



RÉPUBLIQUE DU BÉNIN
MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR
ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITÉ D'ABOMEY-CALAVI

INSTITUT DE FORMATION ET DE
RECHERCHE EN INFORMATIQUE



BP 526 Cotonou Tel : +229 21 14 19 88
<http://www.ifri-uac.net> Courriel : contact@ifri.uac.bj

MÉMOIRE

pour l'obtention du

**Diplôme de Licence Professionnelle en Systèmes d'Information et Réseaux
Informatiques**

Présenté par :

Vincent-Béni ZOSSOU

ITAJA : Système intégré de gestion d'un commerce

Sous la supervision de :

M. Fréjus LALEYE

M. Paterne GAYE

Année Académique : 2016-2017

Dédicace

- ☛ A mon père Gervais ZOSSOU;
- ☛ A ma mère Hélène KOULODJI;
- ☛ A mes frères et sœurs.

Remerciements

Comme toute œuvre humaine, un mémoire ne s'écrit pas seul. Nous avons bénéficié du concours de nombreuses personnes. Ainsi, nous adressons nos profondes gratitude à :

- Le Père Céleste pour son amour ;
- Professeur EZIN Eugène, Directeur de l'Institut de Formation et de Recherche en Informatique (IFRI) pour nous avoir suivis tout au long de notre formation ;
- Docteur EDAH Gaston, Directeur Adjoint de l'Institut de Formation et de Recherche en Informatique (IFRI) pour ses excellents conseils ;
- M. Paterné GAYE, Directeur Général de QANBIO et Maître du stage, pour la mise à notre disposition tous les moyens nécessaires et de son temps ;
- M. Fréjus LALEYE, Maître de ce mémoire, pour avoir accepté de suivre et de coordonner ce travail ;
- Tous les enseignants et le personnel administratif de l'IFRI ;
- Tous les étudiants de ma promotion ;
- Tous ceux qui d'une manière ou d'une autre m'ont aidé tout au long de la rédaction dudit document.

Table des matières

Dédicace	i
Remerciements	ii
Glossaire	1
Résumé/Abstract	2
Introduction générale	4
1 Contexte de l'étude	5
1.1 Cadre du projet	5
1.2 Présentation de la structure de formation	5
1.3 Présentation de la structure de stage	6
1.4 Présentation du projet	7
1.4.1 Problématique	7
1.4.2 Étude de l'existant	8
1.4.3 Objectifs	8
1.5 Conclusion	9
2 Analyse informatique	10
2.1 Introduction	10
2.2 Le diagramme des cas d'utilisation	10
2.3 Le diagramme d'architecture	13
2.4 Le diagramme de séquences	14
2.5 Le diagramme de classes	16
2.6 Conclusion	17
3 Conception et réalisation	18
3.1 Introduction	18
3.2 Architectures utilisées	18
3.2.1 Architecture Orientée Service	18
3.2.2 Architecture Modèle-Vue-Contrôleur	19

3.3	Technologies utilisées	19
3.3.1	Apache Tomcat	19
3.3.2	Spring Boot	20
3.3.3	Angular	20
3.3.4	Bootstrap	20
3.3.5	Groovy	21
3.3.6	TypeScript	21
3.3.7	HTML	21
3.3.8	CSS	22
3.3.9	UML	22
3.4	Environnement technique	22
3.4.1	Méthodologie : Agile - SCRUM	22
3.4.2	Outils de projet	23
3.4.3	Intégration continue : Jenkins	25
3.4.4	Gestionnaire de code source : Bitbucket avec le Git	25
3.5	Réalisation et conditions de déploiement	25
3.6	Conclusion	26
	Conclusion générale	27
	Bibliographie	28
A	Annexe 1	30

Table des figures

1.1	Logo de IFRI	5
1.2	Logo de QANBIO	6
1.3	Logo de ITAJA	7
2.1	Diagramme des cas d'utilisation	11
2.2	Diagramme de séquences	13
2.3	Diagramme de séquence 1 : un gestionnaire souhaite consulter les diagrammes des statistiques financières	14
2.4	Diagramme de séquence 2 : un gestionnaire souhaite consulter les tableaux des opérations effectuées	15
2.5	Diagramme de séquence 3 : un caissier souhaite enregistrer une opération effectuée	15

2.6	Diagramme de classes	16
3.1	Logo de Apache Tomcat	19
3.2	Logo de Spring Boot	20
3.3	Logo de Angular	20
3.4	Logo de Bootstrap	21
3.5	Logo de Groovy	21
3.6	Logo de TypeScript	21
3.7	Logo de HTML5	22
3.8	Logo de CSS3	22
3.9	Logo de UML	22
3.10	Description de Scrum	23
3.11	Logo de G Suite	24
3.12	Logo de Jira	24
3.13	Logo de Slack	24
3.14	Logo de Jenkins	25
3.15	Logo de Bitbucket	25
A.1	Diagramme de présentation des statistiques financières	30
A.2	Diagramme de présentation des indicateurs d'actions	31
A.3	Tableau des approvisionnements	31
A.4	Tableau des ventes	32
A.5	Utilisation de Typescript avec Angular	33
A.6	Utilisation de groovy avec Spring Boot	33
A.7	Utilisation de slack	34
A.8	Utilisation de Bitbucket	34

Liste des tableaux

2.1	Tableau récapitulatifs des besoins globaux	10
-----	--	----

Résumé

Ce projet de fin de formation a pour objet la conception et la réalisation de ITAJA, une solution intégrée de gestion d'un commerce. ITAJA est une solution de type Software As A Service (SAAS).

Le principal objectif du système est d'aider les managers dans leur prise de décision en leur fournissant en temps réel des indicateurs pertinents. De manière spécifique, la solution développée permet la saisie des données depuis un terminal à bas coût, le stockage des données dans une base de données sécurisée dans le cloud, l'exportation des données au format Excel pour une exploitation ultérieure et enfin la mise à disposition d'un dashboard numérique sur l'ensemble de l'activité.

Le présent rapport résume le déroulement de toutes les étapes du projet.

Mots clés : *comptabilité, commerce, stock, indicateurs, comptes, SAAS*

Abstract

This project of the end of formation has for object the conception and the realization of ITAJA, an integrated solution of management of a trade. ITAJA is a solution of type SAAS.

The main objective of the system is to help the managers in their decision-making by supplying them in real time relevant indicators. In a specific way, the developed solution allows the data capture since a low-cost terminal, a storage of the data in a database reassured in the cloud, the export of the data at the format Excel for a later exploitation and finally the provision of a digital dashboard on the whole activity.

The present report summarizes the progress of all the stages of the project.

Key words: *accouting, trade, stock, indicators, accouts, SAAS*

Introduction générale

Des ventes de mains en mains vers des ventes virtuelles, l'environnement fortement concurrentiel des entreprises n'autorise aucune erreur de gestion. La plupart des entreprises des pays en voie de développement tant du secteur public que privé font malheureusement face à des difficultés énormes de gestion tant celui des biens, des services, des employés que des boutiques.

Les solutions informatisées de gestion de stocks sont depuis des années, largement conseillées pour les sociétés qui font du commerce leur activité principale. Ces solutions représentent un dispositif global fournissant aux commerçants et propriétaires de boutiques une plateforme où se retrouvent l'inventaire des entrées de marchandises, des ventes etc..

Le suivi des flux permet aux commerçants et propriétaires de boutiques de connaître les produits les plus vendus, les boutiques qui reçoivent plus de clientèle. La courbe évolutive de leur chiffre d'affaires doit être disponible et quotidiennement mise à jour sans la moindre erreur, ce qui leur guiderait dans leurs investissements et le choix de leurs produits principaux.

Notre projet, réalisé dans le cadre du mémoire de licence professionnelle intitulé : «**ITAJA : Système intégré de gestion d'un commerce**» se veut une solution complète et efficace des problèmes cités ci-haut. Nous allons dans premier temps analyser le problème et ensuite proposer une solution pour organiser de façon optimale la gestion des stocks d'un commerce.

Contexte de l'étude

1.1 Cadre du projet

Durant le deuxième semestre de l'année préparatoire au master, nous sommes appelés à effectuer un stage de deux mois dont le fruit est ce mémoire. Le contexte de l'étude consiste à présenter le domaine d'étude et surtout un cadre de travail spécifique et bien défini. Il s'agit de se poser les questions, dans l'optique de recueillir les informations nécessaires, nous permettant de délimiter notre périmètre d'intervention.

1.2 Présentation de la structure de formation



FIGURE 1.1 – Logo de IFRI

L'**Institut de Formation et de Recherche en Informatique (IFRI)** est une école de l'Université d'Abomey-Calavi (UAC) qui a pour vocation de former des apprenants capables de devenir des acteurs pour des solutions informatiques aux différents problèmes de développement. L'institut accueille des étudiants de toute nationalité et a la vocation de devenir assez rapidement un institut sous-régional de référence. Son directeur actuel est le **Pr. Eugène EZIN**.

Les enseignements offerts par l'IFRI sont en continuelle adaptation à notre environnement social et comporte actuellement la licence en Génie Logiciel et en Sécurité Informatique, la classe préparatoire au master et le master de recherche.

Son slogan étant **IFRI, nous bâtissons l'excellence**, il forme des techniciens en Génie Logiciel, en Sécurité Informatique et des ingénieurs en Informatique Générale capables de :

- créer leurs propres entreprises et de les gérer correctement en offrant des services de qualité ;
- s'adapter aux nouvelles technologies informatiques arrivant sur le marché ;
- faire l'analyse, la modélisation, la conception et le développement de divers types de systèmes informatiques ;
- travailler dans des entreprises publiques ou privées.

1.3 Présentation de la structure de stage



FIGURE 1.2 – Logo de QANBIO

QANBIO est un cabinet de conseil situé à Fidjrossè et spécialisé dans les systèmes d'information et les nouveaux modes d'organisation du travail, notamment les méthodes agiles. L'ambition de QANBIO est d'aider ses partenaires à bâtir un système d'informations performant, résilient et surtout évolutif. L'offre du cabinet couvre plusieurs thématiques en rapport avec les systèmes d'information :

- Conseil et assistance à la maîtrise d'ouvrage ;
- Formation sur mesure pour vos équipes ;
- Sous-traitance ;
- Réalisation de tâches techniques, notamment le développement de logiciel ou la configuration d'un réseau informatique.

QANBIO propose également à sa clientèle plusieurs formes de partenariats, notamment :

- L'infogérance complète ou partielle ;
- Les prestations en régie ;
- Les prestations en forfait.

1.4 Présentation du projet

1.4.1 Problématique

Les Petite et Moyenne Entreprise (PME) et Très Petite Entreprise (TPE) constituent l'essentiel du tissu économique du Bénin et de la sous-région. Compte tenu de leurs organisations caractérisées par des ressources limitées, en particulier en personnel, ces entreprises ont tout à gagner dans une dématérialisation de leurs processus métiers. Cependant, force est de constater que ce n'est pas le cas. Ceci s'explique de manière empirique par les coûts énormes engendrés par l'informatisation.

En effet, pour prendre le cas d'un processus classique qui est la comptabilité dans une PME/TPE, les investissements s'élèvent à plusieurs centaines de milliers de francs incluant d'une part l'acquisition d'un ordinateur et d'une licence logicielle. À ces coûts s'ajoutent le recrutement et la formation d'un personnel qualifié. Enfin, compte tenu de l'environnement béninois et ouest-africain marqué par des variations fluctuantes des caractéristiques nominales de l'électricité et des coupures régulières, il faut rajouter les coûts d'acquisition d'un groupe électrogène et d'un régulateur électrique.

Tout ceci constitue autant de barrières aux projets de transformation numérique dans les PME et TPE au Bénin et dans la sous-région. C'est pour répondre à ce besoin que QANBIO a initié le développement de ITAJA une solution SAAS de **gestion d'un commerce**.



FIGURE 1.3 – Logo de ITAJA

1.4.2 Étude de l'existant

ITAJA est une solution composite de type SAAS qui intégrait déjà avant notre arrivée dans le cabinet plusieurs applications, notamment :

- . une application **frontend** Android déployée sur PlayStore ;
- . une application **backend** Java EE déployée sur un serveur Apache Tomcat.

L'application backend est un pool de web-services permettant d'interagir avec la base de données de manière sécurisée ; tandis que l'application Android est utilisée dans les points de ventes afin de permettre aux collaborateurs d'enregistrer en temps réels les opérations effectuées (ventes, approvisionnements, etc).

☛ Aspects positifs

La plateforme était opérationnelle et en phase de test grandeur nature auprès d'une entreprise cotoise.

☛ Limites

Il manquait une application front office aux managers ou gestionnaires des boutiques. Après l'enregistrement des opérations grâce à l'application Android, il faudrait un outil permettant d'exploiter aisément les données en termes de stocks et de statistiques financières des points de ventes et magasins.

1.4.3 Objectifs

L'objectif global visé par ce travail est de réaliser une application complète Web fonctionnelle qui puisse permettre aux propriétaires de magasins de pouvoir suivre l'évolution de leurs stocks d'une part, et d'autre part l'évolution des statistiques financières (chiffre d'affaires, ventes ...).

Afin de mieux structurer le travail et de faciliter les choses, nous avons divisé le projet en des objectifs spécifiques à savoir :

- proposer une plateforme de suivi en temps réel des entrées et sorties (approvisionnements et ventes) ;
- proposer une plateforme de suivi en temps réel du personnel, des approvisionnements et des inventaires ;
- proposer des diagrammes d'évolution du chiffre d'affaires et des indicateurs d'actions afin de guider dans la prise de décisions et l'orientation des domaines d'investissements ;
- proposer des diagrammes des meilleurs zones de ventes et des produits les plus vendus afin de cibler les zones, produits et populations favorables.

1.5 Conclusion

Au terme de ce chapitre, il convient de préciser que la population directement visée est les commerçants en général et les propriétaires de magasins. Et pour gérer efficacement leur commerce et accroître leur chiffre d'affaires, nous proposons une application Web qui aura pour objectif répondre aux différents besoins dudit service, et ce à un coût vraiment réduit. Dans le chapitre suivant nous parlerons de l'analyse informatique du problème.

Analyse informatique

2.1 Introduction

La phase d’analyse d’un projet informatique consiste à déterminer la description du besoin pour élaborer la solution technique. L’analyse se positionne comme une interface entre les utilisateurs demandeurs de services et des informaticiens au vocabulaire spécifié. À la fin de ce chapitre nous devrons être capable de répondre à cette question : **Que fait le système** ? [1]

2.2 Le diagramme des cas d’utilisation

Un diagramme des cas d’utilisation représente comme son nom l’indique les cas d’utilisation, c’est-à-dire les fonctionnalités ou lots d’actions que devront réaliser nos acteurs (utilisateurs, gestionnaires). Le diagramme des cas d’utilisation met donc en évidence de quelle façon les acteurs utiliseront le logiciel : **qui** doit pouvoir faire **quoi** ? [2]

Voici un petit tableau qui récapitule les différents besoins des acteurs :

Acteurs	Rôles
Utilisateur	Ajouter une vente et une dépense
Gestionnaire	<ul style="list-style-type: none"> • Visualiser les diagrammes des statistiques financières ; • Visualiser les tableaux des opérations ; • Manipuler les collaborateurs et les points de ventes

TABLE 2.1 – Tableau récapitulatifs des besoins globaux

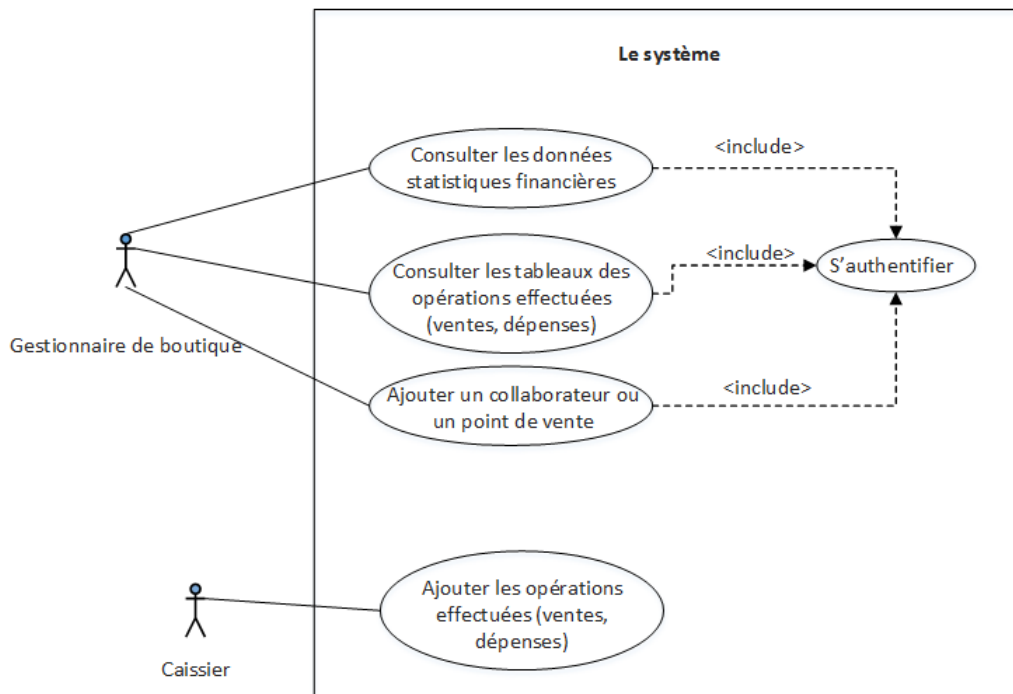


FIGURE 2.1 – Diagramme des cas d'utilisation

Description textuelle des cas d'utilisation :

Nous avons décidé de définir ici trois cas d'utilisation et le fonctionnement du système.

- Cas numéro 1 :

Nom : visualiser les diagrammes des statistiques financières

Description : la visualisation les diagrammes de statistiques financières doit être disponible pour tous les utilisateurs.

Pré-condition : être utilisateur de Itaja, être authentifié et disposer des droits d'un gestionnaire

Démarrage : le gestionnaire a demandé la page de visualisation des diagrammes de statistiques

Scénario principal :

- L'utilisateur arrive sur la page d'accueil.
- L'utilisateur clique sur le menu 'tableau de bord'.
- Le système demande d'entrer la période dont vous souhaitez visualiser les données.
- L'utilisateur entre la période.
- Le système affiche les statistiques de la période concernée.

Scénario d'erreur : la période saisie par l'utilisateur n'existe pas dans les données disponibles

Le système affiche un message d'erreur.

- Cas numéro 2 :

Nom : visualiser les tableaux des opérations effectuées (ventes, dépenses) **Description :** un gestionnaire est appelé à suivre les opérations effectuées dans son (ou ses) point(s) de ventes

Pré-condition : être utilisateur de Itaja, être authentifié et disposer des droits d'un gestionnaire

Démarrage : le gestionnaire a demandé la page de visualisation du tableau d'une opération

Scénario principal :

- (a) L'utilisateur arrive sur la page d'accueil.
- (b) L'utilisateur clique sur le menu correspondant à l'opération dont il souhaite visualiser le tableau.
- (c) Le système affiche le tableau demandé.

- Cas numéro 3 :

Nom : Ajout d'une opération

Description : un caissier est appelé à enregistrer les opérations effectuées dans son point de ventes

Pré-condition : être utilisateur de Itaja

Démarrage : l'utilisateur a demandé la page d'enregistrement d'une opération

Scénario principal :

- (a) L'utilisateur ouvre son application Android.
- (b) L'utilisateur clique sur le menu correspondant à l'opération qu'il veut enregistrer.
- (c) Le système affiche le formulaire correspondant à cette opération.
- (d) L'utilisateur renseigne tous les champs demandés.
- (e) L'utilisateur appuie sur le bouton Valider.
- (f) Le système enregistre l'opération et affiche un message de succès.
- (g) Le système retourne au menu d'accueil.

Scénario d'erreur : l'utilisateur n'a pas saisi tous les champs du formulaire

Le système affiche un message d'erreur et le renvoie vers le formulaire concerné.

- Cas numéro 4 :

Nom : Fonctionnement global du système

Démarrage : Un utilisateur fait une requête quelconque

Scénario principal :

- (a) L'utilisateur arrive sur la page d'accueil.

- (b) L'utilisateur fait une requête.
- (c) La requête est transmise à un serveur Linux, comportant Apache et sur lequel est embarquée la WebApp.
- (d) Ce serveur traite la requête, et la retransmet à un second serveur Linux comportant également Apache et MySQL et sur lequel sont embarqués les WebServices.
- (e) La requête est traitée par le WebService correspondant, questionne la base de données s'il y a lieu et formule la réponse.
- (f) La réponse est transmise au premier serveur, qui la retransmet à l'utilisateur.
- (g) L'utilisateur reçoit la réponse.

2.3 Le diagramme d'architecture

Le diagramme d'architecture nous permettra de décrire comment les éléments du système interagissent entre eux depuis le moment où un utilisateur fait une requête jusqu'à l'obtention de sa réponse.

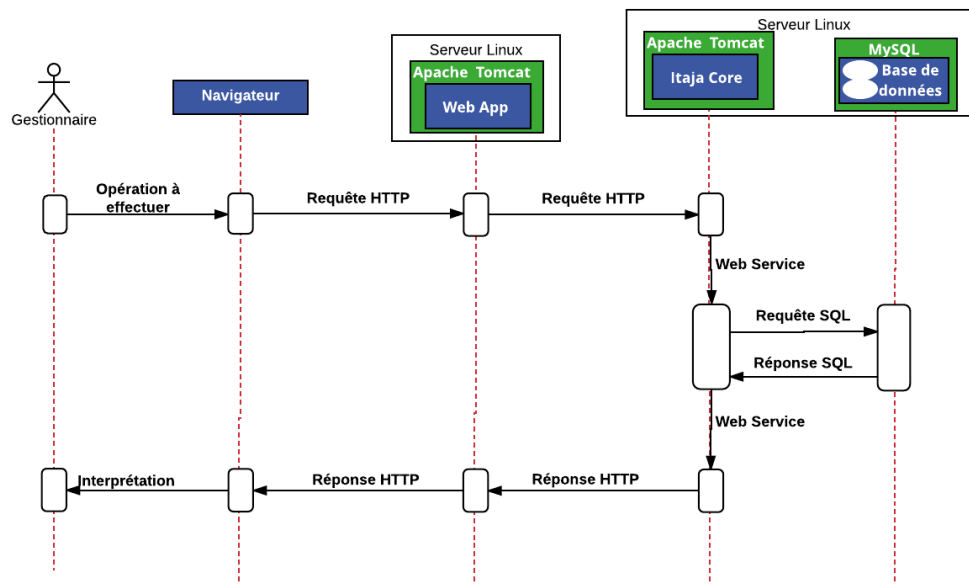


FIGURE 2.2 – Diagramme de séquences

2.4 Le diagramme de séquences

Le diagramme de séquence nous permettra de décrire comment les éléments du système interagissent entre eux et avec les acteurs :

- Les objets au cœur d'un système interagissent en s'échangeant des messages ;
- Les acteurs interagissent avec le système au moyen d'Interface Homme-Machine (IHM).

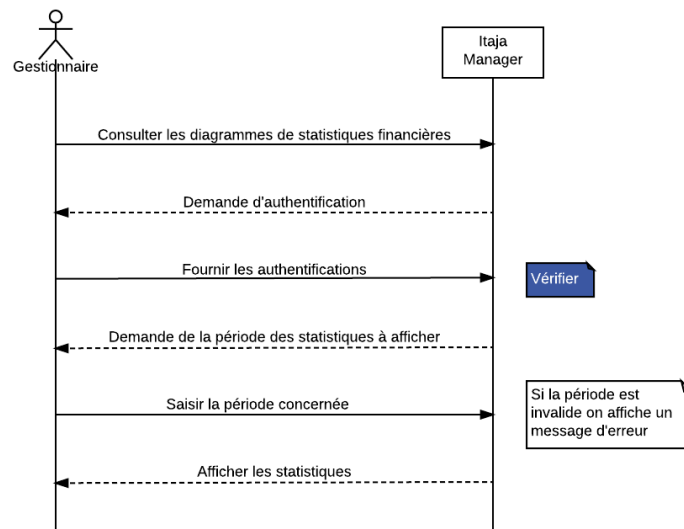


FIGURE 2.3 – Diagramme de séquence 1 : un gestionnaire souhaite consulter les diagrammes des statistiques financières

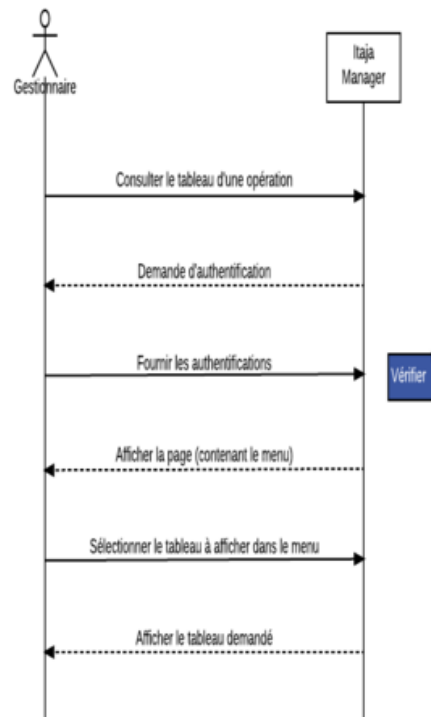


FIGURE 2.4 – Diagramme de séquence 2 : un gestionnaire souhaite consulter les tableaux des opérations effectuées

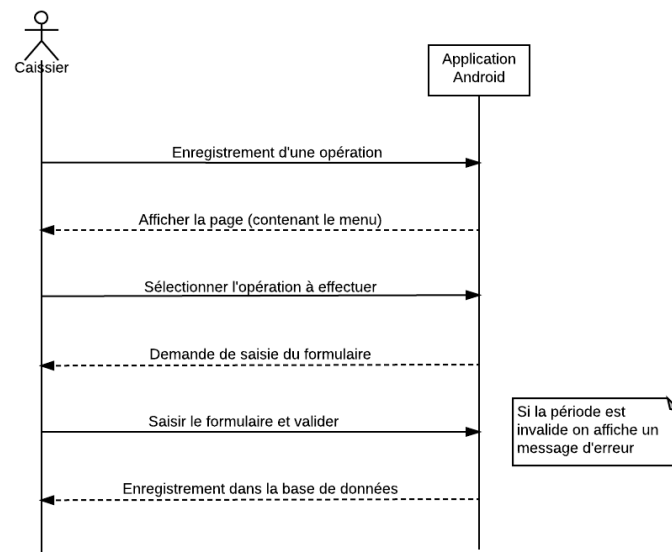


FIGURE 2.5 – Diagramme de séquence 3 : un caissier souhaite enregistrer une opération effectuée

2.5 Le diagramme de classes

Alors que le diagramme des cas d'utilisation nous montre le système du point de vue des acteurs, le diagramme de classes en montre la structure interne. Il permet de fournir une représentation abstraite des objets du système qui vont interagir pour réaliser les cas d'utilisation.

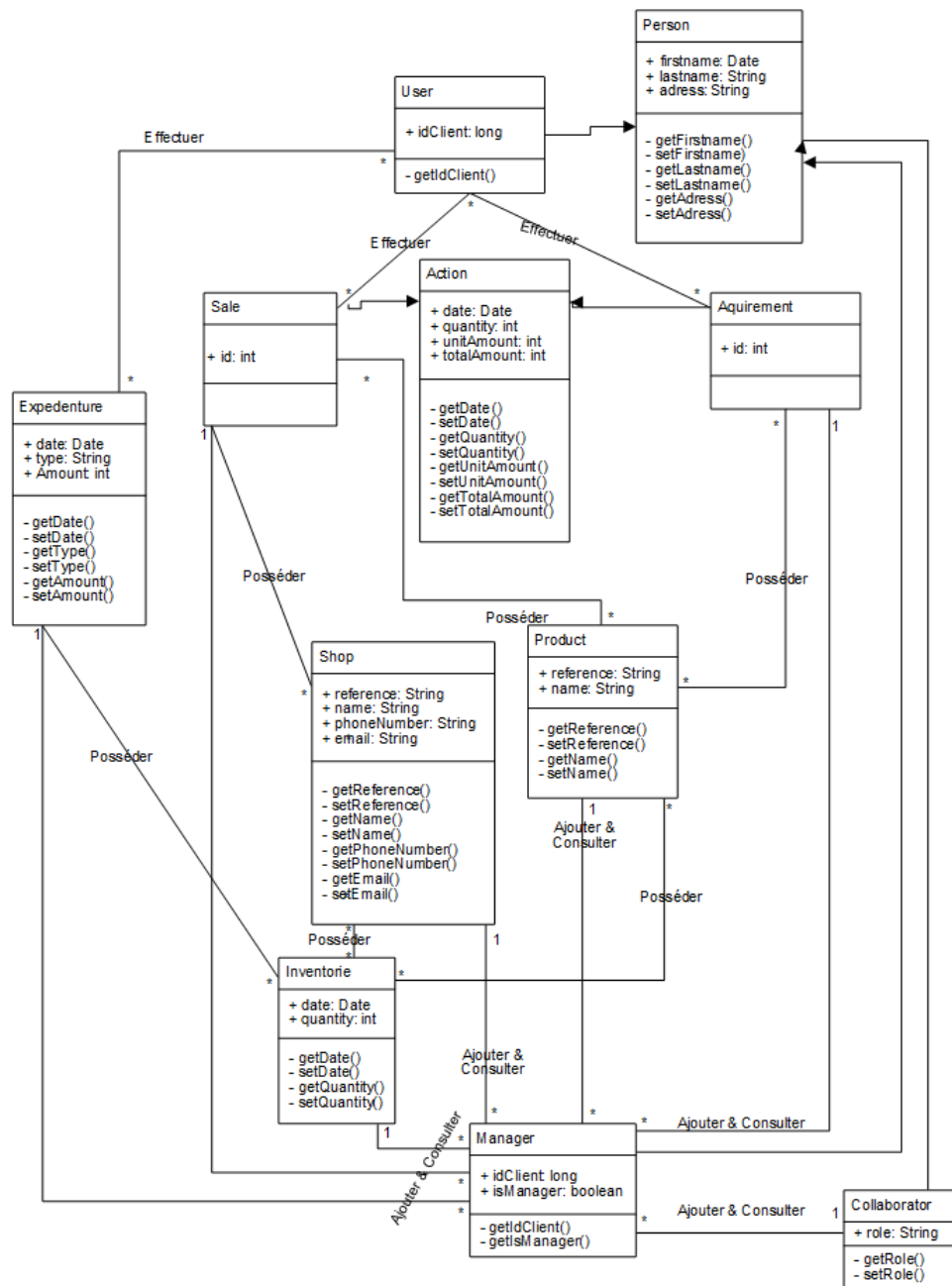


FIGURE 2.6 – Diagramme de classes

2.6 Conclusion

Ce chapitre nous a permis de retenir que l'analyse du système fait partie d'une chaîne de travail à réaliser dans la mise en place de l'application. Elle nous a également permis de déterminer à quoi servira l'application une fois déployée. Dans le prochain chapitre nous parlerons de la conception de l'application, une phase qui suit directement la phase d'analyse.

Conception et réalisation

3.1 Introduction

La conception de logiciel met en œuvre un ensemble d'activités qui, à partir d'une demande d'informatisation d'un processus permettent la conception, l'écriture et la mise au point d'un logiciel jusqu'à sa livraison au demandeur. Ce chapitre nous permettra de décrire **comment** nous allons réaliser l'application et avec quelles **technologies**? [3]

3.2 Architectures utilisées

3.2.1 Architecture Orientée Service

Service Oriented Architecture, en français **Architecture Orientée Service (SOA)** est un modèle d'interaction applicative qui met en œuvre des services (composants logiciels) avec une forte cohérence interne (par l'utilisation d'un format d'échange pivot, Extensible Markup Language (XML) ou JavaScript Object Notation (JSON)) et avec des couplages externes (par l'utilisation d'une couche d'interface interopérable, le plus souvent un service web).

Une architecture orientée service se conforme à divers principes de gestion des services influençant directement le comportement intrinsèque d'une solution logicielle et le style de sa conception :[5]

- L'encapsulation des services.
- La réutilisation des services partageant la logique entre plusieurs services avec l'intention de promouvoir la réutilisation.
- La composition des services.

- L'autonomie des services.
- L'optimisation des services.
- La découverte des services depuis leur description extérieure.

3.2.2 Architecture Modèle-Vue-Contrôleur

Le patron d'architecture logicielle model-view-controller (MVC) (en français modèle-vue-controlleur) est un modèle destiné à répondre aux besoins des applications interactives en séparant les problématiques liées aux différents composants au sein de leur architecture respective. Angular et Spring Boot intègre parfaitement l'architecture MVC.[17] Ce paradigme regroupe les fonctions nécessaires en trois catégories :

- un modèle (modèle de données);
- une vue (présentation, interface utilisateur);
- un contrôleur (logique de contrôle, gestion des événements, synchronisation).

3.3 Technologies utilisées

Pour arriver à réaliser la plate-forme, nous avons fait appel à plusieurs technologies selon les besoins. Après avoir spécifié le serveur Web que nous allons utiliser, nous détaillerons les frameworks avant de spécifier les langages informatiques qui vont avec ces derniers. Nous expliquerons aussi chaque choix.

3.3.1 Apache Tomcat

Tomcat est un serveur HyperText Transfer Protocol (HTTP) qui gère les servlets et les JavaServer Pages (JSP).[4]

Il nous a été d'une double utilité pour notre projet :

- c'est l'environnement où a été déployé **Itaja Core**, la couche de l'application qui contient les WebServices chargés de communiquer avec la base de données;
- c'est également l'environnement où a été déployé la version compilée de l'application permettant l'accès depuis n'importe quel navigateur web



FIGURE 3.1 – Logo de Apache Tomcat

3.3.2 Spring Boot

Spring est un framework libre pour construire et définir l'infrastructure d'une application java, dont il facilite le développement et les tests..[14]

Il nous a facilité le développement le 'Itaja-Core', la partie applicative de Itaja qui constitue le pool de WebServices chargés de communiquer avec la base de données et de répondre aux différentes requêtes des utilisateurs.



FIGURE 3.2 – Logo de Spring Boot

3.3.3 Angular

Angular est un framework libre écrit en JavaScript (JS) par Google qui permet d'améliorer la syntaxe de JS. Il étend le Hypertext Markup Language (HTML) pour le rendre dynamique, et permet de développer ses propres balises et attributs HTML. Il permet d'apporter un aspect applicatif au front-office, et ne se limite plus au simple fait ajouter de animations au Document Object Model (DOM).[6]

Nous l'avons utilisé pour réaliser la partie front, c'est-à-dire la vue ou l'interface utilisateur. Il était particulièrement intéressant car il permet au navigateur, une fois les données chargées, de ne plus avoir à se connecter au serveur avant la navigation entre les différentes pages de l'application. Cela réduit la consommation des données et permet aux utilisateurs de réaliser des économies.



FIGURE 3.3 – Logo de Angular

3.3.4 Bootstrap

Bootstrap est une collection d'outils utile à la création du design (graphisme, animation et interactions avec la page dans le navigateur etc...) de sites et d'applications web. C'est un ensemble qui contient des codes HTML et CSS, des formulaires, boutons, outils de navigation et autres éléments interactifs, ainsi que des extensions JavaScript en option.[18]

Il a le mérite dans notre projet de donner de la beauté et du goût nos pages Webs.



FIGURE 3.4 – Logo de Bootstrap

3.3.5 Groovy

Groovy Groovy est un langage de programmation orienté objet destiné à la plate-forme Java et lui constitue également une alternative. Il peut utiliser les bibliothèques Java et être utilisé dans des classes Java.[15]

Il est utilisé parce qu’il est typé et orienté objet tout comme le Java ; ce qui s’intègre dans un style général de programmation. Il est compris par Spring Boot et nous a permis de réaliser les WebServices.



FIGURE 3.5 – Logo de Groovy

3.3.6 TypeScript

TypeScript est un langage de programmation libre et open source développé par Microsoft qui a pour but d’améliorer et de sécuriser la production de code JavaScript. Le code TypeScript est transcompilé en JavaScript, pouvant ainsi être interprété par n’importe quel navigateur web ou moteur JavaScript.[7]

Il est utilisé parce qu’il est typé et orienté objet tout comme le Java ; ce qui s’intègre dans un style général de programmation.



FIGURE 3.6 – Logo de TypeScript

3.3.7 HTML

HTML est le format de données conçu pour représenter les pages Web. C’est un langage de balisage permettant d’écrire l’hypertexte, d’où son nom. Il permet de structurer sémantiquement et de mettre en forme le contenu des pages, d’inclure des ressources multimédias dont les images, des formulaires de saisie et des programmes informatiques.

Le couple HTML et Cascading Style Sheets (CSS) est indispensable pour tout développement Web.[16]



FIGURE 3.7 – Logo de HTML5

3.3.8 CSS

CSS est un langage qui permet de gérer la présentation d'une page Web. Les styles permettent de définir des règles appliquées à un ou plusieurs documents HTML. Ces règles portent sur le positionnement des éléments, l'alignement, les polices de caractères, les couleurs, les marges et espacements, les bordures, les images de fond, etc...[8]



FIGURE 3.8 – Logo de CSS3

3.3.9 UML

Le Langage de Modélisation Unifiée, de l'anglais **UML**, est un langage de modélisation graphique à base de pictogrammes conçu pour fournir une méthode normalisée pour visualiser la conception d'un système. Il est couramment utilisé dans le développement logiciel et en conception orientée objet.[2]

Ce langage nous a permis de construire les différents diagrammes pour la modélisation de la plate-forme.



FIGURE 3.9 – Logo de UML

3.4 Environnement technique

3.4.1 Méthodologie : Agile - SCRUM

Agile est une méthode de développement informatique permettant de concevoir des logiciels en impliquant au maximum le client, ce qui permet une grande réactivité à ses demandes. Elles visent la satisfaction réelle du besoin du client, et non d'un contrat établi préalablement.

Scrum est un schéma d'organisation qui s'axe autour de petits sprints durant lesquels l'équipe arrive à développer des produits complexes tout en enchaînant les tâches quotidiennes. Pour accompagner les sprints, l'équipe se doit de suivre quelques rituels à savoir des réunions de planification de projet, des rétrospectives de ce qui a été fait ou non, etc).[9]

Dans le cadre de notre projet, nous avons fait des sprints de une semaine chacun et organisons les réunions chaque matin à une heure convenue (9h).

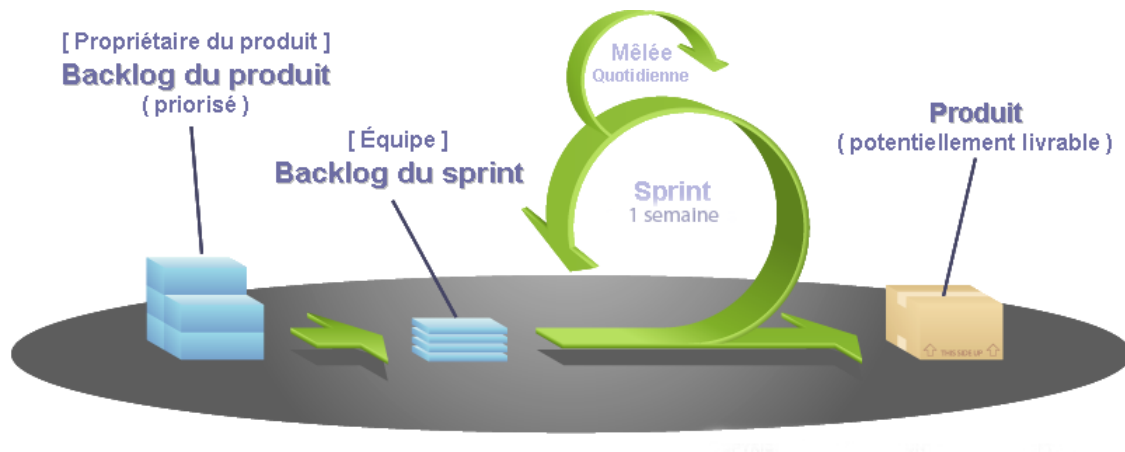


FIGURE 3.10 – Description de Scrum

3.4.2 Outils de projet

☛ G Suite

G Suite est une suite d'outils et de logiciels de productivité destinée aux professionnels, proposée par Google sous la forme d'un abonnement.

La suite inclut les applications Web de Google les plus courantes, comme **Gmail**, **Google Hangouts**, **Google Agenda**, **Google+**, **Google Drive**, **Google Docs**, **Sheets**, **Slides** et **Forms** et **Google Sites**. Elle inclut aussi des fonctionnalités répondant aux besoins des entreprises, comme les adresses de courrier électronique personnalisées, un espace de stockage de 30 Go pour les documents et les e-mails, ainsi qu'une assistance 24h/24, 7j/7 par téléphone et par messagerie électronique.[10]

Il nous permettait de réaliser les échanges et la sauvegarde des données au sein de l'entreprise.

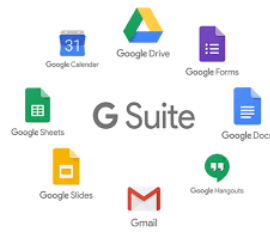


FIGURE 3.11 – Logo de G Suite

☛ Jira

Jira est un système de suivi de bugs, de gestion des incidents, et de gestion de projets développé par Atlassian.[11]

Les fonctions qu'offrent Jira et pour lesquelles nous l'avons utilisé sont :

- **Planification** : Jira nous permet de créer des user stories et des tickets, de planifier des sprints et d'affecter les tâches à l'ensemble de l'équipe de développement.
- **Suivi** : Jira nous permet de définir des priorités et de bénéficier d'une visibilité totale pour discuter du travail de l'équipe dans son contexte.
- **Génération des rapports** : Jira améliore les performances de l'équipe en exploitant en temps réel les données visuelles des rapports.



FIGURE 3.12 – Logo de Jira

☛ Slack

Slack est une plateforme qui centralise les flux de communication. C'est un espace de travail numérique qui s'est placé au cœur de l'organisation et qui, en interconnectant toutes ses ressources (humaines ou techniques), nous permet de faire des gains de productivité. Une équipe Slack se compose du propriétaire de l'équipe, des administrateurs, des collaborateurs et des invités[12]



FIGURE 3.13 – Logo de Slack

3.4.3 Intégration continue : Jenkins

Jenkins est un outil open source d'intégration continue, écrit en Java, et qui fonctionne dans un conteneur de servlets tel qu'Apache Tomcat, ou en mode autonome avec son propre serveur Web embarqué. Les fonctions qu'offrent Jenkins et pour lesquelles nous l'avons utilisé sont :[13]

- **Centralisation** : Jenkins permet à l'équipe de bénéficier d'informations communes.
- **Automatisation** : Jenkins permet d'effectuer l'ensemble des phases de productions (compilation, tests, déploiement...) sans ou avec très peu d'intervention humaine.
- **Historisation** : Jenkins permet de garder les productions précédentes et de voir l'évolution des productions.



FIGURE 3.14 – Logo de Jenkins

3.4.4 Gestionnaire de code source : Bitbucket avec le Git

Bitbucket est un outil qui nous a permis de stocker l'ensemble des fichiers de notre projet tout en conservant la chronologie de toutes les modifications qui ont été effectuées dessus. Il permet notamment de retrouver les différentes versions d'un lot de fichiers connexes et de servir de repo source à l'intégration continue par Jenkins.



FIGURE 3.15 – Logo de Bitbucket

3.5 Réalisation et conditions de déploiement

Nous avons pris une large gamme des captures d'écran lors de la réalisation de l'application. Ces images vous seront présentées dans l'annexe.

Pour que l'application puisse être utilisée par les masses, il faut la déployer en production. Pour ce faire, il nous fallait un ordinateur accessible par des clients (un serveur) équipé de :

- Un système d'exploitation UNIX

-
- Un serveur Web Apache Tomcat,
 - Mysql Server

3.6 Conclusion

Tout au long de ce chapitre, nous avons d'abord parcouru les technologies qui nous ont permis de réaliser notre application. Ensuite nous avons pu constater que pour déployer notre application en production, il nous faut satisfaire certaines exigences techniques tant en matériels qu'en logiciels.

Conclusion générale

Le présent travail de fin de formation intitulé «ITAJA : Système intégré de gestion d'un commerce» nous a permis de proposer aux commerçants et propriétaires de magasins une solution pour gérer leurs stocks(inventaires), les opérations effectuées (ventes, dépenses), et surtout l'évolution des statistiques financières (chiffre d'affaires, charges), et ce à moindre coût. Eu regard à notre sujet, nous nous sommes posé quelques questions à savoir :

- Comment proposer une solution opérationnelle pour les propriétaires de magasins
- Quelles technologies choisir pour résoudre le problème.

Sur ce, la mise en oeuvre d'une application Web pour la gestion des magasins a été le choix déterminant pour résoudre tous les problèmes ci-haut. Au cours de ce travail nous avons présenté les différentes étapes de la conception ainsi que la réalisation de l'application. S'il faut préciser quelque chose, ce projet nous a permis d'approfondir nos connaissances dans le domaine du développement Web.

Aux lecteurs de ce document, nous rappelons que cette application est loin d'avoir la prétention d'être parfaite. Elle n'est qu'une proposition de solutions aux problèmes de gestion de stocks et de magasins pour les commerçants et propriétaires de magasins. Nous restons ouvert aux critiques et suggestions.

Bibliographie

- [1] <http://www.ib-formation.fr/catalogue/nbs-details/catref/universib-gouvernance-informatique-gestion-de-projets-et-methodes-conduite-de-projets/ref/mg722/realiser-lanalyse-fonctionnelle-et-technique-dun-projet-informatique> *du 17 juillet 2017*
- [2] <https://openclassrooms.com/courses/debutez-l-analyse-logicielle-avec-uml> OpenClassrooms, *du 27 juin 2017*
- [3] https://fr.wikipedia.org/wiki/Conception_de_logiciel Wikipedia, *du 18 juillet 2017*
- [4] https://fr.wikipedia.org/wiki/Apache_Tomcat Wikipedia, *du 25 juillet 2017*
- [5] https://fr.wikipedia.org/wiki/Architecture_orient%C3%A9e_services Wikipedia, *du 27 juillet 2017*
- [6] https://www.synbioz.com/blog/introduction_a_angularjs Angular, *du 18 juillet 2017*
- [7] <https://fr.wikipedia.org/wiki/TypeScript> Wikipedia, *du 18 juillet 2017*
- [8] https://fr.wikipedia.org/wiki/Feuilles_de_style_en_cascade Wikipedia, *du 3 août 2017*
- [9] <http://www.groupeafg.com/methodes-agiles-scrum/> Agile-Scrum, *du 3 août 2017*
- [10] https://fr.wikipedia.org/wiki/G_Suite Wikipedia, *du 3 août 2017*
- [11] <https://fr.wikipedia.org/wiki/Jira> Wikipedia, *du 21 août 2017*
- [12] <https://get.slack.help/hc/fr-fr/articles/115004071768-Qu-est-ce-que-Slack-> Slack, *du 23 août 2017*

-
- [13] http://igm.univ-mlv.fr/~dr/XPOSE2010/Lecharpentier_Jenkins/purpose.html Jenkins, *du 3 août 2017*
- [14] [https://fr.wikipedia.org/wiki/Spring_\(framework\)](https://fr.wikipedia.org/wiki/Spring_(framework)) Wikipedia, *du 24 août 2017*
- [15] [https://fr.wikipedia.org/wiki/Groovy_\(langage\)](https://fr.wikipedia.org/wiki/Groovy_(langage)) Wikipedia, *du 24 août 2017*
- [16] https://fr.wikipedia.org/wiki/Hypertext_Markup_Language Wikipedia, *du 14 juillet 2017*
- [17] <https://fr.wikipedia.org/wiki/Mod%C3%A8le-vue-contr%C3%B4leur> Wikipedia, *du 14 juillet 2017*
- [18] [https://fr.wikipedia.org/wiki/Bootstrap_\(framework\)](https://fr.wikipedia.org/wiki/Bootstrap_(framework)) Wikipedia, *du 14 juillet 2017*

Annexe 1



FIGURE A.1 – Diagramme de présentation des statistiques financières



FIGURE A.2 – Diagramme de présentation des indicateurs d’actions



FIGURE A.3 – Tableau des approvisionnements

[Tableau de bord](#)
[Ventes](#)
[+ Approvisionnements](#)
[Autres dépenses](#)
[Inventaires](#)
[Données de base](#)
[Paramètres](#)
[Imprimer un PDF](#)
[Paramètres](#)
[Imprimer un PDF](#)
[+ Approvisionnements](#)
[Autres dépenses](#)
[Inventaires](#)
[Données de base](#)
[Paramètres](#)

Ventes

> Opérations / Ventes

Recherche

Date de début

Date de fin

Produit

Boutique

RECHERCHER

Montant total des ventes : 6 422 512 XOF

Nombre total des ventes : 20

Date ↕	Référence ↕	Nom ↕	Point de vente ↕	Quantité ↕	PU (XOF) ↕	Total (XOF) ↕
02-05-17 13:30:30	MCH-01-001	Daniella	vedoko	10	10	1 000
02-05-17 13:30:30	MCH-01-003	Appeal	vedoko	10		1 000
02-05-17 16:17:04	LAC-01-002	Ultra set 450ml	vedoko	60	3 600	216 000
04-05-17 10:50:02	MCH-01-003	Appeal	vedoko	258	369	95 202
08-05-17 13:19:44	FCL-01-002	en bloc 300F	vedoko	14	10 000	140 000
08-05-17 14:20:07	MCH-01-002	4 Flowers	vedoko	14	5 000	70 000
08-05-17 15:52:38	MCH-01-002	4 Flowers	vedoko	588	15	8 820
27-05-17 21:39:28	PDR-01-002		vedoko	15	933	14 000
30-05-17 15:06:05	RAL-01-002	1000F	vedoko	14	40	560
30-05-17 18:10:47	EYL-01-002	1000F	vedoko	47	50	2 350

12

TÉLÉCHARGER

Tous droits réservés ITAJA Solution © QANBIO 2017

FIGURE A.4 – Tableau des ventes

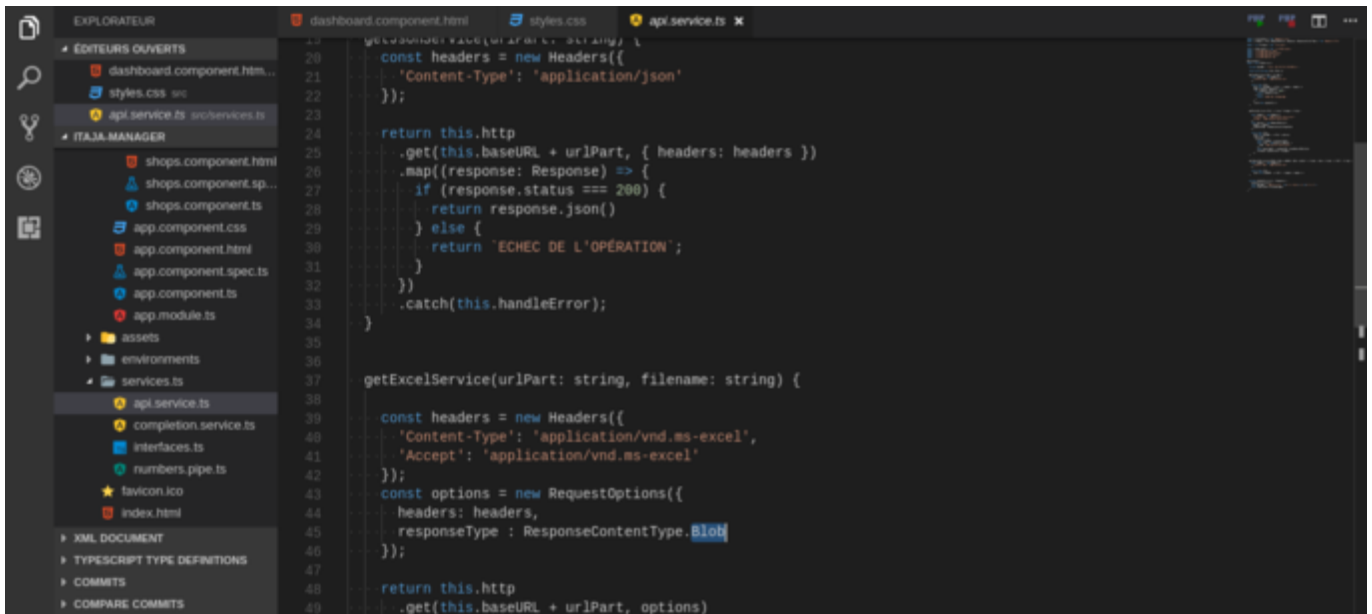


FIGURE A.5 – Utilisation de Typescript avec Angular

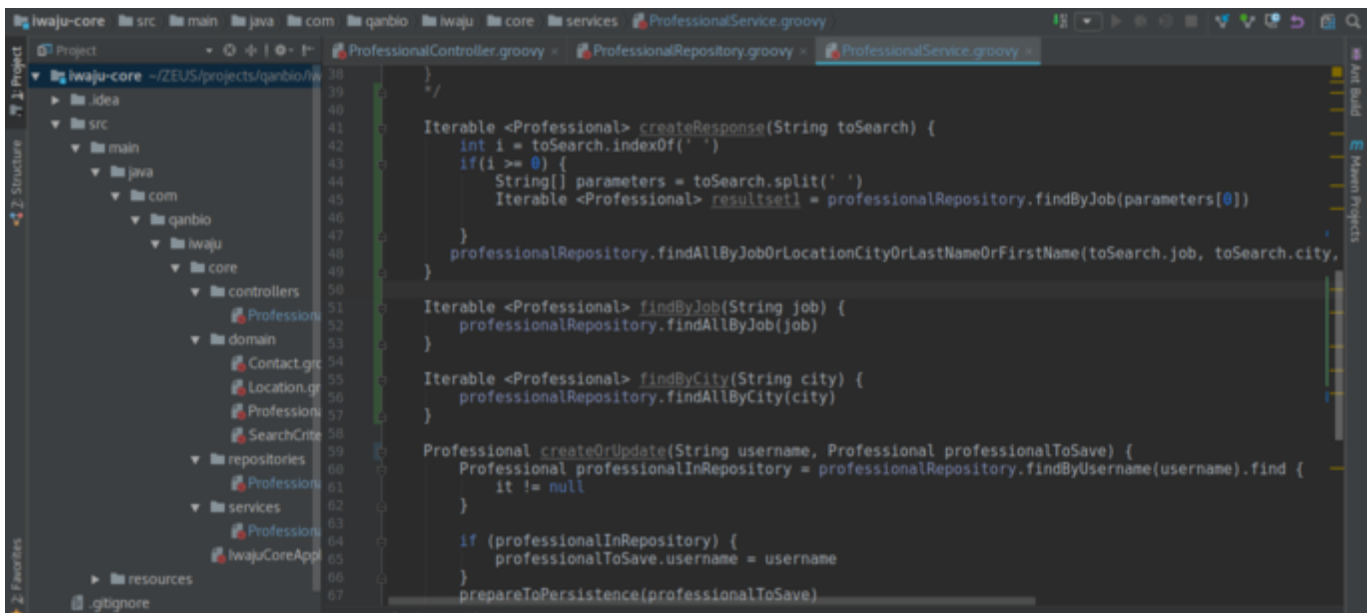


FIGURE A.6 – Utilisation de groovy avec Spring Boot

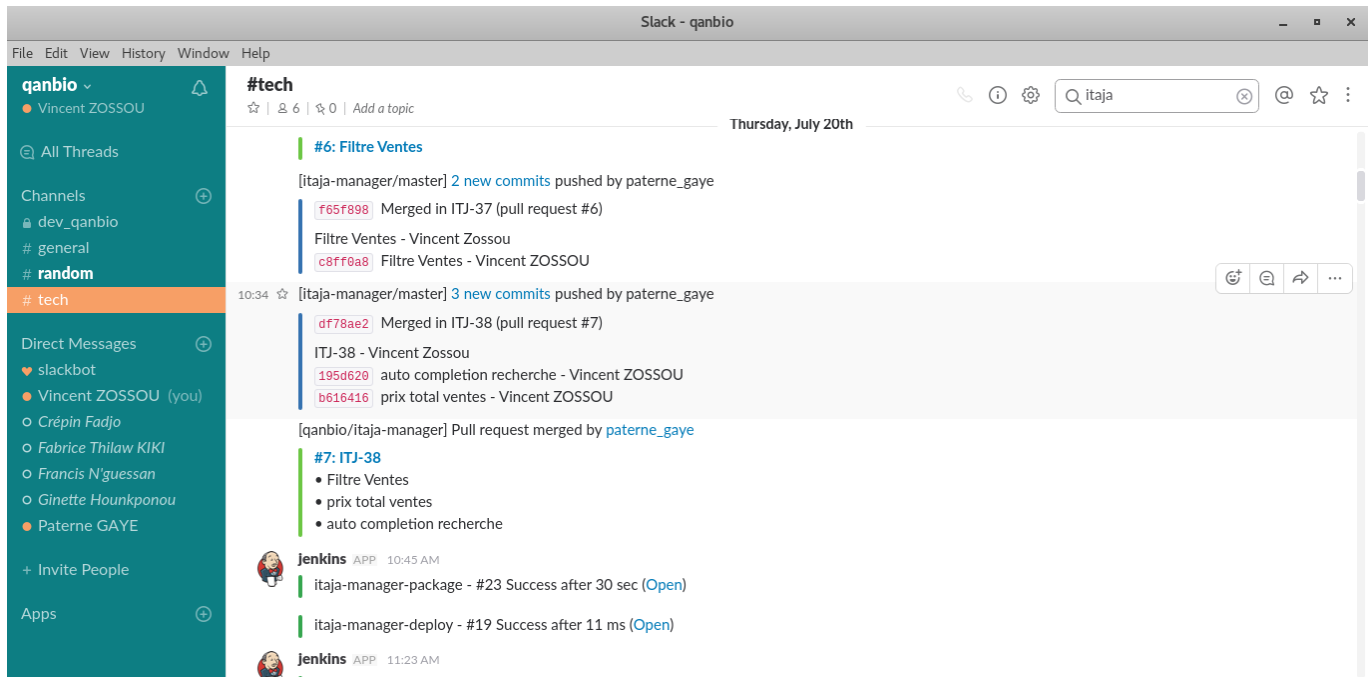


FIGURE A.7 – Utilisation de slack

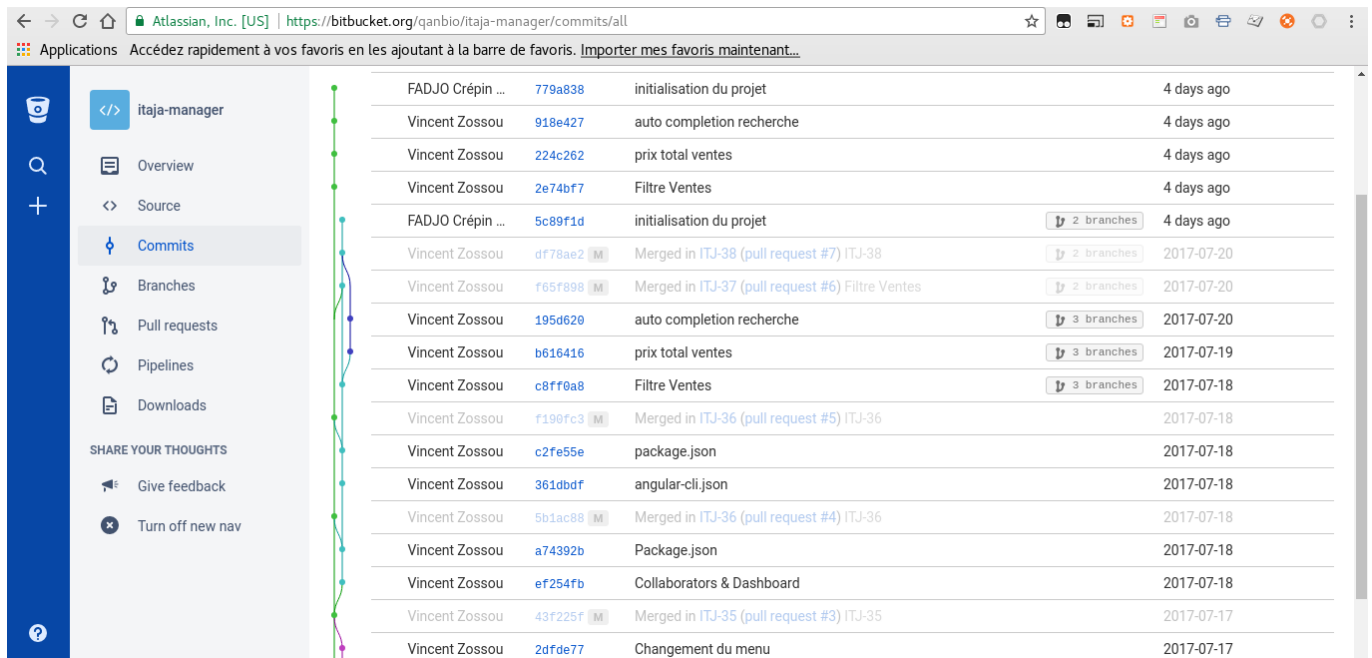


FIGURE A.8 – Utilisation de Bitbucket
