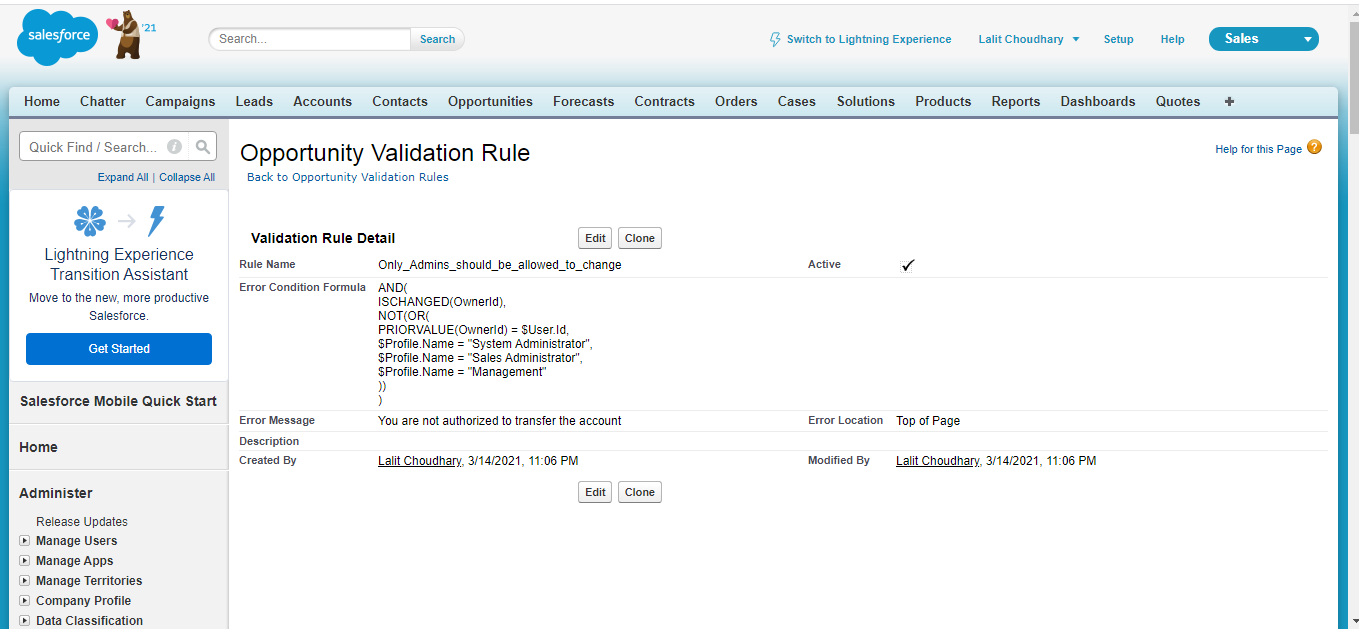
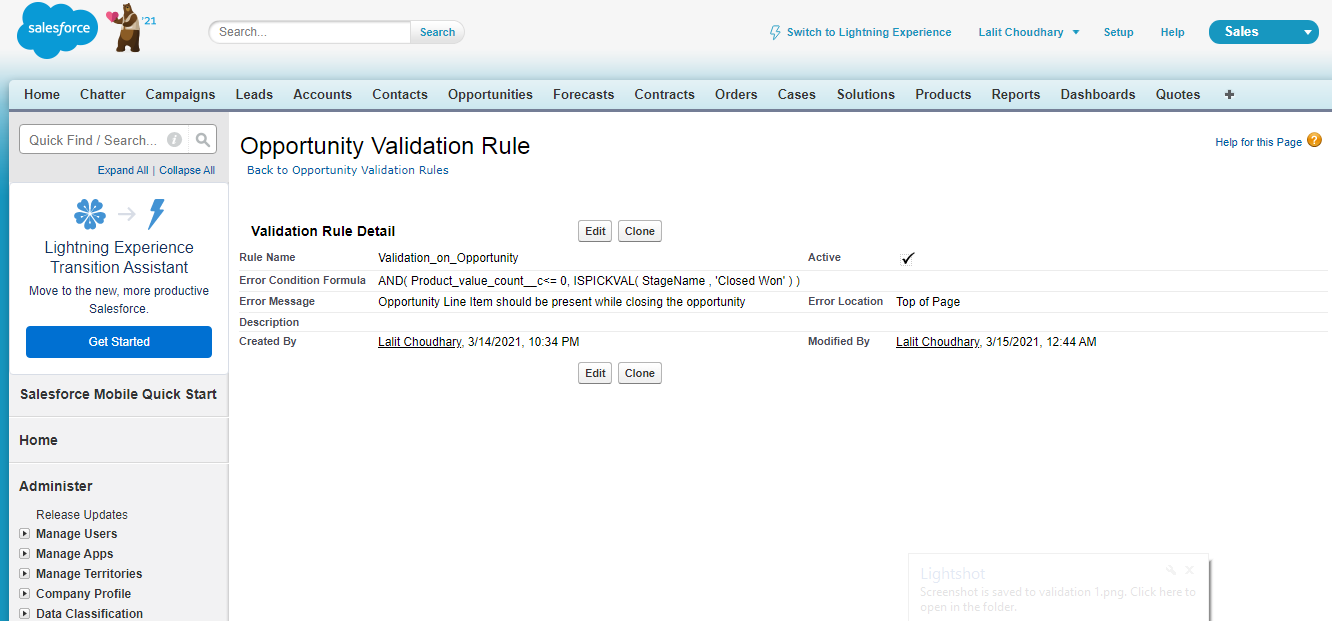
1. **Validation Rules-**

**a)** Only Admins should be allowed to change the account owner. Error Message: "You are not   authorized to transfer the account.

****

b) While closing the opportunity (‘Closed Won') an Opportunity Line Item should be present. If an opportunity line item is not present under an opportunity, it should give the error message "Opportunity Line Item should be present while closing the opportunity."

****

1. **Workflow Rule**

If Amount > $10,000 send an email alert on opportunity **closed won (Stage field value)** to pgoyal@kloudrac.com and msharma@kloudrac.com

Email Template.

Subject: Deal closed Alert - $Amount - Opportunity Name

Opportunity Name:

Account Name:

Account Owner:

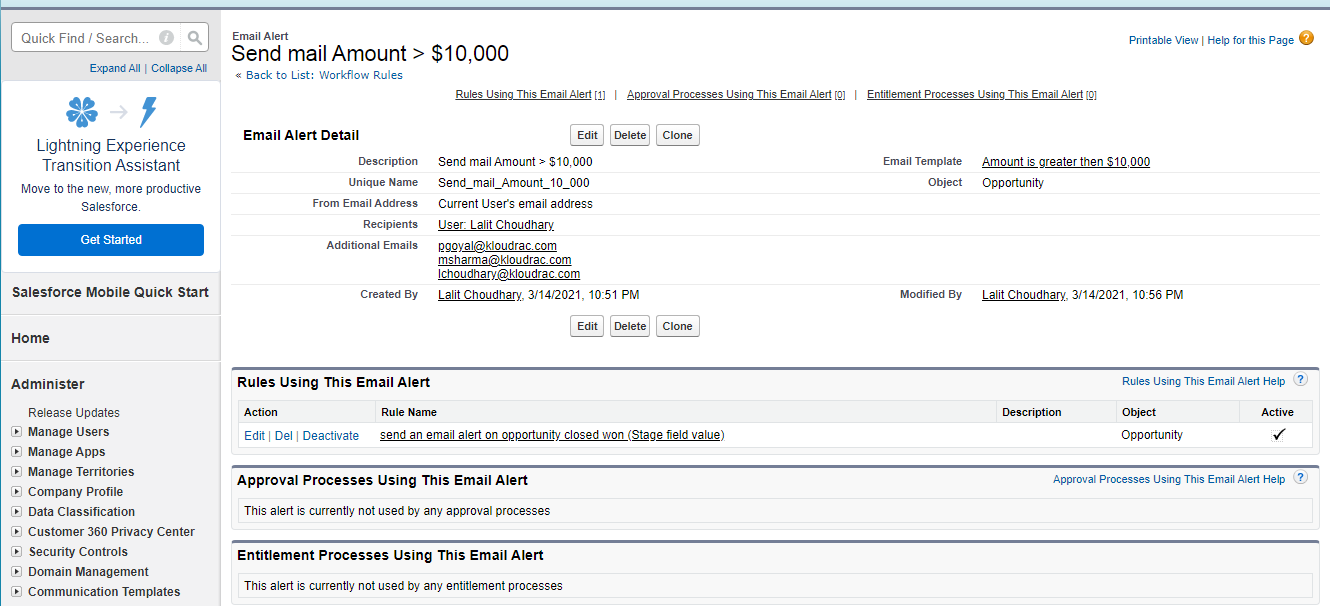
Amount:

Close Date:

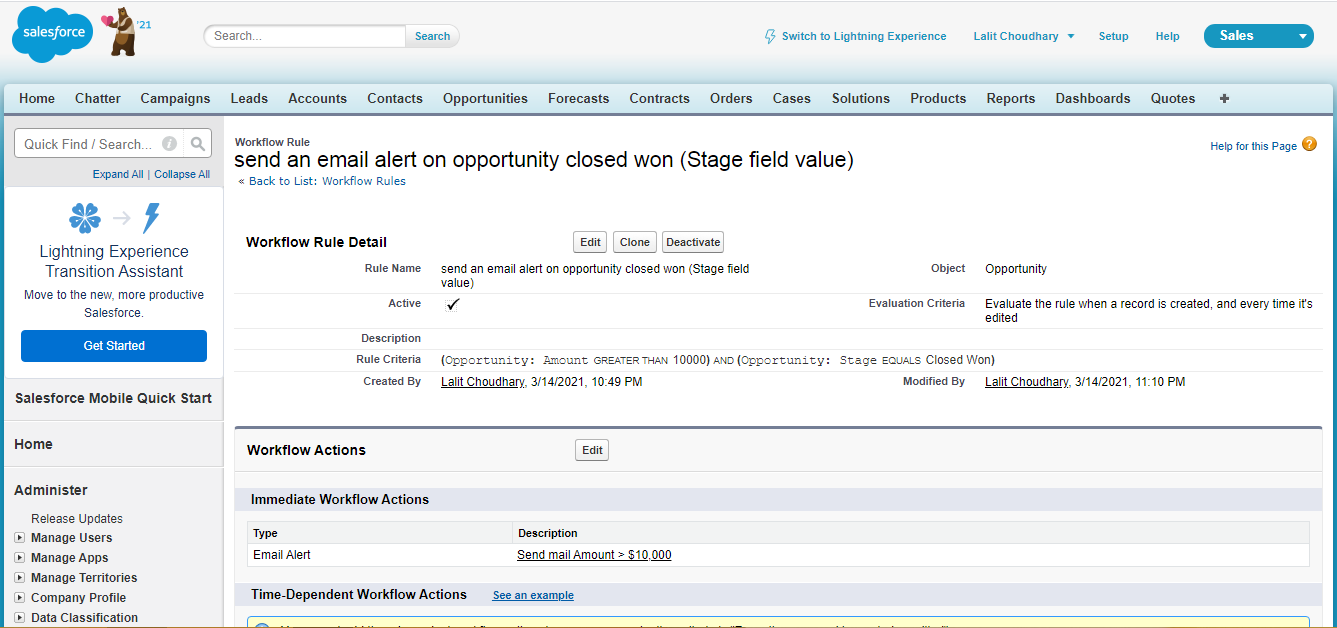
Best Regards

Salesforce Team – Kloudrac

**Email alert:-**

****

**Workflow:-**

****

**Process Builder:-**

**Kloudrac Softwares would like to set up a process where an event will get created on opportunity creation and an automated approval will go to the opportunity owner's manager whenever discount > 20% on opportunity.**

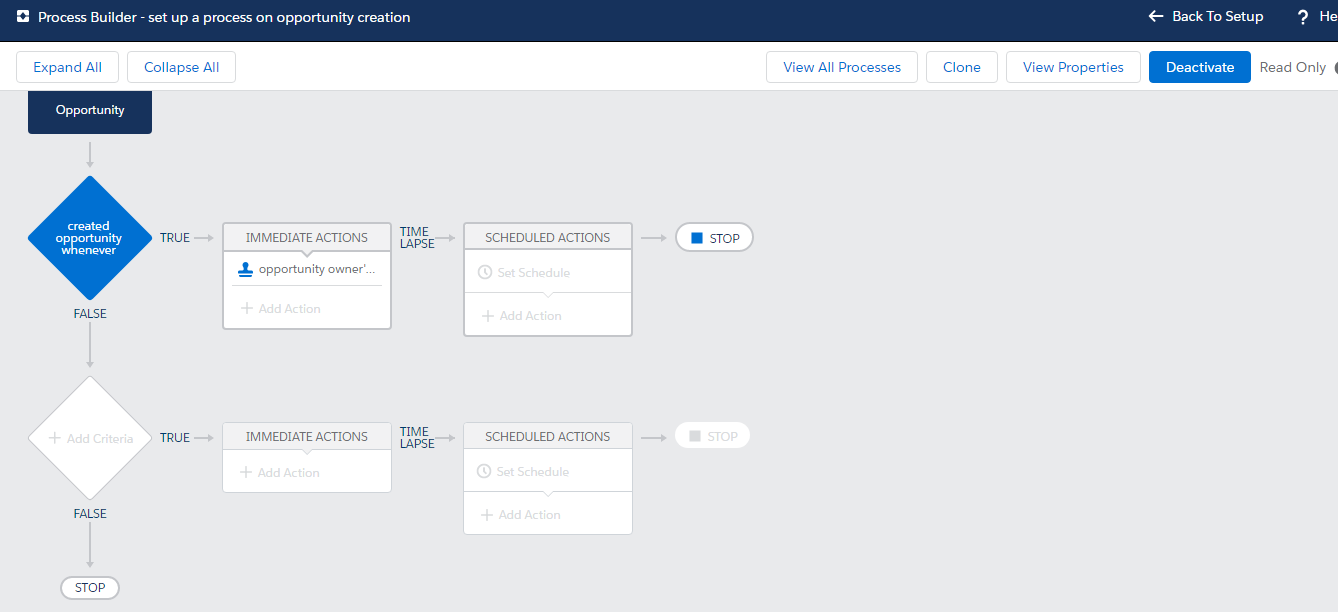
**There is a field Discount, Approval Status on Opportunity.**

**Default value for Discount - "Approval Not Required".**

**When Discount is greater than your allowed matrix - "Approval Required"**

**When submitted for approval - "Approval Pending"**

**When Approved / Rejected - "Approved / Rejected”**



1. **Trigger**
2. **Write a trigger for showing the Quote Line Item count on the related Account.**

**Trigger:-**

**//Trigger Created by : Lalit Choudhary Created Date : 15-03-2021//**

**//Company : Kloudrac //**

trigger CountQuoteLineItem on Quote (after insert, after update){

    if(Trigger.isafter && trigger.isinsert){

    TriggerhandlerCountQuote.m1(trigger.new);

    }

    if(Trigger.isafter && trigger.isupdate){

    TriggerhandlerCountQuote.m1(trigger.new);

}

}

Class:-

**//Class Created by : Lalit Choudhary Created Date : 15-03-2021//**

**//Company : Kloudrac //**

public class TriggerhandlerCountQuote {

    Public static void m1(List<Quote> newQuotelist){

        set<id> Accuntids = new set<id>();

        for(Quote Quot : newQuotelist){

            if(Quot.Accountid != null){

                Accuntids.add(Quot.Accountid);

            }

        }

        //List of Account//

        list<Account> Accupdate = new list<Account>();

        for(Account Acc : [select id,Quote\_Line\_Item\_count\_\_c, name, (select id from Quote) from Account where id in : Accountids]){

            Acc.Quote\_Line\_Item\_count\_\_c = Acc.Quot.size();

            Accupdate.add(acc);

        }

        if(!Accupdate.isempty()){

            Database.update(Accupdate);

        }

    }

    }

**3) Quote and Quote Line Item should get auto created while Opportunity revenue is greater than 100000.**

**Trigger:-**

**//Trigger Created by : Lalit Choudhary Created Date : 15-03-2021//**

**//Company : Kloudrac**

trigger FacturationCreator on Opportunity (before insert) {

List<Id> oppIds = new List<Id>();

for (Opportunity o : Trigger.new)

oppIds.add(o.id);

CreateQuote.createQuote(oppIds);

}

Class:-

**//Class Created by : Lalit Choudhary Created Date : 15-03-2021//**

**//Company : Kloudrac //**

public class CreateQuote {

    public static void createQuote(List<Id> oppIds) {

        List<Quote> lstQ = new List<Quote>();

**//Fetching all the OLIs belonging to the Opportunities of Trigger.new**

        List<OpportunityLineItem> olis =[select id, OpportunityId, quantity, PriceBookEntry.Product2Id, UnitPrice, PricebookentryId from OpportunityLineItem where OpportunityId in :oppIds];

        Map<Id,List<OpportunityLineItem>> mapOppIdOli = new Map<Id,List<OpportunityLineItem>>();

        for (OpportunityLineItem oli : olis) {

            if (mapOppIdOli.containsKey(oli.OpportunityId) {

                mapOppIdOli.get(oli.OpportunityId).add(oli);

            } else {

                List<OpportunityLineItem> lstOlis = new List<OpportunityLineItem>();

                lstOlis.add(oli);

                mapOppIdOli.put(oli.OpportunityId, lstOlis);

            }

                }

**//Preparing the new quotes - one for each Opportunity**

                List<Opportunity> opps =[select id, name, CloseDate from Opportunity where Id in :oppIds];

                for (Opportunity opp : opps) {

                    if (opp.Type == 'Event') {

                        Quote quo : new Quote();

                        quo.name = 'Quote-' + opp.name;

                        quo.Montant\_1\_re\_ch\_ance\_c = 1000;

                        quo.Date\_de\_Facturec = opp.CloseDate;

                        quo.Date\_1\_re\_ch\_ance\_c = opp.CloseDate;

                        quo.opportunityId = opp.id;

                        lstQ.add(quo);

                    }

                }

                insert lstQ;

                List<QuoteLineItem> lstQLI = new List<QuoteLineItem>();

                for (Quote quo : lstQ) {

                    List<OpportunityLineItem> lstOlis = mapOppIdOli.get(quo.OpportunityId);

                    if (lstOlis != null) {

                        for (OpportunityLineItem oli : lstOlis) {

                            QuoteLineItem qli = new QuoteLineItem();

                            qli.quoteId = quo.Id;

                            qli.UnitPrice = oli.UnitPrice;

                            qli.Product2Id = oli.PriceBookEntry.Product2Id;

                            qli.Quantity = oli.Quantity;

                            qli.PriceBookentryid = oli.PriceBookentryId;

                            lstQLI.add(qli);

                        }

                    }

                }

                //Inserting the new QLIs

                insert lstQLI;

                }

                }

                }