



# **MANGAYARKARASI COLLEGE OF ENGINEERING**

## **Department Of Computer Science Engineering**

**Completed a project on**

**CALCULATING FAMILY EXPENSES USING SERVICE NOW**

**Submitted By**

**Team ID: NM2025TMID02223**

**Team Leader: Devadharshini A**

**Team Member: Abi Kannaamma S**

**Team Member: Baga Priya S**

**Team Member: Kiruthika S**

# Calculating Family Expenses Using Service Now

## Problem Statement

In a project of managing family expenses is often challenging due to the lack of a centralization system for tracking income, expenses, and savings. Families typically rely on manual methods such as spreadsheet, paper records, or multiple apps, which can lead to errors, duplication, and difficulty in analyzing financial patterns.

## Objective

**Simplify Expense Tracking-** provide a centralization system for recording and categorizing all family expenses

**Improve Financial Awareness-**offer real-time dashboards and reports to give clear insights into spending patterns

**Enhance Budget Planning-**support better financial planning by identify high-spending categories and opportunities to save

**Skills:** Service now Development, Workflow & Automation, Scripting in Service now, Reporting& Dashboards, Integration Skills

## TASK INITIATION

### Update

#### Activity 1:

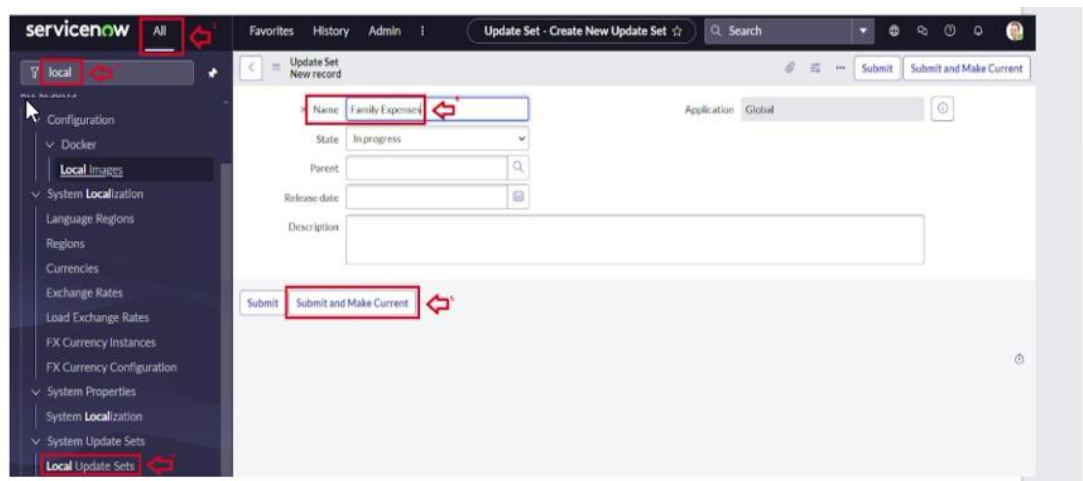
1. Open Service now
2. Go to All>>search for Local Update Set>Click on New



3. Enter the Details as:

**Name:** Family Expenses

4. And then click on Submit and make current



## Table

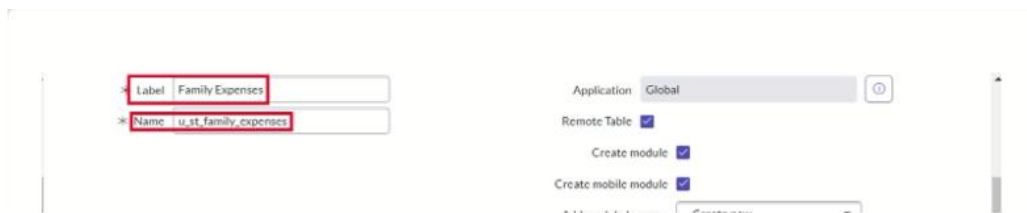
### Activity 2:

1. Open Service now
2. Go to all>search for Tables>Click on New
3. Enter the Details:

**Label:** Family Expenses

**Name:** Auto-Populated

**New menu name:** Family Expenditure



4. Click on save

### Activity 2:

1. Open Service now
2. Near Columns Double Click near insert a new row

Give the details as:

**Column label:** Number

**Type:** String

3. Double click on insert a new row again

Give the details as:

**Column label:** Date

**Type:** Date

4. Double click on insert a new row again

Give the details as:

**Column label:** Amount

**Type:** Integer

5. Double click on insert a new row again

Give the details as:

**Column label:** Expense Details

**Type:** String

**Max length:** 800



6. Click on save

## Activity 2:

1. Open Service now

2. Double click on the Number field/Column

3. Go down and double click on advanced view

4. In Default Value:

**Use dynamic default:** check the box

**Dynamic default value:** Get Next Padded Number

5. Click on update

6. Go to all>>search for Number Maintenance>>select number maintenance
7. Click on new
8. Enter the details:

**Table:** Family Expenses

**Prefix:** MFE

8. Click on Submit

## Activity 2:

1. Open Service now
2. Go to all>>search for Family Expenses>>Open family Expenses
3. Click on new
4. Go to header and click there>>click on configure>>Select Form Design
5. Drag Drop the form as requirement

6. Make Number Read-Only Field by Clicking
7. Make Date, Amount Mandatory field by Clicking
8. Click on save

## Table (Daily Expenses)

### Activity 3

1. Open Service now
2. Go to all>search for Tables>Click on New
3. Enter the Details:

**Label:** Daily Expenses

**Name:** Auto-Populated

**Add Module to menu:** Family Expenditure

The screenshot shows the 'New Table' form in ServiceNow. The 'Label' field is set to 'Daily Expenses' and the 'Name' field is set to 'u\_daily\_expenses'. The 'Extends table' field is empty. On the right, the 'Application' is set to 'Global'. The 'Create module' checkbox is checked. The 'Create mobile module' checkbox is checked. The 'Add module to menu' dropdown is set to 'Family Expenditure'. The 'Application Menu' button is visible.

4. Click on save

### Activity 3:

1. Open Service now
2. Near Columns Double Click near insert a new row

Give the details as:

**Column label:** Number

**Type:** String

3. Double click on insert a new row again

Give the details as:

**Column label:** Date

**Type:** Date

4. Double click on insert a new row again

**Column label:** Amount

**Type:** Integer

5. Double click on insert a new row again

**Column label:** Expense Details

**Type:** String

**Max length:** 800

6. Double click on insert a new row again

Give the details as:

**Column label:** Comments

**Type:** String

**Max length:** 800

6. Click on save

### Activity 3:

1. Open Service now

2. Double click on the Number field/Column

3. Go down and double click on advanced view

4. In Default Value:

**Use dynamic default:** check the box

**Dynamic default value:** Get Next Padded Number

5. Click on update

Choice List Specification Calculated Value **Default Value**

The Default value specifies what value the field has when first displayed.

☒ Use dynamic default

Dynamic default value: Get Next Padded Number

6. Go to all>>search for Number Maintenance>>select number maintenance

7. Click on new

8. Enter the details:

**Table:** Daily Expenses

**Prefix:** MFE

Number  
New record

Table: Daily Expenses

Prefix: MFE

Number: 1,000

Application: Global

Number of digits: 7

Submit

9. Click on submit

### Activity 3:

1. Open Service now
2. Go to all>>search for Daily Expenses>>Open Daily Expenses
3. Click on new
4. Go to header and click there>>click on configure>>Select Form Design
5. Drag Drop the form as requirement



6. Make Number Read-Only Field by Clicking
7. Make Date, Family Member Name Mandatory field by Clicking
8. Click on save

## Relationship

### Activity 4:

1. Open Service now
2. Go to all>>search for relationships>>open relationships
3. Click on new
4. Enter the details:

**Name:** Daily Expenses

**Applies to table:** select Family Expenses

**Daily Expenses:** select Daily Expenses

5. Click save

## Configuring Related List

### Activity 5:

1. Open Service now
2. Go to all>>search for Family Expenses>>Open Family Expenses
3. Click on new



4. Go to header and click there>>click on configure>>select Related Lists
5. Add Daily Expenses to selected area
6. Click on save



## Business Rule

### Activity 6:

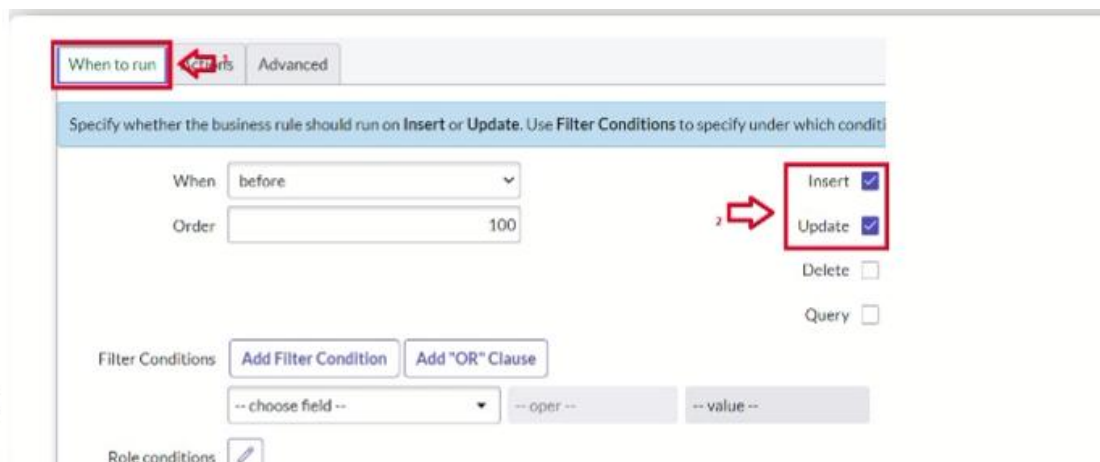
1. Open Service Now
2. Go to all>>search for Business Rules
3. Under system definition select business rules then click new
4. Enter the Details:

**Name:** Family Expenses BR

**Table:** Select Daily Expenses

Check Advanced

5. In when the run check insert and update



6. In Advance (we write the code)

```

1 (function executeRule(current, previous /*null when async*/) {
2
3     var FamilyExpenses = new GlideRecord('u_family_expenses');
4     FamilyExpenses.addQuery('u_date', current.u_date);
5     FamilyExpenses.query();
6     if(FamilyExpenses.next())
7     {
8         FamilyExpenses.u_amount += current.u_expense;
9         FamilyExpenses.u_expense_details += ">" + current.u_comments + ":" + "Rs." + current.u_expense + "/-";
10        FamilyExpenses.update();
11    }
12    else
13    {
14        var NewFamilyExpenses = new GlideRecord('u_family_expenses');
15        NewFamilyExpenses.u_date = current.u_date;
16        NewFamilyExpenses.u_amount = current.u_expense;
17        NewFamilyExpenses.u_expense_details += ">" + current.u_comments + ":" + "Rs." + current.u_expense + "/-";
18        NewFamilyExpenses.insert();
19    }
20
21 }) (current, previous);

```

7. Click on save

## Configure the Relationship

### Activity 7:

1. Open Service now
2. Go to all>>search for relationships>>open relationships
3. In open Daily Expenses relationship

**For Applies to table:** select Family Expenses

**In Query with:** write the below Query

4. Click on update

Relationship Daily Expenses

Name: Daily Expenses

Application: Global

Advanced: ☐

Applies to table: Family Expenses [u\_family\_expenses]

Queries from table: Daily Expenses [u\_daily\_expenses]

This script refines the query in current that will populate the related list. For more information about it, its parameters and control variables, see the [documentation](#). See also the article about the recommended form of the script.

Query with:

```

1 (function refineQuery(current, parent) {
2
3     // Add your code here, such as current.addQuery(field, value);
4     current.addQuery('u_date', parent.u_date);
5     current.query();
6
7 }) (current, parent);

```

Update Delete