

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

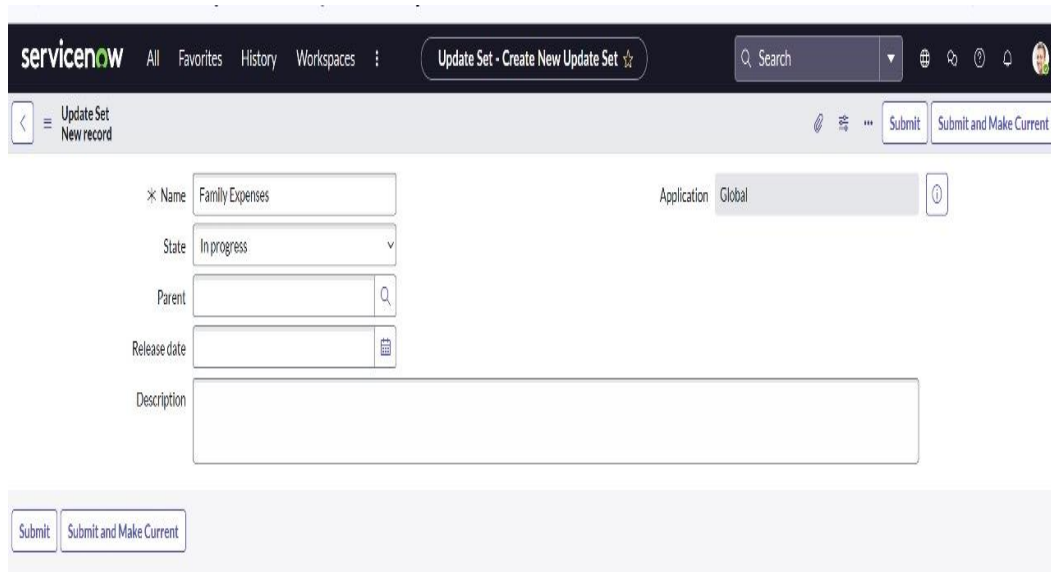
The team executed these milestones:

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



The screenshot shows the ServiceNow interface for creating a new update set. The header includes the ServiceNow logo, navigation links (All, Favorites, History, Workspaces), a search bar, and a user profile icon. The main form is titled 'Update Set - Create New Update Set'. It contains several fields: 'Name' (set to 'Family Expenses'), 'Application' (set to 'Global'), 'State' (set to 'In progress'), 'Parent' (with a search icon), 'Release date' (with a calendar icon), and a large 'Description' text area. At the bottom, 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 configuration page for a table named 'Family Expenses'. The 'Columns' tab is active, displaying a list of columns with their respective types, references, and lengths. The columns include Sys ID, Created, Updates, Updated by, Expense Details, Updated, Number, Date, Created by, and Amount. The 'Number' column is highlighted, indicating it is the current selection.

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

The screenshot shows the ServiceNow configuration page for a new table named 'Daily Expenses'. The 'Columns' tab is active, displaying a list of columns with their respective types, references, and lengths. The columns include Number, Date, Expense, Family Member Name, and Comments. The 'Number' column is highlighted, indicating it is the current selection.

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

The screenshot shows the 'Relationship - New Record' form in ServiceNow. The 'Name' field is set to 'Daily Expenses'. The 'Application' is 'Global'. The 'Applies to table' is 'Family Expenses [u_family_expenses]'. The 'Queries from table' is 'Daily Expenses [u_daily_expenses]'. Below these fields, there is a blue informational banner. At the bottom, there is a 'Query with' section with a toggle for 'Turn on ECMAScript 2021 (ES12) mode' and a code editor containing the following JavaScript code:

```
1 (function refineQuery(current, parent) {  
2  
3   // Add your code here, such as current.addQuery(field, value);  
4  
5 })(current, parent);
```

2. Related List Configuration

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

The screenshot shows the 'Configuring related lists on Family Expenses form' dialog in ServiceNow. It features two lists: 'Available' and 'Selected'. The 'Available' list contains 'Attachments'. The 'Selected' list contains 'Daily Expenses'. There are navigation buttons between the lists and 'Cancel'/'Save' buttons at the bottom. Below the lists, there is a 'View name' dropdown set to 'Default view'. At the bottom left, there is a 'Related Links' section with a 'Show versions' link.

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)

The screenshot shows the 'Business Rule - New Record' configuration page in ServiceNow. The 'Name' field is 'Family Expenses BR' and the 'Table' is 'Daily Expenses [u_daily_expenses]'. The 'Application' is set to 'Global'. The 'Active' checkbox is checked, and the 'Advanced' checkbox is also checked. Below the 'When to run' tab, the 'When' dropdown is set to 'before' and the 'Order' is '100'. The 'Filter Conditions' section has 'Add Filter Condition' and 'Add "OR" Clause' buttons. The 'Role conditions' section has a 'Choose field' dropdown, an 'operator' dropdown, and a 'value' input field. The 'Actions' section has checkboxes for 'Insert' (checked), 'Update' (checked), 'Delete' (unchecked), and 'Query' (unchecked).

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

The screenshot shows the 'Relationship - Daily Expenses' configuration page in ServiceNow. The 'Name' field is 'Daily Expenses'. The 'Application' is set to 'Global'. The 'Advanced' checkbox is unchecked. The 'Applies to table' dropdown is set to 'Family Expenses [u_family_expenses]'. The 'Queries from table' dropdown is set to 'Daily Expenses [u_daily_expenses]'. Below the configuration fields, there is a text box with the following script:

```

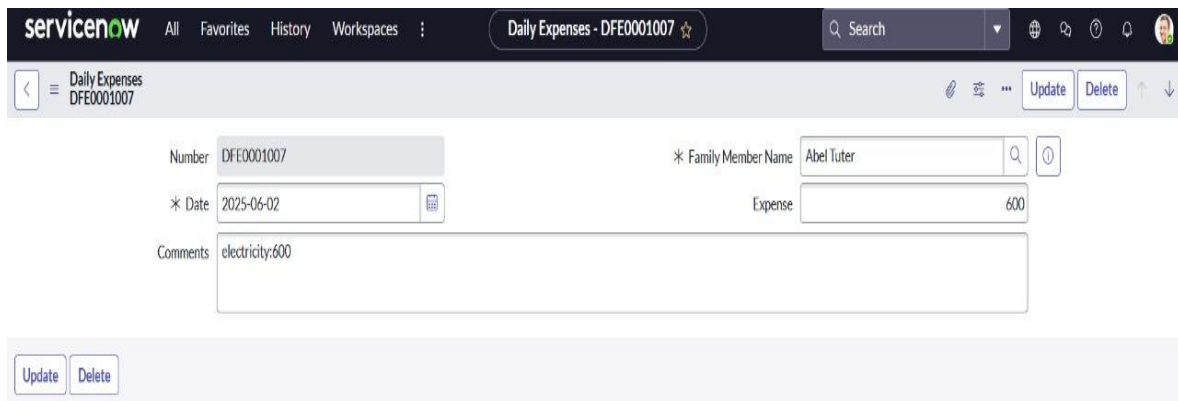
1 (function refineQuery(current, parent) {
2
3 // Add your code here, such as current.addQuery(field, value);
4 current.addQuery('u_date', parent.u_date);
5 current.query();
6
7 })(current, parent);

```

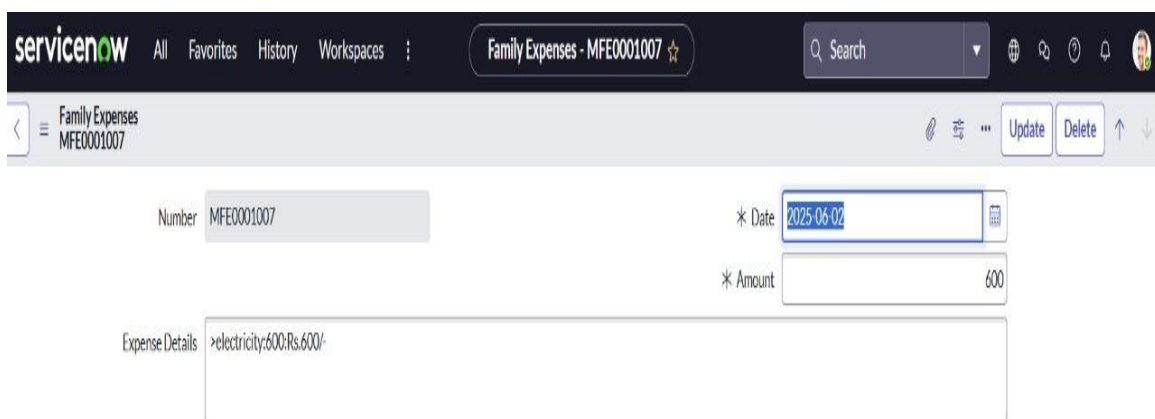
At the bottom, there are 'Update' and 'Delete' buttons, and a 'Related Links' section with a link to 'Run Point Scan'.

Performance Testing

- Tested record creation for both tables.

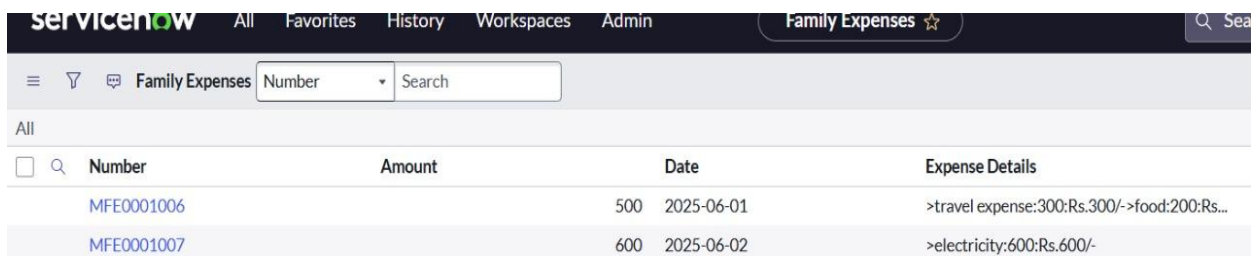


This screenshot shows the 'Daily Expenses' form in ServiceNow. The form is titled 'Daily Expenses - DFE0001007'. It contains the following fields: 'Number' (DFE0001007), '* Family Member Name' (Abel Tuter), '* Date' (2025-06-02), 'Expense' (600), and 'Comments' (electricity:600). There are 'Update' and 'Delete' buttons at the bottom left.



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), and 'Expense Details' (>electricity:600:Rs.600/-). There are 'Update' and 'Delete' buttons at the bottom right.

- Verified auto-numbering with correct prefixes.



This screenshot shows the 'Family Expenses' list view in ServiceNow. The table has columns: 'Number', 'Amount', 'Date', and 'Expense Details'. The data is as follows:

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.



This screenshot shows the 'Family Expenses' form in ServiceNow. The form is titled 'Family Expenses - MFE0001006'. It contains the following fields: 'Number' (MFE0001006), '* Date' (2025-06-01), '* Amount' (500), and 'Expense Details' (>travel expense:300:Rs.300/->food:200:Rs.200/-). There is an 'Update' button at the bottom right.

- Reviewed form design for clarity and ease of use.

Dashboard | Daily Expenses [u_daily_exp] | Default view | Form Design

Fields: Created, Created by, Updated, Updated by, Updates, Activities (filtered), Contextual Search Results

Field Types: Number, Date, Family Member Name, Expense, Comments

Form Design: 2 Column, 1 Column

Family Expenses [u_family_exp] | Default view | Form Design

Fields: Created, Created by, Updated, Updated by, Updates, Activities (filtered), Contextual Search Results, Ratings

Field Types: Number, Date, Amount, Expense Details

Form Design: 2 Column, 1 Column

- 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/-

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.