

Rapport de Projet

Covoiturage Express TN

Réalisé par :

Sfaihi Ridha
Nour Mansour
Smeti Mohamed Taieb

Sommaire

Table des matières

Sommaire	1
Introduction Générale	2
Contexte et Problématique	3
Présentation du domaine :	3
Identification du besoin :	3
Justification du projet :	3
Objectifs du Projet.....	3
Objectif principal	3
Objectifs secondaires	4
Périmètre	4
Structure du Rapport.....	4
Phase d'Analyse et Conception	5
Analyse Fonctionnelle Détailnée	5
Rôles des Utilisateurs	5
Fonctionnalités Clés	5
Diagrammes de Cas d'Utilisation	5
Exigences Non-Fonctionnelles.....	5
Modélisation.....	7
Conception : Diagramme Cas d'utilisation :	7
Conception : Diagramme de Classe :	8
Phase de Réalisation Technique.....	9
Architecture Technique	9
Présentation de la Solution	9
Choix Technologiques	9
Développement Backend	9
Description des Contrôleurs et Routes	9
Implémentation des Logiques Métier	10
Développement Frontend	11
Interface Utilisateur (UI)	11
Conclusion et Perspectives.....	14
Conclusion Générale.....	14

Liste De Figures :

Figure 1:cas d'utilisation.....	7
Figure 2 : Diagramme de Classe	8
Figure 3:Connexion.....	11
Figure 4:créer un compte.....	12
Figure 5:Page d'accueil.....	12
Figure 6:Consulter les trajets.	13
Figure 7:réservation du trajet.	13

Introduction Générale

Contexte et Problématique

Présentation du domaine :

Le covoiturage est une solution de mobilité clé, répondant aux enjeux environnementaux et de congestion. En Tunisie, le besoin est particulièrement aigu en raison des limites des transports publics, créant des pics de difficulté, notamment les lundis matin. C'est pourquoi le projet « Covoiturage Express TN » propose une plateforme web pour structurer et sécuriser le partage de trajets, améliorant ainsi l'efficacité des déplacements dans ces zones critiques.

Identification du besoin :

La pratique actuelle du covoiturage en Tunisie souffre de fragmentation, d'un manque de sécurité et d'inefficacité. La problématique principale se concentre sur trois axes : la mobilité limitée dans certaines zones, le coût élevé du transport individuel (et les embouteillages associés), et l'inefficacité causée par les places vacantes dans les véhicules. L'absence d'une plateforme fiable nuit à la mise en relation et engendre un déficit de confiance. Le projet « Covoiturage Express TN » a pour mission de fournir un service sécurisé, transparent et adapté aux besoins locaux.

Justification du projet :

L'application web est le format le plus pertinent car il garantit une accessibilité universelle (mobile/desktop) essentielle pour structurer un marché fragmenté. Elle permet d'implémenter efficacement les mécanismes de sécurité (authentification, réputation) et de transparence nécessaire au rétablissement de la confiance des utilisateurs, tout en offrant une scalabilité pour gérer les pics de demande.

Objectifs du Projet

Objectif principal

L'objectif principal de « Covoiturage Express TN » est d'établir un lien direct et fiable entre les conducteurs proposant des places libres et les passagers recherchant un trajet. Cette fonctionnalité de mise en relation optimale est au cœur du système et doit garantir que les offres de trajets correspondent précisément aux demandes en termes d'horaires, d'itinéraires et de fiabilité.

Objectifs secondaires

Ces objectifs visent à maximiser l'impact de la plateforme au-delà de la simple mise en relation. Ils incluent la réduction des coûts de transport pour les utilisateurs et la diminution de la congestion routière. De plus, ils se concentrent sur l'amélioration de la mobilité dans les zones mal desservies. Enfin, et de manière cruciale, le projet s'engage à créer un environnement de confiance en garantissant une plateforme sécurisée et transparente via un système de réputation et de modération.

Périmètre

Le projet se définit strictement comme une application web transactionnelle. Il couvre l'ensemble du cycle de vie du covoiturage, depuis la recherche et l'offre de trajets jusqu'à la réservation des places. Le périmètre n'inclut pas, dans sa version initiale, de fonctionnalités externes complexes comme l'intégration de systèmes de paiement ou d'applications mobiles natives, se concentrant sur les fonctionnalités de base nécessaires à la validation du concept.

Structure du Rapport

Le présent rapport est structuré en cinq grandes parties. La première partie introduit le contexte, la problématique et les objectifs du projet. La deuxième partie détaille l'Analyse et la Conception (fonctionnelle et de données). La troisième partie présente la réalisation technique (architecture et choix technologiques). La quatrième partie est consacrée aux Tests et au Bilan de l'implémentation. Enfin, la cinquième partie conclut le travail et ouvre sur les perspectives d'évolution de la plateforme.

Phase d'Analyse et Conception

Analyse Fonctionnelle Détaillée

Rôles des Utilisateurs

- Le système de covoiturage repose sur l'interaction de trois acteurs principaux :

- Conducteur : Il est responsable de la publication et de la gestion de ses offres de trajets. Ses fonctions clés incluent la définition des itinéraires, des horaires et du prix, la mise à jour des disponibilités de places, la gestion des demandes de réservation, et la communication avec les passagers. Il est également évalué après chaque trajet.
- Passager : Son rôle principal est de rechercher des trajets disponibles selon ses critères (lieu, heure, destination), d'effectuer une demande de réservation, et de noter le conducteur et son expérience après le voyage.
- Système Automatique : Cet acteur non-humain gère les logiques d'arrière-plan essentielles au bon fonctionnement de la plateforme. Cela comprend la gestion des correspondances, l'actualisation en temps réel des disponibilités après réservation, le calcul des coûts, et surtout, l'implémentation des mécanismes de gestion de la réputation et du blocage automatique des utilisateurs ne respectant pas les règles (fiabilité, annulations abusives).

Fonctionnalités Clés

- Les fonctionnalités principales de la plateforme structurent l'expérience utilisateur et les logiques métiers :

- Rechercher et Consulter un trajet : Permettre aux passagers d'interroger la base de données par critères (départ, arrivée, date) et d'afficher les résultats détaillés incluant le prix, le véhicule et le profil du conducteur.
- Proposer un trajet : Faciliter la publication des offres par les conducteurs (saisie de l'itinéraire, des places disponibles, du coût et des préférences spécifiques).
- Réserver une place : Permettre aux passagers d'envoyer une demande de réservation qui sera gérée par le système pour valider la disponibilité et informer le conducteur.
- Gérer les réservations : Fournir des tableaux de bord aux conducteurs pour accepter/refuser les demandes et aux passagers pour suivre l'état de leurs réservations.
- Noter et Évaluer : Mettre en place un mécanisme bidirectionnel post-trajet permettant aux conducteurs et aux passagers de s'attribuer une note et un commentaire, alimentant le système de réputation.

Diagrammes de Cas d'Utilisation

Exigences Non-Fonctionnelles

- Sécurité : L'exigence de sécurité est primordiale pour garantir la confiance des utilisateurs. Cela impose l'adoption de mesures robustes telles que : le hachage fort des

mots de passe (utilisation d'algorithmes modernes type bcrypt) avant le stockage ; le chiffrement des informations sensibles (e.g., données personnelles, numéro de téléphone) lors de leur transit et de leur stockage ; et un processus de vérification des utilisateurs (vérification d'identité ou du véhicule) pour prévenir les abus.

- Performance : L'application doit offrir une expérience fluide, notamment lors des requêtes critiques. La vitesse de recherche de trajets doit être optimisée (temps de réponse inférieur à 1-2 secondes), nécessitant une indexation efficace de la base de données. L'interface utilisateur doit également être réactive, avec des temps de chargement minimes.
- Simplicité d'utilisation (UX/UI) : La plateforme doit être intuitive pour maximiser l'adoption. Cela passe par une conception claire et épurée (UI) et un parcours utilisateur logique (UX) nécessitant un minimum de clics pour effectuer les actions principales (rechercher, proposer, réserver). L'adaptabilité sur les différents appareils (responsive design) est essentielle.
- Scalabilité : Bien que le projet soit initialement local, l'architecture doit être conçue pour gérer une croissance future du nombre d'utilisateurs et de trajets. Le choix d'une architecture N-tiers et l'utilisation de MySQL permet d'envisager une montée en charge progressive en optimisant les requêtes et, si nécessaire, en migrant vers des solutions d'hébergement plus puissantes sans refonte majeure.

Modélisation

Conception : Diagramme Cas d'utilisation :

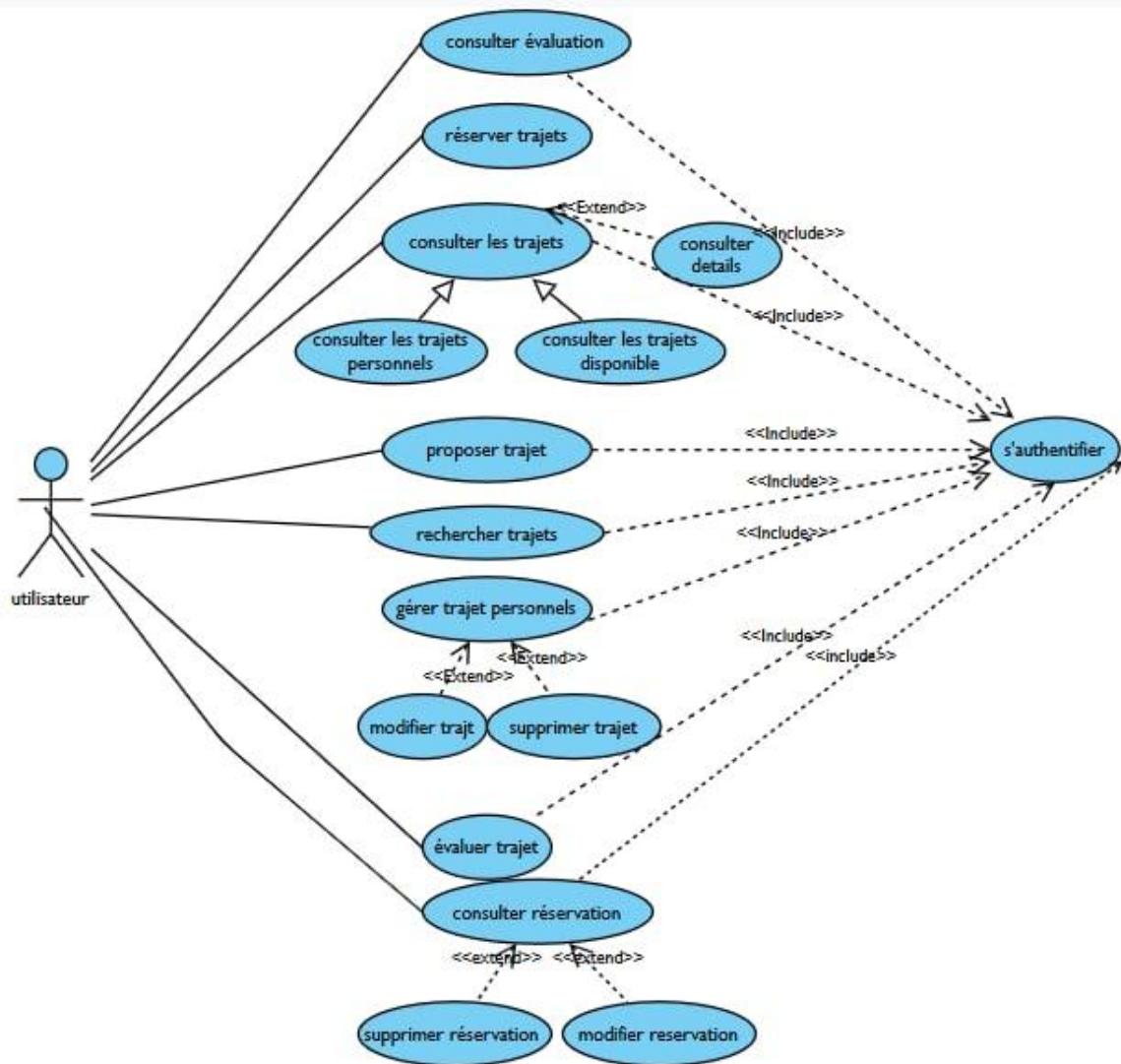


Figure 1:cas d'utilisation

Conception : Diagramme de Classe :

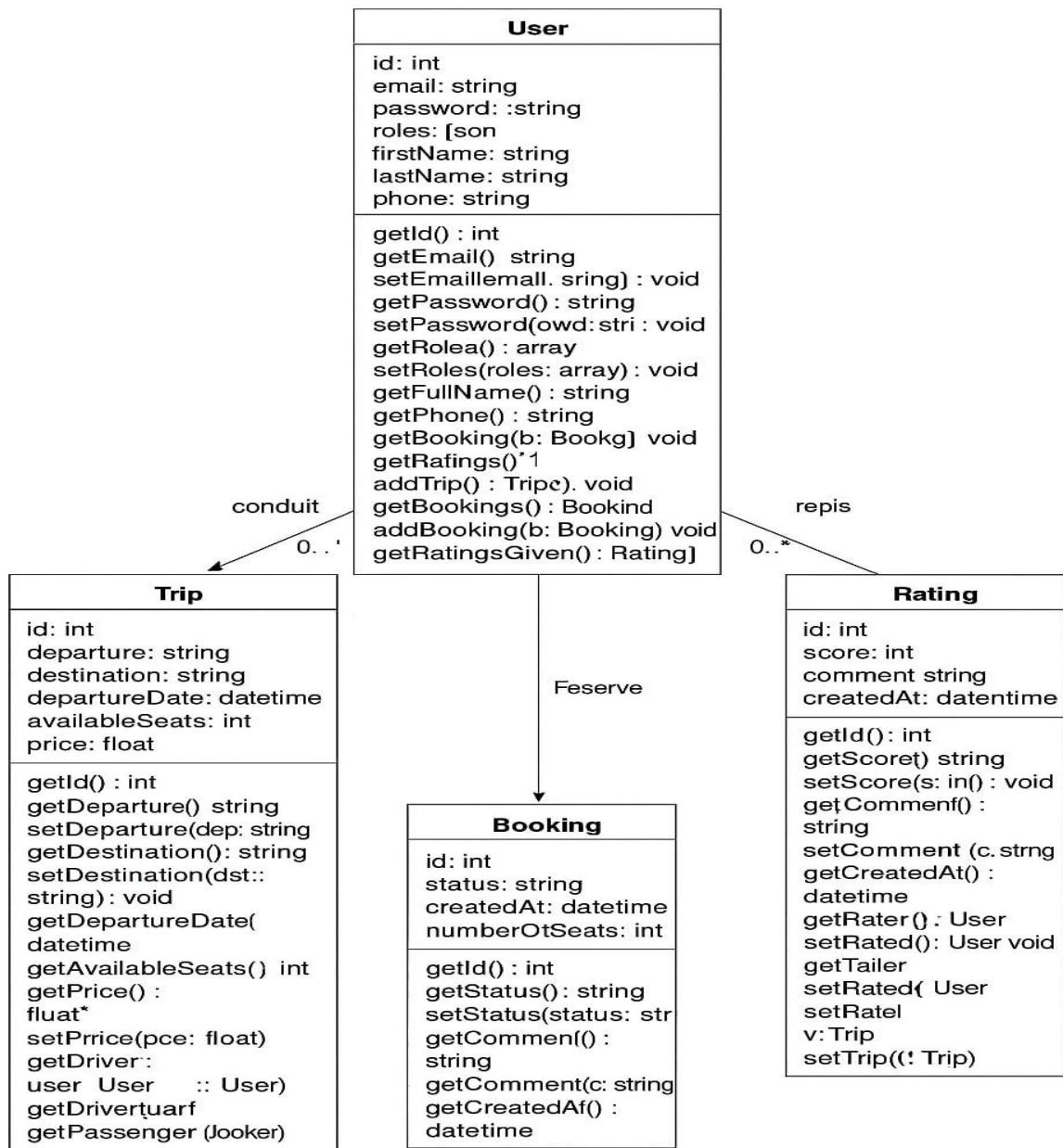


Figure 2 : Diagramme de Classe

Phase de Réalisation Technique

Architecture Technique

Présentation de la Solution

- Pour répondre aux exigences de sécurité, performance et scalabilité, le projet Covoiturage Express TN repose sur une architecture N-tiers (ou 3-tiers) standard pour les applications web. Cette approche sépare logiquement les préoccupations en trois couches distinctes :

1. La Couche Présentation (Frontend) : Gère l'interface utilisateur et l'interaction directe via le navigateur.
2. La Couche Application (Backend/API REST) : Contient la logique métier, assure l'authentification, le calcul des correspondances et la gestion de la réputation.
3. La Couche Données (Base de Données MySQL) : Stocke toutes les informations de manière sécurisée (profils, trajets, réservations).

Cette segmentation est pertinente car elle garantit une meilleure sécurité (l'accès à la BDD est protégé par la couche Application), une maintenance simplifiée (les modifications du Frontend n'impactent pas la logique Backend) et permet une montée en charge horizontale (ajout facile de serveurs pour chaque couche en cas d'augmentation du trafic).

Choix Technologiques

- Frontend & Backend (Full Stack) : Le projet sera développé en utilisant le Framework Symfony (PHP) en mode Full Stack. Ce choix est hautement pertinent car Symfony gère à la fois la logique métier (Backend) via ses contrôleurs et services, et la génération des vues (Frontend) via le moteur de Template Twig. Cela assure une cohérence parfaite entre les couches, une rapidité de développement accrue (moins de configuration d'API séparée), et offre nativement des mécanismes de sécurité robustes essentiels pour la gestion des utilisateurs et des sessions (hachage, CSRF, etc.).
- Base de Données : L'utilisation de MySQL (langage SQL) est maintenue comme SGBD relationnel, conformément au cahier des charges. L'intégration via l'ORM Doctrine (fourni par Symfony) permet de manipuler les données de manière orientée objet tout en garantissant l'intégrité référentielle et la robustesse de l'environnement SQL, indispensable pour les transactions de réservation.
- Hébergement : Pour le contexte de ce projet éducatif, l'hébergement est assuré par un environnement XAMPP ou un serveur local. Cette solution permet un déploiement rapide et un contrôle total de l'environnement, facilitant les tests et la démonstration locale de l'application complète (Symfony, Apache, et MySQL).

Développement Backend

Description des Contrôleurs et Routes

- Étant donné l'utilisation de Symfony en Full Stack, la logique métier est directement exposé via des Contrôleurs (Controllers) plutôt qu'une API REST externe. Chaque fonctionnalité

(authentification, recherche, publication, gestion) est mappée à une URL spécifique via un système de Routes Symfony. Les routes principales du système sont :

- `/connexion et /inscription` : Gèrent l'authentification des utilisateurs (utilisation du composant Security de Symfony).
- `/trajets/rechercher` : Permet au passager de soumettre ses critères de recherche et affiche la liste des résultats (méthode GET/POST).
- `/trajets/proposer` : Gère la soumission des formulaires pour la publication d'un nouveau trajet par le conducteur (méthode POST).
- `/reservations/{id}/reserver` : Déclenche la logique de création d'une nouvelle réservation pour un trajet spécifique.
- `/profil/reservations et /profil/trajets` : Affichent les tableaux de bord personnalisés pour la gestion des activités (réservations en cours, trajets proposés, notes reçues).

Ces routes, gérées par le Framework, permettent au Backend de traiter les données (via Doctrine) et de retourner directement les pages HTML/Twig au navigateur.

Implémentation des Logiques Métier

- Les fonctionnalités critiques sont encapsulées dans des Services Symfony pour garantir la réutilisation et la séparation des préoccupations :
 - Recherche de Correspondances (Matching) : La recherche est gérée par un Service de Requête (e.g., `TrajetRepository` ou `MatchingService`) qui exécute une requête DQL (Doctrine Query Language) complexe. Cette requête interroge la table `TRAJET` en appliquant plusieurs critères :
 1. Correspondance Géographique : Utilisation d'une fonction SQL ou Doctrine pour filtrer les trajets dont le point de départ se situe dans un rayon défini autour du lieu de départ demandé par le passager, et le point d'arrivée dans un rayon autour de la destination demandée.
 2. Date et Heure : Filtrage des trajets disponibles dans une plage horaire acceptable par le passager (par exemple, +/- 30 minutes de l'heure souhaitée).
 3. Disponibilité : S'assurer que le nombre de `places_disponibles` sur le trajet est supérieur ou égal au nombre de places demandées par l'utilisateur.
 4. Priorisation : Les résultats sont ensuite triés, priorisant les trajets avec la meilleure Note Moyenne du conducteur (issue du système de réputation) et le chemin le plus court (si des données de distance sont disponibles).
 - Gestion de la Réputation (Système de Notation) : Cette logique est gérée par un Service de Notation (`NotationService`) qui est appelé après la validation d'un trajet complété. Lorsqu'un utilisateur soumet une `NOTATION` pour un autre utilisateur :
 1. Le service sauvegarde la nouvelle entrée dans la table `NOTATION`.
 2. Il recalcule ensuite la note moyenne du destinataire (`UTILISATEUR.note_moyenne`) en faisant la moyenne de toutes les notes reçues (un `UPDATE` ciblé sur la table `UTILISATEUR`).
 3. Le résultat (la nouvelle note moyenne) est immédiatement affiché sur le profil du conducteur ou passager, renforçant la transparence.

- Blocage Automatique des Utilisateurs : Le système met en œuvre une politique stricte pour gérer la fiabilité, traitée par un Service de Fiabilité (`FiabiliteService`) qui tourne potentiellement via une tâche planifiée (Cron) ou est déclenché après chaque annulation.
 1. Indicateurs de Performance : Le service suit le taux d'annulation du conducteur (nombre d'annulations / nombre de trajets proposés) et le taux de non-présentation du passager.
 2. Seuil Critique : Si un utilisateur dépasse un seuil critique (par exemple, plus de 15% d'annulations sans motif légitime ou plus de 3 non-présentations), le statut de l'utilisateur est automatiquement mis à jour à 'Bloqué' dans la base de données.
 3. Impact : Les utilisateurs bloqués ne peuvent plus publier de trajets ni effectuer de réservations, protégeant ainsi la communauté contre les comportements abusifs.

Développement Frontend

Interface Utilisateur (UI)

L'interface utilisateur a été conçue en mettant l'accent sur la simplicité et l'accessibilité mobile, conformément aux exigences non-fonctionnelles. L'utilisation du moteur de template Twig de Symfony, associée à des frameworks CSS modernes (e.g., Bootstrap ou Tailwind CSS), garantit un *design responsif* sur tous les appareils (ordinateurs de bureau, tablettes et mobiles). La charte graphique utilise des couleurs douces mais contrastées pour une lisibilité maximale.

- **Authentification et Inscription :** Ces pages sont les portes d'entrée de l'application. La conception est minimalistique pour réduire la friction. Les formulaires incluent des validations côté client pour améliorer l'expérience et des indicateurs de sécurité (e.g., force du mot de passe) dès l'inscription.

The screenshot shows the 'Connexion' (Connection) page of the Covoiturage TN website. The top navigation bar includes the site name 'Covoiturage TN' and links for 'Accueil' and 'Trajets'. On the right side of the header, there are 'Connexion' and 'Inscription' buttons. The main content area is titled 'Connexion' and contains two input fields: 'Email' and 'Mot de passe', both with placeholder text. Below these fields is a blue 'Se connecter' button. At the bottom of the form, there is a link 'Pas encore de compte ? [Créer un compte](#)' and a small copyright notice '© 2025 Covoiturage TN. Tous droits réservés.'

Figure 3: Connexion.

Créer un compte

Prénom

Nom

Email

Mot de passe

Confirmer le mot de passe

[S'inscrire](#)

[Déjà un compte ? Se connecter](#)

© 2025 Covoiturage TN. Tous droits réservés.

Figure 4:créer un compte.

- Page d'Accueil / Recherche de Trajet :** Elle est l'écran principal après connexion. Un grand formulaire de recherche intuitif (Départ, Arrivée, Date) est mis en évidence, permettant aux passagers de démarrer immédiatement leur processus. Un bouton ou un onglet clair permet aux conducteurs de basculer vers le formulaire de "Proposition de Trajet".

Covoiturage TN Accueil Trajets Mes réservations Notes  sfaihi ▾

Bienvenue sur Covoiturage TN

Partagez vos trajets, économisez et rencontrez de nouvelles personnes !

Trouvez un trajet ou proposez le vôtre en quelques clics.

[+ Proposer un trajet](#) [🔍 Rechercher un trajet](#)


Économique
 Partagez les frais de transport et réduisez vos dépenses de déplacement.


Écologique
 Réduisez votre empreinte carbone en partageant votre véhicule.


Convivial
 Rencontrez de nouvelles personnes et voyagez dans une ambiance agréable.

Comment ça marche ?

Figure 5:Page d'accuei.

- Consultation et Réservation :** La page des résultats de recherche affiche chaque trajet sous forme de carte concise, présentant les informations essentielles (Heure, Prix, Note du Conducteur). Cliquer sur une carte mène à la page de détails du trajet, où la fonctionnalité de réservation est mise en avant pour un processus transactionnel rapide et clair.

Covoiturage TN Accueil Trajets Mes réservations Notes sfaihi Proposer un trajet

Trajets disponibles

tunis → sousse

Date : 29/04/2026 15:15

Places : 2

Prix : 15.00 DT

Conducteur : sfaihi

Détails Modifier

© 2025 Covoiturage TN. Tous droits réservés.

Figure 6: Consulter les trajets.

Covoiturage TN Accueil Trajets Mes réservations Notes sfaihi

Create new Booking

Nombre de places
1

Message au conducteur (optionnel)
bonjour, il y a encoure une place?

Enregistrer Retour

back to list

© 2025 Covoiturage TN. Tous droits réservés.

Figure 7:réservation du trajet.

- Navigation et Profil :** Une barre de navigation persistante ou un menu hamburger (pour le mobile) permet l'accès rapide au profil, aux trajets proposés/réservés et à la déconnexion. Les tableaux de bord personnels sont conçus pour offrir une vue d'ensemble claire des activités en cours (statut des réservations, notes reçues/données).

Conclusion et Perspectives

Conclusion Générale

Le projet **Covoiturage Express TN** a atteint son objectif principal : fournir une plateforme web transactionnelle, sécurisée et fonctionnelle, destinée à structurer le partage de trajets en Tunisie.

Grâce à une méthodologie de développement structurée (N-tiers, Full Stack Symfony), nous avons pu implémenter toutes les fonctionnalités cœur définies dans le cahier des charges, de l'authentification sécurisée à la gestion complète du cycle de vie des réservations. Le point fort du système réside dans l'intégration réussie de logiques métiers complexes centrées sur la fiabilité : un **système de notation** transparent, et un mécanisme de **blocage automatique** des utilisateurs non fiables.

L'implémentation de la recherche de trajets s'est limitée à la **simple présentation des données transactionnelles brutes** des trajets disponibles. Bien que cette version n'inclue pas de mécanisme de filtrage (par ville ou date), elle a permis de valider la **capacité du système à gérer les transactions de réservation (CRUD)** et à appliquer les règles de fiabilité, comme la notation.

Cependant, l'auto-critique du projet met en lumière quelques axes d'amélioration. Le choix du Full Stack Symfony, bien que garant de la sécurité et de la rapidité de développement, limite la fluidité de l'expérience utilisateur par rapport à une architecture SPA. De plus, la solution actuelle de recherche de trajets est rudimentaire et gagnerait énormément en efficacité par l'intégration future d'une API cartographique dédiée et d'une logique de filtrage avancée. Enfin, l'hébergement local actuel pose des limites à l'exigence de scalabilité.

Le projet est donc validé comme un **Prototype Opérationnel (MVP)**, répondant aux besoins initiaux. Les étapes futures, telles que le support multilingue et l'intégration d'un système de paiement, sont d'ores et déjà envisagées pour transformer cette preuve de concept en une solution commerciale complète et pérenne sur le marché tunisien.