

SushiFastSQM

Table des matières

Présentation	1
Sommaire	1
Thème	1
Analyse	2
1.1 : Diagramme des cas d'utilisations	2
1.2 : Diagramme des différents tiers	2
1.3 : Maquette de l'interface Web	3
1.4 : Diagramme d'activité possible	4
1.5 : Structure JSON du panier	5
Implémentation	6
2.1 RGPD :	6
2.2 Sélection de copies écran :	6
2.3 Cybersécurité :	9
Conclusion	10
3.1 Résumé de l'objectif :	10
3.2 Résumé des résultats :	10
3.3 Résumé de l'expérience :	10
3.4 Perspectives d'avenir :	10
Lien vers notre dépôt :	10

Présentation

Projet réalisé par Quentin AYRAL, Sekou TRAORE et Mathieu MORGADO MARTA rendu le 2 avril 2023.

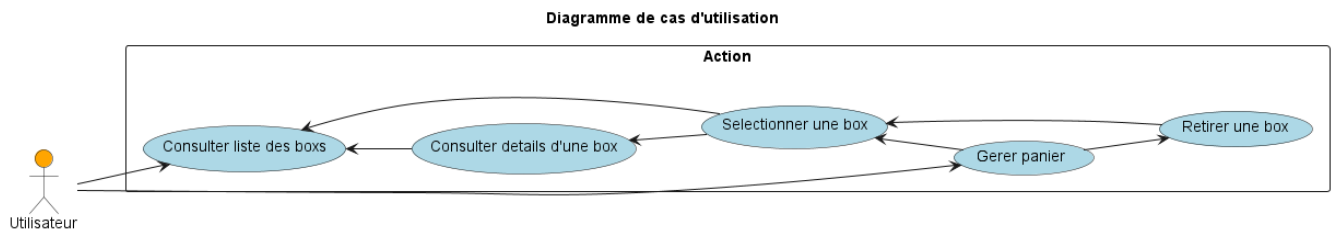
Sommaire

Thème

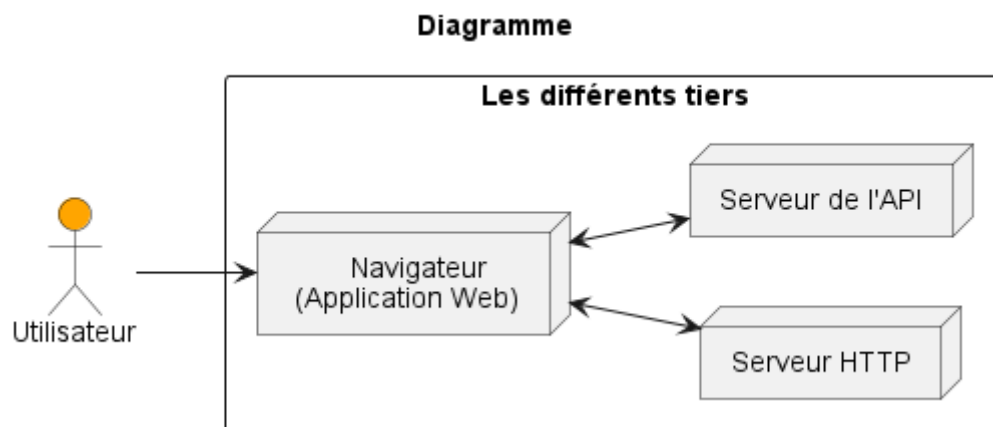
Réaliser une application Web front-end de prise de commande de Menus/Box au niveau d'un point de vente de sushi grâce à une API et le Framework Angular

Analyse

1.1 : Diagramme des cas d'utilisations




1.2 : Diagramme des différents tiers



1.3 : Maquette de l'interface Web


Menus

Salmon Lovers ?




Ajouter

Amateur Mix ?




Ajouter

Master Mix ?




Ajouter

Salmon Classic ?




Ajouter

Saumon Original ?



Ajouter

Tasty Blend ?



Ajouter

Panier

1 Tasty blend

-

2 Salmon Lovers

-

1 Saumon original

-

Total : 59.50€

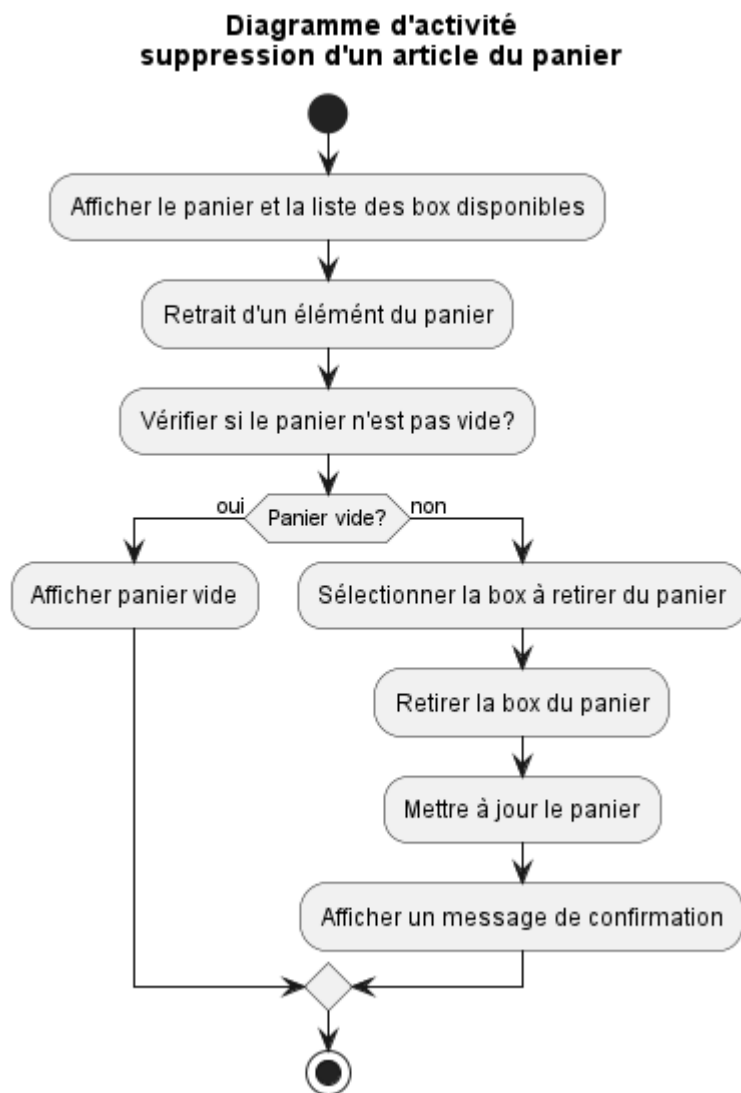
VALIDER LA COMMANDE

ANNULER

[Notre politique de confidentialité](#)

3

1.4 : Diagramme d'activité possible



Le format ci-dessus présente la suppression d'un article du panier.

1.5 : Structure JSON du panier

```
[
  {
    "id": <l'id de la commande>,
    "prix": <prix total de la commande>,
    "nbTotalBox": <nombre total de box>,
    "articles": [
      {
        "nom": "<nom de la box>",
        "quantite": <quantite de box demandé>
      },
      {
        "nom": "<nom de la box>",
        "quantite": <quantite de box demandé>
      }
    ]
  }
]
```

Le format ci-dessus présente un panier contenant 2 box différentes.

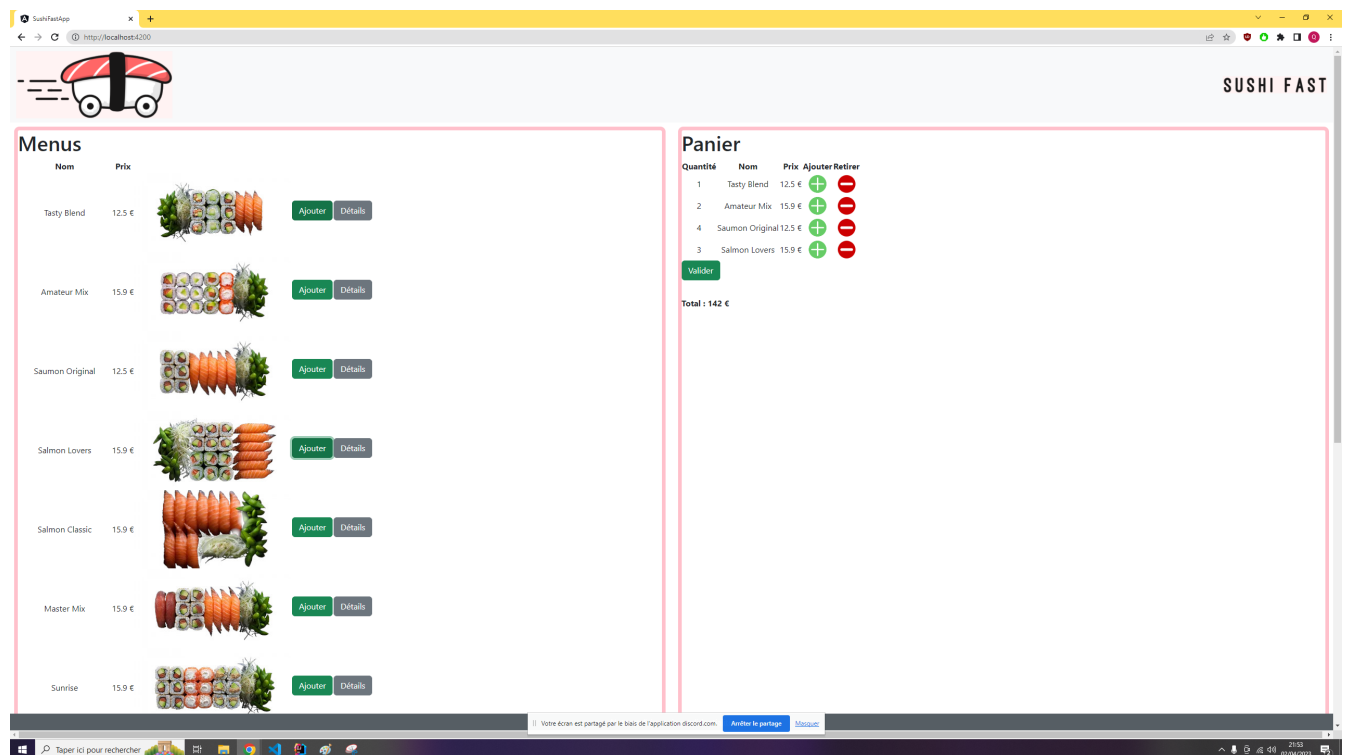
Implémentation

2.1 RGPD :

Nous avons décidé afin d'être le plus transparent possible, de renvoyer l'utilisateur, si ce dernier le souhaite, sur le site de la CNIL via le lien cliquable en bas de page "Notre politique de confidentialité".

2.2 Sélection de copies écran :

Ajout au panier :



Les boutons "Ajouter" visible sur la partie gauche de l'interface renvoient à cette fonction qui ajoute une box dans le panier et vérifie qu'il y a moins de 10 boxes dans ce dernier.

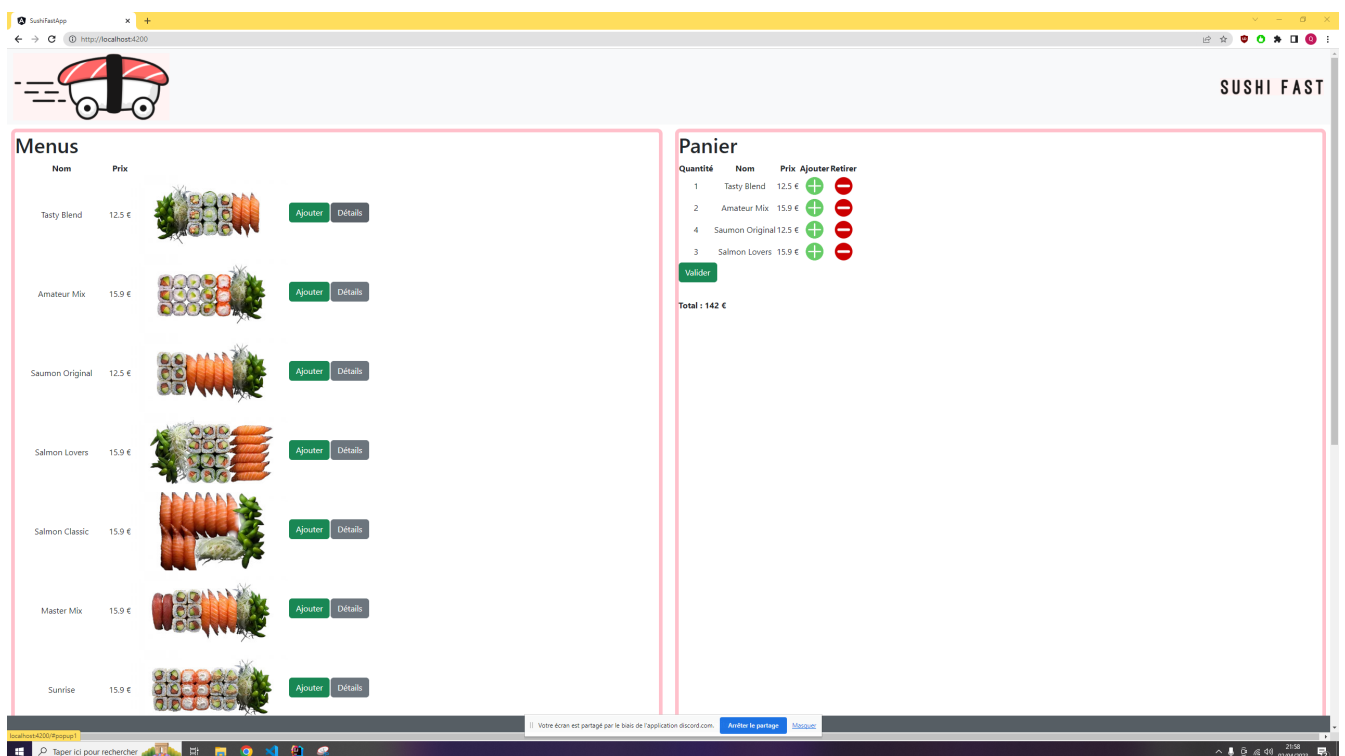
```

ajouterBoxAuPanier(box: Box){
  const index = this.panier.findIndex((element) => element.nom === box.nom);
  const nbBoxes = this.panier.reduce((total, panierBox) => total + panierBox.quantite, 0);

  if (nbBoxes >= 10){
    alert('Le nombre maximum de boîtes dans le panier est atteint !');
  }
  else if (index !== -1) {
    this.panier[index].quantite++;
  }
  else {
    const panierBox = new Panier(box.nom, 1, box.prix);
    this.panier.push(panierBox);
  }
}
}

```

Détails des boxes :



Pour visualiser les détails d'une box, nous avons ajouté pour chacune d'entre elles un lien donnant lieu à un pop-up indiquant composants de la box demandée :

Détails de la box : 3



Nom : Saumon Original

Pieces : 11

Prix : 12.5 €

Aliments :

- California Saumon Avocat (6)
- Sushi Saumon (5)
- Edamame/Salade de chou (1)

Saveurs :

- avocat
- saumon

Tout cela est géré dans le code `html` du menu, soit `menu.component.html` :

```
<div id="popup{{ box.id }}" class="overlay">
  <div class="popup">
    <h2>Détails de la box : {{ box.id }}</h2>
    <a class="close" href="#">&times;</a>
    <ul>Nom : {{ box.nom }} </ul>
    <ul>Pieces : {{ box.pieces }} </ul>
    <ul>Prix : {{ box.prix }} € </ul>
    <ul>Aliments :</ul>
    <ul>
      <li *ngFor="let aliment of box.aliments">{{ aliment.nom }} ({{ aliment.quantite }})</li>
    </ul>
    <ul>Saveurs : </ul>
    <ul>
      <li *ngFor="let saveur of box.saveurs">{{ saveur }}</li>
    </ul>
  </div>
</div>
```

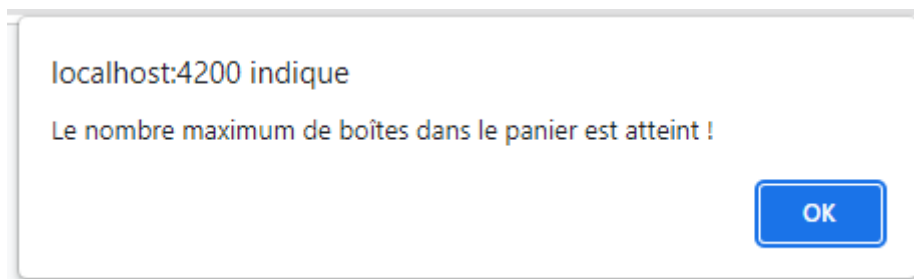

2.3 Cybersécurité :

1 - En tant que personne malveillante, je veux empêcher l'API de fonctionner en réalisant une attaque en réalisant un nombre important de requêtes à cette dernière.

En tant que développeur, nous définissons une limite d'ajout dans le panier afin d'empêcher un trop grand nombre de requêtes simultanées :

```
if (nbBoxes >= 10) {  
    alert('Vous avez atteint la limite de quantité pour le panier !');  
    return;  
}  
else{  
    box.quantite++;  
}
```

Voici le résultat obtenu en cas de tentative d'ajout au dessus de 10 articles :



2 - En tant que personne malveillante, je veux empêcher l'API de fonctionner en ouvrant des paniers sans en valider la commande.

En tant que développeur, nous définissons une limite de temps de dix minutes pour un panier qui se réinitialise lorsqu'un article est ajouté ou supprimé du panier.

Conclusion

3.1 Résumé de l'objectif :

Notre objectif était d'avoir une application fonctionnelle en lien avec une API, ce qui a été réalisé.

3.2 Résumé des résultats :

Nous sommes dans l'ensemble satisfait du résultat de notre application, le projet était compliqué quant à l'acquisition des données de l'API, cependant des solutions ont été trouvées.

3.3 Résumé de l'expérience :

Nous avons su gérer notre temps, le projet en lui-même était réalisable rapidement, cependant nous nous sommes heurtés à plusieurs freins qui nous prirent parfois plusieurs jours de développement et essais.

3.4 Perspectives d'avenir :

Nous pourrions implémenter un système de log et améliorer l'interface utilisateur, de façon à rendre cette dernière plus agréable.

Lien vers notre dépôt :

<https://github.com/MorgadoMathieu/SushiShopSQM/tree/master>