

# ***Projet tutoré***



***Fabrice Arnout - Guillaume Renard  
Marcus Richier - Lucas Weiss***

## Description du projet

Le projet a pour objectif initial de réaliser le diagramme de classe d'un projet java, d'un package ou seulement de quelques classes.



L'objectif du projet est de réaliser une application qui génère visuellement un diagramme de classe selon des fichiers .class donnés au préalable par l'utilisateur. L'application doit répondre à un besoin, et même si plusieurs applications existent déjà, la plupart étant sur Internet, nous souhaitons réaliser une application locale et qui permet de manipuler le positionnement comme il était possible avec ObjectAid.

Il va être utilisé par de jeunes programmeurs en apprentissage. Il se doit donc d'être fonctionnel, mais aussi épuré, et simple d'utilisation.

## c La répartition des rôles

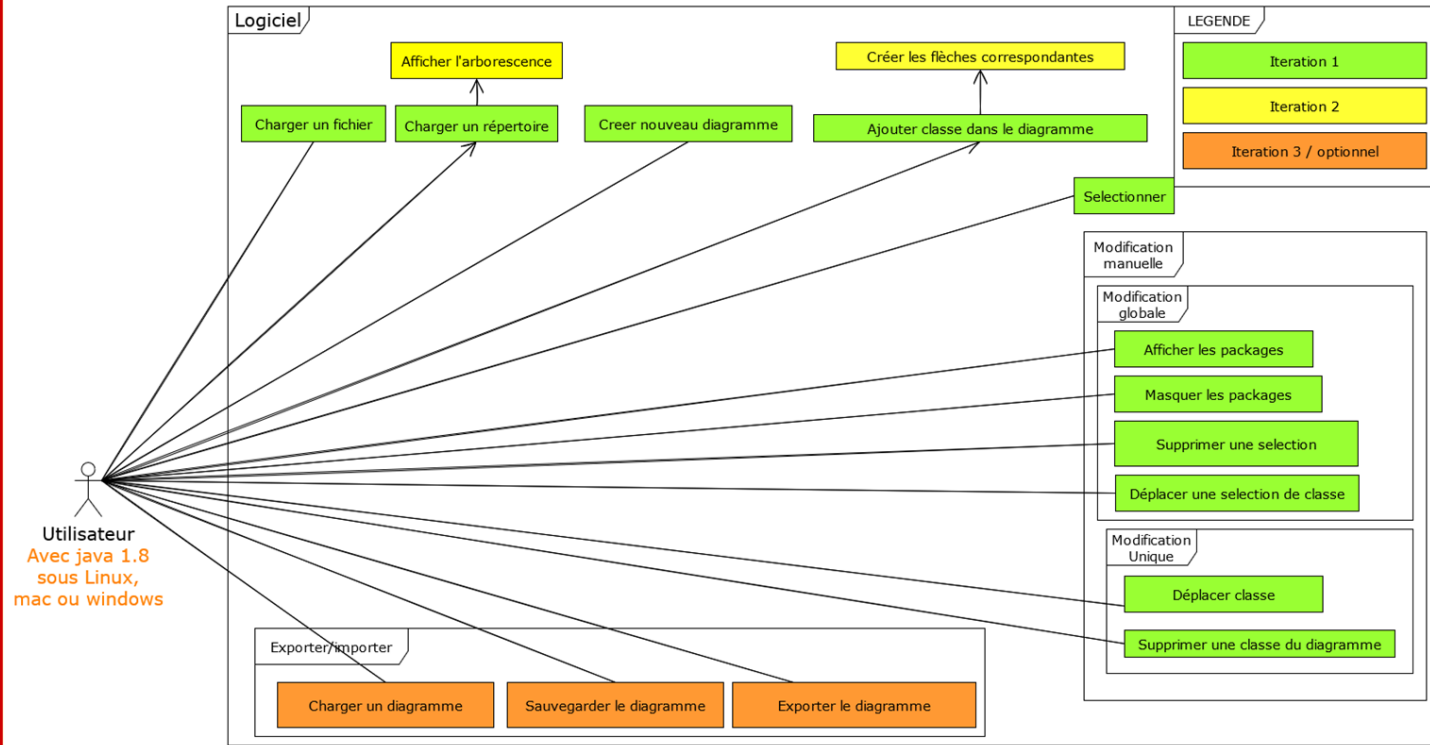
Equipier / Itération	1	2	3
Fabrice	Affichage	Vue et contrôleur Arborescence	
Lucas	Modèle des classes java	Affichage	Affichage des flèches et cardinalités
Guillaume	Introspection	Introspection héritage/implémentation	Sauvegarde, chargement et portage
Marcus	Modèle de l'application	Modèle de l'application + Contrôleurs	Vue et contrôleur Arborescence et exportation



## Sommaire

- I. Etat du projet.
- II. Récapitulatif des itérations
- III. Contraintes
- IV. Fonctionnalité : Sauvegarde d'une classe en .ezuml
- V. Changement de conception
  - ◆ stockage des informations
  - ◆ sélection d'une classe
  - ◆ chargement d'un diagramme
- Conclusion

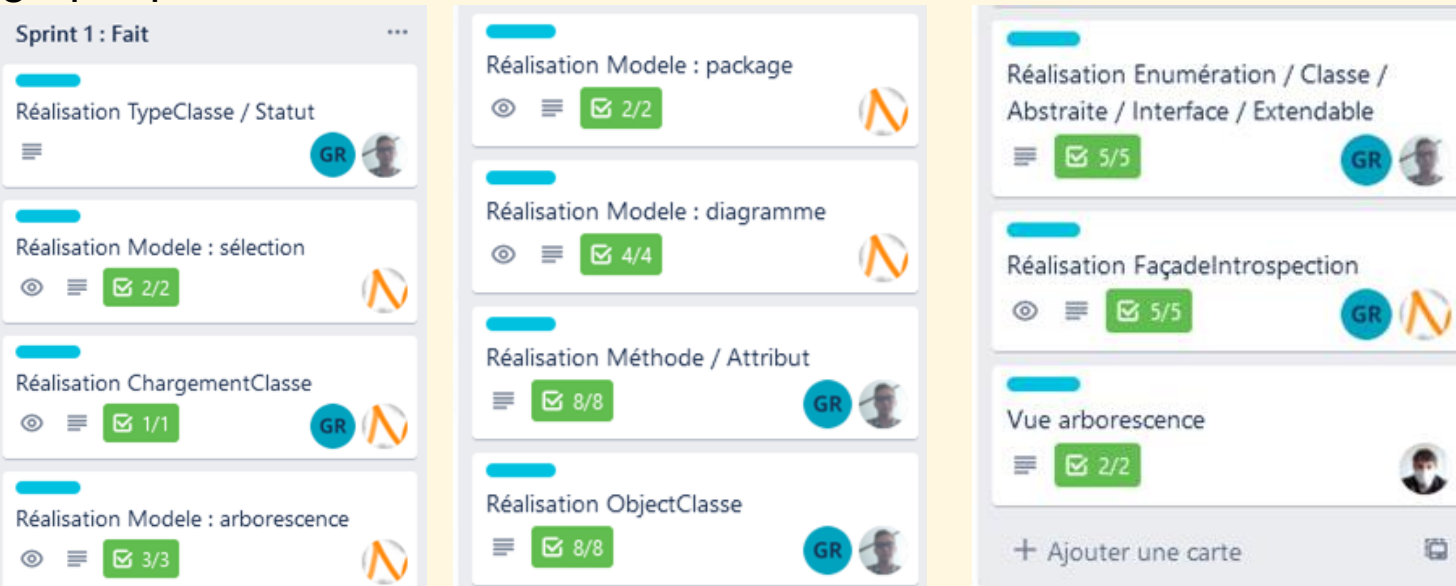
### c Diagramme de cas d'utilisation



Notre diagramme de cas d'utilisation a été revu à la baisse à cause de retard sur la partie visuelle.

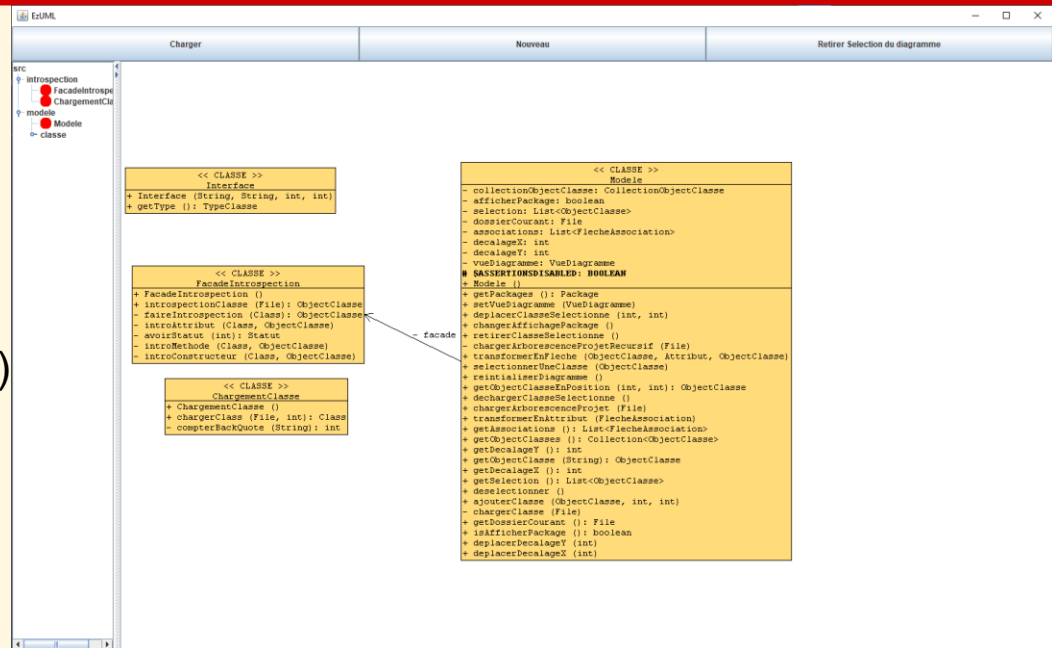
### c Etat à la fin de l'itération 1

Modèle et façade introspection fonctionnelle sans les “flèches” et sans interface graphique réellement fonctionnelle.



### c Etat à la fin de l'itération 2

- Version fonctionnelle
- Affichage
- L'ajout des flèches (incomplet)



### Prévu pour l'itération 3

Objectif:

- Version stable sur la version 8 de java
- Contrôleur de l'arborescence
- L'ajout des cardinalité et des cas complexe de flèches
- Des moyens de récupérer nos diagramme (exportation et sauvegarde)
- Mode simple
- Sortir les attributs : les classes qui ne sont pas chargées ou affichée



### c Etat à la fin de l'itération 3

Résultat :

- Version stable sur la version 8 de Java ✓
- Contrôleur de l'arborescence ~
- L'ajout des cardinalités et des cas complexe de flèches ✓
- Des moyens de récupérer nos diagrammes (exportation et sauvegarde) ✓
- Mode simple ~
- Sortir les attributs : les classes qui ne sont pas chargées ou affichées ✗

## II. Récapitulatif des itérations

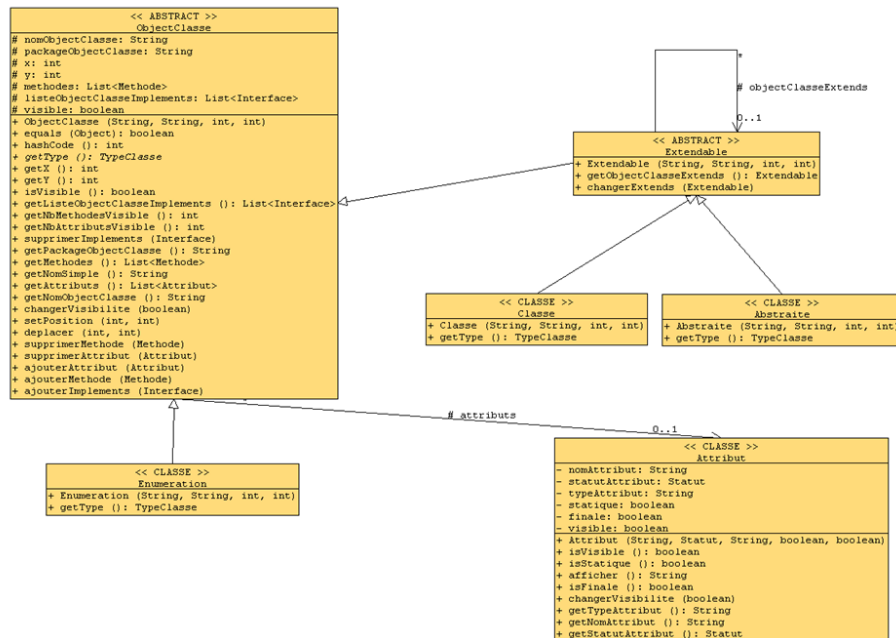


# Application finale

EZ\_EUML

Fichier Edition

src  
↳ introspection  
↳ vue  
↳ controleur  
↳ principal  
↳ modele  
↳ classe  
↳ Methode  
↳ Abstraite  
↳ Extensible  
↳ Interface  
↳ Classe  
↳ Enumeration  
↳ Statut  
↳ Attribut  
↳ TypeClasse  
↳ ObjectClasse



Fichier Edition

Charger fichiers .class - ctrl shift f

Charger dossier - ctrl shift d

Charger .ezuml - ctrl o

Sauvegarder .ezuml - ctrl s

Exporter - ctrl e

Edition

Retirer selection du diagramme - suppr

Recharger dernier chargement - f5

Association: —→  
Hérédité: —|→  
Implémentation: —|→  
Abstrait: En italique  
Statique: En gras

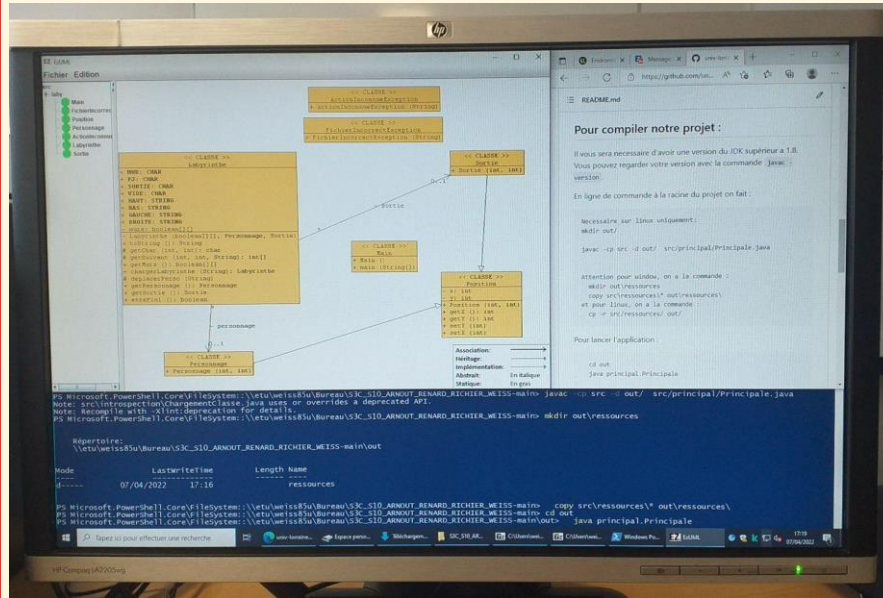


## Les contraintes du sujet

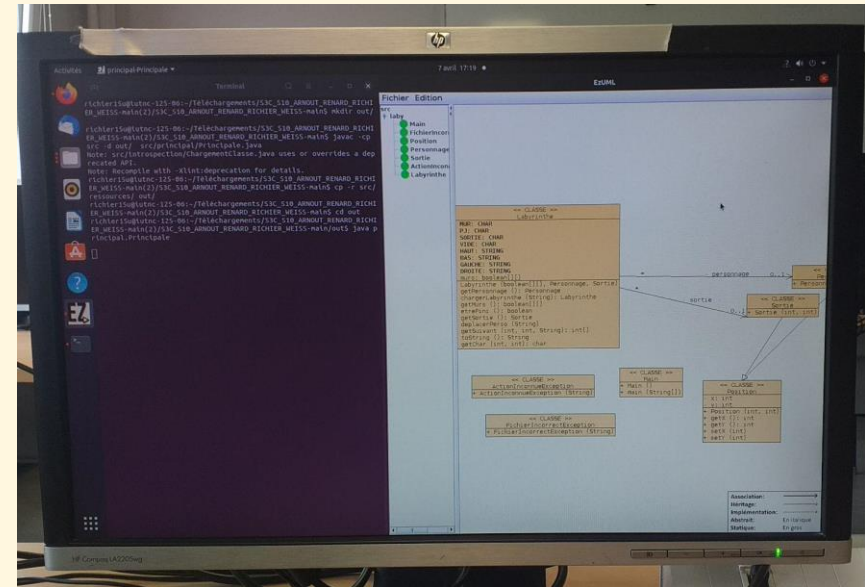
- Le logiciel doit être utilisé par les étudiants de l'IUT.
  - ◆ Il faut donc que la version fonctionne avec la version de Java installée sur les machines de l'IUT (Java 1.8)
  - ◆ Il doit également est portable sur Windows, Linux et Mac.
  
- Il faut également que le temps d'apprentissage soit inférieur à 10 minutes, donc que l'application soit épuré et clair.

### III.Contraintes

## c La résolution des contraintes

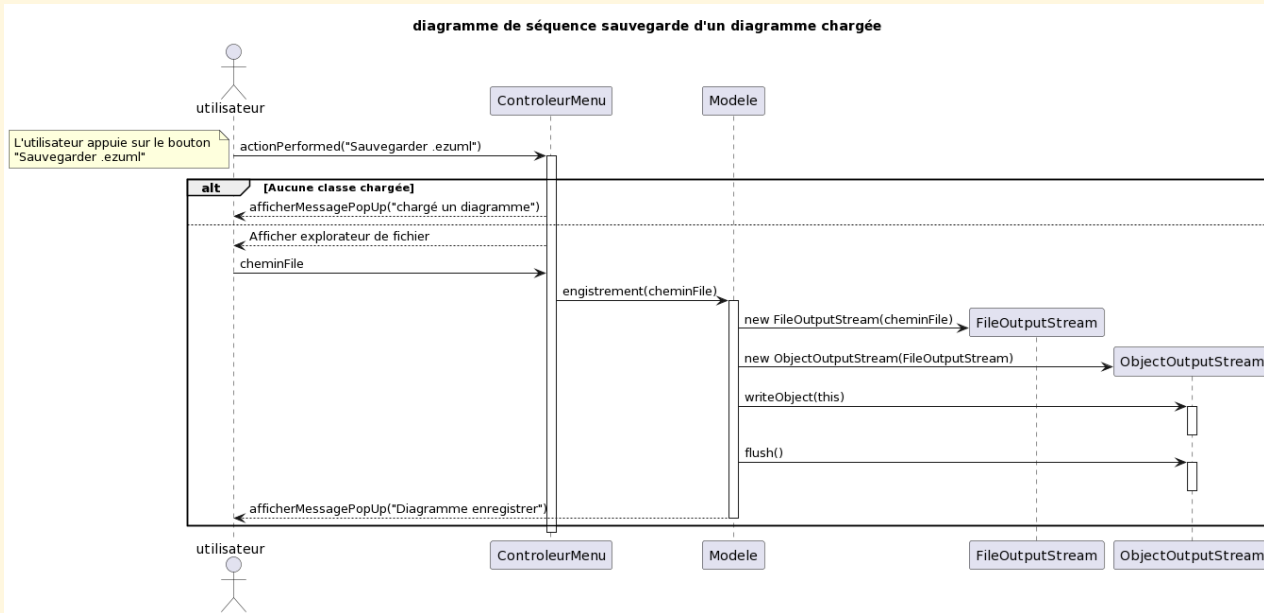


Sous Windows



Sous Linux

## c Sauvegarde d'une classe en .ezuml

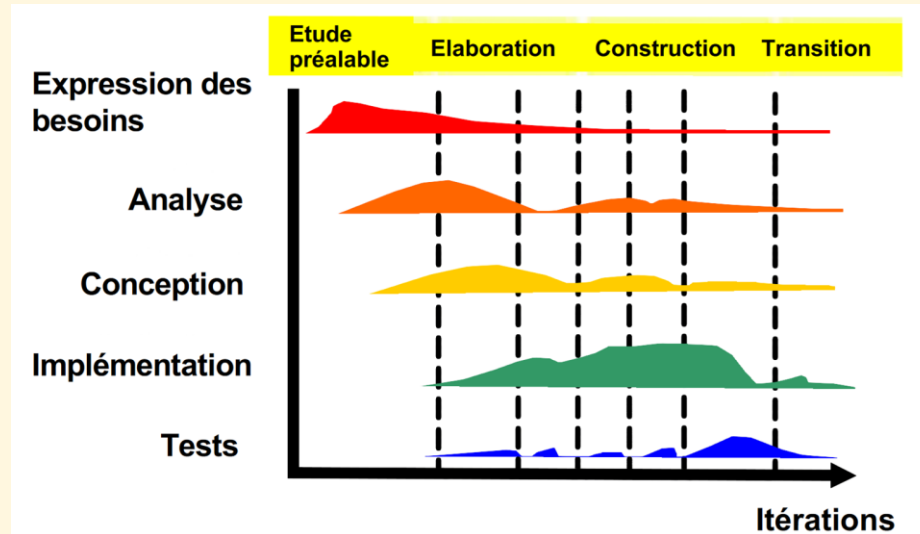


Dans cette itération, nous avons décidé de réaliser la sauvegarde d'un diagramme.



## c Changement de conception

Malgré la réalisation d'une étude préalable de notre projet. On a rencontré quelques difficultés de conception.

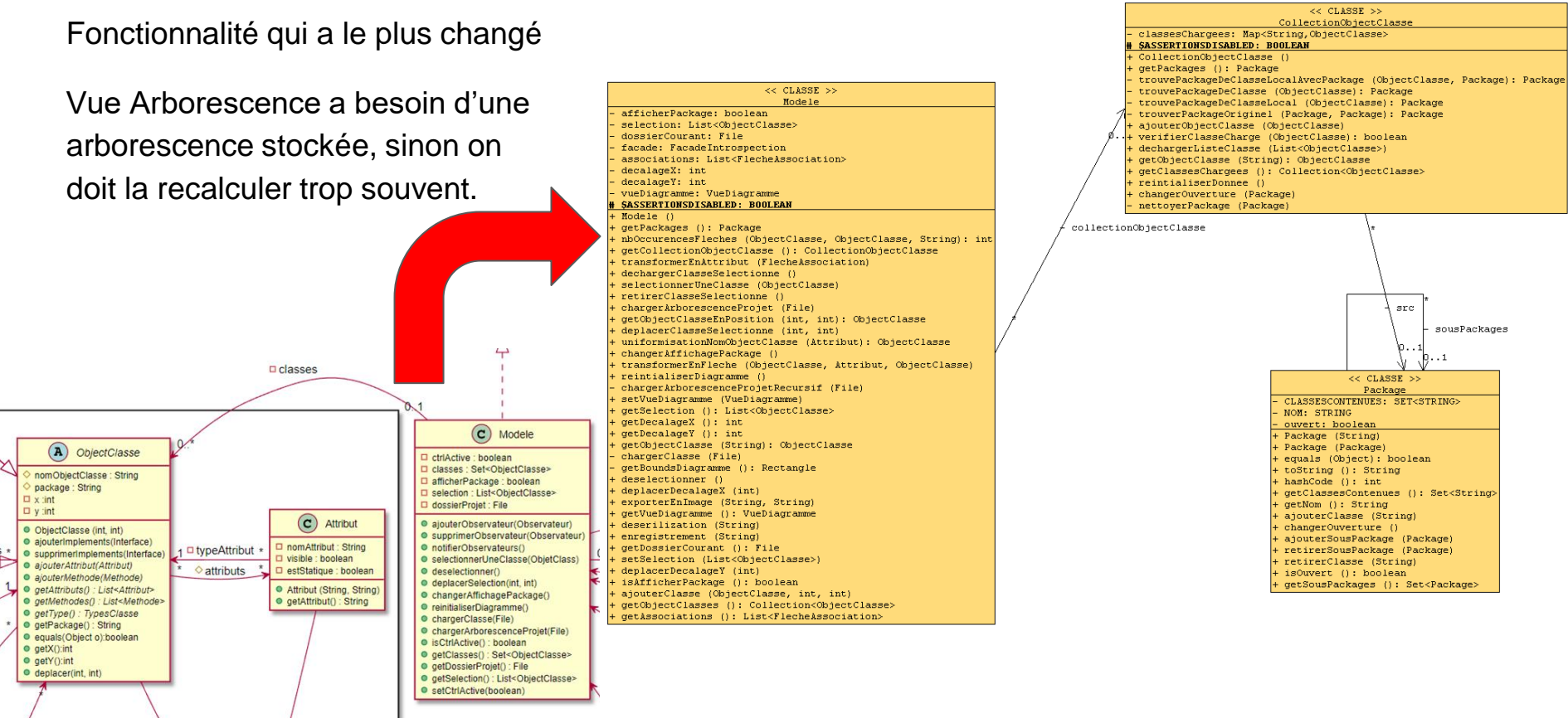




## Stocker les informations des classes du diagramme

Fonctionnalité qui a le plus changé

Vue Arborescence a besoin d'une arborescence stockée, sinon on doit la recalculer trop souvent.



# c Sélection d'une classe

Diagrammes de séquences pour sélectionner une classe

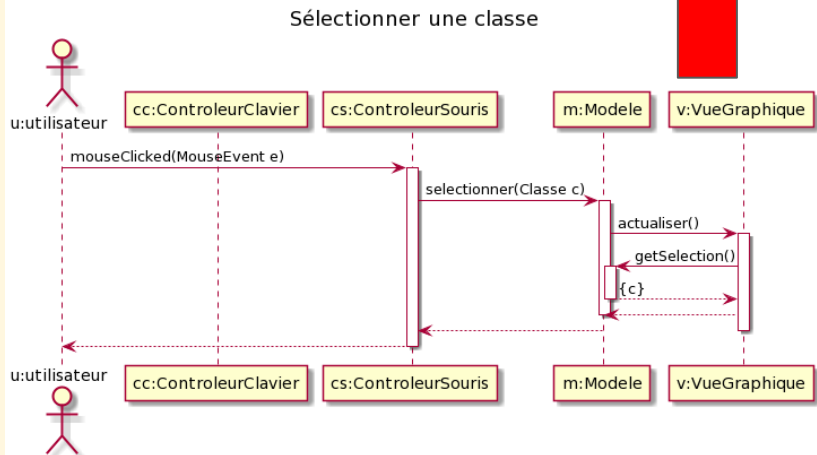
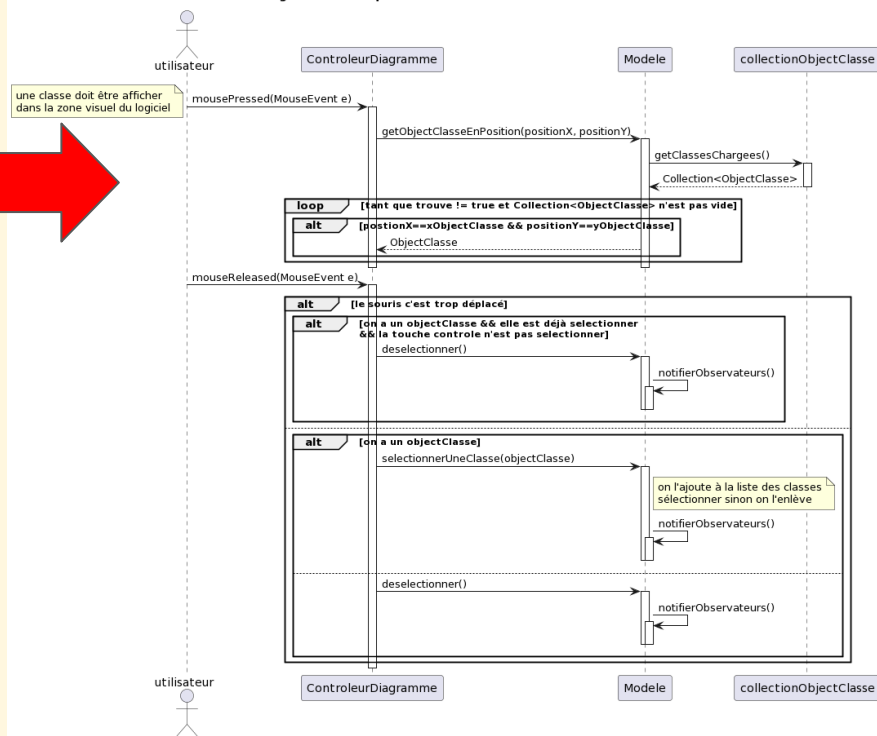
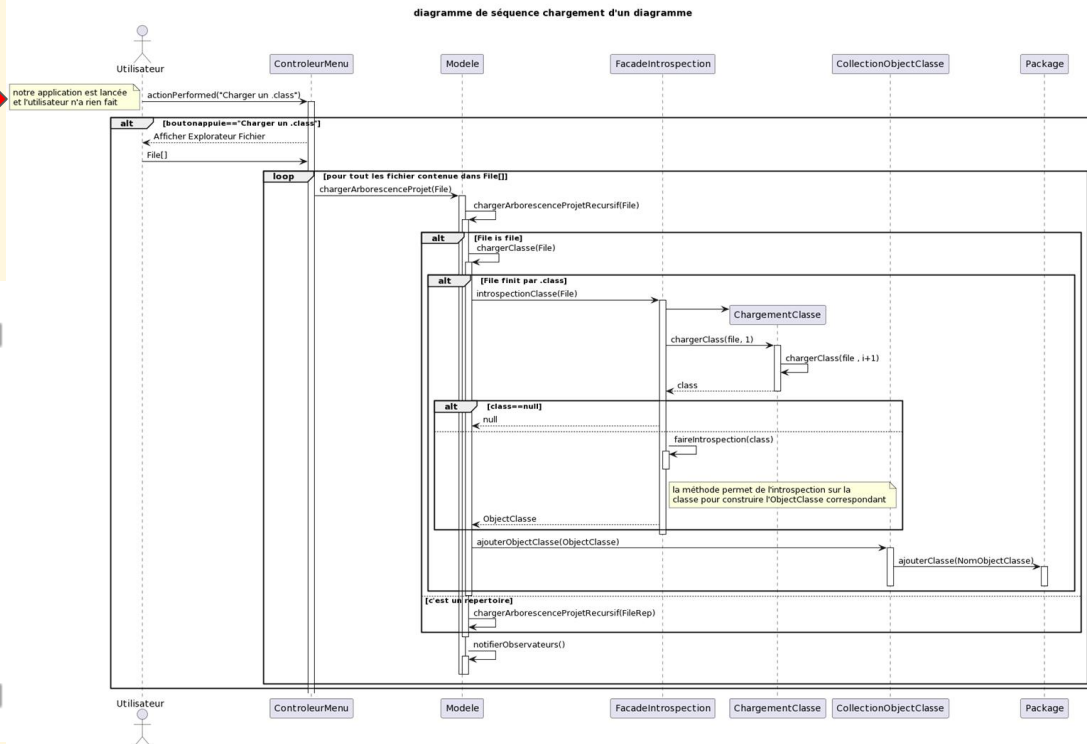
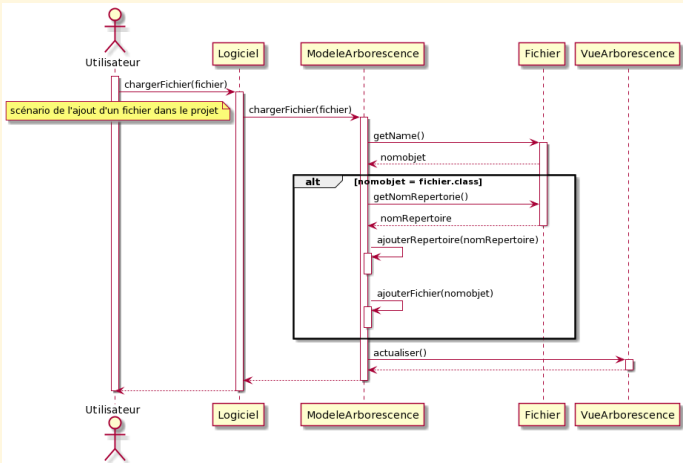


diagramme de séquence lors de la sélection d'une classe





## c Chargement d'une classe



## Conclusion

Pour conclure cette troisième et dernière itération :

- Version stable
- Affichage
- L'ajout des flèches complexe et cardinalité
- Des moyens de récupérer nos diagrammes (exportation et sauvegarde)

Nos critiques envers notre travail:

- Malgré de nombreux tests, il reste toujours de bug dans notre programme surtout au niveau de l'affichage. Il reste encore beaucoup de débogages à faire. Il reste aussi des cas particuliers non gérés.

Pour conclure ce projet tutoré :



- Notre application est entièrement utilisable mais il reste beaucoup de pistes pour l'améliorer.
- Le sujet est très enrichissant et fait un excellent projet tutoré, mais notre projet n'est pas assez stable pour servir de base à un futur projet tutoré

# ***Projet tutoré***

***Merci de nous avoir écoutés!***

