

# Programmation Mobile

## Reverleaf

DIAB Ingo  
KACI Chamy

# Sommaire

1. Présentation
2. Architecture
  - a. ViewManager
  - b. FirebaseManager
  - c. PayPalManager
3. Features
  - a. Inscription/Connexion avec Google
  - b. Inscription manuelle
  - c. Connexion manuelle
  - d. Abonnements
  - e. Hub
  - f. Homepage
  - g. Recherche d'événements
  - h. Mes événements
  - i. Catégories
  - j. Créer un événement
  - k. Supprimer un événement
  - l. Consulter un événement
  - m. Inscription à un événement
  - n. Ajouter un événement aux favoris
  - o. Chercher un événement équivalent
  - p. Groupes
4. Blocages
5. Conclusions & Améliorations possibles

# Présentation

- Gestion d'évènements sportifs et culturels

# Présentation

- Gestion d'évènements sportifs et culturels
- Code en Java/XML et testé sur un Pixel 6 API 33

# Présentation

- Gestion d'évènements sportifs et culturels
- Code en Java/XML et testé sur un Pixel 6 API 33
- Plusieurs API
  - Firebase Realtime Database
  - Firebase Authentication
  - Firebase Storage
  - Picasso
  - Paypal Checkout
  - Google Places

# Architecture

- Utilisations de classes pour les datas
  - UserData
  - EventData
  - AbonnementData

# Architecture

- Utilisations de classes pour les datas
  - UserData
  - EventData
  - AbonnementData
- Utilisations de managers
  - ViewManager
  - FirebaseManager
  - PayPalManager

# Architecture - ViewManager

- Méthodes relatives aux Views
  - Popup
  - findViewById
  - onClickListener



# Architecture - ViewManager

- Méthodes relatives aux Views
  - Popup
  - findViewById
  - onClickListener
- Code plus propre
  - On appelle BindOnClick au lieu de onClickListener(){}

# Architecture - ViewManager

- Méthodes relatives aux Views
  - Popup
  - findViewById
  - onClickListener
- Code plus propre
  - On appelle BindOnClick au lieu de onClickListener(){}
- Moins de réplication de code
  - Code plus robuste
  - Code plus simple à débbugger/écrire
  - Code plus visible

# Architecture - FirebaseManager

- Méthodes relatives aux échanges avec Firebase
  - Load événements
  - Load images
  - Connexion/Inscription

# Architecture - PayPalManager

- Méthodes relatives aux transaction avec PayPal
  - Initialisation des boutons PayPal
  - Facturation d'un abonnement

# Features - Inscription/Connexion avec Google

- Firebase Authentication

# Features - Inscription/Connexion avec Google

- Firebase Authentication
- Création d'un User Reverleaf avec les informations disponible sur le compte Google.

# Features - Inscription/Connexion avec Google

- Firebase Authentication
- Création d'un User Reverleaf avec les informations disponible sur le compte Google.
- Redirection vers les Abonnements

# Features - Inscription manuelle

- Champs obligatoires
  - Email
  - Téléphone
  - Mot de passe + confirmation



# Features - Inscription manuelle

- Champs obligatoires
  - Email
  - Téléphone
  - Mot de passe + confirmation
- Champs optionnels
  - Nom
  - Prénom
  - Adresse

# Features - Inscription manuelle

- Champs obligatoires
  - Email
  - Téléphone
  - Mot de passe + confirmation
- Champs optionnels
  - Nom
  - Prénom
  - Adresse
- Des règles
  - Tous les champs obligatoires remplis
  - Mail de la forme abc@def.xyz et mot de passe de 6 caractères minimum
  - Mot de passe équivalent à la confirmation
  - Accepter les conditions

# Features - Connexion manuelle

- Firebase Authentication

# Features - Connexion manuelle

- Firebase Authentication
- Email + mot de passe

# Features - Connexion manuelle

- Firebase Authentication
- Email + mot de passe
- Si l'utilisateur à un abonnement valide, redirection vers le hub sinon vers les abonnements

# Features - Abonnements

- Firebase Realtime Database (FDR) et Firebase Storage

# Features - Abonnements

- Firebase Realtime Database (FDR) et Firebase Storage
- Pour créer un abonnement
  - Créer une entrée sur FRD pour l'abonnement
  - Créer une entrée sur Firebase Storage pour l'image de l'abonnement

# Features - Abonnements

- Firebase Realtime Database (FDR) et Firebase Storage
- Pour créer un abonnement
  - Créer une entrée sur FRD pour l'abonnement
  - Créer une entrée sur Firebase Storage pour l'image de l'abonnement
- L'utilisateur clique sur l'abonnement pour ouvrir un popup



# Features - Abonnements

- Firebase Realtime Database (FDR) et Firebase Storage
- Pour créer un abonnement
  - Créer une entrée sur FRD pour l'abonnement
  - Créer une entrée sur Firebase Storage pour l'image de l'abonnement
- L'utilisateur clique sur l'abonnement pour ouvrir un popup
- L'utilisateur paye avec PayPal

# Features - Abonnements

- Firebase Realtime Database (FDR) et Firebase Storage
- Pour créer un abonnement
  - Créer une entrée sur FRD pour l'abonnement
  - Créer une entrée sur Firebase Storage pour l'image de l'abonnement
- L'utilisateur clique sur l'abonnement pour ouvrir un popup
- L'utilisateur paye avec PayPal
- L'utilisateur est redirigé vers le hub

# Features - Abonnements

- Firebase Realtime Database (FDR) et Firebase Storage
- Pour créer un abonnement
  - Créer une entrée sur FRD pour l'abonnement
  - Créer une entrée sur Firebase Storage pour l'image de l'abonnement
- L'utilisateur clique sur l'abonnement pour ouvrir un popup
- L'utilisateur paye avec PayPal
- L'utilisateur est redirigé vers le hub
- L'abonnement Premium est le seul à permettre la création d'évènements

# Features - Hub

- Activité principale

# Features - Hub

- Activité principale
- En haut à gauche, bouton off pour se déconnecter

# Features - Hub

- Activité principale
- En haut à gauche, bouton off pour se déconnecter
- Navigation barre avec 4 icônes
  - La Homepage
  - La Recherche d'événements
  - Mes événements
  - Les Groupes (pas implémentés)

# Features - Homepage

- Fragment principal de l'activité principale

# Features - Homepage

- Fragment principal de l'activité principale
- 4 listes différentes
  - Inscriptions => Les événements où l'on s'est inscrit
  - Favoris => Les événements ajoutés en favoris
  - Proximité => Les événement proche de notre géolocalisation
  - Tendances => Les 5 événements avec le plus d'utilisateurs inscrits



# Features - Homepage

- Fragment principal de l'activité principale
- 4 listes différentes
  - Inscriptions => Les événements où l'on s'est inscrit
  - Favoris => Les événements ajoutés en favoris
  - Proximité => Les événement proche de notre géolocalisation
  - Tendances => Les 5 événements avec le plus d'utilisateurs inscrits
- Demande d'accès à la proximité + bouton pour rafraîchir les événements à proximité

# Features - Homepage

- Fragment principal de l'activité principale
- 4 listes différentes
  - Inscriptions => Les événements où l'on s'est inscrit
  - Favoris => Les événements ajoutés en favoris
  - Proximité => Les événement proche de notre géolocalisation
  - Tendances => Les 5 événements avec le plus d'utilisateurs inscrits
- Demande d'accès à la proximité + bouton pour rafraîchir les événements à proximité
- Cliquer sur un événement pour le consulter

# Features - Recherche d'événements

- Rechercher selon un ou plusieurs critères
  - Titre
  - Catégories (une ou plusieurs)
  - Lieu
  - Entre Prix min et Prix max
  - Entre Date début et Date fin

# Features - Mes événements

- Regroupe tous les événements créés

# Features - Mes événements

- Regroupe tous les événements créés
- Permet de créer un événement si l'utilisateur a un abonnement Premium

# Features - Mes événements

- Regroupe tous les événements créés
- Permet de créer un événement si l'utilisateur a un abonnement Premium
- L'utilisateur choisit une Catégorie et est redirigé vers un formulaire de création de l'événement

# Features - Catégories

- Utilisation de la reflection de Java
  - On trouve la méthode à exécuter selon la catégorie de l'événement

# Features - Catégories

- Utilisation de la reflection de Java
  - On trouve la méthode à exécuter selon la catégorie de l'événement
- Enumération pour le type de carte à créer et les présets



# Features - Catégories

- Utilisation de la reflection de Java
  - On trouve la méthode à exécuter selon la catégorie de l'événement
- Enumération pour le type de carte à créer et les présets
- Créer une catégorie
  - Ajouter une nouvelle entrée à la Firebase Realtime Database avec un nom, une Uri pour l'image carrée de la catégorie et une Uri pour l'image de base de la catégorie

# Features - Catégories

- Utilisation de la reflection de Java
  - On trouve la méthode à exécuter selon la catégorie de l'événement
- Enumération pour le type de carte à créer et les présets
- Créer une catégorie
  - Ajouter une nouvelle entrée à la Firebase Realtime Database avec un nom, une Uri pour l'image carrée de la catégorie et une Uri pour l'image de base de la catégorie
  - Stocker une image pour les deux Uri dans Firebase Storage

# Features - Catégories

- Utilisation de la reflection de Java
  - On trouve la méthode à exécuter selon la catégorie de l'événement
- Enumération pour le type de carte à créer et les présets
- Créer une catégorie
  - Ajouter une nouvelle entrée à la Firebase Realtime Database avec un nom, une Uri pour l'image carrée de la catégorie et une Uri pour l'image de base de la catégorie
  - Stocker une image pour les deux Uri dans Firebase Storage
  - Dans le code, ajouter une activité/view "ConsultEvent\_NomDeLaCategorie" (doit hériter de ConsultEvent) et "CreateEvent\_NomDeLaCategorie" (doit hériter de CreateEvent) pour consulter/créer un événement de cette catégorie

# Features - Catégories

- Utilisation de la reflection de Java
  - On trouve la méthode à exécuter selon la catégorie de l'événement
- Enumération pour le type de carte à créer et les présets
- Créer une catégorie
  - Ajouter une nouvelle entrée à la Firebase Realtime Database avec un nom, une Uri pour l'image carrée de la catégorie et une Uri pour l'image de base de la catégorie
  - Stocker une image pour les deux Uri dans Firebase Storage
  - Dans le code, ajouter une activité/view "ConsultEvent\_NomDeLaCategorie" (doit hériter de ConsultEvent) et "CreateEvent\_NomDeLaCategorie" (doit hériter de CreateEvent) pour consulter/créer un événement de cette catégorie
  - Dans le code, ajouter une class "EventData\_NomDeLaCategorie" (doit hériter de EventData) et ajouter les informations spécifiques à cet événement

# Features - Catégories

- Utilisation de la reflection de Java
  - On trouve la méthode à exécuter selon la catégorie de l'événement
- Enumération pour le type de carte à créer et les présets
- Créer une catégorie
  - Ajouter une nouvelle entrée à la Firebase Realtime Database avec un nom, une Uri pour l'image carrée de la catégorie et une Uri pour l'image de base de la catégorie
  - Stocker une image pour les deux Uri dans Firebase Storage
  - Dans le code, ajouter une activité/view "ConsultEvent\_NomDeLaCategorie" (doit hériter de ConsultEvent) et "CreateEvent\_NomDeLaCategorie" (doit hériter de CreateEvent) pour consulter/créer un événement de cette catégorie
  - Dans le code, ajouter une class "EventData\_NomDeLaCategorie" (doit hériter de EventData) et ajouter les informations spécifiques à cet événement
- Pas de if/switch à modifier => plus simple

# Features - Créer un événement

- Cliquer sur “Créer un événement” dans MyEvents (si utilisateur Premium)

# Features - Créer un événement

- Cliquer sur “Créer un événement” dans MyEvents (si utilisateur Premium)
- Informations communes aux catégories et obligatoires
  - Titre
  - Description
  - Lieu
  - Date
  - Prix

# Features - Créer un événement

- Cliquer sur “Créer un événement” dans MyEvents (si utilisateur Premium)
- Informations communes aux catégories et obligatoires
  - Titre
  - Description
  - Lieu
  - Date
  - Prix
- Informations spécifiques et optionnelles
  - Chanteur/Type de musique (Concert)
  - Sport pratiqué/Durée d'un match (Sport)



# Features - Supprimer un événement

- Cliquer sur la croix blanche sur fond rouge au dessus de l'événement à supprimer

# Features - Consulter un événement

- Cliquer sur un événement

# Features - Consulter un événement

- Cliquer sur un événement
- Affichage de toutes les informations de l'événement cliqué

# Features - Consulter un événement

- Cliquer sur un événement
- Affichage de toutes les informations de l'événement cliqué
- 3 actions possibles
  - S'inscrire
  - Ajouter aux favoris
  - Rechercher un événement équivalent

# Features - Consulter un événement

- Cliquer sur un événement
- Affichage de toutes les informations de l'événement cliqué
- 3 actions possibles
  - S'inscrire
  - Ajouter aux favoris
  - Rechercher un événement équivalent
- Visualisation du nombre d'utilisateur inscrits cet événement et qui l'ont ajouté à leur favoris

# Features - Inscription à un événement

- Bouton “S’inscrire”
  - Bouton “Désinscrire” si l'utilisateur est déjà inscrit

# Features - Inscription à un événement

- Bouton “S’inscrire”
  - Bouton “Désinscrire” si l'utilisateur est déjà inscrit
- Incrémente le nombre d'inscriptions à l'événement de 1
  - Décrémenter de 1 si désinscription

# Features - Ajouter un événement aux favoris

- Bouton en forme de coeur gris
  - Bouton en forme de coeur rouge si l'événement est déjà en favoris



# Features - Ajouter un événement aux favoris

- Bouton en forme de coeur gris
  - Bouton en forme de coeur rouge si l'événement est déjà en favoris
- Incrémente le nombre d'utilisateur ayant ajouté l'événement en favoris de 1
  - Décrément de 1 si l'utilisateur l'enlève de ses favoris

# Features - Chercher un événement équivalent

- Cliquer sur certaines informations
  - Catégorie
  - Lieu
  - Date
  - Prix

# Features - Chercher un événement équivalent

- Cliquer sur certaines informations
  - Catégorie
  - Lieu
  - Date
  - Prix
- Cliquer sur rechercher pour rechercher les événements correspondants aux informations cherchées.

# Features - Chercher un événement équivalent

- Cliquer sur certaines informations
  - Catégorie
  - Lieu
  - Date
  - Prix
- Cliquer sur rechercher pour rechercher les événements correspondants aux informations cherchées.
- Exemple : Si nous consultons un événement sportif qui a lieu à Montpellier le 28/05/2023 et que le prix est 10€, si nous cliquons sur la catégorie et le lieu puis sur "Rechercher", nous aurons comme résultat tous les événements Sportif à Montpellier, peu importe la date et le prix

# Features - Groupes

- Créer un groupe
  - Ajouter un ou plusieurs amis
  - Chat textuel
  - Partage d'événement

# Features - Groupes

- Créer un groupe
  - Ajouter un ou plusieurs amis
  - Chat textuel
  - Partage d'événement
- Cliquer sur un événement ouvre les informations de l'événement (comme dans le menu Home)

# Blocages

- Beaucoup de méthodes asynchrones
  - On a réglé ce problème avec l'utilisation de Consumer<> que nous utilisons comme méthodes de callback

# Blocages

- Beaucoup de méthodes asynchrones
  - On a réglé ce problème avec l'utilisation de Consumer<> que nous utilisons comme méthodes de callback
- Utilisation de Picasso pour charger les images qui ne marche pas toujours
  - Picasso possède une weak référence sur une Target donc la Target est supprimée par le garbage collector avant le résultat de la méthode
  - Utilisation d'une classe pour encapsuler la Target et mise de la Target en attribut de classe  
=> n'a pas bien marché



# Conclusion & Améliorations possibles

- L'essentiel de l'application est présente
  - Développement des features les plus importantes en premier
  - Features pas tout le temps exploitées au mieux (abonnements)

# Conclusion & Améliorations possibles

- L'essentiel de l'application est présente
  - Développement des features les plus importantes en premier
  - Features pas tout le temps exploitées au mieux (abonnements)
- Améliorations possibles
  - Fix du problème d'image avec Picasso
  - Implémentation d'un système de cache pour éviter d'utiliser trop de bande passante