**Functional Requirement Document   
Project Name**: Opsiana *– Digital Platform for Vessel & Fuel Support*  
**Prepared By**: Mr. Siddhant Kadus *Business Analyst***Stakeholder**: Fernando Alvarez *(Founder, Opsiana)***Date**:   
**Version**: 1.0

**1. INTRODUCTION**

The Opsiana Vessel & Fuel Management Platform is a SaaS-based digital solution designed to streamline the fuel procurement and management process for vessel operators and fuel suppliers. The platform provides an integrated environment where vessel operators can request fuel quotations, compare prices, confirm bookings, and monitor fuel deliveries in real-time.

Additionally, the platform enables fuel suppliers to respond to fuel requests, manage pricing, and maintain compliance with international maritime fuel standards. By leveraging data-driven insights and automated workflows, Opsiana aims to enhance operational efficiency, reduce procurement costs, and improve transparency in the marine fuel supply chain.

This Functional Requirements Document (FRD) defines the detailed functionalities and features required to achieve these objectives. It serves as a reference for development, testing, and implementation teams to ensure that the system meets the intended business and user needs.

**2. SCOPE**

The scope of this project includes the design, development, and deployment of a web-based SaaS platform that supports vessel operators, fuel suppliers, and system administrators.

Key capabilities within scope include:

1. User Management: Registration, authentication, and role-based access control.
2. Fuel Procurement Workflow: Fuel request creation, supplier quotation management, and order confirmation.
3. Price Monitoring: Integration with live marine fuel price APIs for real-time pricing and trend analysis.
4. Analytics & Reporting: Dashboard for cost analysis, supplier performance tracking, and fuel consumption insights.
5. Notification System: Email/SMS alerts for key events such as quotations, confirmations, and delivery updates.
6. Payment Processing: Secure payment gateway integration for transactions and invoicing.

Out of scope for this phase:

1. Mobile application development.
2. Integration with external ERP systems beyond payment gateway.
3. AI-based demand forecasting (planned for future releases).

### ****3. SYSTEM OVERVIEW****

The Opsiana platform is a **cloud-based SaaS application** designed to provide a comprehensive digital ecosystem for vessel operators and fuel suppliers to manage marine fuel procurement efficiently. The system acts as an intermediary platform that enables smooth communication, transparent pricing, and secure transactions within the maritime fuel supply chain.

#### **Key Stakeholders:**

1. **Vessel Operators**: Users who request fuel quotations, compare supplier offers, confirm orders, and track deliveries.
2. **Fuel Suppliers**: Users who respond to fuel requests with competitive quotes, manage inventory availability, and confirm delivery schedules.
3. **System Administrators**: Users who oversee platform operations, manage user roles, monitor transactions, and ensure compliance with security and operational standards.

#### **System Components:**

1. **Web Application (Frontend)**
   * User-friendly interface accessible through standard web browsers.
   * Separate dashboards for vessel operators and fuel suppliers.
   * Role-based navigation and control.
2. **Backend Services**
   * Built on scalable micro services architecture to ensure performance and modularity.
   * Handles business logic, request processing, and workflow automation.
   * API integration with marine fuel pricing services for real-time data.
3. **Database Layer**
   * Centralized database for user profiles, fuel request details, quotations, orders, and transaction records.
   * Supports relational data structure with high availability and security.
4. **Notification & Communication Module**
   * Email and SMS alerts for key system events such as new requests, quotations, and confirmations.
   * In-app messaging for real-time communication between stakeholders.
5. **Analytics & Reporting Engine**
   * Provides dashboards for fuel price trends, supplier performance, and cost analysis.
   * Supports exporting reports in PDF/Excel formats.
6. **Security & Compliance**
   * Implements multi-factor authentication for login.
   * End-to-end encryption for data transmission.
   * GDPR and maritime industry compliance for data privacy and operational standards.

#### **System Behaviour:**

The system follows a **request-quote-confirmation workflow**:

1. A vessel operator creates a fuel request specifying location, quantity, and date.
2. The system notifies fuel suppliers registered for that region.
3. Suppliers provide quotations through the platform.
4. The vessel operator reviews and confirms the best quote.
5. The system updates the order status, triggers delivery notifications, and records the transaction for auditing.

### ****4. ACTORS & ROLES****

The Opsiana platform involves multiple stakeholders interacting with the system to perform specific business functions. Each actor has distinct roles and responsibilities within the system to ensure smooth operations and fulfillment of business objectives.

#### **4.1 Primary Actors**

1. **Vessel Operator**
   * **Role**:
     + Initiates fuel procurement requests.
     + Reviews and compares supplier quotations.
     + Confirms fuel orders and monitors delivery status.
   * **Responsibilities**:
     + Maintain accurate vessel and voyage information.
     + Ensure timely creation of fuel requests with correct specifications.
     + Confirm or reject quotations based on price, quality, and delivery schedule.
   * **Access Level**:
     + Vessel Operator Dashboard (Fuel Requests, Quotations, Orders, Reports).
2. **Fuel Supplier**
   * **Role**:
     + Responds to fuel requests with quotations.
     + Manages fuel inventory availability.
     + Confirms and updates delivery schedules.
   * **Responsibilities**:
     + Provide accurate and competitive pricing.
     + Update availability of products in real-time.
     + Ensure fulfilment of confirmed orders within agreed timelines.
   * **Access Level**:
     + Supplier Dashboard (Quotation Management, Order Management, Reports).

#### **4.2 Supporting Actors**

1. **System Administrator**
   * **Role**:
     + Oversees platform operations and user activities.
     + Manages user accounts, permissions, and security policies.
     + Monitors system health and ensures compliance with operational standards.
   * **Responsibilities**:
     + Perform user onboarding and role assignment.
     + Configure system settings and access controls.
     + Generate system usage and performance reports.
   * **Access Level**:
     + Admin Panel (User Management, System Monitoring, Reports).
2. **Finance & Billing Officer (Optional)**
   * **Role**:
     + Handles invoicing and payment tracking between vessel operators and suppliers.
   * **Responsibilities**:
     + Generate invoices based on confirmed orders.
     + Ensure timely reconciliation of payments.
   * **Access Level**:
     + Finance Module (Billing, Transactions, Payment Status).
3. **Third-Party Services (External Actor)**
   * **Role**:
     + Provides real-time fuel price indices, currency conversion rates, and weather data.
   * **Responsibilities**:
     + Maintain accuracy and uptime of data feeds.
     + Support secure API integration with the Opsiana platform.
   * **Access Level**:
     + API Integration Layer (No direct UI access).

**5. FUNCTIONAL REQUIREMENTS**

The following functional requirements define the core capabilities that the Opsiana system must deliver to meet business objectives. These requirements are categorized based on system functionality.

#### **5.1 User Management**

* **FR-1:** The system shall allow administrators to create, update, and deactivate user accounts.
* **FR-2:** The system shall support role-based access control for Vessel Operators, Fuel Suppliers, and Administrators.
* **FR-3:** The system shall provide a secure login mechanism with multi-factor authentication (MFA).
* **FR-4:** The system shall allow users to reset passwords via email verification.

#### **5.2 Fuel Request Management**

* **FR-5:** The system shall allow Vessel Operators to create a new fuel request by specifying:
  + Vessel name
  + Fuel type (e.g., MGO, VLSFO)
  + Quantity (in metric tons)
  + Delivery date and location
* **FR-6:** The system shall validate the request to ensure all mandatory fields are filled.
* **FR-7:** The system shall notify relevant fuel suppliers when a new fuel request is submitted.

#### **5.3 Quotation Management**

* **FR-8:** The system shall allow Fuel Suppliers to submit quotations against a specific fuel request.
* **FR-9:** The system shall allow Vessel Operators to view, compare, and filter quotations based on:
  + Price
  + Delivery time
  + Supplier rating
* **FR-10:** The system shall allow Vessel Operators to accept or reject a quotation.

#### **5.4 Order Management**

* **FR-11:** The system shall generate an order confirmation once a quotation is accepted.
* **FR-12:** The system shall track order status (Pending, Confirmed, In Progress, Delivered).
* **FR-13:** The system shall notify both the Vessel Operator and Fuel Supplier about any order status changes.
* **FR-14:** The system shall allow modification or cancellation of orders before confirmation.

#### **5.5 Pricing & Currency Conversion**

* **FR-15:** The system shall integrate with third-party APIs to fetch real-time fuel price indices.
* **FR-16:** The system shall support multi-currency quotations and display conversion rates dynamically.

#### **5.6 Notifications & Alerts**

* **FR-17:** The system shall send email and in-app notifications for:
  + New fuel requests
  + New quotations received
  + Quotation acceptance/rejection
  + Order status updates
* **FR-18:** The system shall allow users to configure notification preferences.

#### **5.7 Reporting & Analytics**

* **FR-19:** The system shall generate reports for:
  + Total fuel requests and their status
  + Quotation comparison and savings analysis
  + Supplier performance (delivery time, quality score)
* **FR-20:** The system shall allow users to export reports in PDF and Excel formats.

#### **5.8 Security & Compliance**

* **FR-21:** The system shall encrypt sensitive data (e.g., passwords, financial details) using industry-standard encryption.
* **FR-22:** The system shall comply with GDPR and relevant data protection regulations.
* **FR-23:** The system shall log all user activities for audit purposes.

#### **5.9 System Integration**

* **FR-24:** The system shall integrate with third-party weather services for real-time weather conditions at delivery ports.
* **FR-25:** The system shall support API integration for future enhancements.

### ****6.NON-FUNCTIONAL REQUIREMENTS****

1. **Performance:**
   * Page load time ≤ 2 seconds.
   * Quotation comparison ≤ 3 seconds.
   * Support 10,000 concurrent users.
2. **Availability & Reliability:**
   * 99.9% uptime.
   * Auto-recovery within 60 seconds.
   * Daily data backups.
3. **Scalability:**
   * Handle 100% user growth without major changes.
   * Easy feature integration.
4. **Security:**
   * AES-256 encryption.
   * Role-based access control.
   * GDPR and ISO 27001 compliance.
   * Auto session timeout after 15 mins.
5. **Usability:**
   * Simple and intuitive UI.
   * Mobile responsive.
   * English support initially.
6. **Maintainability:**
   * Easy configuration updates.
   * Proper documentation for future enhancements.
7. **Audit & Logging:**
   * Track all user actions (with timestamps).
   * Generate audit trails.
8. **Compliance:**
   * Adhere to IMO and data protection laws.

**7.ASSUMPTIONS & CONSTRAINTS**

Assumptions

1. Stable Connectivity – Users will have a reliable internet connection to access the platform seamlessly.
2. Accurate Supplier Data – All vessel, fuel, and logistics data provided by suppliers is assumed to be correct and up to date.
3. User Competency – Users (shipping companies, suppliers) will have basic technical knowledge to operate the system without additional training.
4. API Reliability – External APIs for weather updates, fuel pricing, and route optimization will be available and functional during operations.
5. Timeline Adherence – Development, testing, and deployment will follow the approved schedule and scope agreed upon during planning.

Constraints

1. Regulatory Compliance – The solution must comply with international maritime regulations and data protection laws (GDPR).
2. Budget and Timeline – The project must be delivered within the pre-defined budget and timeline without scope expansion.
3. Limited Integration – Only pre-approved third-party APIs and services can be integrated due to security and compliance restrictions.
4. Platform Responsiveness – The system must function effectively on both desktop and mobile devices with responsive design.
5. Cloud Hosting Restriction – All application and database hosting must be on the approved cloud provider (e.g., AWS) as per client policy.

**8.WIREFRAMES & MOCKUPS**

1. Login Screen
2. Dashboard (with KPIs: Fuel Requests, Vessel Status, Ongoing Transactions)
3. Fuel Request Form
4. Vessel Management Screen
5. Supplier Management Screen
6. Transaction History Page
7. Reports & Analytics Page

### ****9.APPROVAL & SIGN-OFF****

This Functional Requirements Document (FRD) for the Opsiana Vessel & Fuel Management System has been reviewed and approved by the following stakeholders. The signatures below indicate agreement with the requirements and confirm that the document accurately reflects the functionality to be developed.

| Name | Role | Organization | Signature | Date |
| --- | --- | --- | --- | --- |
| Fernando Alvarez | Founder / Client | Opsiana | \_\_\_\_\_\_\_\_\_\_\_\_\_ | \_\_\_\_\_\_\_\_\_\_\_\_\_ |
| Siddhant Kadus | Business Analyst | [Your Company] | \_\_\_\_\_\_\_\_\_\_\_\_\_ | \_\_\_\_\_\_\_\_\_\_\_\_\_ |
| [Tech Lead Name] | Technical Lead | [Your Company] | \_\_\_\_\_\_\_\_\_\_\_\_\_ | \_\_\_\_\_\_\_\_\_\_\_\_\_ |
| [QA Lead Name] | QA Lead | [Your Company] | \_\_\_\_\_\_\_\_\_\_\_\_\_ | \_\_\_\_\_\_\_\_\_\_\_\_\_ |