

Project Documentation : Calculating Family Expenses Using Servicenow

1.Introduction

1.1 Project Overview

The project aims to develop a comprehensive expense calculation system using ServiceNow. This system will enable users to track and manage family expenses efficiently. It will include features such as expense categorization, budget setting, real-time tracking, and reporting capabilities. Utilizing ServiceNow's robust platform, the project will ensure seamless integration, user-friendly interface, and scalability to accommodate varying family sizes and financial complexities. The end goal is to empower users with the tools they need to make informed financial decisions and promote financial well-being within the family unit.

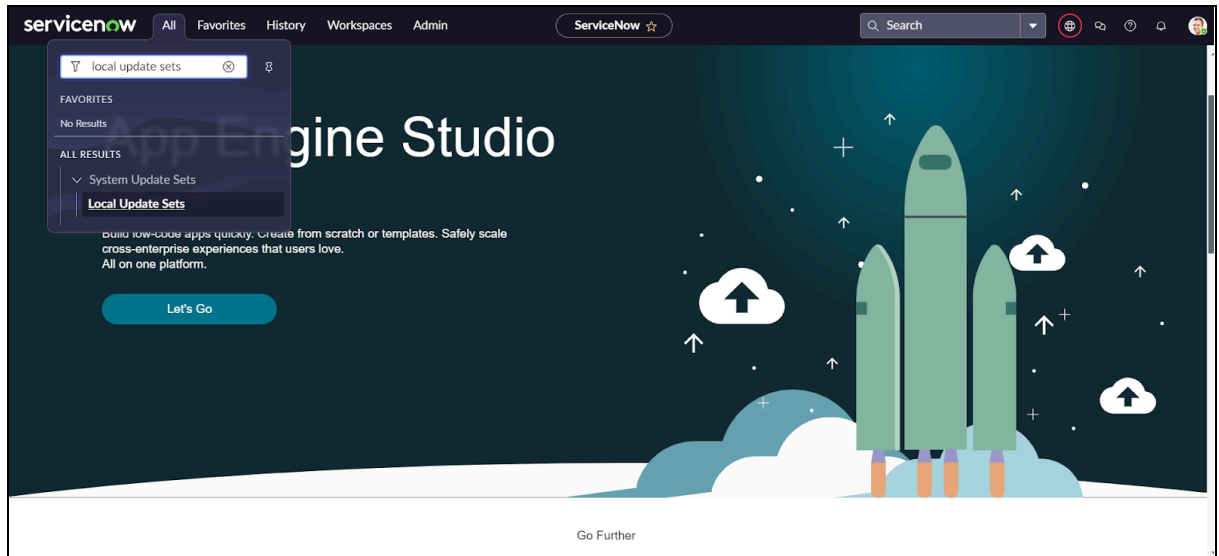
1.2 Purpose

To develop a user-friendly Servicenow Application that allows families to record expenses,categorize them for better financial Management

2.Process

2.1 Creation of update set

1. Go to All >> In the filter search for Local Update set > click on New.



2. Enter the Details as:
Name : Family Expenses
3. Then click on Submit and Make current.
Name : Family Expenses
3. Then click on Submit and Make current.

2.2 Creation of Family Expenses Table

1. Go to All > In the filter search for Tables > click on New.
2. Enter the Details:
Label : Family Expenses
Name : Auto-Populated
New menu name : Family Expenditure

* Label
 * Name

Application ⓘ
 Remote Table ☒
 Create module ☒
 Create mobile module ☒
 Add module to menu
 New menu name

Columns Controls Application Access

Table Columns ⓘ

Dictionary Entries

	Column label ▼	Type	Reference	Max length	Default value	Display
✕ ✎	Number	String				false
✕ ✎	Date	Date				false
✕ ✎	Amount	Integer				false

3. Go to the Header and right click there>> click on Save.

2.2.1 Creation of columns(Fields)

9. Near Columns Double click near insert a new row.

10. Give the details as:

Column label : Number

Type : String

3. Double click on insert a new row again

4. Give the details as:

Column label : Date

Type : Date

5. Double click on insert a new row again

6. Give the details as:

Column label : Amount

Type : Integer

7. Double click on insert a new row again

8. Give the details as:

Column label : Expense Details

Type : String

Max length : 800

Columns						
Table Columns for text Search						
Dictionary Entries	Column label	Type	Reference	Max length	Default value	Display
	Number	String				false
	Date	Date				false
	Amount	Integer				false
	Expense Details	String		800		false
Insert a new row...						

- Go to the Header and right click there>> click on Save.

2.2.2 Making Number field an Auto-Number

- Double click on the Number Field/Column.
- Go down and double click on Advanced view
- In Default Value:
Use dynamic default : check the box
Dynamic default value : Get Next Padded Number
- 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

Delete Column **Update**

-
- Go to All >> In the filter search for Number Maintenance >> select Number Maintenance
- Click on New.
- Enter the below Details:
Table : Family Expenses
Prefix : MFE

Number MFE

Table: Family Expenses

Prefix: MFE

* Number: 1,000

Application: Global

Number of digits: 7

Update Delete

9. Click on Submit.

2.2.3 Configure the form

1. Go to All >> In the filter search for Family Expenses >> Open Family Expenses
2. Click on New
3. Go to the Header and right click there>> click on Configure >> Select Form Design
4. Customize or Drag Drop the form as per your requirement.

Family Expenses [u_family_expenses] 2 Column

Number Date Amount

Expense Details 1 Column

5. Make Number Read-Only Field by clicking on the gear icon and checking Read-Only
6. Make Date, Amount Mandatory Field by clicking on the gear icon and checking Mandatory
7. Click on Save.

2.3 Creation of Daily expense Table

1. Go to All > In the filter search for Tables > click on New.
2. Enter the Details:
Label : Daily Expenses
Name : Auto-Populated

Add Module to menu : Family Expenditure

The screenshot shows the 'Add Module to menu' configuration interface. On the left, there are three input fields: 'Label' with the value 'Daily Expenses', 'Name' with the value 'u_daily_expenses', and 'Extends table' which is empty. On the right, there are two checked checkboxes: 'Create module' and 'Create mobile module'. Below these is a dropdown menu labeled 'Add module to menu' with 'Family Expenditure' selected. A red box highlights the 'Add module to menu' dropdown, and a red arrow points to it with the number 3. Another red box highlights the 'Label' field, and a red arrow points to it with the number 1. A third red box highlights the 'Name' field, and a red arrow points to it with the number 2.

3. Go to the Header and right click there>> click on Save.

2.3.1 Creation of column or fields

1. Near Columns Double click near insert a new row.
2. Give the details as:
Column label : Number
Type : String
3. Double click on insert a new row again
4. Give the details as:
Column label : Date
Type : Date
5. Double click on insert a new row again
6. Give the details as:
Column label : Expense
Type : Integer
7. Double click on insert a new row again
8. Give the details as:
Column label : Family Member Name
Type : Reference
Max length : 800
9. Double click on insert a new row again
10. Give the details as:

Column label : Comments

Type : String

Max length : 800

11. Go to the Header and right click there>> click on Save.

2.3.2 Making Number an auto-Number

1. Double click on the Number Field/Column.
2. Go down and double click on Advanced view
3. In Default Value:
Use dynamic default : check the box
Dynamic default value : Get Next Padded Number
4. 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

Delete Column | **Update**

- 5.
6. Go to All >> In the filter search for Number Maintenance >> select Number Maintenance
7. Click on New.
8. Enter the below Details:
Table : Family Expenses
Prefix : MFE

Number New record

* Table **Daily Expenses** 1

Prefix **DFE** 2

* Number 1,000

Application Global

Number of digits 7

Submit 3

9. Click on Submit.

2.3.3 Configure the form

1. Go to All >> In the filter search for Daily Expenses >> Open Daily Expenses
2. Click on New
3. Go to the Header and right click there>> click on Configure >> Select Form Design
4. Customize or Drag Drop the form as per your requirement.

Daily Expenses [u_daily_expenses] 2 Column

Number ## Family Member Name

Date ## Expense

1 Column

Comments

5. Make Number Read-Only Field by clicking on the gear icon and checking Read-Only
6. Make Date, Family Member Name Mandatory Field by clicking on the gear icon and checking Mandatory
7. Click on Save.

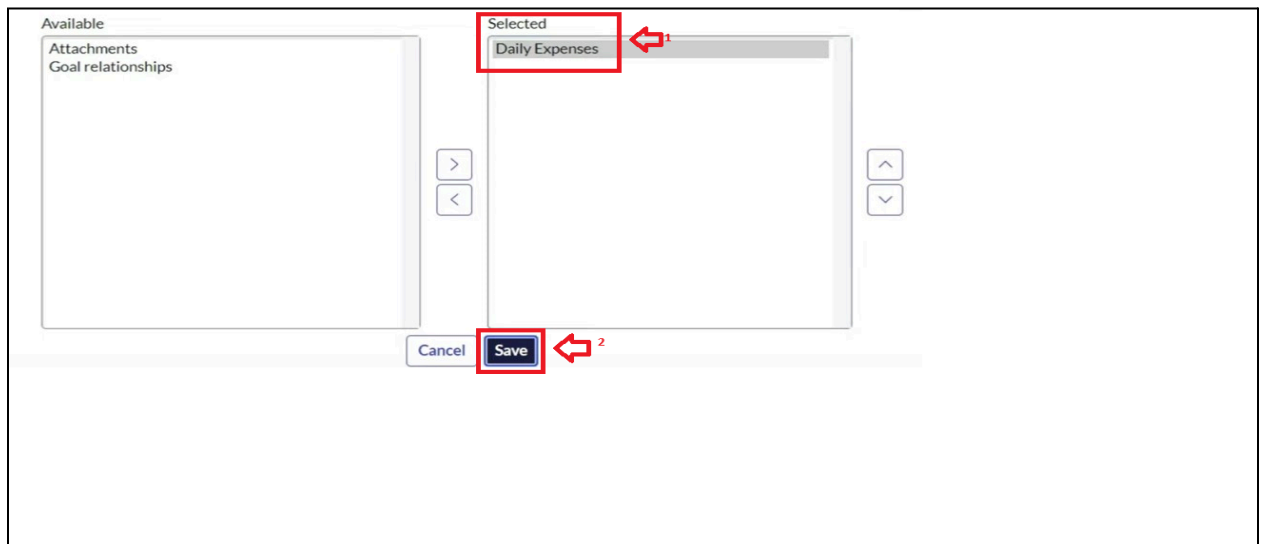
2.4 Creation of relationship between daily expense table and family expense table

1. Go to All >> In the filter search for Relationships >> Open Relationships
2. Click on New.

3. Enter the details:
Name : Daily Expenses
Applies to table : Select Family Expenses
Daily Expenses : Select Daily Expenses
4. Click Save.

2.5 Configuring related list on family expense

1. Go to All >> In the filter search for Family Expenses >> Open Family Expenses
2. Click on New
3. Go to the Header and right click there>> click on Configure >> Select Related Lists
4. Add Daily Expenses to the Selected Area.
5. Click on Save



2.6 Creation of Business Rules

1. Go to All >> In the filter search for Business Rules.
2. Under System Definition Select Business Rules then click on New.
3. Enter the Details:
Name : Family Expenses BR
Table : Select Daily Expenses
Check Advanced

Business Rule
New record

Business rule is a server-side script that runs when a record is displayed, inserted, deleted, or when a table is queried. Use business rules to automatically change values in form fields when the specified conditions are met.

Name: Family Expenses BR

Table: Daily Expenses (u_daily_expenses)

Application: Global

Active: ☒

Advanced: ☒

4. In when to run Check Insert and Update

When to run

Specify whether the business rule should run on **Insert** or **Update**. Use **Filter Conditions** to specify under which conditions.

When: before

Order: 100

Insert: ☒

Update: ☒

Delete: ☐

Query: ☐

Filter Conditions: Add Filter Condition Add "OR" Clause

-- choose field -- -- oper -- -- value --

Role conditions:

5. In Advance (we write the code):

```
(function executeRule(current, previous /*null when async*/) {

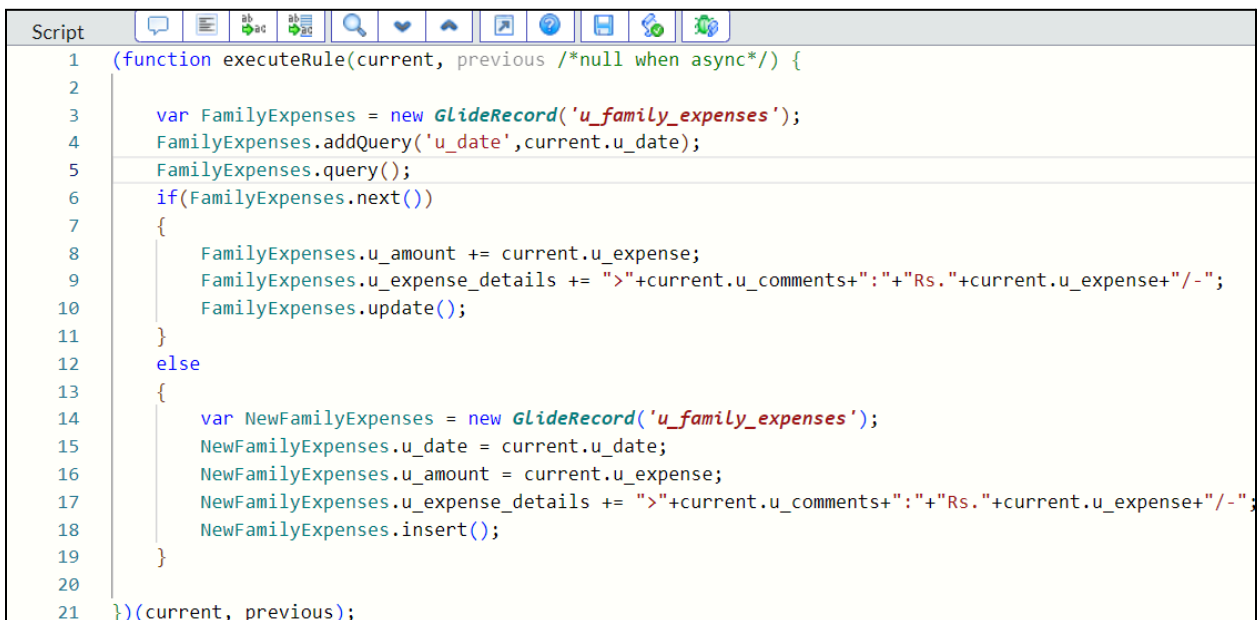
var FamilyExpenses = new GlideRecord('u_family_expenses');
FamilyExpenses.addQuery('u_date',current.u_date);
FamilyExpenses.query();
if(FamilyExpenses.next())
{
FamilyExpenses.u_amount += current.u_expense;
FamilyExpenses.u_expense_details +=
">" + current.u_comments + ":" + "Rs." + current.u_expense + "/-";
}
```

```

FamilyExpenses.update();
}
else
{
var NewFamilyExpenses = new GlideRecord('u_family_expenses');
NewFamilyExpenses.u_date = current.u_date;
NewFamilyExpenses.u_amount = current.u_expense;
NewFamilyExpenses.u_expense_details +=
">" + current.u_comments + ":" + "Rs." + current.u_expense + "/-";
NewFamilyExpenses.insert();
}

})(current, previous);

```



The screenshot shows a script editor window with a toolbar at the top. The script is a JavaScript function named `executeRule` that takes `current` and `previous` as arguments. It uses a `GlideRecord` object to interact with the `u_family_expenses` table. The logic is as follows:

```

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);

```

6. Go to the Header and right click there>> click on Save.

2.7 Configure the Relationship

1. Go to All >> In the filter search for Relationships >> Open Relationships.
2. In that, open Daily Expenses Relationship.
3. For Applies to table : Select Family Expenses.

4. In Query with : write the below Query.

```
(function refineQuery(current, parent) {  
  
  // Add your code here, such as current.addQuery(field, value);  
  current.addQuery('u_date',parent.u_date);  
  current.query();  
  
})(current, parent);
```

5. Click on Update.

Relationship
Daily Expenses

Name:

Application:

Advanced: ☐

Applies to table: ¹

Queries from table:

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);
```

²

³

3.Result

Insert the records in the daily expense table.

Number	Comments	Date	Expense	Family Member Name
DFE0001006	Shopping	2025-09-05	1,000	Meghana
DFE0001005	Parlour	2025-09-05	1,000	Rani
DFE0001007	Market	2025-09-06	300	Raju
DFE0001008	Petrol	2025-09-12	200	Sai

The result in the family expense table is

Number	Amount	Date	Expense Details
MFE0001003	2,000	2025-09-05	>Parlour:Rs.1000/->Shopping:Rs.1000/-
MFE0001004	300	2025-09-06	>Market:Rs.300/-
MFE0001005	200	2025-09-12	>Petrol:Rs.200/-

4. Conclusion

In conclusion, implementing the "Calculation of Family Expenses Using ServiceNow" project offers numerous benefits to the household. By leveraging the robust capabilities of ServiceNow, family members can efficiently submit, track, and manage their expenses in a centralized and streamlined manner.

5.References

Project demo video Link : https://youtu.be/umEqJQ_07xA?si=bH8xKRz70KwhCnAk

Project Github Link : <https://github.com/yvswaraag/Family-Expenses-Servicenow>