

# Calculating Family Expenses using Service Now

## 1. Introduction

### 1.1 Project Overview

This project, “**Calculating Family Expenses using Service Now**”, aims to streamline and manage student admissions, academic performance, and administrative data using the ServiceNow platform. The goal is to create a digital system with custom tables, forms, and business rules to automate educational processes.

### 1.2 Purpose

The Family Expense Tracker aims to help families manage their expenses efficiently. The application will allow users to:

1. Record expenses by category (e.g., groceries, utilities, entertainment)
2. Track expenses over time
3. Set budgets and receive notifications when approaching limits
4. Generate reports to analyze spending habits

## 2. Ideation Phase

### 2.1 Problem Statement

The project aims to develop a comprehensive expense calculation system using ServiceNow. This system will enable users to track and manage family expenses efficiently. It will include features such as expense categorization, budget setting, real-time tracking, and reporting capabilities. Utilizing ServiceNow's robust platform, the project will ensure seamless integration, user-friendly interface, and scalability to accommodate varying family sizes and financial complexities. The end goal is to empower users with the tools they need to make informed financial decisions and promote financial well-being within the family unit.

### 2.2 Empathy Map Canvas

Entity and Map Canvas for ServiceNow Family Expenses Table

## Entities:

1. Expense: Represents a single expense record

- Attributes: Date, Amount, Category, Description, Family Member

2. Category: Represents a category of expenses (e.g., groceries, utilities)

- Attributes: Name, Description

3. Family Member: Represents a family member who incurs expenses

- Attributes: Name, Relationship

4. Budget: Represents a budget for a specific category or overall expenses

- Attributes: Amount, Category, Timeframe

## Map Canvas:

The map canvas would visualize the relationships between these entities, showing:

1. Expense records linked to specific Category and Family Member records

2. Budget records linked to specific Category records or overall expenses

3. Relationships between Family Member records and their corresponding Expense records This entity and

map canvas design would help structure the Family Expenses table in ServiceNow, enabling effective data modeling and relationships.

## 2.3 Brainstorming

### Brainstorming: Calculating Family Expenses using ServiceNow

## Ideas:

**1. Automated Expense Tracking:** Use ServiceNow to automatically track expenses via integrations with bank accounts or credit cards.

**2. Customizable Budgeting:** Create customizable budgets for different expense categories (e.g., groceries, utilities, entertainment).

**3. Real-time Expense Reporting:** Generate real-time reports on family expenses, including charts and graphs to visualize spending habits.

**4. Alerts and Notifications:** Set up alerts and notifications when expenses exceed budgeted amounts or when specific expenses are incurred.

**5. Family Member Expense Tracking:** Allow multiple family members to track their expenses and view shared reports.

**6. Expense Categorization:** Use machine learning or manual categorization to classify expenses into different categories.

**7. Recurring Expense Management:** Track and manage recurring expenses, such as subscription services or utility bills.

## **## Questions:**

**1. How can we ensure data accuracy and security for financial information?**

**2. What integrations would be necessary for automated expense tracking?**

**3. How can we customize the solution for different family needs and preferences?**

This brainstorming session focuses on generating ideas and exploring possibilities for calculating family expenses using ServiceNow.

## **3. Requirement Phase**

### **3.1 Journey Map**

#### **Journey Phase: ServiceNow Calculating Family Expenses**

## **## Phases:**

**1. Discovery:** Identify family expense tracking needs and requirements

**2. Planning:** Define scope, budget, and timeline for the project

**3. Design:** Configure ServiceNow to meet family expense tracking needs

**4. Implementation:** Set up and test the solution

**5. Adoption:** Train family members on using the solution

**6. Review and Refine:** Continuously review and refine the solution to ensure it meets evolving family needs

## **## Activities:**

**1. Requirements Gathering:** Identify family expense tracking requirements

**2. Solution Design:** Design a solution using ServiceNow's features (e.g., forms, workflows, reports)

**3. Configuration and Testing:** Configure and test the solution

**4. Training and Adoption:** Train family members on using the solution

**5. Ongoing Monitoring and Maintenance:** Monitor and maintain the solution to ensure it continues to meet family needs

This journey phase outlines the key stages and activities involved in implementing a family expense tracking solution using ServiceNow.

### **3.2 Solution Requirements**

#### **## Functional Requirements:**

**1. Expense Tracking:** Ability to record and track expenses by category (e.g., groceries, utilities, entertainment)

**2. Budgeting:** Ability to set budgets for different expense categories

**3. Reporting:** Ability to generate reports on family expenses, including charts and graphs

**4. Alerts and Notifications:** Ability to set up alerts and notifications when expenses exceed budgeted amounts

**5. User Access:** Ability for multiple family members to access and track expenses

#### **## Non-Functional Requirements:**

**1. Data Security:** Ensure all financial data is secure and protected

**2. User Experience:** Intuitive and user-friendly interface for family members

**3. Scalability:** Solution should be scalable to accommodate growing family needs

**4. Integration:** Ability to integrate with existing financial systems (e.g., bank accounts)

#### **## Technical Requirements:**

**1. ServiceNow Platform:** Utilize ServiceNow's features (e.g., forms, workflows, reports) to build the solution

**2. Data Modeling:** Design a data model to store expense records, budgets, and family member information

**3. Security Configuration:** Configure security settings to ensure data protection and access control

These solution requirements outline the key functional, non-functional, and technical requirements for

building a family expense tracking solution using ServiceNow.

### 3.3 Data Flow Diagram

[User Input/API]



[Expense\_Record Table]

↓ (Trigger)

[Validation & Categorization (Flow Designer)]



[Aggregate Totals]



[Family\_Expense\_Summary Table]



[Dashboards / Reports / Notifications]

### 3.4 Technology Stack

- **Platform:** ServiceNow
- **Language:** JavaScript (for scripting)
- **Tools:** Update Sets, Table Designer, ClientScripts, Form Designer

## 4. Project Design

### 4.1 Problem-Solution Fit

- **Low-Code Platform:** Easy to build without deep technical skills.
- **Mobile-Friendly:** Track and input expenses from anywhere.
- **Automation:** Reduces manual work, saving time and effort.
- **Secure:** Set access control so only authorized members can edit data.

This solution provides a simple, centralized, and automated system for managing and tracking family expenses, ensuring clarity and control for everyone involved.

## 4.2 Proposed Solution

### Overview

We propose creating a **Family Expense Management Application** within ServiceNow to streamline the tracking, categorization, and budgeting of family expenses. This solution will provide an easy-to-use, automated platform for families to log, view, and analyze their expenses with real-time insights, mobile access, and alerts.

### Benefits of the Proposed Solution

1. **Transparency:** Every family member knows where the money is going and who is spending what.
2. **Control:** Budgeting features and alerts ensure that family spending stays within limits.
3. **Efficiency:** Automated processes save time, from expense calculation to reporting.
4. **Accountability:** Each family member is accountable for their own spending.
5. **Flexibility:** Mobile access allows for seamless expense entry from anywhere

## 4.3 Solution Architecture

- Setting up ServiceNow Instance
- Creation of New Update Set
- Creation of Table
- Creation of Table(Daily Expenses)
- Creation of Relationship
- Configuring Related List on Family Expenses
- Creation of Business Rules
- Configure the Relationship
- Conclusion

## 5. Project Planning & Scheduling

Phase	Estimated Time
Setting Up Instance	1 hour
Update Set Creation	30 mins
Table Creations	2 hours
Relationships & Lists	1.5 hours
Business Rules	2 hours
Final Testing & Conclusion	1 hour

## 6. Functional and Performance Testing

### 6.1 Performance Testing

- **Data Load Test:** Handle 10,000+ expense records without slowdown.
- **Concurrent Users Test:** Support 20–50 users logging expenses simultaneously.
- **Report Generation Speed:** Generate full-year reports in under 5 seconds.
- **Form Load Time:** Expense entry form loads in less than 2 seconds.
- **API Performance:** Expense data API responds within 1 second (if used).
- **Database Query Optimization:** Monthly summaries fetch in under 1 second.

### 6.2 Functional Testing

- Check if users can **add expenses** with date, category, amount, and user.
- Validate that **negative or zero amounts** are rejected with error messages.
- Ensure **only predefined categories** can be selected.
- Verify **automatic monthly totals** are updated when new expenses are added.
- Test **budget threshold alerts** when expenses exceed set limits.
- Confirm **monthly/annual reports** generate with accurate data.

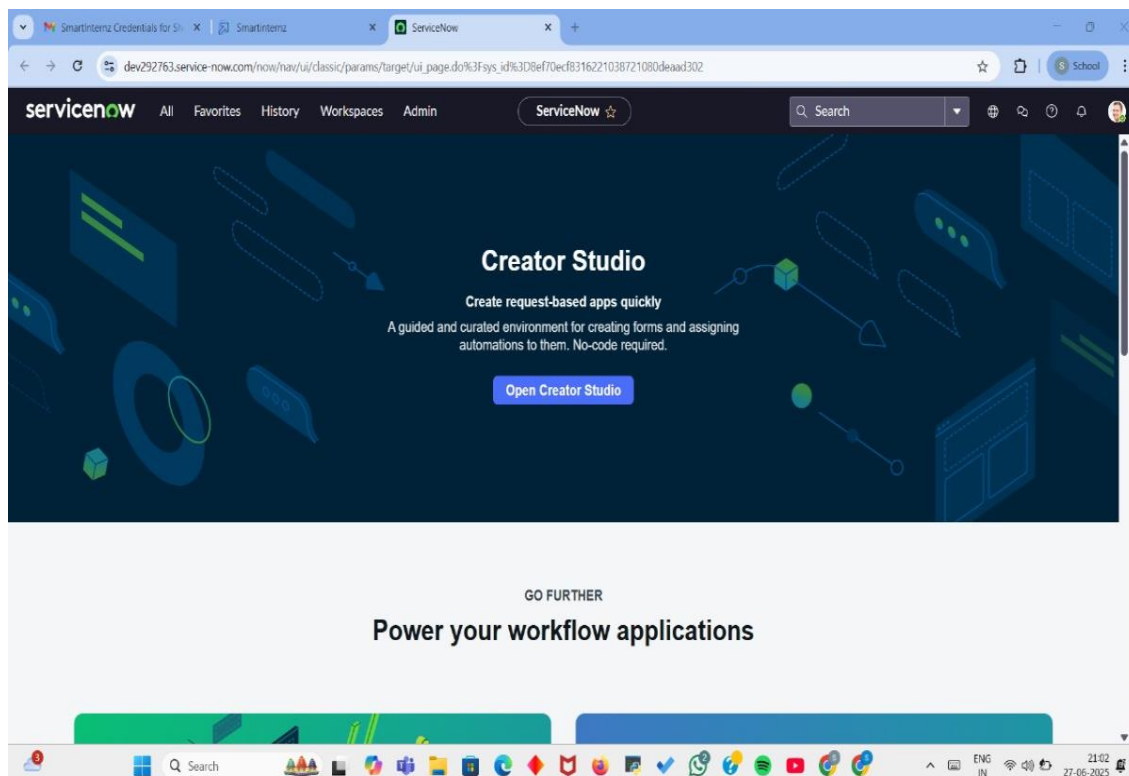
- Validate **multi-user expense logging** shows correct attribution.
- Check **edit/delete actions** update totals and reports instantly.

## 7. Results

### 7.1 Output Screenshots

- Setting up ServiceNow Instance
- Creation of New Update Set
- Creation of Table
- Creation of Table (Daily Expenses)
- Creation of Relationship
- Configuring Related List on Family Expenses
- Creation of Business Rules
- Configure the Relationship
- Conclusion

### Setting up ServiceNow Instance





## Creation of New Update Set

The screenshot shows the 'Create New Update Set' form in ServiceNow. The browser address bar displays the URL: `dev292763.service-now.com/now/nav/ui/classic/params/target/sys_update_set.do%3Fsys_id%3D-1%26sys_is_list%3Dtrue%26sys_target%3Dsys_update_set%26sysparm_checked_items...`. The ServiceNow header includes the logo, navigation tabs (All, Favorites, History, Workspaces), and a search bar. The page title is 'Update Set - Create New Update Set'. The form fields are as follows:

- Name:** Family Expenses
- State:** In progress (dropdown)
- Parent:** (lookup field)
- Release date:** (calendar icon)
- Description:** (text area)
- Application:** Global

Buttons at the bottom include 'Submit' and 'Submit and Make Current'. The Windows taskbar at the bottom shows the time as 21:05 on 27-06-2025.

## Creation of Tables

The screenshot shows the 'Table - New Record' form in ServiceNow. The browser address bar displays the URL: `dev292763.service-now.com/now/nav/ui/classic/params/target/sys_db_object.do%3Fsys_id%3D-1%26sys_is_list%3Dtrue%26sys_target%3Dsys_db_object%26sysparm_checked_items...`. The ServiceNow header includes the logo, navigation tabs (All, Favorites, History, Workspaces, Admin), and a search bar. The page title is 'Table - New Record'. A yellow banner states: 'ServiceNow recommends creating custom tables in scoped applications. To learn more about creating scoped applications, click [here](#).' Below this, a blue box explains: 'A table is a collection of records in the database. Each record corresponds to a row in a table, and each field on a record corresponds to a column on that table. Applications use tables and records to manage data and processes. [More Info](#)'. The form fields are as follows:

- Label:** Daily Expenses
- Name:** u\_daily\_expenses
- Extends table:** (lookup field)
- Application:** Global
- Create module:** ☒
- Create mobile module:** ☒ (Application Menu)
- Add module to menu:** Family Expenditure (dropdown)

Buttons at the bottom include 'Submit' and 'Cancel'. Below the form, there are tabs for 'Columns', 'Controls', and 'Application Access'. The 'Columns' tab is active, showing a 'Table Columns' section with a search bar and a table of 'Dictionary Entries'.

Column label	Type	Reference	Max length	Default value	Display
Insert a new row...					

Buttons at the bottom include 'Submit' and 'Cancel'. Below the form, there are tabs for 'Columns', 'Controls', and 'Application Access'. The 'Columns' tab is active, showing a 'Table Columns' section with a search bar and a table of 'Dictionary Entries'.

Related Links  
[Track in Update Sets](#)

The Windows taskbar at the bottom shows the time as 21:20 on 27-06-2025.

servicenow

Table - New Record

Table: New record View: Tables

A table is a collection of records in the database. Each record corresponds to a row in a table, and each field on a record corresponds to a column on that table. Applications use tables and records to manage data and processes. [More Info](#)

\* Label: Family Expenses

\* Name: u\_st\_family\_expenses

Application: Global

Remote Table: ☒

Create module: ☒

Create mobile module: ☒

Add module to menu: -- Create new --

New menu name: Family Expenditure

Columns Controls Application Access

Table Columns for text Search

Dictionary Entries

Column label	Type	Reference	Max length	Default value	Display
Insert a new row...					

Submit Cancel

Related Links

## Creation of Relationship

servicenow

Relationship - Daily Expenses

Relationship: Daily Expenses

Name: Daily Expenses

Advanced: ☐

Application: Global

Applies to table: Family Expenses [u\_st\_family\_expenses]

Queries from table: -- None --

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

```

1 (function refinequery(current, parent) {
2
3
4 // Add your code here, such as current.addQuery(field, value);
5
6 current.addQuery('u_date', parent.u_date);
7
8 current.query();
9
10
11 })(current, parent);

```

Update Delete

Related Links

Run Point Scan

## Creation of Business Rules

The screenshot shows the 'New Record | Business Rule' form in ServiceNow. The form is titled 'Business Rule - New Record' and includes a 'Submit' button. The 'Name' field is 'Family Expenses BR' and the 'Table' is 'Daily Expenses [u\_daily\_expenses]'. The 'Application' is 'Global'. The 'Active' checkbox is checked, and the 'Advanced' checkbox is also checked. Below the form, there is a section for 'When to run' with tabs for 'When to run', 'Actions', and 'Advanced'. The 'When to run' tab is selected, showing options for 'When' (before), 'Order' (100), and 'Filter Conditions'. The 'Filter Conditions' section has buttons for 'Add Filter Condition' and 'Add "OR" Clause'. Below these are dropdowns for 'choose field', 'oper', and 'value'. The 'Role conditions' section has a link icon. The 'When to run' section also has checkboxes for 'Insert', 'Update', 'Delete', and 'Query', with 'Insert' and 'Update' checked.

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

Table: Daily Expenses [u\_daily\_expenses]

Application: Global

Active: ☒

Advanced: ☒

When to run: before

Order: 100

Filter Conditions: Add Filter Condition Add "OR" Clause

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

Role conditions: [Role conditions](#)

Insert: ☒ Update: ☒ Delete: ☐ Query: ☐

The screenshot shows the 'Family Expenses BR' business rule script editor in ServiceNow. The 'Condition' field is empty. The 'Script' section has a checkbox for 'Turn on ECMAScript 2021 (ES12) mode' which is checked. The script is written in JavaScript and is as follows:

```
1 (function executeRule(current, previous /*null when async*/) {
2
3
4 var FamilyExpenses = new GlideRecord('u_family_expenses');
5
6 FamilyExpenses.addQuery('u_date',current.u_date);
7
8 FamilyExpenses.query();
9
10 if(FamilyExpenses.next())
11 {
12
13
14 FamilyExpenses.u_amount += current.u_expense;
15
16 FamilyExpenses.u_expense_details += ">" + current.u_comments + ">" + "Rs." + current.u_expenses + ">";
17
18 FamilyExpenses.update();
19
20 }
21
22 else
23
```

Update Delete

## 8. Advantages & Disadvantages

### ✓ Advantages

Here are the advantages of calculating family expenses using **ServiceNow** in simple points:

1. **Centralized Expense Tracking:** Keep all your family expenses in one place, making it easy to access and review anytime.
2. **Automation:** Set up automatic logging for recurring expenses (like rent or bills) to save time.
3. **Custom Reports:** Create detailed reports to track your spending patterns and compare them over time.
4. **Budget Alerts:** Get notifications when you reach certain spending limits or when bills are due.
5. **Custom Categories:** Organize expenses into categories (groceries, bills, entertainment) for better tracking.
6. **Collaboration:** Share financial data with family members to keep everyone on the same page.
7. **Audit Trail:** Track who made changes to your expense records for transparency.
8. **Secure Data:** Your financial data is protected with enterprise-level security and privacy controls.
9. **Integration with Other Tools:** Link your bank accounts or financial software to import transactions automatically.
10. **Scalability:** As your financial situation grows, ServiceNow can handle more complex expenses like investments or savings.

This makes managing family expenses more organized, efficient, and secure.

### ✗ Disadvantages

Here are the disadvantages of using **ServiceNow** to calculate family expenses in simple points:

1. **Complex Setup:** ServiceNow is designed for businesses, so setting it up for personal use can be complicated and time-consuming.
2. **Overkill for Simple Needs:** It's a powerful tool, but it may be too complex for basic family budgeting and expense tracking.
3. **Cost:** ServiceNow is typically not free; it may require a paid subscription, which could be unnecessary for personal finances.
4. **Steep Learning Curve:** It takes time to learn how to use all of ServiceNow's features, which may not be worth it for just managing family expenses.
5. **Limited Personal Finance Features:** It doesn't have many built-in features specifically for budgeting, savings goals, or debt tracking.
6. **Requires Customization:** You'll need to customize it for your needs (like adding expense categories), which can be a hassle.
7. **No Mobile App:** ServiceNow isn't as mobile-friendly as other personal finance tools, so it may be harder to manage expenses on the go.
8. **Not Built for Family Use:** It's more suited for businesses, so it might lack features that cater to family-specific needs like shared access or easy family budgeting.
9. **Maintenance:** Ongoing updates and maintenance can be more technical, especially if you're not familiar with IT tools.
10. **Data Privacy Concerns:** Even though it's secure, storing personal family expenses on a platform originally designed for businesses could raise privacy concerns.

In short, **ServiceNow** can be over-complicated and expensive for managing family expenses, especially when there are simpler, cheaper alternatives out there.

## 9. Conclusion

In conclusion, implementing the "Calculation of Family Expenses Using ServiceNow" project offers numerous benefits to the household. By leveraging the robust capabilities of ServiceNow, family members can efficiently submit, track, and manage their expenses in a centralized and streamlined manner.

## 10. Future Scope

- Add analytics dashboards for performance tracking
- Send automatic notifications for admission status
- Create a mobile-friendly version
- Role-based permissions for teachers and students

## 11. Appendix

- Source Code: No external code; used ServiceNow platform
- Dataset Link: Not applicable
- GitHub Link: <https://github.com/Keerthi-28-11>