

## ***Etude Préalable - EZUML***

***Fabrice ARNOUT, Guillaume RENARD, Marcus RICHIER & Lucas WEISS  
Vincent THOMAS - 2021/2021***

### **Tables des matières:**

<b>Partie I. Présentation du projet</b>	<b>2</b>
<b>Partie II. Liste des fonctionnalités du système</b>	<b>4</b>
<b>Partie III. Les cas d'utilisation, les scénarios et les conditions de validation</b>	<b>5</b>
Les diagrammes de cas d'utilisations :	5
Les scénarios	7
Les conditions de validations	15
<b>Partie IV. Les diagrammes de classes</b>	<b>16</b>
<b>Partie V. Le recensement et l'évaluation des risques</b>	<b>17</b>
<b>Partie VI. La répartition des tâches</b>	<b>18</b>

## ***Etude Préalable - EZUML***

***Fabrice ARNOUT, Guillaume RENARD, Marcus RICHIER & Lucas WEISS  
Vincent THOMAS - 2021/2021***

### ***Partie I. Présentation du projet***

Dans le cadre du module de projet tutoré, il nous a été attribué le sujet n°10 de Vincent Thomas: "Logiciel pour créer/manipuler des diagrammes de classes". Avant de pouvoir expliquer notre organisation et nos raisonnements concernant le sujet, nous devons d'abord définir le sujet dans son ensemble.

Avant toute chose, nous devons définir ce qu'est un diagramme de classe. Un diagramme de classe, c'est un schéma utilisé pour représenter les classes, les interfaces du système ainsi que leurs relations. C'est un procédé couramment utilisé afin de représenter d'une façon statique et générale l'architecture et la composition du code d'un projet. Le fait qu'il soit généré de façon automatique sera géré au sein de notre projet de façon à ce qu'une classe soit reconnue, qu'elle soit exploitée afin d'en connaître ses attributs, ses constructeurs et ses méthodes, mais également les dépendances qui la composent vis-à-vis des classes déjà affichées par notre projet et celles qui ne le sont pas encore.

Un exemple simple expliquant cela est la classe String qui est une classe primaire déjà présente dans le compilateur java. En temps normal, un attribut de type string d'une classe est affiché dans le rectangle représentant cette classe, cependant si la classe "String" a été spécifiquement demandée d'apparaître, l'attribut de type string sera alors affiché sous la forme d'une flèche. Ce cas de traitement sera alors traité par le logiciel que nous allons créer, et ce, de façon automatique pour chaque classe insérée ou supprimée de la zone d'affichage du programme.

Concernant ce projet, nous avons décidé de créer une application qui comportera deux modes, où l'utilisateur aura la possibilité de passer d'un mode à l'autre à partir du menu de l'application.

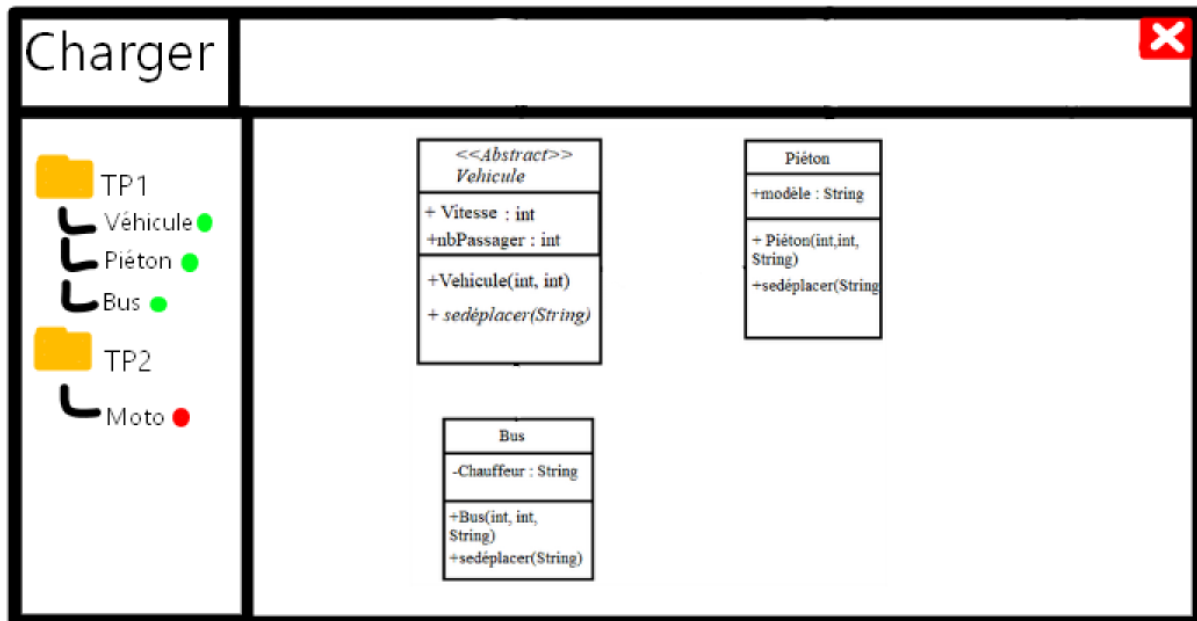
Le premier mode est qualifiable de mode "lite" qui ne comportera que les fonctionnalités basiques afin que le logiciel reste simple d'utilisation, mais également qui réponde à la demande, c'est-à-dire qu'elle a pour objectif d'être utilisée par des premières années de DUT Informatique et plus. Ainsi, pour faciliter la prise en main et l'apprentissage, le mode lite ne permettrait que l'ajout de fichiers .class ou leur suppression, permettant d'afficher un diagramme de classe où ils pourront déplacer les classes librement dans la zone d'affichage.

Le deuxième mode, quant à lui, est qualifiable de mode "complet" qui comportera toutes les fonctionnalités du mode "lite" mais qui aurait la possibilité de modifier le contenu des classes affichées, mais également les dépendances, et si possible comme finalité, la possibilité de créer de nouvelles classes dans le diagramme ainsi que son squelette dans un fichier .java. L'utilisation ici se veut plus poussée, mais également que ce logiciel est une portée éducative : en effet, lors de la conception, lors du passage d'une itération à une autre, notre programme permettrait d'afficher le résultat de l'itération finie et la modification du diagramme de classe afin de pouvoir représenter la suivante.

## Etude Préalable - EZUML

Fabrice ARNOUT, Guillaume RENARD, Marcus RICHIER & Lucas WEISS  
Vincent THOMAS - 2021/2021

Voici une maquette du logiciel à la fin de l'itération 1 :



Cette maquette représente l'interface graphique de la version attendue en fin d'itération 1.

Maquette de l'aspect visuel de notre application :

- Barre des tâches :

- Charger : Permet de charger les fichiers classes afin de générer le diagramme de classes

- Partie à gauche :

Montre l'ensemble des fichiers .class importés et s'ils sont représentés dans le diagramme de classes en cours, un rond vert signifie qu'il est affiché, un rond rouge signifie qu'il n'est pas affiché.

- Partie centrale:

La représentation du diagramme de classes des fichiers .class de la partie gauche, les rectangles composant les classes sont déplaçables et réagénçables afin d'avoir un résultat satisfaisant. On pourra aussi supprimer et ajouter des classes, ce qui sera montré par la partie de gauche, les relations (flèches) entre les classes étant générées automatiquement.

## ***Etude Préalable - EZUML***

***Fabrice ARNOUT, Guillaume RENARD, Marcus RICHIER & Lucas WEISS  
Vincent THOMAS - 2021/2021***

### **Partie II. Liste des fonctionnalités du système**

Pour l'itération 1 seule les fonctionnalités suivantes sont attendues :

- Charger un répertoire / fichier (s'affiche dans l'arborescence pour permettre les glisser déposer)
- Glisser-déposer des fichiers .class de la partie à gauche pour l'ajouter au diagramme
- Raccourci clavier pour supprimer un élément du diagramme (touche Suppr.)
- Déplacer une classe dans la zone d'affichage
- Afficher les packages en format texte dans la classe (et pouvoir les masquer)
- Pouvoir retirer des classes du diagramme
- Pouvoir réinitialiser le diagramme
- Pouvoir sélectionner une ou plusieurs classes et les déplacer (Souris et ctrl-click)

Le logiciel ne disposera que d'une version lite dans l'itération 1, mais une version plus complète accessible via un bouton sera disponible dans une future itération.

Description des fonctionnalités évoquées:

Fonctionnalité	Description
Charger un fichier	Permet de charger un fichier .class ce qui permettra de l'ajouter dans le diagramme. Un fichier chargé est automatiquement ajouté à l'arborescence.
Charger un répertoire	Permet de charger tous les éléments contenus dans un répertoire. Elle charge tous les fichiers .class présents dans le répertoire et les sous-répertoire et les ajoute à l'arborescence en respectant celle du répertoire.
Afficher arborescence	Permet d'afficher sous la forme d'un arbre les différents éléments chargés. Sera constamment affiché à gauche du programme (une fonctionnalité future pourrait permettre de masquer cet affichage).
Créer nouveau diagramme	Réinitialise le diagramme en cours de création/modification en un diagramme vide. Le diagramme sera perdu en effectuant cette action.
Masquer les packages	Permet de masquer les packages affichés sous forme textuelle. Un bouton permet de masquer tous les packages, libérant ainsi l'espace que le texte occupait.
Afficher les package	Permet d'afficher les packages sous forme textuelle. Un bouton permet d'afficher tous les packages, décalant les autres éléments pour le placer entre le nom de la classe et les attributs.

## ***Etude Préalable - EZUML***

***Fabrice ARNOUT, Guillaume RENARD, Marcus RICHIER & Lucas WEISS  
Vincent THOMAS - 2021/2021***

Déplacer classe	Permet de déplacer une classe unique. En restant appuyé sur une classe et en déplaçant la souris, cela déplacera la classe sur la page. Permet d'organiser le diagramme.
Déplacer une sélection de classe	Permet de déplacer les classes sélectionnées sur le diagramme, il faut rester appuyer sur l'une des classes sélectionner puis déplacer sa souris, l'ensemble des classes sélectionnées sera déplacé de manière uniforme par rapport au point d'origine.
Sélectionner	Permet de changer la sélection par un clic gauche, (rien ou une classe) ou d'ajouter d'autre classe à la sélection (ctrl + clic gauche).
Ajouter classe dans le diagramme	Permet d'ajouter une classe chargée dans le diagramme. Glisser déposer une classe de l'arborescence au diagramme permet de l'ajouter au diagramme à l'endroit où il a été déposé.

## Etude Préalable - EZUML

Fabrice ARNOUT, Guillaume RENARD, Marcus RICHIER & Lucas WEISS  
Vincent THOMAS - 2021/2021

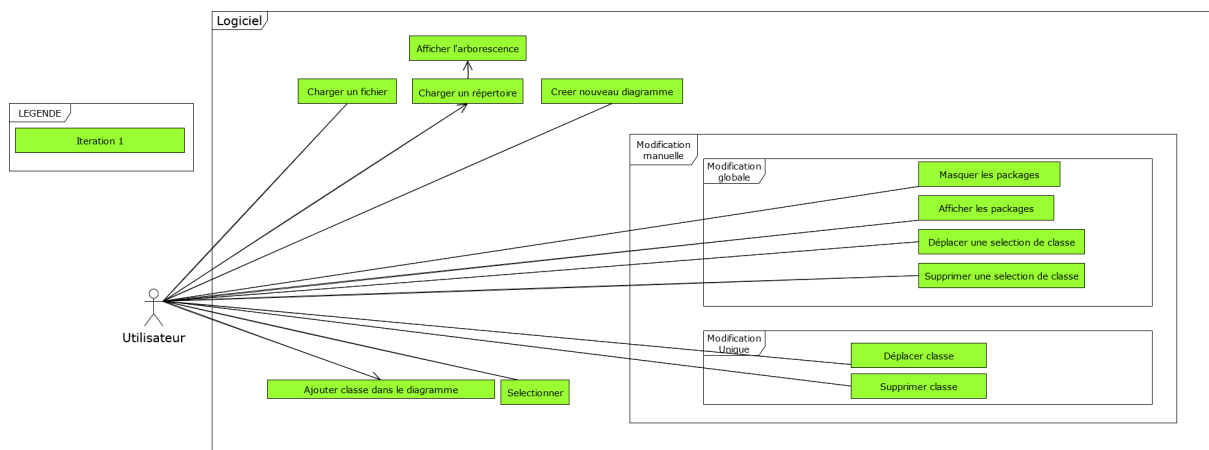
### Partie III. Les cas d'utilisation, les scénarios et les conditions de validation

#### Les diagrammes de cas d'utilisations

**Rappel :** Dès la première itération nous réalisons une interface graphique suivant la maquette (voir partie I)

Notre logiciel n'a qu'un seul type d'utilisateur, toute personne qui démarre le programme a accès aux mêmes options.

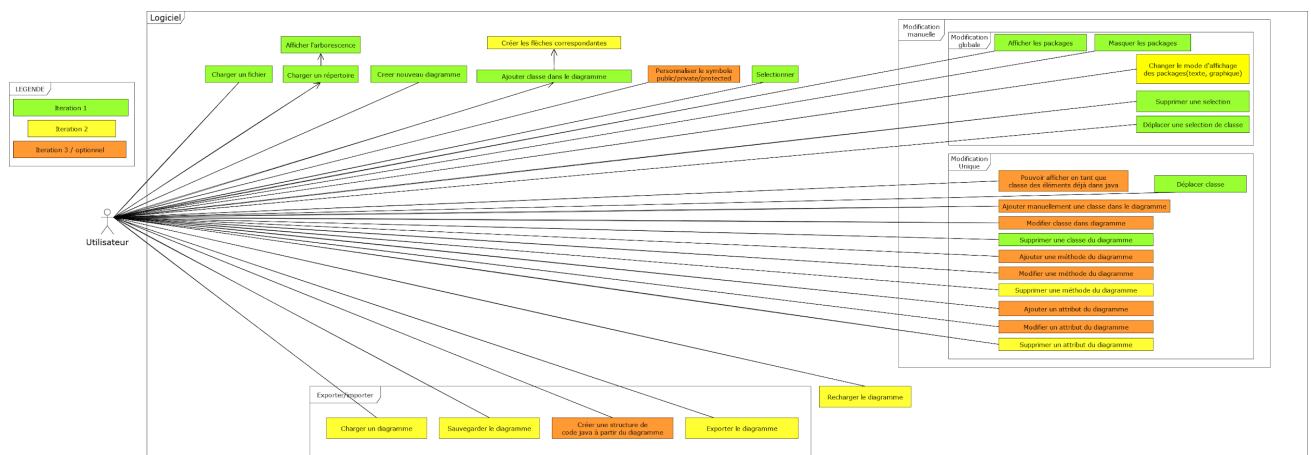
Voici le diagramme de cas d'utilisation de l'itération 1 uniquement :



Les différentes fonctionnalités affichées sont détaillées dans la partie II.

Nous avons également réalisé un diagramme de cas d'utilisation globale, qui pourra être sujet à changement mais qui permet d'évaluer les possibles évolutions futures.

Diagramme de cas d'utilisation global (sujet à changement) :

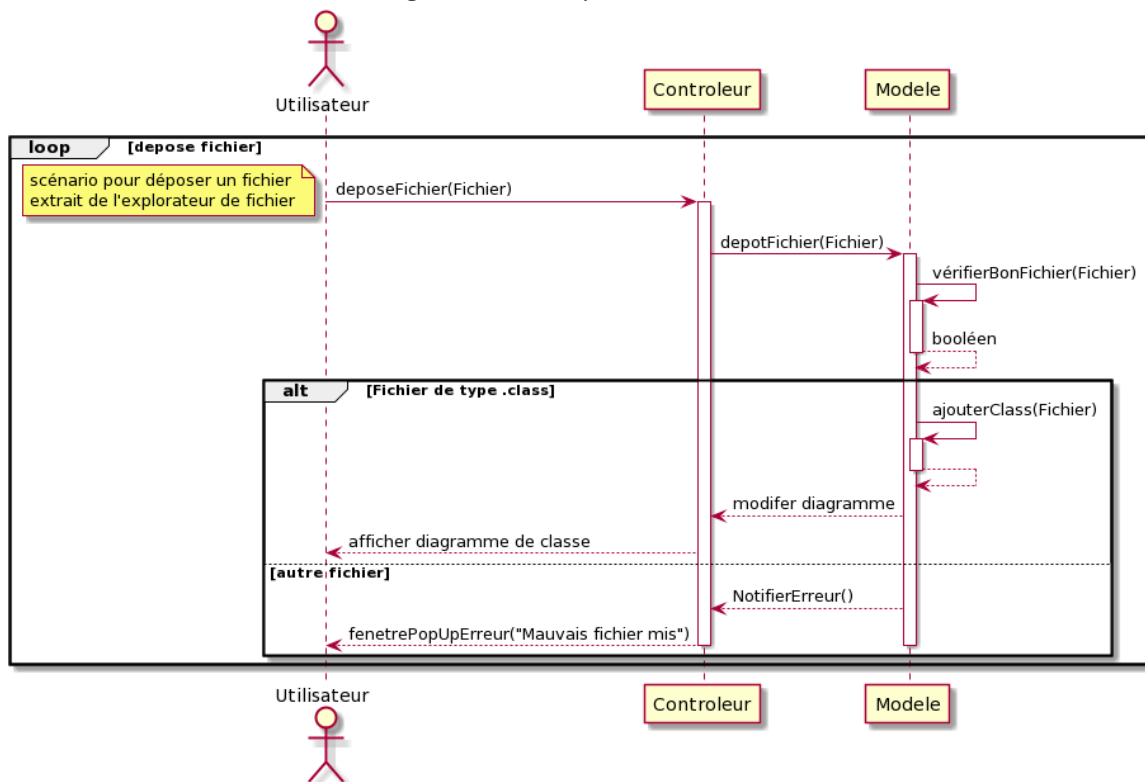


## Les scénarios

### 1 - Création d'un diagramme simple, en cliquer-déposer

1. Patrick ouvre notre application,
2. Il glisse son fichier
3. Si le fichier mis est bien un .class, le diagramme est créé par notre application, les relations (flèches) entre classes étant ajoutées automatiquement conformément aux conventions UML.

diagramme de séquence scénario 1

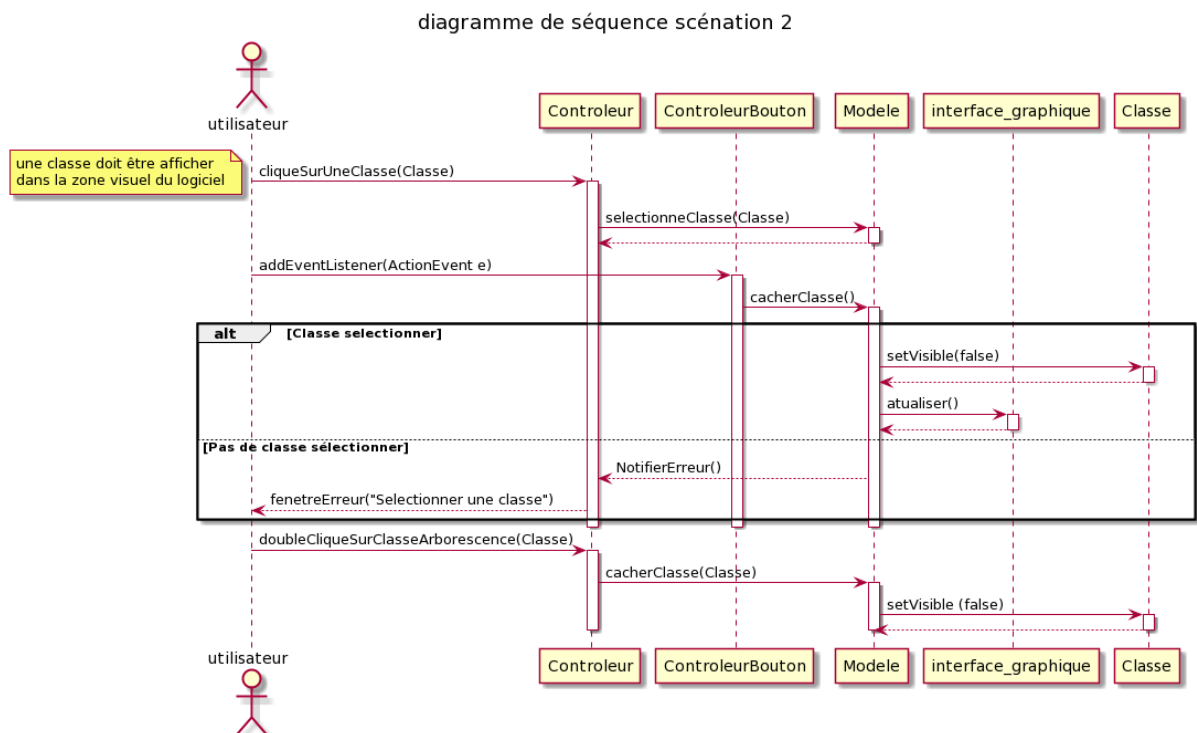


## Etude Préalable - EZUML

Fabrice ARNOUT, Guillaume RENARD, Marcus RICHIER & Lucas WEISS  
Vincent THOMAS - 2021/2021

### 2 - Supprimer une classe d'un diagramme

1. Patrick ouvre notre application, puis ouvre un projet déjà existant,
2. Le logiciel affiche donc le diagramme du projet,
3. Patrick sélectionne une classe du diagramme, puis appuie sur un bouton supprimer ce qui supprime la classe. Il peut aussi appuyer sur suppr ou alors il appuie sur le nom de la classe sur la partie de l'arborescence du projet.
4. Le logiciel supprime la classe du diagramme et enlève les flèches liées à lui.



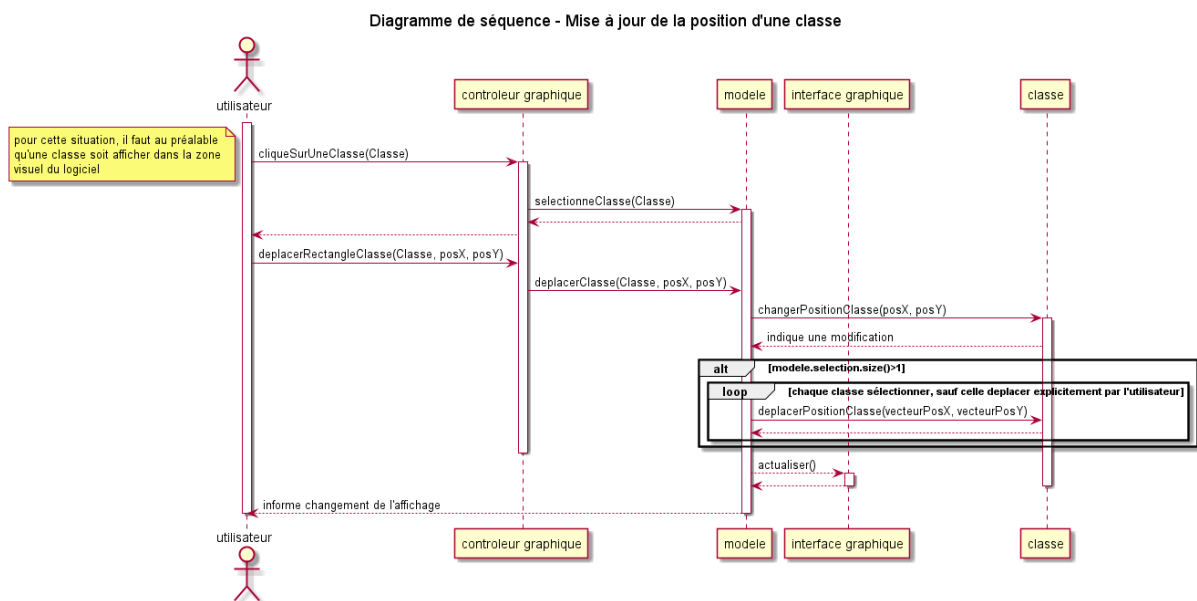


## Etude Préalable - EZUML

Fabrice ARNOUT, Guillaume RENARD, Marcus RICHIER & Lucas WEISS  
Vincent THOMAS - 2021/2021

### 3 - Mise à jour du diagramme après le déplacement d'un objet [Lucas]

1. Au préalable, l'utilisateur a fait afficher une case dans le panneau d'affichage
2. L'utilisateur sélectionne une classe affichée
3. L'utilisateur fait un appui long et déplace sa souris
4. L'interface graphique affiche un cadre gris de taille de la classe sélectionnée selon la position de la souris de l'utilisateur
5. L'utilisateur lâche le clique de la souris
6. Le logiciel déplace la classe sélectionnée à la position décidée par l'utilisateur

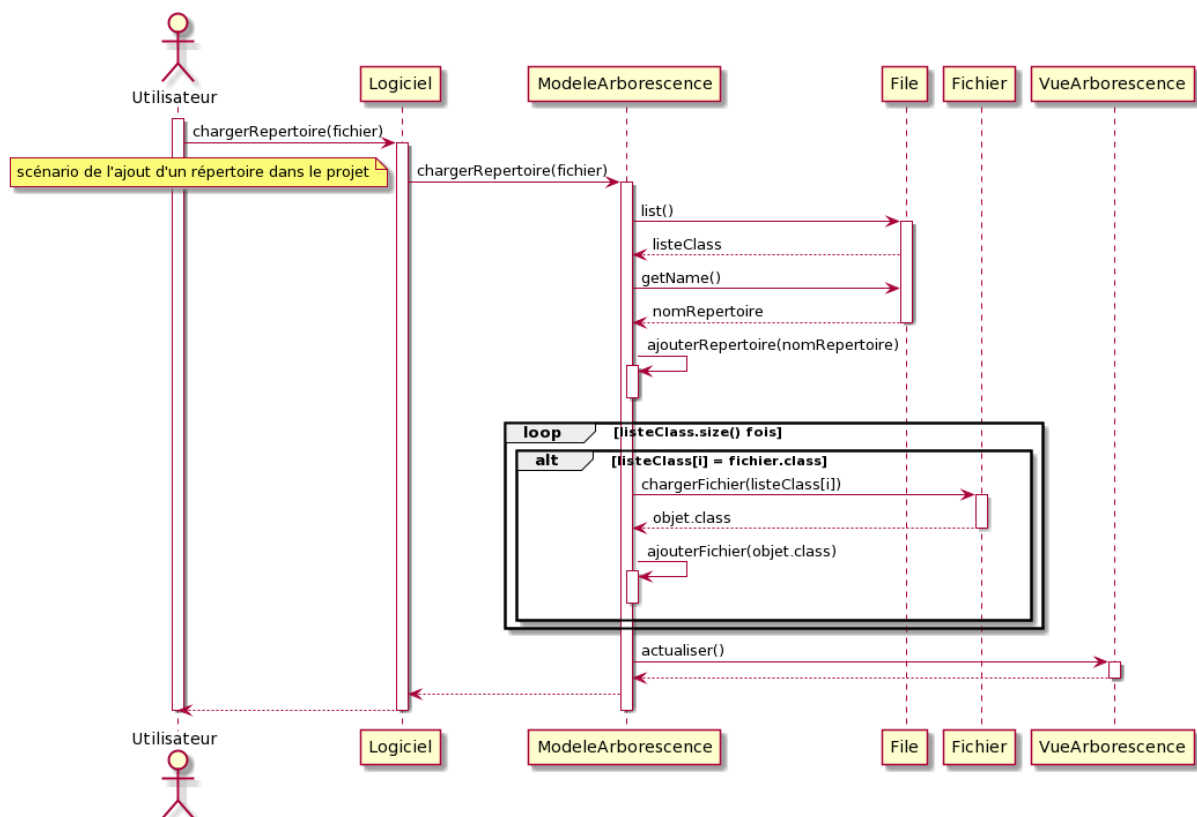


## Etude Préalable - EZUML

Fabrice ARNOUT, Guillaume RENARD, Marcus RICHIER & Lucas WEISS  
Vincent THOMAS - 2021/2021

### 4 - Charger un répertoire dans le projet

1. Patrick souhaite charger un répertoire dans lequel est situé son projet java.
2. Il appuie sur "Ouvrir" dans la barre des tâches et sélectionne le répertoire qu'il veut ouvrir et valide sa sélection.
3. Le logiciel vérifie si le répertoire contient au moins un fichier Java.
4. Le logiciel a détecté au moins un fichier Java.
5. Le répertoire sélectionné s'affiche dans la zone de gauche, avec ce qu'il contient et les fichiers javas contenus dans le répertoire et ses sous-répertoires.

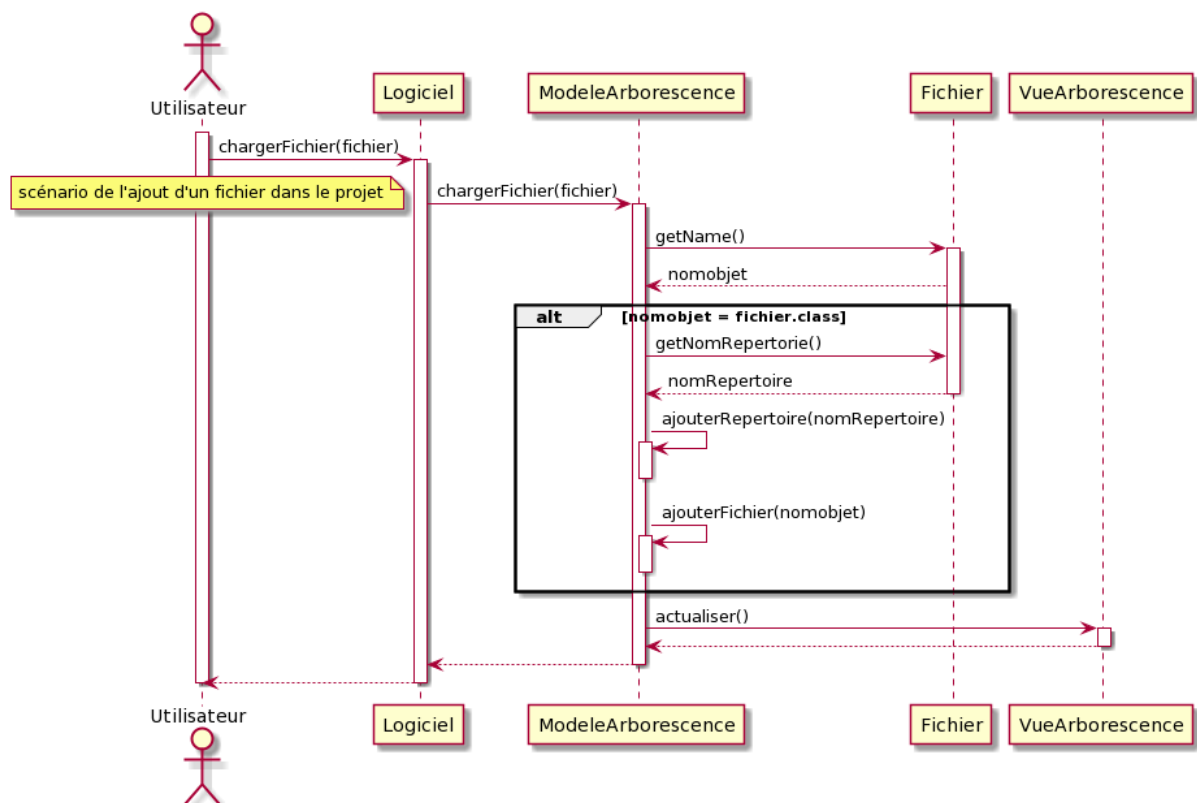


## Etude Préalable - EZUML

Fabrice ARNOUT, Guillaume RENARD, Marcus RICHIER & Lucas WEISS  
Vincent THOMAS - 2021/2021

### 5 - Charger un fichier dans le projet

1. Patrick souhaite charger un fichier dans lequel est situé son projet Java.
2. Il appuie sur "Ouvrir" dans la barre des tâches et sélectionne le fichier qu'il veut ouvrir et valide sa sélection.
3. Le logiciel vérifie si le fichier est bien un fichier Java.
4. Le logiciel a déterminé quel fichier est bien un fichier Java.
5. Le fichier sélectionné s'affiche dans la zone de gauche, avec le répertoire qui le contient.

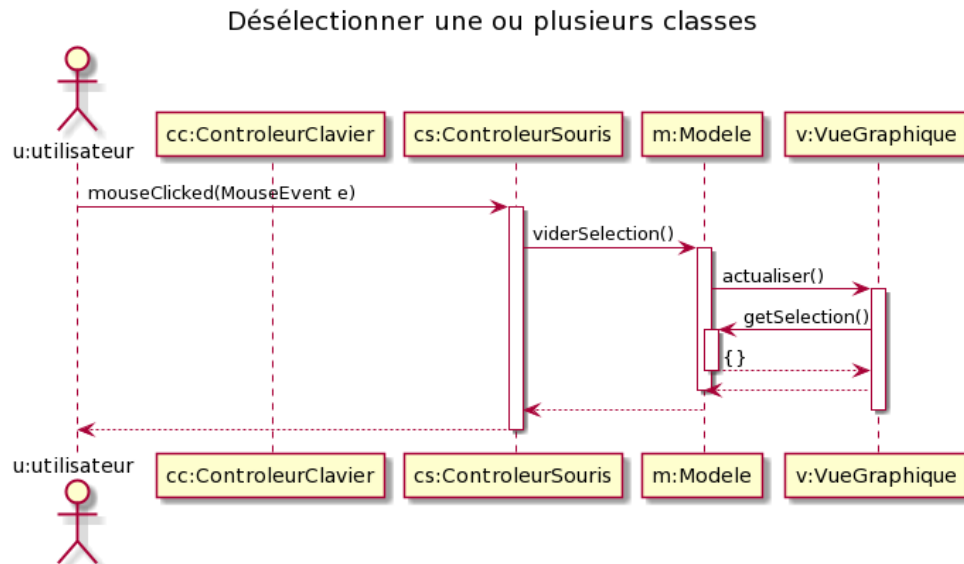


## Etude Préalable - EZUML

Fabrice ARNOUT, Guillaume RENARD, Marcus RICHIER & Lucas WEISS  
Vincent THOMAS - 2021/2021

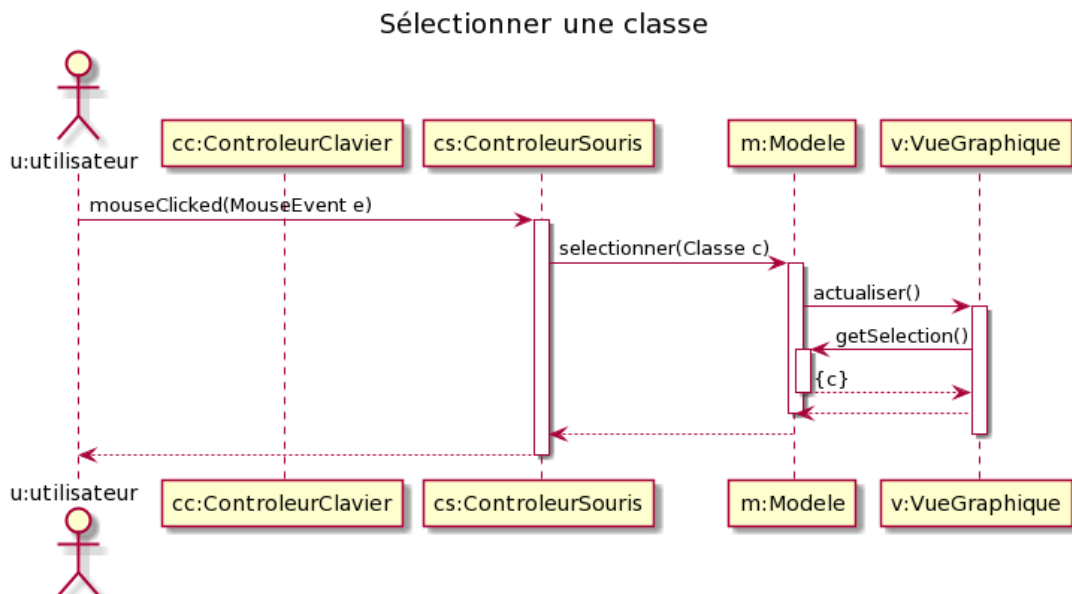
### 6 - Désélectionner une ou plusieurs classes

1. Jean a un diagramme avec 3 classes : classeA, classeB et classeC.
2. Jean fait un clic gauche en dehors des 3 classes, peu importe la sélection actuelle la sélection est vide.



### 7 - Sélectionner une classe

1. Jean a un diagramme avec 3 classes : classeA, classeB et classeC.
2. Rien n'est sélectionné.
3. Jean fait un clic gauche sur la classeA, classeA est sélectionné.
4. Jean fait un clic gauche sur classeB, seule classeB est sélectionnée.

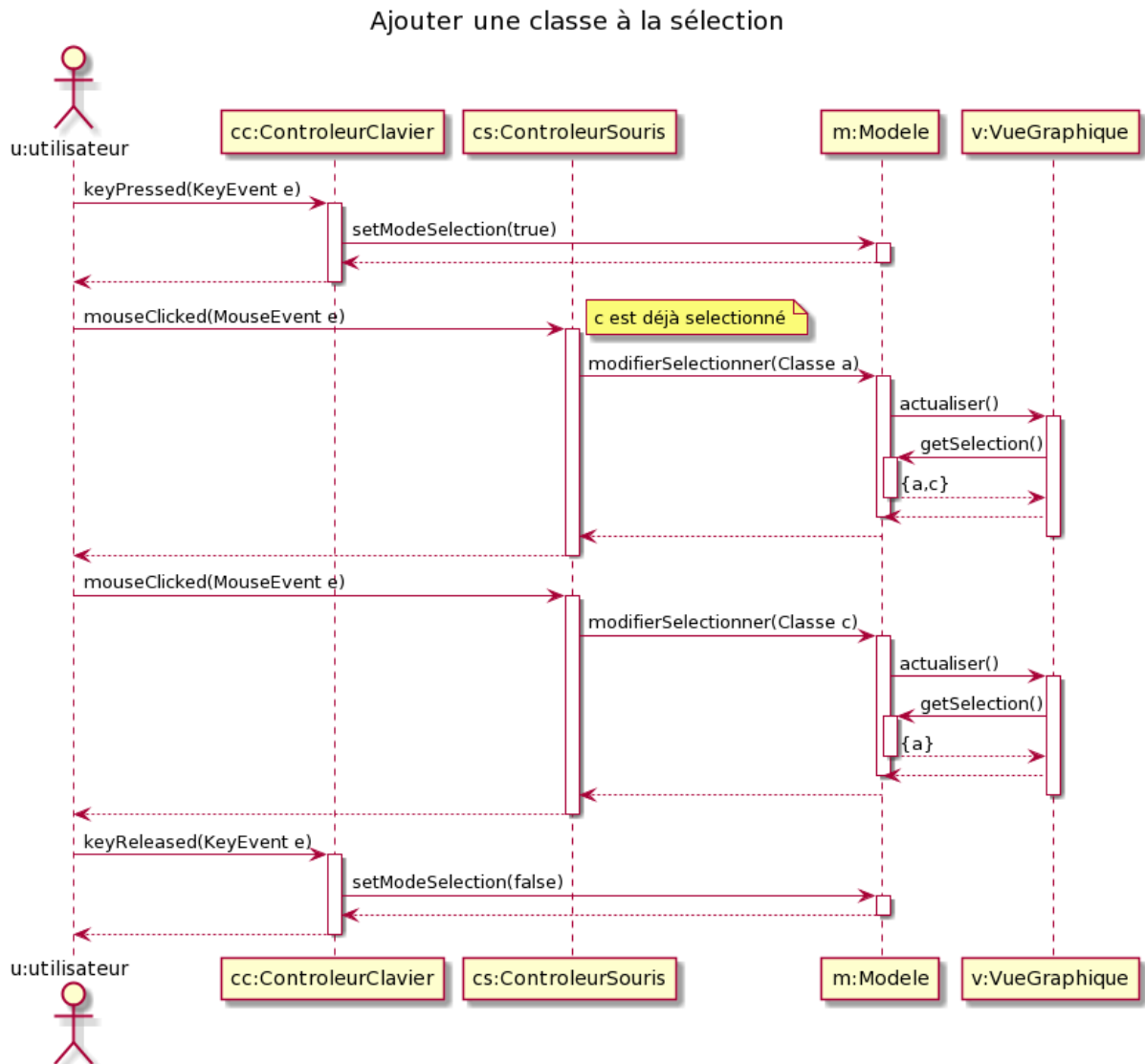


## Etude Préalable - EZUML

Fabrice ARNOUT, Guillaume RENARD, Marcus RICHIER & Lucas WEISS  
Vincent THOMAS - 2021/2021

### 8 - Ajouter une classe à une sélection

1. Jean a un diagramme avec 3 classes : classeA, classeB et classeC.
2. classeC est sélectionné.
3. Jean reste appuyer sur ctrl et fait un clic gauche sur classeA, classeA et classeC sont sélectionnées.
4. Jean reste appuyer sur ctrl et fait un clic gauche sur classeC, seule classeA est sélectionnée.



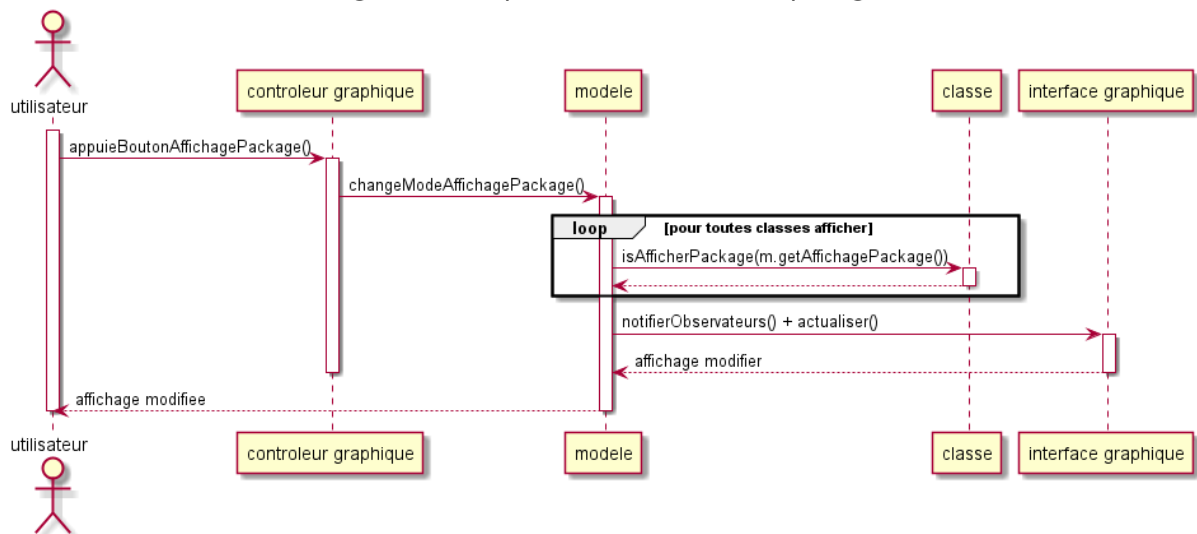
## Etude Préalable - EZUML

Fabrice ARNOUT, Guillaume RENARD, Marcus RICHIER & Lucas WEISS  
Vincent THOMAS - 2021/2021

### 9 - Afficher/Cacher les packages

1. L'utilisateur va cliquer sur le bouton textuel.
2. Le logiciel va alors retenir le choix de l'utilisateur
3. Le logiciel prévient chaque classe qu'il faut afficher ou cacher le package
4. L'interface graphique se met à jour

Diagramme de séquence - Afficher/Cacher les packages



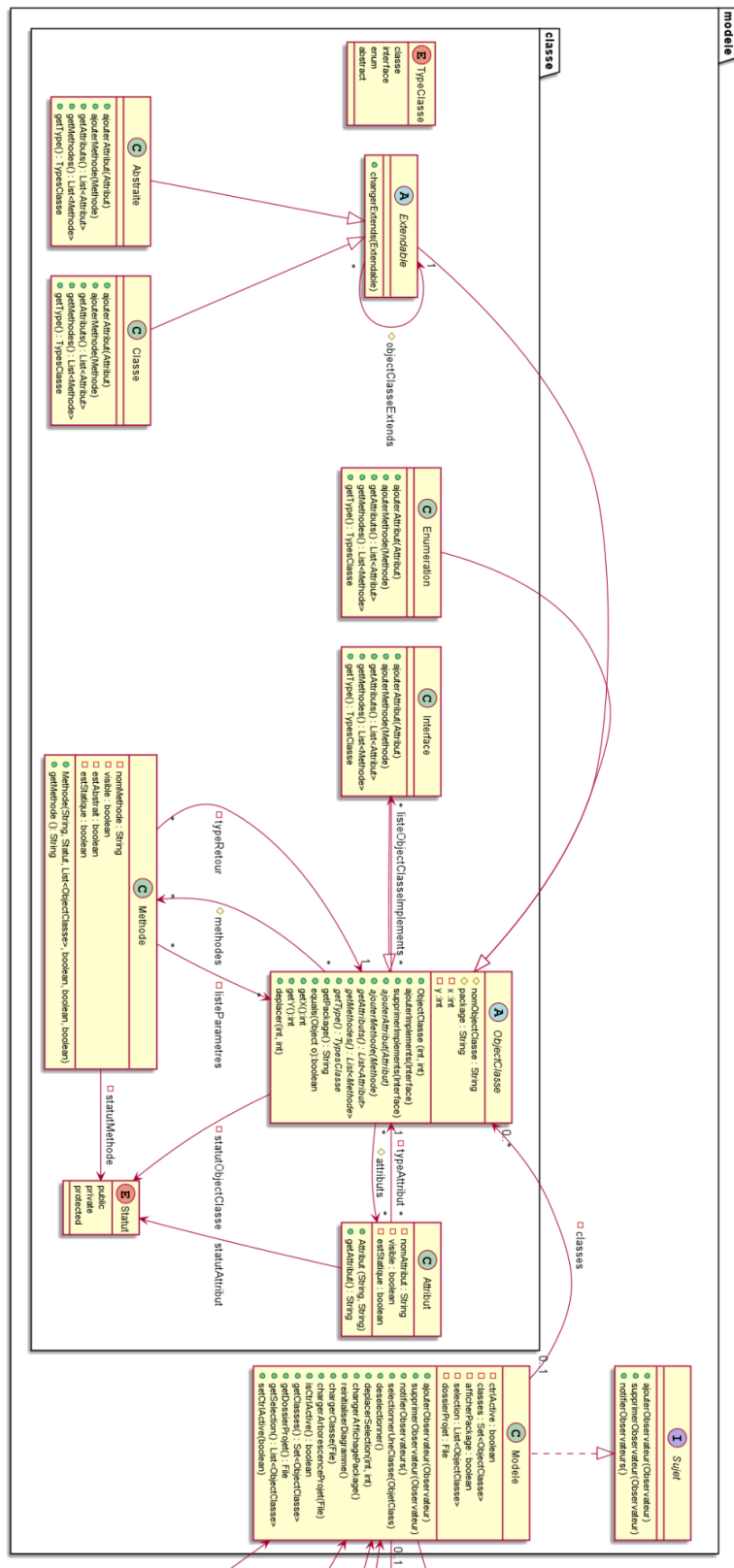
## **Les conditions de validations**

Les conditions de validation de notre itération 1 sont :

- L'affichage des classes dans notre interface graphique, sans les flèches de relation. Il faudra qu'ils contiennent toutes les informations que l'on doit afficher dans les classes des diagrammes.
- L'arborescence se mettra à jour, lorsque l'on glissera un fichier .class dans la fenêtre prévue à cet effet.
- On pourra charger un répertoire, qui affichera les fichiers .class dans l'arborescence de notre application, tout en prenant en compte le placement des fichiers dans le répertoire mis.
- La sélection devra permettre à l'utilisateur de faire des actions sur la classe sélectionnée de manière simple. On pourra aussi ajouter une classe à notre sélection, pour qu'on puisse les déplacer selon nos envies.
- L'affichage des packages devra être lisible et pourra être caché simplement de notre interface graphique.
- La suppression d'une classe devra bien se voir sur l'interface graphique, mais elle sera toujours visible dans l'arborescence de notre projet et on pourra la remettre en visible à tout moment.
- Le déplacement d'une ou plusieurs classes devra se déplacer selon l'envie de l'utilisateur, sans trop de difficulté pour lui.

***Fabrice ARNOUT, Guillaume RENARD, Marcus RICHER & Lucas WEISS  
Vincent THOMAS - 2021/2021***

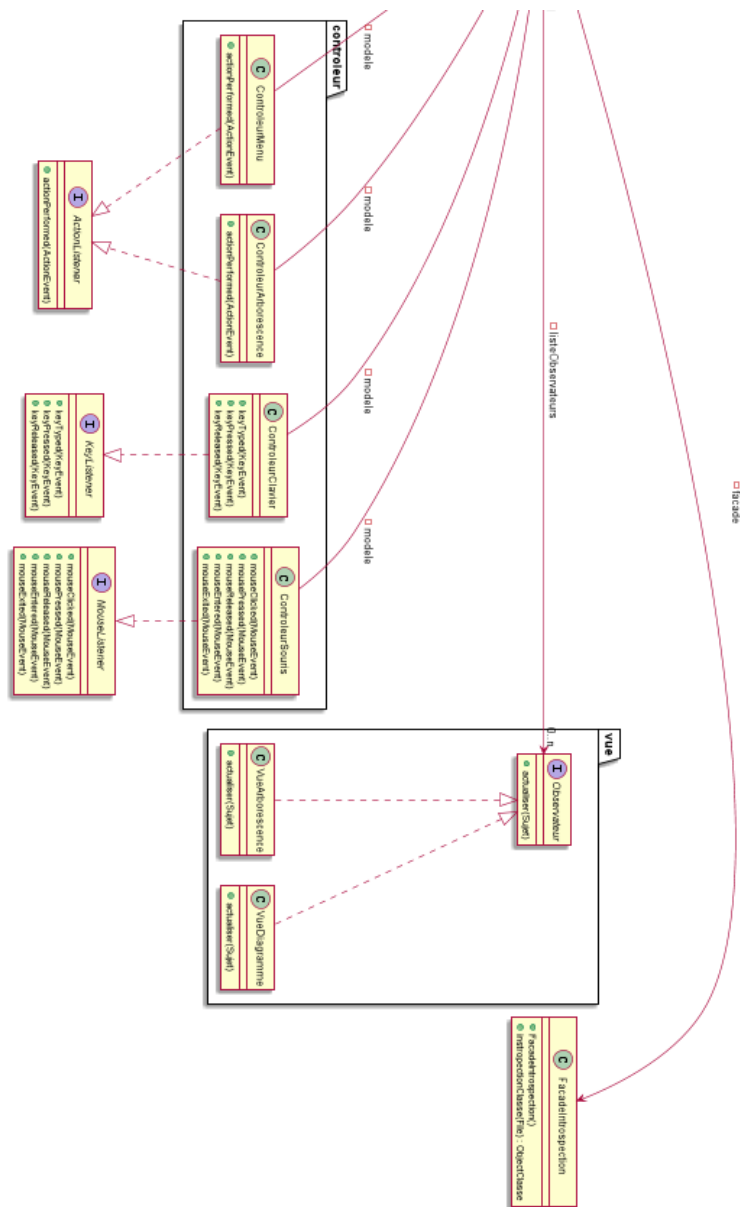
## Partie IV. Les diagrammes de classe





## Etude Préalable - EZUML

Fabrice ARNOUT, Guillaume RENARD, Marcus RICHIER & Lucas WEISS  
Vincent THOMAS - 2021/2021



Voici le diagramme de classe représentant notre première itération. Au vue de sa taille, nous l'avons divisé en deux parties, la première sur le package modèle (sur la page précédente) et la deuxième sur le package contrôleur et vue (cette page). Vous pouvez également le retrouver en format .png accompagné avec cette étude.

Cependant, n'ayant jamais pratiqué l'introspection, le diagramme de classe concernant cette fonctionnalité n'est représenté que par la classe "FaçadeInstrospection" et reste sujet à des changements lors de la première itération.

## ***Etude Préalable - EZUML***

***Fabrice ARNOUT, Guillaume RENARD, Marcus RICHIER & Lucas WEISS  
Vincent THOMAS - 2021/2021***

### **Partie V. Le recensement et l'évaluation des risques**

Pour le recensement et l'évaluation des risques nous avons utilisé la méthode AMDEC.

Risques	conséquence	Fréquence	Gra- vité	P=F* G	Vraisem- blables	plan d'action	Respon- sable	Délai
difficulté à faire la représentation des classes	retard de l'itération	1	2	2		mettre des personnes en plus	Lucas Weiss	à finir avant la fin de l'itération 1
non connaissances de l'introspection en java	retard de l'itération	1	2	2		mettre des personnes en plus	Guillaume Renard	à finir avant la fin de l'itération 1
Personne de notre groupe malade, à cause du covid	la personne est absente et passe en distanciel	2	1	2	X	répartir le travail que la personne ne peut pas faire	Tout le monde	1 semaine
Mauvais évaluation du temps	retard, itération 1 non finie	1	3	3		reporter des fonctionnalités sur l'itération 2	tout le groupe	à finir avant l'itération 2

## ***Etude Préalable - EZUML***

***Fabrice ARNOUT, Guillaume RENARD, Marcus RICHIER & Lucas WEISS  
Vincent THOMAS - 2021/2021***

### **Partie VI. La répartition des tâches**

Pour notre projet, on a prévu de répartir nos tâches de cette manière :

- Fabrice réalise l'interface et les vues de notre projet,
- Marcus est chargé de faire le modèle de notre projet,
- Guillaume s'occupera de toute la partie de l'introspection,
- Lucas se chargera de la partie représentation des classes.

Cependant, selon l'avancement de certains ou les complications de certaines tâches, l'attribution des tâches et plus particulièrement de fonctionnalités pourra être revue. Pour ce faire, nous utiliserons "Trello".

Afin que notre projet avance bien, nous avons décidé de procéder à des vérifications. Pour ce faire, une personne sera chargée de vérifier le travail d'un autre. Nous l'avons décidé ainsi:

- Le travail de Fabrice sera regardé par Lucas.
- Le travail de Marcus sera regardé par Guillaume.
- Le travail de Guillaume sera regardé par Marcus.
- Le travail de Lucas sera regardé par Fabrice.