



A.R.J COLLEGE OF ENGINEERING& TECHNOLOGY

EDAIYARNATHAM, MANNARGUDI-614 001 (APPROVED BY AICTE, NEW  
DELHI & AFFILIATED FOR ANNA UNIVERSITY) (AN ISO 9001:2000 CERTIFIED  
INSITUTION)



## **NM1051—SERVICENOW ADMINISTRATOR**

### **DEPARTMENT OF COMPUTER SCIENCE**

### **PROJECT TITLE: CALCULATING FAMILY EXPENSES USING SERVICE NOW**

**TEAM ID : NM2025TMID06941**

### **SUBMITTED BY**

- 1. Team Leader : Sivasankari S**
- 2. Team member : Kalidhasan S**
- 3. Team member : Ranjith M**
- 4. Team member : Shriwadhsan C**

# Calculating Family Expenses using Service Now

## ABSTRACT:

The project “**Calculating Family Expenses using ServiceNow**” aims to automate the process of recording, managing, and calculating family expenses efficiently. Traditionally, families use manual methods such as notebooks or spreadsheets to track expenses, which can be time-consuming and prone to human errors.

This project uses the **ServiceNow platform** to simplify this process by creating two connected tables — **Family Expenses** and **Daily Expenses** — and linking them with a **Business Rule** that automatically updates total amounts whenever a new daily expense is added. The system ensures data consistency, accuracy, and quick access to information through a user-friendly interface. The main goal of the project is to provide a reliable, low-code solution for effective financial tracking within a family environment, reducing manual work and improving decision-making.

## **TABLE OF CONTENTS:**

1. Introduction
2. 2.1 Overview of Family Expense Management
- 2.2 Need for Automation
- 2.3 Role of ServiceNow in Data Management
3. Existing System
- 3.1 Manual Expense Management
- 3.2 Challenges in Current System
- 3.3 Limitations Identified
4. Proposed System
- 4.1 Objectives of the Proposed System
- 4.2 Key Features
- 4.3 Scope of Implementation
5. System Analysis
- 5.1 Functional Requirements
- 5.2 Non-Functional Requirements
- 5.3 System Architecture
6. System Design
- 6.1 System Architecture Diagram
- 6.2 Use Case Diagram
- 6.3 Entity Relationship Diagram (ERD)
- 6.4 Data Flow Diagram (DFD)
7. Module Description
- 7.1 Family Expenses Module
- 7.2 Daily Expenses Module
- 7.3 Relationship Configuration Module
8. Methodology
- 8.1 Creation of Update Set
- 8.2 Creation of Tables
- 8.3 Table Relationship Setup
- 8.4 Related List Configuration
- 8.5 Reference Field Creation
- 8.6 Query Configuration

9. Result

10. Advantages

11. Disadvantages

12. Future Enhancements

13. Conclusion

15. References

# 1. INTRODUCTION

Managing family expenses is a vital part of maintaining financial stability. In many cases, daily spending is tracked manually, which often leads to calculation errors and difficulties in maintaining records. The project “**Calculating Family Expenses using ServiceNow**” provides an automated way to manage this process using ServiceNow’s low-code environment.

By creating two interconnected tables — *Family Expenses* and *Daily Expenses* — users can easily record their daily transactions and automatically calculate total spending. The use of **Business Rules** ensures that updates happen instantly, maintaining accuracy and efficiency.

This system can be customized to handle any number of family members or spending categories. It offers a modern approach to expense management by combining automation, simplicity, and scalability. Ultimately, the project demonstrates how ServiceNow can be used effectively beyond IT workflows, extending into personal and household financial management.

## 2.1 OVERVIEW OF FAMILY EXPENSE MANAGEMENT

Managing household expenses manually can be time-consuming and error-prone. This project provides a digital method to manage financial data effectively.

## **2.2 NEED FOR AUTOMATION**

Automation reduces human errors and improves efficiency. Using ServiceNow, we can handle expenses faster with accurate data updates.

## **2.3 ROLE OF SERVICENOW IN DATA MANAGEMENT**

ServiceNow acts as a low-code platform that helps in creating custom applications, managing data tables, and automating business processes easily.

## **3. EXISTING SYSTEM**

### **3.1 MANUAL EXPENSE MANAGEMENT**

In the current system, family members record expenses in notebooks or spreadsheets manually.

### **3.2 CHALLENGES IN CURRENT SYSTEM**

- Time-consuming calculations
- Errors in totaling expenses
- No data relationship between daily and family records

### **3.3 LIMITATIONS IDENTIFIED**

The existing system lacks automation, data linkage, and accuracy in tracking overall spending.

## **4. PROPOSED SYSTEM**

### **4.1 OBJECTIVES OF THE PROPOSED SYSTEM**

- Automate expense calculation
- Link family and daily expenses dynamically
- Store all records in one platform

### **4.2 KEY FEATURES**

- Family and Daily Expenses tables
- Reference fields for linking records
- Automatic total calculation using scripts

### **4.3 SCOPE OF IMPLEMENTATION**

The project can be implemented for small families or even extended for organizations to manage expense tracking.

## **5. SYSTEM ANALYSIS**

### **5.1 FUNCTIONAL REQUIREMENTS**

- Create and manage Family Expenses
- Add daily records
- Automatically calculate totals

### **5.2 NON-FUNCTIONAL REQUIREMENTS**

- Easy to use interface
- Reliable and secure data storage
- Fast query response

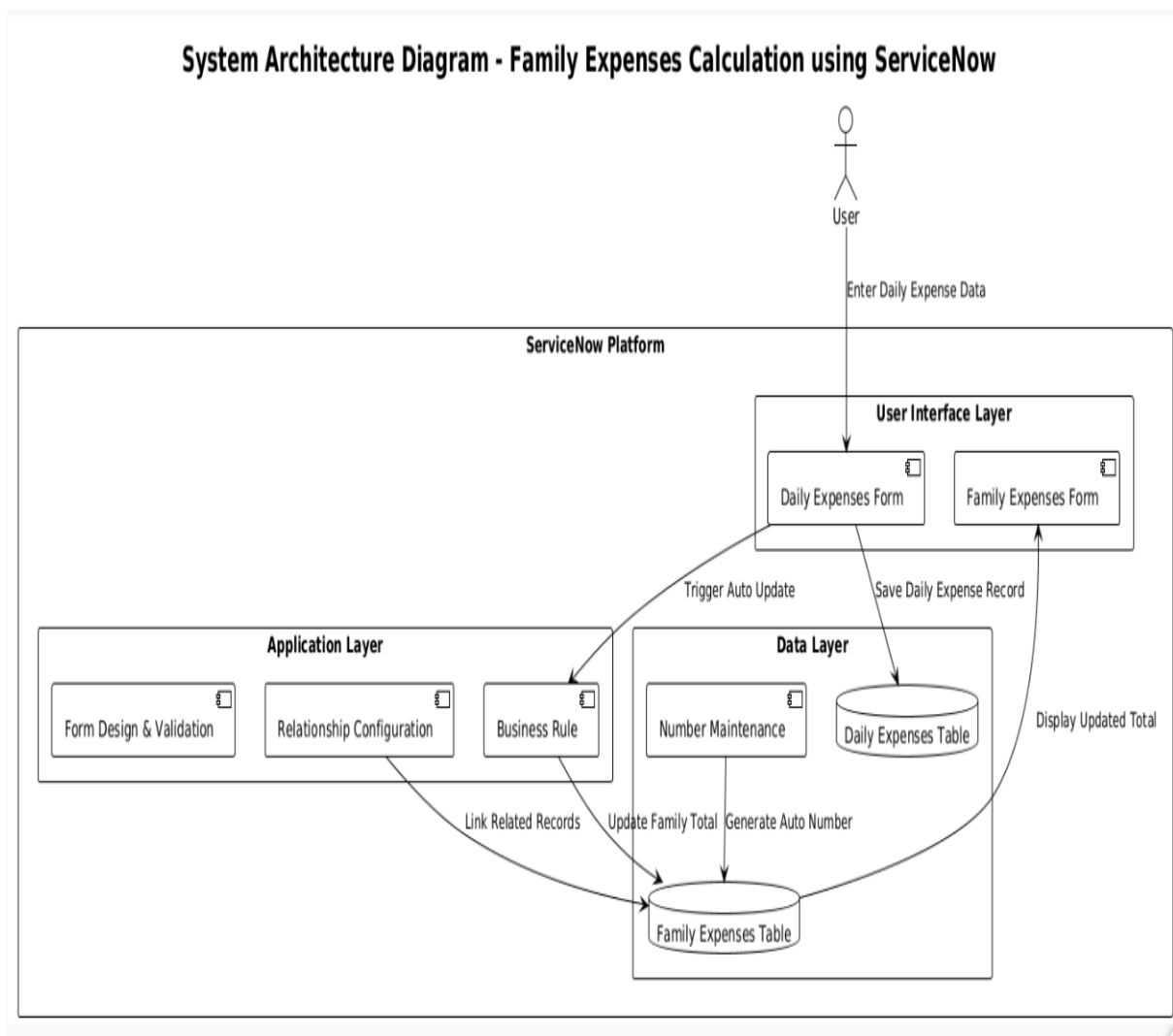
## 5.3 SYSTEM ARCHITECTURE

The architecture includes:

1. User Interface (Forms & Lists)
2. ServiceNow Database Tables
3. Business Rules and Client Scripts for automation

## 6. SYSTEM DESIGN

### 6.1 SYSTEM ARCHITECTURE DIAGRAM



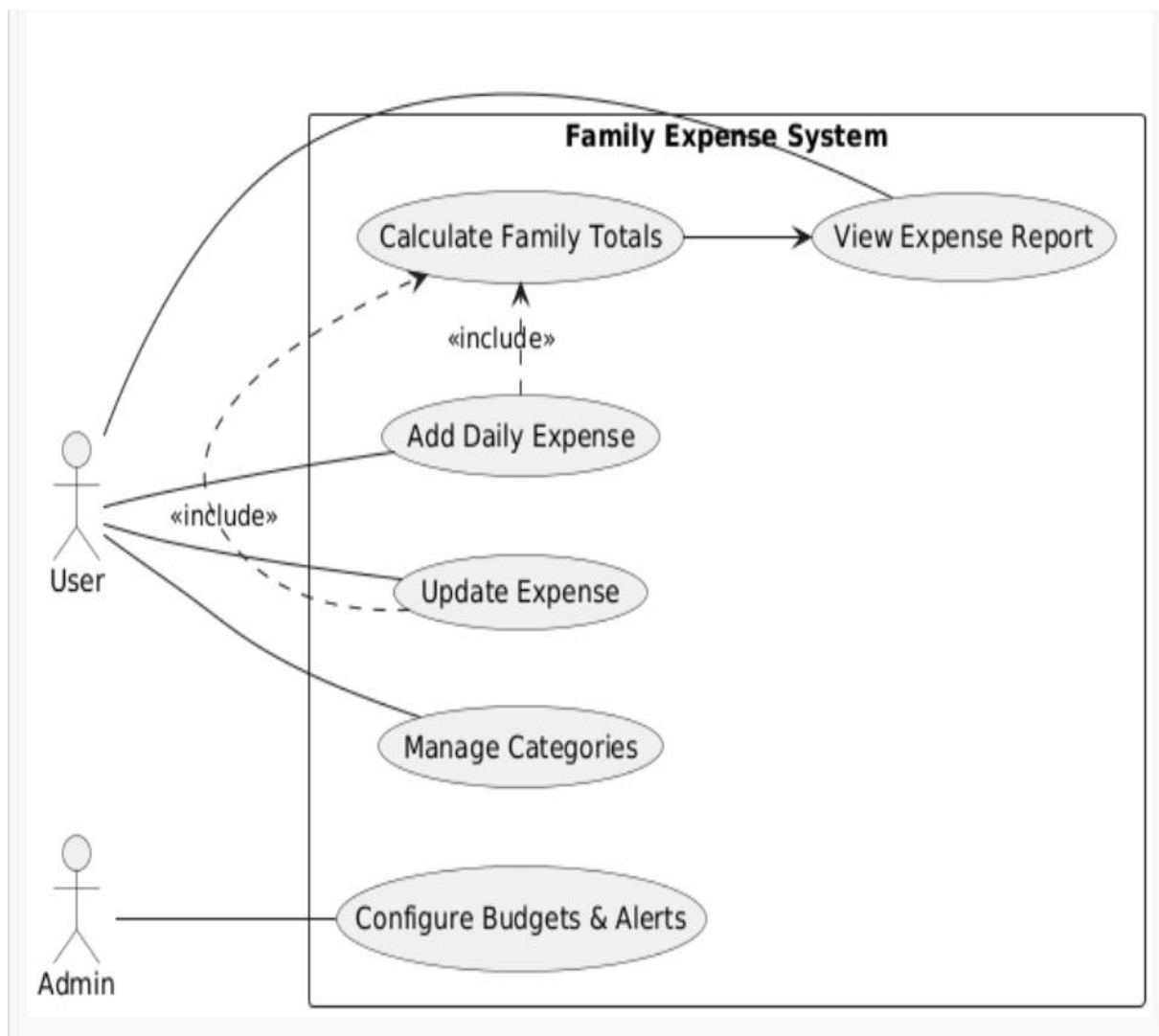


## 6.2 USE CASE DIAGRAM

Actors: User (Family Member), System (ServiceNow)

Use Cases:

- Add family details
- Add daily expenses
- View related expenses
- Calculate total amount
- Generate report



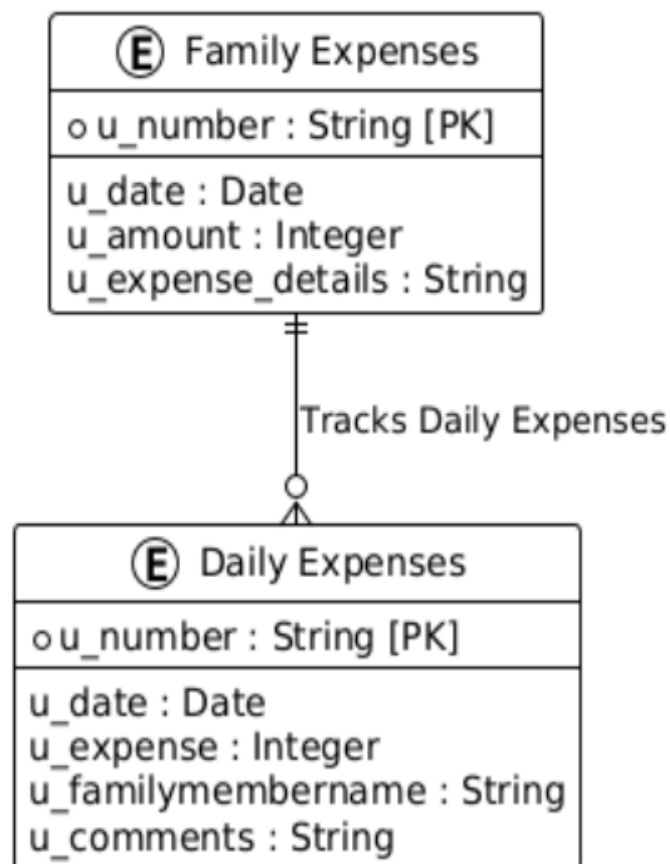
## 6.3 ENTITY RELATIONSHIP DIAGRAM (ERD)

### ENTITIES:

- Family\_Expenses (u\_familyname, u\_total, u\_date)
- Daily\_Expenses (u\_item, u\_amount, u\_date, u\_familyname)

Relationship: Family\_Expenses → Daily\_Expenses (One-to-Many)

### Entity Relationship Diagram (ERD)



## 6.4 DATA FLOW DIAGRAM (DFD)

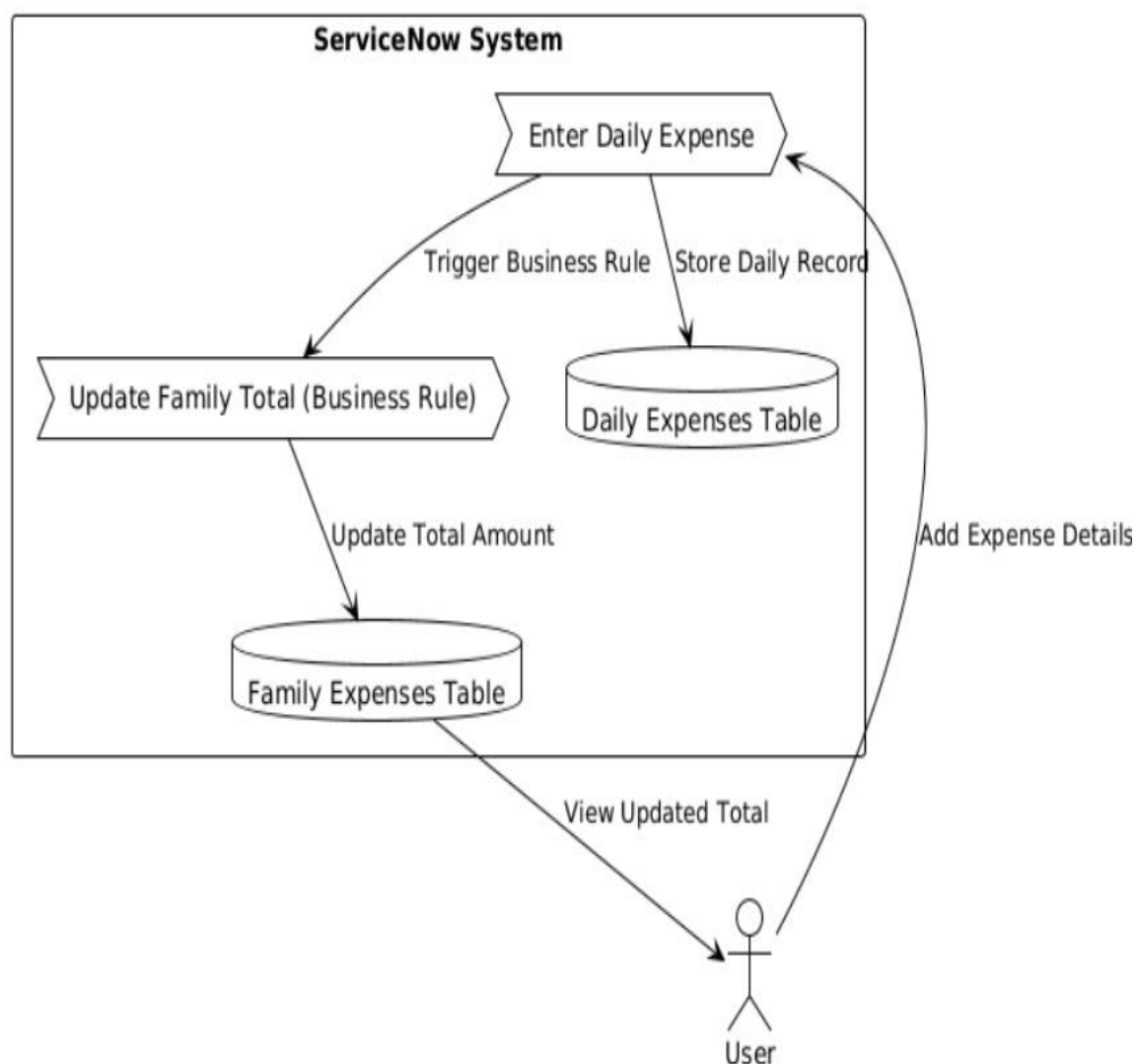
Level 0:

User → ServiceNow → Database → Output (Expense Summary)

Level 1:

Add Expense → Validate Data → Store in Daily\_Expenses → Update Family\_Expenses total

### Data Flow Diagram (Level 1)



## **7. MODULE DESCRIPTION**

### **7.1 FAMILY EXPENSES MODULE**

Stores family details and overall spending totals.

### **7.2 DAILY EXPENSES MODULE**

Stores individual daily spending items.

### **7.3 RELATIONSHIP CONFIGURATION MODULE**

Connects both tables using a reference field and query.

## **8. METHODOLOGY**

### **8.1 CREATION OF UPDATE SET**

Created a new update set named *Family Expenses*.

### **8.2 CREATION OF TABLES**

Two tables created: *Family Expenses* and *Daily Expenses*.

### **8.3 TABLE RELATIONSHIP SETUP**

Configured relationship linking *Family Expenses* with *Daily Expenses*.

### **8.4 RELATED LIST CONFIGURATION**

Added *Daily Expenses* to the *Family Expenses* related list.

### **8.5 REFERENCE FIELD CREATION**

Created reference type field `u_familyname` in the *Daily Expenses* table.

## 8.6 QUERY CONFIGURATION

Configured relationship query:

```
(function refineQuery(current, parent) {  
    current.addQuery('u_date',  
parent.u_date);  
    current.query();  
})(current, parent);
```

## IMPLEMENTATION:

### STEP 1 – CREATION OF UPDATE SET

1. Go to **All** → **Local Update Set** → **New**.
2. Enter:
  - **Name:** Family Expenses
3. Click **Submit** and **Make Current**.

The screenshot shows the ServiceNow web interface for creating a new update set. The browser tabs include 'ServiceNow Developers', 'Student', and 'Family Expenses | Update Set'. The URL is 'dev183710.service-now.com/now/nav/ui/classic/params/target/sys\_update\_set.do%3Fsys\_id%3D4f029356c37c321091b7b1f1b4013113%26syspar...'. A security update banner for MFA implementation is visible at the top. The main header shows 'servicenow' and navigation links. The breadcrumb trail is 'Update Set - Family Expenses'. The form fields are as follows:

Field	Value
Name	Family Expenses
State	In progress
Parent	
Release date	
Install date	
Installed from	
Description	
Application	Global
Created	2025-10-27 07:53:57
Created by	admin
Merged to	

An 'Update' button is located at the bottom left of the form.

## STEP 2 – CREATION OF FAMILY EXPENSES TABLE

1. Navigate to **All** → **Tables** → **New**.
2. Enter details:
  - **Label:** Family Expenses
  - **Menu Name:** Family Expenditure
3. Save the table.
4. Add columns:
  - Number (String)
  - Date (Date)
  - Amount (Integer)
  - Expense Details (String, Max length: 800)
5. Configure Auto-Number with prefix **MFE**.
6. In Form Design, make **Number** read-only and **Date**, **Amount** mandatory.

ServiceNow Developers x - Student x Family Expenses | Table | Service x +

dev183710.service-now.com/now/nav/ui/classic/params/target/sys\_db\_object.do%3Fsys\_id%3D05e49726c3f0f21091b7b11b40131ba%26sys... 🔍 ☆ Verify it's you

**Security Update: MFA Implementation**  
To enhance the security of your instance, we'll enable Multi-Factor Authentication (MFA) for all users performing non-SSO login in the instance. For more information, see MFA Implementation Guide. [Acknowledge](#)

**servicenow** All Favorites History Workspaces Admin **Table - Family Expenses** 🔍 Search

Table Family Expenses View: Tables **Delete** **Update** ↑ ↓

\* Label  Application  ⓘ

\* Name  Remote Table ☒

**Columns** Controls Application Access

≡ ▾ **Table Columns** for text 🔍 Search ⏪ ⏩ 1 to 5 of 5 ⏪ ⏩ **New**

Dictionary Entries

	Column label	Type	Reference	Max length	Default value	Display
X	Number	String	(empty)	40	javascript.getNextObjNumberPadded();	false
X	Date	Date	(empty)	40		false
	Sys ID	Sys ID (GUID)	(empty)	32		false
X	Amount	Integer	(empty)	40		false
X	Expense Details	String	(empty)	800		false
+	Insert a new row...					

## STEP 3 – CREATION OF DAILY EXPENSES TABLE

1. Navigate to **All** → **Tables** → **New**.
2. Enter details:
  - **Label:** Daily Expenses
  - **Menu:** Family Expenditure
3. Save the table.
4. Add columns:
  - Number (String)
  - Date (Date)
  - Expense (Integer)
  - Family Member Name (Reference)
  - Comments (String, Max length: 800)
5. Configure Auto-Number with prefix **MFE**.
6. In Form Design, make **Number** read-only and **Date**, **Family Member Name** mandatory.

The screenshot displays the ServiceNow 'Table - Daily Expenses' configuration page. The top navigation bar includes 'All', 'Favorites', 'History', 'Workspaces', and 'Admin'. The table name 'Daily Expenses' is shown in the breadcrumb. Below the table name, there are fields for 'Label' (Daily Expenses), 'Application' (Global), 'Name' (u\_daily\_expenses), and 'Remote Table'. A description box states: '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](#)'. The 'Columns' tab is selected, showing a list of columns with headers: Column label, Type, Reference, Max length, Default value, and Display. The columns listed are: Updated by (String, 40, false), Sys ID (Sys ID (GUID), 32, false), Updated (Date/Time, 40, false), Created (Date/Time, 40, false), Created by (String, 40, false), Updates (Integer, 40, false), and Number (String, 40, javascript.getNextObj(NumPadded);, false). The 'Number' column is highlighted with a red 'x' in the first column.

Column label	Type	Reference	Max length	Default value	Display
Updated by	String	(empty)	40		false
Sys ID	Sys ID (GUID)	(empty)	32		false
Updated	Date/Time	(empty)	40		false
Created	Date/Time	(empty)	40		false
Created by	String	(empty)	40		false
Updates	Integer	(empty)	40		false
Number	String	(empty)	40	javascript.getNextObj(NumPadded);	false

## STEP 4 – CREATION OF RELATIONSHIP

1. Navigate to **All** → **Relationships** → **New**.
2. Enter details:
  - **Name:** Daily Expenses
  - **Applies to Table:** Family Expenses
  - **Daily Expenses:** Daily Expenses
3. Click **Save**.

The screenshot shows the ServiceNow 'New' form for a Relationship named 'Daily Expenses'. The form is titled 'Relationship Daily Expenses'. It includes fields for 'Name' (set to 'Daily Expenses'), 'Application' (set to 'Global'), 'Applies to table' (set to 'Family Expenses [u\_st\_family\_expenses]'), 'Queries from table' (set to 'Daily Expenses [u\_daily\_expenses]'), and 'Reference field'. There are also checkboxes for 'Advanced' and 'Simple reference'. The 'Run Query Diagnostics', 'Update', and 'Delete' buttons are visible at the top right.

## STEP5 – Configuring Related List

1. Open **Family Expenses** → **New**.
2. Right-click header → **Configure** → **Related Lists**.
3. Add **Daily Expenses** to Selected Area.
4. Click **Save**.

The screenshot shows the 'Configuring related lists on Family Expenses form' dialog. It features two main sections: 'Available Attachments' and 'Selected Daily Expenses'. The 'Available Attachments' section is currently empty. The 'Selected Daily Expenses' section contains a list of 'Daily Expenses'. There are 'Add' and 'Remove' buttons between the two sections. At the bottom, there are 'View name' (set to 'Default view'), 'Cancel', and 'Save' buttons. Below the dialog, there are links for 'Related Links', 'Show versions', and 'Related list performance diagnostics'.



## STEP 6 – CREATION OF BUSINESS RULE

1. Navigate to **All** → **System Definition** → **Business Rules** → **New**.
2. Enter details:
  - **Name:** Family Expenses BR
  - **Table:** Daily Expenses
  - Check **Advanced**, and **Insert, Update**.
3. Add the following script:

```
(function executeRule(current, previous)
{
    var FamilyExpenses = new
GlideRecord('u_family_expenses');
    FamilyExpenses.addQuery('u_date',
current.u_date);
    FamilyExpenses.query();

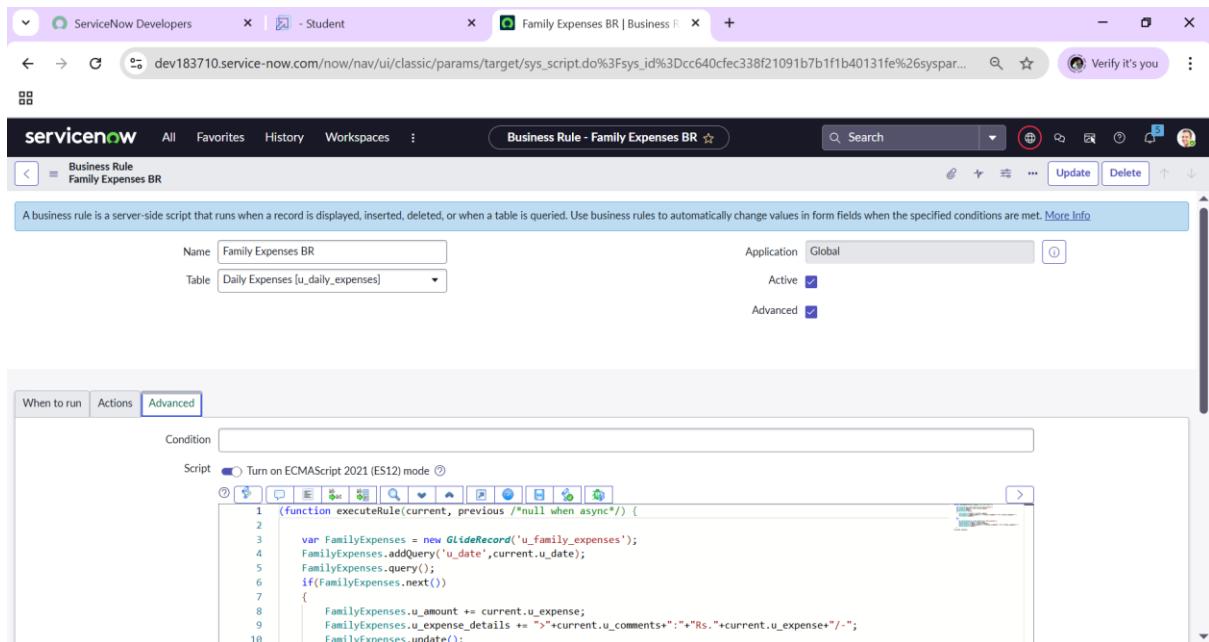
    if (FamilyExpenses.next()) {
        FamilyExpenses.u_amount +=
current.u_expense;
        FamilyExpenses.u_expense_details
+= ">" + current.u_comments + ":" + "Rs."
+ current.u_expense + "/-";
        FamilyExpenses.update();
    } else {
        var NewFamilyExpenses = new
GlideRecord('u_family_expenses');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

```



## STEP 7 – CONFIGURE THE RELATIONSHIP

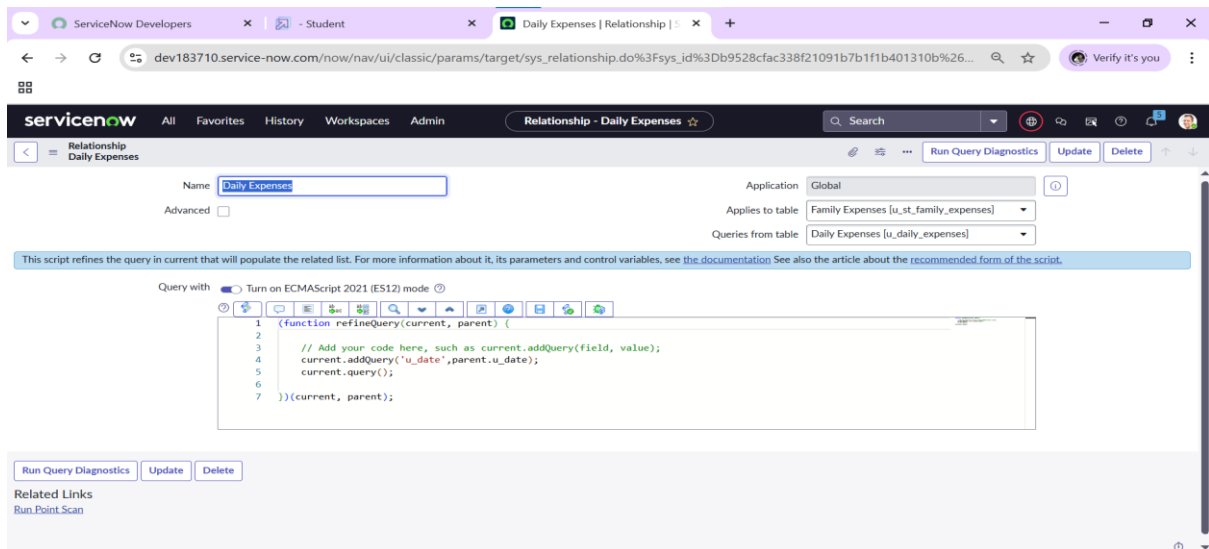
1. Navigate to **All → Relationships → Open Daily Expenses Relationship**.
2. Set **Applies to Table:** Family Expenses.
3. In “Query with” section, add:

```

(function refineQuery(current, parent) {
    current.addQuery('u_date',
parent.u_date);
    current.query();
})(current, parent);

```

4. Click **Update**.



## 9. RESULTS

Successfully created a dynamic and automated expense tracking system.

The Family and Daily Expenses tables are linked, and data flows correctly between them.

## 10. ADVANTAGES

- Simple and user-friendly
- Reduces manual calculation
- Accurate and fast data updates
- Automatic linkage between tables

## 11. DISADVANTAGE

- Works only inside ServiceNow
- Needs internet access
- Limited visual reporting features

## **12. FUTURE ENHANCEMENTS**

- Add data visualization using charts
- Enable mobile access
- Include monthly summary reports
- Integrate export to Excel feature

## **13. CONCLUSION**

The Family Expenses project is successfully developed using ServiceNow.

It automates the expense calculation process, saves time, and provides accurate results.

This project shows how ServiceNow can be effectively used for simple data management applications.

## **14. REFERENCES**

- ServiceNow Documentation Portal – <https://docs.servicenow.com>
- ServiceNow Developer Online Learning – <https://developer.servicenow.com>
- Educational Institution Workflow Automation Research Papers, IEEE Access Journals
- ServiceNow Admin & Fundamentals Training Material
- Official Tutorials and System Configuration Guidelines from ServiceNow Community