

**Cahier des charges -Analyse**

***ADNANE Ilias***

***BOISSEL Arnaud***

***ROUMILI Mehdi***

***TERRIER Axel***

***TOFFIN Cyril***

Elective logiciel

12/12/22

Livrable 1

Table des matières

[Table des figures 4](#_Toc121752126)

[Glossaire 4](#_Toc121752127)

[1. Contexte 5](#_Toc121752128)

[2. Présentation de l'équipe et des spécialités techniques des participants 5](#_Toc121752129)

[Présentation de l'équipe et des spécialités techniques des participants 5](#_Toc121752130)

[Organisation générale du groupe 5](#_Toc121752131)

[3. Reformulation du besoin 6](#_Toc121752132)

[a) Utilisateur final 6](#_Toc121752133)

[b) Restaurateur 6](#_Toc121752134)

[c) Livreur 6](#_Toc121752135)

[d) Développeur 6](#_Toc121752136)

[e) Service commercial 6](#_Toc121752137)

[f) Service technique 6](#_Toc121752138)

[4. Identifications des dates importantes 6](#_Toc121752139)

[5. Identifications des points critiques et analyse des risques 6](#_Toc121752140)

[6. Organisation générale du groupe 6](#_Toc121752141)

[7. Questions à débattre 6](#_Toc121752142)

# Table des figures

# Glossaire

* Micro-services : def
* API : def

## Contexte

Ce projet a pour but de vous faire concevoir, réaliser, déployer, tester et utiliser une plateforme logicielle distribuée. La vocation métier de cette plateforme est la convergence et le traitement des offres commerciales dans le domaine de la restauration. Plusieurs types d'utilisateurs peuvent consommer les services offerts par cette plateforme : l'utilisateur final, le restaurateur, le livreur, le développeur tiers, le service commercial (entreprise porteuse de l'offre de convergence), le service technique (entreprise porteuse de l'offre de convergence). Il s'agit donc d'un service d'offre de restauration par internet. Elle est le concentrateur technique pour la gestion des workflows. Elle propose une gamme de services variés à destination de profils utilisateurs différents.

Ce projet comporte tous les éléments techniques étudiés lors de votre 4ème d'année d'études, mais également fait appel à l'ensemble des autres notions étudiées lors des années précédentes. Il constitue l'aboutissement technique de vos années d'études en informatique en école d'ingénieurs. Il a pour vocation d'être des plus réalistes et donc, comme dans un projet d'ingénierie en entreprise, vous disposez pour le réaliser de certaines connaissances, mais pas toutes. Il vous faudra alors, comme dans un projet réel, vous confronter à des problématiques jusqu'alors inconnues.

Ce projet, fait donc appel à l'ensemble de vos connaissances acquises depuis la première année jusqu'à ce jour. Il comporte des parties non couvertes par le bloc de manière à ressembler le plus possible à un projet réel d'entreprise. Certaines sections sont relativement bien décrites, d'autres non. Il vous faudra analyser le cahier des charges en profondeur, faire remonter les incertitudes et les traiter.

/\* AJOUTER REFORMULATION UBER\*/

## Présentation de l'équipe et des spécialités techniques des participants

### Présentation de l'équipe et des spécialités techniques des participants

Développement front end : Axel, Ilias, Mehdi

Développement back end : Mehdi, Cyril, Ilias

Gestion de la donnée : Arnaud, Cyril

Micro Service : Mehdi, Axel, Cyril

Création d’API : Mehdi, Cyril, Axel

Sécurisation d’API : Mehdi, Arnaud

Routage : Mehdi, Axel

Client lourd : Ilias, Arnaud,

### Organisation générale du groupe

Mise en place d’un Github lors des débuts de développement, mise en place d’un drive pour les documents (cahier des charges, schéma d’architecture, etc…) et répartition des différentes tâches du projet selon les affinités avec les différentes notions et technologie du projet.

Tous les outils nécessaires au bon fonctionnement du projet ont été téléchargés et installés (Visio, NodeJS, TypeScript, VueJS, Visual Studio Code, etc…) sur chaque poste de travail.

Nous avons pensé à mettre en place des points hebdomadaire de manière à assurer un avancement fluide et continue pour ainsi éviter toute forme d’imprévu. Plusieurs deadlines ont été fixées sur les fonctionnalités clefs de l’application web.

## Reformulation du besoin

### Utilisateur final

C’est celui qui commande le repas (consommateur client). Il doit pouvoir :

* Créer, modifier, supprimer, éditer son compte (CRUD)
* Créer, modifier, supprimer, éditer, payer une commande
* Éditer l'historique des commandes
* Suivre une livraison
* Parrainer un ami
* Recevoir des notifications

### Restaurateur

Il propose ses produits aux utilisateurs de l’application. Il peut bénéficier de services proposés par cette dernière afin d’accroître sa visibilité et son offre commerciale. Il doit pouvoir :

* Il propose ses produits aux utilisateurs de l’application. Il peut bénéficier de services proposés par cette dernière afin d’accroître sa visibilité et son offre commerciale. Il doit pouvoir :
* Créer, modifier, supprimer, éditer son compte (CRUD)
* Créer, modifier, supprimer, éditer un article (plat, boisson, sauce, accompagnement,)
* Créer, modifier, supprimer, éditer un menu (composition d'articles)
* Visualiser, valider une commande
* Suivre une livraison
* Éditer l'historique des commandes
* Demander des statistiques
* Parrainer un restaurateur
* Recevoir des notifications

### Livreur

Il est l'intermédiaire entre le client et le restaurateur. Tout comme ce dernier, il est indépendant (adhérent sur la plateforme). Il doit pouvoir :

* Créer, modifier, supprimer, éditer son compte
* Accepter ou refuser une livraison
* Prendre en charge et acquitter la livraison (bonus : aucune application qui permet de vérifier / Validation : Edition d'un QRcode / pour côté resto et côté client)
* Parrainer un livreur
* Recevoir des notifications (hors eMail)

### Développeur

Il peut être indépendant ou collaborateur d'une entreprise. Il a la responsabilité d'intégrer des composants logiciels proposés par la plate-forme, sur ses propres applications. Ainsi, d'autres applications clients développées par des entreprises ou des indépendants peuvent être intégrer à d'autres solutions clientes ou plateformes tierces (middleware).

* Créer, modifier, supprimer, éditer son compte (CRUD)
* Éditer les composants disponibles
* Télécharger un composant

### Service commercial

Elle est la porteuse de l'offre de convergence, mais elle est elle-même cliente de sa propre plateforme. Elle doit être en mesure :

* Éditer, suspendre, supprimer des comptes clients
* Demander des statistiques
* Disposer en temps réel de tableaux de bords de suivi de processus de commande (passation commande, acceptation commande, acceptation livraison, acquittement livraison, chiffre d'affaires transactionnel global en cours)
* Recevoir des notifications (hors eMail)

### Service technique

Il est le garant de la qualité technique des prestations offertes par la plateforme. Il doit être en mesure :

* Ajouter, supprimer des composants réutilisables
* Éditer les logs de connexions
* Éditer les statistiques de performances des serveurs et micro-services
* Éditer les logs de téléchargement des composants réutilisables
* Orchestrer les routes pour les résolutions des demandes entrantes
* Déployer de nouveaux services ou micro-services sans interruption du service client
* Recevoir des notifications (hors eMail)

## Identifications des dates importantes

13/12/22 : Présentation des livrable 0 & livrable 1

16/12/22 : Point d’avancement avant les deux semaines en entreprise

02/01/23 : Point de reprise du sujet après les deux semaines en entreprise + début de la programmation de la solution

13/01/23 : Tests des APIs et du middleware

17/01/23 : Tests de la version finale + correction des derniers bugs

20/01/23 : Livraison de la version finale de l’application et soutenance

## Identifications des points critiques et analyse des risques

### Points critiques

1. Structuration de l’architecture en API
2. Optimisation des µservices
3. Client lourd/gestion des comptes
4. Gestion de la donnée (SQL et NoSQL, différence de structure)
5. Virtualisation de la couche composants
6. Gestion de la répartition de la charge
7. Campagne de tests

### Analyse des risques

* + 1. Mauvais load balancing
    2. Endpoint mal designé
    3. Authentification partielle/manque de sécurité
    4. Non-respect de la deadline
    5. Tests ne couvrant pas l’intégralité des cas
    6. Modifications des besoins et des deadlines
    7. UI pas responsive
    8. UX pas intuitive

## Questions à débattre

Que doit faire le client lourd ?

Choix de la technologie pour chaque élément (µservice, front end, etc…)

Réflexion sur la virtualisation de la couche composants

Framework pour la couche de data ? (dashboard via librairie en JS/TS)

Choix d’un outil pour générer la documentation des APIs (Swagger ? APIDOC ?)

Que doit faire exactement le middleware

Charte graphique à définir

Faire une UI responsive (mobile ?)