

Plan de test



"Application mobile visant à faciliter l'exploration et la découverte du Maroc pour les touristes et les visiteurs."

Réalisé par :

OUGNI Imane

Encadré par :

Mme. Asmaa RETBI

Mme. Asmaa EL KASSIRI

M. Zohair ELMOURABIT

Nom de projet :

Travel Tracer

Projet :

Application mobile visant à faciliter l'exploration et la découverte du Maroc pour les touristes et les visiteurs.

Date :21/04/2024

Auteur : OUGNI Imane

Version : 1.1

Historique des mises à jour :

Version	Date	Auteur	Modifications
1.1	21/04/2024	OUGNI Imane	

Sommaire

- 1 Présentation du projet3
 - 1.1 Contexte.....3
 - 1.2 Approche méthodologique.....3
 - 1.3 Approche à tester.....3
 - 1.4 – Planning du projet.....3
- 2 Description de l’application à tester4
 - 2.1 Éléments à tester.....4
 - 2.2 Éléments à ne pas tester.....4
- 3 Définition de la stratégie de test4
 - 3.1 Parties prenantes.....4
 - 3.2 Niveaux de test.....5
 - 3.3 Priorisation des tests5
 - 3.4 Environnements de test6
 - 3.5 Outils de test.....6
 - 3.6 Référentiels de test.....7
 - 3.7 Jeux de données7
- 4 Définition des responsabilités8
 - 4.1 Matrice RACI8

1 Présentation du projet

1.1 Contexte

Dans un monde où les voyageurs cherchent des expériences de voyage authentiques et mémorables, l'utilisation d'applications mobiles devient de plus en plus essentielle. Naviguer dans des destinations inconnues peut être déroutant face à une multitude d'informations provenant de diverses sources. Les avancées technologiques, telles que la géolocalisation et les smartphones, ouvrent de nouvelles possibilités pour simplifier cette expérience. Les applications de voyage offrent aux voyageurs un accès facile à des informations précises et à jour, leur permettant de planifier efficacement leurs activités et de profiter au maximum de leur séjour. Cette convergence entre les attentes des voyageurs et les solutions technologiques crée un terrain propice à l'innovation dans le secteur du tourisme, offrant ainsi des expériences personnalisées et inoubliables.

1.2 Approche méthodologique

Dans le cadre de mon projet, j'adopte une approche Agile pour le développement de l'application mobile. Cela signifie que je travaille de manière collaborative, itérative et flexible, en mettant l'accent sur la livraison continue de fonctionnalités à valeur ajoutée pour les utilisateurs. L'utilisation de méthodologies Agile me permet d'ajuster rapidement mes priorités et mes efforts en fonction des retours d'utilisateur et des évolutions du marché, assurant ainsi une adaptation continue aux besoins changeants des utilisateurs.

1.3 Approche à tester

L'application à tester est destinée aux touristes et aux visiteurs dans le but de simplifier leurs voyages et de leur montrer les bonnes adresses cachées. Il s'agit d'une application mobile disponible sur les plateformes iOS et Android. Ses fonctionnalités principales incluent la création de compte, la recherche d'hébergements et de restaurants, ainsi que la visualisation de cartes en temps réel pour faciliter la navigation. Parmi les spécificités de l'application, nous devons accorder une attention particulière à sa compatibilité avec différents appareils et systèmes d'exploitation, ainsi qu'à l'utilisation de la géolocalisation pour assurer une expérience utilisateur optimale.

1.4— Planning du projet

Semaine 1-2 :

Mise en place de l'environnement de développement.

Création du squelette de l'application.

Implémentation de la fonctionnalité de création de compte et de connexion.

Intégration de la barre de recherche.

Planification des destinations et des saisons.

Mise en place du menu latéral et du menu inférieur.

Semaine 3-4 :

Finalisation de la fonctionnalité de création de compte et de connexion.

Implémentation de la recherche et du filtrage pour les hébergements et les restaurants.

Intégration de la fonctionnalité de planification par destination et par saison.

Développement des interfaces pour les menus latéral et inférieur.

Semaine 5-6 :

Ajout de la fonctionnalité de suggestions d'itinéraires.

Intégration des commentaires et évaluations des utilisateurs.

Finalisation des menus latéral et inférieur avec toutes les fonctionnalités nécessaires.

Semaine 7-8 :

Implémentation des fonctionnalités pour l'administrateur de l'application.

Développement des outils de modération des contenus et des utilisateurs.

Semaine 9-10 :

Intégration des fonctionnalités pour l'administrateur des restaurants et des hébergements.

Finalisation des tableaux de bord administratifs.

Semaine 11-12 :

Tests intensifs de toutes les fonctionnalités de l'application.

Correction des bugs et optimisation des performances.

Préparation de la version de production pour le déploiement.

Semaine 13 :

Déploiement de la version de production sur les plateformes iOS et Android.

Communication de la sortie de l'application aux utilisateurs.

2 Description de l'application à tester

2.1 Éléments à tester

L'application sera testée pour garantir le bon fonctionnement des fonctionnalités suivantes :

- Création de compte et connexion : Vérification de la possibilité de créer un compte utilisateur en fournissant les informations nécessaires et de se connecter avec succès à l'application.
- Recherches et filtrage : Test des fonctionnalités de recherche et de filtrage des hébergements et des restaurants, y compris l'efficacité des résultats retournés.
- Planification par destination et par saison : Validation de la capacité de l'application à fournir des suggestions de destinations, attractions et activités en fonction des préférences des utilisateurs.
- Suggestions d'itinéraires : Vérification de la pertinence et de l'utilité des itinéraires proposés pour faciliter la planification des voyages.
- Commentaires et évaluations : Test des fonctionnalités permettant aux utilisateurs de consulter, d'ajouter et de partager des commentaires et des évaluations sur les attractions touristiques, les restaurants et les hébergements.
- Menu latéral (hébergement, restaurants, transports, régions, shopping, événements) : Vérification de la navigation et de la convivialité de chaque menu latéral, ainsi que de la disponibilité des fonctionnalités associées.
- Menu inférieur (home, localisation, favoris, profile) : Test de la fonctionnalité et de la facilité d'utilisation de chaque élément du menu inférieur.

2.2 Éléments à ne pas tester

Les éléments suivants sont en dehors du périmètre de test de cette phase :

- Fonctionnalités administratives : Les fonctionnalités réservées aux administrateurs de l'application, des restaurants et des hébergements ne seront pas testées dans cette phase.
- Intégration de paiement : La fonctionnalité de paiement en ligne pour la réservation d'hébergements ne sera pas testée dans cette phase.
- Fonctionnalités spécifiques à certains appareils : Les fonctionnalités spécifiques à certains appareils ou systèmes d'exploitation seront exclues du périmètre de test initial.

3 Définition de la stratégie de test

3.1 Parties prenantes

Destinataires de l'application : Les utilisateurs finaux de l'application, à savoir les touristes et les visiteurs qui utiliseront l'application pour planifier et faciliter leurs voyages au Maroc.

Conception : L'équipe de conception est responsable de concevoir l'interface utilisateur (UI) et l'expérience utilisateur (UX) de l'application, en s'assurant qu'elle est intuitive, conviviale et répond aux besoins des utilisateurs.

Développement : L'équipe de développement est chargée de transformer les conceptions en fonctionnalités et de développer l'application mobile pour les plateformes iOS et Android, en utilisant les meilleures pratiques de développement.

Tests : L'équipe de test est responsable de tester toutes les fonctionnalités de l'application pour s'assurer qu'elle fonctionne comme prévu, qu'elle est exempte de bugs et qu'elle offre une expérience utilisateur optimale.

Validation de version : Avant la mise en production, la version de l'application est validée par l'équipe de développement, l'équipe de test et les parties prenantes concernées pour garantir qu'elle répond aux critères de qualité et aux attentes des utilisateurs.

3.2 Niveaux de test

Tests unitaires : Ces tests sont effectués au niveau du code pour vérifier le bon fonctionnement des unités individuelles ou des composants logiciels. Ils sont généralement réalisés par les développeurs eux-mêmes.

Tests d'intégration : Les tests d'intégration visent à vérifier la bonne interaction entre les différents modules ou composants de l'application une fois qu'ils sont assemblés. Ils sont également réalisés par les développeurs.

Tests de système : Les tests de système évaluent l'application dans son ensemble pour s'assurer qu'elle fonctionne conformément aux spécifications et aux attentes des utilisateurs. Ils sont généralement effectués par des testeurs dédiés.

Tests de régression : Les tests de régression sont effectués pour vérifier que les modifications apportées à l'application n'ont pas introduit de nouveaux bugs ou n'ont pas affecté négativement les fonctionnalités existantes. Ils sont également réalisés par des testeurs dédiés.

Tests d'acceptation : Les tests d'acceptation sont effectués pour valider que l'application répond aux exigences métier et aux attentes des utilisateurs. Ils sont généralement effectués par les parties prenantes, les utilisateurs finaux ou des testeurs représentant les utilisateurs finaux.

3.3 Priorisation des tests

Les tests sont priorisés en fonction de l'importance des fonctionnalités et de leur impact sur l'expérience utilisateur :

Must Have : Les fonctionnalités essentielles sont testées en priorité car ce sont celles sans lesquelles l'application ne peut pas fonctionner correctement.

Should Have : Les fonctionnalités importantes mais non critiques sont testées ensuite pour assurer une expérience utilisateur complète.

Could Have : Les fonctionnalités facultatives sont testées en dernier, car leur absence n'impacte pas significativement l'utilisation de l'application.

Won't Have : Certaines fonctionnalités peuvent être exclues du périmètre de test initial en raison de contraintes de temps ou de ressources. Elles peuvent être testées ultérieurement.

Voici la priorité des fonctionnalités adoptée dans mon projet :

Must Have :

Création de compte et connexion

Barre des recherches

Planification des destinations

Planification par saisons

Menu latéral (hébergement, restaurants, transports, régions, shopping, événements)

Menu inférieur (home, localisation, favoris, profile)

Administrateur de l'application

Administrateurs des restaurants

Administrateurs des hôtels

Commentaires et évaluations

Suggestions des itinéraires

Should Have :

Accès à un tableau de bord administratif pour l'administrateur de l'application, du restaurant et de l'hébergement

Gestion des utilisateurs, contenus et paramètres de l'application par l'administrateur de l'application

Modération des commentaires et évaluations par l'administrateur de l'application

Ajout, modification et suppression de contenu par l'administrateur de l'application

Gestion des informations et paramètres du restaurant par l'administrateur du restaurant

Gestion des informations et paramètres de l'établissement par l'administrateur de l'hébergement

Gestion des réservations par l'administrateur du restaurant et de l'hébergement

Could Have :

Toutes les fonctionnalités de filtrage

3.4 Environnements de test

Pour tester l'application, nous aurons besoin de :

Ordinateurs : Pour développer et exécuter les tests sur l'application.

Téléphones : Pour tester l'application sur différentes plateformes mobiles (iOS et Android).

Environnement de test (sandbox) : Un espace sécurisé où nous pouvons tester l'application sans risque pour les données réelles des utilisateurs.

Environnement de pré-production : Une copie de l'environnement de production où les tests finaux sont effectués avant le déploiement de l'application.

Environnement de production : L'environnement réel où l'application est déployée et accessible aux utilisateurs finaux.

3.5 Outils de test

Pour tester une application mobile dans ce contexte spécifique où l'accent est mis sur la création d'expériences de voyage authentiques et mémorables, Voici une sélection d'outils de test spécifiquement adaptés au contexte des applications mobiles axées sur les expériences de voyage authentiques et mémorables :

Test d'Interface Utilisateur (UI) et d'Expérience Utilisateur (UX) :

Marvel App : Pour créer des prototypes interactifs et tester l'UX.

InVision : Pour créer des prototypes et recueillir des commentaires sur l'UX.

UserTesting : Pour obtenir des retours d'utilisateurs réels sur l'interface et l'expérience utilisateur.

Test de Performance et de Compatibilité :

Firebase Test Lab : Pour tester la compatibilité de l'application sur une large gamme de dispositifs et de versions Android.

Xcode : Pour tester sur différents appareils iOS à l'aide du simulateur d'Apple.

BrowserStack : Pour tester la compatibilité sur une grande variété de navigateurs et de dispositifs.

Test de Fonctionnalités :

Appium : Pour effectuer des tests automatisés sur des applications mobiles Android et iOS.

Calabash : Un framework de test automatisé open source pour les applications mobiles.

Selenium : Pour automatiser les tests sur les navigateurs mobiles.

Test de Sécurité :

OWASP ZAP : Pour tester la sécurité de l'application mobile et détecter les vulnérabilités.

Mobile Security Framework (MobSF) : Pour tester la sécurité de l'application mobile sur les plateformes Android et iOS.

Test de Performance Réseau :

Charles Proxy : Pour analyser le trafic réseau entre l'application et le serveur, ce qui est crucial pour les applications mobiles qui dépendent fortement de la connectivité réseau.

Test de Localisation et de Géolocalisation :

Mock Location : Pour simuler des emplacements géographiques afin de tester les fonctionnalités de géolocalisation.

TestFlight (pour iOS) ou Google Play Console (pour Android) : Pour distribuer des versions bêta de l'application à des testeurs dans différentes régions et collecter des commentaires sur la localisation.

3.6 Référentiels de test

Les tests sont généralement stockés dans des référentiels de test, qui peuvent être physiquement situés dans différents endroits en fonction des besoins de l'organisation et des préférences de l'équipe de test. Voici quelques endroits courants où les tests peuvent être stockés :

Systèmes de gestion de versions (VCS) : Les tests peuvent être stockés dans des systèmes de gestion de versions tels que Git, SVN (Subversion), ou Mercurial. Cela permet de conserver une trace de l'historique des tests et de faciliter la collaboration entre les membres de l'équipe.

Référentiels de test dédiés : Certaines organisations utilisent des référentiels de test dédiés, qui sont des systèmes spécialement conçus pour stocker, organiser et gérer les tests. Ces référentiels peuvent être basés sur des outils de gestion des tests comme TestRail, HP ALM, ou Zephyr.

Systèmes de gestion de projets : Dans certaines équipes, les tests peuvent être intégrés aux systèmes de gestion de projets tels que Jira, Redmine, ou Trello. Cela permet de centraliser la gestion des tests avec d'autres activités de développement et de suivi de projet.

Systèmes de gestion des anomalies : Les tests peuvent également être liés aux systèmes de gestion des anomalies tels que Jira, Bugzilla, ou Mantis. Cela permet de suivre les tests associés à des anomalies spécifiques et de garantir que toutes les anomalies identifiées sont correctement testées et validées.

Systèmes de gestion de tests cloud : Certains services cloud offrent des fonctionnalités de stockage et de gestion des tests, permettant aux équipes de travailler sur des tests dans un environnement cloud sécurisé et accessible à partir de n'importe quel appareil connecté à Internet.

3.7 Jeux de données

Les jeux de données nécessaires pour tester une application de voyage peuvent varier en fonction des fonctionnalités spécifiques de l'application.

Données sur les destinations : Cela inclut des informations telles que les attractions touristiques, les restaurants, les hôtels, les activités, les transports en commun, etc. Ces données peuvent être collectées à partir de sources publiques ou privées, telles que des API de voyage, des bases de données gouvernementales, des sites web spécialisés, etc.

Données de localisation : Pour tester la fonction de géolocalisation de l'application, des jeux de données de localisation réels ou simulés peuvent être nécessaires. Cela peut inclure des coordonnées GPS de lieux touristiques, des itinéraires de transport en commun, des zones géographiques spécifiques, etc.

Données d'utilisateur : Pour simuler différents scénarios d'utilisation, des jeux de données d'utilisateurs peuvent être nécessaires, y compris des profils d'utilisateurs avec des préférences différentes, des historiques de recherche, des réservations passées, etc.

Les jeux de données peuvent être générés de différentes manières en fonction de leur type. Certains jeux de données peuvent être collectés à partir de sources existantes, tandis que d'autres peuvent nécessiter une création ou une simulation. Par exemple :

- Les données sur les destinations peuvent être collectées à partir de bases de données en ligne, de scrapers web ou d'API de services de voyage.
- Les données de localisation peuvent être générées en utilisant des coordonnées GPS réelles ou en simulant des itinéraires et des points d'intérêt.
- Les données d'utilisateur et de paiement peuvent être générées à l'aide de scripts ou d'outils de génération de données de test.
- Les données météorologiques peuvent être obtenues à partir de services météorologiques en ligne ou simulées à l'aide de modèles météorologiques.

Une fois générés, les jeux de données peuvent être stockés dans des bases de données, des fichiers plats ou d'autres formats adaptés aux besoins de l'application. Ils peuvent être hébergés localement ou sur des serveurs cloud en fonction des exigences en termes de disponibilité et de performance. Il est essentiel de garantir que les jeux de données de test sont gérés de manière sécurisée et conforme aux réglementations sur la protection des données, en particulier s'ils contiennent des informations sensibles.

4 Définition des responsabilités

4.1 Matrice RACI

Taches / Activités	Responsable (R)	Impliqué (A)	Consulté (C)	Informé (I)
Définition des exigences de l'application	Chef de projet	Équipe de développement, Équipe de design	Clients, Responsable marketing	Équipe de gestion de projet
Conception de l'interface utilisateur	Designer UX/UI	Équipe de développement	–	Équipe de gestion de projet
Développement des fonctionnalités principales	Développeurs	–	Design UX/UI, Analyse QA	Équipe de gestion de projet
Mise en place de la géolocalisation	Développeurs	–	Architecte logiciel	Équipe de gestion de projet
Intégration des API de voyage	Développeurs	–	Architecte logiciel	Équipe de gestion de projet
Création de la base de données des destinations	Développeurs	–	Architecte logiciel	Équipe de gestion de projet
Élaboration des cas de test	Analyste QA	–	Équipement de développement	Équipe de gestion de projet
Tests d'acceptation utilisateur	Équipe de test	Développeurs	Analyste QA	Équipe de gestion de projet
Correction des anomalies détectées	Développeurs	Équipe de test	Analyste QA	Équipe de gestion de projet
Rédaction de la documentation utilisateur	Rédacteur technique	–	Équipe de développement, équipe de test	Équipe de gestion de projet
Préparation du déploiement	Chef de projet	Administrateur système	Équipe de développement	Clients, Responsable marketing