***Technical Design***

***User Story 454637- Automate cleanup of old transaction data to get rid of space issue in dev environment***

**Revision History**

| **Version** | **Date** | **Description** | **Author** |
| --- | --- | --- | --- |
| V0.1 | 4-1-2017 | Creating Technical Design | radhika.sood |
|  |  |  |  |
|  |  |  |  |

**Associated documents**

| **Document** | **Description** |
| --- | --- |
|  |  |
|  |  |

Table of Contents

[Introduction 3](#_Toc504638916)

[Purpose of this document 3](#_Toc504638917)

[Story Phrase 3](#_Toc504638918)

[US Link: 3](#_Toc504638919)

[Prerequisite 3](#_Toc504638920)

[Dependency: N/A 3](#_Toc504638921)

[Impact Analysis 3](#_Toc504638922)

[SFDC Component Changes 3](#_Toc504638923)

[Detail Design 4](#_Toc504638924)

[Overview 4](#_Toc504638925)

[Process Flow (Optional) 5](#_Toc504638926)

[Apply the isSandbox check before any code execution 9](#_Toc504638927)

[Use Database.delete instead of delete 9](#_Toc504638928)

# Introduction

## Purpose of this document

This document tells about impact analysis and flow of design approach for complete user story as per end user expectation. It also includes the details of the components involved in development.

## Story Phrase

This user story includes automation of cleanup of old transaction data to get rid of space issue in dev environment.

# US Link:

https://ts.accenture.com/sites/MMS/Lists/Scope%20Item%20List/Item/displayifs.aspx?List=2e9c2f58-ecf2-4340-967e-766e4887d26d&ID=2420&Web=445cb111-9e0e-4e30-b990-472d6231734d

FMD:

Publish Event:

Provide Data:

## Prerequisite

### Dependency: N/A

Access Privilege: N/A

# Impact Analysis

## SFDC Component Changes

|  |  |  |  |
| --- | --- | --- | --- |
| **Component Type** | **Component Name** | **Type of Change** | **Change Detail** |
| Apex class | Check\_Available \_Space | *New* | This class (scheduled at regular interval) will calculate the threshold value, measure the storage space used by the objects and call the batch class if storage space used by objects is greater than threshold value. |
| Batch Apex Class | Clean\_Old\_Transaction\_Data | *New* | Deletes the old transaction data for certain time range. User can manually configure which objects (not including reference objects) and number of days for which data will be deleted. |
| Apex Class | Check\_Available\_Space\_Test | *New* | Test class will contain test methods and classes to verify whether a particular piece of code is working properly. |
| Apex Class | Clean\_Old\_Transaction\_Data\_Test | *New* | Test class will contain test methods and classes to verify whether a particular piece of code is working properly. |

**Custom Settings**

**Automate\_Cleanup\_Objects\_List**

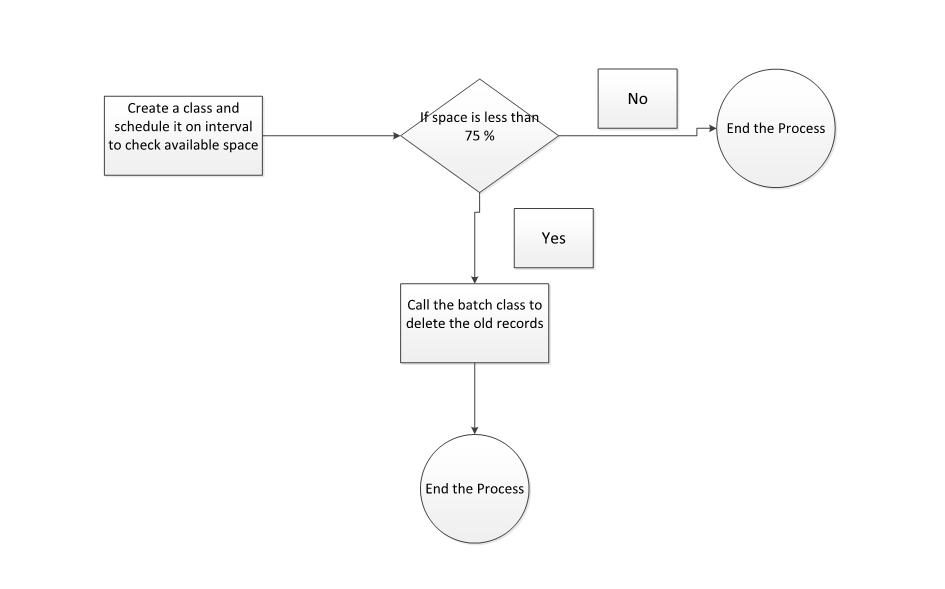
|  |  |  |
| --- | --- | --- |
| **Field Name** | **Field Type** | **Description** |
| Cleanup\_Data\_Number\_of \_Days | Number(18,0) | Contains information for how many days the data will be deleted |
| Object\_Name | Text(120) | Contains list of objects (not including reference objects) whose records will be deleted. |
| Email\_Sent\_to\_Environment\_Team | Email | Contains the email address of the concerned team that will receive the mail, if any error occurs. |
| SenderEmail | Email | The email address from which recipient of the mail will receive the mail. |
| TimeInterval | Text Area(255) | Contains the time when job will be scheduled. |
| BatchSize | Number(18,0) | Contains the size of batch information. |

# Detail Design

## Overview

The page display appropriate error message at top of the page.

## Process Flow (Optional)



***Pseudo code:***

***Check\_Available\_Space***

*public class Check\_Available\_Space implements Schedulable{*

*List<Automation\_details\_\_c> autoDetail;*

*List<Automation\_details\_\_c> autoObjClean;*

*global string query;*

*global void execute(SchedulableContext sc)*

*{*

*//public static String CRON\_EXP = '0 0 0 3 9 ? 2018'; ()*

*autoDetail = Automation\_details\_\_c.getall().values(); // custom setting details*

*autoObjClean = Automate\_cleanup\_objects\_list\_\_c.getall().values();*

*String query = ‘SELECT IsSandbox FROM Organization';*

*return Database.getQueryLocator(query);*

*if (query == ‘true’)*

*{*

*String sid = UserInfo.getSessionId();*

*HttpRequest req = new HttpRequest();*

*req.setMethod('GET');*

*req.setEndpoint('https://accenture--opsdev.cs50.my.salesforce.com'); //Lable.URL*

*req.setHeader('Authorization', 'OAuth '+ sid);*

*Http http = new Http();*

*HTTPResponse res = http.send(req);*

*Map<String, Object> m = (Map<String,Object>)JSON.deserializeUntyped(res.getBody());*

*Map<String, Object> dataStorage = (Map<String,Object>)m.get('DataStorageMB');*

*int max= Integer.valueOf(dataStorage.get('Max'));*

*integer remaining = Integer.valueOf(dataStorage.get('Remaining'));*

*int diff= (max-remaining);*

*int threshold= (max \* 0.75);*

*for(Automation\_details\_\_c d : autoDetail)*

*{*

*if (d.Cleanup\_Data\_Number\_of\_days\_\_c = null && autoObjClean = null)// do nothing*

*if ( query == ‘true’ && diff > threshold && d.Cleanup\_Data\_Number\_of\_days\_\_c>= 0 && autoObjClean != null)*

*Clean\_Old\_Transaction\_Data cot = new Clean\_Old\_Transaction\_Data() ;*

*ID batchprocessid = Database.executeBatch(cot);*

*}*

*}*

*}*

*}*

***Clean\_Old\_Transaction\_Data***

*global class Clean\_Old\_Transaction\_Data implements Database.Batchable<sObject>*

*{*

*global Database.QueryLocator start(Database.BatchableContext BC)*

*{*

*String query = 'SELECT \* FROM Automate\_ Object\_List\_\_c ';*

*return Database.getQueryLocator(query);*

*decimal qfrom = Automation\_details\_\_c.Cleanup\_Data\_Number\_of\_days\_\_c \* -1;*

*string dd = system.now().addDays(qfrom.intValue()).format('yyyy-MM- dd\'T\'HH:mm:ss.SSS\'Z\'');*

*string qobject = Automate\_cleanup\_objects\_list\_\_c;*

*query1 = 'select \* from ' + qobject + ' where lastmodifieddate<= ' + dd+'';*

*}*

*global void execute(Database.BatchableContext BC, List<sObject> scope){*

*try{*

*database.delete();*

*DataBase.emptyRecycleBin(scope);*

*}*

*catch(Exception exp)*

*{*

*UTIL\_LoggingService.logHandledException(exp, UserInfo.getOrganizationId(), UTIL\_Constants.APPLICATION\_NAME\_MMS, UTIL\_Constants.CLASS\_MassDeleteObjects, UTIL\_Constants.METHOD\_execute, null, LoggingLevel.ERROR);*

*}*

*}*

*global void finish(Database.BatchableContext BC){*

*if(NumberOfErrors>0)*

*{*

*Messaging.SingleEmailMessage mail = new Messaging.SingleEmailMessage();*

*String[] toAddresses = new String[] {email1, email2};*

*mail.setToAddresses(toAddresses);*

*mail.setSubject('Batch Subject ' + a.Status);*

*mail.setPlainTextBody('records processed ' + a.TotalJobItems +*

*'with '+ a.NumberOfErrors + ' failures.');*

*Messaging.sendEmail(new Messaging.SingleEmailMessage[] { mail });*

*}*

*}*

|  |  |  |  |
| --- | --- | --- | --- |
| **Reviewed By:** | Vijaykumar.Singh | **Reviewed On:** | 11 January, 2018 |
| **Reviewed By:** |  | **Reviewed On:** |  |

|  |  |
| --- | --- |
| **Review Comments/Observations**  (To be filled by the Reviewer) | **Status**  (To be filled by the Dev POD Lead) |
| Make the batch size and time frequency  configurable |  |
| Apply the isSandbox check before any code execution |  |
| Use Database.delete instead of delete |  |
| Handle the Partial delete scenario |  |
|  |  |
|  |  |
|  |  |