

8INF872 - Programmation de plateformes mobiles

David Levayer  
Corentin Ricou

# Sommaire



Introduction

Concepts clés

Intégration du contexte

Réalisations

Améliorations possibles

Conclusion

# Introduction

- Les téléphones mobiles ont évolué pour devenir bien plus qu'un simple outil de communication
- L'utilisateur accède fréquemment aux applications de son smartphone
- Android donne accès à un market d'applications bien fourni
- Applications dans le domaine du tourisme : domaine sûr et en plein essor (avec le développement de nouvelles clientèles)



« The world is a book, and those who do not travel only read the first page »

# Concepts clés

- Une application Material Design
- Une application simple et intuitive
  - navigation drawer
  - réglages confinés dans le fragment « Paramètres »
  - l'application décide pour vous (grâce au contexte)
- Marché cible
  - entre 20 et 50 ans
  - à l'aise avec un smartphone mais sans connaissances techniques
  - aime voyager / découvrir / explorer
- Objectifs : découvrir le monde qui nous entoure, en proposant des activités proches et adaptées



# Étude de l'existant

- Ce genre d'applications existe déjà sur le Play Store
- Par exemple
  - Field trip : + 1 000 000 téléchargements
  - Trip advisor : + 100 000 000 téléchargements
- Donc le marché est réel !
- Il faut se démarquer de la concurrence
- On remarque que ces solutions n'utilisent pas les informations relatives au contexte (mis à part la localisation)



# Concepts clés

- Problème : il faut disposer de données pertinentes et abondantes
- Solution : utilisation d'APIs
  - Google Maps pour la carte
  - Google Places pour les informations sur les lieux
  - Facebook API pour enrichir et personnaliser le contenu
- Avantages des APIs
  - offrent des services difficilement réalisables autrement
  - accès à beaucoup, beaucoup d'informations
  - utilisation répandue : ressources et communautés importantes
  - offertes par deux géants du Web : garanti d'une certaine continuité (mise à jours au fil des versions d'Android, ajout de contenu, ajout de services, etc.)

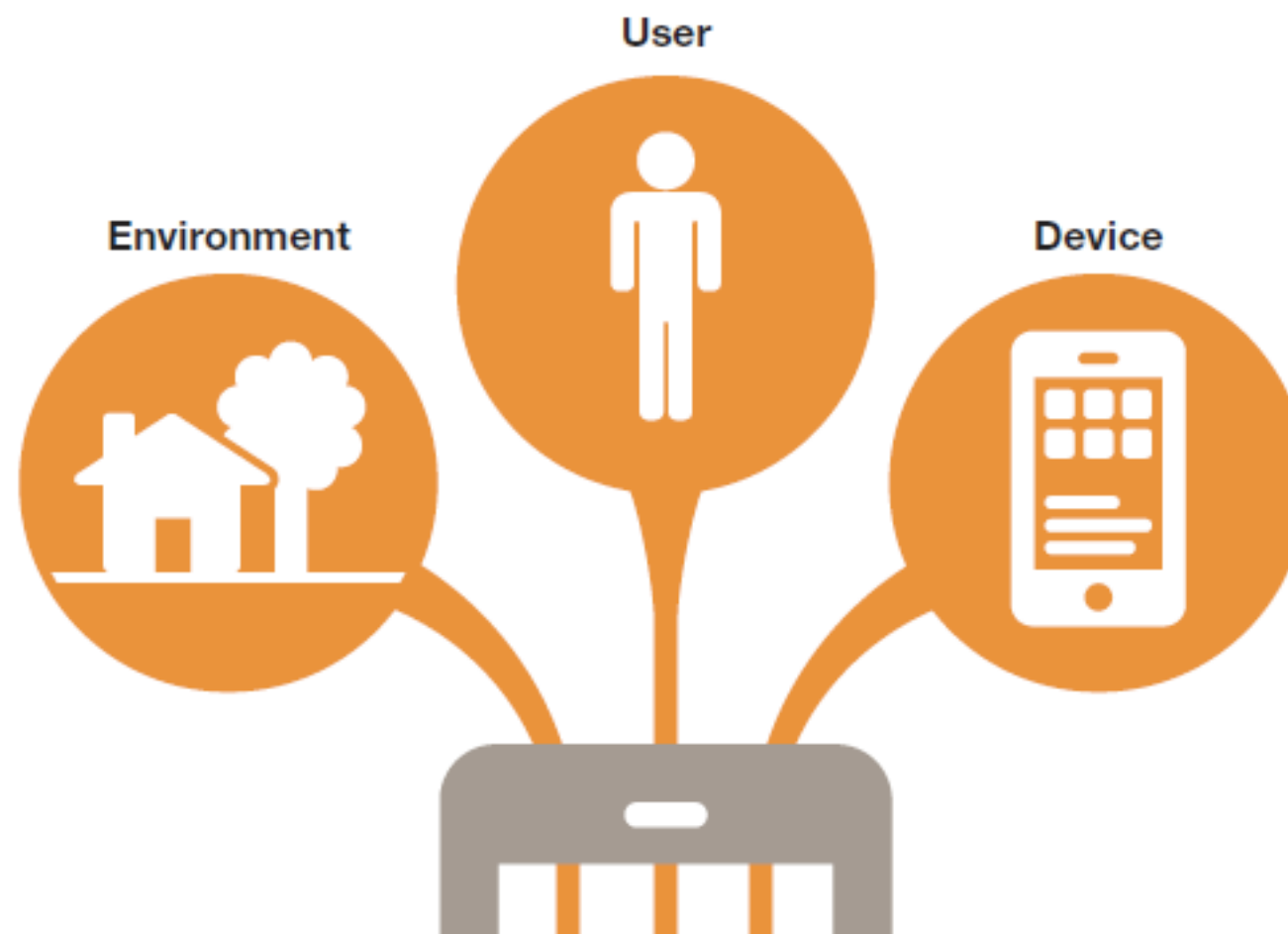
# Concepts clés

- Inconvénients des APIs
  - pas vraiment moyen de faire sans
  - forte dépendance vis-à-vis d'un service extérieur
  - quotas de demandes pour Google Places
  - ATTENTION :
    - requièrent l'activation de capteurs / composants gourmands en énergie : GPS, Wifi, réseaux de données mobiles
    - gourmandes en échange de données ; le téléchargement des informations peut alourdir la facture...



# Intégration du contexte

- Comment se démarquer de la concurrence ?
- Intégration du contexte pour guider l'utilisateur et trier les informations
- Système plus riche et automatique (solution plus intéressante qu'un simple filtre / tri par catégories)





# Intégration du contexte

- Utilisation de la position de l'utilisateur (base indispensable)
- Récupération (et publication) d'événements publiés sur Facebook
  - L'utilisateur se connecte à son compte à l'intérieur de l'app
  - Il crée et partage ses événements
  - les événements sont affichés sur la carte
- Récupération d'informations liées aux goûts de l'utilisateur (API Facebook)
  - goûts musicaux : événements, concerts
  - goûts culinaires : type de restaurants, habitudes alimentaires (bio)
  - goûts artistiques : films, musées, expositions, monuments
- Facebook dispose d'une multitude d'informations pertinentes ; et pour cause, c'est vous qui les avez renseignées !



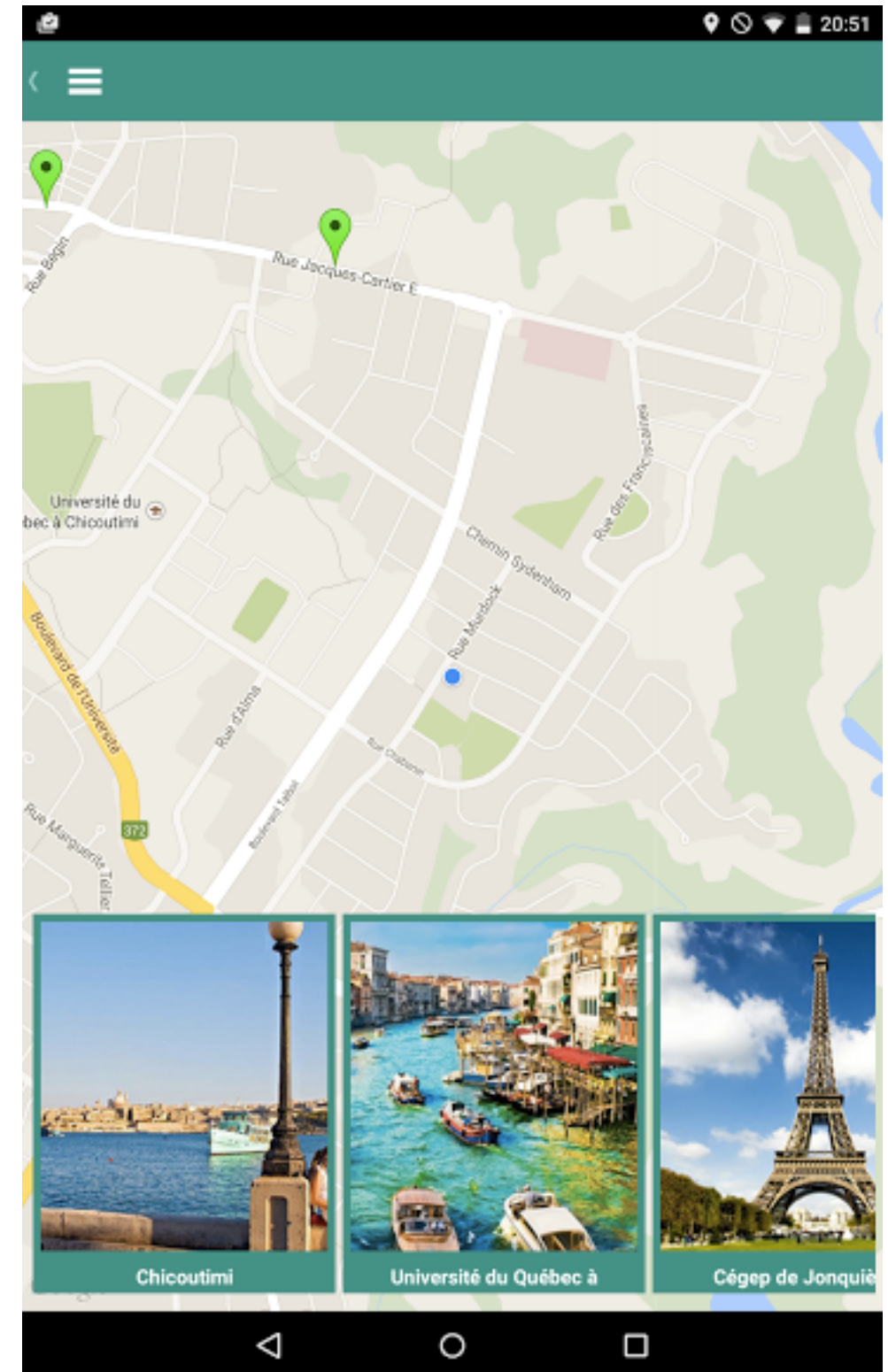
# Intégration du contexte

- Utilisation de l'heure
  - Simple mais efficace
  - Par exemple : restaurants favorisés sur les heures des repas, cinéma et bars le soir, etc.
  - Ne pas afficher les lieux fermés : musée le dimanche ou le soir, fermetures saisonnières, etc.
- Récupération de la météo
  - retrait des activités de plein air quand il pleut !
  - Mise en valeur des cinémas, musées, etc.
- Calendrier de l'appareil
  - Suggestion en fonction du temps disponibles (contraintes professionnelles) : pas de musée en 30 minutes...



# Réalisations

- Application from scratch, sauf pour les APIs
- Utilisation de Git (branches parallèles)
- Intégration de Google Maps et Google Places
  - problème dans Places : récupération des photos
- Intégration de Facebook API
  - problème : récupération de la position bloquée par mesure de sécurité
  - solution : création d'un événement via Nomad (informations complémentaires stockées dans l'app)
- Intégration d'une base de données SQLite
  - stockage de lieux d'intérêts personnels et des informations complémentaires
  - stockage des paramètres de l'application



# Améliorations possibles

- Finitions sur les fonctionnalités principales
- Gestion des paramètres
- Finalisation des normes Material Design
- Intégration des capteurs virtuels (heure, météo)
- Stabilisation des APIs Places et Facebook dans les cas défavorables (absence de connexion, manque de données, etc.)
- Benchmark et analyse des performances (indice de consommation)
  - attention aux quotas



« Strive for continuous improvement, instead of perfection » - Kim Collins

# Conclusion

- Application à fort potentiel
- Il faut utiliser le contexte (capteurs réels et surtout virtuels) : c'est le principal atout !
- Miser sur un design simple et sur une application facile à utiliser
  - en accord avec le public cible
  - même si certaines fonctionnalités avancées sont retirées / masquées
  - gestion multi-device (adaptation du design)



# Questions



David Levayer  
Corentin Ricou