## Proposed Solution

## Change the Salesforce standard force.com application to Custom application

* Login with User name and Password.
* Select SFDC Collapsible bar and Changed to Custom application.

##### Technical Overview

**Step: 1**

* Go to Setup Select Create and Click Apps Select Apps
* Click New Button to Create New Custom Application.
* Give App label and App Name.
* Write a Description about the Purpose of the application.
* Click Next.

**Step: 2**

* Click Insert an Image button to select the External Image.
* Select the Drop down for Image Location.
* Pick the Image in list of Images Organization.

**Step: 3**

* Select tabs for the Custom application
* Select the tabs from available tabs
* Select default page Ex: Sales Order Automation
* Click Next button to redirect next Step.
* Click Save to Save the Application.

## Profile Creation in Salesforce

* Go to Setup and click Manage Users
* Select Profiles
* Click New Profile and Clone the Existing profile
* Select Profile Name and Click Save to Save the Profile
* Select the Assignment settings in Profile
* Assign this Profile to New Users

# 

# External Object Access Level Solution & Design :

## External Object Creation

* Go to Setup Click Develop and Select External objects
* Click New External Object
* Select Label ,Plural label and Object Name ,Table name.
* Click Loop Up and select External data source.
* Click Save to Save the External Object.

**External Object Data Source Creation:**

* Go to Setup Click Develop and Select External Data Sources
* Click New External Data Source
* Select External Data source Name
* Select Type, Identity type and authentication protocol .
* Click Save to Save the External Data Source

Data Source Connection :

global class Connection extends

DataSource.Connection {

private DataSource.ConnectionParams connectionInfo;

global Connection (

DataSource.ConnectionParams connectionInfo) {

this.connectionInfo = connectionInfo;

}

override global List<DataSource.Table> sync() {

List<DataSource.Column> Columns= new List<DataSource.Column>();

columns.add(DataSource.Column.text('opportunityID',255));

columns.add(DataSource.Column.text('opportunityName', 255));

columns.add(DataSource.Column.text('winProbId',255));

columns.add(DataSource.Column.text('oppType',255));

columns.add(DataSource.Column.text('oppCategory',255));

columns.add(DataSource.Column.text('firstName',255));

columns.add(DataSource.Column.text('lastName',255));

columns.add(DataSource.Column.text('oppCustomerId',225));

columns.add(DataSource.Column.text('oppAccountId',255));

columns.add(DataSource.Column.text('company\_name',255));

columns.add(DataSource.Column.text('rating\_name',255));

columns.add(DataSource.Column.text('status\_name',255));

columns.add(DataSource.Column.text('rn',255));

columns.add(DataSource.Column.text('customer\_name',255));

columns.add(DataSource.Column.url('DisplayUrl'));

columns.add(DataSource.Column.text('status\_id',255));

columns.add(DataSource.Column.text('created\_date',255));

columns.add(DataSource.Column.text('created\_by',255));

columns.add(DataSource.Column.text('assign\_to',255));

columns.add(DataSource.Column.text('ExternalId',255));

// columns.add(DataSource.Column.text('oppId',255));

List<DataSource.Table> tables = new List<DataSource.Table>();

tables.add(DataSource.Table.get('opportunity', 'title', Columns));

return tables;

}

override global DataSource.TableResult query(DataSource.QueryContext queryContext) {

List<Map<String, Object>> repos = DataSource.QueryUtils.process(queryContext, getOrgRepos('developerforce'));

DataSource.TableResult tableResult = DataSource.TableResult.get(queryContext, repos);

return tableResult;

}

override global List<DataSource.TableResult> search(DataSource.SearchContext searchContext) {

return DataSource.SearchUtils.searchByName(searchContext, this);

}

public List<Map<String, Object>> getOrgRepos(String org)

{

Http httpProtocol = new Http();

HttpRequest request = new HttpRequest();

//'https://10.90.13.25:28880/SfdcService/';

String url ='https://devqa-tt1.excelacom.in:28880/SfdcService/';

//String url = 'https://202.9.75.12:28880/SfdcService/';

//String url = 'https://202.9.75.12:28880/SfdcService/';

//String url ='http://exind3020-44:8090/udt-service/services/opportunityService/retrieveOppDetailsByOppid?oppId=1062468';

//String url = 'https://api.github.com/orgs/' + org + '/repos?per\_page=1000';

// String url = 'https://www.screener.in';

request.setEndPoint(url);

request.setMethod('GET');

request.setBody('operationName = retrieve');

//operationName = retrieve

request.setHeader('Content-Type', 'application/json');

request.setHeader('Accept','application/json');

HttpResponse response = httpProtocol.send(request);

system.debug('Response body result is+++: ' + response);

List<Map<String, Object>> repos = new List<Map<String, Object>>();

if(response.getstatusCode() == 200 && response.getbody() != null){

// system.debug('Response body result is+++: ' + response.getBody());

for (Object item : (List<Object>)JSON.deserializeUntyped(response.getBody()))

{

Map<String, Object> repo = (Map<String, Object>)item;

repo.put('winProbId', repo.get('WIN\_PROB\_NAME'));

repo.put('oppType', repo.get('L\_TYPE\_NAME'));

repo.put('customer\_name',repo.get('CUSTOMER\_NAME'));

repo.put('oppAccountId', repo.get('ACCOUNT\_ID'));

repo.put('company\_name', repo.get('COMPANY\_NAME'));

repo.put('oppCustomerId', repo.get('CUSTOMER\_ID'));

repo.put('rn',repo.get('RN'));

repo.put('lastName', repo.get('LAST\_NAME'));

repo.put('opportunityName', repo.get('OPPORTUNITY\_NAME'));

repo.put('status\_name', repo.get('OPP\_STATUS\_NAME'));

repo.put('created\_by', repo.get('CREATED\_BY'));

repo.put('created\_date', repo.get('CREATED\_DATE'));

repo.put('assign\_to', repo.get('ASSIGN\_TO'));

repo.put('firstName', repo.get('FIRST\_NAME'));

repo.put('rating\_name',repo.get('RATING\_NAME'));

repo.put('opportunityID', repo.get('OPPORTUNITY\_ID'));

repo.put('ExternalId', repo.get('ID'));

repo.put('DisplayUrl', repo.get('html\_url'));

// repo.put('oppCategory', repo.get('oppCategory'));

repos.add(repo);

}

}

return repos;

}

}

global class ConnectionProvider

extends DataSource.Provider {

override global List<DataSource.AuthenticationCapability>

getAuthenticationCapabilities() {

List<DataSource.AuthenticationCapability> capabilities =

new List<DataSource.AuthenticationCapability>();

capabilities.add(

DataSource.AuthenticationCapability.OAUTH);

capabilities.add(

DataSource.AuthenticationCapability.ANONYMOUS);

return capabilities;

}

override global List<DataSource.Capability>

getCapabilities() {

List<DataSource.Capability> capabilities =

new List<DataSource.Capability>();

capabilities.add(DataSource.Capability.ROW\_QUERY);

//capabilities.add(DataSource.Capability.SEARCH);

return capabilities;

}

override global DataSource.Connection getConnection(

DataSource.ConnectionParams connectionParams)

{

return new Connection(connectionParams);

}

}

**External Object Data Synchronization:**

* Go to Setup Click Develop and Select External Data Sources
* Click New External Data Source
* Select External Data source Name
* Select Type, Identity type and authentication protocol .
* Click Save to Save the External Data Source

## Object Tab Creation:

* Go to Setup Click Create and select Tabs
* Click New Tab creation
* Select Tab Name, Tab Style
* Click Save to Save the External Object Tab .