Requirement Analysis for Jewel Management



This document outlines the requirement analysis for implementing a Jewel Management system within Salesforce. It encompasses a detailed exploration of the customer journey, a comprehensive data flow diagram illustrating the system's architecture, a thorough breakdown of solution requirements, and an overview of the proposed technology stack. This analysis serves as a blueprint for the development and deployment of a robust and efficient Jewel Management solution tailored to the specific needs of the business.

### Implementing Jewel Management System

##### 5 Robust Solution



Achieving a fully functional and

efficient system.

##### 4 Technology Stack

Selecting the appropriate technologies for implementation.

##### 3 Solution Requirements

Defining the necessary features

and functionalities.

##### 2 Data Flow Diagram

Mapping the system's

architecture and data flow.

##### 1 Customer Journey

Understanding the customer's

experience with the system.

# **Customer Journey Map:**

The customer journey map visualizes the end-to-end experience of a customer interacting with the Jewel Management system, from initial awareness to post-purchase engagement. It identifies key touchpoints, customer actions, motivations, pain points, and opportunities for improvements.

### Customer Journey Map Development



##### Seize Opportunities 5

Find ways to enhance

the customer experience.

##### Address Pain

##### Points 4

Identify and resolve customer frustrations.

##### Understand

Motivations 3

Explore why customers behave as they do.

##### Analyze Customer Actions

Understand what 2

customers do at each

touchpoint.

##### Identify Touchpoints

1

Determine key interaction points in the customer

journey.

**\*\*Awareness & Discovery\*\*:**

* + - Touchpoints: Online advertisements (Google Ads, social media), search engine results, website content (blog posts, articles), social media posts, referrals from existing customers.
    - Customer Actions: Searches for specific jewelry types, browses online catalogs, reads

reviews, compares prices, visits the company website.

* + - Motivations: Desire to purchase jewelry for personal use or as a gift, seeking specific styles or materials, looking for competitive pricing.
    - Pain Points: Difficulty finding specific items, lack of detailed product information,

concerns about authenticity and quality, confusing website navigation.

* + - Opportunities: Targeted advertising campaigns, improved search engine optimization (SEO), high-quality product photography and descriptions, user-friendly website design, customer testimonials.

**\*\*Consideration & Evaluation\*\*:**

* + - Touchpoints: Product pages on the website, online chat support, email inquiries, social media interactions, in-store visits (if applicable).
    - Customer Actions: Views product details, compares similar items, reads customer reviews, asks questions about products, requests quotes.
    - Motivations: Seeking detailed information about products, comparing prices and

features, assessing the company's reputation, seeking personalized recommendations.

* + - Pain Points: Insufficient product information, slow response times from customer support, lack of personalized recommendations, concerns about payment security.
    - Opportunities: Detailed product specifications (carat, cut, clarity, color), high-quality

images and videos, proactive customer support, personalized product recommendations based on browsing history, secure payment gateway integration.

**\*\*Purchase & Order Fulfillment\*\*:**

* + - Touchpoints: Online shopping cart, checkout process, order confirmation email, shipping updates, delivery notification.
    - Customer Actions: Adds items to cart, enters shipping and billing information, selects

payment method, confirms order, tracks shipment.

* + - Motivations: Completing the purchase process smoothly, receiving timely updates on order status, ensuring secure payment processing.
    - Pain Points: Complicated checkout process, lack of payment options, high shipping

costs, delayed delivery, damaged or incorrect items.

* + - Opportunities: Streamlined checkout process, multiple payment options, transparent shipping costs, real-time order tracking, proactive communication about potential delays.

**\*\*Post-Purchase & Engagement\*\*:**

* + - Touchpoints: Order confirmation email, thank you notes, customer satisfaction surveys, email newsletters, loyalty programs, social media engagement, warranty information.
    - Customer Actions: Receives order confirmation, provides feedback on purchase experience, subscribes to newsletters, participates in loyalty programs, seeks warranty support.
    - Motivations: Receiving confirmation of purchase, providing feedback to improve the

customer experience, staying informed about new products and promotions, earning rewards for loyalty.

* + - Pain Points: Lack of post-purchase support, difficulty accessing warranty information,

irrelevant marketing emails, lack of personalized engagement.

* + - Opportunities: Personalized thank you notes, proactive customer support, easy access to warranty information, targeted email marketing campaigns, loyalty programs with exclusive benefits.

# **2.Data Flow Diagram:**

The Data Flow Diagram (DFD) illustrates the movement of data within the Jewel Management system. It provides a visual representation of the processes, data stores, external entities, and data flows involved in managing jewel-related information.

*Entities:*

* + - Customer: Interacts with the system to browse, purchase, and manage their jewel-related information.
    - Sales Representative: Uses the system to manage customer interactions, process

orders, and track sales performance.

* + - Inventory Manager: Responsible for managing jewel inventory, tracking stock levels, and updating product information.
    - Supplier: Provides jewel inventory to the system.
    - Payment Gateway: Processes online payments for jewel purchases.

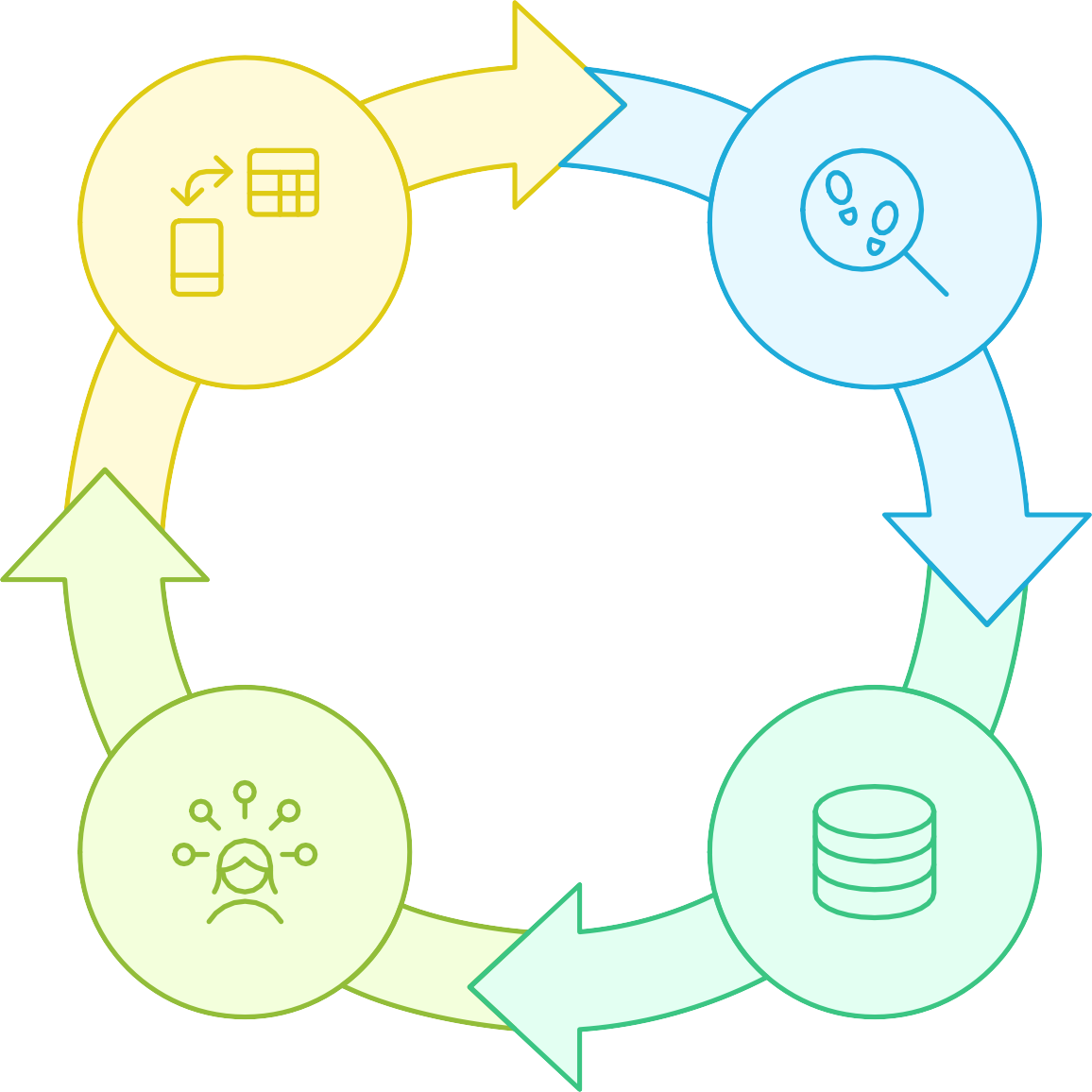
*Processes:*

* + - Browse Jewels: Allows customers to browse available jewels based on various criteria (e.g., type, carat, price).
    - Place Order: Enables customers to place orders for selected jewels.
    - Process Payment: Handles online payment processing through the payment gateway.
    - Manage Inventory: Allows inventory managers to track stock levels, update product information, and receive alerts for low stock.
    - Generate Reports: Creates reports on sales performance, inventory levels, and

customer behavior.

* + - Manage Customer Data: Stores and manages customer information, including contact details, purchase history, and preferences.

**Data Flow Cycle in Jewel Management**



#### Map Data Flows

Illustrate how data moves

#### Identify Processes

Recognize key processes in the system

#### Recognize External

**Entities**

Identify entities outside

the system

#### Define Data Stores

Determine where data is stored

*Data Stores:*

* + - Jewel Inventory: Stores information about available jewels, including type, carat, price, and stock levels.
    - Customer Data: Stores customer information, including contact details, purchase

history, and preferences.

* + - Order Data: Stores information about customer orders, including order date, items ordered, and shipping address.
    - Payment Data: Stores payment information for completed orders.

*Data Flows:*

* + - Customer requests to browse jewels.
    - System retrieves jewel information from the Jewel Inventory data store.
    - Customer selects jewels and places an order.
    - Order information is stored in the Order Data data store.
    - System processes payment through the Payment Gateway.
    - Payment information is stored in the Payment Data data store.
    - Inventory Manager updates jewel information in the Jewel Inventory data store.
    - Sales Representative accesses customer data from the Customer Data data store.
    - System generates reports based on data from various data stores.

## Solution Requirements:

## Functional Requirements Overview



User Needs



Development Process

Stakeholder Understanding



* + - ***Jewel Catalog Management:***
      * Ability to add, edit, and delete jewel information (e.g., type, carat, price, images).
      * Support for different jewel types (e.g., diamonds, gemstones, pearls).
      * Ability to categorize jewels based on various criteria (e.g., style, occasion).
      * Advanced search functionality to allow customers to easily find specific jewels.
    - ***Order Management:***
      * Ability to place orders for selected jewels.
      * Support for different payment methods (e.g., credit card, PayPal).
      * Automated order confirmation and shipping updates.
      * Order tracking functionality for customers.
      * Ability to manage order cancellations and returns.
    - ***Inventory Management:***
      * Real-time tracking of jewel inventory levels.
      * Automated alerts for low stock levels.
      * Ability to manage jewel suppliers and purchase orders.
      * Support for different inventory locations.
    - ***Customer Management:***
      * Ability to store and manage customer information (e.g., contact details, purchase history, preferences).
      * Customer segmentation based on various criteria (e.g., purchase history,

demographics).

* + - * Personalized product recommendations based on customer preferences.
    - **Reporting and Analytics:**
      * Reports on sales performance, inventory levels, and customer behavior.
      * Customizable dashboards to visualize key metrics.
      * Ability to export data in various formats (e.g., CSV, Excel).
    - ***User Management:***
      * Role-based access control to restrict access to sensitive data.
      * Ability to manage user accounts and permissions.
      * Audit logging to track user activity.

## Feature Comparison for Different Management Systems



**C**



|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **haracteristic** | |  | **Jewel Catalog** |  | **Order** |  | **Inventory** |  | **Customer** |  | **Reporting & Analytics** |  | **User** |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | **Add/Edit/Delete** |  | Yes |  | Yes |  | Yes |  | Yes |  | Yes |  | Yes |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | **Support** |  | Yes |  | Yes |  | Yes |  | Yes |  | No |  | No |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | **Categorization** |  | Yes |  | No |  | No |  | No |  | No |  | No |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | **Search** |  | Yes |  | Yes |  | No |  | No |  | No |  | Yes |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | **Automation** |  | No |  | Yes |  | Yes |  | Yes |  | No |  | No |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | **Real-time Tracking** |  | No |  | No |  | Yes |  | No |  | No |  | No |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | **Visualization** |  | No |  | No |  | No |  | No |  | Yes |  | No |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | **Access Control** |  | No |  | No |  | No |  | No |  | No |  | Yes |

**Non-Functional Requirements:**

* + - Performance: The system should be responsive and efficient, with minimal loading times.
    - Scalability: The system should be able to handle a growing number of users and data

without performance degradation.

* + - Security: The system should be secure and protect sensitive data from unauthorized access.
    - Usability: The system should be user-friendly and easy to navigate.

**Usability**

Makes the system user-friendly and easy to navigate



**Security**

Protects sensitive data from unauthorized access

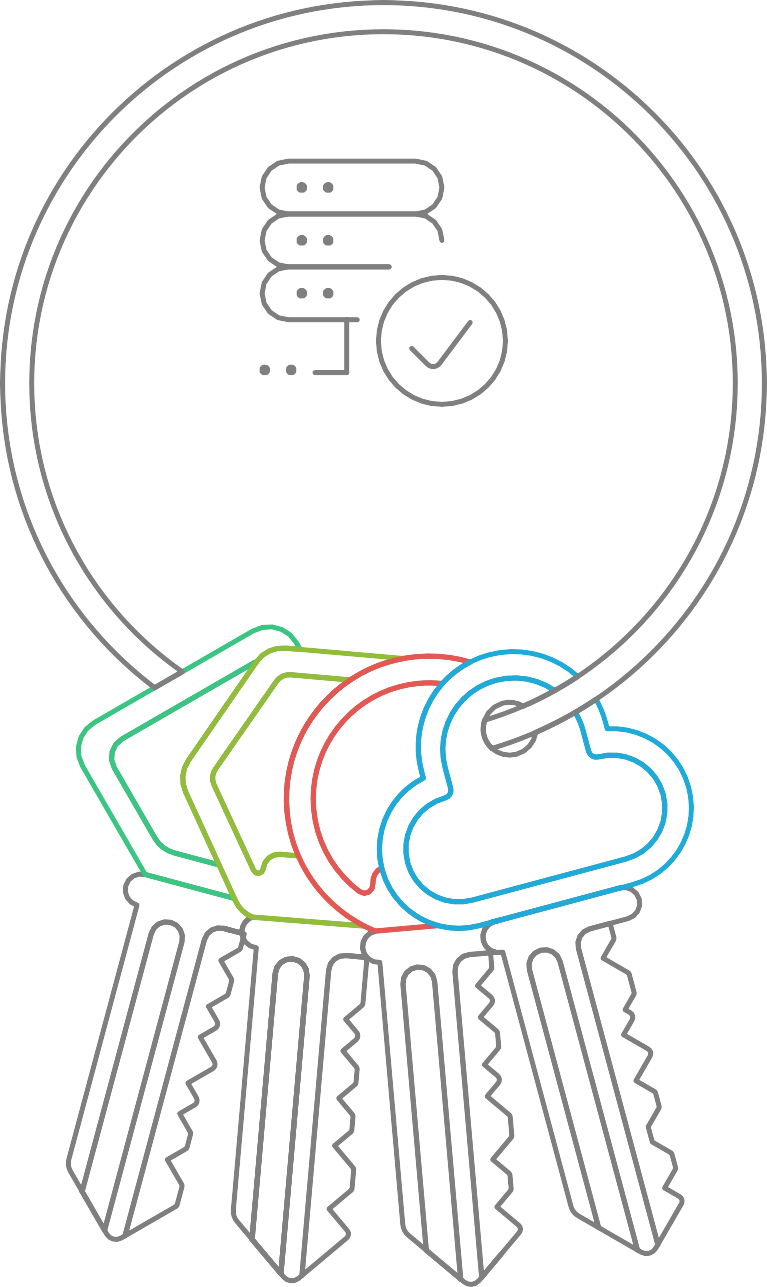
**Scalability**

Allows the system to grow without compromising speed



**Performance**

Ensures swift and efficient system operations



**System Quality**

## 4. Technology Stack:

# Core Salesforce Platform:

* **Salesforce Sales Cloud:** This forms the foundation of the system, providing core CRM functionalities such as account management, contact management, opportunity tracking, and sales process automation. It allows for managing customer interactions, tracking sales pipelines, and generating reports on sales performance.
* **Salesforce Service Cloud (Optional):** If after-sales service, repairs, or customer support are critical aspects of the jewel management process, Service Cloud can be integrated to manage cases, track service requests, and provide a knowledge base for customer self-service.
* **Salesforce Platform (Force.com):** This platform provides the underlying infrastructure for building custom applications and extending Salesforce's functionality. It includes features like custom objects, custom fields, workflows, validation rules, and Apex code execution.

# Programming Languages:

* **Apex:** Salesforce's proprietary programming language, Apex, is essential for developing custom business logic, triggers, and controllers within the Salesforce platform. It's used to automate processes, enforce data validation rules, and integrate with external systems.
* **Lightning Web Components (LWC):** LWC is a modern web development framework for building user interfaces within Salesforce. It uses standard web technologies like HTML, JavaScript, and CSS, making it easier to develop responsive and performant user experiences.
* **JavaScript:** Used extensively within LWC for client-side scripting, handling user interactions, and manipulating data displayed in the user interface.
* **SOQL (Salesforce Object Query Language):** Used to query data stored within Salesforce objects. It's similar to SQL but specifically designed for the Salesforce data model.
* **SOSL (Salesforce Object Search Language):** Used for performing text-based searches across multiple Salesforce objects.

# Frameworks and Libraries:

* **Lightning Design System (LDS):** Salesforce's UI framework provides a set of pre-built components and styling guidelines for creating consistent and visually appealing user interfaces. Using LDS ensures that the application aligns with Salesforce's overall look and feel.
* **JavaScript Libraries (e.g., Lodash, Moment.js):** These libraries can be used to simplify common JavaScript tasks such as data manipulation, date formatting, and utility functions.
* **Testing Frameworks (e.g., Jest, Mocha):** Essential for writing unit tests and integration tests to ensure the quality and reliability of the code.

# Data Modeling:

* **Custom Objects:** Define custom objects to represent jewels, inventory, suppliers, repairs, and other relevant entities specific to the jewel management process.
* **Custom Fields:** Create custom fields on standard and custom objects to store specific attributes of jewels, such as carat weight, cut, clarity, color, metal type, and price.
* **Relationships:** Establish relationships between objects (e.g., one-to-many, many-to-many) to model the relationships between jewels, suppliers, customers, and sales transactions.

# Integration Technologies:

* **REST APIs:** Use Salesforce's REST APIs to integrate with external systems, such as accounting software, e-commerce platforms, and inventory management systems.
* **SOAP APIs:** While REST APIs are generally preferred, SOAP APIs may be necessary for integrating with older systems that do not support REST.
* **Platform Events:** Use platform events to enable real-time communication between Salesforce and external systems. This allows for asynchronous processing of events, such as inventory updates or sales order notifications.
* **Middleware (e.g., MuleSoft):** For complex integrations involving multiple systems, a middleware platform like MuleSoft can be used to orchestrate data flows and transformations.

# User Interface:

* **Lightning Experience:** Salesforce's modern user interface provides a responsive and intuitive experience for users.
* **Lightning Pages:** Use Lightning Pages to create custom dashboards and layouts for different user roles, providing them with the information and tools they need to perform their tasks efficiently.
* **Custom Lightning Components:** Develop custom Lightning Web Components to create specialized user interface elements and interactions that are not available out-of-the-box.

# Reporting and Analytics

* **Salesforce Reports and Dashboards:** Use Salesforce's built-in reporting and dashboarding capabilities to track key metrics, such as sales performance, inventory levels, and customer satisfaction.
* **Einstein Analytics (Tableau CRM):** For more advanced analytics and data visualization, Einstein Analytics can be integrated to provide deeper insights into the data.

# Development Tools:

* **Salesforce CLI (Command Line Interface):** A powerful tool for managing Salesforce metadata, deploying code, and running tests from the command line.
* **Visual Studio Code (VS Code):** A popular code editor with extensions for Salesforce development, providing features like syntax highlighting, code completion, and debugging.
* **Salesforce Extensions for VS Code:** These extensions provide integration with the Salesforce CLI and other development tools, making it easier to develop and deploy code.
* **Git:** A version control system for managing code changes and collaborating with other developers.
* **Scratch Orgs:** Use scratch orgs for development and testing, providing isolated environments for experimenting with new features and changes.

# Security:

* **Salesforce Security Features:** Leverage Salesforce's built-in security features, such as profiles, permission sets, and sharing rules, to control access to data and functionality.
* **Data Encryption:** Encrypt sensitive data at rest and in transit to protect it from unauthorized access.
* **Security Audits:** Regularly conduct security audits to identify and address potential vulnerabilities.

# Deployment:

* **Change Sets:** Use change sets to deploy metadata changes between Salesforce environments.
* **Salesforce DX (Developer Experience):** Use Salesforce DX to manage the entire development lifecycle, from development to deployment.
* **Continuous Integration/Continuous Deployment (CI/CD):** Implement a CI/CD pipeline to automate the build, test, and deployment process.

# Example Scenario: Jewel Inventory Management

To illustrate how these technologies can be applied, consider a scenario where you need to manage jewel inventory.

1. **Custom Object:** Create a custom object called "Jewel" to store information about each jewel.
2. **Custom Fields:** Add custom fields to the "Jewel" object to store attributes like carat weight, cut, clarity, color, metal type, purchase price, and selling price.
3. **LWC:** Develop a Lightning Web Component to display a list of jewels in a table, allowing users to filter and sort the data.
4. **Apex:** Write Apex code to automatically update the inventory count when a jewel is sold or added to stock.
5. **REST API:** Integrate with a supplier's system using REST APIs to automatically import new jewel inventory data.
6. **Reports and Dashboards:** Create reports and dashboards to track inventory levels, sales performance, and profit margins.

This technology stack provides a solid foundation for building a comprehensive jewel management system within Salesforce. By leveraging Salesforce's native capabilities and incorporating custom development where needed, you can create a solution that meets the specific needs of your business. Remember to prioritize security, scalability, and maintainability throughout the development process.