



Projet de Fin d'Etudes

Licence Sciences et Techniques Génie Informatique

Application web pour la gestion des commandes



Lieu de stage : GiantLink

Réalisé par :

Saad HOUMADI
Reda JANJARY

Encadré par :

Pr. Youness OUBENAALLA
Mr. Ayoub CHAMLALI

Soutenu le 06/07/2022 devant le jury composé de :

Pr. Loubna LAMRINI

Pr. Adil BENABBOU

Pr. Youness OUBENAALLA

Remerciements

Nous remercions tout d'abord Dieu tout puissant de nous avoir donné le courage, la force et la patience pour achever ce travail.

Un grand merci à Pr. Youness Oubenaalla notre encadrant à la FST, pour son assistance, sa disponibilité, ses conseils judicieux lors de la réalisation de ce projet et son aide à l'aboutissement de la bonne organisation de ce rapport.

On tient également à exprimer toute notre gratitude aux membres du jury, Pr Loubna LAMRINI et Pr Adil BENABBOU pour avoir accepté d'assister à ce modeste travail. Enfin, nous ne voulons pas oublier tous ceux qui ne cessent de nous encourager de près ou de loin.

Résumé

Ce rapport que on vous présente aujourd'hui contient la synthèse du travail qu'on a réalisé pendant notre projet de fin d'étude au sein de l'entreprise GiantLink. Le but de ce projet est la création d'une application web qui permettra de gérer les commandes. Ce système vise à automatiser toutes les gestions nécessaires qui facilite le travail au sein du centre d'appel. La cause pour laquelle GiantLink nous ont assigné ce projet est le mal fonctionnement de leur ancienne application web qui était faite avec HTML, CSS et PHP.

Le présent document constitue le rapport du projet réalisé dans le cadre de ce stage. Il est organisé en trois chapitres :

Le premier chapitre décrit le contexte général du projet. Nous présenterons les objectifs de notre projet ainsi que le cahier de charges.

Dans, le deuxième chapitre nous allons présenter l'analyse des besoins et la conception du système. Dans ce cadre nous détaillerons les différents diagrammes UML réalisés.

Le dernier chapitre est consacré à la présentation de l'application réalisée ainsi que les outils de développement utilisés.

Enfin, nous terminerons ce mémoire par une conclusion et quelques perspectives futures.

Abstract

This report that we present to you today contains the synthesis of the work that we carried out during our graduation project within the company GiantLink. The goal of this project is the creation of a web application to manage orders. This system aims to automate all the necessary management that facilitates work within the call center. The cause why GiantLink assigned us this project is the malfunction of their old web application which was made with HTML, CSS and PHP.

This document constitutes the report of the project carried out within the framework of this internship. It is organized into three chapters:

The first chapter describes the general context of the project. We will present the objectives of our project as well as the specifications.

In the second chapter we will present the needs analysis and the design of the system. In this context we will detail the different UML diagrams made.

The last chapter is dedicated to the presentation of the realized application as well as the development tools used.

Finally, we will end this dissertation with a conclusion and some future perspectives.

Table des matières

Résumé	3
Abstract	4
Chapitre 1 : Contexte général du projet	9
Introduction.....	10
1.1 Présentation du lieu de stage	10
1.2 Les activités de GiantLink	10
1.3 Problématique et solution	11
1.4 Cahier de charge	11
a- Gestion des agents	11
b- Gestion des projets.....	11
c- Gestion des services	11
d- Gestion des options	11
e- Gestion des produits.....	11
f- Gestions des clients	12
g- Gestion des types d'appels.....	12
h- Gestion des commerciaux	12
i- Gestion des formules.....	12
j- Gestion des supérieurs	12
k- Gestion des équipes	12
l- Gestion des appels.....	12
1.5 Diagramme de Gantt	12
Chapitre 2 : Analyse et conception	14
2.1 Langage et outil de modélisation	15
2.1.1 Langage de modélisation UML	15
2.1.2 L'outil de modélisation	15
2.2 Analyse des besoins	15
2.2.1 Les acteurs du système.....	15
Définition d'un acteur :.....	15
Gestion des agents	16
Gestion des projets	16
Gestion des services	16
Gestion des types d'appels	16
Gestion des options	16

Gestion des produits	17
Gestion des commerciaux	17
Gestion des formules	17
Gestion des supérieurs	17
Gestion des équipes	17
Gestion des appels	17
Gestion des clients	18
2.2.2 Diagramme de cas d'utilisation	18
2.2.3 Diagramme d'activité d'ajout d'un agent.....	21
2.2.4 Diagramme d'activité d'ajout d'un appel	23
2.3 Modèle statique	24
2.3.1 Diagramme de classe.....	24
2.3.2 schémas de la base de données	25
2.3 Conception méthodique	26
Les avantages de la méthode Agile :	27
Le Fonctionnement de la méthode Agile :	27
Conclusion	28
Chapitre 3 : Réalisation	29
3.1 Framework de développement	30
3.1.1 Pourquoi utilisé un Framework ?	30
3.1.2 Angular Framework	30
3.1.2 AngularJS Framework	31
3.1.3 Différence entre Angular et AngularJS	31
3.2 Environnement du développement	32
3.2.1 Environnement matériel	32
3.2.2 Environnement logiciel.....	32
Visual Code Studio.....	32
HTML 5.....	32
Bootstrap.....	32
CSS 3	33
SCSS or SASS	33
Photoshop.....	33
Angular	33
TypeScript.....	33

MySQL.....	34
Spring.....	34
Postman.....	34
GitLab.....	34
3.3 Présentation des interfaces graphiques	35
3.3.1 Le menu	35
3.3.2 L'accueil	37
3.3.3 Gestion des projets.....	38
3.3.4 Gestion des agents	43
Conclusion Et Perspectives	46
Webographie.....	47
Documentation:.....	47

Liste des figures

Figure 1: Diagramme de Gantt	13
Figure 2: Diagramme de cas d'utilisation de l'agent	18
Figure 3: Diagramme de cas d'utilisation de l'administrateur partie 1	19
Figure 4: Diagramme de cas d'utilisation de l'administrateur partie 2	20
Figure 5: Diagramme d'activité d'ajout d'un agent.....	21
Figure 6: Diagramme d'activité d'ajout des informations d'un appel	23
Figure 7 : Diagramme de classes	25
Figure 8: Explication simple de la méthode Agile.....	28
Figure 9: Logo du FrameWork Angular	31
Figure 10: Logo du FrameWork AngularJS	31
Figure 11: Menu de navigation partie 1.....	35
Figure 12: Menu de navigation partie 2.....	36
Figure 13: Menu de navigation partie 3.....	36
Figure 14: Les points cumulés par l'agent connecté.....	37
Figure 15: Dashboard d'accueil montrant le classement des agents selon les points cumulés.....	37
Figure 16: Interface de gestion des projets.....	38
Figure 17: Interface de création d'un projet.....	39
Figure 18: Message de succès après l'ajout d'un projet.....	40
Figure 19: Interface de modification d'un projet.....	40
Figure 20: Message de confirmation lors d'une modification d'un projet	41
Figure 21: Message de succès après la modification d'un projet.....	41
Figure 22: Message de confirmation lors d'une suppression d'un projet	42
Figure 23 : Message de succès après la suppression d'un projet.....	42
Figure 24: Interface de gestion des agents	43
Figure 25 : Interface d'ajout d'un utilisateur	43
Figure 26: Message d'erreur lorsqu'un champ ou plusieurs sont vide	44

Figure 27: Message d'erreur lorsque la forme du mot de passe est incorrecte 44
Figure 28: Message d'erreur lorsque le mot de passe et confirmer le mot de passe ne sont pas identiques .. 45

Liste des tableaux

Tableau 1: Description du cas d'utilisation Ajouter un agent 21
Tableau 2 : Description du cas d'utilisation Ajouter un appel 23

Chapitre 1 : Contexte général du projet

Introduction

Avoir une idée globale et claire sur le projet à réaliser est primordial pour le bon déroulement de tout projet. Cette étude fera donc l'objet de ce chapitre qui est consacré à la présentation de l'organisme d'accueil, la présentation du cahier de charges ainsi que les besoins fonctionnels du système.

1.1 Présentation du lieu de stage

Notre projet de fin d'études est effectué au sein de GiantLink à Fès. GiantLink est une entreprise multinationale, multi sites, spécialisée dans des divers services : Téléventes, Prise de rendez-vous, hotline, Gestion de la relation client, les services de commerce, et la gestion des processus. Depuis sa création en 2010, Giantlink a incubé plus de projets dont certains se sont rapidement imposés comme leader de leur métier au Maroc.

1.2 Les activités de GiantLink

Téléventes : Le centre d'appel complète la force de vente pour promouvoir de nouveaux produits ou pour une campagne commerciale.

Prise de rendez-vous : GiantLink fourni des rendez-vous-qualifiés qui sont validés par le service avec enregistrement selon la demande du client.

Hotline : GiantLink dispose d'outils de réception d'appel mis à la disposition des clients pour le support technique ou la demande de renseignement. Ce service permanent ou ponctuel en fonction des besoins.

Gestion de relation client : GiantLink propose de l'aider dans la gestion de la relation client en complétant le dispositif de ressources pour des études d'opinion.

Administration : Les tâches administratives sont partout dans l'entreprise. Celles-ci peuvent être extrêmement en majeure partie. Le personnel administratif qui travaille en back-office peut opérer à distance pour la réception d'appel. La saisie des données et la gestion de documents.

Services IT : (Information technologie) Pour personnaliser le service en ligne du client, et pour le développement des applications qui facilite le travail pour les responsables.

1.3 Problématique et solution

Depuis la mise en place de l'ancienne application web, les objectives principales étaient atteintes avec une fréquence limitée, c'est pour cela GiantLink nous ont assigné pour développer une nouvelle application web basé sur l'ancienne en ajoutant des nouvelles fonctionnalités.

1.4 Cahier de charge

Après quelques discussions avec les responsables du projet, nous avons identifié les fonctionnalités suivantes, classées par modules.

Pour notre projet on aura 2 types d'utilisateurs : l'agent et l'administrateur.

L'agent a des accès limités contrairement à l'administrateur.

a- Gestion des agents

Chaque administrateur a le droit d'ajouter un nouvel agent ou bien de modifier et/ou de supprimer un qui existe déjà en consultant la liste des agents.

b- Gestion des projets

Chaque administrateur a le droit d'ajouter un nouveau projet ou bien de modifier et/ou de supprimer un qui existe déjà en consultant la liste des projets.

c- Gestion des services

Chaque administrateur a le droit d'ajouter un nouveau service ou bien de modifier et/ou de supprimer un qui existe déjà en consultant la liste des services.

d- Gestion des options

Chaque administrateur a le droit d'ajouter une nouvelle option ou bien de modifier et/ou de supprimer une qui existe déjà en consultant la liste des options.

e- Gestion des produits

Chaque administrateur a le droit d'ajouter un nouveau produit ou bien de modifier et/ou de supprimer un qui existe déjà en consultant la liste des produits.

f- Gestions des clients

Chaque utilisateur a le droit d'ajouter un nouveau client ou bien de modifier et/ou de supprimer un qui existe déjà en consultant la liste des clients.

g- Gestion des types d'appels

Chaque administrateur a le droit d'ajouter un nouveau type d'appel ou bien de modifier et/ou de supprimer un qui existe déjà en consultant la liste des types d'appels.

h- Gestion des commerciaux

Chaque administrateur a le droit d'ajouter un nouveau commercial ou bien de modifier et/ou de supprimer un qui existe déjà en consultant la liste des commerciaux.

i- Gestion des formules

Chaque administrateur a le droit d'ajouter une nouvelle formule ou bien de modifier et/ou de supprimer une qui existe déjà en consultant la liste des formules.

j- Gestion des supérieurs

Chaque administrateur a le droit d'ajouter un nouveau supérieur ou bien de modifier et/ou de supprimer un qui existe déjà en consultant la liste des supérieurs.

k- Gestion des équipes

Chaque administrateur a le droit d'ajouter une nouvelle équipe ou bien de modifier et/ou de supprimer une qui existe déjà en consultant la liste des équipes.

l- Gestion des appels

Chaque utilisateur a le droit d'ajouter un nouvel appel ou bien de supprimer un qui existe déjà en consultant la liste des appels.

1.5 Diagramme de Gantt

Pour une meilleure compréhension de la chronologie des étapes de notre projet, nous avons effectué le diagramme de Gantt.

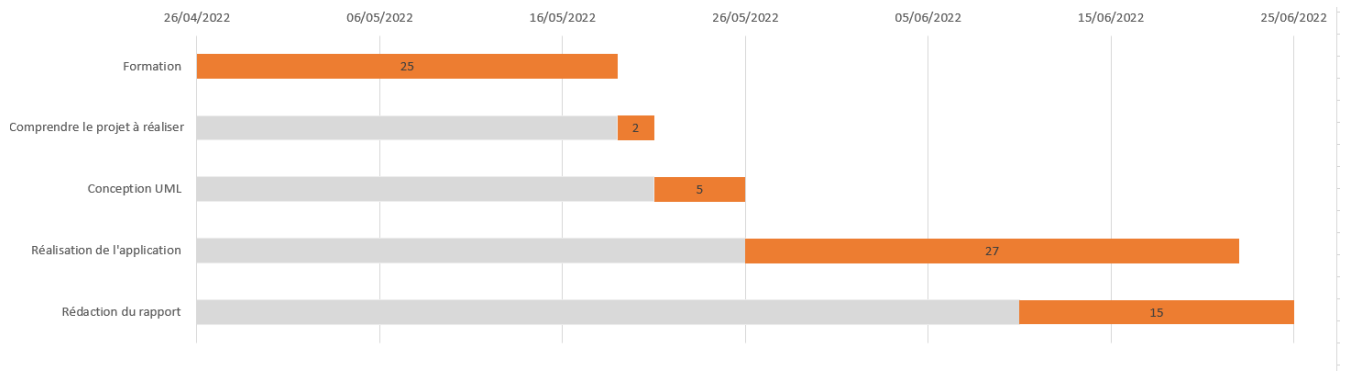


Figure 1: Diagramme de Gantt

Chapitre 2 : Analyse et conception

2.1 Langage et outil de modélisation

2.1.1 Langage de modélisation UML

Pour la conception de notre système nous avons adopté une méthode orientée objet. En effet cette dernière est une approche incontournable dans le cadre du développement des applications. Pour mieux présenter l'architecture de notre système, nous avons choisi le langage de modélisation le plus adopté :

UML : C'est un langage de modélisation, défini comme une norme de modélisation objet qui sert à décrire et à documenter un système d'information.

2.1.2 L'outil de modélisation

Pour faciliter l'établissement des différents diagrammes UML, nous avons utilisé Enterprise Architect. Enterprise Architect est un logiciel de modélisation et de conception UML, édité par la société australienne Sparx Systems. Couvrant par ses fonctionnalités, l'ensemble des étapes du cycle de conception d'application, il est l'un des logiciels de conception et de modélisation les plus reconnus.

2.2 Analyse des besoins

2.2.1 Les acteurs du système

Définition d'un acteur :

Un acteur est un rôle joué par des entités externes qui interagissent directement avec le système étudié : un utilisateur, un matériel externe ou un autre système. Pour notre application, les acteurs qui vont intervenir avec le système sont :

Les administrateurs : ce sont ceux qui ont le droit d'accéder à toutes les gestions se trouvent dans l'application. Leurs principales interactions avec le système se résument comme le suit :

Gestion des agents

- 1- *Lister tous les agents.*
- 2- *L'ajout d'un nouvel agent.*
- 3- *Modification ou/et suppression d'un agent déjà existant.*

Gestion des projets

- 1- *Lister tous les projets.*
- 2- *L'ajout d'un nouveau projet.*
- 3- *Modification ou/et suppression d'un projet déjà existant.*

Gestion des services

- 1- *Lister tous les services.*
- 2- *L'ajout d'un nouveau service.*
- 3- *Modification ou/et suppression d'un service déjà existant.*

Gestion des types d'appels

- 1- *Lister tous les types d'appels.*
- 2- *L'ajout d'un nouveau type d'appel.*
- 3- *Modification ou/et suppression d'un type d'appel déjà existant.*

Gestion des options

- 1- *Lister toutes les options.*
- 2- *L'ajout d'une nouvelle option.*
- 3- *Modification ou/et suppression d'une option déjà existante.*

Gestion des produits

- 1- *Lister tous les produits.*
- 2- *L'ajout d'un nouveau produit.*
- 3- *Modification ou/et suppression d'un produit déjà existant.*

Gestion des commerciaux

- 1- *Lister tous les commerciaux.*
- 2- *L'ajout d'un nouveau commercial.*
- 3- *Modification ou/et suppression d'un commercial déjà existant.*

Gestion des formules

- 1- *Lister toutes les formules.*
- 2- *L'ajout d'une nouvelle formule.*
- 3- *Modification ou/et suppression d'une formule déjà existante.*

Gestion des supérieurs

- 1- *Lister tous les supérieurs.*
- 2- *L'ajout d'un nouveau supérieur.*
- 3- *Modification ou/et suppression d'un supérieur déjà existant.*

Gestion des équipes

- 1- *Lister toutes les équipes.*
- 2- *L'ajout d'une nouvelle équipe.*
- 3- *Modification ou/et suppression d'une équipe déjà existante.*

Les agents : ce sont les utilisateurs responsables des appels et des clients. Leurs principales interactions avec le système se résume comme le suit :

Gestion des appels

- 1- *Lister tous les appels.*

- 2- L'ajout d'un nouvel appel.
- 3- Suppression d'un appel déjà existant.

Gestion des clients

- 1- Lister tous les clients.
- 2- L'ajout d'un nouveau client.
- 3- Modification ou/et suppression d'un client déjà existant.

2.2.2 Diagramme de cas d'utilisation

Le diagramme de cas d'utilisation est un diagramme UML utilisé pour une représentation du comportement fonctionnel d'un système logiciel. Il est utile pour des présentations auprès de la direction ou des acteurs d'un projet, mais pour le développement, les cas d'utilisation sont plus appropriés.

Nous avons élaboré notre diagramme des cas d'utilisation.

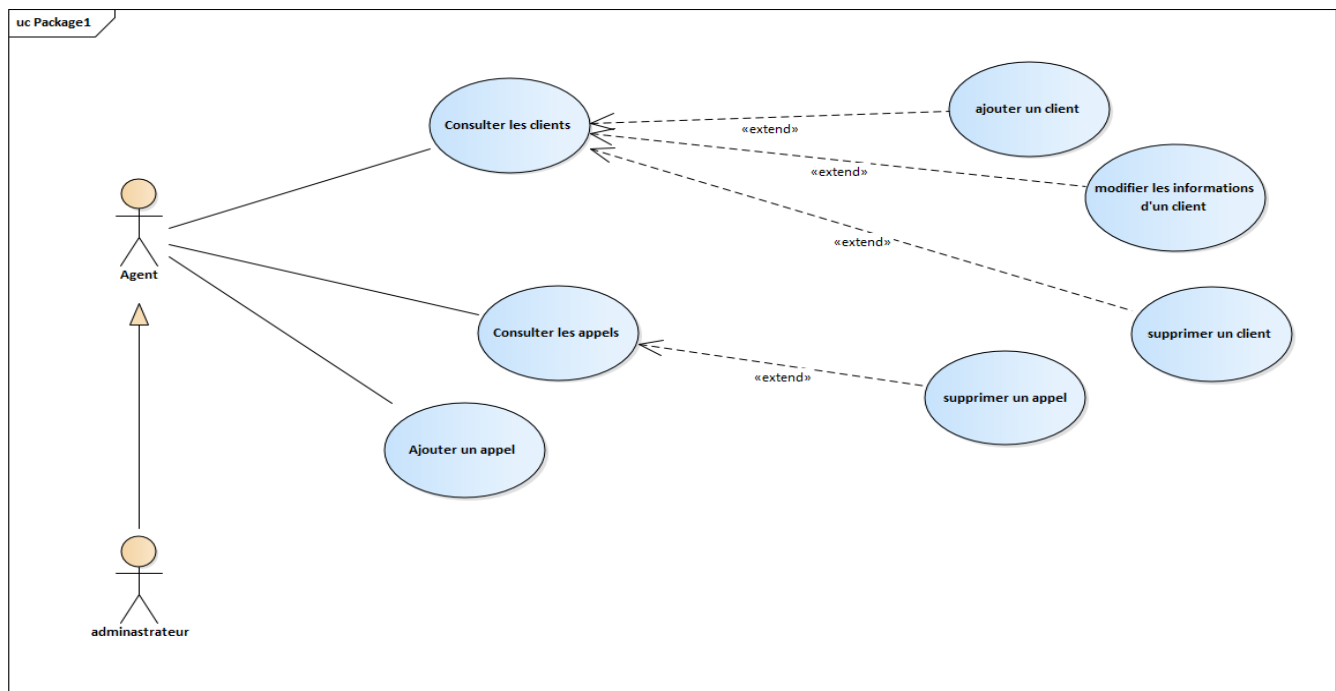


Figure 2: Diagramme de cas d'utilisation de l'agent

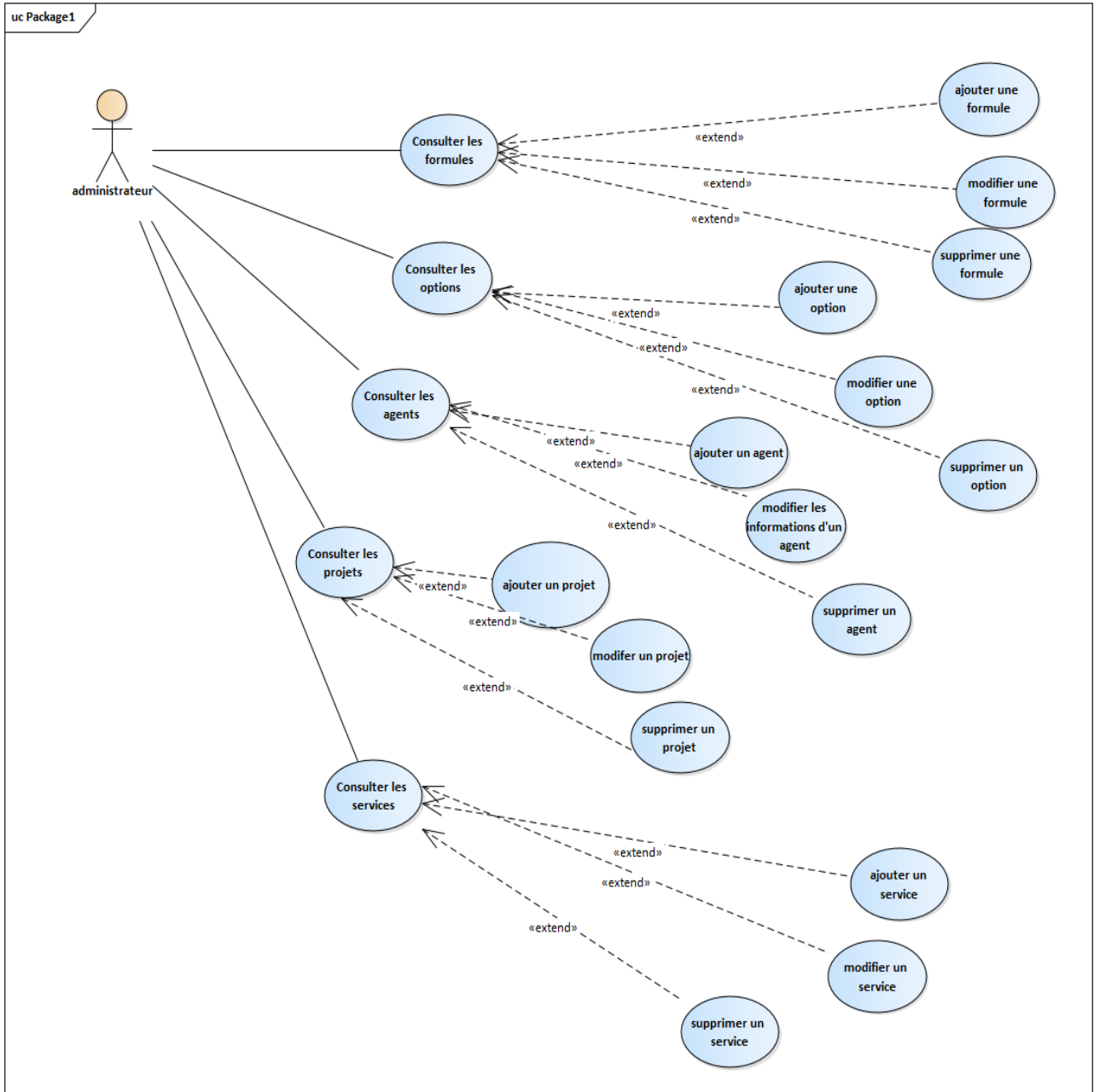


Figure 3: Diagramme de cas d'utilisation de l'administrateur partie 1

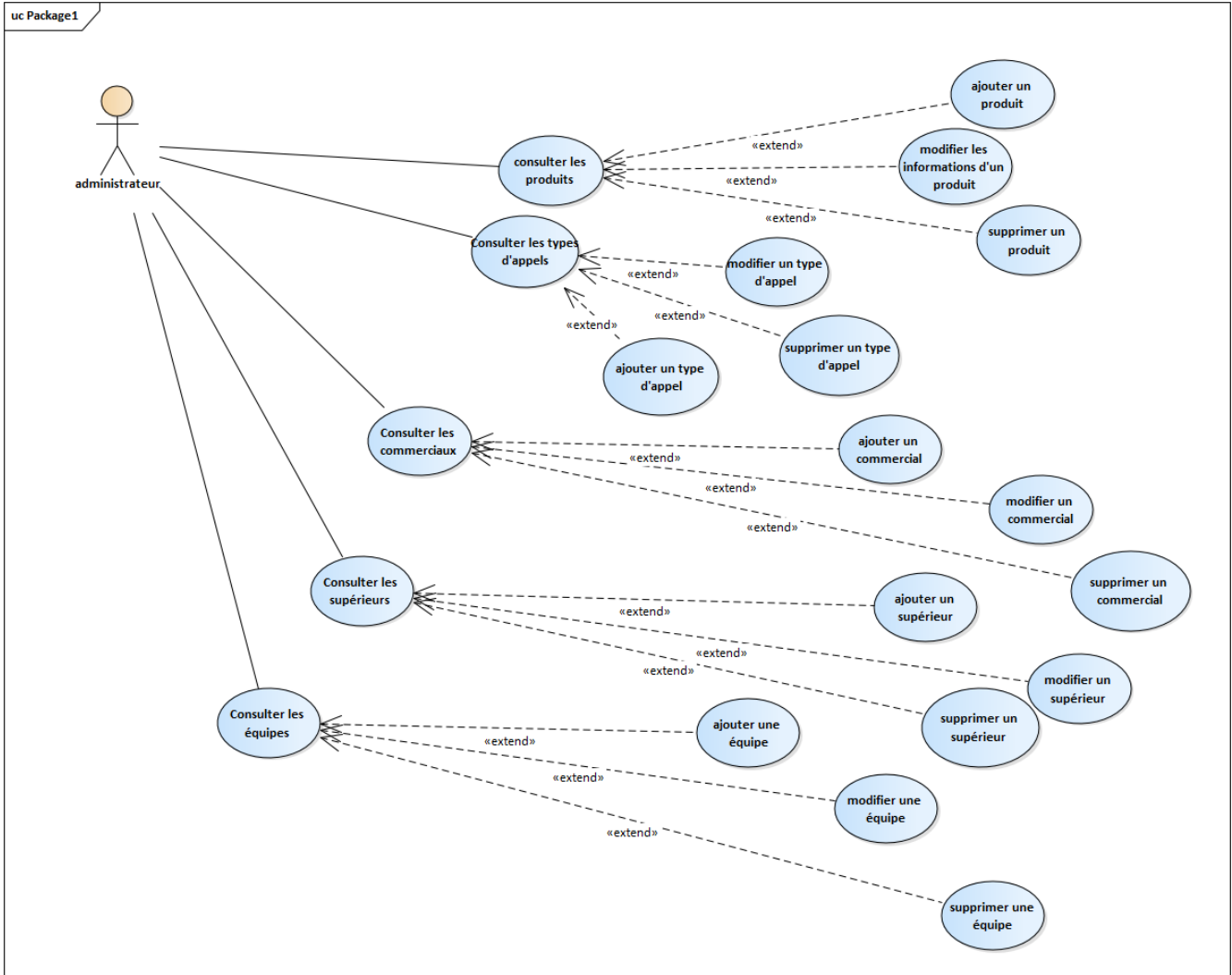


Figure 4: Diagramme de cas d'utilisation de l'administrateur partie 2

2.2.3 Diagramme d'activité d'ajout d'un agent

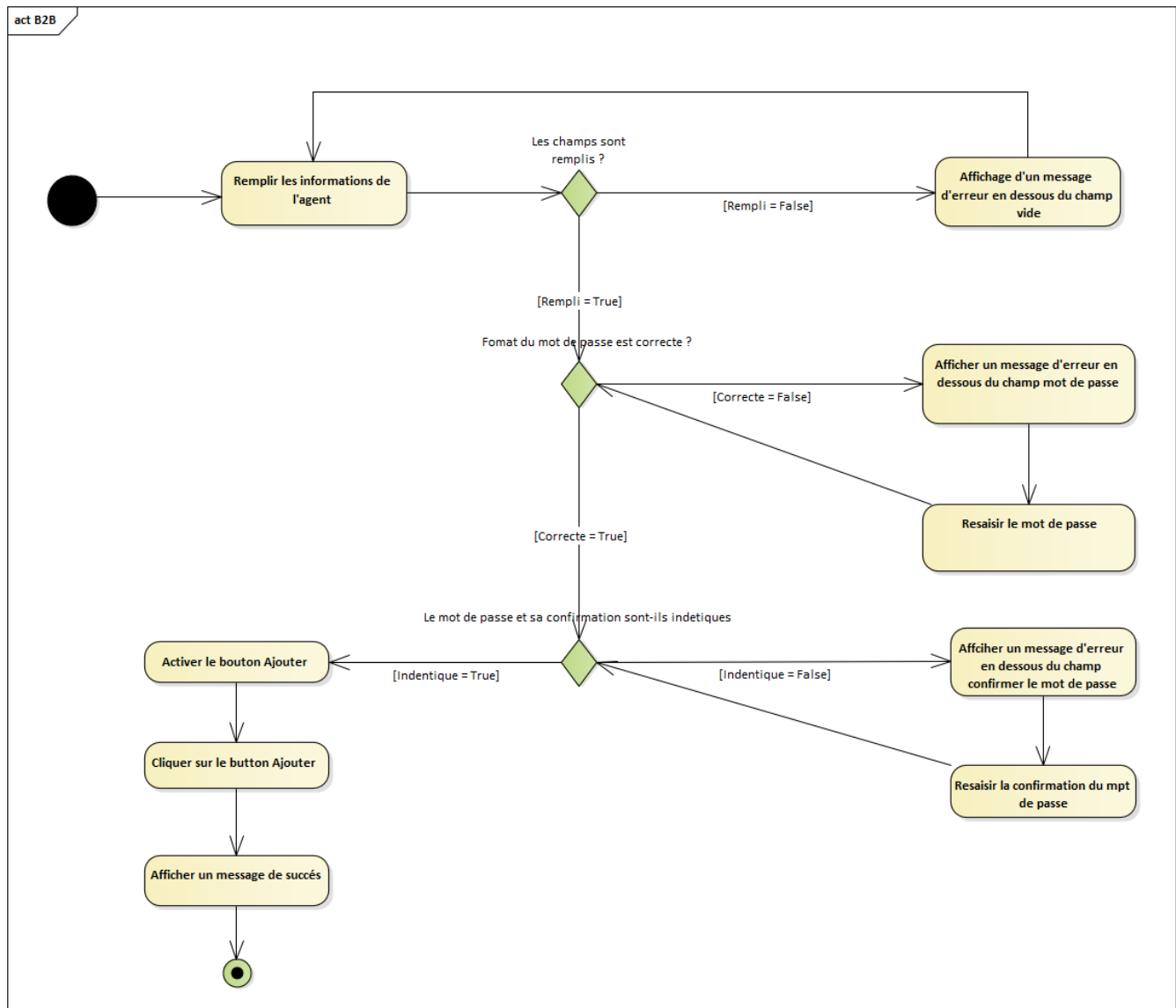


Figure 5: Diagramme d'activité d'ajout d'un agent

Tableau 1: Description du cas d'utilisation Ajouter un agent

Cas d'utilisation	Ajout d'un agent.
Résumé	Cas d'utilisation permettant à l'administrateur d'ajouter les informations concernant un agent.
Acteurs	Administrateur.

<p>Scénario nominal</p>	<ul style="list-style-type: none"> -L'administrateur remplit tous les champs. -L'administrateur respecte le format demander lors de la saisit du mot de passe. -Le mot de passe et la confirmation mot de passe sont identiques. -Le bouton Ajouter est activé. -L'administrateur enregistre les informations de l'agent en cliquant sur Ajouter. -Un message de succès est affiché.
<p>Scénario alternatif 1</p>	<ul style="list-style-type: none"> -L'administrateur entre les informations de l'agent. -Les champs ne sont pas tous remplis. -Un message d'erreur est affiché en dessous des champs vides. -Le bouton Ajouter est toujours désactivé.
<p>Scénario alternatif 2</p>	<ul style="list-style-type: none"> -L'administrateur entre les paramètres de connexion login et mot de passe. -Le format du mot de passe est incorrecte. -Un message d'erreur est affiché en dessous du champ mot de passe avec les restrictions qu'il faut respecter. -Le bouton Ajouter est toujours désactivé.
<p>Scénario alternatif 3</p>	<ul style="list-style-type: none"> -L'administrateur entre les paramètres de connexion login et mot de passe. -Le format du mot de passe est respecté mais la confirmation du mot de passe n'est pas identique au mot de passe. -Un message d'erreur est affiché en dessous du champ confirmer le mot de passe. -Le bouton Ajouter est toujours désactivé.

2.2.4 Diagramme d'activité d'ajout d'un appel

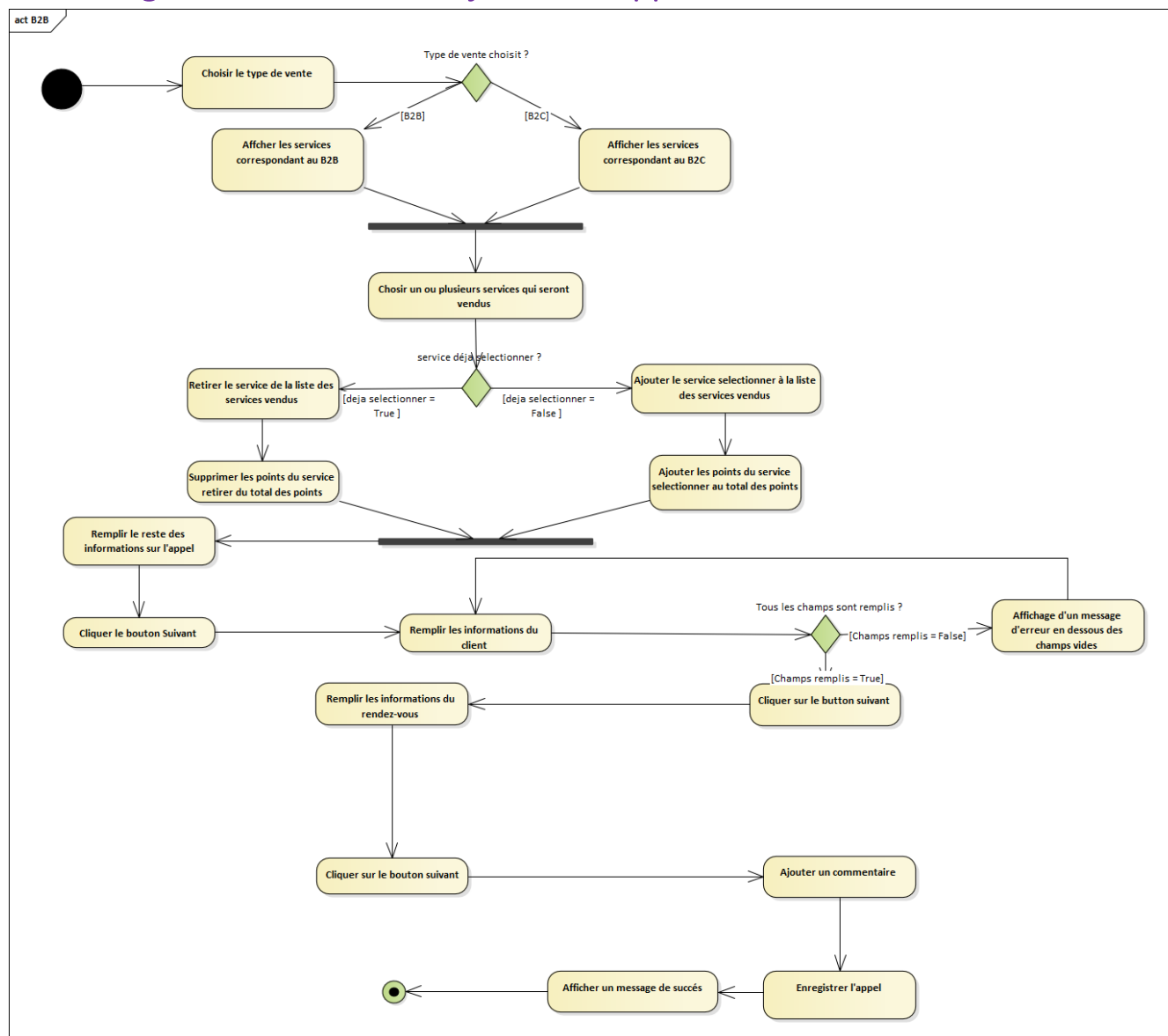


Figure 6: Diagramme d'activité d'ajout des informations d'un appel

Tableau 2 : Description du cas d'utilisation Ajouter un appel

Cas d'utilisation	Ajout d'un appel.
Résumé	Cas d'utilisation permettant à l'utilisateur d'ajouter les informations concernant un appel.
Acteurs	Agent, administrateur.

Scénario nominal	<ul style="list-style-type: none"> -L'utilisateur remplit tous les champs. -L'utilisateur enregistre les informations de l'appel. - Un message de succès est affiché.
Scénario alternatif 1	<ul style="list-style-type: none"> -L'utilisateur entre les informations du client. -Les champs ne sont pas tous remplis. -Un message d'erreur est affiché en dessous des champs vides. -Le bouton Suivant ne sera pas activé.

2.3 Modèle statique

2.3.1 Diagramme de classe

Le diagramme de classe est un schéma utilisé en génie logiciel pour présenter les classes et les interfaces des systèmes ainsi que les différentes relations entre celles-ci. Ce diagramme fait partie de la partie statique d'UML car il fait abstraction des aspects temporels et dynamiques.

- *Nous avons développé le diagramme de classe à partir du cahier des charges et de la description détaillée des cas d'utilisation.*

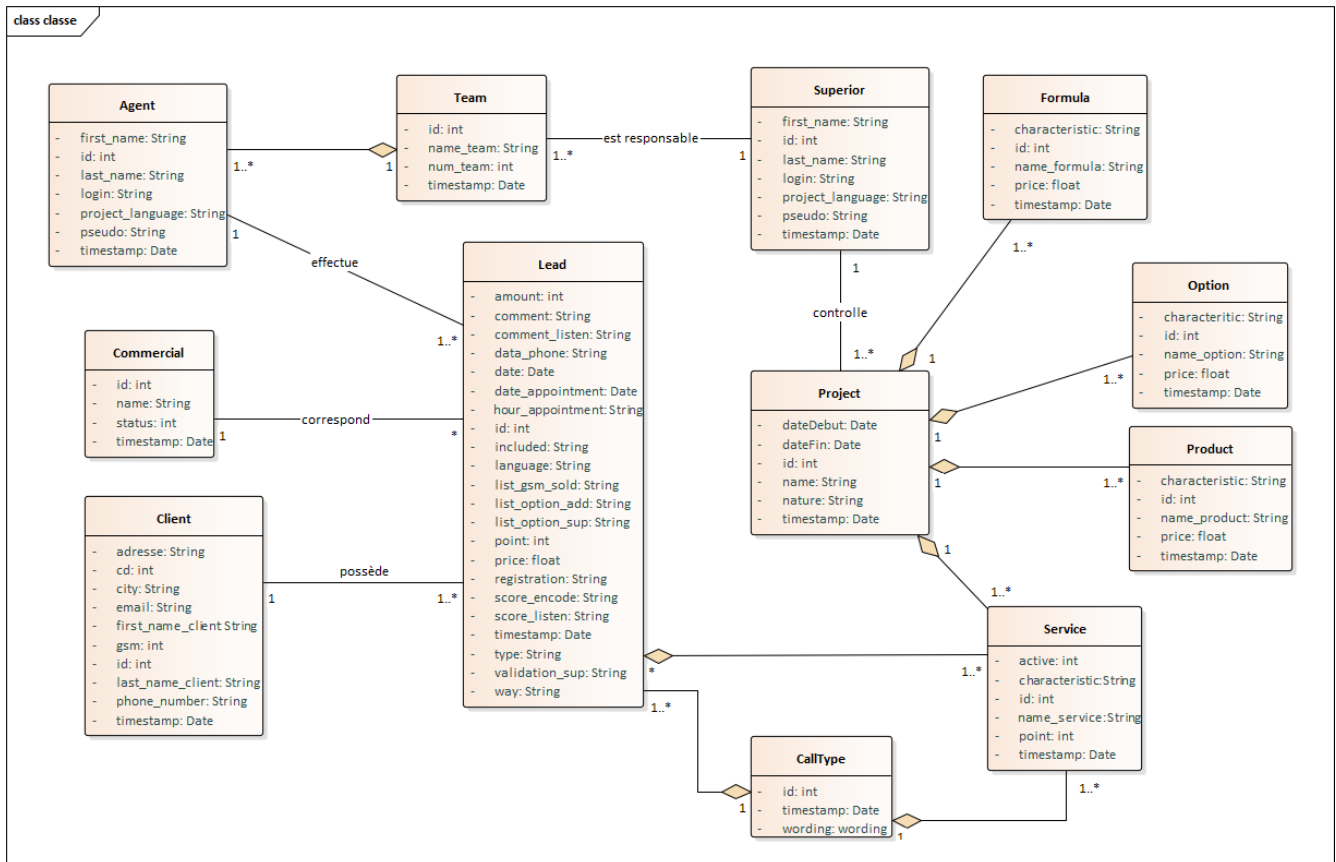


Figure 7 : Diagramme de classes

2.3.2 schémas de la base de données

Avant d’implémenter la BD, nous avons construit son schéma constitué des tables suivantes :

Superior (id , first_name , last_name , login , project_language , pseudo , timestamp)

Project (id , date_begin , date_end , name, nature , timestamp , #id_superior)

Team (id , name_team , num_team, timestamp, #id_superior)

Agent (id , first_name , last_name , login , project_language , pseudo , timestamp , #id_team)

Product (id , characteristic , name_product , price , timestamp , #id_project)

Option (id , characteristic , name_option , price , timestamp , #id_project)

Formula (id, characteristic , name_formula , price , timestamp , #id_project)

Call_Type (id, wording , timestamp)

Service (id, characteristic , active , name_service , point , timestamp , #id_project , #id_call_type)

Commercial (id , name , status , timestamp)

Client (id , adresse , cd , city , email , first_name_client , last_name_client , gsm , phone_number , timestamp)

Lead (id , amount , comment , comment_listen , data_phone , date , date_appointment , hour_appointment , included , language , list_gsm_sold , list_option_add , list_option_sup , point , price , registration , score_encode , score_listen , type , validation_sup , way , timestamp , #id_client , #id_commercial , #id_product , #id_call_type , #id_user)

Lead_Service (id , #id_lead , #id_service)

Timestamp : enregistre la date d'ajout ou la dernière date de modification de n'importe quel élément dans notre base de données.

2.3 Conception méthodique

Pour concevoir notre application nous nous sommes basés sur la méthode Agile.

La méthode Agile réside une plus grande implication du client et une meilleure réactivité des équipes. Ce manifeste prône en outre 4 valeurs fondamentales de la démarche :

L'équipe, soit des individus et des interactions, plutôt que des processus et des outils ;

L'application, c'est-à-dire des fonctionnalités opérationnelles plutôt que de la documentation exhaustive ;

La collaboration avec le client, plutôt que la contractualisation des relations ;

L'acceptation du changement, plutôt que le suivi d'un plan.

Les avantages de la méthode Agile :

- ❖ Satisfaire la clientèle en priorité.
- ❖ Accueillir favorablement les demandes de changement.
- ❖ Livrer le plus souvent possible des versions opérationnelles de l'application.
- ❖ Assurer une coopération permanente entre le client et l'équipe projet.
- ❖ Construire autour de personnes motivées.
- ❖ Privilégier la conversation en face-à-face.
- ❖ Mesurer l'avancement du projet en matière de fonctionnalité de l'application.
- ❖ Faire avancer le projet à un rythme soutenable et constant.
- ❖ Porter une attention continue à l'excellence technique et à la conception.
- ❖ Faire simple.
- ❖ Responsabiliser les équipes.
- ❖ Ajuster à intervalles réguliers son comportement et ses processus pour être plus efficace.

Le Fonctionnement de la méthode Agile :

La méthodologie Agile se base sur une idée simple. Planifier la totalité de votre projet dans les moindres détails avant de le développer est contre-productif.

Vous perdez du temps si vous organisez tous les aspects de votre projet en amont. Il est effectivement rare que tout se passe exactement comme prévu. Souvent, des aléas surviennent et vous forcent à revoir votre planification.

La méthode Agile recommande de se fixer des objectifs à court terme. Le projet est donc divisé en plusieurs sous-projets. Une fois l'objectif atteint, on passe au suivant, et ce jusqu'à l'accomplissement de l'objectif final. Cette approche est plus flexible. Puisqu'il est impossible de tout prévoir et de tout anticiper, elle laisse la place aux imprévus et aux changements.

Autre point important : la méthode Agile repose sur une relation privilégiée entre le client et l'équipe projet. Sa satisfaction étant la priorité, l'implication totale de l'équipe et sa réactivité face aux changements s'imposent. Le dialogue est privilégié. C'est le client qui valide chaque étape du projet. Il convient donc de prendre en compte

l'évolution de ses besoins. Des ajustements sont effectués en temps réel afin de répondre à ses attentes.

Avec l'approche Agile, rien n'est figé. L'équipe projet doit être capable de se remettre sans cesse en cause et de chercher continuellement à évoluer.

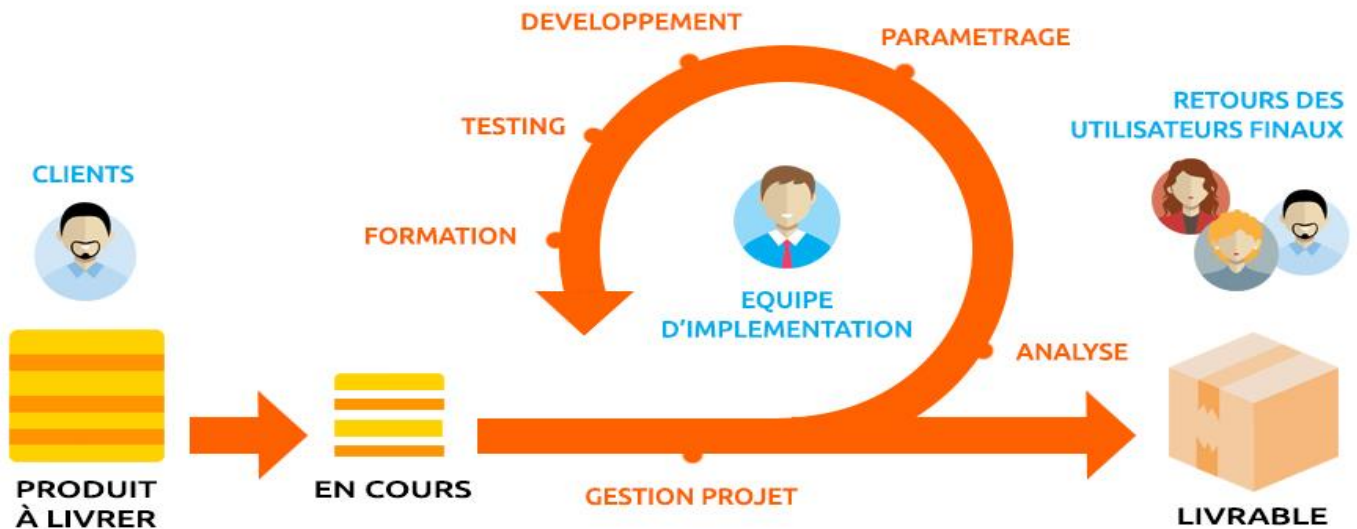


Figure 8: Explication simple de la méthode Agile

Conclusion

Dans ce chapitre, nous avons modélisé les vues fonctionnelles de notre système de gestion commande en se basant sur les diagrammes du langage UML à savoir : le diagramme de cas d'utilisation, le diagramme d'activités et le diagramme de classe. Le prochain chapitre sera dédié à la réalisation de notre application.

Chapitre 3 : Réalisation

3.1 Framework de développement

3.1.1 Pourquoi utilisé un Framework ?

Un Framework est un ensemble d'outils et de composants logiciels organisés conformément à un plan d'architecture et des patterns, l'ensemble formant ou promouvant un « squelette » de programme. Il est souvent fourni sous la forme d'une bibliothèque logicielle, et accompagné du plan de l'architecture cible du Framework.

Les avantages des Frameworks sont nombreux. En effet, un Framework est portable, de la part de son abstraction de la base de données et de la gestion générique du cache. Les temps de développement avec un Framework sont réellement plus courts.

Tous les outils essentiels sont déjà écrits. Le développement des applications sécurisées est facile.

Les Frameworks sont des outils communautaires et ont, par conséquent, des forums, des listes de diffusion et des canaux IRC pour les soutenir. De plus vu que les Frameworks sont largement déployés, la chance de trouver les correctifs des problèmes rencontrés est plus grande.

3.1.2 Angular Framework

Angular est un Framework côté client, open source, basé sur **TypeScript**, et co-dirigé par l'équipe du projet « Angular » à Google et par une communauté de particuliers et de sociétés.

Angular est une réécriture complète d'**AngularJS**, cadriciel construit par la même équipe. Il permet la création d'applications Web et plus particulièrement de ce qu'on appelle des « Single Page Applications » : des applications web accessibles via une page web unique qui permet de fluidifier l'expérience utilisateur et d'éviter les chargements de pages à chaque nouvelle action.

Le Framework est basé sur une architecture du type MVC et permet donc de séparer les données, le visuel et les actions pour une meilleure gestion des responsabilités. Un type d'architecture qui a largement fait ses preuves et qui permet une forte maintenabilité et une amélioration du travail collaboratif.



Figure 9: Logo du Framework Angular

3.1.2 AngularJS Framework

AngularJS est un Framework JavaScript libre et open source² développé par Google. Il permet de développer des pages web.



Figure 10: Logo du Framework AngularJS

3.1.3 Différence entre Angular et AngularJS

La différence la plus fondamentale entre les deux frameworks open-source est qu'Angular est basé sur Typescript alors que AngularJS est basé sur Javascript. Cela implique essentiellement qu'il y aura des différences dans leurs composants.

3.2 Environnement du développement

Pour mettre en place notre système, nous avons utilisé un environnement de développement qui a assuré le bon déroulement de la phase d'implémentation. Cet environnement comporte des outils matériels ainsi que logiciels suivants.

3.2.1 Environnement matériel

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

- Processeur Intel Core i7-3537U avec fréquence 2.5 GHz
- Quantité de mémoire vive 8 Go
- Capacité du disque dur 500 Go

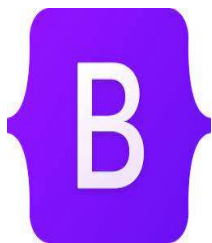
3.2.2 Environnement logiciel



Visual Code Studio : est un IDE que nous avons utilisé pour le développement du module Web App. Il s'agit d'un éditeur de code supportant les langages de programmation Web, comme l'HTML, le CSS, le Javascript et le Typescript.



HTML 5 : HyperText Markup Language 5, C'est un langage de balises utilisé pour structurer et donner du sens au contenu web. Par exemple : définir des paragraphes, titres et tables de données ou encore intégrer des images ou des vidéos dans une page.



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



CSS 3 : (Cascading Style Sheets 3) : Est un langage de règles de style utilisé pour mettre en forme le contenu HTML. Par exemple : en modifiant la couleur d'arrière-plan ou les polices, ou en disposant le contenu en plusieurs colonnes.



SCSS or SASS : (Syntactically awesome stylesheets) est un langage de script préprocesseur qui est compilé ou interprété en CSS (Feuilles de styles en cascades). SassScript est le langage de script en lui-même.



Photoshop : Est un logiciel de retouche, de traitement et de dessin assisté par ordinateur, lancé en 1990 sur MacOS puis en 1992 sur Windows. Édité par Adobe, il est principalement utilisé pour traitement des photographies numériques, mais sert également à la création ex nihilo d'images.



Angular : Est une plateforme de développement, construite sur TypeScript. En tant que plateforme, Angular comprend : Un cadre basé sur des composants pour la création d'applications web évolutives.



TypeScript : Est un langage de programmation conçu et développé par Microsoft en 2012. Il représente un "superset" de JavaScript. C'est-à-dire que la syntaxe est la même que celle JavaScript à quelques différences près



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



Spring : se définit comme étant un framework libre très riche, parmi les plus réputés au monde. Il permet de construire l'infrastructure d'une application Java et d'en faciliter le développement.



Postman : est un logiciel gratuit qui vous permet d'effectuer des requêtes API sans coder. Les requêtes prennent la forme suivante : Verbe HTTP + URI Version HTTP + Headers + Body facultatif.



GitLab : est une plateforme DevOps complète proposée sous la forme d'une application unique. Elle révolutionne le développement, la sécurité, l'exploitation et la collaboration entre les équipes. Créez, testez et déployez des logiciels plus rapidement en n'utilisant qu'une seule solution.

3.3 Présentation des interfaces graphiques

3.3.1 Le menu

Pour une navigation simple et efficace dans nos interfaces nous avons mis à disposition le menu suivant :



Figure 11: Menu de navigation partie 1



Figure 12: Menu de navigation partie 2

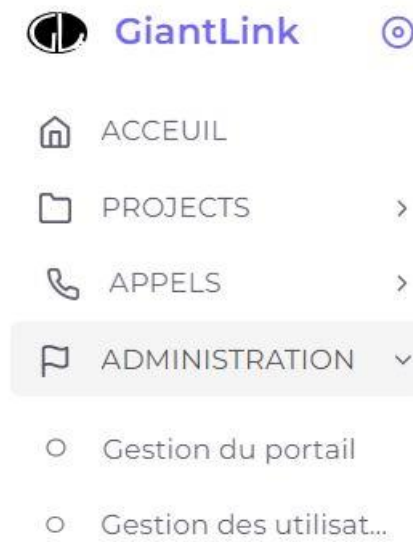


Figure 13: Menu de navigation partie 3

3.3.2 L'accueil

Interface « Accueil » : Dans cette interface on trouve les différentes informations sur les ventes de l'agent connecté, les points qu'il a cumulé et finalement le classement des meilleurs agents.

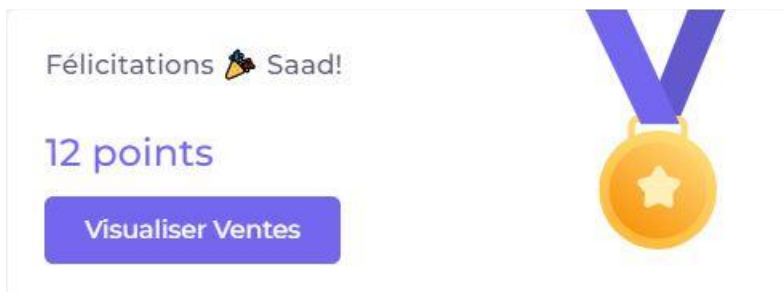
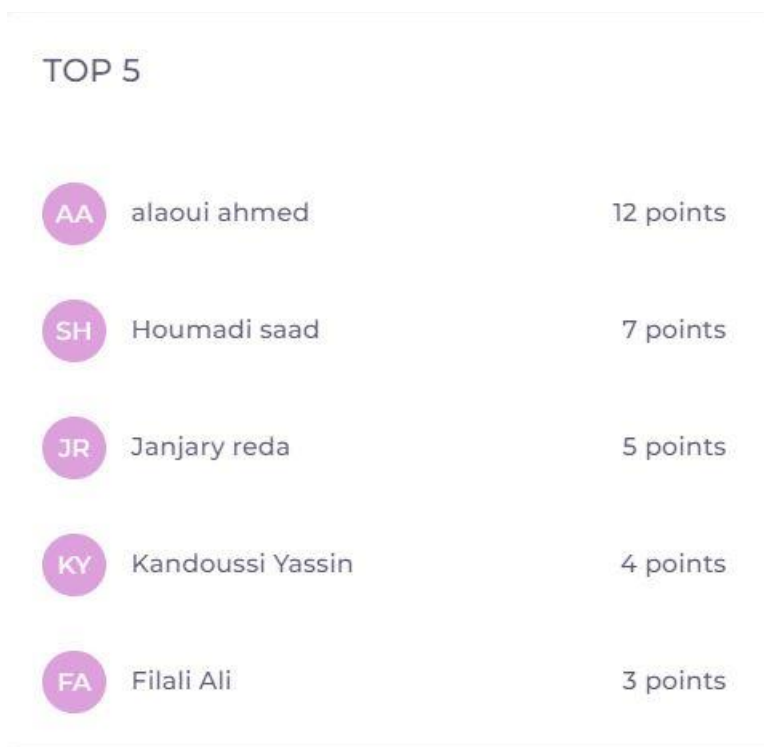


Figure 14: Les points cumulés par l'agent connecté

En cliquant sur le bouton « Visualiser les ventes », l'interface de gestion d'appel sera ouverte.











TOP 5		
AA	alaoui ahmed	12 points
SH	Houmadi saad	7 points
JR	Janjary reda	5 points
KY	Kandoussi Yassin	4 points
FA	Filali Ali	3 points

Figure 15: Dashboard d'accueil montrant le classement des agents selon les points cumulés

3.3.3 Gestion des projets

The screenshot shows a web interface for project management. At the top left is a purple button labeled '+ Ajouter Projet'. Below it is the title 'Liste des Projets' and a search input field labeled 'Search'. To the right of the search field is a dropdown menu labeled 'Afficher' with the number '4' selected. The main content is a table with the following data:

Nom	Nature	Date Début	Date Fin	Actions
Projet1	Prise de rendez_vous	2022-06-16	2022-06-16	 
Projet2	Vente	2022-06-21	2022-06-30	 
Projet3	Vente	2022-06-21	2022-07-03	 
Projet4	Prise de rendez_vous	2022-06-15	2022-07-09	 

At the bottom of the table is a pagination control showing '« « 1 2 3 » »' with the number '1' highlighted. On the right side, there is a dropdown menu labeled 'Afficher' with the number '4' selected. Four callout boxes with arrows point to these elements: 'Bouton pour Ajouter un nouveau Projet' points to the '+ Ajouter Projet' button; 'Pagination' points to the '1' in the pagination control; 'Rechercher les projets par nom ou/et nature' points to the 'Search' input field; 'Bouton modification / suppression d'un projet' points to the edit/delete icons in the 'Actions' column; and 'Sélectionner le nombre de lignes à afficher' points to the 'Afficher 4' dropdown.

Bouton pour Ajouter un nouveau Projet

Pagination

Rechercher les projets par nom ou/et nature

Bouton modification / suppression d'un projet

Sélectionner le nombre de lignes à afficher

Figure 16: Interface de gestion des projets

Interface « Gestion des projets » : Cette interface est accessible seulement par les administrateurs. Dans cette interface on trouve la liste des projets disponibles dans la base de données, qu'on pourra les modifier ou/et supprimer directement en cliquant sur les émoticons affichés à la droite.

En cliquant sur le bouton « Créer Projet » une interface de création d'un projet se présente :

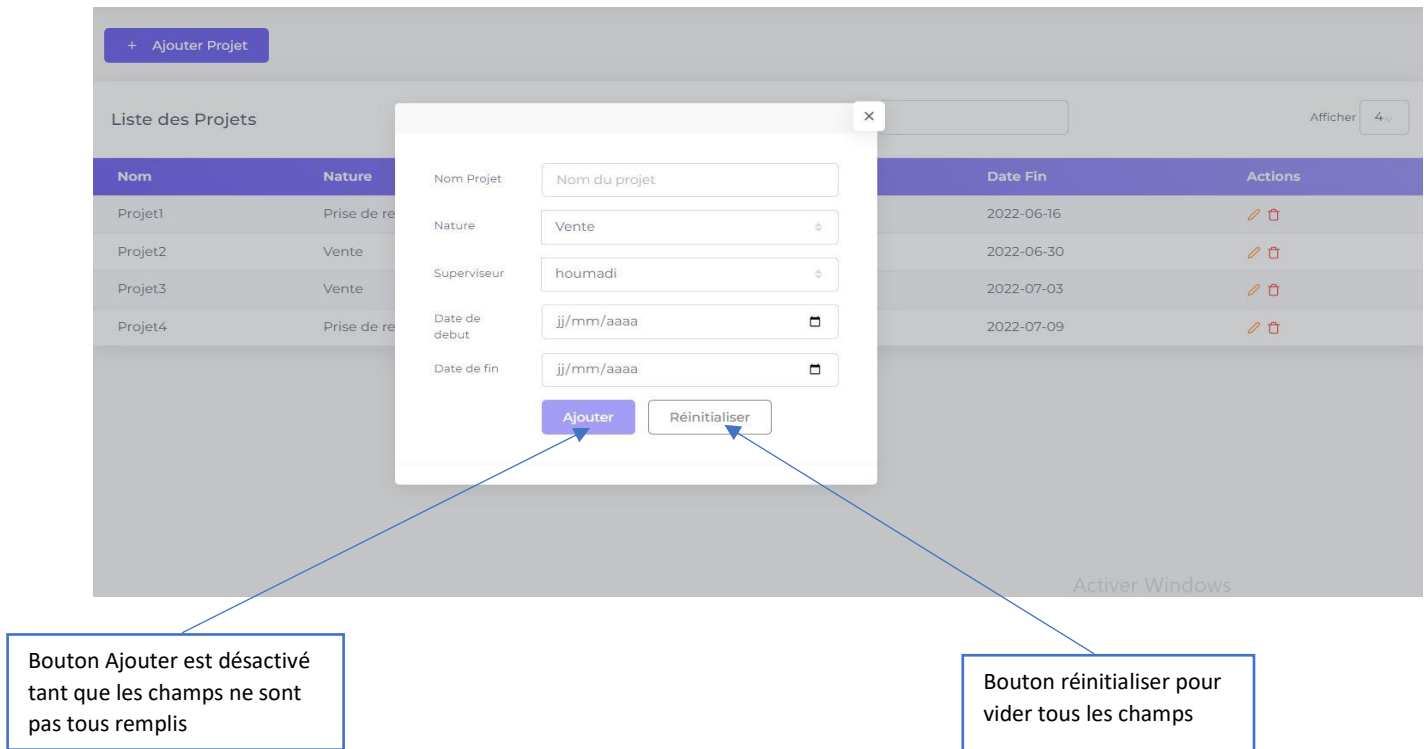


Figure 17: Interface de création d'un projet

Interface de création d'un projet : Cette interface permet à l'administrateur d'ajouter un nouveau projet, si les données sont correctes un message de succès est affiché comme le montre la figure ci-dessous.

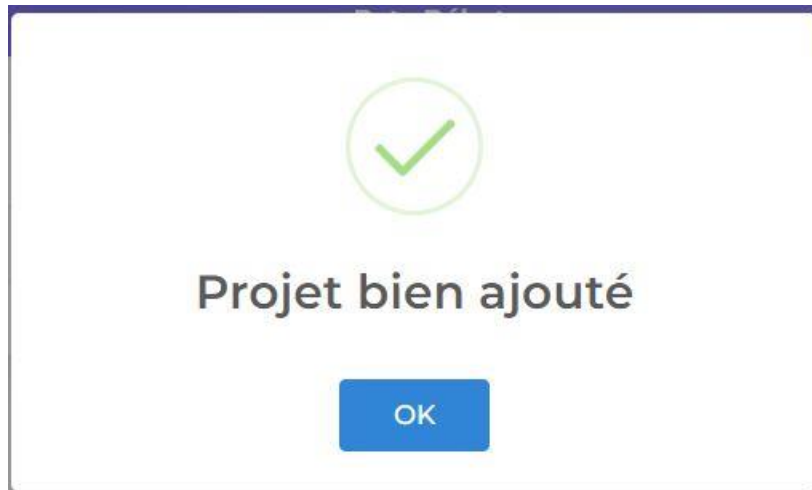


Figure 18: Message de succès après l'ajout d'un projet

En cliquant sur l'émoticône jaune en forme de stylo dans la même ligne du projet que vous voulez modifier, une interface de modification d'un projet se présente :

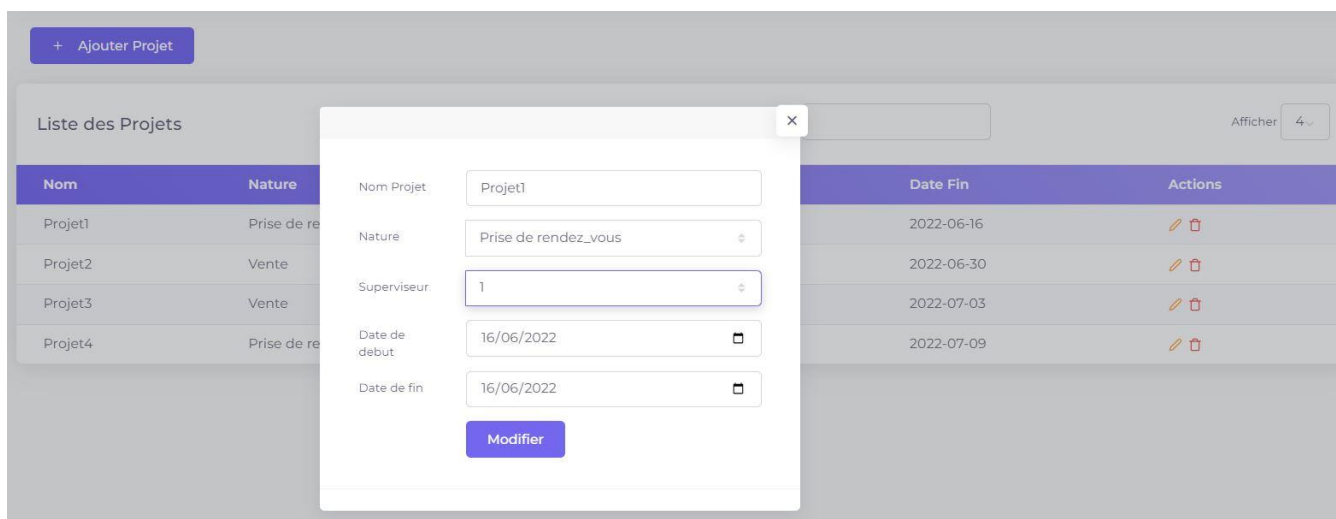


Figure 19: Interface de modification d'un projet

Interface de modification d'un projet : Cette interface permet à l'administrateur de modifier le projet en remplissant les informations dans les champs respectifs. Après la modification des champs désirés un message de confirmation apparaît, comme le montre les figures ci-dessous.

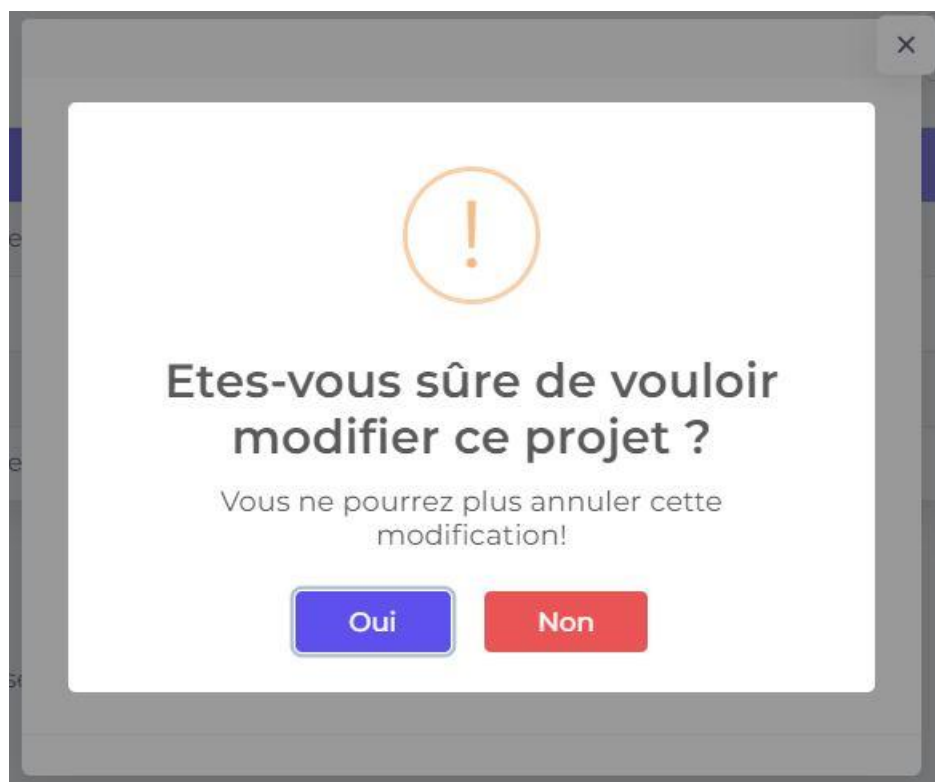


Figure 20: Message de confirmation lors d'une modification d'un projet

En cliquant sur le bouton Oui, un message de succès apparaît :

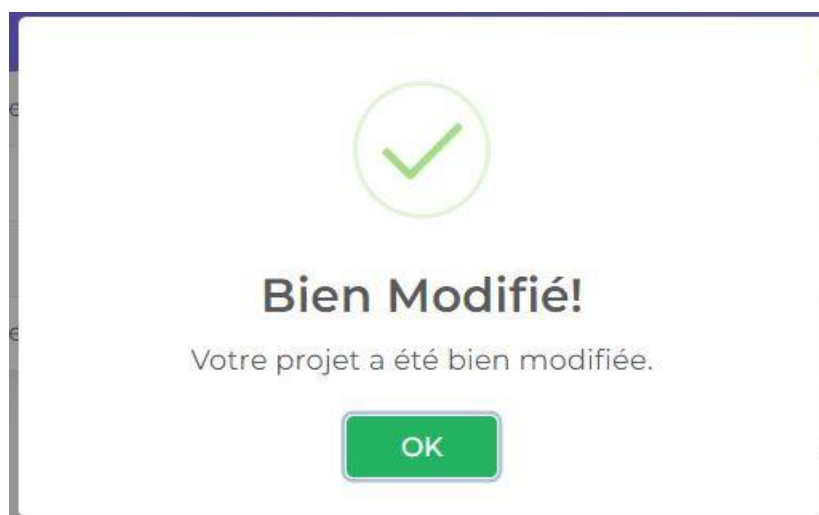


Figure 21: Message de succès après la modification d'un projet

En cliquant sur l'émoticône rouge en forme de poubelle dans la même ligne du projet que vous voulez supprimer, les mêmes messages de confirmation et de succès apparaîtront, comme les montrent les figures ci-dessous :

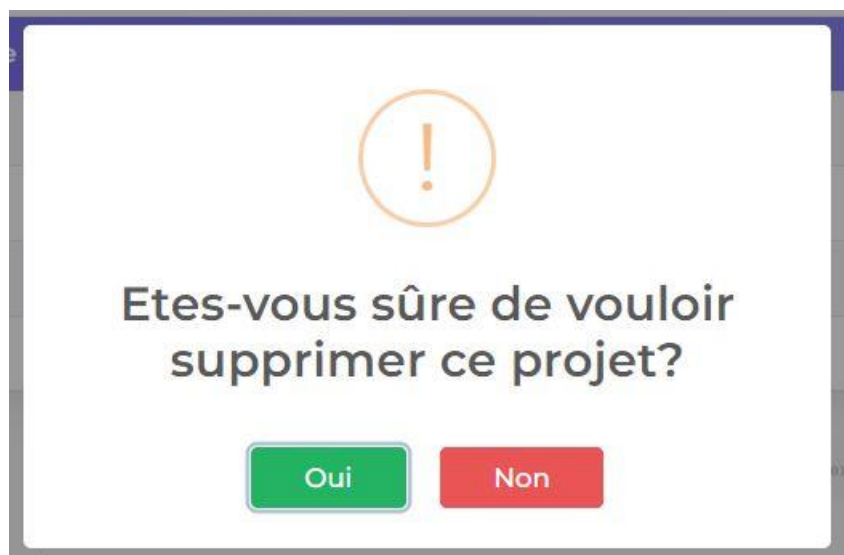
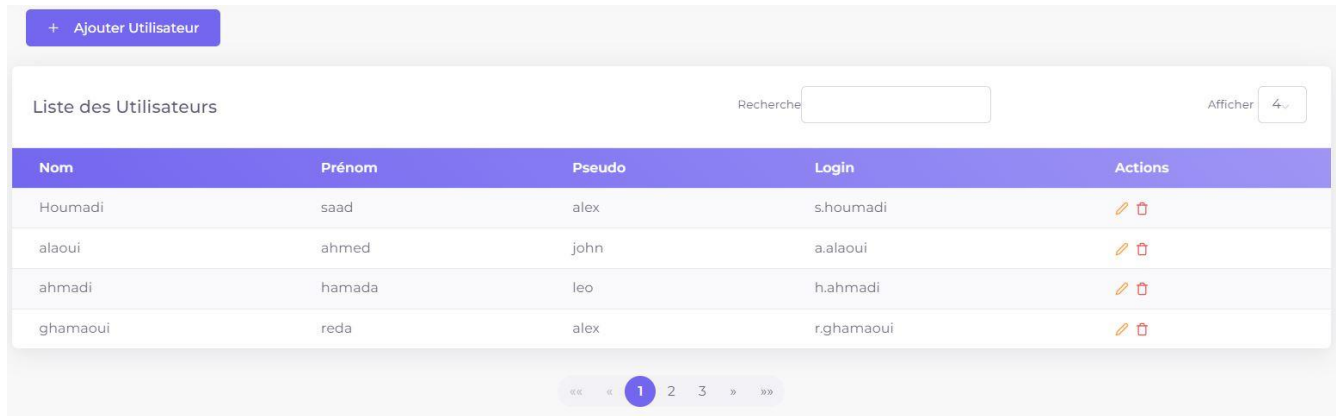


Figure 22: Message de confirmation lors d'une suppression d'un projet



Figure 23 : Message de succès après la suppression d'un projet

3.3.4 Gestion des agents



The screenshot shows a web interface for managing users. At the top left, there is a blue button labeled '+ Ajouter Utilisateur'. Below it, the title 'Liste des Utilisateurs' is displayed. To the right of the title is a search bar labeled 'Recherche' and a button labeled 'Afficher' with the number '4' next to it. The main content is a table with five columns: 'Nom', 'Prénom', 'Pseudo', 'Login', and 'Actions'. The table contains four rows of user data. Each row has a pencil icon and a trash can icon in the 'Actions' column. At the bottom of the table, there is a pagination control showing '1' selected, with '2' and '3' also visible.









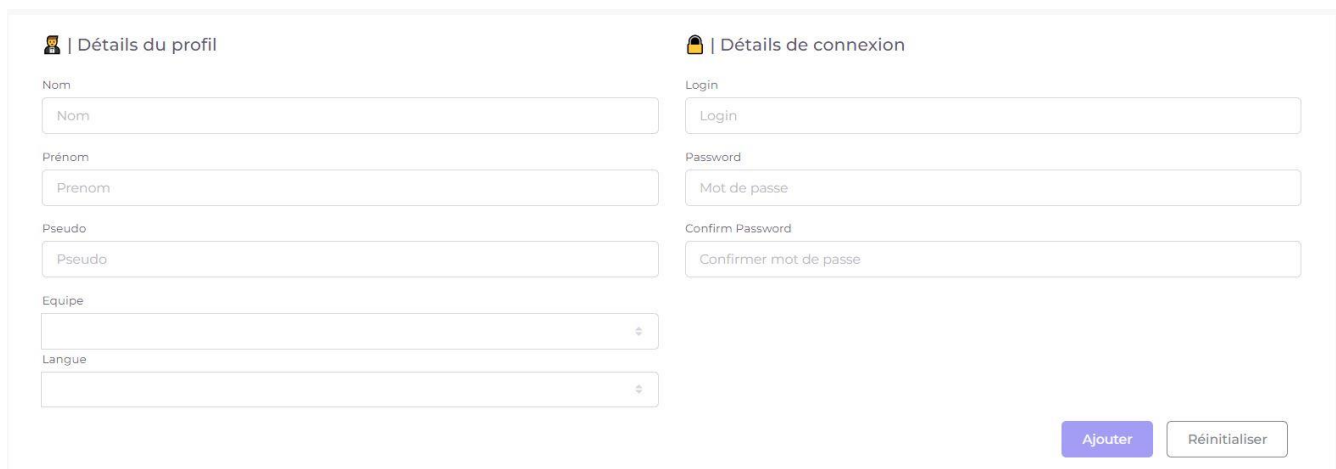
Nom	Prénom	Pseudo	Login	Actions
Houmadi	saad	alex	s.houmadi	 
alaoui	ahmed	john	a.alaoui	 
ahmadi	hamada	leo	h.ahmadi	 
ghamaoui	reda	alex	r.ghamaoui	 

Figure 24: Interface de gestion des agents

Interface « Gestion des agents » : Cette interface est accessible seulement par les administrateurs. Dans cette interface on trouve la liste des agents, qu'on pourra les modifier ou/et supprimer directement en cliquant sur les émoticons affichés à la droite.

En cliquant sur le bouton « Ajouter Utilisateur » une interface d'ajout d'un utilisateur se présente :



The screenshot shows a form for adding a user, divided into two sections: 'Détails du profil' and 'Détails de connexion'. The 'Détails du profil' section has fields for 'Nom', 'Prénom', 'Pseudo', 'Equipe', and 'Langue'. The 'Détails de connexion' section has fields for 'Login', 'Password', and 'Confirm Password'. At the bottom right, there are two buttons: 'Ajouter' (blue) and 'Réinitialiser' (grey).

Figure 25 : Interface d'ajout d'un utilisateur

Interface d'ajout d'un utilisateur : Cette interface permet à l'administrateur d'ajouter un nouvel utilisateur, si les données sont correctes un message de succès est affiché, sinon un message d'erreur apparaît, comme le montre les figures ci-dessous.

The screenshot shows a registration form with two columns: 'Détails du profil' and 'Détails de connexion'. The 'Détails du profil' column contains fields for 'Nom', 'Prénom', 'Pseudo', 'Equipe', and 'Langue'. The 'Détails de connexion' column contains fields for 'Login', 'Password', and 'Confirm Password'. Red error messages are displayed below the 'Nom', 'Prénom', and 'Pseudo' fields, stating 'Ce champ est obligatoire!'. The 'Ajouter' button is highlighted in blue.

Figure 26: Message d'erreur lorsqu'un champ ou plusieurs sont vides

The screenshot shows the same registration form as Figure 26. The 'Password' field now contains '.....' and has a red error message below it: 'Doit contenir au moins un nombre et une letter miniscule et majuscule, et au moins 8 caractères ou plus.' The 'Ajouter' button is highlighted in blue.

Figure 27: Message d'erreur lorsque la forme du mot de passe est incorrecte

The image shows a web interface with two main sections: 'Détails du profil' (Profile Details) and 'Détails de connexion' (Login Details). The 'Détails du profil' section contains input fields for 'Nom' (Name), 'Prénom' (First Name), 'Pseudo' (Nickname), 'Equipe' (Team), and 'Langue' (Language). The 'Détails de connexion' section contains input fields for 'Login' and 'Password', followed by a 'Confirm Password' field. A red error message is displayed below the 'Confirm Password' field: 'Mot de passe et confirmer mot de passe ne sont pas les mêmes.' (Password and confirm password are not the same). At the bottom right, there are two buttons: 'Ajouter' (Add) and 'Réinitialiser' (Reset).

Figure 28: Message d'erreur lorsque le mot de passe et confirmer le mot de passe ne sont pas identiques

•**NB** : Le reste des interfaces des gestions que nous présenterons après sont faites avec les mêmes principes.

Conclusion Et Perspectives

Ce projet de fin d'études, réalisé au sein de la société GiantLink, avait pour objectif la réalisation d'un système de gestion de commande. Ce système vise à automatiser toutes les gestions nécessaires au centre d'appel GiantLink.

Notre travail a consisté, dans un premier temps, à étudier les besoins du centre d'appel et essayer de mettre un plan simple et efficace pour optimiser au maximum possible notre application web. Dans une deuxième étape, nous avons abordé l'analyse et la modélisation des besoins fonctionnels à travers quelques diagrammes UML.

Nous avons conçu et développé, par la suite, les interfaces graphiques servant à réaliser les fonctionnalités appropriées du système en utilisant une large gamme d'outils web tels que Le Framework Angular avec Spring Boot, TypeScript, SCSS ... Ce stage nous a permis, d'approfondir nos connaissances et compétences en développement web et en conception et modélisation UML.

Dans un délai relativement court (deux mois), nous avons pu concevoir et réaliser un système d'information important caché derrière une application web simple et facilement exploitable. Il est pour l'instant prématuré de dire qu'il répond à toutes les attentes des utilisateurs, mais tous les tests que nous avons effectués confirment que notre système répond parfaitement aux exigences et besoins décrits dans le projet.

Malgré la réalisation de tous les objectifs fixés au début du stage, il reste des améliorations qui peuvent être envisagées dans le futur tels que :

- ✓ L'amélioration du design de nos IHMs.
- ✓ Effectuer le maximum de tests de notre application afin de recenser et corriger d'éventuelles erreurs.

Webographie

Documentation:

- ✓ Angularlr: <https://angular.io> Date de consultation *Le 13/06/2022*
- ✓ Spring Boot : <https://spring.io> Date de consultation *Le 26/05/2022*
- ✓ TypeScript : <https://www.typescriptlang.org> Date de consultation *Le 03/06/2022*
- ✓ MySQL : <https://www.mysql.com/fr> Date de consultation *Le 20/05/2022*