ISET SFAX AU 2023/2024 SEMESTRE I		
DEVOIR FINAL		
*Classes : DSI3**	*Matière : **D**éveloppement **M**obile*	*Nb pages : 8=4+4*
*Enseignants : **H**afedh **S**ouissi, **S**ami **H**adhri, **M**ondher **H**adiji*		
Documents Non Autorisés	*Durée : 1 heure 30 minutes*	

Problème

L'Association Tuniso-Palestinienne met à la disposition des étudiants palestiniens poursuivant leurs études en Tunisie une application appelée « **TOOFAN - طوفان** ». Cette application leur permet de signaler à l'association toute difficulté ou tout manque de contact avec leurs membres de famille, assurant ainsi un soutien en cas de besoin.

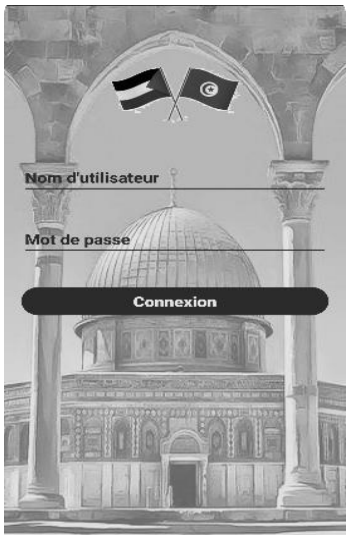
Cette application contient :

- 3 activités : « **MainActivity** », « **Accueil** » et « **AjoutMembre** »
- 1 activité « **MapsActivity »** de type « Google Maps Activity »
- 2 classes Membre et SQLiteFamille (Voir Annexe)

Une fois l'étudiant installe « **TOOFAN - طوفان** », il reçoit un email contenant ses paramètres de connexion (login et password).

L'activité « **MainActivity** » contient les fonctions décrites dans le tableau suivant :

MainActivity

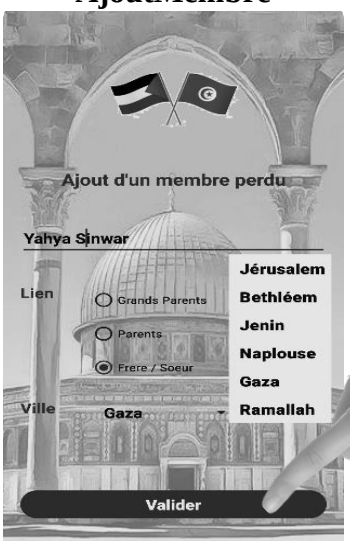


Fonction	Description
initialiser()	Initialise les 2 EditText, le bouton et l'instance **Firebase Firestore**
ecouteurs()	Lorsqu'on clique sur le bouton **btnConnecter** la fonction **connexion()** est exécutée
connexion()	Permet de tester l'existence des paramètres de connexion (email et password) dans la collection **users** de la base de données **Firebase Firestore**

Après authentification, l'interface de l'activité « **Accueil** » est chargée. Son interface est composée d'un **Button « btnAjouter »** et d'une **ListView** « **lvListe** ». Lorsqu'on clique sur le bouton « **btnAjouter**», l'activité « **AjoutMembre** » est lancée. Les interfaces des 2 activités (**Accueil** et **AjoutMembre**) sont les suivantes :

Accueil

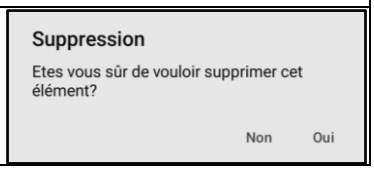
AjoutMembre



Accueil

Le tableau suivant décrit les fonctions de l'activité « **Accueil** » :

Fonction	Description
initialiser()	- Initialise le bouton **« btnAjouter »** et la **ListView** « **lvListe** » - Initialise le ArrayAdapter **adpter** et le lie à la **ListView** « **lvListe** » - Appelle écouteurs() et remplirListView()

ecouteurs()	- Lorsqu'on clique sur le bouton **btnAjouter** la fonction « **ajouter() »** est exécutée - Lorsqu'on clique sur un élément de la **ListView** « **lvListe** » la fonction « **afficherCarte(position) »** est exécutée - Lorsqu'on fait un long click sur un élément de la **ListView** « **lvListe** » la fonction « **supprimer(position) »** est exécutée
remplirListView()	Récupère le contenu de la table « **membre »** de la base SQLite « **famille »** et l'ajoute à la **ListView** « **lvListe** »
ajouter()	Permet de lancer l'activité « **AjoutMembre** » pour obtenir un résultat.
onActivityResult (ActivityResult result)	Si le **ResultCode** est **RESULT_OK** alors elle reçoit le **membre** saisi dans l'activité « **AjoutMembre »,** l'insère dans la table « **membre »** de la base SQLite « **famille »** et actualise le contenu de la **ListView** « **lvListe** »
afficherCarte(position)	Permet de lancer l'activité « **MapsActivity** » en envoyant le **nom de la ville** ainsi que le **nom de la personne**
supprimer(position)	Affiche la boite de dialogue suivante ; si on clique sur le bouton **oui** l'élément cliqué sera supprimé de la table « **membre »** de la base SQLite « **famille »** 

L'interface de l'activité **« AjoutMembre »** est formée d'un **EditText** « edNom », d'un **RadioGroup « rdgLien »**, de 3 **RadioButton (rdGP, rdP et rdFs)**, d'un Spinner « **spVille** » et d'un Button **« btnValider »**.

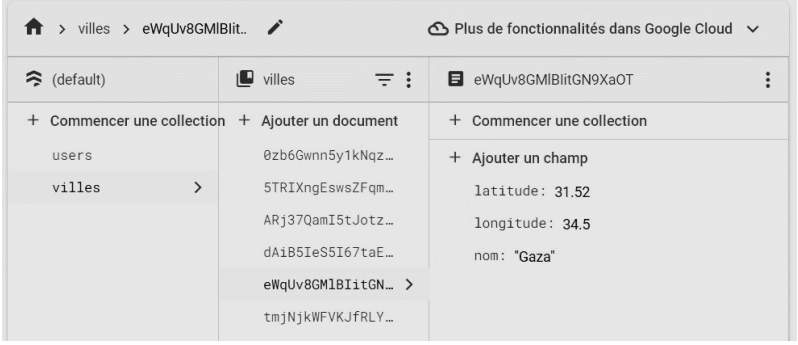
L'activité « **AjoutMembre** » contient les fonctions décrites dans le tableau suivant :

Fonction	Description
initialiser()	- Initialise les widgets et l'instance **Firebase Firestore**. - Initialise le **ArrayAdapter « adpter »** et le lie au **Spinner** « **spVille** » - Appelle ecouteurs() et remplirVille()
ecouteurs()	Lorsqu'on clique sur le bouton « **btnValider »** la fonction « **valider() »** est exécutée
remplirVille()	Récupère le contenu de la collection « **villes »** de la base **Firebase Firestore** et les ajoute au **Spinner** « **spVille** »
valider()	Récupère les données saisies, crée un objet **membre**, retourne un résultat contenant ce membre à l'activité « **Accueil »** et ferme l'activité

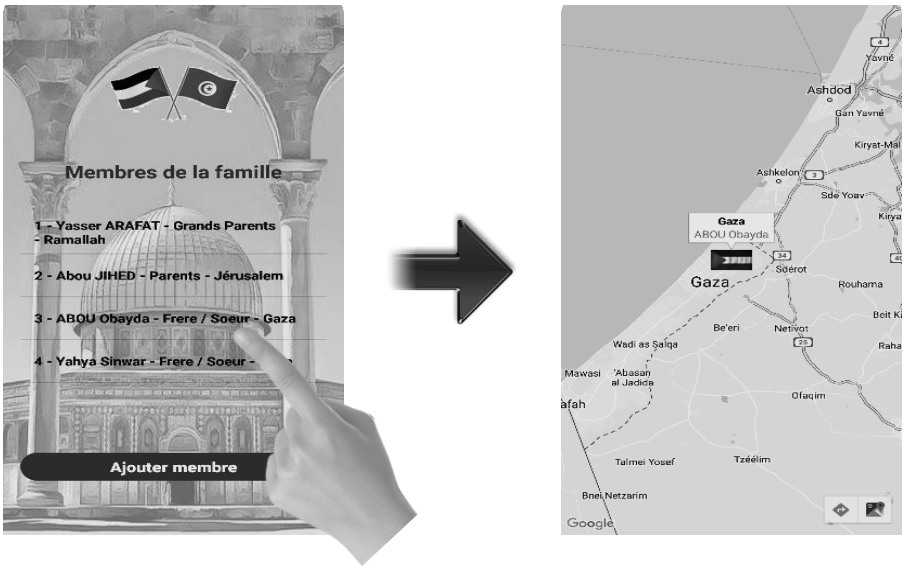
L'activité « **MapsActivity** » est lancée lorsqu'on fait un long click sur un élément de la **ListView** et elle contient les fonctions suivantes

Fonction	Description
recupererVille()	- Récupère le **nom de la ville** et le **nom de la personne** passés par l'activité «**Accueil** » - Récupère de la collection « **villes »** de la base de données de **Firebase Firestore** la « **longitude »** et la « **latitude »** de la ville - Appelle les méthodes « **marquerVille()** » et « **zoomVille()** »
marquerVille()	Ajouter un marqueur sur la ville (longitude, latitude) à la carte. Ce marqueur possède comme « **title »** la ville et comme « **snippet »** le nom de la personne
zoomVille()	Permet de zoomer sur la ville de la personne avec un niveau de zoom égal à 10

Voici un extrait de la base de données **Firebase Firestore** :



Les interfaces des 2 activités **Accueil** et **MapsActivity** sont les suivantes :



<u>Travail demandé</u> :

Compléter le code des activités « Accueil », « AjoutMembre » et « MapsActivity ».

ANNEXE

Classe Membre

```
package com.palestine;
import java.io.Serializable;
public class Membre implements Serializable {
    private int id;
    private String nom, lien, ville ;
    public Membre(String nom, String lien, String ville) {
        this.nom = nom;
        this.lien = lien;
        this.ville = ville; }
    public Membre(int id, String nom, String lien, String ville) {
        this.id = id;
        this.nom = nom;
        this.lien = lien;
        this.ville = ville; }
    public int getId() {
        return id;  }
    public String getNom() {
        return nom;  }
    public String getLien() {
        return lien;  }
    public String getVille() {
        return ville;  }
    @Override
    public String toString() {
        return id + " - " + nom + " - " + lien + " - " + ville;
    } }
```

Classe SQLiteFamille

```
package com.palestine;
//import...
public class SQLiteFamille extends SQLiteOpenHelper {
  public SQLiteFamille(Context context, String name, CursorFactory
factory,int version) {
        super(context, name, factory, version); }
    @Override
    public void onCreate(SQLiteDatabase db) {
```

```
  String Sql = "create table membre (id INTEGER PRIMARY KEYAUTOINCREMENT, nom
text NOT NULL,lien text NOT NULL,ville text NOT NULL);";
        db.execSQL(Sql);
    }
    @Override
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion)
{}  }
```

MainActivity.java

```
package com.palestine;
//import...
public class MainActivity extends AppCompatActivity {
    private FirebaseFirestore db;
    private EditText edLogin;
    private EditText edPassword;
    private Button btnConnecter;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        initialiser();}
    private void initialiser() {
        db = FirebaseFirestore.getInstance();
        edLogin = findViewById(R.id.edLogin);
        edPassword = findViewById(R.id.edPassword);
        btnConnecter = findViewById(R.id.btnConnecter);
        ecouteurs(); }
    private void ecouteurs() {
      btnConnecter.setOnClickListener(new View.OnClickListener()
      {  @Override
            public void onClick(View v) {
                connexion();
            } });  }
    private void connexion() {
        String email=edLogin.getText().toString();
        String password=edPassword.getText().toString();
        Query userQuery = db.collection("users")
                .whereEqualTo("email", email)
                .whereEqualTo("password", password);
        userQuery.get().addOnCompleteListener(
         new OnCompleteListener<QuerySnapshot>() {
            @Override
       public void onComplete(@NonNull Task<QuerySnapshot> task) {
        if (task.isSuccessful()) {
           QuerySnapshot querySnapshot = task.getResult();
           int size = querySnapshot.size();
           if (size > 0) {
           Intent i=new Intent(MainActivity.this, Accueil.class);
           startActivity(i);
           Toast.makeText(getApplicationContext(),"Connexion
réussie",Toast.LENGTH_LONG).show();
                    } else {
                      Toast.makeText(getApplicationContext(),"Utilisateur
inexistant",Toast.LENGTH_LONG).show();
                    }
                } else {
                    Toast.makeText(getApplicationContext(),"Erreur: " +
task.getException().getMessage(),Toast.LENGTH_LONG).show();
                } } }); }   }
```