|  |
| --- |
| Concepteur Développeur d’application |
| Dossier de projet |

Une image contenant Graphique, Police, graphisme, capture d’écran

Description générée automatiquement

PAC Pierre

CDA – Titre de niveau 6

Février 2023 – octobre 2023

Table des matières

[I. Remercîments 2](#_Toc144968768)

[II. Entreprise 3](#_Toc144968769)

[A. Présentation de l’entreprise 3](#_Toc144968770)

[III. Mise en place du projet 4](#_Toc144968771)

[B. Résumé du projet 4](#_Toc144968772)

[C. Compétences couvertes par le projet 5](#_Toc144968773)

[D. Expressions des besoins 6](#_Toc144968774)

[E. Gestion de projet 6](#_Toc144968775)

[F. Spécifications fonctionnelles 6](#_Toc144968776)

[G. Spécifications techniques 6](#_Toc144968777)

[IV. Réalisations du projet 6](#_Toc144968778)

[V. Liste des compétences 6](#_Toc144968779)

[VI. Résumé du projet 6](#_Toc144968780)

[VII. Cahier des charges, expression des besoins, ou spécifications fonctionnelles du projet 6](#_Toc144968781)

[VIII. Spécifications fonctionnelles du projet 6](#_Toc144968782)

[H. Langages, technologies, méthodes et logiciel utilisés : 6](#_Toc144968783)

[IX. Concevoir et développer des composants d'interface utilisateur en intégrant les recommandations de sécurité 7](#_Toc144968784)

[I. Maquetter une application 7](#_Toc144968785)

[J. Développer une interface utilisateur de type desktop 7](#_Toc144968786)

[K. Développer des composants d’accès aux données 7](#_Toc144968787)

[L. Développer la partie frontend d’une interface utilisateur web 8](#_Toc144968788)

[M. Développer la partie backend d’une interface utilisateur web 8](#_Toc144968789)

[X. Annexe 8](#_Toc144968790)

# Remercîments

Je tiens à exprimer ma profonde gratitude à toutes les personnes qui ont contribué à faire de mon stage au Mipih une expérience inestimable et enrichissante. Cette opportunité a été une étape clé dans mon parcours professionnel et académique, et je suis sincèrement reconnaissant pour l'accueil et le soutien que j'ai reçu.

Tout d'abord, un grand merci à l'entreprise Mipih pour m'avoir offert cette précieuse opportunité de stage. Cette expérience m'a permis de me plonger dans un environnement professionnel stimulant et formateur, qui a significativement élargi mes compétences et ma vision du monde du travail dans le secteur de la santé numérique.

Un immense merci à Wilfried Barreaux, Manager en "Construction Gestion du Patrimoine Applicatif RH & GTA", pour m'avoir confié ce projet et m'avoir fait confiance dès le début. Son soutien continu et son professionnalisme exemplaire ont joué un rôle clé dans mon développement professionnel au fil de ce stage. Sa bienveillance a été un pilier de mon évolution et de ma réussite au sein de l'équipe.

Je souhaite aussi exprimer toute ma gratitude à Loïc Kebouët, Nicolas NOWINSKI et Gabriel Roget pour leur précieux soutien technique. Grâce à leur savoir-faire et à leur disponibilité pour répondre à mes questions, j'ai pu approfondir mes compétences et élargir mon horizon professionnel. Leur contribution a véritablement enrichi la qualité de mon apprentissage et de mon expérience globale au sein de l'entreprise.

Je souhaite également souligner l'accueil chaleureux et l'ambiance positive apportés par toute l'équipe de Prodige 2 : Johan Moreaux, le Product Manager, Claudia Mercier, la Product Owner, ainsi que Amaury Mozin, Cédric Wargnier, Grégory Dujardin et Jean Widcoq. Votre esprit d'équipe et votre ouverture m'ont permis de me sentir intégrée et valorisée dès le premier jour.

Enfin, un merci spécial aux équipes de Prodige 1 et d'Agir pour leur accueil au sein de l'entreprise. La bienveillance et la collaboration entre les équipes ont créé un environnement de travail des plus agréables.

Cette expérience au Mipih a été pour moi une véritable aventure humaine et professionnelle, et c'est grâce à chacun d'entre vous que cela a été possible. Merci de tout cœur.

# Entreprise

## Présentation de l’entreprise

Le Mipih est une entité de type GIP (Groupement d'intérêt public) qui se focalise sur les technologies de l'information dans le domaine médical. En sa qualité de concepteur et intégrateur de systèmes, il propose des innovations technologiques spécifiques pour les hôpitaux publics. Fondé à Toulouse en France, il s'efforce de répondre à la demande croissante en matière de gestion des systèmes d'information pour les institutions de santé. L'accent est mis sur le développement de solutions personnalisées aux enjeux de la médecine.

L’entreprise du Mipih, dispose de bureaux à Toulouse, Amiens, Bordeaux et Reims et est un acteur clé dans le paysage sanitaire français. Elle développe des solutions sur mesure pour chaque institution, abordant divers aspects les dossiers médicaux, le parcours patients, les ressources humaines et financières.

Reconnu pour son professionnalisme et la qualité de son service client, le Mipih comprend les défis auxquels le secteur de la santé est confronté et vise à améliorer les services de soins et l'efficacité opérationnelle grâce à des applications spécialisées.

En contribuant à la numérisation des services et infrastructures de santé, l'entreprise offre des applications qui optimisent la gestion des centres de santé.

En outre, le Mipih se concentre sur l'amélioration de l'expérience patient en mettant en place des bornes d'accueil qui simplifient et accélèrent le processus d'admission (Annexe N° 1).

Les centres de données situés à Amiens et Toulouse permettent un stockage sécurisé des applications et des données médicales. Le Mipih est certifié comme Hébergeur de données de santé (HDS), attestant ainsi de sa compétence à gérer des informations médicales sensibles (Annexe N° 2).

En perpétuelle évolution, le Mipih ajuste continuellement ses services pour répondre aux exigences changeantes du domaine médical public. Grâce à son savoir-faire et à son engagement, il joue un rôle important dans l'optimisation des institutions de santé publique.

# Mise en place du projet

## Résumé du projet

During my 12-week internship period, which took place from July 17 to October 6, 2023, I had the opportunity to join the "Prodige 2" team within the DOGRH division. This internship was part of the broader context of the Human Resources Development (HRD) division, led by Wilfried Barreaux. The main objective of the project was to modernize the way candidates were evaluated for various roles within the DOGRH division, ranging from internships and apprenticeships to fixed-term and permanent contracts (CDD and CDI).

To achieve this goal, the back-end of the application was developed using the Java programming language and the Spring Boot framework. This choice was motivated by the advantages these technologies offer in terms of scalability and robustness. On the front-end side, Angular was the technology of choice to ensure a reactive and user-friendly interface.

The application had several key components. On one hand, it allowed each candidate to take a multiple-choice test (QCM) to evaluate their skills in relation to the targeted position. On the other hand, it offered a programming test to assess technical skills. In addition to candidate-oriented features, the application included a comprehensive administrator interface. This offered recruitment managers the ability to manage the question repository, to create or modify multiple-choice tests, to track applications, and to review results, all within a secure environment.

Particular attention was given to the design and ergonomics of the user interface. We placed special emphasis on creating an interface that was not only pleasant to look at but also intuitive to use. In terms of security, several measures were put in place, including two-factor authentication and password encryption, to ensure that sensitive information was handled with the utmost care.

At the end of my internship, the application was not only functional but also well-received by the DOGRH department. It succeeded in streamlining and improving the efficiency of the evaluation process, thereby significantly reducing the time required to evaluate candidates while simultaneously increasing the quality of recruitment.

## Compétences couvertes par le projet

Chacune des compétences citées ci-dessous, est tirée du « Référentiel Emploi Activités Compétences du titre professionnel Concepteur développeur d’applications ».

1. Concevoir et développer des composants d'interface utilisateur en intégrant les recommandations de sécurité
   1. Maquetter une application
   2. Développer des composants d’accès aux données
   3. Développer la partie frontend d’une interface utilisateur web
   4. Développer la partie backend d’une interface utilisateur web
2. Concevoir et développer la persistance des données en intégrant les recommandations de sécurité
   1. Concevoir une base de données
   2. Mettre en place une base de données
   3. Développer des composants dans le langage d’une base de données
3. Concevoir et développer une application multicouche répartie en intégrant les recommandations de sécurité
   1. Collaborer à la gestion d’un projet informatique et à l’organisation de l’environnement de développement
   2. Concevoir une application
   3. Développer des composants métier
   4. Construire une application organisée en couches
   5. Préparer et exécuter les plans de tests d’une application
   6. Préparer et exécuter le déploiement d’une application

## Expressions des besoins

Le service DOGRH du Mipih souhaite optimiser son processus de recrutement grâce au développement d'un outil en ligne spécialisé. Cet outil aura deux composantes majeures : une interface dédiée aux candidats et une plateforme d'administration pour gérer les différents aspects du processus d'évaluation.

### Interface Candidat

L'objectif est de fournir aux candidats une expérience utilisateur intuitive et efficace. Ainsi, après avoir été préalablement enregistré par un administrateur, le candidat pourra accéder à son espace de test simplement en saisissant son nom et son prénom. Il n'aura pas besoin de s'inscrire ni de créer un compte, ce qui facilite la démarche.

#### Fonctionnalités pour le candidat :

Test de Type QCM Chronométré : Le candidat devra répondre à un QCM dans un temps imparti. Ce test évaluera ses compétences sur des technologies spécifiques comme JAVA, Hibernate ou Angular.

Test de Code : Après le QCM, l'interface proposera un test de programmation. Le candidat pourra coder en temps réel, permettant ainsi d'évaluer ses compétences pratiques.

### Interface Administrateur

La partie administrateur se doit d'être complète et polyvalente pour gérer divers aspects du processus de recrutement.

#### Fonctionnalités pour l’administrateur :

Gestion des Questions : Les administrateurs pourront ajouter, modifier ou supprimer des questions. Ils pourront également les trier par langue/technologie (JAVA, Hibernate, Angular, etc.), par niveau de difficulté et par catégorie (Logique, Algorithme, etc.).

Création et Gestion des QCMs : L'interface permettra de créer des tests QCM en choisissant des questions en fonction de critères tels que la technologie, le niveau et la catégorie. Une fois les paramètres sélectionnés, les questions seront choisies aléatoirement pour constituer le QCM.

Gestion des Sandboxes de Code : Les administrateurs auront la possibilité d'ajouter des tests de code qui sont directement liés à un dépôt GitHub. Cela permet d'évaluer les compétences des candidats dans un environnement de développement réaliste.

Gestion des Entretiens et des Candidats : Les administrateurs pourront enregistrer des candidats, leur associer un QCM et une sandbox de code, et suivre leurs performances. Ils auront accès à des indicateurs comme le pourcentage de réussite au QCM et à la partie de code.

### Sécurité et Authentification Back-End

#### Fonctionnalités de Sécurité :

Authentification : Un processus d'authentification robuste est en place pour les administrateurs. L'authentification à deux facteurs peut être envisagée pour une couche de sécurité supplémentaire.

Autorisation : Une fois authentifié, le système contrôlera le niveau d'accès de l'utilisateur pour déterminer les fonctionnalités auxquelles il peut accéder.

Cryptage des Mots de Passe : Conformément aux meilleures pratiques de sécurité, seuls les mots de passe seront cryptés avant leur stockage dans la base de données. Ce cryptage utilise des algorithmes modernes pour assurer la confidentialité et l'intégrité des mots de passe.

#### Gestion des Données et Logique Métier

Une fois les données collectées via l'interface front-end, elles doivent être transmises de manière sécurisée vers la base de données. Le back-end joue ici un rôle crucial en appliquant la logique métier appropriée.

Fonctionnalités de Gestion des Données :

Validation des Données : Avant l'enregistrement dans la base de données, le système validera les données pour s'assurer qu'elles respectent les critères prédéfinis, tels que le format des questions, le niveau de difficulté, etc.

Stockage des Résultats : Le back-end stockera les résultats des tests des candidats d'une manière structurée, permettant des analyses détaillées ultérieures.

Indicateurs de Performance : Le système calculera automatiquement divers indicateurs, tels que les pourcentages de réussite, pour aider les administrateurs à évaluer les performances des candidats plus efficacement.

#### Bonnes Pratiques de Codage en Java

La qualité du code est une priorité. Le développement du back-end respectera les bonnes pratiques de codage en Java, comme l'encapsulation, l'héritage, et le polymorphisme, pour assurer un code propre, modulaire et maintenable.

Respect des Principes SOLID : Ces principes de la programmation orientée objet seront suivis pour garantir un code de haute qualité et facile à gérer. (Annexe N° 3)

Tests Unitaires : Une série de tests unitaires sera implémentée pour s'assurer que chaque composant du système fonctionne comme prévu.

## Gestion de projet

Pendant mon stage de trois mois au Mipih, j'ai eu la chance d'être intégré dans une équipe qui suit les principes de la méthodologie agile SCRUM. Cette expérience m'a permis de constater combien cette approche favorise la flexibilité, la réactivité et la collaboration entre les membres de l'équipe. Ce fut surtout utile dans la gestion de projets qui nécessitent une adaptation constante et rapide aux changements.

Au sein de ce cadre agile, j'ai travaillé sur un projet qui avait deux objectifs principaux. Le premier était le développement d'un test sous forme de questionnaire à choix multiples (QCM). Le second objectif était l'intégration d'une section dédiée au codage. Afin d'aligner les efforts de l'équipe et de maintenir un rythme de travail soutenu, nous avons organisé des réunions de planification et de rétrospective à chaque fin de sprint, qui durait habituellement trois semaines. Ces réunions nous ont permis d'évaluer nos progrès et de redéfinir les objectifs pour les étapes suivantes.

Un aspect central de notre routine quotidienne était le stand-up meeting qui avait lieu chaque matin. Ce rituel incarnait parfaitement l'esprit agile en encourageant la communication ouverte et en identifiant rapidement les obstacles potentiels. Lors de ces réunions, chaque membre de l'équipe partageait ce qu'il avait accompli la veille et ce qu'il prévoyait de faire pendant la journée, ce qui nous aidait à aligner nos efforts et à aborder proactivement les défis.

Au sein du Mipih, nous employons divers outils de haute technicité pour optimiser notre flux de travail et respecter les principes agiles qui guident nos projets.

**GitLab :**

Nous nous appuyons sur GitLab en tant que plateforme de gestion de version de code source. Ce choix n'est pas anodin. GitLab ne se limite pas à héberger le code ; il offre une série de fonctionnalités collaboratives essentielles. Par exemple, la possibilité de créer des branches pour chaque version ou sous-version du projet permet à chaque membre de l'équipe de travailler sur des éléments distincts sans crainte de conflits de code. Ce système de branches est particulièrement efficace lors des revues de code, car il donne une vue d'ensemble des modifications apportées, facilitant ainsi le processus de validation et d'intégration.

**Discourse** :

Pour aller au-delà des simples forums de discussion techniques, nous avons mis en place Discourse comme une sorte de "StackOverflow" interne. Ce choix résonne parfaitement avec notre objectif de promouvoir une culture d'apprentissage et de partage. Grâce à Discourse, chaque membre de l'équipe peut non seulement poser des questions techniques mais également contribuer à résoudre les problèmes rencontrés par les autres. L'outil dispose également d'une fonction de recherche puissante, réduisant la redondance des questions et permettant un accès rapide à des solutions antérieurement discutées et validées.

**Microsoft Teams** :

En tant qu'outil de communication principal, Microsoft Teams est notre hub pour toutes les interactions en temps réel. Il offre une solution intégrée pour la messagerie instantanée, la planification des réunions, et même le partage de documents, ce qui élimine le besoin de jongler entre différentes applications. La fonction de statut en temps réel est particulièrement utile pour connaître la disponibilité des membres de l'équipe, ce qui permet d'ajuster notre approche communicationnelle selon que la personne est occupée, en réunion, ou disponible pour une interaction.

En somme, l'emploi combiné de ces outils sophistiqués nous permet de toucher à chaque facette de la gestion de projet, de la planification au développement et à la communication. Cette synergie d'outils a manifestement amélioré notre efficacité et a eu un impact positif sur le développement des solutions que nous proposons au Mipih.

## Spécifications techniques

Pour la réalisation de ce projet, nous avons mis en œuvre une panoplie de technologies modernes et d'outils de développement afin de garantir des normes élevées en termes d'efficacité, de robustesse et d'évolutivité.

Le cœur du système, le backend de l'application, a été développé en utilisant Java 17, une version récente et performante du langage Java. Pour faciliter la création d'une interface de programmation (API) robuste et flexible, nous avons choisi d'intégrer Spring Boot. Cette technologie a été un pilier essentiel pour construire une API REST qui répond de manière efficace aux demandes des utilisateurs.

En matière de gestion de données, nous avons adopté Hibernate, un framework de persistance très puissante. Hibernate simplifie considérablement les interactions avec la base de données, ce qui nous a permis de gagner du temps et de réduire la complexité de notre code. Cela a également contribué à améliorer la performance globale de l'application.

Pour simplifier davantage le développement Java, nous avons fait usage de Lombok, une bibliothèque qui automatise la création de code répétitif tel que les getters, setters et les constructeurs. Cette approche a permis de réduire la verbosité du code et d'améliorer la lisibilité, tout en minimisant le risque d'erreurs humaines.

Le frontend de l'application a été construit en utilisant Angular, une technologie robuste qui s'appuie sur des normes web modernes telles qu'HTML5, SCSS et TypeScript. Pour enrichir l'interface utilisateur et offrir une expérience utilisateur fluide, nous avons intégré la bibliothèque PrimeNG.

Afin de garantir une cohérence totale entre les environnements de développement, de test et de production, nous avons adopté Docker. Cette technologie de conteneurisation nous a permis de mettre en place une approche uniforme pour le déploiement de nos applications et services, éliminant ainsi les problèmes potentiels liés aux différences d'environnement.

Pour la gestion de nos données cruciales, nous avons opté pour PostgreSQL, un système de gestion de base de données robuste, reconnu pour sa capacité à gérer d'énormes volumes de données avec fiabilité. Pour la conception de la base de données, nous avons utilisé l'outil Looping, et pour la création d'interfaces utilisateur intuitives, Whimsical a été notre choix pour le maquettage et la conception.

La gestion globale du projet, y compris le suivi des versions, la gestion des tâches et la mise en place de l'intégration continue, a été assurée par GitLab. Cette plateforme collaborative nous a permis de coordonner efficacement notre équipe de développement.

Nous avons employé des environnements de développement spécifiques pour chaque aspect de notre projet. Le développement frontend s'est déroulé dans l'environnement VSCode, tandis qu'Eclipse a été notre environnement de prédilection pour le développement backend. DBeaver a été notre outil de choix pour visualiser et gérer la base de données PostgreSQL, tandis que Postman nous a aidés à tester et à valider les fonctionnalités de l'API.

Enfin, pour garantir un déploiement sans faille sur les serveurs internes du Mipih, nous avons utilisé Docker, qui a simplifié et rationalisé le processus de mise en production, tout en contribuant à la cohérence de notre infrastructure interne.

# Réalisations du projet

## Concevoir et développer des composants d'interface utilisateur en intégrant les recommandations de sécurité

### Maquetter une application

### Développer des composants d’accès aux données

### Développer la partie frontend d’une interface utilisateur web

### Développer la partie backend d’une interface utilisateur web

## Concevoir et développer la persistance des données en intégrant les recommandations de sécurité

### Concevoir une base de données

### Mettre en place une base de données

### Développer des composants dans le langage d’une base de données

## Concevoir et développer une application multicouche répartie en intégrant les recommandations de sécurité

### Collaborer à la gestion d’un projet informatique et à l’organisation de l’environnement de développement

### Concevoir une application

### Développer des composants métier

### Construire une application organisée en couches

### Préparer et exécuter les plans de tests d’une application

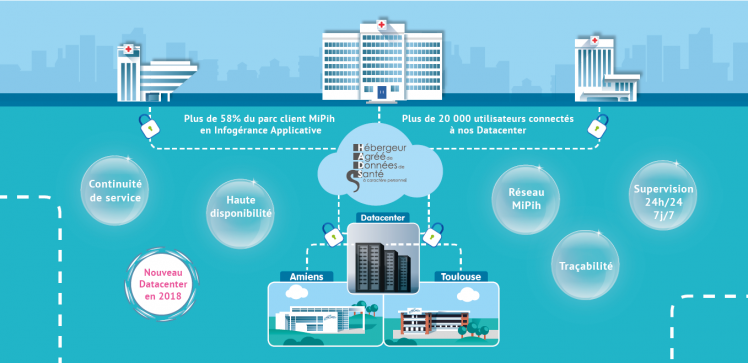
### Préparer et exécuter le déploiement d’une application

# Annexe

Annexe 01 : Bornes admissions patients



Annexe 02 : Datacenter du MiPih



Annexe 03 : Les principes SOLID

Les principes SOLID sont un ensemble de cinq principes de conception de logiciels destinés à rendre les programmes plus efficaces, faciles à comprendre et à maintenir. Ils sont largement reconnus dans le domaine de la programmation orientée objet. Le terme SOLID est un acronyme, où chaque lettre correspond à l'un des principes :

### S - Single Responsibility Principle (Principe de Responsabilité Unique)

Ce principe stipule qu'une classe doit avoir une seule responsabilité ou raison de changer. En d'autres termes, une classe ne doit effectuer qu'une seule "tâche" ou avoir une seule fonctionnalité. Cela simplifie la maintenance et facilite la compréhension du code.

### O - Open/Closed Principle (Principe Ouvert/Fermé)

Ce principe suggère que le code doit être ouvert à l'extension mais fermé à la modification. Cela signifie que vous pouvez ajouter de nouvelles fonctionnalités sans changer le code existant, généralement en utilisant l'héritage et/ou l'implémentation d'interfaces.

### L - Liskov Substitution Principle (Principe de Substitution de Liskov)

Ce principe énonce que les objets d'une classe mère doivent être en mesure d'être remplacés par des objets d'une classe fille sans affecter la correction du programme. Il s'assure que la classe fille respecte le contrat établi par la classe mère.

### I - Interface Segregation Principle (Principe de Ségrégation des Interfaces)

Ce principe suggère que les classes ne doivent pas être forcées de dépendre d'interfaces qu'elles n'utilisent pas. En d'autres termes, une interface devrait avoir uniquement les méthodes qui sont spécifiquement nécessaires pour la classe qui l'implémente.

### D - Dependency Inversion Principle (Principe d'Inversion des Dépendances)

Ce principe encourage à dépendre des abstractions plutôt que des implémentations concrètes. Cela favorise le découplage entre différentes parties d'un système et facilite la maintenance et les tests.