

mojo dojo casa house - Multi-Role Booking Platform

Product Requirements Document

1. Product Overview

****Product Name:**** mojo dojo casa house

****Product Type:**** Multi-role booking platform (similar to Airbnb)

****Target Users:**** Hosts, Guests, and Administrators

mojo dojo casa house is a comprehensive booking platform that enables hosts to list properties (homes, experiences, services) and guests to discover, book, and review them. The platform includes a complete payment system, messaging, reviews, and administrative oversight.

2. Core Features

2.1 Authentication & User Management

- ****Multi-role authentication:**** Users can be Hosts, Guests, or Admins
- ****Sign-in methods:**** Email/Password and Google OAuth
- ****Email verification:**** Required for account activation
- ****OTP verification:**** Alternative verification method
- ****Password reset:**** Forgot password flow
- ****Policy acceptance:**** Hosts must accept terms before listing

2.2 Host Portal Features

- ****Listing Management:****
 - Create listings with multiple images (up to 10)
 - Edit and delete listings

- Auto-save drafts (drafts don't appear in listings view)
- Categories: Home, Experience, Service
- Location picker with map integration
- Availability calendar (available/blocked dates)
- Pricing per night
- Publish fee: 1 day of booking cost

- **Booking Management:**

- View booking requests
- Approve/decline bookings
- Calendar view of bookings

- **Financial:**

- Track earnings
- View transaction history
- Payment processing via PayPal

- **Communication:**

- Real-time messaging with guests
- Notification system

- **Settings:**

- Profile management
- Account settings

2.3 Guest Portal Features

- **Browsing & Discovery:**

- Browse listings by category
- Search with autocomplete
- Advanced filtering (location, price, guests, dates)
- Location-based search

- **Booking:**

- View detailed listing information
- Date selection for bookings
- Apply coupon codes
- Earn points on bookings
- Make deposits via PayPal or wallet
- View booking history

- **Social Features:**

- Save favorites
- Create wishlists
- Leave reviews and ratings
- View average ratings

- **Financial:**

- E-wallet system
- Deposit management
- Transaction history
- Coupon system (apply discounts)
- Points & Rewards system:

- Earn 50 points per booking
- Earn 20 points per review
- Redeem points for discount coupons

- ****Communication:****
 - Real-time messaging with hosts
 - Notification system

- ****Settings:****
 - Profile management
 - Account settings

2.4 Admin Portal Features

- ****Listing Management:****
 - Review pending listings
 - Approve/reject listings
 - View all listings

- ****User Management:****
 - View all users
 - Manage user roles

- ****Financial Monitoring:****
 - View all payments
 - Monitor transactions

- ****Analytics & Reports:****
 - Dashboard with platform metrics
 - Generate reports
- ****Policy Management:****
 - Manage platform policies

3. Technical Architecture

3.1 Frontend Stack

- ****Framework:**** React 18 with TypeScript
- ****Build Tool:**** Vite
- ****UI Library:**** shadcn/ui (Radix UI components)
- ****Styling:**** TailwindCSS
- ****Routing:**** React Router v6
- ****State Management:**** React Query (TanStack Query) + React Context
- ****Forms:**** React Hook Form with Zod validation
- ****Theming:**** next-themes (light/dark mode support)

3.2 Backend Stack

- ****Platform:**** Firebase
- ****Services:****
 - Authentication (Email/Password, Google OAuth)
 - Firestore Database
 - Storage (for images)
 - Hosting

3.3 Integrations

- **Payments:** PayPal SDK
- **Email:** EmailJS
- **Maps:** Leaflet / React Leaflet
- **Images:** Cloudinary
- **Notifications:** Sonner (toast notifications)

4. Data Models

4.1 Firestore Collections

users

- User profiles with roles (host, guest, admin)
- Wallet balance
- Points balance
- Profile information

listings

- Host listings with details
- Images (stored in Cloudinary)
- Pricing, location, availability
- Status: draft, pending, approved, rejected
- Category: home, experience, service

bookings

- Booking requests with status
- Dates, pricing
- Coupon code and discount amount
- Original price and final price
- Guest and host IDs

****reviews****

- Guest reviews and ratings
- Average rating calculation
- Review text and rating (1-5 stars)

****messages****

- Direct messaging between users
- Real-time updates

****transactions****

- Wallet and payment transactions
- Deposit, refund, reward transactions
- Audit trail

****notifications****

- Real-time notifications for users
- Notification types: booking, message, review, etc.

****coupons****

- Discount coupons

- Usage tracking (used, usedAt, usedForBookingId)
- Fixed discount amounts

5. Key User Flows

5.1 Host Listing Creation Flow

1. Host navigates to "Create Listing"
2. Fills out form (title, description, location, price, etc.)
3. Uploads images (up to 10)
4. Sets availability calendar
5. Form auto-saves as draft every 2 seconds
6. Host clicks "Publish"
7. System calculates publish fee (1 day of booking cost)
8. Host pays publish fee via PayPal
9. Listing status changes to "pending"
10. Admin reviews and approves/rejects
11. If approved, listing appears in guest browse/search

5.2 Guest Booking Flow

1. Guest browses/search listings
2. Views listing details
3. Selects dates
4. Optionally applies coupon code
5. Views total (subtotal - discount)
6. Makes deposit payment (PayPal or wallet)
7. Booking status: "pending"

8. Host approves/declines
9. If approved, guest earns 50 points
10. Booking status: "confirmed"

5.3 Guest Points & Rewards Flow

1. Guest makes booking → earns 50 points
2. Guest writes review → earns 20 points
3. Guest accumulates points
4. Guest can redeem points for discount coupons
5. Coupons can be applied to future bookings

5.4 Payment Flow

1. Guest initiates booking
2. System calculates total (with coupon if applied)
3. Guest pays deposit via PayPal or wallet
4. Payment service processes payment
5. If successful:
 - Booking created
 - Coupon marked as used (if applicable)
 - Points awarded (if booking)
 - Transaction recorded
6. If refund needed:
 - Coupon restored (if applicable)
 - Refund processed
 - Transaction recorded

6. Business Rules

1. ****Draft Listings:**** Drafts are saved automatically but don't appear in listings management view
2. ****Publish Fee:**** Hosts pay 1 day of booking cost to publish a listing
3. ****Listing Status:**** draft → pending → approved/rejected
4. ****Booking Status:**** pending → confirmed/cancelled
5. ****Points System:****
 - 50 points per booking
 - 20 points per review
 - Points can be redeemed for coupons
6. ****Coupons:**** Fixed discount amounts, one-time use per coupon
7. ****Guest Visibility:**** Guests only see approved listings
8. ****Host Visibility:**** Hosts see all their listings except drafts

7. Security & Validation

- Input sanitization on all user inputs
- Zod schema validation for forms
- Firebase security rules for database access
- Role-based access control
- Protected routes per user type
- Email verification required
- Password strength requirements (min 8 characters)

8. Error Handling

- Error boundaries catch React errors
- Toast notifications for user feedback
- Loading states for async operations
- Graceful error messages
- Fallback UI for failed operations

9. Performance Optimizations

- Lazy loading for routes
- Code splitting
- Image optimization via Cloudinary
- Video lazy loading with IntersectionObserver
- React.memo, useMemo, useCallback optimizations
- Firestore query optimization

10. Testing Requirements

- Unit tests for utility functions
- Integration tests for key flows
- E2E tests for critical user journeys
- Form validation tests
- Payment flow tests
- Authentication flow tests

11. Known Issues & Future Enhancements

****Current Limitations:****

- Draft listings are saved but filtered from listings view
- Coupons are fixed amount (not percentage)
- Points system is basic (could add more reward tiers)

****Future Enhancements:****

- Percentage-based coupons
- More reward tiers
- Advanced analytics
- Mobile app
- Multi-language support