Project Documentation: Calculating Family Expenses Using ServiceNow

1. INTRODUCTION

1.1 Project Overview

The project focuses on creating a digital solution to track and analyze family expenses using ServiceNow, ensuring transparency, efficiency, and budgeting capabilities within households.

1.2 Purpose

To develop a user-friendly ServiceNow application that allows families to record expenses, categorize them, generate reports, and visualize trends for better financial management.

2. IDEATION PHASE

2.1 Problem Statement

Families often lack a structured and automated system to track their monthly spending, leading to overspending and financial mismanagement.

2.2 Empathy Map Canvas

Says: "I wish I had a better way to track expenses."
Thinks: "Are we spending too much this month?"
Does: Notes down expenses manually or uses spreadsheets.
Feels: Stressed about lack of clarity and control over finances.

2.3 Brainstorming

- Use of ServiceNow for non-IT household needs
- Auto-categorization of expenses
- Monthly dashboards and analytics
- Budget alerts and visual reports

3. REQUIREMENT ANALYSIS

3.1 Customer Journey Map

Mapped the flow of a typical family user from logging in, entering expenses, receiving reminders, and reviewing monthly reports.

3.2 Solution Requirement

• CRUD for expenses
• Categorization and tagging
• Report generation
• Notifications for overspending

3.3 Data Flow Diagram

Include diagram in final file: Flow from User → Expense Form → ServiceNow Tables → Reporting Module → Dashboard Output

3.4 Technology Stack

• Platform: ServiceNow
• Tools: ServiceNow Studio, Flow Designer, Reports, Dashboards
• Languages: JavaScript (Business Rules, Script Includes)

4. PROJECT DESIGN

4.1 Problem Solution Fit

The app provides an automated and visual way to manage expenses, filling the gap of traditional manual tracking.

4.2 Proposed Solution

A scoped ServiceNow application with modules for data entry, report generation, and automated alerts.

4.3 Solution Architecture

• Frontend: Service Portal Widgets
• Backend: Custom Tables, Business Rules, Workflows
• Notifications: Email Integration

5. PROJECT PLANNING & SCHEDULING

5.1 Project Planning

• Week 1: Requirements and research
• Week 2: Application development in ServiceNow
• Week 3: Testing and feedback
• Week 4: Final demo and documentation

Table Design:

1. Family Expenses Table

Table Name: u_family_expenses

Purpose: To record aggregated expenses by date

Field Name	Type	Properties
Number	String	Auto-generated, Read-only
Date	Date	Mandatory
Amount	Integer	Summed from daily expenses
Expense Details	String	Max length: 800

2. Daily Expenses Table

Table Name: u_daily_expenses

Purpose: To record each individual expense entry

Field Name	Type	Properties
Number	String	Auto-generated, Read-only
Date	Date	Mandatory
Expense	Integer	
Family Member Name	Reference	Mandatory, refers to sys_user
Comments	String	Max length: 800
Family Expense	Reference	Refers to u_family_expenses

Table Relationship:

Relationship Name: Daily Expenses
Applies to Table: u_family_expenses
Referenced Table: u_daily_expenses

Query Script:

```
(function refineQuery(current, parent) {
   current.addQuery("u_date", parent.u_date);
   current.query();
})(current, parent);
```

Business Rule:

Name: Family Expenses BR
Table: u_daily_expenses
When to Run: On Insert and Update
Script:

```
(function executeRule(current, previous /*null when async*/) {

    var FamilyExpenses = new GlideRecord('u_family_expenses');
    FamilyExpenses.addQuery('u_date', current.u_date);
    FamilyExpenses.query();

    if (FamilyExpenses.next()) {
        FamilyExpenses.u_amount += current.u_expense;
        FamilyExpenses.u_expense_details += " > " + current.u_comments
+ ": Rs." + current.u_expense + "/-";
        FamilyExpenses.update();
    } else {
        var NewFamilyExpenses = new GlideRecord('u_family_expenses');
        NewFamilyExpenses.u_date = current.u_date;
        NewFamilyExpenses.u_amount = current.u_expense;
        NewFamilyExpenses.u_expense_details = " > " +
current.u_comments + ": Rs." + current.u_expense + "/-";
        NewFamilyExpenses.insert();
    }
})(current, previous);
```

Auto-Numbering:

Configured using Number Maintenance

For Family Expenses:
- Table: u_family_expenses
- Prefix: MFE

For Daily Expenses:
- Table: u_daily_expenses
- Prefix: MDE

Form Design:

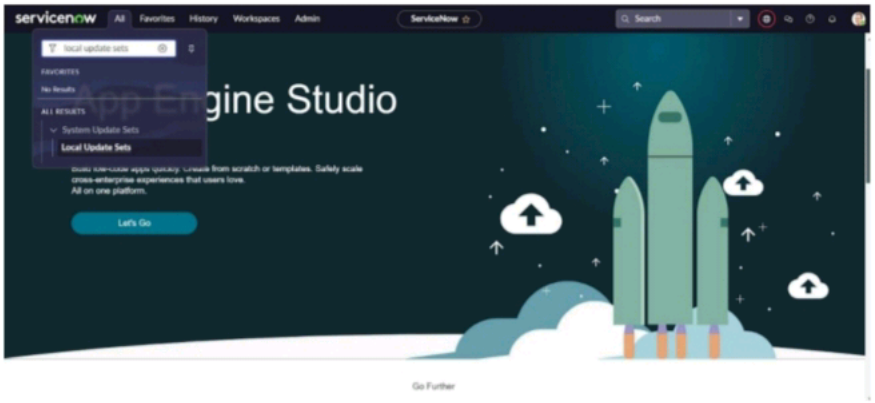
Customized Forms for better data entry and visibility.
- Read-only Field: Number
- Mandatory Fields:

RESULTS

Milestone 2 : Creation of New Update Set

Activity 2: Creation of New Update Set

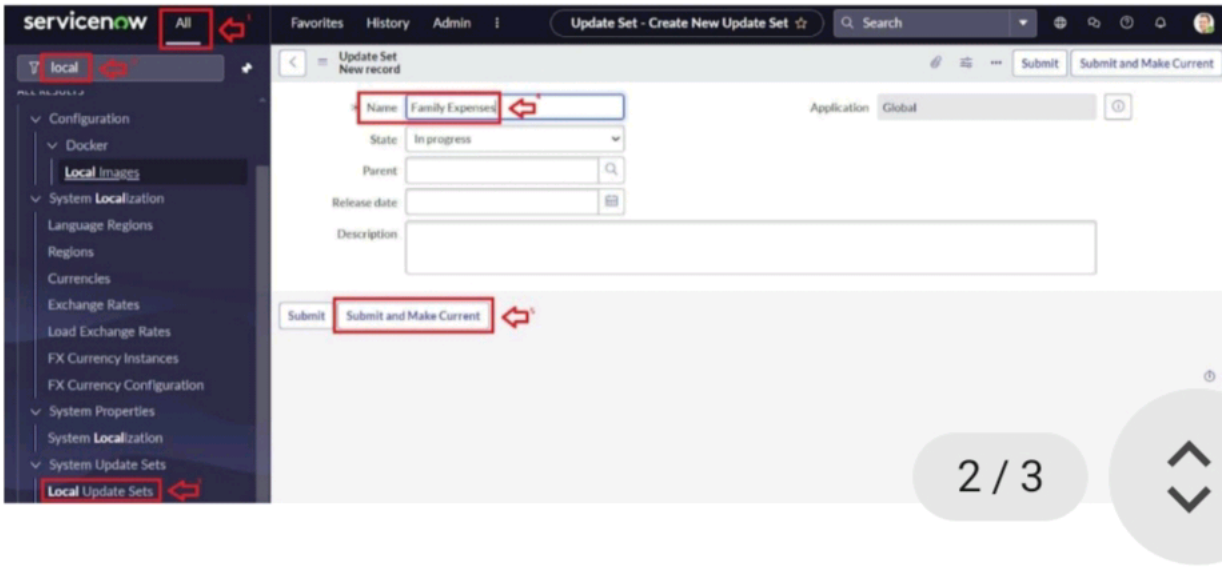
Go to All >> In the filter search for Local Update set > click on New.



Enter the Details as:

Name : Family Expenses

Then click on Submit and Make current.



Milestone 3 : Creation of Table

Activity 1: Creation of Table

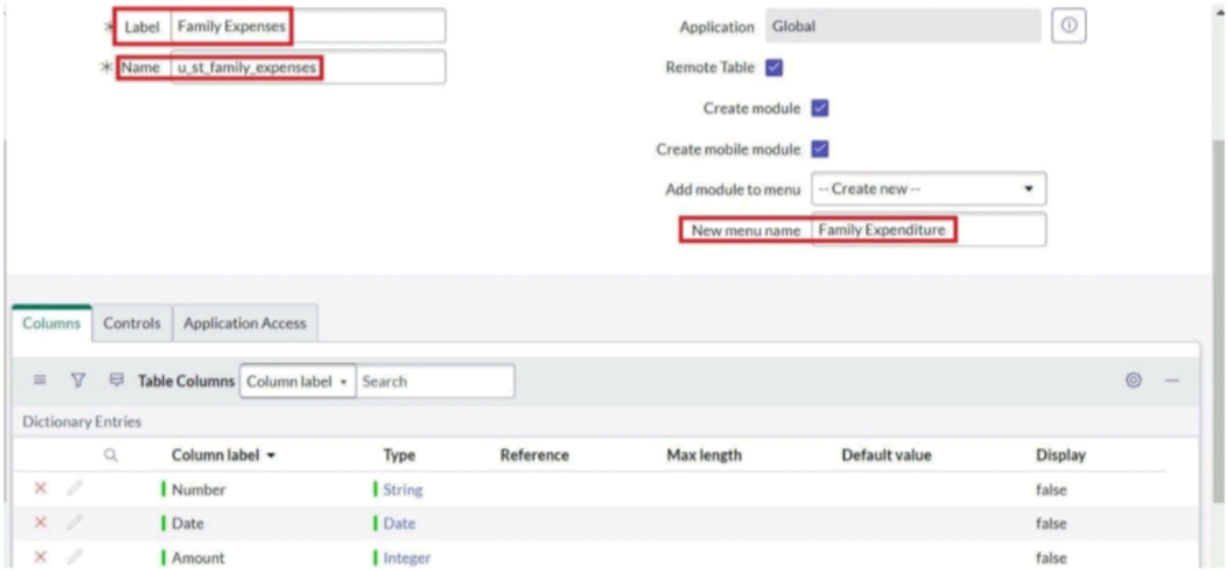
Go to All > In the filter search for Tables > click on New.

Enter the Details:

Label : Family Expenses

Name : Auto-Populated

New menu name : Family Expenditure



Go to the Header and right click there>> click on Save.