

SRI VIDYA COLLEGE OF ENGINEERING & TECHNOLOGY

VIRUDHUNAGAR 626005

Approved by AICTE, New Delhi and Affiliated to
Anna University, Chennai



Subject Name : ServiceNow Administrator (NM1051)
(Under Naan Mudhalvan Scheme)

Project Title : Calculating Family Expenses using ServiceNow

Team ID: NM2025TMID04202

Team Members

K.Sriram - 922022104031

R.Sheela Devi -922022104030

G.Vijaya Prabakar-922022104303

L.Yogesh-922022104041

Calculating Family Expenses using ServiceNow

1. Objective

The main objective of this project is to **develop a family expense management system** using the **ServiceNow platform**. The system helps users efficiently track, categorize, and manage family expenses. It aims to promote financial awareness and better budgeting within the household.

2. Introduction

In today's fast-paced world, managing family expenses is a critical part of financial stability. Manual tracking often leads to confusion, errors, and lack of insight into spending patterns.

This project leverages **ServiceNow**, a robust cloud-based platform, to create an automated and user-friendly **Family Expense Tracker**.

The application allows users to record expenses, set budgets, categorize spending, and generate real-time reports.

By building this project, we demonstrate how ServiceNow can be extended beyond IT service management to real-world, practical use cases like family budgeting.

3. Project Scope

- Expense Categorization
 - Budget Setting and Comparison
 - Real-time Expense Tracking
-

5. Tools and Technologies Used

Tool/Technology Description

ServiceNow Cloud platform for workflow automation

Update Sets To capture and move customizations

Tables & Fields Used to store expense records

Forms & Lists For user interaction and record display

Reports Module For data visualization and analytics

Browser Any modern browser (Chrome/Edge)

6. System Requirements

Hardware Requirements

- Processor: Dual Core or higher
- RAM: Minimum 4 GB
- Internet Connection: Stable broadband

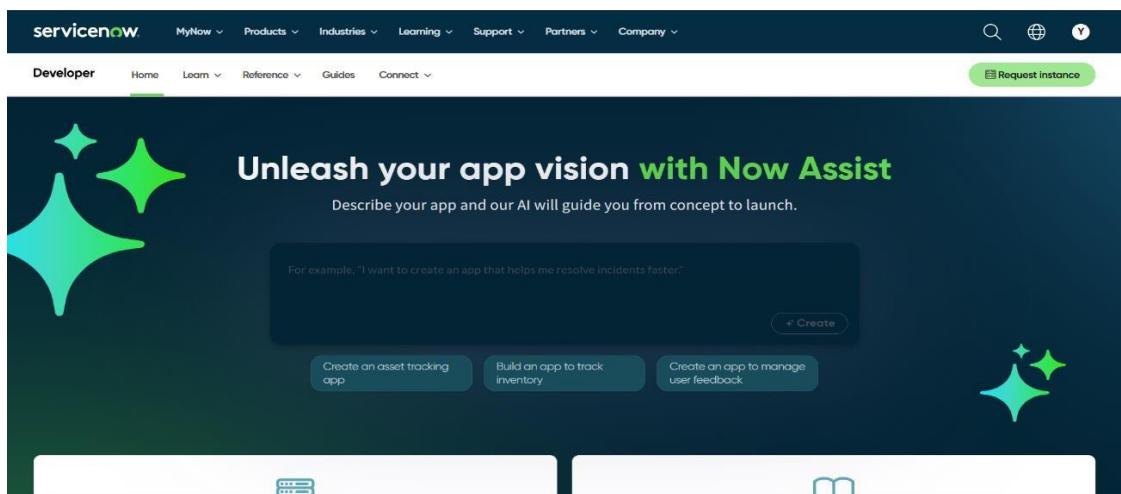
Software Requirements

- Web Browser (Google Chrome preferred)
- ServiceNow Developer Instance
- ServiceNow Account (developer.servicenow.com)

7. Project Implementation Steps

Step 1: Setting up ServiceNow Instance

1. Visit <https://developer.servicenow.com>
2. Sign up for a free developer account.
3. Log in to your instance to start customizing.



Step 2: Creation of a New Update Set

1. Navigate to All → Local Update Sets → New
2. Enter details:
 - **Name:** Family Expenses
3. Click **Submit and Make Current**
This ensures all future configurations are captured under the same update set.

Step 3: Creation of Family Expenses Table

1. Navigate to **All → Tables → New**
2. Enter details:
 - **Label:** Family Expenses
 - **Name:** (Auto-populated)
 - **New menu name:** Family Expenditure
3. Right-click the header and select **Save**.

Step 4: Creation of Columns (Fields)

In the **Columns** section, add the following fields:

Column Label	Type	Max Length
Number	String	-
Date	Date	-
Amount	Integer	-
Expense Details	String	800

After adding all fields, **Save** the table.

Table - Family Expenses

Dictionary Entries

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

Related Links

Step 5: Auto-Number for Family Expenses

Open the Number field → Advanced View → enable Use dynamic default with Get Next Padded Number → Update.

Then go to Number Maintenance → New, set Table: Family Expenses, Prefix: MFE, and click Submit.

Step 6: Configure Family Expenses Form

Go to Family Expenses → New → Configure → Form Design.

Make Number read-only, and set Date & Amount as mandatory → Save.

Step 7: Create Daily Expenses Table

Navigate to Tables → New, name it Daily Expenses, and link to Family Expenditure. Add fields: Number, Date, Expense, Family Member Name, Comments, then Save.

Label: Daily Expenses

Name: u.daily.expenses

Extends table:

Application Global

Create module:

Create mobile module:

Application Menu

Add module to menu: Family Expenditure

Step 8: Auto-Number for Daily Expenses

Open Number field → enable Get Next Padded Number → Update.

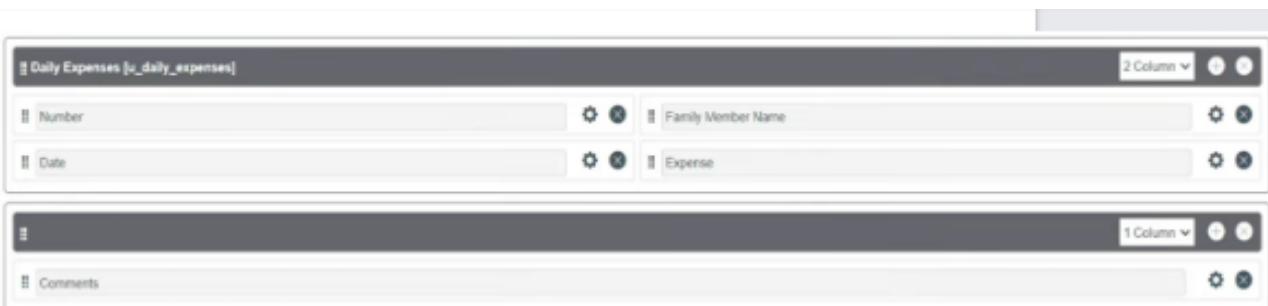
In Number Maintenance, create new with Table: Daily Expenses, Prefix: MDE, then Submit.



Step 9: Configure Daily Expenses Form

Go to Daily Expenses → New → Configure → Form Design.

Set Number read-only, and Date & Family Member Name mandatory → Save.



Step 10: Create Relationship

Open Relationships → New, set Name: Daily Expenses, Applies to Table: Family Expenses, and link to Daily Expenses → Save.

Add Daily Expenses as a related list in Family Expenses → Save.

The screenshot shows the ServiceNow interface for creating a new relationship record. The top navigation bar includes 'All', 'Favorites', 'History', 'Workspaces', 'Relationship - New Record', 'Search', and various system icons. The main form has fields for 'Name' (set to 'Daily Expenses'), 'Advanced' (unchecked), 'Application' (set to 'Global'), 'Applies to table' (set to 'Family Expenses [u_family_expenses]'), and 'Queries from table' (set to 'None'). A script editor area contains a code snippet:

```

Query with: Turn on ECMAScript 2021 (ES12) mode ⓘ
(function refineQuery(current, parent) {
    // Add your code here, such as current.addQuery(field, value);
})(current, parent);

```

A 'Submit' button is at the bottom left.

Step 11: Create Business Rule

Go to **Business Rules** → **New**, name it **Family Expenses BR**, and select **Table: Daily Expenses**. Enable **Advanced, Insert, Update** and add script to auto-update Family Expenses totals → Save.

The screenshot shows the ServiceNow interface for creating a new business rule record. The top navigation bar includes 'All', 'Favorites', 'History', 'Workspaces', 'Relationship - New Record', 'Search', and various system icons. The main form has fields for 'Name' (set to 'Daily Expenses'), 'Advanced' (unchecked), 'Application' (set to 'Global'), and 'Applies to table' (set to 'Family Expenses [u_family_expenses]', highlighted with a red arrow). A script editor area contains a code snippet:

```

Query with: Turn on ECMAScript 2021 (ES12) mode ⓘ
(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);

```

Red arrows point to the 'Update' button at the bottom left and the 'Applies to table' field. A red box highlights the entire query script area.

Step 12: Refine Relationship Query

Open **Daily Expenses Relationship**, set **Applies to: Family Expenses**. Add query script to match **u_date** fields between tables → Update.

This screenshot shows the ServiceNow interface for managing relationships between tables. The top navigation bar includes 'All', 'Favorites', 'History', 'Workspaces', and a search bar. The main title is 'Relationship - Daily Expenses'. Below the title, there's a form with fields for 'Name' (set to 'Daily Expenses'), 'Advanced' (unchecked), 'Application' (set to 'Global'), 'Applies to table' (set to 'Family Expenses [u_auto_populated]'), and 'Queries from table' (set to 'Daily Expenses [u_daily_expenses]'). 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.' A large code editor window contains the following ECMAScript 2021 code:

```

1 (function refineQuery(current, parent) {
2     // Add your code here, such as current.addQuery(field, value);
3     current.addQuery('u_date',parent.u_date);
4
5     current.query();
6
7     current;
8
9     10 })(current, parent);
11

```

Below the code editor are buttons for 'Run Query Diagnostics', 'Update', and 'Delete'. A 'Related Links' section includes a 'Run Point Scan' link.

Step 13: Testing & Output

Add test entries in **Daily Expenses** and verify automatic updates in **Family Expenses**. Check related lists, totals, and generate date-wise or member-wise reports.

This screenshot shows the ServiceNow interface for creating business rules. The top navigation bar includes 'All', 'Favorites', 'History', 'Workspaces', and a search bar. The main title is 'Business Rule - Family Expenses BR'. Below the title, there's a form with tabs for 'When to run' (selected), 'Actions', and 'Advanced'. Under 'Actions', there's a 'Condition' field and a 'Script' field. The 'Script' field contains the following ECMAScript 2021 code:

```

1 (function executeRule(current, previous /*null when async*/) {
2     // Add your code here
3     var FamilyExpenses = new GlideRecord('u_family_expenses');
4
5     FamilyExpenses.addQuery('u_date',current.u_date);
6
7     FamilyExpenses.query();
8
9     if(FamilyExpenses.next())
10    {
11        FamilyExpenses.u_amount += current.u_expense;
12
13        FamilyExpenses.u_expense_details += ">" + current.u_comments + ":" + "Rs." + current.u_expense + "-";
14
15        FamilyExpenses.update();
16
17    }
18
19
20
21
22
23    else

```

Conclusion

A complete ServiceNow-based expense tracker with automated calculations, dynamic relationships, and real-time reporting — demonstrating workflow automation for practical financial management.

