

Calculating Family Expenses using ServiceNow

Project Title : Calculating family expenses using service now

Team ID:NM2025TMID15609

Team Size: 4

Team Leader: Jyothika Kottu

Team Member: Lavanya Tuthika

Team Member: Kamadi Sowmya Sri

Team Member: Kadimi Santhosh Vardhan

Problem Statement:

There is currently no efficient system in place to track, categorize, and analyze family expenses in a way that is integrated, automated, and accessible in real-time. This leads to poor budget planning, missed payments, overspending, and a lack of financial visibility.

Objective:

To design and implement a solution using **ServiceNow** that allows family members to:

1. **Input and categorize expenses** easily via a user-friendly interface or mobile portal.
2. **Automate recurring expenses** (e.g., rent, subscriptions).
3. **Track total and category-wise spending** on a monthly/weekly basis.
4. **Set spending limits and receive alerts** when thresholds are crossed.
5. **Generate reports and dashboards** to visualize trends and support better financial decisions.
6. **Share visibility and responsibilities** across family members via roles or user groups.

Skills:

- **ServiceNow App Development** (App Engine Studio)
- **Custom Tables & Data Modeling**
- **Form Design & UI Customization**
- **Flow Designer** (for automation)
- **Business Rules & Scripting**
- **Reporting & Dashboards**
- **User Roles & Access Control (ACLs)**

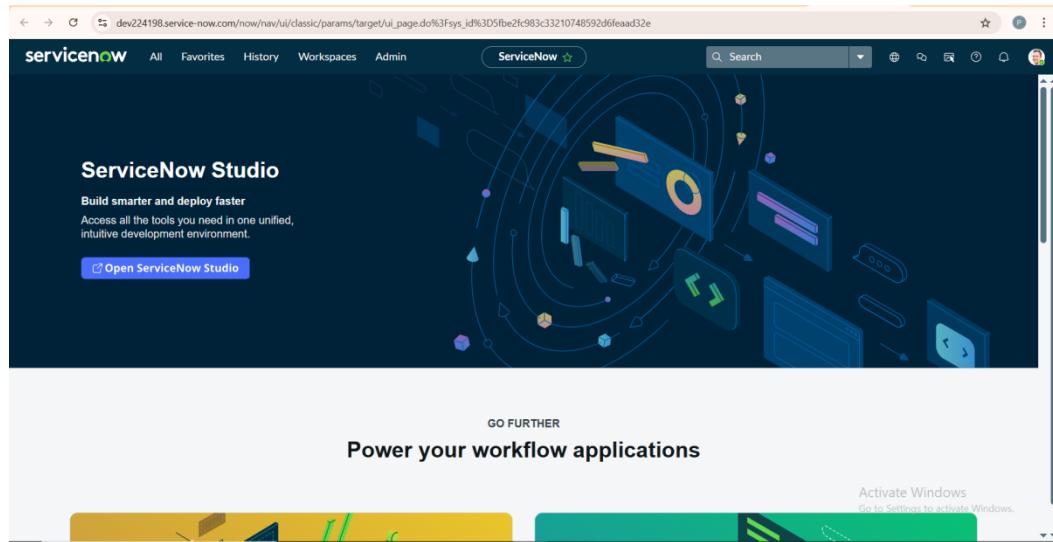
TASK INITIATION

Milestone 1 : Instance

Activity 1: Setting up ServiceNow instance

1. Sign up for a developer account on the ServiceNow Developer site "<https://developer.servicenow.com>".
2. Once logged in, navigate to the "Personal Developer Instance" section.

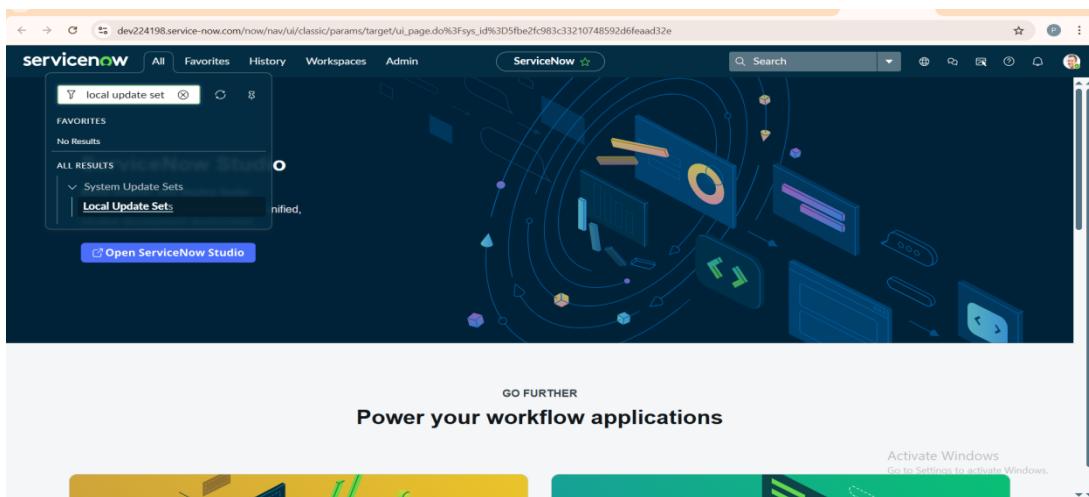
3. Click on "Request Instance" to create a new ServiceNow instance.
4. Fill out the required information and submit the request.
5. You'll receive an email with the instance details once it's ready.
6. Log in to your ServiceNow instance using the provided credentials.
7. Now you will navigate to the ServiceNow.



Milestone 2: New Update Set

Activity 1: Creation of New 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.



The screenshot shows the ServiceNow interface for creating a new 'Update Set'. The top navigation bar includes links for All, Favorites, History, Workspaces, Admin, and a 'Update Sets' button. A search bar is at the top right. The main form has fields for Name (set to 'New Update Set'), State (set to 'In progress'), Parent (empty), Release date (empty), and Description (empty). Buttons for 'Submit' and 'Submit and Make Current' are at the bottom left. A note at the bottom right says 'Activate Windows Go to Settings to activate Windows.'

Milestone 3: Table(Family Expenses)

Activity 1: 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
3. Go to the Header and right click there>> click on Save.

The screenshot shows the ServiceNow interface for creating a new 'Table'. The top navigation bar includes links for All, Favorites, History, Admin, and a 'Table - New Record' button. A search bar is at the top right. The main form has fields for Label (set to 'Family Expenses'), Name (set to 'u_family_expenses'), and Extends table (empty). Under the 'Create module' section, 'Create mobile module' is checked. The 'Add module to menu' dropdown is set to '- Create new --'. The 'New menu name' field contains 'Family Expenditure'. A note at the top says 'ServiceNow recommends creating custom tables in scoped applications. To learn more about creating scoped applications, click [here](#).'. A message at the bottom says 'A table is a collection of records in the database. Each record corresponds to a row in a table, and each field on a record corresponds to a column on that table. Applications use tables and records to manage data and processes.' with a 'More Info' link.

Activity 2: Creation of Family Expenses Table

1. Near Columns Double click near insert a new row.
2. Give the details as:

The screenshot shows the 'Dictionary Entry Number View: Advanced' interface. At the top, there are buttons for 'Delete Column' and 'Update'. Below the header, a message says 'Alters the behavior of a field or functionality that depends on the field. [More Info](#)'. A large text input field is labeled 'Attributes'. Below it, tabs for 'Choice List Specification', 'Calculated Value', and 'Default Value' are visible, with 'Default Value' being the active tab. A message states 'The Default value specifies what value the field has when first displayed.' Underneath, there's a checkbox 'Use dynamic default' which is checked, and a dropdown menu 'Dynamic default value' set to 'Get Next Padded Number'. Buttons for 'Lookup using list', search, and refresh are also present. At the bottom, there are 'Delete Column' and 'Update' buttons.

- 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

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

The screenshot shows the 'Table - Family Expenses' interface. At the top, there are buttons for 'All', 'Favorites', 'History', 'Search', 'Delete', 'Update', and 'Delete All Records'. Below the header, a table titled 'Table Columns' is shown with a search bar. The table has columns for 'Dictionary Entries', 'Column label', 'Type', 'Reference', 'Max length', 'Default value', and 'Display'. The 'Default value' column for the 'Number' row contains the value '800'. The 'Display' column for all rows is set to 'false'.

Dictionary Entries	Column label	Type	Reference	Max length	Default value	Display
	Updated	Date/Time	(empty)	40	false	
	Created by	String	(empty)	40	false	
	Sys ID	Sys ID (GUID)	(empty)	32	false	
	Created	Date/Time	(empty)	40	false	
	Updated by	String	(empty)	40	false	
	Updates	Integer	(empty)	40	false	
x	Number	String			false	
x	Date	Date			false	
x	Amount	Integer			false	
x	Expense Details	String	800		false	

Activity 3: Making Number Field 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.
5. Go to All >> In the filter search for Number Maintenance >> select Number Maintenance
6. Click on New.
7. Enter the below Details:
 Table : Family Expenses
 Prefix : MFE
8. Click on Submit

The screenshot shows a software application window titled "Number New record". At the top, there are standard toolbar icons. Below the title, the form fields are displayed:

- * Table: Family Expenses
- Prefix: MFE
- * Number: 1.000
- Application: Global
- Number of digits: 7

At the bottom left of the form area is a blue "Submit" button. Below the form, under "Related Links", are two items: "Show Counter" and a small gear icon.

Activity 4: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.
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.

Milestone 4: Table(Daily Expenses)

Activity 1: Creation of Table(Daily Expenses)

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

Activity 2: Creation of Columns(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

The screenshot shows the ServiceNow Developers interface with the 'Table - Daily Expenses' tab selected. The 'Columns' tab is active, displaying a list of columns with their properties. The columns listed are:

Column label	Type	Reference	Max length	Default value
Updated by	String	(empty)	40	
Updates	Integer	(empty)	40	
Updated	Date/Time	(empty)	40	
Created by	String	(empty)	40	
Created	Date/Time	(empty)	40	
Sys ID	Sys ID (GUID)	(empty)	32	
Number	String			
Date	Date			
Expense	Integer			
Family Member Name	Reference		800	
Comments	String		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.

Activity 3: Making Number Field an Auto-Number

1. Double click on the Number Field/Column.

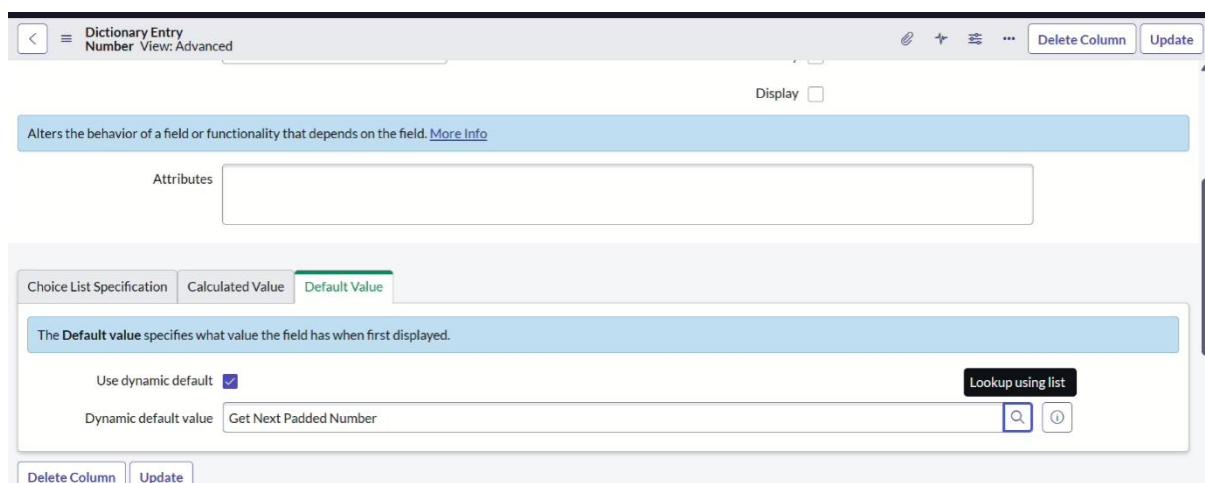
2. Go down and double click on Advanced view

3. In Default Value:

Use dynamic default : check the box

Dynamic default value : Get Next Padded Number

4. Click on Update.



5. Go to All >>In the filter search for Number Maintenance >> select Number Maintenance

6. Click on New.

7. Enter the below Details:

Table : Daily Expenses

Prefix : DFE

8. Click on Submit.

The screenshot shows the ServiceNow interface for creating a new record. The title bar says "Number - New Record". The form fields are as follows:

- Table: Daily Expenses
- Prefix: DFE
- * Number: 1.000
- Application: Global
- Number of digits: 7

At the bottom left is a "Submit" button with a hand cursor icon, and at the bottom right is a "Cancel" button.

The screenshot shows the ServiceNow Form Design interface for the "Daily Expenses [u_daily_expenses]" table. The main area displays the following fields in a 2-column layout:

Number	Family Member Name
Date	Expense

Below this is a single-column section containing a "Comments" field. On the left sidebar, under the "Fields" tab, the following fields are listed: Created, Created by, Updated, Updated by, and Updates. Under the "Formatters" tab, there are sections for Activities (filtered), Contextual Search Results, and Ratings.

Activity 4: 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.
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.

Milestone 5: Creation of Relationship

Activity 1: Creation of Relationship between Family Expenses and Daily Expenses tables

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.

The screenshot shows the ServiceNow interface for creating a new relationship record. The top navigation bar includes links for All, Favorites, History, and a search bar. The main title is "Relationship - New Record". On the left, there's a sidebar with a back arrow and the text "Relationship New record". The main form has fields for "Name" (set to "Daily Expenses"), "Application" (set to "Global"), and checkboxes for "Advanced" and "Scripted". Below these are dropdowns for "Applies to table" (set to "Family Expenses [u_family_expe...]" with a dropdown arrow) and "Queries from table" (set to "Daily Expenses [u_daily_expens...]" with a dropdown arrow). A blue callout box at the bottom left provides documentation on refining queries. At the bottom, there's a "Query with" section containing a radio button for "Turn on ECMAScript 2021 (ES12) mode" and a code editor window with the following script:

```
1 (function refineQuery(current, parent) {  
2     // Add your code here, such as current.addQuery(field, value);  
3 })(current, parent);
```

Milestone 6: Configuring Related List on Family Expenses

Activity 1: Configuring Related List on Family Expenses

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

The screenshot shows a ServiceNow configuration dialog for setting related lists. On the left, under 'Available', there is a list box containing 'Attachments'. On the right, under 'Selected', there is a list box containing 'Daily Expenses'. Between the two boxes are four navigation buttons: a right-pointing arrow above a left-pointing arrow, and an up-pointing arrow above a down-pointing arrow. At the bottom of the dialog are 'Cancel' and 'Save' buttons.

Milestone 7: Business Rules

Activity 1: 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

The screenshot shows the ServiceNow interface for creating a new business rule. The top navigation bar includes links for 'All', 'Favorites', 'History', and a search bar. The main title is 'Business Rule - New Record'. Below the title, there's a note about what a business rule is and a link to 'More Info'. The form fields are as follows:

Name	Family Expenses BR	Application	Global
Table	Daily Expenses [u_daily_expens...]	Active	<input checked="" type="checkbox"/>
Advanced <input checked="" type="checkbox"/>			

At the bottom of the form, there are three tabs: 'When to run' (which is selected), 'Actions', and 'Advanced'. A note below the tabs says: 'Specify whether the business rule should run on Insert or Update. Use Filter Conditions to specify under which conditions the business rule should run.' There is also a 'Submit' button at the bottom right of the form area.

4. In when to run Check Insert and Update

The screenshot shows the 'Business Rule' configuration page for a 'New record' rule. At the top, it says 'Specify whether the business rule should run on Insert or Update. Use Filter Conditions to specify under which conditions the business rule should run.' Below this, there are fields for 'When' (set to 'before') and 'Order' (set to 100). To the right, checkboxes are checked for 'Insert' and 'Update', while 'Delete' and 'Query' are unchecked. Below these are sections for 'Filter Conditions' (with 'Add Filter Condition' and 'Add OR Clause' buttons) and 'Role conditions' (with a pencil icon). A 'Submit' button is at the bottom left.

5. In Advance(we write the code): Write the below code >>

```
(function executeRule(current, previous /*null when async*/) {  
  
    varFamilyExpenses = 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  
    {  
        varNewFamilyExpenses = 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);

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

The screenshot shows the Business Rule configuration interface for a 'New record' rule. The 'Advanced' tab is active. The script editor contains the following ECMAScript 2021 code:

```
28 var NewFamilyExpenses = new GlideRecord('u_family_expenses');
29
30 NewFamilyExpenses.u_date = current.u_date;
31
32 NewFamilyExpenses.u_amount = current.u_expense;
33
34 NewFamilyExpenses.u_expense_details += ">" + current.u_comments + ":" + "Rs." + current.u_expense
+ "/";
35
36 NewFamilyExpenses.insert();
37
38 }
39
40
41 })(current, previous);
```

Milestone 8: Relationship

Activity 1: Configure the Relationship

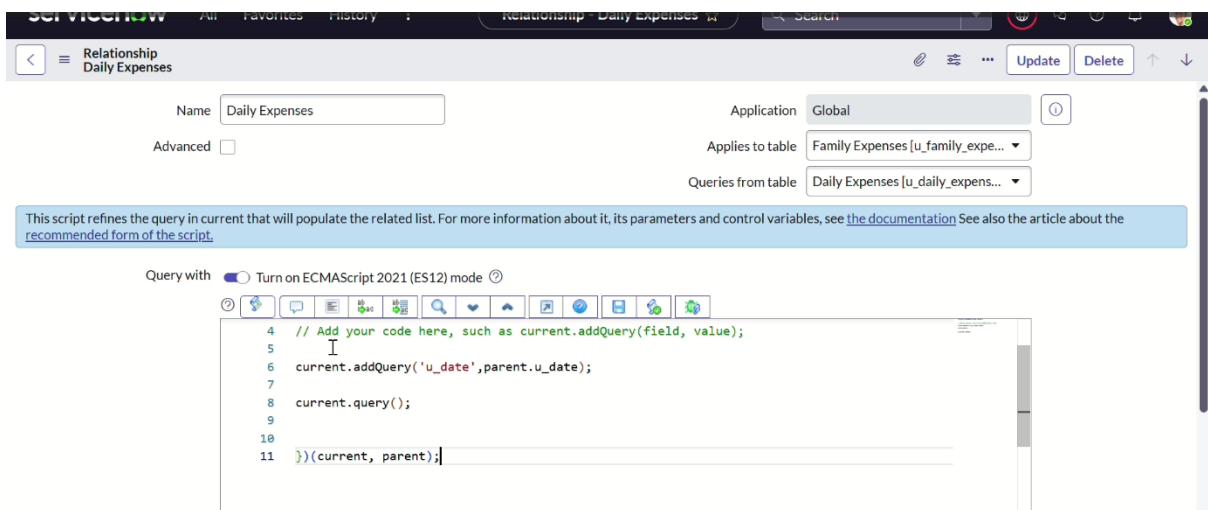
1. Go to All >> In the filter search for Relationships >> Open Relationships.
2. In that, open Daily Expenses Relationship.
3. For Applies to table : Select Family Expenses.
4. In Query with : write the below Query.

```
(functionrefineQuery(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.



The screenshot shows the Zoho CRM interface with the 'Relationship' tab selected for 'Daily Expenses'. The top navigation bar includes 'All', 'Favorites', 'History', 'Search', 'Update', and 'Delete' buttons. The main area displays settings for a relationship named 'Daily Expenses' with 'Global' application scope, applying to the 'Family Expenses [u_family_expe...]' table, and querying from the 'Daily Expenses [u_daily_expens...' table. A note at the bottom states: '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](#)'. Below this is a code editor window titled 'Query with' containing ECMAScript 2021 code:

```
4 // Add your code here, such as current.addQuery(field, value);
5
6 current.addQuery('u_date',parent.u_date);
7
8 current.query();
9
10 })(current, parent);
```