<Name of the project>

Software Requirement Specification

Project Code: <Code of the project>

Document Code: <Code of the document >– v<x.x>

**<Location, issued date of the Document>**

Record of change

\*A - Added M - Modified D - Deleted

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Effective Date | Changed Items | A\* M, D | Change Description | New Version |
| 02/12/2016 | Initial | a | Add project over view |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

SIGNATURE PAGE

**ORIGINATOR:** <Name> <Date>

<Position>

**REVIEWERS:** <Name> <Date>

<Position>

<Name, if it’s needed> <Date>

<Position>

**APPROVAL:** <Name> <Date>

<Position>

TABLE OF CONTENTS

[1 Introduction 5](#_Toc461102219)

[1.1 Purpose 5](#_Toc461102220)

[1.2 Scope 5](#_Toc461102221)

[1.3 Definitions, Acronyms, and Abbreviations 5](#_Toc461102222)

[1.4 References 5](#_Toc461102223)

[1.5 Overview 6](#_Toc461102224)

[2 Overall Description 7](#_Toc461102225)

[3 FUNCTIONAL Requirements 8](#_Toc461102226)

[3.1 <Use Cases Diagram> 8](#_Toc461102227)

[3.2 < Use Case Name 1> 8](#_Toc461102228)

[3.3 < Use Case Name 2> 9](#_Toc461102229)

[4 NON-FUNCTIONAL Requirements 10](#_Toc461102230)

[4.1 Usability 10](#_Toc461102231)

[4.2 Reliability 10](#_Toc461102232)

[4.3 Performance 10](#_Toc461102233)

[4.4 Supportability 11](#_Toc461102234)

[4.5 Design Constraints 11](#_Toc461102235)

[4.6 On-line User Documentation and Help System Requirements 11](#_Toc461102236)

[4.7 Purchased Components 11](#_Toc461102237)

[4.8 Interfaces 12](#_Toc461102238)

[4.9 Licensing Requirements 12](#_Toc461102239)

[4.10 Legal, Copyright, and Other Notices 12](#_Toc461102240)

[4.11 Applicable Standards 12](#_Toc461102241)

[5 Supporting Information 13](#_Toc461102242)

# Introduction

[The introduction of the **Software Requirements Specification (SRS)** provides an overview of the entire **SRS**. It includes the purpose, scope, definitions, acronyms, abbreviations, references, and overview of the **SRS**.]

[Note: The **SRS** document captures the complete software requirements for the system, or a portion of the system. Following is a typical **SRS** outline for a project using only traditional, natural-language style requirements—with **no use-case modeling.** It captures all requirements in a single document, with applicable sections inserted from the Supplementary Specifications (which would no longer be needed). For a template of an **SRS** using use-case modeling, which consists of a package containing Use Cases of the use-case model and applicable Supplementary Specifications and other supporting information, see rup\_srsuc.dot.]

[Many different arrangements of an **SRS** are possible. Refer to [IEEE830-1998] for further elaboration of these explanations, as well as other options for **SRS** organization.]

## Purpose

[Specify the purpose of this SRS. The SRS fully describes the external behavior of the application or subsystem identified. It also describes nonfunctional requirements, design constraints, and other factors necessary to provide a complete and comprehensive description of the requirements for the software.]

The purpose of this Software Requirements Specification (SRS) document is to provide a comprehensive description of the functional and non-functional requirements for the "Feng Shui Koi Fish Consultation System." The system aims to help users select suitable Koi fish and pond characteristics based on their personal destiny according to Feng Shui principles (e.g., Metal, Wood, Water, Fire, Earth). This document serves as a guide for developers, stakeholders, and other parties involved, ensuring a clear understanding of the system's functionalities, scope, and constraints.

## Scope

[A brief description of the software application that the SRS applies to, the feature or other subsystem grouping, what Use-Case model(s) it is associated with, and anything else that is affected or influenced by this document.]

The "Feng Shui Koi Fish Consultation System" will offer a platform for users to receive personalized consultation for selecting Koi fish and designing fish ponds. The system's core features include:

* A homepage featuring information about Feng Shui principles, news, and blog posts.
* A consultation service that advises users on the best Koi fish species, quantities, and pond features (e.g., shape, location, direction) based on their personal destiny.
* A tool for users to check the compatibility of their destiny with their current Koi fish and pond setup.
* Administrative functions for managing Koi fish species and pond characteristics that align with Feng Shui principles.
* An advertisement section for users to promote Koi fish and Feng Shui products related to pond decoration.
* A payment and billing module for consultation services and advertising packages.
* A dashboard providing statistical reports on consultations and advertising revenue.

This document outlines the system’s functionalities, user interactions, and system interfaces to provide a complete understanding of the project’s requirements.

## Definitions, Acronyms, and Abbreviations

[This subsection provides the definitions of all terms, acronyms, and abbreviations required to properly interpret the SRS. This information may be provided by reference to the project’s Glossary.]

1. SRS: Software Requirements Specification
2. Feng Shui: An ancient Chinese practice focusing on the arrangement and orientation of spaces to achieve harmony with the environment.
3. Koi Fish: A type of ornamental fish that symbolizes good luck and prosperity in Feng Shui.
4. Destiny: In Feng Shui, a user's element (Metal, Wood, Water, Fire, Earth) based on their birth year and gender, used to determine compatibility with Koi fish and pond features.
5. Consultation: The process of providing personalized advice on selecting Koi fish and designing ponds based on a user's destiny.

## References

[This subsection provides a complete list of all documents referenced elsewhere in the SRS. Identify each document by title, report number if applicable, date, and publishing organization. Specify the sources from which the references can be obtained. This information may be provided by reference to an appendix or to another document.]

Feng Shui Beginner: Koi Fish Feng Shui

IEEE Standard 830-1998, IEEE Recommended Practice for Software Requirements Specifications.

## Overview

[This subsection describes what the rest of the SRS contains and explains how the document is organized.]

1.5 Overview

**This SRS document is structured to provide a clear and detailed description of the system requirements. It includes the following sections:**

* Introduction: Provides an overview of the SRS, including the purpose, scope, definitions, and references.
* Overall Description: Gives a broad understanding of the system, including its context, user characteristics, and general constraints
* System Features: Details the functionalities that the system must provide, including consultation features, advertising, and administrative tools.
* External Interface Requirements: Describes the system's interactions with external components, including user interfaces and third-party services.
* Nonfunctional Requirements: Specifies the performance, security, and usability criteria the system must meet.
* Other Requirements: Covers any additional requirements, such as legal or regulatory needs, that are not addressed elsewhere in the document.
* This document is intended to be used by all project stakeholders to ensure a shared understanding of the system's objectives and requirements.

# Overall Description

[This section of the SRS describes the general factors that affect the product and its requirements. This section does not state specific requirements. Instead, it provides a background for those requirements, which are defined in detail in Section 3, and makes them easier to understand. Include such items as:

• product perspective

• product functions

• user characteristics

• constraints

• assumptions and dependencies

• requirements subsets]

This section of the SRS provides an overview of the general factors that influence the product and its requirements. It offers background information to help understand the specific requirements detailed in Section 3. It includes the following subsections:

**2.1 Product Perspective**

The Feng Shui Koi Fish Consultation System is designed to be an independent web-based application that provides personalized advice on selecting Koi fish and designing fish ponds according to Feng Shui principles. It interacts with various data sources, such as user profiles and predefined Feng Shui rules, to deliver recommendations based on the user’s personal destiny. This system will serve as a standalone application, accessible to users via a web browser, and does not depend on any external systems.

**2.2 Product Functions**

The primary functions of the system include:

* Consultation Services: Users can receive personalized recommendations for Koi fish and pond characteristics based on their destiny.
* Information Database: The system maintains a database of Koi fish species, pond features, and Feng Shui principles that are used in consultations.
* User Management: Users can create and manage their profiles, including personal information and preferences.
* Advertising Platform: Users can post and manage advertisements related to Koi fish and pond decorations.
* Billing and Payments: The system handles payment processing for consultation services and advertising packages.
* Reporting and Analytics: A dashboard feature provides statistical data on user interactions and system usage.

**2.3 User Characteristics**

The system will cater to two main user groups:

* General Users: Individuals seeking advice on Koi fish and pond setups based on their personal destiny.
* Administrators: Users who manage the system, including content updates, user management, and monitoring of system operations.

**2.4 Constraints**

- The system must be accessible via major web browsers and be compatible with bothdesktop and mobile devices.

- Response times for consultation queries should not exceed 5 seconds under normal operating conditions.

- The system must comply with relevant data protection regulations to ensure the security and privacy of user information.

**2.5 Assumptions and Dependencies**

- The system assumes that users will provide accurate personal information (e.g., birth date, gender) for accurate consultation results.

- The accuracy of the consultation recommendations depends on the correctness of the underlying Feng Shui rules and data provided by the system.

**2.6 Requirements Subsets**

This document addresses all functional and non-functional requirements for the Feng Shui Koi Fish Consultation System, including user interface requirements, system performance criteria, and data management needs.

# FUNCTIONAL Requirements

## <Use Cases Diagram>

[The main Use Case Diagrams of the system]

The main Use Case Diagrams for the system should show interactions between different types of users (e.g., General Users, Managers, Administrators) and the system's features.

## Use Case 1: Create Catalog



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE CASE-n SPECIFICATION** | | | | |
| **Use-case No.** | <UC001> | **Use-case Version** | | <1.0> |
| **Use-case Name** | Create Catalog | | | |
| **Author** | Team | | | |
| **Date** | Dd/mm/yyyy | **Priority** | High | |
| **Actor:**  *Manager*  **Summary:**  *This use case describes the process where a Manager creates a new catalog of Koi fish or pond features.*  **Goal:**  *To provide a structured catalog for users to browse and choose Koi fish and pond features.*  **Triggers**  *<What does lead in using this case?>*  **Preconditions:**  *The Manager must be logged into the system.*  **Post Conditions:**  *<List the required post-conditions for using this case>*  **Main Success Scenario:**   * *The Manager navigates to the "Catalog Management" section.* * *The Manager selects "Create New Catalog" and enters details.* * *The system validates and saves the catalog to the database.* * *A confirmation message is displayed*   **Alternative Scenario:**  *<List other steps for using this case to reach the goal in some alternative conditions >*  **Exceptions:**  *Incomplete data entry prompts an error message.*  **Relationships:**  *<List the relationships that use case relates to>*  **Business Rules:**  *<Any concern about the business>* | | | | |

## < Use Case Name 2>

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE CASE-n SPECIFICATION** | | | | |
| **Use-case No.** | <UC001> | **Use-case Version** | | <1.0> |
| **Use-case Name** | Create Catalog | | | |
| **Author** | Team | | | |
| **Date** | Dd/mm/yyyy | **Priority** | High | |
| **Actor:**  *Manager*  **Summary:**  *This use case describes the process where a Manager creates a new catalog of Koi fish or pond features.*  **Goal:**  *To provide a structured catalog for users to browse and choose Koi fish and pond features.*  **Triggers**  *<What does lead in using this case?>*  **Preconditions:**  *The Manager must be logged into the system.*  **Post Conditions:**  *<List the required post-conditions for using this case>*  **Main Success Scenario:**   * *The Manager navigates to the "Catalog Management" section.* * *The Manager selects "Create New Catalog" and enters details.* * *The system validates and saves the catalog to the database.* * *A confirmation message is displayed*   **Alternative Scenario:**  *<List other steps for using this case to reach the goal in some alternative conditions >*  **Exceptions:**  *Incomplete data entry prompts an error message.*  **Relationships:**  *<List the relationships that use case relates to>*  **Business Rules:**  *<Any concern about the business>* | | | | |

…………………

# NON-FUNCTIONAL Requirements

[This section describes the non-functional requirements of the system. Some examples are listed as below]

## Usability

[This section includes all those requirements that affect usability. For example,

specify the required training time for a normal users and a power user to become productive at particular operations

specify measurable task times for typical tasks or base the new system’s usability requirements on other systems that the users know and like

specify requirement to conform to common usability standards, such as IBM’s CUA standards Microsoft’s GUI standards]The usability of the "Feng Shui Koi Fish Consultation System" is crucial to ensure a smooth and intuitive user experience. The following usability requirements apply:

The usability of the "Feng Shui Koi Fish Consultation System" is crucial to ensure a smooth and intuitive user experience. The following usability requirements apply:

### **Intuitive Interface Design:** The system should have a user-friendly interface that adheres to standard design principles, making it easy for users to navigate and perform tasks without extensive training.

### **Consistency:** All user interface elements should be consistent in terms of style, color, and placement to avoid confusion and improve learnability.

### **Accessibility:** The system should be accessible to users with varying levels of technical expertise. It should support screen readers and keyboard navigation to assist users with disabilities.

### **Error Handling:** Clear and concise error messages should be provided for incorrect user inputs or system errors, guiding the user to correct the issue without frustration.

### **Responsiveness:** The interface should be responsive to different screen sizes, including mobile devices, to ensure that all features are accessible on various platforms.

### **Help and Documentation:** Comprehensive help documentation and tooltips should be provided to guide users through the functionalities of the system.

### <Usability Requirement One>

[The requirement description goes here.]

## Reliability

[Requirements for reliability of the system should be specified here. Some suggestions follow:

Availability—specify the percentage of time available ( xx.xx%), hours of use, maintenance access, degraded mode operations, and so on.

Mean Time Between Failures (MTBF) — this is usually specified in hours, but it could also be specified in terms of days, months or years.

Mean Time To Repair (MTTR)—how long is the system allowed to be out of operation after it has failed?

Accuracy—specifies precision (resolution) and accuracy (by some known standard) that is required in the system’s output.

Maximum Bugs or Defect Rate—usually expressed in terms of bugs per thousand lines of code (bugs/KLOC) or bugs per function-point( bugs/function-point).

Bugs or Defect Rate—categorized in terms of minor, significant, and critical bugs: the requirement(s) must define what is meant by a “critical” bug; for example, complete loss of data or a complete inability to use certain parts of the system’s functionality.]

Reliability is essential to ensure that the system is dependable and functions correctly under different conditions. The following reliability requirements apply:

**Availability:** The system must be available 99.5% of the time during operating hours, excluding scheduled maintenance windows.

**Mean Time Between Failures (MTBF):** The system should have a mean time between failures of at least 1000 hours. This ensures minimal downtime and disruption for users.

**Mean Time to Repair (MTTR):** In the event of a system failure, the mean time to repair should not exceed 2 hours. This includes diagnosing and resolving the issue to restore normal operation.

**Data Integrity:** All user data must be accurately stored and retrieved without corruption. Regular backups should be taken to prevent data loss.

**Accuracy:** The system must provide accurate consultation results based on user inputs, with a precision rate of 95% or higher for compatibility checks between user destiny and Koi fish/pond characteristics.

**Fault Tolerance:** The system should be able to handle minor failures without impacting the overall functionality. For example, if one module fails, other modules should continue to operate independently.

**Error Rate:** The system should maintain an error rate of less than 1% for all processing tasks, including data entry, consultations, and user interactions.

### <Reliability Requirement One>

[The requirement description.]

## Performance

[The system’s performance characteristics are outlined in this section. Include specific response times. Where applicable, reference related Use Cases by name.

Response time for a transaction (average, maximum)

Throughput, for example, transactions per second

Capacity, for example, the number of customers or transactions the system can accommodate

Degradation modes (what is the acceptable mode of operation when the system has been degraded in some manner)

Resource utilization, such as memory, disk, communications, and so forth.

The performance requirements outline the expected behavior of the system in terms of response time, throughput, and resource utilization. These requirements ensure that the system can handle a specified load and maintain acceptable performance levels under different conditions.

**Response Time:** The system should provide consultation results within 3 seconds after a user submits a request under normal operating conditions.

**Throughput**: The system should be able to handle up to 100 concurrent consultation requests without a significant degradation in performance.

**Capacity:** The system must support up to 10,000 active user accounts and at least 1,000 daily consultations.

**Degradation Modes:** In the event of high server load, the system should prioritize basic functionalities (e.g., user login and profile access) over non-essential features (e.g., blog browsing).

**Resource Utilization:** The system should not exceed 70% CPU and 75% memory usage under normal operating conditions, ensuring sufficient resources are available for additional processes.

**Interfaces:** The system should provide an API for external systems to access consultation data, with a response time of less than 2 seconds for API calls

### <Performance Requirement One>

[The requirement description goes here.]

Interfaces

## Supportability

[This section indicates any requirements that will enhance the supportability or maintainability of the system being built, including coding standards, naming conventions, class libraries, maintenance access, and maintenance utilities.]

Supportability requirements focus on the system's maintainability and ease of updates or enhancements. These requirements ensure that the system is built with proper documentation and coding practices, allowing for efficient maintenance and troubleshooting.

**Coding Standards:** All code must adhere to predefined coding standards and naming conventions to ensure consistency and readability.

**Documentation:** The system should include comprehensive technical documentation, covering code structure, API specifications, and database schemas.

**Error Logging and Monitoring:** The system must have error logging mechanisms in place, with real-time monitoring and alert notifications for critical errors.

**Modularity:** The system architecture should be modular, allowing individual components to be updated or replaced without affecting the overall system.

### <Supportability Requirement One>

[The requirement description goes here.]

## Design Constraints

[This section indicates any design constraints on the system being built. Design constraints represent design decisions that have been mandated and must be adhered to. Examples include software languages, software process requirements, prescribed use of developmental tools, architectural and design constraints, purchased components, class libraries, and so on.]

Design constraints specify any limitations imposed on the system design or development process, including technology choices, architectural decisions, and compliance requirements.

**Technology Stack:**

The system must be developed using the following technologies:

Backend: Java Spring Boot

Frontend: React

Database: SQL Server

Cloud Hosting: AWS or Azure

**Security Compliance:** The system must comply with relevant data protection and security standards, such as GDPR and OWASP Top 10.

**Scalability:** The system architecture should be designed to scale horizontally, allowing additional servers to be added as user demand increases.

**Third-Party Integration:** The system must be capable of integrating with third-party payment gateways and external APIs for additional functionalities.

### <Design Constraint One>

[The requirement description goes here.]

## On-line User Documentation and Help System Requirements

[Describes the requirements, if any, for o-line user documentation, help systems, help about notices, and so forth.]

Providing comprehensive user documentation and help resources is essential for ensuring a positive user experience and reducing the need for direct support.

**User Guide:** The system should include a detailed user guide accessible from the main interface, covering all system functionalities and common user tasks.

**Help System:**An integrated help system should provide context-sensitive help topics, offering guidance based on the user’s current activity within the system.

**FAQs and Tutorials:** The system should offer a frequently asked questions (FAQ) section and video tutorials for common user actions, such as requesting consultations and managing advertisements.

**Feedback Mechanism:** A feedback form should be available for users to report issues or suggest improvements, with responses monitored and addressed by the support team.

## Purchased Components

[This section describes any purchased components to be used with the system, any applicable licensing or usage restrictions, and any associated compatibility and interoperability or interface standards.]

This section outlines any external components or software modules that will be purchased and integrated into the system. These components must be compatible with the system’s architecture and meet all licensing and usage requirements.

**Database Management System:** The system will use PostgreSQL as the primary database management system, which must be licensed and supported.

**Payment Gateway:** Integration with a third-party payment gateway, such as Stripe or PayPal, will be required. The usage of these components is subject to their respective licensing agreements.

**User Interface Components:** Pre-built UI components or libraries (e.g., Material-UI for React) may be purchased to streamline the development process.

**Licensing Compliance:** All purchased components must be compliant with the system’s open-source or proprietary licensing requirements and must not conflict with the system's intended distribution model.

## Interfaces

[This section defines the interfaces that must be supported by the application. It should contain adequate specificity, protocols, ports and logical addresses, and the like, so that the software can be developed and verified against the interface requirements.]

The system must support various interfaces to enable interaction between different components, users, and external systems. These interfaces ensure the system can operate cohesively and communicate effectively with other platforms.

***User Interfaces***

**Web Interface:** The primary user interface will be a web-based application, accessible through modern web browsers (Chrome, Firefox, Safari). The design will be responsive, supporting both desktop and mobile devices.

**Dashboard Interface:** Administrators will have access to a dashboard that displays system analytics, user activities, and consultation statistics.

***Hardware Interfaces***

**Server Hardware Requirements:** The system should be hosted on cloud-based infrastructure with a minimum of 8 GB RAM, 4 CPU cores, and 100 GB SSD storage.

**User Devices:** The system should be accessible on devices running at least iOS 12 or Android 8, and desktop devices with Windows 10 or macOS Mojave.

***Software Interfaces***

**API Integration:**The system will provide RESTful APIs for external applications to access consultation data and user information. These APIs will support JSON format and include authentication mechanisms.

**Database Connectivity:** The system will use standard database drivers to connect to the PostgreSQL database. All queries and updates will be handled via ORM (Object-Relational Mapping) to ensure data integrity and security.

Communications Interfaces

**Network Protocols:** The system will use HTTPS for all client-server communications to ensure data is encrypted during transit.

**External System Interfaces:** The system will integrate with third-party services, such as payment gateways and email servers, using their provided APIs and secure communication protocols.

### User Interfaces

[Describe the user interfaces that are to be implemented by the software.]

### Hardware Interfaces

[This section defines any hardware interfaces that are to be supported by the software, including logical structure, physical addresses, expected behavior, and so on.]

### Software Interfaces

[This section describes software interfaces to other components of the software system. These may be purchased components, components reused from another application or components being developed for subsystems outside of the scope of this SRS but with which this software application must interact.]

### Communications Interfaces

[Describe any communications interfaces to other systems or devices such as local area networks, remote serial devices, and so forth.]

## Licensing Requirements

[Defines any licensing enforcement requirements or other usage restriction requirements that are to be exhibited by the software.]

All components of the system, whether custom-developed or purchased, must adhere to their respective licensing terms. The system must not include any component that violates open-source licenses (e.g., GPL, MIT) or commercial licensing agreements.

**Internal Licensing:** Any custom code or internal components developed for the system will be licensed under an appropriate open-source or proprietary license, as determined by the project stakeholders.

**Third-Party Components:** All third-party libraries and frameworks used in the system must be documented, including their licenses and compliance with the project’s overall licensing strategy.

## Legal, Copyright, and Other Notices

[This section describes any necessary legal disclaimers, warranties, copyright notices, patent notices, wordmark, trademark, or logo compliance issues for the software.]

This section covers the legal disclaimers and copyright notices that must be included in the system and its documentation.

**Copyright Notice:** A copyright notice must be included on the main interface and all documentation, specifying the ownership of the software.

**Terms of Service:** Users must agree to the terms of service before using the system. This agreement will outline the rights and responsibilities of both the users and the system provider.

**Privacy Policy:** A privacy policy must be provided, detailing how user data is collected, stored, and used in compliance with regulations such as GDPR or CCPA.

**Trademark and Logo Usage:** Any use of trademarks, logos, or branding in the system must adhere to the respective usage guidelines of the owners.

## Applicable Standards

[This section describes by reference any applicable standard and the specific sections of any such standards which apply to the system being described. For example, this could include legal, quality and regulatory standards, industry standards for usability, interoperability, internationalization, operating system compliance, and so forth.]

This section outlines the standards that the system must comply with, covering legal, quality, regulatory, and industry-specific guidelines. Adherence to these standards ensures that the system meets the required levels of usability, security, interoperability, and data protection.

**Usability Standards:** ISO 9241-11: This standard provides guidance on usability, focusing on the effectiveness, efficiency, and satisfaction of users interacting with the system.

**Security Standards:**

* ISO/IEC 27001: This standard specifies requirements for establishing, implementing, maintaining, and continuously improving an information security management system (ISMS).
* OWASP Top 10: The system must be developed in accordance with the OWASP Top 10 guidelines to protect against common web vulnerabilities such as cross-site scripting (XSS) and SQL injection.

**Accessibility Standards:** WCAG 2.1 (Web Content Accessibility Guidelines): The system’s user interface should adhere to WCAG 2.1 standards to ensure accessibility for users with disabilities.

**Data Protection and Privacy:**

* GDPR (General Data Protection Regulation): For systems operating in or with users from the European Union, compliance with GDPR is mandatory to ensure the protection of personal data.
* CCPA (California Consumer Privacy Act): For systems interacting with users in California, adherence to CCPA requirements is necessary to protect user privacy and data rights.

**Interoperability Standards:** RESTful API Standards: The system's APIs should follow RESTful standards, including the use of standard HTTP methods (GET, POST, PUT, DELETE) and status codes to ensure smooth integration with external systems.

**Quality Standards:** ISO/IEC 25010: This standard provides a framework for evaluating the quality of software systems, including functionality, reliability, usability, and performance.

**Software Development Standards:** IEEE 830-1998: This standard provides recommended practices for creating software requirements specifications, ensuring that all requirements are clearly and consistently documented.

# Supporting Information

[The supporting information makes the SRS easier to use. It includes:

Table of contents

Index

Appendices

These may include use-case storyboards or user-interface prototypes. When appendices are included, the SRS should explicitly state whether or not the appendices are to be considered part of the requirements.]

The supporting information section makes the SRS easier to use and understand by providing additional resources such as appendices, index, and storyboards.

**5.1 Table of Contents**

The table of contents lists all sections and subsections of the SRS, making it easy to navigate through the document.

**5.2 Index**

The index contains keywords and key terms found throughout the document, along with the corresponding page numbers. This allows readers to quickly locate specific information within the SRS.

**5.3 Appendices**

The appendices include supplementary materials that support the main content of the SRS. These may include:

* *Glossary:* Definitions of terms, acronyms, and abbreviations used throughout the document.
* *Use Case Storyboards:* Visual representations of key use cases, illustrating the interactions between users and the system.
* *Data Dictionary:* A detailed description of the data elements used within the system, including attributes, formats, and relationships.
* *User Interface Prototypes:* Mock-ups or screenshots of the system’s user interface, providing a visual guide to the design and layout of key screens.
* *Business Rules:* A comprehensive list of business rules that govern the operation and functionality of the system, ensuring that it aligns with business objectives.

**5.4 Additional References**

Any additional documents or references that are relevant to the SRS should be listed here. This may include related project documents, technical manuals, or industry guidelines.