



République Tunisienne
Ministère de l'Enseignement Supérieur
Et de la Recherche Scientifique
Université de Carthage
Institut Supérieur des Technologies de
L'Information et de la Communication



Projet : Réservation salles de réunion

Rapport de Spécification des besoins

Réalisé Par

Mohamed Amir Jlidi & Mohamed Iheb Ounalli

LGLSI 2A

1- Introduction

2- Spécifications des besoins fonctionnels

2.1 La liste des acteurs

2.2 Les besoins fonctionnels

3- Spécifications des besoins non fonctionnels

3.1 Caractéristiques d'une PWA

3.2 Les attentes non fonctionnels

4- Conception UML

4.1 Diagrammes des cas d'utilisation

4.2 Diagrammes de séquences

5- Conclusion

1 - Introduction :

Le “ coworking space ” ou l'espace de travail partagé est un concept en plein développement. Autrement dit, c'est un nouveau mode d'organisation du travail basé sur le partage, la synergie et les échanges. Ce concept est apparu suite à l'émergence de l'économie collaborative. D'un point de vue juridique, un coworking space se base sur la location d'espace de travail partagés. En outre, ces espaces s'adressent principalement aux entrepreneurs et leurs collaborateurs indépendants, étudiants, chercheurs ...

Travailler au sein d'un espace de travail partagé permet de procurer plusieurs avantages :

- **Minimiser les coûts : Location d'un bureau classique, électricité, internet ...**
- **Travailler sereinement et tranquillement**
- **Développer le réseau**
- **Décrocher de nouvelles opportunités**

En outre, ces espaces proposent également des salles de réunions, un espace de détente, des rétroprojecteurs, une cuisine, un accès internet ...

En Tunisie, et au cours de la dernière décennie, les espaces de travail partagés sont multipliés notamment : **Cogite Coworking Space, Work Zone Coworking Space, Hive 12, Startup Huas** ... Parmi les problèmes rencontrés par les propriétaires des coworking spaces, c'est la gestion des salles de réunion. On se trouve fréquemment dans des situations où on peut trouver des réservations simultanées ou des personnes qui dépassent la durée de réservation.

C'est pourquoi, durant ce projet, on va proposer une plateforme web permettant aux propriétaires des espaces de travail partagés de mieux gérer les salles de réunion.

2 - Spécifications des besoins fonctionnels :

Nous allons identifier, dans cette partie, les besoins de notre projet en mettant l'accent sur ses acteurs, ses besoins fonctionnels ainsi que ses besoins non-fonctionnels pour assurer un développement satisfaisant.

2.1 La liste des acteurs :

Un acteur représente l'abstraction d'un rôle joué par des entités externes (utilisateur, dispositif matériel ou autre système) qui interagissent directement avec le système étudié. L'acteur représenté dans cette application peut être soit un admin soit un client.

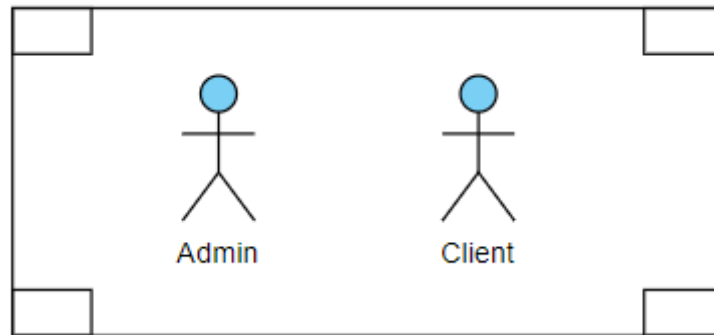


Figure 1: Les acteurs

Acteurs	Rôle
Admin	<ul style="list-style-type: none">- S'authentifier- Créer compte- Accéder à l'historique- Gérer compte-Gérer Salle Réservation- Enregistrer demander de réservation

Client	<ul style="list-style-type: none"> - S'authentifier - Créer compte - Accéder à l'historique - Enregistrer demande de réservation
--------	--

2.2 Les besoins fonctionnels :

- **S'authentifier**
- **Créer compte**
- **Accéder à l'historique**
- **Gérer compte**
 - Consulter Compte
 - Ajouter Compte
 - Supprimer Compte
 - Modifier Compte
- **Gérer Salle Réservation**
 - Consulter Salle Réservation
 - Ajouter Salle Réservation
 - Supprimer Salle Réservation
 - Modifier Salle Réservation
- **Enregistrer demande de réservation**
 - Vérifier la disponibilité de la salle

3 - Spécifications des besoins non fonctionnels :

Nous allons identifier, dans cette partie, les besoins de notre projet en mettant l'accent sur ses besoins fonctionnels ainsi que ses besoins non-fonctionnels pour assurer un développement satisfaisant.

3.1 Caractéristiques d'une PWA :

Quel que soit le « device » (mobile, ordinateur, tablette...), une application web est identifiée comme PWA dès lors qu'elle répond à un ensemble de caractéristiques, dont les principales sont les suivantes :

- **Progressive**: fonctionne pour tous les utilisateurs, quel que soit le navigateur et le « device ».
- **Responsive**: s'affiche correctement et s'ajuste automatiquement à la taille de l'écran du terminal utilisé.
- Avoir une **expérience proche d'une application native** (mobile ou desktop) : une seule et même application développée ;
- **Sécurisée**: la connexion doit se faire en HTTPS.
- Facilement **installable** sur le « device » : raccourci sur le bureau pour un PC et sur l'écran d'accueil sur un smartphone.
- Fonctionne quelle que soit la **connectivité**: bas débit ou perte de connexion (offline).
- **Performante**: l'application doit être rapide à démarrer et fluide ;

- On peut **partager** un lien vers l'application.

3.2 Les attentes non fonctionnels :

Décrivent les objectifs liés aux performances du système et aux contraintes de son environnement. Ses exigences techniques sont souvent exprimées sous forme d'objectifs spécifiques que doit atteindre le système.

Les besoins non-fonctionnels de ce projet sont les suivants :

- Qualité
- Accessibilité et performance
- Fiabilité
- Sécurité
- L'ergonomie des interfaces
- Assurer une cohérence et une crédibilité aux informations stockées dans la base de données
- L'application doit fonctionner en réseau

4 – Conception UML :

4.1 Diagrammes des cas d'utilisation :

Chaque usage que les acteurs font du système est représenté par un cas d'utilisation. Chaque cas d'utilisation représente une fonctionnalité qui lui est offerte afin de produire le résultat attendu. Ainsi ce diagramme décrit l'interaction entre le système et l'acteur en déterminant les besoins, l'utilisateur et tout ce que doit faire le système pour l'utilisateur.

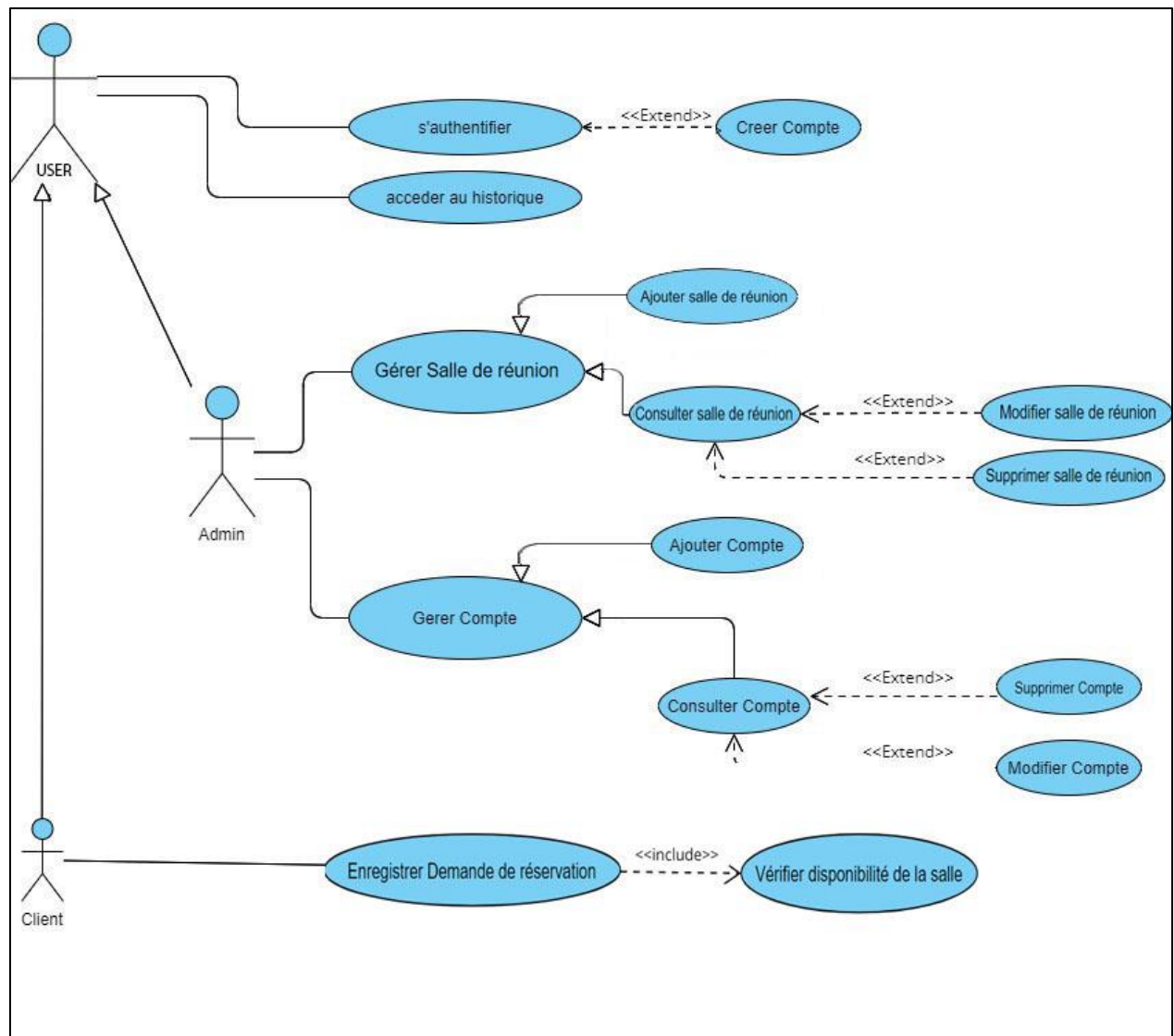


Figure 1 Conception de cas d'utilisation :

Raffinement du cas d'utilisation <S'authentifier>

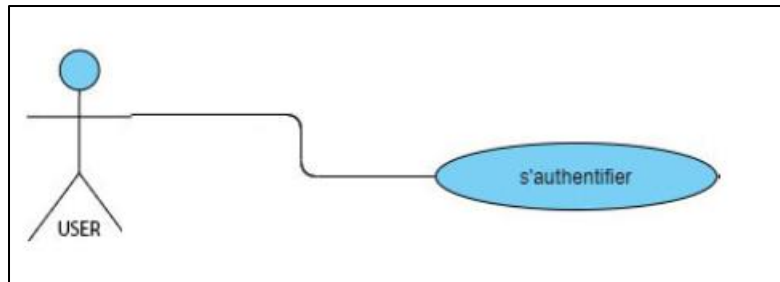


Figure 2 Raffinement du cas d'utilisation <S'authentifier>

Cas d'utilisation	S'authentifier
<i>Acteurs</i>	Client , Admin
<i>Pré- condition</i>	<ul style="list-style-type: none">- Système en marche- Connexion établie- Compte client existant
<i>Post-condition</i>	User connecté
<i>Description des Scenarios principaux</i>	<ol style="list-style-type: none">1. Le système affiche l'interface s'authentifier2. L'acteur saisie les coordonnées3. L'acteur appuie sur le bouton « se connecter »4. Le système valide5. Le système affiche une message de validation
<i>Exception</i>	Le système affiche un message d'erreur si les données sont erronées ou le compte n'existe pas.
<i>Extension</i>	Créer compte

Raffinement du cas d'utilisation <Créer compte>

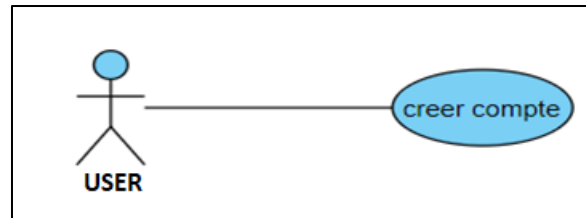


Figure 3 Raffinement du cas d'utilisation <Créer compte>

Cas d'utilisation	Créer compte
<i>Acteurs</i>	Client , admin
<i>Pré- condition</i>	- Système en marche - Connexion établie
<i>Post-condition</i>	Compte crée
<i>Description des Scenarios principaux</i>	<ol style="list-style-type: none">1. Le système affiche l'interface d'inscription2. User remplit le formulaire3. Le User appuie sur le bouton «s'inscrire»4. Le système valide5. Le système affiche un message de validation
<i>Exception</i>	Le système affiche un message d'erreur si les données sont erronées ou le compte existe déjà.

Raffinement du cas d'utilisation <accéder au historique>

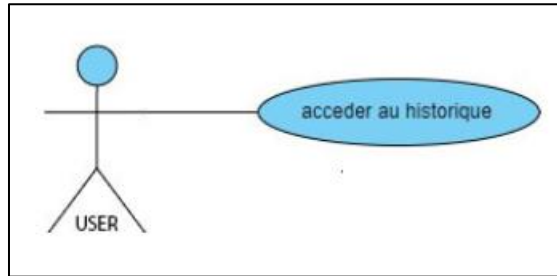


Figure 4 Raffinement du cas d'utilisation <accéder au historique>

Cas d'utilisation	Creer compte
Acteurs	USER
Pré- condition	<ul style="list-style-type: none">- Système en marche- Connexion établie- Compte déjà crée
Post-condition	Acceder au historique de ces reservations.
Description des Scenarios principaux	1. La systeme afficher l 'interface de l'historique
Exception	Connexion a échoué.

Raffinement du cas d'utilisation <gérer réservation>

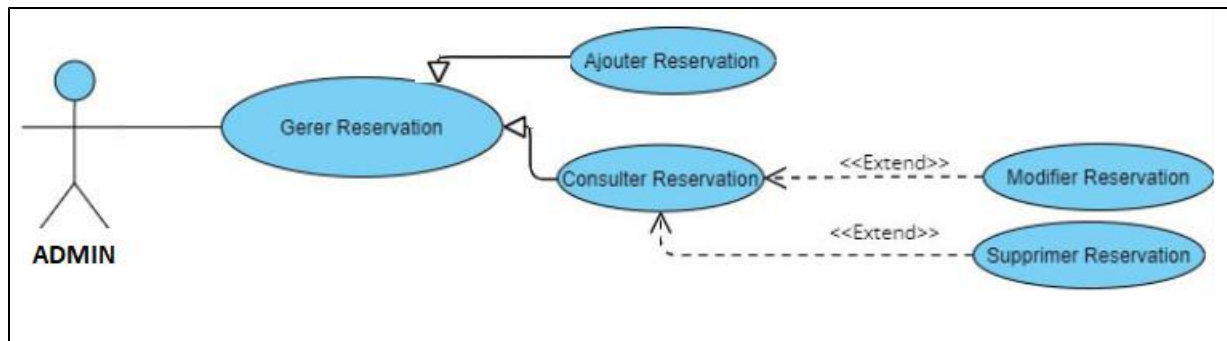


Figure 5 Raffinement du cas d'utilisation <gérer reservation>

Cas d'utilisation	Gerer salle R
Acteurs	Admin
Pré- condition	<ul style="list-style-type: none"> - Système en marche - Connexion établie - Admin connecté
Post-condition	Salle R gérée
Description des Scenarios principaux	<ol style="list-style-type: none"> 1. Le système affiche l'interface de reservation des salles desréunions 2. L' Admin choisit l'opération de gestion 3. Le système affiche l'interface approprié
Scénarii	Ajouter salle R Modifier salle R ...

Raffinement du cas d'utilisation <gérer compte>

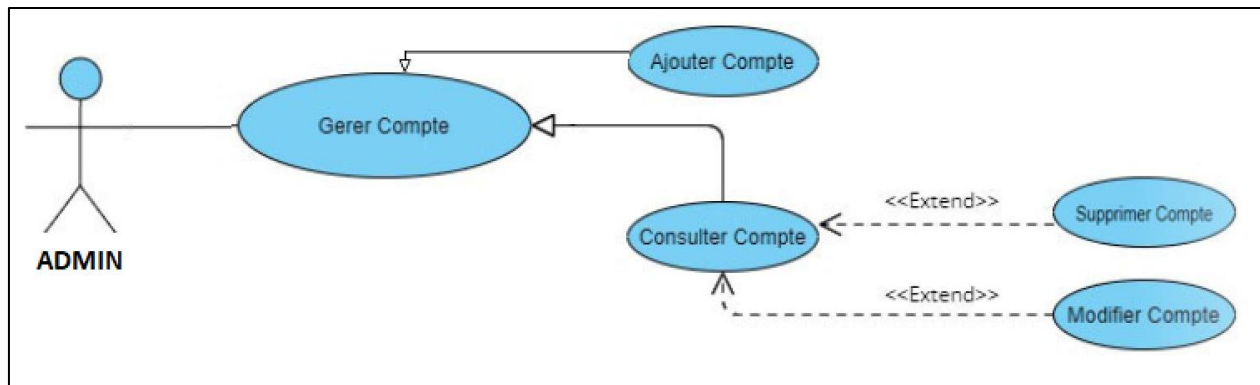


Figure 6 Raffinement du cas d'utilisation <gérer compte>

Cas d'utilisation	Gerer compte
Acteurs	Admin
Pré- condition	-Système en marche -Connexion établie - Client connecté
Post-condition	
Description des Scenarios principaux	1.Le système affiche l'interface compte avec boutons « ajouter » « supprimer » et « modifier » 2.Le Admin appuie sur « supprimer » 3.Le système efface le compte 4.Le Admin appuie sur « modifier » 5.Le système affiche le formulaire d'inscription 6.Le Admin fait les changements nécessaires 7.Le Admin appuie sur « valider » 8.Le Admin appuie sur « ajouter » 9.Le système affiche le formulaire d'inscription 10Le Admin remplit le formulaire 11.Le Admin appuie sur « enregistrer » 12.Le système ajoute une nouvelle compte
Exception	

Raffinement du cas d'utilisation <Enregistrer demande réservation>

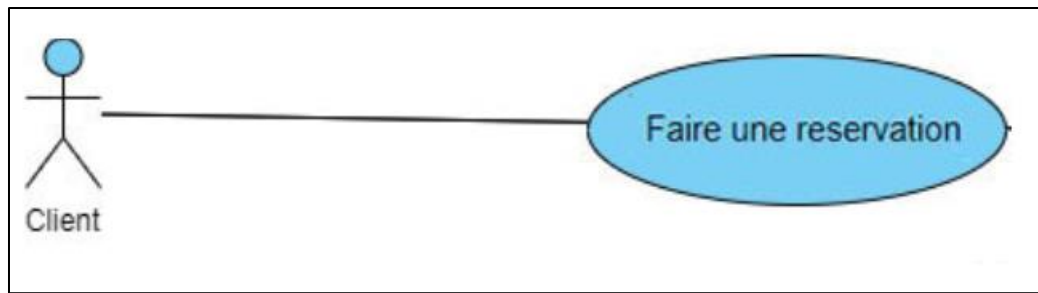


Figure 7 Raffinement du cas d'utilisation <Faire une réservation>

<i>Cas d'utilisation</i>	Faire une reservation
<i>Acteurs</i>	Client
<i>Pré- condition</i>	-Système en marche -Connexion établie - Client connecté
<i>Post-condition</i>	Une demande de réservation est enregistrée
<i>Description des Scenarios principaux</i>	
<i>Exception</i>	

Raffinement du cas d'utilisation <Consulter Salle>

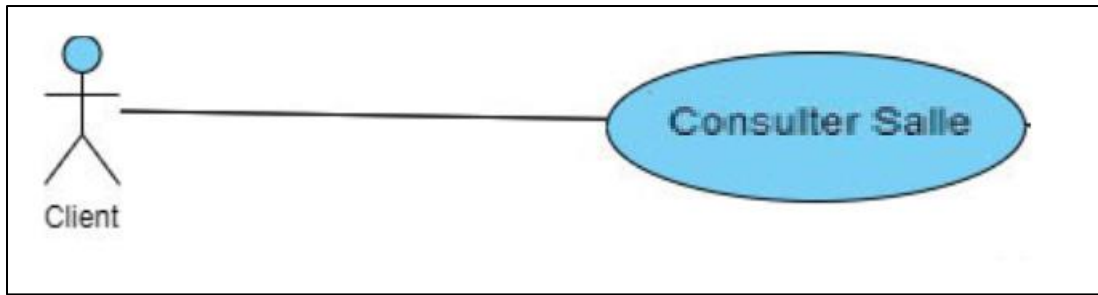


Figure 8 Raffinement du cas d'utilisation <Consulter Salle>

<i>Cas d'utilisation</i>	Consulter Salle
<i>Acteurs</i>	Client
<i>Pré- condition</i>	-Système en marche -Connexion établie - Client connecté
<i>Post-condition</i>	Affichages des salles disponibles
<i>Description des Scenarios principaux</i>	
<i>Exception</i>	

4.2 Diagrammes de séquences :

Diagramme de séquence authentification :

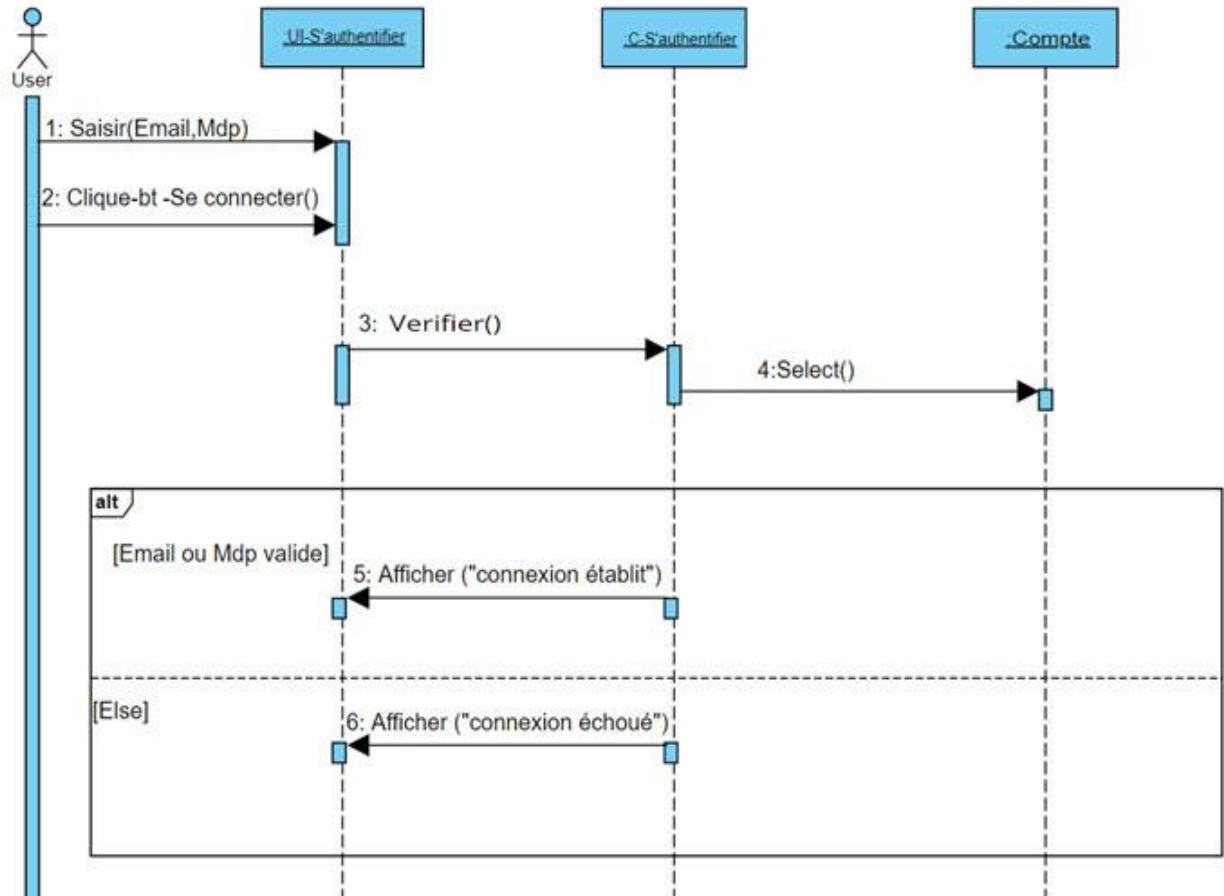


Diagramme de séquence accéder à l'historique

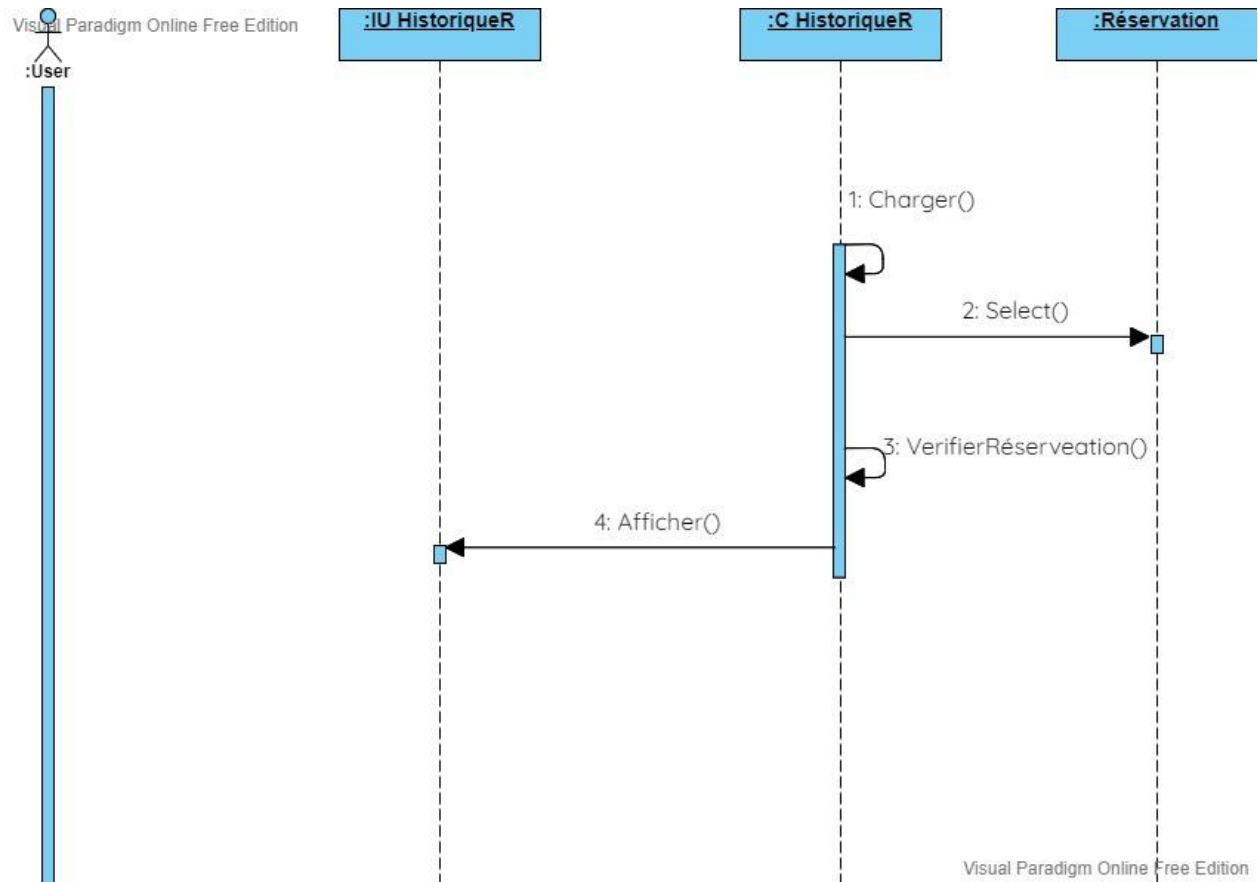


Diagramme de séquence Consulter Salle

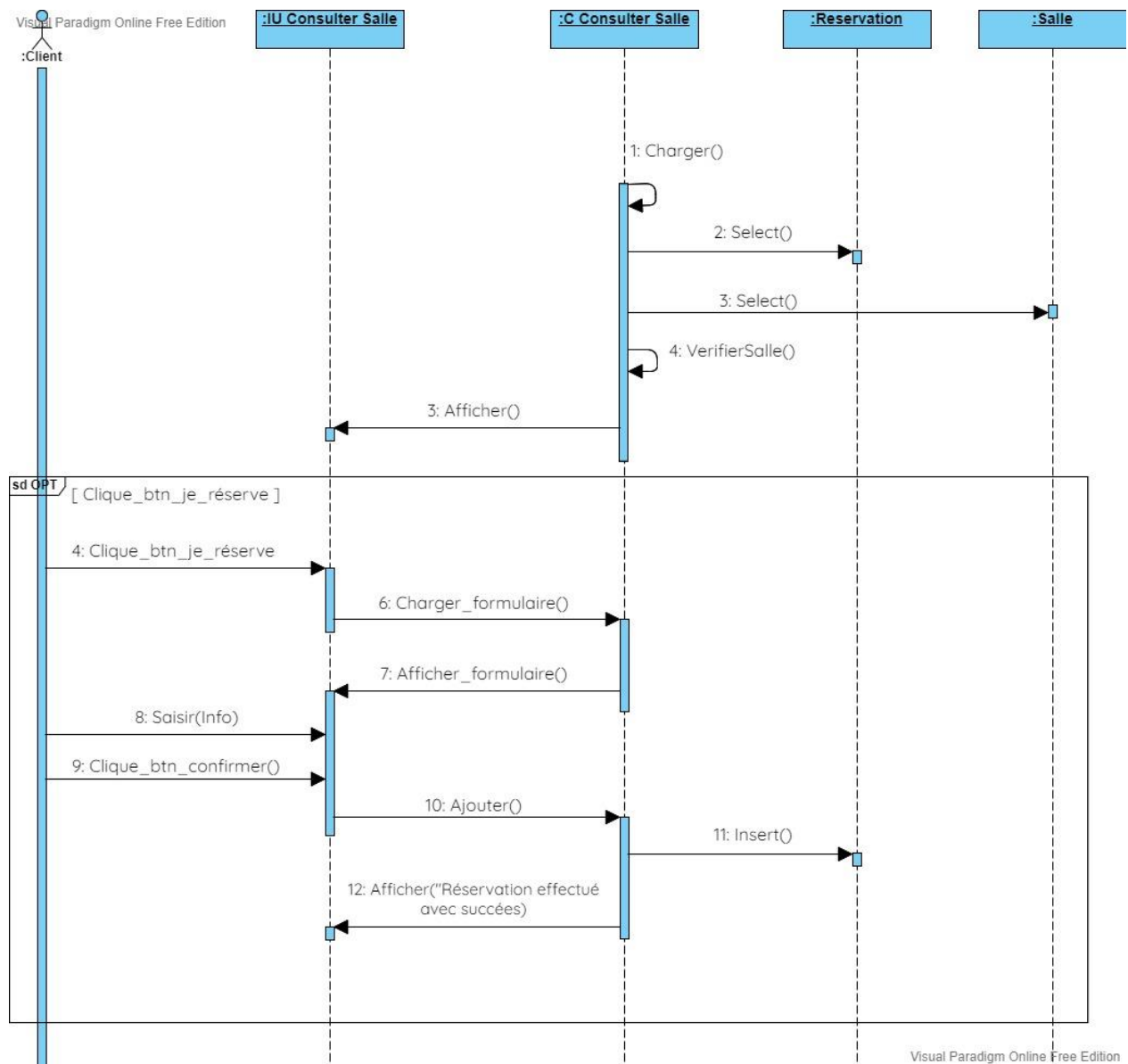


Diagramme de séquence Créer Compte

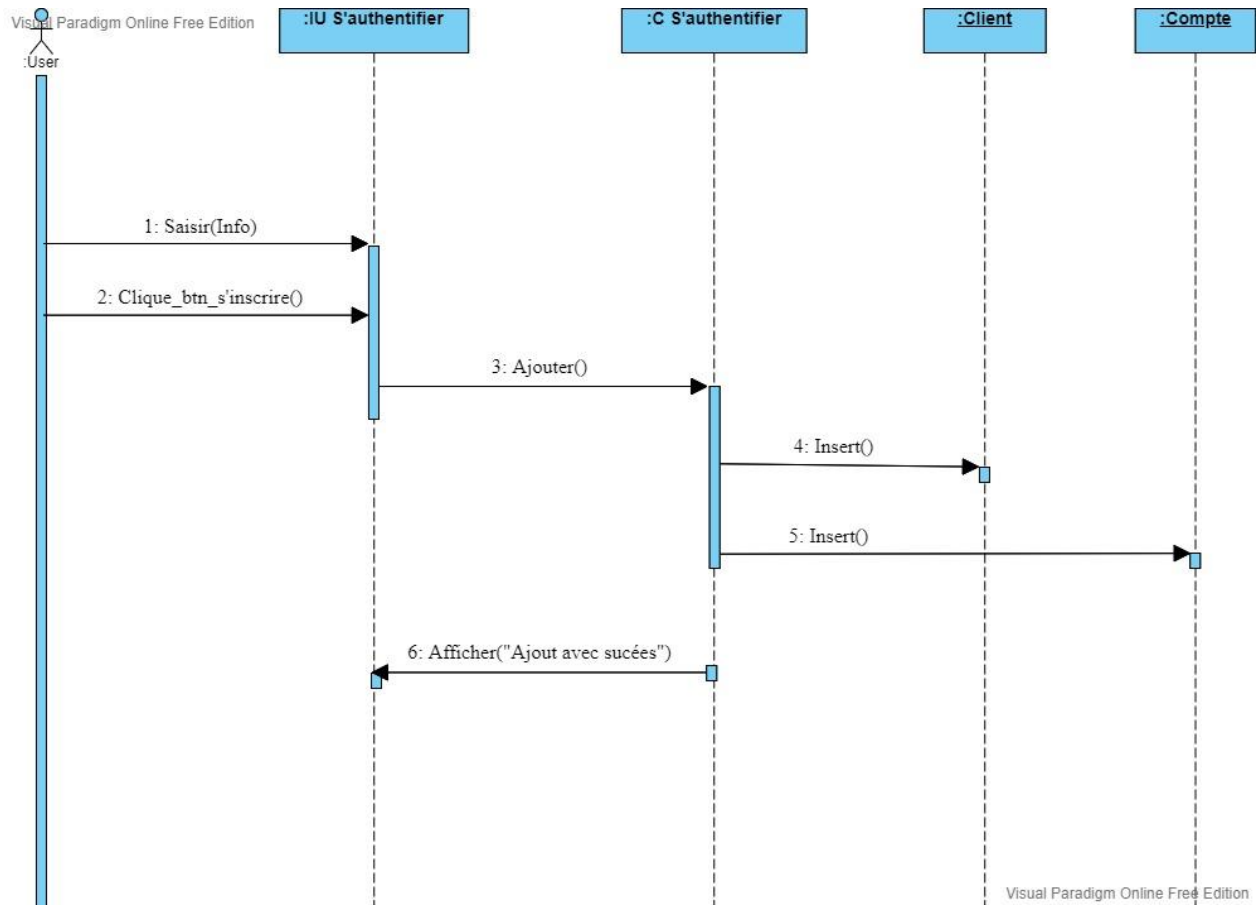


Diagramme de séquence Gérer Compte

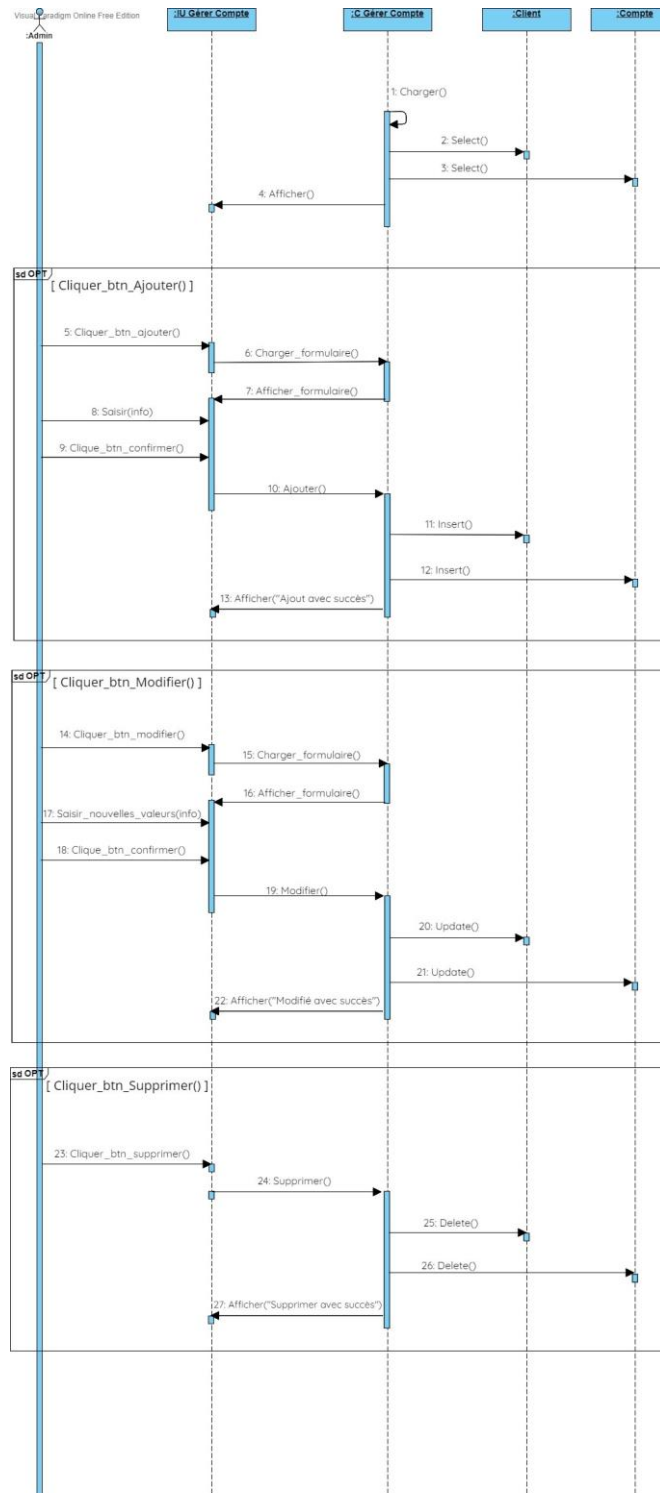
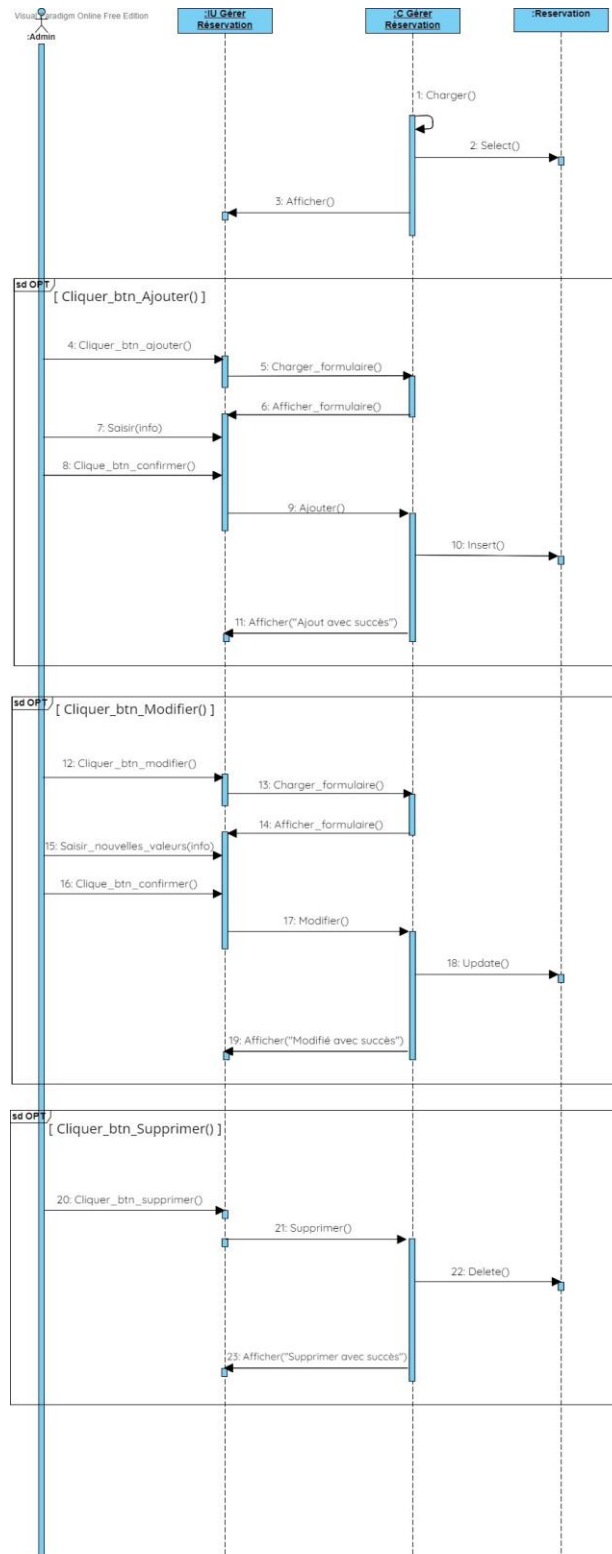


Diagramme de séquence Gérer Réservation



5)- Conclusion :

Afin de bien conduire notre projet et de nous assurer le bon déroulement des différentes phases nous avons utilisé

HTML



HTML désigne un type de langage informatique descriptif. Il s'agit plus précisément d'un format de données utilisé dans l'univers d'internet pour la mise en forme des pages web. [9]

CSS



CSS correspond à un langage informatique permettant de mettre en forme des pages web (HTML ou XML). Ce langage est donc composé des fameuses « feuilles de style en cascade » également appelées fichiers CSS et contient des éléments de codage. [10]



Bootstrap

Bootstrap est une collection d'outils utiles à la création du design de sites et d'applications web. C'est un ensemble qui contient des codes HTML et CSS, des formulaires, boutons, outils de navigation et autres éléments interactifs, ainsi que des extensions JavaScript en option. [11]



Angular est un framework côté client, open source, basé sur TypeScript, et co-dirigé par l'équipe du projet « Angular »

à Google et par une communauté de particuliers et de sociétés.
Angular est une réécriture complet d'AngularJS, cadriciel construit
par la même équipe.