



2019 - 2020

PROJET DE FIN D'ÉTUDES

DIPLÔME NATIONAL D'INGÉNIEUR

SPÉCIALITÉ : INFORMATIQUE
SUJET: COSAFINANCE

Réalisé par: Shayma Mediouni

Encadrant ESPRIT: Rahma Ferjani

Encadrants Entreprise: Mohamed Ben Smida et Maroua Sfari



J'autorise l'étudiant à faire le dépôt de son rapport de stage en vue d'une soutenance.

Encadrants professionnel, Mr Mohamed Smida et Mlle Maroua

Safari

Signature et cachet

COSAVOSTRA SARL
Appt. B2-1 Rue Lac Tchad - 2ème Etage
Imm. Zen - Les Berges du Lac 1
1053 Tunis
MF: 1416503 K/A/M/000

J'autorise l'étudiant à faire le dépôt de son rapport de stage en vue d'une soutenance.

Encadrant académique, Madame Rahma Ferjani

Signature

Dédicaces

Je dédie ce modeste travail

A mes chers parents **Safia** et **Issa** qui m'ont fourni au quotidien un soutien et une confiance sans faille A tous mes frères et sœurs pour leur aide et leur soutien A tous mes amis qui n'ont pas épargnés leurs efforts pour m'encourager.

A mes chers amis **Sabrine**, **Skander**, **Khawla** et **Chayma** qui m'ont beaucoup encouragée et soutenu tout au long de ce projet avec toute ma reconnaissance et gratitude.

Shayma Mediouni

Remerciements

*Ce travail est le fruit de la miséricorde de Dieu, de la parole dure et du soutien de nombreuses personnes que je voudrais remercier. Je remercie tous ceux à qui la générosité, le professionnalisme et le conseil ont permis non seulement la réussite de ce projet de fin d'études mais aussi de me détacher progressivement du contexte académique au contexte professionnel. Je voudrais exprimer ma profonde et sincère gratitude à **Mme Rahma FERJANI** pour ses conseils et sa patience tout au long du projet, son professionnalisme et son dévouement m'ont encouragé à présenter un travail vivant. Un remerciement spécial à **Mme Hajar Hannini** notre scrum master lors de ce projet pour sa patience et ses conseils précieux. à **Mr Mohamed Smida** qui a toujours été présent pour m'aider, Mme **Maroua Sfari** aussi. De plus, je remercie tous les membres de mon équipe pour leur travail acharné et leur dévouement. Enfin, je remercie tout le personnel de CosaVostra pour toute l'aide qu'ils ont fournie.*

Table des matières

Introduction générale	1
1 Etude préalable	2
1.1 Présentation de l'organisme d'accueil	3
1.1.1 Cadre du projet	3
1.1.2 L'organisme de l'accueil	3
1.2 Contexte du projet	4
1.2.1 Problématique	4
1.2.2 Analyse de l'existant	4
1.2.3 Solution proposée	5
1.3 Méthodologies de travail	5
1.3.1 Choix de la méthode	7
1.3.2 Présentation de la méthodologie SCRUM	7
1.3.3 Les rôles dans SCRUM	8
2 Sprint 0 :Capture des besoins	11
2.1 Identification des acteurs du système	12
2.2 Identification des besoins	12
2.2.1 Besoins fonctionnels	12
2.2.2 Besoins non fonctionnels	13
2.3 Diagramme des cas d'utilisation globale	13
2.4 Le Backlog Produit	14
2.5 Architecture	16
2.5.1 Architecture physique	16
2.5.2 Architecture logique	17
2.6 Environnement de travail	18
2.6.1 Environnement Logiciel	19
2.7 Planification du sprint	24

3 Sprint 1	26
3.1 Objectif du Sprint	27
3.2 Analyse	27
3.2.1 Sprint backlog	27
3.3 Analyse spécifique	28
3.3.1 Diagramme des cas d'utilisation	28
3.3.2 Diagramme des séquences	33
3.4 Conception	35
3.4.1 Diagramme de classes	35
3.4.2 Diagramme d'activité	36
3.5 Réalisation et Test	37
3.5.1 S'authentifier	37
3.5.2 Mot de passe oublié	38
3.5.3 Consulter la liste des devis	40
4 Sprint 2	42
4.1 Objectif du Sprint	43
4.2 Analyse	43
4.2.1 Backlog du sprint 2	43
4.3 Analyse spécifique	44
4.3.1 Diagramme de cas d'utilisation	44
4.3.2 Diagramme de séquence	50
4.3.3 Diagramme d'activité	51
4.4 Conception	52
4.4.1 Diagramme de classe globale du sprint 2	52
4.4.2 Diagramme de séquence objet	53
4.5 Réalisation et test	54
4.5.1 Incription	54
4.5.2 Accepter contact client	55
4.5.3 Attribué contact	56
4.5.4 Consulter la liste des contacts des facturations	57
4.5.5 Modifier et/ou exporter des contacts	58

5 Sprint 3	59
5.1 Objectif du Sprint	60
5.2 Analyse	60
5.2.1 Backlog du sprint 3	60
5.3 Analyse spécifique	61
5.3.1 Diagramme de cas d'utilisation	61
5.3.2 Raffinement du cas d'utilisation «Consulter liste des objectifs»	62
5.3.3 Raffinement du cas d'utilisation «Consulter liste des factures»	63
5.3.4 Raffinement du cas d'utilisation «Consulter les statistiques»	65
5.3.5 Diagramme de séquence «Modifier comptes clients»	66
5.4 Conception	68
5.4.1 Diagramme de classe du sprint 3	68
5.4.2 Diagramme d'activité du cas d'utilisation «Ajout objectifs»	68
5.5 Réalisation et test	69
5.5.1 Gestion des compte client	69
5.5.2 Gestion des objectifs	70
5.5.3 Gérer les statistiques	72
Conclusion générale	73
Netographie	74

Table des figures

1.1	Logo de l'agence	3
1.2	Cycle de vie de la méthode SCRUM [1]	9
2.1	Diagramme de cas d'utilisation globale	14
2.2	Architecture physique	16
2.3	Architecture logique	17
2.4	Visual studio code [3]	19
2.5	ReactJS [4]	20
2.6	JavaScript [5]	20
2.7	HTML5 et CSS3 [6]	21
2.8	PhpStorm [7]	21
2.9	PHP [8]	22
2.10	Postman [9]	22
2.11	Git [10]	22
2.12	Visual Paradigm [11]	23
2.13	Sentry [12]	23
2.14	Slack [13]	23
2.15	Trello [14]	24
3.1	Diagramme de cas d'utilisation global	28
3.2	Diagramme de cas d'utilisation «S'authentifier»	29
3.3	Diagramme de cas d'utilisation «Consulter les devis»	31
3.4	Diagramme de séquence objet du cas « Consulter devis Backend»	33
3.5	Diagramme de séquence objet du cas « S'authentifier »	34
3.6	Diagramme de séquence système du cas «Authentification»	35
3.7	diagramme de classe du sprint 1	36
3.8	Diagramme d'activité du cas d'utilisation «S'authentifier»	37
3.9	Authentification	38
3.10	Message d'erreur	38
3.11	Saisir email	39

Table des figures

3.12 Email envoyé	39
3.13 Email envoyé	40
3.14 Liste des devis	40
3.15 Liste des devis	41
4.1 Raffinement du cas d'utilisation : Sprint2	45
4.2 Raffinement du cas d'utilisation «Attribué une contacts»	45
4.3 Raffinement du cas d'utilisation «Consulter des contacts»	47
4.4 Diagramme de séquence système «Attribution des contacts du facturation»	50
4.5 Diagramme de séquence système du cas «Consulter contacts»	51
4.6 Diagramme d'activité du cas : Attribution des contacts du facturation	52
4.7 Diagramme de classe de conception	53
4.8 Diagramme de séquence système du cas «Enregistrer contacts»	53
4.9 Formulaire d'inscription	54
4.10 Email d'un nouveau contact	55
4.11 Accepter contact client	55
4.12 Accepter contact client	56
4.13 Attribution des contacts du facturation	56
4.14 Contacts facturation	57
4.15 Détails facturation	57
4.16 Liste des devis	58
5.1 Raffinement du cas d'utilisation : Sprint3	62
5.2 Raffinement du cas d'utilisation «Consulter liste des objectifs»	62
5.3 Raffinement du cas d'utilisation «Consulter des facture»	64
5.4 Raffinement du cas d'utilisation «Consulter les statistique»	66
5.5 Diagramme de séquence système du cas : «Modifier comptes client»	67
5.6 Diagramme de séquence Objet «Modifier comptes clients»	67
5.7 Diagramme de classe du sprint 3	68
5.8 Diagramme d'activité du cas d'utilisation «Ajout objectifs»	69
5.9 Consultation des comptes clients	69
5.10 Modification des comptes clients	70

5.11 Consultation des objectifs	71
5.12 Ajout des objectifs	71
5.13 Consultation la liste des statistique	72

Liste des tableaux

1.1	Étude comparative des méthodologies	6
2.1	Backlog Produit	14
2.2	Environnement matériel	18
2.3	Planification du sprint	24
3.1	Backlog du sprint 1	28
3.2	Description textuelle du cas «S'authentifier »	30
3.3	Description textuelle du cas «réinitialisation mot de passe »	30
3.4	Consulter les devis «Consulter les devis»	31
3.5	Description textuelle du cas «Exporter les devis »	32
3.6	Description textuelle du cas «Afficher un devis »	32
3.7	Description textuelle du cas «Modifier un devis»	32
4.1	Backlog du sprint 2	43
4.2	Description textuelle du cas «Enregistrer contacts»	46
4.3	Description textuelle du cas «Consulter les contacts»	47
4.4	Description textuelle du cas «Consulter une contact»	48
4.5	Description textuelle du cas «Attribué contact»	48
4.6	Description textuelle du cas «Modifier un contact»	49
4.7	Description textuelle du cas «Ajouter contact à sellsy»	49
4.8	Description textuelle du cas «Exporter les contacts»	50
5.1	Backlog du sprint 3	61
5.2	Description textuelle du cas «Consulter des objectifs»	63
5.3	Description textuelle du cas «Ajouter objectifs»	63
5.4	Description textuelle du cas «Consulter les factures»	64
5.5	Description textuelle du cas «Exporter les factures»	64
5.6	Description textuelle du cas «Modifier des comptes clients»	65
5.7	Description textuelle du cas «Consulter les statistique»	66

Liste des abréviations

- **API** = Application Programming Interface
- **CSS** = Cascading Style Skee
- **ERP** = Enterprise Resource Planning
- **HTML** = Hyper Text Markup Langage
- **HTTP** = Hyper Text Transfer Protocol
- **JWT** = JSON Web Token
- **MVC** = Modèle Vue Contrôleur
- **MySQL** = My Structured Query Language
- **PHP** = Hypertext Preprocessor
- **REST** = REpresentational State Transfer
- **SQL** = Structured Query Language
- **UML** = Unified Lodeling Language

Introduction générale

Pendant les dernières années l'informatique s'est étendue dans le monde entier bien évidemment ainsi qu'en Tunisie. En effet ce domaine a subi une évolution importante dans notre pays, on entend du jours au lendemain la création de nouvelles applications qui peuvent être gratuites ou payantes accessibles sur le desktop localement ou à l'aide de l'internet ces derniers sont nommées des applications web qui fournissent des services numériques qui rendent la vie plus facile et confortable à mener.

Dans ce contexte, l'agence « CosaVostra » dans laquelle nous avons effectué notre stage de fin d'études, voudrait mettre en place une application web d'une outil de direction et reporting financier. Cette solution interrogera différents ERP déjà existants basé sur les APIs, sous une architecture de micro-services, pour centraliser et formaliser des indicateurs de croissance et gestion.

Dès lors, notre projet de fin d'études consiste à concevoir et à réaliser une application web dont le but est de gérer les devis et des factures est pour le but de tracker les clients en retard de paiement et ce en les regroupant selon le délai de retard qui sont déjà importer a travers l'API SELLSY de manipuler ses données selon le besoin de notre clients et d'ajouter des contacts de facturation à sellsy à travers notre application.

Le présent rapport est organisé en cinq chapitres :

- Le premier chapitre intitulé « Cadre Général du projet » présente l'organisme d'accueil, décrit le contexte de notre projet ainsi que la méthodologie adoptée.
- Le deuxième chapitre explique notre démarche, soit, l'identification des futurs acteurs de notre système et l'analyse des besoins. Ainsi que la description de l'architecture choisie pour réaliser notre solution.
- Le troisième chapitre portera sur la partie du développement du premier sprint ayant comme but l'authentification et la gestion des devis. Nous présentons tout au long de ce chapitre la conception et la réalisation du premier sprint.
- Le quatrième chapitre abordera le sprint 2 ayant comme but d'accepter des contacts et la gestion des contacts de facturations.
- Le cinquième chapitre représente le dernier sprint qui permet de mettre en place la gestion des objectifs et afficher les l'ensemble des clients en retard de paiement. Nous présentons tout au long de ce chapitre la conception et la réalisation du premier sprint.

Chapitre 1

ETUDE PRÉALABLE

Plan

1	Présentation de l'organisme d'accueil	3
2	Contexte du projet	4
3	Méthodologies de travail	5

Introduction

Ce chapitre est consacré à la présentation du cadre général du projet ainsi que l'organisme d'accueil « COSAVOSTRA ». Tout d'abord, nous présentons l'organisme d'accueil, le travail demandé. Ensuite, nous effectuons une étude de l'existant. Enfin, nous expliquons notre méthode du travail, le chronogramme du projet et le langage de modélisation que nous avons adoptée au cours de notre travail.

1.1 Présentation de l'organisme d'accueil

1.1.1 Cadre du projet

Notre projet de fin d'étude intitulé « Conception et réalisation d'une application web d'un outil financier» a pour but d'obtenir un diplôme d'ingénieur en informatique de l'Ecole Supérieure Privée d'Ingénierie et de Technologie (ESPRIT).

1.1.2 L'organisme de l'accueil

Le projet a été réalisé au sein de l'entreprise CosaVostra est un cabinet de conseil en innovation, une agence digitale pirate et un startup studio intrépide fondée en 2013 par 4 entrepreneurs du digital, elle est basée à Paris (75017) avec 45 collaborateurs.



Figure 1.1: Logo de l'agence

Comme CosaVostra est proche culturellement et géographiquement de ses clients, elle est présente avec ses services à Paris, Londres, Bordeaux et Tunis. L'approche adoptée par CosaVosta

est basée sur les services. En effet CosaVostra fournit des prestations à haute valeur ajoutée dans les domaines de digital, conseil, marketing, content ou encore le développement web. Les services offerts par CosaVostra sont les suivants :

- Direction de projet, design (UI/UX) et développement :
 - Sites de médias
 - E-commerce / online stores
 - Sites d'entreprises / corporate
 - Applications web et mobiles
- Expertise :
 - Communication en ligne (organique et payante)
 - Stratégie social media
 - Production de contenu

1.2 Contexte du projet

Dans cette partie, nous allons présenter l'étude de l'existant, la problématique, les objectifs ainsi les différentes étapes de déroulement du projet.

1.2.1 Problématique

Dans CosaVostra, attribuer les factures et les devis avec tri selon client causent beaucoup de perturbations. Le saisie de chaque facture et devis se font via une feuille Excel, par un comptable. Parfois l'ordonnancement des devis et des factures créent beaucoup de conflit, pour tenter une nouvelle décision doit être prise qui provoque plus d'intervention, plus de temps perdu et par conséquent, cela crée un grand flux et le gonflage de la feuille Excel ... En se basant sur ces problèmes, le système extant souffre des perte de temps, perte des données ,...

1.2.2 Analyse de l'existant

L'analyse de l'existant permet de détecter les anomalies et les défauts des solution présentes sur le marché afin de proposer une solution meilleure.

1.2.2.1 Étude de l'existant

Il existe plusieurs solution permettant la gestion et le contrôle financière. Après l'investigation que nous avons faite au milieu des experts et d'après les recherches , nous avons noté que peu son les solutions qui font ça et les plus populaires sont :

- Zervant
- Mon AE

1.2.2.2 Critique de l'existant

Les solutions présentes sur le marché sont soit payantes, donc pour bénéficier des services proposés il faut ou bien payer au départ, ou bien propose un trial de maximum 30 jours pour découvrir les fonctionnalités existantes, soit on les trouve gratuits mais avec des fonctionnalités restreintes et il faut payer pour bénéficier des fonctionnalités plus développées.

1.2.3 Solution proposée

Pour remédier à ces problèmes, CosaVosstra a décidé d'implémenter sa propre solution qui consiste à un outil de direction financière avec des fonctionnalités personnalisées et plus développées que celles présentes sur le marché. L'application « Finance Scheduler » permet de récupérer les informations des factures et des clients à travers l'API SELSSY. Notre application offre également un environnement pour la gestion de contrôle des devis , la gestion des factures qui détaille les retards des clients. De plus, elle permet d'ajouter un contact à l'application Sellsy et de notifier les clients afin de faire suivi de leurs devis qui contiennent toutes les informations nécessaires. Pour améliorer et renforcer la permanence de notre application, «CosaFinance» fait une mise à jour des données automatiquement.

1.3 Méthodologies de travail

Un projet informatique, quelle que soit sa taille et la portée de ses objectifs, nécessite la mise en place d'un planning organisationnel tout au long de son cycle de vie. C'est ainsi qu'est apparue la notion de méthode. Nous proposons une étude comparative dans le tableau 1.1 afin de choisir la méthode la plus adéquate.

Tableau 1.1: Étude comparative des méthodologies

Thème	Approche classique	Approche agile
Cycle de vie	Cycle de vie en cascade	Itératif et incrémental
Planification	Préditive où l'identification des besoins est établie au préalable et définitives.	Adaptative et toujours ouverte à tous ajustements en fonction des changements survenus.
Documentation	Produite en quantité importante.	Réduite au strict nécessaire.
Équipe	Une équipe avec des ressources spécialisées, dirigées par un chef de projet.	Une équipe responsabilisée où l'initiative et la communication sont privilégiées, soutenue par le chef projet.
Qualité	Contrôle qualité à la fin du cycle de développement. Le client découvre le produit fini.	Un contrôle qualité précoce et permanent. Le client visualise les résultats tôt et fréquemment.
Changement	Résistance voire opposition au changement. Processus lourds de gestion des changements acceptés.	Accueil favorable au changement inéluctable, intégré dans le processus.
Suivi de lancement	Mesure de la conformité aux plans initiaux. Analyse des écarts.	Un seul indicateur d'avancement : le nombre de fonctionnalités implémentées et le travail restant à faire.

Gestion des risques	Processus distinct, rigoureux, de gestion des risques	Gestion des risques intégrés dans le processus global, avec responsabilisations de chacun dans l'identification et la résolution des risques. Pilotage par les risques.
Mesure de succès	Respect des engagements initiaux en termes de coûts, de budget et de niveau de qualité.	Satisfaction client par la livraison de valeur ajoutée.

1.3.1 Choix de la méthode

Suite à l'étude comparative que nous avons effectuée dans la section précédente, et après une longue délibération basée sur les besoins de notre client, nous avons opté pour «SCRUM» comme méthode de gestion pour notre projet.

En effet, cette méthodologie agile est la plus utilisée et la plus populaire dans le monde. Ceci prouve donc son efficacité et son rendement.

Cette méthodologie est basée sur l'esprit collaboratif et elle est adaptée aux méthodes incrementales. De plus, la méthode Agile engendre des produits de haute qualité tout en tenant compte de l'évolution des besoins du client.

Elle permet de détecter les problèmes le plus tôt au fur et à mesure, permettant ainsi d'entreprendre des actions correctrices sans trop de pénalités dans les coûts et les délais. Il y a plusieurs méthodes AGILE et il ne s'agit pas de choisir la meilleure méthode parmi celles existantes.

1.3.2 Présentation de la méthodologie SCRUM

Scrum offre un cadre précis et souple, parfait pour les projets innovants ou complexes. Cette méthode a pour objectif d'améliorer la productivité des équipes et de favoriser le dialogue entre le client et le prestataire, afin d'optimiser la réussite des projets.

Le principe de Scrum est de développer un logiciel de manière incrémentale en maintenant une liste totalement transparente des fonctionnalités à développer, des demandes d'évolutions ou de corrections à implémenter (backlog). Avec des livraisons très fréquentes, le client reçoit à chaque

fois un logiciel avec des fonctionnalités nouvelles, et en parfait état de fonctionnement. Pour cela, la méthode s'appuie sur des développements itératifs à un rythme constant d'une durée de 1 à 4 semaines.

1.3.3 Les rôles dans SCRUM

Scrum est une méthodologie agile qui consiste à avoir une équipe soudée orientant le projet au fil de son avancement afin d'atteindre un but faisant intervenir trois rôles principaux qui sont : Le Product Owner : dans la majorité des projets, le responsable produit (Product Owner) est le responsable de l'équipe projet client. Dans notre projet le Product Owner est présenté par M.PIERRE STEFANIE comme un expert métier afin de :

- Définir la liste des fonctionnalités du produit.
- Prioriser les fonctionnalités en fonction de leur importance et leur valeur ajoutée pour l'entreprise qu'il présente.
- Choisir la date de livraison des versions ainsi que leurs contenus.
- Valider les lots livrés avec l'équipe de développement
- Clarifier les besoins à l'équipe de développement si nécessaire

Le Scrum Master est présenté par Mme Hajar Hanini, c'est le leader de l'équipe, il assure les tâches suivantes :

- S'assurer que Scrum est bien appliquée et respectée
- Encourager l'équipe à apprendre et à progresser pour qu'elle soit fonctionnelle, productive et créative durant le projet
- Eliminer les obstacles pouvant perturber la progression du travail.

L'équipe de développement est présentée par MEDIOUNI Chayma. Elle regroupe tous les rôles habituellement nécessaires à un projet, à savoir l'architecte, le concepteur, le développeur, le testeur, etc. Elle est « auto organisée » et elle reste inchangée pendant toute la durée du sprint. Son rôle principal est :

- Transformer les besoins exprimés dans le Sprint Backlog (est défini ci-dessous) en fonctionnalités utilisables.
- Livrer régulièrement une version fonctionnelle du produit.

La figure 1.2 montre le cycle de vie de la méthode SCRUM et les relations entre le client et le Product Owner..

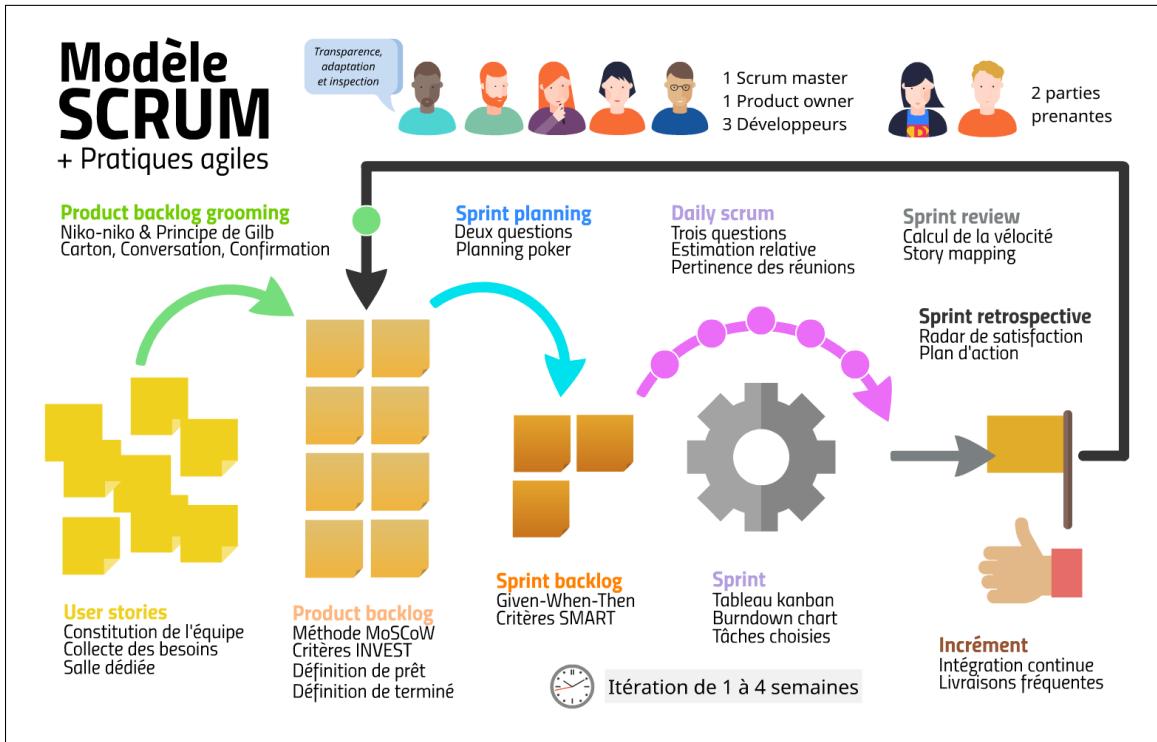


Figure 1.2: Cycle de vie de la méthode SCRUM [1]

Backlog produit (ou catalogue des besoins)

- Besoins priorisés par le Product Owner.
- Besoins évalués par l'équipe.

Sprint (itération)

- Développement des fonctionnalités du Backlog de sprint.
- Aucune modification du Backlog de sprint possible.
- Mélée quotidienne (Rencontre quotidienne)
- Point de contrôle quotidien de l'équipe.
- Intervention régulières – 2 min par personne.

Sprint Backlog

— Extrait du Backlog produit.

Produit livrable livré au Product Owner à la fin du sprint.

Conclusion

Dans ce chapitre, nous avons donné un aperçu du projet en décrivant l'organisme d'accueil, CosaVostra, et le contexte du projet. Nous avons présenté aussi notre méthodologie de travail pour réaliser notre projet, comme la méthode Scrum. Le reste du rapport sera organisé selon cette méthode. Le chapitre suivant sera consacré au lancement du projet et la spécification de ses besoins.

SPRINT 0 : CAPTURE DES BESOINS

Plan

1	Identification des acteurs du système	12
2	Identification des besoins	12
3	Diagramme des cas d'utilisation globale	13
4	Le Backlog Produit	14
5	Architecture	16
6	Environnement de travail	18
7	Planification du sprint	24

Introduction

Ce chapitre est consacré à la capture des besoins fonctionnels et non fonctionnels de notre système, au pilotage de projet avec scrum, par élaboration du Backlog produit avec une planification des sprints ainsi qu'à la description de l'environnement de travail.

2.1 Identification des acteurs du système

Un acteur est l'idéalisation d'un rôle joué par une personne externe, un processus ou une chose qui interagit avec un système. [2] Pour notre application, nous avons identifier deux acteurs un utilisateur et l'admin de notre application.

- L'utilisateur simple de l'application est le client qui suit son état de paiement en temps réel .
- L'admin de notre application a tous les priviléges.Il peut accéder à toutes les interfaces de l'application, ajouter des utilisateurs, gère des comptes client , gère les devis...

2.2 Identification des besoins

2.2.1 Besoins fonctionnels

Les besoins fonctionnels représentent les actions qu'un système doit être capable d'exécuter et qu'il ne sera opérationnel qu'une fois celles-ci sont effectués. Notre solution doit fournir un ensemble de fonctionnalités répondant aux besoins de nos différents acteurs.

— Enregistrer contacts

Un utilisateur doit s'inscrire afin de bénéficier des avantages de notre application après la confirmation de son adhésion par l'admin.

— Authentification

Un utilisateur admin doit s'authentifier pour qu'il puisse accéder à notre platform.

— Confirmer contacts clients

L'admin de notre application peut accepter les nouveaux contacts clients.

— Contrôler la gestion des devis (importer les données via API sellsy et l'exporter via un doc csv)

L'interface Devis permet d'exporter la liste des devis avec l'api sellsy,l'afficher et ajouter ainsi que modifier quelques informations relatives à un devis. Cette interface donne la possibilité

d'importer les fiches des devis.

— **Générer des rapports (statistique pour les client en retard de paiement)**

Les clients qui ont un retard de paiement sont regroupés selon des intervalles de jours de retards.

— **Gérer les objectifs**

L'interface de suivi des objectifs permet de fixer des objectifs pour chaque secteurs, insérer les chiffres d'affaires atteint au fur et à mesure et ainsi avoir une visibilité sur le pourcentage de réalisation des prévisions.

2.2.2 Besoins non fonctionnels

Les besoins non fonctionnels sont des besoins qui expriment la qualité du service rendu en termes de performances, de temps d'exécution, d'évolution, etc. La prise en compte des spécifications non fonctionnelles est essentielle, puisque ceci permet de nous guider lors de la conception de l'architecture du système. Les principaux besoins non fonctionnels respectés par notre application sont :

— **Sécurité :**

Accès sécurisé et personnalisé. Le système doit être sécurisé, Notre plateforme assure pour chaque login un mot de passe bien sécurisé. Mise en place d'un système pour tracer les accès dont la gestion se base sur une politique de gestion de rôle. Par ailleurs, l'accès aux interfaces est fait suite à une analyse des rôles.

— **L'ergonomie :**

Notre application assure des interfaces bien présenté et bien guider.

— **Performance :**

Le temps d'exécution du système doit être faible.

2.3 Diagramme des cas d'utilisation globale

Dans cette section de ce chapitre nous allons détailler les fonctionnalités offertes par notre système ainsi que leurs associations avec les acteurs décrits auparavant à l'aide du diagramme de cas d'utilisation globale de notre application.

La figure 3.1 représente les principes fonctionnalités.

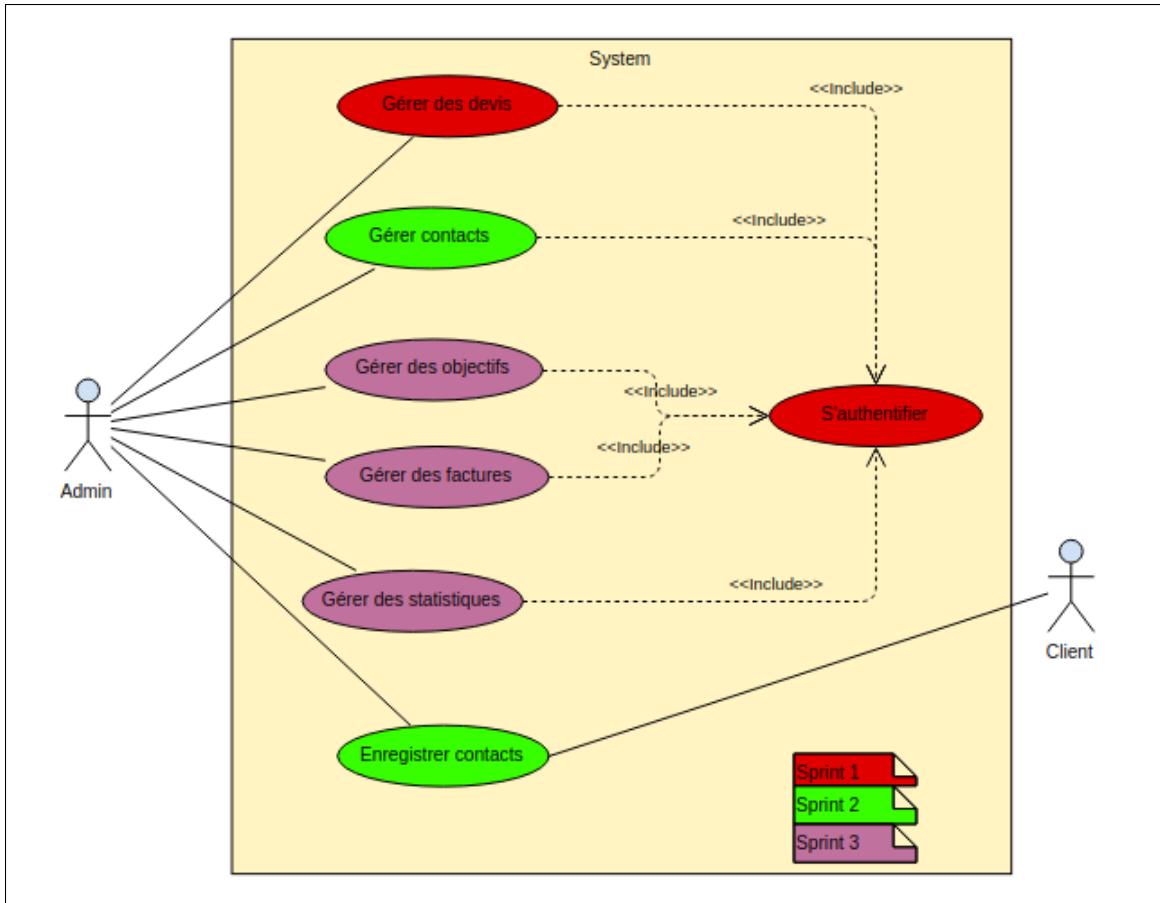


Figure 2.1: Diagramme de cas d'utilisation globale

2.4 Le Backlog Produit

Nous allons présenter les différentes tâches des acteurs dans le backlog produit :

Tableau 2.1: Backlog Produit

Fonctionnalité	ID	User Stories	Tâche	Priorité
Authentification	1	1.1	En tant qu'admin de l'application je dois saisir un email et un mot de passe pour accéder à l'application.	1
		1.2	En tant qu'admin je veux modifier mon mot de passe.	2

Enregistrer contacts	2	2.1	En tant que client je peux m'inscrire à l'application CosaFinance.	3
		2.2	En tant qu'admin je reçois un email contenant les données du nouveaux contacts	4
Gestion des devis	3	3.1	En tant qu'admin de l'application, je peux consulter la liste des devis	5
		3.2	En tant qu'admin je souhaite exporter les données de la liste des devis.	6
		3.3	En tant qu'admin je souhaite modifier quelque information à propos la liste des devis	7
		3.4	En tant qu'admin je souhaite de recevoir un email contenant les devis chaque deux semaines	8
Suivre la gestion des contacts de facturations	4	4.1	En tant qu'admin je souhaite consulter la liste des contacts de facturation.	9
		4.2	En tant qu'admin je souhaite modifier la liste des contacts de facturation	10
		4.3	En tant qu'admin je peux d'exporter la liste des contacts de facturation	11
		4.4	En tant qu'admin je peux attribuer un contacts à la liste des projet qui sont déjà importer du sellsy et d'exporter vers SELLSY.	12
Gestion des clients	5	5.1	En tant qu'admin je peux confirmer des comptes clients	13
		5.2	En tant qu'admin je souhaite suivre des comptes clients	14

		5.3	En tant qu'admin je peux exporter la listes détaillées des clients en csv	15
Gestion des objectifs	6	6.1	En tant qu'admin je souhaite afficher l'interface de suivre des objectifs	16
		6.2	En tant qu'admin je peux ajouter des objectifs	17
Gestion des statistiques	7	7.1	En tant qu'admin je souhaite afficher les statistique	18

2.5 Architecture

2.5.1 Architecture physique

L'architecture physique de notre application (où également nommée l'architecture technique d'une application), décrit l'ensemble des composants matériels constituant l'application web. Dans ce contexte, notre application est constituée principalement d'un client, serveur de base de données MYSQL et un serveur web apache. Nous allons détailler l'architecture physique de notre application web dans la figure 2.2 :

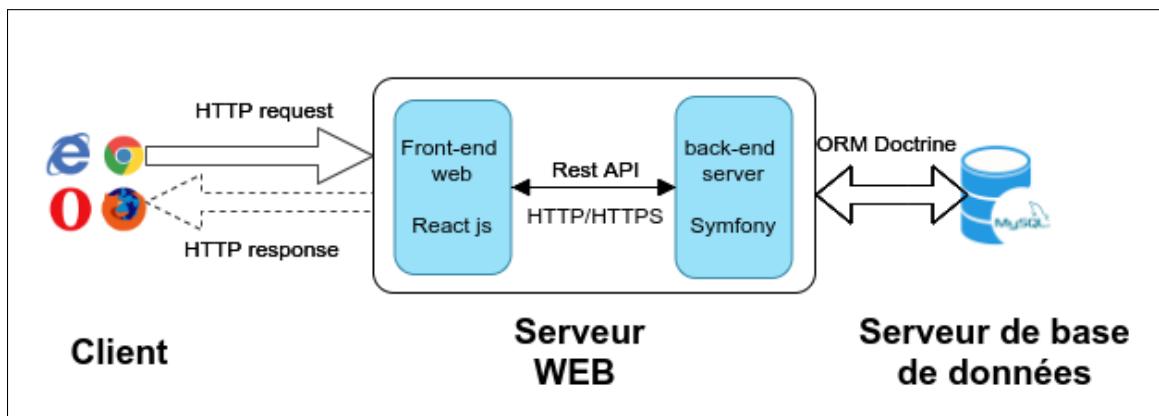


Figure 2.2: Architecture physique

L'architecture est composée :

- D'un niveau client gérant la couche présentation.
- D'un niveau serveur d'applicatif prenant en compte les applications et les objets métiers.
- D'un noeud physique contenant les serveurs de la base de données.

2.5.2 Architecture logique

L'architecture logique est une conception structurelle qui donne autant de détails que possible sans contraindre l'architecture à une technologie ou un environnement particulier. La figure 2.3 nous permet d'avoir une meilleure visibilité sur les différents composants de notre serveur client et serveur d'application.

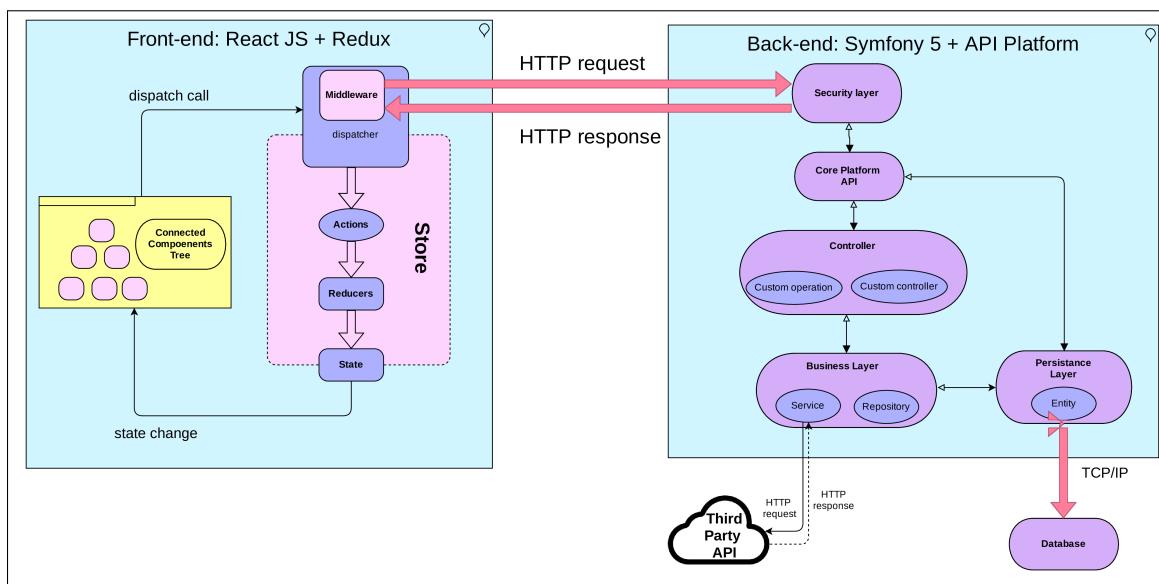


Figure 2.3: Architecture logique

- **Back-end model** : le niveau back-end est composé de 5 modules principaux : couche de sécurité, plate-forme API principale, contrôleur, couche métier et couche de persistance.
- **Security layer** : cette couche est le pare-feu de notre back-end, elle filtre les requêtes HTTP à venir, à l'intérieur nous définissons les routes qui nécessitent une authentification et son accessibilité en fonction du rôle.
- **Core API platform** : c'est une bibliothèque intégrée à Symfony 5, elle aide à créer des API pilotées par hypermédia.

- **Controller** : il gère les requêtes HTTP à venir et renvoie les réponses json-ld dans notre cas. Nous avons utilisé des opérations et des contrôleurs personnalisés pour gérer des scénarios complexes
- **Business layer** : ou la couche service contient les opérations courantes comme Ajouter, persister, Supprimer ... nous définissons toute notre logique métier à l'intérieur des classes de service. Par exemple, nous avons créé des opérations pour conserver et mettre à jour les données entrantes à partir d'API tierces dans les classes de services.
- **Persistence layer** : c'est là que nous définissons nos modèles de classe et les associations entre eux.
- **Front-end model** : le front-end est composé de 2 parties principales notre arborescence de composants et le store Redux.
- **Components** : ce sont des classes ou des fonctions javascript dans lesquelles nous définissons la logique de l'interface utilisateur, elles acceptent d'autres propriétés (props) à venir d'autres composants et renvoient la façon dont notre interface utilisateur prendra forme. Il y a un composant racine qui enveloppe tous les autres composants (App.js) comme des enfants, c'est pourquoi nous nous assimilons à une structure arborescente.
- **Redux store** : il s'agit d'un conteneur de gestion d'état global pour notre application React. Tout composant connecté au store Redux peut envoyer un appel d'action, après les appels d'actions transmettent des propriétés aux réducteurs(reducers), chaque réducteur (reducer) a une action définie et un état donné, le rôle des réducteurs (reducers) est de fournir le nouvel état aux composants.

2.6 Environnement de travail

L'architecture matérielle utilisée pour la réalisation de l'application est la suivante :

Tableau 2.2: Environnement matériel

Marque	HP-Pavillion
Processus	Intel® Core i5-7200U CPU
Mémoire	12 GO
Système d'exploitation	Ubuntu

2.6.1 Environnement Logiciel

2.6.1.1 Outils de développement et modélisation

— Visual studio :

Pour le développement front-end, nous avons utilisé un nouvel éditeur développé récemment par Microsoft : c'est Visual code studio. Cet éditeur a gagné une très bonne satisfaction au sein de développeur front-end, c'est un outil open source qui facilite le développement moyennant les langages JavaScript et TypeScript.

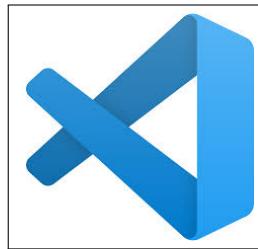


Figure 2.4: Visual studio code [3]

— **ReactJS** : C'est une bibliothèque JavaScript libre développée par Facebook depuis 2013. Conçue principalement pour faciliter la création d'application web (Single page application), en utilisant les composants dépendant d'un état et générant une page (ou portion) HTML à chaque changement d'état.

Dans notre application, on utilise la bibliothèque ReactJS pour le développement frontal.

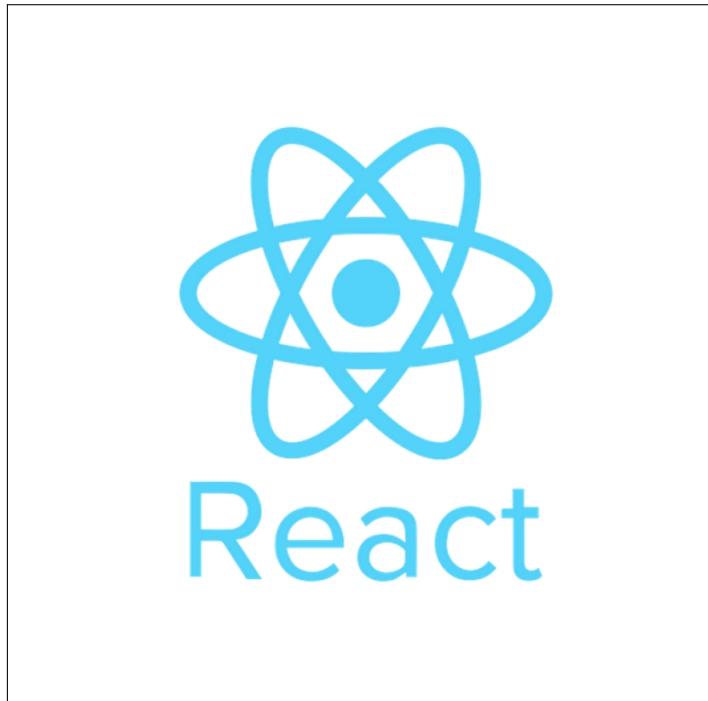


Figure 2.5: ReactJS [4]

— **JavaScript :**

C'est un langage de programmation qui est inclus dans le code HTML. Il permet d'apporter des améliorations au langage HTML en permettant d'exécuter des commandes.



Figure 2.6: JavaScript [5]

— **HTML5 et CSS3 :**

« Hyper Text Markup Langage 5 » HTML5 considéré comme l'une des nouvelles Technologies les plus importantes, qui étaient émergées en 2010, et qui sont encore un travail ne sont pas totalement finalisées par le W3C. Elle est un langage de balisage conçu pour la création des pages Web.

Quant au CSS3 « Cascading Style Sheets » permet d'arrondir les images, faire des ombres sur les divas, des ombres sur du texte, des polices de caractères plus fun, des bordures d'images,

etc. Et surtout l'ajout de l'animation. Il utilisait pour la description des styles et la disposition des éléments des pages HTML



Figure 2.7: HTML5 et CSS3 [6]

— **Jetbrains PhpStorm :**

PhpStorm est un IDE PHP léger et intelligent axé sur la productivité des développeurs, fournit la complétion intelligente de code, une navigation rapide et la vérification des erreurs à la volée. Il est toujours prêt à aider et créer des codes, exécutez des tests et fournir un débogage visuel.



Figure 2.8: PhpStorm [7]

— **PHP :**

Hypertext Preprocessor, connu sous le nom de PHP , c'est un langage de programmation libre, particulièrement utilisé pour la conception des sites Web dynamiques moyennant un serveur HTTP, Il s'agit d'un langage de programmation sous licence libre. PHP est un langage orienté objet.

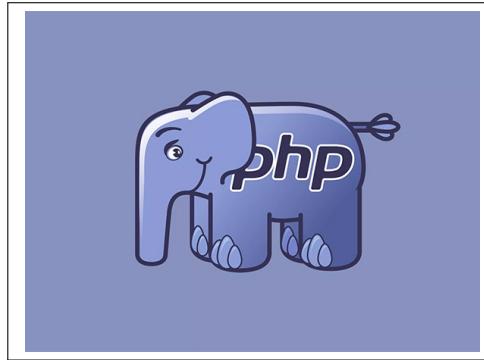


Figure 2.9: PHP [8]

— **Postman :**

Postman est un logiciel qui se focalise sur les tests des API en interrogeant les micro Services, offrant ainsi de nombreuses fonctionnalités, une prise en main rapide et une interface graphique agréable.



Figure 2.10: Postman [9]

— **Git (Version Control) :**

Tout d'abord, GitHub / GitLab n'est pas git. Beaucoup de gens confondent entre les deux. GitHub / GitLab est un site Web pour l'hébergement de projets utilisant git. Git est un type VCS qui facilite le suivi des modifications apportées aux fichiers. Il est utile pour coordonner le travail entre plusieurs personnes sur un projet, et pour suivre les progrès dans le temps en enregistrant des «points de contrôle».



Figure 2.11: Git [10]

— **Visual Paradigm :**

Visual Paradigm (VP) est un logiciel de création de diagrammes dans le cadre d'une programmation. Il possède plusieurs fonctionnalités qui facilite énormément la modélisation en ULM.



Figure 2.12: Visual Paradigm [11]

— **Sentry :**

Sentry est principalement un service qui aide à suivre et à réparer les plantages en temps réel. Le serveur est en Python, mais il contient une API complète pour envoyer des événements à partir de n'importe quelle langue, dans n'importe quelle application



Figure 2.13: Sentry [12]

— **Slack :**

Slack est une plate-forme de communication collaborative propriétaire (SaaS) permettant la communication entre tous les membres de CosaVostra et aussi utilisée comme outil de gestion de projet



Figure 2.14: Slack [13]

— **Trello :**

Trello est un outil de gestion de projet , lancé en septembre 2011. Il repose sur une organisation des projets en planches listant des cartes, chacune représentant des tâches. Chaque carte, dans notre projet, représente un sprint et dans chaque sprint (carte) on trouve les tâches associées avec une date de début et une date de fin.



Figure 2.15: Trello [14]

2.7 Planification du sprint

Comme nous utilisons Scrum dans notre travail, et après une compréhension approfondie du backlog produit nous avons décidé de diviser notre travail en 3 sprints principaux.

Tableau 2.3: Planification du sprint

Sprint	Fonctionnalités	Durée (en jour)
Sprint 1	Authentification. Gestion des devis.	De 18 Mars à 17 Avril
Sprint 2	Gestion des contacts. Enregistrer contacts.	De 23 Avril à 25 Mai
Sprint 3	Gestion des factures. Gestion des objectifs. Gestion des statistiques.	De 29 Mai à 30 Juin

Conclusion

Dans ce chapitre, nous avons planifié notre travail, identifié les besoins fonctionnels et non fonctionnels de notre application, les rôles des utilisateurs, par la suite nous avons présenté le Backlog de notre système. Ainsi nous avons détaillé la phase de planification des sprints. Enfin nous avons choisi l'architecture de notre projet ainsi que notre environnement de travail.

SPRINT 1

Plan

1	Objectif du Sprint	27
2	Analyse	27
3	Analyse spécifique	28
4	Conception	35
5	Réalisation et Test	37

Introduction

Dans ce chapitre nous allons concevoir et réaliser les fonctionnalités du sprint 1 de notre projet. Nous allons en premier lieu présenter le sprint backlog, où nous allons détailler les fonctionnalités demandées, l'étape suivante consiste à faire l'analyse et la conception des fonctionnalités spécifiées. En fin, nous montrons l'output de ce sprint.

3.1 Objectif du Sprint

Dans ce sprint, nous nous concentrerons sur la création de l'archétype du projet et la mise en œuvre des différents outils d'intégration, serveurs et outils de gestion de projet. Sécuriser ensuite l'application à l'aide du ressort sécurité et JWT en d'autres termes préparer des modèles d'authentification. Ce sprint est dédié principalement à l'authentification de l'administrateur de notre application , la synchronisation des devis grâce au API SELLSY.

3.2 Analyse

Cette partie sera consacrée à l'analyse du premier sprint de notre projet, nous allons présenter les backlog produit de sprint 1, les diagrammes de cas d'utilisation et leurs descriptions textuelles ainsi que le diagramme d'activité et le diagramme séquence système.

3.2.1 Sprint backlog

Le backlog sprint est une liste des tâches identifiées par l'équipe Scrum à réaliser durant du premier sprint. Lors de la réunion de planification du sprint, l'équipe sélectionne un certain nombre de produits des éléments de backlog, généralement sous la forme de user stories, et identifie les tâches nécessaires pour user stories. La plupart des équipes estiment également combien d'heures chaque tâche nécessitera achevée. Dans la figure suivante, nous avons l'arrière du sprint 1, l'estimation est en jours.

Tableau 3.1: Backlog du sprint 1

ID	User Stories	Estimation (en jour)
1	En tant qu'admin je veux m'authentifier.	5
2	En tant qu'admin je veux modifier mon mot de passe.	2
3	En tant qu'admin je veux afficher les devis.	2
4	En tant qu'admin je veux modifier un devis.	5
5	En tant qu'admin je veux exporter les devis en csv.	5

3.3 Analyse spécifique

3.3.1 Diagramme des cas d'utilisation

La figure 3.1 présente le diagramme de cas d'utilisation du premier sprint.

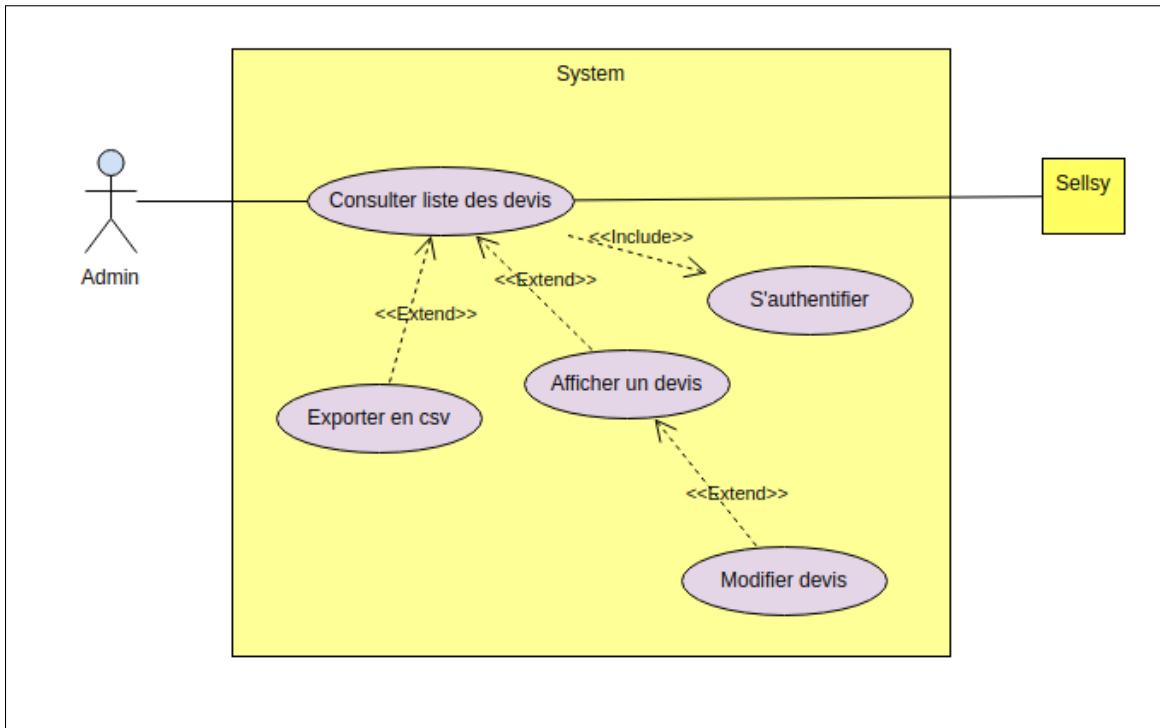


Figure 3.1: Diagramme de cas d'utilisation global

3.3.1.1 Raffinement du cas d'utilisation «S'authentifier»

La figure 3.3 présente le raffinement du diagramme d'utilisation « s'authentifier ».

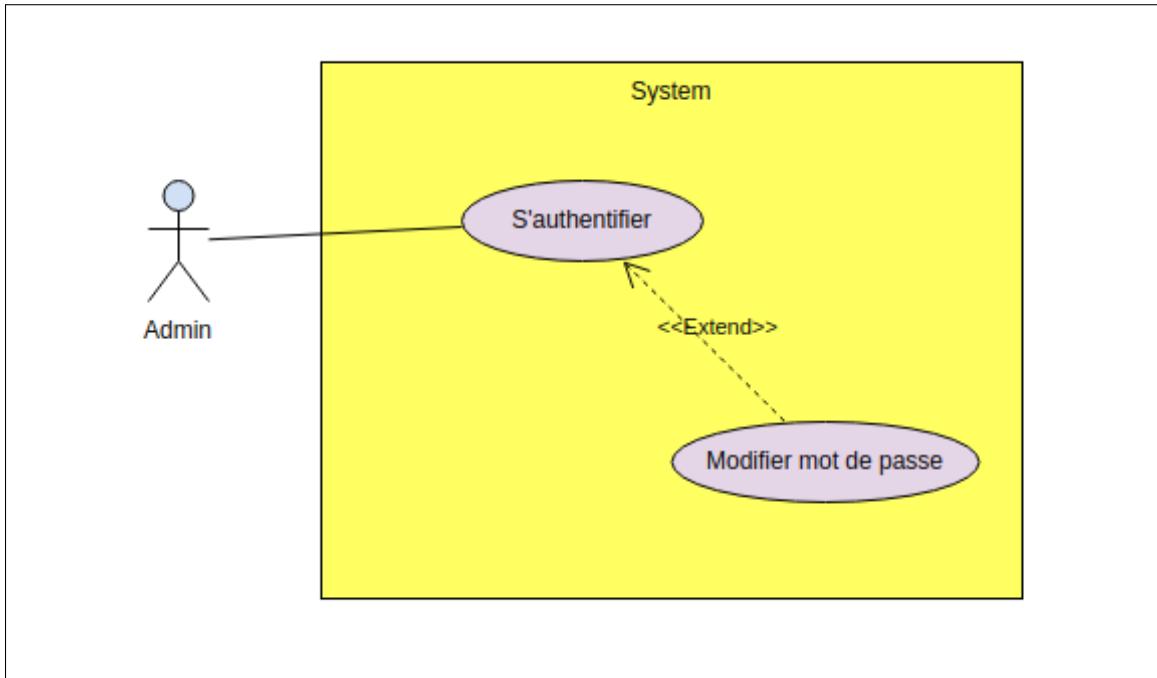


Figure 3.2: Diagramme de cas d'utilisation «S'authentifier»

La description détaillée de cas d'utilisation « s'authentifier » est donnée par le tableau suivant :

Tableau 3.2: Description textuelle du cas «S'authentifier »

Acteur	Admin
Objectif	S'authentifier
Pré-condition	Admin approuvé
Scénario nominal	<ul style="list-style-type: none"> -L'utilisateur saisie son e-mail et sa mot de passe. -L'utilisateur confirme la saisie de ses données d'identification. -Le système vérifie les données d'identification. -Le système affiche l'interface d'accueil.
Post-condition	Utilisateur authentifié
Exception	Si le login et/ou le mot de passe sont incorrects. Le système affiche un message d'erreur.
Extension	En cas d'oubli ou de modification de mot de passe l'admin peut redéfinir un nouveau mot de passe.

Le tableau ci-dessous représente le cas de redéfinir un nouveau mot de passe.

Tableau 3.3: Description textuelle du cas «réinitialisation mot de passe »

Acteur	Utilisateur
Objectif	Redéfinir un nouveau mot de passe
Pré-condition	Utilisateur ayant oublié son mot de passe
Scénario nominal	<ul style="list-style-type: none"> - L'utilisateur accède à la page de connexion -L'utilisateur appuie sur le bouton « Mot de passe oublié ? » -L'utilisateur saisi son adresse email -L'utilisateur reçoit un email sur l'adresse entré -L'utilisateur appuie sur le lien envoyé dans l'email -L'utilisateur est redirigé vers une page de réinitialisation de mot de passe où il entre un nouveau mot de passe et confirme
Post-condition	Changer de mot de passe

Exception	Si l'email saisi n'existe pas dans la base de données une alerte s'affiche (message d'erreur)
-----------	---

3.3.1.2 Raffinement du cas d'utilisation «Gestion des devis»

La figure suivante présente le raffinement de l'item « Gérer devis » par l'administrateur.

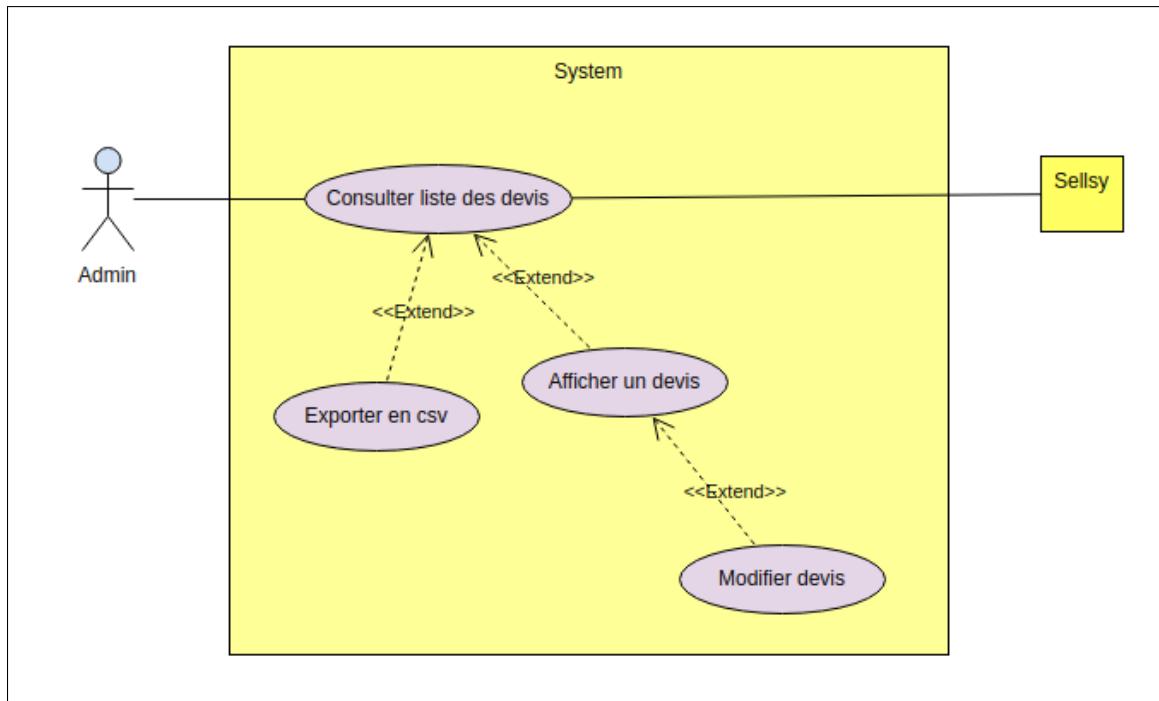


Figure 3.3: Diagramme de cas d'utilisation «Consulter les devis»

Ce tableau décrit le cas d'utilisation «Consulter les devis».

Tableau 3.4: Consulter les devis «Consulter les devis»

Acteur	Admin
Objectif	Consulter les devis.
Pré-condition	L'admin doit être s'authentifié.
Scénario nominal	-L'admin demande de consulter la page des devis. -Le système affiche les détails des devis.
Post-condition	Consulter les devis.
Extension	L'admin peut exporter la liste des devis. L'admin peut afficher un devis.

Ce tableau décrit le cas d'utilisation d'exportation des devis.

Tableau 3.5: Description textuelle du cas «Exporter les devis »

Acteur	Admin
Objectif	Exporter les devis.
Pré-condition	L'admin doit être s'authentifié. La liste des devis consulté.
Scénario nominal	-L'admin demande d'exporter la liste des devis. -Le système télécharge le fichier.
Post-condition	Les devis sont exportés.

Le tableau 3.6 décrit le cas d'utilisation consulter un devis.

Tableau 3.6: Description textuelle du cas «Afficher un devis »

Acteur	Admin
Objectif	Afficher un devis.
Pré-condition	L'admin doit être s'authentifié. La liste des devis consulté.
Scénario nominal	-L'admin demande de consulter un devis. -Le système affiche un devis.
Post-condition	Un devis afficher.
Extension	L'admin peut modifier un devis.

Le tableau 3.7 présente la description textuelle du cas d'utilisation «Modifier un devis».

Tableau 3.7: Description textuelle du cas «Modifier un devis»

Acteur	Admin
Objectif	Modifierr un devis.
Pré-condition	L'admin doit être s'authentifié. La liste des devis consulté.

Scénario nominal	<ul style="list-style-type: none"> -L'admin demande de modifier un devis. -L'admin modifie la devis. -Le système enregistre les modifications effectuées. -Le système notifié l'admin.
Post-condition	Un devis modifier.

3.3.2 Diagramme des séquences

Dans le diagramme qui présenté dans la figure 3.4 nous montrons comment l'ajout des données.

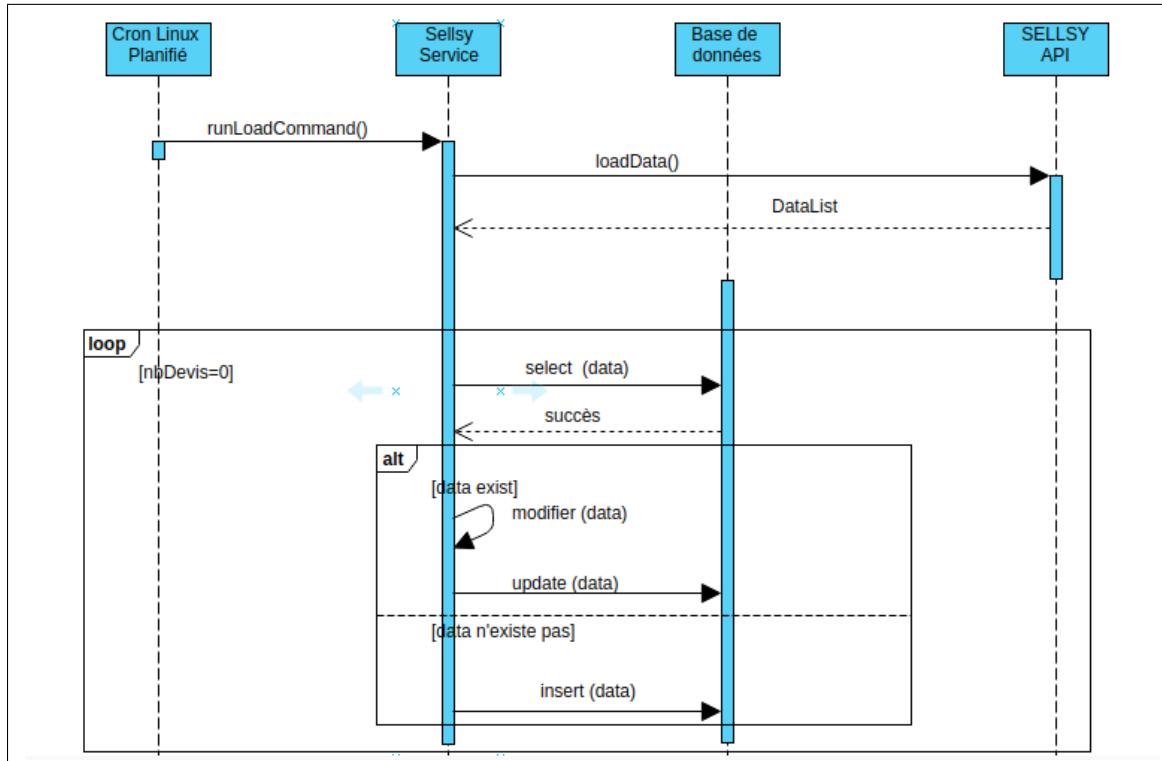


Figure 3.4: Diagramme de séquence objet du cas « Consulter devis Backend »

Nous présentons ci-dessous le diagramme des séquences objets relative au cas d'utilisation s'authentifier.

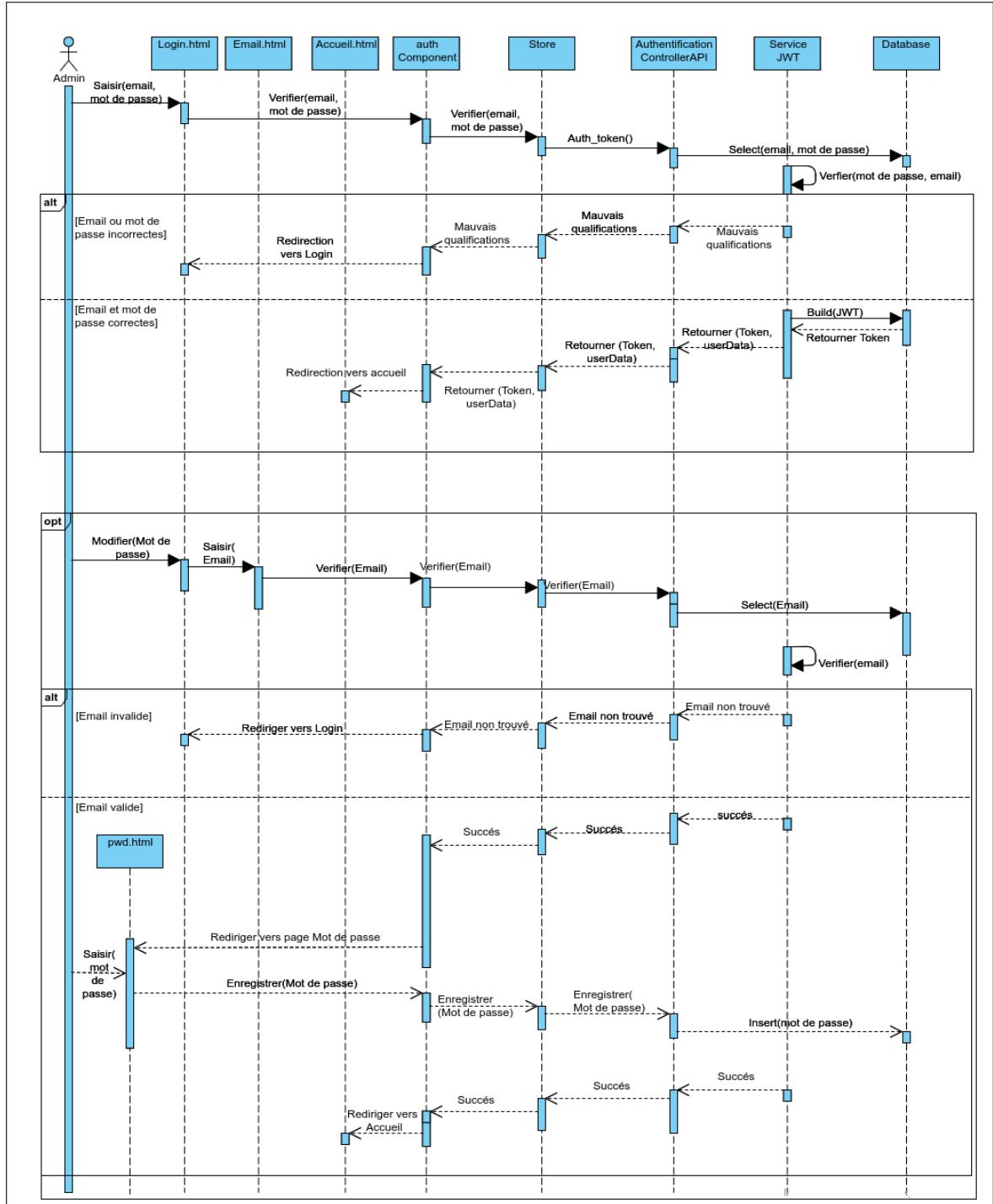


Figure 3.5: Diagramme de séquence objet du cas « S'authentifier »

La figure 3.6 illustre la séquence schéma de cas d'utilisation «S'authentifier».

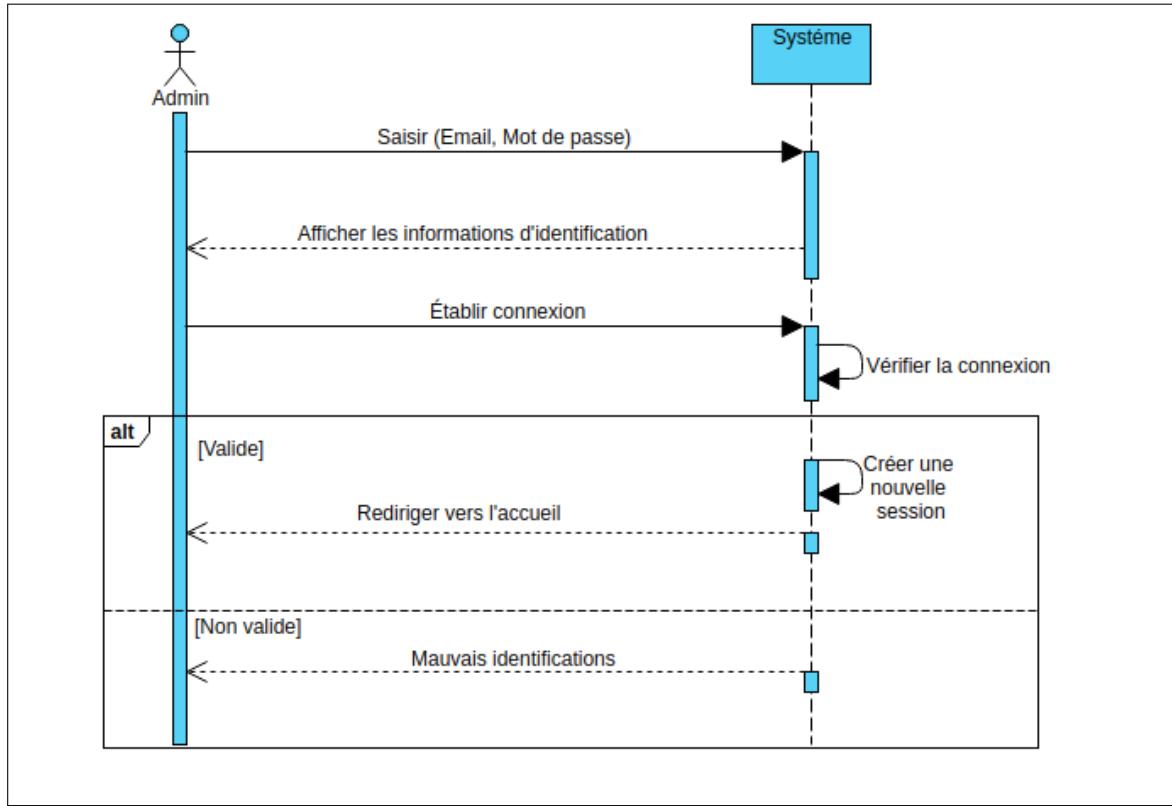


Figure 3.6: Diagramme de séquence système du cas «Authentification»

3.4 Conception

3.4.1 Diagramme de classes

La figure ci-dessous illustre le diagramme des classes utilisées durant le premier sprint.

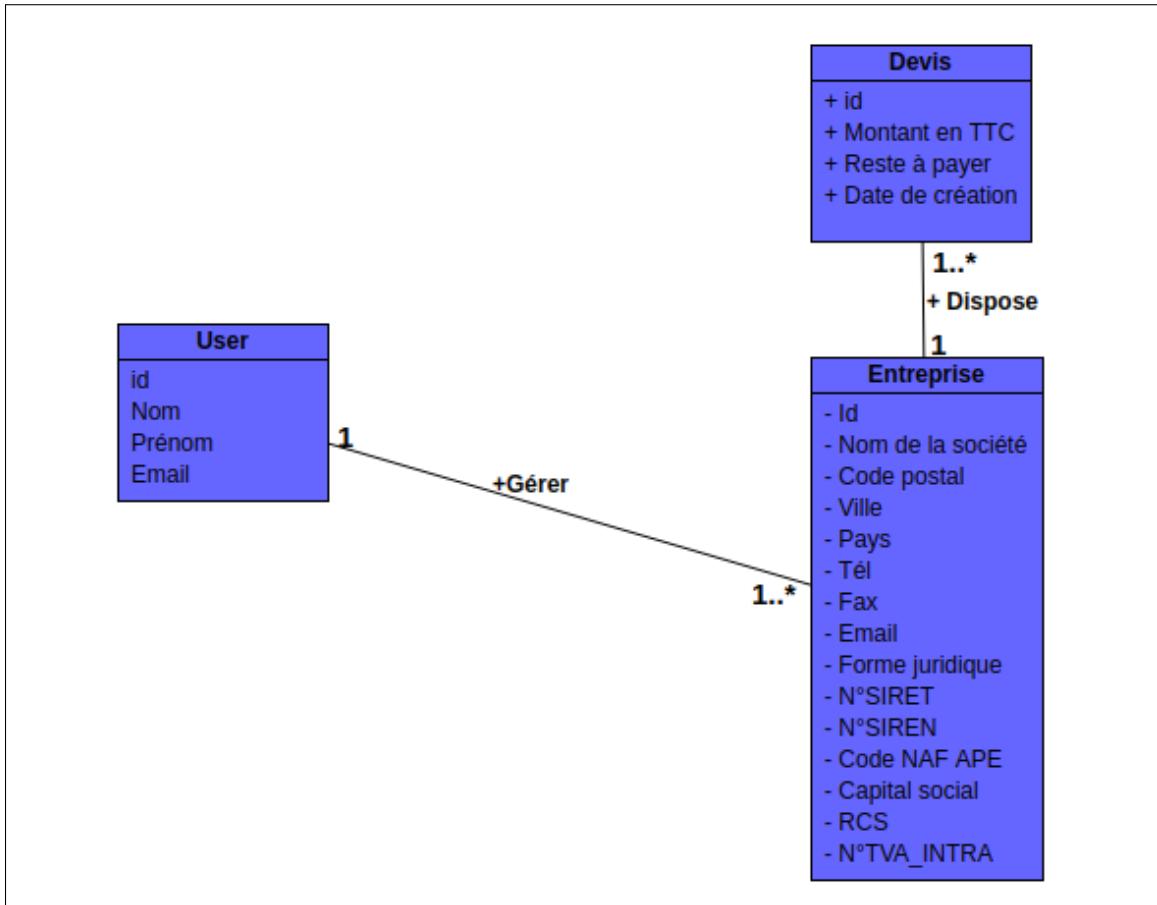


Figure 3.7: diagramme de classe du sprint 1

3.4.2 Diagramme d'activité

Notre diagramme d'activité présente la premier étape pour consulter l'accueil de notre plateforme est l'authentification.

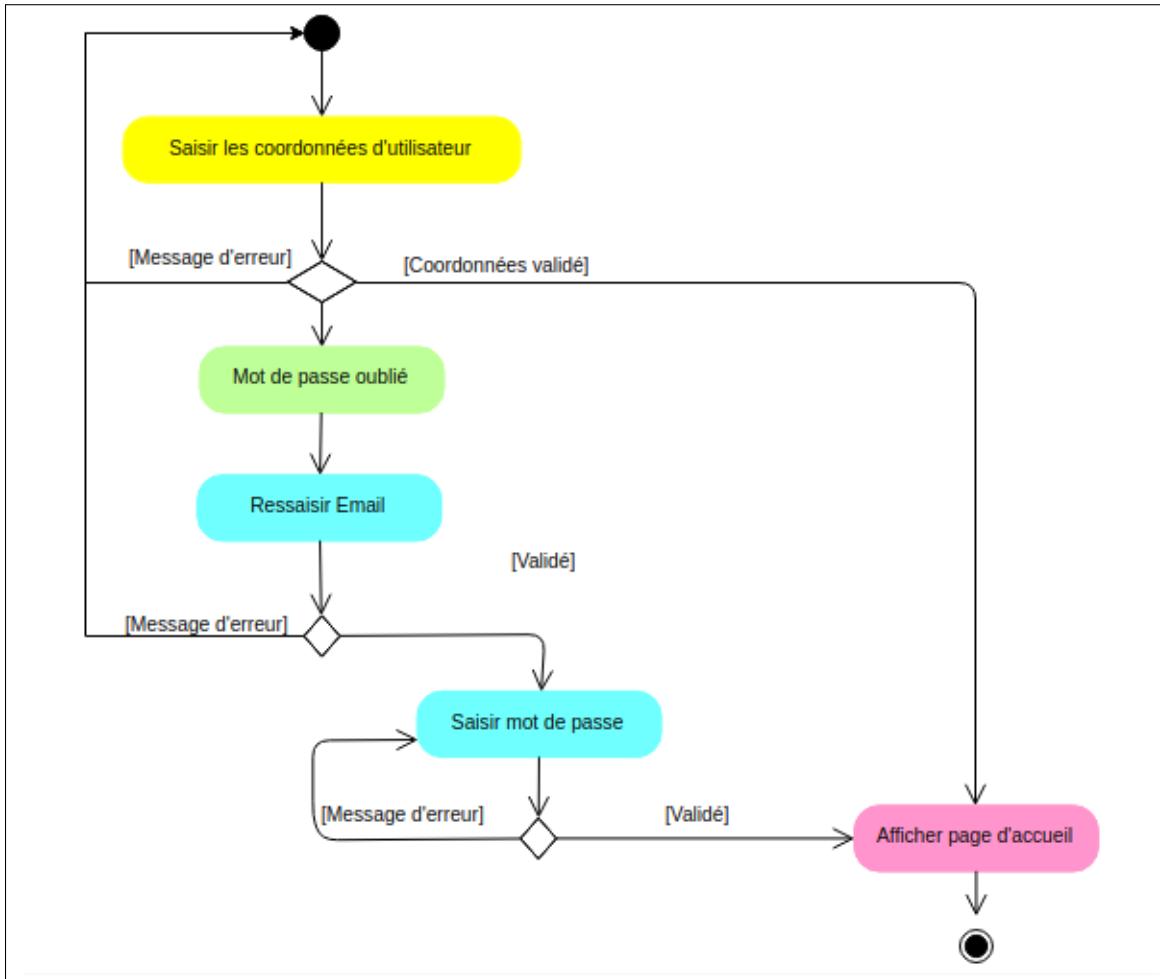


Figure 3.8: Diagramme d'activité du cas d'utilisation «S'authentifier»

3.5 Réalisation et Test

Dans cette partie nous présentons quelques interfaces représentant le travail élaboré dans ce sprint.

3.5.1 S'authentifier

L'accès à la page principale du site s'effectue à l'aide de l'authentification. C'est la première interface de notre plateforme l'authentification. Pour se connecter, l'utilisateur saisit son login et mot de passe et coche son statut soit prestataire ou administrateur.

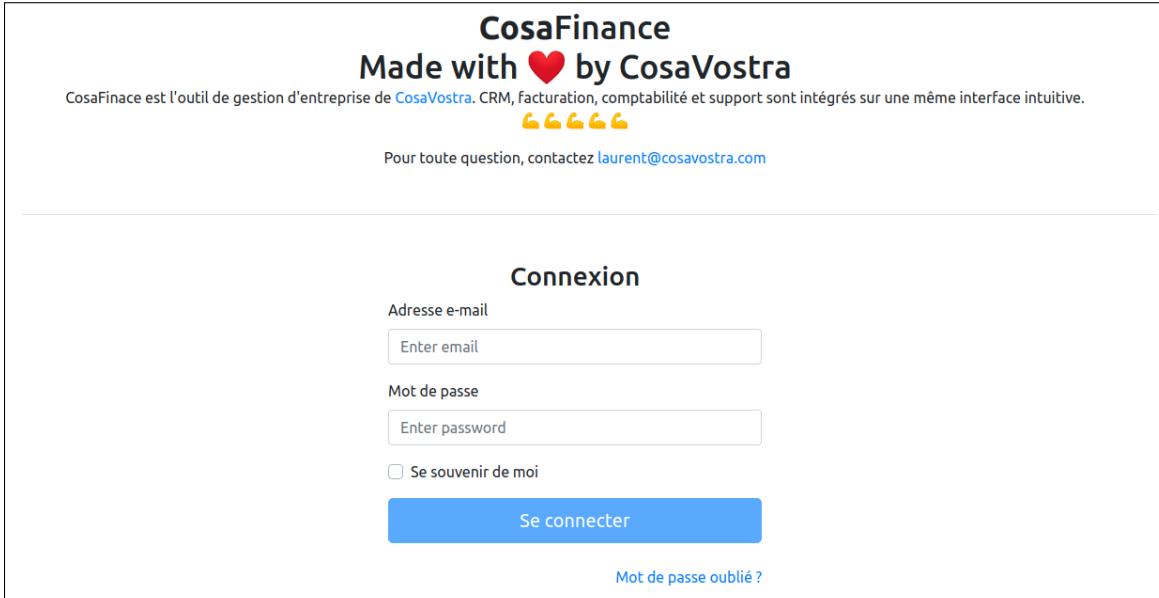


Figure 3.9: Authentification

En cas d'incorrectes identifiants le système affiche de message.

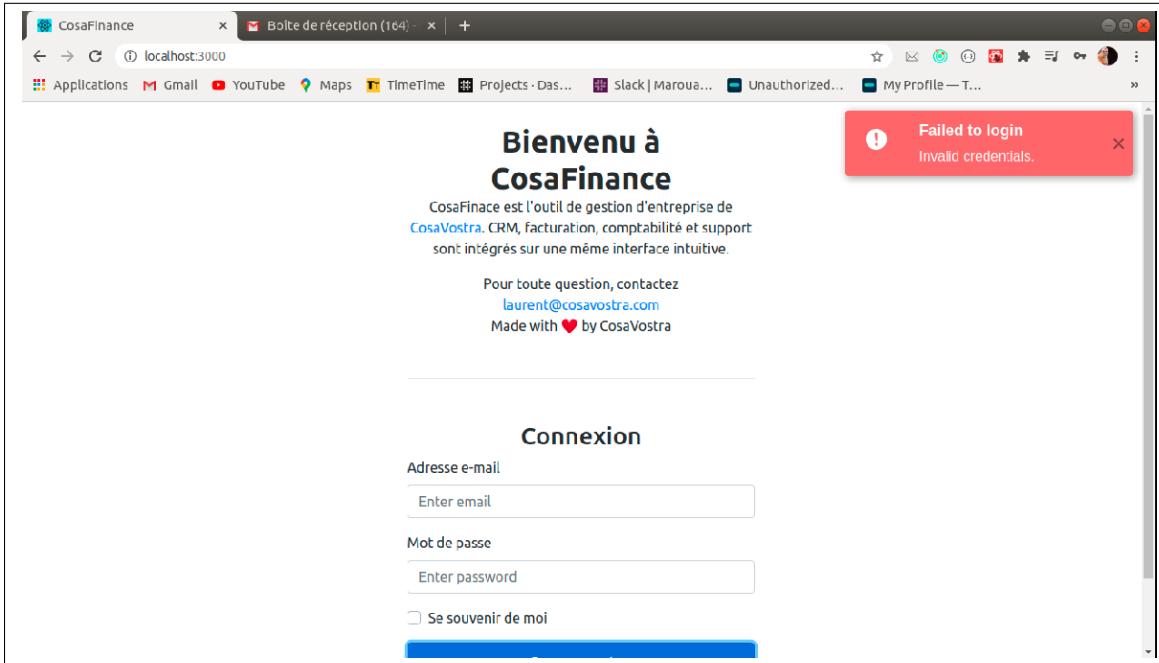


Figure 3.10: Message d'erreur

3.5.2 Mot de passe oublié

Si l'utilisateur oublie son mot de passe il peut le réinitialiser comme suite (après avoir appuyésur le bouton Mot de passe oublié?) :

Chapitre 3. Sprint 1

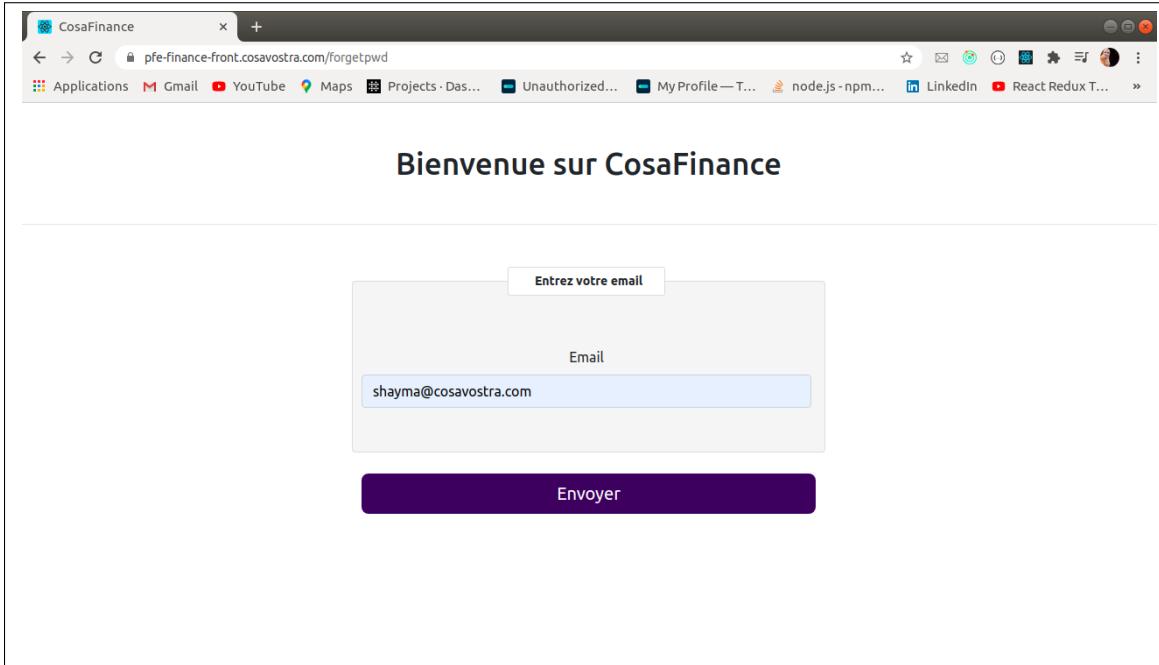


Figure 3.11: Saisir email

Après avoir appuyé sur le bouton Envoyer le système affiche un message de succès et envoie un email à l'adresse entrée

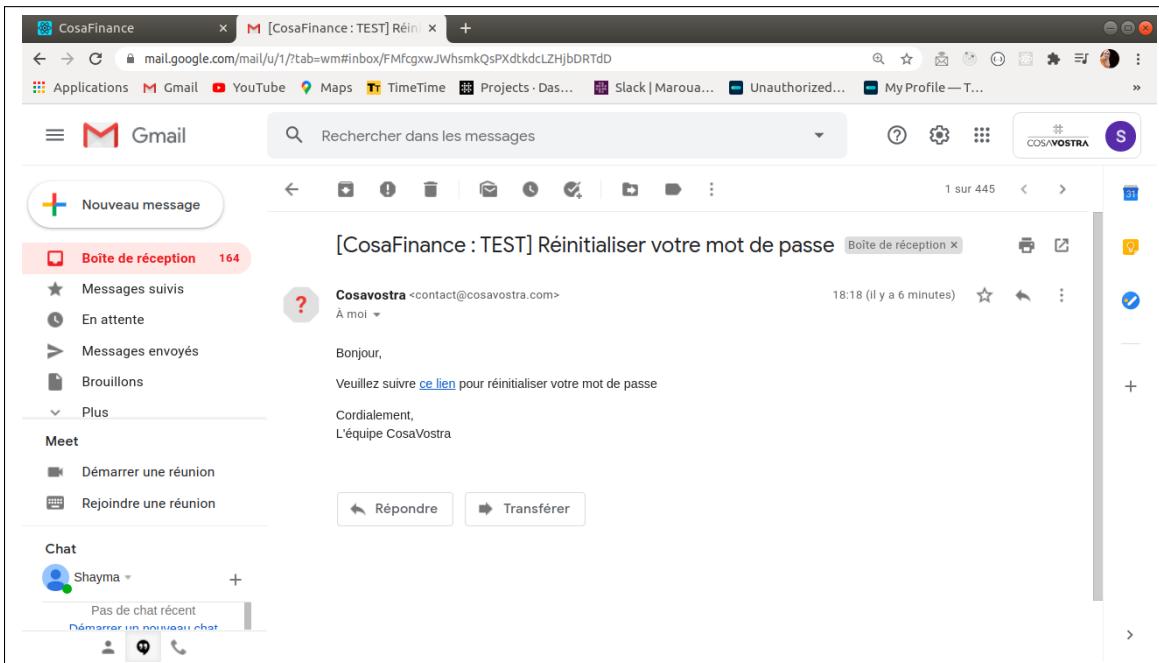


Figure 3.12: Email envoyé

En appuyant sur "ce lien" dans l'email ci-dessus l'utilisateur est redirigé vers cette interface de réinitialisation de mot de passe :

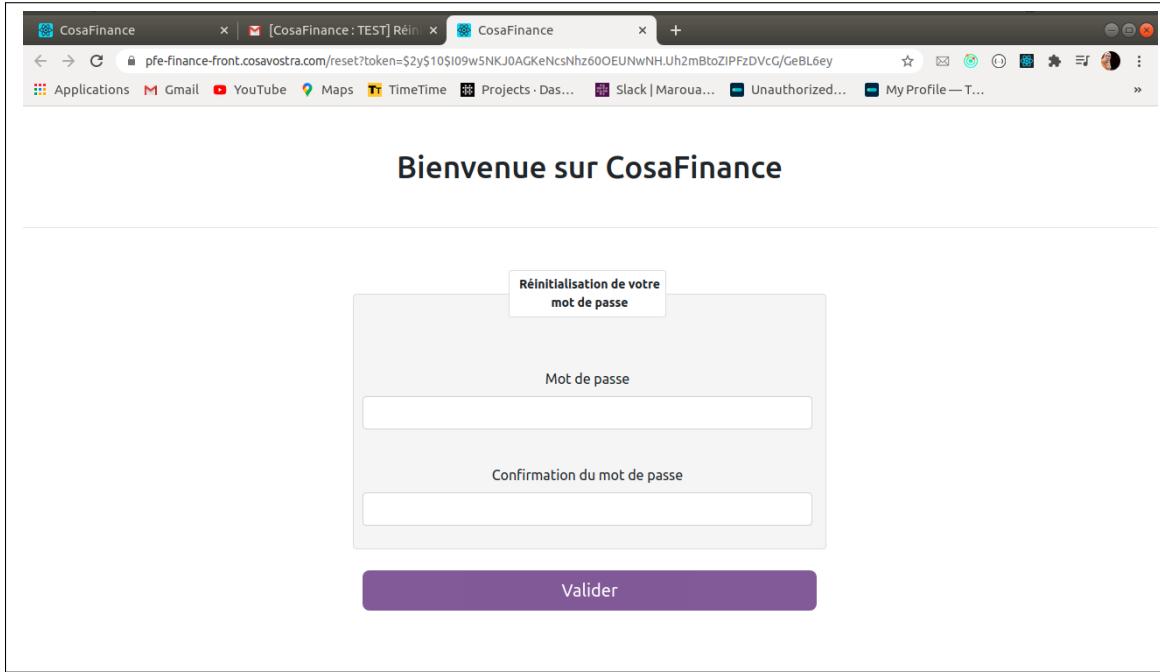


Figure 3.13: Email envoyé

S'ils sont identiques le mot de passe est changé.

3.5.3 Consulter la liste des devis

En tant qu'administrateur, il permet de consulter la liste des devis sur notre plateforme ainsi que les détails de chaque devis tel que la date de devis le reste à payer le pourcentage de paiements..

Devis								
Devis List								
N°devis	Status	Client	Montant_TTC	Facturé_TTC	Reste_à_facturé	Date	%paiement	%du_reste_à_payer
DEV-20200421-00012	Partiellement facturé	DASSAULT SYSTEMES	1 824,00 €	824€	1 000,00 €	21/04/2020	45.17%	54.82%
DEV-20200409-00007	Facturé	ERGACIA Shayma	99,98 €	99€	0,00 €	09/04/2020	100%	0%

Figure 3.14: Liste des devis

Cette interface permet à l'administrateur de consulter la liste de tous les devis de modifier

Chapitre 3. Sprint 1

et l'exporter sous la forme csv.

Client	Montant_TTC	Facturé_TTC	Reste_à_facturé	Date	%paiement	%du_reste_à_payer	commentaire	Actions
ROUSSELOT SAS	1 170,00 €			24/06/2020				
TAG HEUER INTERNATIONAL	47 400,00 €	9 480,00 €	37920€	22/06/2020	80%	20%		
TAG HEUER INTERNATIONAL	79 700,00 €	15 940,00 €	63760€	22/06/2020	80%	20%		
ANDROS	6 945,00 €			11/06/2020				
Coin House	5 520,00 €	0,00 €	5520€	08/06/2020	100%	0%		

Figure 3.15: Liste des devis

Conclusion

Dans ce chapitre, nous avons créé l'artefact du projet avec un modèle principal, l'artefact les services sont sécurisés à l'aide de JWT. L'artefact comprend des services d'authentification, page de connexion, services pour l'importation des devis. Dans le chapitre qui suit, nous allons produire un nouveau sprint couvrant les fonctionnalités suivantes : Gestion des contacts de facturation et valider des nouveaux contacts.

SPRINT 2

Plan

1	Objectif du Sprint	43
2	Analyse	43
3	Analyse spécifique	44
4	Conception	52
5	Réalisation et test	54

Introduction

Dans ce chapitre nous allons passer vers la réalisation du sprint 2 de notre projet, qui est la gestion des contacts de facturation et de valider des nouveaux contacts. Nous allons en premier lieu présenter le sprint backlog, où nous allons détailler les fonctionnalités demandées, l'étape suivante consiste à faire l'analyse et la conception de notre application, finalement nous allons faire la partie conception.

4.1 Objectif du Sprint

Ce chapitre est consacré pour l'administrateur et le client de notre application . A la fin de ce chapitre, nous devons être capables de gérer les contacts de facturations attachés à notre application grâce à l'API SELLSY et aussi de valider des nouveaux contacts.

4.2 Analyse

Cette partie sera consacrée à l'analyse du premier sprint de notre projet, nous allons présenter les backlog produit de deuxième sprint.

4.2.1 Backlog du sprint 2

Nous allons présenter dans le tableau suivant le backlog du sprint 2 traduisant les différentes fonctionnalités achevées de notre application durant ce sprint.

Tableau 4.1: Backlog du sprint 2

ID	User Stories	Estimation (en jour)
6	En tant qu'au client je veux enregistrer contacts à l'application	5
7	En tant qu'admin je veux être informé des nouveaux clients par email.	0.5
8	En tant qu'admin je veux accepter des nouveaux des clients.	4
9	En tant qu'admin je veux modifier les contacts de facturation.	4

10	En tant qu'admin je veux exporter les contacts de facturation en format csv.	4
11	En tant qu'admin je veux attribuer les contacts ajoutées à des entreprises déjà importé à travers sellsy.	3
12	En tant qu'admin je veux ajouter une contacts facturation vers SELLSY.	1.5

4.3 Analyse spécifique

Cette section est consacrée au raffinement des cas d'utilisation de ce sprint en se basant sur les diagrammes des cas d'utilisation raffinés, une description textuelle des cas d'utilisation et leurs diagrammes de séquences système.

4.3.1 Diagramme de cas d'utilisation

La figure ci-dessous présente le cas d'utilisation globale qui illustre l'ensemble de fonctionnalités à faire tout au long de ce sprint.

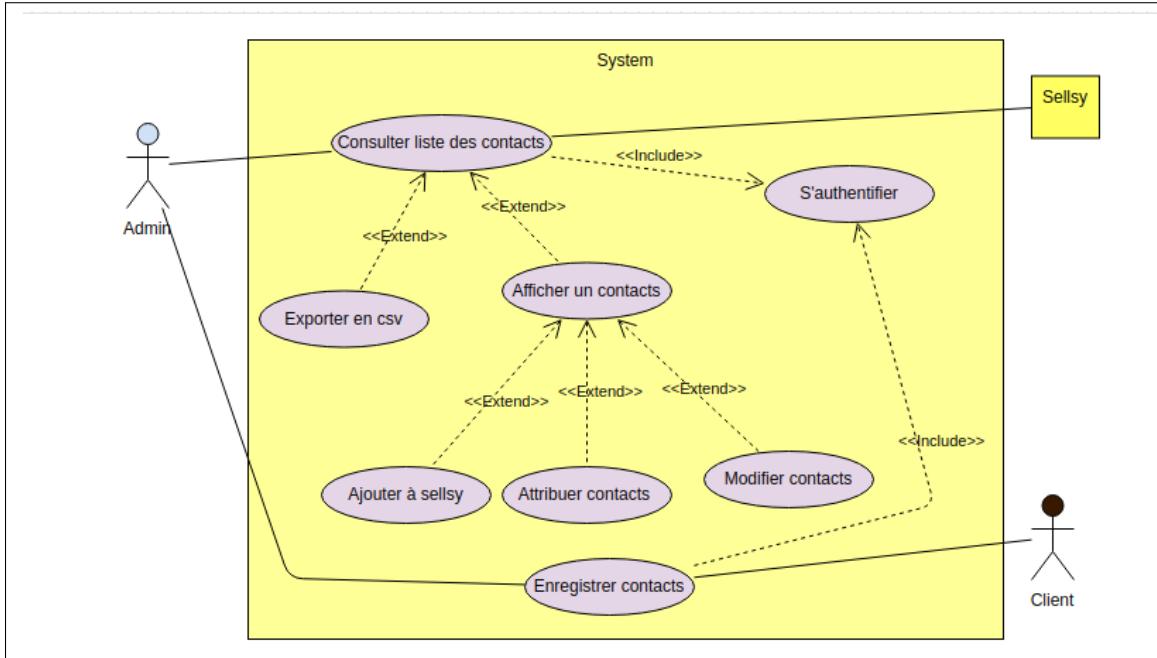


Figure 4.1: Raffinement du cas d'utilisation : Sprint2

4.3.1.1 Raffinement du cas d'utilisation «Enregistrer contacts»

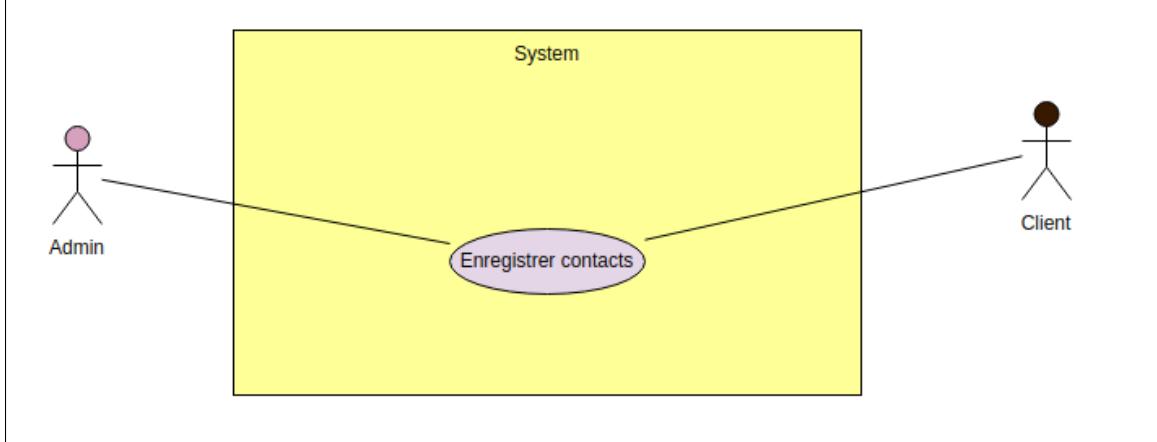


Figure 4.2: Raffinement du cas d'utilisation «Attribué une contacts»

Dans cette partie, nous allons présenter une description textuelle des cas «Enregistrer contacts»

Tableau 4.2: Description textuelle du cas «Enregistrer contacts»

Acteur	Client
Objectif	Envoyer la demande d'inscription à notre administrateur de l'application
Pré-condition	Contacts non ajouter
Scénario nominal	<ul style="list-style-type: none"> -Le client consulte la page d'inscription du notre application. -Le client remplit le formulaire. -Le client clique sur Bouton «Envoyer ». -Le système enregistre la demande et affiche le message « Merci, votre demande sera traitée ». -Le système envoie un email à l'admin de l'application pour infomés les nouveaux contacts. - L'admin peut accepter des nouveaux contacts. -Le système enregistre l'opération.
Post-condition	Demande d'inscription envoyée.
Exception	Le système affiche un message d'erreur «Vérifiez votre saisie ».

4.3.1.2 Raffinement du cas d'utilisation «Consulter des contacts»

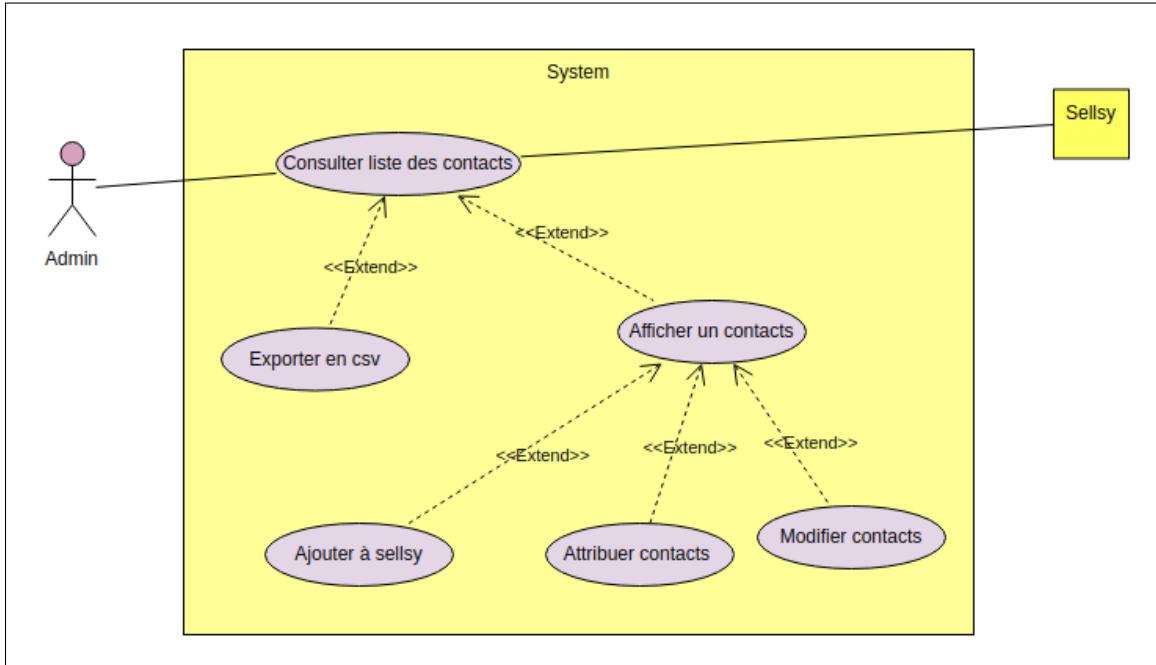


Figure 4.3: Raffinement du cas d'utilisation «Consulter des contacts»

Le tableau 4.3 représente la description textuelle du cas d'utilisation «Consulter les contacts».

Tableau 4.3: Description textuelle du cas «Consulter les contacts»

Acteur	Admin
Objectif	Consulter les contacts.
Pré-condition	L'admin doit être s'authentifié.
Scénario nominal	<ul style="list-style-type: none"> -L'admin demande de consulter la page des suivis des contacts de facturation. -Le système affiche les détails des contacts.
Post-condition	Consulter les contacts.
Extension	<ul style="list-style-type: none"> L'admin peut exporter la liste des contacts. L'admin peut afficher un contact.

Le tableau 4.4 représente la description textuelle du cas d'utilisation «Consulter une contact».

Tableau 4.4: Description textuelle du cas «Consulter une contact»

Acteur	Admin
Objectif	Consulter une contact
Pré-condition	L'admin doit être s'authentifié.
Scénario nominal	<ul style="list-style-type: none"> -L'admin demande de consulter une contact. -Le système affiche une contact.
Post-condition	Consulter une contact.
Extension	<ul style="list-style-type: none"> L'admin peut modifier une contact. L'admin peut attribué contact. L'admin peut ajouter à sellsy.

Dans cette partie, nous allons présenter une description textuelle des cas «Attribué une contacts».

Tableau 4.5: Description textuelle du cas «Attribué contact»

Acteur	Admin
Objectif	Attribué une contacts à une entreprise
Pré-condition	<ul style="list-style-type: none"> L'administrateur doit être authentifié. Nouveaux contact à été ajouté
Scénario nominal	<ul style="list-style-type: none"> -l'administrateur demande la consultation de la liste des contacts de facturation. -le système affiche la liste. -l'administrateur parcourt la liste des nouveaux contacts et au-dessous de chaque compte il y a button «Attribué» ou «Non Attribué» . -le système offre la possibilité d'attribué une contacts à une entreprise. -L'administrateur clique sur le bouton « Valider ». -Le système enregistre l'opération.

Post-condition	Contacts attribué à une entreprise
----------------	------------------------------------

Le tableau 4.6 représente la description textuelle du cas d'utilisation «Modifier un contact».

Tableau 4.6: Description textuelle du cas «Modifier un contact»

Acteur	Admin
Objectif	Modifierr un contact.
Pré-condition	L'admin doit être s'authentifié. La liste des contact consulté.
Scénario nominal	-L'admin demande de modifier un contact. -L'admin modifie la contact. -Le système enregistre les modifications effectuées. -Le système notifié l'admin.
Post-condition	Un contact modifier.

Le tableau 4.7 représente la description textuelle du cas d'utilisation «Ajouter contact à sellsy».

Tableau 4.7: Description textuelle du cas «Ajouter contact à sellsy»

Acteur	Admin
Objectif	Contact ajouté à sellsy
Pré-condition	L'admin doit être s'authentifié. La liste des contact consulté.
Scénario nominal	-L'admin demande d'ajouter un contact vers sellsy. -Le système enregistre l'opération. -Le système notifié l'admin.
Post-condition	Un contact ajouter.

Le tableau 5.5 représente la description textuelle du cas d'utilisation «Exporter les contacts».

Tableau 4.8: Description textuelle du cas «Exporter les contacts»

Acteur	Admin
Objectif	Exporter les contacts.
Pré-condition	L'admin doit être s'authentifié. La liste des devis consulté.
Scénario nominal	-L'admin demande d'exporter la liste des contacts. -Le système télécharge le fichier.
Post-condition	Les contacts sont exportés.

4.3.2 Diagramme de séquence

La figure ci-dessous présente le diagramme de séquence système du cas d'utilisation Attribution des contacts du facturation.

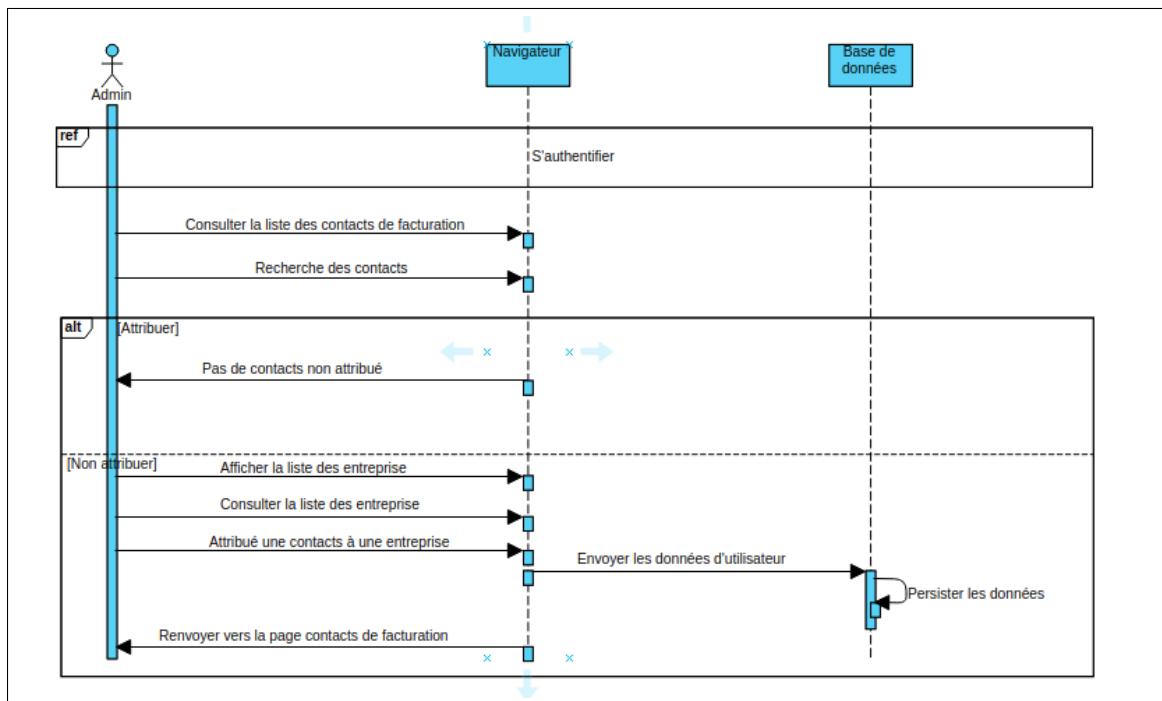


Figure 4.4: Diagramme de séquence système «Attribution des contacts du facturation»

Nous présentons ci-dessous le diagramme des séquences système relative au cas d'utilisation afficher les détails de contacts de facturation.

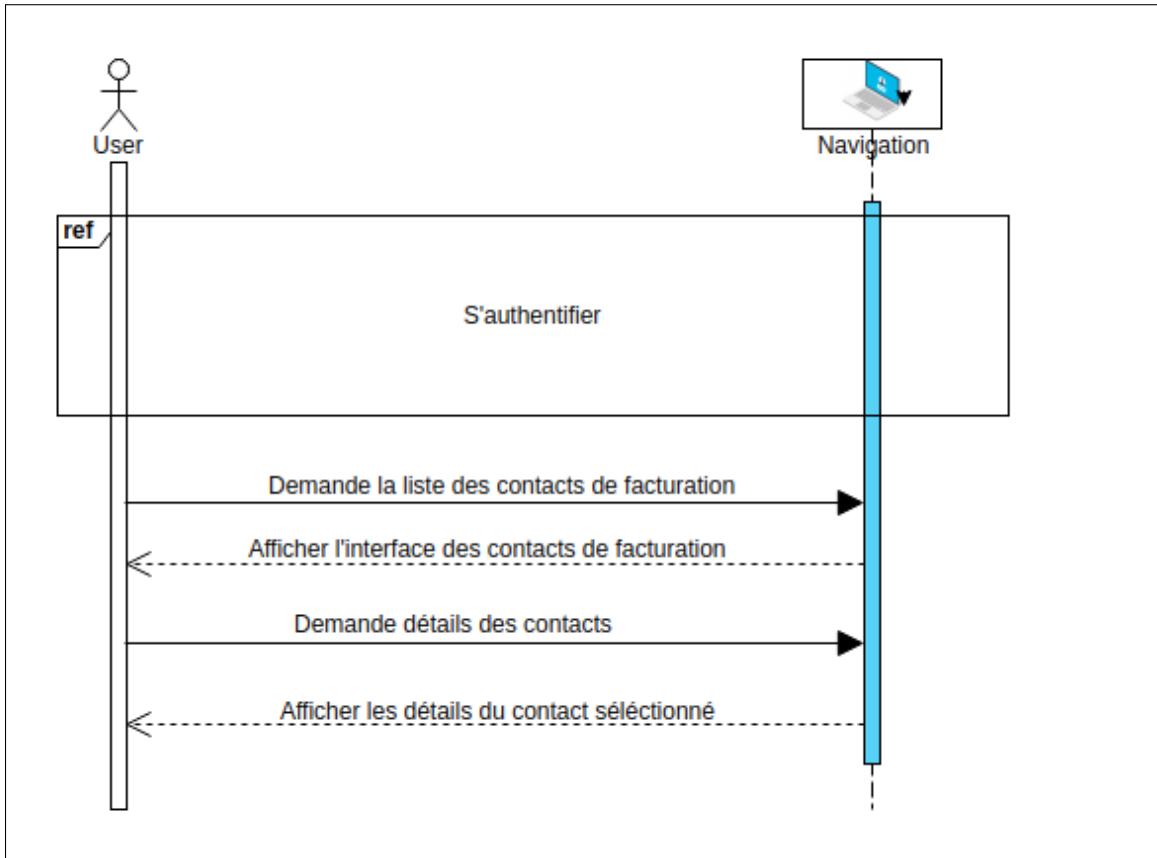


Figure 4.5: Diagramme de séquence système du cas «Consulter contacts»

4.3.3 Diagramme d'activité

La figure 4.6 présente l'enchaînement suivi suite à la gestion des contacts du facturations :

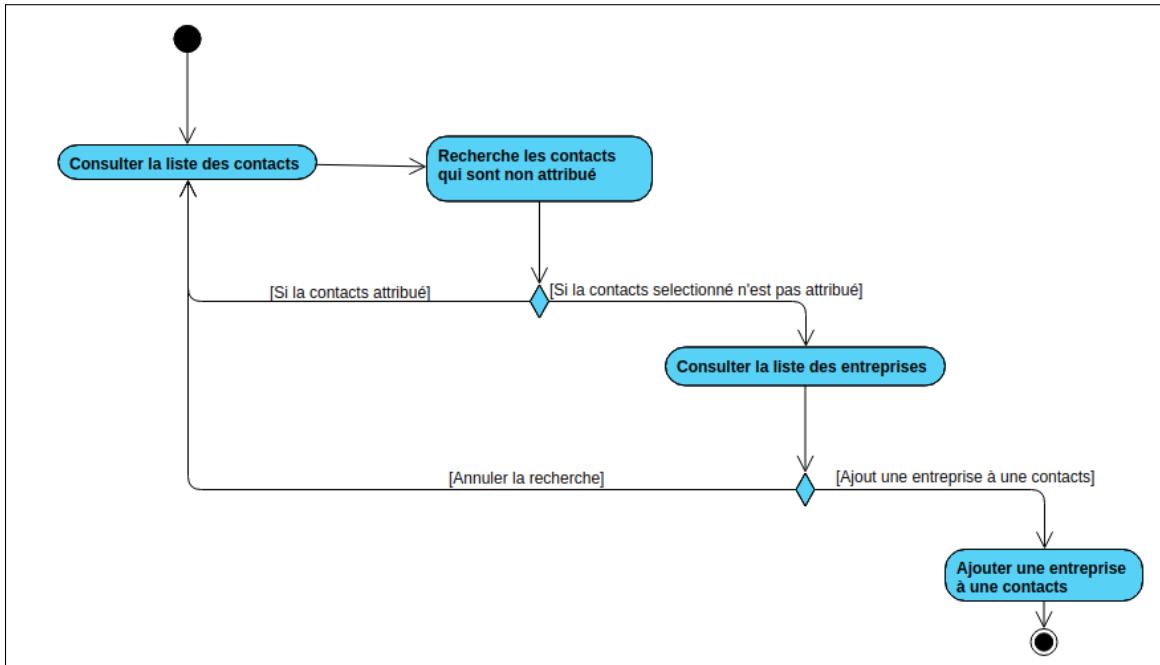


Figure 4.6: Diagramme d'activité du cas : Attribution des contacts du facturation

4.4 Conception

Cette partie est consacrée à la conception du sprint 2, pour ce faire nous commençons par le diagramme de classe détaillé et par la suite le diagramme de séquence objet.

4.4.1 Diagramme de classe globale du sprint 2

La figure ci-dessous présente le diagramme des classes utilisées durant la réalisation des fonctionnalités du deuxième sprint.

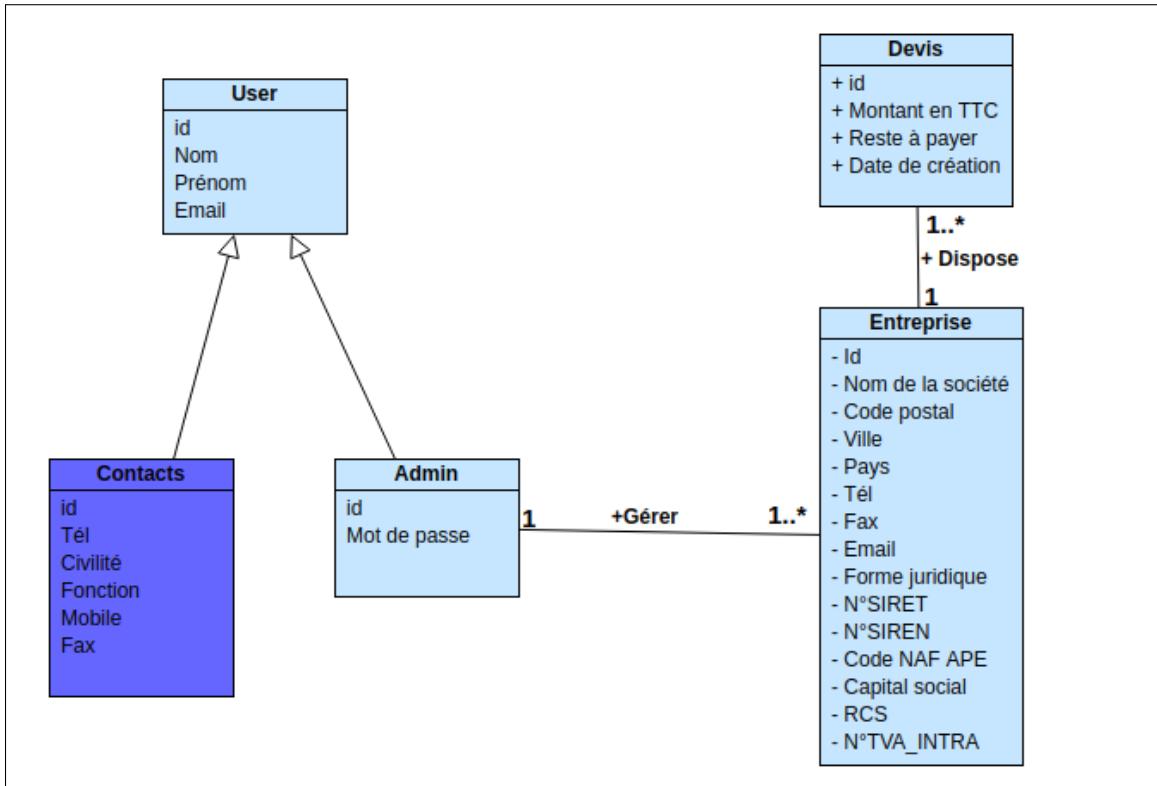


Figure 4.7: Diagramme de classe de conception

4.4.2 Diagramme de séquence objet

Nous présentons ci-dessous le diagramme des séquences objets relative au cas d'utilisation «Enregistrer contacts» coté client.

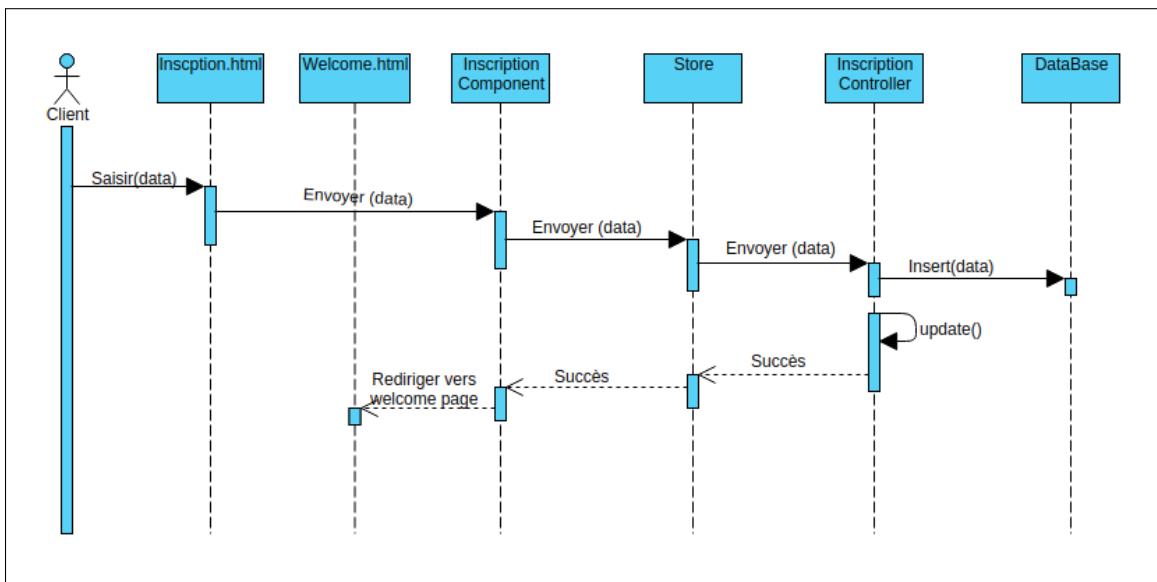


Figure 4.8: Diagramme de séquence système du cas «Enregister contacts»

4.5 Réalisation et test

Cette partie sera consacrée pour la réalisation des tâches du deuxième sprint de notre projet. Nous allons décrire le fonctionnement de l'application via les différentes interfaces réalisées dans ce sprint.

4.5.1 Inscription

Le client doit remplir le formulaire pour qu'il soit ajouté à notre application après l'acceptation du l'admin.

The screenshot shows a web browser window for 'CosaFinance' at the URL 'pfe-finance-front.cosavostra.com/inscription'. The page title is 'Bienvenue chez CosaVosstra'. The form is divided into two main sections: 'Informations générales' (General Information) on the left and 'Autres informations' (Other information) on the right.

Informations générales:

- Nom de la société: Entrer le nom de la société
- Référence: Entrer votre référence ici
- Site web: Entrer votre siteweb ici
- Catégorie tarifaire: Choisir...

Autres informations:

- Forme juridique: champ vide
- N° Siret: champ vide
- N° Siren: champ vide
- Code NAF (APE): champ vide
- Capital social: champ vide
- RCS: champ vide

Figure 4.9: Formulaire d'inscription

Après avoir rempli le formulaire un email sera envoyé automatiquement à l'admin de l'application.

Chapitre 4. Sprint 2

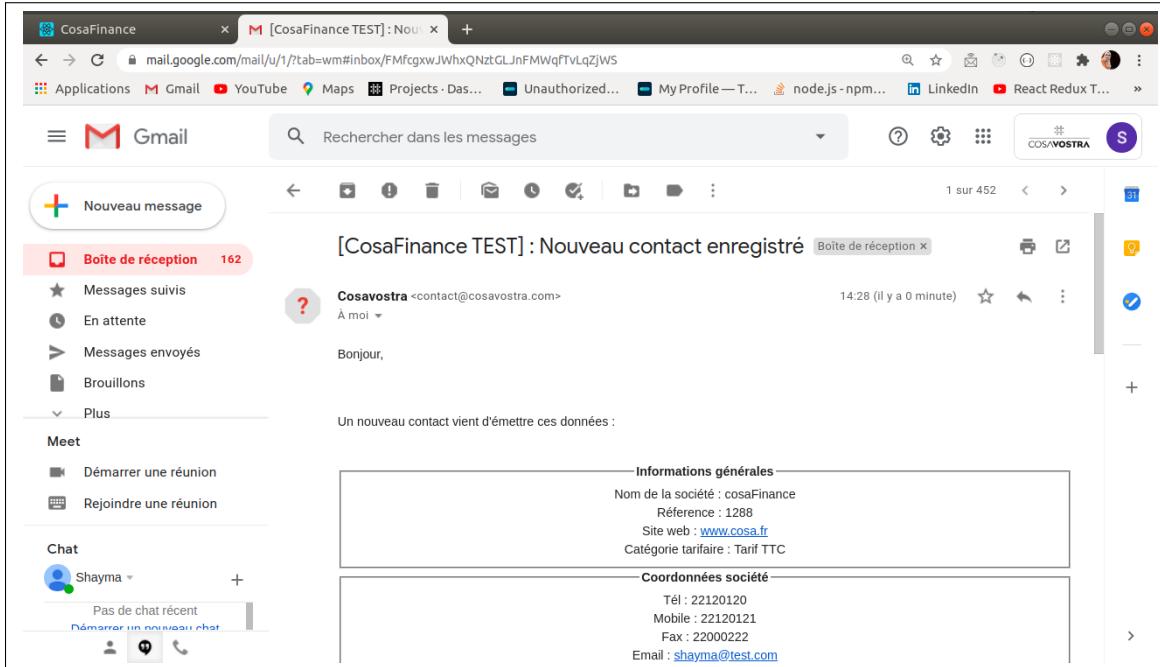


Figure 4.10: Email d'un nouveau contact

4.5.2 Accepter contact client

Une fois la demande du client enregistrer et envoyé, on passe à la liste des demandes comme le montre a figure ci-dessous l'admiistrateur peut acceper cette demande ou non :

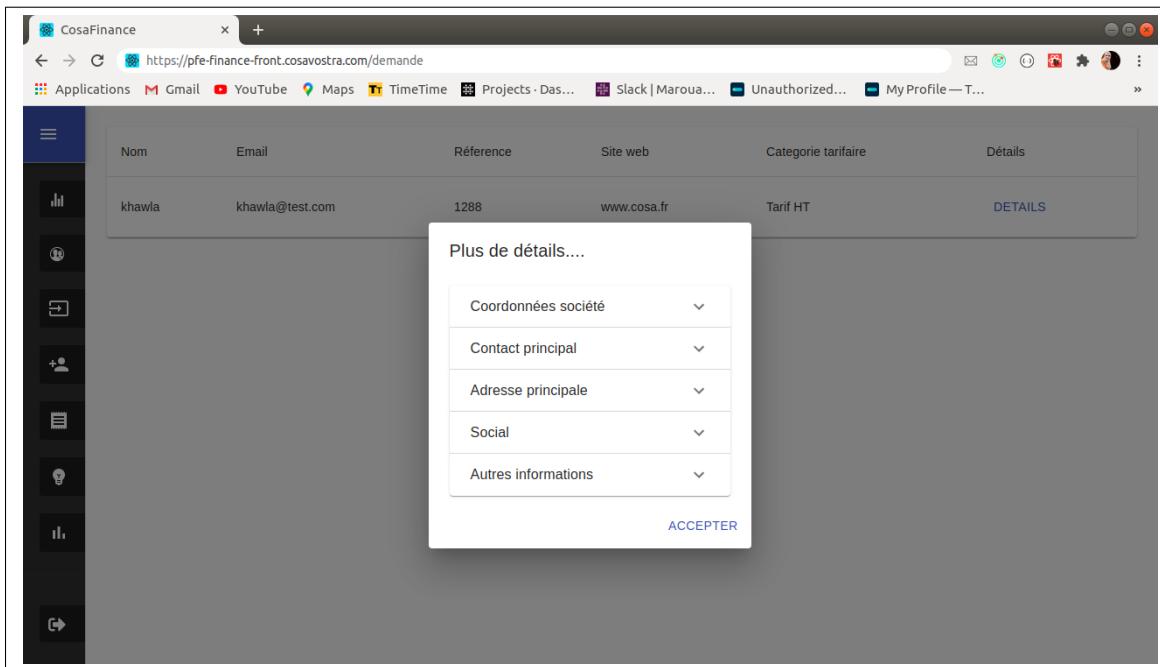


Figure 4.11: Accepter contact client

Après avoir accepter la demande du client il sera ajouter à l'interface contacts client.

The screenshot shows a web browser window for 'CosaFinance' at the URL <https://pfe-finance-front.cosavosta.com/contact>. The page title is 'Contacts de facturation'. On the left is a vertical sidebar with icons for navigation. The main content area displays a table with one row of data:

Actions	Entreprise	Prénom	Nom	Email	Attribué	Sellsy
	cosaFinance	chabchoub	khawla	khawla@test.com	NON ATTRIBUÉ	AJOUTER

Figure 4.12: Accepter contact client

4.5.3 Attribué contact

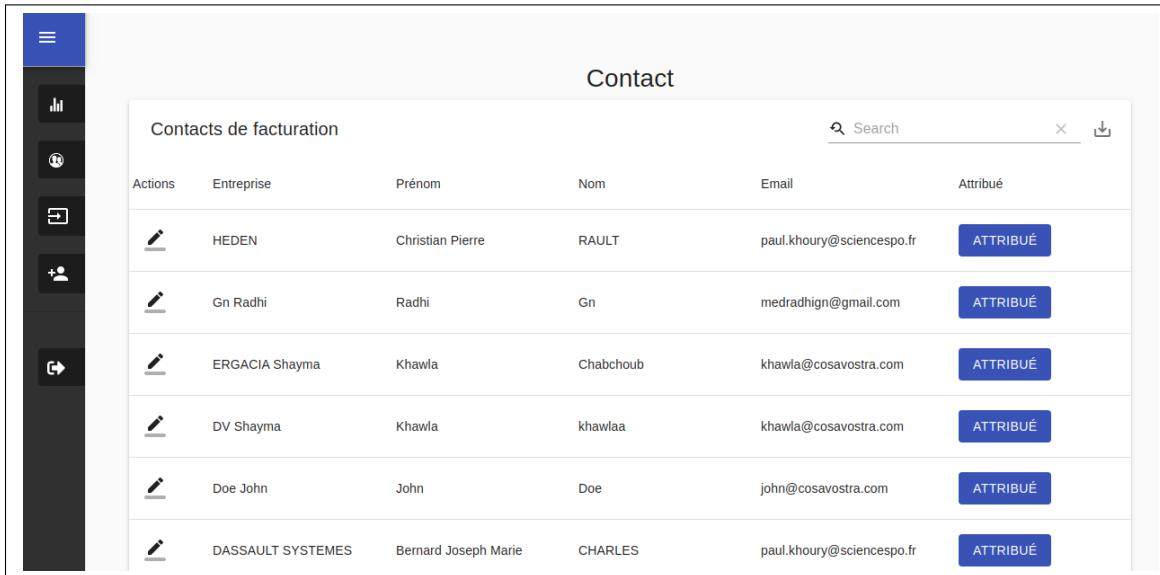
La figure ci-dessous permet a l'administrateur d'attribué une contacts à une entreprise.

The screenshot shows a modal dialog titled 'Attribuer le contact à une fiche client'. On the left is a vertical sidebar with icons. The main area has a label 'Client' above a dropdown menu containing the option 'Willo32'. At the bottom are two buttons: 'Sauvgarder' (Save) and 'Annuler' (Cancel).

Figure 4.13: Attribution des contacts du facturation

4.5.4 Consulter la liste des contacts des facturations

En tant qu'administrateur, il permet de consulter la liste des contacts de facturation avec la bouton attribué.

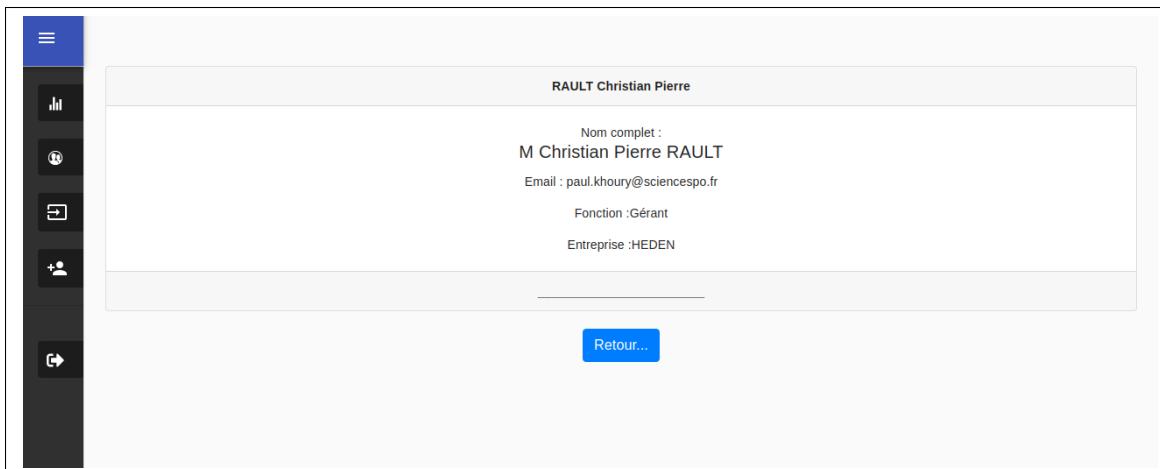


Contacts de facturation					
Actions	Entreprise	Prénom	Nom	Email	Attribué
	HEDEN	Christian Pierre	RAULT	paul.khoury@sciencespo.fr	ATTRIBUÉ
	Gn Radhi	Radhi	Gn	medradhign@gmail.com	ATTRIBUÉ
	ERGACIA Shayma	Khawla	Chabchoub	khawla@cosavosta.com	ATTRIBUÉ
	DV Shayma	Khawla	khawlaa	khawla@cosavosta.com	ATTRIBUÉ
	Doe John	John	Doe	john@cosavosta.com	ATTRIBUÉ
	DASSAULT SYSTEMES	Bernard Joseph Marie	CHARLES	paul.khoury@sciencespo.fr	ATTRIBUÉ

Figure 4.14: Contacts facturation

— Détails du contacts de facturation

Si l'administrateur de notre plateforme demande plus d'information en appuyant sur la button attribué pour afficher plus d'information sur le contacts sélectionner.



RAULT Christian Pierre

Nom complet :
M Christian Pierre RAULT
Email : paul.khoury@sciencespo.fr
Fonction : Gérant
Entreprise : HEDEN

Retour...

Figure 4.15: Détails facturation

4.5.5 Modifier et/ou exporter des contacts

Cette interface permet à l'administrateur de consulter la liste de tous les contacts de modifier et l'exporter sous la forme csv.

Actions	Entreprise	Prénom	Nom	Email	Attribué	Sellsy
	cosaFinance	chabchoub	khawla	khawla@test.com	NON ATTRIBUÉ	AJOUTER

Figure 4.16: Liste des devis

Conclusion

Ce chapitre a été dédié à la présentation du sprint 2. Nous avons établi en premier temps le backlog produit de ce sprint. Ensuite nous avons défini les différentes fonctionnalités réalisées à travers un diagramme de cas d'utilisation globale avec quelques descriptions textuelles. Puis, nous avons passé à la partie conception en présentant le diagramme de classe de conception du module mise à jour, nous avons aussi élaboré un diagramme de séquence objet d'un cas d'utilisation. Finalement, nous avons présenté les interfaces dans la partie réalisation qui montrent notre travail durant ce sprint. Dans le chapitre qui suit, nous allons produire le dernier sprint couvrant les fonctionnalités suivantes : Gestion des objectifs, gestion des compte client «factures» et statistique.

SPRINT 3

Plan

1	Objectif du Sprint	60
2	Analyse	60
3	Analyse spécifique	61
4	Conception	68
5	Réalisation et test	69

Introduction

Dans ce chapitre nous allons passer à la présentation du sprint 3. Nous allons établir en premier temps le backlog produit de ce sprint. Ensuite nous allons définir les différentes fonctionnalités réalisées à travers un diagramme de cas d'utilisation globale avec quelques descriptions textuelles. Puis, nous allons passer à la partie conception en présentant le diagramme de classe de conception , nous allons aussi élaborer un diagramme de séquence objet d'un cas d'utilisation. Finalement, nous allons présenter les interfaces dans la partie réalisation qui montrent notre travail durant ce sprint.

5.1 Objectif du Sprint

Ce chapitre est consacré pour présenter le module gestion des statistiques , gestion des objectifs et la gestion des factures. A la fin de ce livrable l'administrateur être capable de :

- Consulter les statistiques.
- Consulter les objectifs.
- Ajouter un objectifs.
- Consulter des comptes clients «Factures».
- Modifier des comptes clients «Factures».
- Exporter des comptes clients «Factures».

5.2 Analyse

Dans cette section nous allons présenter le backlog du dernière sprint.

5.2.1 Backlog du sprint 3

Nous allons présenter dans le tableau suivant le backlog du sprint 3 traduisant les différentes fonctionnalités achevées de notre application durant ce sprint.

Tableau 5.1: Backlog du sprint 3

ID	User Stories	Estimation (en jour)
13	En tant qu'admin je veux ajouter des objectifs.	4
14	En tant qu'admin je veux afficher les objectifs.	3
15	En tant qu'admin je veux suivre des comptes clients.	3
16	En tant qu'admin je veux modifier des comptes clients.	2
17	En tant qu'admin je veux exporter comptes clients en format csv.	2
18	En tant qu'admin je veux afficher les statistiques.	3
19	En tant qu'admin je veux recevoir des mails.	2

5.3 Analyse spécifique

Cette section est consacrée au raffinement des cas d'utilisation de ce sprint en se basant sur les diagrammes des cas d'utilisation raffinés, une description textuelle des cas d'utilisation et leurs diagrammes de séquences système.

5.3.1 Diagramme de cas d'utilisation

La figure ci-dessous présente le cas d'utilisation globale qui illustre l'ensemble de fonctionnalités à faire tout au long de ce sprint.

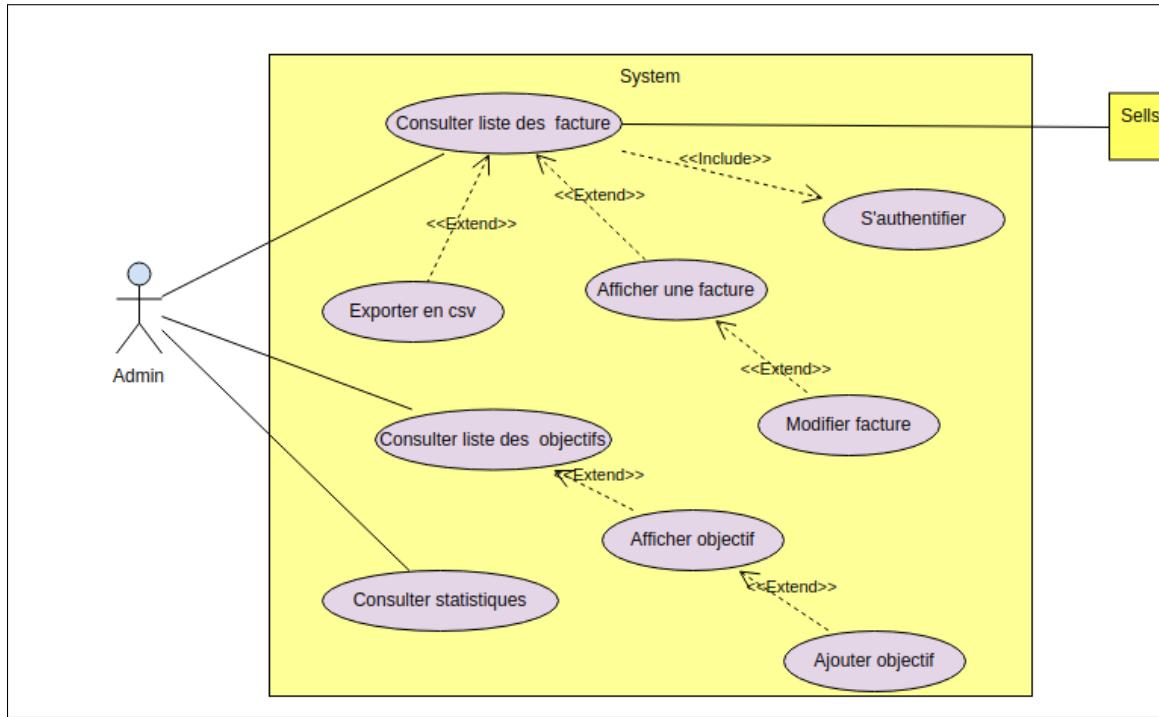


Figure 5.1: Raffinement du cas d'utilisation : Sprint3

5.3.2 Raffinement du cas d'utilisation «Consulter liste des objectifs»

La figure 5.2 représente le raffinement du cas d'utilisation «Consulter liste des objectifs».

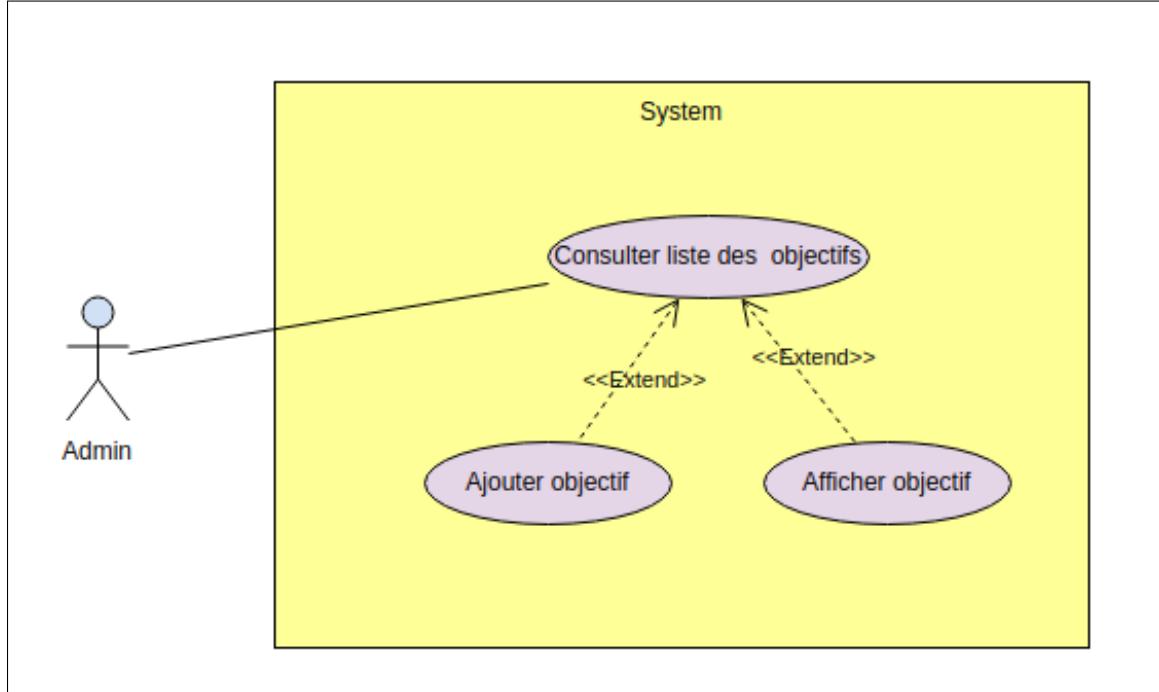


Figure 5.2: Raffinement du cas d'utilisation «Consulter liste des objectifs»

Dans cette partie, nous allons présenter une description textuelle des cas «Consulter des objectifs».

Tableau 5.2: Description textuelle du cas «Consulter des objectifs»

Acteur	Admin
Objectif	Consulter des objectifs
Pré-condition	L'admin doit être authentifié.
Scénario nominal	-l'administrateur demande la consultation des objectifs. -le système consulter la liste des objectifs.
Post-condition	La liste des objectifs consultée
Extension	Admin peut ajouter un objectif. Admin peut consulter un objectif.

Le tableau ci-dessous montre la description textuelle des cas «Ajouter un objectifs».

Tableau 5.3: Description textuelle du cas «Ajouter objectifs»

Acteur	Admin
Objectif	Ajouter des objectifs
Pré-condition	L'administrateur doit être authentifié.
Scénario nominal	-l'administrateur demande la consultation des objectifs. -le système affiche la liste des objectifs. -l'admin appuye sur le bouton «+» pour ajouter des objectifs. -L'admin choisie une année et un mois et un objectifs et le valider. -Le système enregistre l'opération.
Post-condition	Un objectifs à ajouté.

5.3.3 Raffinement du cas d'utilisation «Consulter liste des factures»

la figure 5.3 montre le raffinement du cas d'utilisation «Consulter des facture».

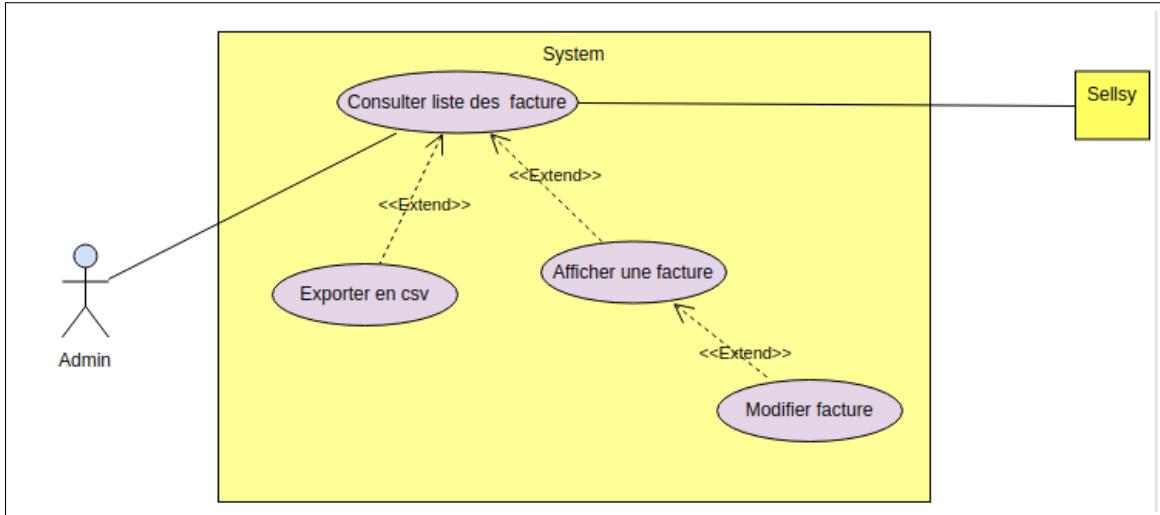


Figure 5.3: Raffinement du cas d'utilisation «Consulter des facture»

Dans cette partie, nous allons présenter une description textuelle des cas «Consulter des facture».

Tableau 5.4: Description textuelle du cas «Consulter les factures»

Acteur	Admin
Objectif	Afficher comptes clients
Pré-condition	L'administrateur doit être authentifié.
Scénario nominal	-l'administrateur demande la consultation la page du comptes clients «factures» . -le système affiche la liste des comptes clients «factures».
Post-condition	Liste des comptes clients «factures» consultée.
Extension	L'admin peut exporter la liste des factures en csv. L'admin peut afficher une facture.

Tableau 5.5: Description textuelle du cas «Exporter les factures»

Acteur	Admin
Objectif	Exporter la liste des factures.
Pré-condition	L'admin doit être s'authentifié. La liste des factures consulté.

Scénario nominal	-L'admin demande d'exporter la liste des factures. -Le système télécharge le fichier.
Post-condition	La liste des factures sont exportées.

Le tableau 5.6 représente la description textuelle des cas «Modifier factures».

Tableau 5.6: Description textuelle du cas «Modifier des comptes clients»

Acteur	Admin
Objectif	Modifier comptes clients
Pré-condition	L'admin doit être authentifié.
Scénario nominal	-l'admin demande la consultation des comptes clients. -le système affiche la liste des comptes clients. -l'admin modifie les champs à modifier. l'admin appuie sur confirmer. -Le système enregistre l'opération. - le système affiche la page de «Suivie comptes clients» avec les nouvelles coordonnées.
Post-condition	Compte clients (facture) modifié

5.3.4 Raffinement du cas d'utilisation «Consulter les statistiques»

La figure 5.4 représente le raffinement du cas d'utilisation «Consulter les statistiques».

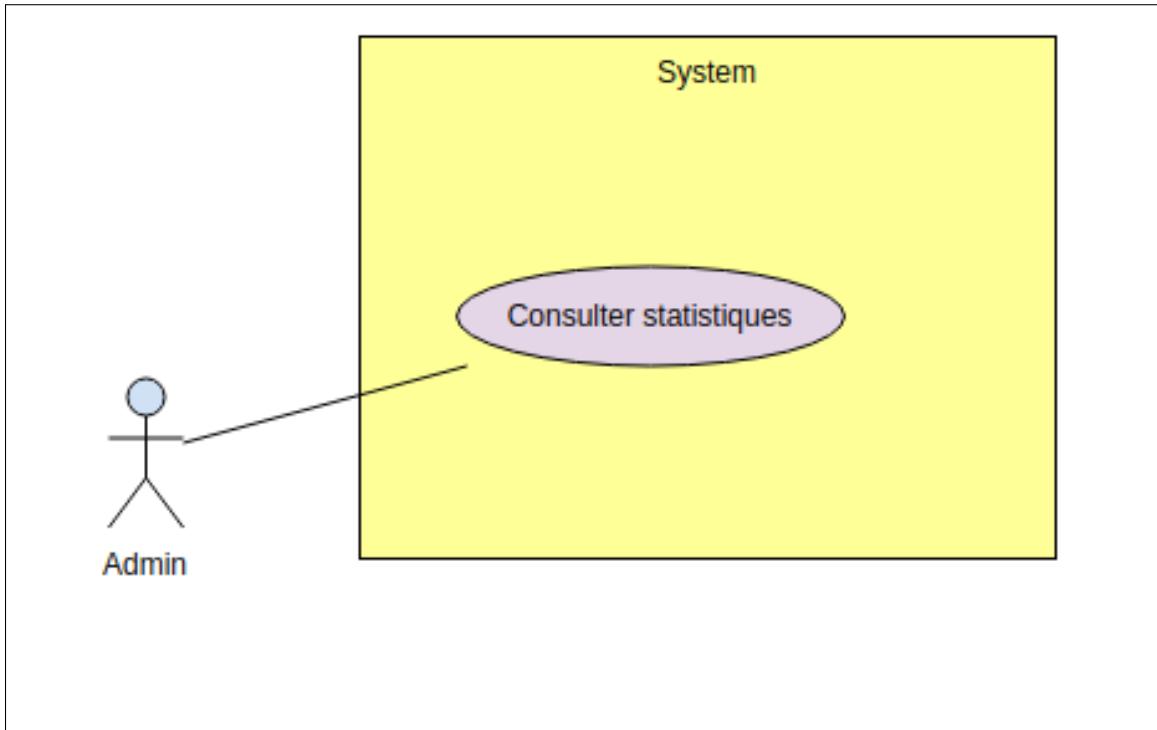


Figure 5.4: Raffinement du cas d'utilisation «Consulter les statistique»

Le tableau ci-dessous représente la description textuelle des cas «Consulter les statistiques».

Tableau 5.7: Description textuelle du cas «Consulter les statistique»

Acteur	Admin
Objectif	Consulter les statistiques
Pré-condition	L'admin doit être authentifié.
Scénario nominal	-l'admin demande la consultation des statistiques. -le système affiche l'interface des statistiques.
Post-condition	Les statistique est consulté

5.3.5 Diagramme de séquence «Modifier comptes clients»

La figure ci-dessous présente le diagramme de séquence système du cas d'utilisation «Modifier comptes clients»

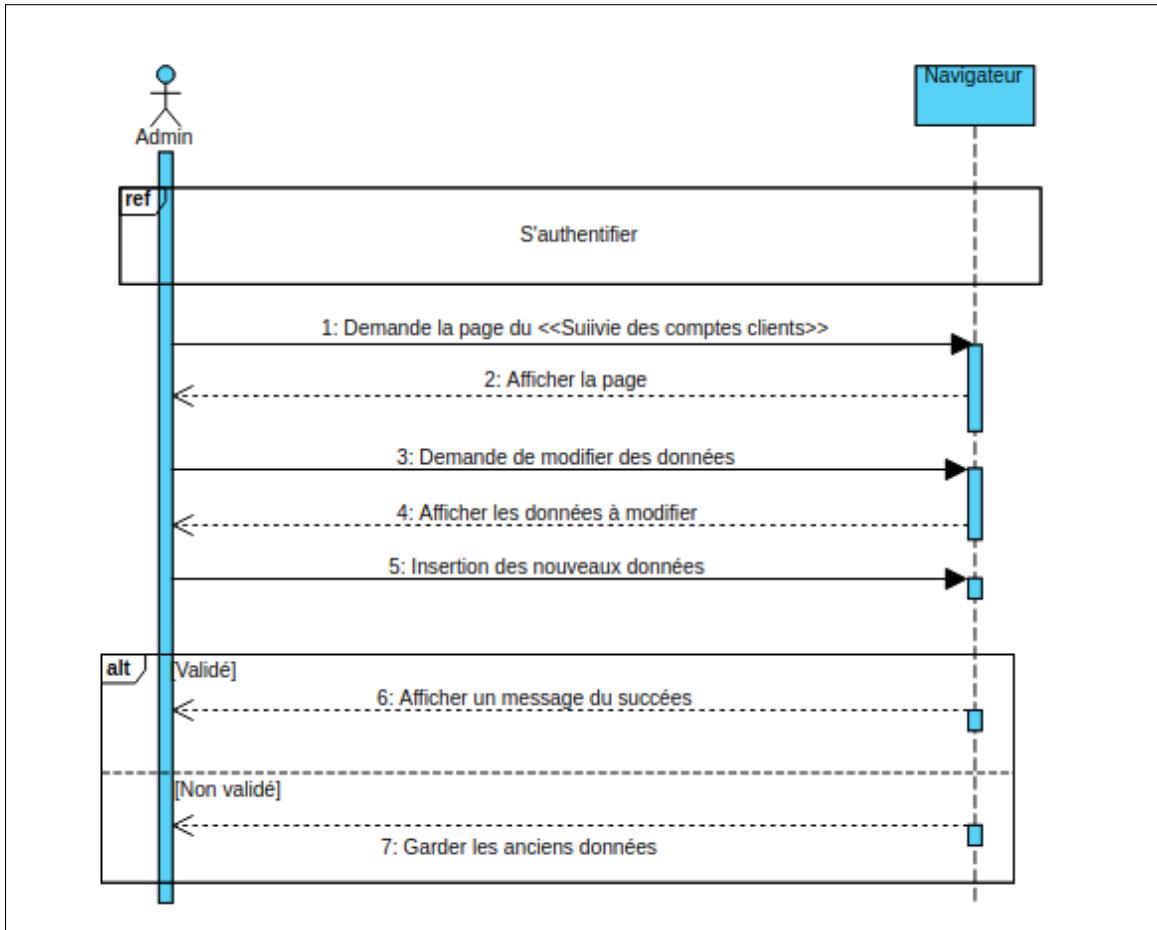


Figure 5.5: Diagramme de séquence système du cas : «Modifier comptes client»

La figure 5.6 présente le diagramme de séquence Objet «Modifier comptes clients».

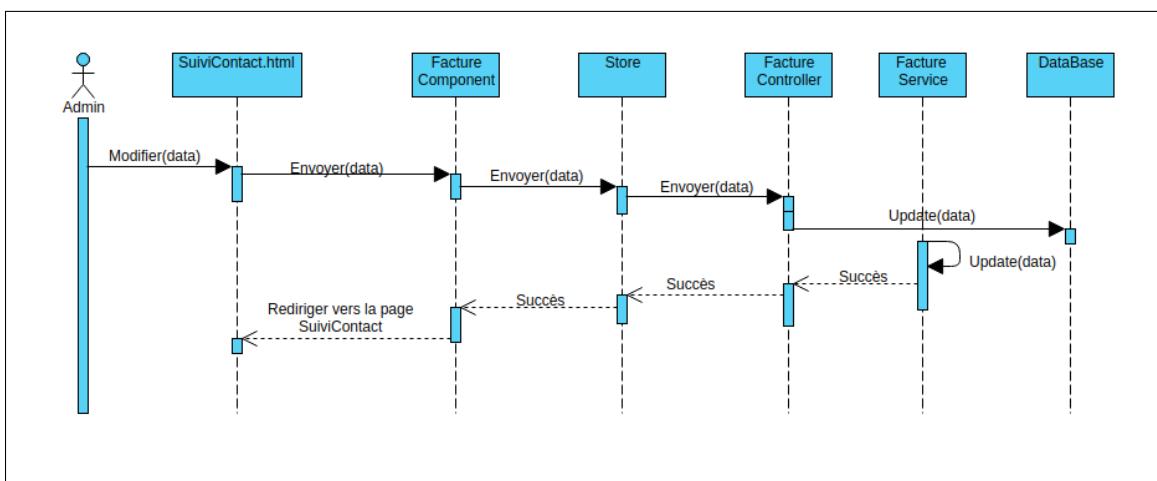


Figure 5.6: Diagramme de séquence Objet «Modifier comptes clients»

5.4 Conception

Cette partie est consacrée à la conception du sprint 3, pour ce faire nous commençons par le diagramme de classe détaillé et par la suite le diagramme de séquence objet.

5.4.1 Diagramme de classe du sprint 3

La figure 5.7 présente le diagramme des classes utilisées durant la réalisation des fonctionnalités du troisième sprint.

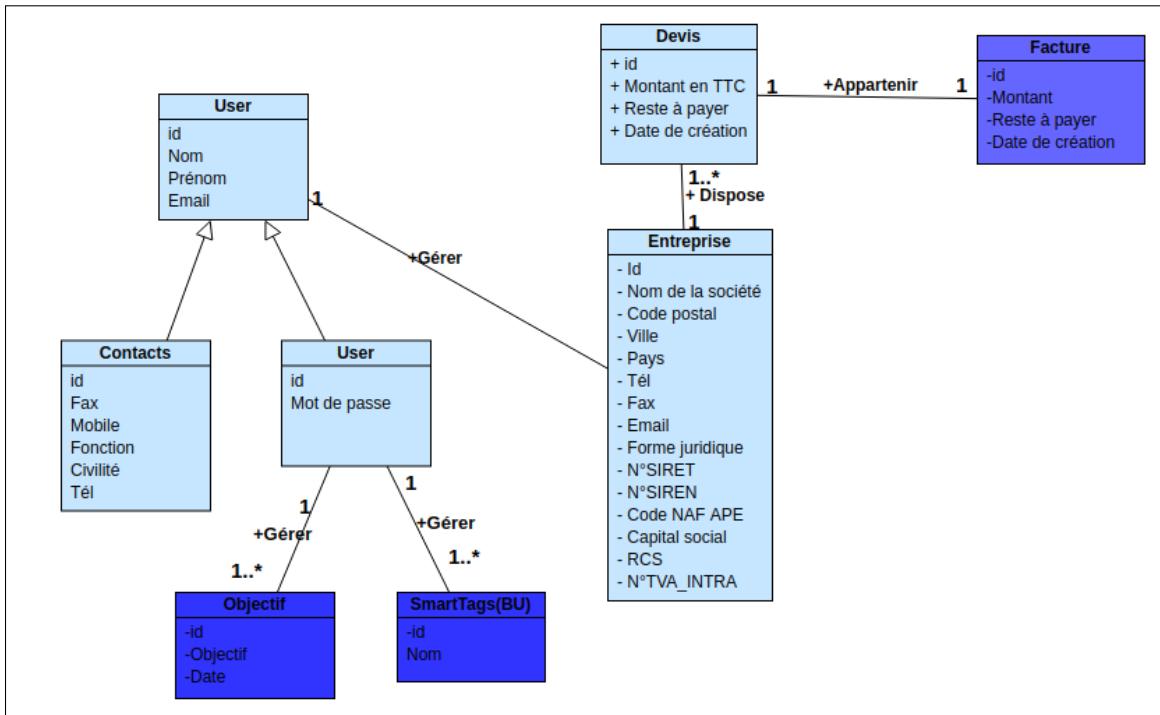


Figure 5.7: Diagramme de classe du sprint 3

5.4.2 Diagramme d'activité du cas d'utilisation «Ajout objectifs»

La figure 5.8 présente le digramme d'activité du cas d'utilisation «Ajout objectifs».

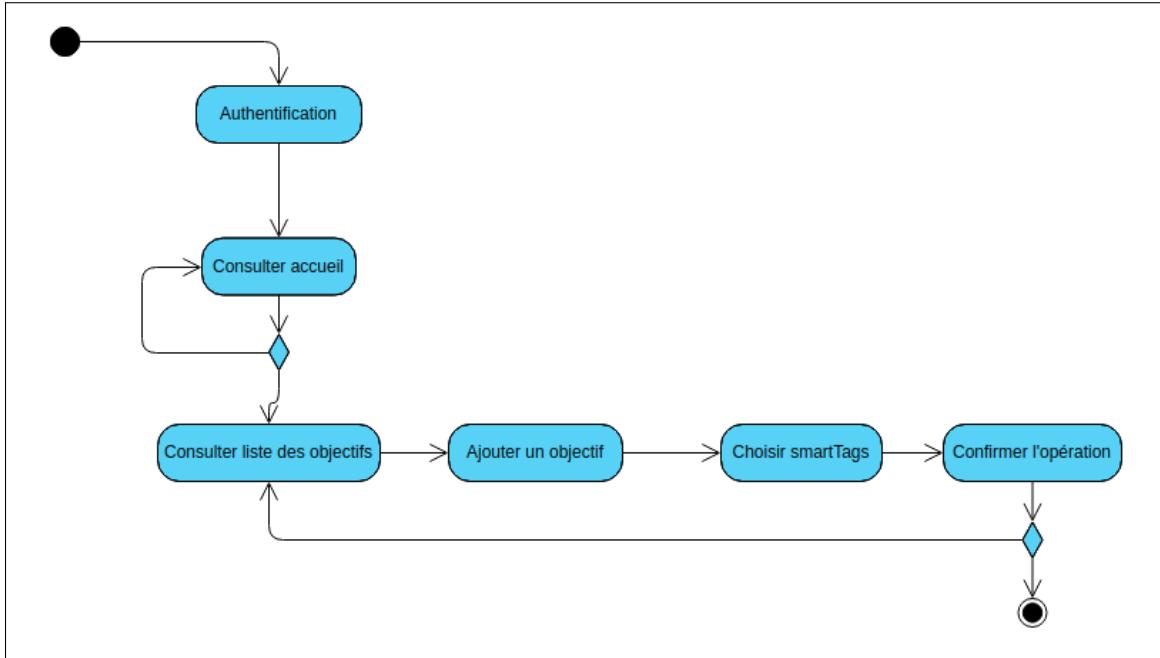


Figure 5.8: Diagramme d'activité du cas d'utilisation «Ajout objectifs»

5.5 Réalisation et test

5.5.1 Gestion des compte client

La figure ci-dessous représente l'interface de consultation des suivies des comptes clients.

L'interface de consultation des comptes clients affiche une liste de clients avec leurs détails financiers et leur état de suivi. Les colonnes sont les suivantes :

Comptes	Total encaissable	Encaissé	0-30jrs	31-60jrs	61-90jrs	+90jours	A échoir 0-30j	commentaire	Actions
10 MEDIAS	0,00 €	6.000,00 €	0,00 €	0,00 €	0,00 €	6.000,00 €	0,00 €		
1706 conseil	0,00 €	17.142,00 €	0,00 €	0,00 €	10.902,00 €	0,00 €	6.240,00 €		
20 Minutes	0,00 €	88.893,38 €	11.533,08 €	27.780,00 €	0,00 €	49.580,30 €	0,00 €		
2DK	0,00 €	489,60 €	489,60 €	0,00 €	0,00 €	0,00 €	0,00 €		
50 Foch	0,00 €	2.261,52 €	2.261,52 €	0,00 €	0,00 €	0,00 €	0,00 €		
Aäsgard Holding	0,00 €	15.048,00 €	5.016,00 €	0,00 €	0,00 €	0,00 €	0,00 €		
ABAC	0,00 €	9.888,00 €	960,00 €	0,00 €	0,00 €	1.440,00 €	0,00 €		
ABBAS Moulay	0,00 €	2.068,20 €	0,00 €	0,00 €	0,00 €	0,00 €	0,00 €		

Figure 5.9: Consultation des comptes clients

La figure ci-dessous représente l'interface de modification des données. En cliquant sur le bouton confirmer, l'admin aura un message du succès et en cliquant sur le bouton valider. Cet interface est exportable sous la forme CSV.

The screenshot shows a web-based application interface for managing client accounts. The main area displays a table with columns representing different financial metrics for various clients. A green success message box is overlaid on the right side of the table, stating "Succès Votre commentaire a bien été mis à jour". The table includes rows for ABAC, ABBAS Moulay, Abilways, ACOR PROPRETE MULTISERVICES, ADN Content, AFBGU, AFDAS, AFFIANCA TE PATRIMOINE, and Agence Axa Thierry Bensimon. Each row has a edit icon (pencil) at the end. On the left, there's a sidebar with several icons. At the bottom, there's a file download button labeled "Suivi des com....csv" and a "Tout afficher" button.

	ABAC	0,00 €	9.888,00 €	960,00 €	0,00 €	0,00 €	1.440,00 €	0,00 €	
ABAC	0,00 €	9.888,00 €	960,00 €	0,00 €	0,00 €	1.440,00 €	0,00 €		
ABBA S Moulay	0,00 €	2.068,20 €	0,00 €	0,00 €	0,00 €	0,00 €	0,00 €		
Abilways	0,00 €	1.098,00 €	0,00 €	0,00 €	1.098,00 €	0,00 €	0,00 €		
ACOR PROPRETE MULTISERVICES	0,00 €	570,00 €	570,00 €	0,00 €	0,00 €	0,00 €	0,00 €		
ADN Content	0,00 €	14.400,00 €	0,00 €	6.000,00 €	0,00 €	0,00 €	0,00 €		
AFBGU	0,00 €	120,00 €	0,00 €	0,00 €	0,00 €	120,00 €	0,00 €		
AFDAS	0,00 €	7.200,00 €	0,00 €	0,00 €	0,00 €	7.200,00 €	0,00 €		
AFFIANCA TE PATRIMOINE	0,00 €	9.414,00 €	0,00 €	0,00 €	0,00 €	0,00 €	0,00 €		
Agence Axa Thierry Bensimon	0,00 €	466,80 €	466,80 €	0,00 €	0,00 €	0,00 €	0,00 €		

Figure 5.10: Modification des comptes clients

5.5.2 Gestion des objectifs

La figure ci-dessous représente l'ensemble des objectifs du smartTags.

Chapitre 5. Sprint 3

Date	Conseil	projet				
	Objectif	Atteint	%de réalisation	Objectif	Atteint	%de réalisation
juin	5000	3000	60%			
juillet				200000	0	0%

Figure 5.11: Consultation des objectifs

L'admin peut ajouter un nouveau objectifs et il doit choisir un smartTags , un objectifs et une date. Puis, l'adhérent clique sur le bouton accepter il aura un message du succès.

Figure 5.12: Ajout des objectifs

5.5.3 Gérer les statistiques

La figure ci-dessous représente l'ensemble des clients en retard de paiement.

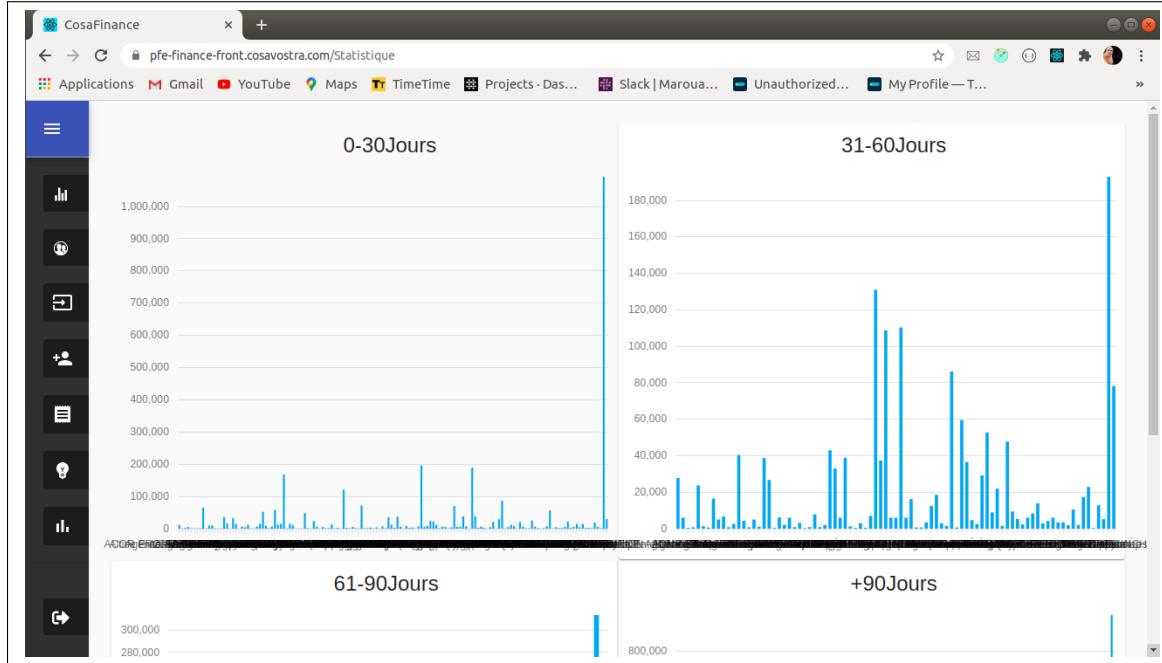


Figure 5.13: Consultation la liste des statistique

Conclusion

A la fin de ce chapitre, nous avons réussi à produire un incrément répondant aux besoins du client et étant utilisé dans un environnement de production. Nous clôturons ce rapport par une conclusion général.

Conclusion générale

Le développement de notre projet intitulé « CosaFinance » nous a permis de répondre aux besoins de notre client la société CoSaVostra.

Notre application web consiste à gérer les devis, les factures, les contacts des différents clients. La mise en place de cette application nous a permis de mettre en œuvre nos connaissances théoriques acquises tout au long de notre formation à 'Ecole Supérieure Privée d'Ingénierie et de Technologie « ESPRIT ».

Ce travail nous a fourni également un grand apport au niveau de plusieurs niveaux. Sur le plan technique, nous avons appris à manipuler la librairie REACT, les services développés par Symfony 5 et pleins d'autres outils. Nous avons également eu l'opportunité de maîtriser la conception en utilisant UML. Le stage quotidien au sein de la société a aussi été pour nous une occasion unique qui nous a apporté beaucoup d'enrichissement sur le plan relationnel. Bien que les principaux objectifs de notre projet aient été atteints, le système que nous avons développé pourrait être enrichi par d'autres fonctionnalités avancées et améliorations selon des nouveaux besoins demandés par notre client. Aussi, Vu le grand volume de données de la direction, une extension de ce travail pourra avoir lieu via l'utilisation de l'informatique décisionnelle qui englobe les solutions IT et qui leur apporte une nouvelle vision contenant des rapports et des tableaux de bord de suivi des activités de l'entreprise à la fois analytiques et prospectifs.

Netographie

- [1] F. LOTHON. (Novembre 2019). The Scrum Guide. [consulté le 13-Mars-2020], Organisme, adresse : <https://agiliste.fr/guide-de-demarrage-scrum/>.
- [2] L. AUDIBERT. (Janvier 2020). UML. [consulté le 15-Mars-2020], Organisme, adresse : <https://laurent-audibert.developpez.com/Cours-UML/?page=diagramme-cas-utilisation>.
- [3] (October 2019). VSCode. [consulté le 17-Mars-2020], Organisme, adresse : <https://visualstudio.microsoft.com/fr/students/>.
- [4] ——, (Mars 2020). VSCode. [consulté le 20-Mars-2020], Organisme, adresse : <https://www.c-sharpcorner.com/article/what-and-why-reactjs/>.
- [5] (Avril 2020). JavaScript. [consulté le 11-Mars-2020], adresse : <https://developer.mozilla.org/fr/docs/Web/JavaScript>.
- [6] (Avril 2020). JavaScript. [consulté le 11-Mars-2020], adresse : <https://www.cours-gratuit.com/cours-html/cours-html5-et-css3-en-pdf>.
- [7] (Janvier 2020). PhpStorm. [consulté le 22-Mars-2020], adresse : <https://www.gladir.com/SOFTWARE/PHPSTORM/presentation.htm>.
- [8] (Juillet 2017). PHP. [consulté le 20-Mars-2020], adresse : <https://www.php.net/>.
- [9] SYLVAIN. (Octobre 2017). Postman. [consulté le 15-Mars-2020], Organisme, adresse : <https://blog.webnet.fr/presentation-de-postman-outil-multifonction-pour-api-web/>.
- [10] (Janvier 2020). Git. [consulté le 15-Avril-2020], adresse : <https://git-scm.com/>.
- [11] (Janvier 2020). Visual Paradigm. [consulté le 20-Avril-2020], adresse : <https://www.visual-paradigm.com/>.
- [12] (Janvier 2020). Sentry. [consulté le 25-Avril-2020], adresse : <https://sentry.io/welcome/>.
- [13] (Avril 2020). Slack. [consulté le 30-Avril-2020], adresse : <https://slack.com/intl/fr-fr/>.
- [14] (Avril 2020). Trello. [consulté le 20-Avril-2020], adresse : <https://trello.com/fr>.

Résumé

Ce travail fait l'objet de sujet de fin d'études au sein de l'École Supérieure Privée d'Ingénierie et de Technologies. Le travail a été effectué durant un stage de quatre mois à l'agence CosaVostra.

Notre mission consistait à concevoir et réaliser une application de direction et reporting financier pour gérer les devis et les factures de CosaVostra ainsi que les contacts de facturations. L'application «CosaFinance» est achevée en version web.

Pour la réalisation de l'application web, nous avons opté pour le Framework Symfony, la bibliothèque ReactJS et pour la base de données nous avons opté pour MySql.

Mots clés : Symfony, ReactJS, plugin, MySql



ECOLE SUPÉRIEURE PRIVÉE D'INGÉNIERIE ET DE TECHNOLOGIES

www.esprit.tn - E-mail : contact@esprit.tn

Siège Social : 18 rue de l'Usine - Charguia II - 2035 - Tél. : +216 71 941 541 - Fax. : +216 71 941 889

Annexe : Z.I. Chotrana II - B.P. 160 - 2083 - Pôle Technologique - El Ghazala - Tél. : +216 70 685 685 - Fax. : +216 70 685 454