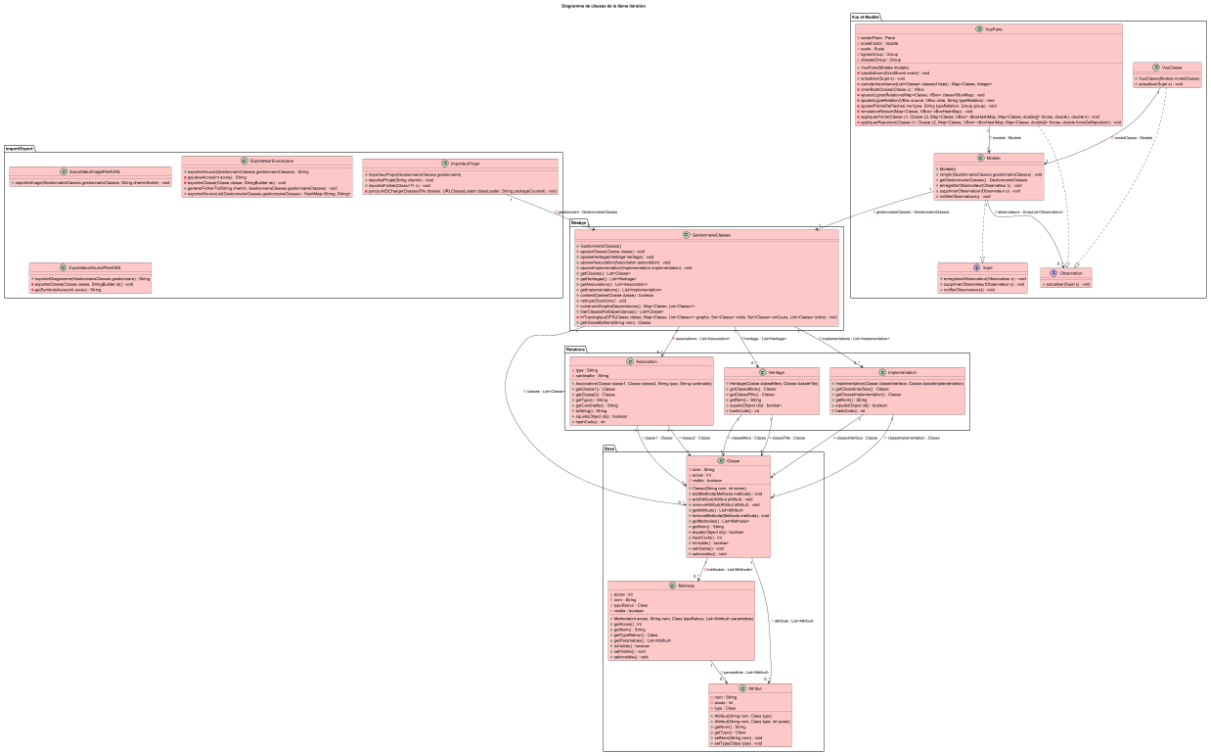
CADET Mattéo
RAY Marcelin
HENNEQUIN Baptiste
DIEUDONNE Quentin
S3B

Rapport SAE 3.01

Liste des fonctionnalités

- Importer des fichiers .class locaux
- Afficher le diagramme de classe du projet
- Exporter le projet source PlantUML
- Exporter le projet en image PlantUML
- Exporter le projet en squelette Java
- Déplacer les classes
- Afficher/Masquer les classes
- Afficher/Masquer les méthodes
- Importer des projets externes
- Créer une nouveau projet
- Ajouter des classes/méthodes
- Ajouter de l'héritage et de l'implémentation
- Affichage intelligent

Diagramme de classe final



Répartition du travail

Marcelin

- Base du projet : classes de bases Classe, Méthode, Attribut
- Gestionnaire de classes
- Exportateur en source PlantUML
- Base pour la visibilité des classes/méthodes
- Ajouter des classes/méthodes
- Créer un nouveau projet
- Ajout des exportateur dans l'application
- Gestion des erreurs pour les actions illégales

Quentin

- Beaucoup de corrections des classes déjà existantes
- Exportateur en image PlantUML
- Afficher/masquer les méthodes
- Ajouter des héritages et implémentations

Baptiste

- Main, model, vues, observateur
- Déplacer les classes
- Relations entre les classes
- Importer un projet externe
- Placement intelligent des classes

Mattéo

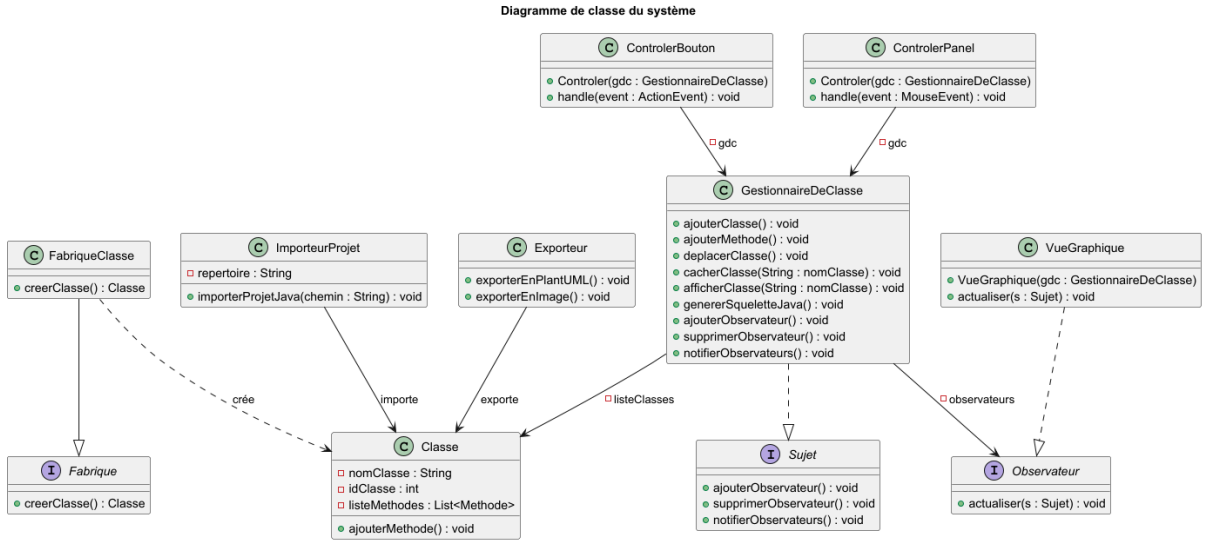
- Exportateur en source Java
- Afficher/masquer les classes
- Apparence des menus
- Apparence et bonne direction des flèches
- Affichage des cardinalités

Commun

- Importeur Projet
- Gestionnaire de classes
- Mise en forme des documents texte

Différences avec l'étude préalable

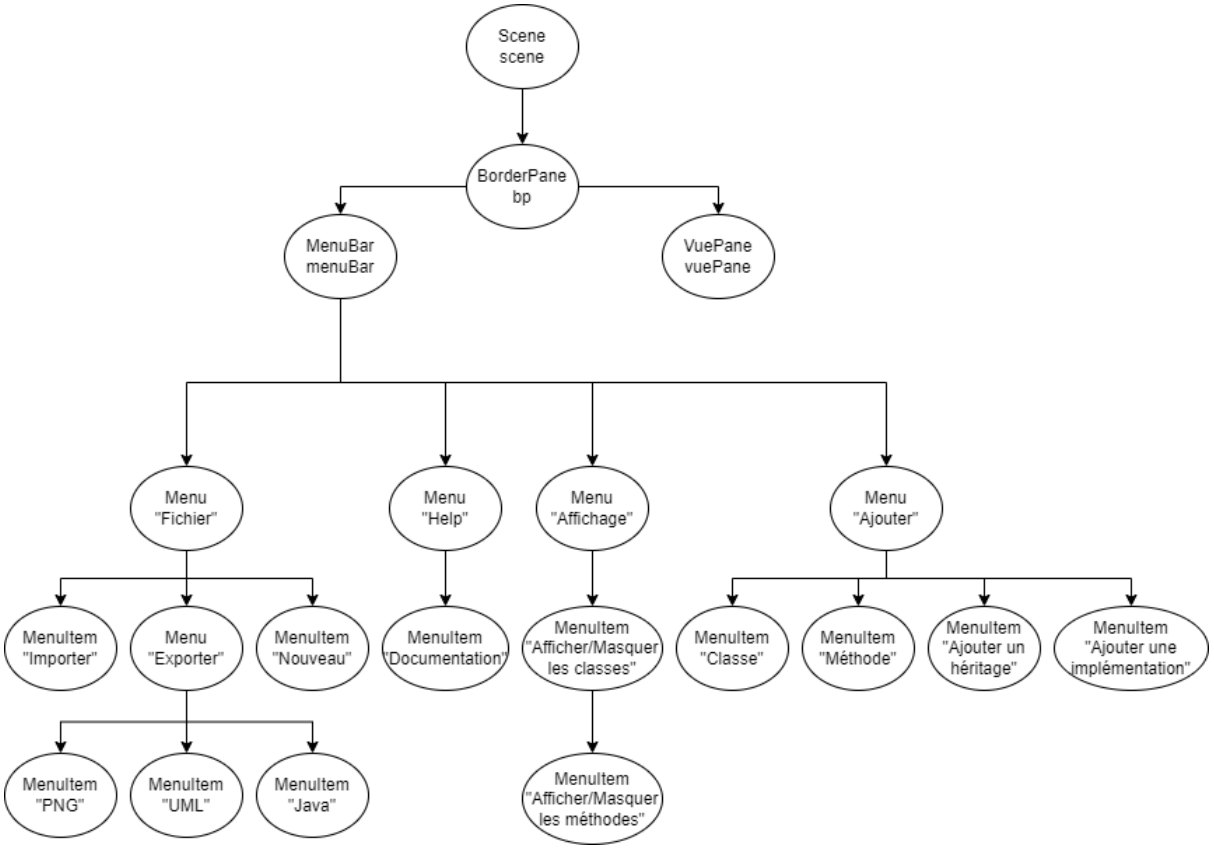
On crée le diagramme de classe à partir des fichiers .class plutôt que des fichiers .java.
Notre diagramme de classe final est complètement différent de celui établi pendant la phase d'analyse (pas de contrôleurs, plusieurs vues).



Patrons de conceptions utilisés

Nous avons utilisé le patron MVC qui gère toutes les interactions avec l'interface graphique.
Au final on utilise pas de patron Fabrique pour créer les classes, méthodes etc...

Graphe de scène



Problèmes rencontrés

Durant la première itération, nous avons oublié pour l'importateur d'importer les liens (Héritages, Associations, Implémentations…) entre les classes.
Au lieu de retourner un List<Classe>, nous devons retourner un GestionnaireClasses, ce qui a beaucoup retardé notre itération 2 : exporter en squeletteJava, sourcePlantUML, imagePlantUML.
Nous avons rencontré un problème qui a persisté sur les deux premières itérations : nous avions eu un problème de duplication des classes.
Dans les dernières itérations, nous avons voulu faire en sorte de pouvoir importer des projets externes, ce qui nous a obligé à revoir notre importateur de projet, qui utilisait les fichiers .java plutôt que les .class.
Le "ressort" pour l'affichage intelligent des classes est une difficulté qui persiste toujours car très complexe à comprendre et à mettre en place dans le projet.