# **Salesforce Design Document**

[Salesforce Design Document 1](#_Toc2045803565)

[1.Overview 1](#_Toc66707259)

[2. Apex Classes 1](#_Toc112536510)

[2.1 Overview 1](#_Toc1275232735)

[2.2 List of Apex Classes 1](#_Toc1548983334)

[2.3 Usage 2](#_Toc541153746)

[3. Flow Components 2](#_Toc215600734)

[3.1 Overview 2](#_Toc810037801)

[3.2 List of Flows 2](#_Toc2002946146)

[3.3 Usage 2](#_Toc1092580809)

[4. Custom Objects 3](#_Toc1648270424)

[4.1 Overview 3](#_Toc566791355)

[4.2 List of Custom Objects 3](#_Toc1118064925)

[4.3 Usage 3](#_Toc456475574)

[5. Custom Tabs 3](#_Toc295458510)

[5.1 Overview 3](#_Toc1865414094)

[5.2 List of Custom Tabs 3](#_Toc950697683)

[5.3 Usage 3](#_Toc1382241091)

[6. Custom Application 4](#_Toc763513417)

[6.1 Overview 4](#_Toc422415115)

[6.2 Custom Application 4](#_Toc1395112446)

[6.3 Usage 4](#_Toc662723533)

[7. FlexiPages 4](#_Toc535157297)

[7.1 Overview 4](#_Toc1953002788)

[7.2 List of FlexiPages 4](#_Toc153916340)

[7.3 Usage 4](#_Toc1488017602)

[8. Quick Actions 5](#_Toc1509540883)

[8.1 Overview 5](#_Toc1391354350)

[8.2 List of Quick Actions 5](#_Toc1233198856)

[8.3 Usage 5](#_Toc1526050725)

[9. Salesforce Version 5](#_Toc371171845)

[10. Functional Workflow 5](#_Toc365442899)

[10.1 Overview 6](#_Toc1847480849)

[10.2 Steps in the Workflow 6](#_Toc946464530)

[11. Conclusion 6](#_Toc1922658263)

## **1.Overview**

SLEEQ is undertaking a high-complexity Salesforce project requiring advanced hierarchical data calculations and rigorous error management. The proposed invocable action will allow Salesforce Flow to process hierarchical data efficiently while dynamically adapting to various data structures. This ensures businesses can leverage real-time calculations without exceeding governor limits or encountering system failures due to unhandled exceptions.

Our newly developed Salesforce Flow Invocable Action empowers organizations with advanced data processing capabilities, ensuring that hierarchical data is handled efficiently while maintaining high system reliability.

## **2. Apex Classes**

### **2.1 Overview**

Apex classes contain the logic for performing hierarchical data calculations, handling requests, and selecting the relevant records. They also include test classes for unit testing the functionality.

### **2.2 List of Apex Classes**

* **HierarchicalDataResponse**: Defines the structure of the response for hierarchical data calculations.
* **HierarchicalDataRequest**: Manages the request for hierarchical data operations.
* **HierarchicalDataSelector**: Determines which records should be processed in hierarchical data calculations.
* **HierarchicalDataCalculator**: Implements the core logic for calculating AVG, MIN, MAX, and SUM for hierarchical data.
* **HierarchicalDataInvocable**: An invocable class that can be called from Salesforce Flow to trigger hierarchical data calculations.
* **HierarchicalDataCalculatorTest**: A test class to ensure the functionality of the HierarchicalDataCalculator.
* **HierarchicalDataInvocableTest**: A test class for the HierarchicalDataInvocable class.
* **HierarchicalDataSelectorTest**: A test class for the HierarchicalDataSelector.
* **HierarchicalDataResponseTest**: A test class for the HierarchicalDataResponse.

### **2.3 Usage**

These classes encapsulate the logic for hierarchical data calculation, record selection, and response formatting. They integrate with other Salesforce components such as Flows and Custom Objects.

## **3. Flow Components**

### **3.1 Overview**

Flows in this package automate the calculation of hierarchical data. They invoke Apex classes to calculate and return results for parent-child records.

### **3.2 List of Flows**

* **Calculate\_Hierarchical\_data**: A Flow for calculating hierarchical data across multiple records.
* **Calculate\_Hierarchical\_Data\_Of\_Parent\_Record**: A Flow that calculates hierarchical data for a specific parent record.

### **3.3 Usage**

These Flows trigger Apex logic to calculate hierarchical data for records, allowing users to automate the data processing workflow.

## **4. Custom Objects**

### **4.1 Overview**

Custom Objects store parent and child record data, which are essential for hierarchical calculations.

### **4.2 List of Custom Objects**

* **Parent\_\_c**: Represents a parent record in a hierarchical structure.
* **Child\_\_c**: Represents a child record related to a parent.

### **4.3 Usage**

The Parent\_\_c and Child\_\_c objects hold data that will be used for hierarchical calculations and are crucial for determining relationships between records.

## **5. Custom Tabs**

### **5.1 Overview**

Custom Tabs provide easy access to the custom objects within the Salesforce user interface.

### **5.2 List of Custom Tabs**

* **Parent\_\_c**: A custom tab for accessing Parent records.
* **Child\_\_c**: A custom tab for accessing Child records.

### **5.3 Usage**

These tabs allow users to navigate quickly to the Parent and Child records to view and manage data involved in hierarchical calculations.

## **6. Custom Application**

### **6.1 Overview**

This custom application integrates all hierarchical data calculation features, providing users with access to data processing tools.

### **6.2 Custom Application**

* **Hierarchical\_Data\_Calculator**: The application that allows users to manage hierarchical data calculations.

### **6.3 Usage**

The Hierarchical\_Data\_Calculator application serves as the central hub for users to interact with the data, access records, and trigger calculation processes.

## **7. FlexiPages**

### **7.1 Overview**

FlexiPages define custom layouts for displaying hierarchical data and calculation results on Salesforce pages.

### **7.2 List of FlexiPages**

* **Hierarchical\_Data\_Calculator**: FlexiPage for managing hierarchical data calculations.
* **Parent\_Record\_Page**: FlexiPage for displaying Parent records and related data.
* **Home\_Page\_DataCalculator**: FlexiPage for the homepage, providing easy access to data calculation tools.

### **7.3 Usage**

These FlexiPages control how hierarchical data and its calculation results are displayed on Salesforce pages, allowing for a streamlined user interface.

## **8. Quick Actions**

### **8.1 Overview**

Quick Actions allow users to perform actions, such as triggering hierarchical data calculations, directly from the Salesforce interface.

### **8.2 List of Quick Actions**

* **Parent\_\_c.Calculate\_Hierarchical\_Data**: A Quick Action for calculating hierarchical data on a Parent record.

### **8.3 Usage**

This Quick Action enables users to perform hierarchical data calculations on Parent records without navigating away from the page.

## **9. Salesforce Version**

* **API Version**: 62.0

This version specifies the metadata API version used for deployment and is compatible with the features provided by Salesforce.

## **10. Functional Workflow**

### **10.1 Overview**

The workflow for calculating hierarchical data involves a series of user interactions and automated processes. The user triggers actions via Quick Actions or Flows, which in turn invoke Apex classes to perform calculations. The results are displayed through FlexiPages and custom tabs.

### **10.2 Steps in the Workflow**

1. **User Interaction**:
   1. Access Parent\_\_c or Child\_\_c records via custom tabs.
   2. Trigger hierarchical data calculations using Quick Actions or Flows.
2. **Flow Execution**:
   1. The Flow calls the relevant Apex classes to perform calculations and return results.
3. **Data Display**:
   1. The results are displayed via the Hierarchical\_Data\_Calculator FlexiPage or on Parent and Child record pages.
4. **Testing**:
   1. Apex test classes ensure the correctness and functionality of the entire process.

## **11. Conclusion**

This package provides a comprehensive solution for calculating hierarchical data in Salesforce. It integrates Apex classes, Flows, Custom Objects, Tabs, FlexiPages, and Quick Actions into a unified system that automates complex data calculations while providing an intuitive interface for users.