

## **Introduction**

MBX Corporation has implemented a Sales application in order to manage daily sales operations for their Accounts/Customers.

A Sales User would like to see a custom UI for Opportunity search on the 'Sales' App which will provide them view to Opportunity matching the search criteria and sending them over to 3<sup>rd</sup> party application.

User Adoption for the new page can be measured on the Report & Dashboard which will be showing the Integration Status for Opportunities.

## **Use Case**

- Create custom search page on Opportunity object in Salesforce classic (with sidebar) or Lightning UI containing a Text box. (VF/LWC)
- User enters search criteria which is matched with below mentioned fields:
  1. Name
  2. Account Name
  3. Stage
  4. Type
  5. Amount.
- Add above mentioned fields on page as part of each column on the search table. (VF/LWC)
- Create 2 new fields on Opportunity object mentioned below:
  1. Integration Status – Text (50 char)
  2. Integration Comments – Text (200 char)
- 'Name' & 'Account Name' column will be containing a link to each Opportunity/Account record page which opens in a new tab. (VF/LWC)
- Add pagination to the custom page & display 20 records on a page after the Search. (VF/LWC)

- Create a custom validation on Opportunity to allow users setting Closed Date only if Integration Status value is 'Success' (Validation Rule/Apex Trigger)
- Provide a custom button/link/action on each search row to send your Opportunity to 3<sup>rd</sup> party application. (Apex Class/Apex REST callout)
- Create a new Controller & add it on the page to perform Search operation and sending out HTTP request. (Apex Class/Apex REST callout)
- Check the response from HTTP request, if the Status Code is 200 then update Integration Status & Comments with 'Success' value on the Opportunity. In case of any other Status code store the error information on same fields. (Apex Class/Apex REST callout)
- Create a Test class providing more than 75% code coverage for above mentioned Controller Class. (Apex Test Class)
- Create a class using Mock interface and use it on previous Apex Test class to cover more than 75% code coverage on REST callout. (Apex Test Class)
- Display the error message on custom search page when either no records meet the criteria or HTTP request fails. (Apex class/VF/LWC)
- Create a new Tab to launch the custom search page and add it to the 'Sales' application. (Custom Tab/Lightning App)

## **Report & Dashboard**

- Create a Report on Opportunity object to see all the records successfully sent over to 3<sup>rd</sup> party app.
- Display a Dashboard based on the previous Report to show Opportunity with successful integration.

## **Object ER Diagram**

