



**Génie Logiciel**

# **BOOK'IT**

Ruben Attal - Thomas Aube - Marius Colin  
Marine Meunier - Simon Weber



IUT de Paris  
L3 MIAGE Apprentissage  
2019/2020

# Table des matières

Introduction .....	3
Le projet.....	3
Contraintes à respecter .....	3
Architecture.....	4
Organisation et conception.....	5
Erreur quant au choix du sujet .....	5
Réorganisation .....	5
Partage de ressources et technologies utilisées .....	8
Partage de ressources .....	8
Technologies utilisées .....	8
Gestion de dépendances.....	8
Outil de tests.....	9
Intégration continue .....	9
Outil de contrôle de la qualité.....	10
Évolutions possibles.....	11
Conclusion.....	12
Nos avis.....	13
Thomas .....	13
Simon .....	13
Marius .....	14
Ruben.....	14
Marine.....	14

## Introduction

Book'IT est une application de gestion des salles de l'IUT de Paris. Développée dans le cadre du cours de Génie Logiciel, elle permet de répondre à la problématique : *Comment optimiser la réservation de salle au sein de l'IUT ?*

Nous sommes 5 étudiants de L3 MAGE en apprentissage à avoir participé à ce projet. Son objectif est de nous permettre de mettre en pratique les connaissances acquises lors des cours. Ces connaissances s'articulent autour de la gestion et de l'organisation de projet mais aussi autour de l'optimisation du code et de l'intégration de bonnes pratiques très largement utilisées dans les entreprises.

Le choix du sujet est libre. Nous avons décidé de nous orienter vers un thème en lien direct avec notre quotidien d'étudiants.

L'objectif de cette synthèse de projet est de rendre compte de notre organisation, des difficultés que nous avons rencontrées, de la manière dont nous les avons solutionnées et des points perfectibles. Pour introduire ce dossier nous feront un point sur l'architecture que nous avons mise en place et les consignes qui ont été notre ligne conductrice.

## Le projet

### Contraintes à respecter

- Utilisation de Git
- Utilisation de Markdown
- Rédaction des besoins
  - User stories
  - Critères d'acceptation
- Tests unitaires
- Réalisation en Java, JavaScript ou Go
- Respect de l'Inversion of Control (Spring...)
- Utilisation d'un outil de gestion des dépendances (Maven...)
- Mise en place d'une documentation
- Bonus : ajout d'une fonctionnalité

Book'IT est une application console qui peut être utilisée via la console d'une IDE ou via l'interface de commande de son PC et l'utilisation d'un JAR.

Il a évolué au cours du développement : modifications du nom des classes et des variables notamment.



4

## Organisation et conception

### Erreur quant au choix du sujet 🚫

Influencés par la période des fêtes de fin d'année et la difficulté à trouver des cadeaux à offrir, nous avons d'abord choisi de développer une application qui répondrait à cette problématique, Giftlist.

Une fois le sujet choisi, nous avons commencé à imaginer les différentes user stories et les fonctionnalités en lien à développer. C'est lors de cette phase que nous avons rencontré des difficultés.

Premièrement, peut-être par manque d'imagination, nous avons eu du mal à définir plusieurs fonctionnalités pertinentes. Nous pensions, grâce aux connaissances acquises lors du module de PAA2, utiliser une API pour récupérer des idées de produits. Finalement, le projet s'éloignait de notre problématique initiale et devenait une interface de recherches de produits.

Face à ce manque d'inspiration, et au manque d'originalité de cette application, nous avons décidé après plus de deux semaines de changer de sujet.

### Réorganisation 🤝

L'idée de développer une application de gestion des salles de l'IUT nous est alors arrivée assez vite : Book'IT. En tant qu'étudiants de l'IUT nous avons pu définir les fonctionnalités très rapidement. En effet, nous faisons partie des potentiels principaux utilisateurs !

Nous avons donc défini les fonctionnalités tous ensemble pendant deux séances. Les 3 fonctionnalités principales sont détaillées dans le cahier des charges.

En parallèle, nous avons défini les rôles des membres de l'équipe. Les tâches ont été séparées selon les rôles suivants :

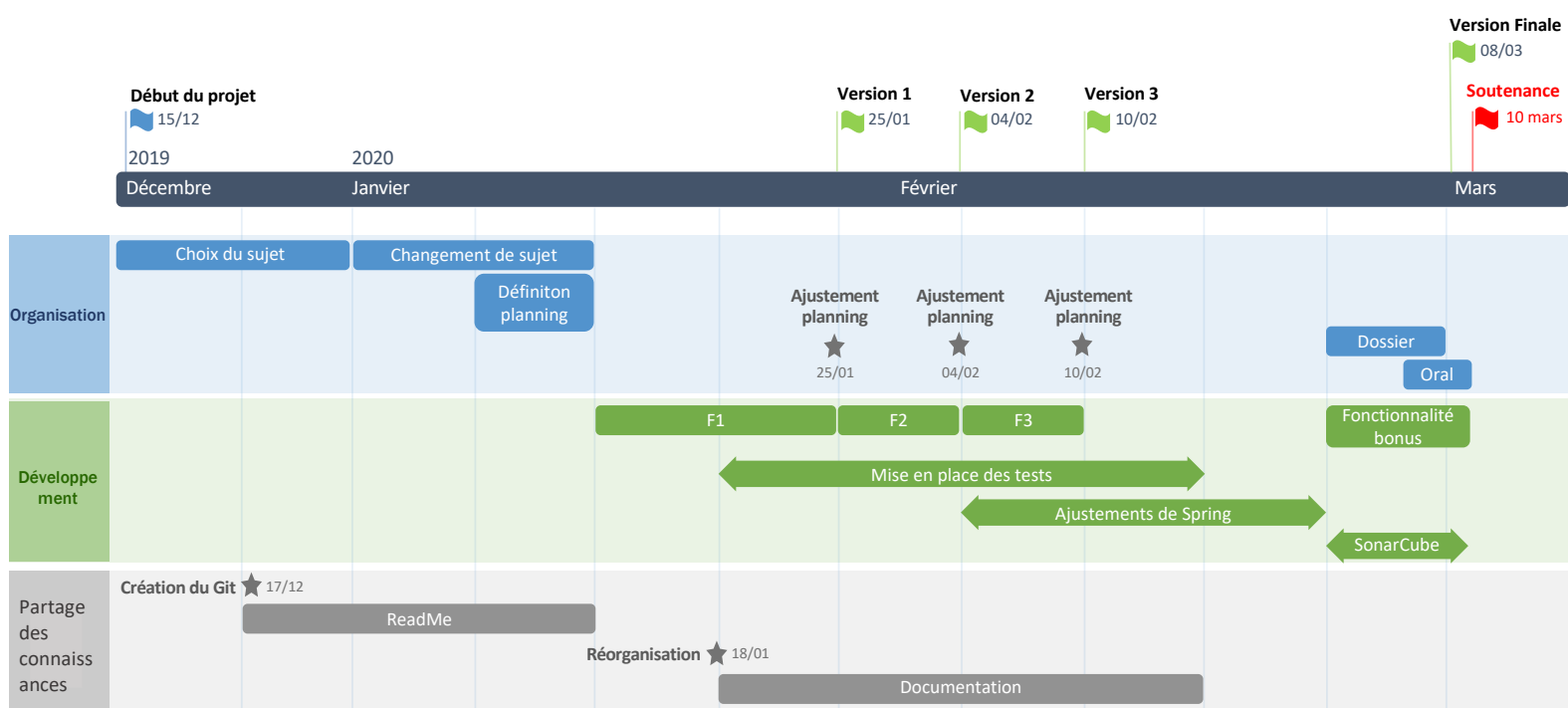
- Chef de projet 🧑
- Responsable GitHub 🧑
- Développeur 🧑🧑🧑

Les étudiants à l'aise en programmation assurent le rôle de développeur, un étudiant familier avec Git est chargé de l'organiser. Un étudiant chef de projet s'occupe, lui, de la mise en place de l'organisation et de la rédaction des documents.

La première étape pour le chef de projet a été de mettre en place une roadmap. Elle permet de fixer les échéances des différentes étapes et versions du projet. Elle concorde avec les rôles que nous avons définis et est donc séparée en 3 parties :

- Organisation
- Développement
- Documentation

Mise en place lors de la 4<sup>ième</sup> semaine de projet, elle prend aussi en compte les avancements antérieurs.



*Roadmap finale du projet*




Bien conscients des imprévus qu'un projet de plusieurs mois peut comporter, nous avons prévu dès l'élaboration de la roadmap des points d'ajustement du planning. Ces points ont comblé les deux semaines de rabais prévues pour palier à d'éventuels retards. Ils ont été effectués à la fin du développement de chaque version.

L'organisation des rôles a rapidement montré ses limites. Lors de l'élaboration de la première fonctionnalité, le manque de définition des rôles des 3 développeurs a entraîné le développement de cette fonctionnalité par un seul membre de l'équipe.

Nous avons affiné notre organisation grâce à deux outils :



- Une roadmap qui retrace les différentes versions de l'application.
- Les feedbacks des développeurs après le développement de la première version de l'application.

Après le développement de la première version, le plus urgent était de distinguer les rôles des différents développeurs. Ainsi les 3 nouveaux postes de développeurs sont :

- Développeur Back 
- Responsable Spring 
- Responsable des tests unitaires 

Les deux autres rôles de notre organisation initiale n'ont pas changé. Le rôle du responsable de GitHub s'est précisé. Lors de chaque séance, il faisait un point de vérifications et de modifications avec les développeurs pour ajouter ensuite leurs avancées au git.

En plus de la roadmap, et pour mettre en place les fonctionnalités de l'application l'équipe des développeurs s'est appuyée sur des tableaux de découpages des étapes. Par exemple, voici le tableau du développement de la réorganisation du code pour l'intégration Spring :

Fonctionnalité n°1 : Recherche d'une salle			
Fonctionnalité	Statut	Responsable	Commentaire
Création d'un service ArgumentParser	Terminée 	Marius	/
Création du contrôleur User	Terminée 	Thomas	/
Création du contrôleur Rooms	Terminée 	Thomas	/
Création de l'objet Command	Terminée 	Marius	/

Cet exemple montre comment nous avons découpé le travail pour chaque développement d'une nouvelle version. Les commentaires éventuels permettent au chef de projet de faire des points avec les responsables de celles-ci pour débloquer des situations. La colonne statut de la tâche a elle aussi une grande importance et permet de veiller au respect des deadlines de la roadmap.

## Partage de ressources et technologies utilisées



Pour le partage de ressources nous avons, comme demandé, utilisé Git. Nous avons choisi d'utiliser GitHub. Seulement deux membres de l'équipe avaient déjà utilisé cet outil. Comme évoqué plutôt, nous avons décidé dès le début du projet de nommer un membre de l'équipe Responsable Git.

Nous avons débuté le projet avec des aprioris quant à l'utilité de l'utilisation de Git.

Les avantages de GitHub que la réalisation de ce projet nous a permis de voir :

- Conservation des anciennes versions du code
- Vision claire de l'évolution de l'application

Compte tenu de notre organisation, et du fait que seulement un membre de l'équipe modifiait le git, nous pensons ne pas avoir utilisé cet outil autant que possible. Notamment, il aurait été plus simple d'identifier les tâches de chacun et de les répartir si chaque personne avait fait évoluer sa branche comme envisagé au tout début. En d'autres termes, le travail en parallèle aurait été simplifié.

Pour partager les ressources liées à l'organisation du projet, nous avons créé un document sur Google Drive. Au fur et à mesure des avancées, les tableaux de développements étaient complétés (voir exemple présenté dans la partie réorganisation).

## Technologies utilisées



Pour l'outil de gestion des dépendances, nous avons choisi d'utiliser Maven. L'outil avait été présenté en cours. Son utilisation n'a pas engendré de difficulté particulière. Nous nous sommes beaucoup aidés de ce qui est disponible dans le git mis à disposition par le professeur.

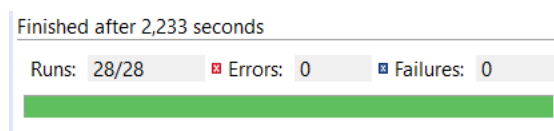


Pour réaliser les tests nous avons utilisé JUnit qui est un Framework de test unitaire pour le langage Java.

Compte tenu de notre organisation, nous avons choisi de ne pas pratiquer de TDD (Test Driven Development). En revanche, nous nous sommes assurés d'avoir des tests performants, rapides et maintenus.

Partout où c'est pertinent, nous avons testé autant que possible les fonctions du projet. Évidemment, nous n'avons testé que quand cela était pertinent, la classe Commande n'a par exemple pas été testée. De cette façon nous avons identifié des dysfonctionnements et les avons corrigés.

Chaque test est explicite et possède un scénario possible. Excepté le dernier point, les tests sont dits de type FIRST, c'est à dire qu'ils sont rapides, isolés, répétables, s'auto-valident.



*Capture d'écran de la barre de réalisation des tests*

Nous avons réalisé 28 tests qui sont exécutés en un peu plus de 2 secondes et qui sont tous fonctionnels.

Nous avons aussi utilisé Mockito, un Framework de mock pour les tests. Il permet de tester des objets, de simuler leur comportement et de rendre le test unitaire.

L'intégration continue n'a pas été mise en place dès le début du projet.

Pour l'intégration continue, nous avons utilisé Spring. L'utilisation de cet outil est ce qui nous a posé le plus de problèmes. En effet, avant de définir les rôles des développeurs plus précisément, les premières fonctionnalités ont été développées sans Spring (voir roadmap).

Nous avons d'abord eu du mal à comprendre son utilité. L'outil a donc été laissé de côté.

Nous avons donc commencé par développer un projet Java sans Spring. Une fois que les premières fonctionnalités étaient opérationnelles nous avons décidé qu'il était temps de prendre en compte Spring dans notre développement.

Au même moment, nous avons redéfini les rôles des développeurs. Cela a été l'occasion de nommer un Responsable Spring. Un projet Spring avait été créé à la fin du développement des premières fonctionnalités et il a servi de base au responsable.

Ce mode d'organisation a permis de se plonger dans cet outil et de mieux comprendre son objectif et la manière dont il fallait l'utiliser.

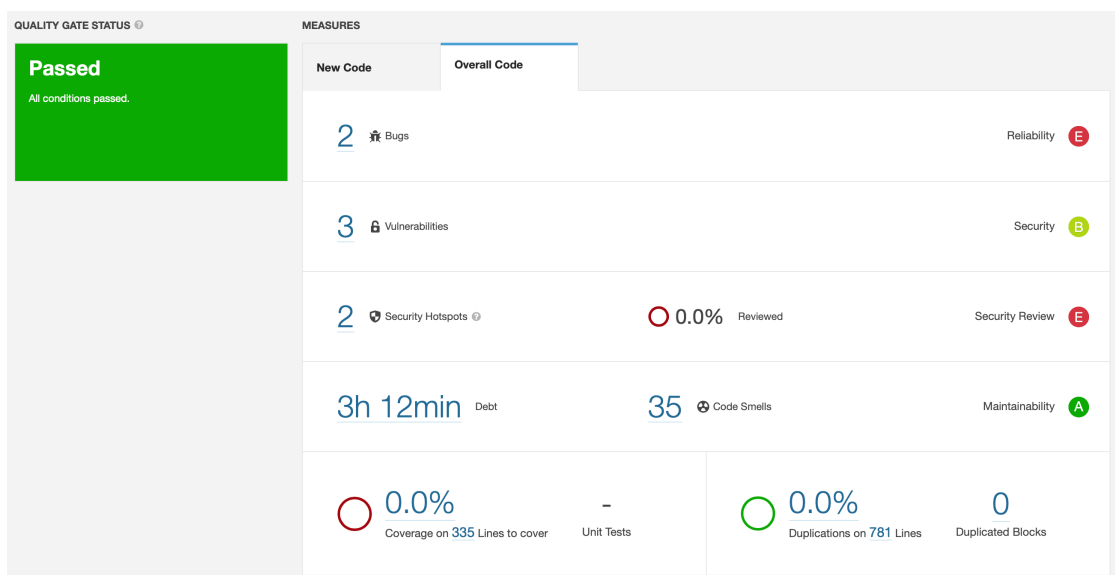
### Outil de contrôle de la qualité **sonarqube**

Grâce à la présentation d'un étudiant lors du dernier cours avant les soutenances, nous avons décidé d'utiliser l'outil SonarQube. Cet outil de qualimétrie a été très simple à installer et à configurer.

Il nous a permis de détecter plusieurs problèmes dans notre code. Il a par exemple pointé plusieurs variables inutilisées, imports inutiles et répétitions de chaînes de caractères à mettre en variable.

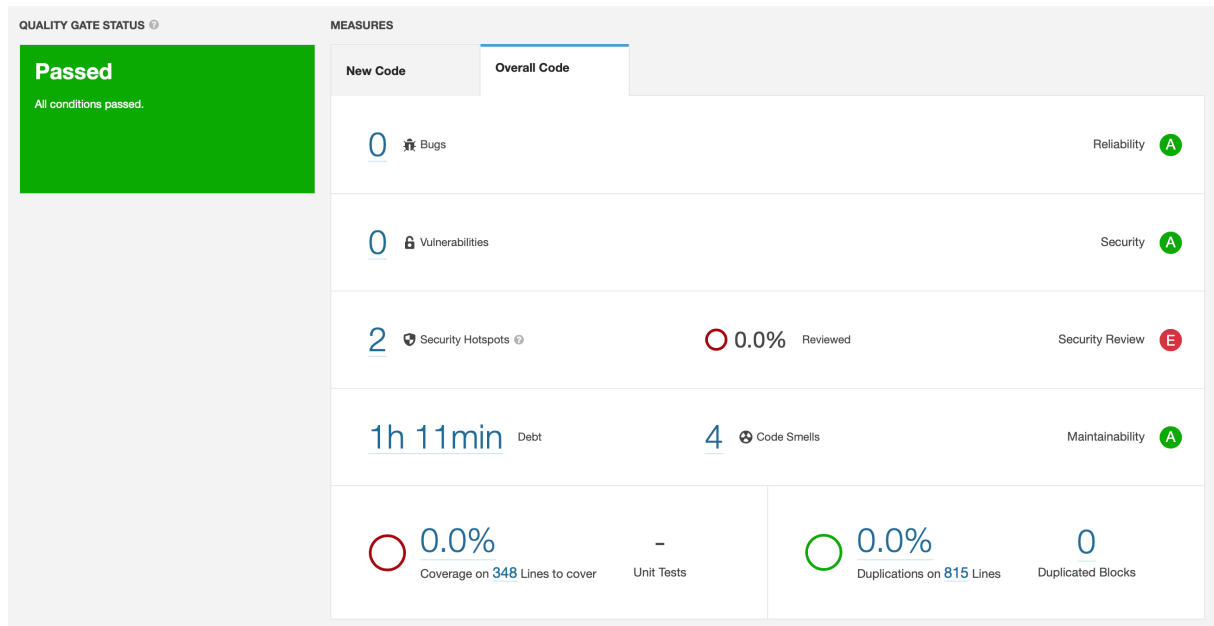
Le grand plus de cet outil est qu'il propose des solutions à chaque erreur détectée.

Voici ce que SonarQube affichait avant que nous corrigions les bugs :



*Capture d'écran SonarQube avant*

Voici ce que nous avons obtenu après avoir géré les bugs :



*Capture d'écran SonarQube après*

Les deux remarques quant à la sécurité ne peuvent pas être corrigées car elles ne sont pas compatibles avec le fait que nous avons développé une application console.

## Évolutions possibles

Les possibilités d'évolutions de l'application sont nombreuses. Quand nous définissions les fonctionnalités de l'application, et lorsque nous l'avons développée nous avons pensé à plusieurs autres fonctionnalités.

- La mise en place d'un régime de priorité/autorisation en fonction du statut de l'utilisateur (professeur ou étudiant)
- L'optimisation de la saisie de date pour ne pas obliger la saisie d'un intervalle
- L'encodage

Parce que ces éléments n'étaient pas prévus dans la roadmap, nous n'avons pas pu les intégrer au planning pour les mettre en œuvre.

## Conclusion

La réalisation de ce projet de génie logiciel nous a permis de mettre en pratique les enseignements piliers de notre formation MIAGE. Nous avons pu évoluer autour des étapes de la mise en œuvre d'un projet informatique : trouver un concept, le modéliser, le développer et le faire évoluer en équipe.

Contrairement à nos appréhensions, le développement de ce projet avec 3h30 de cours par semaine pendant environ 2 mois est réalisable avec une bonne organisation. Influencés par nos expériences précédentes pendant lesquelles le développement de l'application s'effectuait uniquement en dehors des heures de cours, nous pensions que ce mode de fonctionnement allait se répéter. Finalement, et grâce aux consignes axées entre autres sur la bonne organisation du projet, nous avons pu changer ces habitudes.

Comme au sein de nos équipes en entreprise, les tâches ont été réparties en fonction des compétences de chacun. Nous avons aussi pris en compte les envies des membres de l'équipe pour que nous puissions développer nos compétences dans certains domaines que nous avons envie de découvrir. Le projet s'est aussi bien déroulé grâce aux bonnes pratiques acquises en entreprise et à notre habitude à travailler en équipe la moitié de la semaine.

Nous avons aussi pu mettre en place des bonnes pratiques grâce aux présentations individuelles des autres étudiants. C'est grâce à une présentation du dernier cours que nous avons par exemple décidé d'utiliser SonarQube comme outil de test de qualité.

Ce projet est le premier de l'année où nous intégrons réellement la dimension de gestion de projet et d'organisation. Le module de génie logiciel nous a permis de faire le lien entre le développement technique avec lequel nous sommes familiarisés depuis le début de l'année et la gestion de projet qui est un aspect important de notre formation.

## Nos avis

### Thomas

À la fin de ce projet, je suis satisfait du rendu que nous avons à présenter. Le résultat n'est pas orienté sur le graphique, ce qui permet de se concentrer sur d'autres sujets, comme la gestion d'erreur, avec plus d'intérêt et d'apprendre davantage. De plus, faire en sorte de réaliser un projet console est un défi inhabituel, qui nous a permis de s'éloigner des sites web habituels. C'est pourquoi j'ai trouvé appréciable de pouvoir évoluer librement tout en étant conseillés durant l'intégralité du module.

Concernant le cours, il m'a permis de voir ou de revoir certaines notions primordiales pour tout bon développeur, et il est intéressant de les connaître afin de pouvoir les réappliquer plus tard, y compris pour les notions qui n'ont pas été utilisées directement pour notre projet.

Enfin, j'ai trouvé plus qu'appréciable de pouvoir avancer un projet informatique en équipe durant les heures de cours, car c'est de mon point de vue la seule manière de permettre à un projet scolaire d'être réellement réalisé en équipe. Je regrette cependant, d'un point de vue global sur notre formation actuelle, de ne pas avoir de cours totalement dédié à l'organisation d'un travail en équipe ou à la rédaction de documents techniques et fonctionnels, car je ressens un manque de connaissance sur le sujet qui s'avère pénalisant pour notre organisation, notre gestion du temps et la qualité de notre travail d'équipe.

### Simon

Cette matière ainsi que ce projet se sont trouvés plus qu'intéressant pour moi. En effet, une chose que j'ai toujours reproché, bien que discutable, en DUT Informatique à Descartes ainsi qu'en début de L3 MAGE, était le manque d'enseignement des technologies et outils actuels. Il s'avère que cette matière, au travers du projet, nous a permis d'en découvrir plusieurs : Git, Maven, Spring, SonarQube et Mockito. Elles sont toutes au coeur des équipes de développement chez Carrefour pour lequel je travaille à l'exception de SonarQube dont je n'avais jamais entendu parlé ainsi que dans d'autres entreprises après échange avec mes camarades.

Cela m'a donc été grandement utile que ce soit pour mon enrichissement personnel mais surtout pour la nouvelle équipe avec lequel je travaille chez Carrefour qui me demande des connaissances sur ces mêmes technologies.

Pour ce qui est du projet en lui même, le principe de pouvoir réaliser une petite application mais extrêmement complète était intéressante : nous avons toujours été habitué à réaliser des projets à la limite du finissable ou certaines choses sont mises de côté ou bâclées. Ici du fait de la liberté de l'application, le but n'était donc pas de réaliser l'application la plus impressionnante possible mais une application respectant des règles, étant la plus propre et respectueuse des langages de développement utilisés.

L'idée me semblait intéressante du fait que j'ai été témoin de l'agacement de devoir descendre des étages pour aller à la cellule informatique pour obtenir une salle puis les remonter ou tout simplement parcourir les couloirs des étages à la recherche d'une salle libre.

Le seul bémol que je pourrais noter serait le manque d'enseignements pour Spring et Maven qui furent compliqués à comprendre et implémenter pour nous. L'auto apprentissage via internet est certes formateur mais nous a coûté du temps.

### Marius

Le cours m'a permis de découvrir plusieurs outils que je ne connaissais pas avant. Le projet nous a permis de mettre en place les outils étudiés en amont. Le travail en groupe était aussi intéressant mais j'aurai aimé avoir plus de consignes précises pour la rédaction de la documentation.

### Ruben

Cours : Le cours est disponible en ligne H24. Les slides sont concises et claires, ce qui permet de vite trouver ce que l'on cherche si besoin. Nous avons découvert beaucoup d'outils pour améliorer la gestion de projet.

Projet : Le projet m'a permis de découvrir pas mal d'outils intéressant comme JUnit pour les test unitaires, Spring pour l'infrastructure du projet, SonarQube pour la qualité du code ou encore GitHub pour héberger son projet et garder les versions antérieures.

### Marine

J'ai beaucoup aimé cette deuxième partie du module de génie logiciel. Depuis le début de l'année, c'est le premier cours qui donne une réelle importance au rôle du chef de projet. N'étant pas issue d'un DUT informatique mais d'un DUT STID, je ne suis pas encore très à l'aise en code et j'ai été ravie de prendre le rôle de chef de projet.

Premièrement, j'ai réalisé que ce rôle ne se résume pas à dire aux membres de l'équipe « fais ça » et à faire le dossier. J'ai appris qu'il ne faut pas se lancer tête baissée dans le développement dès lors que l'on a trouvé une idée.

Ainsi, j'ai pu mettre en place plusieurs outils pour organiser le projet. L'idée de mettre en place une roadmap m'est venue de mon entreprise. En fin d'année dernière mon manager m'avait demandé de réaliser cet exercice pour rendre compte de mes réalisations et de mes objectifs. J'ai trouvé pertinent de l'adapter au projet Book'IT.

Par cet exemple, je veux expliquer que ce qui est très intéressant dans ce cours c'est que l'on peut appliquer nos bonnes pratiques d'entreprise, en plus des connaissances de cours pour réaliser un projet.

Ainsi j'ai pu découvrir ce rôle que j'ai beaucoup aimé. Même si ce n'est pas ce que je préfère faire, je pense qu'il aurait pu être intéressant au fil du cours, de nous faire coder quelques fonctions pour que les étudiants qui ont mon profil (novices en code et qui sont chef de projet pour le travail en groupe) découvrent plus cet aspect du cours.

C'est donc un bilan très positif !