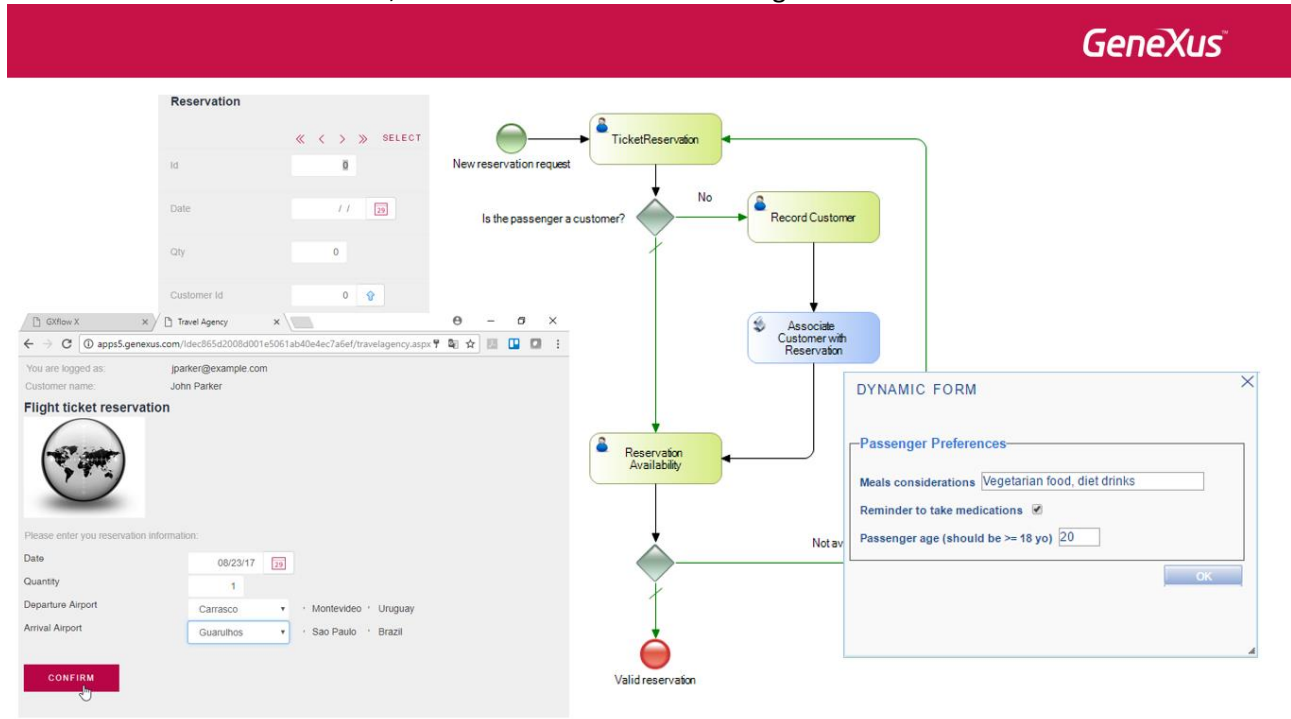


## Entering data with Dynamic Forms

In general, to execute an interactive task in a business diagram process we must have in our KB a transaction object or a web panel to be associated with the task.

Dynamic forms enable us to create data entry screens directly from the workflow client and associate them with interactive tasks in runtime, without the need to write a single line of code.



A typical scenario where we use these forms is the case when we get a new requirement for data entry and the system is already in production. Instead of re-designing the application, we could create dynamic forms and adapt them in runtime to store information, without the need to modify the application's database.

Another example could be, whenever we need to store data relative to a given process that was not provided for in the original workflow, to include information that relates only to each instance in the process. Dynamic forms enable us to store such information in workflow tables, without the need for structures in the database for this type of information.

Let's explain these concepts by developing an example.

Suppose that the travel agency needs to record some preferences of passengers who hold a ticket reservation. The idea is to record the following:

- Special aspects regarding meals (for example: sodium-free, vegetarian, diet beverages, etc.)
- The passenger needs to be reminded as to specific medicine intakes, or
- We must record the passenger's age, specifically, to confirm the passenger is at least 18 years of age.



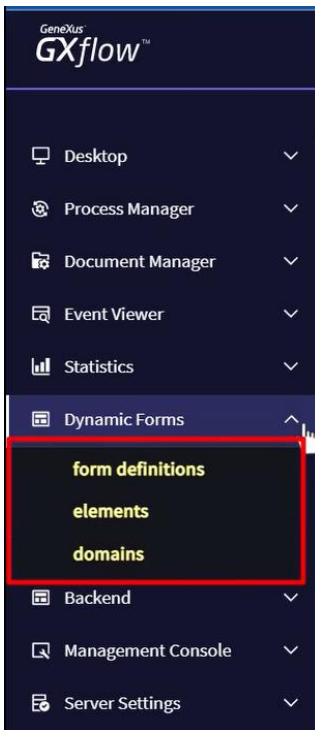
- Meals considerations
- Remainder to take medications
- Passenger age registration

Since we don't want to store this data in the system's database –because it might depend on the passenger's preference for each particular flight–, we will request these details through a task associated to a dynamic form, during the reservation process, upon confirmation.

We will need certain permits to access the creation and modification of dynamic forms. So, in GXflow client, we go to Management Console, select Users, and in the Users window we press the Roles button, check Gxflow Form Designer, and press Ok.

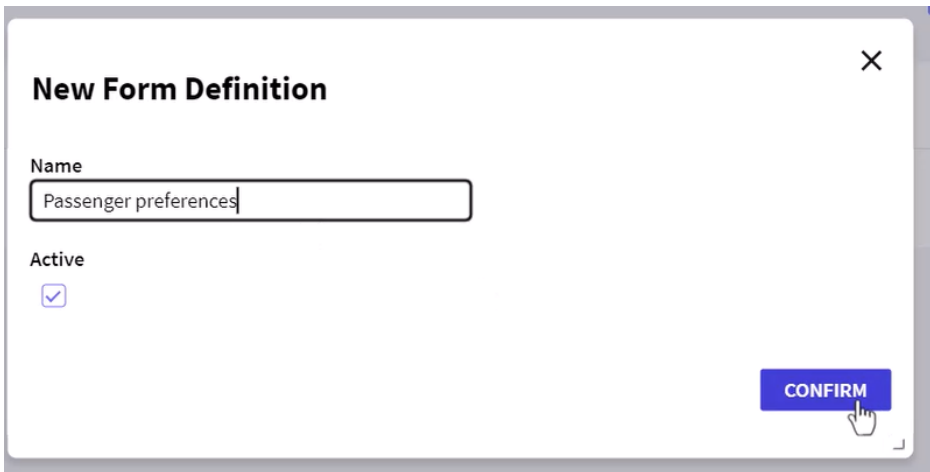
We now close the session and log in again as WFADMINISTRATOR.

We can see that, in the Browser, and under statistics, we now have a group, called Dynamic Forms, with three components: the **definitions of dynamic forms** we determine, the **elements** of the forms defined, and the **domains** used in the definition of the elements in the forms.

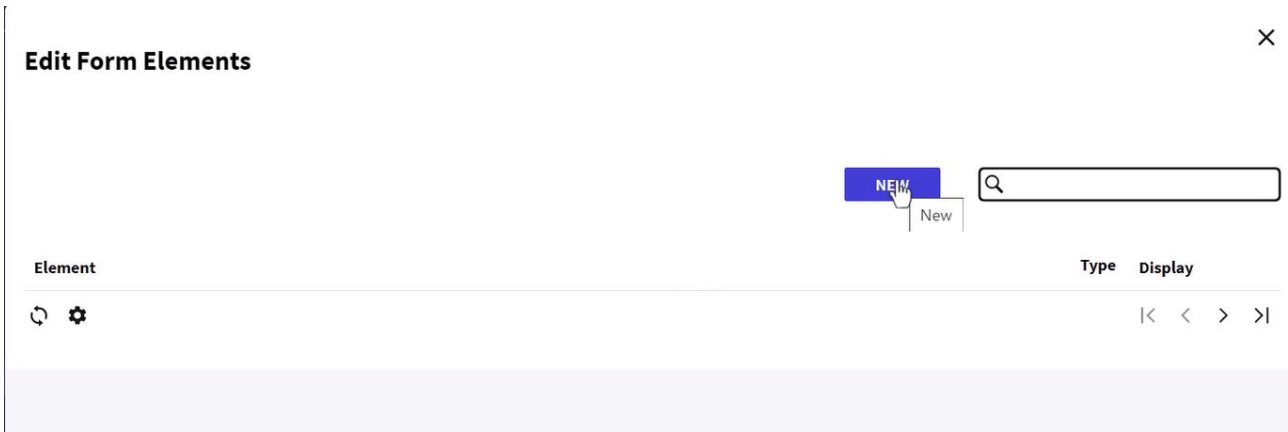


To view these concepts we implement a dynamic form with what the travel agency has requested. We click on Form Definitions, and press the New button.

We call it **Passenger Preferences** and press Confirm.



Now we will define the elements we will include inside the form. We select the Passenger Preferences form and press the Elements button, and we will see that the form window opens up, so we move the mouse inside the frame and press the "New" button.



We get a dialog box to enter the element.

In this element, we will store the details regarding the passenger’s meals, so we enter the name “Meals considerations”, and in the description we enter: “Special requirements for passenger food” We modify the Character type so that it is 60-character long. We leave the remaining fields with their default value and press Confirm.

We now see the form with the first element of the text type already entered. We press the New button “+” sign again, and enter a new element to record whether the passenger wishes to be reminded as to any medicine intake. For the name we write “Reminder to take medications” y en la descripción “Indicates if the passenger wants to be reminded to take medicines.”

**Edit Form Element** [X]

Element: (None) ▾

Name: Reminder to take medications

Description: Indicates if the passenger wants to be reminder to take medicine

Domain: (None) ▾

Type: Boolean ▾

Display: Default ▾

Default Value: [Empty text area]

[OK]

We leave the value of the type in Boolean and press Confirm.

And last, we will create one more element in order to record the passenger's age.

We press the green button and call it "Passenger age – should be over 18", and in the description: "Verify if passenger is over 18 years old". We select the Numeric type, 2 characters in length, and with no decimals.

**Edit Form Element** [X]

Element: (None) ▾

Name: Passenger age-should be gt 18 yo

Description: Verify if the passanger is more than 18 yo

Domain: (None) ▾

Type: Numeric ▾

Length: 2

Decimals: 0

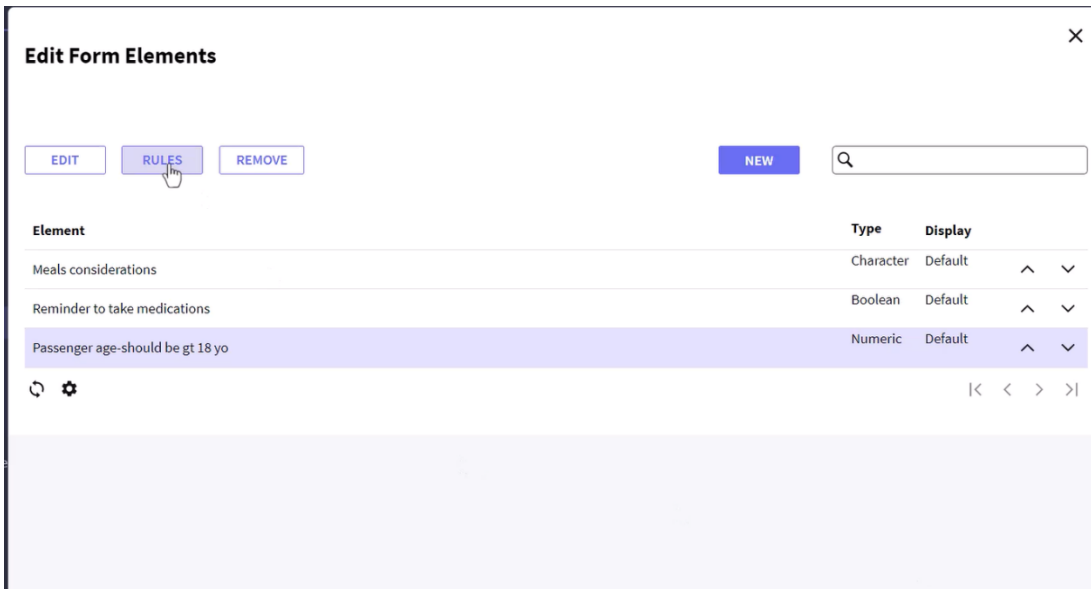
Display: Default ▾

Default Value: [Empty text area]

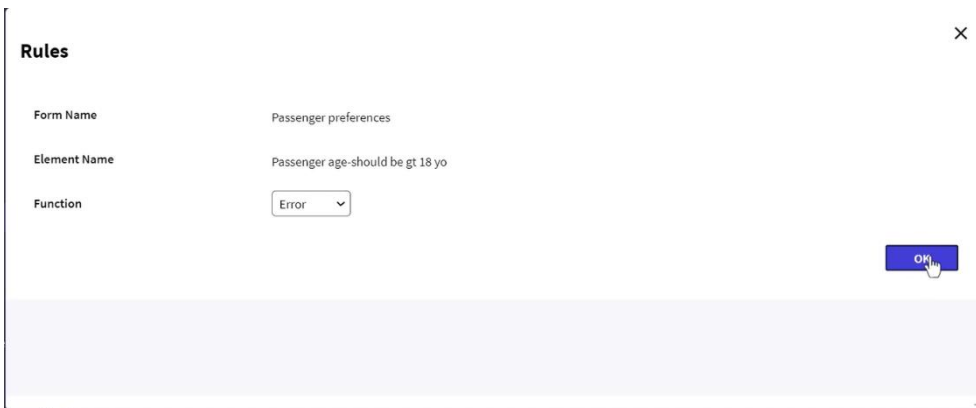
[OK]

We press Confirm and we will see the element added to the form.

We will now add a rule so that this control is done automatically. We go to the Rules tab...



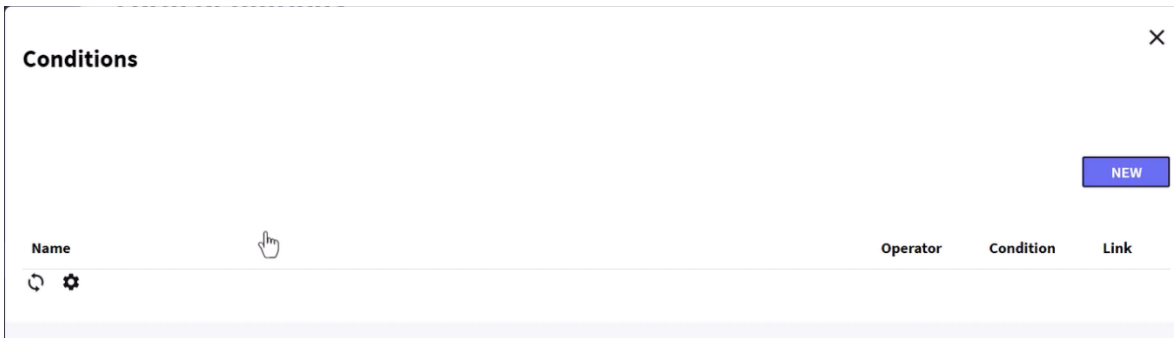
And press the **RULES** button. Then we select the Error function and press Confirm.



Now we can see that a rule with the description "error" has been created. We select it and press the **CONDITIONS** button.



We can see that the window opens up to enter conditions.



We now press the New button and have a window open to specify the condition's details.

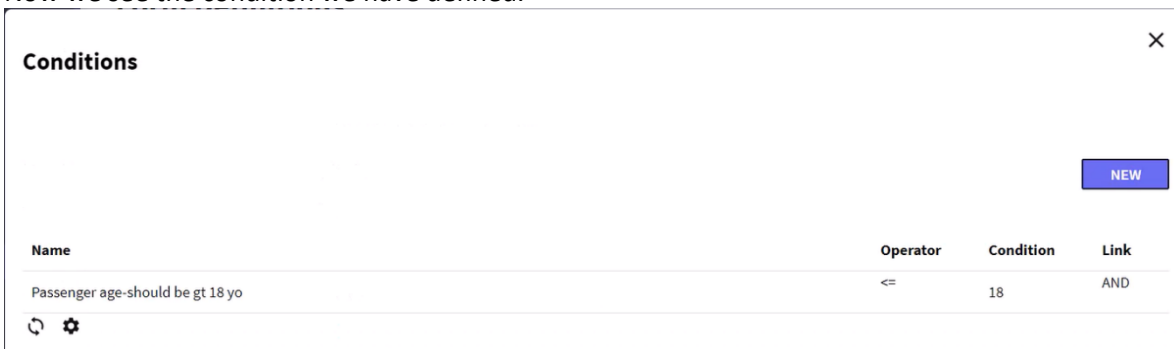


First, select the element to which we want to add the validation rule, "Passenger age" in our case.

Now we select the less than or equal to operator, and set 18 as value and select AND in the link. Then we press Ok.



Now we see the condition we have defined.

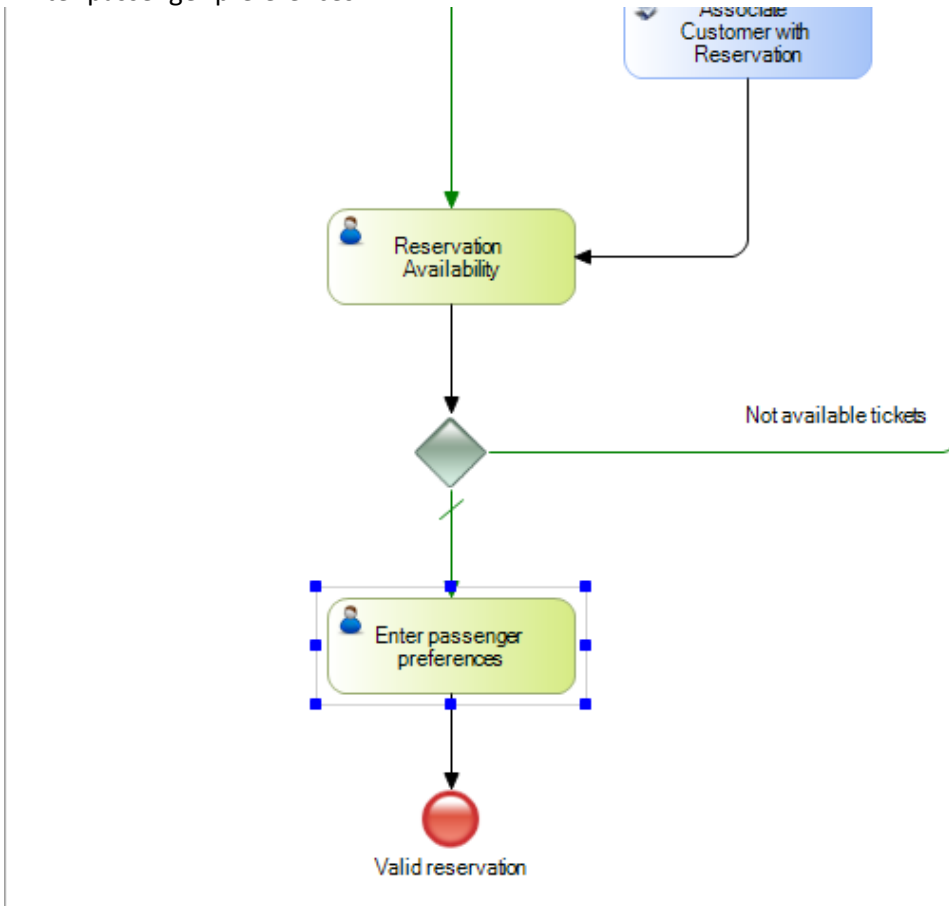


We already have our form defined, and ready to be associated with an interactive task.

So, we open the FlightTicketReservationSD diagram and save it under the name:

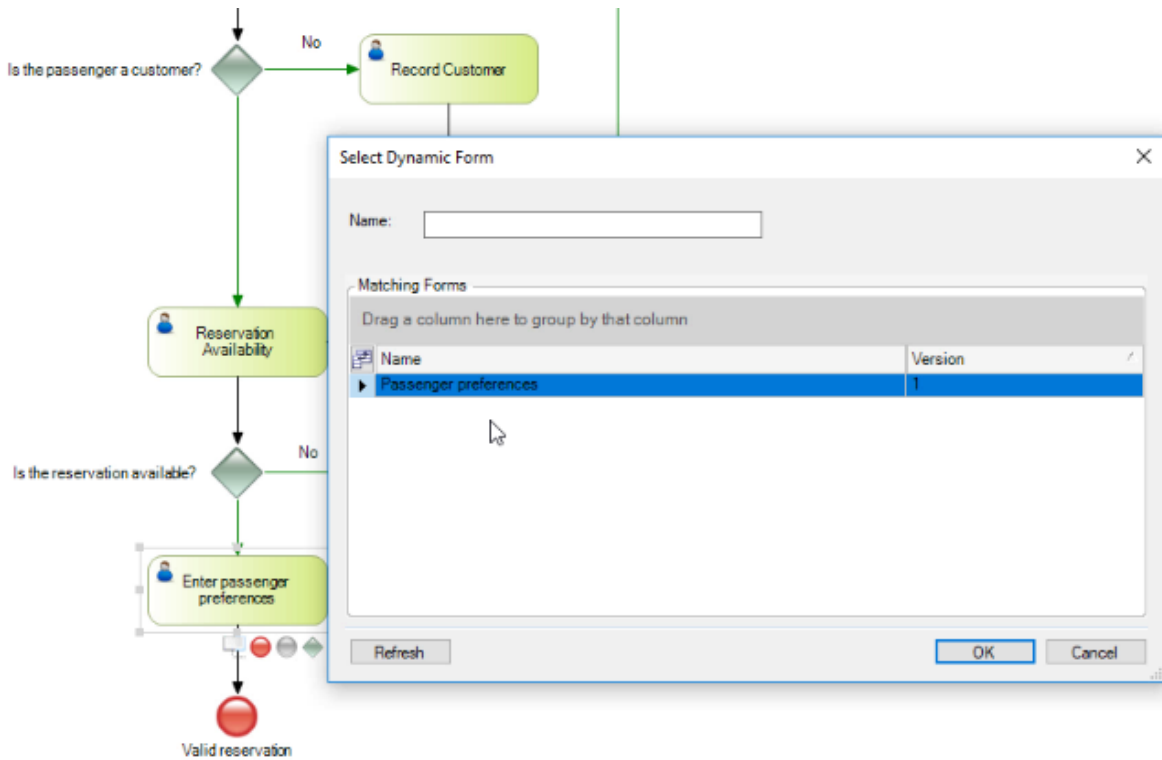
FlightTicketReservationDynForms.

Now, after validating the reservation but prior to the *None End Event*, we add an interactive task called “Enter passenger preferences”.

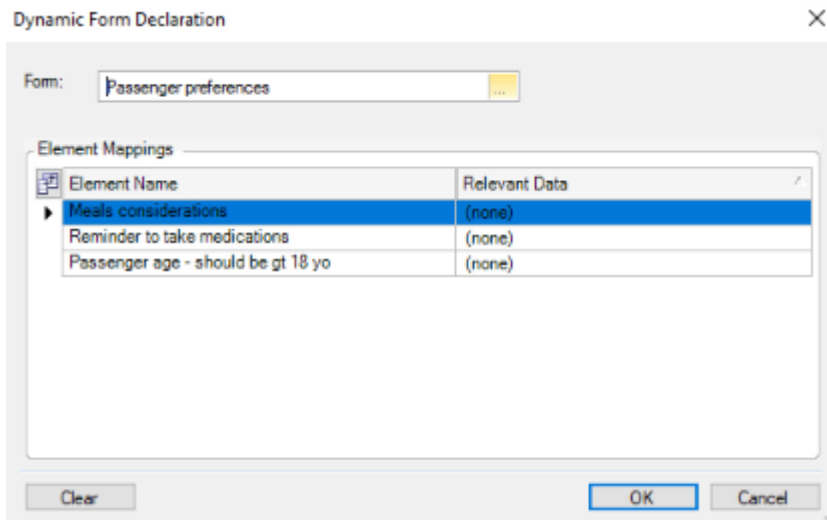


In the task’s properties we go to the Dynamic Forms group and press the button corresponding to the property *Application form*. Now we press the button in Forms, select the *Passenger Preferences* form, and press OK.



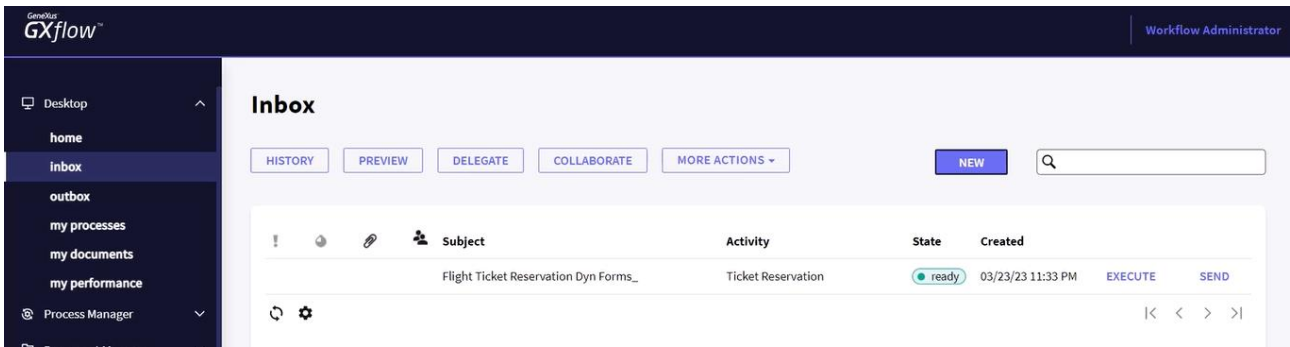


We get the form's elements to have the possibility to map them with some relevant data in the diagram.

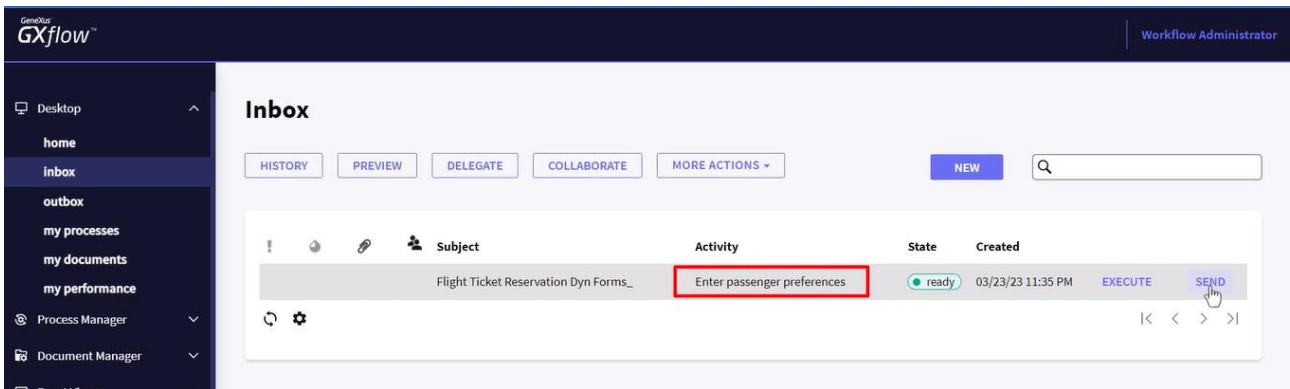


And we press OK again.

To execute the diagram, we right click on it and select Run. As the GXflow client opens up, we can see that the TicketReservation task of the FlightTicketReservationDynForm process appears as pending.



So, we select the task, and execute. To continue with the execution, we enter a reservation by selecting a customer... and now we set the reservation as available. We can see that the following task is **Enter passenger preferences**, so we execute it.



And the Passenger Preferences form is opened, where we will enter data...

The 'Passenger preferences' form is shown. It has three main input areas: a text field for 'Meals considerations', a checkbox for 'Reminder to take medications', and a text field for 'Passenger age-should be gt 18 yo'. At the bottom, there are 'CANCEL' and 'CONFIRM' buttons.

In Meals considerations, we enter: "Vegetarian food, diet drinks", and check the reminder for medicine intake, and also enter the age of 15. We then press Confirm.

We will see the rule triggered, sending us an alert that the value entered as age is not valid, because it must be 18 or greater.

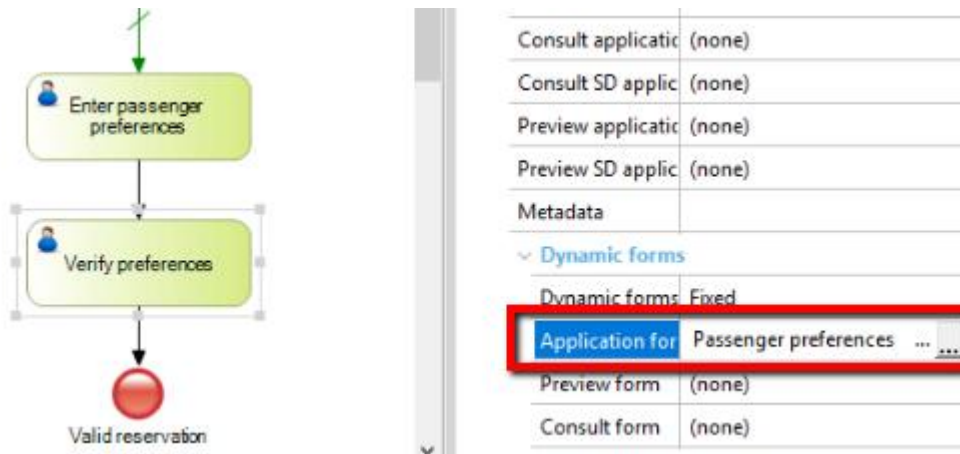
The 'Passenger preferences' form is shown again, but now with data entered. The 'Meals considerations' field contains 'Vegetarian food, diet drinks', the 'Reminder to take medications' checkbox is checked, and the 'Passenger age-should be gt 18 yo' field contains '15'. A red error message is displayed below the age field: 'Passenger age-should be gt 18 yo 7 <= 18'. The 'CONFIRM' button is highlighted in blue.

We now enter a value of 25, and press Confirm, and we will see that the form is closed, with the input tray remaining empty.

The data we have stored is now saved in the workflow tables, and available in the instance of the process.

In our example, the process ends just after that data entry, but if the process were to continue, with another task associated with the same form, it would be opened with the data entered previously.

To prove this, we will modify the process diagram and add a new task, called **Verify preferences**, between the **Enter passenger preferences** task and the *None End Event*. We associate the task we just created to the dynamic form **Passenger preferences**, and execute the process diagram.



We execute the previous tasks.

And now we execute the Verify preferences task, and find that when the associated form is opened, all the fields in it contain the data we entered previously when the task was executed to enter the user preferences.

The data entered on the form is stored in the Workflow database, and will be available in the whole instance of the process.

In this video we have seen the applicability of dynamic forms, which enable us to easily define screens for data entry in runtime, without the need to modify the application's database or write any lines of code.

Further details on dynamic forms are available at the link shown on screen.

<https://wiki.genexus.com/commwiki/servlet/wiki?25809,GXflow+Dynamic+Forms>