

软件系统分析与设计报告

第五次



姓名

班级

学号

电话

Email

日期

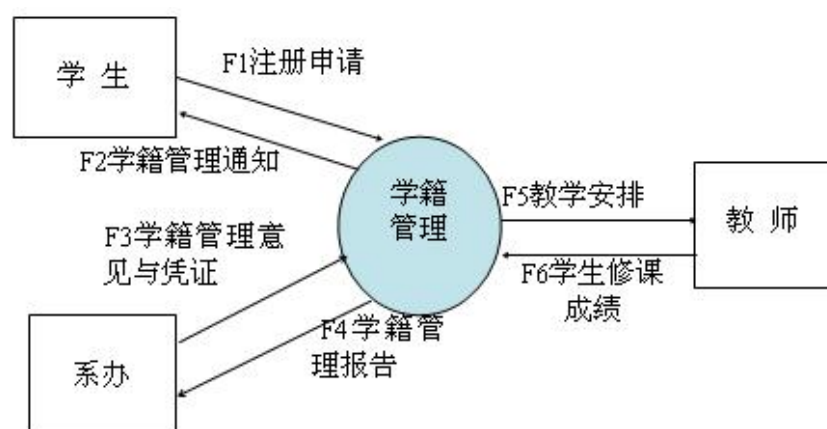
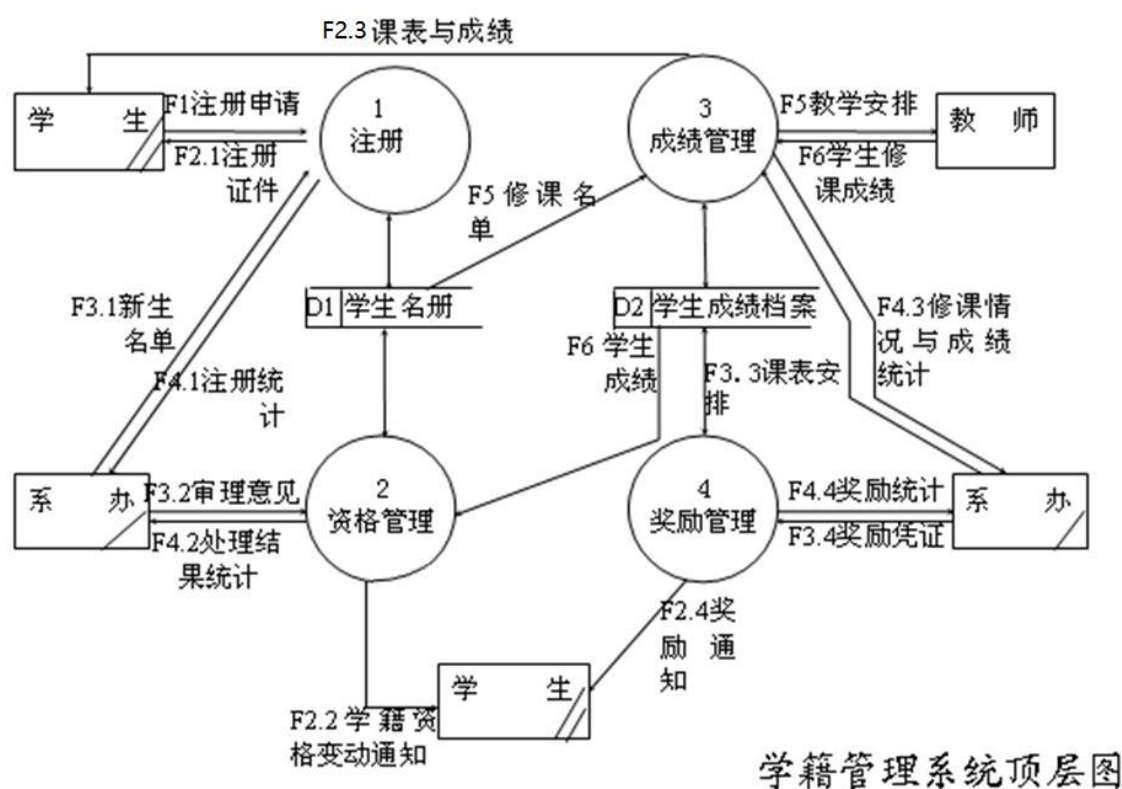
目录

一. 题目	3
二. 功能模型	4
三. 动态模型	8
四. 静态模型	17
五. 数据库 ER 模型	18
六. 总结体会	19

一、题目

系统需求分析与设计：你作为一名软件系统分析员，在某一个高校的学生管理系统中负责系统的分析与设计工作，为了更快地将客户的需求进行建模，你采用了 DFD 的方法建立了两层数据流。

模型如下：



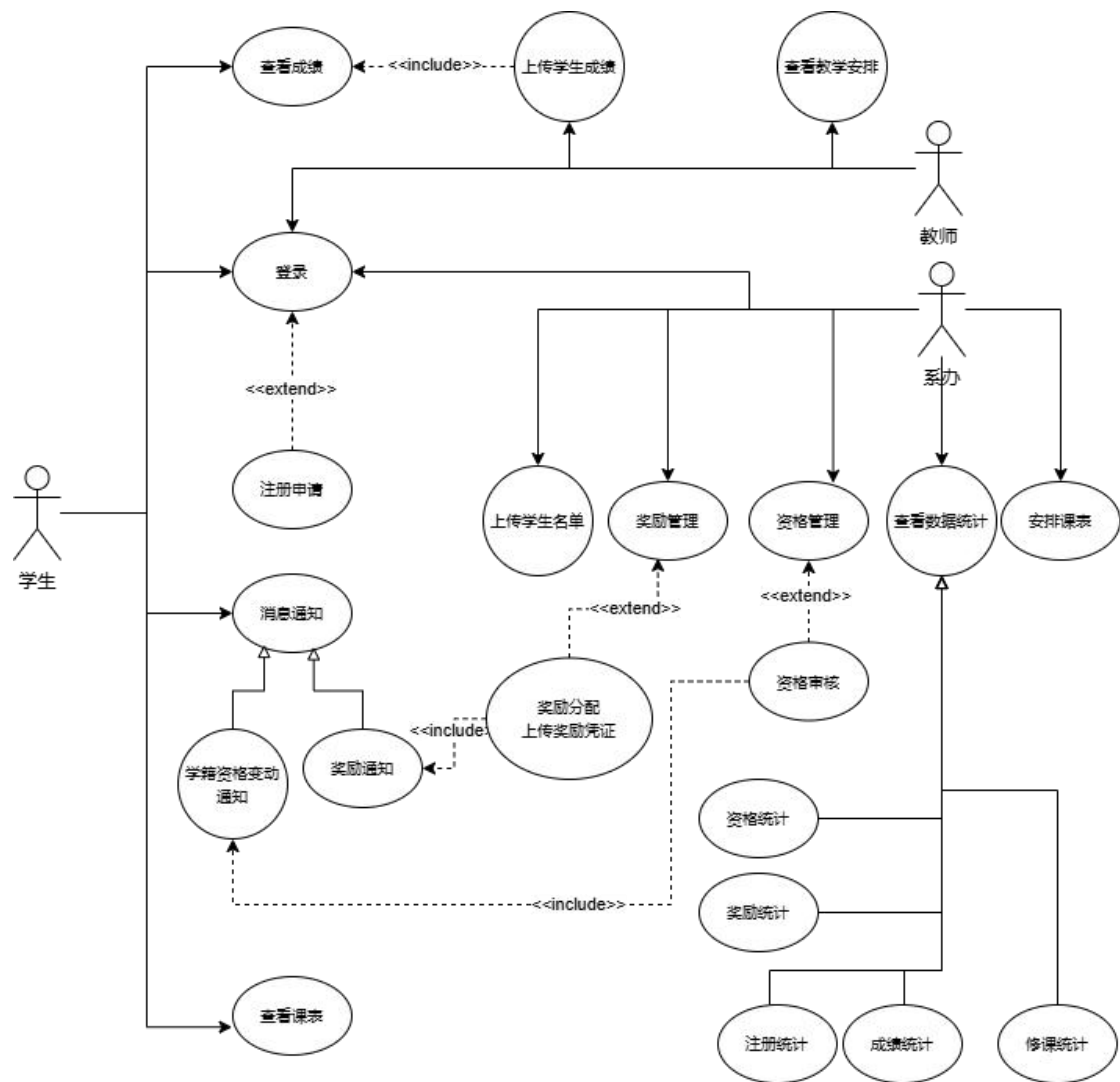
但是在与客户经理以及开发人员进行沟通交流时，大家认为这种描述方法已过时，希望能够采用面向对象的方法来进行业务需求的建模与分析，迫于用户和开发人员的要求，你准备对现有的建模方法进行调整。

请利用 UML 的建模方法将该模型转换成等效的功能模型（USE CASE 图，并简要描述事件流）、动态模型（活动图与分析时序图）、以及静态模型（分析类图），数据库 ER 模型，注意说明并解释模型之间存在的关系，且可以根据需要进行扩展，尽量完整和细化。

二、功能模型

2.1 USE CASE 图

根据题目所给的学籍管理系统的顶层图和关联图可知，该系统中的外部实体有学生、系办和教师，数据处理有注册、资格管理、成绩管理和奖励管理，数据流有 F1 注册申请、F2 学籍管理通知、F3 学籍管理意见与凭证、F4 学籍管理报告、F5 教学安排和 F6 学生修课成绩等。由此绘制用例图：



2.2 事件流的简要描述

对系统中几个重要的用例分析如下:

注册申请:

- 1.简要说明：新生首次进入系统后，需进行账号注册，以便后续登录并使用系统功能。
- 2.参与者：学生。
- 3.前置条件：学生进入系统登录页面，并选择注册选项。
- 4.后置条件：学生账号成功注册，系统中更新学生账户信息。
- 5.基本事件流：

- ①学生在登录页面点击“注册”按钮。
- ②系统加载并显示注册页面。
- ③学生在注册页面填写个人基本信息（如姓名、学号、联系方式等）并提交。
- ④系统验证提交的个人信息是否存在于学生名册中，以确认该学生的身份。
- ⑤学生为其账号设置密码。
- ⑥系统对设置的密码进行格式校验（如长度、字符组合等）。
- ⑦系统确认注册成功，将学生信息更新至数据库，并返回注册成功提示。

查看数据统计：

- 1.简要说明：系办可以通过系统查看相关的学生或课程的统计信息，以便进行分析和决策。
- 2.参与者：系办。
- 3.前置条件：系办人员进入系统并登录后，选择“查看数据统计”功能。
- 4.后置条件：完成统计数据的展示，系办可以浏览或导出相关统计信息。
- 5.基本事件流：
 - ①系办点击“查看数据统计”。
 - ②系统加载并显示统计信息页面。
 - ③系办在页面中选择需要查看的统计信息类别（如成绩统计、奖惩统

计、注册统计等）。

④系统根据所选类别，从数据库中检索相应的数据。

⑤系统将统计结果以表格或图表形式展示给系办。

⑥系办可以选择导出或打印统计数据，便于后续分析。

资格审核：

1.简要说明：系办负责对学生的奖惩资格或学籍资格进行审核，以确保数据的准确性和奖惩过程的公正性。

2.参与者：系办（主要）、学生（次要）。

3.前置条件：系办登录系统，选择“资格管理”功能，进入审核页面。

4.后置条件：资格审核完成，系统更新学生的资格状态并发送通知给相关学生。

5.基本事件流：

①系办点击“资格审核”功能。

②系统加载并显示待审核的资格信息列表。

③系办在列表中选择特定的资格信息条目进行审核。

④系统加载并展示该条信息的详细内容（如学生基本信息、申请资格类型等）。

⑤系办根据详细信息填写审核意见，决定是否通过该资格审核。

⑥系统根据审核意见，更新学生的资格状态。

⑦系统向学生发送资格审核结果的通知，提示审核结果。

上传成绩：

1.简要说明：教师在学期末或考试结束后，通过系统上传学生的成绩，

以便后续查看和统计。

2.参与者：教师。

3.前置条件：教师登录系统后，选择“上传成绩”功能。

4.后置条件：学生成绩信息上传成功，系统更新数据库中的成绩记录。

5.基本事件流：

①教师点击“上传成绩”。

②系统加载并显示成绩上传页面。

③系统展示教师所教授的课程列表，供教师选择课程。

④教师从列表中选择要上传成绩的课程。

⑤系统加载并显示该课程的学生名单。

⑥教师从学生名单中选择具体的学生，开始录入成绩。

⑦系统查询并展示该学生当前的成绩信息（如已存在的成绩，便于修改）。

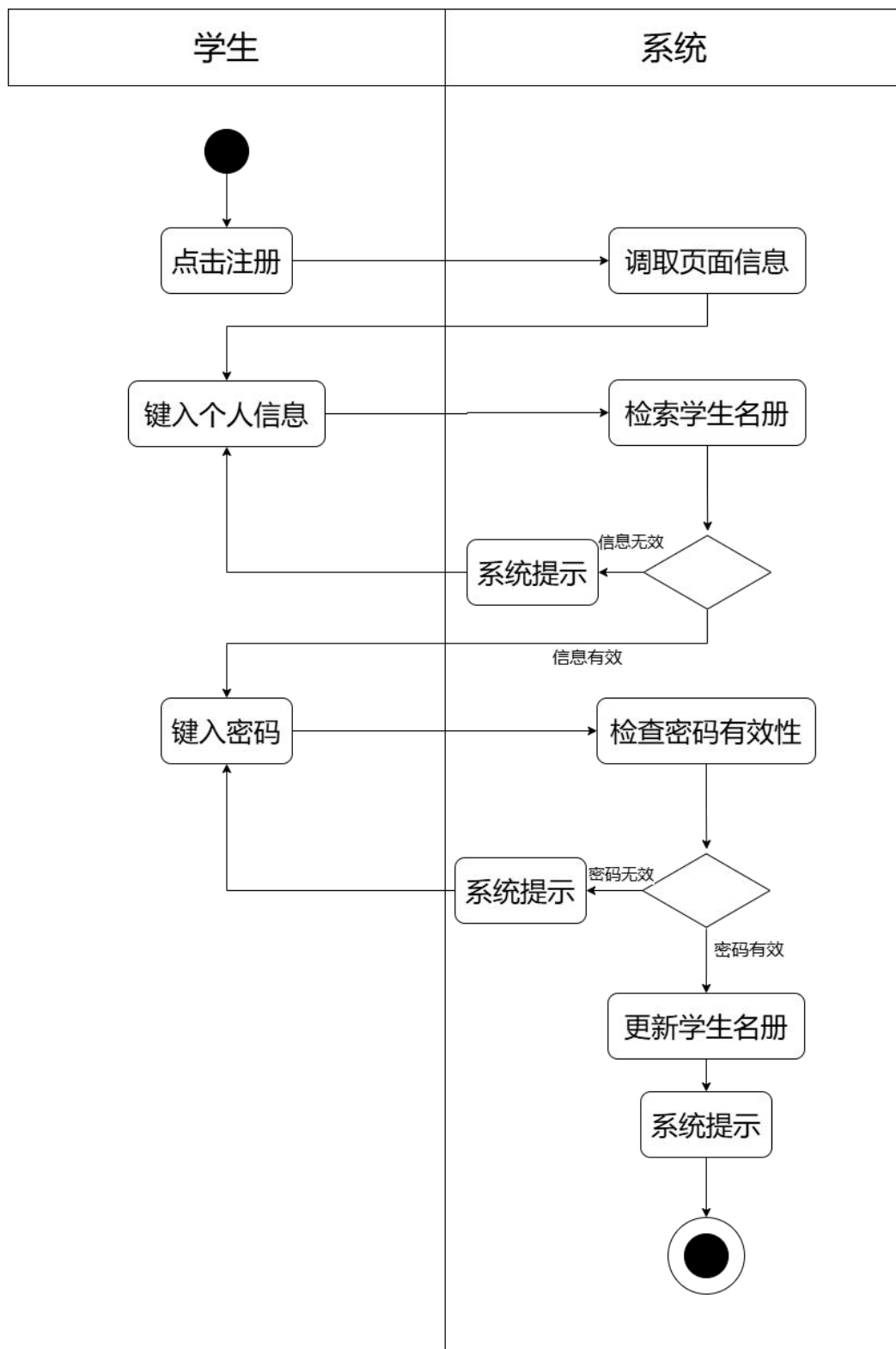
⑧教师输入新成绩或修改现有成绩，并提交。

⑨系统更新学生成绩信息，存入数据库，并向学生发送成绩更新通知。

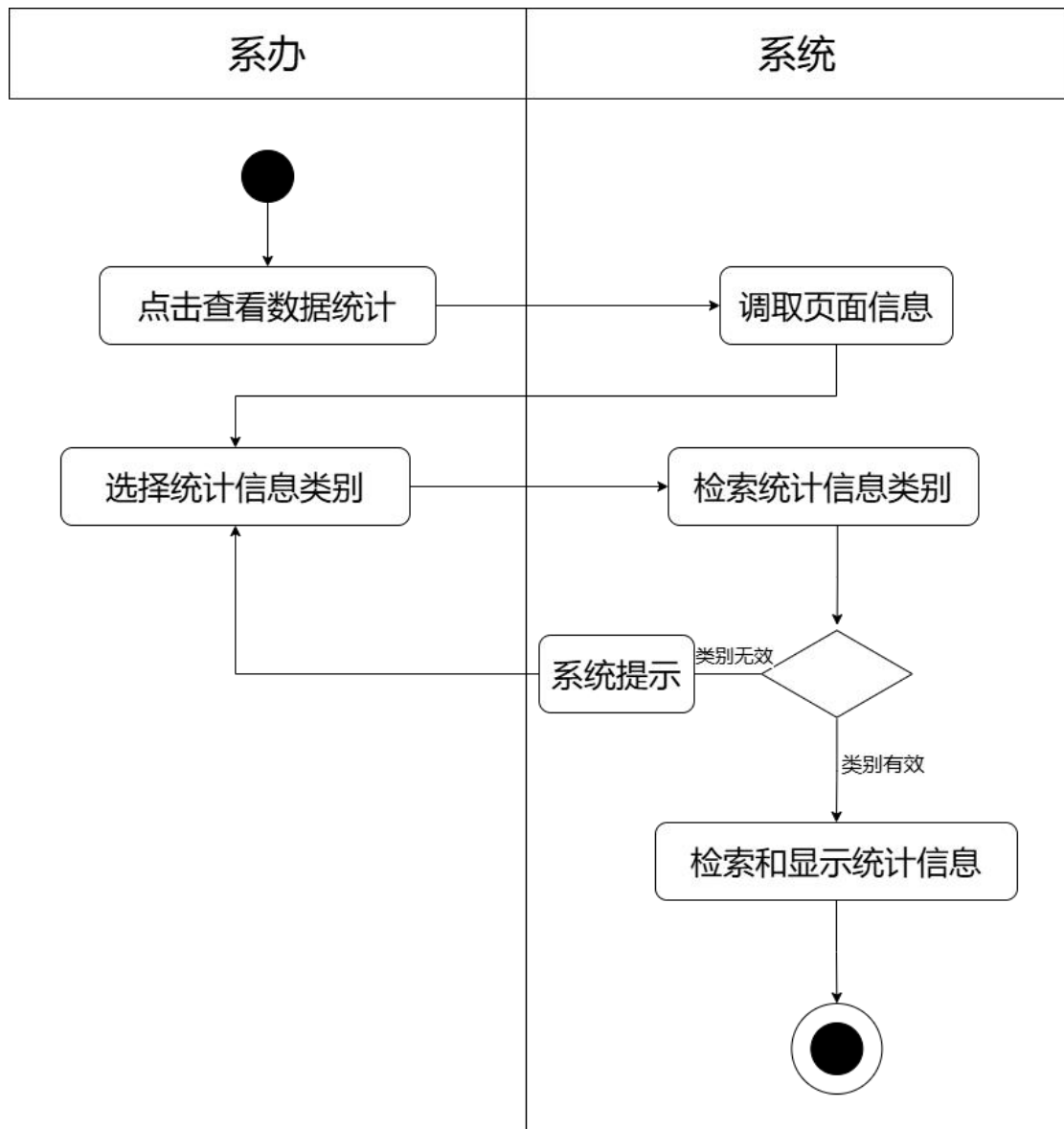
三、动态模型

3.1 活动图

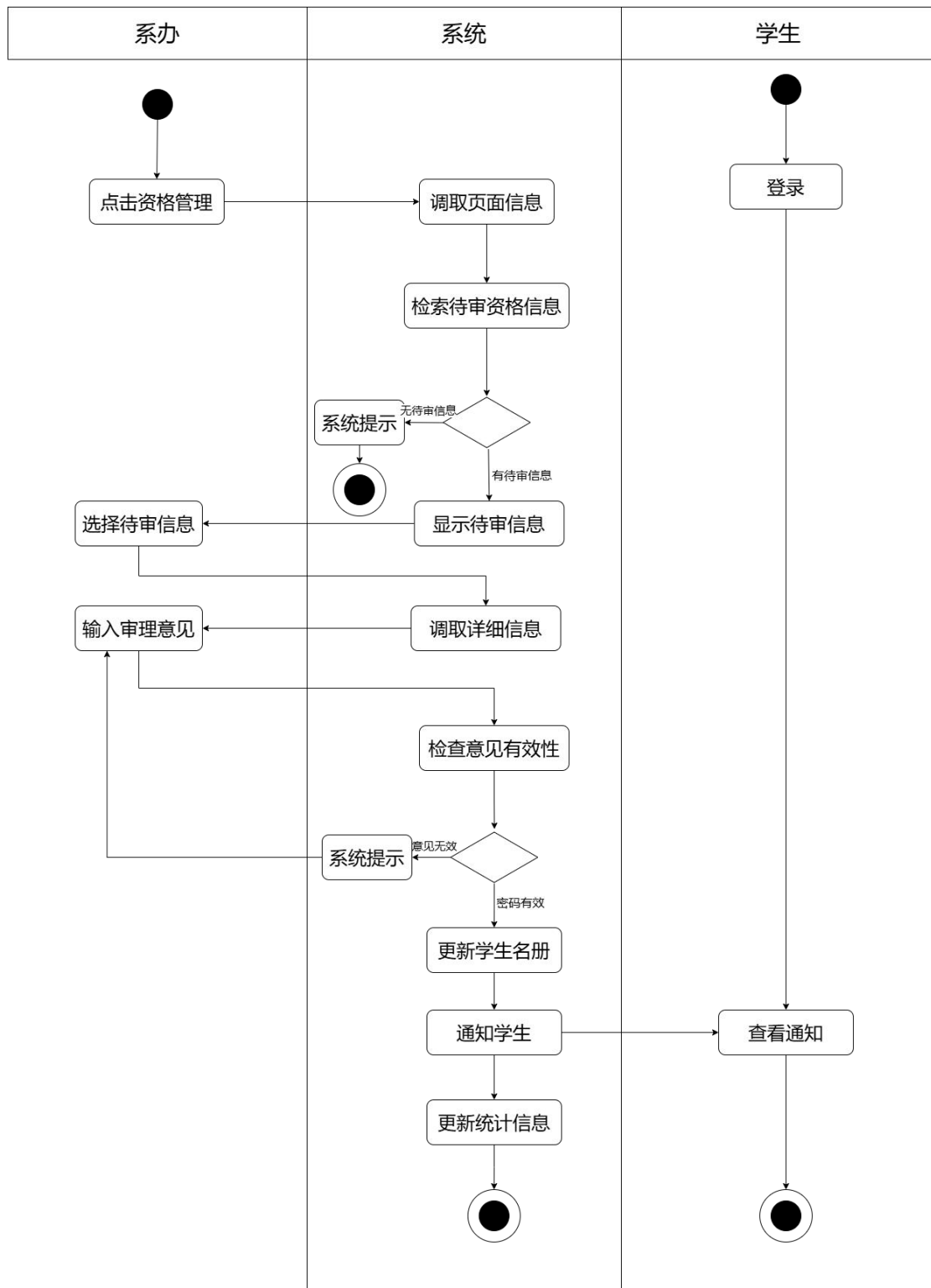
注册申请用例的活动图：



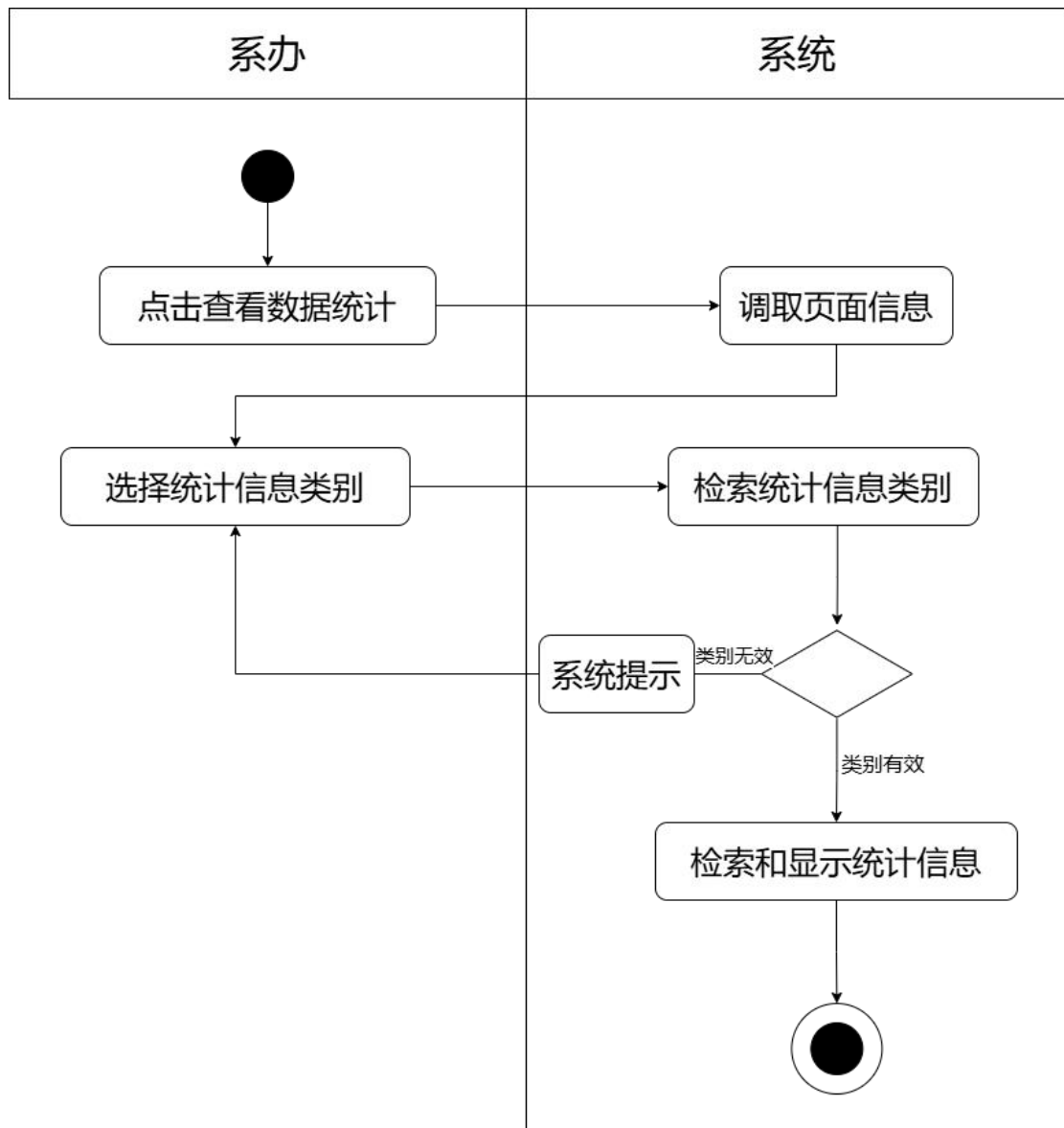
查看数据统计用例的活动图：



资格审核用例的活动图：



上传成绩用例的活动图：



3.2 分析时序图

注册申请时序图：

1. 参与对象：注册界面、注册处理模块、学生名册数据库。

2. 时序流程：

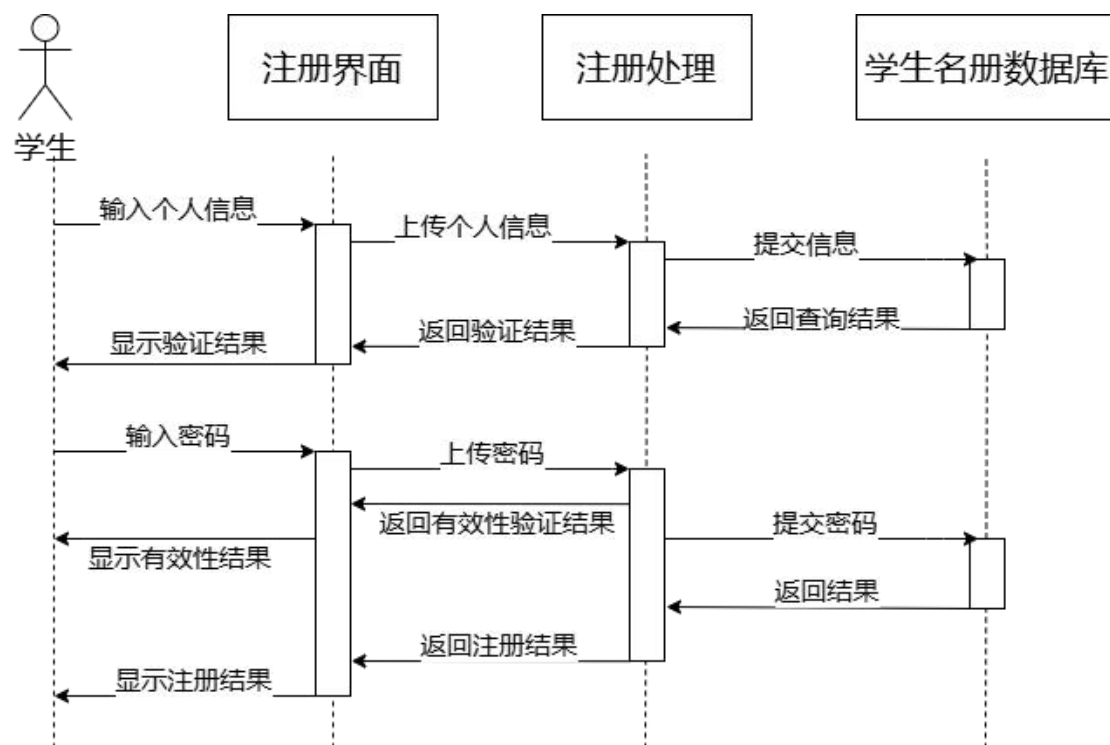
①学生在注册界面输入个人信息。

②系统的注册处理模块上传该个人信息至学生名册数据库，进行验证。

③若信息有效，返回验证通过的结果；若无效，则提示信息无效。

④学生继续输入密码，系统验证密码的有效性。

⑤若密码有效，系统在学生名册中更新新用户的注册信息并返回注册成功提示。



查看数据统计时序图：

1.参与对象：数据统计界面、数据统计处理模块、数据统计数据库。

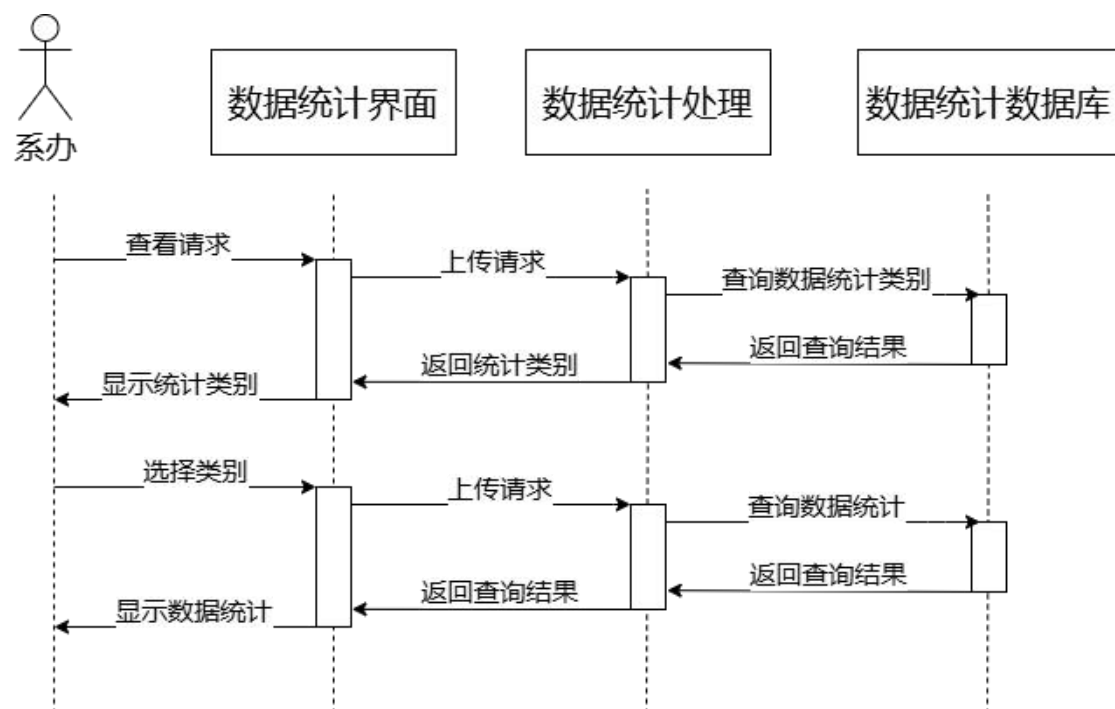
2.时序流程：

①系办用户在数据统计界面发起查看请求。

②数据统计处理模块接收到请求后查询可用的统计类别，并将结果返回至界面。

③系办用户选择所需查看的统计类别，系统在数据库中检索相应统计信息。

④数据统计处理模块返回查询的统计结果，并在界面中显示给用户。



资格审核时序图：

1.参与对象：资格审核界面、资格审核处理模块、数据库。

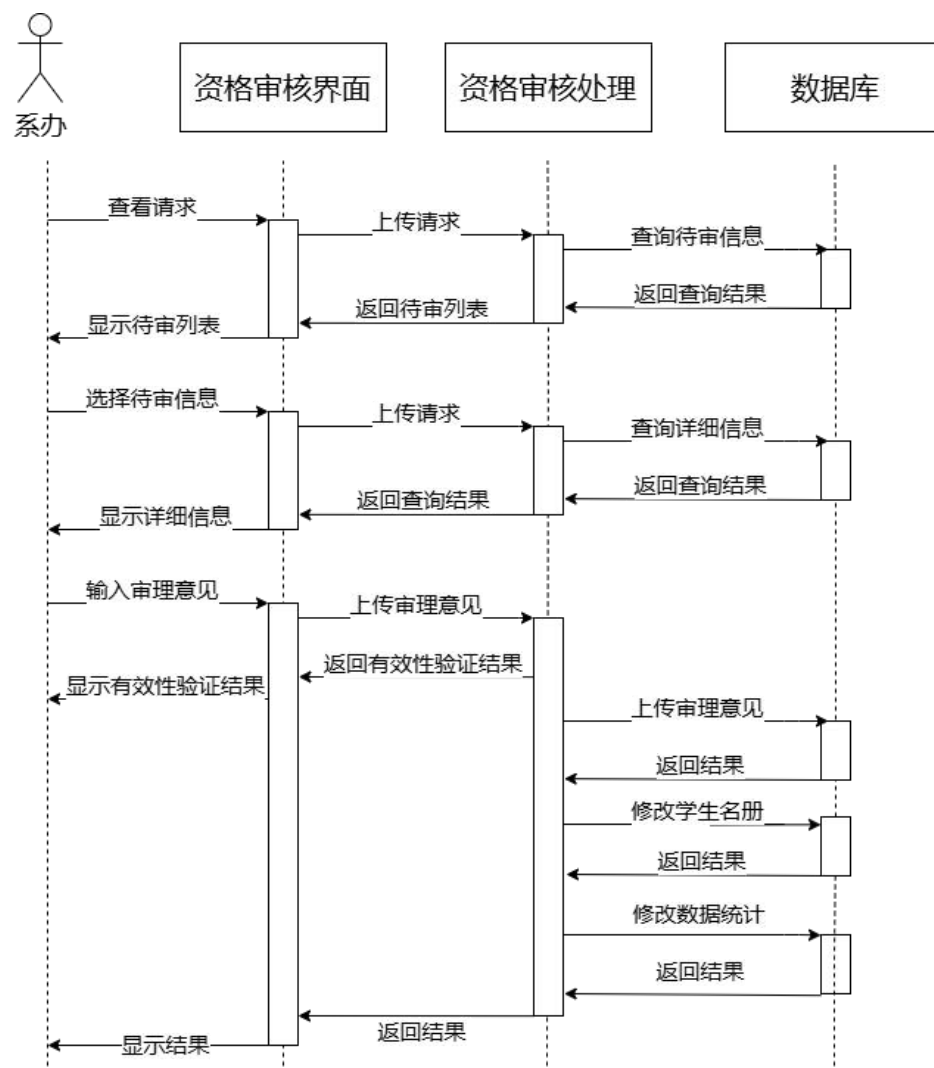
2.时序流程：

①系办在资格审核界面查看待审核的资格列表，系统从数据库中获取列表信息并返回。

②系办选择具体的审核信息，系统查询该信息的详细内容。

③系办填写审核意见并提交，系统验证意见的有效性。

④若审核意见有效，系统在数据库中更新资格审核状态，并向学生发送通知，同时更新数据统计信息。



上传成绩时序图：

1.参与对象：上传成绩界面、上传成绩处理模块、数据库。

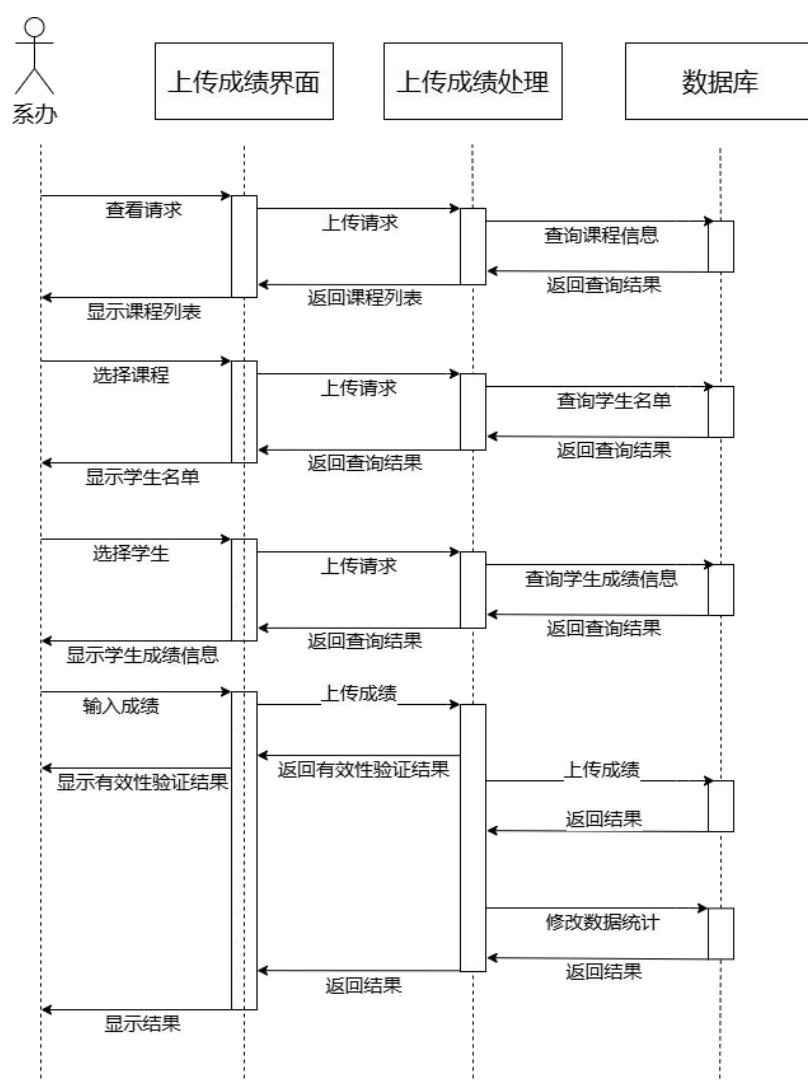
2.时序流程：

①教师在界面上发起查看课程列表的请求，系统从数据库中获取并显示课程信息。

②教师选择具体课程后，系统加载该课程的学生名单。

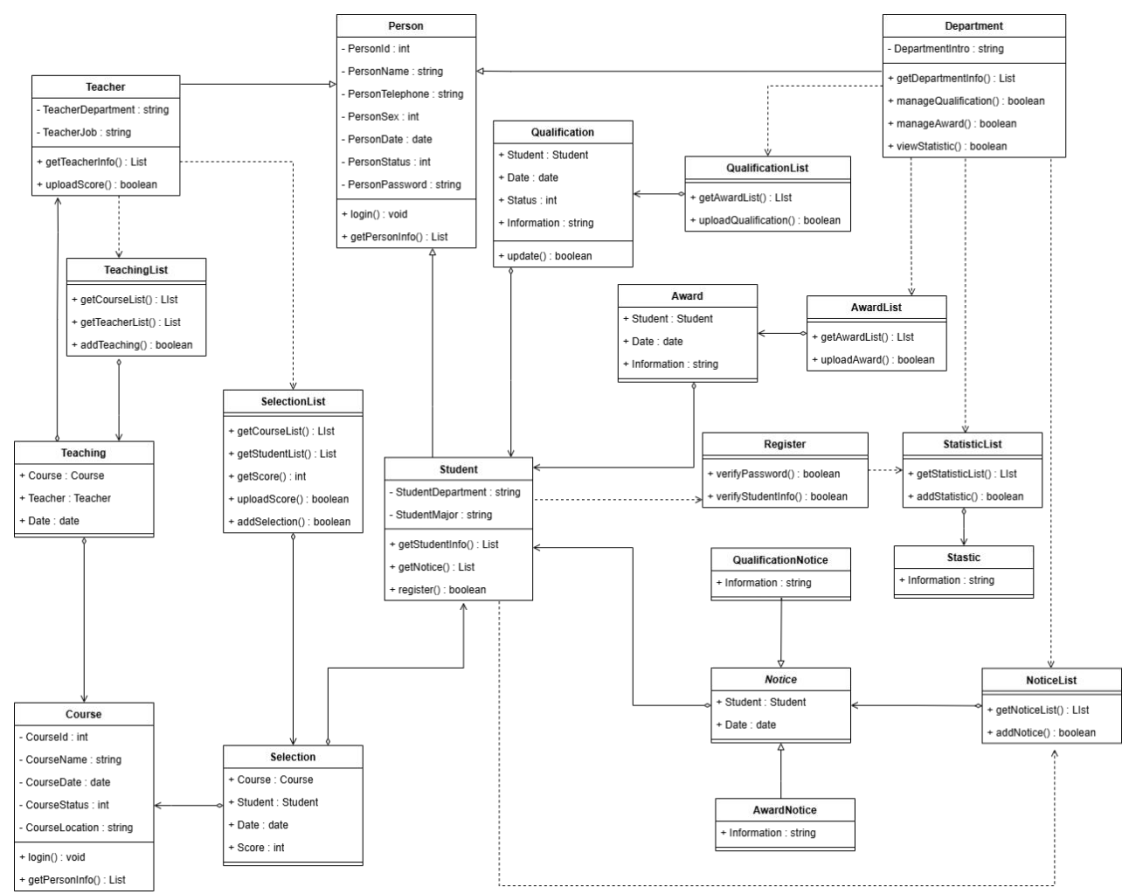
③教师选择学生并输入成绩，系统验证输入成绩的有效性。

④验证通过后，系统在数据库中更新该学生的成绩信息，同时更新数据统计信息并显示上传结果。



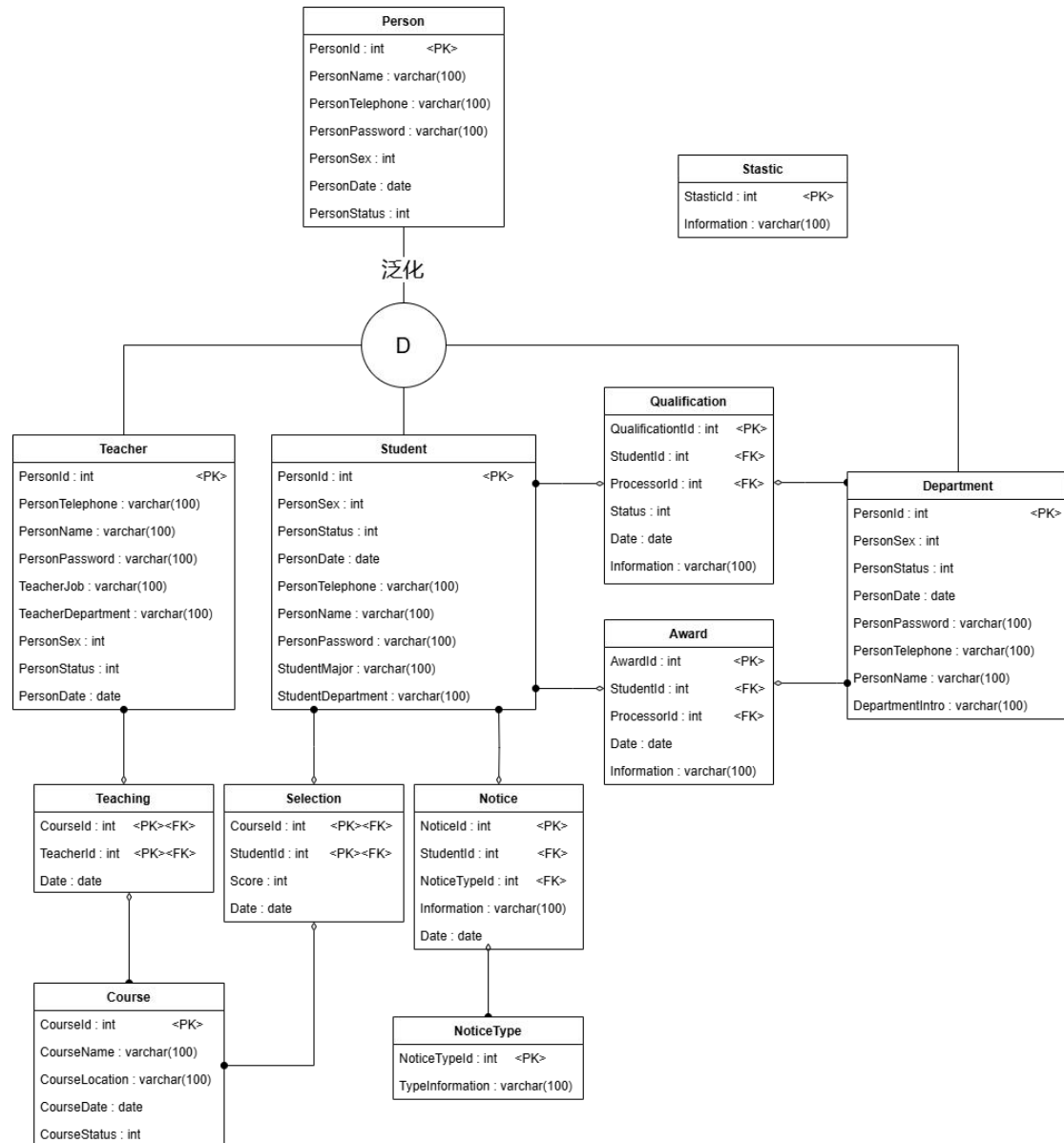
四、静态模型

根据以上分析，可以绘制如下类图：



五、数据库 ER 模型

根据以上分析，可以绘制如下 ER 图：



六、总结体会

本次系统分析与设计的实践围绕学籍管理系统展开，实践过程中逐步建立了功能模型、动态模型、静态模型和数据库 ER 模型，让我对面向对象思维和 UML 建模有了更加深入的理解。

这次建模实践让我更加理解面向对象方法的优势。与传统的数据流图相比，面向对象的 UML 模型让我们以不同的角度来思考系统结构，不再仅仅关注数据的流动，更强调对象的行为和彼此之间的关系。这种方法比传统的设计更具灵活性和可扩展性，使得系统不仅能满足当前的需求，还为未来可能的改动预留了空间。

在建模过程中，我感受到不同类型的模型之间存在着紧密的联系。用例图从宏观上描述了系统的核心功能，活动图和时序图则进一步展示了系统的动态行为 and 用户交互的详细过程，而类图则清晰地展示了系统的静态结构，将系统中的各个实体及其关系一一呈现出来。通过 ER 图，模型进一步向数据库结构延伸，使得数据能够持久化存储。这种从需求到系统结构的层层推进，让设计思路更加清晰，系统架构也更加完善。

总而言之，这次系统分析与设计实践让我深入体会到面向对象分析方法的优点，以及增长了绘制 UML 的各种图型的经验，收获繁多。