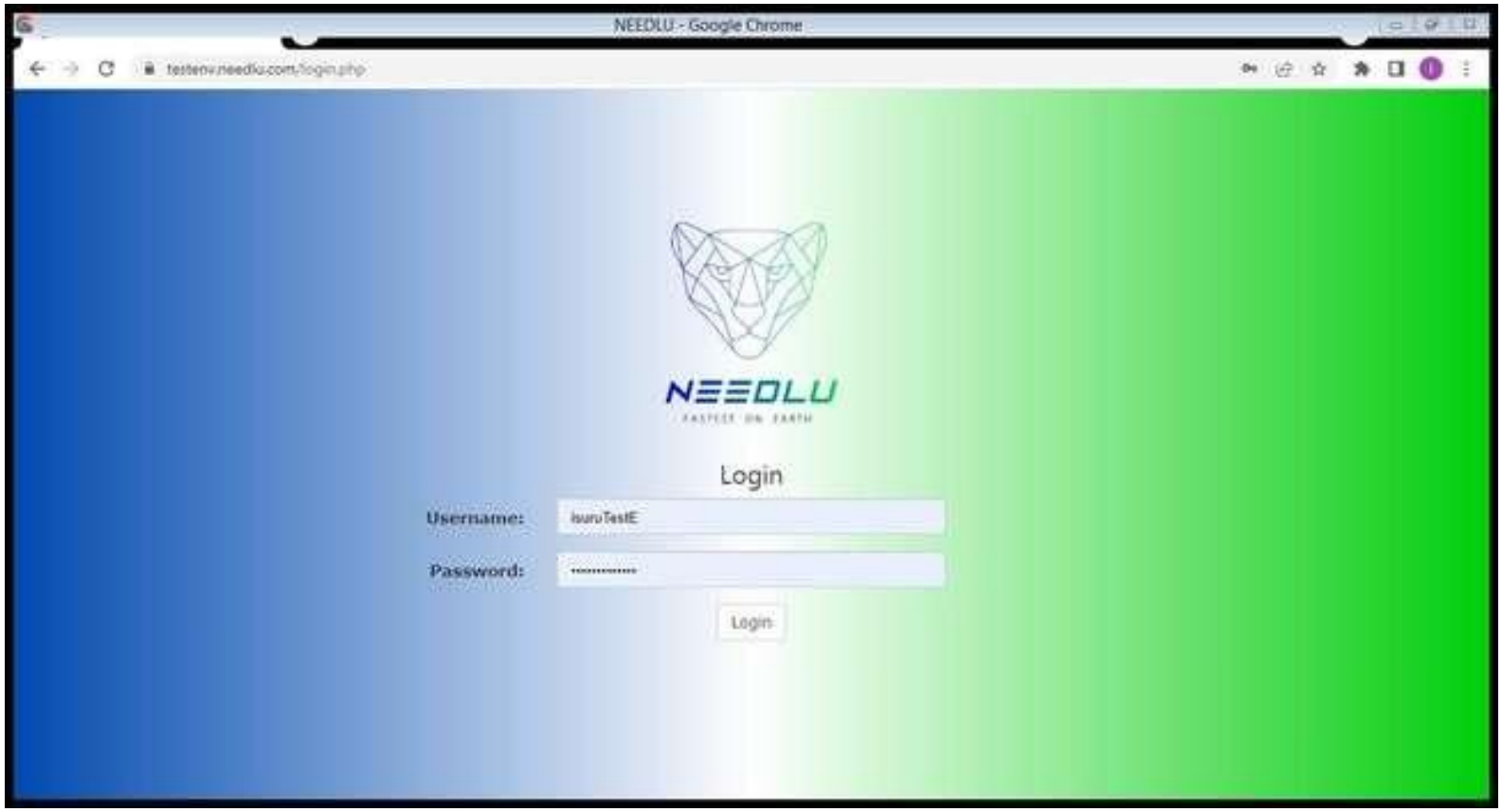
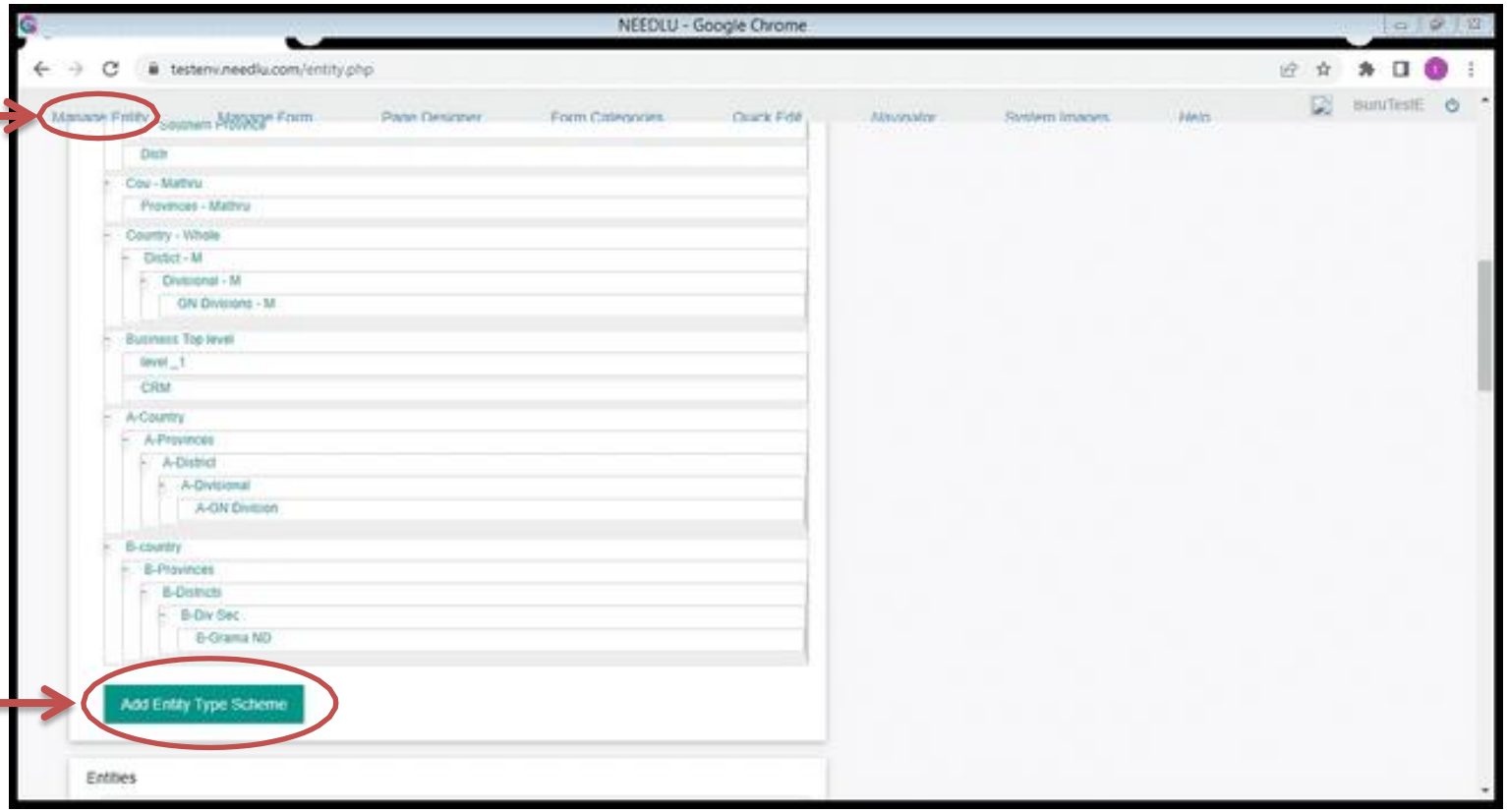
Let’s see how to create a software using the Needlu software developer tool.

- **First of all, you need to log in to the editor role of the Needlu software developer tool using your username and password.**



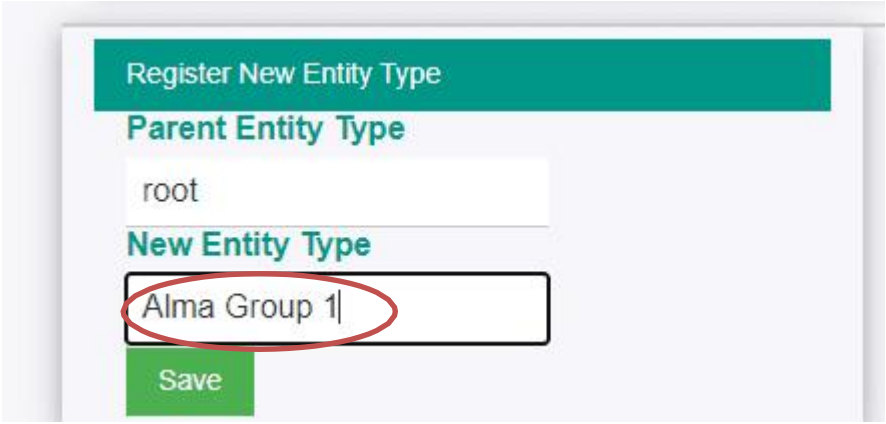
➢ After logging in you must create Entity types and Entities.

1. Click on ‘manage entity’
2. Click on ‘add Entity type scheme



- **Now you can create entity types. We create these entity types to determine the level we want to save data. I will develop a customer relationship management system as an example.**

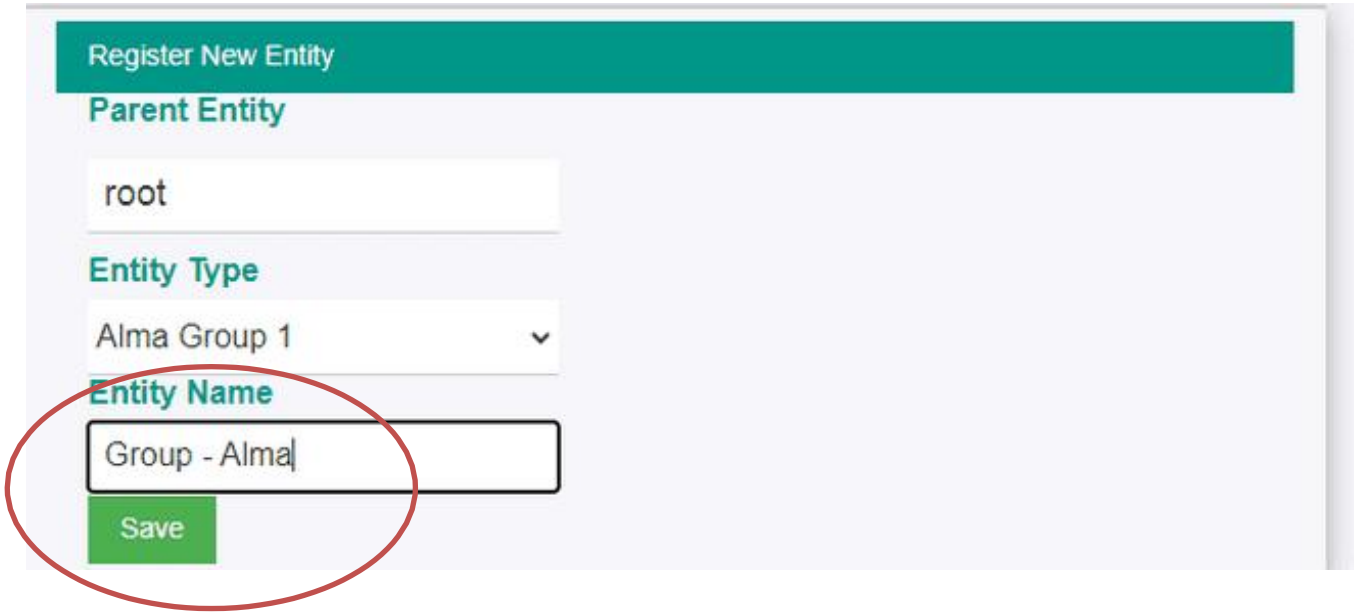
1. After you click on the ‘add entity type scheme’ butt you will get a form like this. Type your entity type and click on save button.



2. We can see the entity type you have created in this way. After that you can create entity levels by clicking on add levels button.

3. Now you can create entities. Click on ‘ add entity scheme’ button.

4. Select your “Entity Type” and type your entity name. Then Click on ‘Save’ button.

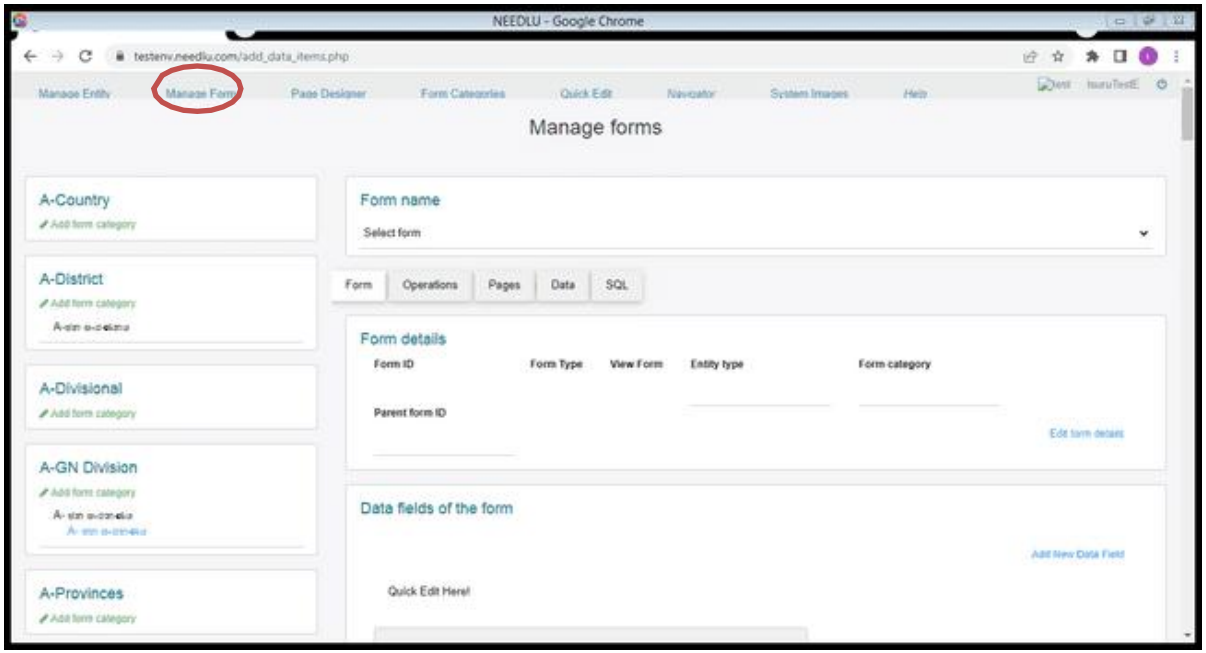


5. The required entity types and entities for the CRM have been created like this.

Entity Types

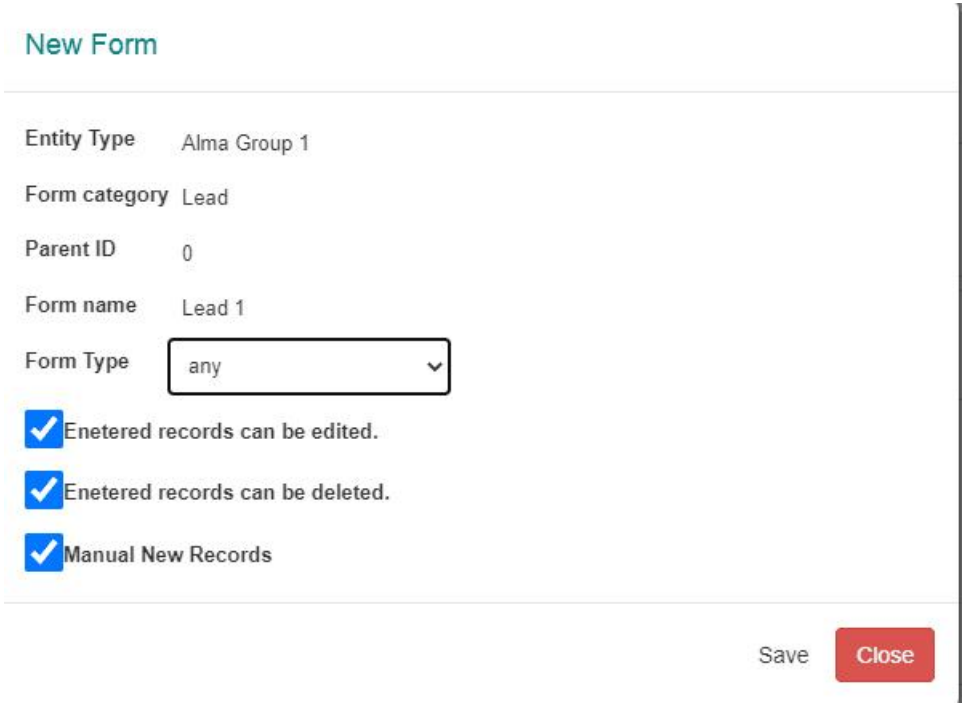
Entities

- **After successfully creating the entities, visit to the manage form page.**

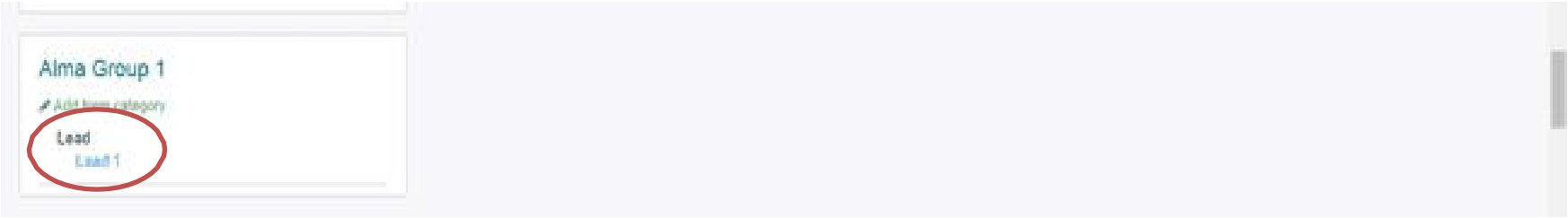


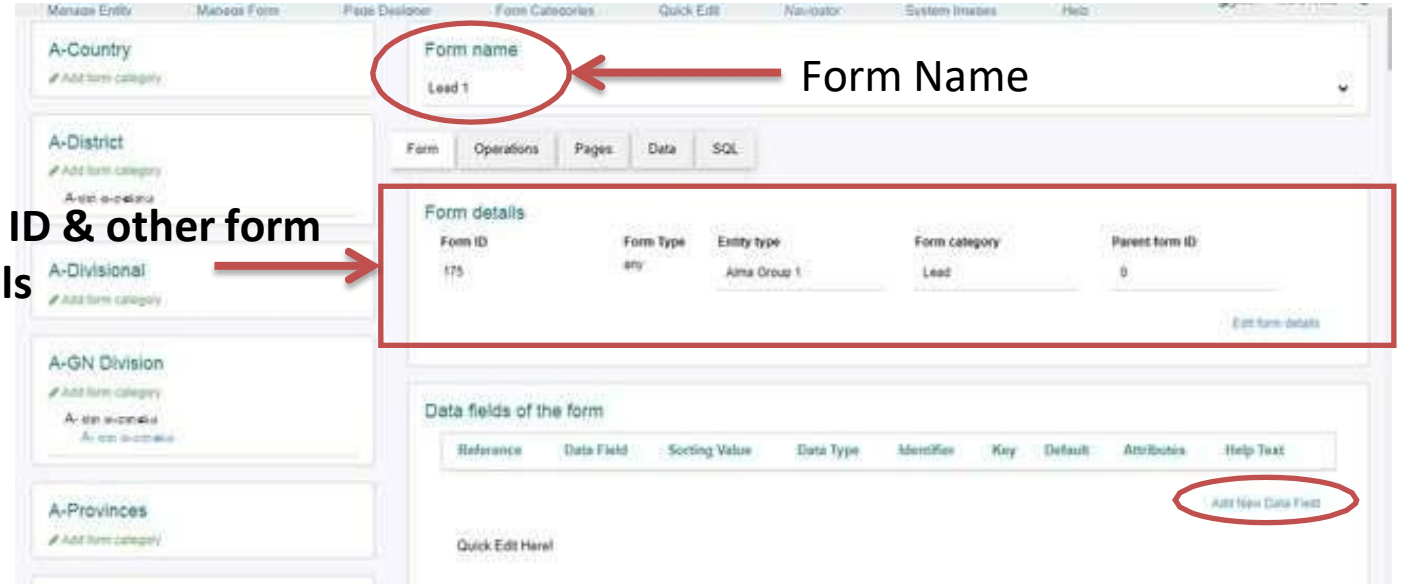
1. After that, select the entity you want to create the form and enter a form category in it.

2. Now you can create new forms. Click on ‘new form’ button enter the form name. Enter the form name and select the form type.



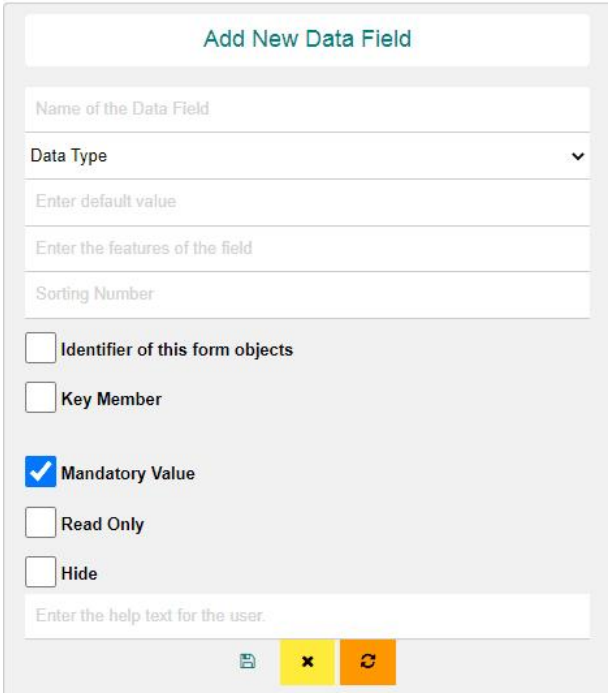
3. Select your page and open it.



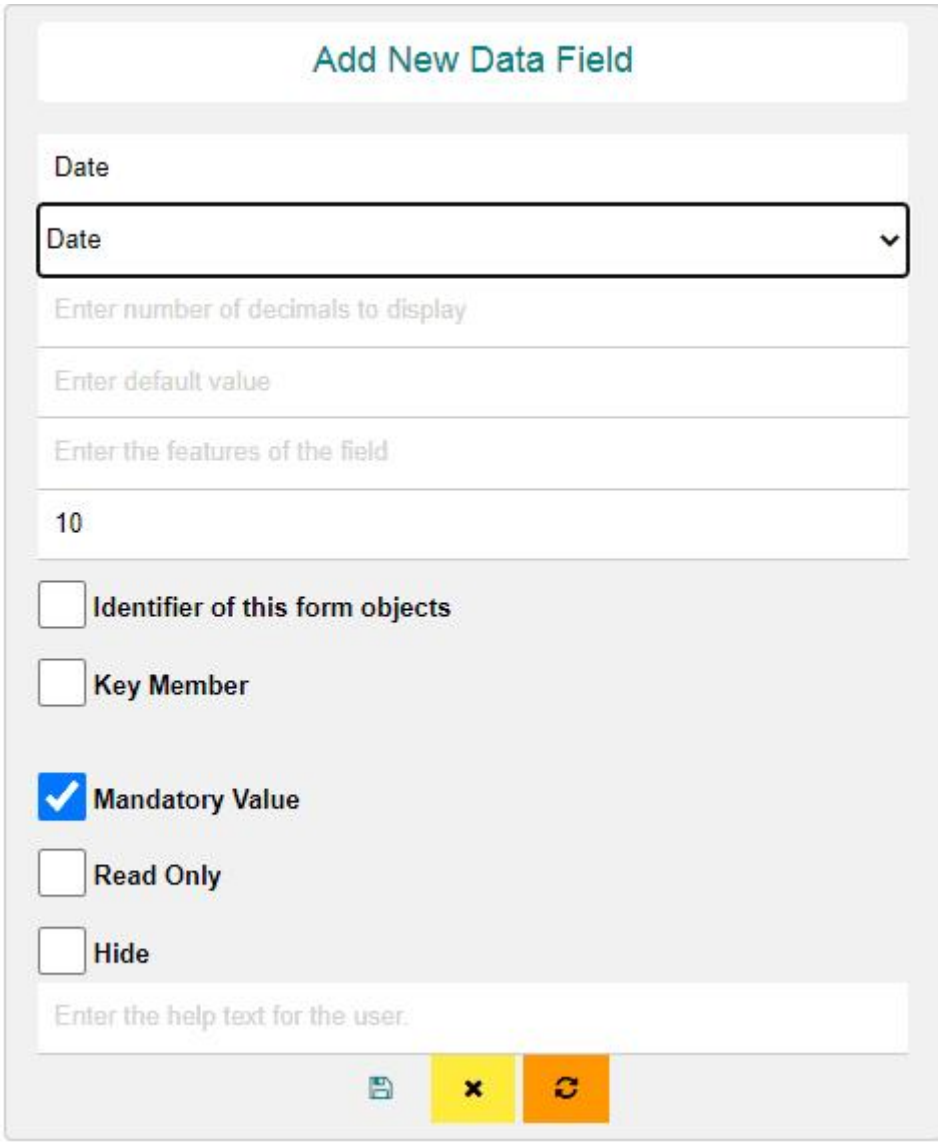


Form ID & other form Details

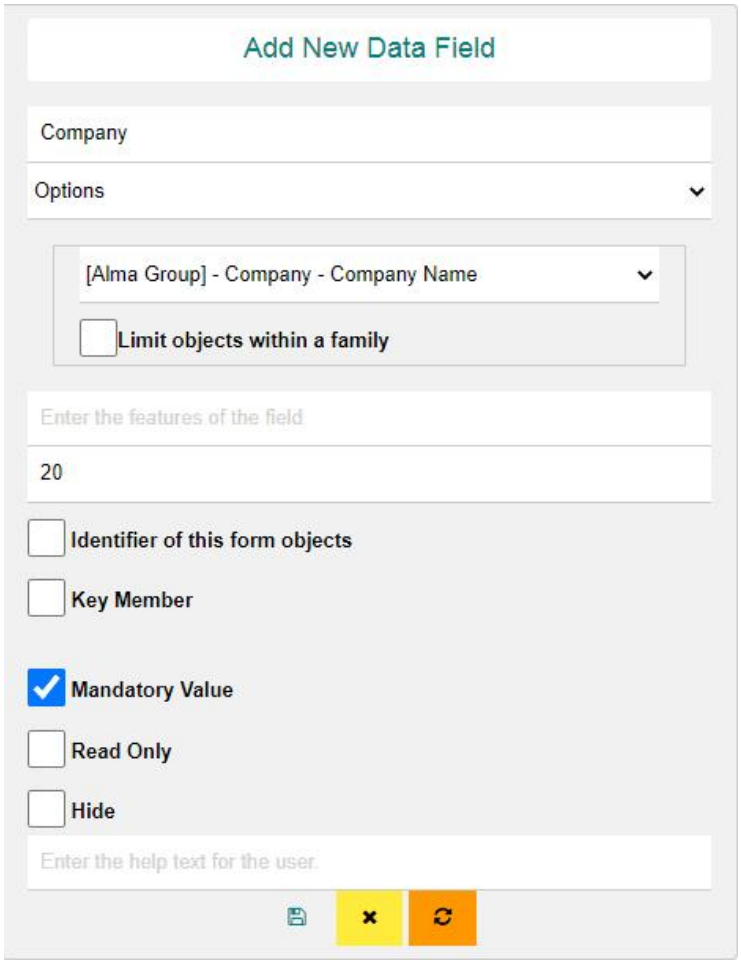
4. Click on ‘Add new data field’ button.



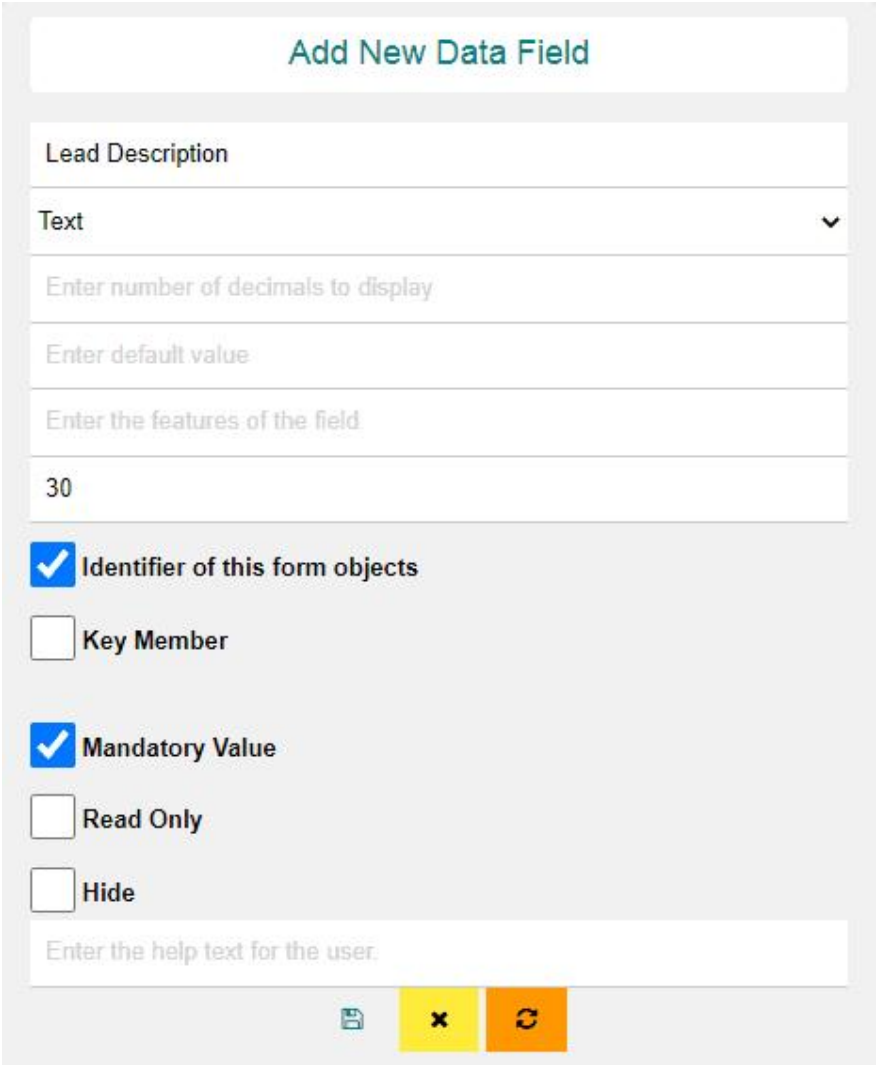
7. Enter the ‘Date’ as the first data field in this form.
After that select the data type. So select the date as data type.
You must enter a sorting number. This means in which order this data field should be displayed in this form



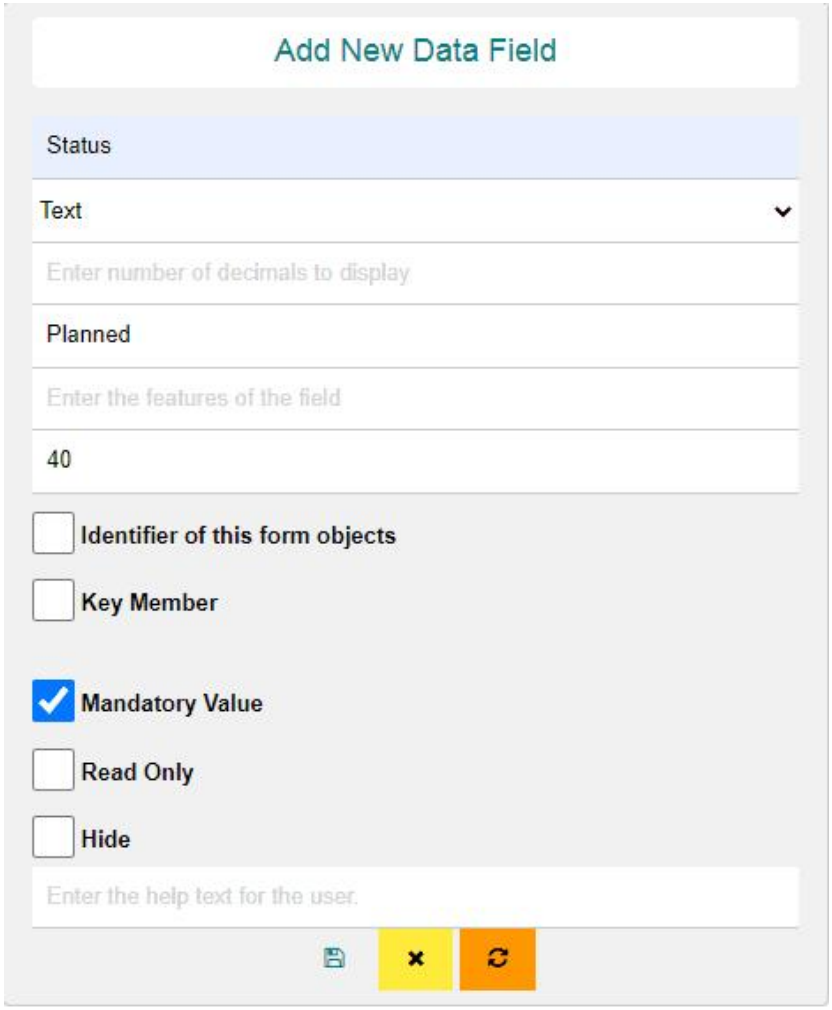
8. Enter the ‘Company’ as the second data field in this form.
After that select the data type. We should enter company details in another form. Because of that we can select the company name from that saved data. So select the ‘options’ as data type. Then you can select the company name. You must enter a sorting number. This means in which order this data field should be displayed in this form



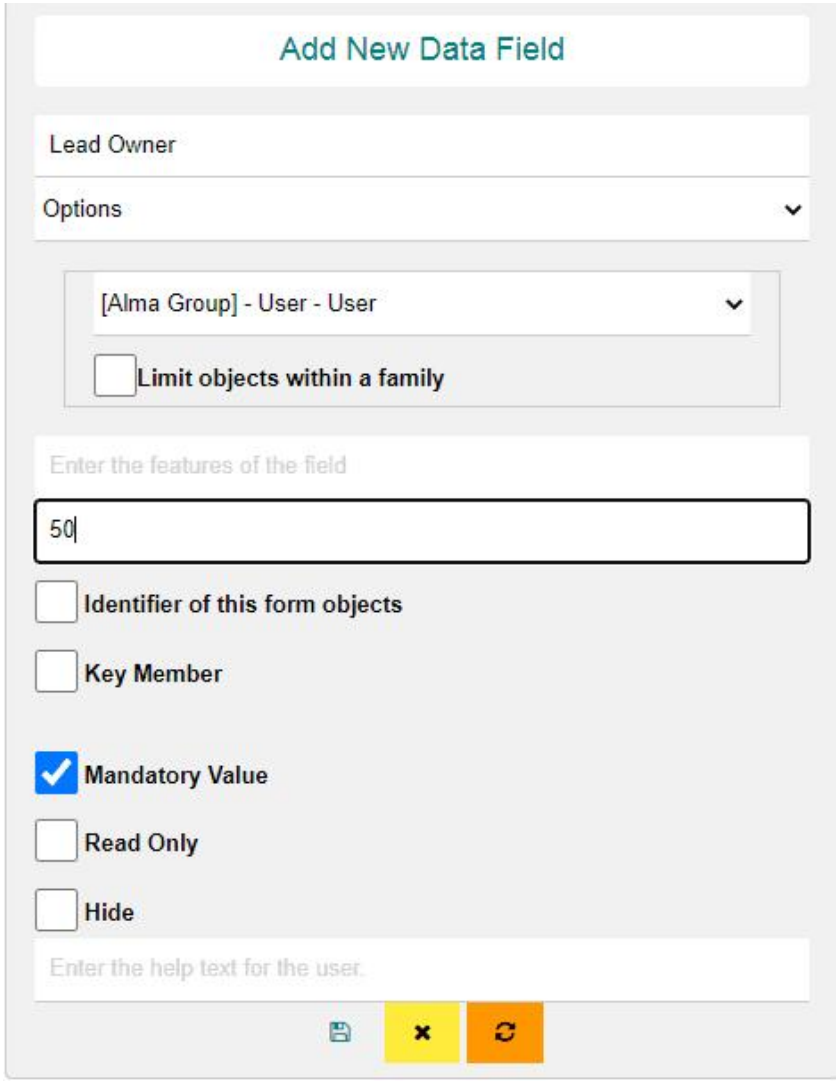
9. Enter the ‘Lead Name’ as the next data field in this form.
After that select the data type. Lead Name is a text. So select the text as data type. Enter a sorting number. Maybe we have to use the lead in another form. In such a case, we select the identifier of this form objects checkbox to get it using the options data type.



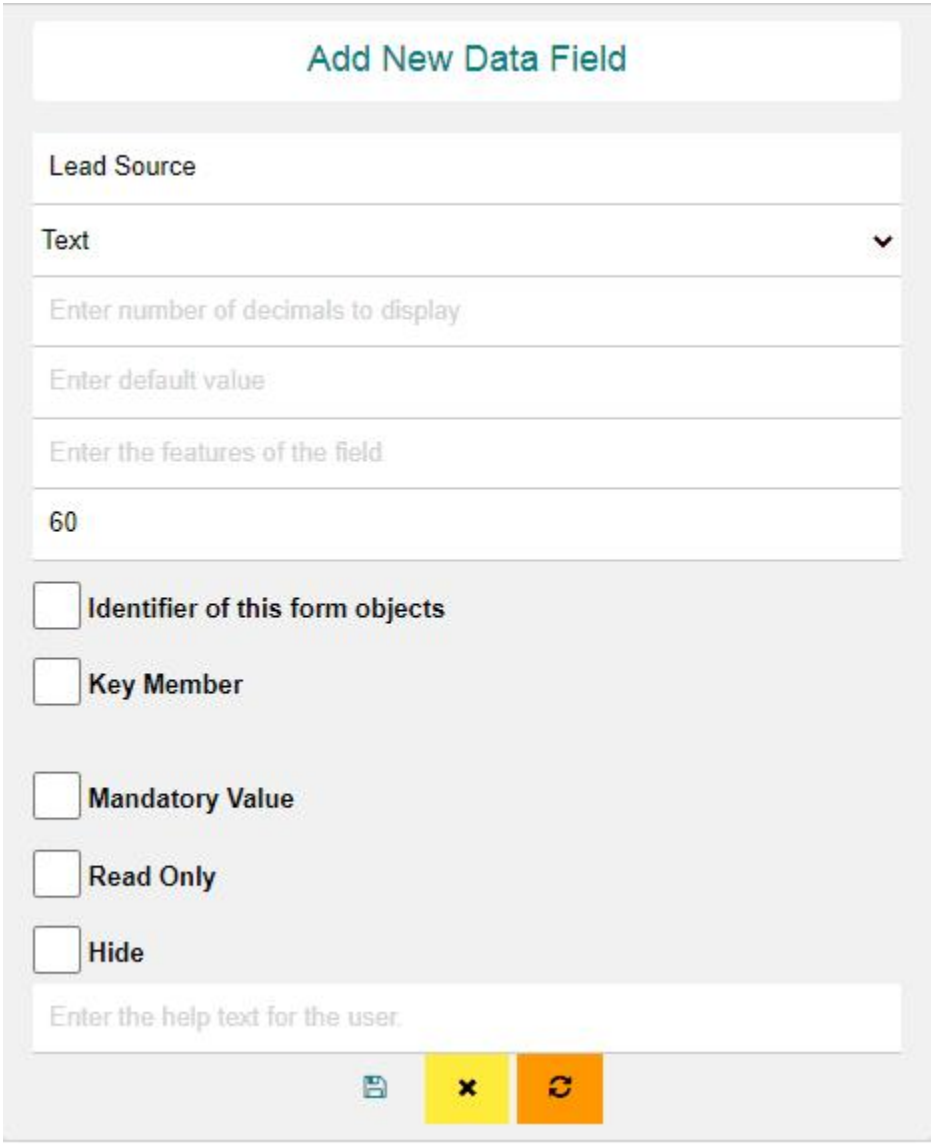
10. Enter the ‘Status’ as the next data field in this form.
After that select the data type. Status is a text. So select the text as data type.
Enter a sorting number. We use this status field to update the progress. When we save this lead we need to update status field as ‘planned’. So you can set the default value as ‘Planned’.



11. Enter the ‘Lead owner’ as the next data field in this form.
After that select the data type. Lead Owner is an user of this application. So select the ‘options’ as data type. Before that you must have entered users in another form.

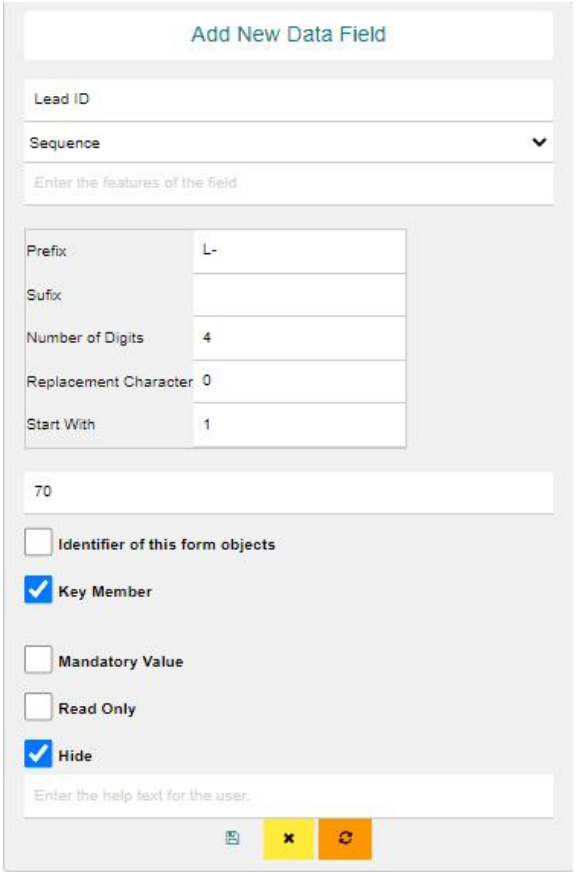


11. Enter the ‘Lead Source’ as the next data field in this form.
After that select the data type. Lead Source is a text field. So select the ‘text’ as data type.



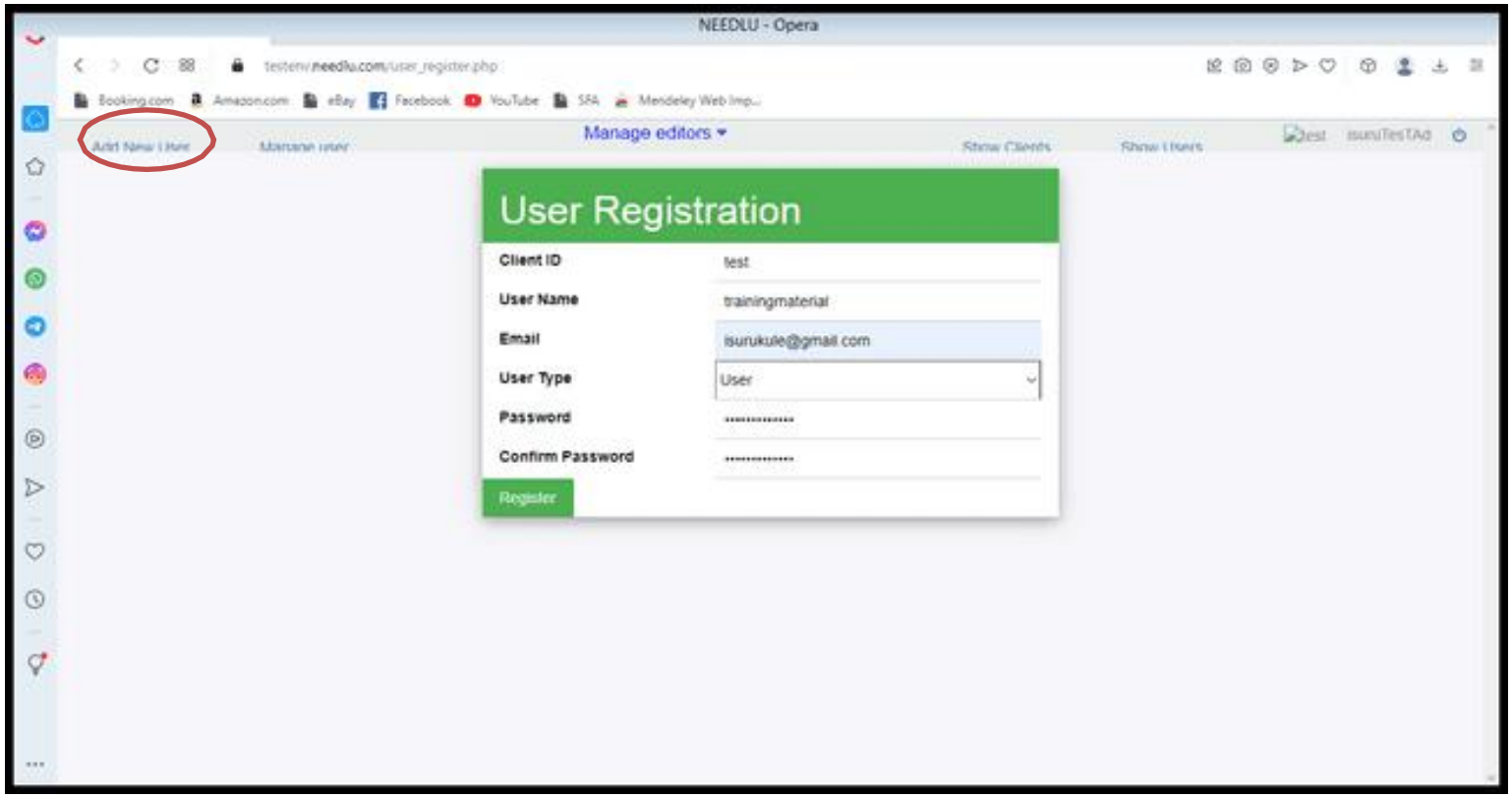
- **Every form you create to enter data must have a data field that does not enter the same value twice. We should introduce that data field as a 'key member' to the system and thereby prevent data duplication.**

- In this 'Lead form' we create a Lead ID field for preventing data duplication. We use 'Sequence' as the data type and it generate codes automatically. Also we do not use this data field for any purpose. So we can hide it.



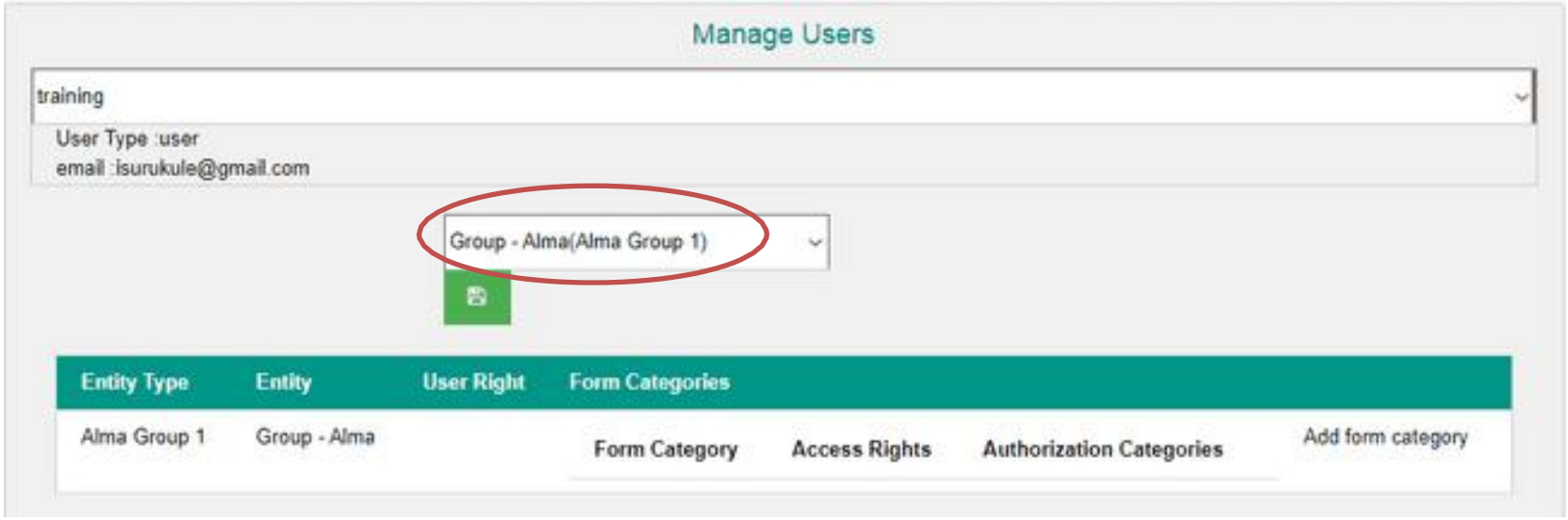
- **Now let's see how to enter data as a user in the form we created.**

1. First log in to the Needlu developer tool using admin role username and password.
2. Click on 'Add new user' and register your user.

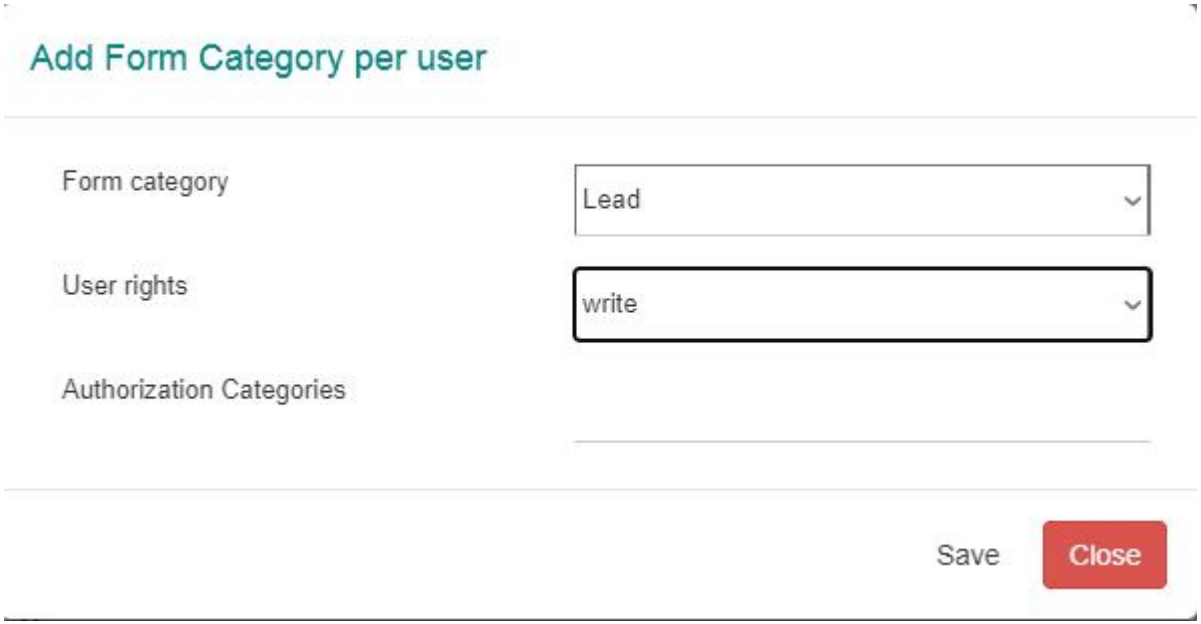


3. Then click on ‘Manage User’. Select your user’s username from the list. After that click on the ‘New Button’ (Green color Plus Mark)

4. Select your entity and save it.

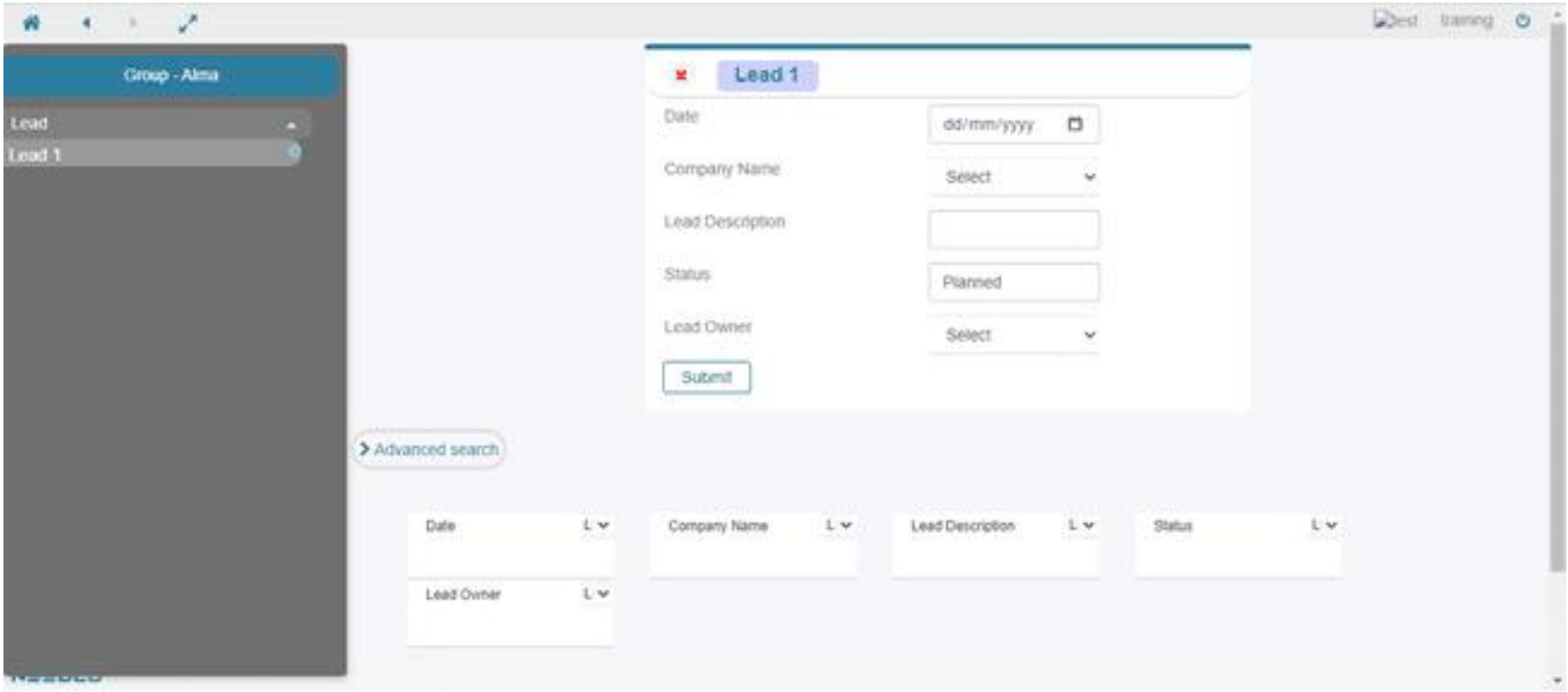


5. After that click on ‘Add form category’, select the form category and save it.
If you needs to give access to enter data you should select ‘write’ as user rights.



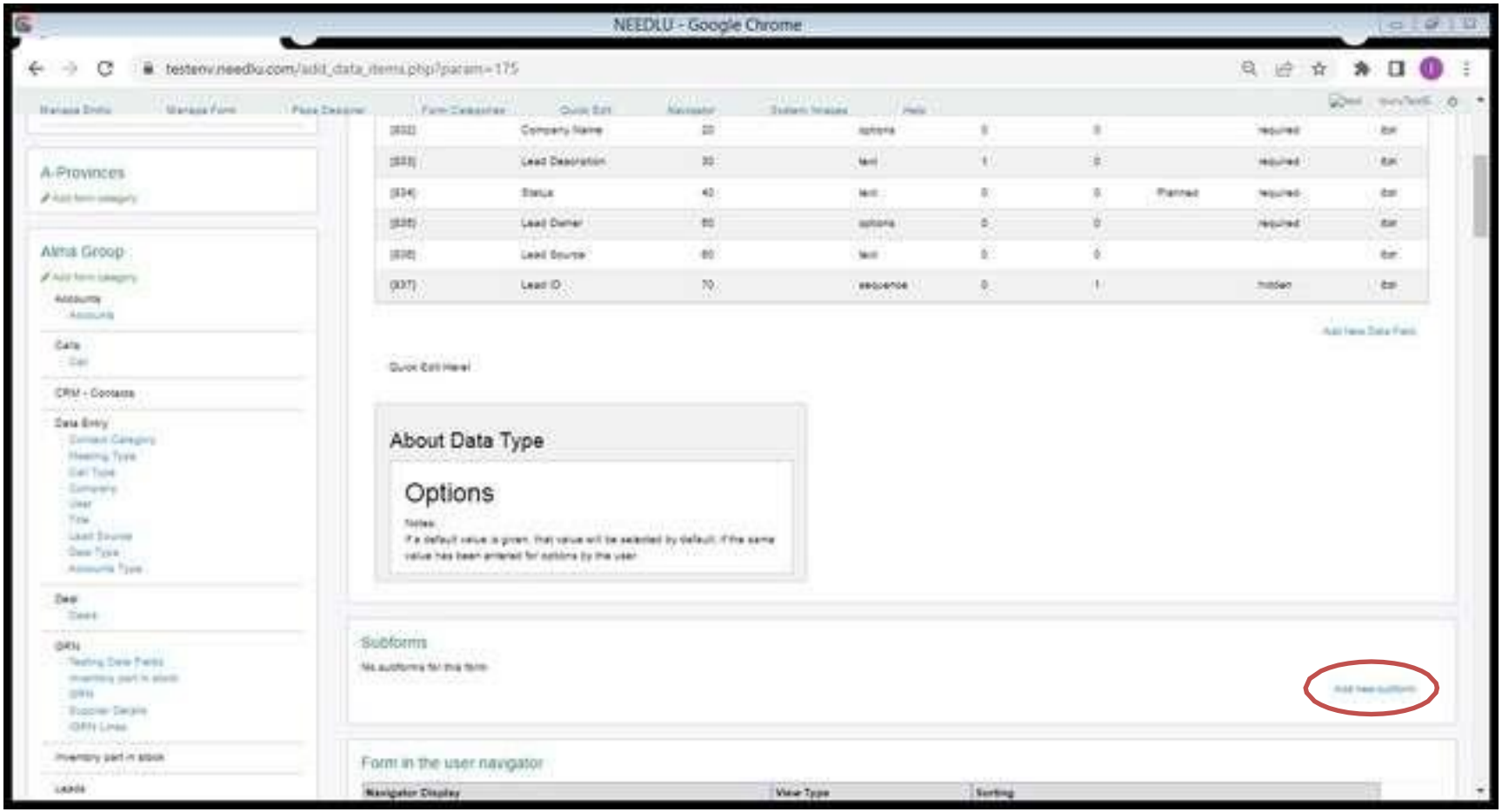
6. Next you access the CRM software as a user.

- **This is the form we created using Needlu Software developer tool.**

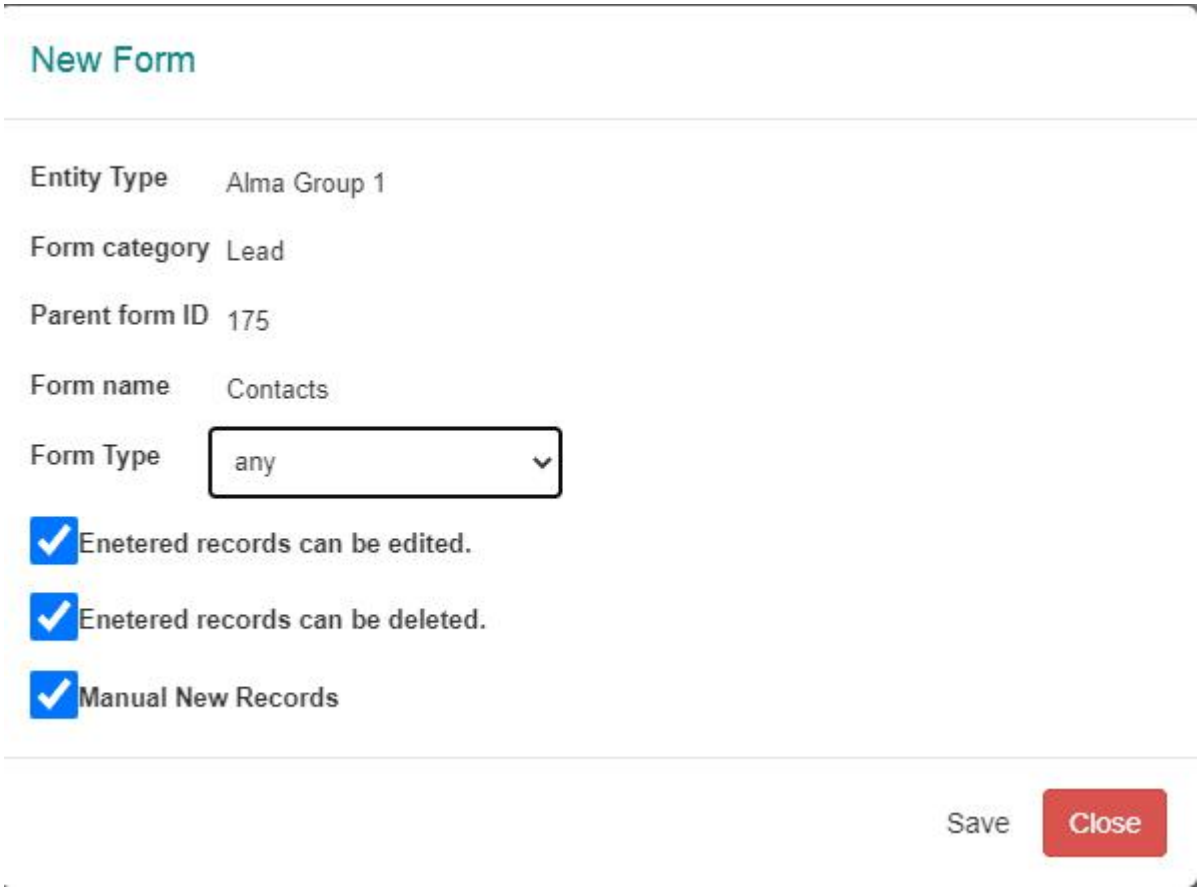


- **We have now successfully developed our ‘Lead’ form. Next, let’s create the sub-form called ‘Contacts’ to enter the relevant contacts.**

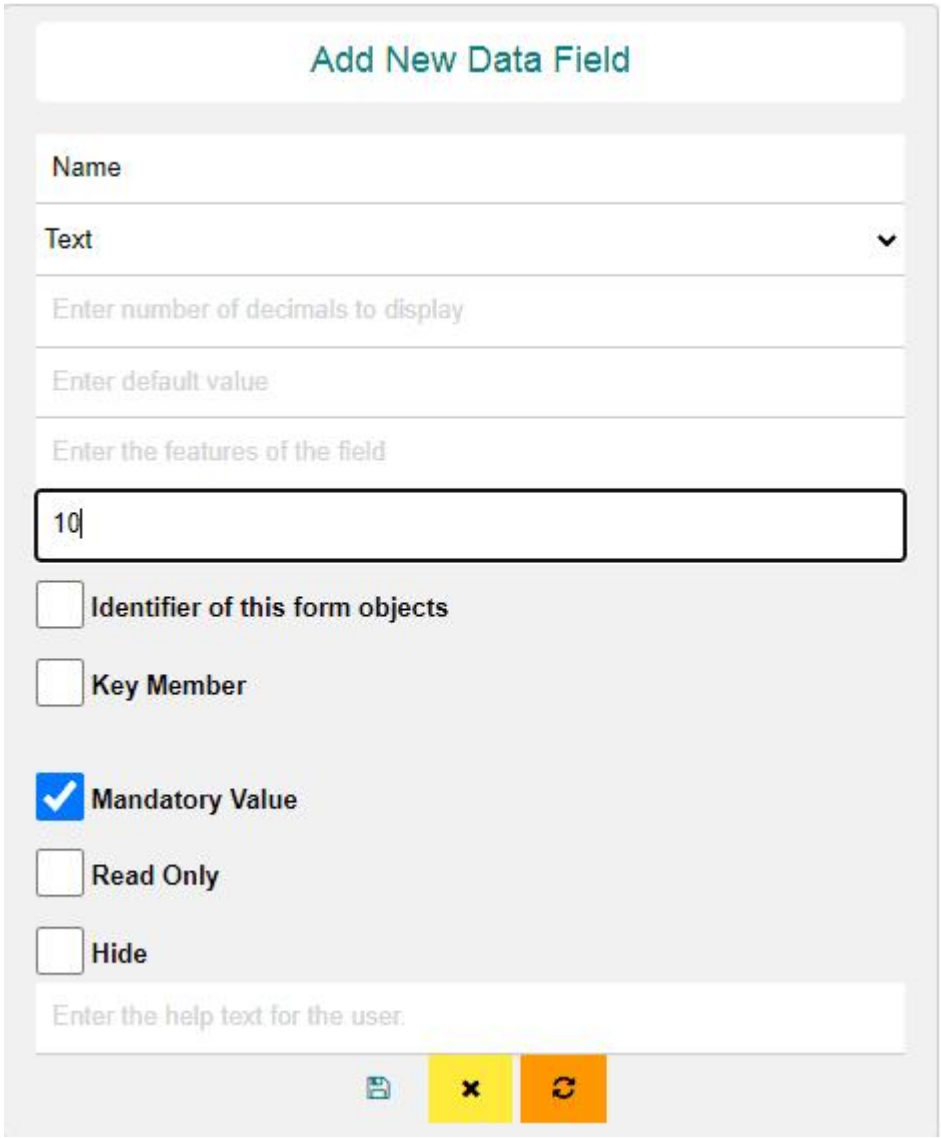
1. Click on ‘Add new sub-form.



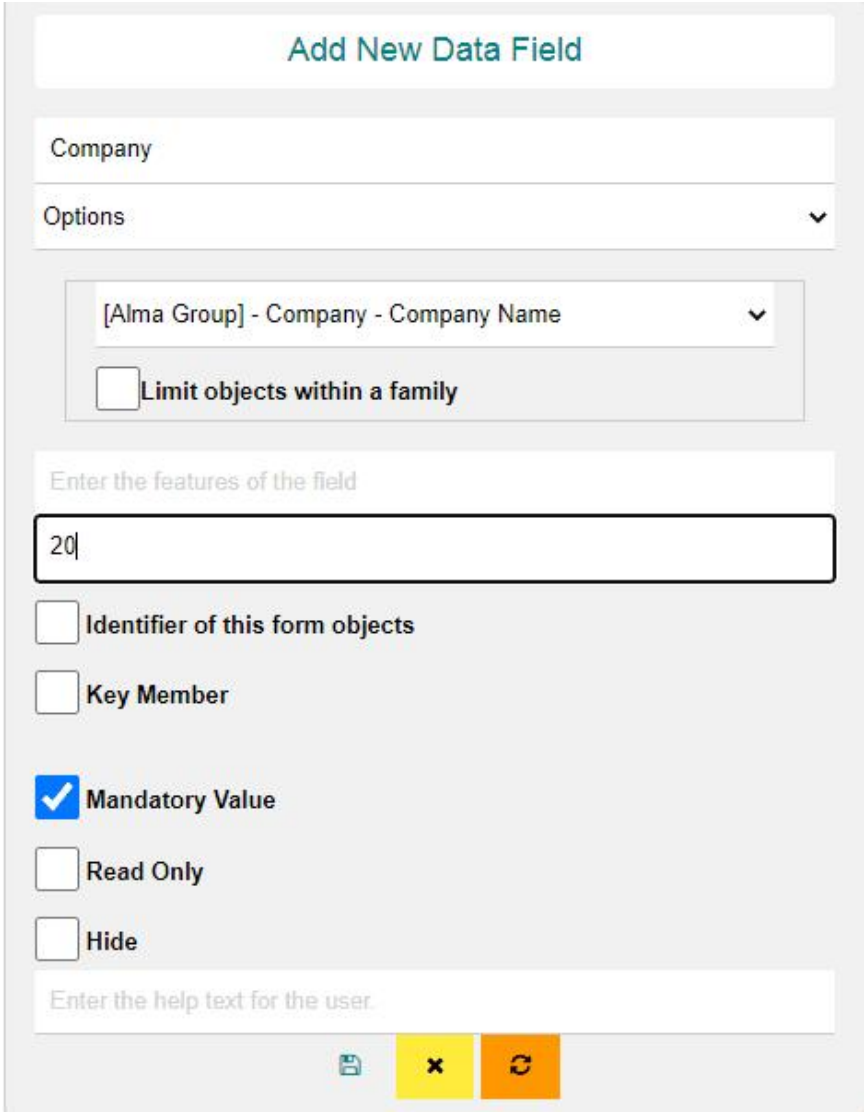
2. Enter the form name and save it.



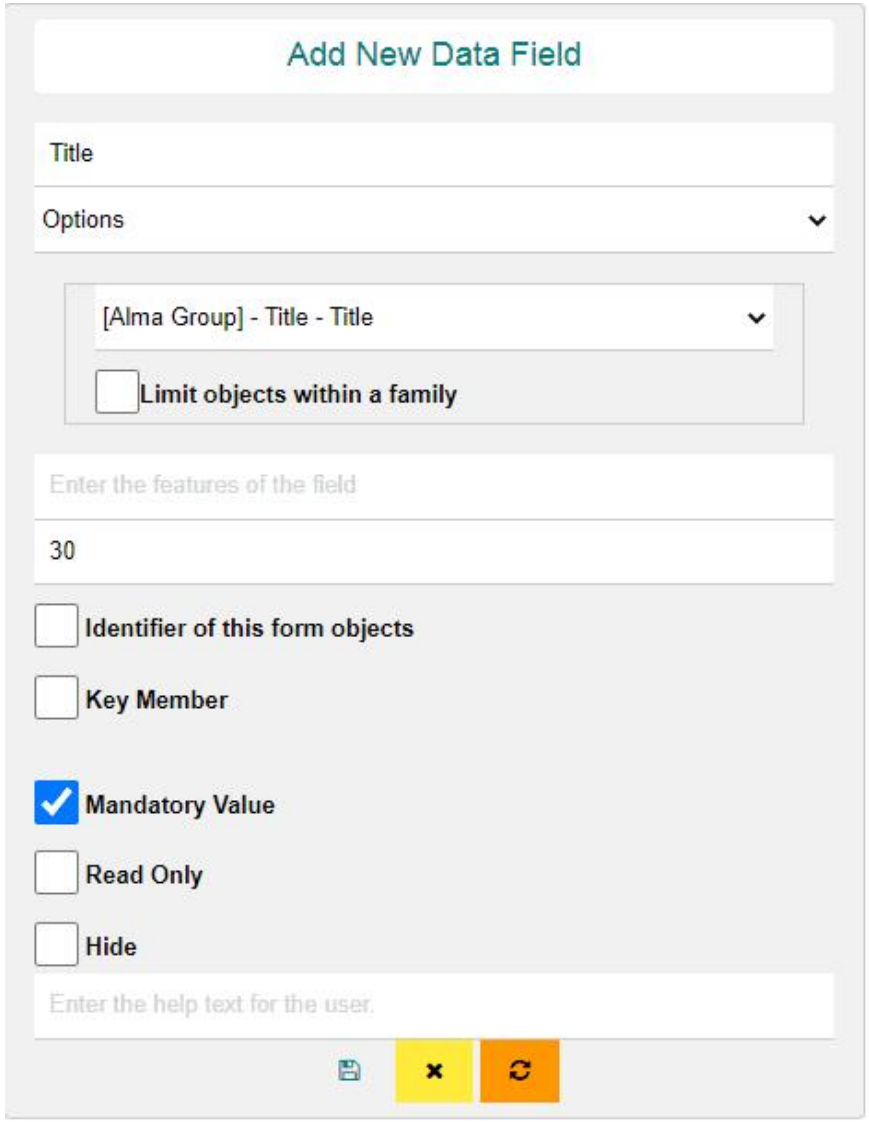
3. Let's create the Name as the first data field in this sub-form. Name is a text. So select 'text' as data type.



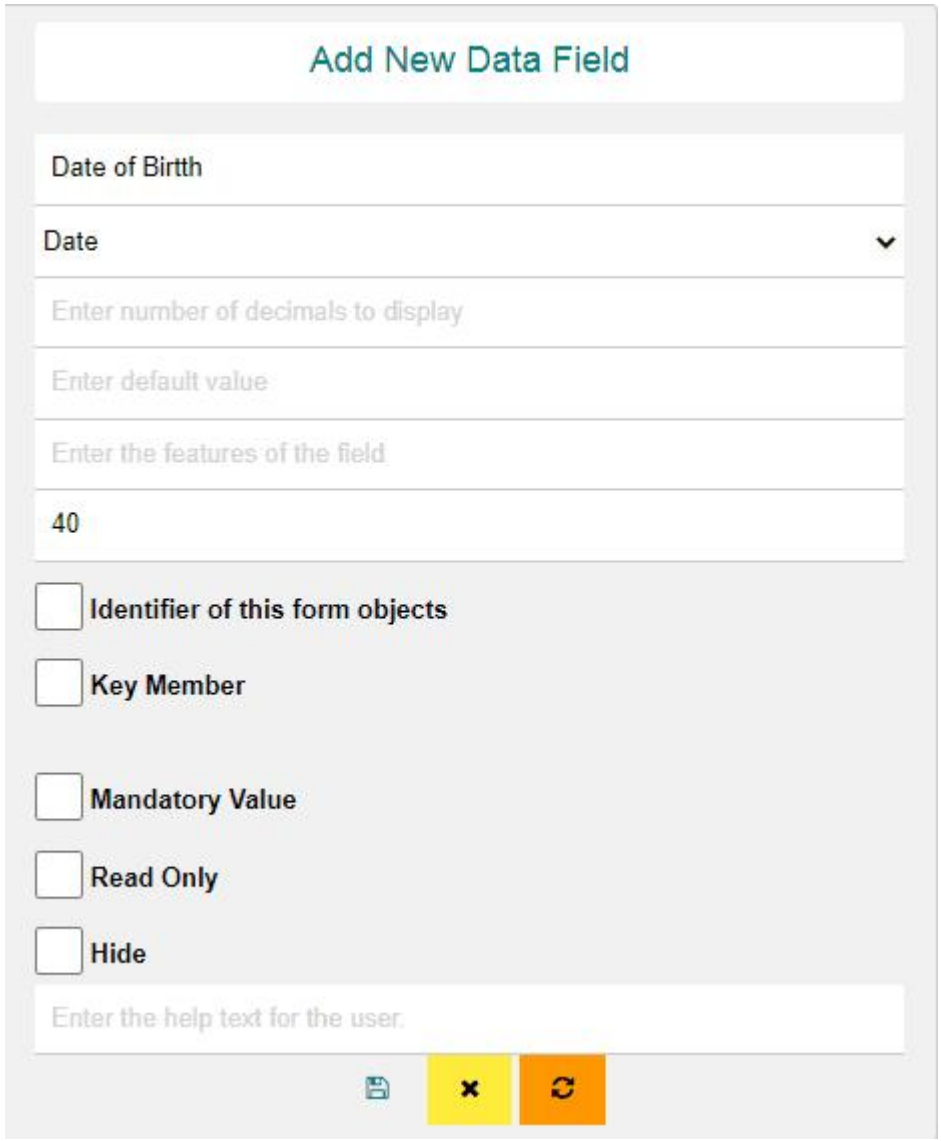
4. Create Company as the next data field. We already saved company information. So select ‘option’ as the data type.



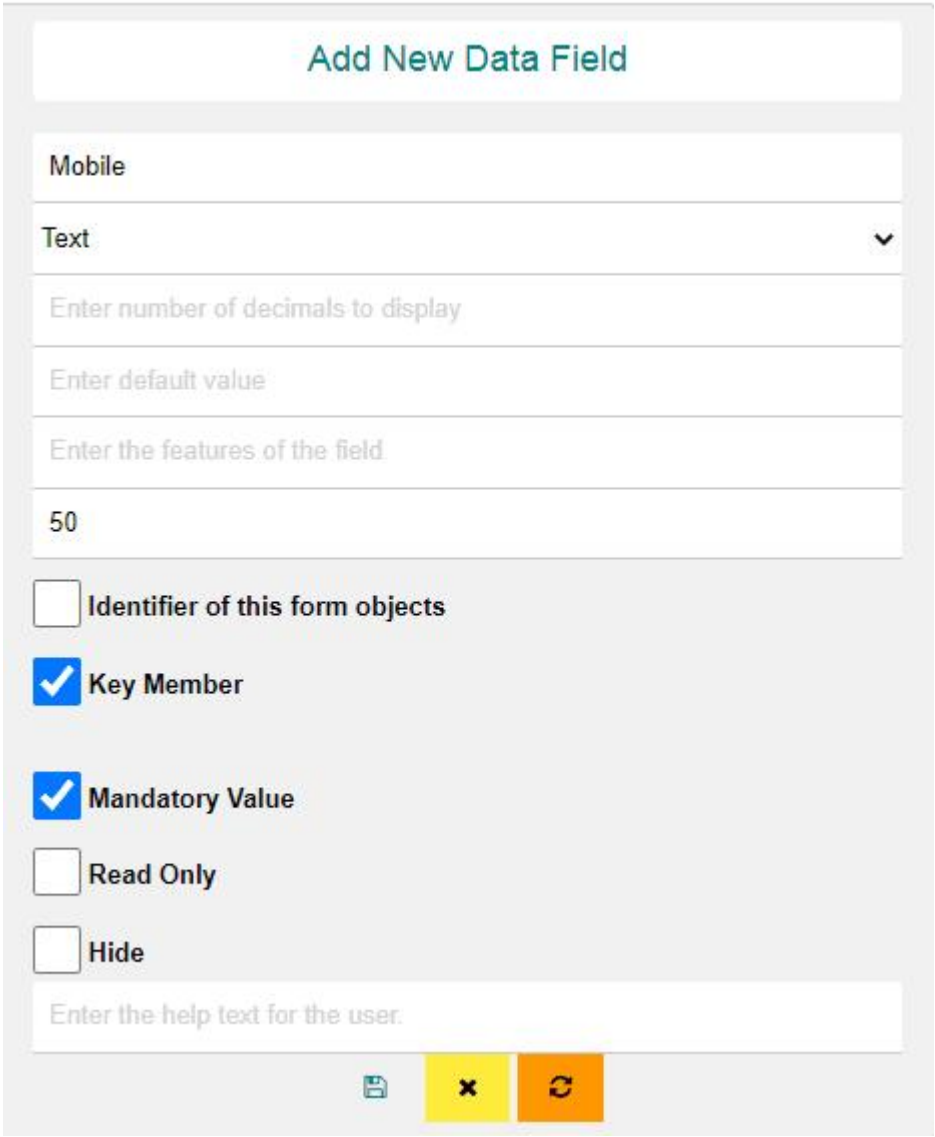
5. Create title as the next data field. We already saved title information. So select ‘option’ as the data type.



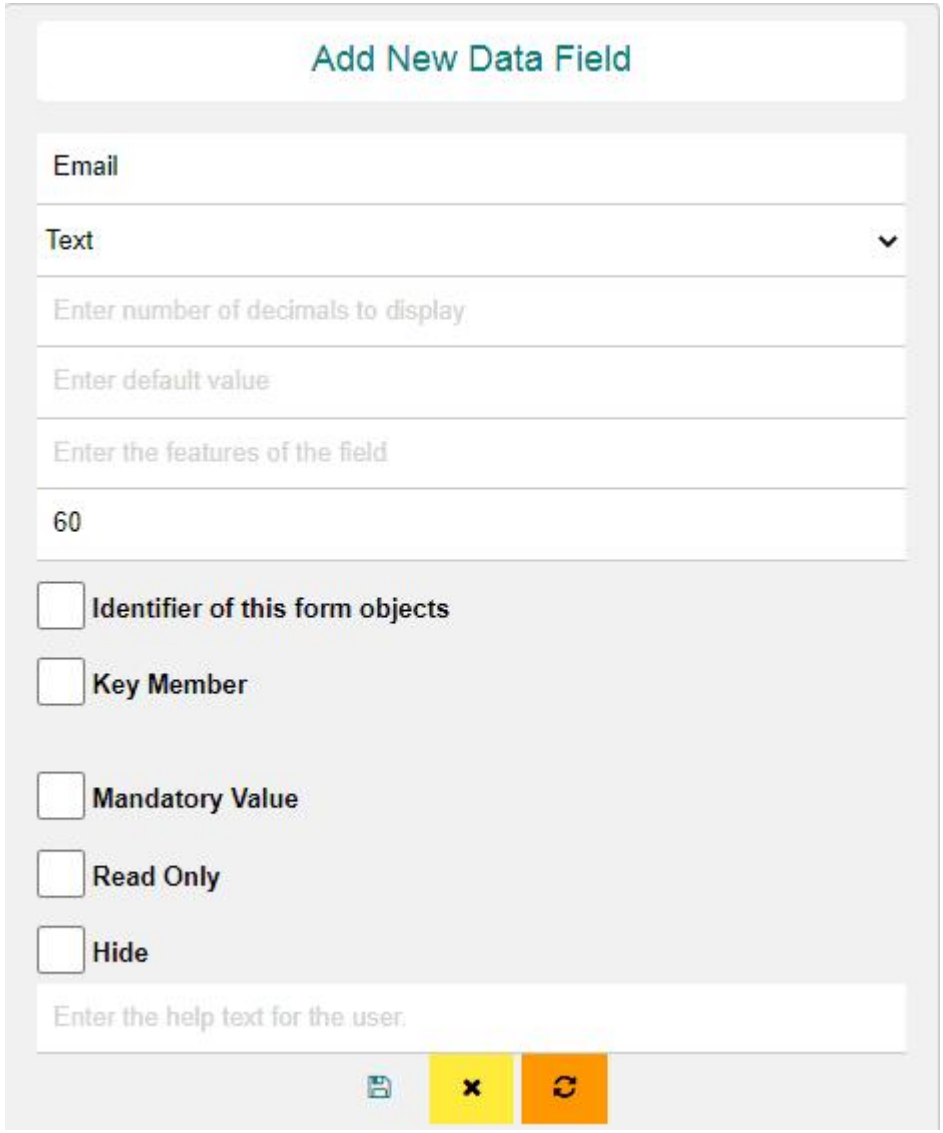
6. Create Date of birth as the next data field. Set ‘date’ as the date type.



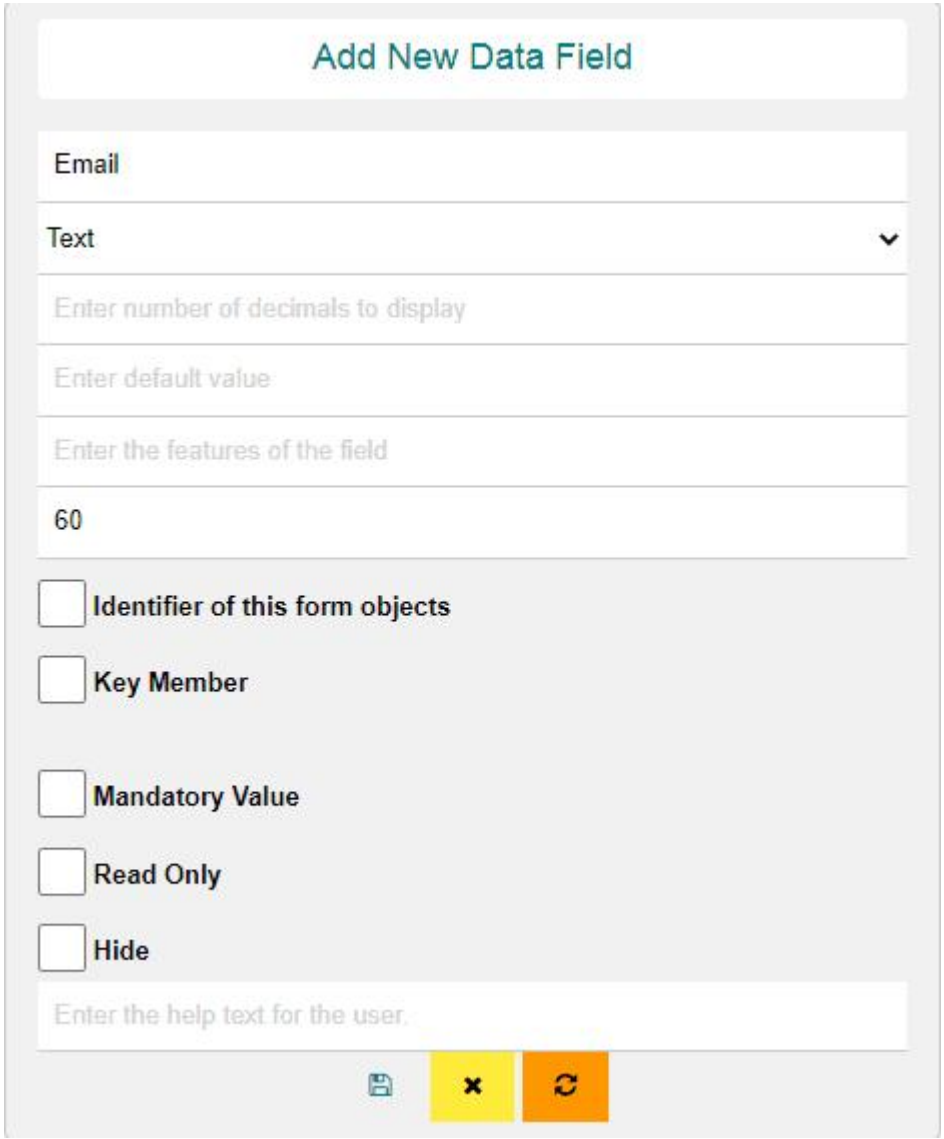
7. Create Mobile as the next data field. So select ‘text’ as the data type. Also this field is a key member.



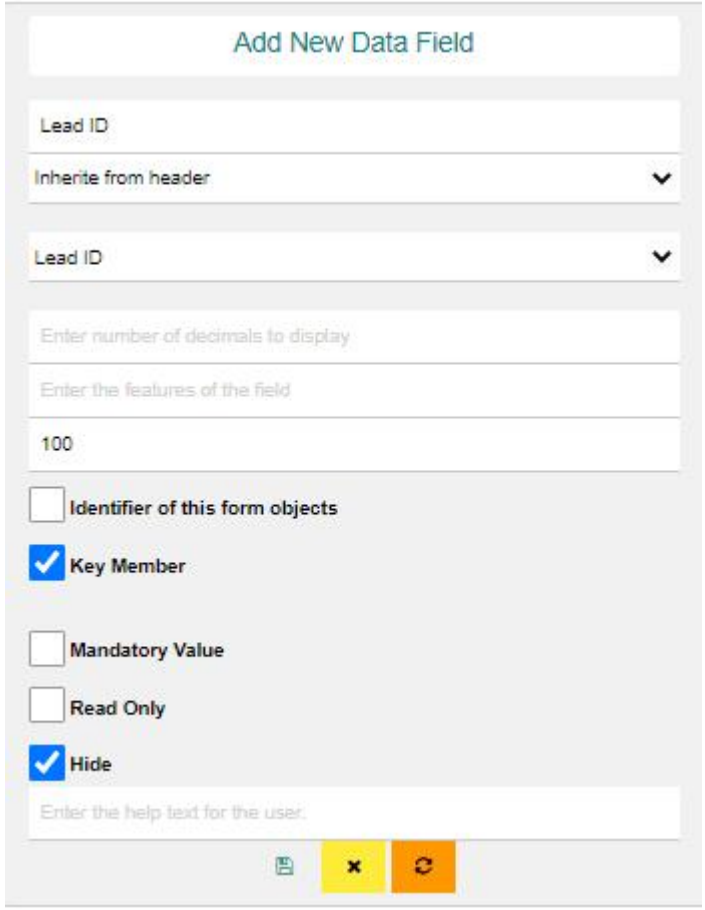
8. 4. Create Email as the next data field. So select ‘text’ as the data type.



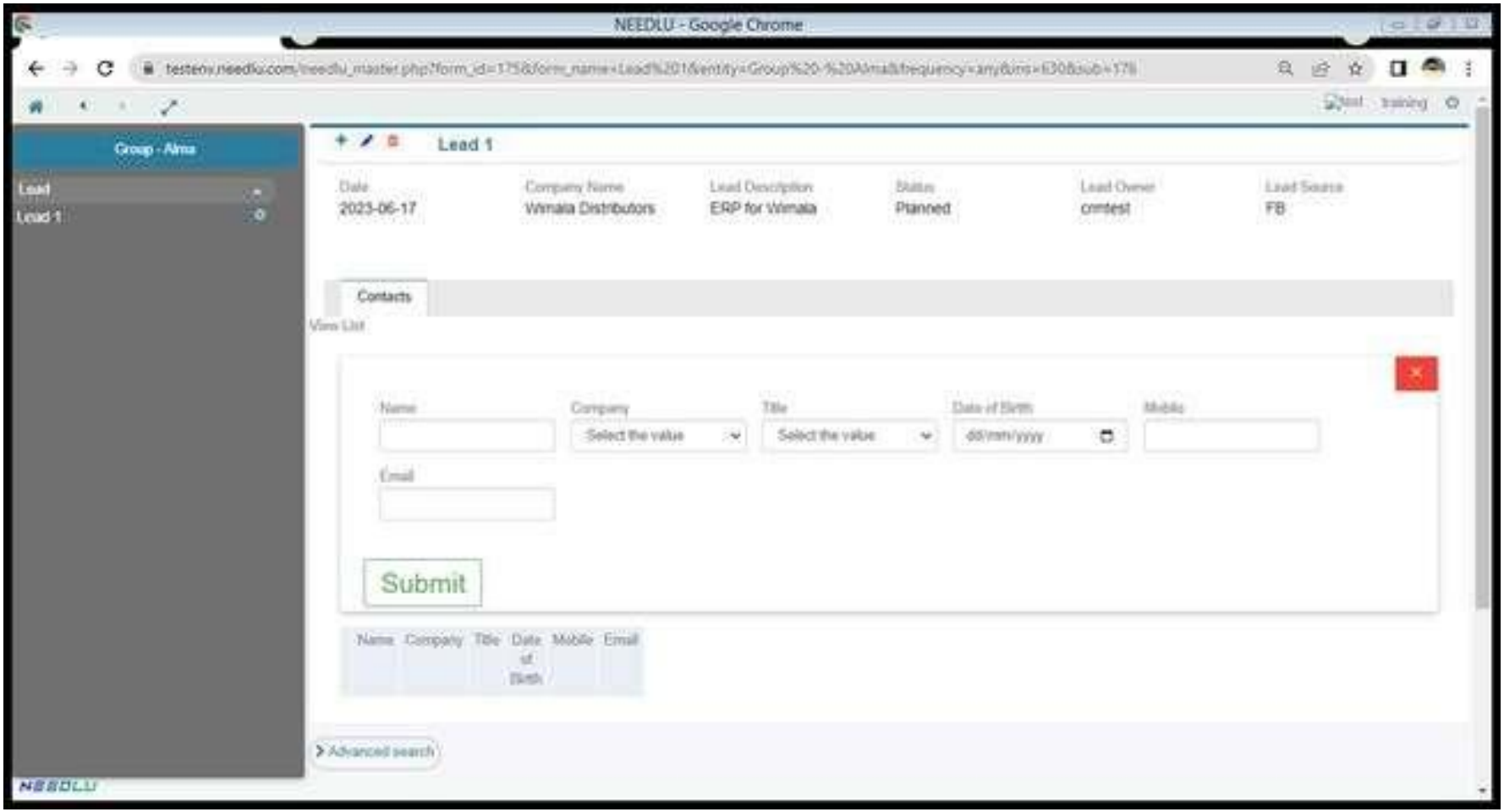
9. Create Address as the next data field. User need more space to type the address. So select 'text box' as the data type.



- **When you create a sub-form, you should inherit the key fields to sub-form from its main form. Use the inherit from header data type.**



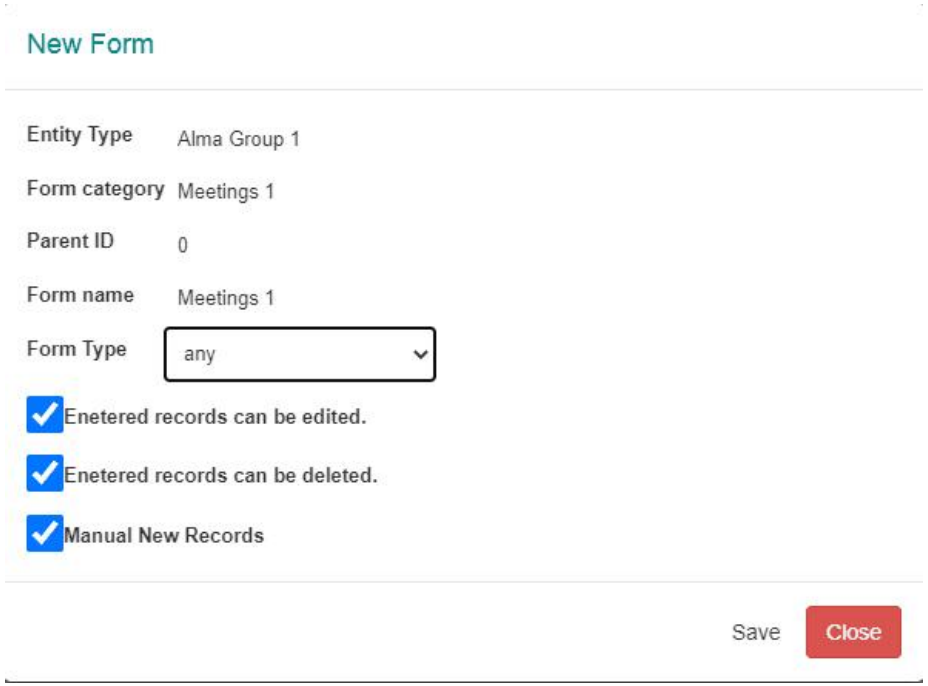
- **Now our sub-form has been successfully created. See how it appears to the user.**



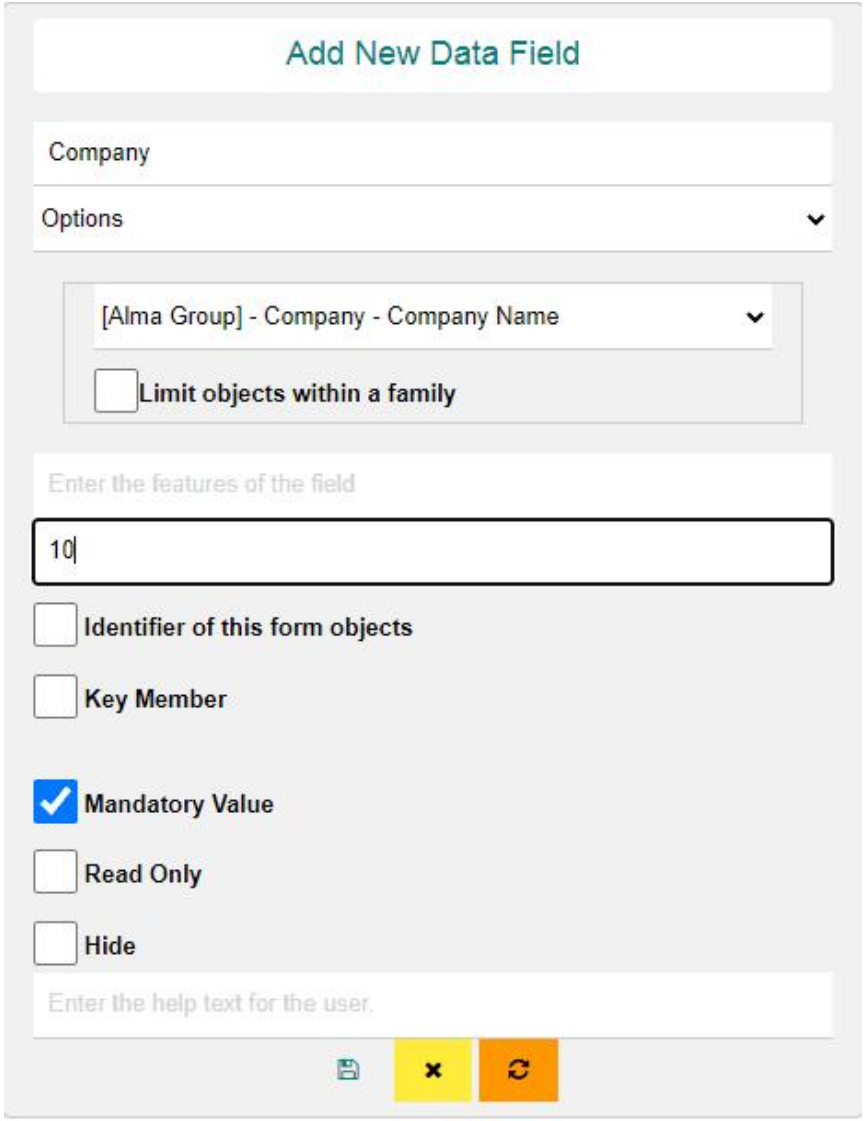
➢ Let's create a form to schedule meetings.

1. Create the form category called 'Meetings 1'

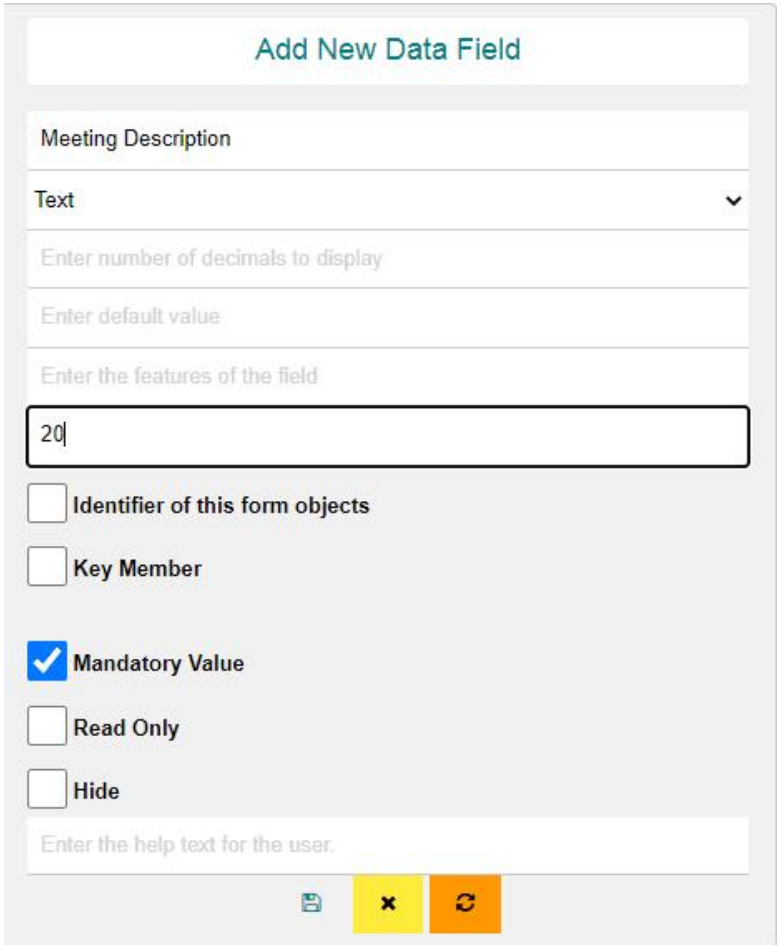
2. Create a new form under the meetings category.



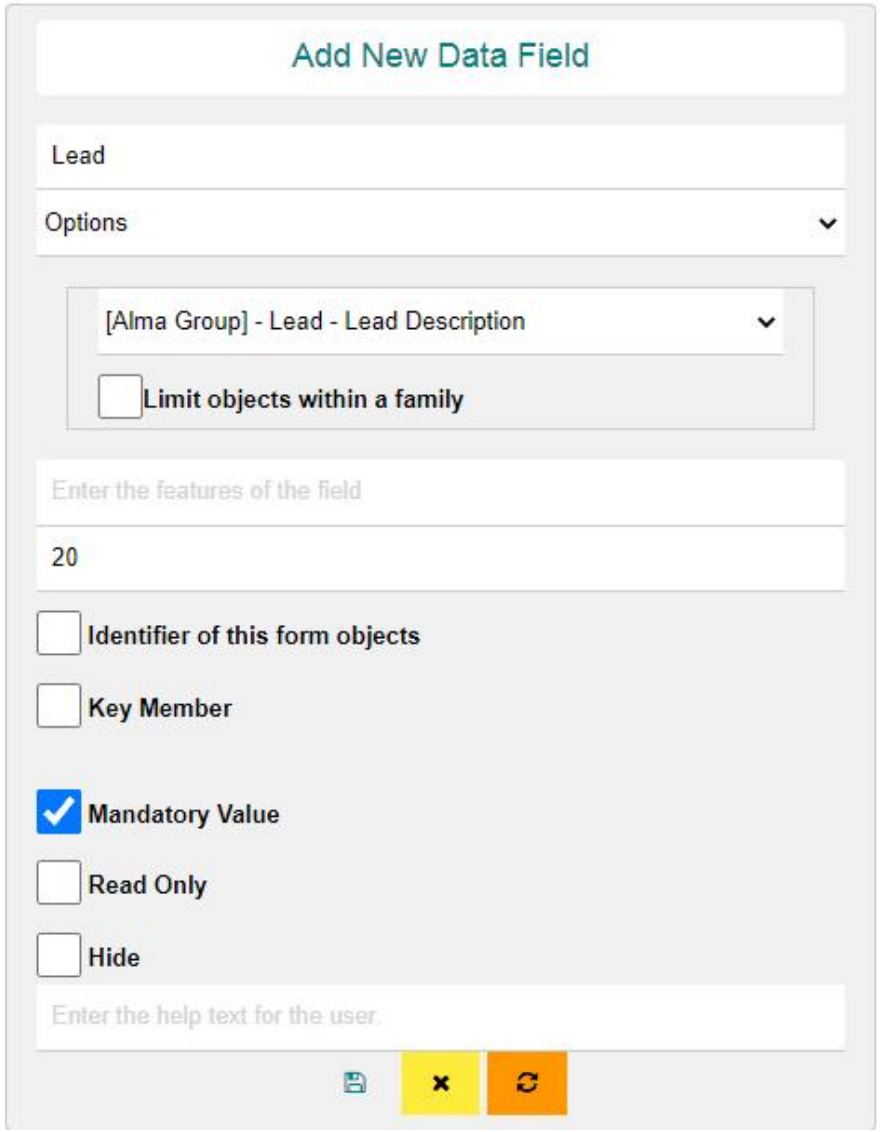
3. Create Company as the first data field. We already saved company information. So select ‘option’ as the data type.



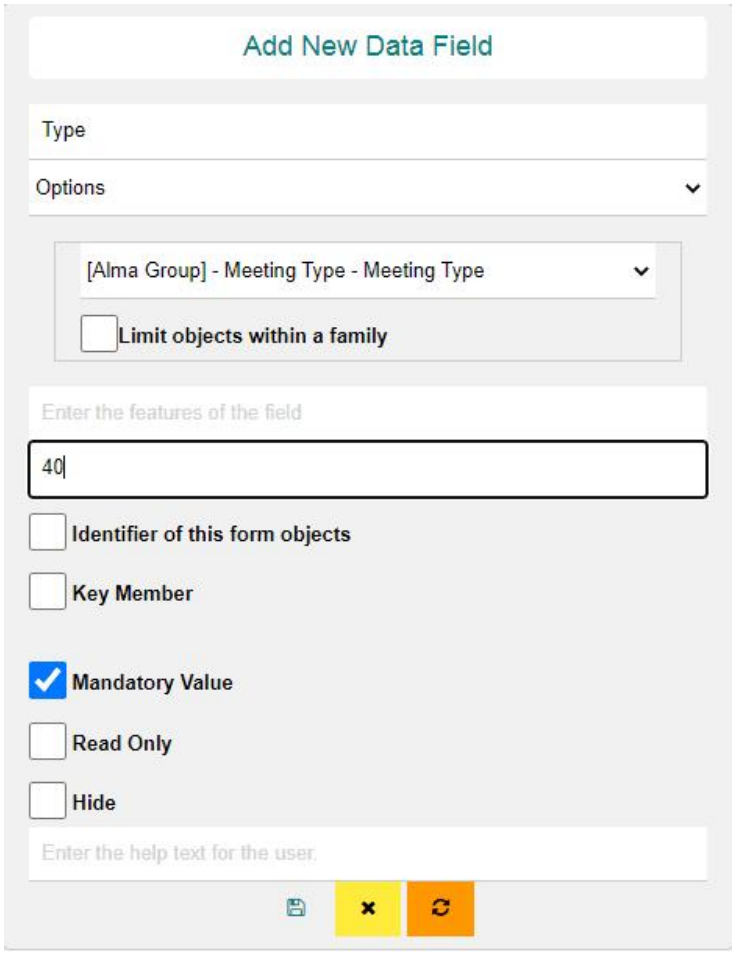
4. Create Meeting Description as the next data field. So select ‘text’ as the data type.



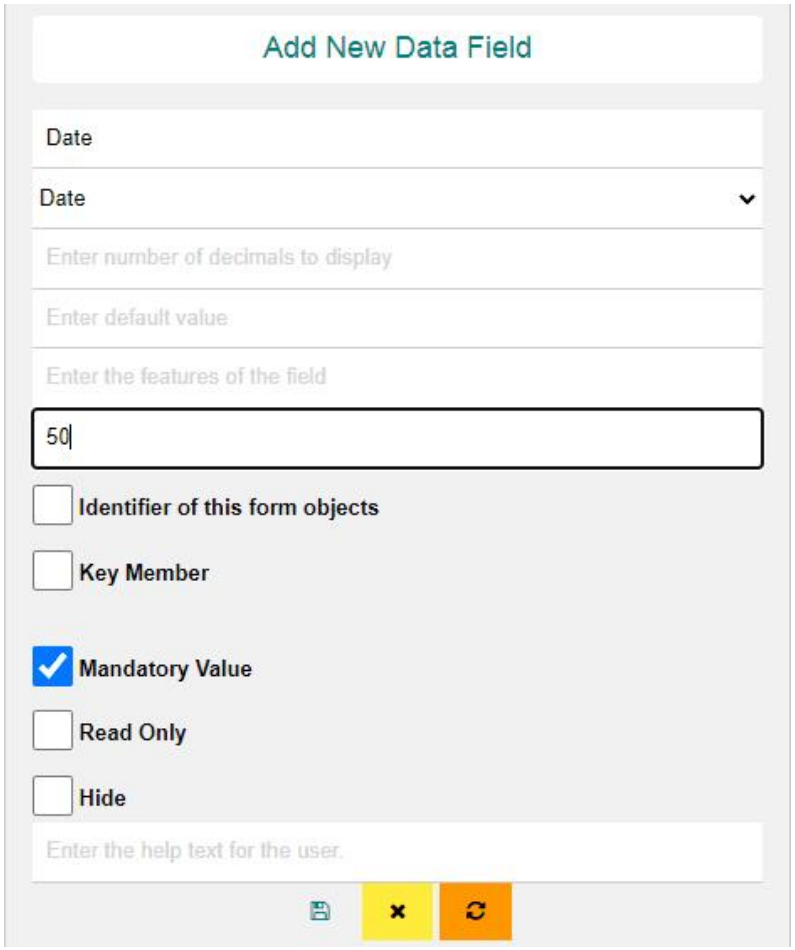
5. Create Lead as the next data field. We already saved lead information. So select ‘option’ as the data type.



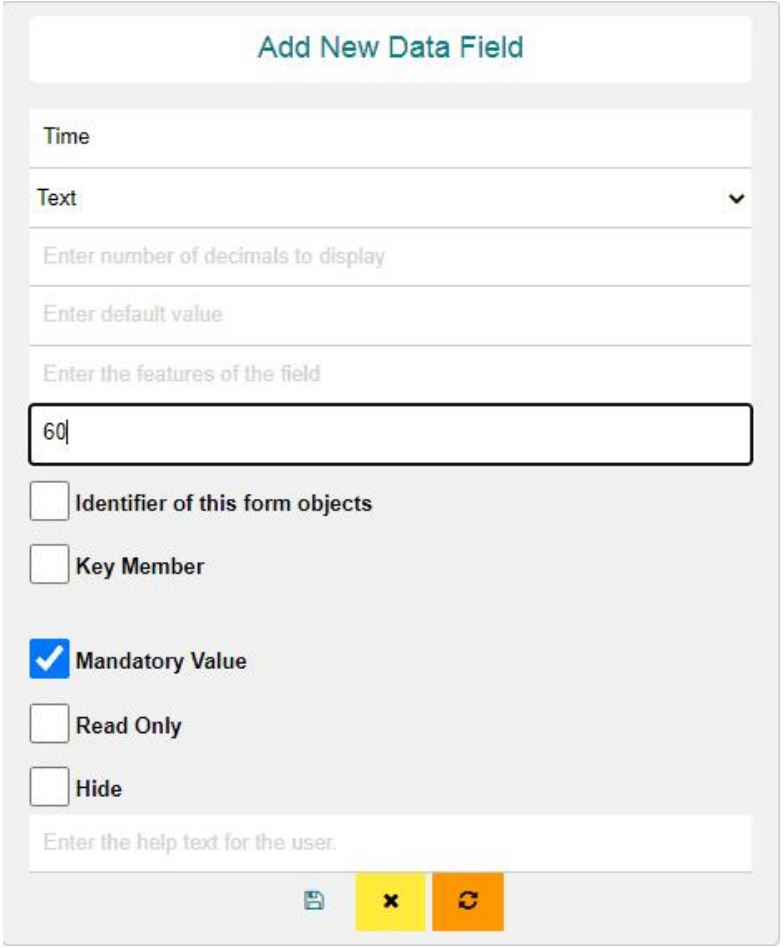
6. Create Type as the first data field. There are only two options to select. So select ‘option’ as the data type.



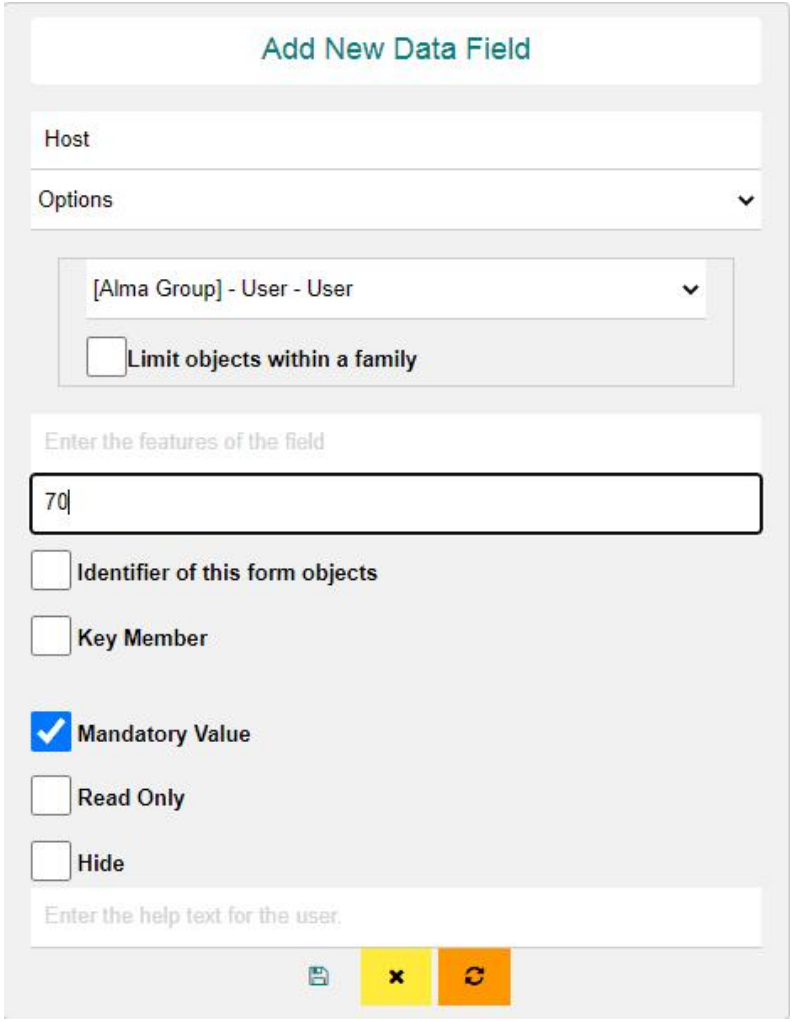
7. Create Date as the next data field. So select ‘date’ as the data type.



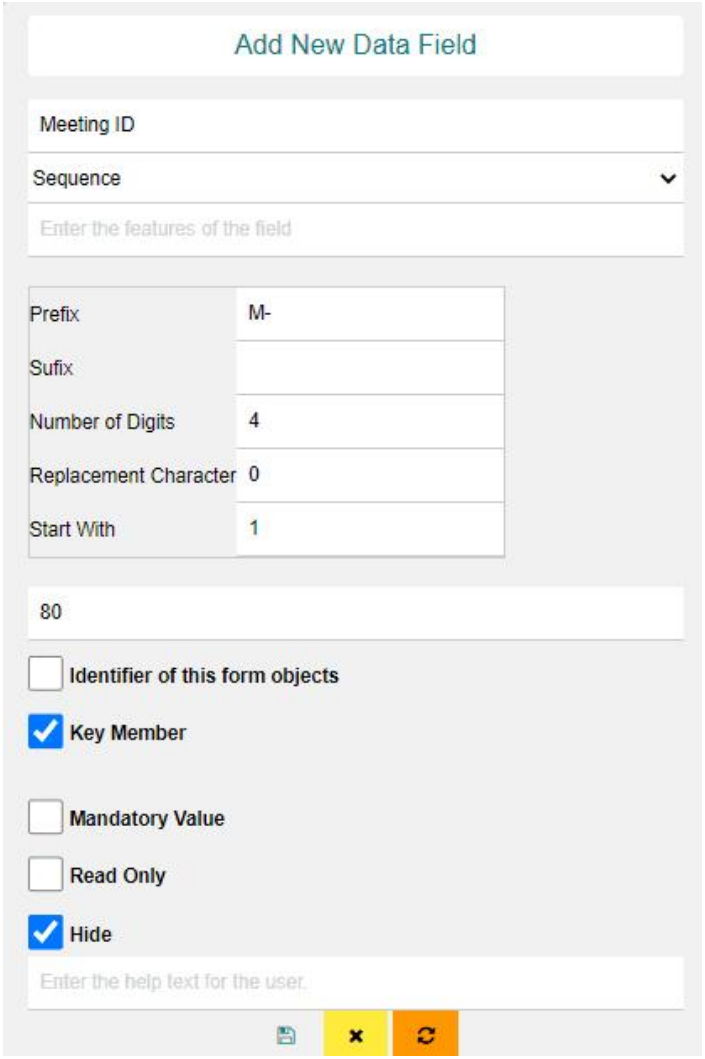
8. Create Time as the next data field. So select ‘text’ as the data type.



9. Enter the ‘Host’ as the next data field in this form.
After that select the data type. Host is a user of this application. So select the ‘options’ as data type. Before that you must have entered users in another form.

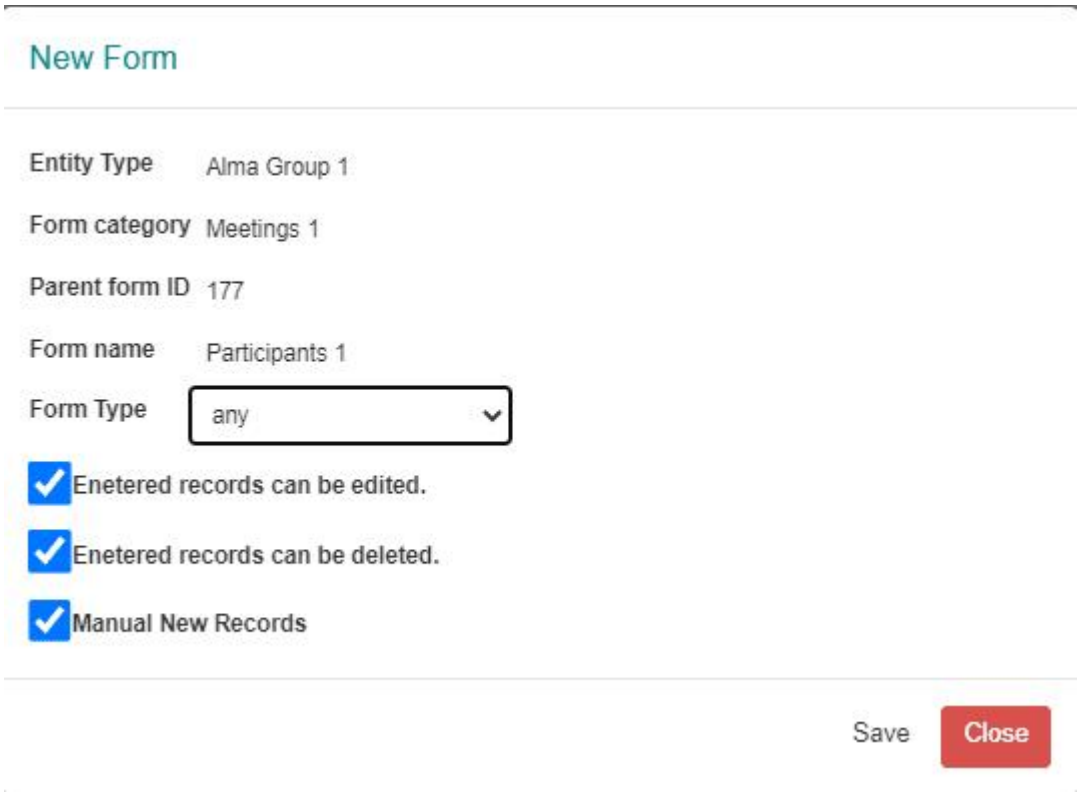


- In this ‘Meetings form’ we create a Meeting ID field for preventing data duplication. We use ‘Sequence’ as the data type and it generate codes automatically. Also we do not use this data field for any purpose. So we can hide it.

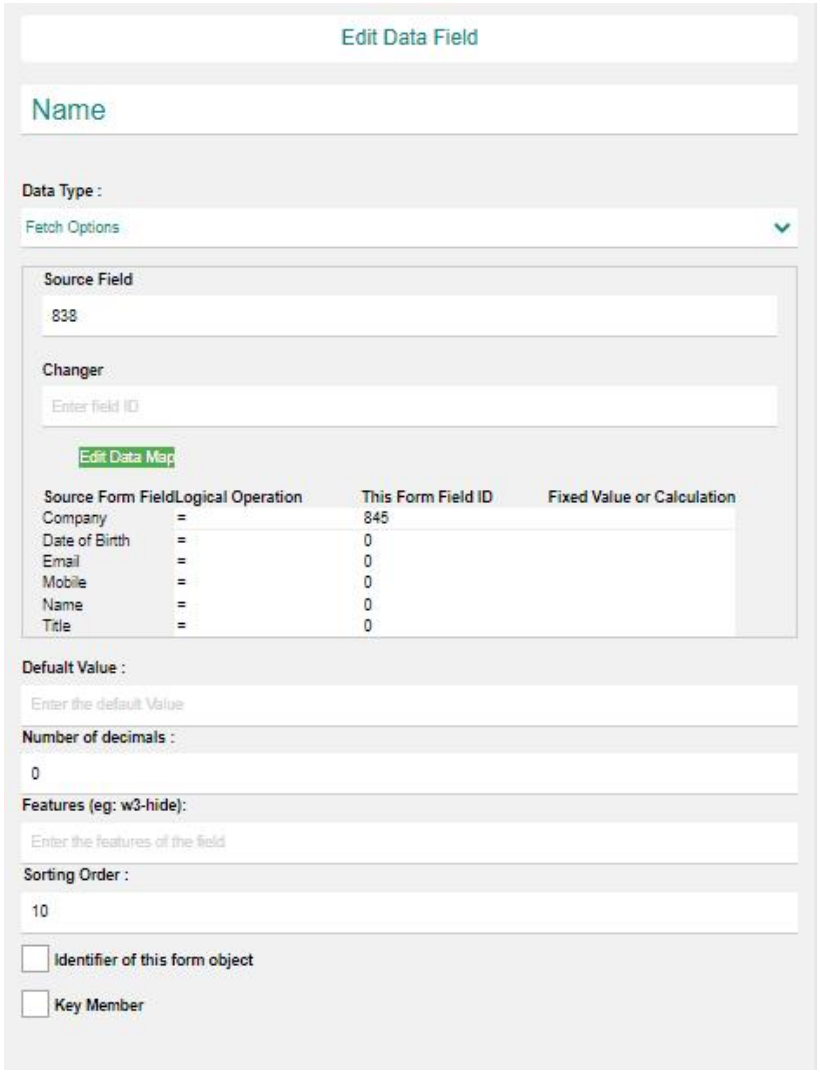


- **We have now successfully developed our ‘Meetings’ form. Next, let’s create the sub-form called ‘Participants’ to enter the relevant participants of the meeting.**

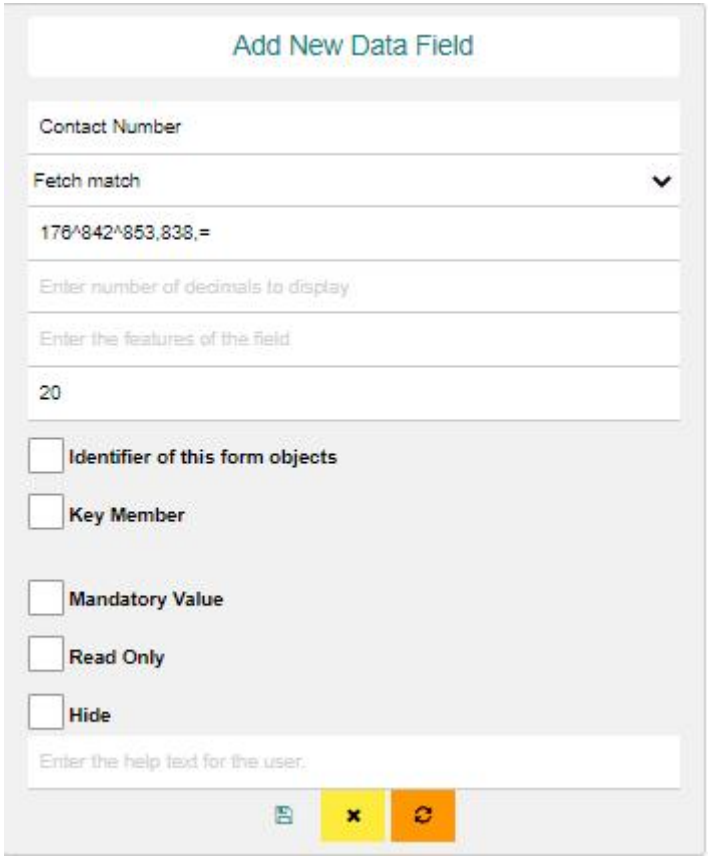
1. Create a sub-form called ‘Participants’.



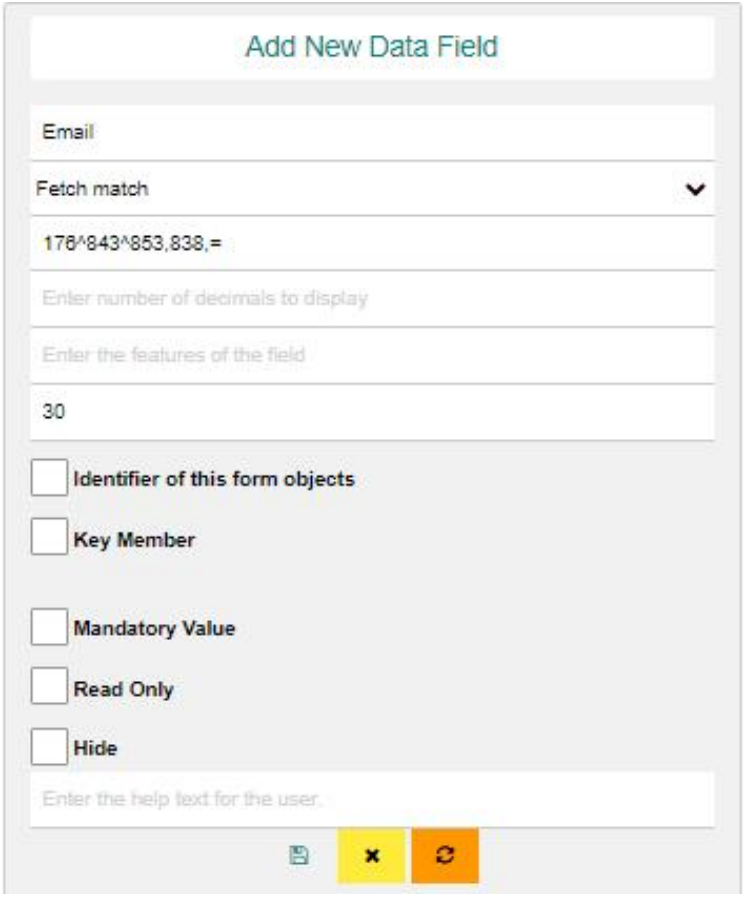
2. Enter the ‘Name’ as first field in this sub-form. You can select the contact name from the contact form. Select the ‘Fetch options’ as the data type. (Read the help in editor role for more information.)



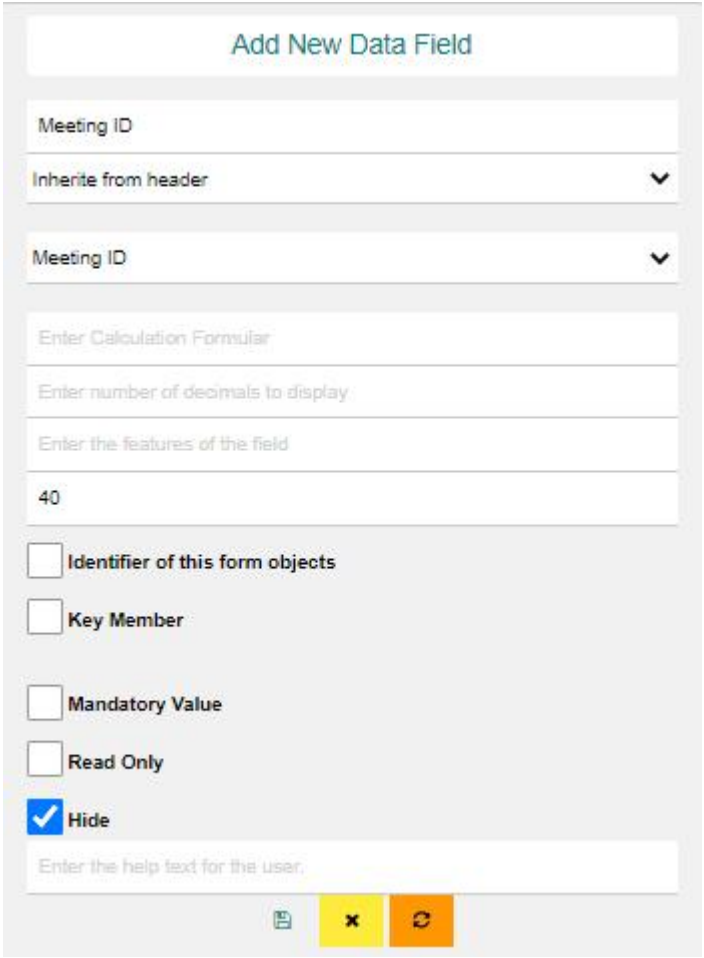
3. Enter the ‘Contact Number’ as next field in this sub-form. Select ‘ Fetch match’ as the data type. When you select the contact name from the contact form system automatically generate the contact number. (Read the help in editor role for more information.)



4. Enter the ‘Email’ as next field in this sub-form. Select ‘ Fetch match’ as the data type. When you select the contact name from the contact form system automatically generate the Email. (Read the help in editor role for more information.)



- When you create a sub-form, you should inherit the key fields to sub-form from its main form. Use the inherit from header data type.



LEVEL 02

What We Have Planned For This Level ……

- In this level, our primary focus is on the creation of invoices for sales transactions. This process is vital as it allows us to maintain a comprehensive record of bill information for future reference whenever we sell goods to our valued customers.

- Within the Invoice Header form, our initial step is to input customer details and total amounts. Following this, we proceed to establish a sub-form dedicated to Invoice Lines, which serves as a convenient platform for entering the details of sales products. This sub-form is particularly useful when dealing with multiple product lines, streamlining our data entry process.

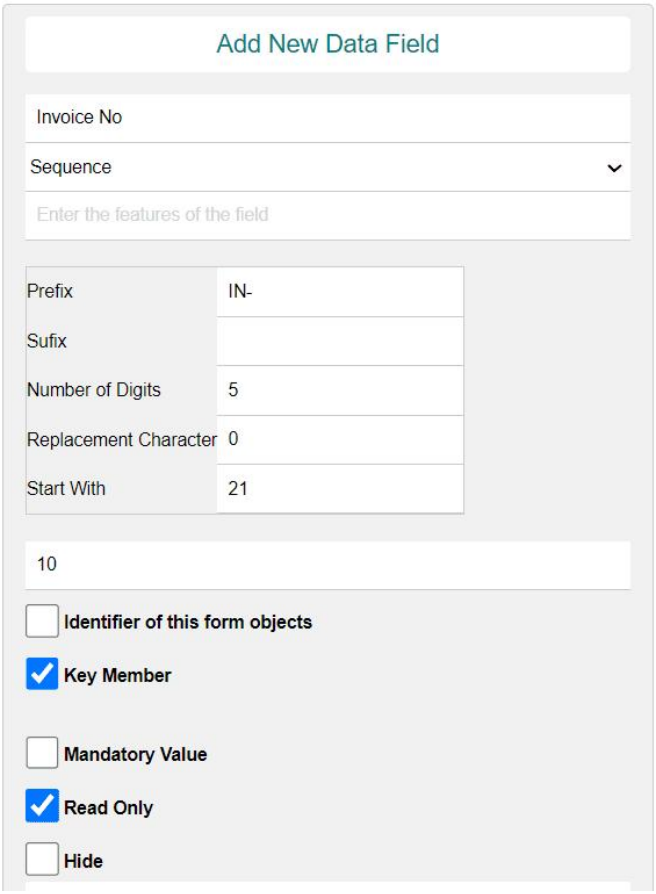
- Additionally, we are in the process of crafting a dedicated page designed to provide a comprehensive overview of today's sales activity. This page is constructed by incorporating the previously created invoice details, utilizing data fields from the INVOICE and INVOICE LINES forms. We have aptly named this page the 'Daily Sales Report.' Within this report, we present the total sales figures for the current day. Furthermore, the page features an aesthetically pleasing bar chart depicting sales by product categories.

Exercise 01

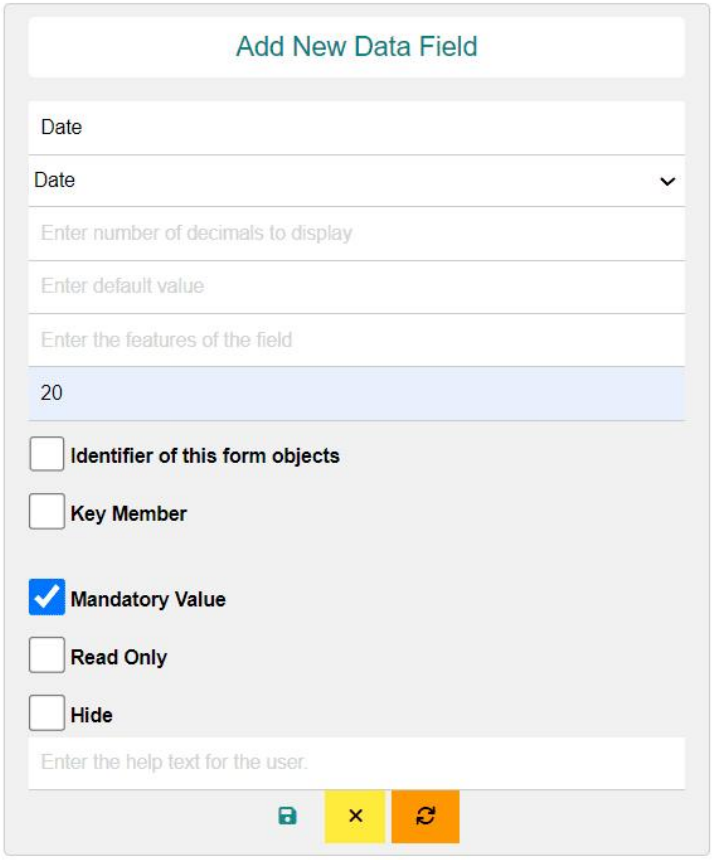
1. how to create a general invoice using this software tool?

- Begin by creating an Entity level and entities. Then, proceed to establish a form category and add a new form named 'Invoice' within it. In this header form, include fields for the invoice number, customer details, and the total sum of the product amounts.

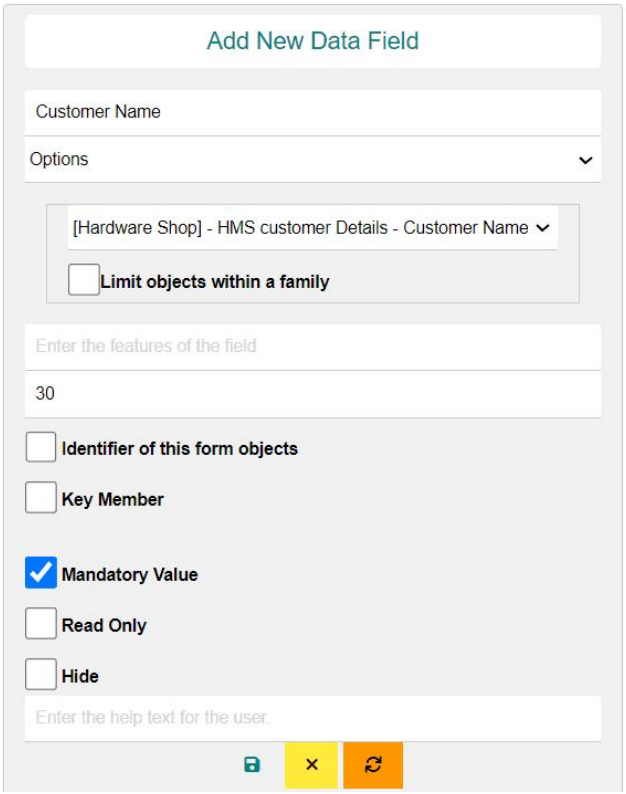
- Start by adding a data field for the 'Invoice No,' utilizing the sequence data type.



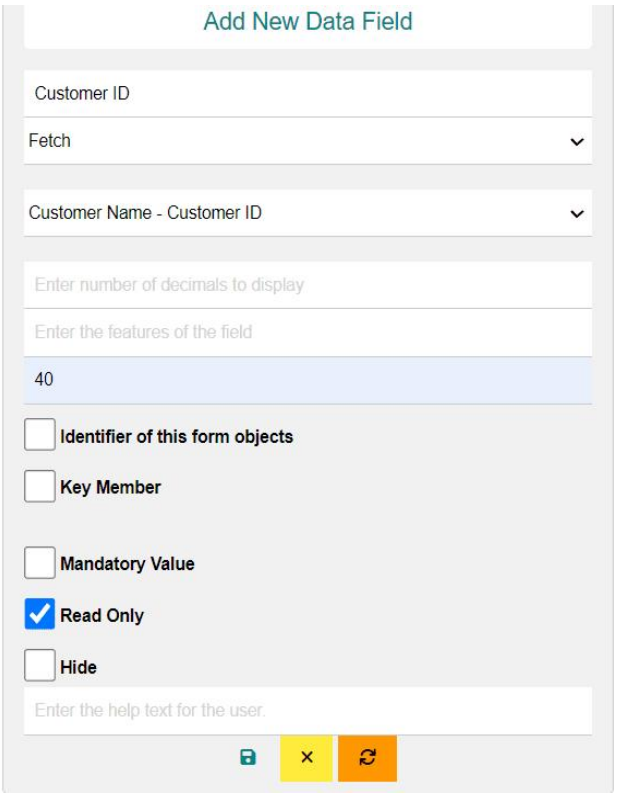
- Next, include a data field for the 'Invoice Date.'



- Introduce the 'Customer Name' as the subsequent data field. If the customer is already registered, you can retrieve their name from the customer database and associate it with this invoice. This field should be of the option data type. However, if the customer isn't registered yet, allow the user to manually input the customer name.

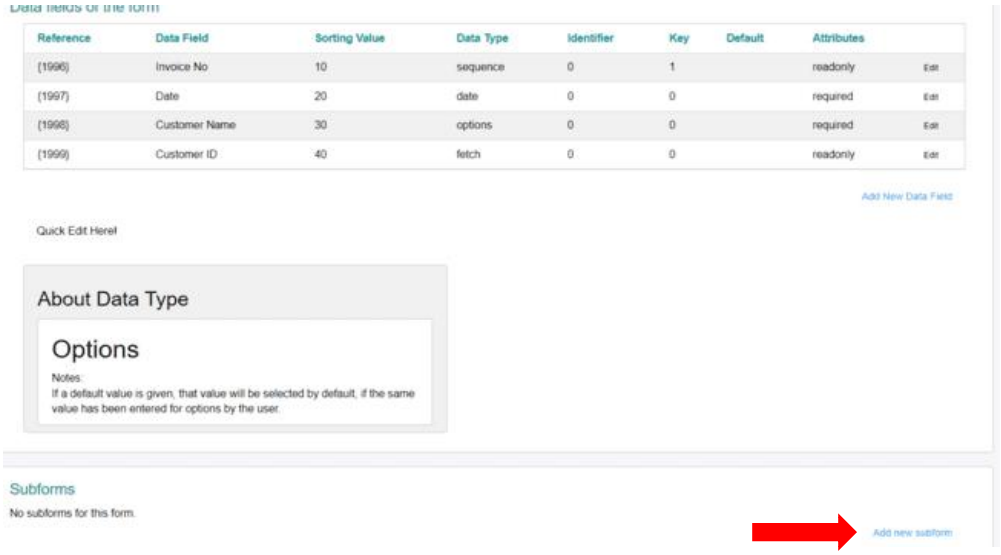


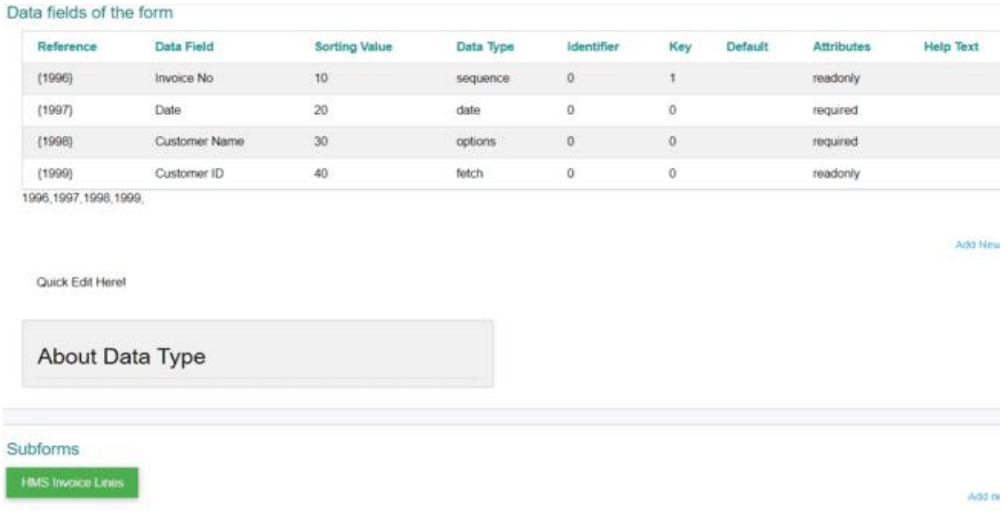
- The following field is the 'Customer ID,' which will be fetched from the customer registry.



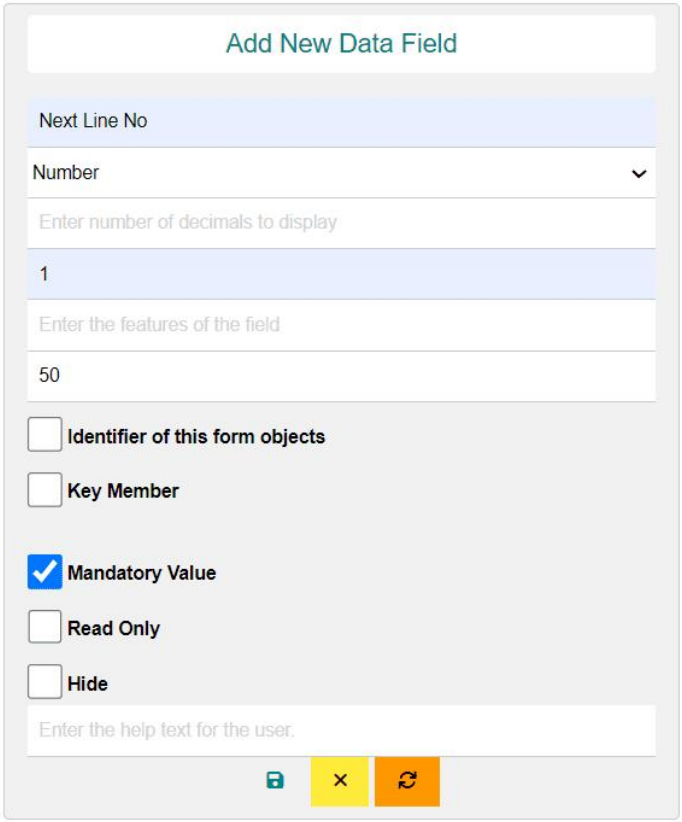
- Now, let's create a sub-form for the invoice lines. Within this sub-form, you'll enter product details, quantities, prices, and other relevant information, especially for invoices with multiple items.

- Initially, create a sub-form within the invoice header form and name it 'Invoice Lines.'



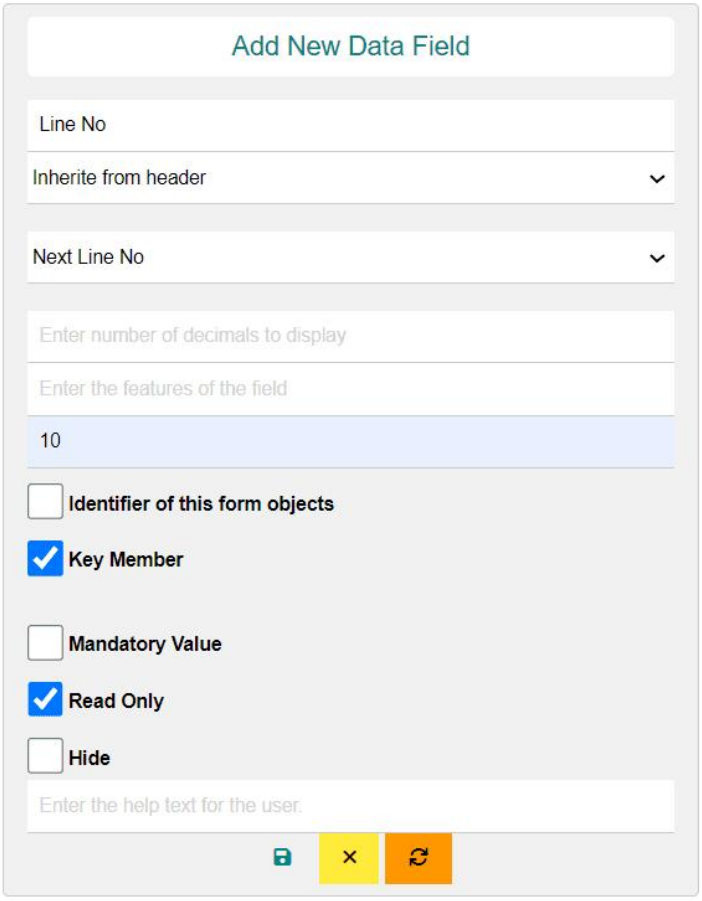


- Inside the sub-form, the first data field should be the 'Line Number.' To generate this line number, first add a data field called 'Next Line Number' to the header form. Ensure its default value is set to '1.'

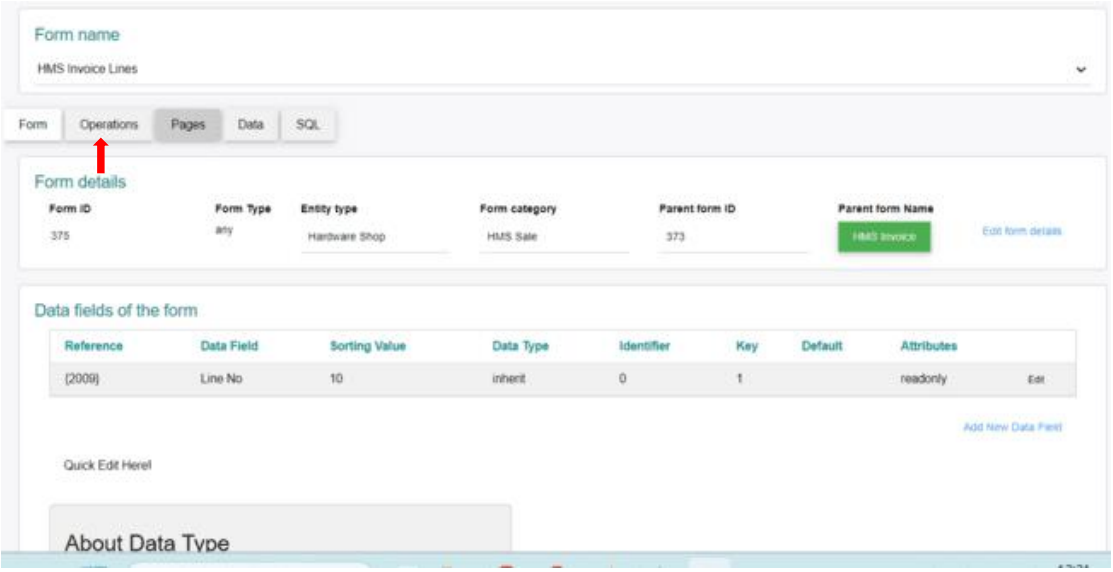


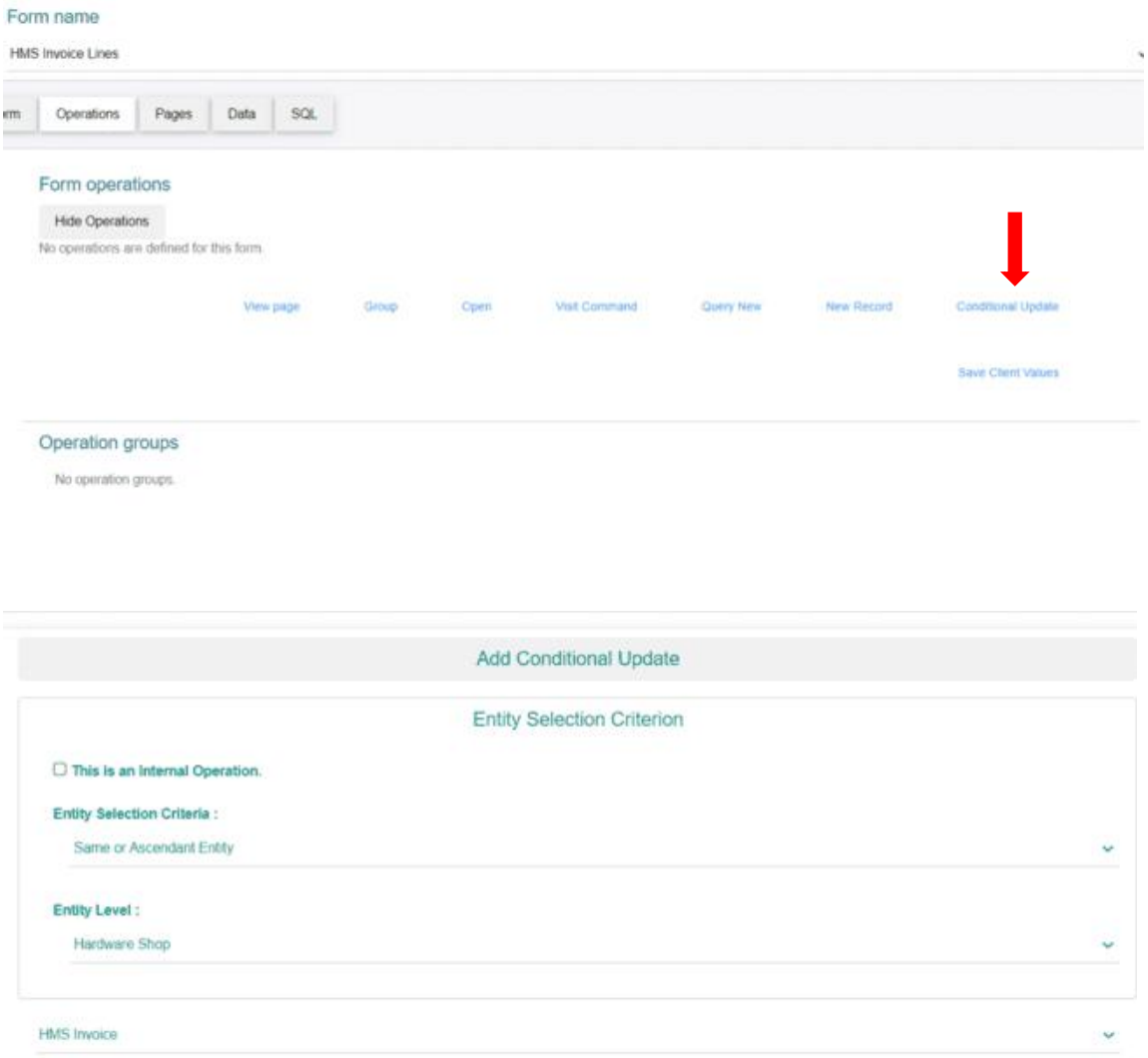
- The next data field in the sub-form should be the 'Line Number'. This data type should inherit from the header.

[We utilize this operation to automatically generate the 'Line No' within the subform. This operation is executed from the subform but affects the 'Next Line No' in the header form. Initially, we set the 'Next Line No' to the default value of '1'. Afterward, we configure the 'add operation' to have a fixed value of '1'. When the operation is active, and we input a new line in the subform, it automatically adds '1' to the 'Next Line No' in the header form, resulting in '2' (1 + 1 = 2). Subsequently, this 'Line No' value is inherited from the header to the subform's 'Line No']

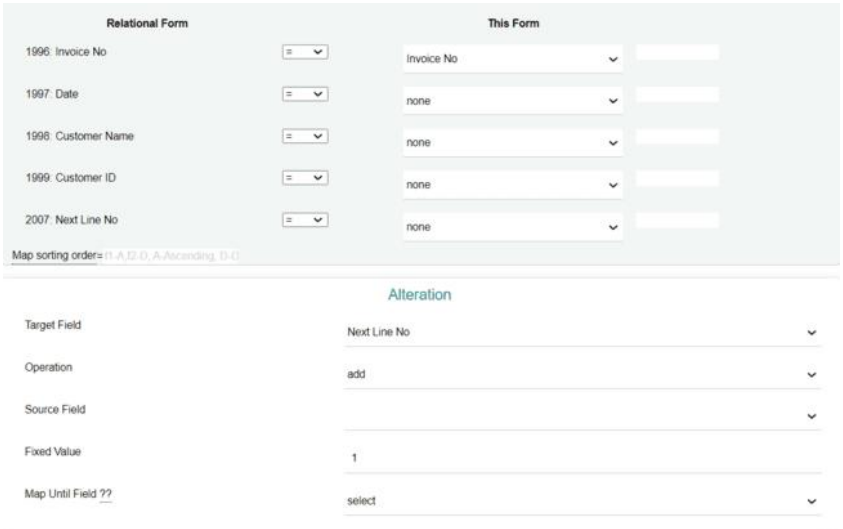


- Proceed to the 'Operation' tab and select 'Conditional Update.'

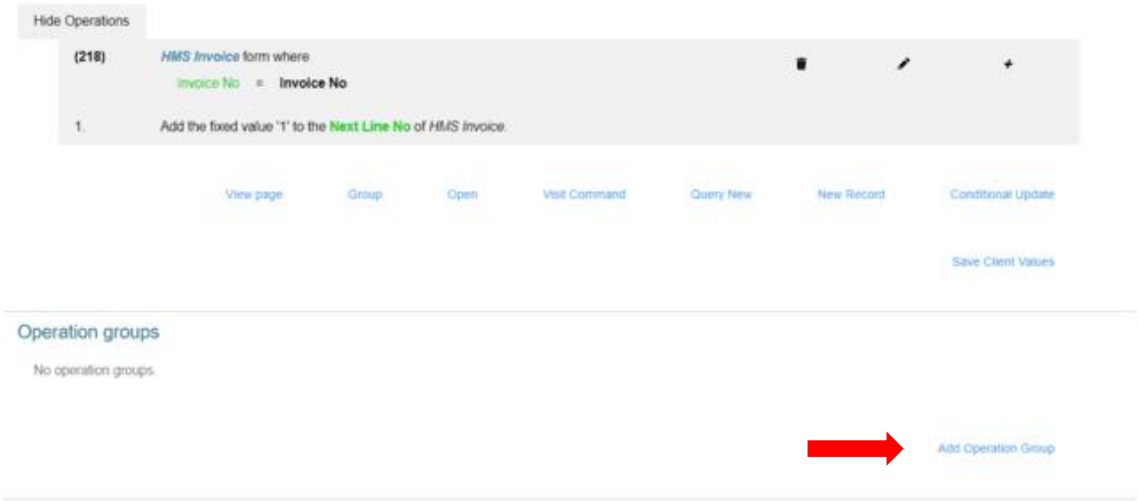


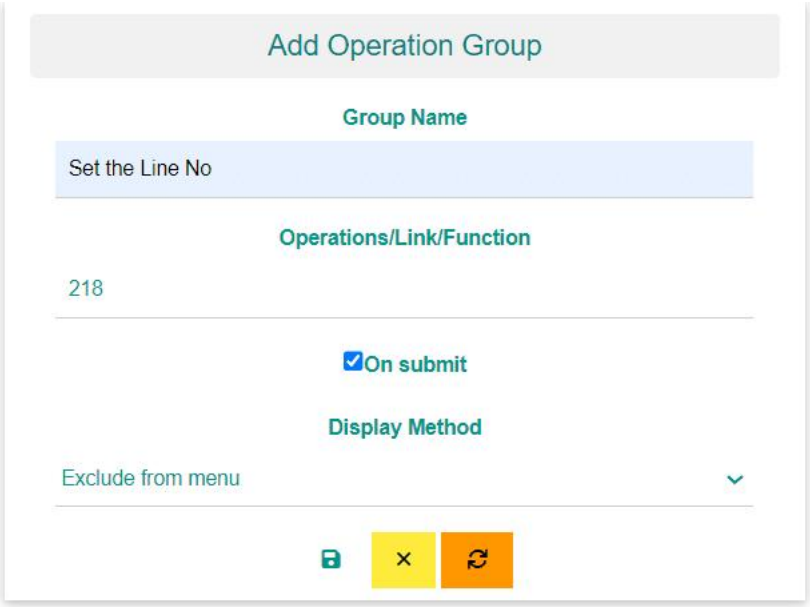


- For the operation target field, choose the 'Next Line Number' in the invoice header form.

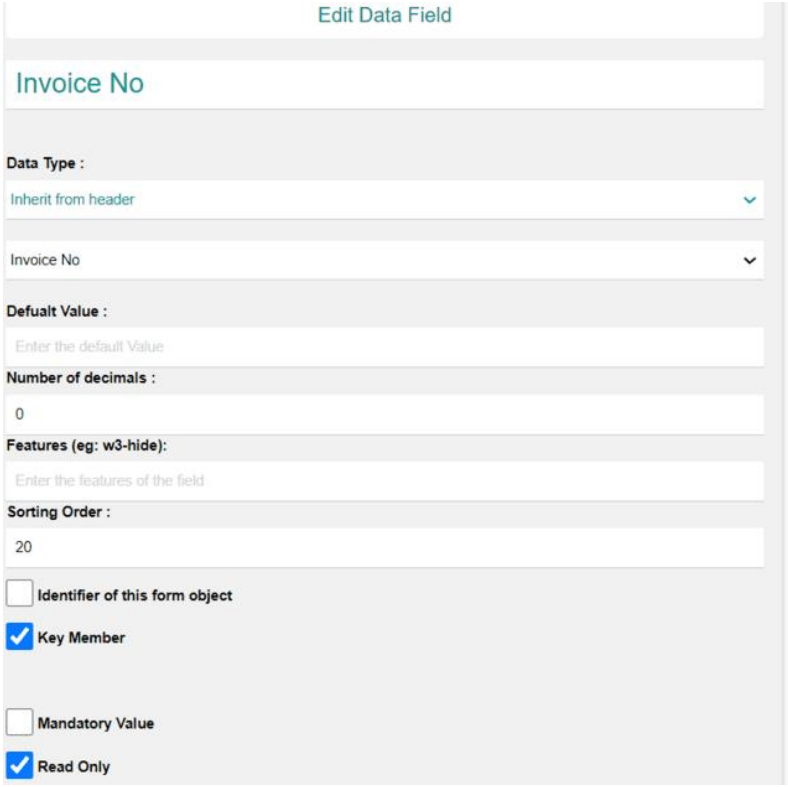


- Next, we need to execute an operation, and for that, we should create a group. Let's see how to do it.

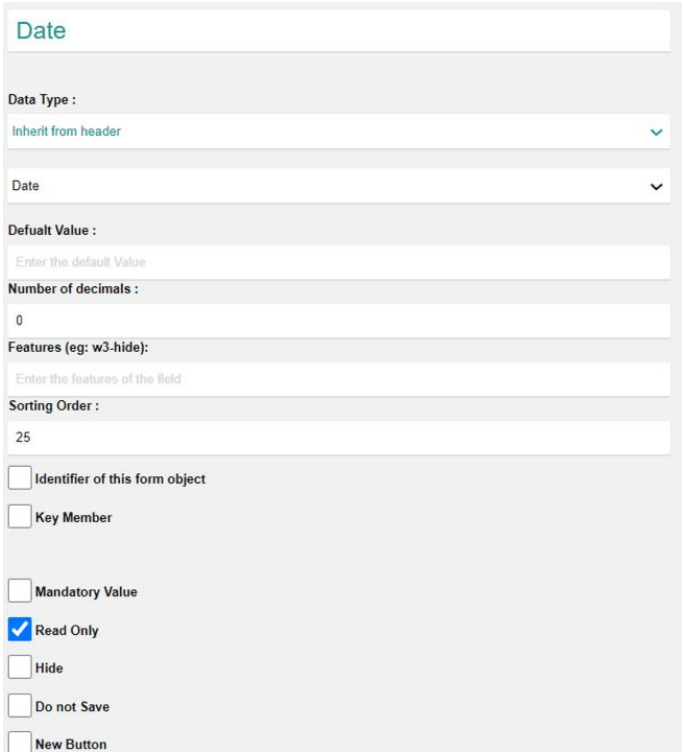




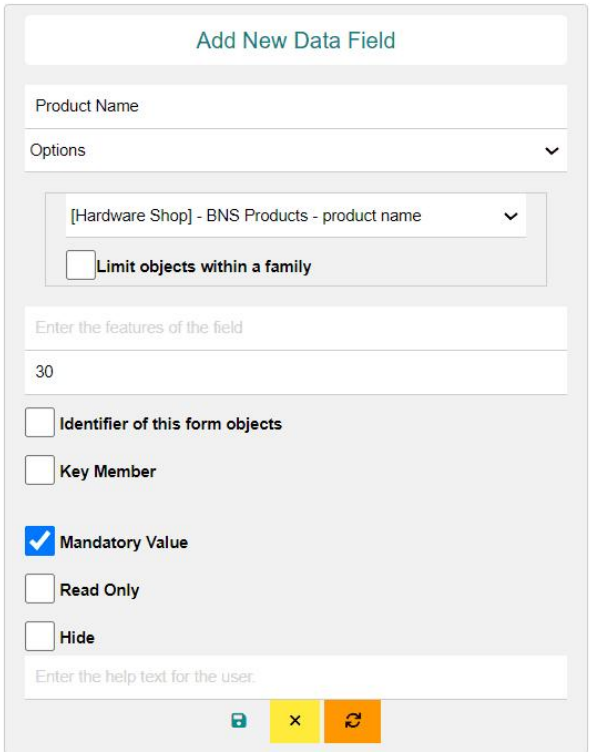
- This operation will automatically adjust the line numbers.
- The second data field in the sub-form should be the 'Invoice Number,' inheriting its data type from the header form.



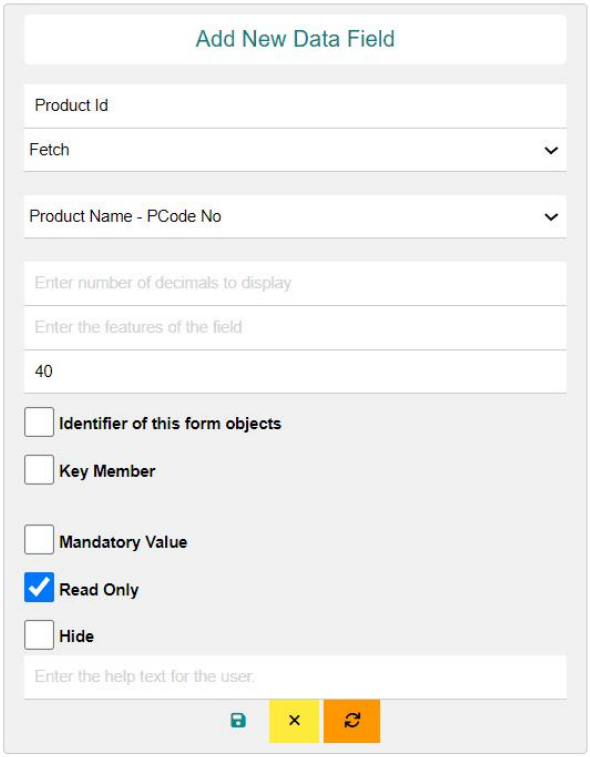
- Next data field is date.



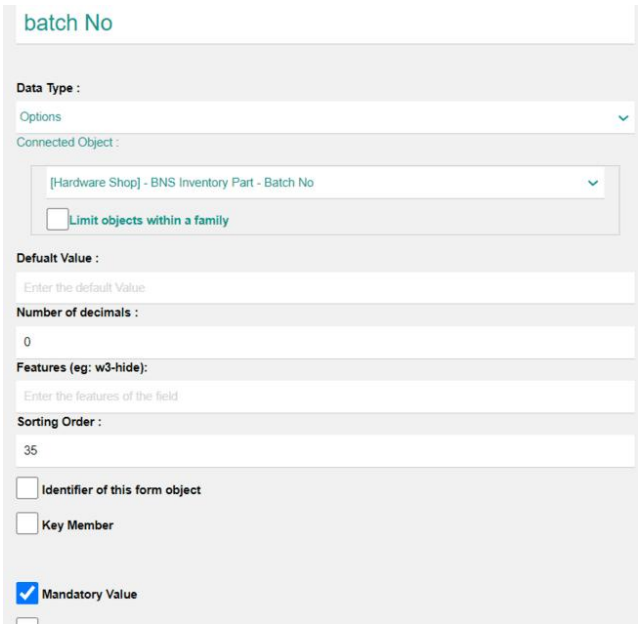
- Proceed by entering the "Product Name" data field Product Name for the 'Options' data type. It's important to have previously added product details in another form.



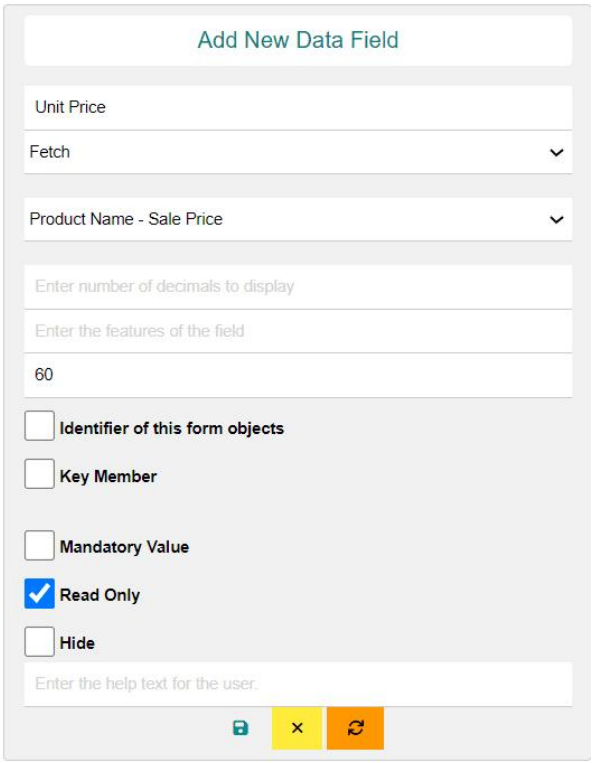
- Moving forward, you'll need to fetch the "Product ID" from the Product form.



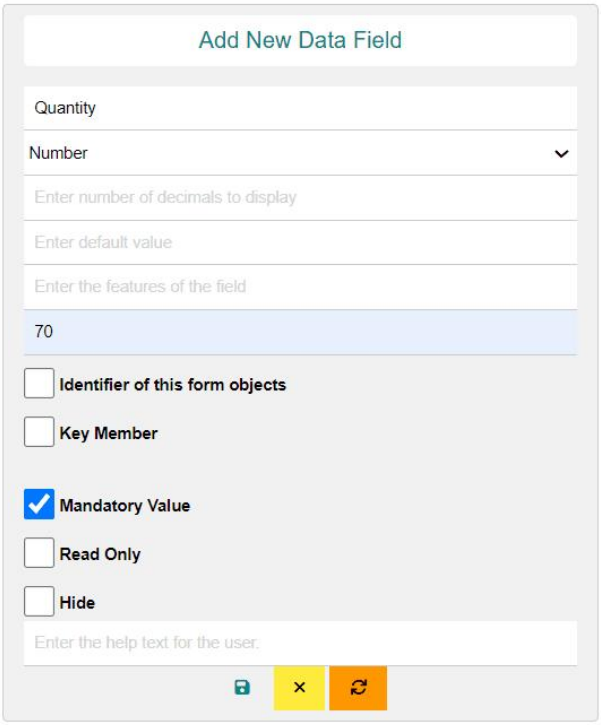
- Next in line is creating the "Batch Number" data field. Set it to utilize the 'Option' data type.



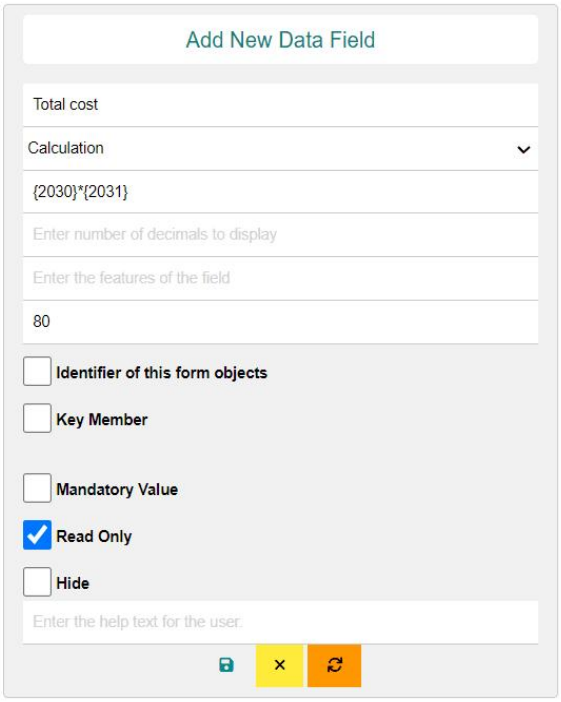
- Add the "Unit Price" as the subsequent data field. You should fetch its data type from the product details form.



- Enter the "Quantity" as the following data field. Make sure to set it with the 'Number' data type.

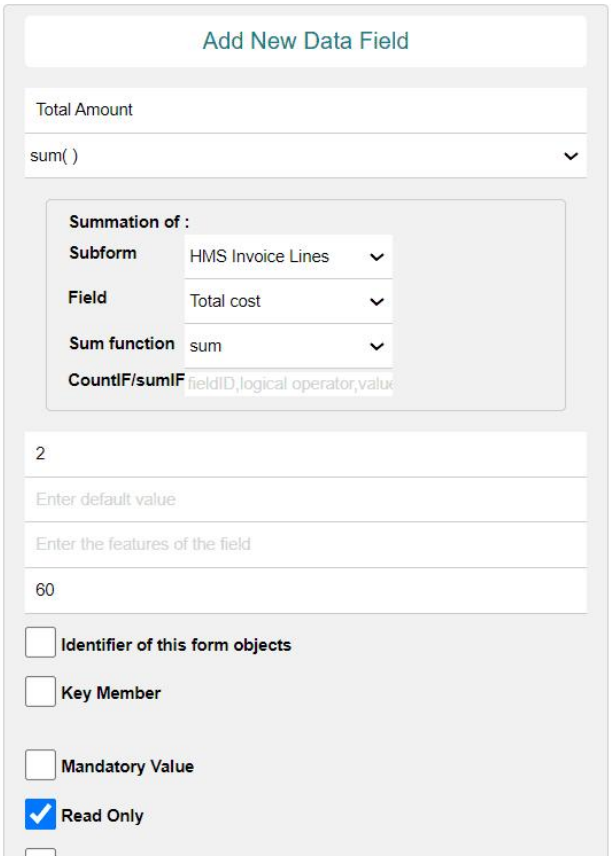


- Lastly, introduce the "Total Cost" data field. For this, select the 'Calculation' data type.

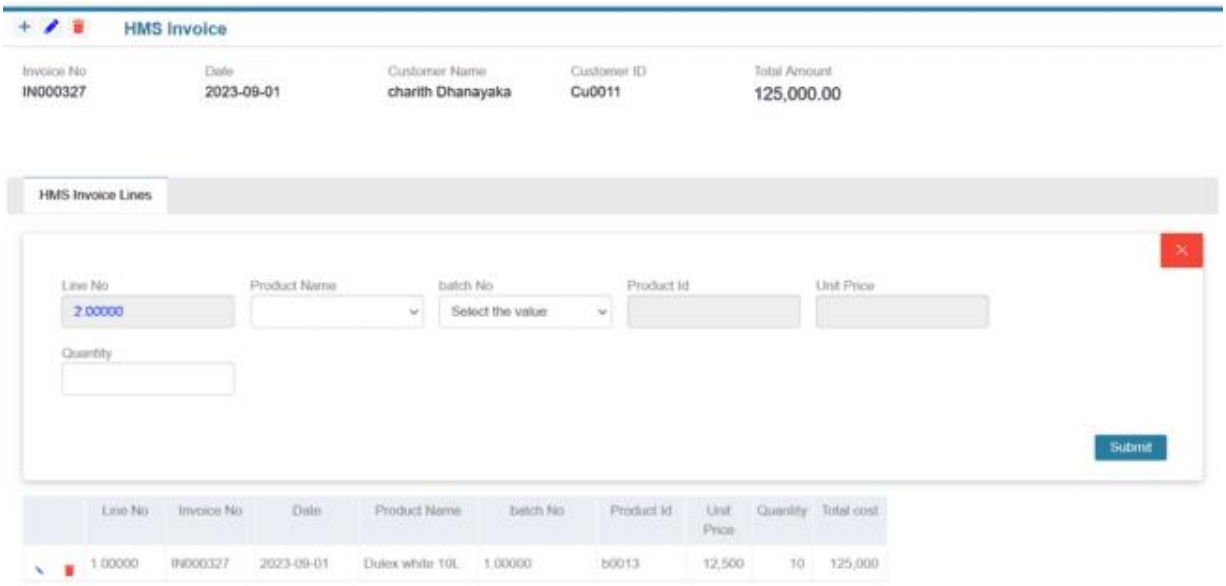


- Next, we need to calculate the total cost by summing up all the expenses. To achieve this, we should include a new data field in the header form labeled as the 'Total Amount.' This field will have the data type set to 'Sum.'

- Within this sum data type, we should incorporate a sub-form that calculates the sum of the desired fields.

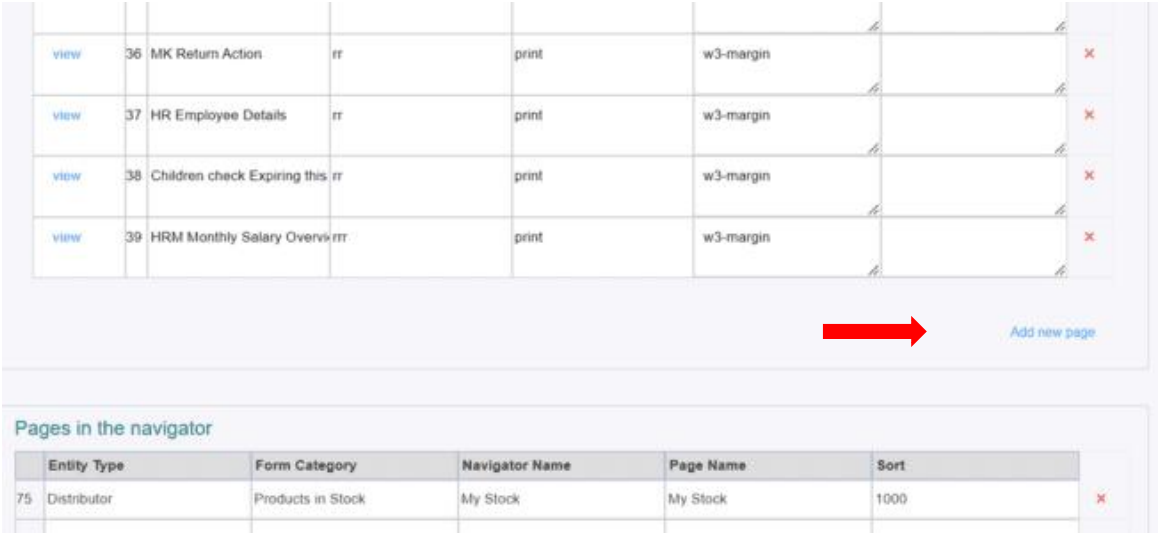


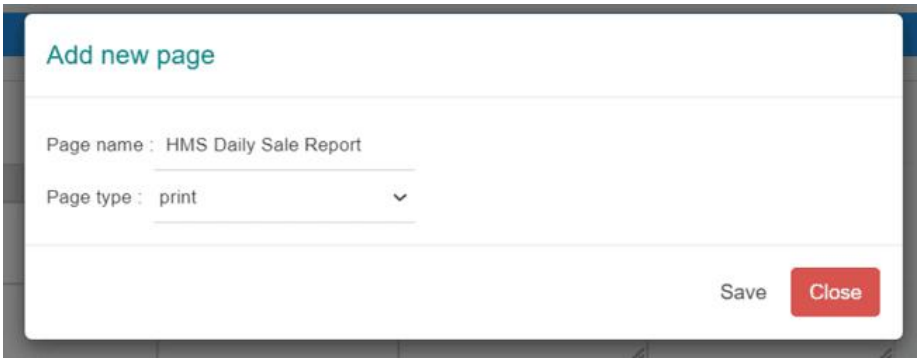
This is the user interface form we have created.



2. How to create a daily sales report and chart ?
   For this, we need to create a new page.

- First, go to the 'Page' tab, select 'Add New Page,' give it a name, and save it.





- In the 'Layout' column, specify the number of rows for the page. For example, 'rr = 2 Rows.'

- Next, click on 'View' for the page you just created.

- Click on the first plus mark, which represents the first row of the page. Here, you can define the header for the page. Enter the header content in the 'Content' column, and you can customize the header's color in the 'Classes' column.

- Click on the second plus mark, representing the second row of the page. This is where you specify what information should be displayed on the page. You can choose the content type, such as a table, count, chart, etc.

- In the third row, you can calculate the sum of sales. Set the content type as 'sum' and enter 'total amount' in the response column.

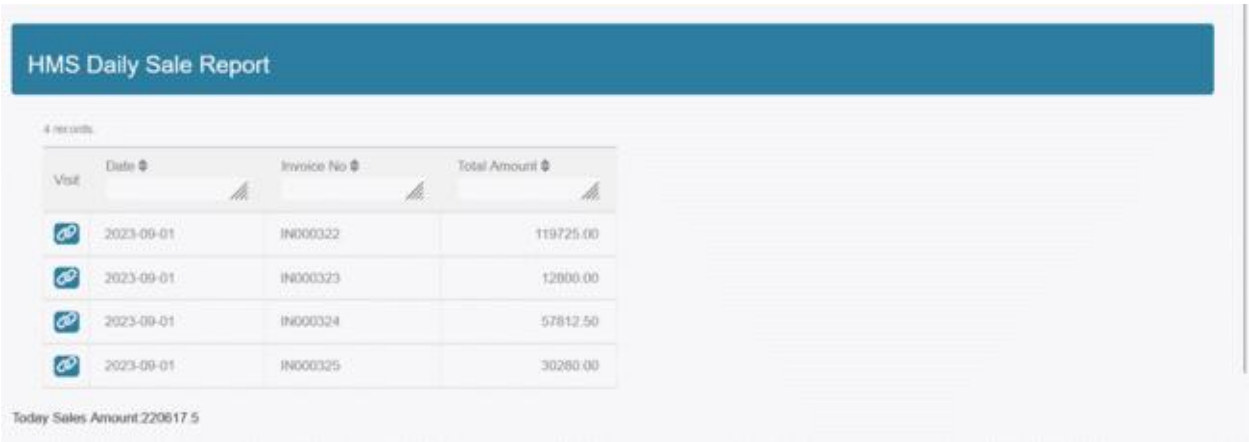
- In the last row, you can create a chart to visualize sales by product. In the content type column, select 'chart.' Add the header as 'Sales by Product' and specify the response column to include product and quantity data.

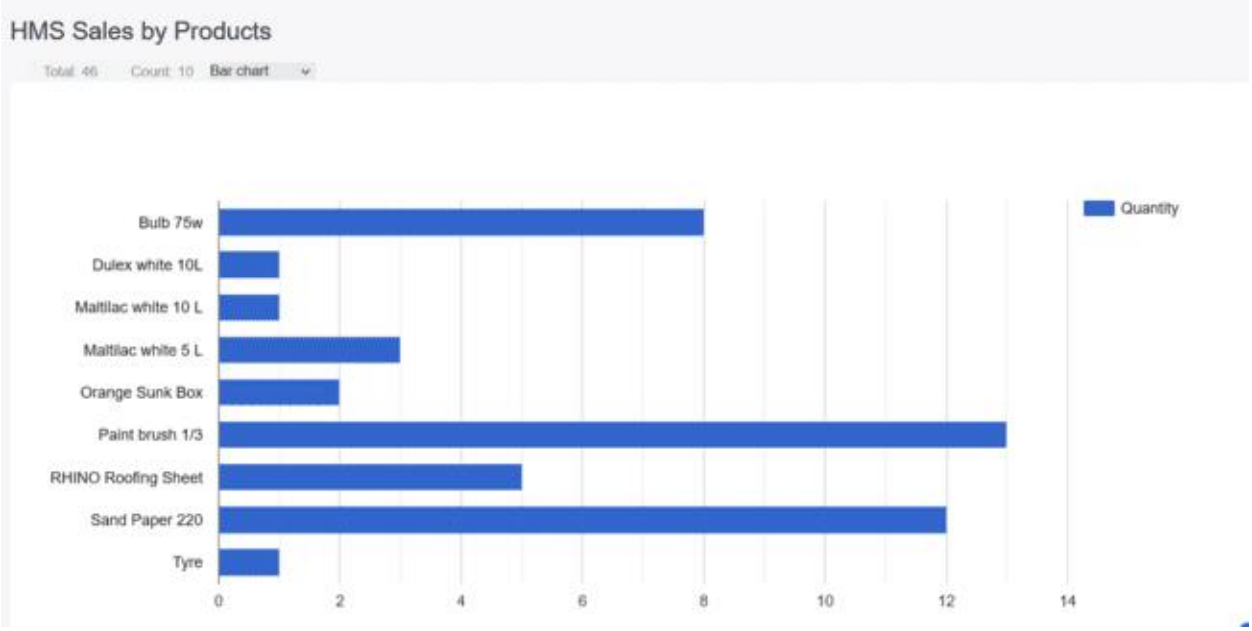
- Remember, we want a daily sales report, so we need to filter the data to show only today's sales. Enter the filter condition in the filter column as 'date {field id} = @today.' Now, you'll be able to view only today's sales. If we don't enter a filter condition, all records we input will be displayed in this form.

- Now, let's learn how to save this page in the navigator. Click on the plus mark below ‘page in navigator’.

- Enter the entity type, form category, navigator name, and page name. Finally, save it.

This represents the ultimate appearance of the report in the user role.





Learning Outcome…….

Data Types

Data Types	*Form*	*Data Field*	*Value from*
Sequence	Invoice	Invoice No	
Option	Invoice	Customer Name	Identifier From customer Registered form Customer Name
	Invoice Lines	Product Name	Identifier from Products form Product Name
Fetch	Invoice	Customer ID	Fetch From customer Registered form Customer ID
	Invoice Lines	Product ID	Fetch from Product form product ID
	Invoice Lines	Unit Price	Fetch from Product form Unit Price
Sum	invoice	Total Amount	Invoice lines form Total Cost
Calculation	Invoice Lines	Total Cost	{price}*{Qty}

Operations

Operation	Operation name	*What should Operation do*
add	Set the Line No	Automatically generated (Invoice Lines) sub form's Line Number.

Page Setting

Creating an overview table and incorporating a chart

Level 03

What We Have Planned For This Level ……

- In this level, we will focus on inventory management during the processes of purchasing and sales. Specifically, we will be creating GRN and GRN Lines, as we did with invoices and invoice lines in the previous level.

- Next, we will establish operations related to GRN Lines, which involve adding newly purchased quantities to our inventory. Additionally, we will create operations for Invoice Lines to deduct sold quantities from the inventory.

- Furthermore, we will implement a systematic approach by creating an operation for each transaction. These operations will be streamlined into a unified inventory transaction form for execution.

- Another essential exercise we will undertake is automatic reservation. To achieve this, we will design a customer order form and order lines form. Once the reservation order is generated, it will prioritize batch numbers in ascending order, and primary and secondary operations will automatically reserve the required inventory. Upon order delivery, the reserved quantities will be deducted from the inventory.

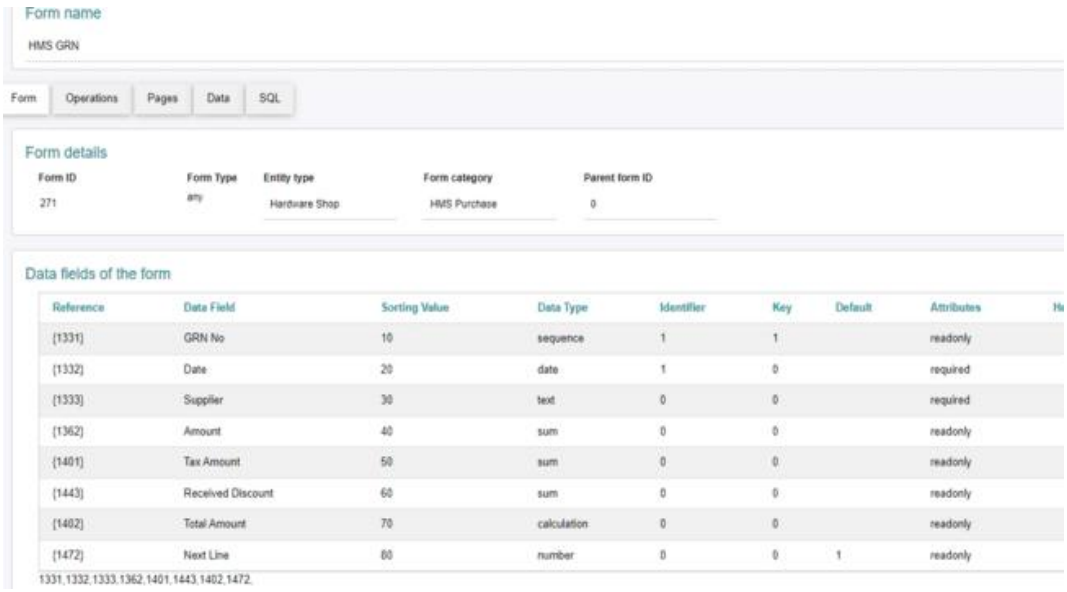
- Next, we will address manual reservation, where users will manually specify batch numbers and the quantity to be reserved. To check the availability of inventory, an operation for visiting the inventory will be executed. As with automatic reservation, when an order is delivered, the corresponding quantity will be deducted from the inventory.

- Lastly, we will create the bill header by entering the contractor details, such as ID, name, and hourly rate. Then, we will apply a filter condition to ensure that only active contractors are included. After the bill header is set up, we will retrieve the relevant data (such as worked time and project details) for the contractor from the approved time records for the specified month. This data will be mapped to the bill line to complete the monthly bill creation. Finally, by following these steps, you will be able to generate the contractor's monthly bill.

Exercise 01

1. How can I set up automatic inventory updates when making purchases and sales?

- Begin by creating a Goods Received Note (GRN) form along with GRN lines, similar to how you would create invoices and invoice lines.
- These are the forms we created for GRN.



Form name

HMS GRN

Form | Operations | Pages | Data | SQL

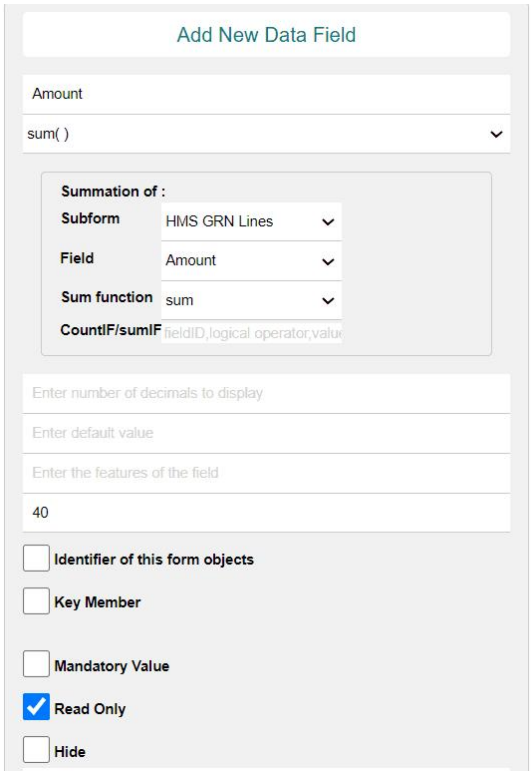
Form details

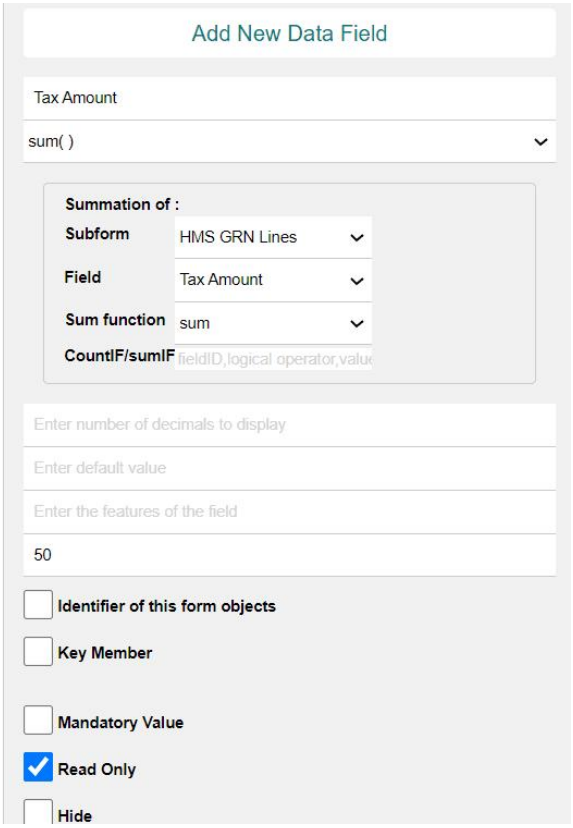
Form ID	Form Type	Entity type	Form category	Parent form ID
271	any	Hardware Shop	HMS Purchase	0

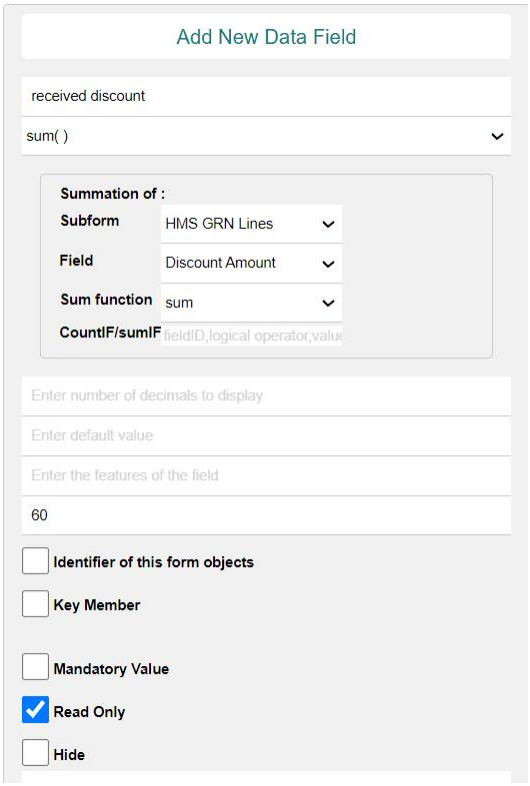
Data fields of the form

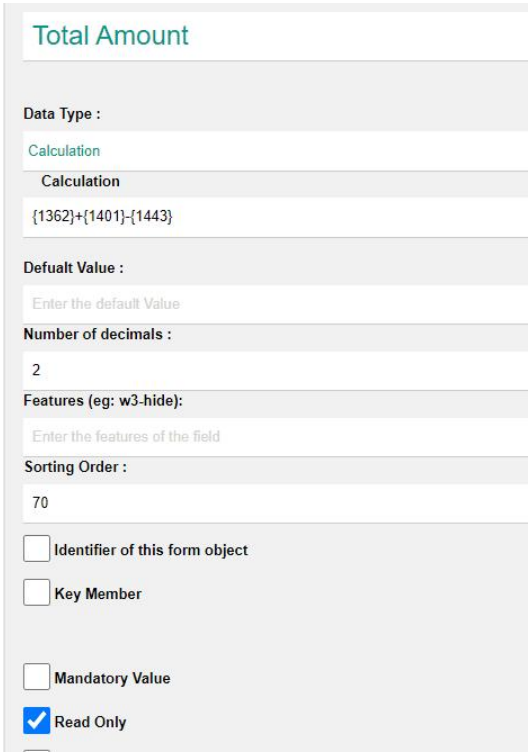
Reference	Data Field	Sorting Value	Data Type	Identifier	Key	Default	Attributes
{1331}	GRN No	10	sequence	1	1		readonly
{1332}	Date	20	date	1	0		required
{1333}	Supplier	30	text	0	0		required
{1362}	Amount	40	sum	0	0		readonly
{1401}	Tax Amount	50	sum	0	0		readonly
{1443}	Received Discount	60	sum	0	0		readonly
{1402}	Total Amount	70	calculation	0	0		readonly
{1472}	Next Line	80	number	0	0	1	readonly

1331,1332,1333,1362,1401,1443,1402,1472,


Add New Data Field
Amount
sum( )
Summation of :
Subform
HMS GRN Lines
Field
Amount
Sum function
sum
CountIF/sumIF
fieldID,logical operator,valu
Enter number of decimals to display
Enter default value
Enter the features of the field
40
Identifier of this form objects
Key Member
Mandatory Value
Read Only
Hide


Add New Data Field
Tax Amount
sum( )
Summation of :
Subform
HMS GRN Lines
Field
Tax Amount
Sum function
sum
CountIF/sumIF
fieldID,logical operator,valu
Enter number of decimals to display
Enter default value
Enter the features of the field
50
Identifier of this form objects
Key Member
Mandatory Value
Read Only
Hide


Add New Data Field
received discount
sum( )
Summation of :
Subform
HMS GRN Lines
Field
Discount Amount
Sum function
sum
CountIF/sumIF
fieldID,logical operator,value
Enter number of decimals to display
Enter default value
Enter the features of the field
60
Identifier of this form objects
Key Member
Mandatory Value
Read Only
Hide


Total Amount
Data Type :
Calculation
Calculation
{1362}+{1401}-{1443}
Defualt Value :
Enter the default Value
Number of decimals :
2
Features (eg: w3-hide):
Enter the features of the field
Sorting Order :
70
Identifier of this form object
Key Member
Mandatory Value
Read Only

Data fields of the form

Reference	Data Field	Sorting Value	Data Type	Identifier	Key	Default	Attributes
[1429]	GRN No	10	inherit	0	1		readonly
[1430]	Date	20	inherit	0	0		readonly
[1431]	Supplier	30	inherit	0	0		readonly
[1470]	Batch No	35	number	0	1		required
[1595]	Line No	40	inherit	0	1		required
[1433]	Product Name	50	options	0	0		required
[1434]	Pcode No	60	fetch_match	0	0		readonly
[1435]	Unit Price	70	fetch	0	0		readonly
[1437]	PQuantity	75	number	0	0		required
[1438]	Amount	80	calculation	0	0		readonly
[1436]	Tax Rate	90	number	0	0	15	required
[1439]	Tax Amount	100	calculation	0	0		readonly
[1440]	Discount Rate	110	number	0	0		
[1441]	Discount Amount	120	calculation	0	0		readonly
[1442]	Sub Total	130	calculation	0	0		readonly

1429,1430,1431,1470,1595,1433,1434,1435,1437,1438,1436,1439,1440,1441,1442;

Product Name

Data Type :

Options

Connected Object :

[Hardware Shop] - BNS Products - product name

☐ **Limit objects within a family**

Defualt Value :

Enter the default Value

Number of decimals :

0

Features (eg: w3-hide):

Enter the features of the field

Sorting Order :

50

☐ **Identifier of this form object**

☐ **Key Member**

☑ **Mandatory Value**

Pcode No

Data Type :

Fetch match

Calculation

267^1328^1409,1433,=

Defualt Value :

Enter the default Value

Number of decimals :

0

Features (eg: w3-hide):

Enter the features of the field

Sorting Order :

60

- [ ] Identifier of this form object
- [ ] Key Member
- [ ] Mandatory Value
- [x] Read Only

Unit Price

Data Type :

Fetch

Product Name - Price

Defualt Value :

Enter the default Value

Number of decimals :

2

Features (eg: w3-hide):

Enter the features of the field

Sorting Order :

70

- [ ] Identifier of this form object
- [ ] Key Member
- [ ] Mandatory Value
- [x] Read Only

Amount

Data Type :

Calculation

Calculation

{1435}*{1437}

Defualt Value :

Enter the default Value

Number of decimals :

2

Features (eg: w3-hide):

Enter the features of the field

Sorting Order :

80

- [ ] Identifier of this form object
- [ ] Key Member
- [ ] Mandatory Value
- [x] Read Only

Tax Rate

Data Type :

Number

Defualt Value :

15

Number of decimals :

0

Features (eg: w3-hide):

Enter the features of the field

Sorting Order :

90

- [ ] Identifier of this form object
- [ ] Key Member
- [x] Mandatory Value

Tax Amount

Data Type :

Calculation

Calculation

{1438}*{1436}/100

Defualt Value :

Enter the default Value

Number of decimals :

2

Features (eg: w3-hide):

Enter the features of the field

Sorting Order :

100

- [ ] **Identifier of this form object**
- [ ] **Key Member**
- [ ] **Mandatory Value**
- [x] **Read Only**

Sub Total

Data Type :

Calculation

Calculation

{1438}+{1439}-{1441}

Defualt Value :

Enter the default Value

Number of decimals :

2

Features (eg: w3-hide):

Enter the features of the field

Sorting Order :

130

- [ ] **Identifier of this form object**
- [ ] **Key Member**
- [ ] **Mandatory Value**
- [x] **Read Only**

- Next, create an inventory part form dedicated to managing your stock effectively.
- Determine the necessary data fields and data types to include in this form.
- Start by creating a form category labeled as 'Inventory' and name the new form as 'Inventory Part.'

- Include the following fields in your 'Inventory Part' form: Batch Number, Product Name, Product ID, Expiry Date, and Quantity in Stock.

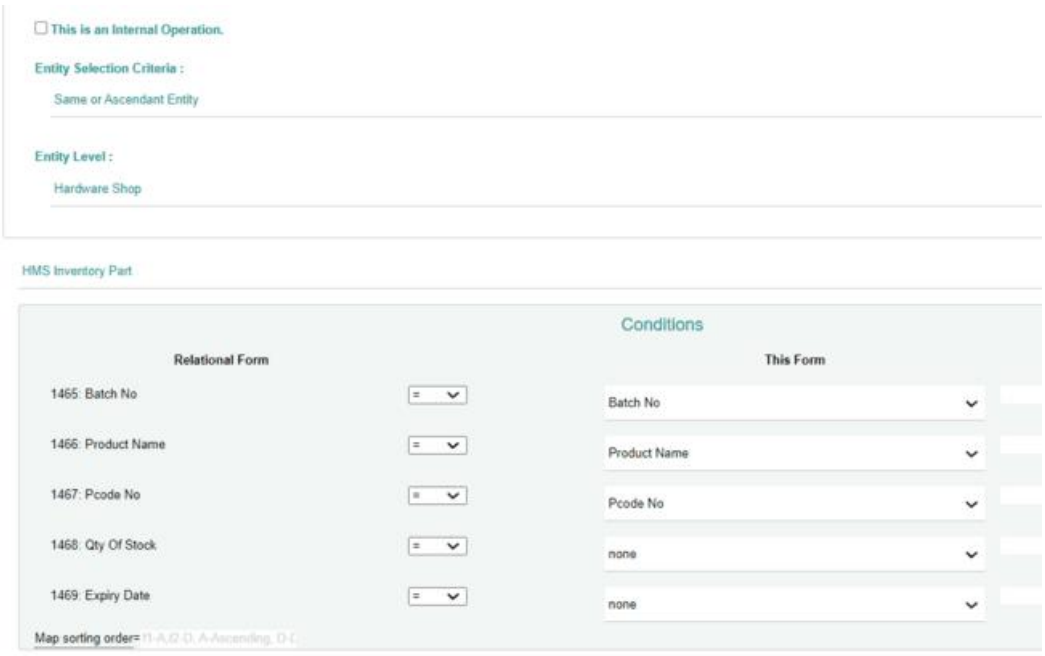
Data fields of the form

Reference	Data Field	Sorting Value	Data Type	Identifier	Key	Default	Attributes	Help Text	
(1465)	Batch No	10	text	1	1		required		Edit
(1466)	Product Name	20	options	0	0		required		Edit
(1467)	Pcode No	30	fetch	0	1		readonly		Edit
(1468)	Qty Of Stock	40	number	0	0		required		Edit
(1469)	Expiry Date	50	date	0	0		required		Edit

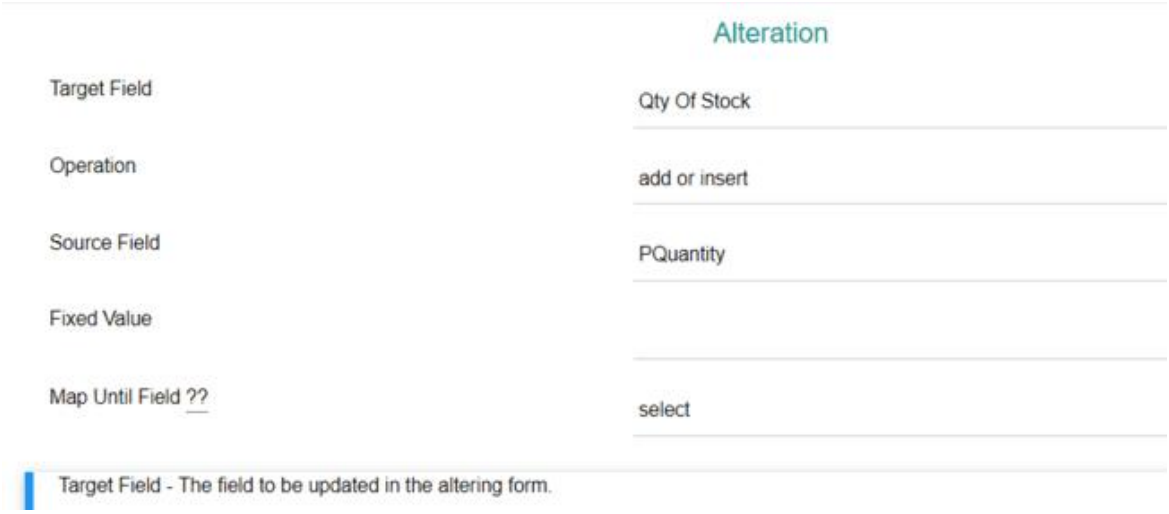
1465,1466,1467,1468,1469,

- Ensure that the 'Quantity in Stock' data field has the data type 'Number' since numerical operations require this data type.

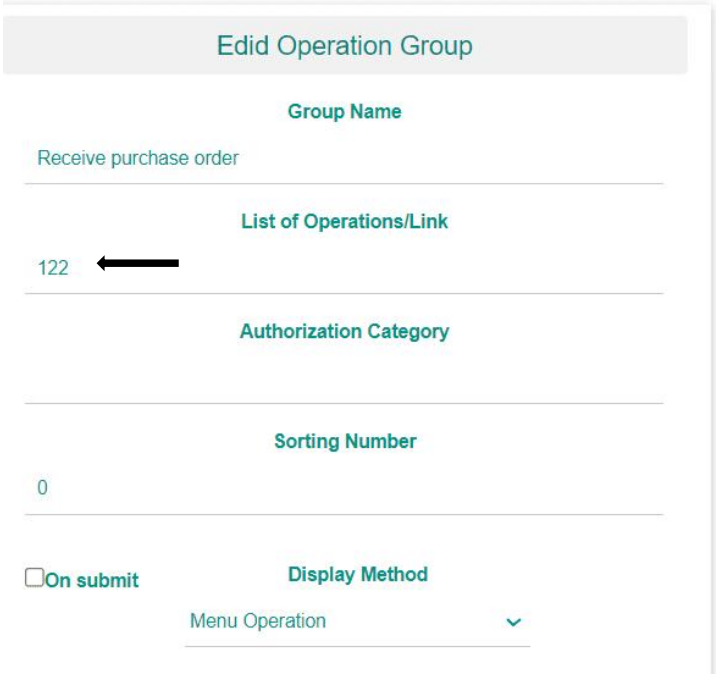
- After setting up the form, you'll need to create operations for it. Start by navigating to the GRN lines form and then access the 'Operations' tab.
- Next, click on 'Conditional Update' and select the entity criteria, along with other settings as described below.



- Specify the target field as Inventory Part ‘Quantity’ since we want this field to be automatically generated.
- Then, set the source field as GRN Lines ‘Quantity’ and choose Add or Insert as the operation.

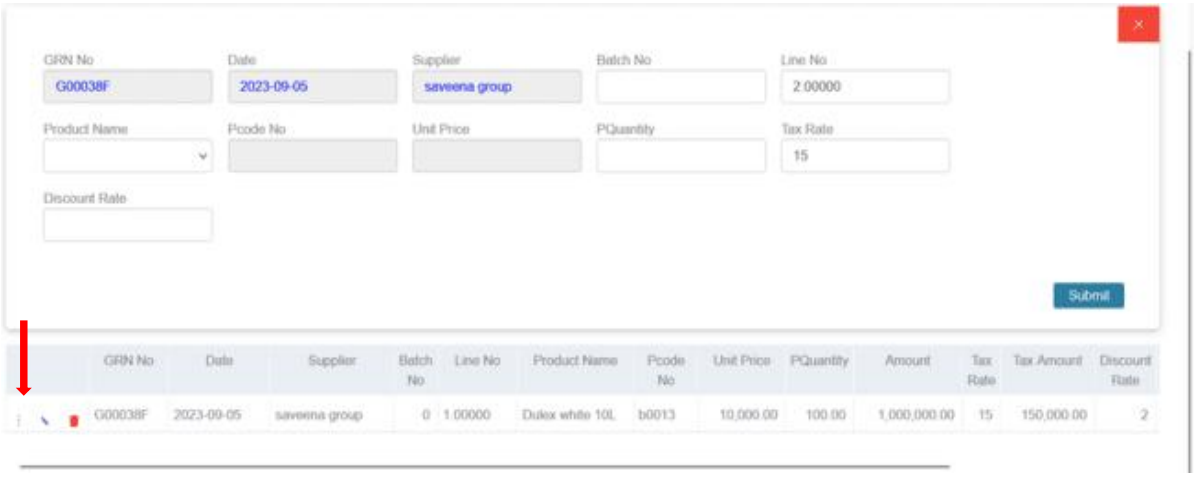


- To execute the operation, configure the group as follows.



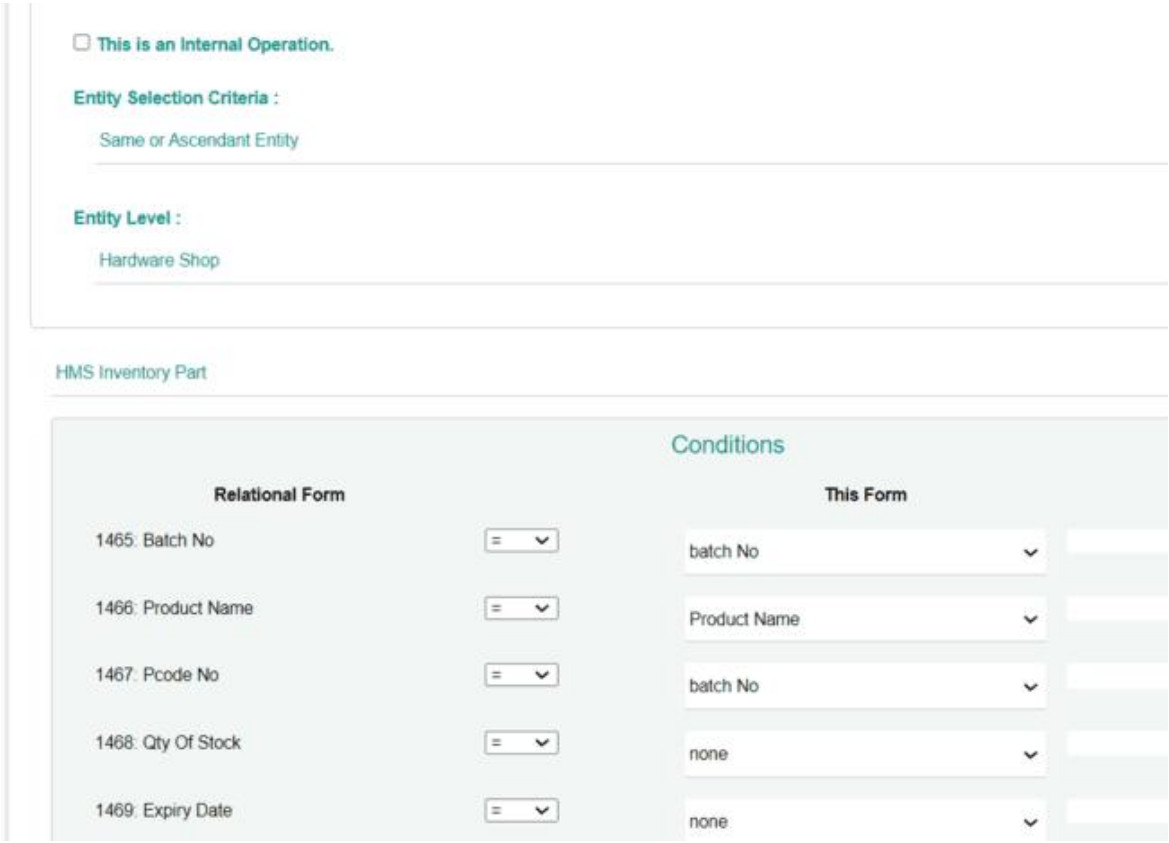
- We have designated 'Operation Type' as the menu operation for receiving purchases.

In the user role, the menu operation appears on the left side as a list with three dots.

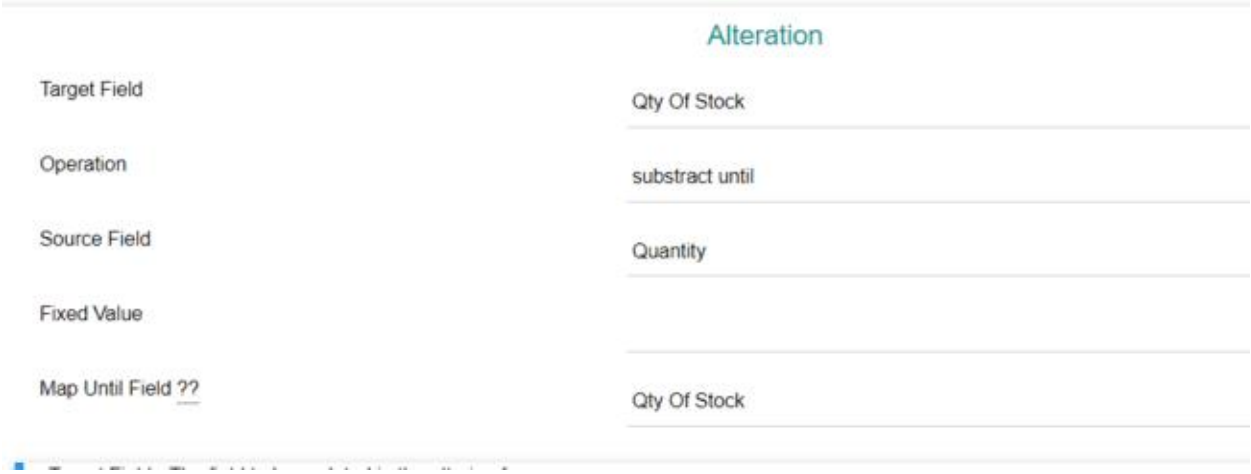


When you click on it, it displays the operation group name that corresponds to this function. You can then click on it to automatically add the item to the inventory.

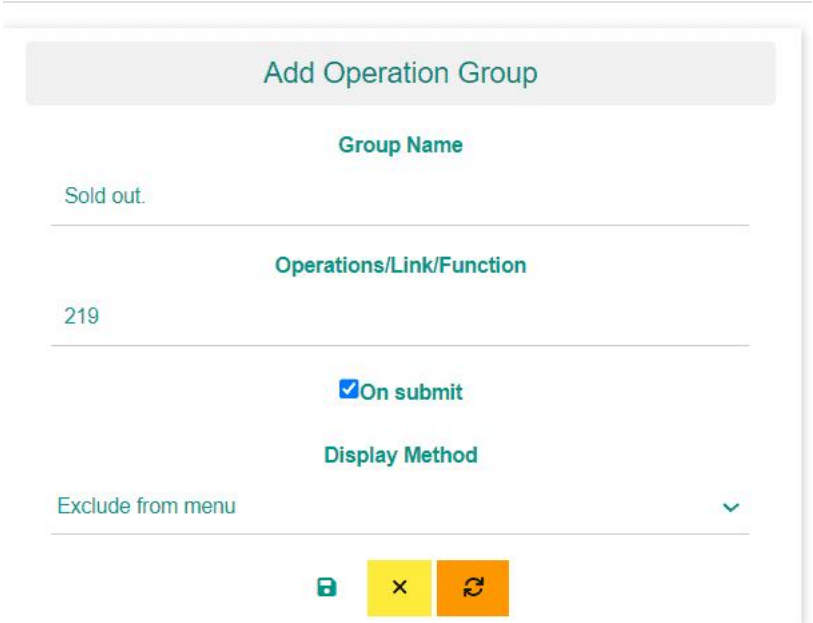
- So far, we have only focused on adding inventory when purchasing goods. Now, we need to manage inventory for sales as well.
  To achieve this, follow these steps:
- Navigate to the 'Invoice Lines' form and select the 'Operation' tab.
- Click on 'Conditional Update' and define the entity criteria and conditions as needed.



- Set the target field as Inventory Part ‘Quantity’, the source field as Invoice Line ‘Quantity’, and choose the operation 'Subtract until' to deduct from inventory. Because if we use the 'Subtract' operation, the inventory quantity can become negative.
- Ensure that the 'Map Until' field corresponds to the target field you've used.



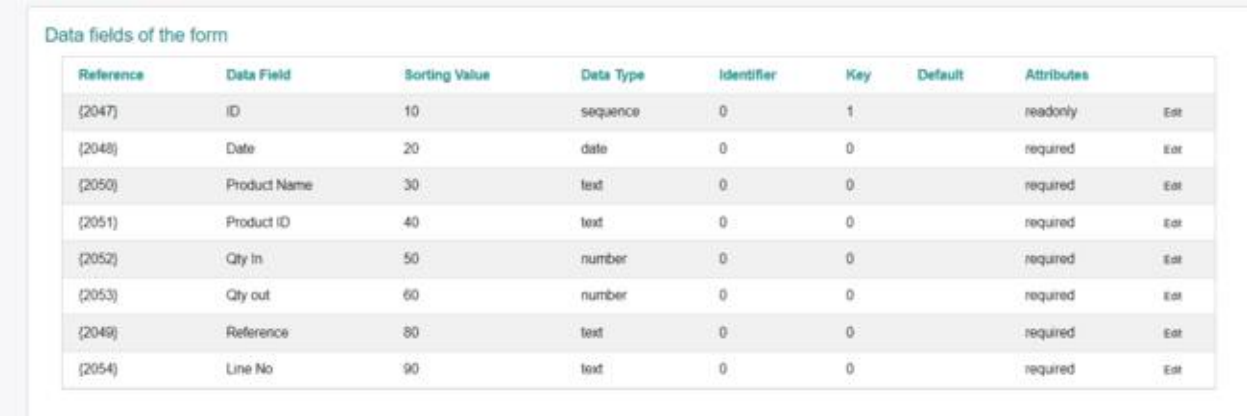
- Configure a group to execute this operation. For greater accuracy, set this operation to trigger 'On Submit.



- If we use 'on submit,' operations are automatically executed when we submit invoice lines.
- Then subtract the invoice quantity from the inventory.

2. How can you automatically enter all transactions in a single form?

- To create an inventory transaction form as outlined below, please ensure that the 'Quantity In' and 'Quantity Out' fields have their data types set to 'Number'.

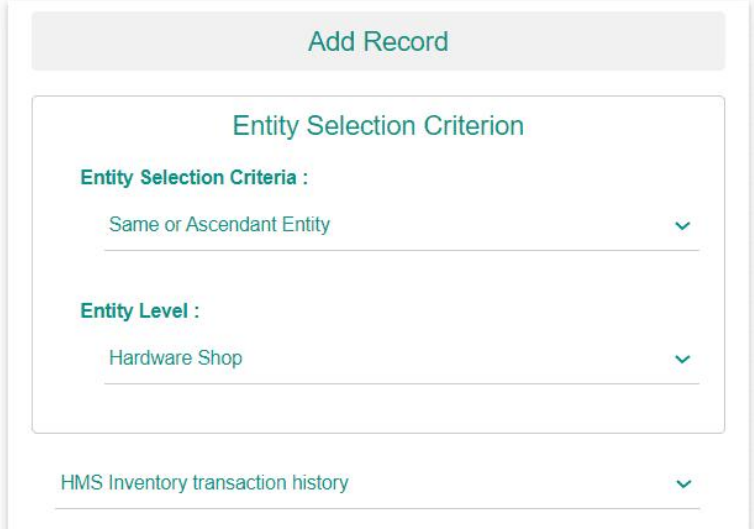


Data fields of the form

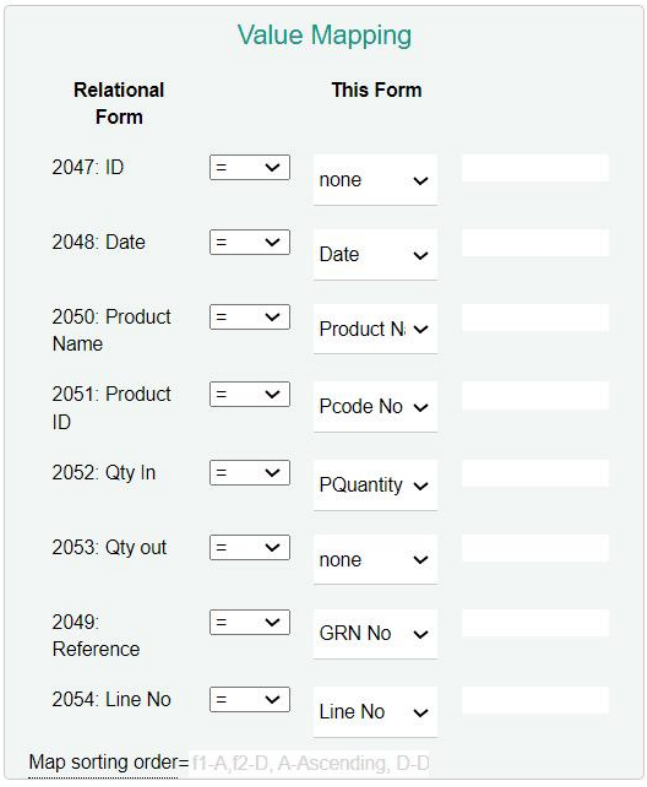
Reference	Data Field	Sorting Value	Data Type	Identifier	Key	Default	Attributes	
(2047)	ID	10	sequence	0	1		readonly	Edit
(2048)	Date	20	date	0	0		required	Edit
(2050)	Product Name	30	text	0	0		required	Edit
(2051)	Product ID	40	text	0	0		required	Edit
(2052)	Qty In	50	number	0	0		required	Edit
(2053)	Qty out	60	number	0	0		required	Edit
(2049)	Reference	80	text	0	0		required	Edit
(2054)	Line No	90	text	0	0		required	Edit

- Next, navigate to the 'Goods Receipt Note (GRN) Lines' section, go to the 'Operation' tab, and create a new record.

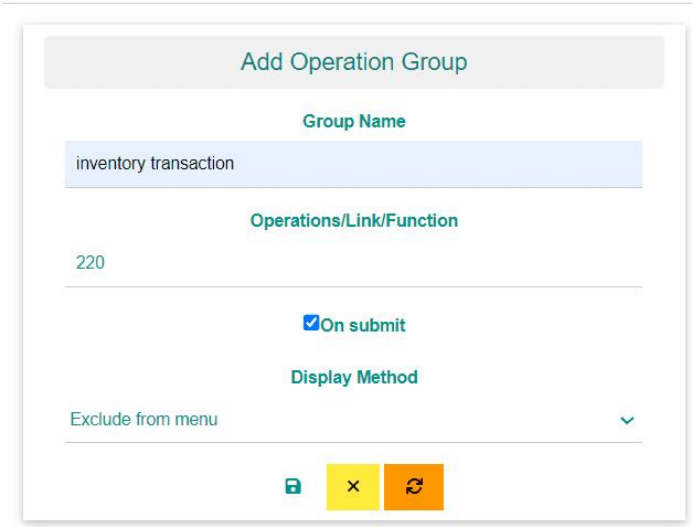
- Ensure that you correctly specify the Entity Selection Criterion, and then proceed with the value mapping.



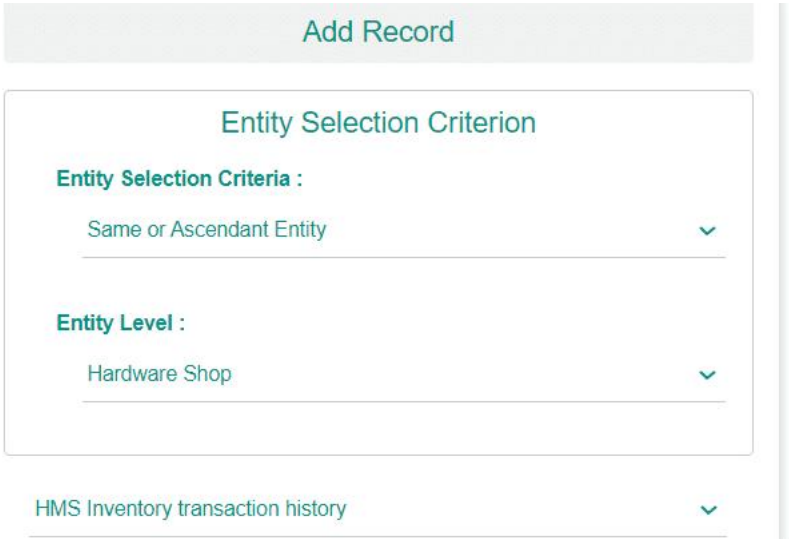
- Make sure to provide the Inventory Transaction Form ID as 'seq' as indicated.
- Also set the 'Purchase Quantity' to be equal to the 'Quantity In', and save your changes.



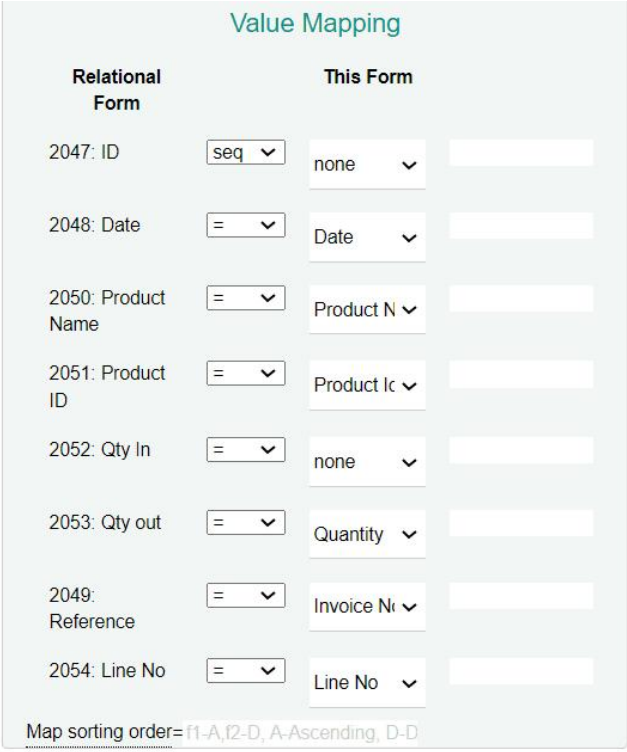
- Lastly, configure a group to execute this operation, and make sure it triggers upon submission.



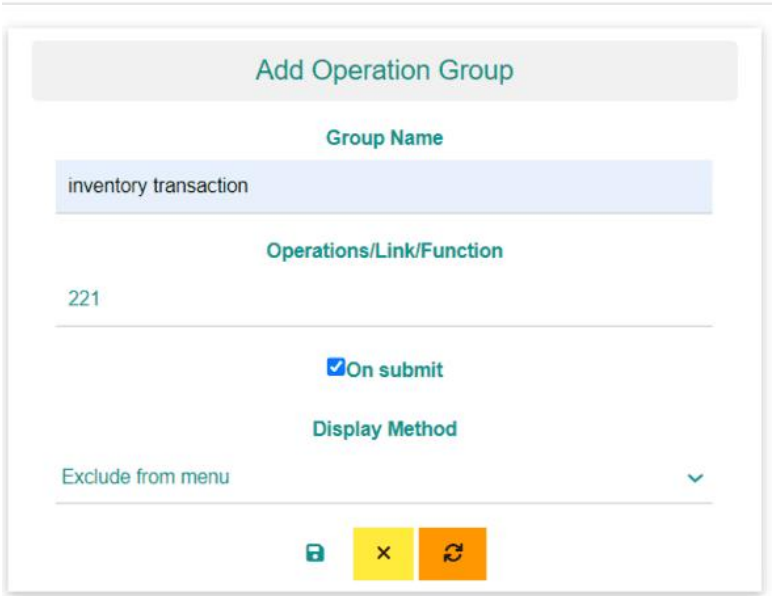
- Now, create a new record for the invoice line, following the same process as you did for the GRN lines.



- Keep in mind that the 'Invoice Transaction Quantity' should be sourced from the 'Quantity Out' data field.



- In this operation, the trigger should be set to ‘On Submit’.



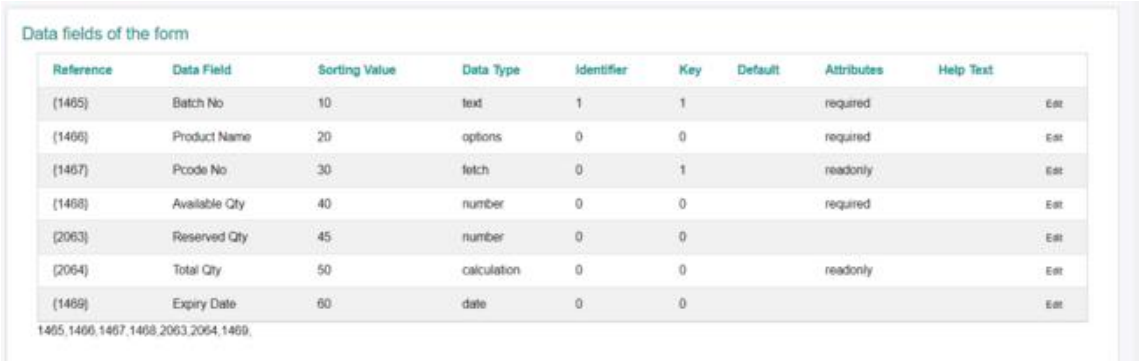
Exercise 02

1. How to implement automatic reservation for customer orders in this solution?

- Let's begin by creating new data fields in the inventory part form.

- In the previous exercise, we created GRN and invoices for purchasing and sales. Now, we will focus on adding the reservation of customer orders to this solution.
- First, let's rename the 'quantity in stock' field in the inventory part form to 'available quantity.' Next, add a data field for 'reserved quantity' with a data type set as 'number.' Then, create another data field for 'total quantity' with a data type set to 'calculation.' The calculation should be defined as follows:

Total Quantity = Available Quantity Data Field + Reserved Quantity Data Field



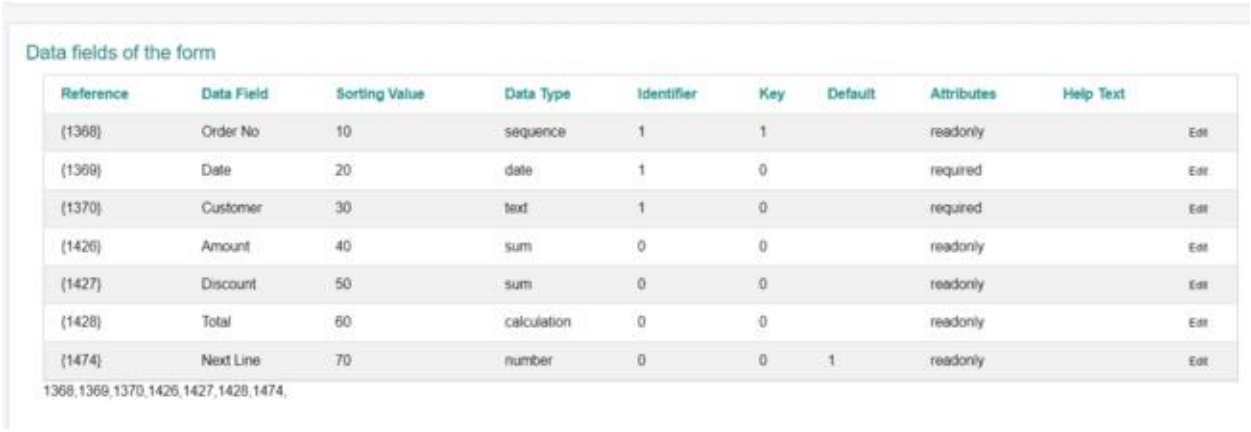
Data fields of the form

Reference	Data Field	Sorting Value	Data Type	Identifier	Key	Default	Attributes	Help Text	
(1465)	Batch No	10	text	1	1		required		Edit
(1466)	Product Name	20	options	0	0		required		Edit
(1467)	Pcode No	30	fetch	0	1		readonly		Edit
(1468)	Available Qty	40	number	0	0		required		Edit
(2063)	Reserved Qty	45	number	0	0				Edit
(2064)	Total Qty	50	calculation	0	0		readonly		Edit
(1469)	Expiry Date	60	date	0	0				Edit

1465,1466,1467,1468,2063,2064,1469,

- With these form changes completed, let's move on to the customer order form.

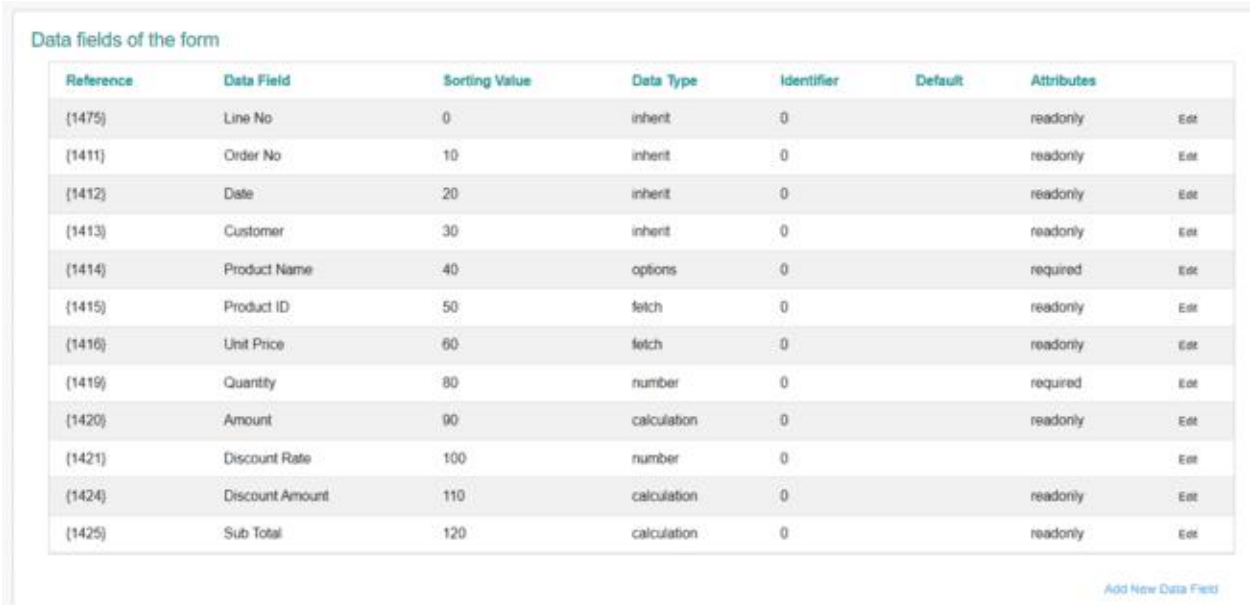
- Create customer orders and order lines in a similar fashion to how we created invoices. Now, let's delve into the specifics of the customer order forms.



Data fields of the form

Reference	Data Field	Sorting Value	Data Type	Identifier	Key	Default	Attributes	Help Text	
{1368}	Order No	10	sequence	1	1		readonly		Edit
{1369}	Date	20	date	1	0		required		Edit
{1370}	Customer	30	text	1	0		required		Edit
{1426}	Amount	40	sum	0	0		readonly		Edit
{1427}	Discount	50	sum	0	0		readonly		Edit
{1428}	Total	60	calculation	0	0		readonly		Edit
{1474}	Next Line	70	number	0	0	1	readonly		Edit

1368,1369,1370,1426,1427,1428,1474,



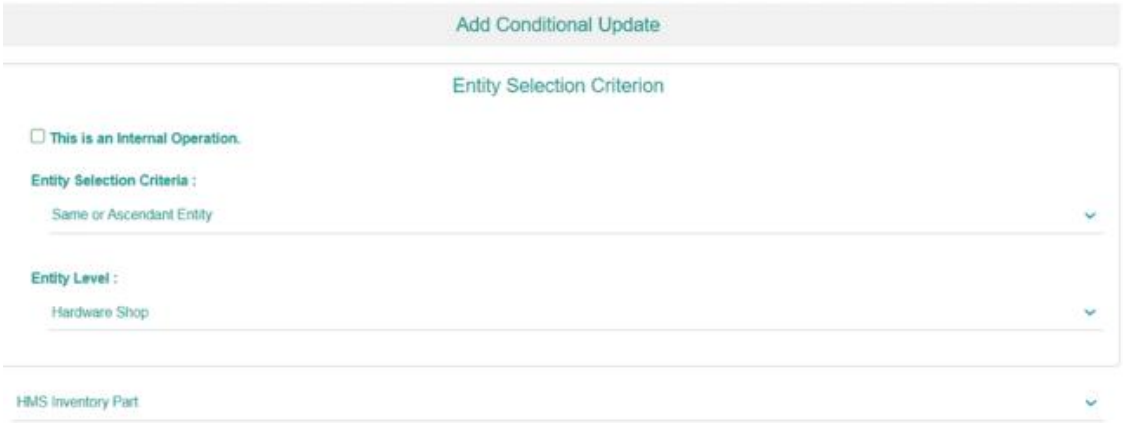
Data fields of the form

Reference	Data Field	Sorting Value	Data Type	Identifier	Default	Attributes	
{1475}	Line No	0	inherit	0		readonly	Edit
{1411}	Order No	10	inherit	0		readonly	Edit
{1412}	Date	20	inherit	0		readonly	Edit
{1413}	Customer	30	inherit	0		readonly	Edit
{1414}	Product Name	40	options	0		required	Edit
{1415}	Product ID	50	fetch	0		readonly	Edit
{1416}	Unit Price	60	fetch	0		readonly	Edit
{1419}	Quantity	80	number	0		required	Edit
{1420}	Amount	90	calculation	0		readonly	Edit
{1421}	Discount Rate	100	number	0			Edit
{1424}	Discount Amount	110	calculation	0		readonly	Edit
{1425}	Sub Total	120	calculation	0		readonly	Edit

Add New Data Field

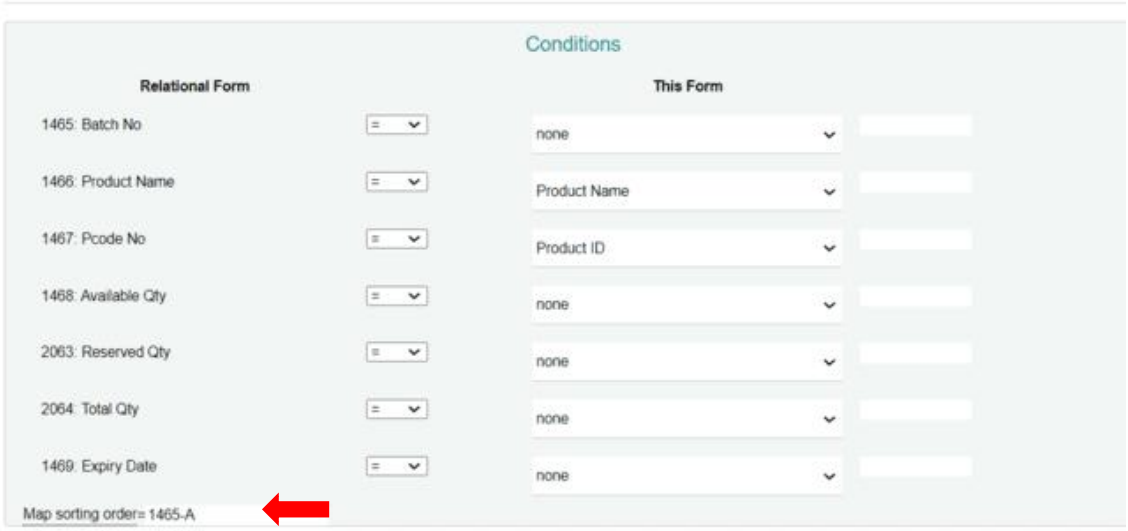
- When utilizing automatic reservation, there's no need to manually enter the batch number in the order lines form if we're selling based on batch numbers. This is because we can configure the batch number as a mapping sorting order in the operation settings. This is the key difference when using automatic reservation.

- Next, navigate to the 'Operation' tab in the order lines form and click on 'Conditional Update.' Specify the entity criteria as follows, ensuring that the condition is correctly applied:

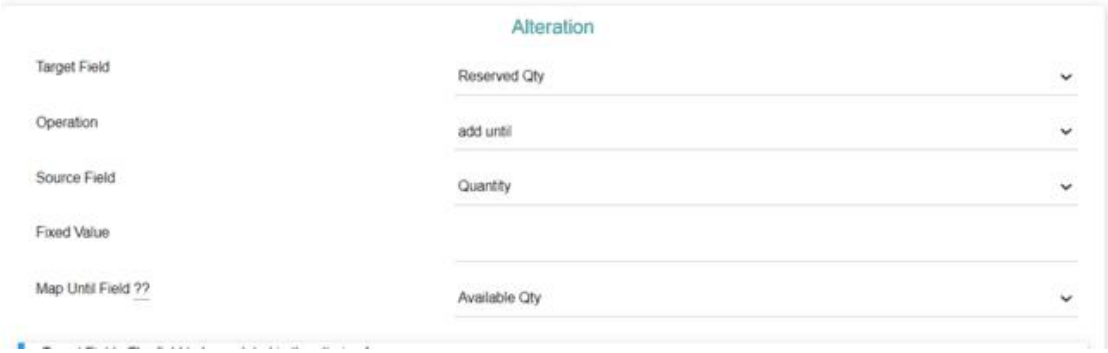


- After setting up the entity criteria, it's essential to establish the mapping sorting order. If we are selling products in ascending order of batch numbers, assign the batch number field to the mapping sorting order with 'F1-A' (for ascending order, e.g., 1>2>3...). If it's in descending order, use 'F1-D' (for descending order, e.g., 3>2>1).

- In this case, assuming the Batch No Field is '1465,' the mapping sorting order would be '1465-A.' This configuration streamlines the automatic reservation process.

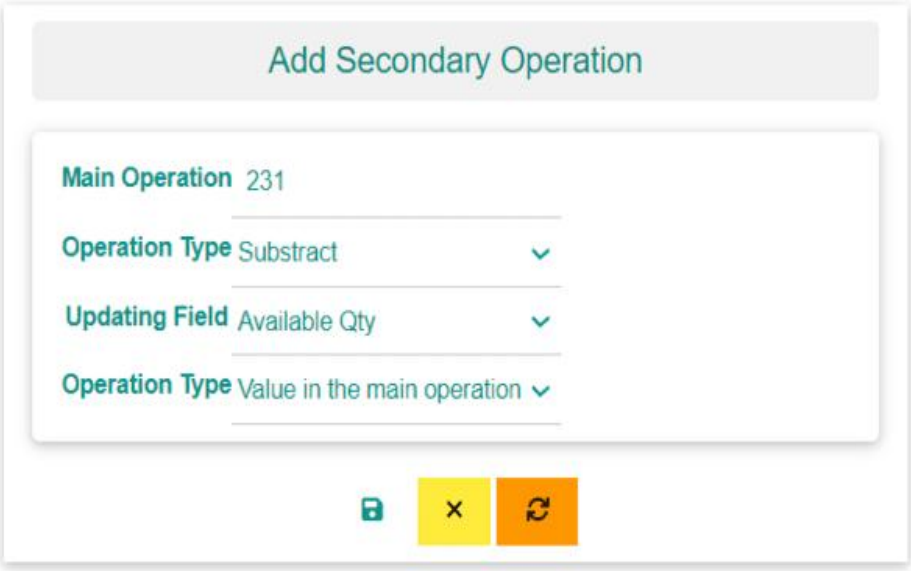


- As for the alteration, the target field should be 'reserved quantity' in the inventory part form, while the source field should be 'quantity' in the customer order lines. Use the 'add until' operation and map until the 'available quantity' field in the inventory part form is reached. It's important to note that this approach prevents reservations exceeding the available stock, avoiding negative quantities.

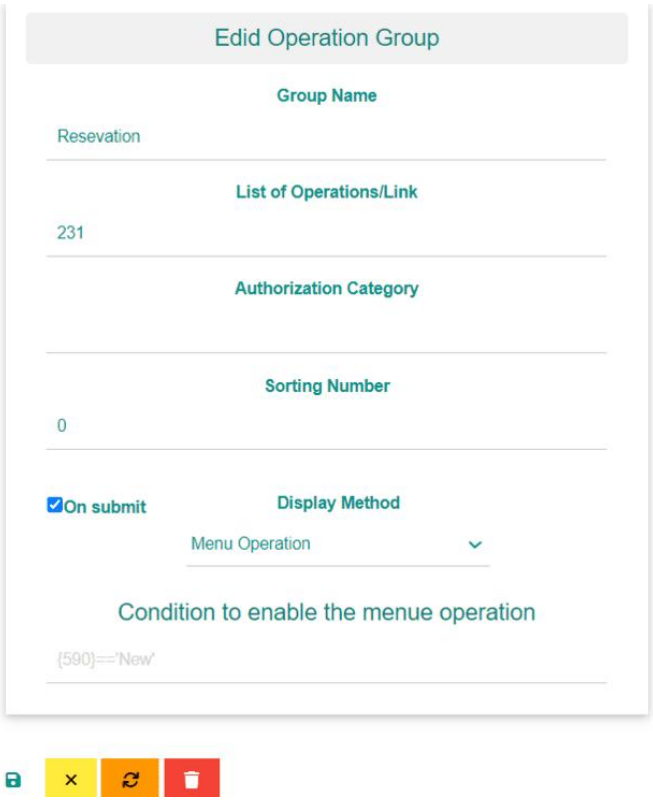


- In this task, we need to perform a secondary operation because we need to subtract the same value from the available quantity. Please follow these steps:
- Navigate to the plus mark in the previous operation.

- Next, add the following condition:



- Add an 'On Submit' action to this task.

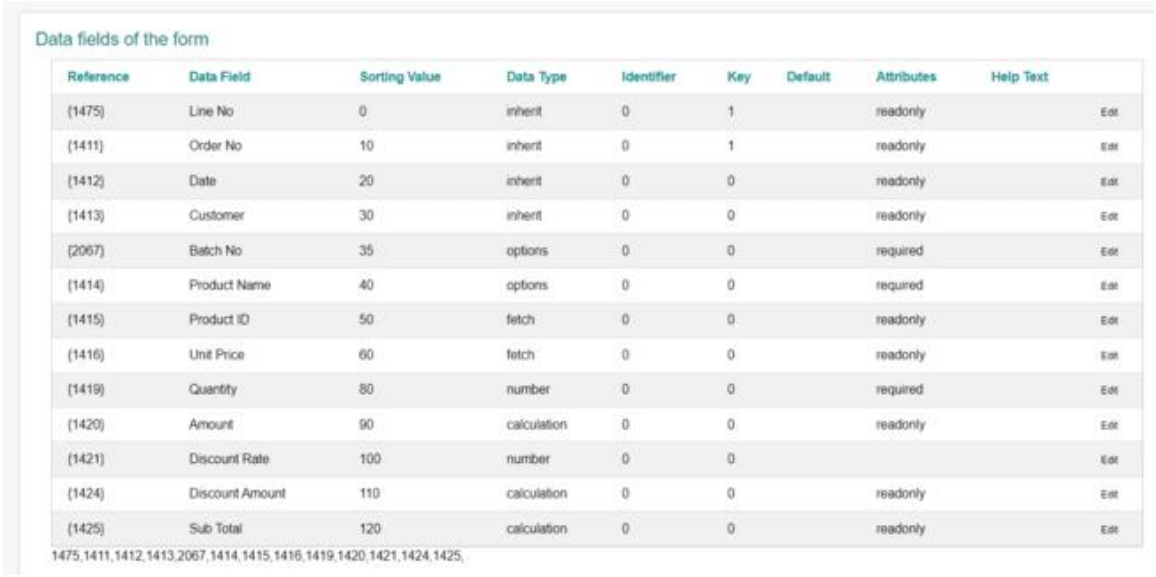


- By implementing these steps, you can effectively manage automatic reservation for customer orders within your solution.

- When the order is delivered, subtract that quantity from the reserved quantity in inventory.
- To do that, you can create a 'subtract primary' operation, just as we create other operations.

2. How to Perform Manual Reservation in this Solution?

- Let's begin by adding a new data field to the customer order line form. This field should be named 'Batch No' with the data type set as 'Option.'
- This should be identified within the inventory part form using the 'Batch No'.

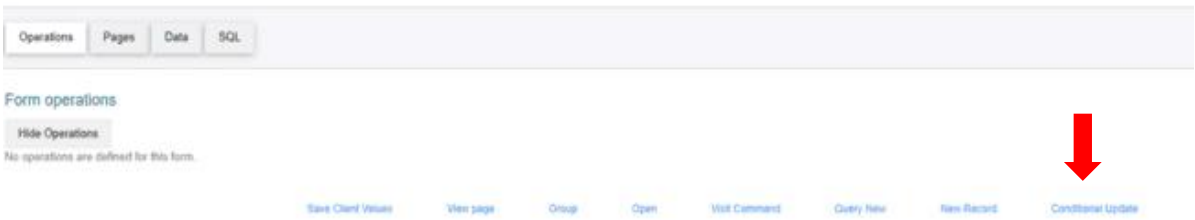


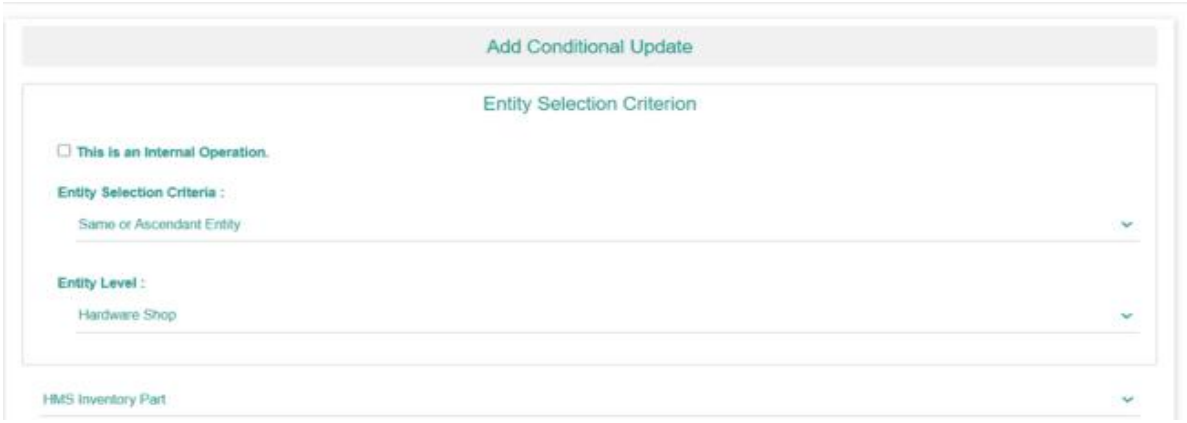
Data fields of the form

Reference	Data Field	Sorting Value	Data Type	Identifier	Key	Default	Attributes	Help Text	
(1475)	Line No	0	inherit	0	1		readonly		Edit
(1411)	Order No	10	inherit	0	1		readonly		Edit
(1412)	Date	20	inherit	0	0		readonly		Edit
(1413)	Customer	30	inherit	0	0		readonly		Edit
(2067)	Batch No	35	options	0	0		required		Edit
(1414)	Product Name	40	options	0	0		required		Edit
(1415)	Product ID	50	fetch	0	0		readonly		Edit
(1416)	Unit Price	60	fetch	0	0		readonly		Edit
(1419)	Quantity	80	number	0	0		required		Edit
(1420)	Amount	90	calculation	0	0		readonly		Edit
(1421)	Discount Rate	100	number	0	0				Edit
(1424)	Discount Amount	110	calculation	0	0		readonly		Edit
(1425)	Sub Total	120	calculation	0	0		readonly		Edit

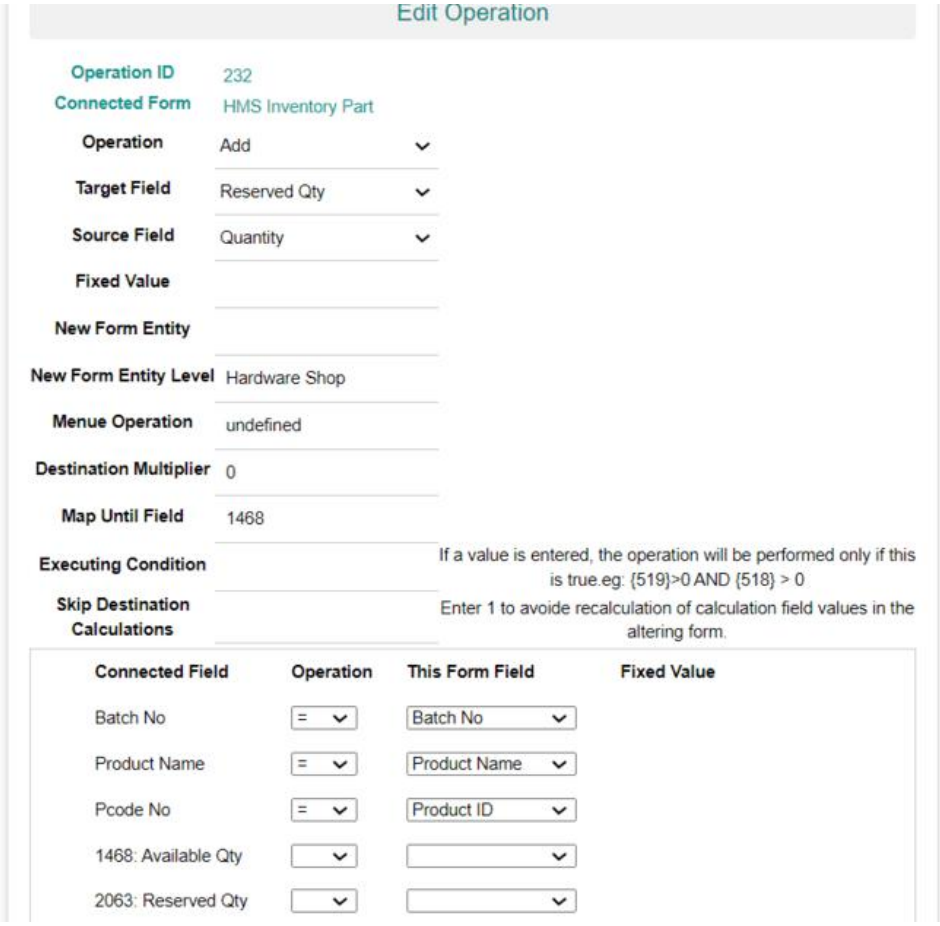
1475,1411,1412,1413,2067,1414,1415,1416,1419,1420,1421,1424,1425,

- Next, navigate to the 'Operation' tab in customer order lines. Click on 'Condition Update' and configure the entity criteria.

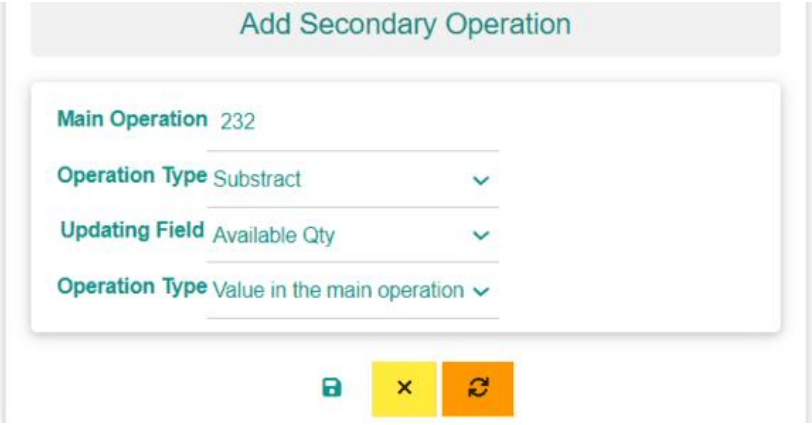




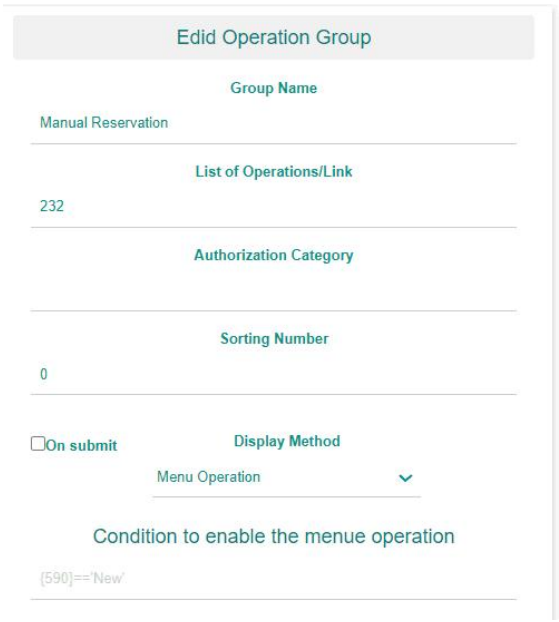
- When defining the condition, ensure that the 'Batch No' is set to be equal in both forms. Keep all other settings the same as for automatic reservations.



- As part of the automatic reservation, add a secondary operation to subtract the available quantity.



- Currently, we have set this operation to trigger 'Menu Operation.'



- When the order is delivered, subtract that quantity from the reserved quantity in inventory.
- To do that, you can create a 'subtract primary' operation, just as we create other operations.

- In the user role, prior to entering the order line, it is essential to begin with an overview of the inventory section. Here, you can verify the available quantity of items. Once you have assessed the inventory, proceed to the order lines and allocate the specific batch number along with the corresponding quantity from that batch.

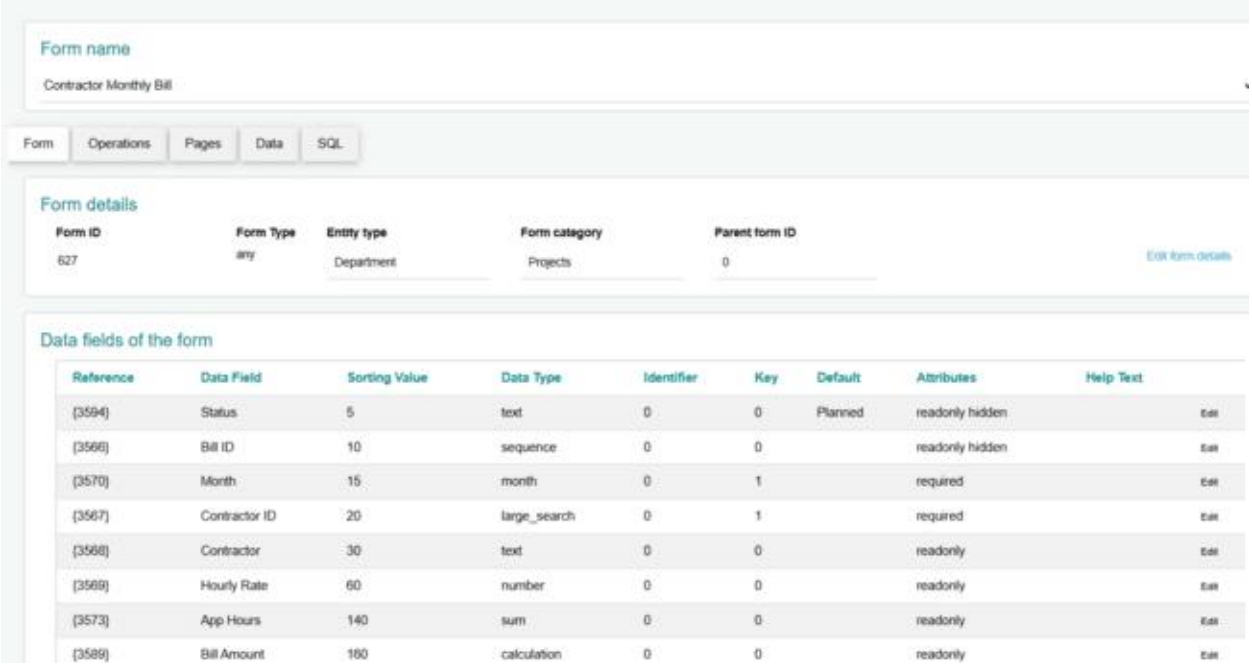
Please note that this manual operation is intended for exercise purposes. There are more efficient ways to handle this process. We can explore a more streamlined approach in a subsequent exercise.

Exercise 03

1. How do you create a Monthly Bill for a Contractor in the system?

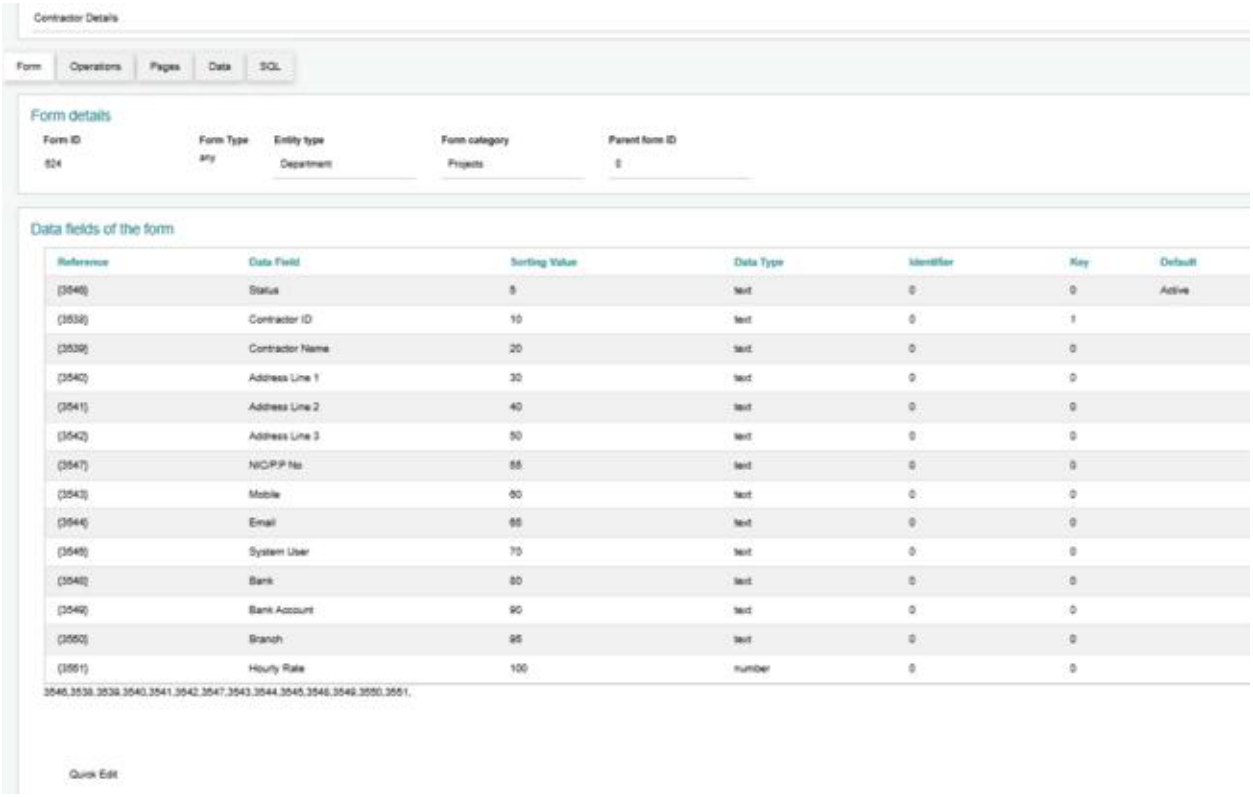
- As a previous exercise, you have to create the Bill Form as a header form. Then, add data fields with contractor-specific details, including what we need to create the bill.

- We can create the monthly bill header with contractor details such as contractor ID, contractor name, month, hourly rate, and status.

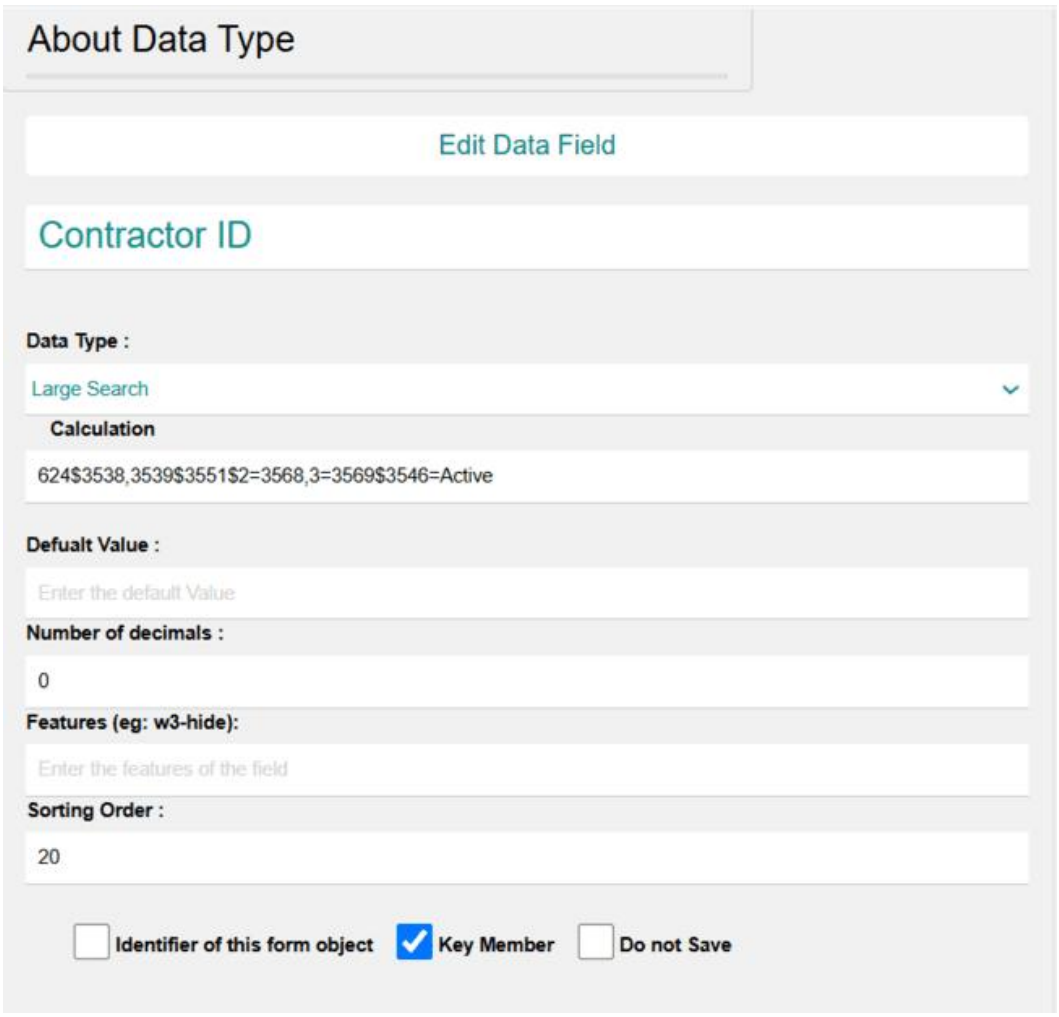


- From the large search, a set of data will be displayed as a drop-down. It can also include filters for specific conditions.Let's look at how to write the syntax for a large search data type.
- In the image below, the contractor details form is shown with their column IDs.

- For the contractor bill, the contractor ID and name will be retrieved from the contractor details form. We will write the syntax to connect with this contractor details form.

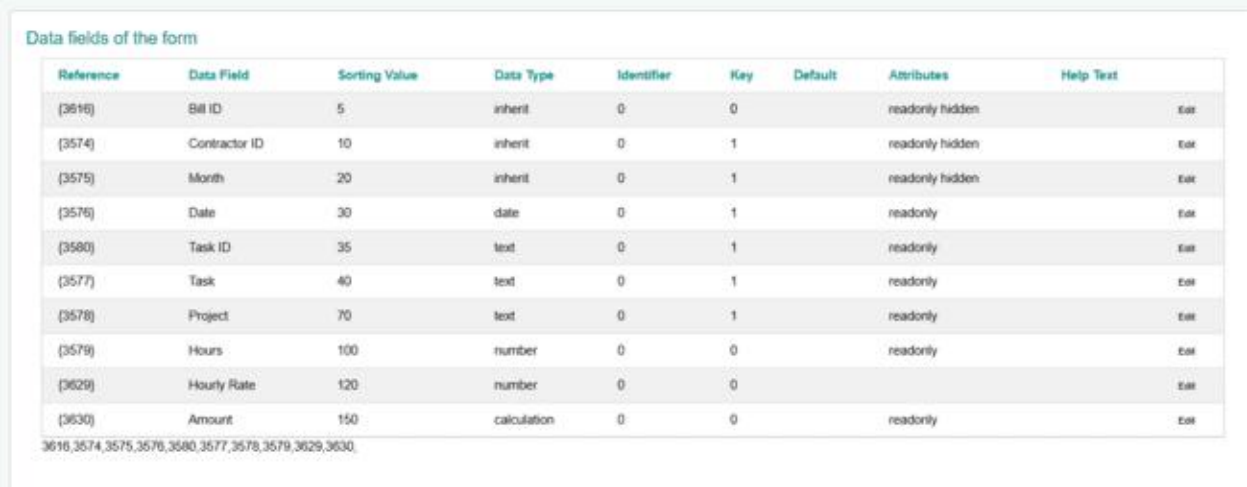


- Now, let's see how to write a large search for contactor ID



- 624 is the form ID of the contractor details form. After the dollar sign, it shows the drop-down data column IDs, including contractor ID and name(3538 & 3539). Following another dollar sign, it displays hidden columns like hourly rate(3551). The next dollar sign specifies the mapping fields, such as where the contractor name should be included in the bill(3539 should be included in 3568)and the rate as well. After the subsequent dollar sign, it defines the filter condition to show only active contractors. Thus, in this drop-down, only contractor details with a status of "Active" will be displayed.(3546=Active)

- Then, create a line level for the bill (sub-form) to enter bill lines, which means the worked time for the month. Set up the sub-form as shown below.



Data fields of the form

Reference	Data Field	Sorting Value	Data Type	Identifier	Key	Default	Attributes	Help Text	
(3616)	Bill ID	5	inherit	0	0		readonly hidden		Edit
(3574)	Contractor ID	10	inherit	0	1		readonly hidden		Edit
(3575)	Month	20	inherit	0	1		readonly hidden		Edit
(3576)	Date	30	date	0	1		readonly		Edit
(3580)	Task ID	35	text	0	1		readonly		Edit
(3577)	Task	40	text	0	1		readonly		Edit
(3578)	Project	70	text	0	1		readonly		Edit
(3579)	Hours	100	number	0	0		readonly		Edit
(3629)	Hourly Rate	120	number	0	0				Edit
(3630)	Amount	150	calculation	0	0		readonly		Edit

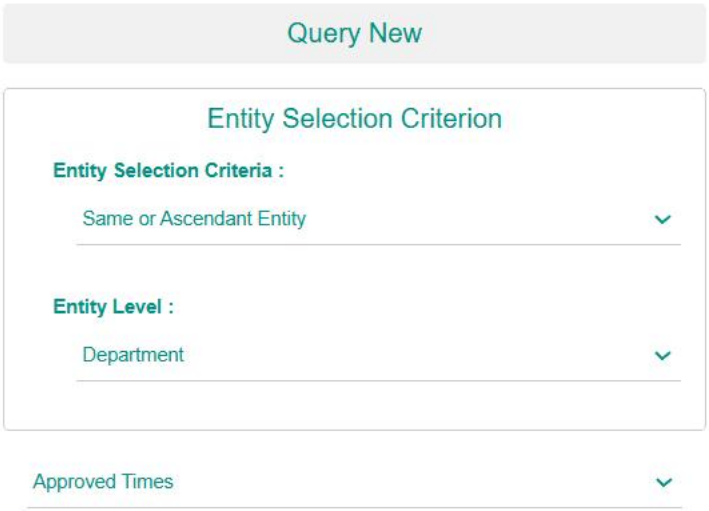
3616,3574,3575,3576,3580,3577,3578,3579,3629,3630,

- Let's see how to get the worked time for this contractor for this month from another form's data. We will retrieve the data from the "Approved Time Details" form for the monthly bill.

- For that, we will use the Query New operation type. It will be executed from the contractor bill header form. This involves three forms:

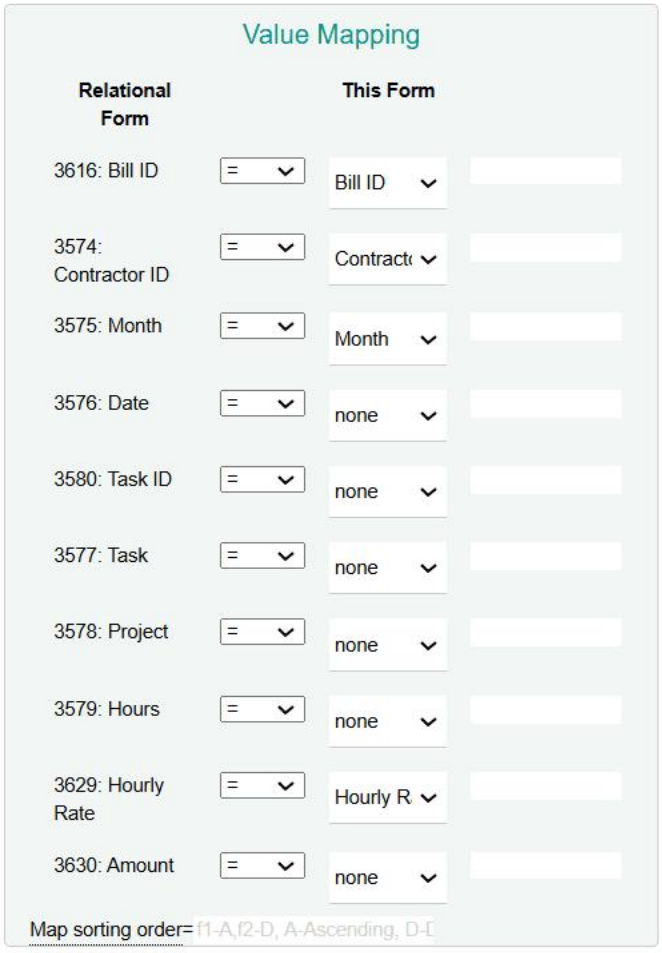
1. The first form is this form, the contractor bill header form.
2. The second form is the query form, where we will get data from the "Approved Time Details" form.
3. The last form is the destination form, which indicates where the data retrieved from the query should be placed (the bill line form).
    - Start by navigating to the Contractor Monthly Bill form and then access the 'Operations' tab.
    - Next, click on 'query new' and select the entity criteria, along with other settings as described below.

- Ensure that you correctly specify the Entity Selection Criterion, and then proceed with the value mapping.

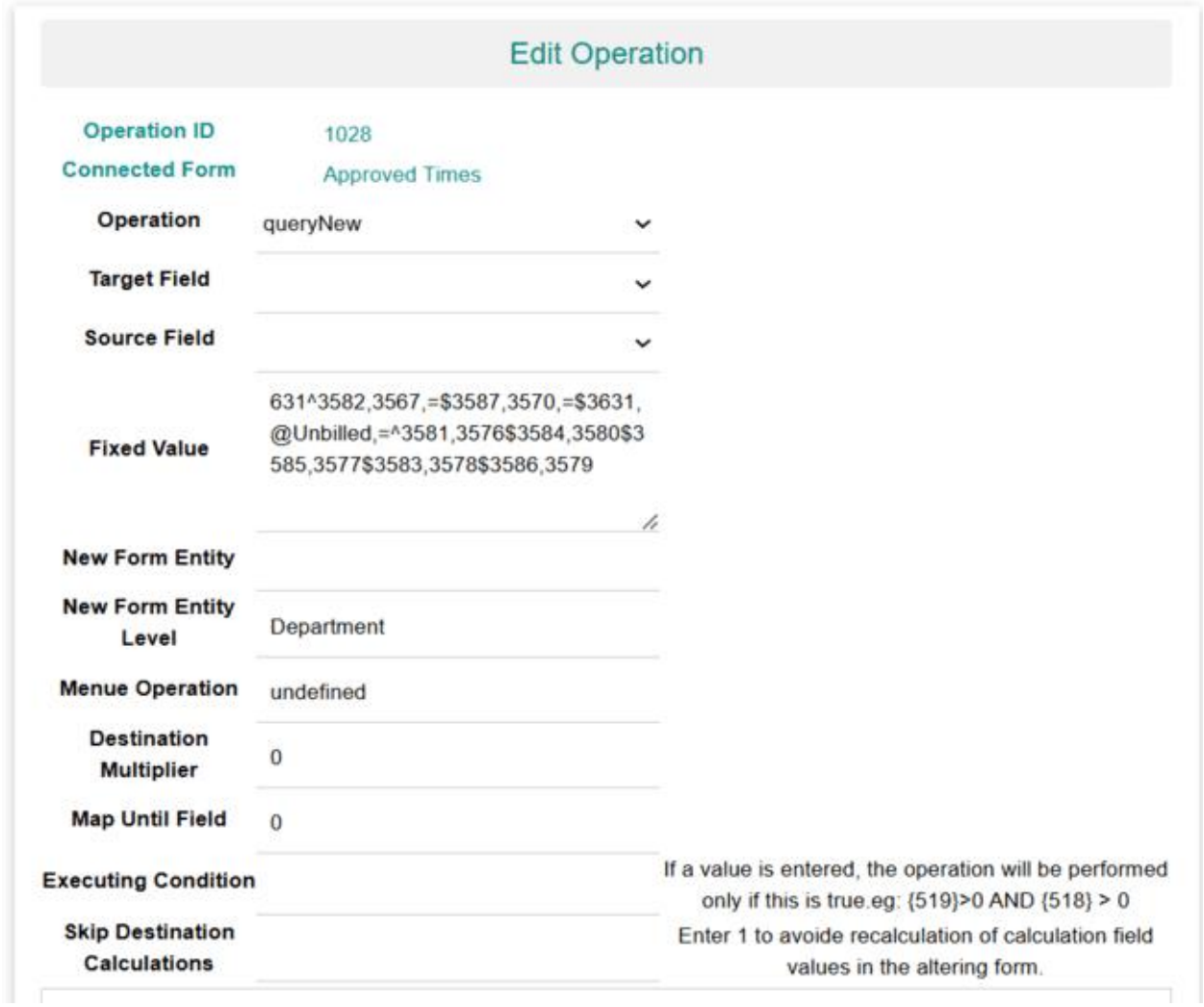


- In this, we are selecting the entity of the destination form (bill line) and selecting the form ID of the Bill Line form, which means the Approved Time Details form.

- Then, value mapping is used to specify what we get from the contractor bill header form to the line form (Approved Time Details form)



- Next, click on the edit button of the operation and type the query new syntax in the fixed value.

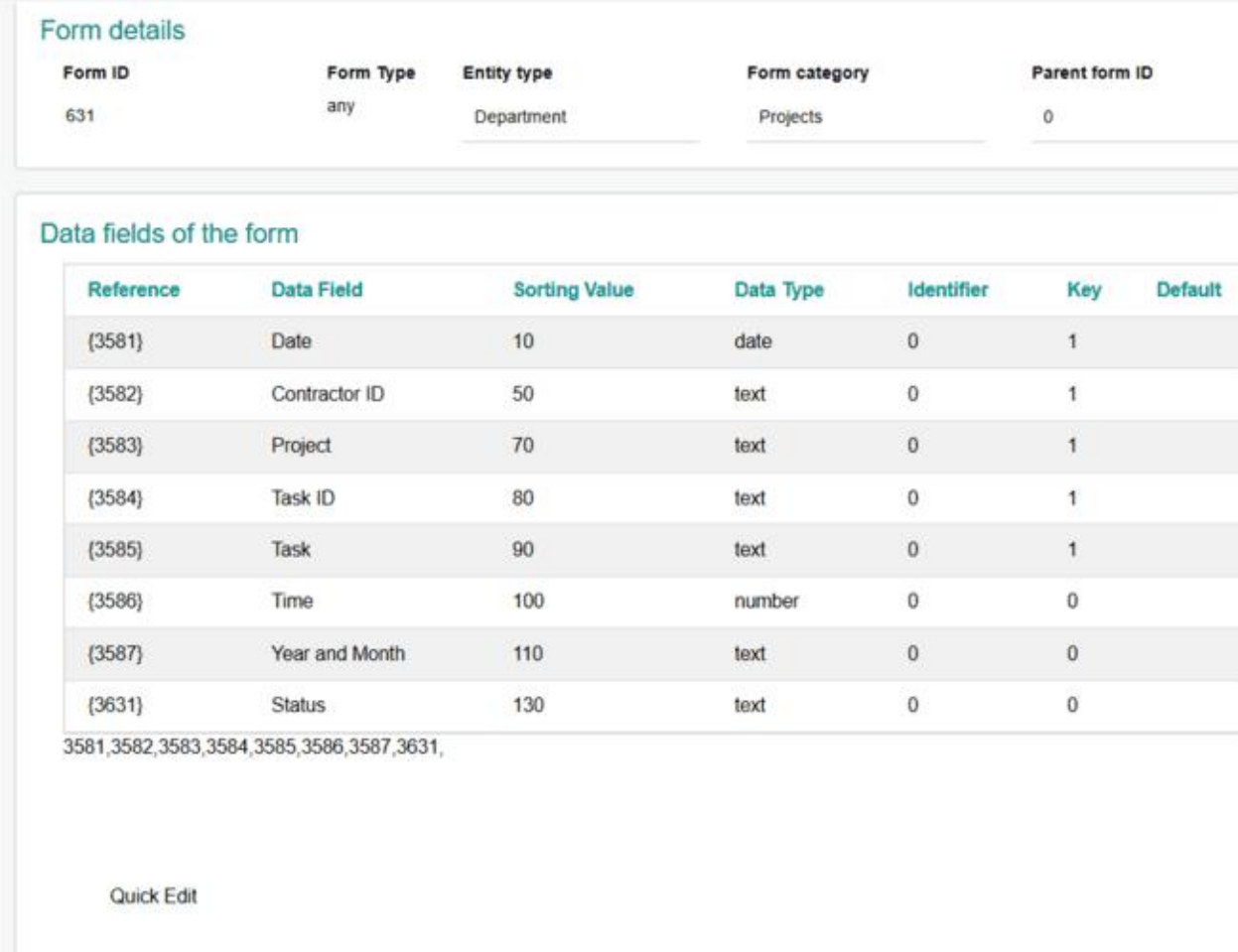


631 ^

3582 , 3567 , = $3587 , 3570 , =$3631, @Unbuiled , = ^

3581, 3576 $ 3584 , 3580 $ 3585 , 3577 $ 3583 , 3578 $ 3586 , 3579

- Below is the time approved details with the column IDs.



Form details

Form ID	Form Type	Entity type	Form category	Parent form ID
631	any	Department	Projects	0

Data fields of the form

Reference	Data Field	Sorting Value	Data Type	Identifier	Key	Default
{3581}	Date	10	date	0	1	
{3582}	Contractor ID	50	text	0	1	
{3583}	Project	70	text	0	1	
{3584}	Task ID	80	text	0	1	
{3585}	Task	90	text	0	1	
{3586}	Time	100	number	0	0	
{3587}	Year and Month	110	text	0	0	
{3631}	Status	130	text	0	0	

3581,3582,3583,3584,3585,3586,3587,3631,

Quick Edit

- 631 is the query form ID then circumflex sign,next filter condition should match this form and the query form (month and contractor ID). Then, again circumflex sign next the query value, which means time-approved details from the "Approved Time Details" form to the bill line (such as time, project, etc.).

Finally, you can give a small description like this: "This is how you can create a monthly bill. By following the steps outlined, you will be able to retrieve contractor details, map the necessary fields, and calculate the worked time for the month from the 'Approved Time Details' form. The process involves querying the required data, applying the filter conditions, and transferring the information to the bill line form to generate the final bill.

Learning Outcome…….

Data Types

Data Types	*Form*	*Data Field*	*Value from*
Fetch Match	GRN Lines	Product No	267^1328^1409,1433,= 267 - product form id 1328 - product forms' product codeNo field id 1409 - product forms' product Name field id 1433 - this form product name field id
Number	GRN Lines	Tax rate	Default value (15)
calculation	GRN Lines	Tax Amount	{Products cost}*{tax rate}/100
Large Search	Contractor monthly Bill	Contacor ID	

Operations

Operation	Operation name	*What should Operation do*
Add or insert	Receive purchase order	Add or insert purchase quantities to inventory
Subtract until	Sold out	Subtract sales quantities from inventory.

New record	Inventory transaction	Auto generate inventory in and out transaction
Add until(primary)	Auto Reservation	Auto reserved customer order from inventory
Subtract(secondary)		
Query New		Creates multiple lines in a form (destination form) taking the values from this form and a third form

Welcome to NEEDLU School!

You are about to have some exciting experience with NEEDLU zero-code software development framework.

New Updates

Month	Area	New Update
24/04	query new operations	In the query new operation it is possible to expand the query records to the sibling entities of the query form. qfm@siblings^qf1,tf1,lo1 $ qf2,tf2,lo2^qv1,dv1$qv2,dv2
24/04	Options_search field	In the options_search query part, fetch_match fields can be used to filter records
24/06	n3-siblings	This form class is used for view forms and view_list forms if the data should be retreived from the sibling entities of the view form.
24/06	n3-openSubform(x)	This form class is used select the subform x, when the main-form instance is opened.

Needlu Structure

Needlu Structure consists of the following concepts.

- Entity Level
- Entity
- Forms and Records
- Subforms
- Form Category
- Data Type
- Operations
- Operation Groups

Entity Level

Entity Level is a blueprint of entities in a hierachy. For example you can create entity levels named Country, States, Village. And then you can make entities of entity level of country such as US, Brazil, India. Then you can make states of US suchg as Florida, Georgia, Texas of entity type state under the entity US.

Entity Levels are used to group and structure the forms. Each form belong to an entity level.

Entities

Each Entity is belogned to an entity level. An entity level can have one or many entiies. An entity should have a parent entity and may have children entities as well. End users are granted access for one or many entities. If the user is granted access for an entity, she can access forms of the entity level of that entity.

Forms and Records

Forms are used to enter, edit and show records. Each form belongs to an entity level. Each record enterd in a form belong to an entity (of the entity level of the form).

Subform

Subforms can be added to forms. There are several types of subforms (Any, View, View Select, View List) whichs are used for diffrent purposes. New records are enterd in 'Any' type subforms. Those records can be considered as children records of the recrod of the main form.

Form category

A set of forms are categorised to a form categary. Access for users are grated for form categories not for individual forms. First users should be granted access for entities. Then the users will be given access to required form categories. Therefore the users will have access to the form categories of the entity (or entities) that she has given access to.

Data Types

You create fields in the forms. Each field has a data type. There are many data types in Needlu. You will learn more about data types.

Operations

You can add operarions to a form. you can use opeartions to tell what shoud be done either when a form is saved or when a menu operation is called.

Operation Groups

You shoud group one or more opearations to an operation group. You can select which operation groups shoud be the menu operations and which operation groups shoupd be called when the form is saved.

User Roles

There are three user roles in Needlu

- Admin
- Editor
- User

Admin

Admin creates users of types Admin, Editor or User. Admin grant access for User type users to entities.

Editor

Create and edit application programs.

User

User use the application programs.

Reserve Operation

Map sorting order syntax:
* f1-A,f2-D

* This means sort the mapped values by field f1 by ascending order then by field f2 by descending order.

Link Button

You can create a button in a form which upon click a new tab is opened for a given url.

How to

Enter a Group Operation with the display type link_button. Enter the link in the List of Operations. You can refer to field values in the form by including a calcuation formula in the url between <cal> and <cal/> tags. For example you can have a link button which open whatsApp chat with the mobile number stored in the field ID, 123 by;
https://wa.me/<cal>{123}</cal>?text=Hi, Please use this link to register. http://localhost/NEEDLU-2.0/register_form.html

Page Design

Cards

Required Data fields will be shown as a list of cards.

Enter the response seperated by commas (eg: 67,68,89)

Enter the filter criteria

Filter citeria
Example for filter cirteria : 23=Released,28>0

For the right-hand side values you can use @today and @user to indicate the current date and the current user.

Note : response and filter criteria is same as for tables.

Table

Required data records will be shown in tabular form

Enter the response seperated by commas (eg: 67,68,89)

Response can be included a sorting column, sorting order (ascending or descending), the maximum number of records, and group-by column as well. (Eg: 67,68,89$89$D$50$68)

You can retriev the records of an acsendant entity(It is not necessary to have the access to the acsendant entity to see the records via a table.) To retreive records from the ascendant entity you may add the eneity level as the fifth parameter of the response. eg: 78,79,80$$$$$Site.

Enter the filter criteria.

Filter citeria
Example for filter cirteria : 23=Released,28>0

For the right-hand side values you can use @today, @entity and @user to indicate the current date, entity and the current user.

You can refer to a field value using {x}. ex: 25={27}. Please note that you cannot refer to field values of data types fetch match, fetch calculation and fetch sum in the righthand side of the query

Drilldown Table Values

This is used to drill down data in tables. Detail table is shown underneeth the table row.

Enter the expanding query in Page Designer content. Format is 'response$filter'. (Eg : 148,151,145$146=@1,147=Reserved. Filter is field ID 146 of detail form should be equal to first in the response and the field id 147 is equal to "Reserved"). Also it is possible to use @user, @thisMonth and @today in the filter.

Limitation : Filter is limited to the = at the moment. (This will be improved on request).

Conditional Formatting

Define conditional formatting in the widget_style column

Formatting filed ^ value field ^ value1,classes1$value2,classes2$...

Eg: 25^33^Invoiced,w3-green$Delivered,w3-orange$Released,w3-yellow

Count

The number of records filetered by the given filter will be shown.

Enter the form ID in response column.

Enter the filter criteria

Filter citeria
Example for filter cirteria : 23=Released,28>0

For the right-hand side values you can use @today and @user to indicate the current date and the current user.

Get Count by SQL

Alternatively, it is possible to get the count by a sql query. The sql query to get intances that are supposed be counted should be written in the response column. The filter column should be left blank.

eg: SELECT ins FROM t5 WHERE c45="Active"

Visit another page on click

Add the page name in the Expanding Query column to make the section clickable. And on click, the page given will be visited.

Sum

The sum of a field of records filetered by the given filter will be shown.

Enter the field ID in response column.

Enter the filter criteria filter column.

Filter citeria
Example for filter cirteria : 23=Released,28>0

For the right-hand side values you can use @today and @user to indicate the current date and the current user.

In the Widget Title column, you may add 'n3-decimal2' to get two decimal places. In the Widget Title column, you may add 'n3-commaformat' to get the number forma with commas.

Visit another page on click

Add the page name in the Expanding Query column to make the section clickable. And on click, the page given will be visited.

Stack Charts

Select the Content type 'stack_charts'.

Enter the field ids of x-axis and y-axis seperated by commas in the Response.

Enter the filter criteria

Filter citeria
Example for filter cirteria : 23=Released,28>0

For the right-hand side values you can use @today and @user to indicate the current date and the current user.

Gantt Chart

Shows projects or chars in a gantt chart.

Enter the field IDs which represent the following values in the response column.

resource, task name, start date, end date

Resource is what this task utilizes. It may be the machine or the assignee of the task. The bars in the gantt chart are applied different colours for the resources.

Enter the filter criteria filter column.

Filter citeria
Example for filter cirteria : 23=Released,28>0

For the right-hand side values you can use @today and @user to indicate the current date and the current user.

Options

The values of a perticular field is retreived to a dropdown list. When the value is selected by the user the respective records from another form can be displayed in given manner in the Expanding Query column.

Enter the field ID in response column.

Enter the filter criteria filter column.

Filter citeria
Example for filter cirteria : 23=Released,28>0

For the right-hand side values you can use @today and @user to indicate the current date and the current user.

Enter the string in the Expanding Query column in the folleing mannar. This explains the display of data upon selection of an option.

Expanding Query
response $ filter $ data-display-method $ cell-id

Eg: 123,245,130$125=thisVal,133=Planned$stack_charts$4

Int the fileter, 'thisVal' refers to the selected value by the user. It is possible to enter any content type (table, charts, stack_charts, gantt_chart) as the data-display-method. Data will be displayd in the cell related to 'cell-id'.

Charts

Select the Content type 'charts'.

Enter the field ids of x-axis and y-axis seperated by commas in the Response.

Enter the filter criteria

Filter citeria
Example for filter cirteria : 23=Released,28>0

For the right-hand side values you can use @today and @user to indicate the current date and the current user.

html_page

Upload the html page to the server.

Select the Content type 'html_page'.

Enter the page name in the Response column.

In the page section the content of the respective html page will be shown.

New Record Button

Upon clicking this button, a form will be opened to enter a new record.

Enter the form ID in the response column.

Click Content New

Upon clicking this section, a form will be opened to enter a new record.

Enter the form ID in the response column.

Click Content Page

Upon clicking this section, the user will be navigated to a page.

Enter the page name in the response column.

Query Table

Query table is used to retrieve data when there is more than one form (table) is involved in the query. You can write the SQL query directly in the response column and put the column names in the Filter column.

Ins (or t25.ins type) column will be automatically converted to links to be navigated. It is important to handle entities in the query. @entity can be used to get the current entity and @entity^entit level^ can be used to get the name of the entity's ascendant.

@user can be used in the query to get the current username. Remember to use LOWER() in the where condition (eg: LOWER(c34)=LOWER(@user)).

- @thisMonth - get the month number
- @thisYear - get the year number
- @lastYear - get the last year number
- @today - get the date string
- @entity - get this entity
- @entity^entit level^ - get the entity's ascendant at the given level.
- @entity^descendants^ - get all of the descendants including the current entity. Since descendants are multiple, you should use "IN" operation in the sql select query. (eg: "WHERE entity IN (@entity^descendants^)"
- @value(formId,fieldId) - Use this to refer to single record froms. For example to refer to company name. The formId may belongs to an ascendant level. You should use double quotes for string values.
-

Table Headers : Enter table headers seperated by commas in the filter column.

Drilldown Table Rows

It is possible to navigated another table upon clicking the row in a query table. Write the query in the reposnce column in another cell. You may refer to the column values in the clicked row of the first table as @row(1), @row(2), ..etc. Put the cell id of the second table in the expanding query column in the first table. You may add any content before the second table in its Widget Title column. You can use @row(x) there as well.

Refer to the Trail Balance page in the ERP2 for an example of this.

Date Table

Date Table is used to retrieve data inbetween two dates selected by the user.

User can select from Date and To Date and click the Submit button. These will be automatically available for a Date Table.

Follow the same instructions as you need to provide a Table. Additionaly enter the field ID of the date field which the date range will be considered in the expanding_query column.

Query Stack Chart

You can give the query as a SQL query to make a stack column chart or a column chart. The values will be shown in tabular form as well.

Enter the SQL query in response column. @entity and @user can be used in the query (eg: entity = "@eneity", c45="@user")

Enter the column names in the filter column.

Query Value

You can give the query as a SQL query to get a value. You must put the alias "value" for that. Ex: "SELECT c30 AS value FROM..."

Check the Query Table for syntaxes that can be used in the query.

Page Designer-Content Type Options

Depending on the content type you can use thes column to set options for cards, charts,..etc

1. Cards-Style classes for a card
2. Charts- Chart options as per google charts.
   For example chart chart options can be set as follows.

   "title":"pieHole": 0.4,"width":500,"height":500,"title":"Sales per Customer"

Upload records by a csv file

Users can update multiple records to a form using a csv. Editor can create a page to facilatate that.

- Select the Content type, upload_csv.
- In the respose column, enter the form id and the feild is to upload in the csv using the following sintax.
- form id$field id1, field id2, fielld id3
- 45$458,561,785,661,777

Form Page Design

Page Content

You can write any html code.

To get a field value in the form, put the field id inside '{' and '}'. To avoide applying number format, use '{t' insted of '{'.

Enter calculation formula inside '<cal>' and '<cal/>' tags. Inside these tags you can:

- Write any calcualtion formula
- Write fetch_match formula
- Use @entity, @user, @today
  You may use the followings as well.

- @clientImages/my_logo.png
- @value(form_id,field_id)- This can be used to get values of the forms where only unique record is allowed for an entity. This can be used to get company details.
- Any calculation formula

Charts

Select the Content type 'charts'.

Enter the field ids of x-axis and y-axis seperated by commas in the Response.

Enter the filter criteria

Filter citeria
Example for filter cirteria : 23=Released,28>0

For the right-hand side values you can use @today and @user to indicate the current date and the current user.

Form Table

Form Table is used to show a table of records from abother form filtered based on values of this instance.

Write "table" in the Content Type column. Refresh the page. Hover the mouse to row menu. Click the "View/Modify Table Query". Enter the view form ID. This is the form which you need to retrieve data from. Then enter the filert criteria. Enter the required filed ids seperated by commas ans field widdths seperated by commas in sequence as in the example below.

56,57,60^100,300,100

Here 56,57,60 are the field ids and 100px, 300px, and 100 px are the repective widdths of those columns.

Response can be included a sorting column, sorting order (ascending or descending), the maximum number of records, and group-by column as well. (Eg: 67,68,89$89$D$50$89^100,300,100)

In case that you need to expand the query results to the sibling entities, you can add the third paramater **siblings** (Eg: 67,68,89$89$D$50$89^100,300,100^siblings)

Enter the filter criteria.

html_page

Upload the html page to the server.

Select the Content type 'html_page'.

Enter the page name in the Response column.

In the page section the content of the respective html page will be shown.

Page Designer-Content Type Options

Depending on the content type you can use thes column to set options for cards, charts,..etc

1. Cards-Style classes for a card
2. Charts- Chart options as per google charts.
For example chart chart options can be set as follows.

"title":"pieHole": 0.4,"width":500,"height":500,"title":"Sales per Customer"

Topic Table

Shows multiple tables under topics taken from a filed. You can specify a fieldid to be taken as the topic field, and a sorting criteria.

Display Fields $ Sorting fields $ Sorting Order ^ Topic field ^ Lengths of Display Fields

eg: 334,335,336,337$331,335$A,D^331^100,600,100,50

Generate Barcode

In a form page, you can include a barcode for a combination of field values in the record. This can be done by adding the svg tag in the content column of the page layout.

```
<svg id='barcode'> formula goes here </svg>
```

The formula should necessarily include the string that is needed to convert to the barcode.

eg: {227}|{228}|{229}
This will create a barcode for the values in the fields 227,228 and 229 seperated by "|".

Optionally, you can change the default settings. You should use '$' sign to sepearate parameters.
By default, the barcode string is displayed under the barcode. if you need to avoid that, you may add 0 as your second parameter
eg: {227}|{228}|{229}$0

The complete formula to address all paramers are as follows.

```
barcode string $ display $ width $ height $ format
```

eg: {227}|{228}|{229}$1$2$100$CODE128

- code String - explained above
- display - If you need the barcode string to de displayed below the barcode, set the value to 1. If you need that to be hidden, set the value to 0.
- width. - Sets the width of the barcode. The value should be either 1,2,3,4. The default value is 2.
- height - Sets the height of the barcode. The default value is 100.
- format - Sets the format of the barcode. The possible values are either CODE128, CODE39, EAN, UPC. The default value is CODE128.
  Note : Out of these five parameters you can enter either only first parameter, first two parameters, first three parameters, first four parameters or all parameters.

Form and Fields Settings

Form Classes

Form classess are used to controll the behavior and apperance of the forms. To add a form class. Select the form and click the "Edit Form Details" button.

n3-noNewAtMain	New button will not be available when the form is opened as the main form. No impact for the subform.
n3-quickSubform(x)	The sub-form given by its form ID in 'x' will be opened in the new record mode, when the main-form instance is opened.
n3-openSubform(x)	The sub-form given by its form ID in 'x' will be selected, when the main-form instance is opened.
n3-fullWidth	The form will get the full width of the screen.
n3-formPage(page name)	When a perticular record is selected the form page given by the page name will be appered. eg: n3-formPage(print_inv)
n3-siblings	This is used for view forms and view_list forms if the data should be retreived from the sibling entities of the view form.
n3-singleColumn	Set the form view to a vertical view

n3-singleColumn_200	Set the form view to a vertical view and input width to 200px.
n3-singleColumn_100	Set the form view to a vertical view and input width to 100px.
n3-organised	Usual Group Open and Group Close given in the form is converted to a much organised responsive grid layout.
n3-conditionalSubform(f,=,a)	Use this class for a subform if that subform should be displayed confditionaly based on a field value in the main form. eg:n3-conditionalSubform(455,=,service)
n3-icon	Set the an icon to the form. eg: n3-icon(fa fa-tools w3-green)
n3-headerRight(x)	Put the value of the field x to the right corner of the form header.
n3-headerLeft(x)	Put the value of the field x to the left corner of the form header.

Field Classes

Field classes add styles and/or behaviours to the input field area and/or at the saved record mode.

n3-notEdit	Hide at New Record mode
n3-hideAtNewSub	Hide at New Record mode in sub-forms
n3-hideAtInstance	Hide at instance mode
n3-hideAtEdit	Hide at edit mode
n3-double_length	Space of two fields are acquired for the field/td>
n3-full_length	Space of of the full width is acquired for the field
n3-half_length	Space of of the half width (of the screen) is acquired for the feield

Cover Classes

Cover classes add styles to the input field area includeing its label.

n3-hideAtNew	Hide at New Record mode
n3-hideAtNewSub	Hide at New Record mode in sub-forms
n3-hideAtInstance	Hide at instance mode
n3-hideAtEdit	Hide at edit mode
n3-double_length	Space of two fields are acquired for the field/td>
n3-full_length	Space of of the full width is acquired for the field
n3-half_length	Space of of the half width (of the screen) is acquired for the feield

How To?

Nested Subform

It is possible to have nested subforms. A Subform record can be opend as a main form and subforms can be enterd for that and so on.

How to

Set the form_display_type of the subform to 'expandingSub'. You may use Quick Edit page to do that.

Hidden Subform

It is possible to skip subform tab after the header form. This might be useful when you want to get sum velues in the header form but don't want to dispay the subform there.

How to

Set the form_display_type of the subform to 'hidden_subform'. You may use Quick Edit page to do that.

HTML Page After Form

It is possible to get details from a html page after the form.

How to

Upload the html file to the server. Edit form and give the html file name for the HTML Page. (eg: test.html)

Custom Duplicate Message

Custom message when records are duplicated.

How to

Edit the quick form.

View List Forms

Show the list without tabs.

How to

Set the form type to view_list. Setup is similar to View forms.

Fetch Match Formula inside calculations

Values in other forms can be included in the calculations by fetch match formula.

Same can be used in primary and secondary operations in the fixed value fields (for both updating values and mapping values)

How to

Eg: {"fetch Match formula"}*{79}

Eg: {"25^100^101,78,="}*{79}

Note: Do not forget the double quotes for the fetch match formula.

Skip Calculation

In operations, the calcuations of calculation type fields in the altering form will not take place after an update from this operation. Suitable for simple update operations where further calcualations inthe altering form are not required.

How to

Edit the operation and enter 1 in the Skip Calculation field.

Conditional Formatting

Fields can be appliedly conditional formatting.

How to

Enter the required condition in the condition_format column for the field and enter 'conditional_format' in the coverClasses column. You may use Quick Edit page to do that.

Conditional Formatting Formula

Conditions are seperated by "^".

The fomula for one condition is "logical condition $ value/cover $ style classes to add if true $ style classes to remove if true

Style classes should be seperated by the space if there are more than one style class.

The second parameme value/cover indicates whether the styles should be applied for the value only or for the covering area including the field name.

Eg: {136}=="Planned"$value$w3-light-blue$^{136}=="Released"$value$w3-lime$w3-light-blue^{136}=="Reserved"$value$w3-yellow$w3-lime

{1019}=="Purchased Product"$cover$$w3-hide

Conditional Show/Hide

Fields or groups can be shown conditionally based on values selected on options and radio data types. The same can be appleid for fetch data fields based on the retrieved value.

How to

Enter the required condition in the condition_show column of the options, radio,or fetch field. The condition_show indicates which field or field group to be shown based on the selected or fetched value. if a field group is given, the entire field group will be shown. You may use Quick Edit page to do that.

Eg:Purchased Product$=$1020^Manufactured Product$=$1022

In the above example, if the 'Purchased product' is selected the field (or group) 1020 will be shown and if the 'Manufactured Product' is selected, field (or group) 1022 will be shown

Tip: If you need to conditional hide and show fields in the instance mode, use the cover class "w3-hide" and remove the same using conditional formatting.

Conditionally validating the form before save

Enter conditionally required fields here. If you need some fields to be mandatory based on values given in a field, use this.

field$value$required fields$Error Message
Eg: 23$Manufactured$56,89,90$Manufacturing Details are required for Manufatured parts
You can include more than one conditions by repeating the same as above and seperating them by '^'.

Managing Field width

Using cover classes it is possible to set the width of the fields.

- n3-double_length : Space of two fields are acquired for this field
- n3-full_length : Space of of the full width is acquired for this field
- n3-half_length : Space of of the half width (of the screen) is acquired for this field

How to

Enter the respective cover class for the field

Hide At Modes

Using cover classes it is possible to hide fields in different modes.

- n3-hideAtNew : The field is hidden when a new record is entering.
- n3-hideAtNewSub : The field in the subform is hidden when when a new record is entering.
-
- n3-hideAtInstance: The field is hidden when the record is displayed.
- n3-hideAtEdit : The field is hidden when the record is editing.

How to

Enter the respective cover class for the field

Conditional Formatting in Table

Table

Required data records will be shown in tabular form

Enter the response seperated by commas (eg: 67,68,89)

Response can be included a sorting column, sorting order (ascending or descending), the maximum number of records, and group-by column as well. (Eg: 67,68,89$89$D$50$68)

You can retriev the records of an acsendant entity(It is not necessary to have the access to the acsendant entity to see the records via a table.) To retreive records from the ascendant entity you may add the eneity level as the fifth parameter of the response. eg: 78,79,80$$$$$Site.

Enter the filter criteria.

Filter citeria
Example for filter cirteria : 23=Released,28>0

For the right-hand side values you can use @today, @entity and @user to indicate the current date, entity and the current user.

You can refer to a field value using {x}. ex: 25={27}. Please note that you cannot refer to field values of data types fetch match, fetch calculation and fetch sum in the righthand side of the query

Drilldown Table Values

This is used to drill down data in tables. Detail table is shown underneeth the table row.

Enter the expanding query in Page Designer content. Format is 'response$filter'. (Eg : 148,151,145$146=@1,147=Reserved. Filter is field ID 146 of detail form should be equal to first in the response and the field id 147 is equal to "Reserved"). Also it is possible to use @user, @thisMonth and @today in the filter.

Limitation : Filter is limited to the = at the moment. (This will be improved on request).

Conditional Formatting

Define conditional formatting in the widget_style column

Formatting filed ^ value field ^ value1,classes1$value2,classes2$...

Eg: 25^33^Invoiced,w3-green$Delivered,w3-orange$Released,w3-yellow

Refer to one record from navigation

You can refer to a single record of a a form by a navigator item by providing the necessary filter criteria.

How To: Go to Quick Edit/Navigator. Enter a new record.

Form ID	Enter 0
Form name	Enter the Navigator display name
Frequency	Enter **view_filter**
Data Entry Level	Enter the navigator level.
Form Category	Enter the form category.
Sort	Enter the sorting order.
Page Name	Follow this formmula. **one $ form Id $ filer criteria** eg: one $ 24 $ 405^@today and 407^Current

Value Based User Access

Value-Based Access allows users to view certain records, even if they do not have access to the forms or categories those records belong to. Access is granted based on specific conditions that check the values within each record.

Enter the formula in the row for the correct form in Quick Edid/Forms.

Example:
A user might not have permission to access the "HR Forms" category, but if a record's "Department" field is set to "IT" (and the user is in IT), they may still be allowed to view that record.
View Access
Users with value-based view access to a particular form can view records that meet the defined value-based access conditions.

The access rule follows this formula:

view$EntityLevel$FieldID

In this setup, the value in the specified FieldID must match the name of an entity. If the user has access to that entity, they will be granted view access to the corresponding record.

Example: If FieldID idicates "Department" and a record's Department value is "HR", only users who have access to the "HR" entity will be able to view that record.

multile field ids can be used as well and they are seperated by "||".

view$Project Sites$3952||view$Project Sites$3953

Button Access

Users with value-based button access can see and interact with operational buttons on a form, such as custom actions or workflows. However, they cannot perform actions like Edit, New, or Delete.

The access rule follows this formula:
button$EntityLevel$FieldID

In this setup, the value in the specified FieldID must match the name of an entity. If the user has access to that entity, they will be able to view and click the enabled buttons for that record.

Example: If FieldID indicates "Region" and a record's Region value is "Western", only users with access to the "Western" entity will see and be able to use the operational buttons for that record.

multile field ids can be used as well and they are seperated by "||".

buttons$Project Sites$3952||buttons$Project Sites$3953

Write Access

Users with value-based write access have full access to records that meet the defined access conditions. This includes the ability to view, edit, and delete existing records, as well as create new records when conditions are satisfied.

Since write access also allows users to create new records, the values in those new records may need to be validated against data from another form.

Example: In the Employee Complaint form, a user may be allowed to submit a new complaint record only if the employee mentioned works in a department the user has access to. Since the employee's department is stored in another form (e.g., Employee), the system may use a fetch-match formula instead of referencing a direct FieldID.

Formula Example:
write$Project Sites$168^3952^926,982,=

Formula Breakdown

Forms

any
These are general forms.

view
View form is assentially a subform. This subform simply show records from another form(which we refer to as the view form) filterd as per the values in the parent form. The filter critera which matches the values in the header form to the view form. The fields (form the view form) that are required to be shown in this subform can be selcted.

If you need extend the records in the view form to the siblings of the view form's entity add the form class **n3-siblings**

user_form
User forms are used to retrict the access to the record(s) for the users.

How to create user_form

Create a form of type user_form. Add fields for the form. There are two methods to controll access to a user_form. One method would be to enter a field ID in the view_response column, where that field holds the value of a username (You may use Quick Edit tab for this). In that case, records carrying the current user's username will be accessible. If you leave the same blank, then the record that are created by the user will be shown. User can create only one record of user_form.

View Select
This is a subform. Records are filtered from a form according to the given mapping with the values in the header form same as fo the form type, **view**

The user can selct the required records by clicking a checkbox. then the sums, averages of the specifield field values of the selected rows can be set to the header level fields.

Also it is possible to add data fields to this form itself as usual (limited to count and text data field types). Those fields too will be there per each line and the user can enter velues for them. Simply define new fields for this form (Similar

to fields for 'any' form type). You can specify thier default values by using the calculation formula ferering to view form and perent form field ids. These fields are not saved but can be used to update in the Update string to update altering form.

Enter the **Select String**to update field values in the parent form uppon slecting records by the user.

Select String Formula : **function $ subform field $ header field ^ ....**

function	The agregate function such as sum,average,count
subform field	The subform field id which to get the value to agregate function if the user selcts the record.
header field	The header form field id where the agregate value is set to. This value is not saved (The value is shown in the client only). To save this value an operation should be called.

Enter the **Update String**to specify what should happen to selected records upon saving the header form (in edit mode).

Update String explains what should happen to each instance selected upon saving. It can have functions connected by 'AND': new, replaceThis and addThis.

Update String Formula : **new $ Altering form $ Altering form fields $ View/parent/This form fields $ getParent parameters AND addThis/replaceThis $ field in the view form $ View/parent/This form fields $ should calculate (1 or 0)**

eg: new $ 53 $ 351,352,353,354,356,355,361,360 $ 343,345,344,337,338,525,336,@Reserved $ 123,343^124,345 AND addThis $ 340 $ 525 $ 1

For new functions

new	The function can be 'new'. This indicates entering a new line to altering form.
Altering form	The form for new/update records
Altering form fields	The field ids of the altering form seperated by commas.
View/parent/This form fields	View field ids, parent fields Ids and this field ids can be given seperated by commas. These values are taken to execute the Function to the Altering form. The number of field ids should be equal to the number of field ids in given for Altering form fields and the respective fields should be in the same order. It is possible to give @user for user, @today for the today's date and @constant for the value 'constant'
get parent parameters	If altering form is a subform, these parameters are used to find the parent of the Altering form record. Parent Field ID, View/parent/This form fields ^ .....

For addThis and replaceThis functions

addThis / replaceThis	The function can be
Field in the view form	The field in the view form which will be changed. (either the value will replaced or added to this fied depending on the functions addThis or replaceThis)
View/parent/This form field	View form's, parent form's or this form's field id that the replacing or adding value is taken. It is possible to give @user for user, @today for the today's date and @constant for the value 'constant'
Should calculate	Indicates wheather the form calcualations and its header form calculations should take place upon update. Value is either 1 or 0.

View List

Show the list without tabs.

How to

Set the form type to view_list. Setup is similar to View forms.

Shadow Sub

Shadow Subforms are used to enter new records and edit records of another subform(the base form) of the same parent form. Shadow forms shows lines from a different form(list form) records. Also some of the base form fields are combined to the same line and they are editable. When you edit those fields which are shadowed from the base form ,a new record will be entered automatically to the base form. If there has been a matching record in the base form then, that record will be altered accordingly.

Shadow string

list form $ list fields $ list form filter $ mapping base fields to list fields

eg: 50 $ 336,337,339,340 $ 335,1530,= and 339,@0,> $ 1578,1573,,

list form filter = list field,parent field,logi_op ^ list field, parent field, logi_op

Select string

Selecting matching base records with list records (if exists). Write the list of base fields and respective list fields that are equal to be identical.

base fields $ list fields

1578,1573 $ 336,337

Update String

Update string gives the base form and base form fields that shooud be shown together whith list fields in the shadow form.

base form $ base fields

eg: 45 $ 1356^350,1320

(In this example, base form is 45. show the value of 1356 from the base form from the mapping record if exists(according to the Select string), else the value is from list form field 350. For the next column take the value from the base field 1320 if a mapping record exists, else leave it blank.)

Data Types

Number
Text
Options
Options Search
Fetch Options
Fetch today
Fetch sum
Calculation
Fetch match
Sequence
Fetch Image
Fetch calculation
Date calculation
Date Difference
Date difference(years)
Fetch date difference
User
Inherit from header
Inherit number from header
Sum( )
File attach
Image attach
This entity
Concat
Text area
Today

Explanation

Options

Notes:
If a default value is given, that value will be selected by default, if the same value has been entered for options by the user.

Options can be improved so that the possible values will be filtered upon typing in the input field. To do that enter 'dropmenu' in the addeitional column for the field using the quick edit.

Options Search

This is a searchable List of Values. The search will show the filtered results in a window and user can select one of those values as the input value to the field. This is similar to the options field but with the searchable option and fetching the other values requered from the same reord to this form. Ideal for large list of values.

Enter the correct calculation formula to get values for this field and the other fields from the same record. The calculation formula comprises of 5 parameters. Each parameter is seperated by '$' sign.

form id $ display fields in the list $ hidden fields $ value mapping $ filter conditions

25$108,109,110$155$2=232,3=234,4=243$109={183},48!=0

form id - reference form

display fields in the list - Display fields to be shown in the LoV seperated by commas. The first field value will be assigned to this field.

hidden fields - These fields will not be displayed in the LoV but can be used to fetch values from the same record to the other fields.

value mapping - take the fields in order of both display fields and hidden fields and mention to which field of this form those values should be assigned to.

filter criteria - This indicates how the list should be filtered.

Entity of the source - By default, acesndant or same level entity which the form belongs to will be taken. Optionally, it is possible to get values from all entities or siblings of the current entity. Specify "allEntities" or "siblings" or "children" as the 6th parameter.

25$108,109,110$155$2=232,3=234,4=243$109={183},48!=0$siblings

Note:In the options_search query part, fetch_match fields can be used to filter records

Using SQL queries in Options Search

For advanced queries, you can use sql queries.

Sql select query $ Hidden fields $ value mapping.

Sql select query - Sql select query should include the hidden feild columns as well. Follow the syntaxes under query value data type

Hidden field Ids - Same as in the general formula for options search

value mapping - Same as in the generala formula. The order is taken from the sql select query (also for hidden fields).

Fetch today

Fetch today data type fetches the today's date. Every time the record is reffered, the currnent date is given.

in particular, Fetch today data type is usefull in validations (for example, in conditional formatting) such as to identify what are the records that are overdue, what are the records that are due today,..etc.

Fetch Sum Data Type

Fetch Sum data type is used to calcluate agreagate value of a field in anywhere in the application. You can use one of the the following functions for a field with the Fetch Sum data type.

- sum
- average
- count

- max
- min
- sum if
- count If

You can decide the criteria for selecting records to consider. For example, you can take the average of age field values in form 'Employee' where the country field value is equal to the fixed value 'Sri Lanka' or it is equal to the value entered in the country field in the current form.

Notes:

- If you are using more than one fetch sum fields from the same form for the current form, make sure all fetch sum fiels from the same form have the same mapping conditions. If you need to have different queries for foetch sum fields taken from the same form please use 'sum if' and 'count if' conditions while using what is common for all as the mapping condition.

Calculation

{x}+{y}
* Use leter 't' to indicate that the value is text. (eg : {tx}).

If you want to get a value from another form for calcualations, instead of a field id, you can give a fetch_match formula inside the curly rackets.

{"25^356^108,200,="}*{340}

Needlu Calculation Functions

- year({date})
- month({date})
- day({date}) - Returns the day (number from 1 to 31) of the {date}
- age({date}) - Returns number of days to today from {date}.
- financialYear({date}) - Returns the financial year starting from 04/01 in the format "23-24".

Fetch_Match

This retreives the value from a different form. The following syntax should be followed to enter Fetch_Match data type. Reference form, reference field ID and mapping criteria is included in the syntax.

This value is not saved for the existing form. Instead, it always retreives the value from the reference form. The benefit is, the edotor don't have to worry about changes in the reference field as this form retrieves the existing value from the reference field.

syntax : fm^fd^rf1,tf1,lo1 and rf2,tf2,lo2 ^ Entity Level Type
fm-form number of which value is required.

fd- field number of which value is required.

rf1 - reference field of mapping.

tf1 - this field ID of mapping. Instead of a field ID, if you need to give a constant, enter @constat (eg: @planned). Also note that you may use @entity, @user and @today for this entity , this user and today's date respectively.

lo - Logival operation (=, <,>, <=,>=).

Entity Level Type - Entity Level Type can get two values: **ascendant** or **siblings**

- ascendant - Ascendant entities or this entity.
- siblings - Sibling entities (or this).

Notes:

- The same syntax is used for operation values as well.
- The values cannot be inherited to line level.

Sequence

In general, sequence data types are used to generate a system defined automatically increasing sequencing text for each record of the same form. You can provide prefix, number of digits, and filling character for precedents (for example, 0), and the starting number.

There are two types of Sequences depending on how the value is increased when there are multiple entities in the same level. Entity specific sequence will increase per entity while general sequence will increase irrespective of the entity.

For entity specific sequence type, you can either define prefix, number of digits, suffix and the next number separately for each entity or else you can leave it to be the general format for all entities (You must define the general sequence format).

How To:
To make a sequence data type entity specific, you may set changer column value to “entity” in the Quick Edit/Fields.

Fetch Image

These data types fetch an image attached via image_attach data type in a different form. The Logic to identify the respective image should be given using a matching statement (Same syntax as in the fetch_match data types.)

syntax : fm^fd^rf1,tf1,lo1 and rf2,tf2,lo2
fm-form number of which value is required.

fd- field number of which value is required.

rf1 - reference field of mapping.

tf1 - this field ID of mapping.

lo - Logival operation (=, <,>, <=,>=).

Notes:

- The same syntax is used for operation values as well.
- The values cannot be inherited to line level.

Fetch calculation

The fetch calculation is the data type of the result of a calculation. This value is not saved for the existing form. Instead, it always retreives the value from the references. It can affect the performance of the application.

Here, the fields used to calculation should be one of the type fetch sum or fetch match.

Date calculation

The Date Calculation function add or subtract numbers of days from a date. This should be given in the calculation formula.

{date field id} + or – Num of Days or {number field id}

{date field id} + (Number)

{date field id} + (-Number)

eg: (2023-01-10) + (10)

2023-01-20

Date Difference

This Calculate the difference between two dates. This should be given in the calculation formula.

{first date field id}-{second date field id}

eg:(2023-01-30) – (2023-01-20)

10 days

Fetch date difference

This gives the number of from the second date to first date. This should be given in the calculation formula.

{first date field id}-{second date field id}

Here, the two fields used to calculation should be of the types Date, Today, Fetch today or Date calcaulation.

User

This data type can be used to show the logging user.

Inherit from header

it is possible to inherit fields for the sub form from the parent form.

Sum ()

The sum() function calculates the sum of a set of values in sub form.

Notes:
You can use one of the the following functions for a field with the Fetch Sum data type.

- sum
- count
- max
- min
- sum if
- count If

File attach

Any type of file of can be attached here.

Image attach

This can be used to upload an image.

This entity

This data type can be used to show the logging entity.

Concat

the concat function adds two or more values together.

syntax: {fd1}, “ ” ,{fd2},“ ” ,{fd3}
eg: {fd1} = NEEDLU

{fd2} = School

{fd3} = 2023

Result: NEDDLU School 2023

Text area

If we need to write large text, such as addresses, we can use this.

Today

Today data type retrieves current date from a database.

Operations

Operations Overview
You have learnt how to do calculations and retrieve values based on entered values by the user using variouse data types. On the other hand, you might need need to update and/or enter new records upon user action. In that case, you can use NEEDLU operations. This overview explains how NEEDLU operations work.

Operations are called via Operation Group. An Operation Group can be called following these user actions.

A button click
Selecting a menu option
New record save
Editing a record
Deleting a record
Operation Group is a sequence of operations carried out in the given order. An operation necessarily consists of Primary operation and optionally consists of one or more secondary operations.

Operation

Primary Operation

An operation has the following components

This Form

The form which operation is created.

Altering Form

This Form which values will be altered or displayed.

Entity

The eneity which the altering form belong to when carrieng out the operation. It can be either a fixed entity, same entity or an ascendant level entity.

Mapping conditions

The conditions which explains the how to find out the instance(s) that the values shoud be updated.

Impact

What should be done to the mapping instnces of the altering form.

Operation group

Operation Group

Operations are groups together to execute them. Even if you need to execute one operation, you have to make an operation group of that operation. Operation Groups can be executed in following manner.

1. On submit
2. Menu option
3. Button
4. Link Button
5. Function Button

How to

On submit, Menu Option, Button
Prerequisites : You must have entered one or more operations except for Link Button operation groups.

1. Select the form and go visit Operations tab.
2. Make sure you have entered operations (not required for Link Buttton).
3. Click Add Operation Group button.
4. Enter following values
    1. Group Name - This will be shown in the system as a menu option or button.
    2. List of Operations - Enter the operation IDs seperated by commas. Operations will be executed in the order you enter them.

Link Button

You can create a button in a form which upon click a new tab is opened for a given url.

How to

Enter a Group Operation with the display type link_button. Enter the link in the List of Operations. You can refer to field values in the form by including a calcuation formula in the url between <cal> and <cal/> tags. For example you can have a link button which open whatsApp chat with the mobile number stored in the field ID, 123 by; *https://wa.me/<cal>{123}</cal>?text=Hi, Please use this link to register. http://localhost/NEEDLU-2.0/register_form.html*

Function Button

You can associate a javascript function to be executed when the button is clicked. Enter the functin name at the Operations/Link/Function field. No brackets should be entered.

Secondary Operation

When an (primary) operation is added, the criteria to select instances of the updated form to updated is given. If you want to execute operations to update the same instances, it is possible to use secondary operations.

Secondary operations updates the same instaces as the primary operations. You can select update values to be either same as the update value in the primary operation or another value from the source form or a fixed value.

There are four operation types for the secondary operation

1. Add
2. Substract
3. Replace
4. Group
5. New Record

New Record

New Record operation under the secondary operation is different to that of primary operations. In this case, you can source values from both source form and destination form in the primary operation. If the primary operation is add_until, sustract_until or replace_until operation then update value in the destination can also be used as a source, if required. Follow the explain formula and enter yours in the source value field for New Record secondary operarions.

New Form $ Source Form Fields $ Destination Form Fields $ General Values $ New Form Fields $ New Form's Parent

New Form - New Form ID

Source Form Fields - Field Ids of the source form, from which values are required to be sourced, seperated by commas.

Destination Form Fields - Fields of the destination form, from which values are required to be sourced, seperated by commas

General Values - Here you may specify either none, one or more values "updval,user, today" to be entered in the new record. updval means the value that update the destination form in the primary operartion. This might be usfull when the primary operation is one of the add_until, sustract_until or replace_until operation.

New Form Fields - Updating field Ids in the new form in order according the given fields in the Source Form Fields, Destination Form Fields and General values

New Form's Parent - Form ID of the new form's parent form. If new the form is not a subform, enter the value 0.

Eg: 53$229,328,231$337,338,$updval$351,352,353,356,354,355$0

Skip operation

Menu Operation

Operation on submit

Sub-operation

Sub-operations are used to excute operations in the child form from an operation call from the header form. Sub-operations are executed over all the subinstances of the instace of the header form.

To include Sub-operations, simply include operations of the subforms in the operation string but sepearate them bu '-' instead of ',' (comma).

ex : 2,3-10-11-12,4,5

In this example, 10,11,12 can be of subforms. If they are operations from subforms, then the operations will be executed for all the respective subinstances of the header instance. If they are operations of the form which operations are called, then they will be executed as without any difference.

Authorization Category

Authorization categories are defined per operation group.

The operation group can be executed only by the users who are belonged to that authorization categary.

To set the authorization category for an operation group:

1. Select the form
2. Select the edit in the operation group
3. Type the authorization category in the Authorization Category field
   Note: Users for the operation group should be granted by the admin or organizer.

Operation Types

Conditional Update

1. add
2. substract
3. replace
4. replace many
5. add or insert
6. queryNew
7. batch queryNew
8. add until
9. substract until
10. replace until
11. new until

New Record

* If the new record is a duplication of keys, then no new rocord will be created. User will not be informed about that and rest of the opeations in the sequence will be carried out.

Setting values in operation

syntax : fm^fd^rf1,tf1,lo1 and rf2,tf2,lo2
fm-form number of which value is required.

fd- field number of which value is required.

rf1 - reference field of mapping.

tf1 - this field ID of mapping.

lo - Logical operation (=, <,>, <=,>=).

Study and write what are suboperations. Look at the method executeOperations_menu in common_functions.php.

Replace Many

Update (replace) many values in records that satisfy the filer conditions

`target Field Ids $ source field Ids` **Example**
26,27,28$56,57,58

Visit

Visits an (different or the same) existing form instace which is mapped by the given mapping.

Open

Opens a (different or the same) new form with the given initial values.

Group

A group operation let the system to execute an operation group set for a different form (reference form). Instances from the reference form are filtered by the given mapping and the given operation group is applied for those instances.

Group operations can be used as seconday operations as well. In that case, group operations consider the each resultant instance of the primary operation as its source instance and mapping for the secondary group operation is done taking the values from the resultant instance(s) of the primary operation.

queryNew

Creates multiple lines in a form (destination form) taking the values from this form and a third form. In this operation, three forms are involved.

This form - The form which the operation is created.

Destination form - The form which new records are entered to

Query form - A third form where multiple records are selected to create new records in the destination form.

The records from the query form are selected based on the a given filter criteria using values in This form. A new record is created in the destination form per each filtered record in the query form.

First create a queryNew operation. Provide the mapping values from this form to destination in the value mapping area. Then Click edit and enter the formula to filter records from query form and field values to be taken from the filtered records to the new records in destination form. The formula should be entered in the Fixed Value field.

syntax : qfm^qf1,tf1,lo1 $ qf2,tf2,lo2^qv1,dv1$qv2,dv2 or qfm@siblings^qf1,tf1,lo1 $ qf2,tf2,lo2^qv1,dv1$qv2,dv2
qfm-form number of the query form (third form). if query should be expanded to sibling entities then use 'qfm@siblings'

qf1- field number of the query form.

tf1- field number of this form

lo1- logical operation (=, <,>, <=,>=)

qv1 - field in the query form to get value.

dv1 - field number of the destination form where qv1 is taken as value.

If the record number should be shown in order as a value in each new record in the destination form, you can use @iteration. (eg:25^345,460,=^@itearion,461$350,462)

If a constant is required in the query form mapping criteria use the constant after the '@' sign. (qfm^345,@0,>^qv1,dv1$qv2,dv2)

If new records should be created for all the records in the query form, then enter "ALL" instead of query form mapping criteria. (qfm^ALL^qv1,dv1$qv2,dv2)

Batch QueryNew

Batch QueryNew should be used instead of QueryNew if a lot of new lines are expected. Batch QueryNew is more efficient in peformance-vise compared to querynew. However, Batch QueryNew:

- Does not execute calculations in the destination. Instead calculations should be handled in the query itself.
- Does not support the secondary operations.
- If exists, Parent instance is unique for the all new instances created by the operation. (In QueryNew, it is possible to have different parent instances)
  If the there aren't any requirement for secondary operations and the parent instance is unique for the new records, then Batch QueryNew is a better option than queryNew as its performance is much better than QueryNew.

```
79^475,2771,=^690,477,476,2343 ^ c477*2^480,481,692,2344,2239^define^80^483,483,=

79^475,2771,=^c690,c477,c476,c2343,c477*2,c358 FROM t79 LEFT JOIN t25 ON c108=c476 AND t25.entity=@formEntity(25)^sql^480,481,692,2344,2239,499^define^80^483,483,=
```

new until

Creates new operations untill certain number of iteration. The number of iterations is given by the value in the source field.

The values for the new record is are given in the conditions sections as in the case of New Record Operations. The value for a field int the new reocord can be either of the following.

1. Value from a source form field
2. Fixed value
3. Calculation formula
4. Iteration Number in the until loop (Set the fixed value to '@iteration').

5. Values in the previouse line - formula = @prev$formula using the values in the previouse line $ formula for the first line using the source form fields.

Secondary Operation

When an (primary) operation is added, the criteria to select instances of the updated form to updated is given. If you want to execute operations to update the same instances, it is possible to use secondary operations.

Secondary operations updates the same instaces as the primary operations. You can select update values to be either same as the update value in the primary operation or another value from the source form or a fixed value.

There are four operation types for the secondary operation

1. Add
2. Substract
3. Replace
4. Group
5. New Record

New Record

New Record operation under the secondary operation is different to that of primary operations. In this case, you can source values from both source form and destination form in the primary operation. If the primary operation is add_until, sustract_until or replace_until operation then update value in the destination can also be used as a source, if required. Follow the explain formula and enter yours in the source value field for New Record secondary operarions.

New Form $ Source Form Fields $ Destination Form Fields $ General Values $ New Form Fields $ New Form's Parent

New Form - New Form ID

Source Form Fields - Field Ids of the source form, from which values are required to be sourced, seperated by commas.

Destination Form Fields - Fields of the destination form, from which values are required to be sourced, seperated by commas

General Values - Here you may specify either none, one or more values "updval,user, today" to be entered in the new record. updval means the value that update the destination form in the primary operartion. This might be usfull when the primary operation is one of the add_until, sustract_until or replace_until operation.

New Form Fields - Updating field Ids in the new form in order according the given fields in the Source Form Fields, Destination Form Fields and General values

New Form's Parent - Form ID of the new form's parent form. If new the form is not a subform, enter the value 0.

Eg: 53$229,328,231$337,338,$updval$351,352,353,356,354,355$0

Operation Group

Operations are groups together to execute them. Even if you need to execute one operation, you have to make an operation group of that operation. Operation Groups can be executed in following manner.

1. On submit
2. Menu option
3. Button
4. Link Button
5. Function Button

How to

On submit, Menu Option, Button
Prerequisites : You must have entered one or more operations except for Link Button operation groups.

1. Select the form and go visit Operations tab.
2. Make sure you have entered operations (not required for Link Buttton).
3. Click Add Operation Group button.
4. Enter following values

1. Group Name - This will be shown in the system as a menu option or button.
2. List of Operations - Enter the operation IDs seperated by commas. Operations will be executed in the order you enter them.

Link Button
For Link Button

Link Button
Function Button
You can associate a javascript function to be executed when the button is clicked. Enter the functin name at the Operations/Link/Function field. No brackets should be entered.

Reserve Operation

Map sorting order syntax:
* f1-A,f2-D

* This means sort the mapped values by field f1 by ascending order then by field f2 by descending order.

Link Button

You can create a button in a form which upon click a new tab is opened for a given url.

How to

Enter a Group Operation with the display type link_button. Enter the link in the List of Operations. You can refer to field values in the form by including a calcuation formula in the url between <cal> and <cal/> tags. For example you can have a link button which open whatsApp chat with the mobile number stored in the field ID, 123 by;
https://wa.me/<cal>{123}</cal>?text=Hi, Please use this link to register. http://localhost/NEEDLU-2.0/register_form.html

Page Design