# Calculating Family Expenses

# Project Title:

Calculating Family Expenses using ServiceNow

# Developed By:

1. A.Surekha
2. M.Umamaheswari
3. M.Triveni
4. B.Usha Rani

# Date:

30th June 2025

# Objective:

To design and implement an automated expense management system using ServiceNow that tracks daily family expenses and aggregates them into family-level summaries with minimal manual effort.

# System Overview:

This project is built using ServiceNow’s Table, Form Design, Business Rules, and Relationships features. It comprises:  
- Two Custom Tables: Family Expenses and Daily Expenses  
- One Business Rule: To auto-update Family Expenses  
- One Relationship: To link Daily Expenses to Family Expenses  
- Auto-Numbering for unique record identification  
- Form customizations for better UI/UX

# Table Design:

## 1. Family Expenses Table

Table Name: u\_family\_expenses

Purpose: To record aggregated expenses by date

|  |  |  |
| --- | --- | --- |
| Field Name | Type | Properties |
| Number | String | Auto-generated, Read-only |
| Date | Date | Mandatory |
| Amount | Integer | Summed from daily expenses |
| Expense Details | String | Max length: 800 |

## 2. Daily Expenses Table

Table Name: u\_daily\_expenses

Purpose: To record each individual expense entry

|  |  |  |
| --- | --- | --- |
| Field Name | Type | Properties |
| Number | String | Auto-generated, Read-only |
| Date | Date | Mandatory |
| Expense | Integer |  |
| Family Member Name | Reference | Mandatory, refers to sys\_user |
| Comments | String | Max length: 800 |
| Family Expense | Reference | Refers to u\_family\_expenses |

# Table Relationship:

Relationship Name: Daily Expenses  
Applies to Table: u\_family\_expenses  
Referenced Table: u\_daily\_expenses  
  
Query Script:

(function refineQuery(current, parent) {  
 current.addQuery("u\_date", parent.u\_date);  
 current.query();  
})(current, parent);

# Business Rule:

Name: Family Expenses BR  
Table: u\_daily\_expenses  
When to Run: On Insert and Update  
Script:

(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);

# Auto-Numbering:

Configured using Number Maintenance  
  
For Family Expenses:  
- Table: u\_family\_expenses  
- Prefix: MFE  
  
For Daily Expenses:  
- Table: u\_daily\_expenses  
- Prefix: MDE

# Form Design:

Customized Forms for better data entry and visibility.  
- Read-only Field: Number  
- Mandatory Fields:  
 - In Daily Expenses: Date, Family Member Name  
 - In Family Expenses: Date  
  
- Related List:  
 - Daily Expenses added to Family Expenses form using Configure > Related Lists

# Outcomes:

- Each Daily Expense auto-updates a Family Expense record.  
- If a Family Expense does not exist for that date, it is created.  
- Clear linkage between summary and detailed entries.  
- Fully automated with minimal manual input required.

# Future Enhancements:

- Create a custom Family Members table instead of using sys\_user.  
- Add filtering by family member in reports.  
- Display totals dynamically in forms.  
- Build dashboards with charts (Bar/Pie).

# Conclusion

The *Family Expenses Management System* is a practical and efficient solution for tracking household expenditures using the ServiceNow platform. By integrating automated business logic, relational data models, and a user-friendly interface, the system minimizes manual work while providing accurate and real-time expense summaries. It offers a strong foundation for household financial planning and is easily extendable to support advanced features like data visualization and analytics. This project demonstrates how low-code tools can be effectively used to address real-world personal finance management challenges.