

République Algérienne Démocratique et Populaire  
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université A/Mira de Béjaïa

Faculté des Sciences Exactes

Département d'Informatique



*Mémoire de fin de cycle*

*En vue de l'obtention du diplôme de Master en Informatique*

*Option : Génie Logiciel*

## Thème

---

Conception et réalisation d'un module pour la  
gestion de réservation de produit/services pour la  
plateforme Vscloud (Vegasoft)

---

Réalisé par

Melle. *MOUHCENE* Narimane Melle. *AGHOULES* Naoual

Déposé pour évaluation le 13 Octobre 2021 par le jury composé de :

Examineur	M. Allem Khaled	M.A.A	U. A. MIRA de Béjaïa
Examinatrice	Mme Mammeri Sohila	M.C.B	U. A. MIRA de Béjaïa
Encadrant	Mr Ouzeguen	M.A.B	U. A. MIRA de Béjaïa

Promotion 2020-2021

# *Remerciements*

Avant toute chose, nous souhaitons remercier nos parents qui nous ont soutenu tout au long de nos études. Nous remercions également tous ceux qui ont contribué de près ou de loin au bon déroulement de notre stage ainsi que ceux qui nous ont aidé lors de la rédaction de ce rapport de fin d'études.

Nous souhaitons remercier aussi la société VegaSoft pour m'avoir accueilli dans le cadre de mon stage. Nous remercions tout particulièrement M. BADJA, notre tuteur de stage en entreprise qui a su nous faire bénéficier de son expertise dans le domaine du développement d'application web.

Ensuite, nous remercions les membres du jury qui ont accepté d'examiner ce travail afin de l'enrichir .

Enfin, nous souhaitons remercier notre tuteur universitaire M. OUZEGANE ainsi que l'ensemble des enseignants du département informatique qui nous ont accompagné durant ces cinq années où nous avons pu acquérir des connaissances fondamentales qui nous permettront d'intégrer le monde professionnel avec aisance.

## *Dédicaces*

Le présent travail n'est pas seulement le fruit de nos propres efforts, mais aussi les efforts de bien de personnes à qui nous exprimons nos vifs remerciements.

C'est avec une très grande joie qu'on dédie ce travail à notre famille qui ne cesse jamais de nous encourager et leurs soutiens qu'ils nous ont accordé tout au long de notre chemin.

À toute notre famille pour leurs amours et leurs soutiens inconditionnels, A nos meilleures amies, pour l'amitié qui nous a toujours unis. A tous ceux qui nous sont chers et qui nous ont supportés tout au long de notre parcours d'études.

**Narimane et Naoual**

## Table des matières

	Page
<b>Table des figures</b>	<b>iv</b>
<b>Liste des tableaux</b>	<b>vi</b>
<b>Introduction générale</b>	<b>1</b>
<b>1 Organisme d’Accueil et Recueil des Besoins</b>	<b>3</b>
1.1 Présentation de l’environnement d’accueil . . . . .	4
1.1.1 Services . . . . .	4
1.1.1.1 Network services, tel que : . . . . .	4
1.1.1.2 Cloud services . . . . .	5
1.1.1.3 Design services . . . . .	5
1.1.1.4 Implémentation services . . . . .	5
1.2 VScloud . . . . .	5
1.2.1 Les différentes fonctionnalités et services proposé par VsCloud . . . . .	5
1.3 Identification de la problématique (critique) . . . . .	6
1.4 Présentation de la solution proposée . . . . .	6
1.5 Les besoins Fonctionnels . . . . .	6
1.6 Les besoins Non Fonctionnels . . . . .	7
1.7 Conclusion . . . . .	8
<b>2 Analyse et conception des besoins</b>	<b>9</b>
2.1 Langage de modélisation unifié UML . . . . .	10
2.2 Les diagrammes . . . . .	10
2.2.1 Identification des acteurs . . . . .	10
2.2.2 Diagramme de contexte : . . . . .	11

2.2.3	Identification des messages . . . . .	12
2.2.4	Relation entre les acteurs . . . . .	12
2.2.5	Identification des cas d'utilisations . . . . .	12
2.2.6	Diagramme de cas d'utilisation . . . . .	12
2.2.6.1	Description textuelle des cas d'utilisation . . . . .	17
2.2.7	Diagramme de séquence . . . . .	23
2.3	Conclusion . . . . .	30
<b>3</b>	<b>Conception et Schéma Relationnel de données</b>	<b>31</b>
3.1	Diagrammes d'Interaction . . . . .	32
3.1.1	Classes d'interface (boundary) . . . . .	32
3.1.2	Classes de Contrôle (Control) . . . . .	33
3.1.3	Classes entités (entity) . . . . .	33
3.2	Les Diagrammes d'interaction des cas d'utilisation . . . . .	33
3.2.1	Diagramme d'interaction de cas d'utilisation «S'authentifier » . . .	33
3.2.2	Diagramme d'interaction de cas d'utilisation «rechercher produit »	35
3.2.3	Diagramme d'interaction de cas d'utilisation «demande réservations »	36
3.3	Diagramme de classe : . . . . .	37
3.3.1	Diagramme de classes du domaine (classes issues des Objets entités ci-dessus) . . . . .	37
3.3.2	Diagramme de classe général . . . . .	39
3.3.3	Diagramme de classes du réservation . . . . .	40
3.3.4	Classes, attributs et Responsabilités (Dictionnaire de données) : . .	41
3.4	Schéma relationnel (Règle de passages du diagramme de classes vers le schéma relationnel) . . . . .	46
3.4.1	Règles de passages au modèle relationnel . . . . .	46
3.4.1.1	Relation (1..*) : . . . . .	46
3.4.1.2	Relation (1..1) : . . . . .	46
3.4.1.3	Relation d'héritage . . . . .	46
3.4.1.4	Le passage au modèle relationnel . . . . .	47
<b>4</b>	<b>Réalisation</b>	<b>49</b>
4.1	Outils de développement : . . . . .	50

---

4.1.1	Visual Studio Code : . . . . .	50
4.1.2	Git et GitHub : . . . . .	50
4.1.3	WampServer : . . . . .	51
4.1.4	Resilio Sync : . . . . .	51
4.2	Langage de programmations utilisés : . . . . .	51
4.2.1	Html (HyperText Markup Language) : . . . . .	51
4.2.2	CSS (Cascading Style Sheets) : . . . . .	52
4.2.3	BOOTSTRAP : . . . . .	52
4.2.4	NodeJS : . . . . .	52
4.2.5	EJS : . . . . .	52
4.2.6	L'interface Homme-Machine « Connexion » pour l'admin : . . . . .	53
4.2.7	L'interface Homme-Machine « Gérer réservation » : . . . . .	55
4.2.8	L'interface Homme-Machine « Modifier réservation » : . . . . .	56
4.2.9	L'interface Homme-Machine «Ajouter réservation » : . . . . .	57
4.2.10	L'interface Homme-Machine «Inscription » Client : . . . . .	58
4.2.11	L'interface Homme-Machine «Réserver produits » : . . . . .	59
4.2.12	L'interface Homme-Machine «Gestion profil » : . . . . .	60
4.2.13	L'interface Homme-Machine « Valider la réservation » : . . . . .	61
4.2.14	L'interface Homme-Machine «Changer état» : . . . . .	62
	<b>Cocclusion générale</b>	<b>63</b>
	<b>Bibliographie</b>	<b>65</b>

# Table des figures

2.1	Diagramme de contexte . . . . .	11
2.2	Diagramme de cas d'utilisation de l'acteur «Utilisateur» . . . . .	14
2.3	Diagramme de cas d'utilisation de l'acteur «Agent System» . . . . .	14
2.4	Diagramme de cas d'utilisation de l'acteur «Membre» . . . . .	14
2.5	Diagramme de cas d'utilisation de l'acteur «Client» . . . . .	15
2.6	Diagramme de cas d'utilisation de l'acteur «Admin» . . . . .	16
2.7	Diagramme de séquence du cas d'utilisation rechercher produit . . . . .	24
2.8	Diagramme de séquence du cas d'utilisation authentification . . . . .	25
2.9	Diagramme de séquence du cas d'utilisation demander réservation . . . . .	26
2.10	Diagramme de séquence du cas d'utilisation gérer réservation part 1 . . . . .	27
2.11	Diagramme de séquence du cas d'utilisation gérer réservation part 2 . . . . .	28
2.12	Diagramme de séquence du cas d'utilisation gérer réservation part 3 . . . . .	29
3.1	Classe interface "IHM" . . . . .	32
3.2	Classe interface "Controle" . . . . .	33
3.3	Classe interface "Entité" . . . . .	33
3.4	Diagramme d'interaction de cas d'utilisation «S'authentifier . . . . .	34
3.5	Diagramme d'interaction de cas d'utilisation «rechercher produit » . . . . .	35
3.6	Diagramme d'interaction de cas d'utilisation «demande réservations » . . . . .	36
3.7	Diagramme de classes du domaine . . . . .	38
3.8	Diagramme de classe général . . . . .	39
3.9	Diagramme de classes du réservation . . . . .	40
3.10	Description détaillée des attributs de classes . . . . .	45
4.1	interface Homme-Machine « Connexion » pour l'admin . . . . .	54
4.2	interface Homme-Machine « Gérer réservation» . . . . .	55
4.3	interface Homme-Machine « Modifier réservation» . . . . .	56

4.4	interface Homme-Machine «Ajouter réservation » . . . . .	57
4.5	interface Homme-Machine « Inscription » . . . . .	58
4.6	interface Homme-Machine « Réserver produits » . . . . .	59
4.7	interface Homme-Machine « Gestion profil » . . . . .	60
4.8	interface Homme-Machine « Valider la réservation » . . . . .	61
4.9	interface Homme-Machine «Changer état» . . . . .	62



# Liste des tableaux

2.1	Identification des messages échangés . . . . .	12
2.2	Description textuelle de cas d'utilisation «S'authentifier». . . . .	18
2.3	Description textuelle de cas d'utilisation «Demande de réservation » . . . .	19
2.4	Description textuelle de cas d'utilisation «rechercher un produit ». . . . .	20
2.5	Description textuelle de cas d'utilisation «Gérer les réservations » . . . . .	23

## LISTE DES SIGLES ET ACRONYMES

- **API** Application Programming Interface.
- **CSS** Cascading Style Sheets.
- **Html** HyperText Markup Language.
- **IHM** Interfaces Homme-Machine.
- **PHP** Hypertext Preprocessor .
- **SAAS** Software As A Service .
- **SCM** gestion du code source .
- **UML** Unified Modeling Language.
- **VSC** Visual Studio Code .
- **Wamp** Windows, Apache, MySQL et PHP.

## INTRODUCTION GÉNÉRALE

La présence en ligne des entreprises aujourd'hui est primordiale pour être visible auprès de prospects au moment où ceux-ci en manifestent le besoin, cela permet également la pérennité des finances et même l'augmentation du chiffre d'affaires quand le site est bien référencé et touche sa cible au bon moment. Une course à l'obtention de sites web est lancée depuis quelques années maintenant. Cette transition au numérique demande certaines compétences techniques que seuls les informaticiens développeurs ont. Néanmoins, ces derniers temps l'apparition de CMS (Content Manager System) , les CMS sont représentés par un ensemble logiciel qui est affecté à deux tâches principales : générer une page web et consiste à gérer les pages ainsi créées. et cela sans une ligne de code , WordPress, Joomla ou encore Wix sont des incontournables de la création de site clé en main pour les non connaisseurs.

Pour une meilleure appréhension de ces CMS révolutionnaire, nous nous sommes rapprochés de VegaSoft car ils sont détenteurs d'une plateforme en ligne s'appelant VsCloud. Cette plateforme est très simple d'utilisation car en effet, l'utilisateur de Vscloud n'a besoin d'aucune connaissance préalable pour pouvoir créer son site soi-même. Elle est donc dédiée aux artisans, aux commerçants et toute entreprise voulant acquérir de la visibilité sur Internet et de ce fait, se placer sur de nouveaux canaux d'acquisition de prospects.

VegaSoft a accepté de nous accueillir en tant que stagiaires et nous a donné la possibilité de faire une analyse de l'existant et du besoin , afin d'avoir ample informations sur les fonctionnalités lancées et celles manquantes. Après cette analyse, nous avons constaté

que la fonctionnalité de réservation ( d'un produit choisit par le client ) n'était pas encore disponibles sur la Vscloud.

Notre mission durant ce stage fut donc l'ajout de plugins de réservation offrant ainsi la possibilité au client de faire sa réservation des produit/services proposé .. L'ajout de fonctionnalités annexes comme l'envoi de notifications ou encore la modification par le client de ses propres informations personnelles a été nécessaire pour offrir un bon service au client.

Pour mieux présenter notre travail, notre mémoire est structuré en quatre chapitres :

- Le premier chapitre intitulé " Organisme d'accueil et recueil des besoins " est consacré à la description de l'organisme d'accueil Vegasoft , Par la suite, en étudiant VScloud, nous avons établi la problématique , et par conséquent, nous avons proposé notre solution qui consiste a ajouter un plugin de réservation ;
- dans deuxième chapitre intitulé «Analyse des besoins» ,on va définir les acteurs de notre application, leurs différents cas d'utilisation, qui modélisent l'aspect fonctionnel de l'application ainsi que leurs descriptions textuels et à travail le diagramme de séquence. ;
- Le troisième chapitre intitulé « Conception et schéma relationnel de données » est centré sur l'élaboration des diagrammes d'interaction qui mettent l'accent sur les messages entre les acteurs et le système , puis on va établir le diagramme de classe et En fin de ce chapitre, et en appliquant les différentes règles de passage du diagramme de classe vers le schéma relationnel de données. ;
- Le quatrième et dernier et chapitre intitulé «Réalisation et test» introduit les outils et logiciels ayant servis à l'implantation du système. Il comporte notamment les différentes interfaces de l'application proposée.

Une conclusion générale marquera la fin de notre mémoire.

# CHAPITRE 1

## ORGANISME D'ACCUEIL ET RECUEIL DES BESOINS

## Introduction

Dans ce chapitre, nous ferons une brève présentation de notre organisme d'accueil VegaSoft, les services proposés par ce dernier. Pour ensuite faire l'étude de l'existant ou nous parlerons de l'application VsCloud, de ses fonctionnalités, et des manques auxquels Nous essayerons d'apporter des solutions.

### 1.1 Présentation de l'environnement d'accueil

**VegaSoft** est une entreprise d'informatique, situé à Lkseur, Bejaia, crée par Monsieur Badja Riad. Cette entreprise est spécialisée dans le développement des applications web dynamique et complexes. Elle a pour but de montrer l'importance de l'informatique dans le quotidien et dans le secteur professionnel.

VegaSoft se compose de :

1. *Directeur* : qui est Monsieur BADJA Riad, qui est aussi un développeur web ;
2. *Equipe* de développeurs : L'équipe VegaSoft est composée essentiellement de développeurs web maîtrisant HTML, CSS, Node JS et Java. L'équipe ne découpe pas les projets sur une base de front end et back end mais sur les fonctionnalités que chacun peut développer.

#### 1.1.1 Services

**VegaSoft** propose une multitude de services à ses clients parmi ces derniers on site ceux-ci :

##### 1.1.1.1 Network services, tel que :

- Des services d'installation,
- Configuration,
- Réparation et sécurisation des réseaux informatiques .

### 1.1.1.2 Cloud services

- Des services web, API et une gestion complète d'entreprise ; ils utilisent le SAAS (Software As A Service) pour simplifier les travaux et les automatiser ;

### 1.1.1.3 Design services

Des maquettes de web design et des vidéos publicitaires avec un design facile et simple à comprendre.

### 1.1.1.4 Implémentation services

Vegasoft n'utilise pas seulement Bootstrap ou le JQuery mais ils ont leur propre Framework, qui est mise à jour fréquemment ; Dont, un service en particulier qui a capté notre attention, c'est le cloud services plus particulièrement le Vscloud et nous en faisons l'étude.

## 1.2 VScloud

**VScloud** est un système de gestion de contenu de service en ligne, il est développé par VEGASOFT pour faciliter la gestion des applications web par des administrateurs autonomes, il est mis à jour régulièrement pour apporter des améliorations, de la stabilité et la correction de bugs, De nouvelles fonctionnalités et/ou des fonctionnalités concernant la sécurité, fluidité et expérience utilisateur.

### 1.2.1 Les différentes fonctionnalités et services proposé par Vs-Cloud

- Scanner des codes Qr ;
- Consulter statistiques ;
- Gestion d'Articles ;
- Consulter aide et infos sur vefgaSoft ;
- Gérer les membres et les modification du site web ;
- Gestion de thèmes.

### **1.3 Identification de la problématique (critique)**

Le service web vsCloud nous permet d'implémenter beaucoup de fonctionnalité, une personnalisation de A à Z d'une application web, il permet au particulier ou aux entreprises de présenter leurs biens/services et de cibler plus de personne sur internet et ainsi de proposer les services au client, toute fois ce n'est pas suffisant, VsCloud essaie de regrouper plus de fonctionnalités possibles, plus d'utilisateurs et essaie de permettre aux clients d'informatiser plus de services.

Pour cela on a opté pour la réservation, un service important, il faut se déplacer pour réserver une chambre, un billet, un produit... etc. faire ses réservations et ses achats tout en restant chez soi, est un rêve, voir les disponibilités de chambres, les différentes chambres proposer, les différents prix...

### **1.4 Présentation de la solution proposée**

Nous allons concevoir et réaliser un service de réservation en extension au service web VsCloud et apporter les modification nécessaire pour y parvenir, qui va permettre aux utilisateurs , d'effectuer leur réservation en ligne , et pouvoir être tenu au courant via des notifications du traitement de leur demande , ainsi que la modification de leur profil après inscription , d'un côté , et d'un autre coté permettre aux administrateurs de gérer les demandes de réservations , ainsi que les réservations existante , et les notification .

### **1.5 Les besoins Fonctionnels**

- Permettre aux clients et à l'administrateur de s'identifier et se connecter correctement ;
- Permettre aux membres utilisateurs du service de pouvoir modifier leur profil sur l'application ;
- Permettre à l'utilisateur d'effectuer une recherche sur les produits disponibles à la réservation sur une date donnée ;



- Permettre à l'utilisateur d'effectuer une demande de réservation ;
- Permettre à l'utilisateur de consulter l'état d'avancement de sa demande de réservation et de la modifier ;
- Permettre à l'utilisateur d'être informé par le changement d'états de sa réservation effectuée via des notifications ;
- Permettre à l'administrateur de gérer les réservations (ajout/ modifier/ supprimer/ afficher/ imprimer/ annuler/ valider/ rechercher) ;
- Gestion des notifications ;
- Facturation des réservations validés.

## **1.6 Les besoins Non Fonctionnels**

Consiste à choisir des solutions adéquates pour protéger l'utilisateur :

- Des ambiguïtés doivent être signalées par des messages d'erreurs bien organisés pour bien guider l'utilisateur et le familiariser avec notre application ;
- Le système doit être conforme à une architecture standard et claire permettant sa maintenance et sa réutilisation .

## **1.7 Conclusion**

Ce chapitre nous a servi à présenter le cadre de notre projet. Nous avons présenté l'organisme d'accueil de notre projet, ses services, l'application VsCloud, on a fait une étude de l'existant, déterminer la problématique et la solution que nous avons proposée. Dans le chapitre suivant nous allons définir la partie conception et analyse de notre projet.

## CHAPITRE 2

# ANALYSE ET CONCEPTION DES BESOINS

## Introduction

La phase analyse décrit de façon la plus précise possible, le fonctionnement de l'application. Chaque fonctionnalité de l'application y est décrite en détail, avec les différents diagrammes d'UML. Dans ce chapitre, Nous allons présenter les différents acteurs de notre système, leurs rôles, ainsi les besoins qui seront modalisés par un diagramme de cas d'utilisation, la description textuelles de ces derniers ainsi les diagrammes de séquence système.

### 2.1 Langage de modélisation unifié UML

UML, abréviation d'Unified Modeling Language, est Née de la fusion des méthodes objet dominantes (OMT, Booch et OOSE), puis normalisé par l'OMG en 1997, UML est rapidement devenu un standard incontournable [1].

UML est un langage de modélisation standardisé composé d'un ensemble intégré de diagrammes, développé pour aider les développeurs de systèmes et de logiciels à spécifier, visualiser, construire et documenter les artefacts des systèmes logiciels et d'autres systèmes non logiciels. UML représente une collection de meilleures pratiques d'ingénierie qui se sont avérées efficaces dans la modélisation de systèmes volumineux et complexes. L'utilisation de l'UML aide les équipes de projet à communiquer, à explorer les conceptions potentielles et à valider la conception architecturale du logiciel [2].

### 2.2 Les diagrammes

#### 2.2.1 Identification des acteurs

Un acteur est défini comme toute personne ou tout élément qui interagit avec le système (personne, organisation ou application) de l'extérieur du système [3].

Dans notre cas, nous présentons les acteurs suivants :

- **Utilisateur** : toute personne qui consulte le site sans s'authentifier.(Anonyme)
- **Admin** : représente le client de VSCloud , il aura un nom et un mot de passe attribuer par le gérant, lui offrant ainsi un accès à l'application Vscloud et donc de créer , personnaliser et de gérer son site.

- **Client** : Est un utilisateur qui s'est inscrit sur le site créé par un particulier (Admin). Ce compte permet au client de faire ses achats, réservations et la consultation du site/profil/historique.
- **Membre** : est un utilisateur possédant un compte avec des privilèges tels que : l'accès à des pages restreintes, la gestion de son espace personnel...

### 2.2.2 Diagramme de contexte :

Il permet la représentation des mouvements de données à l'intérieur d'un système d'information et entre le système et le monde extérieur et il répond à la question "**quels sont les acteurs et éléments environnants au système ?**"[4].

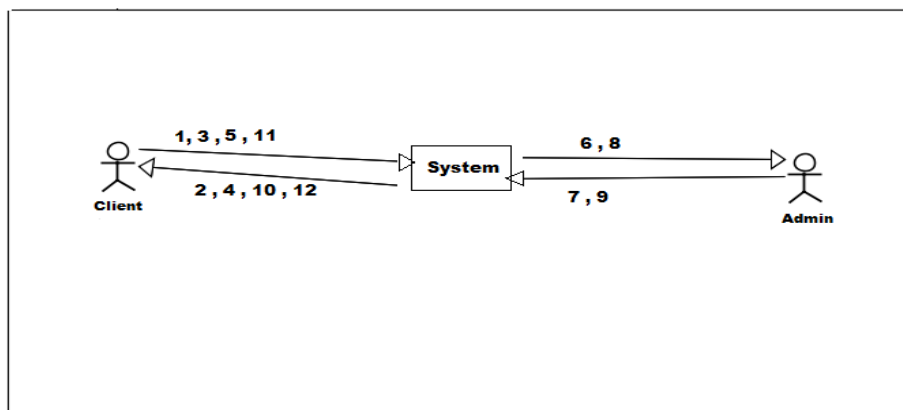


FIGURE 2.1 – Diagramme de contexte

N°	Message acteur->système	N°	Message système ->acteur
1	Demande d'une réservation	2	Redirection vers la page d'authentification/inscription
3	S'authentifier/s'inscrire	4	Redirection vers le formulaire de réservations
5	Validation de réservations	6	Envoi de demande de réservations
7	Consulte les demandes/disponibilités	8	Afficher les disponibilités
9	Changement d'état de réservations	10	Envoi d'une notification
11	Confirmer la réservation	12	Envoi de la facture

TABLE 2.1 – Identification des messages échangés

### 2.2.3 Identification des messages

### 2.2.4 Relation entre les acteurs

La seule relation qui existe entre les acteurs est l'héritage. L'acteur utilisateur permet de représenter toute personne non encore identifiée par le système.

Un héritage de fonctionnalités a été défini dans la figure ci-dessous, entre l'utilisateur et le client :

La relation d'héritage, entre deux acteurs exprime le fait que l'acteur du côté opposé à la pointe de la flèche « est une sorte de ». Il est spécialisé au sens où il peut réaliser tout ce que l'acteur plus général peut réaliser, plus d'autres fonctionnalités.

### 2.2.5 Identification des cas d'utilisations

Dans ce qui suit, nous allons présenter les différents cas d'utilisations, pour chaque acteur du système, de notre application :

### 2.2.6 Diagramme de cas d'utilisation

Le diagramme de cas d'utilisation est un diagramme UML utilisé pour donner une vision globale du comportement fonctionnel d'un système logiciel. Il permet de mettre en évidence les relations fonctionnelles entre les acteurs et le système étudié, représenté par une boîte rectangulaire, produisant un résultat sur un acteur, appelé acteur principal, et ceci indépendamment de son fonctionnement interne. Les diagrammes de cas d'utilisation

modélisent à **QUOI** sert le système.

Un cas d'utilisation représente une unité discrète d'interaction entre un utilisateur (humain ou machine) et un système. Les utilisateurs sont appelés acteurs (actors), ils interagissent avec les cas d'utilisation (use cases). [5]

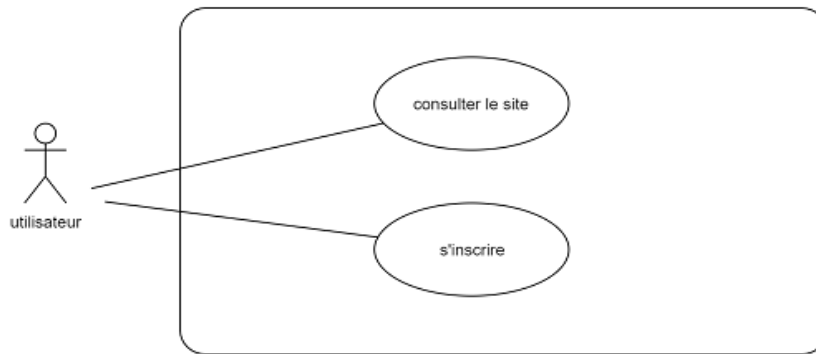


FIGURE 2.2 – Diagramme de cas d'utilisation de l'acteur «Utilisateur»

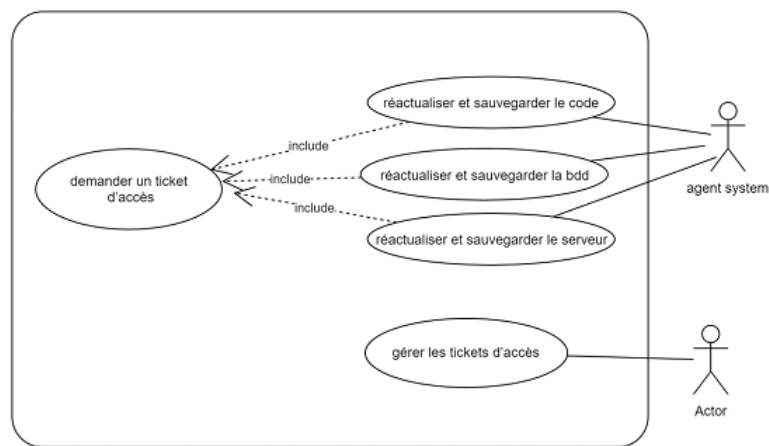


FIGURE 2.3 – Diagramme de cas d'utilisation de l'acteur «Agent System»

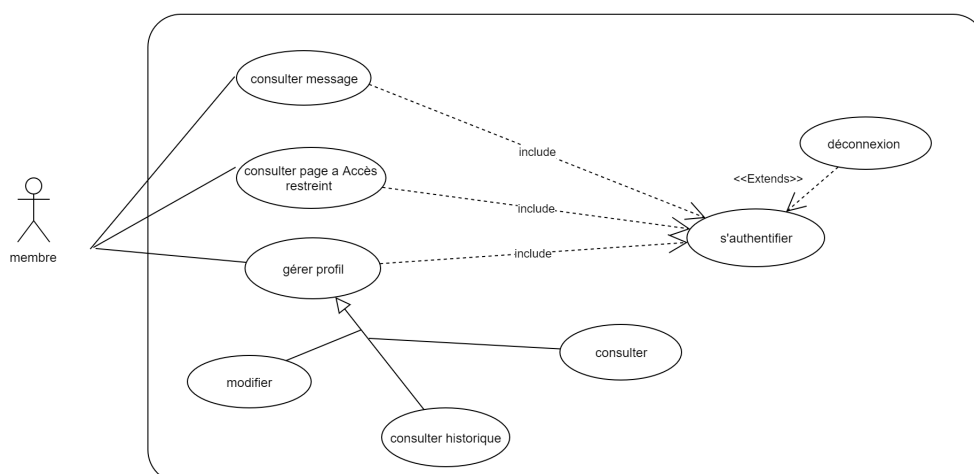


FIGURE 2.4 – Diagramme de cas d'utilisation de l'acteur «Membre»





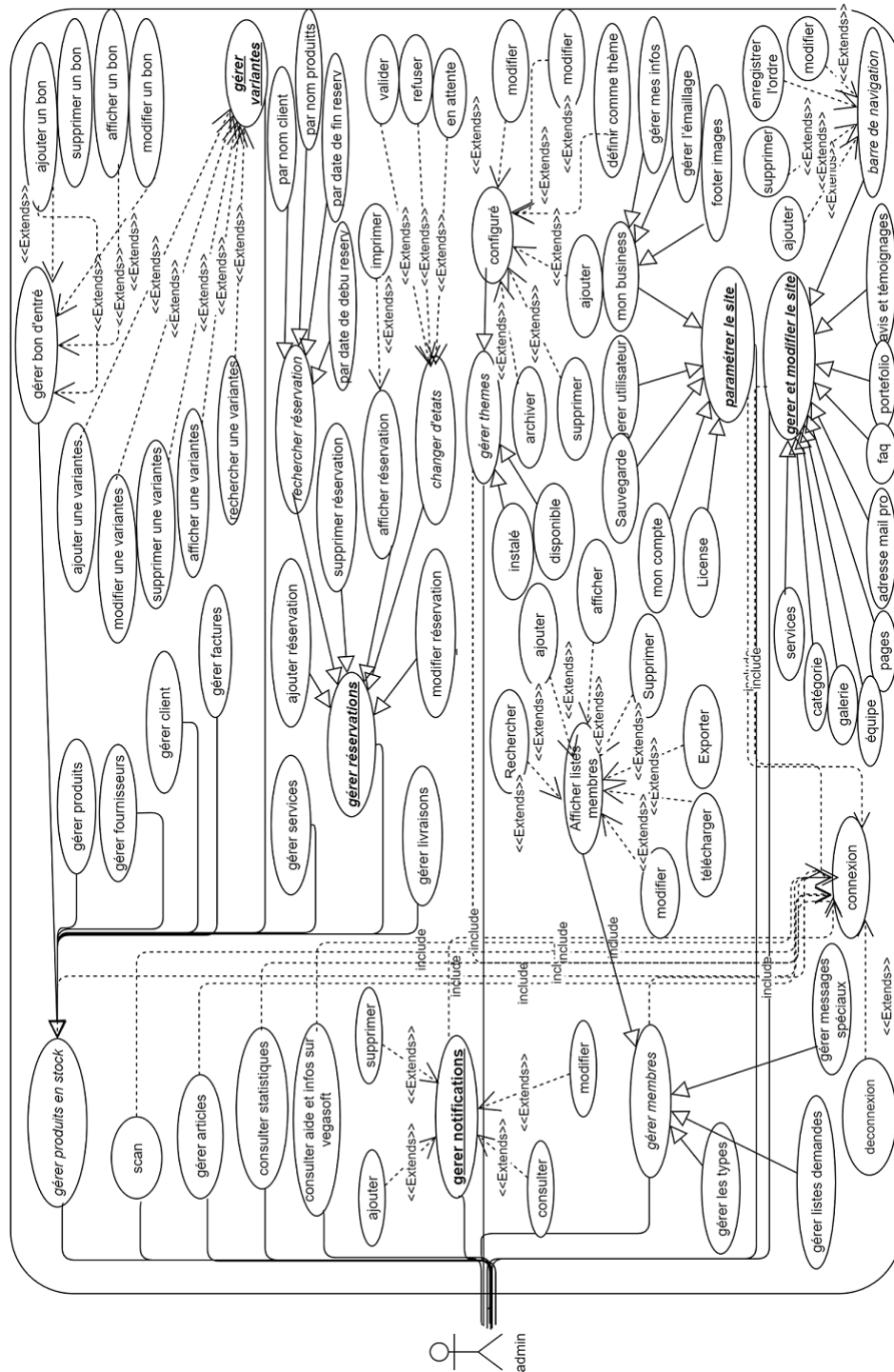


FIGURE 2.6 – Diagramme de cas d'utilisation de l'acteur «Admin»

### 2.2.6.1 Description textuelle des cas d'utilisation

La description textuel d'un cas d'utilisation permet de :

- Clarifier le déroulement de la fonctionnalité ;
- Décrire la chronologie des actions qui devront être réalisées ;
- D'identifier les parties redondantes pour en déduire de cas d'utilisations plus précises qui seront utilisées par inclusion, extension ou généralisation ;
- D'indiquer d'éventuelles contraintes déjà connues et dont les développeurs vont devoir tenir compte lors de la réalisation de l'application ; aussi les descriptions peuvent aider à découvrir d'autres cas d'utilisations que l'on pourrait ajouter. [6]

cas d'utilisation « S'Authentifier »	
But	Permettre à des utilisateurs de créer de nouveau compte client, et au client/ membre déjà détenteur de compte d'accéder a celui-ci.
Acteur	Client Membre
Pré-condition	L'utilisateur saisie ses passe de connexion (login et password)
Description	l'authentification permet a un utilisateur d'accéder a sont compte pour avoir les privilèges qu'ils lui sont appropriés au tant que membre ou client si celui-là est déjà inscrit, ou lui permet de créer un compte dans le cas inverse.
Scenario nominal	<p>Scenario 1 : Authentification</p> <ul style="list-style-type: none"> <li>-</li> <li>L'utilisateur saisit ses données (user name, password)</li> <li>-</li> <li>l'utilisateur soumet la demande de « se connecter » ;</li> <li>-</li> <li>Le system vérifie si les champs ne sont pas vide, sinon Excp1 ;</li> <li>-</li> <li>Le system vérifie si les données saisie sont correct, sinon Excp2 ;</li> <li>-</li> <li>Le system renvoie vers la page demander ou vers l'accueil ;</li> </ul>
Scenario d'exception	<p>Excp1 : message d'erreur : renseigner les champs obligatoire svpl, renvoie vers la page de connexion ;</p> <p>Excp2 : message d'erreur : userName ou passWord incorrect, renvoie vers la page de connexion ;</p>
Post condition	Renvois vers la page d'accueil ou la page de la quel la connexion est faites.

TABLE 2.2 – Description textuelle de cas d'utilisation «S'authentifier».

cas d'utilisation « Demande réservation »	
But	Faire part a un administrateur d'une demande de réservations
Acteur	Client
Pré-condition	Authentification Rechercher et sélectionner un produit
Description	
Scenario nominal	<p>Scenario : Effectuer une réservation</p> <p>Le client demande une reservation</p> <p>Le système renvoie le formulaire de reservation</p> <p>Le client Remplie le formulaire de demande de réservations ;</p> <p>L'utilisateur soumet sa demande.</p> <p>Le system vérifie si les champs ne sont pas vides, sinon Excp1 ;</p> <p>Le system vérifie si les données saisies sont valides, sinon Excp2 ;</p> <p>Le system transmet la demande a l'admin en mettant l'etat par défaut à en attente et génère une notifications ;</p>
Scenario d'exception	<p>Excp1 : message d'erreur : renseigner les champs obligatoire svpl, renvoie le formulaire ;</p> <p>Excp2 : message d'erreur : renseignez des donnée correct svpl ;</p>
Post condition	Le system envoie la demande de réservations à l'admin en mettant l'état par défaut (en attente).

TABLE 2.3 – Description textuelle de cas d'utilisation «Demande de réservation »

cas d'utilisation « rechercher un produit »	
But	Faciliter la recherche d'un produit en particulier
Acteur	Client Utilisateur
Pré-condition	Aucun
Description	Rechercher un produit en particulier.
Scenario nominal	Scenario : Recherche un produit L'utilisateur remplit les champs de recherche. Le system vérifie l'existence ou non de produit/service si aucun article/service ne correspond Excp1 ;
Scenario d'exception	Excp1 : message « aucun produit trouver » ;
Post condition	Afficher la liste des produits correspondant à la recherche

TABLE 2.4 – Description textuelle de cas d'utilisation «rechercher un produit ».

But	Gérer les demandes de réservation et les réservations.
Acteur	Admin
Pré-condition	Connexion
Description	<p>L'administrateur peut ajouter une nouvelle reservation pour les clients qui ne passent pas par le site.il peut aussi gerer les demandes de reservations que les clients soumettent via la plateforme afin de les valider ou de les refuser , il peut egalement consulter,modifier,afficher,imprimer des reservations déjà traites et il a la possibilité de faire une recherche pour faciliter la selection d'une reservation déjà sauvegardée dans la base de données</p>
Scenario nominal	<p>Scenario</p> <p>Ajouter réservation :</p> <p>L'administrateur demande l'ajout d'une réservation.</p> <p>Le système renvoie le formulaire</p> <p>L'admin remplit le formulaire et soumet la demande.</p> <p>Le système vérifie si les champs ne sont pas vides, sinon Excp1 ;</p> <p>Le système vérifie si les données saisies sont valides, sinon Excp2 ;</p> <p>Le système rajoute la réservation à la liste des réservations, en mettant l'état par default en attente s'il n'est pas indiquer lors de l'ajout</p> <p>Rechercher réservation :</p> <p>L'admin Remplit le champ de recherche et valide la recherche</p> <p>Le système envoie la demande à la base de données et fait la recherche</p> <p>si</p> <p>le champ rechercher est introuvable :</p> <p>le system envoie un message d'erreur ;</p> <p>sinon</p> <p>le système affiche la liste de réservations qui correspond au champ introduit ;</p>

	<p>Modifier réservation :</p> <p>L'admin choisi la réservation qui souhaite modifier et clique sur « modifier »</p> <p>Le système affiche le formulaire de réservation en remplissent les cases avec les informations de la réservation sélectionné</p> <p>L'admin effectue les modifications.</p> <p>L'admin sauvegarde.</p> <p>Le système enregistre les modifications et actualise la page</p> <p>Afficher réservation :</p> <p>L'admin sélectionne une réservation et appuie sur afficher</p> <p>Le système affiche la réservation</p> <p>Imprimer :</p> <p>L'admin demande à imprimer</p> <p>Le système affiche toutes les réservations</p> <p>L'admin choisi la réservation souhaitée</p> <p>Le système demande la validation avant de lancer l'impression</p> <p>L'admin valide</p> <p>Le système lance l'impression</p> <p>Supprimer :</p> <p>L'admin choisi une réservation de la liste des réservations</p> <p>Le système demande la confirmation</p> <p>L'admin valide</p> <p>Le système supprime la réservation de la base de données</p> <p>Le système actualise la page et affiche un message de succès de suppression</p> <p>Changer d'état :</p> <p>L'admin choisi la réservation souhaitée</p> <p>L'admin appuie sur les trois boutons à droite et sélectionne de changer l'état</p> <p>L'admin choisi l'état souhaité (Valider / refuser/ en attente)</p> <p>Le système actualise la page et change l'état de la réservation</p>
Scenario d'exception	Excp1 :



Scenario d'exception	message d'erreur : renseigner les champs obligatoire svpl, renvoie le formulaire ; Excp2 : message d'erreur : renseignez des donnée correct svpl ;
Post condition	Le système envoie la demande de réservation à l'admin en mettant l'état par défaut (en attente)

TABLE 2.5 – Description textuelle de cas d'utilisation «Gérer les réservations »

### 2.2.7 Diagramme de séquence

Le diagramme de séquence fait parties des diagrammes comportementaux (dynamique). Il permet de représenter des échanges entre les différents objets et acteurs du système en fonction du temps. Ils permettent de décrire **COMMENT** les éléments du système interagissent entre eux et avec les acteurs.

Les objets au cœur d'un système interagissent en s'échangeant des messages. Les acteurs interagissent avec le système au moyen d'IHM (Interfaces Homme-Machine) [7].

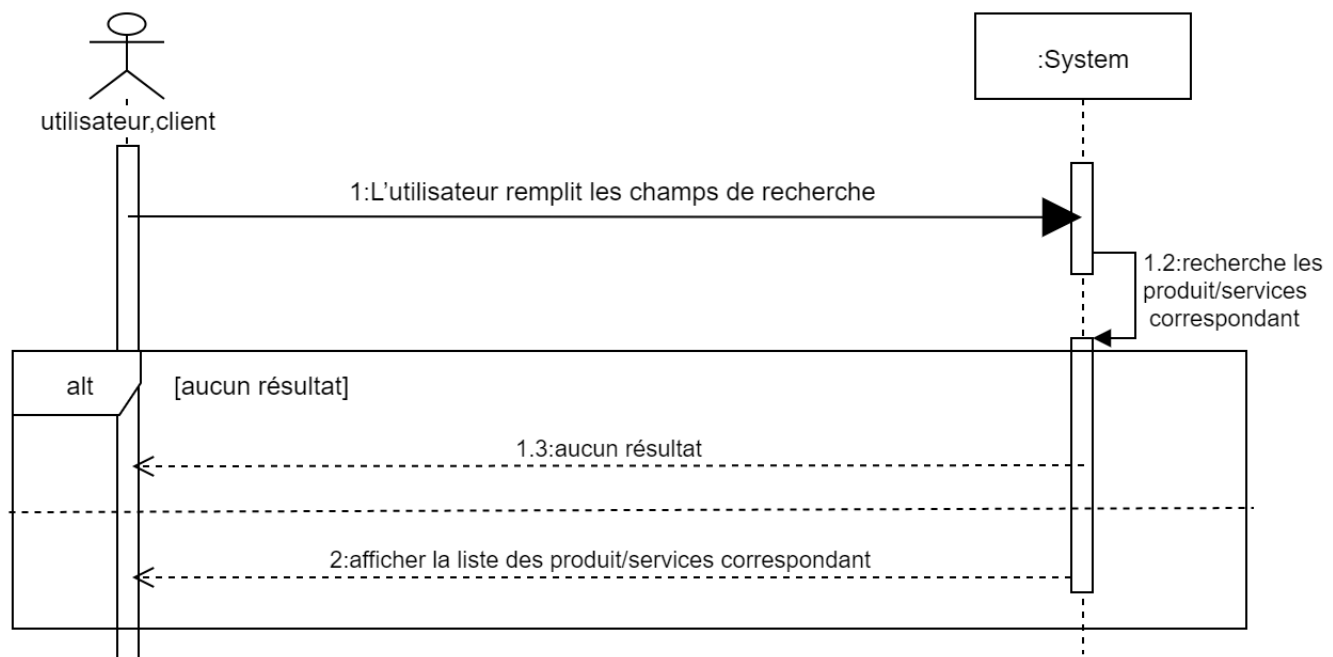


FIGURE 2.7 – Diagramme de séquence du cas d'utilisation rechercher produit

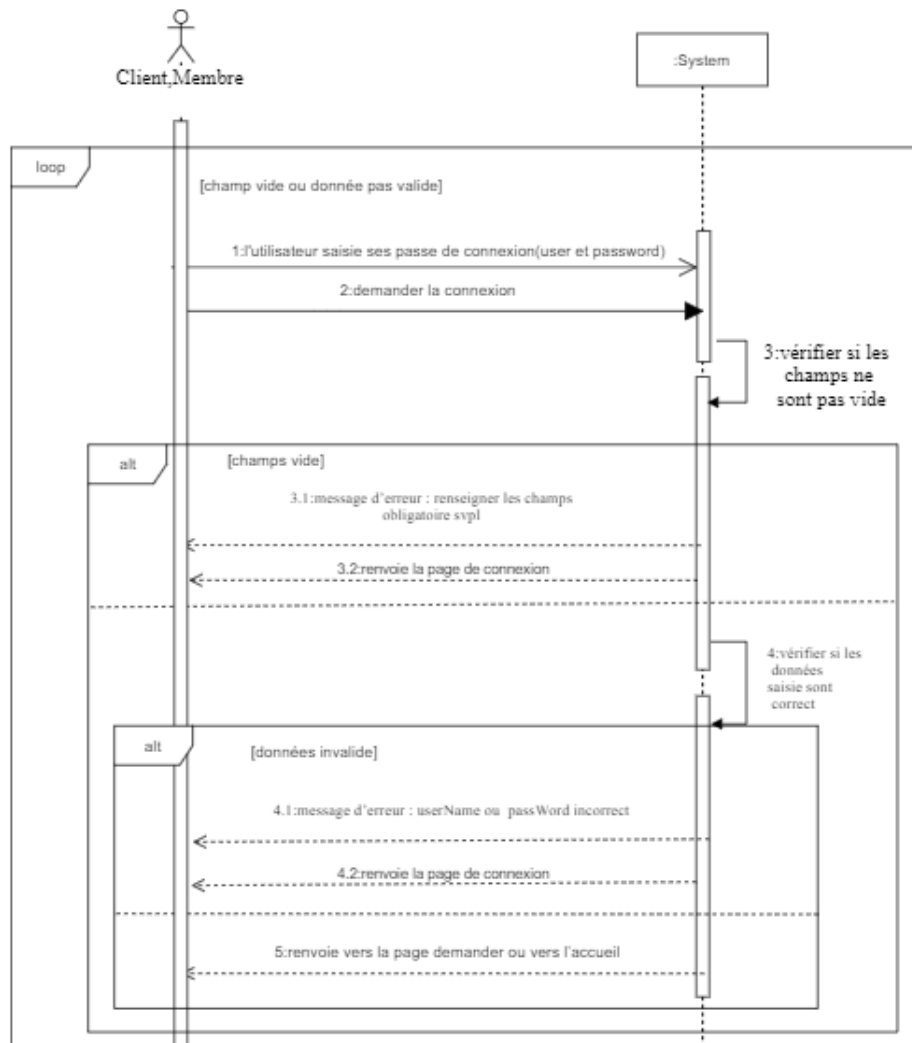


FIGURE 2.8 – Diagramme de séquence du cas d'utilisation authentification

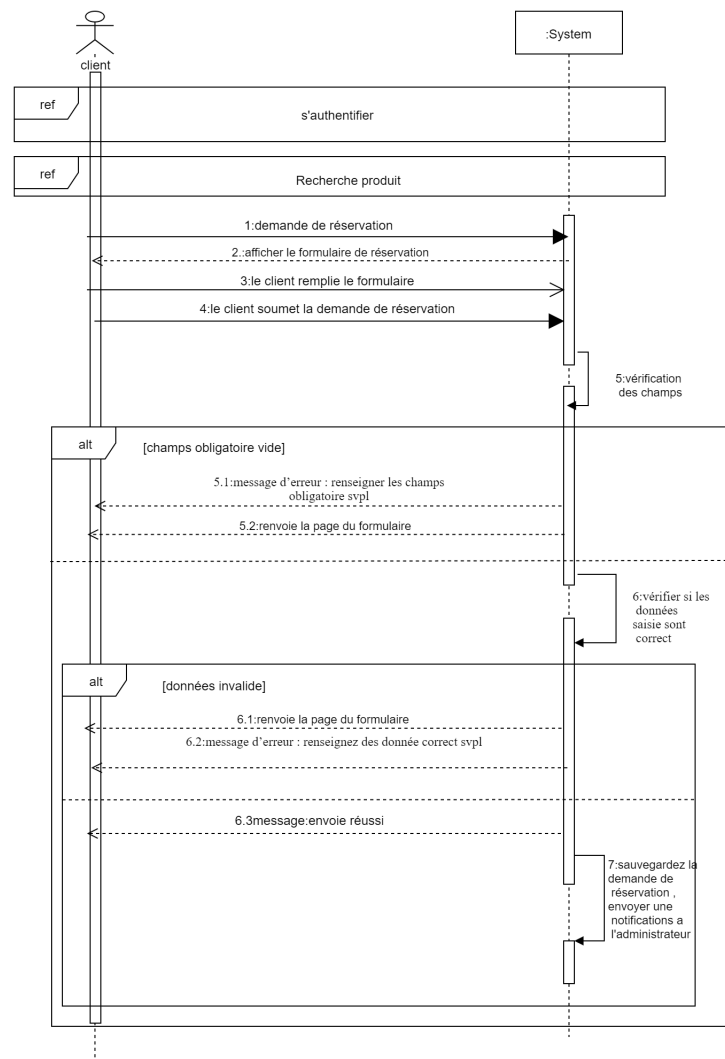


FIGURE 2.9 – Diagramme de séquence du cas d'utilisation demander réservation

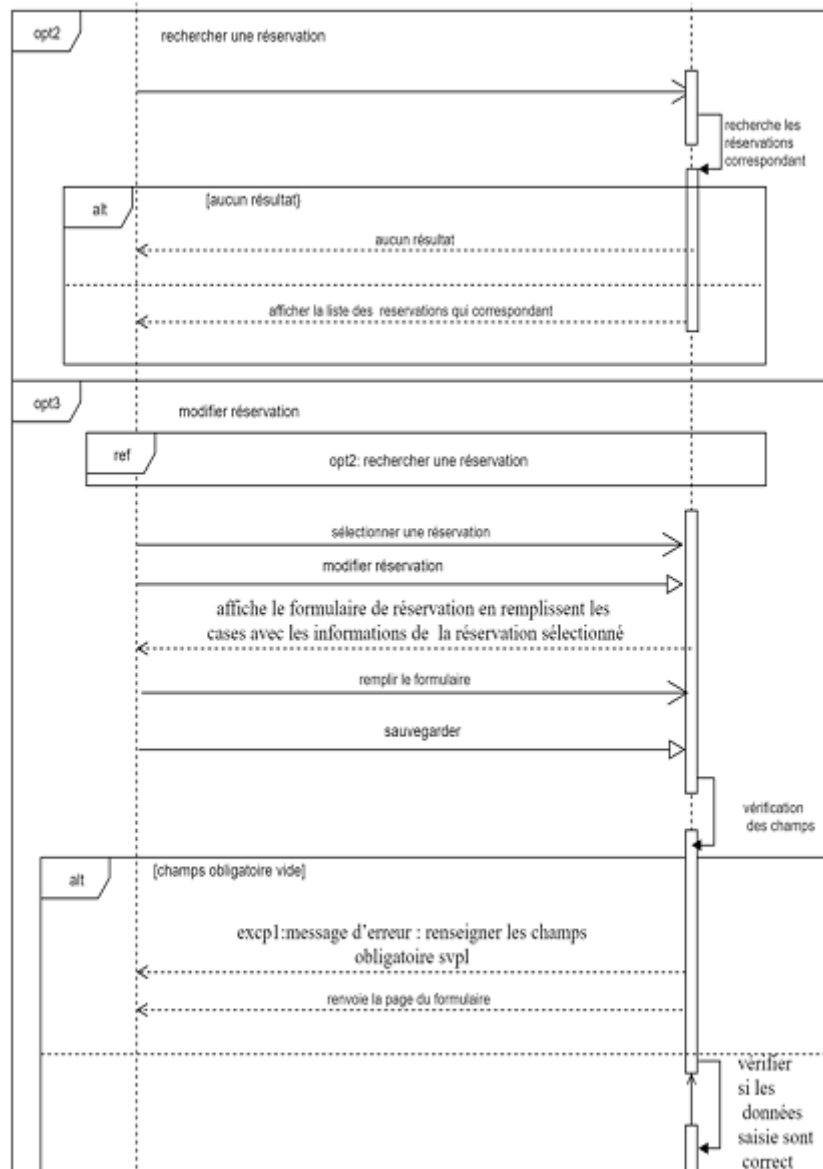


FIGURE 2.10 – Diagramme de séquence du cas d'utilisation gérer réservation part 1

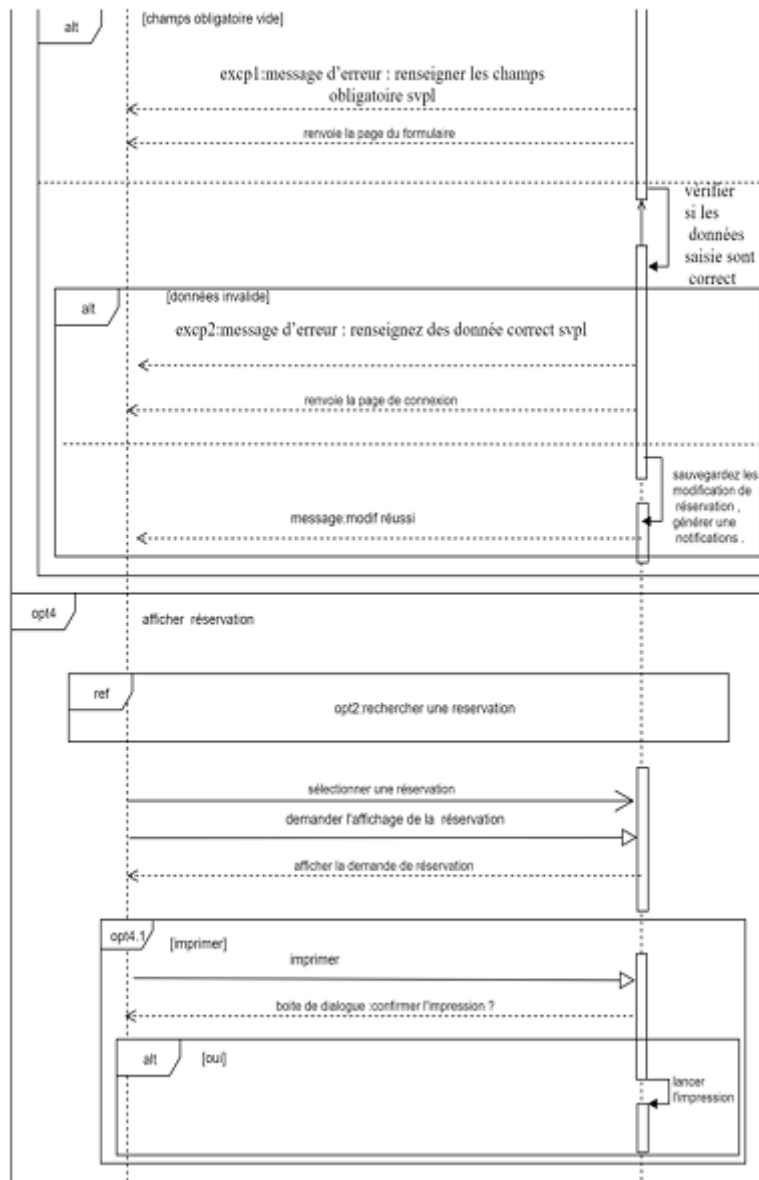


FIGURE 2.11 – Diagramme de séquence du cas d'utilisation gérer réservation part 2

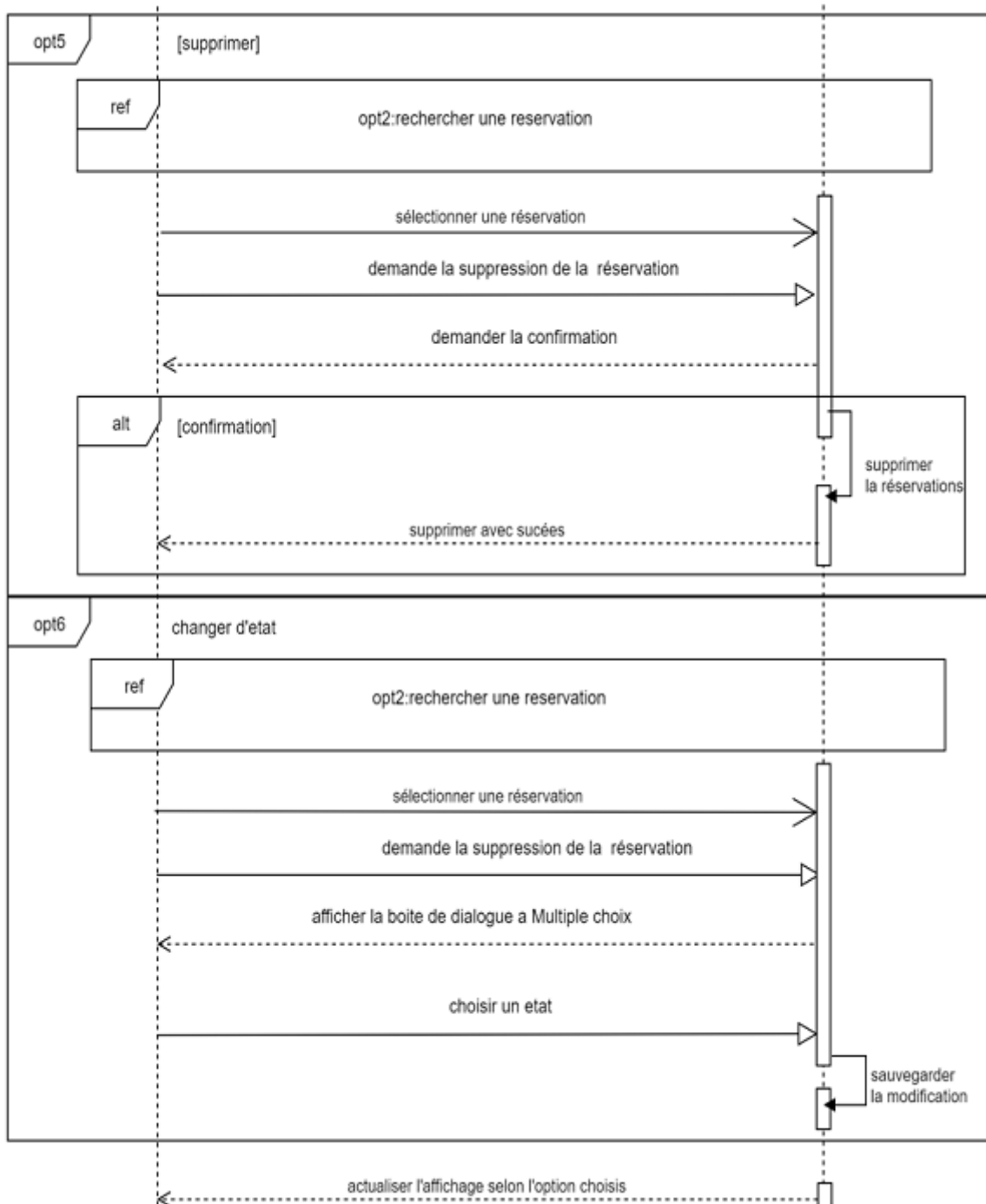


FIGURE 2.12 – Diagramme de séquence du cas d'utilisation gérer réservation part 3

## **2.3 Conclusion**

Nous avons, dans ce chapitre, analysé et lister les résultats attendus par le client en termes de fonctionnalités, performance, robustesse, sécurité, pour les réaliser on a utilisé les différents diagrammes de langage UML, tel que le diagramme cas d'utilisation, diagramme de contexte, diagramme de séquence, et diagramme de classe afin de faciliter la réalisation et la maintenance de notre application. Dans le chapitre suivant on passera à la réalisation de notre application.



## CHAPITRE 3

# CONCEPTION ET SCHÉMA RELATIONNEL DE DONNÉES

## Introduction

Après la phase de l'analyse et de spécification viens La phase de la conception qui est une étape de réflexion importante dans le cycle de développement logiciel qui permet de structurer, organiser, planifier le projet. Dans ce chapitre, nous allons présenter en détails la conception du système ceci sera réalisé en utilisant les diagrammes de séquence d'interaction entre acteur(s) et objets du système. Par la suite nous élaborons le diagramme de classe de domaine et le passage vers le modèle relationnel.

### 3.1 Diagrammes d'Interaction

un diagramme d'interaction UML est un type de diagramme UML utilisé pour représenter le comportement interactif d'un système. Les diagrammes d'interaction se concentrent sur la description du flux de messages au sein d'un système, en fournissant du contexte pour une ou plusieurs lignes de vie. Ils peuvent également servir à illustrer des séquences ordonnées et permettre de visualiser des données en temps réel dans une modélisations de systèmes UML.[8]

Pour chaque cas d'utilisation, on établit un diagramme d'interaction (Système est éclaté en objets intervenant pour réaliser le cas d'utilisation : trois types d'objets : objets d'interface, objets contrôleurs et objets d'entité).

Dans ce diagramme nous allons nous servir de trois types de classes :

#### 3.1.1 Classes d'interface (boundary)

Des classes qui permettent l'interaction entre l'application et ses utilisateurs. Pour chaque cas d'utilisation, il y a au moins une classe d'interface. Ce type de classe est schématisé comme suit :

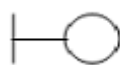


FIGURE 3.1 – Classe interface "IHM"

### 3.1.2 Classes de Contrôle (Control)

Ce sont des classes qui contiennent les traitements et la cinématique de l'application. Elles font la transition entre les classes d'interface et les classes entités. Elles sont schématisées comme suit



FIGURE 3.2 – Classe interface "Contrôle"

### 3.1.3 Classes entités (entity)

Elles représentent les objets métiers, et ce sont très souvent des entités persistantes, c'est-à-dire qui vont garder leurs informations (données) après l'exécution d'un cas d'utilisation particulier. En général, elles sont enregistrées dans une base de données. Leurs schématisations se fait grâce à ce stéréotype



FIGURE 3.3 – Classe interface "Entité"

## 3.2 Les Diagrammes d'interaction des cas d'utilisation

### 3.2.1 Diagramme d'interaction de cas d'utilisation «S'authentifier »

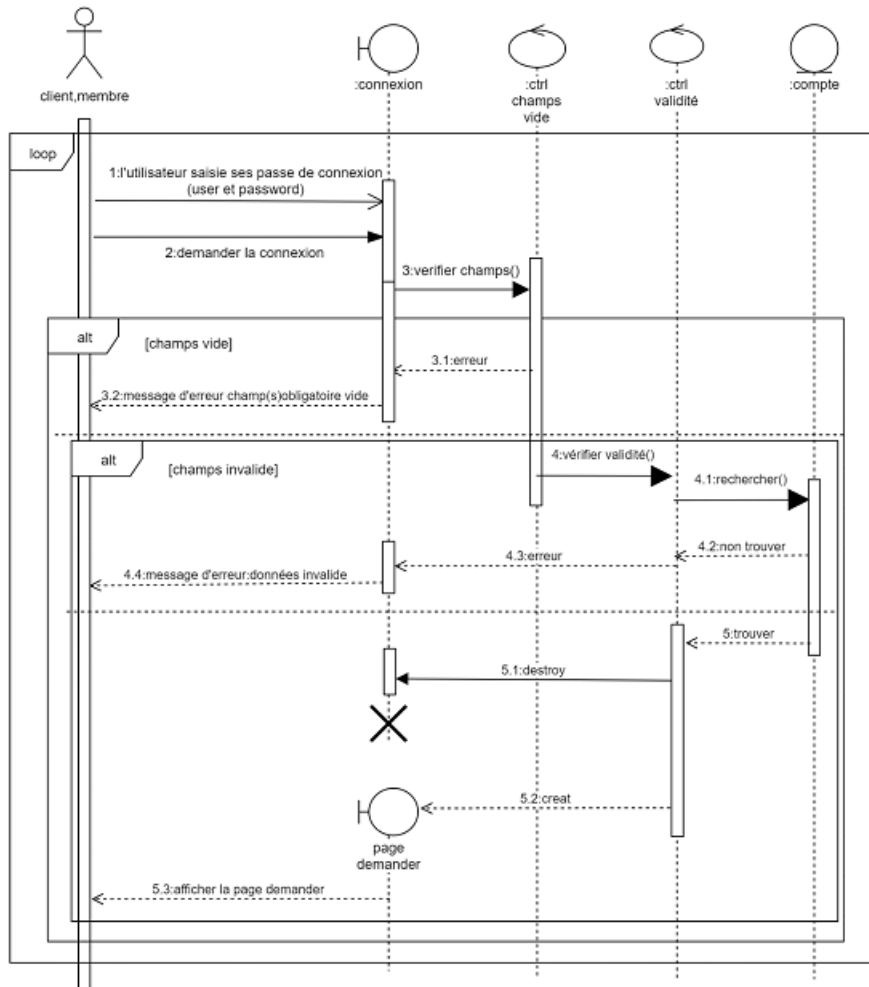


FIGURE 3.4 – Diagramme d'interaction de cas d'utilisation «S'authentifier»

### 3.2.2 Diagramme d'interaction de cas d'utilisation «rechercher produit »

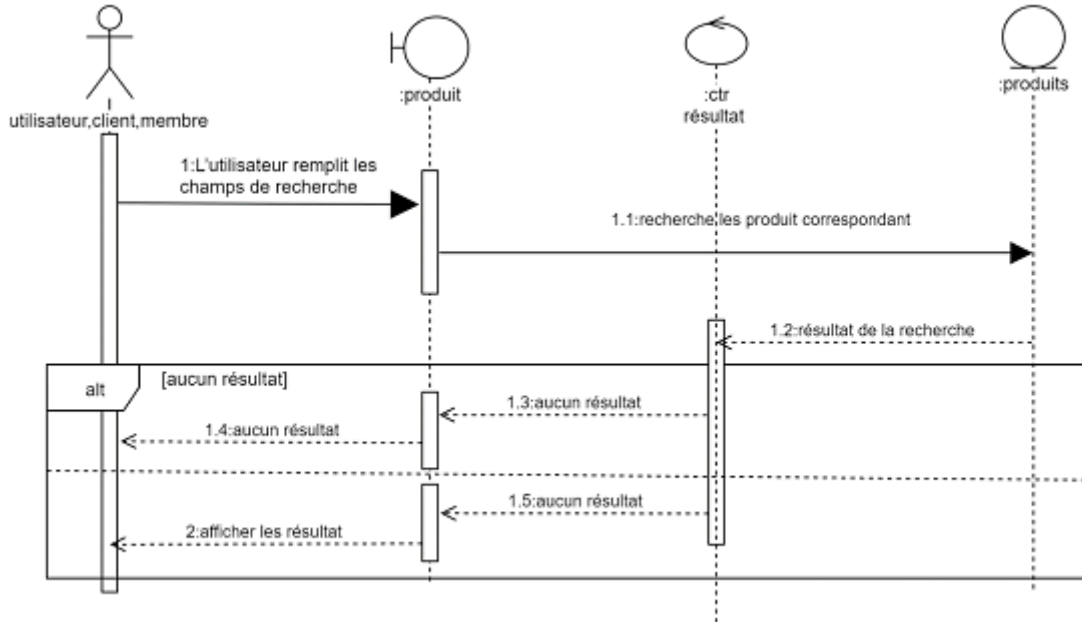


FIGURE 3.5 – Diagramme d'interaction de cas d'utilisation «rechercher produit »

3.2.3 Diagramme d'interaction de cas d'utilisation «demande réservations »

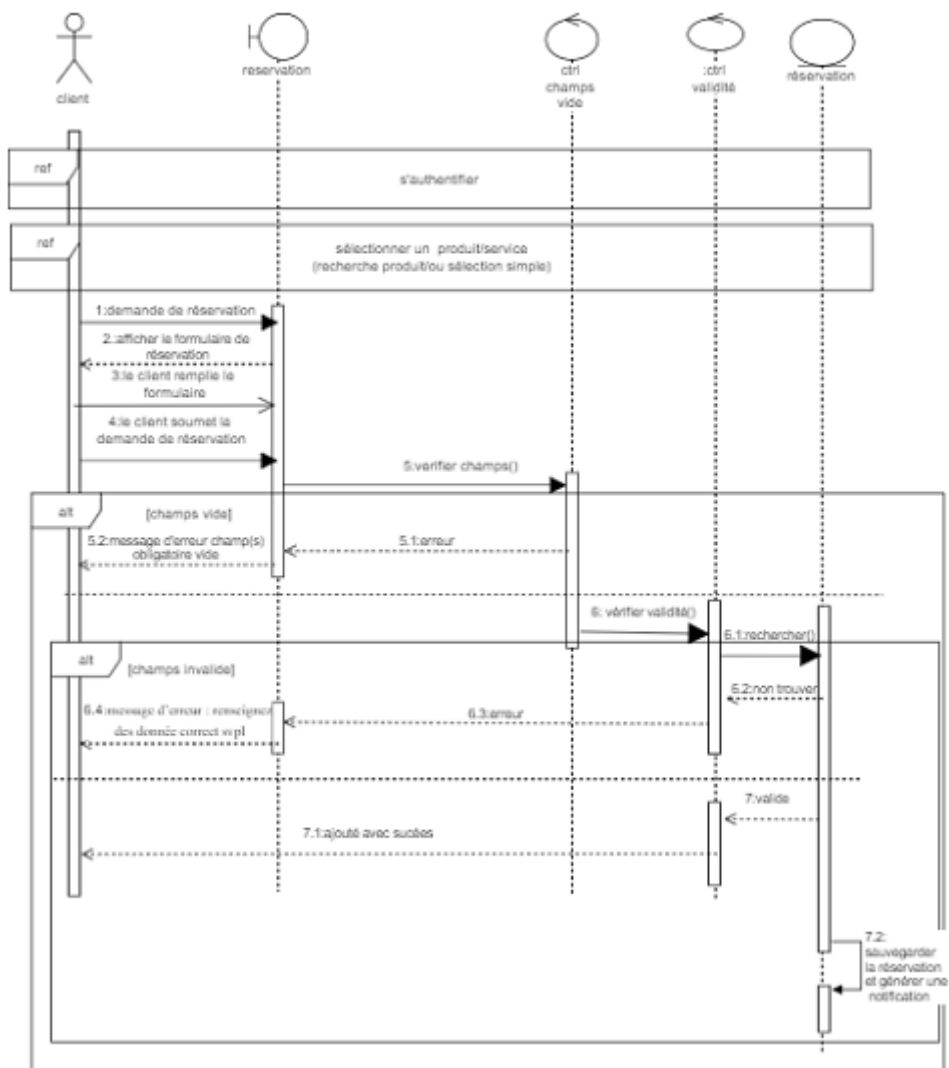


FIGURE 3.6 – Diagramme d'interaction de cas d'utilisation «demande réservations »

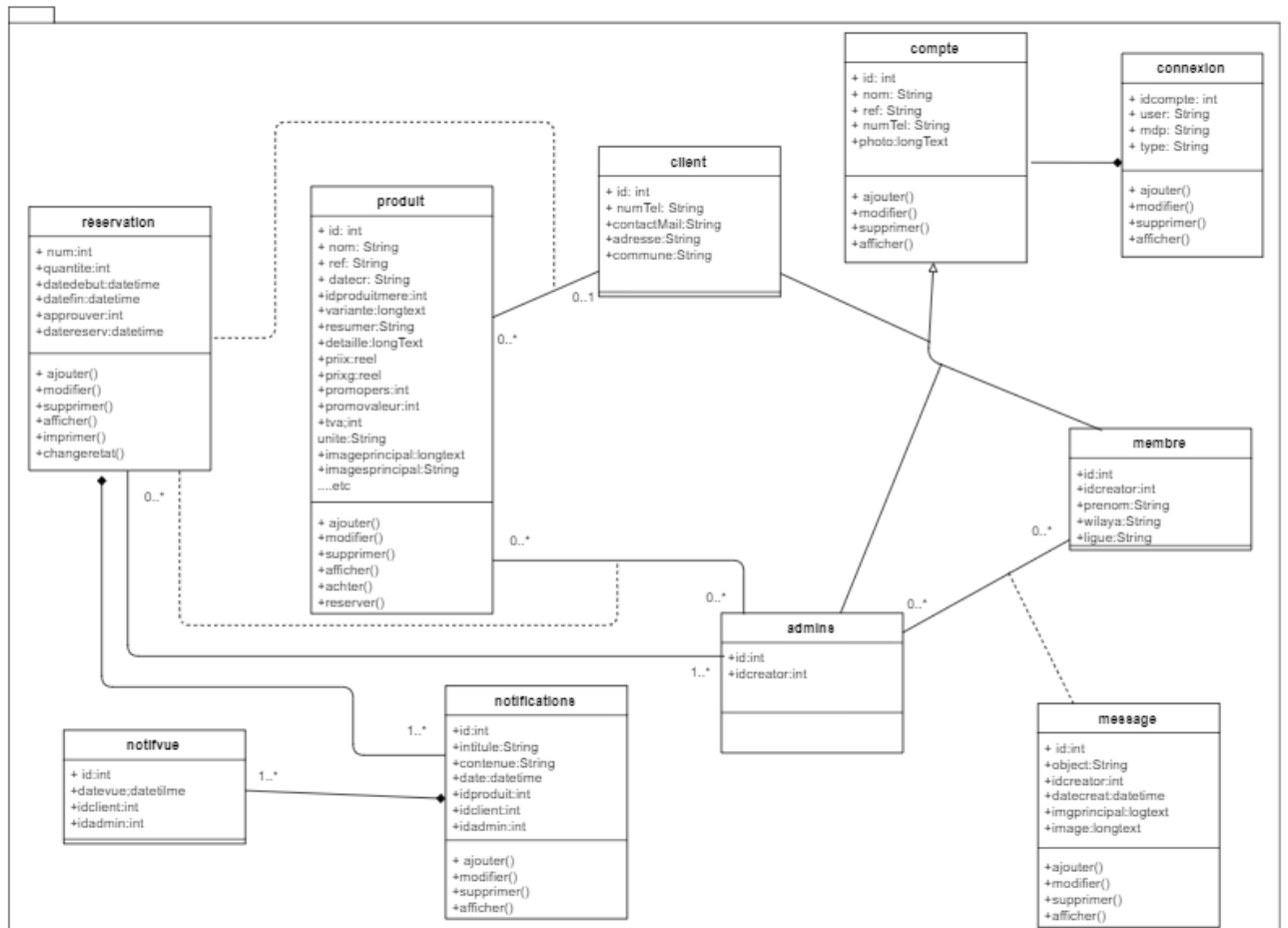
### **3.3 Diagramme de classe :**

Les diagrammes de classes sont l'un des types de diagrammes UML les plus utiles, ils décrivent clairement la structure d'un système particulier en modélisant ses classes, ses attributs, ses opérations et les relations entre ses objets.

Une classe est un ensemble de fonctions et de données (attributs) qui sont liées ensemble par un champ sémantique. Les classes sont utilisées dans la programmation orientée objet. Elles permettent de modéliser un programme et ainsi de découper une tâche complexe en plusieurs petits travaux simples.

#### **3.3.1 Diagramme de classes du domaine (classes issues des Objets entités ci-dessus)**

Le diagramme de classes est le point central dans un développement orienté objet. En analyse, il a pour objet de décrire la structure des entités manipulées par les utilisateurs. En conception, le diagramme de classes représente la structure d'un code orienté objet [23]. Il décrit les classes que le système utilise, ainsi que leurs liens (héritage, agrégation, composition, etc).



Text

FIGURE 3.7 – Diagramme de classes du domaine



### 3.3.2 Diagramme de classe général

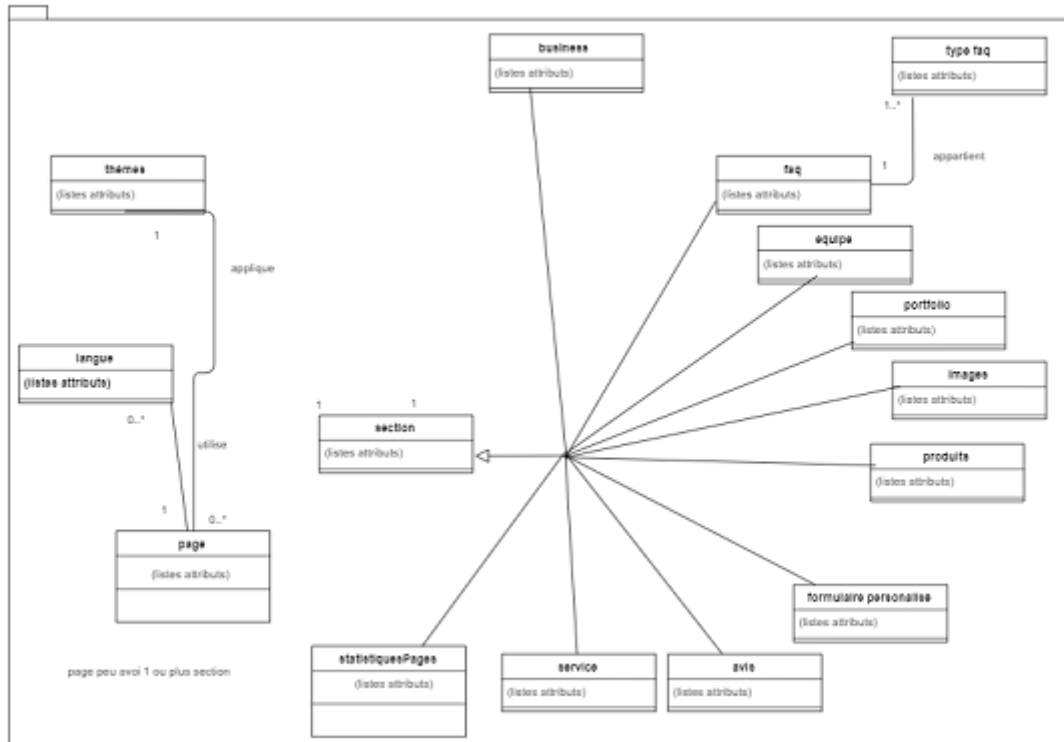


FIGURE 3.8 – Diagramme de classe général



**3.3.4** Classes, attributs et Responsabilités (Dictionnaire de données) :

Classe	Responsabilité	Attributs		
		Intitulé	Définition	Type
Compte	Classe qui contient les informations des utilisateurs ayant un compte	Id	Identifiant	Numérique
		Types	Les types des comptes utilisateurs	Chaîne de caractère
		Nom	Le nome de la personne	Chaîne de caractère
		Photo	La photo de la personne	Chaîne de caractère
Admins	Classe qui contient les informations supplémentaires qu'on peut avoir pour un utilisateur de type admin	Id	Identifiant	Numérique
		Produits	Autorisation	Array
		Statistique web	Autorisation	Array
		Paramètre	Autorisation	Array
		Membres FNCCC	Autorisation	Array
		Thèmes App.	Le thème qu'il a choisi pour son site	Chaîne de caractère
		Prénom	Prénom de la personne	Chaîne de caractère
		IdCreator	L'identifiant de l'admin principale qui a crée le compte de cet admin	Numérique
Étc ...				
Client	Classe qui contient les informations supplémentaires qu'on peut avoir pour un utilisateur de type client	Id	Identifiant	Numérique
		Ref	Autorisation	Array
		Numéro/Tel	Numéro de téléphone	Chaîne de caractère
		Contact Mail	L'adresse email	Chaîne de caractère
		Adresse	L'adresse de la personne	Chaîne de caractère
		DateCr	Date de création du compte	Date

		Tondresse	Autorisation	Array
		Commune	La commune ou habite la personne	Numérique
		Ètc ...		
Membres		Id	Identifiant	Numérique
		Ref	Autorisation	Array
		Prénom	Prenom de la personne	Chaîne de caractère
		Password	Autorisation	Array
		Id Creator	L'identifiant de l'admin qui a cree se compte membre	Numérique
		Numéro Tel	Numero de telephone du membre	Chaîne de caractère
		Wilaya	La wilaya de membres	Chaîne de caractère
		Ligne	Autorisation	Array
Produits	Cette classe nous permet de sauvegarder les produits	Id	Identifiant	Numérique
		Types	Le type de produits	Chaîne de caractère
		Nom	Nom de produits	Chaîne de caractère
		Ref	Autorisation	Array
		Date Cr	Date de l'ajout de se produit a la base de donné	Date
		Id Produit Mere	L'identifiant du produit mere	Numérique
		Variantes	Les variantes de ce produit	Chaîne de caractère
		Resume	Un résumé qui décrit le produit	Chaîne de caractère
		Detaillé	Les details supplémentaires sur le produit	Chaîne de caractère
		Prix D	Le prix de detail du produit	Reele
		Prix G	Le prix de gros du produit	Reele
		Promo Pers	La promotion destiné aux personnes	Numérique
		Promo Valeur	La valeur de la promotion	Numérique

		Tva	La tva qui s'applique a se produit	Reele
		Unite	Le nombre d'unité qu'on a en stock	Chaine de caractère
		Limite	La limite du stock	Reele
		Cont	Autorisation	Array
		Image Principal	L'image qui represente l'article	Chaine de caractère
		Images	D'autres images du produit	Chaine de caractère
		On Home Page	Autorisation	Array
		Tags	Les tags qui concernent le produits	Chaine de caractère
		Archive	Autorisation	Array
		Delevry service	Le service de livraison	Numérique
		Type Vente	Definir est ce que l'article est destiné a la vente ou la location	Chaine de caractère
Réservation		Num	identifiant	Numérique
		Quantité	La quantité reserver	Numérique
		Date debut	La date du debut de reservation	Date
		Date fin	La date de fin de reservations	Date
		Approuver	Definit si la demande de reservation et approuver,refuser,ou en attente de traitement	Numérique
		Date reserv	La date ou la demande de reservation a était soumise	Date
Notification		Id	identifiant	Numérique
		Intitule	L'intitulé de la notification	Chaine de caractère

		Contenu	Le contenu de la notification	Chaîne de caractère
		Date	Date de l'émission de la notification	Date
<b>Notifvue</b>		Id	identifiant	Numérique
		Datevue	Date ou la notification a été vue	Date
		Vue	Autorisation	Array
<b>Fournisseur</b>		Id	identifiant	Numérique
		Ref		Chaîne de caractère
		Nom	Nom du fournisseur	Chaîne de caractère
		NIF	Autorisation	Array
		NIS	Autorisation	Array
<b>Connexion</b>		Id	identifiant	Numérique
		Type	Type du compte	Chaîne de caractère
				Numérique
		User	Nom d'utilisateur	Chaîne de caractère
		password	Mot de passe	Chaîne de caractère

FIGURE 3.10 – Description détaillée des attributs de classes

## 3.4 Schéma relationnel (Règle de passages du diagramme de classes vers le schéma relationnel)

A partir du diagramme de classe nous allons réaliser le modèle relationnel qui est le modèle logique de données, ce modèle décrit comment sont représentées les données dans une base de données.

Pour décrire une relation, nous allons indiquer tout simplement son nom, suivi du nom de ses attributs entre parenthèses.

L'identifiant d'une relation est composé d'un ou plusieurs attributs qui forment la clé primaire. Une relation peut faire référence à une autre en utilisant une clé étrangère, qui correspond à la clé primaire de la relation référencée.

### 3.4.1 Règles de passages au modèle relationnel

Les règles de passage au modèle relationnel sont :

#### 3.4.1.1 Relation (1..\*) :

il faut ajouter un attribut de type clé étrangère dans la relation fils de l'association. L'attribut aura le nom de la clé primaire de la relation père de l'association.

#### 3.4.1.2 Relation (1..1) :

il faut ajouter une relation qui prend les deux clé primaire des classes mère comme clé étrangère

#### 3.4.1.3 Relation d'héritage

Trois décompositions sont possibles pour traduire une association d'héritage en fonction des contraintes existantes :

- **Décomposition par distinction** : il faut transformer chaque sous-classe en une relation. La clé primaire de la classe mère, migre dans la (les) relation(s) issue(s) de



la (des) sous-classe(s) et devient à la fois clé primaire et clé étrangère,

- **Décomposition descendante (push-down)** : Il faut faire migrer tous les attributs de la classe mère dans la (les) relation(s) issue(s) de la (des) sous classe(s),
- **Décomposition ascendante (push up)** : Dans ce cas on supprime les relations issues des sous classes et faire migrer tous les attributs dans la relation issue de la classe mère.

#### 3.4.1.4 Le passage au modèle relationnel

Après avoir appliqué toutes les règles de passage au modèle relationnel, nous avons obtenu le schéma suivant :

- **Admin**(id, user, password, produits, statiqueweb, parametre, membrefncc, theme-sapp, prenom, idcreator, nom, photo) ;
- **Client**(id, ref, numerotel, contactmail, adresse, datecr, toadress, comune, nom,photo) ;
- **Membres**(id , idcompte, ref, prenom, password, idcreator, user, numerotel, wilaya, ligue, nom, photo) ;
- **Produits**(id, types, nom, ref, datecr, idproduitmere, variantes, resume, detaille, prixd, prixg, promopers, promovaleur, tva, unite, limmite, cont, imageprincipal, images, onhomepage, tags, archive, delevryservice, typevente) ;
- **Reservation**(num, quantite, datedebut, datefin, approuver, datereserv, idadmin , idclient, idproduit, idfacture) ;
- **Notifications**(id, intitule, contenue, date, idnotifvue, idclient, idreserv) ;

- **Notifvue**(id, datevue, vue, idclient, idadmin, idnotif) ;
  
- **Fournisseur**(id, ref, nom, nif, nis,) ;
  
- **Connexion**(id, type, iduser, user, password).

## Conclusion

Nous avons présenté dans ce chapitre la phase de conception de notre projet à travers les différents diagrammes( classe, interaction et cas d'utilisation) qui nous ont permis de décrire de manière globale et détaillée le fonctionnement du système. Nous avons donné ensuite les règles de passage du diagramme de classe vers le modèle relationnel qui nous a permis par la suite d'obtenir le schéma des tables pour mettre la base de données existantes.

## CHAPITRE 4

RÉALISATION

## Introduction

Dans ce quatrième et dernier chapitre, nous allons donner quelques définitions relatives aux langages et outils de programmation que nous avons utilisés pour le développement de notre application web. Nous allons également présenter le diagramme de déploiement grâce auquel nous pourrions représenter l'utilisation de l'infrastructure physique par le système et la manière dont les composants du système sont répartis ainsi que les relations entre eux.

Nous concluons par la présentation des interfaces les plus importantes de notre application.

### 4.1 Outils de développement :

#### 4.1.1 Visual Studio Code :

Visual Studio Code est un éditeur de code source qui peut être utilisé avec une variété de langages de programmation, notamment Java, JavaScript, Go, Node.js et C++. Il est basé sur le cadre Electron, qui est utilisé pour développer des applications Web Node.js qui s'exécutent sur le moteur de présentation Blink. [9]

#### 4.1.2 Git et GitHub :

**Git** est un logiciel libre créé par Linus Torvalds, auteur du noyau Linux, et distribué selon les termes de la licence publique générale GNU version 2. Git est un système de contrôle de version distribué permettant de suivre les modifications du code source pendant le développement du logiciel. Il est conçu pour coordonner le travail entre les programmeurs, mais il peut être utilisé pour suivre les modifications dans n'importe quel ensemble de fichiers. Ses objectifs incluent la vitesse, l'intégrité des données et la prise en charge des flux de travail distribués et non linéaires.[10]

**Github** est un site web qui fournit un service cloud aux développeurs pour stocker et gérer leur code qui offre toutes les fonctionnalités de contrôle de révision distribué et de gestion du code source (SCM) de Git, ainsi que l'ajout de ses propres fonctionnalités.

Il apporte aussi un suivi, ainsi qu'un contrôle des modifications qui lui sont apportées. Aujourd'hui, il représente l'un des moyens les plus populaires pour héberger des projets open source et partager du contenu. Récupération d'un code source sur le site web, le transférer sur une machine locale.[11]

### 4.1.3 WampServer :

**Wamp** Signifie "Windows, Apache, MySQL et PHP". WAMP est une variante de LAMP pour les systèmes Windows et est souvent installé en tant que logiciel bundle (Apache, MySQL et PHP).

**WampServer** est une plateforme de développement Web de type WAMP, permettant de faire fonctionner localement des scripts PHP. WampServer n'est pas en soi un logiciel, mais un environnement comprenant trois serveurs (Apache, MySQL et MariaDB), un interpréteur de script (PHP), ainsi que phpMyAdmin pour l'administration Web des bases MySQL.

Il dispose d'une interface d'administration permettant de gérer et d'administrer ses serveurs au travers d'un tray icon.[12]

### 4.1.4 Resilio Sync :

**Resilio Sync** est une solution de synchronisation et de partage de fichiers rapide, facile à utiliser et sécurisée qui fournit un stockage et des tailles de fichiers illimités. Resilio Sync utilise la technologie peer-to-peer qui améliore généralement les vitesses de transfert de fichiers de 2 à 10 fois. Il offre la possibilité de synchroniser les données de manière sécurisée et privée en répliquant les données directement entre des appareils de confiance et en gardant les données chiffrées en transit.[13]

## 4.2 Langage de programmations utilisés :

### 4.2.1 Html (HyperText Markup Language) :

**Le langage HTML** est le langage universel utilisé sur les pages web lisibles par tous les navigateurs web (Internet Explorer, Netscape, Mozilla, etc...). Ce langage fonctionne suivant l'assemblage et la combinaison de balises permettant de structurer et donner

l'apparence voulue aux données textes, images et multimédias suivant la mise en page voulue. [14]

### 4.2.2 CSS (Cascading Style Sheets) :

Css est un langage de mise en forme d'un document HTML. Il définit les règles de style et de disposition appliquées aux éléments d'un document html. On utilise le CSS pour modifier le style de n'importe quel élément html pour corriger ses dimensions, couleurs, bordures, etc. [15]

### 4.2.3 BOOTSTRAP :

**Bootstrap** est un Framework crée et mis en open source en 2012, il facilite la création d'application web. Il regroupe une collection d'outil fournis sous la forme de classes CSS et de bibliothèques JavaScript et jQuery, permettant ainsi de gagner du temps et de réaliser simplement des codes assez complexes tout en réduisant le poids du site web.[16]

### 4.2.4 NodeJS :

**NodeJs** est un environnement d'exécution Javascript piloté par des évènements. NodeJs est conçu pour créer applications réseau évolutives, à chaque connexion le rappel est déclenché, mais s'il n'y a aucun travail à faire NodeJs se mettra en Veil.[17]

### 4.2.5 EJS :

**Ejs** est un langage qui permet de générer une balise HTML avec JavaScript, pas de réinvention de l'itération ni même du flux de contrôle.

**Express JS** est une infrastructure d'application web Node Js minimaliste et flexible qui fournit un ensemble de fonctionnalités robuste pour les applications web et mobile, cette couche de fonctionnalités apportées par Express Js ne masque en aucun cas les fonctionnalités de Node Js.[18]

## Présentation des IHM :

Dans cette section, nous allons présenter quelques interfaces Homme-Machine de notre application.

### 4.2.6 L'interface Homme-Machine « Connexion » pour l'admin :

La connexion permet à un utilisateur d'accéder a son compte pour avoir les privilèges qu'ils lui sont approprié au tant que membre ou client si celui-là est déjà inscrit, ou lui permet de créer un compte dans le cas inverse.



FIGURE 4.1 – interface Homme-Machine « Connexion » pour l’admin



### 4.2.7 L'interface Homme-Machine « Gérer réservation » :

Cette interface permet à l'administrateur d'ajouter une nouvelle réservation pour les clients qui ne passent pas par le site, permet aussi de gérer les demandes de réservations que les clients soumettent via la plateforme afin de les valider ou de les refuser, il peut également consulter, modifier, afficher, imprimer des réservations déjà traités et il a la possibilité de faire une recherche pour faciliter la sélection d'une réservation déjà sauvegardée dans la base de données.



Identifiant	client	produit	date début date fin	date réservation	quantité
10	mouhoene narimane	chambres 2 personnes	2021-10-14 2021-07-11	2021-10-12	1
12	mouhoene narimane	chambre triples	2021-10-05 2021-10-13	2021-10-12	1
11	lydia	chambre triples	2021-10-21 2021-10-16	2021-10-12	1

FIGURE 4.2 – interface Homme-Machine « Gérer réservation»

#### 4.2.8 L'interface Homme-Machine « Modifier réservation » :

dans cette interface, l'admin pourra modifier les informations d'une réservation déjà faites, tel que la date de début, ...

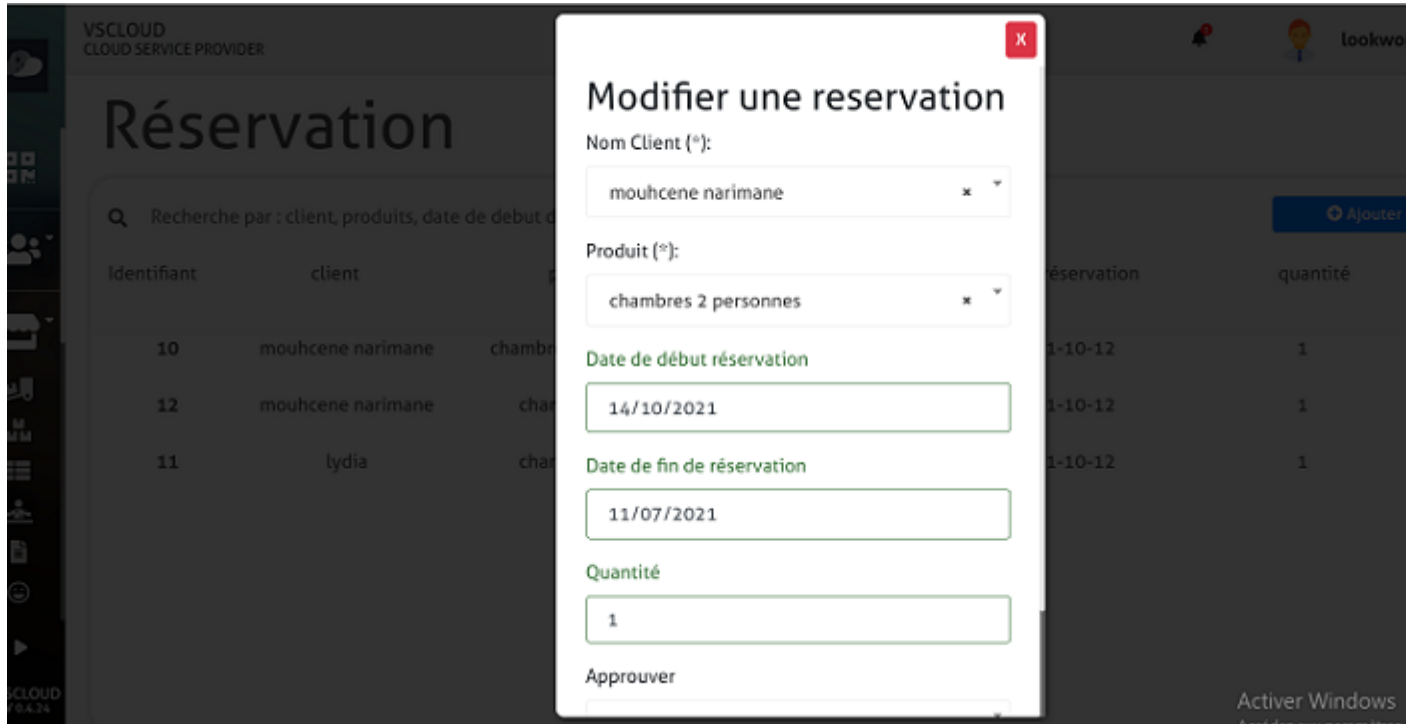


FIGURE 4.3 – interface Homme-Machine « Modifier réservation»

### 4.2.9 L'interface Homme-Machine «Ajouter réservation » :

L'administrateur d'ajouter une nouvelle réservation pour les clients qui ne passent pas par le site.

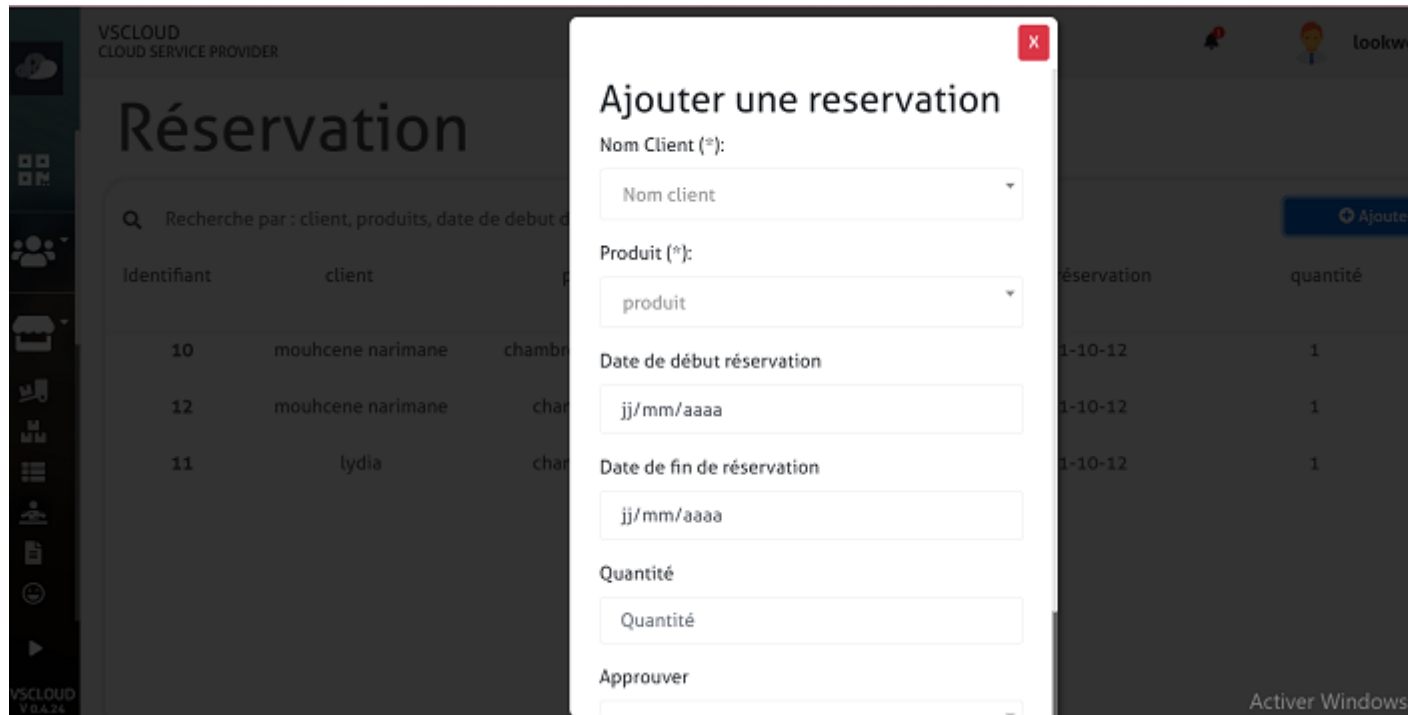


FIGURE 4.4 – interface Homme-Machine «Ajouter réservation »

#### 4.2.10 L'interface Homme-Machine «Inscription » Client :

localPost3000/checkout

Qui-Sommes-Nous? Produits Nos Équipements Contactez-Nous

Vous disposez d'un compte

Créer un compte

Livraison 0,00

Total 0,00

ENVOYER LA COMMANDE

User Password (\*)

Nom (\*) Prénom (\*)

Adresse e-mail (\*)

Commune

Adresse (\*)

Numéro de téléphone (\*)

Adresse de livraison (\*)

Type du compte

Je suis un client final

Je représente une entreprise ou une marque

ENVOYER LA COMMANDE

21°C Très ensoleillé 11:56 13/10/2021

FIGURE 4.5 – interface Homme-Machine « Inscription »

### 4.2.11 L'interface Homme-Machine « Réserver produits » :

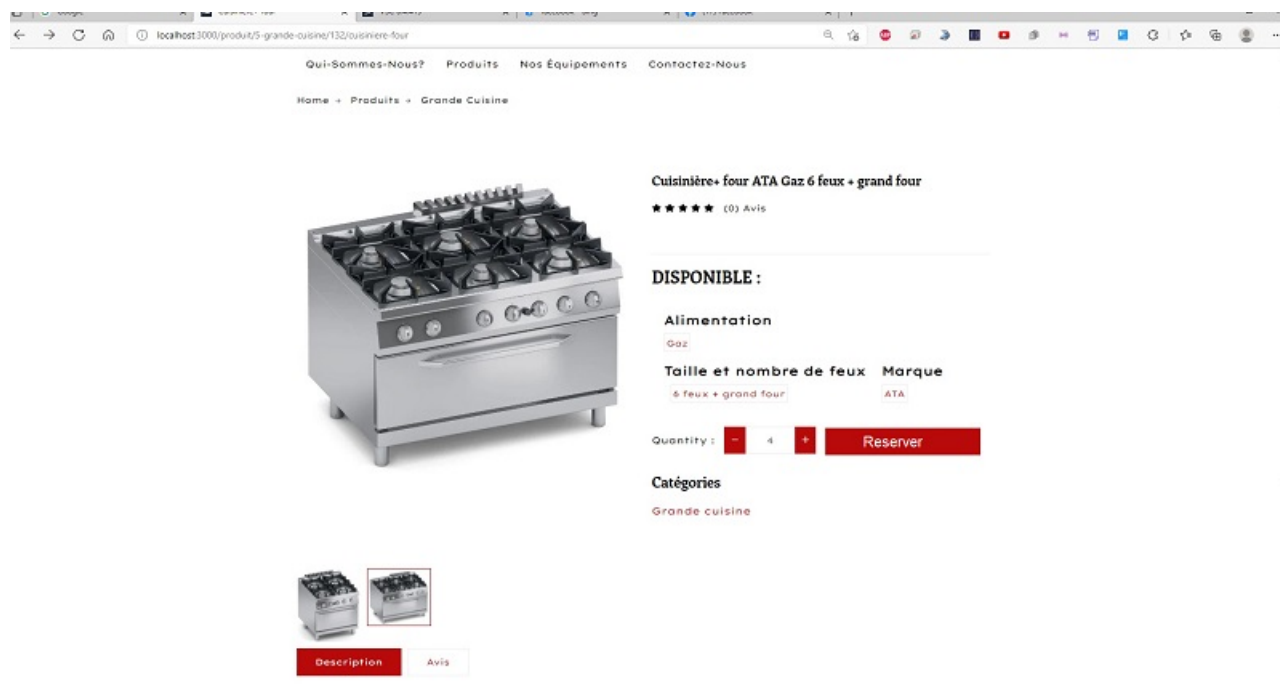


FIGURE 4.6 – interface Homme-Machine « Réserver produits »

### 4.2.12 L'interface Homme-Machine «Gestion profil » :

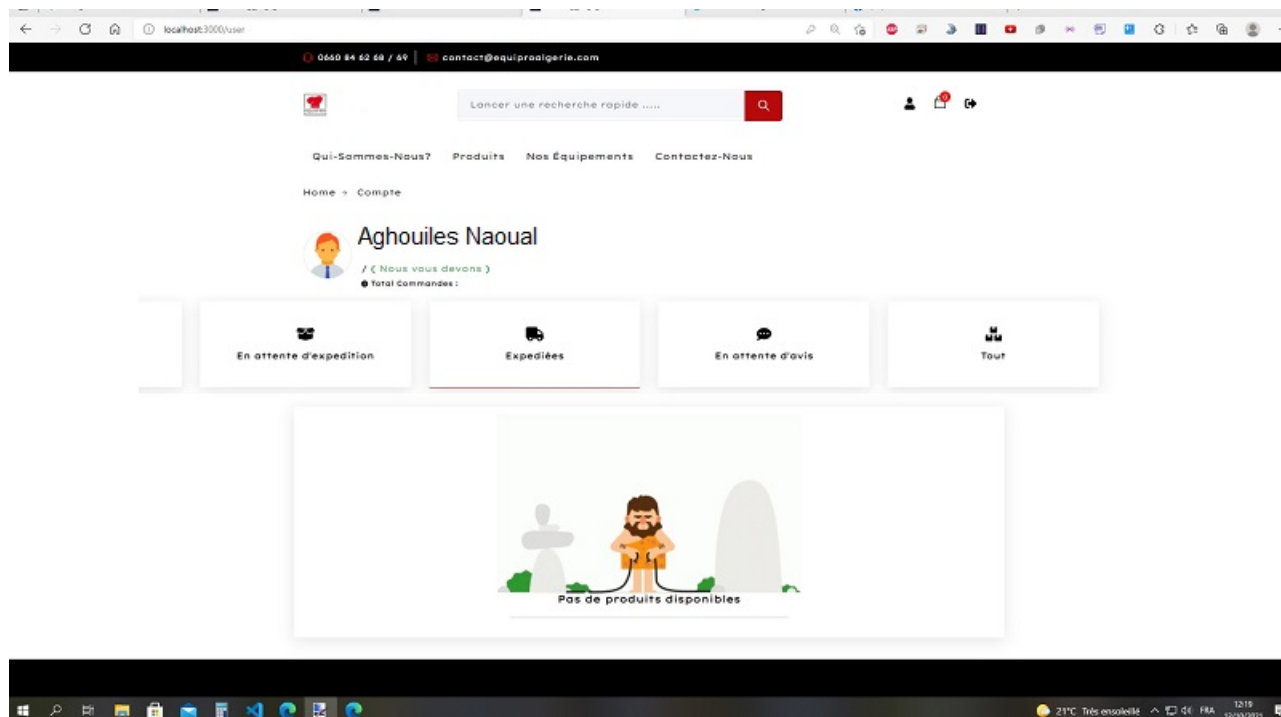


FIGURE 4.7 – interface Homme-Machine « Gestion profil »

## 4.2.13 L'interface Homme-Machine « Valider la réservation » :

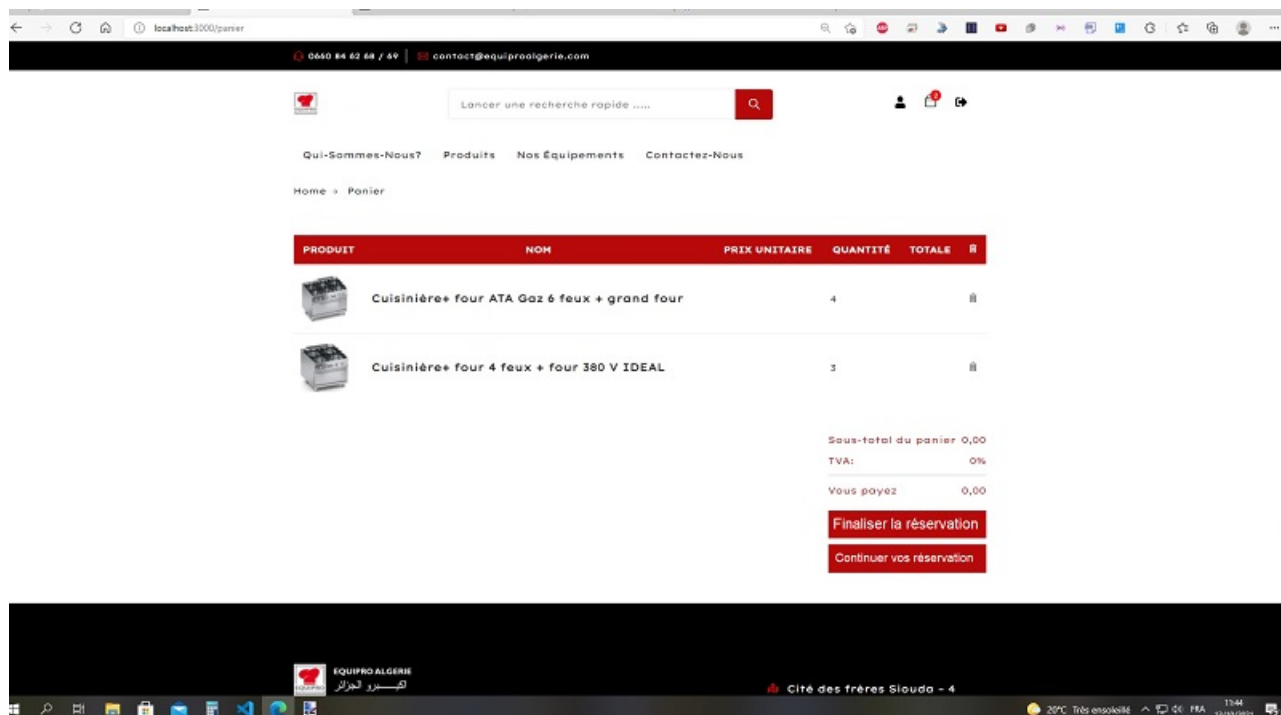


FIGURE 4.8 – interface Homme-Machine « Valider la réservation »

#### 4.2.14 L'interface Homme-Machine «Changer état» :

cette interface permet à l'admin de gérer les demandes de réservations que les clients soumettent via la plateforme afin de les valider ou de les refuser ou les mettre en attente.

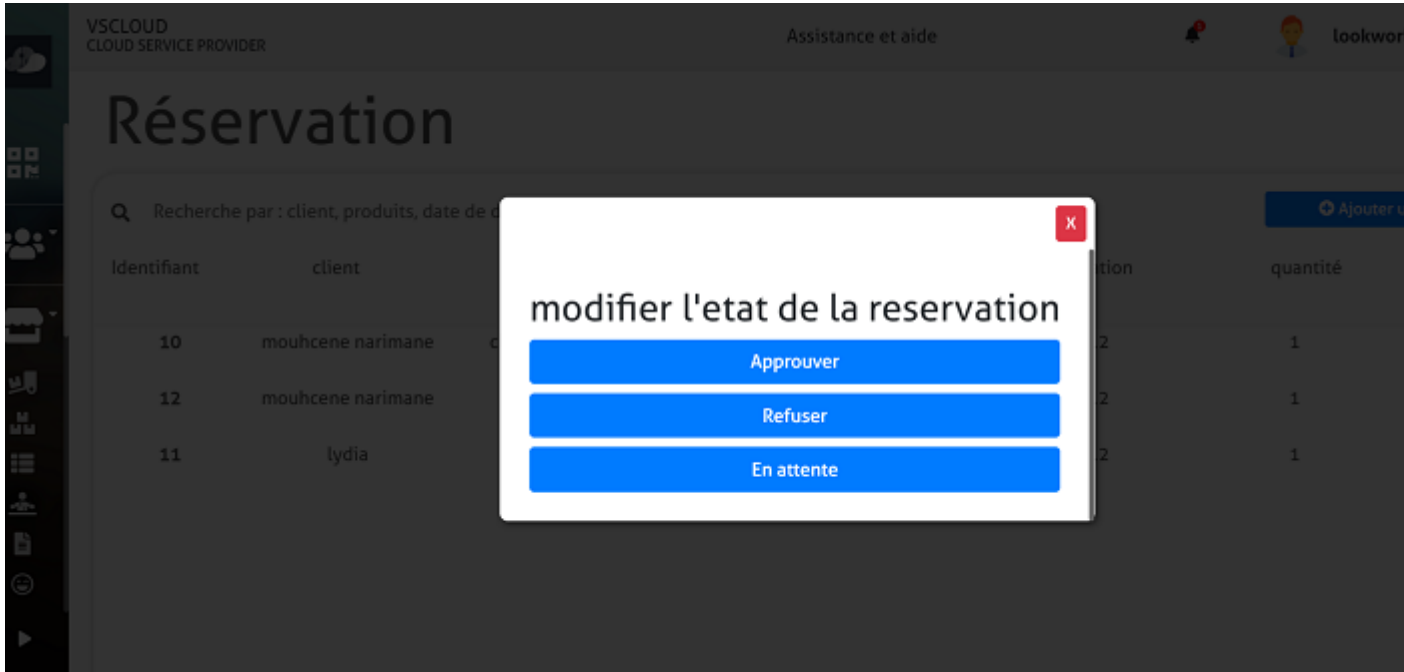


FIGURE 4.9 – interface Homme-Machine «Changer état»

## Conclusion :

Dans cet ultime chapitre, nous avons défini les outils techniques qui nous ont permis de réaliser notre application. Ensuite, nous avons présenté le diagramme de déploiement. Enfin, nous avons illustré quelques interfaces d'interaction homme-machine de notre application.



## CONCLUSION GÉNÉRALE

Le facteur ayant guidé notre choix sur ce thème relatif au développement de site web est que ce dernier est une pensée d'actualité, et que c'est un levier sur lequel se base beaucoup d'entreprises aujourd'hui pour avoir plus de visibilité et toucher directement leur cible au bon moment et à plus grande échelle. De plus, la présence en ligne permet d'augmenter le chiffre d'affaires.

Par ailleurs, cette thématique est une chance inouïe pour nous, tant par la possibilité d'en apprendre plus sur le développement de site, mais aussi par la possibilité de donner l'occasion à beaucoup de personnes la chance de créer leur site elles même selon leur besoin et leur goût.

Après une étude de l'existant, nous constatons le manque de quelques fonctionnalités notamment réservation. Nous avons proposé de développer cette fonctionnalité nécessaire pour fournir un service client complet. à savoir lui permettre de réserver par exemple dans le cas d'une entreprise hôtelière. Nous avons également revu des fonctionnalités déjà existantes pour pouvoir assurer un fonctionnement cohérent de toute la plateforme Vscloud, comme l'ajout des notifications au volet réservation. Mais encore, l'ajout des variantes au volet produit. Nous avons également modifier l'espace mon compte dans le but de permettre au client l'accès et la modification de ses informations.

Les fonctionnalités développées sont conçues selon la démarche de développement UP et avec UML pour une meilleure appréciation du comportement du système. Par la suite, nous avons utilisé HTML, Node JS, CSS, EJS, Bootstrap comme langages de programmation et Visual Studio Code comme outil de développement. Nous avons aussi eu recours à Réslío pour la synchronisation du code avec les membres de l'équipe. Wampserver,

MySQL, PHPMyAdmin pour la gestion de la base de données.

Les fonctionnalités développées répondent en tout point conformément au cahier des charges et aux attentes et exigences du donneur d'ordre.

## BIBLIOGRAPHIE

- [1] <https://apcpedagogie.com/cours-umllangage-de-modelisation-objet-unifie/>
- [2] <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-uml/>
- [3] <https://www.microsoft.com/fr-ca/microsoft-365/business-insights-ideas/resources/guide-to-uml-diagramming-and-database-modeling> .
- [4] [http://www.sii-tannarelli.com/contenus\\_pale/diagrammes\\_sysml\\_wb\\_gena\\_uroraW/co/sysml\\_contexte](http://www.sii-tannarelli.com/contenus_pale/diagrammes_sysml_wb_gena_uroraW/co/sysml_contexte).
- [5] <https://www.commentcamarche.net/contents/1138-uml-cas-d-utilisation-use-cases> .
- [6] <https://openclassrooms.com/forum/sujet/cas-dutilisation-et-description-textuelle> .
- [7] <http://remy-manu.no-ip.biz/UML/Cours/coursUML5>.
- [8] <https://www.lucidchart.com/pages/fr/diagramme-dinteraction-uml>
- [9] <https://fr.wikipedia.org/wiki/VisualStudioCode>
- [10] [https://edutechwiki.unige.ch/fr/Git\\_et\\_Github](https://edutechwiki.unige.ch/fr/Git_et_Github)
- [11] <https://www.lebigdata.fr/Github> : tout savoir sur cette plateforme d'hébergement de code (lebigdata.fr)
- [12] <https://fr.wikipedia.org/wiki/WampServer>
- [13] <https://ls.dm.aws.gartner.com/software/173194/resilio-sync>
- [14] <https://www.comprendre-internet.com/Qu-est-ce-que-le-HTML.html>
- [15] <http://webtuto.weebly.com/cest-quoi-le-css.html>
- [16] <https://www.hostinger.fr/tutoriels/cest-quoi-bootstrap/>

- [17] <https://www.pappleweb.com/index/definition-de-nodejs/>
- [18] <https://fre.skyrunninganz.com/781396-what-is-ejs-what-is-ITBSUI>

# *Résumé*

Aujourd'hui, L'informatique fait partie intégrante de la vie courante dans tous les domaines et prend de plus en plus de place dans le plus simple quotidien. Elle a ouvert les voies à un large éventail de méthodes de communication, permettant à la fois de rapprocher les distances, réduire les délais et minimiser les coûts.

Notre travail consiste à concevoir et réaliser un service de réservation en extension au service web VsCloud et apporter les modifications nécessaires pour y parvenir. Pour ce faire, nous avons suivi la démarche de développement logiciel UP et le langage de modélisation UML. La mise en œuvre de notre application a été effectuée sous l'environnement de développement Visual Studio Code VSC, à l'aide des langages de programmations suivants : Nodejs, HTML, CSS, EJS et de Framework Bootstrap . Nous avons utilisé MYSQL comme serveur de base de données.

**Mot clé :** L'informatique, service web, VsCloud, UP, UML, VSC, Nodejs, HTML, CSS, EJS, Bootstrap, MYSQL.

# *Abstract*

Today, IT is an integral part of everyday life in all fields and takes more and more place in the simplest daily life. It has opened the doors to a wide range of communication methods, simultaneously bringing distances closer together, reducing delays and minimizing costs.

Our job is to design and build a reservation service as an extension to the VsCloud web service and make the necessary changes to achieve it. To do this, we followed the UP software development process and the UML modeling language. The implementation of our application was done under the Visual Studio Code VSC development environment, using the following programming languages : Nodejs, HTML, CSS, EJS using Bootstrap Framework. We used MYSQL as the database server.

**Keywords :** Computing, web service, VsCloud, UP, UML, VSC, Nodejs, HTML, CSS, EJS, Bootstrap, MYSQL.