

TILEBRIGHT APPS — DÉVELOPPEMENT
D'APPLICATION MOBILE

Cahier des charges

Une application de covoiturage native iOS et Android, développée et livrée avant la fenêtre de lancement de Ripple Ride à Austin.

Sommaire

01	Contrôle documentaire	2
02	Introduction et périmètre	3
03	Objet et public visé	3
04	Exigences fonctionnelles	3
05	Exigences non fonctionnelles	5
06	Modules du produit	5
07	Spécification des composants	6
08	Hypothèses, contraintes et dépendances	8
09	Phases de mise en production	8

EN BREF

Contrôle documentaire

SPÉCIFICATION

REQ-2026-0611

Émis le 16 mars 2026

PRÉPARÉ PAR

Tilebright Apps

Austin, TX 78701

PROPRIÉTAIRE

Priya Nair

Responsable des opérations,
Ripple Ride Carpooling

STATUT

Pour révision

Version de référence figée à
l'approbation

EXIGENCES

8

Fonctionnelles et non
fonctionnelles

01

Introduction et périmètre

Ripple Ride ouvre son marché d'Austin cet automne, au moment de la rentrée universitaire où la demande de covoiturage de l'UT Austin et des travailleurs de la tech atteint son pic. L'équipe a sécurisé un partenariat de recrutement de conducteurs et une date de lancement qui ne bougera pas — ce qui manque encore, c'est une application native capable d'absorber ce trafic dès le premier jour.

Objet et public visé

Ripple Ride a constitué son réseau de conducteurs et sécurisé un partenariat de lancement calé sur le semestre d'automne, mais le produit lui-même n'est encore qu'un ensemble de maquettes Figma et un backend prototype.

L'entreprise n'a qu'une seule chance de faire bonne impression sur un marché où les usagers ont déjà l'habitude par défaut des applications de covoiturage, et une application lente, buguée ou inachevée le jour du lancement anéantirait des mois de recrutement de conducteurs.

Tilebright Apps propose un engagement de seize semaines pour concevoir et développer des applications natives iOS et Android en parallèle d'un backend de mise en relation, afin que Ripple Ride se lance avec une application qui inspire autant de rapidité et de confiance que le réseau qui la sous-tend. Nous menons la conception, les deux développements natifs et l'assurance qualité comme une seule piste coordonnée plutôt qu'un relais, afin que rien n'attende plus longtemps que nécessaire. Les sections suivantes présentent l'approche, le périmètre, le calendrier et l'investissement.

Exigences fonctionnelles

ID	EXIGENCE	PRIORITÉ
----	----------	----------

R-01	L'application doit prendre en charge nativement les appareils iOS 16+ et Android 12+	OBLIGATOIRE
R-02	La mise en correspondance des trajets doit renvoyer des correspondances candidates en moins de 2 secondes	OBLIGATOIRE
R-03	Les profils du conducteur et du passager doivent afficher le statut de vérification d'identité en cours	OBLIGATOIRE
R-04	La messagerie intégrée doit être disponible entre le conducteur et le passager mis en relation	OBLIGATOIRE
R-05	L'application doit rester utilisable dans des conditions de réseau 3G dégradé, courantes dans les zones sans réseau du campus	SOUHAITABLE
R-06	Notifications push requises pour la confirmation, l'annulation et l'arrivée du trajet	OBLIGATOIRE
R-07	Les parcours principaux doivent réussir les tests de lecteur d'écran VoiceOver et TalkBack	SOUHAITABLE
R-08	Événements analytiques instrumentés pour les entonnoirs d'intégration et d'achèvement de trajet	SOUHAITABLE

ID	PRIORITÉ	EXIGENCE
R-01	Obligatoire	L'application doit prendre en charge nativement les appareils iOS 16+ et Android 12+
R-02	Obligatoire	La mise en correspondance des trajets doit renvoyer des correspondances candidates en moins de 2 secondes
R-03	Obligatoire	Les profils du conducteur et du passager doivent afficher le statut de vérification d'identité en cours
R-04	Obligatoire	La messagerie intégrée doit être disponible entre le conducteur et le passager mis en relation
R-05	Souhaitable	L'application doit rester utilisable dans des conditions de réseau 3G dégradé, courantes dans les zones sans réseau du campus

Design produit et UX

Cartographie des parcours, wireframes et système d'interface plat, validés par des sessions d'utilisabilité avec conducteurs et passagers.

01

Développement iOS

Développement natif Swift/SwiftUI de l'application complète, de l'accueil à la fin de course.

02

Développement Android

Développement natif Kotlin de l'application complète, réalisé en parallèle d'iOS à partir d'un système de design partagé.

Backend de mise en relation des trajets

Service de mise en relation basé sur l'itinéraire et l'horaire, avec mises à jour de statut en temps réel sur les deux applications.

Contrôle qualité et tests sur appareils

Tests interappareils et en conditions de réseau dégradé avant la publication de l'une ou l'autre application sur les stores.

Lancement et soumission aux stores

Fiches de store, dossiers de revue et accompagnement à la soumission pour l'App Store et Google Play.

SPÉCIFICATION

Spécification des composants

RÉF.	ÉLÉMENT	SPÉCIFICATION	UNITÉ	DÉLAI	PRIX UNITAIRE
SPC-01	Conception produit	Système de design plat, cartographie des parcours et validation d'utilisabilité	3	3 sem.	\$7,000 / wk
SPC-02	Développement iOS	Version native Swift/SwiftUI de l'ensemble de l'expérience passager et conducteur	8	8 sem.	\$9,000 / wk

SPC-03	Développement Android	Version native Kotlin de l'ensemble de l'expérience passager et conducteur	8	8 sem.	\$9,000 / wk
--------	--------------------------	---	---	--------	--------------

SPC-04	Backend et API de mise en relation des trajets	Service de mise en relation basé sur les itinéraires et les horaires, avec mises à jour de statut en temps réel	6	6 sem.	\$8,500 / wk
--------	---	--	---	--------	--------------

SPC-05	Contrôle qualité et tests sur appareils	Phase de contrôle qualité multi- appareils et en réseau dégradé avant soumission	2	2 sem.	\$6,000 / wk
--------	--	---	---	--------	--------------

SPC-06	Lancement et soumission aux stores	Fiches de store, captures d'écran et dossiers de soumission pour les deux plateforme s	1	2 septembre 2026	\$8,000
--------	---	---	---	------------------------	---------

Les spécifications sont fixées à l'acceptation ; tout changement est traité selon la procédure de contrôle des modifications définie dans le présent document.

Interface plate et lisible

Un système de design plat et sobre, à fort contraste de couleurs, conçu pour une utilisation à une main à un arrêt de bus ou sur le trottoir.

01

Performance native

Des bases de code Swift et Kotlin distinctes, et non une surcouche multiplateforme, pour que la mise en correspondance et les cartes restent rapides sur les téléphones plus anciens.

02

QA prête pour le lancement

Une phase de test dédiée sur appareils et réseaux avant soumission, pour que le premier examen App Store soit aussi le dernier.

03

01 SEMAINES 1 À 3 • DÉCOUVERTE ET CONCEPTION

Cartographie des parcours, wireframes et système de design plat, validés lors de sessions d'utilisabilité avec conducteurs et passagers.

02 SEMAINES 4 À 9 • DÉVELOPPEMENT PRINCIPAL

iOS, Android et le backend de mise en relation des trajets développés en parallèle à partir d'un système de design commun.

03 SEMAINES 10 À 12 • FONCTIONS DE CONFIANCE ET DE SÉCURITÉ

Statut de vérification, évaluations et messagerie intégrée superposés au flux de mise en relation principal.

04 SEMAINES 13 À 14 · CONTRÔLE QUALITÉ ET TESTS SUR APPAREILS

Tests multi-appareils et en réseau dégradé sur une matrice de téléphones récents et anciens.



Tilebright Apps

600 Congress Ave, Austin, TX 78701, États-Unis

build@tilebrightapps.com · +1 512 555 0164

· tilebrightapps.com

Tilebright Apps LLC · EIN 47-2810366