### Software Requirements Specification (SRS) for FindPeace

## 1. **Introduction**

### 1.1 Purpose

The purpose of this Software Requirements Specification (SRS) is to provide a detailed outline for developing **FindPeace**, a web and mobile platform designed to simplify travel and event planning. The platform combines flight, hotel, and car rental bookings with event discovery and management, tailored to customers in **Nigeria and beyond**. The document serves as a comprehensive guide for developers, project managers, stakeholders, and testers to understand and implement the required functionalities.

### 1.2 Product Scope

FindPeace is designed to:

* Provide seamless **booking of flights, hotels, and car rentals**.
* Enable **event discovery and ticket booking**, empowering event organizers to manage and promote their events.
* Support **multi-language (English and local Nigerian languages)** and **multi-currency (NGN as primary)**.
* Deliver a **mobile-responsive user experience** with intuitive navigation.

FindPeace aims to enhance user convenience by integrating travel, accommodation, and event services into one platform, ensuring a smooth and personalized planning experience.

## 2. **Overall Description**

### 2.1 Product Perspective

FindPeace is a **new platform** in the travel and event booking space, combining the functionalities of platforms like **Booking.com** and **Eventbrite**. It features:

1. **Web Application:** Provides a rich, user-friendly interface for customers to search, filter, and book services.
2. **Mobile Application:** Offers on-the-go access to all platform features.
3. **Admin Dashboard:** Enables administrators to manage bookings, events, users, and reports.
4. **Partner Portal:** Allows airlines, hotels, car rental companies, and event providers to manage their listings and track performance.
5. **Backend Services:** Built with **C# and .NET Core**, the backend ensures robust, scalable, and secure functionality.

### 2.2 Product Features

FindPeace includes the following core features:

#### ****2.2.1 Flight Booking****

* Search and filter flights by airline, price, duration, stopovers, and departure/arrival times.
* Compare ticket prices across airlines and display availability.
* Provide ticket booking, cancellation, and management features.

#### ****2.2.2 Hotel Booking****

* Search by **location**, **check-in/check-out dates**, and **room preferences**.
* Filter properties by **price range**, **star ratings**, and amenities such as Wi-Fi, parking, and pools.
* Offer map-based searches for property locations.
* Real-time availability updates and dynamic pricing integration.

#### ****2.2.3 Car Rentals****

* Search rental cars by **pickup/drop-off location**, **vehicle type**, and **price range**.
* Display rental terms, included features, and insurance options.
* Book and manage car rental agreements.

#### ****2.2.4 Event Management****

* Allow **event organizers** to create, manage, and promote events.
* Discover events by category (Music, Conferences, Food & Drinks, etc.), date, and location.
* Support for **free and paid tickets**, with ticketing and capacity management.
* Enable users to favorite events, share them on social media, and receive notifications for updates.

### 2.3 User Classes and Characteristics

1. **Guest Users:** Browse the platform but require registration to book services.
2. **Registered Users:** Access booking history, manage preferences, and favorite listings.
3. **Event Providers:** Manage event listings, ticket sales, and attendee details.
4. **Partners (Hotels, Airlines, Car Rentals):** Create and manage service listings.
5. **System Administrators:** Monitor system health, resolve issues, and generate reports.

## 3. **Specific Requirements**

### 3.1 Functional Requirements

#### ****3.1.1 Flight Booking Module****

* **Search Flights:**
  + Input departure and arrival locations.
  + Set flexible date preferences.
  + Filter by airline, stops, price range, and flight duration.
* **Display Results:**
  + Departure and arrival times.
  + Stopover details.
  + Price and airline name.
* **Manage Bookings:**
  + Allow users to confirm, modify, or cancel flight reservations.

#### ****3.1.2 Hotel Booking Module****

* **Search Options:**
  + Input destination, check-in/out dates, and number of guests.
  + Filter by **price**, **ratings**, and **property type** (e.g., hotels, guesthouses).
* **Display Results:**
  + Property details such as amenities, star ratings, and reviews.
  + Dynamic pricing based on availability and demand.
* **Booking Management:**
  + Allow reservation confirmation and cancellations.

#### ****3.1.3 Car Rental Module****

* **Search Rentals:**
  + Input pickup/drop-off location and dates.
  + Filter by price, vehicle type, and rental terms.
* **Display Options:**
  + Include vehicle specifications (e.g., seating capacity, fuel type).
  + Show rental pricing and insurance details.

#### ****3.1.4 Event Management Module****

* **For Event Organizers:**
  + Create and manage events with details like description, category, pricing, and capacity.
* **For Users:**
  + Search events by location, category, date, and format.
  + Purchase tickets for free and paid events.

### 3.2 Non-Functional Requirements

#### ****3.2.1 Security:****

* Use **SSL encryption** for secure communication.
* Implement **OAuth 2.0** for user authentication.
* Ensure compliance with **GDPR** and **NDPR**.

#### ****3.2.2 Performance:****

* Search results must load in **< 2 seconds**.
* The system must handle **100,000 concurrent users**.

#### ****3.2.3 Scalability:****

* Use **cloud infrastructure** to scale dynamically during peak usage.

## 4. **Database Design**

### 4.1 Core Entities

* **Users:** Stores user profiles, login details, and preferences.
* **Bookings:** Tracks reservations for flights, hotels, and car rentals.
* **Events:** Stores event details like date, category, and organizer information.
* **Payments:** Manages transactions and payment statuses.

### 4.2 Relationships

* **Users ↔ Bookings:** 1-to-many relationship.
* **Event Providers ↔ Events:** 1-to-many relationship.

## 5. **System Architecture**

### 5.1 Tech Stack

* **Frontend:** React.js for web, React Native for mobile apps.
* **Backend:** C# and ASP.NET Core for RESTful API development.
* **Database:** SQL SERVER for relational data storage.

### 5.2 External Integrations

* **Payment Gateways:** Paystack, Flutterwave.
* **Mapping Services:** Google Maps API for location searches.
* **Notification Services:** Email and SMS notifications.

## 6. **Implementation Roadmap**

### ****Phase 1: MVP Development****

* Core search and booking features for flights, hotels, and cars.
* Basic event discovery functionality.
* User authentication and partner management.

### ****Phase 2: Advanced Features****

* Event ticketing and management.
* Payment processing and booking confirmation.
* Admin dashboard with reports and analytics.

### ****Phase 3: Enhancements****

* Multi-language support.
* Push notifications and reminders.
* Search optimization with caching.