

OneClickHub

Multi-Tenant Service Marketplace Platform

Version 1.0 — Project Blueprint

Date: February 2026

Connecting Service Providers with Customers — One Click at a Time

Table of Contents

1. [Executive Summary](#)
2. [Technology Stack](#)
3. [System Architecture](#)
4. [User Roles & Access Control](#)
5. [Core Features](#)
6. [Database Schema](#)
7. [Key User Flows](#)
8. [Routes & API Overview](#)
9. [Payment Gateway Integration](#)
10. [Notification System](#)
11. [Future Milestones](#)

1. Executive Summary

OneClickHub is a multi-tenant service marketplace platform built for the Malaysian market. It connects **freelancers** (service providers) with **customers** seeking services across diverse categories — from catering and event planning to web design and digital marketing.

Problem Statement

Freelancers in Malaysia lack a centralized, subscription-based platform to list their services, manage bookings, and receive payments. Customers struggle to discover and compare local service providers.

Solution

OneClickHub provides:

- A **subscription-based model** where freelancers pay to list services on the platform
- A **service marketplace** for customers to browse, book, and review services
- **Integrated payment gateways** (Bayarcash, SenangPay) supporting Malaysian banking
- **Business verification** (SSM) with AI-powered document extraction
- **Order management** with chat, availability calendars, and payment slip verification
- A comprehensive **admin dashboard** for platform oversight

Target Users

Role	Description
Admin	Platform operators managing users, plans, gateways, and revenue
Freelancer	Service providers subscribing to plans and listing services
Customer	End users browsing, booking, and reviewing services

2. Technology Stack

Backend

Technology	Purpose
Laravel 12 (PHP 8.3)	Core framework
MySQL	Primary database
Laravel Sanctum	API authentication (for future mobile app)
Spatie Permission	Role-based access control (RBAC)
Inertia.js 2.x (Server-side)	Server-driven single-page application

Frontend

Technology	Purpose
React 18	UI library
Inertia.js 2.x (Client-side)	SPA routing without a separate API
Tailwind CSS 3.x	Utility-first styling
Framer Motion 12.x	Animations and transitions
Recharts 3.x	Admin dashboard charts
Lucide React	Icon system

Infrastructure & Tools

Technology	Purpose
Vite 7	Frontend build tool
Ziggy	Laravel route helpers in JavaScript
Axios	HTTP client
Laravel Pint	Code formatting (PSR-12)
PHPUnit	Testing framework

3. System Architecture

Architecture Pattern

OneClickHub uses a **monolithic architecture with Inertia.js**, providing the benefits of a single-page application (SPA) without the complexity of a separate frontend API layer.

```

Browser (React SPA)
|
v
Inertia.js Protocol (XHR with JSON props)
|
v
Laravel Controllers (return Inertia::render)
|
v
Eloquent ORM --> MySQL Database
|
v
Service Layer (Payment Gateways)
|
v
External APIs (Bayarcash, SenangPay, OpenAI)

```

Backend Structure

```

app/
└── Http/
    ├── Controllers/          # 22 controllers
    │   ├── Auth/             # 9 authentication controllers
    │   │   ├── DashboardController.php
    │   │   ├── ServiceController.php
    │   │   ├── OrderController.php
    │   │   ├── PaymentController.php
    │   │   ├── SubscriptionController.php
    │   │   ├── ChatController.php
    │   │   ├── CalendarController.php
    │   │   └── ...
    │   └── Middleware/
    │       └── HandleInertiaRequests.php
    ├── Models/                # 15 Eloquent models
    ├── Services/              # Payment gateway services
    │   ├── BayarcashService.php
    │   └── SenangpayService.php
    ├── Notifications/         # 5 notification classes
    └── Providers/

```

Frontend Structure

```
resources/js/
├── Pages/          # 20+ Inertia page components
│   ├── Auth/        # Login, Register
│   ├── Dashboard/   # Admin, Freelancer, Customer dashboards
│   ├── Services/    # Browse, Create, Edit services
│   ├── Orders/      # Order management
│   ├── Subscribe/   # Subscription checkout
│   ├── Chat/        # Messaging system
│   ├── Calendar/    # Availability management
│   ├── Admin/        # Admin-specific pages
│   ├── Settings/    # Banking, SSM verification
│   └── Payment/     # Payment status pages
└── Components/     # 18 reusable components
└── Layouts/        # AuthenticatedLayout, GuestLayout
└── app.jsx         # Inertia app bootstrap
```

4. User Roles & Access Control

Authentication

- **Primary identifier:** Phone number (not email)
- **Email:** Optional, nullable field
- **Auth package:** Laravel Sanctum (session + future API tokens)
- **RBAC:** Spatie Laravel Permission with 4 roles

Role Definitions

Admin

- Full system access
- Manage users (create, edit, delete, assign roles)
- Manage subscription plans and payment gateways
- Review transactions and revenue analytics
- Verify SSM business documents
- Oversee all orders

Freelancer

- Subscribe to plans to access the platform
- Create and manage service listings (title, description, pricing, images, tags)
- Manage availability calendar (available/blocked dates)
- Receive and process orders (accept, deliver, complete)
- Chat with customers on active orders
- Upload banking details for payouts
- Submit SSM verification documents

Customer

- Browse and search services by category
- Book services on available dates
- Upload payment slips for orders
- Chat with freelancers on active orders
- Leave reviews and ratings after order completion

Access Control Implementation

```

Admin Routes:      middleware(['auth', 'role:Admin'])
Freelancer Routes: middleware(['auth']) + controller-level role checks
Customer Routes:  middleware(['auth']) + controller-level role checks
Public Routes:     No middleware (landing page, service browsing)

```

5. Core Features

5.1 Service Marketplace

Freelancers create service listings with rich details:

- **Title & Description:** Service name and detailed description
- **Category:** Assigned to one of 23 service categories (Catering, Web Design, Digital Marketing, etc.)
- **Pricing:** Price range (from/to) in MYR
- **Delivery Time:** Estimated days to complete
- **Tags:** Searchable keywords (JSON array)
- **Images:** Multiple service images (JSON array)
- **Status:** Active/inactive toggle

Customers can browse services by category, view freelancer profiles, and book available dates.

5.2 Subscription System

Freelancers must subscribe to a plan to list services on the platform.

Available Plans (Seeded):

Plan	Price	Duration	Target
Starter Hub	RM 49/month	30 days	New freelancers
Premium Pro	RM 199/month	30 days	Established professionals
Enterprise	RM 999/year	365 days	Agencies and teams

Subscription Lifecycle:

```

pending --> active (on successful payment)
active --> cancelled (user/admin cancels; usable until ends_at)
active/cancelled --> expired (when ends_at passes)
  
```

Plan Features are stored as a JSON array, allowing flexible feature sets per plan.

5.3 Order Management

Orders follow a multi-step workflow with payment verification:

```

pending_payment
    --> pending_approval (customer uploads payment slip)
        --> active (freelancer accepts)
            --> delivered (freelancer marks delivered)
                --> completed (customer confirms)
  
```

Orders can be cancelled or rejected at pending stages. Each order has:

- Unique order number: OCH-YYYYMMDD-RANDOM6
- Linked to a specific service and booking date
- Unique constraint on (freelancer_id, booking_date) prevents double-booking
- Customer notes, agreed price, and payment slip upload

5.4 Chat System

- **Order-linked conversations:** Each order creates a chat between customer and freelancer
- **Message features:** Text messages with optional attachments
- **Read receipts:** Messages marked as read on retrieval
- **Polling-based:** Frontend polls for new messages
- **Auto-lock:** Completed orders disable further messaging

5.5 Availability Calendar

Freelancers manage their availability:

- Set dates as **available** or **blocked**
- Booked dates (from active orders) automatically show as unavailable
- Calendar view for easy date management
- Customers see only available dates when booking

5.6 SSM Business Verification

Malaysian business registration (SSM) verification with AI assistance:

- Freelancer uploads SSM document (image/PDF)
- **OpenAI Vision API** extracts: company name, registration number, expiry date
- Admin reviews and approves/rejects verification
- **30-day grace period** after first subscription for new freelancers
- Services automatically hidden after grace period expires if not verified
- Notifications sent at key milestones (reminder, hidden, failed)

5.7 Admin Dashboard

Comprehensive analytics dashboard with:

- **Stats cards:** Total freelancers, customers, revenue, orders, active subscriptions
- **7-day charts:** Order count and revenue trends (Recharts)
- **Top 5 freelancers:** By earnings with average ratings
- **Category breakdown:** Service distribution across categories
- **Transaction management:** Search, filter, and manual status updates
- **User management:** Create, edit, delete users with role assignment

6. Database Schema

Entity Overview (15 Models)

Model	Table	Description
User	users	All platform users with phone auth
Subscription	subscriptions	User subscription records
SubscriptionPlan	subscription_plans	Available plan definitions
Transaction	transactions	Payment transaction logs
PaymentGateway	payment_gateways	Gateway credentials storage
Service	services	Freelancer service listings
ServiceCategory	service_categories	Service type classification
Order	orders	Marketplace orders
Review	reviews	Customer ratings and feedback
BankingDetail	banking_details	Freelancer payout info
FreelancerAvailability	freelancer_availabilities	Calendar availability
ChatConversation	chat_conversations	Order-linked conversations
ChatMessage	chat_messages	Individual messages
SsmVerification	ssm_verifications	Business document verification
AdminSetting	admin_settings	Key-value configuration

Key Relationships

```

User --(has many)--> Services
User --(has many)--> Subscriptions
User --(has many)--> Orders (as customer)
User --(has many)--> Orders (as freelancer)
User --(has one)---> BankingDetail
User --(has one)---> SsmVerification
User --(has many)--> Reviews (as reviewer)

Subscription --(belongs to)--> User
Subscription --(belongs to)--> SubscriptionPlan

Service --(belongs to)--> User (freelancer)
Service --(belongs to)--> ServiceCategory
Service --(has many)----> Orders
Service --(has many)----> Reviews

Order --(belongs to)--> Customer (User)

```

```
Order --(belongs to)--> Freelancer (User)
Order --(belongs to)--> Service
Order --(has one)-----> ChatConversation
Order --(has one)-----> Review

ChatConversation --(has many)--> ChatMessages
ChatConversation --(belongs to)--> Order
```

Key Schema Details

Users Table:

- `phone_number` (unique) — primary login identifier
- `email` (nullable) — optional
- `identity_document` — for verification
- `position` — custom field

Orders Table:

- `order_number` — auto-generated unique identifier
- `status` — enum: pending_payment, pending_approval, active, delivered, completed, cancelled, rejected
- Unique constraint: `(freelancer_id, booking_date)`

Services Table:

- `tags` — JSON array of searchable keywords
- `images` — JSON array of image paths
- `price_from / price_to` — price range

7. Key User Flows

7.1 Freelancer Onboarding

1. Register with phone number
2. Browse subscription plans
3. Select plan and payment gateway
4. Complete payment (Bayarcash/SenangPay)
5. Subscription activated (30-day SSM grace period starts)
6. Upload SSM verification document
7. Create first service listing
8. Set availability calendar
9. Add banking details for payouts

7.2 Customer Booking Flow

1. Register/login with phone number
2. Browse services by category
3. View service details and freelancer profile
4. Select available date and book service
5. Upload payment slip
6. Wait for freelancer approval
7. Chat with freelancer during service delivery
8. Confirm delivery and leave review

7.3 Admin Operations

1. Monitor dashboard analytics
2. Manage subscription plans (CRUD)
3. Configure payment gateways (credentials, status)
4. Review and manage transactions
5. Verify SSM documents (approve/reject)
6. Manage users (create, edit, assign roles)
7. Manage service categories
8. Oversee orders (view status, intervene if needed)

7.4 Payment Flow

1. User initiates subscription payment
2. Controller validates request
3. Service class creates payment intent/URL
4. User redirected to gateway (Bayarcash/SenangPay)
5. User completes payment on gateway
6. Gateway sends callback to server
7. Server validates signature/checksum (prevents tampering)
8. Transaction status updated in database
9. If success: subscription activated, user redirected to success page
10. If failed: user redirected to failure page with retry option

8. Routes & API Overview

Public Routes (No Auth)

Route	Description
GET /	Landing page with plans
GET /services	Browse all services
GET /services/{slug}	View service details
POST /login	Phone-based authentication
POST /register	New user registration

Payment Callback Routes (No Auth, No CSRF)

Route	Description
GET /payment/bayarcash/return	Bayarcash return URL
GET,POST /payment/senangpay/callback	SenangPay callback
POST /api/payment/bayarcash/callback	Bayarcash server callback
GET /payment/success,failed,pending	Payment status pages

Authenticated Routes

Route Group	Key Routes
Subscription	/subscribe, /subscribe/checkout/{plan}, /subscribe/pay
Services	/my-services, /my-services/create, /my-services/{id}/edit
Orders	/orders/{id}, /my-orders, /my-bookings
Chat	/chat, /chat/{conversation}
Calendar	/calendar
Settings	/settings/banking, /settings/ssm-upload

Admin Routes (role:Admin middleware)

Route Group	Key Routes
Users	/users/freelancers, /users/customers, /users/admins
Subscriptions	/subscriptions/plans, /subscriptions/gateways, /subscriptions/transactions
Orders	/admin/orders
Categories	/admin/categories

SSM	/admin/ssm-verifications
------------	--------------------------

API Routes (Sanctum Auth)

Route	Description
GET /api/user	Authenticated user info
GET /api/categories	Service categories list

9. Payment Gateway Integration

Service Layer Pattern

Each gateway has a dedicated service class implementing:

- Payment link/intent creation
- Signature/hash validation (HMAC-SHA256)
- Callback processing
- Transaction status mapping

Bayarcash

- **API Version:** v3
- **Payment Methods:** FPX, DuitNow Online Banking, DuitNow QR, LineCall Express
- **Amount Format:** Cents (multiply by 100)
- **Security:** HMAC-SHA256 checksum validation
- **Phone Format:** Malaysian format (0XX -> 60XX)

SenangPay

- **Integration:** URL-based with SHA256 hash
- **Amount Format:** 2 decimal places
- **Security:** SHA256 hash validation on callback
- **Status Codes:** '1' = Success, '0' = Failed

Gateway Configuration

Payment gateway credentials are stored in the `payment_gateways` table, manageable by admins through the dashboard. Supports toggling gateways active/inactive.

Pre-configured gateways: Bayarcash, SenangPay, PayPal (inactive), Stripe (inactive)

10. Notification System

Notification Classes

Notification	Trigger	Recipient
NewOrderReceived	Customer creates an order	Freelancer
OrderAutoCancel	Order auto-cancellation reminder	Customer
SsmGracePeriodReminder	Grace period nearing expiry	Freelancer
SsmServicesHidden	Services hidden due to expired SSM	Freelancer
SsmVerificationFailed	SSM verification rejected	Freelancer

Delivery Channels

- **Database:** All notifications stored in the `notifications` table
- **Unread count:** Shared via Inertia middleware to all authenticated pages

11. Future Milestones

Phase 1: Mobile Application

- Build React Native mobile app leveraging existing Sanctum API
- Push notifications via Firebase Cloud Messaging
- Native camera integration for SSM document upload
- Mobile-optimized chat interface

Phase 2: Real-Time Communication

- Replace polling-based chat with WebSockets (Laravel Reverb or Pusher)
- Real-time order status notifications
- Live typing indicators in chat
- Instant notification delivery

Phase 3: Expanded Payment Gateways

- Activate PayPal and Stripe gateways (already pre-configured)
- Add GrabPay and Touch 'n Go eWallet support
- Implement recurring subscription billing (auto-renewal)
- Introduce refund processing workflow

Phase 4: Advanced Analytics & Reporting

- Freelancer earnings dashboard with detailed breakdowns
- Customer spending analytics
- Platform-wide revenue reports (monthly, quarterly, annual)
- Exportable reports (CSV, PDF)
- Conversion funnel tracking (browse -> book -> complete)

Phase 5: Multi-Language Support (i18n)

- Bahasa Malaysia as primary additional language
- Language toggle in user settings
- Translated email/notification templates
- RTL support for future expansion

Phase 6: Enhanced Reviews & Trust System

- Photo reviews from customers
- Freelancer response to reviews
- Trust score / badge system based on completion rate and ratings
- Featured freelancer program for top performers

Phase 7: Escrow Payment System

- Hold customer payments in escrow until order completion
- Automated release upon delivery confirmation
- Dispute resolution workflow
- Partial payment and milestone-based releases

Phase 8: Freelancer Portfolio & Branding

- Portfolio gallery with past work showcases
- Custom freelancer profile pages with branding
- Skill endorsements from customers

- Certification and credential display

Phase 9: Advanced Notifications

- Email notifications (Mailgun/SES integration)
- SMS notifications for critical events
- Web push notifications (browser)
- Customizable notification preferences per user

Phase 10: AI-Powered Features

- Smart service recommendations based on browsing history
 - AI-assisted service description writing for freelancers
 - Automated pricing suggestions based on market data
 - Chatbot for common customer queries
 - Fraud detection on payment slips
-

OneClickHub v1.0 — Built with Laravel 12 & React

This document serves as the living blueprint for the OneClickHub platform.

Last updated: February 2026