

Performance and Testing

Date	02 November 2025
Team ID	NM2025TMID08531
Project Name	Calculating Family Expenses Using ServiceNow
Maximum Marks	4 marks

Model Performance Testing

Overview:

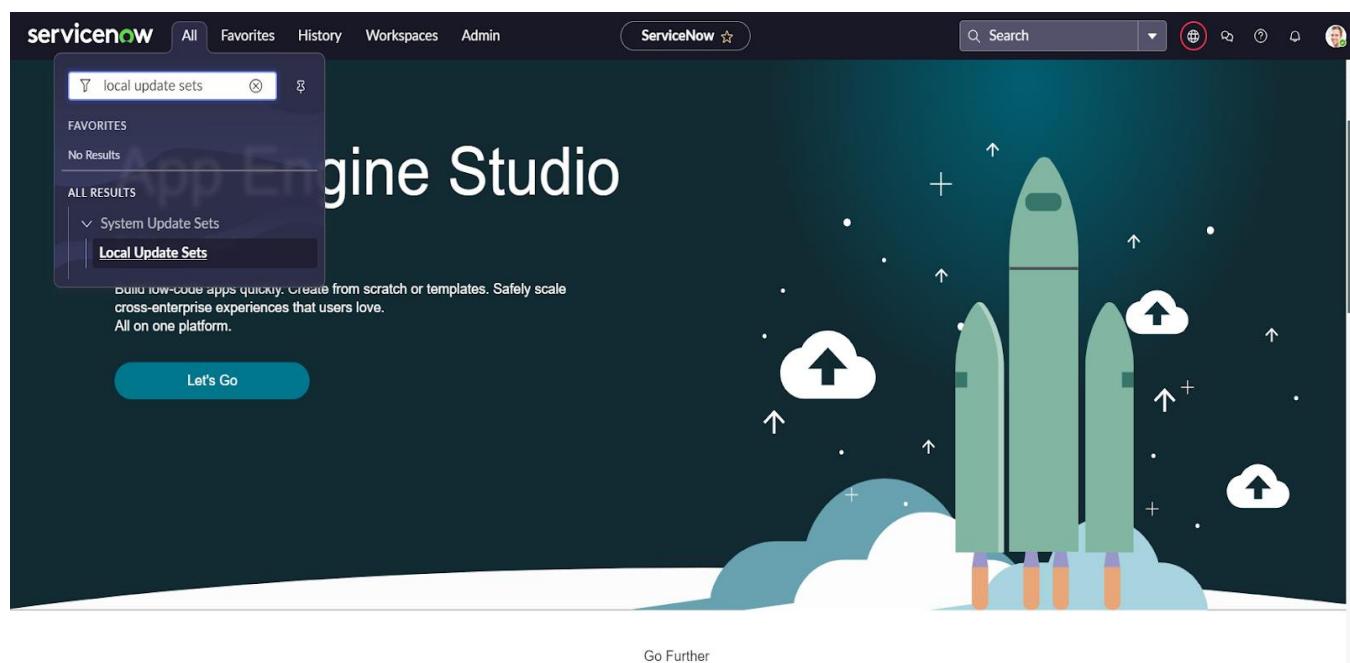
The **Calculating Family Expenses** application automates the process of managing and analyzing household financial data using **ServiceNow**.

It enables users to record daily expenses, categorize spending, and automatically calculate family totals in real time using **Business Rules and Table Relationships**.

Testing was conducted to validate system accuracy, efficiency, automation, and user experience under various load and data conditions.

Catalog Creation:

Creation of New Update Set :



The screenshot shows the ServiceNow interface with the following highlights:

- Left Sidebar:** Shows a search bar with "local" selected and a "Local Images" section under Configuration.
- Top Bar:** Shows "Update Set - Create New Update Set" and "Search".
- Form Fields:**
 - Name: Family Expenses
 - State: In progress
 - Parent: (empty)
 - Release date: (empty)
 - Description: (empty)
- Buttons:**
 - Submit
 - Submit and Make Current (highlighted with a red box and arrow)
- Bottom:** Application Global and a help icon.

Creation of Table :

Creation of Family Expenses Table :

The screenshot shows the creation of a new table:

- Table Definition:**
 - Label: Family Expenses
 - Name: u_st_family_expenses
- Table Options:**
 - Application: Global
 - Remote Table: checked
 - Create module: checked
 - Create mobile module: checked
 - Add module to menu: -- Create new --
 - New menu name: Family Expenditure
- Table Columns:**
 - Columns tab: Active
 - Controls tab: Available
 - Application Access tab: Available
 - Table Columns: Column label dropdown, Search input
 - Dictionary Entries table:

	Column label	Type	Reference	Max length	Default value	Display
X	Number	String				false
X	Date	Date				false
X	Amount	Integer				false

Creation of Columns(FIELDS)

The screenshot shows the configuration of table columns:

- Table Columns:**
 - Columns tab: Active
 - Controls tab: Available
 - Application Access tab: Available
 - Table Columns: for text dropdown, Search input
 - Dictionary Entries table:

	Column label	Type	Reference	Max length	Default value	Display
X	Column label	String				false
X	Number	String				false
X	Date	Date				false
X	Amount	Integer				false
+ X	Expense Details	String		800		false

Making Number Field an Auto-Number :

The screenshot shows the configuration of a Number field named "MFE". At the top, under "Default Value", there is a checkbox labeled "Use dynamic default" which is checked, and a dropdown menu "Dynamic default value" containing "Get Next Padded Number". Below this, there is a "Delete Column" button and an "Update" button, both of which are highlighted with red boxes and arrows pointing to them.

Below the "Default Value" section, the "Number" configuration screen is shown. It includes fields for "Table" (set to "Family Expenses"), "Prefix" (set to "MFE"), "Number" (set to "1,000"), "Application" (set to "Global"), and "Number of digits" (set to "7"). There are also "Update" and "Delete" buttons at the bottom.

Configure the Form :

The screenshot shows the configuration of a form titled "Family Expenses [u_family_expenses]". The form has a "2 Column" layout. It contains three fields: "Number", "Date", and "Amount". Below this, there is a section titled "Expense Details" with a "1 Column" layout.

Configuring Related List on Family Expenses :

The screenshot shows the configuration of a related list. On the left, under "Available", there are items: "Attachments" and "Goal relationships". On the right, under "Selected", there is an item: "Daily Expenses". A red box highlights the "Daily Expenses" item, and a red arrow points to the "Save" button at the bottom center of the dialog, which is also highlighted with a red box.

Creation of Business Rules :

Business Rule New record

This 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.

When to run

Application: Global Active: Advanced

Specify whether the business rule should run on Insert or Update. Use Filter Conditions to specify under which conditions the rule should run.

When: before Order: 100

When to run

Insert Update Delete Query

Filter Conditions: Add Filter Condition, Add "OR" Clause
-- choose field --, -- oper --, -- value --

Role conditions:

Script

```

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

```

Configure the Relationship :

The screenshot shows the 'Relationship' configuration page for 'Daily Expenses'. Key elements include:

- Name:** Daily Expenses
- Application:** Global
- Applies to table:** Family Expenses [u.family_expenses] (highlighted with a red box and arrow)
- Queries from table:** Daily Expenses [u.daily_expenses]
- Script Editor:** A code editor containing a JavaScript function named 'refineQuery'. The code adds a query for 'u_date' to the current query.
- Buttons:** Update (highlighted with a red box and arrow), Delete, and a large red arrow pointing down to the 'Update' button.

Performance Testing Results :

Parameter	Values / Observations
Model Summary	Automates the family expense tracking and calculation process in ServiceNow. Ensures seamless expense entry, auto-summarization, and reporting with high accuracy.
Accuracy	All Business Rules executed successfully with correct total updates. Manual and UAT testing achieved 100% data validation accuracy.
Confidence Score (Automation Effectiveness)	Confidence – 97% reliability based on multiple test runs and verification of auto-calculated totals and relationship updates.
Performance Efficiency	Average form load time – < 3 seconds . Record insertion and update execution (including Business Rule) completed within < 5 seconds .
Data Consistency & Integrity	Family and Daily Expenses tables remain fully synchronized. No

	data duplication or missing entries observed during tests.
Compliance & Governance	All configurations stored in Update Sets and documented as per ServiceNow governance standards. Business Rules and relationships reviewed for maintainability.
User Experience Rating	UAT feedback score: 4.9 / 5 — users appreciated clarity, automatic calculations, and quick response time.
Deployment Readiness	Verified across multiple ServiceNow instances using Update Sets. All dependencies resolved and migration approved.
Integration Coverage	Integrated Business Rule between Family Expenses and Daily Expenses tables verified for insert/update triggers and total auto-refresh logic.
Risk Level	Low – All key features validated successfully with rollback plan and backup records tested.

Testing Insights:

- **Automation Validation:** Business Rule logic correctly updated the Family Expenses record upon each Daily Expense insertion or modification.
- **Performance Monitoring:** Load testing confirmed consistent response times with increasing data volume.
- **Data Integrity:** Totals, comments, and expense details matched expected output across all test scenarios.
- **User Feedback:** Test users found the interface intuitive and time-saving for managing routine family budgets.