

TAXI STAR

BEN SALAH Mohammed

...

Sommaire

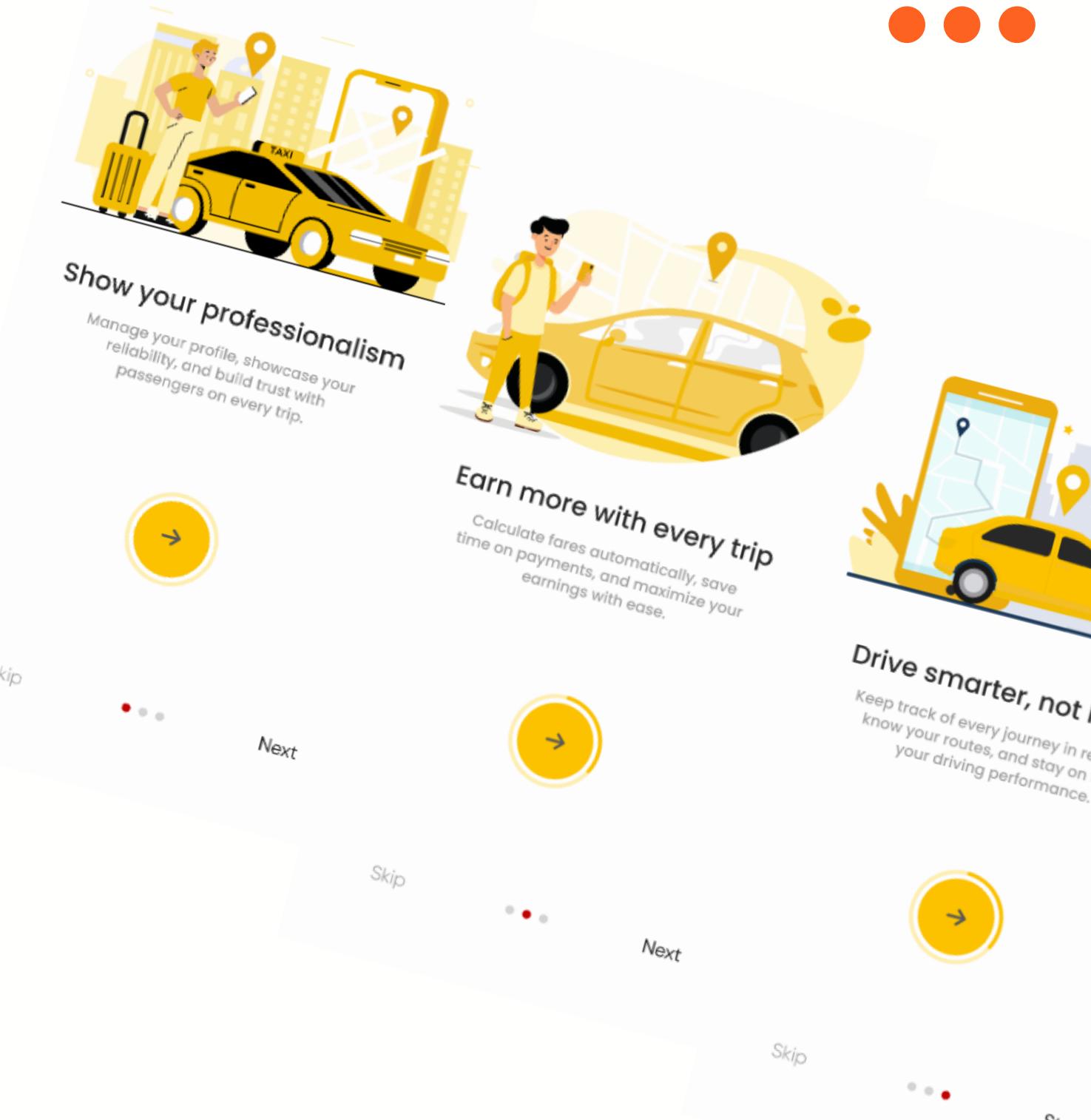
- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9

- Introduction
- Planification
- L'analyse et conception
- Les outiles utilisés
- Les fonctionnalités
- Les problèmes rencontrés
- La conclusion

- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9

Introduction

Le projet Taxi Star est une application mobile Android développée en Kotlin qui a pour objectif de simuler le fonctionnement d'un compteur de taxi numérique. Elle permet de calculer automatiquement le tarif d'une course en fonction de la distance parcourue et du temps écoulé, tout en affichant ces informations en temps réel. Ce projet m'a permis de mettre en pratique plusieurs notions apprises durant la formation, comme la conception d'interfaces en XML, la gestion des activités et fragments, l'intégration de Google Maps, la gestion des permissions avec EasyPermissions, ainsi que l'utilisation des notifications Android. À travers ce travail, j'ai cherché à proposer une alternative moderne et interactive au compteur traditionnel des taxis, en offrant une solution plus pratique, intuitive et adaptée aux technologies actuelles.



Planification

- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9

ID	Task Name	2025-11-03		2025-11-04		2025-11-05		2025-11-06		2025-11-07	
		20:00	08:00	20:00	08:00	20:00	08:00	20:00	08:00	20:00	08:00
1	Création et prototype (Figma + préparation)										
4	Profil, Scan (QR Code) et Historique										
2	Page principale (Home + Google Maps API)										
3	Logique et fonctionnalité complète										
5	Finalisation, déploiement et présentation										

1

2

3

4

5

6

7

8

9

Besoins fonctionnels :

- Calculer et afficher en temps réel la distance parcourue et le tarif.
- Enregistrer les informations des passagers dans la page profil.
- Générer un QR code avec les informations du passager.
- Scanner un QR code pour récupérer les informations lors d'une course.
- Afficher l'historique des courses pour consultation ultérieure.
- Permettre la personnalisation via la page paramètres.

Besoins non fonctionnels :

- Interface simple et intuitive pour une utilisation rapide.
- Application rapide et réactive sur tous les appareils Android.
- Sécurité des données des utilisateurs (profil et historique).

1

2

3

4

5

6

7

8

9

Architecture de l'application :

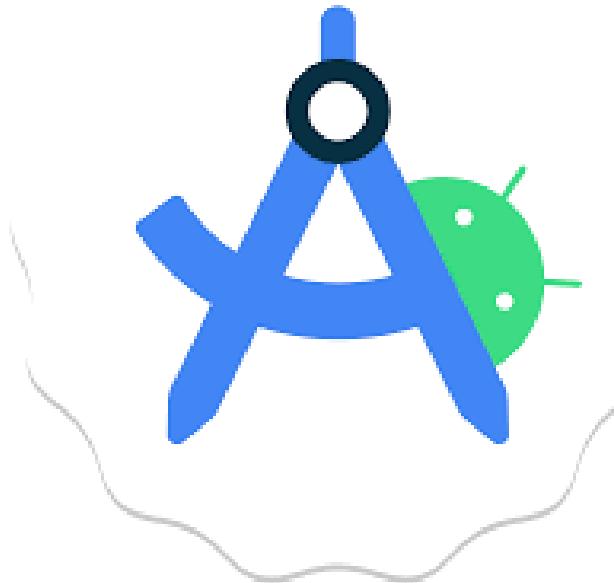
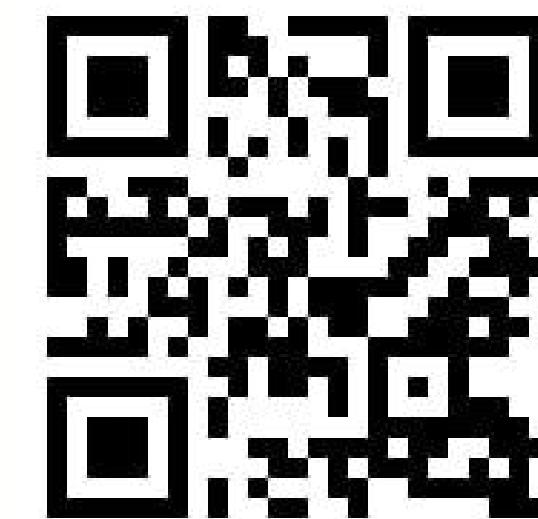
- MVVM (Model-View-ViewModel) pour séparer la logique métier de l'interface utilisateur.
- Modèles (Model) : Stockage des informations des passagers et des courses.
- Vue (View) : Écrans principaux (Home, Profil, Scan, Historique, Paramètres).
- ViewModel : Gestion des données entre l'interface et la base de données.

Diagramme des écrans :

- Home Page : Compteur de taxi + suivi GPS.
- Profil : Formulaire pour saisir et enregistrer les informations du passager.
- Scan QR : Scanner le QR code ou générer un code à partir des informations.
- Historique : Liste des courses passées avec distance, tarif et date.
- Paramètres : Changer les icônes, configuration de l'application, options de personnalisation.

Les outiles utilisés.

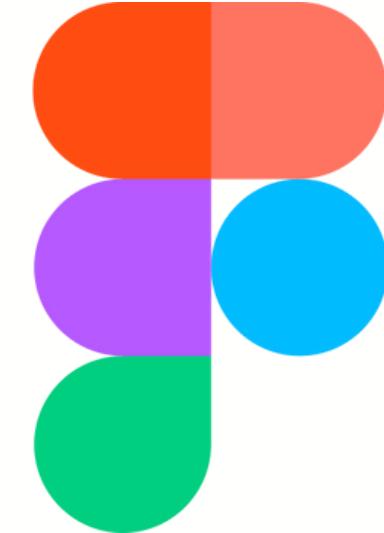
- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9



Kotlin

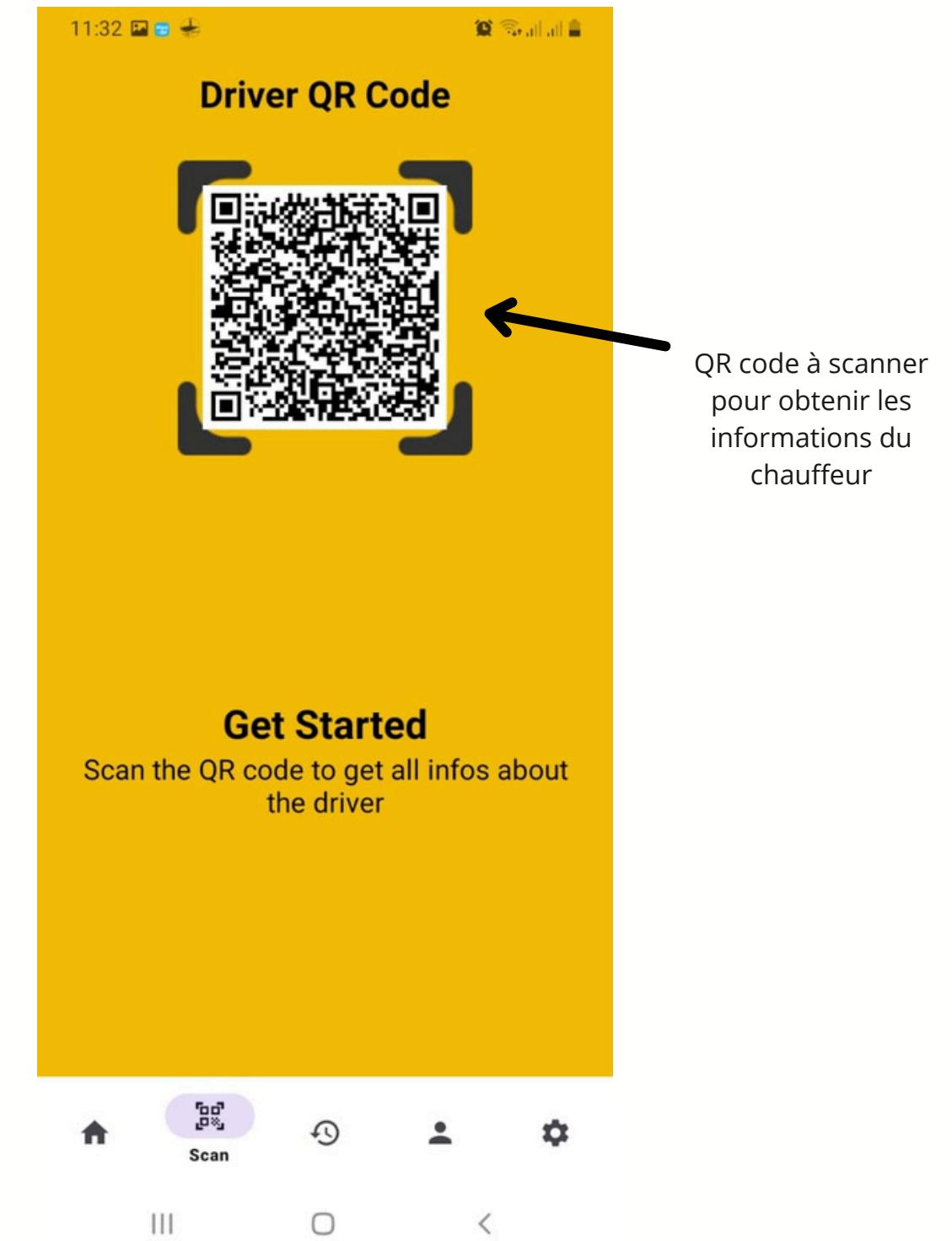
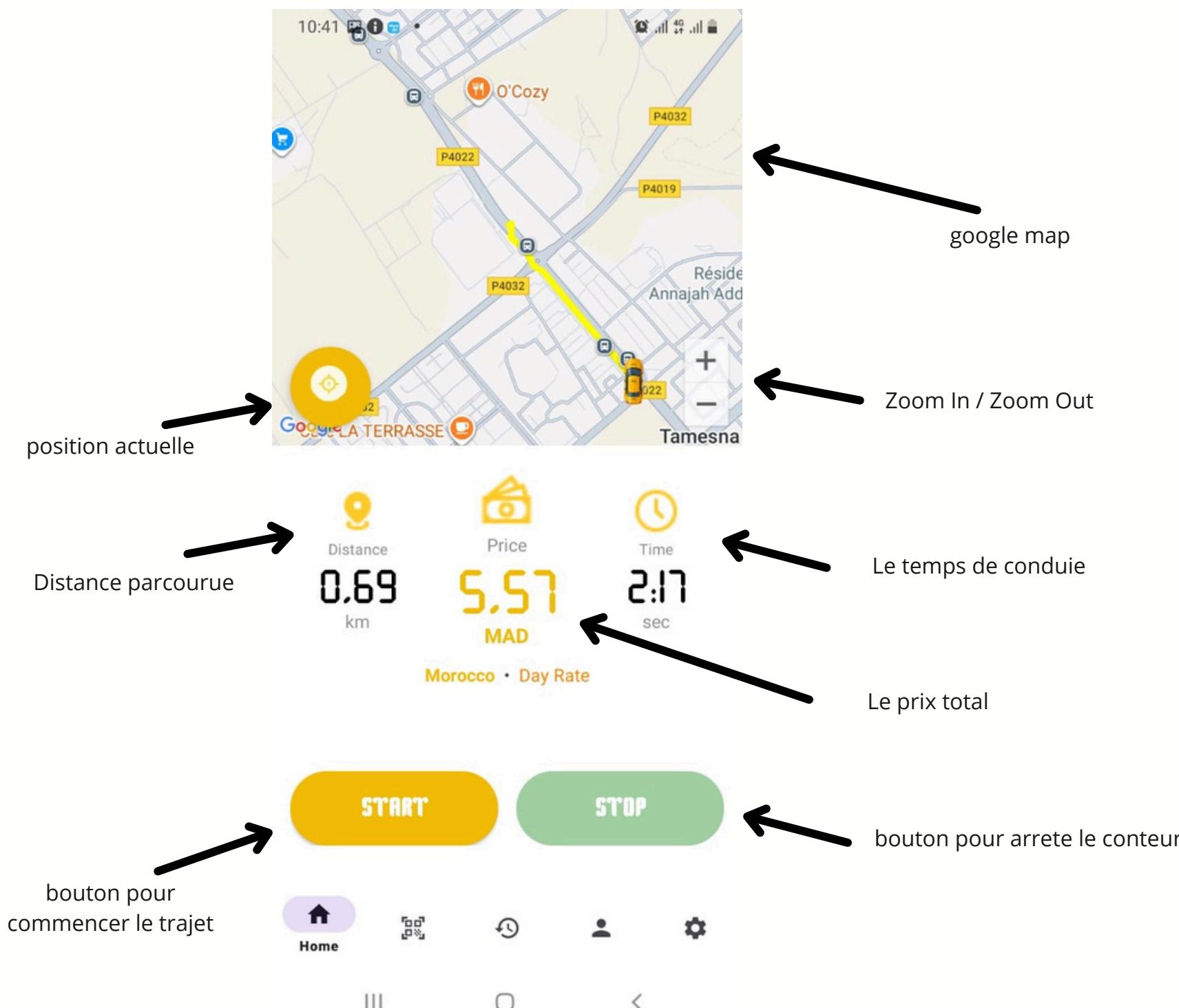


Google Maps API



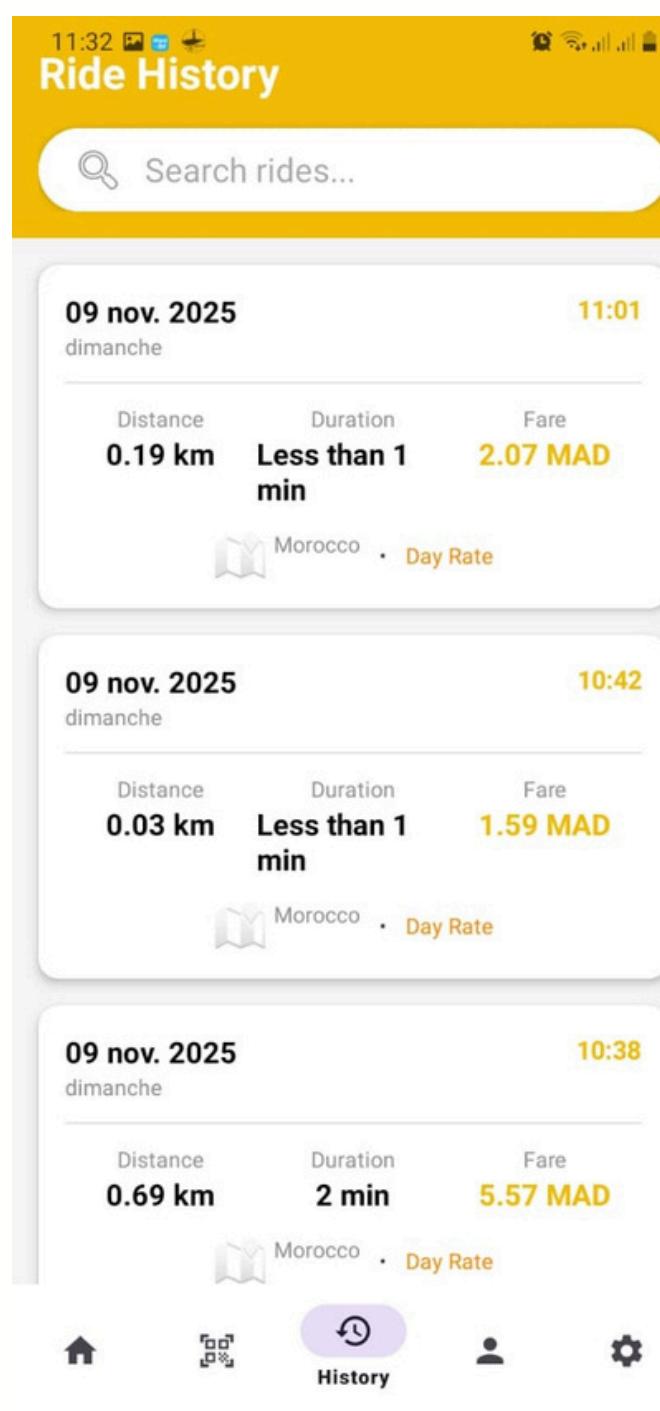
Les Fonctionnalités

- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9



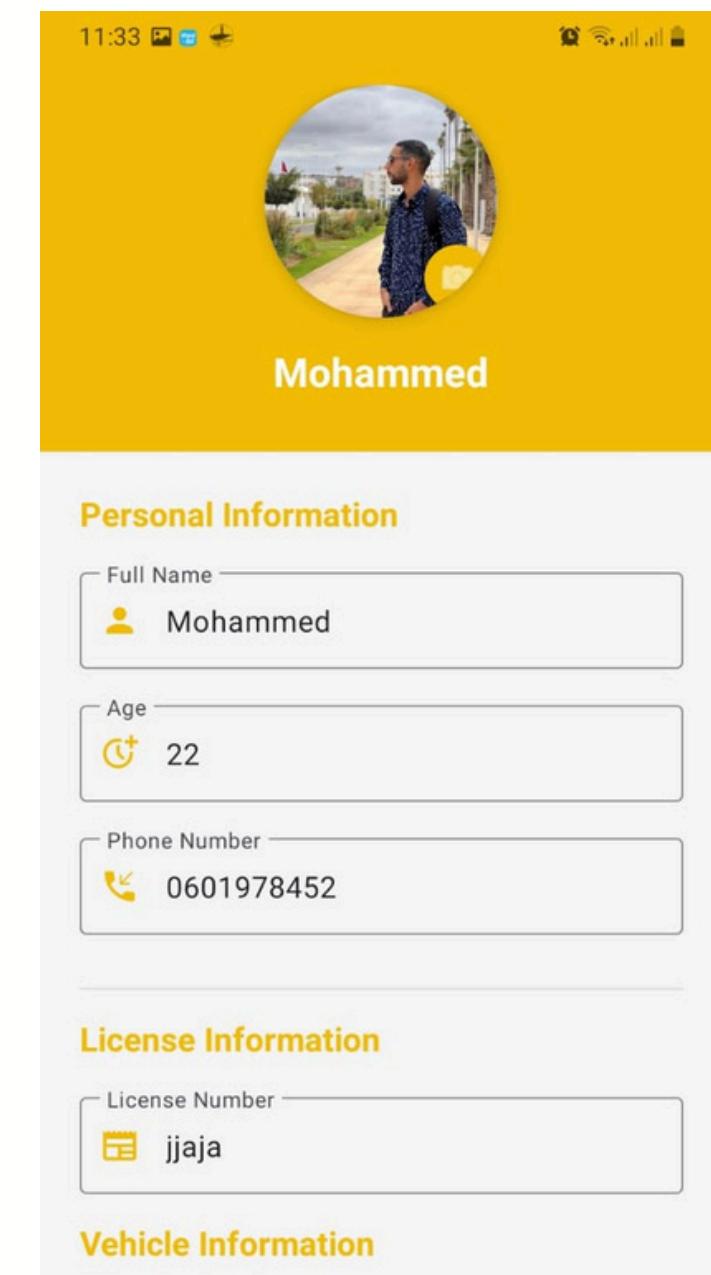
Les Fonctionnalités

- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9



Bar de recherche certain course de chauffeur.

L'historique d'une course contient les informations de la course



Profile de chauffeur.

Nom de chauffeur.

Age de chauffeur.

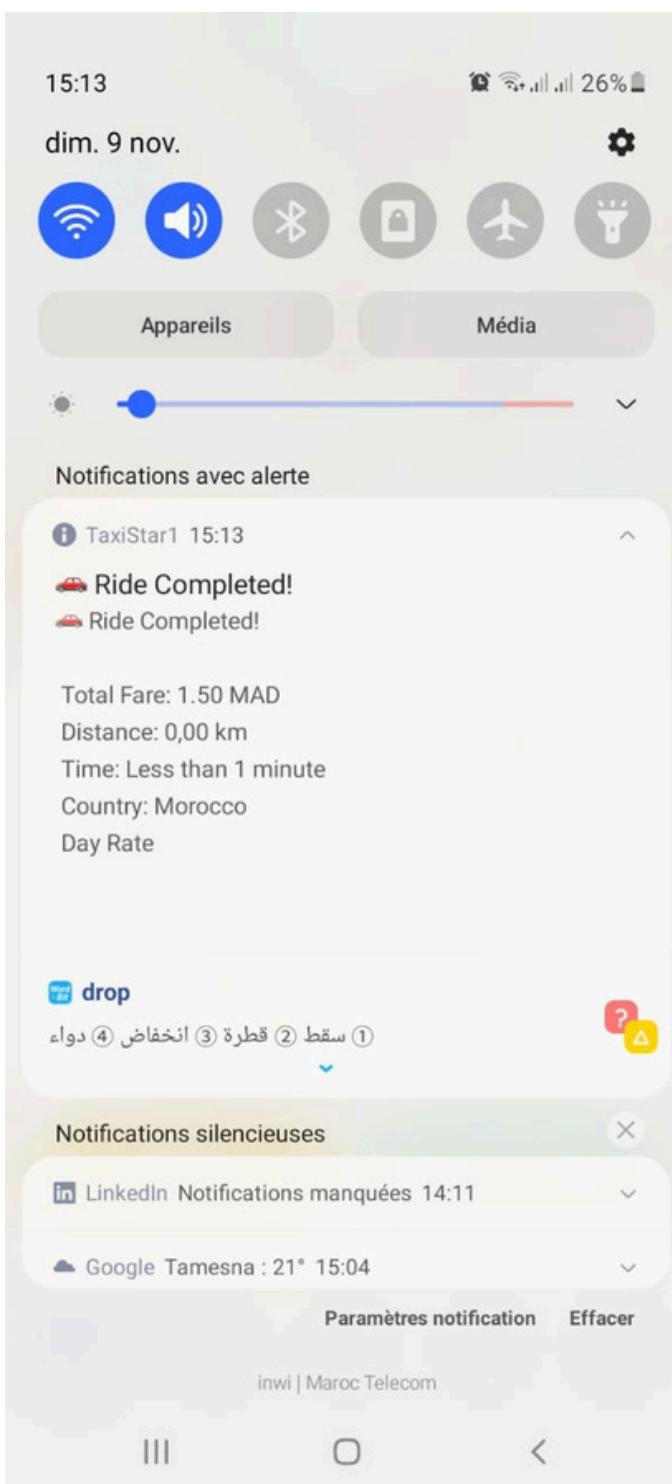
Numero de chauffeur.

Numero de Licence

Profile de chauffeur.

Les Fonctionnalités

- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9



Notifications de course.

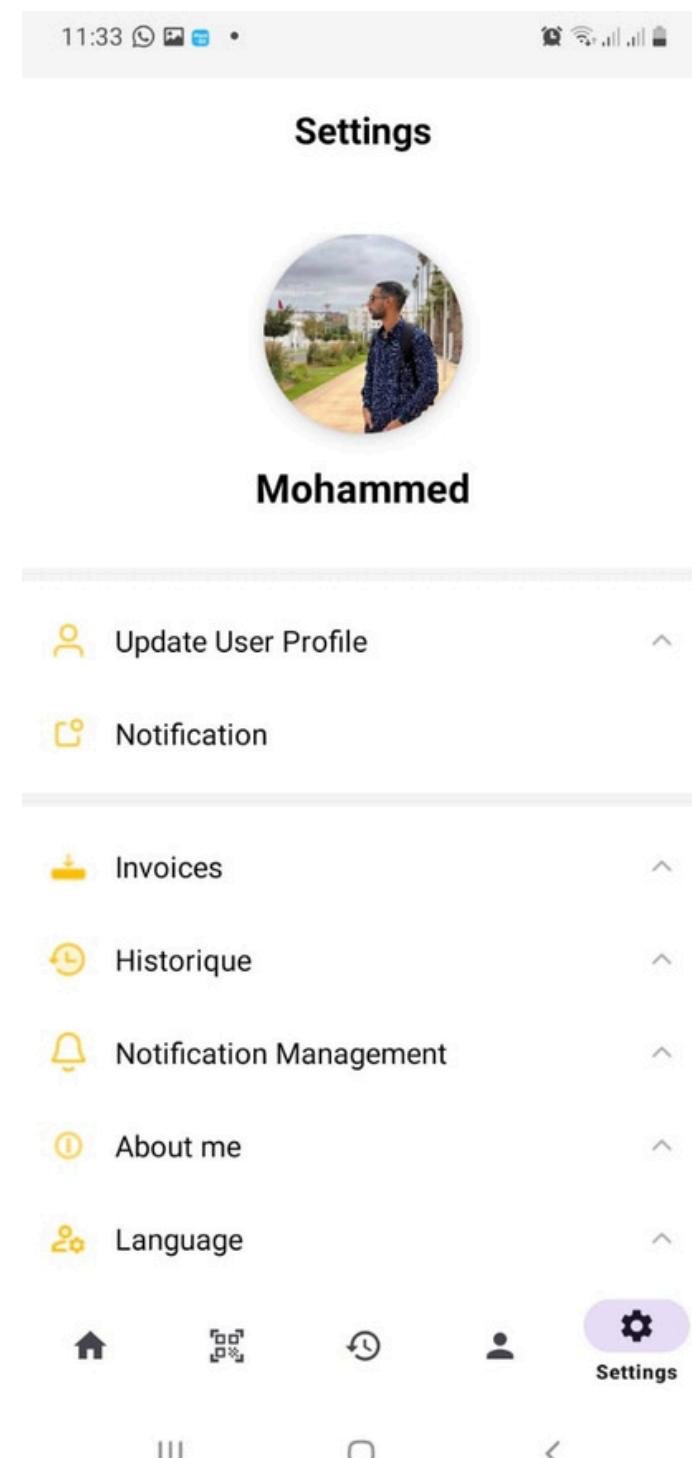


Photo de chauffeur.



modifier le profile.



Notification.



Imprimer la course.



Activer/Desactiver
Notification



Informations de
chauffeur.



Changement de
langue.

Les problèmes rencontrés

- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9

- Intégration de l'API Google Maps et gestion des permissions de localisation
- Gestion des erreurs lors du développement.
- Génération et lecture des QR codes avec synchronisation des données.
- Tests et corrections des bugs pour rendre l'application pleinement fonctionnelle.
- Précision du tarif en fonction du pays du chauffeur.



Conclusion.

- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9

Moi, j'ai réalisé l'application Taxi Star et j'ai pu mettre en pratique mes connaissances en développement Android. Malgré les difficultés rencontrées, comme l'intégration de Google Maps, la gestion des QR codes et l'adaptation du tarif selon le pays, j'ai réussi à les surmonter et à finaliser le projet. Cette expérience m'a permis de renforcer mes compétences techniques et d'apprendre à organiser et développer une application complète de manière autonome.

Merci
Beaucoup



- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9

