Apprentissage de la COO par une mise en pratique sur une application de gestion de Quiz dans l'esprit de PS6

[**L'étude de cas est ici**](https://unice-my.sharepoint.com/:w:/g/personal/mireille_blay_unice_fr/EbdBmcOigzVOn9vIdwGpmIYBwcJDqipvdYibQ0eQ19a1Tg?e=VzqZFD)

**👉 À Faire en équipe**

Vous pouvez faire vos schémas, soit sur papier, soit utiliser au choix [un éditeur de diagramme UML](https://lms.univ-cotedazur.fr/2022/course/view.php?id=17306&section=2#tabs-tree-start) (ne pas utiliser des outils de dessin, ce n'est pas du dessin, l'outil doit au moins vous permettre de vérifier que vous n'inventez pas une nouvelle syntaxe).

**Légende :**

Les questions sont notées par : **✍️**

Les textes surlignés sont des informations sur les diagrammes UML. Ils contiennent en général des références vers la source d'information.

Les textes surlignés en jaune visent à faire le lien entre les approches de modélisation des interfaces et la modélisation du système en UML.

**Ce travail sera à rendre.**

[**👉 À Faire en équipe** 1](#_Toc126440341)

[1. Diagramme de cas d'utilisation 2](#_Toc126440342)

[1.a. Acteurs 2](#_Toc126440343)

[1.b. Cas d'utilisation 2](#_Toc126440344)

[1.c. Relations entre Cas d'utilisation 3](#_Toc126440345)

[2. Diagrammes de séquence 4](#_Toc126440346)

[3. Diagrammes de classes 5](#_Toc126440347)

# 1. Diagramme de cas d'utilisation

## 1.a. Acteurs

Un [**acteur**](https://www.ibm.com/docs/fr/rational-soft-arch/9.5?topic=diagrams-actors) représente un rôle d'un utilisateur qui interagit avec le système que vous modélisez. L'utilisateur peut être un utilisateur humain, une organisation, une machine ou un autre système externe. Vous pouvez représenter plusieurs utilisateurs avec un seul acteur et un seul utilisateur peut avoir le rôle de plusieurs acteurs.  Un **acteur principal** obtient un résultat observable du système tandis qu'un **acteur secondaire**est sollicité pour des informations complémentaires. En général, l'acteur principal initie le cas d'utilisation par ses sollicitations

[**Personas are different of actors**](https://www.agilemodeling.com/artifacts/personas.htm#:~:text=In%20other%20words%2C%20personas%20represent,with%20respect%20to%20your%20system.) because they describe an archetypical instance of an actor. In a use case model we would have a Customer actor, yet with personas we would instead describe several different types of customers to help bring the idea to life.

**✍️  1. Identifiez les acteurs** qui interagissent avec le système en distinguant ceux qui sont primaires et ceux qui sont secondaires.  Dessinez-les.

## 1.b. Cas d'utilisation

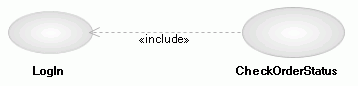
Un [**cas d'utilisation**](https://www.ibm.com/docs/fr/rational-soft-arch/9.5?topic=diagrams-use-cases) décrit une fonction qu'un système exécute pour atteindre l'objectif de l'utilisateur. Un cas d'utilisation doit renvoyer un résultat observable qui est **utile pour l'utilisateur du système.**

✍️ **1. Identifiez les cas d'utilisation.**Pour des raisons de temps, nous ne vous demandons pas de les décrire.

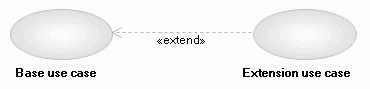
**✍️ 2. Associer cas d'utilisation et acteurs**. Tout acteur est associé à, au moins, un cas d'utilisation (sinon soit le système ne fait rien pour cet acteur, soit nous ne l'utilisons pas, donc ce n'est pas un acteur, cf. définition).  De même, tout cas d'utilisation est associé à, au moins, un acteur...

## 1.c. Relations entre Cas d'utilisation

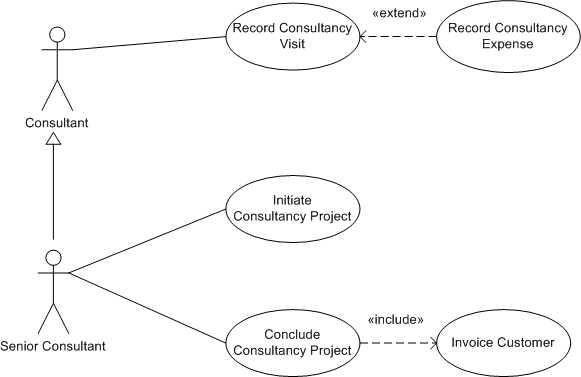
Une [**relation d'inclusion**](https://www.ibm.com/docs/fr/rational-soft-arch/9.5?topic=diagrams-include-relationships) est une relation dans laquelle un cas d'utilisation (le cas d'utilisation de base) inclut les fonctionnalités d'un autre cas d'utilisation (le cas d'utilisation inclus). Dans l'exemple ci-dessous, une relation d'inclusion est dirigée du cas d'utilisation CheckOrderStatus vers le cas d'utilisation LogIn pour indiquer que le cas d'utilisation CheckOrderStatus inclut toujours les comportements figurant dans le cas d'utilisation LogIn.



Vous pouvez utiliser une [**relation d'extension**](https://www.ibm.com/docs/fr/rational-soft-arch/9.5?topic=diagrams-extend-relationships) pour spécifier qu'un cas d'utilisation (l'extension) étend le comportement d'un autre cas d'utilisation (la base).

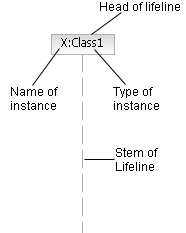


Une relation de généralisation est une relation dans laquelle un élément de modèle (l'enfant) est basé sur un autre élément de modèle (le parent). L'enfant spécialise le parent. Entre acteurs, cela signifie que l'enfant peut faire tout ce que le parent fait. Entre cas d'utilisation, cela signifie que le parent est une généralisation des cas d'utilisation enfants qui eux correspondent à des spécialisations du cas d'utilisation parent.

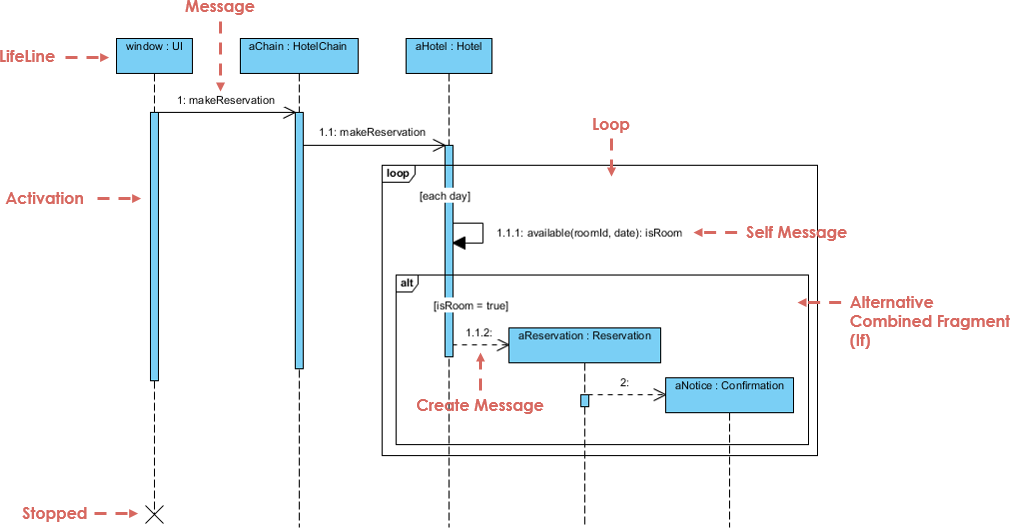
[](https://geeksww.com/tutorials/miscellaneous/uml/resources/making_use_case_diagram_using_generalization_in_uml.php)  
  
  
✍️**1. Identifiez les relations**d'inclusion, *généralisation* ou *extends* qui s'appliquent dans votre diagramme de cas d'utilisation et incluez-les dans le schéma.

# 2. Diagrammes de séquence

Un [**diagramme de séquence**](https://www.ibm.com/docs/fr/rational-soft-arch/9.5?topic=diagrams-creating-sequence) représente la séquence de messages entre les objets au cours d'une interaction. Un diagramme de séquence porte **au maximum sur un cas d'utilisation**. Plusieurs diagrammes de séquences sont généralement nécessaires pour couvrir un cas d'utilisation.   
 Un diagramme de séquence comprend un groupe d'objets, représentés par des lignes de vie, et les messages que ces objets échangent lors de l'interaction.   
[**Les lignes de vie**](https://www.ibm.com/docs/fr/rational-soft-arch/9.5?topic=diagrams-lifelines-in-uml) représentent les objets entrant dans une interaction. Les messages, envoyés et reçus par l'instance, apparaissent sur la ligne de vie dans un ordre chronologique. On doit retrouver les méthodes correspondantes dans les classes  qui reçoivent les messages.



Un [**exemple de diagramme de séquence**](https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-sequence-diagram/) annoté :



**Scenarios and sequence diagrams** are effective models for specifying the behaviors of target systems.

**(Scenario d’usage) Scenarios** can be used for requirements elicitation in requirements definition; they describe the scenario in the form of a text, eventually structured.

**Sequence diagrams** can be used for interactions between a system user and the system and between objects; they describe the scenario in a standardized graphical notation. Sequence diagrams help determine the classes in your system and their interrelationships.

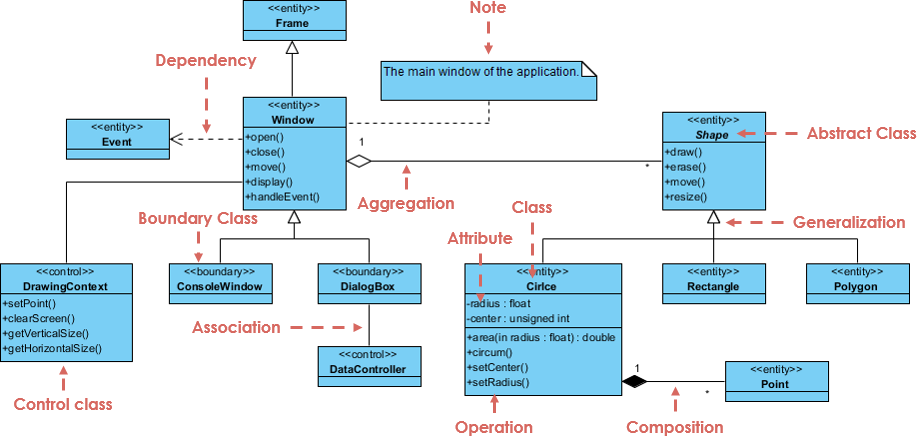
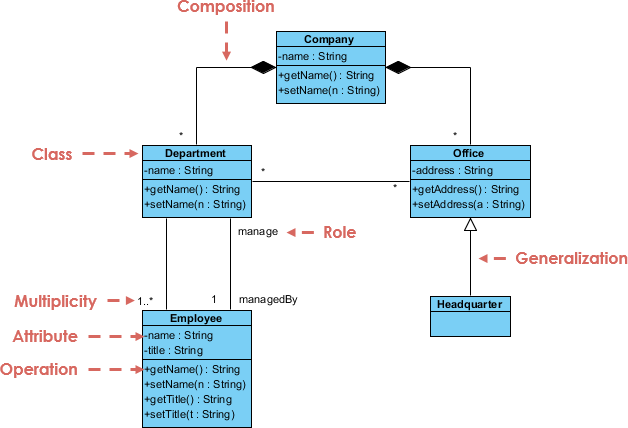
✍️**1. Définissez un diagramme de séquence pour S.C.2**  
**✍️ 2**. **Définissez un diagramme de séquence pour S.B.1.**

# 3. Diagrammes de classes

Vous avez déjà travaillé sur des diagrammes de classes dans le cours de POO, au moins, mais aussi lorsque vous avez généré des diagrammes sous IntelliJ

Les [**diagrammes de classes**](https://www.ibm.com/docs/fr/rational-soft-arch/9.5?topic=diagrams-class) sont fondamentaux pour le processus de modélisation des objets et modélisent la structure statique d'un système. Suivant la complexité d'un système, vous pouvez utiliser un seul diagramme de classes pour modéliser un système complet ou bien, vous pouvez utiliser différents diagrammes de classes pour modéliser les composants d'un système.     Les diagrammes de classes sont les plans de votre système ou sous-système. Vous pouvez utiliser des diagrammes de classes pour modéliser les objets qui constituent le système, pour afficher les relations entre les objets et pour décrire ce que ces objets font et les services qu'ils fournissent.

Un exemple de [**diagramme de classes**](https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-class-diagram/) annoté et focus sur la notion de rôle et multiplicité.

  
  
  
  
✍️**1. Définissez un diagramme de classes qui intègre toutes les classes nécessaires pour remplir le scenario S.C.2 : jouer**