MAR EPHRAEM COLLEGE OF ENGINEERING AND TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

NAAN MUDHALVAN

A PROJECT WORK ON

CALCULATING FAMILY EXPENSES USING SERVICE NOW

SUBMITTED BY

Ashika G **9614222104015**

MAR EPHRAEM COLLEGE OF ENGINEERING AND TECHNOLOGY

MALANKARA HILLS, ELAVUVILAI, MARTHANDAM-629 171,
KANYAKUMARI DISRICT

DEPARTMENT OF ARTIFICIAL INTELLIGENCE & DATA SCIENCE

A PROJECT REPORT

2025-2026

*This is to certify that this is the bonafide record of the work done by **Mr./Ms.** ....**Register Number** .... of the **…………**(sem/Year), B.E/B.Tech, Department of **………………………………** **………………………**in the **………………………………………………………………** **………………………***

Staff-in-charge **Head of the Department**

University Examination held on…………………………….

Internal Examiner **External Examiner**

Table of Contents

CALCULATING FAMILY EXPENSES USING SERVICE NOW

Ideation Phase:

Problem Identification:

In most households, tracking daily expenses is done manually or through basic tools like spreadsheets, which often results in errors, lack of real-time visibility, and difficulty in analyzing spending patterns. Families need a smart and centralized way to manage their financial activities effectively.

Idea Generation:

The idea is to build a Family Expense Management System on the ServiceNow platform. ServiceNow's workflow automation, database management, and reporting features can be utilized to create a system that allows users to:

- Record and categorize daily expenses (e.g., groceries, bills, education, health).
- Set monthly or weekly budgets.
- Generate real-time expense reports and dashboards.
- Analyze spending patterns to make informed financial decisions.

Proposed Solution:

Using ServiceNow, we will design an automated application where users can:

- Log expenses through a user-friendly interface.
- Automatically calculate total expenses and remaining budget.
- View analytics through visual dashboards and reports.

Expected Outcome:

- The system will simplify financial tracking for families, reduce manual errors, and provide insights into saving opportunities. It will promote better financial planning and accountability within households.

Project Planning Phase:

- The project aims to design and develop a Family Expense Management System using ServiceNow to simplify the process of recording, categorizing, and analyzing household expenses. The application will automate calculations, generate reports, and provide insights to support better financial decisions.

Objectives:

- To develop a centralized platform for tracking family expenses.
- To automate expense calculations and report generation.
- To provide real-time dashboards for financial visualization.
- To enhance accuracy and reduce manual effort in expense management.

Project Scope:

The project covers:

- Designing a ServiceNow-based application.
- Creating modules for data entry, expense categorization, and analytics.
- Implementing automated workflows for monthly expense summaries.
- Providing dashboards for expense visualization and comparison.

Out of scope:

- Integration with external banking APIs (for this version).
- Mobile app development (optional for future phase).

Resources Required:

Software:

- ServiceNow Platform
- Excel or Google Sheets (for initial data testing)
- Reporting & Analytics Module in ServiceNow

Hardware:

- Laptop/Desktop with internet connection

Human Resources:

- Project Lead / Developer
- ServiceNow Administrator
- Tester / QA Engineer

Expected Deliverables:

- A fully functional ServiceNow application for family expense tracking.
- Automated calculation and reporting features.
- User guide and documentation.

Success Criteria:

- Accurate calculation and categorization of expenses.
- Real-time dashboard functioning properly.
- Positive feedback from test users on usability and performance.

Project Design Phase :

Overview:

- The design phase focuses on creating the system architecture, workflow, and user interface for the Family Expense Management System. The goal is to ensure smooth data flow, automation of calculations, and an intuitive user experience within the ServiceNow platform.

System Architecture:

The system will be designed using the modular approach of ServiceNow, consisting of:

- **User Interface (UI):** Forms and dashboards for users to input and view expense data.
- **Application Core:** Business rules, workflows, and scripts to automate expense calculations and report generation.
- **Database Layer:** ServiceNow tables to store expense records, user profiles, and category information.

Architecture Flow:

1. User enters daily expenses through a ServiceNow form.
2. Data is stored in a custom expense table.
3. Automated workflow calculates totals and categorizes expenses.
4. Reports and dashboards display visual summaries of monthly or weekly spending.

Module Design:

Module	**Description**	**Features**
User Management	Handles user login and family member profiles	Secure access, user roles
Expense Entry	Allows users to input expenses	Fields for date, amount, category, and notes
Category Management	Manages expense types (Food, Rent, Utilities, etc.)	Add/Edit/Delete categories
Expense Calculation	Automates total expense computation	Auto sum, category-wise split
Reports & Dashboard	Visualizes data for analysis	Charts, monthly summaries, spending trends

Workflow Design:

Step 1: User logs in to the ServiceNow application.
Step 2: Enters expense details in a simple form.
Step 3: System validates data and stores it in the database.
Step 4: Automated script calculates total and category-wise spending.
Step 5: Dashboard displays real-time financial summaries and reports.

User Interface Design:

- **Home Page:** Overview of total expenses, remaining budget, and quick links.
- **Expense Entry Form:** Clean form with dropdowns for category and date selection.
- **Dashboard:** Graphical representation of expenses using charts (bar, pie).
- **Reports Section:** Downloadable summaries for monthly or yearly spending.

Data Design:

Tables Used:

- **Family_Expense:** Stores expense details (date, category, amount, remarks).
- **Expense_Category:** Stores predefined categories.
- **User_Profile:** Stores user or family member data.

Key Fields:

- Expense ID, User ID, Date, Category, Amount, Description.

Security and Access Control:

- Role-based access to ensure data privacy.
- Only authenticated users can view or modify expense records.
- Admins can generate reports and manage categories.

Expected Output:

- User-friendly interface for daily expense entry.
- Automated calculation of total and category-wise spending.
- Interactive dashboard showing real-time financial analytics.

Requirement Analysis:

Overview:

The purpose of this phase is to identify and analyze all the functional and non-functional requirements necessary to develop the Family Expense Management System on the ServiceNow platform. This system will help users efficiently record, manage, and analyze their family expenses through automation and easy-to-use dashboards.

Objectives:

- To define clear requirements for building the expense tracking system.
- To ensure the solution meets the needs of end users (families).
- To establish technical and operational feasibility within the ServiceNow environment.

Functional Requirements:

These requirements describe the features and behaviors the system must perform.

No.	Requirement	Description
1	User Authentication	Users must log in securely to access their expense data.
2	Expense Entry	Users can enter daily, weekly, or monthly expenses with fields such as category, date, amount, and notes.
3	Category Management	Users can add, edit, or delete expense categories (e.g., Food, Rent, Travel, Utilities).
4	Expense Calculation	The system automatically sums up total expenses and categorizes them.
5	Report Generation	Generate monthly and yearly reports summarizing spending by category.
6	Dashboard Visualization	Real-time dashboards with pie charts, bar graphs, and totals for quick analysis.
7	Data Storage	Expense data is securely stored in ServiceNow tables.
8	Budget Limit Alert (Optional)	Alerts the user when expenses exceed a predefined budget.
9	User Roles	Different roles such as Admin (manage all) and Member (own data only).

Non-Functional Requirements:

These describe system qualities and performance expectations.

No.	Requirement	Description
1	Performance	The system should handle multiple users and calculate expenses instantly.
2	Security	Data must be stored securely with restricted access using ServiceNow's ACL (Access Control Lists).
3	Usability	Interface should be simple and easy to use for non-technical users.
4	Reliability	The system should function accurately without data loss or duplication.
5	Scalability	Should allow adding more users or categories without affecting performance.
6	Maintainability	Easy to update workflows and UI through ServiceNow configuration.
7	Availability	System should be accessible anytime via the cloud-based ServiceNow platform.

Hardware & Software Requirements:

Hardware:

- Laptop or Desktop computer
- Stable Internet connection

Software:

- ServiceNow Platform (Developer Instance)
- Web Browser (Chrome, Edge, or Firefox)
- Microsoft Excel (optional for data testing)

User Requirements:

- Users should be able to input expenses easily.
- Users should be able to view total and category-wise expenses.
- Users should be able to generate and download reports.
- Admin users should be able to manage categories and users.

System Requirements:

- Custom tables for storing expenses and categories.
- UI Pages or Forms for data input.
- Business Rules and Workflows for automation.
- Reports and Dashboards for visualization.

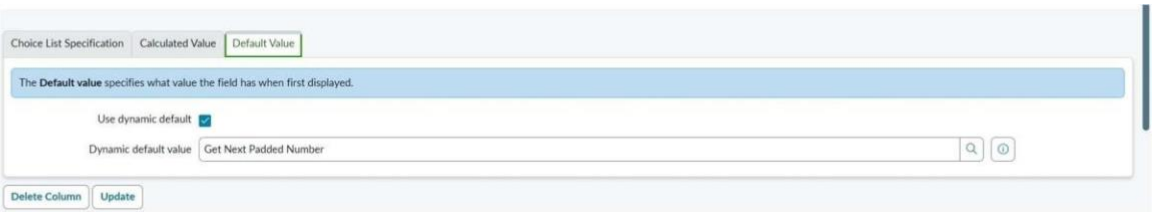
Constraints:

- Limited to ServiceNow's inbuilt functionalities (no external database integration).
- Requires internet access to use the ServiceNow instance.
- Users must have valid login credentials.

Expected Outcome:

The system will provide a centralized and automated solution for tracking family expenses, reducing manual errors, and helping users make better financial decisions through ServiceNow's workflow and reporting tools.

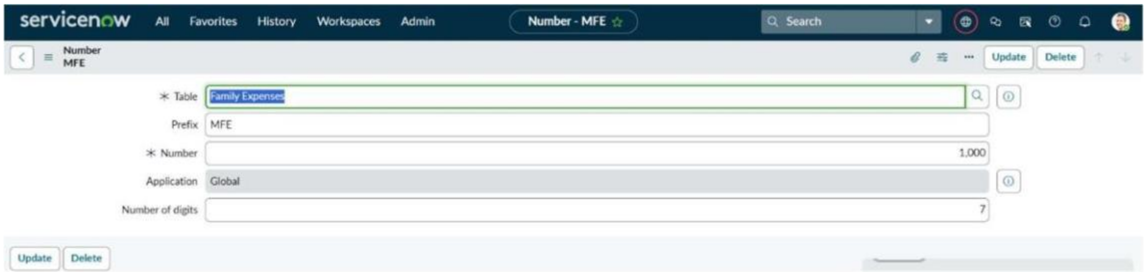
Performance Testing:



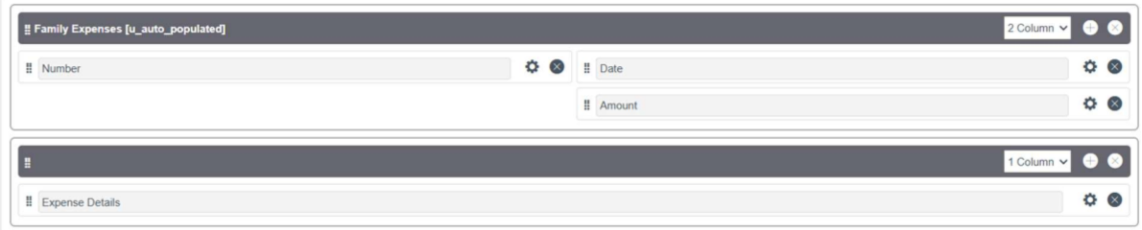
The **Default value** specifies what value the field has when first displayed.

Use dynamic default ☑

Dynamic default value: Get Next Padded Number



* Table	Family Expenses
Prefix	MFE
* Number	1,000
Application	Global
Number of digits	7

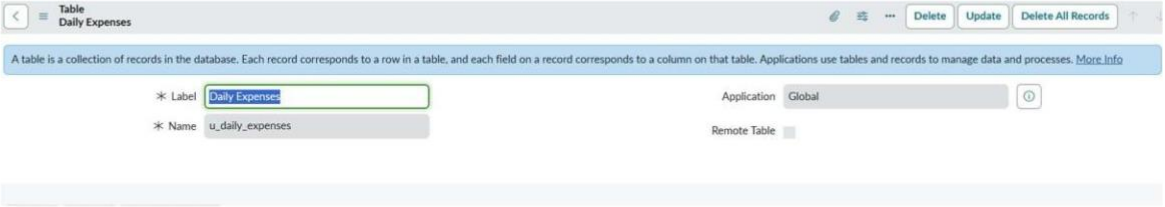


Number

Date

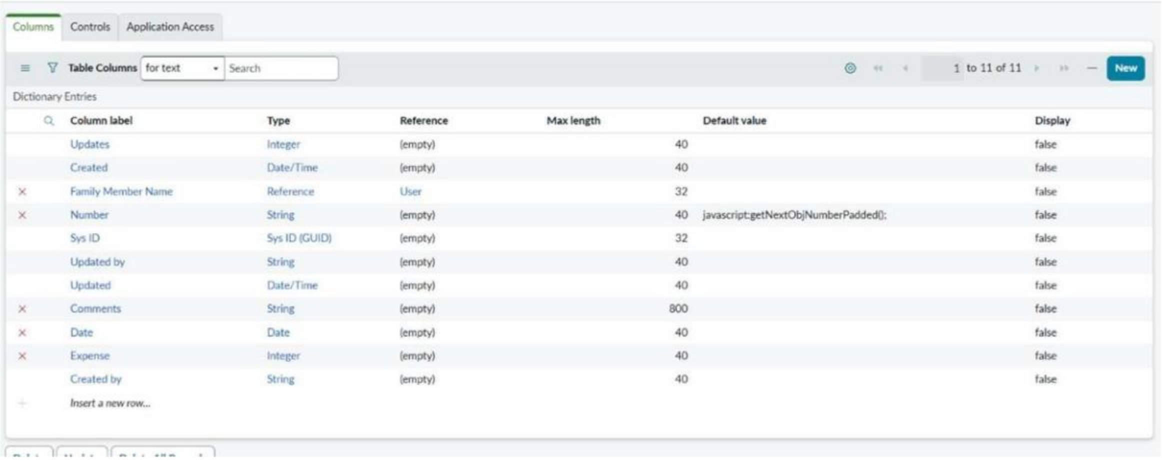
Amount

Expense Details



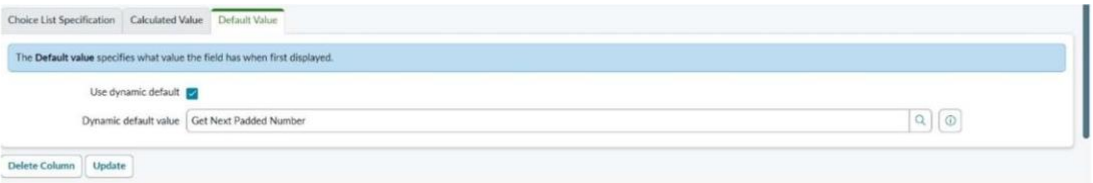
A table is a collection of records in the database. Each record corresponds to a row in a table, and each field on a record corresponds to a column on that table. Applications use tables and records to manage data and processes. More Info

* Label	Daily Expenses	Application	Global
* Name	u_daily_expenses	Remote Table	□



Dictionary Entries

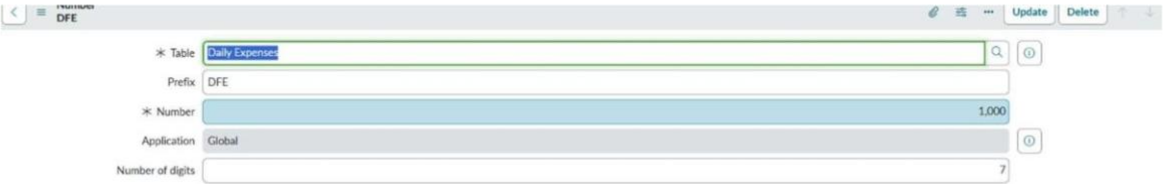
	Column label	Type	Reference	Max length	Default value	Display
	Updates	Integer	(empty)	40		false
	Created	Date/Time	(empty)	40		false
×	Family Member Name	Reference	User	32		false
×	Number	String	(empty)	40	javascript:getNextObjNumberPadded();	false
	Sys ID	Sys ID (GUID)	(empty)	32		false
	Updated by	String	(empty)	40		false
	Updated	Date/Time	(empty)	40		false
×	Comments	String	(empty)	800		false
×	Date	Date	(empty)	40		false
×	Expense	Integer	(empty)	40		false
	Created by	String	(empty)	40		false
+	*Insert a new row...*					



The **Default value** specifies what value the field has when first displayed.

Use dynamic default ☑

Dynamic default value: Get Next Padded Number



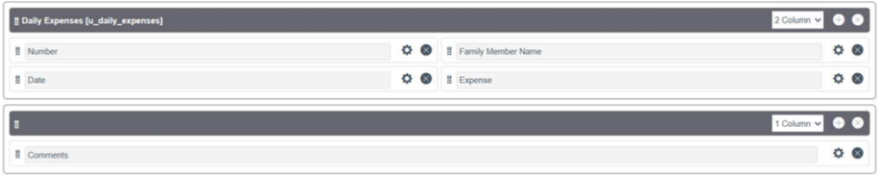
∗ Table: Daily Expenses

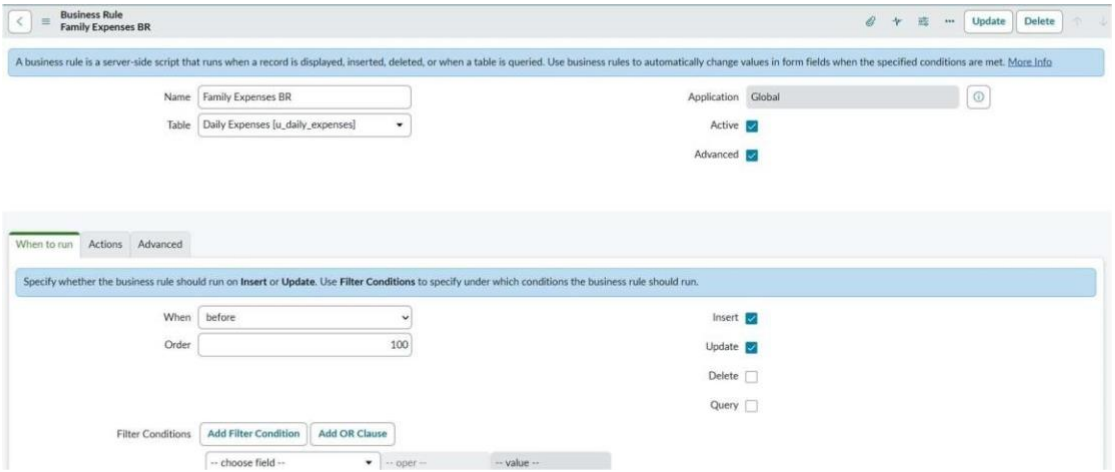
Prefix: DFE

∗ Number: 1,000

Application: Global

Number of digits: 7





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

Table: Daily Expenses [u_daily_expenses]

Application: Global

Active ☑

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.

When: before

Order: 100

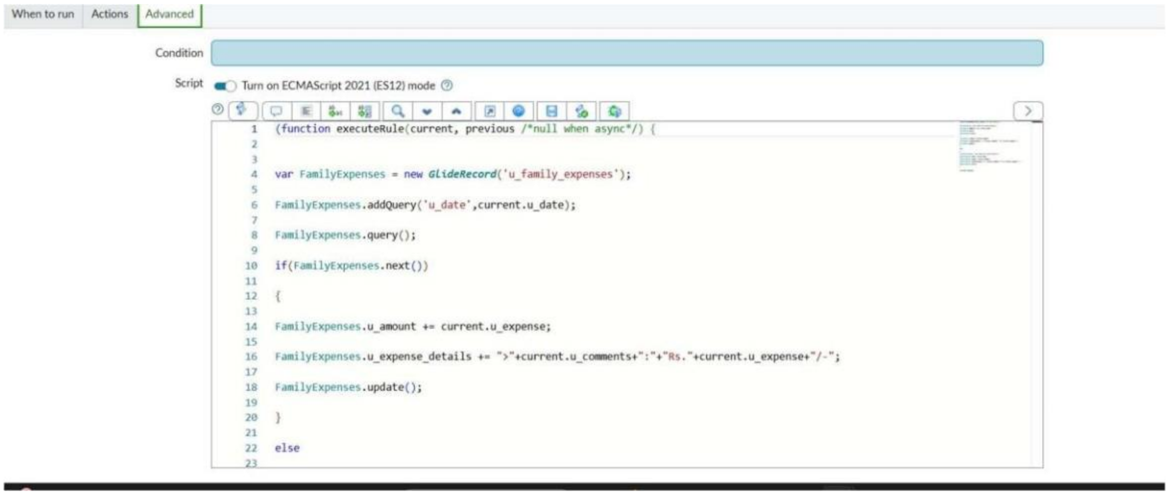
Insert ☑

Update ☑

Delete ☐

Query ☐

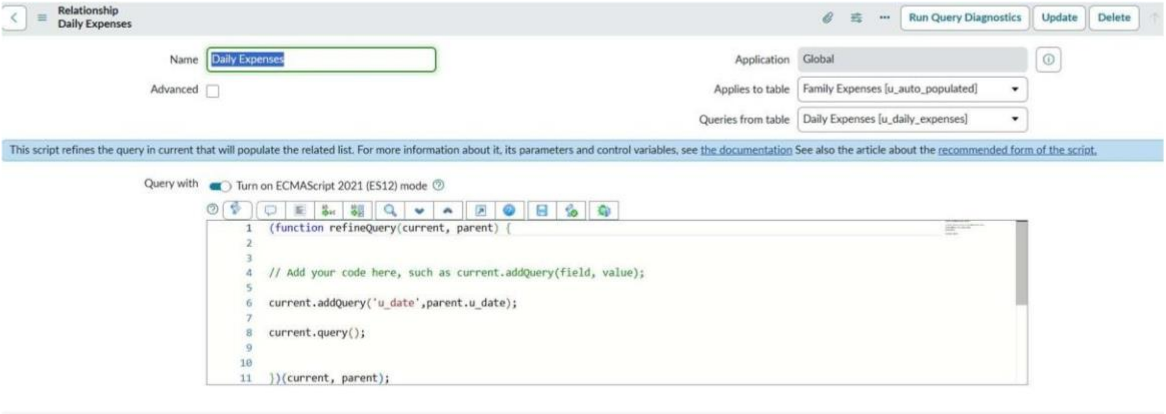
Filter Conditions



Condition

Script: Turn on ECMAScript 2021 (ES12) mode

```
(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
```



Name: Daily Expenses

Advanced ☐

Application: Global

Applies to table: Family Expenses [u_auto_populated]

Queries from table: Daily Expenses [u_daily_expenses]

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.

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);
```