

**CALCULATING FAMILY EXPENSES USING
SERVICE NOW**

NAAN MUDALVAN PROJECT

TEAM ID:

NM2025TMID18629

Team Leader : AJITH KUMAR

Team member : KISHORE V

Team member : LOGESH S

Team member : PANNEERSELVAM V

JOHN BOSCO ARTS AND SCIENCE COLLEGE

INDEX

1. Introduction
2. Abstract
3. Problemstatement
4. Solution
5. Practicaluse
6. Knowledgegained
7. Milestone1:Settingupservicenowinstance
8. Milestone2:Creationofnewupdate set
9. Milestone3:Creationoftablefamilyexpenses
- 10.Milestone 4: Creation of table daily expenses
- 11.Milestone 5: Creation of relationship
- 12.Milestone6:ConfiguringRelatedlistonfamily expenses
- 13.Milestone7:CreationofBusinessrules
- 14.Milestone8:Configuretherelationship
- 15.Conclusion

INTRODUCTION

Managing family expenses is one of the most important aspects of financial planning, yet it is often overlooked or handled in unstructured ways like handwritten notes or spreadsheets. Families frequently face challenges such as overspending, lack of budget control, and difficulty in analyzing where their money goes. To address these issues, technology can play a vital role by providing structured, automated, and user-friendly solutions.

This project, “Calculating Family Expenses Using ServiceNow,” aims to create a digital system that helps track, categorize, and manage family expenses efficiently. ServiceNow, known for its robust workflow automation and application development capabilities, provides an excellent platform for building such a system. By leveraging ServiceNow, this project transforms expense management into a more streamlined process with features like expense categorization, daily tracking, budget limits, reporting, and automated business rules.

The system will not only simplify financial management for families but also provide real-time insights into spending patterns, allowing better decision-making. With its scalability and flexibility, the solution can be expanded to suit different family structures and even adapted for small business expense management. Ultimately, the project showcases how a powerful enterprise platform like ServiceNow can be applied beyond IT workflows to solve everyday problems in an innovative and practical way.

ABSTRACT

Expense management plays a crucial role in maintaining financial stability within families, yet traditional methods such as manual tracking or spreadsheets often lead to errors, lack of visibility, and poor decision-making. To address this challenge, this project focuses on designing and implementing a Family Expense Management System using ServiceNow.

The application leverages ServiceNow's powerful low-code development environment to create structured tables, relationships, and automated workflows for recording and analyzing expenses. Key features include categorizing expenses (such as food, utilities, and transport), maintaining family member details, linking daily transactions, and applying business rules to validate data and automate calculations. By configuring related lists and creating meaningful reports, the system provides users with a clear overview of their financial habits and budget limits.

The proposed solution not only simplifies day-to-day expense tracking but also provides real-time insights into spending patterns, enabling families to make smarter financial decisions. Furthermore, the project demonstrates the versatility of ServiceNow beyond traditional IT service management, showcasing its potential in solving practical, real-world problems.

PROBLEM STATEMENT

Managing household expenses is often a difficult and time-consuming task for families. Most families rely on manual methods such as notebooks, receipts, or spreadsheets to track their daily spending. These methods come with several challenges:

- Lack of real-time tracking of expenses.
- Difficulty in categorizing and consolidating expenses like food, utilities, rent, and transportation.
- Limited ability to analyze spending patterns or generate reports.
- High chances of errors due to manual data entry.
- No automation to alert families about overspending or exceeding budgets.

As a result, families often lose visibility into their financial flow, making it harder to control budgets or make informed financial decisions. This creates the need for a systematic, automated, and user-friendly solution to manage and calculate family expenses efficiently.

SOLUTION

The proposed solution is to build a Family Expense Management System on the ServiceNow platform. ServiceNow, being a robust low-code/no-code platform, provides all the tools required to create structured applications without needing extensive programming knowledge.

The solution involves:

- **Creating custom tables** to store family member details and daily expenses.
- **Defining relationships** between family members and their respective expenses for easy tracking.
- **Configuring related lists** so that expenses linked to each family member can be viewed in one place.
- **Implementing business rules** to automate calculations, validate entries, and ensure data accuracy.
- **Generating reports and dashboards** to visualize monthly/annual spending and highlight budget deviations.
- **Using update sets** to track and migrate customizations, ensuring proper version control.

PRACTICAL USE

The Family Expense Management System built on ServiceNow helps families track, categorize, and analyze their expenses in a structured way. It simplifies budgeting, reduces errors from manual tracking, and provides real-time insights into spending patterns. Families can use it to set limits, monitor monthly expenses, and generate reports for better financial decisions. Beyond households, the same system can be adapted for small businesses to manage cash flow and daily transactions effectively, proving the versatility of ServiceNow in solving practical, non-IT problems.

KNOWLEDGE GAINED

- Learned how to setup and configure a ServiceNow developer instance for building applications.
- Understood the importance of update sets for tracking and migrating customizations.
- Gained practical skills in creating custom tables and defining fields to store structured data.
- Learned how to establish relationships between tables for linked data management.
- Practiced configuring related lists for easier navigation and record visibility.
- Understood how to create and apply business rules for automation and validations.
- Gained insights into data modeling and database concepts within ServiceNow.
- Learned how to generate reports and dashboards for real-time analysis.
- Understood how ServiceNow can be applied to non-IT use cases like family expense tracking.
- Improved overall knowledge of workflow automation and low-code development.

MILESTONE1:SETTINGUPTHESERVICE NOW INSTANCE

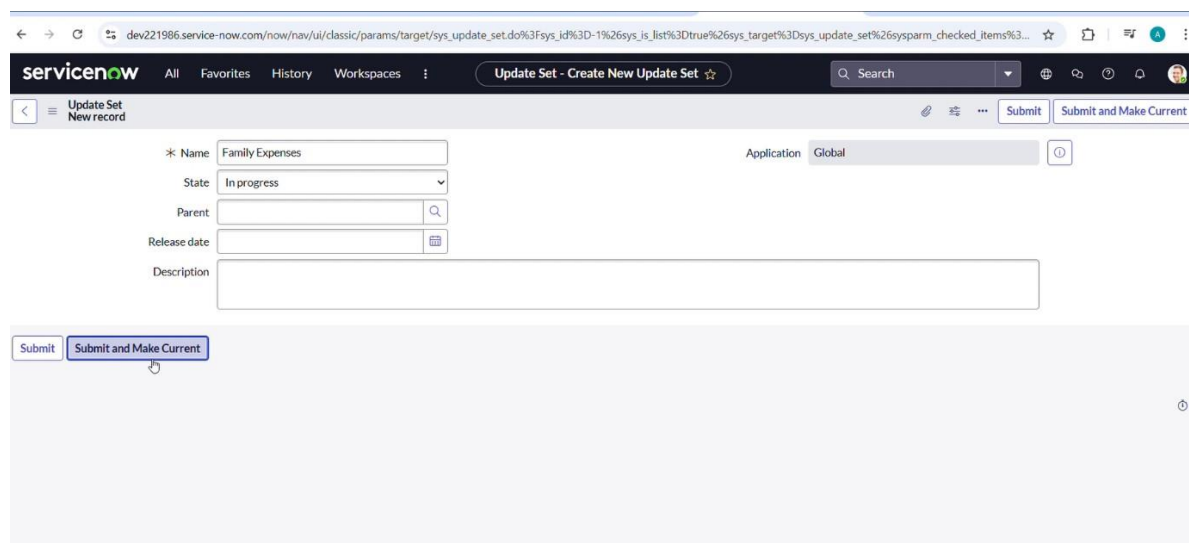
- Go to the official ServiceNow Developer portal: <https://developer.servicenow.com> and create a developer account.
- After signing in, open the Personal Developer Instance section from the dashboard.
- Select Request Instance to generate a fresh ServiceNow environment for development.
- Provide the necessary details (like version selection) and confirm your request.
- Wait for the confirmation email containing your instance URL and login credentials.
- Use the credentials to log into your newly created ServiceNow instance.
- Once inside, explore the interface and begin working on the platform.

The screenshot shows the 'Manage my instance' page for a ServiceNow Developer Instance (dev221986). The page is divided into several sections:

- Status:** Shows the instance is 'Online' with a 'Refresh' button.
- App engine studio, Creator studio, ServiceNow studio:** All three are marked as 'Installed'.
- Your Current Version:** Shows 'Xanadu' with an 'Upgrade release' button.
- Instance URL:** <https://dev221986.service-now.com/>
- User name:** admin
- Current password:** Masked with dots and an eye icon to toggle visibility.
- User role:** Admin
- Useful links:** Includes links to 'Personal Developer Instance (PDI) Guide', 'PDI FAQs', 'Managing your PDI', and 'Developer advocate blog'.
- Plugins for your instance (53):** A section with a search bar and filters for 'All activation statuses' and 'All demo data statuses'.

MILSTONE2:CREATIONOFNEWUPDATE SET

- LogintoyourServiceNowinstanceandgotothe Application Navigator.
- SearchforUpdateSetsandopenLocalUpdateSets under *System Update Sets*.
- Clickon Newtocreateafresh updateset.
- Enterthefollowing details:
 - Name:*Family Expenses*
 - Description:Updatesettocaptureallconfigurations related to the Family Expense Management project.
- SavetherecordandmarkitastheCurrentUpdateSet,so every change you make is tracked under this set.
- Verifythattheupdatesetisactivebycheckingtheheader at the top of the screen.
- From this point forward, all customizations (tables, relationships,andbusinessrules)willberecordedinside the *Family Expenses* update set.



The screenshot shows the ServiceNow interface for creating a new update set. The browser address bar displays a URL from dev221986.service-now.com. The page title is 'Update Set - Create New Update Set'. The breadcrumb trail is 'Update Set > Create New Update Set'. The form fields are as follows:

- Name:** Family Expenses
- State:** In progress (dropdown menu)
- Parent:** (empty field with a search icon)
- Release date:** (empty field with a calendar icon)
- Description:** (empty text area)
- Application:** Global (dropdown menu)

At the bottom of the form, there are two buttons: 'Submit' and 'Submit and Make Current'. A mouse cursor is hovering over the 'Submit and Make Current' button.

MILSTONE3:CREATIONOFTABLEFAMILY EXPENSES

Activity1 –CreatingtheFamilyExpensesTable

- InyourServiceNowinstance,navigatetoAll>Tables using the filter navigator.
- Click onNewtcreateanew table.
- Fillin therequreddetails:
 - Label:*FamilyExpenses*
 - Name:(Thiswillbeauto-generatedbasedonthe label)
 - NewMenuName:*FamilyExpenditure*
- Savetherecord tocreatethenewtable.

dev221986.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%3D...

servicenow All Favorites History Workspaces Admin Table - New Record Search

Table New record Submit Cancel

ServiceNow recommends creating custom tables in scoped applications. To learn more about creating scoped applications, click here.

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 FamilyExpenses I

* Name u_family_expenses

Extends table

Application Global

Create module ☒

Create mobile module ☒

Add module to menu -- Create new --

New menu name FamilyExpenses

Columns Controls Application Access

Table Columns for text Search

Dictionary Entries

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

MILSTONE3:CREATIONOFTABLEFAMILY EXPENSES

Activity2–AddingColumnstotheFamilyExpensesTable

Duration:1 Hour

SkillTags:TableConfiguration,DataModeling,ServiceNowBasics

- Openthe newly created FamilyExpensestable.
- To add columns,double-clicknearthe existing columnstoinsert a new row.
- Enterthefollowingdetailsonesbyone:
 1. ColumnLabel: Number
 - Type:String
 2. ColumnLabel:Date
 - Type:Date
 3. ColumnLabel:Amount
 - Type:Integer
 4. ColumnLabel:ExpenseDetails
 - Type:String
 - MaxLength:800

The screenshot shows the ServiceNow interface for configuring the 'Family Expenses' table. The 'Columns' tab is selected, and a search bar is present. Below the search bar, there is a table of 'Dictionary Entries' with columns: Column label, Type, Reference, Max length, Default value, and Display. The table lists existing columns and a new column 'Expense' being added.

Column label	Type	Reference	Max length	Default value	Display
Updates	Integer	(empty)	40	false	false
Updated	Date/Time	(empty)	40	false	false
Created by	String	(empty)	40	false	false
Sys ID	Sys ID (GUID)	(empty)	32	false	false
Updated by	String	(empty)	40	false	false
Created	Date/Time	(empty)	40	false	false
Number	String			false	false
Date	Date			false	false
Amount	Integer			false	false
Expense	String			false	false

MILSTONE3:CREATIONOFTABLEFAMILY EXPENSES

Activity3–MakingtheNumberFieldanAuto-Number Open the Family Expenses table.

- Locatethe Numberfield/columnanddouble-clicktoopenits properties.
- ScrolldownandswitchtotheAdvancedview.
- IntheDefault Valuessection:
 - Checktheboxfor*UseDynamic Default*.
 - SettheDynamicDefaultValueto*GetNextPadded Number*.
- ClickUpdatetosavethe changes.

The screenshot shows the ServiceNow interface for configuring a Dictionary Entry of type 'Number'. The page is titled 'Dictionary Entry - Number' and is in the 'Advanced' view. The 'Default Value' tab is selected, showing options to set a dynamic default. The 'Use dynamic default' checkbox is checked, and the 'Dynamic default value' is set to 'Get Next Padded Number'. The 'Max length' is set to 40. The 'Mandatory' and 'Display' checkboxes are unchecked. The 'Attributes' field is empty. The 'Delete Column' and 'Update' buttons are visible at the bottom.

servicenow All Favorites History Workspaces : Dictionary Entry - Number Search

Dictionary Entry Number View: Advanced Max length 40 Mandatory Display

Alters the behavior of a field or functionality that depends on the field. [More Info](#)

Attributes

Choice List Specification Calculated Value **Default Value**

The **Default value** specifies what value the field has when first displayed.

Use dynamic default ☒

Dynamic default value Get Next Padded Number

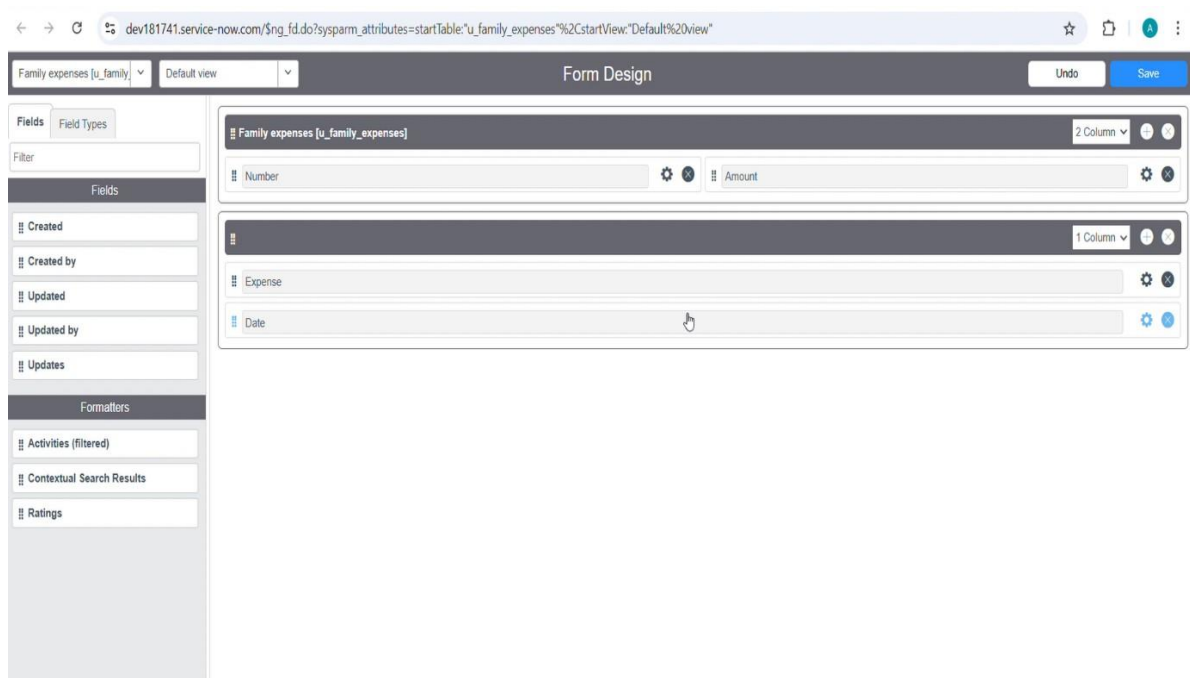
Delete Column Update

Related Links

MILSTONE3:CREATIONOFTABLEFAMILY EXPENSES

Activity4—ConfiguringtheForm

- NavigatetoAll>inthe filter,searchforFamilyExpenses.
- OpentheFamilyExpensestable.
- ClickonNewtocreateanewformentry.
- Ontheformheader,right-clickandselect:
 - Configure>FormDesign.
- IntheFormDesigner,usedrag-and-dropto:
 - Rearrangefields.
 - Grouprelatedfieldstogether.
 - Addsectionsifrequiredforbetterclarity.
- Savethecustomizedformlayout.



MILSTONE4:CREATIONOFTABLEDAILY EXPENSES

Activity1–CreatingTheDailyExpensesTable

- NavigatetoAll>Tablesusingthefilternavigator.
- ClickonNewtocreate anew table.
- Fillintherequireddetails:
- Label:DailyExpenses
- Name:(Auto-populatedbythesystem)
- AddModuletoMenu:*Family Expenditure*
- Gototheformheader,right-click,andselectSave.

dev181741.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%3D%26sys...

servicenow All Favorites History Workspaces Admin Table - New Record Search

Table New record

ServiceNow recommends creating custom tables in scoped applications. To learn more about creating scoped applications, click [here](#).

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 DailyExpenses

* Name u_daily_expenses

Extends table

Application Global

Create module ☒

Create mobile module ☒

Add module to menu -- Create new --

New menu name Daily Expenses

Remote Table ☐

Columns Controls Application Access

Table Columns for text Search

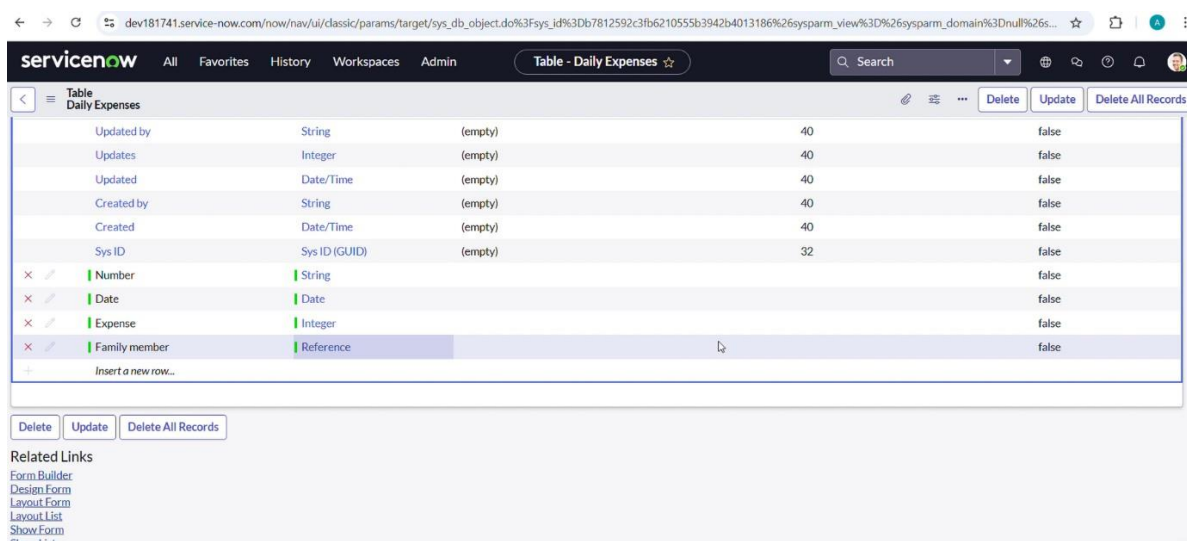
Dictionary Entries

Column label	Type	Reference	Max length	Default value	Display
--------------	------	-----------	------------	---------------	---------

MILSTONE4:CREATIONOFTABLEDAILY EXPENSES

Activity2–CreatingColumns(Fields)

- OpentheDailyExpenses table.
- NearColumns,double-clicktoinsertanew rowandaddthefollowing fields:
 1. ColumnLabel:Number
 - Type:String
 2. ColumnLabel:Date
 - Type:Date
 3. ColumnLabel:Expense
 - Type:Integer
 4. ColumnLabel:FamilyMemberName
 - Type:Reference
 - MaxLength:800
 5. ColumnLabel:Comments
 - Type:String
 - **MaxLength: 800**



Field	Type	Length	Nullable
Updated by	String	(empty)	false
Updates	Integer	(empty)	false
Updated	Date/Time	(empty)	false
Created by	String	(empty)	false
Created	Date/Time	(empty)	false
Sys ID	Sys ID (GUID)	(empty)	false
Number	String	40	false
Date	Date	40	false
Expense	Integer	40	false
Family member	Reference	32	false

MILSTONE4:CREATIONOFTABLEDAILY EXPENSES

Activity3–MakingNumberFieldanAuto-Number

- OpentheDailyExpenses table.
- LocatetheNumberfield/columnanddouble-clicktoopenitsproperties.
- ScrolldownandswitchtotheAdvancedView.
- IntheDefaultValuessection:
 - ChecktheboxforUseDynamicDefault.
 - SettheDynamicDefault ValuetoGetNextPaddedNumber.
- ClickUpdatetosavechanges.

Configuring Number Maintenance:

- NavigatetoAll>NumberMaintenance.
- ClickonNew.
- Enterthedetailsasfollows:
 - Table:FamilyExpenses
 - Prefix:MFE
- Clickon Submit.

The screenshot shows the ServiceNow interface for configuring a 'Dictionary Entry - Number'. The browser address bar shows a URL with a long alphanumeric string. The page title is 'Dictionary Entry - Number' with a star icon. Below the title bar, there are tabs for 'Choice List Specification', 'Calculated Value', and 'Default Value', with 'Default Value' being the active tab. A blue informational box states: 'The Default value specifies what value the field has when first displayed.' Below this, there is a section for 'Dynamic default value' with a checkbox 'Use dynamic default' which is checked. The text input field next to it contains 'Get Next Padded Number'. At the bottom of the configuration area, there are 'Delete Column' and 'Update' buttons. Below the configuration area, there is a 'Related Links' section with links for 'Show Table', 'Run Point Scan', and 'Default view'. At the very bottom, there is a table with columns: Name, Decision Type, Operation, Type, Active, Updated by, and Updated. The table is currently empty.

Invalid insert

* Table Daily Expenses

Prefix DFE

* Number 1,000

Application Global

Number of digits 7

Submit

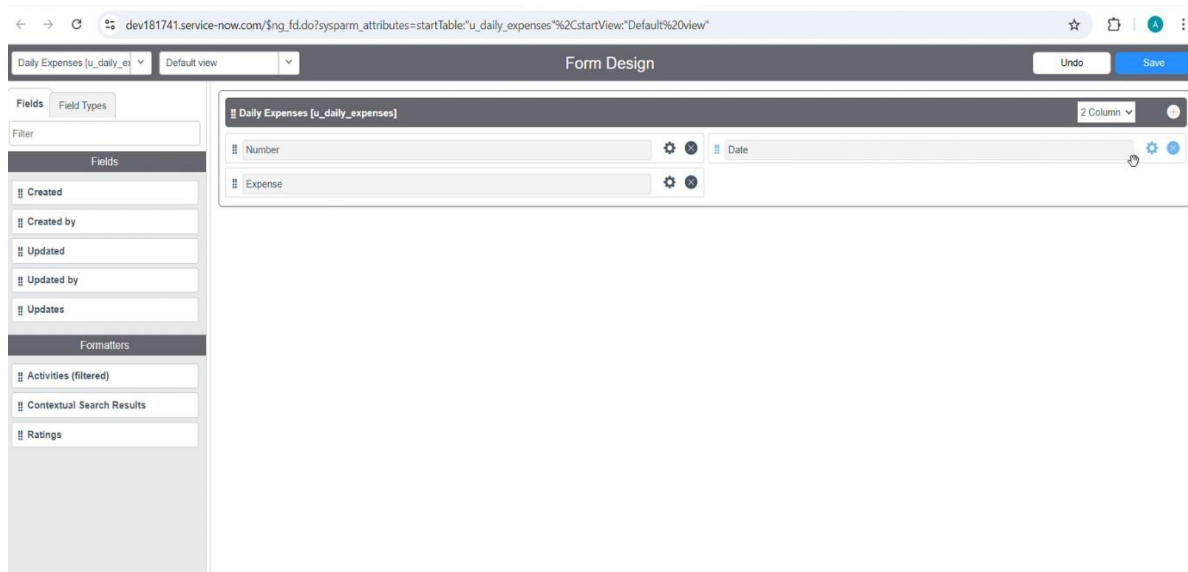
Related Links

Show Counter

MILSTONE4:CREATIONOFTABLEDAILY EXPENSES

Activity4–ConfigureTheForm

- NavigatetoAll>DailyExpensesusingthefilter.
- OpentheDailyExpensestable.
- ClickonNewtocreateanewformentry.
- Ontheformheader,right-click,thenselect:
- Configure>FormDesign.
- IntheFormDesigner,draganddropfieldstocustomizethe form layout as per requirement.
- Applythefollowingconfigurations:
- NumberField→Setas*Read-Only*byclickingthegeariconand checking Read-Only.
- DateField→Setas*Mandatory*byclickingthegeariconand checking Mandatory.
- FamilyMemberNameField →Setas*Mandatory*usingthe same method.
- ClickSavetoapplythechanges.



MILSTONE5:CREATIONOFRELATIONSHIP BETWEEN FAMILYEXPENSESAND DAILY EXPENSES TABLES

- NavigatetoAll>Relationshipsusingthefilternavigator.
- ClickonNewtocreate anew relationship.
- Fillinthedetailsasfollows:
- Name:DailyExpenses
- AppliestoTable:*FamilyExpenses*
- RelatedListTable:*DailyExpenses*
- ClickSave.

dev181741.service-now.com/now/nav/ui/classic/params/target/sys_relationship.do%3Fsys_id%3Da834e116c3fb6210555b3942b4013151%26sysparm_view%3D%26sysparm_domain%3Dnull%26...

servicenow All Favorites History Workspaces Relationship - Daily Expenses Search

Relationship Daily Expenses Update Delete

Name: Application:

Advanced ☐ Applies to table: Queries from table:

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   // Add your code here, such as current.addQuery(field, value);
4
5 }(current, parent);
```

Update Delete

Related Links
[Run Point Scan](#)

MILSTONE6:CONFIGURINGRELATEDLIST ON FAMILY EXPENSES

- NavigatetoAll>FamilyExpensesusingthefilter.
- OpentheFamilyExpensestable.
- ClickonNewtoopentheformview.
- Ontheformheader,right-click,thenselect:
- Configure>RelatedLists.
- Fromtheavailableoptions,add DailyExpensestotheSelected Area.
- ClickSavetoapplythechanges.

dev181741.service-now.com/now/nav/ui/classic/params/target/slusbucket.do%3Fsysparm_referring_url%3Du_family_expenses.do%253Fsys_id%253D-1%254099%2540sysparm_target%25409...

servicenow All Favorites History Workspaces Admin ServiceNow Search

< Configuring related lists on Family expenses form Cancel Save

Available
Attachments

Selected
Daily Expenses

> < ^ v

Cancel Save

View name: Default view

Related Links
[Show versions](#)

MILSTONE7:CREATIONOFBUSINESSRULES

- NavigatetoAll>BusinessRulesusingthefilter.
- UnderSystemDefinition,selectBusinessRulesandclickNew.
- Enterthefollowingdetails:
 - Name:FamilyExpensesBR
 - Table:*DailyExpenses*
 - CheckAdvanced.
- IntheWhentorunsection,check:
 - Insert
 - Update

- IntheAdvancedtab,addthefollowingscript:

```
(functionexecuteRule(current,previous/*nullwhenasync*/){  
  
    varFamilyExpenses=newGlideRecord('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{  
        varNewFamilyExpenses=newGlideRecord('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);
```

Goto the form header, right-click, then select Save.

The screenshot shows the ServiceNow 'Business Rule - New Record' configuration page. The browser address bar shows a URL from dev181741.service-now.com. The page header includes the ServiceNow logo and navigation links like 'All', 'Favorites', 'History', and 'Workspaces'. The main form has the following fields:

- Name:** Family expenses BR
- Table:** Daily Expenses [u_daily_expenses]
- Application:** Global
- Active:** ☒
- Advanced:** ☒

Below these fields are tabs for 'When to run', 'Actions', and 'Advanced'. The 'Advanced' tab is selected, showing a 'Condition' field and a 'Script' section. The 'Script' section has a toggle for 'Turn on ECMAScript 2021 (ES12) mode' which is currently off. The script area contains the following code:

```
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
```

MILSTONE8:CONFIGURETHE RELATIONSHIP

- NavigatetoAll>Relationshipsusingthefilternavigator.
- OpentheexistingDailyExpensesRelationship.
- Updatethedetailsasfollows:
 - AppliestoTable:*FamilyExpenses*
- IntheQuerywith section,enterthefollowingscript:
(function refineQuery(current, parent) {

```
//Addyourcodehere,suchascurrent.addQuery(field,value);  
current.addQuery('u_date', parent.u_date);  
current.query();  
})(current,parent);
```

ClickUpdatetosavethe configuration

The screenshot shows the ServiceNow interface for configuring a relationship. The browser address bar shows a URL with a long ID. The page title is 'Relationship - Daily Expenses'. The configuration fields are as follows:

- Name: Daily Expenses
- Application: Global
- Advanced: ☐
- Applies to table: Family expenses [u_family_expenses]
- Queries from table: Daily Expenses [u_daily_expenses]

A blue informational banner states: "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](#)."

The 'Query with' section has a toggle for 'Turn on ECMAScript 2021 (ES12) mode' which is currently off. Below this is a code editor with the following script:

```
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);
```

At the bottom of the configuration section are 'Update' and 'Delete' buttons. Below that is a 'Related Links' section.

CONCLUSION

The Family Expenses Management System built on ServiceNow demonstrates how the platform can be leveraged beyond IT workflows to solve real-world problems. By systematically creating tables, relationships, forms, and business rules, the project enables seamless tracking of both daily expenses and family-level expenses in an automated manner.

The use of auto-numbering, mandatory fields, related lists, and business rules ensures data integrity, consistency, and accuracy. The relationship configuration further enhances visibility by linking daily records to family-level summaries, providing a clear financial overview.

Through this project, we learned how to apply ServiceNow features such as table creation, form design, field configuration, scripting, and automation to build a complete application. More importantly, it highlights how low-code/no-code platforms like ServiceNow can be extended into personal finance, household management, and non-IT use cases.

Overall, the project provides a practical, scalable, and user-friendly solution for managing family expenses efficiently while also strengthening skills in ServiceNow application development.