République Algérienne Démocratiques et populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université A. Mira Bejaïa
Faculté Des Sciences Exactes
Département D'informatique
Spécialité : Administration et Sécurité des Réseaux Informatique

جامعة بجاية
Tasdawit n Bgayet
Université de Béjaïa

MÉMOIRE DE FIN DE CYCLE

En vue de l'obtention du diplôme de Master en informatique

THÈME

RÉALISATION D'UNE APPLICATION WEB AVEC REACTJS

ÉTUDE DE CAS : GESTION DE RDV CHEZ UN PROFESSIONNEL DE SANTÉ

Réalisé par :

Bousafsafa Souraia

Himeur Chaima

Devant le jury composé de :

Président : M. ALLICHE Abdenour

Examinateur : M.TOULOUM Karim

Examinatrice : Melle AZOUI Aicha

Encadrant : M. IDOUGHI Djilali

Année universitaire 2018/2019

Dédicace

« Louange à Dieu, le seul et unique ».

Je dédie ce modeste travail à

Mon père qui est toujours été là pour moi, qui m'a Donné un magnifique modèle de labeur et de persévérance, pour son
Attention, et soutien tout au long de mes études.

Et surtout à la mémoire de ma chère maman qui nous a quittés très tôt et dont je garde que de bons souvenirs, que dieu l'accueille de son vaste paradis. A ma belle mère

A mes cher frères « Farid et Yasser».

A mes sœurs Chacun avec son nom.

A tout notre petite famille que je porte dans mon cœur.

A mon binôme (Chaima).
A tous mes chères amies spécialement.

A toute personne qui m'a aidé pondons toute ma carrière

n'A toute personne qui me connaisse.

À tous mes camarades.

Bousafsafa Souraia.

Dédicace

« Louange à Dieu, le seul et unique ». Je

dédie ce modeste travail à

Mes chers parents quiont toujours été là pour me soutenir et m'encourager, je vous remercie pour votre éducation et tous les sacrifices que vous m'avez donne tout au longue de ma vie que dieu vous protéger pour moi

A mon cher mari Belgacem A ma chère tanteNessrine et son mari sofiane Ames chers frères « Marouan, djaber et mohamad ».

A mes sœurs Aya, Roudayna, Rihab et son fils Haytham.

A tout notre petite famille que je porte dans mon cœur.

A mon binôme (Souraia).
A tous mes chères amies spécialement.

A toute personne qui m'a aidé pondons toute ma carrière

Himeur Chaima

Remerciement

Nous tenons au premier, à remercie le dieu, qui nous avoir donnée la force et la patience Pour finir notre modeste mémoire.

Aussi, nous tenons à remercie notre encadreur M. IDOUGHI Djilali.

A notre jury pour l'intérêt d'accepter et d'examiner notre travail.

Sans oublier nos parents pour le courage le long de notre parcours.

Enfin, pour que on oublier aucun, nous tenons
À remercier toutes les personnes qui ont
Participé prés ou loin pour réaliser ce
Travail.

« Nous espérons que vous l'aimez notre mémoire ».

Sommaire

Chapitre 02 : Généralité Sur le développement Web

Chapitre 03 : étude de cas

Chapitre 04 : Réalisation et déploiement

Liste des figures

Liste des tableaux

Liste des abréviations

API : interfaces de programmation d'applications.

REST: REpresentational State Transfer.

QL : QueryLanguage (langage de requête).

NPM: Node.js Package Manager.

JSON: JavaScript Object Notation.

BSON: Binary JSON.

MVC: Model-View-Controller.

JSX : JavaScript Syntaxe Xtension (l'extension supplémentaire de syntaxe Javascript).

RDBMS: Relational data base management system.

HTTP*:* HyperText Transfer Protocol

CMS : Content Management System

Introduction générale

Introduction générale

Soigner plus : plus de patients, plus vite, plus efficacement, mais avec moins : moins de ressources, de personnel, de budget ; voilà le challenge auquel se trouve confronté aujourd'hui le monde de la santé en général.

Pour résoudre ce casse-tête, les médecins doivent avoir une approche globale du parcours du patient qui se présente à une clinique, d'y mettre en parallèle les innovations technologiques, afin de proposer de nouveaux modèles de prise en charge qui permettront de solutionner le challenge mentionné en haut.

Actuellement, le monde connait une avance technologique considérable et cela `a l'aide de l'informatique qui étant une science de traitement automatique de données s'avère bénéfique dans tous les domaines qu'ils soient scientifiques ou professionnels, privés et/ou publics. L'informatisation est donc le phénomène le plus important de notre époque. Elle s'immisce maintenant dans la plupart des objets de la vie courante. Le système de prise de rendez-vous en ligne est l'une des applications de santé grand public qui gagnent en popularité.

De plus l'Internet aujourd'hui occupe une place de choix dans la vie de tous les jours à travers ses multiples services et plus particulièrement son gain de temps. On a tenu compte de cette innovation et de tous ces opportunités qu'elles présentent tel que la création d'une application web qu'on l'en a opté pour notre projet de fin d'étude

L'application permettra notamment à un médecin, qu'il utilise ou non un logiciel de gestion d'agenda, d'offrir des plages de rendez-vous à ses Patients et ce, dans la clinique où il pratique.

Dans ce cadre, et afin de réaliser ce projet de fin d'études, on a fait une étude sur l'application web, leurs techniques, leurs systèmes d'exploitation et leurs outils de développement.

Ainsi, l'objectif de notre projet est de développer un système de gestion de RDV fiable, conviviale et facile à s'intégrer dans l'environnement de travail. Cette application vise essentiellement à faciliter aux patients de voir les médecins disponibles et prendre leurs rendez-vous à distance via leurs Smartphones ou tablettes et à diminuer la charge quotidienne des patients en diminuant le temps d'attente dans la prise d'un rendez-vous.

.

Chapitre 1 :

Introduction aux technologies du développement web mobile

Introduction

Le domaine informatique bien qu'étant jeune, a une évolution croisière**.** Développer une application web nécessite deux composants clés pour fonctionner **: un client et un serveur web.**

Du côté client on retrouve les navigateurs web et tout dispositif utilisé par les internautes pour afficher et interagir avec une application web. De ce côté client, on parle de développement **Front-End.**

Côté serveur C'est là que sont stockés tous les fichiers et données associés à l'affichage d'une application web. On parle du **Back-End** qui se compose au minimum d'un serveur, d'une application et d'une base de données.

Dans ce cadre ils sont développé des nouvelles technologies qui nous aide à éviter les problèmes rencontré leur de la programmation. Nous allons présenter quelque une qu'on va utiliser dans notre travail.

1. Technologies web

Les technologies web sont un ensemble de technologies qui composent et utilisent le World Wide Web (généralement abrégé en Web) et ses normes. Le web a été créé en 1990 comme application de partage d'informations à travers le monde et leur échange en utilisant le protocole HTTP puis est devenu une plate-forme à part entière sur laquelle sont développées régulièrement des nouvelles technologies il a changé la manière des gens de communiquer, de faire du commerce et de partager des informations. Il permettre la manipulation d'applications hébergées sur des serveurs Dans de telles applications, les programmes ainsi que les données résident sur un serveur web contrairement aux applications classiques qui résident sur l'ordinateur de l'utilisateur. Pour les développeurs, la mise à disposition d'une application hébergée sur un serveur web est beaucoup moins coûteuse que le fait de graver un CD et de l'envoyer en grande distribution. Les améliorations subséquentes sont également facilitées

2. les technologies utilisé

2.1GraphQL https://graphql.org/learn/[2]

GraphQL est un langage de requête pour notre API et un environnement d'exécution côté serveur permettant d'exécuter des requêtes à l'aide d'un système de types que nous définitions pour nos données. GraphQL n'est lié à aucune base de données ni à aucun moteur de stockage spécifique, mais est soutenu par notre code et nos données existants.

Langage de requêtes

GraphQL est en premier lieu un langage de requête. Celles-ci ressemblent comme on peut le voir dans l'exemple ci-dessous à la réponse que l'on souhaite obtenir sans les valeurs. Le client choisit alors quels champs de chaque *Objet* (ex : orders) il souhaite et dans quel ordre. La réponse du serveur GraphQL sera alors identique à l'ordre dans lequel a été formulée la requête.

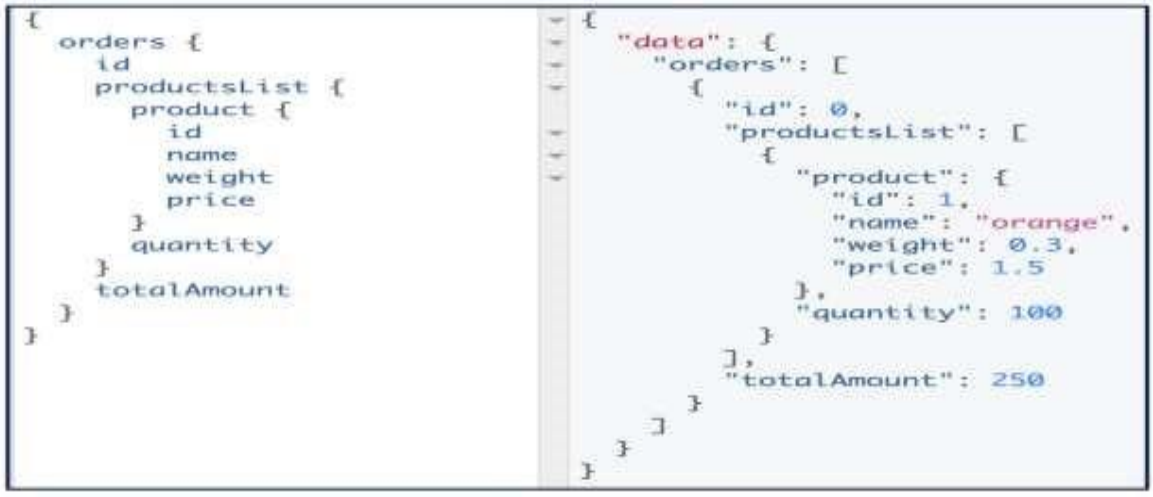


Figure 1 : Exemple de requête GraphQL (à gauche) et sa réponse (à droite)

Dans un second temps, GraphQL est un environnement d'exécution qui interprète et structure ces requêtes à partir du schéma. En effet, après avoir vérifié que la requête correspond bien à la syntaxe du langage, le serveur GraphQL vérifie que la requête est bien disponible dans *Query* (défini ci-dessous) et que les champs demandés correspondent bien au retour de la requête en question.

Le schéma est donc un élément central dans la conception d'une API GraphQL. On y définit tous les objets GraphQL, leurs types (string, int, objet précédemment défini… ou encore un tableau des éléments précédents). On y définit aussi toutes les requêtes disponibles dans l'objet global *Query*, toutes les actions disponibles dans l'objet global Mutation et leurs éventuelles entrées. Enfin il est directement possible de spécifier des règles métier comme par exemple l'ajout d'un point d'exclamation si un champ est obligatoire

2.2.1 Avantages de GraphQL

Parfois, il n'est pas possible d'extraire des données avec une seule demande. C'est ici que GraphQL est utile .GraphQL structure les données sous forme de graphique avec sa syntaxe de requête puissante pour parcourir, extraire et modifier les données.

- Envoyez une requête GraphQL à une API et obtenez exactement ce dont on a besoin. Les requêtes GraphQL renvoient toujours des résultats prévisibles.
- Décrire ce qui est possible avec un système de types

Enfin GraphQL s'avère plus que pertinent pour le développement d'API avec certes des Contraintes mais surtout des avantages indéniables par rapport à d'autres solutions. Son utilisation par des grands acteurs de la technologie prouve qu'à l'état de l'art il y a des enjeux forts autour de cette technologie. Pour nous, GraphQL apporte aussi des réponses à des problématiques existantes telles que le manque de

https://graphql.org/learn/

Standards dans REST, les difficultés autour de la documentation et la communication etc… et de là viennent toute l'agitation qui a entouré sa publication en open source. MAIS GraphQL n'est pas non plus la réponse à tout et encore moins un remplaçant à REST. C'est au contraire un autre outil complémentaire dans notre boîte à outils de développement d'API.

2.2Node.jshttps://nodejs.org/[3]

Node.js n'est pas un langage de programmation. Ce n'est pas non plus un framework JavaScript. Node.js est un environnement d'exécution JavaScript. Est une plateforme logicielle avec une architecture orientée événements qui **permettent** d'utiliser le langage de script JavaScript**,** initialement développé pour une utilisation côté client**,** pour une utilisation côté serveur. Cela fonctionne comme PHP, Java. Il est utilisé pour le développement d'applications JavaScript côté serveur qui doivent assumer de fortes charges en temps réel. L'environnement d'exécution est apprécié pour la réalisation de serveur Web léger.
Les développements de Node est financés par Joyent, à commencer par l'embauche de Ryan Dahl.

En 2006, Ryan Dahl est un étudiant américain en troisième année de doctorat de mathématiques.
Ryan admet que JavaScript dispose des caractéristiques idéales, même s'il n'est pas un adepte du langage. Il manque juste aux machines virtuelles JavaScript la capacité d'accéder à des sockets, au système de fichiers et à d'autres fonctions système.
De ses efforts naît Node.js.Ryan Dahl, alors développeur actif de Node, annonce le dépôt de marque par l'entreprise Joyent. En avril 2011, Joyent dépose la marque Node.js ainsi que son logo.

2.2.1Les modules et les packages de Node.js

Une des principales forces de Node.js est d'être modulaire et de proposer de nombreux modules réseau. Si certains de ces modules sont installés directement en même temps que Node.js, la plupart doivent être installés à la demande.
Lors de la création d'une application qui exige l'installation de modules, deux méthodes sont possibles pour effectuer celle-ci :
– Directement avec le gestionnaire de modules npm (et son option install).
– Indirectement (mais toujours avec npm) via des modules nécessaires à celle-ci dans un fichier nommé package.json.

➢**Le gestionnaire de modules de Node.js : npm**npm est le gestionnaire de modules de Node.js (il est installé avec celui-ci).Les modules sont installés globalement dans le dossier node_modules, situé au niveau des répertoires système si l'option -g est utilisée :
npminstall -g <module>ou sinon (sans l'option g) dans le répertoire courant (mais également dans un dossier nommé node_modules).
La commande permet notamment d'installer et de mettre à jour les modules que l'on sélectionne pour nos projets.

➢**Spécification des dépendances : le fichier package.json**
Pour spécifier les dépendances nécessaires à la création d'une applicationNode.js, il est recommandé de créer un fichier nommé package.json

2.2.2 Les bibliothèques standard Node.js

Voici la liste des bibliothèques contenues dans Node.js considérées comme stable avec une petite description

- **REPL :** c'est l'interpréteur que vous avez quand vous tapez *node* dans votre console.
- **assert :** pour faire des tests.
- **console :** pour les logs.
- **debugger :** point d'arrêt, step, ...
- **dns :** les noms de domaines.
- **event :** tout sur la gestion des événements.
- **fs :** tout sur le système de fichiers.
- **global :** tout ce qui est tout le temps disponible.
- **http :** un serveur, un client, requête, réponse, ...
- **net :** wrapper réseau asynchrone.
- **path :** gestion des chemins sur un système de fichier.
- **os :** gestion du système : dossiers temporaires, noms d'hôtes, ...
- **querystring :** échapper, analyser les arguments d'une requête. **string_decoder :** permet de passer d'un buffer à une chaine.
- **timers :** global, permet d'appeler régulièrement des actions, poser un délai avant, ...
- **tls :** SSL, chiffrer les échanges réseaux.
- **dgram :** datagram, UDP.
- **util :** différents outils, héritage, tests de type, ...
- **zlib :** compression et lecture des formats gzip.

2.2. Le choix de nodejs

Node.js peut être comparé à Python, Java, PHP. On' a choisi Node.js parce qu'il permet de faire du développement avec le langage Javascript. Le fait de passer d'un langage de programmation à l'autre est pénible. Aussi node est un logiciel libre (licence MIT), performance du moteur v8, modèle non bloquant, communauté très active, système de paquet intégré (NPM).

- Node est tout désigné pour créer des applications à nombreuses actions concurrentes ; autrement dit, dès qu'une application ou programme fait appel à des accès réseau, aux fichiers ou au système.
- Node est également adapté pour transformer des flux importants de données en économisant la mémoire.
- Le mécanisme de modules de Node encourage à respecter le principe de responsabilité unique

- Nos applications seront modulaires et autonomes au lieu d'être lourdes et monolithiques.
- Node.js nous permet d'utiliser le langage JavaScript sur le serveur… Il nous permet donc de faire du JavaScript en dehors du navigateur il bénéficie de la puissance de JavaScript pour proposer une toute nouvelle façon de développer des sites web dynamiques.

2.2. Node.js et Express

Express est une infrastructure d'applications Web Node.js minimale et flexible, qui fournit un ensemble robuste de fonctionnalités permettant de développer des applications Web et mobile [11]. Il facilite le développement rapide d'applications Web basées sur des nœuds. Voici quelques-unes des principales fonctionnalités d'Express Framework :

- Permet de configurer les middlewares pour répondre aux requêtes HTTP.
- Définit une table de routage utilisée pour effectuer différentes actions en fonction de la méthode HTTP et de l'URL.
- Permet de rendre dynamiquement les pages HTML en fonction de la transmission des arguments aux modèles.

2.3 Apollo Server https://apollographql.com/docs/apollo-server/[4]

Apollo Server est une bibliothèque qui nous aide à connecter un schéma GraphQL à un serveur HTTP dans Node.js. Il est possible d'utiliser Apollo Server avec tous les serveurs HTTP populaires.

Apollo Server est le meilleur moyen de créer rapidement une API prête à la production et auto documentée pour les clients GraphQL, en utilisant les données de toutes les sources. Il est open source et fonctionne parfaitement comme serveur autonome, complément d'un serveur HTTP ou dans des environnements «sans serveur».

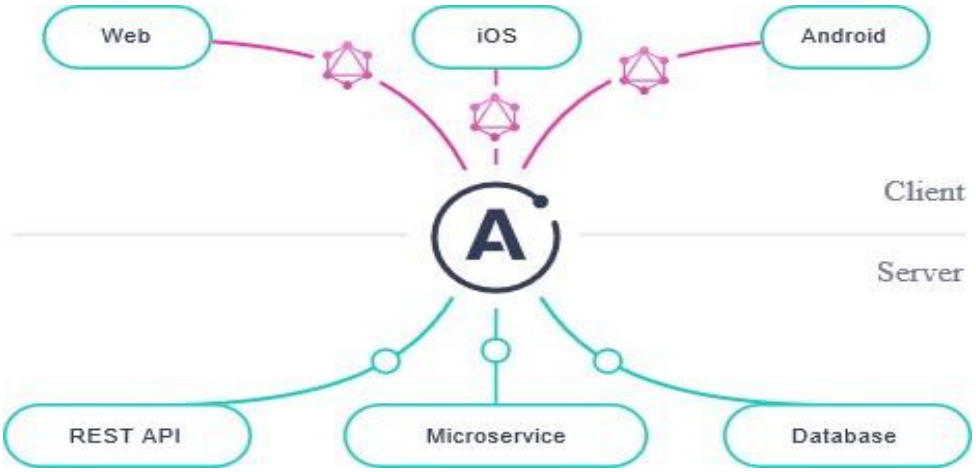


Figure 2 : Apollo Server

Apollo Server implémente un serveur GraphQL conforme aux spécifications qui peut être interrogé à partir de tout client GraphQL, permettant :

https://apollographql.com/docs/apollo-server/

- **Un début facile**, afin que les développeurs front-end et back-end puissent commencer à récupérer des données rapidement.
- **Adoption incrémentielle**, permettant d'ajouter des fonctionnalités avancées lorsque cela est nécessaire.
- **Compatibilité universelle** avec toutes les sources de données, tous les outils de construction et tous les clients GraphQL.
- **La préparation à la production** et ce que vous construisez dans le développement fonctionnent très bien en production.

Apollo Server est conçu avec les principes suivants :

- **Par la communauté, pour la communauté** : Le développement d'Apollo Server est dicté par les besoins des développeurs.
- **Simplicité** : En simplifiant les choses, Apollo Server est plus facile à utiliser, à contribuer et à être plus sécurisé.
- **Performance** : Apollo Server est bien testé et prêt pour la production.

2.4MongoDB https://www.mongodb.com/ [5]

C'est l'entreprise 10gen qui est à l'origine du développement de MongoDB en 2007. C'est une base de données open source, centrée sur les documents. MongoDB a été mis sur le marché en l'espace de deux ans seulement et il aura fallu peu de temps pour que celui-ci devienne une des bases de données NoSQL les plus populaires.

La société 10gen a depuis changé son nom en MongoDB. Et est responsable du développement logiciel tout comme de la distribution de solutions informatiques pour les entreprises. MongoDB a été codé en langage de programmation C++ et enregistre les données en format **BSON**,basé sur le format de JSON. MongoDB travaille sur le concept de collection et de document.

2.4.1. MongoDB et son NoSQLlanguage

Avant la vague de base données NoSQL qui a commencé dans les années 2000, les développeurs utilisaient les bases de données RDBMS. Sont comme SQL Server et Mysql. Pour travailler avec ces bases de données, il faut apprendre le language SQL. Chez MongoDB, une table devient une collection, une ligne dans une table devient un document et une colonne devient un champ (Field). Une collection est un ensemble de document.

MongoDB est un excellent choix pour la mise en place de projets Web qui se basent sur une importante masse de données déstructurées. Le travail orienté documents est prédestiné à de nombreux types de documents qui doivent être stockés et exploités avec rapidité.

Pour assurer la sécurité et la disponibilité des données sur le long terme, des copies peuvent être effectuées facilement avec MongoDB, et mises à disposition sur plusieurs serveurs. L'ensemble des données, présente de nombreux avantages comme la **Disponibilité** (l'application est toujours disponible, même en cas de panne de serveur), **Flexibilité** (notre projet doit toujours pouvoir être adapté de manière dynamique.

2.4.2 avantages de MongoDB

- **La rapidité :** Les requêtes MongoDB peuvent être beaucoup plus rapides dans certains cas, d'autant plus que vos données sont généralement toutes situées en une fois et peuvent être récupérées dans une seule recherche.

- **La Flexibilité :** MongoDB n'exige pas une structure de données unifiée sur tous les objets, de sorte qu'il n'est pas possible de s'assurer que les données seront structurées de manière cohérente, MongoDB peut être beaucoup plus simple à utiliser. Cependant, la cohérence des données est une bonne chose.

Conclusion

De la même manière que pour la création d'applications web, il est fondamental d'établir une bonne expression des besoins. Nous sélectionnerons alors les technologies les mieux qualifiée pour répondre aux objectifs que nous attendrons. Dans le cadre de la création d'applications web, il y a un certain nombre de possibilités. Il faut étudier plusieurs solutions.

Notre projet est spécialisée dans la conception puis le développement d'applications ont utilisons quotidiennement les technologies citer dans ce chapitre.

Chapitre 2 : généralité sur le développement web

Actuellement, le monde connaît une avance technologique considérable dans tous les secteurs et cela grâce à l'informatique qui est une science qui étudie les techniques du traitement automatique de l'information. Elle joue un rôle important dans le développement de travail**.**

Avant, on enregistrait toutes les informations manuellement sur des supports en papier ce qui engendrait beaucoup de problèmes tel que la perte de temps considérable dans la recherche de ces information

L'objectif de notre projet -présenté dans ce Mémoire- est la conception et la réalisation d'un système de prise de rendez-vous en ligne, disponible 24h/24 et 7 jours/7. Et pour cela on a choisi d'utiliser reactjs pour développer une application web qui répond aux besoins d'un patient a la recherche d'un rendez-vous avec un médecin.

La première partie de ce chapitre aborde la notion de développement web et la deuxième partie consacrée le développement d'application web avec reactjs.

1. Le développement web

Le développement web est un terme qui désigne le fait de rédiger le contenu d'un site web dans un langage informatique et technique.il est née avec l'internet**,** en effet les deux choses sont pratiquement indissociables.

Le développement web est par ailleurs une pratique qui se plie aux attentes et aux besoins du client puisqu'il pourrait seulement s'agir d'une écriture de quelques textes et autres éléments visuels qui donnent un air de document à l'interface de notre site, comme il peut également s'agir de toute une conception d'un système interactif qui ira jusqu'à permettre à l'utilisateur de procéder à des transactions commerciales, comme c'est le cas dans les sites web de ventes en ligne.

Le développement web repose sur l'utilisation des langages (HTML/CSS, JavaScript, PHP…) pour écrire des programmes qui sont ensuite exécutés par les ordinateurs. Les instructions sont mises en place sur Internet et sont effectuées sur des serveurs. En fonction des besoins des propriétaires du site ou des pages web.

2. Les langages de programmation web

Il existe différentes sortes de langages de programmation web qui se distinguent par leurs caractéristiques. Les développeurs peuvent faire leur choix parmi ceux qui sont les plus prisés.

Java

Le Java est un langage de programmation open source, indépendant de toute plateforme. Sa polyvalence en fait une option pour presque tous les types de projets. Comme la plupart des langages de programmation Web les plus courants, il est en paradigme orienté objet, ce qui signifie qu'il se détermine en fonction de son champ d'application concret. Un très grand nombre de bibliothèques et de frameworks Web généralement bien fournis et détaillés sont disponibles pour faciliter la réalisation de projets très complexes. En outre, les programmes écrits en Java sont extensibles, évolutifs et faciles à entretenir, sous réserve que le programmeur connaisse son métier.

JavaScript

Ce langage de script dynamique et orienté objet n'a rien à voir avec Java, si ce n'est que les deux sont dérivés du langage C. Il a été développé à l'origine par Netscape en 1995 sous le nom de LiveScriptdans l'objectif d'étoffer l'utilisation d'HTML et des CSS. Il s'agissait de permettre aux programmeurs d'évaluer les interactions entre utilisateurs et de présenter les contenus de façon dynamique. Aujourd'hui, le JavaScript est utilisé non seulement dans les navigateurs, mais aussi dans les microcontrôleurs set les serveurs. Le nom de JavaScript a été choisi à des fins de marketing, pour profiter de la popularité de Java. Le succès a été au rendez-vous : aujourd'hui, presque tous les sites Internet les plus connus utilisent le JavaScript comme langage de programmation de choix côté client

PHP

Hypertext Preprocessor, mieux connu sous son acronyme PHP, est un langage de script dérivé des langages C et Perl. Il est utilisé principalement pour programmer des pages Web et des applications Internet dynamiques**.** PHP est un langage qu'il est possible d'intégrer dans du HTML. Les avantages les plus importants de ce langage résident dans le vaste soutien dont il bénéficie à travers l'existence de différentes bases de données et son association efficace avec les protocoles IP. Depuis sa création, le PHP a connu quelques mises à jour ; aujourd'hui, le langage en est à sa version 7. Il est disponible gratuitement sous licence open source

Python

Python est un langage de programmation dit « évoluer » qui se fonde sur un code à la fois compact et compréhensible. Python est également facile à exploiter ; par exemple, son indentation permet de délimiter les blocs de code par des espaces plutôt que par des symboles.

C'est pour cela que ce langage est considéré comme agréable à apprendre et à utiliser. Il permet de programmer comme on le souhaite : orienté objet, aspect ou fonction. En outre, Python est un langage dynamique, ce qui lui vaut d'être souvent utilisé comme langage descriptif.

C++

Le C++ est dérivé du C, l'un des plus anciens langages de programmation. Envisagés au départ comme un approfondissement du C, les travaux sur le C++ ont débuté dès 1979, mais ce n'est qu'en 1985 qu'il a été mis à disposition du grand public. Sa popularité n'a jamais baissé depuis lors. Le C++ est un langage de programmation qui fait l'objet d'une norme ISO (délivrée par l'Organisation internationale de normalisation). Il est considéré comme un langage proche de la machine, efficace, capable d'un haut niveau d'abstraction et complexe. Le C++ est fondamentalement simple à apprendre, avant tout parce que son noyau de langage est mesurable à l'aide d'environ 60 mots-clés à peine. Il gagne en ampleur et en complexité grâce à sa bibliothèque standard.

C#

Le langage de programmation C# (prononcé « C sharp ») est considéré comme un langage de programmation général. Il est typé, orienté objet et fondamentalement indépendant de toute plateforme. Il s'agit toutefois d'un projet Microsoft développé à l'origine pour le framework .NET. On trouve souvent le C# sous le nom de « Visual C# » : il s'agit d'un outil de développement d'applications en langage C#. La conception du C# est entre autres apparentée à celle de Java et C++. Toutefois, le C# élargit le modèle

orienté objet à travers les notions d'attributs, qui réunissent des informations sur les classes, les objets et les méthodes, et de délégués, qui pointent vers les méthodes de certaines classes. Cela permet principalement de décrire plus efficacement les erreurs lors de la compilation du code, ce qui constitue un gain de temps pour les développeurs

3. Les applications web

Une **application web** désigne un logiciel applicatif hébergé sur un serveur et accessible via un navigateur web.

Contrairement à un logiciel traditionnel, l'utilisateur d'une application web n'a pas besoin de l'installer sur son ordinateur. Il lui suffit de se connecter à l'application à l'aide de son navigateur. La tendance actuelle est d'offrir une expérience utilisateur et des fonctionnalités équivalentes aux logiciels directement installés sur les ordinateurs. Les technologies utilisées pour développer les applications web sont les mêmes que celles employées dans la création des sites internet.

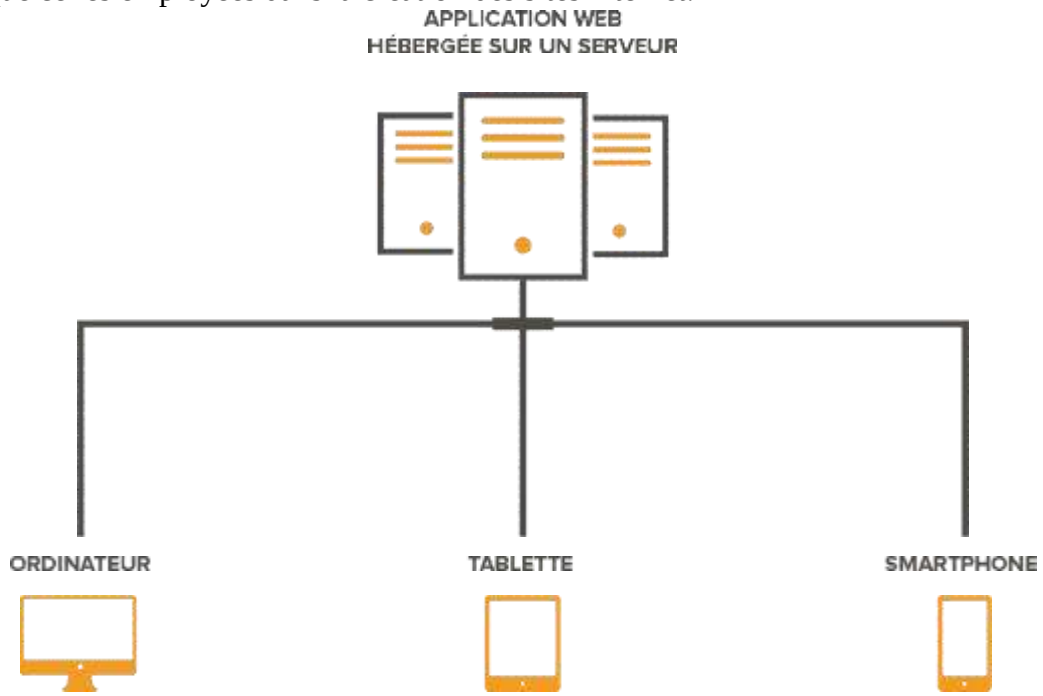


Figure 3 : Application Web

3.1 Composants principaux des applications Web

• interface utilisateur (Front End (DOM, Framework))

• Couche de demande (API Web)

• Back End (base de données, logique)

https://www.ideematic.com/dictionnaire-web/application-web

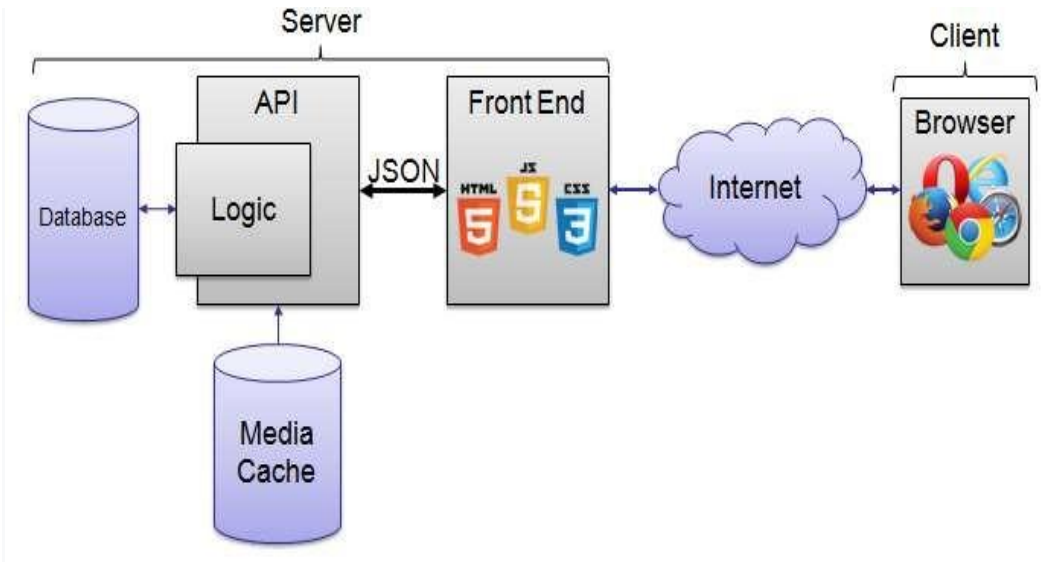


Figure 4 : Composants principaux des applications Web

3.2 Champs d'application

Habituellement, un utilisateur potentiel à l'habitude d'acheter un logiciel qu'il va installer sur son ordinateur. L'éditeur du logiciel, dans une logique cohérente de rentabilité, prévoira d'offrir de nombreuses fonctionnalités pour séduire un public le plus large possible. Mais L'utilisateur se retrouvera avec une solution dont il n'utilisera en fin de compte qu'une infime partie

L'application web s'aborde d'une manière totalement différente. En effet, les coûts de développement sur mesure pour la création d'application web étant très accessibles, l'utilisateur pourra faire appel à une agence web pour se faire développer une solution spécifiquement adaptée à ses besoins. Une fois développée, la solution offrira uniquement les fonctionnalités dont les utilisateurs auront nécessité et pourra évoluer facilement en fonction des nouveaux besoins.

Les exemples d'applications web sont bien entendu infinis. Chaque professionnel peut avoir des besoins qui lui sont spécifiques. À titre d'exemple, nous pourrions citer :

- une gestion de réservation pour un hôtel
- un outil de facturation pour un commerçant une application de gestion de dossiers patients pour un médecin etc.

3.3 Les avantages de l'application web

On peut résumer les principaux avantages d'une application web de la manière suivante :

- **Maîtrise de votre budget et diminution des coûts (**la mise de départ est inexistante, aucune mise à niveau de votre infrastructure).

- **Gain de temps (**la mise en œuvre et le déploiement sont plus rapides tout est installé, la circulation et le partage des données entre utilisateurs sont optimisés)

- **Accessibilité optimisée (**accès universel depuis n'importe quel type de poste : PC, portables, téléphone mobile, tablette, vos données sont disponibles 24h sur 24 et 7j sur7).

- **Meilleure gestion de la sécurité (**profitez des moyens des grandes infrastructures de Datacenter, accès aux données sont contrôlés par identification et certificats, Sauvegarde des automatiques).

- **Evolution et innovation continue (**vous bénéficiez toujours de la version la plus récente).

3.4Types d'applications web

a. Application web statique

La première chose à savoir sur ce type d'application est qu'elle contient peu d'informations et, en général, son contenu n'évolue pas ou très peu. Le développement d'applications web se fait habituellement en HTML et CSS. Il peut, néanmoins y avoir des objets animés tels que GIF, vidéos, etc.

La modification du contenu des applications statiques n'est pas facile. Pour ce faire, vous devez télécharger le code HTML, l'éditer, puis l'uploader de nouveau sur le serveur.

La page d'accueil d'une entreprise pourra se réaliser en application web pour afficher des informations basiques telles que on coordonnées.

b. Application web dynamique

Les applications web dynamiques sont plus complexes sur le plan technique. Elles utilisent des bases de données pour charger des informations, et le contenu est mis à jour à chaque fois que l'utilisateur se connecte à l'application. En général, elles ont un panneau d'administration depuis lequel l'administrateur peut corriger ou modifier le contenu.

Il existe de nombreux langages de programmation pour le développement d'applications web dynamiques. PHP et ASP sont les plus répandus, car ils facilitent l'organisation du contenu.

L**'**actualisation d'un web application dynamique est très simple**,** et il n'est même pas nécessaire d'entrer dans le serveur pour faire des modifications. En outre, il est possible de mettre en œuvre de nombreuses fonctionnalités telles que des forums ou des bases de données. Le design, et non seulement le contenu de l'application, peut être modifié en fonction du goût de l'administrateur.

c. Application web de type e-shop ou e-commerce

Si l'application web est un e-shop (commerce numérique), son développement sera plus complexe, car elle doit permettre les paiements électroniques par carte de crédit. Le développeur doit également créer un panel de gestion pour l'administrateur afin que ce dernier puisse mettre en vente des produits, faire des mises à jour et gérer les commandes.

d. Application web portail

Il s'agit d'une application dont la page d'accueil permet d'accéder aux différentes sections ou catégories. Son contenu peut être très varié : forums, chats, e-mail, moteurs de recherche, formulaire d'enregistrement, contenu le plus récent, etc.

e. Application web animée La technologie FLASH est indispensable pour le développement d'applications web animées. Elle sert à créer le contenu avec des effets d'animation. Permettant un design plus créatif et moderne, FLASH est l'une des technologies les plus utilisées par les designers.

L'inconvénient des applications web animées est le risque d'un référencement faible, car la technologie utilisée empêche les moteurs de recherche de lire correctement les informations.

f. Application web de type « content manager »

Pour les applications web dont le contenu doit être souvent mis à jour, vous pouvez installer un système de gestion de contenu CMS à travers lequel l'administrateur aura la possibilité d'apporter des modifications. Ces systèmes de gestion sont intuitifs et très faciles à gérer.

4. Le développement web avec reactjs

4.1. L'essence de React

Avant que ReactJS ne soit disponible pour plusieurs utilisateurs, le plus grand défi de Facebook était de créer une interface utilisateur rapide et productive. Par exemple, les développeurs avaient besoin de la possibilité d'activer l'ajout de nouvelles dans le fil d'actualités en même temps que le client modifiait son profil de données personnelles ou parlait avec ses amis en discussion.

Afin de concrétiser les choses, Facebook devait améliorer les procédures de développement et Walke était suffisamment intelligent pour utiliser JavaScript pour cela. En particulier, il a suggéré de placer XHP, le motif de marquage de Facebook, dans l'environnement de navigation JS. Cela sonnait un peu fantastique, mais il y a sept ans, la bibliothèque ReactJS basée sur la synthèse de JavaScript et de XHP a été livrée avec succès. Lorsque la solution est devenue publique, Facebook a compris que ReactJS était plus rapide que toute autre solution similaire du même profil.

En résumé, ReactJS est une bibliothèque basée sur JavaScript qui comprend la vitesse de JavaScript et le mécanisme innovant permettant de rendre les sites Web extrêmement rapides et réactifs à l'égard des requêtes des utilisateurs. En effet, cette solution a radicalement transformé la méthodologie de Facebook en développement Web. Depuis le jour où ReactJS est apparu comme logiciel open source deux ans après sa première version, sa diffusion parmi les clients était incroyablement dynamique grâce à son extraordinaire mode de codage de l'interface utilisateur.

4.2. Reactjs et le modèle MVC

Dans le modèle de conception MVC (Model-View-Controller est un modèle de conception logicielle très souvent utilisé pour la mise en œuvre d'interfaces utilisateur). React fonctionne comme le V, avec React on peut concevoir toute la partie avant de notre application.

Cela signifie qu'on peut créer facilement toute l'interface utilisateur de notre application.

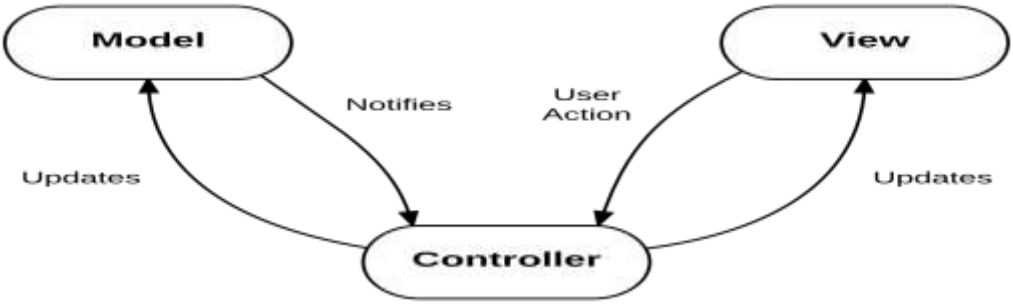


Figure 5: Model-View-Controller explanation graph

Dans une application typique, nous trouverons ces trois parties fondamentales :

- **Modèle (M) :** où les données du DOM sont stockées et traitées).
- **Voir (View) :** comme à une page qui est un seul DOM, Où les modifications apportées à la page sont rendues et affichées.
- **Contrôle (C) : Ceci** gère les entrées utilisateur et les interactions, Boutons, Formes.

Le modèle MVC, en un mot, est le suivant :

Le modèle représente les données et ne fait rien d'autre. Le modèle ne dépend PAS du contrôleur ou de la vue.

La vue affiche les données du modèle et envoie les actions de l'utilisateur (par exemple, des clics de bouton) au contrôleur. La vue peut :

Être indépendant à la fois du modèle et du contrôleur ;ou effectivement le contrôleur, et donc dépendent du modèle.

Le contrôleur fournit des données de modèle à la vue et interprète les actions de l'utilisateur telles que les clics sur les boutons. Le contrôleur dépend de la vue et du modèle. Dans certains cas, le contrôleur et la vue sont le même objet

4.3. Récupération des données

Le front-end comprend tout ce que l'utilisateur voit, y compris le design et certains langages comme HTML et CSS.
React est une bibliothèque front end qui s'exécute dans le navigateur. Comme toute autre bibliothèque front-end (jQuery, etc.), elle est servie par n'importe quel serveur Web ancien - Apache, NGINX - ou tout autre type de serveur - PHP, Rails, etc.

http://stackoverflow.com

React est une bibliothèque purement côté client constituées de fichiers JavaScript, il n'a aucune idée de la façon de récupération des données. Par conséquent, le contenu de back-end n'a aucune importance. Il n'a besoin que de données à afficher.
On peut utiliser n'importe quelle bibliothèque front-end pour récupérer des données, l'idée est de conserver les données à la racine de l'arborescence des composants et de les transmettre aux composants

qui en ont besoin. Il est plus facile de savoir où les données sont chargées quand elles sont centralisées à quelques endroits seulement, plutôt que dispersées dans l'application.

4.4. Rendu côté serveur

Le rendu coté serveur d'une page web ou server side rendering (SSR) est une technique de développement web qui consiste à créer les pages html coté serveur pour les envoyer toutes faites au navigateur. Les pages
Interagissent ensuite avec le serveur. Le rendu est plus rapide, très proche d'une application installée en locale. cette technique est beaucoup utilisée dans les applications web.

Si on souhaite prendre en charge le rendu côté serveur, que ce soit pour augmenter la vitesse de chargement des pages ou pour améliorer le référencement, React s'exécutera à la fois sur le serveur et dans le navigateur : une fois pour rendre la page serveur- côté, puis à nouveau dans le navigateur après téléchargement et affichage du rendu initial.

4.5. Reactjs et le code JSX

JSX (JavaScript XML) est une extension de syntaxe à JavaScript utilisée par React. JSX est essentiellement utilisé pour écrire des balises HTML dans JavaScript. Plus tard, le code JSX sera traduit en JavaScript normal par Babel.

En gros Pour pouvoir combiner HTML et JavaScript, React utilise JSX. En effet, JSX présente certains avantages, tels que la sécurisation du code des entrées.
JSX est une extension de syntaxe facultative à JavaScript qui facilite beaucoup l'écriture de composants. Il accepte les citations HTML et facilite le rendu des sous-composants.

En fait, il s'agit d'un ensemble de raccourcis pour écrire React.createElement avec quelques règles pour rendre votre source plus propre et plus simple.

Bien que souvent contesté, JSX peut s'avérer utile pour la création d'applications volumineuses ou de composants personnalisés, en excluant les fautes de frappe dans les grandes arborescences et en facilitant la conversion des maquettes HTML en arborescences ReactElement. En outre, il fournit aux développeurs de React des messages d'avertissement et d'erreur informatifs et permet également d'empêcher les injections de code.

5. Pour quoi développer des applications web sous React.JS ?

L'objectif principal de ReactJS est de fournir les meilleures performances possibles. Sa force provient sur ses composants individuels. Au lieu de travailler sur l'ensemble de l'application Web, ReactJS permet à un développeur de décomposer l'interface utilisateur complexe en composants plus simples. Elle permet une représentation beaucoup plus rapide du DOM qui consiste en une représentation interne des objets de React.

5.1. DOM et reactjs

DOM, abrégé en Document Object Model, est une représentation logique standard du Consortium World Wide Web de toute page Web. En termes plus simples, DOM est une structure arborescente qui contient tous les éléments et les propriétés d'un site Web en tant que nœud. DOM fournit une interface indépendante de la langue permettant d'accéder au contenu de tout élément d'une page Web et de le mettre à jour.

ReactJS ne met pas à jour le Real DOM directement mais met à jour le DOM virtuel.
Cela entraîne un avantage considérable en termes de performances pour ReactJS Virtual DOM est une représentation en mémoire de Real DOM. C'est un objet JavaScript léger qui est une copie de Real DOM.

ReactJS utilise observable pour trouver les composants modifiés. Chaque fois que la méthode setState () est appelée sur un composant, ReactJS le rend sale et le restitue.

À chaque appel de la méthode setState (), ReactJS crée le DOM virtuel complet à partir de zéro. La création d'un arbre complet est très rapide, elle n'affecte donc pas les performances.
ReactJS maintient à tout moment deux DOM virtuels, l'un avec l'état mis à jour Virtual DOM et l'autre avec l'état précédent Virtual DOM.

ReactJS à l'aide de l'algorithme diff compare à la fois le DOM virtuel pour trouver le nombre minimum d'étapes pour mettre à jour le DOM réel

5.2. Reactjs components

Un composant est l'un des composants de base de React. En termes simples, un composant est une classe ou une fonction JavaScript qui accepte éventuellement des entrées, c'est-à-dire des propriétés (props), et retourne un élément React qui décrit comment une section de l'interface utilisateur (interface utilisateur) doit apparaître.
Nous pouvons dire que chaque application qu'on développe dans React sera composée d'éléments appelés components. Les composants facilitent grandement la construction des interfaces utilisateur. On peut voir une interface utilisateur décomposée en plusieurs composants individuels appelés components, puis les travailler indépendamment et les fusionner dans un composant parent qui sera votre interface utilisateur finale.

Les composants de React renvoient essentiellement un morceau de code JSX qui indique ce qui doit être rendu à l'écran. Dans React, nous avons principalement deux types de composants :

- Composants fonctionnels

Les composants fonctionnels sont simplement des fonctions javascript. Nous pouvons créer un composant fonctionnel dans React en écrivant une fonction javascript.

- Composants de classe

Les composants de classe sont un peu plus complexes que les composants fonctionnels. Les composants fonctionnels ne connaissent pas les autres composants de programme où les composants de classe peuvent fonctionné les uns avec les autres. Nous pouvons transmettre des données d'un composant de classe à un autre composant de classe. Nous pouvons utiliser les classes javascript ES6 pour créer des composants basés sur les classes dans React.

5.3. Forces de ReactJS

- **Le modèle d'objet de document virtuel**

Le groupe de développeurs en charge de React a obtenu l'augmentation de la vitesse de mise à jour du système à l'aide du modèle d'objet de document virtualité. En termes simples, tant que d'autres référentiels s'approchent pratiquement de DOM existants, leur tout dernier concurrent, ReactJS, exploite le modèle non figuratif, le modèle virtuel. Il permet de renouveler les plus infimes alternances soumises par le client, le reste des composants de l'interface restant intact.

- **L'utilisation répétée des éléments de code**

Un autre avantage de React est la possibilité d'exploiter à plusieurs reprises les éléments du code à n'importe quel stade et à tout moment. Cette fonctionnalité permet de réduire le temps requis pour le codage.

Fondamentalement, la réutilisation des composants disponibles est une pratique efficace qui a été prouvée par les concepteurs Web qui réalisent des gains de temps considérables en appliquant les mêmes composants et modules dans divers projets. Cependant, en termes de codage, la situation est plus complexe. Les mises à jour du système peuvent être très difficiles, car chaque petite transformation peut influer sur le fonctionnement d'autres éléments.

- **Le flux de données à orientation unique dans ReactJS** produit un code substantiel ReactJS assure un accès direct aux éléments et exploite une méthode d'association de données descendante pour empêcher la transformation des configurations de niveau supérieur lorsque les structures de niveau inférieur sont modifiées. Cela permet d'obtenir le code équilibré.

Les arrangements plus complexes du type observation de la configuration JS sont caractérisés par une faiblesse tangible : les schémas de flux de données. En fait, les éléments de bas niveau peuvent influencer ceux de niveau supérieur lorsqu'ils sont modifiés dans les structures de modèle de vue JS. Dans le cas de ReactJS, le fournisseur a éliminé ces négatifs en le laissant comme structure d'affichage.

Au lieu d'appliquer une association de données claire, ReactJS préfère utiliser un modèle à orientation unique, un flux de données descendant. C'est pourquoi les composants de niveau inférieur n'affectent pas les composants de niveau inférieur. Par conséquent, tout ce que le codeur doit faire est de changer le statut de l'élément et d'ajouter les fragments mis à jour. En conséquence, seuls les composants autorisés seront modifiés

- **Il joue bien avec votre code existant**

Les ingénieurs de Facebook ont conçu React pour s'intégrer parfaitement aux bases de code existantes. Ils ont investi des années dans la création du site Facebook, de sorte qu'une réécriture complète, juste pour utiliser React, ne serait pas justifiable.

Il est très facile d'utiliser React avec jQuery, Angular, Backbone ou n'importe quel autre framework JavaScript.

5.4. Avantages de reactjs

Lorsqu'on crée une application React, elle est essentiellement composée de deux parties : Les composants, qui contiennent notre code HTML et ce que nous souhaitez que l'utilisateur voie, et un document HTML dans lequel tous nous composants seront restitués.

- React peut rendre les sites facilement référençables par Google grâce à l'utilisation d'un serveur Node. En effet, le code peut être généré côté client et serveur à la différence des autres frameworks traditionnels.
- Rapiditépour les clients il permet d'avoir un site réactif, rapide et évolutif. Pour les développeurs il propose l'intérêt d'être modulable. Lorsqu'une modification a besoin d'être faite, React calcule les changements à opérer et rafraichit uniquement les parties impactées. Les opérations coûteuses sont ainsi évitées.
- Le découpagereact est basé sur des composants réutilisables, testables et combinables. Lorsqu'un problème est identifié dans un composant, il est possible de le corriger pour de bon, car lorsque vous modifiez un composant, les effets de bord sont limités à ce composant uniquement.
- Le langageReact ne se limite pas à la création d'application web, il peut également être utilisé pour beaucoup d'autres choses. Voici quelques exemples :

- react-native pour la création d'application mobile ;
- react-sketch pour la création de maquettes sketch ;
- react-word pour la création de documents word;
- react-pdf pour la création de documents PDF,

Lorsque votre application React est chargée dans le navigateur, tous les composants que n'a créés sont rendus. Mais disons qu'un de vos composants a subi une modification, peut-être à cause de l'interaction de l'utilisateur, ce composant est alors restitué pour indiquer le nouveau changement. Cela signifie que les seules parties de l'application qui sont rechargées sont celles dont l'état change.

Toute la puissance de HTML, CSS et Javascript au sein du composant : lorsqu'on crée un composant React, on peut non seulement utiliser HTML et CSS comme on fait habituellement, mais on peut également intégrer le langage Javascript de manière très agréable.

ReactJS a fourni la solution recherchée par les développeurs. Il utilise JSX (une syntaxe unique qui autorise les guillemets HTML ainsi que l'application de syntaxe de balise HTML pour le rendu de sous-composants spécifiques). Ceci est très utile pour promouvoir la construction de codes lisibles par machine et la composition simultanée de composants dans un fichier unique vérifiable.

6. Comment développer des applications sous React .JS

ReactJS fonctionne en créant des "components", correspondants à un composant dans une page HTML. L'intérêt de ReactJS est que ses composants correspondent à un élément de la page, avec son état propre et ses actions associées. Ces éléments ont leur propre état, leurs propres données et sont indépendants les uns des autres.

Voici différentes méthodes et champs possibles pour créer un composant :

- **props** : ce sont les propriétés qui sont données en paramètre du composant. IL s'agira souvent d'une configuration.
- **state** : contient l'état interne du composant. Celui-ci peut être la liste des éléments à afficher, une valeur ou une configuration.
- **getInitialState** : méthode permettant d'initialiser le "state" interne du composant. La méthode retourne un objet qui sera ensuite mappé sur this.state.
- **setState** : méthode permettant simplement de mettre à jour le "state". Elle est principalement appelée de l'extérieur pour mettre à jour les données.
- **componantDidMount** : appelée lors de la création du composant, elle permet d'initialiser des méthodes et données.
- **componantWillUnmount** : Appeler lors de la destruction du composant. Peut être utilisé pour transmettre des données lors de cette phase.
- **render** : se charge de l'affichage du composant (comme vu auparavant

7. Reactjs et les technologies utilisées

En fait, ReactJS est une bibliothèque de vues qui s'exécute sur le navigateur. Son objectif principal est de restituer tout ce que l'utilisateur voit sur son appareil

Ceci est notre pile frontend, de même on aura également une pile principale, le serveur qui exécute notre application. Et la base de données fait partie de la pile principale. Sur le backend, on a choisir lelangage deJavascript express.

Node.js est un environnement d'exécution qui permet aux utilisateurs de choisir le mode d'utilisation, qu'il soit frontend ou backend. Un langage commun peut être utilisé comme backend et front-end, cet environnement est entièrement basé sur le moteur JavaScript V8.

Il permet d'exécuter facilement des documents JavaScript qui ne sont même pas exécutés dans un navigateur Web, il peut également être utilisé comme environnement backend, alors qu'il offre également une fonctionnalité permettant d'automatiser les tâches de routine, notamment l'exécution de tests de code, de kits d'outils, etc. même l'utiliser comme environnement frontend, ce qui améliorera le développement de logiciels.

Et à partir de notre application serveur, nous nous connectons à notre base de données mongodb.

Notre application React se connecte à notre application serveur généralement avec des requêtes et pour cela on a choisis graphql, accompagné de Appollo server

Et notre application serveur comprendra ce que est demandé, fera une demande à la base de données et renverra des données à l'application React.

Conclusion

Dans ce chapitre nous avons recensé et présenté quelques aspects essentiels de développement web et de environnement de développement reactjs et ses différents outils et avantages avec le choix des technologies utiliser pour la conception.

Le chapitre suivant et consacrés à l'étude de cas relative à la modélisation et conception pour une application web pour la gestion de RDV chez un professionnel de sente.

Chapitre 3 : étude de cas

Introduction

Soigner plus, plus de patients, plus vite, plus efficacement, mais avec moins : moins de Ressources, de personnel, de budget ; voilà le challenge auquel se trouve confronté aujourd'hui le monde de la santé en général.

L'application de prise de rendez-vous en ligne est l'une des applications de santé grand public qui gagnent en popularité. L'application permettra notamment à un médecin, qu'il utilise ou non un logiciel de gestion d'agenda, d'offrir des plages de rendez-vous à ses Patients ou, s'il le souhaite, à toute autre Patient, dans la clinique où il pratique.

L'étude d'un projet est une démarche stratégique qui va nous permettre d'avoir une vision globale sur ce dernier visant ainsi à bien organiser le bon déroulement du projet. Cette étude fera donc l'objet du ce chapitre qui sera consacré à la présentation de notre projet et la spécification des besoins fonctionnels et non fonctionnels de notre application.

1. Architecture du système

La figure ci-dessous montre l'architecture générale de notre système.

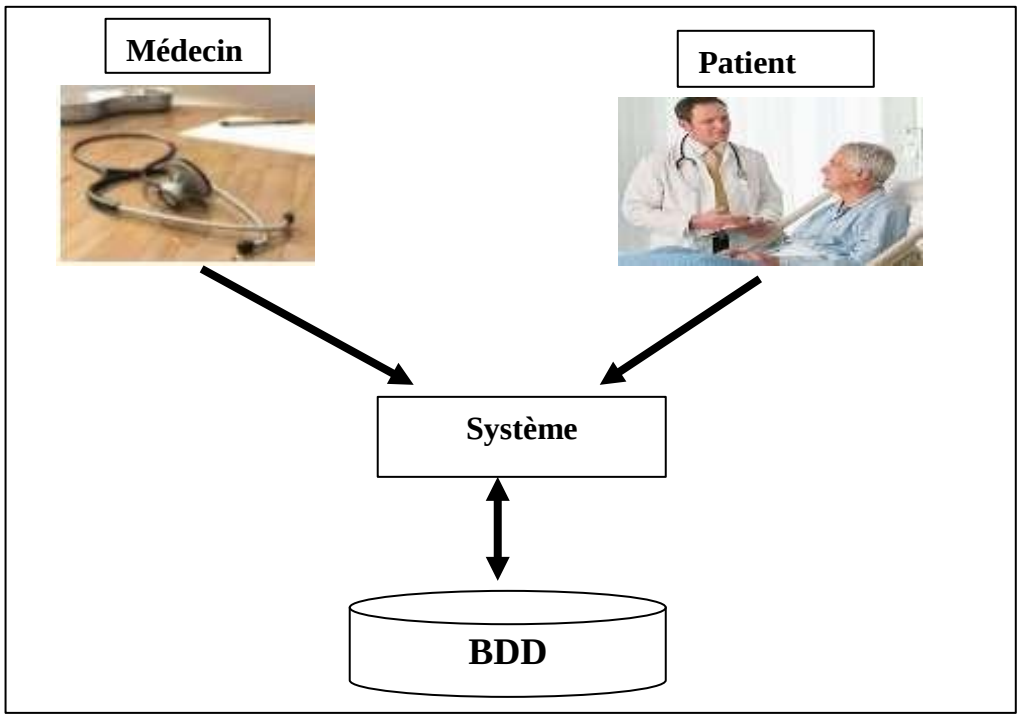


Figure 6 : L'architecture générale du système.

2. Définitions

Patient : c'est la personne qui subit un traitement, diagnostic. Dans le domaine de la médicine, un patient est une personne recevant une attention médicale ou à qui est prodigué un soin. Il bénéficie d'examens médicaux, de traitements prodigues par un médecin ou autres professionnels de santé pour faire face à une maladie ou à des blessures.

Médecin : C'est la personne qui diagnostique et en traitant les maladies. Un médecin est un professionnel de la sante titulaire du diplôme de docteur en médecine, il soigne les maladies, pathologie et blessures, il travaille généralement au sein d'une équipe de professionnels de la sante comme le psychologue, le pharmacien, l'infirmière ou le chirurgien-dentiste

La relation médecin/malade : La relation médecin-malade est une relation faite d'attentes et d'espérances mutuelles.

Le malade attend un soulagement et si possible la guérison, mais il faut bien admettre que le médecin attend aussi une reconnaissance de la part de son malade, une vérification de son pouvoir soignant. La relation médecin-malade est une relation inégale. Elle a pour point de départ la demande d'un sujet souffrant adressée à un sujet disposant d'un savoir.

La relation médecin-malade est une relation paradoxale. Elle a le corps pour objet mais passe le plus souvent par la parole, ce qui peut être source de malentendus et d'incompréhension.

Enfin, la relation médecin-malade est marquée par l'idéalisation : le médecin idéal est pour le patient celui qui pourra être à la hauteur de ses multiples espérances. Le patient idéal est, pour le praticien, celui qui lui permettra au mieux de satisfaire sa vocation ; c'est à dire à la fois ses attentes conscientes et ses désirs inconscients.

Une relation médecin-malade harmonieuse permet une démarche diagnostique efficace, une amélioration de la qualité de vie par la prise en compte du point de vue du malade et une bonne observance thérapeutique.

Les réflexions sur la relation médecin-malade se sont développées à partir, entre autres, de la psychanalyse, de la psychologie sociale, des théories de la communication. La psychanalyse a, par exemple, montré l'importance de la prise en compte de l'influence du malade sur les sentiments inconscients du médecin. La notion de contre-transfert désigne les mouvements affectifs du médecin en réaction à ceux de son patient et en relation avec le vécu de son histoire familiale et personnelle.

La notion de transfert, quant à elle, se situe du côté du patient. Elle consiste en la répétition de modalités relationnelles vécues dans l'enfance, conduisant le malade à imposer certains styles de relation à son médecin. Les médecins savent bien qu'écouter le malade et l'entourage affectif de ce dernier est un aspect fondamental de la relation médecin-malade. Ils ont commencé à apprendre l'importance de s'écouter eux-mêmes, de reconnaître les sentiments induits par le malade, et qui pourraient entraver les démarches diagnostique et thérapeutique [6].

.

3. La prise de rendez-vous en ligne

3.1 Qu'est-ce qu'une solution de prise de rendez-vous en ligne ?

Une solution de prise de rendez-vous en ligne est une application qui permet aux patients de prendre ou de modifier un rendez-vous sans devoir interagir avec un membre du personnel administratif. La plupart des systèmes envoient la confirmation des rendez-vous et des rappels par voie électronique.

Les applications de prise de rendez-vous peuvent aussi gérer des périodes de rendez-vous ou des périodes de consultations de rendez-vous. Elles peuvent également être configurés pour que le médecin puisse réserver des plages pour la consultation de rendez- vous à ses propres patients.

3.2 Les caractéristiques des systèmes de rendez-vous en ligne

Les solutions ne possèdent pas toutes les mêmes fonctionnalités. Toutefois, elles partagent fréquemment certaines caractéristiques communes :

- la prise de rendez-vous en tout temps ;
- un accès sécurisé ;
- des rappels automatisés du rendez-vous ;
- des identifiants individuels pour le personnel ;
- des règles permettant de personnaliser la prise de rendez-vous.

Certaines solutions offrent également des fonctionnalités supplémentaires, par exemple :

- l'inscription du rendez-vous dans un calendrier personnel du patient ;
- des avis d'annulation et des options de modification automatisés ;
- l'envoi de rappels par courriel ou par message texte (SMS) ;
- la création de liste d'attente pour remplacer rapidement les annulations
- une application compatible avec les appareils mobiles.

3.3 Les enjeux juridiques associés

L'utilisation d'une solution de rendez-vous en ligne, comme toute autre technologie de l'information et des communications, comporte généralement des enjeux juridiques liés à la protection de la confidentialité, à la sécurité et à l'intégrité des renseignements communiqués. En tout temps, sauf les rares exceptions prévues au Code de déontologie des médecins, le médecin est tenu de respecter le secret professionnel et de protéger les renseignements personnels de son patient. Il a de plus l'obligation déontologique d'assurer la protection de l'intégrité des renseignements de ses patients. Le médecin doit donc prendre les moyens lui permettant de respecter ses obligations déontologiques et légales lorsqu'il utilise un système de rendez-vous en ligne [8].

3.4 Comment choisir une solution de prise de rendez-vous en ligne ?

Ce mémoire a pour objectif de nous guider dans la réalisation d'une solution de prise de rendez-vous en ligne. Un système de prise de rendez-vous en ligne permet aux patients facilement de prendre un rendez-vous. Ils peuvent choisir la date et l'heure qui leur convient, recevoir une notification de confirmation. L'adoption de la prise de rendez-vous en ligne est un sujet d'actualité et prend de plus en plus d'ampleur. L'utilisation d'un système de prise de rendez-vous en ligne, identifie les bénéfices suivants :

3.5 Les avantages d'aller en ligne

Permet les avantages d'aller en ligne on sites :

- **Réduction du temps consacré à la prise de rendez-vous :** moins d'appels entrants, de confirmations et de rappels téléphoniques à gérer contribuent à une gestion plus efficace du temps pour le personnel. Des rapports internationaux démontrent d'ailleurs que l'implantation de la technologie entraîne une réduction de 80% du temps requis pour prendre un rendez-vous et peut se traduire par des économies de 220$ par semaine, par médecin.

- **Augmentation du taux d'assiduité des patients :** Les rappels automatisés de la prise de rendez-vous enligne « ont grandement contribué à réduire le nombre de rendez-vous manqués et les annulations de dernière minute ». La prise de rendez-vous en ligne permet surtout un meilleur accès à son médecin de famille et son équipe.
- **Augmentation de la satisfaction du personnel :** Une solution en ligne élimine les tâches complexes et les délais reliés à la priorisation des patients au moment de la prise de rendez-vous.
- Effectivement, tous les utilisateurs d'une solution de prise de rendez-vous en ligne interrogés ont observé une amélioration au niveau de la satisfaction de leur personnel.
- **Augmentation de la satisfaction des patients** : Avoir un accès 24/7 et offre la garantie qu'un rendez-vous peut être pris au moment opportun. La plupart des patients préfèrent utiliser la prise de rendez-vous en ligne plutôt que d'appeler et risquer de patienter en ligne.

3.6 Les éléments clés à rechercher dans une solution de rendez-vous patients

- **Automatisation des confirmations de rendez-vous :** La solution de prise de rendez-vous en ligne devrait pouvoir être configurée de façon à envoyer automatiquement une confirmation du rendez vous au patient.
- **Synchronisation des horaires des patients et des médecins :** Les systèmes les plus efficaces peuvent synchroniser tous les horaires de médecin en un seul endroit. Certaines solutions offrent un portail de rendez-vous pour les patients, mais lorsque les membres du personnel reçoivent une demande de rendez-vous, ils doivent l'inscrire manuellement dans le calendrier du médecin.
- **Type de rendez-vous multiple :** il y a souvent plusieurs services qui sont offerts aux patients. Le système devrait permettre le choix parmi plusieurs types de rendez-vous, d'une durée prédéterminée, avec différents professionnels. La gestion du sans rendez-vous doit être également possible.
- **Règles de planification personnalisables :** La solution devrait être personnalisable les besoins en termes de création et de gestion d'horaires. Les médecins produisent souvent leurs horaires plusieurs mois à l'avance, mais on peut choisir d'offrir des plages horaires seulement quelques semaines à l'avance aux patients. La solution doit permettre la gestion des rendez-vous en accès adapté tel que recommandé par la loi local.
- **Confidentiel et sécuritaire** : Les avantages de la prise de rendez-vous en ligne sont évidents, mais il faut tenir compte des risques de sécurité et de confidentialité. Assurez que l'entreprise qu'on a choisissez respecte les normes de sécurité les plus élevées. Cela inclut le cryptage des données à tous les niveaux conformément aux normes internationales les plus strictes, un serveur privé sécurisé avec système de sauvegarde, des protocoles de sécurité et des procédures en place pour prévenir les violations de sécurité, le tout conforme aux réglementations [10].

4. Présentation du Projet

Etant donnée l'émergence de technologie web et le taux d'acquisition croissant des Smartphones et tablettes chez le grand public, beaucoup d'applications ont été développées dans divers domaines parmi ces domaines, nous trouvons les domaines de la santé.

Durant ce projet nous allons faire la conception et le développement d'une application web qui s'appelle **AppMeq (**appointment medical easy and quick**), (**rendez-vous médical facile et rapide**)** permet de trouver un médecin en cas de besoin et de réserver un rendez-vous

L'origine de ce sujet était une simple idée pour fournir des informations concises et pertinentes sur les médecins à Bejaia facilement accessibles en cas de besoin. Au fur et à mesure cette idée a évolué pour concevoir un système de gestion des rendez-vous médicaux.

Ce type d'application métier s'avère très utile non seulement afin de subvenir aux besoins des patients mais il peut aussi représenter un réel avantage pour les médecins. Il s'agit de rapprocher les médecins de leur patients et de le rendre plus disponible et surtout accessible en cas de besoin.

4.1 Spécification des besoins

Besoins fonctionnels :

Dans cette partie nous détaillons l'ensemble des fonctionnalités qu'**AppMeq** doit offrir aux utilisateurs. En effet, le système à réaliser doit répondre aux besoins fonctionnels suivants [9] :

a. Authentification : Chaque utilisateur (médecin, patient) possède un login et un mot de passe spécifique qui lui permet de vérifier son identité, afin d'autoriser l'accès de cette entité a des ressources en toutes sécurité

b. Gestion des rendez-vous : Le patient a la possibilité de prendre un rendez-vous et il peut consulter l'état de sa demande de rendez-vous (acceptée ou refusée) de plus le médecin peut accepter ou refuser une demande de rendez-vous selon sa disponibilité.

c. Chercher un médecin par nom ou par spécialité

d. Voir historique des rendez-vous

Le médecin peut consulter l'historique des demandes qu'il a acceptées et refusées

e. L'ajout de cabinet : C'est le professionnel de santé n'apparait pas dans la recherche, il peut ajouter ses informations.

f. Vérification du médecin : Un médecin doit envoyer une image de sa carte d'identité ou une pièce qui vérifie qu'il est un médecin.

Besoins non fonctionnels

Les besoins non fonctionnels sont importants car ils agissent de façon indirecte sur le résultat et sur la performance du système, besoins non fonctionnels décrivent toutes les contraintes techniques, ergonomiques et esthétiques auxquelles est soumis le système pour sa réalisation et pour son bon fonctionnement. Et ce qui concerne notre application, nous avons dégagé les besoins suivants :

a. La disponibilité : l'application doit être disponible pour être utilisé par n'importe quel utilisateur.

b. La sécurité : de l'accès aux informations critiques : nous devons prendre en considération la confidentialité des données de clients surtout au niveau de l'authentification. Pour cela nous devons restreindre l'accès à ces informations.

c. La fiabilité : les données fournies par l'application doivent être fiables.

4.2 Identification 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. Les principaux profils qui auront à utiliser le Système sont les suivants :

Patient : s'authentifier, réserver un RDV, annuler un RDV **Médecin** : accepter ou refuser un rendez-vous.

Les acteurs et les cas d'utilisation sont résumés dans le tableau suivant :

Cas d'utilisations	Acteurs
Gréer les rendez-vous	Médecin
Gérer horaire de travail	
Cherche médecin	Patient
Consulter historique des rendez-vous	
Prendre un rendez vous	
Gérer le profil	Médecin, patient

Tableau 1 : identification des acteurs et des cas d'utilisations

5. cas d'utilisations

Un cas d'utilisation est utilisé pour définir le comportement d'un système ou la sémantique de toute autre entité sans révéler sa structure interne. Chaque cas d'utilisation spécifie une séquence d'action, y compris des variantes, que l'entité réalise, en interagissant avec les acteurs de l'entité. La responsabilité d'un cas d'utilisation est de spécifier un ensemble d'instances, où une instance de cas d'utilisation représente une séquence d'actions que le système réalise et qui fournit un résultat observable par l'acteur.

5.1 Cas d'utilisation de patient des avis d'annulation et des options de modification automatisés

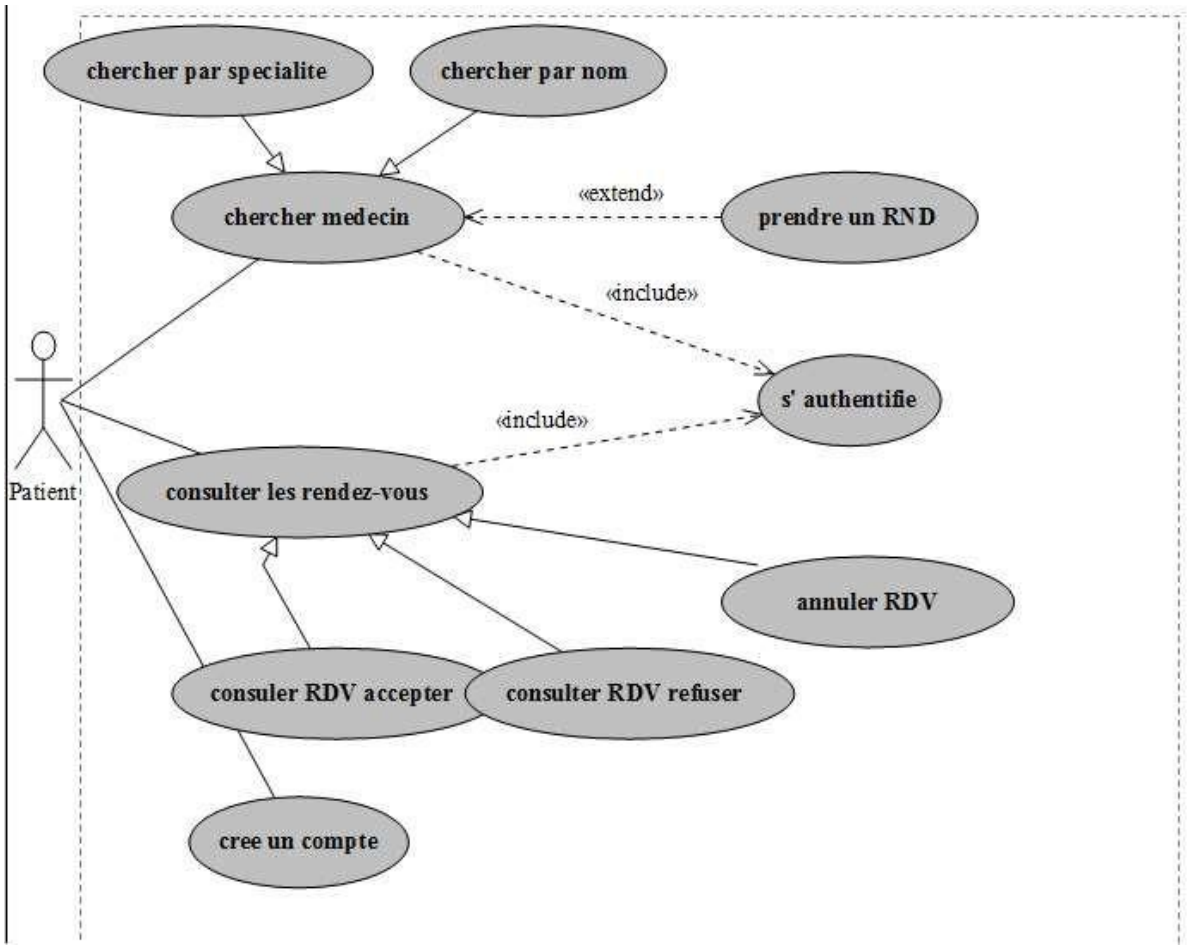


Figure7 : diagramme de cas d'utilisation de patient

- Le patient est l'utilisateur de **AppMeq**, pour pouvoir accéder aux différents fonctionnalités de l'application le patient doit se connecter s il possède déjà un compte sinon il doit créer un compte.
- Un patient peut chercher un médecin directement à partir son nom ou par spécialité
- Une fois le médecin été sélectionné le patient peut prendre un rendez-vous selon la disponibilité du médecin

Cas d'utilisation	Prendre rendez-vous
Résume	Ce cas permet au patient de prendre un rendez-vous avec un médecin
Acteur	Patient
Scenario nominale	1. Le patient sélectionne la date du rendez-vous 2. Le patient sélectionne l'heure du rendez-vous 3. Le patient saisis son nom d'utilisateur et le mot de passe 4. Le rendez-vous est enregistré dans la base de données
Scenario d'erreur	1. Le patient décide de valider sans sélection de la date 2. Le patient décide de valider son sélection de l'heure 3. Le patient décide de valider son saisir le nom d'utilisateur et le mot de passe

Pré condition	Choisir un médecin Etre authentifie
Post condition	Le rendez-vous enregistré

- Apres avoir pris un rendez-vous le patient consulter la liste de ses rendez-vous (accepte, refuse et en
- attente). Tant que le rendez-vous est encore en attente, le patient a la possibilité de l'annuler
- Le patient peut aussi changer les informations de son compte Si un rendez-vous a été accepté par le médecin et le patient rate la consultation plusieurs fois son compte sera supprimé Description **Cas d'utilisation pris de rendez-vous**

Tableau 2 : Description Cas d'utilisation pris de rendez-vous

Description Cas d'utilisation création du compte patient

Cas d'utilisation	création du compte patient
Résume	Ce cas permet au patient de créer un compte
Acteur	Patient
Scenario nominale	1. Le patient saisie les données 2. Le système reçoit les données saisies 3. Le système enregistre les informations dans la base de données
Scenario d'erreur	2. le patient saisit des données erronées 5. le système envoie un message d'erreur au patient 6. le patient sera redirige a l'interface de création du compte
Pré condition	-
Post condition	Compte crée

Tableau 3 : Description Cas d'utilisation création du compte patient

Description Cas d'utilisation recherche de médecin

Cas d'utilisation	recherche de médecin
Résume	Ce cas permet au patient de chercher un médecin
Acteur	Patient

Scenario nominale	1. Le patient choisit une spécialité 2. Le système cherche tous les médecins avec la spécialité choisie 3. Le patient choisit un médecin de la liste 4. Le système affiche le profil du médecin sélectionne 5. Le patient saisit le nom du médecin 6. Le système cherche tous les médecins avec ce nom 7. Le patient choisit un médecin de la liste 8. Le système affiche le profil de médecin
Scenario d'erreur	1. Le patient ne choisit aucune spécialité 2. Le patient ne choisit aucun médecin 3. Le patient saisie un nom erroné 4. Le patient ne saisit aucun nom 5. Le patient ne choisit aucun médecin
Pré condition	Patient authentifié
Post condition	Profil de médecin affiché

Tableau4 : Description Cas d'utilisation recherche de médecin

5.2 Cas d'utilisation de médecin

- Le médecin dans le système peut gérer ses disponibilités il peut ajouter ou modifier ses jours et heure de travail
- De plus le médecin peut accepter ou refuser un rendez-vous
- Si un médecin n'apparait pas il peut ajouter son nom.

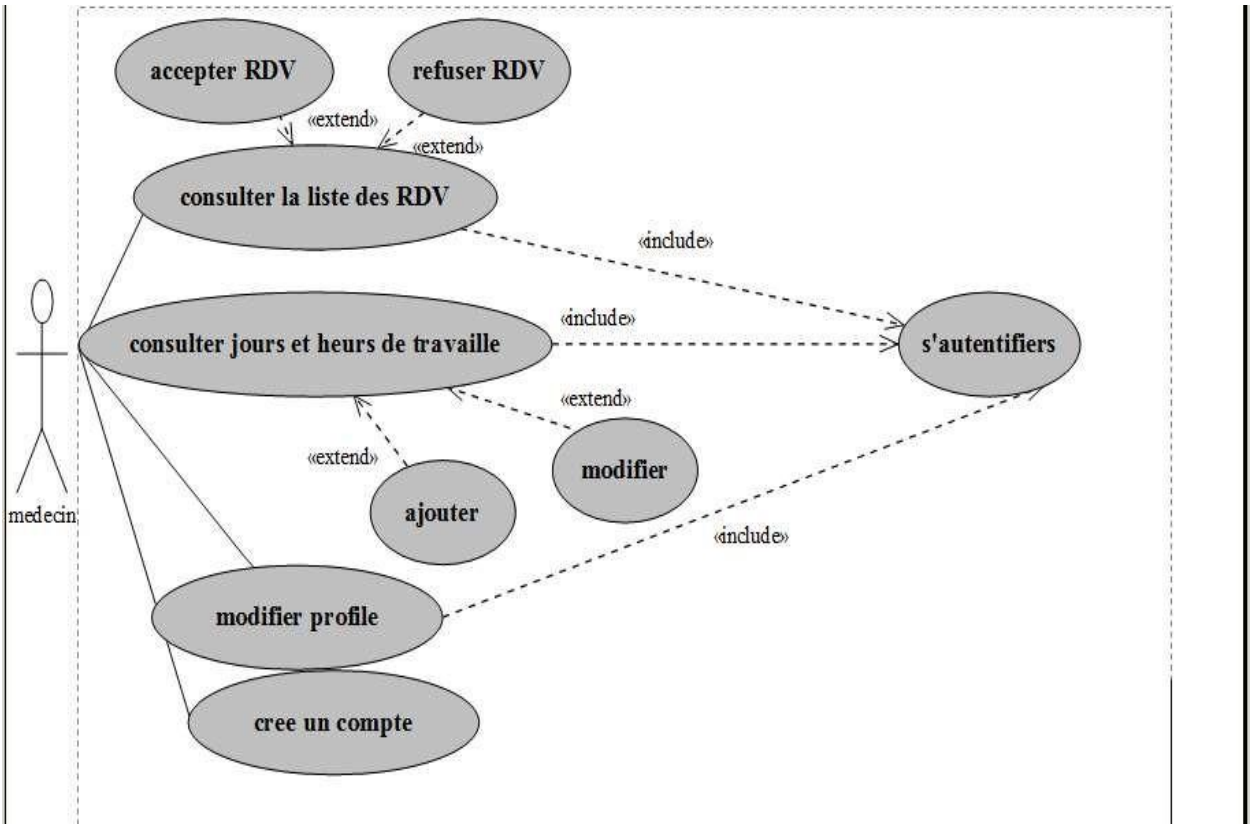


Figure 8 : diagramme de cas d'utilisation médecin

6. Diagramme de séquence

Le diagramme de séquencés fait partie des diagrammes dynamique et plus précisément des diagrammes d'interactions, il permet de représenter des échanges entre les différents objets et acteurs du système en fonction du temps

6.1. Diagramme de séquences Rendez-vous

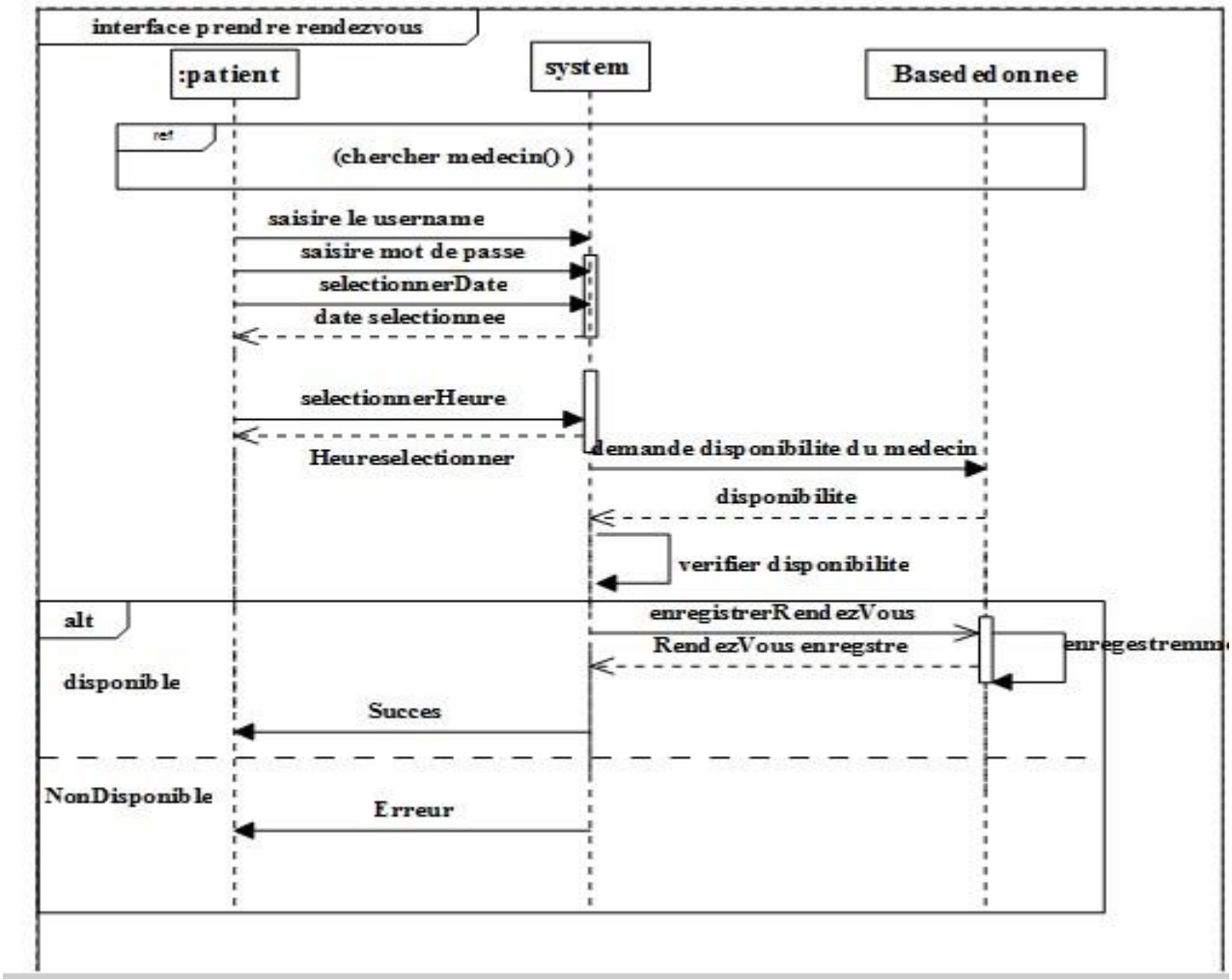


Figure 9 : diagramme de séquence prendre un rendez-vous

Pour prendre un rendez-vous le patient sélectionne la date et l'heure du rendez-vous souhaité Apres la vérification de la disponibilité du médecin le rendez-vous est enregistré dans la base de données et un message de confirmation s'affiche au patient.

Si la date et/ou heure sélectionnée ne correspondent pas à la disponibilité du médecin, le système envoie un message d'erreur au patient pour lui informer que son rendez-vous n'a pas été enregistré et lui demande de choisir une autre date.

6.2 Diagramme de séquence Authentification

Voilà ci-dessous le diagramme d'authentification d'un patient :

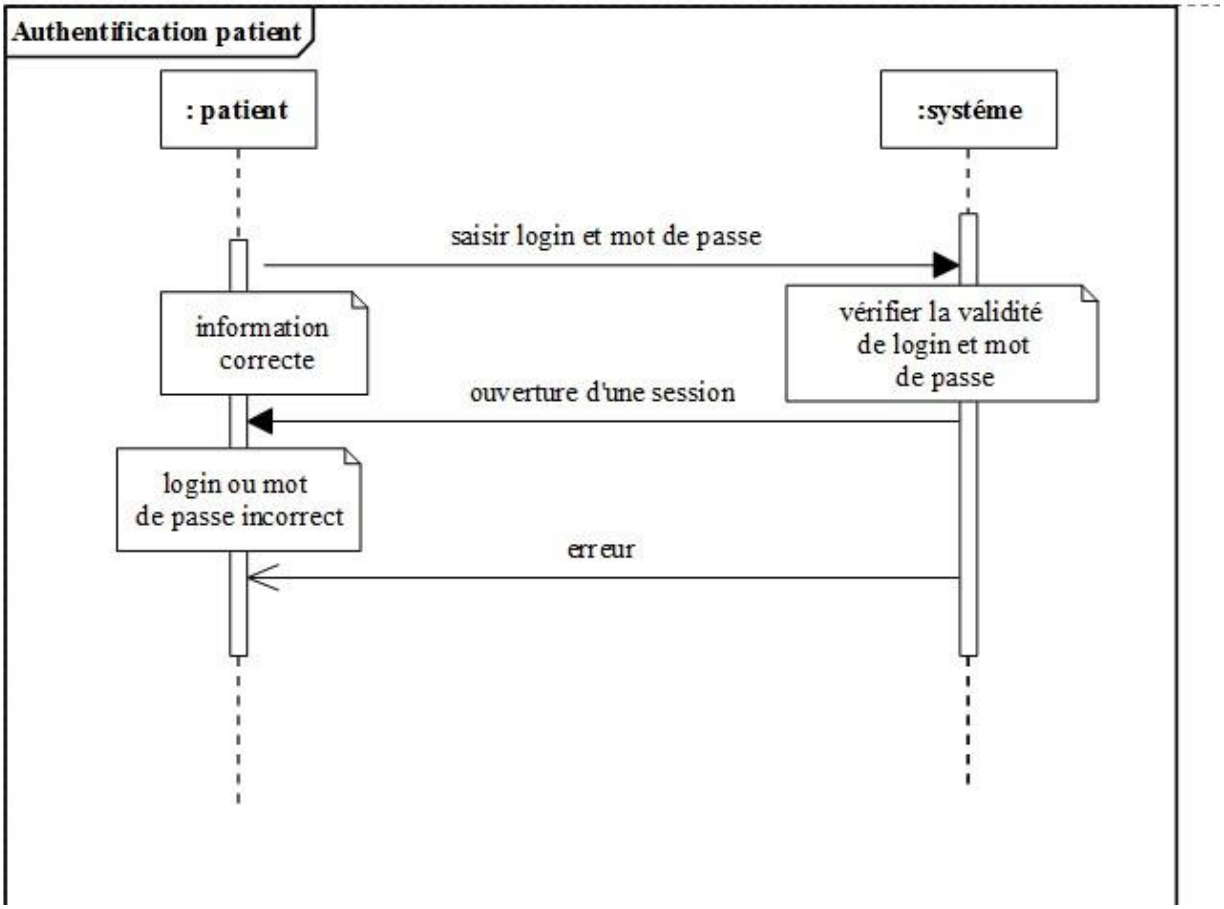


Figure10 : Diagramme de séquence « Authentification »

6.3Diagramme de séquence Recherche médecin

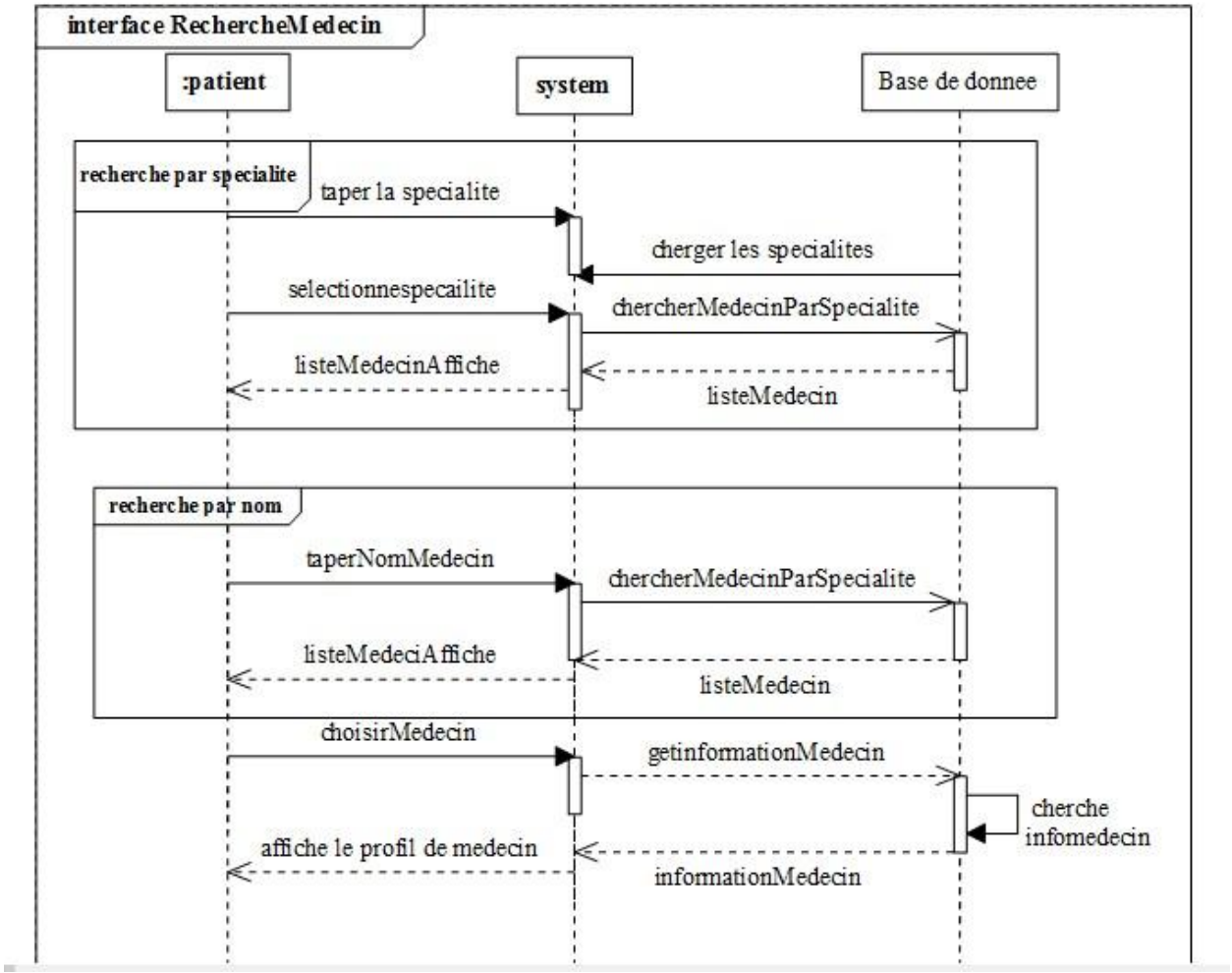


Figure 11: diagramme de séquence recherche un médecin

Le patient peut chercher un médecin par son nom ou par sa spécialité.

Pour chercher un médecin par spécialité, il suffit de choisir une spécialité existante dans la base de données et le système affiche la liste des médecins de cette spécialité

En plus, le patient a la possibilité de chercher un médecin par son nom. Pour cela, il suffit de saisir un nom et le système affiche la liste des médecins avec ce nom

6.4 Diagramme de séquence Création du compte

Pour accéder à l'application, le patient doit avoir un compte

Pour crée un compte, le patient doit saisir les informations demandées correctement puis le système vérifie ces informations et si na pas des erreurs le compte sera créé et les informations seront enregistrées dans la base de données

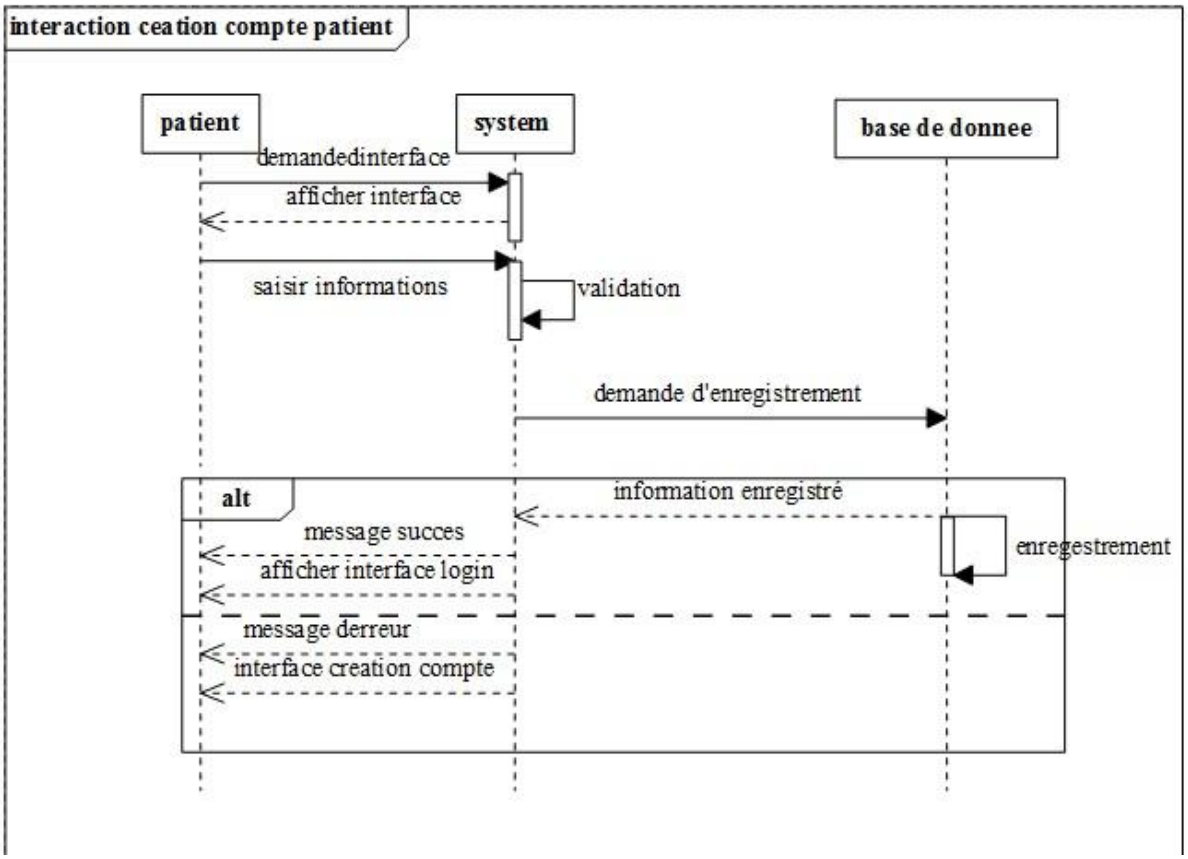


Figure 12 : diagramme de séquence création de compte patient

7. Diagramme de classes

Le diagramme de classe est un schéma utilise pour présenter les classes et les interfaces des systèmes ainsi que les différentes relations entre celles-ci. Ce diagramme fait partie statique d UML car il fait abstraction des aspects temporels et dynamiques

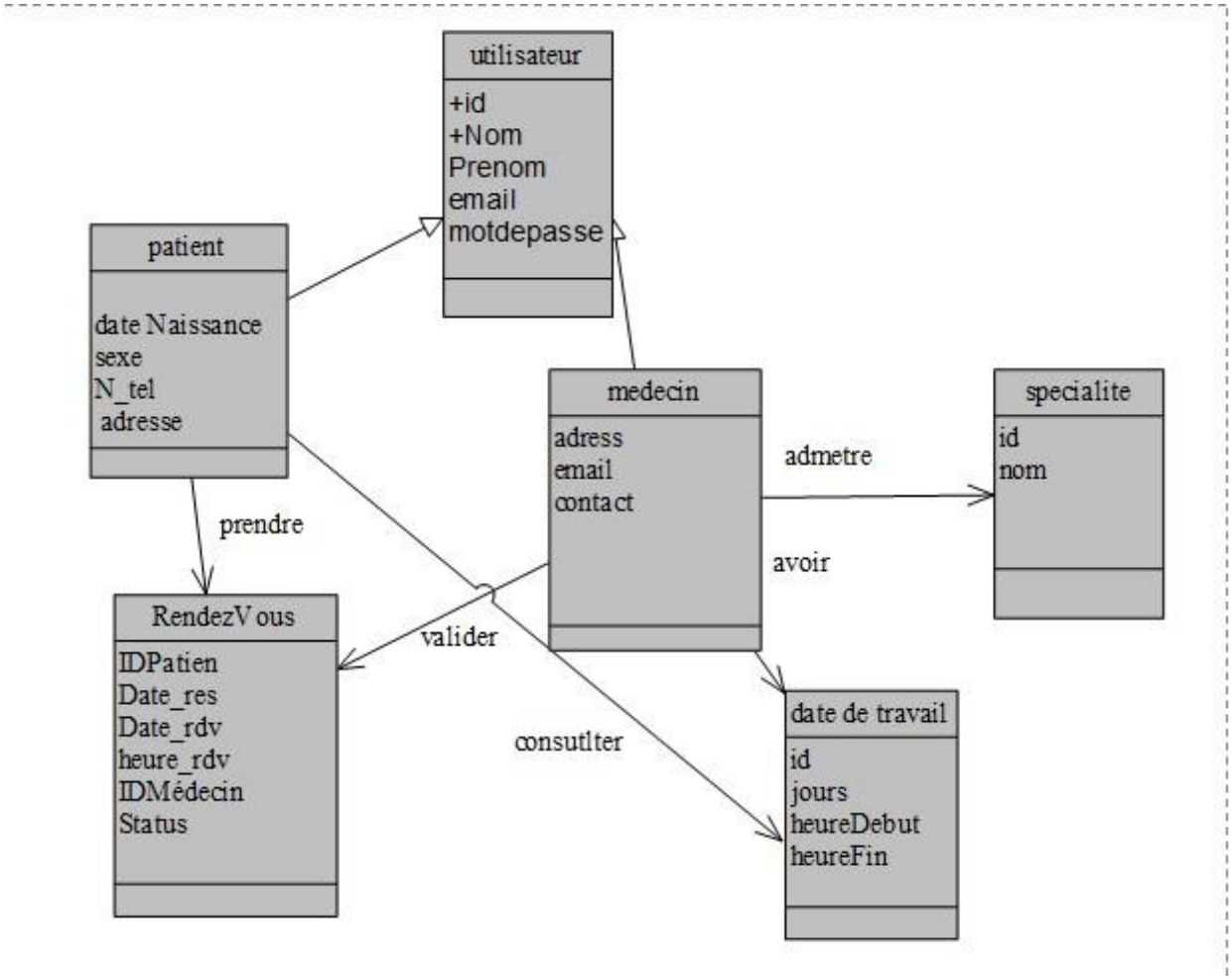


Figure 13 : diagramme de classe

Dans notre diagramme de classe on à deux classes, médecin et patient qui hérite d'une classe mère nomme utilisateur.

Un patient peut prendre un rendez-vous avec un médecin et ce dernier peut accepter ou refuser le rendez-vous selon son horaire de travail Un médecin admet une spécialité.

7.1 Dictionnaire de class et des attributs

Classe Méthode	**Attribut**		**Méthode**
	Champ	**Type**	
Patient	Date_nais Email N_tel adresse	Date Caractère Numérique Caractère	S'authentifier Réserver Modifier Annuler
utilisateur	id Nom Prenom mot de passe	Numérique Caractère Caractère Caractère Caractère	
Rendez-vous	ID_Rdv	Numérique	

	IDPatien Date_resDate_rdvheure_rdv IDMédecin Status	Numérique Date Date Caractère Numérique Caractère	
Médecin	adresse email contact	Caractère Caractère Caractère	S'authentifier consulter Annuler
date de travail	id joursheureDebut heureFin	Numérique Caractère Date Date	Mettre à jour
spécialité	id nom	Numérique Caractère	

Tableau 5 : Dictionnaire de class et des attributs.

Conclusion

Dans ce chapitre nous Avons présenté une étude qui porte l'analyse des besoins et la modélisation de notre application. La modélisation est base sur trois types de diagrammes : diagramme de classe diagramme de cas d'utilisation, et le diagramme de séquence.

Cette étape de modélisation nous a permis de générer les tables de la base de données et avoir une vue générale du comportement théorique des fonctionnalités offertes par notre plateforme cette base théorique va nous guider dans la prochaine étape de développement qui sera décrit dans le chapitre suivant.

Chapitre 4 :
réalisation et déploiement

Introduction

Nous arrivons dans ce chapitre a la description de l'aspect pratique de notre travail .dans la description de notre application qui suivra ou mettrons l'accent sur le cote visuel (les interfaces), afin de montrer sa facilité d'utilisation qui nous a été un objectif principal .en effet, nous avons essayé de concevoir une interface intuitive et pratique. Nous décrivons aussi dans ce chapitre l'ensemble des moyens technologiques utilisé dans le développement de notre application.

Rappelons que, le projet que nous décrivons sa travers ce rapport concerne la réalisation d'une application web .c'est ta dire un site web dynamique utilisable via internet .ce type d'application repose principalement sur une architecture client-serveur .dans notre cas le client est le navigateur web, le serveur est un programme qui fonctionne sur un ordinateur distant.

1. Matériel utilisé

Deux PC portable HP

- Processeur intelCore I3 2 ,40 GHz
- 4 Go de mémoire vive
- Disque dur de capacité 500 Go
- Système d'exploitation Microsoft Windows 7 un 64bit

2. Installation

2.1. Reactjs

Créer une application React ou même un simple composant demandait la mise en place d'un ensemble d'outils et étapes pour la compilation du JSX en JavaScript, gérer les dépendances et optimiser les fichiers sources avant de pouvoir générer un artefact utilisable dans un autre projet

Avec la sortie de "**create-react-app**" développer et générer un artefact devient aussi simple que lancer une seule commande.

Cet outil est disponible sur Linux, Mac OS et Windows. On peut l'installer en utilisant NPM. Pour utiliser NPM, on aura besoin d'installerNodeJS

On aura besoin d'une version de Node>= 6 sur notre machine de développement. Node est utilisé par l'outil **create-react-app**pour gérer l'environnement de développement

Pour ceux qui utilisent des versions de Node différentes de la 6, on peut utiliser nvm pour avoir différentes versions de Node sur notre machine.

Pour créer une application React on lance les commandes suivantes :

Create-react-apptodo-app cd todo-app

Cette commande va créer un dossier appelé **todo-app**dans le répertoire dans lequel nous avons lancé cette commande puis générer la structure du projet et installer les dépendances du projet.

Pour lancer notre application nous pouvons utiliser la commande suivante : npm start

Une fois la commande est exécutée nous pourrons voir et tester l'application sur http://localhost:3000/.
L'application se rechargera à chaque fois qu'une modification est faite sur l'un des fichiers sources.

Pour lancer les tests nous utilisons la commande :

npm test

2.2. Nodejs

Pour installer NodeJS, il suffit simplement :

a. d'aller à l'adresse : https://nodejs.org/en/download/, de télécharger le gestionnaire d'installation au format **.msi** pour windows
b. Pour vérifier que tout s'est bien passé, il nous suffit de saisir la commande suivante qui nous retournera la version actuelle de Node:

node –v v8.11.1

c. Pour changer le répetoire d'installation des packages NPMet le mettre par exemple dans notre répertoire **home**il suffit de faire :

mkdir ~/.npm-global npmconfig set prefix '~/.npm-global'

Cela nous évitera de saisir à chaque fois le mot clé "sudo"à chaque installation de paquet npmen mode global (-g) :

d. Il faut ensuite mettre à jour le path de notre ordinateur pour prendre en compte ce nouveau répertoire.

2.3. Mongodb

Pour l'installation de mongodb voici les étapes qu'on a suivies

a. On a téléchargez la version de MongoDB pour notre système Windows à http://www.mongodb.org/downloads
b. On a Installé dans un répertoire à la racine de notre disque C:\ et utilisez un répertoire sans espaces comme c : \mongodb-2.6, par exemple.
c. Dans le répertoire C : \mongodb-2.6\bin, on créez un document nommé mongod.cfg et insérez-y les lignes suivantes:
   - logpath=C:\mongo\data\logs\mongo.log
   - dbpath=C:\mongo\data\db
d. Création d'un répertoire pour nos bases de données avec la commande *md C:\mongo\data\db*
e. Création d'un répertoire pour nos bases de données avec la commande *C:\mongo\data\logs*
f. Pour confirmer que notre installation a bien fonctionné, on exécute la commande c:\mongodb-2.6\bin\mongod.exe –config C:\mongodb-2.6\bin\mongod.cfg.
g. À ce point-ci, nous devrions être en mesure de nous connecter à notre instance MongoDB sans problèmes en utilisant la commande c:\mongodb-2.6\bin\mongo.exe

2.4. Grapheql

Après avoir installé NodeJs et vérifie la version de node et nmp.OnvaCréez un dossier de projet et ouvrez-le en VSCode

C:\Users\Admin>mkdir test-app C:\Users\Admin>cd test-app C:\Users\Admin\test-app>code.

Après on crée un fichier package.json qui contiendra toutes les dépendances de l'application serveur GraphQL.

```
{
  "name": "hello-world-server",
  "private": true,
  "scripts": {
    "start": "nodemon --ignore data/ server.js"
  },

  "dependencies": {
    "apollo-server-express": "^1.4.0",
    "body-parser": "^1.18.3",
    "cors": "^2.8.4",
    "express": "^4.16.3",
    "graphql": "^0.13.2",
    "graphql-tools": "^3.1.1"
},

  "devDependencies": {
    "nodemon» : "1.17.1"
  }
}
```

On va Installez les dépendances en utilisant la commande comme indiqué ci-dessous -

C:\Users\Admin\test-app>npm install

Après on réer une base de données de fichiers plats dans un dossier de données nous utilisons des fichiers plats pour stocker et récupérer des données et un autre dossier de données pour ajouter deux fichiers students.json et colleges.json. Maintenant on va crées le fichier db.js dans le dossier du projet comme suit -

```
Const { DataStore } = require('notarealdb');

Const store = new DataStore('./data');

module.exports = {
students:store.collection('students'),    colleges:store.collection('colleges')
```

Création d'un :

Fichier de schéma schema.graphqldans le dossier du projet en cours et l'ajout de contenu suivant :

```
typeQuery {     test :
String }
```

Fichier de résolutionresolvers.js dans le dossier du projet en cours et l'ajout le contenu suivant :

```
const Query = {
test: () => 'Test Success, GraphQL server is up & running !!'
} module.exports = {Query}
```

Fichier serveur.js et configurez GraphQL comme suit -

```
constbodyParser = require('body-parser');
constcors = require('cors'); const express
= require('express'); constdb =
require('./db');

const port = process.env.PORT || 9000; const app =
express();

constfs = require('fs')
consttypeDefs = fs.readFileSync('./schema.graphql',{encoding:'utf-8'}) const
resolvers = require('./resolvers')

const {makeExecutableSchema} = require('graphql-tools') const
schema = makeExecutableSchema({typeDefs, resolvers})

app.use(cors(), bodyParser.json());

const{graphiqlExpress,graphqlExpress}=require('apollo-server-express')
app.use('/graphql',graphqlExpress({schema}))
app.use('/graphiql',graphiqlExpress({endpointURL:'/graphql'}))

app.listen(
port, () =>console.info (
`Server started on port ${port}`
)
);
```

Enfin on Lance l'application et on teste avec GraphiQL pour ça on Vérifie la structure de dossiers du projet test-app comme suit -

```
test-app /    -->package.json
  -->db.js    -->data
students.jsoncolleges.js
on    -->resolvers.js
  -->schema.graphql
```

-->server.js
Apréss on lance npm start: C:\Users\Admin\test-app>npm start

Le serveur fonctionne sous le port 9000, nous pouvons donc tester l'application à l'aide de l'outil GraphiQL. On Ouvre le navigateur et on entre l'URLhttp : // localhost: 9000 / graphiql après on tape la requête suivante dans l'éditeur

```
{
  Test
}
```

La réponse du serveur est donnée ci-dessous -

```
{
  "data": {
    "test": "Test Success, GraphQL server is running !!"
} }
```

2.5. Appolo server

Pour l'installer on lance : npminstallapollo-server graphql

Lors de l'ajout d'Apollo Server à une application existante, un package de support de serveur HTTP correspondant doit également être installé.

À ce stade, nous sommes prêts à accepter les connexions au serveur. Ceci est fait en appelant la méthode listen sur l'instance d'ApolloServer qui a été créée à l'étape précédente :

```
server.listen().then(({ url }) => {   console.log(` Server ready at ${url}`)
});
```

Lorsque la configuration ci-dessus est terminée, nous pouvons maintenant lancer l'application Node, avec Apollo Server, pour la première fois.
Après avoir démarré le serveur, il devrait imprimer un message sur la console pour indiquer qu'il est prêt

```
node index.js
 Server ready at http://localhost:4000/
```

3. La structure de notre projet React

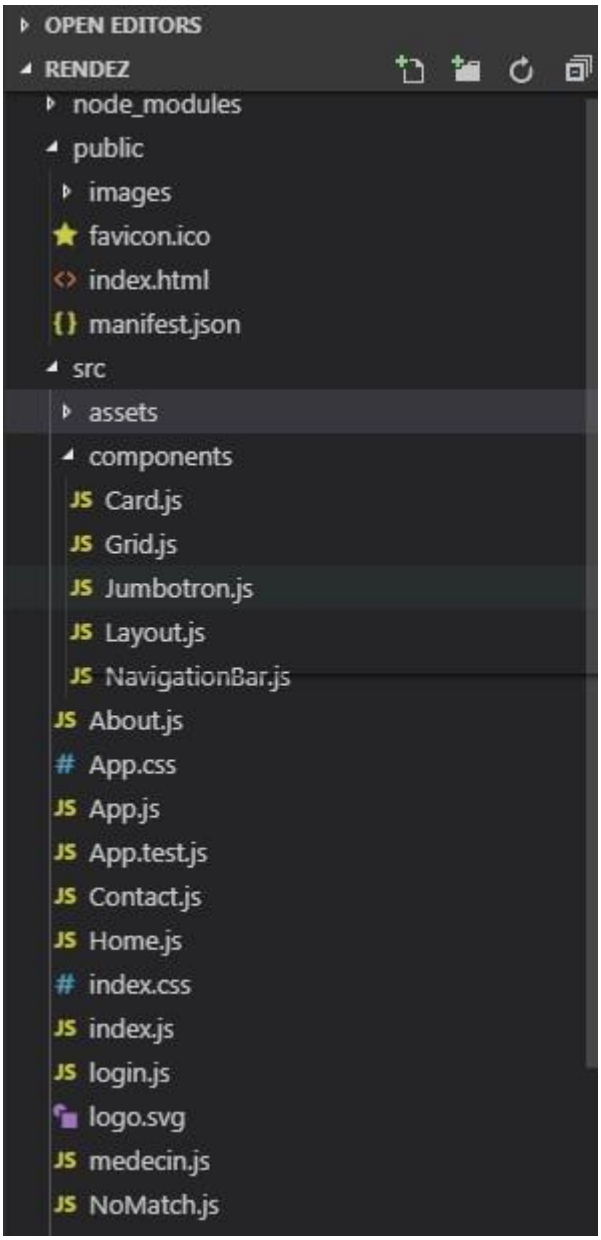
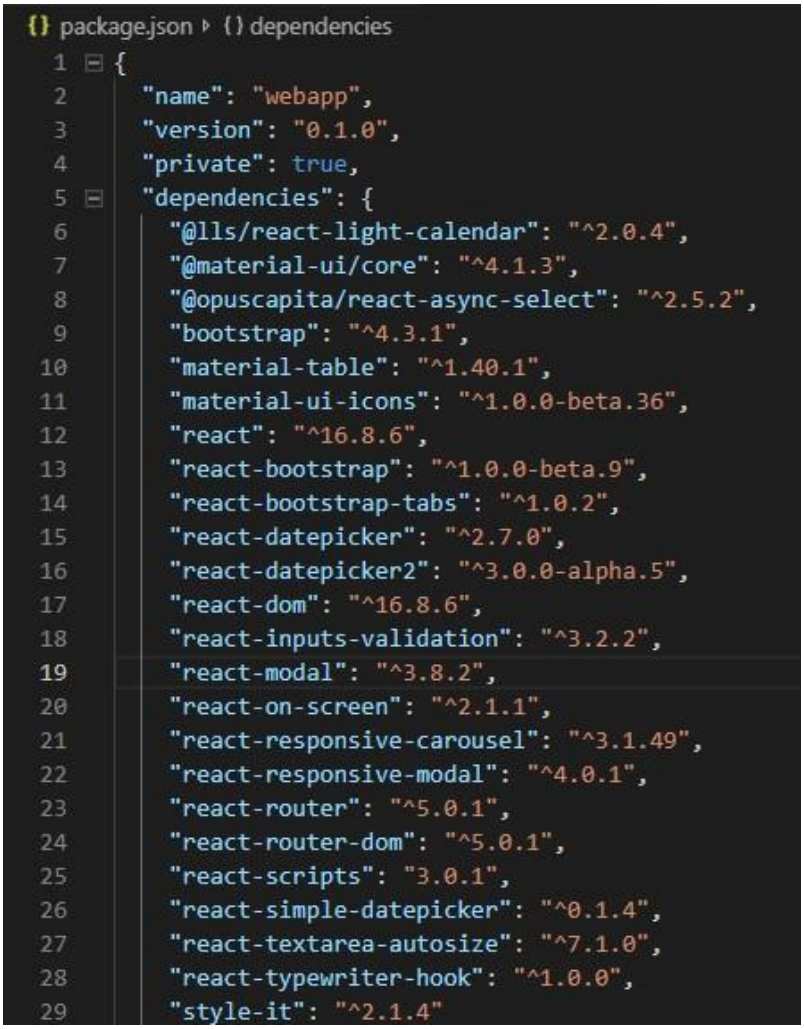


Figure14 : la structure de projet

La figure ci-dessus représente la structure de projet, le dossier public qui contient les images utilisés, le dossier src qui contient les différents composants (components :

home.js, navigationBar.js gridjs.Card.js….), le fichier App.css qui contient les styles css, Et App .js

Les packages utilisés sont présents dans le fichier package.json comme le montre la figure suivante :



```
{} package.json ▸ {} dependencies
{
  "name": "webapp",
  "version": "0.1.0",
  "private": true,
  "dependencies": {
    "@lls/react-light-calendar": "^2.0.4",
    "@material-ui/core": "^4.1.3",
    "@opuscapita/react-async-select": "^2.5.2",
    "bootstrap": "^4.3.1",
    "material-table": "^1.40.1",
    "material-ui-icons": "^1.0.0-beta.36",
    "react": "^16.8.6",
    "react-bootstrap": "^1.0.0-beta.9",
    "react-bootstrap-tabs": "^1.0.2",
    "react-datepicker": "^2.7.0",
    "react-datepicker2": "^3.0.0-alpha.5",
    "react-dom": "^16.8.6",
    "react-inputs-validation": "^3.2.2",
    "react-modal": "^3.8.2",
    "react-on-screen": "^2.1.1",
    "react-responsive-carousel": "^3.1.49",
    "react-responsive-modal": "^4.0.1",
    "react-router": "^5.0.1",
    "react-router-dom": "^5.0.1",
    "react-scripts": "3.0.1",
    "react-simple-datepicker": "^0.1.4",
    "react-textarea-autosize": "^7.1.0",
    "react-typewriter-hook": "^1.0.0",
    "style-it": "^2.1.4"
```

Figure15 : les packages utilisés (package.json)

3.1. La création de la base de données

Notre base de données est de type NOSQL, centrée sur les documents. MongoDB. Voici Un exemple de code utilisé :

```
db.ssousou.insert([
. {
. first_name:"medecin",
. last_name:"varchar",
. pwd:"mot de passe",
. contact:"number",
. specialty:"cardiologie,psicologue,Chirurgie générale",
.
.          email: {
.             bsonType : "string",
.             pattern : "chaymahimeur@gmail.com",
.             description: "must be a string and match the regular expression pattern"
.          }
. }]);
lkWriteResult({
      "writeErrors" : [ ],
      "writeConcernErrors" : [ ],
      "nInserted" : 1,
      "nUpserted" : 0,
      "nMatched" : 0,
      "nModified" : 0,
      "nRemoved" : 0,
      "upserted" : [ ]
```

Figure16 : code compte médecin

L'opération précédente crée un utilisateur nomme médecin dans la base de donnes soussou, pour crée un compte médecin.

3.2. Description d'AppMeq

Les principaux écrans de notre application web

a)Ecran d'accueil « Accueil » :

L'utilisateur va se retrouver dans la page d'accueil du l'application AppMeq où il y a une barre de navigation incluant :

- l'accueil,
- espace patient,
- espace médecin,
- espace conseils,
- zone de la recherche

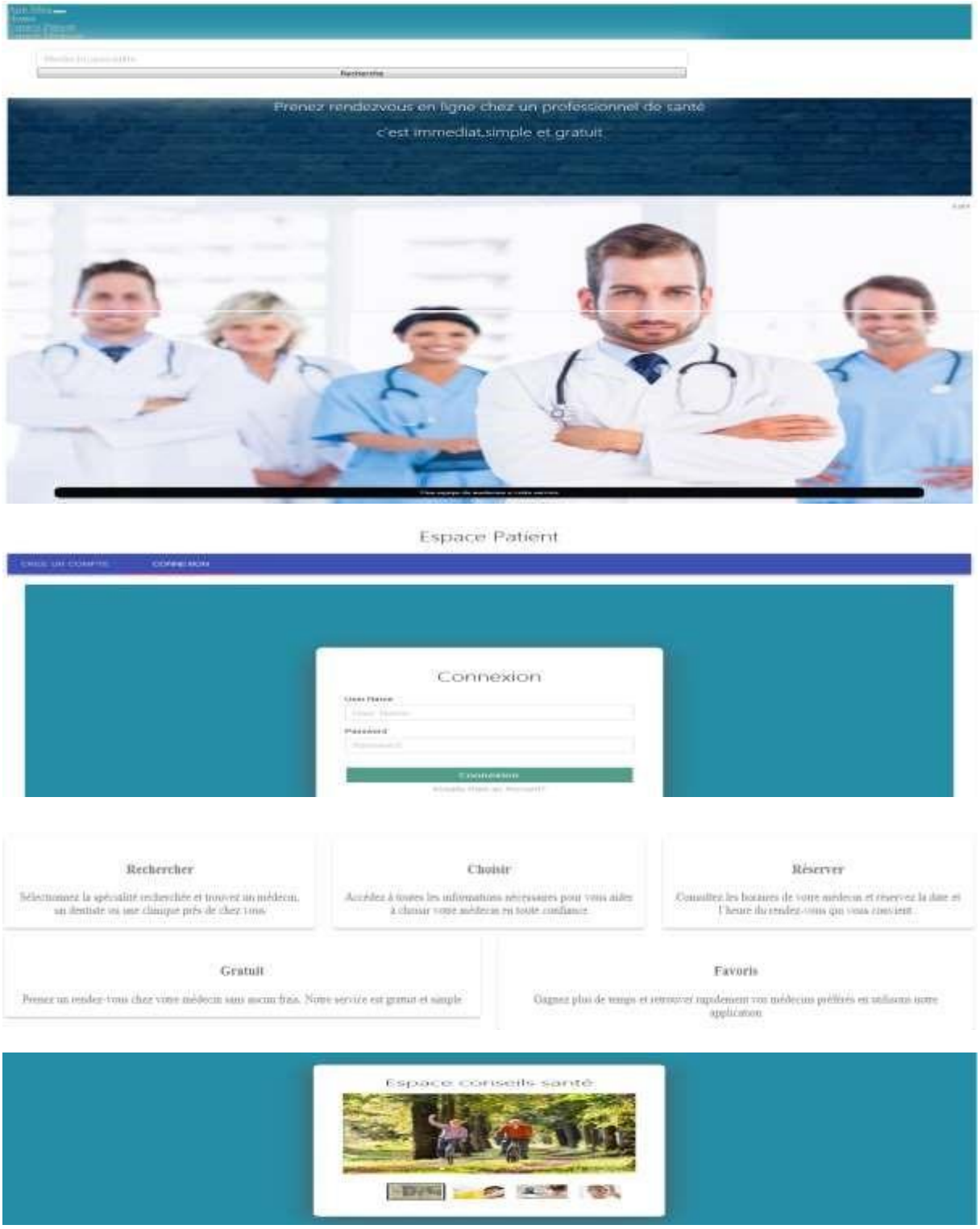


Figure 17 : Ecran d'accueil du l'application AppMeq

Dans cette page le patient peut faire une recherche de médecin par nom /par spécialité, crée un compte (pour patient et médecin) s'authentifier par le nom d'utilisateur et le mot de passe Pour le patient, par email et mot de passe pour le médecin.

```
importReact, { Component } from'react'; import { BrowserRouterasRouter,
Route, Switch } from'react-router-dom'; import { Home } from'./Home';
import { About } from'./About'; import { Contact } from'./Contact'; import
{ NoMatch } from'./NoMatch'; import { Layout } from'./components/Layout';
import { NavigationBar } from'./components/NavigationBar'; import {
Jumbotron } from'./components/Jumbotron'; import"react-responsive-
carousel/lib/styles/carousel.min.css"; import { Carousel } from"react-
responsive-carousel"; import { Card } from'./components/Card'; import {
Grid } from'./components/Grid'; import"./App.css";
classAppextendsComponent {   render() {     return (
<React.Fragment>
<Router>
<NavigationBar/>
<Jumbotron/>
<CarouselautoPlay>
</Carousel>
<Layout>
<Switch>
<Routeexactpath="/home"component={Home}/>
<Routepath="/about"component={About}/>
<Routepath="/contact"component={Contact}/>
<Routecomponent={NoMatch}/>
</Switch>
</Layout>
<Card/>
<Grid/>
</Router>
</React.Fragment>
) ;
  }
}
exportdefaultApp;
```

Figure 18 : Code javascript de la page d'accueil

b) Ecran espace patient :

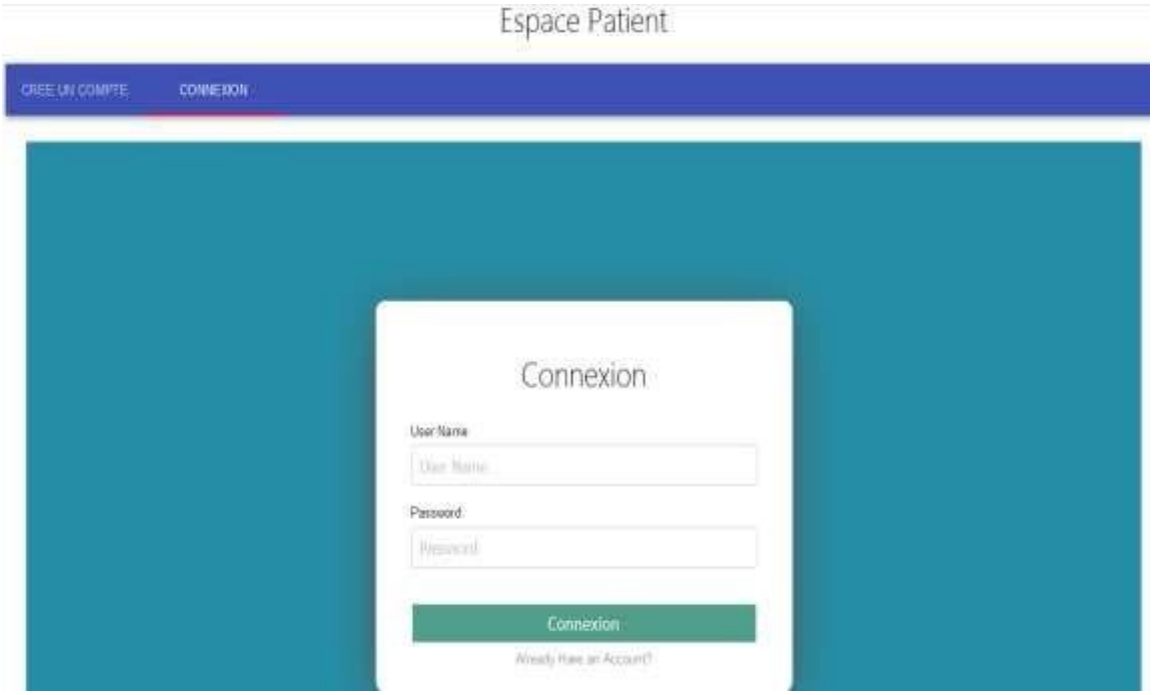


Figure 19 : Connexion patient

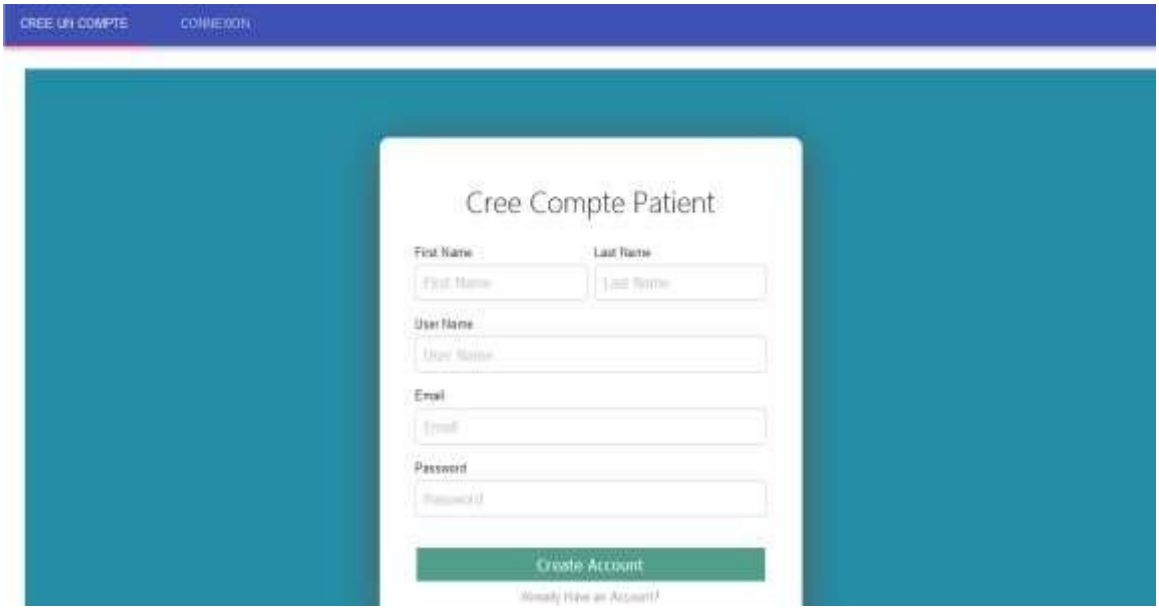


Figure 20 : création de compte patient

Le patient peut créer son compte et s'authentifie à partir de page «espace patient »

c) Ecran espace médecin :

Figure 21 : connexion médecin

Figure 22 : créé compte médecin

La Page espace médecin permet de créer un compte médecin ou de s'authentifie par email et mot de passe.

```css
.createAccount {
  width: 100%;
  display: flex;
  flex-direction: column;
  align-items: center;
}

.createAccount button {
  background-color: #519e8a;
  color: #fff;
  border: 2px solid #fff;
  width: 100%;
  margin-top: 1em;
  padding: 8px 0px;
  font-size: 1em;
  font-weight: lighter;
  letter-spacing: 1px;
  margin-bottom: 0.25em;
}
```

```css
label {
  font-size: 0.8em;
  margin-bottom: 0.25em;
  color: #222;
  font-weight: lighter;
}

input {
  padding: 10px 10px;
  border-radius: 5px;
  outline: none;
  border: 1px solid #cfcfcf;
}

input::placeholder {
  font-size: 1.2em;
  font-weight: lighter;
  color: #999;
}
```

```css
.wrapper {
  height: 100vh;
  width: 100%;
  display: flex;
  flex-direction: column;
  justify-content: center;
  align-items: center;
  background-color: #258ea6;
}

.form-wrapper {
  width: 400px;
  display: flex;
  flex-direction: column;
  padding: 20px 40px;
  border-radius: 10px;
  box-shadow: 0px 10px 50px #555;
  background-color: #ffffff;
}
```

Figure 23 : une partie du code css (app.css)

d) Ecran compte médecin :

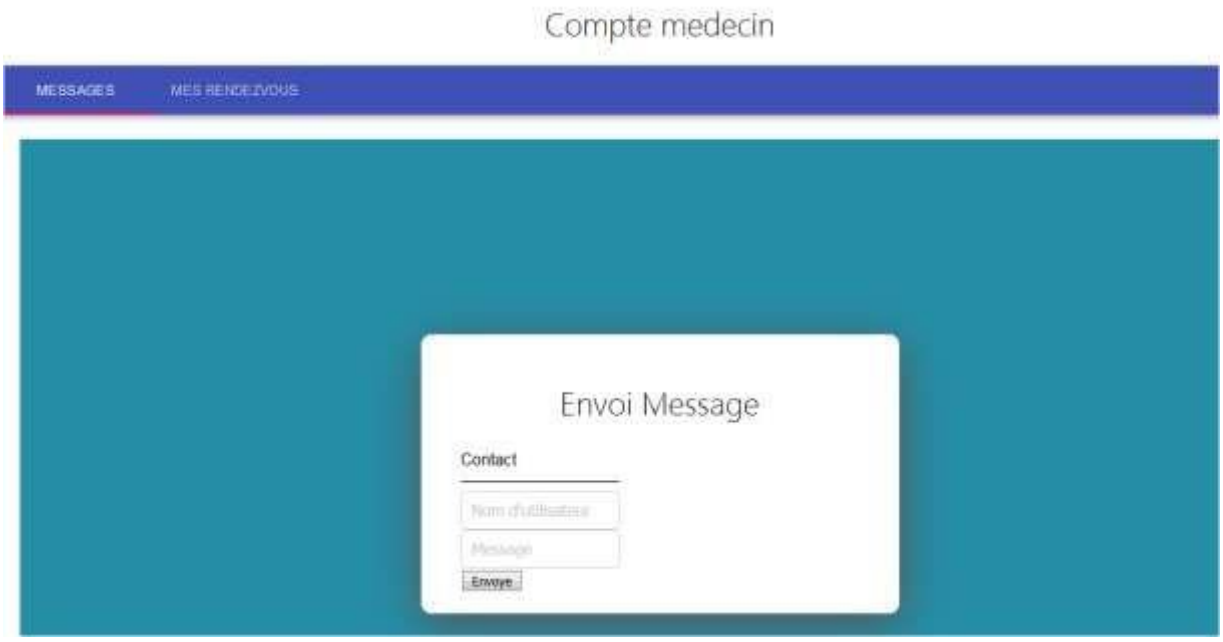


Figure 24 : compte médecin

Figure 25 : liste des patients

Le compte médecin à une table dynamique qui contient la liste des rendez-vous patients, le médecin peut faire une recherche par nom patient ajouter, supprimer et modifier un rendez-vous.

A travers l'espace « message » le médecin peut envoyer des messages à ses patients
Il suffit de taper le user Name de patient et le message et le envoyé on cliquant sur le Botton envoyé.

Apres avoir saisie le nom de médecin ou la spécialité une liste s'affiche contient tous les médecins qui porte le même nom ou spécialité.

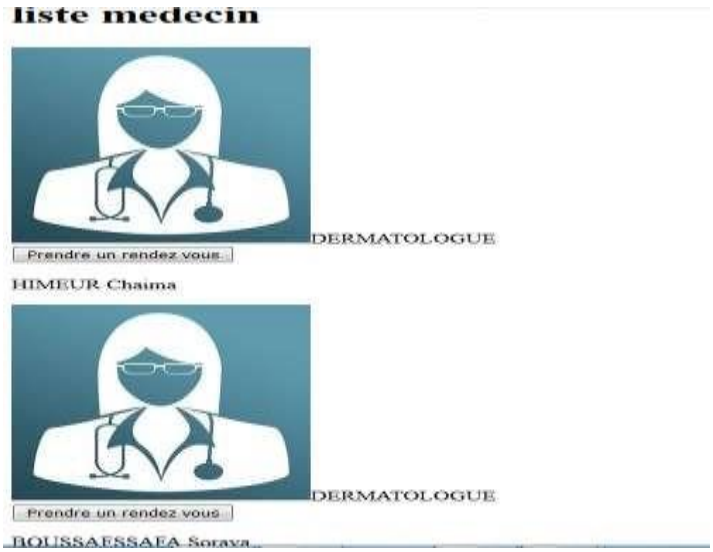


Figure 26 : liste médecins

e) Ecran profil médecin :

Apres avoir choisi le médecin qui vous convient à partir de la liste des médecins un profil vous s'affiche contient une présentation de médecin, planning, la carte et informations d'accès, tarifs, espace pour réserver un rendez-vous et autre pour le annuler

Figure 27 : Profil médecin

Figure 28 : carte et information d'accès

f) Ecran mon rendez-vous :

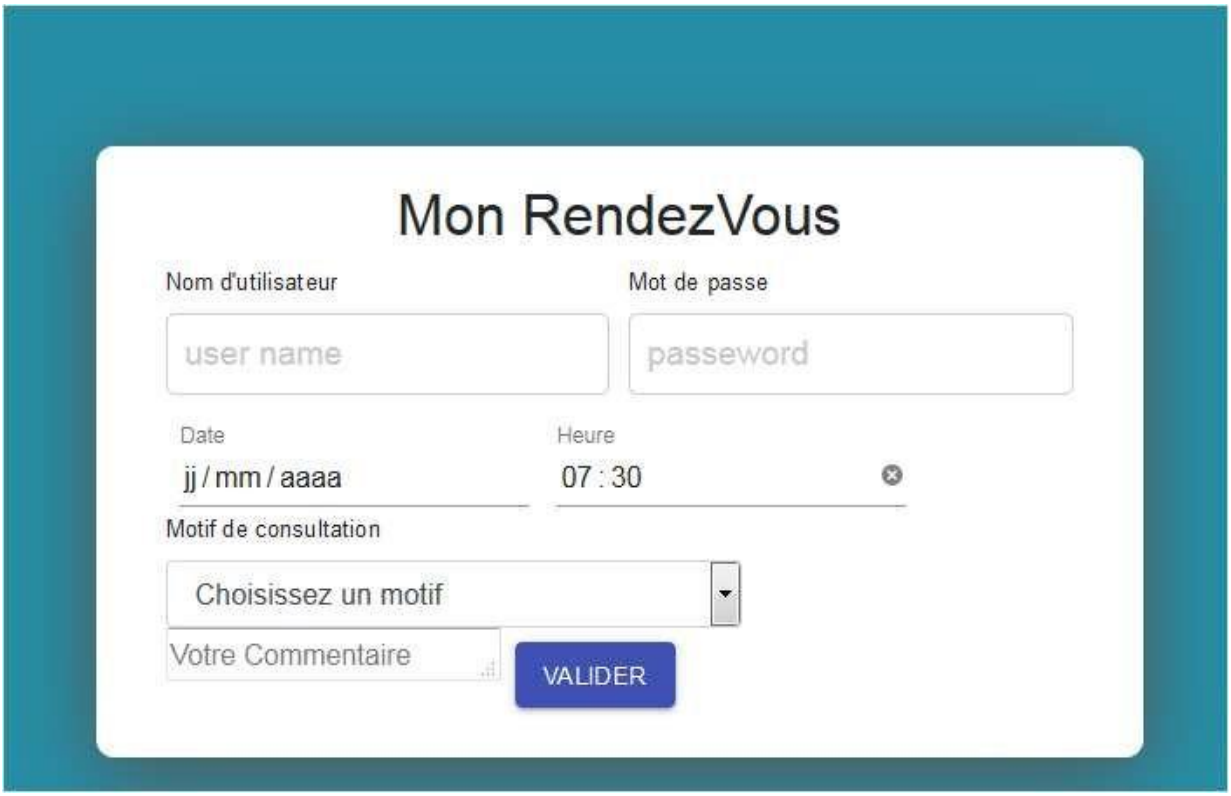


Figure 29 : écran réserver un rendez-vous

La page de réservation permet au patient de sélectionner la date et l'heure du rendez-vous, choisir le motif de consultation et lisser un commentaire et en fin il clique sur le Bouton «valider» après avoir saisis son nom d'utilisateur et mot de passe.

g) Ecran annuler mon rendez-vous :

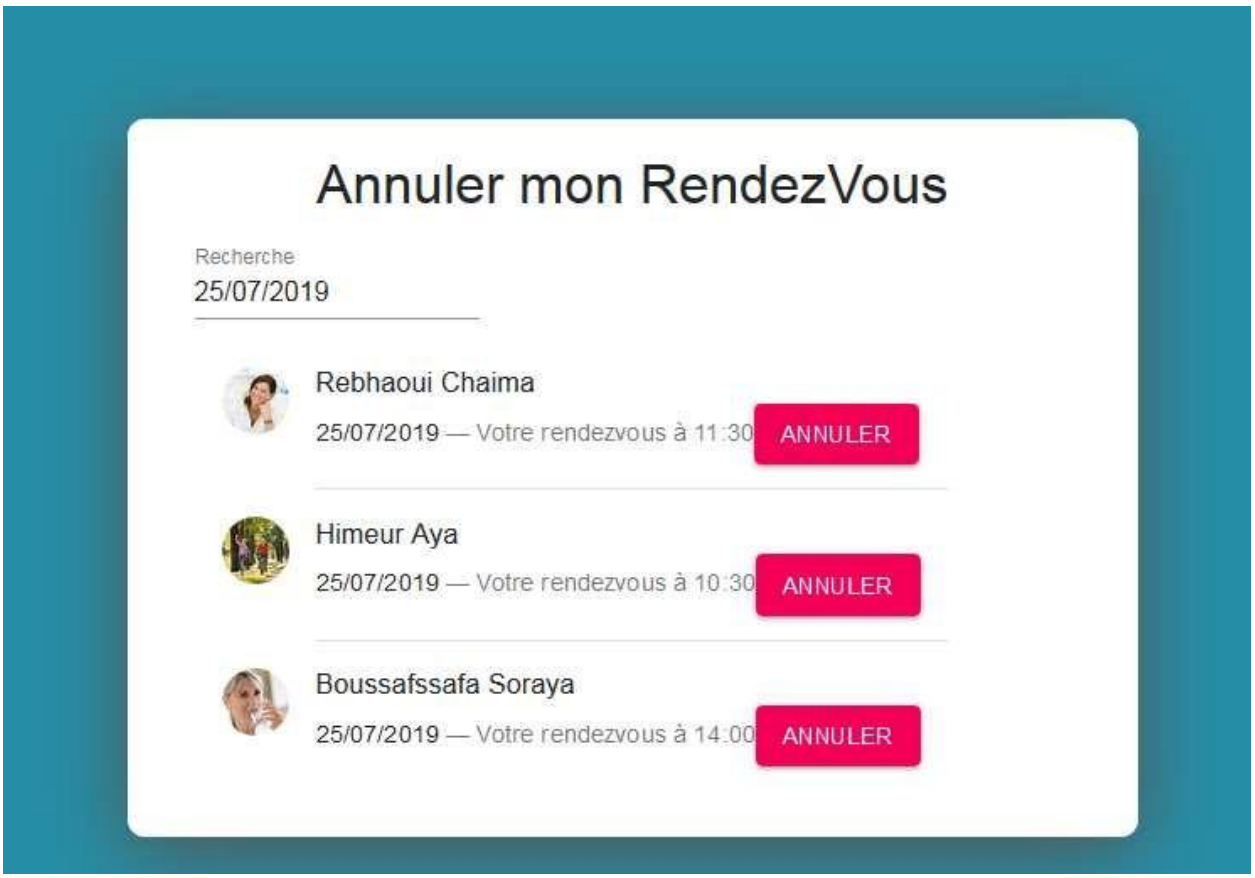


Figure 30 : écran annuler un rendez-vous

Dans la partie « annuler un rendez-vous » le patient peut annuler son rendez-vous on lançon une recherche par date, une liste des patients qui ont la même date s'affiche, le patient sélectionne son nom, clique sur le bouton annuler une boite de dialogue s'affiche pour entrer le mot de passe et confirmer l'annulation.

Conclusion Générale

Conclusion

Dans ce chapitre nous avons décrit l'aspect pratique de notre projet, tout d'abord nous avons, listé l'ensemble des moyens technologiques utilises matériels et logiciels puis nous avons présenté les déférentes interfaces de notre application ainsi que leurs comportements à travers cette réalisation.

Nous nous put atteindre les objectifs fixés lors de la phase d'analyse des besoins nous avons pu découvrir plusieurs outils informatiques lors de développement, nous avons essayé de fournir un ensemble d'interfaces intuitives et simples à utiliser

Conclusion Générale

Gérer efficacement les rendez-vous chez un professionnel de santé est un défi au quotidien car cela requiert un équilibre permanent entre la gestion du personnel, le respect de la réglementation, la gestion des coûts opérationnels et le maintien de la satisfaction des patients. Voilà pourquoi plusieurs cliniques optent pour l'intégration des technologies pour optimiser leur fonctionnement et fournir un meilleur accès à leurs patients.

Notre Projet porte sur l'organisation et l'automatisation de la prise d'un rendez-vous, afin d'augmenter la fiabilité, l'efficacité de l'effort humain et de faciliter les tâches pénibles au sein d'un organisme ; en le déroulant sur une application.

Au cours de ce mémoire, nous avons présenté les déférentes étapes de la conception et la réalisation de notre application web.

L'objectif majeur de notre projet consiste à réaliser un système de gestion des rendez-vous médicaux permettant d'une part aux patients de chercher un médecin et prendre un rendez-vous sont attendre l'ouverture de notre clinique. Nous les libèrent de leurs contraintes professionnelles et familiales, d'où une simplicité de la charge quotidienne. Pour faire gagner du temps et d'autre part aux médecines de gérer les rendez-vous des patients.

De ce fait, notre application englobera plusieurs médecins de différentes spécialités exerçant.

Afin de satisfaire les besoins des utilisateurs nous avons commencé la conception en utilisant le formalisme UML, nous avons procédé à la phase de réalisation au cours de laquelle nous sommes familiarisés avec le langage de programmation reactjs.

Ce projet a fait l'objet d'une expérience très intéressante, car elle nous a permis de nous familiariser avec de nouvelles notions d'une part, et d'améliorer nos connaissances et nos compétences dans le domaine de la programmation, et d'autre part de renforcer notre sens de responsabilité dans la gestion des projets qui permet l'insertion dans le domaine professionnel.

Enfin, nous espérons que notre projet puisse répondre aux besoins fixés et satisfaire toutes les personnes qui ont contribué à sa réalisation, ainsi que les utilisateurs pour l'exercice de leurs professions.

Référence Bibliographiques

Références bibliographiques

[1] :https://fr.reactjs.org/

[2] :https://graphql.org/learn/

[3] :https://nodejs.org/

[4] :https://apollographql.com/docs/apollo-server/

[5] :https://www.mongodb.com/

[6] : D- D-Eddine Gestion de Cabinet Médical 2014-2015

[7] : https://docteur-rendez-vous.fr/

[8] : M-Bilal Un système de prise de rendez-vous en ligne pour une clinique, 2016/2017

[9] : https://prendreunrendezvous.fr/

[10] : Mr. BENEDDRA Rachid Gestion de Cabinet Médical 2014-2015

[11] : https://expressjs.com/fr/

[12] https://www.ideematic.com/dictionnaire-web/application-web

[13] http://stackoverflow.com

Résumé

L'objectif de notre projet de fin d'étude, présenté dans ce rapport, est la conception et la réalisation d'une application web simple rapide et facile à s'intégrer dans l'environnement de travail médical.

Notre application se traduit par la mise en œuvre d'un système qui facilite la prise de rendez-vous en ligne et laisser libre l'accès du client pour prendre le rendez-vous à l'heure qui l'arrange et d'éliminer le temps d'attente des clients entre les rendez-vous. La modélisation du système a été faite par UML.Et la création par des langages qui défaire des langages habitué qui sont reacjs, nodejs, à l'aide du logiciel de bases de donnes mongodb.

Summary

The goal of our end-of-study project, presented in this report, is the design and implementation of a simple web application that is fast and easy to integrate into the medical work environment.

Our application translates into the implementation of a system that makes it easy to make appointments online and give the client free access to make the appointment at the time that suits him and eliminate the waiting time for clients between appointments. The modeling of the system was done by UML. And the creation by the languages that are reacjs, nodejs, using the mongodb database software

ملخص

الهدف من مشروع نهاية الدراسة ، الوارد في هذا التقرير ، هو تصميم وتنفيذ تطبيق ويب بسيط سريع وسهل الاندماج في بيئة العمل الطبي.

يترجم تطبيقنا إلى تنفيذ نظام يجعل من السهل تحديد المواعيد عبر الإنترنت ومنح العميل حق الوصول المجاني لتحديد الموعد في الوقت الذي يناسبه والقضاء على وقت الانتظار للعملاء بين المواعيد. تم عمل نمذجة النظام بواسطة LMU. وإنشاء اللغات التي هي sjcae،sjesae ، باستخدام برنامج قاعدة البيانات bjsgjem