

**Faculty of Computer Science and Artificial  
Intelligence, Helwan University**

**Academic Year:** 2025 – 2026

**Department:** Software Engineering

**Subject:** Software Engineering – 2

**Software Requirement Specification for  
“Kheyma: Camping Reservation & Package Selection  
Website”**

**Submitted By:**

<b>Mohammed Raafat</b>	<b>ID: 20235102</b>
<b>Yehia Mohammed</b>	<b>ID: 20235131</b>
<b>Youssef Mohammed</b>	<b>ID: 20235136</b>
<b>Youssef Kamal</b>	<b>ID: 20235135</b>
<b>Ahmed Mohammed</b>	<b>ID: 20235009</b>
<b>Youssef Mohammed</b>	<b>ID: 20235138</b>

**Under the Supervision of:**

**Dr. Salwa Osama**

**Omar Tarek**

**Date: 10/12/2024**

# Table of Contents

1. Introduction .....	3
1.1 Purpose .....	3
1.2 Document Conventions .....	3
1.3 Intended Audience and Reading Suggestions .....	3
1.4 Project Scope .....	3
2. Overall Description .....	3
2.1 Product Perspective .....	3
2.2 Product Features .....	4
2.3 User Classes and Characteristics.....	4
2.4 Operating Environment .....	5
2.5 Design and Implementation Constraints .....	6
2.6 User Documentation .....	6
2.7 Assumptions and Dependencies .....	6
3. System Features.....	7
3.1 Listing Campsites and Hiking Spots .....	7
3.2 Choosing a location .....	7
3.3 User account and Authentication .....	8
3.4 Secure online payment .....	8
4. External Interface Requirements.....	8
4.1 User Interfaces .....	8
4.2 Hardware Interfaces.....	9
4.3 Software Interfaces .....	9
4.4 Communications Interfaces .....	9

# 1. Introduction

## 1.1 Purpose

The purpose of this Software Requirements Specification (SRS) is to define the complete functional and non-functional requirements of Kheyma, an end-to-end web platform designed to help users browse, compare, and reserve camping locations across Egypt. The system supports package selection, secure online payments, and administrative management tools for campsite owners and platform operators. This SRS describes the full system and outlines the complete scope, features, constraints, and interfaces required for its implementation.

## 1.2 Document Conventions

This document follows standard software engineering conventions for clarity and consistency:

- Terminology Capitalization: User, Registered User, Admin, and System are capitalized as defined roles.
- Requirement Identification Format: Functional requirements are numbered using the format (3.1-1), where the first two digits indicate the major feature and sub-category, and the last digit indicates the sequential requirement within that sub-category.
- Priority Levels: Requirements use High, Medium, and Low priority tags.
- Style and Structure: Headings follow the IEEE SRS organizational structure, ensuring consistency and readability across sections.

## 1.3 Intended Audience and Reading Suggestions

### This SRS is intended for:

- Professors and Teaching Assistants evaluating the project.
- Developers and future maintainers implementing or upgrading the system.
- System designers and testers validating features and behaviors.

Readers should follow the document in its standard IEEE order from 1.1 to 4.4, beginning with the introduction, then the overall description, system features, and interface requirements.

## 1.4 Project Scope

Kheyma is a web-based camping reservation and package selection system that aims to:

- Make campsite booking easier by centralizing camping destinations across Egypt.
- Enable package comparison with clear pricing, feature breakdowns, and included amenities.
- Provide an Admin control panel for managing sites, packages, bookings, and user accounts.

The system includes secure payment processing, ticketless booking options for selected locations, and user profile management. Future recommendations may consist of a mobile application or integration into a larger tourism ecosystem.

# 2. Overall Description

## 2.1 Product Perspective

The Camping Reservation Website is a new, self-contained web application designed to help users search for camping locations, view available packages, and complete reservations. The system operates as an independent application but is designed to integrate easily with optional external services such as online payment gateways, third-party map providers (e.g., Google Maps), and email notification systems. If extended in the future, it may also become

part of a larger ecosystem that includes a mobile app or a partner campsite management system.

## 2.2 Product Features

The Camping Reservation Website offers a range of core features that enable users to easily explore camping locations, compare available packages, and complete bookings. At a high level, the product includes the following major functional groups:

### 1. Browsing Camping Site

- Allow users to view a catalog of camping locations.
- Displays site details such as photos, facilities, maps, and user ratings.
- Supports filtering and searching by location, price, or amenities.

### 2. Package Selection

- Offers multiple package options for each campsite (e.g., Basic, Standard, Premium).
- Presents package descriptions, pricing, and included services.
- Enables users to compare different packages before booking.

### 3. Booking and Reservation System

- Allow registered users to select dates, choose a package, and reserve a campsite.
- Calculate total pricing based on package type and stay duration.
- Provide booking confirmation and a history of past reservations.

### 4. User Account Management

- Allows users to create accounts, log in, and update their profiles.
- Enables users to save favorite campsites and write reviews.

### 5. Reviews and Ratings

- Registered users can leave reviews and ratings for campsites.
- Displays average ratings to help guide user decisions.

### 6. Admin Management Tools

- Admins can add, edit, or remove camping sites and packages.
- Provides tools for managing bookings and user accounts.
- Ensure system content remains accurate and up to date.

## 2.3 User Classes and Characteristics

The Camping Reservation Website will be used by several distinct user groups, each interacting with the system differently based on their roles, privileges, and usage frequency. The main user classes are described below:

### 1. Guest Users

- Description:  
Individuals visiting the website without creating an account.
- Characteristics:  
Low frequency of use.  
No technical expertise required.  
Access only to basic browsing features.  
Cannot book campsites or write reviews.
- Primary Needs:  
Browse camping sites.  
View packages and general information.
- Importance:  
Moderately important, as they represent potential future customers.

### 2. Registered Users

- Description:  
Users who create an account to access full functionality.

- Characteristics:
  - Medium to high frequency of use.
  - Basic technical knowledge (typical web user).
  - Can make reservations, leave reviews, and save favorites.
  - Require secure data handling and account management features.
- Primary Needs:
  - Make bookings.
  - Compare packages.
  - Manage reservations.
  - Write and view reviews.
- Importance:
  - Highly important, as they are the main target users of the system.

### 3. Admin Users

- Description:
  - Staff responsible for managing content, users, and system operations.
- Characteristics:
  - High frequency of use.
  - Higher technical knowledge than regular users.
  - Full access privileges.
  - Responsible for ensuring data accuracy and system integrity.
  - Require efficient and secure management tools.
- Primary Needs:
  - Add, edit, or remove camping sites.
  - Update packages and pricing.
  - Manage user accounts and bookings.
  - Monitor system activity.
- Importance:
  - Critical, as they maintain the system and ensure smooth operation

## 2.4 Operating Environment

The Camping Reservation Website will operate within a modern web-based environment and must be compatible with commonly used hardware and software configurations. The system is designed to function smoothly across various devices and platforms.

- Hardware Platforms
- Desktop computers and laptops.
- Tablets.
- Smartphones.
- Servers hosting the backend and database.
- Supported Operating Systems.

### For end users (web browsers):

- Windows 10 and above.
- macOS (latest three major versions).
- Linux distributions (Ubuntu, Fedora, etc.).
- Android (Chrome, Samsung Internet).
- iOS/iPad iOS (Safari, Chrome).

### For the development team (Software Dependencies)

The system requires the following components to operate:

- Backend Framework: Java Spring Boot.
- Frontend Stack: HTML, CSS, JavaScript, ReactJS.

- Database Management System: MongoDB.
- Web Server: Nginx or Apache (recommended).
- Map Integration: Google Maps API or similar service.

## 2.5 Design and Implementation Constraints

### **Corporate and Regulatory Policies:**

- The system must comply with internal security policies, data-protection standards, and any applicable national or international regulations governing user data.

### **Hardware Limitations:**

- System performance must operate within defined hardware capabilities, including timing restrictions, CPU utilization limits, and memory constraints specific to the deployment environment.

### **Required Technologies and Tools:**

- Development must use the technologies, frameworks, and database systems mandated by the organization (e.g., Spring Boot, MySQL, Docker), limiting alternative architectural approaches.

### **Security Considerations:**

- The system must adhere to strict authentication, authorization, and encryption requirements. Sensitive data handling must match company-approved security guidelines.
- All user passwords must be stored using encryption.
- Admin pages must require authentication.
- Data must be protected from unauthorized access.

### **Design Conventions and Programming Standards:**

- Development must follow coding conventions, architectural patterns, and documentation standards defined by the customer's IT department, as they will be responsible for long-term maintenance.

## 2.6 User Documentation

The following user documentation components will be delivered with the software:

### **User Manual:**

- A concise guide that explains system features, basic operations, and common tasks. It will be provided in PDF format.

### **Online Help System:**

- Built-in help is accessible from within the application, offering step-by-step instructions and descriptions of interface elements.

### **Quick-Start Tutorial:**

- A short introductory walkthrough designed to help new users understand the main functions of the system quickly. It will be delivered as an interactive in-app tutorial.

### **FAQ Section:**

- A list of common questions and solutions available within the Help menu.

### **Documentation Formats and Standards:**

- The user manual will follow the organization's standard documentation template.
- All documentation will be delivered in PDF and HTML (web-based help) formats.
- Language used will follow the customer's documentation guidelines (English by default).

## 2.7 Assumptions and Dependencies

The project is based on the following assumptions and dependencies:

### **Assumed Availability of Third-Party Components:**

- The system assumes that required external libraries, APIs, or commercial tools will remain available, stable, and compatible throughout development.

#### **Stable Operating Environment:**

- It is assumed that the deployment environment (servers, databases, operating systems, and network infrastructure) will remain consistent and meet the minimum system requirements.

#### **User Access and Permissions:**

- The system assumes that users will have proper access rights and authentication credentials provided by the organization.

#### **Reliable Network Connectivity:**

- The system depends on continuous and stable network access for all online features to function correctly.

#### **Dependency on External Services:**

- The project relies on external systems or services (such as authentication servers, payment gateways, or existing organizational databases). Any change or failure in these services may impact system functionality.

#### **Reusability of Existing Components:**

- The project assumes that certain software modules from previous projects will be available and compatible for reuse unless stated otherwise in other project documents.

### **3. System Features**

#### **3.1 Listing Campsites and Hiking Spots**

##### **3.1.1 Description**

- The website should be able to list available campsites and Hiking spots. The locations will either have tickets available, unavailable, or not required.

##### **3.1.2 Stimulus/Response Sequences**

- The user will enter the name of the location, and a list of matching locations will be returned for the user to choose from. If the user doesn't explicitly enter a location but a "type" of location (Desert, Oasis, Protectorate, etc.), all locations associated with this type will be returned. There should also be sorting options based on popularity and price.

##### **3.1.3 Functional Requirements**

- 3.1-1: A list of locations should be present in the database.
- 3.1-2: The list of locations should be accessible via API.

#### **3.2 Choosing a location**

##### **3.2.1 Description**

- The user should be able to choose a location, whether they should buy a ticket or not. Choosing the location in case there is no ticket should be in the form of saving it in the user's booking history as nothing more than a reminder, since there is no transaction to be made.

##### **3.2.2 Stimulus/Response Sequences**

- If the user clicks on "Book" in case of choosing a location with tickets, the user will be redirected to the payment gateway to their preferred payment option. In case of a location without tickets, the ticket will be saved immediately without overhead. The ticket will be saved in the user's history in the "Upcoming" category and will expire when the expiration date of the ticket coincides with the current date and time, which will place it in the "Past Bookings" category.

##### **3.2.3 Functional Requirements**

- 3.2-1: The user must be able to book a location
- 3.2-2: Multiple payment options should be available
- 3.2-3: The system should be able to detect the expiration of tickets.

### 3.3 User account and Authentication

#### 3.3.1 Description

- The user should be able to create an account along with an interface that contains the user's account details, with the option to edit the user data. The account data includes the following areas:
    - Name.
    - Email.
    - Age.
    - Phone number.
- along with a list that shows prior trips and upcoming ones.

#### 3.3.2 Stimulus/Response Sequences

- There will be a button on the navbar that the user can click to go to the signup/login page.

#### 3.3.3 Functional Requirements

- 3.3-1: the user must be able to view and edit the credentials of their account.
- 3.3-2: the user should be able to see past trips and plans.

### 3.4 Secure online payment

#### 3.4.1 Description

- The users should be able to pay for the tickets easily and securely with minimal input.

#### 3.4.3 Functional Requirements

- 3.4-1: The payment must be secure.
- 3.4-2: There should be multiple payment options (Stripe, PayPal, InstaPay, etc.)

### 3.5 Admin Tools

#### 3.5.1 Description

- A collection of tools for admin use that allow for:
- Managing campsite data.
- Managing users.
- Compiling analytics.

#### 3.5.3 Functional Requirements

- 3.5-1: The admins must be able to do CRUD operations on users.
- 3.5-2: The admins should be able to do CRUD operations on campsites.
- 3.5-3: The admins should be able to view analytics on the dashboard.

## 4. External Interface Requirements

### 4.1 User Interfaces

The camping website will provide a responsive and user-friendly interface accessible on desktops, tablets, and smartphones. The following user interface components will be implemented:

#### Pages:

- Home page: Display featured campsites, promotions, and latest blog posts.
- Campsite Listings: List of available campsites with search and filter options.
- Campsite Details: Detailed view of each campsite, including images, and a booking form.
- Booking Page: Form for users to reserve a campsite.

- User Account: Registration, login, profile management, and booking history.
- Blog Page: Articles and tips about camping.
- Contact Page: Form to send messages and feedback.

#### **Navigation and Common Elements:**

- Persistent navigation bar with links to all major pages.
- Footer containing contact info, social media links, and privacy policy.
- Standard buttons include: "Book Now", "Add to Cart", "Subscribe."

#### **UI Standards:**

- Colors: Green, Brown, Beige to match the camping theme.
- Font: Roboto or Arial across all pages.
- Error messages: Displayed in red below the affected input field.

### **4.2 Hardware Interfaces**

The website will interact with standard hardware devices as follows:

- Supported Devices: Desktop PCs, Laptops, Tablets, Smartphones.
- User Input Methods: Mouse clicks, touchscreen taps, keyboard input.
- Special Hardware Access: GPS and camera devices may be used for submitting campsite locations or uploading images.
- Communication Protocols: HTTP/HTTPS for client-server interaction.

### **4.3 Software Interfaces**

The website will interact with the following software components:

- Web Server: Apache.
- Database: MongoDB for storing user data, campsite details, and bookings.
- Third-party APIs / Services:
  - Payment processing: Stripe or PayPal API.
  - Mapping: Utilizing the Google Maps API for accurate campsite locations.
  - Email: SMTP service for notifications.
- Data Exchanged:
  - Input: User registration details, booking information, search queries.
  - Output: Booking confirmations, payment status, notification emails.
- Implementation Constraints: Data shared between components must be serialized in JSON format and transmitted securely over HTTPS.

### **4.4 Communications Interfaces**

The website will implement the following communications requirements:

- Protocols: HTTP/HTTPS for web traffic; SMTP for email communication.
- Message Formats: JSON format for all API communications.
- Security / Encryption: TLS/SSL for secure data transmission; sensitive user data will be encrypted in the database.
- Data Transfer: Booking and registration data will be transmitted in real-time with server acknowledgement.
- Synchronization: Database transactions will ensure data consistency between user requests, payments, and booking records.