

# TD1 - Conception en UML

Le sujet est écrit comme [une vraie étude de cas](#). Il vous appartient de poser le vocabulaire, de poser des questions, bref de concevoir votre projet.

Dans la suite de cet énoncé, les références sont faites relativement à l'étude de cas.

## 1. Objectifs du TD

- Formez et organisez votre équipe [selon les règles données et les enregistrer sous Moodle](#).
- Mettez-vous d'accord sur la compréhension du projet, au travers de la modélisation UML.
- Ce n'est pas urgent, mais réfléchissez également au rôle pris par chaque membre de l'équipe (P1) et explications dans l'annexe de l'étude de cas.
- Dans cette première séance, une grande partie du travail consiste à analyser et comprendre le sujet. De la même manière que le client n'est pas disponible en permanence, vos enseignants ne pourront pas être présents de manière continue, et ne pourront pas rester dans les groupes plus de 5 minutes. Vous devez donc :
  - o préparer vos questions ;
  - o sauvegarder votre travail parce que vous travaillerez encore.
- **Ne dessinez pas, modélisez.**

## 2. Outils

- Commencez par travailler avec papier et crayon.
- Pour les outils de modélisation UML, plusieurs-vous sont donnés [ici](#).
  - o Attention si vous n'utilisez pas d'outils dédiés vous risquez de dessiner au lieu de modéliser.
- La semaine prochaine vous aurez une livraison dans la foulée du TD. Ces diagrammes seront utilisés et mis à jour pendant tout le module pour une deuxième livraison en fin de module !

## 3. Travail à réaliser en équipe

### 3.1. Former les équipes

- ☐ Pour les quelques-uns qui ne l'auraient pas encore fait : [enregistrer sous Moodle](#), attention aux contraintes.
- ☐ Enregistrer votre équipe dans [Github classroom](#) en respectant bien la règle même nom exactement que dans Moodle.

### 3.2. Première analyse des spécifications

- ☐ Lisez l'étude de cas si ce n'est pas encore fait.
- ☐ Mettez-vous d'accord sur les termes qui vous posent questions, notamment dans le glossaire du domaine, en considérant uniquement le périmètre qui vous est donné. Dans cette étape, ne vous perdez pas. Il faut juste que vous commenciez à poser les bases.
- ☐ Partager un document commun dans lequel vous mettez à jour vos définitions. Vous pouvez passer en français si vous le souhaitez.

### 3.3. Portée du système

Les étapes qui suivent sont menées à la convenance de l'équipe.

Dans ces diagrammes vous cherchez la **cohérence** dans votre analyse du domaine. Cette cohérence est essentielle pour faciliter le développement qui va suivre.

Vos diagrammes évolueront en fonction de vos avancées et des nouvelles fonctionnalités. Organisez-vous bien pour partager, sauvegarder et maintenir vos diagrammes.

a) *Grands cas d'utilisation*

*In UML, use-case diagrams model the behavior of a system and help to capture the requirements of the system. Use-case diagrams describe the high-level functions and scope of a system. These diagrams also identify the interactions between the system and its actors. The use cases and actors in use-case diagrams describe what the system does and how the actors use it, but not how the system operates internally. **Error! Reference source not found.** A use case diagram specifies the actors involved, the main functionalities the system supports for these actors, and the relationships (extends, includes, and specializes) between them.*

- ☐ Déterminer quels sont les acteurs
- ☐ Identifier les grands cas d'utilisation
- ☐ Prenez bien compte de l'ensemble des spécifications pour identifier les acteurs en choisissant un nom unique, les grands cas d'utilisation et leurs relations éventuelles.
- ☐ Faire un premier diagramme de cas d'utilisation qui couvre les acteurs et les cas déterminés (ne cherchez pas à faire compliqué).
- ☐ Mettez-vous d'accord et partagez un document commun, quand bien même ce serait des copies d'écran ou des photos de papiers.

b) *Analyse du domaine : Diagramme de classe*

*A class diagram shows a set of classes, interfaces, and collaborations and their relationships. These diagrams are the most common diagram found in modeling object-oriented systems. ... You use class diagrams to model the static design view of a system. For the most part, this involves modeling the vocabulary of the system, modeling collaborations, or modeling schemas. Class diagrams are also the foundation for a couple of related diagrams: component diagrams and deployment diagrams*

- ☐ Identifier les principaux concepts du domaine qui pourraient former des classes dans un diagramme de classes
- ☐ *Rappel : un diagramme de classes au niveau du domaine ne vise qu'à représenter les grands concepts et à identifier les responsabilités<sup>1</sup> portées par les classes. Il ne contient surement pas de structures telles que des tableaux, des maps etc. Cela sera décidé ultérieurement quand on aura une meilleure connaissance du projet et que nous serons en mesure de faire des choix de conception.*

c) *Analyse du domaine : Diagramme de séquence*

*A sequence diagram is a Unified Modeling Language (UML) diagram that illustrates the sequence of messages between objects in an interaction. A sequence diagram consists of a group of objects that are represented by lifelines, and the messages that they exchange over time during the interaction. A sequence diagram shows the sequence of messages passed between objects. Sequence diagrams can also show the control structures between objects. The sequence diagram shows the objects and the messages between the objects. **Error! Reference source not found.***

**Rappel :** un diagramme de séquence fait référence à un unique cas d'utilisation.

---

<sup>1</sup> Une responsabilité est une sorte de contrat, ou d'obligation, pour une classe. Elle se place à un niveau d'abstraction plus élevé que les attributs ou les opérations. En fait, on peut dire que les attributs, les opérations, et les associations représentent les propriétés élémentaires qui contribueront à remplir les responsabilités de la classe. **Error! Reference source not found.**

- ❑ Prendre le cas d'utilisation que vous considérez le plus important et un diagramme de séquences correspondant sans gérer les cas d'erreur. Vous pouvez cependant les faire apparaître par des notes.
- ❑ Assurez-vous de la cohérence entre votre diagramme de séquence et votre diagramme de classe, que vous mettez éventuellement à jour.

Dans ces diagrammes vous cherchez la cohérence entre votre analyse du domaine et le comportement réel du système (essayez de vous mettre à la place de l'acteur). N'hésitez pas à utiliser une esquisse d'interface graphique si cela peut vous aider à comprendre et partager votre vision, **MAIS aucune interface graphique n'est demandée dans ce projet, vous avez déjà beaucoup à faire.**

#### *d) Glossaire*

- ❑ Vous mettez à jour votre glossaire, en donnant vos propres définitions du vocabulaire vraiment utile pour un reel "shared understanding of key concepts and terminology among stakeholders and team members" 😊.
- ❑ N'oubliez pas de clarifier les rôles des acteurs en relation avec le diagramme de cas d'utilisation.