



Développement d'une application complète de livraison  
de repas

## Introduction :

La société Aelion a décidé de créer une nouvelle activité : AELIOOOO ; qui simplifiera la livraison de repas aux stagiaires en manque de bien manger. Afin de limiter les coûts, Xavier fera la cuisine et Stéphanie et Sarah s'occuperont des livraisons à vélo.

Le principe est déjà bien connu. Les clients commandent des menus déjà établis au travers d'un site web et donnent l'adresse de livraison. Xavier dans sa cuisine prépare les menus et Stéphanie et Sarah iront les livrer à l'adresse du client. Le paiement se fera directement à la livraison, sans passer par le site web.

## Objectif du projet :

- Le client doit pouvoir passer une commande sur le site web (smartphone et desktop) et suivre son statut.
- Xavier doit avoir une interface dans ses cuisines pour voir les commandes passées. Une fois la commande prête, il la validera pour permettre la livraison.
- Stéphanie et Sarah devront, grâce à leur smartphone savoir que la commande est disponible chez Xavier. La commande récupérée, elles devront dire que la commande est en cours de livraison. Elles valideront la livraison une fois que le client a payé.

## Contraintes :

Les technos suivantes devront être obligatoirement utilisées :

- Angular
- Qt
- Flask ou Django
- JSON

Le protocole de communication devra être standardisé et utilisé par tous.

## Planning du projet :

M	J	V	L	M
1	2	3	3	3
4	5	5	6	7

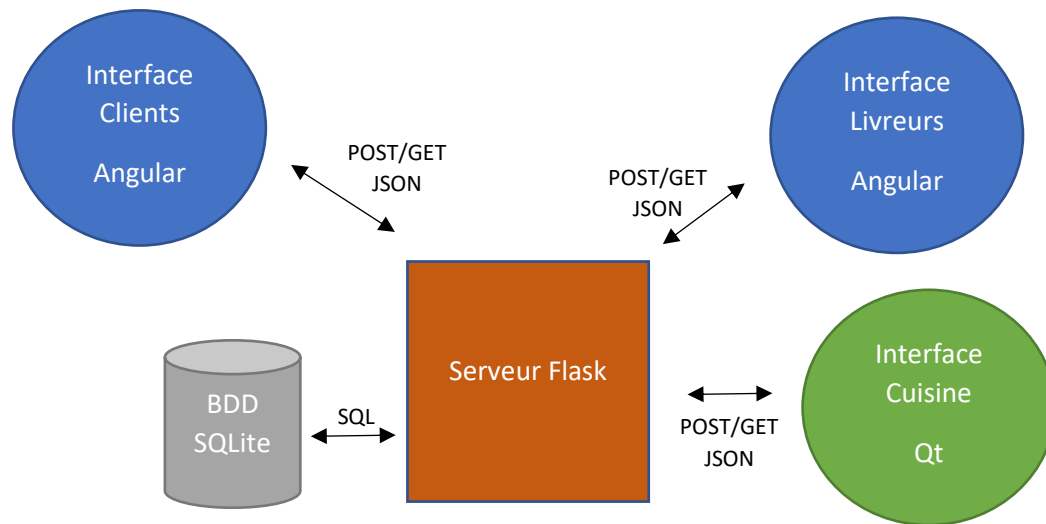
1. Présentation du sujet, étude du problème
2. Détermination des protocoles et conception et objectifs
3. Développements première partie
4. Tests d'intégration généraux et corrections mineures
5. Développement des améliorations
6. Tests d'intégration généraux et corrections mineures
7. Présentation

Lors des deux premières phases du projet :

- Définition de vos objectifs du projet.
- Ce temps est consacré uniquement à la conception de l'application et des protocoles (les PC ne devront pas être utilisés pour coder). Cette conception devra être écrite et sera validée par le formateur.

L'intégration (la communication entre les différents outils) est la partie la plus sensible du projet, pour garantir votre réussite, faites des tests d'intégration régulièrement.

## Architecture du projet



## Cas d'utilisation

1. Le client passe sa commande (POST vers serveur)
2. Le cuisinier valide et prépare la commande (polling GET POST vers serveur)
3. Le cuisinier déclare la commande prête (POST vers serveur)
4. Le livreur voit que la commande est prête et valide la réception de la commande (polling GET vers serveur et POST vers serveur)
5. Le livreur livre et valide le paiement de la commande (POST vers serveur)

## Protocole de communication

Enum des états des commandes :

1 :Nouveau, 2 :En cours de préparation, 3 :Prêt, 4 :En cours de livraison, 5 :Livré

Le client récupère ses commandes :

Requête GET @/commandes ?nomClient=Baratoux  
Réponse JSON {'commandes':[{'id':12,'etat':4},...]}

Le client passe une commandes :

Requête POST @/commandes  
JSON {'nom':'Baratoux','adresse':'Rue machin','contenu':{'....'}}  
Réponse JSON {'retour':'OK'}

La cuisine récupère ses commandes :

Requête GET @/commandes/cuisine  
Réponse JSON {'commandes':[{'id':12,'nom':'Baratoux','etat':4,'contenu':{'....'}},...]}

La cuisine met la commande à jour :

Requête PUT @/commandes  
Réponse JSON {'id':12,'etat':3}

Le livreur récupère ses livraisons :

Requête GET @/commandes/livreur  
Réponse JSON {'commandes':[{'id':12,'nom':'Baratoux','etat':4},...]}

La livreur met la commande à jour :

Requête PUT @/commandes  
Réponse JSON {'id':12,'etat':5}