



Ministère de l'Enseignement Supérieur
Et de la Recherche Scientifique
Université de Tunis El Manar
Faculté des Sciences Economiques et de Gestion de Tunis



Diplôme de mastère en Ingénierie des systèmes D'information des entreprises

Conception et développement d'une Application web de Gestion des Factures des Achats et des Ventes

Réalisé par : Mlle. Maha ZNINE

Encadrant professionnel : M. Oussama Ouda

Encadreur académique : M. Nidhal Jelassi

Réalisé au sein de l'entreprise DEVLOPY



ANNÉE UNIVERSITAIRE : 2020– 2021

Avant-Propos

Cette étude est réalisée dans le cadre de la préparation du projet de fin d'études, au sein de la société **Devlopy**, pour l'obtention du diplôme universitaire en Informatique des système d'information d'entreprise.

Mon stage de fin d'études consiste à créer un projet intitulé « ***Conception et développement d'une application web de Gestion des Factures des ventes et des achats*** » pour concevoir et élaborer une plateforme web qui aide à suivre les dépenses, les paiements et à créer des factures de vente et d'achat, et des estimations professionnelles..

Ce stage a été une expérience enrichissante, il m'a permis de passer une formation, de découvrir des nouveaux concepts dans le monde de développement, de travailler avec une équipe créative, et d'établir un contact réel et direct avec le milieu professionnel.

Dédicaces

Je dédie ce travail :

À la mémoire de mon père,

qui m'a toujours poussée et motivée dans mes étude.

À ma mère,

pour tout le sacrifice, l'amour et le soutien tout au long de mes études.

À mes soeurs Manel, Maissa et mon frère M.Aziz ,

pour leurs encouragements permanents, et leur soutien moral.

À mes amis,

en témoignage de l'amitié sincère qui nous a liées et des bons moments passés ensemble.

Maha

Remerciements

Avant de commencer la présentation de ce rapport, je profite l'occasion pour remercier à tous ceux qui, de près ou de loin, ont contribué à la réalisation de ce travail.

En premier lieu, je tiens à exprimer mon profonde reconnaissance à mon encadrant académique **Mr. Nidhal Jelassi** pour ses conseils précieux, son soutien et surtout sa disponibilité pendant la période de confinement. Je le remercie vivement pour tout.

Je remercie également mon encadrant professionnel **Mr. Oussama Ouda** et toute la famille **Devlopy** pour leur accueil chaleureux et pour l'expérience qu'ils m'ont offert au sein d'une grande société.

Je tiens à exprimer toute ma gratitude et mon respect à l'équipe pédagogique de la faculté des sciences économiques et de gestion de Tunis et les intervenants professionnels responsables de ma formation, qui m'ont permis d'évoluer rapidement dans ce domaine.

Enfin, j'adresse mes plus sincères remerciements à tous les membres du jury qui m'ont honoré par leur présence et en acceptant de juger mon travail.

Table des matières

Avant-Propos	2
Dédicaces	3
Remerciements	4
Introduction générale	14
1 CADRE DU PROJET	16
Introduction	17
1.1 Présentation de l'entreprise d'accueil	17
1.2 Problématique	18
1.3 Étude de l'existant	18
1.4 Solution proposée	18
1.5 Spécification des besoins	18
1.5.1 Identifier les acteurs	18
1.5.2 Analyse des besoins fonctionnels	19
1.5.3 Besoins non fonctionnels	21
1.6 Méthodologie adoptée	21
1.6.1 Méthodes Agiles	21
1.6.2 Tableau comparatif	22
1.6.3 Choix de la méthode Scrum	23
1.7 Conclusion	24
2 Étude Préalable	25
Introduction	26
2.1 Pilotage du projet avec Scrum	26
2.1.1 Équipe et rôle	26
2.1.2 Backlog product	26
2.1.3 Diagramme des cas d'utilisation global	33
2.2 Spécification technique et technologique	34
2.2.1 Conception architecturale	34
2.2.2 Environnement de développement	35

2.3	Planification des Sprints	42
2.4	Conception statique	42
2.5	Chronogramme	45
2.6	Conclusion	45
3	Sprint 1 « Back-office »	46
	Introduction	47
3.1	Backlog du Sprint	47
3.2	Spécification	48
3.2.1	Cas d'utilisation « S'authentifier»	49
3.2.2	Cas d'utilisation «Gérer les utilisateurs»	50
3.3	Conception	53
3.3.1	Diagramme de classes participantes	53
3.3.2	Diagrammes de séquences	53
3.4	Réalisation et tests	57
3.4.1	Authentification	57
3.4.2	Ajout d'un utilisateur	57
3.4.3	Modification d'un utilisateur	58
3.4.4	Suppression d'un utilisateur	59
3.5	Conclusion	59
4	Sprint 2	60
	Introduction	61
4.1	Backlog du sprint	61
4.2	Spécification	61
4.2.1	Cas d'utilisation «Gestion des Clients»	61
4.2.2	Cas d'utilisation «Gestion des Produits»	64
4.2.3	Cas d'utilisation «Gestion des devises»	65
4.2.4	Cas d'utilisation «Gestion des Factures»	68
4.3	Conception	71
4.3.1	Diagramme de classes participantes	72
4.3.2	Diagrammes de séquences	72
4.4	Réalisation et tests	77
4.4.1	Gestion client	77
4.4.2	Gestion des produits	78
4.4.3	Gestion des Devises	79
4.4.4	Gestion des Factures	81
4.5	Conclusion	82
5	Sprint 3	83
	Introduction	84
5.1	Backlog du sprint	84
5.2	Spécifications fonctionnelles	86

5.2.1	Raffinement de cas d'utilisation «Gestion des ordres de livraison»	86
5.2.2	Raffinement de cas d'utilisation «Gestion des paiements»	88
5.2.3	Raffinement de cas d'utilisation «Gestion des dépenses»	90
5.3	Conception	92
5.3.1	Diagramme de classes participantes	92
5.3.2	Diagrammes de séquences	93
5.4	Réalisation et tests	98
5.4.1	Gestion d'ordre de livraison	98
5.4.2	Gestion des paiements	99
5.4.3	Gestion des Dépenses	101
5.5	Conclusion	101
6	Sprint 4	102
	Introduction	103
6.1	Backlog du sprint	103
6.2	Spécification	105
6.2.1	Cas d'utilisation «Gestion des Fournisseurs»	105
6.2.2	Cas d'utilisation «Gestion des Factures d'achat»	106
6.2.3	Raffinement de cas d'utilisation «Gestion des ordres de livraison d'achat»	107
6.3	Conception	110
6.3.1	Diagramme de classes participantes	110
6.3.2	Diagrammes de séquences	110
6.4	Réalisation et tests	114
6.4.1	Gestion Fournisseur	114
6.4.2	Gestion des Factures d'achat	115
6.4.3	Gestion des ordres de livraison d'achat	117
6.5	Conclusion	117
7	Sprint 5	118
	Introduction	119
7.1	Backlog du sprint	119
7.2	Spécifications fonctionnelles	121
7.2.1	Cas d'utilisation «Gestion de Tableau de bord»	121
7.2.2	Description textuelle « consulter le Tableau de bord »	122
7.2.3	Cas d'utilisation «Gestion des paramétrages»	123
7.2.4	Description textuelle « gérer compte »	123
7.2.5	Cas d'utilisation «Gestion des Rapports»	125
7.3	Conception	126
7.3.1	Diagramme de classes participantes	126
7.3.2	Diagrammes de séquences	127
7.4	Réalisation et tests	130
7.5	Conclusion	134

Table des figures

1.1 Devlopy Logo	17
1.2 Les étapes du cycle de vie Scrum	23
2.1 Diagramme de cas d'utilisation Global	33
2.2 Le modèle MVC	35
2.3 Logo du Framework Laravel	36
2.4 Logo du Framework Vue.js	37
2.5 Logo du Bootstrap	38
2.6 Logo du Docker	38
2.7 Logo du MySql	39
2.8 Logo du Yarn	40
2.9 Logo du Composer	41
2.10 Logo du Visual Studio	41
2.11 Logo du Overleaf	42
2.12 Schéma diagramme de classe générale	43
2.13 Diagramme de Gantt réel	45
3.1 Diagramme de raffinement du cas d'utilisation « S'authentifier»	49
3.2 Diagramme de raffinement du cas d'utilisation «Gérer utilisateurs»	50
3.3 Diagramme de classes du premier sprint	53
3.4 Diagramme de séquences Ajouter utilisateur	54
3.5 Diagramme de séquences Modifier utilisateur	55
3.6 Diagramme de séquences Supprimer utilisateur	56
3.7 Interface d'authentification	57
3.8 Interface de gestion des utilisateurs	58
3.9 Interface d'ajout d'un utilisateur	58
3.10 Interface de modification d'un utilisateur	59
3.11 Interface de suppression d'un utilisateur	59
4.1 Diagramme de raffinement du cas d'utilisation «Gestion des Clients»	63
4.2 Diagramme de raffinement du cas d'utilisation «Gestion des Produit»	64
4.3 Diagramme de raffinement du cas d'utilisation «Gestion des devises»	66
4.4 Diagramme de raffinement du cas d'utilisation «Gestion des factures»	69

4.5	Diagramme de classes du deuxième sprint	72
4.6	Diagramme de séquences «Modifie client»	73
4.7	Diagramme de séquences «Ajouter produit»	74
4.8	Diagramme de séquences «Recherche Devis»	75
4.9	Diagramme de séquences «Supprimer Facture»	76
4.10	« ajouter client»	77
4.11	« Recherche client»	78
4.12	« Modifier produit»	79
4.13	«Supprimer produit»	79
4.14	« Consulter Devis»	80
4.15	«envoyer devis par mail»	81
4.16	«Ajouter Facture 1»	81
4.17	«Résultat de l'ajout Facture»	82
5.1	«Diagramme de cas d'utilisation «Gestion des ordres de livraison»	86
5.2	«Diagramme de cas d'utilisation «Gestion des paiements»	88
5.3	«Diagramme de cas d'utilisation «Gestion des dépenses»	91
5.4	Diagramme de classes du troisième sprint	93
5.5	Diagramme de séquences «Modifier ordre de livraison»	94
5.6	Diagramme de séquences «Recherche ordre de livraison»	95
5.7	Diagramme de séquences «Ajouter paiement»	96
5.8	Diagramme de séquences «Supprimer dépense»	97
5.9	« Modifier ordre de livraison»	98
5.10	«Résultat Modifier ordre de livraison»	98
5.11	« Recherche ordre de livraison»	99
5.12	«Ajouter paiement »	99
5.13	«Résultat Ajouter paiement »	100
5.14	«Supprimer Dépense»	101
6.1	Diagramme de raffinement du cas d'utilisation «Gestion des Fournisseurs» . .	105
6.2	Diagramme de raffinement du cas d'utilisation «Gestion des factures d'achat»	106
6.3	«Diagramme de cas d'utilisation «Gestion des ordres de livraison d'achat» .	108
6.4	Diagramme de classes du quatrième sprint	110
6.5	Diagramme de séquences «consulter Fournisseur»	111
6.6	Diagramme de séquences «Recherche Facture d'achat»	112
6.7	Diagramme de séquences «Ajouter ordre de livraison d'achat»	113
6.8	« Consulter Fournisseur 1»	114
6.9	« Consulter Fournisseur 2»	115
6.10	« ajouter Facture d'achat 1»	115
6.11	« ajouter Facture d'achat 2»	116
6.12	« ajouter Facture d'achat 3»	116
6.13	« modifier ordre de livraison d'achat»	117

7.1	Diagramme de raffinement du cas d'utilisation «consulter le Tableau de bord»	122
7.2	Diagramme de raffinement du cas d'utilisation «Gestion des paramétrages» .	123
7.3	Diagramme de raffinement du cas d'utilisation «Gestion des Rapports»	125
7.4	Diagramme de classes du cinquième sprint	127
7.5	Diagramme de séquences « consulter le Tableau de bord»	128
7.6	Diagramme de séquences « Gestion du compte»	129
7.7	Diagramme de séquences «consulter les Rapport»	130
7.8	consulter le Tableau de bord 1	131
7.9	consulter le Tableau de bord 2	131
7.10	gérer compte	132
7.11	Gérer les informations de l'entreprise	132
7.12	consulter les Rapports	133
7.13	Gérer les informations de l'entreprise	133

Liste des tableaux

1.1	Tableau comparatif Méthode traditionnelle vs Méthode agile	22
2.1	Backlog du produit.	32
2.2	Planification des Sprints	42
3.1	Backlog du premier Sprint	48
3.2	Description textuelle du cas d'utilisation «S'authentifier»	49
3.3	Description textuelle du CU «Ajouter un utilisateur»	51
3.4	Description textuelle du CU «Modifier utilisateur»	52
4.1	Backlog du deuxième sprint	62
4.2	Description textuelle du CU «Modifier client»	63
4.3	Description textuelle du CU «Ajouter un produit»	65
4.4	Description textuelle du CU «envoyer de devis par mail»	67
4.5	Description textuelle du CU «convertir devis en facture»	68
4.6	Description textuelle du CU «recherche facture»	70
4.7	Description textuelle du CU «Supprimer facture»	71
5.1	Backlog du sprint 3	85
5.2	Description textuelle du CU «Ajouter ordre de livraison»	87
5.3	Description textuelle du CU «Modifier ordre de livraison»	87
5.4	Description textuelle du CU «Ajouter paiement»	89
5.5	Description textuelle du CU «Supprimer paiement»	90
5.6	Description textuelle du CU «recherche dépense»	92
6.1	Backlog du sprint 4	104
6.2	Description textuelle du CU «consulter fournisseur»	105
6.3	Description textuelle du CU «recherche facture d'achat»	107
6.4	Description textuelle du CU «Supprimer facture d'achat»	107
6.5	Description textuelle du CU «Ajouter ordre de livraison d'achat»	109
6.6	Description textuelle du CU «Modifier ordre de livraison d'achat»	109
7.1	Backlog du sprint 5	121
7.2	Description textuelle du CU «Consulter Tableau de bord»	122

7.3	Description textuelle du CU «gérer compte»	124
7.4	Description textuelle du CU «Gérer les informations de l'entreprise»	124
7.5	Description textuelle du CU «consulter les Rapports »	126

Introduction générale

Aujourd’hui l’informatique joue un rôle très important dans tous les domaines, et son rôle principal est de faciliter la vie quotidienne des êtres humains pour gagner le temps dans les travaux simples ou compliqués.

Pour acquérir une bonne et parfaite qualité, la formation théorique à elle seule ne suffit pas, il est donc nécessaire de suivre une démarche réelle permettant de voir comment se déroulent les tâches dans la vie professionnelle. A cet effet, et afin de valider nos études acquises au fil de cinq ans au sein de Faculté des Sciences économie et de gestion de Tunis et en vue de l’obtention de le Master en informatique des systèmes d’information d’entreprise , nous sommes amenées à effectuer un stage à la société Devlopy d’une durée de 4 mois.

Aujourd’hui, avec l’évolution et l’apparition des nouvelles technologies, La conciliation fait par des logiciels de comptabilité.

En ce sens, notre projet consiste à réaliser une plateforme web sécurisée qui permet de réceptionner des fichiers deux par deux, pour les comparer, et de sortir un compte rendu qui contient le résultat.

Le stage est considéré comme une occasion qui nous a permis le contact direct avec le marché du travail, il nous a aidé à renforcer la théorie par la pratique. Le sujet qui nous a été confié se résume comme suit : Application web pour la Gestion des Factures d’achat et de vente, de devis et de paiement. L’application doit être créée de manière à suivre en temps réel les factures d’achat et de vente de la société, les clients, les fournisseurs et les paiements. C'est-à-dire : Offrir une vision globale de l'état du suivi des factures en cours de traitement et aussi archivées. Il s'agit d'ajouter des factures et les régler et puis les archiver et être accessibles à n'importe quel moment, sans oublier le contrôle administratif de l'application qui va être dédié à un administrateur, Aussi, le suivi des client, des fournisseurs et les paiements. Certes, le bon fonctionnement de l'application et le respect du cahier de charges sont très importants. Pour cela, elle a été prise en considération tout au long de réalisation. Ceci dit notre travail se devise en trois chapitres structurés comme suit :

Le première chapitre intitulé « *Cadre du projet* » concerne la présentation du l’organisme d’accueil ainsi que le sujet de l’application.

Le deuxième chapitre intitulé « *Analyse et Spécification des Besoins* » qui vise à identifier

et décrire les besoins attendus de l'application.

Au niveau des autres chapitres, nous présentons pour chaque sprint les différentes étapes de sa mise en œuvre, la spécification des besoins, la conception et la réalisation de ses différentes tâches.

Enfin, ce rapport sera clôturé par une conclusion générale qui établit le bilan du travail et quelques perspectives envisageables au présent projet.

1

CADRE DU PROJET

Sommaire

Introduction	17
1.1 Présentation de l'entreprise d'accueil	17
1.2 Problématique	18
1.3 Étude de l'existant	18
1.4 Solution proposée	18
1.5 Spécification des besoins	18
1.5.1 Identifier les acteurs	18
1.5.2 Analyse des besoins fonctionnels	19
1.5.3 Besoins non fonctionnels	21
1.6 Méthodologie adoptée	21
1.6.1 Méthodes Agiles	21
1.6.2 Tableau comparatif	22
1.6.3 Choix de la méthode Scrum	23
1.7 Conclusion	24

Introduction

Ce chapitre est consacré à présenter le cadre général du projet. Il s'agit en effet d'une présentation de l'organisme d'accueil, de la problématique du projet, ainsi que de l'étude de l'existant. Après l'exposition de l'étude de l'existant, nous allons présenter la solution proposée. Puis, Nous allons identifier les fonctionnalités et les acteurs du système. Enfin, nous justifierons notre choix concernant la méthodologie de gestion de projet adoptée pour la réalisation de notre application à travers une étude comparative des principales méthodes agiles.

1.1 Présentation de l'entreprise d'accueil

Dans cette partie nous présentons l'organisme d'accueil dans laquelle s'est déroulé notre stage, et nous identifions ses services.



FIGURE 1.1 – Devlopy Logo

Devlopy, entreprise de développement informatique, créé en 2018, totalement Tunisienne, met à votre disposition son savoir-faire, ses services et ses produits afin de réaliser et réussir vos projets. Nous sommes les spécialistes de développement sur mesure , développement des logiciels , application mobile, site web et surtout les Logiciels en tant que Service (SaaS).

Les activités de DEVLOPY s'articulent autour de 4 axes :

1. Créations des solutions informatiques innovantes.
2. Intégration de solutions RFID, codes à barres, NFC, biométrie et GPS.
3. Développement de produits et de solutions spécifiques.
4. S'aligner aux nouvelles technologies.

DEVLOPY est composée de deux groupes., Le premier groupe est spécialisé dans le développement des solutions mobile. Le deuxième est spécialisé dans le développement des applications web. Elle est implantée dans la ville Tunis à la Marsa.

1.2 Problématique

Chaque employé de chaque entité intervenante dans la gestion des factures établi sa propre organisation, ils peuvent se retrouver avec des différents supports comme des fichiers Excel, des dizaines de pages éparses dans une archive, ce qui rend difficile et pas pratique le partage et la mise à jour des informations. Ainsi l'administrateur ne peut pas connaître l'état des factures au cours de traitement, ce qui rend leur gestion plus complexe et leur évaluation un travail pénible. Pour faciliter ces tâches, les responsables ont opté pour une informatisation de leurs besoins et l'ajout d'autres fonctionnalités. D'où la proposition de ce stage pour la réalisation de l'application « **Conception et développement d'une application web de Gestion des Factures des ventes et des achats** » qui doit être une application intranet facile et efficace.

1.3 Étude de l'existant

Toutes les entreprises possèdent des stocks. Ces stocks représentent dans la plupart des cas entre 10 % et 30 % du capital financier immobilisé. Ce qui pousse les entreprises à réfléchir d'une manière sérieuse à une gestion efficace de l'ensemble des processus d'achats, de ventes et de stockage. En effet, cette gestion a connu une évolution : la plupart des entreprises sont passés d'une gestion manuelle, où le magasinier procède aux comptages d'une manière physique, à une gestion automatisée à l'aide des outils informatiques.

1.4 Solution proposée

L'étude de l'existant nous a permis de dégager plusieurs anomalies. Pour faire recours à ces anomalies nous proposons de concevoir et d'implémenter une plateforme web sécurisée qui vous aide à suivre les dépenses, les paiements et à créer des factures des achats et des ventes et des estimations professionnelles.

1.5 Spécification des besoins

Nous commençons par l'identification des acteurs puis l'extraction des besoins fonctionnels et non fonctionnels.

1.5.1 Identifier les acteurs

Un acteur représente un rôle joué par une entité externe (utilisateur humain, dispositif matériel ou autre système) qui interagit directement avec le système étudié. Un acteur peut consulter et/ou modifier directement l'état du système, en émettant et/ou en recevant des messages susceptibles d'être porteurs de données.. Les acteurs intervenant dans notre application sont :

L'administrateur : Il est le responsable des différents paramétrages de l'application. Il a un contrôle absolu et l'accès total à toutes les fonctionnalités et les services offerts par le système. Il a la charge d'administrer tous les aspects de l'application.

L'utilisateur : C'est l'utilisateur qui peut bénéficier des fonctionnalités de la plate-forme mais avec un accès limité.

Visiteur : Il consulte les factures partagées.

1.5.2 Analyse des besoins fonctionnels

- Gestion des Clients :**

-Ajouter les informations essentielles de chaque client, Adresse de facturation et l'adresse de livraison-

-Modifier les informations des clients.

-Saisie des clients.

-Rechercher facilement des clients.

-Option de suppression des clients.

- Gestion des devises :**

-Création des devises : Choisir client , produit et les Dates, Ajouter des notes , Changer le type de Rabais (fixe, pourcentage) , Possibilités d'ajouter une nouvelle taxe , Modifier le modèle de document des ventes fourni.

-Modifier et supprimer tous les informations des devises en suivant la logique de la vente.

-Rechercher facilement des Devises.

- Gestion des Factures :**

-Création des factures : Choisir client , produit et Date de facturation, Ajouter des notes , Changer le type de Rabais (fixe, pourcentage), Timbre Fiscale , Possibilités d'ajouter une nouvelle taxe , Modifier le modèle de document des ventes fourni, gestion du matricule fiscale, ajouter le Timbre.

-Modifier et supprimer tous les informations des factures en suivant la logique de la vente.

-Rechercher facilement des factures

- Gestion des Produits :**

-Ajouter des produits.

-Rechercher facilement des produits.

-Option de suppression et modification des produits.

- Gestion des ordres de livraison :**

- Ajouter des nouvelles ordres de livraison.
- Rechercher facilement des ordres de livraison.
- Option de suppression et modification des ordres de livraison .

- **Gestion des Paiements :**

- Ajouter des nouveaux paiements.
- Rechercher facilement des paiements.
- Option de suppression et modification des paiements.

- **Gestion des Dépenses :**

- Ajouter des nouvelles Dépenses (Option d'ajouter des nouvelles catégories)
- Rechercher facilement des Dépenses.
- Option de suppression et modification des Dépenses.

- **Gestion des utilisateurs :**

- Ajouter des nouveaux utilisateurs
- Rechercher facilement des utilisateurs.
- Option de suppression et modification des utilisateurs .

- **Gestion des Rapports :**

- Rapports des ventes par jour, semaine, mois, trimestre et année.
- Rapports des profits par jour, semaine, mois, trimestre et année.
- Rapports des Dépenses par jour, semaine, mois, trimestre et année.
- Rapports des Taxes par jour, semaine, mois, trimestre et année.
- Option de télécharger le document en PDF et de l'imprimer.

- **Gestion des paramétrages (Réglages) :**

- Gestion de compte.
- Gestion des informations de l'entreprise.
- Gestion des Préférences.
- Gestion des personnalisations de Les factures, d'estimations, Paiements, Articles et ordres de livraison.
- Gestion des Notifications.
- Gestion des Taxes.
- Gestion des Modes de paiement.
- Gestion des champs personnalisés
- Gestion des notes.
- Gestion des Catégories de dépenses.
- Gestion de mail de configuration.
- Gestion des disque de fichiers.
- Gestion de sauvegarde.
- Gestion de mise à jour l'application.

1.5.3 Besoins non fonctionnels

Dans cette perspective et vue l'importance de cette gestion dans le fonctionnement des entreprises, se présente l'application de notre stage qui permet de gérer tous les éléments concernant les ventes effectuées. Donc l'objectif technique de cette application web est de garantir les besoins non fonctionnels suivants

- Facilité d'utilisation :

Pas besoin de connaître la comptabilité, Créer les documents de ventes, devis, factures, ordres de commande, ordres de livraison, relancer les clients et enregistrer les règlements.

- Portabilité :

Les devis et factures, l'état du stock ou le suivi des règlements sont toujours à portée de main, sur l'ordinateur, la tablette ou votre téléphone portable, réaliser les affaires facilement et instantanément.

- Efficacité

Grâce aux outils de suivi des clients et des commerciaux, prévoyez les actions commerciales à entreprendre et économiser du temps et des efforts pour gagner en efficacité. Se concentrer sur l'objectif principal.

1.6 Méthodologie adoptée

En créant une application, nous devons procéder selon une démarche bien définie et nous devons se baser sur des principes et des méthodes afin de réduire au maximum les chances d'échecs. Dans notre cas, nous allons s'intéresser aux méthodes agiles qui se proclament comme une méthodologie efficace qui représente une approche de gestion de projet et de développement itérative et incrémentale. Pour bien conduire notre projet, nous avons opté pour la méthode agile « Scrum » comme méthodologie de conception et de développement. Après le choix de la méthodologie, nous avons besoin d'un langage de modélisation unifié pour la modélisation de notre projet. Pour cela nous avons choisi l'UML comme un langage de modélisation qui est utilisé par toutes les méthodes agiles y compris la méthode Scrum.

1.6.1 Méthodes Agiles

Les méthodes de développement dites « méthodes agiles » (en anglais Agile Modeling) visent à réduire le cycle de vie du logiciel (donc accélérer son développement) en développant une version minimale, puis en intégrant les fonctionnalités par un processus itératif basé sur une écoute client et des tests tout au long du cycle de développement. L'origine des méthodes agiles est liée à l'instabilité de l'environnement technologique et au fait que le client est souvent dans l'incapacité de définir ses besoins de manière exhaustive dès le début

Chapitre 1 : Cadre du projet

du projet. De nos jours, les méthodes agiles sont légion dans la littérature. Il ne s'agit pas d'en choisir la ordene, mais de sélectionner les meilleures pratiques à mettre en œuvre dans chaque contexte. Elles ont toutefois des affinités avec les différentes typologies de projet et présentent donc un positionnement différent.

1.6.2 Tableau comparatif

Le choix de la méthode de travail est très important pour fixer les étapes à suivre. Pour ce faire, nous avons établi une comparaison basée par des indicateurs pertinents et sur des études existantes :

Critères	Scrum	Kanban	Scrumban
Itération	Oui(Sprints)	Selon besoin	Selon besoin
Rôles	<ul style="list-style-type: none">• Product Owner• Scrum Master• Team		Équipe spécialisée
Changement de périmente	Doit attendre le sprint suivant	Selon besoin	Selon besoin
Planification	Au début de chaque sprint	Kanban board, flux continu	Kanban board avec itération
Estimation	oui	Optionnelle	
Règle	Très descriptif	Non descriptif	descriptif
Bords/Artifacts	<ul style="list-style-type: none">• Product Backlog• Scrum board• Burndown chart	<ul style="list-style-type: none">• Kanban board• Diagramme des flux cumulés	<ul style="list-style-type: none">• Kanban board• Diagramme des flux cumulés

TABLE 1.1 – Tableau comparatif Méthode traditionnelle vs Méthode agile

- Scrum : est à ce jour la méthode agile la plus utilisée par les équipes. Ses caractéristiques : des itérations courtes (sprints de 4 semaines maximum), 3 rôles clés dans l'organisation (Scrum master, product owner, équipe de développement) et des timebox. La méthode Scrum est une excellente solution quand il s'agit d'améliorer la productivité d'une équipe, son engagement et la scalabilité des projets. Son caractère cadré et codifié est idéal pour des feuilles de route clairement définies. Scrum est en effet très axé sur le calendrier, contrairement à Kanban.

- Kanban : La méthode Kanban est surtout appliquée dans des projets de maintenance corrective ou d'applications en flux. On y trouve par exemple la gestion de tickets ou le

support. Le concept de Kanban est en effet basé sur une amélioration continue évolutive. Dans la méthode Kanban, il n'existe pas de rôles attitrés, ces derniers évoluent en fonction des besoins du projet et de l'organisation.

- Scrumban : est un mélange des deux concepts « scrum » et « kanban ». Elle conserve l'approche très cadrée de Scrum tout en permettant de mener des projets en flux continu. Scrumban possède une méthode de travail proche de celle de Kanban. Les itérations sont limitées, tout comme les tâches à effectuer, afin de permettre aux équipes de s'adapter et modifier un plan d'actions si nécessaire. La méthode s'appuie également sur une planification basée sur la demande, les équipes ne prévoient plus des tâches pour un sprint complet mais doivent désormais les prioriser au fur et à mesure de l'avancement du projet.

1.6.3 Choix de la méthode Scrum

Pour que la réussite de notre projet soit garantie, nous avons choisi Scrum comme une méthodologie agile. Scrum devrait garantir le succès du projet dans le sens où il permettra d'assurer que le produit final, livré, répond bien aux besoins du client. Il comprend également la possibilité de modifier les exigences formulées et permet d'améliorer la productivité de l'équipe. La méthode retenue est ainsi fondée sur le principe de répartition du projet sur une durée pouvant aller de deux semaines à un mois. L'objectif de ces durées est de parvenir à réaliser certaines des fonctionnalités qui seront réunies dans un backlog de produit. Ce dernier explique les besoins du client durant l'exécution du travail. Scrum favorise une grande importance aux réunions quotidiennes afin de se préparer aux difficultés rencontrées. Grâce à cette méthodologie, nous pouvons réduire le travail inutile en nous concentrant exclusivement sur ce que le client demande et en assurant la transparence de notre processus. La figure 1.2 illustre le cycle de vie d'un projet en utilisant la méthode Scrum :

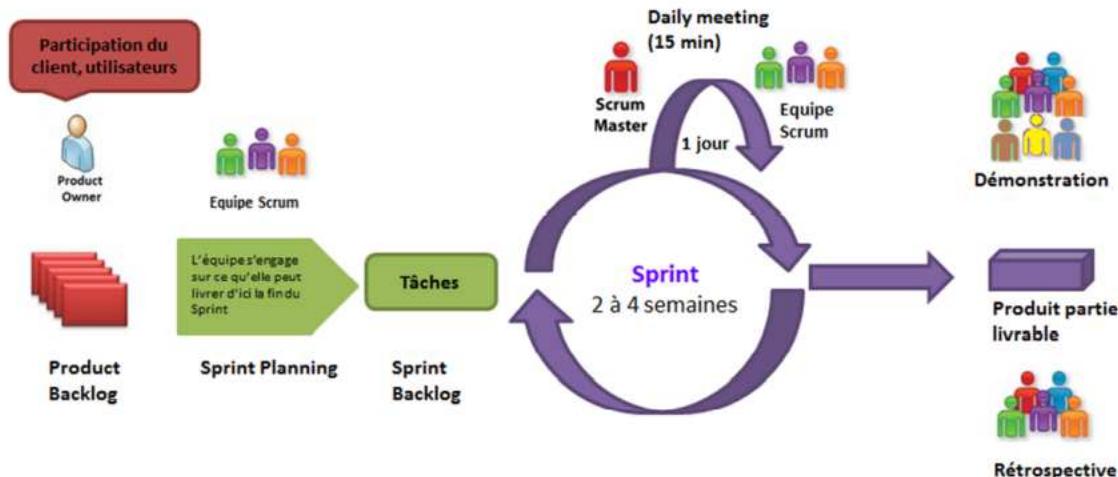


FIGURE 1.2 – Les étapes du cycle de vie Scrum

1.7 Conclusion

Cette phase consiste à comprendre le contexte du système. Nous avons commencé par la présentation de l'organisme en déterminant la problématique et en proposant la solution pour faire face à la situation courante. Puis, Nous avons identifié les fonctionnalités et les acteurs les plus pertinents. Enfin, nous avons présenté la méthodologie utilisée pour la réalisation de ce projet.

2

Étude Préalable

Sommaire

Introduction	26
2.1 Pilotage du projet avec Scrum	26
2.1.1 Équipe et rôle	26
2.1.2 Backlog product	26
2.1.3 Diagramme des cas d'utilisation global	33
2.2 Spécification technique et technologique	34
2.2.1 Conception architecturale	34
2.2.2 Environnement de développement	35
1 -Le Framework Laravel	35
2- Le Framework Vue.Js	36
3- Environnement matériel	37
4- Environnement logiciel	37
2.3 Planification des Sprints	42
2.4 Conception statique	42
2.5 Chronogramme	45
2.6 Conclusion	45

Introduction

L'étude préalable est une étape fondamentale du cycle de développement de notre application basé sur la méthode Scrum. Nous commencerons d'abord par la présentation du pilotage du projet avec scrum le backlog du produit et la planification des sprints afin de construire une ordre vision de la solution. Puis, nous définirons tous les besoins qui seront modélisés par le diagramme du cas d'utilisation général. Cette phase sera consacrée également pour la présentation de l'environnement de travail et les technologies utilisées pour la réalisation de notre application.

2.1 Pilotage du projet avec Scrum

Le cadre Scrum consiste en une équipe avec des rôles bien définis

2.1.1 Équipe et rôle

Scrum définit seulement 3 rôles :

- Le Product Owner (le propriétaire du produit) : c'est le responsable du produit, il représente les clients et les utilisateurs en transcrivant leurs besoins, définit et priori les demandes produit.
- Scrum Team (l'équipe de Scrum) : L'équipe se gère en toute autonomie et a la charge du développement du produit. Il n'y a pas de notion de hiérarchie, toutes les décisions sont prises ensemble. Elle regroupe les rôles habituellement nécessaires à un projet (architecte, concepteur, développeur, etc.).
- Le Scrum Master (le directeur de produit) : qui n'est pas le chef de projet mais il a pour charge de faciliter l'application de Scrum, sa mission est de tout mettre en œuvre pour que l'équipe travaille dans de ordre conditions et se concentre sur l'objectif du projet. Il porte également une attention particulière au respect des différentes phases de Scrum. Dans le contexte de notre projet, notre encadrant au niveau de l'organisme Mr Oussama Ouda sera le propriétaire de ce produit, notre encadrant au sein de la FSEGT Mr Nidhal Jelassi sera la Scrum Master de notre projet et nous-même Maha Znine, le seul membre de l'équipe Scrum.

2.1.2 Backlog product

Le backlog du produit est l'ensemble des fonctionnalités attendues du produit. C'est l'ensemble des spécifications fonctionnelles et techniques de ce dernier. « Le Backlog de produit est un élément indispensable de toute démarche Scrum. C'est une liste ordonnée de tout ce qui pourrait être requis dans le produit. Il retrace la vision utilisateur et reste ouvert

Chapitre 2 : Étude Préalable

et évolutif sur toute la durée du projet. Géré sous la responsabilité du PO, il est créé par décomposition successive à partir de la vision du produit. Le tableau 2.1 décrit le Backlog produit. Les priorités sont choisies d'après l'apparence chronologique et l'importance entre quelques « User stories ».

Chapitre 2 : Étude Préalable

ID	Fonctionnalité	User Story	sprint	Priorité
1	Authentification	En tant qu'utilisateur, je veux m'authentifier Afin d'accéder à mon interface.	1	Élevée
		En tant qu'administrateur, je veux m'authentifier. Afin d'accéder à mon interface		
2	Gestion des utilisateurs	En tant qu'administrateur, je veux ajouter un ou plusieurs utilisateurs.	1	Élevée
		En tant qu'administrateur, je veux modifier un utilisateur bien déterminée.		
		En tant qu'administrateur, je veux supprimer un utilisateur bien déterminée.	1	Élevée
		En tant qu'administrateur, je veux consulter la liste des utilisateurs.		
		En tant qu'administrateur je veux réaliser tous les taches de l'utilisateur.	1	Élevée
3	Gestion des Clients	En tant qu'utilisateur, je veux ajouter un ou plusieurs clients.	2	Élevée
		En tant qu'utilisateur, je veux modifier un client bien déterminée.		
		En tant qu'administrateur, je veux consulter la liste des Clients.	2	Élevée
		En tant qu'administrateur je veux réaliser tous les taches du Client.		
4	Gestion des devises	En tant qu'utilisateur, je veux ajouter un ou plusieurs devises.	2	Élevée
		En tant qu'utilisateur, je veux modifier un devis bien déterminée.		
		En tant qu'utilisateur, je veux supprimer un devis bien déterminée.	2	Élevée
		En tant qu'utilisateur, je veux consulter la liste des devises.		

Chapitre 2 : Étude Préalable

ID	Fonctionnalité	User Story	sprint	Priorité
5	Gestion des factures	En tant qu'utilisateur, je veux ajouter une ou plusieurs factures.	2	Élevée
		En tant qu'utilisateur, je veux modifier une facture bien déterminée.	2	Élevée
		En tant qu'utilisateur, je veux supprimer une facture bien déterminée.	2	Élevée
		En tant qu'utilisateur, je veux consulter la liste des factures.	2	Moyenne
6	Gestion des Produits	En tant qu'utilisateur, je veux ajouter un ou plusieurs produits.	2	Élevée
		En tant qu'utilisateur, je veux modifier un produit bien déterminée.	2	Élevée
		En tant qu'utilisateur, je veux supprimer un produit bien déterminée.	2	Élevée
		En tant qu'utilisateur, je veux consulter la liste des produits.	2	Moyenne
7	Gestion des Paiements	En tant qu'utilisateur, je veux ajouter une ou plusieurs paiements.	3	Élevée
		En tant qu'utilisateur, je veux modifier une paiement bien déterminée.	3	Élevée
		En tant qu'utilisateur, je veux supprimer une paiement bien déterminée.	3	Élevée
		En tant qu'utilisateur, je veux consulter la liste des paiements.	3	Moyenne
8	Gestion des ordres de livraison	En tant qu'utilisateur, je veux ajouter une ou plusieurs ordres de livraison.	3	Élevée
		En tant qu'utilisateur, je veux modifier une ordre de livraison bien déterminée.	3	Élevée
		En tant qu'utilisateur, je veux supprimer une ordre de livraison bien déterminée.	3	Élevée
		En tant qu'utilisateur, je veux consulter la liste des ordres de livraison	3	Moyenne

Chapitre 2 : Étude Préalable

ID	Fonctionnalité	User Story	sprint	Priorité
9	Gestion des Dépenses	En tant qu'utilisateur, je veux ajouter une ou plusieurs ordres de livraison.	3	Élevée
		En tant qu'utilisateur, je veux modifier une ordre de livraison bien déterminée.	3	Élevée
		En tant qu'utilisateur, je veux supprimer une ordre de livraison. bien déterminée.	3	Élevée
		En tant qu'utilisateur, je veux consulter la liste des ordres de livraison..	3	Moyenne
10	Gestion des Fournisseurs	En tant qu'utilisateur, je veux ajouter un ou plusieurs Fournisseurs.	4	Élevée
		En tant qu'utilisateur, je veux modifier un Fournisseur bien déterminée.	4	Élevée
		En tant qu'administrateur, je veux consulter la liste des Fournisseurs.	4	Élevée
		En tant qu'administrateur je veux réaliser tous les tâches du Fournisseurs.	4	Moyenne
11	Gestion des factures d'achat	En tant qu'utilisateur, je veux ajouter une ou plusieurs factures d'achat.	4	Élevée
		En tant qu'utilisateur, je veux modifier une facture d'achat bien déterminée.	4	Élevée
		En tant qu'utilisateur, je veux supprimer une facture d'achat bien déterminée.	4	Élevée
		En tant qu'utilisateur, je veux consulter la liste des factures d'achat.	4	Moyenne

Chapitre 2 : Étude Préalable

ID	Fonctionnalité	User Story	sprint	Priorité
12	Gestion des ordres de livraison d'achat	En tant qu'utilisateur, je veux ajouter une ou plusieurs ordres de livraison d'achat.	4	Élevée
		En tant qu'utilisateur, je veux modifier une ordre de livraison d'achat bien déterminée.	4	Élevée
		En tant qu'utilisateur, je veux supprimer une ordre de livraison d'achat. bien déterminé.	4	Élevée
		En tant qu'utilisateur, je veux consulter la liste des ordres de livraison d'achat.	4	Moyenne
13	Gestion de Tableau de bord	En tant qu'utilisateur, je veux consulter toutes les mesures clés du projet : le montant du, nombre des client, nombre des factures, nombre des devises, montant des ventes, montant reçus, montant des dépenses, et le revenu net	5	Moyenne
		En tant qu'utilisateur, je veux avoir une courbe statistique récapitulatif hebdomadaire fournit un bref résumé des ventes , le montant reçus et des dépenses de l'année passée et cette année	5	Moyenne
		En tant qu'utilisateur, je veux consulter les Factures dues et Estimations récentes sur le tableau de bord	5	Moyenne

ID	Fonctionnalité	User Story	sprint	textbf{Priorité}
14	Gestion des paramétrages (Réglages)	En tant qu'utilisateur, je veux gérer mon compte	5	Élevée
		En tant qu'utilisateur, je veux gérer les informations de l'entreprise	5	Élevée
		En tant qu'utilisateur, je veux gérer les Préférences.	5	Moyenne
		En tant qu'utilisateur, je veux gérer les Taxes, les Modes de paiement.	5	Moyenne
15	Gestion des Rapports	En tant qu'Administrateur, je veux consulter les Rapports des ventes, les Rapports des profits , les Rapports des Dépenses et les Rapports des Taxes par jour, semaine, mois, trimestre et année.	5	Moyenne
		En tant qu'Administrateur, je veux télécharger le document en PDF et de l'imprimer	5	Faible
16	Consulter les factures partager	En tant que visiteur, je veux consulter mes factures.	5	Faible

TABLE 2.1 – Backlog du produit.

2.1.3 Diagramme des cas d'utilisation global

Le diagramme de cas d'utilisation complet nous permet d'avoir une vision générale du comportement de notre système. La figure 2.1 Représente le diagramme de cas d'utilisation de notre projet :

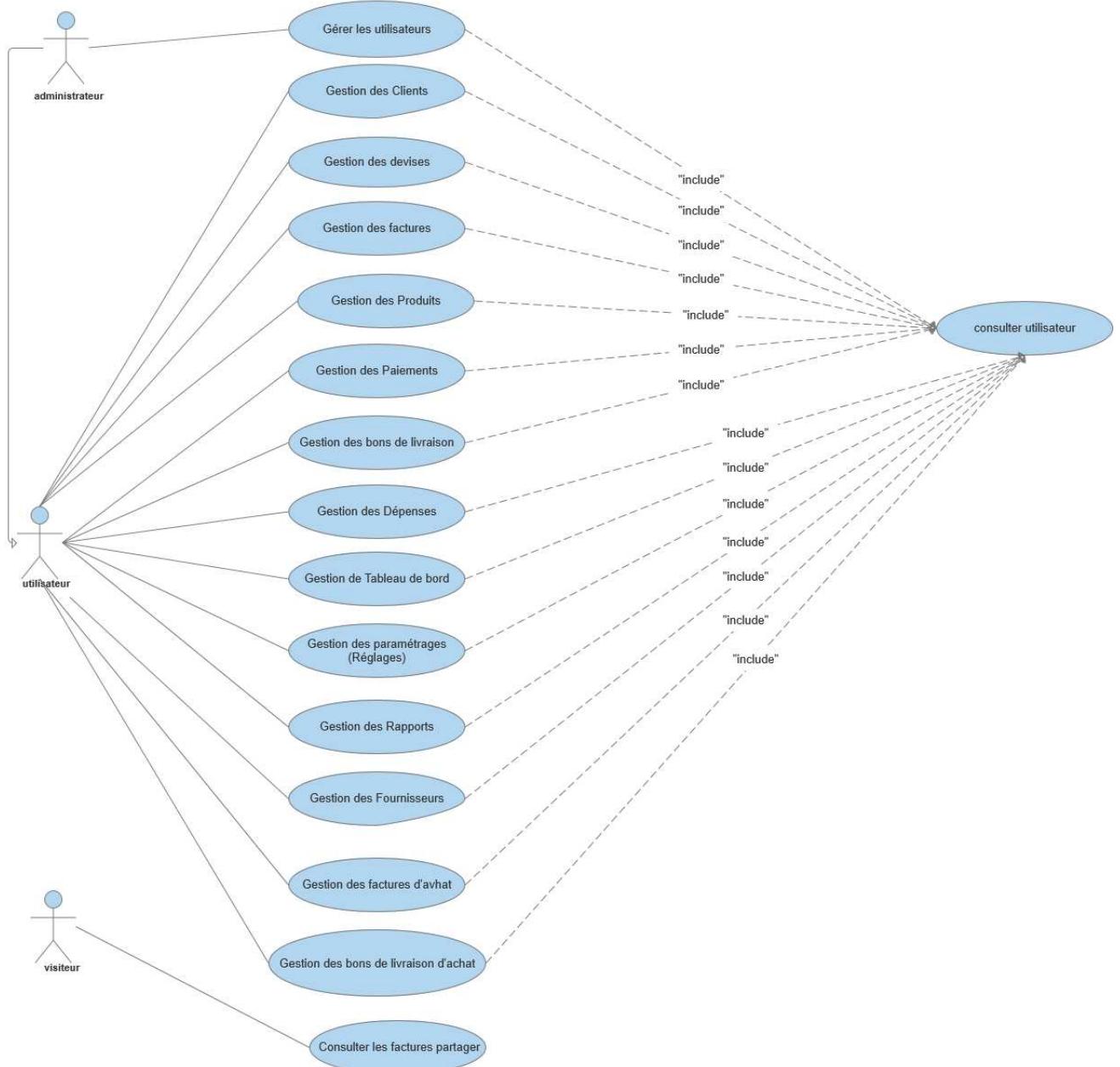


FIGURE 2.1 – Diagramme de cas d'utilisation Global

2.2 Spécification technique et technologique

En considérant l'analyse des besoins mentionnée au début de ce chapitre et vu son intérêt pour notre projet, il est temps de savoir quels outils allons nous utiliser et sur quelles architectures notre choix va se porter

2.2.1 Conception architecturale

Le pattern modèle-vue-contrôleur(en abrégé MVC, de l'anglais model-view-controller), 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. Ses avantages :

- Séparation des compétences (design, base de données, application)
- Simplicité de mise à jour.
- Vitesse de création de pages.

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).

Nous expliquons ces trois parties l'une après l'autre :

a. Le modèle (ou Model) :

Le modèle représente les structures de données. Typiquement, les classes modèles contiennent des fonctions qui aident à récupérer, à insérer et à mettre à jour des informations de la base de données.

Par exemple, lorsque nous disons« le contrôleur récupère les données d'un outil», il va en fait, faire appel au modèle Outil (« Tool »). C'est le modèle qui peut récupérer les données de cet outil, généralement via une requête au serveur SQL. Au final, il permet au contrôleur de manipuler les outils mais sans savoir comment ils sont stockés, gérés, etc, C'est une couche d'abstraction.

b. La vue (ou View) :

La vue correspond à l'interface avec laquelle l'utilisateur interagit. Elle se présente sous la forme d'une Template représentant l'interface. Reprenons l'exemple de l'outil. Ce n'est pas le contrôleur qui affiche le formulaire, il ne fait qu'appeler la ordene vue. Si nous avons une vue formulaire, les balises HTML du formulaire d'édition de l'outil y seront et finalement le contrôleur ne fera qu'afficher cette vue sans savoir vraiment ce qu'il y a dedans. Donc en pratique, c'est le « designer » d'un projet qui travaille sur les vues. La séparation de vues et contrôleurs permet aux designers et aux développeurs PHP de travailler ensemble sans besoin de contact direct.

c. Le contrôleur (ou Controller) :

Il gère l'interface entre le modèle et le client. Il va interpréter la requête de ce dernier pour lui envoyer la vue correspondante. Il effectue la synchronisation entre le modèle et les vues.

Il est la couche qui se charge d'analyser et de traiter la requête de l'utilisateur. Le contrôleur contient la logique de notre application et va se contenter « d'utiliser » les autres composants : les modèles et les vues. Concrètement, un contrôleur va récupérer, par exemple, les informations sur l'utilisateur courant, vérifier qu'il a le droit de modifier un tel outil, récupérer les données de cet outil et demander la page du formulaire d'édition de l'outil.

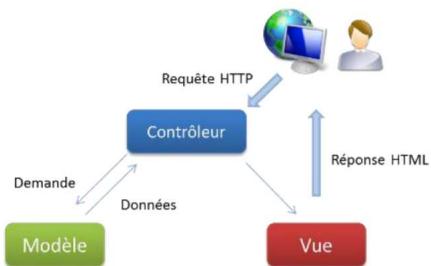


FIGURE 2.2 – Le modèle MVC

2.2.2 Environnement de développement

Cette section présente l'environnement matériel mis à la disposition du présent projet ainsi que l'environnement logiciel qui a permis à l'aboutissement de la mise en œuvre de l'application

1 -Le Framework Laravel

Laravel est un Framework web gratuit, open-source, créé par Taylor Otwell et destiné au développement d'applications Web suivant le modèle-vue-contrôleur (MVC). Certaines des caractéristiques de Laravel sont un système modulaire système d'emballage avec un gestionnaire de dépendances dédié, pour accéder à différentes façons de bases de données relationnelles, les services publics que l'aide au déploiement d'applications et de maintenance, et son orientation vers le sucre syntaxique En Mars 2015, Laravel est considéré comme l'une des plus populaires Framework PHP. Le code source de Laravel est hébergé sur Git Hub sous licence conformément aux termes de licence MIT.



FIGURE 2.3 – Logo du Framework Laravel

Structure du répertoire

- Répertoire App :

Contient le code de base de l'application.

Le répertoire App contient une variété de répertoires supplémentaires tels que console, http qui contient les contrôleurs et Providers et les modèles d'application.

- Répertoire Bootstrap :

Contient des fichiers d'auto loading.

- Répertoire Config :

Comme son nom l'indique, contient tous les fichiers de configuration de l'application.

- Répertoire data base :

Contient la migration de la base de données.

- Répertoire public :

Contient le fichier, ce qui est le point d'entrée pour toutes les demandes entrant dans l'application. Ce répertoire contient également les images, JavaScript et CSS.

- Répertoire Routes :

Contient toutes les définitions de route pour l'application. Par défaut, plusieurs fichiers d'itinéraire sont inclus avec Laravel :

- web.php

- api.php

- console.php

2- Le Framework Vue.Js

Vue est un framework évolutif pour construire des interfaces utilisateur. À la différence des autres frameworks monolithiques, Vue a été conçu et pensé pour pouvoir être adopté de manière incrémentale. Le cœur de la bibliothèque se concentre uniquement sur la partie vue, et il est vraiment simple de l'intégrer avec d'autres bibliothèques ou projets existants. D'un autre côté, Vue est tout à fait capable de faire tourner des applications web monopages quand il est couplé avec des outils modernes et des bibliothèques complémentaires.

JavaScript est utilisé uniquement pour améliorer certaines vues. Cependant, pour des projets plus complexes, ou bien quand le front-end est entièrement généré par JavaScript, des désavantages se manifestent :

- Les définitions globales forcent à avoir un nom unique pour chaque composant
- Les templates sous forme de chaînes de caractères ne bénéficient pas de la coloration syntaxique et requièrent l'usage de slashes disgracieux pour le HTML multiligne.
- L'absence de support pour le CSS signifie que le CSS ne peut pas être modularisé comme HTML et JavaScript
- L'absence d'étape de build nous restreint au HTML et à JavaScript ES5, sans pouvoir utiliser des préprocesseurs tels que Babel ou Pug (anciennement Jade).



FIGURE 2.4 – Logo du Framework Vue.js

3- Environnement matériel

Pour le développement de notre application nous avons utilisé un PC portable «HP ProBook» dont la configuration est la suivante :

- Processeur Intel Core i5-8265U avec fréquence 1.80 GHz
- Quantité de mémoire vive 8 Go
- Capacité du disque dur 8 Go

4- Environnement logiciel

1. Application Web

- Bootstrap :

Bootstrap est un Framework qui facilite et accélère le développement Front-End. Il inclue une base CSS très complète (au format LESS) configurée à partir d'un fichier de variables, un ensemble de conventions de structure HTML et de nommage de classes des librairies JavaScripts simples pour les fonctions les plus courantes.



FIGURE 2.5 – Logo du Bootstrap

- **Docker**

Docker est une technologie qui simplifie et accélère le processus de déploiement de logiciels en créant un environnement de travail unifié et un emballage dans un conteneur à partir duquel les développeurs peuvent travailler sans perdre de temps. Il permet d'installer les logiciels, les dépendances, l'environnement de configuration des bibliothèques. Et le conteneur prend aussi en charge le déploiement sur le serveur.”

- Au cours des tests des technologies, j'ai également passé une semaine et demi à mettre en place le docker et le docker-compose. Actuellement j'ai initialisé un conteneur pour déployer le frontend. Pour créer un environnement docker complet pour le backend et la base de données, il faudra plus de temps, car GeoNature est un projet relativement complexe. La complexité de GeoNature est due aux applications dont il dépend tel que TaxHub et UsersHub. Afin de créer un environnement pour le docker il faut donc beaucoup de temps et il est nécessaire de savoir comment configurer l'environnement pour chaque application

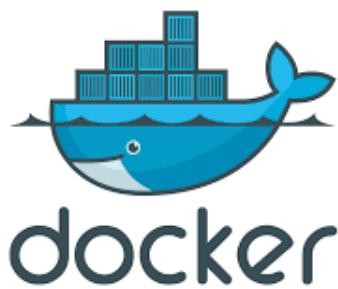


FIGURE 2.6 – Logo du Docker

- **Blade**

Blade est le moteur de modélisation simple, mais puissant. Contrairement à d'autres moteurs de modèles PHP populaires, Blade n'empêche pas l'utilisation du code PHP simple

dans les vues. En fait, toutes les vues de Blade sont compilées en code PHP ordinaire et mises en cache jusqu'à ce qu'elles soient modifiées, ce qui signifie que Blade ajoute une surcharge de zéro à l'application. Les fichiers de vue Blade utilisent l'extension « .blade.php » et sont généralement stockés dans le répertoire ressources/vies.

- **PHP Storm**

PhpStorm est un éditeur pour PHP, HTML et JavaScript, édité par l'entreprise informatique JetBrains. Il soutient PHP 5.3, 5.4 et 5.5 pour des projets modernes et anciens. L'IDE fournit les meilleurs codes auto-complétion, sur la volée de prévention d'erreur, soutient le mélange de langues et plus. Il traite automatiquement le code avec précaution, pour aider les développeurs à effectuer des changements globaux du projet facilement et en toute sécurité.

- **Eloquent**

L'ORM Eloquent fournit une simple implémentation Active Record pour travailler avec la base de données. Chaque table de base de données a un "Modèle" correspondant qui est utilisé pour interagir avec cette table. Les modèles les permettent de demander des données dans les tableaux, ainsi que d'insérer de nouveaux enregistrements dans la table.

- **MySQL**

MySQL est un système de gestion de base de données relationnelle (SGBDR). Il est distribué sous une double licence GPL et propriétaire. Il fait partie des logiciels de gestion de base de données les plus utilisés au monde, autant par le grand public (applications web principalement) que par des professionnels, en concurrence avec Oracle et Microsoft SQL Server



FIGURE 2.7 – Logo du MySql

- **JQuery**

jQuery est une bibliothèque JavaScript libre développée initialement par John Régis et qui est aujourd’hui maintenue et mise à jour par la communauté jQuery Team. Le Framework JavaScript jQuery code rapidement et simplement des traitements à base de code JavaScript pour dynamiser et améliorer l’ergonomie des sites internet.

- **Yarn**

Yarn est un gestionnaire de paquets pour un code. Il permet d’utiliser et de partager du code avec d’autres développeurs du monde entier. Le fil fait cela rapidement, en toute sécurité et de manière fiable afin que vous n’ayez jamais à vous inquiéter. Le code est partagé via ce qu’on appelle un package . Un package contient tout le code partagé ainsi qu’un package.jsonfichier (appelé manifest) qui décrit le package.



FIGURE 2.8 – Logo du Yarn

- **Composer**

Composer est un outil de gestion des dépendances en PHP. Il vous permet de déclarer les bibliothèques dont dépend votre projet et il les gérera (installer/mettre à jour) pour vous.

- Vous permet de déclarer les bibliothèques dont vous dépendez.
- quelles versions des packages peuvent et doivent être installés et les installe (c'est-à-dire qu'il les télécharge dans votre projet).
- Vous pouvez mettre à jour toutes vos dépendances en une seule commande.



FIGURE 2.9 – Logo du Composer

Microsoft Visual Studio

Visual Studio est un ensemble complet d'outils de développement permettant de générer des applications web ASP.NET, des services web XML, des applications bureautiques et des applications mobiles. Visual Basic, Visual C++, Visual C utilisent tous le même environnement de développement intégré (IDE), qui leur permet de partager des outils et facilite la création de solutions faisant appel à plusieurs langages. Par ailleurs, ces langages permettent de mieux tirer parti des fonctionnalités du framework .NET, qui fournit un accès à des technologies clés simplifiant le développement d'applications web ASP et de services web XML grâce à Visual Web Developer.



FIGURE 2.10 – Logo du Visual Studio

Overleaf

Overleaf est une plateforme en ligne gratuite permettant d'éditer du texte en LATEX en temps réel et compiler PDF. En outre, elle offre la possibilité de rédiger des documents de manière collaborative, de proposer ses documents directement à différents éditeurs (IEEE Journal, Springer, etc.) ou plateformes d'archives ouvertes (arXiv, engrxiv, etc.) pour une éventuelle publication. Cette plateforme est très compatible avec différents supports tels que tablettes et smartphones.



FIGURE 2.11 – Logo du Overleaf

2.3 Planification des Sprints

Dans la méthodologie SCRUM, la réunion de planification des sprints est une étape primordiale dans le processus. Durant cette réunion l'équipe va choisir la durée nécessaire pour chaque sprint selon la complexité de ce dernier mais aussi en prenant compte de l'effectif de ce projet. Dans notre cas, nous avons décidé de diviser nos sprints selon les traitements comme le montre la Table 2.2 :

sprint 1	sprint 2	sprint 3	sprint 4	sprint 5
- Authentification - Gestion des utilisateurs	- Gestion des Clients - Gestion des devises - Gestion des factures - Gestion des Produits	- Gestion des Paiements - Gestion des ordres de livraison - Gestion des Dépenses	- Gestion des Fournisseurs - Gestion des factures d'achat - Gestion des ordres de livraison d'achat	- Gestion de Tableau de bord - Gestion des paramétrages(Réglages) - Gestion des Rapports
15-03-21 28-03-21	29-03-21 18-04-21	19-04-21 05-05-21	06-05-21 19-05-21	19-05-21 6-06-21

TABLE 2.2 – Planification des Sprints

2.4 Conception statique

La figure 2.11 Représente le diagramme de classes de notre application

Chapitre 2 : Étude Préalable

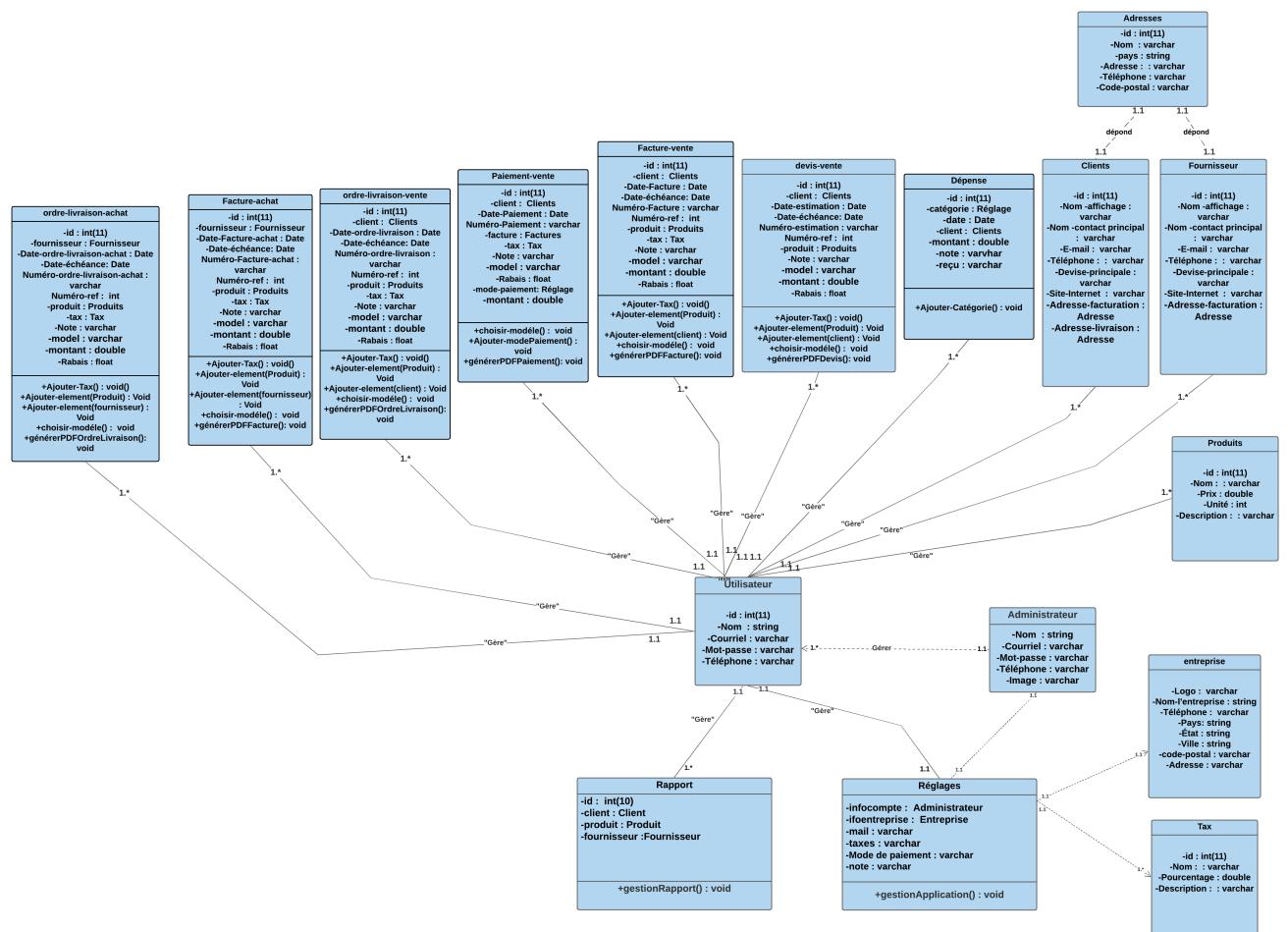


FIGURE 2.12 – Schéma diagramme de classe générale

Chapitre 2 : Étude Préalable

Notre application est construite principalement autour des classes suivantes :

- Classe Client : cette classe contient toutes les informations relatives au clients.
- Classe Produits : cette classe contient toutes les informations relatives au produit.
- Classe devis : cette classe contient toutes les informations relatives au devises.
- Classe Facture : cette classe contient toutes les informations relatives au Factures.
- Classe paiement : cette classe contient toutes les informations relatives au Paiements.
- Classe Dépenses : cette classe contient toutes les informations relatives au Dépenses.
- Classe ordre de livraison : cette classe contient toutes les informations relatives au ordre de livraisons.
- Classe entreprise : cette classe contient toutes les informations relatives au entreprise.
- Classe Utilisateur : cette classe contient toutes les informations relatives au utilisateurs.
- Adresses : cette classe contient toutes les informations relatives au adresses.
- Administrateur : cette classe contient toutes les informations relatives au administrateur.
- Classe Fournisseur : cette classe contient toutes les informations relatives au Fournisseurs.
- Classe Facture d'achat : cette classe contient toutes les informations relatives au Factures d'achat.
- Classe Rapport : cette classe contient toutes les informations relatives au Rapports.
- Classe Réglage : cette classe contient toutes les informations relatives au Réglage.

2.5 Chronogramme

Ce travail été réalisé durant une période de 4 mois. La répartition des tâches durant toute la durée de la réalisation de notre application est illustrée par le diagramme de Gantt réel de figure 2.12 :

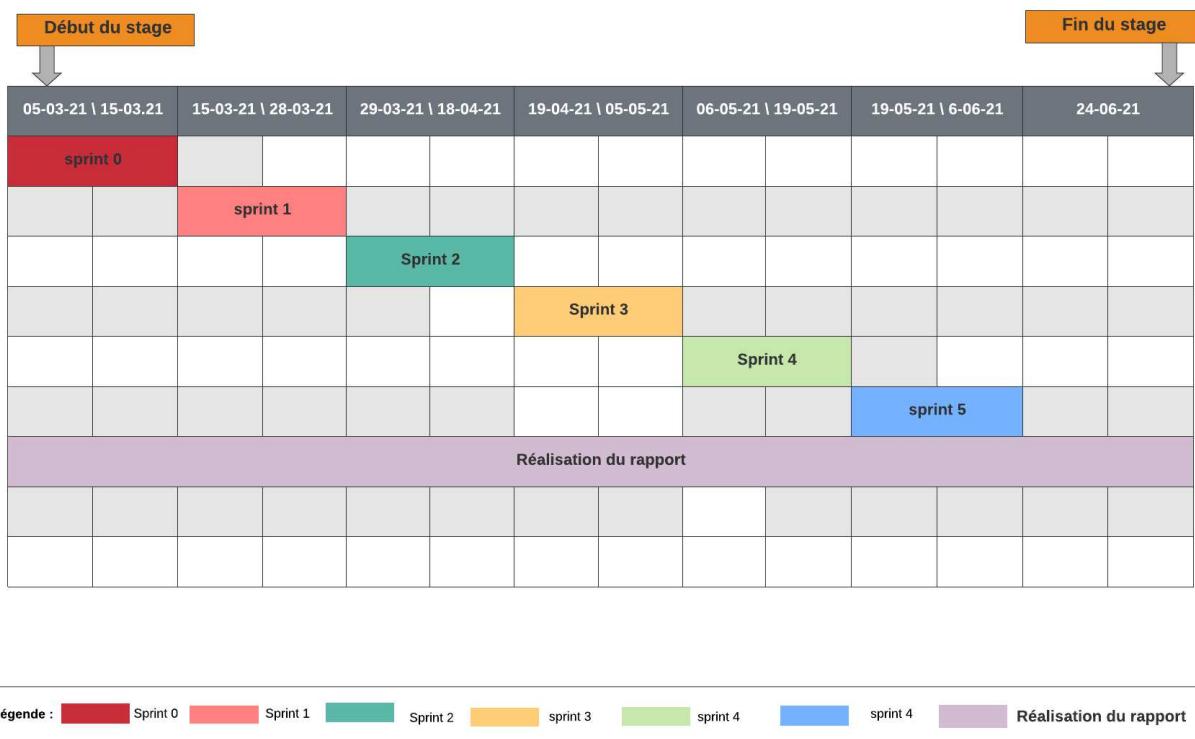


FIGURE 2.13 – Diagramme de Gantt réel

2.6 Conclusion

Dans ce chapitre, on a présenter l'architecture de notre système ainsi que l'environnement matériel et logiciel nécessaire pour notre projet. on a ensuite abordé la conception générale, ensuit, on a détaillé les différents sprints qui compose notre projet. On évoque dans la conclusion générale un résumé du travail effectué et on présente les sprint du projet.

3

Sprint 1 « Back-office »

Sommaire

Introduction	47
3.1 Backlog du Sprint	47
3.2 Spécification	48
3.2.1 Cas d'utilisation « S'authentifier»	49
3.2.2 Cas d'utilisation «Gérer les utilisateurs»	50
3.3 Conception	53
3.3.1 Diagramme de classes participantes	53
3.3.2 Diagrammes de séquences	53
3.4 Réalisation et tests	57
3.4.1 Authentification	57
3.4.2 Ajout d'un utilisateur	57
3.4.3 Modification d'un utilisateur	58
3.4.4 Suppression d'un utilisateur	59
3.5 Conclusion	59

Introduction

Après l'élaboration du backlog général de notre projet, nous allons passer au premier sprint. Ce dernier est dédié au module de gestion. Au niveau de cette partie, nous spécifions le sprint à l'aide des diagrammes de cas d'utilisation et la conception de celui ci se fera à l'aide des représentations statiques et dynamiques. Enfin, nous exposons la réalisation de ce sprint.

3.1 Backlog du Sprint

Le backlog de ce sprint contient la liste des fonctionnalités à réaliser :

ID	Fonctionnalité	User Story	Priorité
1	Authentification	En tant qu'utilisateur, je veux m'authentifier Afin d'accéder à mon interface.	Élevée
		En tant qu'administrateur, je veux m'authentifier. Afin d'accéder à mon interface	Élevée
2	Gestion des utilisateurs	En tant qu'administrateur, je veux ajouter un ou plusieurs utilisateurs.	Élevée
		En tant qu'administrateur, je veux modifier un utilisateur bien déterminée.	Élevée
		En tant qu'administrateur, je veux supprimer un utilisateur bien déterminée.	Élevée
		En tant qu'administrateur, je veux consulter la liste des utilisateurs.	Moyenne
		En tant qu'administrateur je veux réaliser toutes les tâches de l'utilisateur.	Élevée

TABLE 3.1 – Backlog du premier Sprint

3.2 Spécification

Dans cette section nous présentons le raffinement du diagramme de cas d'utilisation pour chacune des fonctionnalités retenues dans ce sprint. Ce premier livrable permet aux utilisateurs et à l'administrateur de D invoice de s'authentifier pour accéder chacun à son profil.

3.2.1 Cas d'utilisation « S'authentifier»

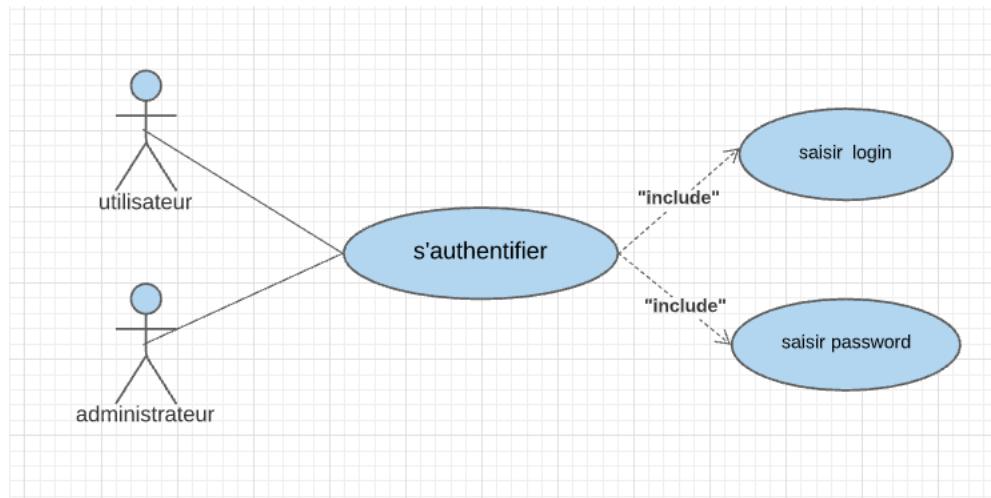


FIGURE 3.1 – Diagramme de raffinement du cas d'utilisation « S'authentifier»

La table 3.2 présente le raffinement du cas d'authentification où l'administrateur et les différents utilisateurs quelque soit le profil, doivent saisir leurs identifiant et leurs mots de passe à fin d'accéder chacun à sa session. Pour mieux connaître les étapes à suivre par l'utilisateur, une description textuelle présentée par le

Titre	S'authentifier
Objectif	L'utilisateur où l'administrateur doivent s'authentifier pour accéder à leur session.
Acteur	Les acteurs principaux : Tout utilisateur quelque soit le profil et l'administrateur
Pré-conditions	L'application doit être ouverte.
Déclencheur	L'utilisateur où l'administrateur souhaitent accéder à leur session.
Le scénario nominal	1- Le système affiche le formulaire d'authentification. 2- L'utilisateur où l'administrateur saisissent leur login et leur mot de passe. 3- Le système vérifie les informations saisies. 4- Le système ouvre la session correspondante.
Le scénario alternatif	3.a- Identifiant et/ou mot de passe invalide. 3.b- Le système affiche un message d'erreur.
Post-conditions	Ouverture de la session.

TABLE 3.2 – Description textuelle du cas d'utilisation «S'authentifier»

3.2.2 Cas d'utilisation «Gérer les utilisateurs»

La figure 3.2 nous donne une vision globale du comportement fonctionnel de gestion utilisateurs :

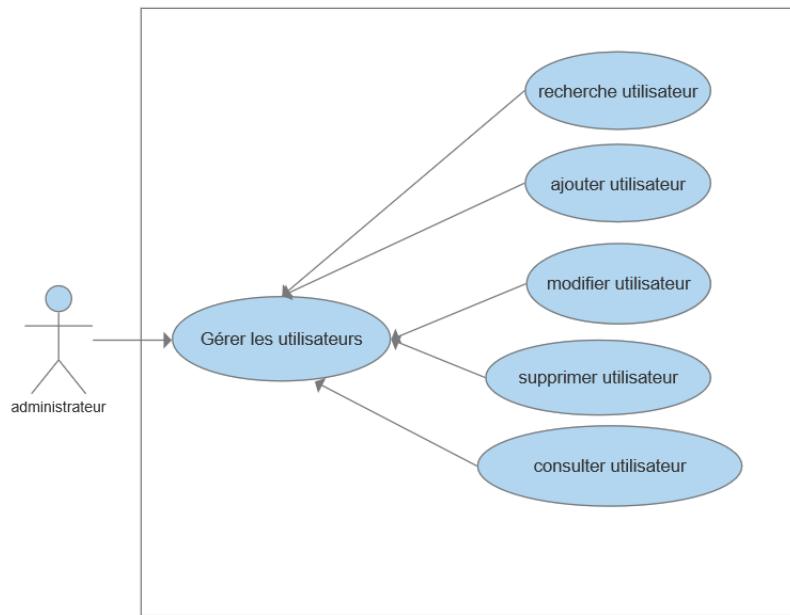


FIGURE 3.2 – Diagramme de raffinement du cas d'utilisation «Gérer utilisateurs»

Chapitre 3 : sprint 1 « Back-office »

Parmi les fonctionnalités autorisées à un administrateur après son authentification est la gestion des utilisateurs. Voici la description textuelle de ce cas d'utilisation qui résume tout :

Titre	Ajouter utilisateur
Objectif	L'administrateur peut créer un nouveau utilisateur.
Acteur	Administrateur profil et l'administrateur
Pré-conditions	-Authentification correcte à l'application. -L'administrateur demandé la page «Gérer utilisateurs»
Déclencheur	L'administrateur a cliqué sur «ajouter»
Le scénario nominal	<ol style="list-style-type: none">1- Le système affiche le formulaire d'ajout.2- L'administrateur remplit le formulaire avec les informations de l'utilisateur à créer.3- l'administrateur valide l'ajout de l'utilisateur.4- Le système vérifie la présence des champs obligatoires.5- Le système vérifie l'unicité du login.6- Le système ajoute l'utilisateur et affiche un message de succès.
Le scénario alternatif	<ol style="list-style-type: none">4.1- Le système renvoie un message d'erreur et signale à l'administrateur que tous les champs obligatoires n'ont pas été remplis.5.1- Si login existe déjà, le système affiche un message d'erreur.
Post-conditions	Un nouveau utilisateur est ajouté.

TABLE 3.3 – Description textuelle du CU «Ajouter un utilisateur»

Chapitre 3 : sprint 1 « Back-office »

Nous abordons maintenant le cas d'utilisation modifier utilisateur :

Titre	Modifier utilisateur
Objectif	L'administrateur peut modifier un utilisateur
Acteur	Administrateur
Pré-conditions	-Authentification correcte à l'application. -L'administrateur demandé la page «Gérer utilisateurs»
Déclencheur	L'utilisateur a cliqué sur «modifier»
Le scénario nominal	1- Le système affiche le formulaire de modification. 2- L'acteur fait les modifications désirées. 3- L'acteur valide la modification. 4- Le système vérifie la présence des champs obligatoires. 5- Le système enregistre les modifications et affiche un message de succès.
Le scénario alternatif	4.1- Le système renvoie un message d'erreur et signale à l'administrateur que tous les champs obligatoires n'ont pas été remplis.
Post-conditions	Utilisateur modifié.

TABLE 3.4 – Description textuelle du CU «Modifier utilisateur»

3.3 Conception

Cette partie définit les éléments résultant de l'analyse des spécifications du premier sprint. Afin d'en faciliter la réalisation, cette phase de conception sera définie comme la description du sprint à développer.

3.3.1 Diagramme de classes participantes

Le diagramme de classes décrit les associations entre les classes et ceci afin de déterminer les dépendances entre les différentes classes de ce sprint.

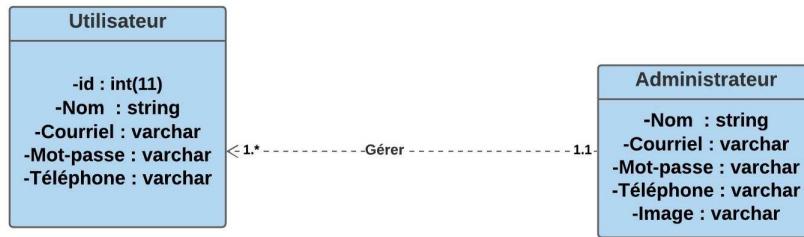


FIGURE 3.3 – Diagramme de classes du premier sprint

3.3.2 Diagrammes de séquences

Grâce à ces diagrammes, nous pouvons décrire comment les éléments, qui composent le premier sprint, interagissent entre eux et avec les acteurs. Commençons par la fonctionnalité d'ajout d'un nouveau utilisateur.

Diagramme de séquences du CU «Ajouter utilisateur»

Le diagramme de séquence illustré par la figure 3.4 explique les interactions entre l'administrateur et le système, dans le cadre d'un scénario, pour ajouter un utilisateur.

Chapitre 3 : sprint 1 « Back-office »

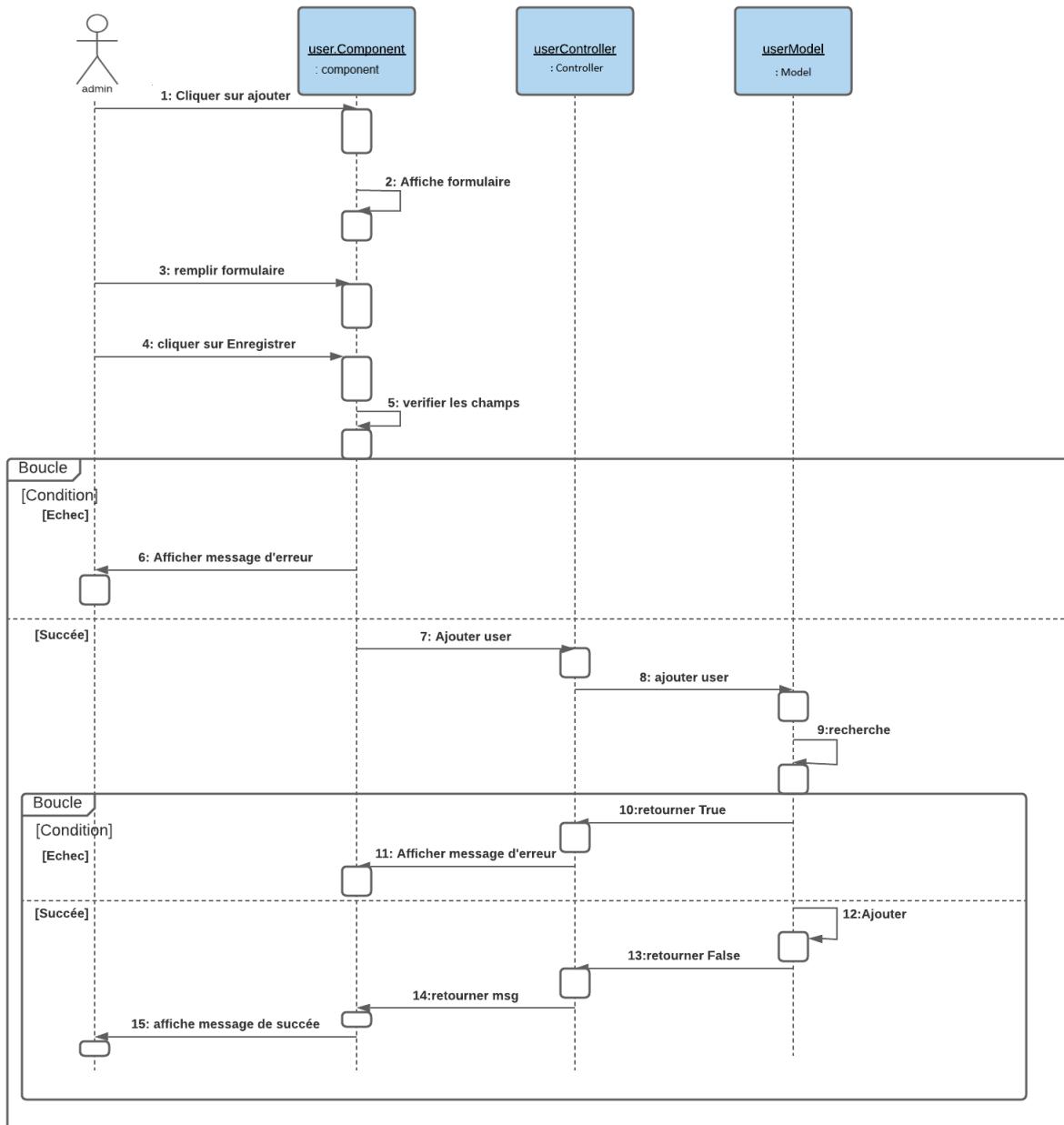


FIGURE 3.4 – Diagramme de séquences Ajouter utilisateur

Diagramme de séquences du CU «Modifier utilisateur»

Le diagramme 3.5 clarifie les interactions entre l'acteur et les éléments du système pour modifier les informations d'un utilisateur

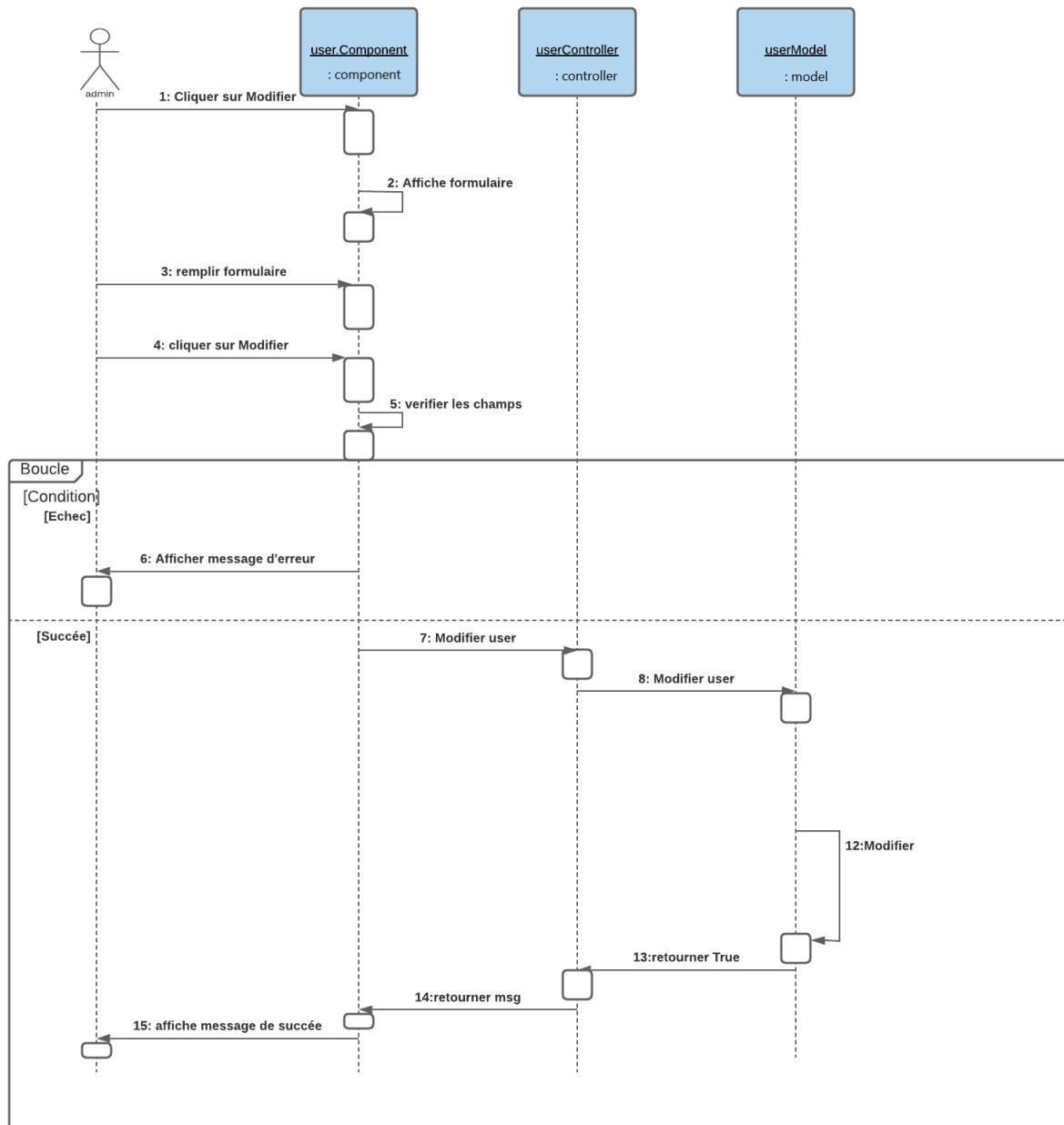


FIGURE 3.5 – Diagramme de séquences Modifier utilisateur

Diagramme de séquences du CU «Supprimer utilisateur»

Le diagramme 3.6 clarifie les interactions entre l'acteur et les éléments du système pour supprimer les informations d'un utilisateur

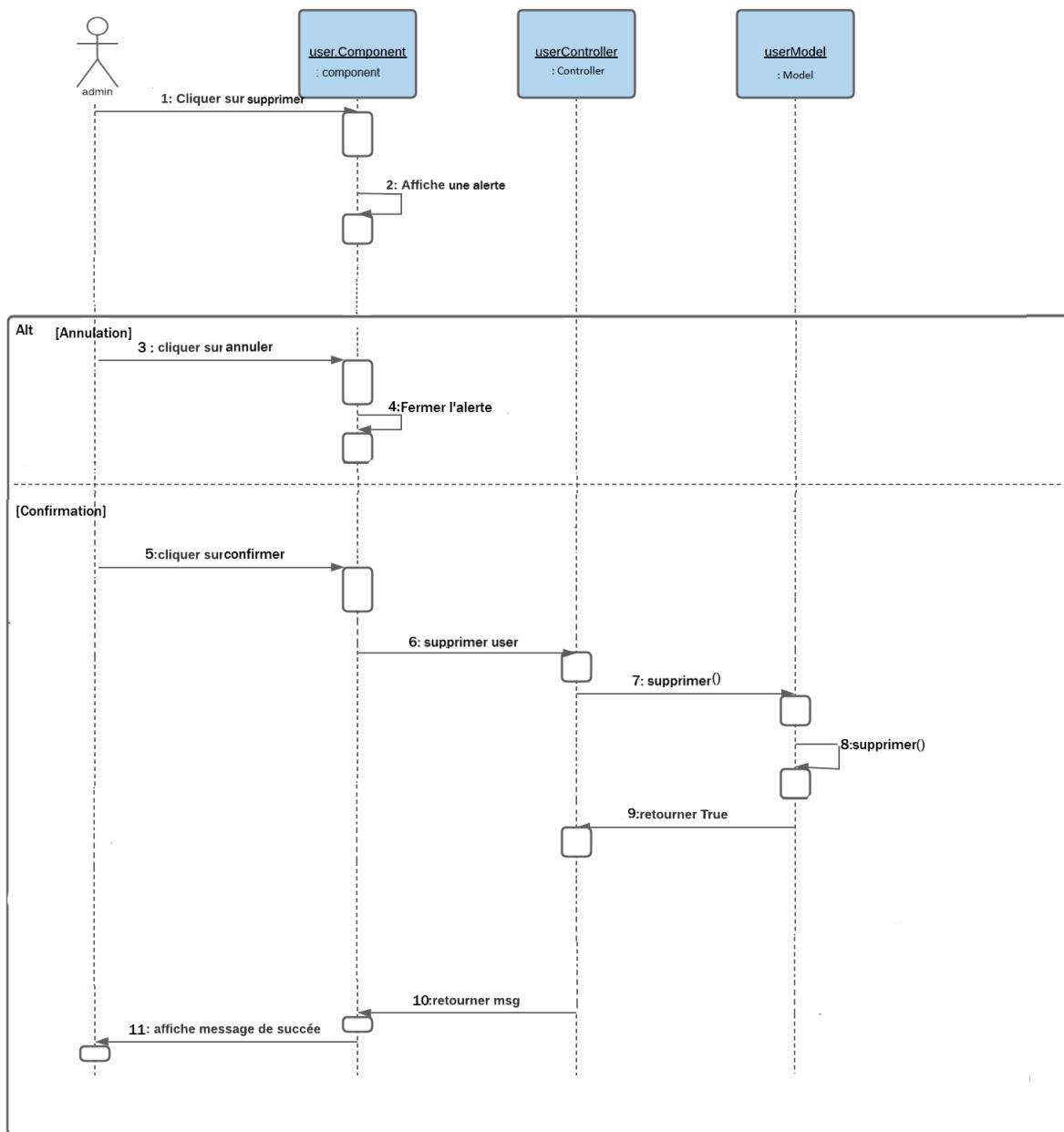


FIGURE 3.6 – Diagramme de séquences Supprimer utilisateur

3.4 Réalisation et tests

Passons maintenant à la présentation de la phase de réalisation du premier sprint. Nous exposons quelques interfaces réalisées durant ce sprint. Chaque interface illustre une fonctionnalité donnée.

3.4.1 Authentification

A fin d'accéder à l'application, l'utilisateur ou l'administrateur doit s'authentifier en saisissant son login et son mot de passe, le système vérifie les données saisies et affiche un message d'erreur en cas d'échec.

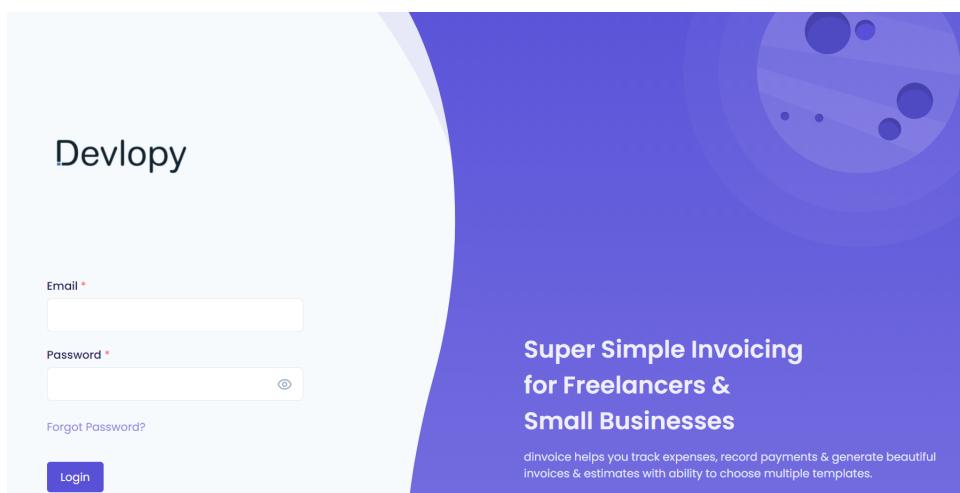


FIGURE 3.7 – Interface d'authentification

3.4.2 Ajout d'un utilisateur

La figure 3.8 montre un l'interface de gestion des utilisateurs.

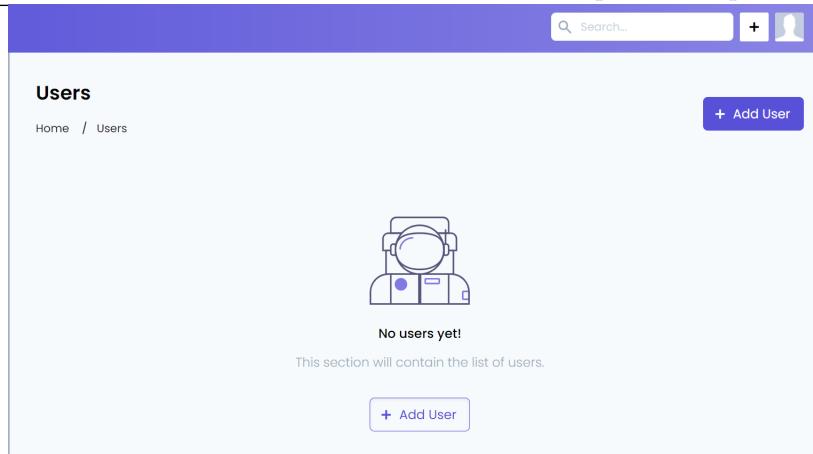


FIGURE 3.8 – Interface de gestion des utilisateurs

La figure 3.9 montre un formulaire d'ajout d'un utilisateur

The screenshot shows a 'New User' addition form within a web application. The header bar is purple with a search bar, a '+' icon, and a user profile icon. The breadcrumb navigation shows 'Home / Users / New User'. The form itself has four input fields: 'Name *' (with a red asterisk), 'Email *' (with a red asterisk), 'Password *' (with a red asterisk), and 'Phone'. Below these fields is a blue button labeled 'Save User' with a save icon.

FIGURE 3.9 – Interface d'ajout d'un utilisateur

3.4.3 Modification d'un utilisateur

L'administrateur peut modifier les données relatives à un utilisateur donné et c'est en cliquant sur le bouton de edit qu'une page s'affiche contenant toutes les informations nécessaires. La figure 3.10 met l'accent sur cette page.

The screenshot shows a user editing interface. At the top, there's a navigation bar with 'Home / Users / Edit User'. Below it is a form with the following fields:

- Name ***: yassin
- Email ***: yasin@gmail.com
- Password ***: (empty field)
- Phone**: 20150527

At the bottom of the form is a blue 'Update User' button.

FIGURE 3.10 – Interface de modification d'un utilisateur

3.4.4 Suppression d'un utilisateur

L'interface présentée dans la figure 3.11 montre une alerte qui va être affichée lors de la suppression d'un profil pour confirmer l'action.

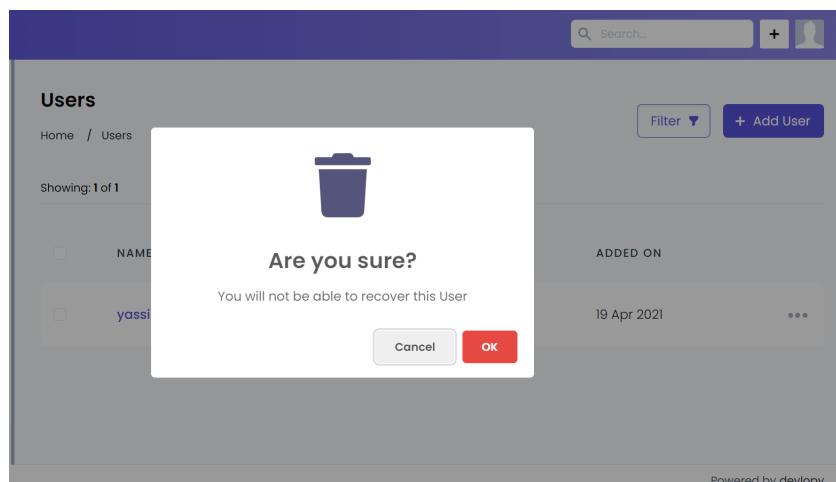


FIGURE 3.11 – Interface de suppression d'un utilisateur

3.5 Conclusion

Ce chapitre a présenté la spécification des fonctionnalités du premier sprint à l'aide des diagrammes de cas d'utilisation. Ensuite, il a donné l'aspect conceptuel en listant les résultats attendus en termes de fonctionnalités. Ces dernières ont été illustrées par quelques interfaces réalisées. Le chapitre suivant fera l'objet d'une étude détaillée du deuxième sprint.

4

Sprint 2

Sommaire

Introduction	61
4.1 Backlog du sprint	61
4.2 Spécification	61
4.2.1 Cas d'utilisation «Gestion des Clients»	61
4.2.2 Cas d'utilisation «Gestion des Produits»	64
4.2.3 Cas d'utilisation «Gestion des devises»	65
4.2.4 Cas d'utilisation «Gestion des Factures»	68
4.3 Conception	71
4.3.1 Diagramme de classes participantes	72
4.3.2 Diagrammes de séquences	72
4.4 Réalisation et tests	77
4.4.1 Gestion client	77
4.4.2 Gestion des produits	78
4.4.3 Gestion des Devises	79
4.4.4 Gestion des Factures	81
4.5 Conclusion	82

Introduction

Après avoir achevé le premier sprint, nous passons à la réalisation du deuxième sprint. Ce chapitre sera consacré en premier lieu à la présentation de son backlog qui sera spécifié à l'aide des diagrammes des cas d'utilisation. En second lieu, nous concevrons ce sprint avec un diagramme de classes, suivi des diagrammes de séquences. Pour finir, nous présentons quelques interfaces graphiques qui correspondent à des fonctionnalités importantes offertes par ce sprint

4.1 Backlog du sprint

D'après le backlog du produit défini dans le premier chapitre, le deuxième sprint est composé des cas d'utilisation suivants qui doivent être réalisés et livrés au client au termes de celui ci. Ce sprint contient la réalisation des fonctionnalités suivantes :

4.2 Spécification

Selon la planification établie, le deuxième sprint porte principalement sur les fonctionnalités, déjà élaborées dans la partie exigences fonctionnelles, qui seront raffinées à l'aide des diagrammes de cas d'utilisation.

4.2.1 Cas d'utilisation «Gestion des Clients»

la Gestion des Clients est une fonctionnalité offerte non seulement à l'admin mais aussi à tout les utilisateurs, la figure 4.1 nous donne une vision globale sur cette fonctionnalité :

Nous abordons le déroulement d'une modification d'un client :

ID	Fonctionnalité	User Story	Priorité
1	Gestion des Clients	En tant qu'utilisateur, je veux ajouter un ou plusieurs clients.	Élevée
		En tant qu'utilisateur, je veux modifier un client bien déterminée.	Élevée
		En tant qu'administrateur, je veux consulter la liste des utilisateurs.	Élevée
		En tant qu'administrateur je veux réaliser toutes les taches de l'utilisateur.	Moyenne
2	Gestion des Produits	En tant qu'utilisateur, je veux ajouter un ou plusieurs produits	Élevée
		En tant qu'utilisateur, je veux modifier un produit bien déterminée.	Élevée
		En tant qu'utilisateur, je veux supprimer un produit bien déterminée.	Élevée
		En tant qu'utilisateur, je veux consulter la liste des produits.	Moyenne
3	Gestion des devises	En tant qu'utilisateur, je veux ajouter un ou plusieurs devises.	Élevée
		En tant qu'utilisateur, je veux modifier un devis bien déterminée.	Élevée
		En tant qu'utilisateur, je veux supprimer un devis bien déterminée.	Élevée
		En tant qu'utilisateur, je veux consulter la liste des devises.	Moyenne
4	Gestion des factures	En tant qu'utilisateur, je veux ajouter une ou plusieurs factures.	Élevée
		En tant qu'utilisateur, je veux modifier une facture bien déterminée.	Élevée
		En tant qu'utilisateur, je veux supprimer une facture bien déterminée.	Élevée
		En tant qu'utilisateur, je veux consulter la liste des factures	Moyenne

TABLE 4.1 – Backlog du deuxième sprint
62

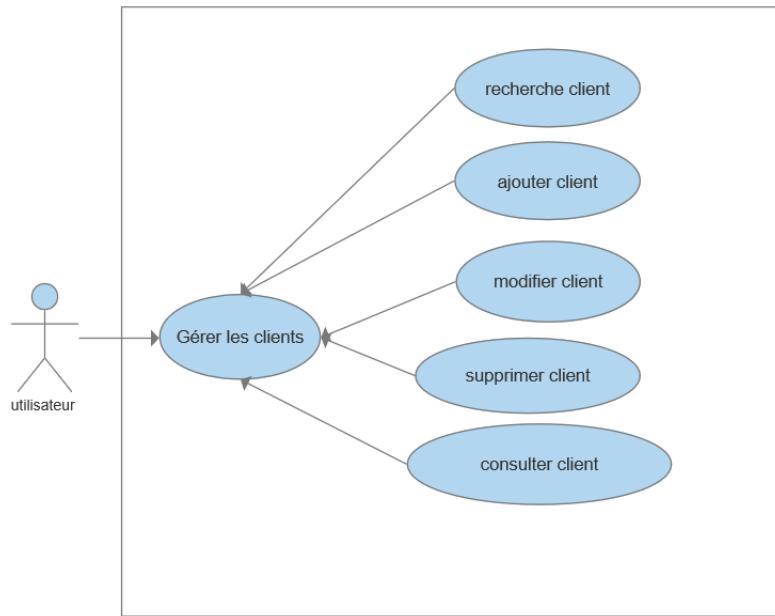


FIGURE 4.1 – Diagramme de raffinement du cas d'utilisation «Gestion des Clients»

Titre	Modifier client
Objectif	L'utilisateur peut modifier un client
Acteur	Administrateur et l'utilisateur
Pré-conditions	-Authentification correcte à l'application. - Consultation de la liste des clients.
Déclencheur	L'utilisateur a cliqué sur «modifier»
Le scénario nominal	1- Le système affiche la page de modification. 2- L'acteur fait les modifications désirées. 3- L'acteur valide la modification. 4- Le système vérifie la présence des champs obligatoires. 5- Le système enregistre les modifications et affiche un message de succès.
Le scénario alternatif	4.1- Le système renvoie un message d'erreur et signale à l'utilisateur que tous les champs obligatoires n'ont pas été remplis.
Post-conditions	client modifié.

TABLE 4.2 – Description textuelle du CU «Modifier client»

4.2.2 Cas d'utilisation «Gestion des Produits»

la Gestion des Produits est une fonctionnalité offerte non seulement à l'admin mais aussi à tout les utilisateurs, la figure 4.2 nous donne une vision globale sur cette fonctionnalité :

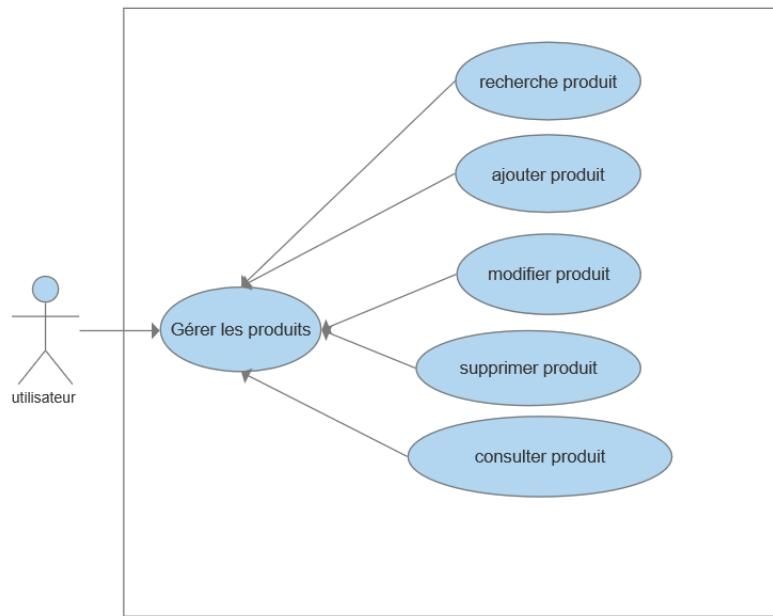


FIGURE 4.2 – Diagramme de raffinement du cas d'utilisation «Gestion des Produit»

Nous abordons en premier lieu le déroulement d'ajout d'un produit :

Chapitre 4 : sprint 2

Titre	Ajouter produit
Objectif	L'utilisateur peut créer un nouveau produit.
Acteur	Utilisateur et Administrateur
Pré-conditions	-Authentification correcte à l'application. -L'administrateur demandé la page «Gérer produits»
Déclencheur	L'utilisateur a cliqué sur «ajouter»
Le scénario nominal	1- Le système affiche une page d'ajout. 2- L'utilisateur remplit le formulaire avec les informations du produit à créer. 3- l'utilisateur valide l'ajout du produit. 4- Le système vérifie la présence des champs obligatoires. 5- Le système vérifie l'unicité du login. 6- Le système ajoute l'utilisateur et affiche un message de succès.
Le scénario alternatif	4.1- Le système renvoie un message d'erreur et signale à l'utilisateur que tous les champs obligatoires n'ont pas été remplis.
Post-conditions	Un nouveau produit est ajouté.

TABLE 4.3 – Description textuelle du CU «Ajouter un produit»

4.2.3 Cas d'utilisation «Gestion des devises»

La Gestion des devises est une fonctionnalité assez importante pour notre application. La figure 4.3 nous donne une idée sur le déroulement de cette fonctionnalité.

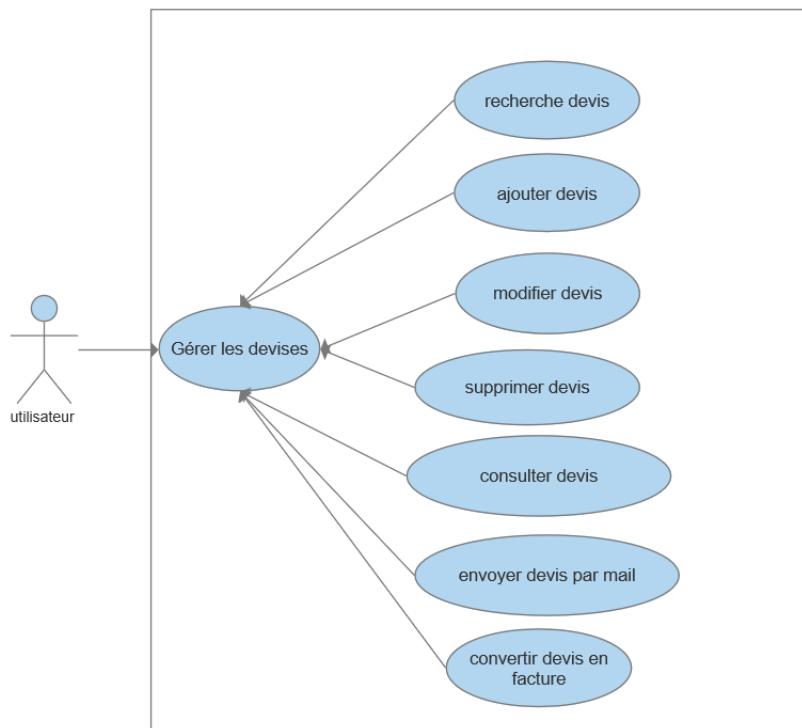


FIGURE 4.3 – Diagramme de raffinement du cas d'utilisation «Gestion des devises»

En outre, avec cette fonctionnalité on peut marquer une devis comme envoyée, acceptée ou rejetée.

Voilà le déroulement d'un envoi de devis par mail :

Chapitre 4 : sprint 2

Titre	envoyer devis par mail
Objectif	L'utilisateur peut envoyer devis par mail.
Acteur	Utilisateur et Administrateur
Pré-conditions	-Authentification correcte à l'application. -L'administrateur demandé la page «Gérer devises»
Déclencheur	L'utilisateur a cliqué sur «envoyer une devis»
Le scénario nominal	1- Le système affiche le formulaire d'envoi. 2- L'utilisateur remplit le formulaire avec les informations du devis à envoyer. 3- l'utilisateur valide l'envoi de devis par mail à un client bien déterminer. 4- Le système vérifie la présence des champs obligatoires. 5- Le système vérifie la validation d'adresse mail. 6- Le système envoi le mail et affiche un message de succès.
Le scénario alternatif	4.1- Le système renvoie un message d'erreur et signale à l'utilisateur que tous les champs obligatoires n'ont pas été remplis. 5.1- Si adresse mail n'est pas validée, le système affiche un message d'erreur.
Post-conditions	Un mail est envoyé avec succès.

TABLE 4.4 – Description textuelle du CU «envoyer de devis par mail»

Nous détaillons dans ce qui suit le cas d'utilisation «convertir devis en facture»

Chapitre 4 : sprint 2

Titre	convertir devis en facture
Objectif	L'utilisateur peut convertir une devis en facture.
Acteur	Utilisateur et Administrateur
Pré-conditions	-Authentification correcte à l'application. -L'administrateur demandé la page «Gérer devises»
Déclencheur	L'utilisateur a cliqué sur «convertir en facture»
Le scénario nominal	<p>1- Le système affiche une page d'ajout Facture qui est déjà rempli par les information de devis.</p> <p>2- L'utilisateur peut modifié le formulaire.</p> <p>3- l'utilisateur valide la conversion de devis en facture .</p> <p>4- Le système vérifie la présence des champs obligatoires.</p> <p>5- Le système vérifie l'unicité du login.</p> <p>6- Le système convertir devis et affiche un message de succès.</p>
Le scénario alternatif	4.1- Le système renvoie un message d'erreur et signale à l'utilisateur que tous les champs obligatoires n'ont pas été remplis.
Post-conditions	une nouvelle Facture est ajoutée.

TABLE 4.5 – Description textuelle du CU «convertir devis en facture»

4.2.4 Cas d'utilisation «Gestion des Factures»

La Gestion des Factures est l'une des fonctionnalités les plus importantes dans notre application. La figure 4.4 illustre ce cas d'utilisation.

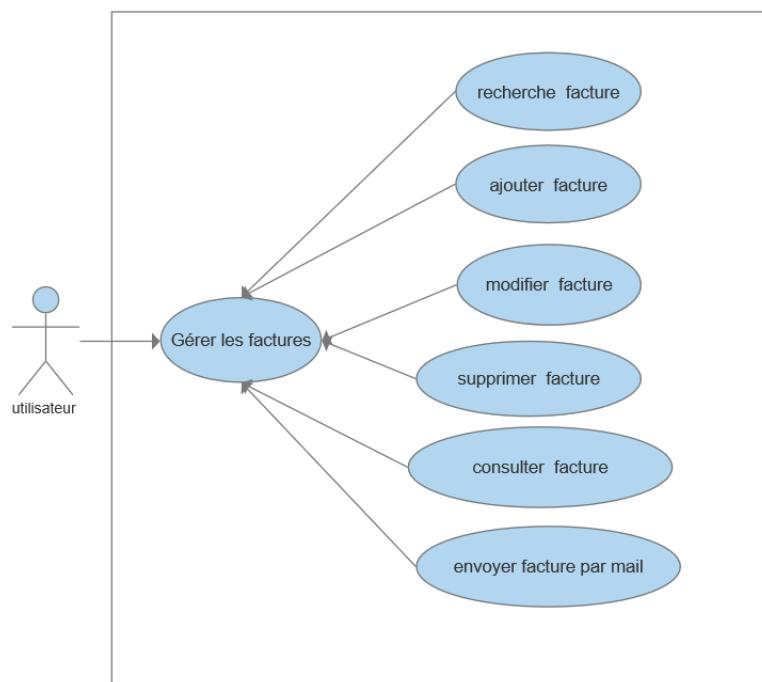


FIGURE 4.4 – Diagramme de raffinement du cas d'utilisation «Gestion des factures»

En décrivant le cas d'utilisation «recherche Facture », nous pouvons mieux comprendre l'enchaînement de cette fonctionnalité.

Chapitre 4 : sprint 2

Titre	Recherche facture
Objectif	Rechercher les factures en utilisant différents critères de recherche
Acteur	Utilisateur et Administrateur
Pré-conditions	-Authentification correcte à l'application. -L'administrateur demandé la page «Gérer factures»
Déclencheur	L'utilisateur a cliqué sur «Filtre»
Le scénario nominal	1- L'utilisateur choisit le critère de recherche et clique sur rechercher. 2- Le système affiche la liste des valeurs possibles pour ce critère 3- L'utilisateur choisit parmi la liste . 4- Le système affiche la liste correspondante .
Post-conditions	Un filtrage sur les données est appliqué

TABLE 4.6 – Description textuelle du CU «recherche facture»

la tableau 4.6 illustre la Description textuelle du CU «supprimer paiement»

Titre	Supprimer Facture
Objectif	L'acteur peut supprimer une facture.
Acteur	Utilisateur et Administrateur
Pré-conditions	-Authentification correcte à l'application. -L'administrateur demandé la page «Gérer factures»
Déclencheur	L'utilisateur a cliqué sur «Effacer»
Le scénario nominal	1- Une alerte de confirmation est affichée. 2- L'acteur confirme la suppression. 3- Le système supprime le profil et affiche un message de succès.
Le scénario alternatif	2-1. L'acteur annule la suppression
Post-conditions	Une facture est supprimée.

TABLE 4.7 – Description textuelle du CU «Supprimer facture»

4.3 Conception

Cette partie définit les éléments résultant de l'analyse des spécifications du deuxième sprint. Afin d'en faciliter la réalisation, cette phase de conception sera définie comme la description du sprint à développer.

4.3.1 Diagramme de classes participantes

Le diagramme de classes décrit les associations entre les classes et ceci afin de déterminer les dépendances entre les différentes classes de ce sprint.

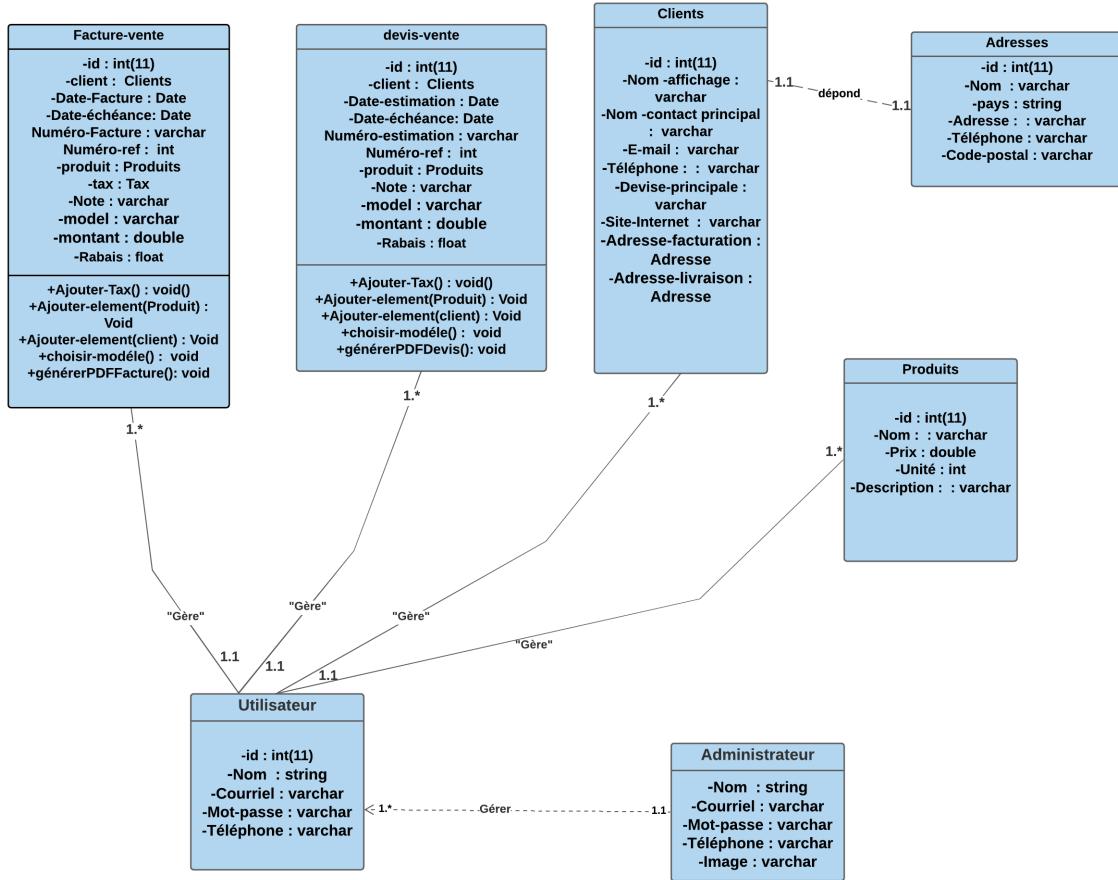


FIGURE 4.5 – Diagramme de classes du deuxième sprint

4.3.2 Diagrammes de séquences

Grâce à ces diagrammes, nous pouvons décrire comment les éléments, qui composent le deuxième sprint, interagissent entre eux et avec les acteurs.

Diagramme de séquences du CU «Modifier client»

La modification d'un client se fait comme la modification des autres objets dans le Back Office. Dans ce cas, pour modifier les informations d'un client, il faut d'abord accéder à la liste des clients, puis choisir le client à modifier. La figure 4.6 présente un diagramme de séquences qui exprime mieux cette fonctionnalité.

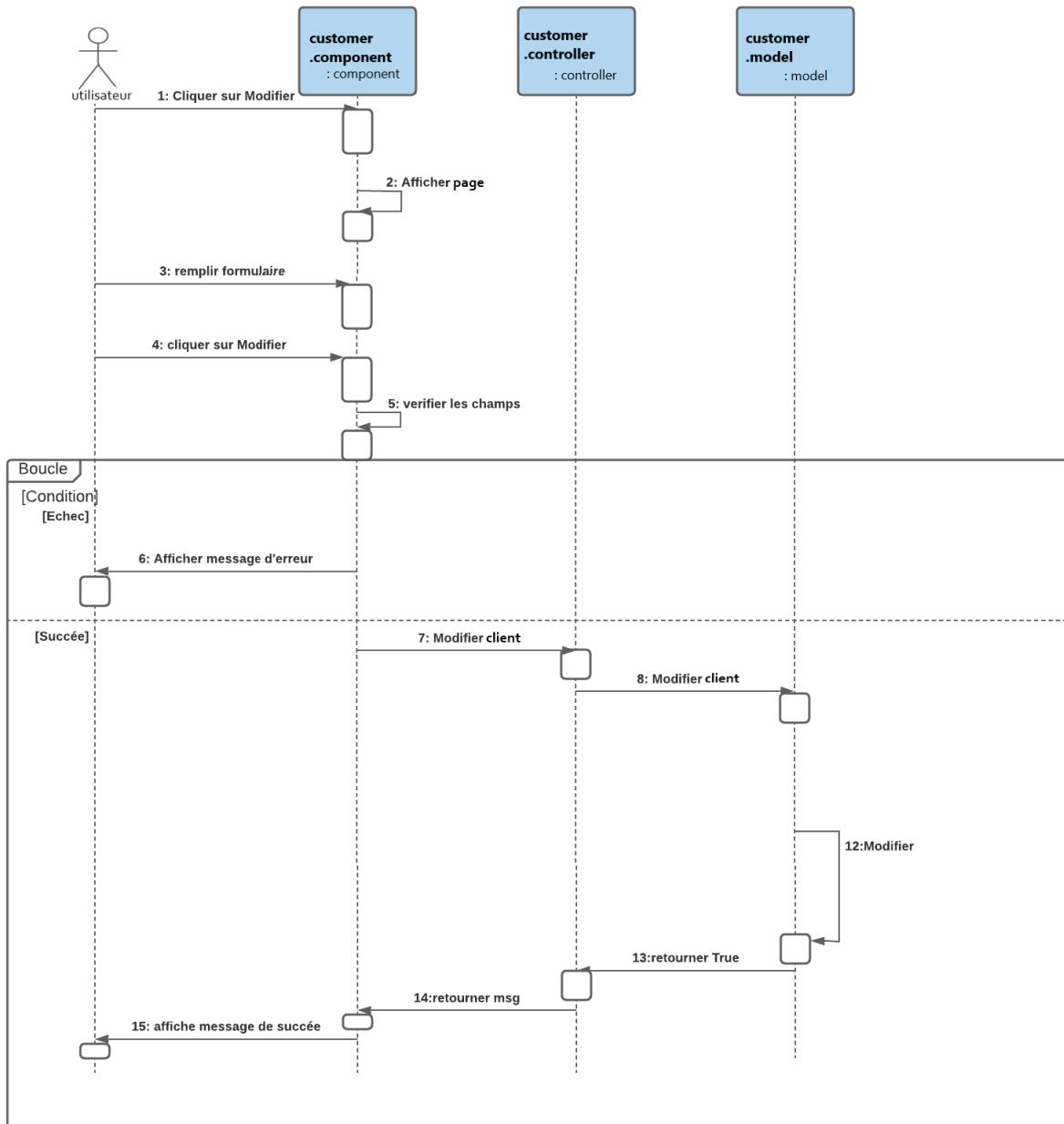


FIGURE 4.6 – Diagramme de séquences «Modifie client»

Diagramme de séquences du CU «Ajouter produit»

Le diagramme présenté dans la figure 4.7 montre l'enchaînement de ce cas d'utilisation.

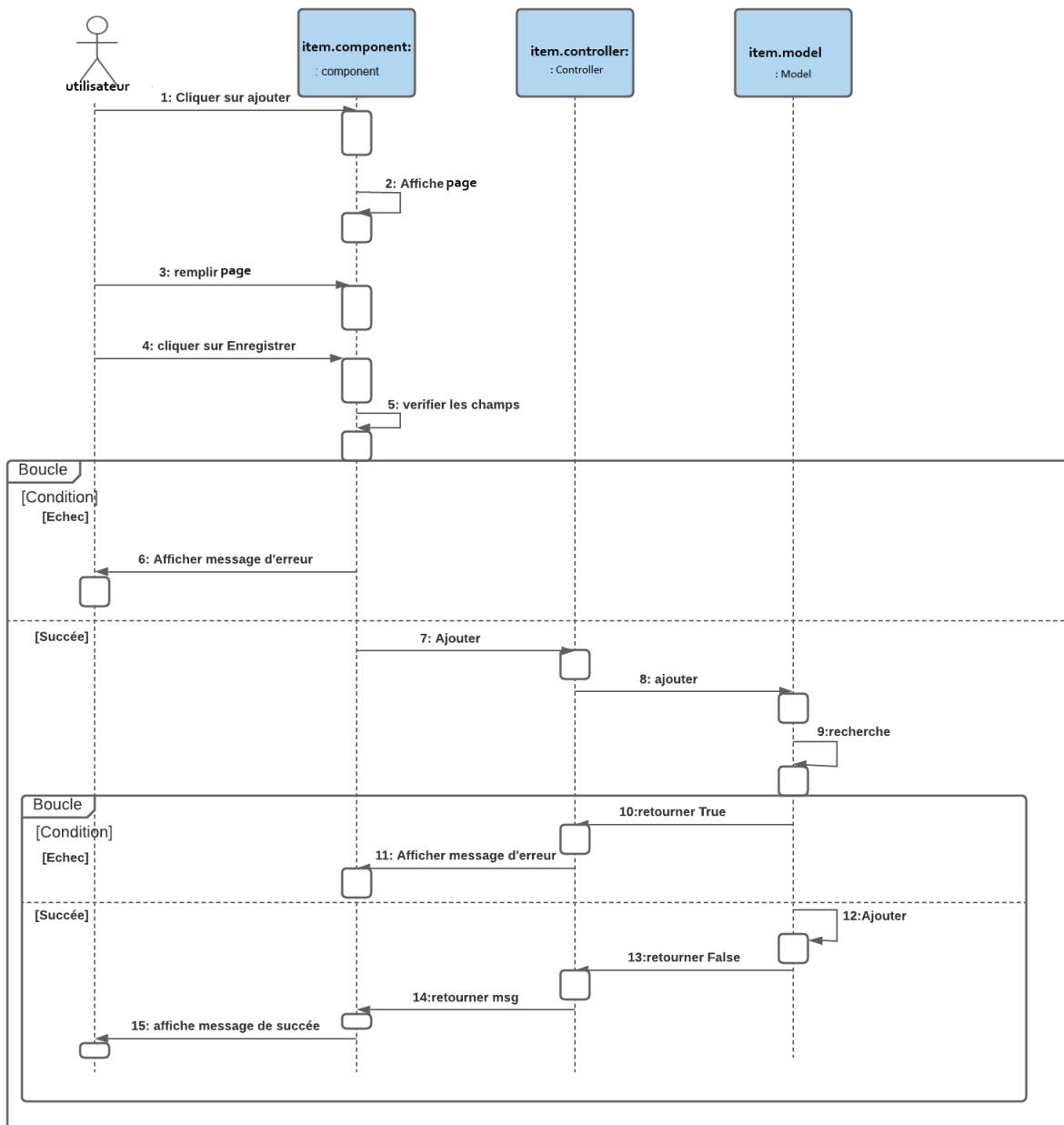


FIGURE 4.7 – Diagramme de séquences «Ajouter produit»

Diagramme de séquences du CU «Recherche Devis »

Le diagramme 4.8 présente les interactions entre l'utilisateur et le système pour réaliser cette fonction.

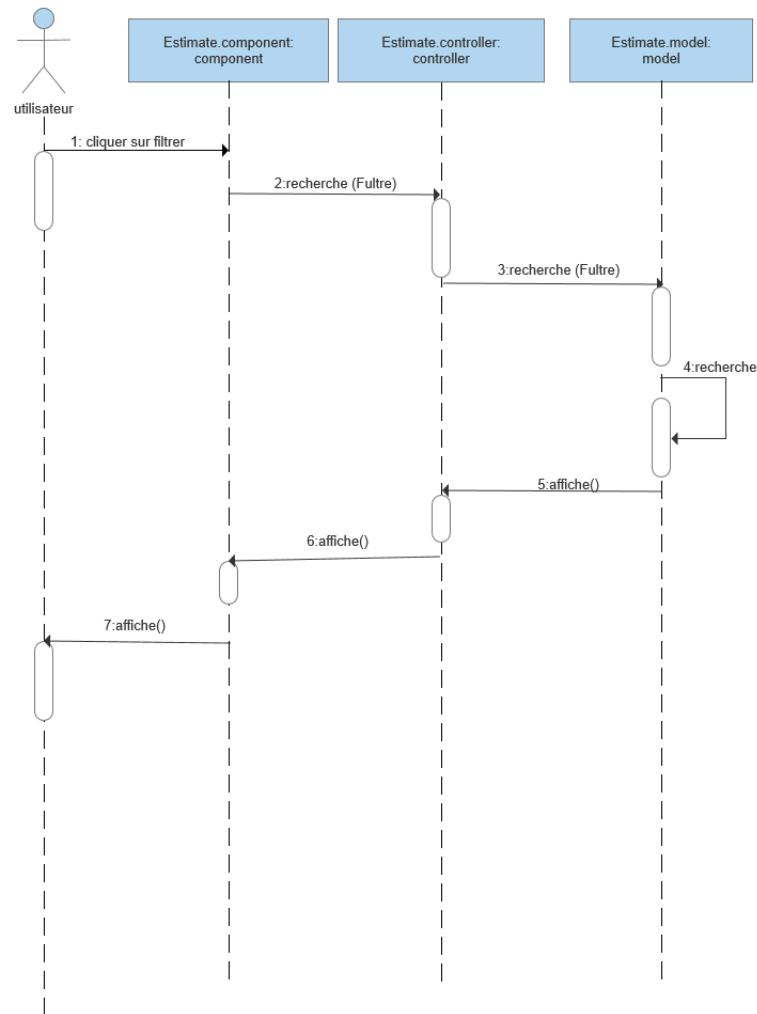


FIGURE 4.8 – Diagramme de séquences «Recherche Devis»

Diagramme de séquences du CU «Supprimer Facture»

Le diagramme 4.9 illustre les étapes de suppression d'une facture.

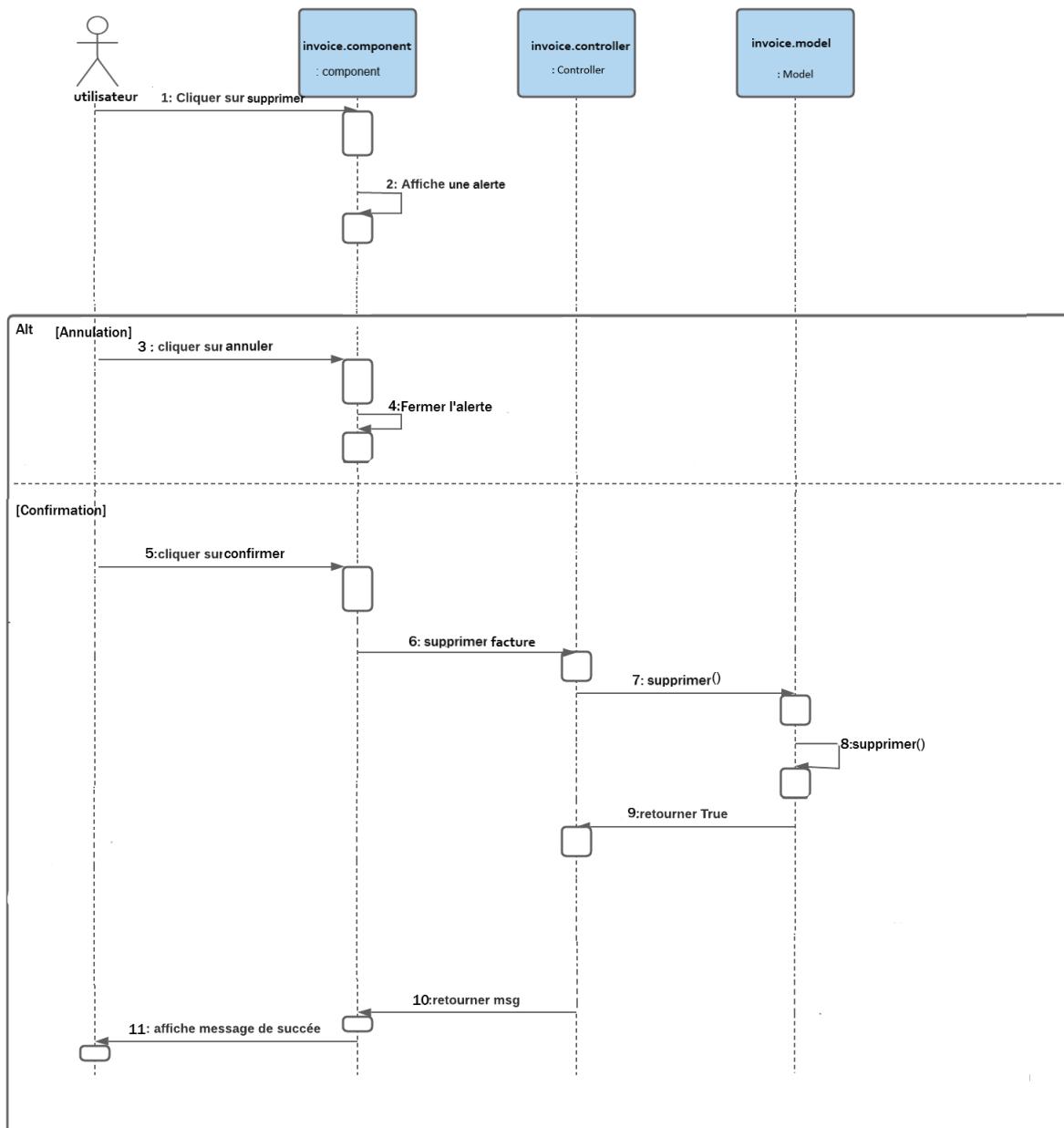


FIGURE 4.9 – Diagramme de séquences «Supprimer Facture»

4.4 Réalisation et tests

Passons maintenant à la présentation de la phase de réalisation du deuxième sprint.

4.4.1 Gestion client

1. Ajouter client

L'interface de la figure 4.10 montre la page qui sera affichée une fois que l'utilisateur va cliquer sur ajouter.

The screenshot shows the Devlopv software interface. The top navigation bar is purple with the title 'Devlopv'. On the right side of the top bar are search, add, and user icons. The left sidebar has a purple header 'Tableau de bord' and a list of menu items: 'Les clients' (selected), 'Articles', 'Estimations', 'Les factures', 'Commandes de livraison', 'Paiements', 'Dépenses', and 'Utilisateurs'. The main content area has a white header 'Nouveau client' with a breadcrumb trail: 'Accueil / Clients / Nouveau client'. On the right is a blue button labeled 'Sauver le client' (Save client). The central form is titled 'Informations de base' (Basic Information). It contains four input fields: 'Nom d'affichage *' (Display name), 'Nom du contact principal' (Primary contact name), 'E-mail' (Email), and 'Téléphoner' (Phone). Below these are two dropdown menus: 'Devise principale' (Main currency) set to 'USD - Dollar américain' and 'Site Internet' (Website).

FIGURE 4.10 – « ajouter client»

1. Recherche client

Pour effectuer la recherche dans l'interface «Clients», l'utilisateur peut se baser sur différents critère comme nom, nom de contact et numéro de téléphone.

AFFICHER UN NOM	NOM DU CONTACT	TÉLÉPHONER	MONTANT DÛ	AJOUTÉ LE
<input type="checkbox"/> mohamed	âge	20150527	\$ 1,891.80	19 avr 2021

FIGURE 4.11 – « Recherche client »

4.4.2 Gestion des produits

1. Modifier produit

L'interface 4.12 montre la page qui sera affichée une fois que l'utilisateur va cliquer sur Modifier.

Devlopay

Accueil / Objets / Modifier l'article

Nom *
NIKE Air Force 1 '07

Prix *
\$ 210.00

Unité
pc

Description
whit shoes

Mettre à jour l'élément

FIGURE 4.12 – «Modifier produit»

1. Supprimer produit

L’interface présentée dans la figure 4.13 montre une alerte qui va être affichée lors de la suppression d’un produit pour confirmer l’action.

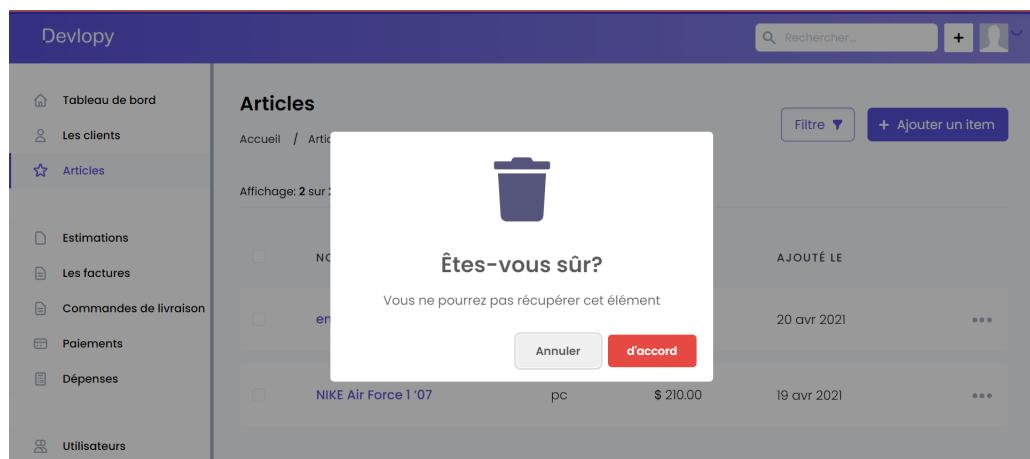


FIGURE 4.13 – «Supprimer produit»

4.4.3 Gestion des Devises

1. Consulter Devis

Chapitre 4 : sprint 2

L'interface de la figure 4.14 "consulter Devise", montre une liste de tous les devises, accompagnées d'une page qui inclut toutes les informations du devis sélectionnées où l'utilisateur peut télécharger cette page en PDF ou bien l'imprimer.

The screenshot shows the Devlopay application interface. On the left, there is a sidebar with various menu items: Tableau de bord, Les clients, Articles, Estimations (which is currently selected), Les factures, Commandes de livraison, Paiements, Dépenses, and Utilisateurs. The main area displays a list of estimates. One estimate is selected, showing details: Mohamed, EST-000003, \$ 210.00, 22 avr 2021, status BROUILLON. Below this, two other estimates are listed: Mohamed, EST-000002, \$ 210.00, 20 avr 2021, status REJETÉ; and Mohamed, EST-000001, \$ 420.00, 20 avr 2021, status ENVOYÉ. At the top right of the main area, there is a search bar, a plus icon, and a user profile icon. Below the list, a modal window titled 'EST-000003' is open, showing the detailed view of the selected estimate. The modal has tabs for 'Estimate' and 'Maram'. The 'Estimate' tab shows a preview of the estimate document. The 'Maram' tab shows a breakdown of the estimate items, including 'Maram' and 'Control Park' with their respective quantities, prices, and amounts.

FIGURE 4.14 – « Consulter Devise»

Il existe le bouton envoyer estimation qui a pour but d'envoyer la devis par mail au client, et un bouton "marquer comme envoyé".

1. envoyer devis par mail

La figure 4.15 montre le formulaire qui va être affichée pour envoyer une devis.

The screenshot shows a software interface for managing invoices and estimates. On the left, a sidebar menu lists 'Tableau de bord', 'Les clients', 'Articles', 'Estimations', 'Les factures', 'Commandes de livraison', 'Paiements', 'Dépenses', and 'Utilisateurs'. The main area is titled 'Nouvelle estimation' (New Estimate). It has fields for 'À partir de' (From) set to 'admin@dinvoice.in', 'À' (To) set to 'yasinl@gmail.com', 'Sujet' (Subject) set to 'New Estimate', and 'Corps' (Body) containing a rich text editor and a download link for a PDF estimate. At the bottom are 'Annuler' (Cancel) and 'Envoyer' (Send) buttons.

FIGURE 4.15 – «envoyer devis par mail»

4.4.4 Gestion des Factures

1. Ajouter Facture

Passons maintenant à l'interface ajouter Facture

The screenshot shows a software interface for managing invoices. On the left, a sidebar menu lists 'Tableau de bord', 'Les clients', 'Articles', 'Estimations', 'Les factures', 'Commandes de livraison', 'Paiements', 'Dépenses', and 'Utilisateurs'. The main area is titled 'Nouvelle facture' (New Invoice). It has fields for 'FACTURER' (Bill To) and 'ENVOYEZ À' (Ship To), both listing 'mohamed' with address details. It also has fields for 'Date de la facture' (Invoice Date) set to '24 Apr 2021', 'Due Date' set to '01 May 2021', 'Numéro de facture' (Invoice Number) set to '# INV - 000005', and 'Numéro de ref' (Reference Number) set to '#'. Below these are tables for adding items and calculating totals. At the bottom is a table showing an item: 'NIKE Air Force 1 '07' with 'Quantité' (Quantity) '1', 'Prix' (Price) '\$ 210.00', and 'Montant' (Amount) '\$ 210.00'. A 'Rechercher...' search bar and an 'Enregistrer la facture' (Save invoice) button are also visible.

FIGURE 4.16 – «Ajouter Facture 1»

La figure 4.17 montre le résultat

The screenshot shows the Devlopy application interface. The left sidebar contains navigation links: Tableau de bord, Les clients, Articles, Estimations, **Les factures**, Commandes de livraison, and Paiements. The main content area is titled "Les factures" and shows a list of invoices. The header of the list table includes columns: DATE, NUMÉRO, CLIENT, STATUT, STATUT PAYÉ, and MONTANT DÛ. A single invoice is listed:

DATE	NUMÉRO	CLIENT	STATUT	STATUT PAYÉ	MONTANT DÛ
22 avr 2021	INV-000005	mohamed	BROUILLON	NON PAYÉ	\$ 210.00

FIGURE 4.17 – «Résultat de l'ajout Facture»

4.5 Conclusion

Ce chapitre a présenté la spécification des fonctionnalités du deuxième sprint à l'aide des diagrammes de cas d'utilisation. Ensuite, il a donné l'aspect conceptuel en listant les résultats attendus en termes de fonctionnalités. Ces dernières ont été illustrées par quelques interfaces réalisées. Le chapitre suivant fera l'objet d'une étude détaillée du troisième sprint.

5

Sprint 3

Sommaire

Introduction	84
5.1 Backlog du sprint	84
5.2 Spécifications fonctionnelles	86
5.2.1 Raffinement de cas d'utilisation «Gestion des ordres de livraison»	86
5.2.2 Raffinement de cas d'utilisation «Gestion des paiements»	88
5.2.3 Raffinement de cas d'utilisation «Gestion des dépenses»	90
5.3 Conception	92
5.3.1 Diagramme de classes participantes	92
5.3.2 Diagrammes de séquences	93
5.4 Réalisation et tests	98
5.4.1 Gestion d'ordre de livraison	98
5.4.2 Gestion des paiements	99
5.4.3 Gestion des Dépenses	101
5.5 Conclusion	101

Introduction

Le but principal du troisième sprint est de présenter les trois cas d'utilisation : Ordres de Livraison, paiements et Dépenses. Ceci va être élaboré avec les mêmes démarches du sprint 2.

5.1 Backlog du sprint

Le tableau 5.1 montre notre troisième sprint backlog :

ID	Fonctionnalité	User Story	Priorité
1	Gestion des ordres de livraison	En tant qu'utilisateur, je veux ajouter une ou plusieurs ordres de livraison.	Élevée
		En tant qu'utilisateur, je veux modifier une ordre de livraison bien déterminée.	Élevée
		En tant qu'utilisateur, je veux supprimer une ordre de livraison bien déterminée.	Élevée
		En tant qu'utilisateur, je veux consulter la liste des ordres de livraison..	Moyenne
2	Gestion des Paiements	En tant qu'utilisateur, je veux ajouter une ou plusieurs paiements.	Élevée
		En tant qu'utilisateur, je veux modifier une paiement bien déterminée.	Élevée
		En tant qu'utilisateur, je veux supprimer une paiement bien déterminée.	Élevée
		En tant qu'utilisateur, je veux consulter la liste des paiements.	Moyenne
3	Gestion des Dépenses	En tant qu'utilisateur, je veux ajouter une ou plusieurs ordres de livraison.	Élevée
		En tant qu'utilisateur, je veux modifier une ordre de livraison bien déterminée.	Élevée
		En tant qu'utilisateur, je veux supprimer une ordre de livraison bien déterminée.	Élevée
		En tant qu'utilisateur, je veux consulter la liste des ordres de livraison.	Moyenne

TABLE 5.1 – Backlog du sprint 3

5.2 Spécifications fonctionnelles

Pour la spécification fonctionnelle de ce sprint, nous commençons par la présentation les diagrammes des cas d'utilisation raffinés d'UML ainsi que les scénarios de ces derniers.

5.2.1 Raffinement de cas d'utilisation «Gestion des ordres de livraison»

la figure 5.1 illustre le diagramme du cas d'utilisation «Gestion des ordres de livraison» :

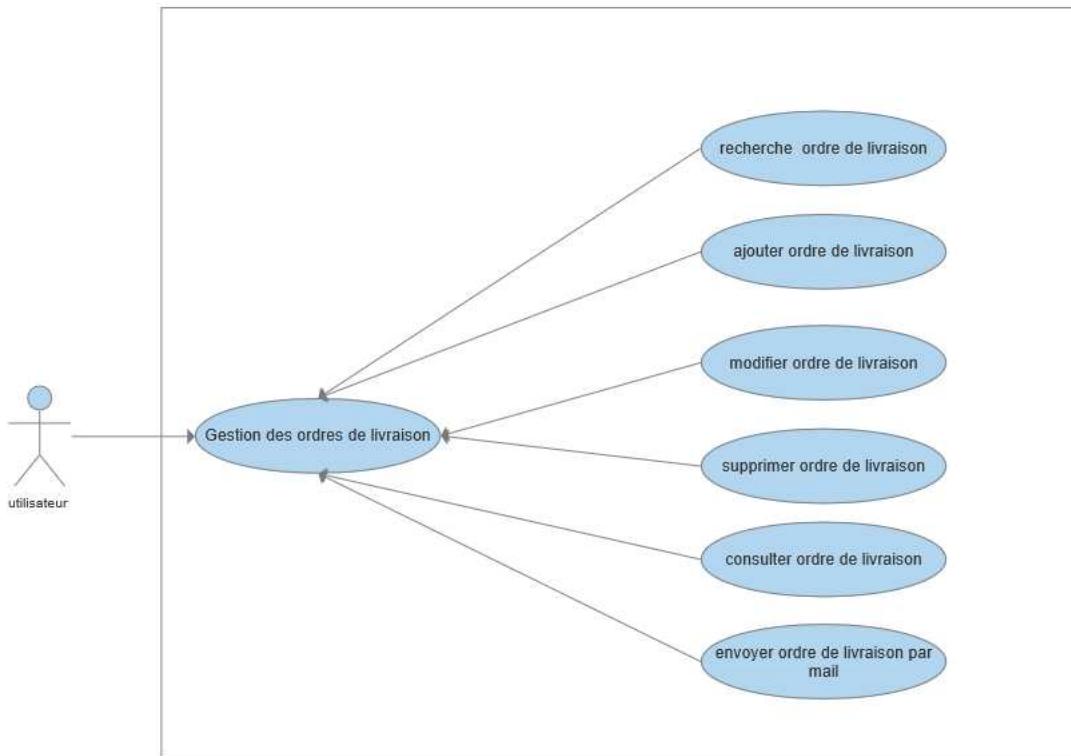


FIGURE 5.1 – «Diagramme de cas d'utilisation «Gestion des ordres de livraison»

Le tableau 5.2 montre le scénario du sous-cas d'utilisation «ajouter ordre de livraison» :

Chapitre 5 : sprint 3

Titre	Ajouter ordre de livraison
Objectif	L'utilisateur peut créer une nouvelle ordre de livraison.
Acteur	Utilisateur et Administrateur
Pré-conditions	-Authentification correcte à l'application. -L'administrateur demandé la page «Gérer ordre de livraison»
Déclencheur	L'utilisateur clique sur «ajouter»
Le scénario nominal	1- Le système affiche une page d'ajout. 2- L'utilisateur remplit le formulaire avec les informations d'ordre de livraison à créer. 3- l'utilisateur valide l'ajout. 4- Le système vérifie la présence des champs obligatoires. 5- Le système vérifie l'unicité du login. 6- Le système ajoute l'utilisateur et affiche un message de succès.
Le scénario alternatif	4.1- Le système renvoie un message d'erreur et signale à l'utilisateur que tous les champs obligatoires n'ont pas été remplis.
Post-conditions	Une nouvelle ordre de livraison est ajouté.

TABLE 5.2 – Description textuelle du CU «Ajouter ordre de livraison»

Le tableau 5.3 montre le scénario du sous-cas d'utilisation «modifier ordre de livraison» :

Titre	Modifier ordre de livraison
Objectif	L'utilisateur peut modifier une ordre de livraison
Acteur	Administrateur et l'utilisateur
Pré-conditions	-Authentification correcte à l'application. - Consultation de la liste des ordres de livraison.
Déclencheur	L'utilisateur clique sur «modifier»
Le scénario nominal	1- Le système affiche la page de modification. 2- L'acteur fait les modifications désirées. 3- L'acteur valide la modification. 4- Le système vérifie la présence des champs obligatoires. 5- Le système enregistre les modifications et affiche un message de succès.
Le scénario alternatif	4.1- Le système renvoie un message d'erreur et signale à l'utilisateur que tous les champs obligatoires n'ont pas été remplis.
Post-conditions	ordre de livraison modifié.

TABLE 5.3 – Description textuelle du CU «Modifier ordre de livraison»

5.2.2 Raffinement de cas d'utilisation «Gestion des paiements»

la figure 5.2 illustre le diagramme du cas d'utilisation «Gestion des paiements» :

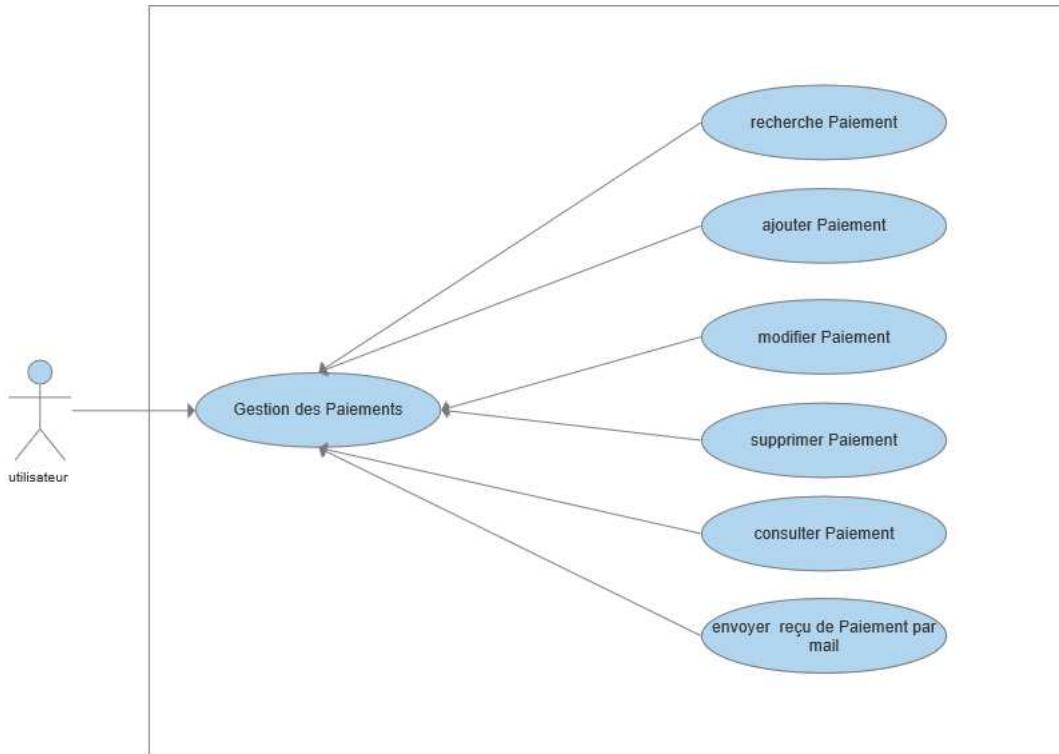


FIGURE 5.2 – «Diagramme de cas d'utilisation «Gestion des paiements»

Le tableau 5.4 montre le scénario du sous-cas d'utilisation «ajouter paiement» :

Chapitre 5 : sprint 3

Titre	Ajouter paiement
Objectif	L'utilisateur peut créer un nouveau paiement.
Acteur	Utilisateur et Administrateur
Pré-conditions	-Authentification correcte à l'application. -L'administrateur demandé la page «Gérer paiements»
Déclencheur	L'utilisateur clique sur «ajouter»
Le scénario nominal	1- Le système affiche une page d'ajout. 2- L'utilisateur remplit le formulaire avec les informations de paiement à créer. 3- l'utilisateur valide l'ajout. 4- Le système vérifie la présence des champs obligatoires. 5- Le système vérifie l'unicité du login. 6- Le système ajoute l'utilisateur et affiche un message de succès.
Le scénario alternatif	4.1- Le système renvoie un message d'erreur et signale à l'utilisateur que tous les champs obligatoires n'ont pas été remplis.
Post-conditions	un nouveau paiement est ajouté.

TABLE 5.4 – Description textuelle du CU «Ajouter paiement»

la tableau 5.5 illustre la Description textuelle du CU «supprimer paiement» :

Titre	Supprimer paiement
Objectif	L'acteur peut supprimer un paiement.
Acteur	Utilisateur et Administrateur
Pré-conditions	-Authentification correcte à l'application. -L'administrateur demandé la page «Gérer paiement»
Déclencheur	L'utilisateur a cliqué sur «Effacer»
Le scénario nominal	1- Une alerte de confirmation est affichée. 2- L'acteur confirme la suppression. 3- Le système supprime le profil et affiche un message de succès.
Le scénario alternatif	2-1. L'acteur annule la suppression
Post-conditions	Un paiement est supprimé.

TABLE 5.5 – Description textuelle du CU «Supprimer paiement»

5.2.3 Raffinement de cas d'utilisation «Gestion des dépenses»

la figure 5.3 illustre le diagramme du cas d'utilisation «Gestion des dépenses» :

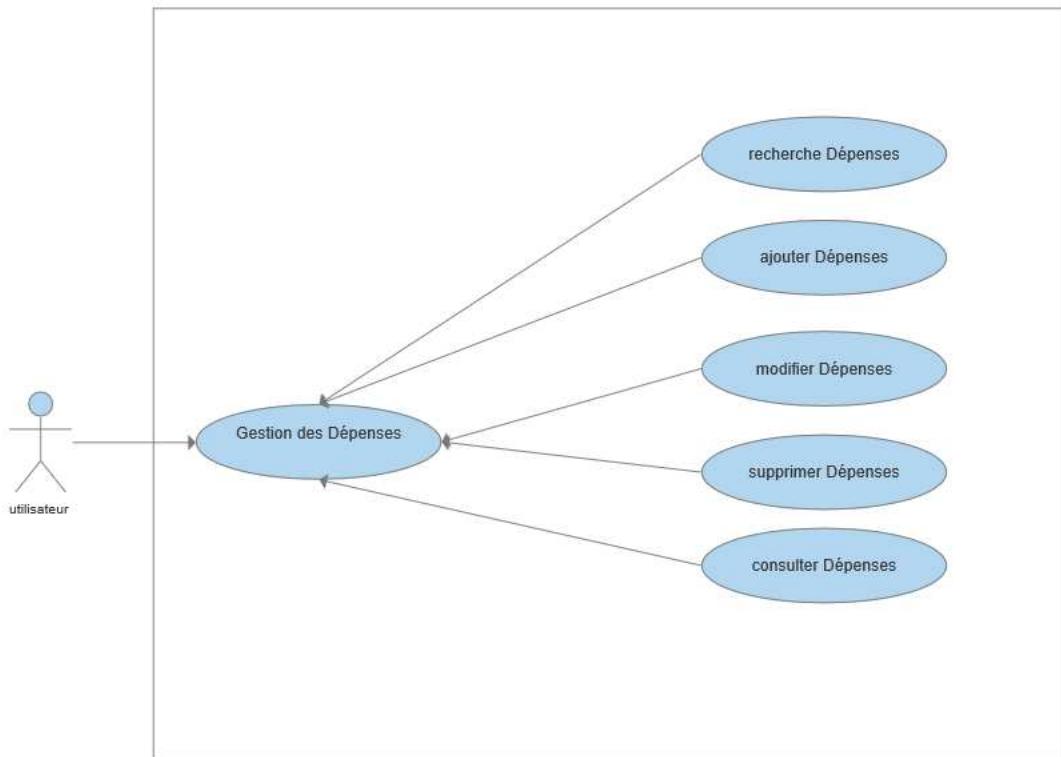


FIGURE 5.3 – «Diagramme de cas d'utilisation «Gestion des dépenses»

En décrivant le cas d'utilisation «recherche Dépense » , nous pouvons mieux comprendre l'enchaînement de cette fonctionnalité.

Titre	Recherche dépense
Objectif	Rechercher les dépenses en utilisant différents critères de recherche
Acteur	Utilisateur et Administrateur
Pré-conditions	-Authentification correcte à l'application. -L'administrateur demandé la page «Gérer dépenses»
Déclencheur	L'utilisateur a cliqué sur «Filtre»
Le scénario nominal	<p>1- L'utilisateur choisit le critère de recherche et clique sur rechercher.</p> <p>2- L'utilisateur peut modifier le formulaire.</p> <p>3- Le système affiche la liste des valeurs possibles pour ce critère</p> <p>4- L'utilisateur choisit parmi la liste .</p> <p>5- Le système affiche la liste correspondante .</p>
Post-conditions	Un filtrage sur les données est appliqué

TABLE 5.6 – Description textuelle du CU «recherche dépense»

5.3 Conception

Cette partie définit les éléments résultant de l'analyse des spécifications du troisième sprint. Afin d'en faciliter la réalisation, cette phase de conception sera définie comme la description du sprint à développer.

5.3.1 Diagramme de classes participantes

Le diagramme de classes décrit les associations entre les classes et ceci afin de déterminer les dépendances entre les différentes classes de ce sprint.

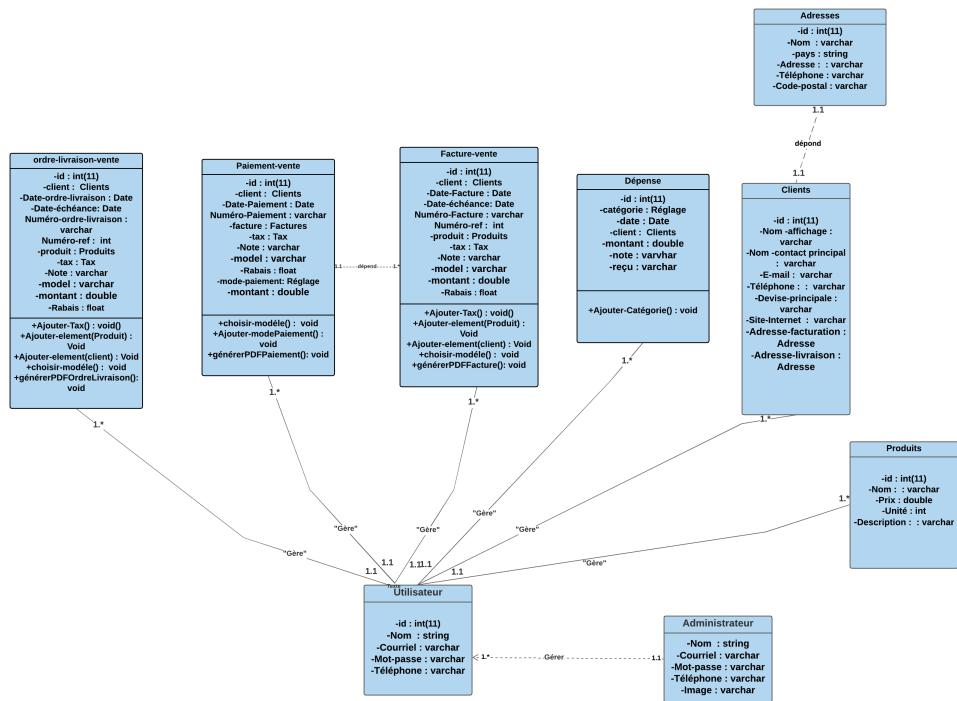


FIGURE 5.4 – Diagramme de classes du troisième sprint

5.3.2 Diagrammes de séquences

Grâce à ces diagrammes, nous pouvons décrire comment les éléments, qui composent le deuxième sprint, interagissent entre eux et avec les acteurs.

Diagramme de séquences du CU «Modifier ordre de livraison»

La modification d'une ordre de livraison se fait comme la modification des autres objets dans le Back Office. Dans ce cas, pour modifier les informations d'une ordre de livraison, il faut d'abord accéder à la liste des ordres de livraison, puis choisir l'ordre de livraison à modifier.

La figure 5.5 présente un diagramme de séquences qui exprime mieux cette fonctionnalité.

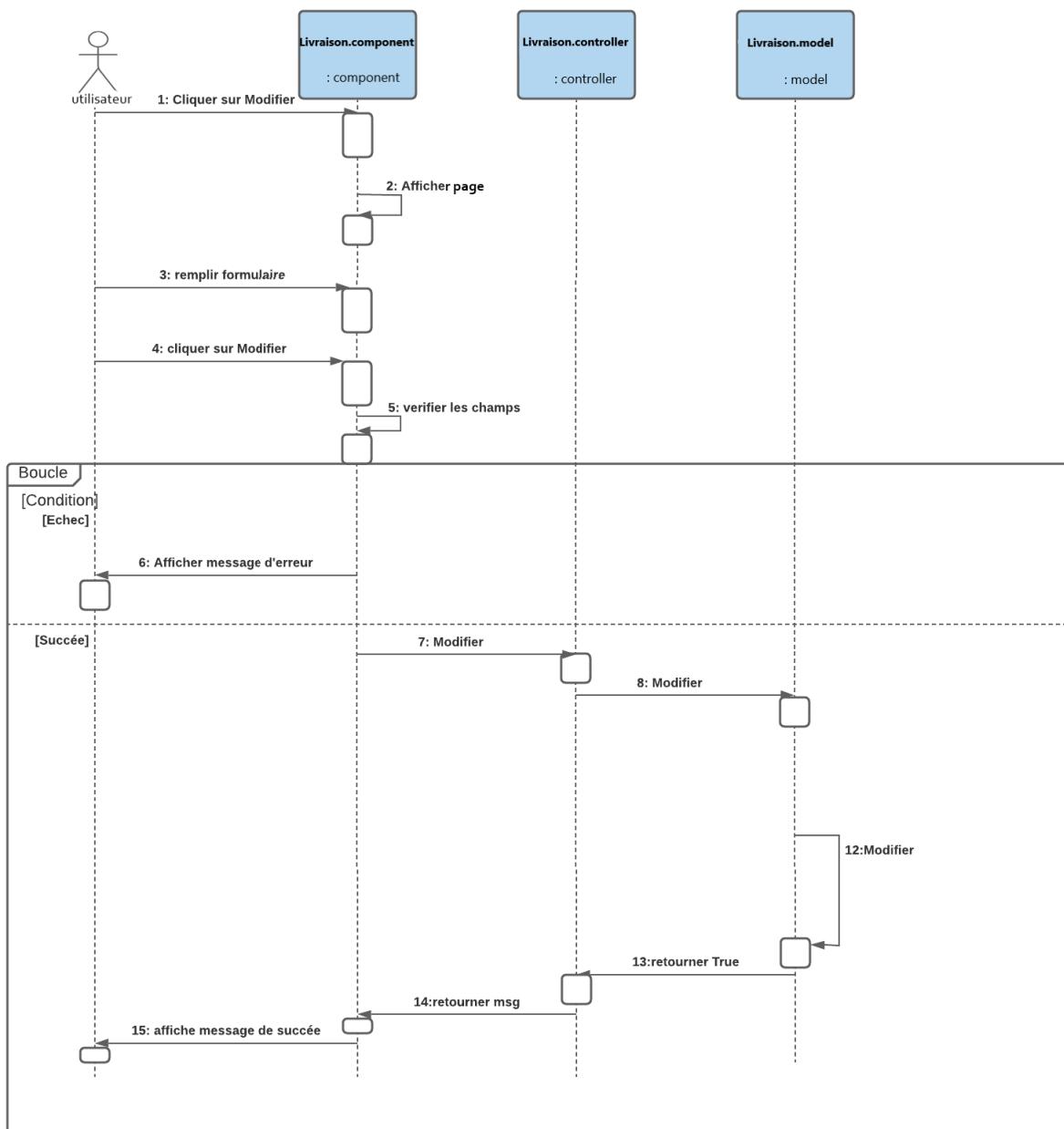


FIGURE 5.5 – Diagramme de séquences «Modifier ordre de livraison»

Diagramme de séquences du CU «Recherche ordre de livraison»

Le diagramme présenté dans la figure 5.6 montre l'enchaînement de ce cas d'utilisation.

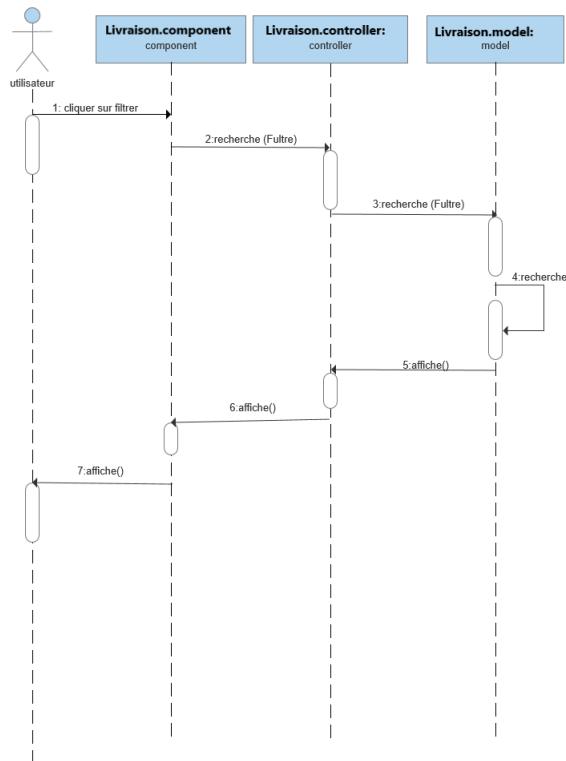


FIGURE 5.6 – Diagramme de séquences «Recherche ordre de livraison»

Diagramme de séquences du CU «Ajouter paiement »

Le diagramme 5.7 présente les interactions entre l'utilisateur et le système pour réaliser cette fonction.

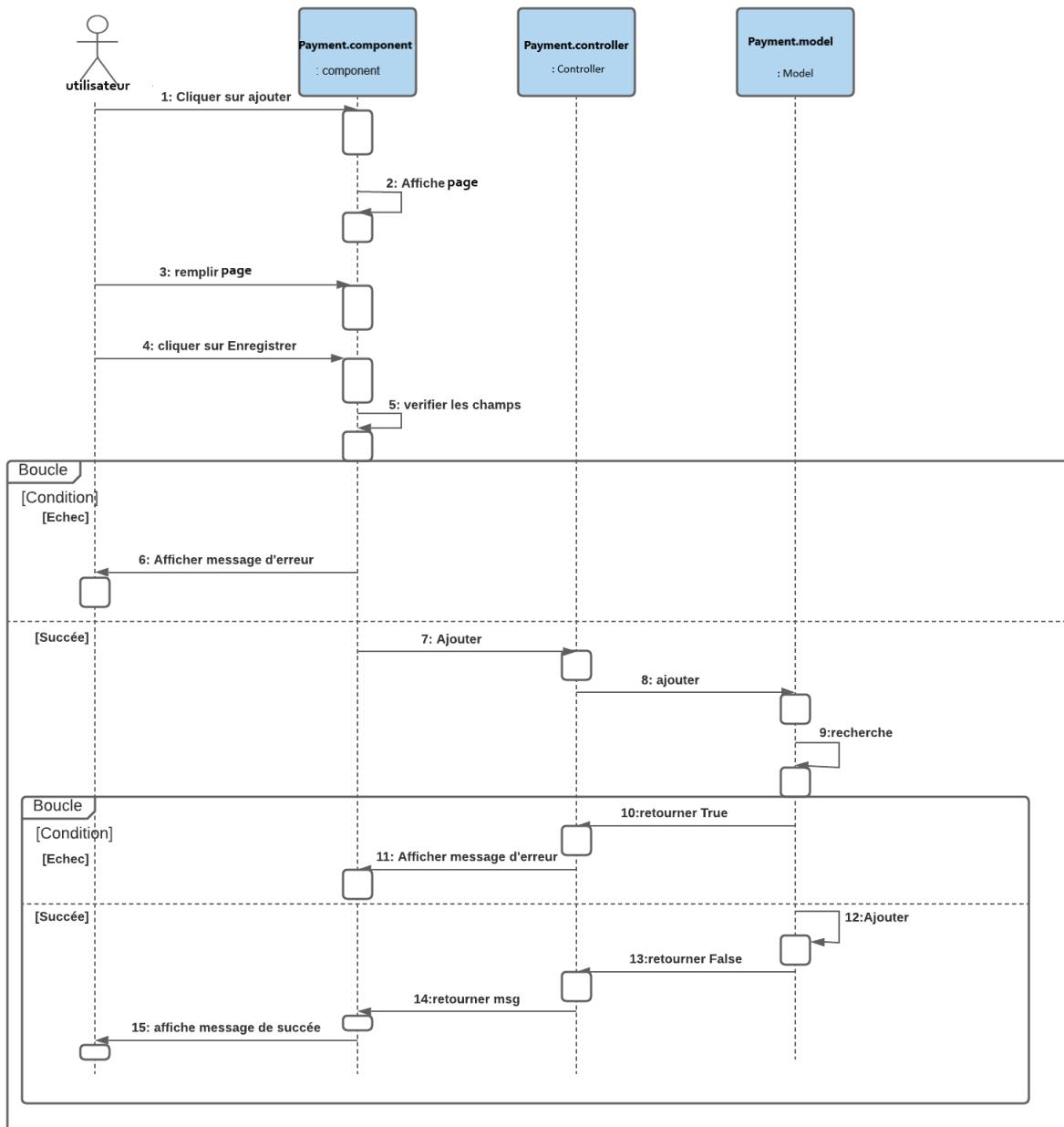


FIGURE 5.7 – Diagramme de séquences «Ajouter paiement»

Diagramme de séquences du CU «Supprimer Dépense»

Le diagramme 5.8 illustre les étapes de suppression d'une dépense.

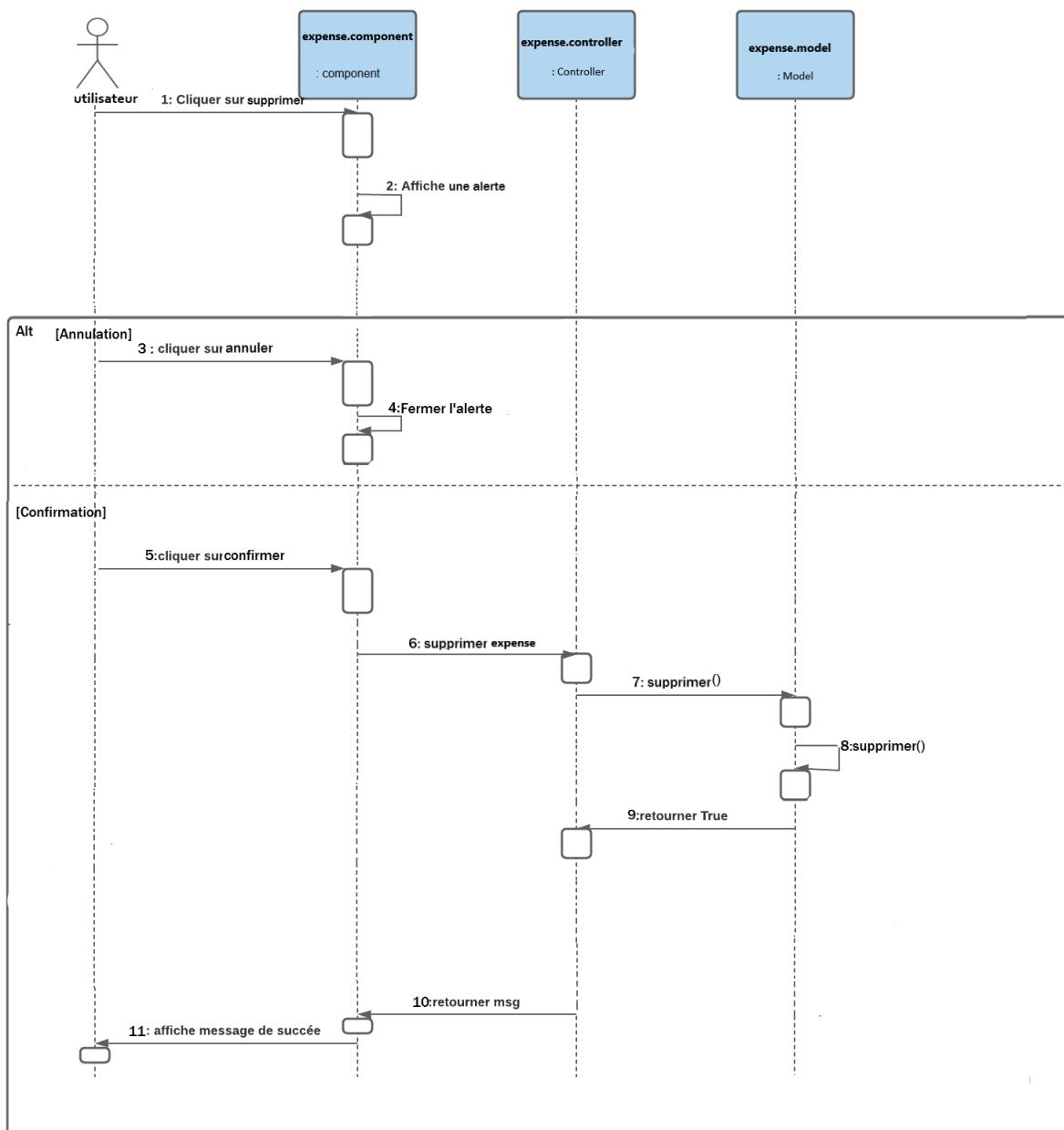


FIGURE 5.8 – Diagramme de séquences «Supprimer dépense»

5.4 Réalisation et tests

Passons maintenant à la présentation de la phase de réalisation du troisième sprint.

5.4.1 Gestion d'ordre de livraison

1. Modifier ordre de livraison

L'interface 5.8 montre la page qui sera affichée une fois que l'utilisateur va cliquer sur modifier.

FIGURE 5.9 – «Modifier ordre de livraison»

Voilà le résultat de cette action :

FIGURE 5.10 – «Résultat Modifier ordre de livraison»

2. Recherche ordre de livraison

Pour effectuer la recherche dans l'interface «ordre de livraison», l'utilisateur peut se baser sur différents critère comme nom, nom de contact et numéro de téléphone.

DATE	NUMBER	CUSTOMER	STATUS	PAID STATUS	AMOUNT DUE
28 Apr 2021	PAY-000001	mohamed	DRAFT	UNPAID	\$ 210.00

FIGURE 5.11 – « Recherche ordre de livraison»

5.4.2 Gestion des paiements

1.Ajouter paiement

L'interface 5.12 montre la page qui sera affichée une fois que l'utilisateur va cliquer sur ajouter.

FIGURE 5.12 – «Ajouter paiement »

Voilà le résultat de l'ajout :

The screenshot shows the Devlop application interface. On the left, there is a sidebar with the following navigation options:

- Dashboard
- Customers
- Items
- Estimates
- Invoices
- Delivery Orders
- Payments** (selected)
- Expenses
- Users

The main content area displays a list of payments under the heading "PAYMENTS". There are three entries:

Customer	Payment ID	Amount	Date
Mohamed	PAY-000003	\$ 211.00	05 May 2021
Mohamed	PAY-000002	\$ 210.98	20 Apr 2021
Mohamed	PAY-000001	\$ 210.98	20 Apr 2021

On the right, a detailed view of the first payment is shown. The payment ID is PAY-000003. The details are:

Mohamed
PAY-000003
INV-000003
\$ 211.00
05 May 2021

Maram
PAYMENT RECEIPT
Payment Date: 05 May 2021
Payment Number: PAY-000003
Payment Mode: Cash
Amount Received: \$211.00

FIGURE 5.13 – «Résultat Ajouter paiement »

5.4.3 Gestion des Dépenses

L'interface présentée dans la figure 5.14 montre une alerte qui va être affichée lors de la suppression d'un Dépense pour confirmer l'action.

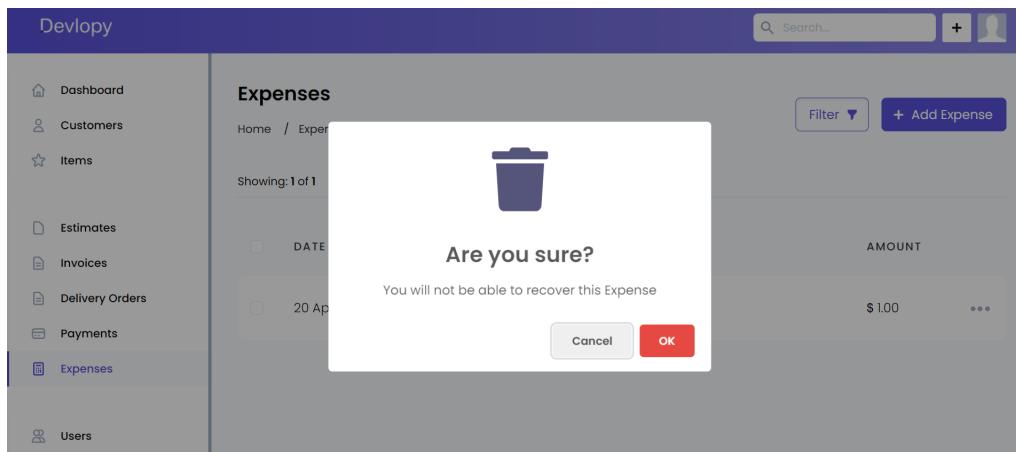


FIGURE 5.14 – «Supprimer Dépense»

5.5 Conclusion

Ce chapitre a présenté la spécification des fonctionnalités du troisième sprint à l'aide des diagrammes de cas d'utilisation. Ensuite, il a donné l'aspect conceptuel en listant les résultats attendus en termes de fonctionnalités. Ces dernières ont été illustrées par quelques interfaces réalisées. Le chapitre suivant fera l'objet d'une étude détaillée du troisième sprint.

6

Sprint 4

Sommaire

Introduction	103
6.1 Backlog du sprint	103
6.2 Spécification	105
6.2.1 Cas d'utilisation «Gestion des Fournisseurs»	105
6.2.2 Cas d'utilisation «Gestion des Factures d'achat»	106
6.2.3 Raffinement de cas d'utilisation «Gestion des ordres de livraison d'achat»	107
6.3 Conception	110
6.3.1 Diagramme de classes participantes	110
6.3.2 Diagrammes de séquences	110
6.4 Réalisation et tests	114
6.4.1 Gestion Fournisseur	114
6.4.2 Gestion des Factures d'achat	115
6.4.3 Gestion des ordres de livraison d'achat	117
6.5 Conclusion	117

Introduction

Le but principal du quatrième sprint est de présenter le cas d'utilisation : Gestion Tableau de bord.

6.1 Backlog du sprint

Le tableau 6.1 montre notre troisième sprint backlog :

ID	Fonctionnalité	User Story	Priorité
1	Gestion des Fournisseurs	En tant qu'utilisateur, je veux ajouter un ou plusieurs Fournisseurs.	Élevée
		En tant qu'utilisateur, je veux modifier un Fournisseur bien déterminée.	Élevée
		En tant qu'administrateur, je veux consulter la liste des Fournisseurs.	Élevée
		En tant qu'administrateur je veux réaliser tous les tâches du Fournisseurs.	Moyenne
2	Gestion des factures d'achat	En tant qu'utilisateur, je veux ajouter une ou plusieurs factures d'achat.	Élevée
		En tant qu'utilisateur, je veux modifier une facture d'achat bien déterminée.	Élevée
		En tant qu'utilisateur, je veux supprimer une facture d'achat bien déterminée.	Élevée
		En tant qu'utilisateur, je veux consulter la liste des factures d'achat.	Moyenne
3	Gestion des ordres de livraison d'achat	En tant qu'utilisateur, je veux ajouter une ou plusieurs ordres de livraison d'achat.	Élevée
		En tant qu'utilisateur, je veux modifier une ordre de livraison d'achat bien déterminée.	Élevée
		En tant qu'utilisateur, je veux supprimer une ordre de livraison d'achat. bien déterminée.	Élevée
		En tant qu'utilisateur, je veux consulter la liste des ordres de livraison d'achat.	Moyenne

TABLE 6.1 – Backlog du sprint 4

6.2 Spécification

le quatrième sprint porte principalement sur les fonctionnalités, déjà élaborées dans la partie exigences fonctionnelles, qui seront raffinées à l'aide des diagrammes de cas d'utilisation.

6.2.1 Cas d'utilisation «Gestion des Fournisseurs»

la Gestion des Fournisseurs est une fonctionnalité offerte non seulement à l'admin mais aussi à tout les utilisateurs, la figure 6.1 nous donne une vision globale sur cette fonctionnalité :

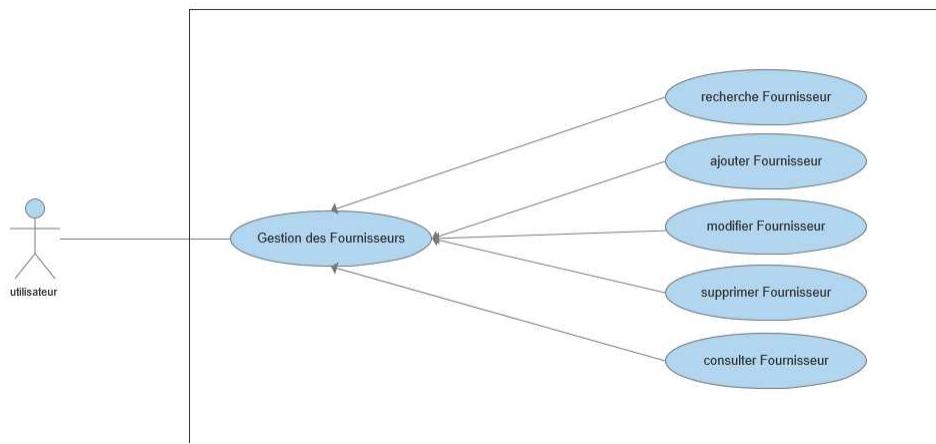


FIGURE 6.1 – Diagramme de raffinement du cas d'utilisation «Gestion des Fournisseurs»

Nous abordons le déroulement de la consultation d'un fournisseur :

Titre	consulter Fournisseur
Objectif	consulter Fournisseur
Acteur	Administrateur et l'utilisateur
Pré-conditions	-Authentification correcte à l'application. - Consultation de la liste des Fournisseurs.
Déclencheur	L'utilisateur clique sur «view»
Le scénario nominal	1- Le système affiche une interface qui contient la list des Fournisseurs accompagnée d'une courbe qui représente le montant d'achat chez le fournisseur selectionner avec ses information

TABLE 6.2 – Description textuelle du CU «consulter fournisseur»

6.2.2 Cas d'utilisation «Gestion des Factures d'achat»

La Gestion des Factures d'achat est l'une des fonctionnalités les plus importantes dans notre application. La figure 6.2 illustre ce cas d'utilisation. En décrivant le cas d'utilisation

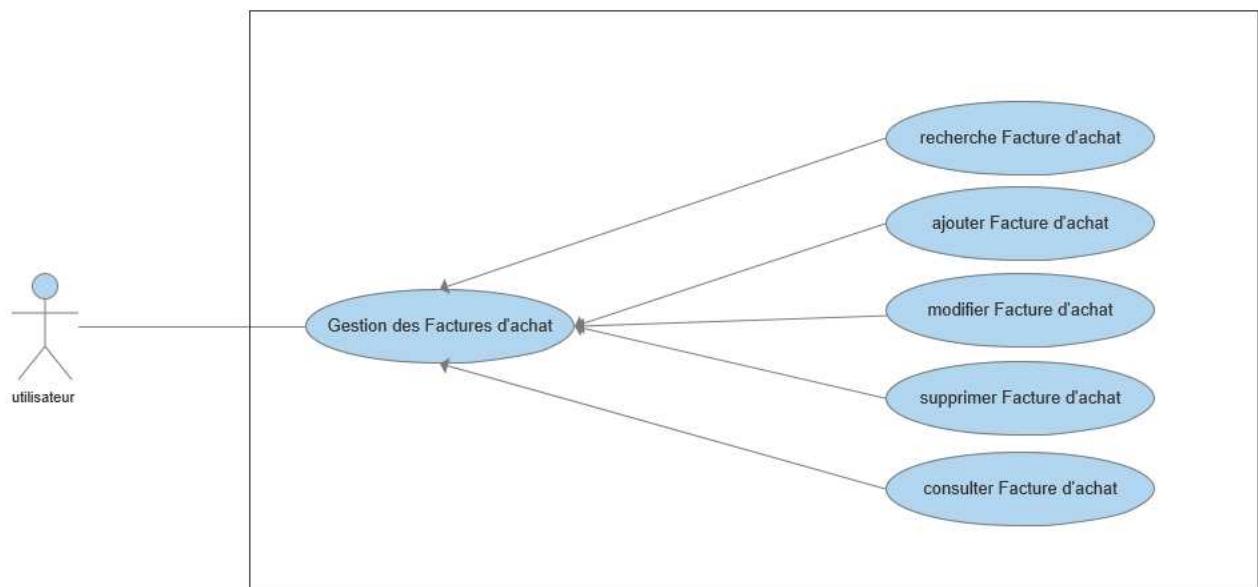


FIGURE 6.2 – Diagramme de raffinement du cas d'utilisation «Gestion des factures d'achat»

tion «recherche Facture d'achat», nous pouvons mieux comprendre l'enchaînement de cette fonctionnalité.

la tableau 6.4 illustre la Description textuelle du CU «supprimer facture d'achat»

Titre	Recherche facture d'achat
Objectif	Rechercher les factures d'achat en utilisant différents critères de recherche
Acteur	Utilisateur et Administrateur
Pré-conditions	-Authentification correcte à l'application. -L'administrateur demandé la page «Gérer factures d'achat»
Déclencheur	L'utilisateur a cliqué sur «Filtre»
Le scénario nominal	<p>1- L'utilisateur choisit le critère de recherche et clique sur rechercher.</p> <p>2- Le système affiche la liste des valeurs possibles pour ce critère</p> <p>3- L'utilisateur choisit parmi la liste .</p> <p>4- Le système affiche la liste correspondante .</p>
Post-conditions	Un filtrage sur les données est appliqué

TABLE 6.3 – Description textuelle du CU «recherche facture d'achat»

Titre	Supprimer Facture d'achat
Objectif	L'acteur peut supprimer une facture d'achat.
Acteur	Utilisateur et Administrateur
Pré-conditions	-Authentification correcte à l'application. -L'administrateur demandé la page «Gérer factures»
Déclencheur	L'utilisateur a cliqué sur «Effacer»
Le scénario nominal	<p>1- Une alerte de confirmation est affichée.</p> <p>2- L'acteur confirme la suppression.</p> <p>3- Le système supprime la facture d'achat et affiche un message de succès.</p>
Le scénario alternatif	2-1. L'acteur annule la suppression
Post-conditions	Une facture d'achat est supprimée.

TABLE 6.4 – Description textuelle du CU «Supprimer facture d'achat»

6.2.3 Raffinement de cas d'utilisation «Gestion des ordres de livraison d'achat»

la figure 6.3 illustre le diagramme du cas d'utilisation «Gestion des ordres de livraison d'achat» :

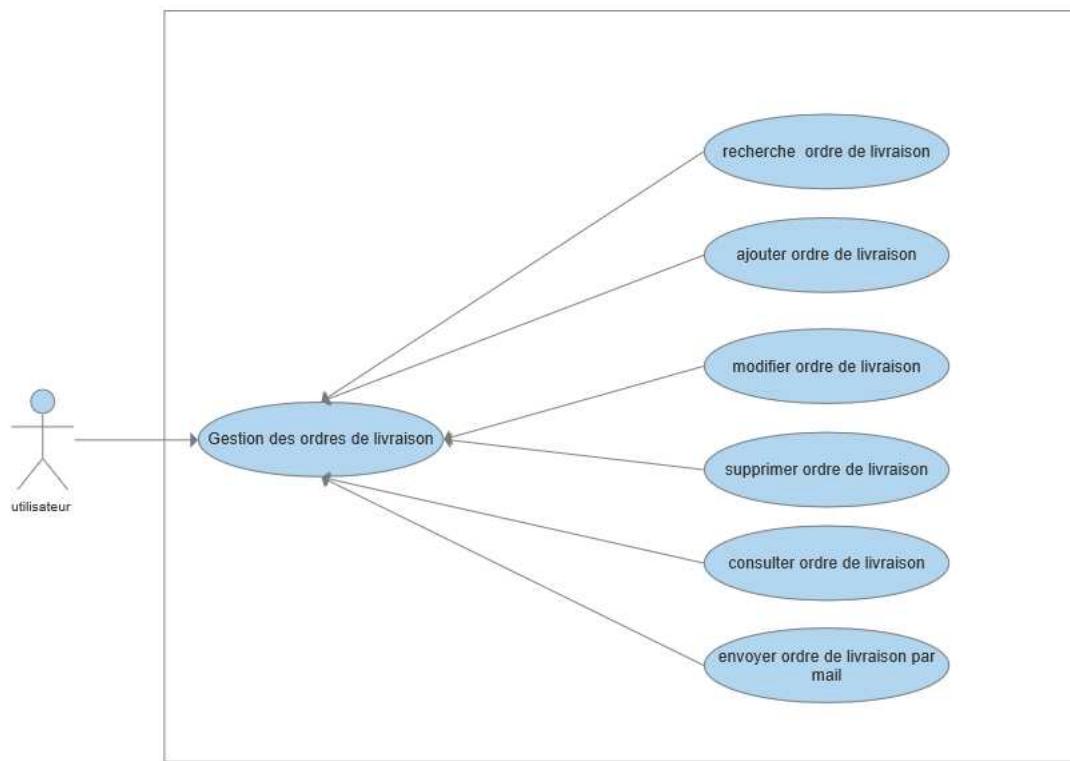


FIGURE 6.3 – «Diagramme de cas d'utilisation «Gestion des ordres de livraison d'achat»

Le tableau 6.5 montre le scénario du sous-cas d'utilisation «ajouter ordre de livraison d'achat» :

Chapitre 6 : sprint 4

Titre	Ajouter ordre de livraison d'achat
Objectif	L'utilisateur peut créer une nouvelle ordre de livraison d'achat.
Acteur	Utilisateur et Administrateur
Pré-conditions	-Authentification correcte à l'application. -L'administrateur demandé la page «Gérer ordre de livraison»
Déclencheur	L'utilisateur clique sur «ajouter»
Le scénario nominal	1- Le système affiche une page d'ajout. 2- L'utilisateur remplit le formulaire avec les informations d'ordre de livraison d'achat à créer. 3- l'utilisateur valide l'ajout. 4- Le système vérifie la présence des champs obligatoires. 5- Le système vérifie l'unicité du login. 6- Le système ajoute l'utilisateur et affiche un message de succès.
Le scénario alternatif	4.1- Le système renvoie un message d'erreur et signale à l'utilisateur que tous les champs obligatoires n'ont pas été remplis.
Post-conditions	Une nouvelle ordre de livraison d'achat est ajouté.

TABLE 6.5 – Description textuelle du CU «Ajouter ordre de livraison d'achat»

Le tableau 6.6 montre le scénario du sous-cas d'utilisation «modifier ordre de livraison d'achat» :

Titre	Modifier ordre de livraison d'achat
Objectif	L'utilisateur peut modifier une ordre de livraison d'achat
Acteur	Administrateur et l'utilisateur
Pré-conditions	-Authentification correcte à l'application. - Consultation de la liste des ordres de livraison d'achat.
Déclencheur	L'utilisateur clique sur «modifier»
Le scénario nominal	1- Le système affiche la page de modification. 2- L'acteur fait les modifications désirées. 3- L'acteur valide la modification. 4- Le système vérifie la présence des champs obligatoires. 5- Le système enregistre les modifications et affiche un message de succès.
Le scénario alternatif	4.1- Le système renvoie un message d'erreur et signale à l'utilisateur que tous les champs obligatoires n'ont pas été remplis.
Post-conditions	ordre de livraison d'achat modifié.

TABLE 6.6 – Description textuelle du CU «Modifier ordre de livraison d'achat»

6.3 Conception

Cette partie définit les éléments résultant de l'analyse des spécifications du quatrième sprint. Afin d'en faciliter la réalisation, cette phase de conception sera définie comme la description du sprint à développer.

6.3.1 Diagramme de classes participantes

Le diagramme de classes décrit les associations entre les classes et ceci afin de déterminer les dépendances entre les différentes classes de ce sprint.

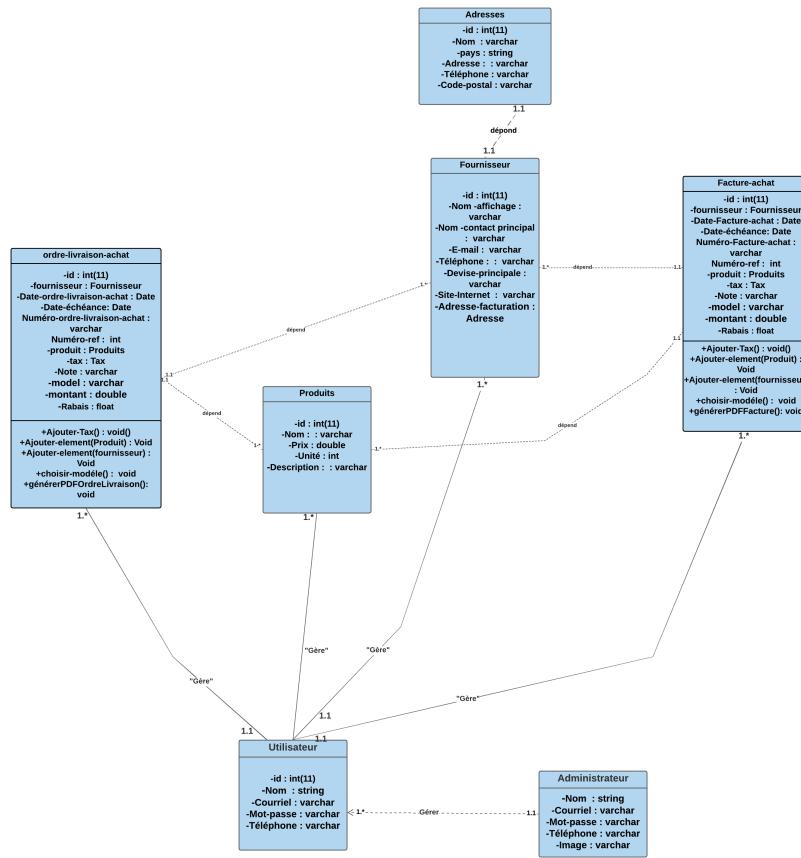


FIGURE 6.4 – Diagramme de classes du quatrième sprint

6.3.2 Diagrammes de séquences

Grâce à ces diagrammes, nous pouvons décrire comment les éléments, qui composent le deuxième sprint, interagissent entre eux et avec les acteurs.

Diagramme de séquences du CU «consulter Fournisseur» :
Le diagramme suivant illustre les étapes pour consulter un fournisseur.

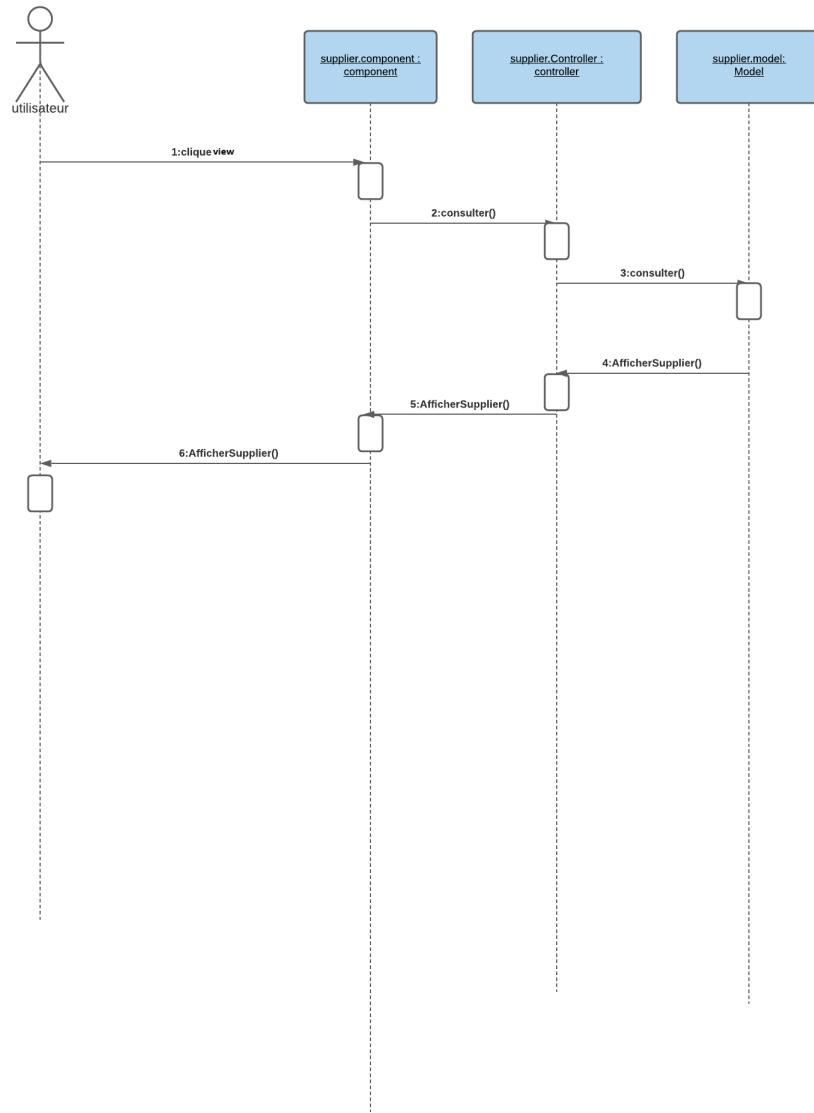


FIGURE 6.5 – Diagramme de séquences «consulter Fournisseur»

Diagramme de séquences du CU «Recherche Facture d'achat »

Le diagramme 6.6 présente les interactions entre l'utilisateur et le système pour réaliser cette fonction.

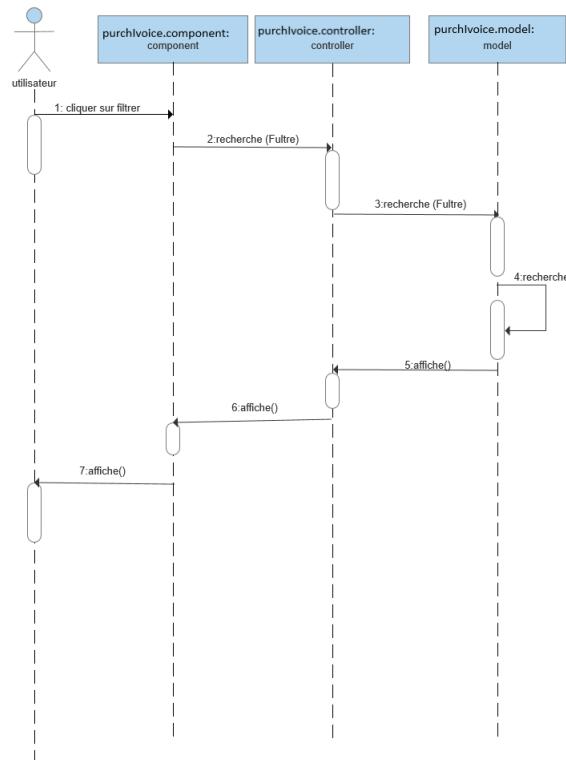


FIGURE 6.6 – Diagramme de séquences «Recherche Facture d'achat»

Diagramme de séquences du CU «Ajouter ordre de livraison d'achat»

Le diagramme présenté dans la figure 6.7 montre l'enchaînement de ce cas d'utilisation.

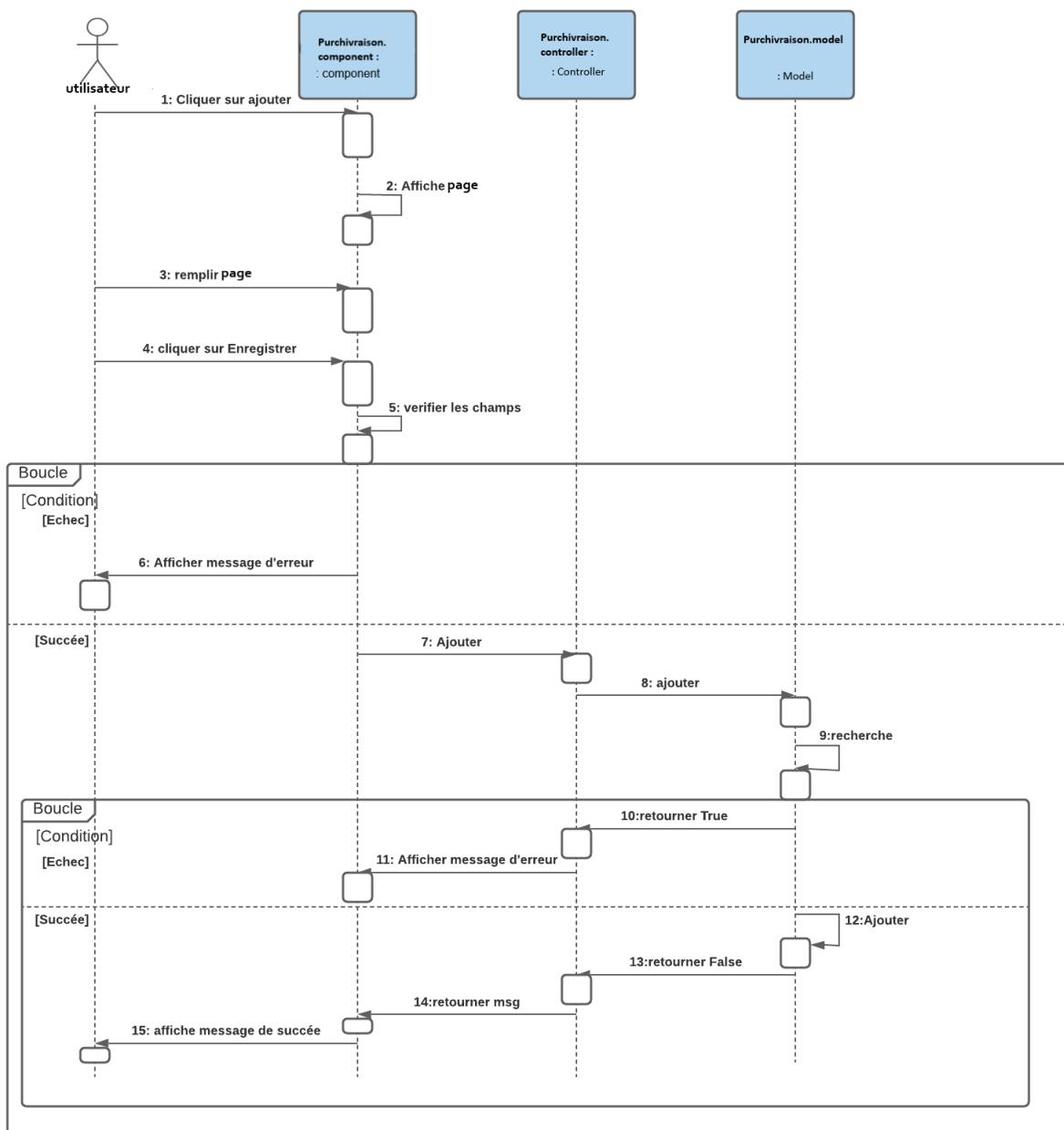


FIGURE 6.7 – Diagramme de séquences «Ajouter ordre de livraison d'achat»

6.4 Réalisation et tests

Passons maintenant à la présentation de la phase de réalisation du quatrième sprint.

6.4.1 Gestion Fournisseur

1. Consulter Fournisseur

L'interface de la figure 6.8 montre la page qui sera affichée une fois que l'utilisateur va cliquer sur "view".

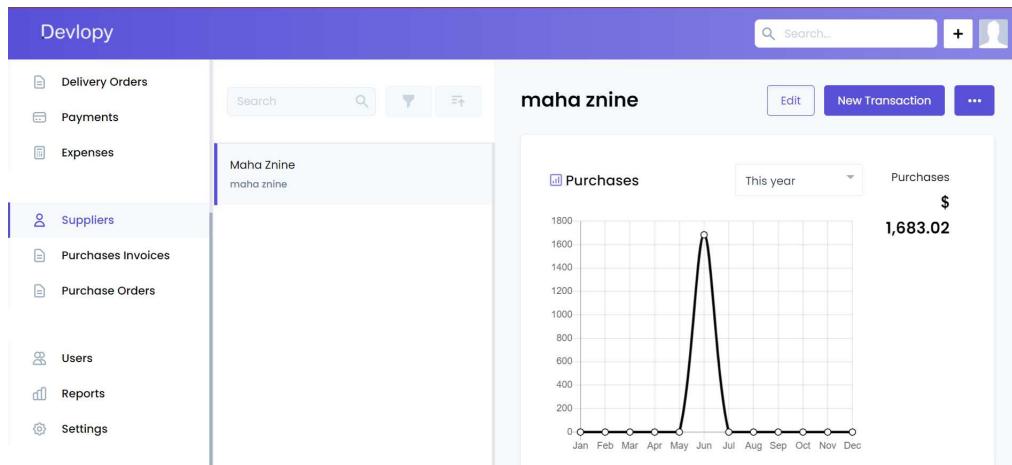


FIGURE 6.8 – « Consulter Fournisseur 1 »

Chapitre 6 : sprint 4

l'interface consulter fournisseur est accompagnée d'une courbe qui montre les achats effectués auprès du fournisseur sélectionné avec toutes ces informations, en plus, on trouve une liste de tous les fournisseurs avec l'option de filtrage.

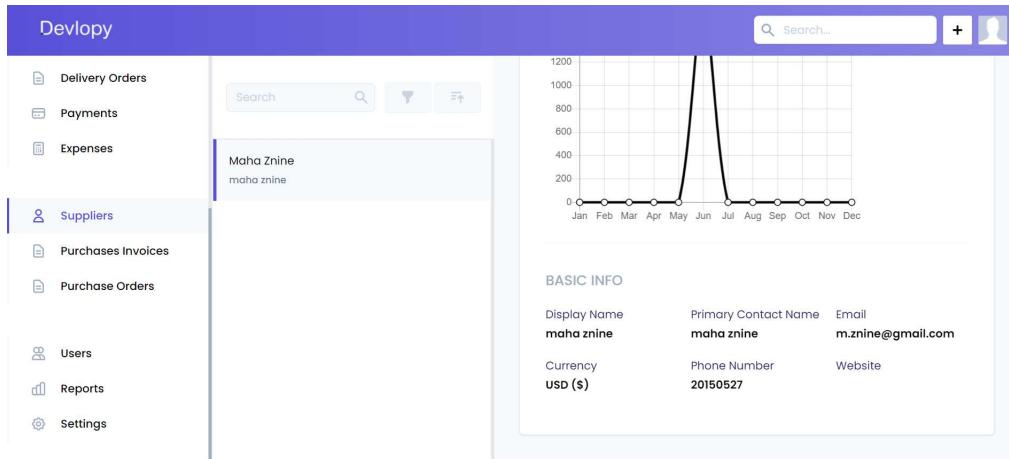


FIGURE 6.9 – « Consulter Fournisseur 2»

6.4.2 Gestion des Factures d'achat

1. ajouter Facture d'achat

La figure 6.10 montre un formulaire d'ajout d'une Facture d'achat.

The screenshot shows the 'New Purchase Invoice' form within the Devlop application. The sidebar on the left is identical to Figure 6.9. The main form area is titled 'New Purchase Invoice' and includes the following fields:

- Supplier selection: 'maha znine' with 'Edit' and 'Deselect' buttons.
- Date fields: 'Purchase Invoice Date *' (06 Jun 2021) and 'Due Date *' (13 Jun 2021).
- Number fields: 'Purchases Invoice Number *' (# PINV - 004) and 'Ref Number' (#).
- Table for item details:

Items	Quantity	Price	Amount
NIKE Air Force 1 '07 whit shoes	2	\$ 210.00	\$ 420.00

FIGURE 6.10 – « ajouter Facture d'achat 1»

Chapitre 6 : sprint 4

l'utilisateur va choisir le fournisseur, le produit et la quantité à acheter, en suite il va prissiez les impôt et le montant du timbre fiscale.

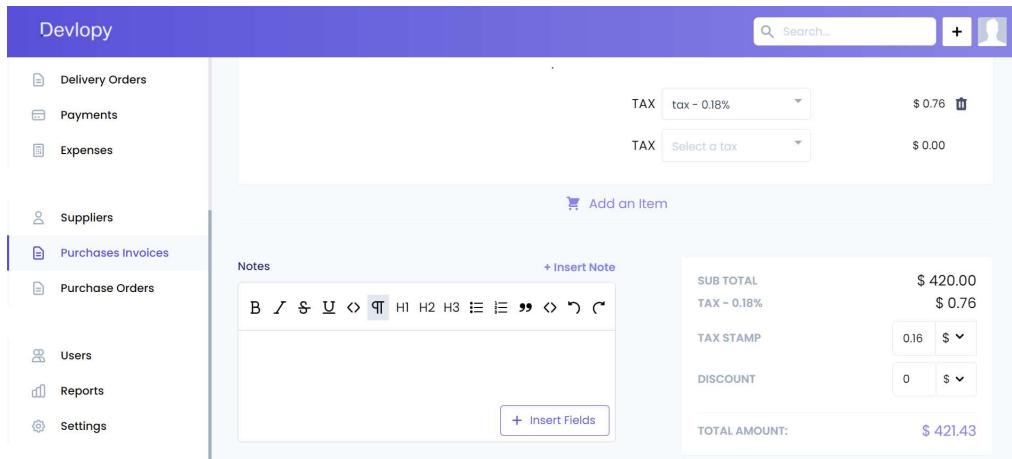


FIGURE 6.11 – « ajouter Facture d'achat 2»

voila le résultat d'ajout d'une nouvelle Facture d'achat.

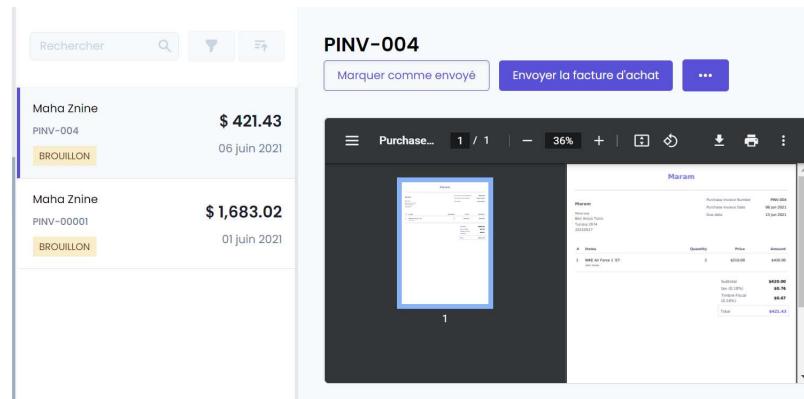


FIGURE 6.12 – « ajouter Facture d'achat 3»

6.4.3 Gestion des ordres de livraison d'achat

1. modifier ordre de livraison d'achat

L'utilisateur peut modifier les données relatives à une ordre de livraison d'achat donné et c'est en cliquant sur le bouton de modification qu'une interface s'affiche contenant toutes les informations nécessaires. La figure 6.13 met l'accent sur cette interface.

Items	Quantity	Price	Amount
NIKE Air Force 1 '07 whit shoes	1	\$ 210.00	\$ 210.00

FIGURE 6.13 – « modifier ordre de livraison d'achat»

6.5 Conclusion

Ce chapitre a présenté la spécification des fonctionnalités du quatrième sprint à l'aide des diagrammes de cas d'utilisation. Ensuite, il a donné l'aspect conceptuel en listant les résultats attendus en termes de fonctionnalités. Ces dernières ont été illustrées par quelques interfaces réalisées. Le chapitre suivant fera l'objet d'une étude détaillée du cinquième sprint.

7

Sprint 5

Sommaire

Introduction	119
7.1 Backlog du sprint	119
7.2 Spécifications fonctionnelles	121
7.2.1 Cas d'utilisation «Gestion de Tableau de bord»	121
7.2.2 Description textuelle « consulter le Tableau de bord »	122
7.2.3 Cas d'utilisation «Gestion des paramétrages»	123
7.2.4 Description textuelle « gérer compte »	123
7.2.5 Cas d'utilisation «Gestion des Rapports»	125
7.3 Conception	126
7.3.1 Diagramme de classes participantes	126
7.3.2 Diagrammes de séquences	127
7.4 Réalisation et tests	130
7.5 Conclusion	134

Introduction

Après avoir achevé les quatre derniers sprint, nous passons à la réalisation du cinquième sprint. Ce chapitre sera consacré en premier lieu à la présentation de son backlog qui sera spécifié à l'aide des diagrammes des cas d'utilisation. En second lieu, nous concevrons ce sprint avec un diagramme de classes, suivi des diagrammes de séquences. Pour finir, nous présentons quelques interfaces graphiques qui correspondent à des fonctionnalités importantes offertes par ce sprint.

7.1 Backlog du sprint

Le tableau montre notre cinquième sprint backlog :

ID	Fonctionnalité	User Story	Priorité
1	Gestion de Tableau de bord	En tant qu'utilisateur, je veux consulter toutes les mesures clés du projet : le montant du, nombre des client, nombre des factures, nombre des devises, montant des ventes, montant reçus, montant des dépenses, et le revenu net	Moyenne
		En tant qu'utilisateur, je veux avoir une courbe statistique récapitulatif hebdomadaire fournit un bref résumé des ventes , le montant reçus et des dépenses de l'année passée et cette année	Moyenne
		En tant qu'utilisateur, je veux consulter les Factures dues et Estimations récentes sur le tableau de bord	Moyenne
2	Gestion des paramétrages (Réglages)	En tant qu'utilisateur, je veux gérer mon compte	Élevée
		En tant qu'utilisateur, je veux gérer les informations de l'entreprise	Élevée
		En tant qu'utilisateur, je veux gérer les Préférences.	Moyenne
		En tant qu'utilisateur, je veux gérer les Taxes, les Modes de paiement.	Moyenne

ID	Fonctionnalité	User Story	Priorité
3	Gestion des Rapports	En tant qu'Administrateur, je veux consulter les Rapports des ventes, les Rapports des profits , les Rapports des Dépenses et les Rapports des Taxes par jour, semaine, mois, trimestre et année.	Moyenne
		En tant qu'Administrateur, je veux télécharger le document en PDF et de l'imprimer	Faible

TABLE 7.1 – Backlog du sprint 5

7.2 Spécifications fonctionnelles

Après avoir présenté le backlog du sprint 5 ,nous allons présenter également les spécifications fonctionnelles de ce sprint.

7.2.1 Cas d'utilisation «Gestion de Tableau de bord»

La figure 7.1 donne une idée sur cette fonctionnalité.

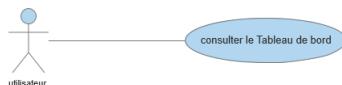


FIGURE 7.1 – Diagramme de raffinement du cas d'utilisation «consulter le Tableau de bord»

7.2.2 Description textuelle « consulter le Tableau de bord »

Le tableau 7.2 illustre le scénario de cas d'utilisation «consulter le Tableau de bord» :

Titre	consulter le Tableau de bord
Objectif	L'utilisateur peut consulter le Tableau de bord
Acteur	Administrateur et l'utilisateur
Pré-conditions	-Authentification correcte à l'application.
Déclencheur	L'utilisateur a cliqué sur Tableau de bord
Le scénario nominal	1)L'utilisateur choisit une option "Tableau de bord". 2)Le système affiche une interface qui contient une courbe descriptive des ventes, achats, Dépenses et le Revenu net, en plus, toutes les mesures clés du projet.
Post-conditions	Tableau de bord affiché

TABLE 7.2 – Description textuelle du CU «Consulter Tableau de bord»

7.2.3 Cas d'utilisation «Gestion des paramétrages»

la figure 7.2 illustre le diagramme du cas d'utilisation «Gestion des paramétrages» :

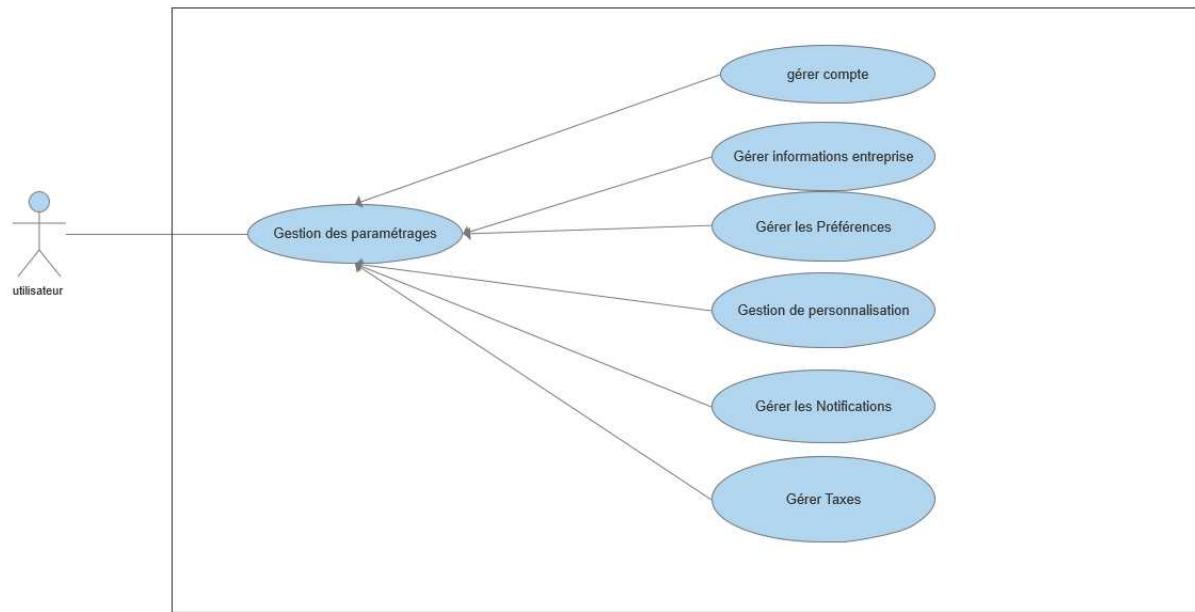


FIGURE 7.2 – Diagramme de raffinement du cas d'utilisation «Gestion des paramétrages»

7.2.4 Description textuelle « gérer compte »

Le tableau 7.3 illustre le scénario de cas d'utilisation «gérer compte» :

Chapitre 7 : sprint 5 «Paramétrage»

Titre	gérer compte
Objectif	L'utilisateur peut modifier les informations du compte
Acteur	Administrateur et l'utilisateur
Pré-conditions	-Authentification correcte à l'application. - L'acteur a consulté «Gestion des paramétrages».
Déclencheur	L'utilisateur a cliqué Paramètres du compte
Le scénario nominal	1- Le système affiche un formulaire qui contient toutes les informations concernant le compte. 2- L'acteur fait les modifications souhaitées. 3- L'acteur valide les modifications. 4- Le système vérifie la présence des champs obligatoires. 5- Le système enregistre les modifications et affiche un message de succès.
Le scénario alternatif	4-1.Si les champs obligatoires sont vides, le système affiche un message d'erreur.
Post-conditions	les informations du compte sont modifiées

TABLE 7.3 – Description textuelle du CU «gérer compte»

Passons maintenant à présenter la description textuelle du cas «Gérer les informations de l'entreprise» :

Titre	Gérer les informations de l'entreprise
Objectif	L'utilisateur peut modifier les informations de l'entreprise
Acteur	Administrateur et l'utilisateur
Pré-conditions	-Authentification correcte à l'application. - L'acteur a consulté «Gestion des paramétrages».
Déclencheur	L'utilisateur a cliqué Informations d'entreprise
Le scénario nominal	1- Le système affiche un formulaire qui contient toutes les informations concernant l'entreprise. 2- L'acteur fait les modifications souhaitées. 3- L'acteur valide les modifications. 4- Le système vérifie la présence des champs obligatoires. 5- Le système enregistre les modifications et affiche un message de succès.
Le scénario alternatif	4-1.Si les champs obligatoires sont vides, le système affiche un message d'erreur.
Post-conditions	les informations de l'entreprise sont modifiées

TABLE 7.4 – Description textuelle du CU «Gérer les informations de l'entreprise»

7.2.5 Cas d'utilisation «Gestion des Rapports»

la figure 7.3 illustre le diagramme du cas d'utilisation «Gestion des Rapports» :

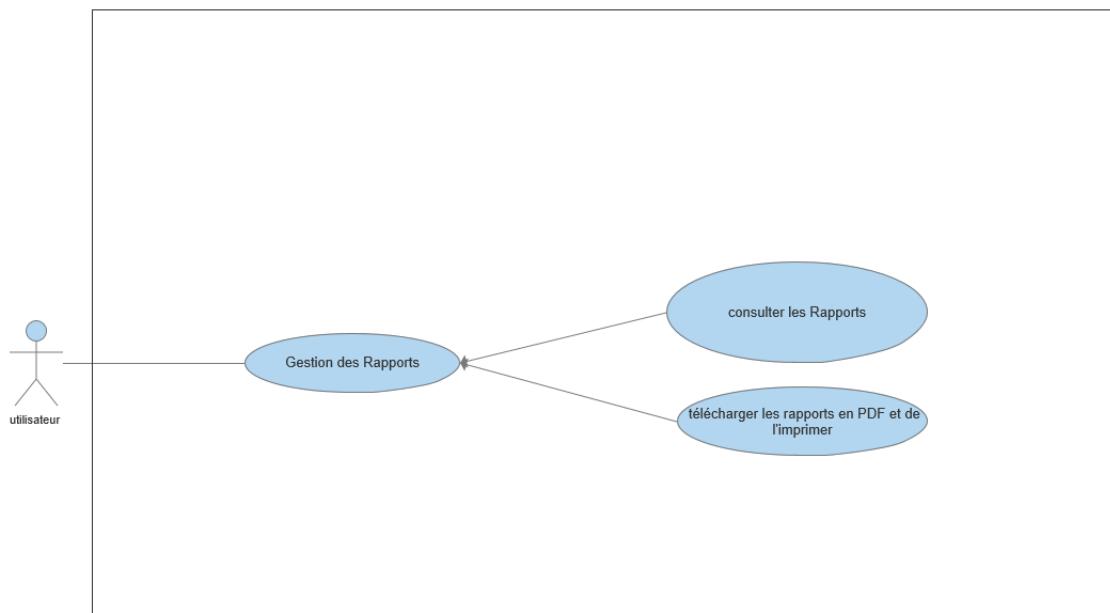


FIGURE 7.3 – Diagramme de raffinement du cas d'utilisation «Gestion des Rapports»

Chapitre 7 : sprint 5 «Paramétrage»

Nous détaillons dans ce qui suit le cas d'utilisation «consulter les Rapports»

Titre	consulter les Rapports
Objectif	L'utilisateur peut consulter les Rapports
Acteur	Administrateur et l'utilisateur
Pré-conditions	-Authentification correcte à l'application. - L'acteur a consulté «Gestion des Rapports».
Déclencheur	L'utilisateur a cliqué sur Rapports
Le scénario nominal	1- Le système affiche l'interface du rapports. 2- L'utilisateur choisit le modèle de rapport (Ventes, Perte de profit, Dépenses, Impôts, Achats) 3- L'utilisateur choisit la date de rapport. 4- L'utilisateur choisit la type de rapport-(par client, par fournisseur ou par produit). 3- L'utilisateur clique sur le bouton "Mettre à jour le rapport". 5- Le système affiche le rapport.
Post-conditions	rapport consulté.

TABLE 7.5 – Description textuelle du CU «consulter les Rapports »

7.3 Conception

Cette partie définit les éléments résultant de l'analyse des spécifications du cinquième sprint. Afin d'en faciliter la réalisation, cette phase de conception sera définie comme la description du sprint à développer

7.3.1 Diagramme de classes participantes

Le diagramme de classes représenté dans la figure 7.4 décrit les associations entre les classes et ceci afin de déterminer les dépendances entre les différentes classes de ce sprint.

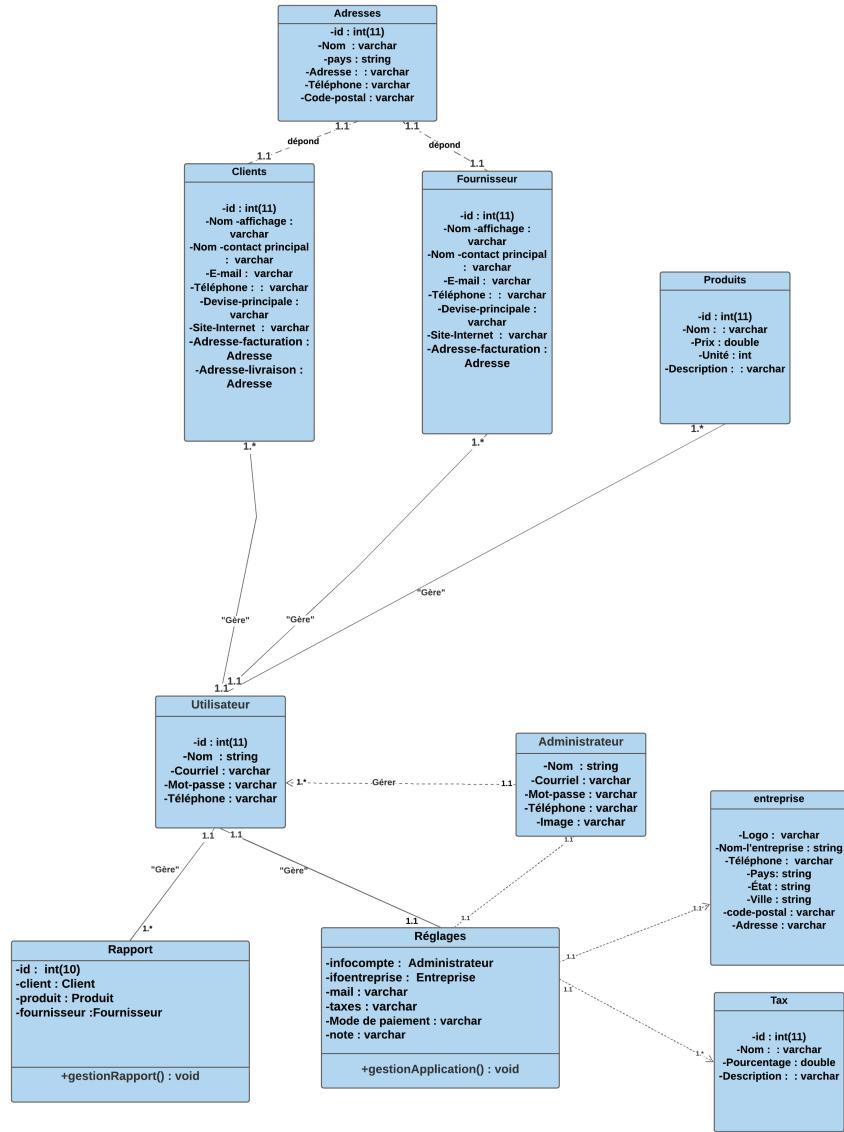


FIGURE 7.4 – Diagramme de classes du cinquième sprint

7.3.2 Diagrammes de séquences

Grâce à ces diagrammes, nous pouvons décrire comment les éléments, qui composent le cinquième sprint, interagissent entre eux et avec les acteurs.

Diagramme de séquences du CU «consulter le Tableau de bord»

Le diagramme présenté dans la figure 7.5 montre l'enchaînement de ce cas d'utilisation.

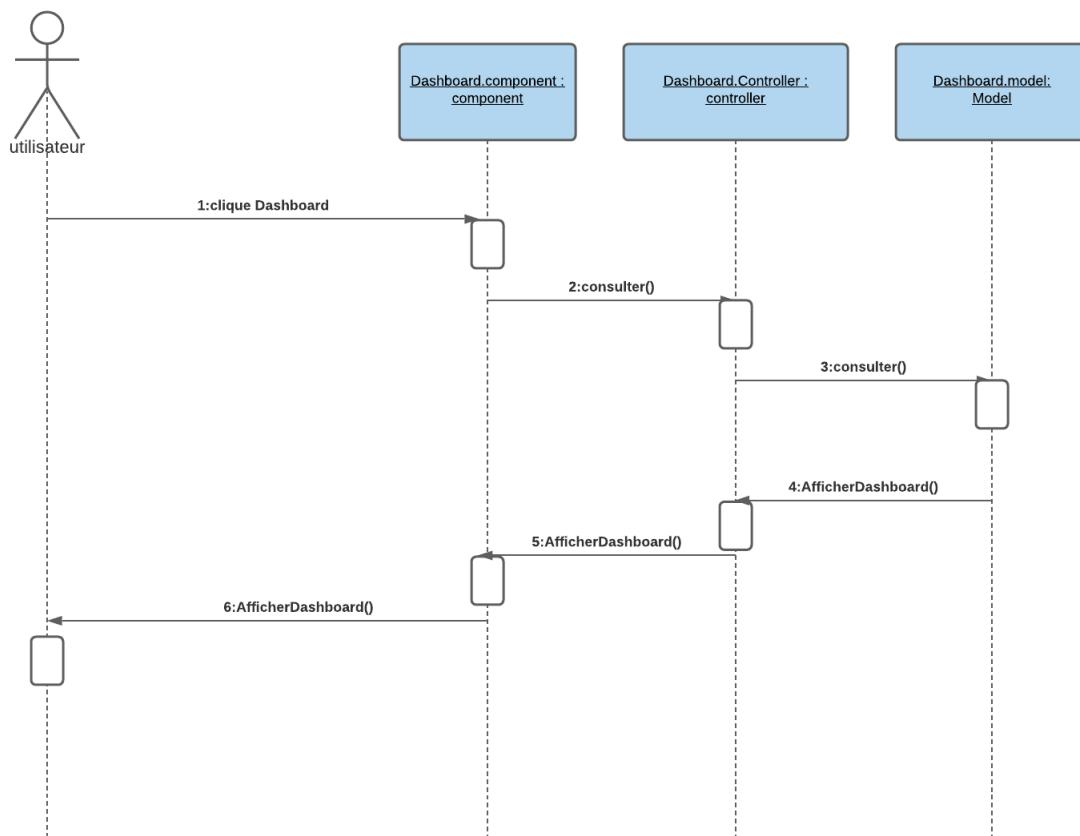


FIGURE 7.5 – Diagramme de séquences « consulter le Tableau de bord»

Diagramme de séquences du CU «Gestion du compte»

La Gestion du compte se fait comme tous les autres options du Gestion du Paramétrage dans le Back Office. Dans ce cas, pour modifier ou remplir les informations du compte, il faut d'abord accéder à l'interface "Paramètres du compte". La figure 7.6 présente un diagramme de séquences qui exprime mieux cette fonctionnalité.

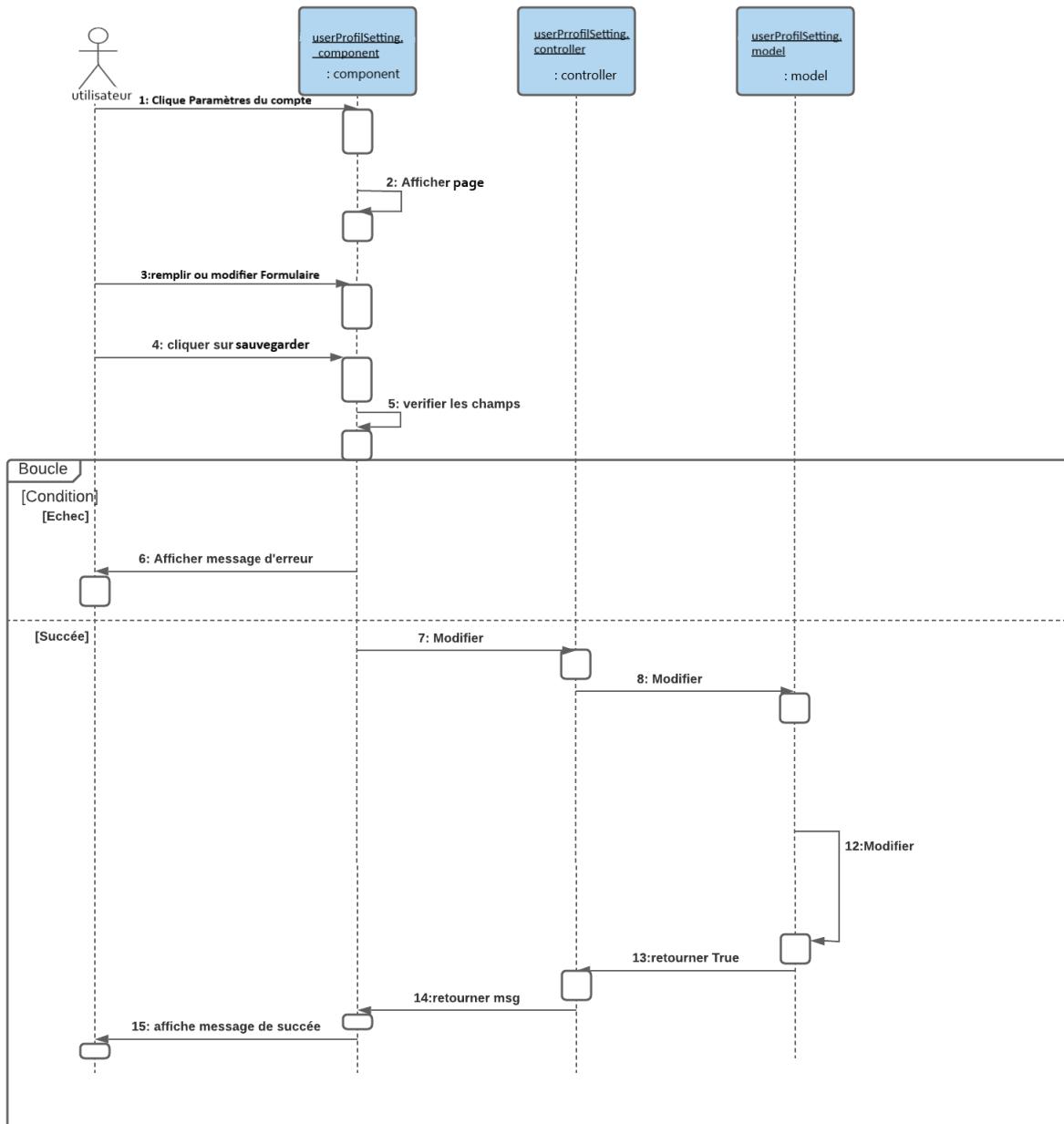


FIGURE 7.6 – Diagramme de séquences « Gestion du compte»

Diagramme de séquences du CU «consulter les Rapport»

Le diagramme présenté dans la figure 7.7 montre l'enchaînement de ce cas d'utilisation.

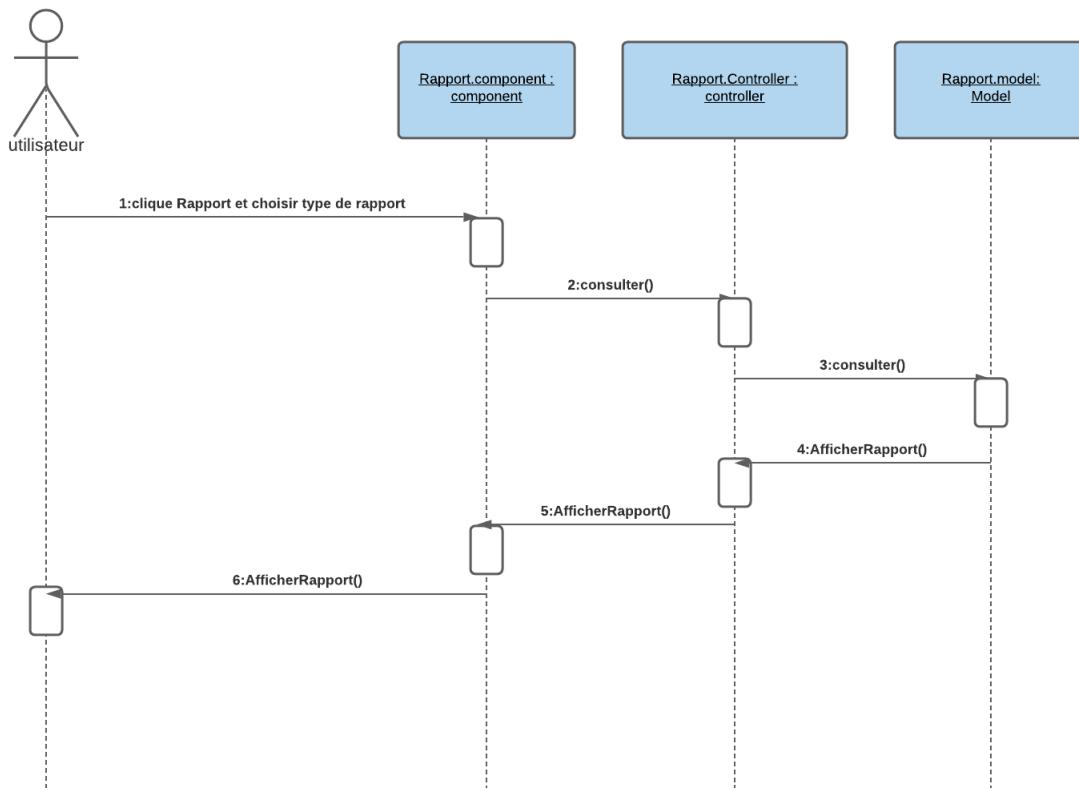


FIGURE 7.7 – Diagramme de séquences «consulter les Rapport»

7.4 Réalisation et tests

Passons maintenant à la présentation de la phase de réalisation du cinquième sprint.

1. consulter le Tableau de bord

La figure 7.8 montre l'interface du Tableau de bord accompagné de toutes les mesures clés du projet : le montant du, nombre des client, nombre des factures, nombre des devises, montant des ventes :

Chapitre 7 : sprint 5 «Paramétrage»

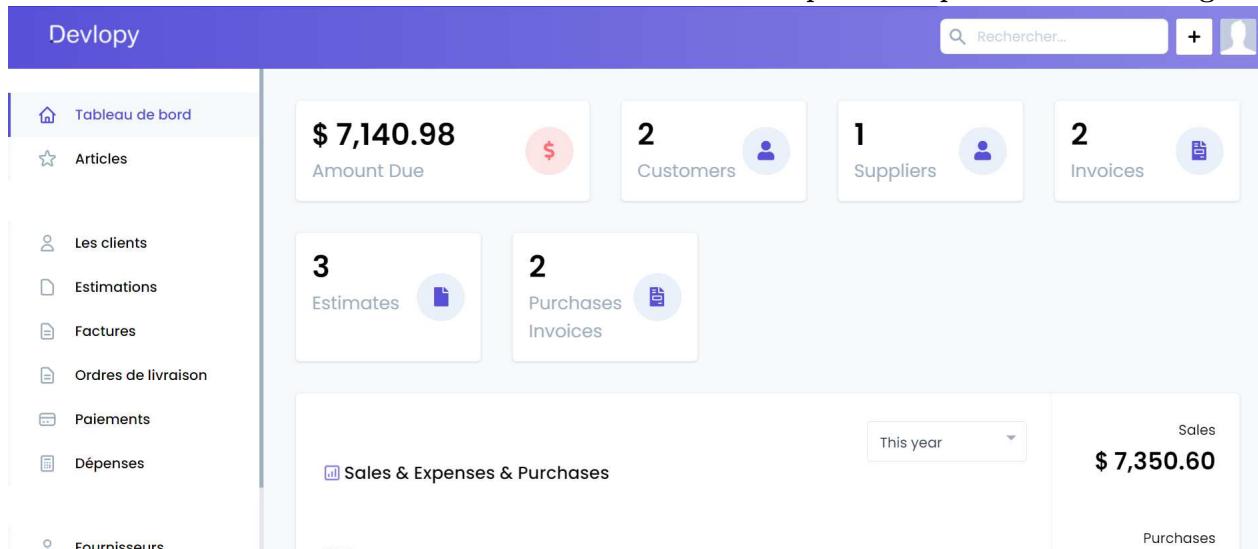


FIGURE 7.8 – consulter le Tableau de bord 1

Ensuite on trouve une courbe qui montre les achats, les ventes , les dépenses, le montant reçu et le Revenu net de cette entreprise

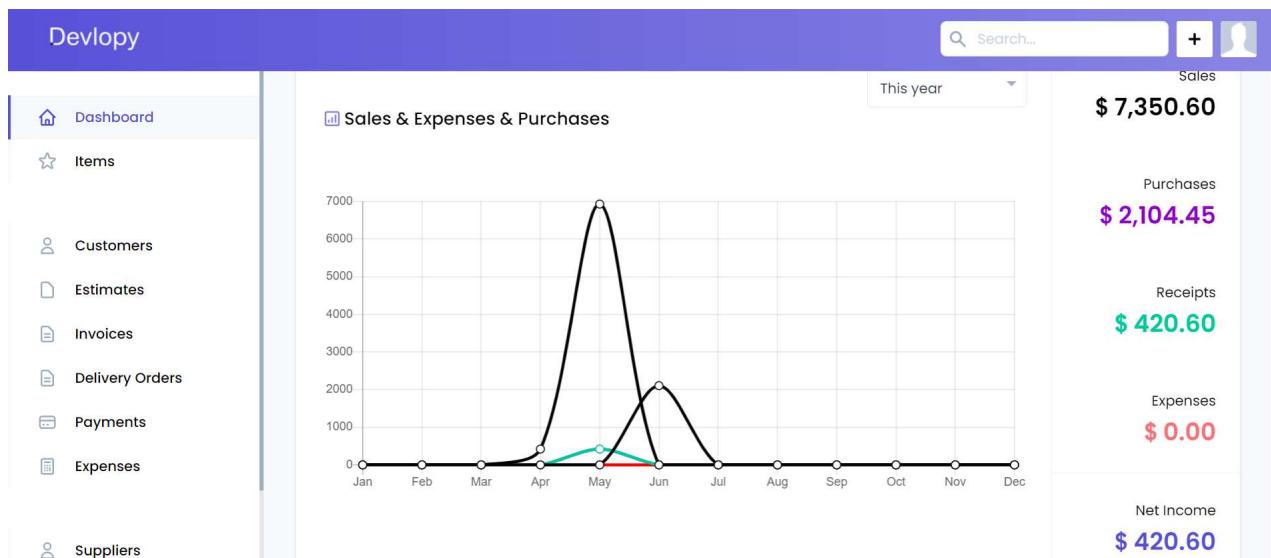


FIGURE 7.9 – consulter le Tableau de bord 2

2.gérer compte

La figure 7.10 illustre l'implantations du cas d'utilisation «gérer compte» :

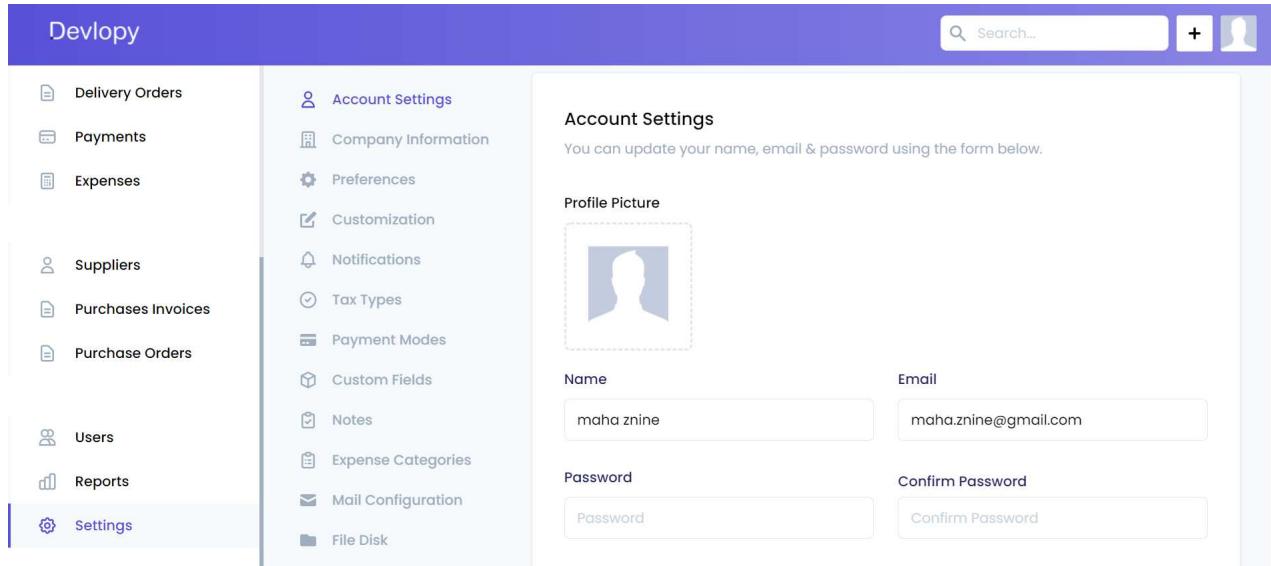


FIGURE 7.10 – gérer compte

3.gérer les informations de l'entreprise

La figure 7.11 montre l'interface «Gérer les informations de l'entreprise» :

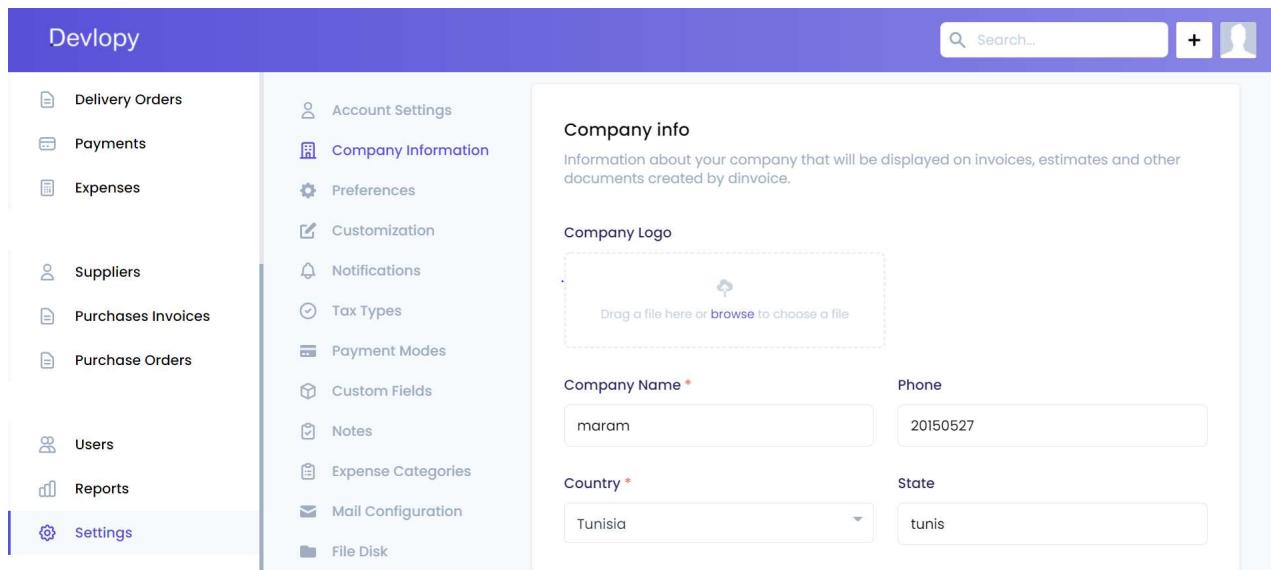


FIGURE 7.11 – Gérer les informations de l'entreprise

4.consulter les Rapports

La figure 7.12 montre l'interface de détails du rapport. En effet, l'administrateur ou l'utilisateur peut choisir le type de rapport,l'imprimer et le télécharger.

FIGURE 7.12 – consulter les Rapports

voila le résultat du rapport choisie : rapport du vente de type "par le Fournisseur".

Date	Purchase ID	Amount
01 Jun 2021	(PINV-00001)	\$1,683.02
06 Jun 2021	(PINV-004)	\$421.43
TOTAL PURCHASES		\$2,104.45

FIGURE 7.13 – Gérer les informations de l'entreprise

7.5 Conclusion

Dans ce chapitre nous avons présenté le dernier sprint du projet, les spécifications fonctionnelles, la conception, le développement , le test et le diagramme de classe générale.

8

Conclusion Générale et Perspectives

Ce projet consiste à réaliser une application web pour la gestion des factures et gestion des dépenses, les paiements et à créer des estimations professionnelles "Dinvoice".

Le présent manuscrit détaille toutes les étapes par lesquelles nous avons passé pour arriver au résultat attendu. Nous avons commencé par la présentation du cadre générale du projet. Par la suite nous avons identifié les besoins et les acteurs afin de garantir une conception fiable. Et nous avons fini par l'implémentassions de l'application réalisée.

Ce projet nous a permis de mettre en application les connaissances théoriques que nous avons pu acquérir au cours des deux années d'études en informatique des système d'information d'entreprise. Ce fut également une opportunité de nous confronter aux difficultés du monde du développement et de la gestion d'un projet.

Finalement, le développement web est un univers très vaste qui ne cesse de s'élargir, nécessitant une veille technologique et une passion pour l'apprentissage. Ce projet a était un déclencheur pour commencer à s'intéresser vraiment à ce domaine, je ne compte pas m'arrêter ici, mais continuer à développer mes compétences et plonger encore et encore dans ce domaine.