

Calculating Family Expenses Using ServiceNow — Final Project Report

1. INTRODUCTION

1.1 Project Overview

The *Calculating Family Expenses Using ServiceNow* project aims to design and implement a centralized, automated expense management system for families. The system allows users to seamlessly record daily and family-level expenses, link them to an overall household budget, and monitor financial activity in real-time. Built entirely on ServiceNow's low-code/no-code platform, the solution utilizes custom tables, forms, business rules, related lists, and automation to simplify expense tracking. It ensures that family members can efficiently manage household finances without the complexities of manual bookkeeping or fragmented records. The project also demonstrates how ServiceNow can be extended beyond traditional IT workflows to solve practical, real-world problems like household financial management.

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.

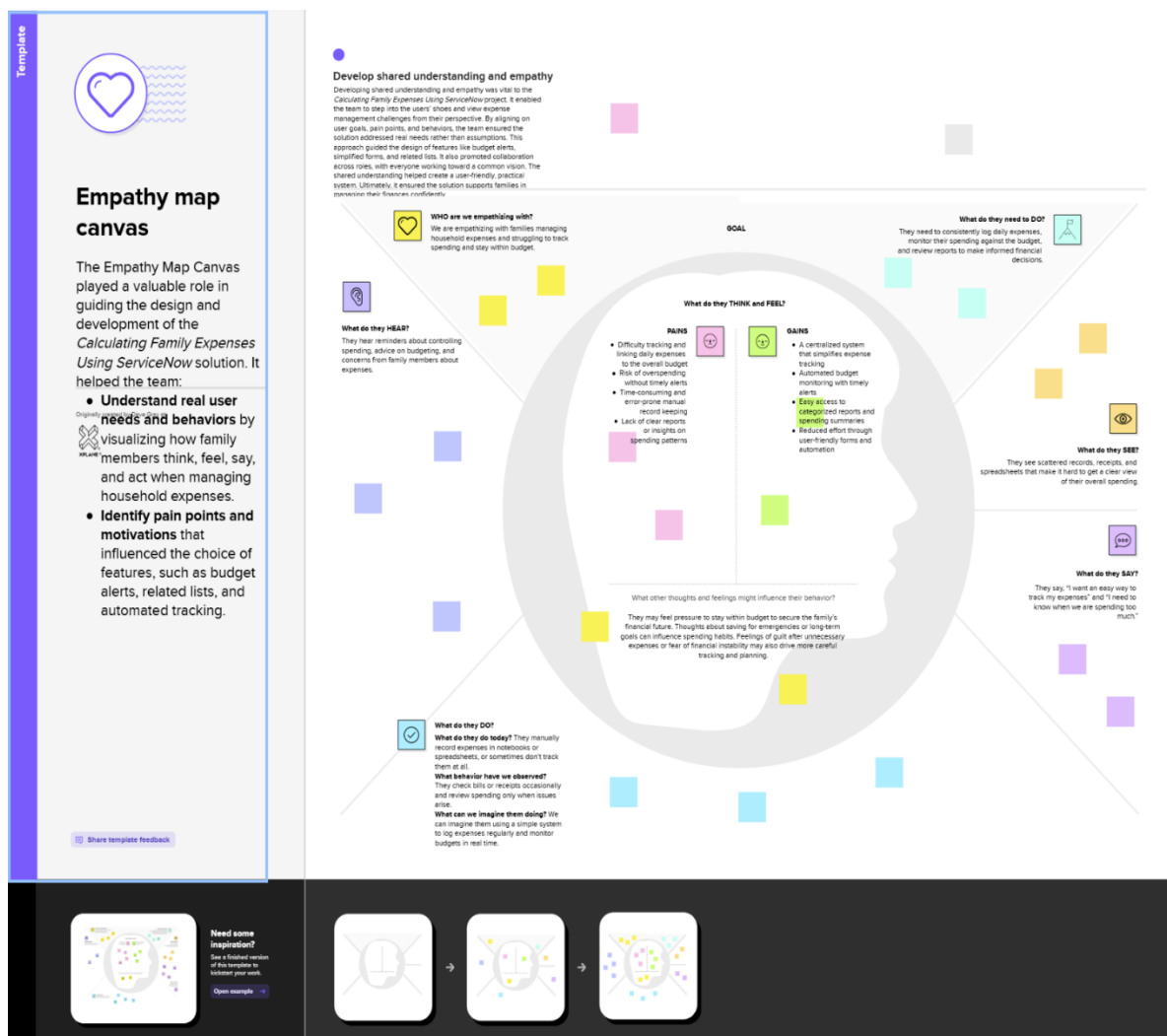
2. IDEATION PHASE

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.

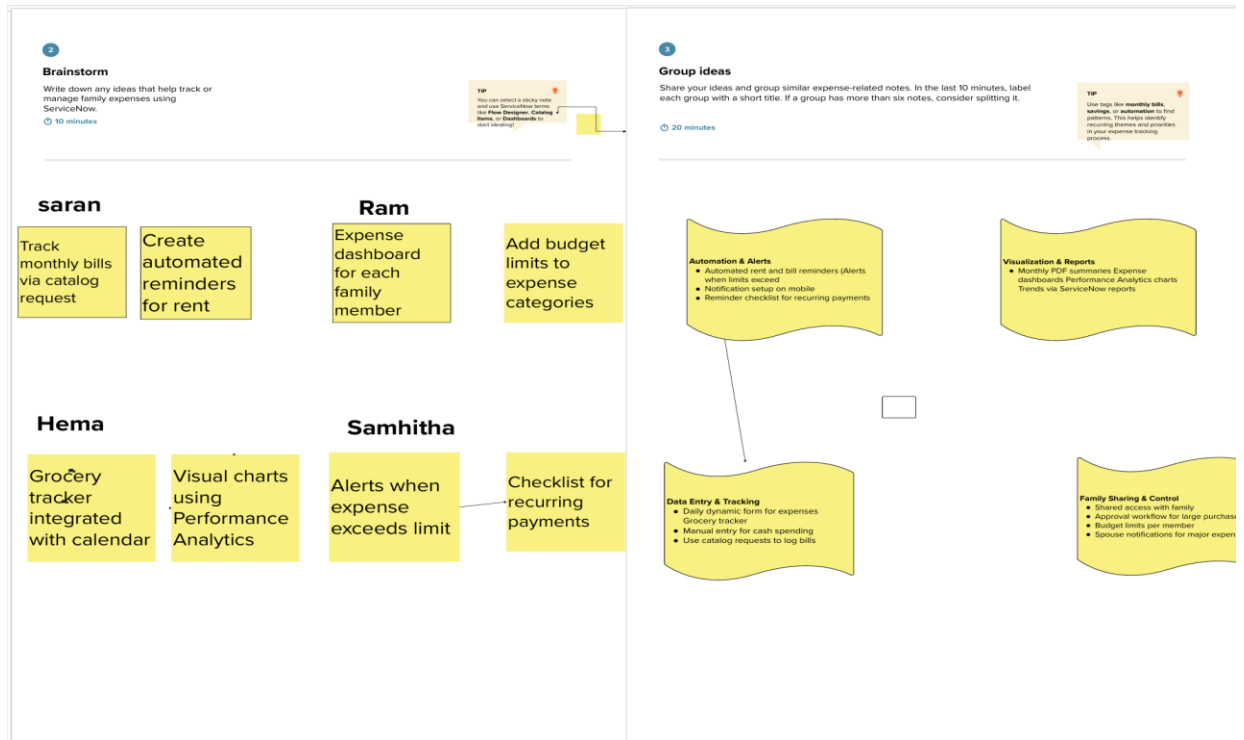
2.2 Empathy Map Canvas

- **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

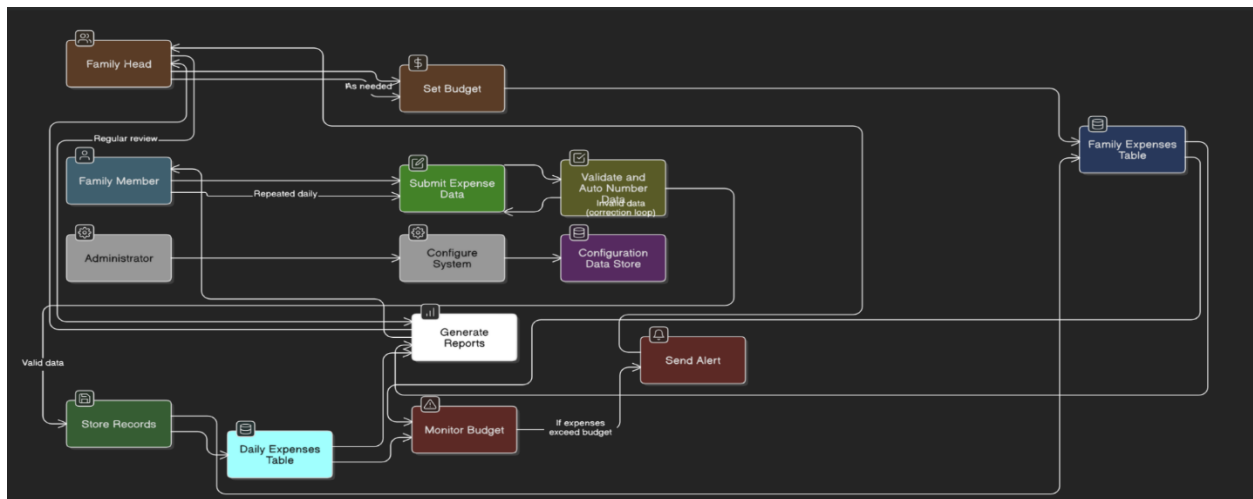
Users log daily expenses, monitor budget status, and review reports. The system provides alerts when spending approaches or exceeds set limits.

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.

Problem-Solution fit canvas 2.0			Purpose / Vision		
Define CS, fit into CC	<div>1. CUSTOMER SEGMENT(S)<div>CS</div></div> <div><ul style="list-style-type: none">Families managing household budgets (e.g. working parents, guardians, or family heads responsible for expenses)Tech-savvy family members who support household financial tracking</div>	<div>6. CUSTOMER CONSTRAINTS<div>CC</div></div> <div><ul style="list-style-type: none">Limited time to enter and manage dataNeed for simplicity (non-technical users)Dependence on free or existing platforms (budget constraints for tools)</div>	<div>5. AVAILABLE SOLUTIONS<div>AS</div></div> <div><ul style="list-style-type: none">Manual tracking (pen, paper, Excel sheets) — low cost, but prone to error and difficult to link dataBudgeting mobile apps — user-friendly, but no custom family-level linkage or advanced reportingNo centralized solution that integrates daily + family-level data + budget monitoring</div>	Explore AS, differentiate	
	<div>2. JOBS-TO-BE-DONE / PROBLEMS<div>J&P</div></div> <div><ul style="list-style-type: none">Record daily and family-level expenses in an organized wayLink daily expenses to household budget easilyMonitor and stay within a set monthly budgetGenerate reports for financial review</div>	<div>9. PROBLEM ROOT CAUSE<div>RC</div></div> <div><ul style="list-style-type: none">No dedicated, automated system to link individual expenses to family-level budget in real-timeTraditional tools are fragmented and error-prone</div>	<div>7. BEHAVIOUR<div>BE</div></div> <div><ul style="list-style-type: none">Record expenses in a notebook or spreadsheetOccasionally check receipts and try to summarize spending manuallyRarely review expenses until an issue arises</div>	Focus on J&P, tap into BE, understand RC	
Identify strong TR & EM	<div>3. TRIGGERS<div>TR</div></div> <div><ul style="list-style-type: none">Rising household costs or overspending incidentsNeed to review monthly spending patternsPlanning for savings or specific family financial goals</div>	<div>10. YOUR SOLUTION<div>SL</div></div> <div><ul style="list-style-type: none">A centralized system built on ServiceNow with:<ul style="list-style-type: none">Family Expenses and Daily Expenses tables linked via relationshipsAuto-numbering for tracking entriesBudget monitoring business rules that send alertsReports generated for categorized spendingUser-friendly custom forms for easy data entry</div>	<div>8. CHANNELS of BEHAVIOUR<div>CH</div></div> <div><div>8.1 ONLINE</div><ul style="list-style-type: none">Look for budgeting templates, Google Sheets, or apps; search for tips on saving<div>8.2 OFFLINE</div><ul style="list-style-type: none">Collect and review receipts; informal discussions on spending; note expenses manually</div>	Extract online & offline CH of BE	
	<div>4. EMOTIONS: BEFORE / AFTER<div>EM</div></div> <div><ul style="list-style-type: none">Before: Disorganized, stressed, anxious about overspendingAfter: In control, confident, relieved with clear expense visibility</div>				

Problem-Solution fit canvas is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 license
Created by Daria Nepriakhina / Amaltama.com

AMALTAMA



Problem-Solution fit canvas is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 license
Created by Daria Nepriakhina / Amaltama.com



4.2 Proposed Solution

A ServiceNow-based tool with:

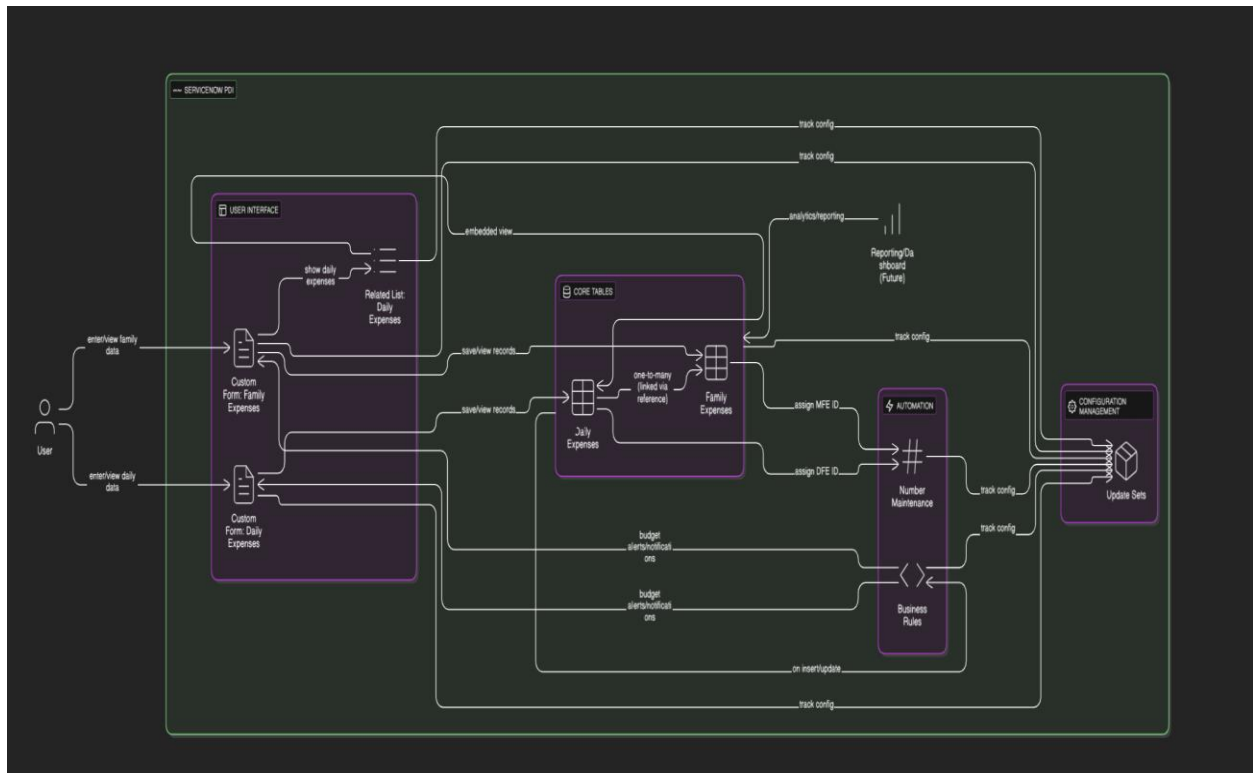
- Custom tables and forms
- Relationships and related lists
- Business rules for automation

- 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:

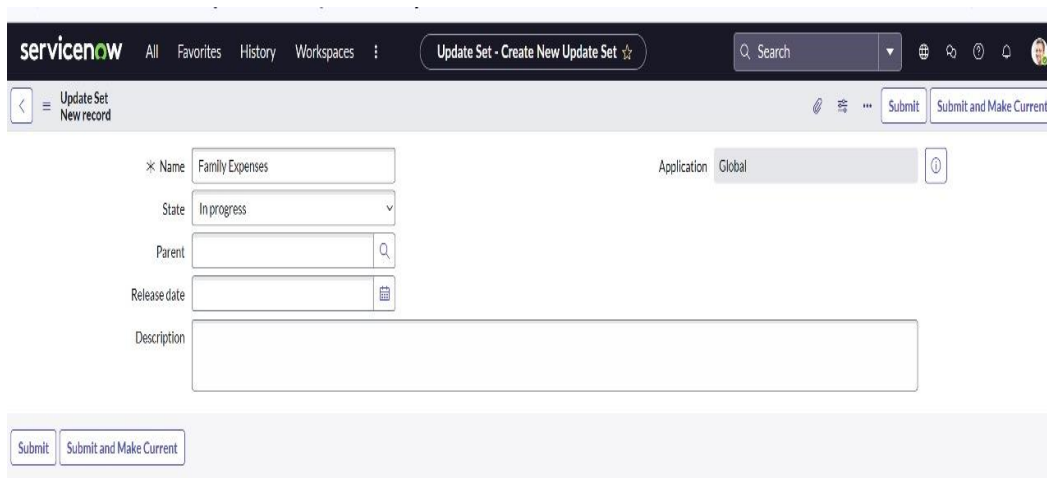
- **Sprint 1:** Instance setup, update set, table creation (9 points)
- **Sprint 2:** Relationships, related list, business rules (5 points)
- **Sprint 3:** Budget alerts, reports (6 points)

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



The screenshot shows the ServiceNow interface for creating a new update set. The header bar includes the ServiceNow logo, navigation links (All, Favorites, History, Workspaces), and a search bar. The main form is titled 'Update Set - Create New Update Set'. It contains the following fields:

- Name:** Family Expenses
- State:** In progress
- Parent:** (empty field with a search icon)
- Release date:** (empty field with a calendar icon)
- Description:** (empty text area)
- Application:** Global

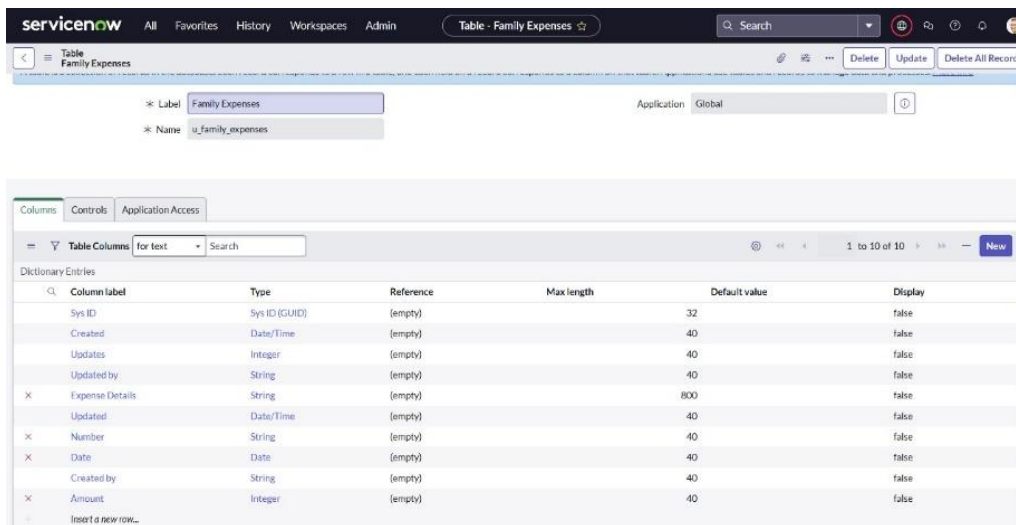
At the bottom of the form, there are two buttons: 'Submit' and 'Submit and Make Current'.

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



The screenshot shows the ServiceNow interface for the 'Family Expenses' table. The 'Columns' tab is active, displaying a list of columns with their respective types, references, max lengths, default values, and display settings.

Column label	Type	Reference	Max length	Default value	Display
Sys ID	Sys ID (GUID)	(empty)	32		false
Created	Date/Time	(empty)	40		false
Updated	Integer	(empty)	40		false
Updated by	String	(empty)	40		false
Expense Details	String	(empty)	800		false
Updated	Date/Time	(empty)	40		false
Number	String	(empty)	40		false
Date	Date	(empty)	40		false
Created by	String	(empty)	40		false
Amount	Integer	(empty)	40		false
Insert a new row...					

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

The screenshot shows the ServiceNow 'Table - New Record' interface for a table named 'Daily Expenses'. The form includes fields for 'Label' (Daily Expenses), 'Name' (u_daily_expenses), and 'Extends table'. On the right, there are checkboxes for 'Create module' and 'Create mobile module', both of which are checked, and a dropdown for 'Add module to menu' set to 'Family Expenditure'. Below these, there are tabs for 'Columns', 'Controls', and 'Application Access'. The 'Columns' tab is active, showing a 'Table Columns' section with a search bar. Below this is a 'Dictionary Entries' table with columns: Column label, Type, Reference, Max length, Default value, and Display. The table lists five entries: Number (String), Date (Date), Expense (Integer), Family Member Name (Reference, Max length 800), and Comments (String, Max length 800). At the bottom of the 'Dictionary Entries' table is a link to 'Insert a new row...'. The 'Submit' and 'Cancel' buttons are at the bottom left.

Column label	Type	Reference	Max length	Default value	Display
Number	String				false
Date	Date				false
Expense	Integer				false
Family Member Name	Reference		800		false
Comments	String		800		false

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

servicenow All Favorites History Workspaces Business Rule - New Record Search

Business Rule New record

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 Application: Global

Table: Daily Expenses [u_daily_expenses]

Active ☒ Advanced ☒

When to run Actions 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 ☐

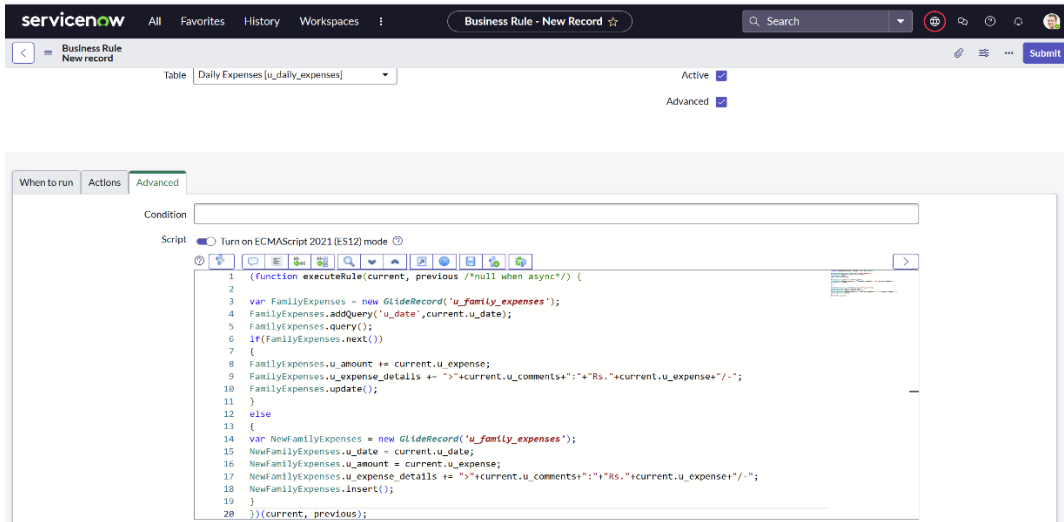
Filter Conditions: Add Filter Condition Add "OR" Clause

-- choose field -- -- oper -- -- value --

Role conditions

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

```
(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);
```



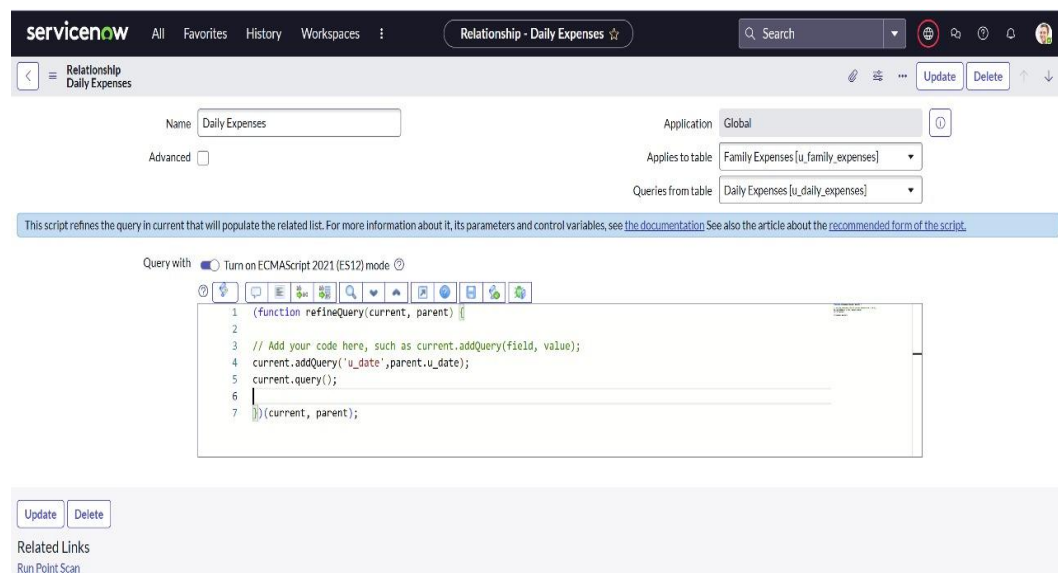
8. Final Relationship Configuration

- 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

```

(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);

```



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.

This screenshot shows the 'Daily Expenses' form in ServiceNow. The form is titled 'Daily Expenses - DFE0001007'. It contains the following fields:

- Number:** DFE0001007
- * Family Member Name:** Abel Tuter
- * Date:** 2025-06-02
- Expense:** 600
- Comments:** electricity:600

At the bottom of the form, there are 'Update' and 'Delete' buttons.

This screenshot shows the 'Family Expenses' form in ServiceNow. The form is titled 'Family Expenses - MFE0001007'. It contains the following fields:

- Number:** MFE0001007
- * Date:** 2025-06-02
- * Amount:** 600
- Expense Details:** >electricity:600:Rs.600/-

At the bottom of the form, there are 'Update' and 'Delete' buttons.

- Verified auto-numbering with correct prefixes.

Number	Amount	Date	Expense Details
MFE0001006	500	2025-06-01	>travel expense:300:Rs.300/->food:200:Rs...
MFE0001007	600	2025-06-02	>electricity:600:Rs.600/-

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

Number: MFE0001006

* Date: 2025-06-01

* Amount: 500

Expense Details: >travel expense:300:Rs.300/->food:200:Rs.200/-

- Reviewed form design for clarity and ease of use.

Form Design

Daily Expenses [u_daily_expenses]

Fields: Number, Family Member Name, Date, Expense, Comments

Family Expenses [u_family_expenses] Default view Form Design

Fields Field Types

Filter

Fields

- Created
- Created by
- Updated
- Updated by
- Updated

Formatters

- Activities (Filtered)
- Contextual Search Results
- Rating

Family Expenses [u_family_expenses] 2 Column

- Number
- Date
- Amount

Expense Details

- Confirmed data linkage integrity across tables.

servicenow All Favorites History Workspaces Daily Expenses - DFE0001007 Search

< Daily Expenses DFE0001007

Number DFE0001007 * Family Member Name Abel Tuter

* Date 2025-06-02 Expense 600

Comments electricity:600

Update Delete

< Family Expenses MFE0001007 Update Delete

Number MFE0001007 * Date 2025-06-02

* Amount 600

Expense Details >electricity:600:Rs.600/-

8. ADVANTAGES & DISADVANTAGES

Advantages

- Centralized, automated expense tracking
- Real-time budget monitoring
- Easy to extend for more features
- Low-code development for rapid deployment

Disadvantages

- Requires ServiceNow knowledge for configuration
- Depends on PDI availability or enterprise licensing

9. CONCLUSION

The project successfully delivered a working prototype that helps families track and manage expenses on ServiceNow. The system simplifies data entry, automates budget monitoring, and provides clear reporting to support better financial decisions.

10. FUTURE SCOPE

- Integration with external financial planning tools
- Advanced analytics and dashboards
- Mobile-friendly forms for easier data entry
- Multi-family or community-level expense tracking