Date	25 June 2025
Team ID	LTVIP2025TMID20412
Project Name	Calculating Family Expenses using Service Now
Mentor Name	Dr Shaik Salma Begum

Calculating Family Expenses Using ServiceNow

Introduction

The *Calculating Family Expenses Using ServiceNow* project aims to build a centralized and scalable system for recording, categorizing, and managing household expenses. This solution enables families to track their daily and overall spending patterns, set budgets, and generate reports, thereby supporting better financial planning. The project leverages ServiceNow's platform capabilities to provide seamless integration of components, user-friendly forms, automated business logic, and scalability for families of varying sizes.

Key Features

- **Centralized expense tracking** at both family and daily levels.
- **Auto-numbering** of records with configurable prefixes (e.g., MFE for Family Expenses, DFE for Daily Expenses).
- **Custom forms** for streamlined data entry and validation.
- **Related lists** to view associated daily expenses under family records.
- **Business rules** to automate actions during record creation and updates.
- **Dynamic queries** to refine related list data based on relationships.
- **Scalability** to handle varying financial complexities and household sizes.
- **Compatibility with update sets** for easy migration or versioning of configurations.

Pre-requisites

- Active **ServiceNow Personal Developer Instance (PDI)** obtained from developer.servicenow.com.
- Basic familiarity with:
  - **ServiceNow navigation and configuration**
  - **Tables, forms, and related lists**
  - **Business rules and Flow Designer**
- GitHub or other repository (optional) for maintaining documentation and exported update sets.

Requirements

Functional Requirements

- Create and manage family-level expense records.
- Create and manage daily expense entries linked to a family record.
- Auto-generate record numbers with prefixes for both tables.
- Display related daily expenses within the family expense form.
- Automate actions via business rules during insert/update operations.
- Enable reporting through ServiceNow capabilities.

Non-Functional Requirements

- Ensure ease of use through intuitive forms and layouts.
- Maintain data integrity between related records.
- Design for scalability to accommodate more records or families over time.

Phases of the Project

Ideation Phase

The project was conceptualized as a family expense tracking solution using ServiceNow. The goal was to empower users to record and monitor expenses at both daily and family levels, and facilitate better financial management through an integrated system.

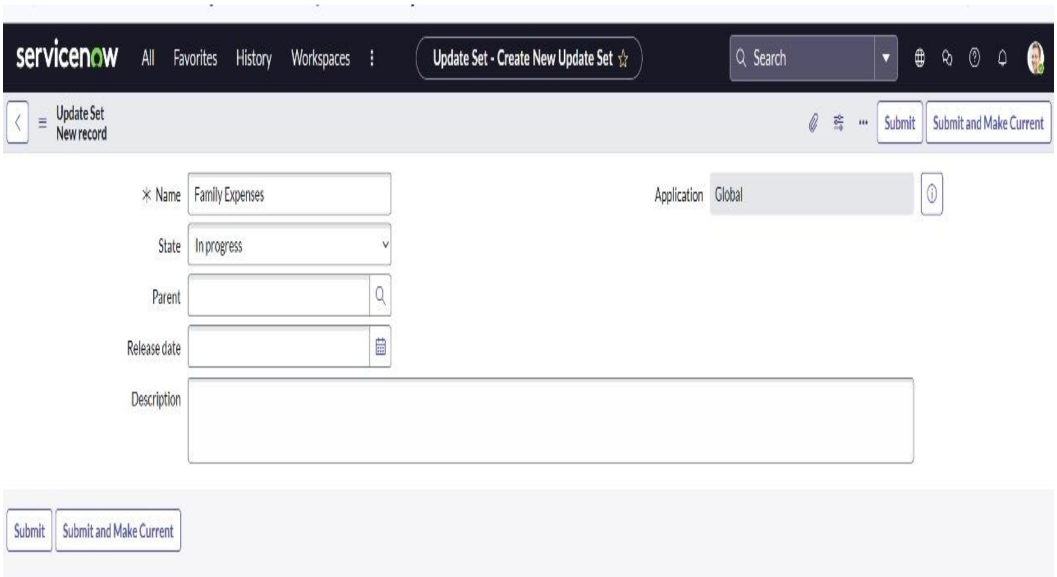
Requirement Analysis Phase

- Identified two key tables:
  - **Family Expenses**: For overall household expense summaries.
  - **Daily Expenses**: For individual daily spending entries linked to a family record.
- Defined the relationship: Daily Expenses linked to Family Expenses.
- Selected features:
  - Auto-numbering with custom prefixes (MFE, DFE)
  - Custom forms for better data entry
  - Related lists for visibility of linked data
  - Business rules for automation
- Chose ServiceNow components: Tables, Form Designer, Number Maintenance, Relationships, Business Rules.

Project Planning Phase

The team executed these milestones:

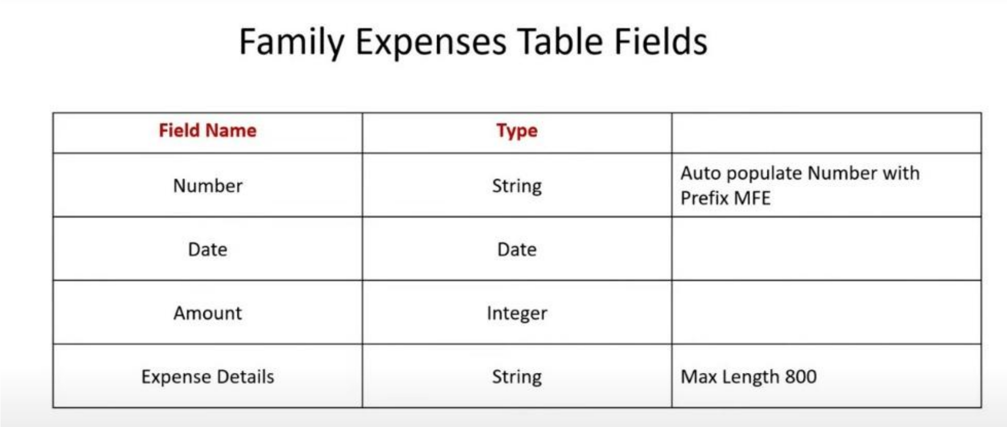
1. ServiceNow Instance Setup
    - Signed up at developer.servicenow.com and requested a Personal Developer Instance (PDI)
    - Filled necessary details; received instance access credentials via email
    - Logged in and prepared the instance for development
2. Creation of New Update Set
    - Navigated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



Update Set
New record

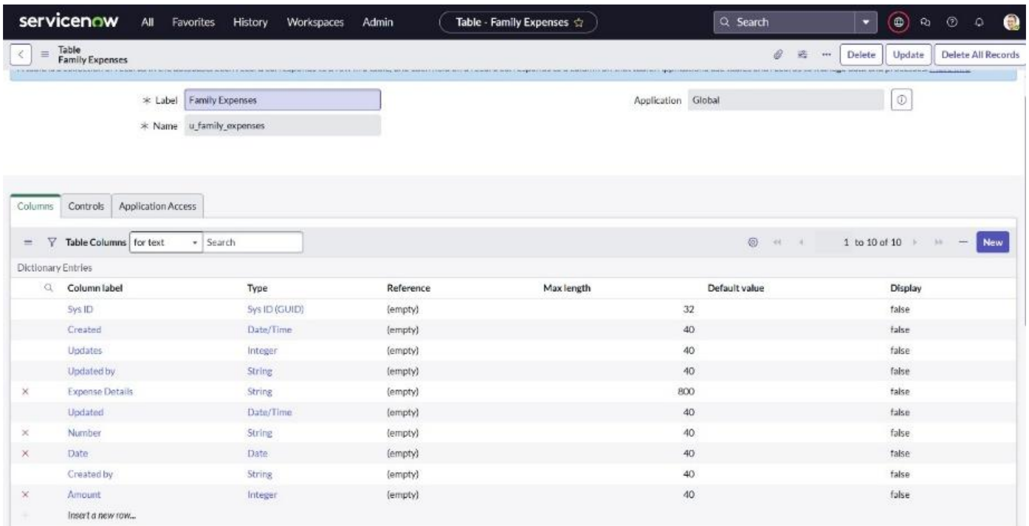
* Name	Family Expenses
State	In progress
Parent	
Release date	
Description	
Application	Global

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



∗ Label: Family Expenses

∗ Name: u_family_expenses

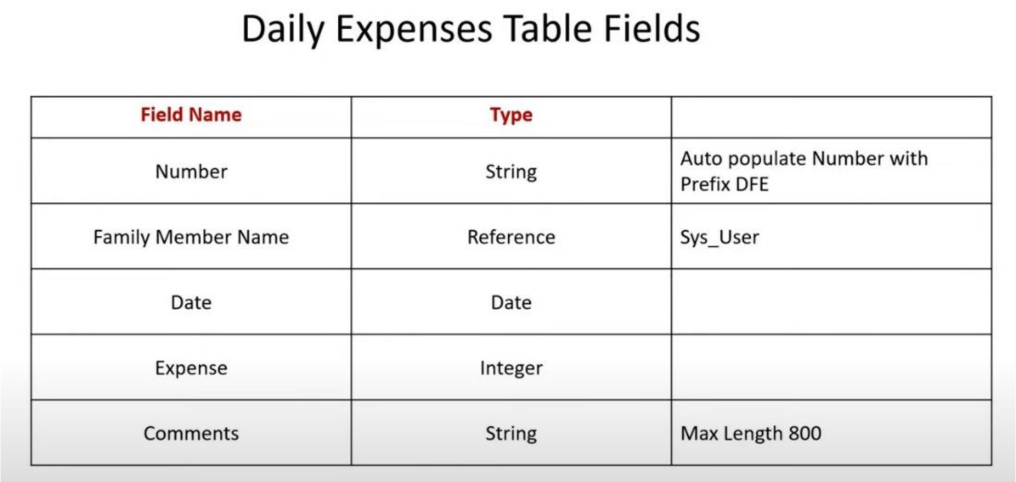
Application: Global

Dictionary Entries

	Column label	Type	Reference	Max length	Default value	Display
	Sys ID	Sys ID (GUID)	(empty)	32		false
	Created	Date/Time	(empty)	40		false
	Updates	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

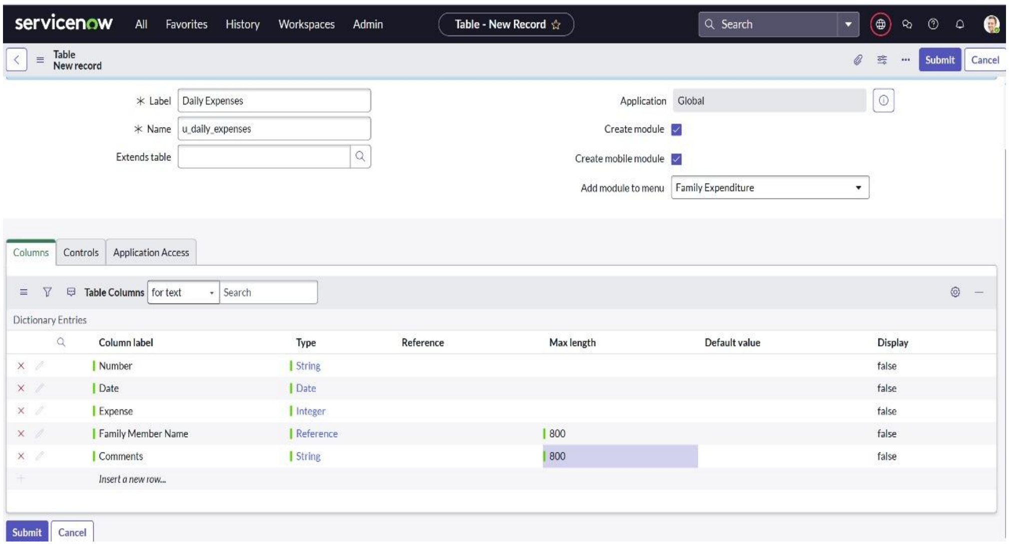
4. Creation of Daily Expenses Table

- 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



∗ Label: Daily Expenses

∗ Name: u_daily_expenses

Extends table:

Application: Global

Create module ☑

Create mobile module ☑

Add module to menu: Family Expenditure

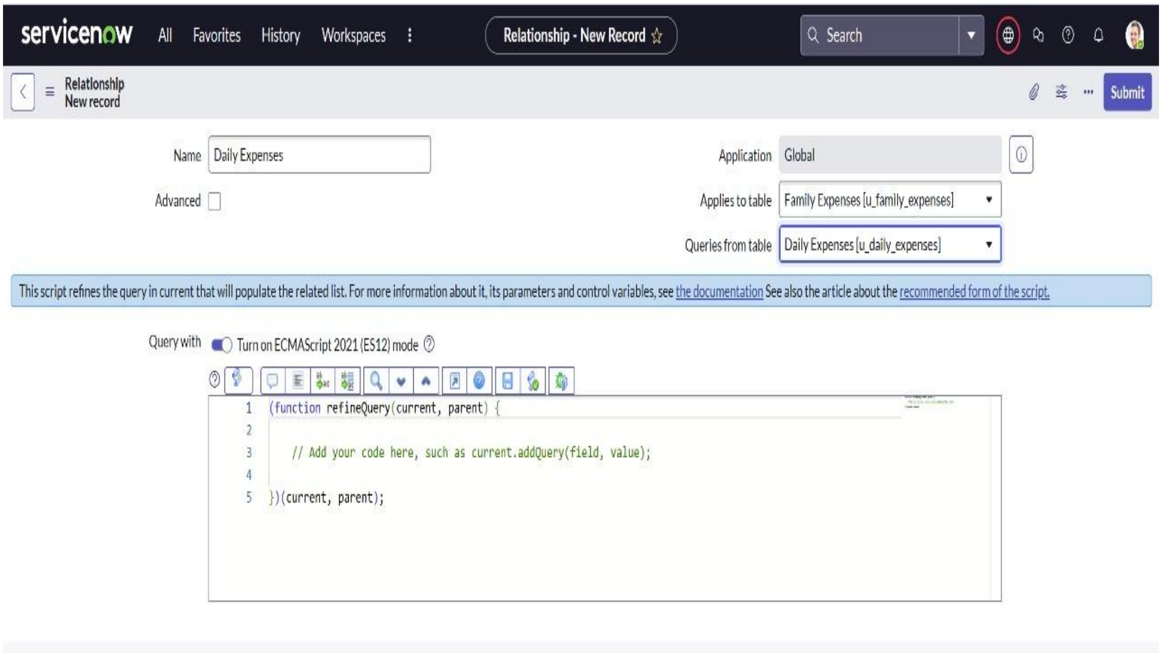
Dictionary Entries

	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

Project Design Phase

The design phase focused on setting up relationships, related lists, and automation:

1. 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



Name: Daily Expenses

Advanced □

Application: Global

Applies to table: Family Expenses [u_family_expenses]

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.

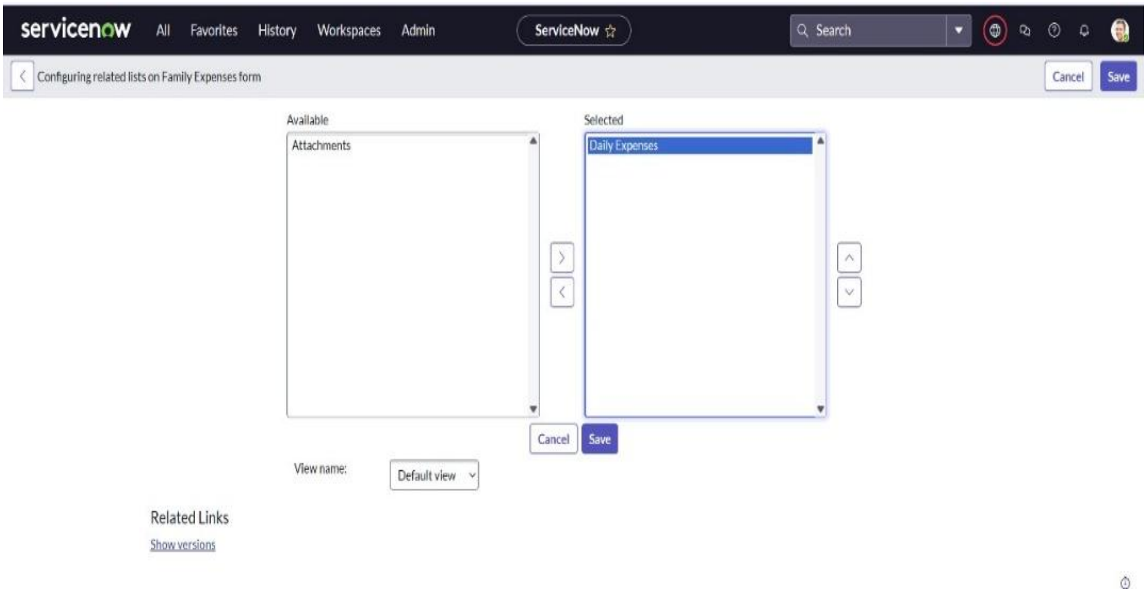
Query with: Turn on ECMAScript 2021 (ES12) mode

```
(function refineQuery(current, parent) {

    // Add your code here, such as current.addQuery(field, value);

})(current, parent);
```

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



Configuring related lists on Family Expenses form

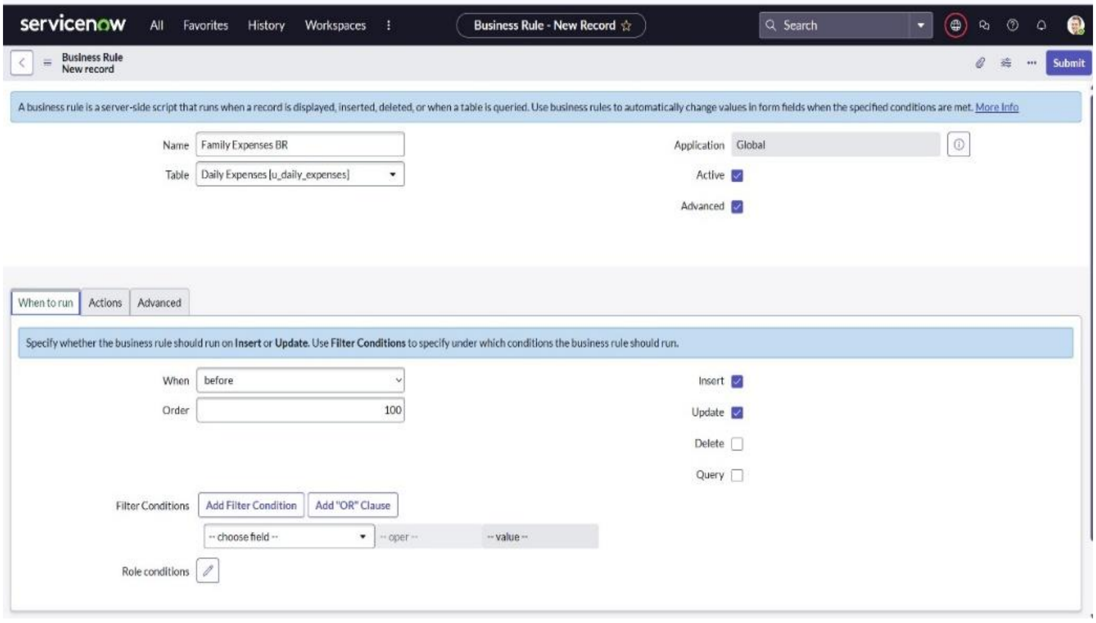
Available: Attachments

Selected: Daily Expenses

View name: Default view

3. 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)



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	☑

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	Insert	☑
Order	100	Update	☑
		Delete	□
		Query	□

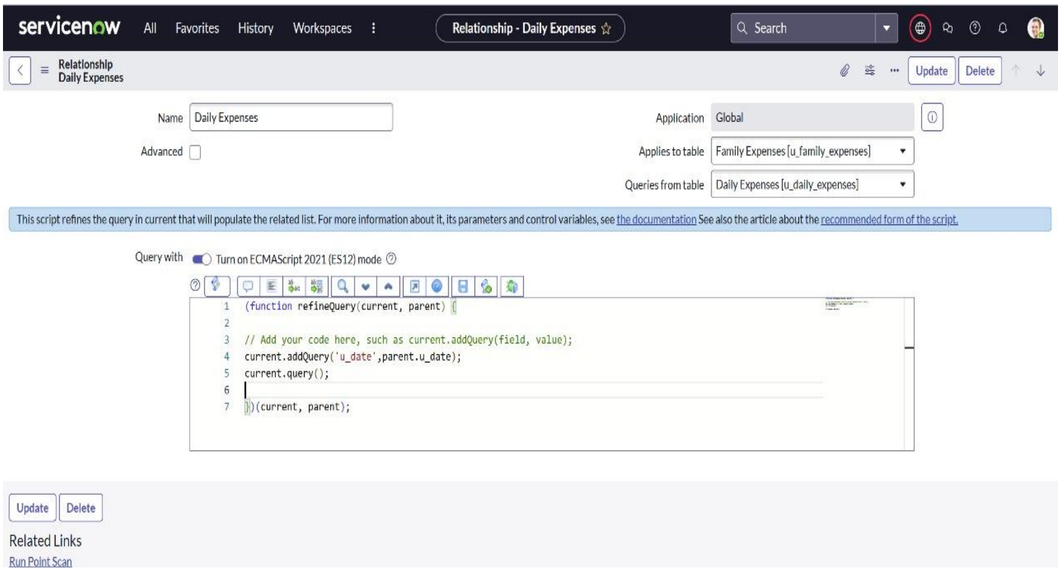
Filter Conditions: Add Filter Condition | Add "OR" Clause

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

Role conditions

4. 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



Name	Daily Expenses	Application	Global
Advanced	□	Applies to table	Family Expenses [u_family_expenses]
		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

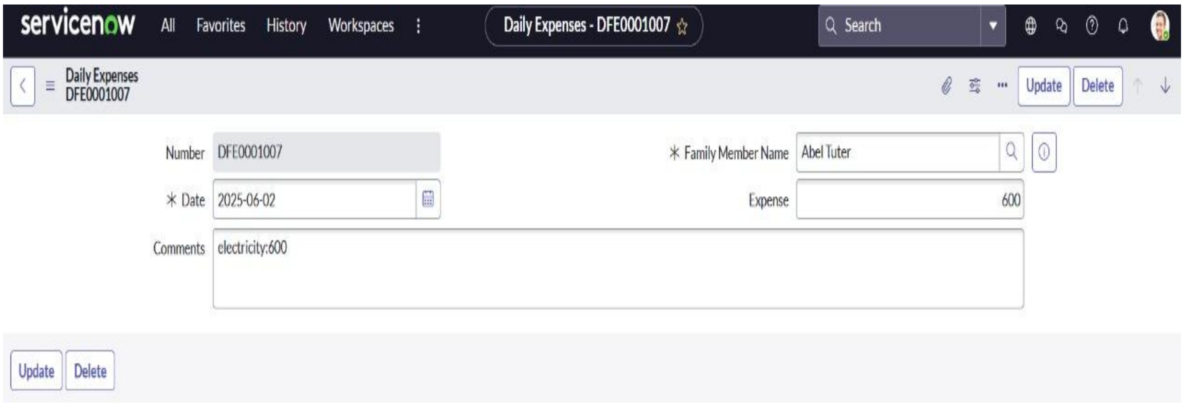
```
(function refineQuery(current, parent) {

// Add your code here, such as current.addQuery(field, value);
current.addQuery('u_date',parent.u_date);
current.query();

})(current, parent);
```

Performance Testing

- Tested record creation for both tables.



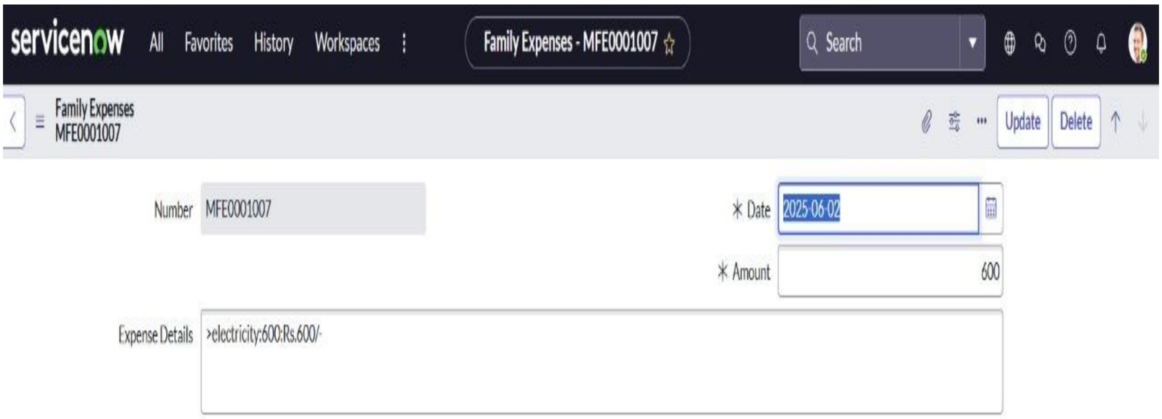
Number: DFE0001007

✱ Family Member Name: Abel Tuter

✱ Date: 2025-06-02

Expense: 600

Comments: electricity:600



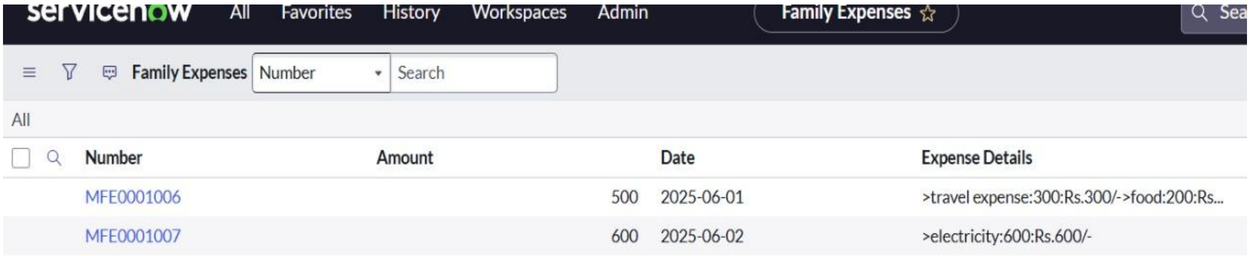
Number: MFE0001007

✱ Date: 2025-06-02

✱ Amount: 600

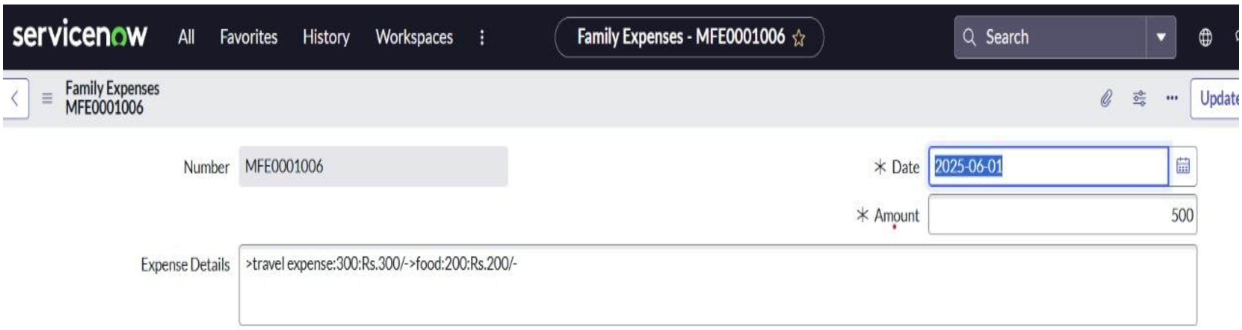
Expense Details: >electricity:600:Rs.600/-

- 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.



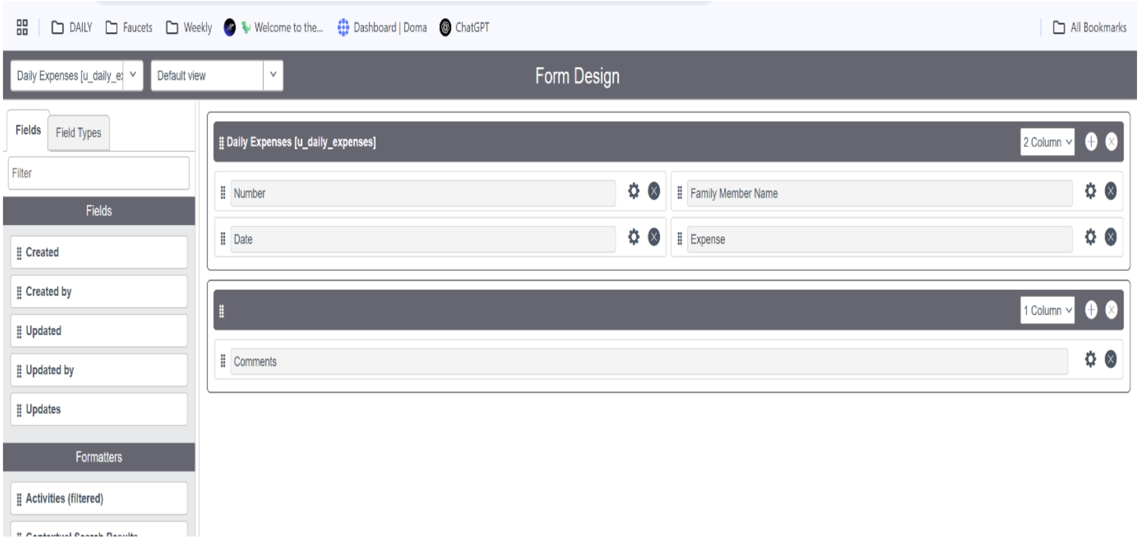
Number: MFE0001006

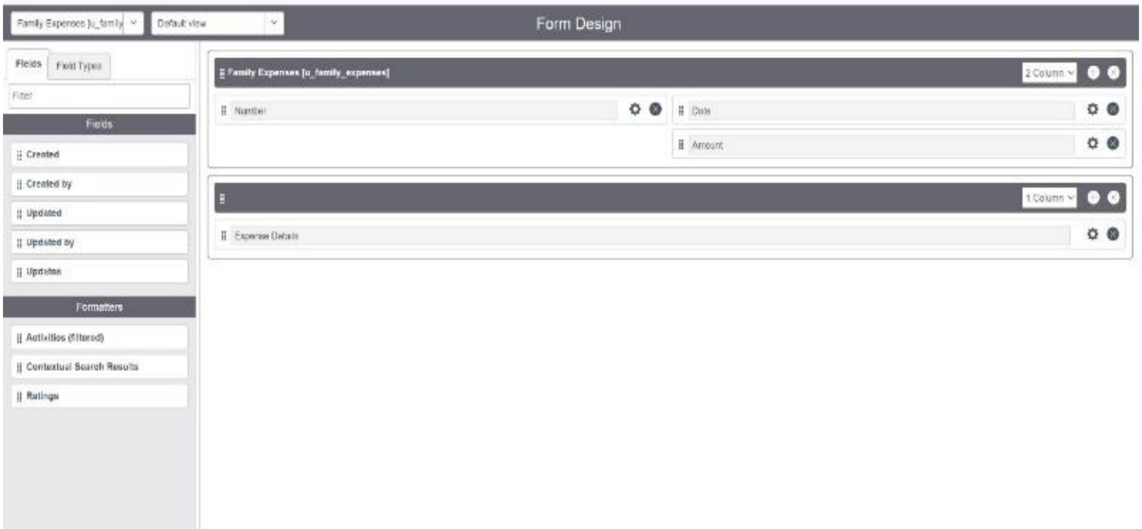
✱ Date: 2025-06-01

✱ Amount: 500

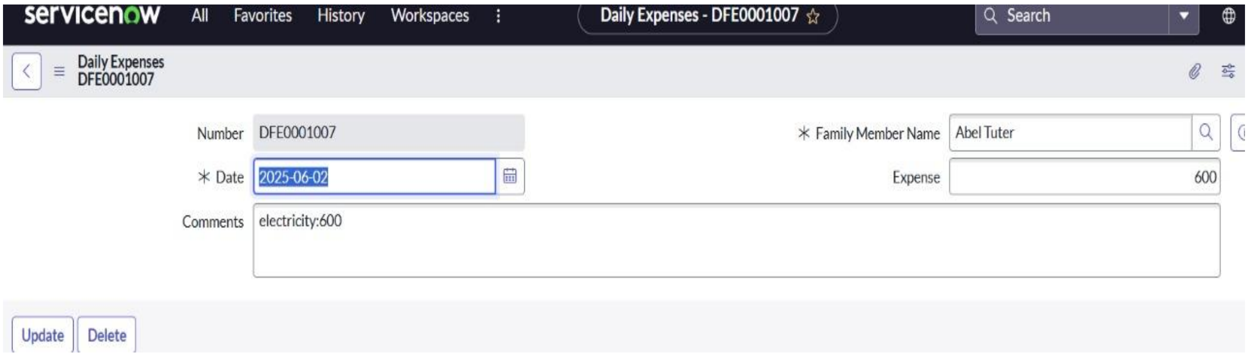
Expense Details: >travel expense:300:Rs.300/->food:200:Rs.200/-

- Reviewed form design for clarity and ease of use.





- Confirmed data linkage integrity across tables.



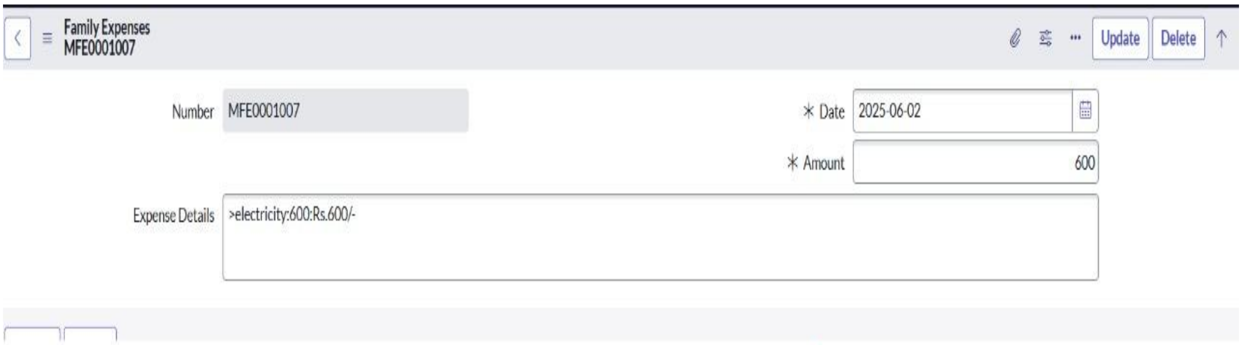
Number DFE0001007

✱ Family Member Name Abel Tuter

✱ Date 2025-06-02

Expense 600

Comments electricity:600



Number MFE0001007

✱ Date 2025-06-02

✱ Amount 600

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

Conclusion

The *Calculating Family Expenses Using ServiceNow* project delivers an effective solution for managing household financial data. By centralizing expense tracking and leveraging automation, the system ensures accuracy, transparency, and ease of use. The project demonstrates how ServiceNow can be adapted beyond enterprise IT to support real-world household applications, providing families with a tool to make informed financial decisions and maintain control over their budgets.