Quentin Forestier

TB – 29.03.2022

Cahier des charges

Travail de bachelor



Table des matières

[Cahier des charges 2](#_Toc99480937)

[Jalon 2](#_Toc99480938)

[Fonctionnalités 2](#_Toc99480939)

# Cahier des charges

Le but de ce travail de Bachelor est de répliqué et amélioré les fonctionnalités de Slyum, un éditeur de diagramme de classes UML développé en Java à la HEIG-VD.

Un journal de travail doit être tenu le long du projet, répertoriant les activités de la semaine. Un rapport intermédiaire est à rendre le 16 mai 2022, et le travail doit être rendu le 29 juillet 2022.

## Jalon

Dans un premier temps, il faudra définir un meta-schéma au moyen d’un diagramme de classes.

Dans un deuxième temps, il s’agira de se familiariser avec le Framework Play !, et de voir dans quelle mesure il est possible d’utiliser les websockets afin de pouvoir collaborer. Une fois cela fait, il faudra choisir si oui ou non, il est possible d’avoir une coopération en directe sur l’édition de diagrammes. Un second diagramme viendra s’ajouter afin de modéliser les messages échangés entre le serveur et le client.

Dans un troisième temps, il faudra effectuer une recherche sur les différentes librairies Javascript permettant un affichage et modification simple et intuitif du diagramme. Puis, un autre diagramme de classes viendra compléter le meta-schéma, représentant les informations graphiques.

Ensuite, il faudra pouvoir sauvegarder les diagrammes de classes dans une base de données PostgreSQL. Pour cela, il faudra rechercher l’ORM le plus adapté et le mettre en place.

Une fois la conception du meta-schéma, ainsi que la familiarisation avec Play! effectué, l’application et ses fonctionnalités seront codés en Java pour la majeure partie, et en Javascript pour l’affichage du diagramme. L’évolution de l’application suivra les jalons, de sorte que chaque étape soit déjà un résultat.

## Fonctionnalités

* Création de classes, interfaces, énumération, classes abstraites
* Création d’attributs et de méthodes (abstraites ou non)
* Création d’associations (binaire, n-aire, compositions, agrégation, classes d’association)
* Création de relation d’héritage
* Création de relation de dépendances
* Affichage des éléments sous format graphique
* Possibilité de modifier graphiquement le diagramme
* Possibilité d’écrire les attributs/fonctions sous forme de texte, qui sera ensuite transformer en entité
* Exportation sous format graphique
* Sérialisation/Désérialisation des diagrammes
* Possibilité d’effectuer une annulation de la dernière action
* Mise en place de raccourcis claviers
* Inscription/Connexion
* Création de projet
* Création de groupes d’utilisateurs
* Gestion des droits dans le groupe
* Gestion des membres du groupe
* Quitter un groupe
* Ouverture des diagrammes de classes
* Coopération directe / indirecte sur un diagramme
  + Dans le cas d’une coopération instantanée, l’accès aux autres utilisateurs sera bloqué sur l’élément du diagramme sélectionné.
  + Dans le cas d’une coopération indirecte, l’accès au diagramme sera bloqué si un autre utilisateur travaille déjà dessus