Calculating Family Expenses Using ServiceNow Final Project Report

1. INTRODUCTION:

1.1 Project Overview:

The "Calculating Family Expenses Using ServiceNow" project focuses on building a unified, automated system to manage household expenses. It allows users to record daily and family-level expenses, link them to a central budget, and track spending in real time. Leveraging ServiceNow's low-code platform with custom tables, forms, business rules, and automation, the solution replaces manual bookkeeping and demonstrates how ServiceNow can solve real-life problems beyond IT workflows.

1.2 Purpose:

The purpose of this project is to provide families with an intuitive, reliable, and scalable tool to manage their household expenses. The solution addresses common challenges such as disorganized tracking, missed entries, and difficulty staying within budget. By linking daily expenses to family-level records, generating auto-numbered entries, and offering automated alerts when budgets are exceeded, the system helps users maintain financial discipline. Furthermore, the tool generates categorized reports that give families clear visibility into their spending patterns, supporting better decision-making and long-term financial planning. This project also showcases the versatility of the ServiceNow platform in addressing non-IT use cases through innovative configurations and automation.

2. IDEATION PHASE:

2.1 Problem Statement:

Families often struggle to organize daily and household-level expenses, leading to poor visibility into spending patterns and risks of overspending. Existing methods are manual, fragmented, and error-prone.

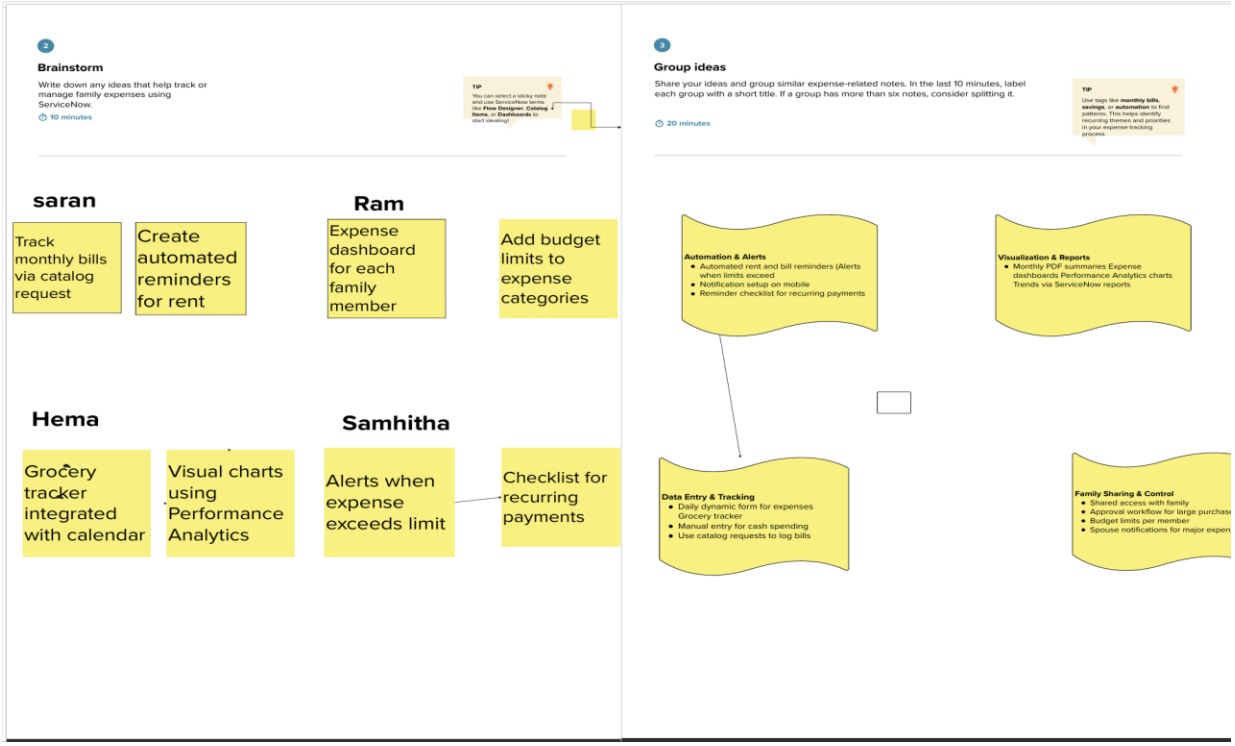
2.2 Empathy Map Canvas:

- **Says:** "I want an easy way to track my expenses." "I need to know when we overspend."
- **Thinks:** "Are we staying within budget?" "Did I miss recording any expenses?"
- **Does:** Logs expenses manually or checks bills/receipts occasionally.

- **Feels:** Anxious about overspending and frustrated with disorganized tracking.

2.3 Brainstorming:

The team explored ideas such as using custom ServiceNow tables for Family and Daily Expenses, adding related lists, configuring auto-numbering, setting up business rules for alerts, and generating reports. The focus was on creating a system that simplifies tracking and provides automated support.



3. REQUIREMENT ANALYSIS:

3.1 Customer Journey Map:

Users log daily expenses, monitor budget status, and review reports. The system provides alerts when spending approaches or exceeds set limits.

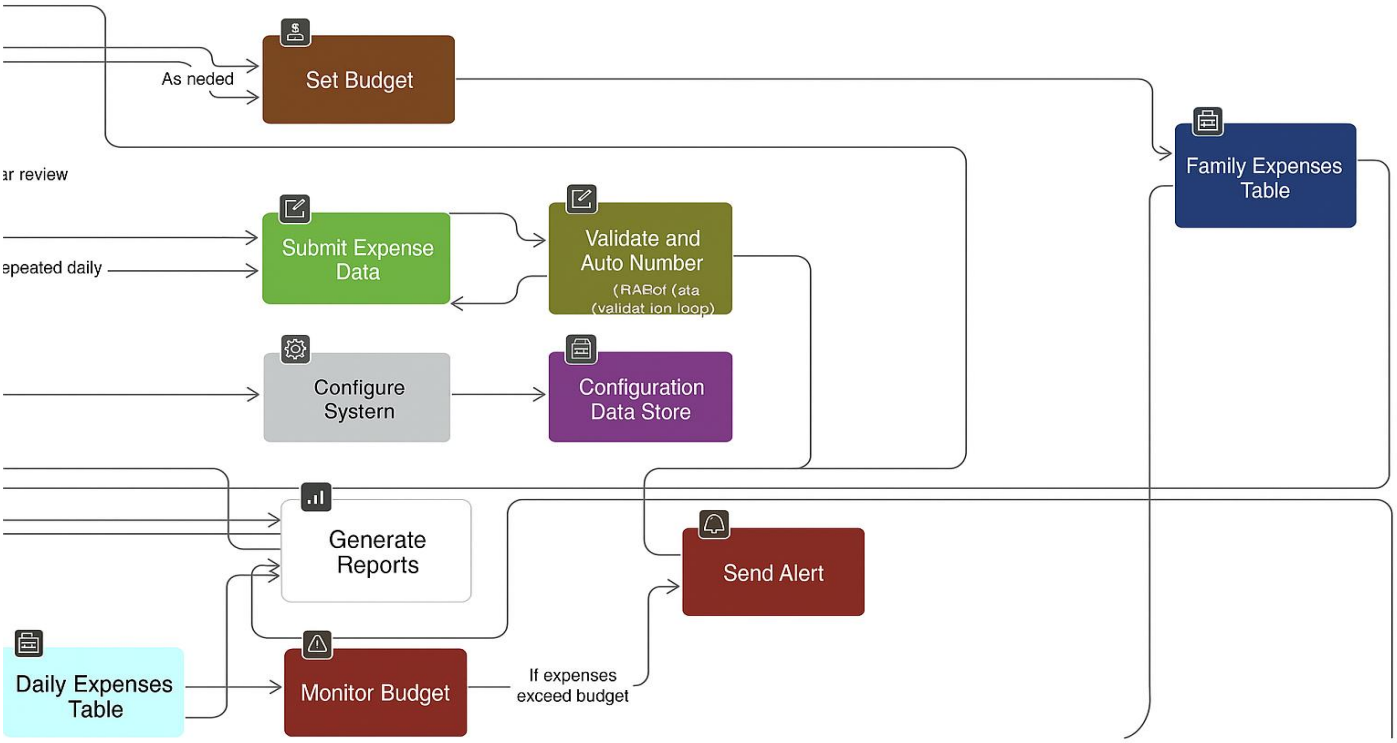
3.2 Solution Requirement:

- Family and Daily Expenses tables
- Auto-numbering with prefixes (MFE, DFE)
- Related lists between Family and Daily Expenses
- Business rules for automation

- Budget alerts
- Reporting capability

3.3 Data Flow Diagram:

The DFD shows data flowing from user forms → validation → storage in tables → automation triggers → reports/alerts generation.



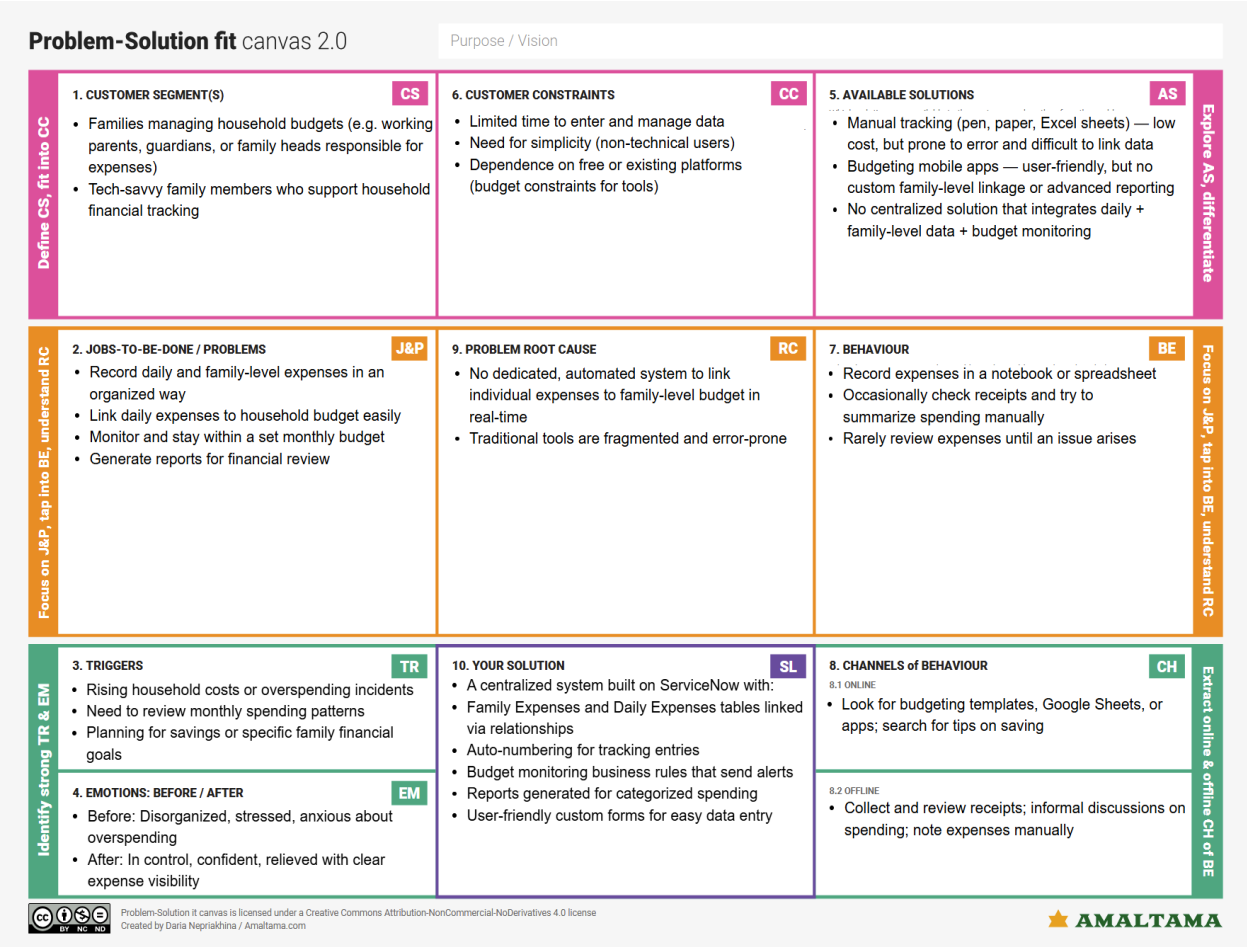
3.4 Technology Stack:

- ServiceNow custom tables and forms
- Glide API, Business Rules, UI Policies
- ServiceNow Notification Engine
- MySQL backend (ServiceNow-managed)
- ServiceNow REST APIs (optional future integrations)

4. PROJECT DESIGN:

4.1 Problem–Solution Fit

The system solves the problem of disorganized expense tracking by providing a centralized platform linking daily expenses to budgets, automating alerts, and simplifying reporting.



4.2 Proposed Solution:

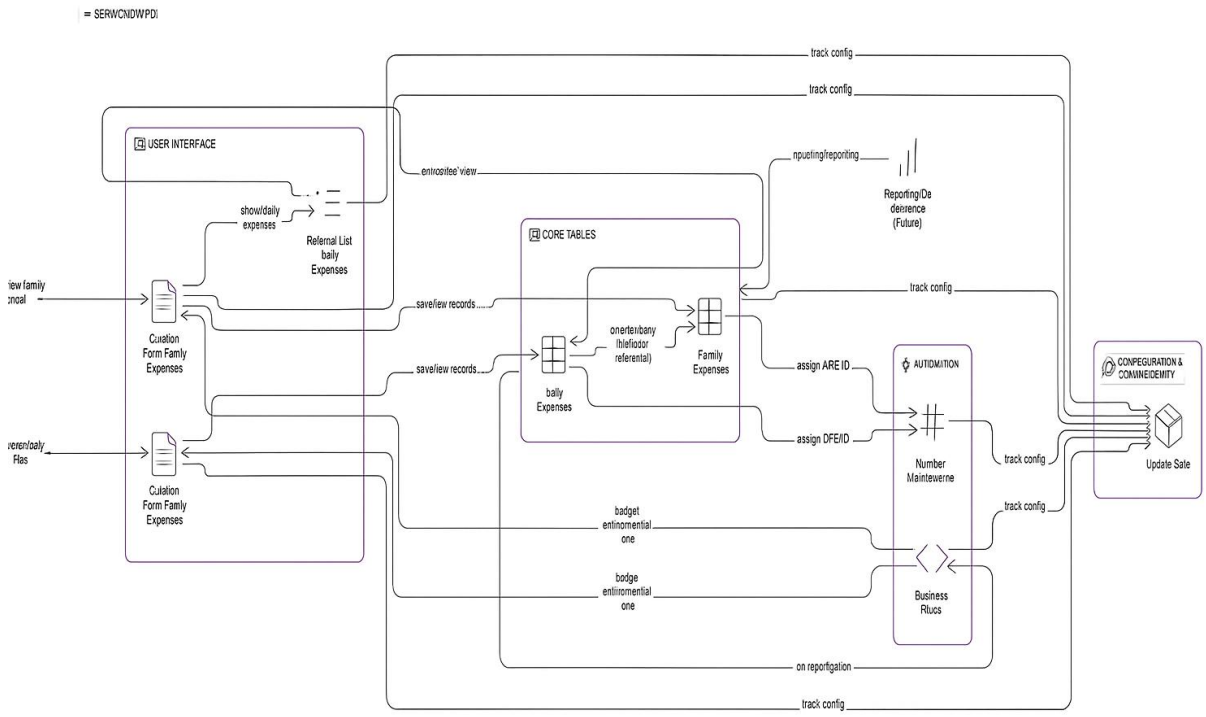
A ServiceNow-based tool with:

- Custom tables and forms
- Relationships and related lists
- Business rules for automation
- Budget alerts
- Categorized reports

4.3 Solution Architecture:

The architecture includes:

- Data Layer: Family and Daily Expenses tables
- Logic Layer: Business rules, number maintenance
- UI Layer: Custom forms, related lists
- Configuration Layer: Update sets



5. PROJECT PLANNING & SCHEDULING:

5.1 Project Planning:

The project was completed over 3 sprints:

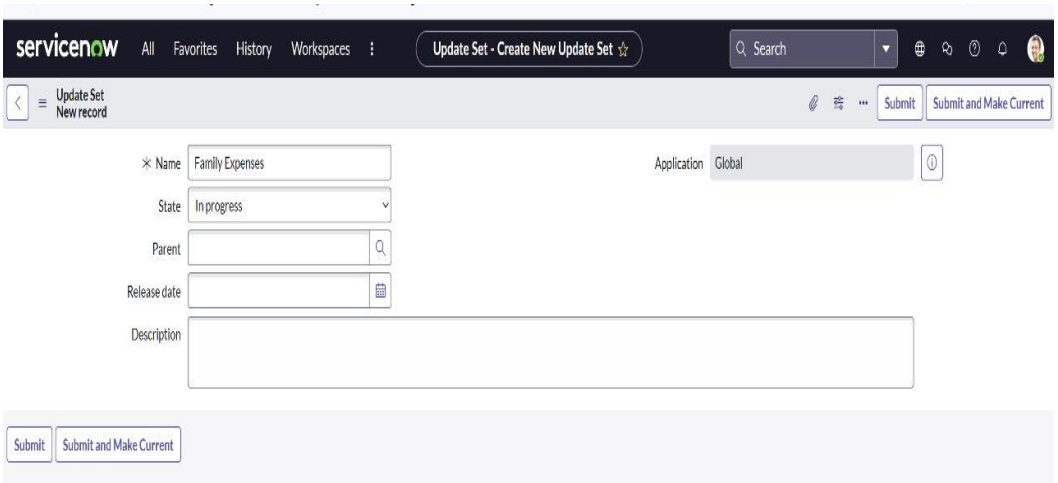
- **Sprint 1:** Instance setup, update set, table creation (9 points)
- **Sprint 2:** Relationships, related list, business rules (5 points)
- **Sprint 3:** Budget alerts, reports (6 points)

Velocity: 20 story points / 3 sprints = ~6.67 points per sprint.

The Project was completed as the following milestones covering 3 sprints

The team executed these milestones:

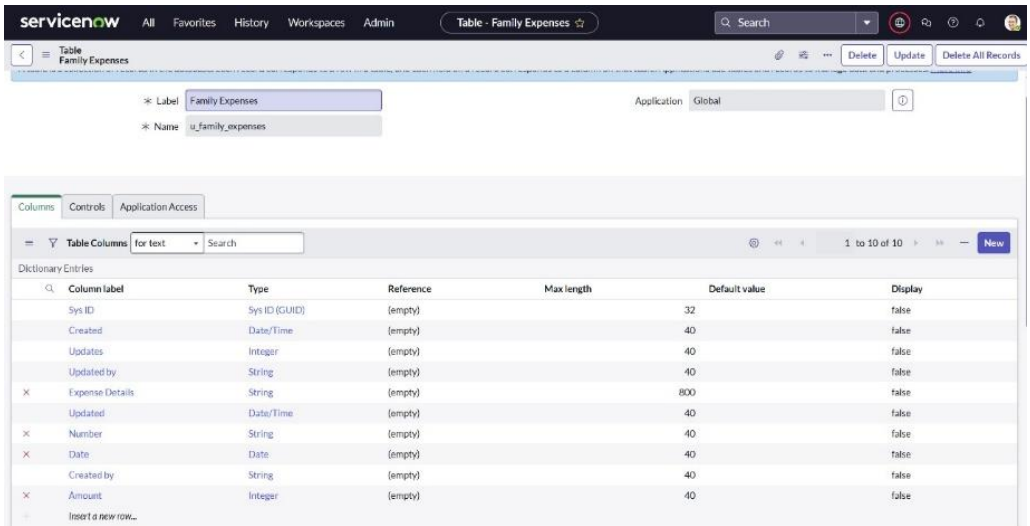
1. ServiceNow Instance Setup
    - Signed up at developer.servicenow.com and requested a Personal Developer Instance (PDI)
    - Filled necessary details; received instance access credentials via email
    - Logged in and prepared the instance for development
2. Creation of New Update Set
    - Navigated to Local Update Sets and created a new update set named *Family Expenses*
    - Submitted and made the update set current to track configurations



3. Creation of Family Expenses Table
    - Created the Family Expenses table under *Family Expenditure* menu
    - Configured number field for auto-numbering (dynamic default: Get Next Padded Number)
    - Set up Number Maintenance with prefix MFE
    - Customized form layout using Form Designer

Family Expenses Table Fields

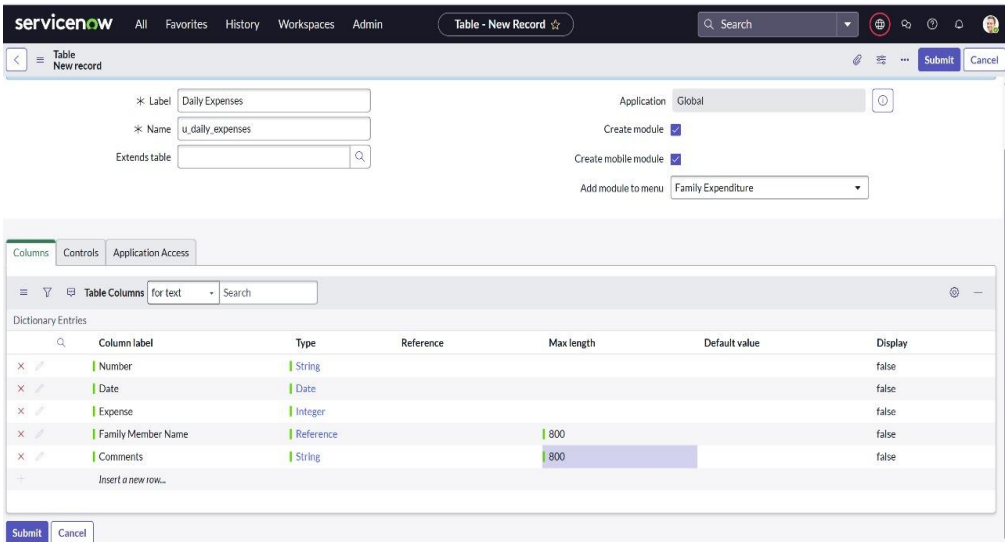
Field Name	Type	
Number	String	Auto populate Number with Prefix MFE
Date	Date	
Amount	Integer	
Expense Details	String	Max Length 800



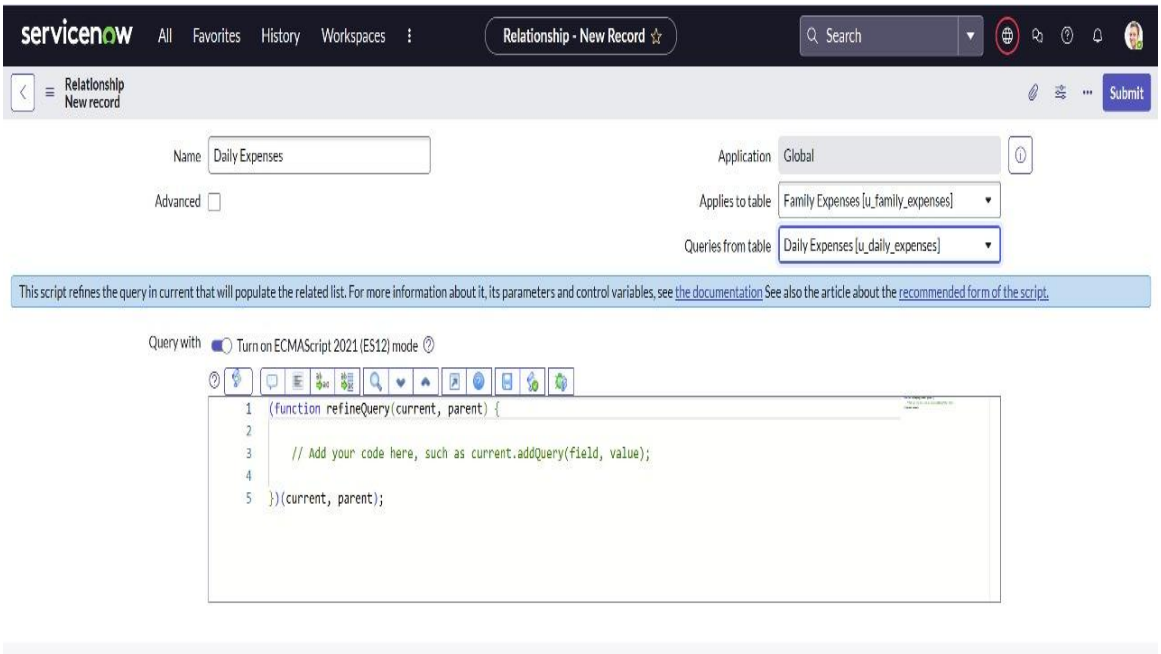
4. Creation of Daily Expenses Table
    - Created the Daily Expenses table under *Family Expenditure* menu
    - Configured number field for auto-numbering (dynamic default: Get Next Padded Number)
    - Set up Number Maintenance with prefix DFE
    - Customized form layout using Form Designer

Daily Expenses Table Fields

Field Name	Type	
Number	String	Auto populate Number with Prefix DFE
Family Member Name	Reference	Sys_User
Date	Date	
Expense	Integer	
Comments	String	Max Length 800

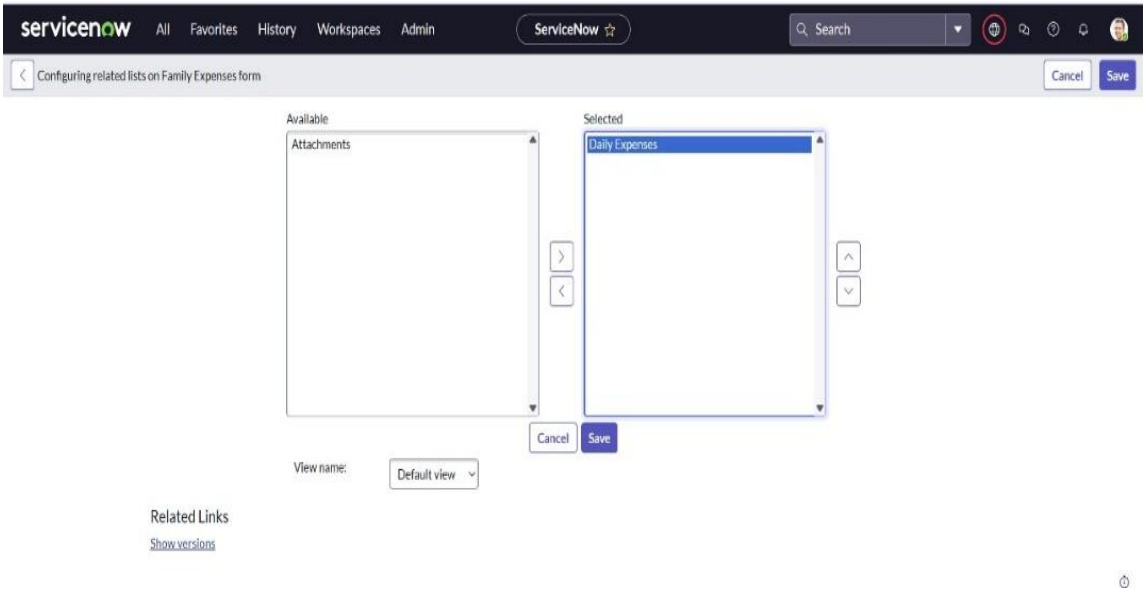


5. Creation of Relationship
    - Created relationship where Daily Expenses records are linked to Family Expenses
    - Configured queries so related list data aligns with the parent Family Expenses record



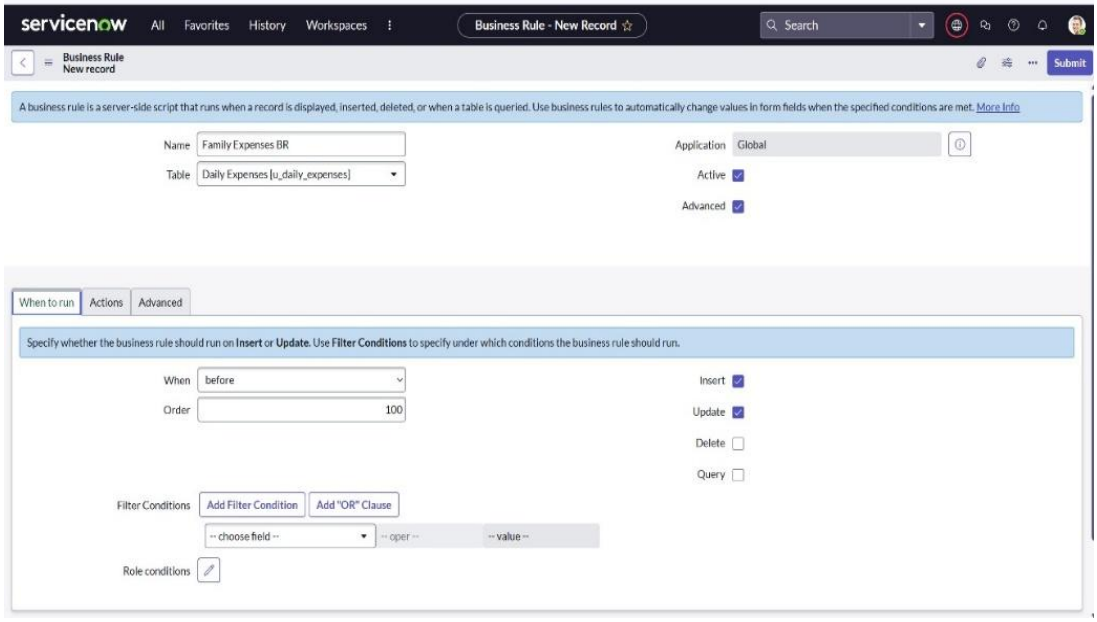
6. Related List Configuration

   - Added Daily Expenses as a related list within the Family Expenses form using the Configure Related Lists feature

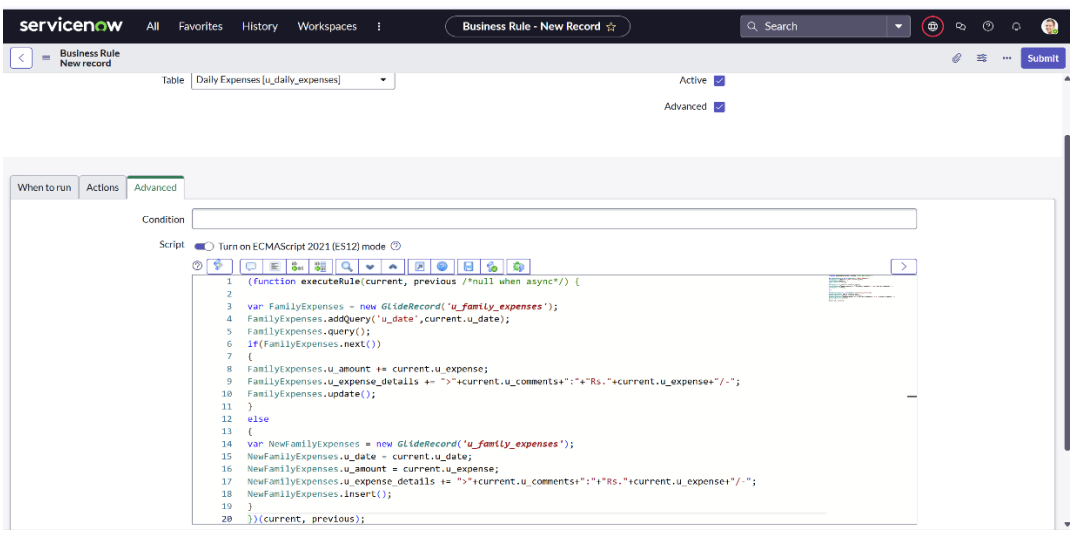


7. Business Rule Creation

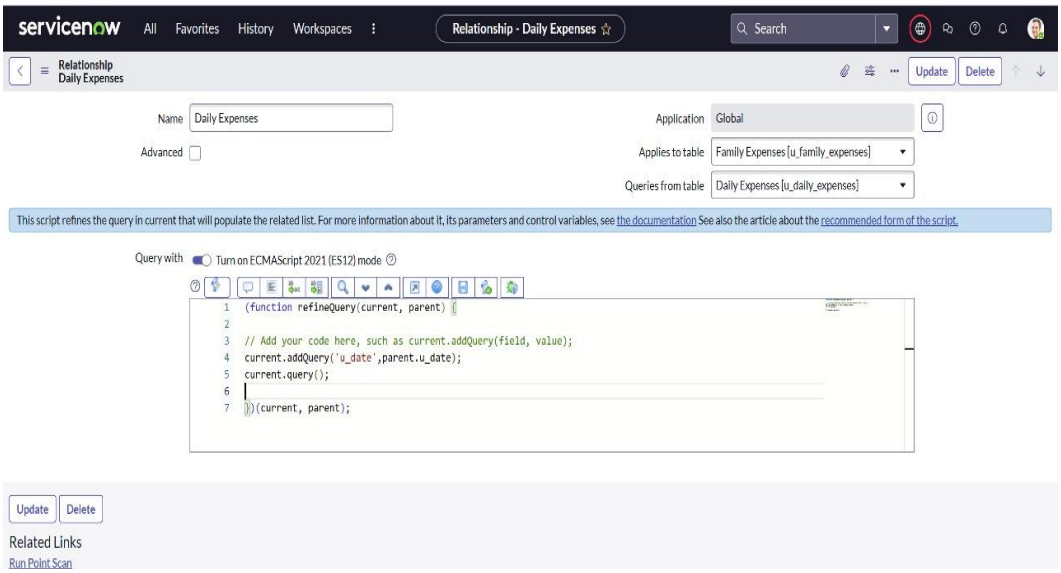
   - Created a business rule named *Family Expenses BR* on the Daily Expenses table
   - Enabled advanced options
   - Configured to trigger on Insert and Update actions



- Implemented the logic to manage data consistency or automate actions (actual script written as part of development)



8. Final Relationship Configuration
    - Opened the previously created Daily Expenses relationship
    - Verified Applies to Table is set to Family Expenses
    - Entered and saved any dynamic query script to refine related list output



6. FUNCTIONAL AND PERFORMANCE TESTING

6.1 Performance Testing

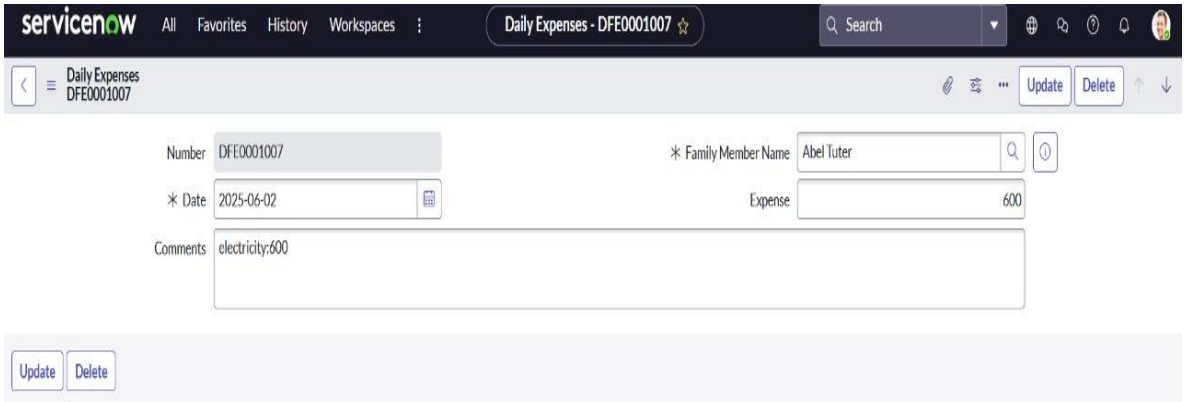
The system was tested for:

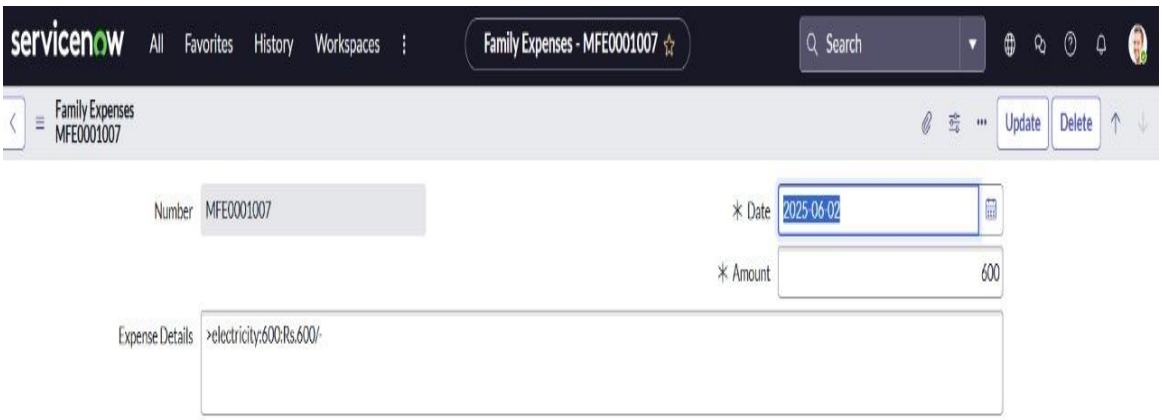
- Correct auto-numbering of records
- Accurate display of related lists
- Business rule triggers on insert/update
- Timely budget alerts
- Proper linkage of daily to family expenses

7. RESULTS

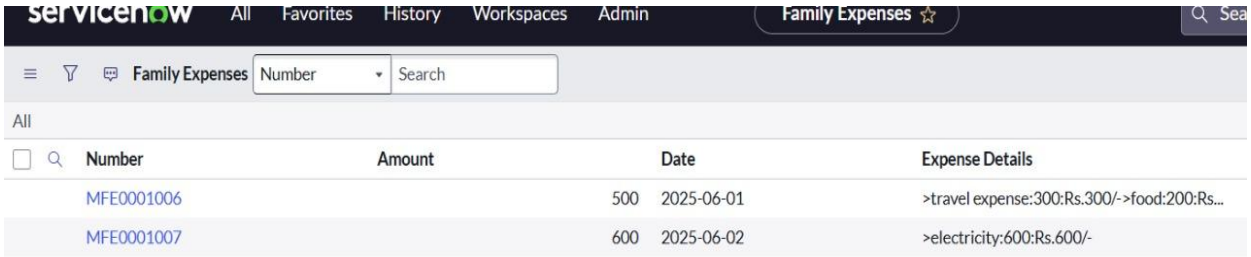
7.1 Output Screenshots

- Tested record creation for both tables.

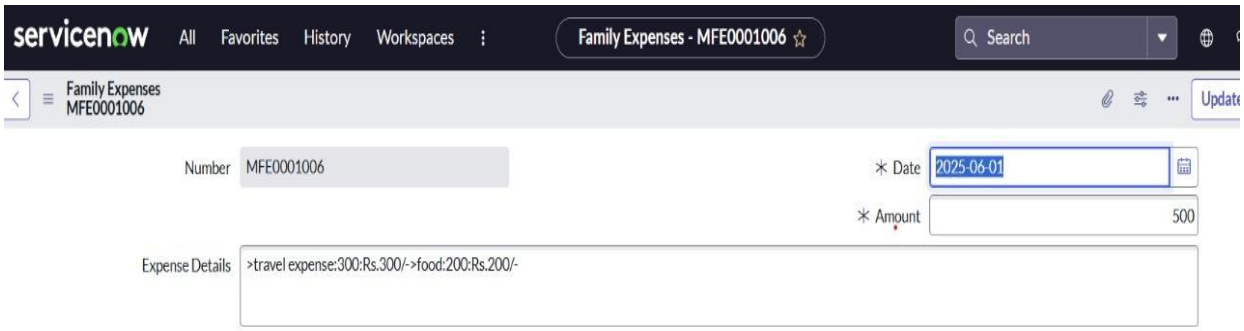




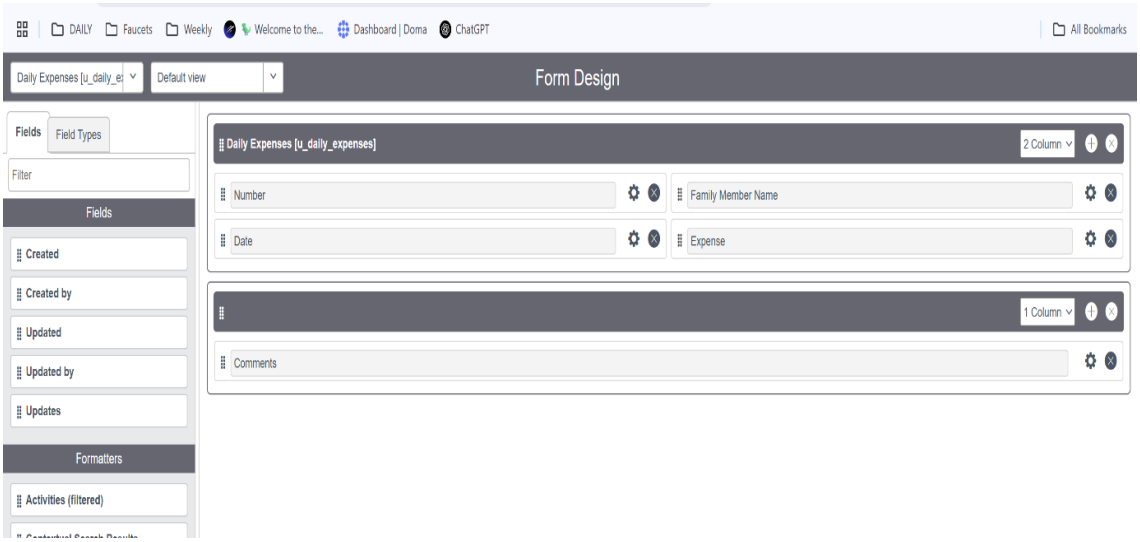
- Verified auto-numbering with correct prefixes.



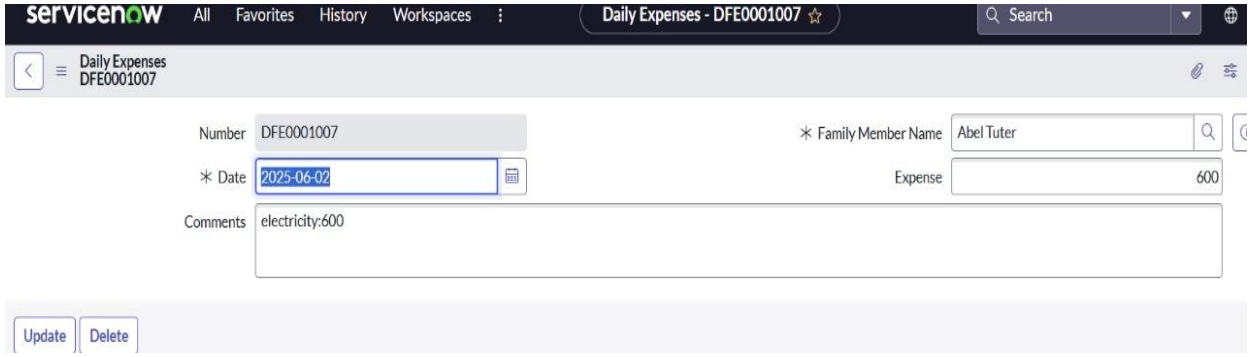
- Checked related list accuracy for displaying linked records.
- Validated business rule execution on insert/update.

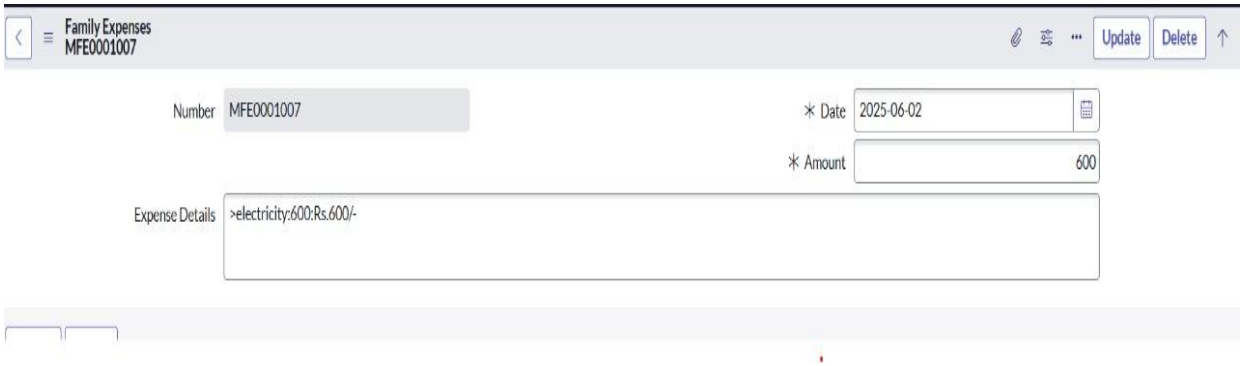


- Reviewed form design for clarity and ease of use.



- Confirmed data linkage integrity across tables.





8. ADVANTAGES & DISADVANTAGES

Advantages

- Provides a unified and automated way to track household expenses
- Enables real-time monitoring of spending against budget limits
- Scalable design allows for easy feature additions
- Quick to build and deploy using low-code tools on ServiceNow

Disadvantages

- Users need familiarity with ServiceNow to configure or modify the system
- Functionality relies on access to a Personal Developer Instance or licensed ServiceNow environment

9.CONCLUSION:

The project delivered a functional prototype for tracking and managing family expenses using ServiceNow. It streamlines data entry, automates budget checks, and offers clear reports for smarter financial decisions.