

151-Documentation Projet

Rapport personnel

Date de création : 09.02.2024
Version 1 du 09.02.2024

Ilhan Kaynak



Module du :
02.02.2024 au
jj.mm.aaa

Table des matières

1	INTRODUCTION	3
2	ANALYSE	3

2.1	Objectifs projet	3
2.1.1	Consignes du projet	3
2.1.2	L'organisation, le fonctionnement et la structure du projet	3
2.2	Présentation projet (TravelFinder)	3
2.2.1	Objectifs	3
2.2.2	Fonctionnalité	4
2.3	Use cases.....	4
2.4	Maquettes.....	4
2.4.1	Vue visiteur	4
2.4.2	Vue connexion	5
2.4.3	Vue utilisateur.....	6
2.4.4	Vue réservation.....	6
2.4.5	Vue mes réservations.....	7
2.5	Diagramme activités.....	8
2.5.1	Activités login	8
2.5.2	Activités réservation	8
2.6	Diagrammes séquence	9
2.6.1	Séquence login	9
2.6.2	Séquence réservation	11
2.7	Schéma ER.....	11

3	CONCEPTION	12
----------	-------------------------	-----------

3.1	Diagramme de classe.....	12
3.1.1	Client	12
3.1.2	Serveur	13
3.2	Schéma relationnel	14

3.3	Diagramme séquence interactions	14
3.4	Conception de tests.....	Erreur ! Signet non défini.

4	IMPLÉMENTATION.....	16
5	CONCLUSION.....	16

5.1	Ce que j'ai appris	16
5.2	Ce que j'ai aimé	16
5.3	Ce que j'ai moins aimé	16

1 Introduction

2 Analyse

2.1 Objectifs projet

2.1.1 Consignes du projet

1. Le projet se fait individuellement.
2. Le projet devra être publié sur serveur emf-informatique.
3. Le projet doit comporter du côté client au moins deux zones différentes (visiteur et utilisateur connecté).
4. L'application cliente HTML (JS-AJAX-JQuery) et celle serveur PHPo doivent être réalisées en Objet.
5. Les données seront enregistrées dans une base de données mariaDB.
6. La réalisation du projet doit être documentée. La documentation (exempte de fautes d'orthographe) devra contenir tous les éléments demandés dans le RP.
7. Il faut réaliser le diagramme d'activité + le diagramme de séquences système + le diagramme de séquence interactions + la descente de code pour le login
8. Il faut réaliser le diagramme d'activité + le diagramme de séquences système + le diagramme de séquence interactions + la descente de code pour une fonctionnalité complète (hors login)

2.1.2 L'organisation, le fonctionnement et la structure du projet

1. Côté client :
 - Il y a une séparation des fichiers images, html, js, css
 - Il ne doit y avoir de scripts js dans les fichiers html
 - Le projet fonctionne parfaitement avec tous les navigateurs
 - L'application cliente doit s'initier par un login-password
2. Côté serveur :
 - A part les fichiers appelés par le client, les autres PHP sont des objets
 - Une séparation doit être faite entre les contrôleurs et les worker
 - Le serveur doit être appelé avec les bons paramètres (GET, POST, DELETE, PUT) et doit retourner les bons codes de réussite ou d'erreur
 - Le serveur doit retourner des données au format XML ou JSON
 - Le mot de passe doit être enregistré hacher dans la base de données
 - L'information que l'utilisateur est bien logué doit être sauvée dans la session de l'utilisateur
 - Le projet doit être robuste face aux injections HTML, JS et SQL
3. La qualité du code :
 - Tous les fichiers JS devront respecter le standard de documentation de JSDoc (<http://usejsdoc.org/>)
 - Toutes les classes PHPo devront respecter le standard de documentation de PHPDoc (<http://www.phpdoc.org>).

2.2 Présentation projet (TravelFinder)

2.2.1 Objectifs

TravelFinder est une plateforme en ligne conçue pour simplifier la recherche et la réservation d'hôtels dans diverses destinations. Le site offre aux utilisateurs la possibilité de rechercher des hôtels disponibles dans un lieu spécifique de leur choix, puis de réserver leur séjour en ligne.

2.2.2 Fonctionnalité

TravelFinder vise à simplifier le processus de recherche et de réservation d'hôtels en offrant une plateforme conviviale et intuitive. Les principaux objectifs du projet incluent :

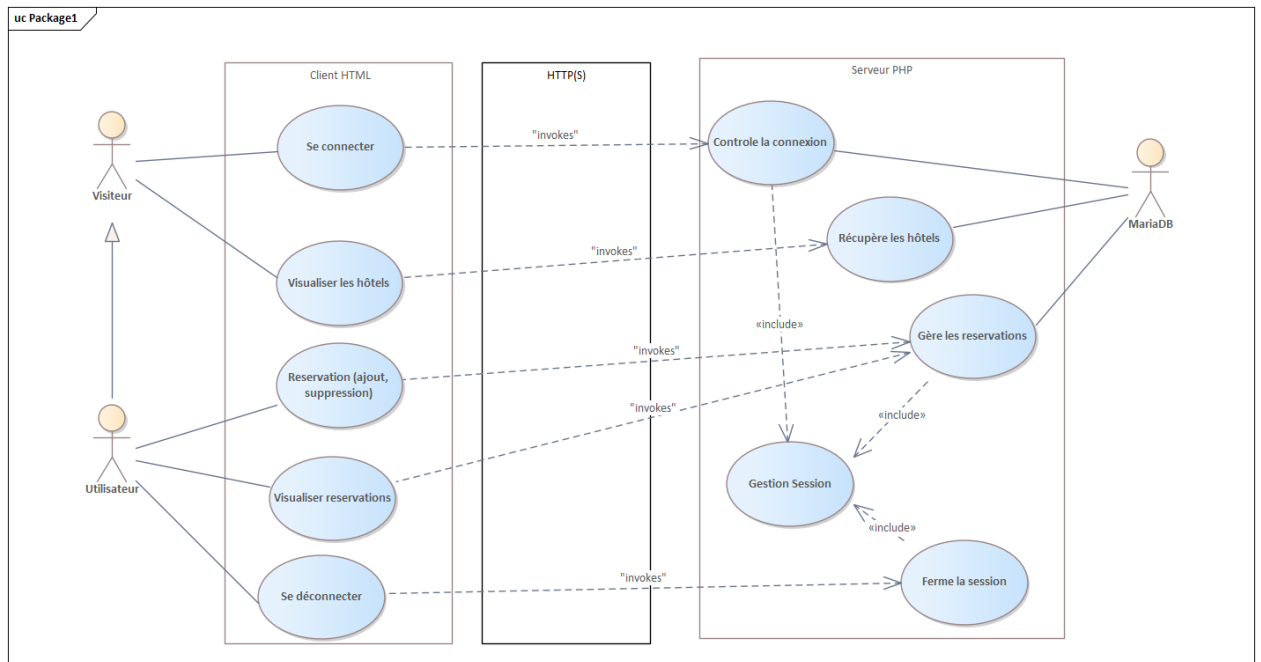
- Offrir une expérience utilisateur fluide et agréable lors de la recherche et de la réservation d'hôtels.
- Fournir des informations détaillées et précises sur chaque hôtel répertorié pour aider les utilisateurs à prendre des décisions informées.

2.2.3 Qui peut faire quoi

Le visiteur : il peut visualiser les hotels mais ne peut pas réserver, il peut se connecter.

L'utilisateur : il peut visualiser et réserver les hotels.

2.3 Use cases



2.4 Maquettes

2.4.1 Vue visiteur

TravelFinder	Connexion
Calendrier dynamique	Détails Réservation (Lieu + nombre personne)
Hôtel disponible (sans lien clicable) (Nom hôtel + image extérieur de l'hôtel)	

2.4.2 Vue connexion

Connectez vous !

Nom d'utilisateur

Mot de passe

valider

Créer un utilisateur

Mot de passe oublier ?

2.4.3 Vue inscription

Enregistrer vous !

Nom d'utilisateur

Mot de passe

valider

2.4.4 Vue utilisateur

TravelFinder		Mes reservations	Deconnexion
Calendrier dynamique	Détails Réservation (Lieu + nombre personne)		
Hôtel disponible (avec lien cliable) (Nom hôtel + image extérieur de l'hôtel)			

2.4.5 Vue réservation

Nom Hôtel

Image hôtel

Details Hôtel (Lieu + Nombre de places disponible)

Reserver

Retour liste
d'hôtel

2.4.6 Vue mes réservations

TravelFinder

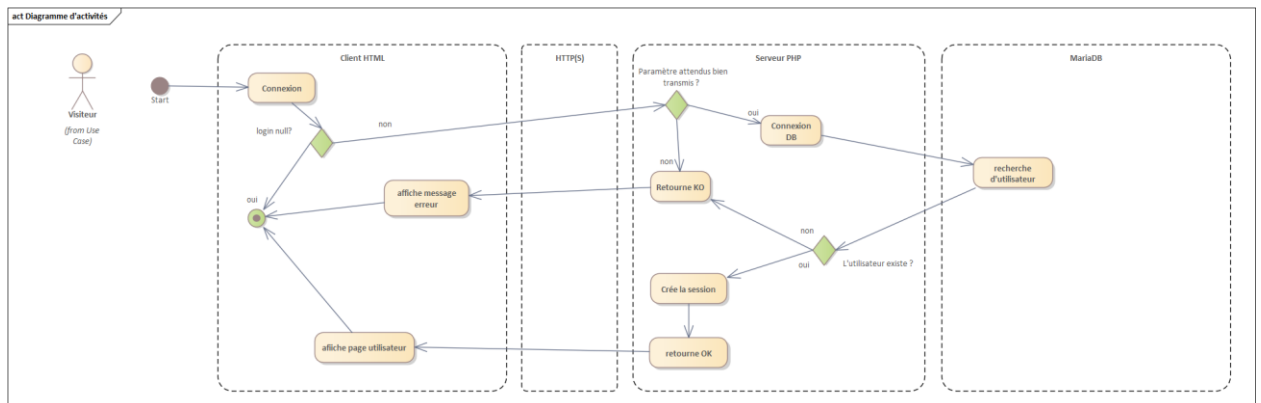
Mes reservations

Retour liste hotel

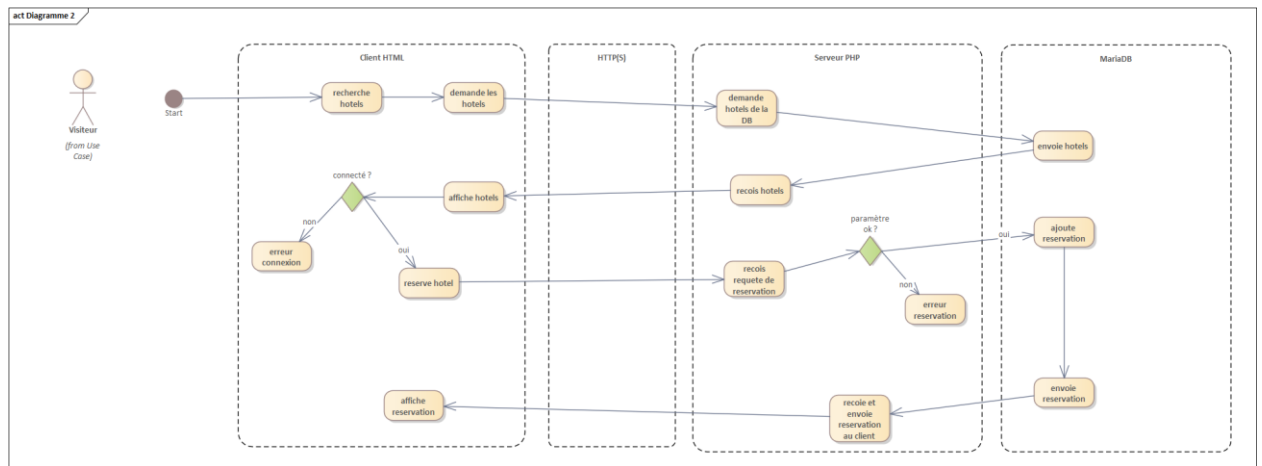
Supprimer
reservation

2.5 Diagramme activités

2.5.1 Activités login

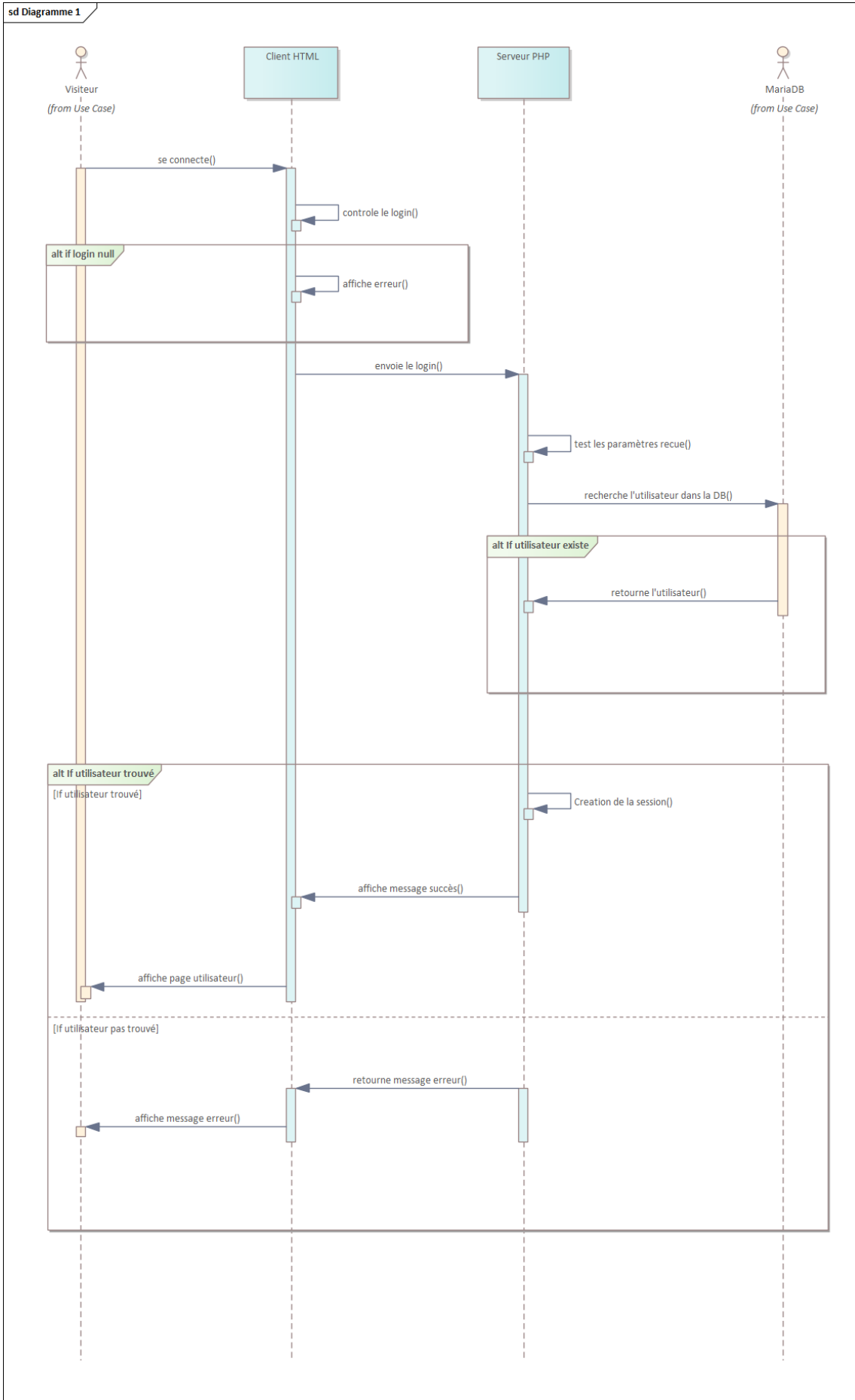


2.5.2 Activités réservation

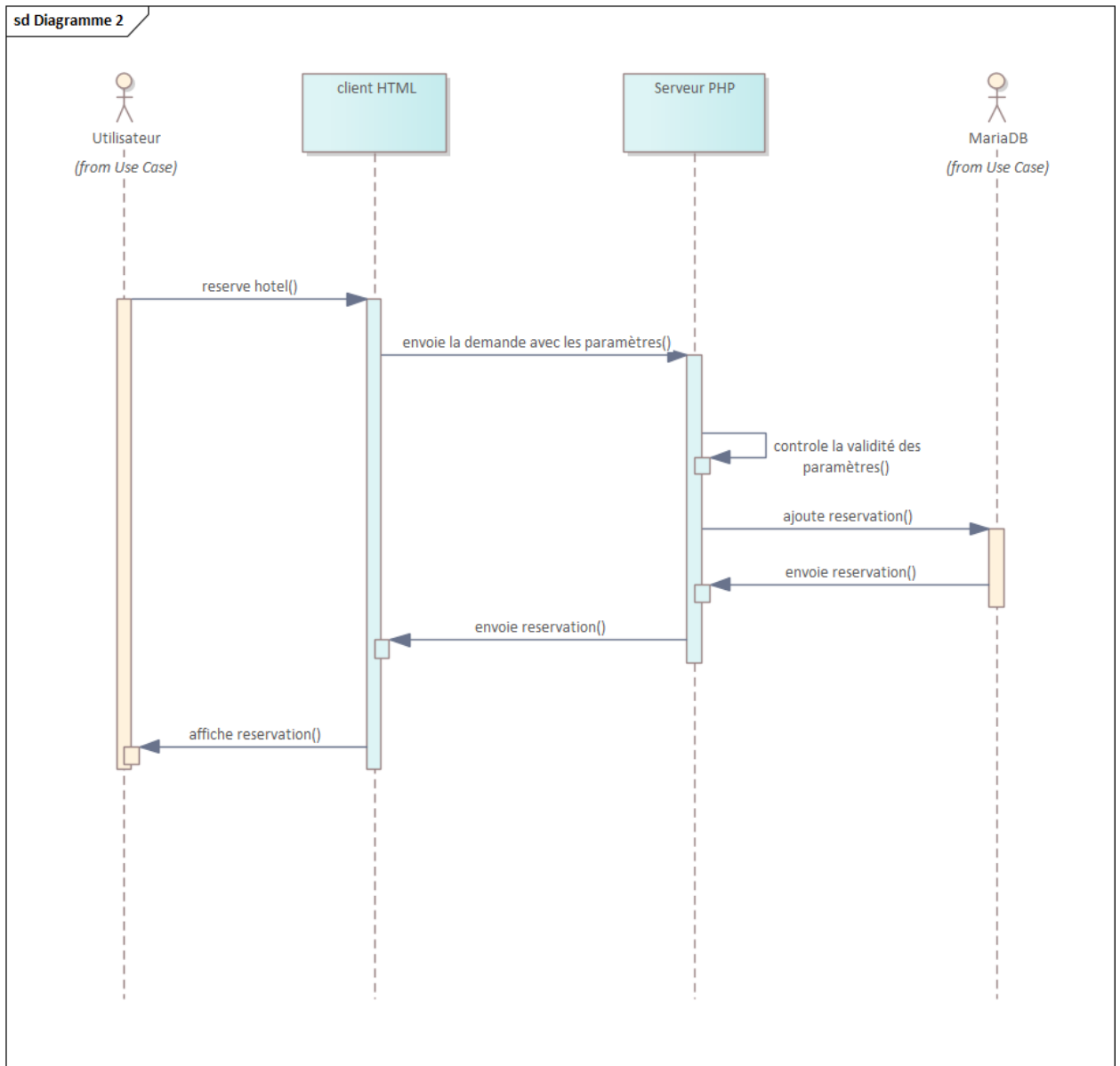


2.6 Diagrammes séquence

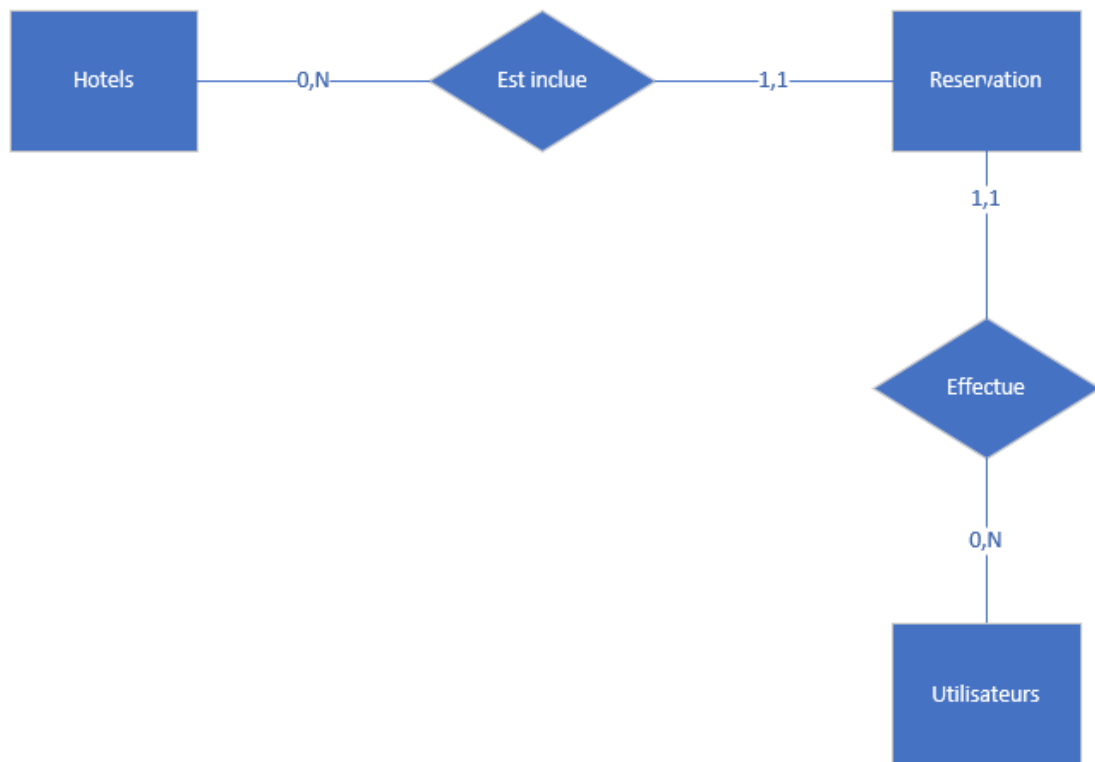
2.6.1 Séquence login



2.6.2 Séquence réservation



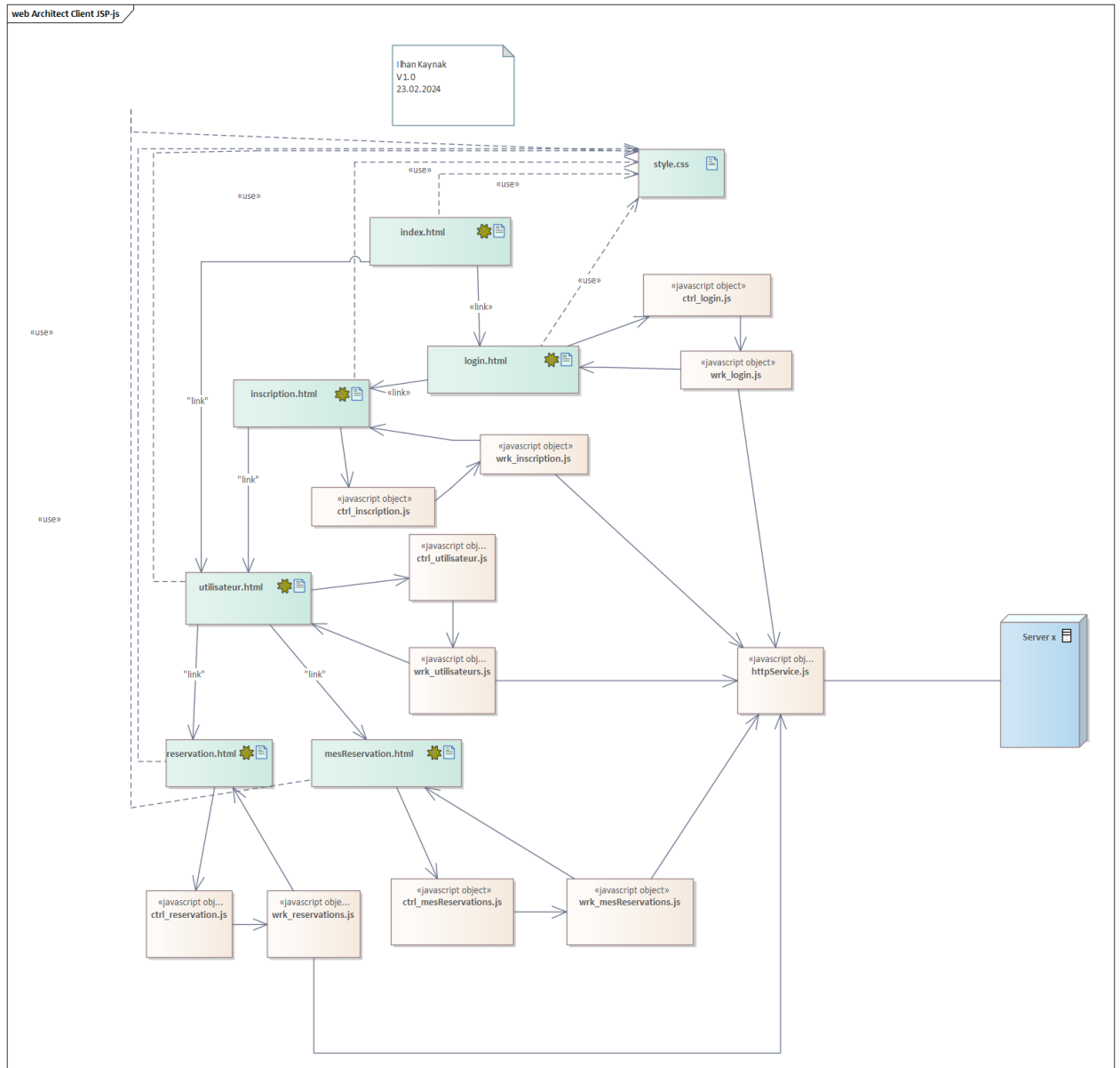
2.7 Schéma ER



3 Conception

3.1 Diagramme de classe

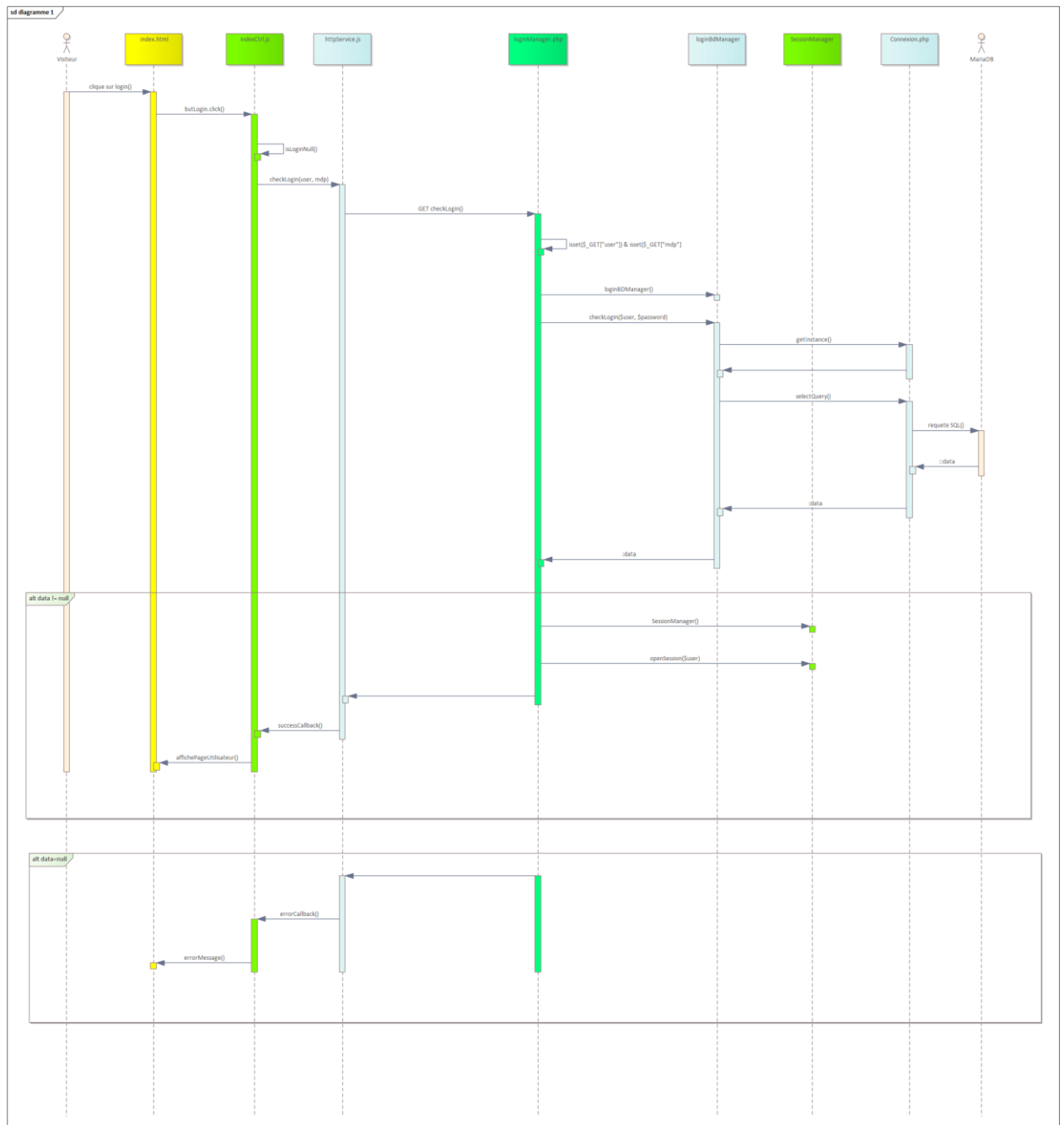
3.1.1 Client



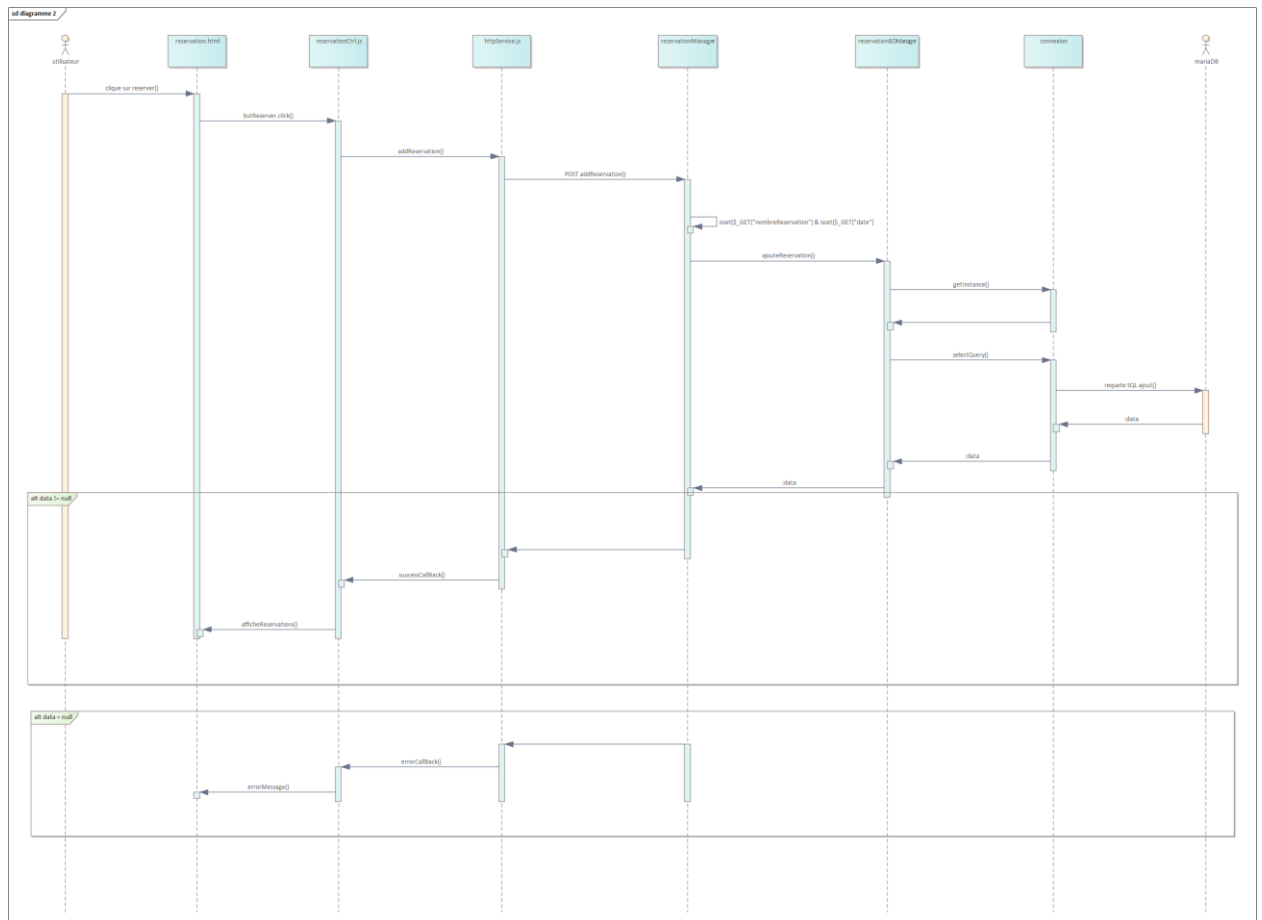
3.1.2 Serveur



Ilhan Kaynak



3.3.2 Réserveation



4 Implémentation

5 Conclusion

5.1 Ce que j'ai appris

5.2 Ce que j'ai aimé

5.3 Ce que j'ai moins aimé