

【上图分析】

1. 不足：用例太多、关联交叉太多，图形看上去混乱。

建议：关联关系不要总是直线，适当地画成折线（方法： 按下 Ctrl键，鼠标 点中线条上某一点拖动）

1. 优点： 参与者的泛化关系，使得子参与者省略了许多与用例之间的关联关系！
2. 建议：增加课程、删除课程、修改课程： 定义一个“管理课程”，在用例描述里面说明子功能，比如，2.1 增加课程；2.2删除课程 2.3修改课程
3. 建议：教授 可以独立成为一个参与者。放到右上角。这样，线条交叉就可以避免
4. 建议：退回\通过、查看开图报告： 定义 一个 “处理开题报告”即可。 处理的不同意见，在用例描述中解释。
5. 博士生为何要查看开题报告？查看开题报告状态

【案例】

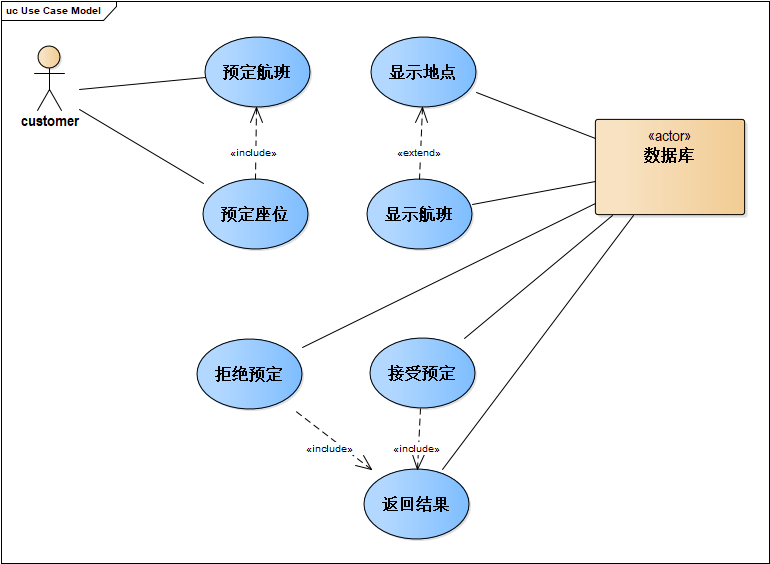
|  |  |
| --- | --- |
| Description | 数据库在接到用户的预定请求后，查看航班是否还有剩余座位或该预定座位是否被预定，若无剩余座位或想要预定的座位已被预定，则拒绝预定，并返回失败结果，否则预定成功，返回成功结果。 |
| Post-condition | 无 |
| exception | 无 |

分析：

1）用例描述太简单；用例描述最好一句话一句话分段，看上去清楚一点。

2）不能没有异常情况。大家在编程的时候 ，是否用到很多的 if …else if …,这些就是异常情况

【案例】



【分析】

1. 数据库 不能拒绝/接受预定。只能在 拒绝、接受预定的过程中，提供数据。这两个用例不是合适的用例，只是其他用例执行过程中的一种判断。比如 预定座位， 可以与 数据库之间有 关联关系
2. 预定航班 与 预定座位 之间的 include 关系，谁主？谁辅？是否反了。预定航班 包含预定座位，或者 预定航班 扩展 预定座位。有时不能预定座位我们也是买到了的航班，到机场时临时分配一个座位。
3. 返回结果：不是用例
4. 数据库 ：它为啥要显示航班、显示地点？ 用例不对、用例与参与者的关系也不对。

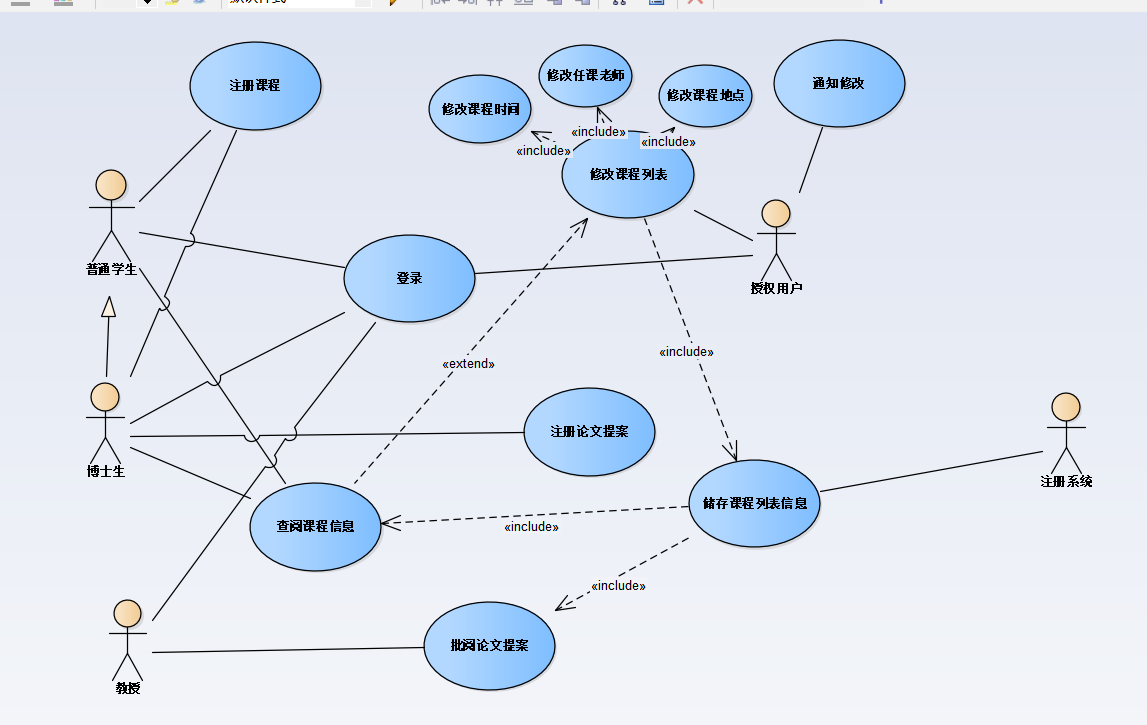
【案例】



【分析】、

1. 思路正确：通过参与者的位置调整，有效回避了 线条的交叉
2. 建议：不必要区分 Teacher 与Professor
3. 用例GetAllCourse、查看某一课程GetOneCourse ： 用例名称不是很合适。 既然是查看，用 InquiryCourse，会更容易让人理解。至于一门、两门、三门：用例描述里面说一下。
4. 参与者 AuthoriedUser: 没必要了。有些相关的功能，有一个代表参与者即可
5. 用例 Login 与 SignIn ： 有一个即可。用例描述是说一句：如果用户没有注册，需要先行注册，然后登陆”
6. 提醒：CourseRegister的主体应该是学生，而不是数据库。另外，这里的extend关系 不合适

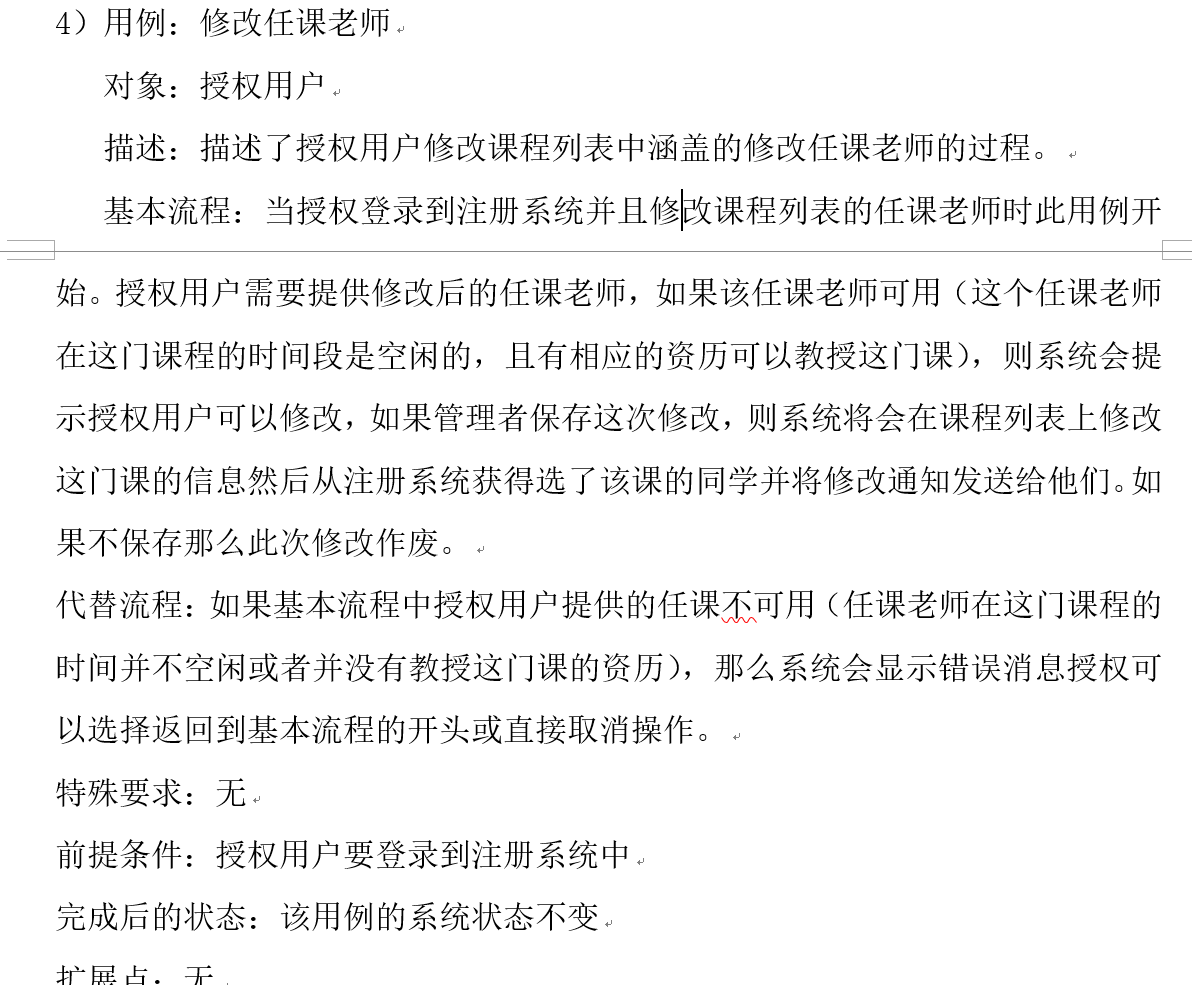
【案例】



分析

1. 各种虚线：构成了密密麻麻的环状。普通学生、博士生 与 登陆 之间的关系，删除 “博士生与登陆”这条线
2. 参与者 注册系统，应该用构造型
3. 其实，所有的用户都是 授权用户，需要登陆。因此，不需要专门的授权用户。这里需要一个系统管理人员
4. 课程修改列表哪里的一片用例，可以简化为 管理课程 一个用例

【案例】

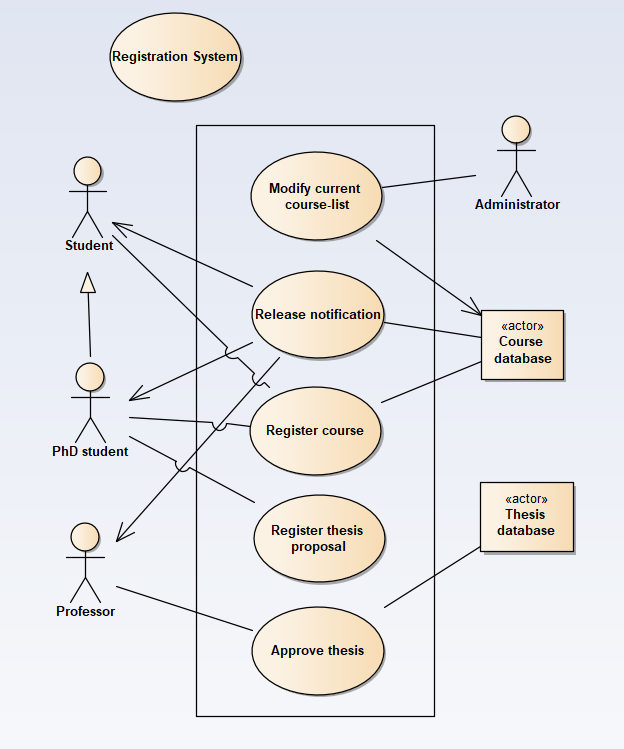


【分析】

这是最不吸引人的 用例描述方式。一大段一大段的文字。

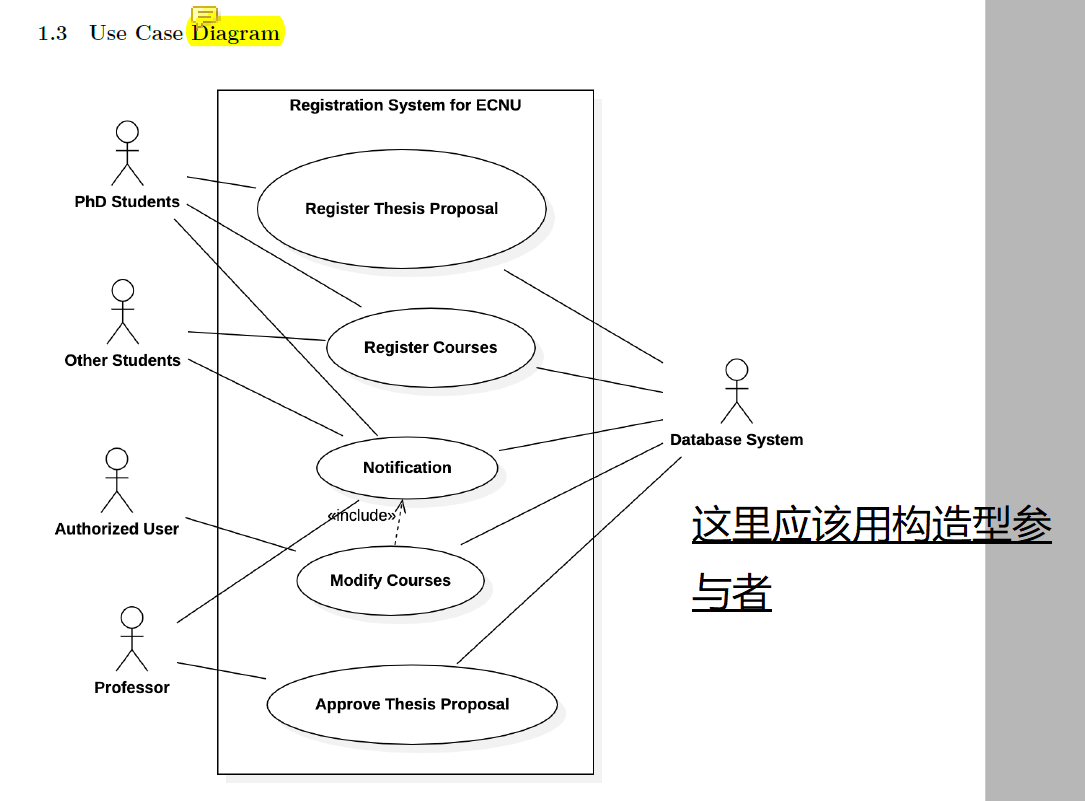
建议： 改为带序号的 一行一行的描述。并且必须有 异常事件的描述

【案例】

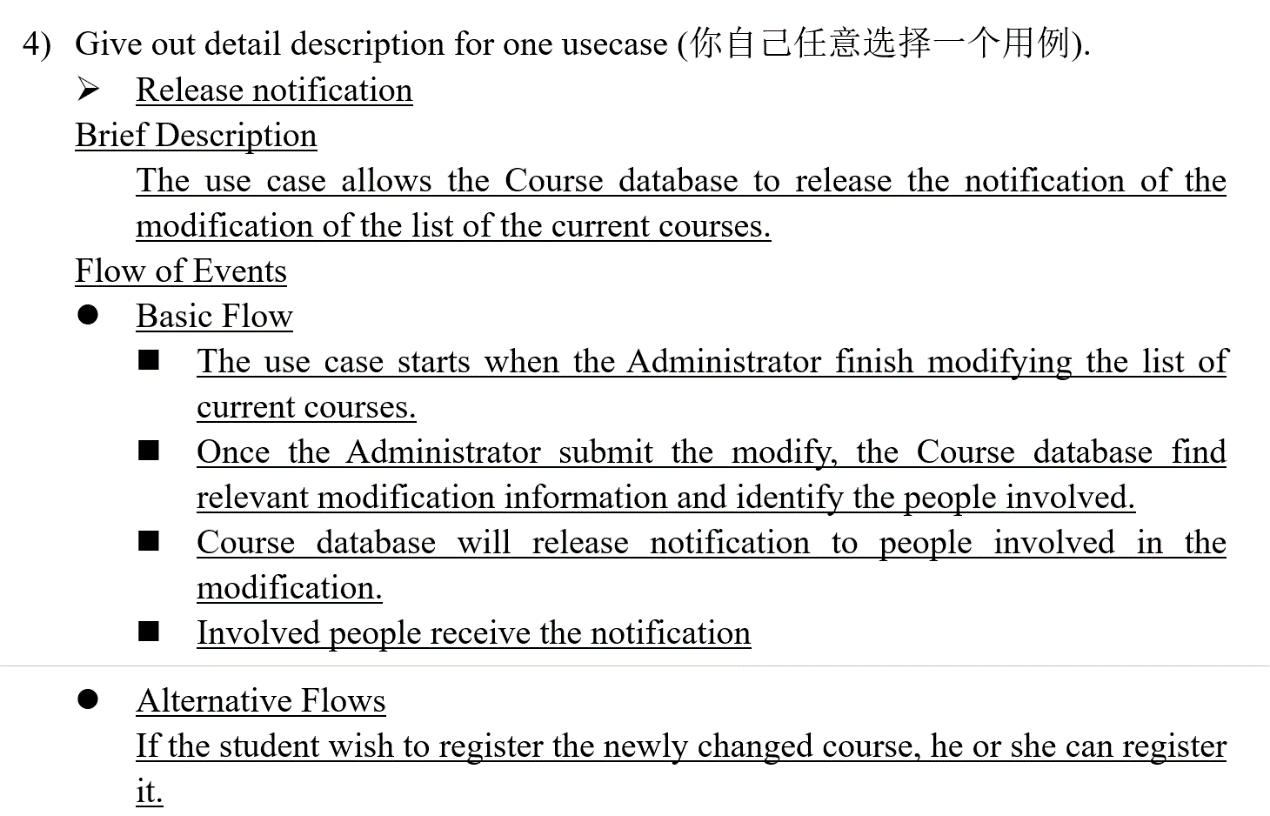


【分析】挺好的

【案例】



【案例】



【建议】

用例描述 还是用序号比较好！尤其是稍微复杂一点的情况下，序号显得清新。

在描述异常情况时，可以轻松描述 正常事件流的哪一步出了状况。

如

1

1.1

1.2

2

2.2

2.3

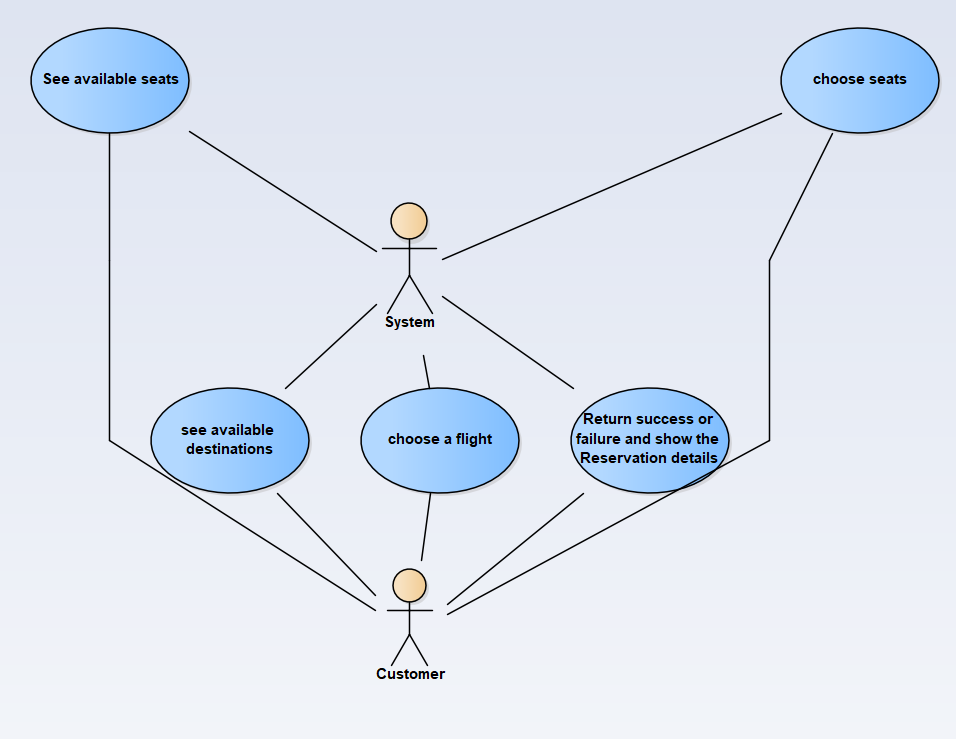
异常事件

1a : <表示上面 第1步，出了问题>该如何处理

1.1a : 上面 1.1出了问题1

1.1b : 上面1.1处理问题2

【案例】



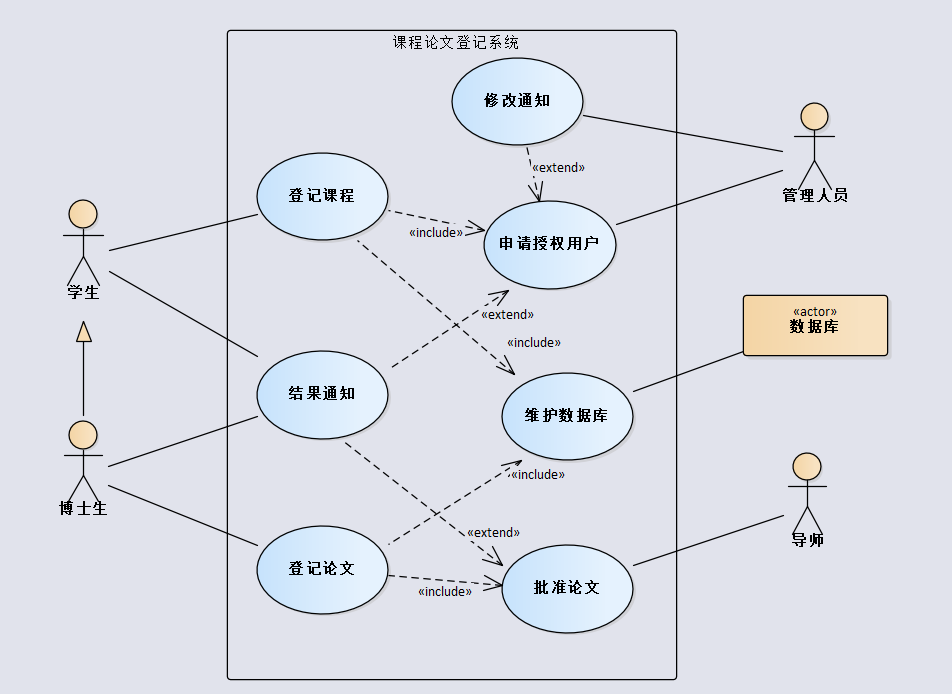
【分析】

1. 不美观！
2. System ： 不能作为参与者。这是有你要开发的 系统，不在你的系统 外面！
3. 大家记住 “System”是保留字，不能作为 用例名、类名、参与者名,等。实在需要用的时候，加上一个定语 ，MySystem、 BookSystem等。
4. 用例 Return success or failure and show the reservation details ，很多人都会犯这种错误。唯恐别人不理解，用一句话来作为用例名称！绝对不允许！

用例起名字的 准则：短小精悍的动名词，例如，改为 ReturenResult . 至于 result包含哪些内容，可以在 用例描述中叙述

1. Choose a flight : 可以 省去 a 。用例图中的 Avaiable 可以省略

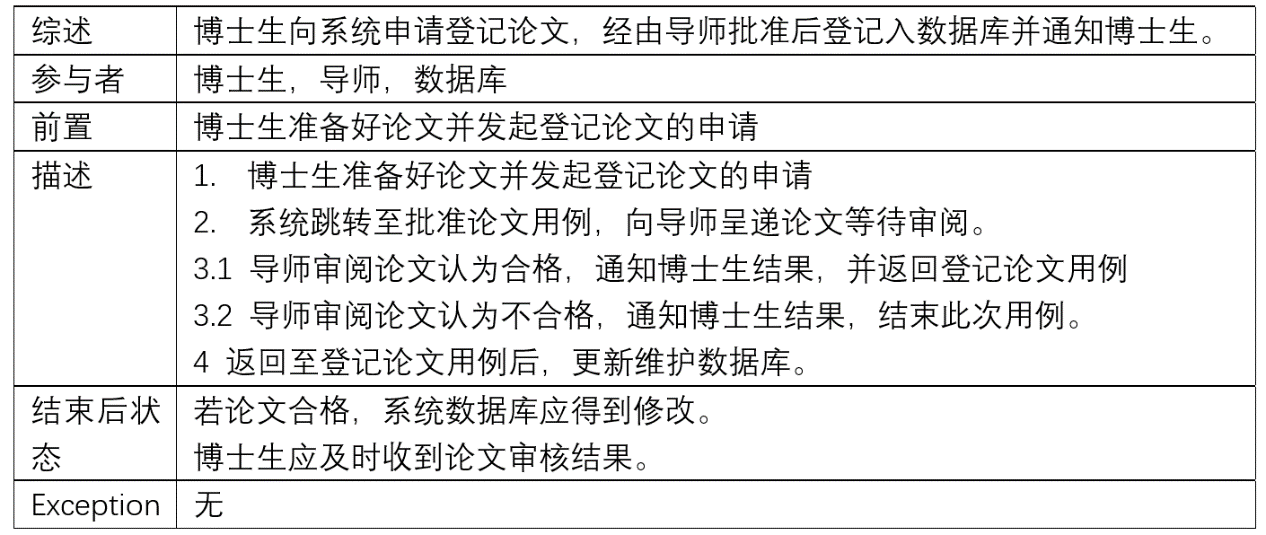
【案例】



【分析】

1. 登记论文 为何要 include 批准论文？
2. 登记论文确实是要往数据库中写入论文，但不需要强调 维护数据库，这是必然的。有些常规的额行为，不需要在建模是额外提出。就像人活着，重要的事情是呼吸，但谁也不会特别去强调。
3. 用例 结果通知： 是在要这个用例的话，改为 通知结果

【案例】



【分析】

1. 用例登记论文：为何没有exception?
2. 用例描述不是很精确，没有反应参与者之间的行为
3. 登记论文的时刻，老师不一定会审阅论文，所以，要把博士生的行为与导师的行为 分割开来
4. 异常举例

4a: 如果因为数据或者网络的原因导致论文提交失败，则本用例结束并通知 参与者

【老师分析结束。 作业太多，我只能随机抽了几篇，进行点评】