

UNIVERSITÉ DE MONTPELLIER - FACULTÉ DES SCIENCES  
ANNÉE UNIVERSITAIRE 2021 - 2022  
MASTER INFORMATIQUE SPÉCIALITÉ GL  
HAI811I

Rapport final de projet :

Application Mobile pour Organisation  
d'évènements

Étudiants :

ALI DABACHIL

MERWANE RAKKAOUI



# Sommaire

<b>1</b>	<b>Introduction</b>	<b>2</b>
1.1	Contexte général du projet . . . . .	2
1.2	Objectif et cahier des charges . . . . .	2
<b>2</b>	<b>Technologies utilisées</b>	<b>3</b>
2.1	Langage utilisé . . . . .	3
2.2	Services utilisées . . . . .	3
<b>3</b>	<b>Développement de l'application</b>	<b>6</b>
3.1	Conception et implémentation des composants . . . . .	6
3.2	Design et Adapter . . . . .	9
3.3	Difficultés rencontrées . . . . .	9
3.4	Bilan et conclusion . . . . .	9

# Partie 1

## Introduction

### 1.1 Contexte général du projet

Nous avons tendance parfois à planifier des événements dans plusieurs cadres, on peut citer le sport, la culture, les séminaires et bien d'autres encore. Notre projet repose sur le développement et l'étude d'une application mobile permettant de répondre à ce besoin là. Planifier des événements en notifiant les personnes concernées, la possibilité de mettre à jour des événements, réserver un événement et éventuellement l'ajouter en tant que favori. L'utilisateur fait ses choix selon ses centres d'intérêts et bien d'autres encore.

Dans le cadre du module HAI811I, nous avons développé une application permettant l'organisation d'événements. Celle-ci offre un panel de fonctionnalités telles que la mise à jour d'événements, un ajout d'événements en tant que favori, la gestion des réservations et bien d'autres qui les accompagneront. Elles seront explicitées tout au long de ce rapport.

### 1.2 Objectif et cahier des charges

L'objectif de ce projet est de pouvoir offrir un ensemble de fonctionnalités selon le type de profil utilisateur qui va profiter de cette application. On peut en lister certaines ci-dessous :

- Ajouter un nouvel événement : saisir toutes les informations de cet événement : type, dates, lieu, tarif, choix d'images, description, catégories des places disponibles, nombre de places disponibles en fonction de leur catégorie, etc.
- Consulter les réservations pour un événement donné et ses statistiques.
- Consulter les événements suivant des critères : dates, lieu, type, etc.
- Réserver des places pour un événement donné.
- Ajouter l'événement à son agenda et le consulter.
- Modifier les informations d'un événement déjà programmé.

# Partie 2

## Technologies utilisées

### 2.1 Langage utilisé

Nous avons décidé de développer entièrement l'application en **Java** en exploitant un environnement de programmation adéquat qui est l'IDE Android Studio. Pour ce qui est du Front-End (interface utilisateur), le développement s'est effectué en Java tout en profitant des bibliothèques spécifiques à Android tel qu'Androidx et d'autres.

### 2.2 Services utilisées

Pour contribuer au développement de cette application, différents services ont été utilisés comme Firebase par exemple qui nous permet de gérer l'envoi des Notifications Push à un utilisateur. Plus précisément il s'agit d'un service cloud cross-plateforme de messagerie nommé Firebase Cloud Messaging.

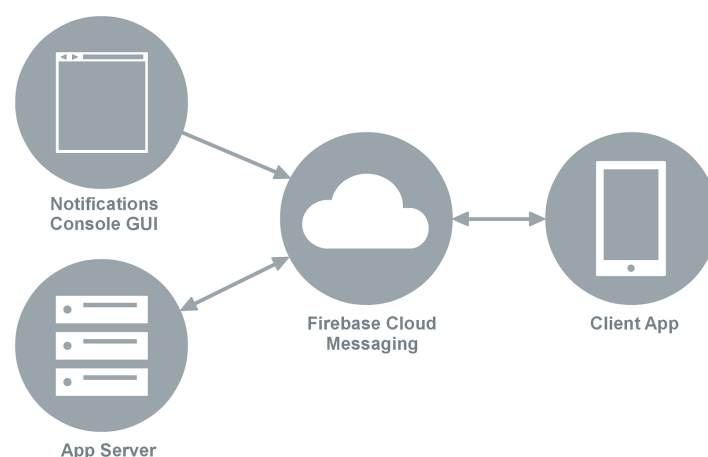


FIGURE 2.1 – Firebase Cloud Messaging

Un tel service a pu contribuer à ce qui concerne la partie back-end de notre application. Sa principale utilisation s'attarde seulement sur l'envoi de notifications Push aux utilisateurs participant à un événement.

Nous avons aussi utilisé un hébergeur cloud intitulé **000webhost**, c'est un hébergeur web qui vous offre la possibilité de créer votre domaine afin de l'utiliser comme une API REST. On a mis au point une API Rest par l'intermédiaire de ce service en ayant recours à des scripts PHP qui vont chacun d'entre eux répondre à des besoins comme la suppression d'un événement, un ajout de ce dernier ou encore la gestion de connexion utilisateur et plusieurs encore.



FIGURE 2.2 – 000webhost logo

Quasiment tout le back-end de l'application réside dans son utilisation, pour mettre en oeuvre toutes nos fonctionnalités, des requêtes HTTP type POST/GET sont envoyées à notre serveur qui lui va les traiter suivant l'URL donnée. L'URL donnée pour une fonctionnalité quelconque correspond au nom de notre domaine prédéfinie par l'hébergeur suivi du nom du script PHP qui répond au besoin de la fonctionnalité. Exemple : `https://eventplannerapp.000webhostapp.com/notify.php`

000webhost offre un panel d'outils, il vous donne la possibilité de changer de nom de domaine et d'administrer une base de données MySQL.

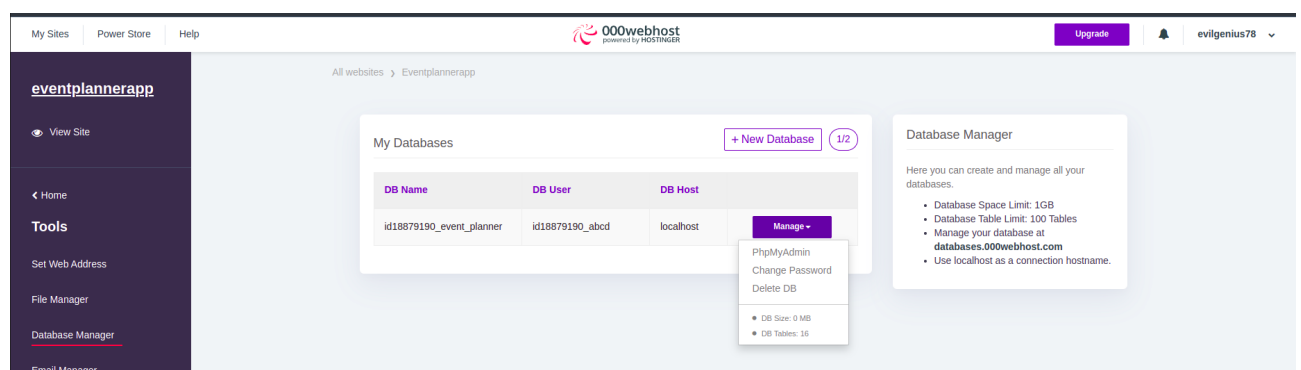


FIGURE 2.3 – 000webhost logo

Cela nous a permis de pouvoir gérer nous même notre base de données MYSQL, avec PHPMyAdmin.

Dans notre application, on a utilisé une base de données SQLite pour l’affichage des événements, des bookmarks et des réservations. Cela concerne l’architecture interne de notre application et ceci aura son importance dans la suite de ce rapport.

Table	Action	Lignes	Type	Interclassement	Taille	Perte
bookings	Parcourir Structure Rechercher Insérer Vider Supprimer	3	InnoDB	utf8_unicode_ci	48,0 kio	-
bookmarks	Parcourir Structure Rechercher Insérer Vider Supprimer	39	InnoDB	latin1_swedish_ci	32,0 kio	-
category	Parcourir Structure Rechercher Insérer Vider Supprimer	5	InnoDB	latin1_swedish_ci	32,0 kio	-
categoryPlace	Parcourir Structure Rechercher Insérer Vider Supprimer	4	InnoDB	utf8_unicode_ci	16,0 kio	-
events	Parcourir Structure Rechercher Insérer Vider Supprimer	70	InnoDB	latin1_swedish_ci	32,0 kio	-
events2	Parcourir Structure Rechercher Insérer Vider Supprimer	23	InnoDB	latin1_swedish_ci	32,0 kio	-
events3	Parcourir Structure Rechercher Insérer Vider Supprimer	28	InnoDB	latin1_swedish_ci	32,0 kio	-
fcm	Parcourir Structure Rechercher Insérer Vider Supprimer	3	InnoDB	latin1_swedish_ci	32,0 kio	-
image	Parcourir Structure Rechercher Insérer Vider Supprimer	23	InnoDB	utf8_unicode_ci	16,0 kio	-
user	Parcourir Structure Rechercher Insérer Vider Supprimer	23	InnoDB	latin1_swedish_ci	32,0 kio	-
UserComple	Parcourir Structure Rechercher Insérer Vider Supprimer	13	InnoDB	utf8_unicode_ci	16,0 kio	-
users	Parcourir Structure Rechercher Insérer Vider Supprimer	8	InnoDB	latin1_swedish_ci	32,0 kio	-
usertype	Parcourir Structure Rechercher Insérer Vider Supprimer	4	InnoDB	latin1_swedish_ci	32,0 kio	-
usertype2	Parcourir Structure Rechercher Insérer Vider Supprimer	2	InnoDB	utf8_unicode_ci	16,0 kio	-
Utilisateur	Parcourir Structure Rechercher Insérer Vider Supprimer	2	InnoDB	utf8_general_ci	16,0 kio	-
visible	Parcourir Structure Rechercher Insérer Vider Supprimer	3	InnoDB	latin1_swedish_ci	16,0 kio	-
<b>16 tables</b>	<b>Somme</b>	<b>237</b>	<b>InnoDB</b>	<b>utf8_unicode_ci</b>	<b>432,0 kio</b>	<b>0</b>

FIGURE 2.4 – Gestion DB PhpMyAdmin

Name	Size	Date	Permissions
.htaccess	0.2 kB	2022-05-03 21:23:00	-rw-r--r--
addEvent.php	4.0 kB	2022-05-10 18:19:00	-rw-r--r--
addEvent2.php	3.7 kB	2022-05-22 18:46:00	-rw-r--r--
addImage.php	0.1 kB	2022-05-09 15:21:00	-rw-r--r--
booking.php	0.1 kB	2022-05-10 23:15:00	-rw-r--r--
category.php	0.6 kB	2022-05-05 11:59:00	-rw-r--r--
categoryPlace.php	0.6 kB	2022-05-05 17:22:00	-rw-r--r--
categoryRegister.php	0.6 kB	2022-05-19 14:38:00	-rw-r--r--
changePassword.php	1.2 kB	2022-05-05 13:38:00	-rw-r--r--
changePassword2.php	1.2 kB	2022-05-06 17:57:00	-rw-r--r--
confirmBooking.php	1.9 kB	2022-05-23 23:40:00	-rw-r--r--

FIGURE 2.5 – Scripts PHP pour une API REST

# Partie 3

## Développement de l'application

### 3.1 Conception et implémentation des composants

Pour le développement de notre application, nous avons d'abord réfléchi aux façons de matérialiser ce qu'est un événement, un utilisateur, une réservation et un bookmark.

#### 1. **Utilisateur :**

Un utilisateur possède des informations nécessaires pour se connecter à l'appli ou s'inscrire à savoir un pseudo, un email et un mot de passe. Il a un type utilisateur (Provider / User ). Le Provider (Organisateur) se charge de planification d'événements, des mises à jour nécessaire et de la confirmation des réservations.

Un utilisateur de type User est simplement un client qui va consulter les événements créés par tous les organisateurs réunis, il peut faire des préférences sur des événements, modifier son mot de passe de connexion, consulter ses réservations en cours ou bien même encore de consulter des événements selon ses critères (Nom, dates, lieu, type).

#### 2. **Evenement :**

Le modèle Evenement constitue une entité importante dans cette application. Il présente un ensemble d'informations (nom, lieu, dates, heures, nombre de places, prix, etc ...). Un événement est créé que par l'organisateur et est publié pour qu'il puisse être réservé par les clients et/ou ajouté en favori. Un événement peut être filtré comme vu précédemment, l'utilisateur peut voir en temps réel tous les événements ajoutés.

#### 3. **Reservation :**

Un événement doit bien pouvoir être associé à un utilisateur. L'interaction entre ces deux entités réside lors de la réservation. Une réservation s'effectue à une date donnée par un client, elle peut être confirmée ou non par l'organisateur de l'événement en question. On ne peut pas réserver plus d'une seule fois, on a représenté un attribut qui indique si la réservation est active ou non et un attribut de confirmation indiquant la confirmation de celle-ci par l'organisateur.

#### 4. Bookmarks :

Une nouvelle interaction peut voir le jour, à savoir de l'ajout des événements en favoris. Le client peut fixer ses préférences en ajoutant un événement en favori et il peut même l'enlever si celui-ci ne l'intéresse plus du tout. En d'autres termes, ça lui laisse une possibilité de choisir suivant ses passions, centres d'intérêt etc...

On a surtout développé des fragments nécessaires à chaque fonctionnalité souhaitée. Des fragments qui s'occupent de l'affichage de la liste des événements , des réservations, des bookmarks et de l'authentification utilisateur.

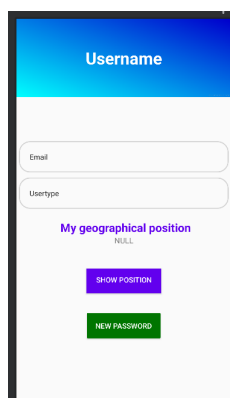


FIGURE 3.1 – Profil utilisateur

Ce fragment gère la gestion profil utilisateur, vous pouvez trouver votre position géographique et même modifier votre mot de passe.

On a d'ailleurs le fragment gérant l'ajout d'événement et il est similaire à celui qui s'occupe de sa mise à jour.

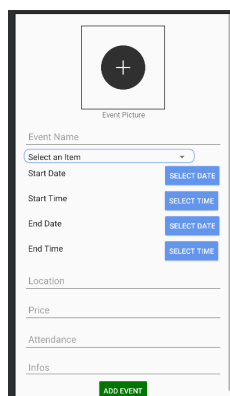


FIGURE 3.2 – Profil utilisateur



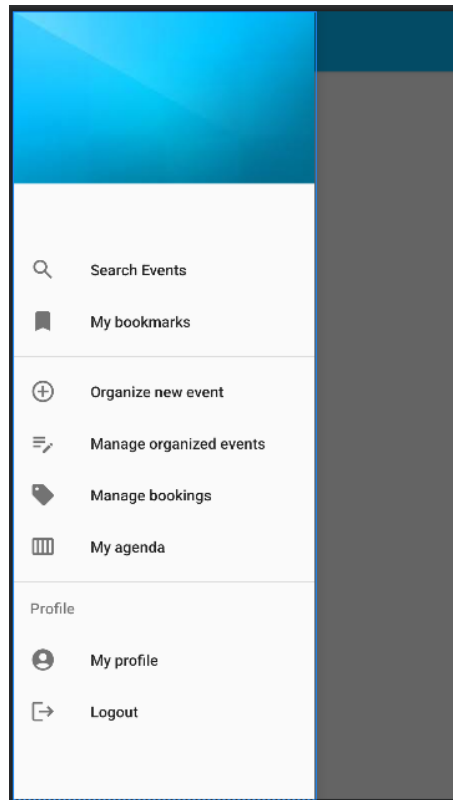


FIGURE 3.3 – Navbar Latérale

L'accès au panel de fonctionnalité se fait par une barre de navigation latérale, tous les items sont présents et différent suivant le type d'utilisateur authentifié. Pour simplifier un peu plus l'accès aux fonctionnalités une bottom navigation bar a été incorporée, la vidéo de démonstration de l'application nous permettra de voir plus clair à cela.

Voici-ci desus la navbar latérale pour un utilisateur de type **Provider**, d'une recherche d'événements, sa création jusqu'à l'extinction de sa session.

Un événement listé est matérialisé sous la forme d'une CardView d'un point de vue design. Une carte de couleur dégradé et des informations d'événement (dates, nom, localisation, une icône indiquant si la réservation est confirmée ou non).

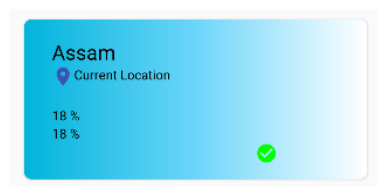


FIGURE 3.4 – CardView Event

## 3.2 Design et Adapter

On a profité de la librairie Material Design pour bien agencer les widgets comme les boutons/zones de saisie dans les fragments principaux de l'application.

On a créé des Adapter pour les entités Evenement et Reservation, afin de les lister correctement à partir des requêtes SQL. Une base de données SQLite a été mise en oeuvre pour pouvoir effectuer cette option d'affichage.

## 3.3 Difficultés rencontrées

Les difficultés se sont rencontrées notamment lors de la représentation des types de place pour un événement, on part du principe qu'une place n'est pas catégorisée ce qui paraît pas forcément réaliste si on prend en exemple un match de football, une place est attitrée selon la catégorie. Pour nous, pas de place attitrée et tarif unique pour cette même place.

On a pas mis au point de système de réduction pour une place à un événement donné, des contraintes de temps ont empêché ça. Les fonctionnalités principales sont tout de même présentes et diverses fonctionnalités auraient pu être peaufinées davantage. On a accordé de l'importance au design de l'application et à la simplicité pour offrir une bonne expérience utilisateur.

## 3.4 Bilan et conclusion

Ce projet nous a permis de consolider nos connaissances dans le développement mobile en ayant recours à différents services pour ce qui est du back-end. La diversité des services pour le côté serveur nous a posé pas mal de difficultés pour son appréhension et nous ont d'un autre côté forgé en connaissance pour des projets de développement futurs.