

94440, Villecresnes
nathanieldayas02@gmail.com
23 ans

COMPETENCES OPERATIONNELLES

- Définir des solutions adaptées à la demande
- Modéliser la conception d'applications
- Assurer le développement d'applications
- Assurer la maintenance corrective et évolutives
- Réaliser des tests unitaires et d'intégration
- Veille technologique

OUTILS DE DEVELOPPEMENTS INFORMATIQUES

- MS Visual Studio
- MS Visual Studio Code
- Eclipse
- IntelliJ
- Unity 3D (Jeux vidéo)
- Sonar Qube
- Postman
- Docker Desktop

METHODOLOGIE DE DEVELOPPEMENTS INFORMATIQUES

- Agilité Scrum

LANGAGES INFORMATIQUES

- Java 17 à 21
- C#
- Window Presentation Foundation (WPF)
- Angular 16 et 17
- TypeScript
- HTML / CSS / JavaScript
- Git
- Jsp / Servlet
- JUnit 5
- Log4J
- Spring Core 6
- Spring Boot 3 (Spring Data, Spring MVC)

BASES DE DONNEES

- MySQL

ATOUTS PERSONNELS

- Rigueur dans le travail
- Curiosité

OUTILS BUREAUTIQUES

- MS Excel
- MS PowerPoint MS Word

LANGUES

- Anglais technique

CENTRES D'INTÉRÊT

- Jeux vidéo
- Dessin
- Lecture de BD

Nathaniel DAYAS
Développeur Informatique

Après quelques mois de formation aux langages informatiques (Java, C#) et aux outils de développement (Eclipse, Visual Studio), j'ai acquis de sérieuses connaissances informatiques. Ces dernières m'ont permis d'effectuer mes premières réalisations informatiques.

REALISATIONS INFORMATIQUES

Projet fil rouge Java et Spring Boot en formation Java (voir Détails en deuxième page)

Développements de jeux vidéo

Jeu de hasard dont la règle consiste à obtenir un score de 20 en lançant un dé. Réalisation du jeu sous deux versions :

1. En C#, WPF
2. Avec Unity 3D

Jeu de mémoire dont la règle consiste à trouver le plus grand nombre de paires. Réalisation sous trois versions :

1. En C#, WPF
2. En Java
3. Avec Angular

Jeu de cartes dont la règle consiste à obtenir un score de 20 en choisissant des cartes. Réalisation sous deux versions :

1. En C#, WPF
2. En Java

Développements divers

Calculatrice avec opérations simples réalisée en Angular

EXPERIENCES PROFESSIONNELLES

- Nov. 2019/Juin 2020 - Vendeur stagiaire, DARTY Créteil (94000)
- Fév. 2016 - Agent immobilier stagiaire, ORPI Villecresnes (94440)

FORMATION

Avril à Juin 2024 - Développeur Java, Aston IT (Arcueil 94)

Juin 2021 - BTS Management Commercial Opérationnel (MCO) Lycée Guillaume Budé (Limeil Brévannes 94)

Description

Développement (en groupe) d'une application web sur le thème de la nutrition.

Objectif

Définir des repas qui permettent aux utilisateurs d'atteindre leur objectif en termes de poids.

Fonctionnalités

- Inscription (renseigner les informations)
- Authentification (saisir d'un email et mot de passe, modifier ses informations)
- Consulter la liste des aliments (recherche par nom, par nutriments)
- Consulter la liste des repas (filtre par type de repas)
- Définir un objectif global (définir le type d'objectif)
- Ajouter ses repas quotidien (pour chaque repas)
- Définir un objectif journalier renvoyer la quantité de nutriments pour chaque repas

Technologies utilisées

- Java 17
- Spring Boot 3 (Spring Data, Spring MVC)
- Angular 16
- MySQL
- Docker
- GitHub

L'application est structurée en trois principales couches : présentation, logique métier, et persistance des données.

Frontend (Angular 16)

- **Components** : Pour la création des interfaces utilisateur.
- **Services** : Pour la gestion de la logique métier côté Client et la communication avec le backend.
- **Modules** : Pour une organisation modulaire du code.
- **Communication avec le Backend** : Utilisation d'HTTPClient pour envoyer des requêtes HTTP aux API RESTful.

Backend (Spring Boot)

- **Controllers** : Pour gérer les requêtes HTTP et retourner les réponses appropriées.
- **Services** : Pour encapsuler la logique métier.
- **Repositories** : Pour interagir avec la base de données via Spring Data JPA.

Base de Données (MySQL)

- **Tables** : Utilisateur, Aliment, Objectif, etc.
- **Relations** : Utilisation des clés étrangères pour maintenir les relations entre les tables.

Déploiement et Gestion (Docker)

- **Dockerfiles** : Pour créer des images Docker pour les services frontend et backend.
- **Docker Compose** : Pour orchestrer plusieurs conteneurs et définir les configurations des services