



Software Requirements Specification

OCTOBER4, 2025

Egypt tourism activities project

Prepared for
DepiR3
Supervised By
Engineer Mohamed Bayoumi

Prepared by
Loay Ehab | Full Stack Developer
Khaled Waleed | Full Stack Developer
Amr Khaled | Full Stack Developer
Abdullah Sabry | Full Stack Developer



Overview & Software Requirements Specification (SRS)

1. Introduction

a) Purpose

The purpose of the Tourists platform is to create a centralized, trustworthy, and user-friendly online marketplace for tourists to discover and book unique entertainment experiences in Egypt. The system aims to empower local hosts by providing them with a platform to showcase their offerings, manage bookings, and receive payments securely. For tourists, it will serve as a reliable guide to authentic Egyptian experiences beyond traditional sightseeing.

b). Project Scope

The scope of this project is the development of a web-based platform and a corresponding mobile-responsive site

Key features within the scope are:

- **Host Functionality:** Experience creation (listing), schedule management, and booking management.
- **Tourist Functionality:** Browsing and searching for experiences, booking, secure online payments, and writing reviews.
- **Administrative Functionality:** User management (Hosts and Tourists), content moderation, and basic reporting.

Features explicitly out of scope for the initial release are:

- Flight and hotel booking.
- Transportation or vehicle rental services.
- A dedicated mobile application (though the website will be mobile-friendly).
- Multi-language support beyond English and Arabic.

c) Glossary and Abbreviations

Term/Abbreviation	Definition
Host	A registered and verified user who creates and offers experiences.
Guest	A registered user who books experiences.

Experience	A specific activity, tour, or event offered by a Host (e.g., Nile cruise, cooking class).
Booking	A confirmed reservation for an Experience made by a Tourist.
Platform	The Tourists web-based application.
Admin	System manager ensuring compliance , quality and Authenticity of documents

d) List of the System Stakeholders

- **Tourists:** End-users booking experiences.
- **Experience Hosts:** Local guides, artists, and business owners offering experiences.
- **Platform Administrator:** Manages the system, moderates content, and oversees operations.
- **System Developer/Maintainer:** The technical team responsible for building and maintaining the platform.
- **Egyptian Tourism Authority:** An external stakeholder whose regulations may impact platform policies

2. Functional Requirements

a) User Requirements Specification

These requirements describe the goals of the users in **natural language**.

ID	User Requirement	Rationale
UR-01	As a Tourist, I want to search for experiences based on location, date, and category so I can find activities that fit my travel plans.	This is the primary way users will discover relevant experiences, making it a critical feature for usability.
UR-02	As a Host, I want to create a detailed listing for my experience, including photos, a description, price, and available dates.	Hosts need to effectively market their services to attract tourists. A comprehensive listing is essential for this.
UR-03	As a Tourist, I want to securely pay for my booking online using a credit card.	Secure and convenient online payment is a standard expectation for any e-commerce platform and builds user trust.
UR-04	As a Host, I want to be able to approve or decline booking requests and manage my calendar of availability.	This gives Hosts control over their schedule and prevents overbooking, which is crucial for their business operations.
UR-05	As a platform Administrator, I need to verify the identity of new Hosts before they can publish an experience.	This "Know Your Customer" (KYC) process is vital for ensuring the safety and security of tourists and maintaining the platform's reputation.

b) System Requirements Specification

ID	Requirement	Type	Input	Output	Rationale
FR1	User Authentication	Functional	Credentials	Access granted/denied	Ensures secure access
FR2	Listing Management	Functional	Property details	Listing added/edited	Allows host control
FR3	Search Engine	Functional	Filters, keywords	Matching listings	Improves user experience
FR4	Booking Process	Functional	Booking details	Confirmation, payment receipt	Core system functionality
FR5	Messaging System	Functional	Text input	Message sent	Facilitates communication
FR6	Review System	Functional	Ratings, comments	Review added	Builds platform trust
FR7	Admin Dashboard	Functional	Management commands	Reports, controls	Ensures system health

c) Requirements Priorities (MoSCoW Scheme)

Priority	Description	Example Requirements
Must Have	Critical core functions	FR1, FR2, FR3, FR4
Should Have	Important but not critical	FR6
Could Have	Enhancements	Social sharing, wishlist
Won't Have (for now)	Deferred	FR5, Features explicitly out of scope, such as integration with flight booking systems

3. Non-functional Requirements (NFRs)

a) The General Categories of Non-Functional Requirements

- **Performance:** Responsiveness, speed, and scalability of the platform.
- **Security:** Protection of user data and secure transactions.
- **Usability:** Ease of use and learnability for both Tourists and Hosts.
- **Reliability:** System availability and uptime.

ID	Description	Category	Rationale
NFR1	System should handle 1000 concurrent users	Performance	Expected high user traffic
NFR2	Average response time < 3 seconds	Performance	Smooth UX
NFR3	System availability ≥ 99.5%	Reliability	Continuous operation
NFR4	User data encrypted (AES-256)	Security	Protect sensitive info

NFR5	UI should follow accessibility standards	Usability	Inclusive design
NFR6	Codebase modular and documented	Maintainability	Easier future updates
NFR7	System horizontally scalable	Scalability	Growth readiness

4. Design & Implementation Constraints

Constraint	Description / Rationale
Language / Framework	ASP.NET MVC (.NET 8) for server-side web application and Razor views. Rationale: strong support for MVC pattern, tight integration with Visual Studio, mature ecosystem.
API Layer	ASP.NET Web API / minimal APIs to expose RESTful endpoints for mobile/web clients. Rationale: clean separation between UI and API, easier to support mobile apps and third-party integrations.
ORM / Data Access	Entity Framework Core (EF Core) for ORM; use code-first or database-first as appropriate. Rationale: simplifies data access, migrations, and mapping between domain models and SQL Server.
Database	SQL Server (Azure SQL or on-premises SQL Server) . Rationale: ACID-compliant relational DB, proven for transactional workloads like bookings/payments.
Authentication / Authorization	ASP.NET Identity + JWT tokens for API clients; role-based authorization (Guest, Host, Admin). Rationale: standard, secure, integrates with EF Core.
Caching	Redis (Azure Cache for Redis or self-hosted) for session data, read-heavy objects (search cache). Rationale: improves performance and scalability.
Payment Integration	Stripe/PayPal SDKs (.NET) . Rationale: use official .NET SDKs to handle PCI concerns and callbacks.
Hosting / Deployment	Azure App Service / IIS / Docker on AKS . Rationale: supports .NET apps, scaling, CI/CD pipelines.
CI/CD	Azure DevOps / GitHub Actions for build/test/deploy. Rationale: automates testing and deployment.
Logging & Monitoring	Serilog / Application Insights . Rationale: structured logs and telemetry for troubleshooting.
Regulatory	GDPR, PCI-DSS considerations for storing user/payment info.

5. System Evolution

a) Anticipated Changes

- Integration with smart locks or IoT check-in systems.
- Addition of AI-based price recommendations.
- Multi-language support for global users.

b) Design Adaptation

- Use modular microservices for easy feature addition.
- Keep a scalable DB schema for new tables or entities.
- Maintain version-controlled APIs for safe evolution.

