
DÉDICACES

“ Je dédie ce travail ..

À ma chère mère "Basma"

Mon bonheur, la source de mes efforts. Aucune dédicace très chère maman, ne pourrait exprimer la profondeur des sentiments que j'éprouve pour vous. Puisse Dieu, tout puissant vous combler de santé, de bonheur et vous procurer une longue vie.

À mon cher père "Adel"

Vous avez toujours été mon école de patience, de confiance et surtout d'espoir et d'amour. J'implore Dieu, tout puissant, de vous accorder une bonne santé, une longue vie et de bonheur.

À mes chers frères "Ayoub" et "Edem"

En signe de l'affection et du grand amour que je vous porte. Puisse Dieu vous éclairer votre route et vous aider à réaliser à votre tour vos vœux les plus chers.

À tous ceux qui me sont chers, à vous tous

Merci.

”

- Eya Ifa

REMERCIEMENTS

Au terme de ce travail, je remercie en premier lieu le **bon Dieu**, tout puissant, de m'avoir donné la force pour dépasser toutes les difficultés.

À ce titre, je remercie mon professeur Monsieur **Adel Khalfallah** pour son écoute, sa motivation et ses conseils précieux tout au long de la rédaction de ce rapport .

Je tiens particulièrement à remercier vivement Monsieur **Med Amine Osman** mon encadrant professionnelle dans la société **APUS-CONSULTING** pour sa patience, sa disponibilité et son partage de ses compétences. C'est grâce à sa confiance que j'ai pu bien élaborer ce projet .

Je souhaite exprimer ma gratitude à toute l'équipe **APUS-CONSULTING** pour leur accueil chaleureux et leur esprit d'équipe.

Mes remerciements les plus sincères à toutes les personnes qui ont contribué de près ou de loin à l'élaboration de ce modeste travail .

Enfin, j'adresse à remercier l'administration et tout le cadre enseignant de mon institut supérieur d'informatique .

TABLE DES MATIÈRES

Introduction générale	1
1 État de l'art	2
Introduction	3
1.1 Organisme d'accueil	3
1.1.1 Objectifs et missions de APUS-CONSULTING	3
1.2 Cadre et contexte du sujet	3
1.2.1 Cadre du sujet	3
1.2.2 Contexte de sujet	4
1.3 Analyse de la concurrence	4
1.3.1 Étude des solutions concurrentes	4
1.3.2 Bilan et synthèse	6
1.4 Solution proposé	7
1.5 Formalisme de modélisation	7
1.6 La méthodologie de travail adoptée	8
1.7 Planification du projet	9
1.7.1 Pilotage du projet avec Scrum	9
1.7.1.1 Équipe projet Scrum	9
1.7.1.2 Backlog du produit	10
1.8 Planification des sprints	15
Conclusion	16

2	Analyse des besoins	17
	Introduction	18
2.1	Capture des besoins	18
2.1.1	Exigences fonctionnelles	18
2.1.2	Exigences non fonctionnelles	19
2.1.3	Identification des acteurs	19
2.1.3.1	Diagramme de cas d'utilisation global	20
2.2	Raffinement des cas d'utilisation	21
2.2.1	Raffinement du CU « Gérer les devises »	21
2.2.2	Raffinement du CU « Gérer les cours devises »	21
2.2.3	Raffinement du CU « Gérer les caisses »	22
2.2.4	Raffinement du CU « Gérer les opérations »	24
2.3	Description des cas d'utilisation	25
2.3.1	Description textuelle du CU «Ajouter devise »	25
2.3.2	Description textuelle du CU «Supprimer devise »	26
2.3.3	Description textuelle du CU «Ouvrir une caisse »	27
2.3.4	Description textuelle du CU «Acheter des devises»	29
2.3.5	Description textuelle du CU «Transfert d'argent d'une caisse à une autre»	30
	Conclusion	31
3	Conception	32
	Introduction	33
3.1	Architecture de la solution	33
3.1.1	Architecture logicielle	33
3.1.2	Le pattern MVC	34
3.2	Conception des interfaces	36
3.2.1	Structure de l'interface modérateur	36
3.2.2	Structure du Dashboard administrateur	37
3.3	Diagramme de classe	37
3.4	Diagrammes de séquence	39
	Conclusion	40

TABLE DES MATIÈRES

4 Réalisation	41
Introduction	42
4.1 Étude et réalisation du Sprint1	42
4.1.1 Backlog du Sprint1	42
4.1.2 Spécification fonctionnelle	44
4.1.2.1 Raffinement des cas d'utilisation du sprint1	44
4.1.3 Conception	46
4.1.3.1 Diagrammes de séquences	46
4.1.3.2 Diagramme de classes du sprint1	47
5 PENTESTING DE LA SOLUTION PROPOSEE	48
Introduction	49
5.1 Pentesting	49
Conclusion générale	50
Références bibliographiques	52

TABLE DES FIGURES

1.1	Logo APUS-CONSULTING	3
1.2	Bureau de change Xchange	5
1.3	Bureau de change ChangeNow	6
1.4	Méthode Scrum	9
1.5	Diagramme de Gant	16
2.1	Démarche adoptée pour la réalisation du modèle de cas d'utilisation	18
2.2	Plan de release	20
2.3	Gérer les devises	21
2.4	Gérer les cours devises	22
2.5	Gérer les caisses	23
2.6	Gérer les opérations	24
3.1	Architecture MVC	33
3.2	Architecture MVC	35
3.3	Architecture Symfony	36
3.4	Architecture Symfony	36
3.5	Structure graphique du Dashboard administrateur	37
3.6	Diagramme de classes d'entité	38
3.7	Diagramme de séquences CU « S'authentifier »	39
3.8	Diagramme de séquences CU « Ajouter un utilisateur »	39

TABLE DES FIGURES

4.1	S'authentifier	44
4.2	Gérer les utilisateurs	45
4.3	Diagramme de séquences CU « S'authentifier »	46
4.4	Diagramme de séquences CU « Ajouter un utilisateur »	46
4.5	Diagramme de séquences CU « Modifier un client »	47
4.6	Diagramme de classes du sprint1	47

LISTE DES TABLEAUX

1.1	Tableau comparatif des solutions concurrentes	6
1.2	Équipe et rôles	9
1.3	Backlog du Projet	10
2.1	Description textuelle du CU «Ajouter devise»	25
2.2	Description textuelle du CU «Ajouter devise»	26
2.3	Description textuelle du CU «Ouvrir une caisse»	28
2.4	Description textuelle du CU «Acheter des devises»	29
2.5	Description textuelle du CU «Transfert d'argent d'une caisse à une autre»	30
4.1	Backlog du Sprint1	42
4.2	Description textuelle du CU « S'authentifier »	44
4.3	Description textuelle du CU «Gérer des utilisateurs»	45

LISTE DES ABRÉVIATIONS

- **CSS** = **Les feuilles de style en cascade**
- **DOM** = **Document Object Model**
- **ERP** = **Progiciel de gestion intégré**
- **GPL** = **General Public License**
- **GRP** = **Gestion des Rencontres Professionnelles**
- **HTML5** = **HyperText Markup Language**
- **HTTP** = **HyperText Transfer Protocol**
- **IDE** = **Integrated Development Environment**
- **ISI** = **Institut Supérieur d'informatique**
- **MVC** = **Modèle-Vue-Contrôleur**

Liste des abréviations

- **MYSQL** = **My Structured Query Language**
- **PHP** = **Hypertext Preprocessor**
- **SGBDR** = **Système de Gestion de Bases de Données Relationnelles**
- **SQL** = **Structured Query Language**
- **UML** = **Le Langage de Modélisation Unifié**
- **W3C** = **World Wide Web Consortium**
- **XML** = **Extensible Markup Language**

INTRODUCTION GÉNÉRALE

CHAPITRE 1

ÉTAT DE L'ART

Introduction	3
1.1 Organisme d'accueil	3
1.2 Cadre et contexte du sujet	3
1.3 Analyse de la concurrence	4
1.4 Solution proposé	7
1.5 Formalisme de modélisation	7
1.6 La méthodologie de travail adoptée	8
1.7 Planification du projet	9
1.8 Planification des sprints	15
Conclusion	16

Introduction

Ce chapitre présente le point de départ de la rédaction du rapport. Nous commençons tout d'abord par la présentation de l'organisme d'accueil. Puis, nous présentons le cadre et le contexte du sujet, quelques solutions de bureau de change. Fortes de ces informations, nous cernons leurs insuffisances afin de projeter une nouvelle solution.

Nous passons par la suite à la présentation de la démarche adoptée pour la modélisation du nouveau système.

1.1 Organisme d'accueil

Créée en septembre 2021 ,APUS-CONSULTING est une entreprise de .. en Tunisie.

Apus Consulting Tunisie.... [1].



FIGURE 1.1 – Logo APUS-CONSULTING

1.1.1 Objectifs et missions de APUS-CONSULTING

APUS-CONSULTING

1.2 Cadre et contexte du sujet

1.2.1 Cadre du sujet

Ce présent travail s'inscrit dans le cadre d'un sujet de fin d'études en vue de l'obtention du diplôme de mastère en Sécurité des Systèmes Informations et d'Infrastructures au de sein de la société APUS-CONSULTING.

Durant ce stage au sein de ladite société, notre mission s'est rapportée essentiellement à **la conception et la réalisation d'une application Web pour la gestion des bureaux de change**

multi-devises. .

1.2.2 Contexte de sujet

Les applications de gestion de change multi-devises se caractérisent par la croissance rapide de l'économie mondiale et le besoin croissant de gérer plusieurs devises pour les transactions internationales. Dans ce cas, il est impératif de disposer d'une solution efficace pour gérer les bureaux de change multidevises afin de traiter les opérations de change avec rapidité et précision, d'optimiser les coûts et de minimiser les risques. Les entreprises opérant sur les marchés internationaux ont besoin de solutions de gestion des équipements pour minimiser l'impact des fluctuations monétaires sur leur résultat net, tandis que les investisseurs et les commerçants ont besoin d'un accès rapide et facile en temps réel aux taux de change pour prendre des décisions éclairées.

1.3 Analyse de la concurrence

L'étude des concurrents est une étape importante dans toute réflexion de développement. L'objectif principal est d'identifier notre positionnement par rapport aux solutions concurrentes les plus répondues.

1.3.1 Étude des solutions concurrentes

Le bureau de change Xchange

Xchange est une application de change qui fournit une solution pratique et sécurisée pour échanger différentes devises. Les utilisateurs peuvent facilement convertir leur argent d'une devise à une autre en utilisant le taux de change en vigueur.

La force de Xchange réside dans les taux de change compétitifs, permettant aux utilisateurs d'obtenir la meilleure valeur lors de l'échange. De plus, les transactions sont généralement rapides, offrant aux utilisateurs une expérience fluide et efficace. La possibilité d'échanger deux devises traditionnelles est un bonus supplémentaire, offrant une plus grande variété d'options de change.

Cependant, certains utilisateurs ont signalé des inconvénients liés à Xchange. Les retards dans le traitement des demandes ont déjà été mentionnés, ce qui peut entraîner de la frustration

pour les utilisateurs et une perte de temps. De plus, la disponibilité de certaines devises peut être limitée, ce qui peut limiter les options pour les utilisateurs souhaitant échanger des devises moins courantes. Dans le cas des problèmes de transaction, certains problèmes ont également été signalés concernant la récupération des fonds, ce qui peut inquiéter les utilisateurs.



FIGURE 1.2 – Bureau de change Xchange

ChangeNow

ChangeNow est une plateforme de conversion de devises numériques qui permet aux utilisateurs d'échanger rapidement et facilement des crypto-monnaies. L'application offre une grande variété de devises numériques à échanger et ne nécessite pas de compte utilisateur pour effectuer des transactions.

De plus, certains utilisateurs ont exprimé des inquiétudes concernant la sécurité de ChangeNow en raison de l'anonymat fourni par la plate-forme. Bien que cela puisse être un avantage pour certains utilisateurs qui souhaitent protéger leur vie privée, cela peut également ouvrir la porte à des activités illégales telles que le blanchiment d'argent.

De plus, certaines personnes ont signalé des problèmes d'utilisation avec certains appareils, ce qui peut être gênant pour ceux qui cherchent à échanger des appareils moins courants. En raison de restrictions légales, cela peut empêcher certains utilisateurs d'utiliser ChangeNow. Malgré ces lacunes, ChangeNow continue de travailler dur pour améliorer ses services et offrir une expérience utilisateur de haute qualité.

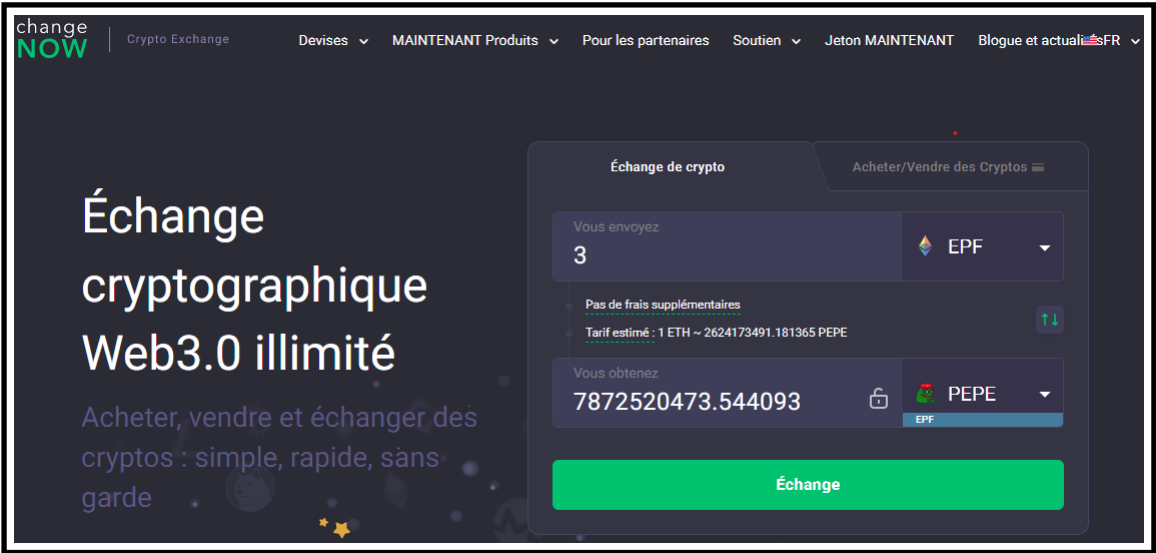


FIGURE 1.3 – Bureau de change ChangeNow

1.3.2 Bilan et synthèse

Comme le montre la tableau 1.1, un récapitulatif autour des solutions concurrentes est présentée. La comparaison se base sur des critères clés (modules) pour bien aider à implémenter notre future solution.

Nous marquons respectivement les sigles ✖ Insatisfaisant, ✔ Satisfaisant.

TABLE 1.1 – Tableau comparatif des solutions concurrentes

Fonctionnalités	Xchange	ChangeNow
Conversion de devises	✔	✔
Taux de change	✔	✔
Variété des devises	✔	✔
Facilité d'utilisation	✔	✖
Sécurité	✔	✔
Anonymat	✖	✔
Support client	✖	✖
Disponibilité	✔	✖

1.4 Solution proposé

Nous envisageons la réalisation d'une solution pour la gestion des rencontres professionnelles, caractérisée par la simplicité et la facilité d'utilisation.

Avec cela, cette solution va permet aux

1.5 Formalisme de modélisation

Nous avons choisi d'utiliser le langage de modélisation unifié **UML** pour spécifier les besoins du projet.

Ce langage fournit une méthode structurée pour décrire, visualiser et spécifier les différents aspects d'un système logiciel. Les diagrammes UML sont utilisés pour représenter les besoins, l'architecture et le comportement du système. Les diagrammes de cas d'utilisation, de classes et de séquence font partie des outils UML couramment utilisés pour décrire respectivement le comportement global du système, la structure de l'application et les interactions entre les éléments du système et les acteurs.

En utilisant UML, il est possible de communiquer efficacement les exigences et la conception du projet entre les membres de l'équipe de développement. Nous vous présenterons ci-dessous les diagrammes qui ont été utilisés pour la conception du projet :

Les différents types de diagrammes

→ **Les aspects fonctionnels** : Qui utilisera le logiciel et pour quoi faire? Comment les actions devront-elles se dérouler? Quelles informations seront utilisées pour cela?

→ **Les aspects liés à l'architecture** : Quels seront les différents composants logiciels à utiliser (base de données, librairies, interfaces, etc.)? Sur quel matériel chacun des composants sera installé?

*** Les besoins des utilisateurs**

- * Le diagramme de cas d'utilisation : représente les fonctionnalités (ou dit cas d'utilisation) nécessaires aux utilisateurs. On peut faire un diagramme de cas d'utilisation pour le logiciel entier ou pour chaque package.

*** L'aspect fonctionnel du logiciel**

- * Le diagramme de classes : Dans la phase d'analyse, ce diagramme représente les entités (des informations) manipulées par les utilisateurs, dans la phase de conception, il représente la structure objet d'un développement orienté objet.
- * Le diagramme de séquence : permet de décrire les différents scénarios d'utilisation du système.

1.6 La méthodologie de travail adoptée

La méthode Scrum

Dans cette section, nous abordons la méthodologie utilisée pour la réalisation du projet, qui est Scrum.

La méthode Scrum suit un processus itératif permettant d'obtenir un produit proche des besoins client en prenant en compte l'évolution de ces derniers et ainsi maximiser la valeur du produit livré.

Cette méthode est articulée autour de trois acteurs clés :

- * le Product Owner : le responsable du produit, il représente les clients et les utilisateurs en transcrivant leurs besoins.
- * le Scrum master : sa mission est de tout mettre en œuvre pour que l'équipe travaille dans les bonnes conditions et se concentre sur l'objectif du projet.
- * l'équipe Scrum : l'équipe de développement, elle est auto-organisée.

Les principaux éléments de Scrum sont :

- * **Sprint** : Il s'agit d'une période de 3 à 4 semaines pendant laquelle une équipe Scrum travaille pour accomplir un ensemble défini de tâches.
- * **Product Backlog** : Il s'agit d'une liste organisée qui contient toutes les fonctionnalités nécessaires pour améliorer le produit. C'est la principale source de travail de l'équipe Scrum.
- * **Planification du sprint** : il s'agit de la réunion où l'équipe Scrum définit ce qui peut être accompli et comment le travail sera effectué pendant le sprint.
- * **Réunions Scrum quotidiennes** : il s'agit de réunions quotidiennes de 15 minutes avec les membres de l'équipe Scrum. Ils sont utilisés pour évaluer les progrès vers les objectifs de sprint et ajuster le backlog de sprint en conséquence.

- * **Sprint Review Meeting** : C'est la réunion où l'équipe Scrum présente les tâches qu'elle a accomplies aux parties prenantes et demande leurs commentaires.

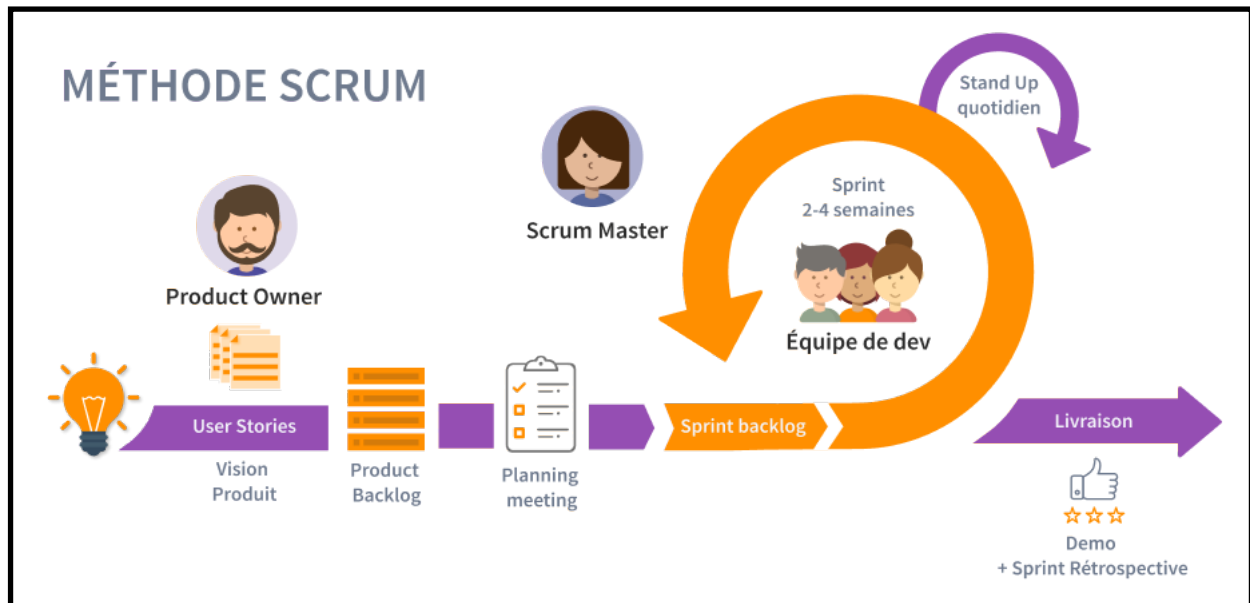


FIGURE 1.4 – Méthode Scrum

1.7 Planification du projet

1.7.1 Pilotage du projet avec Scrum

1.7.1.1 Équipe projet Scrum

Comme le montre le tableau 2.1, voici les différents acteurs impliqués dans la méthodologie Scrum pour le déroulement de ce projet :

TABLE 1.2 – Équipe et rôles

Personne	Rôle	Mission
Scrum Master	Mr Med Amine Osman	<p>→ Animer et faciliter le travail de l'équipe de développement.</p> <p>→ Acquitter de toutes les tâches administratives et s'assurer que la méthodologie Scrum est correctement appliquée.</p>

Product Owner	Mr Med Amine Osman	<ul style="list-style-type: none">→ Participer à l'élaboration des besoins client.→ Porter la vision client du produit final.→ A la charge du backlog produit.→ Décide des priorités à donner aux développements des différentes user stories.
Équipe	Mlle Eya Ifa	<ul style="list-style-type: none">→ Mettre en œuvre les solutions techniques.→ Réaliser les développements.→ Travailler de façon incrémentale.→ Livrer une partie du produit final utilisable et testable à la fin de chaque sprint ou itération.

1.7.1.2 Backlog du produit

Dans le tableau ci-dessous, nous présentons le Backlog qui contient la liste des fonctionnalités attendues pour le produit final. Chaque fonctionnalité est exprimée sous la forme d'une "User Story", qui est une explication générale et informelle du point de vue de l'utilisateur final. Pour chaque User Story, nous spécifions les éléments suivants :

- ID User Story : Le numéro d'identification de la User Story.
- Story Points : Le degré de difficulté de la tâche.
- Priorité : Le niveau de priorité de la User Story.

Les User Stories sont regroupées en modules, représentant chaque catégorie d'exigences fonctionnelles, et numérotées avec leur propre ID.

TABLE 1.3 – Backlog du Projet

Id	Features	User story	Priorité
1	Authentification	En tant qu'administrateur/ utilisateur, je dois me connecter à l'application web.	1
2	Gestion des utilisateurs	En tant qu'administrateur, je veux pouvoir ajouter un nouvel utilisateur avec son nom, son adresse e-mail et son rôle.	1
– Page suivante			

Table 1.3 – Backlog du Produit

Id	Features	User story	Priorité
		En tant qu'administrateur, je veux pouvoir afficher la liste de tous les utilisateurs enregistrés dans le système.	1
		En tant qu'administrateur, je veux pouvoir modifier les informations d'un utilisateur existant, y compris son nom, son adresse e-mail et son rôle.	1
		En tant qu'administrateur, je veux pouvoir supprimer un utilisateur du système.	
		En tant qu'administrateur, je veux pouvoir visualiser les détails d'un utilisateur spécifique, y compris son nom, son adresse e-mail, son rôle et ses permissions.	3
		En tant qu'utilisateur, je veux pouvoir changer mon mot de passe.	3
		En tant qu'utilisateur, je veux pouvoir visualiser mes propres informations de profil.	
		En tant qu'utilisateur, je veux pouvoir recevoir un e-mail de confirmation lors de la création de mon compte.	
		En tant qu'utilisateur, je veux pouvoir me connecter au système à l'aide de mon adresse e-mail et de mon mot de passe.	
9	Gestion des clients	En tant qu'utilisateur, je veux pouvoir filtrer l'historique des opérations par client en utilisant leur numéro de carte d'identité nationale (CIN) ou leur passeport pour identifier rapidement les transactions associées.	1
– Page suivante			

Table 1.3 – Backlog du Produit

Id	Features	User story	Priorité
		En tant que client, je veux pouvoir mettre à jour mes informations personnelles si nécessaire.	
		En tant qu’administrateur, je veux pouvoir gérer la base de données des clients.	
		Le système garantisse la confidentialité et la sécurité des données des clients.	
3	Gestion des caisses	En tant qu’administrateur, je veux pouvoir créer une caisse.	1
		En tant qu’administrateur, je veux pouvoir modifier les informations d’une caisse existante.	
		En tant qu’administrateur, je veux pouvoir supprimer une caisse.	
		En tant qu’administrateur, je veux pouvoir consulter la liste de toutes les caisses.	
		En tant qu’administrateur, je veux pouvoir effectuer une ouverture ou une fermeture de caisse.	
		En tant qu’administrateur, je veux pouvoir consulter l’historique des mouvements d’une caisse.	
		En tant qu’administrateur, j’ai la possibilité de recharger une caisse en demandant des BTC.	
		En tant qu’administrateur, j’ai la possibilité de transférer des devises d’une caisse à une autre.	
4	Gestion des devises	En tant qu’administrateur, je veux pouvoir ajouter une nouvelle devise au système.	1
		En tant qu’administrateur, je veux pouvoir modifier les informations d’une devise existante.	
		En tant qu’administrateur ou modérateur, je veux pouvoir supprimer une devise du système.	

– Page suivante

Table 1.3 – Backlog du Produit

Id	Features	User story	Priorité
		En tant qu'administrateur, je veux pouvoir consulter la liste de tous les devises disponibles.	
		En tant qu'utilisateur, je veux pouvoir consulter les taux de change entre la devise TND et différentes devises.	
		En tant qu'administrateur, je veux pouvoir importer/exporter des données de périphériques à partir de fichiers externes.	
5	Gestion des cours devises	En tant qu'administrateur, je veux pouvoir ajouter un nouveau cours de devise au système.	1
		En tant qu'administrateur, je veux pouvoir gérer les valeurs d'achat et de vente des devises.	
		En tant qu'administrateur, je veux pouvoir modifier les valeurs d'achat et de vente d'un cours de devise existant.	
		En tant qu'administrateur, je veux pouvoir supprimer un cours de devise du système.	
		En tant qu'administrateur, je veux pouvoir consulter la liste de tous les cours de devises disponibles.	
		En tant qu'utilisateur, je veux pouvoir rechercher un cours de devise spécifique en fonction de la devise et de la date.	
		En tant qu'utilisateur, je veux pouvoir consulter l'historique des cours de devise pour une devise spécifique.	
6	Gestion des sociétés	En tant qu'administrateur, je veux pouvoir ajouter une nouvelle société au système.	1
– Page suivante			

Table 1.3 – Backlog du Produit

Id	Features	User story	Priorité
		En tant qu'administrateur, je veux pouvoir modifier les informations d'une société existante.	
		En tant qu'administrateur, je veux pouvoir supprimer une société du système.	
		En tant qu'administrateur, je veux pouvoir consulter la liste de toutes les sociétés enregistrées.	
		En tant qu'administrateur, je veux pouvoir gérer les utilisateurs associés à une société (ajouter, modifier, supprimer des utilisateurs).	
		En tant qu'administrateur, je veux pouvoir attribuer des droits d'accès spécifiques à chaque utilisateur d'une société.	
7	Gestion des historiques	En tant qu'utilisateur, je veux pouvoir consulter l'historique complet des opérations effectuées dans le système.	1
		En tant qu'administrateur, je veux pouvoir gérer les autorisations d'accès à l'historique des opérations, en limitant l'accès aux utilisateurs autorisés.	
9	Gestion des opérations	En tant que client, je veux pouvoir effectuer une opération d'achat de devises	1
		En tant que client, je veux pouvoir effectuer une opération de vente de devises.	
		En tant que client, je veux pouvoir consulter l'historique de mes opérations d'achat, de vente et de transfert.	

– Page suivante

Table 1.3 – Backlog du Produit

Id	Features	User story	Priorité
		En tant qu'administrateur, je veux pouvoir suivre et gérer les opérations effectuées par les clients.	
8	Gestion des statistiques	En tant qu'administrateur, je peux générer les statistiques.	1
		En tant que patient je peux consulter les statistiques.	
13	Gestion des documents	Gérer les documents	5

1.8 Planification des sprints

Le diagramme de **Gantt**, couramment utilisé en gestion de projet, est l'un des outils les plus efficaces pour visualiser l'avancement des différentes activités (tâches) qui composent un projet. La colonne de gauche de la figure répertorie toutes les tâches à effectuer et la ligne d'en-tête représente l'unité de temps la plus appropriée pour le projet (jours, semaines, mois, etc.)

Chaque tâche est représentée par une barre horizontale dont la position et la longueur représentent la date de début, la durée et la date de fin.

Par conséquent, cette image peut vous donner un aperçu :

- Différentes tâches à considérer.

- Les dates de début et de fin de chaque tâche.
- La durée prévue de chaque tâche.
- Le chevauchement possible des tâches et la durée de ce chevauchement.
- Dates de début et de fin pour l'ensemble du projet.

En résumé, un diagramme de Gantt répertorie toutes les tâches à accomplir pour mener le projet à bien, et indique la date à laquelle ces tâches doivent être effectuées (le planning).

Voici le diagramme de Gantt pour notre projet :



FIGURE 1.5 – Diagramme de Gant

Conclusion

Durant ce premier chapitre nous avons présenté un aperçu général sur notre projet, tout en mettant le sujet dans son contexte en introduisant quelques notions sur le domaine décrivant mieux le cadre de travail et faisant la liaison avec l'organisme d'accueil.

Par la suite, nous avons passé à la présentation des méthodologies à utiliser pour nous faciliter la phase de conception dans le suivant chapitre qui est consacré pour l'analyse des besoins et les spécifications.

CHAPITRE 2

ANALYSE DES BESOINS

Introduction	18
2.1 Capture des besoins	18
2.2 Raffinement des cas d'utilisation	21
2.3 Description des cas d'utilisation	25
Conclusion	31

Introduction

Dans ce chapitre, nous débutons tout d'abord par présenter la méthodologie de développement, nous passons par la suite à établir la liste des exigences fonctionnelles et non fonctionnelles du système. Nous énumérons par la suite les différentes étapes de la démarche adoptée afin d'aboutir à la construction du modèle des cas d'utilisation.

Une fois le modèle est prêt, et suite à des négociations avec les différents intervenants du projet (utilisateurs, chefs de projets, développeurs et clients) nous planifions les sprints et nous les priorisons selon les attentes et les besoins du client.

En définitive, nous exposons le plan de la release présentant les différentes séquences de sprints à venir en matière de semaine.

2.1 Capture des besoins

Comme le montre la figure 2.1, nous montrons les étapes nécessaires pour la réalisation du modèle de cas d'utilisation.

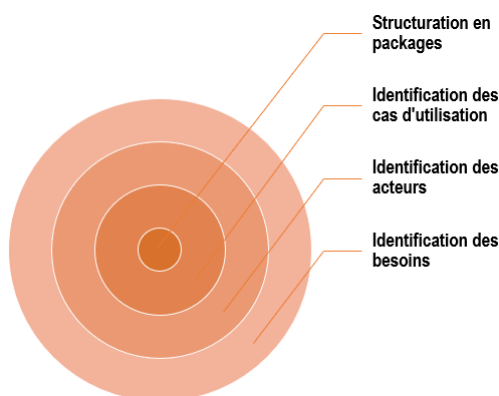


FIGURE 2.1 – Démarche adoptée pour la réalisation du modèle de cas d'utilisation

2.1.1 Exigences fonctionnelles

Nous définissons à travers cette sous-section les fonctionnalités du système à développer, à savoir :

- * Recherche et Découverte :** Les utilisateurs de la plateforme doivent pouvoir effectuer des recherches, qu'il s'agisse des taux de change ou des devises actuelles.
- * Consulter les taux de change :** Les utilisateurs peuvent consulter les taux de change (valeur d'achat et valeur de vente) pour comprendre la valeur d'une devise par rapport à la

devise TND et estimer les coûts associés à la conversion des devises.

- * **S'authentifier** : Les utilisateurs doivent être authentifiés avant d'effectuer des transactions d'achat ou de vente.
- * **Faire des opérations d'achat ou de vente** : Les clients peuvent effectuer des transactions d'achat et de vente de devises contre la devise TND à des fins de voyage, de commerce international ou d'investissement.
- * **Sécurité** : Le système doit assurer l'intégrité des données échangées avec d'autres systèmes. Le système doit garantir la confidentialité des données fournies par les utilisateurs de l'application.

2.1.2 Exigences non fonctionnelles

À part les exigences fonctionnelles, les exigences non-fonctionnelles sont les besoins qui caractérisent les propriétés (qualités) désirées du système telles que :

- * **Exigences de qualité** : Il est important de répondre aux exigences de qualité suivantes :
 - * **Ergonomie sobre et efficace** : Toutes les interfaces doivent être ergonomiques et conviviales. La conception et la présentation de celles-ci doivent être soignées pour ne pas rebuter l'utilisateur.
 - * **Formulaires d'inscription / connexion simple**.
- * **Besoins de disponibilité/fiabilité** : S'assurer que le système doit être toujours disponible sauf pendant les périodes de maintenance.
- * **Exigences de performance** : Prendre en compte des performances d'exécution du système, généralement en termes de temps de réponse : Le chargement d'une page Web, le délai de recherche etc ...

2.1.3 Identification des acteurs

Nous citons dans ce qui suit, les acteurs (entités) qui interagissent avec le système modélisé.

- * **Utilisateur** : C'est un participant au salon qui a le droit de gérer ses choix de rencontres et de consulter les éditions et les fiches des entreprises participantes au salon.
- * **Modérateur** : C'est un participant au salon qui a le droit de gérer ses choix de rencontres et de consulter les éditions et les fiches des entreprises participantes au salon.

- * **Administrateur** : Il s'agit d'une personne chargée de la gestion de toutes les fonctionnalités fournies par l'application via un Dashboard administrateur. Il peut également accéder à l'application normalement comme s'il était un utilisateur normal authentifié.

2.1.3.1 Diagramme de cas d'utilisation global

Pour appréhender les principales fonctionnalités que le système doit offrir ainsi que les acteurs impliqués, nous proposons le diagramme de cas d'utilisation global présenté dans la figure ci-dessous. Cela nous permettra de mieux comprendre les interactions entre les différentes parties prenantes et les différentes fonctionnalités du système.

La figure 2.2 illustre le diagramme de cas d'utilisation global.

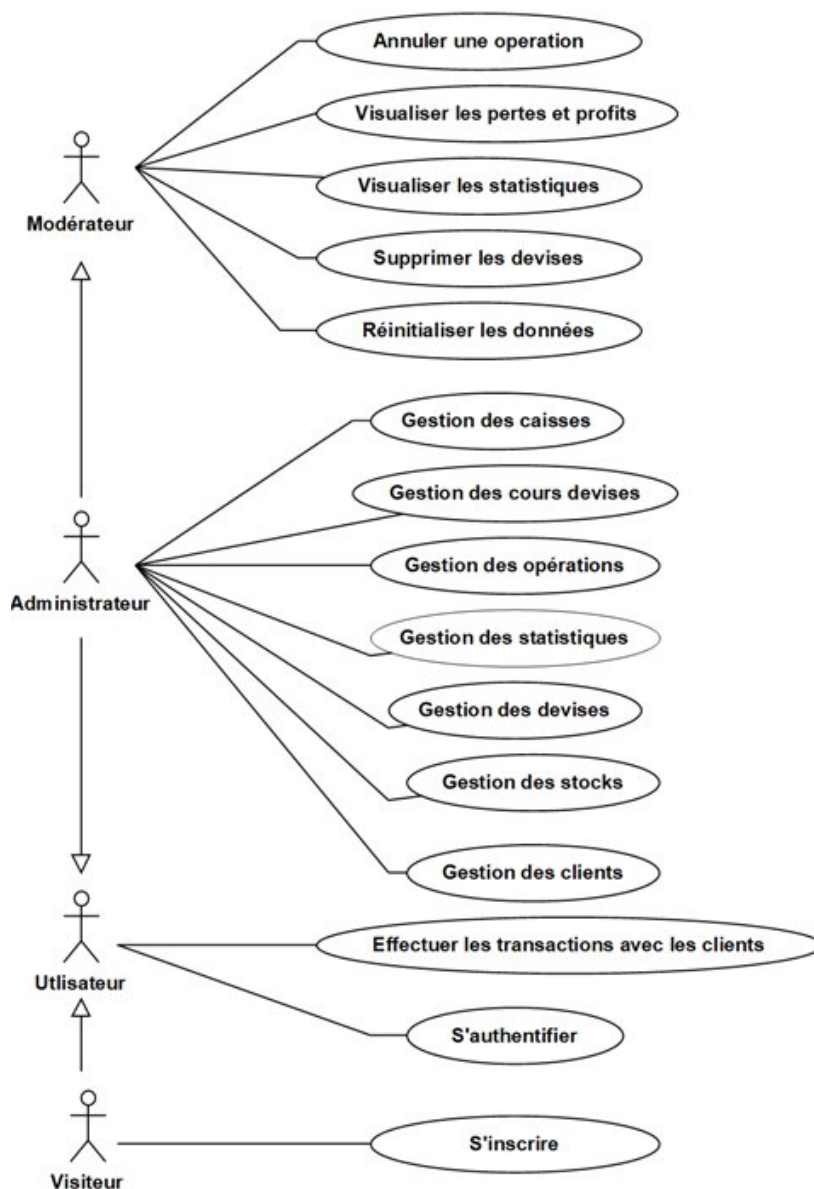


FIGURE 2.2 – Plan de release

2.2 Raffinement des cas d'utilisation

2.2.1 Raffinement du CU « Gérer les devises »

Comme représenté dans le diagramme ci-dessous, l'administrateur dispose d'une interface qui lui permet de gérer les devises. Cette interface lui offre la possibilité de créer de nouvelles devises, supprimer, modifier ou interroger le détail des devises existantes.

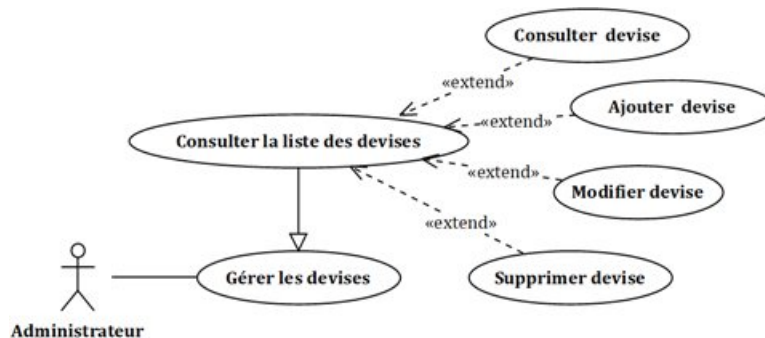


FIGURE 2.3 – Gérer les devises

2.2.2 Raffinement du CU « Gérer les cours devises »

Comme représenté dans le diagramme ci-dessous, l'administrateur dispose d'une interface qui lui permet de gérer les cours de devises. Cependant, il convient de noter que la création de nouveaux cours de devises est conditionnée à la création préalable d'une devise. Cette interface permet à l'administrateur de supprimer, de modifier ou de consulter les détails des cours de devises existants. Chaque devise possède ses propres cours de devises, et l'administrateur est en mesure d'effectuer les opérations CRUD (créer, lire, mettre à jour et supprimer) sur ces cours.

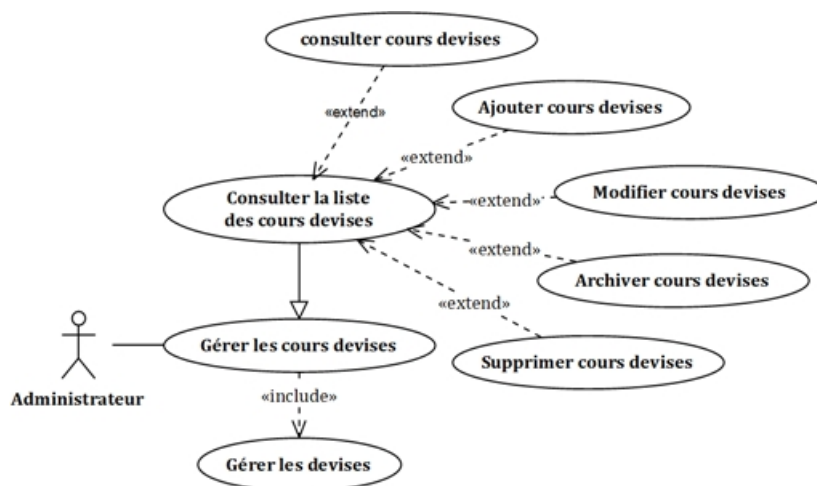


FIGURE 2.4 – Gérer les cours devises

2.2.3 Raffinement du CU « Gérer les caisses »

Comme représenté dans le diagramme ci-dessous, l'administrateur dispose d'une interface lui permettant de gérer les caisses de manière efficace. À travers cette interface, il a la possibilité d'effectuer différentes opérations pour maintenir un contrôle précis sur les caisses. Il peut ajouter de nouvelles caisses afin de répondre aux besoins spécifiques de l'organisation. De plus, il peut consulter les informations détaillées de chaque caisse, permettant ainsi un suivi transparent des transactions et des soldes. L'administrateur a également la capacité d'alimenter les caisses en y ajoutant des fonds supplémentaires ou en retirant des montants, assurant ainsi un équilibre financier adéquat. Si nécessaire, il peut modifier les informations liées aux caisses existantes, comme les noms ou les paramètres spécifiques. De plus, l'administrateur a le pouvoir de supprimer les caisses inutilisées ou obsolètes pour maintenir un environnement de gestion clair et organisé. Enfin, l'interface permet également à l'administrateur de fermer temporairement une caisse ou de l'ouvrir à nouveau, en fonction des besoins opérationnels. Cela offre une flexibilité et une adaptabilité pour gérer les caisses en fonction des situations et des exigences spécifiques de l'organisation.

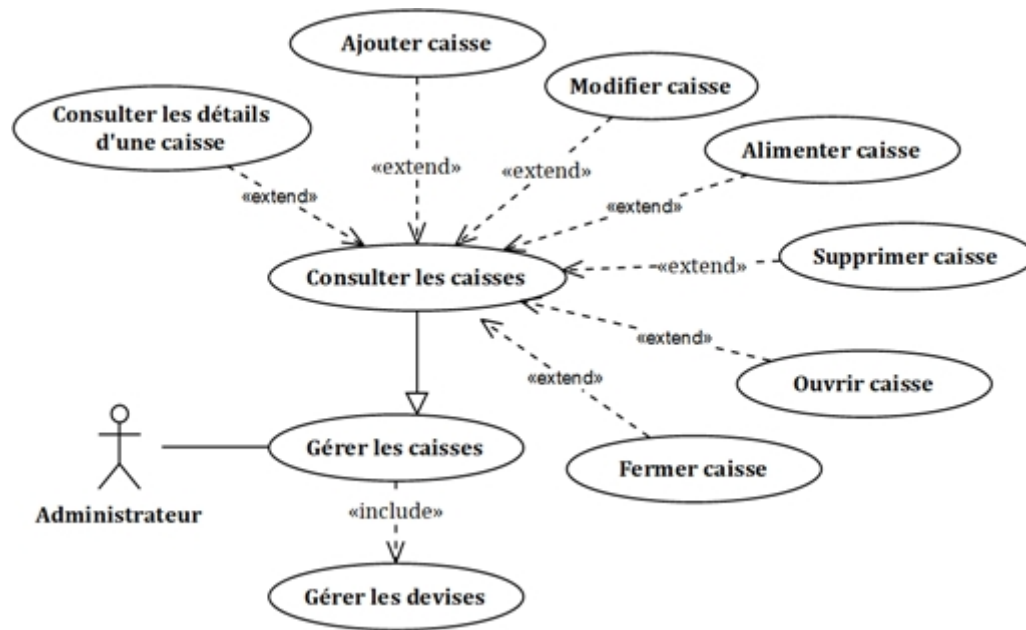


FIGURE 2.5 – Gérer les caisses

2.2.4 Raffinement du CU « Gérer les opérations »

Comme représenté dans le diagramme ci-dessous, L'administrateur dispose des opérations CRUD (créer, lire, mettre à jour, supprimer) suivantes dans la gestion des opérations :

- * **Achats** : L'administrateur peut créer de nouvelles opérations d'achat, consulter les détails des achats existants, mettre à jour les informations des achats, ou supprimer des opérations d'achat enregistrées.
- * **Ventes** : L'administrateur peut créer de nouvelles opérations de vente, consulter les détails des ventes existantes, mettre à jour les informations des ventes, ou supprimer des opérations de vente enregistrées.
- * **Transferts d'une caisse à une autre** : L'administrateur peut effectuer des transferts de fonds d'une caisse à une autre. Il a la possibilité de créer de nouvelles opérations de transfert, consulter les détails des transferts effectués, mettre à jour les informations liées aux transferts, ou supprimer des opérations de transfert enregistrées.
- * **Alimentation d'une caisse** : L'administrateur peut ajouter des fonds à une caisse spécifique. Il peut créer de nouvelles opérations d'alimentation de caisse, consulter les détails des opérations d'alimentation, mettre à jour les informations relatives à ces opérations, ou supprimer des opérations d'alimentation enregistrées.

L'administrateur a donc le contrôle total sur ces opérations et peut les effectuer selon les besoins et les exigences de la gestion des opérations.

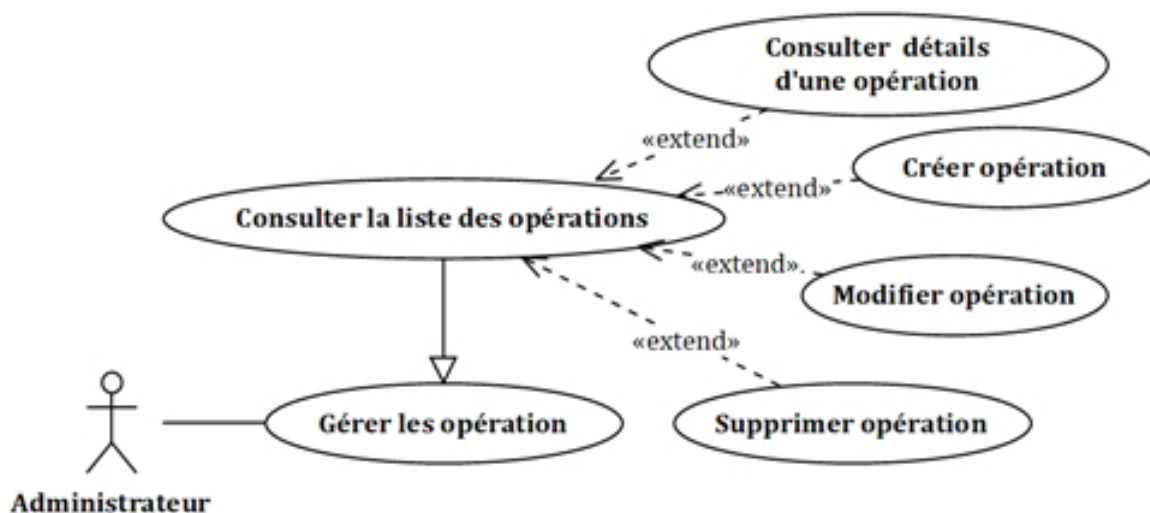


FIGURE 2.6 – Gérer les opérations

2.3 Description des cas d'utilisation

2.3.1 Description textuelle du CU «Ajouter devise»

TABLE 2.1 – Description textuelle du CU «Ajouter devise»

Nom	Ajouter devise
Acteur	Administrateur
Pré-conditions	L'administrateur s'est authentifié sur le système;
Description	L'administrateur ajoute une nouvelle devise à travers l'interface de gestion des devises du système. Cette fonctionnalité permet au système de prendre en charge une nouvelle devise.
S.Nominal	<ul style="list-style-type: none">❶ L'administrateur accède à l'interface de gestion des devises du système;❷ Le système affiche l'interface;❸ L'administrateur sélectionne l'option "Ajouter une devise" dans le menu principal.;❹ Le système affiche un formulaire vide pour la saisie des informations de la nouvelle devise;❺ L'administrateur remplit les champs du formulaire avec les détails de la devise;❻ L'administrateur clique sur le bouton « Ajouter » pour confirmer l'action;❼ Le système effectue les vérifications nécessaires;❽ Les informations vérifiées sont valides, le système enregistre la nouvelle devise dans sa base de données;❾ La page qui contient la liste des devises est rafraîchie.

S.Alternatif	<ul style="list-style-type: none">❶ L'administrateur sélectionne l'option «Ajouter» ;❷ le système détecte les champs manquants et affiche un message d'erreur ;❸ Le système indique à l'administrateur les champs obligatoires manquants et lui demande de les compléter ;❹ L'administrateur corrige les champs manquants et soumet à nouveau le formulaire.❺ Le système procède à l'ajout de la devise une fois que toutes les informations requises sont fournies.
---------------------	--

2.3.2 Description textuelle du CU «Supprimer devise »

TABLE 2.2 – Description textuelle du CU «Ajouter devise»

Nom	Supprimer devise
Acteur	Administrateur, Modérateur
Pré-conditions	L'administrateur ou le modérateur s'est authentifié sur le système ;
Description	L'administrateur ou le modérateur supprime une devise à travers l'interface de gestion des devises du système. Cette fonctionnalité permet au système de prendre en charge une nouvelle devise.

S.Nominal	<ul style="list-style-type: none">❶ L'administrateur ou le modérateur accède à l'interface de gestion des devises du système;❷ Le système affiche l'interface;❸ L'administrateur ou le modérateur consulte la liste des devises existantes et sélectionne la devise qu'il souhaite supprimer;❹ le système vérifie s'il existe des cours de devises associés à cette devise;❺ Si des cours de devises sont associés à la devise sélectionnée, le scénario alternatif "Cours de devises existants" est déclenché;❻ Si aucune donnée n'est associée à la devise ou si l'administrateur/le modérateur confirme la suppression malgré l'existence de cours de devises, le scénario principal se poursuit;❼ Le système supprime la devise sélectionnée de la liste des devises disponibles;❽ Si des cours de devises étaient liés à la devise supprimée, le système met à jour les enregistrements pour refléter la suppression de la devise;❾ La page qui contient la liste des devises est rafraîchie.
S.Alternatif	<ul style="list-style-type: none">❶ Le système identifie des cours de devises associés à la devise sélectionnée pour la suppression;❷ Le système affiche un message d'avertissement indiquant que des cours de devises sont associés à la devise et qu'elle ne peut pas être supprimée tant que ces données sont présentes;❸ L'administrateur ou le modérateur choisit l'une des options suivantes<ul style="list-style-type: none">(a) Supprimer les cours de devises liés à la devise avant de procéder à la suppression.(b) Annuler la suppression de la devise et retourner à l'état initial. <p>puis effectue les actions appropriées pour gérer les cours de devises;</p> ❹ L'administrateur/le modérateur peut réessayer la suppression de la devise, une fois que les cours de devises ont été traités conformément à l'option choisie.

2.3.3 Description textuelle du CU «Ouvrir une caisse »

TABLE 2.3 – Description textuelle du CU «Ouvrir une caisse»

Nom	Ouvrir une caisse
Acteur	Administrateur
Pré-conditions	L'administrateur s'est authentifié sur le système;
Description	L'administrateur active une caisse et autorise les transactions associées. Cette fonctionnalité permet au système de prendre en charge l'activation de la caisse, lui permettant ainsi de traiter les transactions.
S.Nominal	<ul style="list-style-type: none">❶ L'administrateur accède à l'interface de gestion des caisses du système;❷ Le système affiche l'interface;❸ L'administrateur sélectionne la caisse spécifique qu'il souhaite ouvrir;❹ L'administrateur clique sur l'option "Ouvrir la caisse" associée à la caisse sélectionnée;❺ Le système active la caisse et la marque comme "ouverte", permettant ainsi les transactions associées;❻ Le système affiche un message de confirmation à l'administrateur, indiquant que la caisse a été ouverte avec succès;❼ Les utilisateurs autorisés peuvent maintenant effectuer des transactions et accéder à la caisse pour les opérations appropriées.
S.Alternatif	<ul style="list-style-type: none">❶ L'administrateur sélectionne la caisse qu'il souhaite ouvrir;❷ Le système détecte que la caisse est déjà ouverte;❸ Le système affiche un message d'erreur à l'administrateur, l'informant que la caisse est déjà active;❹ L'administrateur doit vérifier l'état de la caisse ou choisir une autre caisse à ouvrir si nécessaire.

2.3.4 Description textuelle du CU «Acheter des devises»

TABLE 2.4 – Description textuelle du CU «Acheter des devises»

Nom	Acheter des devises
Acteur	Administrateur
Pré-conditions	L'administrateur s'est authentifié sur le système;
Description	L'administrateur utilise cette effectue des opérations achats de devises pour les clients. Cette fonctionnalité permet au système de prendre en charge l'enregistrement de l'opération et ces daitalis dans l'interface des opérations.
S.Nominal	<p>❶ L'administrateur accède à l'interface d'achat de devises et le système affiche l'interface;</p> <p>❷ L'administrateur remplit le formulaire d'achat contenant les informations du client;</p> <p>❸ L'administrateur sélectionne la devise du client parmi les options disponibles qu'il souhaite échanger avec les dinars tunisiens;</p> <p>❹ L'administrateur choisit la caisse à partir de laquelle il va retirer l'argent nécessaire pour l'achat de devises;</p> <p>❺ L'administrateur saisit le montant en devise étrangère que le client souhaite échanger;</p> <p>❻ Le système utilise le cours de change actuel pour convertir le montant saisi en dinars tunisiens;</p> <p>❼ Le système affiche le montant converti et le cours de change actuel à l'administrateur avant la validation de l'opération;</p> <p>❽ Le client est consulté pour décider s'il souhaite poursuivre ou annuler l'opération en fonction des informations affichées;</p> <p>❾ Si le client souhaite poursuivre, l'administrateur valide l'opération d'achat de devises.</p> <p>❿ Le système enregistre les détails de l'opération d'achat, met à jour les soldes de la caisse et génère un récapitulatif pour le client.</p>

S.Alternatif	<p>❶ L'administrateur accède à l'interface d'achat de devises et remplit le formulaire avec les informations du client;</p> <p>❷ L'administrateur sélectionne la devise du client et la caisse de retrait de l'argent;</p> <p>❸ L'administrateur saisit le montant en devise étrangère que le client souhaite échanger;</p> <p>❹ Le système affiche le montant converti en dinars tunisiens et le cours de change actuel;</p> <p>❺ Le client décide d'annuler l'opération d'achat pour une raison quelconque;</p> <p>❻ L'administrateur sélectionne l'option d'annulation de l'opération;</p> <p>❼ Le système annule la transaction en cours, n'enregistre aucune modification et retourne à l'état initial.</p>
---------------------	--

2.3.5 Description textuelle du CU «Transfert d'argent d'une caisse à une autre»

TABLE 2.5 – Description textuelle du CU «Transfert d'argent d'une caisse à une autre»

Nom	Transfert d'argent d'une caisse à une autre
Acteur	Administrateur
Pré-conditions	L'administrateur s'est authentifié sur le système;
Description	L'administrateur active une caisse et autorise les transactions associées. Cette fonctionnalité permet au système de prendre en charge l'activation de la caisse, lui permettant ainsi de traiter les transactions.

S.Nominal	<ul style="list-style-type: none">❶ L'administrateur reçoit une réclamation concernant le solde insuffisant dans une caisse spécifique;❷ l'administrateur accède à l'interface de gestion des caisses et consulte les informations relatives à la caisse en question;❸ L'administrateur vérifie si une autre caisse dispose d'un solde suffisant pour effectuer un transfert d'argent vers la caisse déficitaire;❹ L'administrateur sélectionne la caisse source à partir de laquelle l'argent sera transféré;❺ L'administrateur saisit le montant à transférer, qui ne peut pas dépasser le solde disponible dans la caisse source.;❻ l'administrateur choisit la caisse destinataire où le montant sera ajouté;❼ L'administrateur vérifie les informations saisies et valide le transfert;❽ Le système effectue les modifications nécessaires en soustrayant le montant transféré de la caisse source et en l'ajoutant à la caisse destinataire;❾ Le système affiche un message de confirmation indiquant que le transfert a été effectué avec succès;❿ Le système enregistre les détails de l'opération de transfert, met à jour les soldes des caisses.
S.Alternatif	<ul style="list-style-type: none">❶ Lors de la vérification des caisses, l'administrateur constate qu'aucune caisse ne dispose d'un solde suffisant pour effectuer le transfert d'argent requis;❷ L'administrateur informe le demandeur de la réclamation de la situation et propose des solutions alternatives, telles que recharger la caisse déficitaire par d'autres moyens;❸ Le système n'effectue aucun transfert d'argent et retourne à l'état initial.

Conclusion

Dans ce chapitre, nous avons pu analyser et spécifier les besoins du système. Nous avons identifié les différents acteurs du système d'une part, les tâches de chacun, d'autre part et éventuellement leurs associer des priorités.

Nous passons maintenant à présenter l'environnement logiciel et matériel, ainsi que l'architecture adopté pour la réalisation de la solution.

CHAPITRE 3

CONCEPTION

Introduction	33
3.1 Architecture de la solution	33
3.2 Conception des interfaces	36
3.3 Diagramme de classe	37
3.4 Diagrammes de séquence	39
Conclusion	40

Introduction

Dans ce chapitre nous allons décrire notre environnement logiciel ainsi que l'environnement matériel tout en mentionnant l'architecture logicielle et les choix techniques utilisés.

3.1 Architecture de la solution

3.1.1 Architecture logicielle

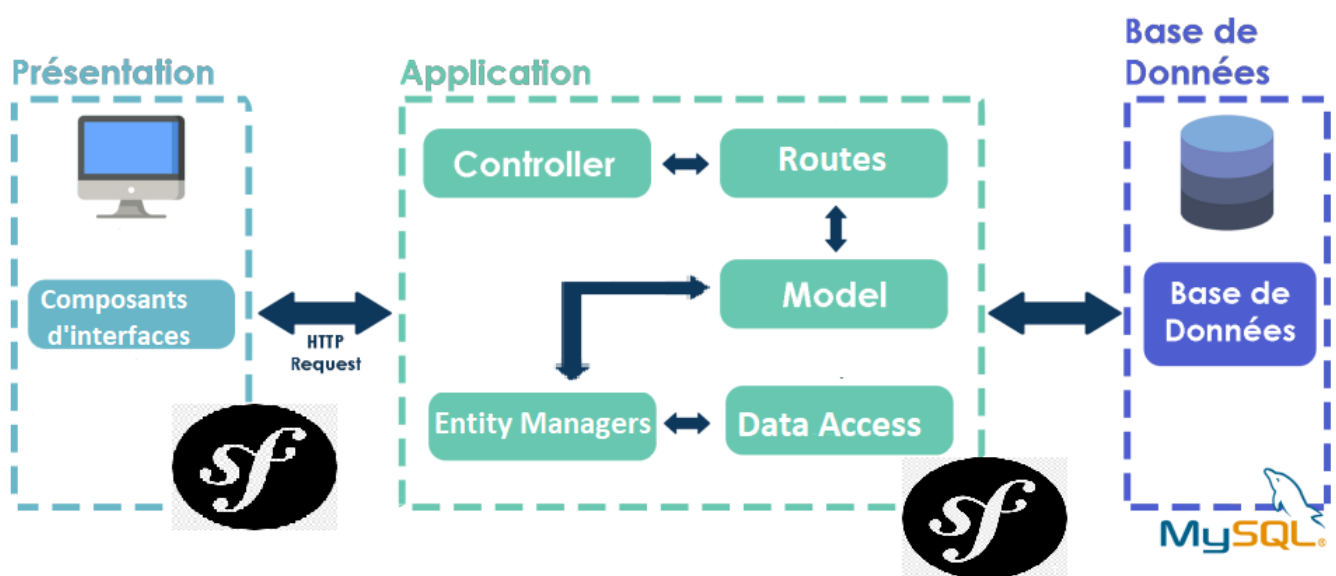


FIGURE 3.1 – Architecture MVC

- **La couche Présentation** : Cette couche, également connue sous le nom de "Front End", représente la partie de notre solution qui s'exécute côté client, c'est-à-dire dans un navigateur Web. Elle est développée à l'aide du framework Symfony. Toutes les interfaces d'interaction avec l'utilisateur se trouvent dans cette couche.

Chaque composant d'interface est créé indépendamment, ce qui le rend hautement modulaire et réutilisable. Chaque composant représente une partie spécifique de l'interface utilisateur.

Pour communiquer avec la couche Application, qui est responsable du traitement des requêtes et de la gestion de la logique métier, la couche Présentation utilise HTTP REQUEST. Ces requêtes sont envoyées depuis un navigateur Web vers un serveur et contiennent des informations telles que l'URL de la requête, la méthode HTTP (GET, POST, etc.) et éventuellement d'autres données dans le corps de la requête.

- **La couche Application** : Cette couche, également connue sous le nom de "Back End", est la partie serveur de notre solution. Elle est composée d'une application Symfony qui s'exécute côté serveur. Cette couche est responsable du traitement des requêtes reçues, de l'accès aux données et de la gestion de la logique métier.

Les principales composantes de l'application dans cette couche sont les suivantes :

Contrôleurs (Controllers) : Les contrôleurs sont responsables de la réception des requêtes HTTP provenant de la couche Présentation. Ils traitent ces requêtes, effectuent les opérations nécessaires et renvoient les réponses appropriées.

Routes (Routes) : Les routes définissent les URL et les actions correspondantes dans les contrôleurs.

Modèles (Models) : Les modèles représentent les entités et les objets métier de notre solution. Ils définissent la structure des données et les règles de validation associées. Les modèles sont généralement liés à la couche de base de données pour accéder et manipuler les données.

Gestionnaires d'entités (Entity Managers) : Les gestionnaires d'entités sont responsables de la gestion des opérations de persistance des entités dans la base de données. Ils fournissent des méthodes pour créer, lire, mettre à jour et supprimer des entités.

Accès aux données (Data Access) : Cette composante gère l'accès aux données, notamment par le biais de requêtes SQL.

- **La base de données** : Cette couche est utilisée pour stocker les données de l'application. Dans notre solution, nous la mettons en œuvre à l'aide d'un système de gestion de base de données relationnelle comme MySQL.

Dans le but de mieux appréhender les différents composants de notre projet, nous allons consacrer cette partie à expliquer l'architecture que nous avons adoptée lors de sa réalisation.

3.1.2 Le pattern MVC

MVC (Modèle-Vue-Contrôleur) est un modèle architectural qui permet d'organiser l'interface utilisateur, les processus de traitement et les données d'une application logicielle. Ce modèle de conception définit une séparation claire en trois couches, comme le montre le schéma ci-dessous.

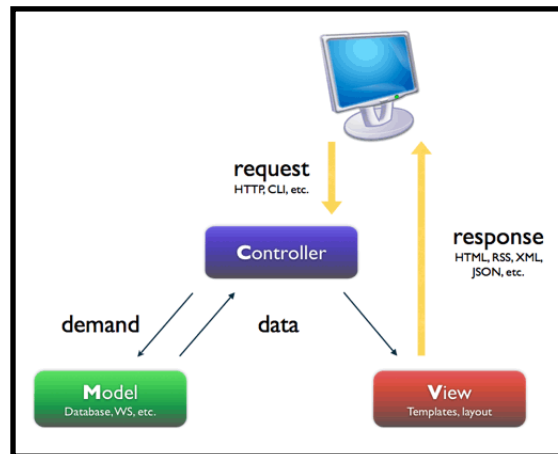


FIGURE 3.2 – Architecture MVC

*** Un modèle (Model)**

C'est la partie qui se concentre sur l'affichage. Elle ne fait presque aucun calcul et se contente de récupérer des variables pour savoir ce qu'elle doit afficher. On y trouve essentiellement du code HTML. (C'est par celle-ci que l'utilisateur interagit avec l'application.)

*** Une vue (View)**

C'est la partie qui gère les données du site. Son rôle est d'aller récupérer les informations dans la base de données, de les organiser et de les assembler pour qu'elles puissent ensuite être traitées par le contrôleur.

*** Un contrôleur (Controller)**

Cette partie désigne l'intermédiaire entre le modèle et la vue : le contrôleur va demander au modèle les données, les analyser, prendre des décisions et renvoyer les informations à afficher à la vue.

Nous allons maintenant parler de l'Architecture de Symfony qui est basée sur les principales entités du framework participant au traitement d'une requête : le Kernel et le Service Container.

*** Le Kernel**

Est le cœur du framework. C'est un composant interne et il n'est pas obligatoire de connaître son existence pour développer sous Symfony. Pourtant, comprendre son fonctionnement et savoir écouter les signaux qu'il émet est des plus incontestable pour bénéficier pleinement des capacités offertes par le framework.

*** Le Service Container**

Le Service Container est un composant indispensable, c'est une sorte de boîte à outils, dans

laquelle vous trouverez ce qu'on appelle des « services ». Ces services sont divers et variés. Certains vous permettront de requêter des bases de données. Vous aurez même le droit de créer vos propres services .

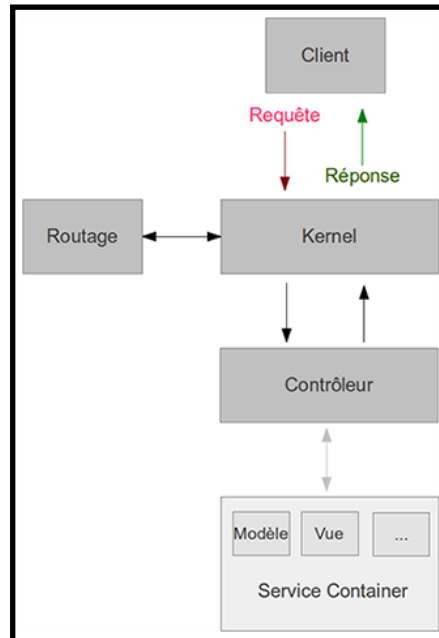


FIGURE 3.3 – Architecture Symfony

3.2 Conception des interfaces

3.2.1 Structure de l'interface modérateur

La figure ci-dessous illustre la structure de la page principale de notre application lorsque le modérateur est authentifié. Elle met en évidence les différentes composantes et fonctionnalités disponibles pour le modérateur. Cela inclut les sections spécifiques aux tâches de modération. Le diagramme permet de visualiser clairement l'interface et les fonctionnalités accessibles au modérateur une fois connecté à l'application.



FIGURE 3.4 – Architecture Symfony

3.2.2 Structure du Dashboard administrateur

L'administrateur dispose d'une interface distincte qui diffère complètement de celle des autres utilisateurs. Le diagramme ci-dessous présente la structure du tableau de bord administrateur, mettant en évidence les différentes fonctionnalités et sections qui lui sont spécifiques. Cela permet à l'administrateur d'avoir un contrôle total et des fonctionnalités supplémentaires par rapport aux autres utilisateurs.

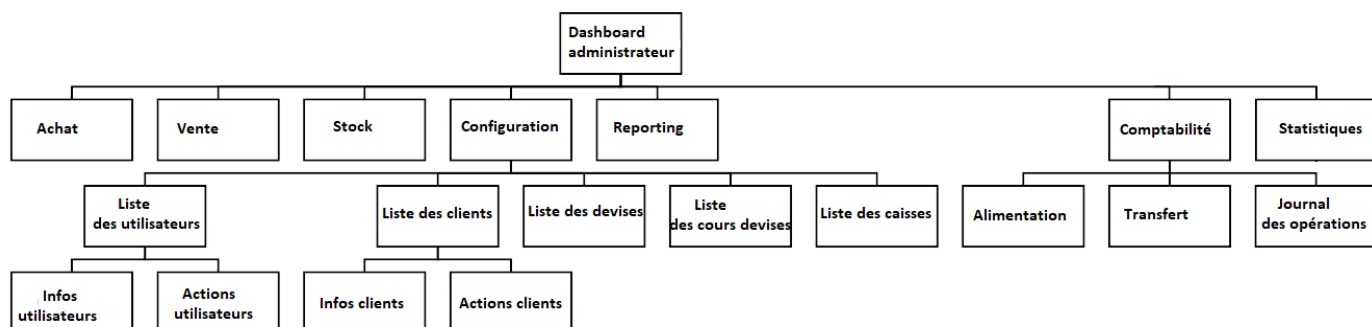


FIGURE 3.5 – Structure graphique du Dashboard administrateur

3.3 Diagramme de classe

Dans cette section, nous abordons la conception essentielle de notre application en utilisant un diagramme de classe. Cela nous permettra d'avoir une meilleure compréhension du mécanisme d'implémentation de l'application. Le diagramme de classe nous permettra de visualiser les relations entre les différentes entités, les attributs clés de chaque classe, ainsi que les associations et les cardinalités entre elles. Cela nous aidera à organiser et structurer notre code de manière efficace, en prenant en compte les dépendances et les interactions entre les différentes parties de notre application.

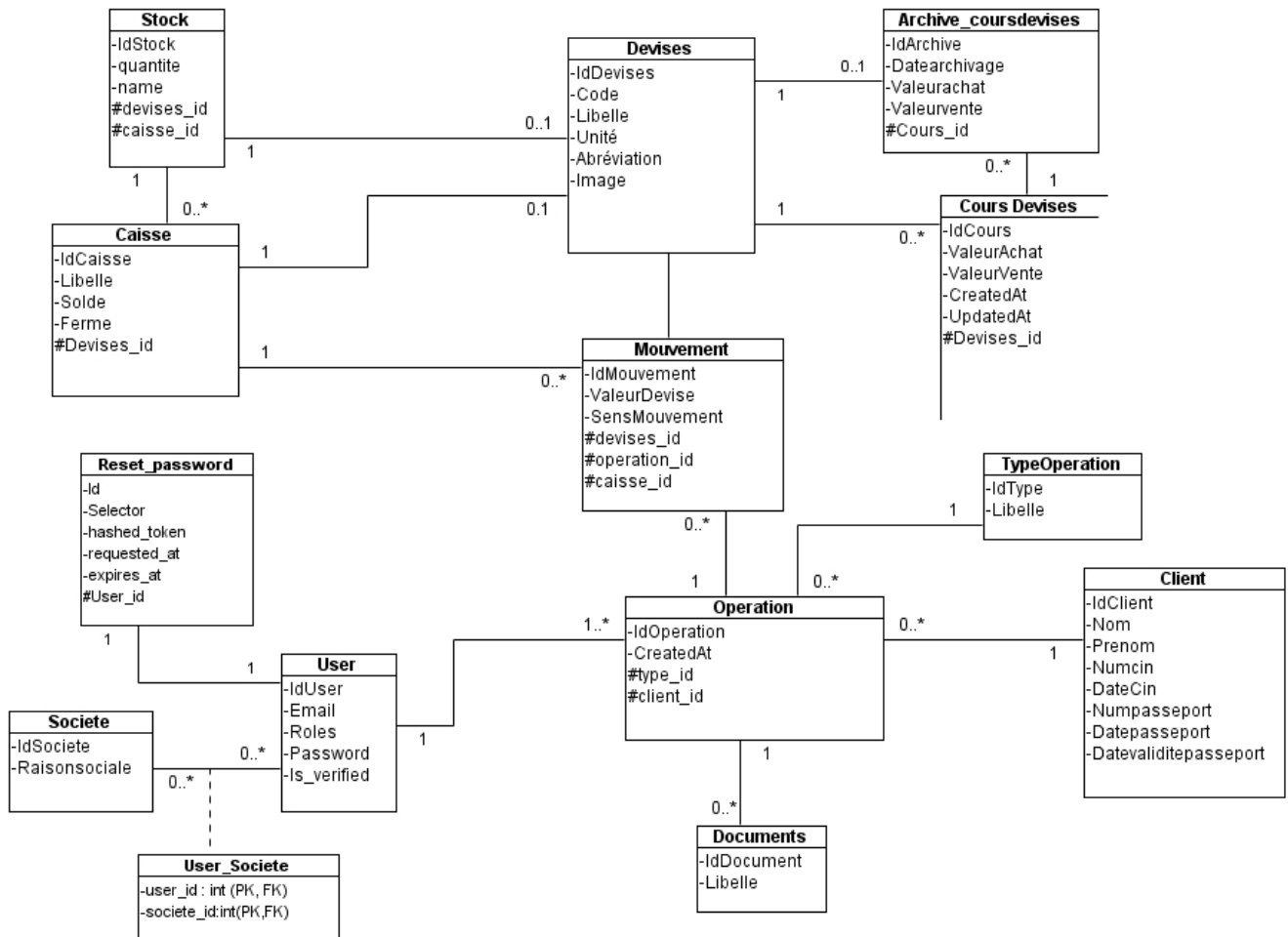


FIGURE 3.6 – Diagramme de classes d'entité

3.4 Diagrammes de séquence

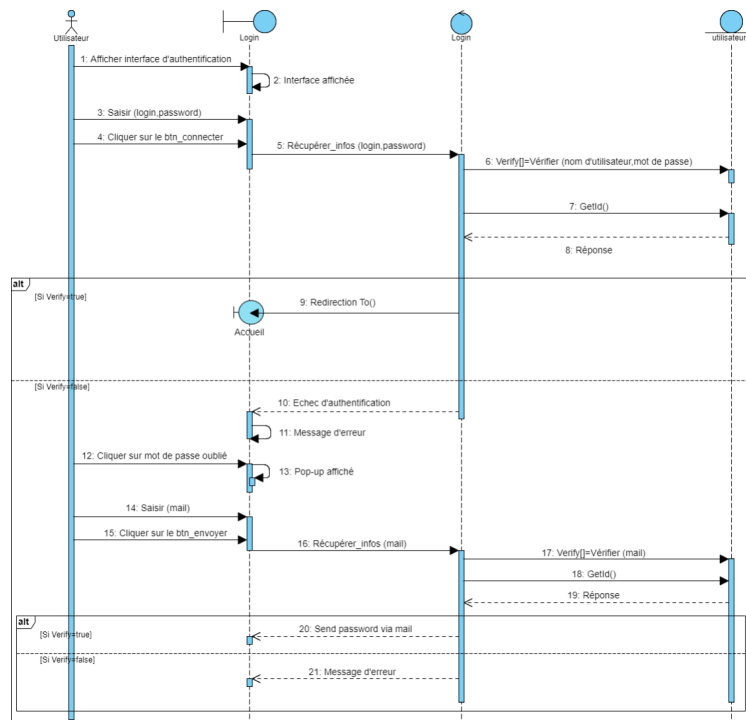


FIGURE 3.7 – Diagramme de séquences CU « S'authentifier »

Diagramme de séquences CU « Ajouter un utilisateur »

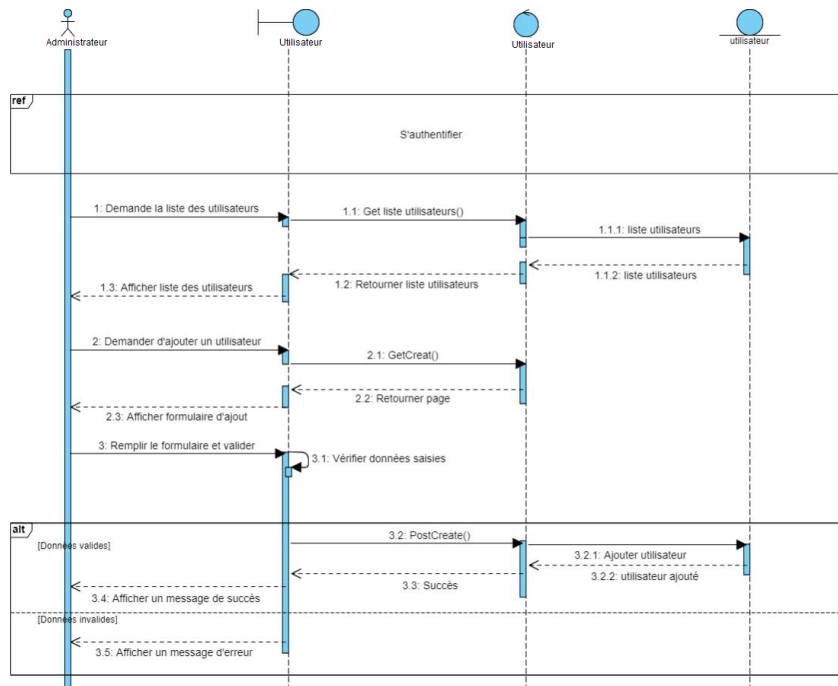


FIGURE 3.8 – Diagramme de séquences CU « Ajouter un utilisateur »

Conclusion

Ce présent chapitre a été dédié à la description des différents outils de réalisation et de conception utilisés pour le développement de la solution.

En outre, nous avons décrit d'une manière symbolique et schématique les différents éléments de l'architecture logicielle déployée.

Nous passons maintenant au quatrième chapitre, dédié à l'étude et la réalisation de la release.

CHAPITRE 4

RÉALISATION

Introduction	42
4.1 Étude et réalisation du Sprint1	42

Introduction

Ce chapitre est consacré à l'étude et la réalisation des différents sprints, Il s'agit essentiellement de définir leurs périmètres : c'est à dire les éléments du backlog de produit qui vont être réalisés.

4.1 Étude et réalisation du Sprint1

4.1.1 Backlog du Sprint1

Nous projetons l'ensemble des fonctionnalités (user-stories) écrites par le Product Owner et pris en charge par l'équipe de développement pendant le Sprint en cours.

Nous estimons un travail quotidien de cinq heures pendant quatre semaines pour la conception et la réalisation de ce sprint.

Nous dressons la liste des fonctionnalités en tenant compte de la quantité de travail (en heures) estimée pour chaque user-storie.

TABLE 4.1 – Backlog du Sprint1

Id	Features	User story	Estimation/h
1	Authentification	En tant qu'administrateur/ utilisateur, je dois me connecter à l'application web.	1
2	Gestion des utilisateurs	En tant qu'administrateur, je veux pouvoir ajouter un nouvel utilisateur avec son nom, son adresse e-mail et son rôle.	1
		En tant qu'administrateur, je veux pouvoir afficher la liste de tous les utilisateurs enregistrés dans le système.	1
		En tant qu'administrateur, je veux pouvoir modifier les informations d'un utilisateur existant, y compris son nom, son adresse e-mail et son rôle.	1
– Page suivante			

Table 4.1 – Backlog du Produit

Id	Features	User story	Priorité
		En tant qu'administrateur, je veux pouvoir supprimer un utilisateur du système.	1
		En tant qu'administrateur, je veux pouvoir visualiser les détails d'un utilisateur spécifique, y compris son nom, son adresse e-mail, son rôle et ses permissions.	3
		En tant qu'utilisateur, je veux pouvoir changer mon mot de passe.	3
		En tant qu'utilisateur, je veux pouvoir visualiser mes propres informations de profil.	2
		En tant qu'utilisateur, je veux pouvoir recevoir un e-mail de confirmation lors de la création de mon compte.	
		En tant qu'utilisateur, je veux pouvoir me connecter au système à l'aide de mon adresse e-mail et de mon mot de passe.	
9	Gestion des clients	En tant qu'utilisateur, je veux pouvoir filtrer l'historique des opérations par client en utilisant leur numéro de carte d'identité nationale (CIN) ou leur passeport pour identifier rapidement les transactions associées.	1
		En tant que client, je veux pouvoir mettre à jour mes informations personnelles si nécessaire.	2
		En tant qu'administrateur, je veux pouvoir gérer la base de données des clients.	2
		Le système garantisse la confidentialité et la sécurité des données des clients.	1

4.1.2 Spécification fonctionnelle

4.1.2.1 Raffinement des cas d'utilisation du sprint1

Le raffinement des cas d'utilisation produit plus de détails sur les caractéristiques fournis par le système et les contraintes qui leurs sont associées.

Raffinement du CU « S'authentifier »

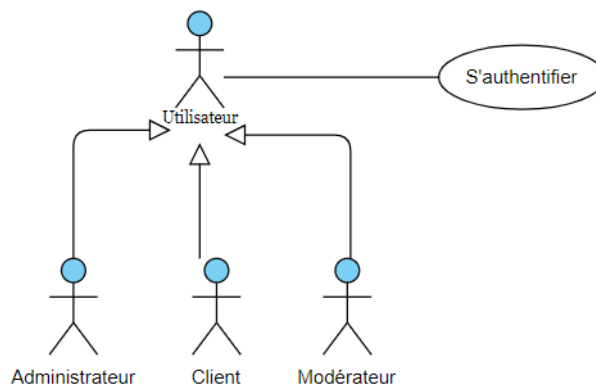


FIGURE 4.1 – S'authentifier

TABLE 4.2 – Description textuelle du CU « S'authentifier »

Nom	S'authentifier
Acteur	Tous les utilisateurs de l'application.
Objectifs	Se connecter à l'espace de l'utilisateur.
Pré-conditions	L'utilisateur doit avoir un compte.
S.Nominal	<p>❶ L'utilisateur remplit le formulaire d'authentification avec l'ensemble des informations nécessaires à son identification;</p> <p>❷ Le système vérifie les informations saisies par l'utilisateur;</p> <p>❸ Le système affiche l'espace de l'utilisateur authentifié;</p>
Exception	<p><i>A1 : Informations erronées</i></p> <p>❷ Le Système signale l'échec à l'utilisateur.</p> <p>Le CU redémarre à l'étape 1 du SN.</p>

Raffinement du CU « Gérer les utilisateurs »

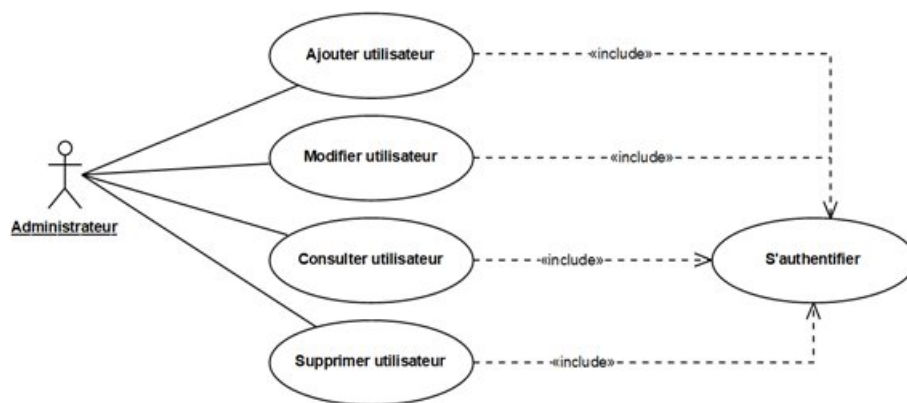


FIGURE 4.2 – Gérer les utilisateurs

Description textuelle du CU «Gérer des utilisateurs »

TABLE 4.3 – Description textuelle du CU «Gérer des utilisateurs»

Nom	Gérer utilisateurs
Acteur	Administrateur
Objectifs	L'administrateur crée un compte, s'il s'agit d'un nouveau utilisateur, sinon il fera la mise à jour nécessaire.(ajout, affichage, modification, suppression)
Pré-conditions	L'administrateur s'est authentifié sur le système;
S.Nominal	<p>❶ L'administrateur accède à la partie de gestion des utilisateurs;</p> <p>❷ Le système affiche l'interface;</p> <p>❸ L'administrateur choisit soit d'ajouter un nouveau utilisateur, de modifier un utilisateur existant , d'afficher les détails d'un utilisateur existant ou bien de supprimer un utilisateur;</p> <p>❹L'administrateur confirme l'opération choisie;</p> <p>❺Le système enregistre l'opération effectuée.</p>
S.Alternatif	<p>❹ Le système indique la détection des erreurs</p> <p>Le CU redémarre à l'étape 2 du SN.</p>

4.1.3 Conception

4.1.3.1 Diagrammes de séquences

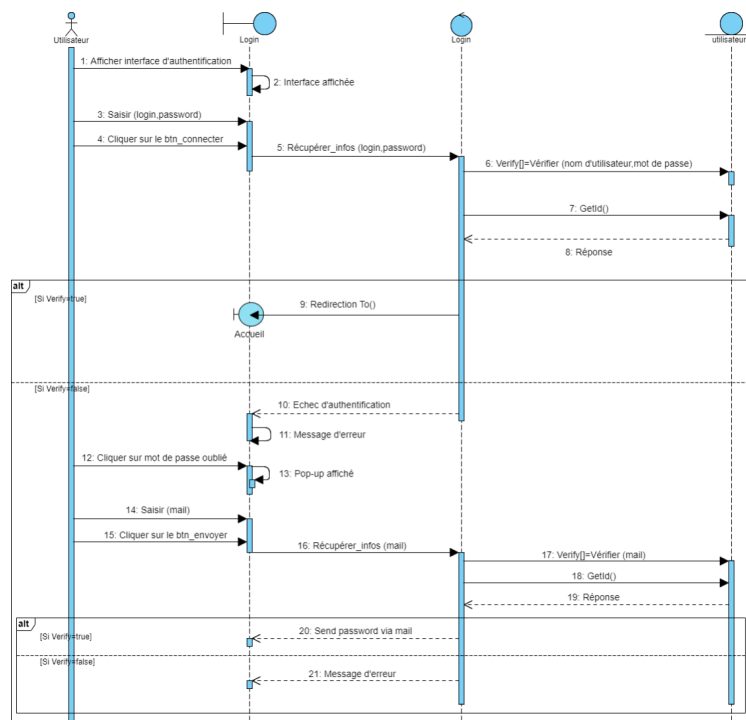


FIGURE 4.3 – Diagramme de séquences CU « S'authentifier »

Diagramme de séquences CU « Ajouter un utilisateur »

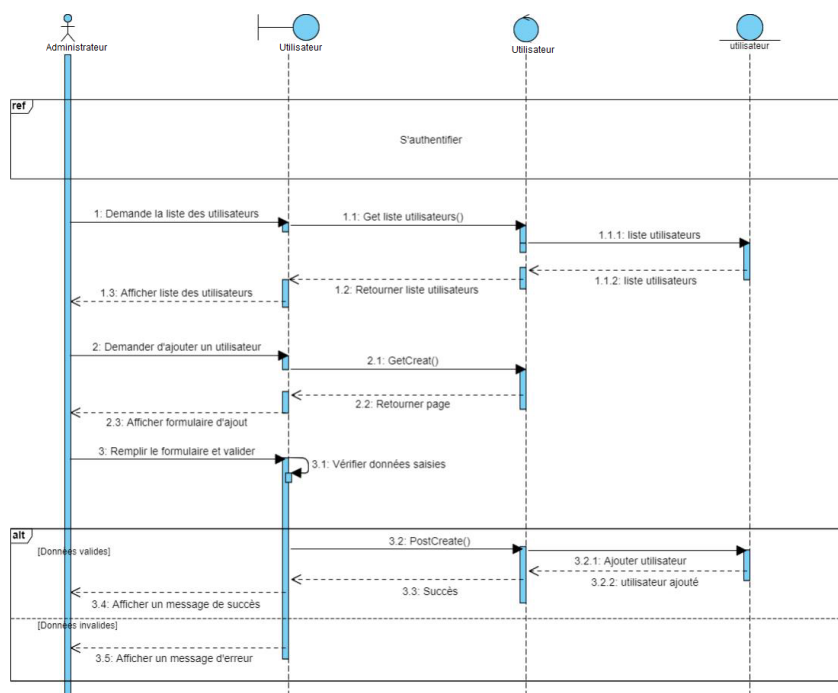


FIGURE 4.4 – Diagramme de séquences CU « Ajouter un utilisateur »

Diagramme de séquences CU « Modifier un client »

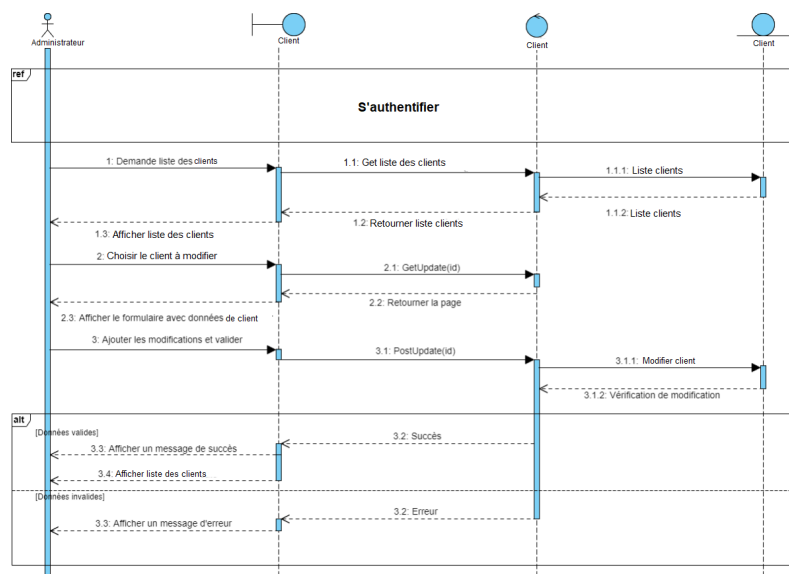


FIGURE 4.5 – Diagramme de séquences CU « Modifier un client »

4.1.3.2 Diagramme de classes du sprint1

Le diagramme de classes montre les blocs de construction de tout système orienté-objet. Les diagrammes de classes représentent une vue statique du modèle. Ou une partie du modèle, décrivant ce que les attributs et les comportements, qu'il a plutôt que de détailler les méthodes pour atteindre les opérations. Les diagrammes de classes sont les plus utiles pour illustrer les relations entre les classes et les interfaces. Généralisations, agrégations et les associations sont tous précieux reflétant l'héritage, la composition ou l'utilisation, et les connexions respectivement.



FIGURE 4.6 – Diagramme de classes du sprint1

CHAPITRE 5

PENTESTING DE LA SOLUTION PROPOSEE

Introduction	49
5.1 Pentesting	49

Introduction

Ce chapitre est consacré à l'étude et la réalisation des différents sprints, Il s'agit essentiellement de définir leurs périmètres : c'est à dire les éléments du backlog de produit qui vont être réalisés.

5.1 Pentesting

CONCLUSION GÉNÉRALE

A^U

RÉFÉRENCES BIBLIOGRAPHIQUES

- [1] Xtensus. Présentation de la société xtensus. <<http://www.xtensus.com/>>, 2019. [En ligne; accès 16-Février-2023].
- [2] Camille Meunier. Le salon professionnel : un boom pour les entreprises! <<https://arxama.com/le-salon-professionnel-un-boom-pour-les-entreprises/>>, 2020. [En ligne; accès 16-Février-2023].
- [3] Symfony. Présentation de symfony. <<https://symfony.com/>>, 2019. [En ligne; accès 16-Février-2023].
- [4] Visual Studio Code. Présentation de visual studio code. <<https://code.visualstudio.com/>>, 2019. [En ligne; accès 20-janvier-2018].
- [5] Visual Paradigm. Présentation de visual paradigm. <<https://www.visual-paradigm.com/>>, 2019. [En ligne; accès 4-mars-2023].
- [6] Composer. Présentation de composer. <<https://getcomposer.org/doc/>>, 2019. [En ligne; accès 16-Février-2023].
- [7] Pacestar. Présentation de pacestar. <<http://www.pacestar.com/uml/>>, 2019. [En ligne; accès 7-mars-2023].
- [8] Overleaf. Présentation d'overleaf. <<https://fr.overleaf.com/>>.
- [9] Bootstrap. Présentation de bootstrap. <<https://getbootstrap.com/>>, 2019. [En ligne; accès 8-avril-2020].
- [10] CSS 3. Présentation de css 3. <<https://www.w3.org/Style/CSS/Overview.fr.html>>, 2019. [En ligne; accès 6-décembre-2019].

- [11] HTML 5. Présentation de html 5. <<https://developer.mozilla.org/fr/docs/Web/Guide/HTML/HTML5>>, 2019. [En ligne; accès 6-décembre-2019].
- [12] MySQL. Présentation de mysql. <<https://www.mysql.com/fr/>>, 2019. [En ligne; accès 13-Février-2020].
- [13] PHP 7. Présentation de php 7. <<https://www.php.net/>>, 2019. [En ligne; accès 16-avril-2020].
- [14] jQuery. Présentation de jquery. <<https://jquery.com/>>, 2019. [En ligne; accès 19-Mars-2023].
- [15] Ajax. Ajax: Présentation et objectifs. <<https://www.chiny.me/ajax.php>>, 2019. [En ligne; accès 18-Mars-2023].