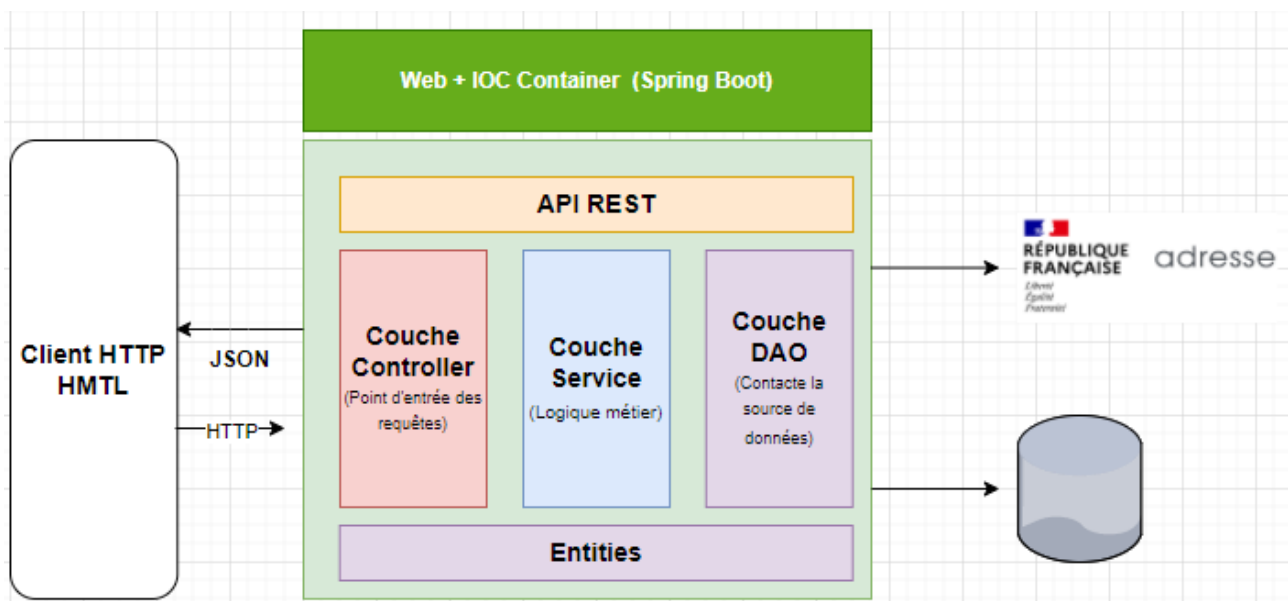


Titre du Projet		Réf. Du projet	
Proof Of Concept Réservation		01_POC_R	
Historique			
Version	Rédacteur	Modification	Date
1.0	Jules MESSOLO		08/08/2021

Table des matières

1. L'Architecture de l'API de réservation.....	3
2. Principe de l'architecture.....	3
3. Technologies préconisés.....	4
4. Hypothèses de validation du POC.....	4
5. Gestion du projet : Méthode Agile et BDD.....	5
6. Outils d'intégration et déploiement continue: GitLab CI.....	6
7. Pipeline CI/CD : GitLab CI.....	7
8. Stratégie de Test et Plan de Test.....	7
9. Validation des exigences.....	12
10. Instructions pour la mise en production.....	13

1. L'Architecture de l'API de réservation



2. Principe de l'architecture

L'architecture de l'API est conçue pour faciliter son développement son évolution, et sa maintenance .

Pour cela l' API Restful est divisé en 3 couches, qui communiquent les unes avec les autres, et aux responsabilités bien distinctes.

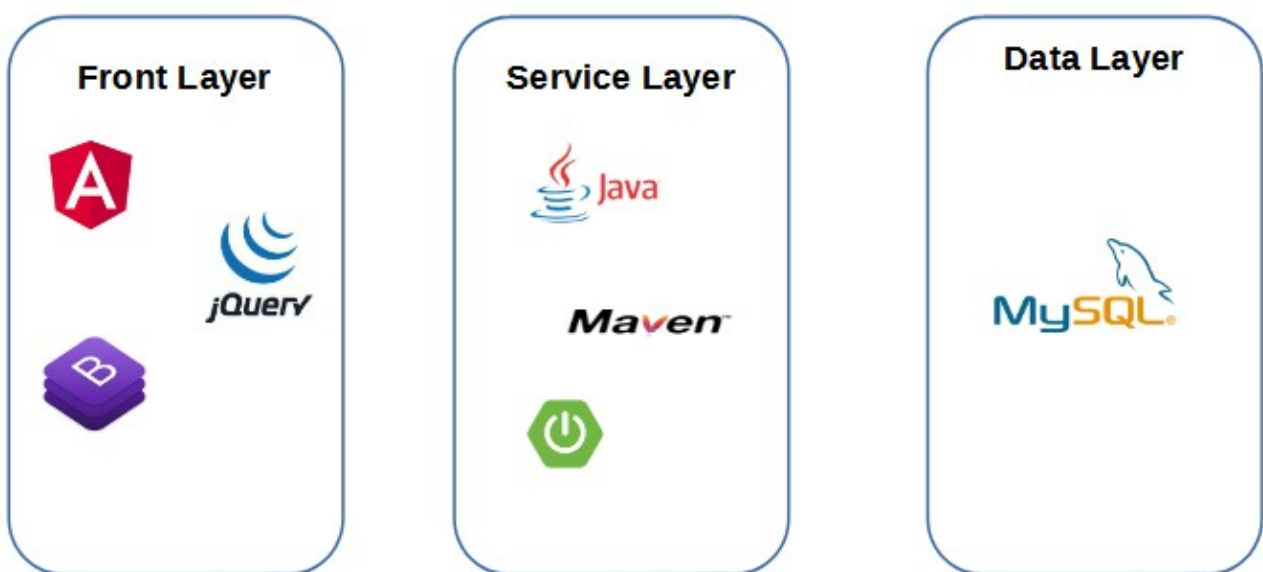
En conformité avec les principes S.O.L.I.D (*Single responsibility ,Open,Liskov substitution, Interface segregation, Dependency inversion*), destiné à produire du code plus compréhensibles, flexibles et maintenables . Elle permettra aussi d'éviter toutes dépendance vis à vis d'une base de données ou d'un quelconque framework front.

Cette API à pour but de communiquer avec une autre API (mis à disposition par le gouvernement) qui met à disposition des adresses disponibles sur le territoire ainsi que les coordonnées GPS de celles-ci.

<https://api.gouv.fr/les-api/base-adresse-nationale>

A la récupération des coordonnées d'une adresse, l'API va retourner l'hôpital stocké en base de données le plus proche de cette adresse et qui répond aux critères saisi par l'utilisateur.

3. Technologies préconisés



Les technologies retenues sont celles autour de **Java**, à savoir les frameworks **Spring Boot et Maven**, qui accélèrent la création d'API Java. **Hibernate** pour l'Objet Relationnel Mapping.

Angular : framework très populaire, avec une communauté importante pour générer des applications front-end. Qui embarque **Bootstrap** et **jQuery**

MySQL : La base de données open source la plus populaire au monde

4. Hypothèses de validation du POC

Le Proof Of concept sera considéré comme validé s'il répond aux hypothèses suivantes :

- ➔ Plus de 90 % des cas d'urgence sont acheminés vers l'hôpital compétent le plus proche du réseau ;
 - a. réception des coordonnées et informations du lieux de l'accident
 - b. Trie des hôpitaux en fonction du lieu de l'accident et de la spécialisation désiré
 - c. Estimation de l'hôpital le plus proche (Calculs à effectuer)
 - ➔ Nous obtenons un temps de réponse de moins de 200 millisecondes avec une charge de travail allant jusqu'à 800 requêtes par seconde, par instance de service ;
 - ➔ Le temps moyen de traitement d'une urgence passe de 18,25 minutes (valeur actuelle) à 12,00 minutes (valeur souhaitée)
 - ➔ Explication des normes à respecter
- Fournir la documentation de l'API avec Swagger
Créer une image pour l'API
- ➔ Instructions pour la mise en production
 - ➔ Mise en œuvre terminée dans les délais impartis
- Planning détaillé de gestion de projet

5. Gestion du projet : Méthode Agile et BDD

La progression du projet est découpée en **Sprint** (2 à 3 semaine)

Le développement se fait suivant l'implémentation de mon Backlog, réaliser dans un premier temps sous **Jira**, puis sous **GitLab**.

Les différents IDE, **Visual Studio Code** pour l'application front-end et **Eclipse** pour l'API Java vont me permettre de développer en parallèle les deux modules.



Projets / POC_MedHead_Reservation

Tableau Sprint 2



6. Outils d'intégration et déploiement continue: GitLab CI

GitLab a été classé n°1 au classement Forrester CI Wave™

Prise en main rapide

GitLab utilise un fichier de configuration écrit en YAML que n'importe quel développeur peut comprendre et prendre en main.

Trois en Un

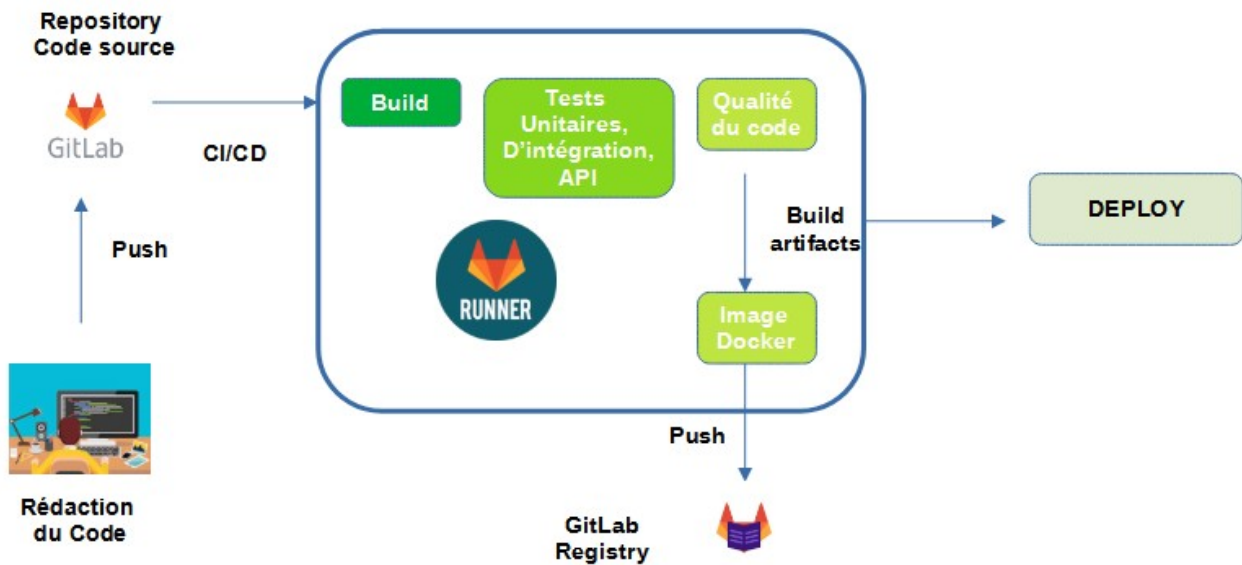
La plateforme tout-en-un de GitLab permet

- de planifier des projets
- de gérer le code source
- de gérer l'intégration et déploiement continue .

Open Source

GitLab est un logiciel open source, qui peut être installé en local ou accessible via internet.

7. Pipeline CI/CD : GitLab CI



GitLab est un logiciel permettant de servir de repository, ainsi les différents modules développés pourront être déposés sur GitLab.

GitLab CI/CD utilise un certain nombre de concepts pour décrire et exécuter le build et déploiement.

A l'aide des runners de GitLab, lors du Push de son code dans le repository approprié par le développeur, des jobs seront automatiquement déclenchés, les uns à la suite des autres.

A la validation des tests (unitaire, intégration et API), un artefact est produit, il permettra de construire une image de l'API en question et permettre par la suite le lancement d'un container Docker. GitLab met à disposition une Registry permettant de stocker les différentes images produites.

8. Stratégie de Test et Plan de Test

Une succession de tests seront réalisés tout au long du projet, par différents intervenant sur le projet.

Les tests unitaires

Qui: les développeurs de l'API

- **Quand:** à la livraison de code sur le repository principal

- **Ou:** sur l'IDE de chaque développeur et dans le pipeline CI/CD mise en place.
- **Pourquoi :** Pour s'assurer du bon fonctionnement et de la qualité du code livré par le développeur.

Les tests API / Les test d'intégrations

- **Qui:** les développeurs
- **Quand:** à la livraison d'une fonctionnalité entière décrite dans les scénarios.
- **Ou:** sur l'environnement de chaque développeur et dans le pipeline CI mise en place.
- **Outils:** Eclipse, Postman
- **Pourquoi:** Assurer la bonne communication entre les composants de la solution.

Les tests utilisateurs de bout en bout

Qui: les utilisateurs ou des testeurs

Quand: régulièrement après chaque livraison

Ou: sur un environnement de test dédié

Pourquoi: S'assurer du bon fonctionnement du système.

Il est préférable de jouer les tests utilisateurs selon la complexité des scénarios, en commençant par les fonctionnalités les plus élémentaires.

Les Test de performance

Qui: des testeurs ou les développeurs

Quand: régulièrement après chaque livraison, ou avant chaque mise en production. Durant les périodes de fort trafic sur l'application.

Ou: sur un environnement de test dédié

Outils: JMeter

Pourquoi: S'assurer que les performances sont maintenues lors de la mise en production.

SCENARIO	Rechercher un hôpital approprié avec un lieu		
FONCTIONNALITÉ	Recherche Hôpital par critère de Lieux	Réf	01_HopRech_Lieu
Description	L'utilisateur à la possibilité de rechercher un hôpital uniquement en fonction du lieu de l'accident.		
GIVEN (En tant que) utilisateur de la solution qui cherche un hôpital WHEN (Quand) quand je renseigne le lieu et clique sur rechercher THEN (alors) je peux sélectionner un hôpital parmi ceux trouvés AND (et) effectuer une réservation de lit.			

N° du Test	Fonctionnalité	Scénario	Action utilisateur	Action système	Résultat	Logiciel
01_HopRech_Lieu	Recherche Hôpital par critère de Lieux	L'utilisateur à la possibilité de rechercher un hôpital uniquement en fonction du lieu de l'accident.	Cliquer sur rechercher et renseigner le lieux de l'accident	Recherche		

SCENARIO	Rechercher un hôpital approprié avec une spécialisation		
FONCTIONNALITÉ	Recherche Hôpital par critère de Spécialisation	Réf	02_HopRech_Spe
Description	L'utilisateur à la possibilité de rechercher un hôpital uniquement en fonction d'une spécialisation		
GIVEN (En tant que) utilisateur de la solution qui cherche un hôpital WHEN (Quand) quand je renseigne la spécialisation et clique sur Rechercher THEN (alors) je peux sélectionner un hôpital parmi ceux trouvés AND (et) effectuer une réservation de lit.			

N° du Test	Fonctionnalité	Scénario	Action utilisateur	Action système	Résultat	Logiciel
02_HopRech_Spe	Recherche Hôpital par critère de Spécialisation	L'utilisateur à la possibilité de rechercher un hôpital uniquement en fonction d'une spécialisation	Cliquer sur rechercher et renseigner la spécialisation de l'accident	Recherche		

SCENARIO	Rechercher un hôpital avec une spécialisation et un lieu		
FONCTIONNALITÉ	Recherche Hôpital par Spécialisation et lieux	Réf	02_HopRech_SpeLieu
Description	L'utilisateur à la possibilité de rechercher un hôpital en fonction d'une spécialisation et d'un lieu donné		
GIVEN (En tant que) utilisateur de la solution qui cherche un hôpital WHEN (Quand) quand je renseigne la spécialisation et le lieu et clique sur Rechercher THEN (alors) je peux sélectionner un hôpital parmi ceux trouvés AND (et) effectuer une réservation de lit.			

N° du Test	Fonctionnalité	Scénario	Action utilisateur	Action système	Résultat	Logiciel
02_HopRech_SpeLieu	Recherche Hôpital par critère de Spécialisation et de lieu	L'utilisateur à la possibilité de rechercher un hôpital en fonction d'une spécialisation et d'un lieu donné	Cliquer sur rechercher après avoir renseigné la spécialisation et le lieu de l'accident	Recherche		

SCENARIO	Faire un choix parmi les hôpitaux trouvés		
FONCTIONNALITÉ	Pouvoir choisir un hôpital adapté	Réf	02_HopCh_01
Description	L'utilisateur à la possibilité de choisir un hôpital adapté après sa recherche		
GIVEN (En tant que) utilisateur de la solution a lancé une recherche WHEN (Quand) quand je clique sur Rechercher THEN (alors) je peux apercevoir pour chaque hôpital les informations - Nombre de lit disponible AND (et) effectuer un choix			

N° du Test	Fonctionnalité	Scénario	Action utilisateur	Action système	Résultat	Logiciel
02_HopCh_01	Pouvoir choisir un hôpital adapté	L'utilisateur à la possibilité de choisir un hôpital adapté après sa recherche	Cliquer sur rechercher et aperçois le nombre de lit disponible pour chaque hôpital	Recherche		

SCENARIO	Créer une réservation de lit(s)		
FONCTIONNALITÉ	Créer une réservation de lit	Réf	02_Réservation_01
Description	L'utilisateur à la possibilité de soumettre une réservation de lit pour un hôpital donné		
GIVEN (En tant que) utilisateur de la solution qui a trouvé un hôpital approprié WHEN (Quand) quand je sélectionne l'hôpital approprié THEN (alors) je peux emmètre une réservation de lit AND (et) retrouver cette réservation			

N° du Test	Fonctionnalité	Scénario	Action utilisateur	Action système	Résultat	Logiciel
02_Réservation_01	Créer une réservation de lit	L'utilisateur à la possibilité de soumettre une réservation de lit pour un hôpital donné	Cliquer sur réserver et valide ma réservation	Recherche		

SCENARIO	MODIFIER UNE RÉSERVATION		
FONCTIONNALITÉ	Modifier une réservation	Réf	02_Réservation_02
Description	L'utilisateur à la possibilité de modifier une réservation		
GIVEN (En tant que) utilisateur de la solution qui a créer une réservation WHEN (Quand) quand j'ai commis une erreur sur la réservation THEN (alors) je peux modifier ma réservation AND (et) la soumettre de nouveau			

N° du Test	Fonctionnalité	Scénario	Action utilisateur	Action système	Résultat	Logiciel
02_Réservation_02	Modifier une réservation	L'utilisateur à la possibilité de modifier une réservation	Cliquer sur la réservation puis la modifier	Modifier		

9. Validation des exigences

Fournir une API Restful qui tient informé en temps réel les intervenants médicaux sur l'hôpital ou se rendre et ce qu'ils doivent faire.

- Le POC fournit une API Restful Java qui permet de trouver un hôpital en fonction de deux caractéristiques que sont le lieu de l'accident et une spécialité recherché.

S'assurer que les données du patient sont correctement protégés.

- Implémentation d'un module d'authentification JWT avec autorisation (Gestion des rôles , en cours)
- Utilisation d'un O.R.M
- Suivi en partie des recommandations de OWASP

S'assurer que le POC est entièrement validé avec des tests reflétant la pyramide de test.

- Mise en place des tests unitaires, d'intégrations, des tests utilisateurs et de performance.
- Couverture de code > 80%

S'assurer que le POC peut être facilement intégré dans le développement futur.

- Utilisation des patterns et bonne pratique durant le développement (pattern DAO, pattern Repository, pattern IOC)
- Possibilité de containeriser l'API

Rendre le code facilement partageable au travers de pipeline d'intégration et livraison continue.

- Repository rendu public accessible en ligne

Documenter la stratégie de test.

- Document de stratégie de test ci-joint

Instruction pour la mise en production doivent être fournies.

- Instruction pour la mise en production ci-joint

S'assurer que la mise en œuvre du POC est terminée dans les délais impartis.

- Estimation des charges de travail documenté, projet découpé en tâches estimables. Planning provisoire fournit

10. Instructions pour la mise en production

Le déploiement de l'application vers l'environnement de production ne sera jamais automatisé, et se fera après validation des différentes hypothèses ci dessous.

Objectifs	Valeur à obtenir	Résultat obtenu après test
Acheminement vers l'hôpital compétent le plus proche.	<i>Pouvoir acheminer 90% des demandes vers les services appropriés</i>	90 %
Temps moyen de traitement d'une urgence	< 12 min	9,5 min
Temps de réponse	< 200 ms	188 ms

Taux de couverture des Tests unitaires	>80 %	86%
Taux de réussite des Tests API	100%	100 %
Réalisation des Tests de performance	OUI	OUI
Tests unitaires en échec	0	0
Réalisation des tests Utilisateurs	75%	75%
Fonctionnalités attendues	85%	85%