

Examen de Développement Mobile Natif

Groupe :

Durée : 1h30

Etudiant :

Documents non autorisés

Numéro de compostage

Note :

/20

Accueil.java

```
//import...
public class Accueil extends AppCompatActivity {
    private ListView lvListe;
    private Button btnAjouter;
    private ArrayAdapter<Membre> adapter;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_accueil);
        initialiser(); }
    private void initialiser() {
        lvListe = findViewById(R.id.lvListe);
        btnAjouter = findViewById(R.id.btnAjouter);

}
private void ecouteurs() {
    btnAjouter.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) { ajouter(); } });
    lvListe.setOnItemClickListener(new AdapterView.OnItemClickListener() {
        @Override
        public void onItemClick(AdapterView<?> parent, View view, int position, long id) {
            afficherCarte(position); } });
    lvListe.setOnItemLongClickListener(new AdapterView.OnItemLongClickListener() {
        @Override
        public boolean onItemLongClick(AdapterView<?> parent, View view, int position, long id) {
            supprimer(position);
            return true; } }); }
private void ajouter() {

}
private void supprimer(int position) {
    AlertDialog.Builder alert = new AlertDialog.Builder(this);
    alert.....
    alert.....
    alert.....("Oui", new DialogInterface.OnClickListener() {
        @Override
        public void onClick(DialogInterface dialog, int which) {

}
} });
```

Ne rien écrire ici

```
alert. .... ("Non", new DialogInterface.OnClickListener() {
    @Override
    public void onClick(DialogInterface dialog, int which) { }    });
    alert.show(); }
ActivityResultLauncher<Intent> ActivityLauncher =
registerForActivityResult(
    new ActivityResultContracts.StartActivityForResult(),
    new ActivityResultCallback<ActivityResult>() {
        @Override
        public void onActivityResult(ActivityResult result) {
            if (result.getResultCode() == RESULT_OK) {

                } } });
private void remplirListView() {

}
private void afficherCarte(int position) {

} }
```

AjoutMembre.java

```
//import...
public class AjoutMembre extends AppCompatActivity {
    private FirebaseFirestore db;
    private EditText edNom;
    private RadioGroup rdgLien;
    private RadioButton rdGP, rdP, rdFS;
    private Spinner spVille;
    private Button btnValider;
    private ArrayAdapter<String> adapter;
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_ajout_membre);
        initialiser(); }
    private void initialiser() {
        edNom = findViewById(R.id.edNom);
        rdgLien = findViewById(R.id.rdgLien);
        rdGP = findViewById(R.id.rdGP);
        rdP = findViewById(R.id.rdP);
        rdFS = findViewById(R.id.rdFS);
        spVille = findViewById(R.id.spVille);
        btnValider = findViewById(R.id.btnValider);

}
    private void ecouteurs() {
        btnValider.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                valider(); } }); }
    private void valider() {
        String nom = edNom.getText().toString();
        if (!nom.isEmpty()) {

                setResult(RESULT_OK, i);
                finish(); } else
Toast.makeText(getApplicationContext(), "Entrer le nom", Toast.LENGTH_LONG).show(); }
    private void remplirVilles() {
        .....
        .get()
        .addOnCompleteListener(new OnCompleteListener<QuerySnapshot>() {
            @Override
```

```

        public void onComplete(@NonNull Task<QuerySnapshot> task) {
            if (task.isSuccessful()) {
                for (QueryDocumentSnapshot document : task.getResult()) {
                    .....
                    .....
                } } else {
                    Toast.makeText(getApplicationContext(), "Erreur lors de la
récupération de la liste des villes", Toast.LENGTH_SHORT).show(); }
            } }); } }

```

MapsActivity

```
//import...
public class MapsActivity extends FragmentActivity implements OnMapReadyCallback {
    private FirebaseFirestore db;
    private GoogleMap mMap;
    private String personne, ville;
    private double ville_LONG;
    private double ville_LAT;
    private ActivityMapsBinding binding;
    @Override
    protected void onCreate(Bundle savedInstanceState) {...}
    @Override
    public void onMapReady(GoogleMap googleMap) {
        mMap = googleMap;
        db = FirebaseFirestore.getInstance();
        recupererVille(); }
private void recupererVille() {

    .whereEqualTo("nom", ville);
    .get().addOnCompleteListener(new OnCompleteListener<QuerySnapshot>() {
        @Override
        public void onComplete(@NonNull Task<QuerySnapshot> task) {
            if (task.isSuccessful()) {
                for (QueryDocumentSnapshot document : task.getResult()) {

                } } else {
Toast.makeText(getApplicationContext(), "Erreur: " +
task.getException().getMessage(), Toast.LENGTH_LONG).show();
        } } });
    private void marquerVille() {
        MarkerOptions marker = new

marker.setIcon(BitmapDescriptorFactory.fromResource(R.drawable.flag));
        mMap.addMarker(marker); }
    private void zoomVille() {

} }
```