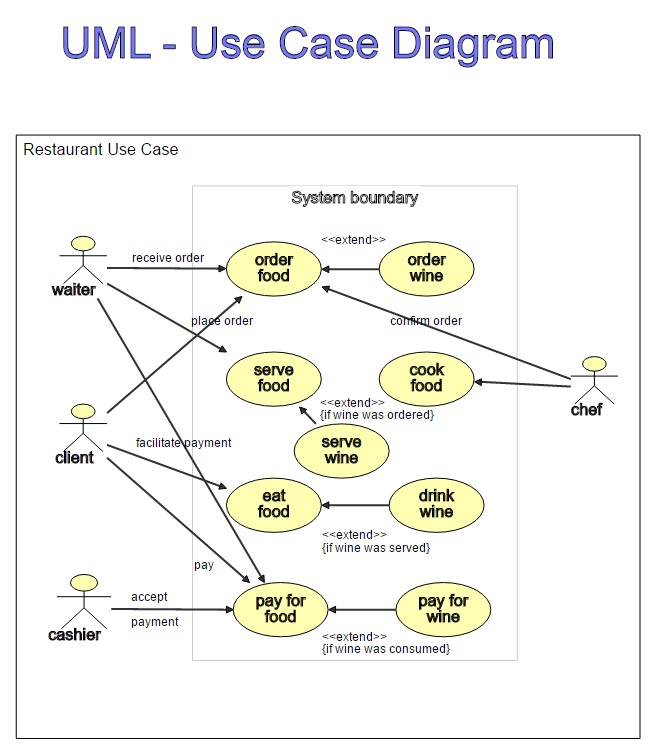
**Use-case図：**

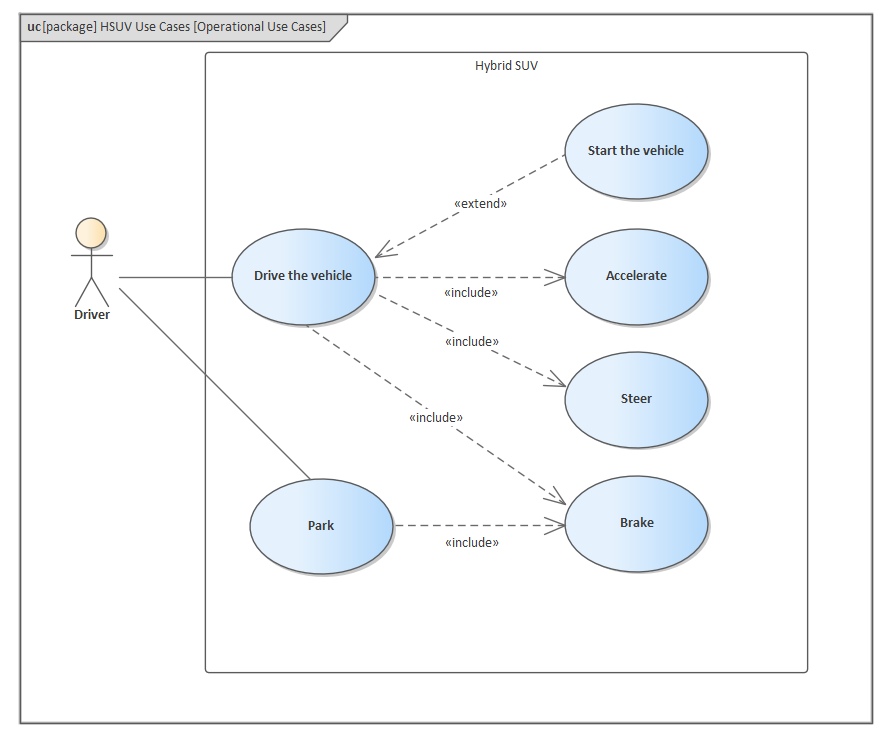
* システム設計の最初に用いる図となります
* 要件と利用者の関係を表して、システムが実現すべき流れを表現する
  + 要件：Use-case
  + 利用者：Actor/Stakeholder
  + 関係を線で表示する
* ユースケースはシステムがアクターに対して提供する機能・操作
* ユースケース図を考える時、ユースケースは要件定義を基に、アクターに対して何らかの目的を達成するためにある
  + ユースケースが価値を与える = アクターはユースケースを利用することで便益を得る

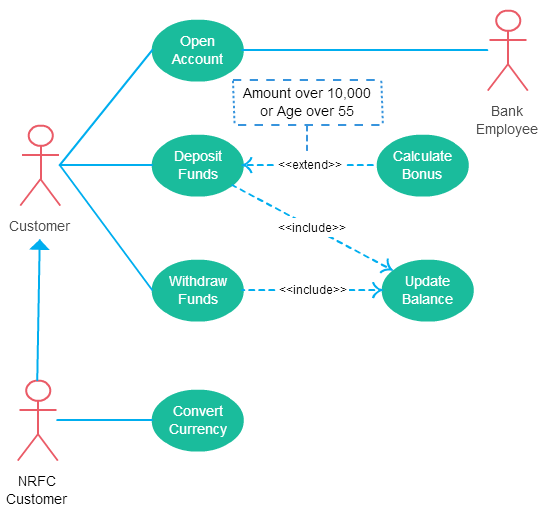
**Use-case図書く方法：**

* システムで実現したいことを考える
* ユースケースとアクターを考える
* システムの範囲を明確にして、ユースケース・アクターを再考する
* ユースケース・アクターの関係を明確にする
* UMLでユースケースを楕円に表記されています
  + ユースケース名は楕円の中、楕円の外と楕円を付けた長方形の中に書く方法があります
    - 何れも意味が同じ
* アクターはシステムを操作する役割を果たす、役割によってアクターは定義する
* システムの領域を境界線で表す



* 1つのユースケースは複数のユースケースを共通利用することが出来る、その関係を「包含/include」と「拡張/Extend」で表します
  + 共通利用するユースケース側から共通利用されるユースケース側へ線を引いて、<<include>>を付記する
  + 包含されるユースケースは必須となることに用いる
* 「拡張 / Extend」は条件付きで他のユースケースを利用することを示す
  + 拡張点：ユースケースを利用される条件を楕円を線で割って、下に「Extension points」を付けて記載される





* あるユースケースが他のユースケースを継承する場合、汎化・特化関係が発生する
* 機能を細かく分類することによってユースケースが特化して、共通機能をまとめることによってユースケースが汎化する
* アクターも同様に汎化関係を表現出来ます
  + 社員＜―一般社員
  + 社員＜―社長

**注意点：**

* 最初の段階で細かく・打ち所なく描く必要はありません
  + 分析・設計の段階でユースケース図を修正することもあります
* アクターが処理を中断する粒度で考える
* ユースケース図を動詞で表す
* CRUDのユースケースを「管理する」という風に定義する
  + 個別に書くと読みづらくなる