Manuel de développement de l'application Mobistory

Développé par Marius DONNÉ et Thomas VELU

VERSION 1.1

SOMMAIRE

Introduction

Objectif

```
Cette application a été développé dans le cadre d'un cours d'interface
graphique par M. Michel CHILOWICZ. Le but de cette application est
d'apprendre à des étudiants à l'ESIPE en deuxième année d'ingénierie en
informatique l'utilisation de Jetpack Compose sous Android.

Ce manuel vise à aider le développeur afin qu'il s'approprie le code de
l'application avec la plus grande aisance possible.
```

Présentation de l'application

```
L'application "Mobistory" permet à l'utilisateur d'en apprendre plus sur
l'histoire du monde grâce à des événements sélectionnés, ou à des quizz sur
ses connaissances en histoire.
L'application est développé sous Kotlin et Jetpack Compose avec l'IDE
Android Studio.

La philosophie de l'application est la suivante :
Une interface épurée et une navigation simple.

Épurée dans le sens où nous voulons que les informations, boutons soient
rapidement visibles pour l'utilisateur.
Simple dans le sens où la page principale est la liste d'événements, et la
navigation s'articule autour de cette page.
```

Architecture du code

```
L'application possède une architecture précise.
```

RACINE

MainActivity.kt

```
Le fichier MainActivity.kt permet de lancer l'application en chargant la
configuration de l'application, de la base de données et la composante
racine de l'application.
```

DATABASE

AppDataBase.kt

```
Le fichier représente une interface utilisé pour instancier un
RoomDataBase.
L'interface permet de définir trois fonctions :
    - eventDao, instancie les Data Access Objects représentant les "Event"
(représente les événements de notre application) ;
    - eventRelationDao, instancie les Data Access Objects représentant les
événements apparentés à d'autres événements ;
    - eventLocationDao, instancie les Data Access Objects représentant les
localisations géographiques des événements ;
```

DatabaseManager.kt

```
Le fichier DatabaseManager.kt permet de définir la base de données utilisée
pour notre application et de la gérer avec des mises à jours.
Il permet aussi de définir le dépôt de notre application afin qu'elle
puisse obtenir tous les événements et lire le JSON du dépôt.
```

ENTITY

```
Ce package contient tous les entités utilisés par la base de données,
passant par les représentations directes de chaque table (Event,
EventLocation, EventItem, EventRelation) au DAO de toutes ces tables.
```

JSON

```
Ce package contient le "parser" de JSON ainsi que sa représentation en tant
que objet.
```

LOCALSTORAGE

AppConfig.kt

```
Ce fichier permet de définir la configuration de l'application (langage,
liste d'événements favoris de l'utilisateur).
Le fichier permet aussi de modifier la liste "favoris".
```

AppData.kt

```
Ce fichier définit un objet qui possède un DatabaseManager et un AppConfig.
Cet objet est très utile surtout dans l'interface utilisateur.
```

UI

COMPONENTS

```
Ce package contient tous les éléments visuels mineurs (barre de recherche,
tiroir, filtre, etc.).
```

SCREENS

```
Ce package contient toutes les vues de notre application (page des
événements, favoris, quizz, paramètres).
```

THEME

```
Ce package permet de définir un style global de l'application.
```

Navigation.kt

```
Ce fichier définit les routes de chaque page.
Elle est essentielle dans certaines pages où une redirection est nécéssaire
```

```
(par exemple, SCREENS.AllEvents = page des événements où nous pouvons
rediriger les utilisateurs sur des événements spécifiques).
```

Scaffold.kt

```
Ce fichier représente la composante racine de notre projet.
Cette composante appelle la navigation comme contenu principal.
```

UTILS

Date.kt

```
C'est un fichier basique qui permet de convertir des strings représentant
une date obtenue à partir de la base de données.
Il y a deux conversions disponibles :
    - Une conversion en LocalDate, permettant de trier les listes
d'événements par date ;
    - Une conversion en string formatté par rapport au langage de
l'appareil (le format de la date n'est pas la même entre les français et
les anglais) ;
```

QuizQuestion.kt

```
Une classe représentant un événement ainsi que 4 choix de dates.
Pour connaître la vraie réponse, l'événement a un attribut date.
Cette classe est fortement utilisée dans l'écran Quiz.
```

WIKI

WikiAPI.kt

```
Ce fichier permet de faire appel à l'API de wikipédia afin d'obtenir des
images et des descriptions plus détaillés d'un événement.
```

Notes générales

```
Points importants :
    - Pas de string brut dans le code, nous essayons de maintenir une
traduction correcte entre l'anglais et le français ;
    - Il manque de la documentation dans le code !
```