

一、需求分析

- (1)**读卡**：用户将银行卡插入读卡器，读卡器识别银行卡的真伪，并在屏幕上提示输入密码；
- (2)**输入密码**：用户通过键盘输入密码，系统验证密码是否有效，如果有效则提示客户进行选择要操作的义务
- (3)**选择操作**：用户根据自己的需要可以进行存款、转账、取款修改密码、查询账户等选择操作
- (4)**系统交互**：在用户选择后显示器进行交互提示和操作确认消息等
- (5)**退卡**：操作完成后用户退卡
- (6)**运维**：维护人员对ATM机进行加钞、加凭条纸、维护、维修等工作
- (7)**用户管理**：数据管理员管理用户信息

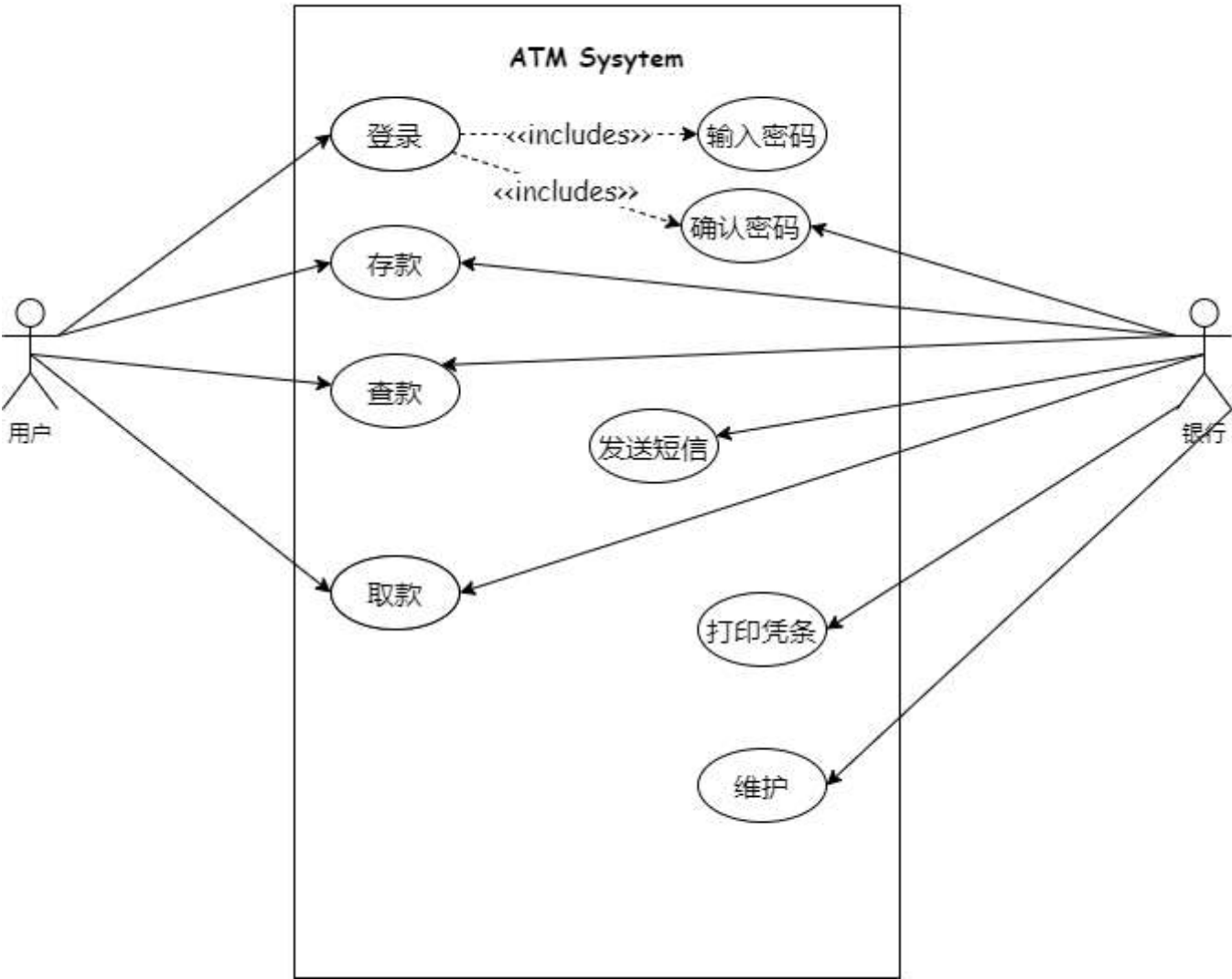
二、建模过程

1.用例图

用例图——用例图是指由参与者（Actor）、用例（Use Case）以及它们之间的关系构成的用于描述系统功能的静态视图。用例图是被称为参与者的外部用户所能观察到的系统功能的模型图，呈现了一些参与者和一些用例，以及它们之间的关系，主要用于对系统、子系统或类的功能行为进行。

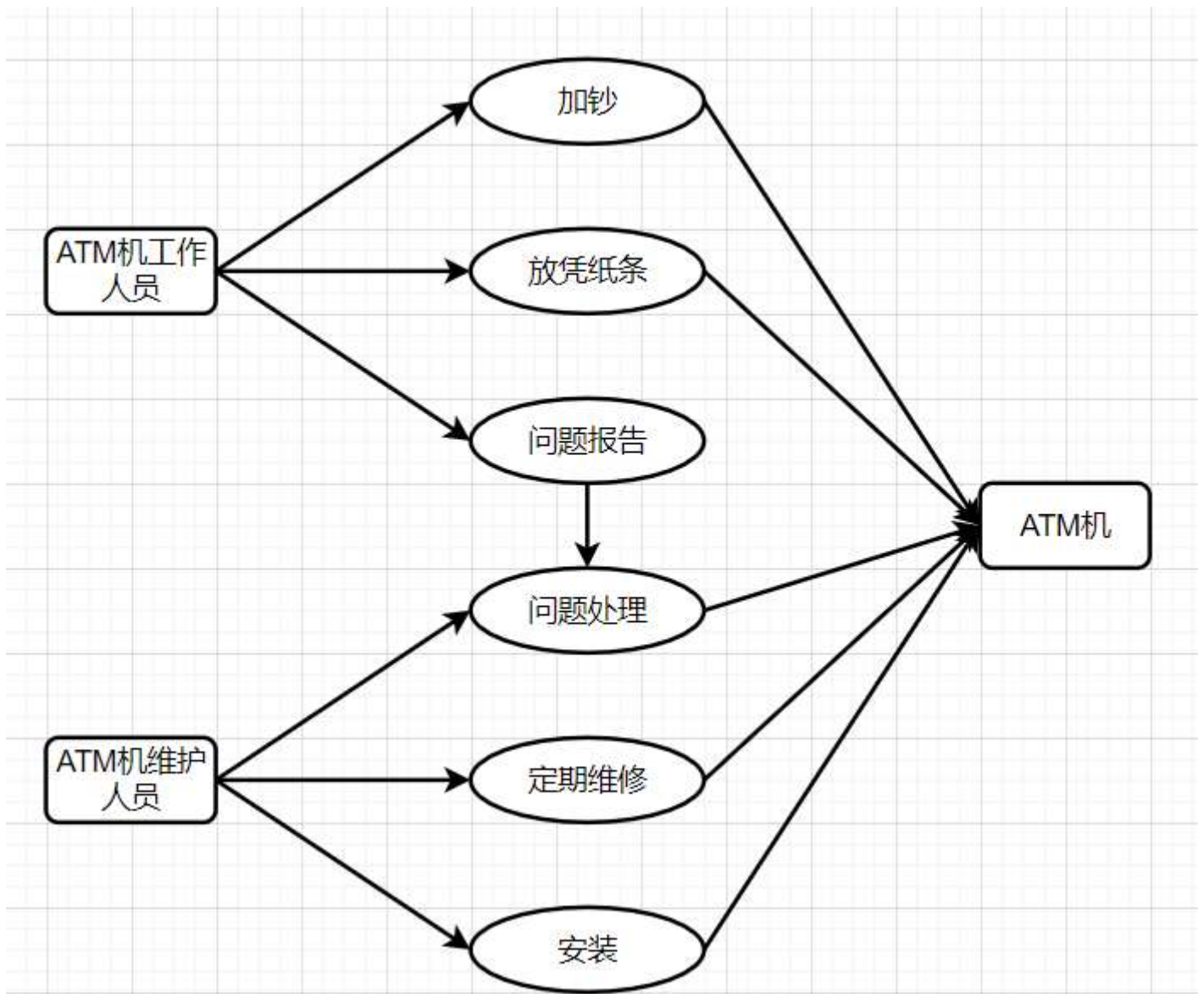
- (1) 用户用例模型：用户在ATM机上可以进行存款、转账、取款修改密码、查询账户等选择操作
- 客户用例图

用例图use case diagram



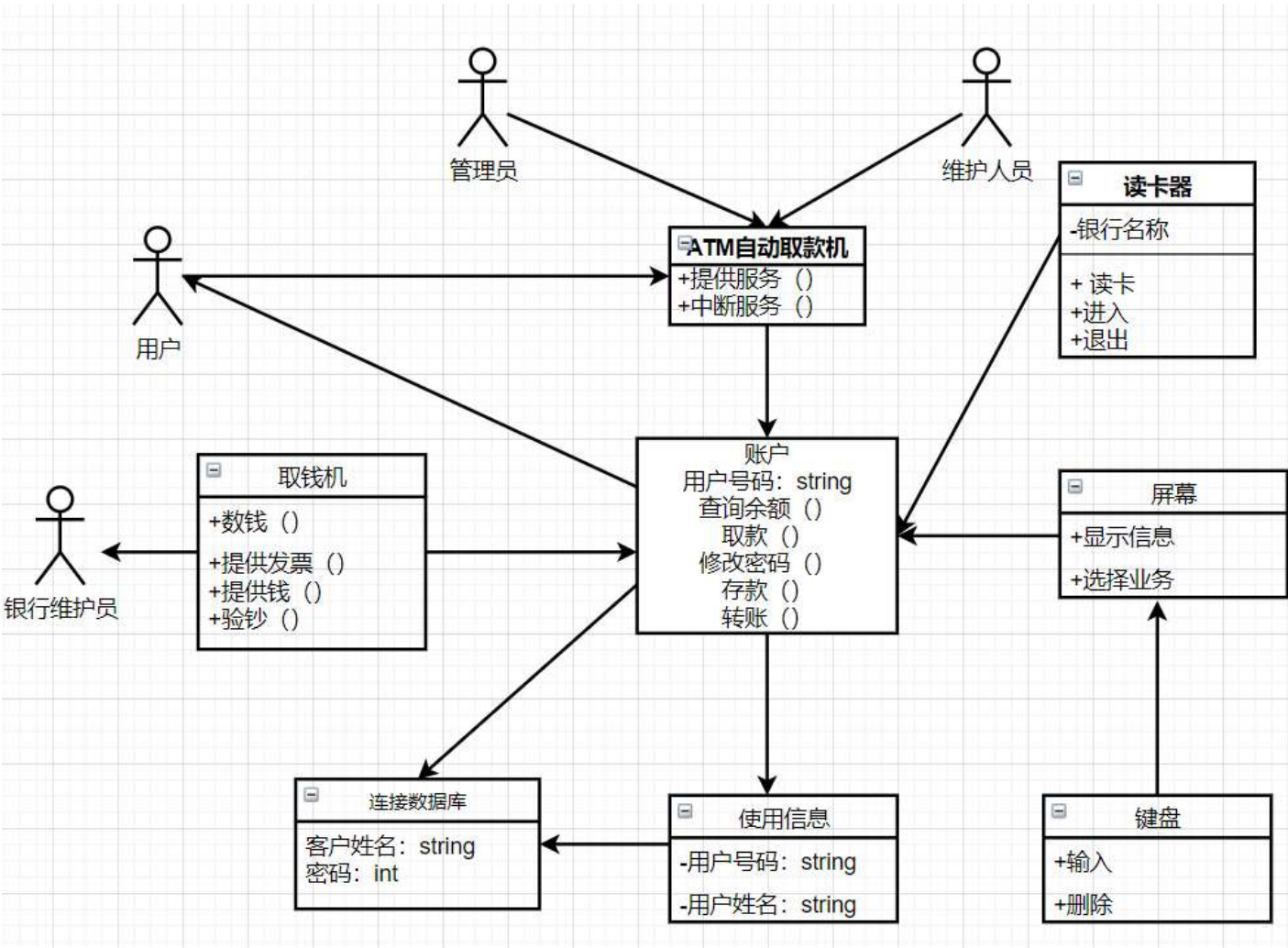
(2) 维护人员用例模型：维护人员对ATM机进行加钞、加凭条纸、维护、维修等工作

维护人员用例图

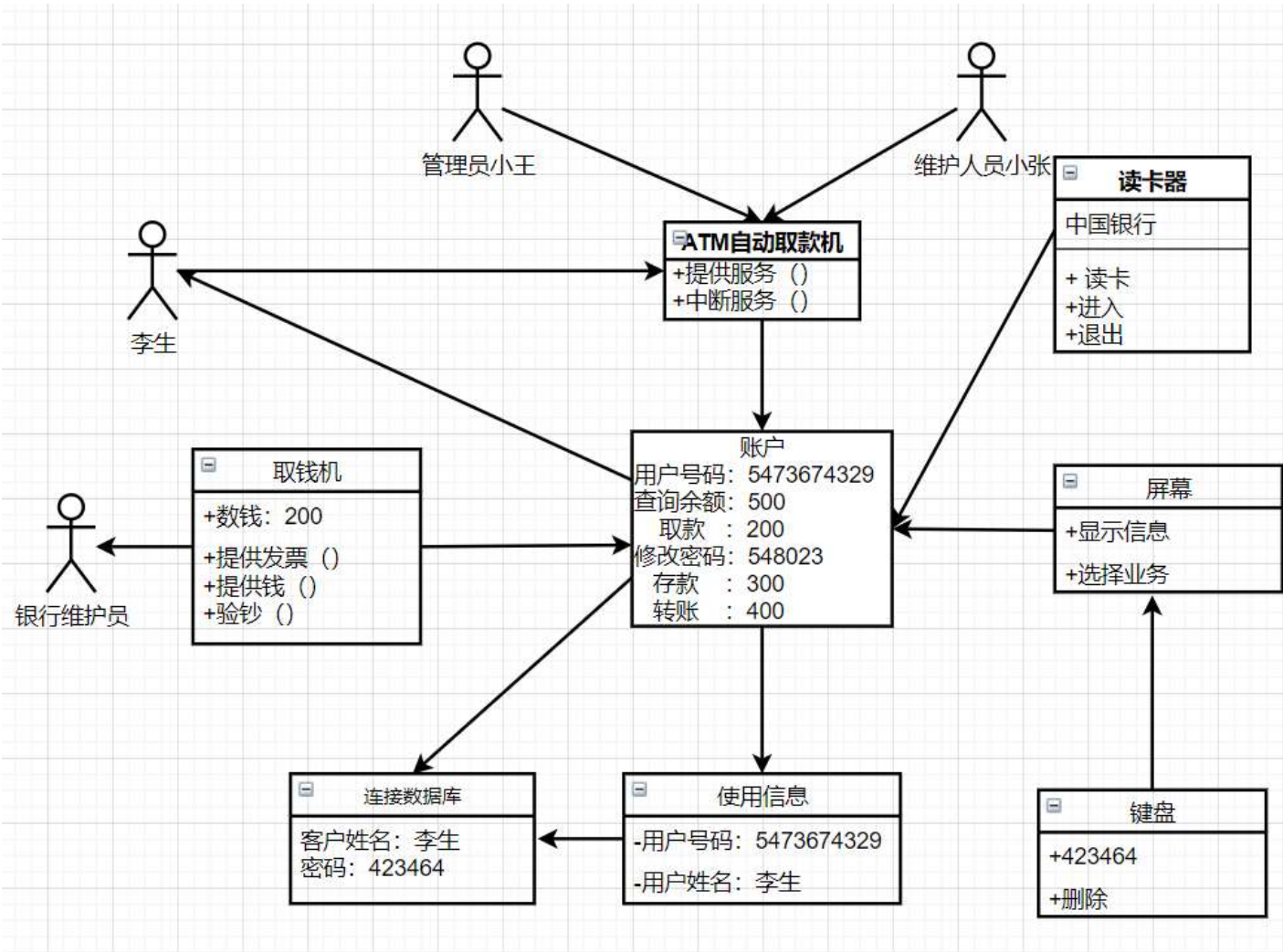


2、类图

ATM自动提款机、客户、维护人员、管理员、读卡器、屏幕、键盘、取钱机和数据库等创建完整类图

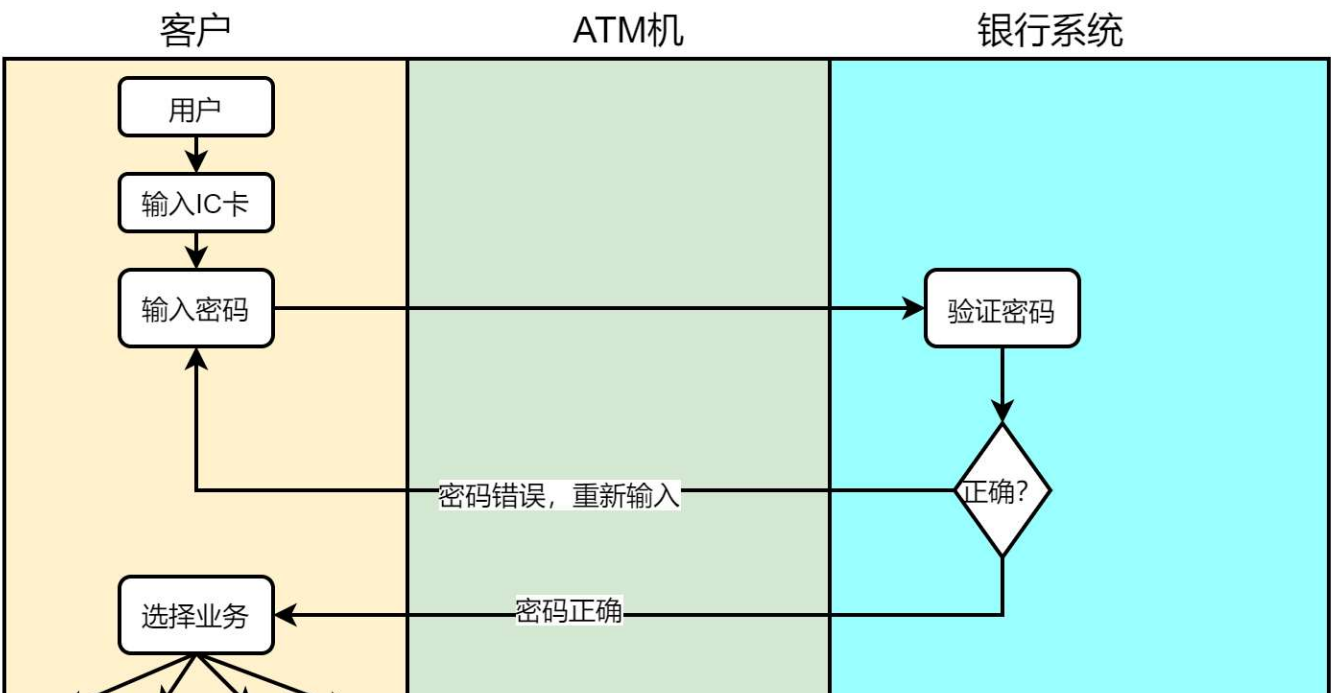


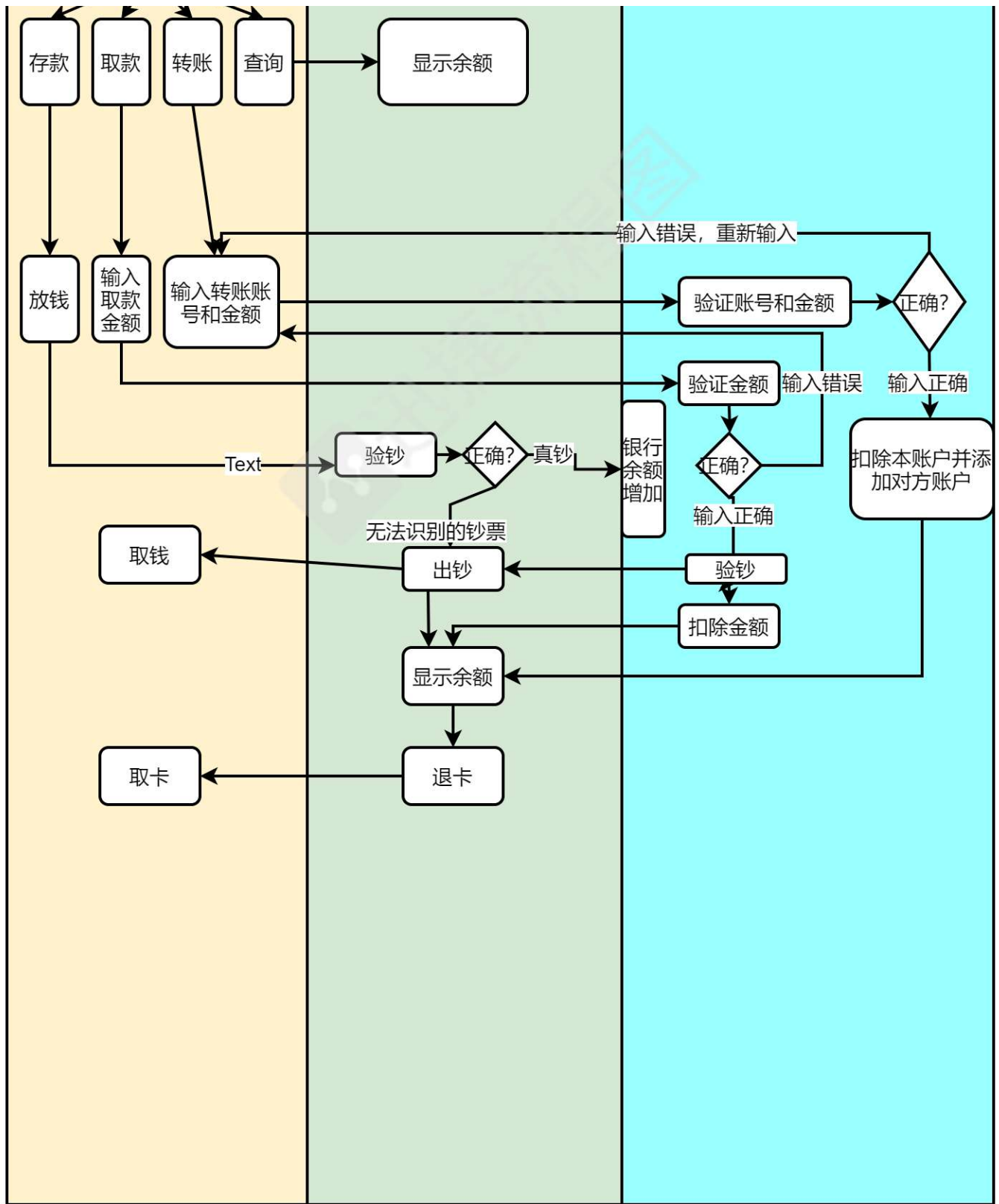
3.对象图



4.活动图

活动图是阐明了业务用例实现的工作流程。业务工作流程说明了业务为向所服务的业务主角提供其所需的价值而必须完成的工作。业务用例由一系列活动组成，它们共同为业务主角生成某些工件。工作流程通常包括一个基本工作流程和一个或多个备选工作流程。工作流程的结构使用活动图来进行说明。





5.状态图

状态图是描述一个实体基于事件反应的动态行为，显示了该实体如何根据当前所处的状态对不同的事件做出反应的。通常我们创建一个UML状态图是为了以下的研究目的：研究类、角色、子系统、或组件的复杂行为。

实体类

1.银行类

- (1) 在银行办理业务的客户，应该被银行记录在册；

2.客户类

- (1) 记录客户拥有的每一张银行卡的银行卡号；

登录管理类

1.登陆 login类

- (1) 针对每张银行卡的卡号，存储密码；
- (2) 给与客户三次输入密码的机会；
- (3) 针对客户输入的密码，辨认真误，并给出反馈；
- (4) 若密码正确，则允许登入；若错误次数超过三次，冻结此账号；
- (5) 若超时，则退出登录；

2.存款 save类

- (1) 限制存取、取款的最高限额；
- (2) 根据存入、取出的数额，判断是否成功，给出反馈，并记录；
- (3) 若超时，则退出登录；

3.转账

- (1) 限制转账的最高限额；
- (2) 输入转账账号和金额，若符合限制，则转账成功后给予反馈；若失败，则提醒用户；
- (3) 若超时，则退出登录；

3.查询余额 Query类

- (1) 记录银行卡内的余额，接受请求后返回余额；

4.修改密码 Passwd类

- (1) 记录原密码；
- (2) 客户输入新密码后，检测密码是否格式正确，则给出修改成功的反馈，并记录新密码；若格式不正确，则给出修改失败的反馈；
- (3) 若超时，则退出登录；

5.扣除手续费 charge类

(1) 客户办理完业务后，检测账号是否跨行，若跨行，收取手续费；

6.打印凭条 print类

(1) 客户办理完业务后，询问是否打印凭条；

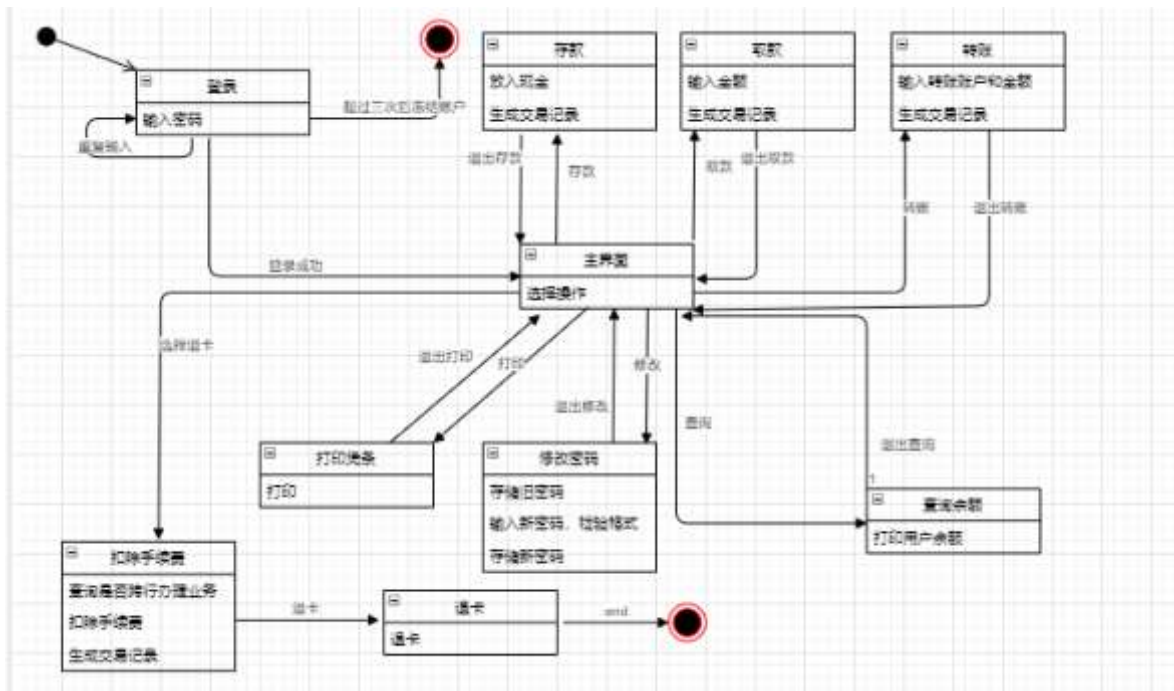
(2) 若客户超时未响应，则退出登录；

界面类

1.用户界面设计 UI类

(1) 在该类中，进行文本设计，通过调用上述类中的函数，给用户提供实现功能的接口和简单的交互界面；

如上所述，得出下列UML：

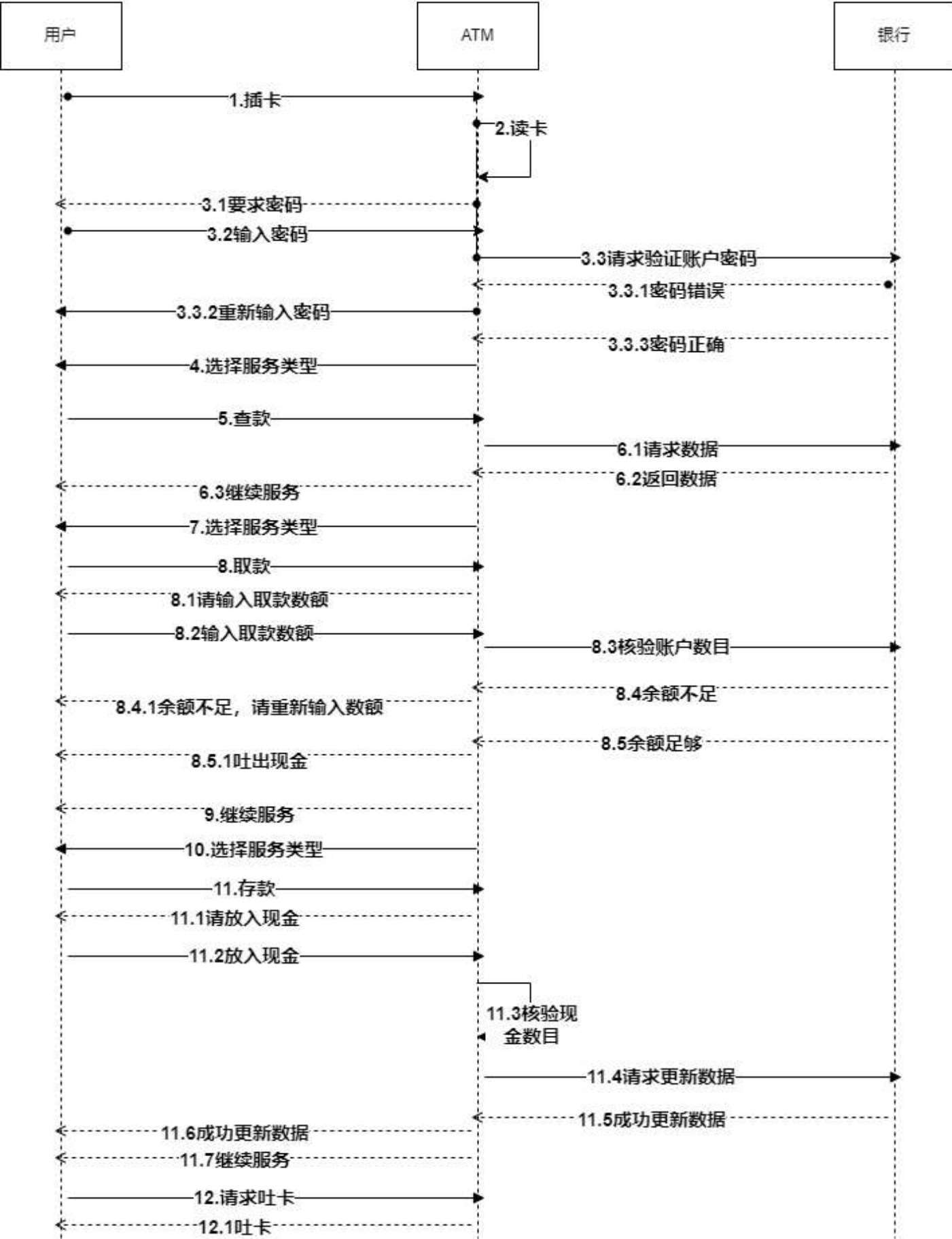


6.时序图

时序图（Sequence Diagram），亦称为序列图或循序图，是一种UML行为图。它通过描述对象之间发送消息的时间顺序显示多个对象之间的动态协作。它可以表示用例的行为顺序，当执行一个用

例行为时，时序图中的每条消息对应了一个类操作或状态机中引起转换的触发事件。

Sequence Diagram



7.协作图

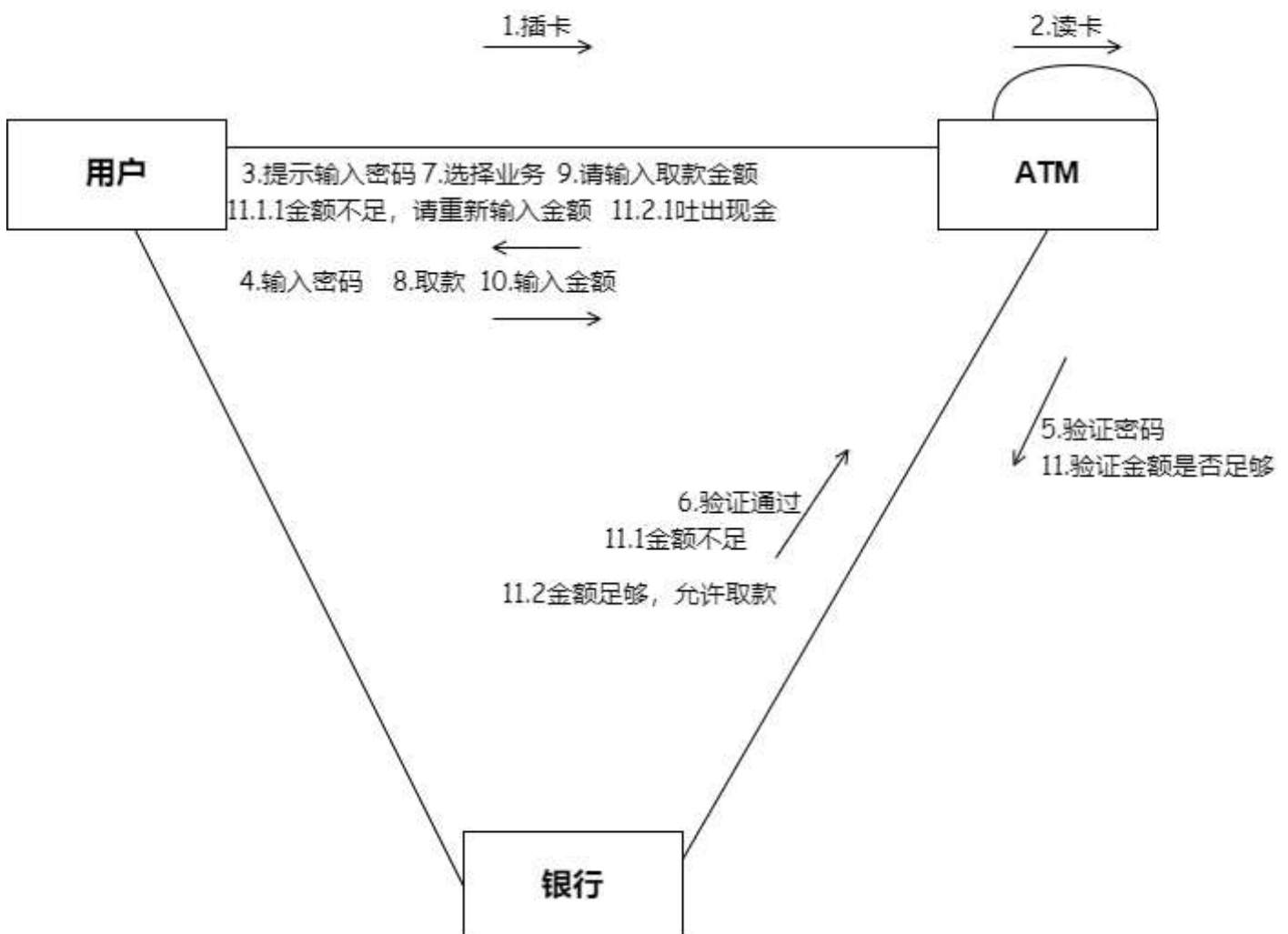
协作图是一种交互图，强调的是发送和接收消息的对象之间的组织结构。一个协作图显示了一系列的对象和在这些对象之间的联系以及对象间发送和接收的消息。对象通常是命名或匿名的类的实例，也可以代表其他事物的实例，例如协作、组件和节点。使用协作图来说明系统的动态情况。协作图使描述复杂的逻辑或多个平行事务变得容易。

协作图显示某组对象如何为了由一个用例描述的一个系统事件而与另一组对象进行协作的交互图。使用协作图可以显示对象角色之间的关系，如为实现某个操作或达到某种结果而在对象间交换的一组消息。如果需要强调时间和序列，最好选择序列图；如果需要强调上下文相关，最好选择协作图。

协作图用于显示对象之间如何进行交互以执行特定用例或用例中特定部分的行为。设计员使用协作图和序列图确定并阐明对象的角色，这些对象执行用例的特定事件流。它们是主要的信息来源，用于确定类的职责和接口。

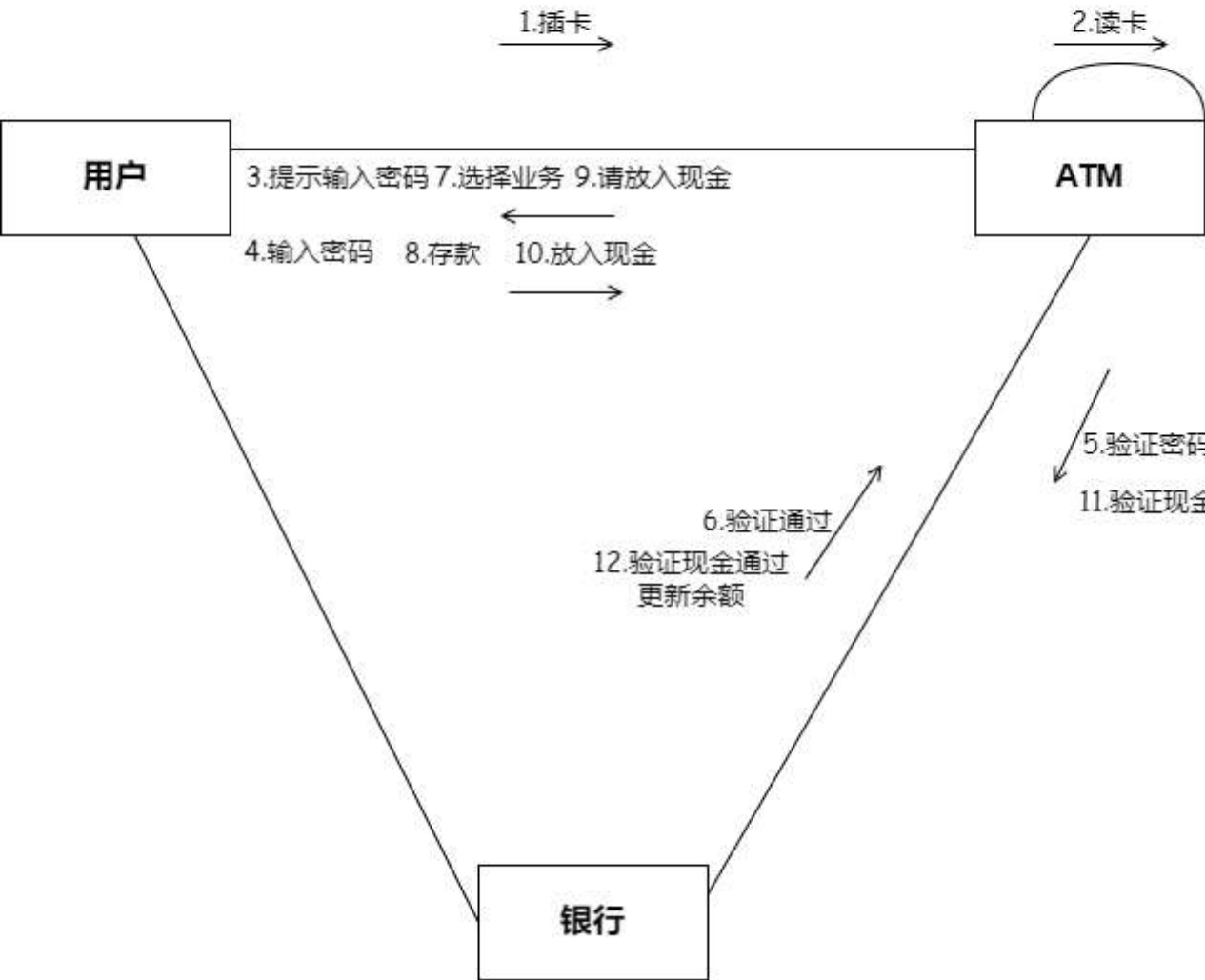
a) 取款协作图

Communication Diagram (取款)



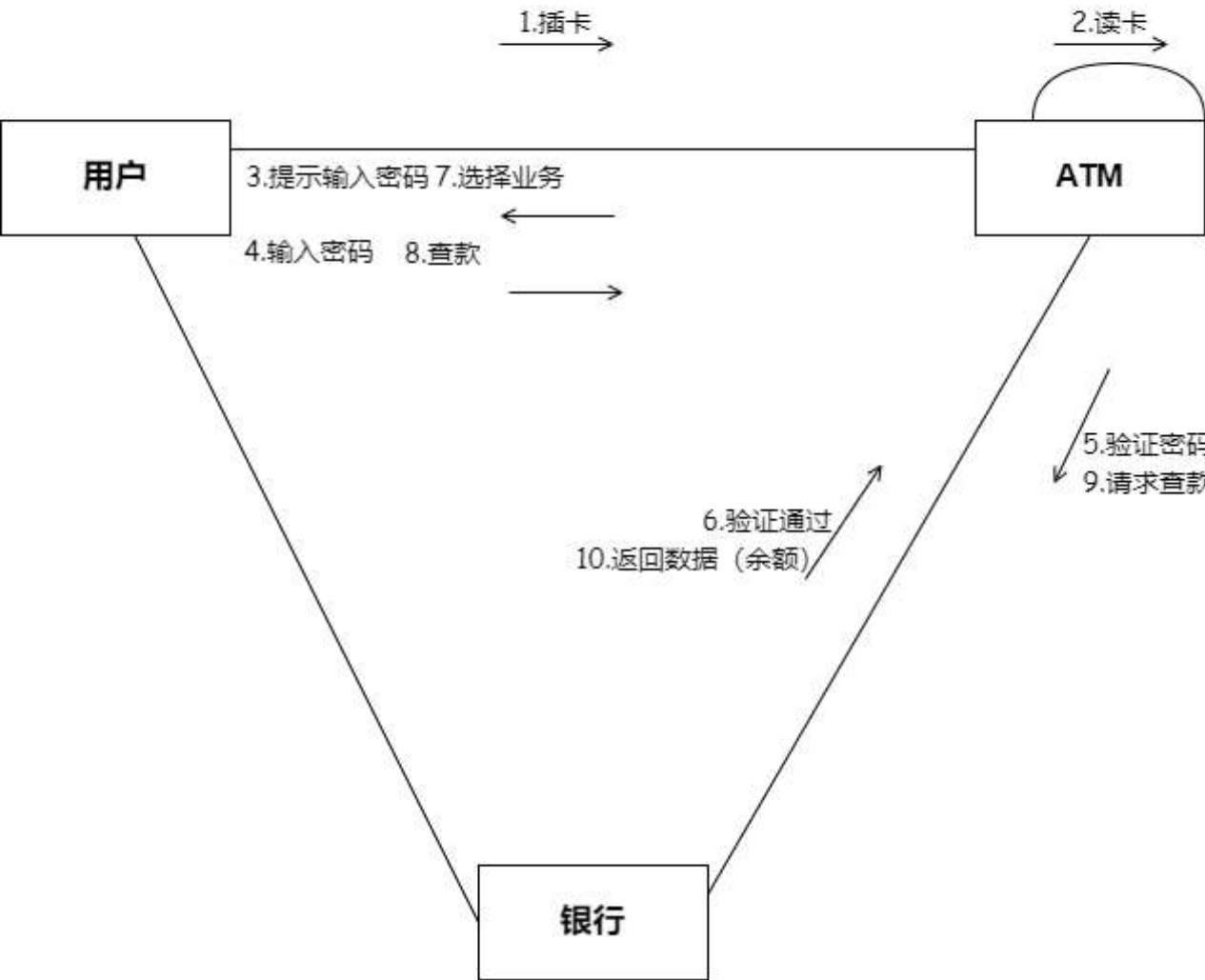
b) 存款协作图

Communication Diagram (存款)

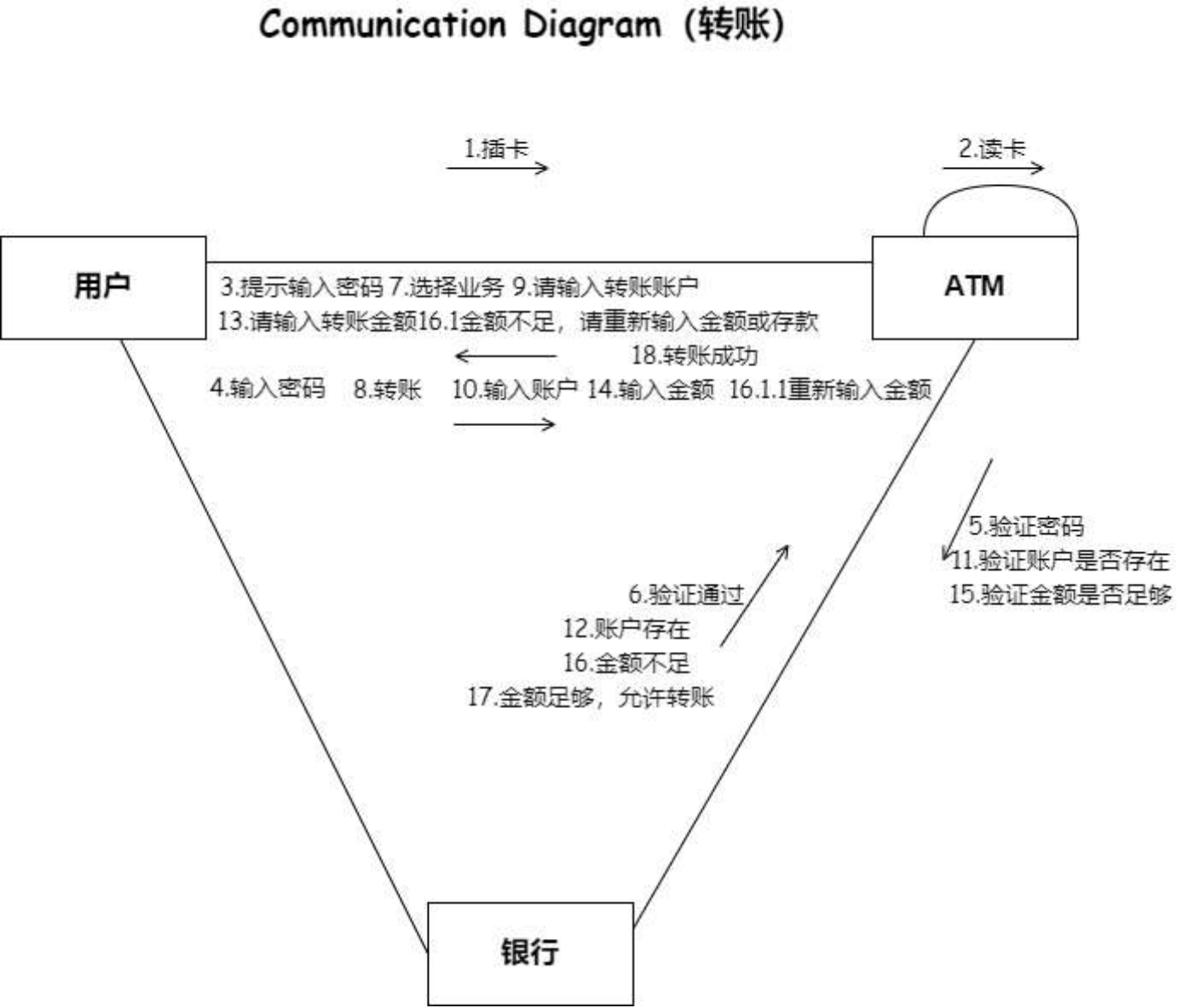


c) 查询余额协作图

Communication Diagram (查款)

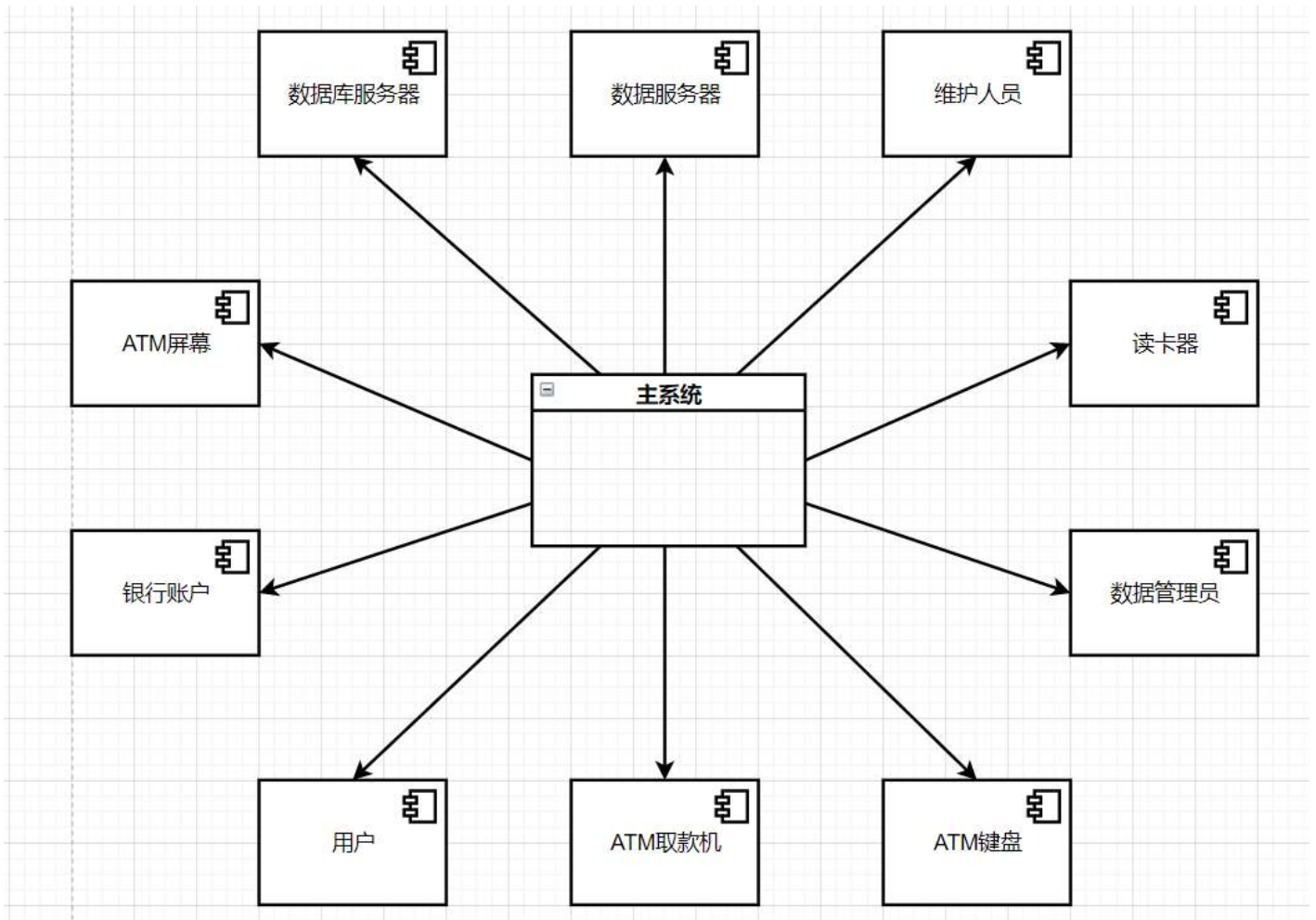


d)转账协作图



8.组件图

(1) 系统组件图：在本系统中，我们可以对数据库服务、维护人员、读卡器、数据管理员、ATM键盘、ATM取款器、用户、银行账户、ATM屏幕分别创建对应的构件进行映射。



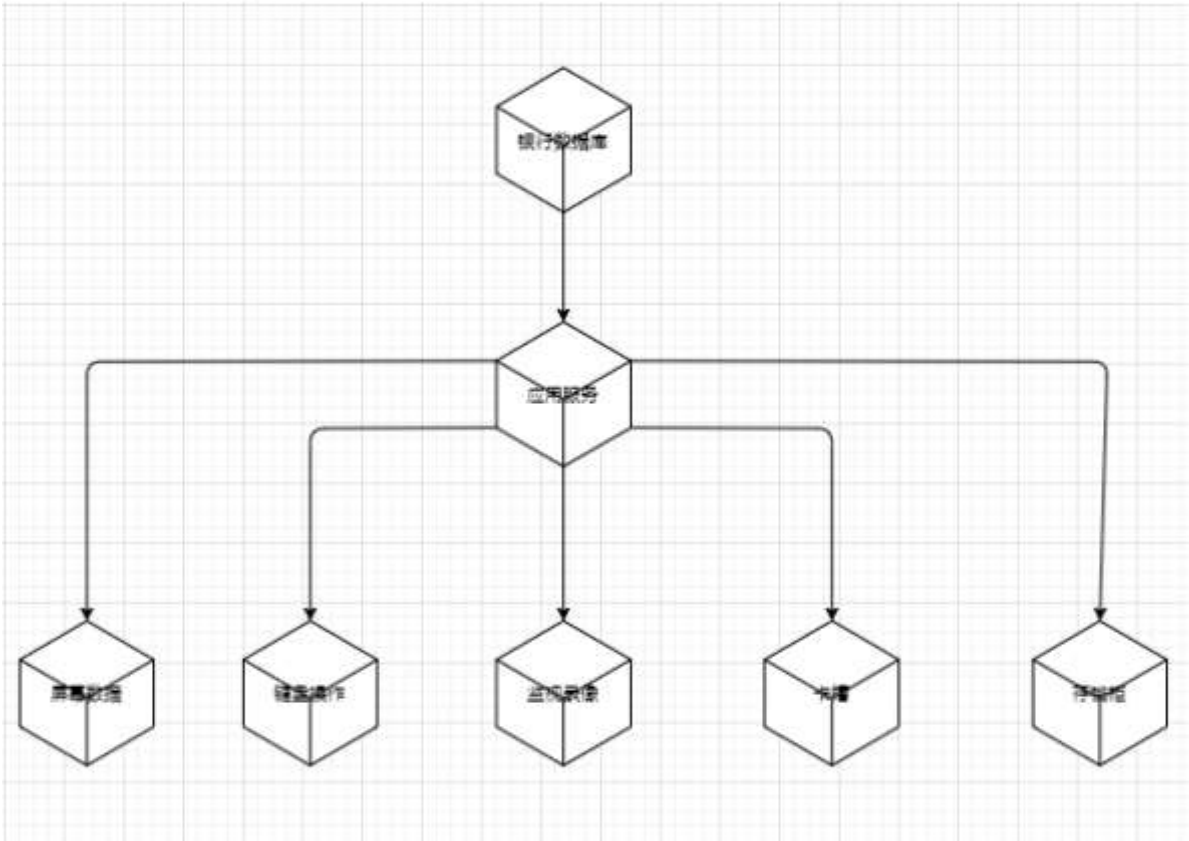
9.部署图

部署图是用来显示系统中软件和硬件的物理架构。从部署图中，您可以了解到软件和硬件组件之间的物理关系以及处理节点的组件分布情况。使用部署图可以显示运行时系统的结构，同时还传达构成应用程序的硬件和软件元素的配置和部署方式。

银行的内部部署：

银行作为提供服务的客体，为客户、银行职员提供相应的设备，

- (1) 职员手动办公时的键盘操作；
- (2) 为客户、职员提供的屏幕数据；
- (3) 保障金钱安全的监控；
- (4) 为用户自主办理业务提供的卡槽、存钱柜；



10.定时图

根据状态图部分的分析，在进行相应操作时，若超时未响应，为保证钱财安全，将退出当前操作；

