



FAST RIDER
Delivery Services

One Call, Few Seconds at Your Doorstep

PROJECT PROGRESS & TECHNICAL BLUEPRINT REPORT

Fast Rider Delivery Services

UI/UX Simulation Phase, Approved Architecture, and Next Development Stage

Prepared by Cypher Africa Group
10 June 2026

Current Stage	Overall Status	Next Phase
UI/UX & Workflow Simulation	Strong progress / client-approved direction	Backend foundation & integrations

1. Project Overview

Fast Rider Delivery Services is a multi-service transport and logistics platform designed to connect clients, boda boda riders, recipients, and platform administrators through one coordinated digital system. The platform supports parcel delivery, food delivery, rider errands, passenger ride requests, payment collection, rider assignment, live tracking, recipient confirmation, and operational oversight.

2. Approved Platform Modules

Client Application: For account registration, phone OTP verification, service booking, location selection, M-Pesa payment, tracking, history, and profile management.

Rider Application: For rider onboarding, document submission, capability and equipment verification, online status, matched job requests, navigation, proof of delivery, earnings, and history.

Recipient Tracking: A lightweight secure web experience accessed through a delivery link or tracking details without requiring full app installation.

Admin Dashboard: For dispatch, rider approvals, delivery management, client management, payments, pricing, reports, disputes, and support.

Demo Portal: A landing page linking the client, rider, recipient, and admin simulations for review and approval.

3. Completed Client Application Simulation

- Client account creation and personal/business account selection.
- Phone number verification using a simulated six-digit OTP flow.
- Send Package workflow for parcels, documents, shopping, food, and fragile items.
- Send a Rider workflow for collection, purchase, delivery, or errand requests.
- Request Rider workflow for passenger boda transport.
- Uber-style map picker for pickup, first stop, destination, and passenger locations.
- Package category selection and equipment-aware food delivery requirements.
- M-Pesa STK Push payment simulation before rider assignment.
- Rider search, rider assignment, delivery tracking, completion, history, and profile management.
- Official Fast Rider Delivery Services logo and red/blue/white brand theme.

4. Completed Rider Application Simulation

- Rider phone/account signup, personal details, bike details, and document submission.
- National ID, driving licence, insurance, and motorcycle information workflow.
- Rider capability registration for parcels, errands, food, fragile items, and passenger rides.
- Equipment recording for insulated food carriers, passenger helmets, rain covers, and load limits.
- Admin verification status and approval simulation.
- Online/offline availability, incoming requests, job acceptance and decline.
- Navigation to pickup, pickup-code verification, in-transit workflow, proof of delivery, and job completion.
- Rider earnings, history, wallet, and profile screens.

- Filtering of food or passenger requests according to verified rider equipment.

5. Completed Recipient Tracking Simulation

- Secure delivery lookup using delivery reference and recipient phone number.
- Live delivery map simulation with rider movement and estimated arrival time.
- Rider name, motorcycle details, package details, and progress timeline.
- Call and message rider actions within the tracking interface.
- Recipient delivery confirmation code and successful receipt confirmation.
- Issue reporting, delivery completion, and rider rating.

6. Completed Admin Dashboard Simulation

- Operations overview with active deliveries, online riders, revenue, and pending approvals.
- Delivery management for parcel, food, errand, and passenger requests.
- Live dispatch map with active riders and unassigned requests.
- Rider application review, document verification, capability approval, rejection, and suspension concepts.
- Client and business account management.
- M-Pesa collections, rider payouts, platform commissions, refunds, and finance reporting.
- Pricing rules, base fares, per-kilometre rates, food surcharges, waiting charges, and commission settings.
- Disputes, damaged package cases, passenger no-shows, late deliveries, and support workflows.

7. Key Business Rules Captured

Rule	Approved Direction
Food requests	Only riders with approved insulated, warmer, or cooler carriers should receive eligible food requests.
Passenger rides	Only approved riders with a verified passenger helmet should receive passenger bookings.
Recipient access	Recipients should access tracking through a secure link or tracking reference without requiring a full account.
Payment before matching	Rider matching should begin only after KopoKopo confirms a successful M-Pesa payment callback.
Rider privacy	A future call-masking or call-bridging service should hide personal rider and client numbers during an active trip.
Proof of delivery	Delivery completion should use recipient OTP, signature, photograph, or another approved verification method.
Equipment-based matching	Rider requests must be filtered by service capabilities, location, equipment, availability, and account approval status.

8. Agreed Technology Stack

Frontend: Mobile-first Progressive Web App for clients and riders; responsive web portal for recipients and administrators.

Backend: PHP with dedicated API endpoints and structured service classes.

Database: MySQL using PDO for secure database operations.

Authentication: Phone-based SMS OTP login and verification for clients and riders.

SMS Provider: Africa's Talking API for OTP and operational SMS notifications.

Payments: KopoKopo API for M-Pesa STK Push, callbacks, payment status, and reconciliation.

Maps: Google Maps or Mapbox for geocoding, route distance, navigation, and live tracking.

Deployment: Hosted PWA first, followed by Android packaging using Capacitor after backend stabilization.

9. Proposed Authentication Flow

Enter phone number -> Generate OTP -> Send SMS through Africa's Talking -> Verify code -> Create or authenticate account -> Issue secure session/API token

- Six-digit OTP with a five-minute expiry.
- OTP stored as a secure hash rather than plain text.
- Resend cooldown, verification attempt limits, and IP/phone rate limiting.
- Kenyan phone normalization to the 2547XXXXXXXX format.
- Separate OTP purposes for registration, login, phone changes, and delivery confirmation.

10. Proposed M-Pesa Payment Flow

Create service request -> Calculate fare -> Initiate KopoKopo STK Push -> Save payment as pending -> Receive callback/webhook -> Mark payment paid -> Begin rider matching

Important control: a rider should not be assigned based only on the initial STK request response. The platform must wait for a verified payment callback or successful transaction status confirmation.

11. Proposed Backend Structure

- /api/auth/request-otp.php and /api/auth/verify-otp.php
- /api/client/register.php, profile.php, and service-request endpoints
- /api/rider/register.php, documents.php, capabilities.php, go-online.php, and accept-job.php
- /api/payments/stk-push.php, status.php, and webhook.php
- /api/tracking/update-location.php and recipient tracking endpoints
- /api/admin/approve-rider.php, dispatch.php, pricing.php, and reporting endpoints
- /services/SmsService.php, OtpService.php, KopoKopoService.php, PaymentService.php, and TrackingService.php

12. Proposed Core Database Tables

clients	riders	rider_documents
rider_capabilities	rider_equipment	admin_users
otp_requests	auth_tokens	service_requests
delivery_stops	recipient_tokens	tracking_locations
payments	payment_attempts	payment_callbacks
rider_earnings	withdrawals	ratings
notifications	disputes	pricing_rules
login_activity		

13. Current Status

Completed / Documented	Pending Real Development
✓ Official branding and themed project portal	• MySQL production database schema
✓ Client UI/UX and workflows	• Real PHP APIs and authentication
✓ Rider UI/UX and onboarding	• Africa's Talking SMS integration
✓ Recipient tracking workflow	• KopoKopo STK Push and webhook integration
✓ Admin dashboard simulation	• Live maps, geocoding, and GPS tracking
✓ Phone OTP and STK Push simulations	• Real rider matching and dispatch logic
✓ Rider capabilities and equipment rules	• Real notifications and recipient secure links
✓ Versioned deployment ZIP packages	• Android packaging and production launch

14. Recommended Development Sequence

1. Database schema and secure configuration structure.
2. Phone OTP authentication for clients and riders.
3. Rider registration, document upload, and admin approval backend.
4. Client service-request creation and pricing engine.
5. KopoKopo M-Pesa STK Push, callbacks, and payment reconciliation.
6. Rider availability, matching, request acceptance, and assignment.
7. Status updates, recipient secure tracking, and delivery confirmation.

- 8. Live GPS tracking and dispatch map.
- 9. Rider earnings, commissions, withdrawals, and finance reporting.
- 10. Operational testing, PWA installation, Android packaging, and go-live.

15. Revised Commercial Milestones - Total Project Cost: KSh 40,000

Milestone	Amount	Coverage
50% Deposit	KSh 20,000	Project activation, technical foundation, database, authentication, and core backend development.
40% Delivery Payment	KSh 16,000	Upon completion and delivery of the agreed functional system and integrations.
10% Post-Go-Live Balance	KSh 4,000	One month after deployment and go-live, subject to the agreed support arrangement.

16. Conclusion

The Fast Rider Delivery Services project has progressed successfully through the concept, workflow, branding, and UI/UX simulation phase. The client, rider, recipient, and administration experiences have been mapped and demonstrated, while the key business rules and technology choices have been agreed. The project is now ready to transition into structured backend development using PHP, MySQL, Africa's Talking SMS OTP, KopoKopo M-Pesa STK Push, and live mapping services.

Prepared by Cypher Africa Group - Bringing The Best Together