



# Application Mobile Festival

04.09.2022

---

**QUINQUIS Jean François**

M2 Master Miage

Université de la Nouvelle-Calédonie

**TRAN Duc Huy Dimitri**

M2 Master Miage

Université de la Nouvelle-Calédonie

## Table des matières

<b>Table des matières</b>	<b>1</b>
<b>Vue d'ensemble</b>	<b>2</b>
<b>Gestion de projet</b>	<b>3</b>
Définition du backlog projet	3
Outils	3
Trello	3
Github	4
Mockflow (Maquettage)	5
Android studio	6
Technologie (Front/Back)	6
Java native mobile	6
SQLite	7
<b>Application</b>	<b>9</b>
Analyse du besoin	9
Besoin utilisateur	9
Base de données	10
Schémas	10
Réalisation de la base de donnée	14
Conclusion	16
Prototyper la solution (Maquettage)	16
Développement	19
Description de l'application	19
Layout	19
Java	21
Fonctionnement général	21
Manuel d'utilisation	22
<b>Conclusion</b>	<b>26</b>
Résultats	26
Axes d'amélioration	26



## Vue d'ensemble

Ce projet à bout but de proposer une application permettant de aux participants d'un festival de se renseigner sur le planning des passages des groupes de musique.

### **Mais pourquoi est-il intéressant pour un utilisateur de posséder l'application ?**

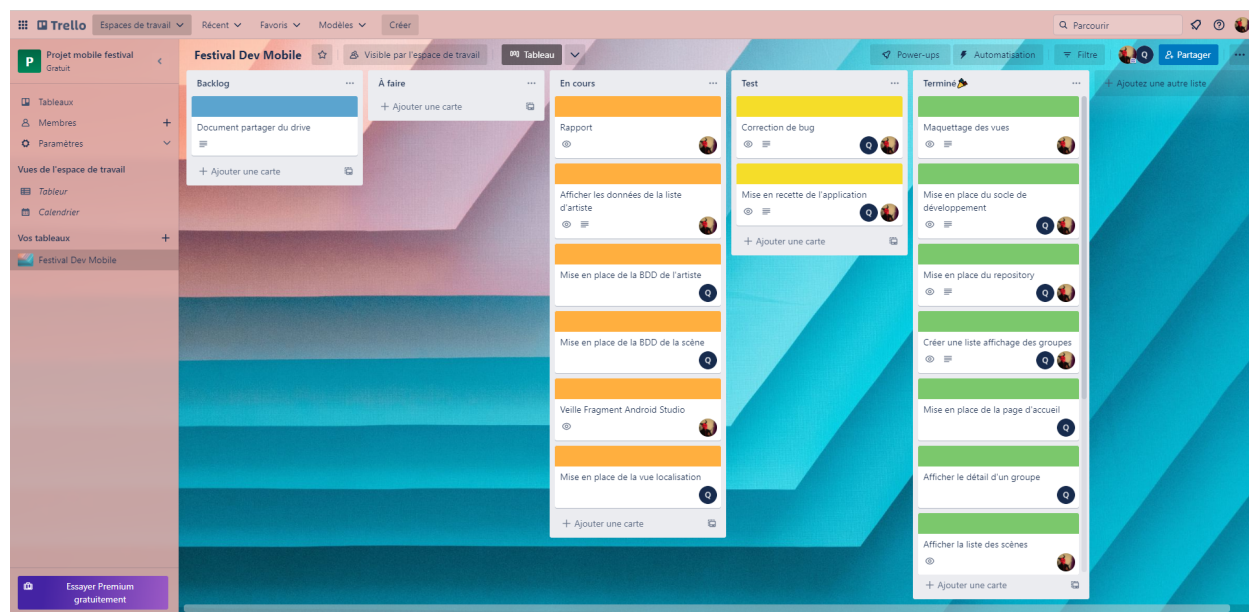
Il y a plusieurs raisons pour lesquelles il peut être intéressant de créer une application mobile pour lister les informations présentes dans un festival de musique :

- Améliorer l'expérience des visiteurs : Une application mobile peut offrir une expérience de festival plus fluide et plus agréable aux visiteurs. Ils pourront accéder à toutes les informations du festival en temps réel, comme la liste des artistes, les horaires de concerts, les plans de site, etc. Ils pourront également recevoir des notifications pour être tenus au courant des dernières mises à jour ou des changements d'horaire.
- Augmenter la visibilité du festival : Une application mobile peut être un excellent moyen de promouvoir le festival et d'attirer de nouveaux visiteurs. Les gens peuvent télécharger l'application et la partager avec leurs amis, ce qui peut contribuer à étendre la portée du festival.
- Collecter des données sur les visiteurs : Une application mobile peut être utilisée pour collecter des données sur les visiteurs, comme leur localisation, leurs préférences musicales, etc. Cela peut être utile pour améliorer l'expérience du festival et mieux cibler les futurs événements.
- Gérer les transactions : Une application mobile peut également être utilisée pour gérer les transactions liées au festival, comme la vente de billets ou l'achat de nourriture et de boissons sur place. Cela peut rendre le processus de paiement plus rapide et plus pratique pour les visiteurs.

En résumé, une application mobile pour lister les informations présentes dans un festival de musique peut être un outil très utile pour améliorer l'expérience des visiteurs, augmenter la visibilité du festival et collecter des données sur les visiteurs. Elle peut également faciliter la gestion des transactions liées au festival.

## Gestion de projet

### Définition du backlog projet



### Outils

#### Trello

Trello est une application de gestion de projet en ligne qui vous permet de planifier et de suivre les tâches de votre équipe de manière visuelle. Elle utilise une interface basée sur des "tableaux" et des "cartes", qui vous permettent de créer des listes de tâches et de les organiser en colonnes selon l'état d'avancement du projet.

Voici quelques fonctionnalités clés de Trello :

- **Création de tableaux :** Vous pouvez créer autant de tableaux que vous le souhaitez pour organiser vos projets et vos tâches. Chaque tableau peut être partagé avec d'autres membres de votre équipe, afin que tout le monde puisse travailler ensemble.
- **Création de cartes :** Vous pouvez créer des cartes pour chaque tâche ou élément de votre projet. Chaque carte peut être déplacée d'une colonne à l'autre pour suivre l'avancement du projet. Vous pouvez également ajouter des détails, des pièces jointes et des commentaires à chaque carte.



- Collaboration : Trello vous permet de travailler en temps réel avec d'autres membres de votre équipe. Vous pouvez ajouter des membres à vos tableaux, leur attribuer des tâches et suivre leur avancement.
- Intégration : Trello est compatible avec de nombreuses autres applications, comme Google Drive, Slack et Evernote. Vous pouvez ajouter ces intégrations à vos tableaux pour rendre votre workflow encore plus efficace.

## Github

GitHub est un service en ligne de gestion de code source qui permet aux développeurs de stocker et de partager leur code. Il est basé sur le logiciel de gestion de version Git, qui est un système de contrôle de version qui permet de suivre les modifications apportées au code source d'un projet.

Voici quelques-unes des fonctionnalités clés de GitHub :

- Repositories : sur GitHub, vous pouvez créer des "dépôts" ou "repositories" pour stocker votre code. Chaque dépôt peut être partagé avec d'autres développeurs, qui peuvent y contribuer en ajoutant des fichiers ou en proposant des modifications.
- Collaboration : GitHub facilite la collaboration avec d'autres développeurs en permettant de discuter de modifications et de suivre l'avancement du projet. Vous pouvez ajouter des "issues" pour signaler des problèmes ou des idées, et utiliser les "pull requests" pour proposer des modifications au code.
- Gestion de version : GitHub utilise Git pour suivre les modifications apportées au code source. Vous pouvez créer des "branches" pour travailler sur des fonctionnalités spécifiques, puis fusionner vos modifications dans la branche principale lorsque vous êtes prêt à les publier.
- Intégration : GitHub intègre de nombreux outils de développement, comme les services de build et de déploiement, les outils de test et de gestion de projet. Vous pouvez également utiliser GitHub avec d'autres services, comme Slack et JIRA, pour une gestion de projet encore plus efficace.

Git, quant à lui, est un logiciel de gestion de version qui permet de suivre les modifications apportées au code source d'un projet. Il est principalement utilisé pour le développement de logiciels, mais peut également être utilisé pour suivre les modifications apportées à tout type de fichier.

Voici quelques-unes des principales fonctionnalités de Git :

- Contrôle de version : Git suivre les modifications apportées au code source d'un projet, en enregistrant chaque version de manière indépendante. Cela vous permet



de retourner à une version précédente si vous avez besoin de faire des changements, ou de comparer les différences entre les versions.

- Suivi des modifications : Git permet de suivre les modifications apportées au code source d'un projet au fil du temps. Vous pouvez revenir à des versions antérieures de votre code à tout moment et comparer les différentes versions pour voir les changements qui ont été apportés.
- Collaboration : Git vous permet de travailler sur un projet avec d'autres développeurs de manière efficace. Vous pouvez créer des "branches" de votre projet pour travailler sur des fonctionnalités distinctes, puis fusionner vos modifications dans la branche principale lorsque vous êtes prêt à les partager.
- Gestion de conflits : Git vous permet de gérer les conflits lorsque plusieurs personnes modifient le même fichier en même temps. Il vous indique où se situent les conflits et vous permet de les résoudre manuellement.
- Hébergement : Git vous permet d'héberger votre code sur des "dépôts" en ligne, comme GitHub ou GitLab. Cela vous permet de travailler sur votre code de n'importe où et de le partager facilement avec d'autres développeurs.

## Mockflow (Maquettage)

MockFlow est une plateforme en ligne de prototypage et de design de l'interface utilisateur (UI). Elle vous permet de créer des maquettes et des wireframes de vos projets de conception de manière simple et rapide.

Voici quelques fonctionnalités clés de MockFlow :

- Création de maquettes : MockFlow vous permet de créer des maquettes de votre interface utilisateur en glissant-déposant des éléments de design prédéfinis. Vous pouvez ajouter du texte, des images et des formes à vos maquettes et les personnaliser selon vos besoins.
- Création de wireframes : MockFlow vous permet également de créer des wireframes, qui sont des versions simplifiées de votre interface utilisateur conçues pour mettre en avant les principaux éléments de navigation et de fonctionnalité.
- Collaboration : MockFlow vous permet de travailler avec d'autres membres de votre équipe en temps réel sur vos projets de design. Vous pouvez ajouter des commentaires et des notes pour discuter de vos idées et de vos choix de design.
- Intégration : MockFlow est compatible avec de nombreuses autres applications, comme Google Drive et Slack. Vous pouvez intégrer ces applications à MockFlow pour rendre votre workflow encore plus efficace.



## Android studio

Android Studio est un environnement de développement intégré (IDE) gratuit et open source conçu spécifiquement pour le développement d'applications Android. Il est développé par Google et il est largement utilisé par les développeurs Android à travers le monde.

Voici quelques fonctionnalités clés d'Android Studio :

- Éditeur de code : Android Studio inclut un éditeur de code intuitif qui vous permet de créer et de modifier facilement vos fichiers de code. Il prend en charge la coloration syntaxique et l'auto-complétion de code, ainsi que de nombreuses autres fonctionnalités utiles pour les développeurs.
- Debugging : Android Studio vous permet de déboguer votre code en temps réel grâce à son débogueur intégré. Vous pouvez ajouter des points d'arrêt, examiner les variables et suivre l'exécution de votre code pour trouver et corriger les erreurs.
- Prévisualisation : Android Studio vous permet de prévisualiser votre interface utilisateur en temps réel, en utilisant des "maquettes" de téléphones et de tablettes Android. Vous pouvez facilement ajuster votre interface utilisateur en modifiant votre code et en observant les changements en direct.
- Emulateur : Android Studio inclut un émulateur Android intégré qui vous permet de tester votre application sur différents appareils et versions d'Android sans avoir à utiliser de matériel réel.

## Technologie (Front/Back)

Le choix de ses technologies est fait par rapport à nos expériences passées lors du Master MIAE.

### Java native mobile

Le développement d'applications natives en Java est le processus de création d'applications mobiles pour des appareils Android en utilisant le langage de programmation Java. Les applications natives sont des applications qui sont installées directement sur l'appareil et qui s'exécutent en utilisant les ressources de l'appareil, telles que la mémoire, le processeur et le stockage.

Voici quelques avantages du développement d'applications natives en Java :

- Performance : Les applications natives en Java s'exécutent généralement plus rapidement que les applications hybrides ou Web, car elles utilisent directement les ressources de l'appareil.





- Fonctionnalités avancées : Les applications natives en Java ont accès à toutes les fonctionnalités de l'appareil, telles que la caméra, le GPS et les capteurs, ce qui les rend particulièrement utiles pour les applications de réalité augmentée ou de suivi de la santé.
- Interaction avec d'autres applications : Les applications natives en Java peuvent être facilement intégrées à d'autres applications sur l'appareil, ce qui les rend utiles pour les applications de partage de contenu ou de traitement de données.

## SQLite

SQLite est une base de données relationnelle légère qui peut être utilisée pour stocker des données dans des applications Android. Elle est très populaire en raison de sa simplicité d'utilisation et de sa petite taille, qui la rend idéale pour les applications mobiles à faible encombrement.

Voici quelques points clés à savoir sur l'utilisation de SQLite avec Android Studio :

- Utilisation de SQLite avec Android Studio : Pour utiliser SQLite avec Android Studio, vous devez d'abord inclure la bibliothèque SQLite dans votre projet. Cela peut être fait en ajoutant la dépendance suivante à votre fichier build.gradle.
- Création de tables : Vous pouvez créer des tables dans votre base de données SQLite en utilisant des instructions SQL CREATE TABLE. Par exemple, pour créer une table "utilisateur" avec les colonnes "id", "nom" et "âge", vous pouvez utiliser la commande suivante :

```
CREATE TABLE utilisateur (id INTEGER PRIMARY KEY, nom TEXT, age INTEGER);
```

- Insertion de données : Vous pouvez insérer des données dans votre base de données SQLite en utilisant l'instruction SQL INSERT. Par exemple, pour insérer un nouvel utilisateur dans la table "utilisateur", vous pouvez utiliser la commande suivante :

```
INSERT INTO utilisateur (nom, age) VALUES ('John', 30);
```

- Requêtes de sélection : Vous pouvez récupérer des données de votre base de données SQLite en utilisant l'instruction SQL SELECT. Par exemple, pour sélectionner tous les utilisateurs de la table "utilisateur", vous pouvez utiliser la commande suivante :

```
SELECT * FROM utilisateur;
```

Il existe de nombreuses autres instructions SQL que vous pouvez utiliser avec SQLite, comme UPDATE, DELETE et JOIN.

SQLite est un outil très utile pour stocker et gérer des données dans des applications Android.





## Application

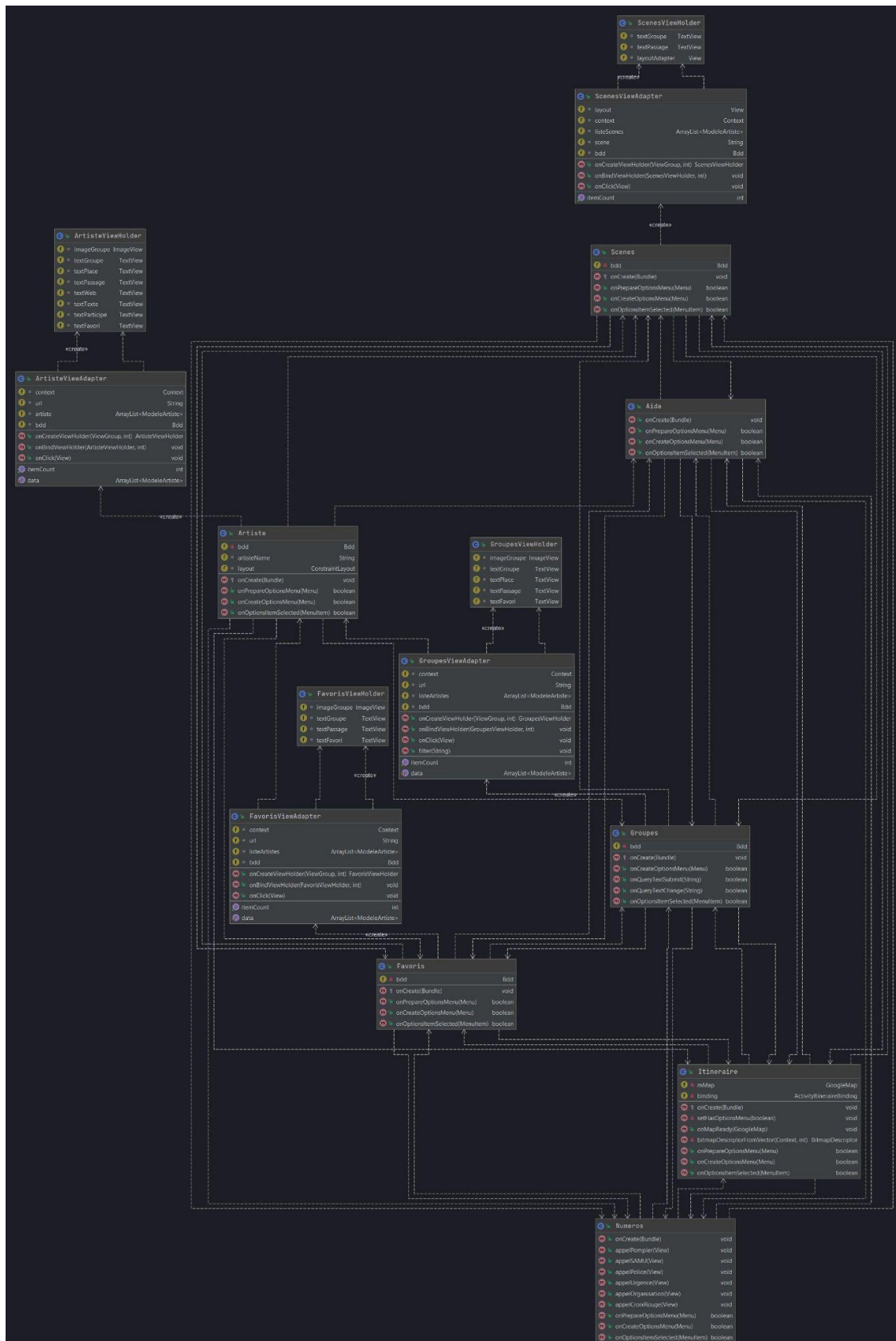
### Analyse du besoin

L'idée étant de pouvoir utiliser l'application avec l'accès à internet la première fois pour le chargement des données et ensuite la possibilité d'utiliser l'application et ses données sans un point d'accès internet.

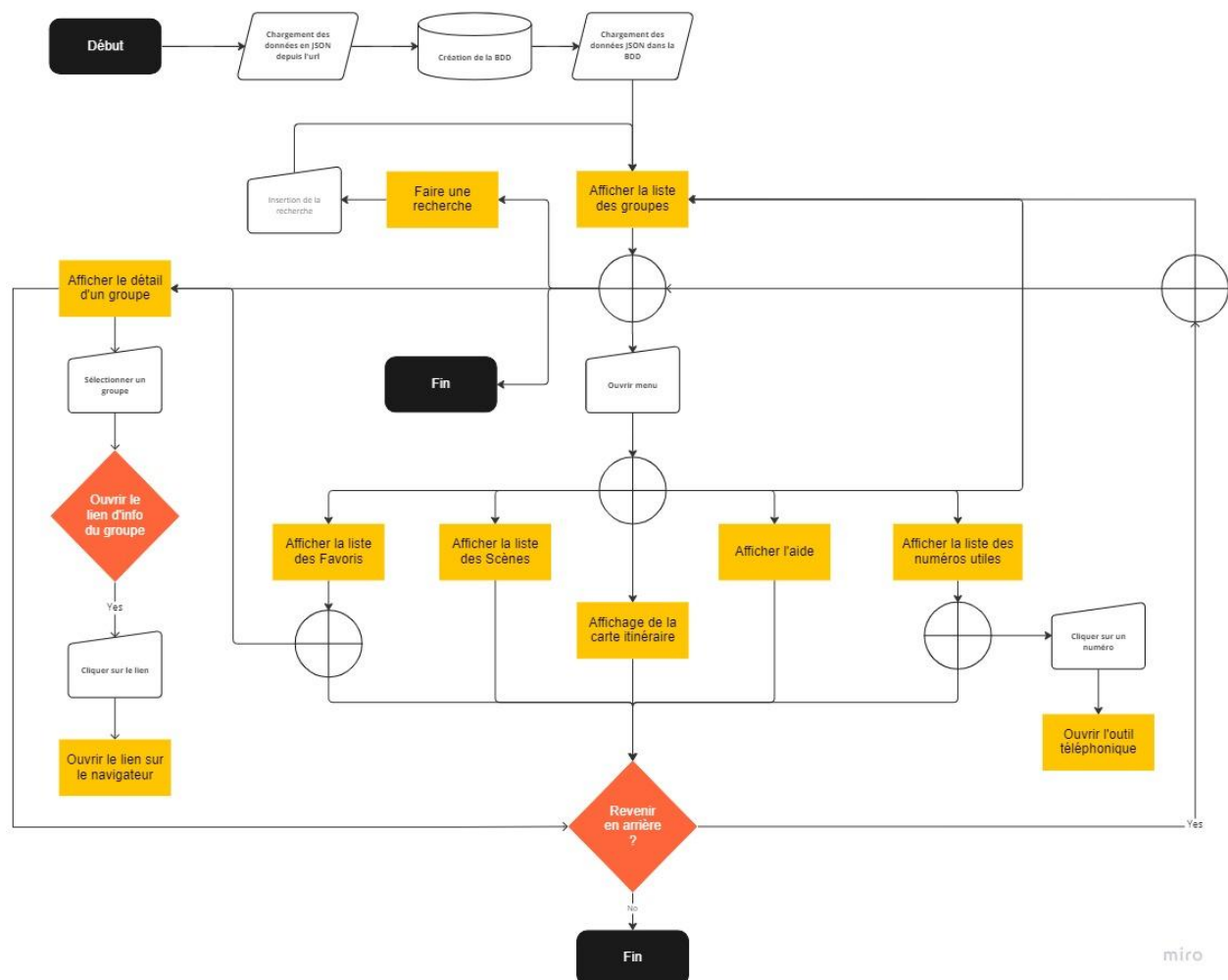
#### Besoin utilisateur

- Liste des groupes
  - Liste permettant d'afficher tous les groupes de musique.
- Liste des groupes favoris
  - L'application nous permet de mettre en favoris les groupes de musique que l'on aime et tout ça depuis la liste des groupes ou dans le détail du groupe
- Filtrer
  - Le filtre nous permet de filtrer en fonction du jour ou de la scène souhaitée. Nous sommes passés par une barre de recherche qui nous permet de faire une recherche full text qui ajoute aussi une possibilité d'auto-complétion.
- Notifications passages groupes favoris
  - Une notification est envoyée lorsque le groupe mis en favori passe sur scène.
- Notification
  - La notification se fait via le calendrier.
  - On ajoute ou l'on retire l'événement dans le calendrier.
  - On utilise le rappel du calendrier paramétré à 15 minutes.
- Géolocalisation sur le site (scènes, restauration, urgence...)
  - Pouvoir afficher une carte de site où se déroule le festival en question.
    - Une carte du site a été dessinée avec les points d'intérêt, toilettes, restaurations, croix rouge, scènes.
    - En cliquant sur un point on peut demander l'itinéraire (ouverture de google maps avec l'insertion du point d'arrivée en coordonnées.
- N° utiles
  - Une section numéro de téléphone utile sera mise à disposition, en effet on ne connaît pas forcément les numéros locaux des services utiles (Police, Samu, Pompier, etc...). On n'a qu'à cliquer sur le numéro cela permet d'ouvrir un service d'appel directement avec le service choisi.
- Aide
  - Une page permettant de mettre quelques informations sur les utilisations de l'application ou autres.

## Schémas

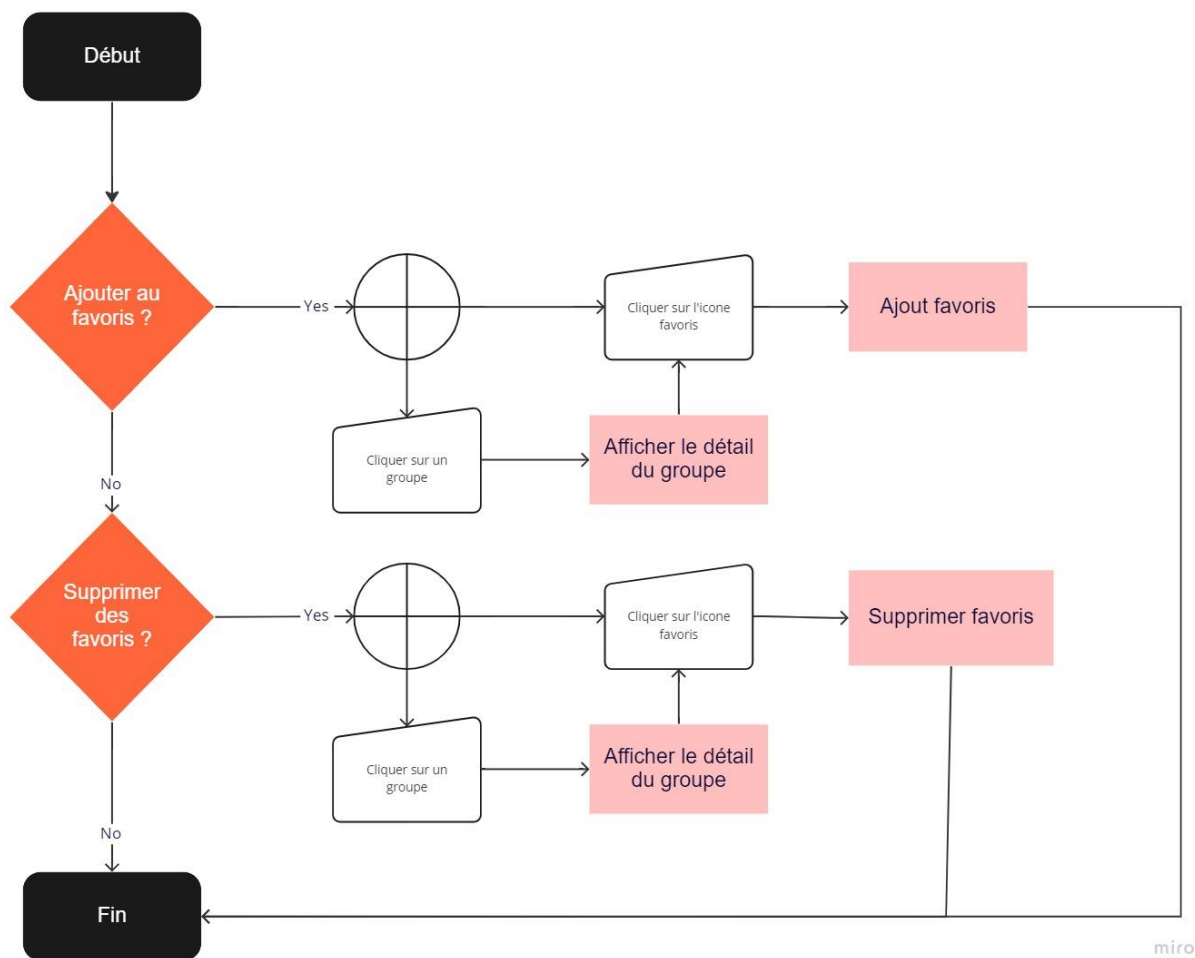


## Schémas des processus :



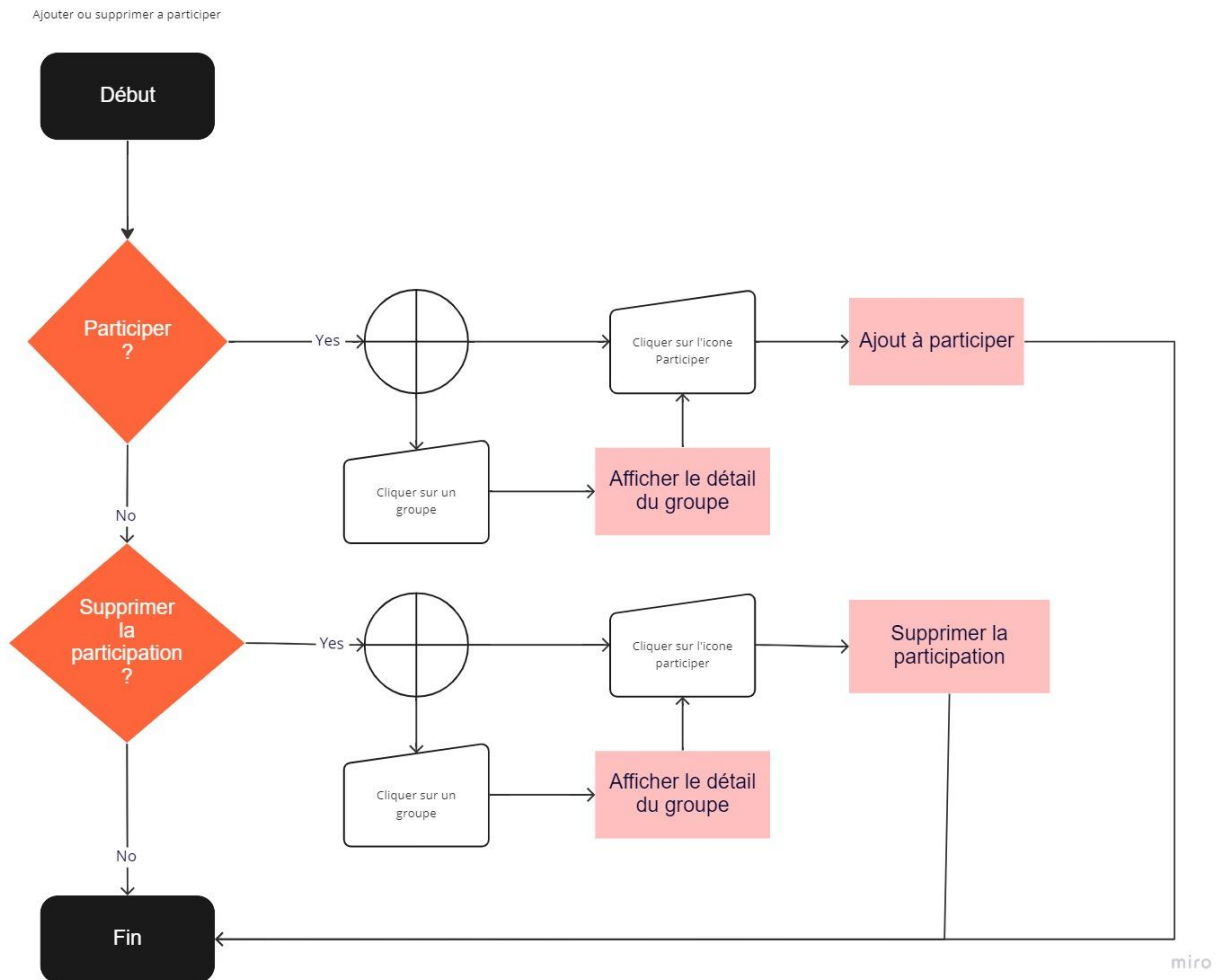
Représentation du séquençage des tâches et activités de l'application mobile Festival

Ajouter ou supprimer des favoris



Le processus d'ajout au favoris est utilisée lorsque l'on est dans l'action :

- Afficher les listes des groupes
- Afficher la liste des Favoris
- Afficher le détail d'un groupe



Le processus de supprimer la participation est utilisée lorsque l'on est dans l'action :

- Afficher les listes des groupes
- Afficher la liste des Scènes
- Afficher le détail d'un groupe

Le choix de l'utilisation de la base de données SQLite est de pouvoir utiliser l'application en mode hors ligne avec le choix de la structure de la base.

## Réalisation de la base de donnée

On crée 3 Tables dans la classe MaBaseSQLite:

- la table Artiste
  - Nous donnons les informations sur l'artiste et le détail sur son passage

```
private final static String CREATE_TABLE_ARTISTES = "create table artistes (" +
    "artiste varchar(50) primary key," +
    "texte BLOB default null," +
    "web varchar(250) default null," +
    "image varchar(250) default null," +
    "date timestamp default CURRENT_TIMESTAMP," +
    "jour varchar(10)," +
    "heure varchar(5)," +
    "duree int," +
    "place varchar(50) default 'Rock en Seine'," +
    "scene varchar(50)" +
    ");";
```

- la table Favoris
  - Nous donnons les favoris de l'utilisateur

```
private final static String CREATE_TABLE_FAVORIS = "create table favoris (" +
    "artiste varchar(50) primary key," +
    "favori int default 1" +
    ");";
```

- la table Participe
  - Nous donnons l'information de la participation de l'utilisateur

```
private final static String CREATE_TABLE_PARTICIPES = "create table participes (" +
    "artiste varchar(50) primary key," +
    "participe int default 1," +
    "eventId int" +
    ");";
```

Chaque table possède une clé primaire qui est "artiste" car il n'y a pas d'artiste en doublon actuellement, cela nous permet de relier les données provenant des 3 différentes tables présente. On utilise une vue pour exécuter des requêtes SQL, ce qui nous permet de naviguer et d'interroger la base de données.

```
private final static String CREATE_VIEW_INFORMATIONS = "create view informations AS " +
    "SELECT " +
    "a.*, b.favori, c.participe, c.eventId " +
    "FROM artistes a " +
    "LEFT JOIN favoris b ON a.artiste=b.artiste " +
    "LEFT JOIN partecipes c ON a.artiste=c.artiste;";
```

Dans la classe Bdd, on y retrouvera toute nos opérations CRUD (CREATE, READ, UPDATE, DELETE) comme par exemple:

- L'ouverture et la fermeture à l'accès de la Base de données :

```
public void open() {
    bdd = maBaseSQLite.getWritableDatabase();
}

public void close() {
    bdd.close();
}
```

Elles sont appelées pour initialiser la connexion à la base de données et pour fermer la connexion lorsque celle-ci n'est plus nécessaire.

- la récupération de la liste des artistes :

```
public ArrayList<ModeleArtiste> getListeScene(String scene) {
    //Récupère dans un Cursor les valeur correspondant à un livre contenu dans la BDD
    Cursor cursor = bdd.rawQuery("SELECT * FROM informations WHERE scene = ? " +
        "ORDER BY jour desc, heure;", new String[]{scene});
    return cursorArtiste(cursor);
}
```

Donné l'énoncé nous avons les url suivants :

- ["https://daviddurand.info/D228/festival/"](https://daviddurand.info/D228/festival/)
- ["https://daviddurand.info/D228/festival/liste"](https://daviddurand.info/D228/festival/liste)
- ["https://daviddurand.info/D228/festival/info/"](https://daviddurand.info/D228/festival/info/)

Avec la fonction "public String JSONObject (String url) throws MalformedURLException { ... }, on charge et convertit l'url donné en objet JSON. Elle ouvre une connexion URL et lit les données à partir de l'entrée, puis les convertit en chaîne de caractères et les retourne.

Ensuite on vient charger la liste des artistes et le détail des artistes présents dans le JSON dans la base de données sqlite que l'en a créé précédemment.



## Conclusion

Notre choix s'est basé sur l'utilisation du SQLite car il permet d'être utilisé pour stocker les données pour une application mobile, étant adapté pour des applications légères et peu volumineuses, il stocke des données structurées de manière relationnelle. Il peut donc stocker des informations sur les groupes de musique (nom, genre, date, etc.) ainsi que les informations sur le planning des passages (date, heure, lieu, etc.).

On peut utiliser des requêtes SQL pour rechercher les informations dans la base de données et afficher les résultats dans l'application mobile. Il ne nécessite pas de configuration ni d'administration. Il est facile à intégrer à une application mobile et peut fonctionner en local sur l'appareil de l'utilisateur, ce qui permet à l'application de fonctionner même lorsque l'utilisateur n'a pas accès à une connexion internet.

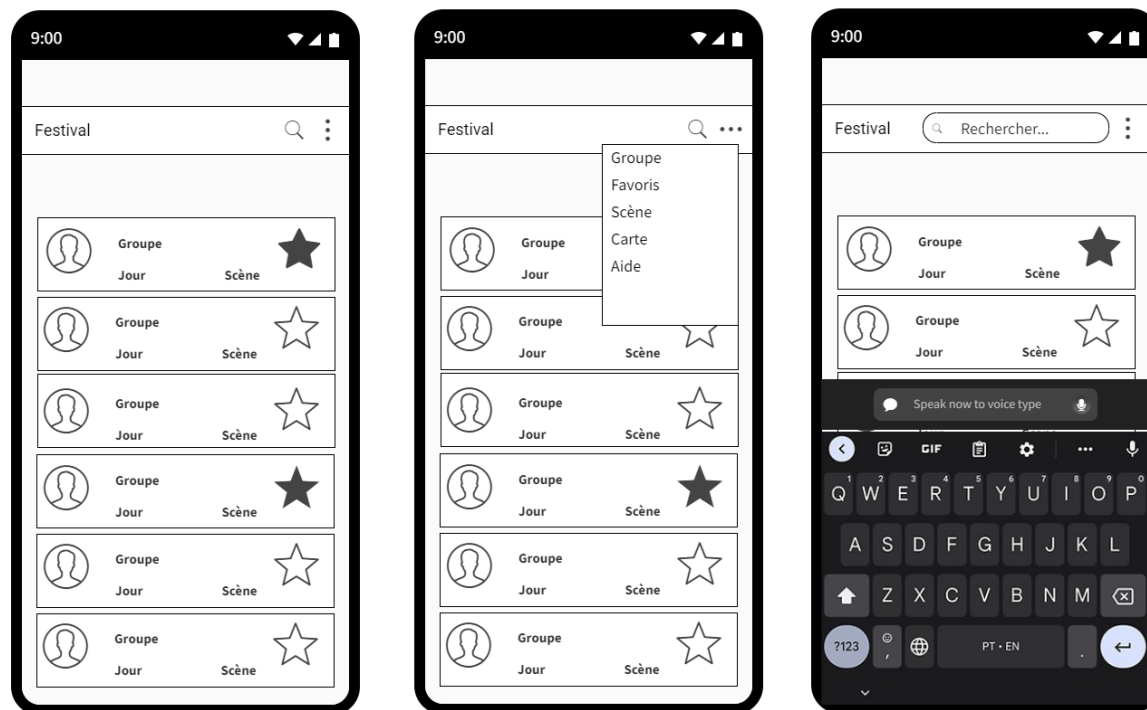
Cependant, si notre base de données devient plus grande ou plus complexe, il serait peut-être préférable de considérer d'autres options de gestion de base de données telles que Room vue en cours.

## Prototyper la solution (Maquettage)

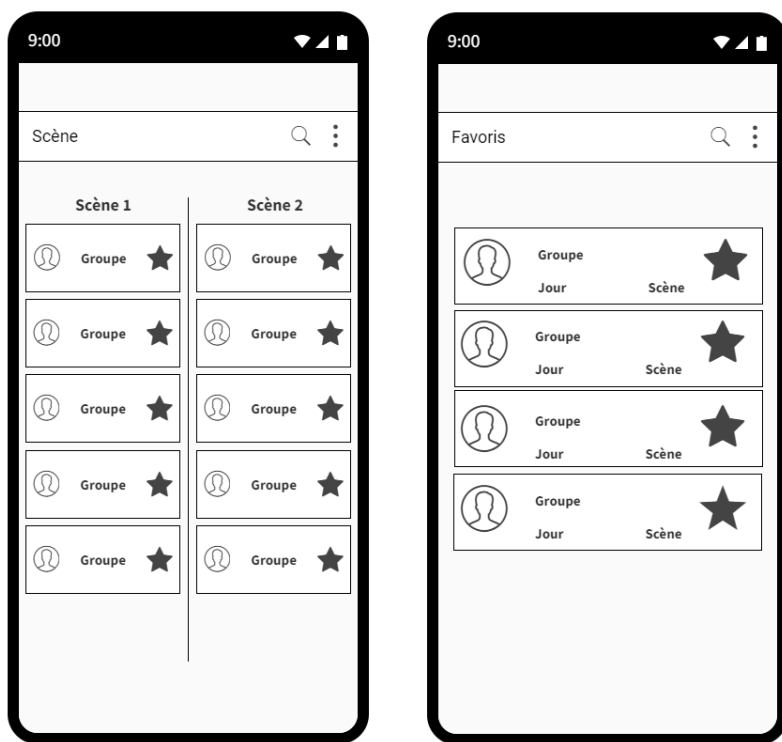
Page d'accueil



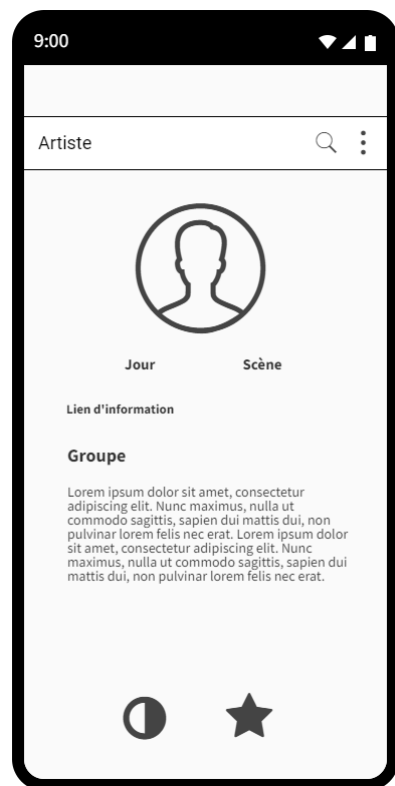
## Listes des groupes - menu - recherche



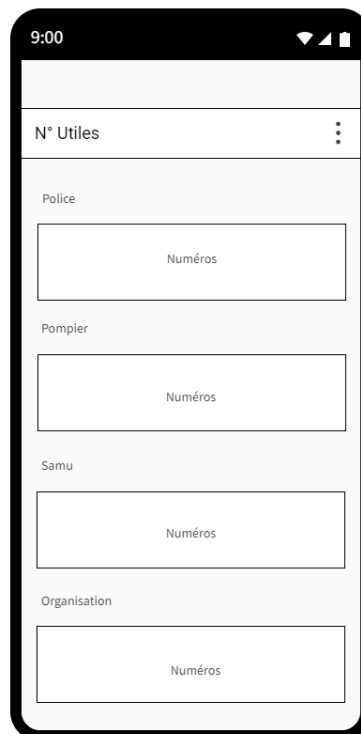
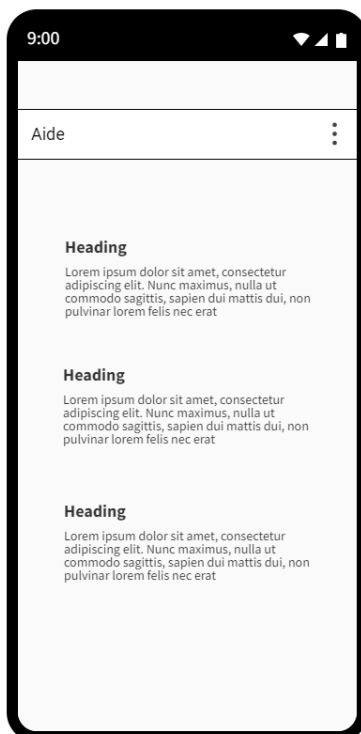
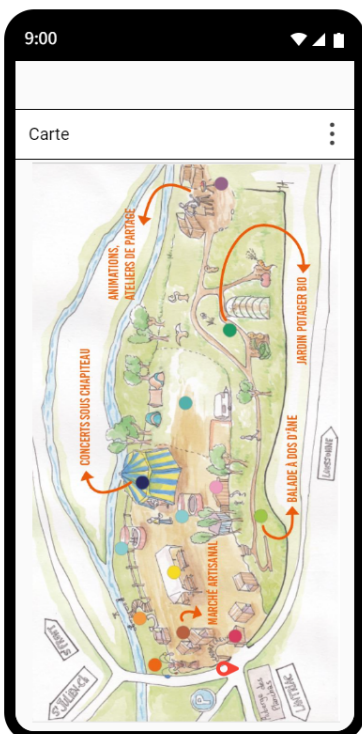
## Scènes - Listes des groupes favoris



## Détails groupe



Localisation - Aide - N° utiles



## Développement

Au niveau du développement, il y'a eu des modifications entre le résultat développer et les maquettes :

- Choix de la charte couleur
- Choix des émoticons
- Forme des boutons

Réalisation :

- Utilisation de github et de git pour avoir un historique et un versionning du projet.
- Utilisation du Trello nous servant de backlog projet, nous permettant de nous situer sur l'avancement du projet.
- Utilisation de la méthode de peer programming.
- Utilisation des cours de développement mobile vue pendant le cursus MIAGE.

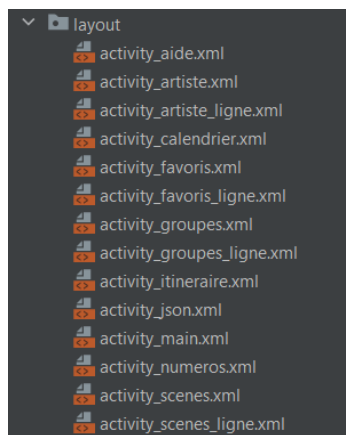
L'application Festival :

- Voir le dossier ou se situe le code.

## Description de l'application

### Layout

L'application est constituée des layout ci dessous:



Chaque layout représente une activité et l'application est organisée autour d'elles:

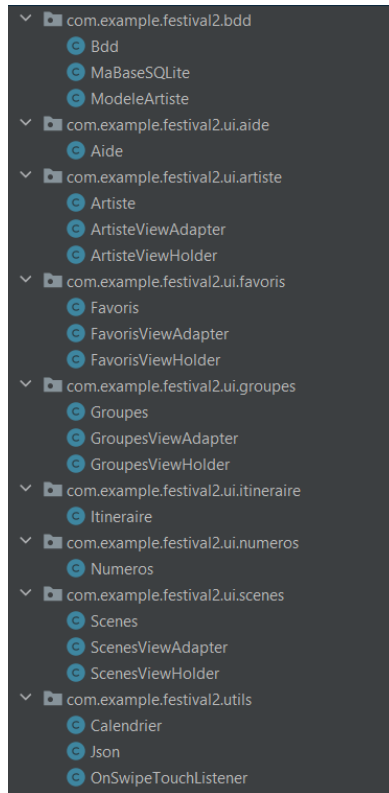
- activity\_aide.xml:
  - affiche des descriptions de l'application pour aider les utilisateurs.
- activity\_artiste.xml:
  - met en place la liste pour les artistes avec un RecyclerView.
- activity\_artiste\_ligne.xml:



- affiche le détail du groupe du groupe sélectionné, son image, le sélectionner comme favori, ajouter à la participation ou accéder à sa page internet.
- activity\_favoris.xml:
  - met en place la liste pour des favoris avec un RecyclerView.
- activity\_favoris\_ligne.xml:
  - affiche dans la liste des favoris sélectionner les informations des artistes (image, nom, jour, scène et icône favoris).
- activity\_groupe.xml:
  - met en place la liste pour les groupes avec un RecyclerView.
- activity\_groupe\_ligne.xml:
  - affiche dans la liste des groupes les informations des artistes (image, nom, jour, scène et icône favoris).
- activity\_numeros.xml:
  - affiche les numéros sous forme de bouton cliquable avec le label juste devant.
- activity\_scenes.xml:
  - met en place 2 listes en fonction des types de scène avec le RecyclerView.
- activity\_scenes\_ligne.xml:
  - affiche dans les listes les groupes correspondant à la bonne scène.

## Java

L'application est constituée des fichiers java ci-dessous:



L'ensemble du code a été organisé pour regrouper les fichiers par les fonctionnalités.

Chaque classe demandant l'utilisation de la bdd est accompagnée de 2 classes supplémentaires:

- Un viewAdapter, il représente et gère les données dans la vue, lie les données aux éléments de la vue et aide à les afficher.
- Un viewHolder, il contient les références aux éléments de la vue.

## Fonctionnement général

Lors de la première utilisation de l'application, celui-ci va vous demander l'autorisation d'accès à certaines informations pour l'utilisation de certaines fonctionnalités.

Lors de la première connexion à l'api via l'url, on récupère le contenu de l'url au format JSON la liste des groupes et le détails des groupes, on charge les données récupérées dans une base de données. Un spinner et un message s'affichent le temps que les données soient téléchargées.

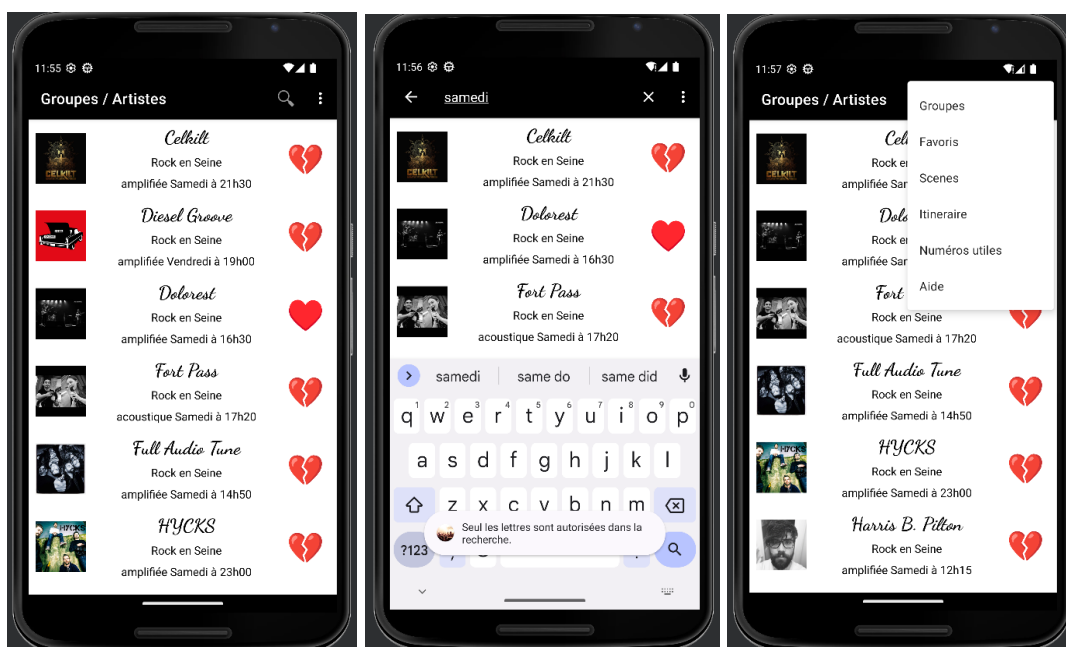
Après le chargement de la page, la liste des groupes s'affiche sur le premier écran, nous laissons le choix de plusieurs fonctionnalités:

- Un menu qui redirige vers diverses options (Aide, N°Utils, Carte, Liste Favoris, Liste Scène).
- Cliquer sur un groupe ouvre l'écran de détail du groupe avec les informations suivantes : Image du groupe, Nom, Description, Lien d'information, Jour, Scène et 2 boutons (Participe et Favoris).
- Cliquer sur le Bouton favoris présent sur la liste des groupes, ajoute ou supprime des favoris.
- Une barre de recherche permettant une recherche et rafraîchit la liste en question.

La mise en favoris ou la suppression de favoris d'un groupe est enregistrée en base et en parallèle une notification apparaît dans l'application pour notifier la mise en favoris ou la suppression des favoris.

### Manuel d'utilisation

- Écran d'accueil:
  - Cliquez sur un groupe pour afficher le détail correspond.
  - Cliquez sur le bouton favoris pour ajouter ou supprimer des favoris.
  - Cliquez sur l'icône de la loupe pour afficher la barre de recherche et filtrer en fonction.
  - Cliquez sur l'icône des 3 points vertical pour ouvrir le menu et découvrir les différentes options.

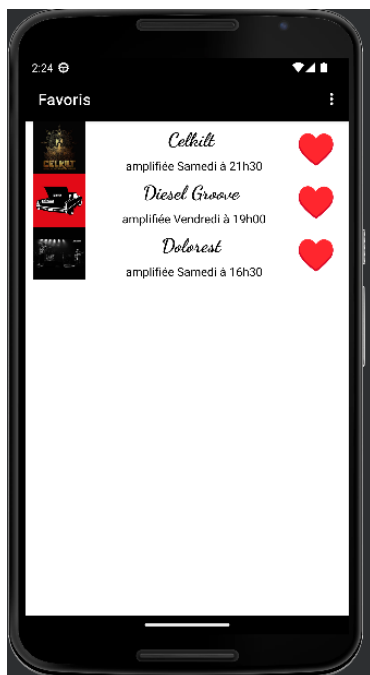




- Écran de détail
  - Double cliques maintenu et swip sur la gauche permet de revenir en arrière
  - Cliquez sur le bouton favoris pour ajouter ou supprimer des favoris.
  - Cliquez sur le bouton Participer pour ajouter ou supprimer des participations.
  - Cliquez sur le lien pour ouvrir un navigateur et visualiser la page du groupe.



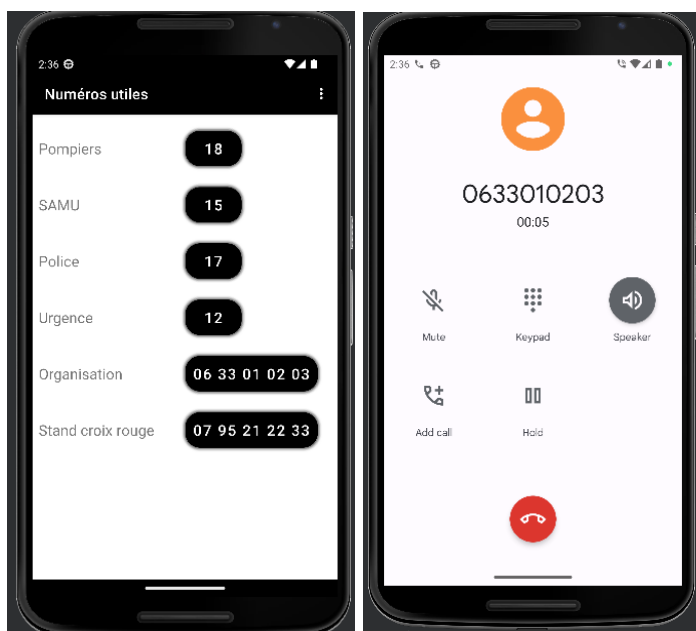
- Écran favoris:
  - Cliquez sur un groupe pour afficher le détail correspond.
  - Cliquez sur le bouton favoris pour ajouter ou supprimer des favoris.



- Écran Scène:
  - Cliquez sur le bouton Participer pour ajouter ou supprimer des participations.



- Écran N°Utils
  - Cliquez sur un des boutons avec le numéro ouvre un onglet appel



- Écran Aide





## Conclusion

### Résultats

Dans l'ensemble l'application et le design correspondent à nos attentes. Quelques bogues sont encore en diagnostic à l'heure actuelle malheureusement.

### Axes d'amélioration

Le livrable contient la version 1 de l'application, prochainement à faire:

- Debugger les bugs potentiel ou encore actifs
- Revoir l'UX/UI de l'application.