



# Rendu Final

---

## Projet Informatique Individuel

Joanne Gangapal

Groupe de TD 3

5 Avril 2024



# Sommaire

<b>Sommaire</b>	<b>2</b>
<b>Contextualisation</b>	<b>3</b>
<b>Choix techniques et base de données</b>	<b>4</b>
Partie Front-end	4
Partie Back-end	5
Partie Versionnage	7
<b>Gestion de projet</b>	<b>8</b>
<b>Critique, prise de recul et bilan du projet</b>	<b>10</b>
Exigences Fonctionnelles	11
Exigences Optionnelles	12
<b>Annexe</b>	<b>12</b>

# Contextualisation

Si vous êtes un habitué de Bordeaux, vous connaissez sûrement le restaurant L'Entrecôte. Ce dernier est un restaurant célèbre à Bordeaux pour sa file d'attente interminable, que ce soit à midi ou le soir. En passant à de nombreuses reprises devant, nous nous sommes posé plusieurs questions comme :

- "Pourquoi y a-t-il autant de monde faisant la queue devant ce restaurant précisément alors qu'il y a le quatrième mur juste à côté ?"
- "Comment les serveurs s'organisent-ils à l'intérieur ?"
- "Est-ce que les employés ont une charge mentale supplémentaire de travail sachant qu'il y a autant de monde à l'extérieur ?"
- "Est-ce la nourriture qui attire les personnes ou est-ce l'organisation du service mal effectuée qui engendre une file d'attente aussi longue ?"
- 

De ces nombreuses questions est né une idée. Une application de gestion de restaurant. Cette dernière aiderait les restaurateurs à visualiser le niveau d'occupation de leurs restaurants mais permettrait également d'offrir une expérience de restauration plus agréable . Qu'il s'agisse du personnel, de la logistique mais également des services. Ainsi est né notre volonté de créer FluidService l'ami des restaurateurs mais également de leurs personnels.



Image 1 : Logo de l'application

# Choix techniques et base de données

## Partie Front-end

Afin de renseigner les restaurateurs et de soulager la charge mentale de leurs équipes, nous avons fait le choix de créer une application dans un premier temps en React Native. Cependant après une étude approfondie du marché et au vu de de la numérisation des bons en restauration grandissante nous avons opté pour plus de souplesse.

C'est avec ces nouveaux éléments en tête que nous nous sommes orientés vers le langage React JS. Cette approche nous offre une expérience fluide et plus accessible pour le personnel , tout en permettant aux restaurateurs de gérer efficacement les opérations. Nous souhaitons mettre en place une interface fluide, adaptée et réactive. Nous espérons contribuer ainsi à une expérience gastronomique plus efficace et satisfaisante pour tous.

L'étude de marché nous a montré également que des entreprises telles que [zenchef](#) , [Combo](#), [The fork](#) ou encore [Lightspeed](#) sont déjà bien ancrées dans les secteurs de la logistique ou des ressources humaines. De ce fait certains objectifs ont été revu afin de mieux nous concentrer sur des éléments peu voir pas encore élaborés par ces entreprises. Ainsi nous avons décidé de nous concentrer sur nos nouveaux objectifs tel que le service en salle et son organisation ainsi que sur les réservations. En effet, si des clients souhaitent se restaurer sans avoir réservé, le réceptionniste pourra en un coup d'œil savoir l'état de la salle et donc répondre à la demande beaucoup plus aisément.

De plus, notre projet se voulant à la base accessible aux restaurateurs, aux personnels mais également aux consommateurs nous nous sommes restreint aux besoins des runners (employé qui fait le lien entre la cuisine, le bar et les tables sans pour autant être assigné à une seule et même table), réceptionnistes et serveurs.

Ainsi pour la partie front-end nous avons utilisé Html, Css, Bootstrap, Javascript et ReactJS.



Image 2 : Technologies utilisées pour le front-end

## Partie Back-end

Concernant la partie back-end nous avons fait le choix d'utiliser phpMyAdmin à l'aide de Uwamp avec un serveur MySQL. Le lien entre le front-end et le back-end s'est effectué à l'aide de reactJS.

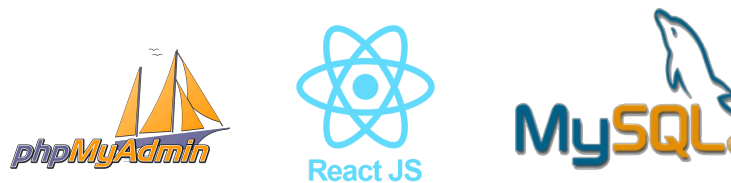


Image 3 : Technologies utilisées pour le back-end

L'image suivante est notre schéma relationnel de données pour ce projet. Les connexions, les inscriptions et l'obtention des données se font via un dossier extérieur (Server) à la vu client (FluidService) pour faire la distinction entre le côté serveur et le côté client.

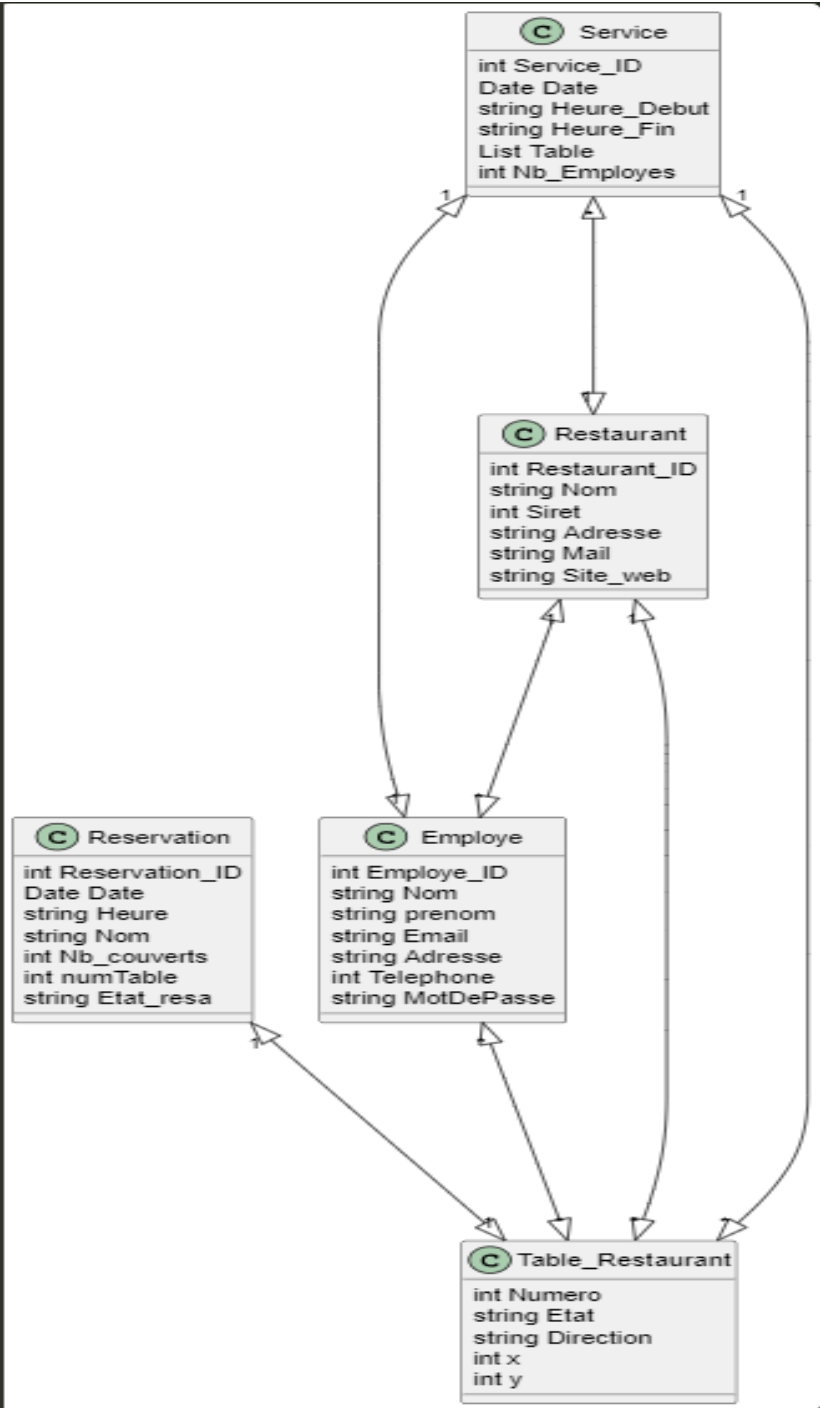
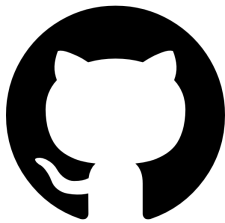


Image 4 : Diagramme UML

## Partie Versionnage



Pour le versionnage nous avons fait le choix d'utiliser [Github](https://github.com).

Image 5 : Technologies utilisées pour le versionnage

Projet Individuelle informatique 2024 Prérequis :

- Télécharger le projet
- Importer la base de donnée dans phpMyAdmin via uwamp ou xampp
- Télécharger NodeJS ( version 20.xx.X)
- aller dans le dossier téléchargé
- dans un terminal à cette endroit effectué les commandes suivantes :
  - `cd fluidservice`
  - `npm install`
  - `npm install react-scripts`
  - `npm install --save office-map --force`
  - `npm start`
- dans un autre terminal en parallele :
  - `cd server`
  - `npm install -g nodemon`
  - si les scripts ne sont pas autorisé sur l'ordinateur :
    - ouvrir une fenêtre powershell
    - `Set-ExecutionPolicy`
    - `O`
  - `nodemon index.js`

Sur ce dernier le projet peut être télécharger et lancer à l'aide d'un fichier readme faisant office de tutoriel d'installation et de lancement du projet.

Image 6 : Tutoriel pour lancer l'application

# Gestion de projet

Le début du projet a été ralenti par une période de maladie assez lourde ainsi qu'un retard dans la bonne direction à prendre pour ce projet. En effet, le monde de la restauration étant très vaste, le nombre de fonctionnalités que nous voulions implémenter au départ était beaucoup trop nombreuses pour la période de temps alloué à ce projet.

Ainsi dans un premier temps nous avons revu à plusieurs reprises le cahier des charges, afin d'être certain de nous éparpiller le moins possible et d'avoir un unique public visé (les restaurateurs).

Par la suite, nous avons élagué et recentré nos fonctionnalités sur notre public ciblé afin d'au mieux répondre aux besoins cités dans la contextualisation. C'est -à -dire: " Aider les restaurateurs à visualiser le niveau d'occupation de leurs restaurants et également alléger la charge mentale des employés."

A la fin nous avons mis en place un planning prévisionnel :

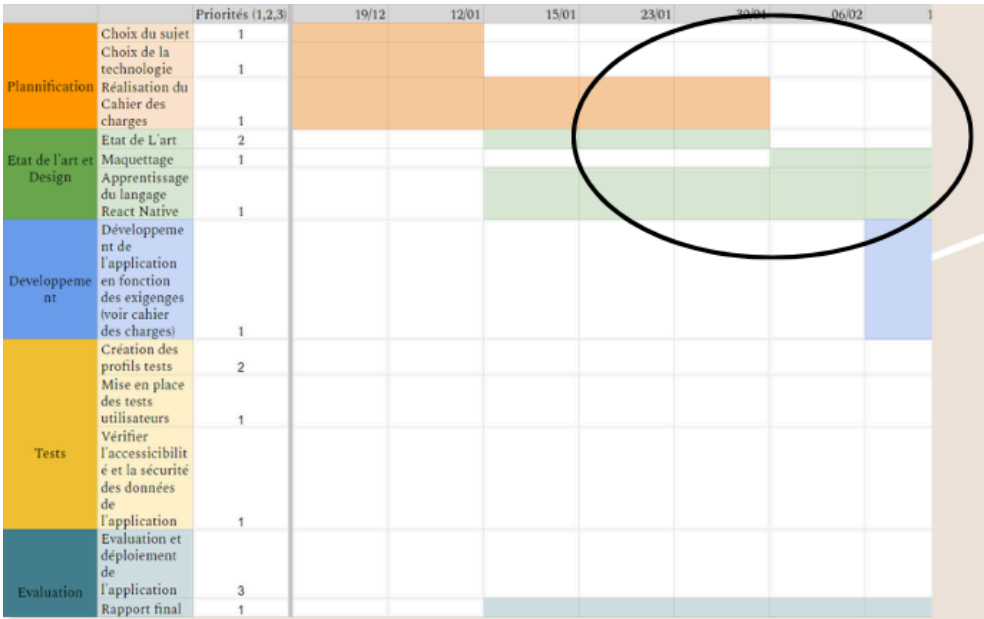
		Priorités (1,2,3)	19/12	12/01	15/01	23/01	30/01	06/02	13/02
Plannification	Choix du sujet	1							
	Choix de la technologie	1							
	Réalisation du Cahier des charges	1							
Etat de l'art et Design	Etat de L'art	2							
	Maquettage	1							
Developpement	Apprentissage du langage React Native	1							
	Développement de l'application en fonction des exigences (voir cahier des charges)	1							
	Création des profils tests	2							
Tests	Mise en place des tests utilisateurs	1							
	Vérifier l'accessibilité et la sécurité des données de l'application	1							
	Evaluation et déploiement de l'application	3							
Evaluation	Rapport final	1							





Image 7 : Planning prévisionnel

Avec les incidents des premiers jours, notre planning a pris un retard considérable. Cependant avec la mise en place d'une réunion à approximativement toutes les deux semaines avec Monsieur Laurent MOULUQUET, notre tuteur, et la connaissance de la technologie sur laquelle nous nous sommes finalement orientés; le retard a été atténué.



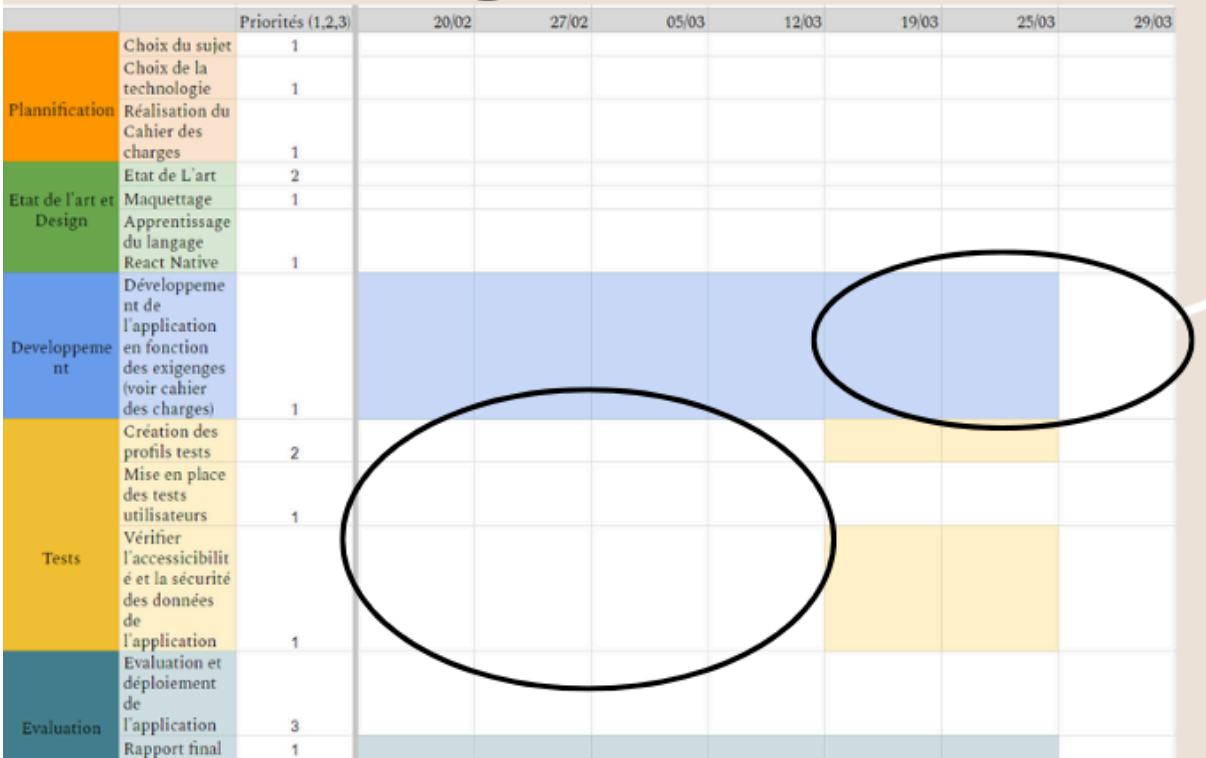


Image 8 : Planning réel

# Critique, prise de recul et bilan du projet

Pour conclure, ce projet présente de nombreuses pistes d'améliorations notamment sur l'interaction des pages. Comme faire une seule et unique page pour que le runner ou le serveur afin qu'il puisse voir le service suggéré sur le plan de salle.

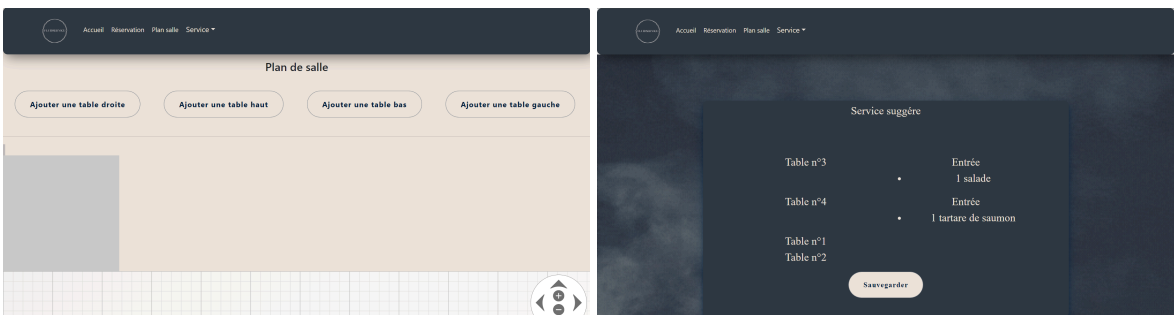


Image 9 : Page pour le plan de salle et page pour le service suggéré

Ou encore faire en sorte que le réceptionniste ajoute de lui-même les clients qui arrivent sans avoir réservé et ainsi changer depuis l’entrée l’état de la salle.



*Image 10 : Page reflétant l’occupation du restaurant*

Nous avons aussi des pistes concernant la prise de réservations qui se feraient à l’aide d’un autre outil; tout comme la mise en place d’un nouveau service chaque jour qui serait géré par un CRM (Gestionnaire de la relation client).

Ainsi, ce projet m’a permis à la fois de consolider et d’approfondir mes connaissances dans le développement d’une application en ReactJS mais également de découvrir des manières de faire différentes de celles dont j’ai l’habitude. Ce fut pour moi un projet très enrichissant.

Concernant la gestion de projet, j’ai appris à prioriser mes tâches à gérer les difficultés de mon projet individuel mais inscrit dans un projet collectif plus important, ce qui a également été très instructif. Bien qu’il soit frustrant de ne pas avoir pu terminer toutes les tâches prévues, je suis satisfaite de vous lister les exigences fonctionnelles et optionnelles que j’ai pu élaborer (selon mon cahier des charges).

### Exigences Fonctionnelles

Code	Nom de l'exigence	Description
Func_01	Identité du restaurant sur l'application	➤ Création d'un compte

Func_02	Gestion de l'état de la ou les salles du restaurant	<ul style="list-style-type: none"> <li>➤ Pouvoir visualiser en temps réel l'état du restaurant ( Quelles sont les tables réservées, les tables present, les tables libres)</li> </ul>
Func_03	Gestion du personnel	<ul style="list-style-type: none"> <li>➤ Indiquer aux serveurs le chemin le plus court en fonction de l'état du restaurant</li> <li>➤ Indiquer au réceptionniste si il y a des tables de libres ou s'il y a potentiellement une table au niveau du dessert</li> </ul>

### Exigences Optionnelles

Code	Nom de l'exigence	Description
Func_02	Gestion de l'état de la ou les salles du restaurant	<ul style="list-style-type: none"> <li>➤ Possibilité de déplacer des tables pour des grands groupes</li> </ul>

## Annexe

Support de la soutenance finale : [Lien](#)