

# La modélisation avec UML

---

\*\*\*

- Qu'est ce que UML
- Installer MySQL
- Diagramme de cas d'utilisation
- Diagramme des classes
- Diagramme de séquences

UML est l'acronyme de Unified Modeling Language qui signifie Langage de modélisation unifié. C'est un langage qui en utilisant des diagrammes, permet de donner une vision sur un projet sans entrer en profondeur dans les notions techniques. Il permet d'unifier la communication entre développeurs et non-développeur et permet surtout d'avoir une meilleure visibilité sur le projet avant d'écrire les premières lignes de code.

# Les différents type de diagrammes

UML utilise un langage basé sur des diagrammes.

UML propose 13 diagrammes officiels. Ces diagrammes permettent d'avoir un aperçu structurel, comportemental et structurel d'un projet.

Nous ne ferons pas le tour des 13 diagrammes. Nous allons nous limiter à trois diagrammes qui sont suffisants pour concevoir un site Web:

- diagramme de classes
- diagramme de cas d'utilisation
- diagramme de séquences

# Diagramme de cas d'utilisation:

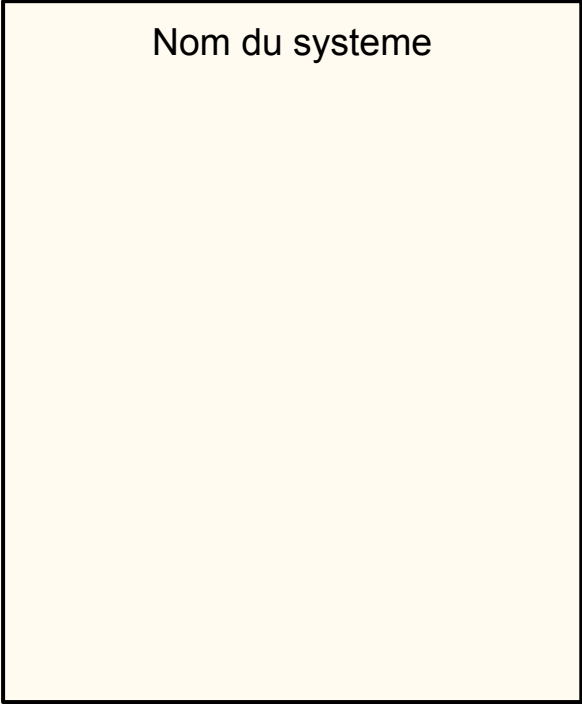
Ce diagramme regroupe les fonctionnalités du site web et les interactions avec les différents utilisateurs.

Il est composé de quatre graphiques:

- le système
- les acteurs
- les cas d'utilisation
- les relations.

# Diagramme de cas d'utilisation: le système

Le système dans un diagramme de cas d'utilisation représente la plateforme sur lequel on doit faire le diagramme. Cela peut représenter une application complète ou un composant d'une application. Dans le diagramme le système est représenté par un rectangle avec le nom du système précisé en en-tête du rectangle.



Nom du systeme

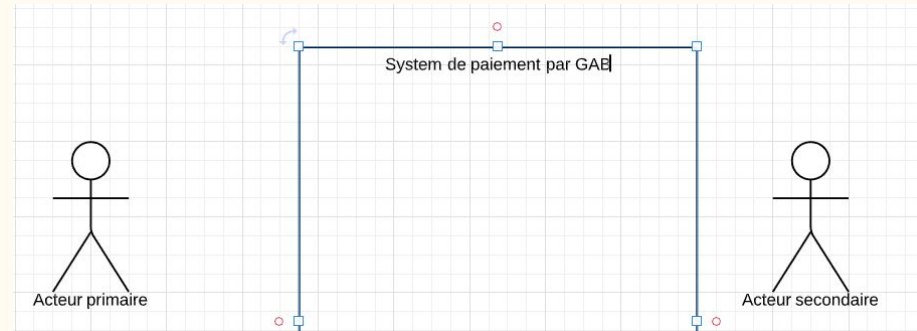
# Diagramme de cas d'utilisation: les acteurs

Un acteur représente quelqu'un ou quelque chose qui utilise le système sur lequel nous travaillons. Cela peut être un utilisateur une organisation ou un autre dispositif externe. Dans le diagramme les acteurs sont représentés avec un bonhomme en bâton placé à l'extérieur du système avec un nom précise en dessous.

Les acteurs peuvent être regroupés en deux groupes:

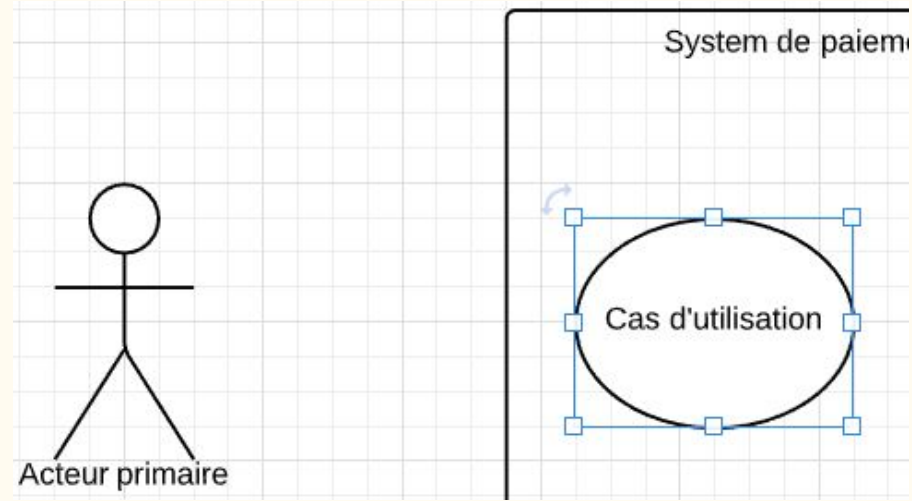
- l'acteur primaire qui va réaliser la première action sur le système.
- Et un acteur secondaire qui va agir en fonction d'une action d'un autre acteur ou d'une opération dans le système.

Les acteurs primaires sont à gauche du système les acteurs secondaires à droite.



# Diagramme de cas d'utilisation: les cas d'utilisations

Les cas d'utilisation sont les actions réalisées par les acteurs. Ils sont représentés sous forme d'ellipses qui sont placées dans le rectangle du système. Dans chaque ellipse on mettra le nom de l'action.





# Diagramme de cas d'utilisation: les relations

Les relations permettent de lier les acteurs aux cas d'utilisation. Il existe quatre types de relations:

- **l'association:** représenté par un trait plein qui relie l'acteur à un cas d'utilisation.
- **l'inclusion:** représente une relation de dépendance. Exemple une action d'authentification implique l'action de vérification de mot de passe. Pour tracer une relation d'inclusion on utilise une flèche en pointillés qui va du cas de base vers le cas inclus. Et on ajoute le libellé 'inclus' sur la flèche.

- **l'extension:** cette relation implique deux actions dont l'exécution du premier implique **occasionnellement** l'exécution du second. Exemple, l'action “*erreur d'une authentification*” n'est exécuté que si les données rentrées sont erronées. Cette relation est représentée par une flèche en pointillés partant du cas d'utilisation d'extension vers le cas d'utilisation de base.
- **la généralisation:** Elle permet de déterminer les différentes options qui peuvent être liées à un cas d'utilisation. Exemple dans un cas de paiement en ligne, nous pouvons avoir les options de payer avec une carte de crédit, avec Paypal ou avec virement bancaire.

# Diagramme des classes

Ce diagramme est le seul obligatoire dans une étude de projet. IL est utilisé pour concevoir les différentes tables de la base de données. En programmation orientée objet, ses composants représentent les classes de l'application. Le diagramme de classes est représenté par un rectangle composé de trois sections:

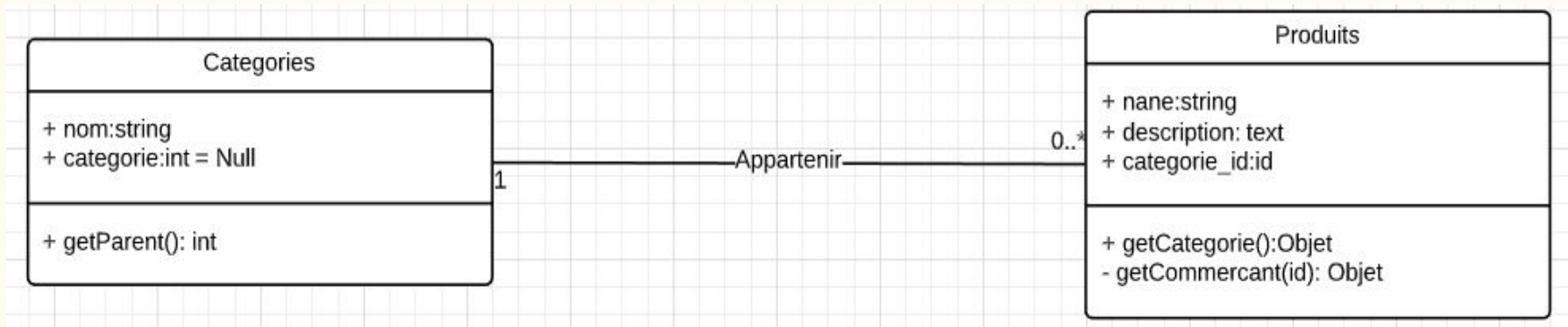
- section du nom qui renseigne sur le nom de la class
- section des attributs: cette section doit contenir les propriétés de la classe. Il s'agit des propriétés obligatoires ou optionnelles.
- section des méthodes: cette section va renseigner sur les méthodes de la classe.

Les deux dernières sections sont optionnelles.

# Diagramme des classes

Ces diagrammes sont liés par des relations souvent appelées associations. Ces relations décrivent les connexions structurelles entre les instances. Ils sont souvent représentés par une ligne droite. À l'extrémité des lignes on précise les cardinalités. Les cardinalités représentent la multiplicité associée à une terminaison d'association. Voici quelques exemples de multiplicité:

- exactement un : 1 ;
- plusieurs : n ou 0..\* ;
- au moins un : 1..\* ;
- de un à six : 1..6.



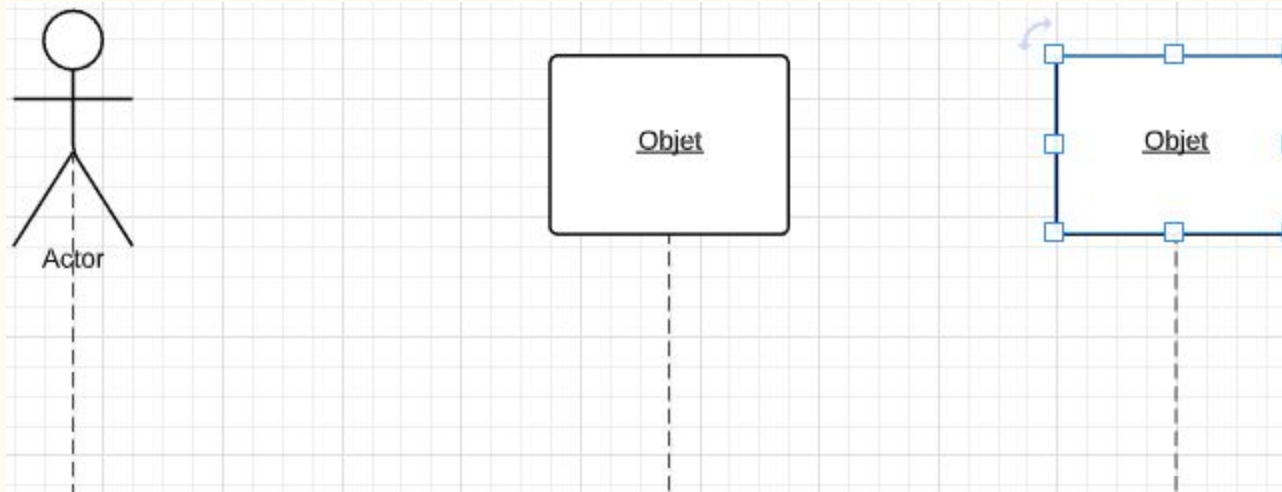
# Diagramme de séquences:

Ce diagramme va décrire les scénarios d'interactions entre les éléments d'un système ou les classes d'une application. Il peut s'agir de scénario d'une fonctionnalité en particulier ou de toute l'application. Il affiche les interactions de manière chronologique. Pour construire un diagramme de séquence il faut commencer par définir les différents composants qui vont être impliqués dans le système. Ces composants sont nommés objets. Il faut aussi définir les acteurs qui sont les services ou personnes qui vont interagir avec le système. Les objets et les acteurs sont liés par des messages qui vont décrire les interactions au sein du système. Les échanges de messages sont représentés de manière chronologique de haut en bas.

# Diagramme de séquences:

Un acteur est représenté par un bonhomme en bâton et les objets sont représentés par des rectangles avec le nom de l'objet et sont positionnés sur la même ligne que l'acteur. Les objets sont positionnés dans l'ordre de leur utilisation.

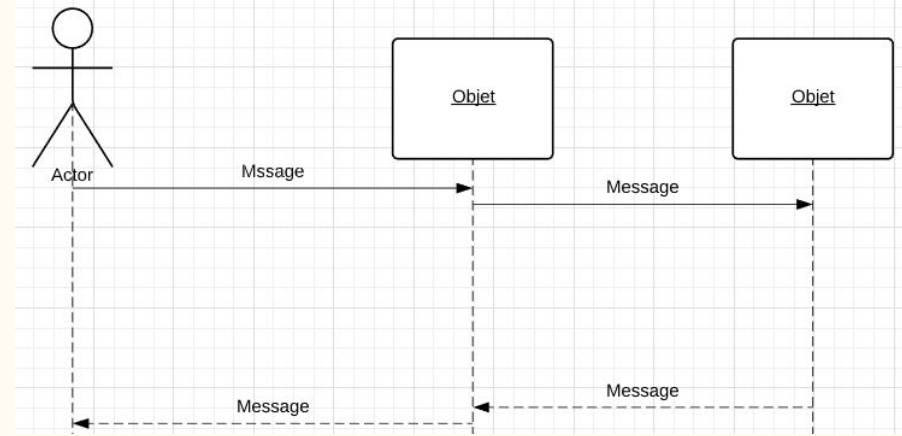
En dessous des acteurs et des objets nous allons par la suite dessiner une ligne verticale en pointillés qui représentent le temps d'exécution de l'objet.



# Diagramme de séquences:

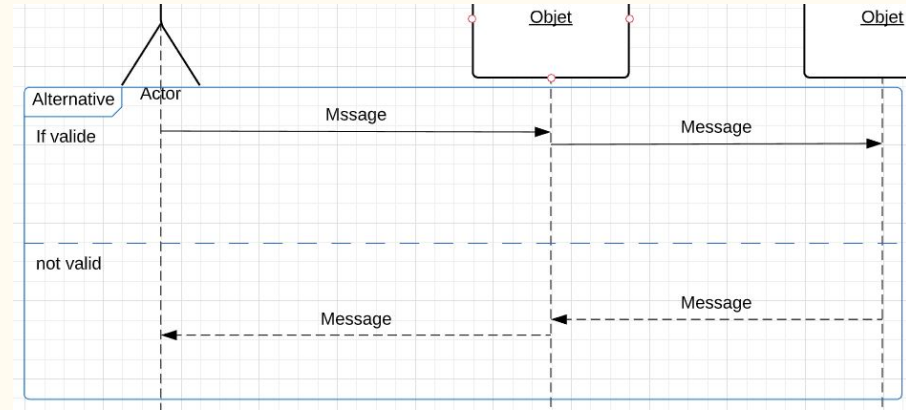
Pour représenter les actions nous utilisons des messages. Les messages sont représentés sous deux formats.

- Les messages avec flèche pleine. Ceux-là représentent des actions dans le système. Ils sont souvent des messages qui exécutent une requête et qui contiennent des données.
- Les messages avec flèche en pointillés: ceux-là représentent une réponse à une action.



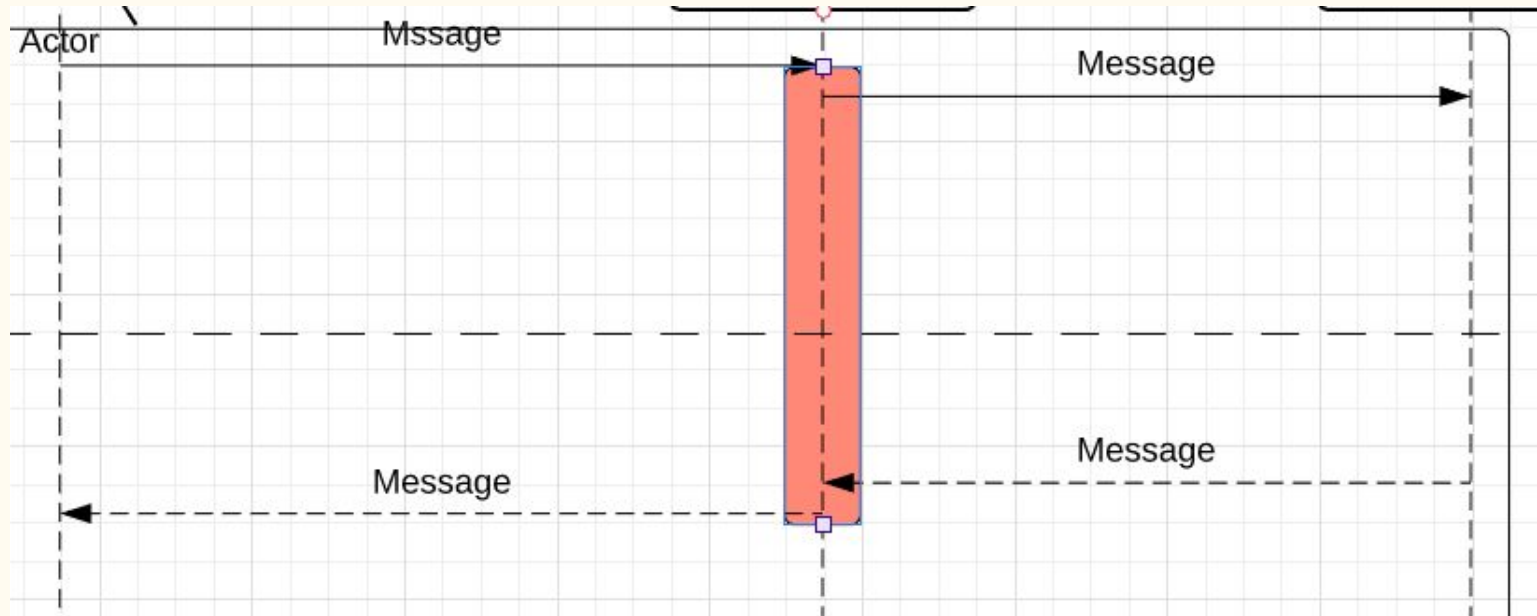
# Diagramme de séquences:

Nous pouvons regrouper les actions dans des conditions appelées alternatives options ou alt. Un alt est représenté par un rectangle ayant deux zones. La zone du haut regroupe les actions quand la condition est vraie et la zone du bas représente le cas où la condition est fausse.



## Diagramme de séquences: action box

Après avoir affiché toutes les interactions nous pouvons ajouter un autre élément qui va représenter la durée d'exécution des actions. Cet élément appelé action box est un rectangle vertical qui s'étend du premier message au dernier message d'une action.





# Liens utiles

- ❖ <https://openclassrooms.com/fr/courses/918836-concevez-votre-site-web-avec-php-et-mysql/913655-quest-ce-quune-base-de-donnees>
- ❖ <https://phpdelusions.net/pdo>
- ❖ <https://www.digitalocean.com/community/tutorials/how-to-install-and-secure-phpmyadmin-on-ubuntu-18-04>