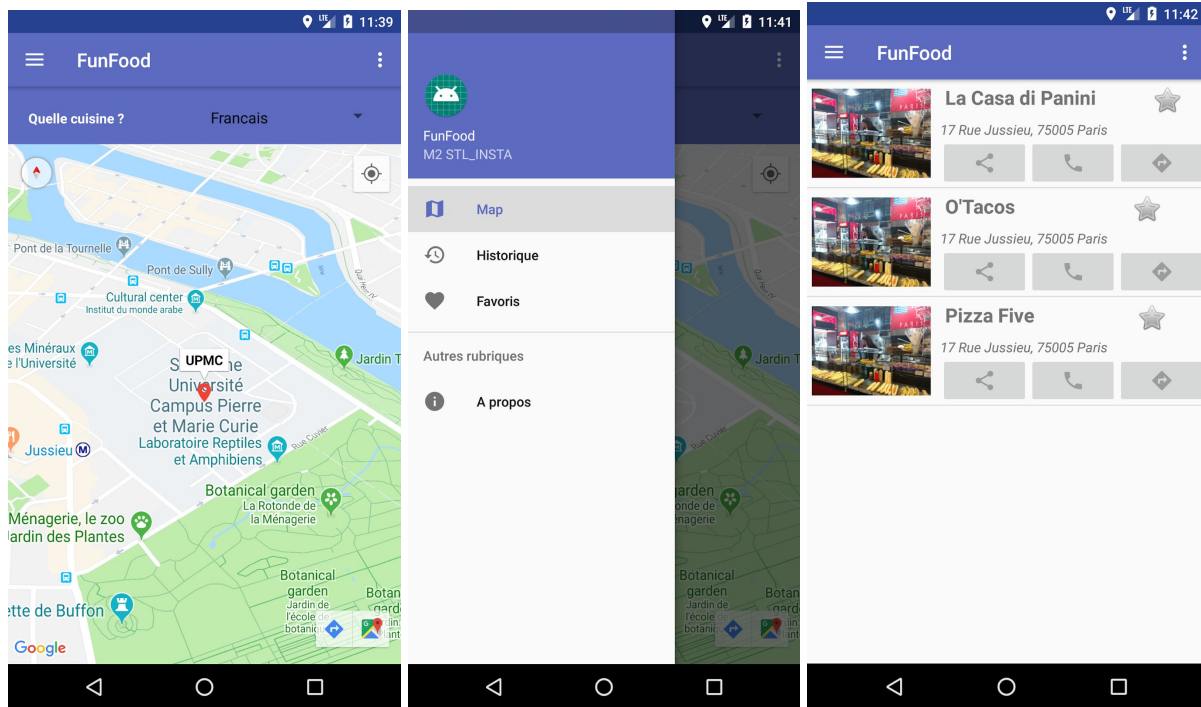


Mini rapport du projet TPALT

Partie UI

L'application FunFood est conçue dans l'objectif de nous permettre de trouver les restaurant autour de la position actuelle selon le type de cuisine qu'on aura choisi.



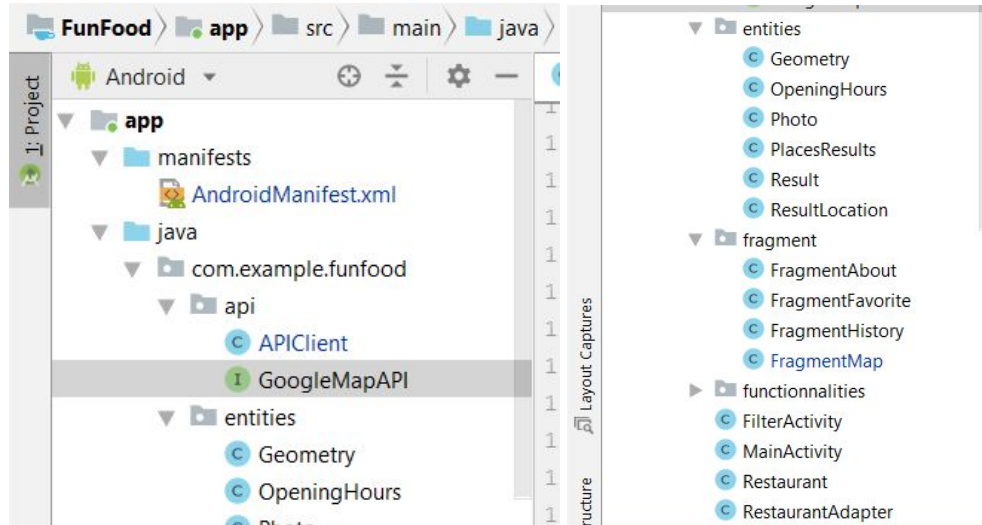
Partie Fonctionnelle

L'application nous permet de :

1. Afficher la position de Sorbonne Université - campus Jussieu dès le démarrage, si on clique sur le position, il y aura la proposition d'itinéraire et chercher sur Google Map.
2. Se localiser sur notre position actuelle en appuyant le bouton en haut à droite.
3. Chercher les restaurants autour de position actuelle avec le rayon de 5000 mètres, en utilisant le service **Google Maps Plateforme - Place API**.
4. Ajouter les restaurants favoris et les restaurants récents (**En cours**)
5. Appeler les restaurants
6. Partagez les informations avec des amis (SMS, Whatsapp, etc)

Partie Qillin ZHANG

Je travaille plutôt sur la partie back-end, le projet est créé comme un navigation drawer Android projet, organisation du projet est comme ci-dessous :



1, MainActivity contient plusieurs fragment, comme FragmentAbout, FragmentFavorite, etc.

2, Pour la partie de Google Map, j'utilise le **service d'Android** au lieu de service Google pour obtenir la position, en plus, chaque fois le fragment Map est créé, je vais comparer la position GPS et la position Internet, et prendre le meilleur position, si on n'arrive pas obtenir ces deux positions, je vais prendre la dernière position. Les détails de toutes ces théories sont dans les fonctions **getMyLocation()** et **getLastBestLocation()**.

3, J'utilise Spinner pour organiser les différents types de cuisine, pour manipuler les événements de clique et choisit de ce Spinner, j'utilise setOnItemSelectedListener(). Cette partie est compris dans la fonction **void updateSearch**.

4, Pour envoyer la requête afin de rechercher les restaurants, j'utilise le service **Google Maps Plateforme - Place API**.

5, Pour organiser l'envoi et la réception de la requête, j'utilise [Retrofit](#) comme le client HTTP. Les classes de l'envoi de requête se trouvent dans le package **api**, les classes de réception de réponse sont dans le package **entities**.

6, Une fois on reçoit les réponses, à partir de PlaceID, je récupère les latitude, les longitudes de chaque restaurant, je construis un **ArrayList<MarkerOptions> markerList** pour sauvegarder les Marker, qui peut nous servir à afficher sur le Map.

7, Pour afficher les restaurants trouvés sur le map, j'obtiens le **SupportMapFragment** à partir de context, puis **mapFragment.getMapAsync(new OnMapReadyCallback())** pour mettre à jour le map.

Partie Dimitri NGOYI

Pendant le projet, j'ai réalisé la plupart des layouts qui n'ont pas été générés automatiquement par Android Studio : **layout_details** , **layouts_filter**, **fragment_favorite**, **fragment_history**.

Pour les faire, j'ai utilisé essentiellement des **LinearLayout** parce que j'avais plus de facilité à les manipuler .

J'ai ajouté une classe **Restaurant** pour pouvoir gérer les restaurant que nous récupérons à partir de l'API.

J'ai ajouté une classe **RestaurantAdapter** pour pouvoir afficher les restaurants sous la forme du **layout_details**.

J'ai ajouté la possibilité de partager les informations d'un restaurant par SMS, Whatsapp, Twitter, etc.

J'ai ajouté la possibilité d'appeler le restaurant à partir du numéro récupéré via l'API.

J'ai ajouté une classe **SharedPreferences** pour pouvoir stocker les différents restaurants favoris mais l'appel aux fonctions de cette classe font crasher l'application et je n'ai pas encore su régler le problème.