**Salesforce Integration**

**Team Mentor – Ankit Kumar**

**Team Members – Shrusti Sanket, Darshan Yadav, Devesh Choudhary**

To create a dashboard using D3.js with Salesforce data fetched from Salesforce using Java as the server-side language, we can follow these steps:

* Set up a Salesforce account and create a connected app to obtain the required credentials for authentication.
* Use the Salesforce REST API or the SOAP API to fetch data from Salesforce. For this example, we will use the REST API.
* Once we have the data, we can use D3.js to create visualizations.

Here's a high-level design to create a dashboard using D3.js with sales cloud data fetched from Salesforce using Java as the server-side language:

1. *Data Collection*: First, we need to collect the sales data from Salesforce using the Salesforce REST API. We can use the Java Salesforce REST API library to fetch data from Salesforce.
2. *Data Processing*: Once we have the data, we need to process it to get the required data for the dashboard. We can use Java to filter, sort, and aggregate the data as per the requirement.
3. *Dashboard Layout*: We can use D3.js to create the dashboard layout. We can use D3.js to create various visualizations such as charts, graphs, and tables to display the sales data.
4. *Dashboard Interactivity*: We can add interactivity to the dashboard using D3.js. We can use D3.js to create filters and sliders to allow the user to interact with the data and see the data change in real-time.
5. *Deployment:* Once the dashboard is ready, we can deploy it on a web server. We can use a cloud-based hosting service such as Heroku or AWS to deploy the dashboard.

Java API

SaleSSaSalesforce

Salesforce API

REST API Response

REST API Request

Dashboard frontend

SaD3.js

Data Visualization

User Interface Interactive Features

SVG Rendering

In this diagram, the Salesforce API provides access to the Salesforce data through a REST API request from the Java API. The Java API is responsible for retrieving the data and returning a REST API response to the frontend, which is built using D3.js. The frontend consists of data visualization components built using SVG rendering, as well as interactive features that allow users to manipulate and explore the data. The user interface is designed to be intuitive and easy to use, with a focus on displaying data in a clear and understandable format. Overall, this design allows for seamless integration of Salesforce data with an interactive and dynamic dashboard frontend.

**Introduction:**

The purpose of this project is to create an interactive dashboard using D3.js, which displays relevant Salesforce data in a visually appealing and user-friendly way. The dashboard will be developed using Java as the server-side language to fetch data from Salesforce and provide it to the frontend.

**Architecture:**

The overall architecture of the application will consist of the following layers:

**Frontend Layer:**

This layer will contain the D3.js code responsible for creating and rendering the dashboard. The frontend will interact with the server-side Java code to fetch data and display it in the dashboard.

**Backend Layer:**

The backend layer will consist of Java code responsible for fetching data from Salesforce and returning it to the frontend. The backend will use Salesforce REST API to connect to the Salesforce data and fetch the required data.

**Salesforce Layer:**

This layer will contain the data source, i.e., the Salesforce data that will be displayed in the dashboard. The Salesforce data will be accessed via REST API by the Java backend layer.

**Key Components:**

1. *D3.js Library:*

The D3.js library will be used to create the interactive visualizations in the dashboard. D3.js is a powerful JavaScript library that enables the creation of custom visualizations.

1. *Java:*

Java will be used as the server-side language to fetch data from Salesforce and provide it to the frontend. The Java code will use Salesforce REST API to access the Salesforce data.

1. *Salesforce REST API:*

Salesforce REST API will be used to access the Salesforce data. The API provides a way to access and interact with Salesforce data via HTTP methods.

1. *Database:*

A database will not be required for this project as Salesforce data will be fetched and displayed directly in the dashboard.

**Conclusion:**

Overall, this high-level design should provide a good starting point for creating a dashboard using D3.js with sales cloud data that is fetched from Salesforce using Java as the server-side language.

This high-level design document outlines the architecture and key components of the D3.js dashboard with Salesforce data using Java as the server-side language. With this design, we can create an interactive dashboard that will help users to make data-driven decisions using Salesforce data.