

Programmation Orientée Objet en Python

#5 UML (class diagram)

par David Albert

Table des matières

01 Cycles de développement

Cycle en V et méthodes AGILE.

02 Introduction à UML

Motivations. Diagrammes. Chaîne de conception.

03 Diagramme de cas d'utilisation

Quelques exemples.

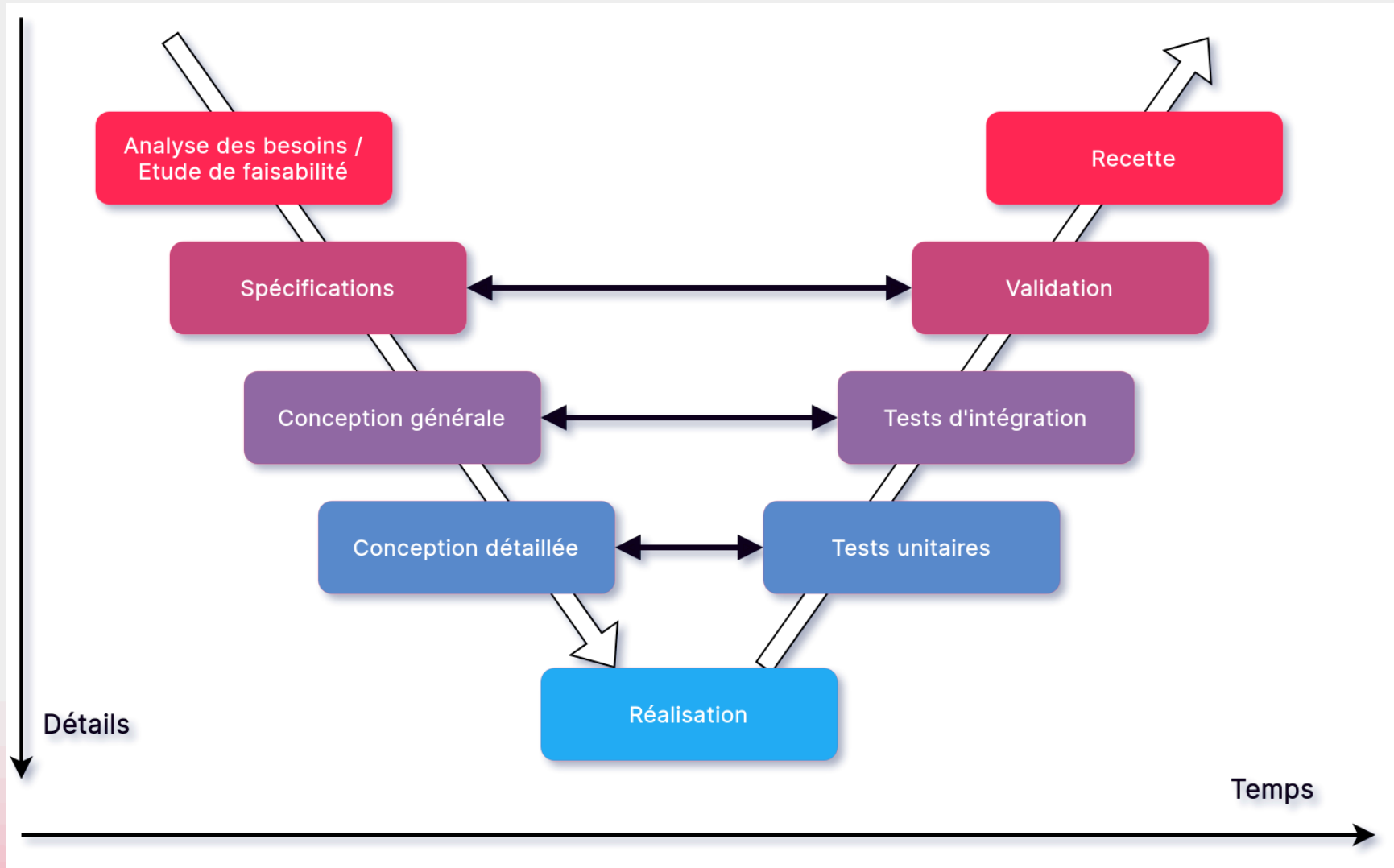
04 Diagramme de classes

Typing. Documentation. Gestion des erreurs. Tests unitaires.

01

Cycles de développement

Cycle en V



Vers une méthodologie **AGILE**



Le **cycle en V** a un **inconvenient majeur**. La vérification de la conformité aux besoins client attend la fin du développement du produit. S'il y a un soucis, on s'en rend compte **très tardivement**.

Méthologie AGILE

Pour pallier à cela, les entreprises privilégient de plus en plus des **cycles courts** et successifs. On répétera successivement les étapes de *spécifications, conception, développement, test et validation*.



02

Introduction à UML

Unified Modeling Language

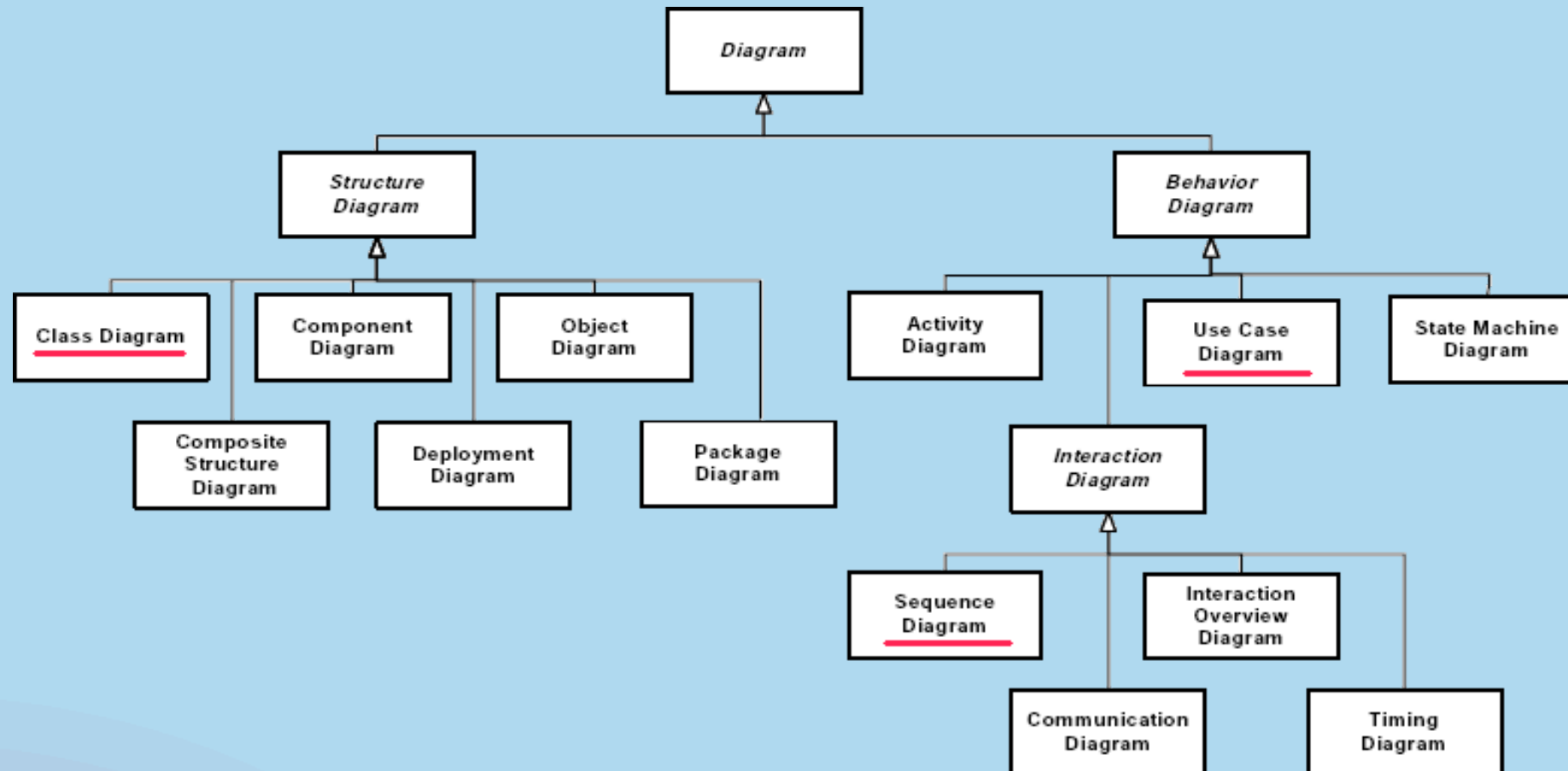
Motivations

- Besoin de conception pour réaliser une architecture complexe.
- Besoin de se comprendre

UML c'est quoi ?

- un langage de modélisation de systèmes informatiques
- modèle graphique (à base de pictogrammes)
- indépendant du langage de programmation
- intervient dans la phase de conception (générale et détaillée)

Fun fact : UML est décrit en UML.



Quelques diagrammes

Diagrammes structurels

Diagramme de classes

Définit l'ensemble des classes et de leurs relations

Diagramme de composants

Liste les composants logiciels

Diagramme de déploiement

Définit la répartition des composants sur une architecture matérielle

Diagrammes de comportement

Diagramme des cas d'utilisation

Définit les scénarios d'interaction entre les utilisateurs et le système

Diagramme d'activité

Représente les états du système et leurs transitions par événements

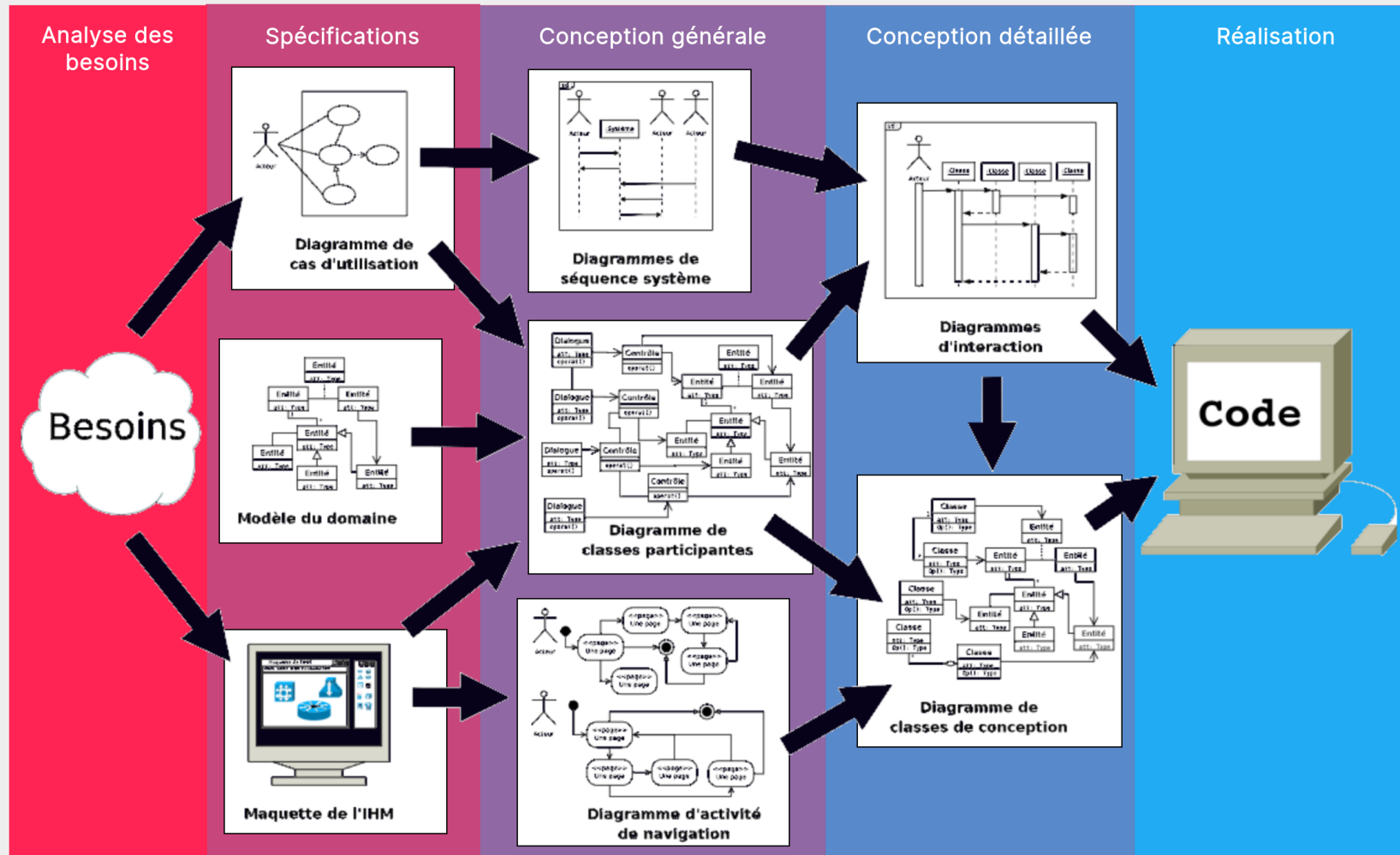
Diagramme de séquence

Représente les scénarios d'interactions entre entités du système

Référence: [Laurent Verouter, Cours UML, Insa Rouen](#)

Chaîne de conception

Différents diagrammes arrivent à différents moments dans la chaîne de conception.



Référence:
[Laurent Audibert](#)

03

Diagramme de cas d'utilisation

Etude de cas

Gestion d'un restaurant

On souhaite réaliser une application pour un restaurant qui lui permettra de gérer les réservations et les commandes de ses clients.

i

On utilisera cet exemple comme fil rouge tout au long de ce cours.

Enoncé détaillé

Le restaurant accueille des clients décrits par un nom, un email et un numéro de téléphone. Le restaurant est ouvert tous les jours de 19h et 23h30 durant lequel il peut accueillir au maximum 20 clients par service (1 service dure 1h30). Les clients peuvent réserver une table sur ces créneaux. S'il n'y a plus de place, ils peuvent également commander leur repas et payer via l'application et venir le récupérer dans la foulée. Sur place, un serveur s'occupera de des commandes des clients et de la paie.

Diagramme de cas d'utilisation

Résumé

Objectifs

- Premier diagramme réalisé pour définir les scénarios d'usage
- A réaliser avec le client
- À utiliser tout au long du développement

Exemple

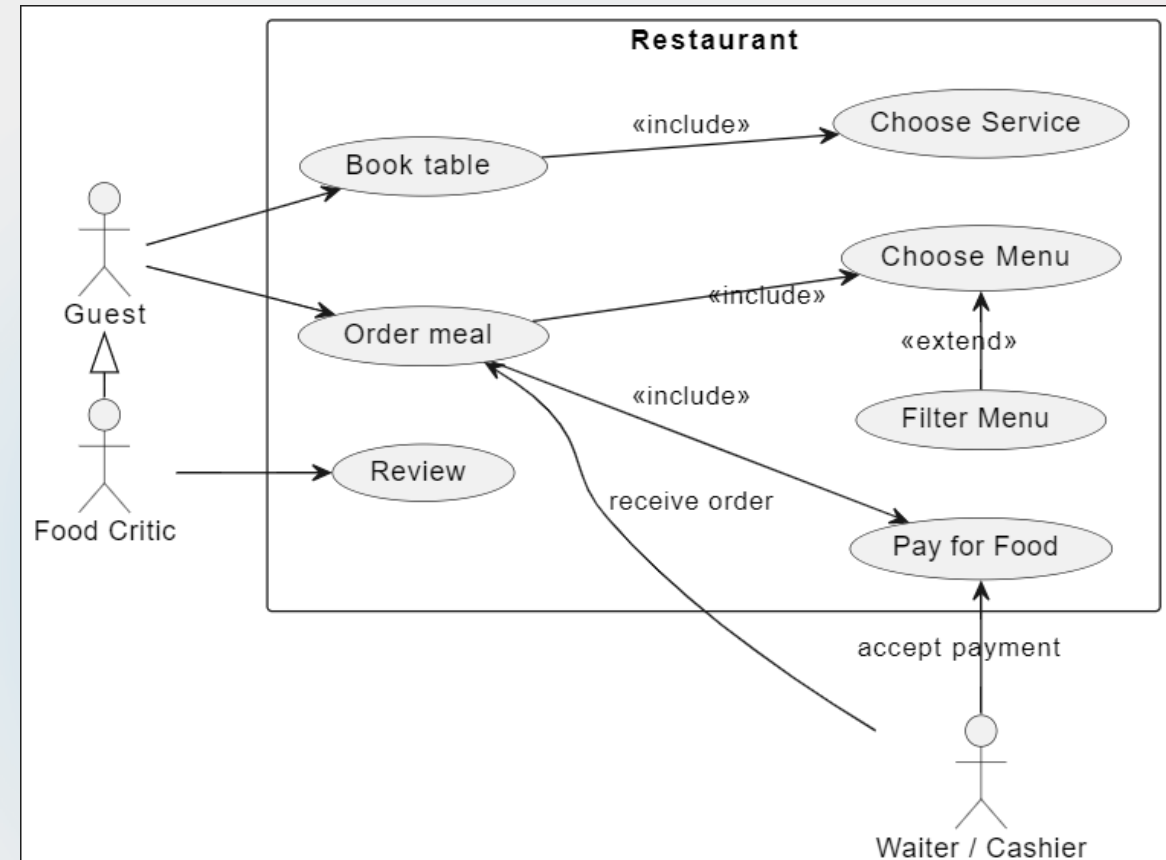
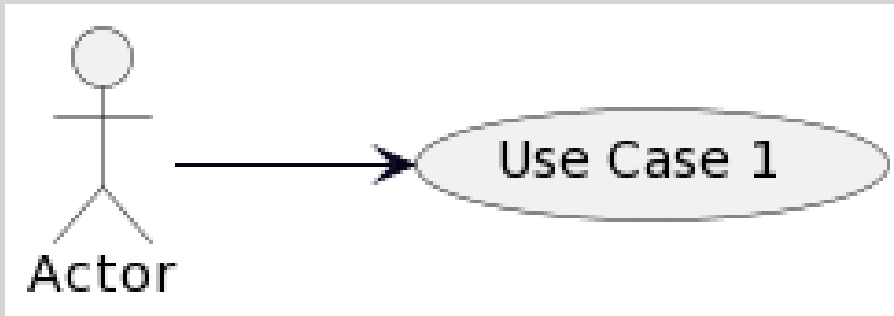


Diagramme de cas d'utilisation

Syntaxe



Déclenchement



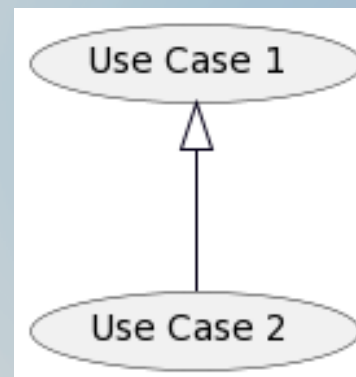
Prolongement



Pré-requis



Héritage



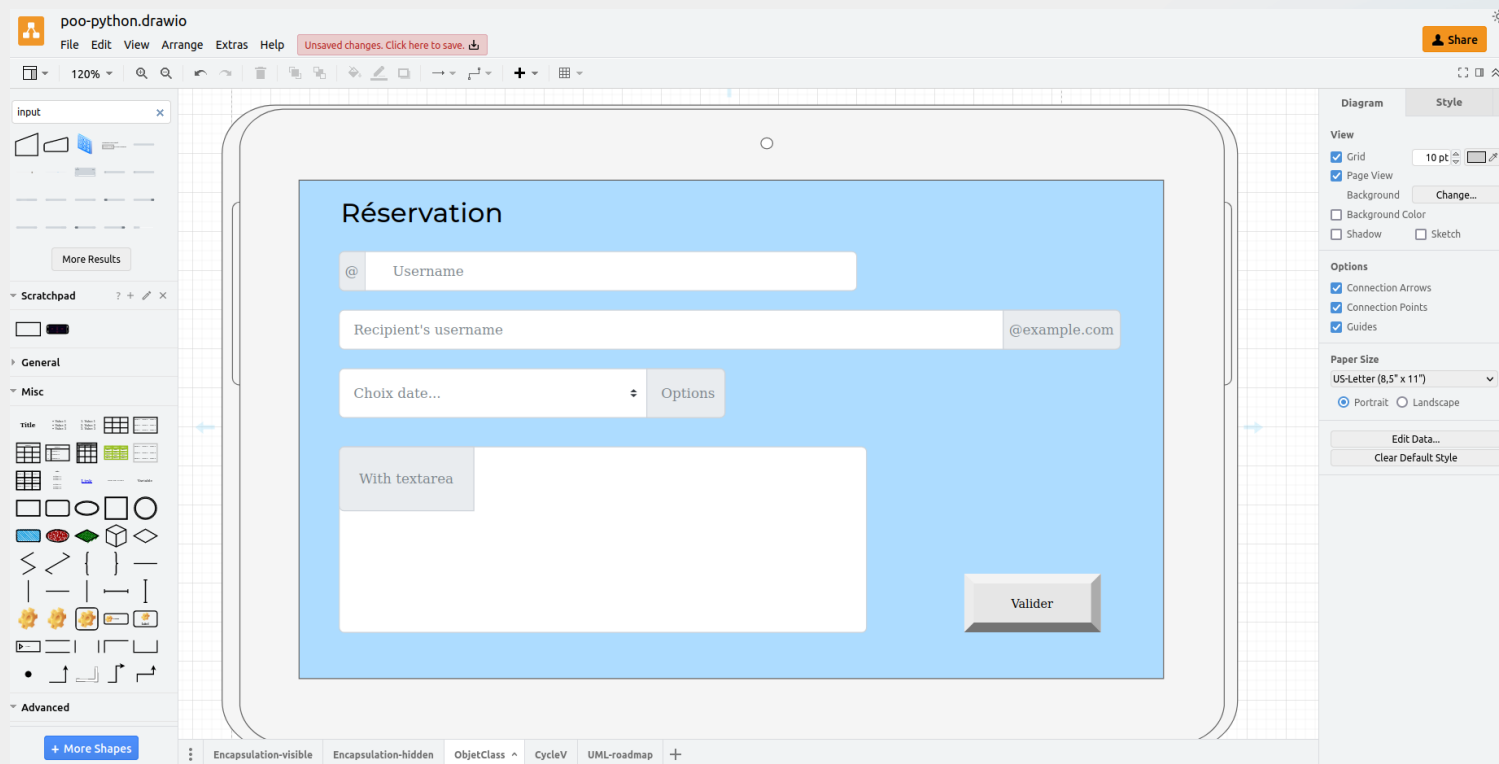
04

Maquettes

Réaliser les premières maquettes

Rien de tel que quelques maquettes pour mettre tout le monde d'accord sur l'interface homme-machine et ses interactions.

Outils: [drawio](#) et [Figma](#)



05

Diagrammes de classes

Diagramme de classes

Syntaxe 1

Classe

+ : attributs **publics**

- : attributs **privés**

ClassName

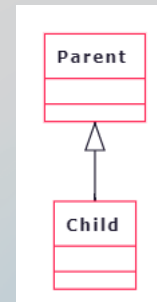
attributs

methods()

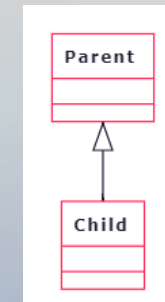
+methodAbstraite2()

+methodAbstraite()

Héritage



Association



Association

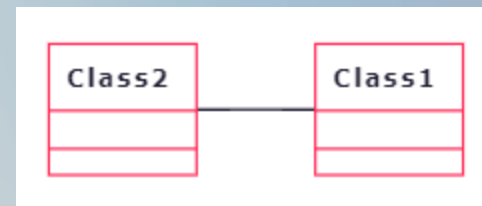
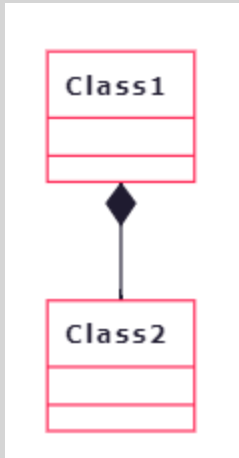


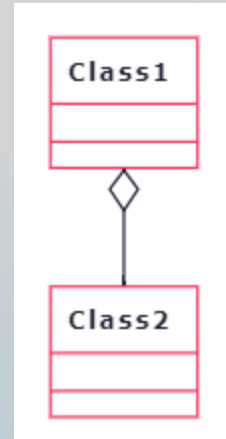
Diagramme de classes

Syntaxe 2

Composition



Aggrégation



Modèle du domaine

Phase

Début de conception générale. Intervient juste après les premières maquettes et cas d'utilisation.

Objectifs

- Premier diagramme de classes à réaliser
- Indépendant des fonctionnels de l'application
- Représente le domaine métier

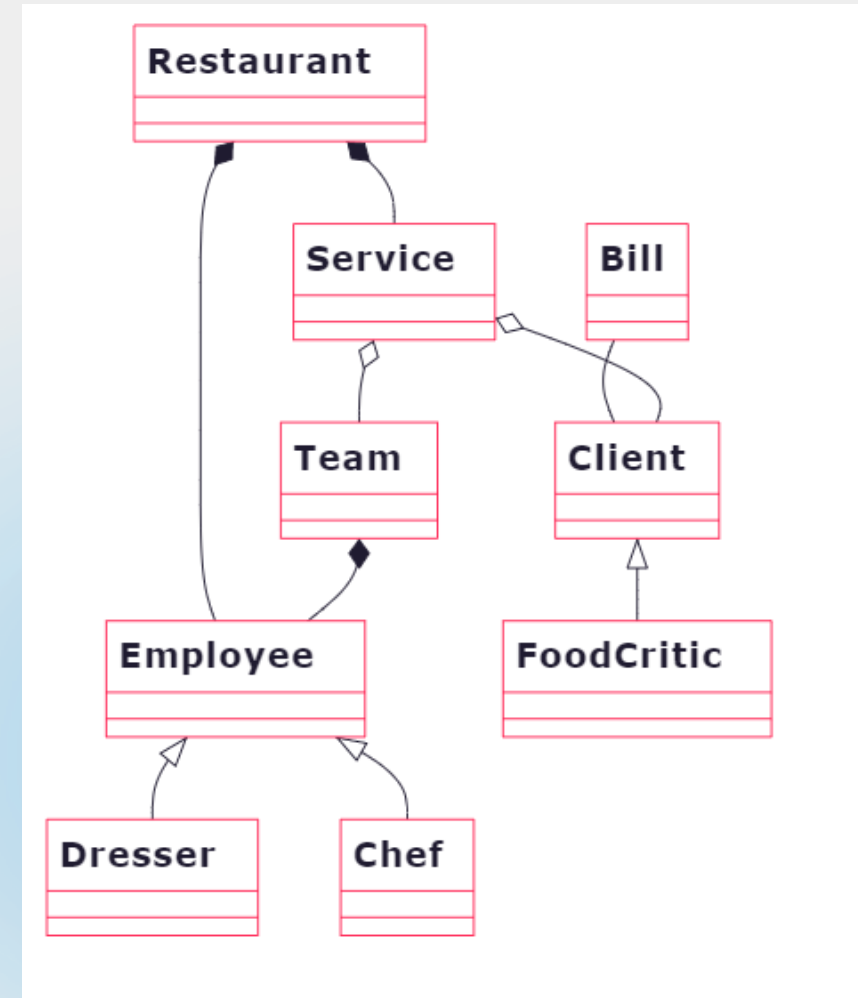


Diagramme de classes participantes

Phase

Fin de conception générale. Intervient dans la dernière phase de la conception générale en même temps que les diagrammes de séquence et d'activité.

Objectifs

- Enrichissement du modèle de domaine
- Modélisation guidée par les besoins

06

Diagramme d'activité

07

Diagrammes de séquence