

Ca Mange Oùùù

CMO

[[Build Documentation]](<https://github.com/redarmz/CMO/actions/workflows/documentation.yml/badge.svg>)](<https://github.com/redarmz/CMO/actions/workflows/documentation.yml>) [[Code Coverage with Coveralls]](<https://coveralls.io/repos/github/redarmz/CMO/badge.svg?branch=main>)
[[License]](<https://img.shields.io/github/license/redarmz/CMO.svg?style=flat-square>)](LICENSE)
[[Tag]](<https://img.shields.io/github/tag/redarmz/CMO.svg?label=tag&style=flat-square>)](build.gradle) [[Release]](<https://img.shields.io/github/release/redarmz/CMO.svg?style=flat-square>)](build.gradle) [[(Pre-)Release]]([https://img.shields.io/github/release/redarmz/CMO/all.svg?label=\(pre-\)release&style=flat-square](https://img.shields.io/github/release/redarmz/CMO/all.svg?label=(pre-)release&style=flat-square))](build.gradle)
 [[Code Climate maintainability]](<https://img.shields.io/codeclimate/maintainability/redarmz/CMO?style=flat-square>)](<https://codeclimate.com/github/redarmz/CMO>) [[Codacy Project Certification]](<https://img.shields.io/codacy/grade/9b4c36635f044db0a56f8ee5bedddd167.svg?style=flat-square>)](<https://app.codacy.com/gh/redarmz/CMO/dashboard>) [[Quality Gate Status]](https://sonarcloud.io/api/project_badges/measure?project=redarmz_CMO&metric=alert_status)](https://sonarcloud.io/summary/new_code?id=redarmz_CMO)
[[Coverage]](https://sonarcloud.io/api/project_badges/measure?project=redarmz_CMO&metric=coverage)](https://sonarcloud.io/summary/new_code?id=redarmz_CMO) [[Security Rating]](https://sonarcloud.io/api/project_badges/measure?project=redarmz_CMO&metric=security_rating)](https://sonarcloud.io/summary/new_code?id=redarmz_CMO) [[Technical Debt]](https://sonarcloud.io/api/project_badges/measure?project=redarmz_CMO&metric=sqale_index)](https://sonarcloud.io/summary/new_code?id=redarmz_CMO)
[[Maintainability Rating]](https://sonarcloud.io/api/project_badges/measure?project=redarmz_CMO&metric=sqale_rating)](https://sonarcloud.io/summary/new_code?id=redarmz_CMO) [[Bugs]](https://sonarcloud.io/api/project_badges/measure?project=redarmz_CMO&metric=bugs)](https://sonarcloud.io/summary/new_code?id=redarmz_CMO)
[[Vulnerabilities]](https://sonarcloud.io/api/project_badges/measure?project=redarmz_CMO&metric=vulnerabilities)](https://sonarcloud.io/summary/new_code?id=redarmz_CMO) [[Code Smells]](https://sonarcloud.io/api/project_badges/measure?project=redarmz_CMO&metric=code_smells)](https://sonarcloud.io/summary/new_code?id=redarmz_CMO)

Besoin :

Nous allons manger à la pause mais nous ne savons pas où, notre temps et budget sont limités ...

Solution :

Nous souhaitons créer un site web qui va nous donner la liste des restaurants, fast food,... disponibles aux alentours selon plusieurs critères de sélection (prix, temps de service, régime alimentaire ...) en utilisant la géolocalisation. Plus tard, nous souhaiterions intégrer une IA qui nous pose 3 questions lors de l'entrée sur l'appli telles que

- "Combien de temps as-tu ?",
- "Souhaites tu manger chaud ou froid ?" et qui déterminera une liste bien plus restreinte de restaurants selon la demande. Nous nous connecterons à une BDD externe afin de récolter de multiples infos sur les restaurants. ## Type d'application (mobile, Web, ...) : Application Web

Fonctionnalités :

(3 ou 4, en prévisionnel, cela pourra évoluer en fonction des itérations) : - Gérer les profils des utilisateurs - Se renseigner sur un restaurant - Réserver un restaurant - Noter les restaurant - Créer un espace de chat

Lancement du projet :

Pour le Front-end :

La partie Front-end du projet est réalisé à l'aide du framework Angular. Pour pouvoir lancer le Front-end assurez-vous d'avoir : - Installer le runtime Node, que vous pouvez installer sur "nodejs.org". Fourni avec Node vous avez le package manager "npm" qui vous permettra d'installer le CLI d'Angular. (Si vous êtes sur Mac consulter cette documentation "<https://docs.npmjs.com/resolving-eacces-permissions-errors-when-installing-packages-globally>" pour résoudre les éventuels erreurs de permission) - Installer le CLI d'Angular avec la commande "npm i -g @angular/cli" (pour vérifier la version d'Angular taper la commande "ng v") - Tapez la commande "npm install" avant de lancer le serveur - Enfin vous pouvez lancer le Front-end en tapant la commande "ng serve", un lien vous sera donner (<http://localhost:4200>) ouvrir ce lien sur votre navigateur (!! Assurez-vous d'avoir le port 4200 de libre, sans cela vous ne pourrez pas accéder au données du back-end).

Pour le Back-end :

La partie Back-end du projet est réalisé à l'aide de Node.js et de différents frameworks tels que Express.js. Pour pouvoir lancer le Back-end assurez-vous d'avoir : - Installer le runtime Node si ce n'est pas déjà fait. - Installer l'outil nodemon avec la commande "npm install -g nodemon" (Assurez-vous que le port 3000 soit libre) - Enfin vous pouvez lancer le server back-end avec la commande "nodemon server"

Infos :

!!! Lancez le serveur back-end avant de lancer le front-end pour que les données soit charger en avant !!!

Fichiers

Matrice Contribution en Python

Contributeur	Commits	Lignes ajoutées	Lignes supprimées	Back-end	Front-end	Principal
wassim.hendali2@gmail.com	13	71016	122	2772	253	67869
146457330+amiirmmr@users.noreply.github.com	12	216	11	0	0	205
reda.ramzi0104@hotmail.com	51	601132	2904	569744	28465	19
mammeriamir4@gmail.com	20	508263	6153	67722	1568	432820
119413560+redarmz@users.noreply.github.com	8	231265	45	230265	469	486
113892426+WassimH94@users.noreply.github.com	5	10	10	0	0	0

Matrice Contribution en Ruby

Contributeur	Commits	Lignes ajoutées	Lignes supprimées	Back-end	Front-end	Principal
wassim.hendali2@gmail.com	13	71016	122	2772	253	67869
146457330+amiirmmr@users.noreply.github.com	12	216	11	0	0	205
reda.ramzi0104@hotmail.com	50	601132	2904	569744	28465	19
mammeriamir4@gmail.com	20	508263	6153	67722	1568	432820
119413560+redarmz@users.noreply.github.com	8	231265	45	230265	469	486
113892426+WassimH94@users.noreply.github.com	5	10	10	0	0	0

MatriceFeatures

	Mise en place du système d'événement	Création du système de tirelire	Mise en relation d'événement et de la tirelire	Ajout d'un système de réservation	Création de salon de discussion dédié à chaque événements	Mise en place d'un système de log pour accéder au site et un système d'inscription
Amir		X	X		X	
Wassim	X			X		X
Reda	X		X	X		X

	Affichage de tous les événements avec nom de l'hôte	Ajout d'un API carte pour afficher les restaurants et leurs itinéraire	Recherche et affichage des événements selon le nom	Suivi des dettes pour l'option tirelire	Numéro d'événement créer automatiquement pour la création événement	Amélioration du design de l'application
Amir		X		X		X
Wassim	X		X			X
Reda	X		X		X	X

Personas :

1. **Le Travailleur Pressé :**
 - **Âge :** 25-40 ans
 - **Profession :** Employé de bureau avec un emploi du temps chargé
 - **Besoin principal :** Trouver rapidement un lieu pour manger, qui respecte son temps et son budget limités.
 - **Comportement :** Utilise son temps de pause de manière efficace, préfère les commandes à emporter ou les repas rapides mais sains.
2. **L'Étudiant :**
 - **Âge :** 18-24 ans
 - **Profession :** Étudiant universitaire
 - **Besoin principal :** Trouver des options de repas abordables et rapides, adaptées à un régime alimentaire spécifique (végétarien, vegan, etc.).
 - **Comportement :** Cherche souvent à manger en groupe, ouvert à essayer de nouveaux lieux, mais limité par le budget.
3. **Le Touriste :**
 - **Âge :** Varie
 - **Profession :** Touriste explorant la ville
 - **Besoin principal :** Découvrir des restaurants locaux avec des options de repas typiques de la région.
 - **Comportement :** Disponible pour explorer, plus flexible sur le temps et le budget, intéressé par l'expérience culinaire plus que la rapidité.

Scenarios :

Trouver un Repas Rapide :

- **Persona** : Le Travailleur Pressé
- **Scénario** : Durant une pause de midi limitée à 30 minutes, le travailleur cherche un restaurant qui peut servir un repas chaud en moins de 10 minutes à une courte distance de son bureau.
- **Critères** : Temps de service, distance, options de repas chauds.

Trouver un Café pour une Pause-Café Rapide :

- **Persona** : Le Travailleur Pressé
- **Scénario** : Thomas a une journée chargée au bureau et cherche un café où il peut rapidement acheter un café et une collation saine sans faire la queue, tout en étant à moins de 5 minutes à pied.
- **Critères** : Temps de service rapide, distance de marche, options de collations saines.

Planification d'un Dîner entre Étudiants :

- **Persona** : L'Étudiant
- **Scénario** : Un groupe d'étudiants souhaite trouver un restaurant qui peut accueillir six personnes, offrant des options végétariennes et dans une fourchette de prix abordable.
- **Critères** : Capacité, options végétariennes, gamme de prix.

Organiser une Soirée Étude de Groupe dans un Café :

- **Persona** : L'Étudiant
- **Scénario** : Léa et son groupe d'étude cherchent un café calme ouvert tard le soir, offrant des prises de courant et une connexion Wi-Fi fiable, pour préparer leur projet final ensemble.
- **Critères** : Heures d'ouverture tardives, ambiance calme, disponibilité de prises électriques, connexion Wi-Fi.

Exploration Culinaire pour Touristes :

- **Persona** : Le Touriste
- **Scénario** : Un couple en vacances cherche un restaurant offrant une expérience culinaire authentique locale, avec la possibilité de réserver à l'avance.
- **Critères** : Authenticité, possibilité de réservation, recommandations.

Découverte de Cuisine Locale pour le Dîner :

- **Persona** : Le Touriste
- **Scénario** : Marco et sa compagne, en visite dans une nouvelle ville, cherchent un restaurant local offrant une expérience culinaire authentique. Ils préfèrent un établissement qui sert des plats typiques de la région, recommandé par les locaux, avec la possibilité de réserver une table en terrasse.
- **Critères** : Authenticité des plats locaux, recommandations des locaux, option de réservation, disponibilité d'une terrasse.

Connexion au Compte Utilisateur

- **Persona** : Tous les utilisateurs
- **Scénario** : L'utilisateur souhaite accéder à son espace personnel sur l'application CMO pour consulter ou utiliser les services disponibles.
- **Critères** : Interface de connexion simple, option de récupération du mot de passe, connexion sécurisée.

Modifier son Profil

- **Persona** : Tous les utilisateurs
- **Scénario** : Après avoir expérimenté l'application, l'utilisateur souhaite mettre à jour ses préférences alimentaires et son budget moyen pour des recommandations plus précises.
- **Critères** : Accès facile aux paramètres du compte, interface intuitive pour la modification des informations, sauvegarde instantanée des modifications.

Changer de Mot de Passe

- **Persona** : Tous les utilisateurs
- **Scénario** : Pour des raisons de sécurité, l'utilisateur décide de changer son mot de passe.
- **Critères** : Processus de changement de mot de passe sécurisé, instructions claires.

Se Déconnecter

- **Persona** : Tous les utilisateurs
- **Scénario** : L'utilisateur souhaite se déconnecter de l'application pour sécuriser son compte lorsqu'il n'utilise pas l'appareil.
- **Critères** : Option de déconnexion accessible facilement, déconnexion rapide.

Réserver un Restaurant

- **Persona** : Tous les utilisateurs
- **Scénario** : L'utilisateur a trouvé un restaurant qui répond à ses critères et souhaite faire une réservation directement via l'application.
- **Critères** : Processus de réservation simple et rapide, confirmation immédiate dans l'application, possibilité d'ajouter la réservation à son calendrier.

Déclarer un Incident lors d'une Réservation

- **Persona** : Tous les utilisateurs
- **Scénario** : L'utilisateur rencontre un problème avec sa réservation (par exemple, le restaurant ne trouve pas sa réservation) et souhaite le signaler.
- **Critères** : Accès rapide au service client via l'application, formulaire simple pour décrire l'incident, suivi du dossier en temps réel.