République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université M'Hamed Bougara de Boumerdès

Faculté des Sciences
Département d'informatique

Domaine : Mathématiques Informatique

Filière : Informatique

Spécialité : Technologie de l'information

Année universitaire :

2021 / 2022

Mini projet pour le module technologie des composants

Thème

Application de réservation d'hôtel utilisant NextJS

Présenté par :

Boudraa Mohamed Ouadjih

Bakiri Insaf

Neggazi Mohamed Lamine

RÉSUMÉ

Il existe de nombreuses technologies qui sont utilisées sur Internet pour partager des fichiers, chacune d'entre elles ont des fonctionnalités, des méthodes et des protocoles différents. Cependant, le plus commun et le plus facile est le Web qui a été établi par quelques fonctionnalités simples. Le Web se développe continuellement pour être aussi facile pour les utilisateurs. Les développeurs Web veulent créer une machine qui pense comme des humains en ajoutant de nouveaux outils, méthodes et protocoles au Web actuel.

L'objectif de cet article est de présenter NextJS, l'un des frameworks les plus populaires qui aide les développeurs du monde entier à créer des sites Web, et également de présenter notre exemple de site Web créé par NextJS et hébergé par Vercel.

Le résultat final n'est pas le site Web lui-même, mais plutôt l'expérience de développement dans laquelle les développeurs bénéficieront le plus.

Mots clés : Internet, Web, NextJS, Vercel, code.

ABSTRACT

There are many technologies which are used on the Internet to share files, each of them have different features, methods and protocols. However, the most common and easiest one is the Web which was established by a few simple features. The Web continuously developing to be as much as easy for the users. Web developers want to make a machine which thinks like humans by adding new tools, methods and protocols to the current Web.

The objectif of this paper is to showcase NextJS, one of the most popular frameworks that helps developers around the world create websites, and also showcase our sample website created by NextJS and hosted by Vercel.

The final outcome is not the website itself, but rather the development experince in which developers will benefit the most.

Keywords: Internet, Web, NextJS, Vercel, code.

TABLE DES MATIÈRES

Liste Des Figures

LISTE DES TABLEAUX

INTRODUCTION GÉNÉRALE

Au dernier siècle, L'internet devient de plus en plus important pour nous les humains, car il est utilisé littéralement pour tout ce que nous faisons, nous y accédons quotidiennement pour faire diverses activités telles que faire des achats, communiquer, vérifier la map...Et pour cela on a besoin de sites Web pour faire de telles choses.

De nos jours, le développement d'un site web est un parcours long et compliqué, il est à la fois coûteux et difficile pour les développeurs d'en créer un. Le développement Web a vu un énorme avènement de l'application à page unique (SPA) au cours des deux dernières années. Le développement initial était simple : rechargez une page complète pour effectuer une modification de l'affichage ou effectuer une action de l'utilisateur. Le problème avec cela était un énorme temps d'aller-retour pour la demande complète pour atteindre le serveur Web et revenir au client.

C'est dans ce contexte, le NextJS offre une excellente expérience utilisateur, une facilité d'utilisation et un fractionnement automatique du code qui profite aux développeurs débutants et avancés. Cette solution est devenue populaire car elle résolve un problème que de nombreux développeurs Web avaient l'habitude d'avoir avec les applications Web rendues côté client (dans le navigateur), NextJS fournit une solution prête à l'emploi pour le rendu côté serveur (SSR) des composants React.

Dans ce cadre, on a proposé de faire un système de gestion des chambres d'hôtel on se basant sur la technologie de NextJS.
Dans ce document nous présentons trois chapitres :

- Le premier chapitre contient une étude générale sur la technologie des composants et la façon dont NextJS peut être votre prochaine technologie à apprendre.
- Le deuxième chapitre se compose de divers diagrammes qui sont à l'origine du développement de notre site Web.
- Le troisième et le dernier chapitre est consacré à la réalisation où nous allons définir tous les outils qui nous ont permis de concevoir notre site web. Quelques interfaces y seront présentées.

CHAPITRE I : ÉTUDE GÉNÉRAL

1. Introduction

Dans ce présente chapitre, nous présentons ce qu'est la technologie des composants et comment cela nous aide en tant que développeurs. On va aborder également des sujets détaillés tels que JavaScript et ses bibliothèques et frameworks. Enfin, pourquoi NextJS a-t-il été construit et en quoi cette technologie nous aide ?

2. Programmation orientée composants

2.1 Définition

C'est une approche modulaire de développement des applications informatique, afin de crée des unités d'une façon indépendante de production et de déploiement et qui peut être combiné à d'autre unités (composants) pour former une application. [3]

2.2 La structure d'un composant

Le code d'un composant peut être séparé en deux parties :

- Les méthodes qui ne sont pas accessible de l'extérieur (méthodes privée).
- Les méthodes accessible (les interfaces) est le moyen de communication avec le code client, ou chaque composant possède des interfaces fournie et requise.

2.3 Les avantages de la programmation orientée composants

- **Spécialisation :** L'équipe de développement peut être divisée en sous-groupes, chacun se spécialisant dans le développement d'un composant.
- **Sous traitance :** Le développement d'un composant peut être externalisé, à condition d'en avoir bien réalisé les spécifications au préalable.
- **Facilité de mise à jour :** La modification d'un composant ne nécessite pas la recompilation du projet complet.

- **Facilité de livraison/déploiement :** Dans le cas d'une mise à jour, d'un correctif de sécurité, ... alors que le logiciel a déjà été livré au client, la livraison en est facilitée, puisqu'il n'y a pas besoin de re-livrer l'intégralité du projet, mais seulement le composant modifié.

- **Choix des langages de développement :** Il est possible, dans la plupart des cas, de développer les différents composants du logiciel dans des langages de programmation différents. Ainsi, un composant nécessitant une fonctionnalité particulière pourra profiter de la puissance d'un langage dans un domaine particulier, sans que cela n'influe le développement de l'ensemble du projet.

- **Productivité :** La réutilisabilité d'un composant permet un gain de productivité non négligeable car elle diminue le temps de développement, d'autant plus que le composant est réutilisé souvent.

2.4 Les inconvénients de la programmation orientée composants

Le découpage fonctionnel d'une application en divers composants doit être mené de manière très rigoureuse, tout en pensant au fait que le composant doit être réutilisable, c'est à dire penser également à son utilisation future. Ceci peut amener un système développé en employant la méthode de programmation orientée composants à consommer sensiblement plus de ressources qu'une application programmée avec une méthode traditionnelle.
En effet, le système de communication entre les composants est plus coûteux en temps de calcul qu'un simple appel de fonction.

2.5 Les modèles à composants

A l'heure actuelle, il existe principalement trois fournisseurs de modèles à composants : Sun Microsystems, Microsoft et l'Object Management group (OMG).
Ces modèles sont classés en trois catégories :

- Modèle orienté IHM / Client : OLE, COM, ActiveX, JavaBeans SUN, React JS, Angular JS.

- Modèle orienté métier : ComT, MTS, .NET Microsoft, JavaBeans SUN, NodeJS.

- Modèle généraliste : Fractale de consortium Object Web, OSGI.

3. JavaScript

3.1 Définition

JavaScript souvent abrégé JS, est un langage de programmation créé par Brendan Eich en 1995, est l'une des technologies de base du World Wide Web. JavaScript fait partie de la triade de technologies que tous les développeurs Web doivent apprendre : HTML pour spécifier le contenu des pages Web, CSS pour spécifier la présentation des pages Web et JavaScript pour spécifier le comportement des pages Web. [2] JavaScript interagit avec le code HTML et rend les pages Web plus actives. [5]

3.2 Bibliothèques et Frameworks

Les frameworks et les bibliothèques JavaScript se ressemblent dans la mesure où les deux outils facilitent le travail des développeurs dans certains domaines de la programmation. Les bibliothèques JS sont des collections d'extraits de code pré-écrits qui peuvent être utilisés (et réutilisés) pour exécuter des fonctions JavaScript courantes. D'autre part, les frameworks sont un ensemble complet d'outils qui aident à façonner et à organiser votre site Web ou votre application. Lorsque vous essayez de définir des frameworks dans le contexte du framework JavaScript par rapport à la bibliothèque, pensez-y de cette façon : les bibliothèques JavaScript sont comme des meubles qui ajoutent du style et de la fonction à une maison déjà construite. Alternativement, les frameworks sont un modèle que vous utilisez pour construire la maison elle-même.

Voici quelques-unes des bibliothèques les plus populaires:

- ReactJS
- JQuery
- D3

3.3 ReactJS

React est une bibliothèque JavaScript créée en 2011 par Facebook (maintenant Meta) qui se spécialise dans l'aide aux développeurs pour créer des interfaces utilisateur, ou UIs. En termes de sites Web et d'applications Web, ReactJS tente de résoudre le problème à partir de la couche View. Il peut très bien être défini et utilisé comme le V dans n'importe lequel des frameworks MVC. Il n'a pas d'opinion sur la façon dont il devrait être utilisé. Il crée des représentations abstraites de vues. Il décompose des parties de la vue dans les composants.

Ces composants englobent à la fois la logique pour gérer l'affichage de la vue et la vue elle-même. Il peut contenir des données qu'il utilise pour afficher l'état de l'application.

React est fondé sur l'idée que la manipulation DOM est une opération coûteuse et doit être minimisée. Il reconnaît également que l'optimisation de la manipulation DOM à la main entraînera beaucoup de code standard, qui est sujet aux erreurs, ennuyeux et répétitif. React résout ce problème en donnant au développeur un DOM virtuel à restituer au lieu du DOM réel. Il trouve la différence entre le DOM réel et le DOM virtuel et effectue le nombre minimum d'opérations DOM requis pour atteindre le nouvel état. [6]

4. NextJS

4.1 Définition

NextJS est un framework React réalisé en 2016 par Vercel, une société de serverless cloud computing fondée par Guillermo Rauch en 2015.
Ce framework permet d'activer des fonctionnalités comme la création de sites statiques et le rendu coté serveur.

4.2 Pourquoi NextJS ?

- Permet aux développeurs d'écrire facilement des applications universelles avec React de manière transparente, facile et efficace, un routage simple des pages.
- Code splittent : Cela signifie que vous pouvez charger votre code depuis le serveur uniquement quand vous en avez besoin ; vous pouvez le faire par page ou par composant.
- Il offre un contenu hautement interactif pour les utilisateurs.
- Génère le contenu avancé dans le serveur afin que l'utilisateur voie la page html entièrement rendu. Il a trois façons de générer le contenu de la page web ou il offre une grande flexibilité dans développement :
  - Static Site Generatiion : générer toutes les pages au temps d'exécution (on utilise cette méthode quand on a des données qui ne change pas souvent).
  - Server Side Rendering : générer chaque page au moment de la demande (les données ici change souvent).
  - Incremental Static Regeneration : génère des pages uniques en arrière-plan.

4.3 Les caractéristiques de NextJS

- Standardisation
- Routage intégré des domaines et sous-domaines et détection automatique des langues.
- Support prêt à l'emploi.
- Adaptabilité et réactivité.
- Sécurité des données.
- Idéal pour l'optimisation des moteurs de recherche.
- Délai de mise sur le marché rapide.
- Zéro configuration.
- Fractionnement automatique du code.
- Facilité de mise à niveau.
- Vous pouvez pré rendre une page au moment de la demande (SSR) ou de la construction (SSG).

5. Conclusion

Dans ce chapitre introductif, nous avons présenté la programmation orientée des composants ainsi que la structure d'un composant et les modèles a composant, afin d'éclaircir l'objectif de notre étude. En se basant sur cette étude, nous spécifierons dans le chapitre suivant les différentes fonctionnalités de notre projet.

CHAPITRE II : ANALYSE ET CONCEPTION

1. Introduction

La conception d'un projet est une phase très importante pour définir les objectifs et les fonctionnalités de notre application. Dans ce chapitre nous nous intéressons à l'étude de conception de notre application, nous adoptons l'UML comme langage de modélisation.

Cela consiste à présenter le diagramme de cas d'utilisation décrivant les scénarios nominaux de chaque acteur ainsi que le diagramme composant qui représente visuellement les relations entre les différents composants de notre système et le diagramme de déploiement qui est utilisé pour visualiser les processeurs matériels, les nœuds et les dispositifs d'un système.

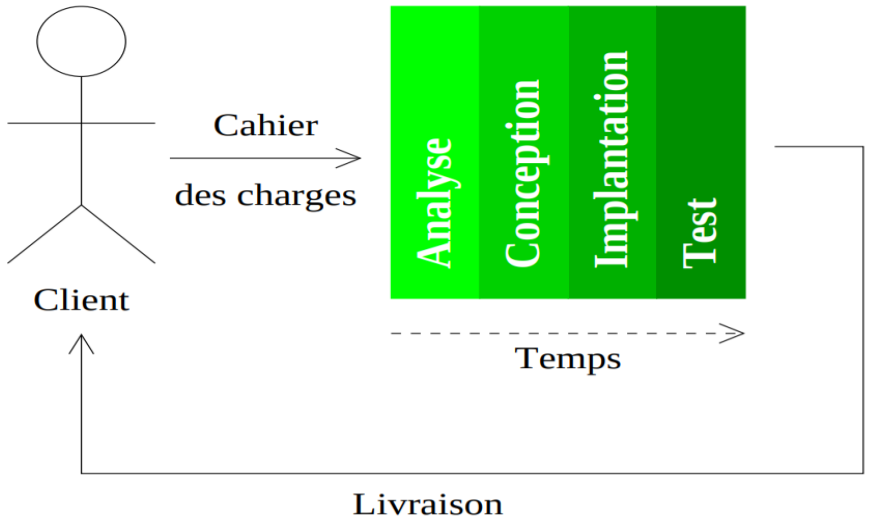

Figure 1: Développement en cascade

2. L'objectif

Pour bien comprendre la technologie de NextJS, nous avons opté la réalisation d'une application web qui répondre aux attentes suivantes :

- Développer un plan d'étude avec une bonne analyse du problème posé.
- Planifier les différentes étapes de conception du projet.
- Réaliser une application englobant le maximum de tâches fonctionnelles avec des solutions optimales.
- Permettre à l'administrateur de gérer la base de données.
- Assure la sécurité de l'application.

3. Spécification des besoins

La phase de spécification des besoins est la première phase formelle obligatoire dans le développement d'une application informatique puisque la capacité de persuasion d'un produit ne peut se réaliser parfaitement sans une spécification préalable élaborée des besoins et des exigences.

3.1 Les besoins fonctionnels

Les besoins fonctionnels servent à présenter les actions que doit effectuer le système en réponse à une demande présentée par un utilisateur.
Notre application doit répondre aux exigences suivantes :

- L'application doit permettre aux clients de consulter le site ainsi que rechercher des chambres sans l'authentification.
- Pour réserver une chambre le client doit s'authentifier.
- Le client peut annuler ça réservation.
- Administrateur gère les comptes des clients.
- L'application doit permettre à l'administrateur de gérer les chambres avec la possibilité d'ajouter ou supprimer ou modifier une chambre.

3.2 Les besoins non-fonctionnels

Les besoins 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.

- **La sécurité :** L'application doit tenir compte de confidentialité des données des utilisateurs.
- **La performance :** Temps de réponse court (une réponse aux requêtes doit être rapide pour minimiser le temps d'attente de l'utilisateur).
- **La fiabilité :** Minimiser des risques et des erreurs afin d'assurer le bon fonctionnement du système.
- **Ergonomie et convivialité :** L'application doit fournir une interface simple et élégante pour l'utilisateur afin de faciliter l'exploitation des services de l'application.
- **Guidage :** L'ensemble des moyens mis en œuvre pour orienter, informer les utilisateurs lors de ses interactions avec l'application.

4. Langage de modélisation

4.1 UML

Unified **M**odeling **L**anguage ou langage de modélisation unifié est un langage de modélisation de développement à usage général dans le domaine du génie logiciel qui vise à fournir un moyen standard de visualiser la conception d'un système

Figure 2: Logo UML (Unified Modeling Language)

5. Les diagrammes utilisés

Pour mieux concevoir notre application, nous avons besoin d'utiliser les trois types de diagrammes suivants :

Diagramme	Objectifs	Type
Diagramme de cas d'utilisation	1. Identifier la fonctionnalité du système 2. Il décrit l'interaction des personnes ou du dispositif externe avec le système en cours de conception. 3. Il résume les relations entre les cas d'utilisation, les acteurs et les systèmes.	Fonctionnel
Diagramme de composants	1. Il permet aux concepteurs d'applications de vérifier que les fonctionnalités requises d'un système sont mises en œuvre par les composants, garantissant ainsi que le système final sera acceptable. 2. De plus, le diagramme des composants est un outil de communication utile entre les parties prenantes pour discuter, analyser ou améliorer la conception du système.	Statique
Diagramme de déploiement	1. Décrire les composants matériels utilisés dans les implémentations de systèmes ainsi que les environnements d'exécution et les artefacts déployés sur le matériel. 2. Il permet de visualiser le système de topologie du matériel, de modéliser les éléments matériels physiques et la relation de communication entre eux, et de planifier l'architecture du système.	Statique

Tableau 1: Les diagrammes utilisé

6. Diagramme de Cas d'utilisation

6.1 Définition

Un diagramme de cas d'utilisation dans sa forme la plus simple est une représentation de l'interaction d'un utilisateur avec le système qui montre la relation entre l'utilisateur et les différents cas d'utilisation dans lesquels l'utilisateur est impliqué. Un diagramme de cas d'utilisation peut identifier les différents types d'utilisateurs d'un système et les différents cas d'utilisation et seront souvent accompagnés également d'autres types de diagrammes.
Les diagrammes de cas d'utilisation contiennent généralement :

- Cas d'utilisation.
- Acteurs.
- Relations de dépendance, généralisation et association.

Comme tous les autres diagrammes, les diagrammes de cas d'utilisation peuvent contenir des notes et des contraintes.

Les diagrammes de cas d'utilisation peuvent également contenir des packages, qui sont utilisés pour regrouper des éléments de votre modèle en blocs plus volumineux. Parfois, vous souhaiterez également placer des instances de cas d'utilisation dans vos diagrammes, en particulier lorsque vous souhaitez visualiser un système d'exécution spécifique.[1]

Figure 3: Exemple de cas d'utilisation

6.2 Identifications des acteurs

Avant d'entamer la présentation des diagrammes, il faut identifier les acteurs qui interagissent directement avec le système. Un acteur est une entité externe qui peut être un utilisateur humain, un dispositif matériel ou un autre système.
Dans notre système on distingue deux acteurs principaux :

- Administrateur : c'est lui qui va gérer les comptes clients ainsi que les chambres d'hôtel (Ajouter, Modifier et Supprimer).
- Clients : c'est lui qui va réserver ou bien consulter le site web afin d'authentifier.

6.3 Identification des cas d'utilisation

Un cas d'utilisation (en anglais use case) permet de mettre en évidence les relations fonctionnelles entre les acteurs et le système étudié. Le diagramme de cas d'utilisation permet de représenter visuellement une séquence d'actions réalisées par un système.

Voici quelques cas d'utilisation de notre système :

- Gérer les comptes clients. Gérer les chambres.
- Rechercher une chambre, Réserver une chambre. . . .

6.4 Diagramme de cas d'utilisation général

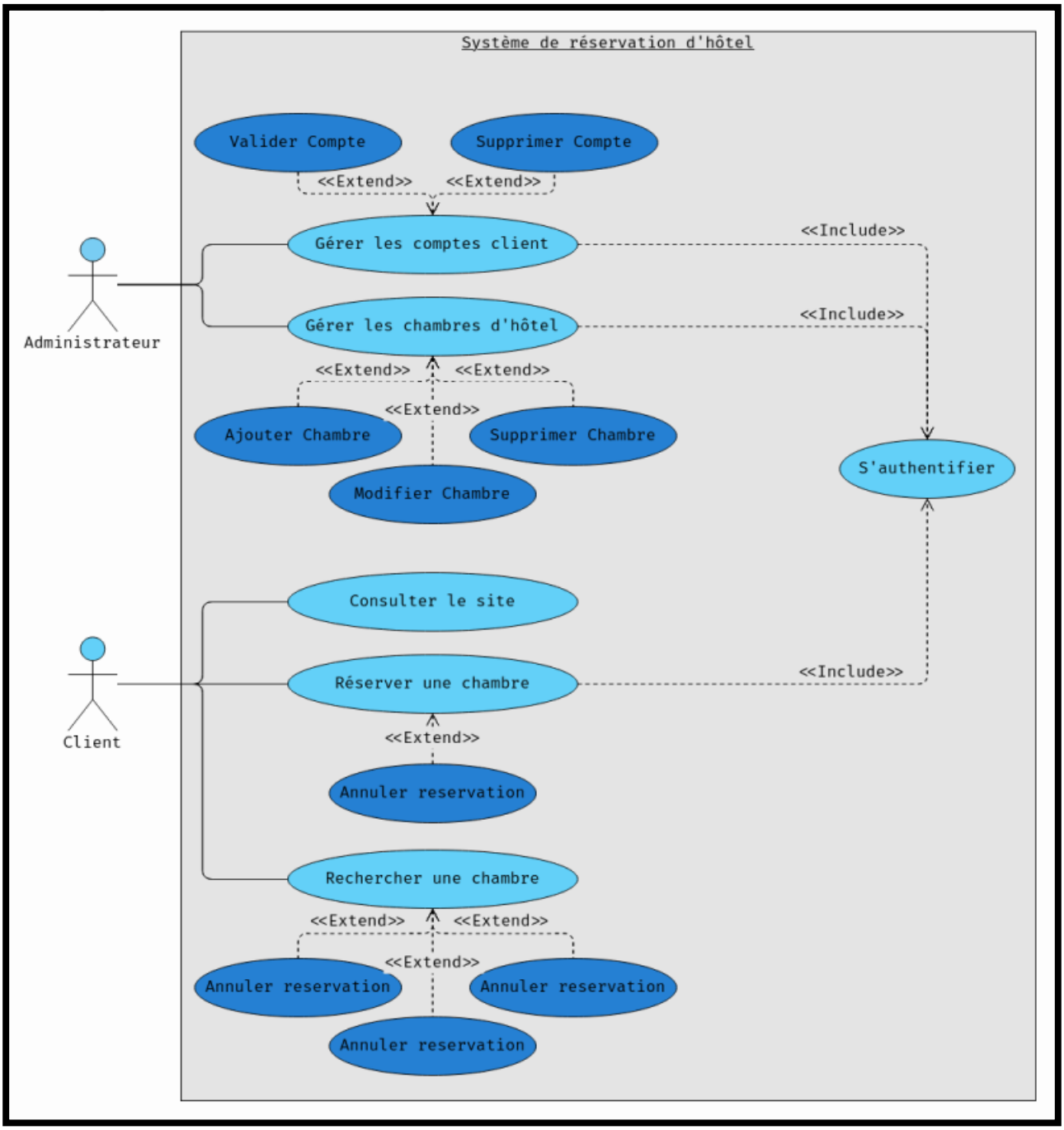

Figure 4: Diagramme de cas d'utilisation général

7. Diagramme des composants

7.1 Définition

Décrivent les composants du logiciel, leurs interfaces et leurs dépendances, les diagrammes de composants sont utiles pour les raisons suivantes :

- Définition des aspects exécutables et réutilisables d'un système logiciel.
- Mise en évidence des problèmes de configuration logicielle à travers les relations de dépendance
- Représentation précise d'une application logicielle avant d'y apporter des changements ou des extensions.

7.2 Diagramme des composants général

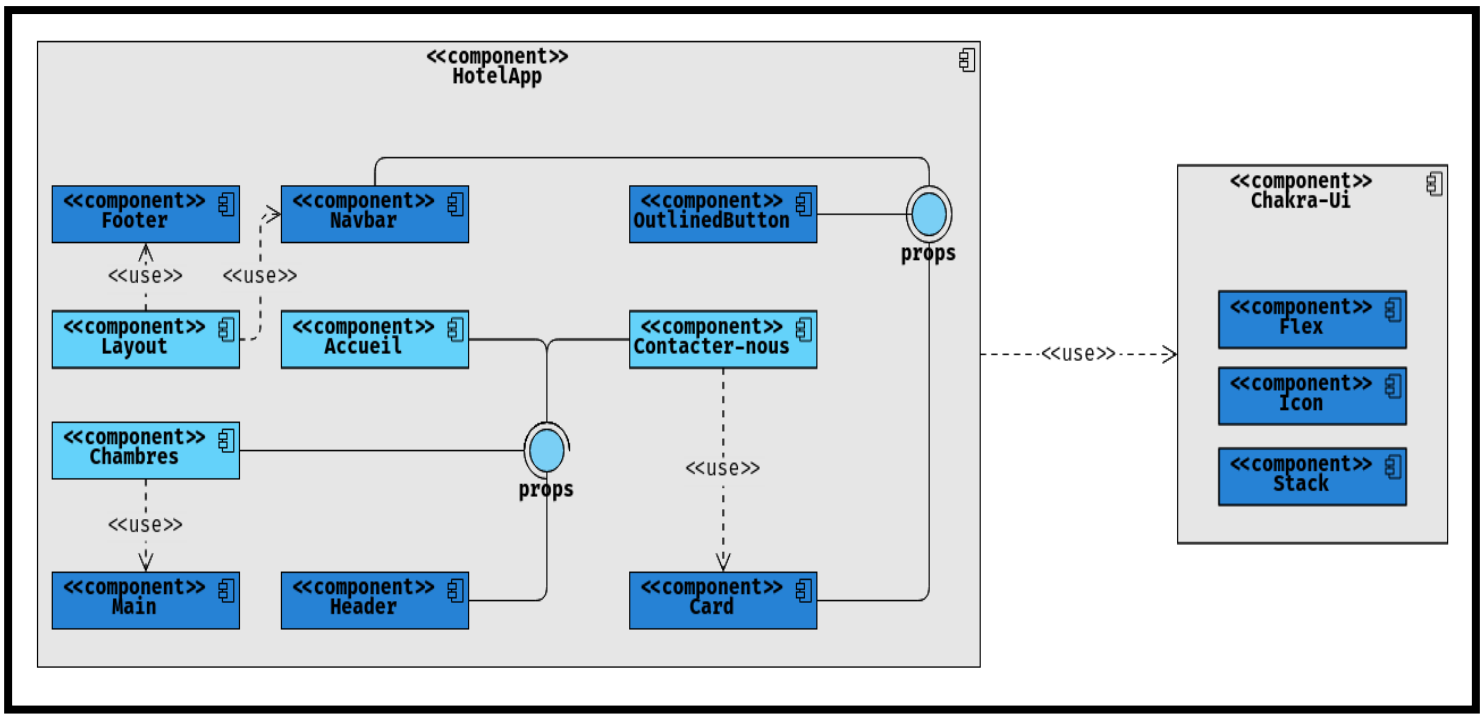

Figure 5: Diagramme des composants général

8. Diagrammes de déploiement

8.1 Définition

Un diagramme de déploiement dans le langage de modélisation unifié modélise le déploiement physique des artefacts sur les nœuds. Pour décrire un site Web, par exemple, un diagramme de déploiement montrerait quels composants matériels «nœuds» existent (par exemple, un serveur Web, un serveur d'applications et un serveur de base de données), sur quels composants logiciels «artefacts» s'exécutent chaque nœud (par exemple, application Web, base de données), et comment les différentes pièces sont connectées (par exemple JDBC, REST, RMI). [1]

Les nœuds apparaissent sous forme de cases et les artefacts alloués à chaque nœud apparaissent sous forme de rectangles dans les cases. Les nœuds peuvent avoir des sous-nœuds, qui apparaissent comme des boîtes imbriquées. Un seul nœud dans un diagramme de déploiement peut représenter conceptuellement plusieurs nœuds physiques, tels qu'un cluster de serveurs de base de données.

8.2 Diagramme de déploiement

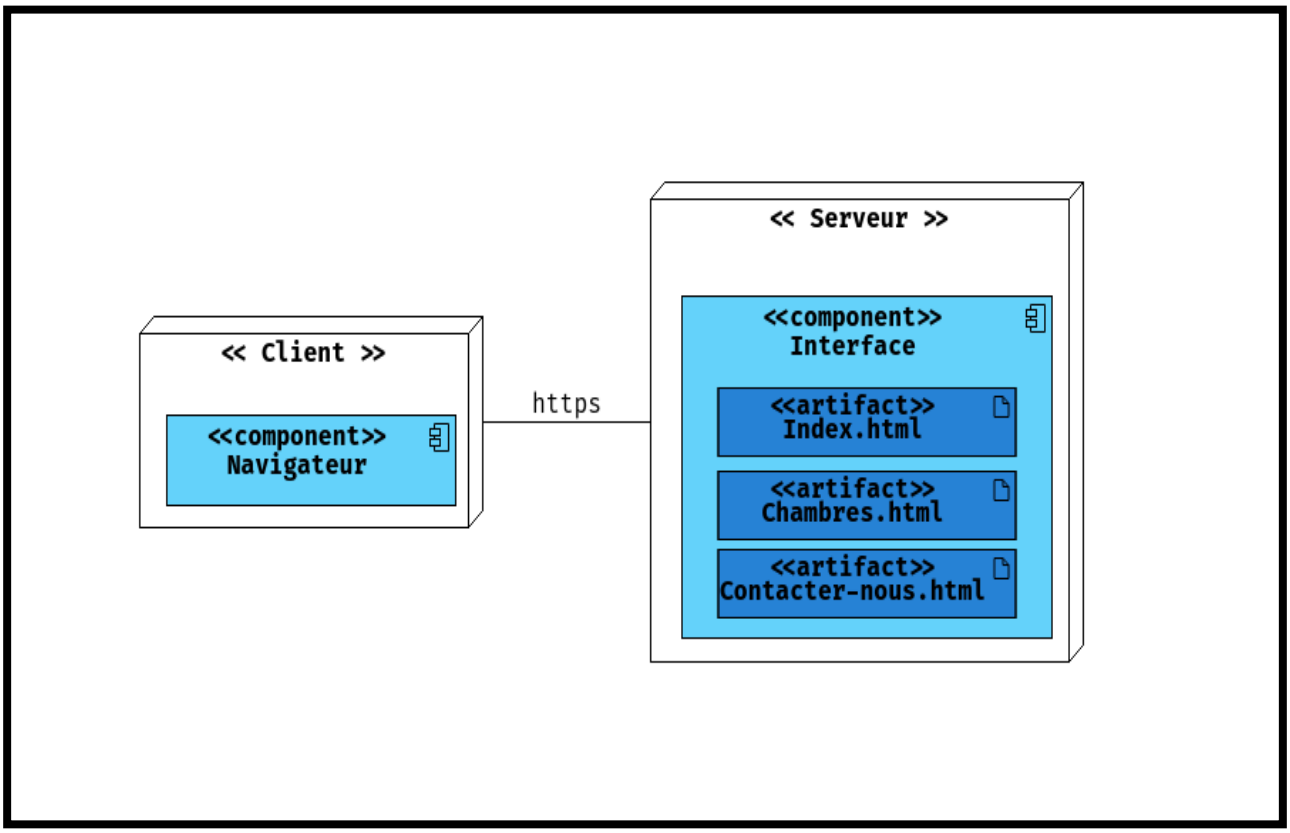

Figure 6: Diagramme des composants général

9. Conclusion

Dans ce chapitre, nous avons défini d'abord la signification de l'analyse et la conception avec les outils qu'on a utilisés pour réussir notre conception, ensuite nous sommes passés aux différents diagrammes là où nous avons expliqué notre solution en détail.

Cette étude nous a permis de définir la structure de l'application et cela aussi nous a facilité l'implémentation que vous verraient dans le chapitre suivant.

CHAPITRE III : IMPLÉMENTATION

1. Introduction

Ce chapitre contient la dernière partie de ce rapport. Il a pour objectif d'exposer le travail achevé. Dans un premier temps, nous présentons l'environnement matériel, logiciel et les différents outils de développement utilisés. Dans un second temps, nous illustrons la réalisation de notre travail par des imprimes écran des interfaces les plus importantes de notre application.

2. L'environnement matériel

Nous présentons dans cette section l'environnement matériel mis à la disposition du présent projet. Pour la réalisation de ce projet, nous avons disposé de 3 ordinateurs caractérisé par :

3. L'environnement logiciel

Dans ce qui suit, nous présentons l'environnement logiciel utilisé pour mener à terme ce sujet :

3.1 Visual studio Code

C'est un éditeur de code open-source développé par Microsoft supportant un très grand nombre de langages grâce à des extensions.

Figure 7: Logo VScode

3.2 Neovim

C'est un éditeur de texte très rapide et fiable qui aide beaucoup en termes de flux de travail et pour couronner le tout.

Figure 8: Logo Neovim

3.3 Visual Paradigme

C'est un outil de modélisation UML, permet de dessiner tous les types de diagrammes.

Figure 9: Logo Visual Paradigm

3.4 JavaScript

Est un langage côté client, c'est à dire qu'il est compilé et exécuté par votre navigateur. Il permet entre autres d'améliorer et d'épicer un peu le HTML grâce à de nombreuses fonctions.

Figure 10: Logo javascript

3.5 React

C'est une bibliothèque JavaScript frontale a code source ouvert permettant de créer des interfaces utilisateur ou des composants d'interface utilisateur.

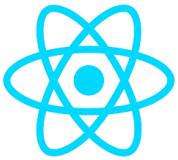

Figure 11: Logo react

3.6 NextJs

C'est un framework gratuit et open source s'appuyant sur la librairie JavaScript React et sue la technologie Node.js.

Figure 12: Logo NextJS

3.7 ChakraUi

C'est une bibliothèque de composants simple, modulaire et accessible qui vous donne les éléments de base dont vous avez besoin pour créer vos applications React.

Figure 13: Logo ChakraUI

4. Exposition de travail réalisé

Dans cette partie, nous présentons notre travail en exposant des captures d'écran des interfaces les plus importantes de notre application.

5. Conclusion

Au cours de ce chapitre dédié à l'implémentation, nous sommes passés de la théorie à la pratique, elle nous a permis d'atteindre nos objectifs décrits dans les étapes précédentes. Nous avons présenté les outils qui nous ont aidé à concevoir notre application, les langage de programmation et les framework en passant par la représentation graphique de l'architecture global de l'application en question et les interfaces pour bien illustrer le travail qui a été fait.

CONCLUSION GÉNÉRALE

Avec l'évolution des technologies de l'information et de communications, on constante chaque jour des milliers d'innovations. A cet effet, beaucoup de chercheurs se sont attelés cette tâche, et proposent des solutions pour l'utilisation.

Au cours de ce travail, nous avons présenté les différentes étapes ayant conduit à la mise en œuvre d'une application web dédiée à la réservation de chambre d'hôtel en utilisant la technologie de NextJS.

Nous avons commencé par définir la programmation orientée composants ainsi que le framework d'application universelle NextJS.

Le langage de modélisation UML est constitué le support de l'analyse des besoins et la conception de notre application web via les différents diagrammes UML couvrant les aspects fonctionnels et statiques de tout le développement.

Pour enfin réaliser l'application, nous avons utilisé le langage JavaScript via le framework NextJS. "La vérité ne peut être trouvée qu'à un seul endroit: le code". [4]

Ce projet a fait l'objet d'une expérience intéressante, très bénéfique pour nous. En effet, il nous a permis d'enrichir nos connaissances théoriques et compétences dans le domaine de la conception et de la programmation. Ajoutant à ceci, la mise en application des connaissances acquises tout au long de nos études.

En plus, c'était une bonne occasion pour réaliser un travail concret avec des objectifs clairs et bien définis.

BIBLIOGRAPHIE

[1] Grady Booch. *The unified modeling language user guide*. Pearson Education India, 2005.

[2] David Flanagan. Javascript: the definitive guide, 2013.

[3] Stéphane Fréno. Compléments poo : Programmation orientée composants. page 6, 2002.

[4] Robert C Martin, James Grenning, and Simon Brown. *Clean architecture: a craftsman's guide to software structure and design*. Number s 31. Prentice Hall, 2018.

[5] Linda Dailey Paulson. Building rich web applications with ajax. *Computer*, 38(10):14–17, 2005.

[6] AM Vipul and Prathamesh Sonpatki. *ReactJS by Example-Building Modern Web Applications with React*. Packt Publishing Ltd, 2016.