

## Phase 5: Performance Testing

DATE	31 OCTOBER 2025
TEAM ID	NM2025TMID06233
PROJECT NAME	Calculating Family Expenses using ServiceNow

### 5.1 Testing Objective:

The objective of testing is to ensure that the Family Expense Tracker in ServiceNow works correctly, calculates totals accurately, and updates records in real-time without errors.

### 5.2 Types of Testing Performed:

Testing Type	Description
Unit Testing	Tested each table and field individually
Functional Testing	Verified all features work as expected
Integration Testing	Checked link between Daily Expense & Family Expense tables
Validation Testing	Ensured mandatory fields & correct data entry
Performance Testing	Verified system updated totals instantly
User Acceptance Testing (UAT)	Ensured the solution is user-friendly & meets objectives

### 5.3 Test Cases:

Test Case ID	Test Description	Input	Expected Output	Result
TC-01	Add daily expense record	Dad → 1000	Record saved	Passed
TC-02	Add another daily expense	Dad → 2000	Total = 3000	Passed
TC-03	Add expense for another member	Mom → 500	Record saved	Passed
TC-04	Check total for Mom	Mom → 500	Correct total	Passed

TC-05	Check auto-generated number	Save record	Auto-ID generated	Passed
TC-06	Invalid amount entry	Blank amount	Error shown	Passed
TC-07	Submit without date	Blank date	Error shown	Passed

## 5.4 Performance Metrics:

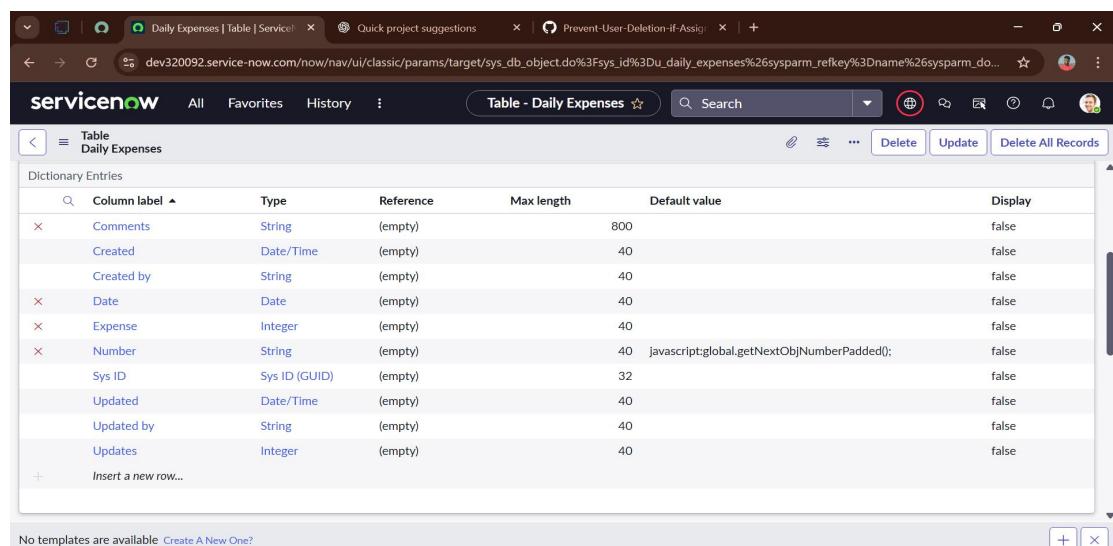
Parameter	Result
Record Save Time	Less than a second
Total Calculation Time	Instant update
Page Load Speed	Smooth
System Load Handling	Stable for multiple entries

## 5.5 Observations & Findings:

- Daily expenses correctly update Family Total
- Business Rule triggers worked properly
- Auto-Numbering works
- Reference field links correctly
- No performance lag
- Data remains consistent after multiple entries

## 5.6 Pictures for references:

### 5.6.1 Daily Expense Table:



The screenshot shows the ServiceNow dictionary interface for the 'Daily Expenses' table. The title bar reads 'Table - Daily Expenses'. The main area displays a list of columns with their properties:

Column label	Type	Reference	Max length	Default value	Display
Comments	String	(empty)	800		false
Created	Date/Time	(empty)	40		false
Created by	String	(empty)	40		false
Date	Date	(empty)	40		false
Expense	Integer	(empty)	40		false
Number	String	(empty)	40	javascript:global.getNextObjNumberPadded();	false
Sys ID	Sys ID (GUID)	(empty)	32		false
Updated	Date/Time	(empty)	40		false
Updated by	String	(empty)	40		false
Updates	Integer	(empty)	40		false

At the bottom, there is a note: 'No templates are available Create A New One?' with a '+' button.

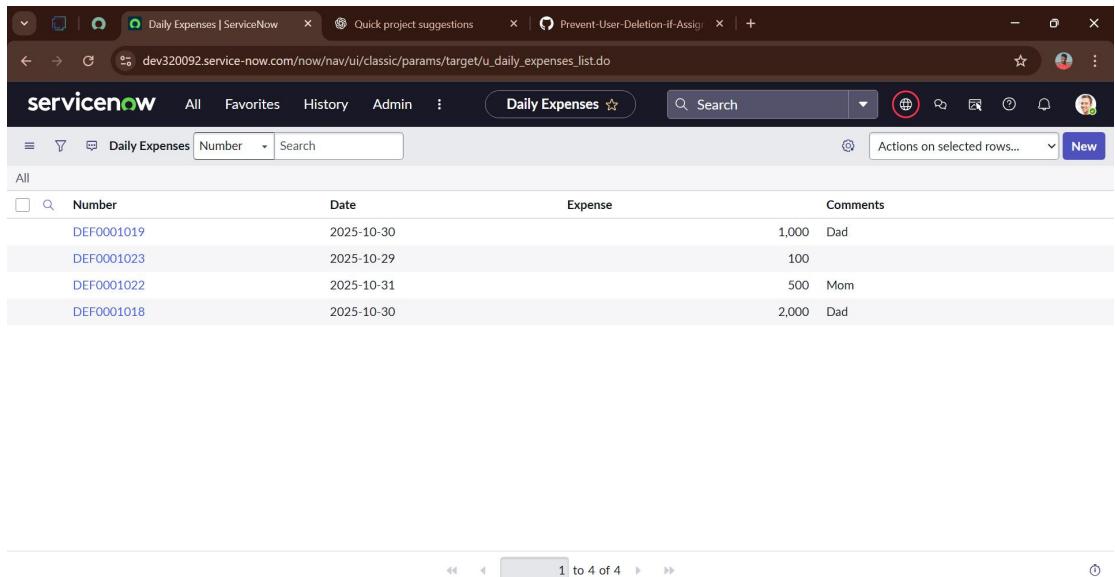
## 5.6.2 Family Expense Table:

The screenshot shows the ServiceNow interface for configuring the 'Family Expenses' table. The top navigation bar includes links for 'Family Expenses | Table | Service', 'Quick project suggestions', 'Prevent-User-Deletion-if-Assigned', and a search bar. Below the header, the title 'Table - Family Expenses' is displayed, along with buttons for 'Delete', 'Update', and 'Delete All Records'. A dictionary entry for 'Family Expenses' is shown, listing columns with their labels, types, references, maximum lengths, default values, and display settings. The columns include: Amount (Integer), Created (Date/Time), Created by (String), Date (Date), Expense Details (String), Number (String), Sys ID (Sys ID (GUID)), Updated (Date/Time), Updated by (String), and Updates (Integer). The 'Default value' column contains various values such as '(empty)', '40', 'false', and 'javascript:getNextObjNumberPadded();'. An 'Insert a new row...' button is located at the bottom left of the table view. A message at the bottom states 'No templates are available' with a 'Create A New One?' link.

## 5.6.3 Business Rules Script:

The screenshot shows the ServiceNow interface for configuring a business rule named 'Business Rule - Family Expenses BR'. The top navigation bar includes links for 'Family Expenses BR | Business Rule', 'Quick project suggestions', 'Prevent-User-Deletion-if-Assigned', and a search bar. Below the header, the title 'Business Rule - Family Expenses BR' is displayed, along with buttons for 'Update' and 'Delete'. The main area contains a code editor with a syntax-highlighted script. The script uses GlideRecord objects to interact with the 'u\_family\_expenses' table. It defines a function 'executeRule' that iterates through records ('next') and adds them to a current record ('addQuery'). If the current record has a next record, it adds the current record's amount to the next record's amount and updates the current record ('update'). If there is no next record, it creates a new record ('NewFamilyExpenses') with the current record's details and inserts it ('insert'). The code ends with a call to the function, passing 'current' and 'previous' as arguments. The code editor includes line numbers from 1 to 20.

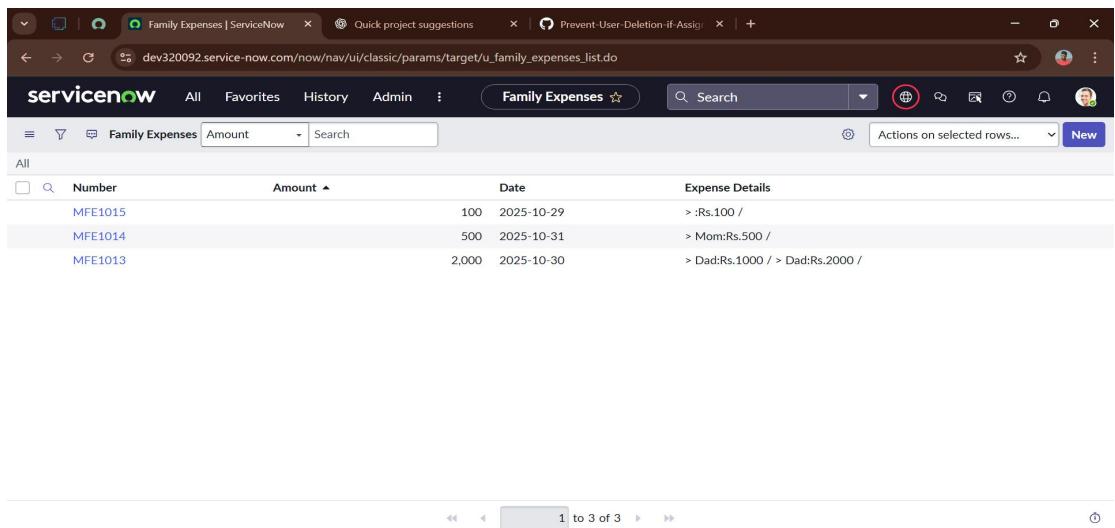
#### 5.6.4 Input in Daily Expenses



A screenshot of a ServiceNow web interface titled "Daily Expenses". The page shows a list of four expense entries. The columns are Number, Date, Expense, and Comments. The entries are:

Number	Date	Expense	Comments
DEF0001019	2025-10-30	1,000	Dad
DEF0001023	2025-10-29	100	
DEF0001022	2025-10-31	500	Mom
DEF0001018	2025-10-30	2,000	Dad

#### 5.6.5 Output in Family Expenses:



A screenshot of a ServiceNow web interface titled "Family Expenses". The page shows a list of three expense entries. The columns are Number, Amount, Date, and Expense Details. The entries are:

Number	Amount	Date	Expense Details
MFE1015	100	2025-10-29	> :Rs.100 /
MFE1014	500	2025-10-31	> Mom:Rs.500 /
MFE1013	2,000	2025-10-30	> Dad:Rs.1000 / > Dad:Rs.2000 /