择“学年学期”并选择“学期类型”

4. 教师或学生点击“确认”按钮

5. 教师或学生查看与自己相关的考表





### 教师安排考试时间与考试教室

1. 教务员登录系统

2. 教务员导入需要进行考试的课程名单

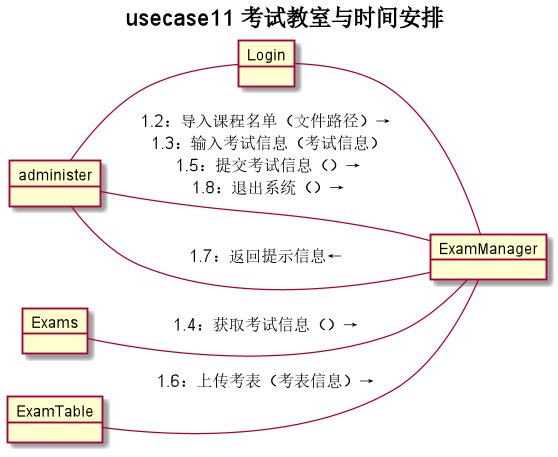
3. 教务员输入各科目的考试时长

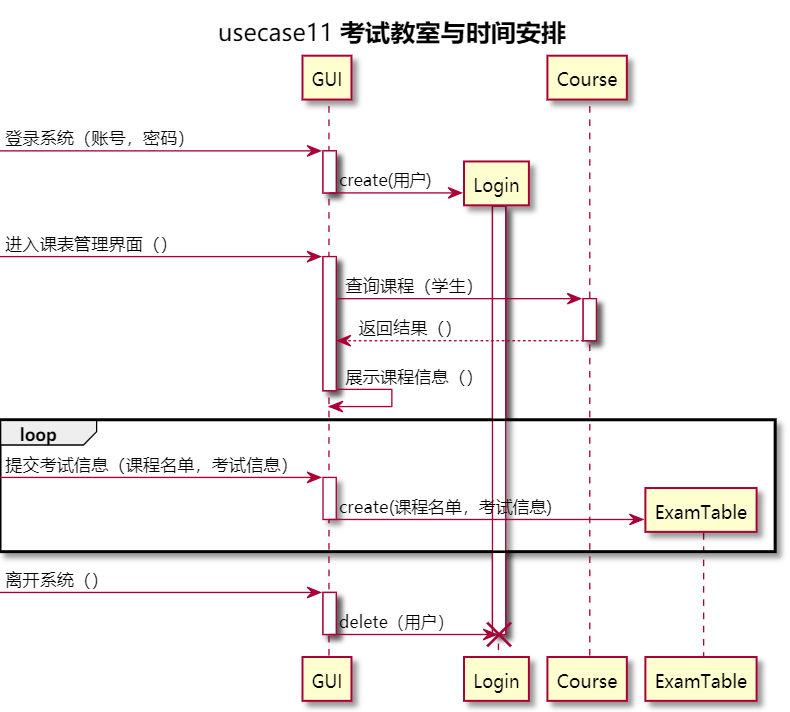
4. 教务员选择考试周范围

5. 系统自动生成考表，并将考表信息导入系统数据库

6. 教务员将考表上传到信息门户

7. 学生、老师可以在信息门户查看所有考试安排或在教务系统上查看与自己相关的考试安排





### 考核形式的申请

1. 教务员在信息门户上发布考核方式申请的文本格式与申请截止时间

2. 教师填写申请文本

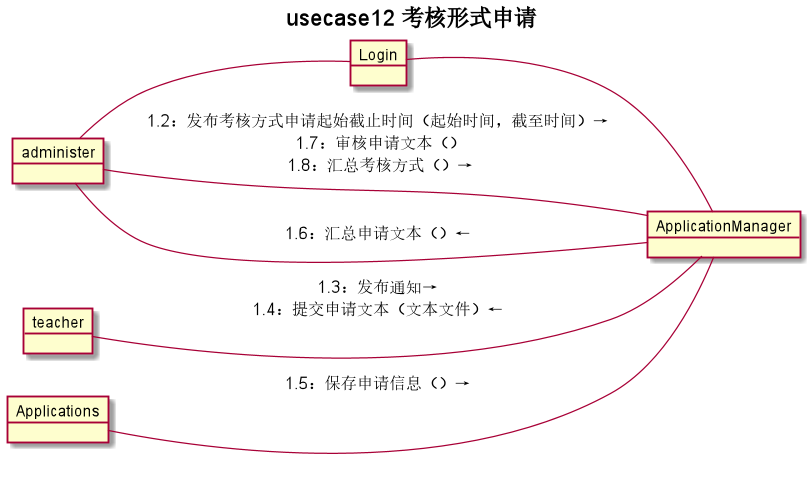
3. 教师、教务员登录教务选课系统

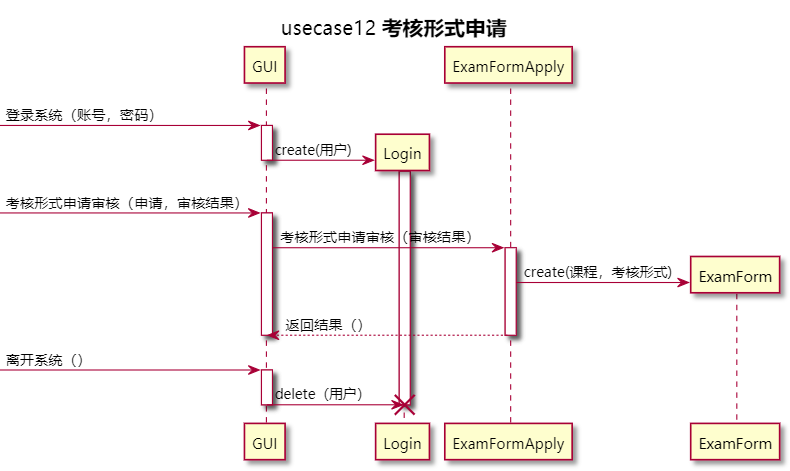
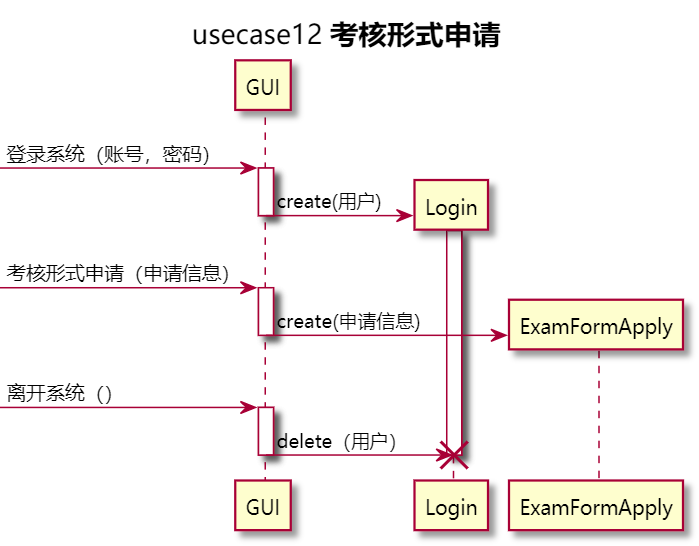
4. 教师提交申请文本

5. 教务员审核申请文本

6. 教师查看考核方式申请处理状态

7. 教务员汇总科目的考核形式





### 学生缓考、免修申请

1. 学生、教务员、教师登录系统

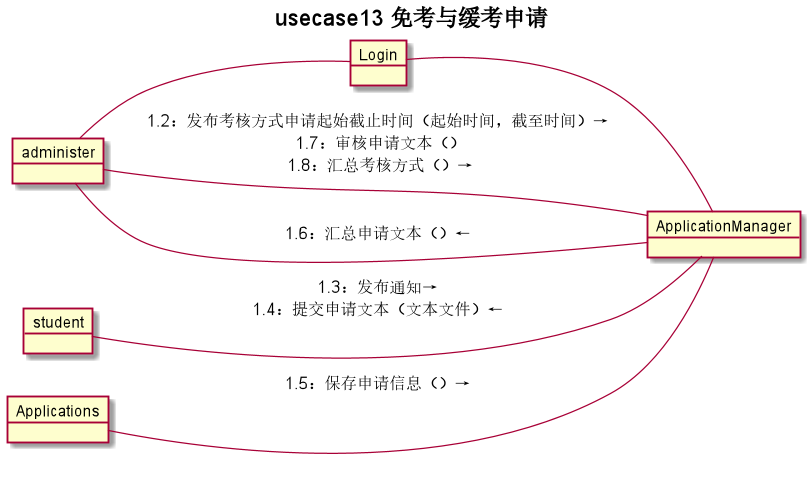
2. 学生在教务系统上选择免修考试科目，并进行免修考试申请

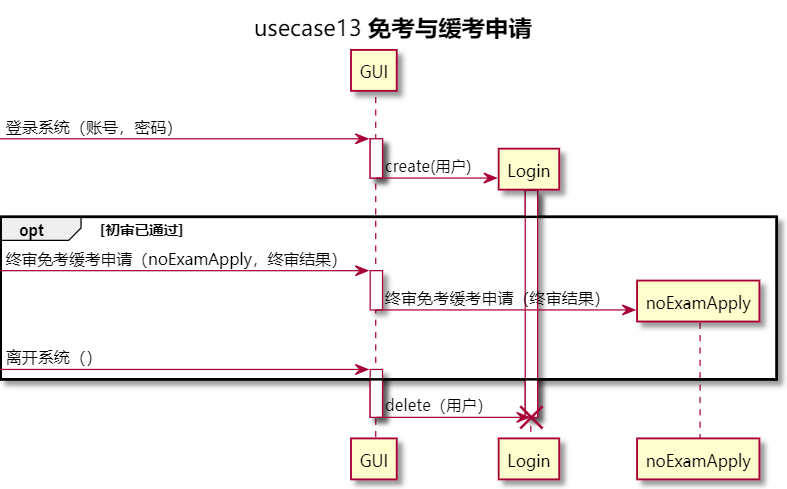
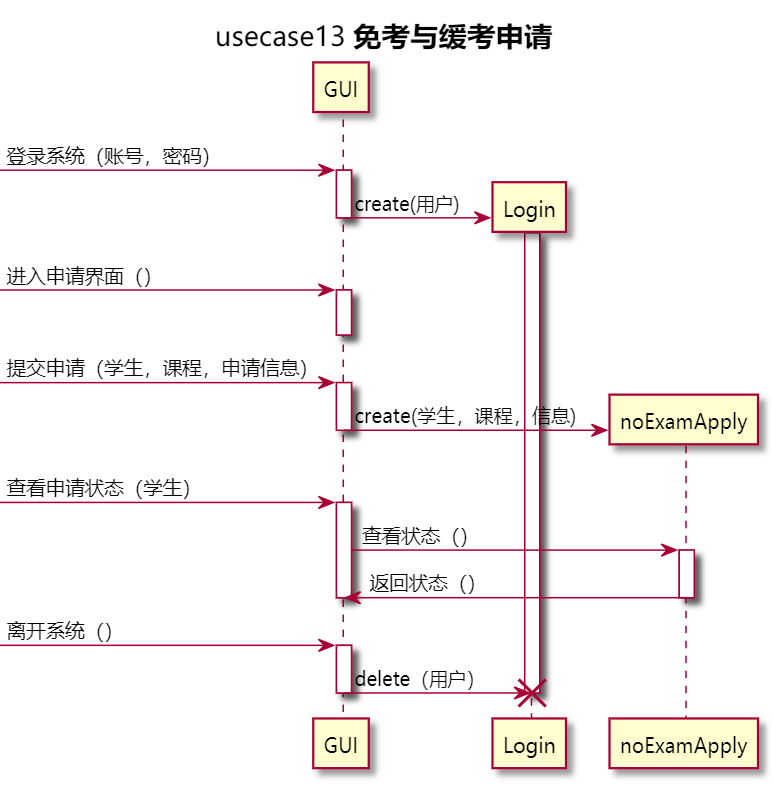
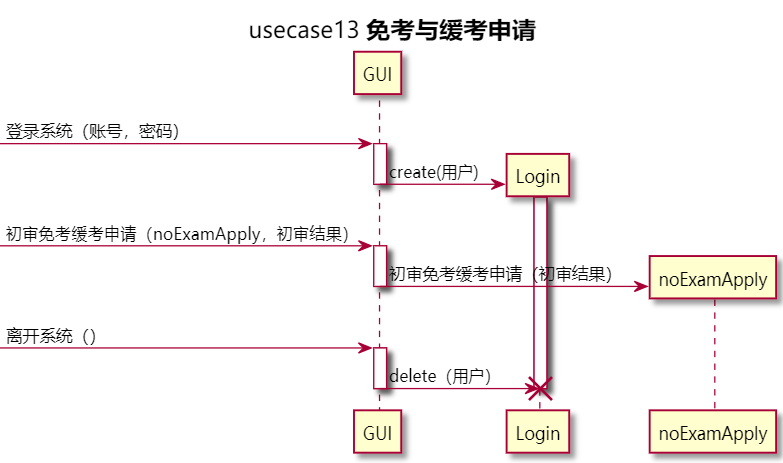
3. 教务员审核免修考试申请

4. 教务员统计免修考试清单，并通知相关老师

5. 教师准备免修考试试卷

6. 教务员安排免修考试的教室与时间





### 补考相关信息的发布

1. 教务员登录系统

2. 教务员导入需要进行考试的课程名单

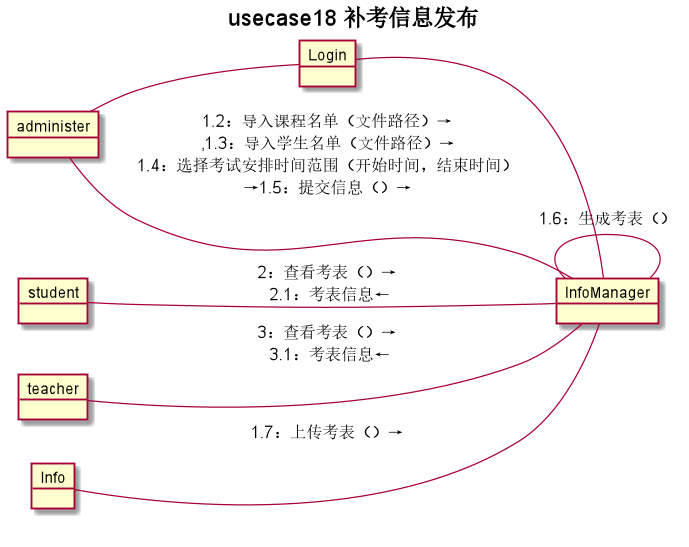
3. 教务员导入需要参与补考地学生名单

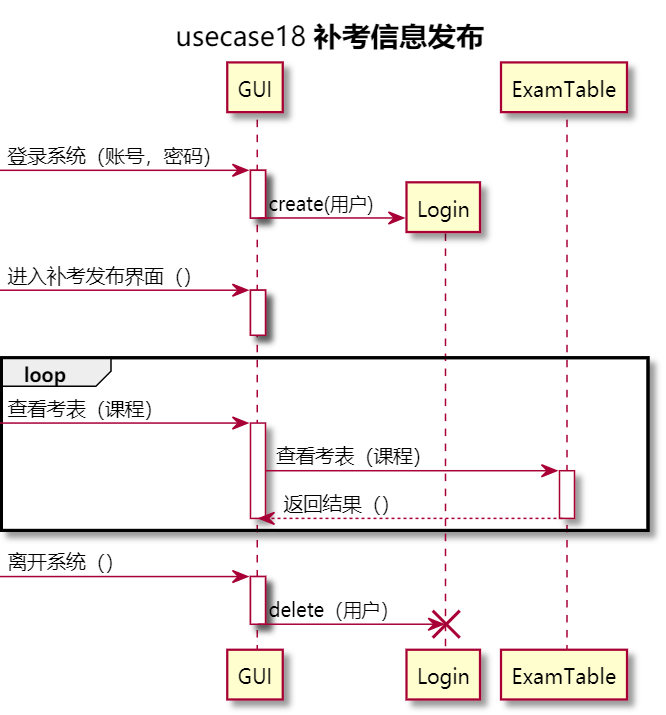
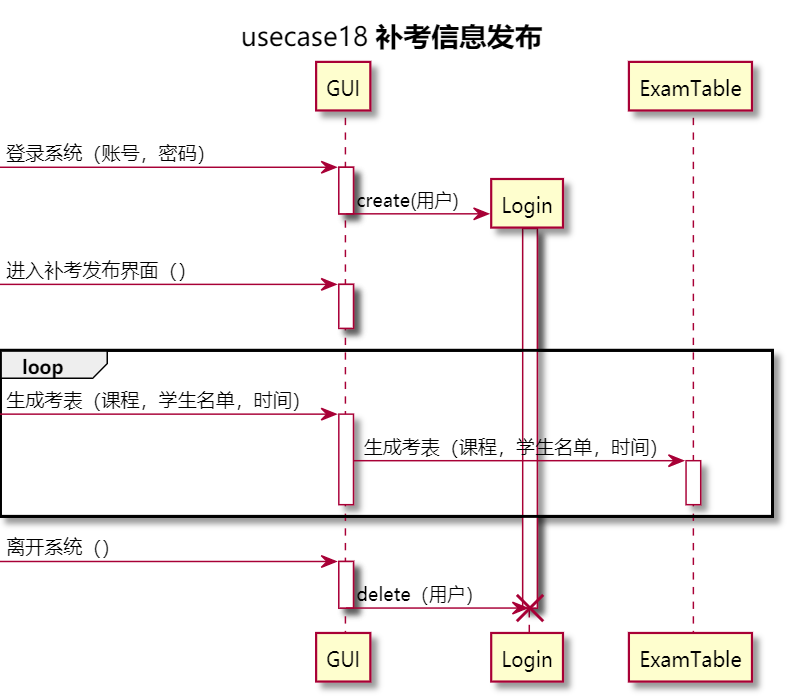
4. 教务员选择考试安排的时间范围

5. 系统自动生成考表，并将考表信息导入系统数据库

6. 教务员将考表上传到信息门户

7. 学生、老师可以在信息门户查看所有考试安排或在教务系统上查看与自己相关的考试





### 免修考试的安排

1. 学生、教务员、教师登录系统

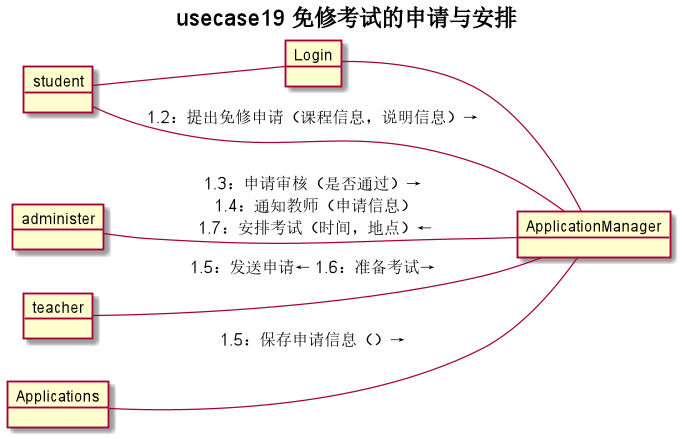
2. 学生在教务系统上选择免修考试科目，并进行免修考试申请

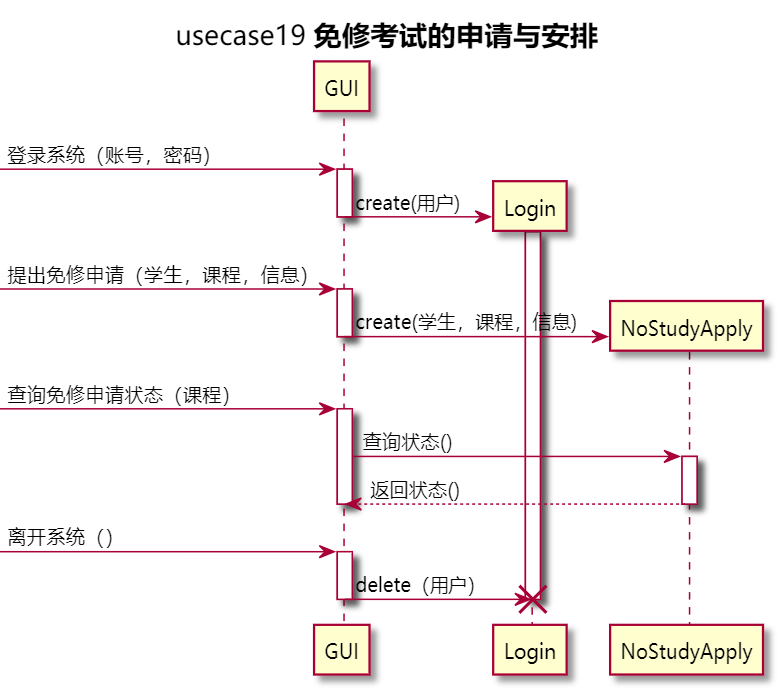
3. 教务员审核免修考试申请

4. 教务员统计免修考试清单，并通知相关老师

5. 教师准备免修考试试卷

6. 教务员安排免修考试的教室与时间





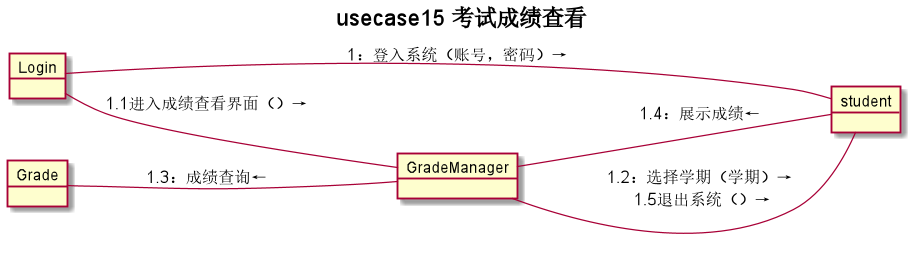
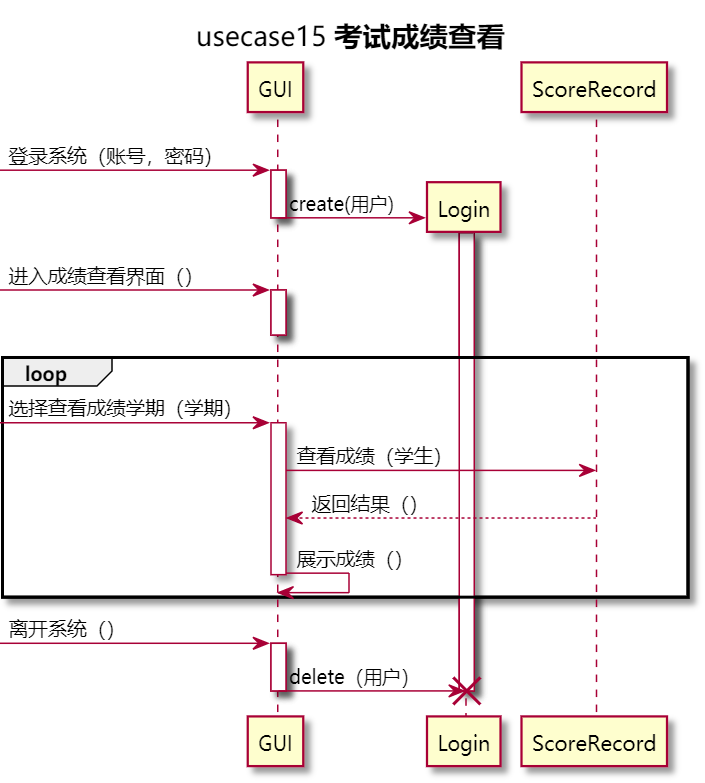
### 学生查看课程成绩

1. 学生登录系统

2. 学生点击“查看考试成绩”选项

3. 学生选择具体的学期

4. 系统显示该学期学生所有已录入成绩科目的成绩



### 考试成绩复查

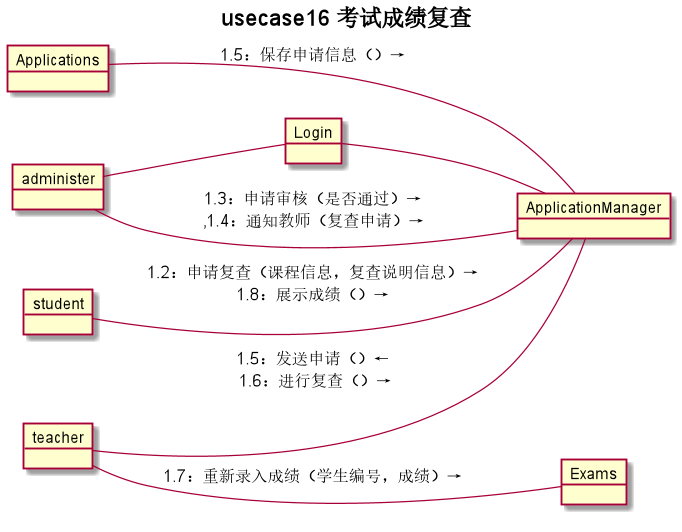
1. 学生、教务员登录系统

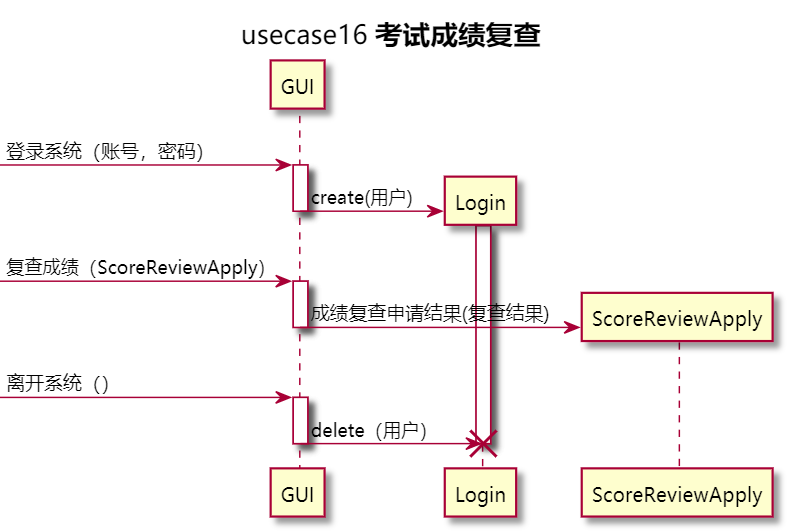
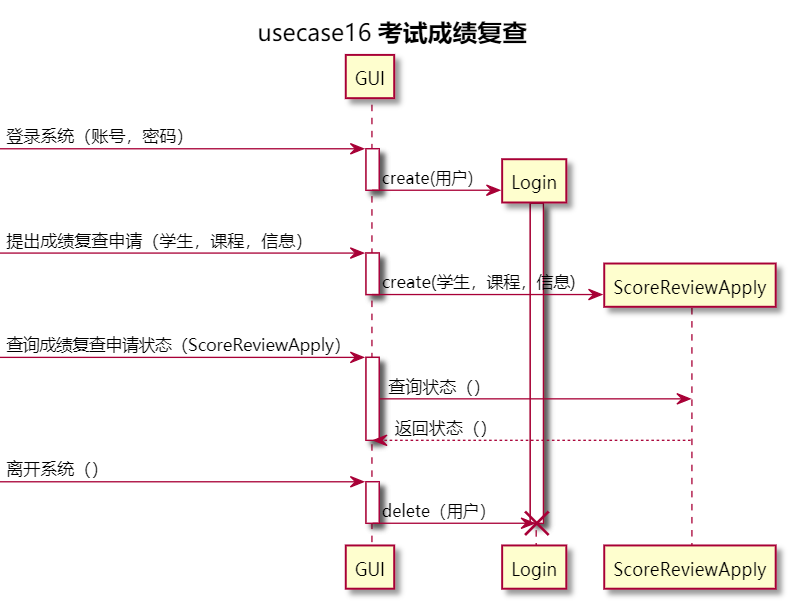
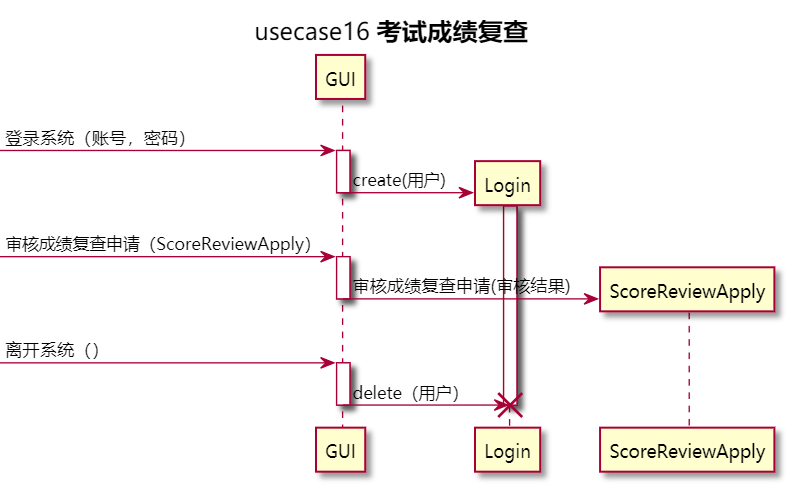
2. 学生在教务系统上提出成绩复查申请

3. 教务员在教务选课系统上通过申请

4. 教务员通知教师准备相关学生的试卷

5. 教务员复查相关试卷并将复查结果填写在教务系统上





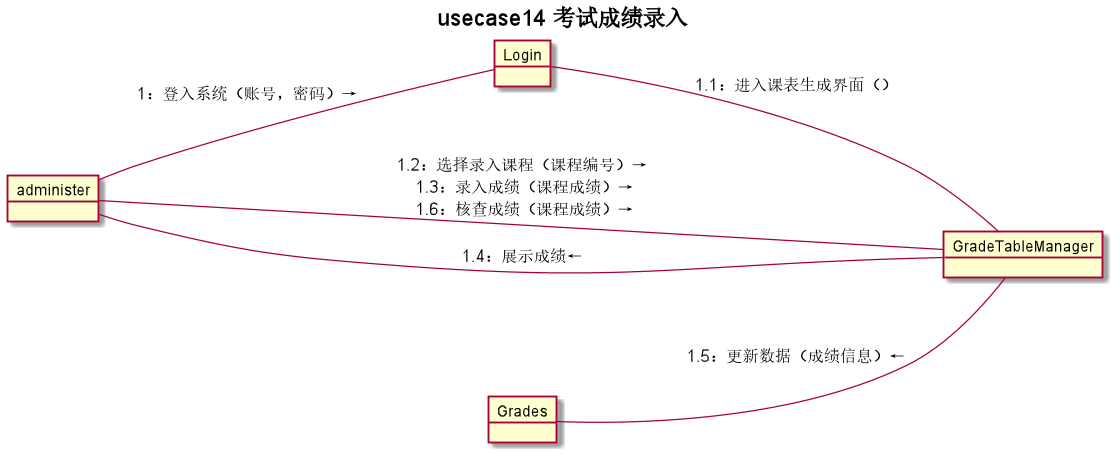
### 教室录入课程成绩

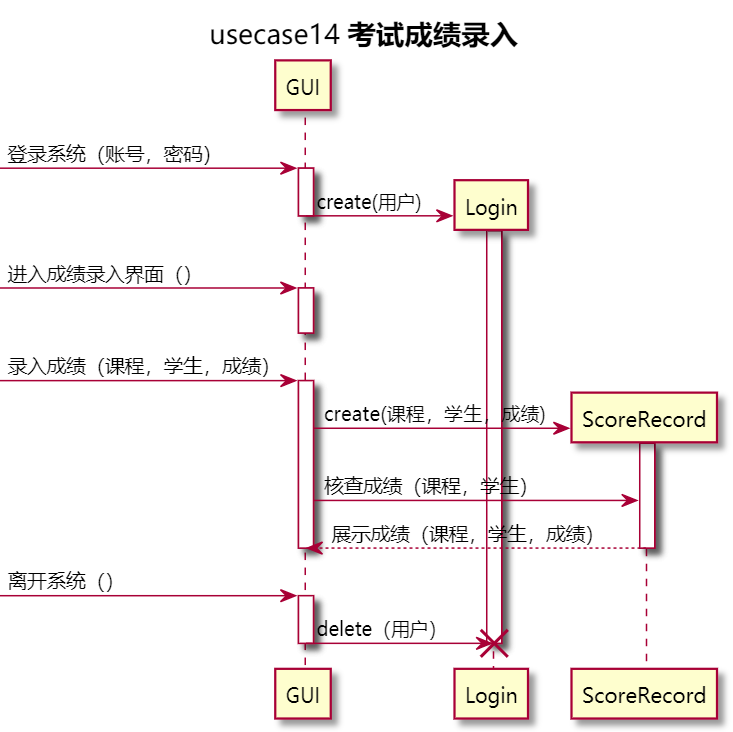
1. 教师登录系统

2. 教师选择需要录入成绩的科目

3. 教师录入成绩

4. 教师核查录入的成绩是否有误





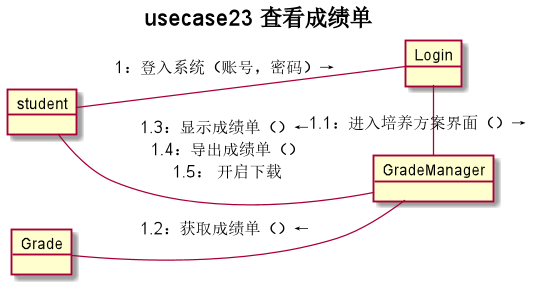
### 学生查看成绩单

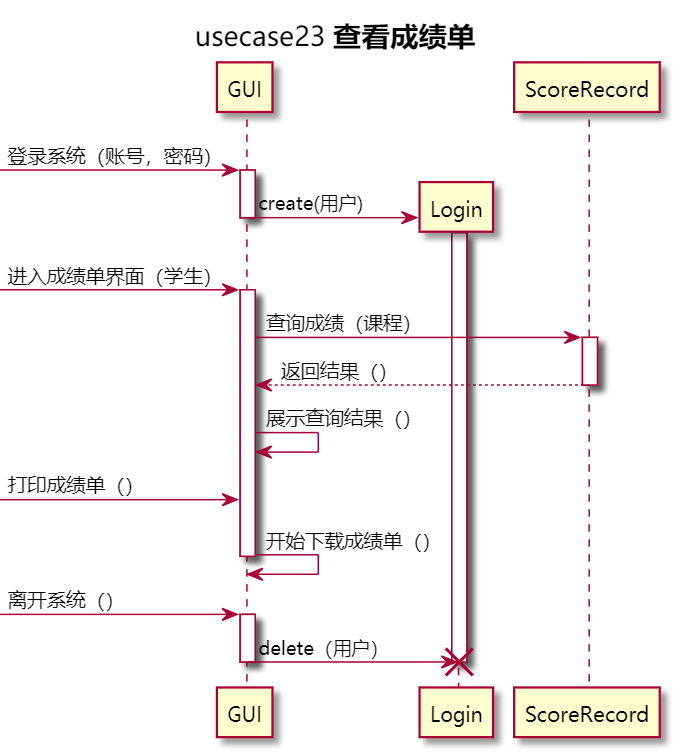
1. 学生登录教务系统

2. 学生点击“查看成绩单”选项

3. 学生点击“导出成绩单”选项

4. 系统输出.pdf格式的成绩单



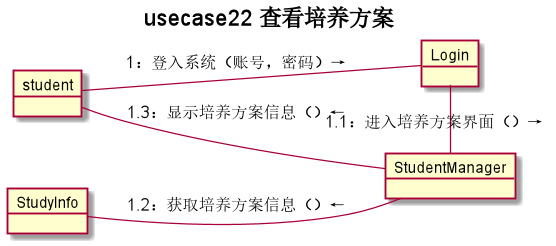


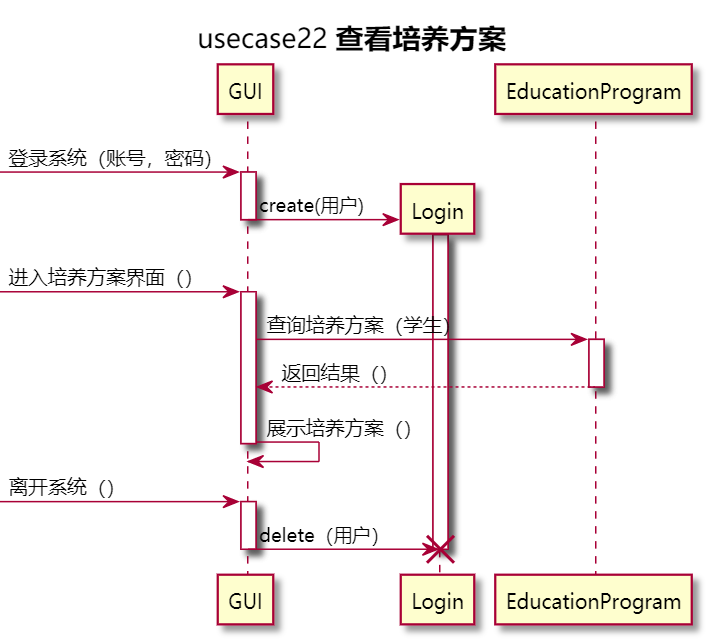
### 学生查看培养方案完成情况

1. 学生登录教务选课系统

2. 学生点击“查看培养方案”按钮

3. 系统显示学生的培养方案



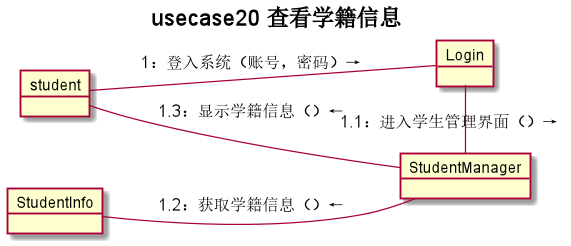


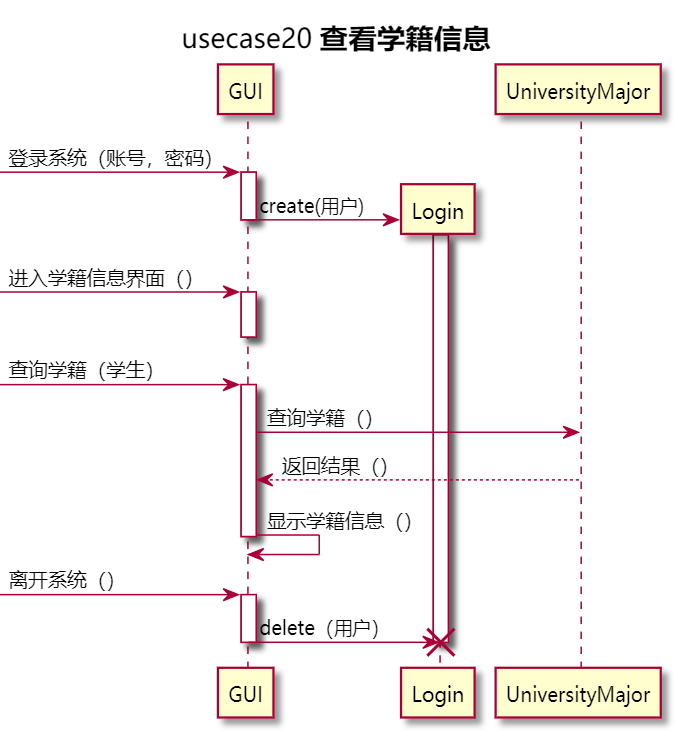
### 学生查看学籍信息

1. 学生登录教务选课系统

2. 学生点击“查看学籍信息”按钮

3. 系统显示学生的学籍信息





### 教务员更改学籍信息

1. 教务员登录系统

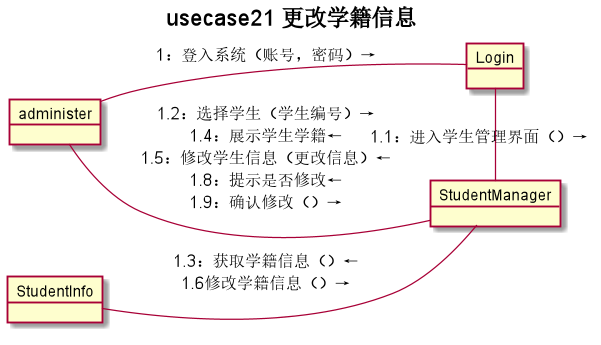
2. 教务员进入学生学籍变更界面

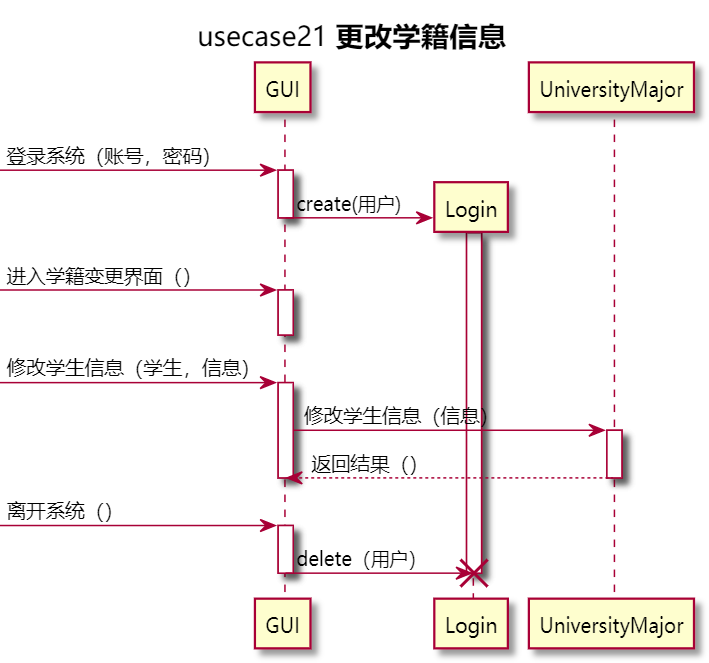
3. 教务员输入学生信息，进入指定学生的界面

4. 教务员修改学生信息

5. 系统提示是否确认修改

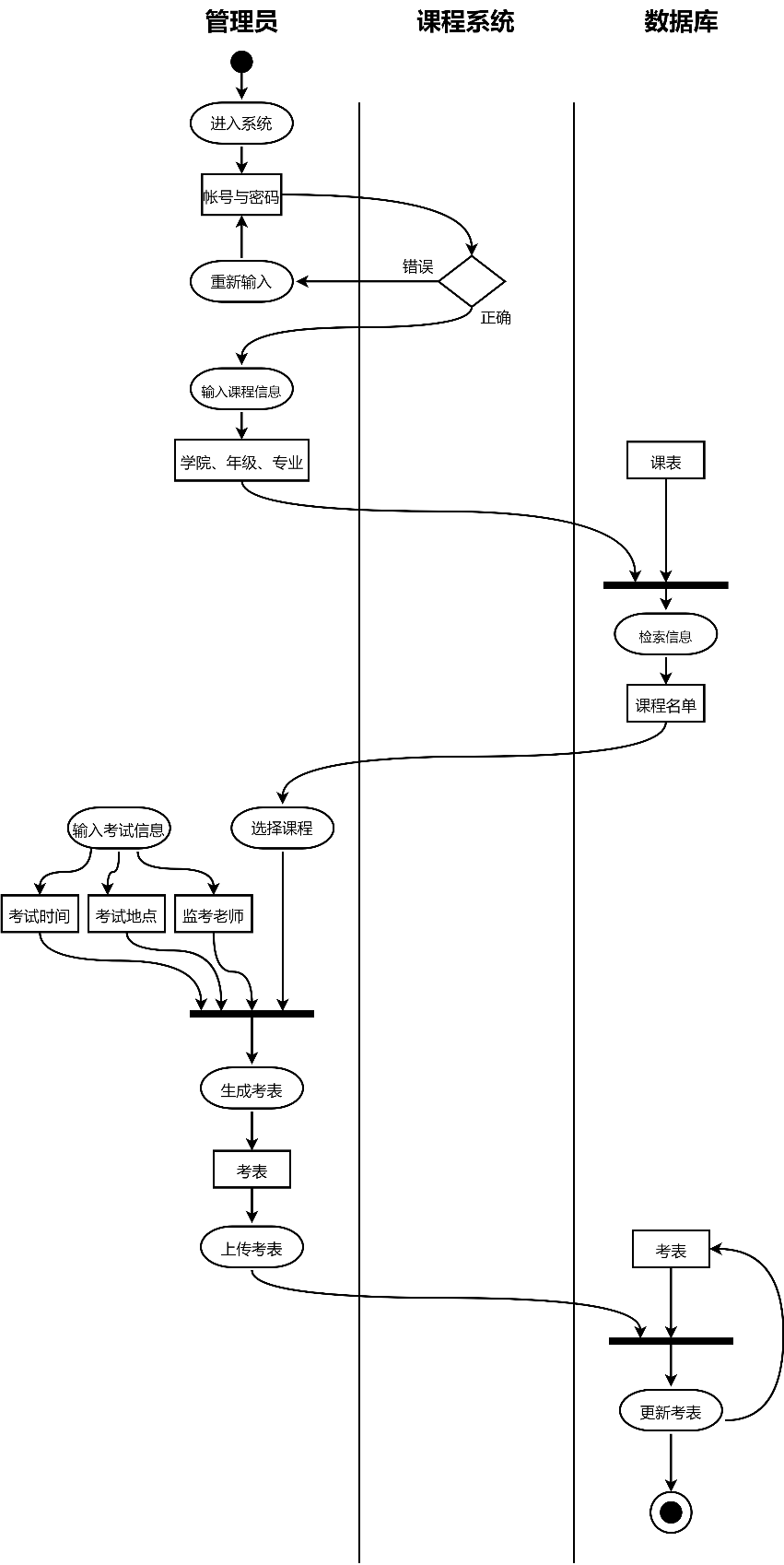
6. 教务员确认修改



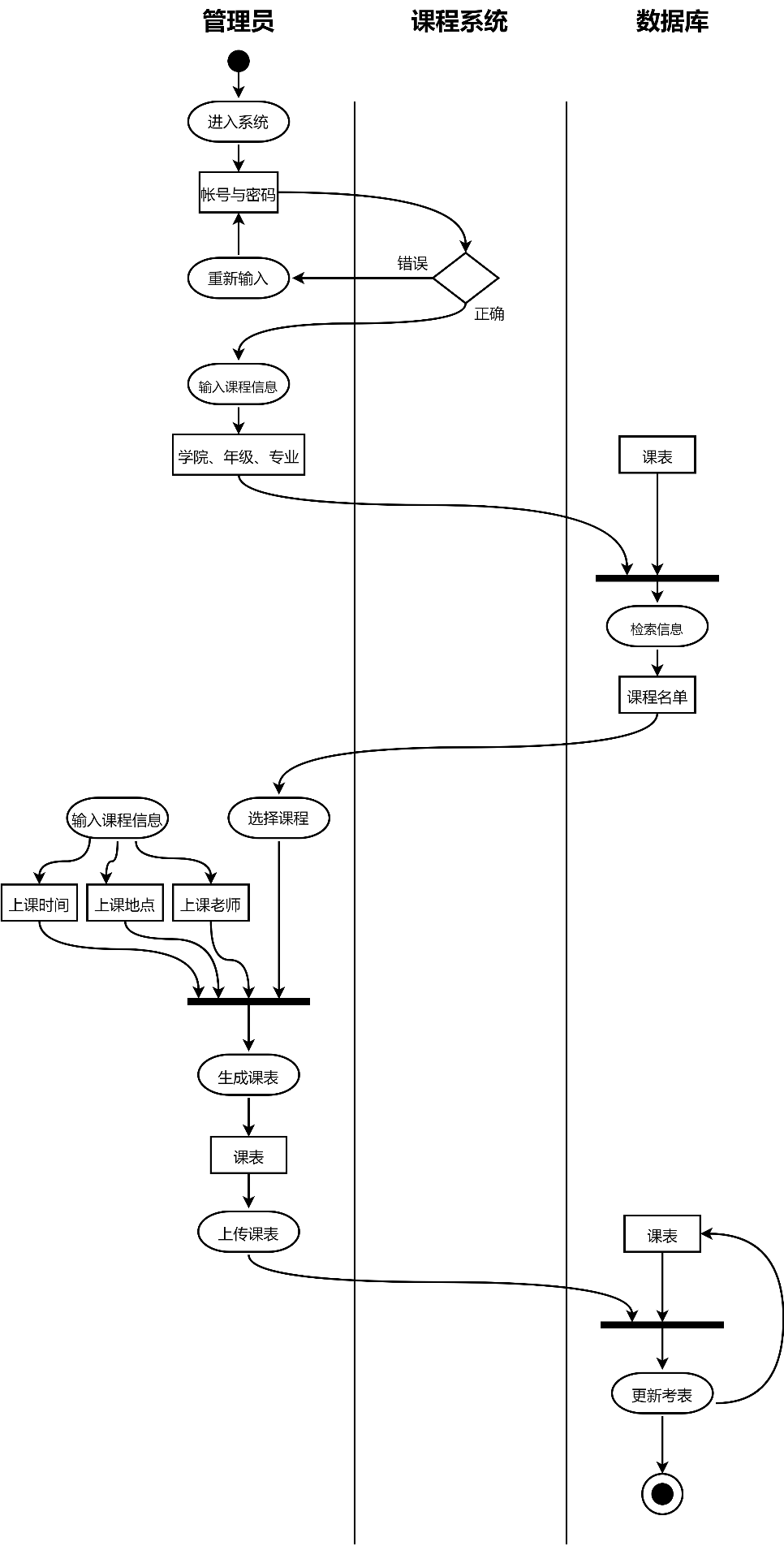


**活动图**

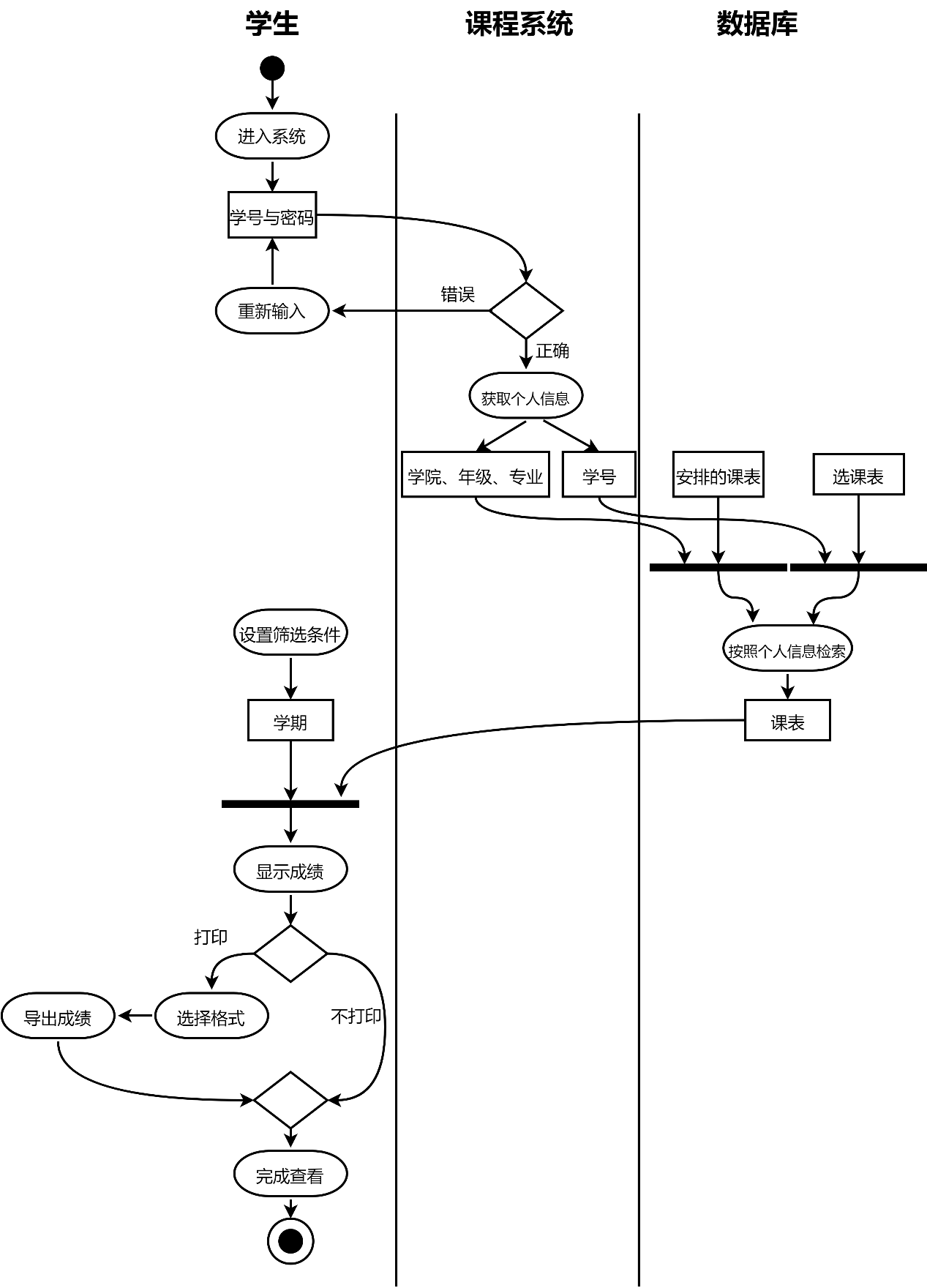
### 安排考表



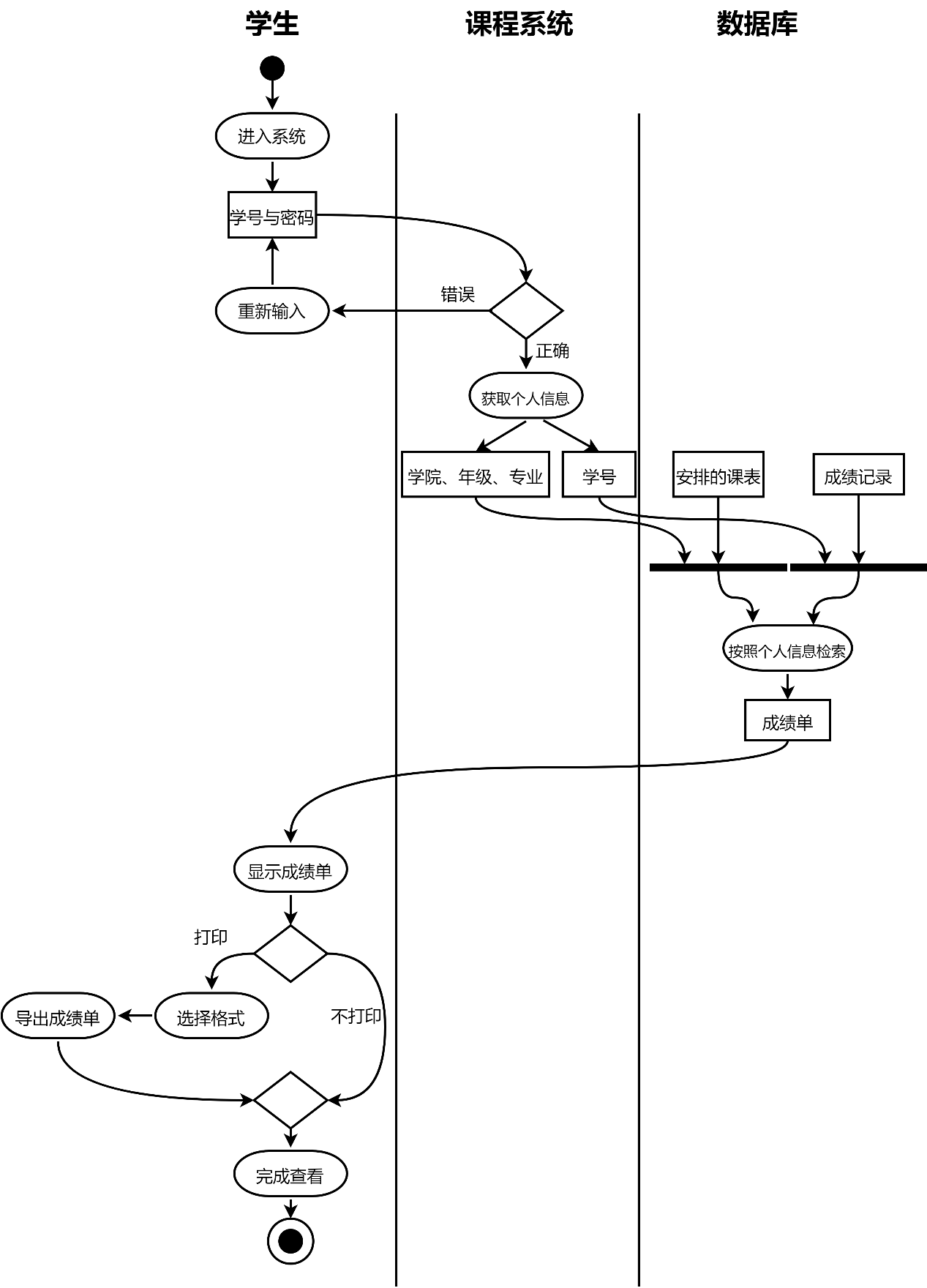
### 安排课表



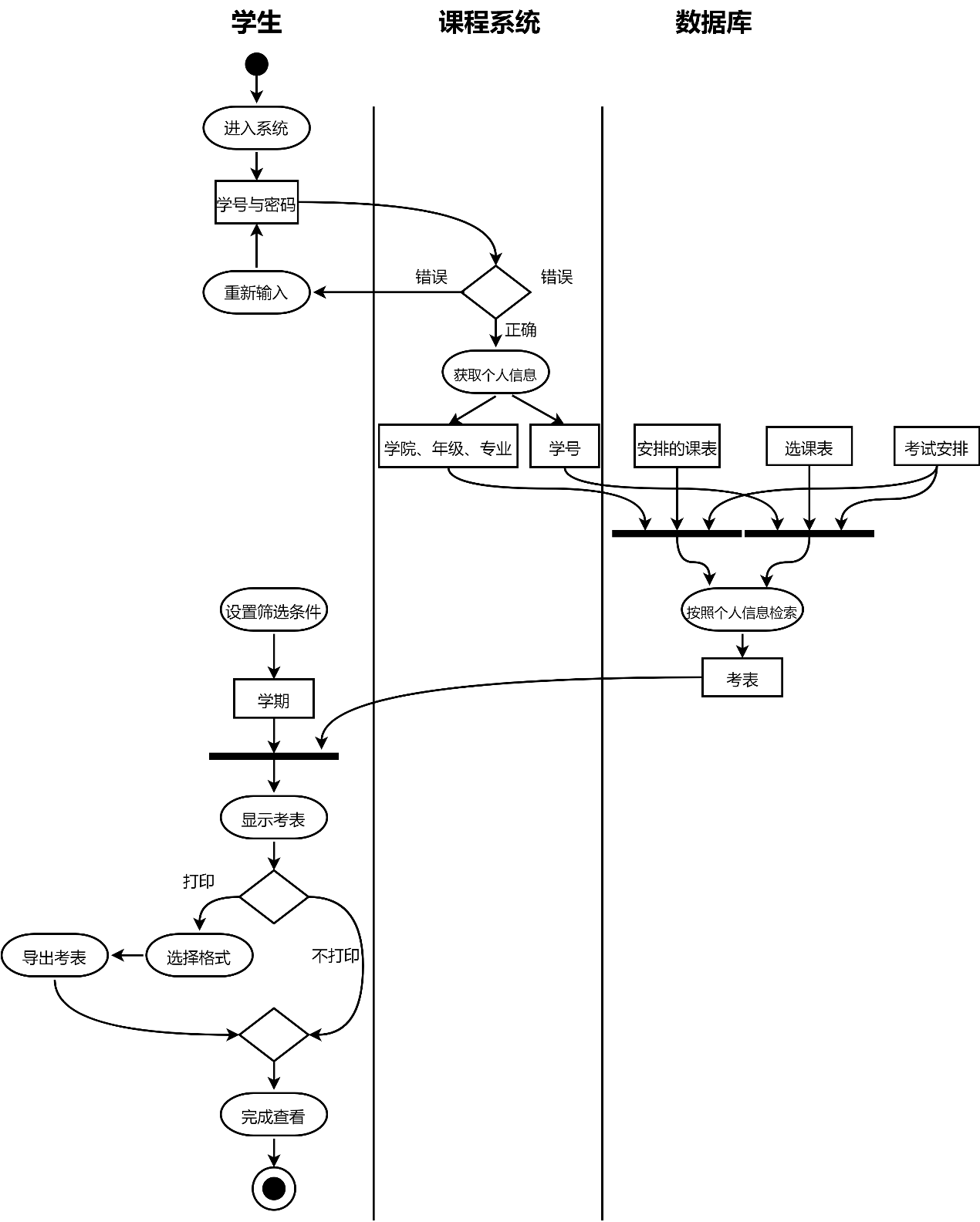
### 查看成绩



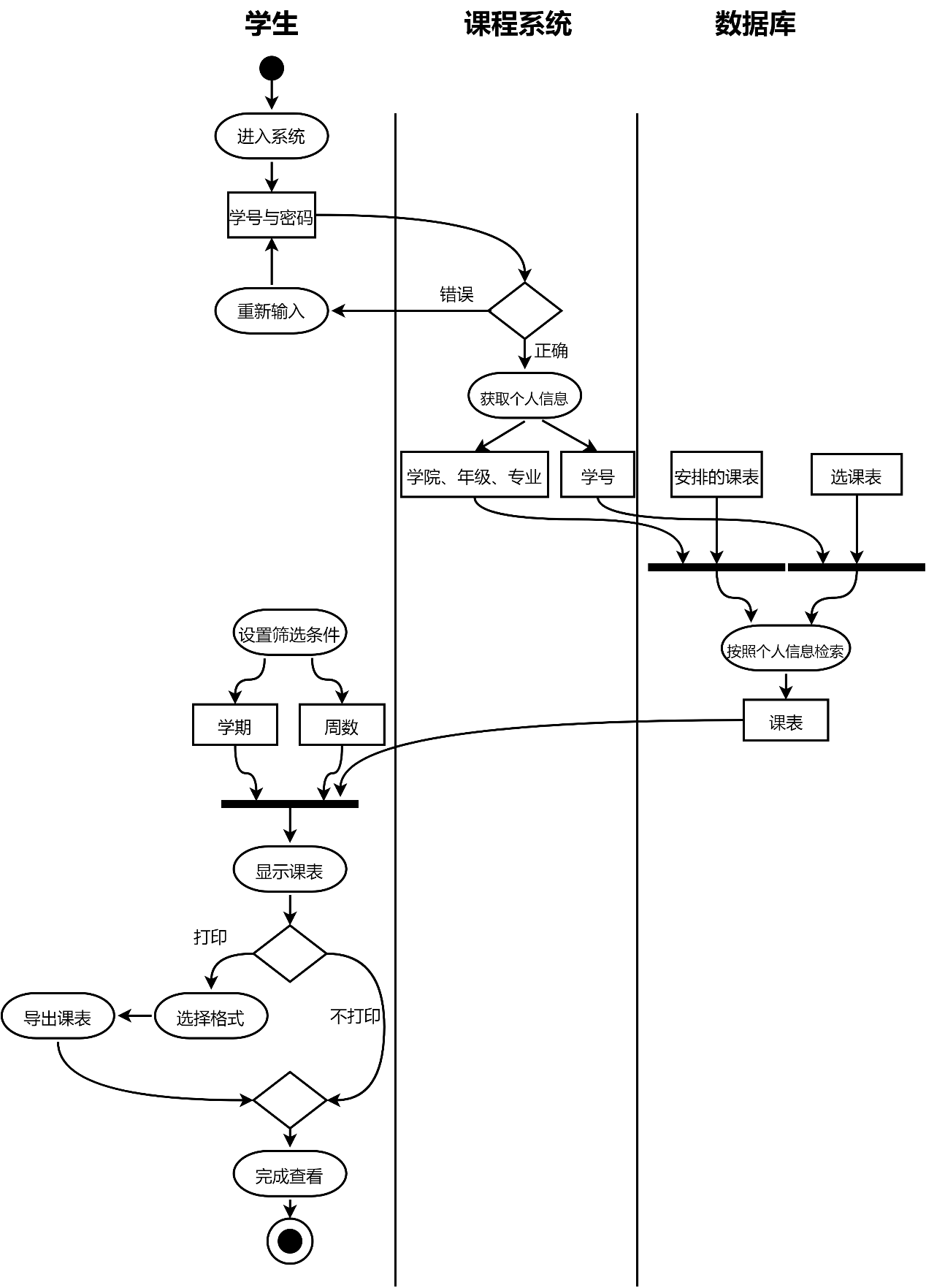
### 查看成绩单



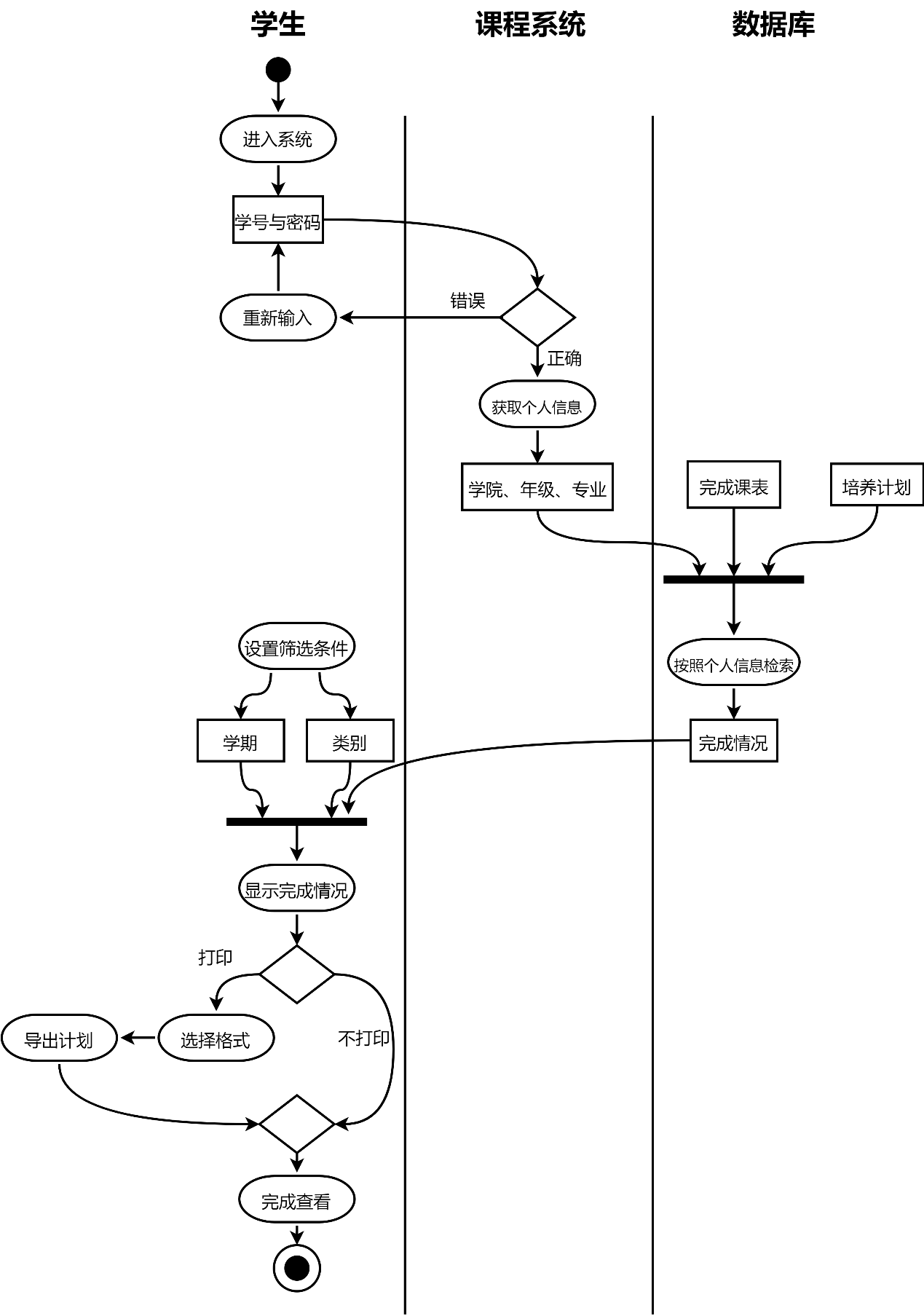
### 查看考表



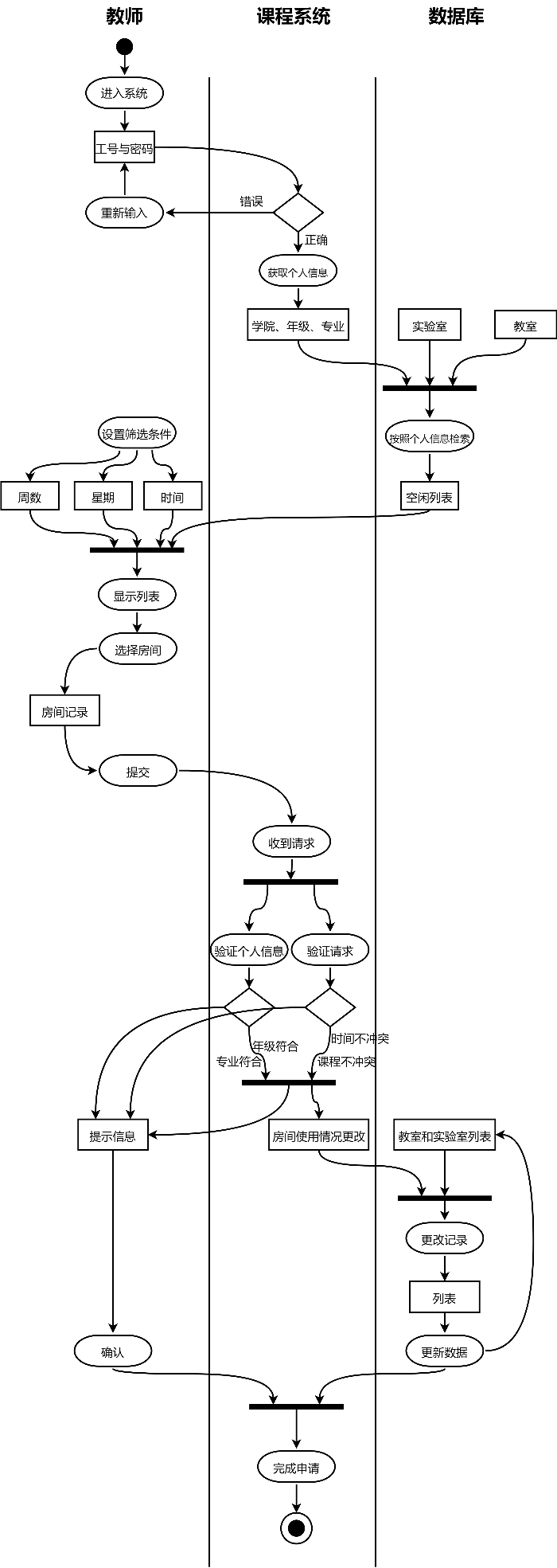
### 查看课表



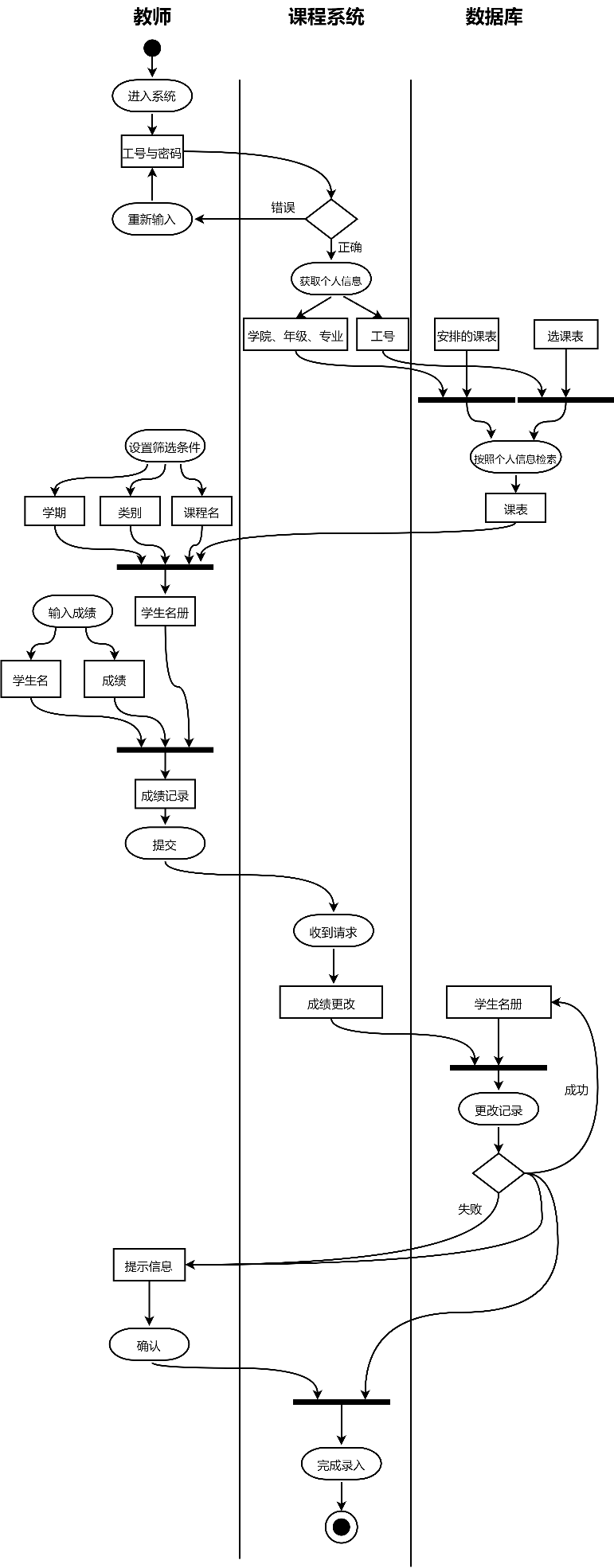
### 查看培养方案



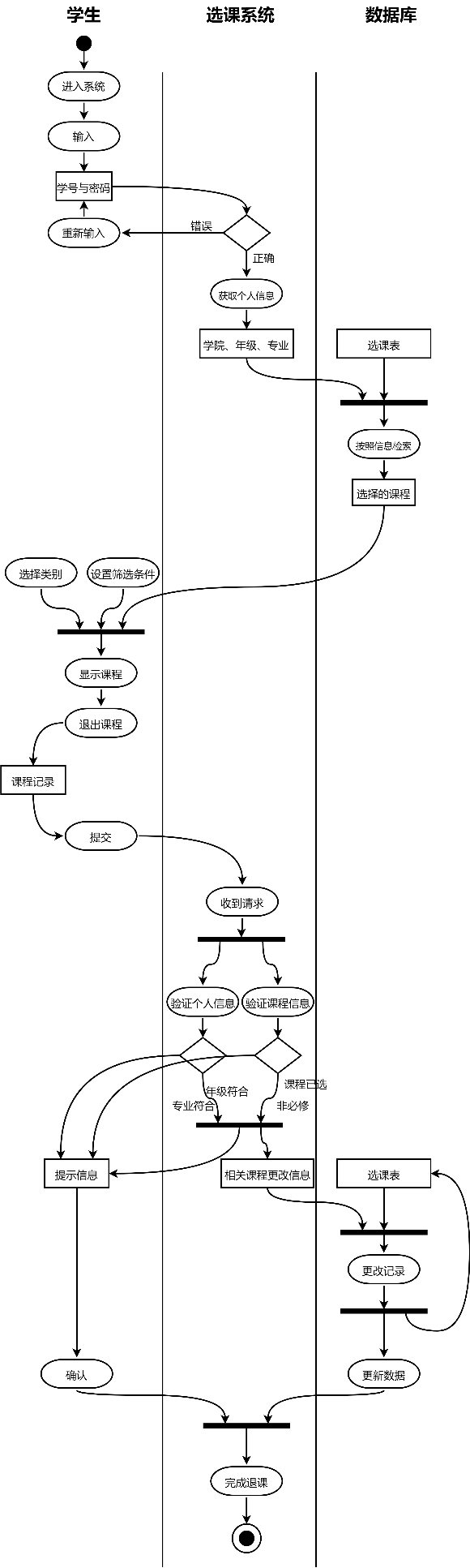
### 教室申请



### 录入成绩



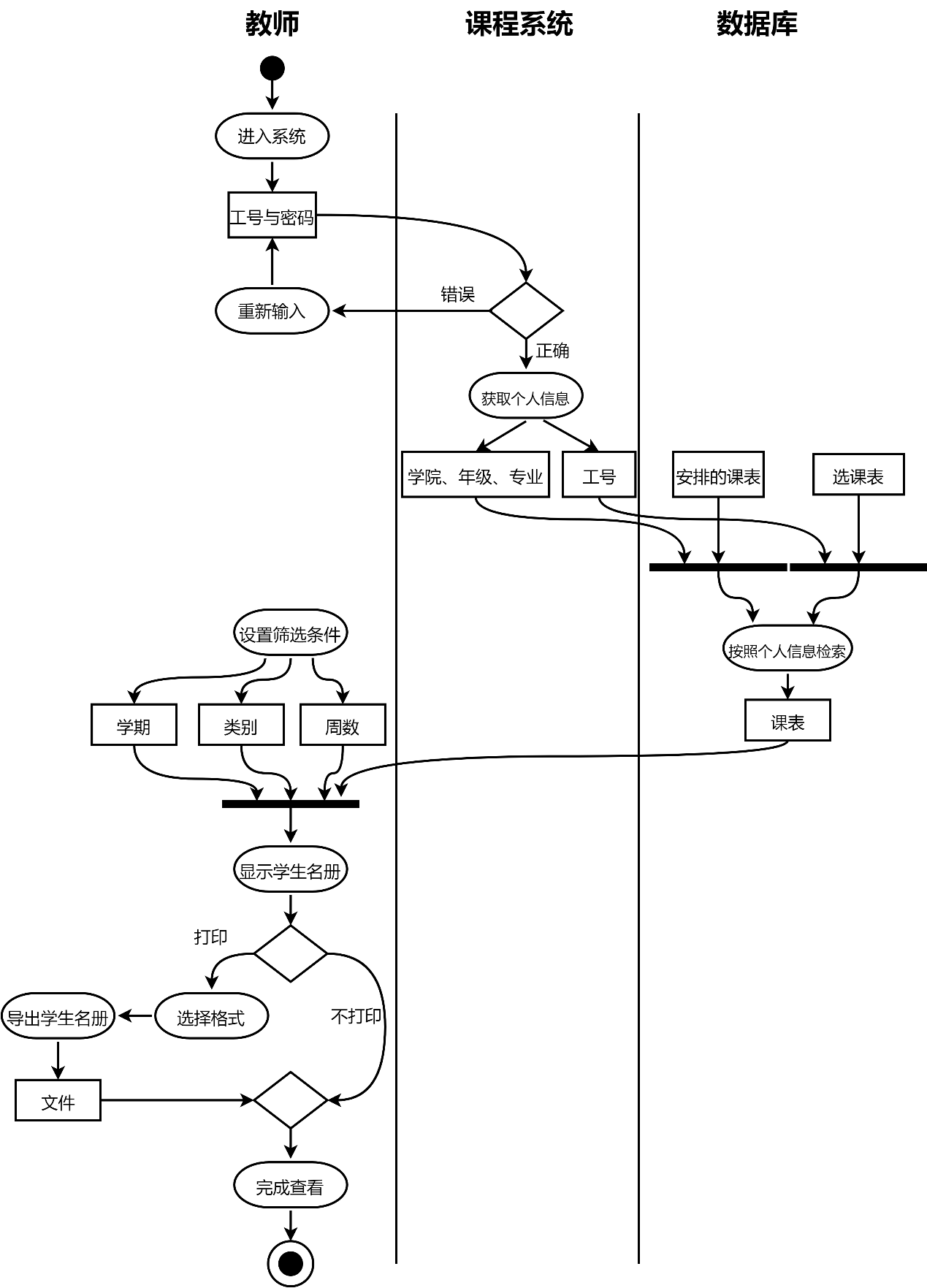
### 退课



### 选课

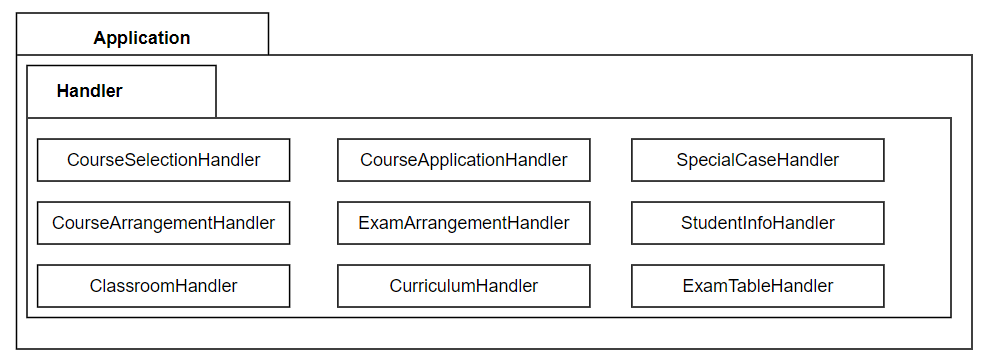


### 学生名册

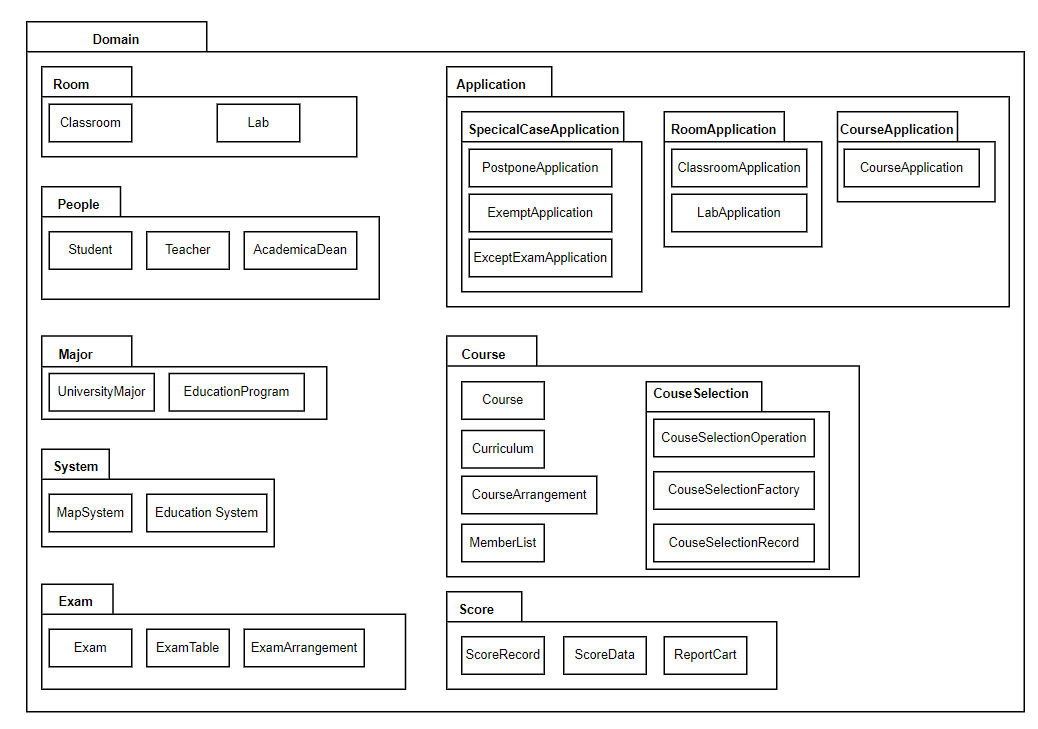


**逻辑结构设计与包图**

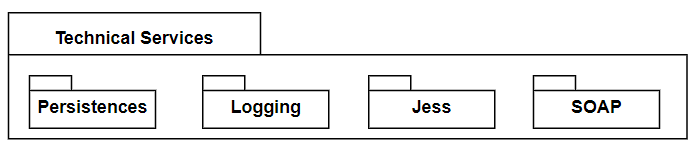
### Application



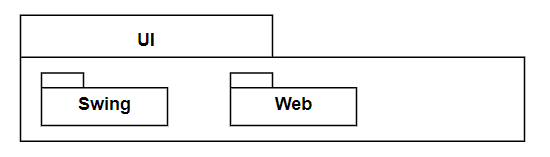
### Domain



### TechnicalService



### UI



**类图**

## 应用GRASP进行对象设计

**创建者**

我们遍历了所有由领域模型中衍生而来的类，考虑如何选择这些类的创建者，在实现低表示差异的基础上，我们将考虑的结果以表格的形式展现：

|  |  |  |  |
| --- | --- | --- | --- |
| 类名 | 从属 | 创建者 | 说明 |
| Student | 人员部分 | **Education System** | 教务系统中记录有学生的基本信息 |
| Teacher | 人员部分 | Education System | 教务系统中记录有教师的基本信息 |
| Academic Dean | 人员部分 | Education System | 教务系统中记录有教务员的基本信息 |
| Course | 课程安排部分 | Course Arrangement | 课程安排中聚集了大量的课程信息 |
| Course Arrangement | 课程安排部分 | Education System | 教务系统中记录有课程安排的基本信息 |
| Curriculum | 课程安排部分 | Course Arrangement | 课程安排中具有课表初始化的信息 |
| Member List | 课程安排部分 | **Course Selection Factory** | 选课记录中记录了课程相关的人员信息 |
| Score Record | 成绩部分 | Teacher | 老师具有初始化某一条具体的评分记录的数据信息 |
| Report Cart | 成绩部分 | **Score Data** | 总的成绩数据记录中含有初始化每个人的成绩单的相关数据 |
| Course Selection Operation | 选课部分 | Student | 学生具有初始化某一条具体的选课记录的数据信息 |
| Course Selection Record | 选课部分 | Education System | 由总的教务系统负责选课数据列表的初始化 |
| Exam Arrangement | 考试部分 | Education System | 由总的教务系统负责考试数据列表的初始化 |
| Exam Table | 考试部分 | Exam Arrangement | 考试总安排中包含了考表初始化的信息 |
| Exam | 考试部分 | Academic Dean | 由教务人员进行考试安排，教务人员含有考试初始化的数据的能力 |
| University Major | 学籍部分 | Education System | 教务系统中记录有各专业的基本信息 |
| Education Program | 学籍部分 | Education System | 教务系统中记录有学生学籍的基本信息 |
| Classroom | 其他 | **Map System** | 使用一个新类储存所有与地理位置相关的信息 |
| Lab | 其他 | Map System | 使用一个新类储存所有与地理位置相关的信息 |

通过应用创建者模式进行分析，我们发现了进行领域建模时的一些遗漏

1. 缺乏一个总体的类负责一些独一无二的信息的初始化，例如人员基本信息、整体课程安排等，因此我们需要一个能够代表系统整体的类 “Education System” ，该类主要负责部分类的初始化。
2. 教室与实验室两个类同样也缺少创建者，但是因为这两者并不包含于教务系统中，只是作为基本的信息与教务选课系统中的一些类相关联，因此我们新建了一个类 “Map System” 专门负责这些地理信息相关类的创建与生成。
3. 对象的创建常常具有相当的复杂性，例如为了性能而使用回收的实例，基于某些外部特征值有条件地创建一个或一族对象类的实例。在教务选课系统中选课记录可能需要被批量地创建选课记录，或是选课记录需要被频繁地增加、修改、删除，因此我们考虑使用一个工厂类 ”Course Selection Factory” 去完成课程信息地创建。
4. 关于成绩部分，我们之前在领域模型中建立了一个类 “Score Record” 用于记录具体的一条成信息记录，但是缺乏一个类能够揽括所有的成绩记录，并负责这些成绩记录的生成，因此我们新建了一个类 “Score Data”负责这一职责。

**信息专家**

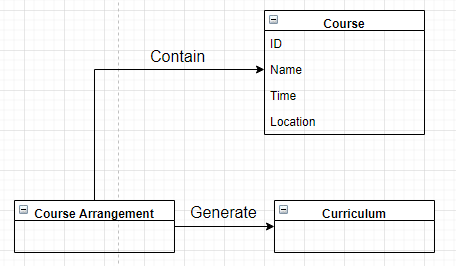
该模式主要需要解决地问题是“谁应该了解相关信息”

|  |  |  |  |
| --- | --- | --- | --- |
| 信息 | 类别 | 相关对象 | 说明 |
| 学生可选课程 | 课程部分 | Course Arrangement | 整体课程安排中含有课程所属类别，以此可以筛选出学生可选课程 |
| 学生已选课程 | 课程部分 | Curriculum | 学生已选的课程会更新入课表中，通过课表可以查询到学生已选课程 |
| 教师任教课程 | 课程部分 | Curriculum | 教师任教的课程会更新入课表中，通过课表可以查询到教师任教课程 |
| 学生课表 | 课程部分 | Curriculum | 类基本职能 |
| 教师课表 | 课程部分 | Curriculum | 类基本职能 |
| 选课学生名单 | 课程部分 | Member List | 类基本职能 |
| 课程基本信息 | 课程部分 | Course | 类基本职能 |
| 教师开课申请 | 课程部分 | **Course Application** | 区别于课程内容等描述内容，具体到某一学期的课程信息 |
| 教师考表 | 考试部分 | Exam Table | 类基本职能 |
| 学生考表 | 考试部分 | Exam Table | 类基本职能 |
| 考试基本信息 | 考试部分 | Exam | 类基本职能 |
| 考试安排 | 考试部分 | Exam Arrangement | 类基本职能 |
| 参与考试学生名单 | 考试部分 | **Exam Member List** | 因为考试还分为补考、免修考试，因此考试名单于课程名单并不对等 |
| 考试形式申请 | 考试部分 | Course Application | 考试形式申请应包含在课程相关申请中，在申请课程时一并进行 |
| 免考、缓考与免修考试的申请 | 考试部分 | **Special Case Application** | 关于免考与缓考行为的申请在之前涉及的类中并未包含 |
| 学生成绩 | 成绩部分 | Score Record | 类基本职能 |
| 学生成绩单 | 成绩部分 | Report Card | 类基本职能 |
| 学生选课记录 | 选课部分 | Course Selection Record | 类基本职能 |
| 学生学籍信息 | 学籍部分 | Student | 类基本制能 |
| 学生培养方案 | 学籍部分 | Education Program | 类基本制能 |

通过信息专家模式，我们发现我们在领域模型中设计的对象无法将全部的信息表示，因此，在了解缺失信息的基础上，我们新增了一些类，以弥补信息表示不全的缺点。

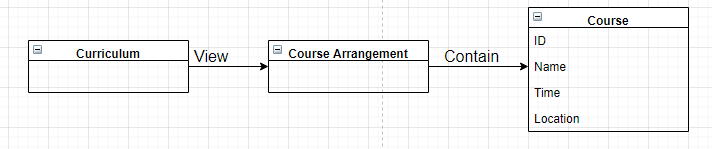
1. 在用例上，教师先应提交开课申请，开课申请中应包含了课程名字、主要内容等基本信息，同时也应包含课时安排、考核安排这样与具体学期相关的信息，这样的信息在之前所设计的类中并不能完整地表达。因此我们新设计了 “Course Application” 这各类充当这部分信息恶信息专家。
2. 在之前的领域模型上，我们为课程设计了与课学生名单这一对象，我们发现，对于考试，并不能完全将这一名单沿用，因为考试除正常期末考试外，还有补考与免修考试，这一类考试的参与者名单于选课学生名单并不完全对等，因此，我们为考试专门设计了一个参与考试人员名单类，以充当这一部分信息的信息专家。
3. 同样是在考试上，我们发现免考、缓考、免修考试等各类考试申请应保存在教务选课系统中，但之前所设计的类并不能认知到这一部分信息，因此我们专门设计了一个 “Special Case Application” 这一申请类，充当这一部分内容的信息专家。

**低耦合**



类设计中关于课程的几个关键部分疑似存在耦合性问题，但实际上这只是 “Curriculum” 与 “Course Assangement” 之间的之间的关系表达不恰当造成的。

“Curriculum”实际上相当于“Course Assangement”类的一个视图，对其中的信息进行筛选，然后选择性地呈现，因此实际上所有的类都耦合于“Course”类，故可以将几个类之间的排列关系进行调整，正确的表示方法如下所示：



**控制器**

控制器是UI层之上的第一个对象，主要负责接收和处理系统操作信息。

在之前的领域模型设计中，主要是针对用例场景中出现的实体进行类的设计，而那些用户操作的处理，几乎没有设计，因此我们需要引入一些“控制器”类，负责对用户的操作进行处理。

一般而言，控制器有两种选择，要么是表示整个“系统”、“根对象”的类，要么就是专门的系统事件的接收者或处理者。在教务选课系统中，教务系统本身承担着大量类的创建职责，因此我们选择为事件新建专门的事件处理类。

|  |  |  |
| --- | --- | --- |
| 控制器 | 类名 | 说明 |
| 学生选课操作 | **Course Selection Handler** | 包括学生进行选课、退课、补选操作 |
| 老师提交课程相关申请操作 | **Course Application Handler** | 包含对课程内容的首次提交以及之后的补交 |
| 学生提交考试、成绩相关申请操作 | **Special Case Handler** | 包含学生缓考、免考、免修、成绩复查等申请 |
| 教务员管理课程操作 | **Course Arrangement Handler** | 包含对于课程内容的增删改查 |
| 教务员管理考试操作 | **Exam Arrangement Handler** | 包含对于考试内容的增删改查 |
| 教务员管理学籍操作 | **Student Info Handler** | 包含对于学生个人信息的增删改查 |

而对于一些“显示”任务，如学生查询课表或成绩单，则应直接由GUI对象负责，因此我们在类的设计中忽略关于这一类任务的描述。

通过这些新增的类，我们可以更加完备地覆盖用例中的场景，使得用例的设计与现实情况更加符合，用户需求得到满足。

**多态**

多态是一种基本设计原则，用来设计系统如何组织以处理类似真实世界中的变化。基于多态分配职责的设计能够被简便地拓展以处理新的变化。

在之前的类设计中，有一些类实际上具有较广的职责范围，我们考虑可以通过继承与多态的思想将这些类进一步具体化，以实现软件设计的实际要求



|  |
| --- |
| 基类：Special Case Application  说明：该类主要负责储存学生特殊情况的申请，包括缓考、免修、进行免修考试等等  子类：  Postpone Application：主要负责储存学生缓考申请信息  Exempt Application：主要负责储存学生免修申请信息  Exempt Exam Application：主要负责学生免修考试申请相关信息 |

而对于一些其他也可以继续细分的类，例如课表我们可以分成学生课表与老师课表，考试我们可以分成一般考试、补考、免修考试，因为这样细分形成的子类之间并没有太大的差别，因此我们不再额外对这些类进行子类的设计。

**类图说明**

**教学**

**教室 Classroom**

location: 位置

NumberofPeople: 人数

* ClassroomHandler
  + 排序 sortResult
  + 提交 submitApplication
* MapSystem
  + 学校名称 schoolName
  + 教室列表 classroomList
* ClassroomApplication
  + NumberofStudent：学生数
  + time: 使用时间
  + reason: 使用目的
  + ClassroomName: 教室名字

**实验室 Lab**

location: 位置

NumberofPeople: 人数

usage: 用途

* MapSystem
  + 学校名称 schoolName
  + 实验室列表 labList
* LabApplication
  + NumberofStudent: 学生数
  + time: 使用时间
  + usage: 使用用途
  + reason: 使用原因
  + labName: 实验室名字

**用户**

**学生 Student**

ID: 学号

password: 密码

name: 姓名

gender: 性别

apartment: 学院

major: 专业

* 大学专业 UniversityMajor

majorName: 专业名称

educationProgram: 培养计划

college: 大学

* + 培养计划 EducationProgram
    - 内容 content
    - 所属专业 belong

**老师 Teacher**

name: 名字

gender: 性别

College: 大学

officeLocation: 办公室

email: 邮箱

**课程 Course**

ID: 课程号

name: 课程名

location: 上课地点

time: 上课时间

* 课程选择 CourseSelectionOperation

studentName: 学生名

CourseName: 课程名

operation: 操作

* + 操作记录 CourseSelectionRecord
  + 操作 CourseSelectionFactory
    - CourseSelectionFactory
    - createMemberList
    - modifySelectionRecord
  + 处理 CourseSelectionHandler
* 成员 MemberList

courseName: 课程名

teacherName: 老师名字

List: Student[] 学生

* 课程属性 CourseApplication

CourseName：课程名

CourseDuration：课程持续时间

CoursePlan：课程计划

CourseAssessmentForm：课程评估表

* + 提交 submitApplication
  + 回顾 reviewApplication
* 课程安排 CourseArrangement

Arrangemeny: Course[]

* + addCondition：添加条件
  + createArrangement：创建排列
  + deleteCourse：删除课程
  + addCourse: 添加课程
* 学期 Curriculum
  + selectFormat：选择模板
  + outputCurriculum：输出课程
  + printCurriculum：打印课程
  + refreshCurriculum：刷新课程

**考试 Exam**

courseName：课程名

memberList：课程名单

time：考试时间

location：考试地点

* 分数 ScoreRecord

studentName：学生名字

teacherName：老师名字

Score：分数

* + 分数数据 ScoreData

data: ScoreRecord[]

* + 报告 ReportCart

Cart: ScoreRecord[]

* 考表 ExamTable

table: Exam[]

* + 考表处理 ExamTableHandler

selectFormat：选择模板

outputExamTable：输出考表

printExamTable：打印考表

refreshExamTable：刷新考表

* 考试安排 ExamArrangement

arrangement: Exam[]

* + 考试安排处理 ExamArrangementHandler

addCondition：添加条件

modifyExam：修改考试

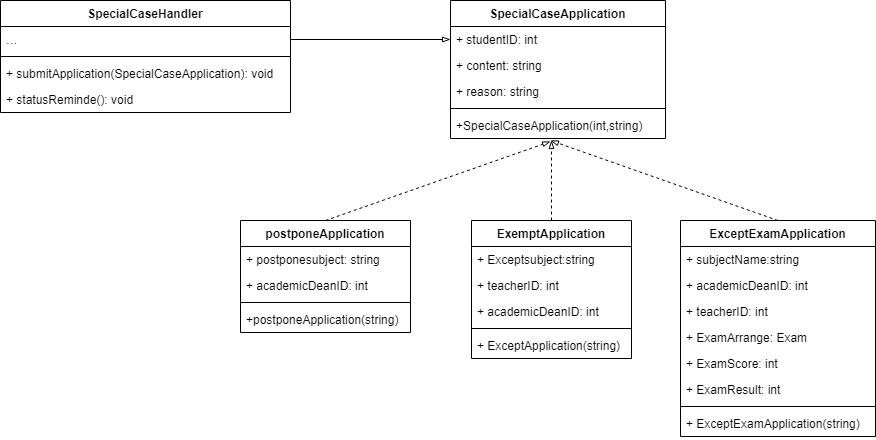
deleteExam：删除考试

addExam: 添加考试

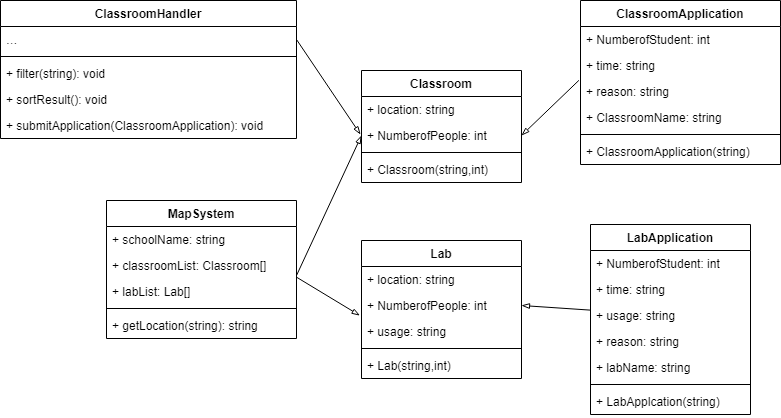
createArrangement：创建安排

## 类图设计结果

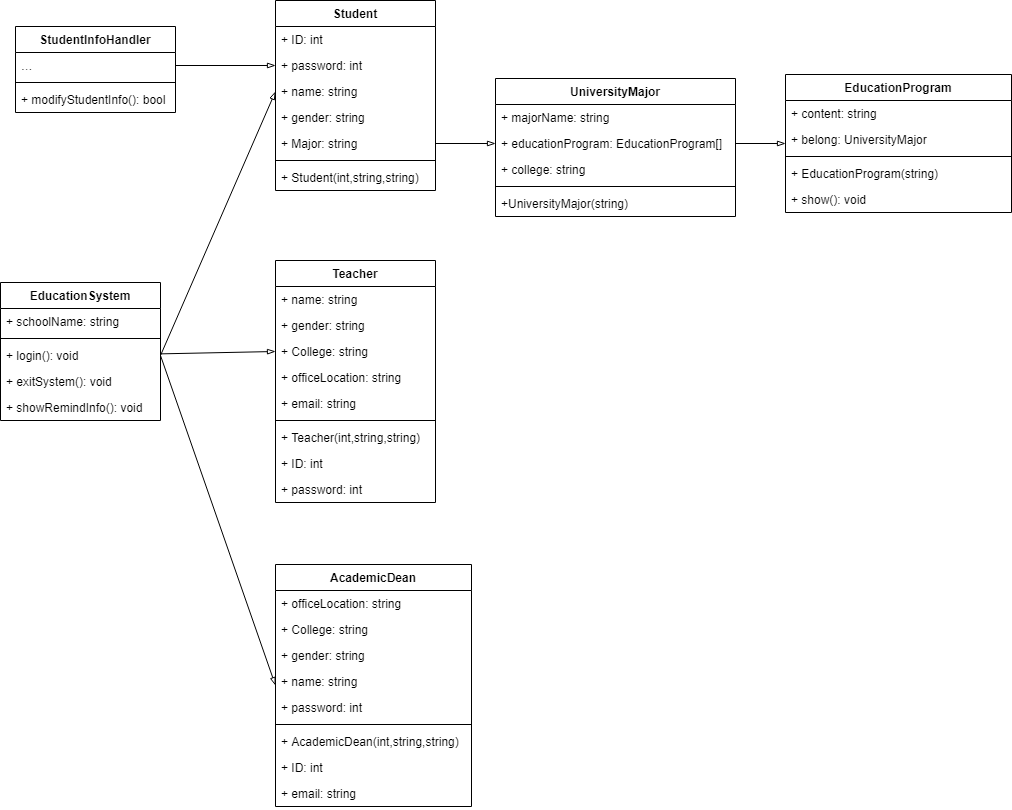
**申请相关类**



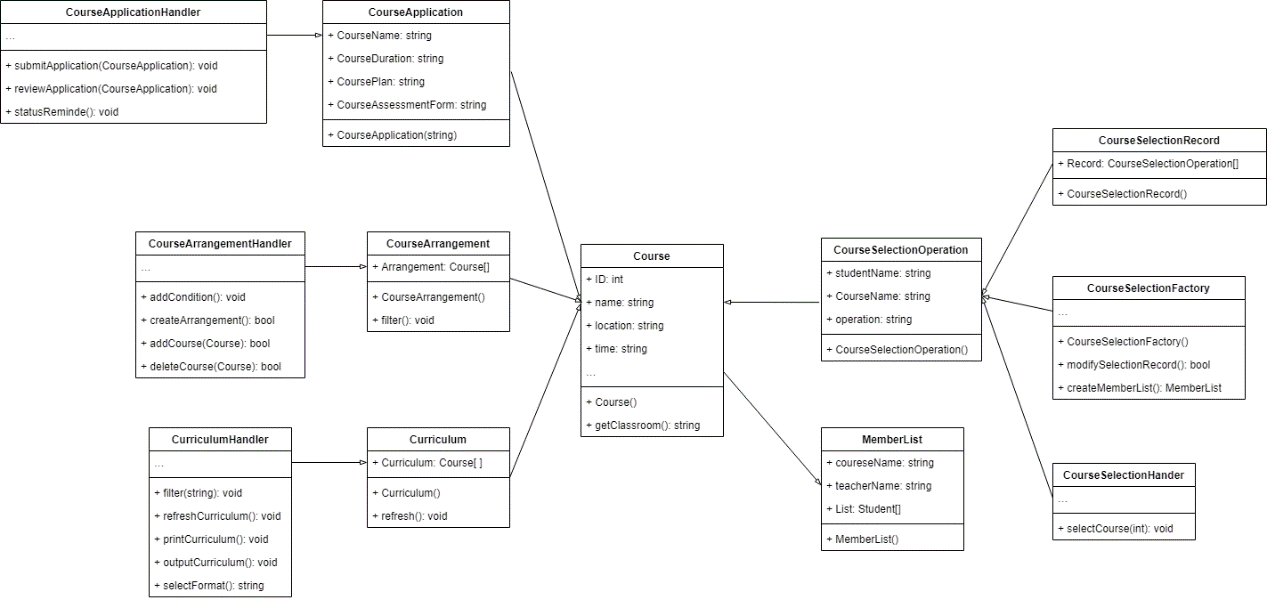
**教室及实验室相关类**

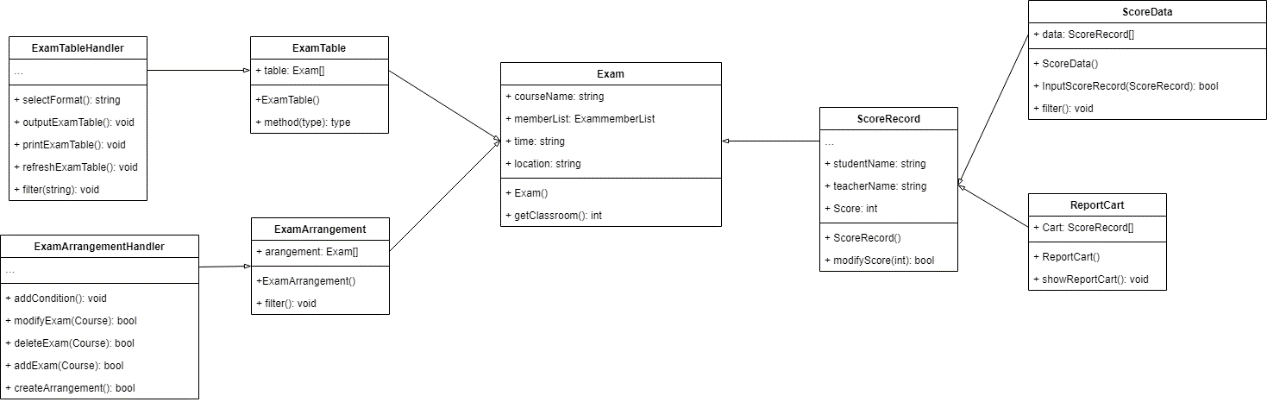


**人员及基本信息相关类**



**课程及选课相关类**



**考试及成绩相关类**