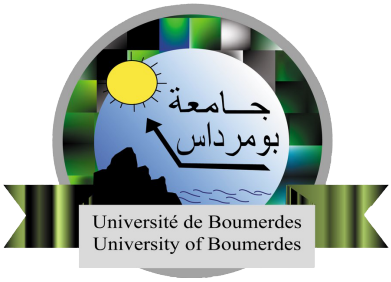
République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université M'hamed Bougara - Boumerdès



Faculté des Sciences
Département d'Informatique

Domaine : Mathématiques Informatique
Filière : Informatique
Spécialité : Systèmes informatiques
Module : Devloppement d'application mobile

Thème

Mini-Projet Apps-Mobiles
Realisation d'un guide touristique

Présenté par :
Mlle. BOURTACHE Ikram
Mlle. BOUKHEDOUNI Meriem
Mlle. BOUZERKOUNE Oumaima
Mlle. TAHMI Hadjer

Prof Responsable :
Mr R.Bitit

Table des matières

Table des figures **2**

1 Introduction **3**
1.1 Conception . . . 4
1.2 Architecte de la plateforme . . . 4
1.3 XML . . . 5
1.4 Contenu de la racine . . . 5
1.5 Utilisation des widgets de base . . . 6
1.5.1 Labels . . . 6
1.5.2 Boutons . . . 6
1.5.3 Champs de saisie . . . 6
1.5.4 Méthodes utiles . . . 7
1.6 Conclusion . . . 7

2 Conception **8**
2.1 Conception . . . 9
2.2 UML . . . 9
2.2.1 Les diagrammes . . . 9
2.2.2 Diagramme de cas d'utilisation . . . 10

3 Réalisation **11**
3.1 Android Studio . . . 12
3.2 Représentation de l'application . . . 12
3.3 Conclusion . . . 18

Table des figures

1.1 Architecte de la plateforme. 4
1.2 Contenu de la racine. 5
1.3 Partie du code XML. 6
1.4 Partie du code XML pour les methodes 7

2.1 Diagramme de cas d'utilistion general 10

3.1 Interface principale de l'application 13
3.2 le premier interface de l'application 13
3.3 l'interface de présentation des membres de l'équipe 14
3.4 l'interface "Discovery" 14
3.5 l'interface des hotels 15
3.6 l'interface des Restaurants 15
3.7 Appel un Resto 16
3.8 l'interface des lieux touristiques 16
3.9 l'interface "Notre wilaya" 17
3.10 Application en arabe 17

Chapitre 1

Introduction

1.1 Conception

Développement Android (en utilisant Java dans Android Studio) C'est l'une des trois méthodes pour créer des applications mobiles, appelée le développement natif. En réalité, cette approche est considérée comme la meilleure comparée aux applications Web et hybrides, car elle cible des modèles matériels spécifiques, ce qui rend les applications natives plus rapides. Notre objectif est de créer une application Android pour gérer les employés dans le cadre d'un petit projet, et pour y parvenir, nous devons aborder certains concepts fondamentaux tels que l'architecture de la plateforme, le modèle MVVM, SQLite, etc.

1.2 Architecte de la plateforme

L'architecte de la plateforme désigne la structure fondamentale sur laquelle repose une application Android. Elle définit comment les différentes couches de l'application interagissent entre elles, notamment l'interface utilisateur, la logique métier et les données. Comprendre cette architecture permet de développer des applications plus efficaces, modulaires et faciles à maintenir.

- **Applications :** Les applications, ou applications natives, sont des programmes de base qui permettent aux utilisateurs d'accéder à des fonctionnalités telles que l'appareil photo...
- **Cadre d'application (Application Framework) :** Tous les développeurs peuvent utiliser ces applications du cadre de développement pour créer de meilleures applications utilisateur, leur permettant de gérer les notifications, la géolocalisation, etc.
- **Bibliothèques :** Tout développeur peut accéder aux bibliothèques via le cadre Android. Un exemple est la bibliothèque SQLite, utilisée pour gérer la base de données locale.
- **Android Runtime (Dalvik VM – maintenant appelée Android Run Time) :** Elle est responsable de l'exécution des applications Android dans le système d'exploitation Android.
- **Noyau Linux (Linux Kernel) :** Il agit comme une couche d'abstraction entre le matériel et le logiciel ; il gère, par exemple, les capteurs.

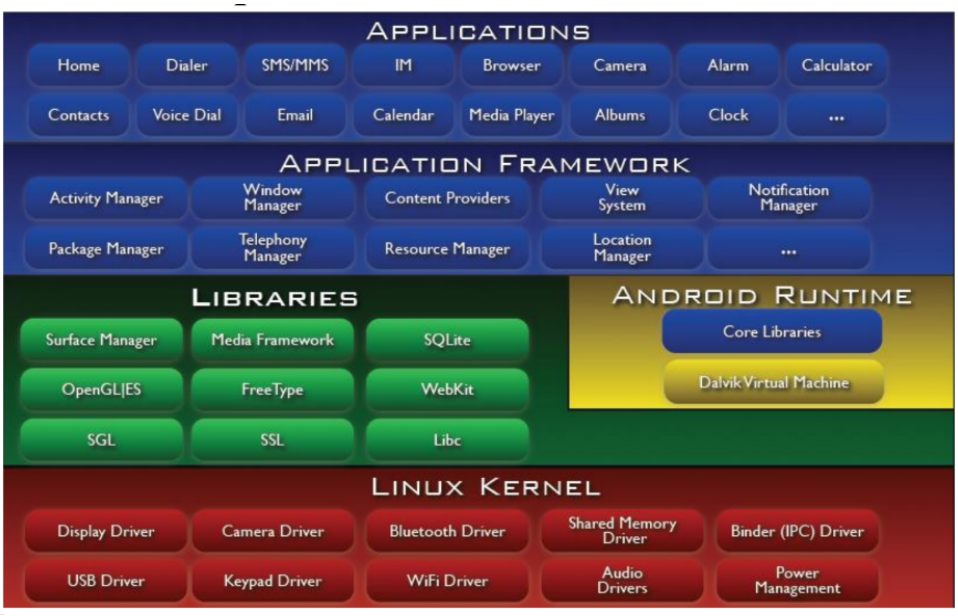


FIGURE 1.1 – Architecte de la plateforme.

1.3 XML

Nous allons utiliser le XML pour créer des interfaces d'applications, c'est pourquoi nous avons décidé de faire un petit résumé dans les quelques pages suivantes... pour nos cours de la matière DAM.

1.4 Contenu de la racine

plusieurs éléments peuvent construire le répertoire racine du projet :
AndroidManifest.xml est un fichier XML qui décrit l'application à construire et les composants activités, services, etc. fournis par celle-ci.
bin/ contient l'application compilée.
gen/ contient le code source produit par les outils de compilation d'Android.
libs/ contient les fichiers JAR extérieurs nécessaires à l'application.
src/ contient le code source Java de l'application.
res/ contient les ressources icônes, descriptions des éléments de l'interface graphique (layouts), etc. empaquetées avec le code Java compilé. La première fois que vous compilerez le projet, la chaîne de production d'Android créera le fichier R.java dans le paquetage de l'activité "principale". Ce fichier contient un certain nombre de définitions de constantes liées aux différentes ressources de l'arborescence **res**/.
l'arborescence res/ contient les ressources, c'est-à-dire des fichiers statiques fournis avec l'application, soit sous leur forme initiale soit, parfois, sous une forme prétraitée. Parmi les sous-répertoires de**res**/, citons :
res/**drawable**/ pour les images (PNG, JPEG, etc.) .
res/**layout**/ pour les descriptions XML de la composition de l'interface graphique.
res/**menu**/ pour les descriptions XML des menus.
res/**raw**/ pour les fichiers généraux (un fichier CSV contenant les informations d'un compte, par exemple).
res/**values**/ pour les messages, les dimensions, etc.
res/**xml**/ pour les autres fichiers XML généraux que vous souhaitez fournir.

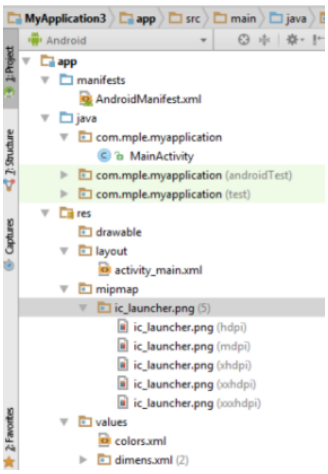


FIGURE 1.2 – Contenu de la racine.

1.5 Utilisation des widgets de base

1.5.1 Labels

Le label (TextView pour Android) est le widget le plus simple, En Java, un label est une instance de la classe TextView. Un élément TextView possède de nombreux autres attributs, notamment :
android :typeface pour définir le type de la police du label (monospace, par exemple).
android :textStyle pour indiquer si le texte doit être en gras (bold), en italique (italic) ou les deux (bold-italic).
android :textColor pour définir la couleur du texte du label, au format RGB hexadécimal .
Voici le contenu du fichier de positionnement du projet Basic/Label :

< ?xml version="1.0" encoding="utf-8" ?>
<TextView xmlns :android="http ://schemas.android.com/apk/res/android"
android :layout-width="fill-parent"
android :layout-height="wrap-content"
android :text="Vous vous attendiez à quelque chose de plus profond ?"
/>

1.5.2 Boutons

Button étant une sous-classe de TextView, tout ce qui a été dit dans la section précédente concernant le formatage du texte s'applique également au texte d'un bouton.

```
<LinearLayout
                android:layout_width="match_parent"
                android:layout_height="wrap_content" >

                <Button
                    android:id="@+id/button1"
                    android:layout_width="wrap_content"
                    android:layout_height="wrap_content"
                    android:textColor="#ffaaaaaa"
                    android:text="Button" />

                <Button
                    android:id="@+id/button2"
                    android:layout_width="wrap_content"
                    android:layout_height="wrap_content"

                    android:onClick="afficher" />

            </LinearLayout>
```

FIGURE 1.3 – Partie du code XML.

1.5.3 Champs de saisie

Outre les boutons et les labels, les champs de saisie forment le troisième pilier de la plupart des outils de développement graphiques. Avec Android, ils sont représentés par le widget **Edit-**

Text, qui est une sous-classe de **TextView**, déjà vue pour les labels.
En plus des propriétés standard de **TextView** (android :textStyle, par exemple), EditText possède de nombreuses autres propriétés dédiées à la construction des champs.

1.5.4 Méthodes utiles

findViewById() permet de retrouver un widget fils d'après son identifiant.
setContentView() on définit le View que contiendra (affichera) notre activité. Ici c'est le layout activite-principale (identifié par R.layout.activite-principale).

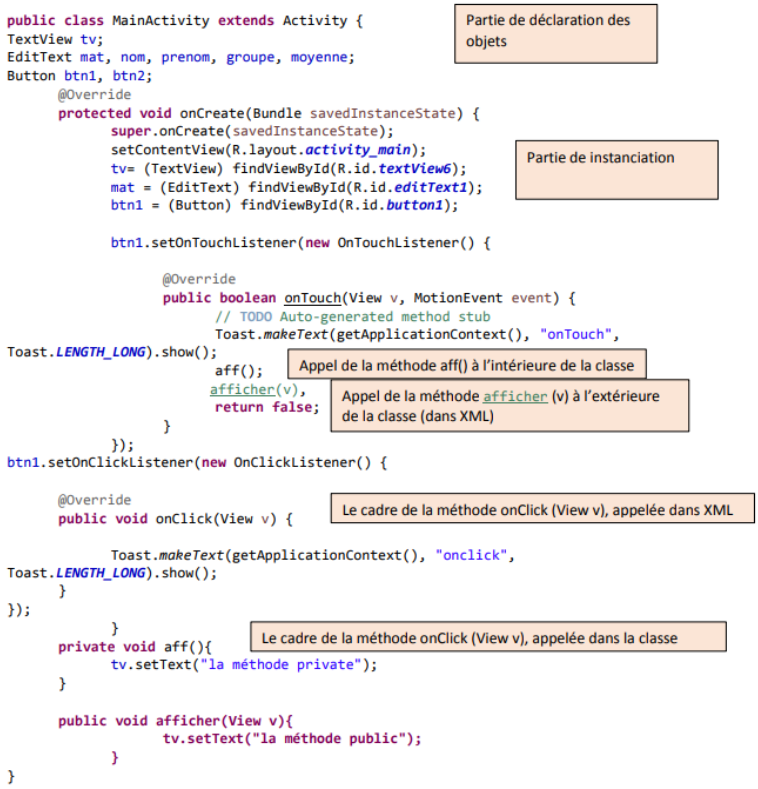


FIGURE 1.4 – Partie du code XML pour les methodes .

1.6 Conclusion

Dans ce chapitre, nous avons découvert l'importance de XML dans la conception des interfaces graphiques des applications Android, ainsi que la structure générale d'un projet Android. Cette organisation hiérarchique permet une séparation claire entre le code, les ressources et les composants essentiels de l'application.

Chapitre 2

Conception

2.1 Conception

Un projet bien conçu est la clé du succès dans l'étude des projets. Notre projet repose sur la création d'une application qui répond parfaitement aux besoins des utilisateurs. La phase de conception conceptuelle est cruciale, car elle détermine l'utilité du produit en traduisant des besoins réels en une solution informatique. Des diagrammes **UML** seront utilisés tout au long de cette phase de conception afin d'améliorer la modélisation du système.

2.2 UML

UML ou Unified Modeling Language est, comme son nom l'indique un langage de mod´elisation qui permet de d´ecrire les syst'emes d'informations du point de vue conceptuel
et physique. L'OMG ou Objet Management Group d´efinit UML comme un langage graphique de visualisation, sp´ecification, construction et documentation d'un syst'eme

2.2.1 Les diagrammes

UML comporte treize types de diagrammes représentant autant de vues distinctes pour représenter des concepts particuliers du système d'information. Ils se répartissent en deux grands groupes :

Diagrammes structurels ou diagrammes statiques (UML Structure)

- diagramme de classes (Class diagram)
- diagramme d'objets (Object diagram)
- diagramme de composants (Component diagram)
- diagramme de déploiement (Deployment diagram)
- diagramme de paquetages (Package diagram)
- diagramme de structures composites (Composite structure diagram)

Diagrammes comportementaux ou diagrammes dynamiques (UML Behavior)

- diagramme de cas d'utilisation (Use case diagram)
- diagramme d'activités (Activity diagram)
- diagramme d'états-transitions (State machine diagram)

Diagrammes d'interaction (Interaction diagram)

- diagramme de séquence (Sequence diagram)
- diagramme de communication (Communication diagram)

- diagramme global d'interaction (Interaction overview diagram)
- diagramme de temps (Timing diagram)

Ces diagrammes, d'une utilité variable selon les cas, ne sont pas nécessairement tous produits à l'occasion d'une modélisation. Les plus utiles pour la maîtrise d'ouvrage sont les diagrammes d'activités, de cas d'utilisation, de classes, d'objets, de séquence et d'états-transitions. Les diagrammes de composants, de déploiement et de communication sont surtout utiles pour la maîtrise d'œuvre à qui ils permettent de formaliser les contraintes de la réalisation et la solution technique.

2.2.2 Diagramme de cas d'utilisation

Le diagramme de cas d'utilisation représente la structure des grandes fonctionnalités nécessaires aux utilisateurs du système. C'est le premier diagramme du modèle UML, celui où s'assure la relation entre l'utilisateur et les objets que le système met en œuvre.

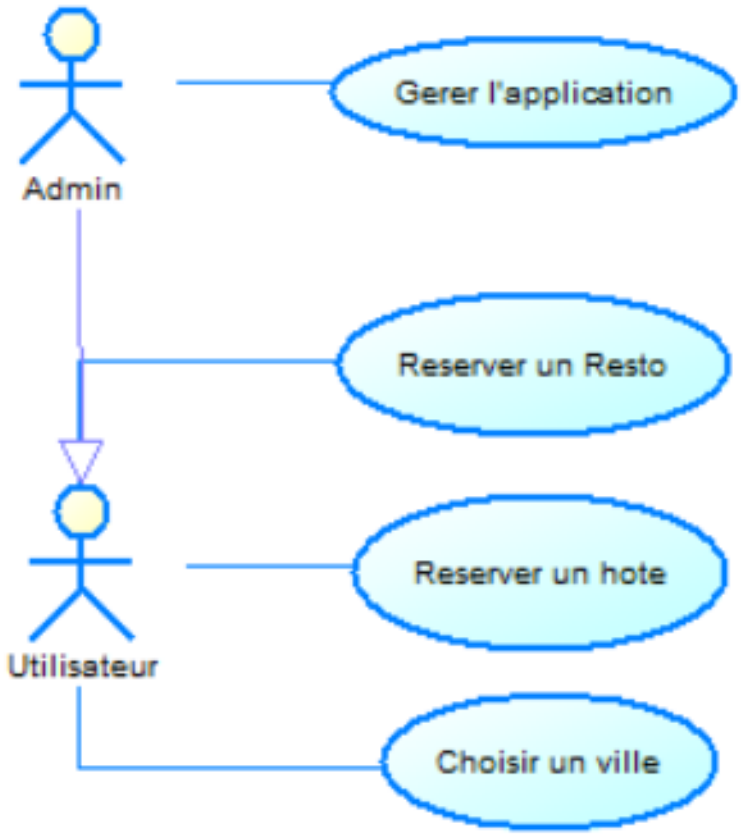


FIGURE 2.1 – Diagramme de cas d'utilistion general

Chapitre 3

Réalisation

La phase de réalisation marque la transition de la conception à la mise en œuvre concrète de notre application. Elle consiste à traduire les modèles établis en code fonctionnel, en respectant les besoins définis lors de l'analyse. Durant cette étape, nous avons utilisé Android Studio pour développer l'interface utilisateur en XML et implémenter la logique de l'application en Java.

3.1 Android Studio

Android Studio est l'environnement de développement intégré (IDE) officiel pour la création d'applications Android. Développé par Google, il est basé sur IntelliJ IDEA et fournit tous les outils nécessaires pour concevoir, coder, tester et déboguer des applications mobiles. Il prend en charge les langages Java, Kotlin et XML, et offre des fonctionnalités puissantes comme un éditeur de code intelligent, un émulateur Android, un concepteur d'interface graphique (layout editor), ainsi qu'un système de gestion de projet basé sur Gradle.
Un AVD (Android Virtual Device) est utilisé pour imiter le comportement physique d'un matériel (téléphone) à l'aide d'un logiciel. Cela permet de tester notre application sur plusieurs plateformes matérielles sans avoir à acheter les téléphones correspondants.

Le SDK Android ne contient aucun appareil virtuel préconfiguré, il est donc nécessaire d'en créer au moins un. Android vise une très large gamme d'appareils, et pour mieux gérer cette diversité, Google a développé un système capable d'émuler une grande variété de terminaux.

Un AVD est en réalité un terminal virtuel possédant plusieurs caractéristiques propres, telles que la taille de l'écran, la version du système, la présence ou non d'une carte SD, etc. Lorsque l'émulateur se lance, il utilise un AVD et s'adapte aux différentes propriétés de ce dernier.

Lorsqu'on commence à développer sur Android, il peut être nécessaire de créer plusieurs AVDs, surtout pour les développeurs qui ne disposent pas d'un terminal Android physique.

3.2 Représentation de l'application

Dans le cadre de notre mini-projet en développement d'applications mobiles, nous avons réalisé une application Android servant de guide touristique pour une wilaya du sud algérien. Cette application a pour objectif principal de faciliter l'organisation des séjours touristiques dans cette région. Elle permet à l'utilisateur de consulter les différentes villes, choisir les sites à visiter, ainsi que réserver des hôtels ou des restaurants selon ses préférences. L'interface est conçue pour être simple, intuitive et adaptée à tous les profils d'utilisateurs, qu'ils soient touristes locaux ou étrangers. Ce projet met en pratique nos connaissances en conception mobile, en intégrant à la fois la gestion des ressources (layouts XML), la navigation entre les activités, et les fonctionnalités essentielles pour offrir une expérience utilisateur fluide et pratique.

Lors du lancement de l'application, l'utilisateur est accueilli par une interface d'accueil moderne et conviviale.

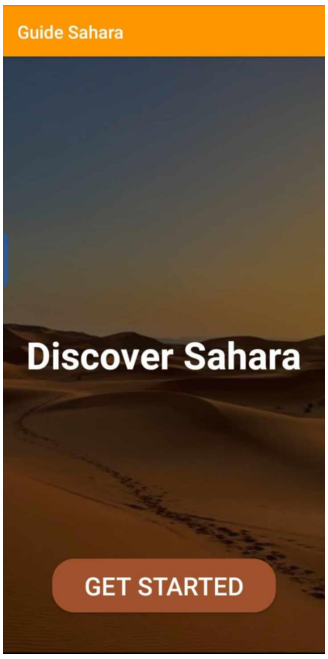


FIGURE 3.1 – Interface principale de l'application

Une fois que l'utilisateur clique sur le bouton "Get Started", il est automatiquement redirigé vers l'interface principale de l'application

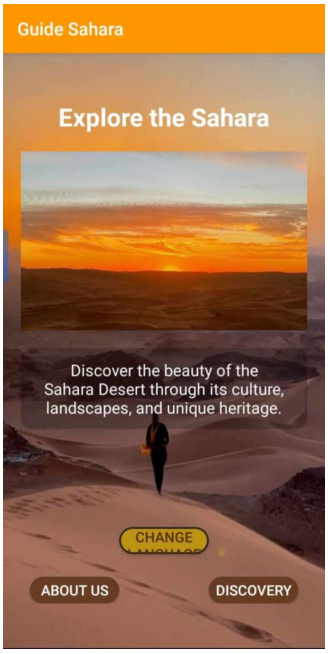


FIGURE 3.2 – le premier interface de l'application

Lorsque l'utilisateur clique sur le bouton "About Us", il est redirigé vers une interface dédiée à la présentation des membres de l'équipe ayant participé à la réalisation de l'application.

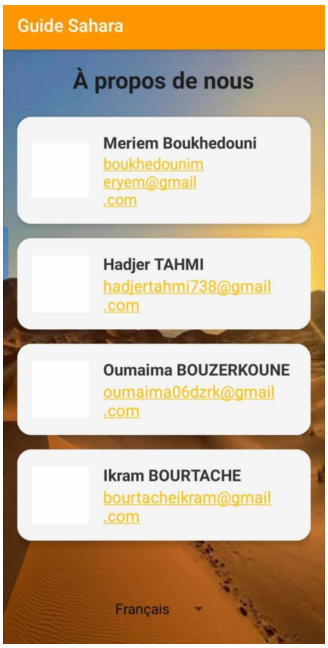


FIGURE 3.3 – l'interface de présentation des membres de l'équipe

Lorsque l'utilisateur clique sur le bouton "Discovery", il est dirigé vers une interface qui lui permet de commencer l'exploration des fonctionnalités principales de l'application.

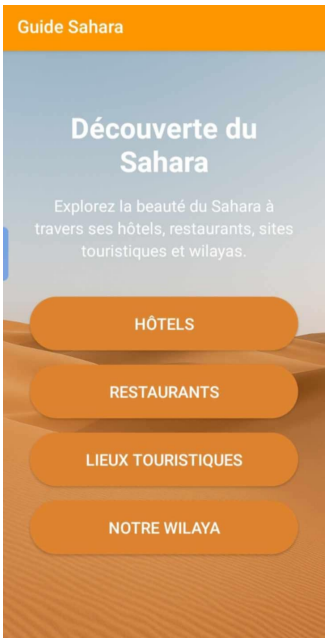


FIGURE 3.4 – l'interface "Discovery"

Lorsque l'utilisateur clique sur le bouton "Hotel", il accède à une interface dédiée à la recherche et à la réservation d'hôtels disponibles dans la wilaya de Sahara.

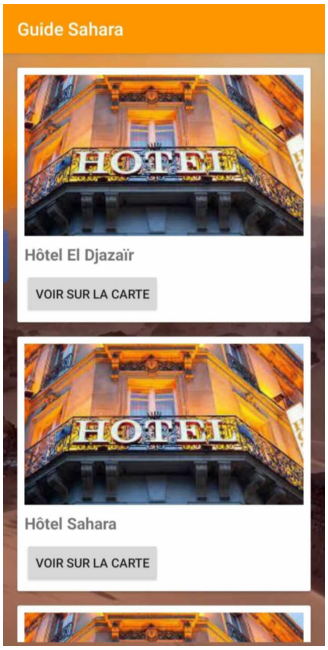


FIGURE 3.5 – l'interface des hotels

Lorsque l'utilisateur clique sur le bouton "Restaurants", une liste de restaurants s'affiche.

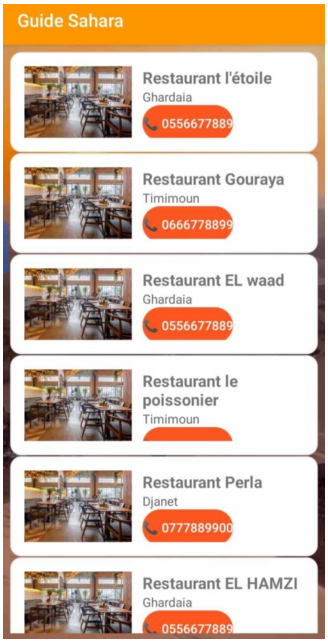


FIGURE 3.6 – l'interface des Restaurants

L'utilisateur peut choisir d'appeler le restaurant directement via le numéro de téléphone ou annuler l'action.

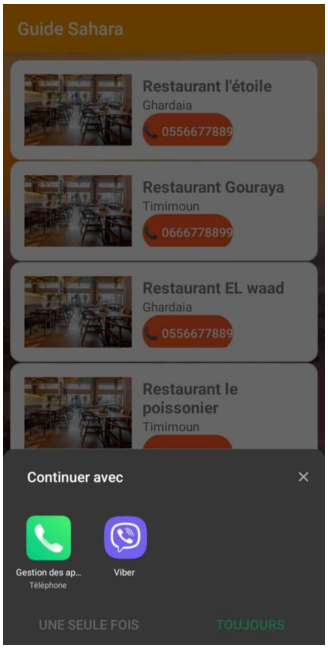


FIGURE 3.7 – Appel un Resto

Clique sur le bouton "lieux touristiques" , l'application charge une liste des lieux touristiques disponibles dans la région.

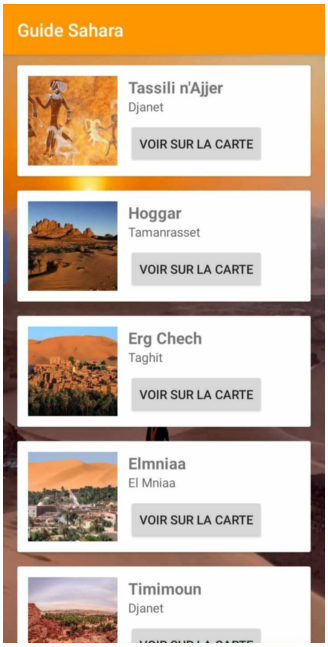


FIGURE 3.8 – l'interface des lieux touristiques

Lorsque l'utilisateur clique sur le bouton "Notre wilaya", l'application affiche les différentes villes ou lieux touristiques de la wilaya.

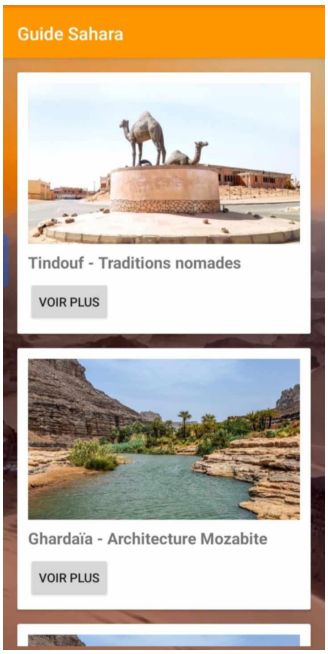


FIGURE 3.9 – l'interface "Notre wilaya"

Nous avons intégré une fonctionnalité qui permet à l'utilisateur de changer dynamiquement la langue de l'application (français/arabe/anglais), ce qui améliore l'accessibilité et l'expérience utilisateur pour un public diversifié.

FIGURE 3.10 – Application en arabe

3.3 Conclusion

En conclusion, ce mini-projet nous a permis de concevoir et de développer une application mobile Android servant de guide touristique pour une wilaya du sud algérien. L'objectif principal était de proposer un outil simple, intuitif et efficace pour faciliter l'organisation des séjours touristiques dans la région, en mettant à disposition des informations sur les villes, les sites à visiter, ainsi que les options d'hébergement et de restauration.

Ce travail nous a offert l'opportunité d'appliquer concrètement les notions abordées dans le cadre du module de développement mobile, telles que la conception d'interfaces en XML, la navigation entre les activités, la gestion des ressources, ainsi que l'adaptation multilingue de l'application.
numériques accessibles à tous.