

# UML: Diagramme de cas d'utilisation

# Introduction

- ▶ Le diagramme de cas d'utilisation (ou use case) est un diagramme UML réalisé dans la phase d'analyse fonctionnelle du projet
- ▶ Le diagramme de cas d'utilisation représente les fonctions d'un système du point de vue de l'utilisateur (ou acteur)
- ▶ Un acteur n'est pas forcément un être humain, un autre système qui interagit avec notre système est également un acteur (exemple : si notre application contacte une banque pour valider le paiement la banque est considérée comme un acteur)
- ▶ Le diagramme de cas d'utilisation générale a pour but de représenter les principales fonctionnalités du système dans ce diagramme il ne sera pas représenté le détail de chaque fonctionnalité
- ▶ La réalisation d'un diagramme de cas d'utilisation permet ensuite par exemple la réalisation d'une maquette

# Logiciels et outils

- ▶ StarUML : <https://staruml.io/>
- ▶ PlantUML : <https://plantuml.com/fr/> (il existe une extension Vscode)

# Les concepts

- ▶ Le **système** : Il s'agit du notre projet, il est représenté par un grand carré
- ▶ Les **acteurs** : ils s'agit des différents type d'utilisateurs et les autres systèmes externes qui peuvent interagir le système, il sont représenté par des personnages
- ▶ Les **use-cases** (ou fonctionnalités) : il s'agit des différentes fonctionnalités dans notre système, ils sont représenté par des ovales

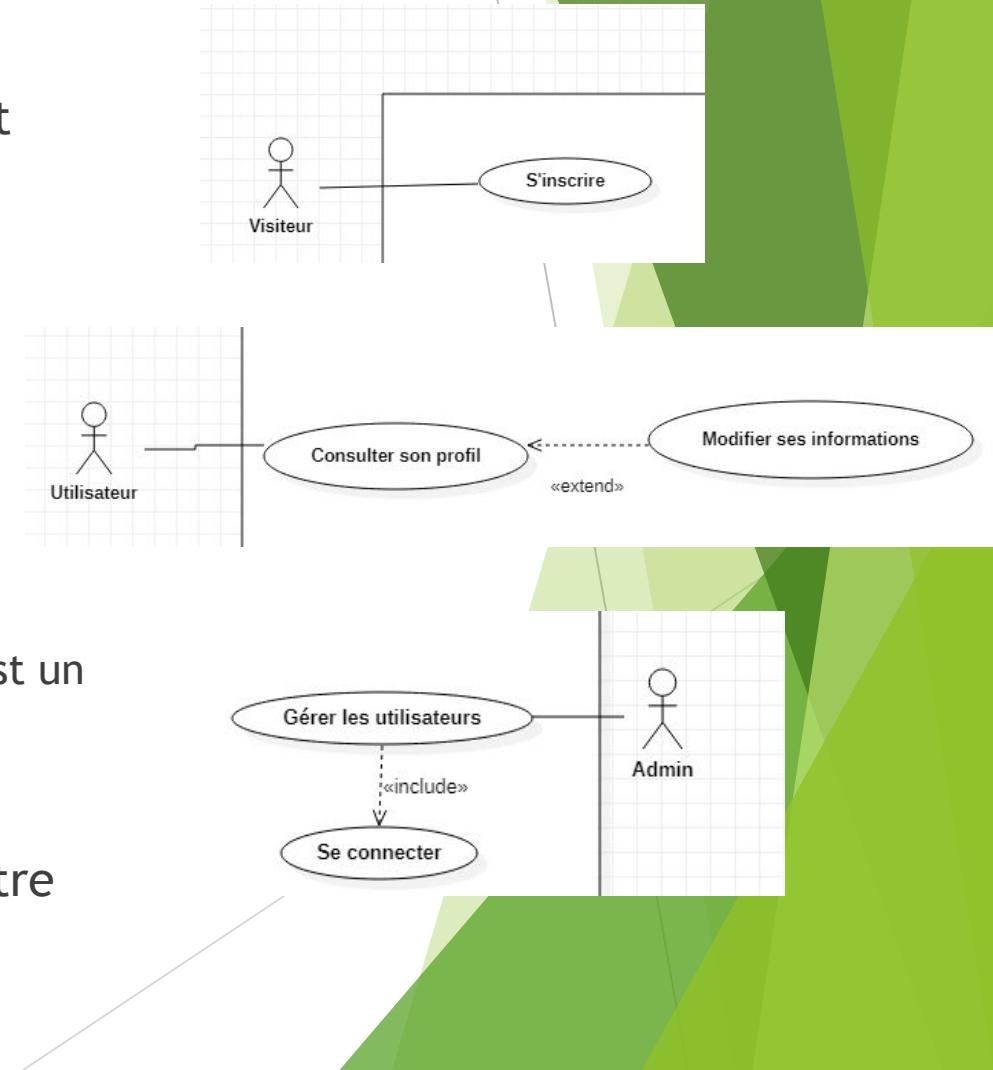


# Les concepts

- ▶ Les acteurs sont toujours placé à l'extérieur du système
- ▶ Les use-cases sont toujours placé dans le système
- ▶ De manière générale on considère que les acteurs situés à gauche du système sont les acteurs qui utilisent le front-office, les acteurs situés à droite sont des acteurs qui utilisent le back-office
- ▶ Le système porte en générale le nom du projet

# Les relations

- ▶ Les associations entre acteur et use case : représenté par un trait continu sans flèche, cette association indique qu'un acteur peut utiliser la fonctionnalité
- ▶ Les associations entre use-cases : représenté par une flèche en pointillé elles sont deux types :
  - ▶ Les associations extend : le use-case à l'origine du trait est une extension (une option) de l'use-case vers laquelle la flèche pointe
  - ▶ Les associations include : le use-case vers laquelle la flèche pointe est un prérequis de l'use-case à l'origine du trait
- ▶ Une relation d'héritage (ou généralisation) entre 2 acteurs ou entre 2 cas d'utilisations permet d'éviter les répétitions



# Liens

- ▶ [https://fr.wikiversity.org/wiki/Mod%C3%A9lisation\\_UML/Le\\_diagramme\\_de\\_cas\\_d%27utilisation](https://fr.wikiversity.org/wiki/Mod%C3%A9lisation_UML/Le_diagramme_de_cas_d%27utilisation)
- ▶ <https://www.ionos.fr/digitalguide/sites-internet/developpement-web/diagramme-de-cas-de-utilisation/>
- ▶ <https://www.lucidchart.com/pages/fr/diagramme-de-cas-dutilisation-uml>