

Composants Interactifs

RESTo



Antoine BUQUET

-

Benoit GAUDET

-

Ayoub IMAMI

-

Mourad KARRAKCHOU

Sommaire

I - Contexte	2
II - Solutions	2
Présentation et justification de l'étape 1	2
Présentation et justifications de l'étape 2	6
III - Prise de recul	11
Étape 1	11
Étape 2	14
Évolution de besoins	14
Impacts	14
IV - Répartition des tâches	17
V - Conclusion	18

I - Contexte

Nous sommes une équipe de quatre développeurs qui ont, par le passé, travaillé sur la conception de l'interface des bornes du restaurant RESTo et dont la spécificité était de montrer aux clients les plats tendances du restaurant. Notre travail ayant été apprécié par ces derniers, nous avons été recontactés afin de travailler sur la conception de l'interface de leurs tables interactives.

Nous devions permettre aux clients de commander sur une table accueillant quatre personnes au maximum. Son écran est découpé en cinq parties, une partie d'écran pour chaque client et une partie commune au centre.

La spécificité de notre conception est que l'on offre la possibilité aux clients de re-commander pendant le temps d'attentes de préparation de leur première commande, mais également durant leur repas. Ainsi, ces derniers peuvent rapidement commander ce qu'ils ont pu oublier ou commander des boissons ou des desserts en fin de repas, conservant de ce fait leur température et appréciant chaque instant de leur repas.

Nous avons apporté une attention particulière sur la conformité de l'IHM aux exigences des clients. En effet, les interfaces de la borne et de cette table doivent être simples d'utilisation et intuitives.

Nous avons pu choisir les technologies sur lesquelles nous avons travaillé, à savoir Java Spring pour le Backend For Frontend - BFF et Angular pour le Frontend, car ce sont des technologies avec lesquelles nous sommes familiers. Mais nous avons aussi des contraintes. Les équipes Frontend et Backend sont souvent dissociés et il n'est pas forcément possible de modifier le Backend pour arriver à ses fins. Nous nous sommes donc imposés cette contrainte et n'avons pas apporté de changements au Backend, en plus du respect des deadlines.

Pour l'étape 2 et après étude de l'existant, nous avons noté plusieurs éléments que nous pouvions réutiliser comme les composants du Frontend affichant les menus ou le panier, ou encore notre BFF qui s'est avéré utile.

II - Solutions

Présentation et justification de l'étape 1

Lors de cette première étape, nous devions permettre aux clients de commander via une borne RESTo, de récupérer un identifiant de commande et d'être informé du moment de retrait de sa commande. Pour rappel, la spécificité de notre borne est le fait que nous montrons aux clients les plats tendances du moment, que propose le restaurant.



Nous avons décidé de mettre un header sur lequel l'utilisateur peut naviguer librement entre les Tendances, les Entrées, les Plats, les Desserts et les Boissons. En plus du titre, nous avons ajouté des icônes afin que l'utilisateur comprenne de quoi il s'agit d'un simple coup d'œil.

Sur l'onglet RESTo, tout le menu est présent, mais il est possible de trier par croissance ou décroissance de noms ou de prix, permettant ainsi au client de s'y retrouver. Le tri peut s'effectuer également sur les autres onglets.

L'utilisateur peut voir son panier, le prix de sa commande, l'annuler ou la confirmer, à tout moment grâce au “footer”. S'il l'annule, une confirmation apparaît pour éviter toute erreur de manipulation.

Figure 1.1 : Menu

Lorsque l'utilisateur navigue sur les onglets, il voit simplement les plats concernés par l'onglet. Ici, après avoir cliqué sur “Starters”, l'utilisateur ne voit que les entrées. Cela permet de voir plus facilement les plats proposés en fonction de leur catégorie.

Comme dit précédemment, nous notons que l'utilisateur peut toujours trier par noms ou par prix.

Il peut par ailleurs avoir accès à son panier, car il est important que cet accès soit assuré à tout moment de la commande.

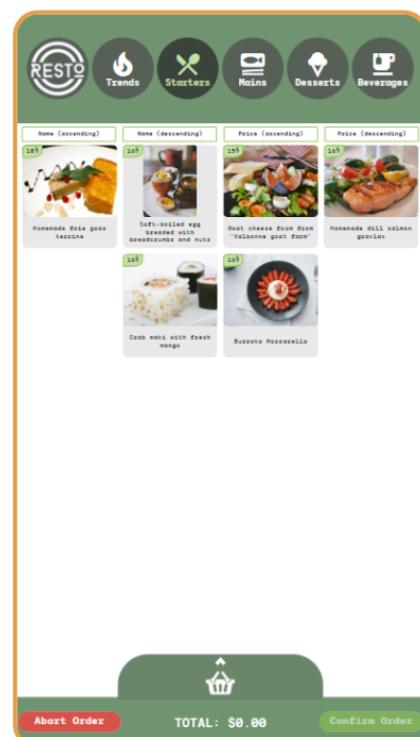


Figure 1.2 : Onglet “Starters”

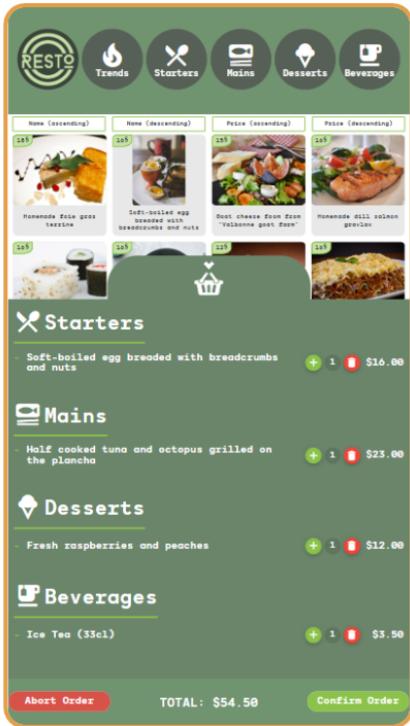


Figure 1.3 : Panier déroulant

En ce qui concerne les tendances du restaurant, nous affichons les trois plats, mais également le menu complet le plus populaire. L'utilisateur peut ajouter ces plats à son panier depuis cette page.

Les préférences des clients jouent un rôle crucial dans l'expérience utilisateur. Répondre à ces préférences satisfait le besoin fondamental des clients d'avoir une expérience personnalisée, de leur faire découvrir les plats les plus appréciés.

Cela permet aussi au restaurant de mieux comprendre les goûts de ses clients afin de toujours leur proposer des plats qui leur conviennent, ce qui peut augmenter leur satisfaction et leur fidélité envers le restaurant.



Figure 1.4 : Tendances

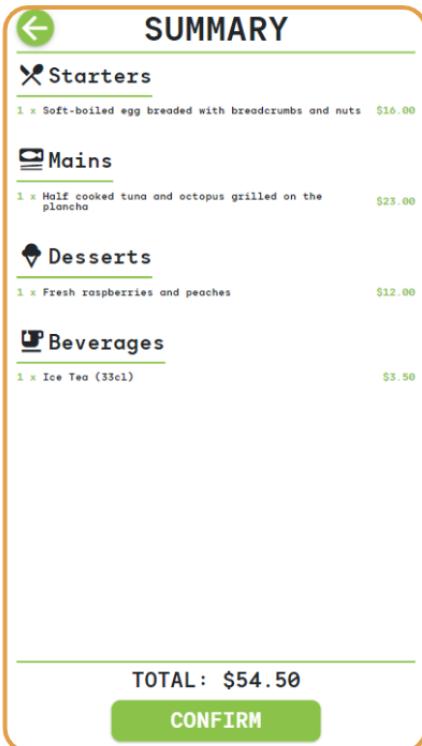


Figure 1.5 : Résumé de commande

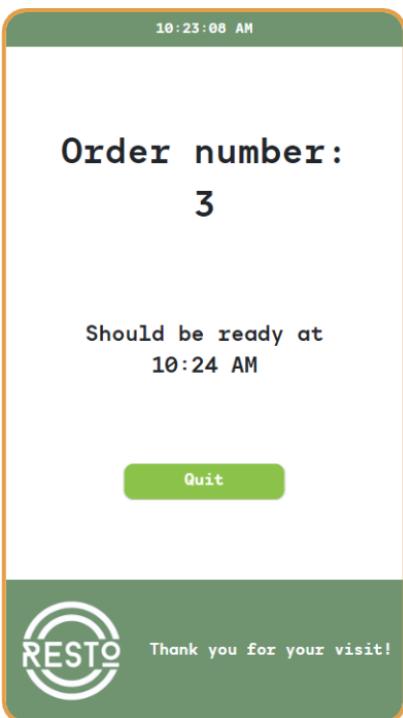


Figure 1.6 : Fin de commande

Lorsque l'utilisateur confirme sa commande, un récapitulatif lui est montré afin qu'il s'assure qu'elle est correcte.

Il a la possibilité de revenir en arrière pour continuer sa commande ou l'annuler, car nous voulons laisser la liberté à l'utilisateur de modifier sa commande tant qu'il n'a pas payé.

Enfin, il peut effectuer le paiement en cliquant sur confirmer.



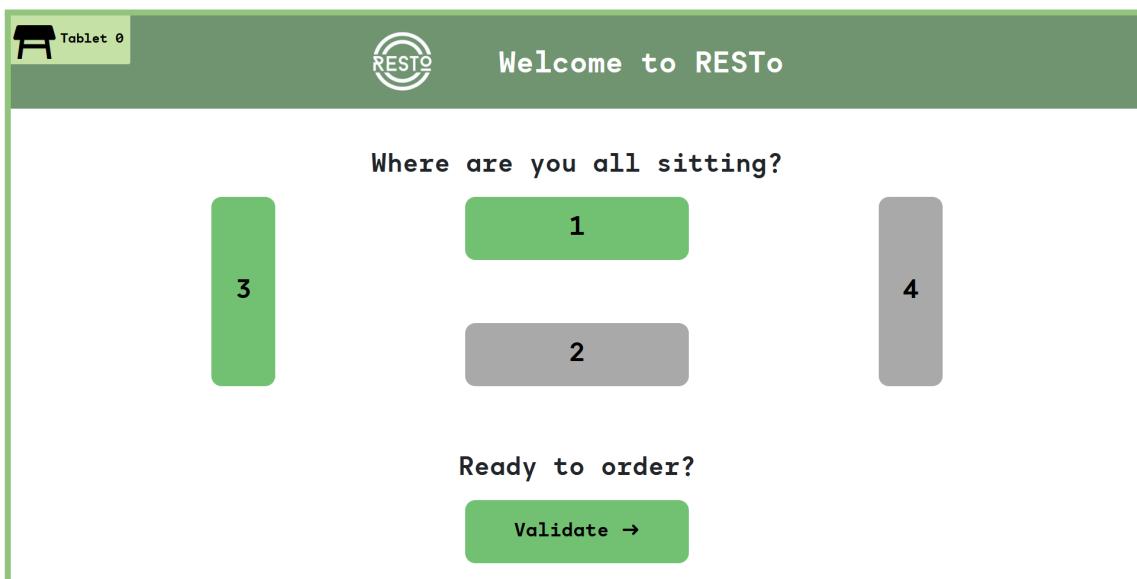
Figure 1.7 : Page d'accueil

Voici ce qu'affiche la version **avec BFF**, une comparaison est faite avec la version **sans BFF**, dans la partie **Prise de recul**.

Présentation et justifications de l'étape 2

Exactement comme dans un projet réel, les besoins de notre application ont évolués. Le sujet a désormais pour objectif de porter notre solution sur une table interactive comportant 5 tablettes, une principale au centre et une sur chaque côté. Pour rappel, il faut que les tables de quatre puissent s'adapter aux plus petits groupes. Dans le cas d'un groupe de plus de 4 personnes, ils seront répartis sur plusieurs tables et considéré comme des groupes indépendants. Chaque convive prend sa commande et cela constitue maintenant une commande groupée. Le groupe doit avoir la possibilité de choisir à la fin, s'ils souhaitent payer individuellement leur part ou s'il s'agit de la responsabilité d'une seule personne, dans le cadre d'un repas d'entreprise par exemple. Enfin, un jeu doit leur être proposé en attendant que leurs plats soient servis, tout en ayant à disposition le suivi de la commande.

Au début, toutes les tablettes affichent une page d'attente. Elles sont globalement similaires, la seule différence résidant dans le texte et l'impossibilité de cliquer sur la page des tablettes clients. Cela permet de forcer le client à d'abord cliquer sur la tablette centrale pour y faire apparaître la page de configuration. Cette page permet aux clients de choisir leur place autour de la table et donc d'indiquer leurs nombres indirectement. Il faut évidemment qu'au moins une tablette soit sélectionnée pour pouvoir valider et avancer.



Une fois validé, les tablettes non choisies s'éteignent tandis que celles sélectionnées affichent désormais la page de catalogue. Il était inutile de garder l'écran d'attente durant tout le repas sur les tablettes inutilisée, cela pourrait déranger les clients.

Tablet 1

RESTO

Tzends Starters Mains Desserts Beverages

Name (ascending)		Name (descending)		Price (ascending)		Price (descending)	
18\$	16\$	15\$	16\$	16\$	16\$	12\$	16\$
Homemade foie gras terrine	Soft-boiled egg breaded with breadcrumbs and nuts	Goat cheese foam from 'Valbonne goat farm'	Homemade dill salmon gravlax	Crab maki with fresh mango	Burrata Mozzarella	Delicious Pizza Regina	Lasagna al forno
19\$	24\$	23\$	6.5\$	12\$	12\$	12\$	12\$
Homemade beef burger	Beef chuck cooked 48 hours at low temperature	Half-cooked tuna and octopus grilled on the plancha	Brownie (home made)	Valzhana chocolate declination with salted chocolate ice cream	Marmalade of Menton's lemon - Lemon cream - Limoncello jelly...	Fresh raspberries and peaches	Dessert of fresh strawberries and vanilla mascarpone mousse
12\$	18\$	3.5\$	3.5\$	1\$	1.5\$	5\$	6.5\$
Fresh seasonal fruit	Speculoos tiramisu	Coca-Cola	Iced coffee	Water	Glass of water	Spritz	Margarita

▲ ▾ ⚡

Abort Order **Confirm Order**

TOTAL: \$0.00

La page de catalogue fonctionne de la même manière que la borne, les clients peuvent ajouter, consulter et supprimer des articles dans leur panier. Ils peuvent aussi trier les articles par noms ou par prix. Nous avons également gardé notre système de préférence qui était notre caractéristique spéciale lors de l'étape 1. Le fait d'avoir gardé le même design, mais de l'avoir simplement adapter au nouveau format, permet aux clients habitués du restaurant de ne pas être perdu avec ce nouveau support.

Tablet 1

RESTO

Tzends Starters Mains Desserts Beverages

Most popular items

Add to basket (12\$) **Add to basket (12\$)** **Add to basket (1.8\$)**

Delicious Pizza Regina Fresh seasonal fruit Café

Most popular menu

Best starter	Best main	Best dessert	Best beverage
18\$	12\$	12\$	1.8\$
Homemade foie gras terrine	Delicious Pizza Regina	Fresh seasonal fruit	Café
Best starter	Best main	Best dessert	Best beverage

Add the full menu to the basket (43.8\$)

Abort Order **Confirm Order**

TOTAL: \$0.00

La page récapitulative du panier, apparaissant après avoir fait ses choix, est, elle aussi, toujours présente. Une fois que le client valide son panier, celui-ci est en réalité pré-validé. C'est-à-dire que le client est prêt pour passer la vraie commande de groupe. Il peut néanmoins revenir au catalogue et modifier son panier s'il s'avère qu'il a omis quelque chose.

SUMMARY

Starters
1 x Homemade foie gras terrine \$18.00

Mains
1 x Delicious Pizza Regina \$12.00

Desserts
1 x Fresh seasonal fruit \$12.00

Beverages

TOTAL: \$43.80

CONFIRM

Tablet 1 10:38:57 AM

Order pre-validated!

Modify order

RESTO

Thank you!

Pendant que les clients composent leur panier, le récapitulatif de tous les paniers différents est affiché sur la tablette centrale, ainsi que le prix total et l'état de chaque panier, prêt ou pas prêt à commander. Il est aussi possible de visualiser la commande de manière générale comme s'il s'agissait d'un seul panier.

Tablet 0

SUMMARY

Individual recap **Global recap**

● : Ready to order ● : Ordering

Customer 1 ●

Mains
1 x Delicious Pizza Regina \$12.00

Customer 3 ●

Starters
1 x Homemade foie gras terrine \$18.00

TOTAL: \$30.00

CONFIRM

Une fois que tous les clients ont pré-validé leur panier, il devient possible de confirmer la commande avec le bouton "Confirm" ci-dessus sur la tablette centrale. Cela aura pour effet d'envoyer la commande à la cuisine ainsi que de rediriger toutes les tablettes clients vers la page du jeu d'attente : un Cookie Clicker, et la tablette centrale vers la page d'attente des préparations où l'avancement des

préparations est indiqué pour chaque catégorie d'article tel que plat chaud, plat froid ou boisson, mais aussi le classement du jeu avec les nombres de cliques de chaque client. Grâce à notre système de panier prêt/non prêt et la possibilité de commander seulement tous les paniers prêts, nous pouvons créer qu'une seule commande groupée à envoyer à la cuisine et cela permet de servir tous les clients en même temps.

ORDER PREPARATION ADVANCEMENT

(10:55:53 AM)

Category	Status	Ready At	Items
Drinks	Not started	Should be ready at 10:53:40 AM	x1 Café
Cold dishes	Not started	Should be ready at 10:53:57 AM	x1 Fresh seasonal fruit
Hot dishes	Not started	Should be ready at 10:53:57 AM	x1 Homemade beef burger x1 Homemade foie gras terrine

Leaderboard

Tablet	Score
Tablet 1	13
Tablet 3	6
Tablet 2	0
Tablet 4	0

Start the meal

Sur la tablette client, en plus du jeu où le but est de cliquer le plus possible sur le cookie, le client peut choisir de re-commander s'il le souhaite. Cependant, à la fin de sa nouvelle commande, celle-ci sera considérée comme à part de la première et donc un nouveau temps estimé de préparation lui est donné. Cela a pour but de pouvoir individualiser la partie dédiée à la commande ultérieure et ainsi donner la liberté à chacun des clients de pouvoir ajouter des plats selon ses envies une fois que la commande groupée est terminée. Il sera également possible de re-commander plus tard pendant le repas et le fonctionnement sera similaire.

Tablet 1

RESTO

Welcome to RESTO

You have eaten 13 cookies!

← Order again

10:58:31 AM

Should be ready at:
11:58 AM

RESTO

Thank you!

Une fois que le repas est servi, c'est aux clients de choisir quand commencer le repas, en fonction de s'ils souhaitent continuer à jouer au jeu ou non. Une fois le repas commencé, toutes les tablettes clients s'éteignent et la tablette centrale devient noire, afin de ne pas déranger visuellement les clients pendant leur repas. Seuls quelques boutons sont affichés sur la tablette centrale, mais on peut choisir de la passer en mode veille pour avoir un écran totalement vide si besoin. Les boutons présents sont ceux permettant de rallumer une tablette précise si l'un des clients à l'envie de re-commander quelque chose et enfin ceux permettant de choisir la méthode de paiement, séparément ou en groupe, pour procéder au paiement.



Pour terminer, lors du paiement, indépendamment de la façon choisie, le reçu et la note globale est indiqué sur la tablette centrale et le reçu et la note individuelle est indiqué sur chacune des tablettes clients. Si le groupe a choisi de faire un unique paiement, alors le bouton "Pay" se trouvera sur la tablette centrale. Dans le cas inverse, il y aura un bouton "Pay" sur chacune des tablettes clients et chacun devra procéder au paiement.

GLOBAL RECEIPT:

- Customer 1:**
 - Starters:** 1 x Goat cheese foom from "Valbonne goat farm"
 - Mains:** 1 x Half cooked tuna and octopus grilled on the plancha
- Customer 3:**
 - Desserts:**

TOTAL: \$56.00

PAY

RECEIPT:

- Customer 1:**
 - Starters:** 1 x Goat cheese foom from "Valbonne goat farm"
 - Mains:** 1 x Half cooked tuna and octopus grilled on the plancha
- Customer 3:**
 - Desserts:**

TOTAL: \$38.00

III - Prise de recul

Étape 1

Du point de vue de l'interface utilisateur, l'objectif était de personnaliser et d'améliorer au maximum l'expérience utilisateur. Cela a été réalisé notamment grâce à plusieurs fonctionnalités permettant une navigation simplifiée au sein de l'application, telles qu'un système de tri par prix ou par ordre alphabétique. Parallèlement, nous avons cherché à créer une expérience unique avec la fonctionnalité des plats populaires, contribuant ainsi à offrir une expérience personnalisée.

Pour la première étape, nous avons choisi de nous confronter à la contrainte de l'indisponibilité de l'équipe Backend, rendant impératif que toutes nos fonctionnalités soient implémentables malgré cette restriction. Cette décision a entraîné des défis, surtout pour simplifier l'identifiant de la commande pour le client, avec la mise en place du numéro de commande (*Figure 3.1*). Dans la solution avec BFF, nous avons été contraints de stocker l'information dans le BFF, le rendant ainsi stateful, ce qui peut poser des problèmes en cas d'une panne de celui-ci, entraînant la perte de l'ensemble des numéros de commande. Dans la solution sans BFF, le défi résidait dans l'impossibilité de simplifier l'identifiant de la commande et de le traduire en numéro de commande sans modifier le Backend, nous contraignant ainsi à limiter l'expérience utilisateur, une approche peu souhaitable (*Figure 3.2*). Cela met en lumière un avantage du BFF mais aussi la limite de ne pas pouvoir modifier directement le Backend. En effet, cette fonctionnalité était vraiment le point bloquant où nous aurions voulu pouvoir modifier celui-ci, car même la solution avec le BFF nécessite de le rendre stateful, et comme mentionné précédemment, cela comporte aussi ses propres défauts.

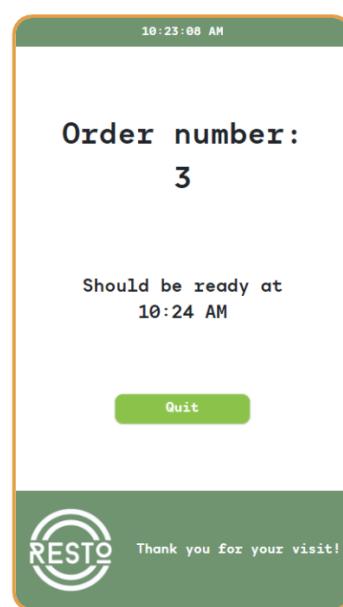
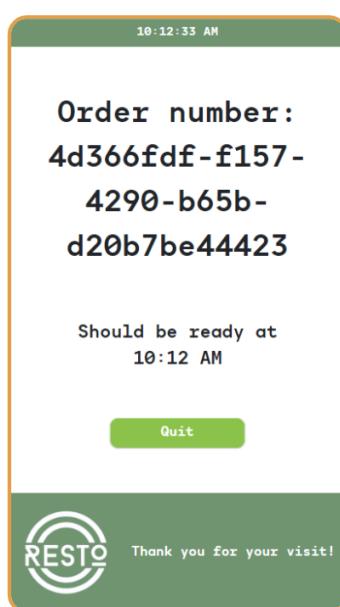


Figure 3.1 : Page de confirmation version sans BFF

Figure 3.2 : Page de confirmation version avec BFF

La mise en œuvre de la fonctionnalité des plats populaires a nécessité deux approches différentes. Dans la version avec BFF, le calcul des plats populaires s'effectue et est ensuite stocké dans un cache au sein de celui-ci, éliminant ainsi la nécessité de refaire l'appel et le calcul à plusieurs reprises entre le BFF et le Backend. Ainsi, lorsque différente borne feront appelle au BFF, la réponse n'aura pas besoin d'être recalculée. De plus, étant donné que le BFF est déployé au même endroit que les services, la communication et le réseau sont rapides et économiques. Il est alors nécessaire de réaliser uniquement deux appels, comme illustré dans la Figure 3.3 : le premier pour obtenir la liste des plats les plus vendus et le deuxième pour identifier le plat le plus vendu dans chaque catégorie. Ce sont des appels légers qui contiennent uniquement les noms des plats concernés. En revanche, dans la version sans BFF, l'obligation d'envoyer l'ensemble de l'historique du magasin vers la borne constitue une opération potentiellement coûteuse, notamment en termes de temps. Dans la Figure 3.4, on peut observer le premier appel qui récupère l'ensemble des tableOrders, constituant l'historique complet de toutes les commandes. De plus, le calcul doit être effectué directement sur la borne, qui peut ne pas disposer d'une puissance de calcul élevée, constituant ainsi un obstacle. Du point de vue de la sécurité, il n'est pas souhaitable d'envoyer toutes ces informations vers le Frontend, car la borne est plus facilement accessible et potentiellement vulnérable aux attaques, permettant à un individu malveillant de récupérer l'historique complet des commandes du magasin en piratant la borne.

État	Méthode	Domaine	Fichier
200	GET	localhost:3001	tableOrders
200	GET	localhost:3000	1f68e2c0-0cf2-436e-917a-128de61f20af
200	GET	localhost:3000	27ce73d-a2cc-4391-8c1f-761d422fefc6
200	GET	localhost:3000	f56bd8c5-a7c5-4b1d-b03d-a1a87443b241

Figure 3.3 : Appelle vers le Backend en utilisant la version sans BFF

État	Méthode	Domaine	Fichier
200	GET	localhost:8080	preference
200	GET	localhost:8080	by-category

Figure 3.4 : Appelle vers le BFF

Pour effectuer une commande, l'utilisation du BFF permet de minimiser le nombre d'appels nécessaires au Backend. Comme illustré dans la Figure 3.5, dans la version sans BFF, il est requis de réaliser neuf requêtes au Backend pour passer une commande comportant quatre articles. En revanche, la Figure 3.6 montre que dans la version avec BFF, seulement deux requêtes sont nécessaires. Cette différence peut avoir un impact significatif sur les performances de l'application. En effet, multiplier les appels vers un serveur distant peut entraîner des latences. En confiant cette responsabilité au BFF, situé à proximité des autres serveurs, nous sommes en mesure de réduire la latence.

État	Méthode	Domaine	Fichier
200	GET	localhost:3001	tables
201	POST	localhost:3001	tableOrders
200	OPTIONS	localhost:3001	9d7048a2-239b-418b-a5f5-444ace62f349
201	POST	localhost:3001	9d7048a2-239b-418b-a5f5-444ace62f349
201	POST	localhost:3001	9d7048a2-239b-418b-a5f5-444ace62f349
201	POST	localhost:3001	9d7048a2-239b-418b-a5f5-444ace62f349
201	POST	localhost:3001	prepare
200	POST	localhost:3001	bill

Figure 3.5 : Appelle vers le Backend en utilisant la version sans BFF

État	Méthode	Domaine	Fichier
200	OPTIONS	localhost:8080	order
200	POST	localhost:8080	order

Figure 3.6 : Appelle vers le BFF

Les requêtes ne nécessitaient pas toutes de transiter par le BFF. En effet, celui-ci est mis en place pour ajouter une surcouche de logique au Backend, mais lorsqu'elle n'est pas nécessaire, il est préférable d'appeler directement le Backend. Un exemple concret est celui des menus, où il est possible d'accéder directement à l'ensemble des menus dans le Backend. Cela évite d'ajouter un appel supplémentaire, réduisant ainsi la latence. Un seul appel nous donne alors l'ensemble du menu (voir Figure 3.7).

État	Méthode	Domaine	Fichier
200	GET	localhost:3000	menus

Figure 3.7 : Requête pour avoir accès au menu

En conclusion, l'adoption d'un BFF dans le développement présente des avantages significatifs par rapport à son absence dans l'architecture. Sans le BFF, le code du Frontend peut rapidement devenir complexe et peu sécurisé en raison de la présence excessive de logique. Les changements nécessitent également un redéploiement sur l'ensemble des bornes, ce qui peut être inefficace. En revanche, l'introduction d'un BFF permet une centralisation judicieuse de la logique métier, facilitant la gestion des requêtes et évitant un couplage excessif grâce à l'ajout d'un micro-service. De plus, les changements ne nécessitent qu'une modification unique, simplifiant ainsi la maintenance et améliorant la flexibilité du système. Ainsi, l'adoption du BFF émerge comme une approche plus efficace et sécurisée pour la gestion du code frontal.

Étape 2

Évolution de besoins

La réalisation de la borne nécessitait que chaque commande soit indépendante et que les utilisateurs puissent commander même s'il n'y avait pas de tables disponibles dans le restaurant. Cela incluait aussi qu'une fois la commande validée et payée, elle ne pouvait pas être modifiée. Cependant, avec le passage à la table interactive de l'étape 2, les utilisateurs se doivent d'être associés à une table existante du restaurant. Contrairement à la borne, la commande peut évoluer au fur et à mesure du repas, en fonction des envies et des oubliés des utilisateurs.

Avec la borne, les utilisateurs recevaient un numéro de commande, ainsi que l'heure à laquelle leur commande devrait être préparée. Avec la table interactive, les utilisateurs n'ont plus de numéro de commande, car ils sont associés à une table du restaurant et ont un suivi de la commande en temps réel durant lequel ils peuvent aussi se divertir en jouant à un jeu.

La borne permettait seulement à un utilisateur de faire la commande, donc à un seul utilisateur de payer l'entièreté de la commande. Cependant, l'utilisation d'une table interactive implique aussi par exemple qu'un groupe d'amis ou de collègues viennent manger ensemble. Ces types d'utilisateurs peuvent vouloir payer indépendamment ce qu'ils ont commandé. La table doit en conséquence permettre aux utilisateurs de payer individuellement leur propre commande et non pas seulement l'option de payer pour toute la table.

Impacts

Pour le développement de la nouvelle étape, donc de la table interactive, un certain nombre de besoins sont apparus ou ont évolués par rapport à la borne. Toutefois, le code que nous avions développé pour l'étape 1 était majoritairement réutilisable. En effet, de nombreuses pages et composants ont simplement pu être adaptées au nouveau format, mais conservait leurs logiques de fonctionnement originel, comme la page de passage de la commande individuelle ou encore les composants de récapitulatif de la commande. Tous les composants graphiques utilisés lors de la borne ont pu être réutilisés pour la réalisation de la table interactive, bien que certains aient été adaptés ou aient été modifiés. Voici quelques exemples :

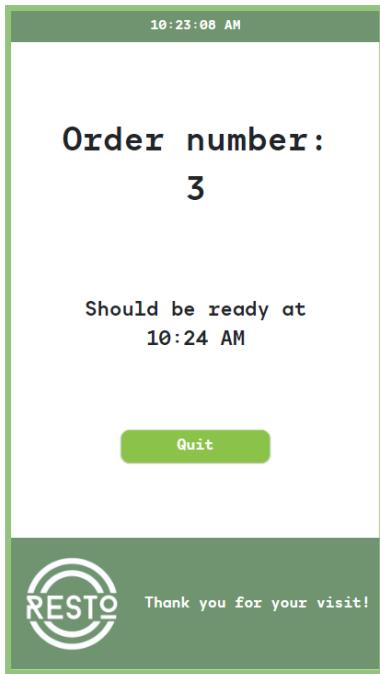


Figure 4.1 : Composant d'affichage du numéro de commande

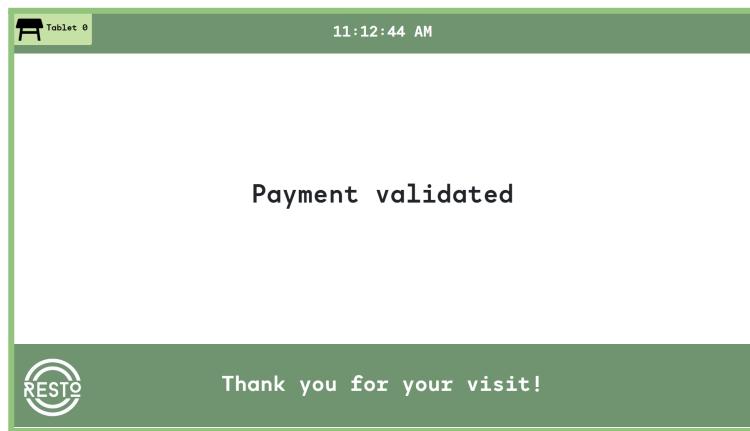


Figure 4.2 : Composant de confirmation de paiement modifié depuis le composant de gauche

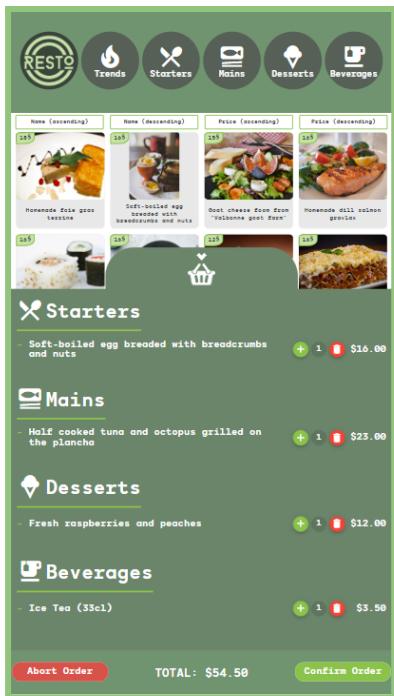


Figure 4.3 : Composant de sélection de la commande



Figure 4.4 : Composant de sélection de la commande adapté depuis le composant de gauche

Dans l'étape 2, nous avons gardé l'utilisation du BFF. En effet, comme dit précédemment, le BFF était préférable pour des raisons de sécurité, de latence et pour l'implémentation de fonctionnalité. La borne et les tables interactives peuvent utiliser le BFF simultanément puisqu'il a été adapté afin de pouvoir gérer les deux types d'applications avec leurs spécificités, en ajoutant des routes et de la logique quand cela était nécessaire. Bien que les deux étapes utilisent un BFF, la table interactive fait plus de requêtes que la borne pour passer une commande, car cette dernière possède de la logique en plus comme la capacité de pouvoir re-commander ou encore avoir le suivi des commandes.

État	Méthode	Domaine	Fichier
200	OPTIONS	localhost:8080	order
200	POST	localhost:8080	order

Figure 4.5 : Passer une commande sur la borne avec BFF

⌚ order	200
⌚ 1	200
⌚ 1	200
📄 bill	200

Figure 4.6 : Passer une commande sur la table interactive avec BFF

En conclusion, même s'il y a eu une évolution des besoins entre la borne de commande et la table interactive, les composants graphiques existants avec la borne ont pu être adaptés ou modifiés dans le Frontend de la table interactive. D'autres composants ont bien sûr dû être créés, mais principalement dans le cas où la fonctionnalité était nouvelle comme pour le suivi des commandes ou encore pour l'implémentation d'un jeu d'attente. Les composants existants étaient suffisamment bien pensés pour être réutilisés pour une utilisation proche.

L'utilisation d'un BFF est réellement positif dans notre application, que ce soit pour la borne, comme pour la table interactive. Il permet de grandement réduire le nombre de requêtes nécessaire pour passer une commande entre le Frontend et le Backend. Il permet aussi d'alléger le Frontend en prenant la responsabilité d'une partie de la logique. Le BFF existant a été légèrement modifié afin de pouvoir fournir à la table les nouvelles fonctionnalités demandées, tout en pouvant continuer de fonctionner pour l'utilisation de la borne et de la table interactive.

IV - Répartition des tâches

Afin de travailler efficacement, nous avons utilisé des issues que nous avons inclus dans 3 Milestones différentes : User Interface, Backend For Frontend et Next Step: Interactive Table.

La première milestone concerne les besoins utilisateurs, le design et l'adéquation de l'IHM aux besoins des utilisateurs. La deuxième porte sur l'intégration du Backend For Frontend et de son intégration entre le Frontend et le Backend. Et la dernière traite l'évolution des besoins avec la table interactive, en conservant l'utilisation du BFF.

Lors de la première étape, Antoine a mis en place le catalogue avec les différentes pages, dont la page de tendance, et les fonctions de tri. Il a également travaillé sur le composant très utilisé qui résume le contenu du panier et le service de panier en lui-même. Enfin, il s'est occupé de passer une commande sans utiliser le BFF. Benoit a principalement travaillé sur la partie passage de commande dans le BFF, sur la création de page d'attente et sur la création de numéros de commande simplifié. Ayoub a travaillé sur le panier, son contenu et son animation afin de l'afficher ou de le cacher. Il a également travaillé sur l'intégration du BFF et sa liaison entre le Frontend et le Backend. Mourad a développé la logique des préférences dans la partie avec et sans BFF, ainsi que plusieurs composants dans le Frontend, tels que le header ou l'idle.

En ce qui concerne la deuxième étape, Antoine s'est chargé d'adapter le service panier pour gérer plusieurs paniers simultanément, créer la page de paiement des tablettes clients et enfin mettre en place le système d'état des tablettes, indispensable pour la démonstration. Benoit a travaillé sur l'adaptation du passage de commande du BFF de la borne de façon à ce qu'il fonctionne sur la table interactive. Il a aussi réalisé le suivi des commandes en cuisine sur la partie BFF, ainsi que les composants Frontend pour son affichage, et celui du classement du jeu. Ayoub s'est occupé de l'écran central pour permettre aux clients de choisir leur place et donc indirectement leur nombre. Il a ajouté le mode veille qui permet entre autres, pendant le repas, de ne pas déranger les clients, mais aussi de re-commander et de payer ensemble ou séparément. Il a de plus travaillé sur le jeu de l'écran client tout en leur donnant la possibilité de re-commander. Mourad s'est concentré sur la liaison du paiement avec le BFF, la page de paiement en gérant les différents types de paiement, à savoir groupé ou individuel, les récapitulatifs des commandes groupées sur l'écran central, et la logique permettant de re-commander.

Notez toutefois que nous avons tous travaillé sur des périmètres bien plus grands, car nous avons chacun été amenés à travailler sur d'autres parties, que ce soit pour résoudre des problèmes, améliorer des designs, ou faire évoluer des fonctionnalités. Cela permettait notamment d'avoir une cohérence entre les différentes parties, les différentes pages, et une communication constante sur la vision globale de l'application et du flux que notre application devait suivre.

Antoine BUQUET	100
Benoit GAUDET	100
Ayoub IMAMI	100
Mourad KARRAKCHOU	100

Tableau illustrant l'implication de chacun des membres de l'équipe

V - Conclusion

L'un des objectifs de ce cours était de se confronter à une problématique en entreprise sur la relation entre les équipes Frontend et Backend. En effet, en tant que développeur Frontend, nous ne sommes pas toujours en mesure de modifier le Backend pour arriver à nos fins, il faut donc trouver des alternatives et faire des choix, tout en offrant le meilleur service possible aux utilisateurs en mettant l'accent sur l'adéquation de l'IHM à leurs besoins.

Nous avons également étudié les différentes solutions telles que l'utilisation d'un Backend For Frontend et du Frontend directement afin de communiquer avec une architecture en micro services pour le Backend.

Enfin, notre prise de recul nous a permis de comprendre les problèmes liés à la conception et l'intérêt des différentes solutions. Dans le cadre de l'architecture logicielle, il convient de concevoir le système en prenant en compte les problématiques que peuvent rencontrer l'équipe Frontend.