

**Jewelry Production Order System**

**Software Requirement Specification**

Project Code: JPOS001

Document Code: JPOSSRS01

**FPTU HCM Campus, 08-05-2024**

**Record of change**

\*A - Added M - Modified D - Deleted

| **Effective Date** | **Changed Items** | **A\* M, D** | **Change Description** | **New Version** |
| --- | --- | --- | --- | --- |
| 08/05/2024 | Initial | a | Add project overview |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**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**](#_1fob9te) **Introduction 5**

[1.1](#_3znysh7) Purpose 5

[1.2](#_2et92p0) Scope 5

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

[1.4](#_3dy6vkm) References 5

[1.5](#_1t3h5sf) Overview 6

[**2**](#_4d34og8) **Overall Description 7**

[**3**](#_2s8eyo1) **FUNCTIONAL Requirements 8**

[3.1](#_17dp8vu) <Use Cases Diagram> 8

[3.2](#_3rdcrjn) < Use Case Name 1> 8

[3.3](#_26in1rg) < Use Case Name 2> 9

[**4**](#_lnxbz9) **NON-FUNCTIONAL Requirements 10**

[4.1](#_35nkun2) Usability 10

[4.2](#_1ksv4uv) Reliability 10

[4.3](#_44sinio) Performance 10

[4.4](#_2jxsxqh) Supportability 11

[4.5](#_z337ya) Design Constraints 11

[4.6](#_3j2qqm3) On-line User Documentation and Help System Requirements 11

[4.7](#_1y810tw) Purchased Components 11

[4.8](#_4i7ojhp) Interfaces 12

[4.9](#_2xcytpi) Licensing Requirements 12

[4.10](#_1ci93xb) Legal, Copyright, and Other Notices 12

[4.11](#_3whwml4) Applicable Standards 12

[**5**](#_2bn6wsx) **Supporting Information 13**

# 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 Requirement Specification (SRS) document is to present a detailed description of the Jewelry Production Order System. It will describe the external behavior and non-functional requirements, identify potential design constraints, and incorporate other necessary factors vital for successful software development and implementation. This document is intended for both the stakeholders and the developers of the system and will be proposed for its approval.

The SRS also serves as a foundational document for subsequent project planning, design, coding, system testing, and user documentation phases, thoroughly aligning all stakeholders including developers, managers, sales staff, design team, manufacturing staff, and customers on what to expect from the final product.

## Scope

This software system will be an Online Jewelry Production Order Management System for a local jewelry designer who collaborates with different artisans and craftsmen across the area. This system is engineered to elevate the designer's productivity by providing advanced tools aimed to automate the order management and production process, tasks which would otherwise require strenuous manual effort. By notably enhancing the designer's efficiency and production capacity, the system meets the designer's needs, while keeping its design intuitively simple and easy to use.

This system is specifically architected to enable a designer to manage and coordinate with a diverse group of artisans, suppliers, and customers to process orders with ease. The software will facilitate communication between designers, artisans, and customers through integrated messaging channels. Preformatted templates are present at each stage in the order's lifecycle to foster a standardized production process; the mapping of these templates is adaptable via the application's settings. The system houses a robust relational database containing information on Designers, Artisans, Suppliers, Customers, and Orders.

## 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.]*

| ***Term*** | ***Definition*** |
| --- | --- |
| *Software Requirements Specification* | *A document that completely describes all of the functions of a proposed system and the constraints under which it must operate. For example, this document.* |
| *Originator* | *Person submitting an article to be reviewed. In case of multiple originators, this term refers to the principal author, with whom all communication is made.* |
| *Reviewer* | *A person that examines an article and has the ability to recommend approval of the article for publication or to request that changes be made in the article.* |
| *Approval* | *A person responsible for providing final approval on the document or project phase.* |
| *Stakeholders* | *People or organizations who are actively involved in the project, or whose interests may be positively or negatively affected by the execution of the project.* |
| *Database* | *Collection of all the information monitored by this system.* |
| *Mapping* | *The process of linking two things together. In this context, it refers to establishing connections between the aspects of the production process and the corresponding templates.* |
| *Customer* | *The person who uses this web application to request for jewelry production.* |
| *Sale Staff* | *The staff who are in charge of discussing with customers, managing users’ requests, users’ payment and sending quotes to customers.* |
| *Design Staff* | *The staff who are in charge of designing 3D models for customers for the purpose of previews, as the user requests.* |
| *Production Staff* | *The staff who are in charge of recording the production process and manage production order when the product is completed.* |
| *Manager* | *The person who is in charge of staffs’ assignment, approves quotes sent by sale staff and manages production orders.* |
| *Administrator* | *The person who is in charge of user account management and system management.* |

## References

*IEEE. IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications. IEEE Computer Society, 1998.*

## Overview

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

*The next chapter, the Overall Description section, of this document gives an overview of the functionality of the product. It describes the informal requirements and is used to establish a context for the technical requirements specification in the next chapter.*

*The third and fourth chapter, Requirements Specification section, of this document is written primarily for the developers and describes in technical terms the details of the functionality of the product.*

*All three sections of the document describe the same software product in its entirety, but are intended for different audiences and thus use different language.*

# 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]*

**Phần mềm quản lý việc đặt gia công trang sức của khách hàng cho công ty gia công trang sức** - Trang chủ giới thiệu công ty, bộ sưu tập trang sức, mẫu thiết kế, blog chia sẻ, … - Khách hàng có thể đặt yêu cầu gia công đồ trang sức theo mẫu thiết kế công ty hoặc theo yêu cầu thiết kế của khách. - Quản lý quá trình đặt gia công trang sức của khách hàng.

*<< Khách hàng gửi yêu cầu --> NV kinh doanh tiếp nhận và trao đổi với khách --> NV kinh doanh gửi yêu cầu phê duyệt báo giá --> Người quản lý phê duyệt --> NV kinh doanh gửi báo giá đã phê duyệt--> khách hàng chấp nhận báo giá và tạo đơn hàng gia công --> NV thiết kế gửi bản thiết kế 3D cho khách --> khách hàng phê duyệt bản thiết kế 3D --> NV gia công thực hiện gia công trang sức --> NV kinh doanh bàn giao trang sức đã gia công và giấy bảo hành cho khách hàng >>*

- Cập nhật chi phí của đơn hàng gia công theo định mức để báo giá và thực tế gia công để tính giá vốn báo giá và thực hiện

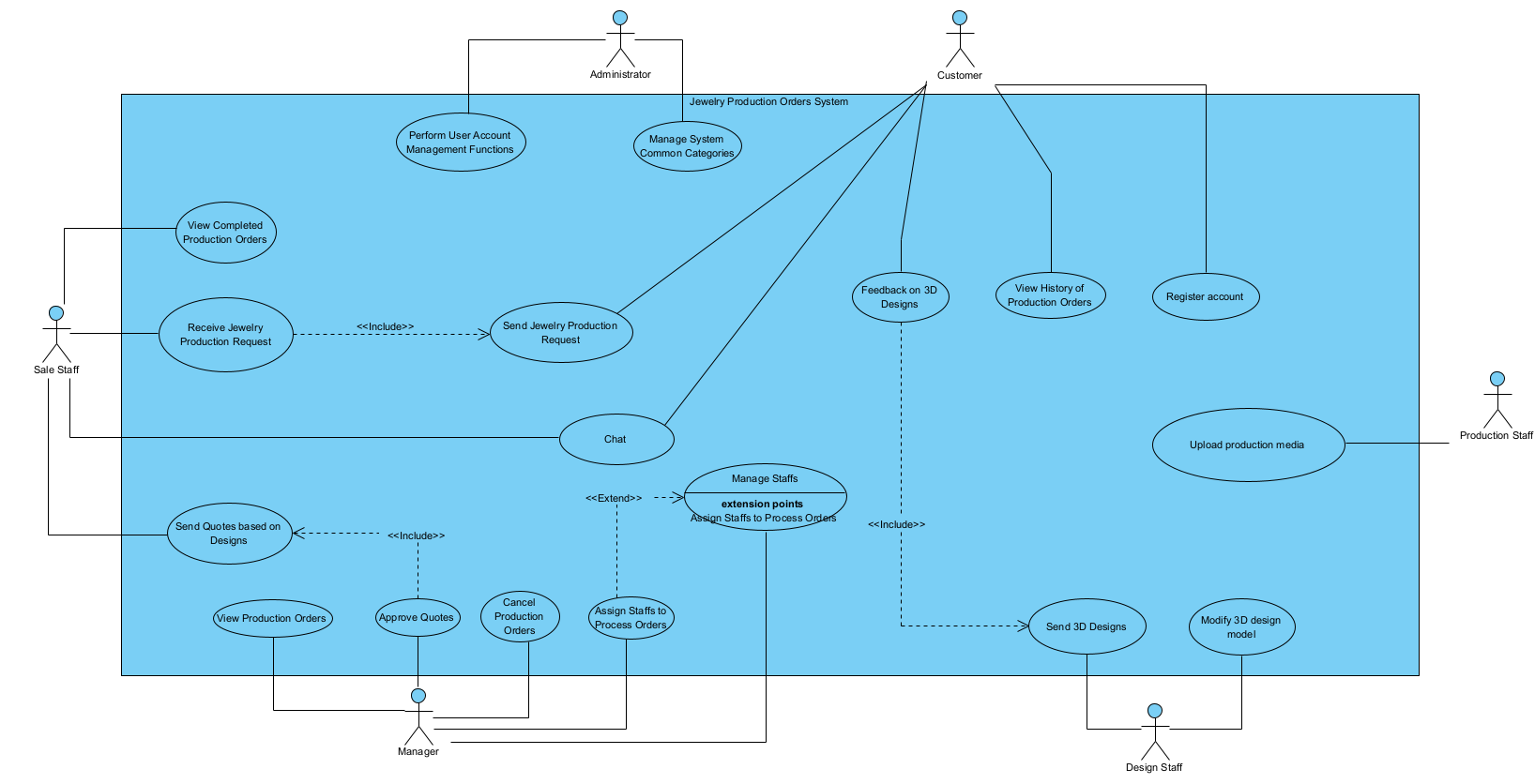
*<< Giá vốn sản phẩm = [giá vàng thời điểm \* trọng lượng sản phẩm] + tiền công + tiền đá >>*

- Khai báo chính sách thanh toán, hủy đơn hàng gia công trang sức. - khai báo giá vàng và giá đá áp dụng của công ty. - Khai báo mẫu thiết kế của công ty và định mức chi phí thực hiện kèm theo. - Dashboard thống kê.

# FUNCTIONAL Requirements

## <Use Cases Diagram>

*[The main Use Case Diagrams of the system]*



## < Use Case Name 1>

| **#** | **Use Case** | **Actors** | **Description** |
| --- | --- | --- | --- |
| UC-01 | Send Jewelry Production Request | Customer | Customers are allowed to send request to produce jewelry for them through the website. |
| UC-02 | Make Payment | Customer | Customers initiate this use case by sending payment via an online payment gateway, which the system will then automatically accept the order. |
| UC-03 | View History of Production Orders | Customer | Customers are allowed to view previous production orders made by them on the web application. |
| UC-04 | Feedback on 3D Designs | Customer | During the design process, customers are allowed to feedback on 3D designs sent by Design Staff in order for them to refine and improve. |
| UC-05 | Receive Jewelry Production Request | Sale Staff | Sales Staff are able to receive requests sent by customers and view their details. |
| UC-06 | Send Quotes based on Designs | Sale Staff | Sales Staff can send quotes based on designs finalized with the customer. |
| UC-07 | Confirm Customer Payment | Sale Staff | In case the customer uses cash payment or bank transfer, Sales Staff have to confirm it manually. |
| UC-08 | Discuss with Customer | Sale Staff | After receiving customers’ requests, Sales Staff can discuss with customers using a chat function in the system. |
| UC-09 | Send 3D Designs | Design Staff | During the design process, Design Staffs are allowed to send 3D designs of the jewelry to the customer. |
| UC-10 | Modify 3D Designs | Design Staff | After customers feedback on the 3D Designs, Design Staffs can edit and adjust the designs according to the customers’ desire. |
| UC-11 | Update Images or Videos of Production Process | Production Staff | During the production process, Production Staff record and update images and videos of the production process on the web app for the customer to view. |
| UC-12 | Update Status of Completed Production Order | Production Staff | After the jewelry is finished, Production Staff can update the production order. |
| UC-13 | Approve Quotes | Manager | Manager can approve quotes sent by the sales staff to ensure correctness. |
| UC-14 | Assign Staffs to Process Orders | Manager | Manager is able to assign staff to process orders made by the customer. |
| UC-15 | Cancel Production Orders | Manager | Manager is able to cancel production orders made by the customer. |
| UC-16 | Perform User Account Management Functions | Administrator | Administrators can perform various account management functions on the web app. |
| UC-17 | Approve Customer Account Registration | Administrator | When the customer registers an account, the admin is able to approve those registration requests to ensure security. |
| UC-18 | Manage System Common Categories | Administrator | Administrator is able to manage system common categories |
| UC-19 | Register Account | Customer | Customers are allowed to register an account on the platform. |
| UC-20 | Chat with Staff | Customer | Customers are allowed to communicate with staff through a built-in chat function |
| UC-21 | Manage Staff | Manager |  |

| **USE CASE-n SPECIFICATION** | | | | |
| --- | --- | --- | --- | --- |
| **Use-case No.** | <UC001> | **Use-case Version** | | <1.0> |
| **Use-case Name** | <Name> | | | |
| **Author** | <Members> | | | |
| **Date** | Dd/mm/yyyy | **Priority** | <High/Normal/Low> | |
| **Actor:**  *<List all actors>*  **Summary:**  *<Briefly describe the used case >*  **Goal:**  *<Briefly describe the goal of used case >*  **Triggers**  *<What does lead in using this case?>*  **Preconditions:**  *<List the required pre-conditions for using this case>*  **Post Conditions:**  *<List the required post-conditions for using this case>*  **Main Success Scenario:**  *<List the main steps for using this case to reach the goal successfully >*  **Alternative Scenario:**  *<List other steps for using this case to reach the goal in some alternative conditions >*  **Exceptions:**  *<List exceptions of this use case >*  **Relationships:**  *<List the relationships that use case relates to>*  **Business Rules:**  *<Any concern about the business>* | | | | |

## < Use Case Name 2>

…………………

# 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]*

### <Usability Requirement One>

*[The requirement description goes here.]*

* Training Time:
* Normal Users: The system should be intuitive enough for a normal user to become productive within 1 hour of training.
* Power Users: Power users should be able to efficiently navigate and utilize advanced features within 2 hours of training.
* Accessibility:
* The system should be accessible to users with disabilities, following WCAG (Web Content Accessibility Guidelines) standards.
* Text should have sufficient color contrast for readability, and interactive elements should be easily distinguishable.
* Task Efficiency:
* The system should allow users to complete common tasks, such as placing an order or approving a design, within 3 minutes on average.
* Complex tasks, like managing production orders or generating reports, should not take longer than 10 minutes for experienced users.
* User Interface Design Standards:
* The system's user interface should adhere to common usability standards, such as Microsoft's GUI standards or web accessibility guidelines like WCAG (Web Content Accessibility Guidelines).
* Navigation should be intuitive, with clear labeling and consistent layout across all pages.
* Feedback and Error Handling:
* The system should provide clear feedback to users for successful actions, errors, and system responses.
* Error messages should be informative and suggest corrective actions to users.
* Customization and Personalization:
* Users should be able to customize their dashboard layout and preferences to suit their individual workflow.
* Personalized recommendations or shortcuts based on user behavior should enhance user experience.
* Mobile Responsiveness:
* The system's interface should be responsive and optimized for mobile devices, ensuring usability across various screen sizes and resolutions.
* Mobile-specific features, such as touch gestures, should be utilized to enhance usability on smartphones and tablets.
* Performance:
* Web Pages should fully download in an average of 5 seconds or less with a 35 megabits/second internet connection to ensure swift access and responsiveness.
* Response time for common actions, such as loading a page or submitting a form, should be under 2 seconds to maintain user engagement.

## 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 Requirement One>

* Availability:
* The system should be available for use 99.9% of the time, excluding scheduled maintenance windows.
* Maintenance windows should be scheduled during off-peak hours, preferably during weekends or non-business hours, with advanced notice provided to users.
* In the event of system maintenance or unexpected downtime, a degraded mode of operation should be available, allowing users to access essential functions such as viewing orders and communicating with customers.
* Mean Time Between Failures (MTBF):
* The system should have a Mean Time Between Failures (MTBF) of at least 1000 hours of continuous operation.
* Mean Time To Repair (MTTR):
* The Mean Time To Repair (MTTR) for any system failure should not exceed 2 hours during business hours and 4 hours during non-business hours.
* Accuracy:
* The system output should have a precision (resolution) of at least 0.01 units for all numerical values.
* Accuracy should be maintained within 95% confidence intervals of known standards for all calculations involving pricing, material quantities, and production costs.
* Maximum Bugs or Defect Rate:
* The maximum acceptable bugs or defects rate should not exceed 0.1 critical bugs per thousand lines of code (bugs/KLOC).
* Bugs or defects are categorized as minor if they do not significantly impact system functionality, significant if they affect usability or performance, and critical if they result in complete loss of data or inability to use essential system functions.

## 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.*

### <Performance Requirement One>

* Response Time:
* The average response time for processing a transaction, such as creating a quote or updating a production status, should not exceed 2 seconds.
* The maximum response time for any transaction should not exceed 5 seconds to ensure timely user interaction and feedback.
* Throughput:
* The system should support a minimum throughput of 10 transactions per minute during peak usage hours.
* During off-peak hours, the system should be capable of handling up to 20 transactions per minute to accommodate increased user activity.
* Capacity:
* The system should be able to accommodate at least 100 concurrent users without experiencing performance degradation.
* It should have the capacity to manage a minimum of 500 production orders simultaneously without affecting system responsiveness.
* Degradation Modes:
* In the event of high traffic or system overload, the system should prioritize essential functions such as order processing and communication with customers.
* Non-essential features, such as generating reports or accessing historical data, may experience slower response times or temporary unavailability during peak usage periods.
* Resource Utilization:
* Memory utilization should not exceed 70% of available system memory under normal operating conditions.
* Disk space usage should be monitored regularly, and alerts should be generated when disk space reaches 80% capacity to prevent performance degradation.
* Network communications should be optimized to minimize latency and ensure smooth data transfer between system components.
* Interfaces:
* The system should provide intuitive user interfaces for both customers and internal staff, supporting multiple devices such as desktops, tablets, and smartphones.
* User interfaces should be responsive, with fast loading times and efficient navigation to enhance user experience.
* Integration with third-party payment gateways and communication channels (e.g., email, chat) should be seamless and reliable to facilitate customer interactions and transactions.

## 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 Requirement One>

* All code written for the system shall adhere to a standardized set of coding standards and naming conventions.
* Coding standards should be based on industry best practices and established guidelines for the selected programming languages and frameworks (e.g., Python, Django, React.js).
* Naming conventions for variables, functions, classes, and other code entities should be descriptive, consistent, and follow a logical naming scheme to enhance readability and maintainability.
* The use of meaningful comments and documentation within the codebase should be encouraged to provide clarity and context for future developers and maintainers.
* Code reviews shall be conducted regularly to ensure compliance with coding standards and to identify opportunities for improvement in code quality and maintainability.
* The development team shall maintain a documented coding style guide that outlines the specific coding standards and naming conventions to be followed throughout the project's lifecycle.

## 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 Constraint One>

* Integration with Existing Systems:
* The system must seamlessly integrate with the company's existing Customer Relationship Management (CRM) system, Salesforce, to ensure real-time data exchange and consistency in customer information.
* Technology Stack:
* The system must be developed using specific technologies: Python for programming language, Django for web framework, React.js for frontend, PostgreSQL for database management, Git for version control, and Docker containers managed with Kubernetes for deployment.
* Coding Standards and Naming Conventions:
* All code written for the system must adhere to industry-standard coding practices, follow a consistent coding style guide (such as PEP 8 for Python code), and use meaningful and descriptive naming conventions to enhance readability and maintainability.

*[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.]*

* User Manual:
* A comprehensive user manual shall be provided, covering all aspects of system functionality, including but not limited to:
* User registration and account management.
* Creating and managing production orders.
* Communicating with customers.
* Approving quotes and designs.
* Updating production statuses.
* Accessing historical data and reports.
* Context-Sensitive Help:
* Context-sensitive help shall be available throughout the system, providing relevant guidance and instructions based on the user's current location or task.
* Help icons or buttons shall be strategically placed within the user interface to allow users to access relevant help content easily.
* Searchable Knowledge Base:
* A searchable knowledge base shall be implemented, containing articles, FAQs, troubleshooting guides, and best practices related to system usage and maintenance.
* Users shall be able to search for specific topics or keywords to quickly locate relevant help articles.
* Interactive Tutorials:
* Interactive tutorials or walkthroughs shall be provided for key system functionalities, guiding users through common tasks and workflows step by step.
* Tutorials shall include interactive elements such as tooltips, annotations, and simulated interactions to enhance user engagement and learning.
* Responsive Design:
* The online user documentation and help system shall feature a responsive design, ensuring optimal viewing and usability across different devices and screen sizes.
* Users shall be able to access help content seamlessly from desktop computers, tablets, and smartphones.
* Versioning and Updates:
* The documentation shall be versioned to align with system releases and updates, ensuring that help content remains accurate and up to date with the latest system features and changes.
* Users shall be notified of documentation updates and changes, with release notes highlighting new additions or revisions.
* Accessibility:
* The online user documentation and help system shall adhere to accessibility standards (e.g., WCAG) to ensure accessibility for users with disabilities.
* Text alternatives for images, keyboard navigation support, and other accessibility features shall be implemented to enhance usability for all users.

## 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.]*

* Salesforce CRM Integration:
* The system will integrate with Salesforce CRM as a purchased component to manage customer relationship data.
* Licensing and usage restrictions: The company holds a valid license for Salesforce CRM usage.
* Compatibility and interoperability: The integration will adhere to Salesforce API standards to ensure seamless data exchange between systems.
* Payment Gateway:
* A third-party payment gateway will be utilized to facilitate online payments for production orders.
* Licensing and usage restrictions: The company has entered into a service agreement with the payment gateway provider.
* Compatibility and interoperability: The system will integrate with the payment gateway's API to securely process online transactions and handle payment information.
* Cloud Hosting Services:
* The system will be hosted on a cloud platform, such as Amazon Web Services (AWS) or Microsoft Azure.
* Licensing and usage restrictions: The company holds subscriptions or usage agreements with the cloud hosting provider.
* Compatibility and interoperability: The system architecture will be designed to leverage cloud services effectively, ensuring scalability, reliability, and security.
* Development Tools and Libraries:
* Various development tools and libraries may be purchased or licensed for use during system development, such as IDEs (Integrated Development Environments), code libraries, and testing frameworks.
* Licensing and usage restrictions: Usage rights and licensing agreements for these tools and libraries will be ensured in compliance with vendor terms.
* Compatibility and interoperability: Development tools and libraries will be selected based on compatibility with the chosen technology stack and adherence to industry standards.

## 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.]*

### 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.]

* User Interfaces:

1. Customer Interface:

* Description: Web-based interface for customers to interact with the system.
* Protocols: HTTP/HTTPS.
* Port: Standard web ports 80 (HTTP) and 443 (HTTPS).
* Logical Address: URL of the web application (e.g., https://jewelrymanagement.com).
* Expected Behavior: Customers can access the system via a web browser, create accounts, place orders, view order history, and provide feedback.

1. Staff Interface:

* Description: Web-based interface for sales, design, and production staff.
* Protocols: HTTP/HTTPS.
* Port: Standard web ports 80 (HTTP) and 443 (HTTPS).
* Logical Address: URL of the web application (e.g., https://jewelrymanagement.com).
* Expected Behavior: Staff members can access the system via a web browser, log in with their credentials, manage customer requests, create quotes, update production statuses, and communicate with customers.
* Hardware Interfaces:
* Server Hardware:
* Description: Hardware components hosting the software application.
* Logical Structure: Servers hosting the application code, databases, and other necessary components.
* Physical Addresses: IP addresses or domain names of the servers.
* Expected Behavior: The servers run the application, handle user requests, and store data.
* Software Interfaces:

1. Salesforce CRM Integration:

* Description: Interface for integrating with Salesforce CRM.
* Protocols: RESTful API.
* Logical Address: API endpoint provided by Salesforce (e.g., https://api.salesforce.com).
* Expected Behavior: The system communicates with Salesforce CRM to synchronize customer data, orders, and communications.

1. Payment Gateway Integration:

* Description: Interface for integrating with a third-party payment gateway.
* Protocols: HTTPS.
* Logical Address: API endpoint provided by the payment gateway provider.
* Expected Behavior: The system communicates securely with the payment gateway to process online payments for production orders.
* Communications Interfaces:

1. Internal Communication:

* Description: Communication between different modules and components of the software system.
* Protocols: HTTP/HTTPS.
* Logical Address: Internal API endpoints.
* Expected Behavior: Modules communicate via RESTful APIs to exchange data and trigger actions within the system.

1. External Communication:

* Description: Communication with external systems such as email servers or SMS gateways.
* Protocols: SMTP, SMS API.
* Logical Address: SMTP server address for email communication, SMS gateway API endpoint for SMS communication.
* Expected Behavior: The system sends notifications and alerts via email and SMS using the respective protocols and interfaces.

## Licensing Requirements

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

* Third-Party Components:
* Any third-party components or libraries used in the software must be compliant with their respective licenses.
* Developers must ensure that the usage of third-party components adheres to the terms and conditions specified in their licenses.
* Proprietary Code:
* Any proprietary code developed specifically for the Jewelry Production Order Management System remains the property of the company.
* Usage of proprietary code is restricted to authorized personnel for system development, deployment, and maintenance purposes.
* Software Distribution:
* The software may not be distributed or sublicensed without explicit permission from the company.
* Users may only access the software through authorized channels provided by the company.
* Usage Restrictions:
* The software is intended for internal use by the company and its authorized users only.
* Unauthorized reproduction, modification, or reverse engineering of the software is strictly prohibited.
* Compliance with Regulations:
* The software must comply with all applicable laws, regulations, and industry standards governing software licensing and usage.
* License Agreement:
* Users must agree to the terms and conditions outlined in the software's End-User License Agreement (EULA) before accessing or using the system.
* The EULA shall outline the rights and responsibilities of users regarding software usage, intellectual property rights, and liability.

## 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.]*

* Copyright Notice:
* The software is protected by copyright laws, and all rights are reserved by the company.
* © [Year of Copyright] [Company Name]. All rights reserved.
* Trademark Notice:
* Any trademarks, service marks, logos, or trade names used in the software are the property of the company.
* All trademarks, service marks, logos, and trade names are registered or unregistered trademarks of [Company Name].
* End-User License Agreement (EULA):
* Users must agree to the terms and conditions outlined in the EULA before using the software.
* The EULA shall govern the rights and responsibilities of users regarding software usage, intellectual property rights, and liability.
* Legal Disclaimers:
* The software is provided "as is" without any warranties, express or implied.
* The company shall not be liable for any damages arising from the use or inability to use the software, including but not limited to indirect, incidental, or consequential damages.
* Compliance Notices:
* Users must comply with all applicable laws, regulations, and industry standards when using the software.
* The company reserves the right to terminate or suspend access to the software for users who violate the terms of service or engage in illegal activities.
* Patent Notices:
* Any patented features or technologies incorporated into the software shall be identified with appropriate patent notices.
* Wordmark Compliance:
* The company's wordmark or logo shall be displayed in accordance with branding guidelines and compliance standards.

## 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.]*

* ISO 9001:2015 (Quality Management Systems):
* Section 7: Support - This section covers requirements related to documentation, resources, competence, awareness, and communication, which are relevant for ensuring the quality of the software development process.
* ISO/IEC 27001:2013 (Information Security Management Systems):
* Section 6: Planning - This section outlines requirements for risk assessment, treatment, and control, which are crucial for ensuring the security of customer data and sensitive information within the system.
* GDPR (General Data Protection Regulation):
* Articles 5, 6, 7, 9, 17, 20 - These articles specify requirements related to data protection, lawful processing, consent, data subject rights, data minimization, and accountability, which must be adhered to for handling customer data in compliance with EU data protection regulations.
* PCI DSS (Payment Card Industry Data Security Standard):
* Requirements 3, 6, 7, 9, 10 - These requirements address security controls for protecting payment card data, including network security, access control, encryption, vulnerability management, and monitoring, which are relevant for ensuring the security of online payment transactions.
* WCAG (Web Content Accessibility Guidelines):
* Level AA - These guidelines provide recommendations for making web content more accessible to people with disabilities, including visual, auditory, physical, speech, cognitive, language, learning, and neurological disabilities, which are important for ensuring accessibility compliance of the user interfaces.
* HTTP/HTTPS Protocol Standards:
* RFC 7230, RFC 7231, RFC 7232, RFC 7233 - These RFCs define the HTTP/1.1 protocol standards for message syntax and routing, request and response semantics, conditional requests, and range requests, which are relevant for implementing secure and efficient communication between clients and servers.

# 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.]*