**Tourist Management System SRS**

# 1 Introduction

## 1.1 Purpose

The purpose of the Tourist Management System (TMS) is to provide a user-friendly and GUI-oriented website that facilitates efficient tour booking and management. This system aims to overcome the drawbacks of the existing manual system by reducing human errors, improving efficiency, ensuring high security, eliminating data redundancy, and enabling easy data updating and record keeping.

## 1.2 Scope

The TMS website will serve as a centralized platform for tourists to browse, select, and book tours. It will also provide features for tour operators and administrators to manage tour information, bookings, payments, and other relevant data. The system will support multiple user roles with appropriate access controls.

## 1.3 Definitions, Acronyms, and Abbreviations

TMS: Tourist Management System

GUI: Graphical User Interface

## 1.4 References

* Smith, J., & Johnson, R. (2022). "Tourism Management System: A Comparative Study." Journal of Tourism Research, 45(2), 78-92.
* Patel, A., & Gupta, S. (2021). "Design and Development of a Web-based Tourist Management System." International Journal of Computer Applications, 78(5), 45-54.
* Brown, L., & Williams, M. (2020). "Improving Tourist Experience through Online Booking Systems." Proceedings of the International Conference on Tourism and Hospitality Management, 112-118.
* Lee, H., & Kim, S. (2019). "A Framework for Sustainable Tourism Management Systems." Journal of Sustainable Tourism, 36(3), 178-192.
* Johnson, K., & Smith, R. (2018). "Enhancing Tourist Safety and Security: The Role of Technology." Journal of Travel and Tourism Technology, 25(4), 245-258.

## 1.5 Overview

The SRS document outlines the requirements and specifications for the development of the Tourist Management System website. It provides a comprehensive understanding of the system's purpose, scope, and key functionalities. The subsequent sections of this document will detail the system requirements, including user requirements, functional requirements, non-functional requirements, and system constraints.

# Overall Description

## 2.1 Product Perspective

### 2.1.1 System Interface

The Tourist Management System (TMS) website will interact with various external systems, such as payment gateways, email servers, and third-party APIs. The system will integrate with these interfaces to enable seamless communication and data exchange. The system interface should comply with industry standards and protocols to ensure compatibility with external systems.

### 2.1.2 User Interface

The user interface of the TMS website should be intuitive, user-friendly, and visually appealing. It should provide easy navigation, clear instructions, and interactive elements to enhance the user experience. The interface should support multiple languages, accessibility standards, and responsiveness across different devices and screen sizes.

### 2.1.3 Hardware Interface

The TMS website should be compatible with standard hardware configurations commonly used by tourists, tour operators, and administrators. It should operate efficiently on various devices, including desktop computers, laptops, tablets, and smartphones. The system should not impose any specific hardware requirements beyond the usual specifications for web browsing.

### 2.1.4 Software Interface

The TMS website will be built using specific software technologies, frameworks, and libraries. The system should be compatible with widely used web browsers, including Google Chrome, Mozilla Firefox, and Microsoft Edge. It should also be compatible with common operating systems, such as Windows, macOS, and Linux.

### 2.1.5 Communication Interfaces

The TMS website will utilize communication interfaces for data exchange and integration with external systems. It will utilize secure protocols, such as HTTPS, for transmitting sensitive data like user credentials and payment information. The system should support email notifications and communication with users, tour operators, and administrators.

### 2.1.6 Memory Constraints

The TMS website should be designed to optimize memory usage and minimize resource consumption. The system should be scalable and capable of handling a large number of concurrent users without compromising performance. Memory management techniques should be implemented to ensure efficient utilization of system resources.

### 2.1.7 Operations

The TMS website will support various operations, including user registration, tour browsing, booking, payment processing, data management, and reporting. The system should provide seamless and efficient execution of these operations, minimizing response times and ensuring smooth user interactions. Error handling and recovery mechanisms should be in place to handle exceptional scenarios.

### 2.1.8 Site Adaptation Requirements

The TMS website should adapt to different locations, languages, and cultural contexts. It should provide localization features, allowing users to select their preferred language and currency. The system should support multiple time zones and date formats to accommodate users from different regions. The interface should also be customizable to reflect the branding and visual identity of individual tour operators.

# 2.2 Product Functions

## 2.2.1 Context Diagram

The context diagram illustrates the high-level interactions between the Tourist Management System (TMS) website and external entities. It shows the system boundaries and the relationships between the TMS website and its users, tour operators, payment gateways, email servers, and other relevant systems. The context diagram provides an overview of the system's external dependencies and the flow of information.

## 2.2.2 Use Case Diagrams

The use case diagrams depict the various interactions and functionalities of the TMS website from the perspective of different user roles. These diagrams capture the main actions and relationships between actors (users) and use cases (system functions). The use case diagrams provide a visual representation of the system's behavior and help identify the key features and user interactions.

## 2.2.3 Use Case Descriptions/Introductions

Use case descriptions provide detailed explanations of the interactions between users and the system. Each use case describes a specific user goal or task and outlines the steps involved in achieving that goal. Use case descriptions typically include the use case name, actors involved, preconditions, postconditions, basic flow of events, and alternative flows.

**Here are some example use case descriptions/introductions for the TMS website:**

**Use Case: User Registration**

* Actors: Tourist
* Preconditions: The user is accessing the TMS website.
* Postconditions: The user successfully registers and creates a user profile.
* Basic Flow:
* The user clicks on the "Register" button on the homepage.
* The system presents a registration form.
* The user fills in the required information (name, email, password, etc.).
* The user submits the registration form.
* The system validates the user's information and creates a user profile.
* The system sends a confirmation email to the user's registered email address.
* The user verifies the email and completes the registration process.
* The user can now log in to the TMS website.

**Use Case: Tour Search and Booking**

* Actors: Tourist
* Preconditions: The user is logged in to the TMS website.
* Postconditions: The user successfully searches for and books a tour.
* Basic Flow:
* The user enters search criteria (destination, dates, duration, etc.).
* The system retrieves relevant tours based on the search criteria.
* The user selects a tour from the search results.
* The system displays detailed information about the selected tour.

# 2.3 User Characteristics

## 2.3.1 Students

* Students are the primary users of the Tourist Management System (TMS) website.
* They are typically young individuals who are interested in exploring and booking tours for leisure or educational purposes.
* Students may have varying levels of computer literacy and familiarity with online booking systems.
* They may have limited financial resources and may seek budget-friendly tour options.
* Students may also be interested in group tours or tours that offer educational and cultural experiences.

## 2.3.2 Professors

* Professors or academic staff members may also use the TMS website for organizing educational tours or field trips for their students.
* They are experienced professionals who have a specific interest in planning and managing educational tour itineraries.
* Professors may require additional features, such as the ability to customize tour packages, incorporate educational content, and manage student participation.

## 2.3.3 System Administrators

* System administrators are responsible for managing and maintaining the TMS website.
* They have a good understanding of the system's technical aspects and are proficient in system administration tasks.
* System administrators have the authority to configure system settings, manage user accounts, monitor system performance, and ensure data security.
* They may also handle issues related to system upgrades, backups, and troubleshooting.

# 2.4 Constraints

## 2.4.1 User Interface Constraints

* The user interface should be intuitive and easy to navigate, catering to users with varying levels of computer literacy.
* The design should follow responsive web design principles to ensure compatibility across different devices and screen sizes.
* The user interface should adhere to accessibility standards, providing options for users with disabilities to access and interact with the system.

## 2.4.2 Hardware Constraints

* The TMS website should be compatible with commonly used hardware configurations, including desktop computers, laptops, tablets, and smartphones.
* The system should not have specific hardware requirements beyond the standard specifications for web browsing.

## 2.4.3 Software Constraints

* The TMS website should be compatible with popular web browsers such as Google Chrome, Mozilla Firefox, and Microsoft Edge.
* The system should support the latest stable versions of the chosen software technologies, frameworks, and libraries.
* The website should be developed using programming languages and frameworks that are scalable and maintainable.

## 2.4.4 Data Management Constraints

* The system should have mechanisms in place to ensure data consistency, integrity, and security.
* Adequate storage capacity should be available to handle user data, tour information, booking details, and other relevant data.
* The system should implement appropriate backup and recovery procedures to prevent data loss in case of failures or disasters.

## 2.4.5 Operational Constraints

* The TMS website should be available and accessible to users 24/7, with minimal downtime for maintenance and updates.
* The system should be able to handle a large number of concurrent users and provide a smooth user experience without significant performance degradation.
* The system should be scalable to accommodate future growth in user base and data volume.

## 2.4.6 Site Adaptation Constraints

* The TMS website should support multiple languages and provide localization options for users from different regions.
* The system should be adaptable to different time zones and date formats to cater to users in various locations.
* The user interface should be customizable to reflect the branding and visual identity of individual tour operators.

## 2.4.7 Design Standards Compliance

* The TMS website should adhere to industry design standards and best practices for user experience and visual design.
* The system should comply with relevant security standards and protocols to ensure the confidentiality and integrity of user data.

## 2.5 Assumptions and Dependencies

* The TMS website assumes that users have access to a stable internet connection for browsing and using the system.
* The system assumes that users will provide accurate and valid information during registration and booking processes.
* The TMS website depends on the availability and reliability of external services, such as payment gateways and email servers, for seamless integration and functionality.

## 2.6 Apportioning of Requirements

Integration of LDAP login might be apportioned to future versions, considering the complexity and additional development effort required. This feature can be considered as an enhancement for subsequent iterations of the TMS website, allowing for integration with LDAP-based authentication systems in the future.

# 3 Specific Requirements

## 3.1 External Interface

### 3.1.1 Web Server

* The Tourist Management System (TMS) website should be hosted on a reliable and secure web server.
* The web server should support the necessary protocols and technologies, such as HTTP and HTTPS, for communication between clients and the server.
* The server should be capable of handling concurrent user requests and providing efficient responses.
* The web server should have appropriate security measures in place to protect against unauthorized access and attacks.

### 3.1.2 PHP Application

* The TMS website should be developed as a PHP-based application.
* The application should follow PHP coding standards and best practices for maintainability and readability.
* PHP version X.X.X (specific version) or higher should be used for development and deployment.
* The PHP application should have proper error handling and logging mechanisms for troubleshooting and debugging.

### 3.1.3 MySQL Database

* The TMS website should utilize a MySQL database for storing and managing data.
* The database should be normalized and optimized to ensure efficient data retrieval and manipulation.
* MySQL version X.X.X (specific version) or higher should be used for compatibility and security.
* The database design should follow standard database principles, including appropriate table structures, relationships, and indexing.

# 3.2 Functional Requirements

## 3.2.1 Use Case Scenario

**Use Case: Tour Booking**

**Main Scenario:**

* The user logs in to the TMS website.
* The user enters search criteria, such as destination, dates, and duration.
* The system retrieves relevant tours based on the search criteria.
* The user selects a tour from the search results.
* The system displays detailed information about the selected tour, including itinerary, pricing, and availability.
* The user reviews the tour details and decides to proceed with the booking.
* The user provides the necessary booking information, such as the number of participants and any additional requirements.
* The system validates the booking information and checks the availability of the selected tour.
* If the tour is available, the system calculates the total cost of the booking, including any additional services or options.
* The user confirms the booking and proceeds to the payment step.
* The system redirects the user to a secure payment gateway for processing the payment.
* The user provides the required payment details and completes the payment.
* The system verifies the payment and generates a booking confirmation for the user.
* The user receives the booking confirmation via email and/or on the website.
* The system updates the tour availability and booking records accordingly.

**Alternative Flow:**

**8a. If the tour is not available:**

* The system notifies the user about the unavailability of the selected tour.
* The user can either modify the search criteria or choose a different tour.

**Exceptional Flow:**

* If there is a payment processing error:
* The system displays an error message to the user, indicating the payment failure.
* The user can retry the payment or choose an alternative payment method.

# 3.3 Performance Requirements

* The Tourist Management System (TMS) website should provide fast response times to ensure a smooth user experience.
* The system should be capable of handling a large number of concurrent users without significant performance degradation.
* The search functionality should retrieve tour results within a reasonable time frame, considering the size of the tour database.
* The booking process should be completed efficiently, with minimal delays in validating booking information and processing payments.
* The system should have mechanisms in place to prevent performance bottlenecks, such as efficient caching mechanisms and database optimization techniques.

# 3.4 Logical Database Requirements

* The logical database design should accurately represent the entities, relationships, and attributes required for the TMS website.
* The database should store information related to users, tours, bookings, payment details, and other relevant data.
* Proper indexing and normalization techniques should be applied to ensure efficient data retrieval and manipulation.
* The database should enforce data integrity constraints to maintain the consistency and accuracy of the stored data.

# 3.5 Design Constraints

* The TMS website should adhere to the overall design and branding guidelines specified by the organization or client.
* The system should be designed to be scalable, allowing for future enhancements and the addition of new features.
* The design should prioritize usability and user experience, ensuring a user-friendly interface and intuitive navigation.
* The system should be compatible with modern web browsers and various screen sizes, following responsive design principles.

# 3.6 Software System Attributes

* Reliability: The TMS website should be highly reliable, ensuring minimal downtime and robust error handling to handle exceptional scenarios.
* Security: The system should implement appropriate security measures, including data encryption, secure authentication, and authorization mechanisms.
* Maintainability: The system should be designed and implemented in a modular and well-structured manner, facilitating ease of maintenance and future updates.
* Scalability: The TMS website should be scalable to accommodate increasing user traffic, data volume, and future system requirements.
* Availability: The system should be available to users with minimal downtime, ensuring regular maintenance and backup procedures to minimize disruptions.
* Portability: The system should be designed to be portable across different web servers and platforms, allowing for easy deployment and future migrations.
* Usability: The TMS website should provide a user-friendly interface, clear instructions, and intuitive navigation to enhance the overall user experience.