



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

Remerciements

Merci à tous ! Bonne journée

Table des Matières

Liste des Figures	iv
Liste des Tableaux	v
Résumé	vi
Abstract	vii
Introduction Générale	1
I Cadre du projet	2
1 Cadre du projet	2
2 Problématique	2
3 Solutions existantes	3
3.1 Version éditeur	3
3.2 Version libre	4
3.3 Etude comparative	4
4 Objectif du projet	6
5 Contexte méthodologique du projet	6
5.1 Le choix de la Méthode Scrum	6
5.2 Les rôles et les notions	7
II Capture,analyse et spécification des besoins	8
1 Étude de l'existant	8
2 Critique de l'existant	9
3 Analyse des besoins	9
3.1 Les acteurs	9
3.2 Les besoins fonctionnels	9
3.3 Les besoins non fonctionnels	10
3.4 Planification du projet	10
3.4.1 Acteurs du projet	10
3.4.2 Backlog produit	10
3.4.3 Les sprints du projet	11
3.5 Spécification des besoins diagramme de cas d'utilisation global	11

III Etude conceptuelle	15
1 Architecture physique	15
2 Architecture logicielle	16
3 Raffinement des diagrammes des cas d'utilisations	18
3.1 module gestion de demande	18
3.2 module gestion des utilisateurs	18
3.3 Module gestion des paramètres de la demandes.	19
3.4 Module exportation des données	19
4 Diagrammes structurels	21
4.1 Diagramme de paquetages	21
4.2 Diagramme de classes	22
5 Diagramme comportementaux	23
5.1 Diagrammes de séquences	23
5.2 Diagramme d'activité	26
IV Réalisation	35
1 Environnements de travail et choix techniques	35
1.1 Environnements de travail	35
1.2 Technologies	37
2 Présentation de l'application	40
Conclusion Générale et Perspectives	44
Bibliographique	45
Annexe : Remarques Diverses	47

Liste des Figures

I.1	État de l'art	3
I.2	Fonctionnalités offertes par Everwin CXM [1]	4
I.3	architecture Dolibarr [2]	5
I.4	Ecran d'accueil de Dolibarr [2]	5
II.1	Diagramme de cas d'utilisation global	11
III.1	Architecture physique	16
III.2	Architecture applicative de la solution	17
III.3	Diagramme de cas d'utilisation Gestion d'une demande	19
III.4	Diagramme de cas d'utilisation gestion utilisateurs	22
III.5	Diagramme d'utilisation du paramétrage d'une demande	27
III.6	Diagramme de cas d'utilisation exportation des données	28
III.7	Diagramme de packaging	29
III.8	Le modèle du domaine	30
III.9	Le modèle du domaine	31
III.10	Diagramme de séquence ajout d'une demande	32
III.11	Diagramme de séquence suppression d'un site	33
III.12	Diagramme d'activité : envoi d'une notification	34
IV.1	Commits pendant le mois de Juillet	36
IV.2	Toutes les contributions	37
IV.3	Disposition des Frameworks	39
IV.4	Interface d'authentification	40
IV.5	Différence entre deux rôles	41
IV.6	Vue complète de l'interface d'accueil	42
IV.7	Changer l'état d'une demande	43
IV.8	Le graphe des demandes	43

Liste des Tableaux

I.1	Comparatif entre les solutions existantes	6
II.1	Les acteurs du SCRUM	10
II.2	Backlog du produit	13
II.3	Les sprints du projet	14
III.1	Ajout d'une demande	20
III.2	Ajouter un nouveau commentaire	21
III.3	Ajout d'un utilisateur	23
III.4	Ajout d'un nouveau site	24
III.5	Exporter la liste des demandes	25
III.6	Descriptif des entités de l'application	25

Résumé

Ceci est le résumé en français de votre projet. Il devra être plus détaillé que le résumé se trouvant dans le verso de votre rapport.

Abstract

This is the english abstract of your project. It must be longer and presented in more details than the abstract you write on the back of your report.

Introduction Générale

Pour écrire un bon rapport [3] de projet en informatique, il existe certaines règles à respecter. Certes, chacun écrit son rapport avec sa propre plume et sa propre signature, mais certaines règles restent universelles [4].

La Table de matière est la première chose qu'un rapporteur va lire. Il faut qu'elle soit :

- Assez détaillée¹. En général, 3 niveaux de numéros suffisent ;
- Votre rapport doit être réparti en chapitres équilibrés, à part l'introduction et la conclusion, naturellement plus courts que les autres ;
- Vos titres doivent être suffisamment personnalisés pour donner une idée sur votre travail. Éviter le : « Conception », mais privilégier : « Conception de l'application de gestion des ... » Même s'ils vous paraissent longs, c'est mieux que d'avoir un sommaire impersonnel.

Une introduction doit être rédigée sous forme de paragraphes bien ficelés. Elle est normalement constituée de 4 grandes parties :

1. Le contexte de votre application : le domaine en général, par exemple le domaine du web, de BI, des logiciels de gestion ?
2. La problématique : quels sont les besoins qui, dans ce contexte là, nécessitent la réalisation de votre projet ?
3. La contribution : expliquer assez brièvement en quoi consiste votre application, sans entrer dans les détails de réalisation. Ne pas oublier qu'une introduction est censée introduire le travail, pas le résumer ;
4. La composition du rapport : les différents chapitres et leur composition. Il n'est pas nécessaire de numéroter ces parties, mais les mettre plutôt sous forme de paragraphes successifs bien liés.

1. Sans l'être trop

Chapitre I

Cadre du projet

Plan

1	Cadre du projet	2
2	Problématique	2
3	Solutions existantes	3
3.1	Version éditeur	3
3.2	Version libre	4
3.3	Etude comparative	4
4	Objectif du projet	6
5	Contexte méthodologique du projet	6
5.1	Le choix de la Méthode Scrum	6
5.2	Les rôles et les notions	7

Introduction

Une étude théorique [5] peut contenir l'une et/ou l'autre de ces deux parties :

1 Cadre du projet

C'est une étude assez détaillée sur ce qui existe sur le marché ou dans la littérature (d'où le terme état de l'art), qui permet de répondre à la problématique. L'idée ici est de faire un comparatif entre les solutions existantes, mais surtout d'analyser le résultat de cette comparaison et de dire pourquoi ne sont-elles pas satisfaisantes pour répondre à votre problématique.

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



Figure I.1 – État de l’art

partage d’informations, l’outil utilisé est le mail. Cependant, cela vire au désordre, i.e. messages dissparus, 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 côtes à côtes. Activées ensemble, elles fonctionnent de 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 [6].
- 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 EPEverwin SX et Everwin GX [1] . La figure I.2 illustre l'ensemble des fonctionnalités couvertes par Everwin CXM.



Figure I.2 – 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.3 illustre l'architecture sur laquelle est basée la solution Dolibarr :

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.4 présente l'écran d'accueil de Dolibarr :

3.3 Etude comparative

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

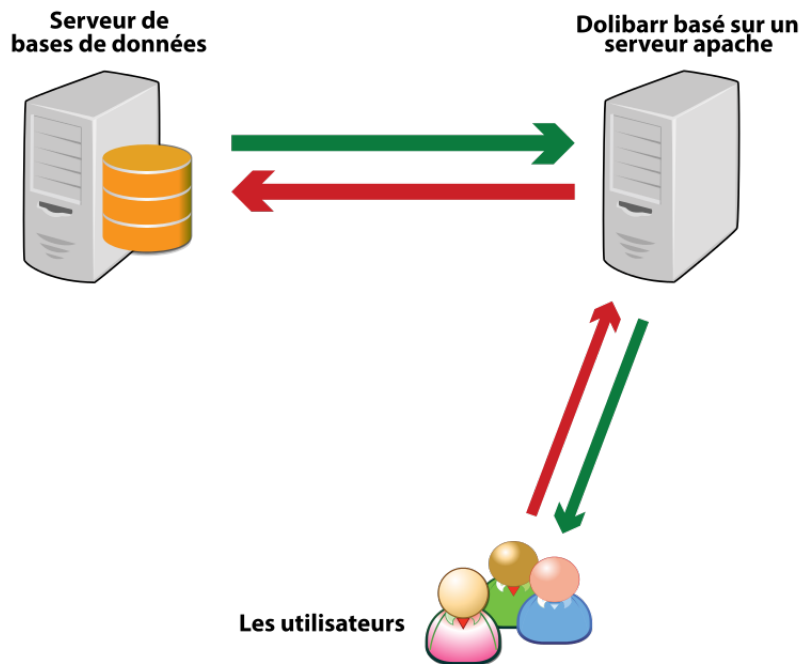


Figure I.3 – architecture Dolibarr [2]

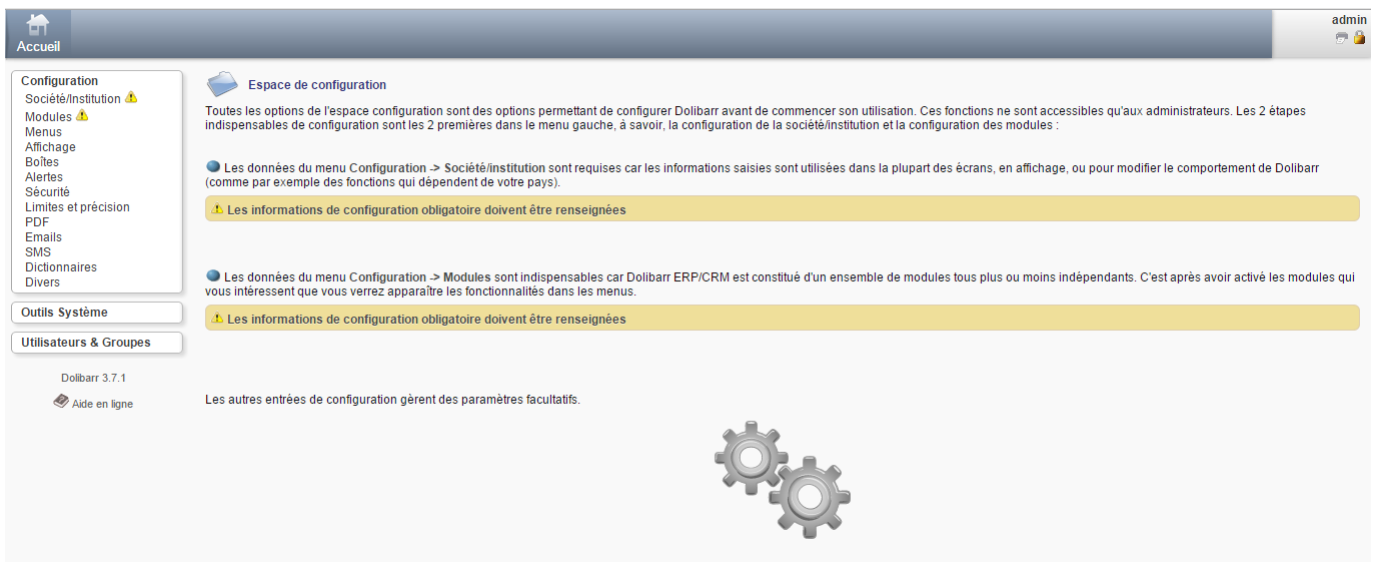


Figure I.4 – Ecran d'accueil de Dolibarr [2]

Les applications étudiées, qu'elles soient en versions éditeurs ou même gratuites, et malgré leurs adaptabilités et la diversité de leurs fonctionnalités offertes, restent incapables 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.

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

4 Objectif 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. 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 [7], la méthode XP [8], la méthode SCRUM [9] et la méthode RAD [10].

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 choisissons 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 brefs 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 membre 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 [11].

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-organisé.
- 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 cour [9].

Conclusion

Dans ce chapitre, nous avons présenté le cadre général du travail, en commençant par la présentation de l'entreprise Ellouze and Partners, 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	8
2	Critique de l'existant	9
3	Analyse des besoins	9
3.1	Les acteurs	9
3.2	Les besoins fonctionnels	9
3.3	Les besoins non fonctionnels	10
3.4	Planification du projet	10
3.5	Spécification des besoins diagramme de cas d'utilisation global	11

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 EllouzeAndPartners 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'EllouzeAndPartners répond à cette demande par un autre formulaire Excel prédéfini.
- Le suivi de la demande. Ca se fait à travers l'Email, Skype ou par téléphone.

2 Critique de l'existant

Il faut Bien choisir les diagrammes adéquats pour votre application. En général, les diagrammes obligatoires sont les diagrammes de cas d'utilisation, de classe et de séquence. Vous pouvez ajouter en plus le diagramme qui vous semble pertinent : par exemple, pour une application sur plusieurs tiers, il est intéressant de montrer le diagramme de déploiement ;

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, leur modification et leur suppression.
- La gestion des sites : l'application doit permettre l'ajout des sites relatifs à chaque client ainsi que leur modification et suppression.
- Extraction des données : toutes listes d'utilisateurs, clients, sites, demandes, tâches peuvent être extraites sous 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

des notifications aux utilisateurs à chaque événement important comme l'ajout ou la modification d'une demande ou l'ajout d'un nouveau commentaire.

3.4 Planification du projet

3.4.1 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 de travail. Il est utilisé essentiellement pour planifier les releases. À 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 annuler les idées non concluantes du départ [9]. 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 établis à l'aide du backlog produit tout en respectant la priorité des différents modules. Le tableau II.3 présente les sprints du projet :

À 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 :

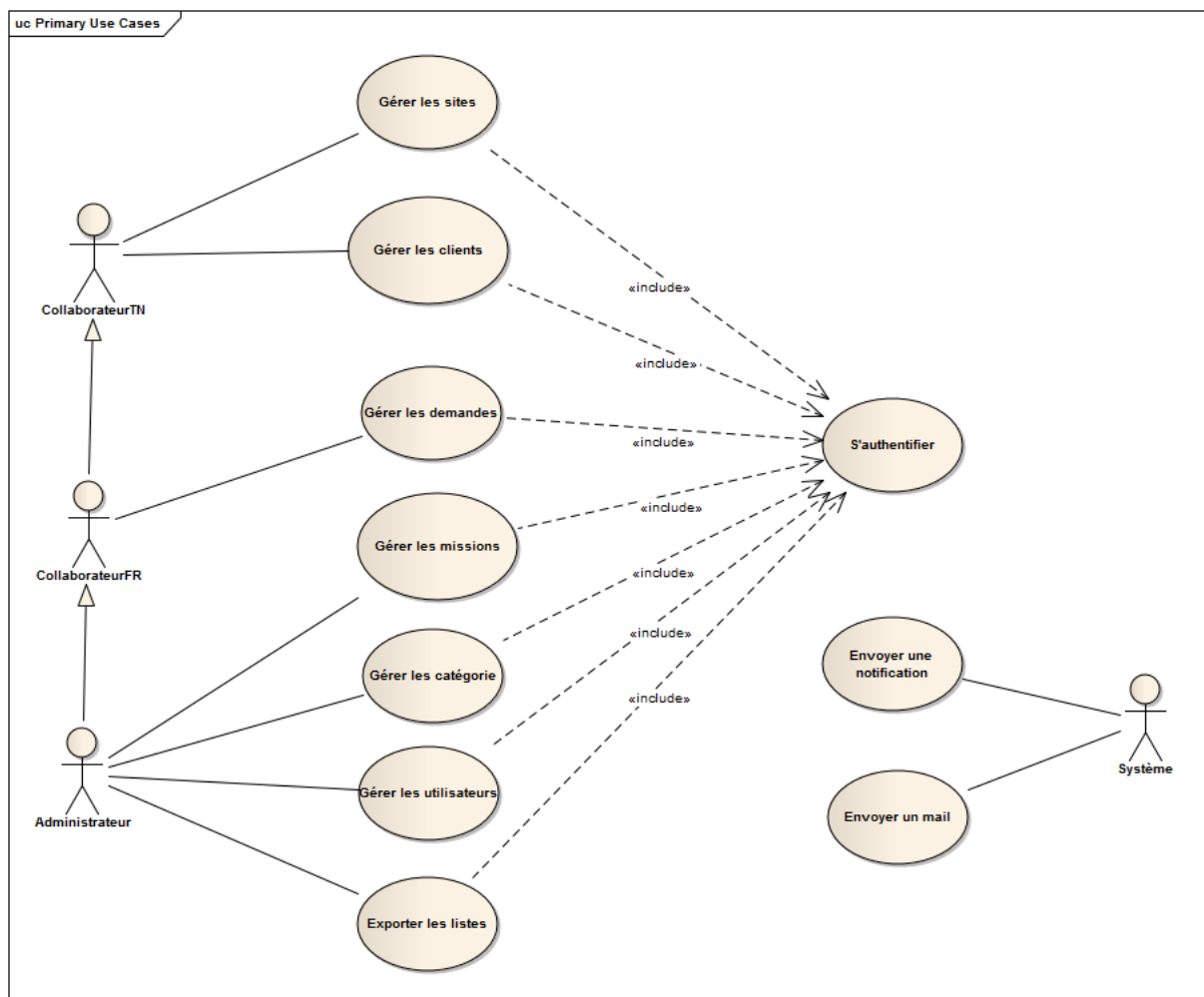


Figure II.1 – Diagramme de cas d'utilisation global

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.

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 ma 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 du 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	15
2	Architecture logicielle	16
3	Raffinement des diagrammes des cas d'utilisations	18
3.1	module gestion de demande	18
3.2	module gestion des utilisateurs	18
3.3	Module gestion des paramètres de la demandes.	19
3.4	Module exportation des données	19
4	Diagrammes structurels	21
4.1	Diagramme de paquetages	21
4.2	Diagramme de classes	22
5	Diagramme comportementaux	23
5.1	Diagrammes de séquences	23
5.2	Diagramme d'activité	26

Introduction

phrase introductive

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 module 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.

- Le client : ou la partie présentation qui est dans notre cas le navigateur web (google Chrome, Mozilla Firefox etc).

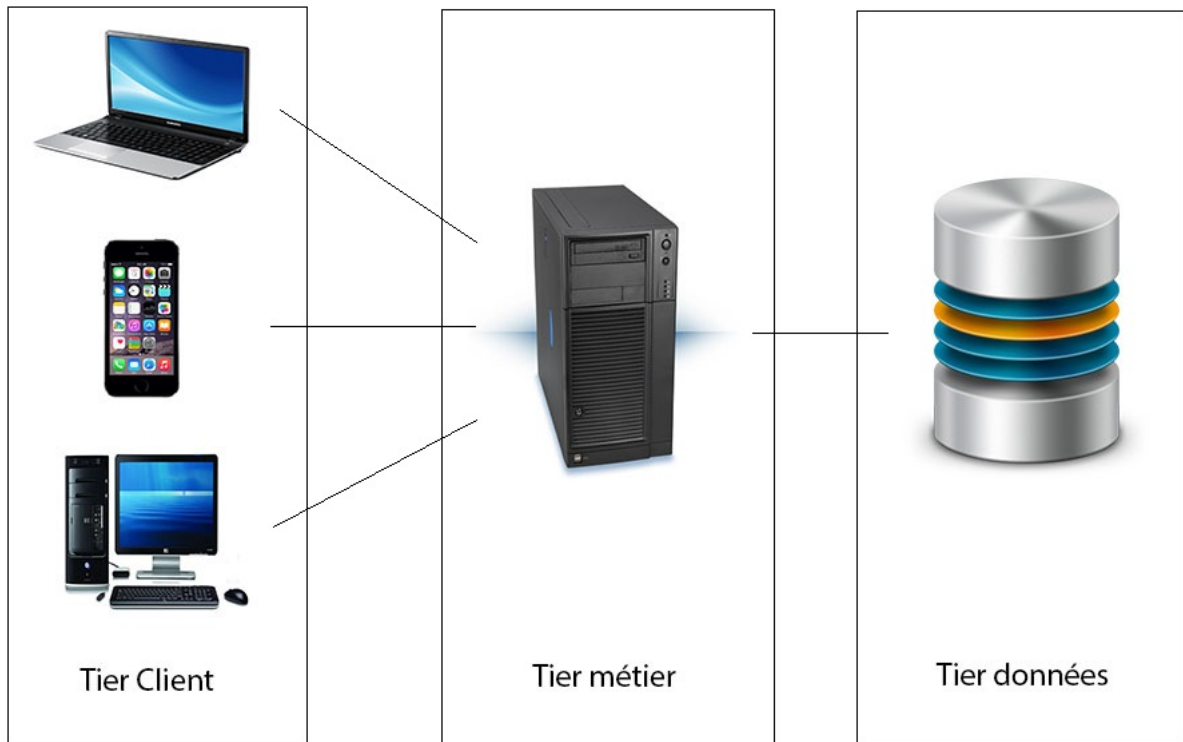


Figure III.1 – Architecture physique

- 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.

- 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,

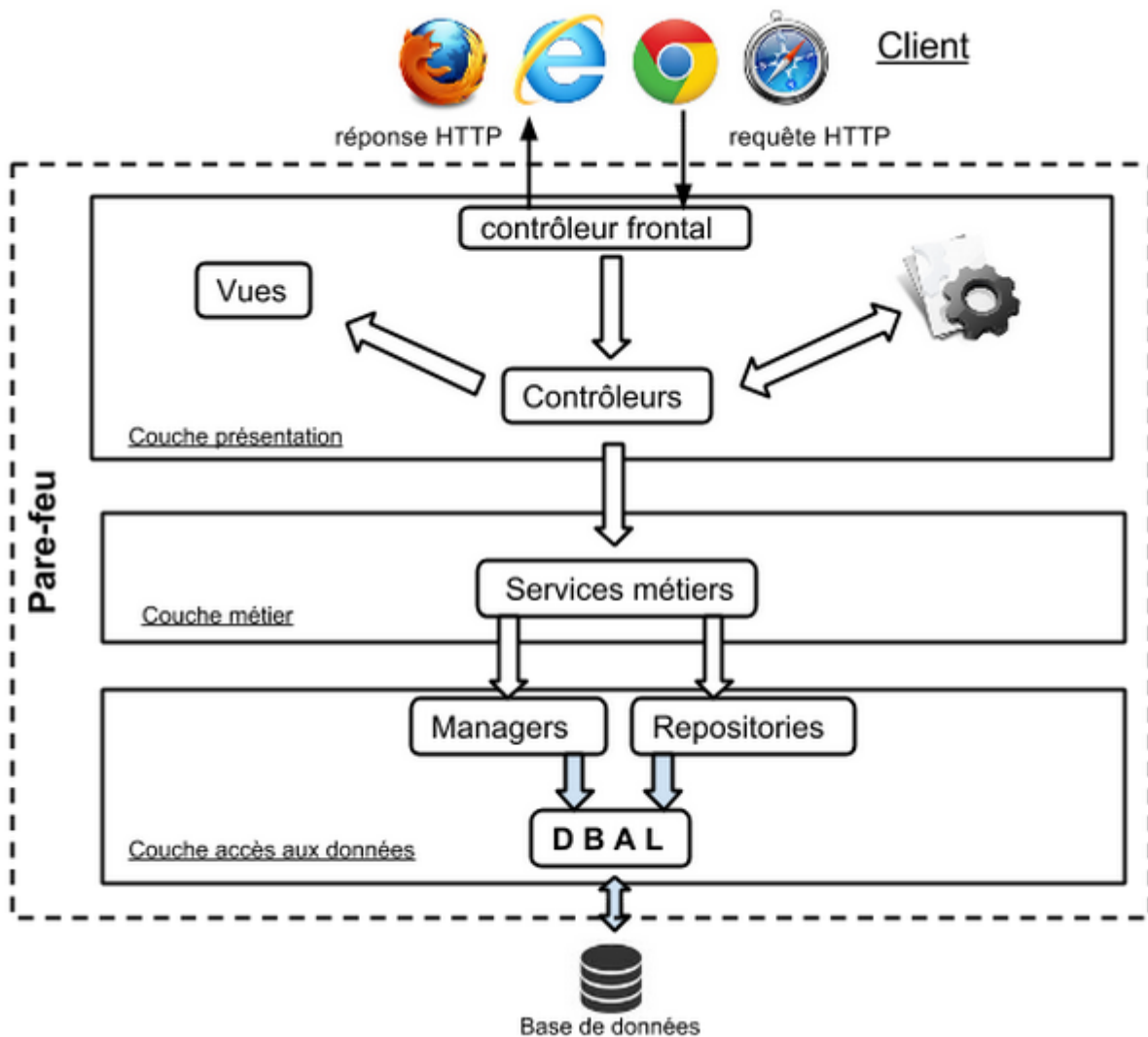


Figure III.2 – Architecture applicative de la solution

quel service sera invoqué et quelle vue sera rendu, 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 au 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 [W5]. 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 de demande

Ce module regroupe toutes les rubriques de gestion d'une demande. La création des demandes est conduite par les collaborateurs en France. Les collaborateurs à Tunis auront une visibilité sur les détails du travail à faire ainsi que sur les interactions qu'ils peuvent effectuer avec la demande. La figure [III.3](#) montre le diagramme de cas d'utilisation de gestion d'une demande.

- Description du cas d'utilisation : ajout d'une demande
Le tableau 3. 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](#).

3.2 module gestion des utilisateurs

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.

- Description du cas d'utilisation : ajout d'un utilisateur
Ce cas est décrit par le tableau [III.3](#).

III.3 Raffinement des diagrammes des cas d'utilisations

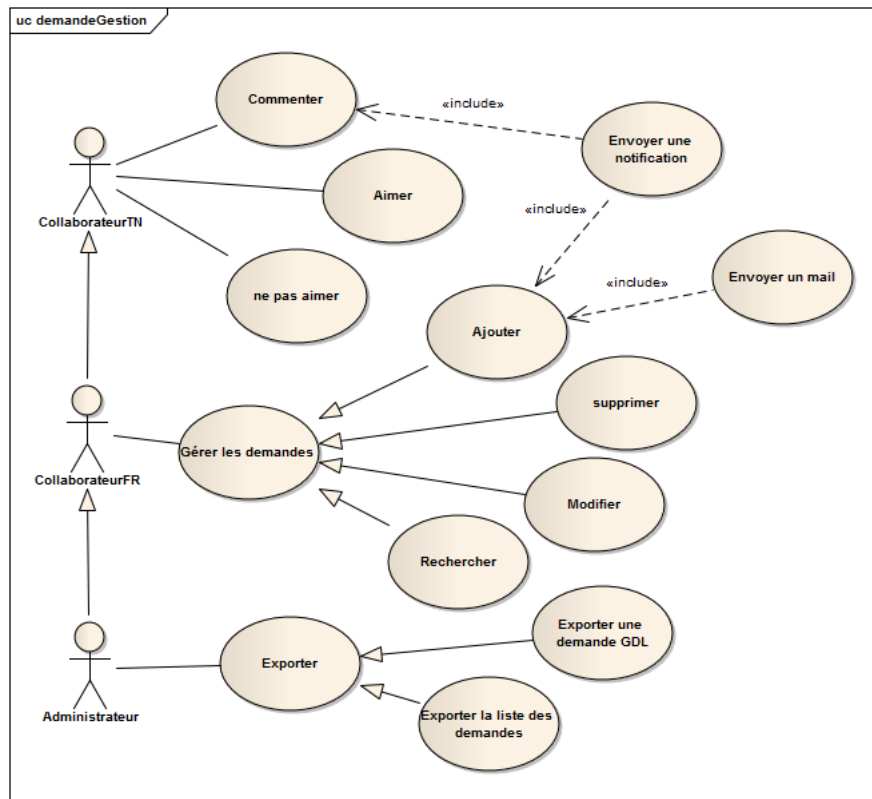


Figure III.3 – Diagramme de cas d'utilisation Gestion d'une demande

3.3 Module gestion des paramètres de la demandes.

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.

- 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 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

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

met en évidence ces cas d'utilisation.

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

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.

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

4 Diagrammes structurels

4.1 Diagramme de paquetages

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 [12]. La figure III.7 présente le diagramme de packetages.

En fait, ce diagramme met en évidence le modèle Model View Controller (MVC) [13], à 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

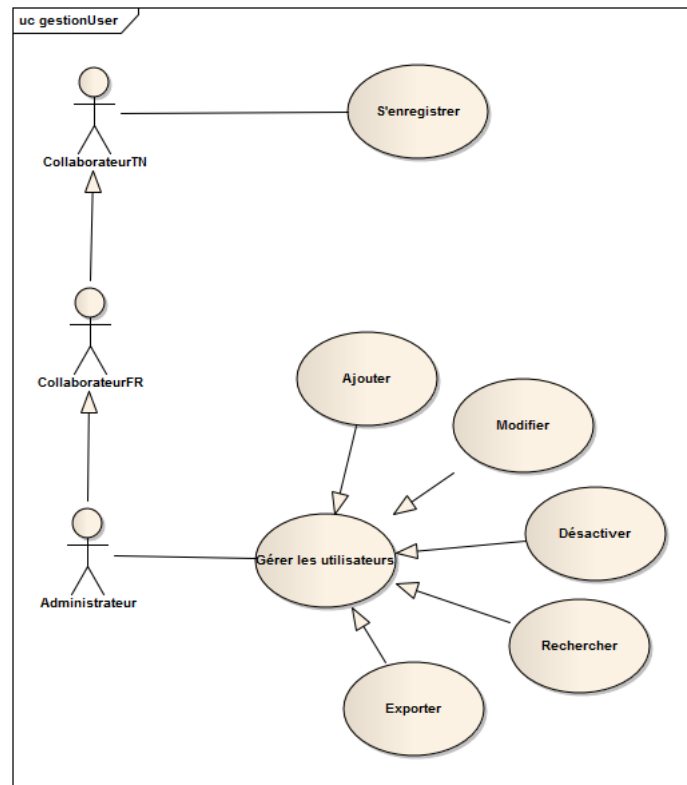


Figure III.4 – Diagramme de cas d'utilisation gestion utilisateurs

existe aussi d'autre packages : Service : il regroupe les services liés à notre application. Form : il regroupe les formulaires de l'application modélisés par des classes. Config : il contient le routage de l'application et les services.

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 3.6, nous représentons le rôle de chacune de ces entités.

- Diagramme de classe des contrôleurs.

La figure III.9 montre 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.

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

5 Diagramme 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

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

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

III.5 Diagramme comportementaux

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

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 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 dans 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

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

il manque la conclusion.

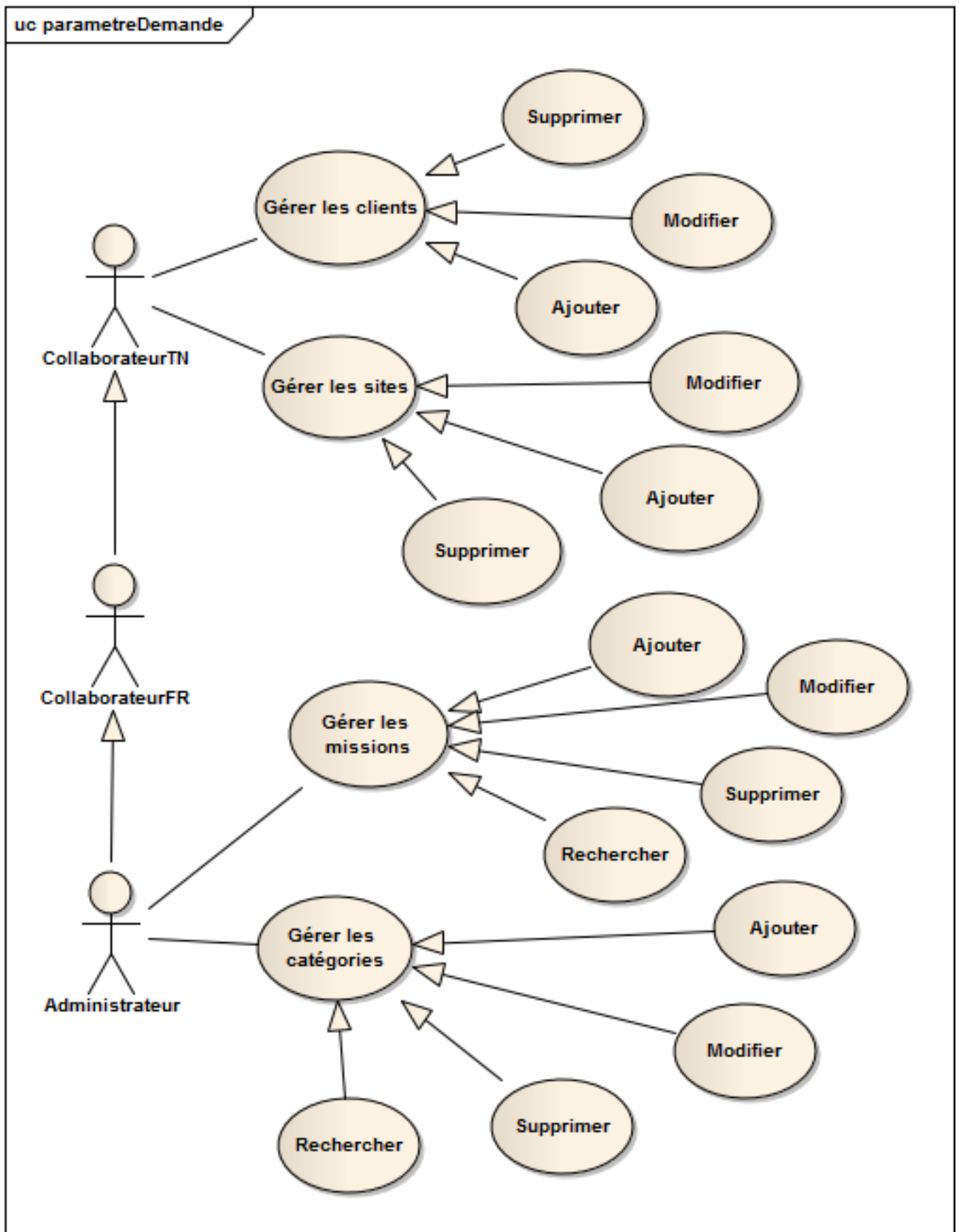


Figure III.5 – Diagramme d'utilisation du paramétrage d'une demande

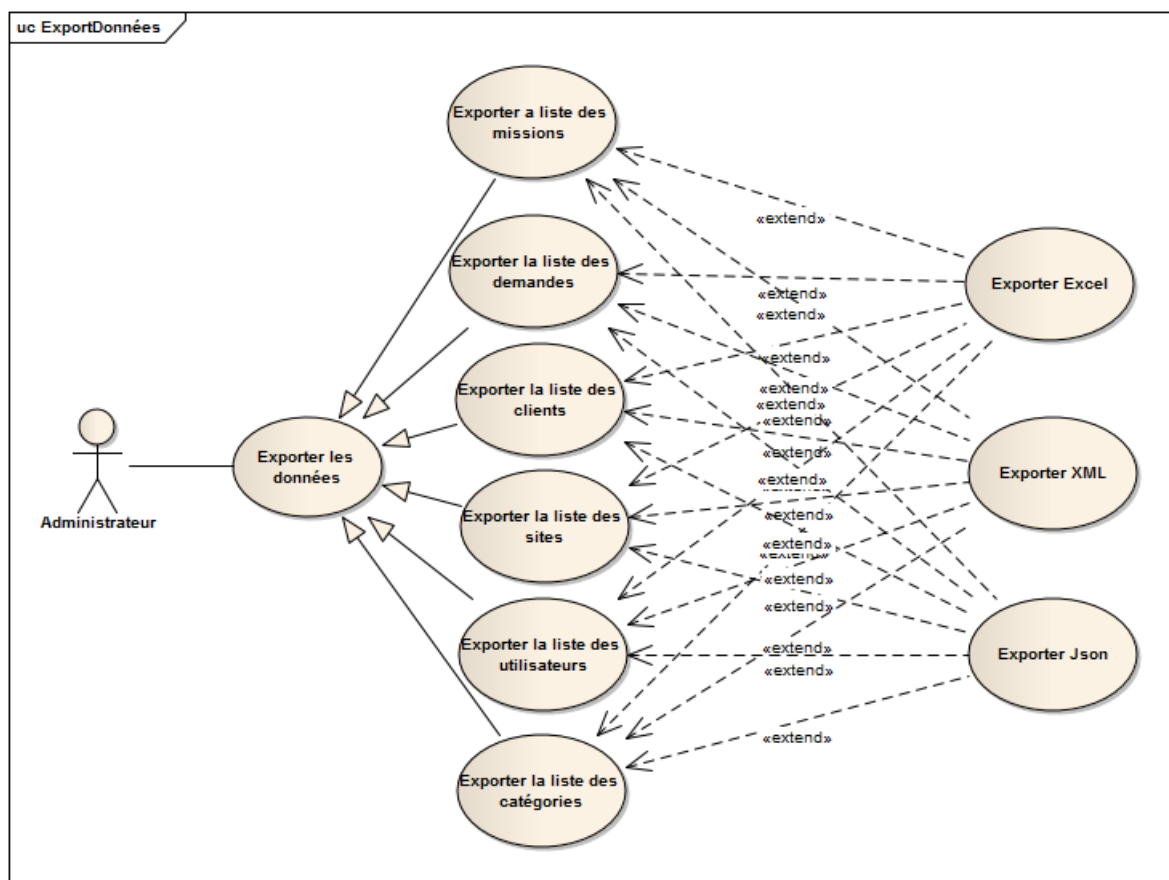


Figure III.6 – Diagramme de cas d'utilisation exportation des données

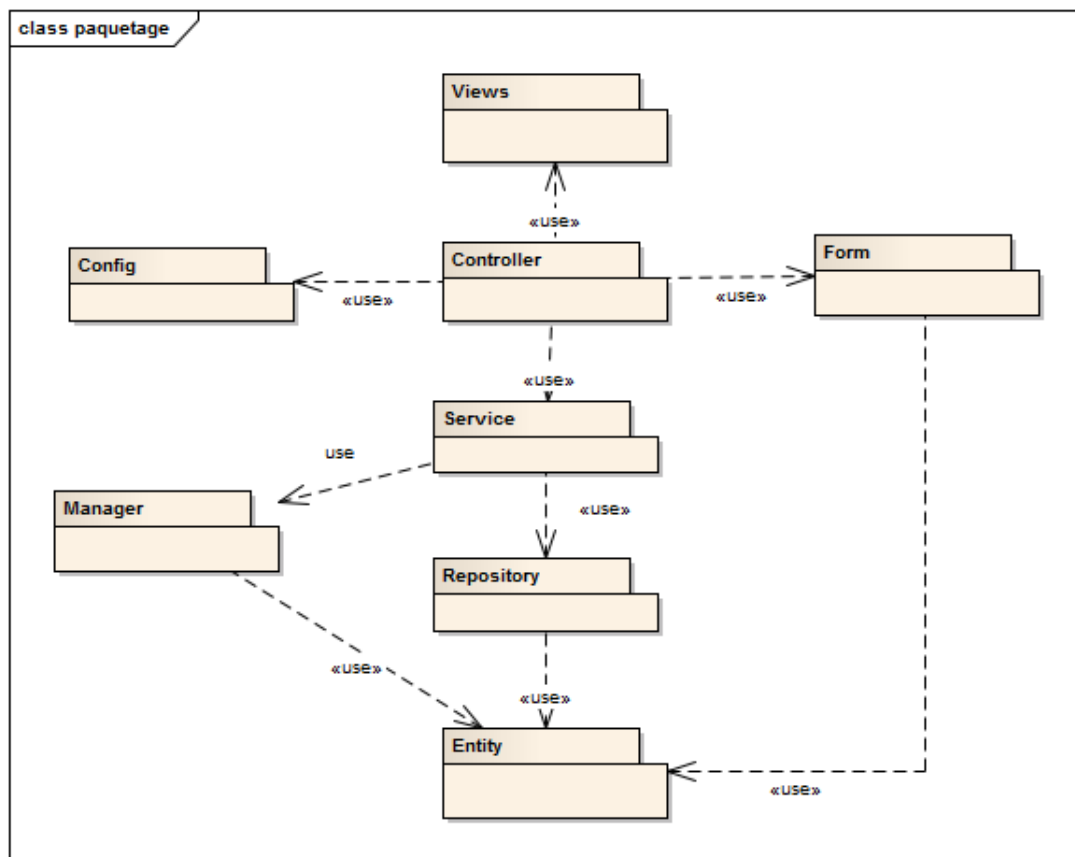


Figure III.7 – Diagramme de paquetage

III.5 Diagramme comportementaux

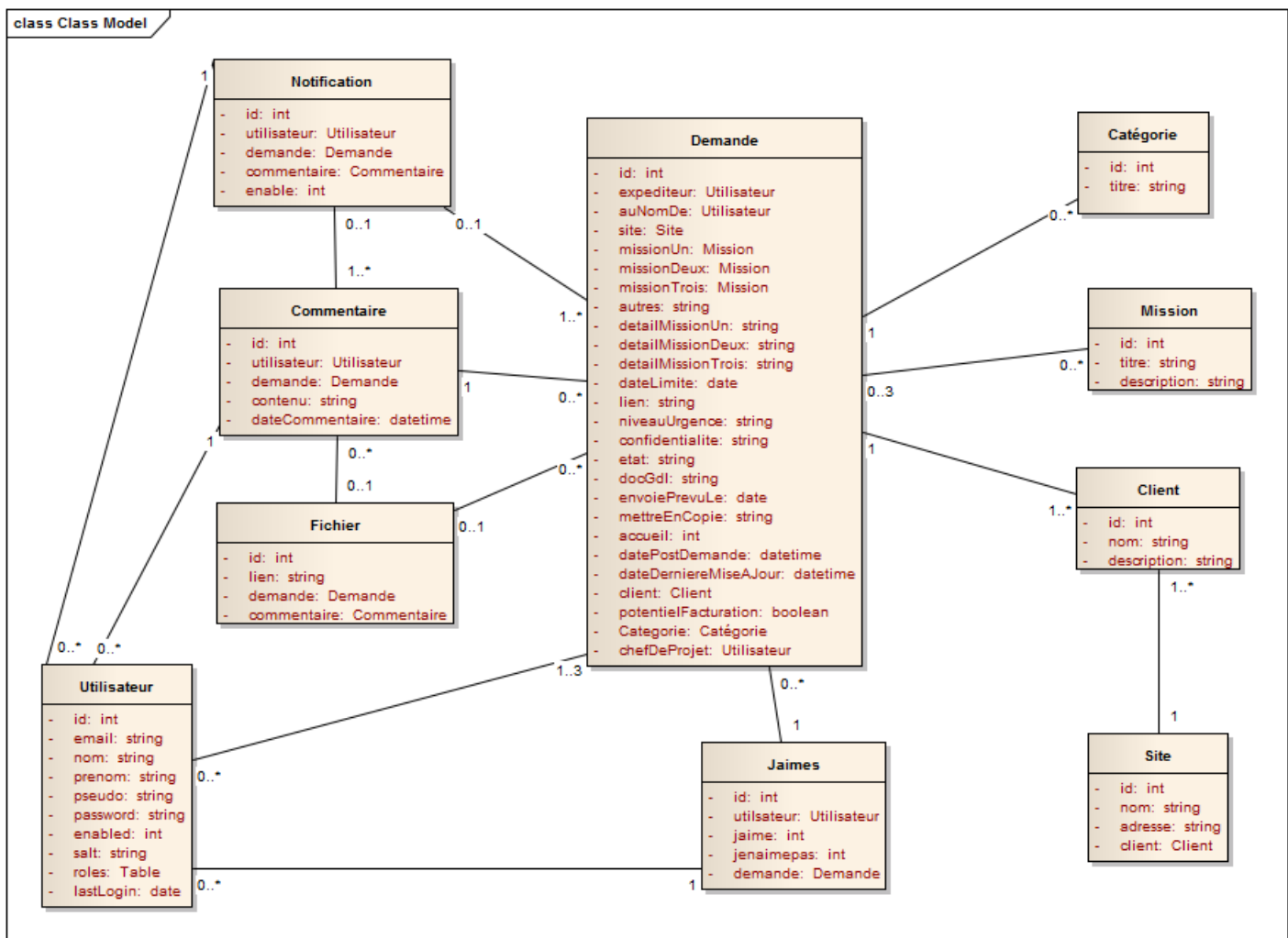


Figure III.8 – Le modèle du domaine

III.5 Diagramme comportementaux

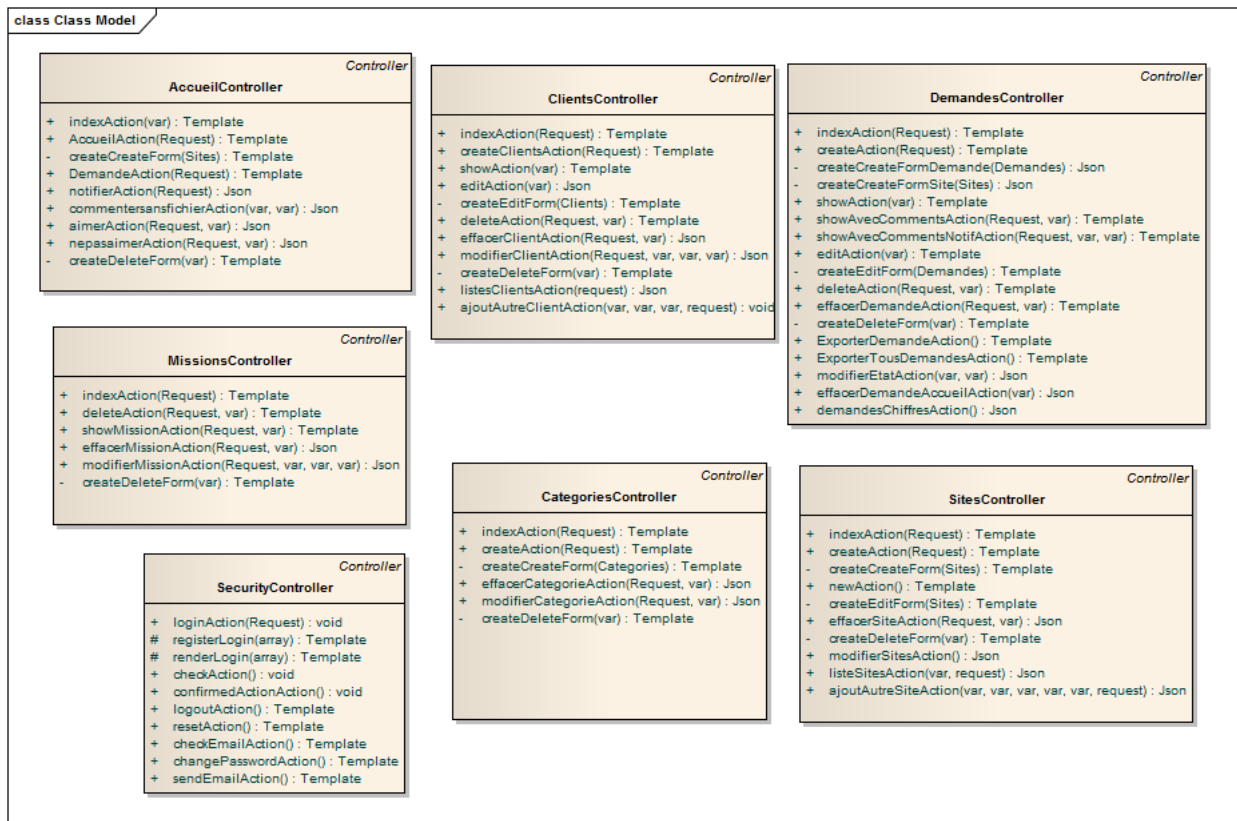


Figure III.9 – Le modèle du domaine

III.5 Diagramme comportementaux

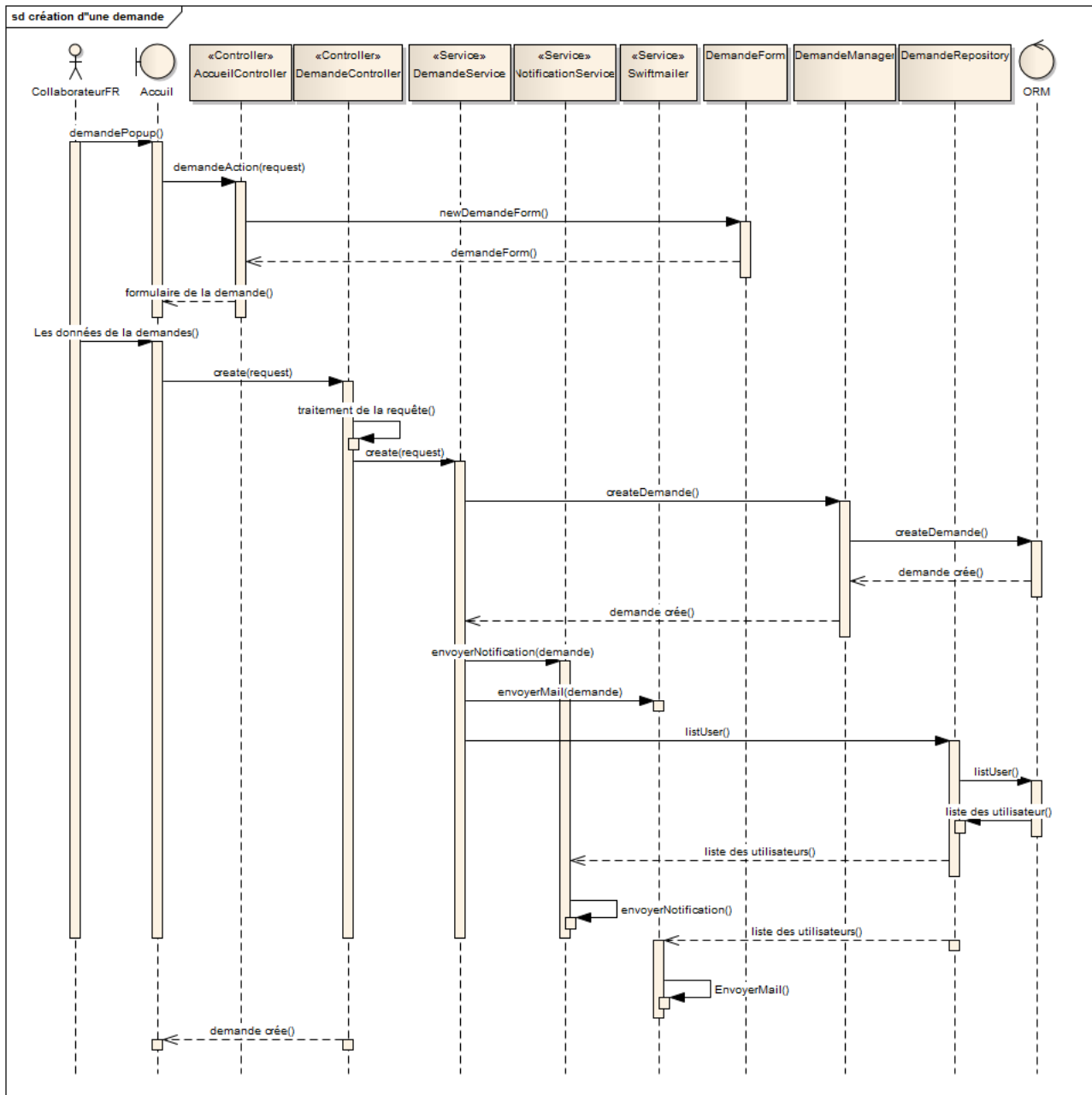


Figure III.10 – Diagramme de séquence ajout d'une demande

III.5 Diagramme comportementaux

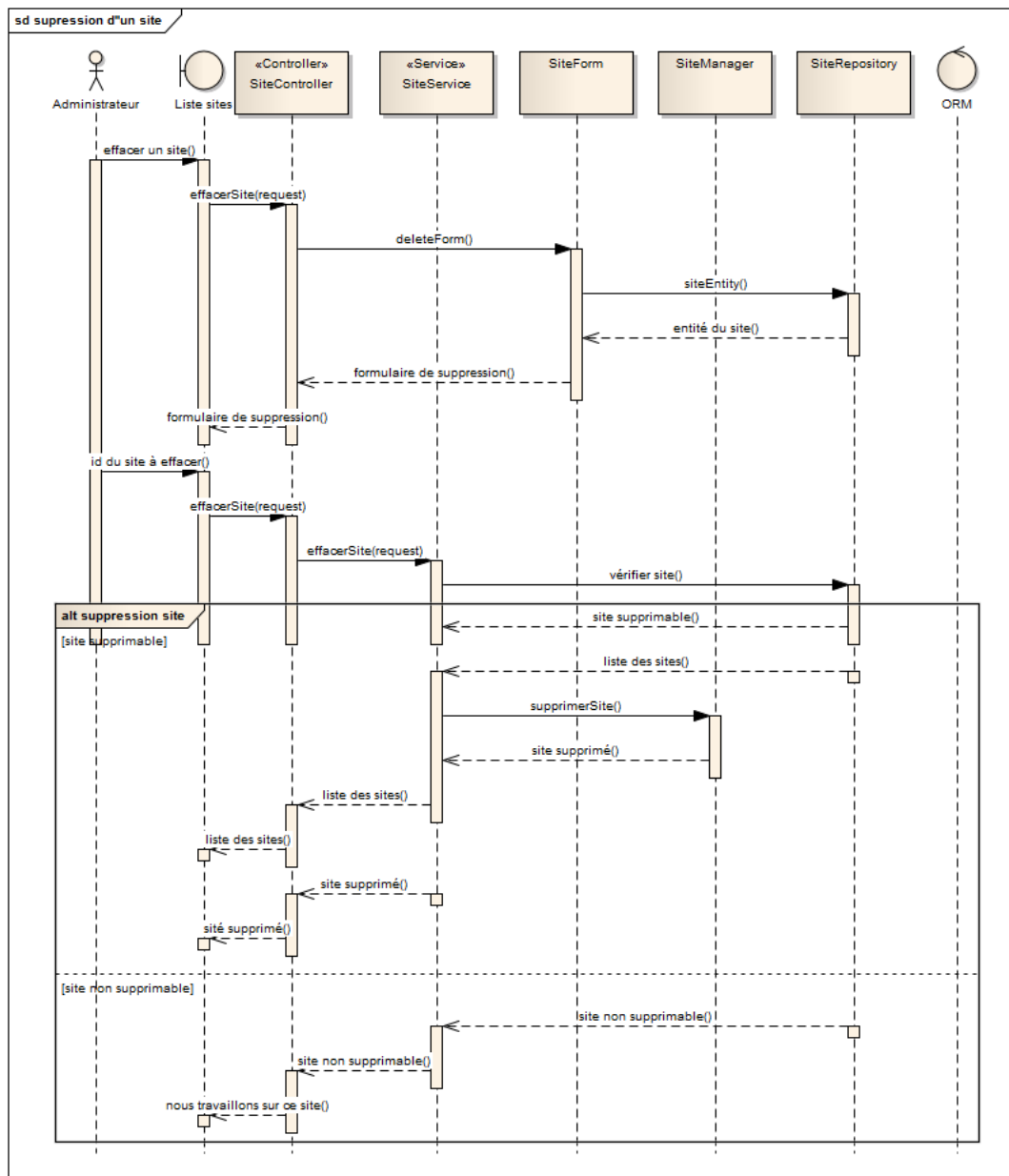


Figure III.11 – Diagramme de séquence suppression d'un site

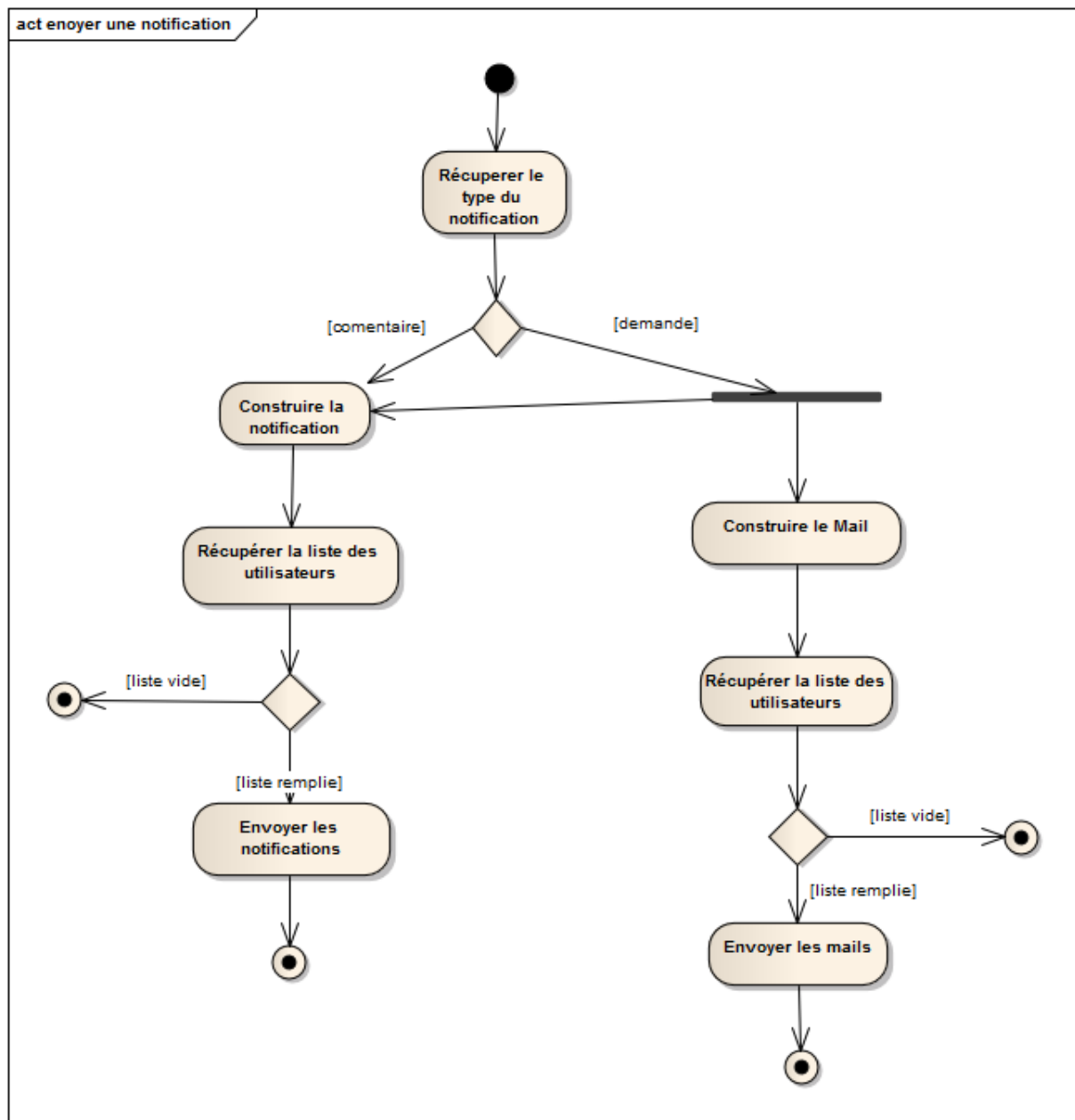


Figure III.12 – Diagramme d'activié : envoie d'une notification

Chapitre IV

Réalisation

Plan

1	Environnements de travail et choix techniques	35
1.1	Environnements de travail	35
1.2	Technologies	37
2	Présentation de l'application	40

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 langage e.g., php, java, c. C'est un produit oracle qui est maintenu à jour pour supporter les derniers technologies et framework les plus utilisés [14].
- Microsoft Office : Word 2013 pour la rédaction du cahier de charge et Excel 2013 pour l'exportation des différents 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.

IV.1 Environnements de travail et choix techniques

- 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 de 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 qualité optimale.
- Git et Github : afin de garder un 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 du Github avec quelques "commit" et toutes les contributions effectuées pendant notre travail.

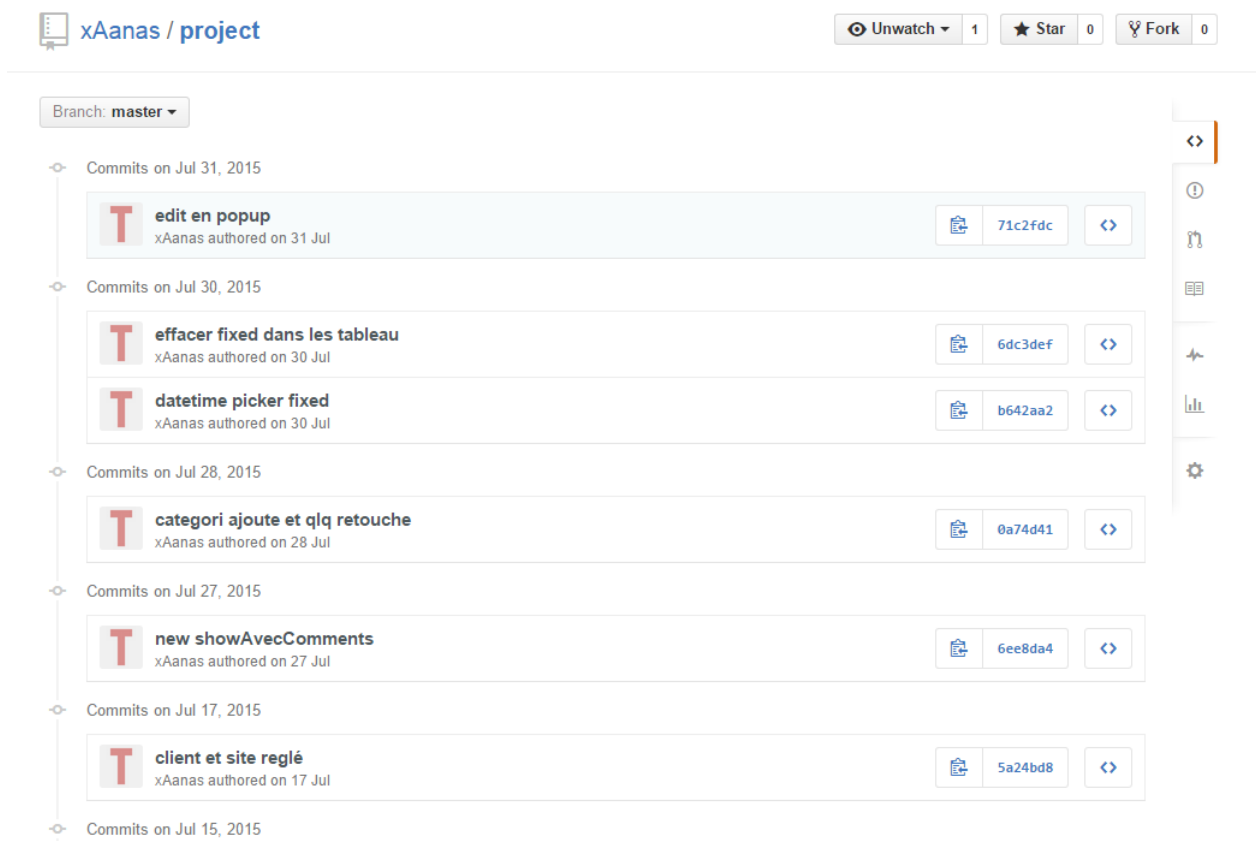


Figure IV.1 – Commits pendant le mois de Juillet

- Sublime Text 2 : pour le développement ou la modification de quelques scripts.
- Miktex : pour la génération du rapport.

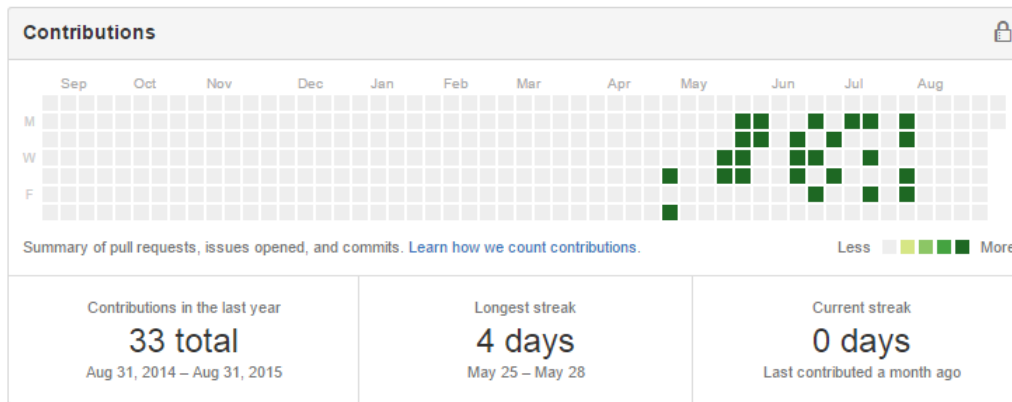


Figure IV.2 – Toutes les contributions

1.2 Technologies

Durant notre travail, plusieurs choix technique 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 donnée qu'un Framework n'est pas indispensable pour le développement d'une application 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 au règles du métier qui nous fournira un code maintenable, évolutif et des librairies réutilisables facilitant plusieurs tâches. Il nous permet donc d'éviter les anti-pattern comme don't repeat yourself (DIE), 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). Vue 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 [15]. En effet, avec le Cookbook symfony2 [16] 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 fonction-

nalités facilitant la manipulation de la base de données et communique directement avec la DBAL [17]. 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 qu'on vient 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 D3js 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

IV.1 Environnements de travail et choix techniques

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.

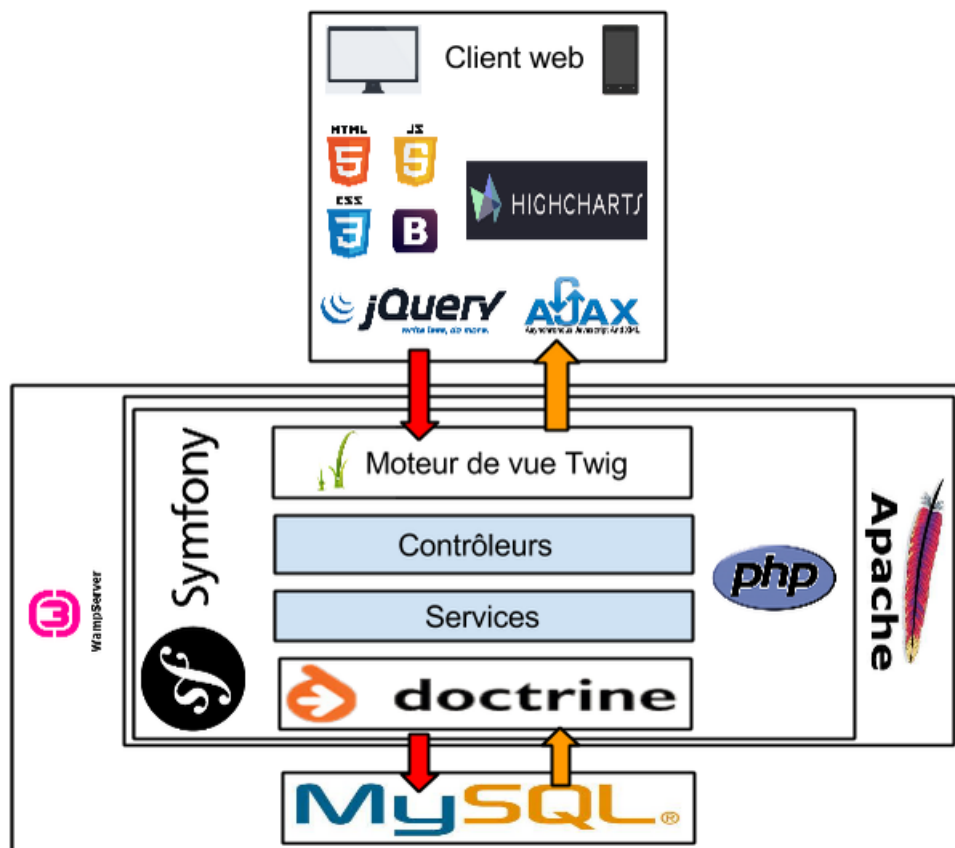


Figure IV.3 – Disposition des Frameworks

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 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

ellouze+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.

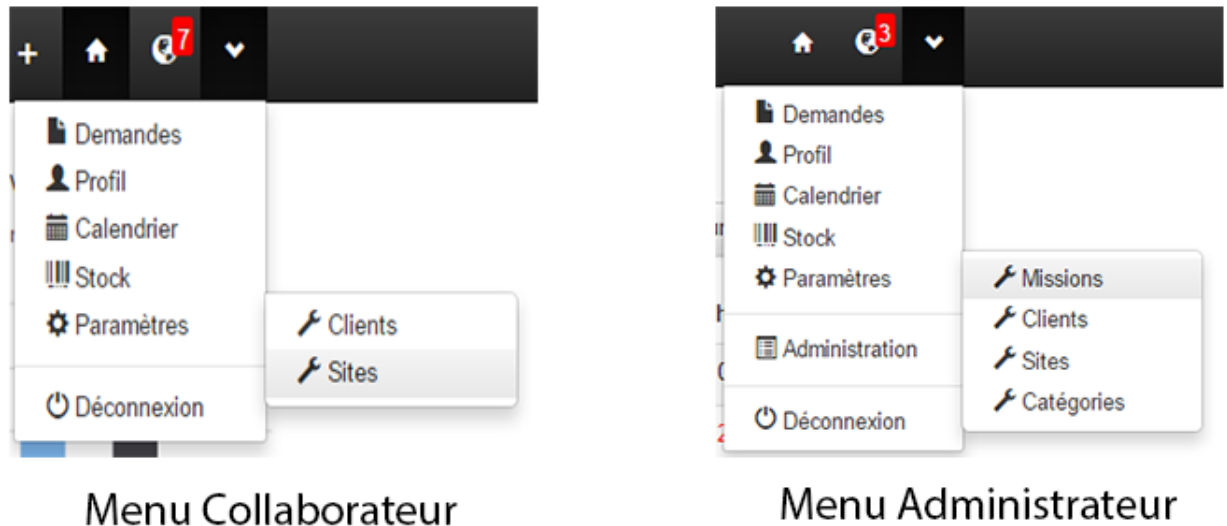


Figure IV.5 – Différence entre deux rôles

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 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 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.

– 3

Conclusion

Voilà.

IV.2 Présentation de l'application

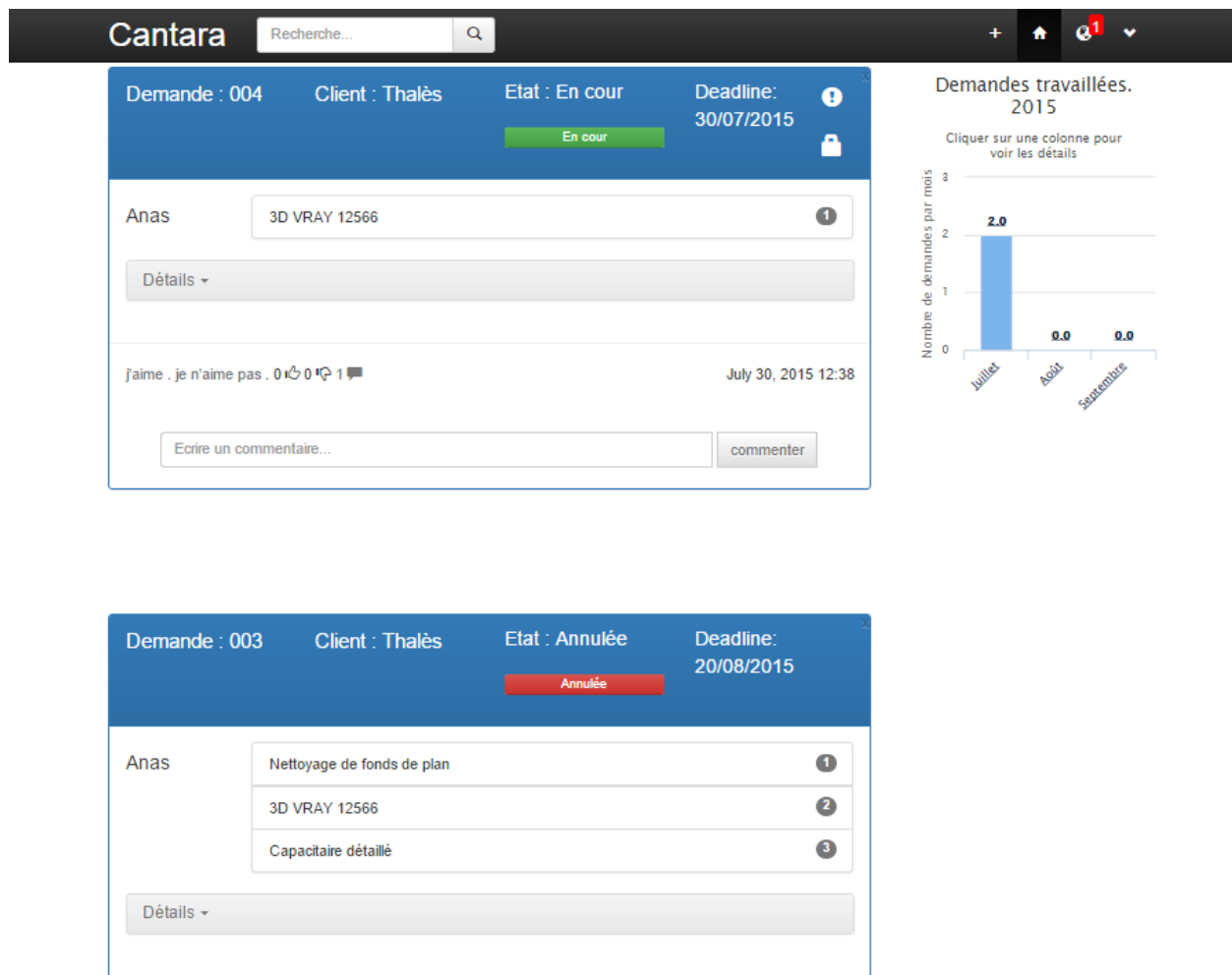


Figure IV.6 – Vue complète de l'interface d'accueil

IV.2 Présentation de l'application

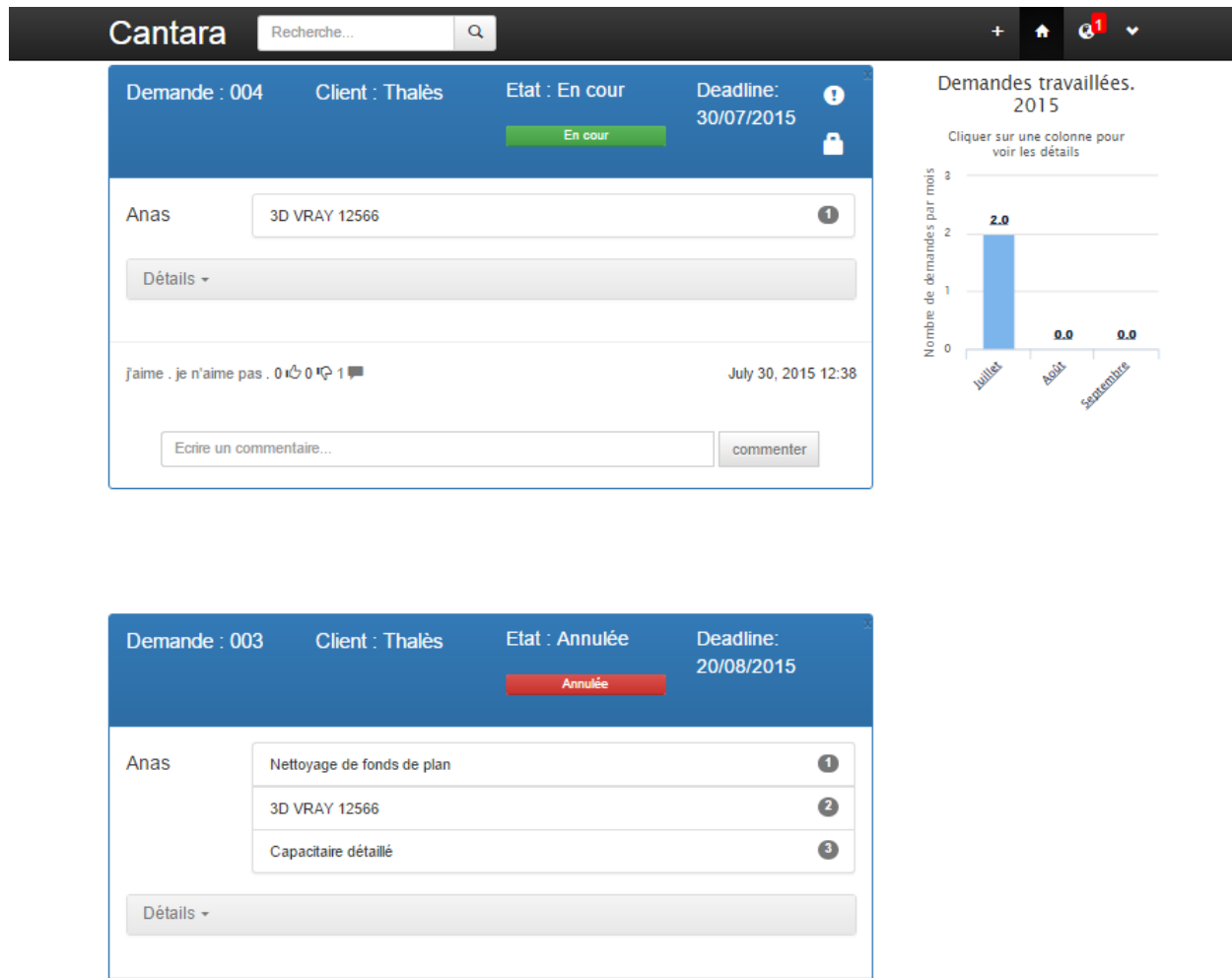


Figure IV.7 – Changer l'état d'une demande

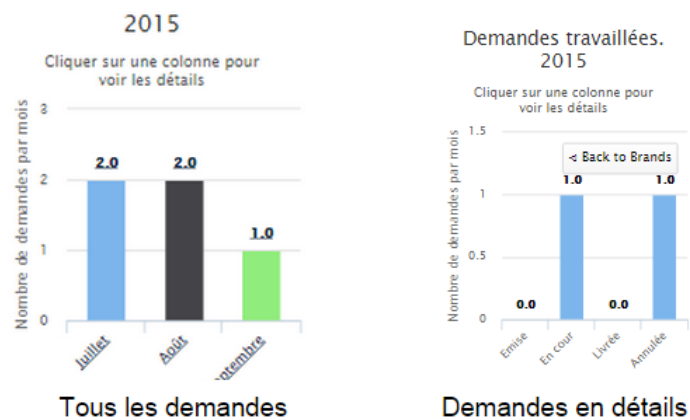


Figure IV.8 – Le graphe des demandes

Conclusion Générale et Perspectives

C'est l'une des parties les plus importantes et pourtant les plus négligées du rapport. Ce qu'on ne veut pas voir ici, c'est combien ce stage vous a été bénéfique, comment il vous a appris à vous intégrer, à connaître le monde du travail, etc.

Franchement, personne n'en a rien à faire, du moins dans cette partie. Pour cela, vous avez les remerciements et les dédicaces, vous pourrez vous y exprimer à souhait.

La conclusion, c'est très simple : c'est d'abord le résumé de ce que vous avez raconté dans le rapport : vous reprenez votre contribution, en y ajoutant ici les outils que vous avez utilisé, votre manière de procéder. Vous pouvez même mettre les difficultés rencontrées. En deuxième lieu, on y met les perspectives du travail : ce qu'on pourrait ajouter à votre application, comment on pourrait l'améliorer.

Bibliographique

- [1] <http://www.everwin.fr/solutions>. [En ligne ; consulté le 21-Juin-2015]. iv, 4
- [2] <http://www.dolibarr.fr/>. [En ligne ; consulté le 22-Juin-2015]. iv, 4, 5
- [3] LILIA SFAXI AND SOUHEIB YOUSFI. *Pour bien écrire un rapport*. Département Math-info (2015). 1
- [4] MR. LATEX. Débuter avec Latex. www.latex.com, (2008). [En ligne ; consulté le 19-Juillet-2008]. 1
- [5] SOUHEIB YOUSFI AND LILIA SFAXI. *Rapport Latex*. Département Math-info (2015). 2
- [6] <http://www.fitnetmanager.com/crm-avant-vente>. [En ligne ; consulté le 21-Juin-2015]. 3
- [7] PHILLIPE KRUCHTEN. *The Rational Unified Process : An Introduction*. (publié le 10 Décembre 2003). 6
- [8] KENT BECK. *Extreme Programming Explained : Embrace change*. (publié le 26 novembre 2014). 6
- [9] KEN SCHWABER ET JEFF SUTHERLAND. *Le Guide SCRUM*. (publié en juillet 2013). 6, 7, 10
- [10] JAMES MARTIN. *Rapid Application Development*. (1991). 6
- [11] https://stefanini.com/FR/2014/03/les_avantages_de_la_methode_scrum/. [En ligne ; consulté le 24-Juin-2015]. 7
- [12] [https://fr.wikipedia.org/wiki/Paquet_\(logiciel\)](https://fr.wikipedia.org/wiki/Paquet_(logiciel)). [En ligne ; consulté le 13-Septembre-2015]. 21
- [13] <https://netbeans.org/>. [En ligne ; consulté le 31-Août-2015]. 21
- [14] <http://blog.nicolashachet.com/technologies/php/rex-zend-2-vs-symfony-2/>. [En ligne ; consulté le 09-Septembre-2015]. 35
- [15] <https://fr.wikipedia.org/wiki/Modèle-vue-contrôleur>. [En ligne ; consulté le 31-Août-2015]. 37
- [16] <http://symfony.com/fr/doc/current/cookbook/index.html>. [En ligne ; consulté le 31-Août-2015]. 37
- [17] <https://packagist.org/>. [En ligne ; consulté le 13-Septembre-2015]. 38

- [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]. 38
- [19] <http://www.highcharts.com/>. [En ligne; consulté le 01-Septembre-2015]. 38
- [20] <http://getbootstrap.com/>. [En ligne; consulté le 01-Septembre-2015]. 38
- [21] <https://github.com/twbs/bootstrap>. [En ligne; consulté le 01-Septembre-2015]. 38
- [22] <http://jqueryui.com/>. [En ligne; consulté le 01-Septembre-2015]. 39
- [23] <https://angularjs.org/>. [En ligne; consulté le 13-Septembre-2015]. 39

Annexe : Remarques Diverses

- Un rapport doit toujours être bien numéroté ;
- De préférence, ne pas utiliser plus que deux couleurs, ni un caractère fantaisiste ;
- Essayer de toujours garder votre rapport sobre et professionnel ;
- Ne jamais utiliser de je ni de on, mais toujours le nous (même si tu as tout fait tout seul) ;
- Si on n'a pas de paragraphe 1.2, ne pas mettre de 1.1 ;
- TOUJOURS, TOUJOURS faire relire votre rapport à quelqu'un d'autre (de préférence qui n'est pas du domaine) pour vous corriger les fautes d'orthographe et de français ;
- Toujours valoriser votre travail : votre contribution doit être bien claire et mise en évidence ;
- Dans chaque chapitre, on doit trouver une introduction et une conclusion ;
- Ayez toujours un fil conducteur dans votre rapport. Il faut que le lecteur suive un raisonnement bien clair, et trouve la relation entre les différentes parties ;
- Il faut toujours que les abréviations soient définies au moins la première fois où elles sont utilisées. Si vous en avez beaucoup, utilisez un glossaire.
- Vous avez tendance, en décrivant l'environnement matériel, à parler de votre ordinateur, sur lequel vous avez développé : ceci est inutile. Dans cette partie, on ne cite que le matériel qui a une influence sur votre application. Que vous l'ayez développé sur Windows Vista ou sur Ubuntu n'a aucune importance ;
- Ne jamais mettre de titres en fin de page ;
- Essayer toujours d'utiliser des termes français, et éviter l'anglicisme. Si certains termes sont plus connus en anglais, donner leur équivalent en français la première fois que vous les utilisez, puis utilisez le mot anglais, mais en italique ;
- Éviter les phrases trop longues : clair et concis, c'est la règle générale !

Rappelez vous que votre rapport est le visage de votre travail : un mauvais rapport peut éclipser de l'excellent travail. Alors prêtez-y l'attention nécessaire.