

PROJET

Gestionnaire de Restaurant



Sommaire

1.	Fonctionnalités développées.....	1
1.1.	Fonctionnalités minimales.....	1
1.2.	Fonctionnalités supplémentaires / bonus	2
2.	Architecture de l'application	2
3.	Ergonomie de l'application.....	5

1. Fonctionnalités développées

1.1. Fonctionnalités minimales

Gestion des menus :

- ✓ **1.1** Visualiser la liste des menus
- ✓ **1.2** Visualiser le détail d'un menu (avec la liste de ses plats)
- ✓ **1.3** Créer un menu
- ✓ **1.4** Modifier un menu
- ✓ **1.5** Supprimer un menu

Gestion des plats :

- ✓ **2.1** Ajouter un plat dans un menu
- ✓ **2.2** Modifier un plat dans un menu
- ✓ **2.3** Supprimer un plat d'un menu

1.2. Fonctionnalités supplémentaires / bonus

Gestion des menus :

- ✓ **1.6** Visualiser le nombre total de calories d'un menu
- ✓ **1.7** Visualiser le nombre moyen de calories par plats dans un menu
- ✓ **1.8** Filtrer les menus selon leur statut (Actif / Inactif)
- ✓ **1.9** Filtrer les menus selon leur date de création
- ✓ **1.10** Filtrer les menus par mots clés

Gestion des plats :

- ✓ **2.4** Utiliser des modèles de plats (plats prédéfinis) lors de l'ajout d'un plat dans un menu avec suggestion selon la saisie
- ✓ **2.5** Filtrer les modèles de plats par mots clés

Gestion des commandes :

- ✓ **3.1** Visualiser la liste des commandes servies et non servies
- ✓ **3.2** Visualiser le statut d'une commande (servie / non servie), le statut des plats qui la composent et l'heure à laquelle elle a été enregistrée
- ✓ **3.3** Ajouter une commande à partir d'un menu actif
- ✓ **3.4** Marquer un plat comme servi dans une commande
- ✓ **3.5** Marquer une commande entière comme servie
- ✓ **3.6** Clôturer une commande servie
- ✓ **3.7** Effacer / nettoyer la liste des commandes à servir
- ✓ **3.8** Effacer / nettoyer la liste des commandes servies

Accessibilité :

- ✓ **4.1** Sélectionner la langue de l'application (Français ou Anglais)

2. Architecture de l'application

Cette section décrit le contenu du répertoire **src/app** de l'application.

constants/ : Constantes.

- **food-icons.const.ts** : Déclare une liste d'icônes Font Awesome représentant de la nourriture et utilisée pour décorer le Hero Header (Voir composant **hero/**).

content/ : Contenu des pages de l'application.

- **dish/** : Composants relatifs aux plats.
 - **dish-item/** : Un plat dans une liste, cf. **1.2** **2.1** **2.2** **2.3**.
 - **dish-list/** : La liste des plats d'un menu, cf. **1.2** **1.6** **1.7** **2.1** **2.2** **2.3**.
- **dish-order/** : Composants relatifs aux plats dans une commande.
 - **dish-order-item/** : Un plat dans une commande de menus, cf. **3.1** **2.2** **3.4**.
 - **dish-order-list/** : La liste des plats dans une commande, cf. **3.1** **3.4**.
- **dish-template/** : Composants relatifs aux modèles de plats, cf. **2.4** **2.5**.
 - **dish-template-list/** : La liste des modèles de plats.
 - **dish-template-suggestion/** : Un modèle de plat (suggestion) dans une liste.
- **hero/** : Hero Header visible sur la page d'accueil de l'application.
- **menu/** : Composants relatifs aux menus.
 - **menu-edition-form/** : Formulaire d'édition d'un menu, cf. **1.3** **1.4**.
 - **menu-item/** : Un menu dans une liste, cf. **1.1** **1.4** **1.5** **1.8** **1.9** **1.10**.
 - **menu-list/** : La liste des menus, cf. **1.1** **1.3** **1.8** **1.9** **1.10**.
 - **menu-status-badge/** : Un badge indiquant le statut d'un menu, cf. **1.1** **1.2**.
 - **menu-view/** : Page du détail d'un menu, cf. **1.2** **1.4** **1.5** **1.6** **1.7**.
- **menu-order/** : Composants relatifs aux commandes de menus.
 - **menu-order-id-badge/** : Un badge de mise en forme du numéro d'une commande, cf. **3.1**.
 - **menu-order-item/** : Une commande dans une liste, cf. **3.2** **3.5** **3.6**.
 - **menu-order-list/** : La liste des commandes, cf. **3.1** **3.7** **3.8**.
- **order/** : Composants relatifs aux commandes.
 - **add-order-form/** : Formulaire d'ajout d'une commande, cf. **3.3**.
 - **order-status-badge/** : Badge indiquant le statut d'une commande, cf. **3.2**.

dialog/ : Fenêtres / boîtes de dialogue.

- **confirmation-dialog/** : Boîte de dialogue demandant une confirmation, cf. **1.5**.

directive/ : Directives TypeScript utilisées pour l'affichage d'éléments.

- **auto-width.directive.ts** : Calcule, dans une liste d'items, la largeur d'une colonne en fonction de son contenu, cf. **1.1**.

enum/ : Classes d'énumération de types.

- **dialog-state.ts** : Énumère les états d'un dialogue (En attente, Validé, ...), cf. **1.5**.
- **language.ts** : Énumère les langues disponibles (Français, Anglais), cf. **4.1**.
- **loading-status.ts** : Énumère les statuts de chargement (En cours, chargé, ...), utilisée à l'initialisation des pages nécessitant une récupération de données depuis la base de données.
- **menu-status.ts** : Énumère les statuts d'un menu (Actif, Inactif), cf. **1**.
- **order-status.ts** : Énumère les statuts d'une commande (Servie, Non servie), cf. **3**.
- **processing.ts** : Énumère les types de traitements (Ajout, Modification, Suppression), cf. **1.3 1.4 1.5 2.1 2.2 2.3 3.3**.
- **processing-status.ts** : Énumère les statuts d'un traitement (En cours, Traité, Échoué), cf. **1.3 1.4 1.5 2.1 2.2 2.3 3.3**.
- **view-mode** : Énumère les modes d'une vue (Visualisation, Édition, ...), utilisée pour des vues multifonction, cf. **1.3 1.4 1.5 2.1 2.2 2.3**.

include/ : Éléments de contenu récurrents.

- **footer/** : Pied des pages de l'application.
- **header/** : Entête des pages de l'application contenant le logo du restaurant, un menu de navigation et les options de langues, cf. **4.1**.

model/ : Classes métier.

- **dish-order.ts** : Représente un plat dans une commande de menu, cf. **3**.
- **dish.ts** : Représente un plat, cf. **2**.
- **menu-order.ts** : Représente une commande de menu, cf. **3**.
- **menu.ts** : Représente un menu, cf. **1 3.3**.

pipe/ : *Pipes* personnalisés utilisés dans les *templates*.

- **formatted-date.pipe.ts** : Formate une date au format JJ/MM/AAAA.
- **split-for-highlight.pipe.ts** : Divise une chaîne de caractères en sections pour permettre une mise en évidence de certains caractères, cf. **1.10 2.5**.
- **translate.pipe.ts** : Traduis un contenu dans la langue de l'application (retourne la traduction associée à une clé JSON), cf. **4.1**.

pop-up/ : Pop-ups.

- **loading-pop-up/** : Affiché lorsque des données sont en cours de récupération depuis la base de données, principalement à l'initialisation d'une page, cf. **1.1 1.2 3.3**.

- **processing-pop-up/** : Affiché lorsqu'un traitement de type ajout / modification / suppression est en cours, cf. **1.3** **1.4** **1.5** **2.1** **2.2** **2.3** .

service/ : Services TypeScript.

- **date.service.ts** : Utilisé pour accéder à la date courante (notamment dans le FOOTER).
- **dish.service.ts** : Utilisé pour lire et modifier les données des plats en base de données.
- **filtering.service.ts** : Fournit une méthode de filtre sur des listes de chaînes de caractères avec prise en compte des traductions, cf. **1.10** **2.5** .
- **menu.service.ts** : Utilisé pour lire et modifier les données des menus en base de données.
- **translation.service.ts** : Utiliser pour gérer la langue de l'application et récupérer les traductions à partir de fichiers JSON, cf. **4.1** .

ui/ : Éléments d'interface personnalisés.

- **drop-down-menu/** : Menu déroulant personnalisé, cf. **1.8** **1.9** .

utility/ : Fonctions utilitaires.

- **object-conversion.ts** : Fonctions de conversion d'objet notamment utilisées pour faire correspondre les objets de la base de données avec les classes métiers.
- **text-normalisation.ts** : Fonctions utilitaires de normalisation de chaînes de caractères retirant les accents et les caractères spéciaux, cf. **1.10** **2.5** .

3. Ergonomie de l'application

Cette section décrit les routes de l'application ainsi que les liens entre les différentes pages.

L'en-tête des pages de l'application (*header*) inclut un menu de navigation permettant d'accéder à la page d'Accueil, aux Commandes et aux Menus.

ROUTE	PAGE COURANTE	PAGES ACCESSIBLES
/	Accueil + Commandes	[+ Ajouter une commande] → Nouvelle commande
/menus	Menus	[+ Ajouter un menu] → Nouveau menu
/menu/new	Nouveau menu	[Ajouter] / [Annuler] → Menus
/menu/:id	Détail du menu de numéro id	[Modifier] → Édition du menu [Supprimer] / [Retour] → Menus
/menu/:id/edit	Édition du menu de numéro id	[Enregistrer] / [Annuler] → Détail du menu
/orders	Commandes	[+ Ajouter une commande] → Nouvelle commande
/order/new	Nouvelle commande	[Ajouter] / [Annuler] → Commandes