

SALESFORCE- CASE STUDY

High-Level Requirements:

1.Initial Setup, Configuration and Data Migration

Created an App named **ABC Limited**

Created Accounts, Contacts, Products and Assets in excel sheets. Using Data Import wizard and Data Loader.io tool imported the data to my org. Imported using insertion and operation and then mapped fields.

Accounts:

	A	B	C	D	E
1	Account Name	Type	Industry	Prospect Rating	
2	Mindtree Ltd Kolkata	Prospect	Technology	Hot	
3	Policy Protectors	Prospect	Insurance	Hot	
4	HDFC Bank	Prospect	Finance	Hot	
5					
6					

Contacts:

	A	B	C	D	E	F
1	First Name	Last Name	Phone	Lead Source	Salutation	Account Name
2	Ankush	Rathore	(982) 473-6102	Web	Mr.	Policy Protectors
3	Arjun	Rampal	(726) 718-7328	Phone Inquiry	Mr.	HDFC Bank
4	Shruti	Patnayak	(726) 479-8327	Phone Inquiry	Mr.	Mindtree Ltd Kolkata
5						
6						
7						
8						

Assets:

	A	B	C	D	E
1	Asset Name	Product Code	Status	Account Name	Contact Name
2	Watch asset	7139WL01	Shipped	Mindtree Ltd Kolkata	Shruti Patnayak
3	Insurance Asset	LA19150	Shipped	Policy Protectors	Ankush Rathore
4	Credit Card Asset	ACC0001	Shipped	HDFC Bank	Arjun Rampal
5					
6					

Products:

	A	B
1	Product Name	Product Code
2	Versatile 2 Silver Dial Leather Strap Watch	7139WL01
3	Atlas Credit Card	ACC0001
4	Life and Accident Insurance Plan	LA19150
5		

Asset Example



Asset Watch asset

Account
[Mindtree Ltd Kolkata](#)

Contact
[Shruti Patnayak](#)

Quantity
1.00

Related

Details

Asset Name	Account
Watch asset	Mindtree Ltd Kolkata
Product	Contact
Versatile 2 Silver Dial Leather Strap Watch	Shruti Patnayak
Product Code	Competitor Asset
7139WL01	<input type="checkbox"/>
Serial Number	Purchase Date
10001	7/11/2021
Install Date	Usage End Date
	7/11/2022
Status	
Shipped	
Quantity	
1.00	
Price	
\$200	
Created By	Last Modified By

Entitlement example

Feed	Amount Fed	Weight Gain	Weighted Average
...

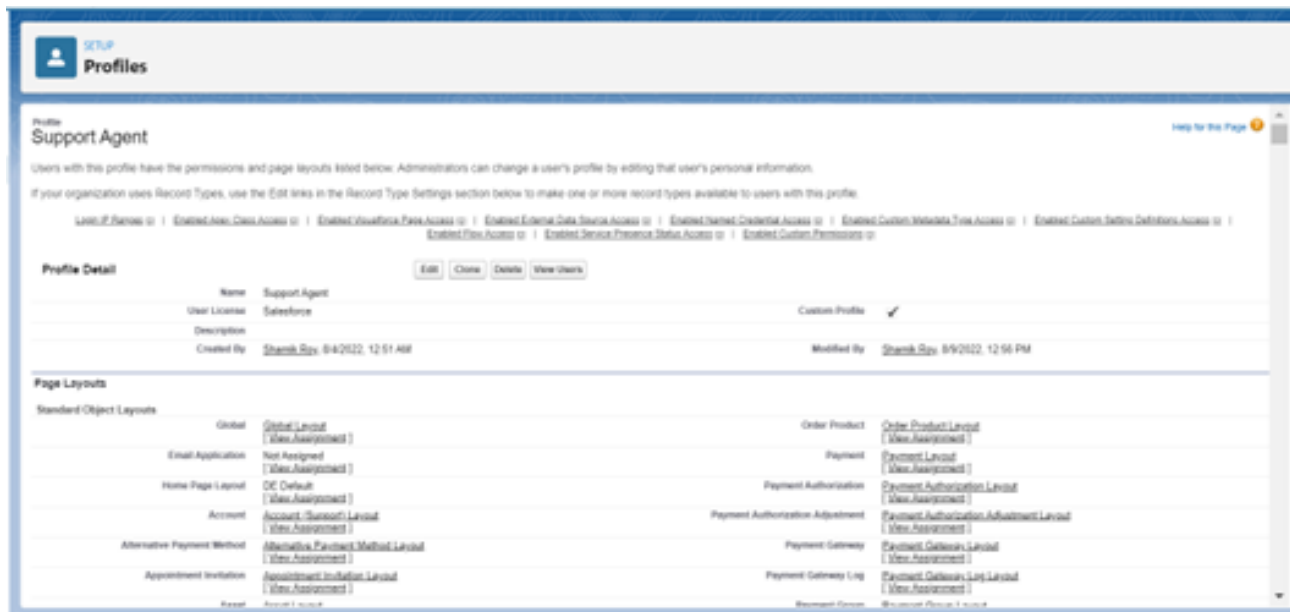
Created an user with the Business Admin role

The screenshot shows the Salesforce Setup Users page for a user named Lee Joseph. The page is titled "User: Lee Joseph" and includes a "User Probability for this Page" indicator. The user's details are as follows:

Field	Value	Field	Value
Name	Lee Joseph	Role	Admin
Alias	jlee	User License	Salesforce
Email	shanku@my@gmail.com	Profile	Business Admin
Username	shanku.gu@my@gmail.com	Active	<input checked="" type="checkbox"/>
Nickname	User16696366067615336745	Marketing User	<input type="checkbox"/>
Title		Offline User	<input type="checkbox"/>
Company		Knowledge User	<input type="checkbox"/>
Department		Flow User	<input type="checkbox"/>
Division		Service Cloud User	<input type="checkbox"/>
Address		Site.com Contributor User	<input type="checkbox"/>
Time Zone	(GMT+05:30) India Standard Time (Asia/Kolkata)	Site.com Publisher User	<input type="checkbox"/>
Locale	English (United States)	WDC User	<input type="checkbox"/>
Language	English	Mobile Push Registrations	View
Delegated Approver		Data.com User Type	1
Manager		Accessibility Mode (Classic Only)	1
Receive Approval Request Emails	Only if I am an approver	Debug Mode	1
Federation ID		High-Contrast Palette on Charts	1
App Registration: One Time Password Authenticator	1	Load Lightning Pages While Scrolling	1
App Registration: Salesforce Authenticator	1	Send Apex Warning Emails	<input type="checkbox"/>

Profile-2

Created Support Agent profile and gave read access on all Service cloud objects and Full access (i.e., CRUD capabilities) on Case Object.




Created an user with the Support Agent role



Role Hierarchy:

Configured a hierarchy where Business Admin and Support Agent are at the same level.

<div>  <div> <div>Setup</div> <div>Roles</div> </div> </div>			
Roles			
Below is a list of the roles for your organization. You can view more information by clicking the role link.			
<div> <div>New Role</div> <div>Show in list view</div> </div>			
Action	Role	Reports To	Report Display Name
Edit / Del / Assign	CEO		CEO
Edit / Del / Assign	Admin	CEO	Business Admin
Edit / Del / Assign	CEO	CEO	
Edit / Del / Assign	COO	CEO	
Edit / Del / Assign	MD	CEO	
Edit / Del / Assign	Support	CEO	Support Agent
Edit / Del / Assign	ApexG	Support	
Edit / Del / Assign	ApexG	Support	
Edit / Del / Assign	SVP Customer Service & Support	CEO	SVP Customer Service & Support
Edit / Del / Assign	Customer Support International	SVP Customer Service & Support	Customer Support International
Edit / Del / Assign	Customer Support North America	SVP Customer Service & Support	Customer Support North America
Edit / Del / Assign	Installation & Repair Services	SVP Customer Service & Support	Installation & Repair Services
Edit / Del / Assign	SVP Human Resources	CEO	SVP Human Resources
Edit / Del / Assign	SVP Sales & Marketing	CEO	SVP Sales & Marketing
Edit / Del / Assign	VP International Sales	SVP Sales & Marketing	VP International Sales
Edit / Del / Assign	VP Marketing	SVP Sales & Marketing	VP Marketing
Edit / Del / Assign	Marketing Team	VP Marketing	Marketing Team
Edit / Del / Assign	VP North American Sales	SVP Sales & Marketing	VP North American Sales

Milestones

Created milestones for Entitlement Process

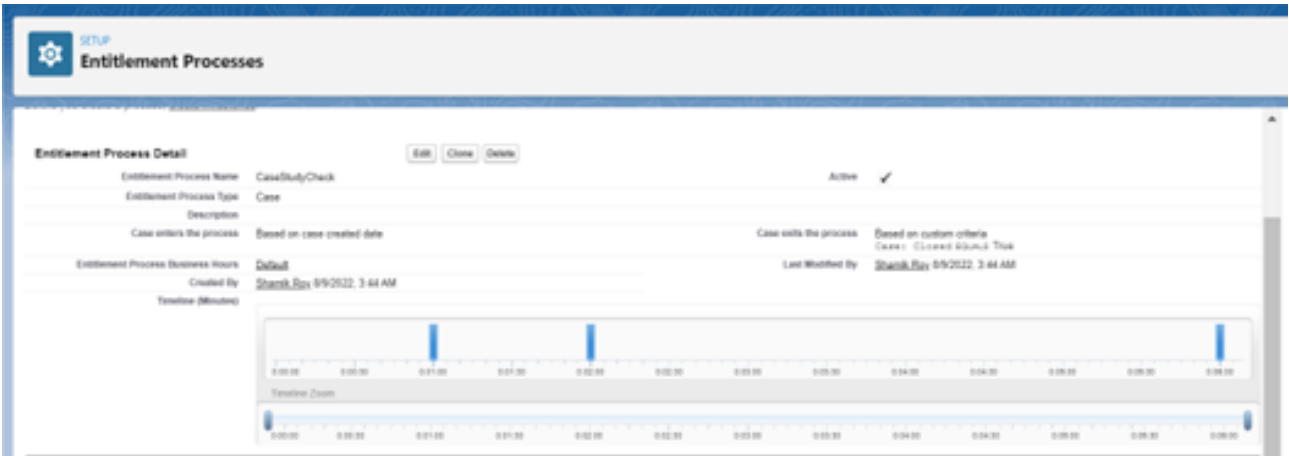
<div> <div>Milestone</div> <div>New</div> </div>							
Order	Name	Recurrence Type	Criteria	Minutes To Complete Milestone	Time Trigger Apex Class	Start Time	Business Hours
1	First Response Time	No Recurrence		60		Entitlement Process	Default
2	Resolution Time	No Recurrence		120		Entitlement Process	Default
3	Closure Time	No Recurrence		360		Entitlement Process	Default

[Back To Top](#)

Always show me [more records per related list](#)

Entitlement Process

Configured Entitlement Process with above milestones which will be active once Case is created and should exit once Case is closed.



Choose Case Type

Choose a Case type

Incident

Service Request

Cancel

Choose Case Type component code

```
<aura:component implements="force:appHostable,flexipage:availableForAllPageTypes,flexipage:availableForPageTypes">
<aura:attribute name="isModalOpen" type="boolean" default="true"/>
    <div class="slds-m-around_xx-large">
        <aura:if isTrue="{!v.isModalOpen}">
            <!-- Modal/Popup Box starts here-->
            <section role="dialog" tabindex="-1" aria-labelledby="modal-heading-01" aria-modal="true">
                <div class="slds-modal__container">
                    <lightning:navigation aura:id="navService"/>
                    <!-- Modal/Popup Box Header Starts here-->
                    <header class="slds-modal__header">
                        <lightning:buttonIcon iconName="utility:close"
                            onclick="{!c.handleCancel}"
                            alternativeText="close"
                            variant="bare-inverse"/>
                    </header>
                </div>
            </section>
        </aura:if>
    </div>
</aura:component>
```

JavaScript Controller

```
{
    openModel: function(component, event, helper)
    {
        component.set("v.isModalOpen", true);
    },
    closeModel: function(component, event, helper)
    {
        component.set("v.isModalOpen", false);
    },
    handleCancel: function(cmp, event, helper)
    {
        var navService = cmp.find("navService");
        var pageReference = {
            type: 'standard__objectPage',
            attributes: {
```

There are two types of cases to choose:-

1. **Incident Type-** For incident type: Standard page layout is used. I used lookup filter to create lookup for the account name related to contacts and entitlements.

New Case: Incident

Case Information

Case Owner

Shamik Roy

* Status

New

Case Number

Priority

Medium

Account Name

Mindtree Ltd Kolkata

* Case Origin

Email

Contact Name

Shruti Patnayak

Entitlement Name

Watch guarantee

Type

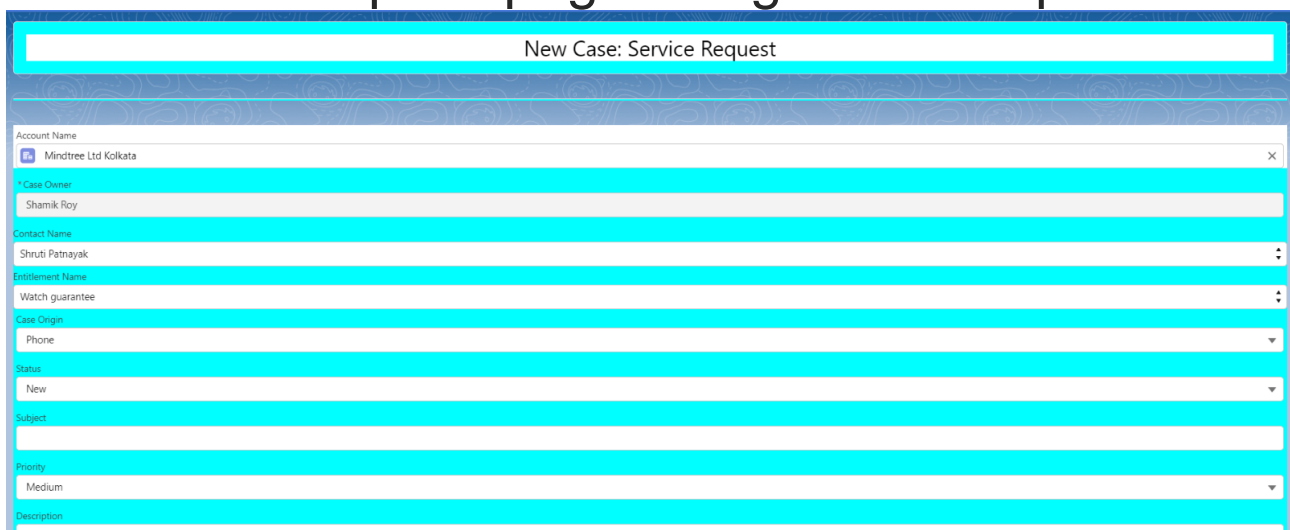
Incident

Cancel

Save & New

Save

1. **Service Request Type-** Created a custom service request page using aura component.



Service Request Type component code:

```
<aura:component controller="CaseStudyapex" implements="force:lightningQuickAction,flexipage:availableForAllPageTypes" >
    <aura:attribute name="contactList" type="list"/>
    <aura:attribute name="entitlementList" type="list"/>
    <div class="slds-box">
        <div class="slds-text-heading_large">
            <div class="slds-align_absolute-center" style="color:black;background-color:white">New Case
            </div></div>
        <hr></hr>
        <lightning:recordEditForm aura:id="recordEditForm"
                                objectApiName="Case">
            <lightning:messages />
            <lightning:inputField style="color:black;background-color:white" name="selectAcc" aura:
            <lightning:inputField fieldName="OwnerId" />
        </lightning:recordEditForm>
    </div>
</aura:component>
```

Apex class code:

```
public class CaseStudyapex
{
    @AuraEnabled
    public static list<account> fetchAccount(){
        list<account> acclist= new list<account>();
        acclist=[select name from account];
        return acclist;
    }
    @AuraEnabled
    public static list<contact> getContact(id accid){
        list<contact> conlist= new list<contact>();
        conlist=[select Id, Name from contact where accountid =:accid];
        return conlist;
    }
}
```

Java Script code

```
({
    getData:function(component,event,helper)
    {
        helper.helperGetData(component,event,helper)
        helper.helperGetEnt(component,event,helper)
    },
    handleSuccess: function(cmp, event, helper)
    {
        var navService = cmp.find("navService");
        var pageReference = {
            type: 'standard__objectPage',
            attributes: {
                objectApiName: 'Case',
                actionName: 'home'
            }
        }
    }
})
```

Helper code

```

|({
  helperGetData : function(component,event,helper)
  {
    var acid=component.find('AccName').get('v.value');
    console.log(acid);
    var action=component.get('c.getContact');
    action.setParams({
      accid:acid
    });
    action.setCallback(this,function(response){
      console.log('response==> '+JSON.stringify(response.getReturnValue()));
      if(response.getReturnValue()!=null){
        console.log('success');
        component.set('v.contactList',response.getReturnValue());
      }
    });
  }
}

```

Queue- Created a queue where cases will automatically get assigned to the queue after a case gets created.

Queue
Case Management Support Team Help for this Page

[Edit](#) [Delete](#)

Label	Case Management Support Team	Queue Name	Case_Management_Support_Team
Queue Email		Send Email to Members	<input checked="" type="checkbox"/>
Supported Objects	Case		
Created By	Shamik Roy, 8/6/2022, 12:25 AM	Modified By	Shamik Roy, 8/21/2022, 10:52 PM

[View All Users](#)

Name	Type
Sarjom Murmu	User

Case Assignment Rule- Case assignment should be automated with members being notified once a case is assigned.

Case Assignment Rule
Caserule1
[Help for this Page](#)

Add rule entries that specify the criteria used to route cases. You can reorder rule entries on this page after you create them.

Rule Detail
Edit

Rule Name	Caserule1	Active	✓
Created By	Shamik Roy, 8/16/2022, 4:36 AM	Modified By	Shamik Roy, 8/16/2022, 4:38 AM

Edit

Rule Entries
New
Reorder

Action	Order	Criteria	Assign To	Email
Edit Del	1	Case: Status EQUALS New	Case Management Support Team	✓

- Created a Validation rule where an owner must take the ownership before starting working on a case

Case Validation Rule
[Back to Case Validation Rules](#)
[Help for this Page](#)

Validation Rule Detail
Edit
Clone

Rule Name	Case_Validation	Active	✓
Error Condition Formula	LEFT(OwnerId,3) <> '005'	Error Location	Top of Page
Error Message	Case owner not valid		
Description			
Created By	Shamik Roy, 8/6/2022, 12:35 AM	Modified By	Shamik Roy, 8/30/2022, 4:23 AM

Edit Clone

- Case will show error until ownership is taken

Related

Open Activities (0)

New Task New Event

Activity History (0)

Log a Call View All

Case Milestones (3)

Name	Target Date	Completion Date	Time Remaining (Min:Sec)
First Response Time	8/30/2022, 6:05 AM		59:56
Resolution Time	8/30/2022, 7:05 AM		119:56
Closure Time	8/30/2022, 11:05 AM		359:56

View All

Milestones

First Response Time

59 min 17 sec remaining

Mark Completed

Show More

Details

We hit a snag.

Review the errors on this page.

- Case owner not valid

Account Name Cancel Save

- Created an action button to Take Ownership before starting to work on a case using apex class

```

public class Assignowner
{
    @AuraEnabled
    public static Boolean assignToCurrentUser(Id recordId)
    {
        Case caseObj=[Select Id,OwnerId from Case where Id=:recordId];
        if(caseObj.OwnerId!=UserInfo.getUserId())
        {
            caseObj.OwnerId = UserInfo.getUserId();
            update caseObj;
            return true;
        }
        else
        {

```

Taking Ownership component code

```

<aura:component implements="force:lightningQuickAction,force:hasRecordId,lightning:isUrlAddressable">
<aura:attribute name="loading" type="Boolean" default="true" />
<aura:handler name="init" action="{!c.myAction}" value="{!this}" />
<aura:if isTrue="{!v.loading}">
    <div class="slds-spinner_container">
        <div class="slds-spinner slds-spinner-brand slds-spinner-medium" role="status">
            <div class="slds-spinner__dot-a"></div>
            <div class="slds-spinner__dot-b"></div>
        </div>
    </div>
</aura:if>
</aura:component>

```

AssignOwnership		
Ctrl + Shift + 1	COMPONENT	
Ctrl + Shift + 2	CONTROLLER	
Ctrl + Shift + 3	HELPER	
Ctrl + Shift + 4	STYLE	Create
Ctrl + Shift + 5	DOCUMENTATION	Create
Ctrl + Shift + 6	RENDERER	Create
Ctrl + Shift + 7	DESIGN	Create
Ctrl + Shift + 8	SVG	Create
Bundle Version Settings		

JS Controller code

```

|{
  myAction : function(component, event, helper)
  {
    var action = component.get("c.assignToCurrentUser");
    var params = {"recordId": component.get("v.recordId")};
    action.setParams(params);
    action.setCallback(this, function(response)
    {
      var state = response.getState();
      component.set("v.loading", false);
      var result=response.getReturnValue();
      if(result===true)
      {
        console.log(response.getReturnValue());
      }
    });
  }
}

```

AssignOwnership			»
Ctrl + Shift + 1	COMPONENT		
Ctrl + Shift + 2	CONTROLLER		
Ctrl + Shift + 3	HELPER		
Ctrl + Shift + 4	STYLE	Create	
Ctrl + Shift + 5	DOCUMENTATION	Create	
Ctrl + Shift + 6	RENDERER	Create	
Ctrl + Shift + 7	DESIGN	Create	
Ctrl + Shift + 8	SVG	Create	
Bundle Version Settings			

Helper Code

```

|{
  helperMethod : function(toasttype, title, message, data)
  {
    data = data || [];
    var toastEvent = $A.get("e.force:showToast");
    toastEvent.setParams({
      type: toasttype,
      mode: "sticky",
      title: title,
      message: message,
      messageTemplate: message,
      messageTemplateData: data
    });
    toastEvent.fire();
  }
}

```

AssignOwnership			»
Ctrl + Shift + 1	COMPONENT		
Ctrl + Shift + 2	CONTROLLER		
Ctrl + Shift + 3	HELPER		
Ctrl + Shift + 4	STYLE	Create	
Ctrl + Shift + 5	DOCUMENTATION	Create	
Ctrl + Shift + 6	RENDERER	Create	
Ctrl + Shift + 7	DESIGN	Create	
Ctrl + Shift + 8	SVG	Create	
Bundle Version Settings			

Case Milestones

Milestone							
New							
Order	Name	Recurrence Type	Criteria	Minutes To Complete Milestone	Time Trigger Apex Class	Start Time	Business Hours
1	First Response Time	No Recurrence		60		Entitlement Process	Default
2	Resolution Time	No Recurrence		120		Entitlement Process	Default
3	Closure Time	No Recurrence		360		Entitlement Process	Default

[Back To Top](#) Always show me [more records per related list](#)

Case Milestones (3)			
Name	Target Date	Completion Date	Time Remaining (Min:Sec)
First Response Time	8/31/2022, 8:51 AM		59:54
Resolution Time	8/31/2022, 9:51 AM		119:54
Closure Time	8/31/2022, 1:51 PM		359:54
View All			

For this requirement I created triggers using trigger helper to implement this case milestone.

- Once Case is moved from New to In Progress, First Response Time Milestone must be completed.
- Once Case is moved from In Progress to resolved, Resolution Time Milestone must be completed
- Once Case is moved to Closed, All Milestones must be completed.

Created a trigger on case object to update the milestone

```
trigger completerestimetrigger on Case (after update,before insert,before update,before delete,after delete,after undelete)
{
    if(Trigger.isBefore && Trigger.isUpdate)
    {
        for(case cs:Trigger.new){
            if(cs.Status == 'In Progress')
                MilestoneUtils.completeMilestone(cs.Id,'First Response Time');
            else if(cs.Status== 'Resolved')
                MilestoneUtils.completeMilestone(cs.Id,'Resolution Time');
            else if (cs.Status== 'Closed')
            {
                MilestoneUtils.completeMilestone(cs.Id,'First Response Time');
                MilestoneUtils.completeMilestone(cs.Id,'Resolution Time');
            }
        }
    }
}
```

Created an apex class to calculate the milestone completion time

```

public class MilestoneUtils
{
    public static void completeMilestone(Id caseIds,String milestoneName)
    {
        if(Trigger.isExecuting){
            List<CaseMilestone> updateCase = [select Id, completionDate
            from CaseMilestone cm
            where caseId = :caseIds and cm.MilestoneType.Name=:milestoneName
            and completionDate = null limit 1];
            if (updateCase.isEmpty() == false){
                for (CaseMilestone cm : updateCase){
                    cm.completionDate = System.now();
                }
                update updateCase;
            }
        }
    }
}

```

Created an apex class to count the number of cases per account

```

public class CaseTriggerHandler
{
    public static void updateCount( List<Case> listCases, Map<Id,case> mapOldContacts ) {
        Set<Id> setAccountId = new Set<Id>();
        if( mapOldContacts == null || mapOldContacts.isEmpty() ) {
            for( Case objCont : listCases ) {
                if( objCont.AccountId != null ) {
                    setAccountId.add( objCont.AccountId );
                }
            }
        }
        else if( mapOldContacts != null && !mapOldContacts.isEmpty() ) {
            for( Case objCont : listCases ) {
                if( objCont.AccountId != null ) {
                    setAccountId.add( objCont.AccountId );
                }
            }
        }
    }
}

```

Created a trigger on Case object to update the count of total number of Cases per Account

```
if( Trigger.isAfter )
{
    if( Trigger.isInsert )
    {
        CaseTriggerHandler.updateCount( trigger.new, null );
    }
    else if( Trigger.isUpdate)
    {
        CaseTriggerHandler.updateCount( trigger.new, trigger.oldMap );
    }
    else if( Trigger.isUndelete)
    {

```