



Institut National des Sciences Appliquées et de Technologie

UNIVERSITÉ DE CARTHAGE

Projet de Fin d'Études

Filière : GL ou RT

Titre du projet

Présenté par

Flen FOULENI

Encadrant INSAT : **Mr FOULENI Flen**
Encadrant ENTREPRISE : **Mme FALTEN Flena**

Présenté le : --/--/2015

JURY

M. President FLEN (Président)
Mme. Rapporteur FLENA (Rapporteur)

Année Universitaire : 2014/2015

Dédicace

A mes chers parents Naoufel et Hajer qui m'ont donné l'amour et la tendresse dont j'avais toujours besoins. Pour leur confiance et leur soutien pour tous les choix de ma vie. Je ne pourrai jamais exprimer la reconnaissance dont je vous apporte. Que dieu vous bénisse et vous procure une longue vie pleine de joie.

A mon cher frère Malek que dieu le protège.

Mes chers cousins et chères cousines et tous membres de ma famille qui ne cessaient de m'épauler à chaque instant.

A mes chers amis, mes chers camarades de l'INSAT et surtout mes amours PEOPLE OF THE BOX qui ont converti ma vie universitaire en un rêve dont je ne souhaiterais jamais la fin.

Remerciements

Au terme de ce travail je tiens à remercier toutes personnes, qui par leur conseil, par leur encouragement ou même par leur présence de près ou de loin, ont contribué à la réalisation de ce travail.

Je tiens aussi bien à remercier la société Urbaproduct qui m'a accueilli, et son directeur général Aymen ELLOUZE pour sa confiance et l'honneur qu'il m'a donné en travaillant au sein de son équipe.

Mes remerciements s'adresse à mon encadrant Ons BEN CHEIKH pour son accueil chaleureux et tous les collaborateurs d'Urbaproduct ainsi que mes camarades de stage Hamza BEINI et Mariem Nfaiedh qui m'ont rendu ce stage assez spécial.

Mes profondes gratitudes s'adresse à Monsieur Aymen SELLAOUTI qui était plus qu'un superviseur pendant cette période, il était un grand support avec son assistance continue, avec ses conseils qui me remontaient à chaque fois le morale et surtout pour le temps précieux qu'il m'a consacré.

Enfin je remercie tous les enseignants qui ont assuré ma formation, et qui sont présents aujourd'hui comme jury afin d'évaluer mon travail.

Table des Matières

Liste des Figures	v
Liste des Tableaux	vii
Introduction Générale	1
I Cadre du projet	2
1 Présentation de l'organisme d'accueil	2
1.1 L'entreprise "UrbaProd"	2
1.2 Domaine d'expertise	3
2 Problématique	4
3 Solutions existantes	4
3.1 Version éditeur	5
3.2 Version libre	6
3.3 Etude comparative	6
4 Objectifs du projet	7
5 Contexte méthodologique du projet	8
5.1 Le choix de la Méthode Scrum	8
5.2 Les rôles et les notions	9
II Capture, analyse et spécification des besoins	10
1 Étude de l'existant	10
2 Critique de l'existant	11
3 Analyse des besoins	11
3.1 Les acteurs	11
3.2 Les besoins fonctionnels	12
3.3 Les besoins non fonctionnels	12
3.4 Planification du projet	13
3.4.1 Les acteurs du projet	13
3.4.2 Backlog produit	13
3.4.3 Les sprints du projet	13
3.5 Spécification des besoins : diagramme de cas d'utilisation global	14

III Etude conceptuelle	18
1 Architecture physique	18
2 Architecture logicielle	19
3 Raffinement des diagrammes des cas d'utilisations	21
3.1 module gestion des demandes	21
3.2 module gestion des utilisateurs	22
3.3 Module gestion des paramètres de la demande	23
3.4 Module exportation des données	24
4 Diagrammes structurels	25
4.1 Diagramme de paquetages	25
4.2 Diagramme de classes	28
5 Diagrammes comportementaux	29
5.1 Diagrammes de séquences	29
5.2 Diagramme d'activité	30
IV Réalisation	39
1 Environnements de travail et choix techniques	39
1.1 Environnements de travail	39
1.2 Technologies	40
2 Présentation de l'application	44
Conclusion Générale et Perspectives	53
Bibliographie	54
Annexe : Remarques Diverses	56

Liste des Figures

I.1	Organigramme de l'entreprise	3
I.2	Références de l'entreprise	4
I.3	Fonctionnalités offertes par Everwin CXM [1]	5
I.4	Architecture Dolibarr [2]	6
I.5	Ecran d'accueil de Dolibarr [2]	7
II.1	Diagramme de cas d'utilisation global	14
III.1	Architecture physique	19
III.2	Architecture applicative de la solution	20
III.3	Diagramme de cas d'utilisation Gestion d'une demande	22
III.4	Diagramme de cas d'utilisation gestion utilisateurs	25
III.5	Diagramme de cas d'utilisation du paramétrage d'une demande	31
III.6	Diagramme de cas d'utilisation exportation des données	32
III.7	Diagramme de paquetage	33
III.8	Le modèle du domaine	34
III.9	Diagramme de classe des contrôleurs	35
III.10	Diagramme de séquence ajout d'une demande	36
III.11	Diagramme de séquence suppression d'un site	37
III.12	Diagramme d'activité envoi d'une notification	38
IV.1	Commits pendant le mois de Juillet	41
IV.2	Toutes les contributions	41
IV.3	Disposition des Frameworks	44
IV.4	Interface d'authentification	45
IV.5	Différence entre deux rôles	46
IV.6	Vue complète de l'interface d'accueil	47
IV.7	Changer l'état d'une demande	48
IV.8	Le graphe des demandes	48
IV.9	Formulaire de la demande (vue administrateur)	49
IV.10	Pop-up affectation d'un site	49
IV.11	Demande émise	50
IV.12	Demande après traitement	51

IV.13 Interface écran commun	51
IV.14 Interface administration des utilisateurs	52
IV.15 Interface administration des demandes	52
A.1 Demande sous sa forme originale	56

Liste des Tableaux

I.1	Comparatif entre les solutions existantes	7
II.1	Les acteurs du SCRUM	13
II.2	Backlog du produit	16
II.3	Les sprints du projet	17
III.1	Ajout d'une demande	23
III.2	Ajouter un nouveau commentaire	24
III.3	Ajout d'un utilisateur	26
III.4	Ajout d'un nouveau site	27
III.5	Exporter la liste des demandes	28
III.6	Descriptif des entités de l'application	29

Introduction Générale

Aujourd'hui, le monde de l'architecture urbaine est entrain d'évoluer d'une façon terrible vue la croissance de la population, la baisse des surfaces habitables et la grande révolution technologique. Chaque bureau essaie de créer de nouvelles procédures de travail pour se démarquer des autres concurrents et améliorer sa productivité. Parce qu'une boîte qui n'évolue pas, finira certainement par disparaître tôt ou tard.

De ce fait, les bureaux d'architecture favorise l'automatisation de leur système de travail. Alors, ils nécessitent des solutions modernes et bien étudiées afin de bénéficier le plus rapidement possible de ses avantages parce que maintenant, le temps est la chose la plus précieuse, autant en avoir des résultats immédiats.

Dans ce contexte, l'entreprise "Urbaprod" pense s'aligner à cette vague technologique en mettant en place son propre système de gestion de demande client. Cette solution permettra la centralisation des informations de l'entreprise ainsi que la facilité d'organisation et d'accès aux données.

Le présent rapport est structuré en quatre chapitres brièvement décrits :

- Cadre du projet : ce chapitre est consacré à l'introduction du cadre général du projet ainsi qu'une petite étude comparative des solutions existantes afin de se mettre dans le tas.
- Analyse et spécification des besoins : cette section représente le vrai point d'entrée de notre projet, elle porte sur la spécification des besoins et la planification de notre travail.
- Étude conceptuelle : dans cette partie, nous proposons l'architecture de notre application et la modélisation conceptuelle de la solution proposé à travers des diagrammes de "Unified Modeling Language" (UML) [3].
- Réalisation : c'est la dernière section de notre rapport, et elle présente notre contribution.

Nous clôturons ce rapport par une conclusion, dans laquelle nous évaluerons les résultats atteints et nous exposerons les perspectives éventuelles du présent projet.

Chapitre I

Cadre du projet

Plan

1	Présentation de l'organisme d'accueil	2
1.1	L'entreprise "UrbaProd"	2
1.2	Domaine d'expertise	3
2	Problématique	4
3	Solutions existantes	4
3.1	Version éditeur	5
3.2	Version libre	6
3.3	Etude comparative	6
4	Objectifs du projet	7
5	Contexte méthodologique du projet	8
5.1	Le choix de la Méthode Scrum	8
5.2	Les rôles et les notions	9

Introduction

1 Présentation de l'organisme d'accueil

1.1 L'entreprise "UrbaProd"

UrbaProd est une société de conseil en organisation par l'espace spécialisée dans le secteur d'aménagement des espaces de travail. Filiale de la société mère Génie des Lieux, depuis 2010, UrbaProd participe à l'aménagement et le réaménagement des espaces de travail pour des grands comptes répartis essentiellement en France. "UrbaProd" est composée essentiellement de deux pôles : le pôle "3D" et le pôle "space planning". Nous illustrons dans la figure [I.1](#) l'organigramme

I.1 Présentation de l'organisme d'accueil

de "UrbaProd". Notre projet a été effectué au sein du pôle "IT" qui est nouvellement créé afin de mettre en place une solution de gestion de production Offshore.

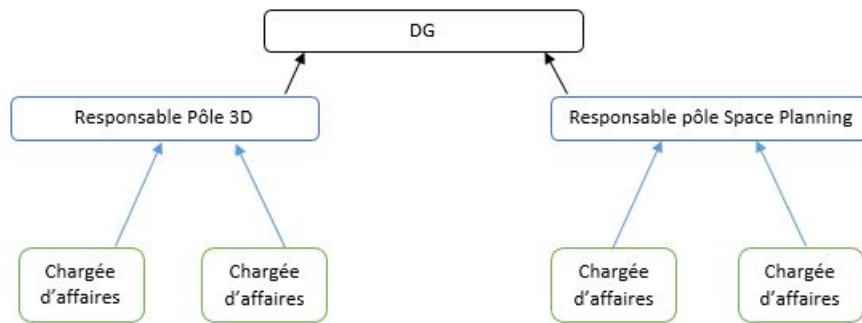


Figure I.1 – Organigramme de l'entreprise

1.2 Domaine d'expertise

Elle est spécialisée dans la définition, la conception et l'appropriation des espaces. "UrbaProd" est présente tout au long du cycle de vie du processus d'optimisation et de conception des lieux afin d'améliorer la disposition des collaborateurs :

- En amont pour le recueil des besoins et le cadrage stratégique.
- Puis pour la conception et la réalisation des plans des espaces.
- Enfin pour la gestion et le pilotage de mobilité et l'ingénierie de transfert.

Les missions de "UrbaProd" incluent :

- Audit et programmation - Concept et charte d'aménagement - Etudes et space planning.
- Prospective - Management de projet - Conduite du changement.
- Architecture d'intérieur - Prescription mobilier - Conduite de travaux.
- Pilotage et gestion de la mobilité - Ingénierie de transfert.
- Solutions numériques de gestion et visualisation des espaces.

"UrbaProd" travaille et accompagne les processus d'aménagements des grands comptes en France et en Europe, à savoir : EDF, Thales, Cartier, Danone, INPI, l'Oréal, Technicolor, Hachette Livre dont nous retrouvons les logos dans la figure I.2.

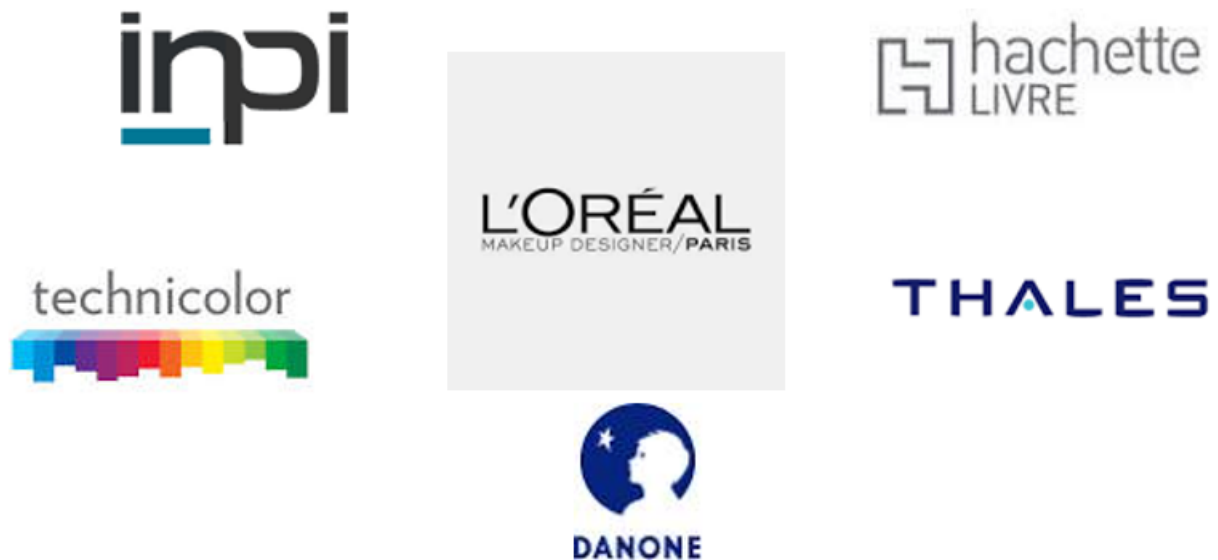


Figure I.2 – Références de l'entreprise

2 Problématique

Aujourd'hui, nous avons de plus en plus de demandes à traiter, sans avoir un support informatique pour les gérer. Ce n'est pas très aisé de gérer l'historique de la clientèle d'une entreprise. De plus, lors des différentes interactions avec la clientèle, et en particulier lors du partage d'informations, l'outil utilisé est le mail. Cependant, cela vire au désordre, i.e. messages disparus, méthodes de classements qui diffèrent d'un collaborateur à un autre, absence de suivi. Et comme la relation entre collaborateurs est la priorité stratégique de la société, ce point est à travailler d'urgence.

3 Solutions existantes

Il existe plusieurs solutions de gestion de relation client sur le marché. En examinant ces applications, nous citons les plus importantes :

3.1 Version éditeur

- **CRM de Fitnet Manager** : Fitnet Avant-vente est l'outil de gestion commerciale de Fitnet Manager. Les solutions CRM et ERP existent cotes à cotes. Activées ensemble, elles fonctionnent d'une manière entièrement intégrée. Fitnet Manager couvre ainsi l'ensemble du cycle de vie de l'activité : depuis la prospection jusqu'à la facturation et l'analyse dans les reporting [4].
- **Everwin CXM** : c'est une solution qui vise les cabinets d'architectures, elle couvre plusieurs fonctionnalités citant à titre d'exemple :
 - Base de données de la société et des contacts avec gestion automatisée de la téléprospection.
 - Gestion des actions commerciales et des campagnes marketing.
 - Agenda partagé et planning général des collaborateurs.

Cette solution est fondée sur une technologie Client/Serveur sous windows avec une base de données SQL Server. il est disponible en mode SaaS ou licence et peut être couplé aux ERP ERP Everwin SX et Everwin GX [1] . La figure I.3 illustre l'ensemble des fonctionnalités couvertes par Everwin CXM.



Figure I.3 – Fonctionnalités offertes par Everwin CXM [1]

3.2 Version libre

Dolibar ERP/CRM : cette solution gère les activités professionnelles ou associatives de point de vue contact, commandes, stock. Elle gère aussi la gestion des projets et les avancements de leurs tâches, et assure même la gestion de la ressource humaine. Dolibarr est disponible pour toute plate-forme puisqu'elle est développée en PHP, MySQL ou encore PostgreSQL . La figure I.4 illustre l'architecture sur laquelle est basée la solution Dolibarr :

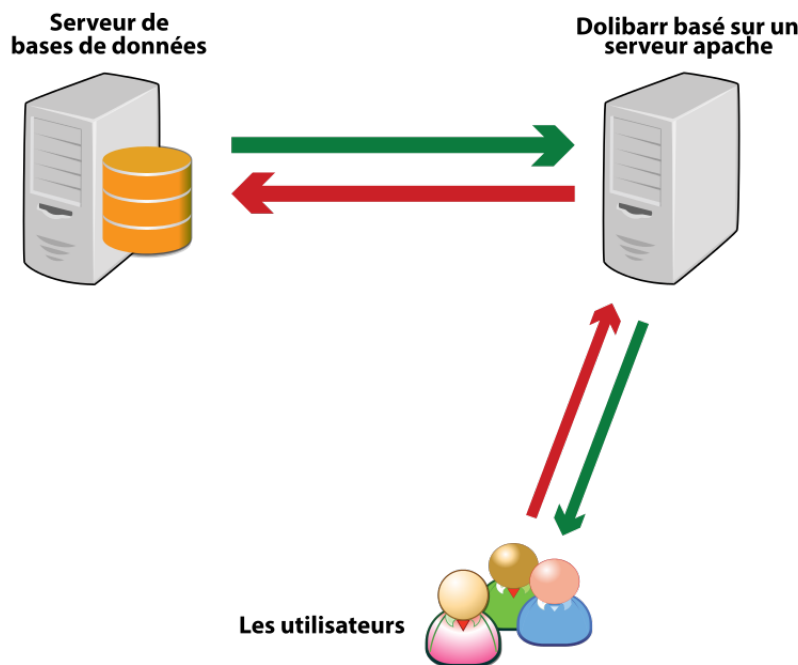


Figure I.4 – Architecture Dolibarr [2]

Pour les non connaisseurs dans le domaine du développement, il existe un auto-installateur qui se charge d'installer la solution avec tous ses pré-requis (apache,mysql,php) [2]. La figure I.5 présente l'écran d'accueil de Dolibarr :

3.3 Etude comparative

Le tableau I.1 présente un comparatif entre les solutions existantes présentées précédemment :



Figure I.5 – Ecran d'accueil de Dolibarr [2]

Solution	Payante	Commentaire	Alertes	Mails	Accueil simple
Fitnet	oui	non	oui	oui	non
Everwin CXM	oui	non	oui	oui	non
Dolibarr	non	non	oui	oui	non

Tableau I.1 – Comparatif entre les solutions existantes

Les applications étudiées, qu'elles soient en versions éditeurs ou même gratuites, et malgré leurs adaptabilités et la diversité de leurs fonctionnalités offertes, restent incapable de satisfaire les exigences de la société. En effet, elles n'intègrent pas un système de commentaire et leur manipulation reste plutôt complexe comparée à la solution à laquelle nous voulons aboutir.

4 Objectifs du projet

Nous voulons offrir un meilleur service dans nos réponses aux collaborateurs à l'aide d'un véritable outil de gestion des demandes. Aujourd'hui nous visons de :

- Faciliter la gestion des demandes.
- Rendre l'interaction entre les collaborateurs situés en Tunisie et en France plus fluide.
- Avoir un système de notification dans l'application, par mail et par sms.
- Mesurer leur taux de satisfaction vis-à-vis des réponses aux demandes.

- Identifier le dysfonctionnement dans notre processus de travail.
- Avoir un suivi et une évaluation.

5 Contexte méthodologique du projet

La grande évolution dans le domaine du développement est accompagnée par une évolution des moyens assurant le bon fonctionnement de ce dernier. D'où l'apparition des méthodes agiles permettant d'organiser le cycle de développement des projets informatiques.

Les méthodes agiles sont basées sur des principes communs définis dans l'Agile Manifesto qui est rédigé par des experts dans ce domaine [5]. Ils se reposent essentiellement sur une approche itérative incrémentale et adaptative évoluant en parallèle avec les besoins du client, afin de livrer un produit de qualité. Il existe plusieurs méthodes agiles, à savoir, la méthode RUP [6], la méthode XP [7], la méthode SCRUM [8] et la méthode RAD [9].

5.1 Le choix de la Méthode Scrum

Dans la majorité des projets, il est difficile d'anticiper les attentes du client. Ceci nous oriente vers une approche itérative permettant de s'adapter aux exigences du client au fur et à mesure de l'avancement du projet. Pour ce faire, nous avons choisi d'adopter la méthode Scrum. Aujourd'hui, Scrum est la méthode agile la plus utilisée. Elle permet de produire une solution de la plus haute qualité dans des bref délais. Cette méthode est munie des atouts suivants :

- Meilleure vue d'ensemble du projet : nous avons une vue globale sur l'avancement du projet par tous les membres des différentes équipes avec un traitement régulier des problèmes rencontrés durant chaque phase.
- Mise à jour des priorités : le client, qui n'est pas nécessairement un informaticien, n'a pas toujours une vision complète sur le produit final. Pour cela, et grâce à la composition séquentielle du contenu des sprints, il bénéficie d'une flexibilité au niveau de la définition, de l'évolution des priorités et des séquences d'activités.
- Qualité du produit mise en avant : Cette méthode se repose sur une évaluation régulière du travail, ce qui permet un meilleur traitement des problèmes (bug), une meilleure

productivité et un produit satisfaisant [10].

5.2 Les rôles et les notions

Les rôles dans Scrum :

- Le product owner est le responsable du produit. Il est généralement le client et c'est lui qui exprime les différentes spécifications fonctionnelles et leurs priorités.
- L'équipe de développement est responsable de la réalisation du livrable. Elle est constituée par l'entreprise et elle est auto-organisée.
- Le scrum master est le premier responsable sur le bon déroulement des processus de la méthode scrum.

Notion :

- Sprint : une itération de travail qui dure entre 15 et 30 jours.
- Le product backlog : représente la liste des fonctionnalités rédigées par le product owner avec tous les correctifs et les améliorations. Il est donc modifiable tout au long du projet. Le product backlog est présenté sous forme d'items.
- Le sprint backlog : est l'ensemble des items planifiés pour le sprint en cours [8].

Conclusion

Dans ce chapitre, nous avons présenté le cadre général du travail, en commençant par la présentation de l'entreprise Urbaprod, passant par la problématique du projet, ainsi qu'une étude des solutions existantes sur le marché. Et pour finir, nous avons présenté la méthode qui va guider notre travail tout au long du projet. Dans le chapitre suivant, nous introduisons les spécifications de notre projet.

Chapitre II

Capture, analyse et spécification des besoins

Plan

1	Étude de l'existant	10
2	Critique de l'existant	11
3	Analyse des besoins	11
3.1	Les acteurs	11
3.2	Les besoins fonctionnels	12
3.3	Les besoins non fonctionnels	12
3.4	Planification du projet	13
3.5	Spécification des besoins : diagramme de cas d'utilisation global . . .	14

Introduction

Ce chapitre représente la case de départ dans notre travail. En effet, nous analysons et spécifions les besoins du projet. Ensuite, nous identifions les différents acteurs. Et enfin, nous modélisons le tout dans un diagramme de cas d'utilisation général qui sera notre file conducteur durant la prochaine phase.

1 Étude de l'existant

Pour soumettre une demande, nos collaborateurs chez Génie des Lieux (GDL) à Paris, ont quelques étape à suivre :

- Téléchargement des fichiers : Le téléchargement des fichiers Autocad avec extension DWG ou les fichiers 3DS sur lesquels nos collaborateurs à Urbapro à Tunis vont travailler. Ce téléchargement s'effectue sur notre plate-forme privée qui nous génère automatiquement un lien pour le téléchargement de ces fichiers ce qui nous mène à l'étape suivante.

- Soumission du formulaire de demande : Ensuite, les collaborateurs chez GDL remplissent un formulaire Excel pour détailler les missions à traiter dans cette demande. Les liens de téléchargement des fichiers téléchargés sont soumis dans ce formulaire. Ce dernier est soumis via un mail.
- Réponse des collaborateurs à tunis : L'équipe d'Urbaproduct répond à cette demande par un autre formulaire Excel prédéfini.
- Le suivi de la demande : Ça se fait à travers l'Email, Skype ou par téléphone.

2 Critique de l'existant

La procédure de travail de UrbaProd semble satisfaisante. En effet, elle répond aux besoins mais elle est répartie sur plusieurs plates-formes ce qui implique l'éparpillement du travail. En même temps, cette solution est trop manuelle, d'où la grande perte de temps à chaque nouvelle tâche. D'autre part, il y a le risque de la faille humaine, e.g. tomber dans l'oubli et ne pas faire une copie de sauvegarde. Ceci implique que l'automatisation de ces procédures et leur centralisation devient une nécessité.

3 Analyse des besoins

Dans cette section, nous introduisons les différents acteurs ainsi que les besoins fonctionnels et non fonctionnels.

3.1 Les acteurs

- L'administrateur : c'est la personne chargée d'affecter les différents rôles et de gérer les comptes.
- Collaborateur-FR : ils envoient les demandes depuis la France.
- Collaborateur-TN : ils assurent la réponse aux demandes.

3.2 Les besoins fonctionnels

- La gestion des utilisateurs : l'administrateur doit disposer d'une interface permettant la gestion des utilisateurs, ainsi que la recherche et la gestion des rôles. En cas d'oubli, les utilisateurs peuvent changer leur mot de passe à travers leur adresse mail.
- La gestion des demandes : l'application doit permettre la création d'une demande, modifier ses données, la gestion de ses différentes phases ainsi que la recherche selon un ou plusieurs critères.
- La gestion des tâches : l'application doit permettre la gestion complète des tâches à faire dans chaque demande, i.e., la création, la modification et la suppression.
- La gestion des clients : l'application doit permettre l'ajout des clients, leurs modification et leurs suppression.
- La gestion des sites : l'application doit permettre l'ajout des sites relatifs à chaque client ainsi que leurs modification et suppression.
- Extraction des données : toutes listes d'utilisateurs, clients, sites, demandes, tâches peuvent être extraite sous la forme de fichier Excel, la demande est extraite sous la forme originale du formulaire utilisé auparavant par l'équipe en France.
- L'envoi des notifications : l'application envoie automatiquement des notifications aux utilisateurs à chaque événement important comme l'ajout ou la modification d'une demande ou l'ajout d'un nouveau commentaire.

3.3 Les besoins non fonctionnels

Les besoins ne s'arrêtent pas au niveau fonctionnel mais ils tendent vers des exigences qui contribuent à une meilleure qualité de l'application. Les plus importantes sont :

- Elle doit être robuste en terme de scalabilité.
- Au niveau de la sécurité, elle doit être invulnérable aux différentes attaques web puisqu'elle sera hébergée sur le web.
- Un autre aspect de la sécurité est le contrôle d'accessibilité, il doit être stricte aux différents acteurs. Nous gardons aussi les traces de chaque modification sur les demandes.

- L’interface Graphique et le traitement Javascript doit être compatible avec les navigateurs Google Chrome et Mozilla Firefox.
- Elle doit se doter d’une interface conviviale et moderne pouvant être exploitée par un simple utilisateur puisque la plupart d’eux sont des architectes.
- L’interface doit répondre aux bonnes pratiques de l’IHM notamment la gestion des erreurs et l’accès aux différents menus [11].

3.4 Planification du projet

3.4.1 Les acteurs du projet

Dans le tableau II.1 nous présentons les différents acteurs participants dans ce projet :

Rôle	Acteur	Mission
Équipe	Anas BEN HAJ ALI	Conception,développement, tests unitaires, déploiement.
SCRUM master	Ons BEN CHEIKH	Assurer le bon déroulement de la méthode SCRUM.
Product owner	Aymen ELLOUZE	Définir les fonctionnalités du produit et s’assurer de leur conformités.

Tableau II.1 – Les acteurs du SCRUM

3.4.2 Backlog produit

Il est élaboré par le product owner. Il comporte toutes les fonctionnalités du produit à développer par l’équipe du travail. Il est utilisé essentiellement pour planifier les releases. A la fin de chaque sprint, nous effectuons une mise à jour du backlog afin de prendre en compte les nouveaux besoins qui surviennent durant les sprints, et d’annuler les idées non concluantes du départ [8]. Dans le tableau II.2, nous présentons le Backlog établi au début du projet.

3.4.3 Les sprints du projet

Pour le bon déroulement du projet, le travail sera découpé en sprint. Ces sprints sont établi à l’aide du backlog produit tout en respectant la priorité des différents modules. Le tableau II.3

présente les sprints du projet.

A la fin de chaque sprint, nous aurons un release qui sera examiné par le product owner afin de planifier les modifications et les évolutions à effectuer dans le sprint suivant.

3.5 Spécification des besoins : diagramme de cas d'utilisation global

La figure II.1 représente le diagramme de cas d'utilisation global de l'application et les différents acteurs qui interfèrent :

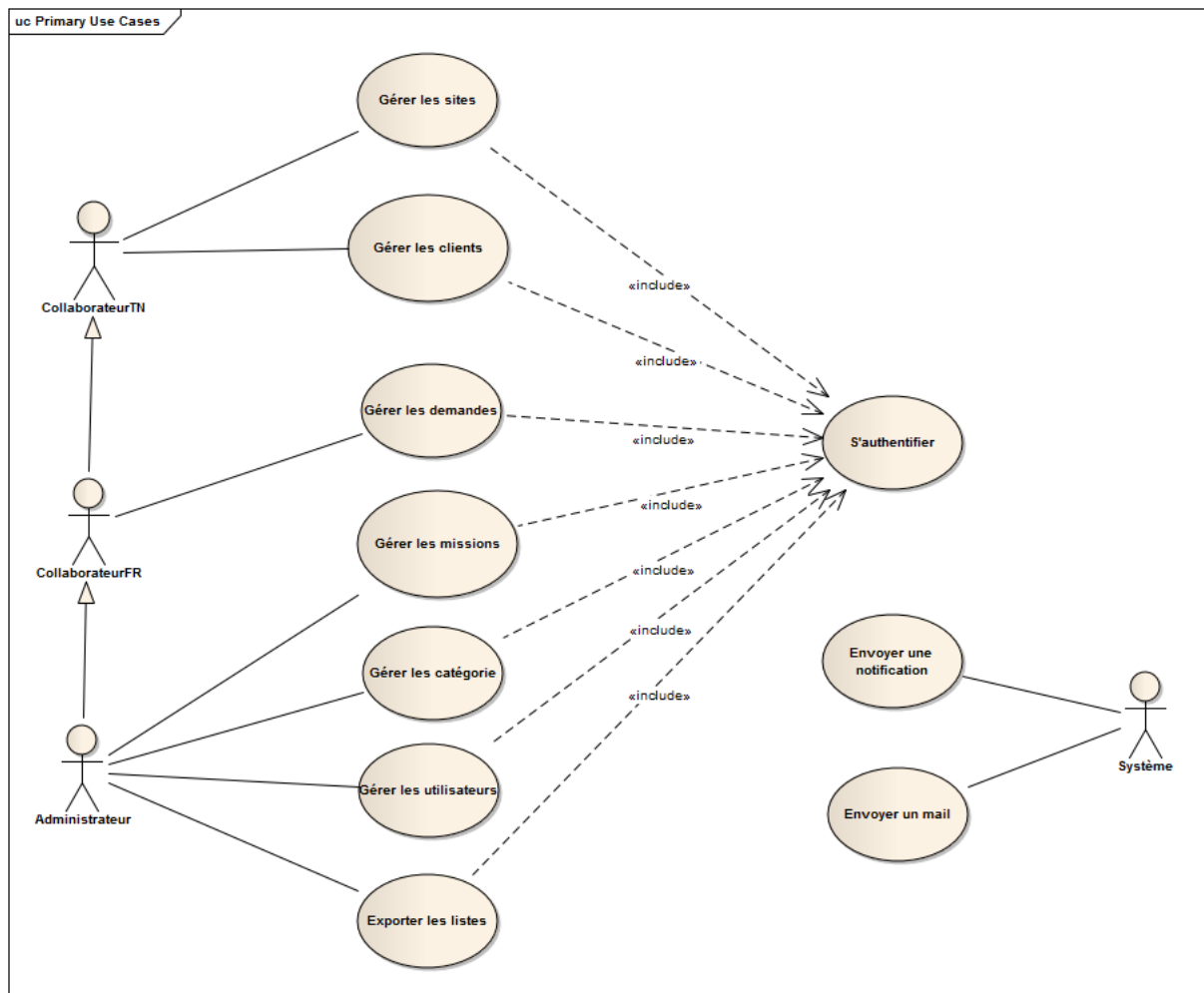


Figure II.1 – Diagramme de cas d'utilisation global

Conclusion

Dans ce chapitre, nous avons déterminé les acteurs principaux dans notre projet ainsi que leur besoins. Ensuite, nous avons établi le Backlog produit sur lequel nous nous sommes appuyés pour bâtir nos sprints. Cette étude sera notre base de travail dans le restant du chemin à savoir : la conception et la réalisation de notre projet. Dans le chapitre suivant nous allons exposer notre vue conceptuelle vis-à-vis du projet.

II.3 Analyse des besoins

Nom	Description	Effort
Ajouter une mission	L'administrateur peut ajouter une mission.	2
Modifier une mission	L'administrateur peut modifier une mission.	3
Rechercher d'une mission	L'administrateur peut rechercher une mission par son titre.	2
Exporter les missions	L'administrateur peut exporter la liste des missions sous plusieurs formats.	2
Ajouter un client	Le collaborateur peut ajouter un client.	2
Modifier un client	Le collaborateur peut modifier un client.	3
Rechercher un client	L'administrateur peut rechercher un client.	2
Exporter les clients	L'administrateur peut exporter la liste des client sous plusieurs formats.	2
Affecter un site au client	Le collaborateur peut affecter un site au client.	3
Modifier un site	Le collaborateur peut modifier un site.	3
Rechercher un site	L'administrateur peut filtrer les sites.	2
Exporter les sites	L'administrateur peut exporter la liste des sites sous plusieurs formats.	2
Ajouter une catégorie	L'administrateur peut peut ajouter une catégorie de demande.	2
Modifier une catégorie	L'administrateur peut modifier une catégorie	3
Rechercher une catégorie	L'administrateur peut rechercher une catégorie.	2
Exporter les catégories	L'administrateur peut exporter la liste des catégories sous plusieurs formats.	2
Ajouter un utilisateur	L'administrateur peut ajouter un utilisateur.	2
Modifier un utilisateur	L'administrateur peut modifier un utilisateur.	3
Désactiver un utilisateur	Désactiver un utilisateur	1
Rechercher un utilisateur	L'administrateur peut filtrer les utilisateurs.	2
Exporter les utilisateurs	L'administrateur peut exporter la liste des utilisateurs sous plusieurs formats.	3
Initier une demande	Le collaborateur peut soumettre une demande.	4
Modifier l'état d'une demande	Le collaborateur peut prendre en charge, livrer ou annuler une demande.	2
Modifier une demande	Le collaborateur peut modifier une demande.	3
Aimer une demande	Le collaborateur peut aimer une demande.	4
ne pas aimer une demande	Le collaborateur peut ne pas aimer une demande.	4
Commenter une demande	Le collaborateur peut commenter une demande.	5
Rechercher une demande	L'administrateur peut filtrer les demandes selon plusieurs critères.	2
Exporter la listes des demandes en Excel.	Le collaborateur peut exporter la liste des demandes sous format Excel avec des informations restreintes.	3
Exporter la listes des demandes	L'administrateur peut exporter la liste des demandes sous différents formats.	2
Exporter une demande	Le collaborateur peut exporter la demande originale sous format Excel.	3
Accéder à un fil d'actualités	Le collaborateur peut accéder à un fil d'actualités.	5
Écran d'accueil	Le collaborateur peut accéder à l'écran d'accueil.	4
Voir le feed-back graphique de l'avancement des demandes	Le collaborateur a un suivi graphique des demandes par mois.	4
Envoyer une notification	Le système doit notifier les utilisateurs à chaque nouvelle demande ou commentaire.	5
Envoyer un mail de notification	Le système doit émettre un mail aux utilisateurs à chaque nouvelle demande.	3

Tableau II.2 – Backlog du produit

II.3 Analyse des besoins

Sprint	description	durée
sprint 0	Établissement du cahier de charge.	3 semaines
sprint 1	Choix de la technologies et formation sur cette technologie.	2 semaines
sprint 2	Développement des modules gestion des missions et gestion des catégories.	1 semaine
sprint 3	Développement des modules gestion des clients et gestions des sites.	1 semaine
sprint 4	Développement du module gestion des utilisateurs.	1 semaine
sprint 5	Développement du module gestion des demandes.	3 semaines
sprint 6	Développement de la partie exportation des demandes.	1 semaine
sprint 7	Développement de la gestion des commentaires et des "j'aimes".	3 semaines
sprint 8	Développement du module de notification.	2 semaine
sprint 9	Développement du fil d'actualité.	3 semaines
sprint 10	Développement de l'écran commun et du suivi graphique.	1 semaine
sprint 11	Test et déploiement.	1 semaine

Tableau II.3 – Les sprints du projet

Chapitre III

Etude conceptuelle

Plan

1	Architecture physique	18
2	Architecture logicielle	19
3	Raffinement des diagrammes des cas d'utilisations	21
3.1	module gestion des demandes	21
3.2	module gestion des utilisateurs	22
3.3	Module gestion des paramètres de la demande	23
3.4	Module exportation des données	24
4	Diagrammes structurels	25
4.1	Diagramme de paquetages	25
4.2	Diagramme de classes	28
5	Diagrammes comportementaux	29
5.1	Diagrammes de séquences	29
5.2	Diagramme d'activité	30

Introduction

Cette partie est très importante dans le cycle de vie du projet. Toute la base du projet est construite ici en théorique, après, il ne reste que la mise en oeuvre.

1 Architecture physique

Notre application est une application client/serveur qui dispose d'une architecture 3 tiers. C'est l'architecture la plus répandue dans les application web, dans laquelle la logique métier, l'accès et le stockage de données et l'interface utilisateur sont maintenus chacune dans un mo-

dule indépendant et pouvant être répartis sur des plates-formes différentes ou même regroupés sur une seule machine. La figure III.1 donne un aperçu sur cette architecture.

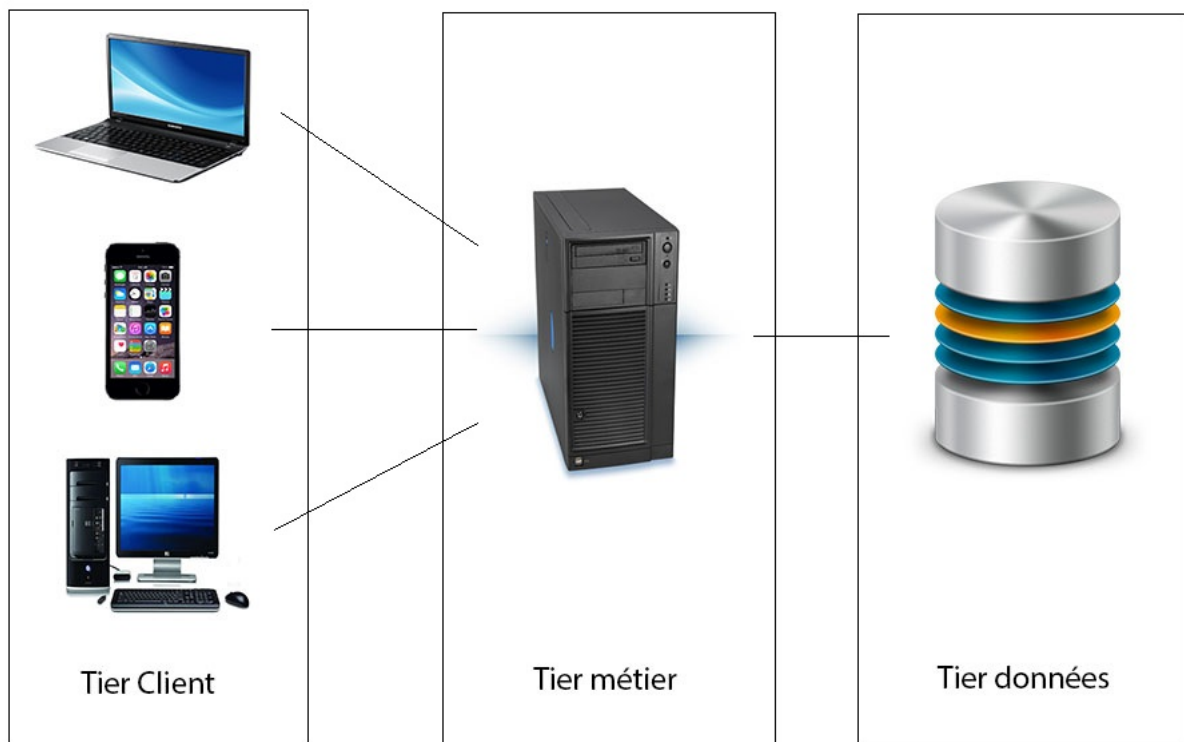


Figure III.1 – Architecture physique

- Le client : ou la partie présentation qui est dans notre cas le navigateur web (google Chrome, Mozilla Firefox etc).
- Le serveur d'application : ou la partie métier, elle est chargée de faire des traitements spécifiques afin de répondre au requête HTTP du client venant de la couche présentation.
- Le serveur de base de données : dans notre cas c'est le système de gestion de base de données MySQL.

2 Architecture logicielle

Maintenant, nous présentons l'architecture applicative de notre application, sa structure en couche et leur description, à savoir, la couche présentation, la couche service et la couche accès

à la base de données. Nous modélisons par la figure III.2 ces couches ainsi que les différents invocations entre eux.

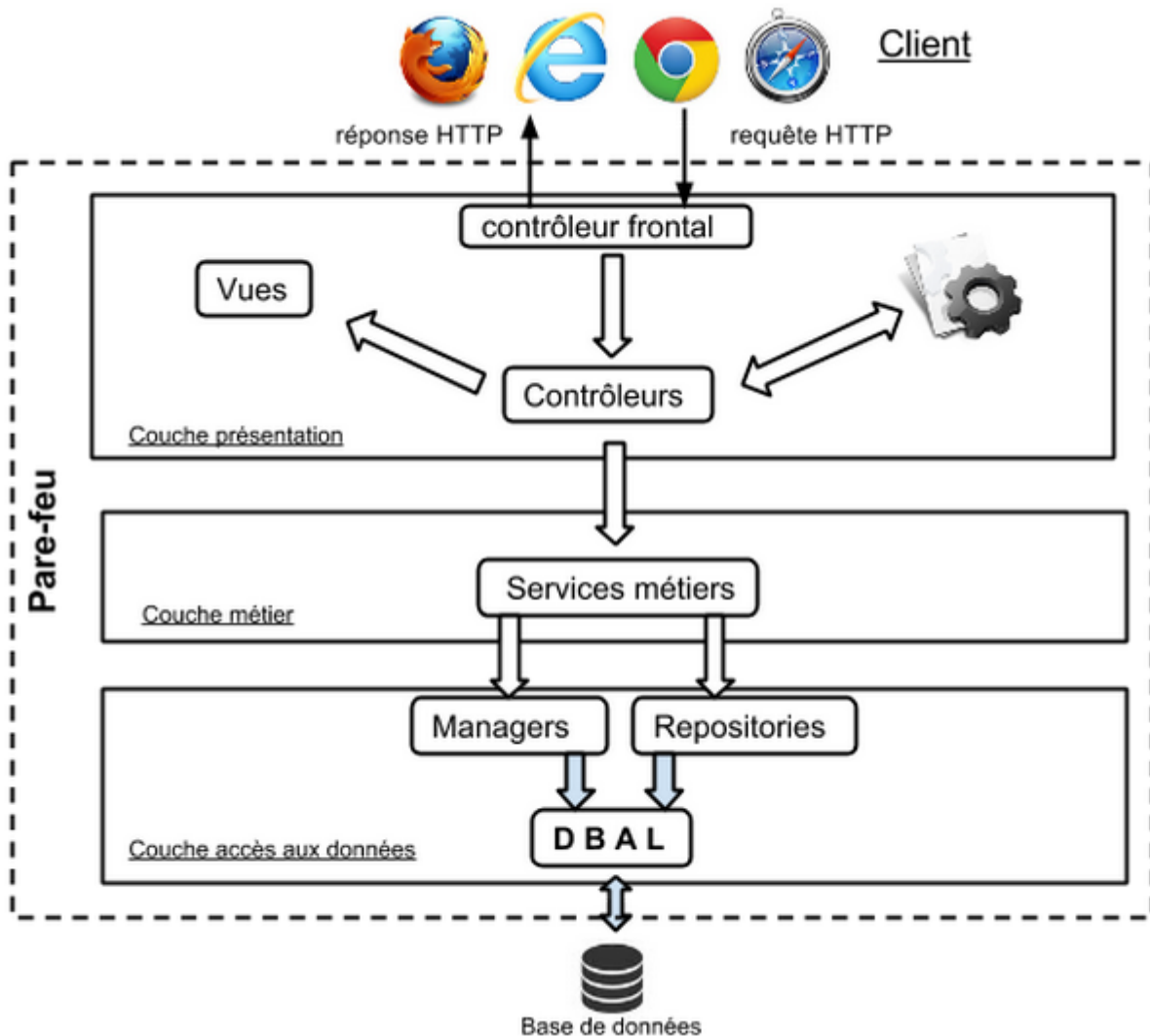


Figure III.2 – Architecture applicative de la solution

- **La couche présentation** : elle représente le contact de l'utilisateur avec l'application, elle lui permet le pilotage et la configuration de l'application. Elle invoque le service métier adéquat à la requête du client et lui retourne le résultat. Cette couche est composée de :
 - **Contrôleur frontal** : il représente le point d'entrée de notre application.
 - **Contrôleurs** : ils décident selon la requête du client et les configuration du système,

quel service sera invoqué et quelle vue sera rendue, autrement dit, ils orchestrent les différents éléments.

- Vues : le résultat de la requête de l'utilisateur est récupéré et traduit en HTML et Javascript puis renvoyé au client sous forme de vue selon la charte graphique utilisée dans l'application.
- **La couche métier** : c'est l'élément qui englobe la logique métier et ses traitements spécifiques. Elle est invoquée via les contrôleurs pour traiter les requêtes client.
- **La couche accès à la base de données** : cette couche assure la communication avec la source de données afin d'assurer la séparation entre la logique métier et la logique accès au données. Elle est composée de :
 - Managers : Ils assurent la persistance des données dans la base de données.
 - Repositories : Ils assurent l'extraction des données de la base de données.
 - Database Abstract Layer DBAL : Elle offre un accès facile et rapide à la base de données [12]. Elle procure plusieurs service : l'insertion, la mise à jour, la suppression et la lecture des données. Elle est caractérisée par un Object Relation Mapping (ORM) qui traduit les tables de la base en objets facilement manipulable.

Dans les paragraphes précédents, nous avons présenté les modèles architecturaux de notre application ce qui nous mènera à la partie conception dans la quelle nous allons détailler nos diagrammes de cas d'utilisation, présenter les diagrammes de classes ainsi que quelques exemple des cas d'activité les plus importants.

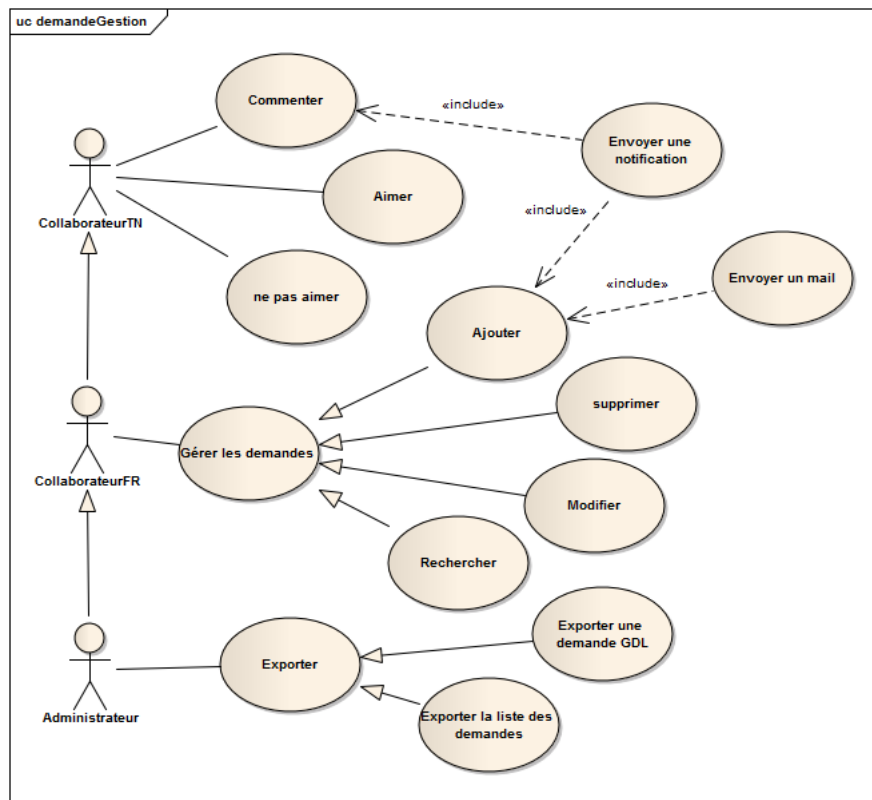
3 Raffinement des diagrammes des cas d'utilisations

Dans cette partie, nous décortiquons les cas d'utilisation vus dans le chapitre précédent d'une façon plus détaillée afin de clarifier le fonctionnement de notre système.

3.1 module gestion des demandes

Ce module regroupe toutes les rubriques de la gestion d'une demande. La création des demandes est conduite par les collaborateurs en France. Les collaborateurs à Tunis auront une

III.3 Raffinement des diagrammes des cas d'utilisations



- Description du cas d'utilisation : ajout d'une demande
Le tableau [??](#). décrit le cas d'utilisation "ajout d'une demande".
- Description du cas d'utilisation : commenter une demande
Le cas d'utilisation "ajout d'un nouveau commentaire" est décrit dans le tableau [III.2](#).

Ce sont les différentes fonctionnalités offertes afin de gérer les collaborateurs au sein de l'application. Le figure III.4 illustre le diagramme de cas d'utilisation de gestion des utilisateurs.

Ce cas est décrit par le tableau III.3.

III.3 Raffinement des diagrammes des cas d'utilisations

Titre	Ajouter une nouvelle demande.
Acteur	Collaborateur FR.
Résumé	Le collaborateur peut ajouter une nouvelle demande en spécifiant les différentes informations relatives à son exécution, e.g. date limite, client, format du rendu.
Pré condition	Le collaborateur doit être authentifié.
Scénario principal	<ol style="list-style-type: none">1. Le collaborateur se rend dans la page d'accueil ou sur la page de la liste des demandes.2. Le collaborateur affiche le pop-up du formulaire de la demande.3. Le collaborateur saisie les informations nécessaires et appuie sur créer.4. Le collaborateur joint les fichiers sur lesquels les collaborateurs à Tunis vont travailler.5. Le système enregistre la demande.6. Le système émet une notification à tous les utilisateurs pour les informer de la nouvelle demande.7. Le système envoie un mail à tout les utilisateurs pour les informer de la nouvelle demande.8. Le système affiche un message de confirmation et le redirige vers la page d'accueil ou sur la page de la liste des demandes.
Scénario d'exception	<p>Erreur dans les informations saisies :</p> <ul style="list-style-type: none">– Le système affiche les messages d'erreur devant les champs concernés.– Nous reprenons depuis l'étape 3.

Tableau III.1 – Ajout d'une demande

3.3 Module gestion des paramètres de la demande

Cette partie regroupe les différentes fonctionnalités qui servent à paramétrer les demandes. Vue que les missions effectuées dans l'entreprise sont connues d'avance, c'est le directeur de l'entreprise qui est chargé de les prédéfinir. Les clients et leurs sites sont définis au préalable comme ils peuvent être définis lors de la soumission d'une nouvelle demande. La figure [III.5](#) montre ces différents cas d'activités.

III.3 Raffinement des diagrammes des cas d'utilisations

Titre	Commenter une demande.
Acteur	Collaborateur TN.
Résumé	Afin d'intervenir dans le cycle de vie d'une demande ou de poser son point de vue, un collaborateur peut ajouter un commentaire tout en joignant des images.
Pré condition	Le collaborateur doit être authentifié.
Scénario principal	<ol style="list-style-type: none">1. Le collaborateur se rend dans la file d'actualité ou dans la vue d'une demande.2. Le collaborateur saisie son commentaire.3. Le collaborateur joint les images.4. Le système enregistre le commentaire.5. Le système affiche le nouveau commentaire.6. Le système place le collaborateur dans l'espace de commentaire afin qu'il puisse ajouter un autre.
Scénario d'exception	Erreur dans le type des fichiers joint : <ul style="list-style-type: none">– le système affiche l'impossibilité de charger ce type de fichier (fichier.php) .– on se rend depuis l'étape 4.

Tableau III.2 – Ajouter un nouveau commentaire

Description du cas d'utilisation : ajout d'un nouveau site

Ce cas d'utilisation est décrit par le tableau [III.4](#).

3.4 Module exportation des données

Ce module permet en fait plusieurs types d'exportation. Tout d'abord, il permet l'extraction de la demande sous sa forme originale adaptée avant le développement de notre application (comme indiqué dans l'annexe [A.1](#)). Pour le suivi du travail ou un contrôle quotidien, nous pouvons exporter les données dans des fichiers Excel. Vue la nature des autres activités de l'entreprise, nous aurons besoins aussi d'autre types de flux d'information i.e. JSON et XML. La figure [III.6](#) met en évidence ces cas d'utilisation.

Description du cas d'utilisation : exporter la liste des demandes.

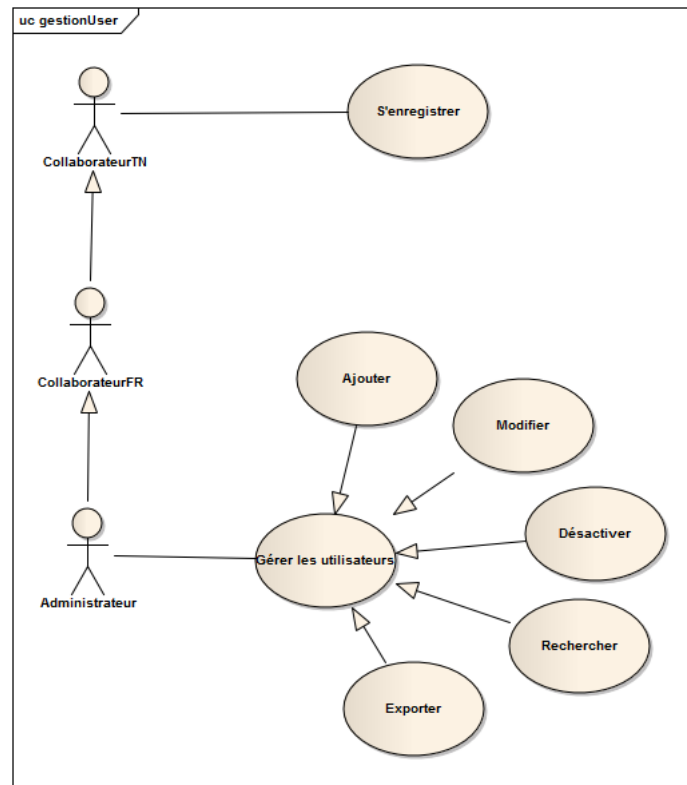


Figure III.4 – Diagramme de cas d'utilisation gestion utilisateurs

Le tableau [III.5](#) décrit ce cas d'utilisation .

Après la présentation des différents diagrammes de cas d'utilisation, nous détaillons dans la section suivante, la conception de notre application à travers les diagrammes structurels et comportementaux.

4 Diagrammes structurels

4.1 Diagramme de packages

Pour une application maintenable, nous avons opté pour la division en packages. Le package est un regroupement de classe ou d'éléments de modélisation de préférence de même type. Donc ce diagramme met en évidence l'organisation de notre travail [\[13\]](#). La figure [III.7](#) présente le diagramme de packages.

En fait, ce diagramme met en évidence le modèle Model View Controller (MVC) [\[14\]](#), à

Titre	Ajout d'un utilisateur.
Acteur	Administrateur , Collaborateur.
Résumé	Ce cas permet à l'administrateur de bien organiser les utilisateurs de l'application en leur affectant les différents rôles. En fait, chaque rôle permet à son utilisateur des accès spécifiques et limités.
Pré condition	Un administrateur est authentifié.
Scénario principal	<ol style="list-style-type: none"> 1. L'administrateur envoie la vue d'enregistrement à un collaborateur. 2. Le collaborateur saisi ses informations. 3. Le système enregistre ces informations. 4. Le système met à jour la liste des collaborateurs. 5. L'administrateur affecte les rôles de ce collaborateur et l'active pour qu'il puisse accéder à l'application.
Scénario d'exception	L'utilisateur essaie de s'inscrire avec une adresse mail existante : <ul style="list-style-type: none"> – Le système affiche l'erreur près du champs concerné. – Le processus reprend depuis l'étape 2.

Tableau III.3 – Ajout d'un utilisateur

travers les packages Views, Controller, Repository, Manager et Entity.

Entity : il regroupe les entités de notre application.

Manager : il regroupe les éléments gérant la persistance dans la base de données.

Repository : il regroupe les éléments responsable de l'extraction des données de la base de données.

Views : il regroupe les vues de notre application. Il représente le contacte directe avec l'utilisateur.

Controller : il regroupe les contrôleurs de l'application. Ils jouent le rôle de chef d'orchestre en appelant le modèle et en le redirigent vers la vue convenue.

Il existe aussi d'autre packages :

Service : il regroupe les services liés à notre application.

Titre	Ajout d'un nouveau site.
Acteur	Collaborateur.
Description	La plupart des clients sont de grande entreprise donc ils possèdent plusieurs sites et à chaque fois nous travaillons sur l'un d'eux. C'est la raison pour laquelle l'affectation du client seulement n'est pas suffisante. Nous devons donc, dans chaque demande, préciser le site sur lequel nous travaillons.
Pré condition	Le collaborateur doit être authentifié.
Scénario principal	<ol style="list-style-type: none"> 1. Le collaborateur se rend sur la page des sites. 2. Le collaborateur affiche le pop-up de l'ajout d'un site. 3. Le collaborateur indique le client. 4. Le collaborateur saisi les informations du site. 5. Le système enregistre le nouveau site.
Scénario alternatif	<ol style="list-style-type: none"> 1. Le collaborateur saisi les informations d'une nouvelle demande. 2. Le collaborateur ne trouve pas le site voulu. 3. Nous reprenons dès l'étape 2 du scénario principal.
Scénario d'exception	<p>Le collaborateur ne trouve pas de client disponible :</p> <ul style="list-style-type: none"> – Le système affiche l'erreur indiquant l'absence des client. – Le collaborateur doit ajouter un nouveau client. – Le scénario reprend depuis l'étape 1.

Tableau III.4 – Ajout d'un nouveau site

Form : il regroupe les formulaires de l'application modélisés par des classes.

Config : il contient le routage de l'application et les services.

Titre	Exporter la liste des demandes.
Acteur	Administrateur.
Description	Afin de faire le suivi du travail dans l'entreprise, ou de facturer le travail effectué par l'équipe, le directeur a besoin d'une vue globale sur les demandes.
Pré condition	Un administrateur est authentifié.
Scénario principal	<ol style="list-style-type: none"> 1. L'administrateur se rend sur la page d'administration. 2. L'administrateur affiche les demandes. 3. L'administration choisit le type de flux de sortie. 4. L'administrateur exporte les données.

Tableau III.5 – Exporter la liste des demandes

4.2 Diagramme de classes

● Le modèle du domaine

La figure [III.8](#) illustre le modèle du domaine de l'application web avec les différentes dépendances.

Ce sont les différentes entités qui constituent le modèle du domaines. Elles représentent la façon avec laquelle les données sont stockés dans la base. Dans le [tableau III.9](#), nous représentons le rôle de chacune de ces entités.

● Diagramme de classe des contrôleurs

La figure ?? illustre le diagramme de classe des contrôleurs.

Les contrôleurs sont invoqués selon les routes, chaque route pointe vers une seule méthode d'un contrôleur bien spécifique. Donc une seule route ne peut pas invoquer plusieurs actions simultanément.

Entité	Description
Utilisateur	Elle représente un collaborateur chez l'entreprise, elle contient les informations de chacun d'eux et précise leur rôle dans l'entreprise.
Demande	Cette entité représente les exigences du client transmises par nos collaborateurs en France. Les exigences sont prédéfinies et représentées par les missions.
Commentaire	Ce sont les différentes impressions, interrogations, ou déclarations que laissent les utilisateurs à propos d'une demande. Ils peuvent être accompagnés par des photos.
Fichier	Il peut être un fichier attaché à une demande, une image dans un commentaire ou l'image d'un utilisateur.
Notification	C'est un signal envoyé au utilisateur pour leur notifier d'un nouvel événement.
Jaime	Ceux sont les impressions des collaborateurs à propos d'une demande. Soit ils l'aiment soit ils ne l'aiment pas.
Catégorie	La catégorie d'une demande sert à la facturation.
Mission	C'est une tâche que nous pouvons effectuer dans notre entreprise.
Client	Elle représente nos clients.
Site	Elle représente où se situe le locale de l'entreprise avec laquelle nous traitons.

Tableau III.6 – Descriptif des entités de l'application

5 Diagrammes comportementaux

5.1 Diagrammes de séquences

● Diagramme de séquence du cas : ajouter une demande

Dans la figure [III.10](#), nous présentons la création d'une demande à travers les principales interactions entre les différents éléments de l'application. L'exemple traité est le cas d'un collaborateur qui crée une demande avec des données correctes.

Le collaborateur doit remplir tous les champs obligatoires pour que la demande soit correcte. Dans le cas de la soumission avec un champ vide, l'erreur s'affiche près de ce champs. Dans le cas du contournement du vérifieur frontal du formulaire, nous captons l'erreur depuis la base de données et elle est affichée de nouveau dans le formulaire de création. Il y a plusieurs autres interactions que nous voulons préciser lors de l'envoi

d'une requête, notamment le point d'entrée, à savoir, notre contrôleur frontal, le système de routage et le noyau de SF2. Lors de la réponse, DemandeController invoque le moteur de vue TWIG pour l'affichage de la demande dans l'accueil.

● Diagramme de séquence du cas : suppression d'un site

La figure [III.11](#) illustre le cas du suppression d'un site.

Afin de supprimer un site, il faut d'abord vérifier s'il est entrain d'être traité par une demande. Sinon nous pouvons le supprimer sans problème et le système nous affiche la nouvelle liste sans le site supprimé. Le SiteRepository est le responsable de la vérification de l'existence du site dans une demande et de nous rendre la liste des sites de la base de données. Le SiteManager nous permet de supprimer le site après cette vérification.

5.2 Diagramme d'activité

Dans la section précédente, nous avons détaillé le déroulement de quelques scénario d'utilisations à travers des diagrammes de séquences. Dans cette partie nous allons expliquer le déroulement de l'envoi d'une notification via le diagramme d'activités. Ce diagramme est le plus adéquat pour modéliser l'acheminement du flot de contrôle et des flots de données.

La figure [III.12](#) met en exergue le processus de l'envoi d'une notification.

Conclusion

Dans ce chapitre, nous avons préparé nos packages, nos diagrammes, nos séquences d'utilisation, donc tout ce qu'on a besoin pour l'élaboration de l'application. Dans le chapitre suivant, nous allons entrer dans le vif du sujet et réaliser le projet.

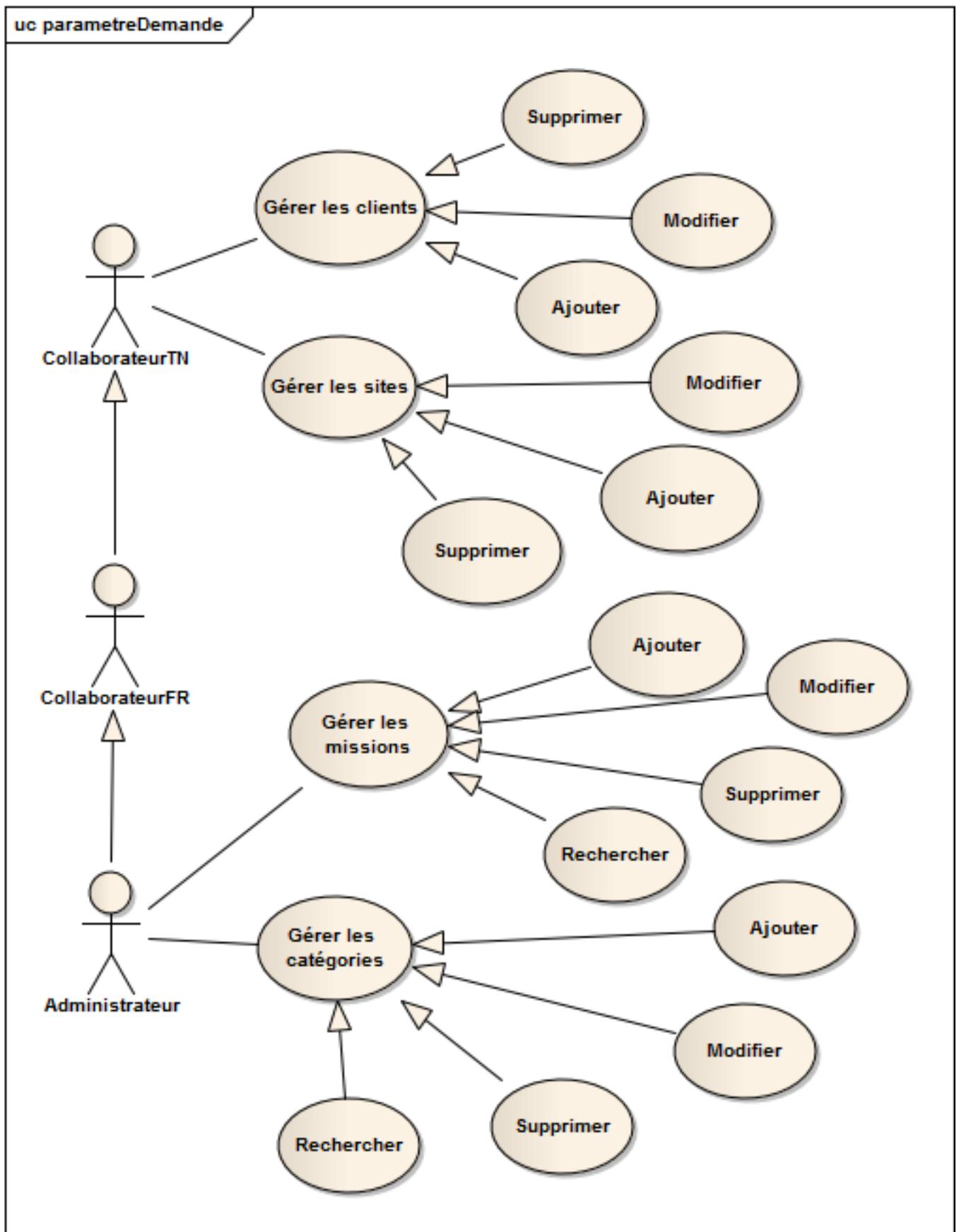


Figure III.5 – Diagramme de cas d'utilisation du paramétrage d'une demande

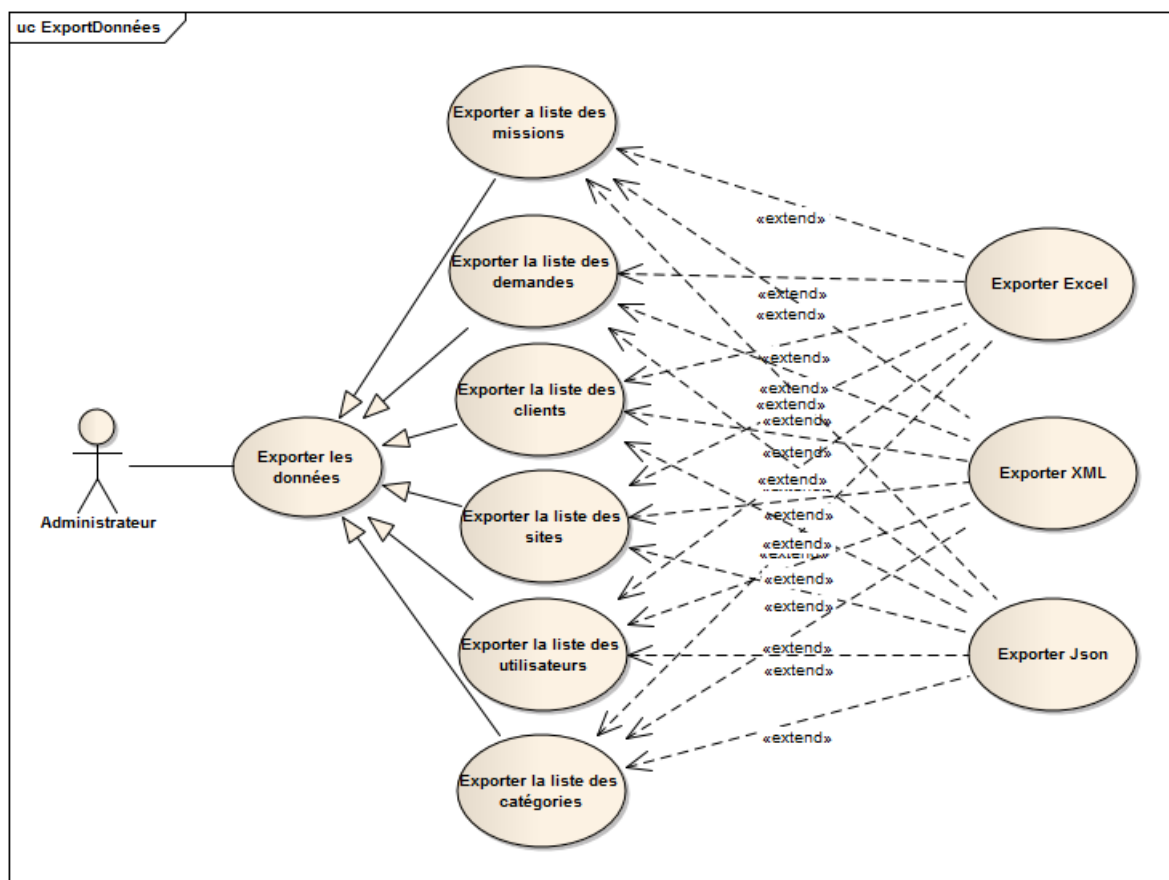


Figure III.6 – Diagramme de cas d'utilisation exportation des données

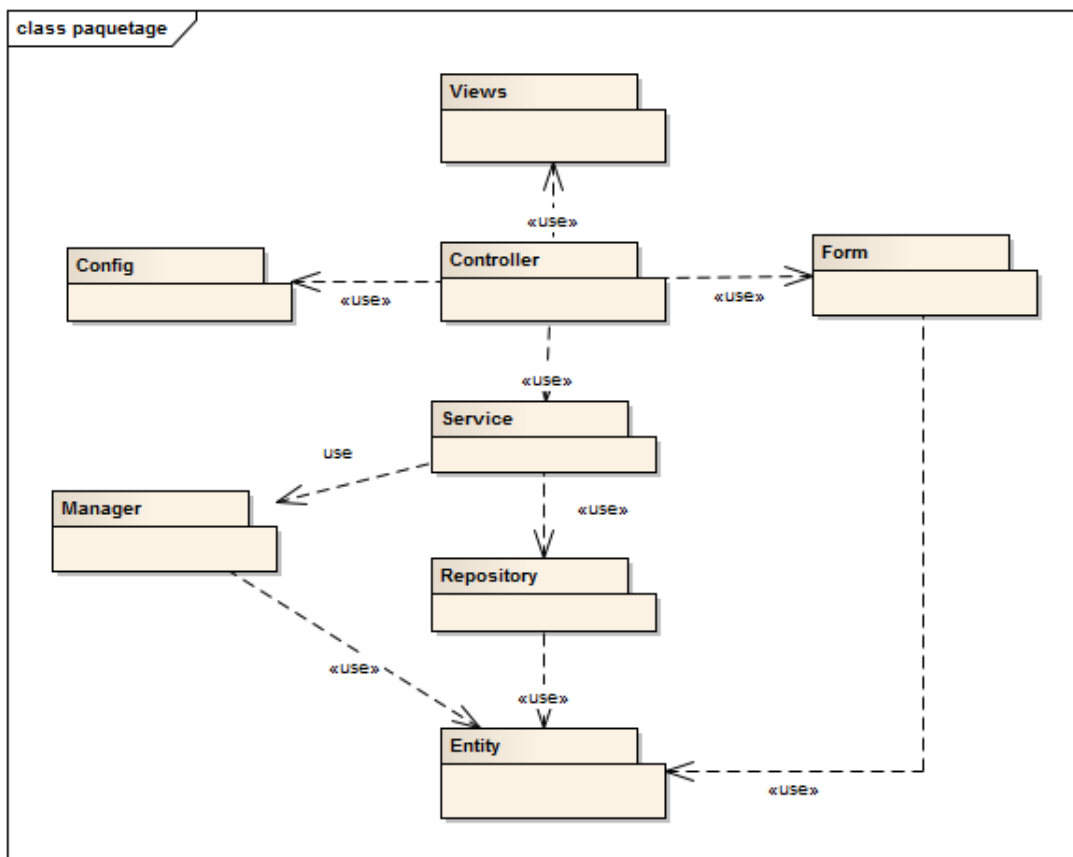


Figure III.7 – Diagramme de paquetage

III.5 Diagrammes comportementaux

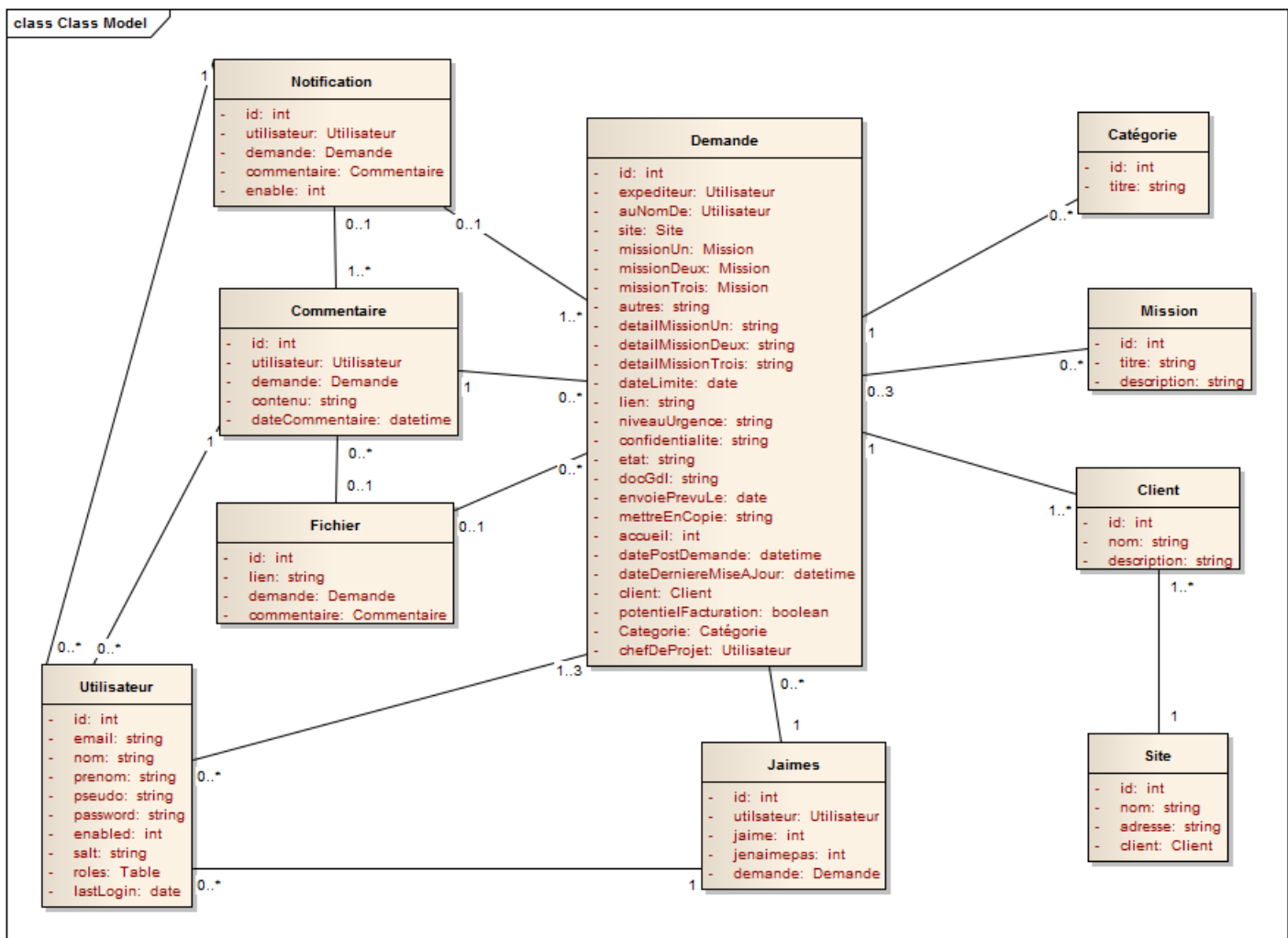


Figure III.8 – Le modèle du domaine

III.5 Diagrammes comportementaux

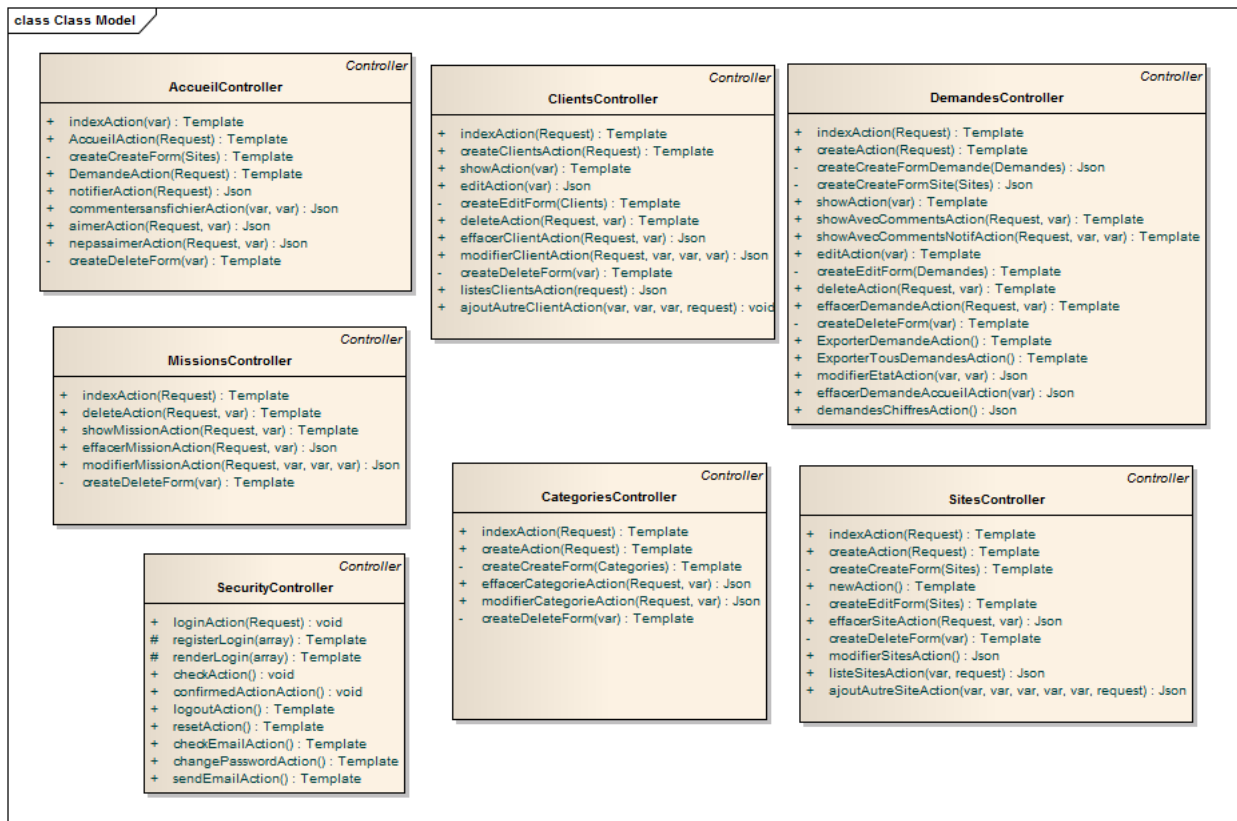


Figure III.9 – Diagramme de classe des contrôleurs

III.5 Diagrammes comportementaux

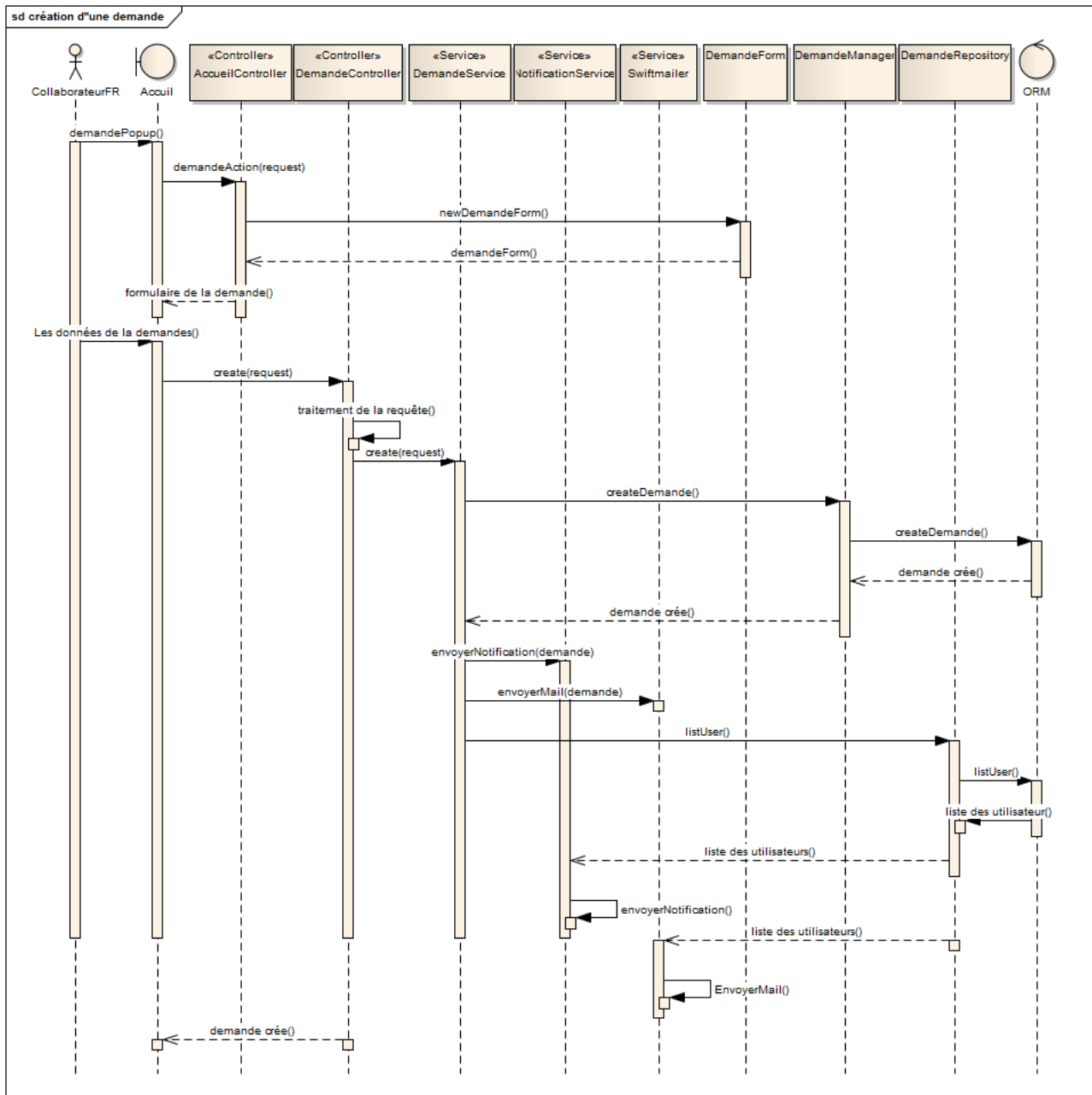


Figure III.10 – Diagramme de séquence ajout d'une demande

III.5 Diagrammes comportementaux

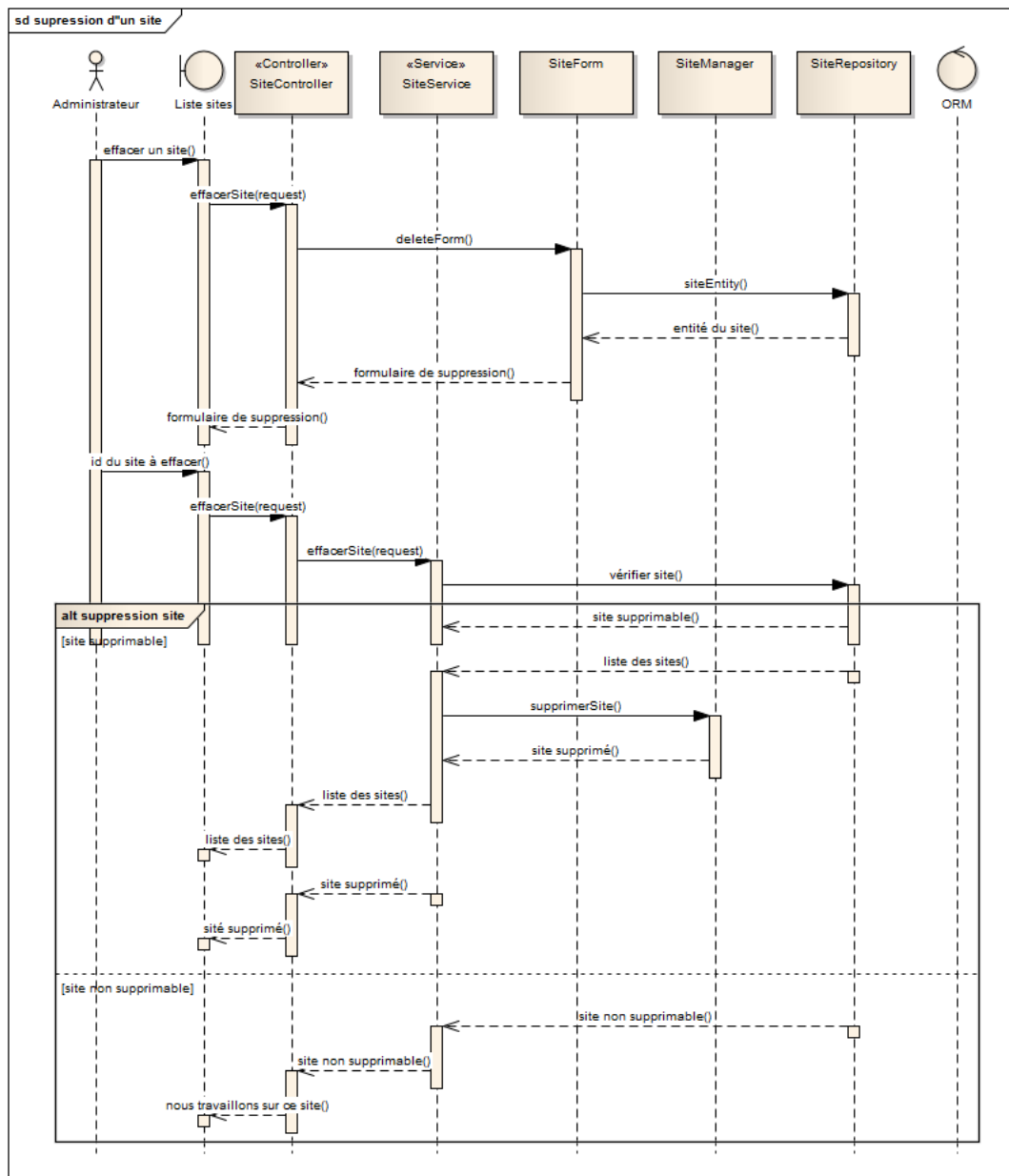


Figure III.11 – Diagramme de séquence suppression d'un site

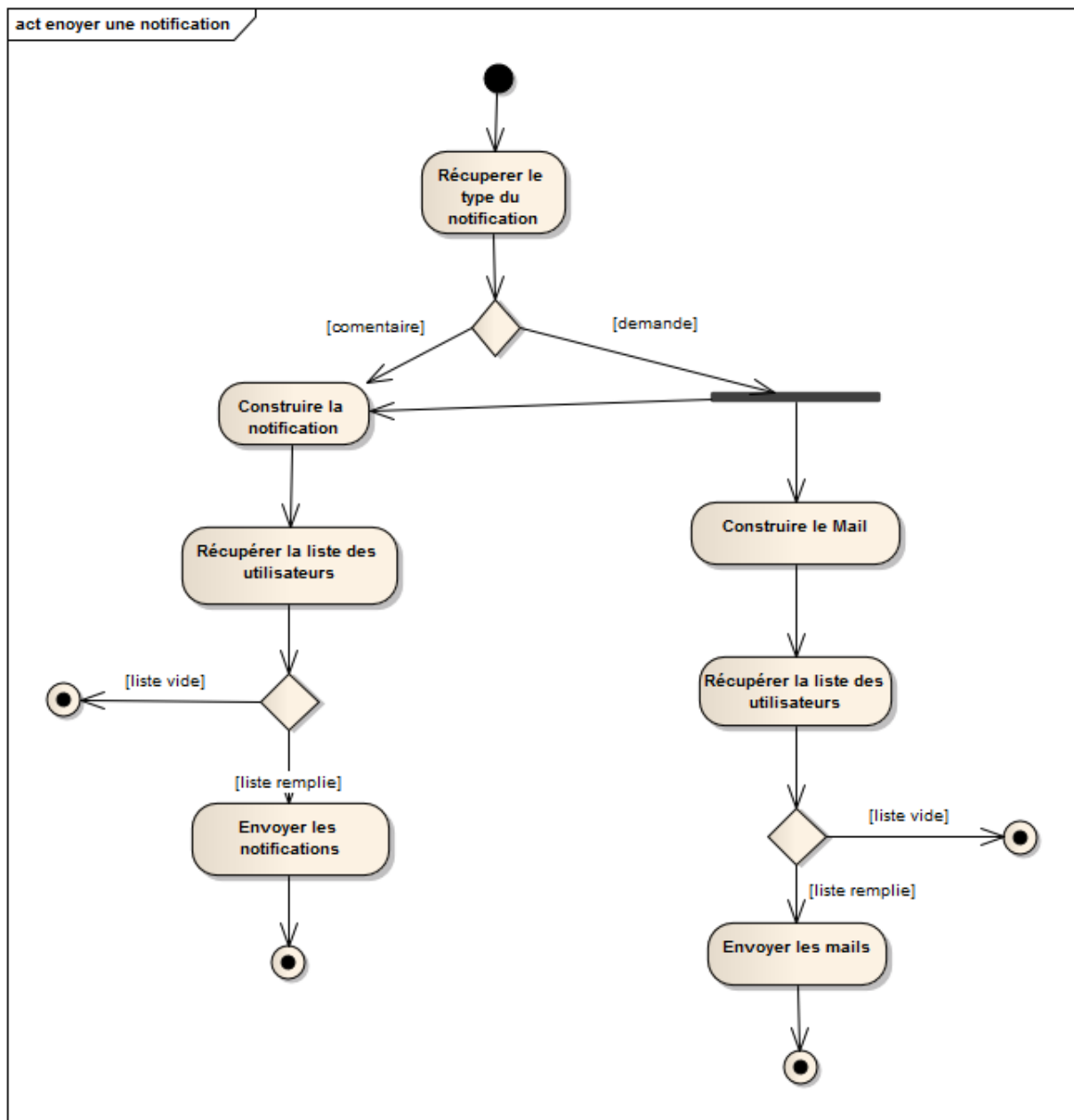


Figure III.12 – Diagramme d'activité envoie d'une notification

Chapitre IV

Réalisation

Plan

1	Environnements de travail et choix techniques	39
1.1	Environnements de travail	39
1.2	Technologies	40
2	Présentation de l'application	44

Introduction

Dans les chapitres précédents, nous avons détaillé la méthodologie suivie durant notre travail ainsi que la conception de notre application. Dans l'étape suivante, nous procédons à la présentation de notre environnement de travail et les différentes technologies utilisées pour enfin terminer avec les principales fonctionnalités réalisées et leurs interfaces.

1 Environnements de travail et choix techniques

Cette section met l'accent sur les logiciels utilisés durant ce travail. Puis nous abordons les choix des différentes technologies mises en place.

1.1 Environnements de travail

Les principaux logiciels utilisés sont :

- Netbeans 8.0.2 : c'est un environnement de développement intégré (IDE) qui supporte plusieurs langages e.g., php, java, c. C'est un produit oracle qui est maintenu à jour pour supporter les dernières technologies et framework les plus utilisés [15].

- Microsoft Office : Word 2013 pour la rédaction du cahier de charge et Excel 2013 pour l'exportation des différentes informations dans notre application.
- Adobe Photoshop CS6 : utilisé pour le traitement des images utilisées tout le long du stage que ce soit dans le rapport, élaboration du cahier de charge ou le développement de l'application.
- WampServer 2.5 : pour la simulation locale du serveur web, nous avons choisi ce serveur pour la facilité du partage des données dans un serveur local ou à distance c'est à dire pour des utilisateurs situés à l'extérieur de notre réseau.
- Mozilla Firefox, Google Chrome : nous avons effectué le test des vues dans ces navigateurs sur un écran pc 23 pouces et sur une tablette Galaxy Tab 4 de Samsung pour une bonne qualité de rendu.
- Git et Github : afin de garder une trace de notre travail, d'avoir la possibilité d'un retour en arrière en cas de problème, nous avons utilisé l'outil de versionning Git qui offre la possibilité de sauvegarder notre travail avec un simple "commit" et permet même d'effectuer des sauvegardes en ligne avec le "push". La figure IV.1 et IV.2 présentent l'interface de Github avec quelques "commit" et toutes les contributions effectuées pendant notre travail.
- Sublime Text 2 : pour le développement ou la modification de quelques scripts.
- Miktex : pour la génération du rapport.

1.2 Technologies

Durant notre travail, plusieurs choix techniques ont été faits. Dans ce qui suit, nous illustrons les plus importants :

– Le Framework Symfony2.

Une exigence du client dès le début du projet, était de travailler avec le langage PHP sur une base de données MySQL. L'idée du client était de travailler avec du PHP natif étant donné qu'un Framework n'est pas indispensable pour le développement d'une ap-

IV.1 Environnements de travail et choix techniques

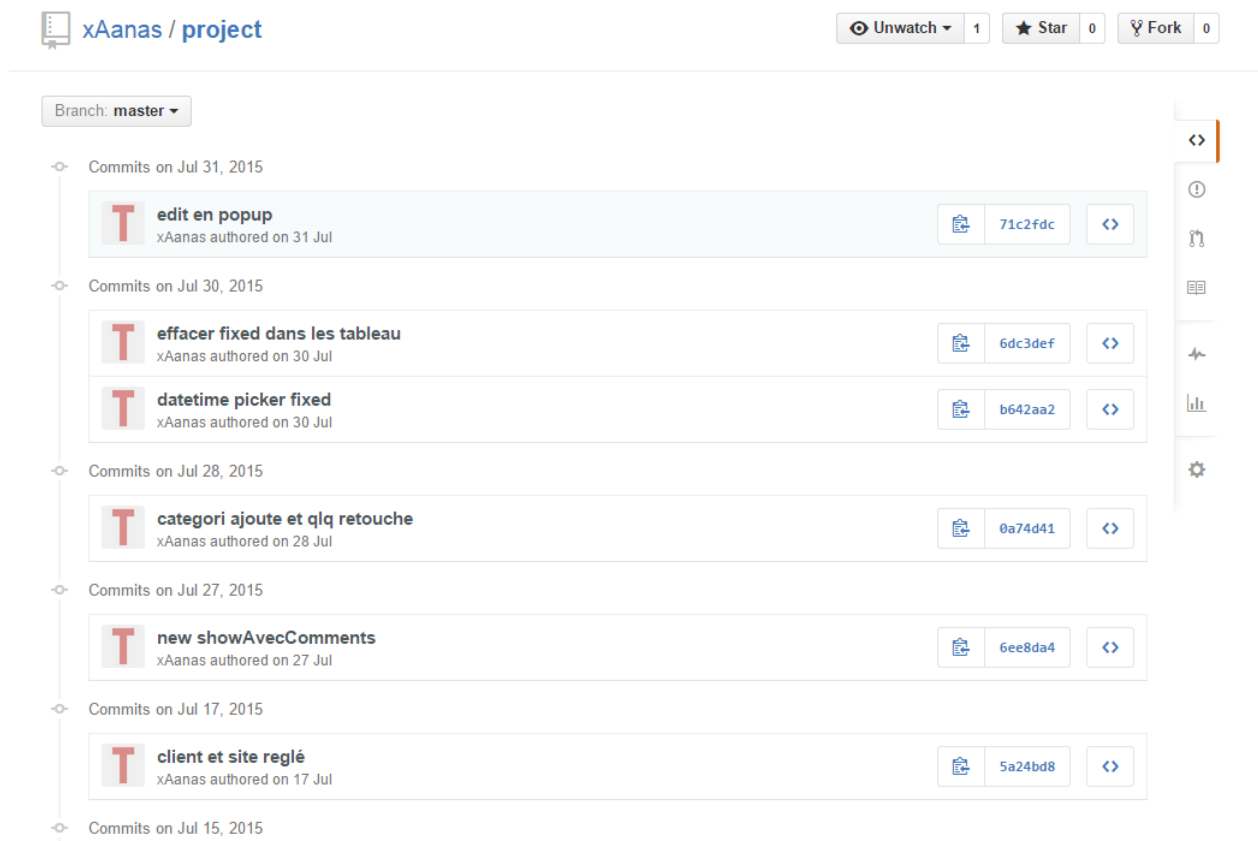


Figure IV.1 – Commits pendant le mois de Juillet

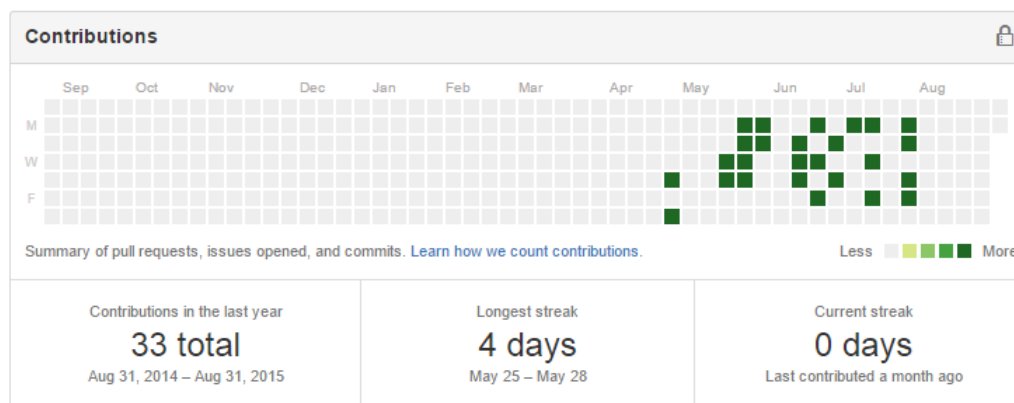


Figure IV.2 – Toutes les contributions

plication Web mais après l'élaboration du cahier de charge et vue le volume du travail demandé et la nature des fonctionnalités, il serait impossible de terminer le projet dans les délais. Le recourt à un Framework PHP devient indispensable. Ses bénéfices ne se limitent pas au gain énorme du temps que nous aurions mais aussi de la structuration

du projet. Nous aurons un cadre de travail conforme aux règles du métier qui nous fournira un code maintenable, évolutif et des bibliothèques réutilisables facilitant plusieurs tâches. Il nous permet donc d'éviter les anti-pattern comme *don't repeat yourself* (DRY), qui consiste à ne pas réinventer la roue et mettre en oeuvre des composants existants. Parmi les Framework gratuits sur le marché nous avons identifié les trois les plus utilisées et qui sont Laravel, Zend2 (ZF2) et Symfony2 (SF2). Vu que Laravel est encore nouveau et immature, le choix s'est limité entre ZF2 et SF2. Ces deux Framework sont matures et stables. Ils adaptent une architecture similaire, la seule différence frappante entre ses deux leaders est la richesse de la documentation qui a joué en faveur du SF2 et surtout sa communauté très active [16]. En effet, avec le Cookbook symfony2 [17] et la grande variété des tutoriels ça devient facile de gérer ses erreurs et répondre à nos besoins. D'autre part, SF2 est doté de plusieurs outils tels que Doctrine2 qui est l'ORM de SF2, il offre plusieurs fonctionnalités facilitant la manipulation de la base de données et communique directement avec la DBAL [12]. Le moteur de template TWIG qui nous garantit un code net et propre séparant le code php du code html et Javascript.

- **Composer.**

C'est le gestionnaire de dépendance dédié au PHP. Il facilite l'import des bundles afin de les réutiliser et permet également de maintenir notre application à jour à travers un simple update (semblable au MAVEN pour le langage Java).

- **Bundles.**

Ce sont des modules développés par la communauté Symfony pour éviter de réinventer la roue. En effet, leur intégration est facile grâce au Composer que nous venons de définir dans le paragraphe précédant et permettent de gagner énormément de temps [18].

- **Highcharts.**

Pour un rendu simple et clair des données nous avons eu recours à Highcharts qui est une bibliothèque Javascript spécialisée dans la représentation graphique. Ayant la possibilité de consommer du JSON et d'exporter des schémas sous format PDF ou même en image, cette bibliothèque a pris le dessus sur tous ses concurrents, e.g. Charts.js ou encore D3.js qui n'assurent pas l'exportation des données [19].

– **Twitter Bootstrap.**

Nous avons utilisé ce Framework CSS afin de garantir des vues conviviales et homogènes. En effet, il propose une collection d'outils HTML et des extensions Javascript permettant le développement des sites et des applications web. Ses principaux avantages sont :

- La réactivité (responsive) : les vues créées s'adaptent automatiquement avec différents types d'écran, même les écrans des appareils mobiles.
- La facilité et la rapidité du développement : Bootstraps offre des blocs html, css et js tout prêts dont le seul but est d'éviter de tout coder à partir de zéro, e.g. des formulaires, des barres de navigation, des boutons [20].

Enfin, Twitter Bootstrap s'est doté d'une grande communauté très active, le projet est partagé sur github et il est continuellement mis à jour selon les nouveautés du monde du web [21].

– **jQuery UI.**

C'est une collection de widgets, des interfaces utilisateur graphique (GUI) élégantes permettant de développer des applications web hautement interactives. Ces composants sont mis en place avec jQuery (bibliothèque Javascript), html et css. Une implémentation facile et rapide, c'est le but ultime de la bibliothèque jQuery, en effet, en l'utilisant nous gagnons un nombre très important de lignes de codes avec une meilleure lisibilité. Grâce à sa communauté très active, il est maintenu à jour pour rester parmi les leaders dans le développement web, chaque jour de nouvelles APIs et plugins sont injectées offrant une grande diversité de choix [22].

– **AJAX (Asynchronous JavaScript and XML).**

Cette technologie basée principalement sur du javascript est la combinaison de plusieurs langages (e.g., CSS, JSON, XML, XMLHttpRequest) afin d'effectuer des traitements complexes sur les vues sans recharger la totalité de la page, autrement dit, sans que l'utilisateur ne s'en aperçoive. En outre, Il permet d'avoir des applications plus réactives et d'implémenter de nouvelles fonctionnalités avec une présentation moderne tout en diminuant le temps de latence. AJAX a été notre choix devant plusieurs autres Frameworks offrant le service single-page comme l'AngularJS [23], parce qu'il est intégré par défaut

au SF2 et est facile à mettre en place.

La diversité de ces technologies nous a imposé une cartographie présentée par la figure IV.3 qui illustre leur disposition dans l'application.

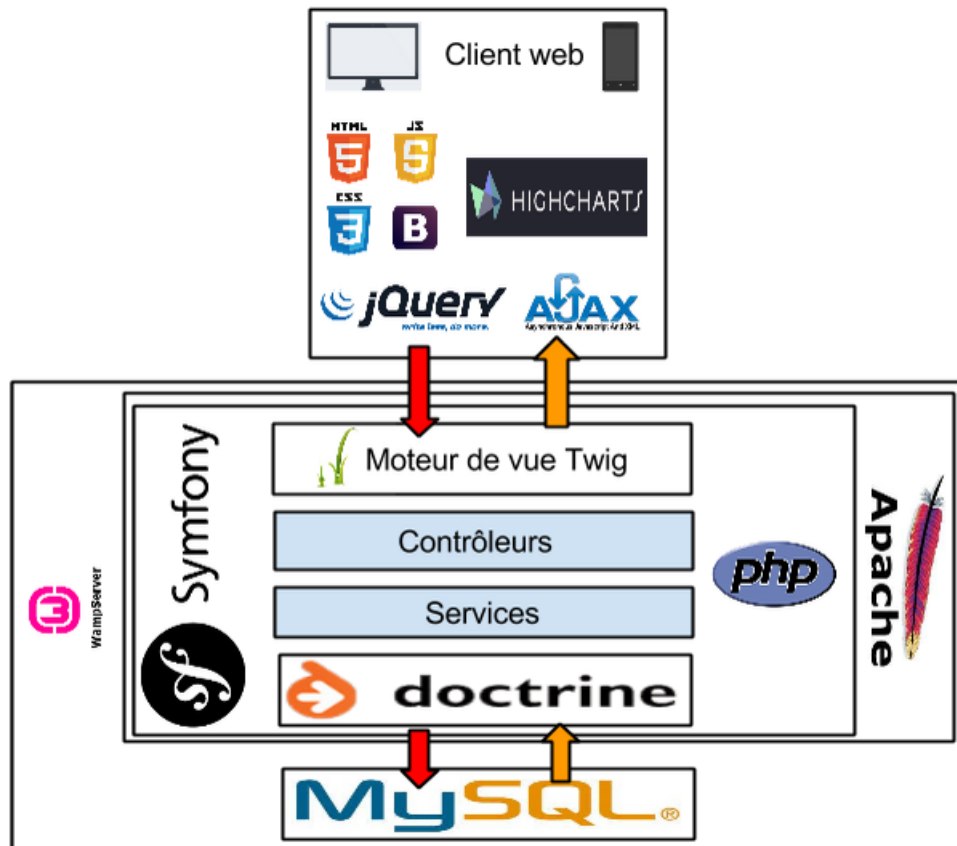


Figure IV.3 – Disposition des Frameworks

2 Présentation de l'application

Dans cette Dernière partie, nous allons présenter à travers quelques captures d'écran, les principales fonctionnalités de notre application.

● Authentification

Cette interface est le point d'entrée de notre application. Pour s'authentifier, il existe deux moyens, soit à travers un nom d'utilisateur et un mot de passe, soit à travers une

IV.2 Présentation de l'application

adresse mail. La figure IV.4 illustre l'interface d'authentification.

Cantara

Urbaprod
Version : V.0.1 - Cantara du : 15.07.2015

Indentifiant :

Mot de passe :

[Mot de passe oublié?](#)

☐ Garder ma session active

ATTENTION : Vous tentez d'accéder à une application propriétaire strictement réservée aux utilisateurs expressément autorisés par le propriétaire. Tous les accès et tentatives d'accès à ce site sont détectés. Tout accès ou tentative d'accès non autorisé à cette application constitue conformément à l'article 323-1 du Code pénal, une atteinte à un système de traitement automatisé de données et est passible de 2 ans d'emprisonnement et de 30 000 euros d'amende.

Copyright 2012-15 - Génie Systèmes

allouze+partners
s y s t e m s

Figure IV.4 – Interface d'authentification

En cas d'échec, une erreur s'affiche pour indiquer la cause sinon l'utilisateur est renvoyé directement vers l'interface d'accueil. Ce processus comporte deux phases de sécurité, la première est l'authentification, c'est à dire la vérification des données saisies par l'utilisateur afin de lui donner l'accès à notre application. La deuxième est l'autorisation, elle consiste à vérifier les droits de cet utilisateur pour accéder à la ressource demandée. S'il ne possède pas les droits requis, il sera redirigé vers la page d'accueil, si un utilisateur non connecté essaie d'accéder à une ressource, il sera automatiquement redirigé vers la page d'authentification. La figure IV.5 met en évidence la différence entre un rôle collaborateur et un rôle administrateur. Afin d'implémenter cette phase, nous avons procédé à la sécurisation des URLs.

● Interface d'accueil

C'est l'interface principale de notre application, elle se présente sous la forme d'une file d'actualités composée de plusieurs éléments simples et faciles à utiliser. En mettant en place des icônes fréquemment utilisées dans les réseaux sociaux et dans les applications web

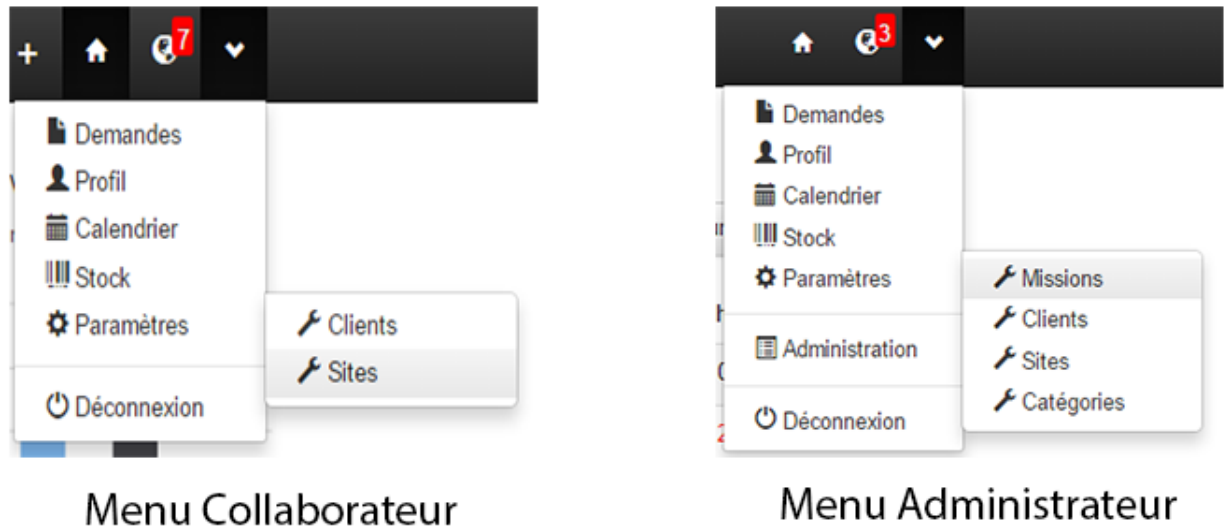


Figure IV.5 – Différence entre deux rôles

ainsi que des interfaces ludiques avec des emplacement bien spécifique, nous rendons les vues plus manipulables pour un débutant permettant de s'y habitué rapidement. La figure IV.6 représente notre écran d'accueil.

Par exemple, pour changer l'état de la demande, il suffit d'appuyer sur l'état actuel de la demande, un "pop-up" apparaît avec un formulaire pour contrôler l'état, et enfin le changement s'effectue sans rechargement de la page. La figure IV.7 met en évidence les étapes citées.

Le graphe à droite de l'écran montre les demandes traitées pendant les 3 derniers mois, en choisissant le mois, on aura l'accès au états des demandes en détails i.e. émise, en cours, annulée et livrée.

Les composants principaux de cette file d'actualité sont les demandes. Les collaborateurs en France émettent une demande à travers un pop-up dans l'accueil.

La catégorie d'une demande est par défaut de type Génie des lieux (GDL), seul l'administrateur peut changer la catégorie. Au cour de la création d'une demande, l'utilisateur, à travers des pop-up, peut ajouter au fur et à mesure des clients et leurs affecter des sites sans besoins de recharger le formulaire de la demande illustré par la figure IV.10.

IV.2 Présentation de l'application

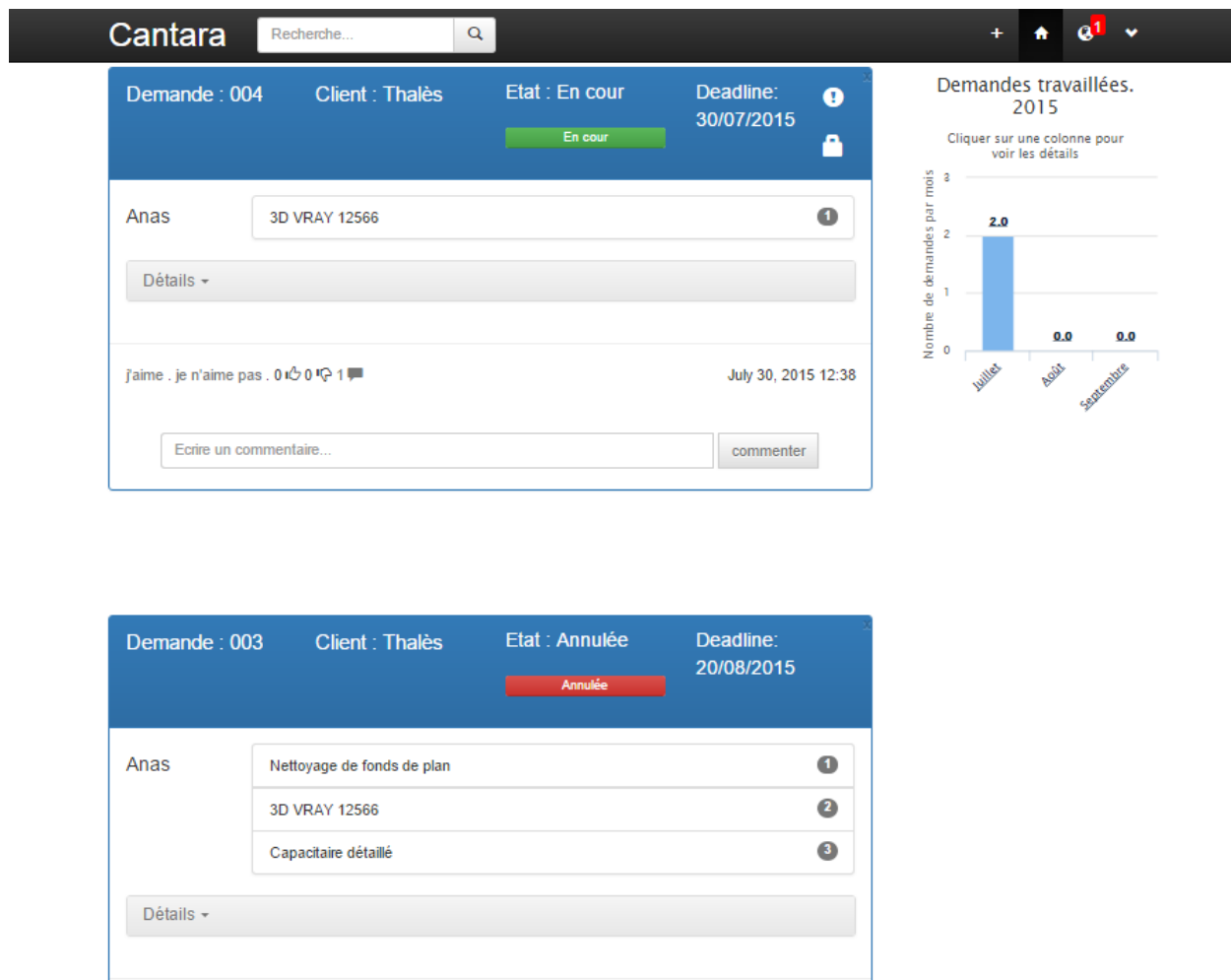


Figure IV.6 – Vue complète de l'interface d'accueil

La demande est émise dans la file d'actualité comme introduit par la Figure IV.11.

Une fois la demande émise, les collaborateurs à Tunis peuvent l'aimer, la commenter, changer son état, télécharger les fichiers joints etc. Dans la Figure IV.12 nous illustrons une demande après plusieurs traitements.

● Interface écran commun

Elle est illustré par la figure IV.13. C'est une interface affichée dans un grand écran devant tous les collaborateurs. Elle donne des informations spécifiques afin d'aider dans le suivi des demandes comme l'arrivée d'un nouveau commentaire ou l'affectation d'un nouveau fichier.

Cette interface permet à l'administrateur de créer une nouvelle demande ainsi que d'ex-

IV.2 Présentation de l'application



Figure IV.7 – Changer l'état d'une demande

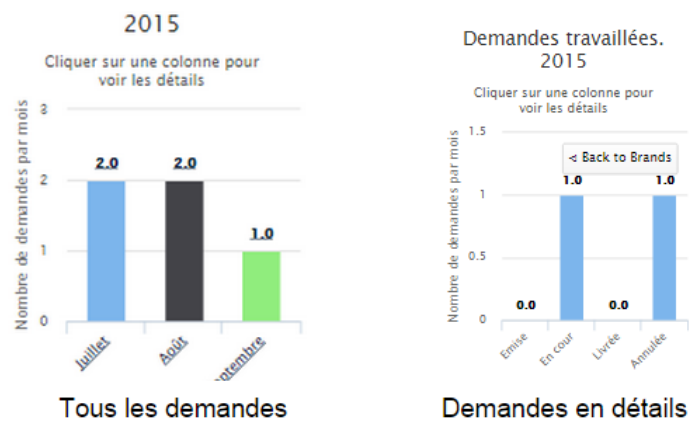


Figure IV.8 – Le graphe des demandes

porter la liste de toutes les demandes dans le système.

- Administration des utilisateurs.

Elle permet à l'administrateur toutes les manipulations sur les utilisateurs comme la mise à jour, l'activation ou l'affectation d'un nouveau rôle. Elle est représenté par la figure IV.14.

- Administration des demandes

IV.2 Présentation de l'application

Nouvelle demande

Catégorie : Choisir une categorie

Potentiel facturation : Choisir un potentiel

Chef de projet : Choisir un chef de projet

Demande au nom de : Anas

Client : Choisir un client

Site : Choisir un site

Mission 1 : Choisir une mission type

Mission 2 : Choisir une mission type

Mission 3 : Choisir une mission type

Autres :

Détails mission 1 :

Détails mission 2 :

Détails mission 3 :

Date limite : 2015/09/05 18:00

Liens :

Niveau d'urgence :

Confidentialité :

Documents à envoyer par GDL :

Envoie prévu le :

Mettre en copie :

Quitter Créer

Figure IV.9 – Formulaire de la demande (vue administrateur)

Nouveau site

Client : Verneg

Nom :

Adresse :

Description :

Quitter Créer

Figure IV.10 – Pop-up affectation d'un site

Elle garantit l'accès direct aux données des demandes afin de les modifier à volonté ou les exporter sous plusieurs formats i.e. json, xml, csv.

Cette interface montrée par la figure IV.15, assure aussi la filtration des demandes et la recherche des informations selon un ou plusieurs critères en même temps.

IV.2 Présentation de l'application

Demande : 005

Client : Thalès

Etat : Emise

Deadline: 05/09/2015

Emise

Anas

Nettoyage de fonds de plan

3D SPACIO

Capacitaire détaillé

Détails ▾

Détails mission 1 : nettoyer tous
Détails mission 2 : le spacio interne
Détails mission 3 : ne pas trop détailler
Autres : sans calcul svp

j'aime . je n'aime pas . 0 0 0 0

September 5, 2015 15:07

Ecrire un commentaire...

commenter

Figure IV.11 – Demande émise

Conclusion

A travers ce chapitre, nous avons mis en exergue les aspects de la réalisation de notre travail. Nous avons justifié le choix de la technologie adaptée et nous avons présenté notre environnement de travail. Enfin, nous avons concrétisé le tout avec quelques capture d'écran de notre application.

IV.2 Présentation de l'application

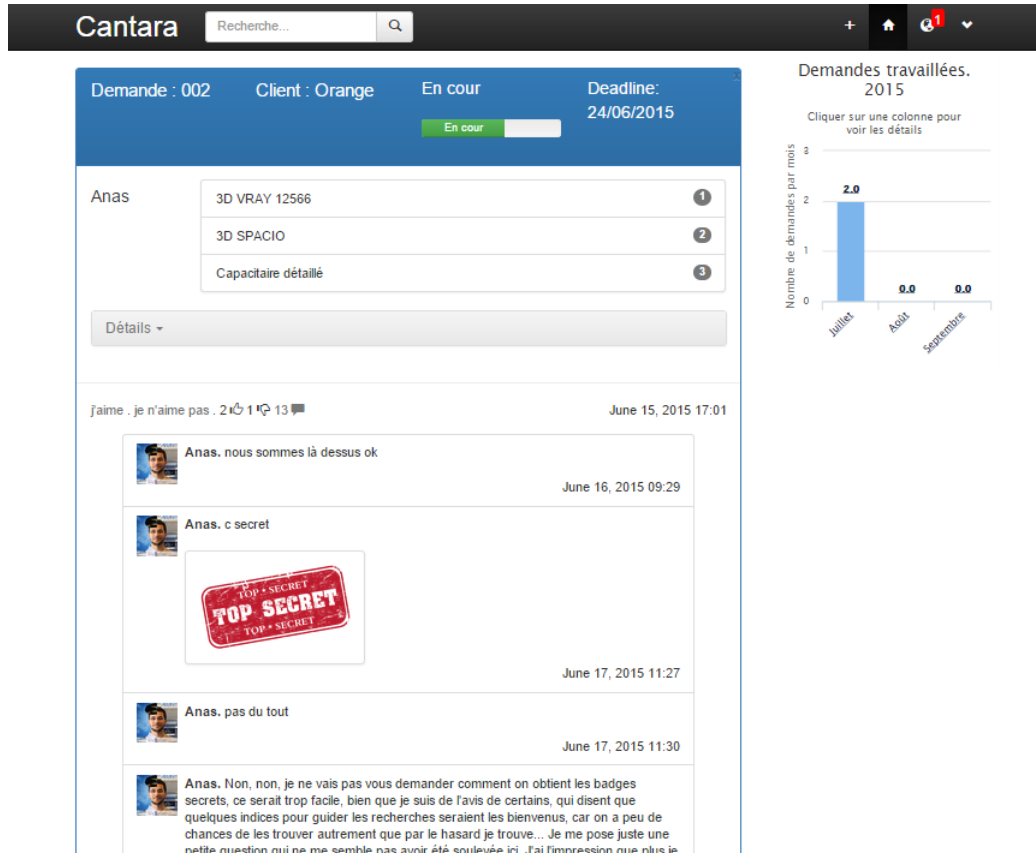


Figure IV.12 – Demande après traitement

Demandes

📄 télécharger tous ➕ Créer une nouvelle demande

N °	Demandeur	Client	Site	Date limite	Urgence	Etat	Commentaires	Fichiers	Actions
5	Anas	Thalès	site 3	2015-09-05 18:00	⚠️	Emise	0	0	✎️ 🗑️ 📄
4	Aymen	Thalès	site Alsas	2015-07-30 16:00	⚠️	En cour	1	2	✎️ 🗑️ 📄
3	Hamza	Thalès	site Alsas	2015-08-20 00:00	⚠️	Annulée	0	0	✎️ 🗑️ 📄
2	Hamza	Orange	site 51	2015-06-24 00:00	⚠️	En cour	0	0	✎️ 🗑️ 📄
1	Hamza	Orange	site 51	2015-06-24 00:00	⚠️	Livrée	2	0	✎️ 🗑️ 📄

© Copyright 2015 - Ellouze and partners system

Figure IV.13 – Interface écran commun

IV.2 Présentation de l'application

<input type="checkbox"/>	Id	Username	Prenom	Nom	Email	Enabled	Roles
<input type="checkbox"/>	1	Anas	anas	ben haj ali	anasbena07@gmail.com	<input checked="" type="checkbox"/>	[0 => ROLE_ADMIN] [1 => ROLE_USER]
<input type="checkbox"/>	2	Hamza	Hamza	Beyni	hamza.beyni@gmail.com	<input checked="" type="checkbox"/>	[0 => ROLE_USER]
<input type="checkbox"/>	3	Aymen	Ayme	ELLOUZE	aymen@gmail.com	<input checked="" type="checkbox"/>	[0 => ROLE_ADMIN] [1 => ROLE_USER]
<input type="checkbox"/>	4	mariem	Mariem	Nfaiedh	mariem@gmail.com	<input checked="" type="checkbox"/>	[0 => ROLE_USER]

☐ Tous les éléments (4)

 - 1 / 1 - 4 résultats - Par page 25

Figure IV.14 – Interface administration des utilisateurs

<input type="checkbox"/>	Id	Utilisateur	Chef De Projet	Au Nom De	Client	Sites	Mission One	Mission Two	Mission Three	Date Limite	Jaime	Je Naime Pas	Niveau Urgence	Etat	Confident
<input type="checkbox"/>	1	Anas		Hamza	Thalès	site 51	3D VRAY 12566	3D SPACIO	Capacitaire détaillé	2015-06-24 00:00:00	1	1	critique	Livrée	Haute
<input type="checkbox"/>	2	Anas		Hamza	Orange	site 51	3D VRAY 12566	3D SPACIO	Capacitaire détaillé	2015-06-24 00:00:00	2	1	ordinaire	En cour	Normale
<input type="checkbox"/>	3	Anas		Hamza		site Alsas	Nettoyage de fonds de plan	3D VRAY 12566	Capacitaire détaillé	2015-08-20 00:00:00	1	0	ordinaire	Annulée	Normale
<input type="checkbox"/>	4	Anas	mariem	Aymen	Thalès	site Alsas	3D VRAY 12566			2015/07/30 16:00	0	0	urgente	En cour	Haute
<input type="checkbox"/>	5	Anas	Hamza	Anas	Thalès	site 3	Nettoyage de fonds de plan	3D SPACIO	Capacitaire détaillé	2015/09/05 18:00	0	0	urgente	Emise	Haute

☐ Tous les éléments (5)

Filtres

Id

Utilisateur

Chef De Projet

Au Nom De

Client

Sites

Mission One

Figure IV.15 – Interface administration des demandes

Conclusion Générale et Perspectives

Dans une entreprise, la gestion des commandes reçues est une étape primordiale dans le processus, de conception et/ou de production. C'est l'étape au cours de laquelle l'écoute du client est importante afin de bien comprendre ses besoins et de tenir compte de ses envies. Grâce à l'informatique, nous avons pu répondre à cette problématique en utilisant le Customer Relationship Manager (CRM) qui est défini comme étant l'ensemble des outils et techniques permettant de traiter et d'analyser toutes les informations relatives aux clients dans le but de les fidéliser en leur offrant les meilleurs services

Mais, comme toute solution, le CRM souffre d'un ensemble d'inconvénients. Parmi eux, surgit le problème de la facilité d'utilisation surtout pour les non-connaisseurs, comme les architectes. Nous avons donc pensé à une adaptation du concept du CRM aux besoins d'un architecte qui est une personne peu connaisseuse en informatique. Dans ce cadre, nous avons effectué ce stage de fin d'étude dans la société « UrbaProd ». Nous étions chargé de concevoir une plateforme de CRM qui permet, à la fois, de gérer les demandes des clients en un laps de temps raisonnable, et de sauvegarder l'historique des différentes demandes passées par le client. Notre plate-forme devra en plus être facile à utiliser par tous les utilisateurs, que ce soient des clients ou des architectes. Afin d'atteindre ces objectifs, nous avons utilisé le "Framework" Symfony2. Il assure une grande performance et une facilité d'extensibilité, nous avons exploité les composantes qu'il offre comme la gestion de sécurité pour l'authentification et les droits d'accès. A la fin de ce travail, nous avons répondu aux besoins de la société à travers l'ensemble des fonctionnalités fournies par notre solution, à savoir, le module de gestion des demandes, le module de notification et le module de gestion des clients. Mais cela n'empêche que nous avons connu quelques difficultés pendant la période de la collecte des besoins et des difficultés pendant la période de l'implémentation de la solution puisque l'entreprise ne comprend pas dans son effectif des spécialistes dans ce domaine. L'utilisation de Symfony2 facilitera ensuite l'intégration des améliorations envisagés par la société comme la notification par sms afin de garantir la visibilité de l'information par tous les collaborateurs. Même si nous avons utilisé des interfaces responsives, nous avons proposé d'implémenter une application mobile qui sera plus adéquate pour les smartphones.

Bibliographie

- [1] <http://www.everwin.fr/solutions>. [En ligne ; consulté le 21-Juin-2015]. v, 5
- [2] <http://www.dolibarr.fr/>. [En ligne ; consulté le 22-Juin-2015]. v, 6, 7
- [3] NEILPITMAN DAN PILONE. *UML 2.0 in Nutshell*. (publié le Juin 2005). 1
- [4] <http://www.fitnetmanager.com/crm-avant-vente>. [En ligne ; consulté le 21-Juin-2015]. 5
- [5] <http://www.agilemanifesto.org/iso/en/principles.html>. [En ligne ; consulté le 14-Septembre-2015]. 8
- [6] PHILLIPE KRUCHTEN. *The Rational Unified Process : An Introduction*. (publié le 10 Décembre 2003). 8
- [7] KENT BECK. *Extreme Programming Explained : Embrace change*. (publié le 26 novembre 2014). 8
- [8] KEN SCHWABER ET JEFF SUTHERLAND. *Le Guide SCRUM*. (publié en juillet 2013). 8, 9, 13
- [9] JAMES MARTIN. *Rapid Application Development*. (1991). 8
- [10] https://stefanini.com/FR/2014/03/les_avantages_de_la_methode_scrum/. [En ligne ; consulté le 24-Juin-2015]. 9
- [11] <http://blocnotes.iergo.fr/concevoir/les-criteres-heuristiques-de-bastien>. [En ligne ; consulté le 14-Septembre-2015]. 13
- [12] <https://packagist.org/>. [En ligne ; consulté le 13-Septembre-2015]. 21, 42
- [13] [https://fr.wikipedia.org/wiki/Paquet_\(logiciel\)](https://fr.wikipedia.org/wiki/Paquet_(logiciel)). [En ligne ; consulté le 13-Septembre-2015]. 25
- [14] <https://netbeans.org/>. [En ligne ; consulté le 31-Août-2015]. 25
- [15] <http://blog.nicolashachet.com/technologies/php/rex-zend-2-vs-symfony-2/>. [En ligne ; consulté le 09-Septembre-2015]. 39
- [16] <https://fr.wikipedia.org/wiki/Modèle-vue-contrôleur>. [En ligne ; consulté le 31-Août-2015]. 42
- [17] <http://symfony.com/fr/doc/current/cookbook/index.html>. [En ligne ; consulté le 31-Août-2015]. 42

- [18] <http://symfony.com/fr/doc/current/cookbook/doctrine/dbal.html>.
[En ligne; L'un des sites les plus connues pour l'intégration des bundles, consulté le 01-Septembre-2015]. 42
- [19] <http://www.highcharts.com/>. [En ligne; consulté le 01-Septembre-2015]. 42
- [20] <http://getbootstrap.com/>. [En ligne; consulté le 01-Septembre-2015]. 43
- [21] <https://github.com/twbs/bootstrap>. [En ligne; consulté le 01-Septembre-2015].
43
- [22] <http://jqueryui.com/>. [En ligne; consulté le 01-Septembre-2015]. 43
- [23] <https://angularjs.org/>. [En ligne; consulté le 13-Septembre-2015]. 43

Annexe : Remarques Diverses

A renseigner par Génie des Lieux			
Demande à envoyer	tunis@ellouzeandpartners.com		
Mettre en copie :	e.fioroni@geniedeslieux.fr		
Client :	VALOPHIS		
Site :			
Chef de projet GDL :	Gérard Pinot		
Expéditeur de la demande	LAUREN GROLET		
I - Demande initiale			
Mission :	N°	Nettoyage de fonds de plan	
		3D SPACIO	
		3D VRAY (3D calculée)	
		Capacitaire détaillé	
		Capacitaire général	
		Implantation mobilier	
	Autres :	Mise à la charte	
Détails du travail à réaliser :	N°1 :	Nettoyage de fonds de plan + mise à la charte	
	N°2 :		
	N°3 :		
	N°4 :		
	N°5 :		
Délai de rendu :	Fin de semaine		

Figure A.1 – Demande sous sa forme originale
