

Ride-Hailing MVP

Comparable to: Uber / Lyft

An on-demand ride-hailing MVP for one pilot city or metro area. It connects riders with admin-approved drivers for real-time trips.

Estimated MVP cost

\$19,450

Estimated effort

344 hours

To get your own customised cost for an app like this, visit www.chatstack.app and start a ChatStack interview describing how you want it customised.

Key Features

- Rider trip requests, driver acceptance, trip completion, and real-time GPS tracking
- Nearby-driver matching and base, time, and distance fare calculation
- Stripe card payments, receipts, and rider-to-driver ratings
- Driver applications with document uploads and admin approval
- Account onboarding, push notifications, and in-app account deletion

Out of Scope for This MVP

- Launch is limited to one US pilot city or metro area
- No surge pricing in the MVP
- US hosting, Stripe payments, and applicable privacy and transport compliance
- English-only experience with iOS and Android app store compliance
- Scheduled rides, multi-stop trips, pooling, promotions, tips, and advanced driver or rider features are out of scope for the MVP.

Full Cost Breakdown

Item	Hours	Cost
Visitor registration as rider or driver (email/phone auth, role-based access)	10	\$550
Driver application with document uploads	11	\$610
Admin review and approval of driver applications	13	\$720
Ride request with pickup and destination	12	\$660
Fare estimation before ride confirmation	8	\$440
Nearby-driver matching	11	\$610
Trip lifecycle (pickup, active trip, completion)	12	\$660
Live driver location and trip tracking	11	\$610
Stripe card payments	12	\$660
Receipts and fare detail	8	\$440
Ratings and feedback	6	\$330
Push notifications for ride events	11	\$610
Admin trip monitoring and operational support	12	\$660
In-app account and data deletion	11	\$610
Planning	38	\$2,090
Project Management	30	\$1,650
Testing	18	\$990
DevOps	15	\$830
Design & Frontend Build	45	\$2,480
AI Tokens	0	\$490
Security	18	\$990
Risk (10% Contingency)	32	\$1,760

Estimate Assump
tions

- Estimates cover imple
mentation
and feature-
level
integration,
excluding
NFR
overhead,
project man
agement,
design
discovery,
and third-
party
provider
fees.
- Rider
and driver e
xperiences
share
backend
services
but have
separate
mobile
navigation
and role-
specific
screens.
- Mapping,
geocoding,
push
notification,
Stripe, and
document-
storage
provider
accounts
and
credentials
are
available
when devel
opment
begins.

- Basic UI designs, fare rules, document types, cancellation rules, and pilot-market eligibility rules are supplied before implementation.
- Receipt delivery is in-app only for the base estimate; email delivery is excluded.
- Active location tracking uses foreground or active-trip behavior rather than unrestricted background tracking.
- Account deletion is implemented with configurable legal-retention exclusions and does not require manual review workflows.

Risks That Could Change This

- Mapping and location-provider behavior, quotas, accuracy, and platform permission changes may increase ride-request and tracking effort.
- Stripe webhook failures, payment disputes, refunds, and reconciliation edge cases may require additional implementation and testing.
- Push notification delivery and permission denial can create operational inconsistencies requiring fallback and recovery work.
- Driver-document privacy, retention, and compliance requirements may require stronger review workflows or storage controls.
- App-store review or transportation-service compliance findings may delay release and require changes to onboarding, payments, or deletion flows.

User Stories

US-1 - Visitor

As a visitor, I want to create a rider or driver account so that I can use the appropriate ride-hailing services.

US-2 - Logged In Rider

As a logged-in rider, I want to request a ride by entering my pickup and destination so that I can arrange transportation.

US-3 - Logged In Rider

As a logged-in rider, I want to see the estimated fare before requesting a ride so that I can make an informed decision.

US-4 - Logged In Driver

As a logged-in driver, I want to see and accept nearby available ride requests so that I can serve riders and earn fares.

US-5 - Logged In Rider

As a logged-in rider, I want to view my driver's position and trip progress so that I know where my ride is.

US-6 - Logged In Driver

As a logged-in driver, I want to update a ride through pickup, active trip, and completion so that the platform maintains accurate trip status.

US-7 - Logged In Rider

As a logged-in rider, I want to pay for a completed ride with a saved card so that I can complete the trip without cash.

US-8 - Logged In Rider

As a logged-in rider, I want to receive a receipt and review fare details so that I can understand what I was charged.

US-9 - Logged In Rider

As a logged-in rider, I want to rate my completed ride and driver so that I can provide feedback about the service.

US-10 - Visitor

As a visitor, I want to apply as a driver and upload required documents so that I can be considered for approval.

US-11 - Platform Administrator

As a platform administrator, I want to review driver applications and approve or reject applicants so that only eligible drivers operate on the platform.

US-12 - Platform Administrator

As a platform administrator, I want to monitor active trips and support riders and drivers so that I can resolve

operational issues.

US-13 - Logged In App User

As a logged-in app user, I want to receive trip-related notifications so that I stay informed about important ride updates.

US-14 - Logged In App User

As a logged-in app user, I want to delete my account from within the app so that I can stop using the service and request removal of associated data.

Requirements

FR-1 - Account Management

The mobile application shall allow visitors to register as riders or drivers using email or phone-based authentication and access role-appropriate functionality.

FR-2 - Driver Onboarding

The application shall allow visitors to submit a driver application and upload required eligibility documents for the pilot market.

FR-3 - Driver Administration

The web administration console shall allow authorized administrators to review driver applications, access submitted documents, and approve or reject applicants.

FR-4 - Ride Request

The rider application shall allow logged-in riders to enter pickup and destination locations and request an immediate ride.

FR-5 - Fare Estimation

The platform shall calculate and display an estimated fare before a rider confirms a ride request using configured base, time, and distance rates.

FR-6 - Ride Matching

The platform shall identify nearby available approved drivers and make eligible ride requests available to them based on proximity and availability.

FR-7 - Trip Lifecycle

The driver application shall allow an accepted ride to progress through pickup, active trip, and completion states, with status updates persisted by the platform.

FR-8 - Location Tracking

The rider application shall display the driver's position and trip progress during an active ride or approach to pickup.

FR-9 - Payments

The platform shall allow riders to pay for completed rides using a saved card processed through Stripe.

FR-10 - Receipts and Fare Details

The platform shall provide riders with a receipt showing the fare calculation, final charge, and applicable cancellation or refund information.

FR-11 - Ratings and Feedback

The rider application shall allow riders to rate a completed ride and driver, and the platform shall associate the rating with the completed trip.

FR-12 - Notifications

The platform shall send logged-in riders and drivers notifications for operational ride events, including request status, driver arrival, trip changes, and completion.

FR-13 - Operations and Support

The administration console shall allow authorized administrators to monitor active trips and perform basic operational support for riders and drivers.

FR-14 - Account and Data Deletion

The mobile application shall allow logged-in users to delete their account and request deletion of associated personal data from within the app.

NFR-1 - Security and Privacy

The platform shall protect account, payment, location, rating, and driver-document data using appropriate authentication, authorization, encryption, and secure storage controls.

NFR-2 - Privacy and Consent

The mobile applications shall clearly disclose location use, obtain required consent, and limit collection and retention to data necessary for the service.

NFR-3 - Performance

Platform APIs shall target response times below 200 milliseconds for standard operations under expected MVP load.

NFR-4 - Scalability

The backend data layer shall use a managed Supabase database configuration capable of supporting growth beyond the initial pilot city.

NFR-5 - Observability

The mobile applications shall use Firebase Performance Monitoring, and the platform shall provide sufficient logging and monitoring to diagnose ride, payment, and notification failures.

NFR-6 - Reliability and Data Integrity

The platform shall maintain consistent trip, payment, driver availability, and notification states when requests are retried, interrupted, or processed concurrently.

NFR-7 - Availability and Recovery

The MVP shall provide operational recovery procedures for service outages, failed payments, lost location updates, and interrupted trip-state updates.

NFR-8 - Platform Compliance

The iOS and Android applications shall comply with applicable app-store, privacy, payment, and data-deletion requirements for a real-world transportation service.

NFR-9 - Usability and Accessibility

Core onboarding, ride request, fare review, payment, trip tracking, and account deletion flows shall be usable on supported iOS and Android devices with clear error handling.

NFR-10 - Auditability

The administration console shall record auditable changes to driver approval status, trip status, payment status, and account or data-deletion actions.

Requirements Assumptions

- The MVP serves one pilot US city or metro area.
- The mobile solution includes separate rider and driver experiences, while administration is provided through a web console.
- No integrations are assumed beyond Stripe, mapping or geolocation services, push notifications, Supabase, and Firebase Performance Monitoring.
- Stripe payment processing covers card tokenization, authorization, capture, refunds, and payment-status callbacks.
- A mapping and geocoding provider is required for pickup, destination, routing, distance, and driver-location functionality.
- Administrators, drivers, and riders use role-based access controls with separate permissions.
- Driver documents are securely stored and accessible only to authorized administrators and relevant compliance processes.
- Fare rates, cancellation rules, supported document types, and pilot-market eligibility rules are configurable without code changes where practical.
- The platform supports immediate rides only; scheduled rides, pooling, multi-stop trips, surge pricing, promotions, tips, and cash payments are excluded.
- Legal retention periods, exact availability targets, accessibility conformance, and detailed operational SLAs remain implementation decisions.

Technical Specification

TS-1 - Mobile Applications

Build separate rider and driver mobile experiences for iOS and Android using Flutter and Dart, with Riverpod state management and Material Design 3.

TS-2 - Authentication

Use Supabase Auth with email or phone-based registration and JWT-based session management for rider and driver accounts.

TS-3 - Authorization and RBAC

Implement role-based access control for riders, drivers, and administrators across mobile applications, REST APIs, database policies, and the admin console.

TS-4 - Backend and APIs

Implement backend business logic using Supabase-managed services, PostgreSQL, REST APIs, and optional edge functions for asynchronous or provider-integrated operations.

TS-5 - Data Storage

Use a managed Supabase PostgreSQL database for users, roles, driver applications, documents, rides, locations, fares, payments, ratings, notifications, and audit records.

TS-6 - Payments

Integrate Stripe for saved-card payment processing, authorization, capture, refunds, payment callbacks, and payment-status reconciliation.

TS-7 - Location and Ride Operations

Integrate a mapping and geolocation provider for address search, geocoding, routing, distance calculation, driver location, proximity matching, and active-trip tracking.

TS-8 - Administration Web Console

Build the administrator console with Next.js, React, TypeScript, Shadcn UI, and Tailwind CSS, hosted on Vercel.

TS-9 - Notifications

Use OneSignal and Firebase Cloud Messaging for operational push notifications, with in-app notification states and fallback messaging when push permission is unavailable.

TS-10 - Hosting and Environments

Host the backend and database on managed Supabase services, the administration console on Vercel, and maintain separate development, staging, and production environments.

TS-11 - CI/CD and Release Management

Use GitHub Actions for backend, database, web, quality, and deployment pipelines, and CodeMagic for iOS and Android build and store-release automation.

TS-12 - Security and Privacy

Protect account, payment, location, rating, and driver-document data with TLS, encryption at rest, secure secrets management, tokenization, row-level security, and least-privilege access.

TS-13 - Observability and Support

Implement structured application logging, error monitoring, audit logging, operational dashboards, and alerts for ride, payment, location, notification, and trip-state failures.

TS-14 - Testing and Quality

Provide unit, integration, API, database-policy, mobile, web, and end-to-end tests covering authentication, ride lifecycle, payments, notifications, deletion, and administrative workflows.

TS-15 - Reliability and Data Integrity

Design critical ride, payment, availability, notification, and deletion operations to be idempotent and transactionally consistent under retries, concurrency, and interrupted processing.

TS-16 - Compliance and App-Store Readiness

Implement app-store, privacy, payment, transportation-service, and data-deletion requirements applicable to the pilot US market and selected service providers.

Technical Assumptions

- The MVP serves one pilot US city or metro area.
- Supabase provides managed PostgreSQL, authentication, row-level security, and backend hosting.
- The reference technology stack is used: Flutter, Next.js, Supabase, Vercel, GitHub Actions, CodeMagic, OneSignal, Firebase Cloud Messaging, and Stripe.
- A mapping and geolocation provider will be selected during implementation; provider costs and quotas are excluded from this specification.
- Development, staging, and production environments are maintained with isolated data, credentials, and provider configurations.
- Fare rates, cancellation rules, eligibility rules, supported document types, and retention policies are configurable without code changes where practical.
- Scheduled rides, pooling, multi-stop trips, surge pricing, promotions, tips, cash payments, and multi-city operations are excluded from the MVP.
- Legal retention periods, data residency requirements, exact availability targets, accessibility conformance, and detailed recovery objectives remain implementation decisions.
- Email and SMS delivery are optional integrations using SendGrid and Twilio if required by final product decisions.

To get your own customised cost for an app like this, visit <https://www.chatstack.app> and start a ChatStack interview.

This is a starting estimate for a custom-built MVP, designed and built from scratch. You own all the code and IP with no platform lock-in. Uber / Lyft are trademarks of their respective owners; ChatStack / App Developer Studio is not affiliated with, endorsed by, or sponsored by Uber / Lyft. © 2026 ChatStack.

Created with ChatStack (www.chatstack.app)