

# Software Requirements Specification (SRS)

## Ntizo - Service Marketplace Platform

Document ID	NTIZO-SRS-001
Version	1.0
Date	2026-01-15
Status	Draft
Standard	ISO/IEC/IEEE 29148:2018

## 1. INTRODUCTION

### 1.1 Purpose

This Software Requirements Specification (SRS) describes the functional and non-functional requirements for the **Ntizo Service Marketplace Platform** following ISO/IEC/IEEE 29148:2018 standard.

**Intended Audience:**

- Development team
- Project managers
- Quality assurance team
- Stakeholders and investors

### 1.2 Scope

**Product Name:** Ntizo

**Description:** Two-sided service marketplace connecting customers with verified service providers for task completion.

**Objectives:**

ID	Objective
OBJ-001	Launch MVP in Mozambique
OBJ-002	Onboard 1,000 providers in 6 months
OBJ-003	Process 10,000 bookings in first year
OBJ-004	Achieve 4.0+ average rating
OBJ-005	Expand to 3 countries in 2 years

**In Scope:**

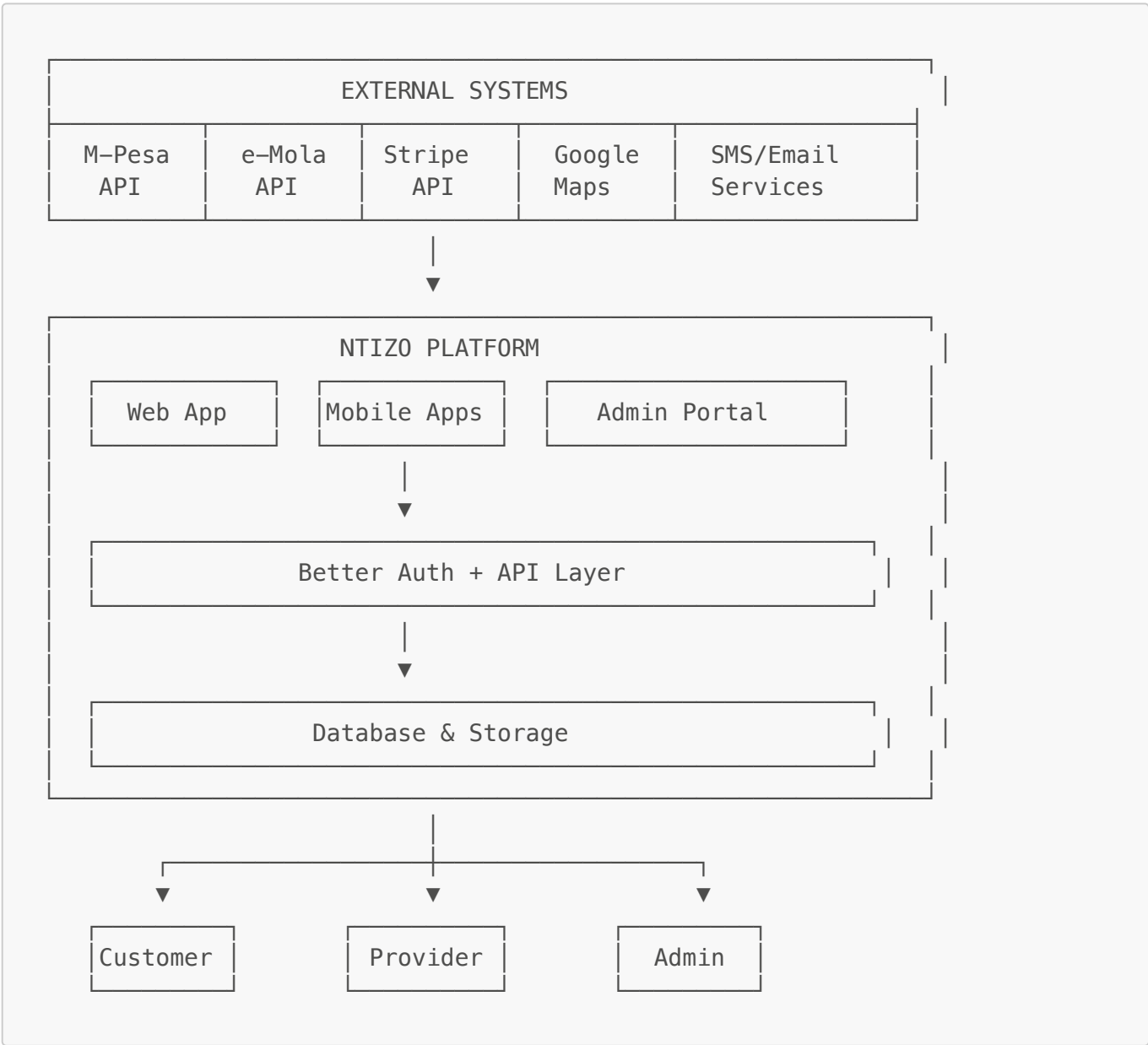
- User authentication (Better Auth)
- Service provider profiles and verification
- Service catalog and search
- Direct booking and task bidding
- In-app messaging
- Payments (M-Pesa, e-Mola, cards)
- Reviews and ratings
- Admin portal

**Out of Scope:**

- Physical logistics/delivery
- Employment relationships
- Financial lending
- Insurance provision

1.3 Product Overview

1.3.1 Product Perspective



### 1.3.2 Product Functions

ID	Function	Description
PF-01	Authentication	Via Better Auth
PF-02	User Management	Profiles
PF-03	Service Catalog	Categories
PF-04	Search	Discovery
PF-05	Direct Booking	Book providers
PF-06	Task Posting	Post for bids
PF-07	Bidding	Provider proposals
PF-08	Messaging	In-app chat
PF-09	Payments	Process transactions
PF-10	Payouts	Provider withdrawals
PF-11	Reviews	Ratings
PF-12	Verification	Provider ID
PF-13	Administration	Platform management

### 1.3.3 User Characteristics

User	Description	Tech Level	Frequency
Customer	Service seekers	Low-Medium	Weekly
Provider	Service professionals	Low-Medium	Daily
Admin	Operations staff	Medium-High	Daily

### 1.3.4 Assumptions

ID	Assumption
ASM-001	Users have Android 8+ or iOS 13+
ASM-002	Users have mobile money or bank cards
ASM-003	Providers can provide valid ID
ASM-004	Payment APIs available in Mozambique

### 1.3.5 Dependencies

ID	Dependency
DEP-001	Better Auth library

ID	Dependency
DEP-002	M-Pesa API access
DEP-003	e-Mola API access
DEP-004	Cloud infrastructure
DEP-005	Firebase Cloud Messaging

## 1.4 Definitions

Term	Definition
Booking	Confirmed customer-provider arrangement
Bid	Provider proposal for a task
Customer	User seeking services
Escrow	Payment holding until completion
Payout	Provider withdrawal
Provider	User offering services
Task	Customer work request

## 2. REFERENCES

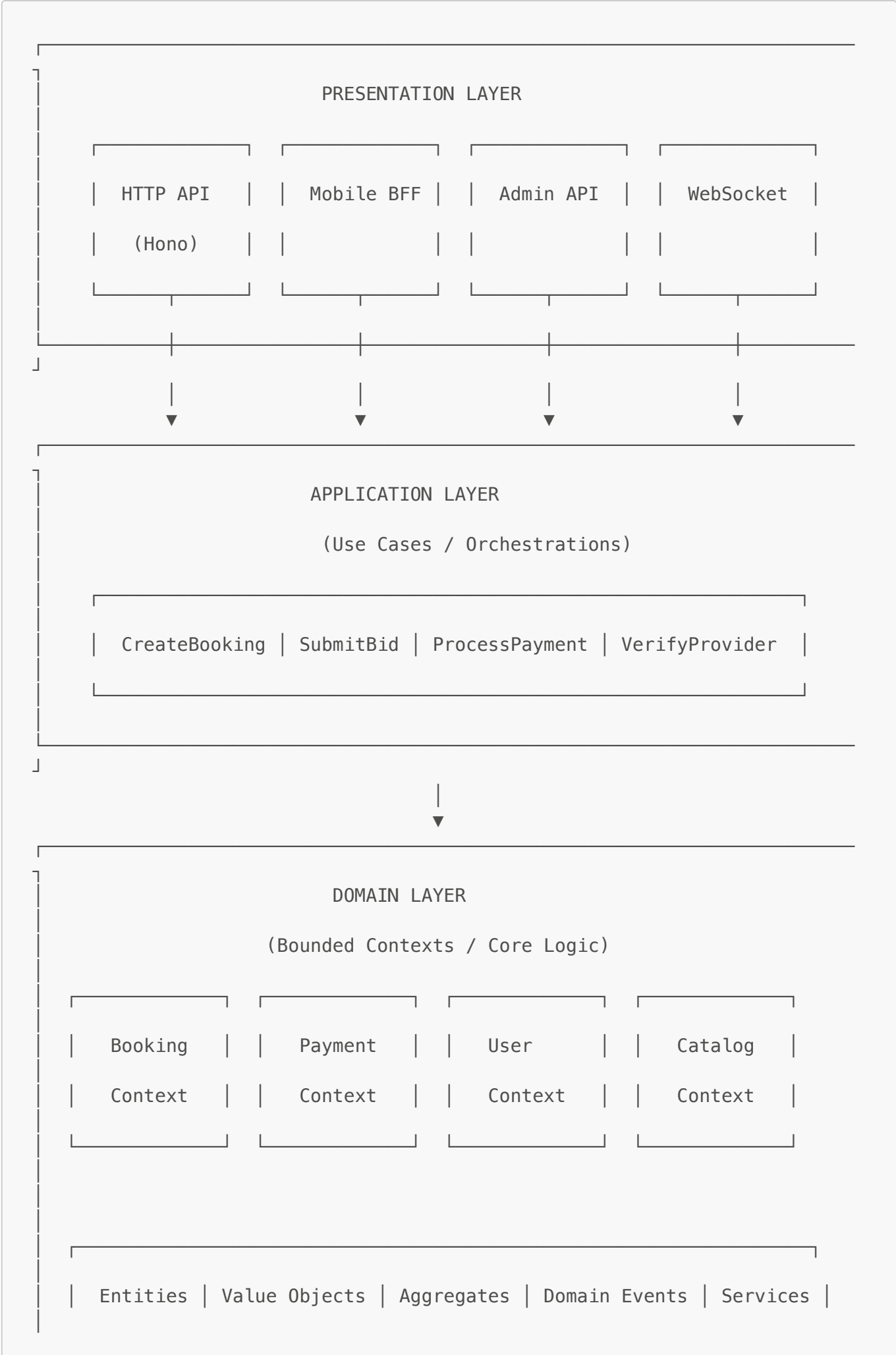
ID	Document
REF-001	ISO/IEC/IEEE 29148:2018
REF-002	Better Auth Documentation
REF-003	OWASP Top 10 (2021)
REF-004	PCI DSS v4.0
REF-005	Onion-Lasagna Architecture ( <a href="https://github.com/Cosmneo/onion-lasagna">github.com/Cosmneo/onion-lasagna</a> )
REF-006	Domain-Driven Design (Eric Evans)

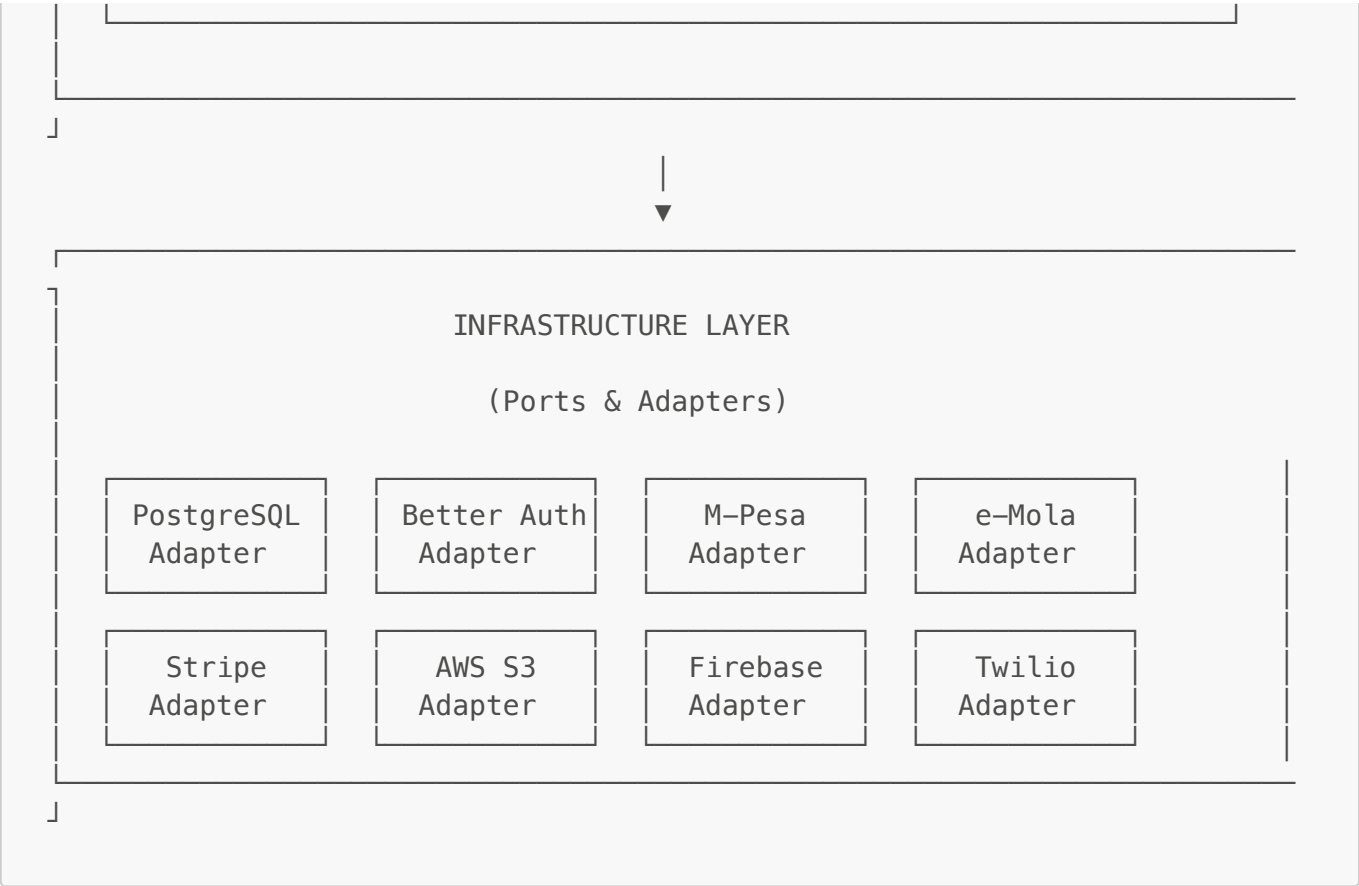
## 3. SPECIFIC REQUIREMENTS

### 3.0 Architecture Requirements

#### 3.0.1 Architecture Pattern

The system shall implement **Hexagonal Architecture** (Ports & Adapters) using the **onion-lasagna** library, following Domain-Driven Design (DDD) principles.





3.0.2 Bounded Contexts

Context	Responsibility	Key Aggregates
User	Authentication, profiles, verification	User, ProviderProfile
Catalog	Services, categories, search	ServiceCategory, Service
Booking	Bookings, tasks, bids, lifecycle	Booking, TaskPost, Bid
Payment	Transactions, escrow, payouts	Payment, Payout, Wallet
Communication	Messaging, notifications	Conversation, Message, Notification
Review	Ratings, reviews	Review

3.0.3 Domain Model

Aggregates

Aggregate	Root Entity	Entities	Value Objects
User	User	ProviderProfile	Email, Phone, UserId, Name
Booking	Booking	-	BookingId, BookingStatus, Location, Money
TaskPost	TaskPost	Bid	TaskId, BidId, Budget
Payment	Payment	-	PaymentId, PaymentMethod, Money, Currency

Aggregate	Root Entity	Entities	Value Objects
<b>Payout</b>	Payout	-	PayoutId, PayoutMethod
<b>Conversation</b>	Conversation	Message	ConversationId, MessageId
<b>Review</b>	Review	-	ReviewId, Rating
<b>ServiceCategory</b>	ServiceCategory	Service	CategoryId, ServiceId

#### Value Objects (using onion-lasagna built-ins)

Value Object	Type	Validation
<b>UserId</b>	UUID	onion-lasagna UUID
<b>Email</b>	Email	onion-lasagna Email
<b>Phone</b>	Text	+258 format, validated
<b>Money</b>	Custom	Amount + Currency (MZN)
<b>Location</b>	Custom	Address + Lat + Lng
<b>Rating</b>	Number	1-5 range
<b>BookingStatus</b>	Enum	pending, confirmed, in_progress, completed, cancelled, disputed
<b>PaymentStatus</b>	Enum	pending, processing, completed, failed, refunded
<b>VerificationStatus</b>	Enum	pending, verified, rejected

#### Domain Events

Event	Trigger	Handlers
<b>UserRegistered</b>	New user created	Send welcome email, create wallet
<b>ProviderVerified</b>	Admin approves provider	Update profile, send notification
<b>BookingCreated</b>	Customer creates booking	Notify provider, hold payment
<b>BookingConfirmed</b>	Provider accepts	Block calendar, notify customer
<b>BookingStarted</b>	Provider marks started	Update status, notify customer
<b>BookingCompleted</b>	Provider marks done	Start confirmation timer
<b>BookingCancelled</b>	Either party cancels	Process refund, update calendar
<b>PaymentProcessed</b>	Payment succeeds	Update booking, send receipt
<b>PaymentFailed</b>	Payment fails	Notify customer, cancel booking
<b>PayoutRequested</b>	Provider requests payout	Process payout
<b>PayoutCompleted</b>	Payout succeeds	Update wallet, notify provider

Event	Trigger	Handlers
TaskPosted	Customer posts task	Notify matching providers
BidSubmitted	Provider bids	Notify customer
BidAccepted	Customer accepts bid	Create booking, reject others
ReviewSubmitted	User submits review	Update ratings, notify reviewee
MessageSent	User sends message	Push notification

### 3.0.4 Application Layer (Use Cases)

#### User Context

Use Case	Input	Output	Events
RegisterUser	email/phone, password, name	User	UserRegistered
LoginUser	credentials	Session	-
UpdateProfile	userId, profileData	User	-
CreateProviderProfile	userId, providerData	ProviderProfile	-
SubmitVerification	providerId, documents	VerificationRequest	-
ApproveVerification	providerId, adminId	ProviderProfile	ProviderVerified
RejectVerification	providerId, adminId, reason	ProviderProfile	-

#### Booking Context

Use Case	Input	Output	Events
CreateDirectBooking	customerId, providerId, details	Booking	BookingCreated
ConfirmBooking	bookingId, providerId	Booking	BookingConfirmed
DeclineBooking	bookingId, providerId, reason	Booking	BookingCancelled
StartBooking	bookingId, providerId	Booking	BookingStarted
CompleteBooking	bookingId, providerId	Booking	BookingCompleted
ConfirmCompletion	bookingId, customerId	Booking	PaymentReleased
DisputeBooking	bookingId, customerId, reason	Dispute	BookingDisputed
CancelBooking	bookingId, userId, reason	Booking	BookingCancelled
PostTask	customerId, taskData	TaskPost	TaskPosted
SubmitBid	taskId, providerId, bidData	Bid	BidSubmitted



Use Case	Input	Output	Events
AcceptBid	bidId, customerId	Booking	BidAccepted, BookingCreated
WithdrawBid	bidId, providerId	Bid	-

### Payment Context

Use Case	Input	Output	Events
AuthorizePayment	bookingId, paymentMethod	PaymentAuth	-
ProcessPayment	paymentAuthId	Payment	PaymentProcessed
RefundPayment	paymentId, amount, reason	Refund	PaymentRefunded
AddTip	bookingId, amount	Payment	TipAdded
RequestPayout	providerId, amount, method	Payout	PayoutRequested
ProcessPayout	payoutId	Payout	PayoutCompleted

### Communication Context

Use Case	Input	Output	Events
SendMessage	conversationId, senderId, content	Message	MessageSent
GetConversation	conversationId, userId	Conversation	-
MarkAsRead	conversationId, userId	-	-
SendNotification	userId, notification	Notification	-

### Review Context

Use Case	Input	Output	Events
SubmitReview	bookingId, reviewerId, reviewData	Review	ReviewSubmitted
RespondToReview	reviewId, providerId, response	Review	-
ReportReview	reviewId, reporterId, reason	Report	-

## 3.0.5 Infrastructure Ports (Interfaces)

### Repository Ports

```
// Domain defines these interfaces (ports)
interface UserRepository {
  findById(id: UserId): Promise<User | null>
  findByEmail(email: Email): Promise<User | null>
  findByPhone(phone: Phone): Promise<User | null>
}
```

```
    save(user: User): Promise<void>
    delete(id: UserId): Promise<void>
}

interface BookingRepository {
    findById(id: BookingId): Promise<Booking | null>
    findByCustomer(customerId: UserId, pagination: Pagination):
    Promise<Booking[]>
    findByProvider(providerId: UserId, pagination: Pagination):
    Promise<Booking[]>
    save(booking: Booking): Promise<void>
}

interface PaymentRepository {
    findById(id: PaymentId): Promise<Payment | null>
    findByBooking(bookingId: BookingId): Promise<Payment | null>
    save(payment: Payment): Promise<void>
}
```

## External Service Ports

```
interface AuthService {
    register(data: RegisterData): Promise<User>
    login(credentials: Credentials): Promise<Session>
    logout(sessionId: string): Promise<void>
    verifyOTP(phone: Phone, otp: string): Promise<boolean>
}

interface PaymentGateway {
    authorize(amount: Money, method: PaymentMethod): Promise<PaymentAuth>
    capture(authId: string): Promise<PaymentResult>
    refund(paymentId: string, amount: Money): Promise<RefundResult>
}

interface PayoutService {
    sendMPesa(phone: Phone, amount: Money): Promise<PayoutResult>
    sendEMola(phone: Phone, amount: Money): Promise<PayoutResult>
    sendBank(account: BankAccount, amount: Money): Promise<PayoutResult>
}

interface NotificationService {
    sendPush(userId: UserId, notification: PushNotification): Promise<void>
    sendEmail(email: Email, template: EmailTemplate): Promise<void>
    sendSMS(phone: Phone, message: string): Promise<void>
}

interface StorageService {
    upload(file: File, path: string): Promise<string>
    delete(path: string): Promise<void>
    getUrl(path: string): Promise<string>
}
```

```
interface LocationService {
  geocode(address: string): Promise<Coordinates>
  reverseGeocode(coords: Coordinates): Promise<Address>
  calculateDistance(from: Coordinates, to: Coordinates): Promise<number>
}
```

### 3.0.6 Infrastructure Adapters

Port	Adapter	Technology
AuthService	BetterAuthAdapter	Better Auth
UserRepository	PostgresUserRepository	PostgreSQL + Drizzle/Prisma
BookingRepository	PostgresBookingRepository	PostgreSQL
PaymentGateway	MPesaAdapter	M-Pesa API
PaymentGateway	EMolaAdapter	e-Mola API
PaymentGateway	StripeAdapter	Stripe API
PayoutService	MPesaPayoutAdapter	M-Pesa B2C API
PayoutService	EMolaPayoutAdapter	e-Mola API
NotificationService	FirebaseAdapter	Firebase Cloud Messaging
NotificationService	TwilioAdapter	Twilio SMS
NotificationService	SendGridAdapter	SendGrid Email
StorageService	S3Adapter	AWS S3
LocationService	GoogleMapsAdapter	Google Maps API

### 3.0.7 Error Hierarchy (onion-lasagna)

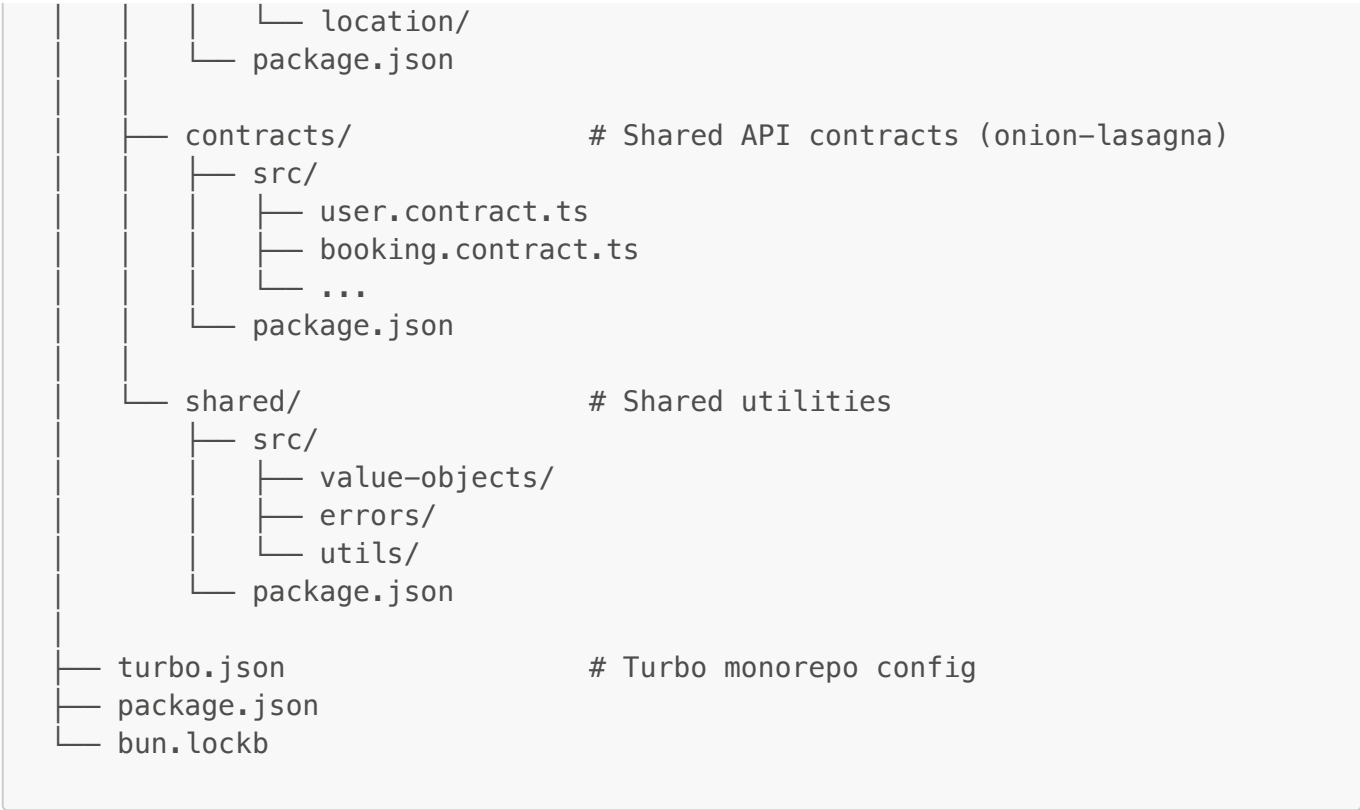
Error Type	HTTP Status	Example
NotFoundError	404	User not found, Booking not found
ConflictError	409	Email already registered, Bid already exists
ValidationError	422	Invalid email format, Price out of range
UnauthorizedError	401	Invalid credentials, Session expired
ForbiddenError	403	Not your booking, Provider not verified
DomainError	400	Cannot cancel completed booking
InfrastructureError	500	Database connection failed

### 3.0.8 Project Structure

```

ntizo/
├── apps/
│   ├── api/                                # Main API (Hono/Elysia)
│   │   ├── src/
│   │   │   ├── routes/                    # HTTP route handlers
│   │   │   ├── middleware/                # Auth, logging, error handling
│   │   │   └── index.ts
│   │   └── package.json
│   ├── admin/                             # Admin portal (React/Next)
│   ├── mobile/                           # Mobile app (React Native)
│   └── web/                              # Customer/Provider web app
├── packages/
│   ├── domain/                            # Domain layer (framework-agnostic)
│   │   ├── src/
│   │   │   ├── user/
│   │   │   │   ├── entities/
│   │   │   │   ├── value-objects/
│   │   │   │   ├── events/
│   │   │   │   ├── repositories/         # Port interfaces
│   │   │   │   └── index.ts
│   │   │   ├── booking/
│   │   │   ├── payment/
│   │   │   ├── catalog/
│   │   │   ├── communication/
│   │   │   └── review/
│   │   └── package.json
│   ├── application/                       # Application layer (use cases)
│   │   ├── src/
│   │   │   ├── user/
│   │   │   │   ├── register-user.ts
│   │   │   │   ├── create-provider-profile.ts
│   │   │   │   └── index.ts
│   │   │   ├── booking/
│   │   │   ├── payment/
│   │   │   └── ...
│   │   └── package.json
│   └── infrastructure/                    # Infrastructure layer (adapters)
│       ├── src/
│       │   ├── database/
│       │   │   ├── schema/
│       │   │   ├── repositories/
│       │   │   └── migrations/
│       │   ├── auth/
│       │   │   └── better-auth-adapter.ts
│       │   ├── payment/
│       │   │   ├── mpesa-adapter.ts
│       │   │   ├── emola-adapter.ts
│       │   │   └── stripe-adapter.ts
│       │   ├── notification/
│       │   └── storage/

```



3.0.9 Technology Stack

Layer	Technology	Rationale
Runtime	Bun	Fast, TypeScript-native
Monorepo	Turborepo	Build optimization
API Framework	Hono	Lightweight, edge-compatible
Validation	Zod	Runtime type checking
Database	Neon PostgreSQL	Serverless, auto-scaling
ORM	Drizzle	Type-safe, performant
Auth	Better Auth	Full-featured, TypeScript
Architecture	onion-lasagna	DDD building blocks
Mobile	React Native	Cross-platform
Web	Next.js	SSR, React
State	TanStack Query	Server state management
Deployment	Vercel / Cloudflare	Serverless, edge
Cache	Upstash Redis	Serverless Redis
Queue	Inngest / QStash	Serverless background jobs
Storage	Cloudflare R2 / S3	Object storage
WebSockets	Abyl / Pusher	Real-time messaging

### 3.1 External Interfaces

#### 3.1.1 User Interfaces

ID	Interface
UI-001	Customer Mobile App (iOS/Android)
UI-002	Provider Mobile App (iOS/Android)
UI-003	Customer Web App
UI-004	Provider Web App
UI-005	Admin Portal

#### 3.1.2 Software Interfaces

ID	Interface	Purpose
SW-001	Better Auth	Authentication
SW-002	M-Pesa API	Mobile money
SW-003	e-Mola API	Mobile money
SW-004	Stripe API	Card payments
SW-005	Google Maps API	Location
SW-006	Firebase CM	Push notifications
SW-007	Twilio	SMS
SW-008	SendGrid	Email
SW-009	AWS S3	File storage

### 3.2 Functional Requirements

#### 3.2.1 Authentication (Better Auth)

ID	Requirement	Priority
FR-AUTH-001	The system shall implement Better Auth for authentication	High
FR-AUTH-002	The system shall allow email registration	High
FR-AUTH-003	The system shall allow phone registration	High
FR-AUTH-004	The system shall support social login (Google, Facebook, Apple)	Medium
FR-AUTH-005	The system shall send OTP for phone verification	High
FR-AUTH-006	The system shall send email verification	High
FR-AUTH-007	The system shall require password min 8 chars with letter and number	High

ID	Requirement	Priority
FR-AUTH-008	The system shall support passwordless OTP login	Medium
FR-AUTH-009	The system shall lock account after 5 failed attempts	High
FR-AUTH-010	The system shall support password reset via email/SMS	High
FR-AUTH-011	The system shall manage sessions via Better Auth	High
FR-AUTH-012	The system shall expire sessions after 7 days inactivity	High
FR-AUTH-013	The system shall invalidate sessions on logout	High
FR-AUTH-014	The system shall implement RBAC (customer, provider, admin)	High

### 3.2.2 User Profile

ID	Requirement	Priority
FR-PROF-001	The system shall allow profile creation with name and photo	High
FR-PROF-002	The system shall allow multiple verified phones	Medium
FR-PROF-003	The system shall allow multiple verified emails	Medium
FR-PROF-004	The system shall allow setting primary contact	Medium
FR-PROF-005	The system shall allow profile updates	High
FR-PROF-006	The system shall resize images to max 1MB	High
FR-PROF-007	The system shall support account deletion (30-day retention)	High

### 3.2.3 Provider Profile

ID	Requirement	Priority
FR-PROV-001	The system shall allow provider registration	High
FR-PROV-002	The system shall require bio (min 50 chars)	High
FR-PROV-003	The system shall allow portfolio images (max 10)	High
FR-PROV-004	The system shall allow service category selection	High
FR-PROV-005	The system shall allow hourly or fixed pricing	High
FR-PROV-006	The system shall allow service area definition	High
FR-PROV-007	The system shall allow availability schedule	High
FR-PROV-008	The system shall allow date blocking	Medium
FR-PROV-009	The system shall allow instant booking toggle	Medium
FR-PROV-010	The system shall display average rating	High

ID	Requirement	Priority
FR-PROV-011	The system shall display completed task count	High
FR-PROV-012	The system shall display verification badges	High

3.2.4 Provider Verification

ID	Requirement	Priority
FR-VER-001	The system shall require government ID upload	High
FR-VER-002	The system shall accept BI, passport, license	High
FR-VER-003	The system shall accept front and back images	High
FR-VER-004	The system shall encrypt stored documents	High
FR-VER-005	The system shall queue for admin review	High
FR-VER-006	The system shall notify verification status	High
FR-VER-007	The system shall allow admin approve/reject	High
FR-VER-008	The system shall allow re-submission if rejected	High
FR-VER-009	The system shall display "Verified" badge	High

3.2.5 Service Catalog

ID	Requirement	Priority
FR-CAT-001	The system shall support hierarchical categories	High
FR-CAT-002	The system shall allow admin CRUD on categories	High
FR-CAT-003	The system shall support category icons and order	Medium
FR-CAT-004	The system shall allow enable/disable categories	High

Service Categories:

Category	Subcategories
Home Cleaning	Regular, Deep, Move-in/out
Repairs	Plumbing, Electrical, General, Painting
Assembly	Furniture, TV mounting, Appliances
Moving	Home, Office, Delivery
Gardening	Lawn, Garden, Trees
Personal	Assistant, Errands, Shopping
Events	Planning, Catering, Photography



Category	Subcategories
Beauty	Hair, Makeup, Massage
Tutoring	Academic, Language, Music
Technology	Computer, Tech support, Web
Professional	Accounting, Legal, Translation
Automotive	Washing, Repairs, Driver

### 3.2.6 Search and Discovery

ID	Requirement	Priority
FR-SRCH-001	The system shall allow keyword search	High
FR-SRCH-002	The system shall allow category browsing	High
FR-SRCH-003	The system shall allow location filter	High
FR-SRCH-004	The system shall allow distance filter	High
FR-SRCH-005	The system shall allow price range filter	Medium
FR-SRCH-006	The system shall allow rating filter	Medium
FR-SRCH-007	The system shall allow verification filter	Medium
FR-SRCH-008	The system shall allow availability filter	Medium
FR-SRCH-009	The system shall support sorting	Medium
FR-SRCH-010	The system shall paginate (20/page)	High

### 3.2.7 Direct Booking

ID	Requirement	Priority
FR-BOOK-001	The system shall allow direct booking from profile	High
FR-BOOK-002	The system shall display availability calendar	High
FR-BOOK-003	The system shall allow date/time selection	High
FR-BOOK-004	The system shall allow location entry	High
FR-BOOK-005	The system shall allow GPS location	High
FR-BOOK-006	The system shall allow saving addresses	Medium
FR-BOOK-007	The system shall allow task description (1000 chars)	High
FR-BOOK-008	The system shall allow image attachments (max 5)	Medium
FR-BOOK-009	The system shall calculate total price	High

ID	Requirement	Priority
FR-BOOK-010	The system shall display price breakdown	High
FR-BOOK-011	The system shall require payment authorization	High
FR-BOOK-012	The system shall send to provider for confirmation	High
FR-BOOK-013	The system shall allow accept/decline (24h)	High
FR-BOOK-014	The system shall auto-cancel on timeout	High
FR-BOOK-015	The system shall notify booking status	High
FR-BOOK-016	The system shall block calendar on confirmation	High

### 3.2.8 Task Posting and Bidding

ID	Requirement	Priority
FR-TASK-001	The system shall allow task posting	High
FR-TASK-002	The system shall require title (100 chars)	High
FR-TASK-003	The system shall require description (20-2000 chars)	High
FR-TASK-004	The system shall require category	High
FR-TASK-005	The system shall require location	High
FR-TASK-006	The system shall allow optional date/time	High
FR-TASK-007	The system shall allow optional budget range	Medium
FR-TASK-008	The system shall allow images (max 5)	Medium
FR-TASK-009	The system shall publish to matching providers	High
FR-TASK-010	The system shall allow providers to browse tasks	High
FR-TASK-011	The system shall allow bid submission	High
FR-TASK-012	The system shall require bid price	High
FR-TASK-013	The system shall require bid message (20 chars)	High
FR-TASK-014	The system shall allow alternative date in bid	Medium
FR-TASK-015	The system shall notify customer of bids	High
FR-TASK-016	The system shall display all bids	High
FR-TASK-017	The system shall allow accepting one bid	High
FR-TASK-018	The system shall reject others on acceptance	High
FR-TASK-019	The system shall convert accepted bid to booking	High
FR-TASK-020	The system shall allow closing without accepting	Medium

ID	Requirement	Priority
FR-TASK-021	The system shall auto-expire after 7 days	Medium
FR-TASK-022	The system shall allow bid withdrawal	Medium

### 3.2.9 Booking Lifecycle

ID	Requirement	Priority
FR-LIFE-001	The system shall track status: pending, confirmed, in_progress, completed, cancelled, disputed	High
FR-LIFE-002	The system shall allow marking "started"	High
FR-LIFE-003	The system shall record start timestamp	High
FR-LIFE-004	The system shall allow marking "completed"	High
FR-LIFE-005	The system shall record completion timestamp	High
FR-LIFE-006	The system shall require customer confirmation (24h)	High
FR-LIFE-007	The system shall auto-confirm after 24h	High
FR-LIFE-008	The system shall release payment on confirmation	High
FR-LIFE-009	The system shall allow dispute within 24h	High
FR-LIFE-010	The system shall allow cancellation before start	High
FR-LIFE-011	The system shall apply cancellation policy	High
FR-LIFE-012	The system shall allow provider cancellation	High
FR-LIFE-013	The system shall track cancellation rate	Medium
FR-LIFE-014	The system shall send reminders (24h, 1h)	Medium

#### Cancellation Policy:

Time Before	Refund	Provider Penalty
24+ hours	100%	None
12-24 hours	75%	None
2-12 hours	50%	None
< 2 hours	0%	None
Provider cancels	100%	Rate impact

### 3.2.10 Communication

ID	Requirement	Priority
FR-COMM-001	The system shall provide in-app messaging	High
FR-COMM-002	The system shall create conversation on confirmation	High
FR-COMM-003	The system shall allow text (2000 chars)	High
FR-COMM-004	The system shall allow images (5MB)	Medium
FR-COMM-005	The system shall display timestamps	High
FR-COMM-006	The system shall show read status	Medium
FR-COMM-007	The system shall send push for messages	High
FR-COMM-008	The system shall preserve history 12 months	High
FR-COMM-009	The system shall support dispute evidence	High
FR-COMM-010	The system shall warn on contact sharing	Medium
FR-COMM-011	The system shall allow reporting messages	High

### 3.2.11 Payments

ID	Requirement	Priority
FR-PAY-001	The system shall support M-Pesa	High
FR-PAY-002	The system shall support e-Mola	High
FR-PAY-003	The system shall support Visa/Mastercard	High
FR-PAY-004	The system shall allow saving payment methods	Medium
FR-PAY-005	The system shall process in MZN	High
FR-PAY-006	The system shall hold in escrow	High
FR-PAY-007	The system shall release on completion	High
FR-PAY-008	The system shall deduct platform commission	High

ID	Requirement	Priority
FR-PAY-009	The system shall apply configurable commission	High
FR-PAY-010	The system shall display earnings breakdown	High
FR-PAY-011	The system shall support tipping	Medium
FR-PAY-012	The system shall pass 100% tips to provider	Medium
FR-PAY-013	The system shall process refunds per policy	High
FR-PAY-014	The system shall generate receipts	High
FR-PAY-015	The system shall maintain transaction history	High

### 3.2.12 Provider Payouts

ID	Requirement	Priority
FR-POUT-001	The system shall track balance (pending/available)	High
FR-POUT-002	The system shall move to available on completion	High
FR-POUT-003	The system shall allow payout requests	High
FR-POUT-004	The system shall support M-Pesa payouts	High
FR-POUT-005	The system shall support e-Mola payouts	High
FR-POUT-006	The system shall support bank payouts	Medium
FR-POUT-007	The system shall enforce minimum threshold	High
FR-POUT-008	The system shall process within 2 business days	High
FR-POUT-009	The system shall notify on processing	High
FR-POUT-010	The system shall provide payout history	Medium

### 3.2.13 Reviews and Ratings

ID	Requirement	Priority
FR-REV-001	The system shall prompt review after completion	High
FR-REV-002	The system shall allow 1-5 star rating	High
FR-REV-003	The system shall allow text review (500 chars)	High
FR-REV-004	The system shall allow aspect ratings	Medium
FR-REV-005	The system shall allow provider to rate customer	Medium
FR-REV-006	The system shall limit to 7 days after completion	High
FR-REV-007	The system shall display on profile	High

ID	Requirement	Priority
FR-REV-008	The system shall calculate average rating	High
FR-REV-009	The system shall allow provider response	Medium
FR-REV-010	The system shall allow reporting reviews	Medium
FR-REV-011	The system shall not allow editing reviews	High

### 3.2.14 Notifications

ID	Requirement	Priority
FR-NOT-001	The system shall send push for critical events	High
FR-NOT-002	The system shall send email for bookings	High
FR-NOT-003	The system shall send SMS (optional)	Medium
FR-NOT-004	The system shall allow preferences	Medium
FR-NOT-005	The system shall maintain notification center	High
FR-NOT-006	The system shall track read/unread	Medium

#### Notification Matrix:

Event	Push	Email	SMS
New booking	✓	✓	✓
Confirmed	✓	✓	✓
Cancelled	✓	✓	✓
New message	✓	-	-
New bid	✓	✓	-
Bid accepted	✓	✓	✓
Task started	✓	-	-
Completed	✓	✓	-
Payment	✓	✓	-
Payout	✓	✓	✓
New review	✓	✓	-
Verification	✓	✓	✓
Reminder	✓	-	✓

### 3.2.15 Administration

ID	Requirement	Priority
FR-ADM-001	The system shall provide dashboard with metrics	High
FR-ADM-002	The system shall display KPIs	High
FR-ADM-003	The system shall allow user search/view	High
FR-ADM-004	The system shall allow user suspension	High
FR-ADM-005	The system shall allow reactivation	High
FR-ADM-006	The system shall allow deletion (soft)	Medium
FR-ADM-007	The system shall provide verification queue	High
FR-ADM-008	The system shall allow approve/reject	High
FR-ADM-009	The system shall allow category management	High
FR-ADM-010	The system shall allow booking viewing	Medium
FR-ADM-011	The system shall provide dispute queue	Medium
FR-ADM-012	The system shall allow dispute resolution	Medium
FR-ADM-013	The system shall allow payout management	High
FR-ADM-014	The system shall allow configuration	Medium
FR-ADM-015	The system shall maintain audit log	Medium
FR-ADM-016	The system shall support admin roles	Medium

### 3.3 Usability Requirements

ID	Requirement	Target
NFR-USE-001	Registration time	< 3 min
NFR-USE-002	Booking time	< 5 min
NFR-USE-003	Consistent navigation	-
NFR-USE-004	Standard UI conventions	-
NFR-USE-005	Clear error messages	-
NFR-USE-006	Session during connectivity loss	-

### 3.4 Performance Requirements

ID	Requirement	Target
NFR-PERF-001	App load (3G)	< 3s
NFR-PERF-002	API response (p95)	< 500ms

ID	Requirement	Target
NFR-PERF-003	Search results	< 2s
NFR-PERF-004	Payment processing	< 10s
NFR-PERF-005	Concurrent users	10,000
NFR-PERF-006	API requests/sec	1,000
NFR-PERF-007	Push delivery	< 5s
NFR-PERF-008	Image upload (5MB)	< 10s
NFR-PERF-009	Database scale	1M+ users

### 3.5 Database Requirements

#### 3.5.1 Data Entities

Entity	Retention
User (Better Auth)	Lifetime + 30 days
ProviderProfile	Lifetime
ServiceCategory	Permanent
Service	Lifetime
TaskPost	2 years
Bid	2 years
Booking	7 years
Payment	7 years
Payout	7 years
Message	12 months
Review	Lifetime
Notification	90 days

#### 3.5.2 Entity Definitions

USER (Better Auth)

- id, email, emailVerified, phone, phoneVerified
- name, image, role
- createdAt, updatedAt

PROVIDER\_PROFILE

- id, userId, bio, hourlyRate
- serviceAreas, skills, portfolioImages



- idDocumentUrl, verificationStatus
- averageRating, totalReviews, totalCompletedTasks
- instantBooking, createdAt, updatedAt

#### SERVICE\_CATEGORY

- id, name, nameEn, description
- iconUrl, parentId, isActive, sortOrder
- createdAt, updatedAt

#### SERVICE

- id, providerId, categoryId
- title, description, priceType, price
- durationEstimate, isActive, createdAt

#### TASK\_POST

- id, customerId, categoryId
- title, description, location (address, lat, lng)
- preferredDate, preferredTime, budget (min, max)
- images, status
- createdAt, updatedAt, expiresAt

#### BID

- id, taskId, providerId
- proposedPrice, message
- proposedDate, proposedTime, status
- createdAt, updatedAt

#### BOOKING

- id, customerId, providerId
- serviceId, taskId, bidId
- scheduledDate, scheduledTime, durationHours
- location (address, lat, lng), description, images
- totalAmount, platformFee, providerEarnings, tipAmount
- status, startedAt, completedAt, customerConfirmedAt
- cancelledAt, cancellationReason, createdAt

#### PAYMENT

- id, bookingId, payerId
- amount, currency, paymentMethod
- providerReference, status
- paidAt, refundedAt, createdAt

#### PAYOUT

- id, providerId, amount, currency
- method, destination, status
- processedAt, failureReason, createdAt

#### MESSAGE

- id, conversationId, senderId, receiverId
- bookingId, content, imageUrl
- isRead, readAt, createdAt

#### REVIEW

- id, bookingId, reviewerId, revieweeId

└─ rating, punctualityRating, qualityRating

└─ communicationRating, comment

└─ response, respondedAt, createdAt

NOTIFICATION

└─ id, userId, type, title

└─ message, data, isRead

└─ readAt, createdAt

### 3.6 Design Constraints

ID	Constraint	Rationale
NFR-CON-001	Hexagonal Architecture (onion-lasagna)	Clean separation, testability
NFR-CON-002	Domain-Driven Design patterns	Complex domain modeling
NFR-CON-003	Bun runtime	Performance, TypeScript-native
NFR-CON-004	Turborepo monorepo	Build optimization, code sharing
NFR-CON-005	Hono API framework	Lightweight, edge-compatible
NFR-CON-006	Better Auth for authentication	Full-featured, TypeScript
NFR-CON-007	Zod for validation	Runtime type checking
NFR-CON-008	PostgreSQL (serverless) + Drizzle ORM	Neon/Supabase/PlanetScale
NFR-CON-009	React Native for mobile	Cross-platform
NFR-CON-010	Next.js for web	SSR, React ecosystem
NFR-CON-011	Serverless deployment	No container management
NFR-CON-012	Edge-first architecture	Low latency, global
NFR-CON-013	PCI DSS compliance	Payment security

#### 3.6.1 Serverless Infrastructure

Component	Service Options
API	Vercel Functions, Cloudflare Workers, AWS Lambda
Database	Neon (PostgreSQL), Supabase, PlanetScale
Storage	Cloudflare R2, AWS S3, Supabase Storage
Auth	Better Auth (self-hosted on serverless)
Queue/Jobs	Upstash QStash, Inngest, Trigger.dev
Caching	Upstash Redis, Vercel KV

Component	Service Options
WebSockets	Ably, Pusher, PartyKit
Email	Resend, SendGrid
SMS	Twilio
Push	Firebase Cloud Messaging

### 3.7 System Attributes

#### 3.7.1 Reliability

ID	Requirement	Target
NFR-REL-001	Uptime	99.9%
NFR-REL-002	MTTR	< 1 hour
NFR-REL-003	Graceful failure handling	-
NFR-REL-004	Auto failover	-

#### 3.7.2 Security

ID	Requirement
NFR-SEC-001	TLS 1.2+ for transit
NFR-SEC-002	AES-256 at rest
NFR-SEC-003	Better Auth password hashing
NFR-SEC-004	Better Auth sessions
NFR-SEC-005	Rate limiting
NFR-SEC-006	Input validation
NFR-SEC-007	CSRF protection
NFR-SEC-008	SQL injection prevention
NFR-SEC-009	Auth logging
NFR-SEC-010	Admin audit logging
NFR-SEC-011	OWASP Top 10 compliance

#### 3.7.3 Scalability

ID	Requirement
NFR-SCL-001	Horizontal scaling

ID	Requirement
NFR-SCL-002	Database read replicas
NFR-SCL-003	Redis caching
NFR-SCL-004	CDN for assets
NFR-SCL-005	Multi-region support
NFR-SCL-006	Message queues

3.7.4 Maintainability

ID	Requirement
NFR-MNT-001	> 80% documentation
NFR-MNT-002	Comprehensive logging
NFR-MNT-003	Semantic versioning
NFR-MNT-004	Feature flags
NFR-MNT-005	Zero-downtime migrations
NFR-MNT-006	> 70% test coverage
NFR-MNT-007	CI/CD pipeline

3.8 Localization

ID	Requirement
NFR-LOC-001	Portuguese (pt-MZ) primary
NFR-LOC-002	English secondary
NFR-LOC-003	MZN currency
NFR-LOC-004	Format: X.XXX,XX MZN
NFR-LOC-005	Timezone: Africa/Maputo (UTC+2)
NFR-LOC-006	Date: DD/MM/YYYY
NFR-LOC-007	Time: 24-hour
NFR-LOC-008	Phone: +258 XX XXX XXXX

4. VERIFICATION

4.1 Methods

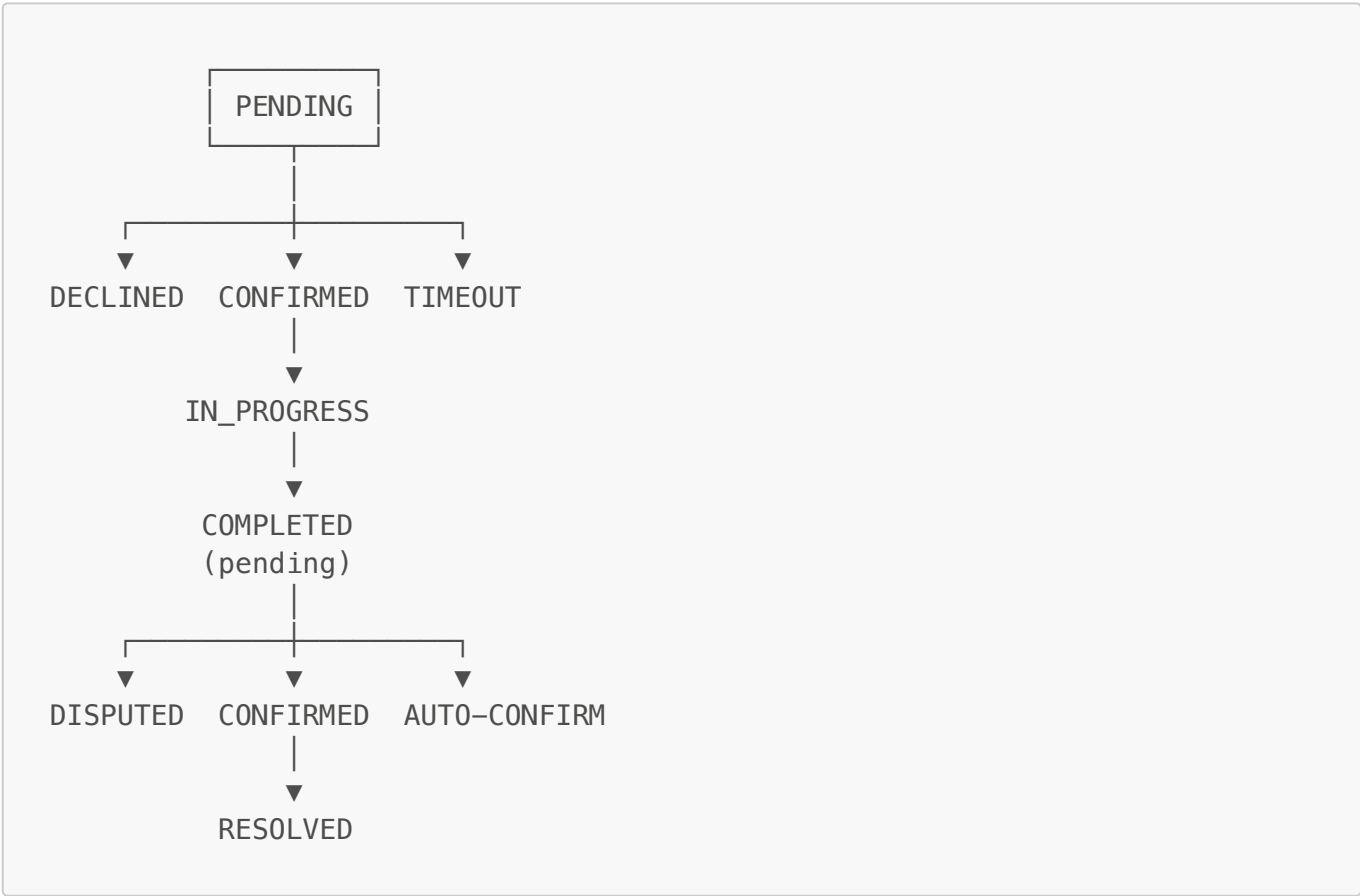
Method	Description
Test	Execute test cases
Inspection	Manual review
Analysis	System evaluation
Demonstration	Operational demo

## 4.2 Acceptance Criteria

- 1. All **High** priority requirements verified
- 2.  $\geq 95\%$  of **Medium** requirements verified
- 3. 99% uptime during acceptance
- 4. Performance benchmarks met
- 5. No critical security vulnerabilities
- 6.  $\geq 90\%$  user satisfaction

# 5. APPENDICES

## 5.1 Booking State Diagram



## 5.2 Open Issues

ID	Issue	Owner
----	-------	-------

ID	Issue	Owner
ISSUE-001	Commission percentage	Business
ISSUE-002	Minimum payout threshold	Business
ISSUE-003	Cancellation time windows	Business
ISSUE-004	M-Pesa API credentials	Technical
ISSUE-005	e-Mola API credentials	Technical
ISSUE-006	Better Auth configuration	Technical

Document End

Role	Name	Date	Signature
Prepared		2026-01-15	
Reviewed			
Approved			