Build Your First Application

TS No-code Platform

Version 01

Table of contents

During the course of this guide we will take you through the steps of building your first application on TS No-code Platform. For the purpose of introducing you to some of the most frequently used Data Type Components and a relatively simple workflow, we will build a simple yet highly useful application for handling employee expenses (Expense Management). First off though, we should like to introduce you to our proprietary development methodology - Praximity™ - that we use with partners and customers alike for most development projects.

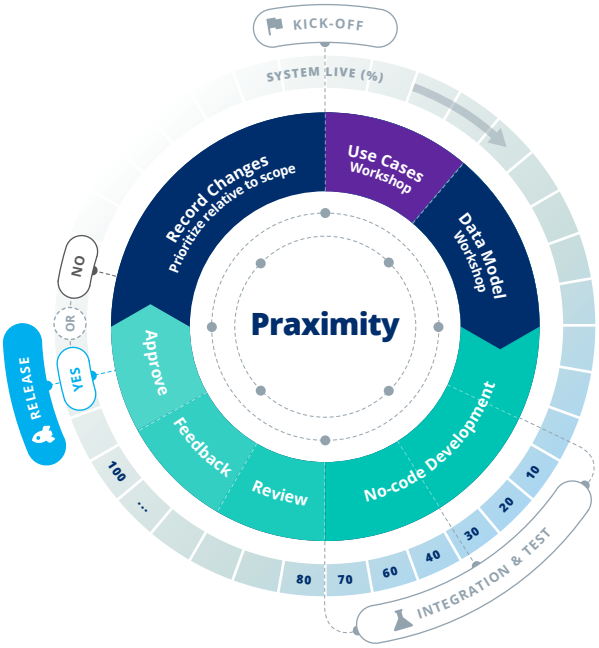
Praximity™

Praximity™ is the name of our agile development methodology. It is a combination of “Practice” and “Proximity”, which describes the physical and conceptual form of collaboration in a development process very close to the business. Praximity does away with previous dogmas such as waterfall and rigid system specifications that are outdated long before the last page is printed.

By utilizing the development speed and elasticity of TS No-code Platform, the Praximity method delivers greater value output for the customer, solid anchoring in the organization, and 8 to 10 times faster time-to-market compared to traditional development.

The Praximity method comprise of five interrelated steps. In the following we will focus on the practical implementation (steps 2 through 5). Step 4 – 5 are repeated for the ongoing adaptation of the application over time:

1) The Team
2) Use Case Workshop
3) Data Model Workshop
4) No-code Development
5) Security & Permissions



The Team

Put together a team of people with relevant domain knowledge and process understanding, preferably people engaged with the process from different perspectives. In this case; Employee (creator), Manager (approver) and Finance (payroll and book keeping).

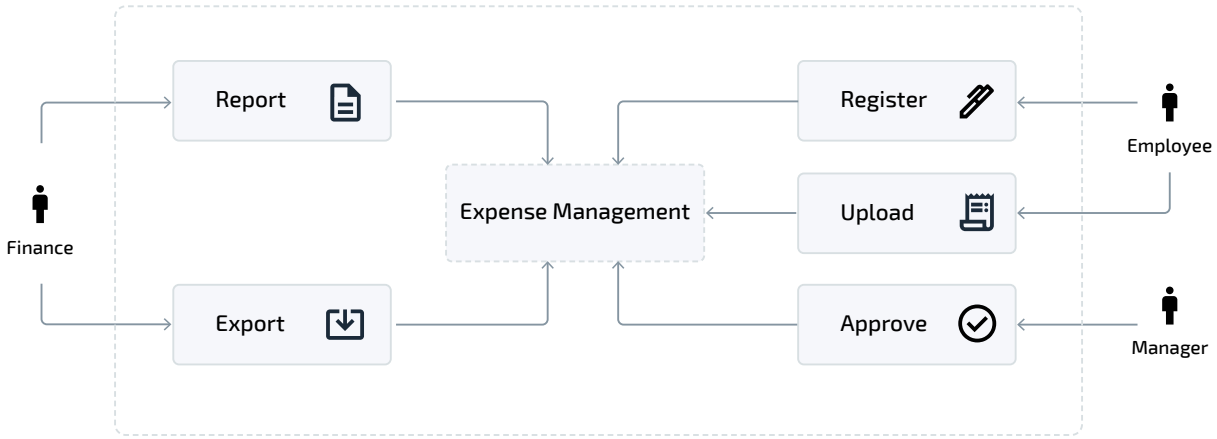
Use Case Workshop

The customer explains what it is they want – Example: We want a system for reporting, approving and handling employee expenses including driving allowance. At the end of each month employees complete an expense report. Creating records for each expense incurred, upload receipts, and submit the report for approval and reimbursemen.

Core functionality

- Recording expenses and uploading receipts (Creator)
- Registration of kilometers driven (Creator)
- Expense report sent to Manager for approval (Approver)
- If rejected, the employee can adjust and resubmit (Creator)
- Upon approval, the report is sent to accounting (Finance)
- Accounting processes salary and post expenses (Finance)

Use Case Diagram



Data Model Workshop

Based on the identified use cases, we know who will use the system and how. With this information ready at hand, we can draw up a basic system diagram with the necessary entities and their connection points.

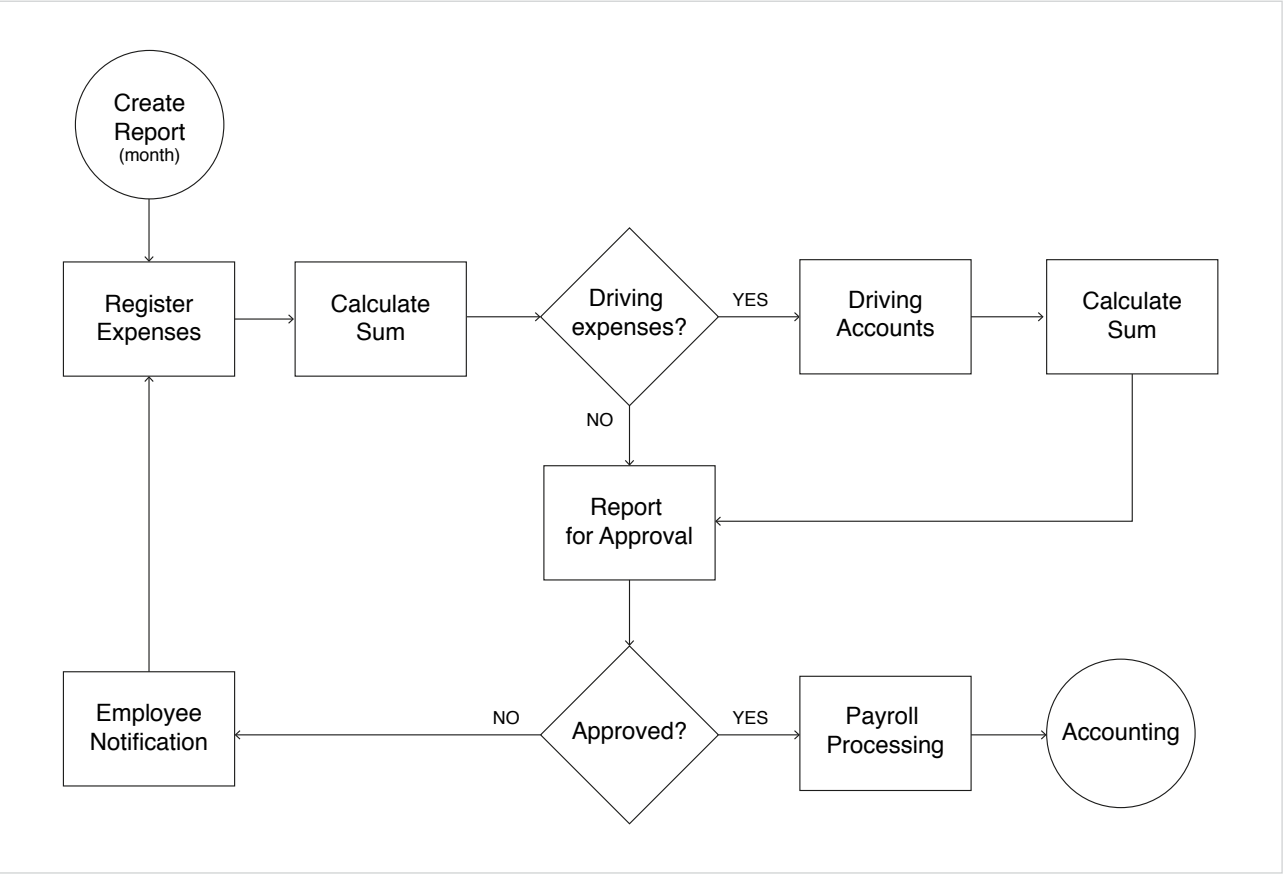


Figure 1: Application Flow Diagram, showing the basic functions of the application.

UML Principals in Brief

If you have experience with data base design, you are probably familiar with UML principles. When building applications on TS No-code Platform, one of the most important things to understand is when to include normalization entities.

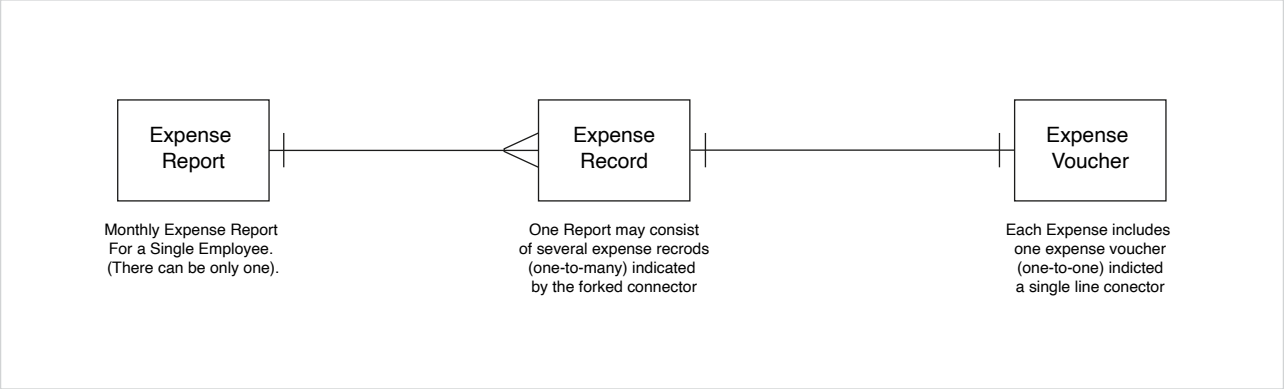


Figure 2: Object Diagram, showing the relational connections also known as parent/child relations.

From the example above (Figure 2) we can see that one expense report may consist of multiple expense records, indicated by the forked connection, and that each record references one expense voucher (an invoice or a receipt).

Building the Data Model

Based on the System Diagram and the use case mapping we are now ready to start building the actual data model on TS No-code Platform, creating the necessary entities, populate them with the relevant Data Type Components (attributes) and set up the relational connections between the entities.

In the example below (Figure 3) we have mapped out three entities “Expense Report”, “Expense” and “Mileage”. The Expense Report entity is a so-called Parent element. This is the actual report that the employee will submit for approval by the end of each month. The Expense entity is a so-called Child element. For each expense the employee will create such an Expense record by filling in information such as Date, Expense type, Amount, etc. and upload a receipt of the expense.

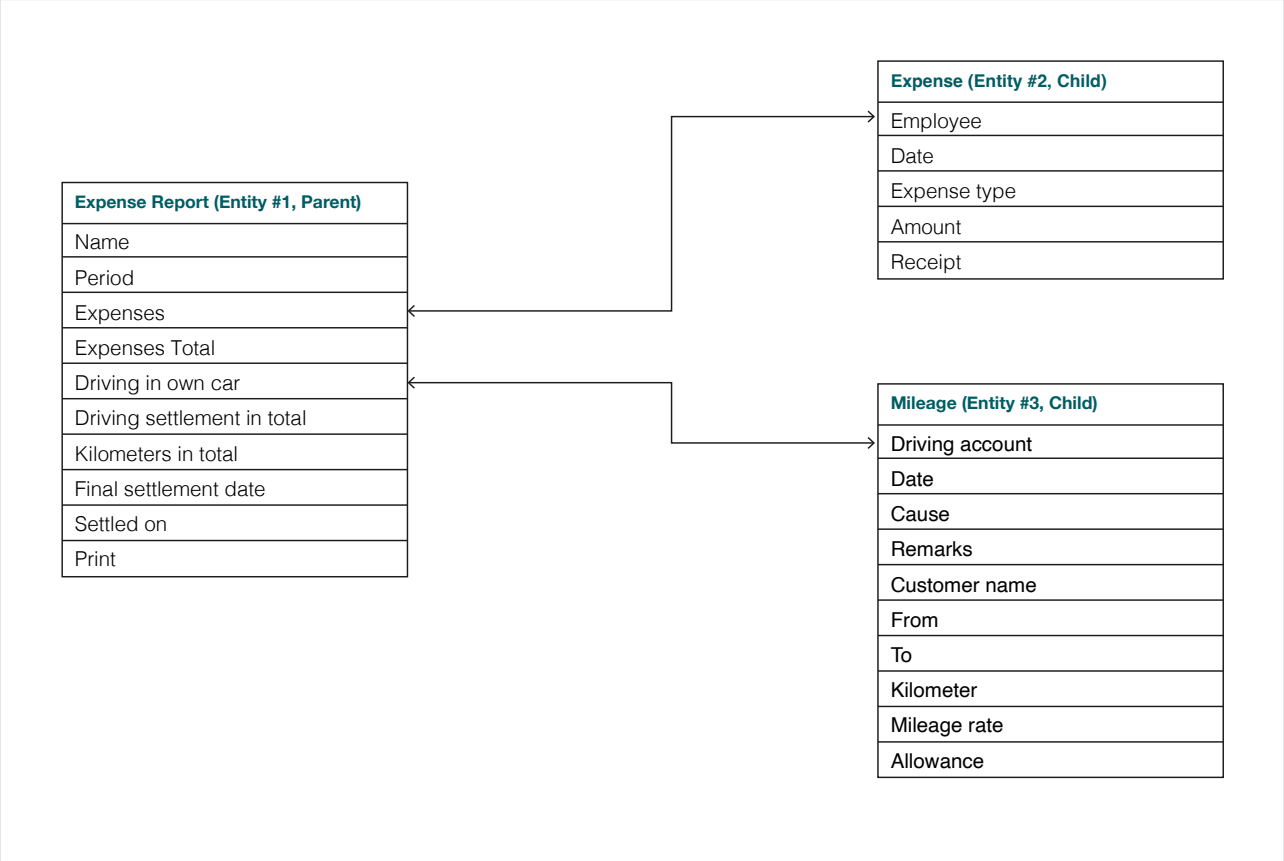
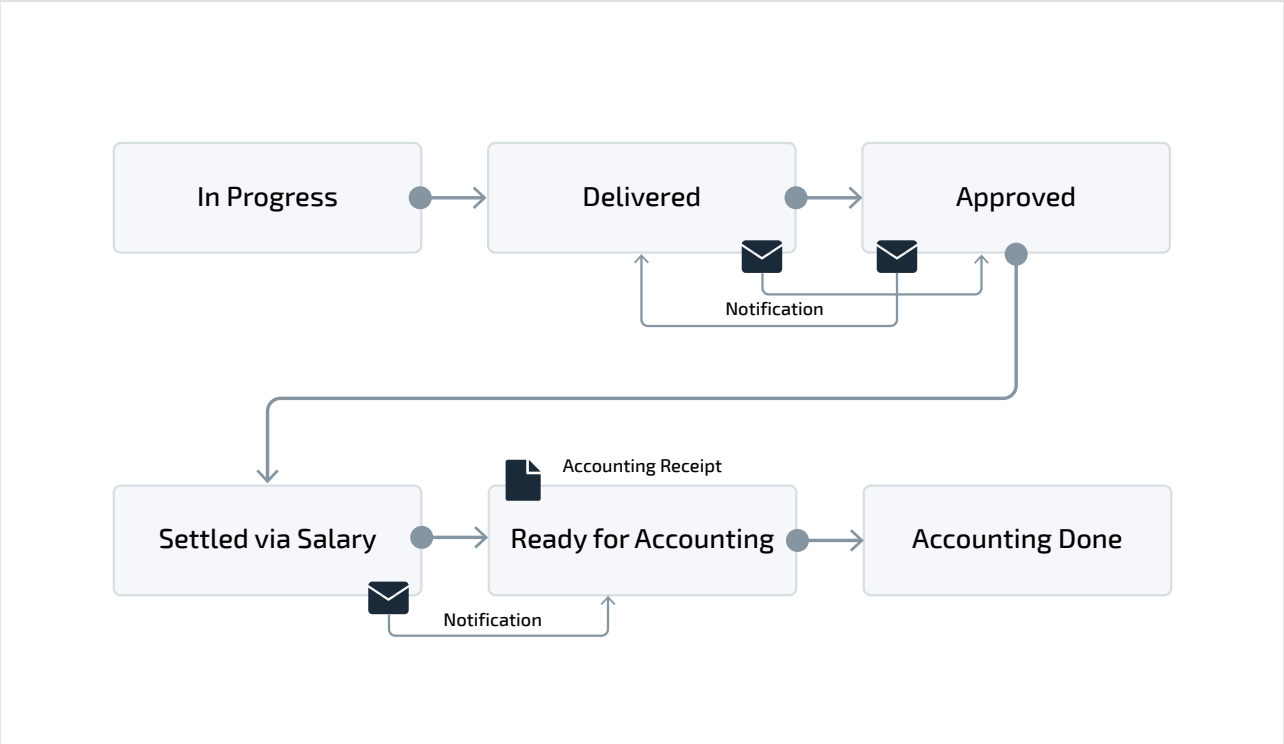


Figure 3: Entities with relevant attributes.

Outlining the Workflow

Before we continue, let's have a quick look at the workflow that we want an Expense Report to go through from creation over processing to completion (see illustration below).



Now that we have a clear idea of the entities involved, the attributes of each of these entities and the workflow of an Expense Report from creation to completion, let's jump right in and begin building the application.

TIP: You can add as many attributes as you need in each entity, but if you are just starting out using TS No-code Platform, we recommend that you stick to those defined in Figure 3 as we will go through the configuration of those in this guide.

Building the Application

To get started. Go to the personal menu in the upper-right corner of your TS Application where it says your name (see example below). Click the dropdown button and click Designer. This will open the backend of TS No-code Platform in a new tab.

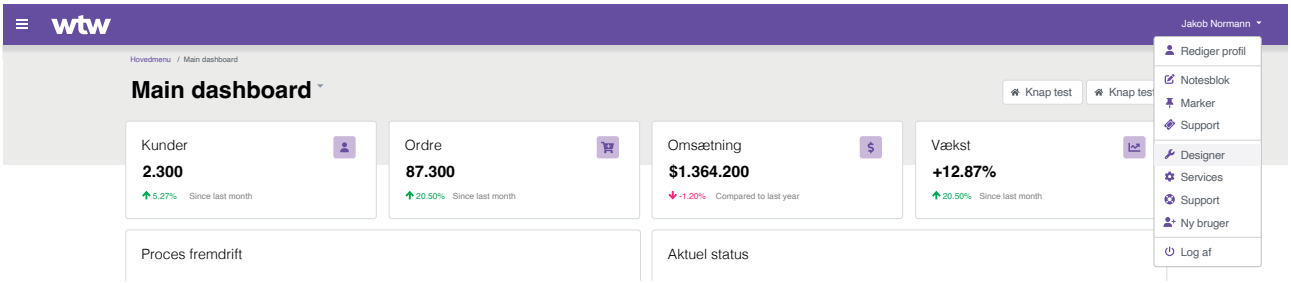
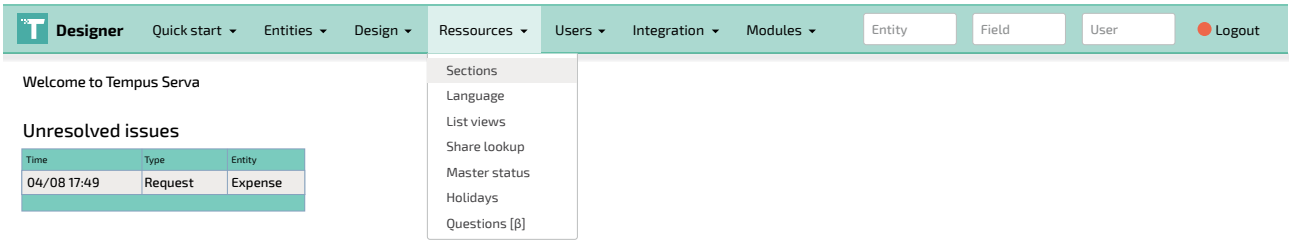


Figure 4: Example of a TS Application (front-end).

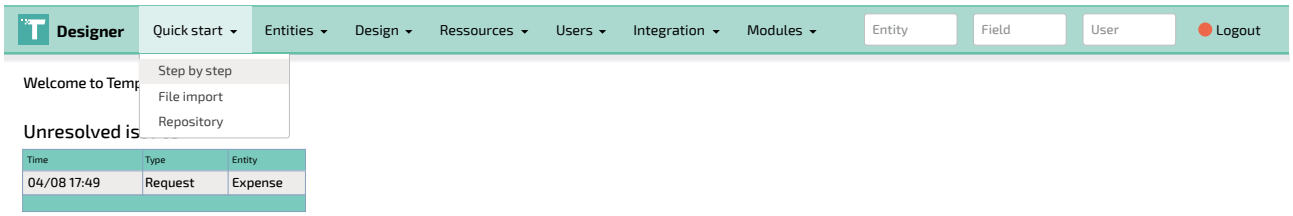
From here navigate to Ressources > Sections. This opens the Solution sections page, where you can see existing sections of the instance (if any) and create new sections. To create a new section for your application, click the Add button right above the Solution sections list (see below).

This opens the Solution Section dialogue. Fill in the Section title “Expense Management” and a Sort Order as shown below, and click Submit to commit the changes and create the new section.



Build the Expense Record entity

Now, navigate to Quick start and press Step by step.



This opens the Step by step dialogue (Step 1 of 5 as shown below). From the Section dropdown list select the section “Expense Management” that you have just created. Now fill in the name of your first entity (“Expense Report”) in the Entity name field.

Next it is time to fill in the name of each attribute (Data Type Component) in the Fields field. Fill in each of the attribute names listed in Figure 3 (table 1), separating them by a line break / return.

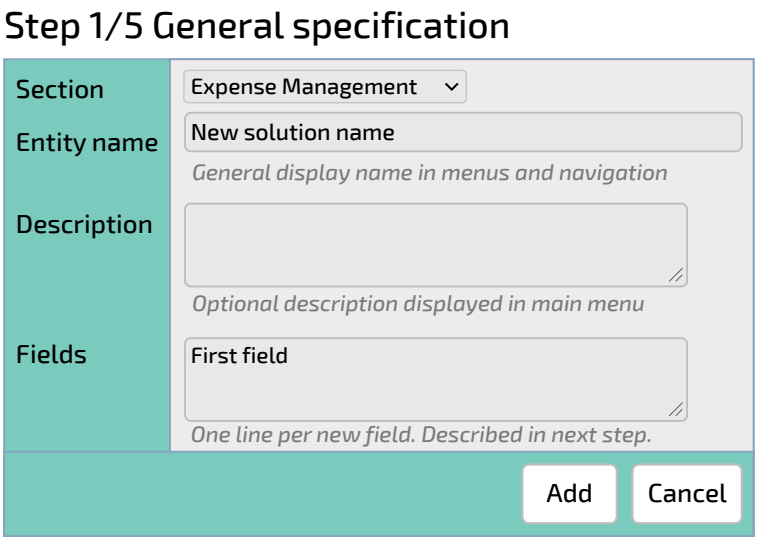


Figure 5: Quick start Step #1 (no attributes defined).

Once you have entered the Entity name and completed the list of attributes it should look like this (see Figure 6). To complete Step 1 click the Add button to commit the changes.

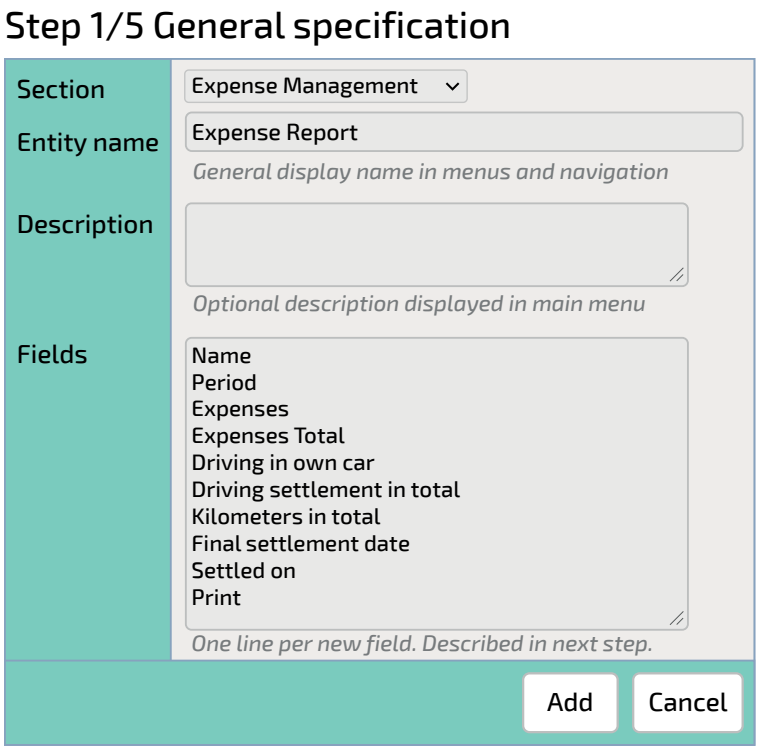


Figure 6: Quick start Step #1 (attributes defined).

This will take you to Step 2 of 5. In this step it is time to select the relevant Data Type Component for each attribute. TS No-code Platform offers a wealth of Data Type Components, allowing you to quickly build almost anything.

All the attributes are by default assigned a “Basic: Text” component. For the purpose of first attribute “Name” this could be fine since it is a single line text field. However, we can do something smarter. We can take advantage of the actual user account information. In the following we will go through each of the attributes and select/assign the most suitable Data type Component for each of them.

Step 2/5 Field details

Field	Type	Helptext	Resume	In List	Validate
Name	Basic: Text		Yes	Yes	Yes
Period	Basic: Text		No	No	No
Expenses	Basic: Text		No	No	No
Expenses Total	Basic: Text		No	No	No
Driving in own car	Basic: Text		No	No	No
Kilometers Total	Basic: Text		No	No	No
Driving Settlement Total	Basic: Text		No	No	No
Final Settlement Date	Basic: Text		No	No	No
Settled on	Basic: Text		No	No	No
Print	Basic: Text		No	No	No

SUBMIT

Figure 7: Quick start Step #2 (data type components not defined).

Name: Since we already know the name of the user from his/her user account we will assign this attribute the Data Type Component “Lookup: User”. Thereby the user’s name is filled in by default.

Period: Since the employees should complete an Expense Report on a monthly basis we will assign this attribute the Data Type Component “Basic: Month”.

Expenses: Since each report may include one or more expense records we will assign the Data Type Component “Parent/child: List of children”. Which will later be configured to create a list of records based on the entity “Expense Record”. We will get back to that later.

Expenses Total: In order to calculate the total sum of expenses incurred we will assign the “Parent/child: Calc children:Sum” component.

Driving in own car: Since each report may include one or more millage records we will assign the Data Type Component “Parent/child: List of children”. This will later be configured to create a list of records based on the entity “Millage Record”.

Kilometers Total: In order to calculate how many kilometers the employee have been driving in his/her own car we will assign the “Parent/child. Calc children:Sum” component.

Driving Settlement Total: In order to calculate the total settlement sum we will assign the “Parent/child: Calc children:Sum” to this attribute for later configuration.

Final Settlement Date: In order for the employee to specify the expected date of settlement we will assign the “Basic: Date” component to this attribute.

Settled on: In order for finance to specify the actual settlement date we will assign the “Basic: Date” component to this attribute.

Print: To allow users to export a template-based report, we will assign an “Action button: Button Export template” to the “Print” attribute. We will get back to the configuration of the action button and how to create a template for export later on.

When you have assigned Data Type Components to each attribute your list should look like this (see Figure 8). At this point you also have the option of defining a helptext for each field, select which attributes should be part of the entity Resume, be displayed “In List” view and/or if the field should be Validated, but we will deal with this later on. Press Submit to commit the changes.

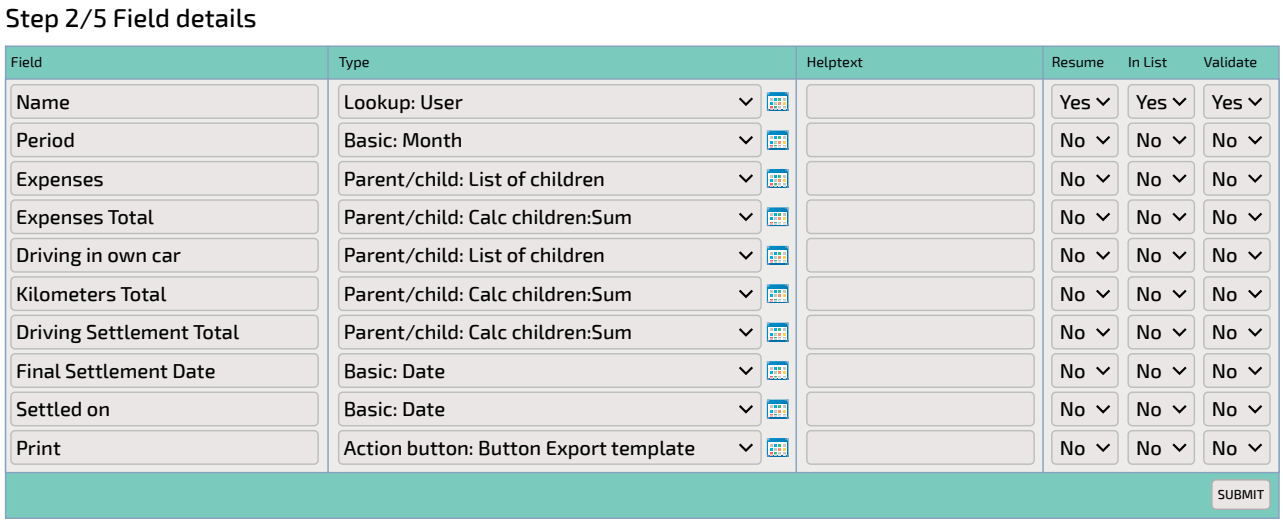


Figure 8: Quick start Step #2 (data type components defined for each attribute).

This will take you to Step 3 of 5 “State and flow”. In this step we will define the workflow (state model) that each record based on this entity must go through from creation to completion. In this case; In Progress > Delivered > Approved > Settled via Salary > Ready for Accounting > Accounting Done. In the dialogue “List of status” field (see Figure 9) enter each state separating them by a line break / return.

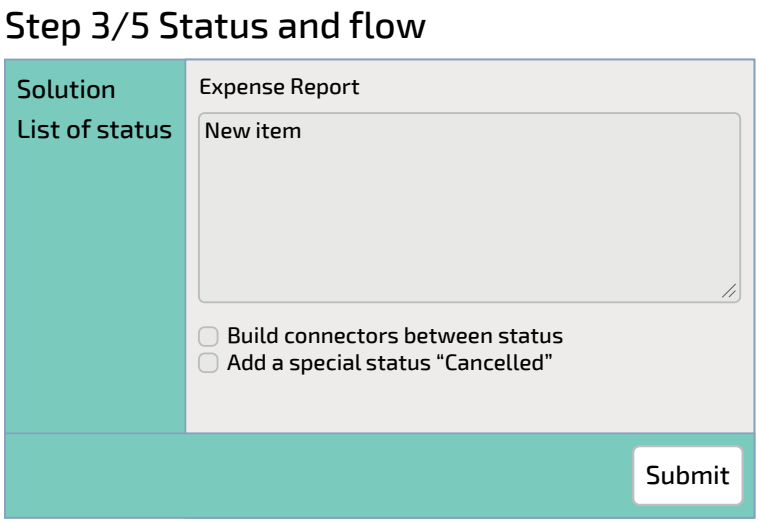


Figure 9: Quick start Step #3 (no states defined).

When you have completed the workflow (state model), check the box Build connectors between status. Now the dialogue should look like this (see Figure 10) and you are ready to proceed. Press Submit to commit the changes.

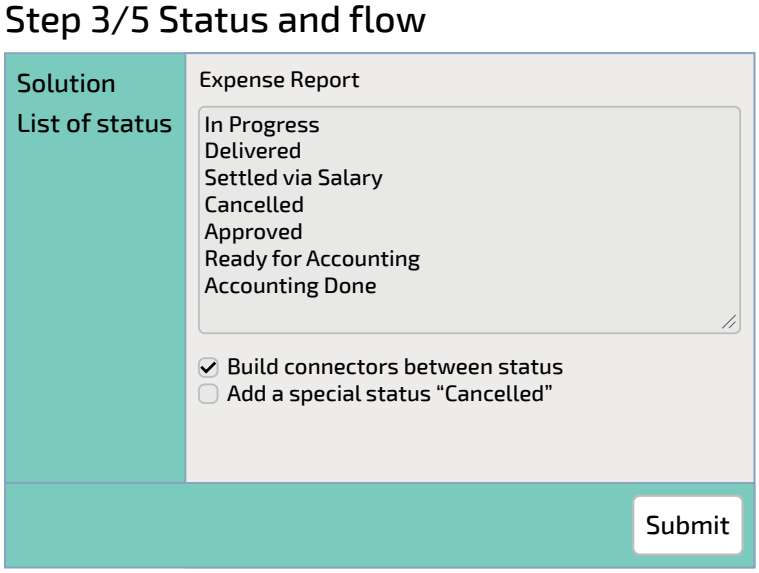


Figure 10: Quick start Step #3 (state model for the entity has been defined).

This will take you to Step 4 of 5 “Permissions” (Figure 11). In this step of the build process we will define the rights or permissions of one or more user groups. For the purpose of this guide we will keep it simple and give “Full permissions” to a single user group. Permissions can easily be changed and refined later on, and you can create your own user groups.

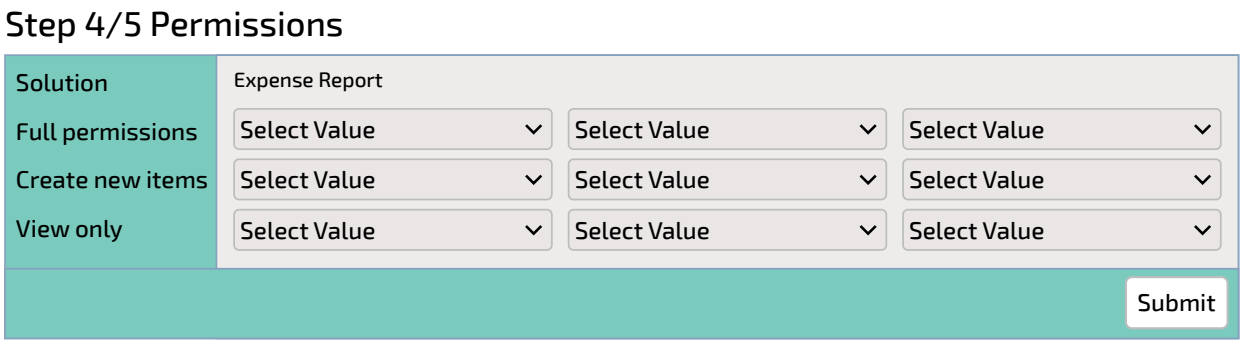


Figure 11: Quick start Step #4 (no permissions have been assigned).

To assign Full permissions to a user group, simply click the first dropdown list next to the Full permission row and select the user group you want to give Full permissions.
In this case we will select the default user group “Internal users”. Now your Permissions matrix should look like this (see Figure 12) and you are ready to proceed. Press Submit to commit the changes.

Note. Select a user group assigned to your user profile. Otherwise, you will not be able to access the entity in the frontend. Alternatively, you can add it by navigating to Users > Edit Users: Open your user profile and add Assigned Group “Internal Users” by clicking [Add], select the group from the dropdown list and press Submit to save the changes.

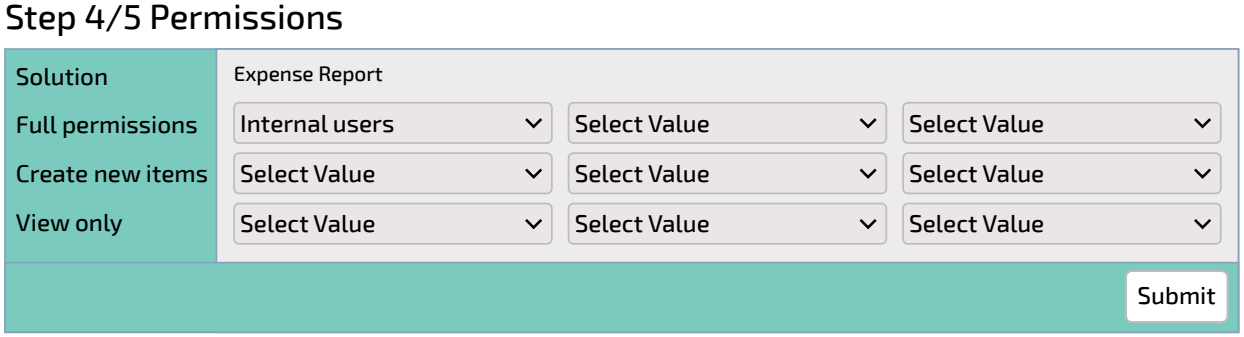


Figure 12: Quick start Step #4 (Full permissions have been assigned to the user group Internal users).

This will take you to Step 5 of 5 “Solution functionality options” (Figure 13). In this step we can setup additional fields e.g. Modified (date) or Modified by (user), Created (date) or Created by (user). For the purpose of this guide, we will not add these fields.

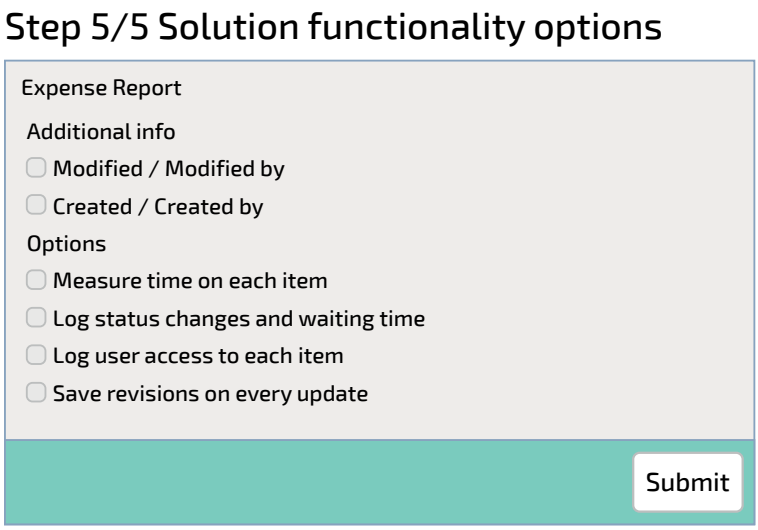


Figure 13: Step #5 (No Solution functionality options have been selected).

We will however setup logging for status change (date/time) and logging of user access to each item. To do this, make sure the respective options are checked. Your Solution functionality options dialogue should look like this (see Figure 14) and you are ready to proceed. Press Submit to commit the changes.

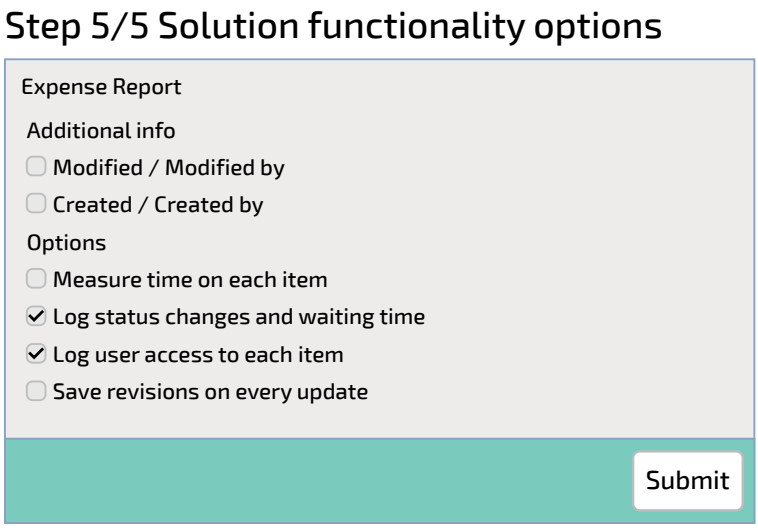


Figure 14: Figure 14: Step #5 (Log status changes and user access have been selected and activated).

This concludes the Quick start guide, and takes you to the Entity page (Figure 15). Here you will see all the fields, entity status and permissions that you have defined in the previous steps. This is where you will finalize field configurations and deploy live. Any future changes or additions will be made here from the Entity page. In this case we can see that the status (S) of several fields indicate that additional configuration is required.

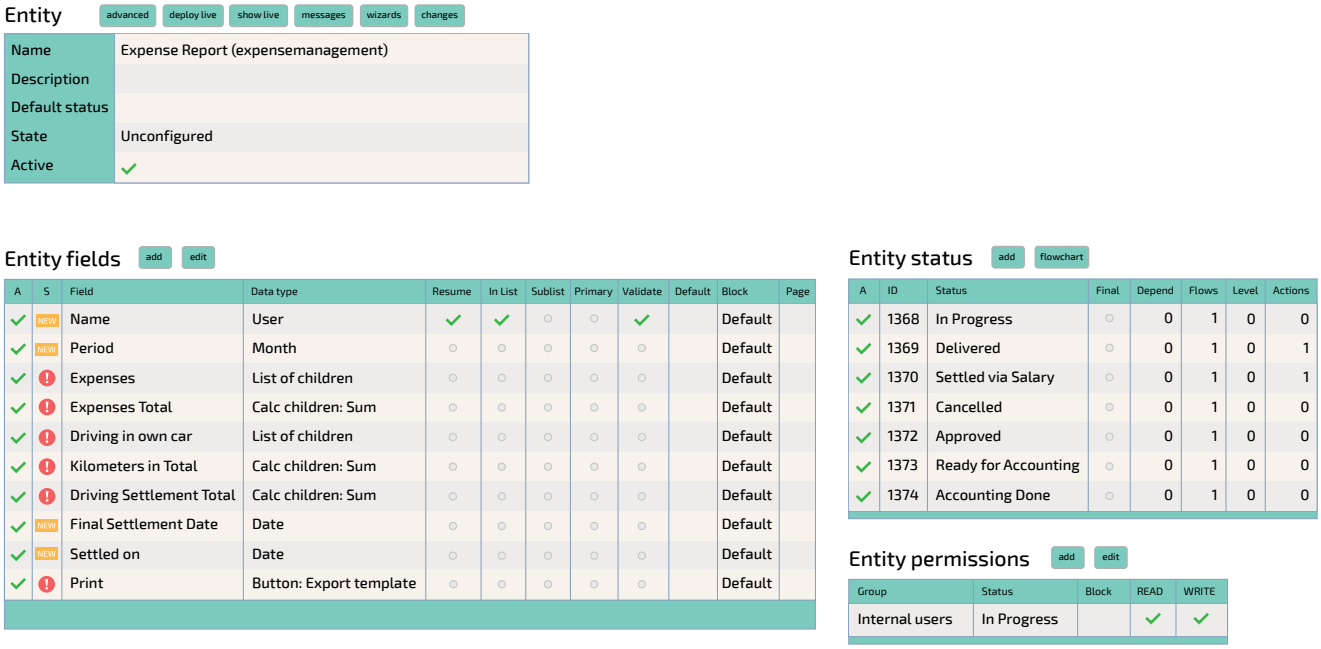


Figure 15: Entity (Expense Report)

Click the Field name “Print” to open the Field dialogue (Figure 16). Here we can see the basic settings for the field Print. To see all settings, toggle the switch in the Field Functions Menu (see Figure 17).

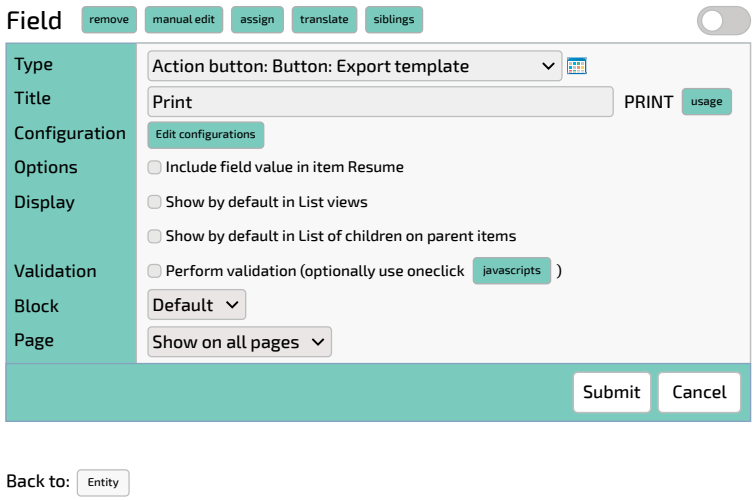
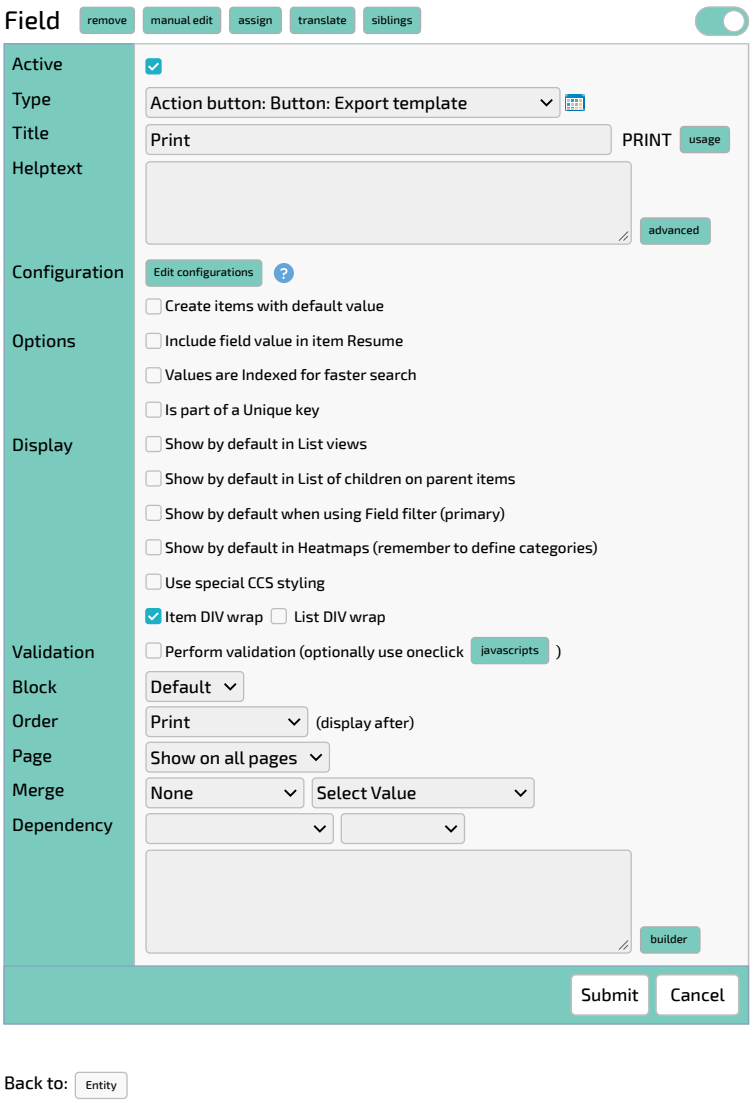


Figure 16: Field dialogue (Print).



The advanced switch has been toggled to reveal all settings options for the field. In this case the field “Print” of the “Expense Report” entity.

Figure 17: Field dialogue (Print), advanced settings.

In order to finalize configuration, click the “Edit configuration” button to open the Edit field configuration dialogue (see Figure 18).

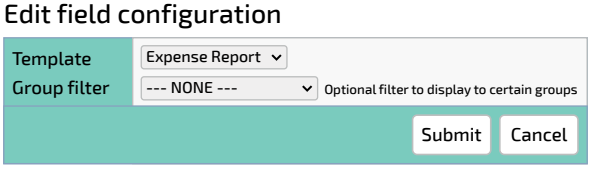


Figure 18: Edit field configuration dialogue).

Here select the Document Template “Expense Report”. Since we have not yet created this template, it will not show up in the list. For information on how to build the Document Template see the section “Creating the Expense Report Template” (page 35) of this guide and refer to the TS Guide “Working with Templates” for more details.

When all field configurations are done click “deploy live”. The Entity page should now look like this (see Figure 19), and you are ready to proceed building the next entity “Expense”.

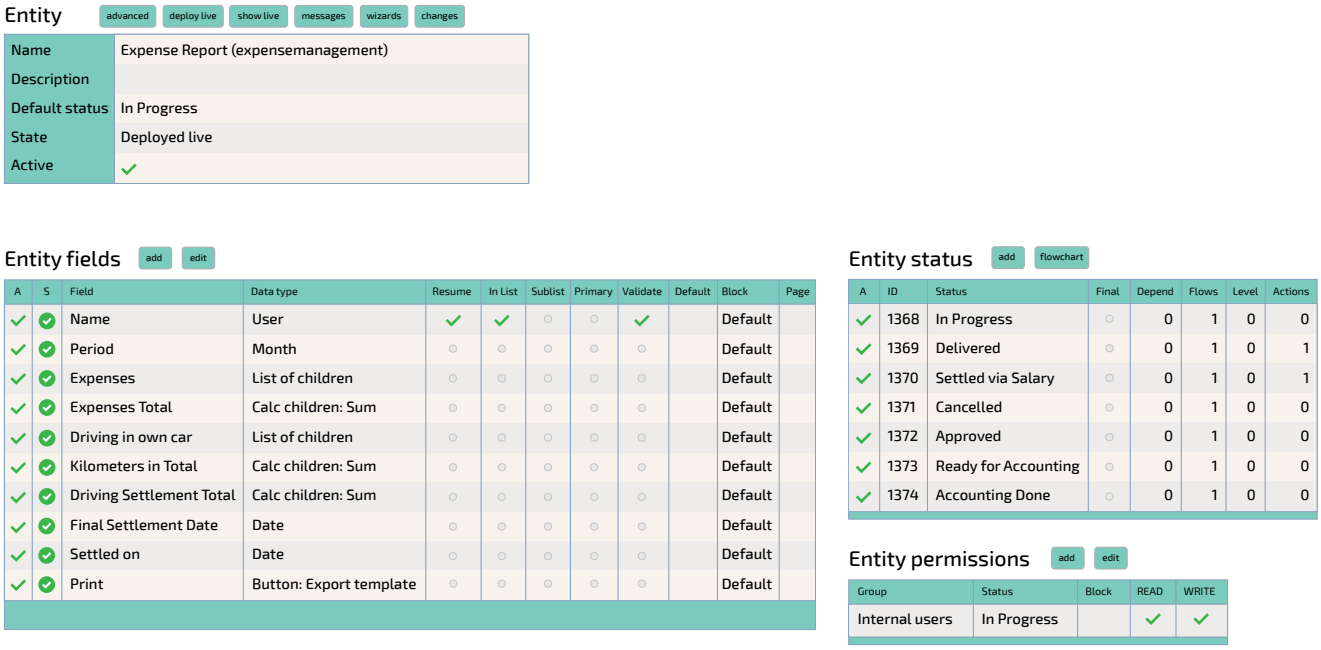


Figure 19: Entity (Expense Report all fields are deployed live).

TIP: You may have noticed that some fields are indicated by a Sync symbol. This indicates that the field, due to changes, needs to be deployed. Click deploy live button in the Entity Functions menu to commit any changes.

Build the Expense entity

Build the next entity “Expense” - a child entity - following the same process: Open the Step-by-step dialogue. From the Section dropdown list select the section “Expense Management” that you created earlier. Now fill in the name of the entity (“Expense”) in the Entity name field. Next it is time to fill in the name of each attribute. Fill in each of the attribute names listed in Figure 3 (Entity #2) on page 7, separating them by a line break / return.

Once you have completed the list of attributes it should look like this (see Figure 20). To commit the changes, click the Add button.

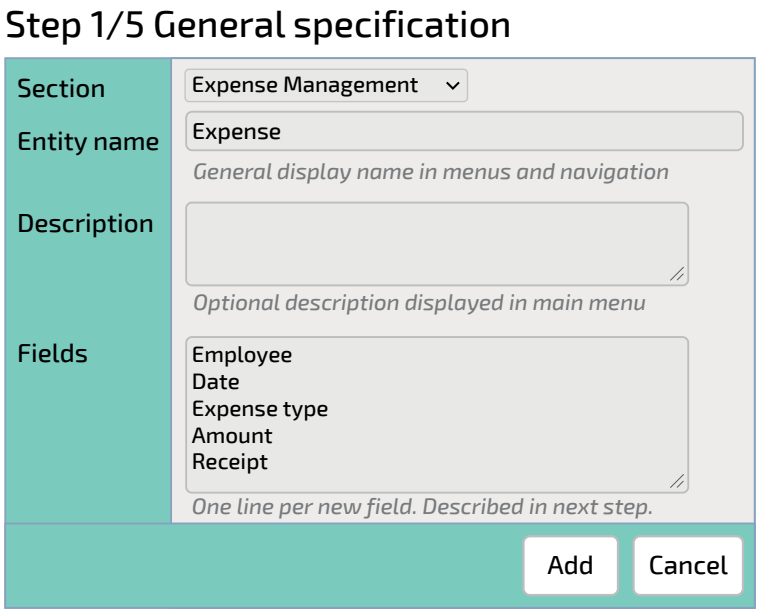


Figure 20: Quick start Step #1 (attributes defined).

This will take you to Step 2 of 5. In this step it is time to select the relevant Data Type Component for each attribute. Just like before all attributes are by default assigned a “Basic: Text” component. In the following we will go through each of the attributes and select/assign the most suitable Data type Component for each of them.

Employee: In order to display the individual expenses on the parent entity “Expense Report” we will use this first attribute as a parent reference, by assigning the Data Type Component “Parent/child: Parent reference”, establishing the relational connection between the two entities.

Date: Since each expense must be registered correctly on the date of purchase, we will assign this attribute the Data Type Component “Basic: Date”. This will result in a date picker for quick and easy registration.

Expense type: In order to categorize each expense correctly we will ask employees to specify the type of expense by selecting from a dropdown list of options (e.g. Small Purchase, Travel Expense, Catering). To accomplish this, we will assign this attribute the Data Type Component “Lookup: Lookup”. Which will be configured to include the relevant expense type options.

Amount: In order register the accurate amount for each expense we will assign the “Basic: Decimal” component for this attribute.

Receipt: Since each expense must be documented with an invoice or receipt, we will create an upload function by assigning the Data Type Component “Files: Documents”. This will result in a drag and drop upload field. When you have assigned Data Type Components to each attribute your list should look like this (see Figure 21). Press Submit to commit the changes.

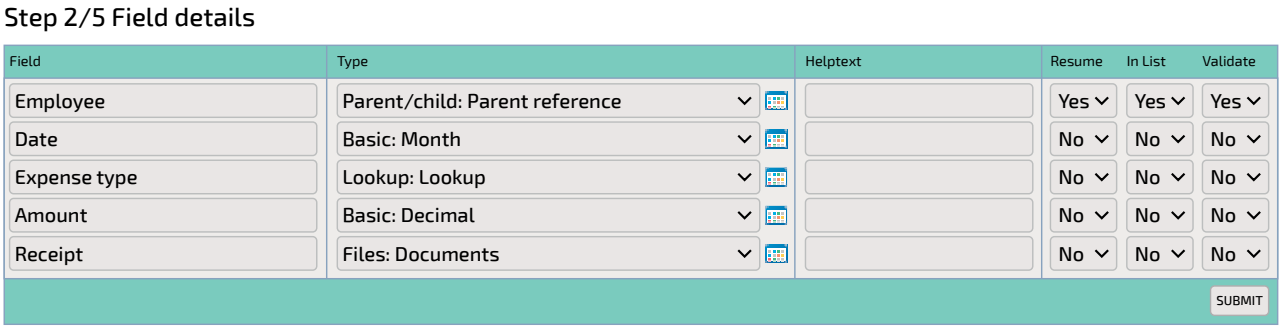


Figure 21: Quick start Step #2 (data type components defined for each attribute).

This will take you to Step 3 of 5 “State and flow”. In this step we will define the workflow (state model) that each “Expense” record will go through from creation to completion. In this case simply; New item, which is the default state name (see Figure 22). Pres Submit to continue.

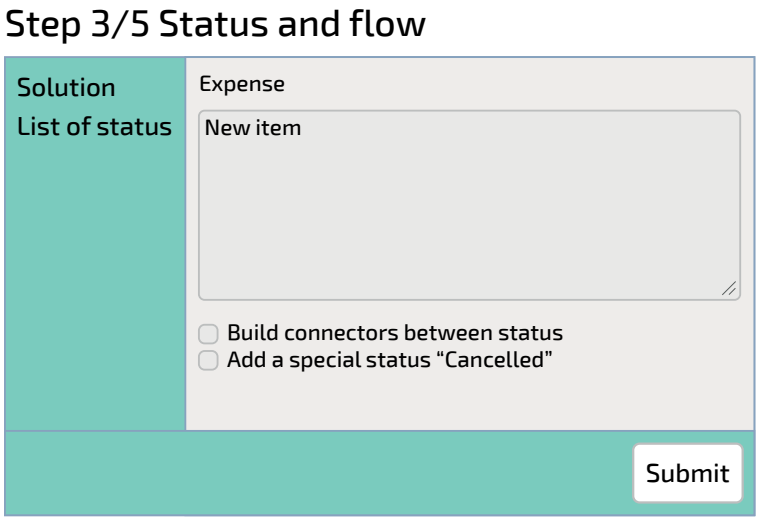


Figure 22: Quick start Step #3 (default single state workflow).

TIP: You can easily setup elaborate workflows, but since an expense is a simple registration created and completed with every detail all at once, all need is a single state. You can name the state whatever you want or use the default “New item” as suggested here.

This will take you to Step 4 of 5 “Permissions”. In this step, we will define the rights or permissions of one or more user groups. In this case we will assign “Full permissions” to the default group “Administrator” and “Create new items” permission to the default group “Internal users” (see Figure 23). Permissions can easily be changed and refined later on and you can create your own user groups. Press Submit to continue.

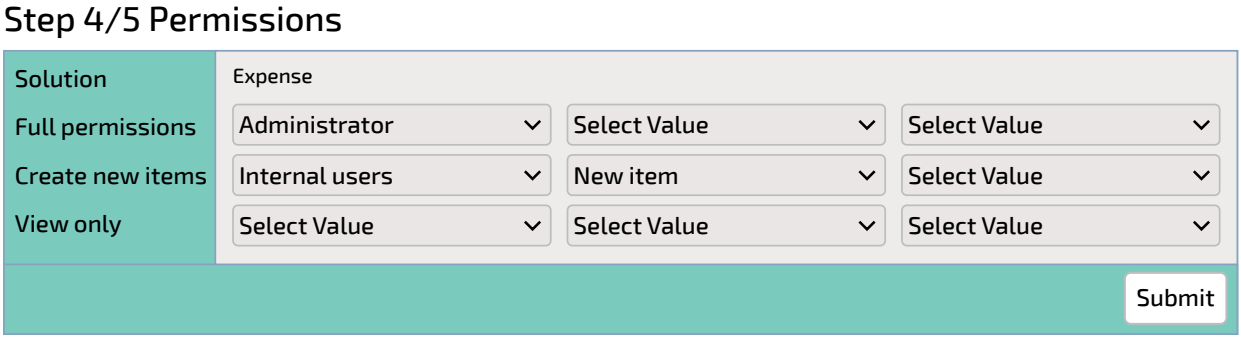


Figure 23: Quick start Step #4 (permissions assigned for the groups Administrator and Internal users).

This will take you to Step 5 of 5 “Solution functionality options”. In this step we will setup additional fields e.g. Modified (date) or Modified by (user), Created (date) or Created by (user). In this case we do not need any additional functionality except for access logging (see Figure 24). Press Submit to continue.

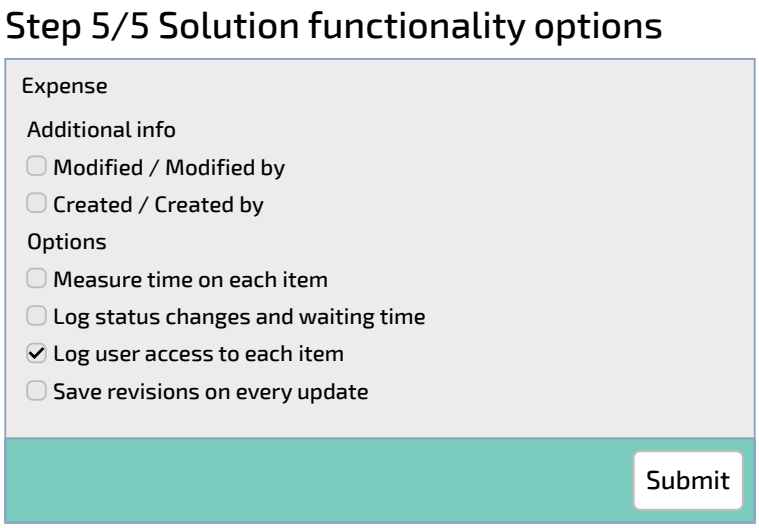


Figure 24: Quick start Step #5 (only access logging selected).

This concludes the Quick start guide and takes you to the Entity page (Figure 25). Here you will see the fields, entity status and permissions that you have defined in the previous steps. This is where you will finalize field configurations and deploy live. Any future changes or additions will be made here from the Entity page.

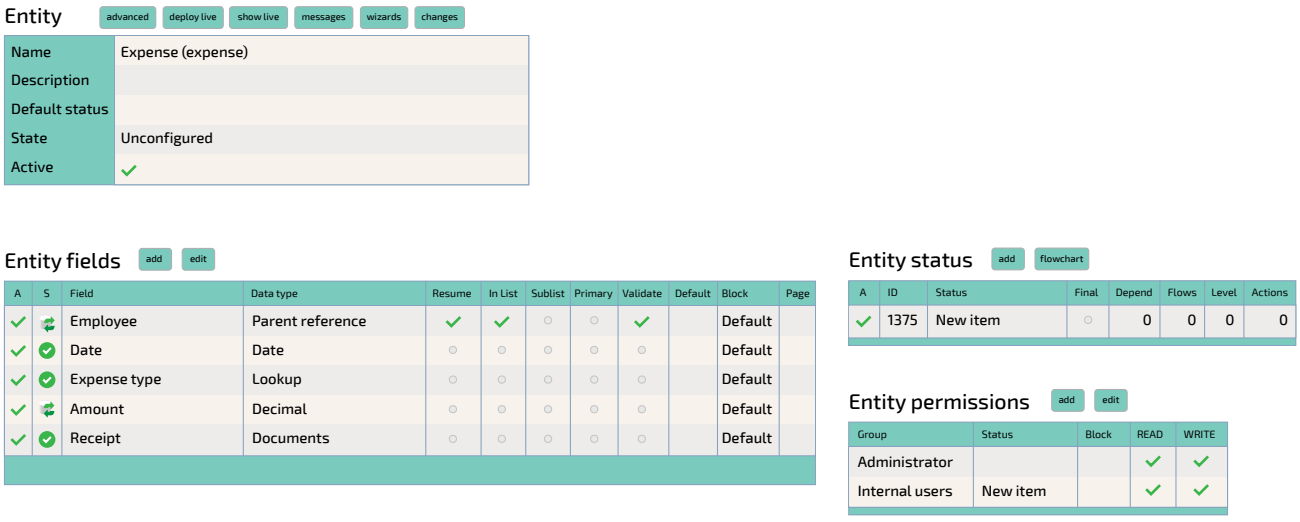


Figure 25: Entity (Expense all fields are configured and ready to be deployed).

Before we continue, consider which attributes you would like to display on each Expense Report. We suggest the following attributes: Date, Expense type and Amount. To accomplish this, click the individual Field name to open the Field dialogue and check the box “Show by default in Lists of children on parent items”.

Build the Mileage entity

Good work! You are now ready to build the last of the three entities “Milage” – a child entity – following the same process: In the backend header menu click Quick start and select Step by step from the dropdown list to open the Step-by-step dialogue. From the Section dropdown list select the section “Expense Management”. Now fill in the name of the entity (“Milage”) in the Entity name field. Next it is time to fill in the name of each attribute (Field / Data Type Component). Fill in each of the attribute names listed in Figure 3 (Entity #3) on page 7, separating them by a line break / return.

Once you have completed the list of attributes it should look like this (see Figure 26). To commit the changes, click the Add button.

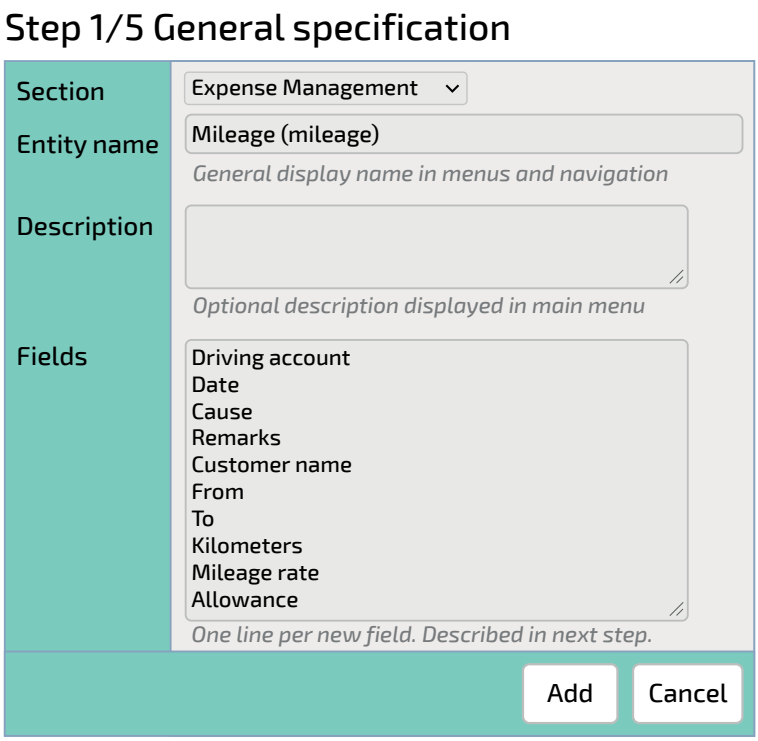


Figure 26: Quick start Step #1 (attributes defined).

This will take you to Step 2 of 5. Select the relevant Data Type Component for each attribute. In the following we will go through each of the attributes and select/assign the most suitable Data type Component for each of them.

Driving account: In order to display individual driving accounts on the parent entity “Expense Report” we will use this first attribute as a parent reference, by assigning the Data Type Component “Parent/child: Parent reference”.

Date: Each driving account must be registered on a date. We will assign this attribute the Data Type Component “Basic: Date”. This will result in a date picker for quick and easy registration.

Cause: Since driving account allowance require an eligible purpose, we will create predefined options for the employee to choose from. To accomplish this, we will assign this attribute the Data Type Component Lookup: Lookup.

Remarks: To give the employee the opportunity to add a simple note we will assign this attribute the Data Type Component Basic: Text, which will result in a single line text field.

Customer name: Often times, registration of driving accounts has to do with customer visits and therefore we give the employee the opportunity to add the name of the customer, by assigning the Data Type Component Basic: Text, resulting in a single line text field.

From: We want the employee to state where they are driving from. For this purpose, we assign the Data Type Component Basic: Text, resulting in a single line text field.

To: We also want the employee to state the destination. For this purpose, again we assign the Data Type Component Basic: Text, resulting in a single line text field.

Kilometers: In accordance with tax authority standards, kilometers are expressed as integer, and therefore we assign the Data Type Component Basic: Integer. To help employees calculate the correct travel distance, we will insert a help text with a link to the online service: https://www.ruteplanner-afstand.com Suggestion: Calculate the distance in kilometers.

Mileage rate: In accordance with tax authority standards, the current rate is DKK 3.73 pr kilometer. For this purpose, we assign the Data Type Component Basic: Decimal. Later on, we will define said rate as the default value.

Allowance: To calculate allowance for each driving account, we will assign the Data Type Component Complex: Calc Formula. Later on, we will setup this field to do some simple math for us [KILOMETERS] * [RATE] = Allowance.

When you have assigned Data Type Components to each of the attributes your list should look like this (see Figure 27).

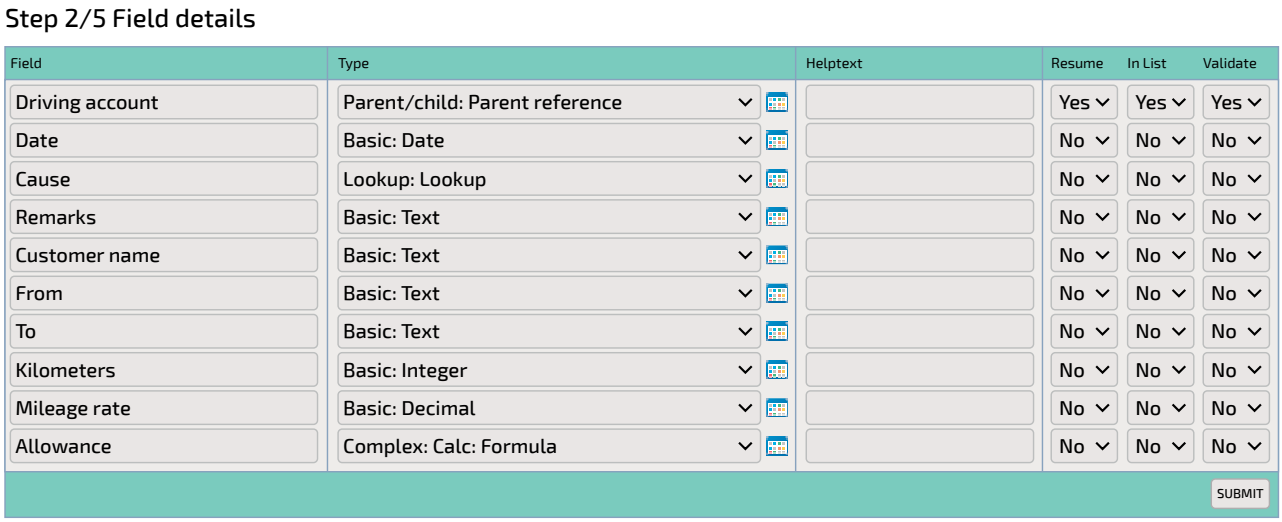


Figure 27: Quick start Step #2 (data type components defined for each attribute).

This will take you to Step 3 of 5 “State and flow”. In this step we will define the workflow (state model) that each “Milage” record will go through from creation to completion. In this case simply; New item, which is the default state name (see Figure 28). Pres Submit to continue

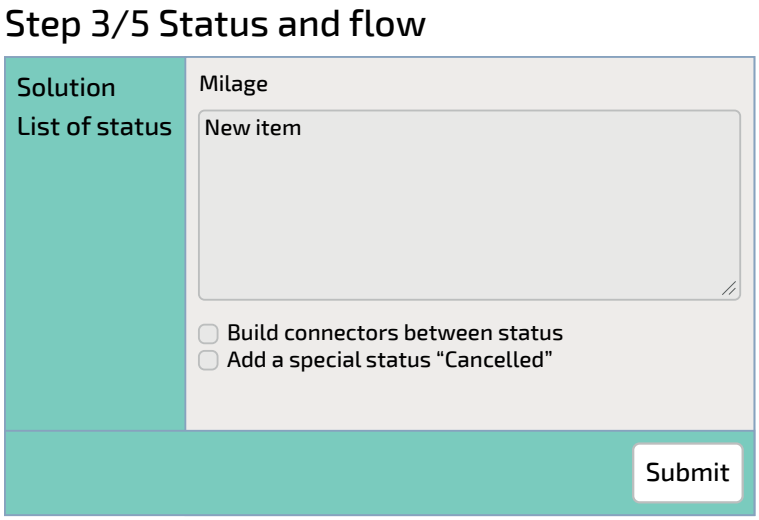


Figure 28: Quick start Step #3 (default single state workflow).

This will take you to Step 4 of 5 “Permissions”. In this step, we will define the rights or permissions of one or more user groups. In this case we will assign “Full permissions” to the default group “Administrator” and “Create new items” permission to the default group “Internal users” (see Figure 29). Permissions can easily be changed and refined later on and you can create your own user groups. Press Submit to continue.

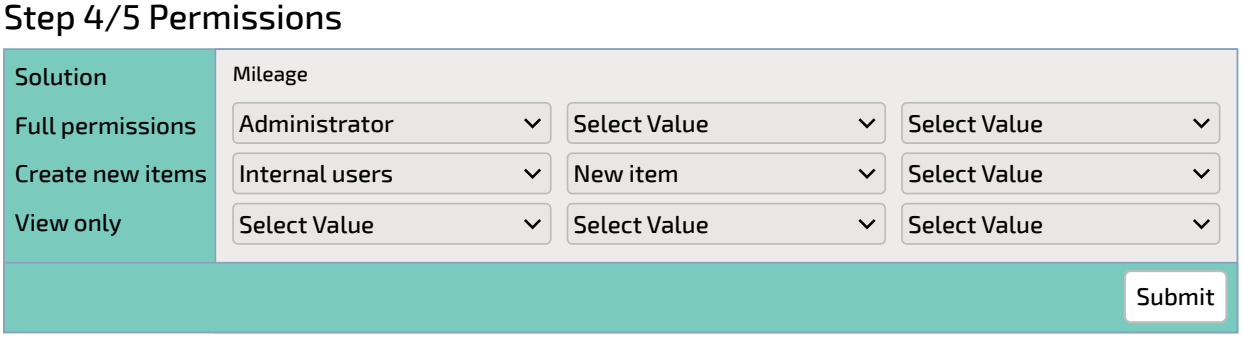


Figure 29: Quick start Step #2 (data type components defined for each attribute).

This will take you to Step 5 of 5 “Solution functionality options”. In this step we will setup additional fields or functionality options as needed. In this case we do not need any so we just keep the default settings (see Figure 30). Press Submit to continue.

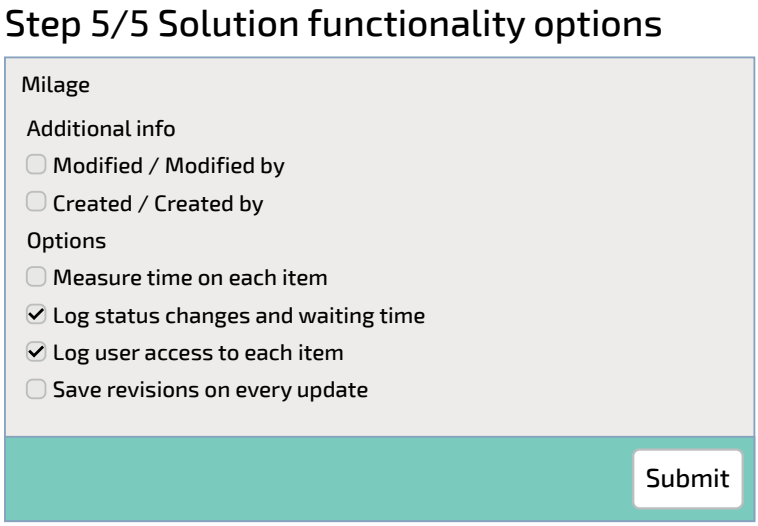


Figure 30: Quick start Step #5 (only access logging selected).

This concludes the Quick start guide and takes you to the Entity page (Figure 31). Here you will see the fields, entity status and permissions that you have defined in the previous steps. This is where you will finalize field configurations and deploy live. Any future changes or additions will be made here from the Entity page.

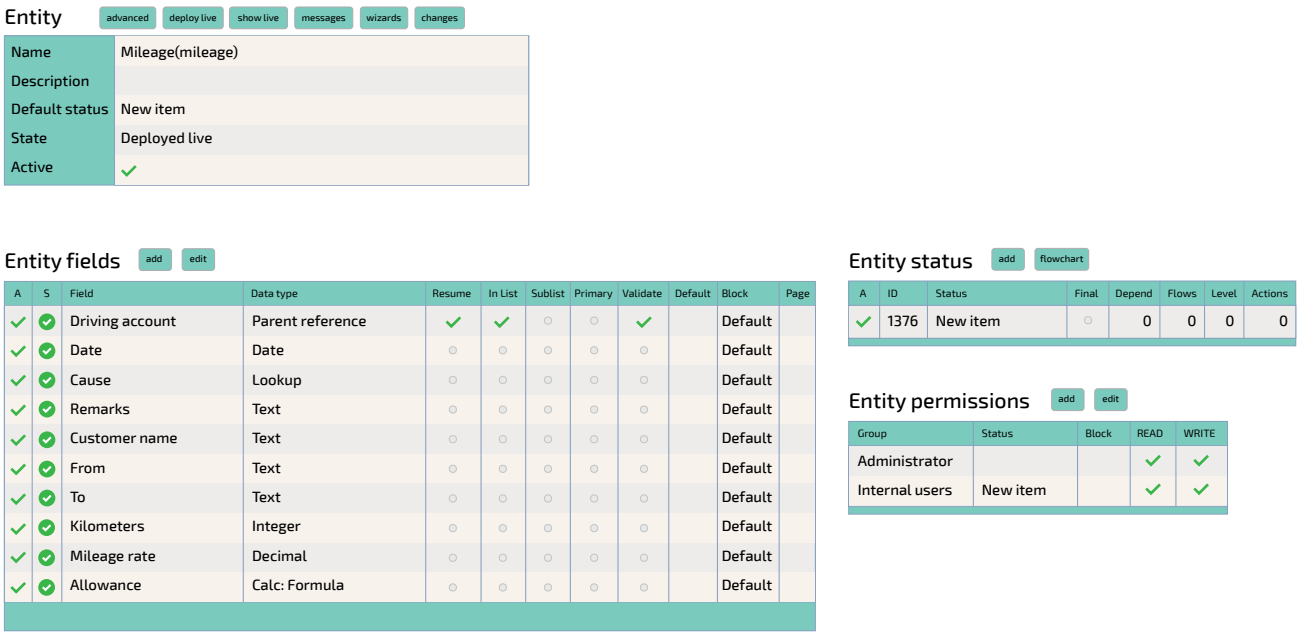


Figure 31: Entity (Milage all fields are configured and ready to be deployed).

Before we continue, consider which attributes you would like to display on each Expense Report. We suggest the following attributes: Date, Cause, Kilometers and Allowance. To accomplish this, click the individual Field name to open the Field dialogue and check the box “Show by default in Lists of children on parent items”.

Revisit attributes for additional configuration

You might remember, that for each entity we have gone through the various attributes and given an explanation of their use and the resulting choice of Data Type Component. In some cases, it was mentioned that we will return and complete the configuration later. Now is the time to do so.

In relational models we need to create the Parent reference(s) first in order to display child elements like expenses and driving accounts on the parent entity (in this case on the Expense Report). This means that first we must establish a reference or point to the parent entity from each of the child entities.

Let’s start with the Child entity Expense

This entity contains two attributes that need further configuration: “Employee” that we will use to create the parent reference and “Expense type” that we will use to create a list of options.

Employee: In order to establish the parent reference using the Employee attribute, click the field name “Employee” to open the field dialogue. From here click the button Edit configurations. This opens the Edit field configuration dialogue (see Figure 32). Select Parent solution (entity) from the dropdown list (“Expense Management: Expense Report”) and check Enable link to master. Now click Submit to save the changes.

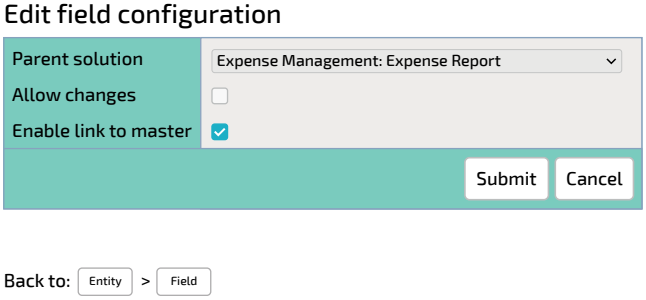


Figure 32: Edit field configuration dialogue for the field Employee.

Expense type: To create the list options; Small Purchase, Travel Expense, Catering, click the field name “Expense type” to open the field dialogue. From here click the button Edit configurations. This opens the Edit field configuration dialogue (see Figure 33). Here select List style “Dropdown list” and in the Add new values field enter the options above separated by a line break/return. Now click Submit to save the changes.

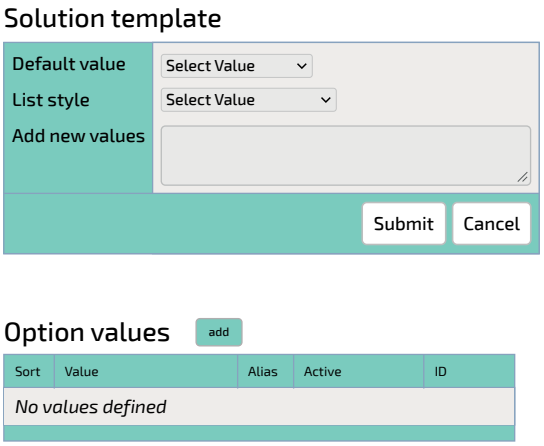


Figure 33: Edit field configuration dialogue for the field Expense type.

Next, let’s deal with the Child entity Mileage

This entity contains four attributes that need further configuration: “Driving account” that we will use to create the parent reference, “Cause” that we will use to create a list of options, “Rate” for which we will insert a default value, and “Allowance” that we will use to perform the calculation.

Driving account: In order to establish the parent reference using the Driving account attribute, click the field name “Driving account” to open the field dialogue. From here click the button Edit configurations. This opens the Edit field configuration dialogue (see Figure 34). Select Parent solution (entity) from the dropdown list (“Expense Management: Expense Report”) and check Enable link to master. Now click Submit to save the changes.

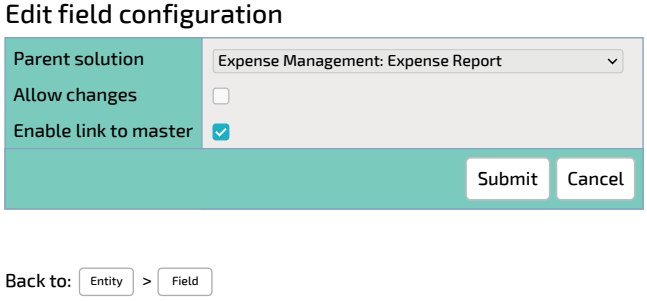


Figure 34: Edit field configuration dialogue for the field Driving account.

Cause: To create the list options; Customer Meeting, Partner Meeting, Other Meeting, Exhibition, Seminar, Course, click the field name “Cause” to open the field dialogue. From here click the button Edit configurations. This opens the Edit field configuration dialogue (see Figure 35). Here select List style “Dropdown list” and in the Add new values field enter the options above separated by a line break/return. Now click Submit to save the changes.

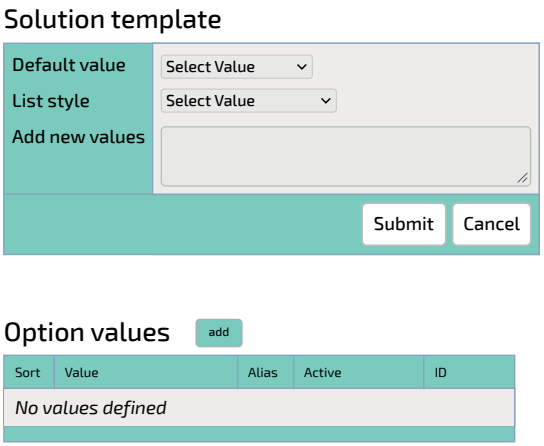


Figure 35: Edit field configuration dialogue for the field Cause.

Rate: To set a default value, click the field name “Rate” to open the field dialogue. From here click the button Edit configurations. This opens the Edit field configuration dialogue (see Figure 36). In the Default value field enter 3.73 (the official rate) and for Decimal / precision select 2 decimals. Now click Submit to save the changes.

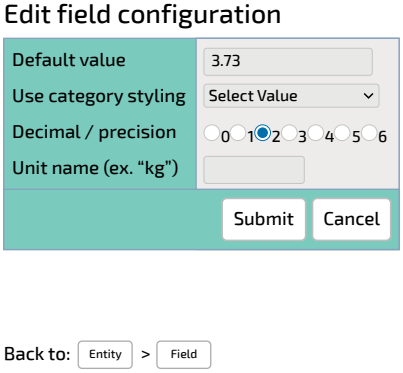


Figure 36: Edit field configuration dialogue for the field Rate.

Allowance: To calculate the applicable allowance, click the field name “Allowance” to open the field dialogue. From here click the button Edit configurations. This opens the Edit field configuration dialogue (see Figure 37). From the Display dropdown list select “Only value”, and then from the Dropdown list “Select a field to add expression”. First select Kilometers, then klick the multiply symbol “x” and then select Rate. The Formula field it should now read: [KILOMETERS] * [RATE].

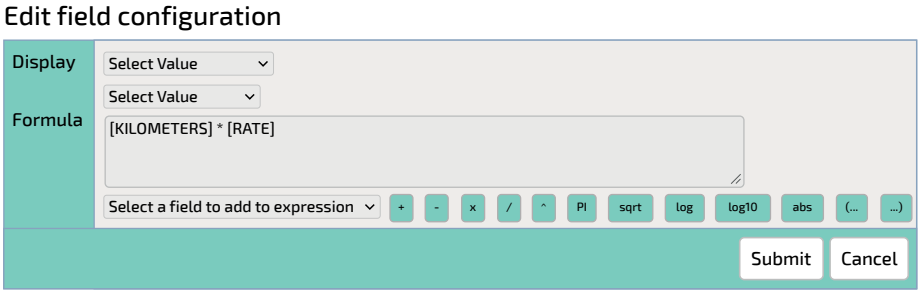


Figure 37: Edit field configuration dialogue for the field Allowance.

Finally, let’s deal with the Parent entity Expense Report

This entity contains five attributes that need further configuration: “Expenses” that we will configure to display a list of children – meaning the individual expense records, “Expenses Total” that we will configure to display the Total amount of the expenses. “Driving in own car”, which we will configure to display a list of children – meaning the individual mileage records, “Kilometers in Total” which we will configure to display the calculated total sum, and “Driving Settlement Total” that we will configure display the calculated total driving settlement.

Expenses: To display a list of the individual expenses, click the field name “Expenses” to open the Field dialogue. From here click the button Edit configurations. This opens the Edit field configuration dialogue (see Figure 38). Now, select the Child solution key “Expense Management > Expense > Employee” from the dropdown list, to define the child entity to display. Then select New items action “Button: Below table” to add a create new item button, and click Submit to save the changes.

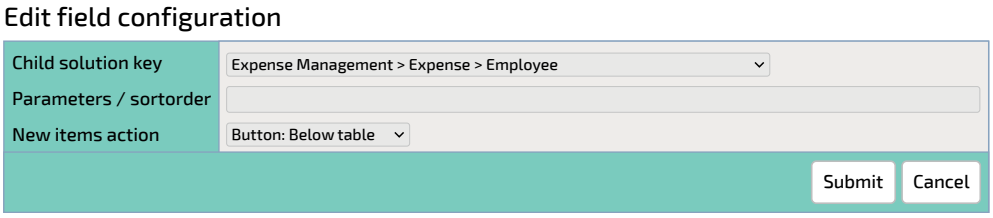


Figure 38: Edit field configuration dialogue for the field Expenses.

Expenses Total: To display the calculated total of expenses on each Expense Report, click the field name “Expenses” to open the Field dialogue. From here click the button Edit configurations. This opens the Edit field configuration dialogue (see Figure 39). Now, select the Child solution key “Expense Management > Expense > Employee” from the dropdown list, to define the child entity to display, and specify in the Child value expression which field value to display. In this case we want to display the calculated Amount. To do so enter the system name (always capital letters) “AMOUNT” in the Child value expression field. Click Submit to save the changes.

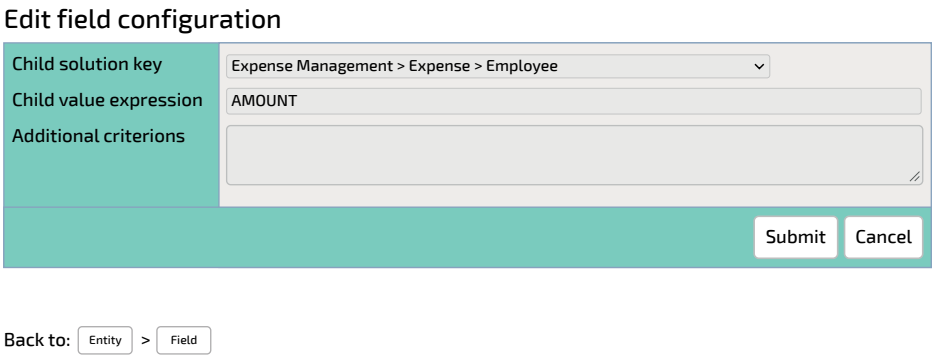


Figure 39: Edit field configuration dialogue for the field Expenses Total.

Driving in own car: To display a list of the individual driving accounts, click the field name “Driving in own car” to open the Field dialogue. From here click the button Edit configurations. This opens the Edit field configuration dialogue (see Figure 40). Now, select the Child solution key “Expense Management > Mileage > Driving account” from the dropdown list, to define the child entity to display. Then select a New items action “Button: Below table” to add a create new item button, and click Submit to save the changes.

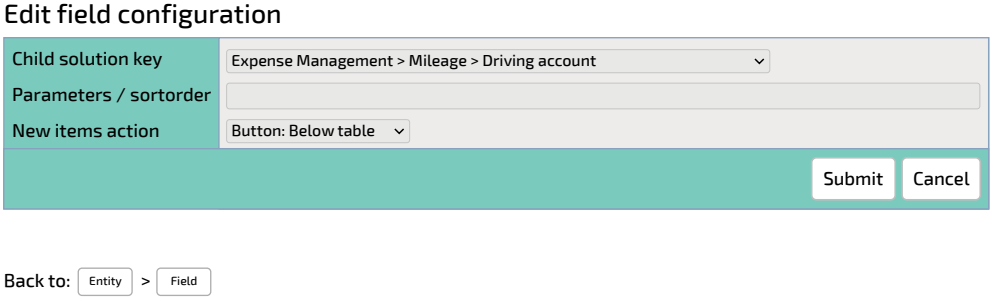


Figure 40: Edit field configuration dialogue for the field Driving in own car.

Kilometers in Total: To calculate the total amount of kilometers based on each mileage record, click the field name “Kilometers in Total” to open the Field dialogue. From here click the button Edit configurations. This opens the Edit field configuration dialogue (see Figure 41). Now, select the Child solution key “Expense Management > Mileage > Driving account” from the dropdown list, to define the child entity to display, and specify in the Child value expression which field value to display. In this case we want to display the calculated Kilometers. To do so enter the system name (always capital letters) “KILOMETERS” in the Child value expression field. Click Submit to save the changes.

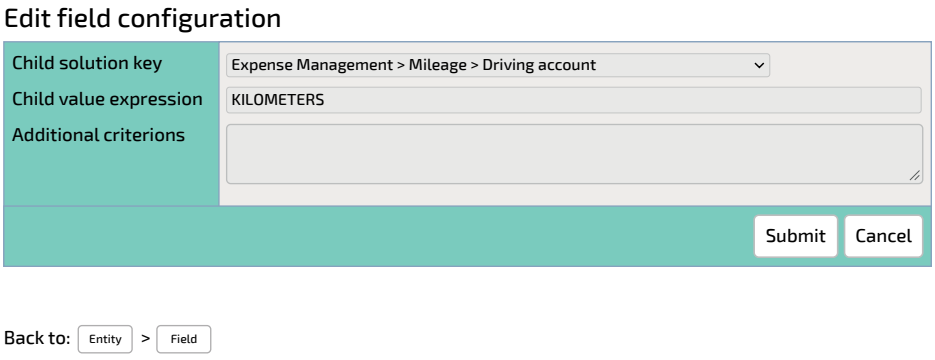


Figure 41: Edit field configuration dialogue for the field Kilometers in Total.

Driving Settlement Total: To calculate the applicable settlement, click the field name “Driving Settlement Total” to open the Field dialogue. From here click the button Edit configurations. This opens the Edit field configuration dialogue (see Figure 42). Now, select the Child solution key “Expense Management > Mileage > Driving account” from the dropdown list, to define the child entity to display, and specify in the Child value expression which field value to display. In this case we want to display the calculated Allowance. To do so enter the system name (always capital letters) “ALLOWANCE” in the Child value expression field. Click Submit to save the changes.

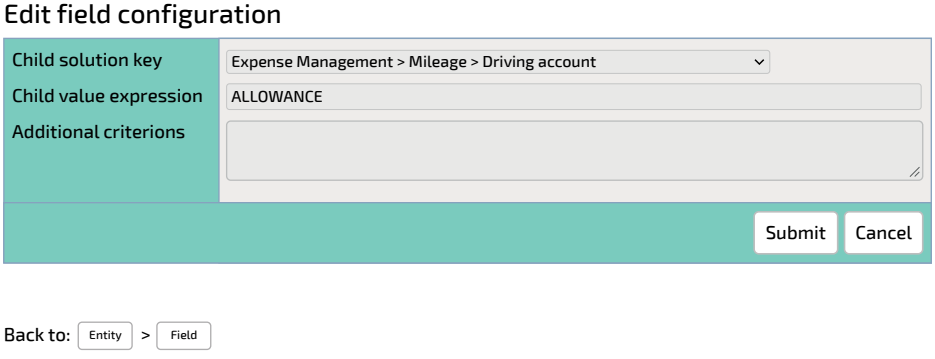


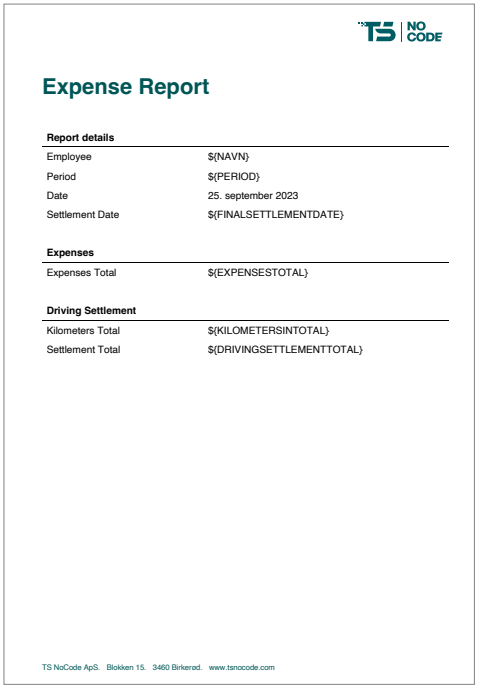
Figure 42: Edit field configuration dialogue for the field Driving Settlement Total.

Good work! Now you have built all three entities: “Expense Report” our parent entity, and “Expense” and “Mileage” our child entities, that comprise the Expense Management application. If you have completed the configurations correctly and every field has been deployed live, you can go to the front end and test your application.

Creating the Expense Report Template

Once you have confirmed the basic functionality of your application. It is time to create and attach the “Expense Report” Document Template to the [Print] button and setup an automated action on the “Ready for Accounting” state, that will generate and send the Expense Report to Finance (ex. finance@domain.com) when an Expense Report record is saved in this state/status. If needed, see “**TS User Guide - Working with Templates**” for more details before you continue.

- Open a blank MS Word document.
- Create the structure of your document
- Apply desired layout and styles
- Add the equivalent field tokens to Include dynamic data based on Expense Report records: ${NAME}, ${PERIOD}, ${FINALSETTLEMENTDATE} , etc.



Expense Report

Report details	
Employee	${NAVN}
Period	${PERIOD}
Date	25. september 2023
Settlement Date	${FINALSETTLEMENTDATE}

Expenses	
Expenses Total	${EXPENSESTOTAL}

Driving Settlement	
Kilometers Total	${KILOMETERSINTOTAL}
Settlement Total	${DRIVINGSETTLEMENTTOTAL}

TS NoCode ApS. Blokken 15. 3460 Birkerød. www.tsnocode.com

TIP: MS Word formatting has an unfortunate tendency to apply additional styling classes that may break field tokens. To avoid this, complete your template document with the desired layout, fonts and font styles etc. and then paste in field tokens as plain text using the “Keep text only” paste option in MS Word.

Now that you understand the fundamentals of the document template creation process, you are ready to go ahead and create your first template “Expense Report”. Go to Entities > Expense Report, to open this entity. Click the [advanced] button to open advanced entity settings, and toggle the switch to see all settings (see Figure 43).

TIP: In this screen, a template named Expense Report with the format PDF has already been added. In your case you will have to add a template to the entity. We will deal with that next.

Figure 43: Advanced settings for the entity Expense Record.

In the advanced settings panel “Templates” click the [add] button. This will open the Entity/Solution Template dialogue (Figure 44). From the Format droplist select the option MS Word (docx). Click the [MS Office] button to see MS Office specific tokens in the correct syntax. Example: ${NAME}. Now in MS Word, copy the relevant field tokens into your Document Template (see Figure 45).

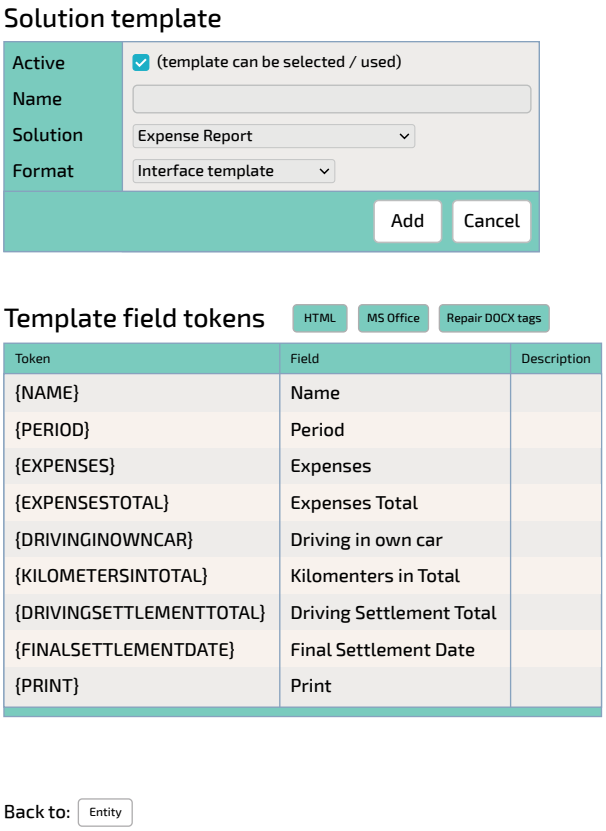


Figure 44: Template field tokens (specific to the entity).

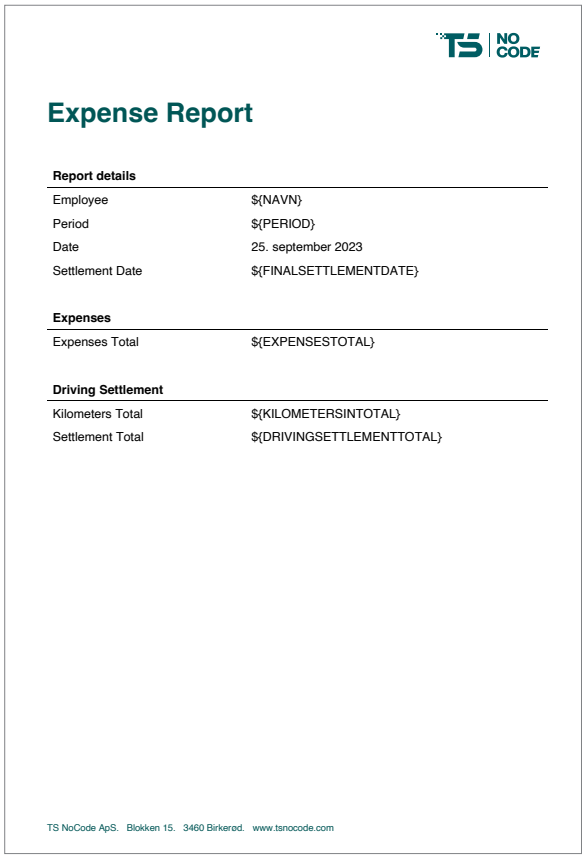


Figure 45: Example Template Document (MS Word).

Once your Document Template is completed, go ahead and add/attach it to the Expense Report entity. If you have already added/attached a template and you simply wish to update it with the latest changes. Open advanced settings of the entity. In the Template settings panel click the name of the template you want to update.This opens the Solution Template dialogue (see Figure 46).

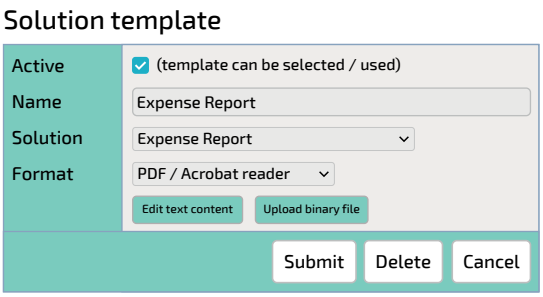
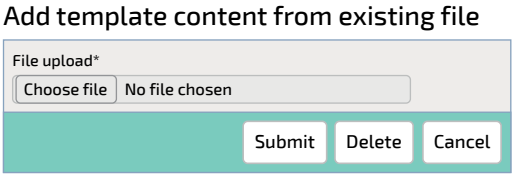


Figure 46: Solution template dialogue for the entity.

To update at document template, cick the [Upload binary file] button.

This opens the Template Upload Dialogue (see Figure 47), alowing you to drag and drop or point select a file from your computer.



Type	File / program type	Extension	Syntax	Notes for usage
Text	HTML / XHTML document	.html	{fieldname}	Full support for all content types
Text	Rich text format	.rtf	[fieldname]	Support for tables, pictures, links and styling
Text	XML markup	.xml	{fieldname}	Values only
Text	Outlook / Thunderbird message	.eml	${fieldname}	Full support for all content types
Binary	MS Office Word 2003	.xlsx	${fieldname}	Support for tables, pictures, links and styling
Binary	MS Office Powerpoint 2003	.pptx	${fieldname}	Values only
Binary	Acrobat reader / PDF	.docx	${fieldname}	Same as docx files + conversion to pdf files

* Files not supported by the template engine will not be allowed to upload. Maximum size is 10 MB.

Back to: Entity

Figure 47: Template Upload Dialogue. No file selected.

Good Development Practice

We recommend that you build any and all applications using English naming for every field, state, help text, headings, dashboard widgets, etc. and then make use of the multi-language features of TS No-code Platform to translate your application into local language. If you have followed the naming conventions used in this guide you will have named everything in English. This means that you can simply go to each of the three entities in your application, click the advanced button and then set the default language to English.

Translate the application into your local language

Since both TS No-code Platform and any TS Application is UTF8 encoded, you can create one or more translations in order to serve up the languages you need. To create a translation, go to each of the entity pages of your application, click the [wizards] button and then click the [Internationalization] button. This opens the Internationalization wizard. Here you will be able to provide translations for each of the following elements:
Solution (Entity) > Field > Page > Status > Lookups.

Example: to create a Danish translation prefix each text string with **da:** “da:Udgift”. To create a German translation prefix each text string with **de:** “de:Kosten”.

For more information about the multi-language features in TS No-code Platform and how to apply translations for your applications see: TS User Guide / Multi-language features.

We
Democratize
Digitalization

TS NoCode ApS Blokken 15 3460 Birkerød Denmark Phone: +45 3150 7377 www.tsnocode.com