

# <sup>1</sup>Project Doc: 2

## Superbadges

### Process Automation-Super-badge:

1. Here we have automated leads for the company where according to the validation rules and assignment rules for whom the leads have to be sent have been set
2. Accounts have been automated and a new object named "Robot setup" have been created which checks for the leads and availability according to the validation rules and the flows.
3. A Sales process have been created which checks for the sales and automates the process of sales
4. Opportunities have also been automated which is all monitored by a user "Nushi Davoud" where he is the one who makes the decision.
5. A flow named "Product quick search" have been created which automates products according to different category.
6. A process have been created which sets up the data and describes about the business requirements.
7. Finally all the Automation for taking in leads to changing, creating, sending email alerts and updating records are all have been done.

### Apex Specialist Super-Badge:

1. This is a trigger that automates a Maintenance request when any new record is created or updated.

```
trigger MaintenanceRequest on Case (before update, after update) {  
  if (Trigger.isUpdate && Trigger.isAfter) {  
    MaintenanceRequestHelper.updateWorkOrders(Trigger.New,  
    Trigger.OldMap);  
  }  
}
```

2. This is a Helper class which gets activated by the Request trigger which creates a time slot for the maintenance.

```
public with sharing class MaintenanceRequestHelper {  
  public static void updateWorkOrders(List<Case> updWorkOrders,  
  Map<Id,Case>  
  nonUpdCaseMap) {  
    Set<Id> validIds = new Set<Id>();  
    for (Case c : updWorkOrders) {  
      if (nonUpdCaseMap.get(c.Id).Status != 'Closed' && c.Status ==  
      'Closed') {  
        if (c.Type == 'Repair' || c.Type == 'Routine Maintenance') {  
          validIds.add(c.Id);  
        }  
      }  
    }  
  }  
}
```

//When an existing maintenance request of type Repair or Routine Maintenance is closed,

//create a new maintenance request for a future routine checkup.

```
if (!validIds.isEmpty()) {
```

```
  Map<Id,Case> closedCases = new Map<Id,Case>([SELECT Id,
```

```
Vehicle__c,  
Equipment__c, Equipment__r.Maintenance_Cycle__c,  
(SELECT Id,Equipment__c,Quantity__c FROM  
Equipment_Maintenance_Items__r)  
FROM Case WHERE Id IN :validIds]);  
Map<Id,Decimal> maintenanceCycles = new Map<ID,Decimal>();  
//calculate the maintenance request due dates by using the  
maintenance cycle  
defined on the related equipment records.  
AggregateResult[] results = [SELECT Maintenance_Request__c,  
MIN(Equipment__r.Maintenance_Cycle__c)cycle  
by s.srinath
```