

# Trouvez un restaurant pour déjeuner avec vos collègues

MISSION   GUIDE MENTOR   COURS   RESSOURCES

 100 heures

Mis à jour le mercredi 6 octobre 2021

## Introduction

Félicitations ! Vous venez de décrocher votre premier emploi en tant que développeur Android. La société que vous allez rejoindre est une start-up en plein essor, spécialisée dans la reconversion professionnelle des stars du show-biz. En effet, lorsque vous étiez venu pour votre premier entretien, il vous semblait bien avoir croisé quelques têtes connues...

C'est le jour J, vous arrivez dans vos nouveaux locaux, l'ambiance dynamique de la start-up vous motive déjà à donner le meilleur de vous-même. Après une dure matinée de travail, il est grand temps d'aller déjeuner. Cela tombe bien, il paraît que le quartier regorge de petits restaurants et fast-food atypiques. Lorsque vous relevez la tête, vous constatez que c'est un peu la pagaille : vos collègues peinent à se mettre d'accord sur le choix du restaurant. Votre mentor, Henry, vient vous voir et vous dit : *"Viens, je t'invite à déjeuner dans un resto sympa, j'ai une super idée pour toi"*.

Vous mourez de faim. A peine avez-vous eu le temps d'ouvrir la carte des plats que votre mentor poursuit : *"Tu vois, c'est la même chose tous les midis. Chacun a ses idées de restaurant, les gens ont du mal à se mettre d'accord, parfois plusieurs personnes se retrouvent au même endroit sans s'être concertés alors qu'ils voulaient manger la même chose, etc"*. Le burger végétarien vous dit bien, mais pas le temps de vous prononcer sur votre choix que Henry continue : *"Il est temps de fédérer tout ce beau monde, et tu tombes à pic : j'ai débloqué un budget pour que tu nous développes une super application mobile que j'ai baptisée Go4Lunch"*.

## L'application Go4Lunch

L'application Go4Lunch est une application collaborative utilisée par tous les employés. Elle permet de rechercher un restaurant dans les environs, puis de sélectionner celui de son choix en faisant part à ses collègues. De la même manière, il est possible de consulter les restaurants sélectionnés par les collègues afin de se joindre à eux. Un peu avant l'heure du déjeuner, l'application informe les différents employés pour les inviter à rejoindre leurs collègues.

### Back-end

Pour fonctionner correctement, l'application mobile a besoin de dialoguer avec un serveur, plus communément appelé [back-end](#). Afin de simplifier l'implémentation, Go4Lunch se repose sur le back-end [Firebase](#) proposé par Google. Ce service permet de gérer très facilement :

- Les comptes utilisateur ;
- L'authentification des utilisateurs via des services tiers (dont Facebook et Google, bien évidemment) ;
- La sauvegarde des données ;
- L'envoi de messages Push.

Vous devrez posséder un compte Google pour pouvoir accéder à Firebase. Vous devrez ensuite [installer la dépendance Firebase dans votre projet Android Studio](#).

### Connexion

L'accès à l'application est restreint : il est impératif de se connecter avec un compte Google ou Facebook.



L'écran de connexion

## Écran d'accueil

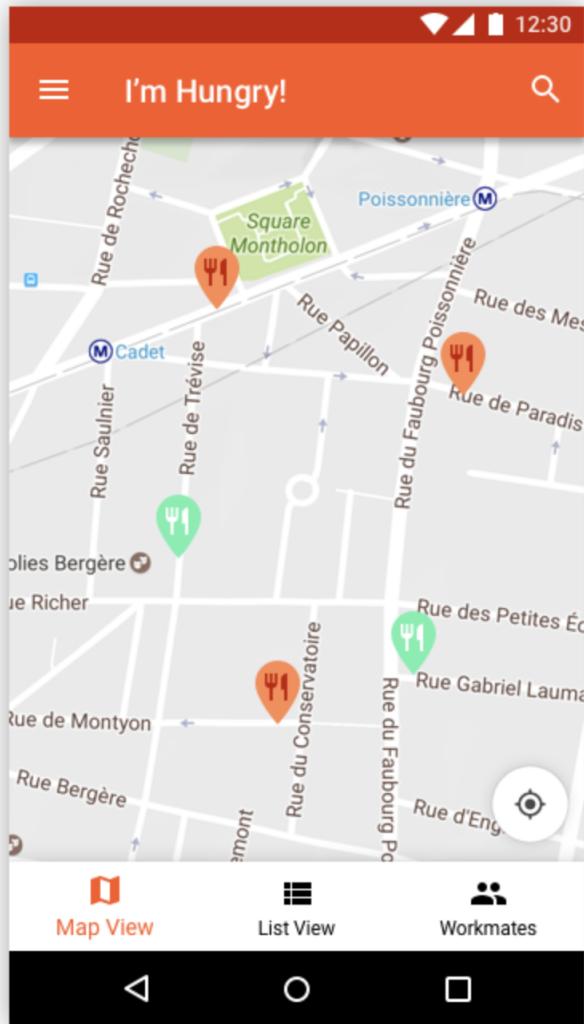
L'application est composée de trois vues principales, accessibles grâce à trois boutons situés en bas de l'écran :

- La vue des restaurants sous forme de carte ;
- La vue des restaurants sous forme de liste ;
- La vue des collègues qui utilisent l'application.

Une fois l'utilisateur connecté, l'application affiche par défaut la vue des restaurants sous forme de carte.

## Vue des restaurants sous forme de carte

L'utilisateur est automatiquement géo-localisé par l'application, afin d'afficher le quartier dans lequel il se trouve. Tous les restaurants des alentours sont affichés sur la carte en utilisant une punaise personnalisée. Si au moins un collègue s'est déjà manifesté pour aller dans un restaurant donné, la punaise est affichée dans une couleur différente (verte). L'utilisateur peut appuyer sur une punaise pour afficher la fiche du restaurant, décrite plus bas. Un bouton de géolocalisation permet de recentrer automatiquement la carte sur l'utilisateur.



L'écran de cartographie des restaurants



Vous pouvez pour cela utiliser l'API de cartographie de votre choix. Pour vous aider, voici une liste d'API de cartographie disponibles gratuitement :

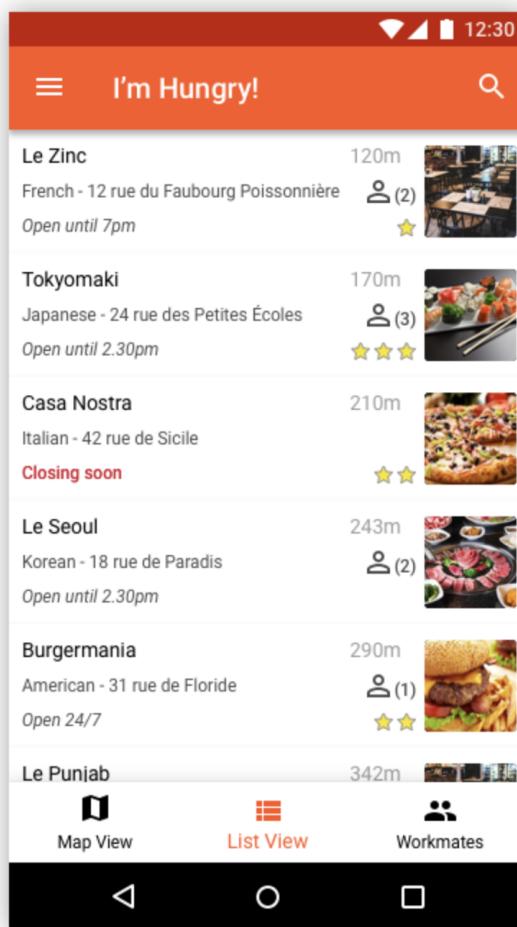
- [Overpass API](#)
- [OpenLayers](#)
- [LeafletJS](#)
- [MapQuest](#)
- [Mapbox](#)

Il est aussi tout à fait possible d'utiliser les APIs [\*\*Google Maps Android\*\*](#) et [\*\*Google Places for Android\*\*](#) mais il est important de noter que celles-ci sont facturées. Cependant, un Crédit mensuel de 200 \$ est attribué aux développeurs. Le crédit offert n'autorise qu'un nombre limité d'appels à l'API (70 000 requêtes). Pour ne pas dépasser ce nombre, vous pouvez ne lancer la requête qu'après avoir saisi 3 caractères minimum. Vous devrez également stocker la clé de [\*\*l'API de Google Places\*\*](#) dans votre fichier **gradle.properties** ([ce lien](#) peut vous aider).

## **Vue des restaurants sous forme de liste**

Cette vue permet d'afficher le détail des restaurants qui se situent sur la carte. Pour chaque restaurant, les informations suivantes sont affichées :

- Le nom du restaurant ;
- La distance du restaurant par rapport à l'utilisateur ;
- Une image du restaurant (si disponible) ;
- Le type de restaurant [optionnel] ;
- L'adresse du restaurant ;
- Le nombre de collègues qui se sont déclarés intéressés à y aller ;
- Les horaires d'ouverture du restaurant ;
- Le nombre d'avis favorables sur ce restaurant (entre 0 et 3 étoiles).



L'écran de liste des restaurants

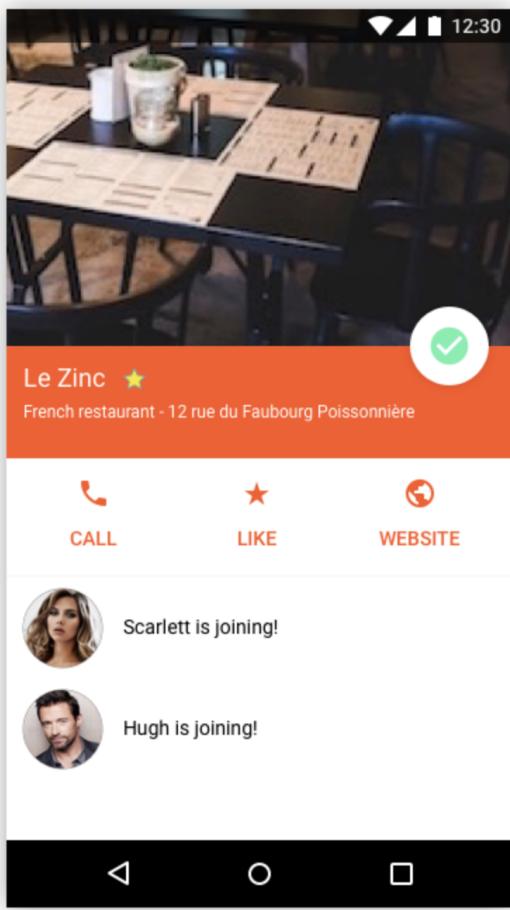


À vous de déterminer la règle pour calculer le nombre d'étoiles à afficher !

## Fiche détaillée d'un restaurant

Lorsque l'utilisateur clique sur un restaurant (depuis la carte ou depuis la liste), un nouvel écran apparaît pour afficher le détail de ce restaurant. Les informations affichées reprennent celles de la liste, et également :

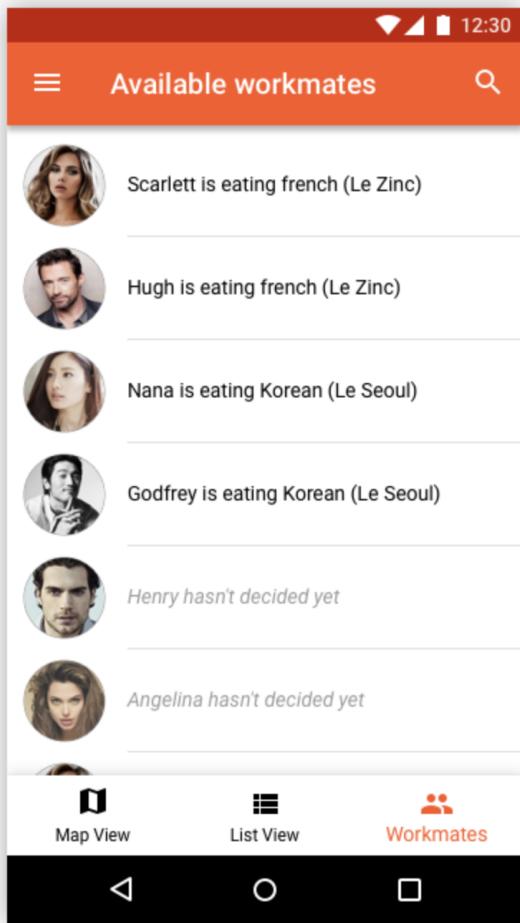
- Un bouton permettant d'indiquer son choix de restaurant ;
- Un bouton permettant d'appeler le restaurant (sous réserve qu'un numéro soit disponible) ;
- Un bouton permettant d'aimer le restaurant (Like) que vous stockerez sur Firebase ;
- Un bouton permettant d'accéder au site du restaurant (sous réserve que le restaurant possède un site) ;
- La liste des collègues déclarés intéressés à aller déjeuner dans ce restaurant. Si aucun collègue ne s'est manifesté, aucune liste n'est affichée.



L'écran de détail d'un restaurant

## Liste des collègues

Cet écran affiche la liste de tous vos collègues, avec leur choix de restaurant. Si un collègue a choisi un restaurant, vous pouvez appuyer dessus (sur l'écran, pas sur votre collègue) pour afficher la fiche détaillée de ce restaurant.



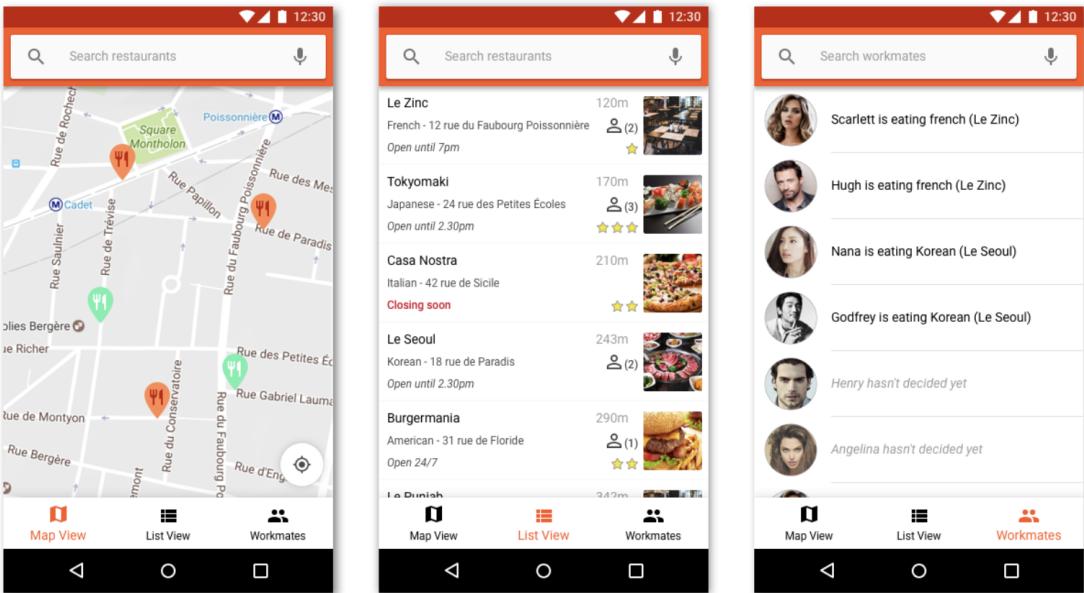
L'écran listant les utilisateurs de l'application

## Fonctionnalité de recherche

Sur chaque vue, une loupe située en haut à droite de l'écran permet d'afficher une zone de recherche. Cette recherche est contextuelle, et met automatiquement à jour les données de la vue correspondante. La recherche s'effectuera uniquement sur les noms des restaurants.



Vous n'êtes pas obligé d'implémenter la fonctionnalité de recherche pour les collègues, car cela est un peu délicat sur Firebase (il faut passer par un service tiers comme Algolia).



La fonctionnalité de recherche sur les différents écrans



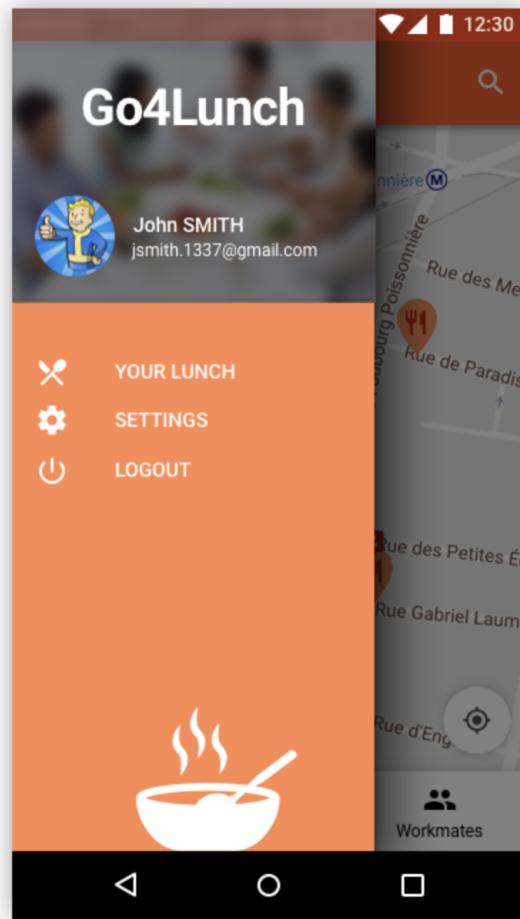
Vous pouvez utiliser l'API [Place Autocomplete](#)

---

## Menu

En haut à gauche se situe un bouton de menu. Non, ce n'est pas le menu du déjeuner. En cliquant dessus, un menu latéral s'affiche, avec les informations suivantes :

- Votre photo de profil ;
- Votre prénom et votre nom, au cas où vous ne sachiez plus comment vous vous appelez ;
- Un bouton permettant d'afficher le restaurant où vous avez prévu d'aller déjeuner ;
- Un bouton permettant d'accéder à l'écran des paramètres (pour configurer par exemple la gestion des notifications) ;
- Un bouton permettant de vous déconnecter et de retourner à l'écran de connexion.



## Notifications

Un message de notification devra être automatiquement envoyé à tous les utilisateurs qui ont sélectionné un restaurant dans l'application. Le message sera envoyé à 12h. Il rappellera à l'utilisateur le nom du restaurant qu'il a choisi, l'adresse, ainsi que la liste des collègues qui viendront avec lui.

## Traduction

Vos collègues étant de toutes les nationalités, vous devrez à minima proposer une version française et anglaise de l'application.

## Fonctionnalité complémentaire

Vous avez très probablement une tonne d'idées pour améliorer l'application existante, comme :

- Intégrer un chat dans l'application, permettant de dialoguer avec vos collègues ;
- Intégrer les modes d'authentification suivants : Twitter, identifiant et mot de passe ;
- Trier les restaurants suivant différents critères (proximité, type, nombre d'étoiles, etc).

Choisissez la fonctionnalité de votre choix parmi les trois proposées puis développez-la.

## Contraintes

Le langage utilisé devra être Java

## Soutenance

Vous transmettrez tous vos livrables au mentor qui vous fera passer la soutenance, quelques jours avant celle-ci, afin qu'il puisse en prendre connaissance.



Il vous est demandé de vous mettre en situation réelle : en effet, il s'agit **d'un rendez-vous** professionnel. Vous vous adresserez à un client qui vous a contacté pour la réalisation d'une application mobile et vous échangez donc avec lui dans le cadre de ce projet.

La soutenance, d'une durée maximum de 30 minutes, se déroulera en deux parties :

- Partie 1 – 25 minutes : **Simulation d'une réunion professionnelle**  
*L'évaluateur jouera le rôle du client* (il pourra vous interrompre pour vous poser des questions)
  - [~ 5 minutes] Vous présenterez votre application mobile (à l'aide de la documentation)
  - [~ 20 minutes] Vous réaliserez une « relecture » avec le client de votre code afin que ce dernier le valide
- Partie 2 – 5 minutes : **Retour sur la soutenance**
  - Échange de questions/réponses : le mentor pourra revenir sur certains points pour vous questionner sur vos choix
  - L'évaluateur vous fera un retour sur votre prestation en soutenance

## Consignes



Rappel : si vous utilisez les APIs **Google Maps Android** et **Google Places for Android** celles-ci sont facturées. Le crédit offert par Google n'autorise qu'un nombre limité d'appels à l'API. Pour ne pas dépasser ce nombre, vous êtes autorisé à ne lancer la requête qu'après avoir saisi 3 caractères minimum. Ceci n'est pas une obligation mais bien une liberté d'implémentation qui vous est accordée. De plus, faites attention à ne pas faire des requêtes par erreurs, à cause d'une boucle infinie ou en laissant tourner votre code trop longtemps. Ceci pourrait vous coûter très cher en requêtes à l'API.

Vous devez impérativement respecter les consignes suivantes pour que votre projet soit valide :

- Votre code est hébergé sur Github avec des commits les plus fins possible et un historique cohérent ;
- Votre code est entièrement écrit en anglais (commentaires, variables, méthodes, etc) ;
- Le projet compile correctement et ne contient aucun message d'avertissement ou d'erreur ;
- Votre application contient des tests unitaires qui couvrent la majorité de la logique de votre code ;
- L'application doit s'afficher correctement sur toutes les tailles de téléphone Android en mode portrait ;
- L'application doit supporter Android 4.4 et les versions supérieures.

## Livrables

Pour ce projet, vous fournirez un dossier compressé en ZIP contenant le projet **Android Studio**, bonus inclus.



Pour faciliter votre passage au jury, déposez sur la plateforme, dans un dossier nommé "P7\_nom\_prenom", tous les livrables du projet. Chaque livrable doit être nommé avec le numéro du projet et selon l'ordre dans lequel il apparaît, par exemple "P7\_01\_projetAndroid".

## Compétences évaluées



Modifier les paramètres d'un projet Android



Définir et appliquer un thème global



Afficher du contenu multi-langue



Créer, gérer et afficher des fragments d'interface



Créer et afficher un menu



Afficher une liste de données avec une RecyclerView et un Adapter