

Calculating Family Expenses using ServiceNow

Team ID: LTVIP2026TMIDS49388

Team Size: 4

Team Leader: Nalle Siri Chandana

Team Member: Tavudooji Santhosh Teja

Team Member: Aketi Chandra Kantha Gayatri Devi

Team Member: Kapavarapu Hema Sireesha

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)**

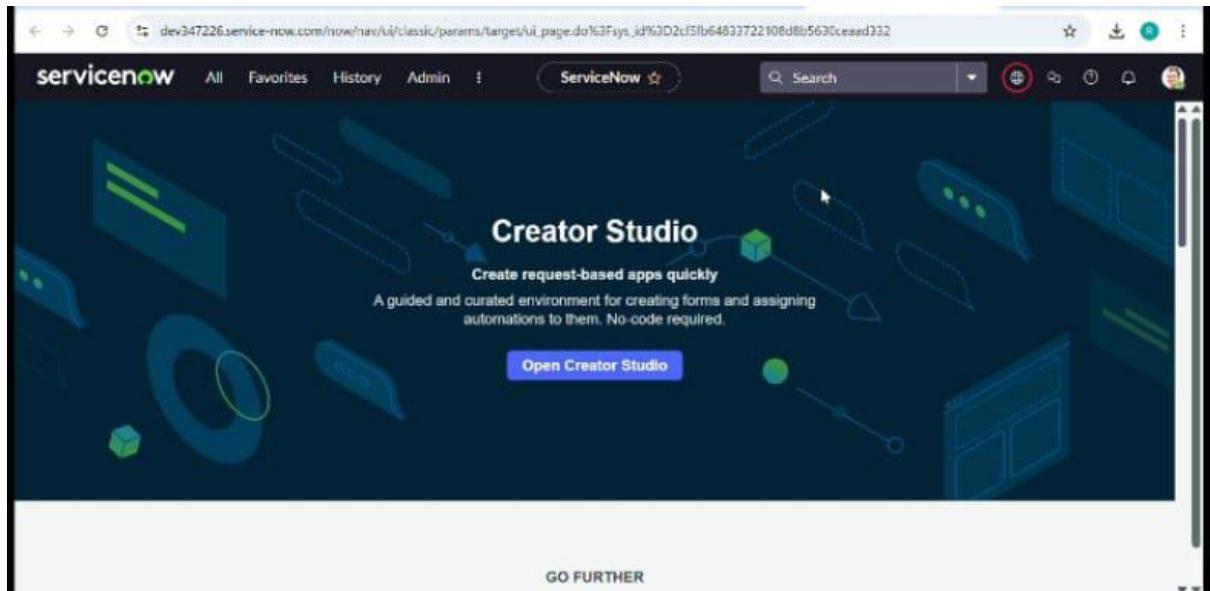
TASK INITIATION

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 a 'Create Educational org...' form for an 'Update Set'. The form fields include:

- * Name: Family Expenses
- State: In progress
- Parent: (empty)
- Release date: (empty)
- Description: (empty)

At the bottom of the form, there are two buttons: 'Submit' and 'Submit and Make Current'. Above the buttons, there are links for 'Application scope: Global' and 'Update set: Default [Global]'. The top navigation bar includes 'All', 'Favorites', 'History', 'Workspaces', and 'Update Set - Create Educational org...'. The status bar at the bottom right shows 'Relaunch to update'.

Milestone 3: Table(Family Expenses)

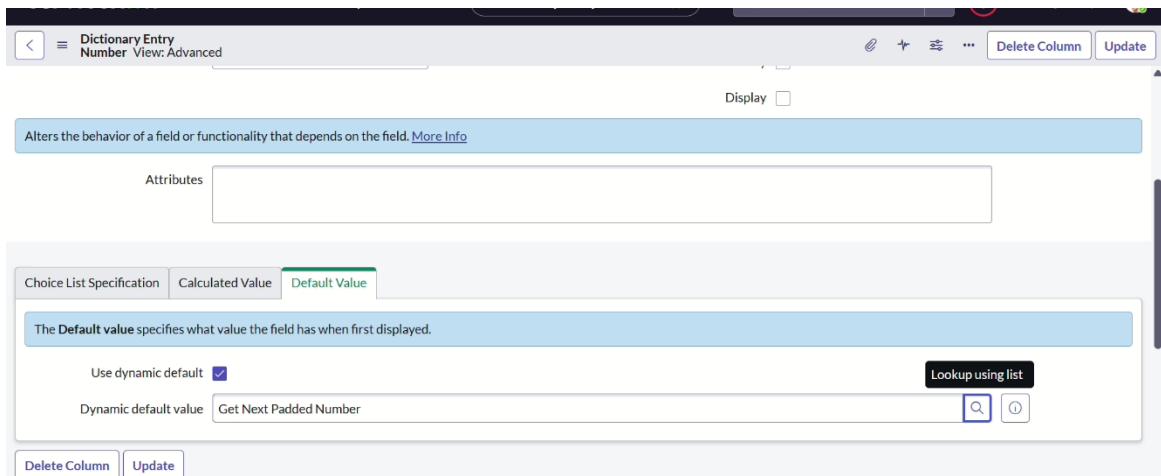
Activity 1: Creation of Family Expenses Table

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 'Table - New Record' page. The 'Label' field is set to 'Family Expenses', 'Name' is 'u_family_expenses', and 'New menu name' is 'Family Expenditure'. Other settings include 'Application' as Global, 'Create module' checked, 'Create mobile module' checked, and 'Add module to menu' set to 'Create new'. A note at the top says 'ServiceNow recommends creating custom tables in scoped applications. To learn more about creating scoped applications, click [here](#).'. A tooltip at the bottom left 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. [More Info](#)'.

Activity 2: Creation of Family Expenses Table

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 : 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.

Table - Family Expenses						
Dictionary Entries		Column label	Type	Reference	Max length	Default value
		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 'Number New record' form for the 'Family Expenses' table. The 'Table' field is set to 'Family Expenses'. The 'Prefix' field contains 'MFE'. The 'Number' field is set to '1,000'. The 'Application' field is set to 'Global'. The 'Number of digits' field is set to '7'. At the bottom left is a 'Submit' button.

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.

The screenshot shows the 'Form Design' interface for the 'Family Expenses [u_family_expenses]' table. The main area displays a 2-column layout with fields: 'Number', 'Amount', and 'Date'. Below this is a 1-column section for 'Expense Details'. The left sidebar lists fields such as 'Created', 'Updated', and 'Activities (filtered)'. The top right features 'Undo' and 'Save' buttons.

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.

The screenshot shows the ServiceNow 'Table - New Record' page. At the top, it says 'Table - New Record'. Below that, there's a note: '(1) ServiceNow recommends creating custom tables in scoped applications. To learn more about creating scoped applications, click [here](#).'. A blue box contains the text: '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.' Below this, there are several input fields:

- * Label: Daily Expenses
- * Name: u_daily_expenses
- Extends table: (empty)
- Application: Global
- Create module:
- Create mobile module:
- Add module to menu: Family Expenditure
- Remote Table:

At the bottom right are 'Submit' and 'Cancel' buttons.

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

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.

The screenshot shows the ServiceNow Developers interface with the 'Table - Daily Expenses' tab selected. In the 'Dictionary Entries' section, a new column 'Comments' is being added. The 'Type' is set to 'String' and the 'Max length' is set to '800'. Other columns listed include Updated by, Updates, Updated, Created by, Created, Sys ID, Number, Date, Expense, Family Member Name, and another 'Comments' column.

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.

The screenshot shows the 'Advanced' view for a 'Number' field. The 'Default Value' tab is selected. It shows that 'Use dynamic default' is checked and 'Dynamic default value' is set to 'Get Next Padded Number'. There is also a 'Lookup using list' button. At the bottom, there are 'Delete Column' and 'Update' buttons.

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, and at the bottom right are "Related Links" (Show Counter).

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.

The screenshot shows the ServiceNow Form Design tool. The title bar says "Form Design". The main area displays the "Daily Expenses [u_daily_expenses]" form with the following fields arranged in two columns:

Number	Family Member Name
Date	Expense

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

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". Below the title, there are fields for "Name" (set to "Daily Expenses") and "Application" (set to "Global"). Under the "Advanced" section, there is a checkbox. To the right, dropdown menus show "Applies to table" as "Family Expenses [u_family_expe...]" and "Queries from table" as "Daily Expenses [u_daily_expens...>". A note at the bottom left explains a script used to refine the query. The bottom half of the screen features a code editor with a toolbar above it, containing various icons for file operations and code snippets. The code itself is a function named "refineQuery" with a comment encouraging users to add their own logic.

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

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 the ServiceNow classic UI for configuring related lists. The top navigation bar includes links for All, Favorites, History, Admin, and a search bar labeled 'Search'. Below the header, there's a toolbar with icons for search, refresh, and user profile. The main content area has a title 'Configuring related lists on Family Expenses form'. On the left, a list titled 'Available' contains 'Attachments'. On the right, a list titled 'Selected' contains 'Daily Expenses'. Between these two lists are two blue arrows: a right-pointing arrow above a left-pointing arrow, indicating the direction of selection between the two panels. At the bottom of each panel are 'Cancel' and 'Save' buttons. The 'Selected' panel has scroll bars on its right and bottom edges.

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

A 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. [More Info](#)

Name: Family Expenses BR Application: Global

Table: Daily Expenses [u_daily_expenses...]

Active: Advanced:

When to run: Actions Advanced

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

4. In when to run Check Insert and Update

Business Rule
New record

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

When: before Order: 100

Insert:
Update:
Delete:
Query:

Filter Conditions: [Add Filter Condition](#) [Add OR Clause](#)

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

Role conditions: [Edit](#)

Submit

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

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

```
    var familyExpenses = new GlideRecord('u_family_expenses');
```

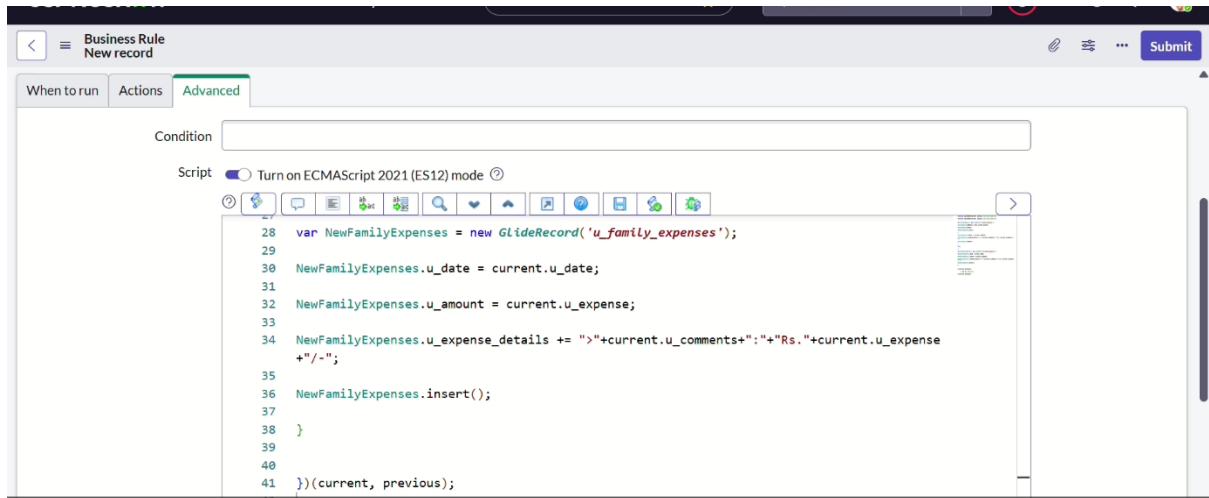
```

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
{
    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.



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.

