

Calculating family expenses using Servicenow

TEAM ID : NM2025TMID05646

TEAM SIZE :4

TEAM LEADER : Rasika .P

TEAM MEMBER : Ishwaryah J

TEAM MEMBER : Janani shri K

TEAM MEMBER : Shobana S

Ideation phase

1. Define Problem Statement

The "Define Problem Statement" stage is crucial for validating the project's necessity by clearly articulating the user pain points and the impact of the current state. For the Family Expense Management System, the core problem lies in the absence of a unified, real-time method for tracking household spending. This deficiency subjects the **Household Budget Manager** to unnecessary stress and error-prone manual reconciliation. The current, disjointed process—relying on paper receipts, verbal reports, and separate spreadsheets—makes accurate, timely financial control impossible, leading to frequent budget overruns and delayed corrective action.

The defined problem, therefore, is rooted in **data opacity and manual inefficiency**. We established a clear Customer Problem Statement: "As a Household Budget Manager, I'm trying to see how much money my family has spent this month across all categories, but I have to manually collect and reconcile receipts and spreadsheet entries from all family members because there is no single system for all family members to quickly log their spending, which makes me feel anxious about budget overruns and frustrated with the time spent on manual data entry." The solution—the ServiceNow application—must directly solve this by providing a simple, mobile-accessible logging mechanism and an immediate, automated calculation of total spend against the budget.

2. Empathy Map Canvas

The Empathy Map Canvas moves beyond the abstract problem to focus specifically on the user's emotional and practical experience. By adopting the persona of the **Household Budget Manager**, we gain insight into their daily actions and underlying motivations. The user **Says** things that indicate a lack of control ("Where did all that money go?") and **Does** tedious tasks like physically collecting receipts and manually updating data. More importantly, we uncover what they **Think & Feel**: they are frequently **stressed, anxious, and overwhelmed** by the sheer administrative load of household finance management, constantly wondering if the family is over budget.

By understanding what the user **Sees** (cluttered data, confusing spreadsheets) and **Hears** (low balance alerts, questions from family), we solidify the core **Pain** points: manual data entry fatigue, lack of real-time tracking, and inconsistent expense categorization. This empathic process ensures the final design is user-centric, prioritizing a simple mobile interface and immediate visual feedback. The ultimate **Gain** the project promises is **peace of mind**, financial accuracy, and the confidence that comes from having a single source of truth for all family expenditure.

3. Brainstorming & Idea Prioritization

The final step in the ideation phase transforms the problems and user insights into actionable technical features. This involves brainstorming a comprehensive list of capabilities for the **Family Expense Management System (FEMS)** and then prioritizing them using the **MoSCoW** method (Must Have, Should Have, Could Have, Won't Have). The highest priority, or "**Must Have**", is foundational stability: creating the necessary **Custom Tables** (`u_daily_expense` and `u_family_budget`) to structure the financial data, along with basic **Custom Forms** for data input.

The most critical functional piece is the **Total Calculation Logic**, which must be implemented as a server-side **Business Rule** in ServiceNow. This rule ensures that every time a new expense is entered, the corresponding budget record is updated instantly, fulfilling the "real-time" requirement of the project. Features like **Reporting Dashboards** for visual analysis and **Automated Budget Alerts** (using Flow Designer) are ranked as "**Should Have**" or "**Could Have**". They are highly valuable for the end-user experience but can be developed after the core data structure and calculation logic are proven stable, thereby ensuring a functional and reliable Minimum Viable Product (MVP) is delivered first.

[System Flowchart: Family Expense Management in ServiceNow](#)

This diagram illustrates the flow from a family member logging an expense to the system automatically updating the budget and alerting the manager.

Flow Description

1. **User Input (Start):** A family member incurs an expense (e.g., buys groceries) and accesses the **FEMS Custom Form** on the ServiceNow mobile or web interface.

- They fill in the details: **Expense Amount**, **Date**, and **Expense Category** (e.g., 'Groceries', 'Utilities').
2. **Data Storage:**
- The record is immediately saved to the custom table: **Daily Expenses** (**u_daily_expense**). This table serves as the raw ledger of all transactions.
3. **Automated Calculation (Core Logic):**
- An **After Business Rule** on the `Daily Expenses` table is instantly triggered upon insertion or update.
 - **Action:** This Business Rule executes a script to query all daily expense records for the current month and category. It calculates the **Total Actual Spend** for that category.
4. **Budget Reconciliation:**
- The Business Rule updates the corresponding parent record in the **Family Budget** (**u_family_budget**) table. This central table holds the predefined monthly budget limits and the running total of actual expenses.
5. **Automated Notification (Flow Designer):**
- The update to the `Family Budget` table triggers a **Flow Designer** automation.
 - **Condition Check:** The Flow checks if the **Total Actual Spend** for the category has exceeded a critical threshold (e.g., 90% of the Allocated Budget).
 - **Action:** If the threshold is met, the Flow generates an automatic **Notification** (via email or a ServiceNow alert) to the Household Budget Manager.
6. **Real-Time Reporting (End):** The Budget Manager views the **FEMS Dashboard**, which uses data from the `Family Budget` table to display the *Budget vs. Actual Spend* variance in real-time.