

UML et POO

Xavier Nodet, xavier.nodet@gmail.com

Janvier 2022

Plan du cours

- Introduction
- UML
 - Étude fonctionnelle : acteurs et cas d'utilisation
 - Modélisation statique : classes et objets, attributs, opérations, etc
- Programmation orientée objet
 - Python, Java et C++



Comment le client a exprimé son besoin



Comment le chef de projet l'a compris



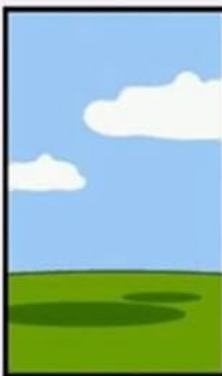
Comment l'ingénieur l'a conçu



Comment le programmeur l'a écrit



Comment le responsable des ventes l'a décrit



Comment le projet a été documenté



Ce qui a finalement été installé



Comment le client a été facturé



Comment la hotline répond aux demandes



Ce dont le client avait réellement besoin

Introduction

- Communication avec les clients
- Communication avec les developpeurs
- Textes et diagrammes UML
- Quelques notions de POO

Plan du cours

- Introduction
- UML
 - Étude fonctionnelle : acteurs et cas d'utilisation
 - Modélisation statique : classes et objets, attributs, opérations, etc
- Programmation orientée objet
 - Python, Java et C++

UML : les acteurs

UML : les acteurs

- Rôle joué par une entité
- Ne fait pas partie du système étudié
- Humain ou non

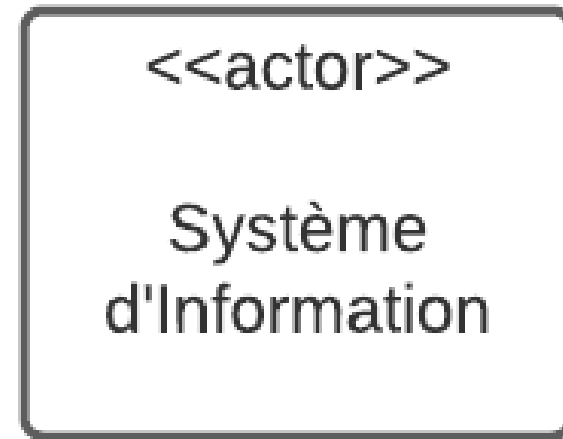
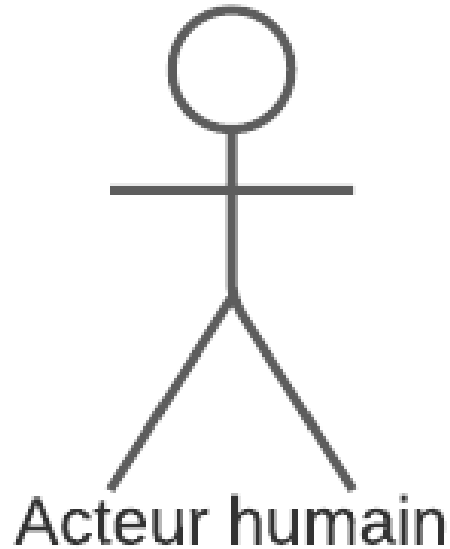
UML : les acteurs

- Rôle joué par une entité
- Ne fait pas partie du système étudié
- Humain ou non

Exemples:

- L'utilisateur d'une carte de paiement lors d'une transaction sur Internet.
- Le système de gestion des stocks, dans l'étude d'une caisse de supermarché.

UML : les acteurs





<https://forms.gle/z5gwPvn2E6UDomQZ9>

Cas d'utilisation

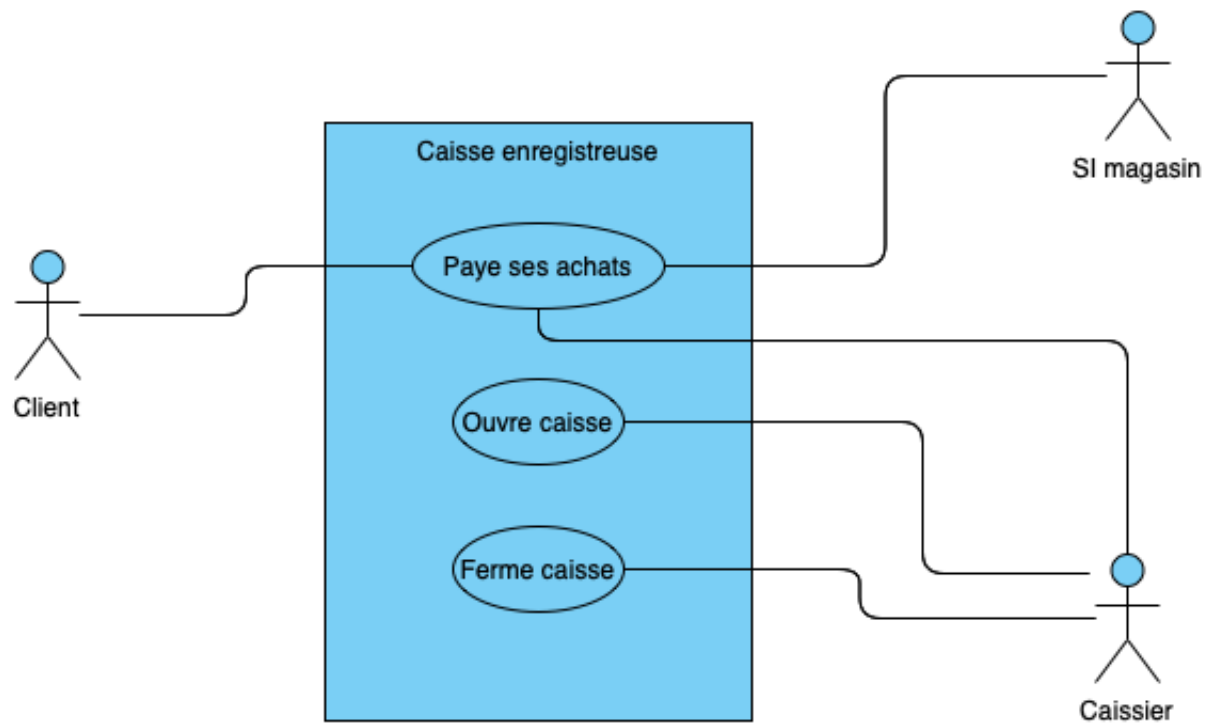
- Séquence d'événements au cours de laquelle l'acteur principal interagit avec le système
- Description du comportement attendu du système
- Description du *quoi*, et non pas du *comment*

Cas d'utilisation

- Scénario nominal
- Enchainements alternatifs :
 - Le porteur de carte fait une ou deux (mais pas trois) erreurs de code.
 - Le client présente sa carte de fidélité à la caisse
- Enchainements d'erreur :
 - Pas d'autorisation de retrait
 - Livre déjà réservé

Cas d'utilisation

- Pré-conditions, post-conditions
- Exigences non fonctionnelles

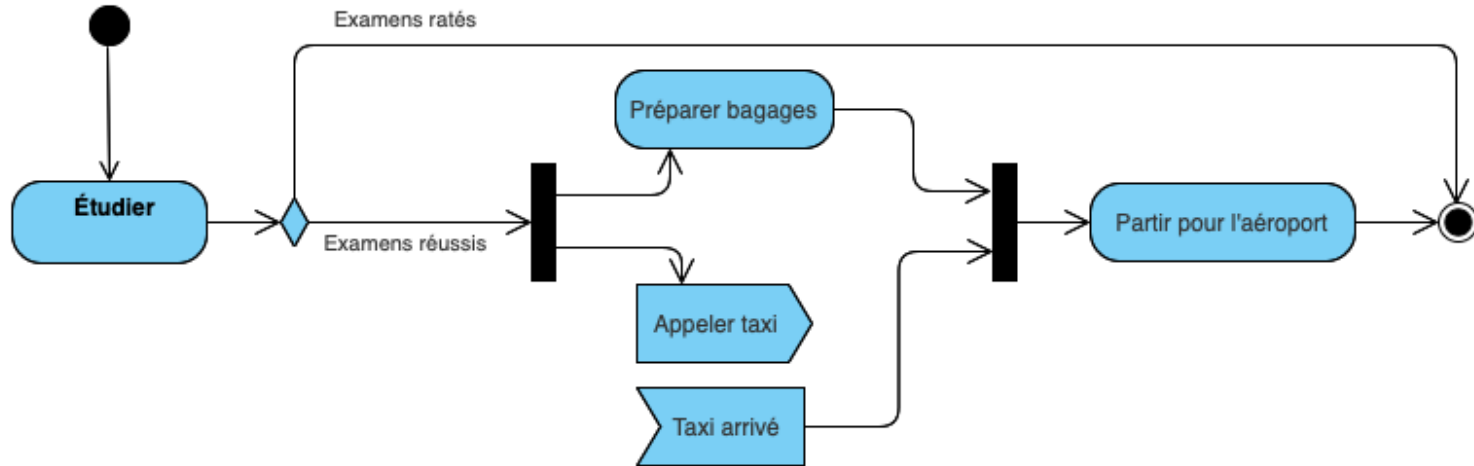




<https://forms.gle/6Ti8UEjQKR2H6c7G9>

Diagramme d'activité

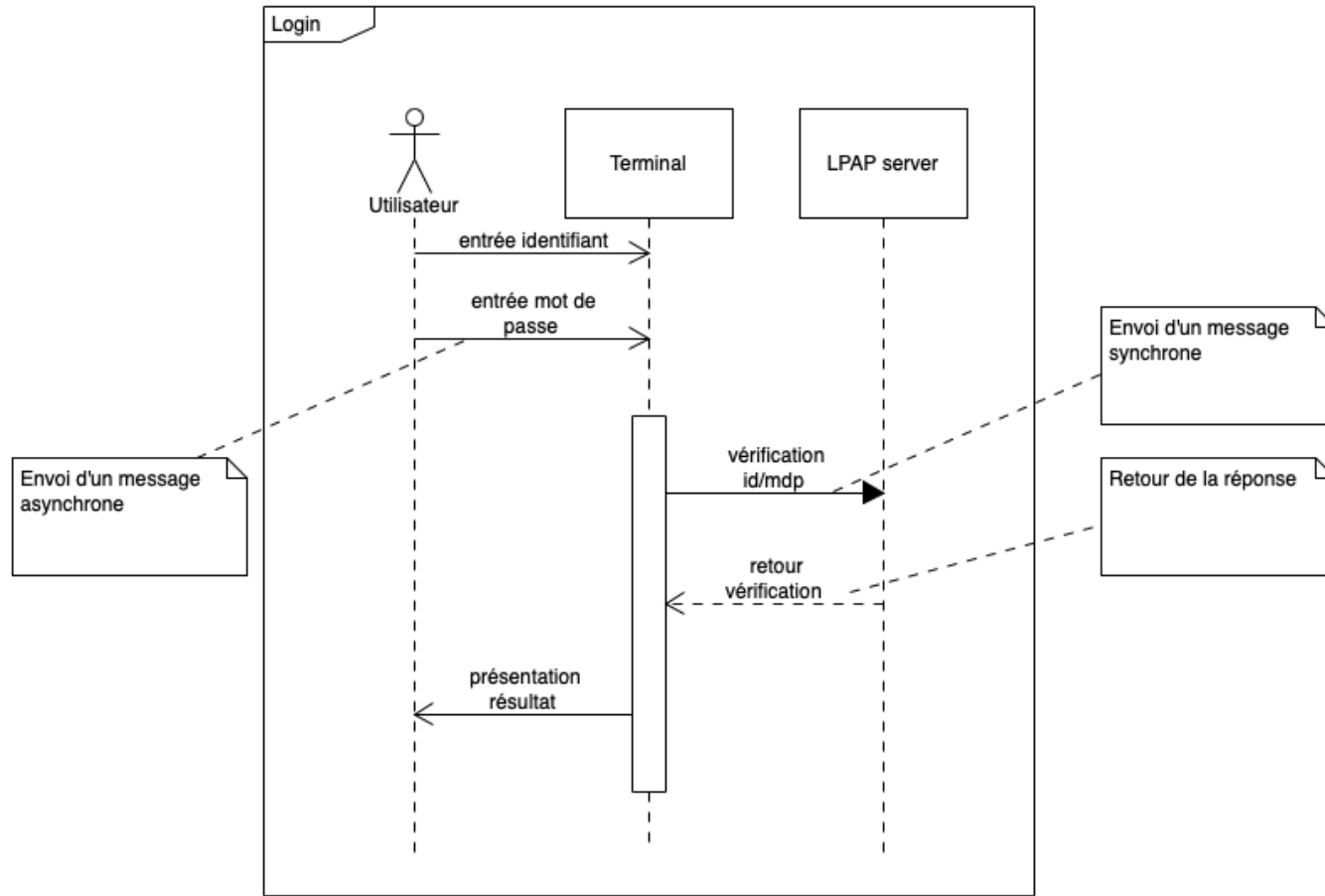
Visual Paradigm Online Free Edition

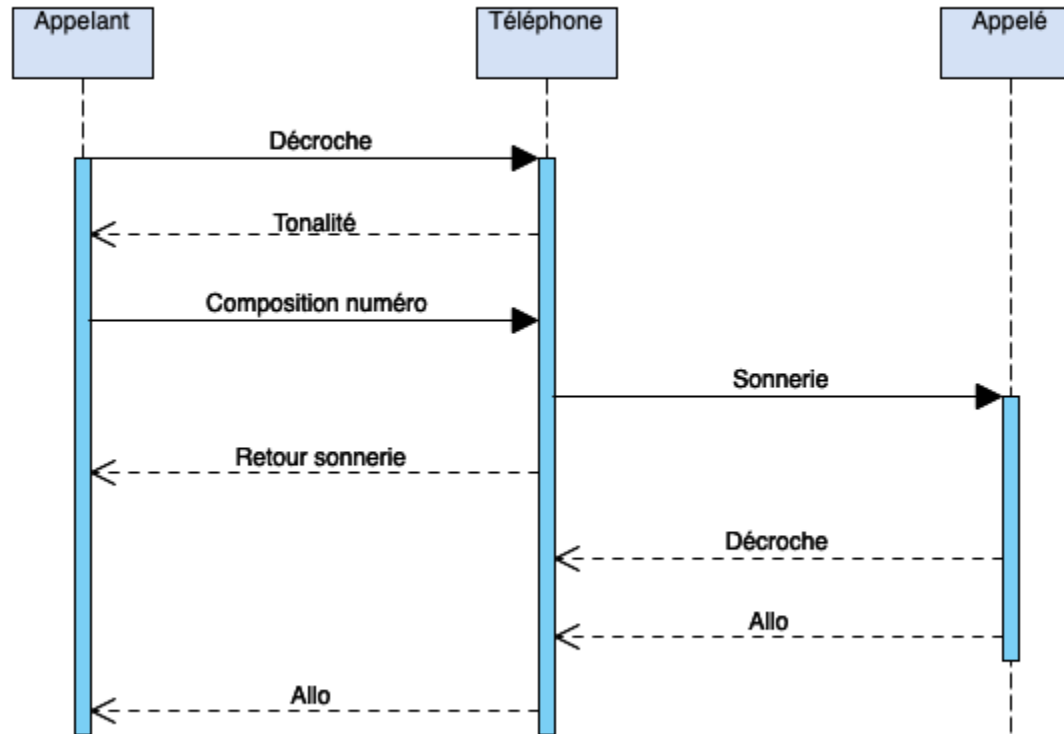


Visual Paradigm Online Free Edition

Diagramme de séquence

- Décrit les actions et messages échangés entre les acteurs
- Une *ligne de vie* verticale pour chaque acteur
- Messages synchrones ou asynchrones





Plan du cours

- Introduction
- UML
 - Étude fonctionnelle : acteurs et cas d'utilisation
 - Modélisation statique : classes et objets, attributs, opérations, etc
- Programmation orientée objet
 - Python, Java et C++