



**MINISTRE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA
RECHERCHE SCIENTIFIQUE**

UNIVERSITE DE SOUSSE

**INSTITUT SUPERIEUR D'INFORMATIQUE
ET DES TECHNIQUES DE COMMUNICATION – HAMMEM SOUSSE**



RAPPORT DE PROJET DE FIN D'ETUDES

Présenté en vue de l'obtention du diplôme de Licence Appliquée en
Technologies des Réseaux Informatiques

**Développement d'une application web de gestion des
rendez-vous chez les salons de coiffure**

Elaboré par :

IhebBelhaj

Chiheb Eddine Louhichi

Supervisé par :

Mr.HaythemSaoudi (ISTICOM)

Mr.Hassine El Arbi (VEGANET)

Société d'accueil : VEGANET



Année Universitaire : 2020-2021

**MINISTRE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA
RECHERCHE SCIENTIFIQUE**

UNIVERSITE DE SOUSSE

**INSTITUT SUPERIEUR D'INFORMATIQUE ET DES
TECHNIQUES DE COMMUNICATION – HAMMEM SOUSSE**

المعهد العالي للإعلامية وتقنيات الاتصال بحمامسوسة

RAPPORT DE PROJET DE FIN D'ETUDES

Présenté en vue de l'obtention du diplôme de Licence Appliquée en
Technologies des Réseaux Informatiques

Réalisé par :

IhebBelhaj

Chiheb Eddine Louhichi

Superviseur : HaythemSaoudi

Date :....

Signature :....

Encadrant : Hassine El Arbi

Date :....

Signature :....

Année Universitaire : 2020-2021

Dédicaces

Cette étude est entièrement dédiée à nos parents bien-aimés, sans lesquels nous ne serons pas les hommes que nous sommes aujourd'hui, qui fournissent continuellement leur soutien moral, spirituel, émotionnel et financier. Merci de toujours donner la priorité à vos enfants. Merci à nos amis et nos collègues pour leur encouragement, leurs conseils et pour tous les bons moments qu'on a vécus ensemble. J'espère que notre amitié durera éternellement.

Chiheb&Iheb

Remerciements

Au terme de ce travail, nous adressons nos sincères remerciements à M.HaythemSaoudi notre superviseur à l'ISITCOM, pour son assistance, sa disponibilité, ces conseils judicieux lors de la réalisation de ce projet et son aide à l'aboutissement de la bonne organisation de ce rapport.

Un grand merci à M.Hassine El Aarbi, directeur général de Veganet, de nous avoir accordé cette opportunité d'entreprendre ce projet au sein de l'équipe de Veganet.

Nous tenons à remercier les membres du jury pour avoir accepté d'assister à ce modeste travail.

Enfin, je ne veux pas oublier tous ceux qui ne cessent de m'encourager de près ou de loin.

Sommaire

Introduction générale.....	9
Chapitre 1 : Présentation générale	10
Introduction :.....	11
I. Présentation de l'organisme d'accueil :	11
II. Cadre de projet.....	12
1) Présentation du projet :	12
2) Motivation :	12
3) Travail demandé:.....	12
4) Etude et critique de l'existant:	13
5) Cycle de vie de l'application :	13
6) Planification du projet :	14
Conclusion :	14
Chapitre 2 : Analyse et spécification des besoins	15
Introduction :.....	16
I. Spécification des besoins fonctionnels et non fonctionnel :.....	16
1) Spécification des besoins fonctionnels :.....	16
2) Spécification des besoins non fonctionnels :	16
3) Identification des acteurs.....	17
II. Diagrammes des cas d'utilisation.....	18
1) Diagramme de cas d'utilisation global :	18
2) Raffinement des cas d'utilisation :	19
Conclusion :	25
Chapitre 3 : Conception.....	26
Introduction :.....	27
I. Diagramme des classes :	27
1) Définition :	27
2) Modèle logique des données :	29
II. Diagrammes des séquences :	31
1) Définition :	31
2) Diagramme de séquence d'authentification :	32

3) Diagramme de séquence de la recherche du partenaire :	33
4) Diagramme de séquence de prise de rendez-vous :	34
5) Diagramme de séquence de création compte client :	35
Conclusion :	35
Chapitre 4 : Implémentation	36
Introduction :	37
I. Environnement de travail :	37
1) Environnement matériel :	37
2) Environnement logiciel :	37
II. Développement :	41
1) Les Frameworks utilisé :	41
2) Les interfaces graphiques :	42
Conclusion :	47
Conclusion générale et perspectives	48

Liste des figures

Figure 1: Logo de VEGANET	11
Figure 2: Cycle de vie de l'application	13
Figure 3: Fonctionnement de JWT.....	17
Figure 4: Diagramme des cas d'utilisation global.....	19
Figure 5: Diagramme de cas d'utilisation raffiné « Gérer les partenaires »	19
Figure 6: Diagramme de cas d'utilisation raffiné « chercher partenaire ».....	20
Figure 7: Diagramme de cas d'utilisation raffiné « prendre rendez-vous».....	22
Figure 8: Diagramme de cas d'utilisation raffiné « Gérer les prestations »:	23
Figure 9: Diagramme de cas d'utilisation raffiné « Gérer les rendez-vous »:	24
Figure 10: Diagramme des classes	28
Figure 11: Diagramme de séquence d'authentification.....	32
Figure 12: Diagramme de séquence de la recherche du partenaire	33
Figure 13: Diagramme de séquence de prise de rendez-vous	34
Figure 14: Diagramme de séquence de création compte client.....	35
Figure 15: Des interactions entre les différentes couches de MVC	40
Figure 16: Interface d'accueil.....	43
Figure 17: Interface login client	43
Figure 18: Interface sign up client.....	44
Figure 19: Interface de la recherche de partenaire (type activité: coiffeur)	44
Figure 20: Interface liste des rendez-vous.....	45
Figure 21: Interface liste des clients.....	45
Figure 22: Interface des horaires	46
Figure 23: Interface des prestations.....	46
Figure 24: Interface des details prestations	47

Liste des tableaux

Tableau 1 : Détails de la société.....	11
Tableau 2: Planification	14
Tableau 3: Identification des acteurs et les cas d'utilisation.....	18
Tableau 4: Description du cas d'utilisation « Gérer les partenaires».....	20
Tableau 5: Description de cas d'utilisation « chercher partenaire »	21
Tableau 6: Description de cas d'utilisation « prendre un rendez-vous »	22
Tableau 7: Description de cas d'utilisation « Gérer les prestations»	23
Tableau 8: Description de cas d'utilisation « Gérer les rendez-vous »	24
Tableau 9: Table Partenaire.....	29
Tableau 10: Table client.....	29
Tableau 11: Table horaires_ouverture.....	30
Tableau 12: Table rendez_vous.....	30
Tableau 13: Table Prestation.....	30
Tableau 14: Table Exécutant.....	31
Tableau 15: Table Avis	31

Introduction générale

Etant donné la forte croissance du marché des applications Web, aujourd'hui, le développement d'application web intéresse énormément d'utilisateurs et il est reconnu dans la plupart des domaines y compris les domaines de la beauté. En effet, les logiciels et les applications web dans le domaine de la beauté connaissent actuellement un essor important. Leurs utilisations se multiplient et ces produits peuvent être très variés.

C'est dans ce contexte, que s'intègre notre projet de fin d'étude effectué au sein de la société Veganet et qui consiste à réaliser un système de gestion des rendez-vous pour les centres de beauté intitulé «Hajjami».

Nous sommes appelé à concevoir, développer et intégrer un système incluant des interfaces claires et faciles à utiliser afin de mettre en place une solution web pour rapprocher les salons de ses clients et faciliter le processus de prise des rendez-vous.

Le présent rapport est organisé en quatre chapitres :

Dans le premier chapitre de ce rapport intitulé « Présentation générale », nous présentons le contexte de notre stage, le cadre général de projet pour identifier les différents éléments formant notre système.

Le deuxième chapitre intitulé «Analyse et spécification des besoins », on détermine les besoins fonctionnels et non fonctionnels ainsi que la présentation de diagramme de cas d'utilisation général et raffiné.

Le troisième chapitre « Conception » présente la partie de l'étude conceptuelle dans laquelle nous identifierons les acteurs du système à l'aide du langage de modélisation UML et qui sera matérialisée par l'élaboration des diagrammes nécessaires.

Le quatrième chapitre du rapport expose les outils matériels et logiciels utilisés dans notre système et aussi les interfaces de certaines fonctionnalités mises au point.

Enfin, Nous clôturons ce rapport par une conclusion générale ainsi par la proposition de quelques perspectives sur lesquelles peut s'ouvrir le présent travail.

Chapitre 1 : Présentation générale

Chapitre 1 : Présentation générale

Introduction :

Cette étude fera l'objet du premier chapitre qui va s'intéresser à la présentation générale de l'entreprise d'accueil, la présentation du projet et la spécification des besoins fonctionnels et non fonctionnels de ce système.

I. Présentation de l'organisme d'accueil :

VEGANET est une société de services en ingénierie informatique partiellement exportatrice basée en Tunisie (Tunis,Sousse) anciennement dénommée MAPPNET fondée en 2012.

Chez VEGANET, chaque projet est analysé avec une attention particulière afin de développer des sites fiables qui répondent aux besoins des clients.



Figure 1: Logo de VEGANET

VEGANET travaille dur pour atteindre sa réputation, et si elle travaille pour les petites ou les grandes entreprises, elle applique les mêmes niveaux de pensée, de soins et d'attention aux détails.

Site web	http://www.veganet.com.tn
La date de création	2012
Type	Société civile/Société commerciale/Autres types de sociétés.
Secteurs	Technologies et services de l'information
Les principaux axes de développement	Mobile, Web, Outsourcing, Développement Spécifique, CRM, ERP, Conseil et DevOps.
Siège social	Tunis, Le Kram.

Tableau 1 : Détails de la société

II. Cadre de projet

Ce stage s'inscrit dans le cadre du projet de fin d'étude présenté en vue de l'obtention du diplôme de Licence Appliquée en Réseaux Informatique. Le présent projet est élaboré au sein de la société VEGANET

1) Présentation du projet :

L'origine de ce projet était une simple idée qui est innovante dans la Tunisie, c'est une application web qui peut nous aider à trouver un salon de coiffure, institut de beauté ou bien un barbier et tous ce qui concerne le domaine de beauté. Le discours marketing, les modes de tarification pratiqués ou l'image de marque qui sera développée ciblera principalement une clientèle relativement jeune (25-35 ans) utilisant les supports digitaux de manière intensive.

2) Motivation :

Durant ce stage, il nous a été demandé de faire la conception et le développement d'une application web qui permet de trouver les centres de beauté en cas de besoin.

Pendant toute la période de ce projet, nous avons une grande opportunité de découvrir des nouvelles technologies, des nouvelles techniques de programmation, des logiciels et d'améliorer nos connaissances. De plus, ce stage nous permet de s'approcher du monde réel de ce domaine et s'intégrer dans la vie des entreprises et à bien connaître comment traiter les services demandé par le client.

3) Travail demandé:

Cette idée a évolué pour concevoir un système de gestion de rendez-vous chez un salon de beauté constitué de trois parties :

- Une partie web permettant au partenaire d'ajouter son propre établissement et au client d'avoir la possibilité de le géo-localisé et de prendre un rendez-vous. (Le site web frontal)
- Une partie web où un partenaire peut gérer ses rendez-vous, ses prestations et son temps de travail. (Le Back-office pour le partenaire)
- Une partie web qui permet d'administrer et de gérer les partenaires et leurs droits d'accès. (Le Back-office pour l'administrateur de l'application)

4) Etude et critique de l'existant:

De nos jours, nous remarquons que le secteur de la beauté est diffusé dans la Tunisie et que ce dernier manque une sorte d'organisation qui s'intéresse à faciliter la connexion entre le client et les services des centres de beauté demandés et de rendre les centres plus disponible et accessible en cas de besoin.

Nous avons pris cette idée d'un site français existant qui a fait une forte croissance dans deux ans et qui représente plus que 10000 salons et institut de beauté et presque un rendez-vous pris toutes les secondes.

Donc, puisque cette idée de projet n'existe pas en Tunisie, cela est considéré un avantage pour que les Tunisiens réussissent mieux à l'utiliser et pour digitaliser le secteur de la beauté.

5) Cycle de vie de l'application :

Nous avons suivi le modèle en cascade qui se base sur les exigences de Winston Walter Royce qui divise les processus de développement de projet en cinq phases, qui sont les suivantes :

- **Analyse** : planification, analyse et spécification des besoins
- **Conception** : conception et spécification du système
- **Implémentation** : programmation et tests des modules
- **Test** : intégration du système, tests du système et de l'intégration
- **Exploitation** : livraison, maintenance, amélioration.

Le schéma ci-dessous présente l'une des extensions du modèle à suivre :

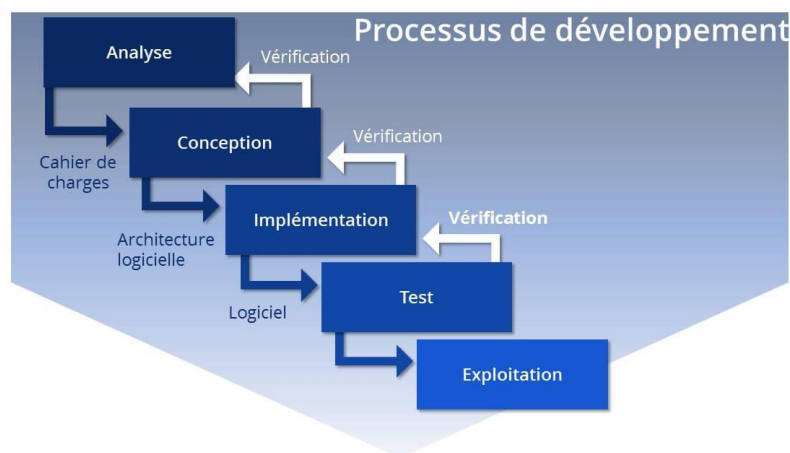


Figure 2: Cycle de vie de l'application

6) Planification du projet :

Le digramme suivant présente les taches à faire pendant la période de stage, avec des durées d'accomplissements approximatives.

	Mars				Avril				Mai				Juin			
Semaine	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
Formation Spring boot & Angular (individuelle)																
Etude préalable																
Modélisation conceptuelle																
Réalisation																
Validations & tests																
Rédaction du rapport																

Tableau 2: Planification

Conclusion :

Dans ce chapitre, nous avons présenté le contexte général de notre projet. Dans le chapitre suivant, nous allons s'intéresser aux spécifications des besoins du projet pour bien organiser le travail avant la mise en place de l'application.

Chapitre 2 : Analyse et spécification des besoins

Chapitre 2 : Analyse et spécification des besoins

Introduction :

Dans ce chapitre on va spécifier les différents besoins fonctionnels et non fonctionnels de cette application pour identifier par la suite les différents acteurs et pour présenter le diagramme de cas d'utilisation de notre système.

I. Spécification des besoins fonctionnels et non fonctionnel :

1) Spécification des besoins fonctionnels :

Dans cette partie nous détaillons l'ensemble de fonctionnalité que l'application doit offrir aux utilisateurs.

En effet, le système à réaliser doit répondre aux besoins fonctionnels suivants :

- Authentification : chaque utilisateur, possède un login et un mot de passe spécifique que lui permet de vérifier son identité pour avoir l'accès d'utiliser l'application.
- Gestion des rendez-vous : le client a la possibilité de prendre un rendez-vous et il peut consulter l'état de sa demande de rendez-vous (en attente, acceptée, refusée), de plus le partenaire peut accepter ou refuser une demande de rendez-vous selon sa disponibilité.
- Voir l'historique des rendez-vous : le partenaire peut consulter l'historique des demandes qu'il a acceptées ou refusées.
- Chercher un salon par nom, spécialité ou par localité.
- Vérification des partenaires : un partenaire doit envoyer une image de sa carte d'identité ou une autre pièce qui vérifie qu'il est un professionnel de beauté.

L'administrateur vérifie cette pièce justificative avant de valider le compte du partenaire et lui donne le droit d'accès pour ajouter son profil.

2) Spécification des besoins non fonctionnels :

Les besoins non fonctionnels définissent les règles à suivre et à respecter afin d'assurer une bonne qualité de l'application à réaliser et le bon fonctionnement du futur système, pour cela il faut répondre aux exigences suivantes :

- Ergonomie : le système doit être facile et compréhensible à utiliser et il doit permettre à l'utilisateur s'adapter rapidement avec le contenu et avoir une interface Homme Machine conviviale
- Le système doit détecter tous les messages d'erreurs
- La sécurité : les données du système doivent être sécurisées : contrôle d'accès et authentification.

Alors il faut utiliser le jeton de sécurité web (JSON WEB TOKEN ou bien JWT) qu'il s'agit d'un standard pour échanger de l'information d'une manière sécurisée à l'aide d'un jeton signé.

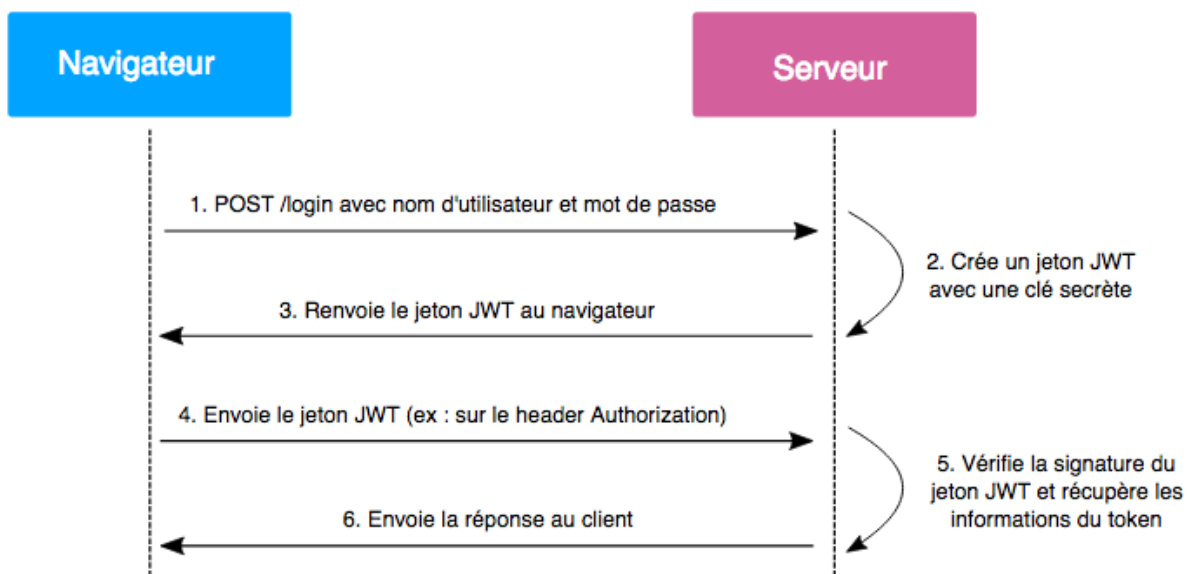


Figure 3: Fonctionnement de JWT

3) Identification des acteurs

Notre système présente trois parties qui sont nos trois applications web et chaque application concerne un acteur.

Liste des acteurs :

- Partenaire
- Client
- Administrateur

Cas d'utilisation	Acteur
Gérer ses rendez-vous	Partenaire
Gérer ses prestations	
Gérer ses horaires de travail	
Uploader CIN et la pièce de vérification	
Consulter les statistiques	
Chercher partenaire	Client
Consulter l'historique des rendez-vous	
Prendre un rendez-vous	
Consulter l'état des rendez-vous	
Gérer les partenaires	Administrateur

Tableau 3: Identification des acteurs et les cas d'utilisation

II. Diagrammes des cas d'utilisation

1) Diagramme de cas d'utilisation global :

Ce diagramme présente le système entier avec ses trois applications, chaque application a ses propres cas d'utilisation et son propre acteur comme présenter dans le diagramme ci-dessous.

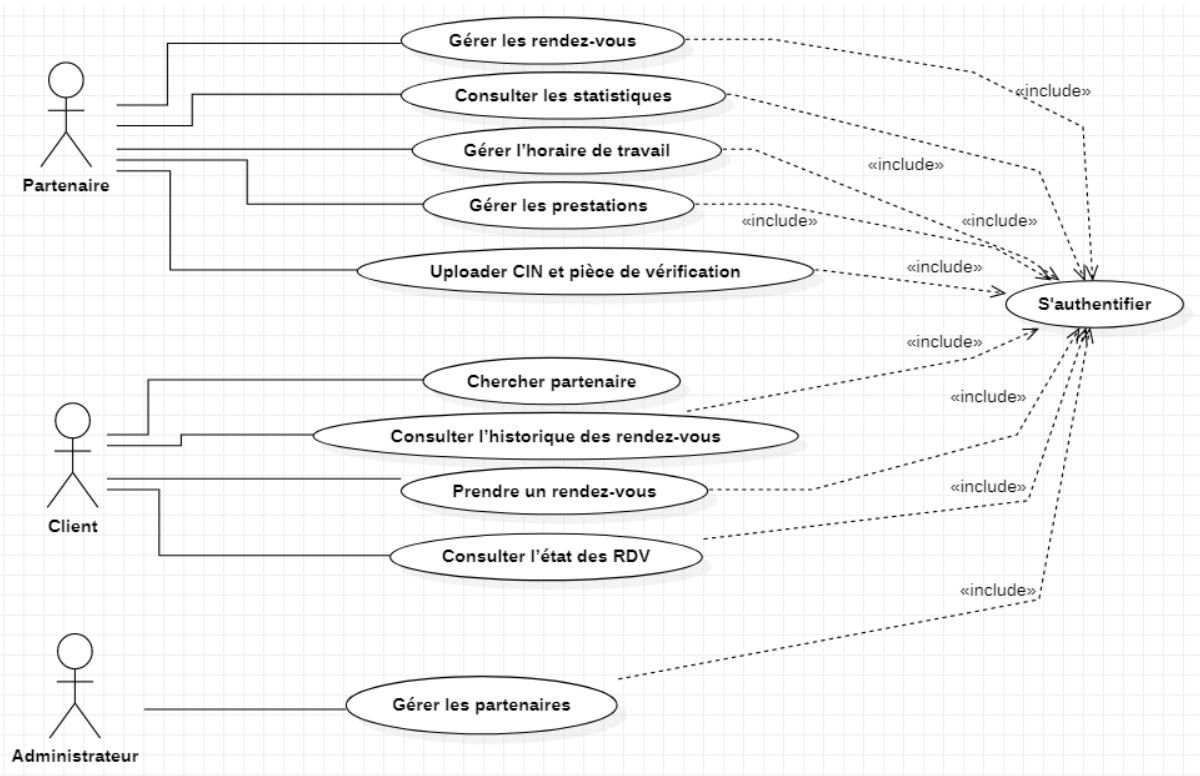


Figure 4: Diagramme des cas d'utilisation global

2) Raffinement des cas d'utilisation :

a) Raffinement des cas d'utilisation «Gérer les partenaires » :

L'administrateur a un back-office personnel, il peut accepter ou refuser l'inscription d'un partenaire après la vérification de la carte d'identité et la pièce de vérification envoyées par ce dernier.

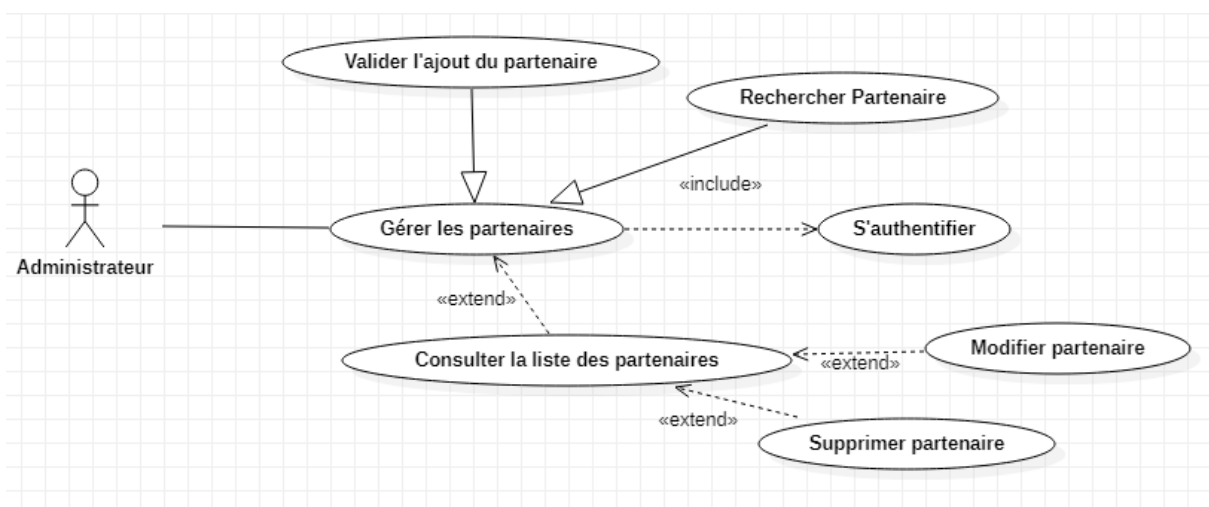


Figure 5: Diagramme de cas d'utilisation raffiné « Gérer les partenaires »

Description du cas d'utilisation « Gérer les partenaires » :

Casd'utilisation	Gérer les partenaires
Résumé	L'administrateur peut gérer les comptes qui existent dans le système : il peut ajouter ou modifier un compte.
Acteurs	Administrateur
Scénarionominale	<ul style="list-style-type: none"> - L'administrateur vérifie les informations des partenaires - Le système vérifie les informations des partenaires (existe déjà ou on) - Le système enregistre le partenaire s'il n'existe pas - L'administrateur reçoit un message de succès
Scénariod'erreur	<ul style="list-style-type: none"> - L'administrateur saisie un nom d'un partenaire existant - L'administrateur ne saisit rien
Pré-condition	Administrateur authentifié
Post condition	<ul style="list-style-type: none"> - Partenaire ajouté - Partenaire mis à jour

Tableau 4: Description du cas d'utilisation « Gérer les partenaires »

b) Raffinement de cas d'utilisation « chercher partenaire » :

Le client est l'utilisateur de l'application web. Il peut chercher un prestataire directement par son nom, son type d'activité ou par son localité.

Une fois le client a choisit le prestataire, il peut ainsi prendre un rendez-vous selon les disponibilités du prestataire.

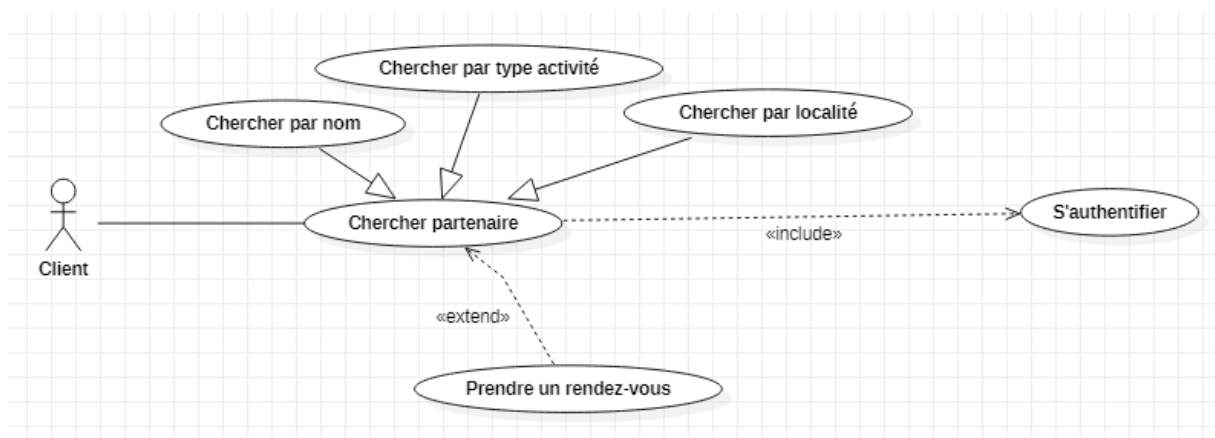


Figure 6: Digramme de cas d'utilisation raffiné « chercher partenaire »

Description de cas d'utilisation « chercher partenaire »:

Casd'utilisation	ChercherPartenaire
Résumé	Ce cas permet au client de chercher un prestataire
Acteurs	Client
Scenarionominal	<p>1- Le client saisit le nom du prestataire</p> <ul style="list-style-type: none"> - Le système cherche tous les prestataires avec ce nom. - Le client choisit un prestataire de la liste. - Le système affiche le profil du prestataire sélectionné. <p>2- Le client choisit un type d'activité du prestataire</p> <ul style="list-style-type: none"> - Le système cherche tous les prestataires avec ce type d'activité - Le client choisit un prestataire de la liste. - Le système affiche le profil du prestataire sélectionné. <p>3- Le client saisit la localisation du prestataire</p> <ul style="list-style-type: none"> - Le système cherche tous les prestataires avec cette localisation. - Le client choisit un prestataire de la liste. - Le système affiche le profil du prestataire sélectionné.
Scenariod'erreur	<ul style="list-style-type: none"> - Le client ne choisit pas aucun critère de recherche (nom, localisation, type activité) - Le client ne choisit aucun prestataire - Le client saisie un critère de recherche erroné

Tableau 5: Description de cas d'utilisation « chercher partenaire »

c) Raffinement de cas d'utilisation « prendre un rendez vous »:

Pour pouvoir réserver un rendez-vous, il doit se connecter s'il possède un compte déjà sinon il doit créer un compte .

Dès que le rendez-vous est pris, le client peut consulter la liste des rendez-vous(accepté,refusé ou en attente) et il a le droit de l'annuler.

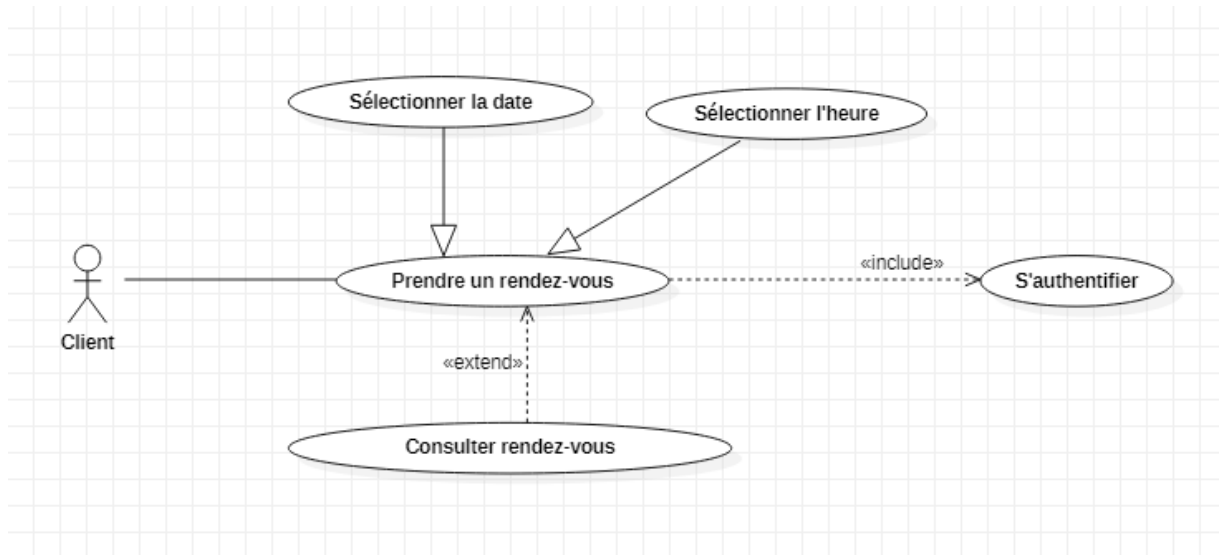


Figure 7: Digramme de cas d'utilisation raffiné « prendre rendez-vous »

Description de cas d'utilisation « prendre un rendez-vous »:

Casd'utilisation	Prendre un rendez-vous
Résumé	Ce cas permet au client de prendre un rendez-vous avec un prestataire
Acteurs	Client
Scénariominale	Le client sélectionne la date du rendez-vous Le client sélectionne l'heure du rendez-vous Le rendez-vous est enregistré dans la base de données
Scénariod'erreur	Le client décide de quitter l'interface de sélection de la date Le client décide de quitter l'interface de sélection de l'heure La date ne correspond pas à la disponibilité du prestataire
Pré-condition	Choisir un prestataire. Etreauthentifié.
Post condition	Rendez-vousenregistré

Tableau 6: Description de cas d'utilisation « prendre un rendez-vous »

d) Raffinement de cas d'utilisation « Gérer les prestations »:

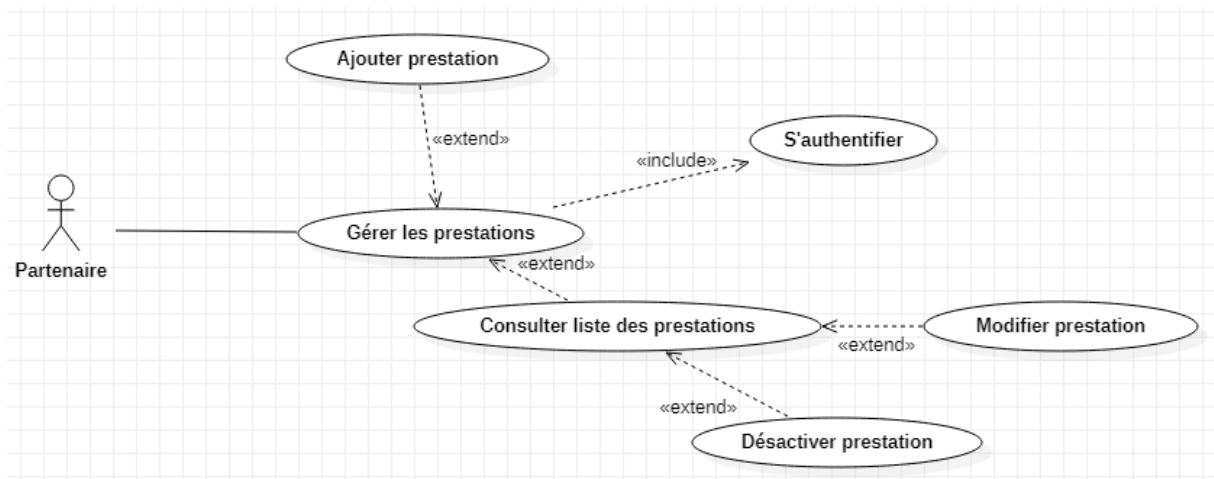


Figure 8: Digramme de cas d'utilisation raffiné « Gérer les prestations »:

Description de cas d'utilisation « Gérer les prestations»:

Casd'utilisation	Gérer les prestations
Résumé	Ce cas permet au partenaire de gérer les prestations
Acteurs	Partenaire
Scénarionominale	Le partenaire accède à son propre back-office où il peut ajouter, modifier et désactiver ses prestations
Scénariod'erreur	Le partenaire laisse un champ vide lors de l'ajout d'une prestation Le partenaire décide de quitter l'interface des prestations
Pré-condition	Etre authentifié Remplir tous les champs lors de l'ajout
Post condition	Prestation ajouté Prestation modifié Prestation désactivé

Tableau 7: Description de cas d'utilisation « Gérer les prestations»

e) Raffinement de cas d'utilisation « Gérer les rendez-vous »:

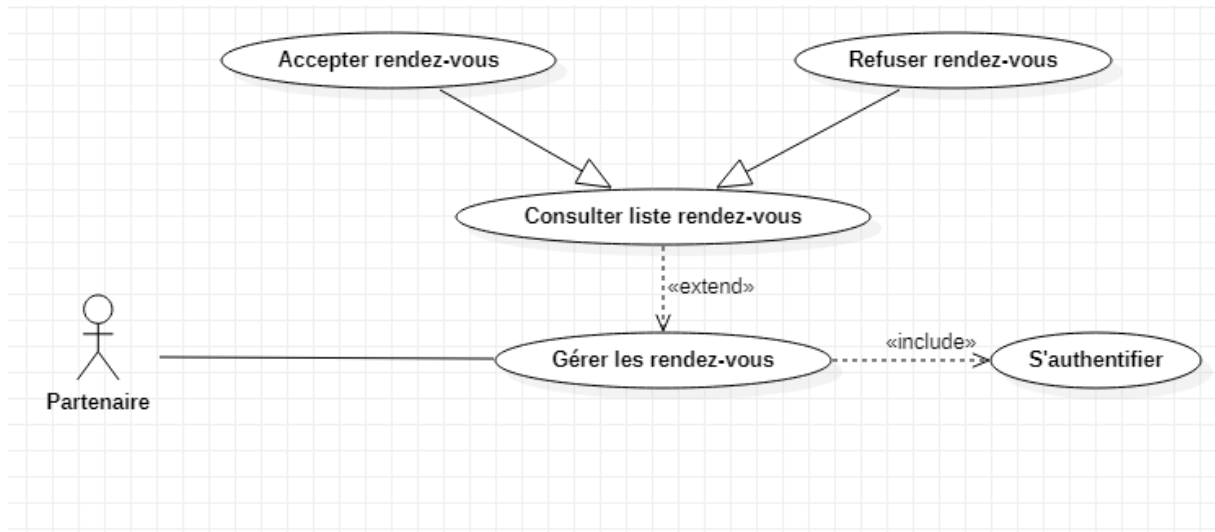


Figure 9: Diagramme de cas d'utilisation raffiné « Gérer les rendez-vous »:

Description de cas d'utilisation « Gérer les rendez-vous »:

Casd'utilisation	Gérer les rendez-vous
Résumé	Ce cas permet au partenaire de Gérer les rendez-vous
Acteurs	Partenaire
Scénarionominale	Le partenaire accède à son propre back-office Il consulte la liste des rendez-vous Il décide d'accepter ou rejeter les rendez-vous selon ses disponibilités
Scénariod'erreur	Exécutant non disponible
Pré-condition	Etre authentifié
Post condition	Rendez-vous accepté Rendez-vous rejeté

Tableau 8: Description de cas d'utilisation « Gérer les rendez-vous »

Conclusion :

Ce chapitre a été consacré pour la spécification des besoins fonctionnels et non fonctionnels du système en indiquant les rôles de chaque acteur ainsi la présentation des diagrammes des cas d'utilisation qui indiquent les différentes interactions entre les utilisateurs et le système résultant. Dans le chapitre suivant, nous entamerons la phase conceptuelle.

Chapitre 3 : Conception

Chapitre 3 : Conception

Introduction :

Dans ce chapitre on va tout d'abord présenter les différents outils logiciels et les langages de modélisation utiliser et après nous détaillons les diagrammes des classes et les diagrammes de séquences pour simplifier l'interaction des objets d'un système qu'on est en train de modéliser et montrer les messages échangés entre les lignes de vie, présentés dans un ordre chronologique.

Alors pour présenter l'architecture de notre système on va utiliser le langage de modélisation le plus fameux UML.

Les objectifs de la modélisation sont:

- Formaliser la conception d'application.
- Faciliter la communication entre les différents intervenants au sein d'un projet informatique.
- Coordonner les activités entre les différents intervenants.

I. Diagramme des classes :

1) Définition :

Le diagramme des classes est un schéma qui présente des classes, des interfaces et des packages et leurs différentes relations entre celle-ci pour fournir une vue logique du système informatique et pour simplifier l'interaction des objets d'un système qu'on est en train de modéliser.

Figure 10: Diagramme des classes

2) Modèle logique des données :

- Partenaire : (id_partenaire, Nom, Prenom, Description, UserName, mail, adresse, #id_ville, #id_pays, #id_type_activite)

Propriété		Type
Attribut	Désignation	
Id_partenaire	Le numéro qui identifie chaque partenaire	Long
Nom	Nom du partenaire	String
Prenom	Prénom du partenaire	String
Description	Les détails de chaque partenaire	String
UserName	Le nom de l'établissement	String
Mail	L'adresse mail du partenaire	String
Adresse	L'adresse exacte de l'établissement	String
id_ville	L'identifiant de la ville auquel le partenaire appartient	Long
id_pays	L'identifiant du pays auquel le partenaire appartient	Long
id_type_activite	L'identifiant des types d'activité auxquelles le partenaire appartient	Long

Tableau 9: Table Partenaire

- Client : (Id_client, Nom, Prenom, Mail, Num_tél)

Propriété		Type
Attribut	Désignation	
Id_client	Le numéro qui identifie chaque client	Long
Nom	Le nom du client	String
Prenom	Le prénom du client	String
Mail	L'adresse mail du client	String
Num_tél	Le numéro du téléphone	Long

Tableau 10: Table client

- horaires_ouverture : (id_ouv, jour_semaine, heure_ouverture, heure_fermeture, #id_partenaire)

Propriété		Type
Attribut	Désignation	
Id_ouv	Le numéro qui identifie chaque horaire d'ouverture	Long
jour_semaine	Les jours de travail	String
heure_ouverture	L'heure de fermeture	Time
heure_fermeture	L'heure d'ouverture	Time
id_partenaire	Le partenaire qui correspond a l'horaire d'ouverture	Long

Tableau 11: Table horaires_ouverture

- rendez_vous : (id_Rdv, date, statut, #id_partenaire, #id_executant, #id_client, #id_detail_prestation)

Propriété		Type
Attribut	Désignation	
Id_Rdv	L'identifiant de chaque rendez-vous	Long
Date	La date de rendez-vous	Date
Statut	Le statut qui caractérise chaque rendez-vous(en attente, accepté,rejeté, traité, en cours)	String
id_partenaire	L'identifiant du partenaire qui va adopter ce rendez-vous	Long
id_executant	L'identifiant de l'executant du rendez-vous	Long
Id_client	L'identifiant du client qui à reserver le rendez-vous	Long
id_detail_prestation	L'identifiant du detail du prestation	Long

Tableau 12: Table rendez_vous

- Prestation : (id_prestation, nom_prestation, #id_partenaire)

Propriété		Type
Attribut	Désignation	
Id_prestation	L'identifiant de chaque prestation	Long
Nom_prestation	Le nom du prestation	String
Id_partenaire	L'identifiant du partenaire auquel appartient la prestation	Long

Tableau 13: Table Prestation

➤ Executant : (id_exec, nom_exec, # id_partenaire)

Propriété		Type
Attribut	Désignation	
id_exec	L'identifiant de chaque avis	Long
nom_exec	Nom de l'exécutant	String
Id_partenaire	L'identifiant du partenaire qui lui appartient	Long

Tableau 14: Table Exécutant

➤ Avis : (id_avis, commentaire, nbr_etoile, note, #id_partenaire)

Propriété		Type
Attribut	Désignation	
Id_avis	L'identifiant de chaque avis	Long
Commentaire	Commentaires	String
Nbr_etoile	Nombre des étoiles	Int
Id_partenaire	L'identifiant du partenaire	Long

Tableau 15: Table Avis

II. Digrammes des séquences :

1) Définition :

Le digramme de séquence est un digramme d'interactions qui expose en détail la façon dont les opérations sont effectuées en précisant la chronologie des échanges de messages.

2) Diagramme de séquence d'authentification :

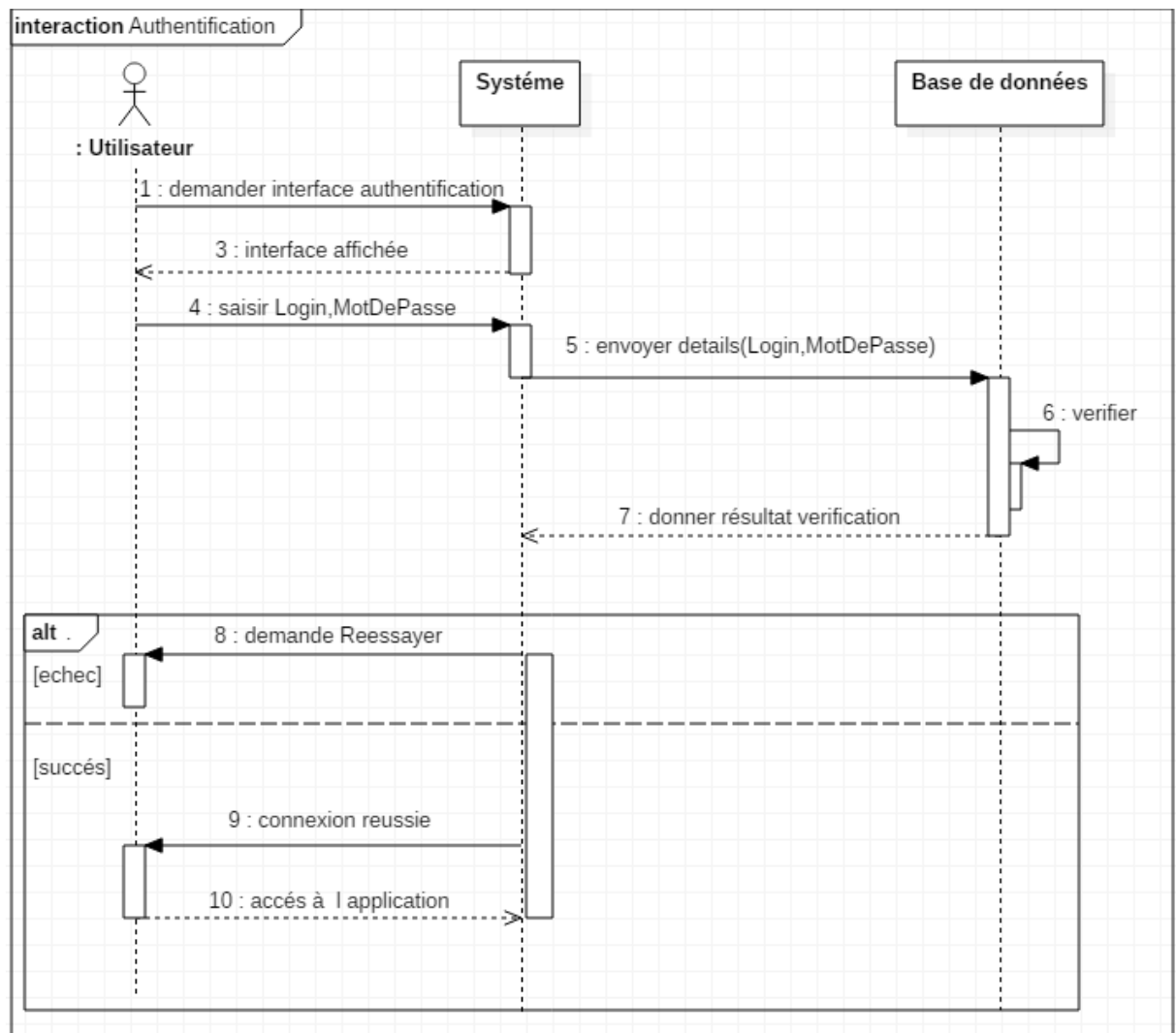


Figure 11: Diagramme de séquence d'authentification

Pour pouvoir accéder à son espace privé, chaque utilisateur doit s'identifier par son login et son mot de passe à travers le système de l'application qui prend en charge de vérifier et consulter la base de données.

- Si l'utilisateur est accepté, alors il y 'aura l'accès à l'application et à son espace souhaité selon son login.
- Sinon, le système lui affiche un message d'erreur pour redresser ses données.

3) Diagramme de séquence de la recherche du partenaire :

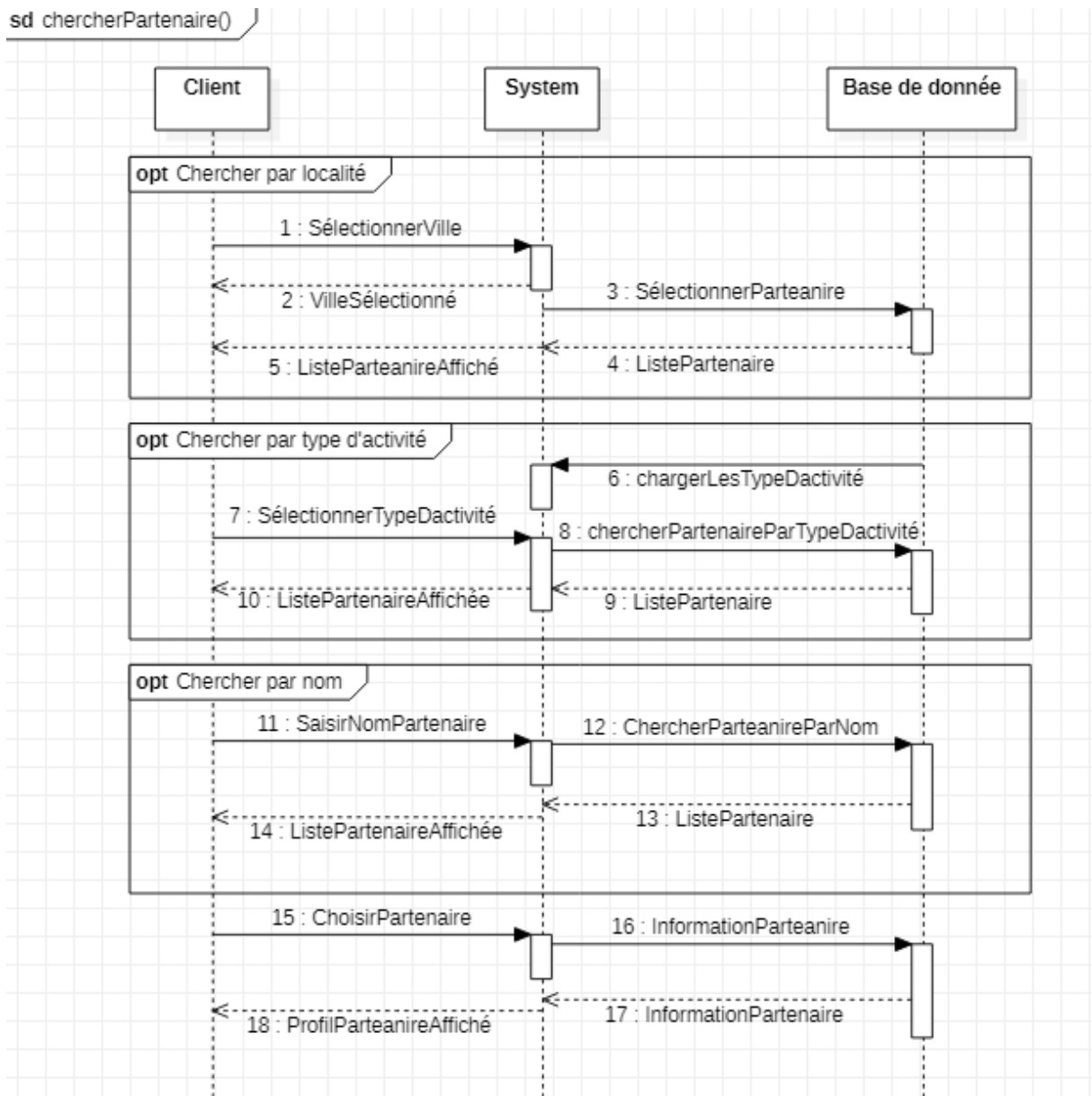


Figure 12: Diagramme de séquence de la recherche du partenaire

Le client peut chercher un salon par son nom, localisation ou bien par son type d'activité

Le client à la possibilité de chercher un salon en saisissant son nom et le système affiche tout de suite la liste des partenaires avec ce nom.

Pour chercher un salon par son type d'activité, le client choisi parmi les types existant dans la barre de navigation.

En plus, le client peut chercher un salon par sa localisation et peut choisir celui qui lui convient le mieux, c'est une fonctionnalité déployée par l'API de Google adaptée pour un usage facile et rapide.

4) Diagramme de séquence de prise de rendez-vous :

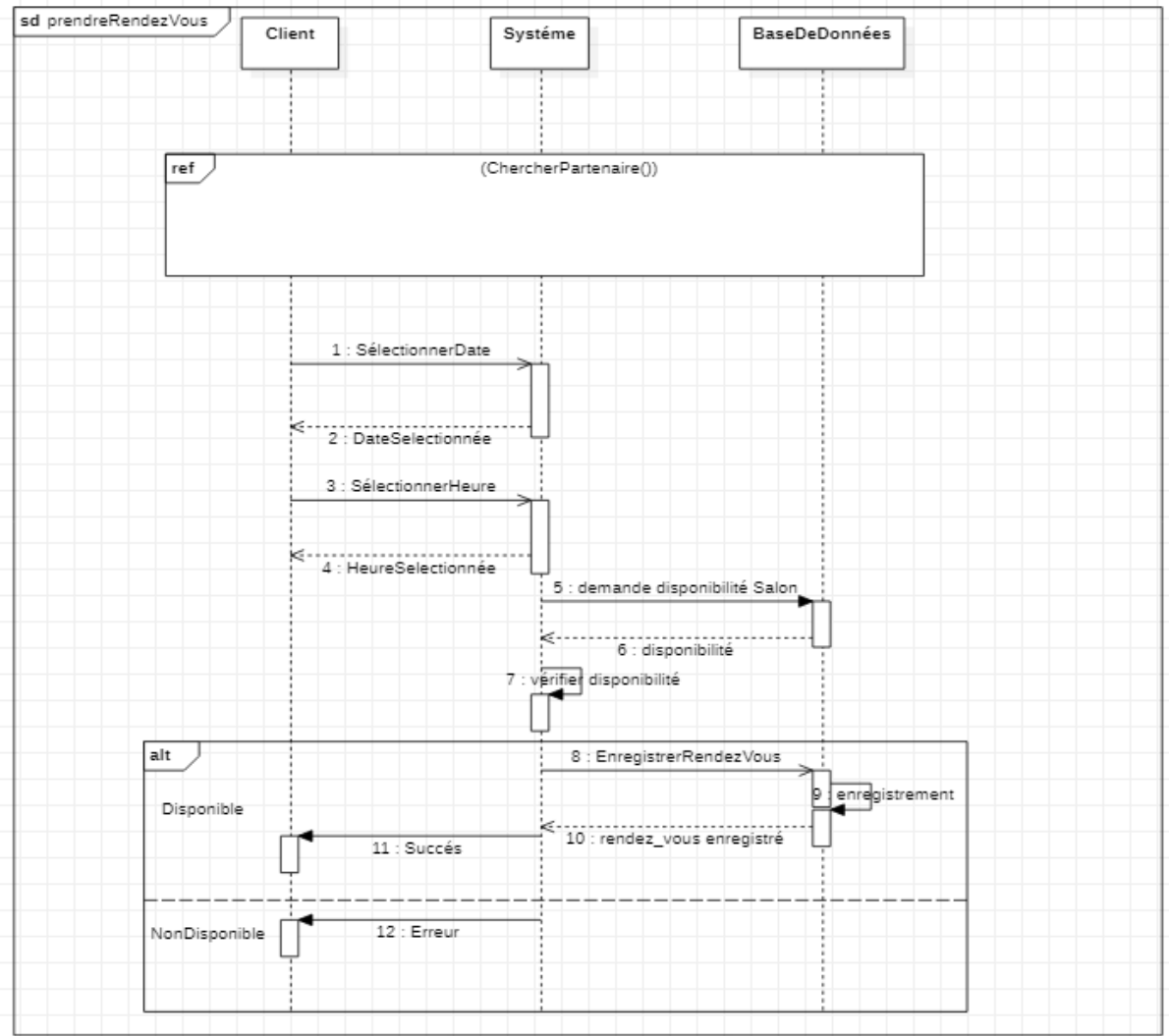


Figure 13: Diagramme de séquence de prise de rendez-vous

Pour prendre un rendez-vous, le client doit sélectionner la date et l'heure du rendez-vous ambitionné.

Après la vérification de la disponibilité depuis le prestataire, le rendez-vous sera enregistré dans la base de données, un message de confirmation s'affiche, ensuite le rendez-vous sera ajouté dans la liste des rendez-vous du client. Si la date et/ou l'heure sélectionnée ne correspondent pas à la disponibilité du partenaire, le système envoie un message d'erreur au client pour lui informer que son rendez-vous n'a pas été enregistré et lui demande de choisir une autre date.

5) Diagramme de séquence de création compte client :

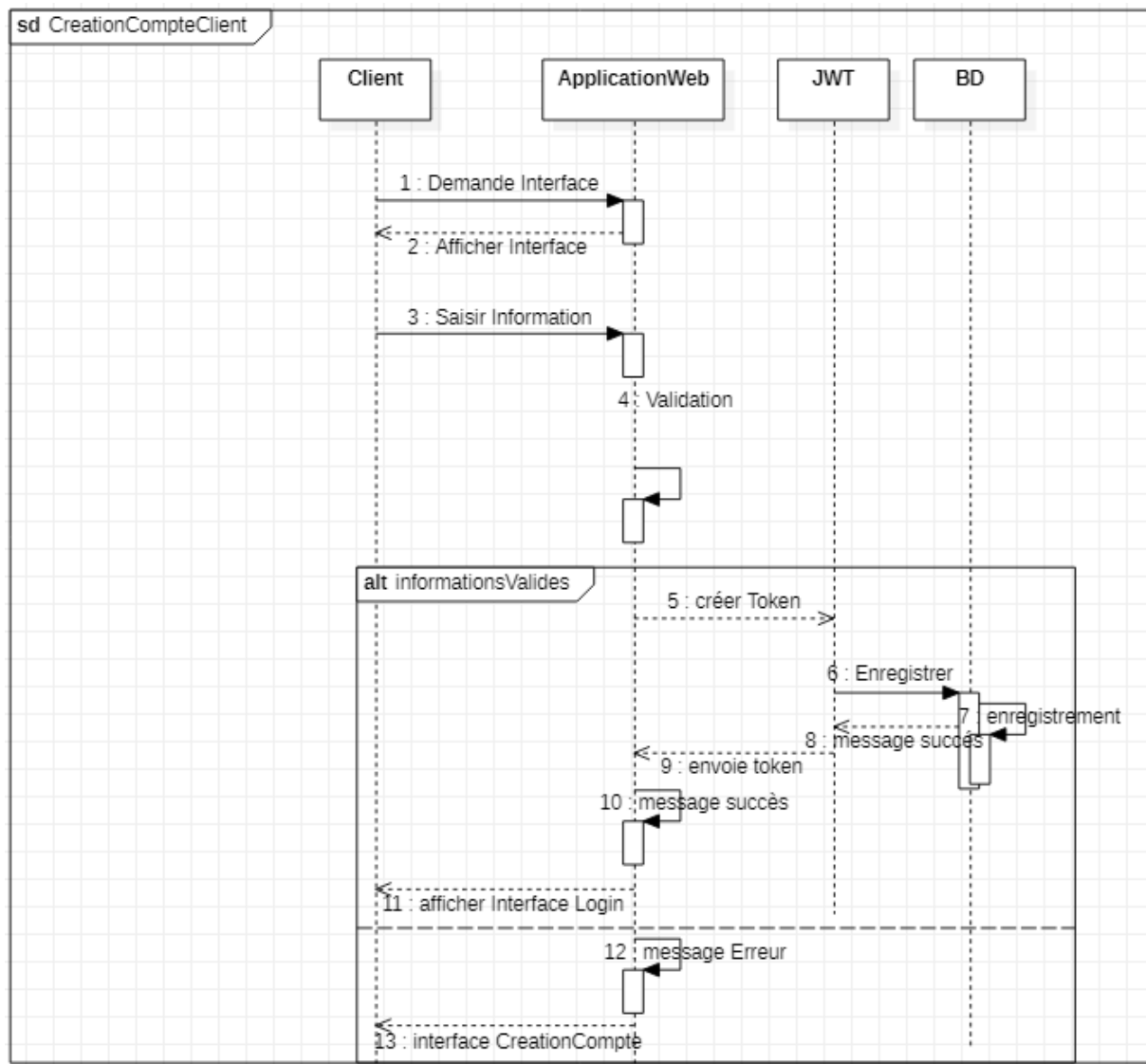


Figure 14: Diagramme de séquence de création compte client

Pour prendre un rendez-vous le client doit avoir un compte.

Pour crée un compte, il faut que le client saisi les informatients demandées correctement, puis le système vérifie ses inforamations et si n'a pas des erreurs le compte sera créé et les informations seront enregistrées dans la base de données.

Conclusion :

Ce chapitre présente une phase essentielle pour l'étude et l'analyse de notre application où nous avons présenté la phase conceptuelle de notre solution.

Dans le chapitre suivant, nous entamerons la phase de l'implémentation.

Chapitre 4 : Implémentation

Chapitre 4 : Implémentation

Introduction :

Nous arrivons maintenant au dernier volet de ce rapport. Cette partie est très importante, elle met en réalité toute la théorie étudiée dans les chapitres précédents. D'abord, nous commençons par la présentation de l'environnement matériel et logiciel et après nous présentons des captures d'écrans de notre application pour mettre en évidence l'aspect ergonomique et fonctionnel des interfaces développées qui sont organisées selon les fonctionnalités offertes à l'utilisateur.

I. Environnement de travail :

Pour mettre en place notre système, nous avons exploité un environnement de développement qui a assuré le bon déroulement de la phase implémentation et qui comporte des outils matériels et logiciels.

1) Environnement matériel :

Notre application a été développée sur deux ordinateurs portables dont les caractéristiques sont les suivantes :

- 1er ordinateur : ASUS VivoBook

Processeur : Intel(R) Core(TM) i5-8250U CPU @ 1.60GHz 1.80 GHz
Mémoire installée (RAM) : 8,00 Go (7,88 Go utilisable)
Type du système : Système d'exploitation 64 bits, processeur x64

- 2ème ordinateur : ASUS X550JX

Processeur : Intel(R) Core(TM) i7-4750HQ CPU @ 2.00GHz 2.00 GHz
Mémoire installée (RAM) : 12,0 Go (11,7 Go utilisable)
Type du système : Système d'exploitation 64 bits, processeur x64

2) Environnement logiciel :

a) Système d'exploitation :

Microsoft Windows 10 qui apporte un lot de nouvelles fonctionnalités mais améliore aussi sensiblement des technologies apparues avec Windows 7 et Windows 8 comme par exemple :

- ✓ Interface utilisateur
- ✓ Continuum
- ✓ Bureau virtuel
- ✓ IPv6

b) L'outil de modélisation :

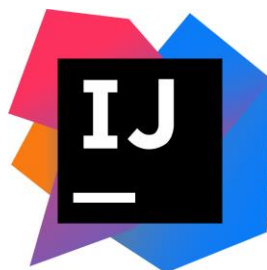
StarUML est notre logiciel de modélisation UML, cédé comme open source par son éditeur, à la fin de son exploitation commerciale, sous une licence modifiée de GNU GPL.



c) Environnement de développement intégré :

IntelliJ Educational Edition 2020.3.4 est notre IDE (Integrated Development Environment) destiné au développement de logiciels informatiques reposant sur la technologie Java.

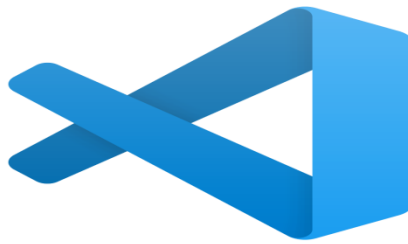
Il est développé par JetBrains (anciennement « IntelliJ ») et disponible en deux versions, l'une communautaire, open source, sous licence Apache 2 et l'autre propriétaire, protégée par une licence commerciale.



d) Editeur de code :

Visual Studio Code est notre éditeur de code extensible développé par Microsoft en 14 novembre 2015 qui peut être utilisé avec une variété de langages de programmation. Il est basé sur le cadre Electron, qui est utilisé pour développer des applications Web Node.js.

Dans le Stack Overflow 2019 Developer Survey, Visual Studio Code a été classé comme l'outil d'environnement de développement le plus populaire, avec 50,7 % des 87 317 répondants déclarant l'utiliser.



e) Outils d'administration de la base de données :

XampServer est l'outil de l'implémentation de notre application. c'est un ensemble de logiciels permettant de mettre en place un serveur Web local, un serveur FTP et un serveur de messagerie électronique. Il est à la portée d'un grand nombre de personnes puisqu'il ne requiert pas de connaissances particulières et fonctionne, de plus, sur les systèmes d'exploitation les plus répandus.



MySQL est la base de données open source la plus populaire au monde, il est distribué sous une double licence GPL et propriétaire. Il fait partie des logiciels de gestion de base de données (SGBD) les plus utilisées au monde. MySQL peut vous aider à concevoir des applications de base de données évolutives et hautement performantes.



f) Postman :

Postman est une application permettant avec un navigateur Web de lancer des appels d'API et de les tester.

L'onglet Tests de Postman nous permet d'envoyer des requêtes vers l'API de site en lui ajoutant des en-têtes clés /valeurs puis il permet de formater le résultat sur plusieurs formats tels que JSON, XML, HTML et autres.



g) L'architecture MVC :

L'architecture adoptée pour la couche présentation de notre application est l'architecture Model-View-Controller MVC qui consiste à la réalisation des différentes interfaces de l'application et leur affichage pour établir le dialogue avec l'utilisateur. C'est un pattern architectural qui sépare les données (le modèle), l'interface homme-machine (la vue) et la logique de contrôle (le contrôleur).

Spring MVC et Angular intègrent ensemble une pile de développement frontal très productive et attrayante pour la création d'applications Web à forte intensité de formulaire.

Voici un schéma des interactions entre les différentes couches :

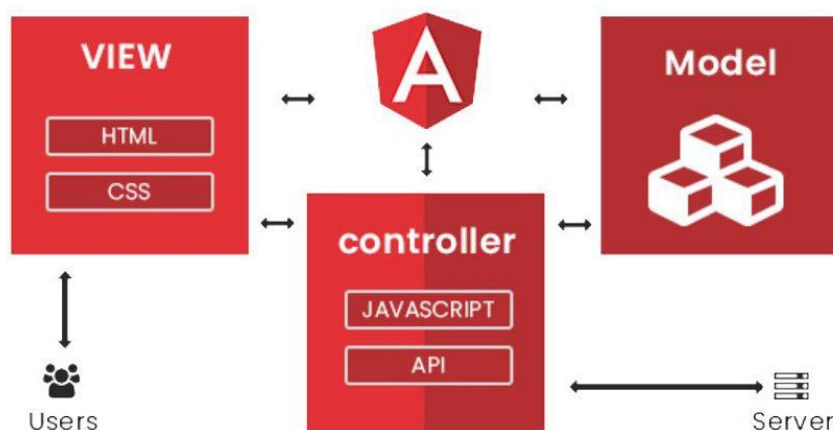


Figure 15: Des interactions entre les différentes couches de MVC

Les avantages de l'approche MVC :

- Un gain de temps de maintenance et d'évolution du site
- La Séparation des tâches, séparer la logique métier, l'interface utilisateur et la dynamique du système.
- Le Développement, les CI(continuous Integration), CD(continuous development) en parallèle, la possibilité pour les développeurs de travailler en parallèle, en même temps sans se marcher dessus.

II. Développement :

1) Les Frameworks utilisé :

Le framework est un ensemble d'outils et de composants logiciels organisés selon un plan d'architecture et des patterns, l'ensemble formant 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.



a) Le Framework Spring-Boot :

Spring Framework fournit un modèle complet de programmation et de configuration pour les applications d'entreprise modernes basées sur Java - sur tout type de plate-forme de déploiement.

Un élément clé de Spring est la prise en charge des infrastructures au niveau des applications : Spring se concentre sur la « plomberie » des applications d'entreprise afin que les équipes puissent se concentrer sur la logique métier au niveau des applications, sans liens inutiles avec des environnements de déploiement spécifiques. « 3 »



Fonctionnalités :

- Créer des applications Spring autonomes
- Intégrez directement Tomcat, Jetty ou Undertow (pas besoin de déployer des fichiers WAR)
- Fournissez des dépendances « de démarrage » avisées pour simplifier votre configuration de construction
- Configurez automatiquement les bibliothèques Spring et tierces dans la mesure du possible
- Fournir des fonctionnalités prêtes pour la production telles que des métriques, des vérifications de l'état et une configuration externalisée
- Absolument aucune génération de code et aucune exigence de configuration XML
- L'inversion de contrôle

b) Le Framework Angular :

Angular est une plate-forme et un framework permettant de créer des applications clientes d'une seule page à l'aide de HTML et de TypeScript. Angular est écrit en TypeScript. Il implémente les fonctionnalités principales et facultatives sous la forme d'un ensemble de bibliothèques TypeScript que vous importez dans vos applications. « 4 »

2) Les interfaces graphiques :

La conception des interfaces de l'application est considérée comme une étape très importante puisque toutes les interactions avec le cœur de l'application passent à travers ces interfaces,

on doit alors guider et informer l'utilisateur avec les messages d'erreurs et de notification si besoin, ainsi présenter un système complet qui répond au besoin de l'utilisateur.

a) Interface d'accueil :

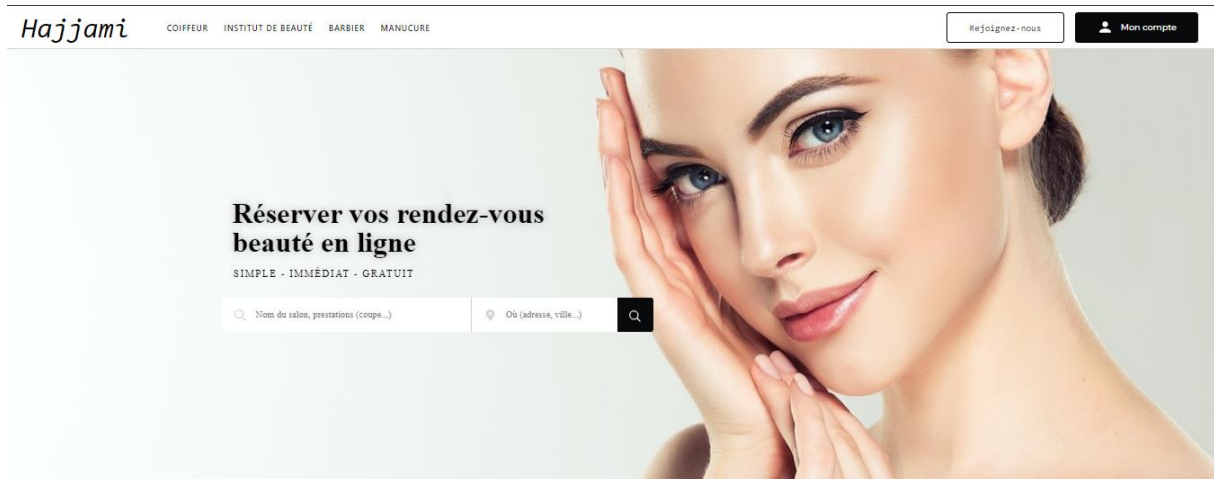


Figure 16: Interface d'accueil

b) Interface login du client :

L'interface d'authentification est une des interfaces les plus importantes dans l'application car chaque client doit être enregistré dans le système pour qu'il puisse prendre un rendez-vous.

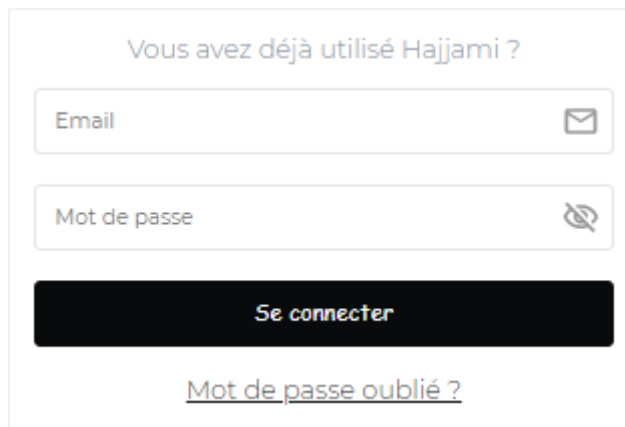
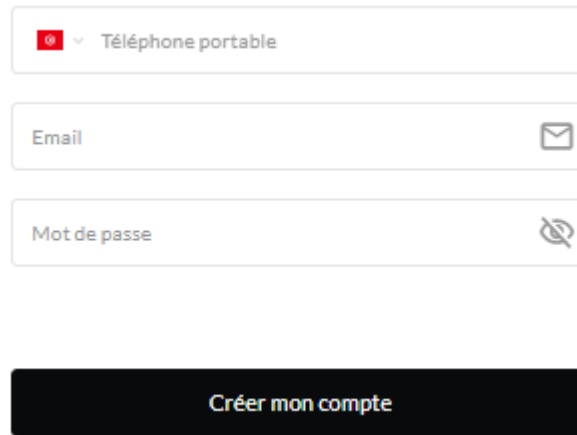


Figure 17: Interface login client

c) Interface sign up client :

Si un client n'avait pas un compte, il doit s'inscrire pour pouvoir réserver un rendez-vous



Form fields:

- Téléphone portable
- Email
- Mot de passe
- Créer mon compte

Figure 18: Interface sign up client

d) Interface de la recherche de partenaire:

A travers les interfaces de recherche d'un partenaire, l'utilisateur a la possibilité de chercher un salon par type activité ou par localité comme le montre la figure suivante :

➤ Exemple : coiffeur

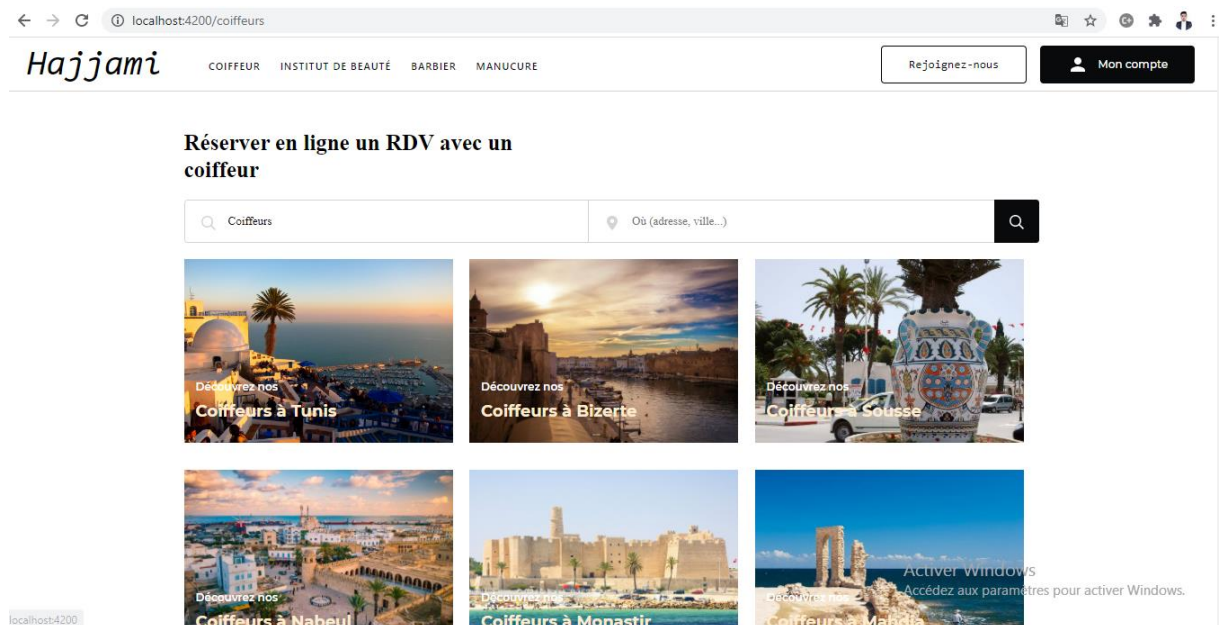
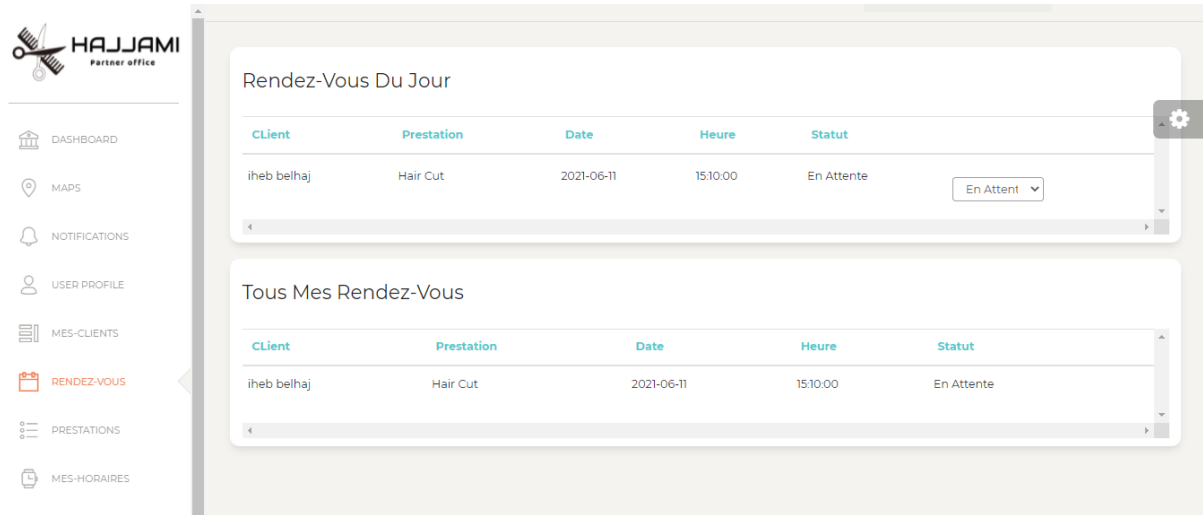


Figure 19: Interface de la recherche de partenaire (type activité: coiffeur)

e) Interface liste des rendez-vous:

Dans cette interface le partenaire peut trouver la liste de tous les rendez-vous classés par date:

- Une première liste contient les rendez-vous du jour que le partenaire a la possibilité de les mettre en attente, accepté ou bien rejeté.
- Une deuxième liste contient tous les rendez-vous qui existent dans sa base de données quelque soit leurs statut (traité, accepté, rejeté)



Rendez-Vous Du Jour

Client	Prestation	Date	Heure	Statut
iheb belhaj	Hair Cut	2021-06-11	15:10:00	En Attente

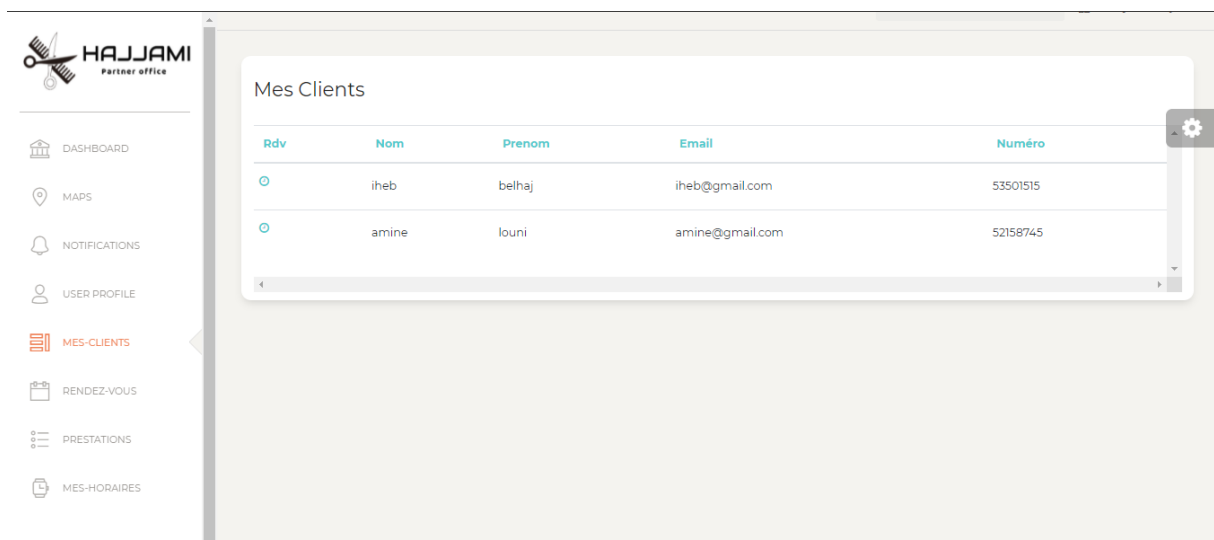
Tous Mes Rendez-Vous

Client	Prestation	Date	Heure	Statut
iheb belhaj	Hair Cut	2021-06-11	15:10:00	En Attente

Figure 20: Interface liste des rendez-vous

f) Interface liste des clients

A partir de cette interface, le partenaire peut consulter la liste de tous les clients avec leurs rendez-vous.



Mes Clients

Rdv	Nom	Prenom	Email	Numéro
🕒	iheb	belhaj	iheb@gmail.com	53501515
🕒	amine	louni	amine@gmail.com	52158745

Figure 21: Interface liste des clients

g) Interface des horaires:

Cette interface permet au partenaire de consulter ses horaires et les modifier selon ses disponibilités.

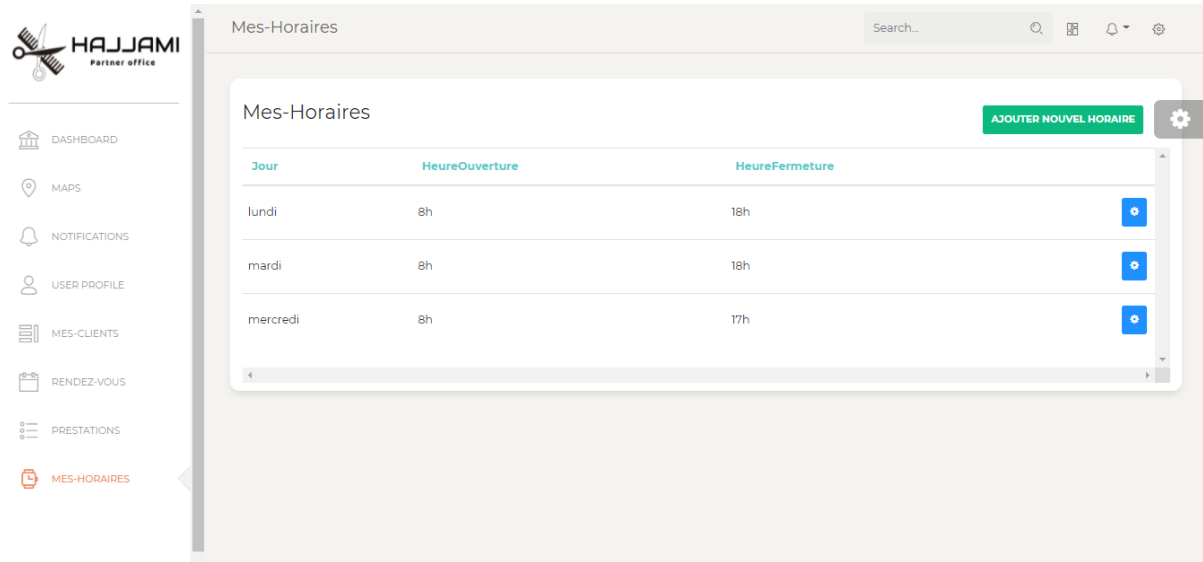


Figure 22: Interface des horaires

h) Interface des prestations:

Le partenaire est le seul responsable d'ajout et de modification des prestations. A partir de cette interface, il peut consulter la liste de toutes les prestations qui existent dans sa base de données, comme il peut ajouter une nouvelle prestation ou modifier une qui est déjà existante ou encore la désactiver.

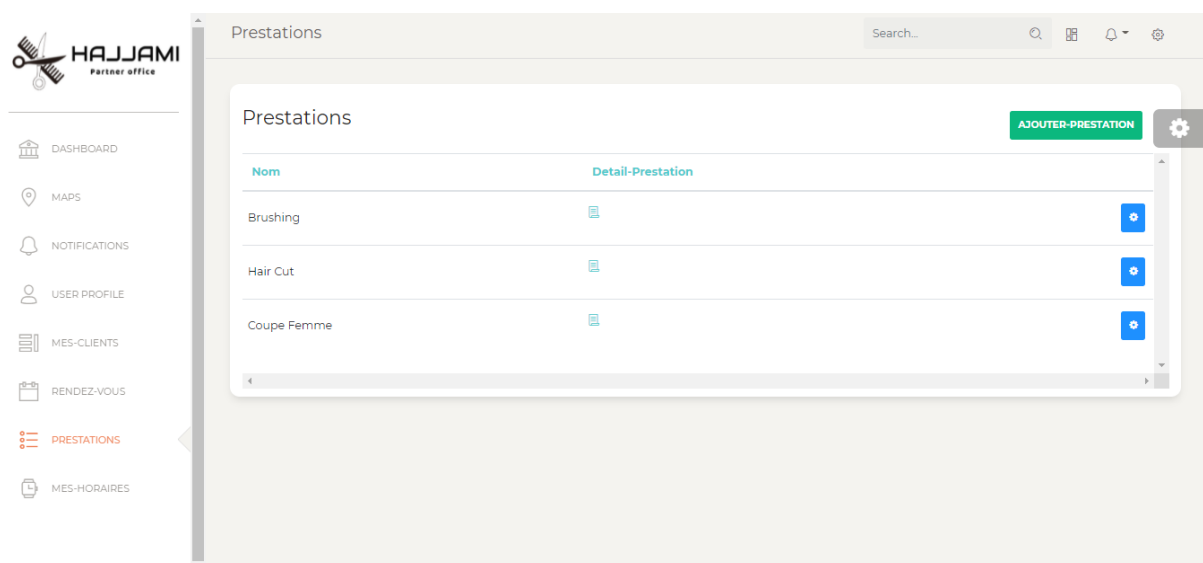


Figure 23: Interface des prestations

Chaque prestation a ses propres détails, le partenaire peut les consulter, modifier ou bien ajouter un détail.

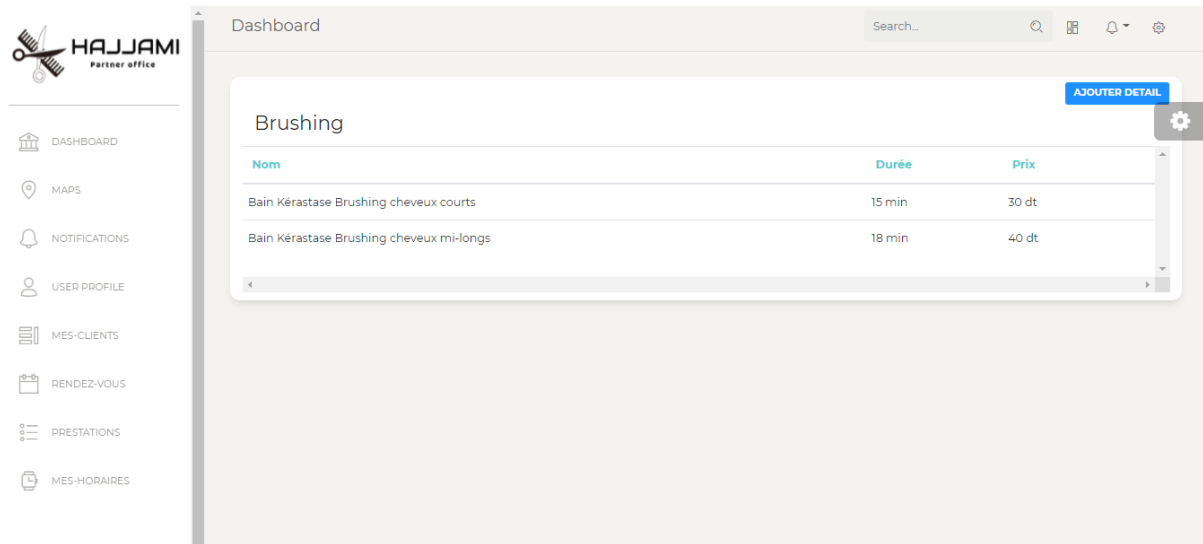


Figure 24: Interface des details prestations

Conclusion :

Au cours de ce chapitre, nous avons décrit les plateformes matérielles et logicielles sur lesquelles nous avons développé notre système en expliquant quelques interfaces graphiques constituant notre application que nous avons jugé les plus importantes.

Conclusion générale et perspectives

Notre projet a été réalisé dans le cadre d'un stage de fin d'études au sein de la société VEGANET. Nous avons conçu et développé une application web qui consiste à la gestion des rendez-vous des salons de beauté permettant d'une part aux clients de chercher un salon (partenaire) et prendre un rendez-vous, et d'autre part aux partenaires de gérer les rendez-vous. Ce rapport détaille toutes les étapes par lesquelles nous sommes passées pour arriver à ce résultat.

Ce projet comporte quatre étapes essentielles : Etude préalable, l'analyse, la conception et la réalisation (implémentation). Nous avons commencé par comprendre le contexte général du projet et identifié les différentes exigences du futur système. Nous avons utilisé les diagrammes de cas d'utilisation pour spécifier les exigences de notre application. Ensuite, nous avons abordé la phase conceptuelle de l'application en utilisant le diagramme de classes et les diagrammes de séquence. Pour la réalisation, nous avons utilisé les frameworks Spring-Boot et Angular et MySQL comme système de gestion de base de données. En plus ce projet était une grande opportunité pour découvrir des nouvelles technologies, des nouvelles techniques, des logiciels et d'améliorer nos compétences de programmation.

Bien que les principaux objectifs de notre projet soient atteints, l'application que nous avons développée pourrait être enrichie par d'autres fonctionnalités.

Notre plate-forme Web sera beaucoup plus complète par le développement et la mise en place d'une application mobile destinées à la clientèle afin de pouvoir recevoir des push notification de confirmation de réservation en temps réel, les partenaires peuvent aussi envoyer leurs promotions et produits en vente sous forme de notifications informatives.

Nous allons continuer à développer cette application mobile dans le cadre de stage d'été au sein de la société VEGANET.

Bibliographie

- « 1 » <https://stackoverflow.com> (Consulté le 17-03-2021)
- « 2 » <https://openclassrooms.com/fr/courses/4668271-developpez-des-applications-web-avec-angular> (Consulté le 17-03-2021)
- « 3 » <https://spring.io/> (Consulté le 17-03-2021)
- « 4 » <https://angular.io/> (Consulté le 17-04-2021)
- « 5 » <https://fr.wikipedia.org/> (Consulté le 10-06-2021)
- « 6 » <https://www.w3schools.com/> (Consulté le 17-04-2021)

Résumé

Ce présent travail s'inscrit dans le cadre de projet de fin d'étude pour l'obtention du diplôme de Licence Appliquée Technologies des Réseaux Informatiques. Ce stage est effectué au sein de l'entreprise VEGANNET, consiste à développer une application web de gestion des rendez-vous chez les salons de coiffure. Pour réaliser ce travail, nous avons utilisé les Framework spring boot et Angular et java comme un langage de programmation,MySQL pour la gestion de la base de données et la méthodologie en cascade comme un cycle de vie pour notre projet.

Mots clés: Application Web, Gestion des rendez-vous, Framework Spring boot, Framework Angular, Java, MySQL.

Abstract

This presentworkis part of the end-of-studyproject for obtaining the Applied License in Computer Network Technologies diploma. This internshipiscarried out within the company VEGANNET, consists in developing a web application for managingappointments at hairdressing salons. To carry out thiswork, weused the spring boot and Angular and java frameworks as a programminglanguage, MySQL for the database management and the waterfallmethodology as a life cycle for ourproject.

Keywords: Web application, Appointment management, Spring boot framework, Angular framework, Java, MySQL.

