

Food Delivery MVP

Comparable to: DoorDash / Uber Eats

A lean on-demand food delivery marketplace for one city or tightly defined service area, inspired by DoorDash and Uber Eats. Connects customers, restaurants, and platform-managed independent delivery drivers.

Estimated MVP cost

\$24,750

Estimated effort

437 hours

Key Features

- Restaurant discovery, menu browsing, cart, checkout, and ASAP or scheduled delivery orders.
- Stripe payment processing, order-status tracking, and customer ratings.
- Restaurant menu and order management.
- Driver delivery offers, navigation handoff, and delivery-status updates.
- Administrator oversight of restaurants, drivers, orders, and marketplace operations.

Out of Scope for This MVP

- Initial launch is limited to one US city or tightly defined service area.
- Delivery is fulfilled only by platform-managed drivers, not restaurant self-delivery.
- US hosting, payment processing, and applicable regulatory compliance are required.
- Lean MVP scope with simple industry-standard defaults.
- Pickup is not included in the initial MVP.
- Multi-city launch and restaurant self-delivery may be considered after MVP validation.

Full Cost Breakdown

Item	Hours	Cost
The customer experience shall allow visitors to discover restaurants within the configured delivery service area using search and basic filters.	5	\$280
The customer experience shall display restaurant menus with item names, descriptions, prices, availability, and supported modifiers.	6	\$330
The platform shall support authenticated customer, restaurant staff, driver, and administrator accounts with role-based access.	8	\$440
Logged-in customers shall add, remove, modify quantities, and configure menu items in a cart containing items from one restaurant only.	9	\$500
Logged-in customers shall provide or select a delivery address, review order details and fees, and choose ASAP or predefined scheduled delivery windows.	11	\$610
The platform shall securely authorize and capture card payments for food orders, including applicable restaurant commissions, delivery fees, and service fees.	12	\$660
The platform shall create an order after successful checkout and provide order confirmation and persistent order details to the customer.	8	\$440
	14	\$770

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. DoorDash / Uber Eats are trademarks of their respective owners; App Developer Studio is not affiliated with, endorsed by, or sponsored by DoorDash / Uber Eats. © 2026 App Developer Studio.

Created by App Developer Studio

Restaurant staff shall create and edit menu items, prices, descriptions, modifiers, availability, operating hours, and temporary item unavailability through web tools.

Restaurant staff shall receive incoming delivery orders, accept or decline them, view fulfillment details, and update preparation status.	11	\$610
Drivers shall receive delivery offers containing pickup, drop-off, estimated distance, and expected compensation, and shall accept available offers.	10	\$550
Drivers shall update delivery progress through pickup and drop-off, use an external mapping app for navigation, and share location during relevant delivery operations.	16	\$880
Customers shall view order status covering accepted, preparing, ready for pickup, driver assigned, picked up, and delivered.	9	\$500
The platform shall send simple push notifications for material order-status changes and relevant delivery events.	7	\$390
Customers shall rate completed orders and delivery experiences, and the platform shall support basic reporting and moderation of submitted content.	11	\$610
Administrators shall manage accounts, oversee restaurants and drivers, monitor orders and payments, view operational statuses, and handle basic issues.	16	\$880
Customers and drivers shall manage account information and submit account-deletion requests in the mobile apps; a web deletion path shall also be available.	10	\$550
The platform shall publish support contact information and provide applicable privacy, reporting, blocking, moderation, and data-handling controls.	10	\$550
The platform shall provide iOS and Android release builds, review instructions, and test access covering core customer, restaurant, driver, location, payment, rating, and deletion flows.	15	\$830
Planning	48	\$2,640
Project Management	38	\$2,090
Testing	23	\$1,270
DevOps	19	\$1,050
Design & Frontend Build	57	\$3,140
AI Tokens	0	\$630
Security	23	\$1,270
Risk (10% Contingency)	41	\$2,260
Total	437	\$24,750

Estimate Assumptions

- One Flutter codebase serves customer and driver roles with role-specific navigation.
- Supabase Auth, PostgreSQL, Stripe, push notifications, Vercel, GitHub Actions, and CodeMagic are used as specified.
- Email/password authentication is sufficient; social login, SSO, and multifactor authentication are excluded.
- One restaurant per cart and order is enforced, with no pickup, cash, promotions, loyalty, or multi-restaurant ordering.
- Scheduled delivery uses predefined time windows and navigation is handed off to an external mapping app.
- Restaurant and administrator tools are web-based; customer and driver experiences are mobile applications.
-

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. DoorDash / Uber Eats are trademarks of their respective owners; App Developer Studio is not affiliated with, endorsed by, or sponsored by DoorDash / Uber Eats. © 2026 App Developer Studio.

Created by [App Developer Studio](http://www.appdeveloperstudio.co.za)

Email and SMS notifications, custom routing, dispatch optimization, accounting, tax, and logistics integrations are excluded.

- Legal, privacy, labor, insurance, accessibility, and app-store policy decisions are supplied or approved by the product owner.

Risks That Could Change This

- Payment, push, mapping, and location-provider behavior may require additional implementation for edge cases and provider failures.
- Marketplace order-state coordination across restaurants, drivers, customers, and administrators may expand integration and reconciliation effort.
- App-store review requirements and privacy or location policies may change before release.
- Operational rules for driver assignment, compensation, cancellations, refunds, and restaurant declines are not fully defined.
- Address validation, delivery fees, serviceability boundaries, and scheduled-slot capacity rules may require additional business logic.

User Stories

US-1 - Visitor

As a visitor, I want to discover restaurants available in my delivery area so that I can choose where to order.

US-2 - Visitor

As a visitor, I want to browse restaurant menus and item details so that I can decide what to order.

US-3 - Logged In Customer

As a logged-in customer, I want to add menu items to a cart and review my order so that I can confirm my selections before checkout.

US-4 - Logged In Customer

As a logged-in customer, I want to place an ASAP or scheduled delivery order so that I can receive food when needed.

US-5 - Logged In Customer

As a logged-in customer, I want to pay securely for my order so that the restaurant and delivery operation can process it.

US-6 - Logged In Customer

As a logged-in customer, I want to track my order status so that I know when the restaurant is preparing it and when the driver is delivering it.

US-7 - Logged In Customer

As a logged-in customer, I want to rate my completed order and delivery experience so that I can provide feedback and help maintain marketplace quality.

US-8 - Restaurant Staff

As restaurant staff, I want to manage my menu so that customers see accurate items, prices, options, and availability.

US-9 - Restaurant Staff

As restaurant staff, I want to receive and update incoming delivery orders so that food is prepared and handed to the assigned driver on time.

US-10 - Logged In Driver

As a logged-in driver, I want to receive and accept delivery offers so that I can choose available deliveries to complete.

US-11 - Logged In Driver

As a logged-in driver, I want to update delivery progress and share my location when needed so that customers and administrators can follow fulfillment.

US-12 - Platform Administrator

As a platform administrator, I want to oversee restaurants, drivers, orders, payments, and marketplace activity so that I can operate the delivery service reliably.

US-13 - Logged In App User

As a logged-in app user, I want to manage my account and request account deletion so that I can control my personal information.

US-14 - Platform Administrator

As a platform administrator, I want marketplace users to have privacy, support, and content-reporting controls so that the service can handle user-generated content responsibly.

US-15 - Platform Administrator

As a platform administrator, I want the mobile apps and marketplace flows prepared for app-store review so that customers and drivers can access the MVP through approved iOS and Android releases.

Requirements

FR-1 - Restaurant Discovery

The customer experience shall allow visitors to discover restaurants within the configured delivery service area using search and basic filters.

FR-2 - Menu Browsing

The customer experience shall display restaurant menus with item names, descriptions, prices, availability, and supported modifiers.

FR-3 - Identity and Access

The platform shall support authenticated customer, restaurant staff, driver, and administrator accounts with role-based access.

FR-4 - Cart

Logged-in customers shall add, remove, modify quantities, and configure menu items in a cart containing items from one restaurant only.

FR-5 - Checkout

Logged-in customers shall provide or select a delivery address, review order details and fees, and choose ASAP or predefined scheduled delivery windows.

FR-6 - Payments

The platform shall securely authorize and capture card payments for food orders, including applicable restaurant commissions, delivery fees, and service fees.

FR-7 - Order Management

The platform shall create an order after successful checkout and provide order confirmation and persistent order details to the customer.

FR-8 - Restaurant Operations

Restaurant staff shall create and edit menu items, prices, descriptions, modifiers, availability, operating hours, and temporary item unavailability through web tools.

FR-9 - Restaurant Operations

Restaurant staff shall receive incoming delivery orders, accept or decline them, view fulfillment details, and update preparation status.

FR-10 - Driver Operations

Drivers shall receive delivery offers containing pickup, drop-off, estimated distance, and expected compensation, and shall accept available offers.

FR-11 - Delivery Fulfillment

Drivers shall update delivery progress through pickup and drop-off, use an external mapping app for navigation, and share location during relevant delivery operations.

FR-12 - Order Tracking

Customers shall view order status covering accepted, preparing, ready for pickup, driver assigned, picked up, and delivered.

FR-13 - Notifications

The platform shall send simple push notifications for material order-status changes and relevant delivery events.

FR-14 - Ratings and Content

Customers shall rate completed orders and delivery experiences, and the platform shall support basic reporting and moderation of submitted content.

FR-15 - Administration

Administrators shall manage accounts, oversee restaurants and drivers, monitor orders and payments, view operational statuses, and handle basic issues.

FR-16 - Account and Privacy

Customers and drivers shall manage account information and submit account-deletion requests in the mobile apps; a web deletion path shall also be available.

FR-17 - Support and Trust

The platform shall publish support contact information and provide applicable privacy, reporting, blocking, moderation, and data-handling controls.

FR-18 - Release Readiness

The platform shall provide iOS and Android release builds, review instructions, and test access covering core customer, restaurant, driver, location, payment, rating, and deletion flows.

NFR-1 - Performance

The platform API shall target response times below 200 milliseconds for normal read and transactional requests, excluding third-party provider latency.

NFR-2 - Scalability

The backend shall use a managed database architecture capable of scaling with the initial marketplace, with Supabase managed DB as the default assumption.

NFR-3 - Observability

The platform shall provide application performance monitoring, error visibility, and operational logging using Firebase Performance Monitoring or an equivalent service.

NFR-4 - Security

The platform shall enforce role-based authorization, secure authentication, encrypted data transport, secure secrets management, and payment-token handling through the payment provider.

NFR-5 - Privacy

The platform shall minimize collection of personal and location data, disclose collection and use accurately, restrict location access to necessary operations, and support deletion workflows.

NFR-6 - Reliability

Order, payment, and delivery state transitions shall be durable, auditable, and resilient to retries, duplicate requests, and temporary third-party failures.

NFR-7 - Availability

Core browsing, ordering, restaurant handling, driver fulfillment, and administration functions shall be designed for dependable operation during configured service hours.

NFR-8 - Compatibility

The customer and driver experiences shall support approved iOS and Android releases, while restaurant and administration tools shall support current major desktop browsers.

NFR-9 - Usability and Accessibility

The MVP shall provide a clear English-only experience with consistent navigation, readable content, understandable errors, and standard mobile accessibility practices.

NFR-10 - Compliance

The mobile apps and marketplace flows shall comply with applicable Apple App Store and Google Play requirements, including physical-goods payment, privacy, deletion, location, and review-access rules.

Requirements Assumptions

- The MVP launches in one US city or tightly defined delivery service area.
- Pickup orders, multi-restaurant carts, subscriptions, loyalty, promotions, gift cards, and cash payments are out of scope unless added later.
- Delivery is fulfilled only by platform-managed independent drivers; restaurant self-delivery is excluded.
- Stripe or an equivalent card-payment provider will handle payment processing and tokenization.
- A single restaurant per order is enforced.
- Scheduled delivery uses predefined time windows rather than arbitrary customer-selected times.
- Navigation is handed off to a standard external mapping application.
- Push notifications are supported on mobile devices where users grant notification permission.
- Restaurant and administrator tools are web-based; customer and driver experiences are mobile apps.
- English is the only supported language for the MVP.
- A managed backend and database are preferred; Supabase managed DB is assumed for estimation.
- No integrations are assumed beyond payment processing, push notifications, mapping handoff, app stores, and monitoring unless specified.
- Legal, tax, insurance, labor, accessibility, privacy, and marketplace compliance decisions will be confirmed by the product owner and counsel.

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. DoorDash / Uber Eats are trademarks of their respective owners; App Developer Studio is not affiliated with, endorsed by, or sponsored by DoorDash / Uber Eats. © 2026 App Developer Studio.

Created by [App Developer Studio](#)

- Legally required financial, fulfillment, moderation, and audit records may be retained after account deletion.
- Formal service-level objectives, peak volumes, retention periods, and minimum device/OS versions are not yet defined and require estimation assumptions.

Technical Specification

TS-1 - Mobile Applications

Build customer and driver mobile apps with Flutter and Dart, using Riverpod for state management and Material Design 3 for the shared UI system.

TS-2 - Web Applications

Build restaurant and administrator tools as responsive Next.js applications using React, TypeScript, Tailwind CSS, and shadcn/ui.

TS-3 - Backend and APIs

Use a managed backend with PostgreSQL on Supabase, REST APIs, and optional edge functions for transactional workflows and provider callbacks.

TS-4 - Authentication

Use Supabase Auth with JWT-based sessions for customer, restaurant staff, driver, and administrator accounts.

TS-5 - Authorization and RBAC

Enforce role-based access at API and database layers, with separate permissions for customers, restaurant staff, drivers, and administrators.

TS-6 - Data and Order State

Model users, restaurants, menus, carts, orders, payments, deliveries, status events, ratings, reports, and deletion requests in PostgreSQL.

TS-7 - Payments

Integrate Stripe for card authorization and capture, payment tokenization, delivery and service fees, restaurant commissions, refunds, and payment status reconciliation.

TS-8 - Hosting and Infrastructure

Host web and administration applications on Vercel and use Supabase-managed PostgreSQL and backend services for the initial single-city deployment.

TS-9 - Environments and CI/CD

Maintain separate development, staging, and production environments with environment-specific secrets, database configurations, and provider credentials.

TS-10 - Notifications and Integrations

Use OneSignal or Firebase Cloud Messaging for push notifications, Stripe for payments, and external mapping applications for driver navigation.

TS-11 - Security and Privacy

Use HTTPS with TLS 1.2 or higher, encrypted data in transit and at rest, managed secrets, JWT validation, least-privilege access, and privacy-aware data retention.

TS-12 - Observability

Provide centralized application logs, error monitoring, performance monitoring, audit events, and operational dashboards across mobile, web, backend, payment, and delivery workflows.

TS-13 - Testing

Implement unit, integration, API, database, and end-to-end tests covering checkout, order state transitions, restaurant handling, driver fulfillment, permissions, deletion, and provider failures.

TS-14 - Release Readiness

Produce signed iOS and Android builds, store metadata, review instructions, test accounts, and test data covering payment, location, scheduled orders, ratings, and deletion flows.

Technical Assumptions

- The MVP launches in one US city or tightly defined service area with a Supabase region selected for US data residency.
- Supabase Auth, PostgreSQL, Vercel, GitHub Actions, and CodeMagic are the default platform choices for estimation.
- Email/password authentication is sufficient for the MVP; social authentication, SSO, and multifactor authentication are not included unless required.

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. DoorDash / Uber Eats are trademarks of their respective owners; App Developer Studio is not affiliated with, endorsed by, or sponsored by DoorDash / Uber Eats. © 2026 App Developer Studio.

Created by App Developer Studio

Stripe is the sole payment provider for the MVP; RevenueCat and in-app purchases are excluded because orders are physical goods.

- Development, staging, and production environments are required, with separate credentials and controlled production access.
- One mobile codebase serves customer and driver roles, with role-specific navigation and permissions.
- Restaurant and administrator applications are web-based; customer and driver experiences are mobile applications.
- No custom mapping, routing, dispatch optimization, accounting, tax, or logistics integrations are included beyond stated provider integrations.
- Push notifications use OneSignal or Firebase Cloud Messaging; email and SMS are excluded from the core MVP unless explicitly implemented.
- English is the only supported language, and advanced localization is excluded.
- Formal uptime targets, peak volumes, retention periods, and minimum device or OS versions remain estimation assumptions.
- Legal, tax, insurance, labor, accessibility, privacy, and marketplace compliance decisions will be confirmed by the product owner and counsel.