

# **SAE 3.01 : Développement d'une application**

## **Conception**

Kevin Ngo  
Hugo Bena  
Arthur Hollebeque  
Hugo Rochedreux

# Bilan de l'application

- **Bilan de l'application**
- L'application permet de gérer et manipuler des classes dans un diagramme UML avec génération automatique de code Java.

## **Générer un diagramme de classe:**

- Glissement d'une classe de l'arborescence dans l'espace d'affichage du diagramme
- Import d'un dossier contenant des fichier .class
- Export en format PNG
- Export en format PlantUML
- Sauvegarder un diagramme généré via l'interface
- Chargement d'un diagramme déjà existant

# Bilan de l'application

## **Gestion des attributs**

Affichage des attributs avec leur nom, type et accessibilité (public/private).

Ajout d'attributs dans les fichiers .java avec mise à jour automatique.

## **Gestion des méthodes**

Affichage des méthodes avec leur nom, type de retour, accessibilité et paramètres.

Ajout de méthodes avec génération et insertion dans les fichiers .java

## **Gestion des dépendances**

Affichage ou masquage des relations entre classes avec des flèches.

Gestion de la cardinalité des dépendances.

## **Gestion des classes**

Suppression de classes du diagramme.

Déplacement des classes directement sur le diagramme.

Génération de code Mise à jour automatique et bien structurée des fichiers .java après chaque modification.

## **Interface contextuelle**

Menu contextuel (clic droit) pour accéder aux actions principales sur les classes.

Création rapide d'une classe via le menu clic droit sur le diagramme.

## **Autres fonctionnalités**

Changement de couleur des éléments.

Fonctionnalité d'annulation (retour arrière).

Effacement complet du diagramme.

# Répartition

- **Trello:**  
<https://trello.com/b/DP7Jcfaq/saeqdev>

The image shows a Trello board with five columns representing different iterations of a project. Each column contains several task cards, each with labels for status, priority, and type. The tasks are distributed across these iterations, showing a progression of work.

Première itération	2e iteration finit	3e iteration finit	4e iteration finit	5ème Itération finit
<ul style="list-style-type: none"><li><b>Fin</b> <b>Conception</b> Diagramme de séquence (Ouverture d'un fichier)</li><li><b>Fin</b> <b>Conception</b> Diagramme de séquence (Création fichier UML)</li><li><b>Fin</b> <b>Prioritaire</b> <b>Fonctionnalité</b> Création fichier UML</li><li><b>Fin</b> <b>Fonctionnalité</b> Structure générale</li><li><b>Fin</b> <b>Prioritaire</b> <b>Fonctionnalité</b></li></ul>	<ul style="list-style-type: none"><li><b>Fin</b> <b>Prioritaire</b> <b>Fonctionnalité</b> Création Tree View</li><li><b>Fin</b> <b>Prioritaire</b> <b>Fonctionnalité</b> Bouton exportation différents formats</li><li><b>Prioritaire</b> <b>Fonctionnalité</b> Ajouter une classe au diagramme (double clic)</li><li><b>Fin</b> <b>Prioritaire</b> <b>Fonctionnalité</b> Bouton sélection dossier</li></ul>	<ul style="list-style-type: none"><li><b>Fin</b> <b>Prioritaire</b> <b>En cours</b> <b>Fonctionnalité</b> Finir drag and drop</li><li><b>Fin</b> <b>Bof</b> <b>Prioritaire</b> <b>Fonctionnalité</b> Affichage du diagramme</li><li><b>Fin</b> <b>Semi-prioritaire</b> <b>Fonctionnalité</b> Affichage log</li><li><b>Fin</b> <b>Semi-prioritaire</b> <b>Fonctionnalité</b> menu contextuel lors d'un clic droit sur une classe du diagramme</li></ul>	<ul style="list-style-type: none"><li><b>Fin</b> <b>Java</b> <b>Prioritaire</b> <b>Fonctionnalité</b> Création dépendance entre classes/création flèches</li><li><b>Fin</b> <b>Fonctionnalité</b> Supprimer une classe du diagramme (easy)</li><li><b>Fin</b> <b>Fonctionnalité</b> finir l'arborescence de fichier (image, retirer .idea etc..)</li><li><b>Fin</b> <b>Fonctionnalité</b> Gérer la visibilité des attributs/méthodes/dépendances avec le menu contextuel</li></ul>	<ul style="list-style-type: none"><li><b>Fin</b> <b>Semi-prioritaire</b> <b>Fonctionnalité</b> Ctrl+z (facultatif)</li><li><b>Fin</b> <b>Fonctionnalité</b> <b>Optionnelle</b> save un screen du diagramme</li><li><b>Fin</b> <b>Java</b> <b>Prioritaire</b> Sérialise (sauvegarde d'un diagramme)</li><li><b>Fin</b> <b>Fonctionnalité</b> <b>Interface</b> <b>Optionnelle</b> Changer la couleur du diagramme</li><li><b>Fin</b> <b>Semi-prioritaire</b></li></ul>

# Itération 1

- **Génération d'une arborescence de classes à partir d'un fichier composite.**
- **Sauvegarde d'une classe en format PlantUML dans un fichier texte et affichage dans la console.**
- **Utilisation de l'introspection pour accéder aux informations d'un objet classe.**

# Itération 2

- **Exportation des classes au format source PlantUML.**
- **Import d'un dossier contenant des sous-dossiers et fichiers .class depuis n'importe quel répertoire.**
- **Visualisation d'une arborescence de fichiers créée à partir du dossier racine importé.**
- **Import d'un fichier .class sur le diagramme, visible sous forme de rectangle avec le nom de la classe.**

# Itération 3

- **Affichage des logs dans l'interface pour suivre les actions récentes de l'utilisateur.**
- **Import d'un fichier .class en glissant-déposant directement sur le diagramme.**
- **Effacement complet du diagramme.**
- **Affichage des attributs et des méthodes des classes sur le diagramme.**

# Itération 4

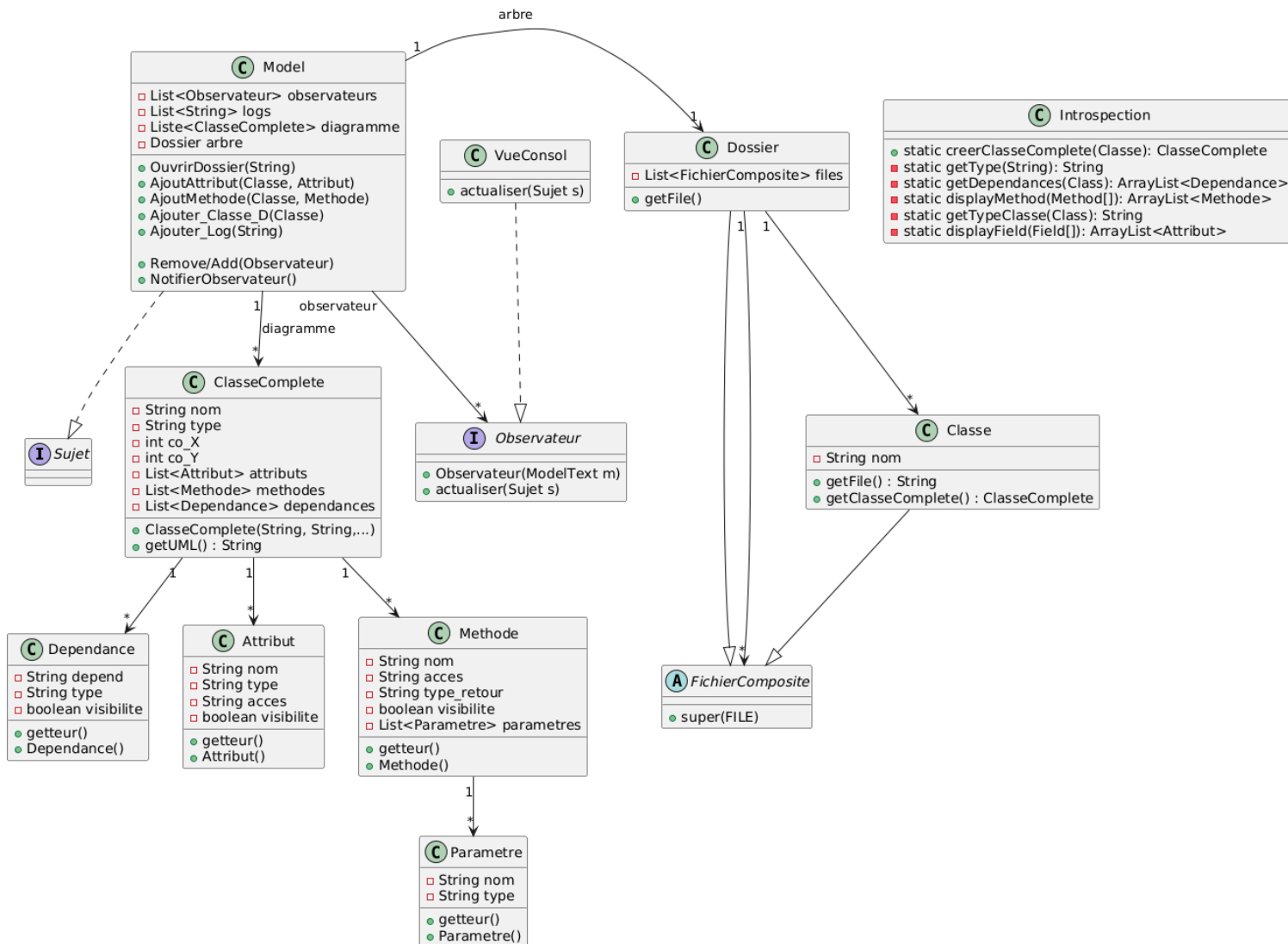
- **Gestion de la visibilité des attributs, méthodes et dépendances des classes (afficher/masquer).**
- **Déplacement dynamique des classes sur le diagramme via glisser-déposer.**
- **Visualisation des flèches représentant les dépendances entre classes (héritage, implémentation, association), actualisées en temps réel lors du déplacement des classes.**
- **Suppression d'une classe spécifique du diagramme.**



# Itération 5

- **Génération automatique du squelette de classes en fichiers .java.**
- **Ajout de méthodes et d'attributs directement dans les fichiers .java.**
- **Personnalisation de la couleur du diagramme.**
- **Sauvegarde du diagramme en capture d'écran.**
- **Sauvegarde et chargement d'un diagramme dans un format compatible avec l'application.**
- **Ajout d'attributs et de méthodes dans l'affichage des classes sur le diagramme.**
- **Fonctionnalité de retour arrière sur une pile de 20 actions.**
- **Ajout des constructeurs à l'affichage des classe du diagramme**

# Diagramme de classe Analyse



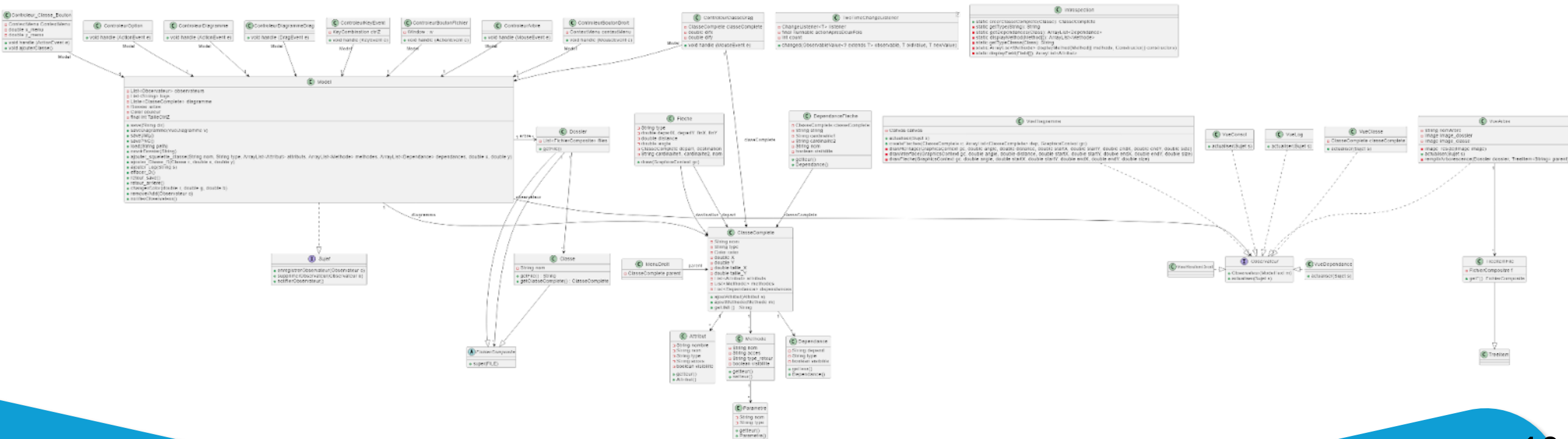
Patron Stratégie pour exporter μ  
le diagramme en image

Patron MVC pour tout  
le visuel et le côté utilisateur

Patron Composite pour  
les Dossier/Classe dans l'arbre

# Diagramme de classe final

Aller sur [PlantUML](#) pour plus de visibilité



# Conclusion

## **Parfaire**

Planning  
Diagramme de classe  
Code plus lisible

## **Réalisations**

Trello  
Toutes les tâches réalisées  
Tâches bonus  
Pas de BUG