



第4章 面向对象的软件分析与设计

4.1 4+1模型及UML语言实现

4.2 面向对象的软件工程

4.3 用例图

4.4 活动图

4.5 用户界面设计

4.6 类图

4.7 交互图

4.8 包图

4.9 系统与子系统

4.10 部署图

4.3 用例图

4.3.1 概念

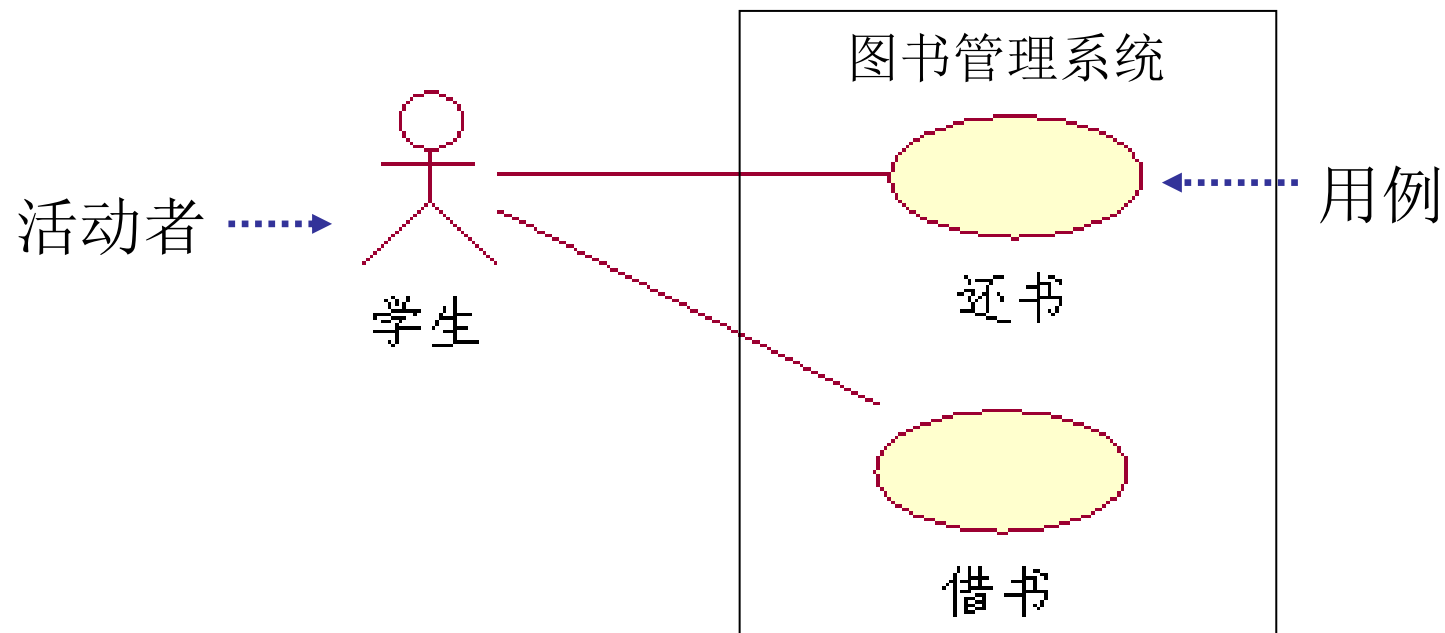
4.3.2 用例建模的原因

4.3.3 Case:图书管理系统

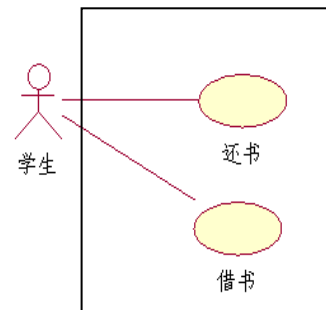
4.3.4 Case2:用户权限系统

4.3 用例图

4.3.1 概念



4.3 用例图



4.3.1 概念

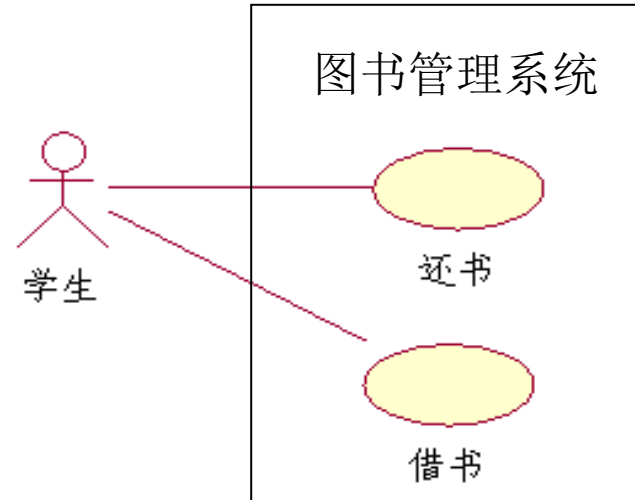
- 用例(**use case**): 用例是一系列活动(**actions**)的描述,系统通过完成这一系统的活动可以得到用户(**actor**)需要的结果。
 - ❖ 用例代表着系统的**功能需求**。

图书管理系统: 借书、还书
 - ❖ 用例从**系统层面**描述了一系列步骤(sequence), 这些步骤代表着系统外的事物(活动者actors)和系统(system)本身的交互(interaction)。
 - ❖ 对于每个用例, 可以从文本(text)、状态机(state machine)、交互图(interactions)等方面加以规范化(specification)。

4.3 用例图

4.3.1 概念

■ 用例(use case)

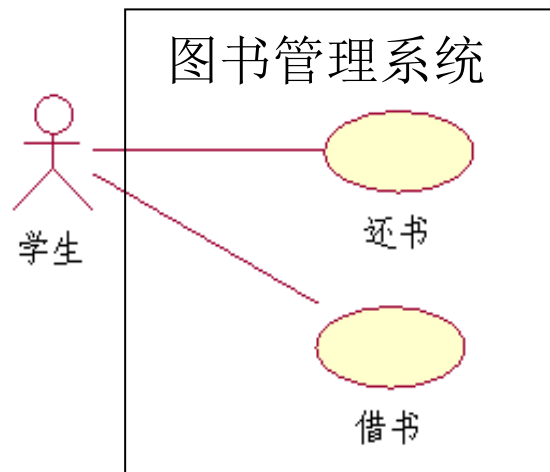


❖ 建模用例最主要的原因是为了对系统、子系统的
功能（行为）建模。

What that element does, not how it does it.

4.3 用例图

4.3.1 概念

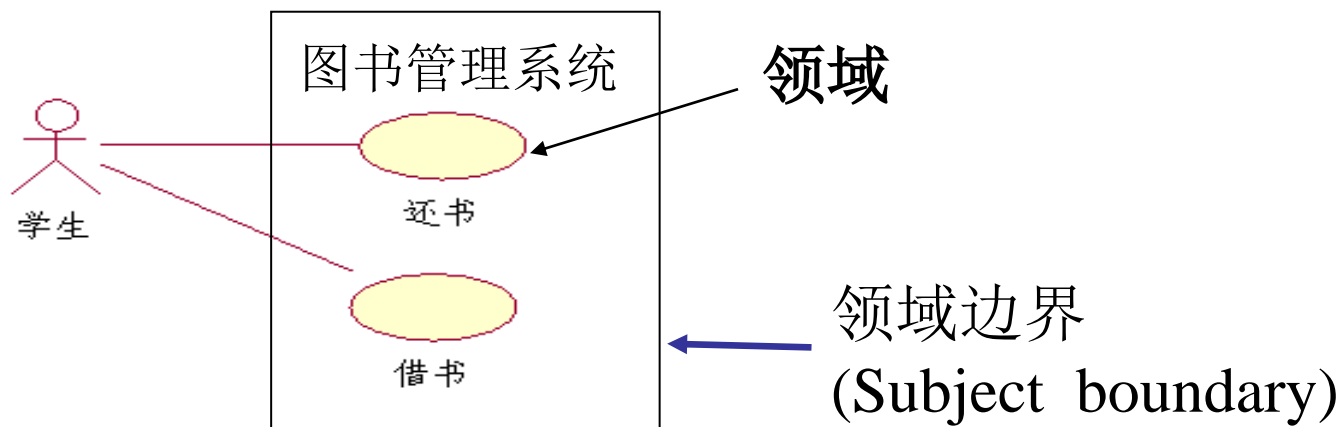


- 活动者(**actor**): 人、硬件或者其它系统与系统交互过程中所代表的角色(role)。
- ❖ 活动者(Actors)与用例(use cases)间只能是一种关联关系 (association)。
它们间的关联关系表明两者可相互通信的(communicate)。

4.3 用例图

4.3.1 概念

- 领域(subject): 系统或者子系统。由用例集描述。

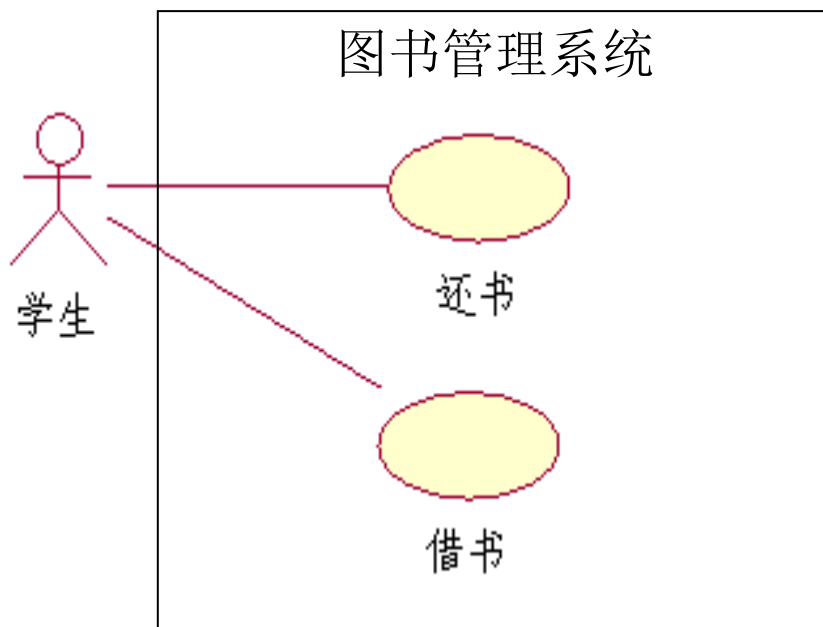


- ❖ 用例代表领域的功能（行为）。
- ❖ 活动者代表与领域交互的一方。
- ❖ 所有的用例描述了领域全部的功能（行为）。

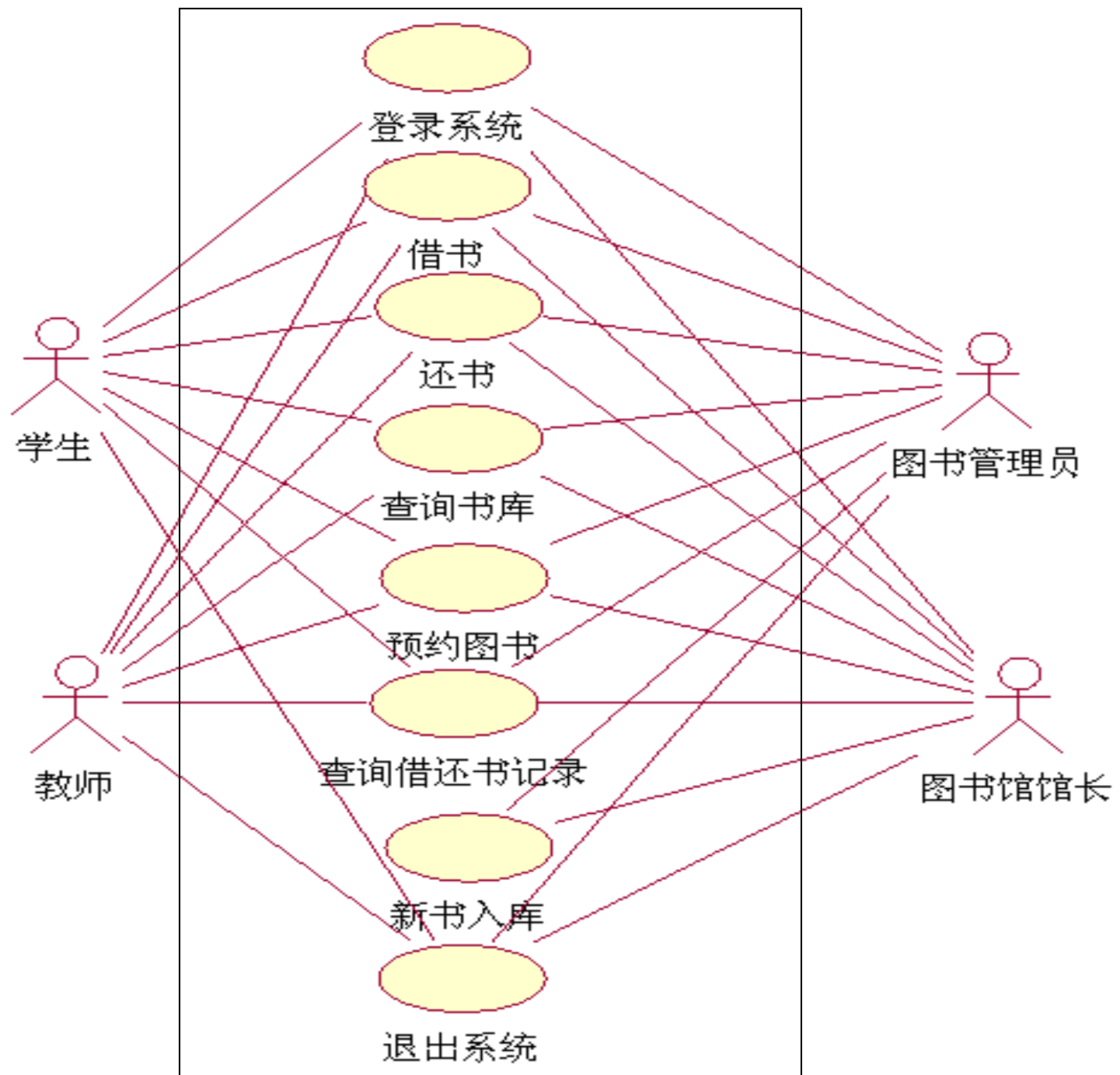
4.3 用例图

4.3.1 概念

- 用例图(use case diagram)：用例(use cases)、活动者(actors)和它们的关系。



❖ 用例图主要是用来对系统的功能（行为）进行建模。





4.3 用例图

4.3.2 用例建模的原因

- 便于领域专家(domain experts)和开发者从不同层面上描述、构建整个系统，同时也使得开发者能构建系统。

domain experts	:	outside view	描述
developers	:	inside view	构建

- 便于开发者理解系统。
- 用例可用于测试。（功能测试：黑盒测试）



4.3 用例图

4.3.3 用例建模

图书管理系统

■ 项目描述

图书管理系统问题陈述

问题陈述

- 我们为一所大学图书馆开发图书管理系统软件，该系统服务于全校师生。
- 图书馆每月初在网上公布新入库图书。
- 图书馆每天可办理借书证、退证业务。
- 用户可以通过系统查询库存信息（单机或网络）。
- 用户可以预约图书。
- 用户通过系统办理借书、还书业务。
- 图书馆将通过该软件为用户提供快捷、方便的服务。



4.3 用例图

假设

- 一种电子接口，例如WEB，可能对用户更适合。

风险因素

- 高：

- ❖ 某些程序员没有开发经验
- ❖ 系统应该使得非专业人员便于使用
- ❖ 如果不支持Web接口，我们是否会成功？

- 中：

- ❖ 如何处理并发用户

- 低：

- ❖ 如何在系统出错时防止丢失借、还书信息？
- ❖ 如果系统收到大量借书将会怎样？
- ❖ 如何处理数据库崩溃？

市场因素

- 在大多数大学中，设备等硬件条件都不错，办公人员以及师生 都习惯使用计算机处理日常事务，因此愿意采用图书管理系统。



4.3 用例图

4.3.3 用例建模

■ 确定系统边界(领域边界) (通过确定执行者和用例来确定系统边界)

❖ 确定执行者

- 谁使用这个系统? ← 教工、学生、馆员、系统管理员
- 谁安装系统? ← 系统管理员
- 谁启动系统? ← 系统管理员
- 谁维护系统? ← 系统管理员
- 谁关闭系统? ← 系统管理员
- 哪些其他的系统使用这个系统? ← 教务系统
- 谁从这个系统获取信息? ← 教工、学生、馆员、系统管理员
- 谁为这个系统提供信息? ← 馆员
- 是否有事情自动在预计的时间发生? ← 是否有图书超期

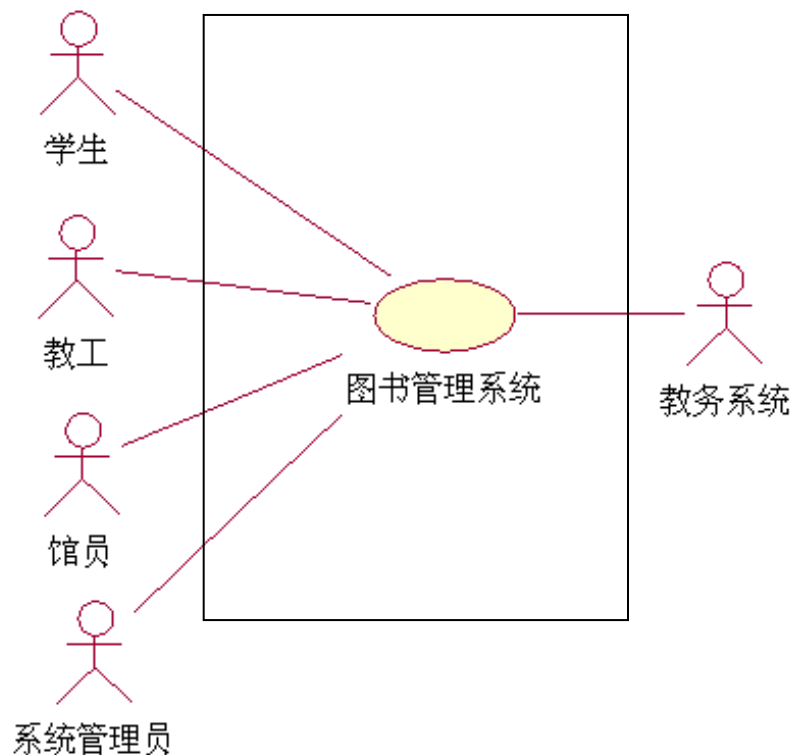
4.3 用例图

4.3.3 用例建模

■ 确定系统边界(领域边界)

❖ 确定执行者

执行者





4.3 用例图

4.3.3 用例建模

■ 确定系统边界(领域边界)

❖ 确定用例

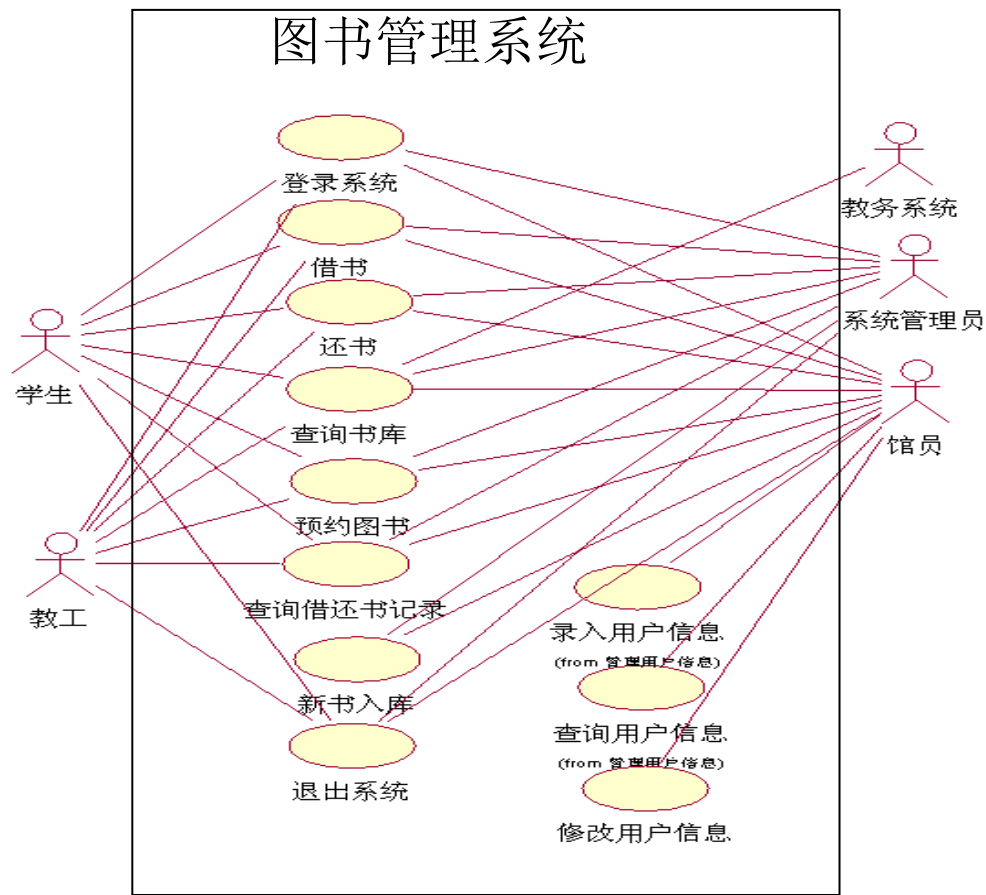
- 仔细检查所有的执行者，并为每一个执行者确定用例。
 - ✓ 执行者希望系统提供什么样的功能？
 - ✓ 系统存储信息吗？执行者将要创建、读取、更新 或删除什么信息？
 - ✓ 系统是否需要把自身内部状态的变化通知给执行者？
 - ✓ 系统必须知道哪些外部的事件？执行者将怎样通知系统这些事件？
 - ✓ 其它用例：启动、关闭、诊断、安装、维护、培训和改变商业过程。

4.3 用例图

4.3.3 用例建模

■ 确定系统边界(领域边界)

❖ 确定用例



4.3 用例图

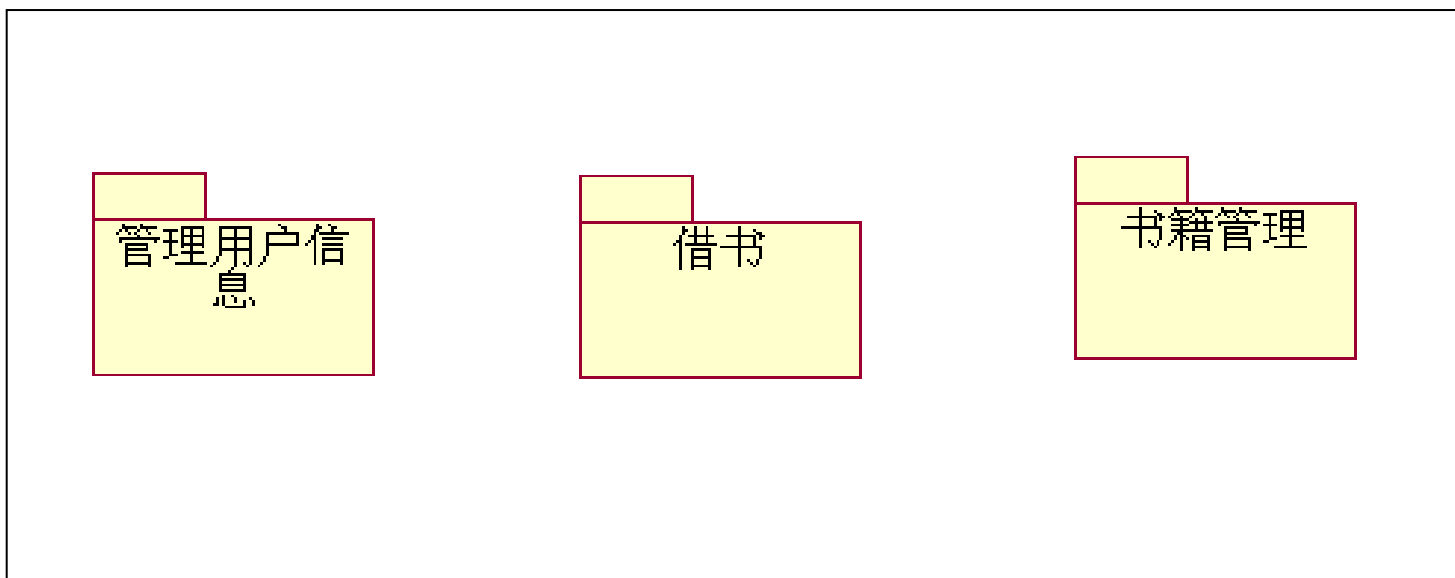
4.3.3 用例建模

■ 确定系统边界(领域边界)

❖ 确定用例

- 如果用例图过于庞大和杂乱：创建多个用例图

图书管理系统的用例包

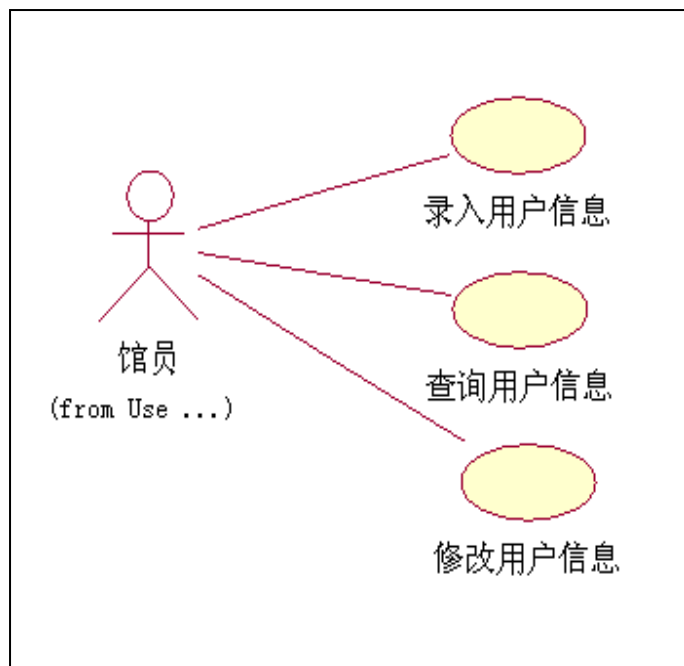


4.3 用例图

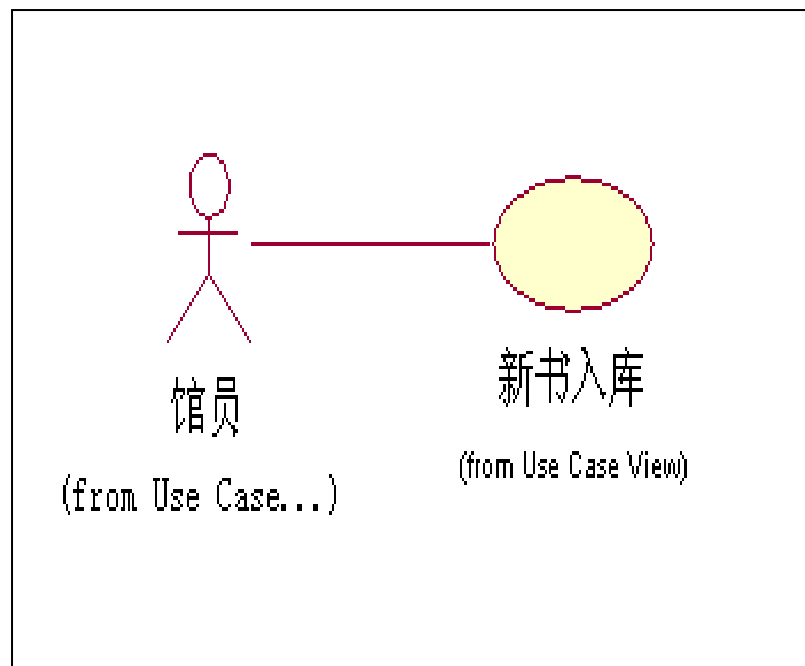
4.3.3 用例建模

■ 确定系统边界(领域边界)

❖ 确定用例

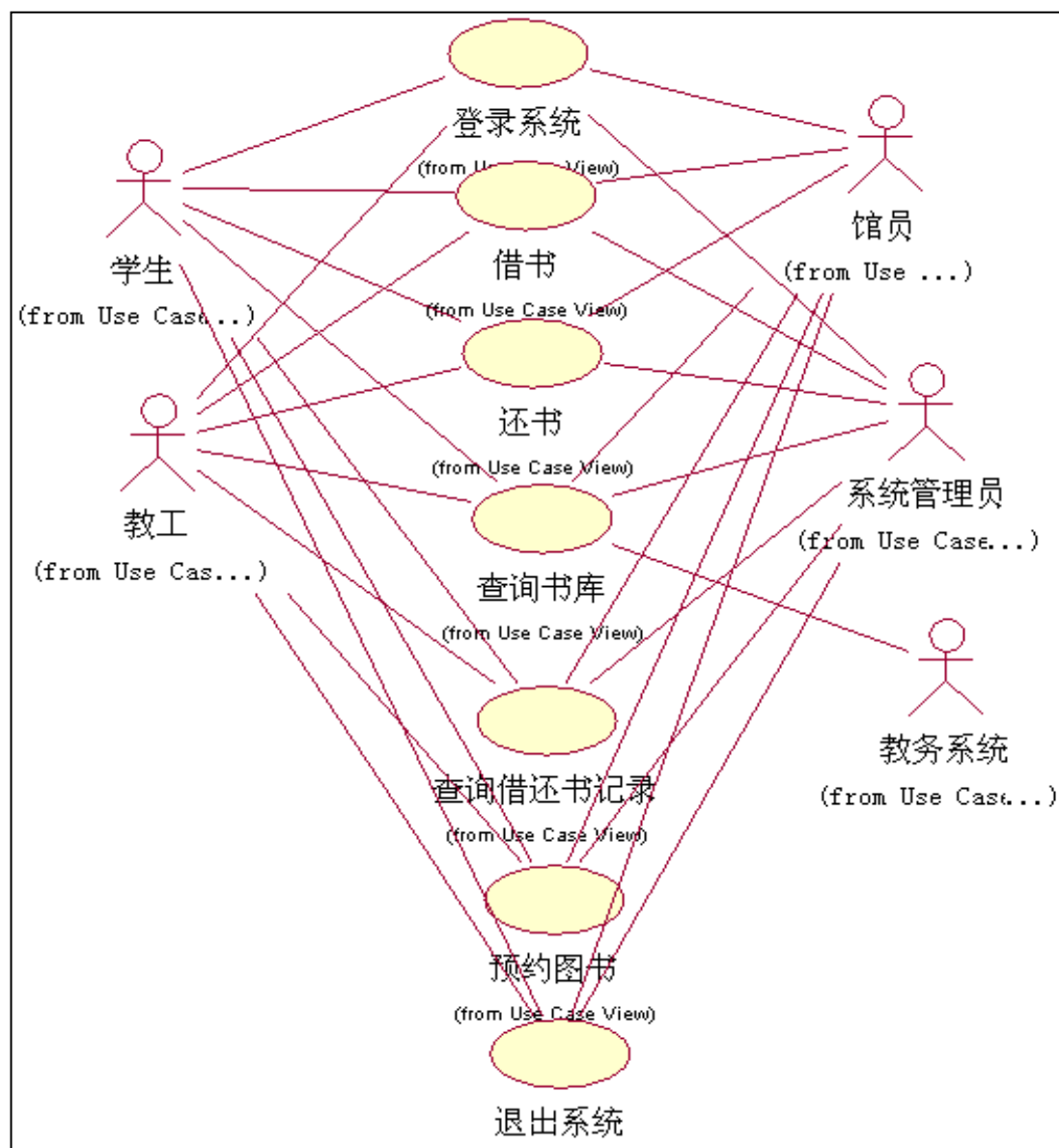


管理用户信息用例图



书籍管理用例图

4.3 用例图



借书用例图

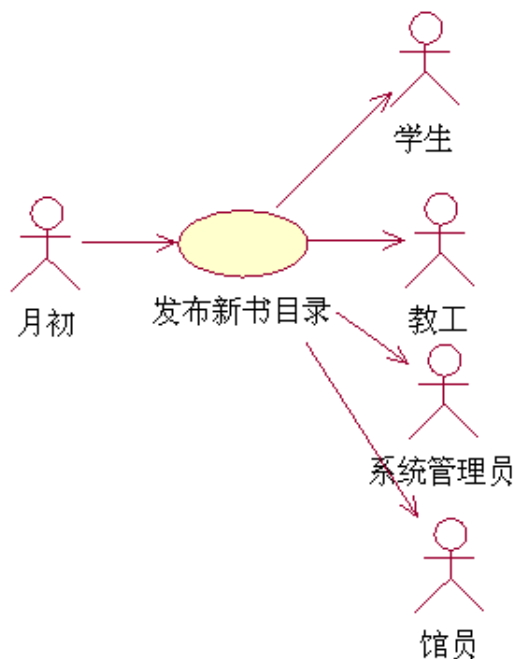
4.3 用例图

4.3.3 用例建模

■ 确定系统边界(领域边界)

❖ 确定用例

- 处理时间：某些活动发生在特定的时间
每月的月初（第一天）发布一次新书目录





4.3 用例图

4.3.3 用例建模

■ 确定系统边界(领域边界)

❖ 描述执行者和用例

图书管理系统**执行者描述**

- 馆员： 图书馆的教工，负责借书、还书、新书入库、收取超期图书的费用
- 教工： 大学的教师、教学辅助人员、行政人员，可以通过系统查询、预定图书和借书。
- 教务系统： 教务系统是另外一个系统，该系统可通过图书管理系统提供的接口查询书库的信息。
- 系统管理员： 图书馆的机房人员，负责系统的启动、维护、关闭。
- 学生： 大学的学生(本科生、硕士、博士)，可以通过系统查询、预定图书和借书。
- 月初： 是一个特定的执行者，在每个月的初期，由系统自动启动一个发布新书目录的用例。



4.3 用例图

4.3.3 用例建模

■ 确定系统边界(领域边界)

❖ 描述执行者和用例

图书管理系统用例描述

- 查询借还书记录：用户查询自己借还书记录
- 查询书库：用户查询馆藏图书
- 登录系统：用户通过用户和密码进入系统
- 发布新书目录：每月发布一次新书入库情况
- 还书：用户归还图书
- 借书：用户借书
- 退出系统：完成所有操作，退出系统
- 新书入库：馆员将新书情况输入系统
- 预约图书：用户预约想借的书



4.3 用例图

4.3.3 用例建模

■ 归档用例

❖ 基本用例

细节：告诉人们需要完成哪些步骤才能实现这个用例的功能。

- 基本功能、
- 所有可选方案、
- 异常情况、
- 进入用例之前必须正确的一切、
- 退出用例后必须正确的一切。



4.3 用例图

4.3.3 用例建模

■ 归档用例

❖ 基本用例

登录系统用例

前置条件:


事件流:

1. 当用户通过浏览器输入图书管理系统地址: library.ecust.edu.cn,用例开始。
2. 系统显示登录界面。
3. 用户输入用户名和密码。
4. 用户选择提交。
5. 系统检测用户信息,如是非法用户,系统提示用户为非法用户,同时提示用户选择退出系统或者重新登录。如果选择重新登录,就回到第2步;如果选择退出系统,用例结束。
6. 系统检测用户信息,如是合法用户,系统设置访问权限,并显示主界面,用例结束。

后置条件: 用户进入系统,系统显示主界面或者不能进入系统。

前置条件: 用例开始时系统必须处在什么状态。

后置条件: 用例结束时系统必须处在什么状态。无论如何,后置条件必须为真



前置条件：到了公司每个月上交税金的日期
事件流：

1. 当系统检测 到了上交税金的日期，用例开始。
2. 系统确定这个月税金总量。
3. 系统发送一个电子支付税金给政府。
4. 系统核实上交税金已经被接收，用例结束。

后置条件：政府已经接收了税金，并且更新了公司的税金记录。

上交税金用例中前后置条件的例子

4.3.3 用例建模

■ 归档用例

❖ 基本用例

- 事件流：一系列陈述句，它列出了用例的各个步骤。
分支(if)、循环(for、while)

前置条件：

事件流：

1. 当用户通过浏览器输入图书管理系统地址：library.ecust.edu.cn,用例开始。
2. 系统显示登录界面。
3. 用户输入用户名和密码。
4. 用户选择提交。
5. 系统检测用户信息，如是非法用户(if---)，系统提示用户为非法用户，同时提示用户选择退出系统或者重新登录。
 - a) 如果选择重新登录，就回到第2步；
 - b) 如果选择退出系统，用例结束。
6. 系统检测用户信息，如是合法用户(if---)，
 - a) 系统设置访问权限，并显示主界面，用例结束。

后置条件：用户进入系统，系统显示主界面或者不能进入系统。

在登录系统用例中使用if示例

4.3.3 用例建模

■ 归档用例

❖ 基本用例

- 事件流： 分支(if)、循环(for、while)

前置条件： 一个合法的用户已经登录系统

事件流：

1. 当用户选择借书时,用例开始。
2. 系统查询用户借书记录。
 - a) 如有超期图书，用例结束。
3. 系统显示借书界面。
4. 用户输入条形码
 - a) 系统查询用户的借书记录，如超出所借图书数目(8本)，用例结束。
 - b) 系统给出书籍的具体信息：书名、作者、出版社、日期。
 - c) 用户可以删除不需要借的图书。
 - d) 系统保存所借图书。
 - e) 用户选择退出，用例结束。

后置条件： 系统保存所借图书或者超出所借图书数目，不能借书。

借书用例中使用while循环

4.3.3 用例建模

■ 归档用例

❖ 基本用例

- 事件流： 分支(if)、循环(for、while)

前置条件： 一个合法的用户已经登录系统

事件流：

1. 当用户选择借书时,用例开始。
 2. 系统查询用户借书记录。
 - a) 如有超期图书，用例结束。
 3. 系统显示借书界面。
 4. 用户输入条形码。
 5. 对于每一个输入的条形码。
 - a) 系统查询用户的借书记录，如超出所借图书数目(8本)，用例结束
 - b) 系统给出书籍的具体信息：书名、作者、出版社、日期
 - c) 用户可以删除不需要借的图书。
 - d) 系统保存所借图书。
 - e) 用户选择退出，用例结束。
- 循环结束。

后置条件： 系统保存所借图书或者超出所借图书数目，不能借书。

借书用例
中使用
for循环



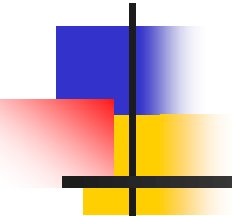
4.3 用例图

4.3.3 用例建模

■ 归档用例

❖ 其他需求（非功能性需求）：安全性、稳定性、可维护性、性能、负载等。

如果这些需求是针对一个特定的用例，可以在用例描述中添加一个专门的需求部分。



前置条件：一个合法的用户已经登录系统

事件流：

1. 当用户选择借书时,用例开始。
 2. 系统查询用户借书记录。
 - a) 如有超期图书，用例结束。
 3. 系统显示借书界面。
 4. 用户输入条形码
 - a) 系统查询用户的借书记录，如超出所借图书数目(8本)，用例结束。
 - b) 系统给出书籍的具体信息：书名、作者、出版社、日期。
 - c) 用户可以删除不需要借的图书。
 - d) 系统保存所借图书。
 - e) 用户选择退出，用例结束。
- 循环结束。

后置条件：系统保存所借图书或者超出所借图书数目，不能借书。

特殊需求：系统必须在2秒内响应用户的输入。

借书用例中的非功能性需求



4.3 用例图

4.3.3 用例建模

■ 归档用例

❖ 处理复杂的用例

事件流：基本路径+可选路径

基本路径：一系列没有分支的简单陈述句。
每一步都假设是正确的。



前置条件:

事件流:

1. 当用户通过浏览器输入图书管理系统地址: library.ecust.edu.cn,用例开始。
2. 系统显示登录界面。
3. 用户输入用户名和密码。
4. 用户选择提交。
5. 系统检测用户信息,如是非法用户(if---),系统提示用户为非法用户,同时提示用户选择退出系统或者重新登录。
 - a) 如果选择重新登录,就回到第2步;
 - b) 如果选择退出系统,用例结束。
6. 系统检测用户信息,如是合法用户(if---),
 - a) 系统设置访问权限,并显示主界面,用例结束。

后置条件: 用户进入系统,系统显示主界面或者不能进入系统。

登录系统用例



前置条件:

事件流:

基本路径

1. 当用户通过浏览器输入图书管理系统地址:
library.ecust.edu.cn, 用例开始。
2. 系统显示登录界面。
3. 用户输入用户名和密码。
4. 用户选择提交。
5. 系统显示主界面, 用例结束。

后置条件: 用户进入系统, 系统显示主界面

登录系统用例的基本路径



前置条件：一个合法的用户已经登录系统

事件流：

1. 当用户选择借书时,用例开始。
 2. 系统查询用户借书记录。
 - a) 如有超期图书，用例结束。
 3. 系统显示借书界面。
 4. 用户输入条形码
 - a) 系统查询用户的借书记录，如超出所借图书数目(8本)，用例结束。
 - b) 系统给出书籍的具体信息：书名、作者、出版社、日期。
 - c) 用户可以删除不需要借的图书。
 - d) 系统保存所借图书。
 - e) 用户选择退出，用例结束。
- 循环结束。

后置条件：系统保存所借图书或者超出所借图书数目，不能借书。

借书用例



事件流:

基本路径

1. 当用户选择借书时,用例开始。
 2. 系统查询用户借书记录。
 3. 系统显示借书界面。
 4. 用户输入条形码
 - a) 系统给出书籍的具体信息: 书名、作者、出版社、日期
 - b) 用户可以删除不需要借的图书。
 - c) 系统保存所借图书。
 - d) 用户选择退出,用例结束。
- 循环结束。

借书用例的基本路径



4.3 用例图

4.3.3 用例建模

■ 归档用例

❖ 处理复杂的用例

事件流：基本路径+可选路径

- 可选路径：允许不同事件序列的路径。
方法：沿着基本路径一条一条去找，并且考虑：
 - ✓ 在这个点上还可以执行别的活动吗？
 - ✓ 在这个点上有没有有什么可能出错的？
 - ✓ 有什么随时可能发生的行为吗？

事件流:

基本路径

1. 当用户选择借书时,用例开始。
 2. 系统查询用户借书记录。
 3. 系统显示借书界面。
 4. 用户输入条形码
 - a) 系统给出书籍的具体信息: 书名、作者、出版社、日期。
 - b) 用户可以删除不需要借的图书。
 - c) 系统保存所借图书
 - d) 用户选择退出, 用例结束。
- 循环结束。

可选路径

在第2步, 如果有超期的图书, 用例结束。

在第4.a步前, 系统检查用户的借书记录, 如超出所借图书数目, 用例结束。

后置条件: 系统保存所借图书或者超出所借图书数目, 不能借书。

借书用例的可选路径



借书用例

前置条件：一个合法的用户已经登录系统

事件流：

1. 当用户选择借书时,用例开始。
 2. 系统查询用户借书记录。
 - a) 如有超期图书，用例结束。
 3. 系统显示借书界面。
 4. 用户输入条形码
 - a) 系统查询用户的借书记录，如超出所借图书数目(8本)，用例结束。
 - b) 系统给出书籍的具体信息：书名、作者、出版社、日期。
 - c) 用户可以删除不需要借的图书。
 - d) 系统保存所借图书。
 - e) 用户选择退出，用例结束。
- 循环结束。

后置条件：系统保存所借图书或者超出所借图书数目，不能借书。

直接往用例文本中**添加可选路径**



4.3 用例图

4.3.3 用例建模

■ 归档用例

- ❖ 场景(Scenario): 任何一种贯穿用例的特定路径。
一个场景代表着用例的一个实例。

借书用例

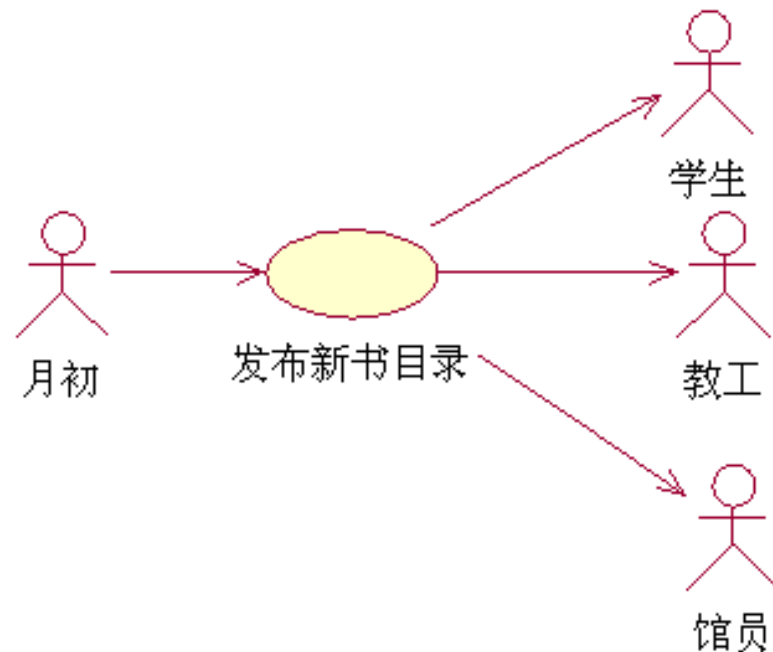
- 一个没有超期的借书单的到达。
- 一个有超期的借书单的到达。
- 一个超出了借书数目的借书单的到达。

4.3 用例图

4.3.3 用例建模

■ 归档用例

❖ 添加通信关联的指向：



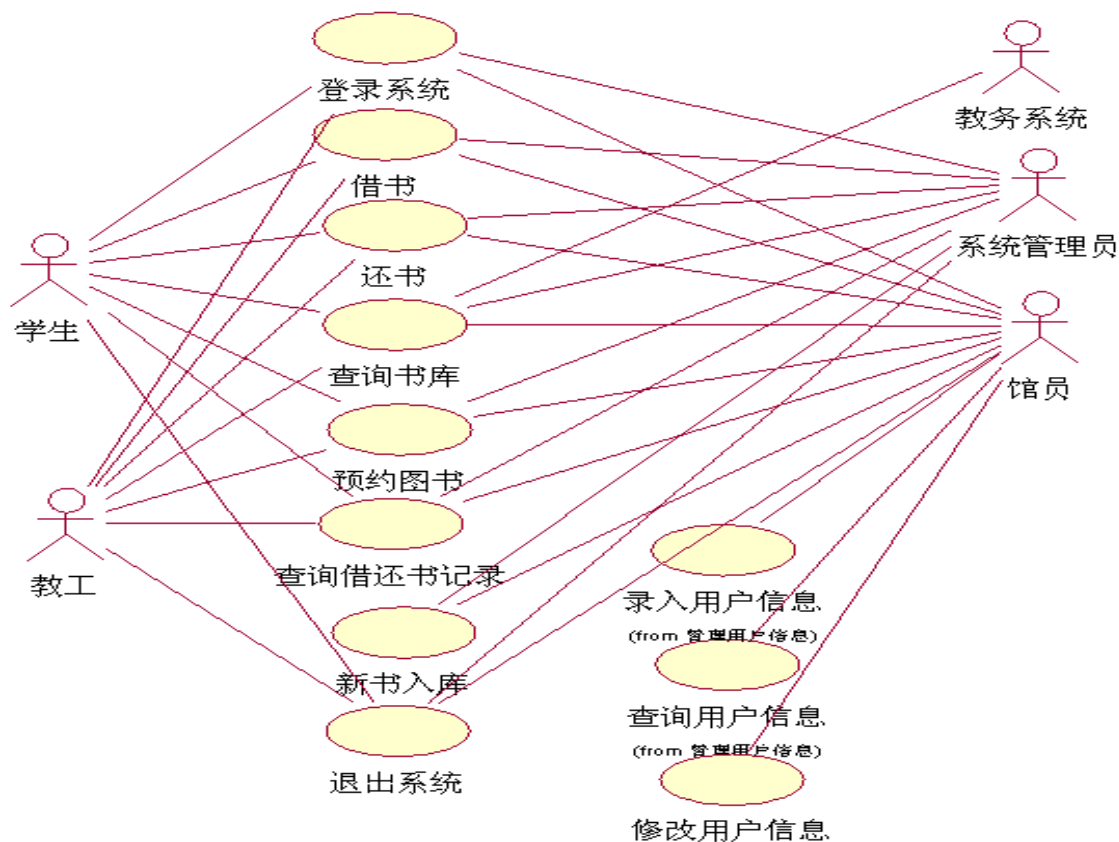
4.3 用例图

4.3.3 用例建模

■ 高级归档用例技术 (用例间的关系)

❖ include(包含)

系统原来的用例图中每个用例是否可以简化？

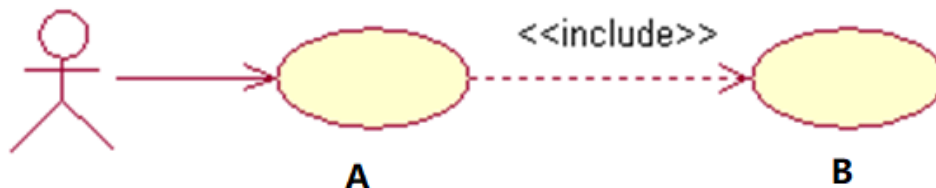


4.3 用例图

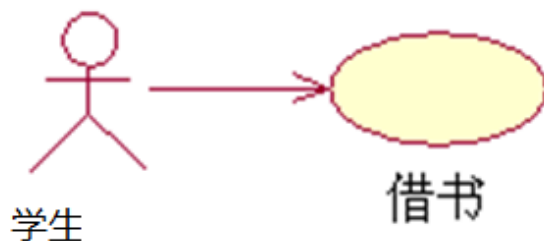
4.3.3 用例建模

■ 高级归档用例技术 (用例间的关系)

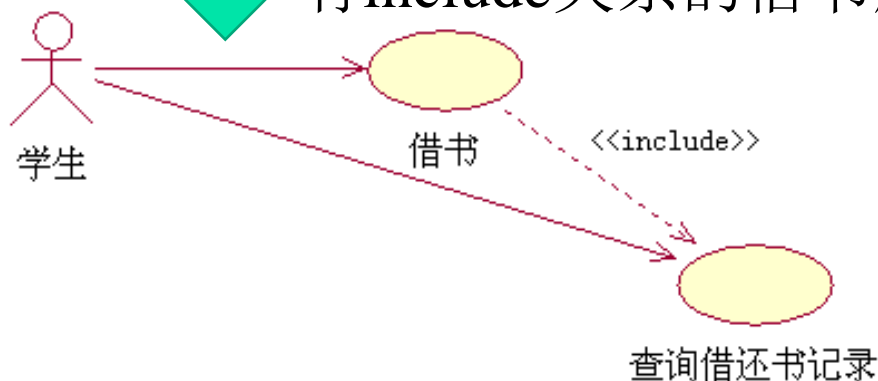
❖ include(包含)



例



有include关系的借书用例图



说明:

包含用例不完整, 被包含用例不依赖于任何包含它的用例。

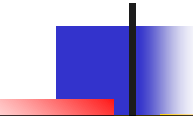
前置条件：一个合法的用户已经登录系统

事件流：

1. 当用户选择借书时,用例开始。
2. 系统查询用户借书记录。
 - a) 如有超期图书，用例结束。
3. 系统显示借书界面。
4. 用户输入条形码
 - a) 系统查询用户的借书记录，如超出所借图书数目(8本)，用例结束。
 - b) 系统给出书籍的具体信息：书名、作者、出版社、日期。
 - c) 用户可以删除不需要借的图书。
 - d) 系统保存所借图书。
 - e) 用户选择退出，用例结束。
 - a) 循环结束。

后置条件：系统保存所借图书或者超出所借图书数目，不能借书。

原借书用例




前置条件：一个合法的用户已经登录系统

事件流：

1. 当用户选择借书时,用例开始。
 2. **包含查询用户借书记录。**
 - a) 如有超期图书,用例结束。
 3. 系统显示借书界面。
 4. 用户输入条形码
 - a) **包含查询用户的借书记录** , 如超出所借图书数目(8本) , 用例结束。
 - b) 系统给出书籍的具体信息: 书名、作者、出版社、日期。
 - c) 用户可以删除不需要借的图书。
 - d) 系统保存所借图书。
 - e) 用户选择退出,用例结束。
- 循环结束。

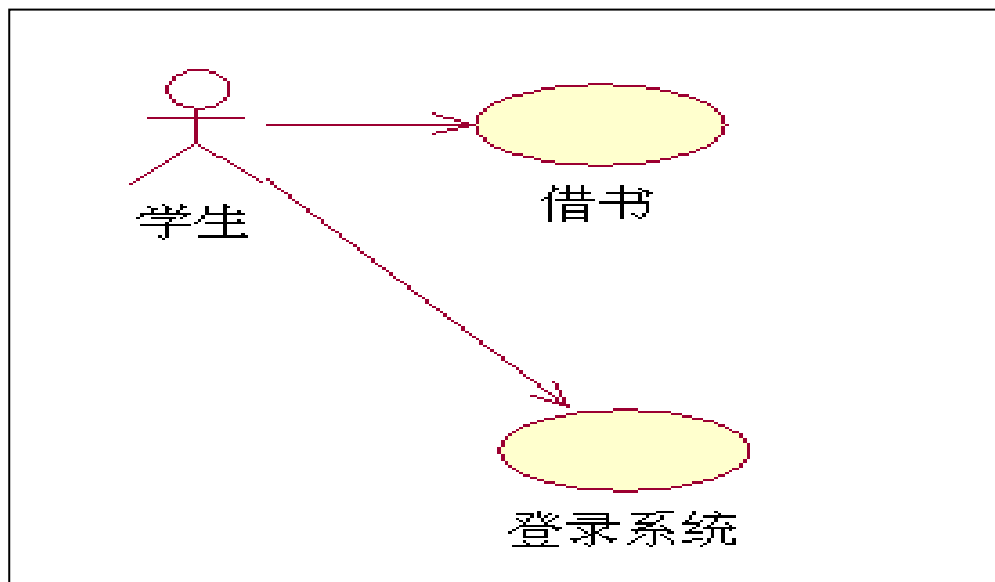
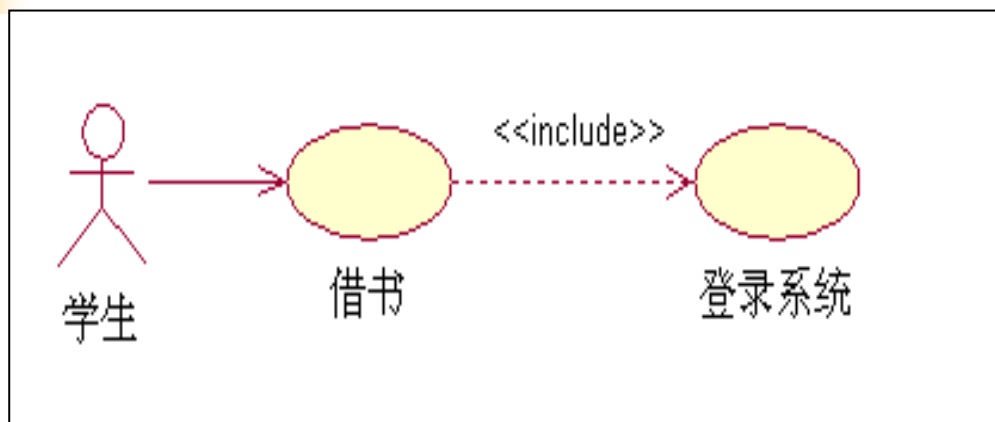
后置条件：系统保存所借图书或者超出所借图书数目,不能借书。

新借书用例

- 
1. 用户输入借书卡号，用例开始。
 2. 系统检测是否是一个合法的借书卡号。
 - a) 如为一非法用户，用例结束。
 3. 系统检测用户记录
 - a) 如有超期图书，返回超期图书的信息。
 - b) 返回借书数目。
 4. 用例结束。

查询用户的借还书记录用例

例



两个用例图的区别

4.3 用例图

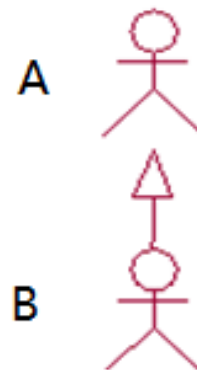
4.3.3 用例建模

■ 高级归档用例技术 (用例间的关系)

❖ Generalization (泛化)

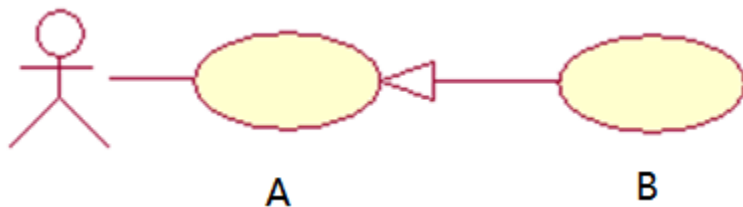
● 执行者与执行者

一个执行者可以完成另一个执行者的同样的任务，它也可能补充额外的角色。

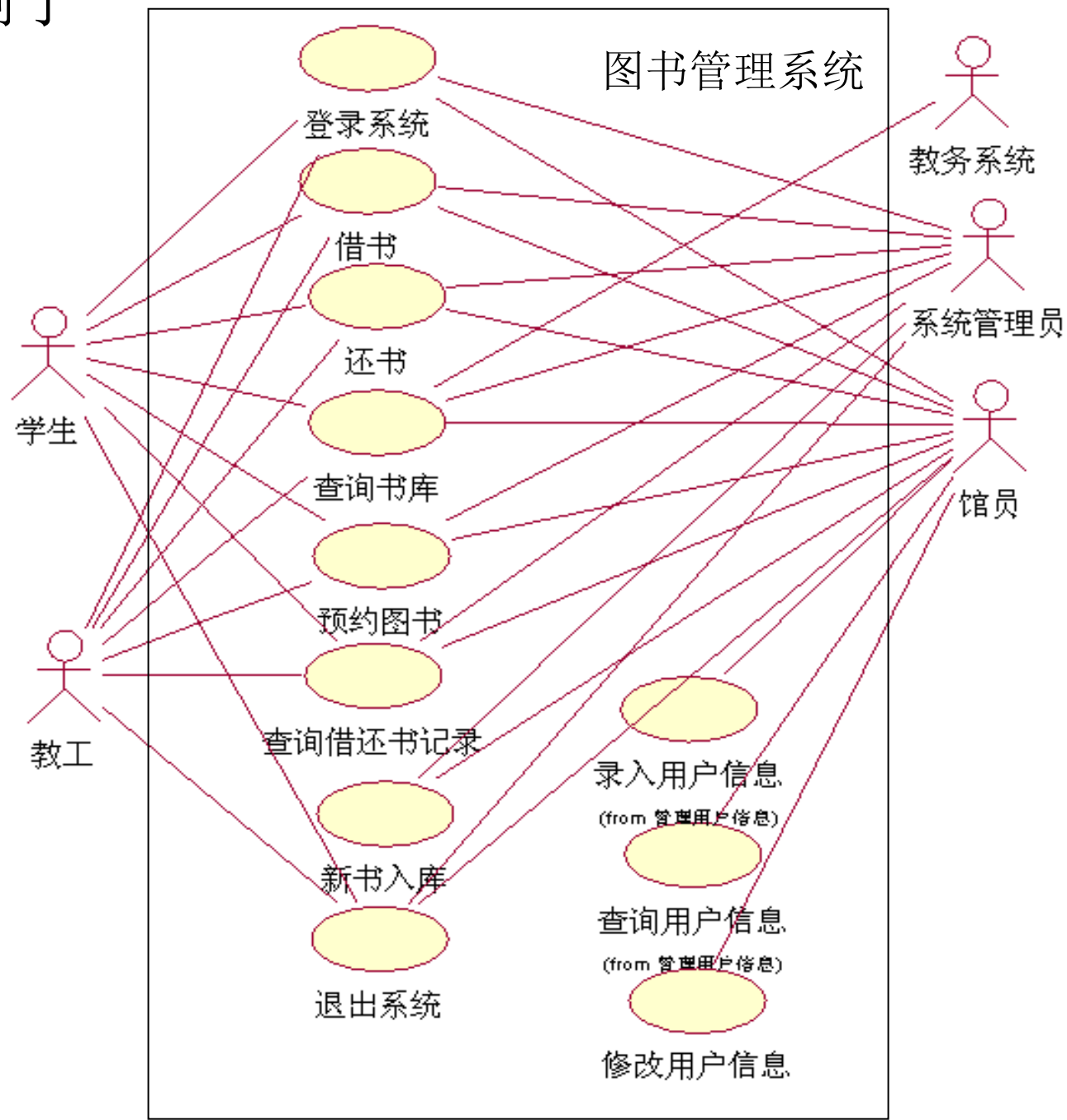


● 用例与用例

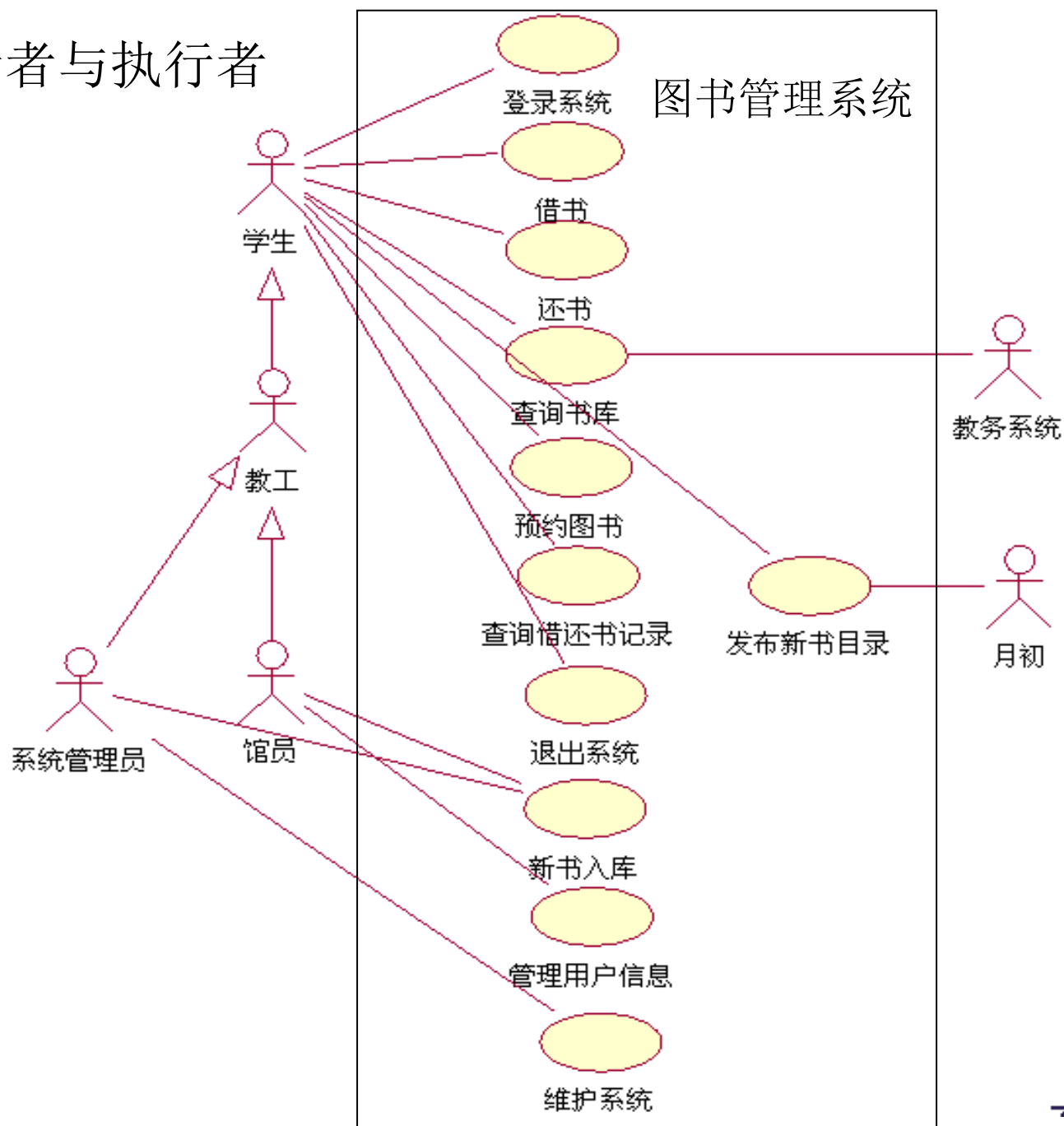
一个用例是另一个用例的特殊版本。



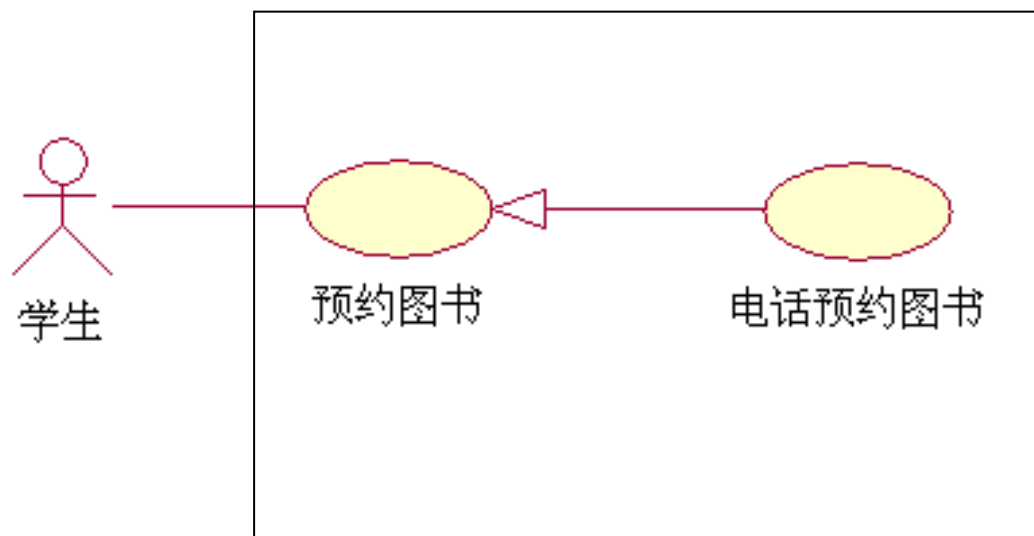
例子



执行者与执行者



用例与用例：





4.3 用例图

4.3.3 用例建模

■ 高级归档用例技术 (用例间的关系)

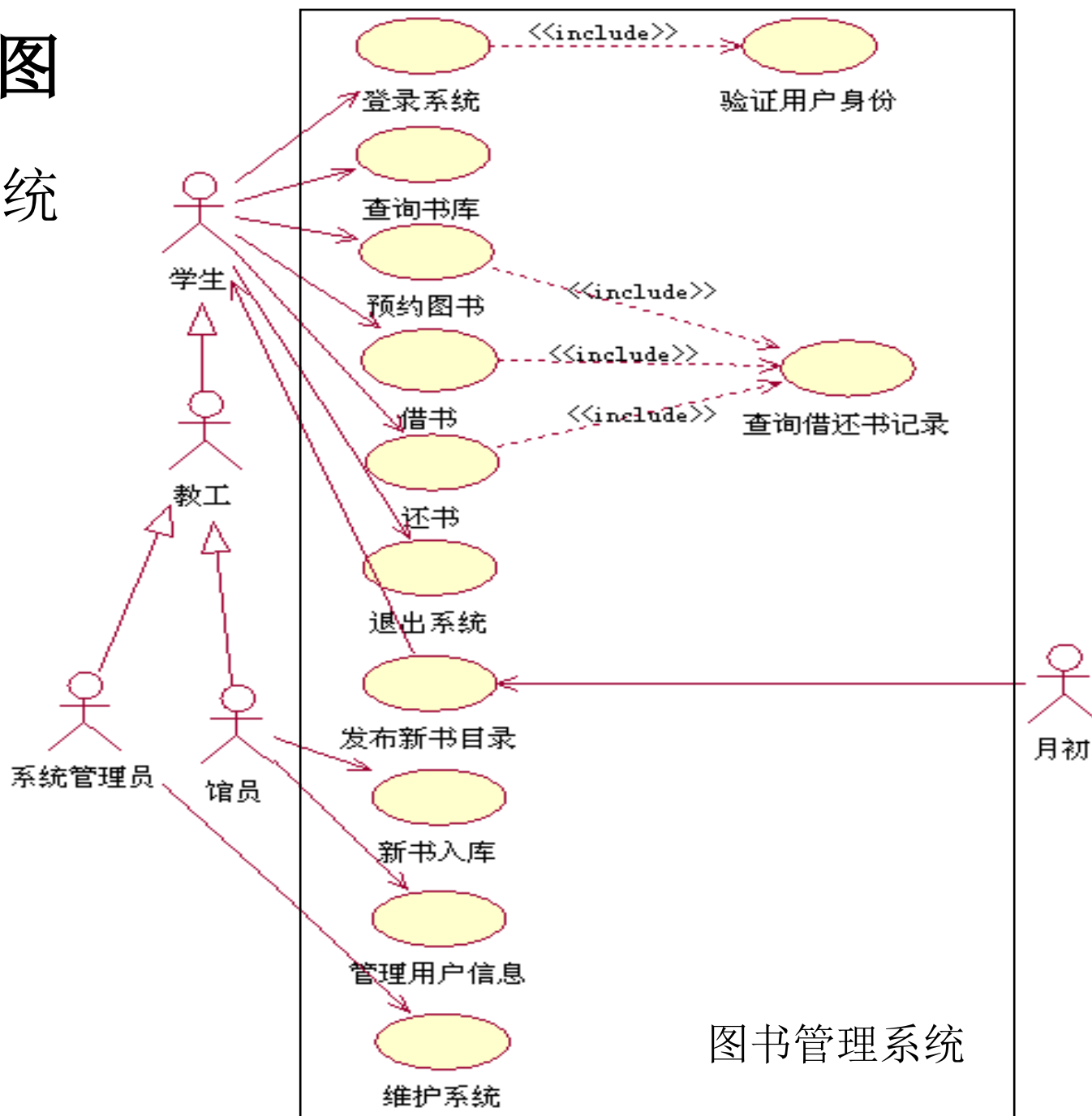
❖ 说明

对于最终用户，用于归档用例的唯一技术是include。

❖ 其他的技术(extend和generalization)只是开发人员所感兴趣的

4.3 用例图

4.3.4 图书管理系统用例图





4.3 用例图

4.3.5 Case:用户权限系统

■ 项目描述

用户权限系统问题陈述

问题陈述

- 我们为任意JavaEE应用软件开发一个通用的用户权限系统，该系统服务于用户、软件需求分析员、权限管理人员。
- 任意JavaEE应用软件可以通过该用户权限系统配置和管理用户的使用权限。

4.3.5 Case:用户权限系统

■ 确定系统边界

(通过确定执行者和用例来确定系统边界)

❖ 确定执行者

- 谁使用这个系统? ←——— 用户、软件需求分析员、权限管理人员
- 谁安装系统? ←——— 系统管理员
- 谁启动系统? ←——— 系统管理员
- 谁维护系统? ←——— 系统管理员
- 谁关闭系统? ←——— 系统管理员
- 哪些其他的系统使用这个系统? ←——— 其他JavaEE应用软件
- 谁从这个系统获取信息? ←——— 用户、权限管理人员
- 谁为这个系统提供信息? ←——— 用户、软件需求分析员
- 是否有事情自动在预计的时间发生? ←——— 无



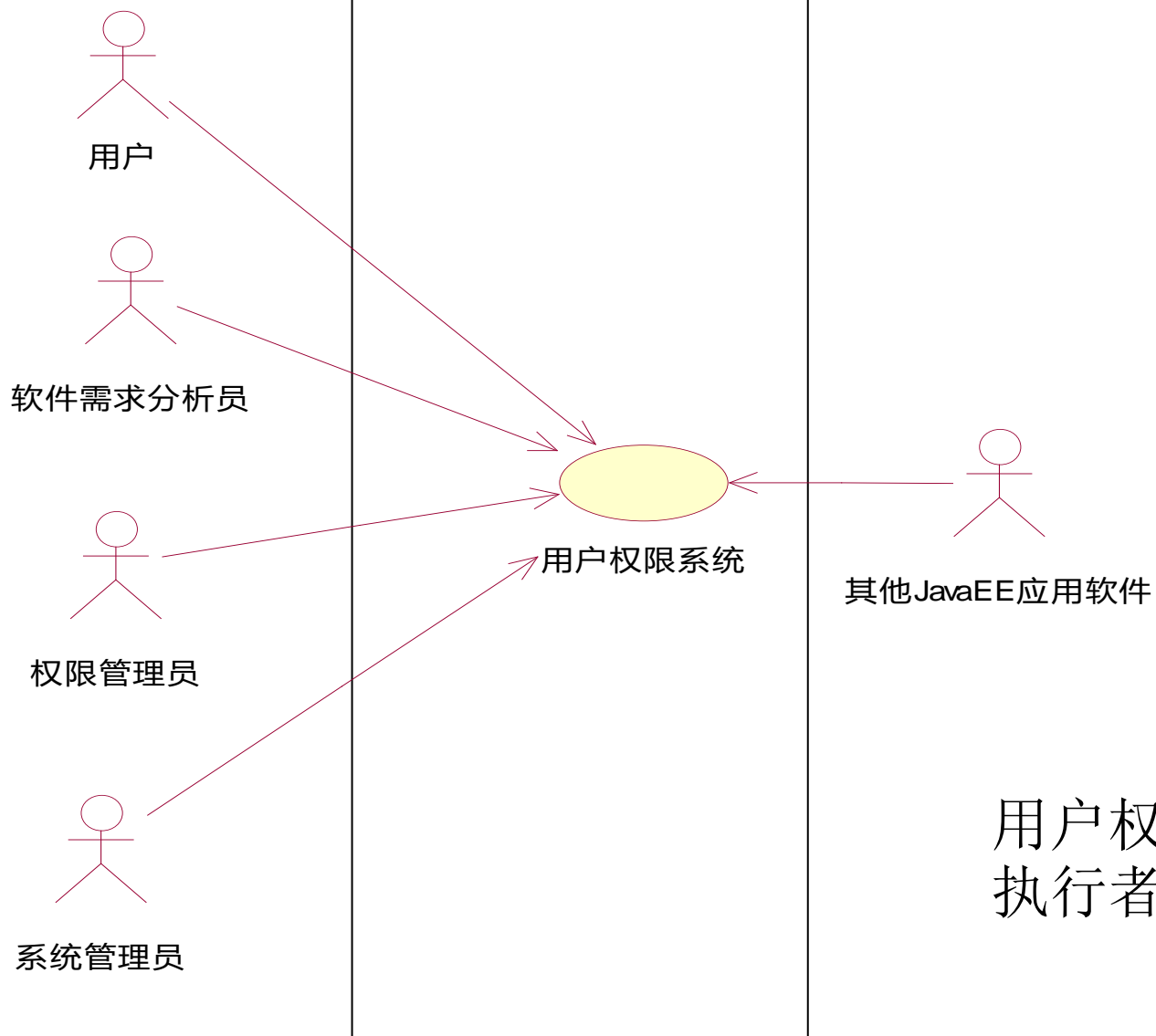
4.3.5 Case:用户权限系统

- 确定系统边界

(通过确定执行者和用例来确定系统边界)

- ❖ 确定执行者

处理时间（某些活动发生在特定的时间）：无



用户权限系统
执行者



4.3.5 Case:用户权限系统

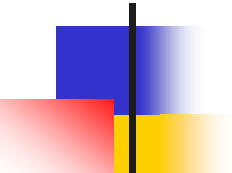
■ 确定系统边界

(通过确定执行者和用例来确定系统边界)

❖ 确定用例

仔细检查所有的执行者，并为每一个执行者确定用例。

- 执行者希望系统提供什么样的功能？
- 系统存储信息吗？
- 执行者将要创建、读取、更新 或删除什么信息？
- 系统是否需要把自身内部状态的变化通知给执行者？
- 系统必须知道哪些外部的事件？ 执行者将怎样通知系统这些事件？
- 其它用例： 启动、关闭、诊断、安装、维护、培训和改变商业过程。



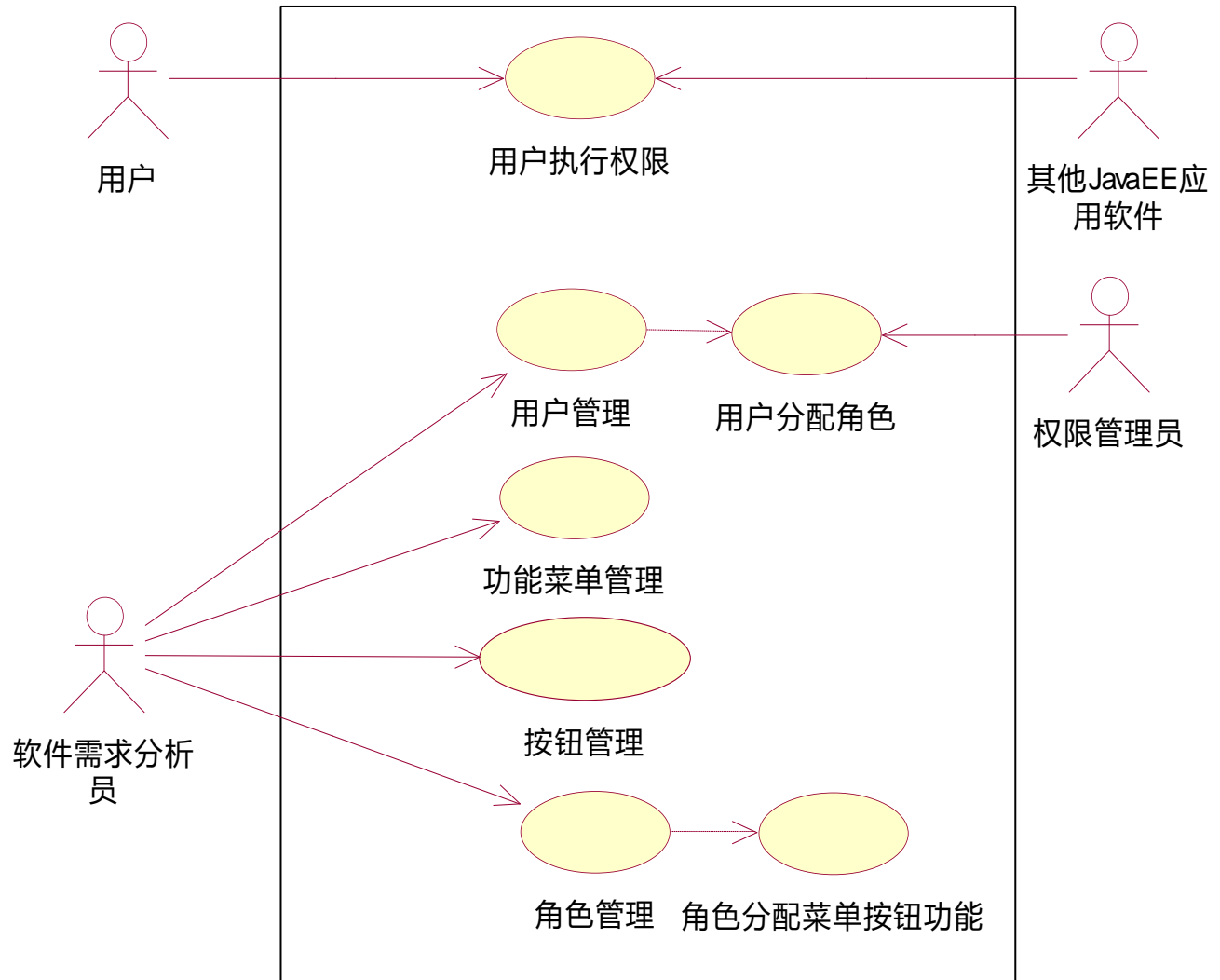
❖ 用户权限系统的业务逻辑

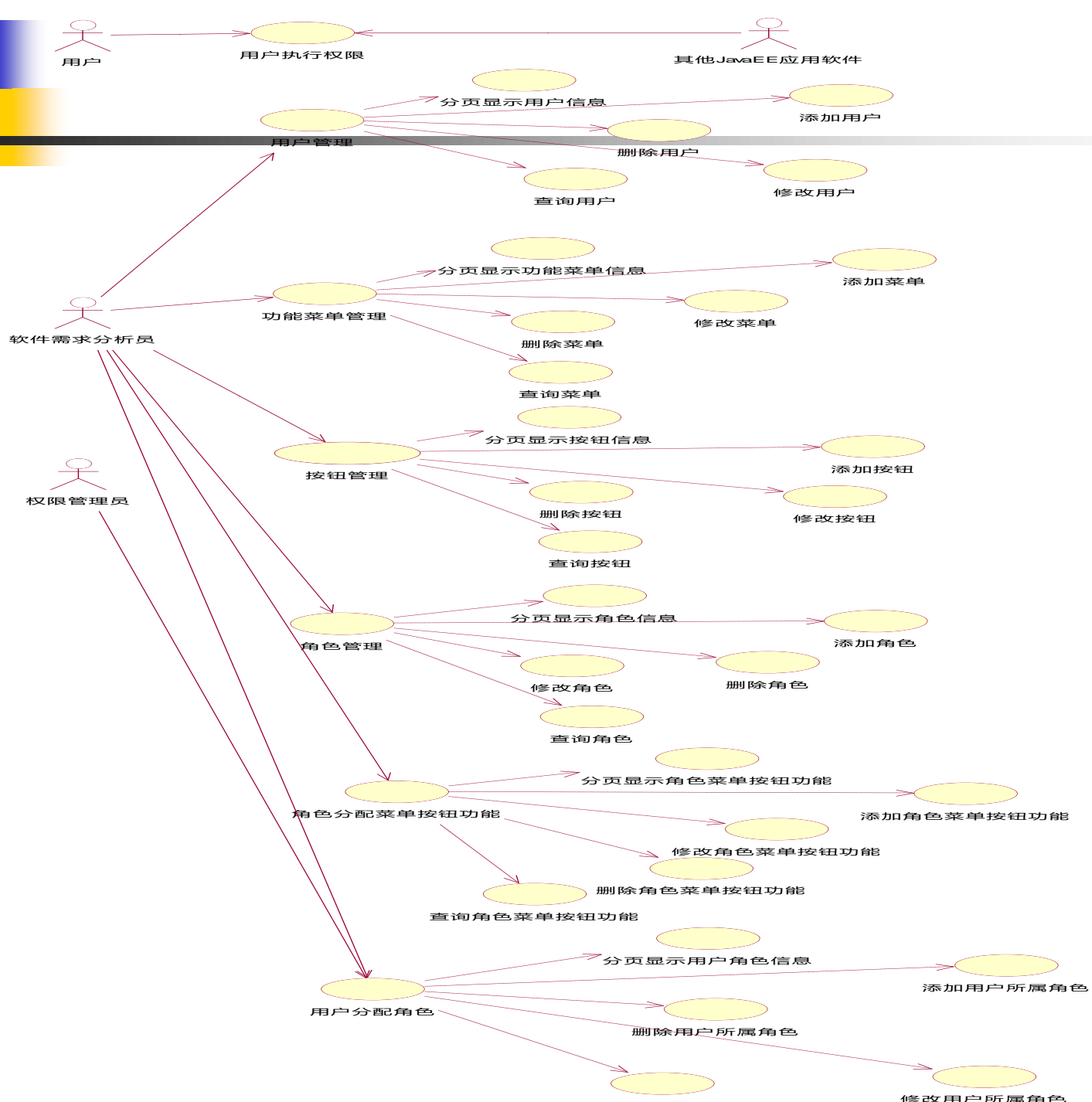
- 1 用户管理（添加、删除、修改、查询）
 - 2 角色管理（添加、删除、修改、查询）
 - 角色是指：用户，管理员等等
 - 3 给用户分配某些角色。
-
- 4 菜单管理（添加、删除、修改、查询）
 - 菜单是指：打开，新建等等菜单
 - 5 操作管理（添加、删除、修改、查询）
 - 操作按钮是指：查询、打印等等按钮
 - 6 给每个菜单分配操作
-
- 7 给每个角色分配某些菜单的某些操作功能
 - 例如：给软件需求分析员分配用户管理菜单对应窗口界面的添加功能。
-
- 8 点击每个按钮时候，查看是否有此权限，如果有则执行，否则弹出提示框：“无此权限”。

4.3.5 Case:用户权限系统

确定系统边界

❖ 用例图



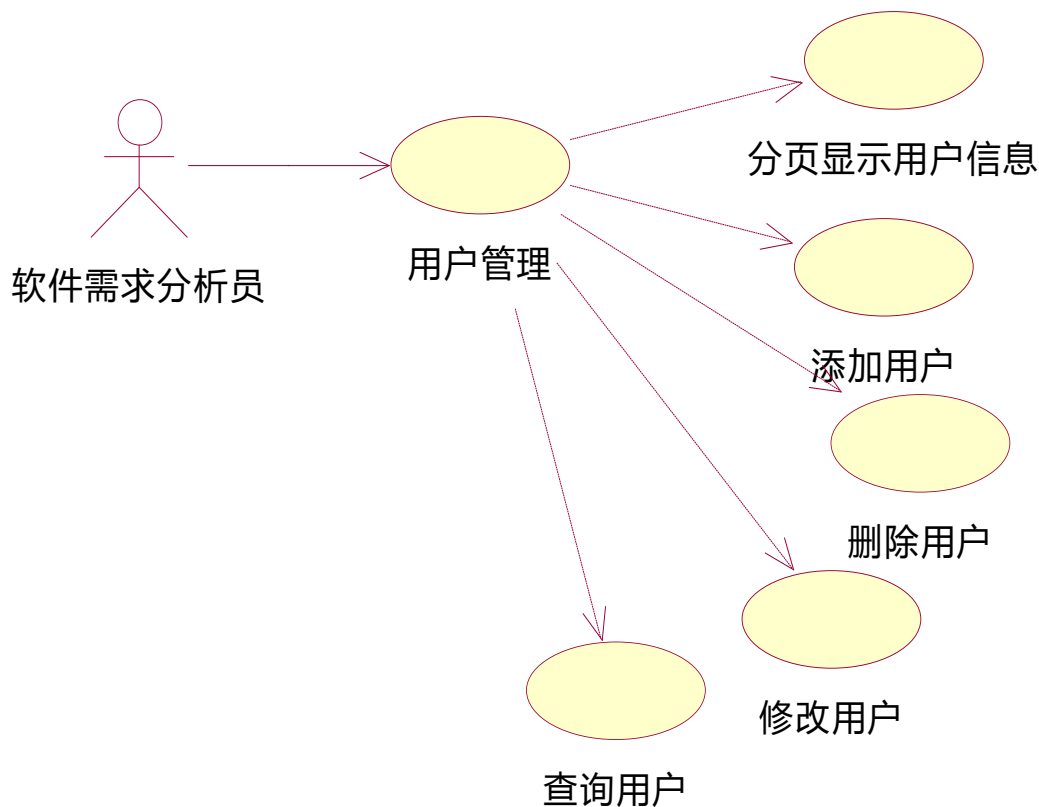


4.3.5 Case:用户权限系统

■ 确定系统边界

❖ 用例图

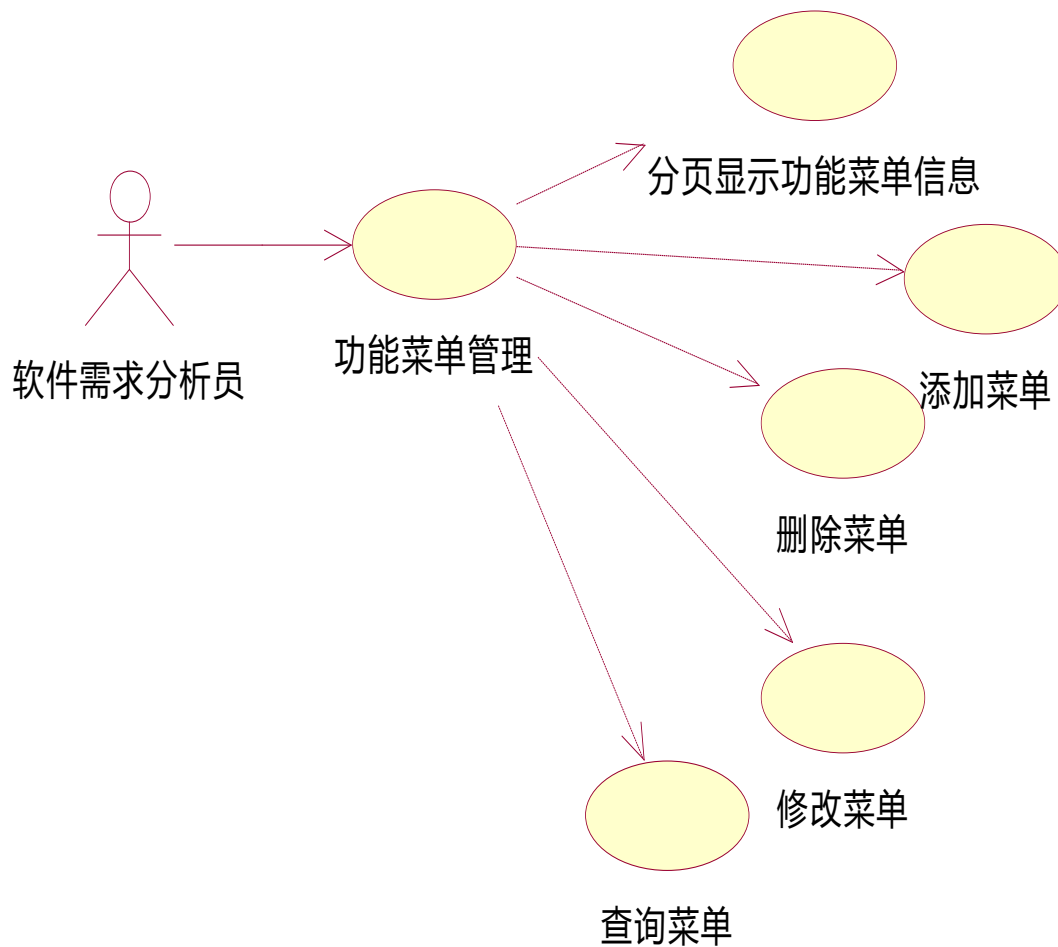
如果用例图过于庞大和杂乱：创建多个用例图



4.3.5 Case:用户权限系统

■ 确定系统边界

❖ 用例图



4.3.5 Case:用户权限系统

■ 系统界面

❖ 主界面



4.3.5 Case:用户权限系统

系统界面

❖ 用户管理主界面

🕒 控制面板

🔑 权限管理 3

👤 用户维护

👤 角色维护

🔒 许可维护

✅ 业务审核 3

🏢 业务管理 7

📋 参数管理

📊 数据列表

查询条件

请输入查询条件

🔍 查询

+ 新增

✖ 删除

#	<input type="checkbox"/>	账号	名称	邮箱地址	操作
1	<input type="checkbox"/>	Lorem	ipsum	dolor	<div>📄✎✖</div>
2	<input type="checkbox"/>	amet	consectetur	adipiscing	<div>📄✎✖</div>
3	<input type="checkbox"/>	Integer	nec	odio	<div>📄✎✖</div>
4	<input type="checkbox"/>	libero	Sed	cursus	<div>📄✎✖</div>
5	<input type="checkbox"/>	dapibus	diam	Sed	<div>📄✎✖</div>
6	<input type="checkbox"/>	Nulla	quis	sem	<div>📄✎✖</div>
7	<input type="checkbox"/>	nibh	elementum	imperdiet	<div>📄✎✖</div>
8	<input type="checkbox"/>	sagittis	ipsum	Praesent	<div>📄✎✖</div>
9	<input type="checkbox"/>	Fusce	nec	tellus	<div>📄✎✖</div>
10	<input type="checkbox"/>	augue	semper	porta	<div>📄✎✖</div>

4.3.5 Case:用户权限系统

系统界面

❖其他**用例主界面

🕒 控制面板

🔑 权限管理 3

👤 用户维护

👤 角色维护

🔒 许可维护

✅ 业务审核 3

🏢 业务管理 7

📄 参数管理

📊 数据列表

查询条件

请输入查询条件

🔍 查询

+ 新增

✖ 删除

#	<input type="checkbox"/>	账号	名称	邮箱地址	操作
1	<input type="checkbox"/>	Lorem	ipsum	dolor	<div>📄✎✖</div>
2	<input type="checkbox"/>	amet	consectetur	adipiscing	<div>📄✎✖</div>
3	<input type="checkbox"/>	Integer	nec	odio	<div>📄✎✖</div>
4	<input type="checkbox"/>	libero	Sed	cursus	<div>📄✎✖</div>
5	<input type="checkbox"/>	dapibus	diam	Sed	<div>📄✎✖</div>
6	<input type="checkbox"/>	Nulla	quis	sem	<div>📄✎✖</div>
7	<input type="checkbox"/>	nibh	elementum	imperdiet	<div>📄✎✖</div>
8	<input type="checkbox"/>	sagittis	ipsum	Praesent	<div>📄✎✖</div>
9	<input type="checkbox"/>	Fusce	nec	tellus	<div>📄✎✖</div>
10	<input type="checkbox"/>	augue	semper	porta	<div>📄✎✖</div>



4.3.5 Case:用户权限系统

■ 描述执行者和用例

用户权限系统执行者描述

❖ 软件需求分析人员

分析用户权限系统的人员，负责分析整个系统的功能菜单、按钮、角色分配、菜单按钮分配、用户管理等.

❖ 用户

用户权限系统的使用者，需要权限系统判断用户的执行权限.

❖ 权限管理员

给用户分配角色



4.3.5 Case:用户权限系统

■ 描述执行者和用例

用户权限系统用例描述

- ❖ 用户管理：管理用户（分页显示、添加、删除、修改、查询）
- ❖ 菜单管理：管理菜单（分页显示、添加、删除、修改、查询）
- ❖ 按钮管理：管理按钮（分页显示、添加、删除、修改、查询）
- ❖ 角色管理：管理角色（分页显示、添加、删除、修改、查询）
- ❖ **角色分配菜单按钮功能**：分配角色功能（分页显示、添加、删除、修改、查询）
- ❖ **用户分配角色**：分配用户角色功能（分页显示、添加、删除、修改、查询）
- ❖ **用户执行权限**：判断用户的执行权限。



4.3.5 Case:用户权限系统

■ 归档用例

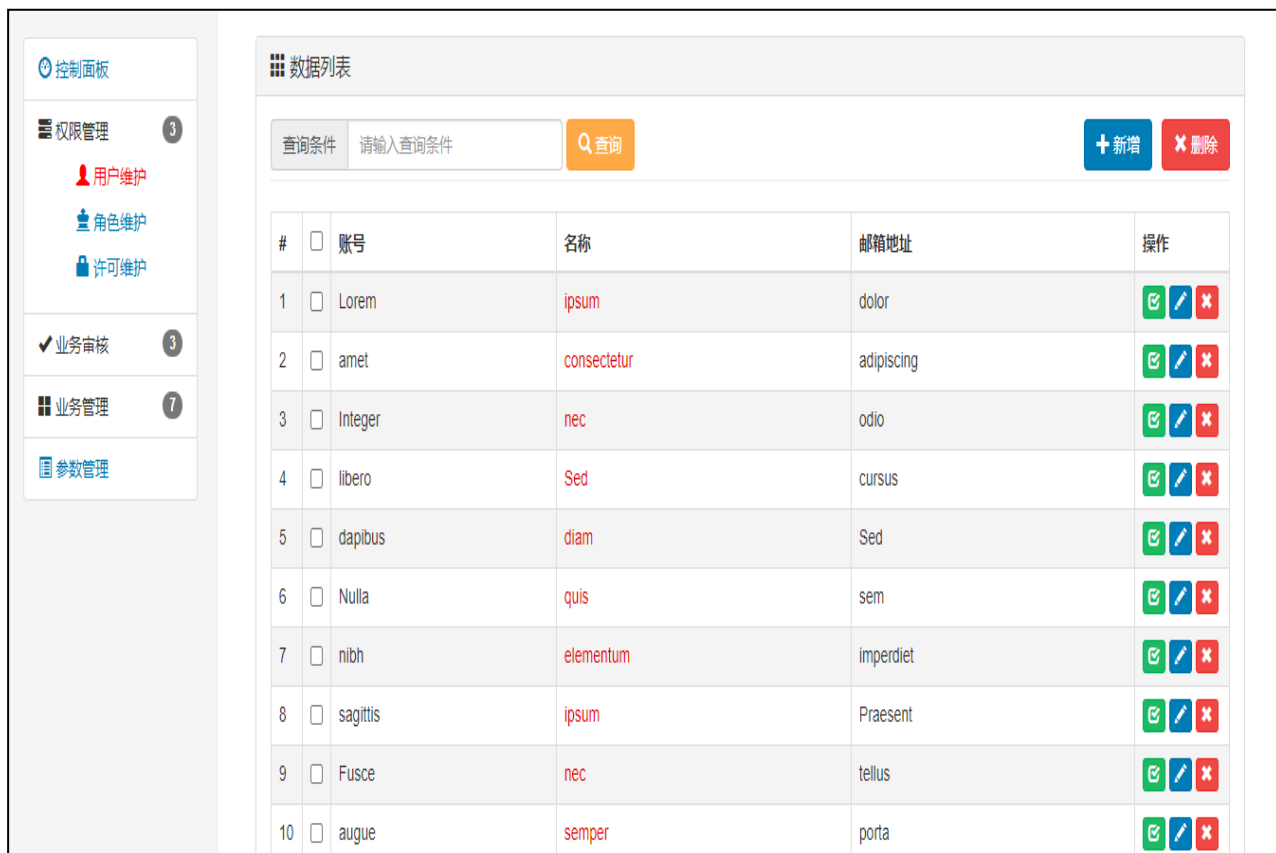
- ❖ 基本功能、
- ❖ 所有可选方案、
- ❖ 异常情况、
- ❖ 进入用例之前必须正确的一切、
- ❖ 退出用例后必须正确的一切。





















4.3.5 Case:用户权限系统

■ 归档用例

❖ 1 添加用户用例

❖ 用户管理主界面



#	<input type="checkbox"/>	账号	名称	邮箱地址	操作
1	<input type="checkbox"/>	Lorem	ipsum	dolor	 
2	<input type="checkbox"/>	amet	consectetur	adipiscing	 
3	<input type="checkbox"/>	Integer	nec	odio	 
4	<input type="checkbox"/>	libero	Sed	cursus	 
5	<input type="checkbox"/>	dapibus	diam	Sed	 
6	<input type="checkbox"/>	Nulla	quis	sem	 
7	<input type="checkbox"/>	nibh	elementum	imperdiet	 
8	<input type="checkbox"/>	sagittis	ipsum	Praesent	 
9	<input type="checkbox"/>	Fusce	nec	tellus	 
10	<input type="checkbox"/>	augue	semper	porta	 

4.3.5 Case:用户权限系统

■ 归档用例

❖ 1 添加用户用例

● 添加用户界面



首页 / 数据列表 / 新增

表单数据

登陆账号

用户名称

邮箱地址

请输入合法的邮箱地址, 格式为: xxxx@xxxx.com

+ 新增 重置



4.3.5 Case:用户权限系统

■ 归档用例

❖ 1 添加用户用例

● 文本

前置条件：一个合法的需求分析人员已经登录系统
事件流：

1. 当需求分析人员在用户维护主界面选择新增按钮时,用例开始。
2. 系统显示新增用户界面。
3. 需求分析人员输入该用户的用户名、邮箱
4. 点击保存按钮。

后置条件：系统保存新增的用户（密码默认888888）。



4.3.5 Case:用户权限系统

■ 归档用例

❖ 2 **用例



thanks