|  |
| --- |
| [YouCode] |
| SwiftCar |
| [ Tbibzat Charaf Eddine ] |

|  |
| --- |
| YouCode safi  24/01/2025 |

***Description :***

Ce projet vise à créer une plateforme révolutionnaire de covoiturage conçue pour établir une connexion directe et transparente entre les conducteurs et les passagers. Contrairement aux solutions classiques souvent rigides et standardisées, cette application se démarque par sa flexibilité accrue, sa capacité d’adaptation aux besoins individuels et son interface conviviale et moderne.

En mettant au cœur du projet une communication fluide et en temps réel, l’application permettra aux utilisateurs d’interagir directement pour négocier les tarifs, organiser des trajets personnalisés et partager des informations de manière instantanée. L’approche adoptée garantit une expérience utilisateur améliorée, intégrant des outils modernes comme des fonctionnalités de géolocalisation, des notifications interactives, et un système de messagerie intuitif.

Cette plateforme ne se limite pas à offrir un service de covoiturage standard, mais propose une solution conçue pour s’adapter aux exigences contemporaines et répondre aux attentes de divers profils d’utilisateurs, en alliant simplicité, efficacité, et innovation technologique.

***problématiques***

**1. Problématique de la disponibilité des conducteurs**

* Les passagers peuvent avoir du mal à trouver un conducteur disponible dans des zones moins fréquentées ou à des heures de faible demande.

**2. Problémat que de la sécurité pendant le trajet**

* Certains passagers peuvent se sentir inquiets à propos de leur sécurité, que ce soit en raison de la conduite du conducteur ou du manque de garanties en cas de comportement inapproprié.

**3. Problématique des tarifs imprévisibles**

* Les passagers peuvent être frustrés par des tarifs trop élevés, en particulier si les prix sont soumis à des variations en fonction de la demande ou de la négociation avec les conducteurs.

**4. Problématique des trajets longs et des délais excessifs**

* Un autre problème pour les passagers peut être un trajet excessivement long en raison de l'itinéraire choisi par le conducteur, qui peut ne pas être optimal.

**5. Problématique de la mauvaise communication entre passagers et conducteurs**

* Parfois, les passagers et les conducteurs ont du mal à communiquer, surtout si des problèmes surviennent, comme des changements de lieu de prise en charge ou des retards.

**6. Problématique de la qualité du service (confort et propreté)**

* Le confort du véhicule et sa propreté peuvent ne pas être à la hauteur des attentes du passager, surtout dans des trajets longs ou réguliers.

**7. Problématique de l'accessibilité pour les personnes à mobilité réduite**

* Les passagers ayant des besoins spécifiques (personnes handicapées ou âgées, par exemple) peuvent ne pas trouver de véhicules adaptés à leurs exigences.

**8. Problématique de l'absence d'assistance en cas de problème avec un conducteur**

* Si un passager a un problème avec un conducteur, par exemple un comportement inapproprié ou une situation conflictuelle, il peut ne pas savoir comment obtenir de l’aide rapidement.

***Objectifs de l'application***

* Ce que l'application vise à accomplir, en particulier en résolvant les problèmes identifiés dans la section précédente. Par exemple, fournir une solution de transport rapide, sûre, et pratique pour les passagers tout en garantissant une expérience fluide et fiable.
* **Objectifs spécifiques** :
  + Réduire le temps d'attente pour les passagers.
  + Améliorer la sécurité et la communication.
  + Offrir des tarifs transparents et prévisibles.
  + Assurer un service de qualité avec un feedback constructif pour les conducteurs.

***Planification et Développement***

Définir les étapes de développement et les délais pour chaque phase du projet.

* **Découpage du projet** :
  + Phase 1 : Analyse et conception.
  + Phase 2 : Développement du front-end et back-end.
  + Phase 3 : Tests et validation.
  + Phase 4 : Déploiement et mise en production.
* **Durée estimée de chaque phase**.
* **Méthodologie de gestion de projet** : Scrum.
* **Outils de gestion de projet** : Jira.

***Fonctionnalités de l'application***

* **Messagerie intégrée** : Un système de chat sécurisé pour la communication entre passagers et conducteurs, permettant des ajustements de dernière minute, des informations de sécurité, etc.
* **Système de notation et de feedback** : Permet aux passagers de noter les conducteurs et vice versa, afin d'assurer une bonne qualité de service et de responsabiliser les deux parties.
* **Inscription et gestion des profils** :

-Inscription par e-mail, numéro de téléphone ou via un réseau social.

-Profils personnalisés avec photos, numéro de téléphone, et historique des trajets.

-Vérification d’identité pour les conducteurs (upload de documents).

* **Gestion des trajets proposés** :
  + Ajouter, modifier, ou supprimer des trajets.
  + Fixer un tarif initial pour chaque trajet.
  + Recevoir des notifications pour les demandes de réservation.
* **Gestion des horaires** :

-Disponibilité configurable (exemple : "Je suis disponible entre 8 h et 18 h").

* **Gestion des utilisateurs** :

-Validation des comptes des conducteurs (après vérification des documents).

-Blocage ou suppression des comptes en cas de non-respect des règles.

* **Gestion des trajets** :

-Visualiser, modifier ou supprimer les trajets proposés.

* **Tableau de bord administratif** :

-Statistiques globales sur l’utilisation de l’application (nombre d’utilisateurs, trajets, revenus, feedback).

-Suivi des transactions et des paiements.

* **Tarification transparente et dynamique** : Affichage clair des coûts avant la réservation avec une option de négociation flexible.
* ect…

**Bonus :**

* **Géolocalisation en temps réel** : Permet aux passagers de suivre les conducteurs en temps réel et de garantir la rapidité et la précision du service.
* **Paiements sécurisés** : Intégration de plateformes de paiement comme **Stripe**, **PayPal**, ou des options locales pour garantir des transactions fiables.
* **Alertes de sécurité et bouton d'urgence** : Pour une tranquillité d’esprit pendant le trajet.

***Architecture de l'application***

**1. Présentation générale de l'architecture**

* L'application sera basée sur une architecture **client-serveur** pour assurer une communication fluide entre l'utilisateur (client) et les services du système (serveur).
* Les trois principales couches de l'architecture :
  + **Frontend (interface utilisateur)** : Partie visible de l'application utilisée par les conducteurs et les passagers.
  + **Backend (logique métier)** : Partie invisible qui gère les données, les transactions et les fonctionnalités principales.
  + **Base de données** : Stocke les informations critiques comme les profils des utilisateurs, les trajets, les paiements, etc.

**2. Technologies utilisées**

* **Frontend** :
  + **Framework** : React.js.
  + **Bibliothèques supplémentaires** : MUI ,TailwindCSS pour le design, Redux pour la gestion d’état…
* **Backend** :
  + **Framework** : Laravel (PHP) pour gérer la logique métier et les APIs REST.
  + **API de géolocalisation** : Google Maps API pour suivre les trajets en temps réel.
  + **Authentification** : Laravel Passport pour un accès sécurisé.
* **Base de données** :
  + **SGBD** : PostgreSQL pour les données relationnelles (utilisateurs, trajets, transactions).
* **Outils et environnements de développement** :
  + GitHub/GIT, IDE (VS Code).
  + StarUML
  + Postman
* **Notifications** : Firebase pour les notifications push (sur mobile).
* **Paiements sécurisés** : Intégration avec Stripe, PayPal ou CMI (option locale pour le Maroc).

**5. Scalabilité et performance**

* L’application sera conçue pour être scalable et pouvoir supporter un grand nombre d’utilisateurs en simultané.

***Définition des acteurs***

Cette section décrit les différents acteurs qui interagiront avec l'application, leurs rôles respectifs et les fonctionnalités auxquelles ils auront accès.

**1. Les acteurs principaux**

* **Passager**  
  Le passager est l'utilisateur final cherchant à réserver un trajet.
  + **Rôles et responsabilités** :
    - Créer un compte et se connecter.
    - Rechercher un trajet en fonction de sa localisation et de sa destination.
    - Contacter les conducteurs via la messagerie intégrée.
    - Réserver un trajet après avoir négocié le prix.
    - Suivre le conducteur en temps réel grâce à la géolocalisation.
    - Noter et commenter le conducteur après le trajet.
    - Effectuer un paiement en ligne ou en espèces (selon les options disponibles).
* **Conducteur**  
  Le conducteur est l'utilisateur qui propose des trajets aux passagers.
  + **Rôles et responsabilités** :
    - Créer un compte conducteur avec validation (ex. : vérification des papiers du véhicule).
    - Publier des trajets avec des détails comme l’itinéraire, les horaires, et le tarif initial.
    - Répondre aux demandes de négociation des passagers via la messagerie.
    - Accepter ou refuser des réservations.
    - Fournir des informations sur la localisation en temps réel pendant le trajet.
    - Recevoir les paiements des passagers.
    - Noter les passagers pour maintenir la qualité des interactions.

**2. Les acteurs secondaires**

* **Administrateur de la plateforme**  
  L'administrateur est responsable de la gestion et de la maintenance de l'application.
  + **Rôles et responsabilités** :
    - Valider les comptes conducteurs pour garantir leur authenticité.
    - Superviser les interactions entre conducteurs et passagers.
    - Gérer les réclamations et résoudre les litiges entre utilisateurs.
    - Gérer les paiements et s’assurer qu’ils respectent les réglementations locales.
    - Ajouter ou supprimer des fonctionnalités selon les retours des utilisateurs.
    - Surveiller les performances de la plateforme et assurer son bon fonctionnement.

**3. Acteurs techniques (indirects)**

* **Équipe de développement** : Responsables du développement et de la maintenance technique de l'application.
* **Fournisseurs de services tiers** :
  + API de géolocalisation (Google Maps API).
  + Plateformes de paiement (Stripe, PayPal, etc.).
  + Services de notifications .

**4. Relations entre les acteurs**

* Les **passagers** interagissent directement avec les **conducteurs** via la messagerie intégrée et le système de réservation.
* Les **conducteurs** et les **passagers** reçoivent un support technique et une médiation de l’**administrateur** en cas de problème.
* L’**administrateur** supervise les interactions globales pour garantir le bon fonctionnement de l’application et le respect des règles.

***Les Diagrammes***

**1 : use case**

**Une image contenant diagramme, dessin, croquis, ligne

Le contenu généré par l’IA peut être incorrect.**

**Une image contenant croquis, dessin, diagramme, ligne

Le contenu généré par l’IA peut être incorrect.**

**2 : Diagramme de Classe**

**Une image contenant texte, diagramme, Plan, Dessin technique

Le contenu généré par l’IA peut être incorrect.**

***Charte graphique***

Cette section définit l'identité visuelle de l'application et guide le choix des couleurs, typographies, et autres éléments de design pour garantir une cohérence visuelle.

**Maquettage et Wireframe**