**图书管理系统UML图**

案例：图书管理系统

一、图书管理系统功能描述

图书管理系统能够对图书进行注册登记，也就是将图书的基本信息（如编号、书名、价格、作者等）预先存入数据库中，供以后检索，并且能够对借阅人进行注册登记，包括记录借阅人姓名、编号、班级、年龄、性别、地址、电话等信息。同时，图书管理系统提供方便的查询方法。如以书名、作者、出版社、出版时间等信息进行图书检索，并能反映出图书的借阅情况：以借阅人编号对借阅人信息进行检索，以出版社名称查询出版社联系方式等信息。图书管理系统提供对书籍进行预订的功能，也提供旧书销毁的功能，对于淘汰、损坏、丢失的书名可及时对数据库进行修改。图书管理系统能够对使用该管理系统的用户进行管理，按照不同的工作职能提供不同的功能授权。总的来说，图书管理系统主要包含下列功能。

1. 读者管理：读者信息的制定、输入、修改、查询、包括种类、性别、借书数量、借书期限、备注等
2. 书籍管理：书籍基本信息的限定、输入、修改、查询、包括书籍编号、类别、备注、关键词、备注
3. 借阅管理：包括借书、还书、预订书籍、续借、查询书籍、过期处理和书籍丢失后的处理

**图书管理系统UML各种图**

1. 用例图

主要用来描述“用户，需求，系统功能单元之间的关系”它展示了一个外部用户能够观察到的系统功能模型

用途：帮助开发团队以一种可视化的方式理解系统的功能需求。

用例图所包含的元素如下：actor,usecase,子系统，四种关系

确定参与者：读者、管理员

确定用例：

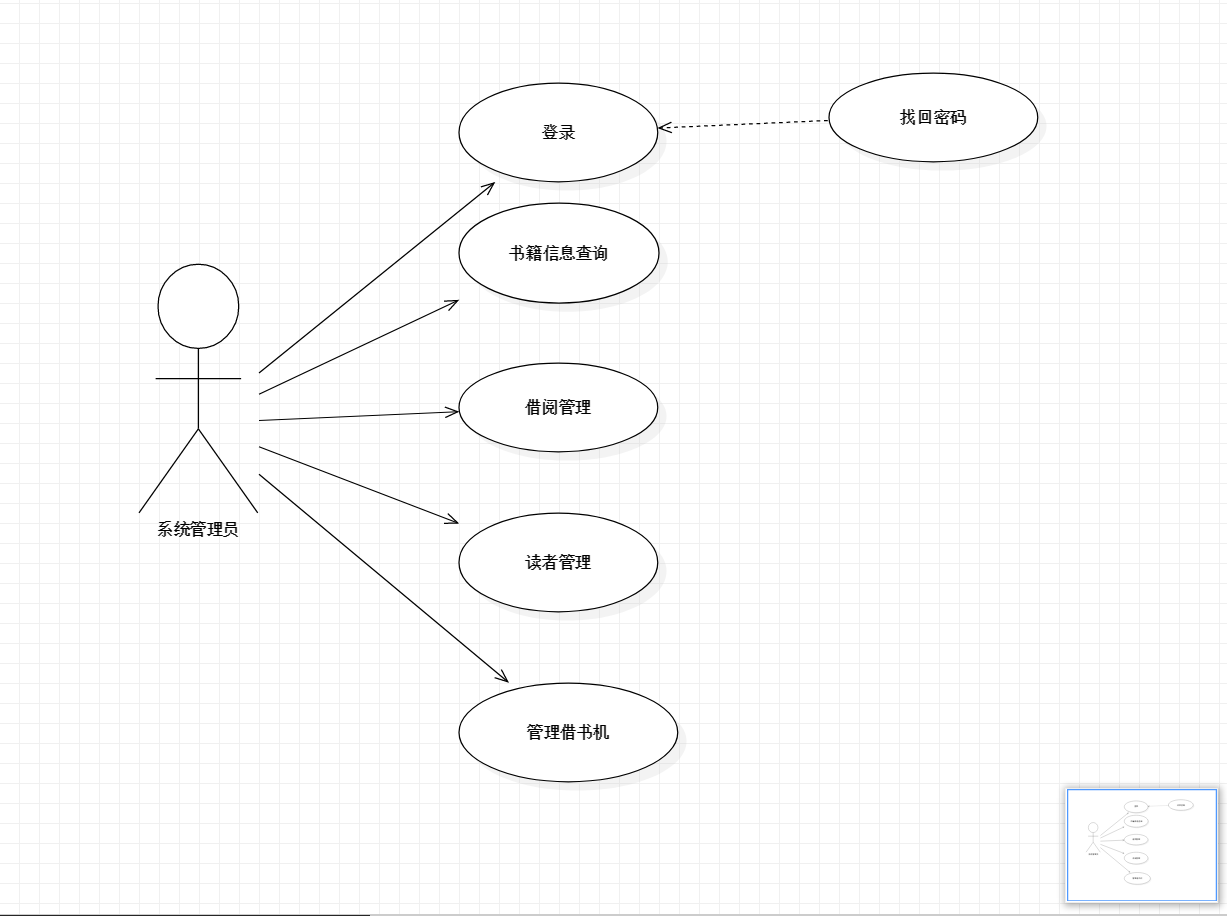
管理员包括的用例：

1. 登录系统：管理员可以通过登录该系统进行各项操作。
2. 书籍管理：包括对书籍的增删改查操作
3. 书籍借阅管理：包括借书、还书、预订、书籍逾期处理和书籍丢失处理
4. 读者管理：包括对读者的增删改查操作

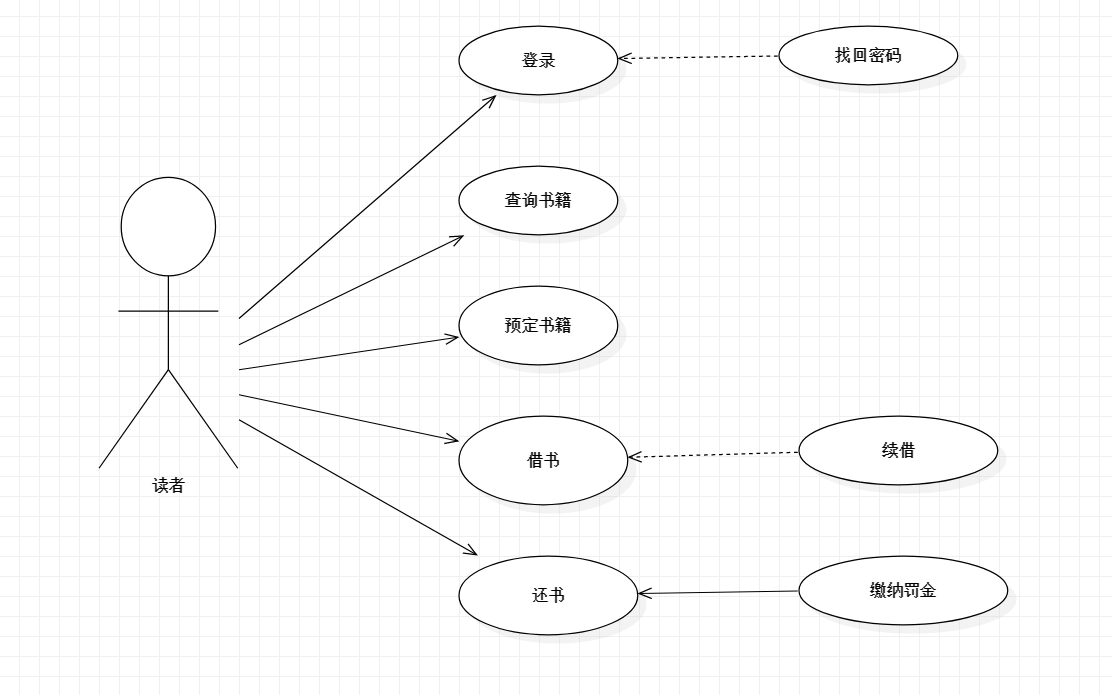
读者包括的用例

1. 登录系统
2. 借书
3. 还书
4. 查询：包括对个人信息和书籍信息的查询业务
5. 预订：读者对书籍业务的预订业务
6. 逾期处理：书籍过期缴纳罚金等
7. 书籍丢失处理：对书籍丢失后的不同措施进行处理
8. 自动借书机的使用

以下是图书馆系统中管理员用例图：



借阅者用例图



1. 类图

在UML类图中，类使用包含类名、属性和方法，且带有分割线的矩形来表示UML类图中表示可见性的符号有三种：

“＋”：表示public

“—”：表示private

“#”：表示protected(friendly也归入这类)

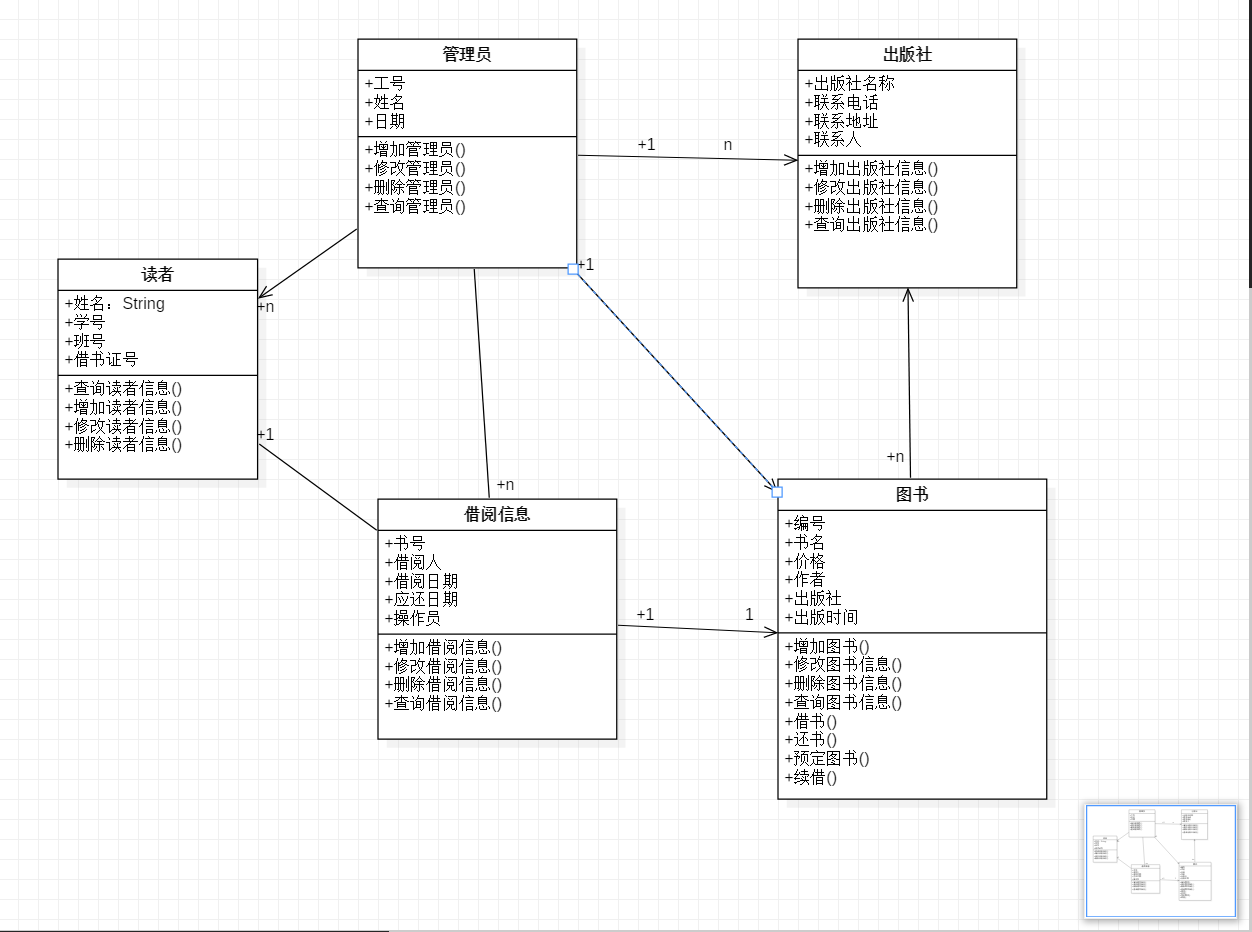
类之间的关系是类图中比较复杂的内容。有关联、聚合、组合、泛化、依赖

关联：是模型元素之间的一种语义联系，是类之间的一种很弱的联系。关联可以有方向，可以是单向关联，也可以是双向关联。可以给关联加上关联名来描述关联的作用。关联两端的类也可以以某种角色参与关联，角色可以具有多重性，表示可以有多少个对象参与关联。可以通过关联类进一步描述关联的属性、操作以及其他信息。关联类通过一条虚线与关联连接。对于关联可以加上一些约束，以加强关联的含义

聚合是一种特殊的关联，聚合表示整体与部分的关系。通常在定义一个整体类后，再去分析这个整体类的组成结构。从而找出一些组成类，该整体类和组成类之间就形成了聚合关系。例如舰队是由一系列的舰船组成。需求描述中“包含”、“组成”、“分为….部分”等词常意味着聚合关系。组合也是一种特殊的关联，也表示类之间整体和部分的关系，但是组合关系中部分和整体具有统一的生存期。一旦整体对象不存在，部分对象也将不存在。部分对象与整体对象之间具有共生死的关系。

泛化定义了一般元素和特殊元素之间的分类关系，类之间的这种泛化关系也就是继承关系。泛化关系是“a-kind-of”关系，定义了一般元素和特殊元素之间的分类关系

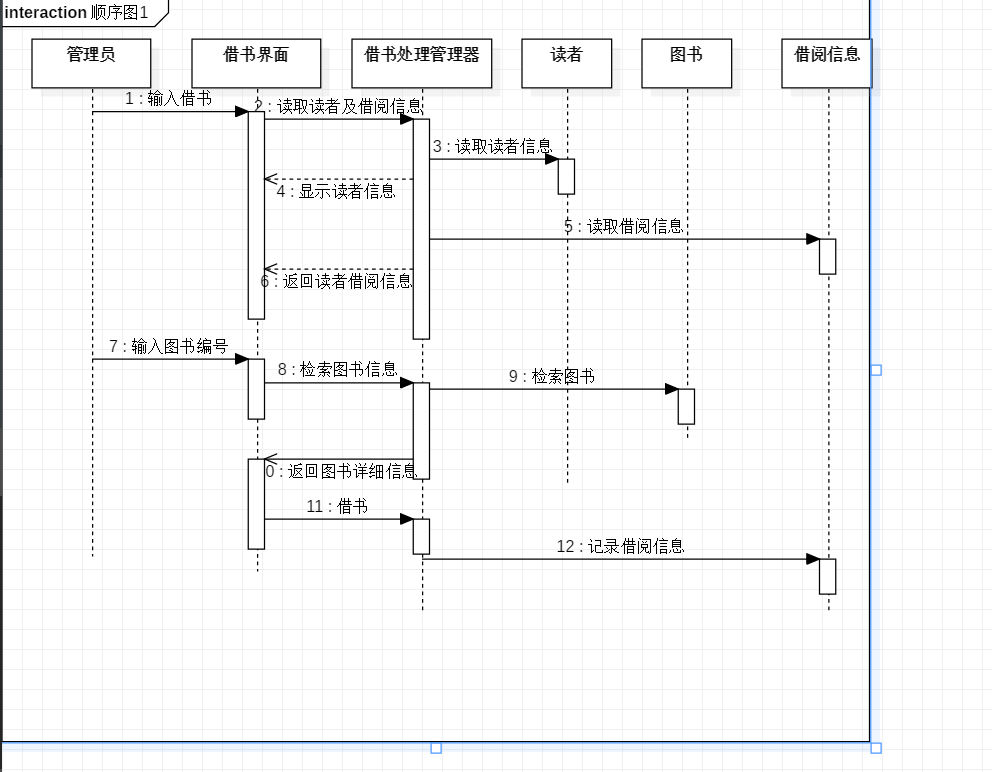
下图为图书馆管理系统的类图



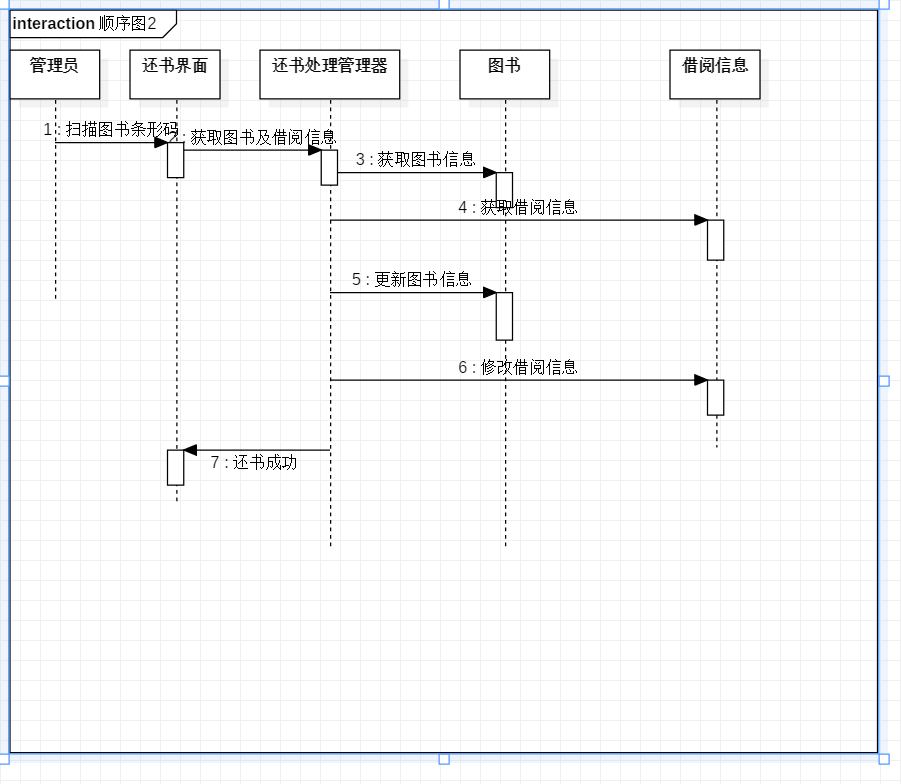
1. 顺序图

UML顺序图一般用于：确认和丰富一个使用情境的逻辑。一个使用情境就是系统潜在的使用方式的描述，也就是它的名称所要描述的。一个使用情境的逻辑可能是用例的一部分，或是一条备选路线；一个贯穿单个用例的完整流程，动作基本过程的逻辑描述，或是动作的基本过程的一部分再加上一个或多个的备用情境的逻辑描述

下图为借阅图书顺序图



还书顺序图



1. 状态图

状态图主要用于描述一个对象在其生存期间的动态行为，表现为一个对象所经历的状态序列，引起状态转移的事件，以及因状态转移而伴随的动作。一般可以用状态机对一个对象的生命周期建模，状态图用于显示状态机，重点在与描述状态图的控制流。如下图例子，状态机描述了门对象的生存期间的状态序列，引起转移的事件，以及因状态转移而伴随的动作.

状态：指在对象的生命周期中的某个条件或者状况，在此期间对象将满足某些条件、执行某些活动活活等待某些事件。所有对象都有状态，状态是对象执行了一系列活动的结果，当某个事件发生后，对象的状态将发生变化。

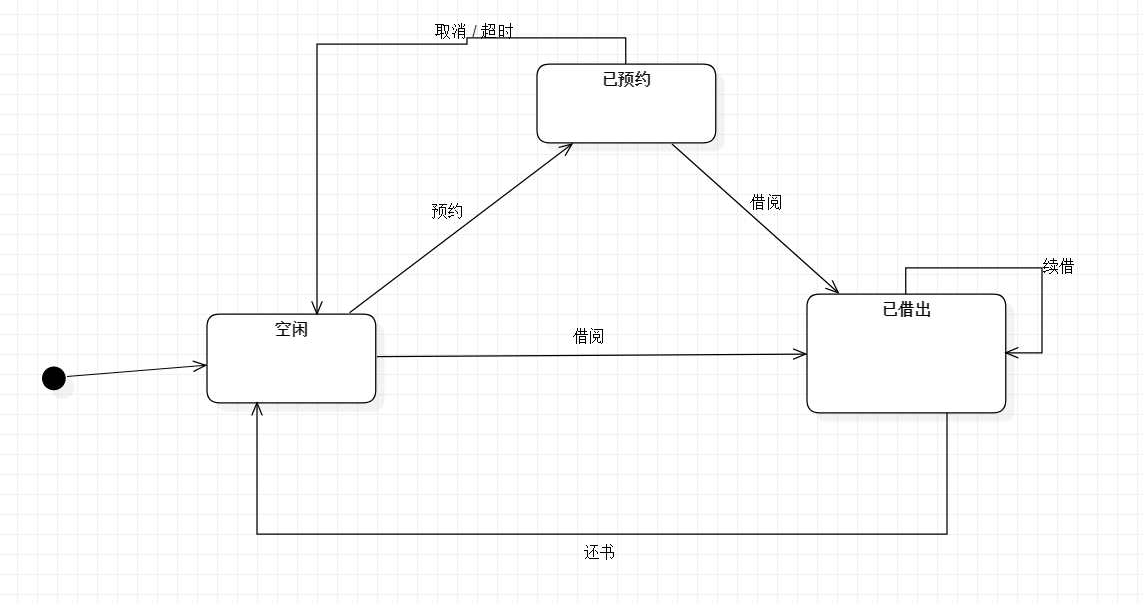
转移：是两个状态之间的一种关系，表示对象将在源状态（Source State）中执行一定的动作，并在某个特定事件发生而且某个特定的警界条件满足时进入目标状态

动作：是一个可执行的原子操作,也就是说动作是不可中断的，其执行时间是可忽略不计的。

自身转移：状态可以有返回自身状态的转移，称之为自身转移

组合状态：嵌套在另外一个状态中的状态称之为子状态,一个含有子状态的状态被称作组合状态

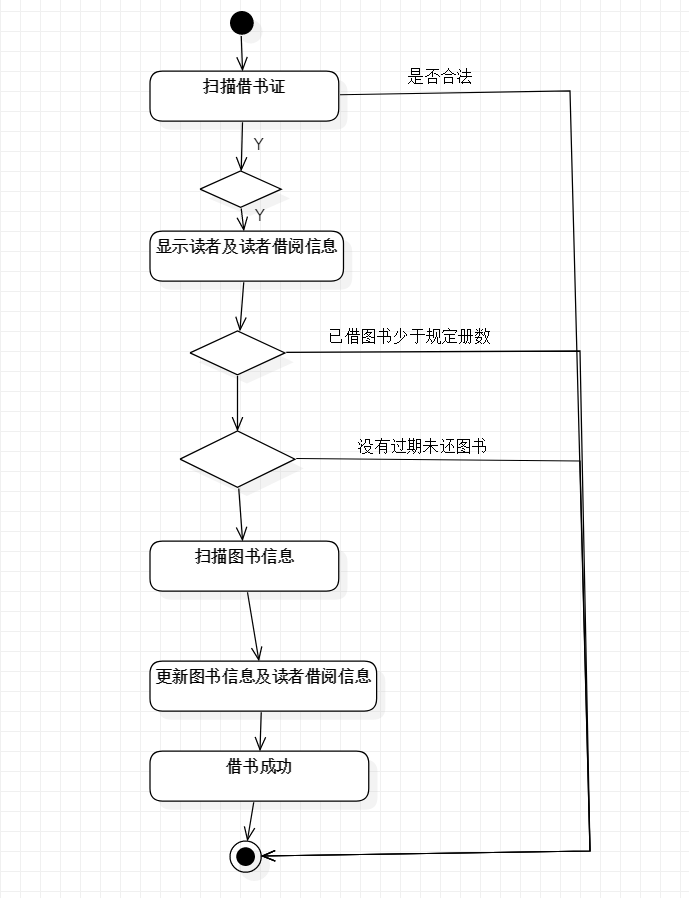
下图为图书管理系统状态图：



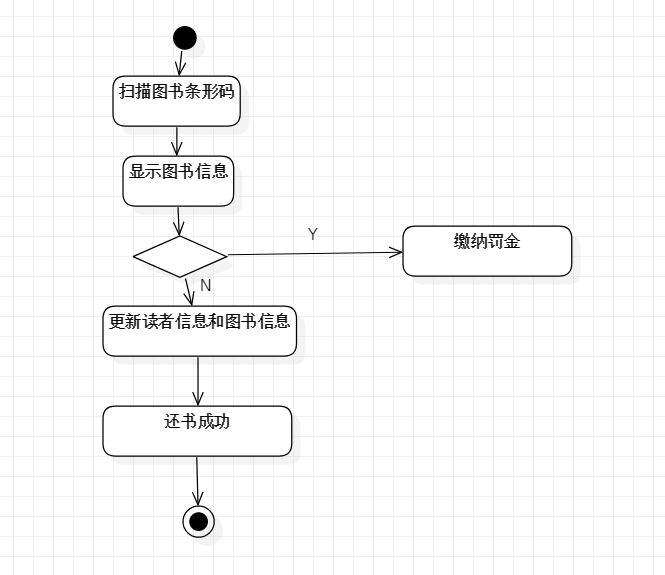
1. 活动图

活动图是UML用于对系统的动态行为建模的另一种常用工具，它描述活动的顺序，展现从一个活动到另一个活动的控制流。活动图在本质上是一种流程图。活动图着重表现从一个活动到另一个活动的控制流，是内部处理驱动的流程

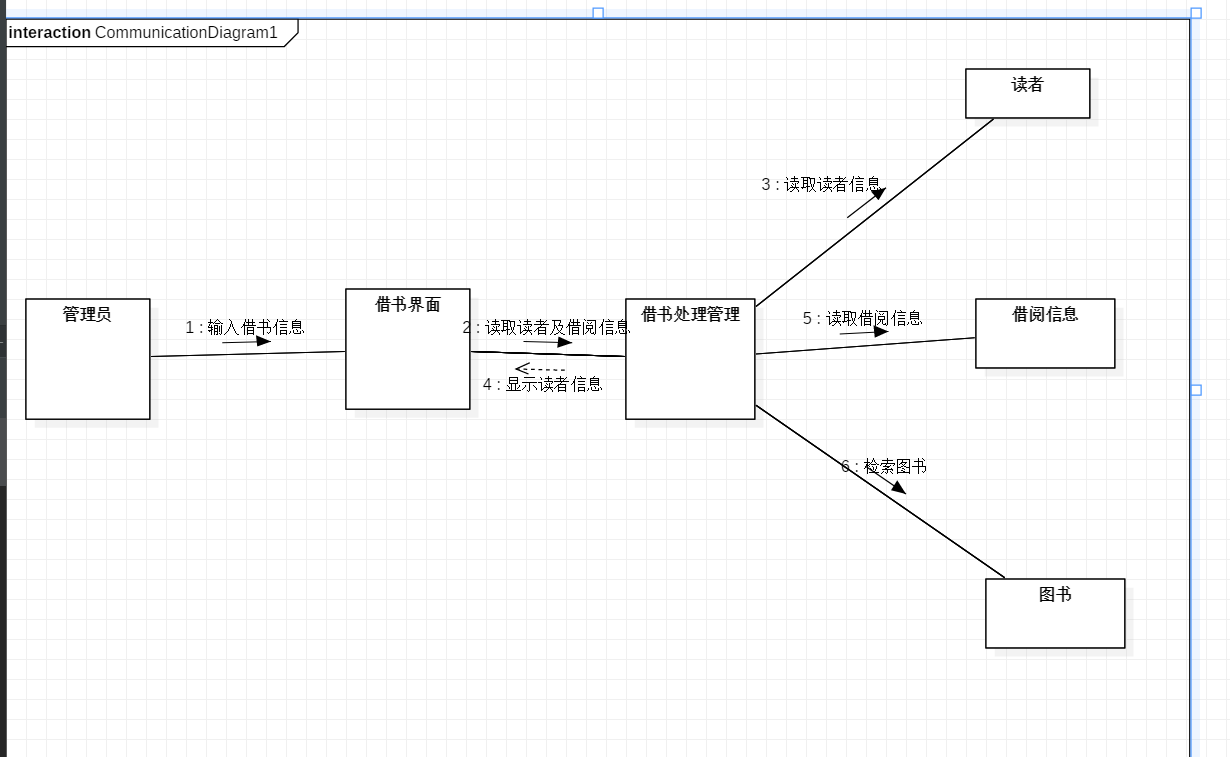
下图为借书人活动图



还书活动图



1. 协作图



七、部署图

部署图描述的是系统运行时的结构，展示了硬件的配置及其软件如何部署到网络结构中。一个系统模型只有一个部署图，部署图通常用来帮助理解分布式系统

