



Professeur encadrant: M.Tewfik ZIADI
UE: LU3IN104
Génie Logiciel

Décembre 2021

Projet réalisé par:

EL HITARY Doha

HABBI Sarah

SIVANATHAN Siyani

CHABLIS Dalil

DJORDJEVIC Darko

HASSANEIN Youssef

Sommaire

I. Introduction.....	3
II. Rapports individuels.....	4
III. Conclusion.....	10

I. Introduction

Le projet iBuySU est un projet qui vise à permettre aux étudiants de la licence générale informatique L3 DANT (Sorbonne Université) d'appliquer le cours de modélisation UML en génie logiciel.

L'objectif principal de la mission est de développer une application web en adoptant les concepts de modélisation proposés par UML. A partir du cahier des charges, nous devons concevoir les diagrammes de cas d'utilisation, de classe et de séquence. L'idée est de générer automatiquement une grande partie du code à partir de nos diagrammes de classes en appliquant les notions de forward et reverse engineering.

Pour la réalisation, nous avons décidé de structurer notre application en deux parties, un back-end sous forme d'API REST en **JAVA** avec le framework **Spring boot**. Et un front-end en **Typescript** avec le framework **Angular**.

Le choix d'utiliser ces deux frameworks nous permettra non seulement de découvrir, d'apprendre, de gagner du temps, mais aussi d'adopter les bonnes pratiques de développement imposées par ces frameworks.

Bien sûr, nous gardons à l'esprit l'objectif principal qui est la génération de code **Java** à partir de nos modèles (Back-end).

Membres de l'équipe

- [Dalil CHABLIS](#) : Chef de projet
- Doha EL HITARY
- Sarah HABBI
- Siyani SIVANATHAN
- Darko DJORDJEVIC
- Youssef HASSANEIN

Contexte

La société iBuySU souhaite mettre en place un site commercial permettant à des millions d'acheteurs et de vendeurs d'acheter et de vendre tout type d'objet dans le monde.

- Tout utilisateur peut visiter le site iBuySU sans être inscrit.
- Les utilisateurs peuvent s'inscrire pour acheter et/ou vendre des articles.
- Les utilisateurs peuvent rechercher des articles par catégorie ou par mots-clés.
- Les acheteurs peuvent se connecter à leur compte d'acheteur pour soit enchérir sur des articles, soit les acheter.
- Les vendeurs peuvent se connecter à leur compte de vendeur pour soit vendre des articles aux enchères, soit directement.
- Les acheteurs et les vendeurs peuvent s'évaluer mutuellement.

II. Rapports individuels

1. EL HITARY Doha

a) Partie modélisation:

J'ai travaillé principalement sur la partie Modelio/Modélisation.

J'ai participé dans les missions suivantes:

- Diagramme de use case.
- Des diagrammes de séquence de conception.
- Des diagrammes de classe.
- Trouver un rt.jar, qui va nous nous permettre la génération de tous les types. (car le rt.jar utilisé au début ne génèrait pas le type "String")
- La génération de code/et mise à jour du code .
- Le reverse engineering.
- Vérification du code et diagramme

b) Partie développement:

Le développement de la partie évaluation, sur les 2 niveaux front et back.

2. HABBI Sarah

a) Partie modélisation

- Etude du modèle MVC
- Diagramme de use case
- Diagrammes de séquence de conception:
 - inscription d'utilisateur (acheteur et vendeur)
 - vendre des articles aux enchères
 - recherche d'article par catégorie ou par mot-clé
 - demande d'achat d'un article par l'acheteur
 - acceptation ou refus d'une demande d'achat par le vendeur

a) Partie développement

Back-end

- Apprendre à utiliser le framework Spring boot
- Ajouter des utilisateurs (vendeur ou acheteur) dans la base de donnée
- Ajouter des articles dans la base de donnée
- Trouver des articles avec un mot clé ou la catégorie.
- Ajout d'article aux enchères dans la base de donnée

Front-end

- Apprendre à utiliser le framework Angular avec TypeScript
- Créer un compte en tant qu'acheteur ou vendeur en remplissant un formulaire
- Se connecter à son compte
- Se déconnecter de son compte
- Visiter le profil d'un vendeur et voir ses articles en vente
- Afficher les articles d'un vendeur dans son espace personnel

3. **SIVANATHAN Siyani**

a) Partie modélisation

- Diagramme de fonctionnalités
- Etude du modèle MVC
- Diagramme de classe analyse (partie model)
- Diagramme de classe de conception sur plusieurs niveaux
- Diagramme de séquence
- Reverse engineering
- Mise à jour de code

b) Partie développement

- Apprentissage du Typescript
- Apprendre à utiliser Spring Boot
- Filtrer les articles selon l'utilisateur dans le front
- Programmation du use case enchérir dans le front et back
- Tester le système à partir de chaque use case

Parcourir Ibuysu comme un utilisateur

4. **CHABLIS Dalil**

a) Partie modélisation

- Diagramme de use case
- Diagramme de classes
- Diagrammes de séquence de conception
- Test génération de code et reverse engineering
- Ajout des bibliothèques Java, Spring boot, lombok... dans le rt.jar

b) Partie développement

- Choix des outils (Spring boot, Angular)
- Mise en place de l'environnement de développement sur gitlab (branche protégé, merge request...).
- Structuration du projet
- Gestion du projet.

Back-end

- Configuration du framework
- Ajouts des annotations de spring boot
- Gestion des dépendances circulaire
- Gestion des problèmes CORS (Cross-origin resource sharing)
- Inscription utilisateurs
- Authentification
- Evaluation
- Listes des articles

Front-end

- Configuration du framework
- Inscription utilisateurs
- Authentification
- Evaluation
- Listes des articles

5. DJORDJEVIC Darko

a) Partie modélisation

- Diagramme de use case
- Diagramme de classes
- Diagrammes de séquence de conception
- Test génération de code et reverse engineering

b) Partie développement

Back-end

- ajouter des users (vendeur ou acheteur) dans la base de donnée
- mettre à jour l'article (statuts et acheteur) après une demande d'achat
- mettre à jour l'article après que le seller ai accepté/refusé une purchase request
- mettre à jour l'article (prix et dernier enchérisseur) après qu'un acheteur ait enchéri.

Front-end

- Créer un compte en tant qu'acheteur ou vendeur en remplissant un formulaire
- Authentification
- Ajout d'un article en vente direct ou enchère (formulaire à remplir)
- Se déconnecter de son compte
- Ajout de la fonctionnalité demande d'achat
- Ajout de la fonctionnalité accepter/refuser la demande d'achat
- Ajout de la fonctionnalité enchérir sur un article
- Gérer la fin de l'enchère (le vendeur doit accepter ou refuser)

6. HASSANEIN Youssef

a) Partie modélisation

- Diagrammes de fonctionnalités
- Diagrammes de classe
- Diagrammes de séquence de conception
- Tester le round-trip engineering (reverse puis génération)

b) Partie développement

Back-end

- Apprendre à utiliser Spring Boot
- Ajouter le endpoint permettant les achats directs (purchase requests)
- Ajouter le endpoint permettant d'enchérir sur un article (auction requests)
- Ajouter le endpoint permettant de répondre à la demande d'achat ou d'enchère
- Ajouter le endpoint permettant d'obtenir les articles qui n'ont pas encore été achetés

Front-end

- Apprendre à utiliser Angular et Typescript
- Ajouter des filtres pour la recherche d'articles (par catégories, par mots clés ou par mode de vente (achat direct ou enchères))
- Ajouter des boutons pour la demande d'achat direct ou d'enchères (mode acheteur)
- Ajouter des boutons pour l'acceptation/refus de la vente d'un article (mode vendeur)

c) Tests manuels

- Tester l'application dans sa globalité

III. Conclusion

Ce projet nous a permis d'appliquer les concepts de la modélisation UML, mais aussi de tester l'outil modelio pour la génération de code et le reverse engineering.

Nous avons essayé de nous mettre dans un contexte professionnel en utilisant les frameworks Spring boot et Angular.

La génération de code a été faite principalement pour le backend (Spring boot) car l'outil modelio ne permettait que la génération de code java.

Nous sommes tous d'accord pour dire qu'un outil de génération de code et de modélisation peut être un grand atout pour les équipes de développement.

Concernant l'outil modelio, nous ne recommandons pas son utilisation pour le moment car il nous a retardés lors du développement.

En effet, cet outil ne fonctionne pas très bien sur windows.

Très lent sur linux lorsque l'on commence à ajouter beaucoup de classes.

Et pas proposé sur Mac.

Ce qui veut dire que pendant notre projet, tout le monde ne pouvait pas travailler sur modelio. Et une seule personne pouvait travailler à la fois et partager le zip du projet avec les autres.

La documentation html générée avec modelio est très pratique malgré quelques bugs lorsque le projet comporte un grand nombre de packages.

Néanmoins, nous sommes satisfaits de notre travail et sommes convaincus de l'utilité de tels outils.