

### 3. PROJECT DESIGN PHASE

#### System Overview

The Family Expense Management System is developed on the ServiceNow platform with the goal of simplifying the process of recording and managing family expenses.

The system is built around two key modules:

- Family Expenses Table - Stores summarized or overall expense information.
- Daily Expenses Table - Records detailed daily transactions and links them to the family's main expense record.

Both tables are linked through a relationship configuration to ensure proper data consistency and association between daily entries and the main expense category.

#### Workflow Steps

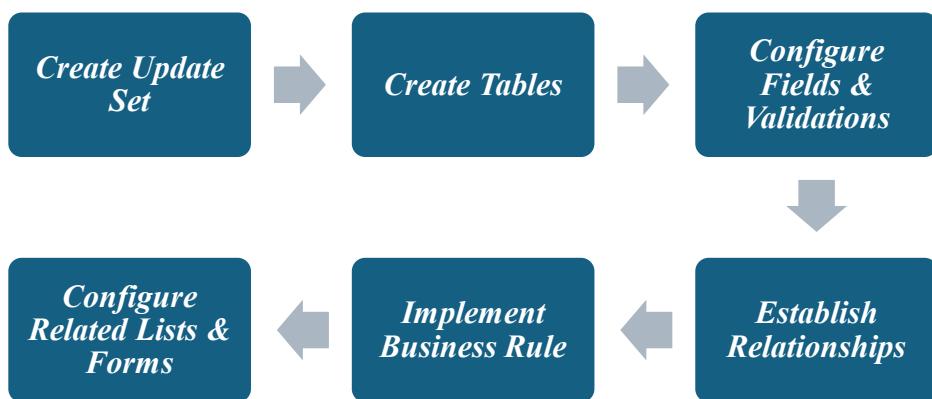


Fig 1: Workflow of Family Expense Management System in ServiceNow

#### System Architecture

Layer	Description
Frontend (User Interaction)	ServiceNow Forms allows users to input and view data through custom forms.
Database Layer	Family Expenses & Daily Expenses tables for structured data storage.

Logic Layer	Business Rules and Number Maintenance for automation and data validation.
Reporting Layer	Dashboards and Analytics for expense tracking and visualization.

## Table Design Summary

Table Name	Columns	Description
Family Expenses	Number, Date, Amount, Expense Details	Stores high-level expense information for each category.
Daily Expenses	Number, Date, Amount, Expense Details, Comments	Records detailed daily expenses linked to the main family expense record.

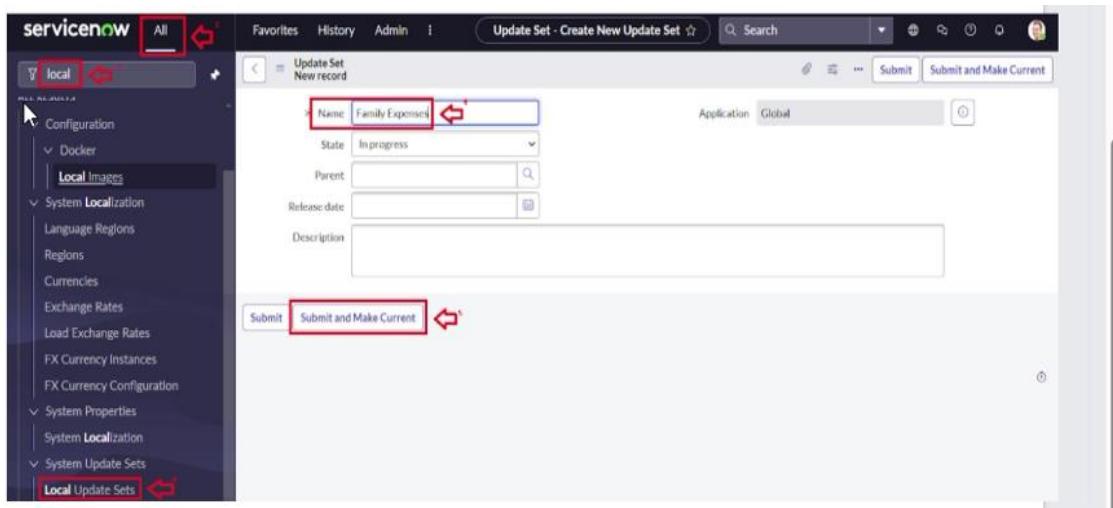
## Design & Configuration Activities

### Activity 1:

1. Open Service now
2. Go to All > search for Local Update Set > Click on New



3. Enter the Details as:
- Name: Family Expenses
4. And then click on Submit and make current



### Activity 2:

1. Open Service now
2. Go to all > search for Tables > Click on New
3. Enter the Details:
  - Label: Family Expenses
  - Name: Auto-Populated
  - New menu name: Family Expenditure
4. Click on save

### Activity 3:

1. Open Service now
2. Near Columns Double-click near insert a new row
  - Column label: Number
  - Type: String
3. Double-click on insert a new row again
  - Column label: Date
  - Type: Date
4. Double click on insert a new row again
  - Column label: Amount
  - Type: Integer
5. Double click on insert a new row again
  - Column label: Expense Details
  - Type: String

Max length:800

Column label	Type	Reference	Max length	Default value	Display
Number	String				False
Date	Date				False
Amount	Integer				False
Expense Details	String		800		False

6.Click on save

#### **Activity 4:**

- 1.Open Service now
- 2.Double click on the Number field/Column
- 3.Go down and double click on Advanced view
- 4.In Default Value:
  - Use dynamic default:check the box
  - Dynamic default value:Get Next Padded Number
- 5.click on update

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

Use dynamic default

Dynamic default value: Get Next Padded Number

6.Go to all >> search for Number Maintenance >> select number maintenance

7.Click on new

8.Enter the details:

Table:Family Expenses

Prefix:MFE

Table: Family Expenses

Prefix: MFE

\* Number: 1,000

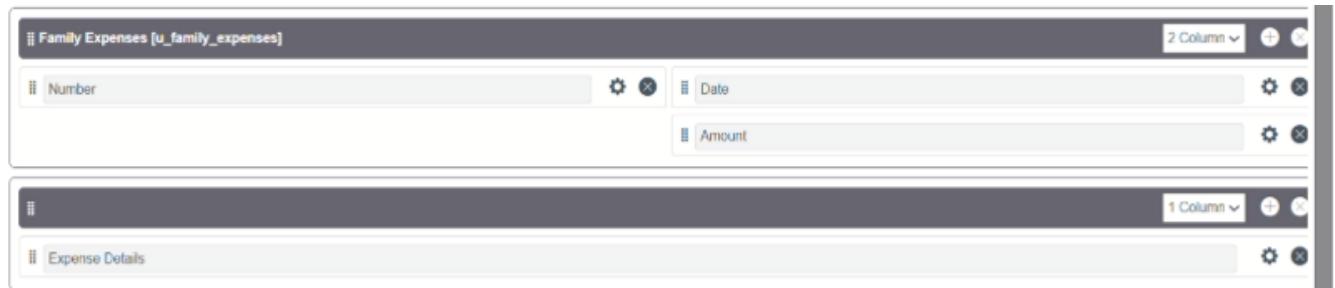
Application: Global

Number of digits: 7

9.Click on Submit

### **Activity 5:**

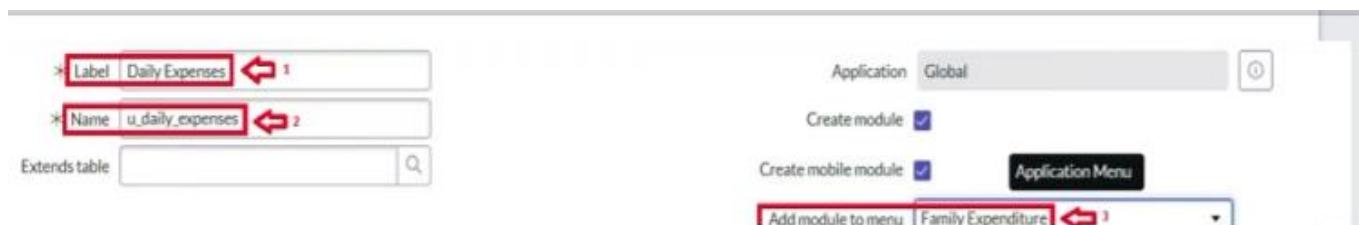
- 1.Open Service now
- 2.Go to all >> search for Family Expenses >> Open family Expenses
- 3.Click on new
- 4.Go to header and click there >> click on configure >> Select Form Design
- 5.Drag Drop the form as requirement



- 6.Make Number Read-Only Field by Clicking
- 7.Make Date, Amount Mandatory field by Clicking
- 8.Click on save

### **Activity 6:**

- 1.Open Service now
- 2.Go to all > search for Tables > Click on New
- 3.Enter the Details:
  - Label:Daily Expenses
  - Name:Auto-Populated
  - Add Module to menu :Family Expenditure



- 4.Click on save

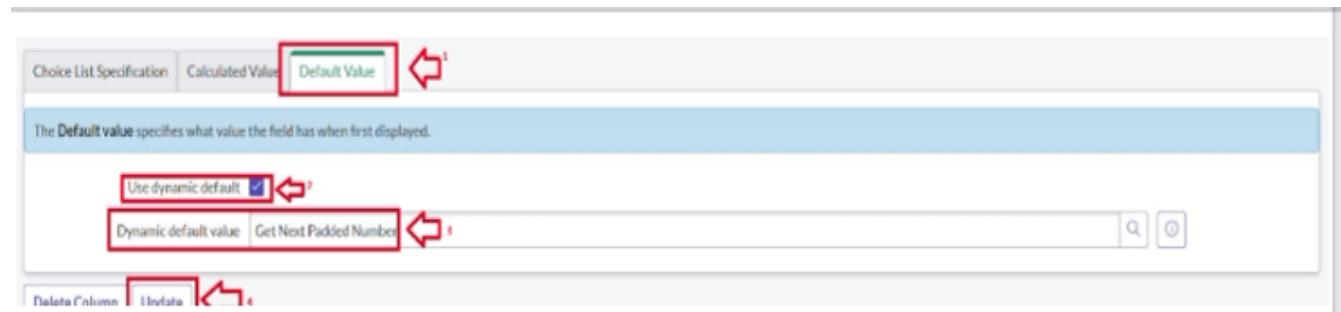
### **Activity 7:**

- 1.Open Service now
- 2.Near Columns Double Click near insert a new row
  - Column label:Number
  - Type:String
- 3.Double click on insert a new row again

- Column label:Date  
Type:Date
- 4.Double click on insert a new row again  
Column label:Amount  
Type:Integer
- 5.Double click on insert a new row again  
Column label:Expense Details  
Type:String  
Max length:800
- 6.Double click on insert a new row again  
Column label:Comments  
Type:String  
Max length:800
- 7.Click on save

#### **Activity 8:**

- 1.Open Service now
- 2.Double click on the Number field/Column
- 3.Go down and double click on Advanced view
- 4.In Default Value:  
    Use dynamic default:check the box  
    Dynamic default value:Get Next Padded Number
- 5.Click on update



- 6.Go to all >> search for Number Maintenance >> select number maintenance
- 7.Click on new
- 8.Enter the details:  
    Table:Daily Expenses  
    Prefix:MFE

\* Table Daily Expenses 1

\* Prefix DFB 2

\* Number 1,000

Application Global

Number of digits 7

**Submit** 3

9.Click on submit

### **Activity 9:**

- 1.Open Service now
- 2.Go to all >> search for Daily Expenses >> Open Daily Expenses
- 3.Click on new
- 4.Go to header and click there>>click on configure>>Select Form Design
- 5.Drag Drop the form as requirement

2 Column	
Number	Family Member Name
Date	Expense

1 Column	
Comments	

- 6.Make Number Read-Only Field by Clicking
- 7.Make Date,Family Member Name Mandatory field by Clicking
- 8.Click on save

### **Activity 10:**

- 1.Open Service now
- 2.Go to all >> search for relationships >> open relationships
- 3.Click on new
- 4.Enter the details:  
 Name:Daily Expenses  
 Applies to table:select Family Expenses  
 Daily Expenses:select Daily Expenses
- 5.Click save

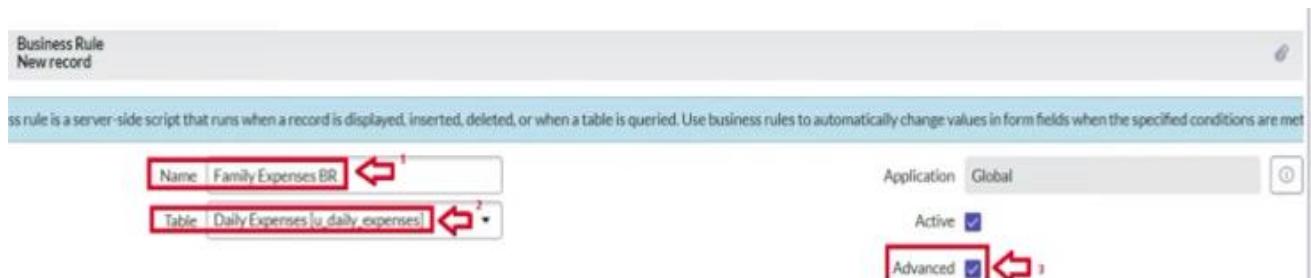
### **Activity 11:**

- 1.Open Service now
- 2.Go to all >> search for Family Expenses >> Open FamilyExpenses
- 3.Click on new
- 4.Go to header and click there >> click on configure >> select Related Lists
- 5.Add Daily Expenses to selected area
- 6.Click on save

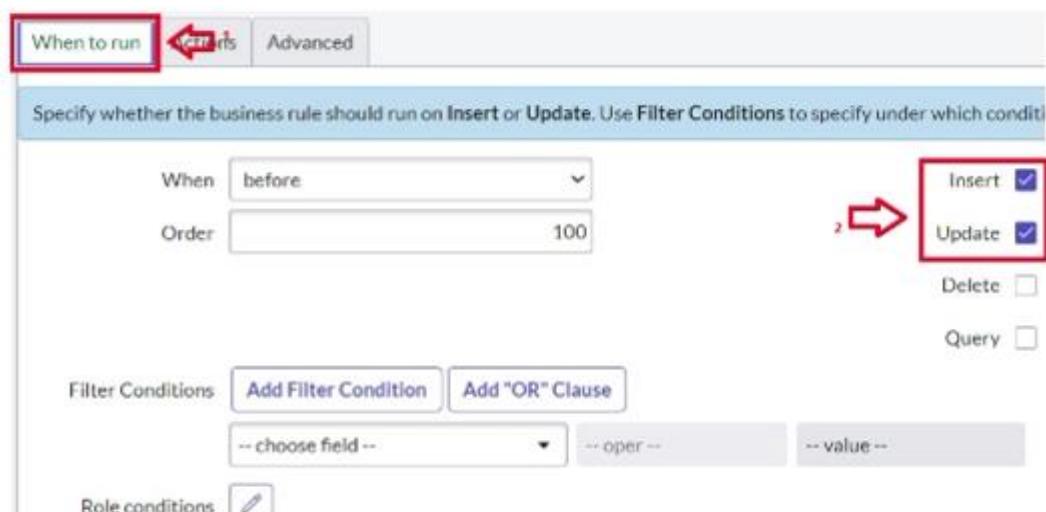


### **Activity 12:**

- 1.Open Service Now
- 2.Go to all >> search for Business Rules
- 3.Under system definition select business rules then click new
- 4.Enter the Details:
  - Name:Family Expenses BR
  - Table:Select Daily Expenses
  - Check Advanced



- 5.In when the run check insert and update



## 6.In Advance (we write the code)

```

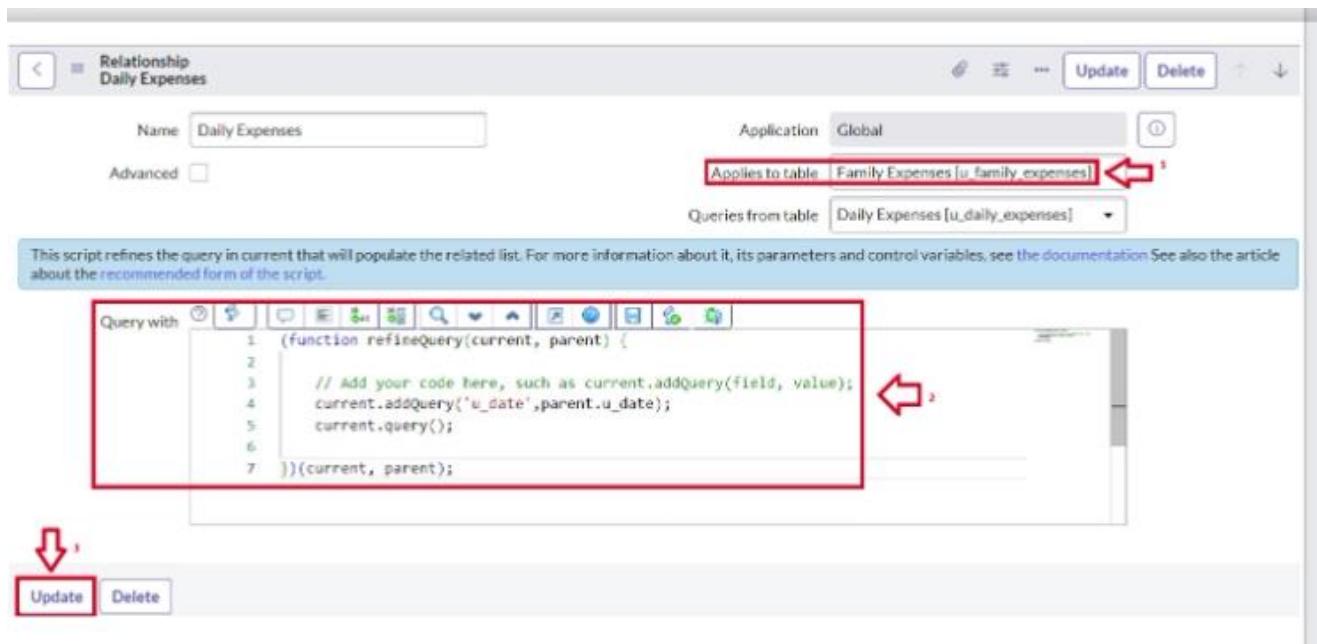
1  (function executeRule(current, previous /*null when async*/) {
2
3      var FamilyExpenses = new GlideRecord('u_family_expenses');
4      FamilyExpenses.addQuery('u_date',current.u_date);
5      FamilyExpenses.query();
6      if(FamilyExpenses.next())
7      {
8          FamilyExpenses.u_amount +=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

```

## 7.Click on save

### Activity 13:

- 1.Open Service now
- 2.Go to all >> search for relationships >> open relationships
- 3.In open Daily Expenses relationship  
For Applies to table: select Family Expenses  
In Query with: write the below Query
- 4.Click on update



## System Features

- Auto-generated expense numbers for both tables.
- Mandatory validation for Date and Amount fields.
- Automated linking between Family and Daily Expenses records.
- Easy-to-use ServiceNow form interface.
- Configurable related lists and table relationships.

## Design Outcome

The Family Expense Management System was successfully designed with a functional data model, logical workflow, and automated operations.

It provides:

- Centralized expense tracking
- Seamless relationship between modules
- Real-time data validation and automation

## Conclusion

The Family Expense Management System built on ServiceNow offers a centralized and automated solution for tracking household expenses. It enables users to record, categorize, and analyze spending in real time, promoting better financial planning and transparency. This project demonstrates how digital automation can simplify personal finance management, making it more accurate, efficient, and accessible.