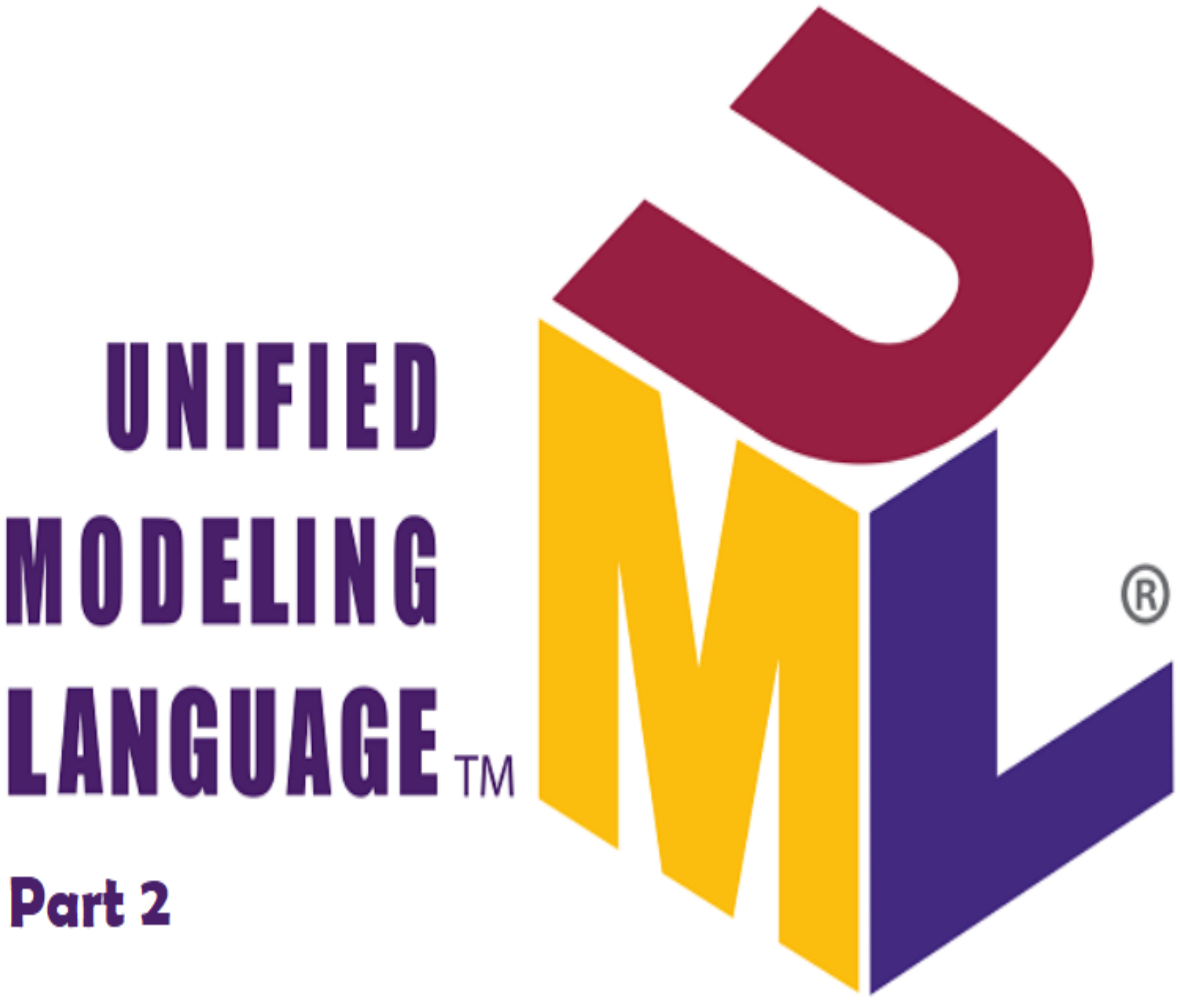
(Use Case Diagrams)



نمودار use case

نمودار use case در ساده ترین حالت خود نمایشی از تعامل کاربر با سیستم است که رابطه بین کاربر و use case های متفاوتی که کاربر در آن درگیر است را نشان می دهد. این نمودار UML شکل اولیه الزامات سیستم برای یک برنامه نرم افزاری جدید در دست توسعه است. use case ها رفتار مورد انتظار (چه چیزی) را مشخص می کند و نه روش دقیق ایجاد آن (چگونه). مفهوم کلیدی مدل‌سازی use case این است که به ما کمک می‌کند یک سیستم را از **دیدگاه کاربر نهایی** طراحی کنیم. بنابراین، یک نمودار use case، نقشه‌های اولیه سیستم شما است، که نمایش ساده و گرافیکی کاری را که سیستم واقعاً باید انجام دهد را ارائه می‌کند.

نمودارهای use case، عملکردهای سطح بالا و دامنه یک سیستم را توصیف می کنند. این نمودارها همچنین تعاملات بین سیستم و بازیگران آن را مشخص می کند. در واقع توصیف می‌کنند که سیستم چه کاری انجام می‌دهد و بازیگران چگونه از آن استفاده می‌کنند، اما با نحوه عملکرد داخلی سیستم ارتباطی ندارد. این طراحی سطح بالا بارها و بارها اصلاح می شود تا تصویری کامل و کاربردی از سیستم به دست آید.

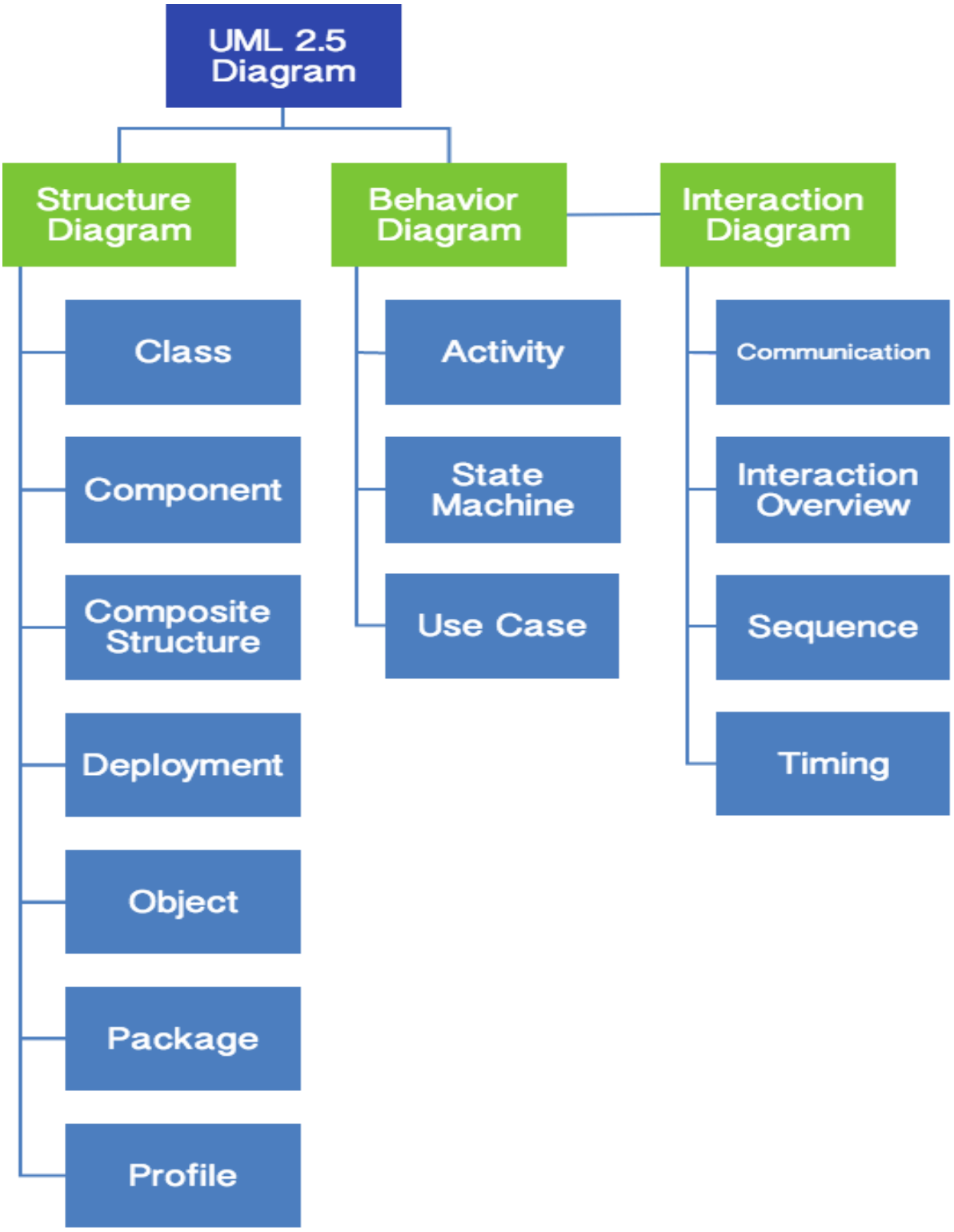
هر use case باید نتایج قابل مشاهده و ارزشمندی را به بازیگران سیستم ارائه دهد.

نمودار use case معمولا ساده است. use case جزئیات را نشان نمی دهد.

نمودار use case باید ساده باشد و فقط شامل چند شکل باشد. اگر سیستم شما بیش از مثلا ۲۰ ، use case دارد، احتمالاً در مسیر اشتباهی از مدلسازی سیستم خود در استفاده از نمودار usecase هستید.

نمودارهای use case فقط الزامات عملکردی یک سیستم را نشان می دهد. سایر الزامات مانند قوانین کسب و کار، الزامات کیفیت خدمات و محدودیت های پیاده سازی باید به طور جداگانه و دوباره با سایر نمودارهای UML نشان داده شوند.

شکل زیر سلسله مراتب نمودار UML و موقعیت UML Use Case Diagram را نشان می دهد. همانطور که مشاهده می کنید، نمودارهای use case از خانواده نمودارهای رفتاری هستند.



سلسله مراتب نمودار UML **تصویری از یک use case diagram**

use case diagram

شرح نمادها (Notations)

بازیگر

بازیگر(Actor)

- کسی که با use case (عملکرد سیستم) تعامل دارد.
- با اسم مشخص(مثلا مشتری/متصدی باجه و ...) نامگذاری می شود.
- بازیگر نقشی را در business(کسب و کار مورد نظر/محیط عملیاتی مورد نظر ما) ایفا می کند.
- مشابه مفهوم کاربر است، اما یک کاربر می تواند نقش های متفاوتی ایفا کند

مثلا:

یک پروفسور می تواند مربی و محقق نیز باشد

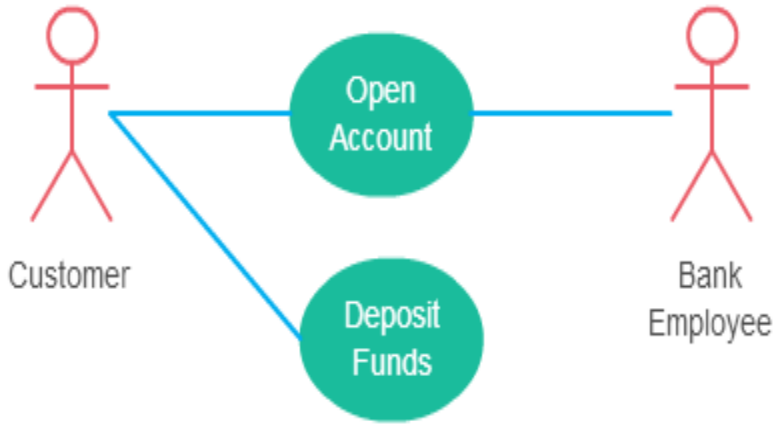
۲ نقش با دو سیستم بازی می کند

- بازیگر بر روی use case ها اثر می گذارد.
- بازیگر نسبت به سیستم (ورودی ها) مسئولیت دارد و Actor انتظاراتی از سیستم (خروجی ها) دارد.

use case

مورد کاربرد (use case)

- عملکرد سیستم(System function) فرآیند بصورت خودکار یا دستی است.
- با فعل + اسم (verb + Noun) یا عبارت اسمی نامگذاری می شود.
- یعنی یه کاری بکن(Do something)
- هر بازیگر باید به یک use case مرتبط باشد، در حالی که برخی use case ها ممکن است به بازیگری مرتبط نباشند.



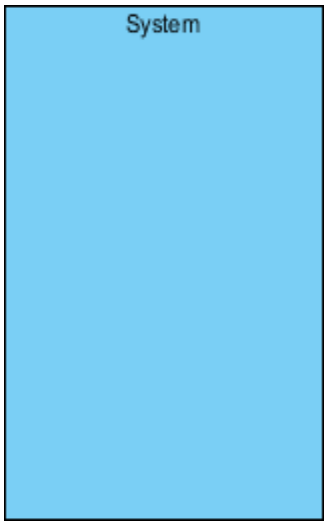
مثالی از use case ها و بازیگران

Communication Link

لینک ارتباط (Communication Link)

ارتباط یک Actor با یک use case توسط یک پیوند(خط) نشان داده می شود.

بازیگران ممکن است توسط association ها به use case ها متصل شوند، که نشان می دهد بازیگر و use case با استفاده از پیام ها با یکدیگر ارتباط برقرار می کنند.

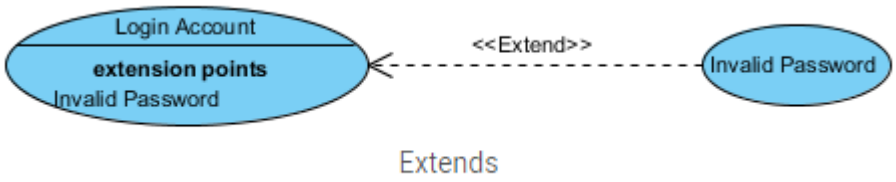


Boundary of system

مرز سیستم (Boundary of system)

- مرز سیستم به طور بالقوه کل سیستم است که در سند الزامات تعریف شده است.
- برای سیستم های بزرگ و پیچیده، هر ماژول ممکن است مرز سیستم باشد.به عنوان مثال، برای یک سیستم ERP برای یک سازمان، هر یک از ماژول ها مانند پرسنل، حقوق و دستمزد، حسابداری و غیره.

ساختار نمودار use case با روابط



Extends

ارتباط Extends

همانطور که از نام آن پیداست، use case پایه را گسترش داده و عملکرد بیشتری را به سیستم اضافه می کند.

یک use case ممکن است برای گسترش(Extend) رفتار use case دیگر استفاده شود.

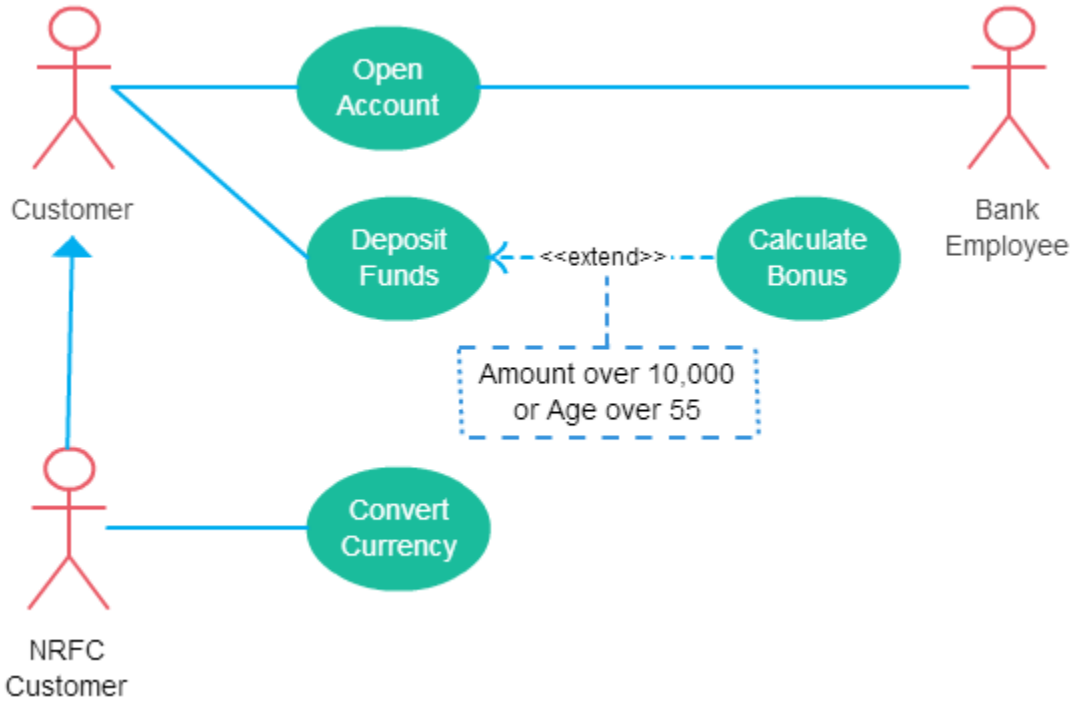
نشان می‌دهد که یک use case با عنوان «Invalid Password» ممکن است شامل رفتاری باشد که توسط use case پایه «Login Account» مشخص شده است.

آن را با یک فلش جهت دار با خط نقطه چین ترسیم کنید.

کلیشه "<<extends>>" به عنوان یک رابطه extend relationship شناسایی می شود.

به عنوان مثالی دیگر، اگر قبل از تغییر سفارش یک مشتری، کاربر باید از یک مرجع بالاتر تأییدیه دریافت کند، در این صورت use case با نام فرضی «Get Approval» ممکن است use case با عنوان مثلا « Modify Order» را گسترش دهد.

چند نکته مهم در قالب مثالی دیگر :



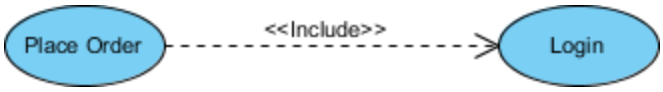
مثالی دیگر از رابطهExtends

- توجه کنید use case گسترده شده به use case (پایه) وابسته است. در تصویر بالا، use case «محاسبه سود/Calculate Bonus» بدون استفاده از «وجوه سپرده/Deposit Funds» چندان معنا ندارد.

- توجه کنید use case گسترده معمولا **اختیاری** است و می تواند به صورت مشروط فعال شود. در تصویر بالا، مشاهده می کنید که use case گسترده فقط برای سپرده های بیش از ۱۰۰۰۰ یا زمانی که سن بالای ۵۵ سال است فعال می شود.
- توجه کنید use case (پایه) باید به تنهایی معنادار باشد. این بدان معناست که باید مستقل باشد و نباید به رفتار use case در حال گسترش تکیه کند.

اگرچه گسترش use case در بیشتر مواقع اختیاری است، اما الزامی نیست. یک use case گسترده می تواند رفتار غیر اختیاری نیز داشته باشد. این بیشتر زمانی اتفاق می افتد که شما رفتارهای پیچیده را مدل سازی می کنید.

به عنوان مثال، در یک سیستم حسابداری، یک use case ممکن است با نام «افزودن ورودی به دفتر حسابداری» باشد. این ممکن است بدنبال خود use case های دیگری را نیز گسترش دهد "افزودن ورودی به دفتر مالیاتی" و "افزودن ورودی به ...". اینها اختیاری نیستند اما به ورودی دفتر حسابداری بستگی دارند. همچنین، آنها رفتار خاص خود را دارند تا به عنوان یک use case جداگانه مدل شوند.

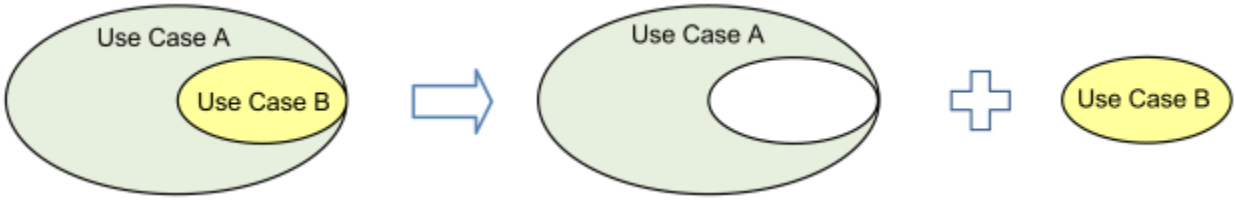


رابطه Include

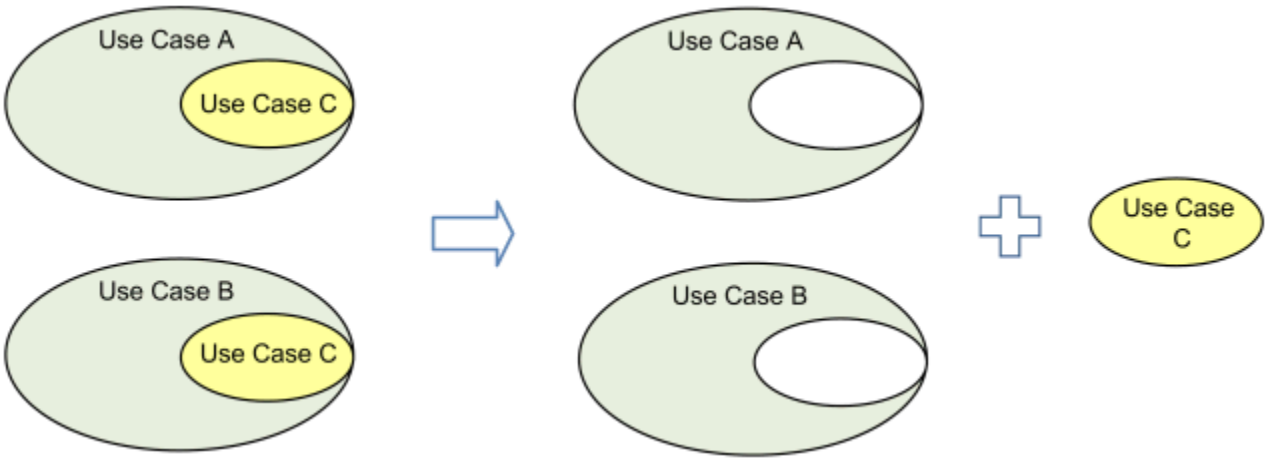
این رابطه نشان می دهد که رفتار use case شامل بخشی از use case (پایه) است.توجه داشته باشید این رابطه Include اجباری است و اختیاری نیست.

رابطه Include در موارد زیر باید استفاده شود :

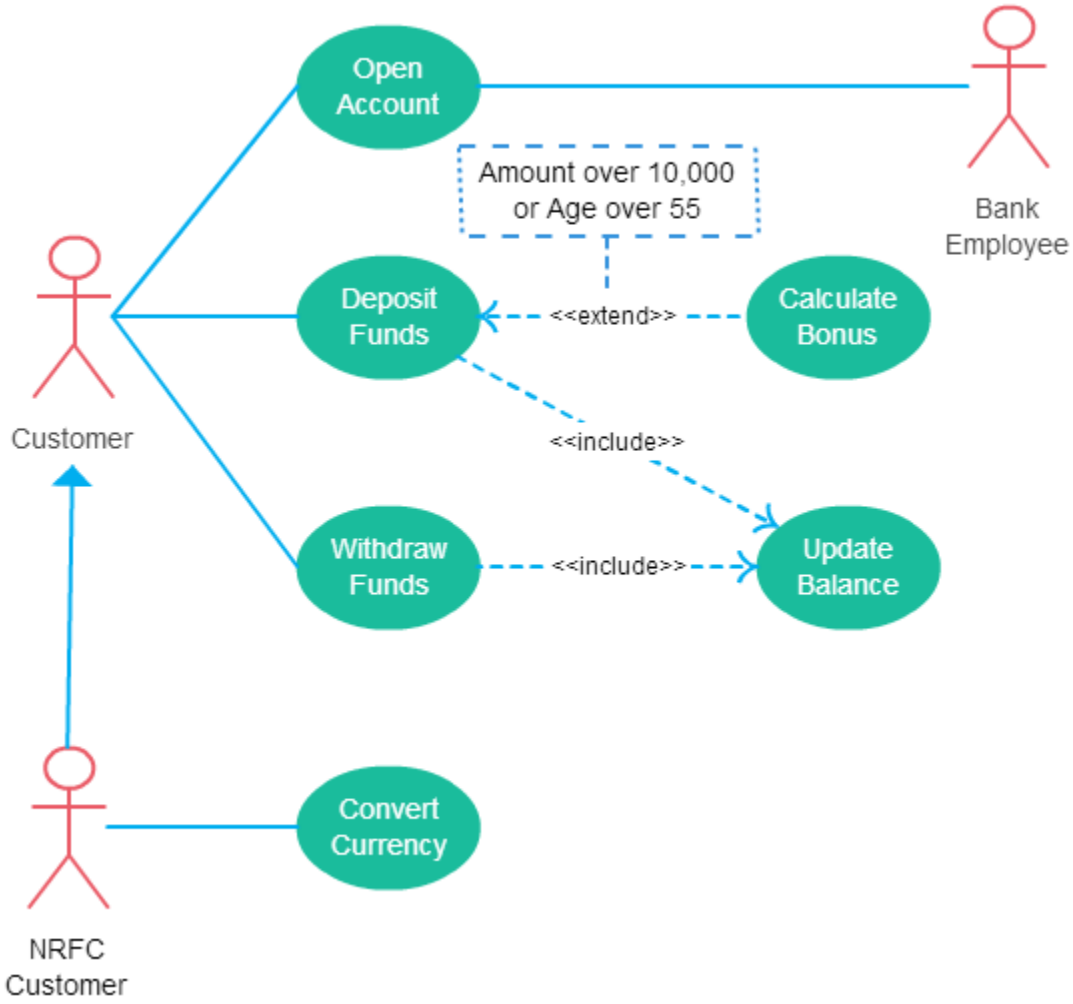
- برای ساده کردن یک use case بزرگ از طریق تقسیم آن به چند use case کوچکتر



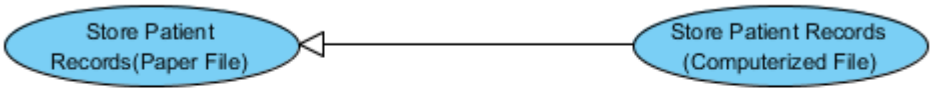
- برای استخراج رفتارهای مشترک دو یا چند use case



- کلیشه "<<Include>>" رابطه را به عنوان یک رابطه Include relationship شناسایی می کند.



- همانطور که از تصویر بالا پیداست use case با نام «Deposit Funds/ وجوه سپرده» و « withdraw funds/برداشت وجوه» هر دو شامل usecase (Include) با نام «update blance/بروز رسانی وجوه» هستند.



رابطه Generalization

رابطه تعمیم یا ارث بری (Generalization)

دو نوع رابطه تعمیم داریم

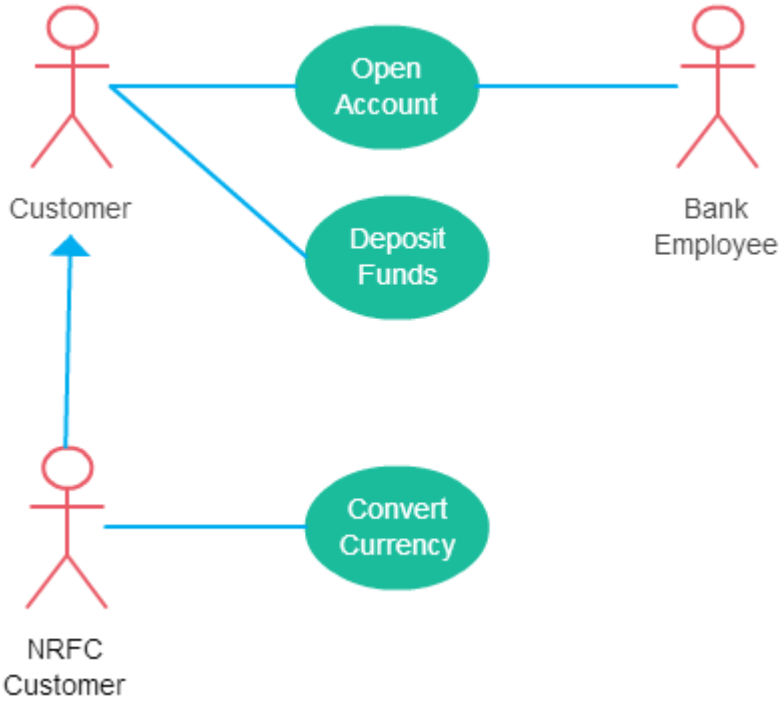
اولی :Generalization of a Use Case

- رابطه تعمیم یک رابطه والد-فرزند بین use caseها است.
- مشخصا use case فرزند پیشرفتی از use case والد است.
- تعمیم به صورت یک فلش جهت دار با سر فلش مثلثی نشان داده می شود.
- نوک پیکان به use case والد اشاره دارد.

- این رابطه زمانی استفاده می شود که رفتار مشترک بین دو use case و همچنین رفتار تخصصی خاص برای هر use case وجود داشته باشد.

برای مثال، در مثال بانکداری قبلی، ممکن است use caseی به نام «پرداخت قبوض» وجود داشته باشد. این را می توان به "پرداخت با Credit Card"، "پرداخت با Bank Balance" و غیره تعمیم داد.

دومی: Generalization of an Actor



Non-Resident Foreign Currency

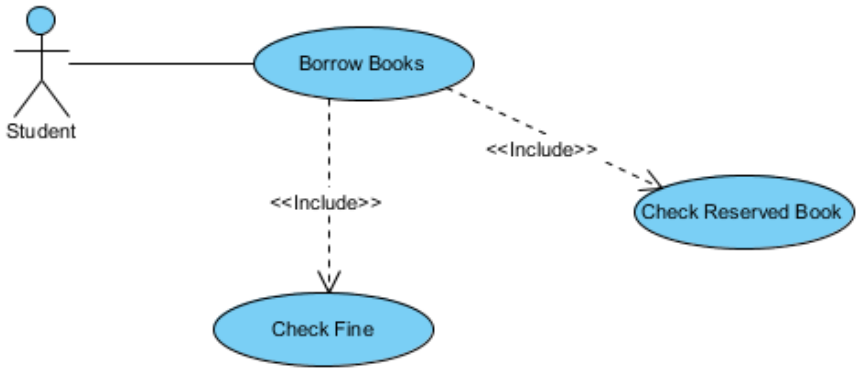
تعمیم یک بازیگر به این معنی است که یک بازیگر می تواند نقش بازیگر دیگر را به ارث ببرد.

مثال دیگر از مطالب بالا را مرور کنیم :

Association Link

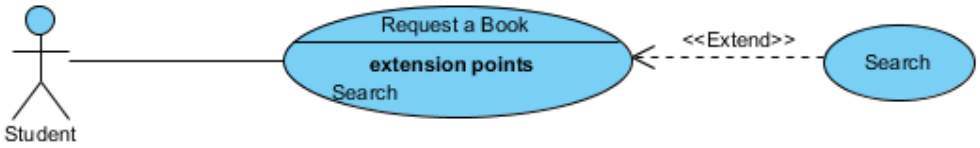
Association Link

Include Relationship



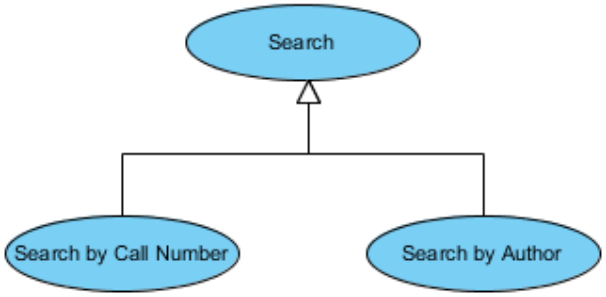
Include Relationship

Extend Relationship



Extend Relationship

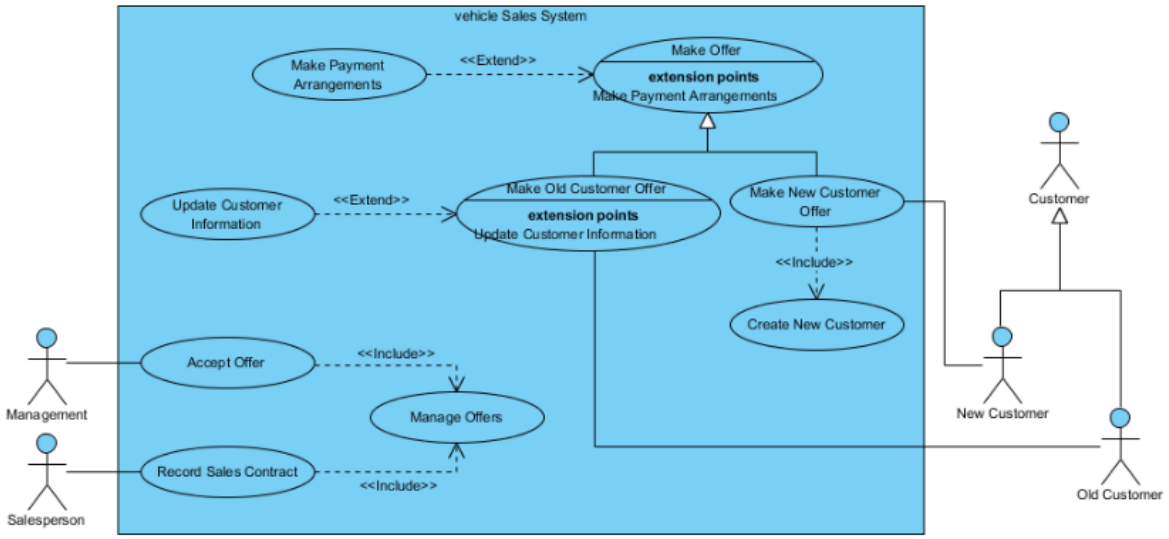
Generalization Relationship



Generalization Relationship

- شکل زیر یک مثال نمودار use case برای یک سیستم فروش خودرو را نشان می دهد. همانطور که می بینید حتی یک سیستم به بزرگی یک سیستم فروش خودرو شامل بیش از ۱۰ use case نیست! این زیبایی مدل سازی use case است.

مدل use case نیز استفاده از extend and include را نشان می دهد. علاوه بر این، ارتباطاتی وجود دارد که بین بازیگران و use caseها است.



- نکته : معمولاً use caseها را به داخل سیستم و بازیگران را خارج از سیستم نشان می دهند.

چگونه بازیگران سیستم خود را شناسایی کنیم

اغلب، توسعه دهندگان و معماران نرم افزار آسان‌ترین راه را برای شروع فرآیند استخراج نیازمندی‌ها با شناسایی بازیگران پیدا می‌کنند. سوالات زیر می تواند به شما در شناسایی بازیگران سیستم خود کمک کند:

- چه کسی از سیستم استفاده می کند؟
- چه کسی سیستم را نصب می کند؟

- چه کسی سیستم را راه اندازی می کند؟
- چه کسی سیستم را نگهداری می کند؟
- چه کسی سیستم را خاموش می کند؟
- چه سیستم های دیگری از این سیستم استفاده می کنند؟
- چه کسانی از این سیستم اطلاعات می گیرند؟
- چه کسی اطلاعات را در اختیار سیستم قرار می دهد؟
- آیا در زمان کنونی هر چیزی به طور خودکار اتفاق می افتد؟

چگونه use caseها را شناسایی کنیم؟

شناسایی use caseها، و سپس فرآیند استخراج مبتنی بر سناریو با پرسیدن اینکه هر بازیگری چه ارزش قابل مشاهده ای را می‌خواهد ادامه می‌یابد. هنگامی که بازیگران شما شناسایی شدند، می توان سوالات زیر را برای شناسایی use caseها پرسید :

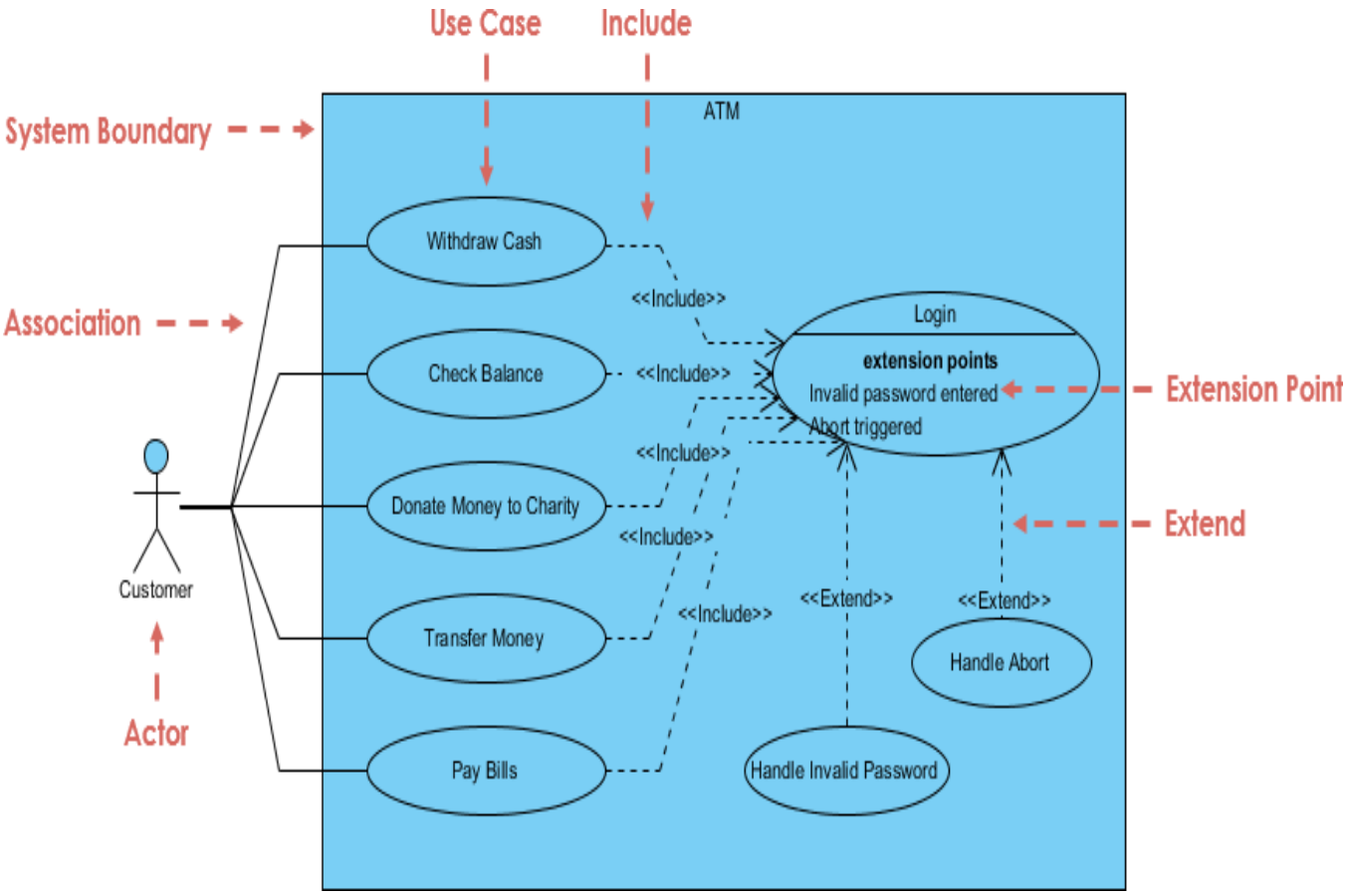
- بازیگر چه کارکردهایی از سیستم می خواهد؟

- آیا سیستم اطلاعات را ذخیره می کند؟ چه بازیگرانی این اطلاعات را ایجاد، می خوانند، به روز رسانی یا حذف خواهند کرد؟
- آیا سیستم نیاز به اطلاع یک بازیگر از تغییرات در وضعیت داخلی دارد؟
- آیا رویدادهای خارجی وجود دارد که سیستم باید درباره آن بداند؟
- چه بازیگری سیستم را از آن اتفاقات آگاه می کند؟

نکات

- همیشه نمودار use case را از دیدگاه بازیگران سازماندهی کنید.
- نمودارهای use case باید ساده و در بالاترین نمای ممکن شروع شوند. تنها در این صورت است که می توان آنها را اصلاح و تفصیل داد.
- نمودارهای use case بر اساس عملکرد هستند و بنابراین باید بر روی "چه" تمرکز کنند نه "چگونه".

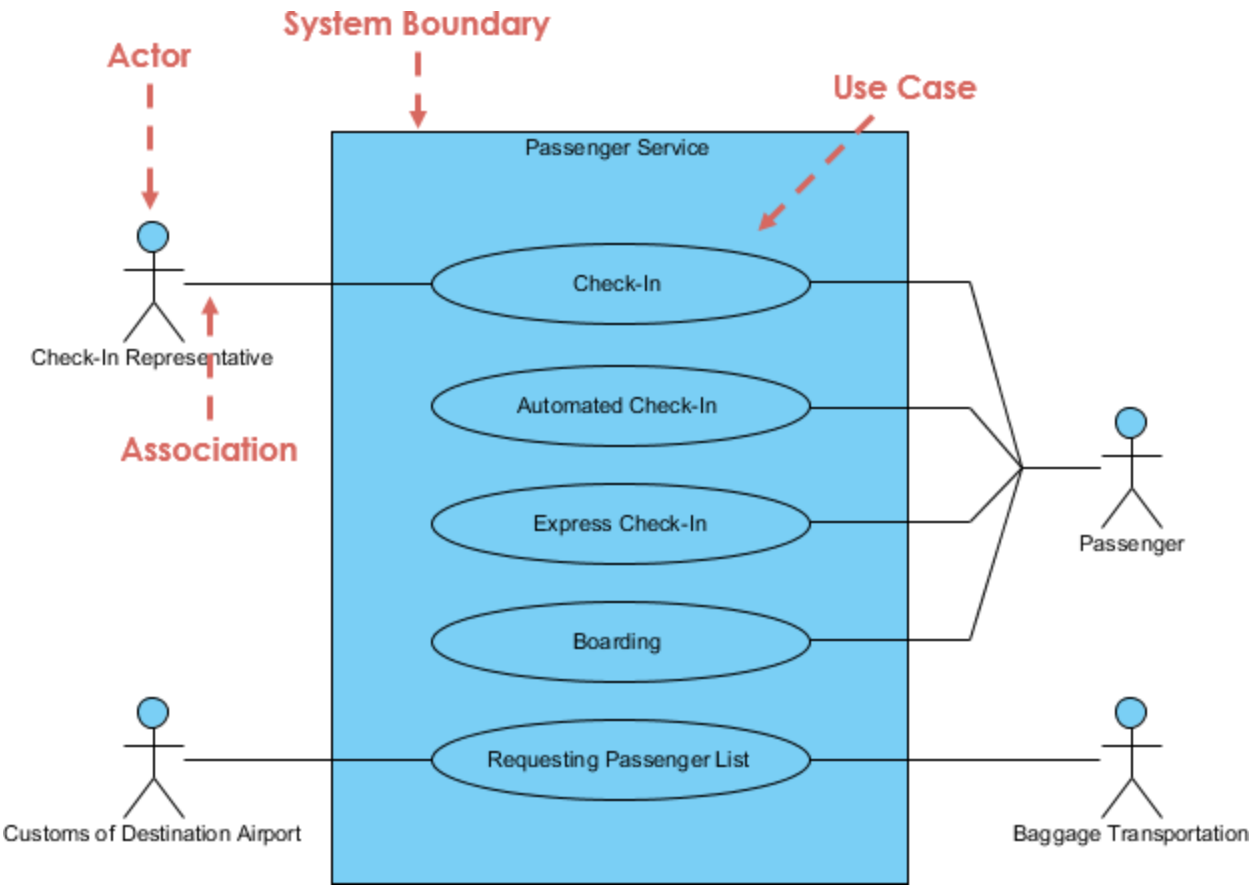
بیایید در انتهای این مطلب چند مثال رایج را با هم ببینیم:



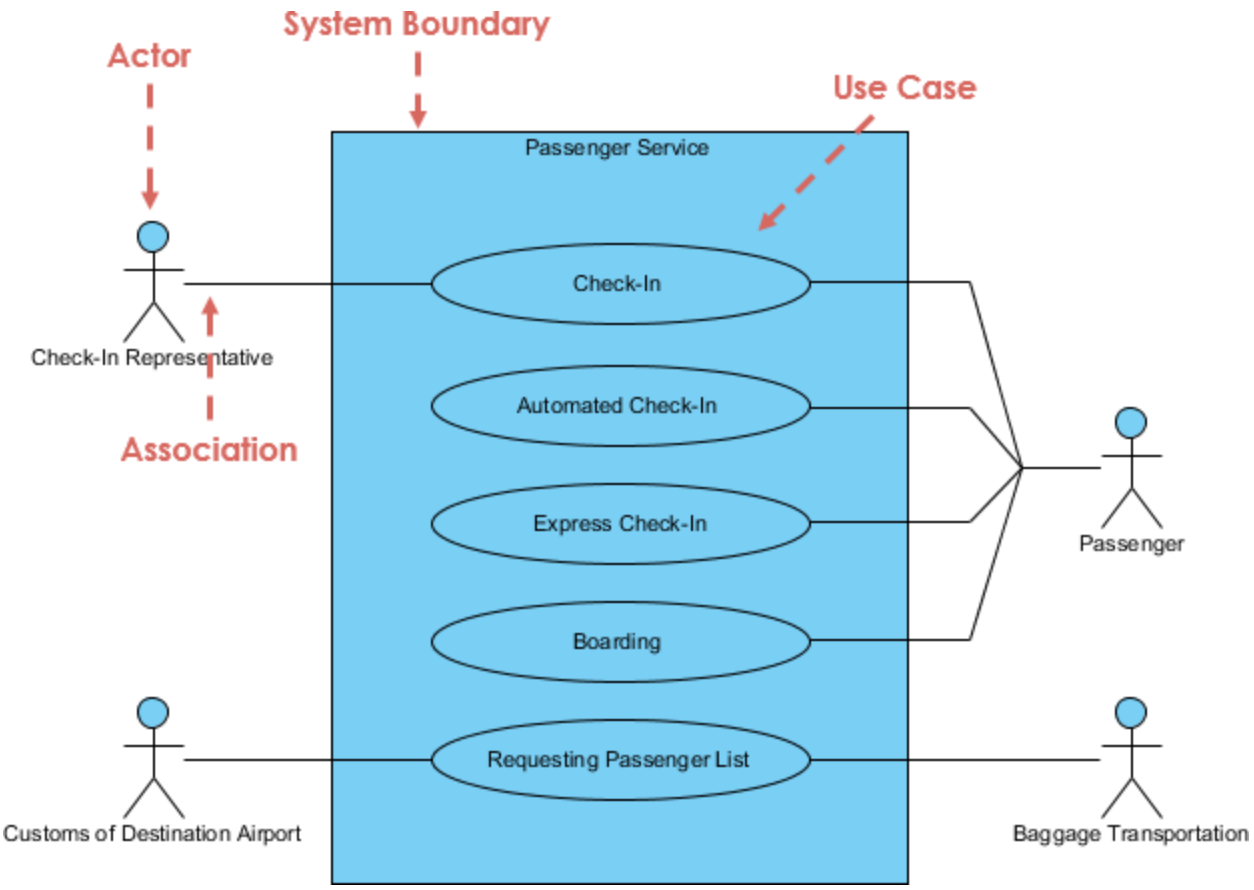
Use Case Diagram ATM

- این مثال نمودار use caseی را نشان می دهد که use case با نام «login» در بین همه use caseها پایه با دو extension points به اشتراک گذاشته شده است تا پردازش ها و درخواست های نامعتبر لغو شود.

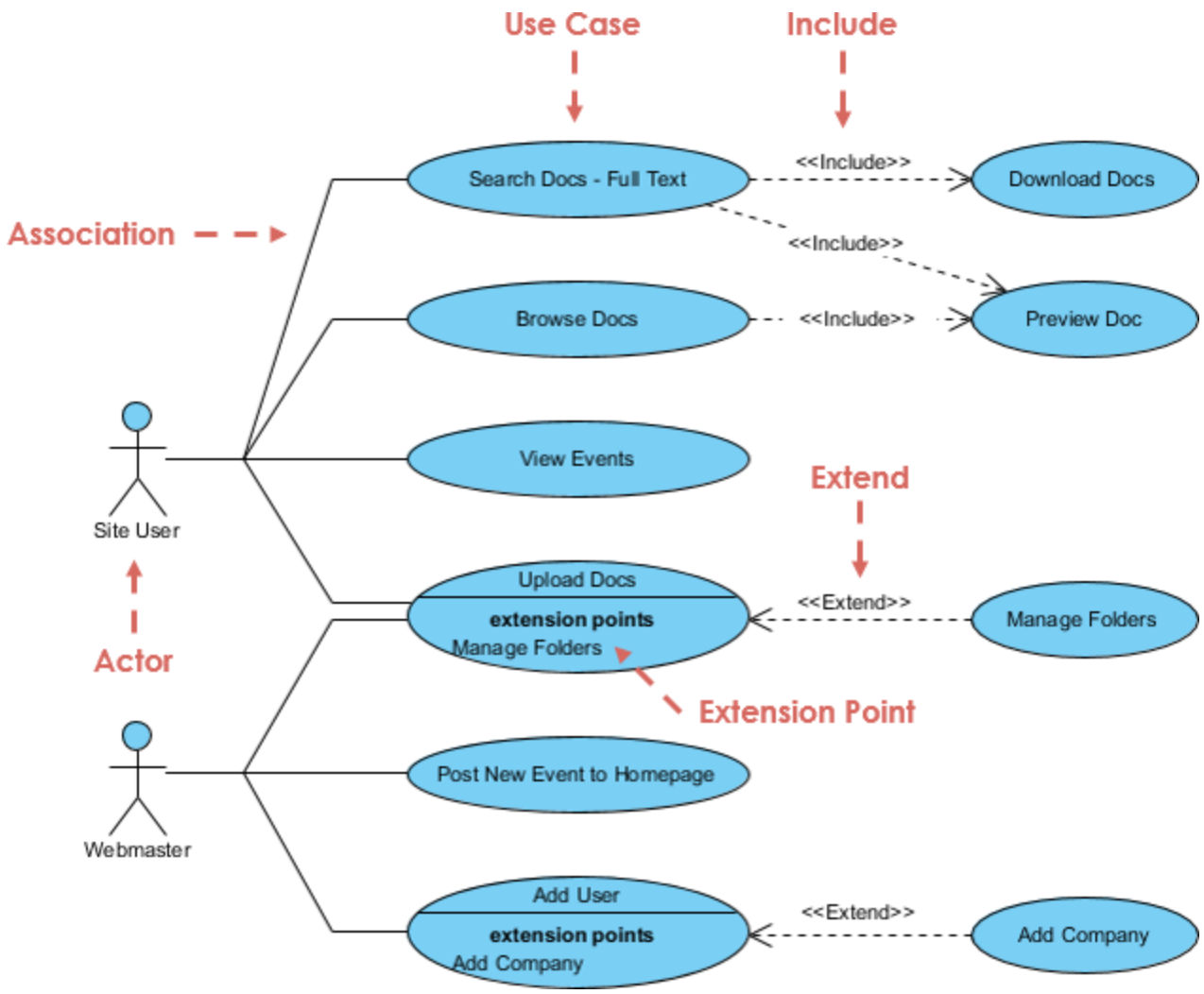
Browse catalog with security user cases in electronic commerce product line.



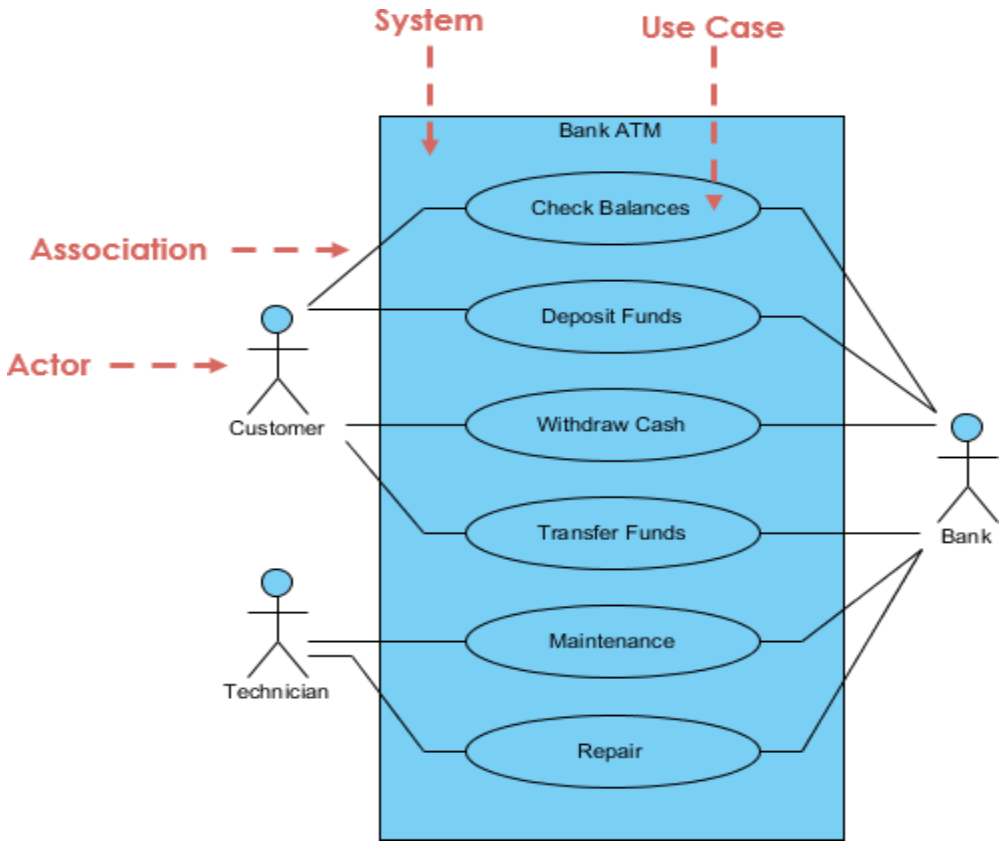
This is a use case diagram example of Passenger Service



This is a use case diagram example of Online Examination System.



Website (structuring use cases with extend and include use case)



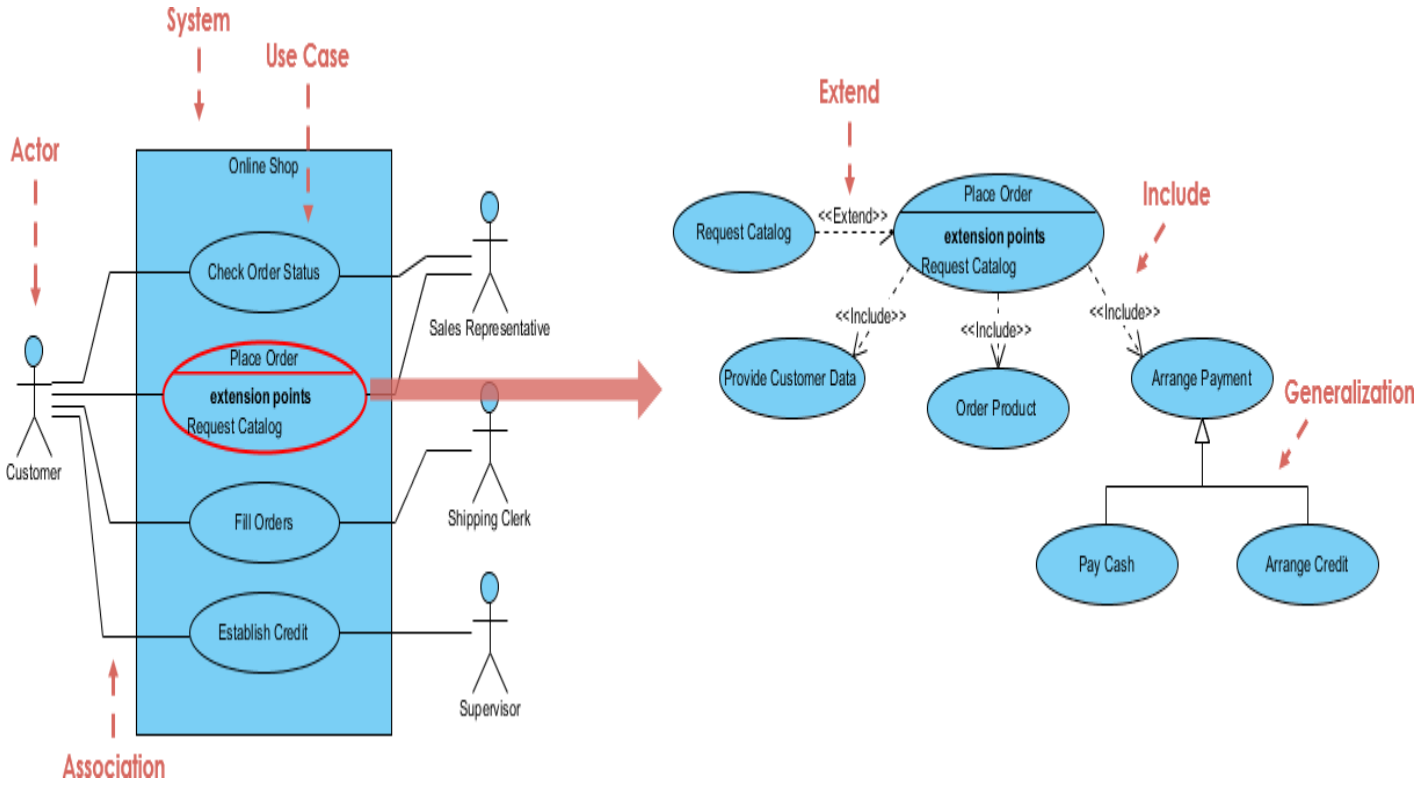
ATM

در تصویر بالا دستگاه عابربانک (ATM) زیرسیستم بانکی است که امکان دسترسی مشتریان بانک به تراکنش های مالی در یک فضای عمومی را بدون نیاز به صندوقدار، کارمند یا عابر بانک فراهم می کند.

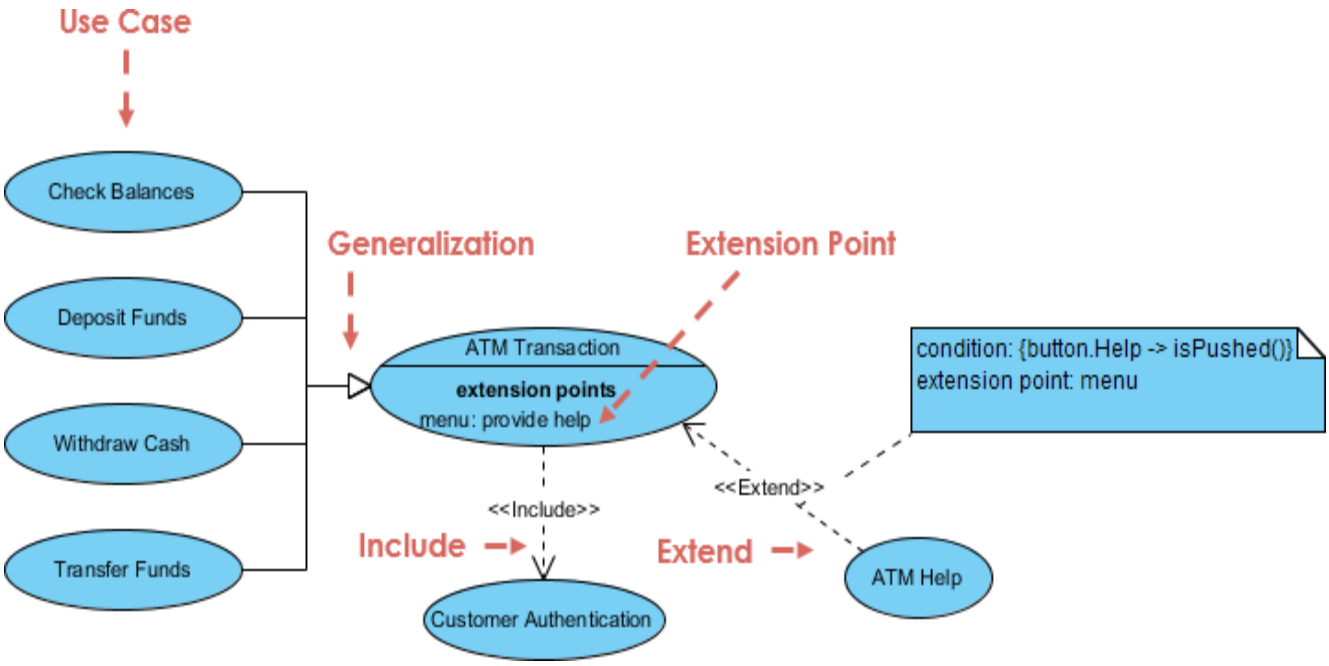
مشتری (بازیگر) برای بررسی موجودی حساب های بانکی خود، واریز وجوه، برداشت وجه نقد و/یا انتقال وجه از دستگاه خودپرداز بانکی استفاده می کند. تکنسین ATM تعمیر و نگهداری و تعمیرات را انجام می دهد. همه

این use case ها، خواه مربوط به تراکنش های مشتری باشد یا به خدمات خودپرداز، بانک عامل را نیز در بر می

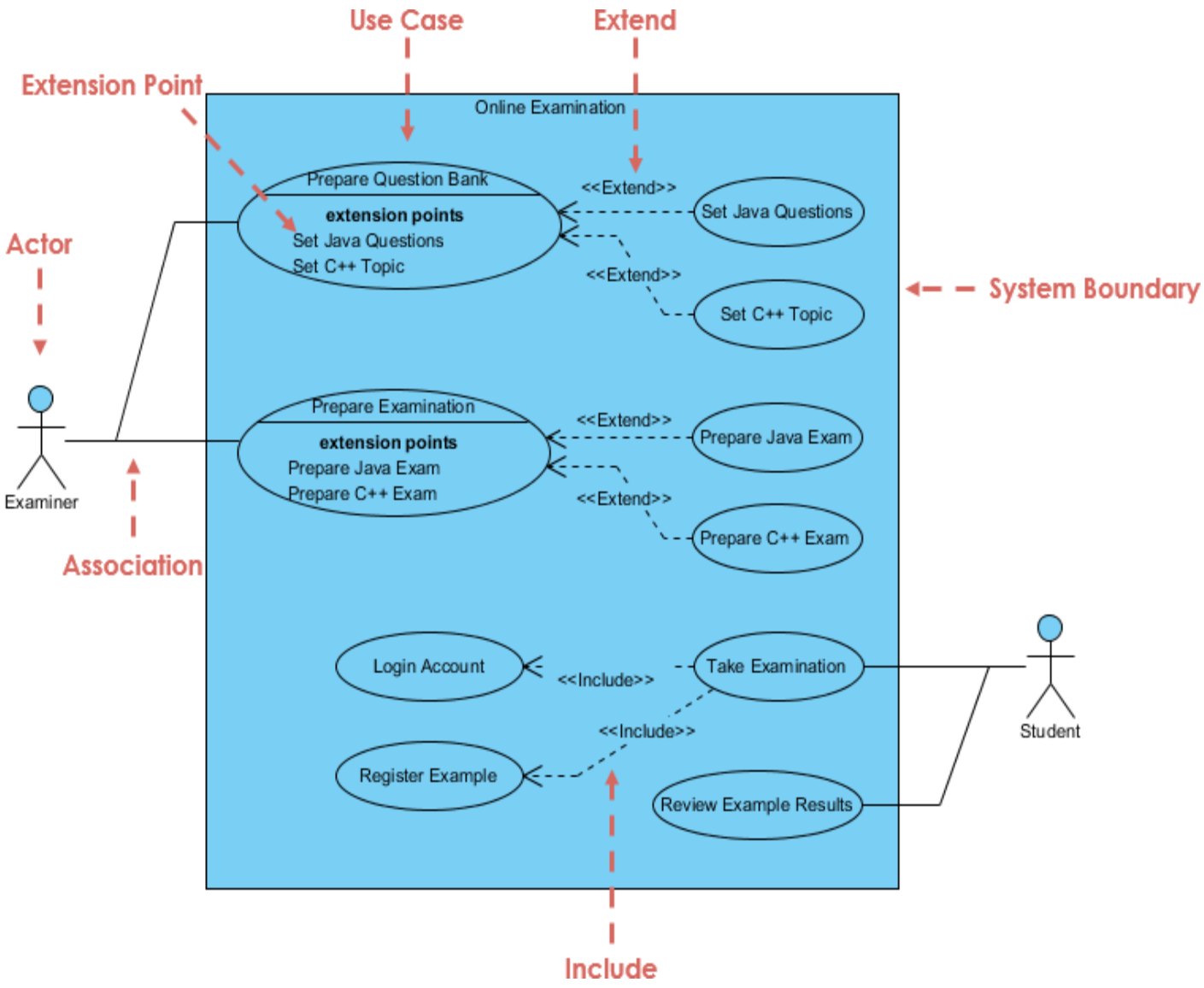
گیرد.



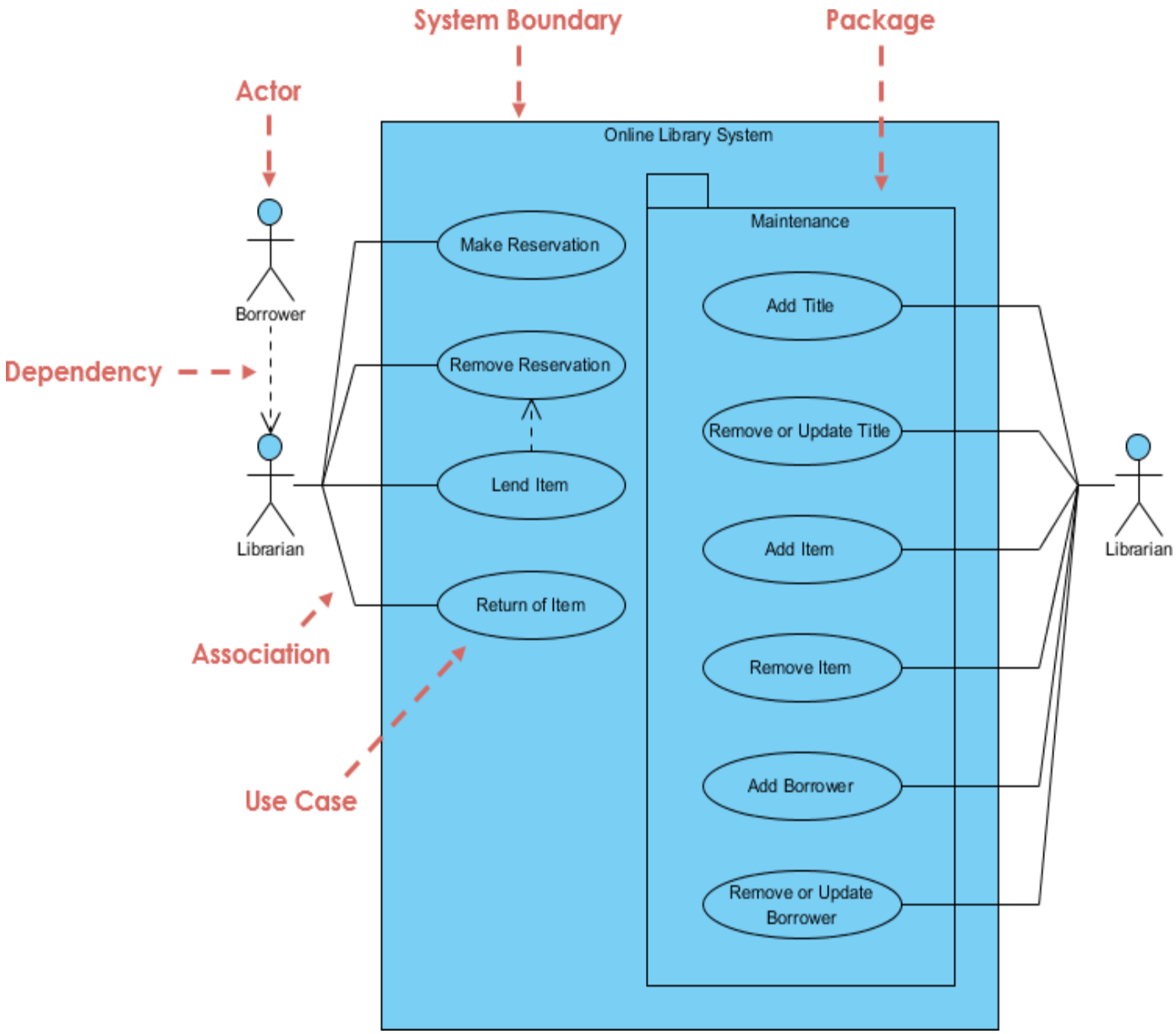
Refining Use Case



Using Extension Point



Online Examination System



Online Library System