Chapitre 4 : Réalisation

Introduction :

Dans ce paragraphe nous allons décrit sommairement et conformément à la méthodologie SCRUM, les étapes de réalisation de notre projet, ensuite nous allons présenter la réalisation par rapport au produit final en décrivant l’environnement de travail et les technologies utilisées pour la mise en place de notre plateforme. Enfin, nous présentons quelques captures d’écran de l’application réalisée.

1.Réalisation par rapport au backlog :

La particularité de la méthodologie SCRUM, déjà explicité dans la partie 1.3.3, fait que le dé- roulement de la réalisation du projet est planifié dés son début. Dans ce paragraphe nous allons décrire brièvement le déroulement des sprints pour ensuite parler du facteur de disponibilité.

-Le déroulement des sprints :

La décomposition du projet en des sous tâches appelés aussi sprint a été faite au début de notre projet. Il s’agit ensuite, à chaque étape, de faire une réunion d’équipe afin de :

* + - * Enoncer les buts du sprint (le produit du sprint).
      * Proposer des estimations temporelles en nombre de jour.
      * Valider la réalisation à la fin du sprint en effectuant une démonstration du travail réalisé.
      * Modifier éventuellement certains objectifs définis au départ en fonction de nouvelles re- quêtes.

2.Environnement de travail :

Cette partie présente l’environnement matériel mis à la disposition du présent projet ainsi

que l’environnement logiciel que nous avons exploité et nous présentons les choix

technologiques en les justifiant.

2.1.Environnement matériel :

Dans notre projet nous avons développé notre application en utilisant un ordinateur dont la configuration est décrite ci dessous :

|  |  |  |
| --- | --- | --- |
| Système d’exploitation | Carte graphique | Processeur |
| Windows 7 64bit | **NVIDIA® GeForce®  GTX 1650TI avec 4 Go GDDR6** | **Intel® I7-10870H** |

-

2.2. Environnement logiciel :

2.2.1 : La modélisation UML :

Pour la modélisation UML j’ai utilisé 2 logiciels :

-J’ai créé le diagramme de cas d’utilisation avec :

Pacestar UML :



Pacestar UML Diagrammer permet de créer des diagrammes en UML.  
  
Il dispose d'un module WYSIWYG, supporte l'OMG UML 2.0, des fonctions classiques de diagramme...  
  
Il supporte en import et export les fichiers JPG, GIF, PNG, BMP, GIF, DIB, WMF et EMF.  
  
On peux insérer des liens hypertexte vers d'autres diagrammes et fichiers externes.  
  
Il peut fonctionner à l'aide d'un système de glisser-déposer.

-Pour le diagramme des classes et des séquences j’ai utilisé le site :



Figure :Draw .io

Draw .io est un site de modélisation de UML dont on peux utilisé plusieurs types de diagrammes comme :

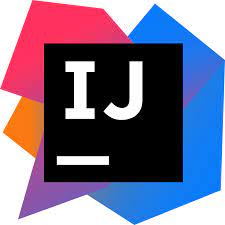
-diagramme des classes

-diagramme de séquences

-diagramme d’activités …

2.2.2 :Backend :

2.2.2.1



Au niveau BackEnd j’ai utilisé le logiciel de développement Intelliji Idea ultimate car il géré facilement la framework ‘ Spring boot’ et elle fait la mise a jour des dépendances d’une façon automatique aussi elle est facil pour pushé en github.

2.2.2.2 : Spring Boot :



Figure 1 : Spring boot

Pour la partie BackEnd j’ai utilisé Spring boot car il adapte l’architecture microservice aussi cette language est bien demandé dans le marché de travail

2.2.2.3 SQL et MongoDB :

Au Niveau de base de données j’ai utilisé aussi le MongoDB et aussi MySQL comme j’ai expliqué dans le chapitre précèdent et pour gérer les deux base de données on a utilisé pour le MySQL :

-XAMPP

XAMPP est un ensemble de logiciels servant à mettre en place aisément un serveur Web, un serveur FTP et un serveur de messagerie électronique. C'est une distribution de logiciels libres (X Apache MySQL Perl PHP) offrant une bonne souplesse d'utilisation, reconnue pour son installation simple et rapide. Ainsi, il est à la portée de la plupart de personnes dans la mesure où il ne requiert pas de connaissances spécifiques et fonctionne, qui plus est , sur les dispositifs d'exploitation les plus communs.

Et pour MongoDB on a utilisé :

-MongoDB compass :



MongoDB Compass analyse vos documents et affiche des structures riches au sein de vos collections via une interface graphique intuitive. Il vous permet de visualiser et d'explorer rapidement votre schéma pour comprendre la fréquence, les types et les plages de champs de votre ensemble de données.

-Studio 3T :



Studio 3T peux :

-Générer le code du conducteur en sept langues

-Décomposer les requêtes d'agrégation en étapes

-Requêtes de saisie semi-automatique dans le shell mongo

-Créer des requêtes par glisser-déposer

-Utiliser le masquage des données pour protéger les données sensibles

-Transformer le schéma d'une collection en quelques clics

-Explorer le schéma de données et trouver les valeurs aberrantes

-Migration entre SQL et MongoDB

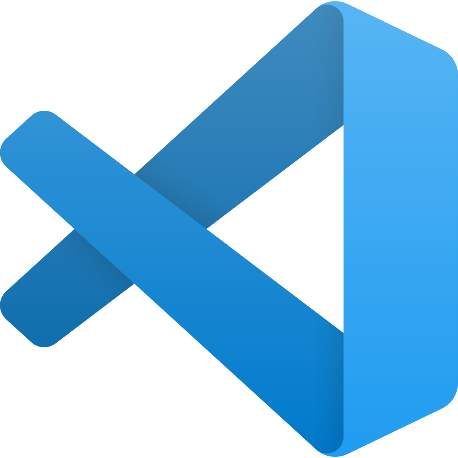
-Écrire SQL pour interroger MongoDB

-Planifier les tâches MongoDB

2.2.2.4

2.2.3 : FrontEnd :

2.2.3.1 Visual Studio Code:



Au niveau de FrontEnd j’ai utilisé Vs code pour la manupilation de framework Angular car on la bien maitrisé pendant notre formation .

#### 2..2.3.2 : Angular :



Figure 2: Angular cli 11

Angular est un Framework Javascript côté client qui permet de réaliser des applications Cross-platform Web, Mobile et Desktop.

Angular est Open Source et développé par Google. Il utilise l’architecture MVM (Modèle Vue Modèle), proche du modèle MVC.

Leurs Concepts est Orienté Single Page Application (SPA), Étendre le langage HTML (directives), Data Binding bidirectionnel, Dépendance Injection, pipe ... et le transpiler (Type Script)

#### 2.2.3.3 type script :



Figure 3 : Type Script

Type script est un langage transcompilé dont son code n’est pas directement exécuté, il est compilé en Javascript. C’est le code Javascript qui sera exécuté par les navigateurs. Donc  tout JavaScript est déjà Type Script et c’est ça sont point fort.

L’avantage de ce langage par rapport à Javascript est de pouvoir typer fortement les variables ce qui permet de rendre ce langage moins sensible aux bugs. D’autres apports syntaxiques sont faits par Type script comme les classes, les modules ou les interfaces ce qui permet de rendre le langage plus attrayant.

2.2.3.4 : Node JS

Une image contenant texte, clipart

Description générée automatiquement

Figure 4:Node .js

Est le Runtime de JavaScript en dehors du browser construit au-dessus de la machine virtuelle JavaScript V8 de Google (généralement pour la programmation de serveur et desktop).

Tandis que les moteurs de JavaScript sont traditionnellement exécutés dans des navigateurs Internet pour former le côté de client d'une application de client /serveur, les bibliothèques NodeJs sont concentrées en construction des applications côté serveur dans le JavaScript.

NodeJs traite des requêtes entrantes dans un événement constant empilent et envoient de petites requêtes l'un après l'autre sans attendre des réponses.

2.2.3.5 : Html (HyperText Markup Language) :



Figure 5: HTML

Le ***HyperText Markup Language***, généralement abrégé **HTML** ou dans sa dernière version [**HTML5**](https://fr.wikipedia.org/wiki/HTML5), est le [langage de balisage](https://fr.wikipedia.org/wiki/Langage_de_balisage) conçu pour représenter les [pages web](https://fr.wikipedia.org/wiki/Page_web).

Ce langage permet :

d’écrire de l’[hypertexte](https://fr.wikipedia.org/wiki/Hypertexte), d’où son nom,

de structurer [sémantiquement](https://fr.wikipedia.org/wiki/S%C3%A9mantique) la page,

de mettre en forme le contenu,

de créer des formulaires de saisie,

d’inclure des [ressources](https://fr.wikipedia.org/wiki/Ressource_du_World_Wide_Web) [multimédias](https://fr.wikipedia.org/wiki/Multim%C3%A9dia) dont des [images](https://fr.wikipedia.org/wiki/Image_num%C3%A9rique), des [vidéos](https://fr.wikipedia.org/wiki/Vid%C3%A9o), et des programmes informatiques,

de créer des documents [interopérables](https://fr.wikipedia.org/wiki/Interop%C3%A9rabilit%C3%A9_en_informatique) avec des équipements très variés de manière conforme aux exigences de l’[accessibilité du web](https://fr.wikipedia.org/wiki/Accessibilit%C3%A9_du_web).

2.2.3.6 CSS :



Css est un outil de design pour donner une vue agréable a notre site à travers des balises et des fonctions .

2.2.4 : Les API :

Bien que l’analyse soit ici séparée du développement, cela n’est pas en lien avec le déroulement chronologique du stage. Etant en recherche et développement, le développement a évolué au fur et à mesure des recherches autour du projet.

### 2.2.4.1 Web service

Un service Web met à disposition un service via Internet. Il constitue ainsi une interface permettant à deux machines (ou applications) de communiquer. Pour y parvenir, la technologie doit disposer de deux propriétés essentielles :

* être multiplateforme : il n’est pas nécessaire que le client et le serveur aient la même configuration pour pouvoir communiquer. Le service Web leur permet de se retrouver à un même niveau.
* être partagée : dans la plupart des cas, un service Web est à disposition de plus d’un client. Différents clients accèdent à ce service via Internet.

Lorsqu’un service Web est utilisé, un client envoie une requête à un serveur et déclenche ainsi une action auprès de ce serveur. Le serveur renvoie ensuite une réponse au client.

2.2.4.2 Services Web RESTFul

Les services REST sont sans états (Stateless), Chaque requête envoyée au serveur doit contenir toutes les informations relatives à son état et est traitée indépendamment de toutes autres requêtes

Minimisation des ressources systèmes (pas de gestion de session, ni d’état)

Interface uniforme basée sur les méthodes HTTP

(GET, POST, PUT, DELETE)

Les architectures RESTFul sont construites à partir de ressources uniquement identifiées par des URI(s)

2.2.4.3 : API Google Map :

Pour développer une carte de map sur mon site j’ai utilisé l’API de google Map pour le mettre sur mon site même que si il n’est pas encore prêt d’être usable pour des problème de paiement.

2.2.4.4 : API Gateway :

Une API Gateway (également appelée passerelle API) est le point d'entrée unique pour les API et microservices back-end définis (qui peuvent être à la fois internes et externes). Assise devant les API, l'API Gateway agit en tant que protecteur, renforçant la sécurité et assurant l'évolutivité et la haute disponibilité.

2.2.5 : GitHub :



2.2.5.1 : Repertoire du premièer microservice UserRole :

Une image contenant texte, capture d’écran, ordinateur, portable

Description générée automatiquement

2.2.5.2 Répertoire de la 2ème microservice DATACENTER :

Une image contenant texte, capture d’écran, ordinateur, intérieur

Description générée automatiquement

2.2.5.3 : Répertoire de la partie FrontEnd :

Une image contenant texte

Description générée automatiquement

## .3 Résultat

Les interfaces Homme/Machine constituent un élément important dans la réussite d’une application. Cette contribution est d’autant plus importante lorsqu’il s’agit d’une application Web. Ainsi, nous essayerons dans cette partie de présenter les interfaces les plus indicatives de notre application