

**CALCULATING FAMILY EXPENSES USING
SERVICE NOW**

NAAN MUDALVAN PROJECT

TEAM ID:NM2025TMID17799

Team Leader: Rishikeshwar.B

Team Member : Nisanth

Team Member: Vidhya sri. B

Team Member: Kamal raj

ST.JOSEPH'S COLLEGE (ARTS & SCIENCE)

INDEX

1. Introduction
2. Abstract
3. Problem statement
4. Solution
5. Practical use
6. Knowledge gained
7. Milestone 1:Setting up servicenow instance
8. Milestone 2:Creation of new update set
9. Milestone 3:Creation of table family expenses
- 10.Milestone 4: Creation of table daily expenses
- 11.Milestone 5: Creation of relationship
- 12.Milestone 6:Configuring Related list on family 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

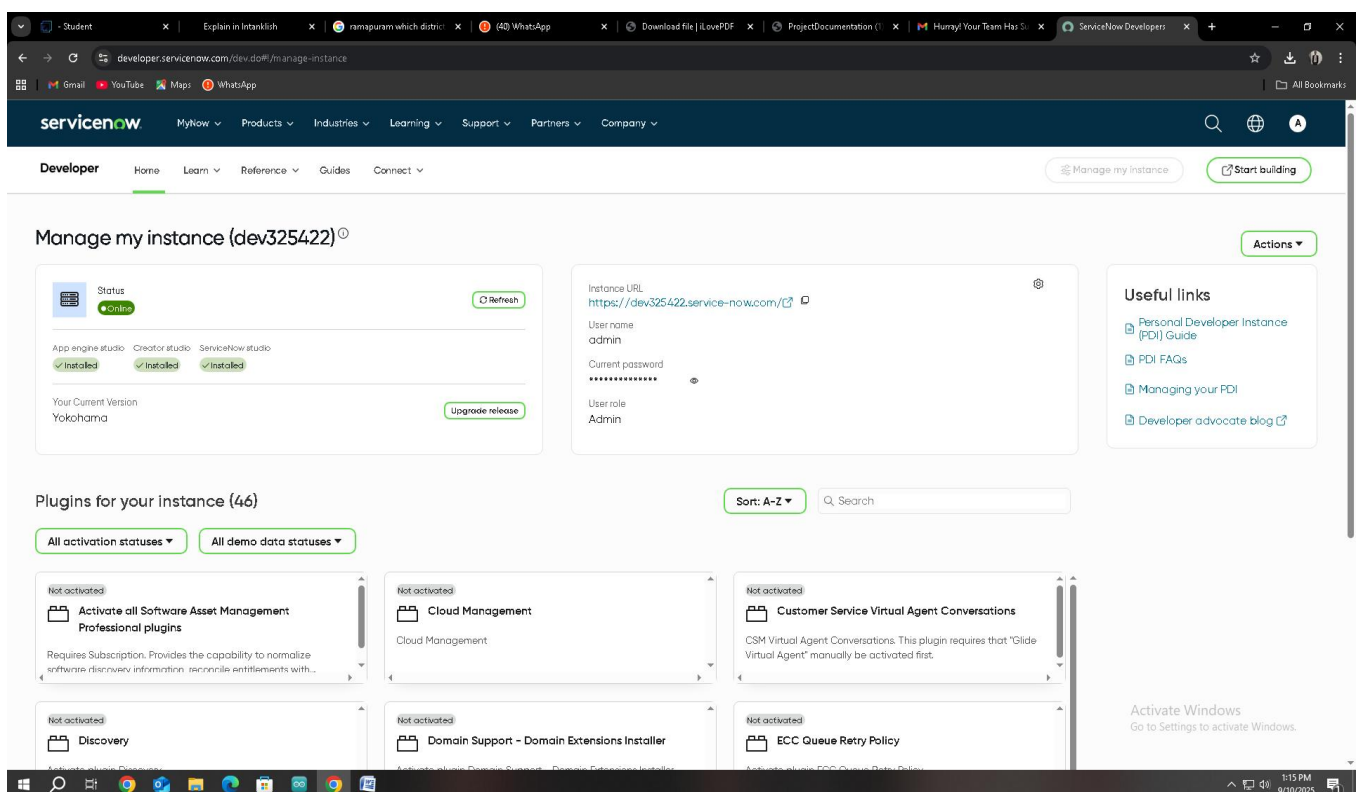
The Family Expense Management System built on ServiceNow helps families track, categorize, and analyze their expenses in a structured way. It simplifies budgeting, reduces errors from manual tracking, and provides real-time insights into spending patterns. Families can use it to set limits, monitor monthly 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:SETTING UP THE SERVICE NOW INSTANCE

- Go to the official Service Now 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 Service Now 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 Service Now instance.
- Once inside , explore the interface and begin working on the platform.



MILSTONE2:CREATION OF NEW UPDATE SET

- Log into your Service Now instance and go to the Application Navigator.
- Search for Update Sets and open Local Update Sets under *System Update Sets*.
- Click on New to create a fresh updateset.
- Enter the following details:
 - Name: *Family Expenses*
 - Description: Updateset to capture all configurations related to the Family Expense Management project.
- Save the record and mark it as the Current Update Set, so every change you make is tracked under this set.
- Verify that the update set is active by checking the header at the top of the screen.
- From this point forward, all customizations (tables, relationships, and business rules) will be recorded inside the *Family Expenses* update set.

The screenshot shows the ServiceNow interface for creating a new update set. The browser address bar shows the URL: `dev325422.service-now.com/now/nav/ui/classic/params/target/sys_update_set.do%3Fsys_id%3D898fa2e493f722109d16393efaba10fe%26sysparm_record_target%3Dsys...`. The page title is "Update Set - Family Expenses". The form contains the following fields:

Field	Value
Name	Family Expenses
State	In progress
Parent	
Release date	
Install date	
Installed from	
Description	
Application	Global
Created	2025-09-02 01:50:30
Created by	admin
Merged to	

Below the form, there is an "Update" button and a section for "Related Links" with the following links:

- [Make This My Current Set](#)
- [Merge With Another Update Set](#)
- [Scan Update Set](#)

At the bottom, there is a tabbed interface with the following tabs:

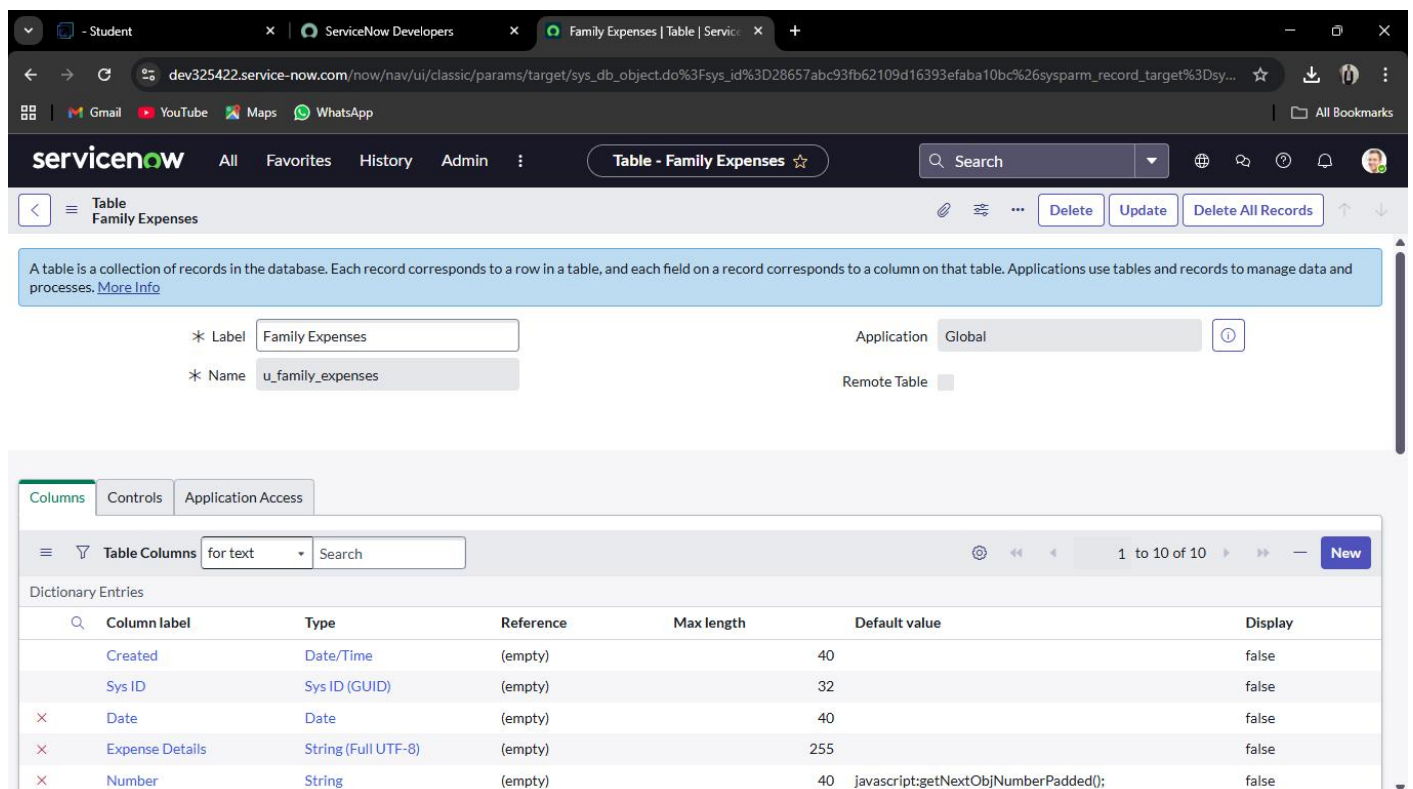
- Customer Updates (27)
- Update Set Logs
- Child Update Sets
- Install History

The "Customer Updates (27)" tab is selected, showing a table with columns: Created, Search, and Actions on selected rows...

MILSTONE3:CREATIONOFTABLEFAMILY EXPENSES

Activity1 –CreatingtheFamilyExpensesTable

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



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 Application

* Name Remote Table ☐

Columns Controls Application Access

Table Columns for text Search

Dictionary Entries

	Column label	Type	Reference	Max length	Default value	Display
	Created	Date/Time	(empty)	40		false
	Sys ID	Sys ID (GUID)	(empty)	32		false
X	Date	Date	(empty)	40		false
X	Expense Details	String (Full UTF-8)	(empty)	255		false
X	Number	String	(empty)	40	javascript:getNextObjNumberPadded();	false

MILSTONE3:CREATIONOFTABLEFAMILY EXPENSES

Activity3–MakingtheNumberFieldanAuto-Number Open the Family Expenses table.

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

The screenshot shows the ServiceNow interface for editing a Dictionary Entry of type 'Number'. The page is titled 'Dictionary Entry - Number' and is in 'Advanced' view. The 'Default Value' tab is selected, showing a configuration for the default value. The 'Use dynamic default' checkbox is checked, and the 'Dynamic default value' is set to 'Get Next Padded Number'. The 'Attributes' field is empty. The 'Display' checkbox is unchecked. The page includes a search bar, navigation links, and a bottom section for 'Related Links' with links to 'Show Table' and 'Run Point Scan'.

dev325422.service-now.com/now/nav/ui/classic/params/target/sys_dictionary.do%3Fsys_id%3D51063afc93fb62109d16393efaba102a%26sysparm_record_list%3Dname...

servicenow All Favorites History Admin Dictionary Entry - Number Search

Dictionary Entry Number View: Advanced* Delete Column Update

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

Delete Column Update

Related Links

[Show Table](#)

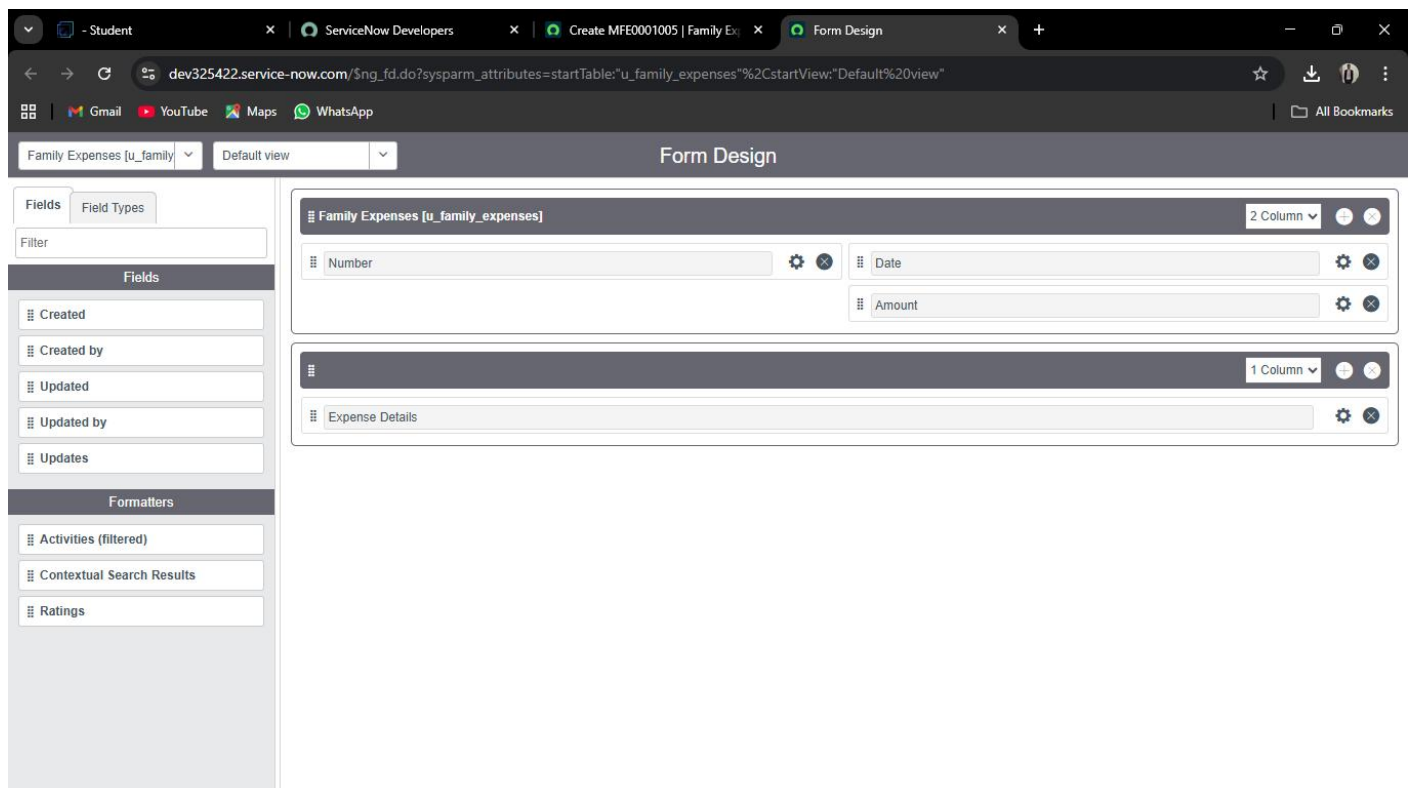
[Run Point Scan](#)

Access Controls Choices Attributes Labels (1)

MILSTONE3:CREATIONOFTABLEFAMILY EXPENSES

Activity4–ConfiguringtheForm

- NavigatetoAll>inthefilter,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.

The screenshot shows the ServiceNow interface for creating a new table. The browser tabs include 'Student', 'ServiceNow Developers', 'Daily Expenses | Table | ServiceNow', and 'Form Design'. The URL is 'dev325422.service-now.com/now/nav/ui/classic/params/target/sys_db_object.do%3Fsys_id%3D73ac7a38933f62109d16393efaba106a%26sysparm_record_target%3Dsys_db_...'. The page title is 'Table - Daily Expenses'. The interface includes a search bar, a 'Table - Daily Expenses' header, and a 'New' button. Below the header, there is a description of a table and a form to define the table's properties. The form includes fields for 'Label' (set to 'Daily Expenses'), 'Name' (set to 'u_daily_expenses'), 'Application' (set to 'Global'), and 'Remote Table' (set to 'false'). Below the form, there is a 'Columns' tab and a 'Table Columns' section. The 'Table Columns' section shows a list of columns with their labels, types, references, max lengths, default values, and display status. The columns are: 'Family Members Name' (String, 800, empty, false), 'Sys ID' (Sys ID (GUID), 32, empty, false), 'Created' (Date/Time, 40, empty, false), 'Updated by' (String, 40, empty, false), and 'Updates' (Integer, 40, empty, false).

* Label Application

* Name Remote Table ☐

Columns Controls Application Access

Table Columns for text Search 1 to 10 of 10 New

Dictionary Entries

Column label	Type	Reference	Max length	Default value	Display
Family Members Name	String	(empty)	800		false
Sys ID	Sys ID (GUID)	(empty)	32		false
Created	Date/Time	(empty)	40		false
Updated by	String	(empty)	40		false
Updates	Integer	(empty)	40		false

MILSTONE4:CREATIONOFTABLEDAILY EXPENSES

Activity2–CreatingColumns(Fields)

- Open the Daily Expenses table.
- Near Columns, double-click to insert a new row and add the following 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**

The screenshot shows the ServiceNow interface for configuring the 'Table - Daily Expenses'. The table configuration is as follows:

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

Buttons at the bottom: Delete, Update, Delete All Records.

Related Links: Form Builder, Design Form, Layout Form.

MILSTONE4:CREATIONOFTABLEDAILY EXPENSES

Activity3–MakingNumberFieldanAuto-Number

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

Configuring Number Maintenance:

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

The screenshot shows the ServiceNow web interface for configuring a Number field. The browser tabs include 'Student', 'ServiceNow Developers', 'Number | Dictionary Entry | Ser...', and 'Form Design'. The URL is 'dev325422.service-now.com/now/nav/ui/classic/params/target/sys_dictionary.do%3Fsys_id%3D6ffc3638933f62109d16393efaba1048%26sysparm_record_list%3DnameONE%...'. The page title is 'Dictionary Entry - Number'.

The configuration page is titled 'Dictionary Entry - Number View: Advanced*'. It features a 'Display' toggle and buttons for 'Delete Column' and 'Update'. A blue informational box states: 'Alters the behavior of a field or functionality that depends on the field. [More Info](#)'. Below this is an 'Attributes' section with an empty text input field.

The 'Default Value' tab is selected, showing a blue box: 'The Default value specifies what value the field has when first displayed.' Below this, the 'Use dynamic default' checkbox is checked. The 'Dynamic default value' field contains 'Get Next Padded Number' with search and help icons.

At the bottom, there are 'Delete Column' and 'Update' buttons. A 'Related Links' section includes 'Show Table' and 'Run Point Scan'. A tabbed interface at the very bottom shows 'Access Controls' (selected), 'Choices', 'Attributes', and 'Labels (1)'.

Student

ServiceNow Developers

New Record | Number | ServiceNow

Form Design

dev325422.service-now.com/now/nav/ui/classic/params/target/sys_number.do%3Fsys_id%3D-1%26sys_is_list%3Dtrue%26sys_target%3Dsys_number%26sysparm_checked_it...

GmailYouTubeMapsWhatsApp

AllFavoritesHistoryWorkspaces

Number - New Record

Search

All Bookmarks

<≡Number
New record

Submit

* TableDaily Expenses

PrefixDFE

* Number1,000

ApplicationGlobal

Number of digits7

Submit

Related Links

[Show Counter](#)

MILSTONE4:CREATIONOFTABLEDAILY EXPENSES

Activity4–ConfigureTheForm

- NavigatetoAll>DailyExpensesusingthefilter.
- OpentheDailyExpensestable.
- ClickonNewtcreateanewformentry.
- 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.

The screenshot shows the ServiceNow Form Designer interface for the 'Daily Expenses' table. The browser address bar indicates the URL: `dev325422.service-now.com/sng_fd.do?sysparm_attributes=startTable:"u_daily_expenses"%2CstartView:"Default%20view"`. The interface is titled 'Form Design' and shows the 'Daily Expenses [u_daily_expenses]' form. The form layout includes a 'Number' field, a 'Date' field, a 'Family Members Name' field, and an 'Expense' field. A 'New Section' is also visible, containing a 'Comments' field. The left sidebar shows the 'Fields' and 'Field Types' tabs, with a list of fields including 'Created', 'Updated', 'Updated by', 'Updates', 'Activities (filtered)', 'Contextual Search Results', and 'Ratings'. The 'Formatters' section is also visible, showing 'Activities (filtered)', 'Contextual Search Results', and 'Ratings'.

MILSTONE5:CREATIONOFRELATIONSHIP BETWEEN FAMILYEXPENSESAND DAILY EXPENSES TABLES

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

Relationship - Daily Expenses

Name

Advanced ☐

Application

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
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);
12
13
```

MILSTONE6:CONFIGURINGRELATEDLIST ON FAMILY EXPENSES

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

The screenshot displays the ServiceNow interface for configuring related lists on the 'Family Expenses' form. The browser window shows the URL: `dev325422.service-now.com/now/nav/ui/classic/params/target/slusbucket.do%3Fsysparm_referring_url%3Du_family_expenses.do%253Fsys_id%253D-1%254099%2540sysp...`. The ServiceNow header includes the logo, navigation tabs (All, Favorites, History, Workspaces), a search bar, and user profile information. The main content area is titled 'Configuring related lists on Family Expenses form' and includes a 'Back' button and 'Cancel'/'Save' buttons. The interface is divided into two main sections: 'Available' and 'Selected'. The 'Available' section contains a list with the item 'Attachments'. The 'Selected' section contains a list with the item 'Daily Expenses'. Between these two sections are two sets of arrows: a right-pointing arrow (>) and a left-pointing arrow (<). To the right of the 'Selected' list are up and down arrow buttons (^ and v). Below the lists, there is a 'View name' dropdown menu currently set to 'Default view'. At the bottom left, there is a 'Related Links' section with a link to 'Show versions'. A small gear icon is located at the bottom right of the interface.

MILSTONE7:CREATIONOFBUSINESSRULES

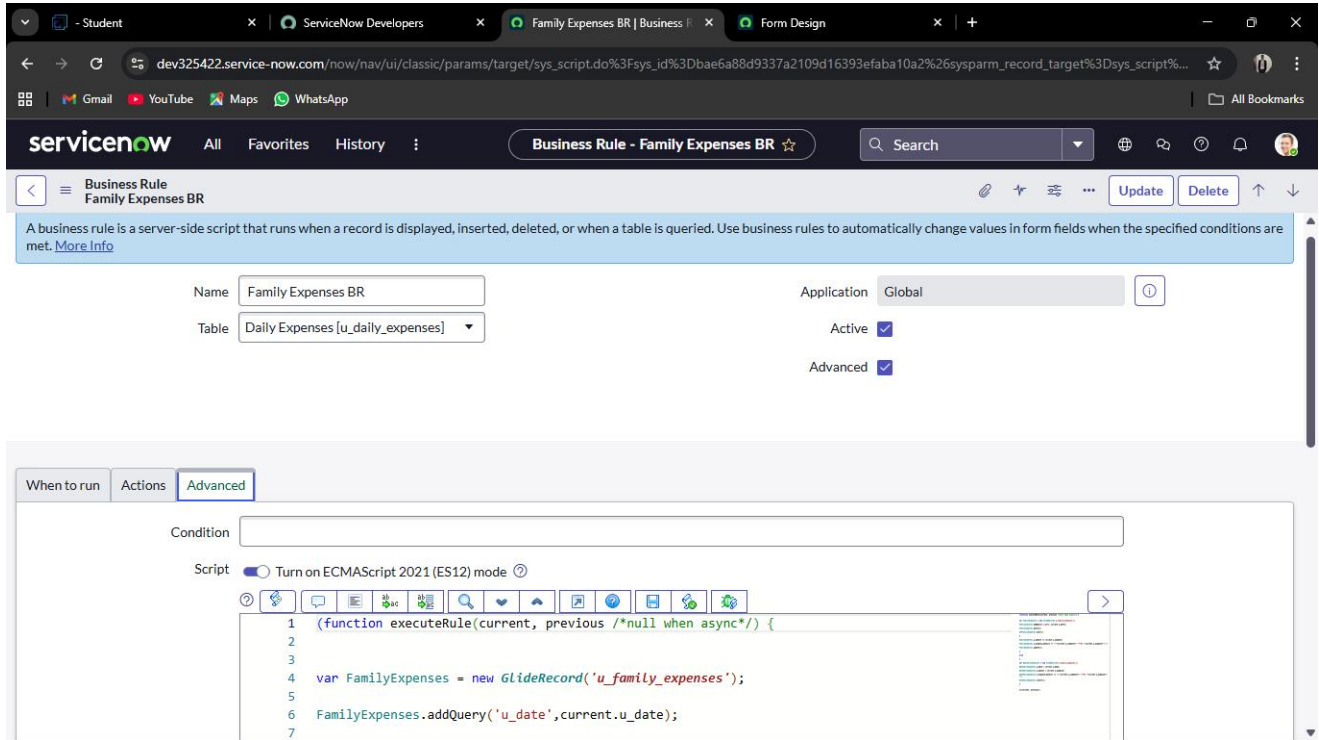
- NavigatetoAll>BusinessRulesusingthefilter.
- UnderSystemDefinition,selectBusinessRulesandclickNew.
- Enterthefollowingdetails:
 - Name:FamilyExpensesBR
 - Table:*DailyExpenses*
 - CheckAdvanced.
- IntheWhentorunsection,check:
 - Insert
 - Update

- 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.



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 web interface for configuring a relationship. The browser tabs include 'Student', 'ServiceNow Developers', 'Daily Expenses | Relationship', and 'Form Design'. The URL is 'dev325422.service-now.com/now/nav/ui/classic/params/target/sys_relationship.do%3Fsys_id%3D1d15a00d9337a2109d16393efaba109f%26sysparm_record_target%3Dsys_re...'. The page title is 'Relationship - Daily Expenses'. The configuration fields are: Name: 'Daily Expenses', Application: 'Global', Applies to table: 'Family Expenses [u_family_expens...', and Queries from table: 'Daily Expenses [u_daily_expenses]'. A blue information box 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 turned on. The script editor contains the following code:

```
1 (function refineQuery(current, parent) {  
2  
3  
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);  
12  
13
```


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.