

Thème : Gestion des commandes d'un restaurant de sushi.

Objectifs de la SAé

Dans le cadre de cette SAé, les objectifs à atteindre sont :

- Développement Front en utilisant un framework
- Développement Back pour la création d'API
- Interaction Front back.

Présentation du contexte

Cette situation repose sur le développement d'une application Front-end avec le framework Angular pour une prise de commande au niveau d'un point de vente de sushis (restauration).

Deux scénarios peuvent coexister :

- L'opérateur prend la commande par téléphone pour une livraison à domicile
- Le client utilise une borne de commande.

On comprend bien pourquoi, étant une application interne, il n'existe pas d'authentification pour l'instant, mais l'application sera évolutive.

Pour ce qui est de la gestion des données, on utilisera au niveau Back une application PHP (avec ou sans framework), qui permettra de créer une API REST présentant la gamme de boxes mises à la vente et qui sera exploitée par la partie front de votre application (sous Angular).

La structure attendue pour l'API sera présentée ci-après.

Les attendus fonctionnels demandés de l'application :

L'application Web doit permettre à un utilisateur de passer commande de boxes de sushis en vue d'établir une commande transmissible à la production (cuisine).

Elle comportera, notamment :

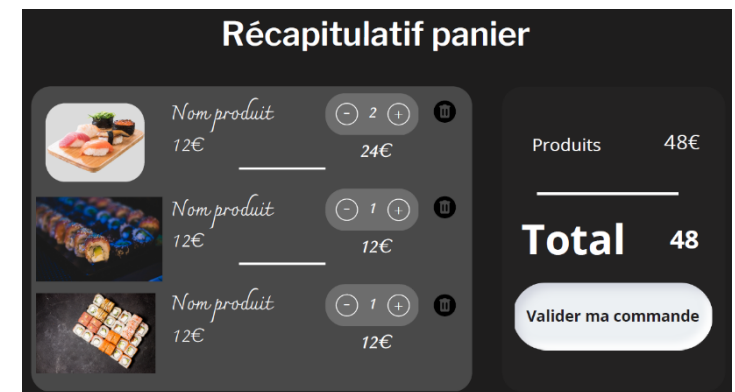
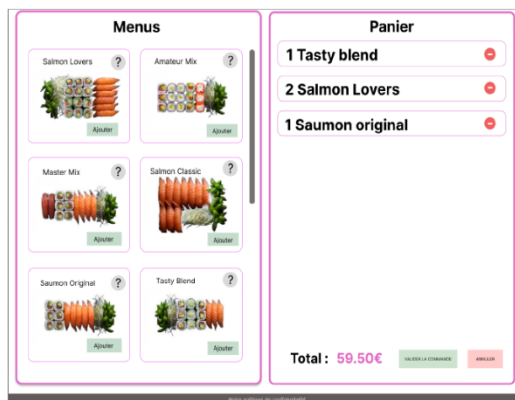
- L'affichage de la liste des boxes Sushi,
- La possibilité de voir le détail d'une box,
- La possibilité de sélectionner une ou plusieurs boxes pour un panier de commande,
- Le calcul du montant total de la commande,
- La visualisation des commandes sauvegardées localement,
- Une page spécifique concernant la mise en place du RGPD (en prévision d'une commande par internet)
- ...

Attentes techniques demandés de l'application :

- Création d'une API REST avec PHP (avec ou sans framework)
- Définition des structures des données métier (classes et diagramme de classes)
- Implémentation des codes de nécessaire pour la création de l'API
- Hébergement de l'API
- Sauvegarde du panier côté client (LocalStorage) lors de sa validation par le client.
- Prise en compte d'au moins deux scénarios malveillants
- Etablir un tableau des actions redoutées
- Proposer des contres mesures de scénarios malveillants (Evil User Stories)
- Tests unitaires - en bonus

Maquette de l'application

Des exemples de maquettes de votre application :



Structure de votre API REST

La structure de l'API REST aura la structure présentée par l'URL suivante :

https://elearning.univ-eiffel.fr/pluginfile.php/523573/mod_resource/content/1/boxes.json

L'API sera créée à partir d'une base de données MySQL. La base de données sera conçue et implémentée dans le cadre de votre application.

Les images sont à récupérer à partir de l'URL : <https://elearning.univ-eiffel.fr/course/view.php?id=8642>

Pour la conception de la base, on utilisera un outil du langage UML, à savoir le diagramme des classes.

Contraintes à prendre en charge

Les contraintes suivantes doivent être respectées dans la réalisation de votre projet :

- Une commande ne doit pas contenir plus de 10 Boxes
- Les images des menus ne doivent pas être stockées dans la base, mais juste le chemin vers l'image doit être enregistré.
- On doit avoir la possibilité d'annuler toute commande en cours.

Liste des livrables à fournir

Livable L401_1 : Dossier d'analyse

- Planification du projet (Trello) à fournir avant toute tâche.
- Répartition des tâches
- Nom et logo de votre application
- Analyse des besoins de l'application, en utilisant un diagramme de "Use Case"
- Réalisation d'un diagramme de collaboration pour présenter les différentes étapes du projet
- Conceptualisation de votre base, en fournissant un diagramme de classes (utiliser des outils comme : draw.io, starUml, ...)
- Architecture fonctionnelle de l'application
- ***Ce livable doit être consultable en ligne via github (README.md)***

Livable : L401_2

- Implémentation de la partie back, en fournissant les différents codes de la création de l'API REST.
- Architecture de l'implémentation de vos différents codes

Livable : L401_3

- Maquettes de l'application : Accueil, panier, détails

Livrable : L401_4

- L'intégralité du code de la partie Front (Sur GitHub ou GitLab avec suivi des mises à jour)

Livrable : L401_5

- Comptes rendus des tests unitaires effectués
- Page pour la prise en compte des règles RGPD liées à l'application
- Document explicitant au moins deux scénarios malveillants et les mesures pour les contrecarrer.

Planification et ressources de la SAé

- Date de lancement : **13/02/2024**
- Date de fin : **20/3/2024 à 23h59**
- Heure CM : 2
- Heures TD : 10
- Heures TP : 4

Modalités de déroulement de cette SAé

- 1) Tout support déposé sur e-learning doit porter les prénoms de toutes les personnes du groupe, sous le format : **prénom1_prénom2_prénom3.***
- 2) Les groupes de cette SAé doivent contenir au mini 3 étudiants et au maxi 4 étudiants.
- 3) La partie gestion de projet aura une place importante dans cette SAé.
- 4) Une date de "review" vous sera communiquée ultérieurement pour faire une première recette. Cette review sera évaluée et représentera 25% de la note finale.

Exemple de contenu du fichier : README.md

- La date, le nom de l'équipe et les participants au projet
- Le lien vers votre dépôt distant
- Un lien vers votre application opérationnelle, déployée sur un serveur en ligne
- Un diagramme des cas d'utilisation
- Un diagramme de classes
- Le champ d'utilisation du RGPD dans ce projet sachant qu'il sera amené à être plus tard en ligne,
- Le format de la structure JSON (interface ts ou structure exemple JSON) des commandes enregistrées dans le LocalStorage.
- Une sélection de copies écran mettant en relation le code source et l'UI, le tout commenté.
- Une conclusion