Projet Développement mobile : Planee

BERJOLA MATTHIAS 37000961 - ADOLPHE BENJAMIN 37001213

12 mai 2020

TABLE DES MATIÈRES

CHAPITRE

1

INTRODUCTION

Cette année, afin de valider l'UE développement mobile 2, nous avons dû réaliser un projet. Ce projet consistait à réaliser une application ou un jeu respectant certaines contraintes :

- Le choix de l'application est laissé libre, mais une partie de la notation portera sur sa complexité (quelque chose de trop simpliste ne donnera pas lieu à beaucoup de points).
- L'application doit proposer : plusieurs écrans, une présentation sous forme de liste, une sauvegarde persistante.
- L'application doit fonctionner correctement sur tout type d'écran (grand, petit...), en mode portrait et paysage : ressources alternatives sous Android + contraintes sur les composants graphiques.

De ce fait mon collègue et moi même avons opté pour une application. Après un BrainStorming intensif, nous avons décidé d'appeler notre application Planee et nous expliquerons son fonctionnement par la suite.

CHAPITRE

2

L'APPLICATION PLANEE

2.1 Principe

Planee est une application de gestion d'évènements. Elle permettra à l'utilisateur d'organiser et créer ses différents évènements. En effet, chaque évènement est composé d'un titre, d'une date limite de tâches et chaque tâche est composée d'un titre, d'un nom de magasin et éventuellement d'une URL indiquant le site du magasin si l'utilisateur a besoin de passer une commande. Sur le long terme l'application devrait supposer des magasins ou différents sites pour passer commande.

2.2 Architecture de l'application

L'application Planee est structurée en 3 Activités :

- L'activité MainActivity représente la page d'Accueil.
- L'activité AddActivity représente la page d'ajout d'un évènement.
- L'activité DetailActivity représente la page de détails d'un évènement.

2.2.1 MainActivity

Comme dit précédemment la MainActivity représente la page d'accueil. En effet, la MainActivity est composée principalement d'un élément ListView. La ListView permettra de lister les différents évènements. Les évènements sont récupérés dans la Base de Données locale du téléphone, si aucun évènement est présent dans la Base de Donnée alors un message apparaîtra à l'écran.

Quelques fonctionnalités de la page d'accueil :

- Appuie long sur un évènement pour le supprimer
- Appuie sur un évènement pour avoir les détails suivants :
  - Nom de l'évènement
  - Date de l'évènement

    - Liste des taches de l'évènement
- Appuie sur le bouton retour pour quitter l'application

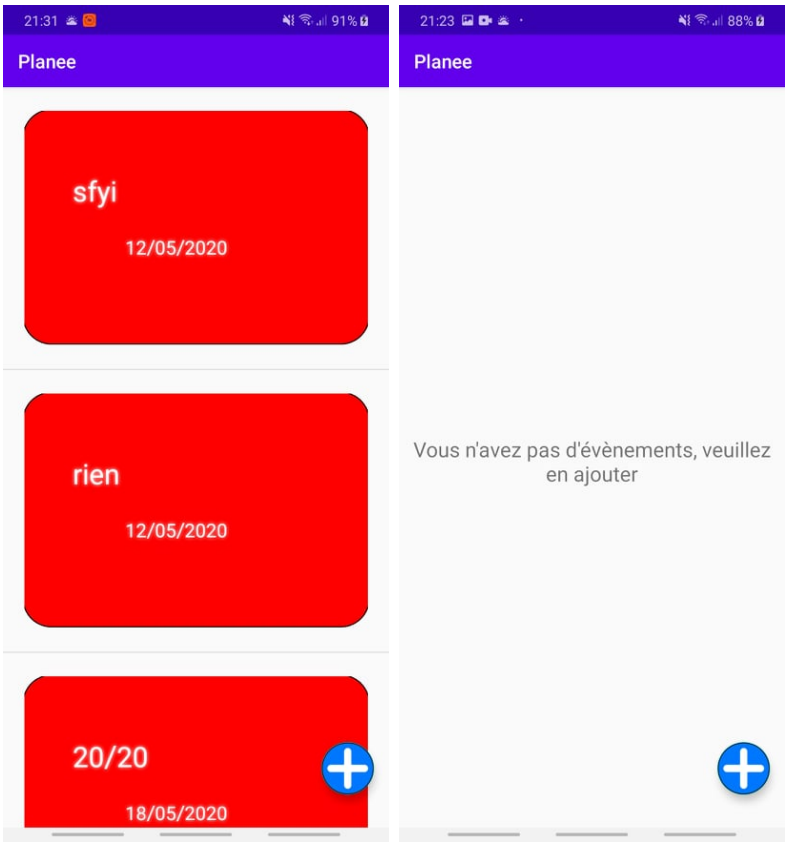


FIGURE 2.1 – Page d'accueil avec et sans évènements

2.2.2 AddActivity

La page contient un formulaire permettant d'entrer le nom, la date limite de l'évènement, l'heure et l'ensemble des tâches. Le formulaire d'ajout de tâches est dynamique grâce au bouton "Nouvelle tâche".

Quelques fonctionnalités du formulaire :

- Appuie sur le bouton "Nouvelle tâche" afin d'ajouter un champs de saisie d'une nouvelle tâches, ces champs sont composées de 3 champs de saisie de textes :
    - Nom de la tâche
    - Potentiellement le nom du magasin
    - Potentiellement l'URL du site du magasin
- Appuie sur le bouton "Ajout de l'évènement" afin d'ajouter l'évènement dans la base de données locale de l'appareil et déclencher une alarme à la date et l'heure entrées par l'utilisateur (Voir partie problèmes rencontrés)

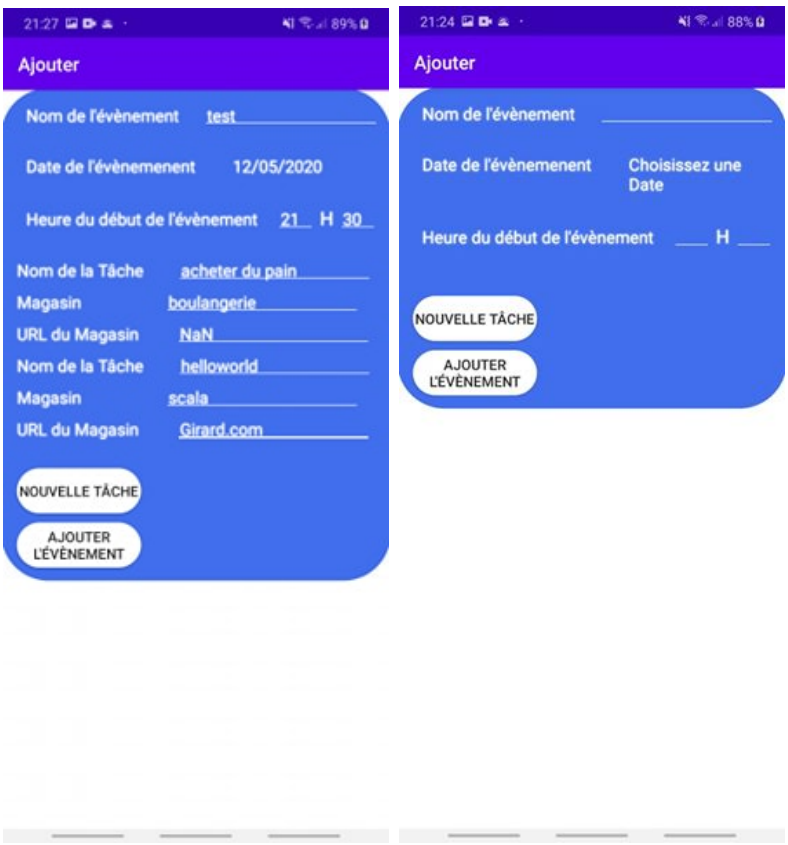


FIGURE 2.2 – Page du formulaire d'ajout avec et sans tâches

2.2.3 DetailsActivty

Comme dit dans la partie concernant la MainActivity, lors d'un appui sur un évènement on accède à la page de détails de cet évènement. Ainsi, on va récupérer tout les éléments dans la Base de données locale en rapport avec cet évènement.

2.2.4 Bases de données

Pour ce projet, nous avons utilisé une base de données composée de 2 tables :

- Une table Event_Table
- Une table Tache_Table

La table Event_Table sert à stocker les différents évènements créé. À l'intérieur on y trouve :

- l'id de l'évènement
- Le nom de l'évènement
- Sa date limite(La date où l'évènement se produira)
- L'heure limite (L'heure à laquelle l'évènement se déroulera)

La table Tache_Table sert quand à elle à stocker les différentes tâches à stocker les différentes tâches à effectuer afin de préparer l'évènement convenablement. Elle est composée de :

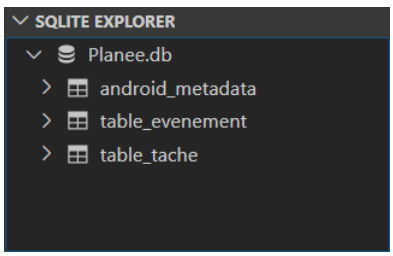
- L' id de la tâche
- Du nom du magasin dans lequel la tâche doit être effectué
- De l'URL du magasin si l'achat peut être effectuer en ligne
- IdEvent qui est une clé étrangère permettant de savoir à quel évènement appartient la tâche

On trouve également deux méthodes indispensables à la base de données :

- La méthode InsertEvent

- La méthode getAllEvent

La première méthode a été crée dans le but de pouvoir insérer chaque événements(composé de tâches) dans la base de données. La seconde méthode permet de récupérer tout les évènements crées par l'utilisateur afin de pouvoir les afficher de nouveau lorsque l'utilisateur reviens sur l'application après l'avoir fermé.
Le choix d'une base données a été fait dans le but de garantir la persistance des données de l'utilisateur à chaque fois que l'application est quittée puis ré-ouverte.



#	id	Nom	date_limite	heure
1	9	test	10/06/2020	15h59
2	10	test 2	12/08/2020	15H00
3	11	test 3	12/05/2023	15H00
4	12	test 4	12/07/2020	15H02

#	id	Nom	magasin	url_magasin	id_event
1	6	fifbb	rkdjk	didj	9
2	7	dbdidj	fifk	fifkk	10
3	8	fifjdj	rijfif	rifjfn	10
4	9	dhdu	didib	didi	11

FIGURE 2.3 – Schéma de la Base de données

2.3 Problèmes rencontrés

Au cours de ce projet, nous avons rencontré plusieurs problèmes. Nous allons donc citer dans ce rapports les différents problèmes que nous avons rencontré.

2.3.1 Fragments

Au début de ce projet, lors du choix de la première activité, nous avons choisi l'activité "Navigation Drawer Activity".

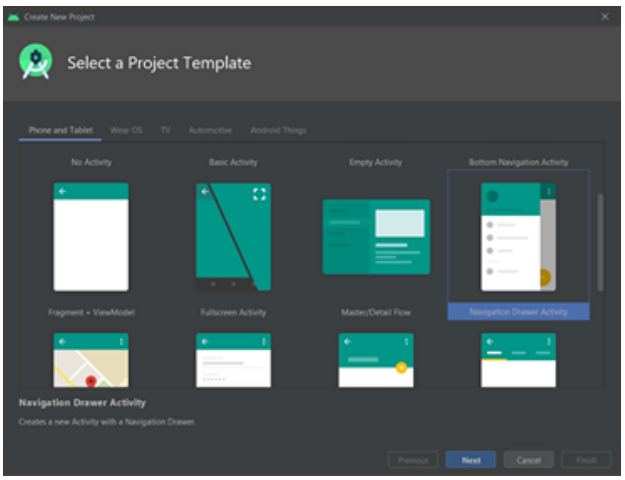


FIGURE 2.4 – Template

Cependant, ce template est basé sur l'utilisation des Fragments qui est une notion pas vue en cours.

2.3.2 Notifications

2.3.3 Layout

Lors de la réalisation du design de l'application nous devions géré en plus des éléments de bases(couleurs formes etc) les layouts. Deux problèmes revenais souvent : La conservation de l'échelle lorsque l'on passe à un appareil plus petit et la superposition des éléments lors de ce passage.
Au départ le design de l'application était réalisé entièrement avec des ConstraintLayout. Le problème avec les Constraint Layout est le manque de précision et de rigidité dans les contraintes que nous mettons. De ce fait des éléments qui semblait bien placés se retrouvait déplacés sans aucune intervention de notre part lorsque nous lançons l'application. Nous avons donc pris la décision de remplacer les Constraint Layout par des Relative Layout là où cela posait problème(page détail, page addDétail notamment). Après ce changement nous avons pu convenablement placé les différents éléments sans erreurs apparentes.

2.4 Amélioration futures

CHAPITRE

3

CONCLUSION

CHAPITRE

4

BIBLIOGRAPHIE