

Documentation fonctionnelle

Plan de test

Stratégies de Test Approfondies

Tests Unitaires Approfondis

- **Couverture de Code** : Viser une couverture de code significative (par exemple, > 80%) pour s'assurer que la majorité du code est validée contre des bugs potentiels.
- **Tests de Composants UI** : Pour Flutter et React, implémenter des tests unitaires vérifiant le rendu et le comportement des composants UI en réponse aux interactions utilisateur.

Tests d'Intégration Avancés

- **APIs et Services Backend** : Tests d'intégration axés sur la communication entre le frontend et le backend, y compris les requêtes API RESTful, pour valider les réponses du serveur, les codes d'état HTTP, et l'intégrité des données échangées.
- **Intégration de Base de Données** : Tests vérifiant les interactions entre l'application et la base de données MySQL, en s'assurant que toutes les opérations CRUD sont correctement exécutées et que les transactions sont gérées efficacement.

Tests Système Complet

- **Scénarios d'Utilisation End-to-End (E2E)** : Développer des tests E2E simulant des parcours utilisateurs complets pour valider le flux de travail intégral de l'application, de l'inscription à la réservation de trajets, en passant par les paiements et les évaluations.
- **Tests Cross-Plateforme** : Assurer que l'application offre une expérience cohérente et fonctionnelle sur différentes versions d'OS mobiles, navigateurs web, et tailles d'écran.

Tests de Performance et de Charge

- **Scénarios de Charge** : Établir des scénarios de test simulant un nombre élevé d'utilisateurs simultanés effectuant des recherches de trajets, des réservations, et des paiements pour identifier les goulots d'étranglement et optimiser les performances.
- **Tests de Stress** : Pousser l'application à ses limites en termes de charge utilisateur pour évaluer sa stabilité sous stress extrême et déterminer les points de rupture.

Sécurité et Tests de Conformité

- **Tests d'Injection SQL et XSS** : Exécuter des tests spécifiques pour détecter les vulnérabilités aux injections SQL et aux attaques par script cross-site (XSS), en particulier dans les champs de saisie utilisateur et les paramètres des requêtes API.
- **Conformité aux Réglementations de Protection des Données** : Vérifier que l'application respecte les normes de protection des données (par exemple, GDPR), notamment en

matière de consentement des utilisateurs, de cryptage des données, et de politiques de confidentialité.

Cas de Test Détaillés

Inscription et Gestion de Profil

Validation de l'Inscription : Tester l'inscription avec différents ensembles de données (valides et invalides) pour vérifier la validation côté serveur et client.

Authentification et Sécurité : Essayer des tentatives de connexion avec des identifiants incorrects, tester la réinitialisation de mot de passe, et valider le processus d'authentification à deux facteurs si implémenté.

Recherche et Réservation de Trajets

Filtrage et Sélection de Trajet : Valider la fonctionnalité de recherche avec divers critères (date, lieu, préférences) et tester le processus de sélection et de réservation d'un trajet.

Annulation et Remboursement : Tester le processus d'annulation d'une réservation par un passager et par un conducteur, et vérifier la politique de remboursement.

Paielement et Transactions

Intégration de Passerelle de Paiement : Simuler des transactions de paiement réussies et échouées pour s'assurer que l'application gère correctement toutes les situations.

Sécurité des Transactions : Tester la sécurisation des transactions et la protection des données financières des utilisateurs.

Fonctionnalités Supplémentaires

Notification et Communication : Tester la fiabilité et la promptitude des notifications envoyées aux utilisateurs (par exemple, rappels de trajet, confirmations de réservation).

Assistance Utilisateur et Feedback : Évaluer les mécanismes de support client intégrés, y compris la soumission de feedback et la gestion des requêtes d'assistance.